Log Data Real Time Analysis Using Big Data Analytic Framework with Storm and Hadoop

LV Jia-Ke [1,2,a], LI Yang [1] and WANG Xuan[1,2]

[1] College of computer and information science, Southwest University, Chongqing, China
[2] The Key laboratory of Chongqing digital agriculture, Chongqing, China

Abstract. The log data real-time processing platform which is built using Storm On YARN integrated MapReduce and Storm that use MapReduce to complete large-scale off-line data global knowledge extraction, sudden knowledge extraction of small-scale data in Kafka buffers through Storm, and continuous real-time calculation of streaming data in combination with global knowledge. We tested our technique with the well-known KDD99 CUP data set. The experimentation results prove the system to be effective and efficient.

1 Introduction

Last decade has witnessed the expansion of the scale and complexity of log data. Log data reflect the operation of the user during the operation of the program, or the user's access behaviour generated during the interaction with the computer system [1]. The effort of collecting, storing and analysing a large number of logs is aggraded more when the logs are diverse, heterogeneous and dynamic. Due to the size of datasets, conventional database solutions become deficient in processing large amount of data, however, cloud systems can provide achieve scalability and elasticity [2].

Data mining, machine learning and other technologies are widely used in log analysis. Many processing ideas and algorithms have been proposed [3,4].The batch processing technology of the MapReduce parallel computing framework borrowing from Google's distributed computing thought is widely used to extract knowledge from large data volume log data, but it is generally called off-line processing because of the long processing time [5]. Real-time performance and high throughput parallel computing for large data volumes have become the basic requirements for log data processing. Streaming computing process can complete the real-time processing of the log stream data, and can complete the knowledge extraction for a small-scale data set within a certain period of time, however, since the data size limit the applicable algorithms and the degree of reliability of the results, the knowledge extracted and dependent on real-time calculations needs to be combined with off-line batch processing techniques for large-scale off-line data analysis results. Currently, integration of log data off-line calculations and real-time calculations provides some feasible solutions. For example, the designer of Storm, the famous real-time stream processing computing system, which is based on the deficiencies of a single computing architecture, proposed the Lambda architecture theory. Kappa, Summing Bird, Lambdoop and other big data processing systems are all built on the basis of lambda [6-8].For the knowledge extraction, integration of off-line data and real-time streaming data, the distributed system infrastructure Hadoop, the real-time processing platform of log data is constructed by integrating the two different computing frameworks of MapReduce and Storm from the resource scheduling level through "Storm On YARN". Flume and HBase complete the distributed collection and storage of log data, and use the MapReduce with high throughput rate to complete the global knowledge extraction of large-scale off-line data. Through Storm, the sudden knowledge extraction of small-scale data in Kafka buffer and the integration of knowledge flow Real-time continuous calculation of data improves accuracy while ensuring real-time performance.

This paper is organized as follows. Following the introduction, Section 2 gives a detailed description of the proposed framework. Section 3 describes how to possess abnormal analysis based on the shared knowledge base. Section 4 presents the experimental operation and results. Conclusions are finally made in section 5.

2 Proposed framework

The log data real-time processing framework (Figure 1) consists of three main stages or layers, which is summarized as follows:

- Data service layer: data collection and storage;
- Business logic layer: data analysis;
- Web presentation layer: data visualization.

