American Journal of Applied Scientific Research
2019; 5(1): 6-16
http://www.sciencepublishinggroup.com/j/ajasr
doi: 10.11648/j.ajasr.20190501.12
ISSN: 2471-9722 (Print); ISSN: 2471-9730 (Online)

Innovation Method of Distributed Storage for Huge Data of Geological and Mineral Resources Based on Hadoop

Li Chaokui[1, *], Zhao Yanan[1], Xiao Keyan[2], Chen Jianhui[1]

[1]National-Local Joint Engineering Laboratory of Geospatial Information Technology, Hunan University of Science and Technology, Xiangtan, China

[2]Institute of Mineral Resources, Chinese Academy of Geological Sciences, Beijing, China

Email address:
chkl_hn@163.com (Li Chaokui)

[*]Corresponding author

To cite this article:
Li Chaokui, Zhao Yanan, Xiao Keyan, Chen Jianhui. Innovation Method of Distributed Storage for Huge Data of Geological and Mineral Resources Based on Hadoop. *American Journal of Applied Scientific Research*. Vol. 5, No. 1, 2019, pp. 6-16.
doi: 10.11648/j.ajasr.20190501.12

Received: January 24, 2019; **Accepted**: March 4, 2019; **Published**: March 28, 2019

Abstract: With the emergence of big data of TB and PB geological and mineral resources, the storage of large geological data has become a worldwide problem puzzling geologists. The traditional storage and service model of geological data is facing a great challenge. For example, when the scale of data increases dramatically, general relational database can not solve the problem of insufficient scalability, stability and efficiency of database system. In response to the above problems, this paper proposes a new method of geological and mineral data storage based on cloud computing environment combined with hadoop. Taking the mineral resources potential evaluation data of Chongqing as the research object, The proposed method in this paper is compared with the traditional Oracle database storage method in data storage experiments: (1) Small file optimization comparative experiment; (2) Hadoop and Oracle comparative experiment. The performance of writing operation, memory occupancy, data import and data export are tested in different way, and the comparison chart of performance is given. The experimental results show that the new storage method proposed in this paper is more efficient than the traditional method. At the same time, it effectively overcomes the problem of small file storage in Hadoop storage. The research results provide a new technical for the storage and management of geological and mineral data all over the country.

Keywords: Geological and Mineral Data, Hadoop, Oracle, Storage of Small Files

1. Introduction

Geological data is an important information resource formed by geological work. Massive geological data accumulated over many years in China provide an important support for national economic and social development [1]. The geological industry has a long history and the accumulation of geological data is rich [2]. Therefore, it is of great practical significance to study the storage and retrieval methods of massive geological data. The construction of geological database in China began in the middle and late 1980s, and it was later than the western countries. So far, the geological database has mainly included oil and gas and solid mineral resources exploration, geological environment investigation, hydrogeology and mineral resources potential evaluation. The number has also had a certain scale [3]. The main storage methods for relational databases, such as ArcGIS based information system developed by Zhang Bo [4], through the ArcCatalog built-in function to establish the space of geological disasters in Fugu County Based on Personal Geodatabase database; PU Kai [5] in the creation of geological spatial database system to Oracle as the basic tool for all storage and management of spatial data, under the support of ArcSDE implementation the unified storage and management of spatial data, and the application of Liu Canjuan [6] in Microsoft Visual; Studio. NET environment, combined with the technology of Oracle 10G database system, according to the application requirements of the design and completed the storage management of geological data. However, with the continuous accumulation of data and the appearance of massive data, the general relational database

cannot cope well with the problem of system expansion and performance when data size is increasing rapidly. Hadoop is a good solution to the problem of large data storage. Because the Hadoop process is mainly through the Hadoop distributed file system (HDFS) to achieve the files [7], this paper proposes a new storage method, which uses Hadoop's HDFS as the underlying storage architecture for HBase metadata storage mechanism of the stored data of Geology and mineral resources. The mineral resources potential evaluation data of Chongqing are taken as experimental objects to carry out comparative experiments. The results show that the storage method proposed in this paper is obviously better than the traditional relational database storage mode.

2. Design of Data Storage Method Based on Hadoop

Geological data are multiple source, multiple element, heterogeneous, spatial-temporal, directional, correlation, randomness, fuzziness, nonlinearity and so on. Therefore, the geological big data is of diversity [8], and it is multiple-dimensional, structured and unstructured. As a distributed file system of Hadoop, HDFS [9] can be used to store data that are basically sequential, massive structured and unstructured. It is suitable for running on common hardware [10], and HBase's main advantage is fast and random access to data. Therefore, only the combination of the two can better realize the storage and retrieval of mass geological and mineral data such as images, vectors, text and so on. The following methods: the design of this store actual geological and mineral data is stored in HDFS (direct memory for image geological data entry big HDFS; text, vector geological data entry, small to the small files and other measures, and then stored in HDFS), HBase index data storage. The application writes the results in a new HDFS file and updates the meta-data based on the HBase.

2.1. Hadoop System Framework

Hadoop provides users an open source, big data distributed processing framework, including distributed file system HDFS, distributed processing model MapReduce, and non relational database HBase. The framework helps users to complete the storage and processing of large data through functional programming and operating interfaces on the basis of understanding the underlying implementation [11]. Its system framework is shown in Figure 1.

