Processing Technology of Massive Human Health Data Based on Hadoop

Miao Liu[1, a], Junsheng Yu[1, b], Zhijiao Chen[1, c], Jinglin Guo[1, d], Jun Zhao[1, e]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100000, China;

[a]lm87v5@163.com, [b]jsyu@bupt.edu.cn, [c]z.chen@bupt.edu.cn, [d]478817379@qq.com, [e]zhaojun@chinasiwei.com

Keywords: Massive human health data, Hadoop, small files, prefetching.

Abstract. With the development of science and medical industry, people pay more and more attention to their health status. And massive human health data are generated in this process. As an important component of the cloud computing technology, the open source framework Hadoop provides us with a platform for storing and processing massive data. For the bottleneck of the existing Hadoop framework to deal with the small files in human health data, this paper proposes two optimization strategy: index optimization and metadata prefetching. At the end of the paper, the simulation results show that the method has excellent performance.

Introduction

With the rapid development and wide application of computer technology, medical information industry continues to accelerate the process. The amount of human health data presents a situation that is in geometric growth, and massive health data and complex data types bring enormous pressure for storage and processing. How we efficiently store and process large amounts of human health data, and provide efficient data service and data support have become a problem to be solved.

In the field of process massive human health data, Hadoop has a wide range of application in Hospital Information System (HIS), Picture Archiving and Communication System (PACS), auxiliary medical diagnosis and so on. Many research organizations and researchers have begun to use Hadoop to carry out research on medical services and clinical programs [1, 2, 3],Among them, Taylor, R.C. (2010) introduced in detail the Hadoop in bioinformatics applications [4], Schatz M.C. (2009) developed an open-source software package called Cloudburst, providing parallel algorithms for biomedical information analysis based on Hadoop [5].

Hadoop provides a stable system for sharing, analysis and storage. HDFS storage system provides high throughput to access application data, suitable for applications that have a large data set. However, small files occupy a large proportion of the human health data. There is a big bottleneck of the existing Hadoop framework to deal with the small files in human health data.

Hadoop Archives (HAR files) is introduced to reduce memory consumption caused by a large number of small files. Reading the files in HAR need to read the two layer index files and read the file data itself. And what's more, because HAR does not allow to continue to append the generated files, the HAR system access performance is affected.

In order to solve the above-mentioned difficulties an improved HAR index method is proposed in this paper. By improving the management of metadata through the redesign of the index, and optimizing the index strategy at the same time, the reading efficiency and the storage capacity of HDFS are highly increased when processing small files in human health data.

Hadoop 2.0 Ecosystem

Hadoop 2.0. Hadoop 2.0 ecosystem is a very good choice for us to deal with large numbers of human health data. Hadoop is an open source framework of the Apache foundation. It is a distributed software framework built on Linux cluster, able to store and process date in PB levels [6]. At the same time, Hadoop is a software platform that is easy to develop and run large data processing. People can make full use of the cluster of high-speed computing and powerful storage capacity, and they don't need to know the underlying details of the distributed framework. It is obviously that Hadoop provides a solution to the problem of massive data storage and processing [7, 8].

Under the operation of the Apache foundation, Hadoop have gained unprecedented development. Because of its own open source nature and ease of use, Hadoop has gradually developed into a general standard for large data processing. Hadoop Distributed File System (HDFS) and MapReduce, a framework for parallel computation, are the cores of Hadoop [9]. In addition, the rapidly developing Hadoop ecosystem provides convenient and effective means for large data collection, storage management and analysis. At present, there are many sub projects based on Hadoop, making the Hadoop ecosystem more colourful. Fig. 1 shows the relationship between the various projects in Hadoop 2.0 ecosystem.

