Heterogeneous Data Resource Integration Based on Big Data Platform

Qing Nie[1], Wenqiang Zhang[2], Tingting Zhang[3]

[1]Taiyuan University of Technology, Beijing 100005, China;
[2]School of Sichuan University, Beijing 100005, China;
[3]North China Electric Power University, Yantai 264001, China.

Keywords： Multi-Source Heterogeneous Data, Hadoop, Distributed Storage, Distributed Storage, Mapreduce, Spark.

Abstract： The heterogeneous resource integration system based on big data platform is designed to solve the problems of data resource storage and resource application in the media field for many years. By introducing the related technologies of the big data platform, this paper presents the overall design scheme based on Hadoop from the technical architecture and data architecture. Compared with the traditional data warehouse, the system has greatly improved its performance in terms of data storage capacity, data computing capacity and data query capability, which embodies the efficiency of big data platform in TB/PB data processing.

1. Introduction

With the development of enterprises, more and more structured data, semi-structured data and unstructured data are accumulated inside and outside, and these data exist in many kinds of data sources. Solving the storage and unified management and utilization of multi-source heterogeneous data has always been the key work of enterprises. In the past, federated database model[1], data warehouse model[2] and middleware-based data integration model[3] can be used. The advantage of federated database management system in document[1] is that it integrates various database systems in different degrees by mapping and provides control and collaboration for all systems as a whole. The disadvantage is that the system is not easy to expand and the data sources are not too many. The characteristics of data warehouse mode in document[2] determine this kind of data warehouse mode.The method is suitable for the situation of stable data sources and infrequent changes of data, and is not satisfied with the system that data changes frequently and needs to be queried. In document[3],the author mainly puts forward the principle and data flow process based on XML middleware, but for the current large number of multi-source data storage, the scale of this mode exists. And available limitations. On the basis of studying the integration of traditional heterogeneous data warehouse and the related technology of big data platform, this paper proposes a heterogeneous data resource integration scheme based on big data platform.

Chapter 2 introduces the related technologies of big data platform, and understands the related technologies needed to build heterogeneous data resources integration in this paper. Chapter 3 designs a storage scheme of big data resources which is suitable for multi-source data sources and based on big data platform technology, it introduces the technical framework of the whole big data platform as well as the relevant data flow architecture of related technologies in big data platform. Compared with resource storage, computing power and query ability of traditional data warehouse, it reflects the efficiency of the system. Chapter 4 summarizes the capability of heterogeneous data resource integration based on big data platform for massive data storage and application.

2. Key Technologies of Big Data Platform

Big data has four V characteristics in the industry. Volume, Variety, Value and Velocity are the main characteristics of big data. In order to make full use of big data, Hadoop technology is used in the design of big data platform architecture in this paper.

2.1. Data Storage

The system involves a variety of data types. According to the data structure, it can be divided into structured data, semi-structured data and unstructured data. According to the use of data, it can be divided into Atlas data, offline data and real-time data. The Hadoop framework provides a good solution for storing the above data. By using HDFS, the system data is stored on a big data platform, which provides a good solution for multi-source heterogeneous data integration.

HDFS[4](Distributed File System) is a file system that can reliably store big data on a large cluster. It is the storage cornerstone of distributed computing. Hive and HBase are big data storage based on HDFS. Hive is a distributed relational data warehouse. There are a lot of constructed atlas data that need to be stored in mongdb graph database and processed structured data that need to be stored in mysql. The system uses Sqoop technology to associate traditional database with Hive. HBase is a distributed non-relational database based on column storage. There are a lot of historical records, business content and other historical data in the system. This kind of data needs to be processed offline and stored uniformly in HBase. There are also data streams that need real-time processing on big data platforms. These real-time data streams, which are stored in distributed memory cache Kafka[5].The working principle of Kafka is introduced in document[5]. Kafka is a high throughput distributed message publishing and subscribing system, which carries out data according to topic. Classify and persist the processed data to the local database.