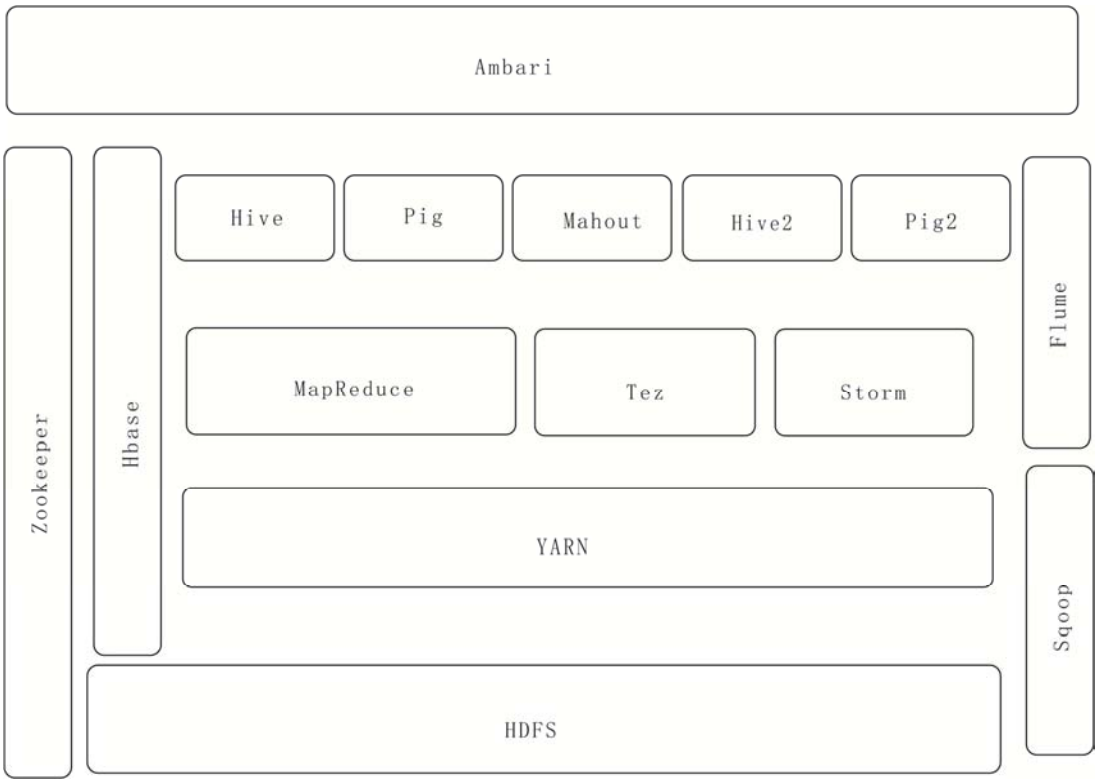

Figure 1. *Hadoop System Framework.*

2.1.1. HDFS

HDFS is a distributed file system. It provides data storage function in the whole software ecosystem, and provides data stream access interface. Compared with centralized data storage mechanism, it can effectively improve the throughput of system data access. HDFS is deployed in a cluster consisting of a large number of common PC servers. It adopts a multiple access file access model written at once, and has strong fault tolerance and good platform portability.

2.1.2. Map Reduce

MapReduce is a programming model to deal with large data [12]. Two simplification processes of Map and Reduce are

used to divide the complex tasks into independent sub problems and automatically dispatch computing nodes to deal with these sub problems. Using data / code mutual location technology can effectively reduce data communication between nodes, and can also hide the function of the system layer for application developers to implement details.

2.1.3. HBase

HBase is a distributed database that provides column storage, real-time reading and writing, and high performance. It is a specific implementation of the non-relational database NoSQL. HBase is used to store unstructured and semi-structured loose data, and can be used to implement the retrieval function through the primary key (RowKey) and the primary key range.

2.2. Geological Spatial Data Storage Model

2.2.1. Storage Model Based on Relational Database

A relational database is a database that uses a relational model to organize data. The relational model first proposed by IBM researcher Codd in 1970 [13]. Geographic Information System (Geographic Information System, referred to as GIS) is a new technology [14], which is mainly supported by computer science and applied specifically to the application of spatial data. GIS can be used in query management of geological data, spatial analysis, visualization and so on. Combining GIS with Oracle large spatial database and developing database system based on certain criteria is an existing storage plan for massive geological data storage and management. In the past half century, China has invested hundreds of kinds of national basic geological databases, and the total investment data has exceeded 100TB, and the number is increasing by [15]. Most of these data management is based on GIS platform, which is organized, produced and displayed in combination with large database storage technology and computer technology. Some of them even take three-dimensional modeling technology for integrated management. However, these management applications seldom make full use of the functions of database. Therefore, in the face of massive data storage, the existing database will become overwhelmed.

2.2.2. Distributed Storage Model

With the rapid growth of geological data, the storage mode based on relational database has been difficult to meet the storage requirements of massive geological data. At present, with the emergence of the cloud platform and the development of large data, more and more scholars have carried out the research of data storage using Hadoop related technology. In this paper, a complex geological data storage system is designed by using the advantages of HDFS and HBase. The system stores the original geological data in HDFS, stores the file directory in HBase, and realizes the storage and efficient retrieval of geological big data. The system architecture is shown in Figure 2.

