Log Layering Based on Natural Language Processing

Hanji Shen [ab], Chun Long [a], Wei Wan [a], Jun Li [a], Yakui Qin [a], Yuhao Fu [a], Xiaofan Song [a]
a Computer Network Information Center, Chinese Academy of Sciences, China
b University of Chinese Academy of Sciences, China
shenghanji@cnic.cn, longchun@cnic.cn, wanwei@cnic.cn, jlee@cstnet.cn, qiniyakui@cnic.cn, fuyuhao@cnic.cn, songxiaofan@cnic.cn

Abstract— **With the increasing number and variety of logs, the requirement of storage space is growing rapidly. Meantime, the speed and accuracy of querying in massive logs are becoming increasingly important. Although the well-built distributed storage technique solves the problem of mass storage and fast query, the cost is too high. As logs are created as the method to trace the historical operation, the requirement for query rate is not high. To balance the storage cost and query rate, this paper proposes a real-time log layering storage technique based on natural language processing. According to the characteristics of the log data, this technique is combined with the text language processing technique. It compresses the real-time log data effectively while considering the query efficiency. Firstly, the method extracts the feature of each log that flows in, which will be the type name of the log. Then, the method performs word segmentation on the log and encodes each word to store the key value pairs. Finally, the key value pairs of the log are stored in the memory, and the code of each log is stored in the database. Experiments show that this method can ensure the integrity of the data effectively, decompression time dropped to 40%, compression rate down to 35%.**

Keywords— **Real-time Log Data; Natural Language Processing; Data Compression**

A. Introduction

As the types and numbers of logs continue increasing, the required storage space also grows rapidly, and the speed and accuracy of querying in massive logs are becoming higher and higher[1][2]. There is an urgent need to improve the storage capacity, query speed and accuracy of the log storage system[3]. In the traditional log storage scheme, text files are used for storage. In the traditional field, this scheme can meet most of the requirements, but cannot meet the requirements for storage and fast query of massive data.

The distributed log processing system solves the problem of storage and fast query of massive logs, however, the problems of highly hardware requirements and large storage space cannot be ignored, this problem increases user's storage costs significantly.

B. Background

In recent years, computer systems have produced data in mounting numbers, and the variety of log data has become various. In order to ensure the normal operation of the information system, storage and analysis of various logs is necessary for information security. In traditional log analysis, the log data generated by the log source (system, application) is firstly written to the local log file. The log server periodically detects and collects individual log files and compresses the collected files. When querying the log file, log server selects the required zip file to decompress and then retrieves the extracted file. As shown in the left in Figure 1, each step of this method involves IO operations or network operations, which will increase the operating time. Confronting massive data, the traditional query methods cannot meet query efficiency and storage requirements.

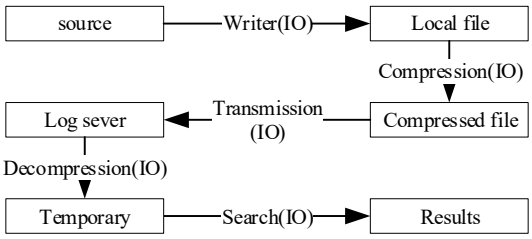

Figure 1. Traditional storage of log data

The current solution is distributed storage[4], such as Elasticsearch[5][6] uses distributed technology to increase storage capacity and query speed, which is at the expense of increasing hardware resources. Firstly, as shown in the left part in Figure 2, in the Elasticsearch storage scheme, the log source and the processed log are

simultaneously stored, which results in data redundancy and the wasting of storage space. Although Elasticsearch uses inverted index[7] which considers both efficiency and space, as shown in the right part in Figure 2, the fields of Doc_1 and Doc_2 are stored[8] in the form of the class matrix, the sequence of each field in Doc_1 cannot be seen. Therefore, the inverted index does not support multifield ordered queries. In conclusion, this storage scheme still has defects.

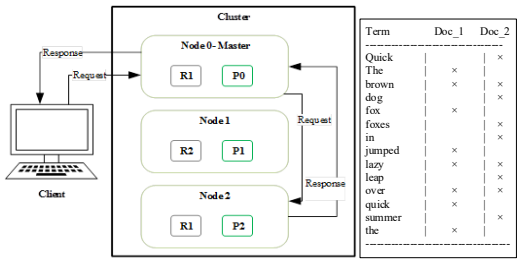

Figure 2. ElasticSearch distributed storage

C. Method

1. Data Storage

1) Data cleaning. The data is filtered according to[9] the simple characteristics of each piece of data (string length, feature characters), and non-target data is filtered out.

2) Data splitting. The data is split into fields based on special characters[10], after that, fields will be stored in an array.