[a] Corresponding author: 1211027620@qq.com

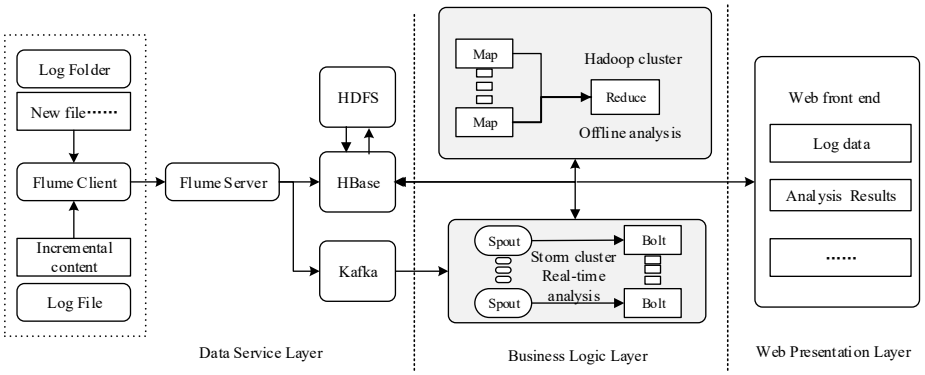

Figure 1 Architecture of the proposed framework for processing log data

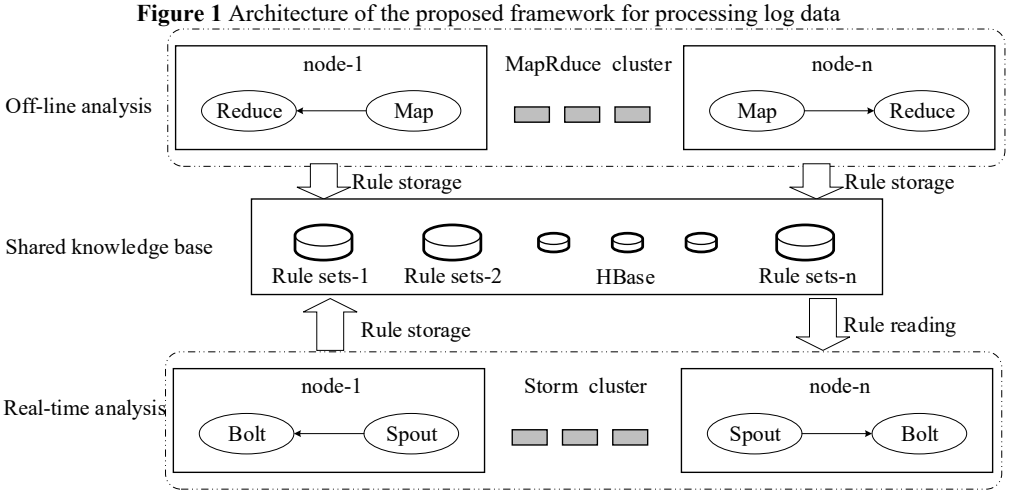

Figure 2 The flow of log data analysis

A more detailed description about each individual layer and their functionalities is presented as below.

2.1 Data service layer

Flume is utilized to achieve real-time acquisition and Kafka, as a distributed and highly available channel that provides data integrity and sequence in real-time log stream data buffers. The flume agent on the application server acts as the collector for the newly added file under the Watch directory and the incremental content of the specified log file, and then pushes the content to the receiving end. The sink agent on the deployment server in a distributed cluster acts as a receiver, which preprocess the received data and distributes it to HBase and Kafka.

2.2 Business logic layer

The business logic layer is divided into two parts: off-line analysis and real-time analysis. Two parts interact through the shared knowledge base stored in HBase.The Storm cluster implements real-time continuous calculation of real-time log stream data in the Kafka buffer based on the extracted knowledge. Data mining technology and MapReduce computing model are utilized to analyse historical log data and extract knowledge. The flow of log data analysis is depicted as Figure 2.

Real-time log data record current state of the system in nearly real time. Knowledge represents the characteristics of systems and devices in a short time. However, The off-line analysis module can discover and extract more comprehensive knowledge from the large-scale historical data. The shared knowledge base is the key in the off-line analysis and the real-time analysis results. The two kinds of knowledge provide different decision weight for the real time continuous computing. All of them are stored in HBase.

In the off-line analysis part, data mining algorithm is parallelized and transplanted to MapReduce to complete the in-depth analysis of massive static historical log data and store it in the off-line knowledge base. The rule library provides decision support for real-time analysis.
In real-time analysis part, Storm's Kafka Spout and HBase Spout, as Judge Bolot's data sources, read real-time log streaming data, blacklist and association rules data. Judge Bolt component parses the log data with blacklist and association rule data to judge whether an abnormal network event occurred. If abnormal network event is detected, Judge Bolt immediately pushes log data to the Warn Bolt component. Warn Bolt component receives network anomalies log and then send alarm. Host IP will be stored in HBase blacklist data table and Warn Bolt update the number of abnormal records of IP address.

