User Behavior Audit System of Real-time Web Log by Spark

Guonong Li¹, Wenyu Hu¹*, Jinbo Chen¹ and Kanghui Ying¹
¹College of Computer Science and Mathematics, Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian, 350118, China
*Corresponding author’s e-mail: huwenyu@fjut.edu.cn

Abstract. With the rapid development of information technology, single sign-on authentication systems have been adopted by organizations to protect their information management systems and other service platforms. With the single sign-on method, users just log in the authentication systems once to access all the information in systems, which are more convenience, but accompanied by user permissions vulnerabilities and information security risks. Users’ browsing behaviors reflect the users’ usage habits, purposes, and work contents. Risks of information leakage can be reduced by auditing the users’ browsing behaviors. After analyzing the web logs of several service systems, we designed a user behavior audit system based on the Spark computing engine, which takes the real-time web log file as the data set. A possible preprocessing method for real-time web log data stream is proposed to extract users’ web session data. We used the Spark computing engine and the corresponding analysis to mine users’ behavior patterns. The online real-time detection and offline detection of user behavior are realized, and users are divided into three levels: normal, abnormal, and risky. The system administrators or auditors can view the calculation results of each stage online, manage the parameter configuration of the algorithm online, and adjust the user’s access right in real-time.

1. Introduction
With the advances in information technology, it has become important for human beings to easily analyze and extract accurately valuable information from massive data. Traditional database technologies do not meet the needs of the current big data processing. The problem of big data processing has been there until Hadoop and Spark were invented. Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models [1]. The two main technologies of Hadoop are Hadoop distributed file systems (HDFS) and MapReduce. The HDFS is a distributed storage system while the MapReduce is a distributed computing model. The intermediate calculated results of MapReduce will be saved on HDFS when processing data, which increase IO overhead, and reduce the speed of processing data. To solve this problem, University of California, Berkeley AMP Lab developed the Spark framework. Unlike the MapReduce, the Spark framework uses the algorithm of interactive query and generation selection, and memory for data calculation. It also provides memory storage and efficient error recovery [2]. The Spark framework can directly store and call the intermediate results of jobs in memory without reading the data in HDFS, which is efficient and easy to use. The Spark framework can use Yet Another Resource Negotiator (YARN) as its cluster manager to read Hadoop data, including HDFS and HBase, which makes Spark easy to join Hadoop clusters.

Nowadays, the number of web service systems or information management systems has been increasing and the systems’ functionalities are becoming increasingly rich. Usually, an institution runs
and maintains multiple web systems at the same time. However, different systems often adopt different identity authentication methods, which causes inconvenience of cross-domain access and reduction in efficiency. The use of single sign-on authentication systems provides the convenience for people to just login once but access multiple systems. However, many web applications that are accessed via single sign-on authentication systems, have no unified development technology and programming specification, which causes risks in information security and privacy leakage. To solve this problem, it is very important to ensure that users use web resources reasonably and legally, and the user behavior audit provides a feasible solution.

Based on real-time web log as data set, we design a user behavior audit system via Spark. This system consists of four modules: log collection module, log storage module, log analysis module, and log visualization module. The log collection module uses Flume Agent to collect web logs of Tomcat server. The log storage module stores the log files collected by flume in Kafka and HDFS. The main part is the log analysis module, which is mainly used for real-time and offline statistical analysis of data. Real-time statistical analysis uses Spark Streaming to process the log information stored in the Kafka cluster, and takes the processing results as the input data of model training or pattern recognition. Offline data statistical analysis uses Spark SQL to process the offline data of HDFS and store the results in MySQL database as the input data of training model. Log visualization module uses ECharts and Spring Boot framework to visualize the results of analysis, so that the system auditors can clearly find the abnormal or risky users in the web systems.

2. Three main technologies of behavior audit system.

2.1. Flume Agent
Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data [3]. It has a simple and flexible architecture based on streaming data flows. It is robust and fault-tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows online analytic application.

Flume Agent is the simplest deployment unit in Flume. It consists of Source, Channel, and Sink. The Source can listen to one or more network ports, and is used to receive data components. Channel is used to buffer agent and receive data that have not been written out to another agent or storage system. Channel acts like a queue. The Source writes data to them in order and Sink reads data from them following the order. Sink is the export of data, which are the port of pushing events to the next stage. Channel goes on to delete events until the data is fully received by the next stage. In this system, Flume is used to collect real-time web log and distribute log to Kafka for real-time calculation or to HDFS for offline calculation, as shown in Fig. 1.