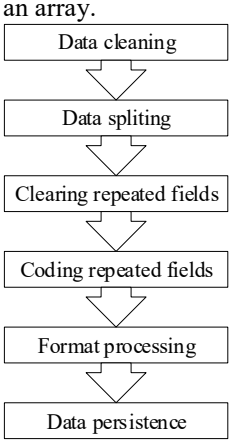

Figure 3. The proposed data storage algorithm steps

3) Clearing repeated fields. In the arithmetic proposed in this paper, the feature and the repeated field will be cleared according to the pre-set model[11] and the position of the field in the array. Creating a word bag with the name of the feature field in the word bag space, if the word bag already exists, skip this step and write the word bag number into the log data.

4) Coding repeated fields. Encode repeated fields and write the code to the word bag, skip if the code exists. Write the field code to the log.

5) Format processing. The non-repeating fields in the log are sorted with the encoding and separated by a unified flag.

6) Data persistence. Persist the formatted log.

2) Data reading

1) Reading data: Storing the data into memory.

2) Translating data: According to the type of data, selecting different word bags, and translating data.

3) Using data: Formatting the translated data and restoring the log format.

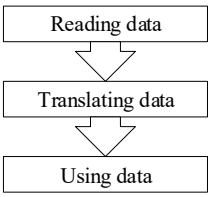

Figure 4. The proposed data storage algorithm procedure

D. Experiments

This section will turn its attention to evaluating the proposed method and comparing its performance with traditional log processing methods.

1. Datasets

To test the log system implementation on a diverse set of benchmarks, some experiments are designed by using different magnitudes of data.

1) Process the log data set using traditional methods.

2) Process the log data set using log layered compression algorithm.

2. Experimental Setting

All the experiments and the other techniques in this section were run on a 4-core Intel i5-6400T machine with 8 GB of memory.

The data set is divided into three parts: 100,000 rows data set, one million dataset, and five million dataset. Then, the proposed method and the traditional method are implemented respectively.

4.3 results

Initially, count the ratio of duplicate fields in the overall log. Then, according to the ratio of the repeated fields, and controlling the compression ratio by controlling the number of repeated fields. Finally, the best point of compression time and compression ratio can be found.

1) Statistical field repetition rate.

TABLE 1. Decompression Rate and Compression Rate

Repeated field	Ratio
@timestamp	41.06
interface	40.04
xethernet0/2	40.08
…	…
sessionEnd	1.26%

2) Compression rate and time under different datasets.

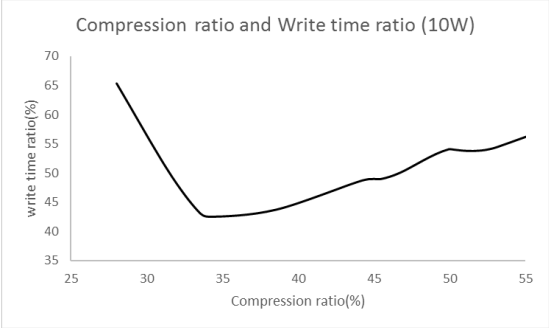

Figure 5. The Compression Ration and Write Time Ratio(10W)

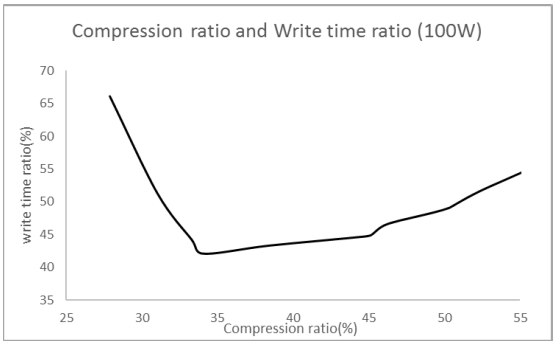

Figure 6. The Compression Ration and Write Time Ratio(100W)

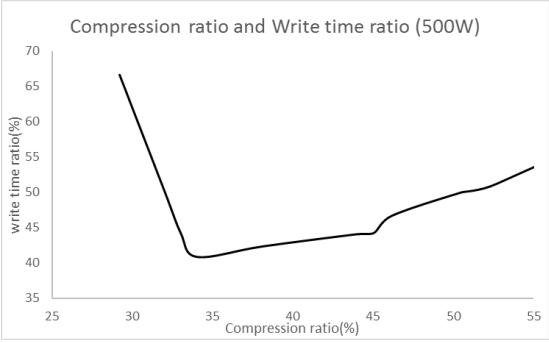

Figure 7. The Compression Ration and Write Time Ratio(500W)

The above graphs are compression ratio and compression time rate of different data sets. As can be seen from the figure, the log data size begins to decrease rapidly with the compression of the repeated fields, which reduced the I/O time. Hence, the overall time consumption is significantly reduced. When the field with a small repetition rate is compressed, the time consumed by the program will starts to increase, at the same time, the data size begins to decrease, which leads to the increase of overall time consumption.