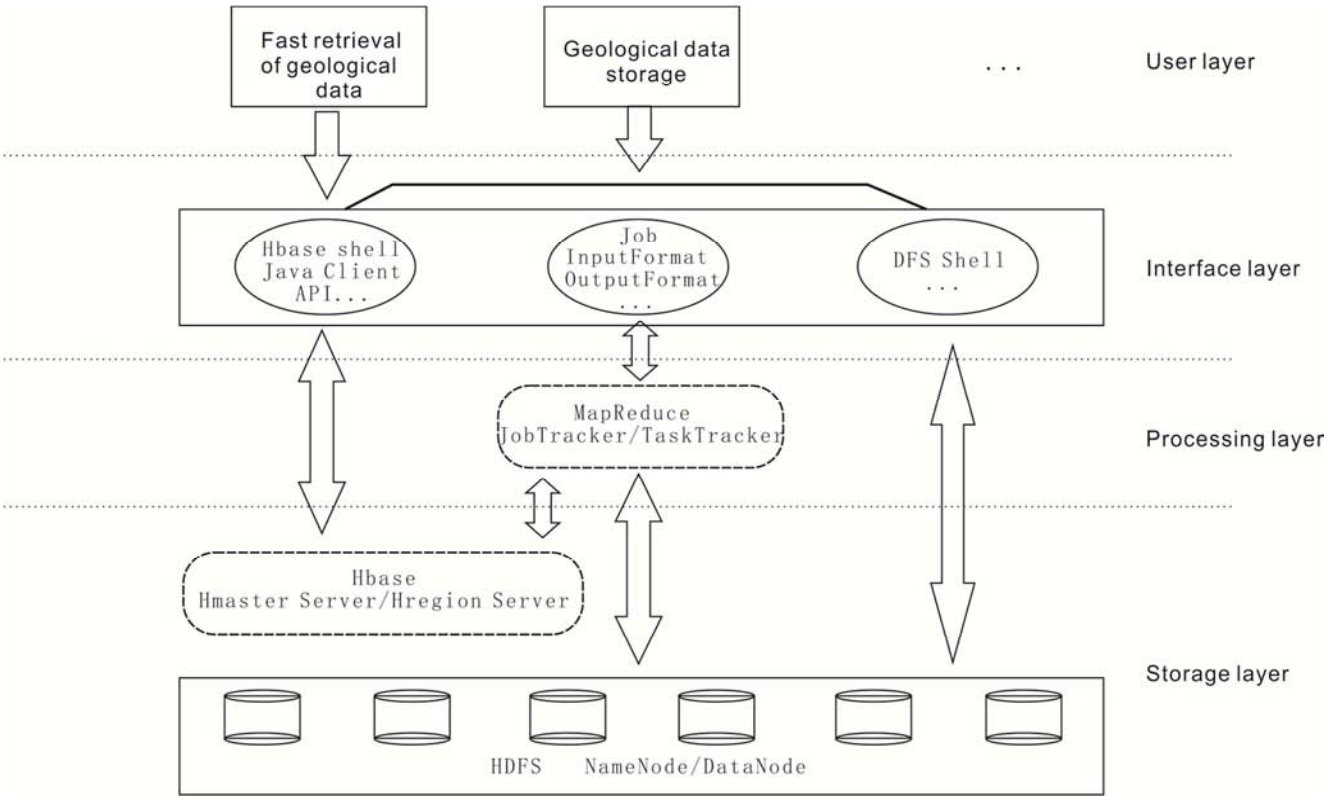

Figure 2. *Database System Architecture.*

(i) HDFS File System

1. HDFS Architecture

HDFS is mainly composed of NameNode and a series of DataNode. Among them, NameNode and Secondary

NameNode collaborate to manage the HDFS's directory tree and related metadata files. NameNode fsimage is a metadata file, which is stored in the file system directory tree; edits is a metadata operation log, and the size effect of NameNode starting and running speed, so the need for Secondary NameNode by HTTP get to obtain regularly the two documents from the NameNode, merge them and generate a new fsimage file, and return the new file to the NameNode, then NameNode update the local fsimage file, and the empty the operation log of edit, so as to enhance the management performance of HDFS node [16]. HDFS architecture, as shown in figure 3.

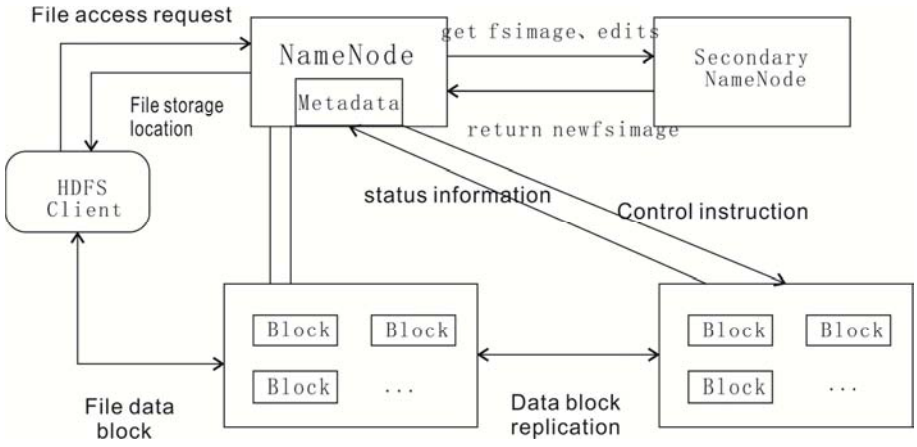

Figure 3. *HDFS Architecture.*

2. HDFS File Operations

HDFS file operations include read and write two major processes, both of them are initiated by the HDFS client.

a. File Write

(1) Initialization examples will be written to the File by calling a create() function from Distributed File System in the client.

(2) The NameNode will first verify that the system do not have file to create and ensure that the client has the right to create a file, otherwise it will have an abnormal occurrance.

(3) FSData Output Stream will cut file into many packets after finished checking. It will coordinate the NameNode and DataNode and use write packet method to put the object to DataNode.

(4) It will return an Ack Packet to client when the last file was written successfully. The client will use the close method to close all data streams and report the accomplish information to NameNode.

This paper selects the evaluation of potential mineral data of Chongqing city "Chongqing city of South Dabashan intrusive Chengkou rock type gold deposit prediction work area remote sensing image GEOTIF (230MB)" as test data stored in HDFS. The core code is as follows:

FSDataOutputStream fsDataOutputStream fs.create (New = Path ("/opt/hadoop/dfs/Data/ / Chongqing / Chongqing City gold Chengkou Southern Dabashan intrusive rock type gold deposit prediction work area remote sensing image GEOTIFF.tif")); / / open a file output stream in HDFS

FileInputStream fileInputStream = new FileInputStream ("/home/hadoop/Desktop/ Chongqing, Chengkou Southern Dabashan intrusive rock type gold deposit type prediction area remote sensing image GEOTIFF.tif"); / / open a file input stream in HDFS

IOUtils.copy (fileInputStream, fsDataOutputStream);

b. File read

(1) The client calls the open function and initiates the RPC request to the remote NameNode. At the same time, it calls the metadata node and gets the data block information of the file.

(2) The client calls the read() function on the open file stream. DFSInputStream connection save this file's recent DataNode and read the data back to the client.

(3) It will break the link to the DataNode after reading the current block of data and then select the next DataNode to obtain the next block of data.

(4) It will call the close function to close the data flow after all data is read.

This article selects the vector data of HDFS's data as the object of reading and the core code is as following:

```
RemoteIterator<LocatedFileStatus> listFiles = fs.listFiles(new Path("/opt/hadoop/dfs/data"), true);
While(ListFiles.hasNext()) {
LocatedFileStatus file = ListFiles.next();
System.out.println(file.getPath().getName());}
```

3. Small File Storage Structure

HDFS is a distributed file system with high fault tolerance and low cost, which is designed to enlarge the file storage. While it will cause problems when dealing with small files [17]. Geological spatial data compared with other spatial data has some characteristics. They have large data volume, more thematic content and large amount of information. Geological spatial data mainly has large vector data and remote sensing data file data that is very suitable for HDFS storage method. We have Chongqing gold mine as an example and show the potential evaluation data in the figure 4.

