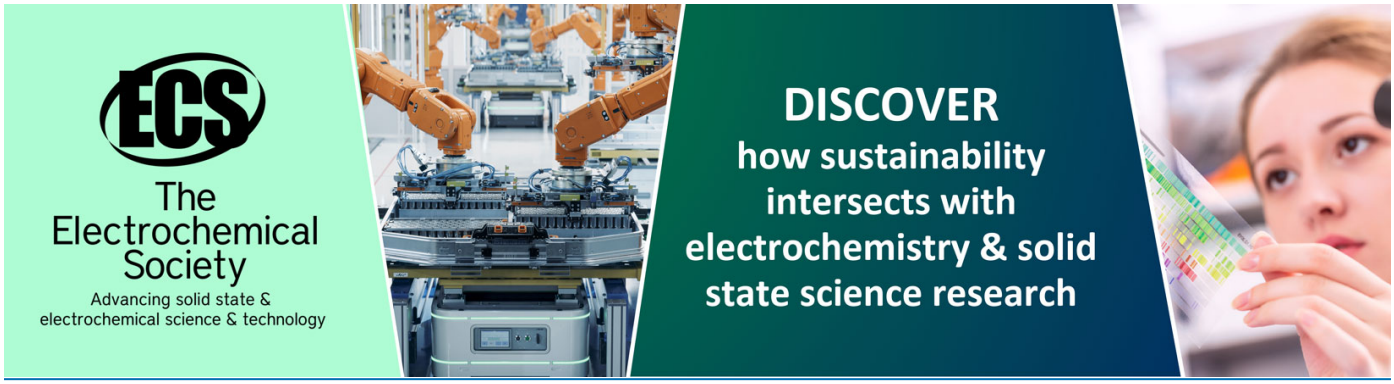
Research and Implementation of Geography Information Query System Based on HBase



Research and Implementation of Geography Information Query System Based on HBase

Lin Qian [1, a], Jun Yu [1, b], Guangxin Zhu [1, c], Hengmao Pang [1, d], Feng Mei [2, e], Wenda Lu [2, f], Haiyang Chen [1, g], Mingjie Xu [1, h], Lin Wang [1, i], Zhu Mei [1, j]

[1]State Grid Electric Power Research Institute, Nanjing, China
[2]State Grid Zhejiang Electric Power Company Information & Telecommunication Branch, Hangzhou, China

[a] qianlin@sgepri.sgcc.com.cn, [b] yujun@sgepri.sgcc.com.cn,
[c] zhuguangxin@sgepri.sgcc.com.cn, [d] panghengmao@sgepri.sgcc.com.cn,
[e] meifeng@zj.sgcc.com.cn, [f] luwenda@zj.sgcc.com.cn,
[g] chenhaiyang@sgepri.sgcc.com.cn, [h] xumingjie@sgepri.sgcc.com.cn,
[i] wanglin18@sgepri.sgcc.com.cn, [j] meizhu2016@aliyun.com

Abstract. With the rise of cloud computing research, the advantages of cloud computing and storage resources are discovered continuously: distributed computing, massive, dynamic and so on. Consequently, more and more application systems begin to migrate to the cloud platform. However, the difference between cloud platform and traditional single or multiple sever model bring certain challenges to the system development. In this thesis, a Geography information query system based on HBase was studied and implemented rellied on National Geography Public Welfare Project. The query system can make users to retrieve the information of sea wind and satellite images by the graphical interface. Through the effective application of query technology, the system can efficiently extract the concerned Geography data information for the fishery production and Geography disaster prevention. This thesis firstly displayed the Geography information query system with the running examples, and secondly tested and analized the HBase query optimization technology to vertify its great expandability and usability. Finally, the performance test of Geography information query system indicated that the system had high scalability and high reliability in the cloud environment.

Keywords: HBase; Cloud Computing; Geography Information; Spatial Temporal Data.

