IOP Conf. Series: Materials Science and Engineering **758** (2020) 012078 doi:10.1088/1757-899X/758/1/012078

Research of elevator real time monitoring method based on spark

Huang Jianpeng[1], Jiang Xiyang[2*], Tong Yifei[1] and Gu Feng[3]
[1] Special Equipment Safety Supervision Inspection Institute of Jiangsu Province , Nanjing, China
[2]School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, China
[3]Nantong Vocational University, Nantong226007, P. R, China
E-mail: 2176609317@qq.com
*Corresponding author

Abstract. With the development of economy, elevator has become one of the essential means of transportation in people's life. However, due to the complexity of the elevator internal system, the sensors collect a large number of elevator operation data so that the reliability and real-time of data processing is very important for the safety monitoring of elevator operation. To solve this problem, this paper proposes an elevator real-time monitoring method based on spark. By using the stream data processing method provided in spark streaming, a large number of real-time data are divided into time windows, the data is divided into small batches of data below the second level and submitted to spark engine for analysis and processing, which greatly improves the speed and throughput of data processing.

1. Introduction

Elevator plays an increasingly important role in people's daily life. However, due to the large number of elevators, the maintenance quality of maintenance units and elevator manufacturers is uneven, the safety of elevators will be difficult to guarantee if the maintenance of elevators is supervised in the traditional way. In addition, due to the high frequency use of elevators, the number of old elevators is increasing, and there is vicious competition in the current elevator maintenance market, resulting in frequent accidents. Therefore, it is very important to carry out real-time and effective safety monitoring of elevator operation. At the same time, due to the elevator real-time flow data has the characteristics of high-speed production and large date volume, it is necessary to filter the invalid data in the massive real-time flow data when processing these elevator data, and detect the abnormal state in the elevator operation process in time, so as to facilitate the subsequent fault diagnosis and maintenance.

In recent years, many scholars have studied the processing technology of condition monitoring data in many aspects. Zhuang regards monitoring data as flow data, and proposes a real-time processing framework based on flow [1]. Zhang proposed a k-nearest neighbor algorithm based on MapReduce, which runs in a distributed environment, thus greatly improving the efficiency of the algorithm [2]. Papadimitriou designed a distributed clustering framework, which can be directly applied to large-scale data [3]. Zheng proposed a big data-based urban air quality inference system, which can infer urban air quality in real time by analyzing historical air quality data, urban traffic flow data, and personnel mobility data [4]. Kim built data mining algorithm in Hadoop, a distributed parallel processing framework, and used MapReduce parallel collaborative filtering algorithm to process massive twitter big data [5]. Hadoop is a parallel batch processing technology based on distributed file

system, which is suitable for large-scale calculation of massive archived data. However, due to the large amount of disk operations required in the calculation process, which affects the calculation efficiency, it is usually used for offline analysis. In order to solve the problem of low performance of parallel computing caused by disk I / O, spark, a memory based parallel computing framework, has been widely used [6]. The variability and complex data relationship of elevator operation monitoring data are significantly different from the characteristics of Internet industry and manufacturing industry. Under the existing distributed data access framework, according to the characteristics of elevator monitoring data, a massive real-time data access scheme based on memory or efficient disk access needs to be studied, this paper discusses the real-time processing technology of elevator monitoring data. Under the framework of Hadoop technology, Kafka message queue is used to realize the real-time access of massive data, spark streaming is used to complete the analysis of process data, and spark is used to complete the data batch processing [7,8]. It improves the real-time property of data access and processing, meets the needs of real-time processing of large quantities of data in the monitoring system, and explores the application and promotion of real-time analysis technology based on big data in the field of elevator operation monitoring.

2. Analysis of real-time monitoring method of elevator running state

2.1. Overall research route

The overall analysis architecture of this paper is shown in Figure 1, which is mainly composed of five functional modules: data acquisition layer, data transmission layer, data processing layer, data storage layer and data application layer.