2.2. Data Computing

In big data platforms, the data processing methods are selected according to different data types. At present, there are many kinds of big data computing models and corresponding big data computing systems and tools for the diversity of big data processing, as shown in the following Table 1:

Table 1: Big Data Computing Model and Typical System.

Big Data Computing Model	Typical System
Large Data Query Analysis and Computation	Hbase、Hive、Impala、Hana
Batch Computation	Hadoop、MapReduce、Spark
Flow Computation	Flume、Storm、Spark Streaming
Iterative Computation	MapReduce、spark
Graph Computation	Giraph、PowerGraph、GraphX
Memory Computation	Hana、Spark

2.2.1. Real-Time Data Processing

For data that need real-time processing, data are usually collected and distributed through Flume and Kafka, and real-time data is processed through stream computing by storm[6]. The computing and processing framework of storm system for real-time data is introduced in document[6]. Storm is a distributed and fault-tolerant real-time stream processing system, which can process and operate data at a single node level of millions. Storm system ensures that every message is processed at least once, and when the task fails, it retrieves the message from the source for calculation. This design ensures that the message can be processed quickly, and ensures the timeliness and real-time of real-time data processing.

2.2.2. Offline Data Processing

For a large number of offline data in the system, real-time data calculation is not required. The offline data can be processed by asynchronous task mode. In Hadoop technology, such data is generally processed by MapReduce[7] or SparkSql[8]. MapReduce technology is described in document[7]. MapReduce consists of two stages: Map and Reduce. Map () function takes key/value pair as input and generates a series of key/value pairs as intermediate output to write to local disk. The same key value data is unified to reduce () function to process. MapReduce framework has efficient data processing capabilities at TB and PB levels, and meets the requirement of massive historical data computation in offline computing scenarios. However, when MapReduce cluster runs, it generates a large amount of intermediate data, and the transmission overhead time accounts for 70% of the total time. To solve the problem of insufficient MapReduce framework, it proposed SparkSql Memory Computing Framework in document[8]. In document[8], Spark uses lighter threads as execution period, which greatly reduces the cost of job startup and communication but improves the response speed of scheduling. SparkSql uses RDDs to cache data into Spark cluster memory through Row objects, which reduces a large number of Map tasks. In this paper, we use MapReduce framework and Spark memory computing framework to process offline data in the design of big data platform.

2.2.3. Result Storage

In big data platforms, data results are generally stored in the data service layer. The main function of the data service layer is to make data application more convenient. In this paper, Redis is used as the result storage database. Redis[9] is a key-value database product based on memory, which supports storage of various data types and provides rich operations for each data. Redis as an in-memory database has the following characteristics:

(1) Data is stored in memory.

(2) New data can be written to the database asynchronously on a regular basis without affecting the service.

(3) Implement master-slave replication, improve load capacity and prevent single point failure effectively.

Redis has excellent read/write performance, but due to memory storage problems, it is necessary to build a cluster for horizontal expansion. Redis cluster can be used as a highly available distributed cache. The cluster uses a non centralized architecture. Data is allocated and stored among nodes through pre-bucket strategy. The cluster uses a master-slave approach to ensure the integrity and availability of the cluster.

3. Overall Design Of Big Data Platform

3.1. Technical Architecture

The big data platform designed in this paper is to solve the need of the transformation from the traditional business structure to the Internet model in the media field. The media field has accumulated a large amount of business data over the years. These data have great differences in data format and data type. In order to store and manage these data uniformly and ensure the scalability of the system when adding other data types in the future. By studying the current general heterogeneous data storage mode, the integration scheme of heterogeneous data resources based on big data platform is determined, and the mainstream big data technology is adopted to ensure the technological advancement of the system platform in data storage. The technical framework of the big data platform designed in this paper is as follows:

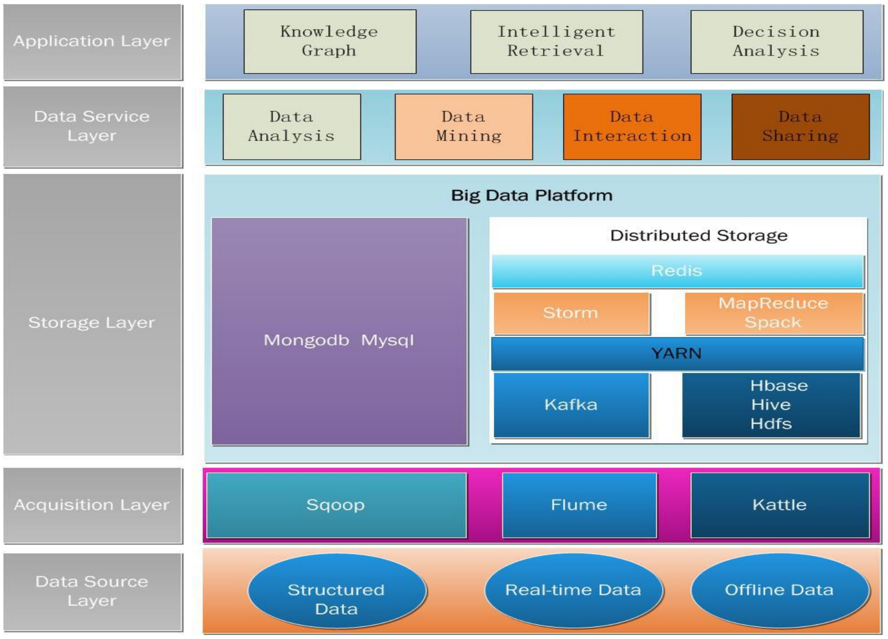

Figure 1: Overall Technical Architecture of Big Data Platform.

The overall technical framework of the platform is mainly divided into five layers: data source layer, data acquisition layer, storage layer, data service layer and application layer.

(1) Data Source Layer. The data sources involved in this system are mainly divided into three types: structured Atlas data, document data and other structured data, massive semi-structured and unstructured data such as pictures and videos, and real-time data that need real-time processing by the system.

(2) Data acquisition layer. According to different data structures of data sources, different technical means are adopted to collect data in data acquisition layer to . Sqoop technology is used to collect structured data, Flume technology is used to collect real-time data, and Kattle technology is used to collect data that need to be processed offline.

(3) Storage layer. All the data in the system are stored on the big data platform, using relational database mongdb and MYSQL to store structured data, Kafka distributed memory cache to store real-time data, and Hdfs, Hbase and Hive to store a large number of offline data.

(4) Data service layer. data analysis, mining, interaction and sharing are realized by loading data from traditional database or redis result database.

(5) Application layer.Processed data are aplied to Knowledge atlas, intelligent search, decision analysis.

3.1.1. Data Architecture

Different data types have different data flow processes on big data platforms. According to the overall framework of technical architecture, the data flow process on big data platforms is designed as follows:

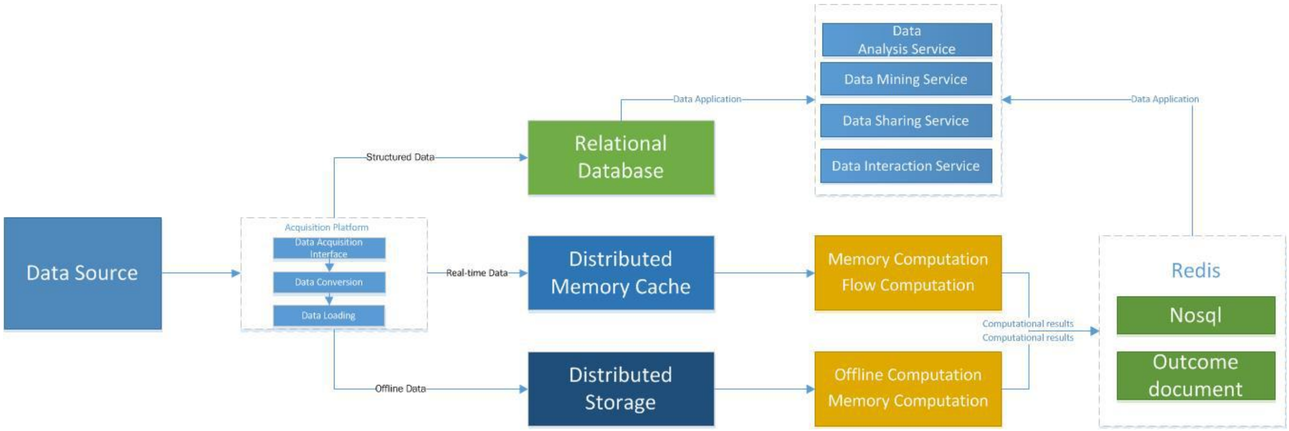

Figure 2: Structural Diagram of Data Flow in Big Data Platform.

The data flow on the big data platform of this design is mainly divided into three parts:

(1) Structured data is stored in traditional relational data warehouse through data acquisition layer, and then can be provided to data service layer for data analysis, etc.

(2) Massive offline data are stored on Hdfs, HBase and Hive through data acquisition layer. These offline data are further processed by offline computing and parallel computing. The processing results are stored on redis to provide data support for data service layer.

(3) Real-time data flow is sent to Kafka data cache through data acquisition layer. Real-time data is processed by memory computing and stream computing. The processing results are stored on redis to provide data support for data service layer.

3.1.2. Real-time Data Processing Flow

In this paper, the real-time stream data processing process is based on Flume data acquisition and Kafka data distribution technology. Flume collects real-time streaming data, processes the data simply and then transmits it to Kafka. Kafka caches and further processes the data, classifies and distributes the data to Storm according to Topic for data processing. Storm real-time streaming message processing engine calculates and further processes the data. Then the calculation results are stored in redis memory database.