1. Introduction

In recent years, China's National Geographic Bureau has been committed to the construction of "digital geography" [1]. China is a large geographic country, with 3 million square kilometers of jurisdictional geographical territory [2]. In order to further develop the convenience of sharing information resources brought about by digital geography, the National Geographic Bureau undertook the National Geographic Public Welfare Project of "Cloud Computing of Geographic Environment Information [3] and Framework of Cloud Service System", in which Northeast University is responsible for "the technical

research and construction of cloud computing platform". The main research contents of this sub-task include: the division and placement strategy of geographical data in cloud environment, data loading and update technology in cloud environment, data query processing and optimization technology in cloud environment, and monitoring technology of cloud computing platform. In addition, according to the status of system task execution, the project also studies the strategy of dynamic allocation of system storage and computing resources, the strategy of energy-saving and efficient data distribution equilibrium, the task description model based on quality of service (QoS) and the scheduling algorithm based on quality of service, and provides data operation interface, parallel data processing interface and monitoring interface for upper application. Geographic information data processing is undoubtedly a large amount of data processing, the use of cloud computing platform for large data processing capabilities, can effectively solve the data processing problem [4-6]. This paper is based on cloud computing platform, and takes "the technology research and construction of cloud computing platform" as the research topic, including HDFS image data storage, data storage and query, and finally improves query efficiency by improving query algorithm.

2. Related Technology

2.1. Hadoop File System

HDFS (Hadoop Distributed File System [7]) is the main distributed storage mode used by Hadoop [8] applications. HDFS is suitable for applications that process large amounts of data and can be deployed on low-cost hardware. HDFS frees up some POSIX requirements to achieve streaming access to file data.

2.2. Zookeeper: A Reliable Coordination System for Distributed Systems

Zookeepe is a distributed collaborative service for distributed applications. Its motivation is to reduce the burden of redeveloping collaborative services for distributed application [9]. Zookeeper is a distributed application based on Hadoop's coordinated distribution. Partial Failure may occur to a general extent, and Zookeeper can be used to solve this problem. Partial failure here refers to the interruption of the network when information is transmitted between two Zookeeper services, and the sender does not know whether the transmission is successful or not. Zookeeper's solution to this problem is to make the uncontrollable problem controllable.

3. Prepare Design of the Overall Framework of Geographic Information Query and Processing System

3.1. Design Idea

As a part of the application based on cloud computing platform, query system should not only satisfy users'needs, but also minimize the system resources occupied by its runtime. This requires a trade-off between computational speed and system overhead when designing a system. Therefore, the following principles should be paid attention to in the design:

Reduce some unnecessary functions in order to reduce the overhead of the system, and make the user interface of the query system as concise as possible and convenient for users to use under the condition of guaranteeing to meet the needs of users.

This query system has high availability. When the cloud platform increases or decreases the number of nodes, the query system of the cloud platform can quickly resume normal use without affecting the normal operation of the whole system.

The query system should be portable and adaptable to different operating environments. The query system can run on cloud platforms built by different organizations and enterprises.

The query system should have good scalability. When a new node joins the cloud platform, it can dynamically expand the function without modifying the system code; or when the number of query items increases, it can add a module to the original system to realize the new query function.

The query system should be real-time, and the system can return the data needed by the user within a certain period of time according to the conditions put forward by the user.

The system should simplify the deployment and configuration steps as much as possible so that users can use it more easily.

3.2. System Framework

This system adopts a traditional C/S architecture. Data transmission between modules and between client and server is shown in Fig.1. When the client starts, it can present an operation interface to the user. In this interface, MODIS [10] data operation part and meteorological data operation part are included. The data operation part of MODIS includes the functions of querying data according to time and space restrictions, displaying query results, user's operation of query results, and uploading local data to server. The operation part of meteorological data mainly includes the functions of querying data according to base station name and geographical location, uploading local data to server and so on. The server side is responsible for data storage, in which MODIS data is stored in two parts, image description information is stored in HBase [11], and MODIS image is stored in HDFS.