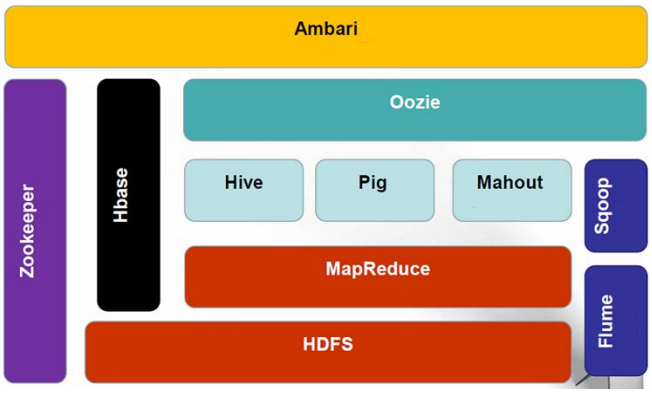

Fig. 1 Hadoop 2.0 ecosystem

Hadoop Distributed File System (HDFS). HDFS is the distributed file system in Hadoop framework. Compared with the existing common types of distributed file system, HDFS has many similarities, but also has its own unique characteristics. The feature of HDFS is that it can achieve high fault tolerance on inexpensive hardware. HDFS applies to large data applications and provides high throughput data access capabilities. The architecture of HDFS is shown in Fig.2, which is a Master / Slave structure.

A HDFS cluster consists a single NameNode, the master server that manages the file system namespace and access to the files stored in the cluster. As the manager of HDFS, NameNode can coordinate the actual data storage in DataNode. Other servers in the cluster acts as the role of DataNodes. Data are stored in DataNodes, which are the working nodes of HDFS.

In the system, a file is divided into one or a lot of data blocks that are storage in a group of DataNodes. And through the data block copy mechanism, one block is copied into multiple data blocks (the default is 3) and stored in other DataNodeso enhance the fault tolerance. NameNode is responsible for the implementation of the basic operation of the file system namespace, such as file access and file rename, and determine the specific mapping of each data block to the DataNodes. DataNodes are responsible for the reading and writing requests from file system client. And DataNodes create, delete and copy data blocks following NameNode orders.

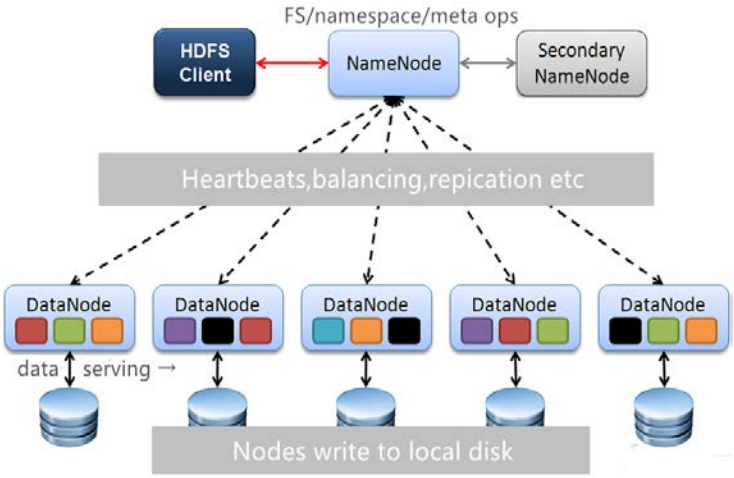

Fig. 2 Hadoop 2.0 ecosystem

Processing Technology

For massive human health data, Hadoop framework not only can provide storage for the data, but also provides a new way for efficient processing of data. By building an efficient, secure, scalable distributed cluster, the use of Hadoop technology can meet the requirements of efficient storage of massive health data.

People's own physical condition monitoring records and the data generated during the process of medical treatment are part of human health data. Human health data include doctor's prescription, drug list, and patient's registration information, personal health information files, CT and other image information, etc. In the data files generated by the hospital, the traditional small files occupy a large proportion, and the size of these small files are usually far less than the size of the HDFS blocks.

HDFS is designed to store and deal with large files. A large number of small files will occupy a lot of memory of NameNodes, resulting in a waste of namespace. Hadoop Archives (HAR) can be used to control these problems. The user can arrange small files in the archive (.Har) and access the archive as well as the normal file. This method reduces the number of files, thereby reducing the memory consumption of NameNode and reducing the number of access to NameNode. HAR package small files to large files through a MapReduce task, and the original file can be transparently accessed in parallel.