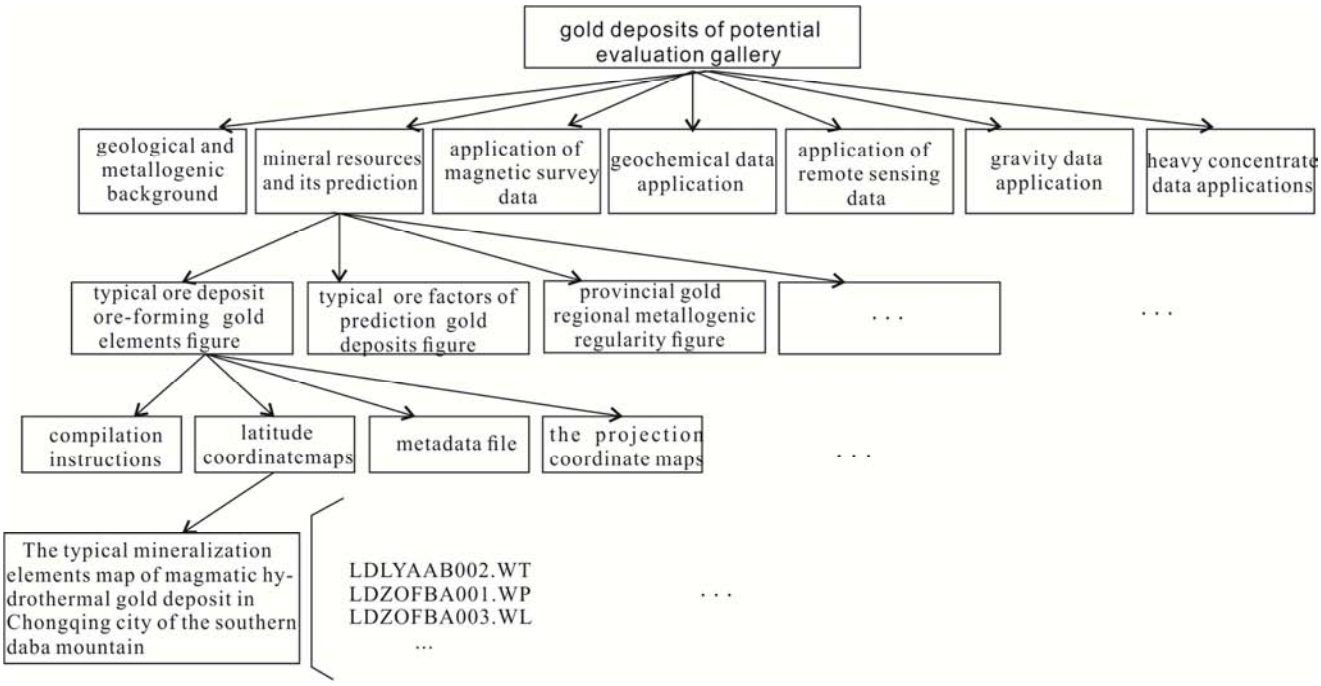

Figure 4. *Potential Evaluation Data (gold mine in Chongqing as an example).*

However, the Experiment shows that the vector layer is not use the "map" as a memory cell and then stored in a single block memory. But it uses the "WL", "WT" files as a memory cell and then stored in it, occupy a block of memory. This will produce numerous small files and cause HDFS storage bottlenecks. The specialized storage conditions is as shown in figure 5.

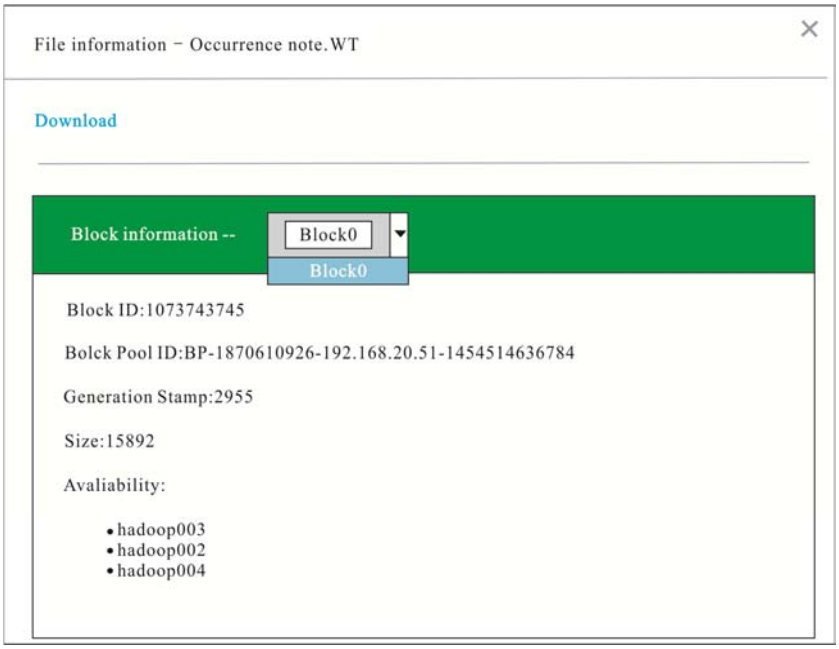

Figure 5. *Occurrence note. WT File Information.*

Figure 5 shows a single vector elements from "the typical mineralization elements map of magmatic hydrothermal gold deposit in Chongqing city of the southern DaBa mountain ". From the figure 5, we can see that it only have 15892 bytes but occupies a block node. It occupies a single split and push the storage pressure to the cluster.

a. Hadoop's own solution

Hadoop system provides three kind of solutions to the problem of small files, which respectively are Hadoop Archive [18], Sequence File [19] and Combine File Input Format.

HAR (Hadoop Archive) file archiving technology are completed by making the small files on the HDFS package into HAR, which used to alleviate the problems of lots of small files in the Name Node memory consumption. But read files from HAR need to be read two layers which include

index data file and the file itself. HAR does not support archiving the file compression, so HAR handling small files has a low efficiency.