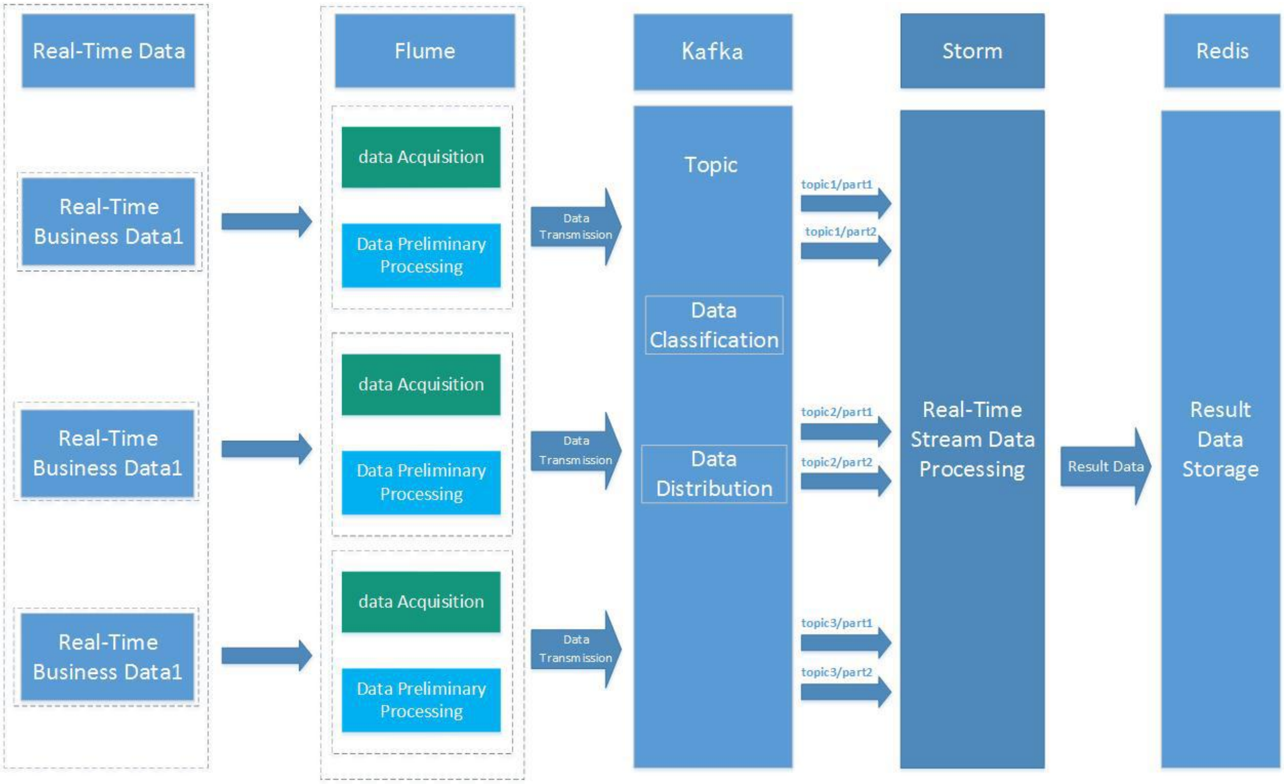

Figure 3: Real-time Flow Data Processing Flow Based on Flume and Kafka.

3.1.3. Offline Data Processing Flow

For a large number of offline data, the system adopts MapReduce framework and Spark memory computing framework to process. The principles of MapReduce framework and Spark memory computing framework have been explained in literature[7] and literature[8].This paper does not elaborate in detail. In this system, MapReduce framework is mainly used for batch calculation of offline data, while Spark framework is mainly used for memory calculation of offline data, as shown in the following Figure 4:

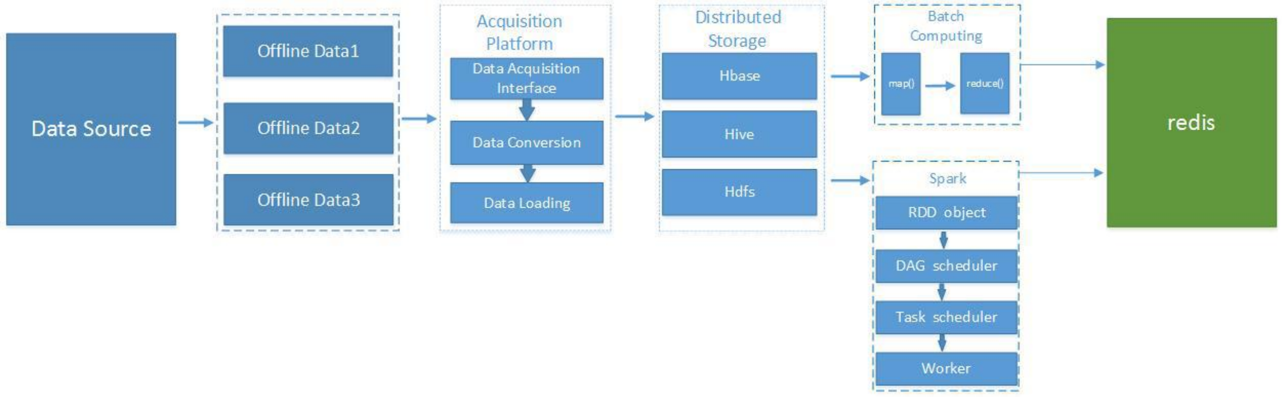

Figure 4: Offline Data Processing Based on MapReduce and Spark.