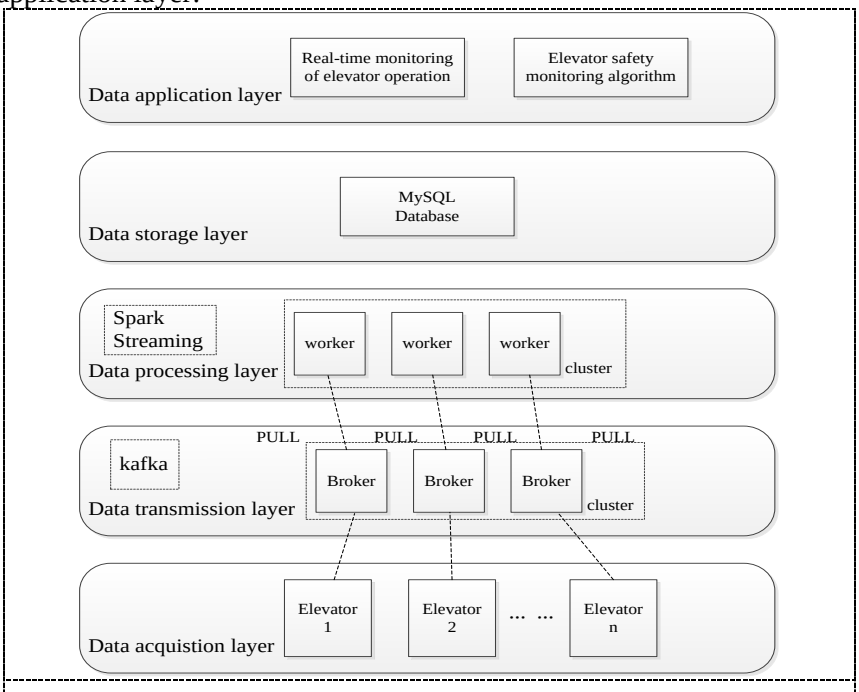

Figure 1. The overall analysis architecture

2.2. Data transmission layer: elevator monitoring data real-time transmission technology

The real-time access of elevator operation status monitoring data includes the collection, transmission and storage of status monitoring data, and provides data input for other real-time processing frameworks (such as spark streaming processing framework). Traditional elevator condition monitoring data transmission mainly uses the method of direct connection with distributed database or real-time database to store, but there are some problems, such as the speed of data collection and data writing does not match, excessive consumption of database space and so on. Kafka is a distributed and high-throughput open source message system supported by Apache foundation. By reducing the

complexity of message queuing system and improving the performance, scalability and throughput of real-time access system, Kafka can solve the delay problem caused by massive state monitoring data written into HDFS in real-time storage, and ensure that all collected state monitoring data can be obtained to real-time processing.

Therefore, this paper uses Kafka as a bridge between data distribution layer and data processing layer, classifies messages in Kafka cluster through topic, and consumers and producers use topic to publish and subscribe to topics of messages, and uses partition to group topics, as shown in Figure 2. When a producer sends a message, it sends the message to a specific partition according to a specific distribution policy. Kafka can ensure the order of messages in the same packet. In the application of detection algorithm to detect elevator fault, it is necessary to detect the state transition process of elevator operation, so the order of messages is very important in the processing of elevator real-time data. In order to ensure the order of the same elevator data when being processed, this paper uses elevator ID as the partition basis, and divides the elevator real-time data by calculating the hash value of elevator ID To ensure that the data of the same elevator is divided into the same partition. In order to ensure fault tolerance, the partitioned data will be saved on multiple nodes in the form of copies, so that the data will not be lost when some nodes go down.