Sequence File is a binary file technology provides by HDFS and it through the < key, value > serialization to SequenceFile to implement many small file merging. At the same time it supports blocks of data compression, significantly reduce the Name Node memory and reduce the disk space data nodes. However, this method did not establish the corresponding small files to large file mapping that made reading efficiency lower.

Combine File Input Format is a new input format that combines multiple files on the HDFS into a single split to improve efficiency. But this method is suitable for the existence of large number of small files in HDFS, and Combine File Input Format is an abstract class, which requires users to implement entity classes themselves.

Compared with the above three methods, the Combine File Input Format scheme is more feasible. However, a large number of small file data have not been stored on the Hadoop platform. If the storage is first stored and then optimized, the efficiency is lower. So the best strategy is to combine the vector factors under the "layer" into a large file to improve the storage performance of the small files.

b. IPutMerge solution

The *Hadoop combat* provides us with a PutMerge program to store small file to HDFS. It was used to merge local disk file and then copied to the HDFS. But the programme only works for local single folder's data storage. The efficiency is relatively lower if we have large number of different paths. Therefore, we design an IputMerge (Improved PutMerge) program that we can store data from different paths. And the path of data will be recovered automatically when the data is downloaded and it has more convenient operation.

The total time to manage split and the total time to build map tasks determine the whole execution time of the job. Therefore, the reasonable size of the partition determines the quality of the load balancing. Here we set the size of the 128M to block size. Second, we call Configuration conf to get the File System instance you need. The Configuration object encapsulates the configuration of the client or server, and is implemented by setting the configuration file to read the class path (core-site.xml). After the File System instance, the factory method FileSystem. getLocal (Configuration CONF) is called. The Hadoop file API uses the Path object to write the file and directory name, and uses the FileStatus object to store the metadata of the file and directory. The IPutMerge program will call a recursive algorithm to merge all the files in the local different directories. We use the ListStatus () method of FileSystem to get the local directory. FSDataInputStream is a subclass of the Java standard class java.io. Data Input Stream, which provides the function of random access, which we use to write the data to HDFS. In order to implement IPutMerge, we create a for loop to read all the files in inputFiles one by one. The part of the code is as follows:

```
Configuration conf = new Configuration();
.....
Path localPath = new Path((“/home/hadoop/Desktop/
Chongqing / typical ore deposit
ore-forming gold elements figure”));
Path hdfsFile = new
Path((“/opt/hadoop/dfs/data/Chongqing/ Gold deposits of
potent-
al evaluation gallery / mineral resources and its prediction /
typical ore deposit ore-
forming gold elements figure”));
if(out==null) {
out = hdfsFS.create(hdfsPath);
}
readLocalFileWriteToHDFS(localPath, localFS);
.....
FileStatus[] inputFiles = localFS.listStatus(localPath);
 for (int i =0; i <inputFiles.length; i++) {
 boolean isDir = inputFiles[i].isDirectory();
 if(isDir) {
 readLocalFileWriteToHDFS(inputFiles[i].getPath(),
localFS);
 }
 if(inputFiles[i].isFile()) {
……
 }
}
}
```

Finally, We call the MapReduce program to upload the data to HDFS. Shell command is as following:

[hadoop@hdmaster1 Desktop]$ hadoop jar HDFSpd.jar Data.HDFSpd2

(ii) HBase Database

HBase is a distributed, column oriented open source database. It is based on the Google paper "Bigtable: A distributed storage syatems for structured data" open source to achieve [20]. HBase takes HDFS as its underlying storage system, analyzes massive data with HadoopMapReduce, and uses Zookeeper to manage the data in a unified way.

Having the vector of geological data as an example, this paper uses the data of each province as a table, each mineral corresponds to a line, with the corresponding mineral provinces said Hangjian information; different column families represent "comprehensive research data", "provincial mineral resources"; with different column chart elements of different generations. The HBase column storage table is designed as table 1.

Table 1. *HBase Column Storage Design Table.*

RowKey	TimeStamp	Column Family SPE					Column Family S_SPE				
		Minerals ID	Project ID	Map ID	route	other attributes	minerals D	Project ID	map ID	route	other attributes
minerals RowKey	TimeStamp	MID	SID	FID	LCT	...	MID	SID	FID	LCT	...

In the table, the column "SPE" represents "comprehensive research results" and "S_SPE" represents "provincial mineral resources". The "MID" in the column family "SPE" represents the coding information of the mine (coded information is prescribed by the state), for example, 01 represents "iron", and the 02 represents "manganese". The storage case is as follows:

4319 column=SPE: LCT, timestamp=1478462443280, value=hdfs://192.168.20

.51/opt/hadoop/dfs/data/HUNAN

4319 column=SPE: MID, timestamp=1478462443280, value=4319

The case is Hunan province "sulfur" information encoding and storage path information.

3. Experimental Verification

3.1. Comparison Experiment of Small File Optimization

In order to verify the superiority of the optimization scheme, this paper compares it from two aspects: (1) the write performance comparison of experimental data on HDFS; (2) the comparison of HDFS NameNode memory occupancy results during file operation.

3.1.1. Experimental Environment

In this paper, we use the four nodes Hadoop cluster, which including one NameNode and three DataNode. The specific configuration is shown in Table 2 below.

Table 2. *Cluster configuration.*

component	configuration
Hadoop version	Hadoop2.7.2
Operating system	Red Hat Enterprise Linux Sever release 6.7(Santiago)
Linux Kernel version	2.6.32-573.el6.x86_64
JDK	1.7.0_91
Wideband of network	100MB
NameNode	8 nuclear 2.40GHz CPU, 16G Memory, 600G Hard disk, number 1
DataNode	8 nuclear 2.40GHz CPU, 8G Memory, 600G Hard disk, number 3

3.1.2. Writing Performance Comparison Test

In this experiment, we select 500, 1000, 10000, 20000, 30000, 40000 small files of 1~1000KB size to store experimental data, and do data storage experiments respectively. Record their respective storage time and take the average of the results of the three experiments. The results of the experiment are shown in Figure 6 below.