2.3. Web presentation layer

The struts framework is utilized for visual page development. Users can conveniently complete log management through the Web terminal and observe the results of off-line and real-time log data analysis.

3 Real time analysis based on shared knowledge base

Real-time analysis is divided into real-time cluster analysis and real-time anomaly analysis. Low-latency continuous processing of each real-time log data can be used to derive the abnormal type of the log within a certain time window. Multiple log data complete cluster analysis.

Real-time anomaly analysis is mainly responsible for the classification and processing of real-time log data. In order to make full use of off-line analysis and real-time clustering extraction knowledge, the anomaly analysis module is mainly designed for three types, which include cluster center-based anomaly analysis, KNN-based anomaly analysis and rule-based anomaly analysis, and The analysis results are multiplied by the sum of the weights to determine whether an abnormality has occurred and an alarm is issued.

After the real-time analysis module is started, the Storm cluster continuously subscribes to data in Kafka and performs corresponding processing and analysis, completes low-latency cluster analysis, and performs real-time processing on abnormal log data to provide timely feedback. . The data flow of real-time analysis is shown in Figure 3.

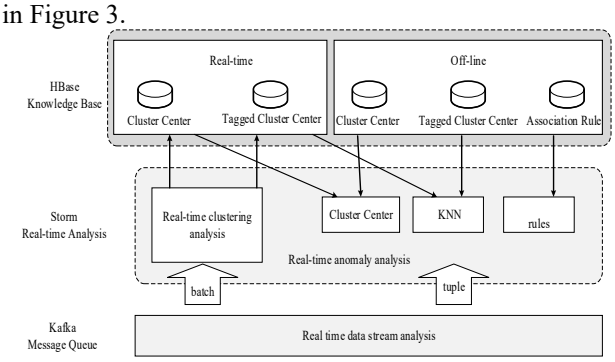

Figure 3 Real-time Analysis Data Flow

3.1 Cluster-based anomaly analyses

This module obtains the log data from the data source and the values of several cluster centers table in the knowledge base, and calculates the distance between the data and each center, thereby obtaining the type of the log, and output the abnormal boolean type Flag bit and log type ID. This module obtains the cluster center-based anomaly analysis by acquiring the cluster center table in the off-line knowledge base and the real-time cluster center table of the real-time knowledge base.

3.2 Anomaly analyses based on KNN classification algorithm

The KNN algorithm is a classic classification algorithm whose main idea is that the similarity between samples with the same classification number is larger. The priority queue PriorityQueue is used to store the 20 nearest KNN nodes. The comparison function of the priority queue is set. The larger the distance is, the larger the priority is. so that the head of the queue stores the node with the farthest distance, and the new node only needs to be compared with the head node.

3.3 Anomaly analyses based on the association rules

The obtained log data is compared with the rules in the HBase association rule table to obtain a matching log type ID. If multiple rule matches for a log data, the rule with the highest confidence is taken as the rule.

4. Experimental setup and results

In this section, we present the experimental setup and the outcome of the proposed framework.

4.1 Experimental setup

Virtual cluster is a simple but fast environment to build this platform. The proposed framework is implemented using the vSphere virtualization platform. In our work, 5 virtual machines (CPU Xeon L5420 2*2.493GHz, memory 4G, hard disk 200GB) with a consistent configuration of operating system Ubuntu 14.04.1 are employed. Storm built on YARN and Flume agent is deployed on the host named Host1 and application server. The development environment is Java 1.8.0_45.The versions of components deployed in the cluster is Hadoop-2.7.2, Storm On Yarn-0.9.6,HBase-1.1.4,Zookeeper-3.4.6,Kafka_2.10-0.10.2.0 and Flume-1.6.0.