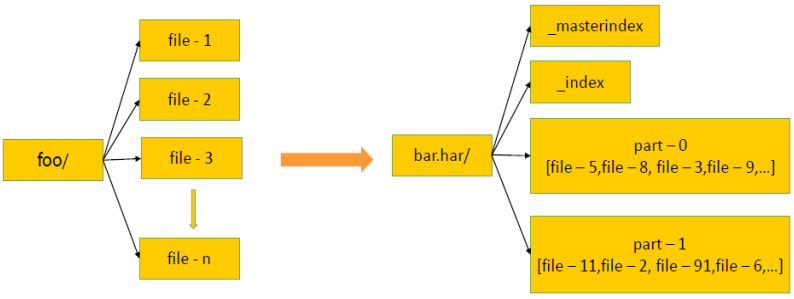

Fig. 3 Structure of HAR file

As shown in Fig.3, the small files are stored in a plurality of parts of the archive files and data stay separate according to the index. The file index structure is shown in Fig.4. Reading files in the HAR file is more slowly than in the HDFS because reading a small file in the HAR need to access indexes with two levels. In addition, once the HAR file is created, you can only recreate the HAR file when you want to add files. As a result, it will spend a lot of time.

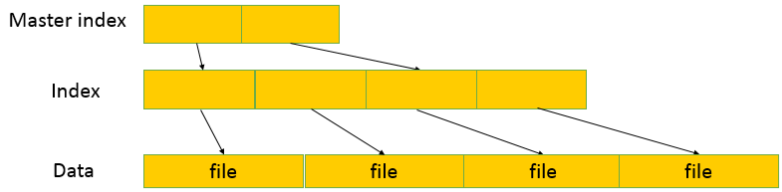

Fig. 4 Structure of HAR index

Optimization Strategy

To solve the problems above, in this paper optimization strategy mainly includes two aspects: (1) Merging small files to reduce the NameNode memory consumption and to improve the performance; (2) The new index based on HAR realizes HDFS client index prefetching and reduce NameNode load.

Index Optimization. In order to improve the access efficiency of the existing HAR technology, this paper combine the single layer index and hash, giving up the double layer index. A lot of index files are separated by creating a hash table. The new index structure is shown in Fig.5.

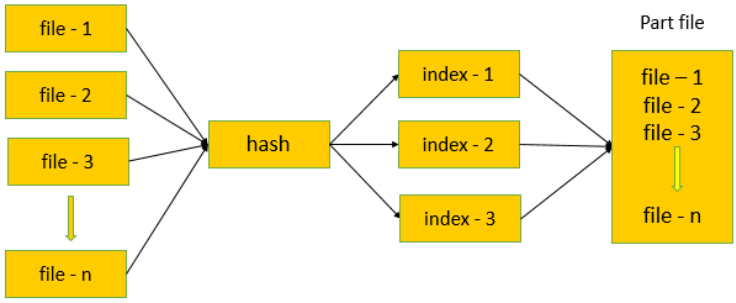

Fig. 5 Structure of new HAR index

In this index mechanism, filing procedure derives a hash code according to the file name and uniform hash algorithm when accessing a file. Then it will locate the index file that contains metadata according to the key value and hash code that are drawn from the number of index file.

As shown in Fig.5, suppose that the Hash code for file-1 is 1002. With this Hash code divided by 4, the number of files, we can get number 2, and this number is positioned to the index_2 file of metadata files. The actual files will also be maintained in the Part file and it is similar to HAR structure. In this way, it actually turn the tree structure into the combination of tree and hash data structure, and sacrifice a certain space efficiency for faster reading and writing speed.

Metadata Prefetching. The index file generated by the new index mechanism is stored in NameNode. In order to reduce the NameNode load, we add the index prefetching mechanism that is specially used for small file reading to the HDFS client and NameNode. When HDFS client attempts to read a small file in HAR file, we prefetch the metadata information of other small files that are in the same part files. Suppose there is significant correlation among the small archived files, and the large possibility to be accessed together, the client does not need to start the requests to the NameNode because there are small file metadata in the HDFS client cache. Under this prefetching mechanism, the request to NameNode will be greatly reduced, thus improving the performance of the NameNode.

Simulation and Experimental Results

Experimental Environment. In this paper, the processing technology of big data of human health needs to be carried out on the Hadoop cluster. So we carried out related experiments on a cluster system consisting of three computers. And one computer acts as the role of the NameNode, the other two computers act as the role of DataNodes. Specific experimental conditions are as follows:

Development platform: Linux operating system, Hadoop-2.6.0-cdh5.4.8

Development tools: Eclipse 4.2.1, Java 1.6.

System size: three ordinary PC machines.