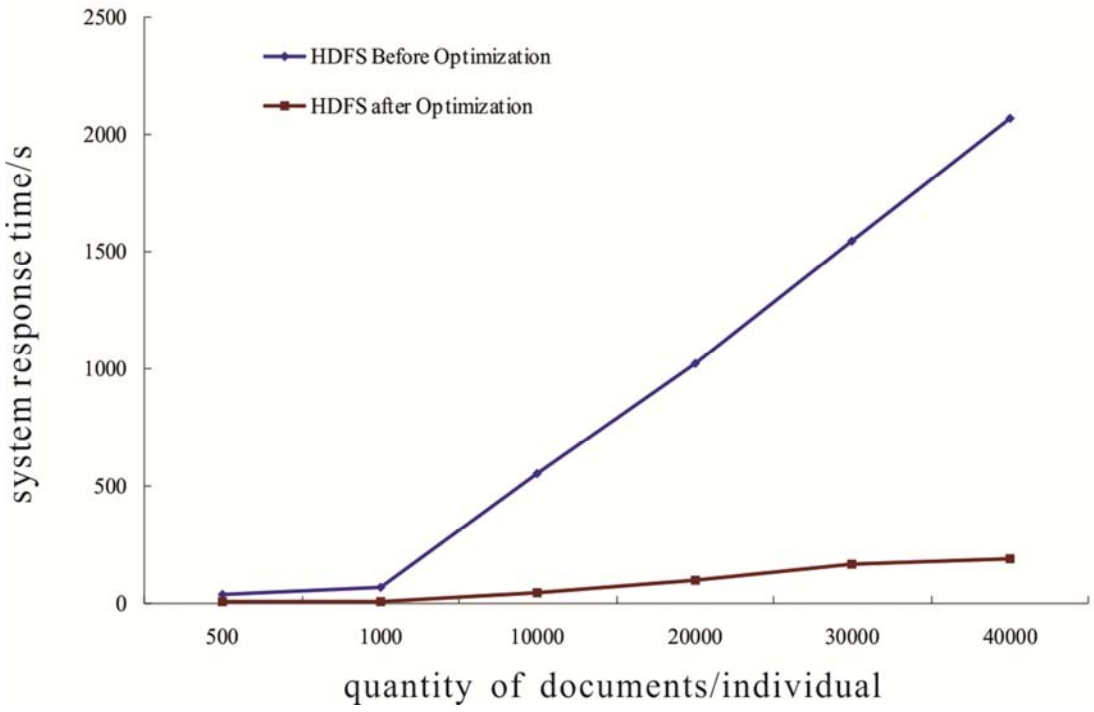

Figure 6. *Performance Test Results of Write Operations.*

Figure 6 shows that the write operation of the HDFS optimized with the IPutMerge method is significantly faster than the original HDFS. This is because, when writing a file to HDFS, NameNode has to create and allocate data block operations, while in optimized HDFS, when N large files are composed of small files, NameNode is called only once. In addition, Client uses a caching mechanism for large files, which sends requests for 128MB data to NameNode, not every small file.

3.1.3. Memory Occupancy Contrast Test

Massive small files will consume the master node memory and causing the problem of NameNode bottleneck, so this will seriously influence the performance of HDFS. In order to analyze the NameNode occupancy in a HDFS based distributed file storage system, the design experiment is as follows.

In this experiment, we select 500, 1000, 10000, 20000, 30000, 40000 small size 1~1000KB files as experimental data, write files to HDFS in half an hour, and observe the memory changes of NameNode system. Take the average of the results of the three experiments. The amount of memory usage is shown in Figure 7 below.

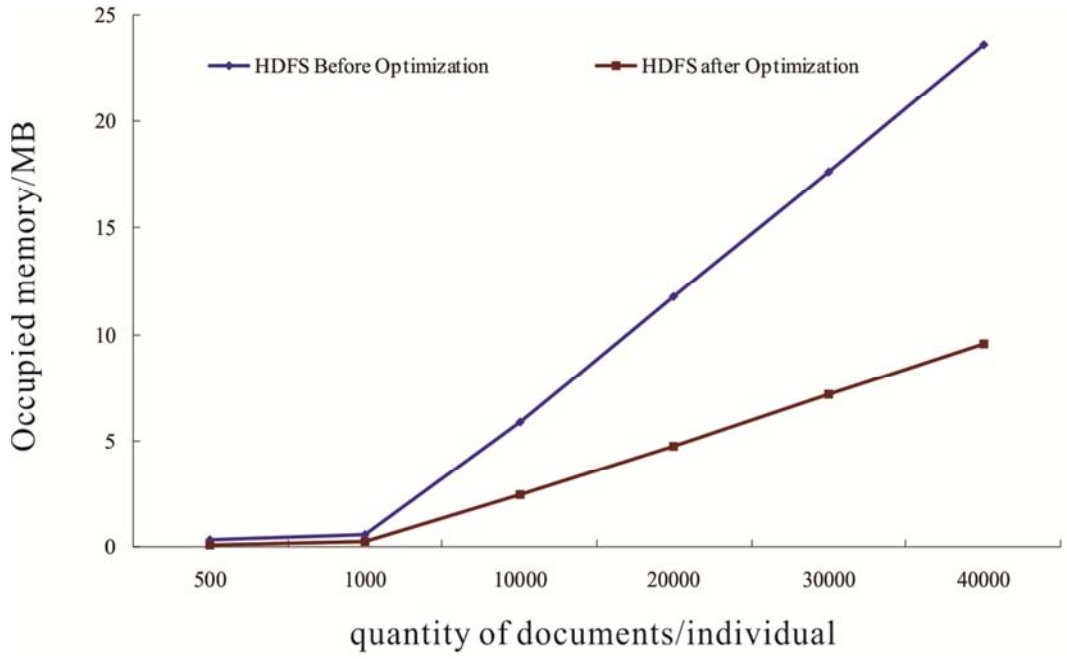

Figure 7. *Memory Occupancy Test Results.*

Figure 7 shows the two cases of the memory usage. It can be seen from the results that the effect of the optimization is not obvious if the number of files under 1000. However, with the rapid growth of the small files, it uses less memory than the original HDFS. This is because that the NameNode is mainly save metadata and the metadata of block's file. In the original HDFS, each small file occupies a block, so that the NameNode takes up a lot of memory to store the data. The optimized HDFS, NameNode preserves a single composite file and block metadata, so the system memory footprint will be greatly reduced.

