Research on Design and Development of Data Processing Platform Based on Hadoop

Yaning Yan

Xi'an PeiHua University, Xi'an, 710125, China

Keywords: Data processing platform, Big data, Hadoop

Abstract: With the widespread popularity of computers and networks in people's work and life, the amount of data has increased dramatically. Nowadays, how to collect and store the data effectively is the main problem that needs to be solved. Based on Hadoop technology, this paper gives the system requirement analysis, system architecture design and key modules design of massive data processing platform, and elaborates the key points of the data processing platform development based on Hadoop technology to provide some references for the relevant researchers.

1. Introduction

The storage and processing of large data sets is always a problem faced by programmers in application development, and the massive and dynamic data is a severe test for server's software and hardware. With the rapid development of mobile Internet, the data storage and access response speed will also face new bottlenecks. In many industries and institutions, the fast batch processing of large and medium-sized data sets in large and medium-sized databases and data warehouses has extensive application requirements. How to realize the batch processing fast interactive mass data management information system is facing an increasingly prominent problem. The key problem is data integration project in information technology architecture of the urgent need for a dynamic and scalable storage model, to achieve rapid response mechanism. Large scale data query, analysis, extraction, update and other batch processing of large scale database is facing severe practical problems. The traditional database is complex and the query time is too long. It cannot even be completed when it is faced with a large amount of data query task. When the system faces large concurrent data tasks, the system performance will be dramatically reduced. Traditional database scalability is poor, and additional extensible hardware cannot effectively improve the performance of the system. The traditional data platform based on parallel processing cannot meet the actual requirements of mass data processing, and the cost is high and the maintenance cost is great. Research on massive data and timely and efficient processing technology will effectively improve the application performance of computer system to better provide social basic services and drive the growth of economic benefits. As a new business computing model based on the Internet, Hadoop provides flexible computing power and efficient mass data analysis and processing methods. In this paper, a massive data processing platform based on Hadoop is designed and built, which provides a new method to solve the problem of massive data processing in relational database.

2. Design of data processing platform based on hadoop

2.1 System demand analysis

The system will store the offline data in it. For online data, the system can isolate and store the data in an orderly way, and users can choose whether to allow data loss to improve the performance of the application. Finally, once the user submits an online processing task for a data source, the data will enter the subsequent data flow processing. The processing logic for data flows varies with the application scene. To enable the system to support a variety of data processing logic, the general data

processing operations are abstracted into a separate functional component. Therefore, the user only needs to flexibly assemble the functional components needed and specify the topology relationship between the components. The system supports user custom data processing rules. Users can assign one or more data sources as data to be processed, and can choose the data processing components needed flexibly, and can specify the topology between processing components. In the face of flexible and changeable business logic requirements, this system is an online data flow processing platform, and it is very important to be versatile and easy to use. Because the system supports multiple online tasks running at the same time, it often processes multiple data sources at the same time. The speed of data generated by different data sources is different. Therefore, the stability of the system is reflected in the stable access of the data source and the stable operation of the online task. In the design, the problem of single point fault of the system should be considered specially to ensure the stable service of the system. For streaming data, the system can not only get these data in time, but also need enough computing power to process and store quickly, and feed the results back to the application. Therefore, the computing power of the core calculation part of the system is higher.

2.2 System architecture design

According to the general process of data processing, the system is divided into layers, from top to bottom, the data source layer, the computing layer and the storage layer respectively. The data source layer provides services externally, and is responsible for the access of external data sources. The computing layer will process data based on business processing logic, and the storage layer is responsible for the persistence of processing results. A data source can be an offline data in a database, or a log data generated by an online application. It is convenient for external applications to send data to the system by providing stable services to the outside world. In addition, to alleviate the problem that the speed of data source does not match the speed of system processing, all incoming data sources will be segregated and stored in the message queue within the system according to the theme. It saves the offset of consumption record consumption data, which not only simplifies the calculation method of the system, but also facilitates the recovery of the system. The computing layer is the core of the whole system, providing all the real-time computing services. There are two main functions of the computing layer, which converts the custom rules into an online task, that is, and runs in it. After the submission is run, the data source component will pull the message from the data source layer and send it out, and the message flow will be processed by a series of components. Ensure that each will be handled successfully at least once. Each of them is processed after a series of components, and the tree structure is called a tree. It can track whether each tree and tree can be handled successfully. If there is no successful execution of a tree in the timeout time, it will be marked as execution failure and then will be launched again, each with default message timeout settings. The storage layer is the landing link of the data processing results, and is updated in real time according to the calculation results of the computing layer. In the data storage layer, it is necessary to ensure that the system is persisted in a high performance.

2.3 Key modules design

The data source access module provides two modes of work: synchronous and asynchronous calls. The meaning of synchronous invocation is to send a data to it and wait for confirmation, so that we can continue sending the next data. Asynchronous calls do not need to wait for confirmation, but send data as soon as there is data. Therefore, for applications with high real-time requirement and tolerance for a small number of data loss, asynchronous mode can be chosen, while for applications that cannot tolerate data loss, synchronous mode can be chosen. For example, the data source of reptile's application requires relatively high time to interface calls, and the time occupied by synchronous calls is longer than that of asynchronous calls, which may cause the suspension service of crawler applications. In this case, it is necessary for the rice to be called asynchronously. The core computing module provides all data processing operations, as described in the section. The functions and uses of each data processing component are independent of each other, so it can facilitate the flexible combination of users to complete the required processing tasks. The function of this component is

persistent data, and data has two properties. One is the result of a data flow after a series of processing logic; one is the data that needs to be stored. The amount of data in the former is generally small, and the amount of data in the latter is generally increasing. The growing table can be automatically divided, and each area after the partition is made up of a subset of the rows in the table. For a table, it is made up of a region. There was only one area at the beginning, but as the region began to grow. When it exceeds the threshold of the set size, it will divide the table into two new areas with the same size on the boundary of a row, and the number of regions will also increase.