The data set comes from the Firewall log in the VAST Challenge 2013 dataset. The log records the work information of a multinational company's internal network firewall in seven days in a row. There are 22 different data items including srclp, destlp, desPort, time, operation, protocol and priority. The data set also records many network exception events. We choose data from 2013-04-10 07:02:35 to 2013-04-15 09:36:41 time period as an experimental dataset for historical log data analysis, which includes total 16.572 million transactions. In view of the KDD99 dataset applied, the application of the analysis function of the platform to the real-time monitoring of network security has achieved fast and accurate judgment on whether the network has been abnormal, and the abnormal type is separated. In the off-line analysis, there are mainly two parts: clustering analysis and association rules analysis. Real-time analysis mainly includes clustering analysis and anomaly analysis.

The experimental dataset is stored in MySQL, and a transaction is extracted sequentially every 10 ms, and then pushed to the Kafka cluster to simulate real-time log stream data. The function test is carried out to verify the accuracy of real-time analysis, and the real-time analysis experiments are performed in the Storm cluster environment and the local stand-alone environment respectively.

4.2 Real time cluster experiments

The size of data processed by real time clustering is much less than that of off-line clustering. In order to ensure the accuracy of clustering, the initial K value of clustering is set to 2, which is divided into normal class and abnormal class. In addition, the time window is set to one minute. At the end of clustering, the correct detection rate, false positive rate and false negative rate of different classes are counted as the evaluation index of cluster analysis. In order to ensure the stability and authenticity of the result, the final result data are taken as the average of many experiments.

Table 1 Results of Real-time Cluster Analysis

Exception type	Correct rate	False rate	Missing rate
Normal	86.19%	13.81%	0.30%
Abnormal	97.79%	2.21%	1.80%

Table 1 shows the clustering result of real-time analysis. It can be clearly seen that clustering accuracy of normal type data reach 86.19% and the accuracy of abnormal type data is even as high as 97.79%. The reason is that the data volume of the exception type is large, and the normal type data is small.Clustering can not guarantee its accuracy when the data record is less.However, if the data of exception type is large and the type is diverse, the rate of recognition of abnormal data may decrease, and the false positive rate will be improved. The results show that the platform real-time analysis can effectively extract the burst knowledge in real-time data, and has high precision, which can provide reliable decision support for real-time anomaly analysis.

4.3 Real-time Anomaly Analysis

As showed in table 2, real-time exception analysis can effectively handle alarms for network anomalies. For all datasets, the correct detection rate of anomaly analysis reaches 90.67%, the correct detection rate for normal log is 82.37%, and the correct detection rate for abnormal log is as high as 92.71%.The more knowledge extracted from the training sets and the higher the precision is.It can improve the ability of anomaly analysis and recognition in real time analysis. However, there is also a certain false positive rate in the analysis. The experimental results show that the real-time Anomaly analysis module can successfully realize the high accuracy real-time analysis of the convective data.

Table2 Real-time Anomaly Analysis

Log type	Classification count	Correct detection number	False Positives	Correct rate
All	311029	282010	29019	90.67%
Normal	61439	50602	10837	82.37%
Abnormal	249590	231408	18182	92.71%

The real-time anomaly analysis shows the statistics of each major class as shown in Table 3. It can be seen from the table that the platform has certain recognition ability for the network anomaly type, but there is also certain false positive rate and false negative report number. It is due to the limited knowledge of training set extraction. For example, association rule analysis can only extract two valuable rules for real-time analysis, and only extract rules from normal classes and DOS classes, resulting in low recognition ability for other exception classes. In addition, there are 17 types of exceptions in the test dataset that do not appear in the training set, resulting in insufficient recognition ability for new types of exceptions. For example, U2R categories with the high false negative rate have new http tunnel, ps, xterm, and sqlattack. There are 189 log records of attack types, accounting for 82.89% of the total number of U2R records. The new exception type of R2L class exceeds 10000 records, which also cause the accuracy of R2L class to be low. The experimental results show that the real-time anomaly analysis has certain precision for the abnormal type identification of log data, which has high recognition ability for DOS and Probe exception classes, which indicates that the accuracy of real-time analysis can be greatly improved after combining the information of the knowledge base.

