Distributed Data Streams Processing Based on Flume/Kafka/Spark

WangJun[1, a *], Wang Wenhao[2,b], Chen Renfei[3,c]

[1,2]Henan Xuji metering limited Company, Xuchang Henan Province 461000, China

[3] School of Electronic Engineering, Beijing University of Post & Telecommunication, Beijing 100876, China

[a]4948791@@qq.com, [b]tianyumingyue@163.com,[c]313410646@qq.com

Keywords: Distributed System, Stream Processing, Kafka, Flume, Spark

Abstract. Designed and implemented a distributed data streams processing system based on Flume, Kafka and Spark, fetch and analyze datastreams and mining business intelligence information efficiently, real-timely and reliably, With high scalability and high reliability of Flume, the data of multiple sources can be collected accurately and extended easily. Kafka's characteristics of high throughput, scalability, distribution meet the distribution requirements of massive data. Spark Streaming provides a set of efficient, fault-tolerant and real-time large-scale stream processing frame.Thereby services and strategyof enterprise can be improved.

Introduction

In the information age, the data shows a tendency of explosive growth.Enterprise can produce a lot of logs in many aspects ,for example production and transaction.The scale of the logdata has jumped from GB to TB and even PB such orders of magnitude.Faced with massive logs, conventional data processing system framework has been unable to meet the current needs of the enterprise.At the same time, the real-time demand of enterprise business to data processing is gradually increasing.Throughput and fault tolerance of the traditional stream data processing framework have congenital defects, which does not apply to the high-speed extended business needs of these industries such as the Internet[1].

The change of business requirements has intensified the demand for data processing technology innovation. The current data system does not consider the distributed design, and just some single-components have distributed design, and there is no integrated distributed scheme. INaddition, the reliability of data is not well protected in the process of the datacollection. The forwarding throughput performance of data is inadequate in the context of highly concurrent massive logs. The processing of stream data and the fault tolerance of the system are not well considered. Aiming at a series of problems, a distributed data stream processing system is designed, it has the characteristics of distribution, scalability, large throughput and high reliability. It is able to extract target information accurately and in real time for the enterprise according to the datacontent, to help enterprise to take the relevant marketing measures, to provide users with timely personalized service, to enhance the competitiveness of enterprise [1].

Architecture Design and Implementation

The system is divided into three modules: data collection module, distribution module logs, and data analysis module. Data collection module is responsible for collecting data from different data sources logs and sending to data distribution module. When the data distribution module receives the data, the data stream is grouped according to different topics, and then the corresponding data stream is sent to the data analysis module of the subject. The data analysis module receives a specific data analysis after receiving the data of a specific topic. Seamless connection between the three modules of the system, with the characteristics of distributed, high scalability, high reliability and real-time performance requirements.

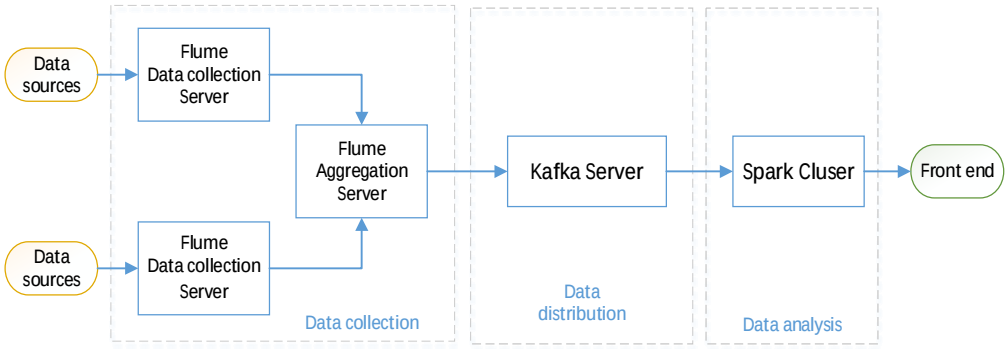

Fig. 1 System architecture

data collection module

Data collection module requires a distributed, reliable, highly available, capable of handling massive data framework, and the need to support multi source collection and centralized storage. In a lot of data collection system, Flume Apache isused [2]. From OG Flume to NG Flume, it has many advanced features. Flume has a simple and flexible architecture based on streaming media data flow. Flume uses transaction based data delivery to ensure the reliability of the transaction. The source and destination of the data are encapsulated into a transaction. Transactions are stored in the channel until the transaction is processed, channel in the transaction can only be removed. This is flume to provide point-to-point reliable mechanism. Flume supports users to build a multi-level flow, with high scalability, supports multi - level extension, the topology of the entire Flume module is based on the specific logical needs by one or more agents. From the multilevel flow, the source of the former agent and the post of a proxy is also the reliability of the data. Thus, the reliability of the Flume is better than the other, such as Scribe, etc... In this paper, the structure of the data collection module is as follows:

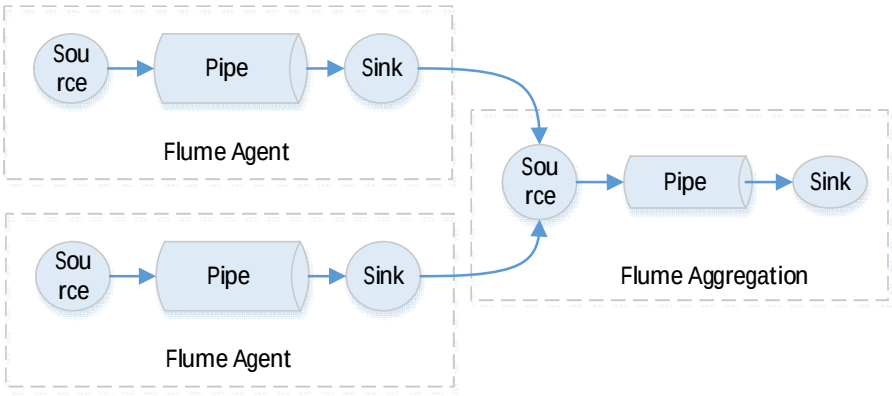

Fig. 2 Data collection module

Flume's "proxy" is the basic unit of data flow, which is composed of three parts: source, channel and sink [3]. Source 1 and source 2 are Exec types in Figure 2, which is used to monitor the data file, and when the file is added at the end of the file, the Flume will be acquired in real time. Channel achieves the function of the message queue, is able to cache memory, but also to achieve the persistence of data. Places 1, 2, 3 source belong to the Avro type, the data transmission between them is based on the Avro event, which can realize the remote transmission of the cross node, through the configuration of address and port to achieve data transmission and reception. 3 is a custom type, as a Kafka message production side, achieve the function about sending the collected data to Kafka.

Flume's "proxy" is the basic unit of data flow, which is composed of three parts: source, channel and sink [4]. Source 1 and source 2 are Exec types in Figure 2, which is used to monitor the data file, and when the file is added at the end of the file, the Flume will be acquired in real time. Channel achieves the function of the message queue, is able to cache memory, but also to achieve the persistence of data. Places 1, 2, 3 source belong to the Avro type, the data transmission between them is based on the Avro event, which can realize the remote transmission of the cross node, through the

configuration of address and port to achieve data transmission and reception. 3 is a custom type, as a Kafka message production side, achieve the function about sending the collected data to Kafka.

data distribution module

Existing message queuing systems, such as ZeroMQ[3], are able to handle real-time or near real-time applications, but the data is not normally written on disk, which may be a problem for Hadoop or Spark applications[4]. Kafka is a publish / subscribe message-based system, support the Hadoop or Spark data parallel loading, for not only offline analysis, but also real-time processing. Kafka achieves the ability to provide message persistence in a time complexity of O (1), even for the TB level data can also guarantee the performance of a constant time complexity. Kafka is a fully distributed system, the proxy server, the message production side, and the message consumer end are all native support for the distributed automatically [5]. The number of the three can be more than one. Multiple proxy servers, message production, message consumption can be run on a large cluster, as a whole, as a whole external service. And through the use of Zookeeper (a distributed application management framework) to solve the problem of data management in distributed applications[6]. Kafka based on Java development, with many other framework to be friendly to convergence, the structure is flexible, good scalability, better than the use of Erlang prepared by the very heavyweight RabbitMQ. In this paper, the architecture of the data distribution module is as follows:

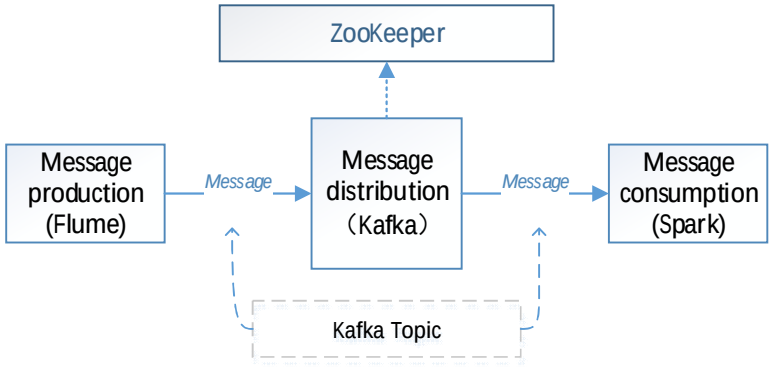

Fig. 3Data dissemination module

data analysis module

Spark is a highly efficient distributed computing system, using its in-memory characteristics, especially good at iterative and interactive data processing [7]. Streaming Spark is a real time computing framework based on Spark, through the rich API provided by Spark, based on the memory of the high-speed execution engine, it can be implemented to extend, high throughput, and has the fault-tolerant real-time data stream processing. It divides the input data into a DStream (discrete data stream), each section is converted into RDD (Spark), and then converted into Streaming Spark to convert Stream to RDD, and the RDD is converted into the intermediate result. The entire flow calculation based on business needs can be superimposed on the results of the middle, or stored to the external device. Fault tolerance is very important for flow calculation. Every RDD is a variable distributed recalculation of the data set, the record inheritance relation deterministic operations, as long as the input data is fault tolerant of, then an arbitrary RDD partition error or are not available are can use the original input data by switching operation and re calculated. This guarantees the fault tolerance of Streaming Spark. In this paper, the data processing flow chart of the data analysis module is as follows:

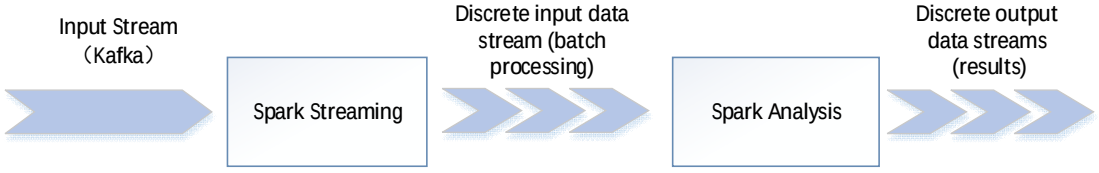

Fig. 4 Log analysis module

System Test

Experimental Environment

Tab. 1 The hardware and software environment of experiment

Node	N1	N2	N3
Hardware	RAM：4G Hard Disk：100G CPU：2 cores	RAM：2G Hard Disk：100G CPU：2 cores	RAM：2G Hard Disk：100G CPU：2 cores
OS	ubuntu-12.04.5-amd64	ubuntu-12.04.5-amd64	ubuntu-12.04.5-amd64
Software	spark-1.3.1-bin-hadoop2.4		
	kafka_2.10-0.8.2.1		
	apache-flume-1.5.2-bin		
Role	master node, process node	Backuo master node, process node	process node

Experimental Data

The experimental data come from Tianchi data laboratory of Alibaba. The data set contains anonymous users' shopping records in the six months. The information of the transaction data is analyzed to get the relationship between order quantity and the user's age, gender. In the test, a script is used to transform data into high concurrent data streams that are sent to the system, and the data stream data is analyzed in real time.

Experimental Conclusion

During the experiment, the data stream data is analyzed in real time, and the following experimental results are obtained.