3.2. Comparison Experiment Between Hadoop and Oracle

This paper compares two kinds of spatial data storage methods: (1) traditional spatial data storage mode, that is, the combination of Oracle and MapGis for spatial data storage; (2) Hadoop platform, data storage through HDFS. The experimental results are analyzed to verify the superiority of the method.

3.2.1. Experimental Environment

Table 3 shows the hardware environment that we used in the test. Oracle uses a single node and Hadoop is deployed on a cluster of four computers. The specific configuration is as table 3.

Table 3. *Test Database Hardware Environment.*

component	Oracle	Hadoop
operating system	Windows 7 professional	Red Hat Enterprise Linux Server release 6.7(Santiago)
number of servers	1(single node)	4(Hadoop cluster)
wideband of network	100MB	100MB
CPU performance	4 nuclear	8 nuclear
CPU main frequency	2.80GHz	2.40GHz
memory	8G	8G

3.2.2. Data Import Contrast Experiment

This experiment takes the mineral resources potential evaluation data of Chongqing as the experimental object. 100, 1000, 10000, 50000, 100000 small files, each size is not more than 1~1000KB, were selected as storage data respectively. Data stored by Oracle and Hadoop were stored successively, and the average value of three times comparison experiment was obtained. The results of the experiment are shown in Figure 8 below.

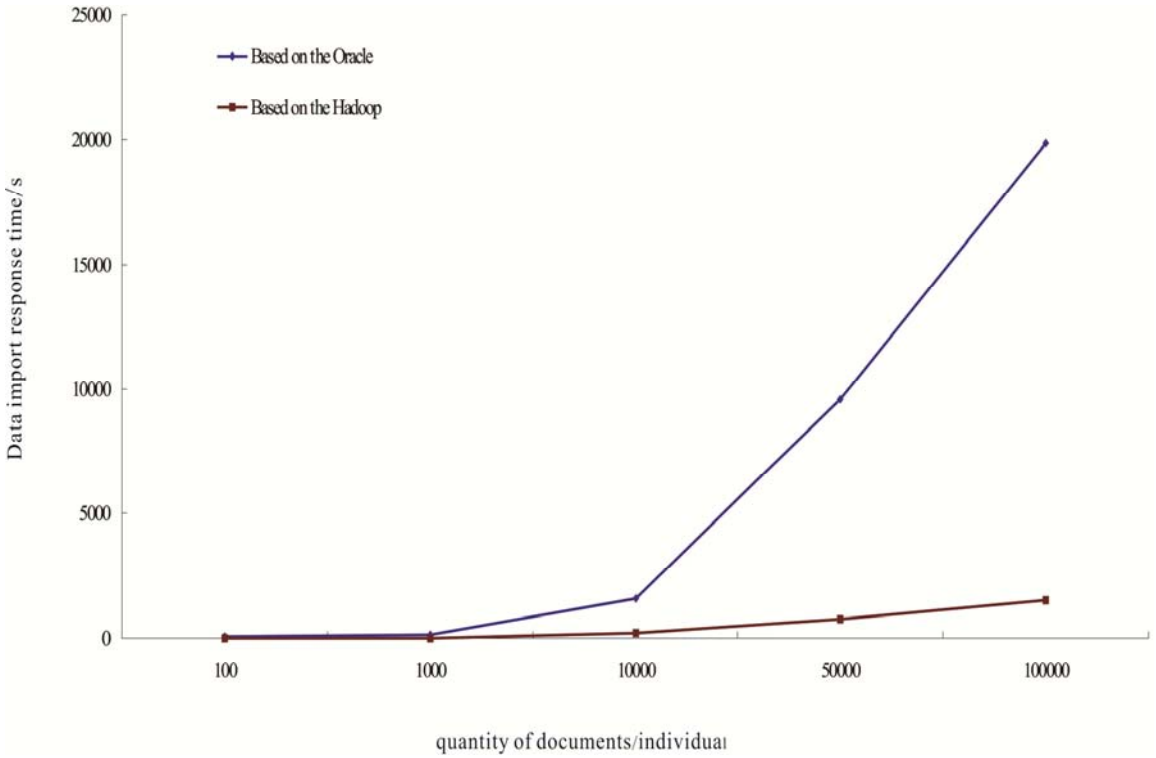

Figure 8. *Data Import Experimental Results.*

As we can see from figure 8, when the number of files is small, the efficiency of the two storage methods is flat, and the improved Hadoop storage method is slightly better. However, when the amount of data is small, compared with the traditional Hadoop memory, Oracle efficiency may be higher. This is because the use of Hadoop for data storage involves selecting metadata storage and DataNode NameNode, time consuming, and we optimized Hadoop only need to call a small amount of NameNode, greatly saves the system overhead time. When the amount of data is large, especially after reaching 10000, Hadoop shows great advantages in geological data storage, and the time needed to write data is greatly shortened. Our traditional Oracle database, data storage mainly focuses on a physical storage node, and grows linearly with the increase of data volume.

3.2.3. Data Export Comparison Experiment

This experiment takes the mineral resources potential evaluation data of Chongqing as the experimental object. The 100, 1000, 10000, 50000, and 100000 small files of 1~1000KB were selected as experimental data. The data stored in Oracle and Hadoop are downloaded respectively, and their respective response times are recorded. The results of the three experiments were compared and the average value was obtained. The results of the experiment are shown in Figure 9 below.