3.1.4. Platform Performance

By comparing the big data platform with traditional data warehouse in three aspects: storage capacity, computing capacity and query capacity of multi-source heterogeneous data, the feasibility and efficiency of the integration scheme of heterogeneous data resources based on big data platform are determined.

3.1.5. Data Storage Capacity

Traditional data warehouse is mainly structured data with high density and high value generated by analyzing business data, and it is very weak to process semi-structured and unstructured data such as pictures, text and audio. Traditional data warehouse belongs to single-node operation. With the increase of heterogeneous data sources, very complex ETL and other programs are needed. When the data volume reaches TB/PB level, performance will become a bottleneck. The Hadoop-based big data platform designed in this paper realizes the collection and storage of all kinds of data. First, it guarantees the performance of structured data in storage and use by using traditional data warehouse. Second, it guarantees the high availability of massive offline data and real-time data in storage and use by using big data platform technology. It effectively solves the performance problem of traditional data warehouse when facing multi-source heterogeneous data storage, and connects the data exchange problem between traditional data warehouse and big data platform through Hive technology.

3.1.6. Computing Power

Traditional data warehouse relies on single-node computing. When the amount of data needed to be processed is too large, it has great limitations on computing power. The big data platform based on Hadoop adopts distributed computing, through parallel computing of multiple nodes, and computing reduces data transmission locally. Storage based on big data platform greatly improves the computing power of data. MapReduce greatly improves the performance of big data platform in massive offline data processing and real-time stream data processing based on stream processing.

3.1.7. Query Ability

The big data platform technology designed in this paper can process the data in the data layer, and the data will be transmitted to the result cache. The advantage of using this kind of method is that when users query, the system can respond to the same query request at a high speed for a period of time without calculating every query. It effectively saves the waiting time of user query.

4. Conclusions

The data integration of heterogeneous resources based on big data platform is designed in this paper. By giving a technical architecture design based on Hadoop, it can solve the problem of multi-source heterogeneous data storage. By discussing the data flow of structured data, offline data and real-time data on big data platform, the data architecture design based on Hadoop. The big data platform in this paper is to solve the problem of resource storage in the media field, to meet the current platform resource storage requirements, and to meet the sustainable expansion of resource storage in the future. At the same time, the big data platform designed in this paper has been greatly improved in improving the computing power of resources and the performance of solving user query problems, meeting the user-oriented functional requirements.

References

[1] Zhang Bing, Zhang Rongxiao, Pan Yuping. Federated Database System [J]. Computer Systems & Applications, 1995, 4(1):50-54.

[2] Ning Z. Research on Technology of ETL in Data Warehouse[J]. Computer Engineering and Applications, 2002.

Yu Yonghong. Heterogeneous data sources integration based on XML middleware [J]. Journal of Hunan Institute of Science and Technology, 2006, 19(3):16-18.

[3] Li uojie, Cheng Xueqi, Research status and scientific thinking of big data[J]. Bulletin of the Chinese Academy of Sciences. 2012, 27(6):647-657.

[4] Ma Haoran, Simulate and Implement of Kafka distributed message system based on NS3[J]. Computer engineering & Software, 2015, 36(1):94-99.

[5] Li Chuan, E Haihong, Song Meina. Research & Application of real-time compute framework based on Storm [J]. Computer engineering & Software, 2014(10):16-20.

[6] Li Jianjiang, Cui Jian, Wang Dan, et al. Survey of MapReduce Paralel Programming Model [J]. Chinese Journal of Electronics, 2011, 39(11):2635-2642.

[7] Wu Libing, Qiu Xin, Ye Luyao, et al. Research on SQL-on-Hadoop systems [J]. Journal of Central China Normal University（Natural Sciences）, 2016, 50(2):174-182.

[8] Yao Jingwei, Yang Fujun. Application of Redis Distributed Caching Technology in Hadoop Framework[J]. Computer Technology and Development, 2017(6).