Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Mail Scheme Log Processing Based on ELK.

Bu Yun [a)]

School of Computer Science &Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a)] Corresponding author: 404759692@qq.com

Abstract. With the continuous development of Internet technology, how to deal with and analyze a large number of data has become a hot spot. The mail system generates a large number of logs every day, and the traditional technology is not efficient in handling huge log data and is unable to make use of the information in the log. Proposing an information processing architecture based on ELK for mail logs to solve these problems. It ex-tracts information from logs by regular expressions, and define the concept of mail events, modeling data and storing them in graph database. The graph database is stored with the original graph of the data. When dealing with a large number of network relationships, it avoids the consumption of data connection in the traditional relational database. The experiment proves that the scheme can realize real-time processing and modeling storage of large moduli data and meet the needs of mail system.

Key words: ELK; mail system; hot spot.; original graph; Internet technology.

INTRODUCTION

With the advent of the information age, e-mail has become an indispensable means of communication because of its convenience, speed, and cheap-ness. Users send emails frequently. The mail server generates a large number of logs. These logs contain a lot of valuable information. They record people's previous communication networks, communication habits, and even living habits. Mail is the medium for transmitting information.

The effective analysis and processing of the mail log is an important task for the operation and management of the mail system. The mail server generates a large amount of data every day. Since most of the mail logs (such as smtpd, pop3, etc.) are not only large in data size, they also look obscure. In the face of the discovery of mail anomalies, and the need to check the delivery status of mail, if only relying on the manual work of the administrator to view the log records, each time a message is queried, it takes a minute or two, when the demand slightly increases. Large, the workload is very heavy, and the operation is inefficient and error-prone.

The mail communication network is a complex flow network, similar to the social dynamic network diagram, without a fixed main structure. Everything has been continuously developed and updated over time [1]. In order to achieve rapid search and mining of these data in real time. This article provides a solution for processing mail logs, aiming to extract fragmented mail information for modeling, making it easier to use the information in the logs for data analysis and research.

RELATED WORK

The Status of Email Log Research

In recent years, the processing and analysis of email logs has been one of the hot topics for researchers. Using email interaction data to mine user behavior patterns, Li Quangang et al. used Enron public data in the literature [1] to extract the structural features and functional characteristics of the mail network and used non-negative matrix factorization to calculate the basic behavioral units of the network, using vectors to represent User behavior pattern

[1]. Yang Zhen et al. [2] also used the improved EM algorithm to determine mail labels in the Enron mail network. According to the interaction strength between users, a collaborative filtering mechanism was designed to filter spam [2]. Hu Tiantian et al. used JavaMail to parse the data in the literature [3], and then built a mail network, calculated the weighted center degree according to the node's connection center degree, closeness center and middle center degree, and excavated the modularity indication to mine the core community. [3]. Chen Bin et al [4] used the mail transfer protocol session log to analyze the behavior of the host based on the failed message in the log record and used the incremental passive attack learning algorithm to effectively adjust the host of the detected spam host. Recent mail classification behavior [4].

The massive data processing generated by the mail server is often not a single node in the traditional technology. The distributed software processing framework provides a feasible solution to solve the impact brought by the information wave. Zhang Jianzhong and his colleagues used the ElaticSearch distributed indexing technology to perform distributed indexing and retrieval of resources in the literature [5]. The HDFS distributed file system was used to implement the university library resource retrieval system [5]. Bai Jun et al [6] proposed a software integration scheme based on ElasticSearch real-time large log data search. The experimental results show that with the increase in the number of logs, does not affect the search response time, indicating the feasibility of this program.

Framework Introduction

With the increase of data processing capacity, the storage, computing capacity and processing efficiency of a single node cannot meet the requirements of application scenarios. Traditional methods based on relational database management systems cannot handle analysis problems efficiently.

ELK, which is a data processing tool chain consisting mainly of three open source software, Elasticsearch, Logstash, and Kibana, implements distributed and scalable data storage and search. It is a zero-configuration and easy-to-use full-text search mode, supporting distributed processing and supporting systems. Extensions.

Elasticsearch, as an open source distributed search and data processing platform, is not only a database, but also an open source, distributed, RESTful-based information retrieval framework built on Lucene that enables real-time search, efficient retrieval, and adoption of JSON data formats. The Ruby DSL design pattern provides Aggregations-based statistics capabilities, while providing easy deployment and setup. The cluster can be easily extended to hundreds of servers to handle structured or unstructured data at the PB level, but it can also run on Single PC [7].