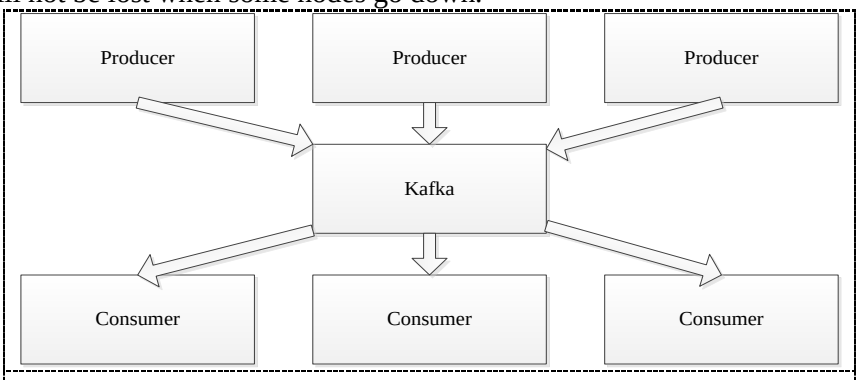

Figure 2. Distributed subscription architecture of Kafka

2.3. Data acquisition layer: building finite state machine for elevator operation

Through the monitoring terminal to collect the real-time state of the elevator running, because of the amount and complexity of the monitoring data, it brings a lot of challenges for the diagnosis of the elevator state. However, the elevator can be described with a certain state at every time in the whole operation cycle. When a specific event (such as elevator being called up and down, door opening at the arrival station, etc.) occurs, the state transition will occur. In order to solve the above problems, aiming at this characteristic of elevator operation, this paper intends to use the finite state machine (FSM) to model the elevator operation state and its state transition process. The finite state machine of elevator running state can be defined as a quintuple:

$$M = (S, \Sigma, \delta, S_0, F) \tag{1}$$

Where S represents the elevator system operation set, that is, the elevator stop state, elevator operation state, elevator arrival state and other elevator operation processes defined in this paper; Σ represents the system input set, and the input of the elevator system is composed of the signals of the elevator control system: up and down call signal, door opening and closing signal, floor button signal, elevator arrival signal and leveling completion signal; δRepresents the state transition function, that is, the elevator system changes the state into the next state through the input signal in one states_0 Indicates the initial state of the elevator； F Indicates the elevator termination status. Combined with the finite state machine model of elevator operation, the process of elevator operation state transfer is shown in Table 1 below:

Table1: Table of elevator operation state transfer

Original state	Input	Target status

Original state	Input	Target status
Elevator stop	Up (down) signal	Elevator operation
	Up（down）signal and on the current floor	Elevator opening
Elevator operation	Arrival	Elevator arrival
Elevator arrival	Stop at the district level layer	Elevator opening
Elevator opening	Execution of the door opening command	Completely opening
Completely opening	Timeout or close button is pressed	Elevator closed
Elevator closed	Execution of the closing command	Completely closed
Completely closed	No other instructions	Elevator stop
	Up (down) signal	Elevator operation

2.4. Data processing layer: Spark streaming data processing technology

The elevator data collection system transmits the real-time data of the elevator running state to the analysis platform through the monitoring platform designed in this paper. Because each piece of data sent by the data distribution system to the message middleware only represents the state of the elevator running at a certain time, and does not directly distinguish the elevator in a specific running process, Therefore, in order to judge the running state of the elevator and carry out fault detection, it is necessary to collect the data for a period of time to determine the collection time interval, which has a great impact on the success rate and efficiency of detection. If the time is too small, it may not be able to effectively determine the running state change process of the elevator, and if the time is too long, it will affect the execution efficiency of the algorithm Therefore, this paper proposes to use the sliding window mechanism in the preprocessing of stream data. First, the micro batch data is divided according to the set sliding window size, and then each data is processed by spark engine in a similar batch processing way. As shown in Fig. 3, it is a schematic diagram of sliding window. The sliding window is 3, and the sliding step is 2. That is to say, the calculation is performed every 3 time units, and the length of each time interval is 2. By setting the window size to control the time interval of single detection, the fault detection algorithm detects the state change of elevator in this period according to the data in this time interval. The sliding step size can avoid the problem that the fault can not be detected due to the time span of elevator operation.