Table 3 Real-time Anomaly Analysis

IDS Type	Classification count	Correct detection number	Classification count	Correct rate
Dos	130806	130799	81102	99.99%
R2L	60122	9648	5369	16.04%
U2R	10	8	220	80%
Probe	653	555	3611	84.99%

4.4 Anomaly Analysis Based on Different Kinds of Knowledge Base

In order to verify the importance and necessity of off-line knowledge base and real-time knowledge base, 3 different knowledge base methods were used to complete the real-time anomaly analysis. Table 4 shows the exception analysis and comparison for different repositories.

Table 4 Comparison of Different Knowledge Base

Knowledge base	Abnormal detection accuracy rate	False positive rate	specific abnormal type recognition accuracy
Off-line	89.42%	10.58%	73.60%
Real-time	90.67%	9.33%	0.00%
Both	90.67%	9.33%	73.60%

Refer only to the real-time reference repository, although the detection accuracy rate is as high as 90.67%, but not the ability to identify complex exception types. Refer to offline Knowledge Base, the correct rate reaches 89.42%, and can identify complex anomaly types. On the basis of combining the two kinds of knowledge base, the

detection precision the highest and the ability anomaly of complex anomaly type is recognized. The results show that, based on shared knowledge base, real-time anomaly analysis can obtain more powerful data processing ability. As showed in table 4, the performance of performing exceptional data detection is improved when combining the real-time knowledge base with the offline knowledge base. In the massive data processing, the storm-distributed environment provides a great advantage. The shared knowledge base can extend the learning ability of the platform and improve the data processing ability of the platform, also extend the knowledge base by analysing the data mining in off-line analysis and real time analysis.

5 Conclusions

Nowadays, How to obtain useful knowledge from the massive log data at present is a hot topic in the decision-making basis. By constructing real-time processing architecture of log data, this research effectively solves the problems of log data collection, storage and knowledge extraction. The framework incorporates the advantages of Hadoop and Strom, using MapReduce to extract knowledge from historical log data. Based on the storm extraction of the burst knowledge in the small-scale real-time log data and the two kinds of knowledge extracted, the real-time continuous computation of real-time log stream data using storm traditional flow processing can provide a new technical reference for log data acquisition, storage and analysis, and has certain practical and popularizing value.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities (No. XDJK2014B034), Southwest University Education and Teaching Reform Research Project (No.2015JY029) and Chongqing Key Laboratory of Digital Agriculture (China).

References

1. Y. Choi, S. Chang, Y. Kim, H. Lee, W. Son and S. Jin, "Detecting and monitoring game bots based on large-scale user-behavior log data analysis in multiplayer online games", The Journal of Supercomputing, vol.72, no.9, pp.3572-35872016.
2. A. Oliner, A. Ganapathi and W. Xu, "Advances and challenges in log analysis",in,pp.55-61, 2012.
3. R. Nachmias, "Web mining and higher education: Introduction to the special issue", The Internet and Higher Education, vol.14, no.2, pp.65-66, 2011-03-012011.
4. S. Goedertier, J. De Weerdt, D. Martens, J. Vanthienen and B. Baesens, "Process discovery in event logs: An application in the telecom industry", APPL SOFT COMPUT, vol.11, no.2, pp.1697-1710, 2011-03-012011.
5. I. Mavridis and E. Karatza, "Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark", Journal of Systems & Software, no.2016.
6. B. Lakhe, Case study: Implementing lambda architecture, Apress, 2016.
7. O. Boykin, S. Ritchie, I. O'Connell and J. Lin, "Summingbird: A framework for integrating batch and online MapReduce computations", Proceedings of the VLDB Endowment, vol.7, no.13, pp.1441-14512014.
8. V. A. D. Silva, A. J. C. S. Dos, D. F. E. Pignaton, T. J. Lampoltshammer and C. F. Geyer, "Strategies for big data analytics through lambda architectures in volatile environments", IFAC-PapersOnLine, vol.49, no.30, pp.114-1192016.