3) Statistical optimal solution

The table below shows the best of each data set.

TABLE 2. Decompression Rate and Compression Rate

Data Set(W)	Decompression Rate	Compression Rate
10	41.06	34.98
100	40.04	34.98
500	40.08	34.67

Table1 combines the experiment's results of the domain name classification. It summarizes the decompression rate and compression ratio of different methods for different algorithms under various data sets. As can be seen from the table, the most advantageous is (40, 35).

E. Conclusion

The log layering based on natural language processing that proposed in this paper uses field splitting and repeated field encoding storage to significantly reduce the storage cost of massive logs and improve the search rate of logs. Experiments show that this method can ensure the integrity of the data effectively. Manwhile, decompression time can drop to 40% and compression rate down to 35%.

REFERENCES

[1] Manogaran G, Lopez D. Disease surveillance system for big climate data processing and dengue transmission[M]//Climate Change and Environmental Concerns: Breakthroughs in Research and Practice. IGI Global, 2018: 427-446.

[2] Zhang Y, Chen M, Mao S, et al. CAP: Community activity prediction based on big data analysis[J]. Ieee Network, 2014, 28(4): 52-57.

[3] He Y, Yu F R, Zhao N, et al. Big data analytics in mobile cellular networks[J]. IEEE access, 2016, 4: 1985-1996.

[4] Rawat A S, Papailiopoulos D S, Dimakis A G, et al. Locality and availability in distributed storage[J]. IEEE Transactions on Information Theory, 2016, 62(8): 4481-4493.

[5]Wang Wei,Wei Le, Liu Wenqing, Shu Hongping. Design and Implementation on Distributed Full-Text Search System Based on ElasticSearch[J]. Electronic Technology 2018,31(08):56-59+65.

[6] Yao Pan, Ma Yupen, Xu Chunxiang. Research and application of log analysis system based on ELK Stack[J]. Computer Engineering and Design. ISSN, 2018,39(07):2090-2095.
[7] Wang B, Song W, Lou W, et al. Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee[C]//Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE, 2015: 2092-2100.
[8] Catena M, Macdonald C, Ounis I. On inverted index compression for search engine efficiency[C]//European Conference on Information Retrieval. Springer, Cham, 2014: 359-371.
[9] Conneau A, Schwenk H, Barrault L, et al. Very deep convolutional networks for natural language processing[J]. arXiv preprint, 2016.
[10] Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A. Recent advances and emerging challenges of feature selection in the context of big data[J]. Knowledge-Based Systems, 2015, 86: 33-45
[11] Christ M, Kempa-Liehr A W, Feindt M. Distributed and parallel time series feature extraction for industrial big data applications[J]. arXiv preprint arXiv:1610.07717, 2016.

Hanji Shen is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. He received M.S. degree in Engineering in the field of Computer Technology from University of Chinese Academy of Sciences, and he is currently studying for a Ph.D. in University of Chinese Academy of Sciences, majoring in Computer Software and Theory. He has been an IEEE member since 2014. His research interest is Cyber Security.

Chun Long is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. He received Ph.D. degree in Computer Software and Theory from University of Chinese Academy of Sciences. He developed Compound Attack Prediction Method based on the Attack Graph and Multi-source Security Event Fusion method based on EA-DS Evidence Theory. His research interests include Information Security, Cyber Security, Cyber Risks and Web vulnerabilities.

Wei Wan is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. He received Ph.D. degree in Computer Software and Theory from University of Chinese Academy of Sciences. He has presided over and participated in many national projects. Moreover, he developed Investigation of state division in botnet detection model, Botnet detecting method based on activity similarity and so on. His research interests include Information Security, Cyber Security, Cyber Risks and Web Vulnerabilities.

Jun Li is currently a researcher, deputy chief engineer and doctoral supervisor in Computer Network Information Center, Chinese Academy of Sciences, China. He received Ph.D. degree in Computer System Structure from University of Chinese Academy of Sciences. He is one of the earliest experts engaged in computer network technology research in China. He has been engaged in scientific research and engineering practice in the field of computer network for a long time. His research interests include Network Architecture and Cyber Security.

Yakui Qin is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. He received B.S. degree in Data Analysis from Fuyang Teachers College. His research interests include Data Security, Network Security Guarantee Technology and Cyber Space Security.

Yuhao Fu is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. He received B.S. degree in Computer Science and Technology from Chongqing University of Post and Telecommunications. His research interests include Network Security Guarantee Technology and Cyber Security.

Xiaofan Song is currently a full engineer in Computer Network Information Center, Chinese Academy of Sciences, China. She received M.S. degree in Cyber Security from University of Southampton. Her research interests include Information Security, Cyber Security, and Cyber Risks