![Fig. 1. The main process of Flume](image)

2.2. Kafka
Apache Kafka is an open-source distributed event streaming platform for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. It allows one publish and subscribe to streams of records just like a queue. It allows one to store streams of records in a fault-tolerant way and process streams of records as they occur [4]. Therefore, Kafka is a good tool for processing log files in an orderly, reliable, and real-time manner.
2.3. Spark
Spark is a unified analytics engine for large-scale data processing. It provides high-level APIs including Spark Core20, Spark SQL, GraphX, and Spark Streaming for streaming calculation [5]. In this system, two core components of Spark are used to clean log files, Spark Streaming, and spark SQL. Spark Streaming is an extension of spark core APIs, which can process high-throughput, and fault-tolerant real-time stream data. Also, it supports Kafka, Flume, HDFS or other sources as real-time input data, and helps to restore output data in HDFS or database, as shown in Fig. 2.

![Fig. 2. The input and output of Spark Streaming](image)

Spark Streaming provides a high-level abstraction called DStream [6], which can periodically generate Resilient Distributed Dataset (RDD) from live data or by transforming the RDD generated by a parent DStream. The internal working mechanism of DStream is that Spark Streaming receives the real-time input data stream and divides the data into batches and then Spark engine process to generate result stream in batches, as shown in Fig. 3. We will use Spark Streaming to clean the real-time log files, and store the results in MySQL as the training data set of users behavior models.

![Fig. 3. The internal working mechanism of DStream](image)

Furthermore, Spark SQL is a module used by spark to process structured data. It provides two programming abstractions called data frame and dataset, which are used as distributed SQL query engine. Compared with Hive SQL, Spark SQL is built on Spark, while Hive SQL is based on Hadoop, so the execution speed of Spark SQL is ten to hundred times faster than Hive SQL. Moreover, we can use Spark SQL to import data from many different data sources, such as MySQL, HDFS, and JSON files. In this system, we use Spark SQL Engine for offline log files cleaning and output the results to HBase and MySQL for visual display of web front-end.

3. Framework of the system
In this paper, our user behavior system takes the log data of several web application systems as the data source, uses Flume Agent to collect log data, and distributes the data to Kafka message cluster, or persists it to HDFS distributed file system. Spark Streaming computing engine reads the log data in Kafka in real-time for processing and store the results in MySQL. If the data streamed exceed the load of Spark Streaming computing engine, the log data collected by Flume are persisted to HDFS distributed file system for offline processing by Spark SQL Engine. The results of the offline processing will be output to MySQL. When the system administrator turns on the user behavior model training function, we use the log data stored in MySQL to train the model. Further, when the system administrator turns on the user behavior model recognition function, we use the log data stored in MySQL for model recognition. The output data of log cleaning, model training, and user behavior
recognition are stored in MySQL. These data will be read by the front-end web for display and data visualization, which is convenient for system auditors to manage. This system framework is shown in Fig. 4.

Fig. 4. The framework of user behavior system of real-time web log by Spark

4. Functions of the system

4.1. Log Preprocessing
Because the log data recorded by Tomcat has a relatively fixed format, the main work of data preprocessing is deleting irrelevant data, removing duplicate items, standardizing the format of URL, and completing the missing fields caused by the concurrent process. According to the need of user
behavior audit, we select information including the web application system name, URL, IP, username, operation type, operation time, operation status, as shown in Fig.5.

![Log preprocessing diagram](image)

Fig. 5. Log preprocessing

We use session identification to calculate the time accurately that the user viewed a page. A session is a set of web pages that users visit in a period. Through the pages that users continuously request to visit, we can obtain customers’ browsing behavior or interest in the website.

Users’ access request is intermittent. If a user’s request interval in the same page is more than 200 seconds, the probability of the user sending the request again becomes very small. Therefore, we can give a time-out value according to the situation. If the time difference between the user visiting the page exceeds this value, we can evaluate that the user has started a new session. If the users’ request interval on a page exceeds this value, we can evaluate that the user has started a new session.

4.2. User Behavior Audit

User behaviors include multiple operations, which can be represented by behavior sequence. The behavior of a single user is regular, accidental, and repetitive.