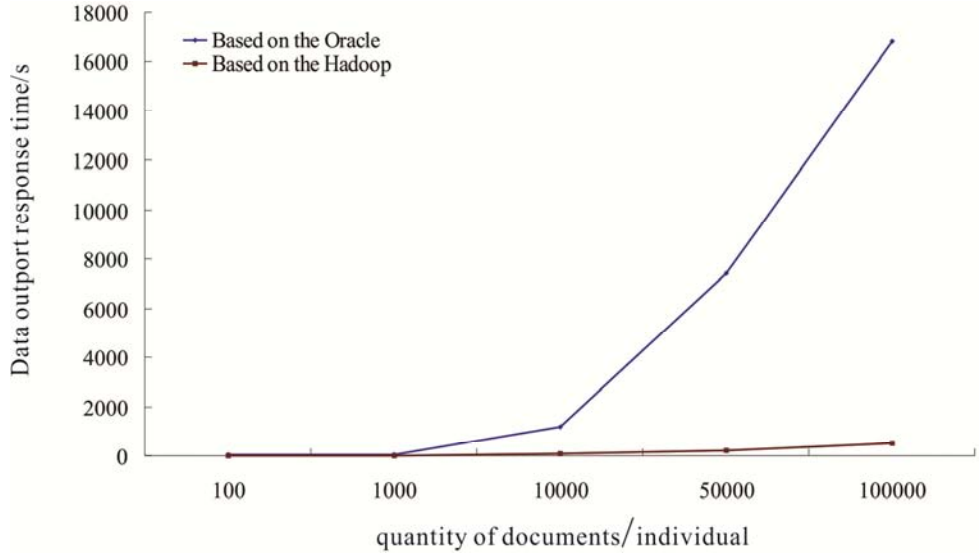

Figure 9. *Data Export Experimental Results.*

Two ways are compared. From the results of Figure 9, we can see that when data is very small, using Hadoop to read data is not a great advantage, which is only a little better than traditional Oracle. This is because, when the amount of data is small, the cost of Hadoop itself needs more data processing time. However, the improved Hadoop combines multiple small files into a large file, which greatly reduces the amount of data access. As a result, the efficiency is a little higher than the Oracle, which has the advantage of storing small amounts of data. When the amount of data increases significantly, especially when the data volume is over 10000, Oracle's single node data output is more difficult. The response time of the system is generally increasing linearly, while Hadoop's multi node parallel work highlights a greater advantage.

4. Discussion

The era of huge data is coming, and this leads to the development of geological and mineral data. Geological and mineral data has an important guiding significance for urban planning and construction, city safety and so on, how to effectively store and administrate geological and mineral data has become a realistic problem urgently to be solved right now. Aimed at the storage problem of the mass data of geology and mineral resources, in this paper having put forward a new data storage solution, this innovation data storage soluton can provide new kind of technical support for the development of geological industry. In the past, mainly using relational database for geological and mineral data management. But with the continuous accumulation of geological and mineral data and the emergence of massive geological and mineral data, general relational database can not solve the scalability, stability and operation efficiency of data management system when the data scale increases sharply. Further investigation show that the innovative method of distributed storage for huge data of geological and mineral resources based on hadoop proposed in this paper are very usfull and practical for our country geological and mineral data management.

5. Conclusion

In order to meet the practical needs of data storage solution of geological and mineral data, a new method of distributed storage for huge data of geological and mineral resources based on hadoop is proposed in this paper, and a series of experiments such as data import test and data export test were carried out to verify the method. Experiments results show:

(1) HDFS is not suitable for mass geological small file storage and the optimized HDFS data storage has significant advantages.
(2) The data storage speed of Hadoop is much faster than that of Oracle which shows the advantage of mass data storage.
(3) The article proposed scheme is effective, for geological data storage and management provides a feasible technical solutions.

Acknowledgements

This paper was jointly funded by the NSFC (No.:41571374) and National Key R&D Program Projects (No.:2017YFB0503802).

References

[1] Zhu, Y. X. and Hou, J. G., 2014. Research on geological data storage and management based on large data// Jiangsu province surveying and mapping geographic information society Proceedings of the 2014 academic annual meeting.

[2] Chen, J. P., Li, J., Cui, N., and Yu, P. P., 2015. The construction and application of geological cloud under the big data background. Geological Bulletin of China, 34(7):1260-1265.

[3] Qian, X., and Zhang, D. Y., 2014. Application of geological exploration in geological prospecting. Luoyang: Journal of Henan Science and Technology, 2014(4):56-56.

[4] Zhang, B., 2009. Based on ArcGIS Fugu county geological disaster database established and the evaluation of vulnerable areas research. Xi'an Chang'an University.

[5] Pu, K., 2009. Design and implementation of multi-source geological spatial database storage management system. Chengdu: University of Electronic Science and Technology of China.

[6] Liu, C. J., 2011. The research and development of the comprehensive geological database management system. Changsha: Central South University.

[7] Zhai, Y. D., 2011. The reliability of the Hadoop distributed file system (HDFS) research and optimization. Wuhan: Huazhong University of Science and Technology.

[8] Li, J., Chen, J. P. and Wang, X., 2015. A study of the storage technology of geological big data. Geological Bulletin of China, 2015, 34(8): 1589-1594.

[9] Hao, S. K., 2012. Brief Analysis of the Architecture of Hadoop HDFS and MapReduce. Designing Techniques of Posts and Telecommuni, 2012(7):37-42.

[10] Chen, K. and Zheng, W. M., 2009. Cloud computing: System instances and current research. Journal of Software, 20(5):1337-1348.

[11] Armbrust, M., Fox, A., Griffith, R., et al. 2010. A view of cloud computing. Communications of the ACM, 53(4):50-58.

[12] Ma, H. T., 2015. Based on the nested HBase data storage system design and implementation. Hangzhou: Zhejiang University.

[13] Li, X. A., 2005. Future development of the database technology. Journal of Shangdong College of Electric Power, 8(2):40-43.

[14] Cui, J. S., 2014. The Present Situation and Tendency in the Integration of Remote Sensing and Geographic Information System. TopCapital, 2014(10):247-247.

[15] Li, C. K., Yan, W. Y. and Xiao, K. Y. 2015. The construction of geological database with MapGIS and Oracle. Geological Bulletin of China, 34(7):1359-1364.

[16] Shvachko, K., Kuang, H. and Radia, S., 2010. The hadoop distributed filesystem. In Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium. 2010:1-10.

[17] Li, M., Cao, S. and Qin, Z. G., 2016. Storage Optimization Method of Small Files Based on Hadoop. Chengdu: Journal of University of Electronic Science and Technology of China, 2016(1):141-145.

[18] Henzinger, T., Jhala, R., Majumdar, R., et al. 2002. Lazy Abstraction/ /POPL'02 Proc. of the 29th ACM SIGPLAN-SIGACT symposium onPrinciples of programming languages, New York, NY, USA: ACM Press.

[19] Godefroid P., 1997. Model Checking for Programming Languages using VeriSoft//Proc. of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, NY, USA: ACM Press.

[20] Chang, F., Dean, J., Ghemawat, S., et al. 2008. Bigtable: A Distributed Storage System for Structured Data. Acm Transactions on Computer Systems, 26(2):205-218.