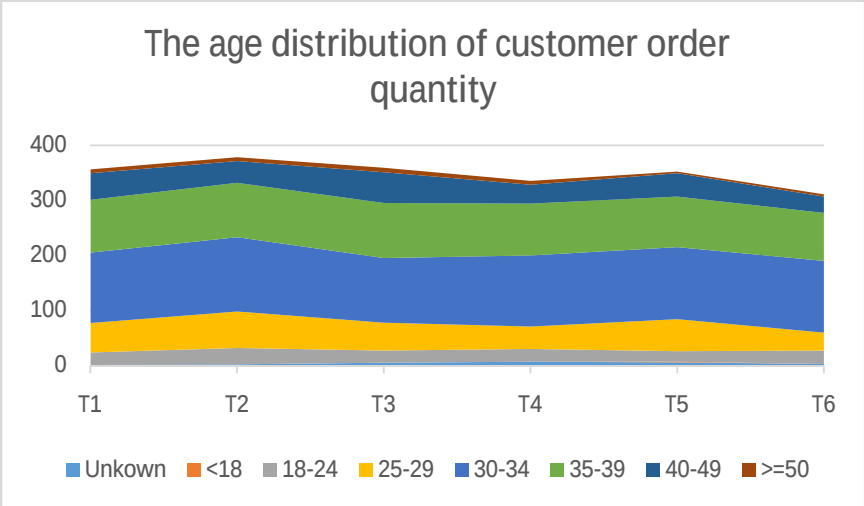

Fig. 5 The age distribution of customer order quantity

According to Fig. 5, the age distribution of order quantity. Customers are mainly distributed in the range of 18 to 49 years old, and the order quantity of 30 to 34 years old users is the largest. And according to the experimental results, we can quickly get the proportion of every age group.

According to Fig. 6, the gender distribution of order quantity. The order quantity of female customer is significantly larger than the order quantity of male customer. And according to the experimental results, we can get the proportion of gender.

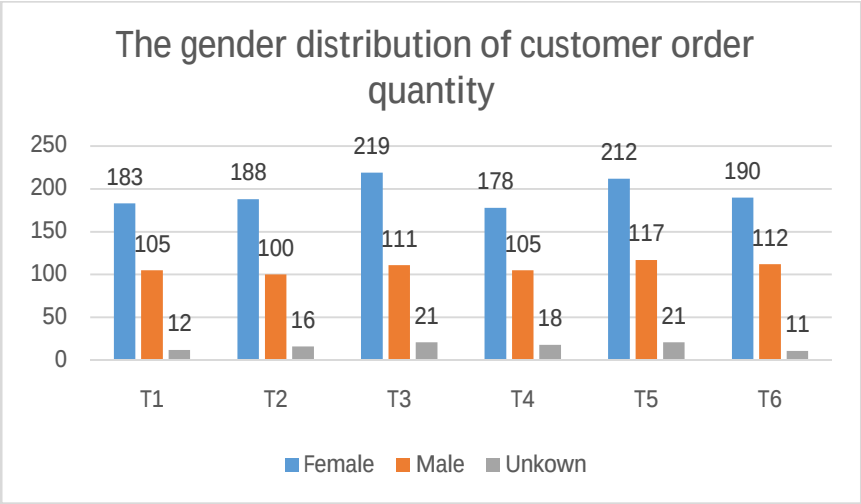

Fig.6 The genderdistribution of customer order quantity

Conclusion

A distributed data stream processing system based on Flume/Kafka/Spark is presented. It can handle large-scale dataefficiently and accurately.With high scalability and high reliability of Flume, the data of multiple data sources can be collected accurately and extended easily. Kafka's characteristics of high throughput, scalability, distribution meet the distribution requirements of massive data. Spark Streaming provides a set of efficient, fault-tolerant and real-time large-scale stream processing frame that can quickly analyze and calculate a large amount of data stream data. The architecture of the distributed data stream processing system is novel, and can realize the overall processing demand of the data stream data collection, distribution and analysis. According to the results of the analysis, the enterprise can adjust the advertising target population and take other relevant measures. The system completes the processing of data stream data efficiently and accurately, extracts the relevant information by analyzing the order data, helps enterprise to adjust the marketing strategy, and improves the performance of enterprise.

References

[1] Jay Kreps. I Heart Logs : Event Data, Stream Processing, and Data Integration.Sebastopol, CA, USA: O'Reilly Media, 2014.

[2] Steve Hoffman. Apache Flume : Distributed Data Collection for Hadoop (2nd Edition). Olton Birmingham, GBR: Packt Publishing Ltd, 2015.

[3] Faruk Akgul.ZeroMQ.Olton, Birmingham, GBR: Packt Publishing, 2013.

[4] Sandeep Karanth. Mastering Hadoop.Olton Birmingham, GBR: Packt Publishing Ltd, 2014.

[5] Nishant Garg. Learning Apache Kafka (2nd Edition).OltonBirmingham, GBR: Packt PublishingLtd, 2015.

[6] Flavio Junqueira, Benjamin Reed. ZooKeeper : Distributed Process Coordination.Sebastopol, CA, USA: O'Reilly Media, 2013.

[7] Krishna Sankar, Holden Karau. Fast Data Processing with Spark (2nd Edition). Olton Birmingham, GBR: Packt Publishing Ltd, 2015.