Our system used association rules analysis to extract the regularity of user behavior from the historical behavior audit data by considering the contingency and multiple repetitions of user behavior. We mine frequent itemset from processed log data, then use the Apriori algorithm to select association rules which meet the minimum support, confidence and lift from the frequent itemset. The support is the probability that $X$ and $Y$ appear at the same transactions [7], as showed in Eq. (1). The confidence is the probability of $Y$ when $X$ appears, as showed in Eq. (2). The lift is the correlation between $X$ and $Y$ in association rules [8]. The $lift > 1$ means positive correlation. The $lift < 1$ means negative correlation. The $lift = 1$ means no correlation, as showed in Eq. (3).

$$ support(X \rightarrow Y) = P(X \cup Y) = \frac{|X \cup Y|}{|D|} $$

$$ confidence(X \rightarrow Y) = P(Y \mid X) = \frac{P(X \cup Y)}{P(X)} = \frac{|X \cup Y|}{|X|} $$

$$ lift(X \rightarrow Y) = \frac{P(Y \mid X)}{P(Y)} = \frac{|X \cup Y|}{|X||Y|} $$

The normal behavior pattern was obtained from normal historical behavior data through sequential pattern mining. The current behavior data was compared with the normal behavior pattern, and the correlation was calculated to quantify the anomaly detection results. The calculated correlation was divided into three levels of normal, abnormal, and risky, and the results were saved to MySQL for web front-end display.
4.3. Menu of the System

This user behavior audit system has five main menus, which are Application, Setting, Function, Output, and User, as shown in Fig. 6.

- Application can be used to manage web application systems that are audited by a user behavior audit system. Administrators can add a new website to the system, delete audit-free website from the system or modify existing websites information.
- Setting can be used to manage the configuration information of various technologies including Flume Setting, Kafka Setting, HDFS Setting, Spark Streaming Setting, and Spark SQL Setting.
- Function is used to turn the status of model training or behavior recognition.
- Output can be used to show and manage the output of log files cleaning, model training, or behaviors recognition.
- User can be used to view the users’ risk level and manage users’ permissions.

![Fig. 6. The menu of user behavior system of real-time web log by Spark](image)

5. Conclusion

Information explosion brings convenience and efficiency, but causes information security risks. Information management systems and online service systems in organizations contain users’ personal information, business secrets, and research results. If there are business logic vulnerabilities in single sign-on authentication systems, it will cause enormous losses to the organization. Auditing the users’ abnormal behaviors, and adjusting the users’ access rights in time can effectively avoid information leaks.

Based on our review of big data distributed storage technologies, big data distributed computing technologies, and real-time stream data processing technologies, we understand the current situation of web log mining, propose a feasible real-time web log data stream preprocessing method, and design a user behavior audit system supporting real-time web log file processing. The system includes a web log preprocessing module, mining and analysis module, and online identification module. The technologies used include flume, Hadoop, Kafka, spark, and so on.

After completing the program, we tested the function of the system and drew the following conclusions: the preprocessing results of the real-time log file meet the expected requirements, and the user behavior model obtained through data training can distinguish the types of users better. However, the detection ability of low-frequency and small-scale risky operation of users are insufficient. In
follow-up studies, we hope to use fine-grained models to subdivide the users’ behavior to get more scientific and reasonable behavior audit models.

Acknowledgments
The authors gratefully acknowledge the financial support provided by Sub-project of National Key R&D Program(2018YFC1201103). Wen Yu Hu is the corresponding author.

References
[1] ZAGAN, E. and M. DANUBIANU, HADOOP: A Comparative Study between Single-Node and Multi-Node Cluster. International Journal of Advanced Computer Science and Applications (IJACSA), 2021. 12(2).
[2] Guoli, C., et al., Efficient Performance Prediction for Apache Spark. Journal of Parallel and Distributed Computing, 2021. 149.
[3] Birjali, M., A. Beni-Hssane and M. Erritali, Analyzing Social Media through Big Data using InfoSphere BigInsights and Apache Flume. Procedia Computer Science, 2017. 113.
[4] Bunrong, L., et al., Improvement of Kafka Streaming Using Partition and Multi-Threading in Big Data Environment. Sensors (Basel, Switzerland), 2019. 19(1).
[5] Yaoxue, Z., et al., A Survey on Emerging Computing Paradigms for Big Data. Chinese Journal of Electronics, 2017. 26(01): p. 1-12.
[6] Yang, C., et al., The implementation of data storage and analytics platform for big data lake of electricity usage with spark. The Journal of Supercomputing, 2020(prepublish).
[7] Yang, Z., Research on the application of university teaching management evaluation system based on Apriori algorithm. Journal of Physics: Conference Series, 2021. 1883(1).
[8] Hongqin, W., Correlation analysis of College students’ achievement based on improved Apriori algorithm. Journal of Physics: Conference Series, 2021. 1848(1).