PC specifications: 2.1 GHz dual-core Intel Core processor, 2 gigabytes of ram, 320 GB hard disk.

Experimental Results. The experiment mainly focuses on the comparison of HAR method and improved HAR method in the following two aspects: file reading time and NameNode memory consumption.

(1)File Reading Time

The small file storage efficiency of HDFS is very low because it is designed to achieve large file storage. In order to directly reflect the performance of handling small files of the system, this paper selects 20000, 40000, 80000 sets of small files to test the reading time.

As shown in Fig.6, applied to the same situation we can find that the improved HAR method is faster than the HAR method while reading small files in human health data.

Fig. 6 Performance of handling small files

(2) NameNode Memory Consumption

Similarly, we select 20000, 40000, 80000 sets of small files to test the NameNode Memory Consumption. The results are shown in Fig.7.

According to Fig.7, compared with HAR, we can find the improved HAT can bring some improvements in the memory consumption.

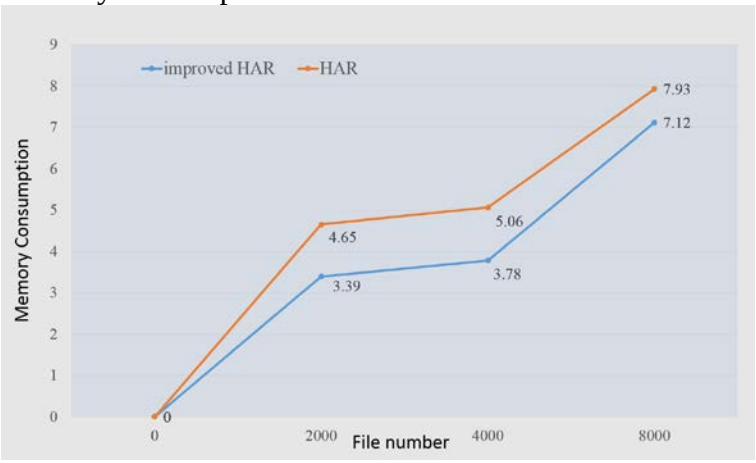

Fig. 7 NameNode Memory Consumption of HDFS

Summary

There are lots of small files in massive human health data. Consider the performance degradation of HDFS while processing these files, this paper proposes a new index strategy based on existing HAT technology. The two layer index are replaced by a single layer index to reduce the indexing time.

To optimize the load balance of HDFS, we improve HAR by prefetching the metadata information of related part files. Experimental results show the processing technology proposed in this paper can obviously improve the performance of small file storage access. And it is a valuable technique for Hadoop framework to break through the bottleneck of processing small files in massive human health data.

References

[1] Horiguchi H, Yasunaga H, Hashimoto H, et a1.A user—friendly tool to transform large scale administrative data into wide table format using a mapreduce program with a pig latin based script[J]. Bmc Medical Informatics & Decision Making,2012,12(30):4456-4471.

[2] Liu B, Madduri R K, Sotomayor B, et a1.Cloud-based bioinformatics work flow platform for large-scale next-generation sequencing analyses. [J].Journal of Biomedical Informatics,2014, 49(6): 119-133.

[3] Santana-Quintero L.HIVE-Hexagon: High-Performance, Parallelized Sequence Alignment for Next-Generation Sequencing Data Analysis [J]. Plos One,2014,9 (6): e99033.

[4] Taylor R C. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. [J]. Bmc Bioinformatics,2010,11 suppl 12(6): 3395-3407.

[5] Schatz M C. CloudBurst: highly sensitive read mapping with MapReduce [J]. Bioinformatics,2009,25 (11): 1363-1369.

[6] Apache Hadoop [EB/OL]. Available from: http://hadoop.apache.org /.

[7] Yu HOngyong, Wang Deshuai. “Research and Implementation of Massive Health Care Data Management and Analysis Based on Hadoop” In 2012 4th International Conference on Computer and Information Sciences, August 2012: 514-517.

[8] Dalia Sobhy, Yasser El-Sonbaty, Mohamad Abou Elnasr. “MedCloud: Healthcare Cloud Computing System”, December 2012: 161-166.

[9] G S Aditya Rao1 PP. Big Data Problems:Understanding Hadoop Framework [J]. International Journal of Scientific Engineering and Technology, 2014, 1: 40-42