Logstash can collect, analyze, and convert related network logs, store them for later use, store them in Elasticsearch, and convert/store them to other destinations. Logstash itself does not generate logs. It is only a pipeline that accepts a wide variety of log input and is processed and forwarded to multiple different destinations [8].

Kibana can help aggregate, analyze, and search important data logs and provide a friendly visual interface.

As one of the emerging NoSql, Neo4j is currently the most popular graphics database. It stores data in the form of nodes, edges, attributes, and graphs. It provides transaction operations similar to traditional databases for highly connected data, and at the same time It is also several orders of magnitude higher than traditional databases. For a meshed data structure, it turns out to be an ideal choice for dealing with complex data.

APPLICATION IMPLEMENTATION

Preprocessing of data Figures

An e-mail system is mainly composed of three parts: user agent, mail server, mail sending protocol (SMTP) and mail reading protocol. Log in to the email account, log out, delete emails, send emails, receive emails, and delete emails. These operations are logged. Taking the campus mail system as an example, there are 760,000 lines per day for access logs, and up to several million lines for passing logs.

The data in the real world is incomplete, in-consistent, and most of the data is unstructured or semi-structured and cannot be used directly. In the experiment, incomplete log records were filtered out, and the daily generated logs were imported into Elasticsearch in JSON format. Visualized by Kibina, the log format in this experiment was as follows.

@timestamp	August 8th 2016, 09:23:29.000
@version	1
_id	AVayOSR3TNtNPyoH9qaj
_index	mail-2016.08.08
_score	-
_type	logs
host	DESKTOP-H3IGCI6
message	Aug 8 09:23:29 mta02 postfix/lmtp[4584]: DE27D262403: to=<nash@sjtu.edu.cn>, relay=mailstore08.sjtu.edu.cn[202.121.179.18]:7025, delay=0.11, delays=0.02/0/0/0.08, dsn=2.1.5, status=sent (250 2.1.5 Delivery OK)
path	D:\NELK\data\sample.txt
timestamp	Aug 8 09:23:29

FIGURE 1. visual logs in Kibana

This article selects the log of the message. The message contains the time of the operation, the name of the mail server, the action record, the ID of the current server, the email address, and the opera-tion status. By parsing the information of the mes-sage, you can understand the dynamic behavior of the mail in the current server.

The Definition and Structure of Mail Events

In order to effectively organize the data in the log, the event definitions in the mail log are given below.

Definition 1: A complete mail event refers to the process of sending and receiving a mail in the network. Has the following properties:

(1) The unique identifier of the mail.
(2) Outgoing mailboxes and incoming emails.
(3) Shipping time and receiving time.
(4) Sending IP and receiving IP.
(5) Mail delivery status.

The sending relationship of the mail is in line with the graph of the network. We define nodes to represent users and mails, and edges represent the user's sending behavior to mails.

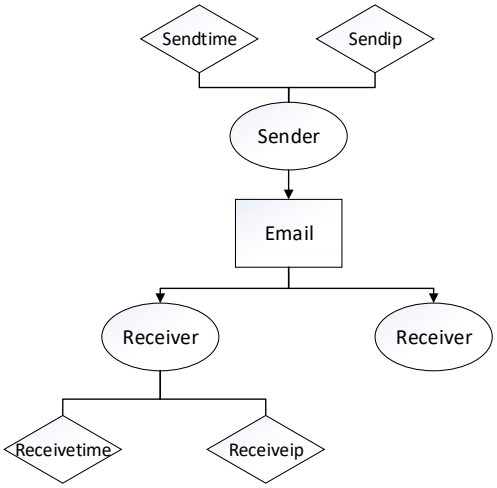

FIGURE 2. mail event model

Algorithm Description

According to the definition requirements, in order to restore event events in a large number of tedious logs, it is logically divided into two steps. First, each message received from the beginning of the transmission to the first server becomes the only one on the server. The ID is identified and the event is restored using each of the above-mentioned IDs obtained after processing.