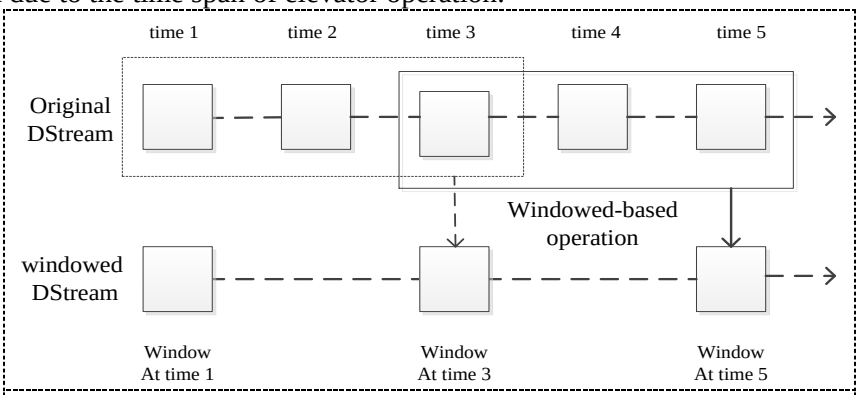

Figure 3. Sliding window mechanism

2.5. Data application layer: Elevator Fault Monitoring Algorithm Based on flow data

Step1. First, the number of spark streaming partition will be divided according to the partition of Kafka, and the micro batch data will be divided using the sliding window mechanism.

Step2. Take the micro batch data as the processing unit, use map to process each stream data, extract the elevator ID in the data as the key value, and filter the useful data as the value.

Step3. Use the groupByKey operator to gather the data according to the key value, that is, the elevator ID. since Kafka can ensure the order of messages in the same partition, it will still ensure the order after aggregation.

Step4. Use mapToPair operator to aggregate the data in the form of < key, value > into the form of < key, list < value > >. At this time, the 'list' corresponding to each 'key' is the data contained in the elevator operation represented by the key.
Step5.Then using map operator, according to the data in each list, analyze whether the running state of the current elevator is abnormal.
Step6.Finally, use foreachRdd and foreachPartition to read the results in each partition and save the results to the database.

3. Conclusion

This paper proposes to use the finite state machine (FSM) to model the elevator operating state, and establish the state change process of the entire elevator operating cycle. At the same time, according to the characteristics of a large number of real-time data generated during the elevator operation, this paper adopts the mixed mode of Kafka, Hadoop, spark streaming and spark to establish a real-time processing framework for elevator state monitoring. Kafka message queue is used to complete real-time data transmission to prevent data loss caused by data access rate mismatch. The sliding window mechanism in spark streaming is used to process the data in batches to realize the real-time processing of the flow data, so as to ensure that all monitoring data and alarm information can be processed in a timely and effective manner, and stored in the distributed file system or real-time database; Finally, through the memory based distributed processing framework spark, combined with mapToPair, reduceByKey and other operators to realize the in-depth analysis of the processed state monitoring data, which provides a new idea and method for the real-time safety monitoring of elevator operation state

Acknowledgments
This work was financially supported by the Fundamental Research Funds for the Central Universities (No. 30919011205).

References
[1] ZHUANG Xue-yin ZHANG Li WENG Xiao-qi 2013 Realtime stream data processing framework for complex equipment condition monitoring *Computer Integrated Manufacturing Systems* (vol 19) pp 2929-2939
[2] Zhang C Li F 2012 Efficient parallel kNN joins for large data in MapReduce *Proceedings of the 15th International Conference on Extending Database Technology ACM* pp 38-49.
[3] Papadimitriou S Sun J 2008 Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining *Eighth IEEE International Conference on IEEE* pp 512-521
[4] Zheng Y Liu F Hsieh H P 2013 U-Air:when urban air quality inference meets big data *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM* pp 1436-1444
[5] Kim Y Shim K 2011 TWITOBI: A Recommendation System for Twitter Using Probabilistic Modeling pp 340-349
[6] Zaharia M Chowdhury M Franklin M J 2010 Spark: cluster computing with working sets *Usenix Conference on Hot Topics in Cloud Computing USENIX Association* pp 10-10
[7] Kafka A 2014 A high-throughput distributed messaging system *URL: kafka. apache. Org* (vol5)
[8] Zaharia M Xin R S Wendell P 2016 Apache spark: a unified engine for big data processing *Communications of the ACM* (vol59) pp 56-65