3. Development of data processing platform based on hadoop

3.1 Development of data source

The data acquisition layer is the bottom of the architecture of the whole mass network data processing platform. Massive network data platform first needs to receive the collected data and complete the distributed storage of massive network data, which requires that these data be distributed to different data storage nodes according to certain rules for storage. At present there are two main types of original data, one is the original message flow data collected, the data size is larger, often an hour a few sizes, so the physical medium will be sent to the cache, the physical medium mass network data processing platform in the data center, and then through the local upload stored to the cloud platform. Another kind of key traffic indicators used for real-time analysis is based on real-time monitoring of the quality of mobile Internet related businesses based on the business needs of operators. To protect the integrity and reliability of the data, there is a need for an independent module to undertake the forwarding of massive data. At present, traffic monitoring equipment includes industrial control computer to collect data of mobile Internet traffic critical data, generate data flow record list and binary message list separately, and transmit to mass network data processing platform. At the same time, with the change of traffic data, the changes of external configuration environment such as storage and data nodes may lead to the change of data forwarding rules, so the data forwarding layer will be compatible with the expansion of the system. In addition, the data forwarding layer also provides control management interfaces for the entire distribution process. From the perspective of function implementation, the task of data forwarding layer is to converge distributed and forward functional modules, which use connection to achieve communication and distribution of control information and raw data.

3.2 Development of data calculation

The preprocessing component of the framework is designed as a parallel structure, and the module that needs to load is specified through the configuration file. In the initialization stage, the framework creates the object of the specific implementation class specified by preprocessing through reflection technology. In preprocessing components, frameworks belong to specific topic roles. In the framework, an abstract observer role queue is maintained, and each specific implementation class implements the interface. When the framework is initialized, the object of the module defined by the configuration file is created and added to the queue. When the framework gets data from the data preparation component, all observers will be notified. When users submit files to the file system through the client, they need to apply the data block information to the node, which includes every node information to write. When the application of the client is received, the write allocation node will be used for the data block, and the rack sensing strategy is adopted in the node selection algorithm. It is configured in the default configuration of the file system, and the default is that all machines are on the same rack. A distributed computing framework consists mainly of functions and functions, and is used as input and output with key value pairs. Key value pairs can be data types such as strings, integers, bytes, or complex user defined data structures. Functions and functions are defined by the user. First, the function receives a set of key value pairs, and then operates on the input data according to the user's custom function, and generates a set of intermediate key value pairs. The framework then converge the intermediate key values generated by all functions to the same value,

and then pass it to the function. The function receives the value of a middle key pair and the set of corresponding values. According to the user defined function, the corresponding values are processed, and the key value pairs after processing are processed and processed once.

3.3 Development of data storage

In the data storage area, we first use the preprocessing module to process the original data preliminarily, get the primary data with applied value, and provide convenience for the application layer to reprocess data. The data storage area is composed and composed. Modules read raw data in different ways, and process them initially. Then, the intermediate results are stored on the data upload interface. The module consists of data and data analysis into two sub modules, including data analysis module is built on the framework, provide the module mounting the serial, the main function of massive network data processing analysis work to read the data on the execution of a job. The data import framework adopted in this paper is used in the storage of massive network data. The framework first receives the input data and then writes the original data into the storage in a distributed way through user defined data preparation. At the same time, the corresponding index information and statistical data are generated for the data. It needs to be explained that this section does not modify any processing of the data to ensure that the data is stored in a complete way. The way of data input is not directly defined in the framework. Instead, it defines an abstract data input interface like that in the framework, which determines which concrete data input module is loaded by the configuration file, and creates the object of the specified module by reflection. Ad locum. The framework belongs to the specific observer role in the observer pattern, while the loaded data input module is a specific theme role. When a record is received or read, the data input module notifies the framework and passes the record to the framework. If the data of different formats and different contents are passed to the framework by writing different implementations of the class. In addition, because the framework does not provide data writing function, in the implementation class, we need to decide whether to write raw input data and provide specific methods for writing. The framework calls the method of writing after the data preparation component is completed.

4. Conclusions

The contradiction between the technology demand of mass data fast processing and the lag of current technology status will become increasingly fierce, which has become a hot topic of wide concern in the world. Design and build a platform for the realization of massive data based on Hadoop. This system is a practical application of Hadoop technology for mass data processing. It can provide reference and reference for the system development oriented to mass data processing.

Acknowledgements

The paper is the result of Special Plan Project of Science Research of Educational Department of Shaanxi Province (Grant No. 17JK1059).

References

[1] Huang Suping, Ge Meng. Research on the Application of Hadoop Platform in the Big Data Processing [J]. Modern Computer, 2013(10): 12-15.
[2] Zhu Haodong, Feng Jiamei, Zhang Zhifeng. Study on big data processing platform based on Hadoop [J]. Journal of Central China Normal University (Natural Sciences), 2017, 51(5): 585-590.
[3] Song Jun, Zhu Lin. Mass Data Processing Platform Design and Implementation Based on Cloud Computing [J]. Telecommunication Engineering, 2012, 52(4): 566-570.
[4] Wang Xi, Xie Ping, Wang Ying. Performance Analysis of Hadoop-based Synchrophasor Datasets Processing Platform [J]. Electric Power Information and Communication Technology, 2014, 12(9): 1-5.