1. Use a regular expression to extract the initial ID in the following log:
Jul 28 20:58:39 mx postfix/smtpd[10206]: 8504634015B:
Cli-ent=smtpbg335.qq.com [14.17.44.30] AVYxleipJ4h_jrOyBL_D
Get the initial ID set Q= {ids1, ids2..., idsn}.
2. For j=1, 2..., n, get idsi∈Q, do
3. Enter idsi,S:=Search(idsi), where S is the set of all idsi logs, S={p1,p2,...}.
4. Traverse every log in S. The regular email address, the email address, the time of sending, the email, the unique ID of the email, the IP of the email, and the IP of the email in the log.
5. Check the log for "status" and "queued as".
(1) If "status" is included and the mail delivery status is extracted, the event is restored.
(2) If "status" is not included, extract the ID containing the word "queued as" and repeat step 3.
6. Take the next ID from Q and repeat step 3.

Experimental Results

The mail event recovery document format and the import graph database neo4j are visualized as follows:

```
2:
C6E0C340679
messageId:20160808031637664272@mOs.nsbq.com
sender:txok@mOs.nsbq.com
sendIP:49.75.227.22
sendTime:Aug  8 03:16:34
receiver:gycao@sjtu.edu.cn
receiveIp:202.121.179.14
receiveTime:Aug  8 03:16:21
status:sent (250 2.1.5 Delivery OK)

3:
9DF5D340057
messageId:
sender:Ivan@icmea2016.org
sendIP:58.19.56.235
sendTime:Aug  8 03:16:25
receiver:dxcui@sjtu.edu.cn
receiveIp:202.121.179.12
receiveTime:Aug  8 03:16:26
status:sent (250 2.1.5 Delivery OK)
```

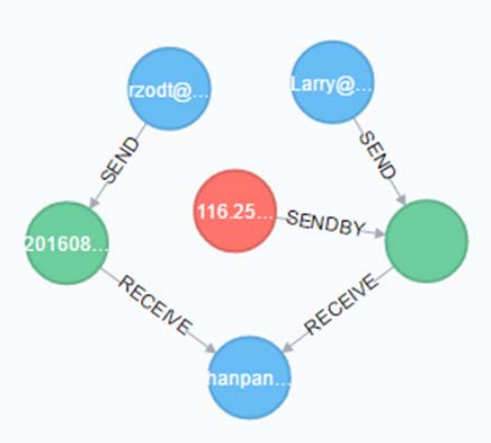

FIGURE 3. mail event in TXT and Neo4j

CONCLUSION

A traditional mail log processing method cannot meet the needs of large-scale enterprises or colleges and universities for mail systems. This paper proposes a data processing program based on the ELK software framework that can deal with a large number of mail logs in real time. By introducing the concept of mail events, it extracts the log information from the mail server and establishes a suitable model to achieve efficient query. Visualizing email dynamic network and user behavior has important use value for detecting spam and user behavior patterns.

REFERENCES

1. Li Jingang, Shi Jinqiao, Qin Zhiguang, Liu Hallwen. User Behavior Pattern Mining for Email Network Event Monitoring[J]. Chinese Journal of Computers, 2014,37(5): 1135-1146.

2. Yang Zhen, Lai Yingxu, Duan Lijuan, Li Yujian, Xu Wei. Research on Collaborative Filtering Mechanism of Mail Networks[J]. Acta Automatica Sinica,2012,38(3):399-411.
3. Hu Tiantian, Dai Hang, Huang Dongxu. CN-M Based Email Network Core Community Mining[J]. Computer Technology and Development. 2014, 24(11):9-12.
4. Ian Robinson et al. Fig. Database [M]. Liu Wei et al. Beijing: People's Posts and Telecommunications Press, 2016.
5. Zhang Jianzhong, Huang Yanfei, Xiong Yongjun. Digital Library Retrieval System Based on ElasticSearch[J]. Computer and Modernization. 2015, 6: 69-73.
6. Bai Jun, Guo Hebin. Research on software integration scheme for real-time search of big logs based on ElasticSearch[J]. Jilin Normal University (Natural Science Edition).2014,1:85-87
7. Chen Bin, Dong Yizhou, Mao Mingrong.Infrastructure learning algorithm-based campus network spam detection model[J]. Journal of Computer Applications, 2017,37(1): 206-216.
8. Gao Kai. Big Data Search and Log Mining and Visualization Scheme [M]. Beijing: Tsinghua University Press, 2016.
9. (U.S.) Ian Robinson et al. Fig. Database [M]. Liu Yi et al. Beijing: People's Posts and Telecommunications Press, 2016.
10. Chen Bin, Dong Yizhou, Mao Mingrong.Infrastructure learning algorithm-based campus network spam detection model[J]. Journal of Computer Applications, 2017,37(1): 206-216.