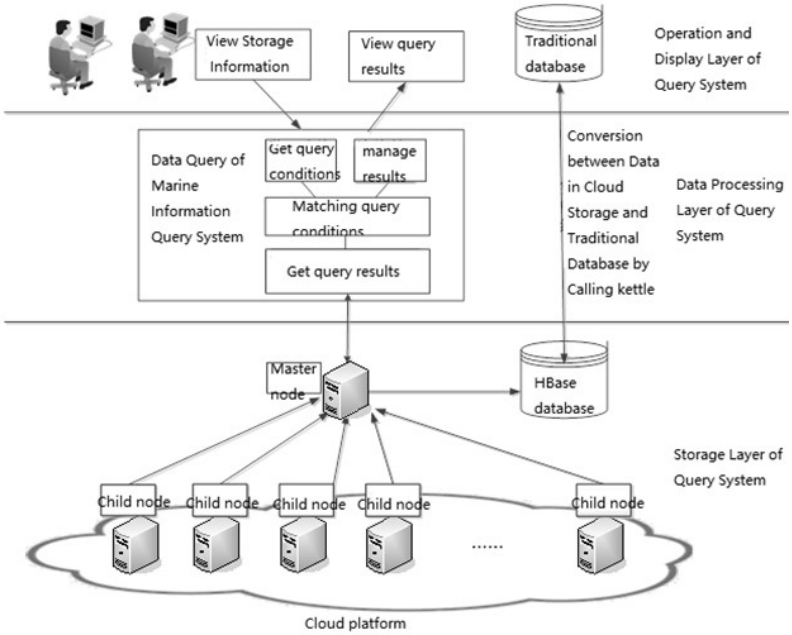


Figure 1. Query system srchitecture diagram

3.3. Design and Implementation of Data Storage Layer

At this level, the main task is to complete data storage and management. Hadoop's HDFS and HBase are used to manage data. And the interface for query access for the upper layer. And the storage of MODIS data and Geo-Meteorological data will be introduced. The overall implementation process is shown in Fig.2.

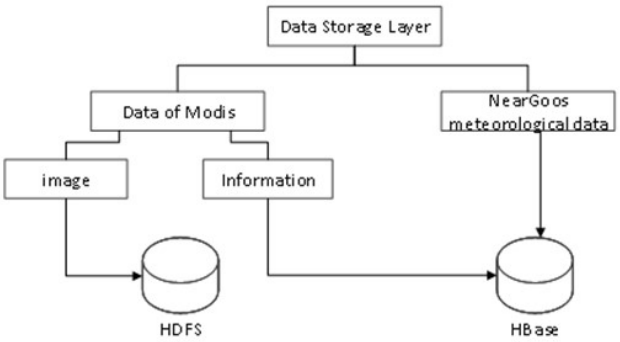


Figure 2. Storage layer diagram

For example, MODIS is an important sensor on terra and aqua satellites. It is the only satellite-borne instrument that broadcasts real-time observation data directly to the world through x-band, and can receive data free of charge and for free use. Many countries and regions around the world are receiving and using MODIS data. MODIS data mainly includes two parts: one is the image information collected. As shown in Fig.3, the other part is text data describing image attribute information.

Figure 3. MODIS data sample

The samples of text information are shown in Table 1:

Table 1. MODIS data

Data name	Date	Spatial resolution	SceneID	Description	File name
Data	20040401	250/500/1000	355	Less cloudiness in the Bohai Sea, Yellow Sea and South China Sea	20040401.355.jpg
Data name	**Begin time**	**End time**	**Coverage area**		
Data	3:55:52	4:07:26	91.67050934, 127.62878418, 108.31600952, 87.07913208 57.51202393,52.31939316,13.10152149,16.19052124		

Attribute information data includes: date, scene number, start time, end time, spatial resolution (m), fast vision, longitude, latitude, description, image name.

1) Call function of getting through a single rowkey.
2) Call function of scanning by specifying Rowkey range.
3) Perform a full table scan to obtain the desired data.

Table 2. MODIS data table structure

Column family name	Time	Describe
Column name	Time_date	Describe_jinghao
	Time_begin	Describe_shikongfenbianlv
	Time_end	Describe_kuaishi
		Describe_fugaifanwei
		Describe_describe
		Describe_name

4. Design and Application of Algorithms

This chapter mainly analyses the solutions to the problems encountered in the implementation of the system. Firstly, the characteristics of geographic information and HBase itself are analyzed. A space filling algorithm based on B-order value is proposed to reduce the dimension of two-dimensional space, so that HBase can effectively reduce the search range of query, and then improve the query efficiency.

Then the ray method is used to judge the position relationship between the spatial query condition and the MODIS image, and the spatial query of MODIS data is completed.

HBase only supports three kinds of queries in the process of query: (1) full table scanning; (2) range scanning for keywords; (3) single reading according to keywords. The spatial information in the EarGoos meteorological data is a typical two-dimensional data. In order to improve the search speed of the system in NearGoos space, it is necessary to effectively add spatial information to rowkey. Rowkey is a string which is one-dimensional data, so here we need to find an effective dimension reduction method to make the existing two-dimensional data can be effectively converted into one-dimensional data, and is conducive to adding rowkey to the HBase data table. The two-dimensional space is divided effectively, and then each data block is coded, using this code to represent the spatial information of the data block.

4.1. Spatial Keyword Query Algorithm Based on B-order Value

In order to improve the efficiency of spatial keyword query as much as possible (focus on the spatial information of NearGoos meteorological data), we divide the space into several regions, and use a value to represent all data in a region. The basic idea is shown in Fig.4, which is named B-order naming rule.

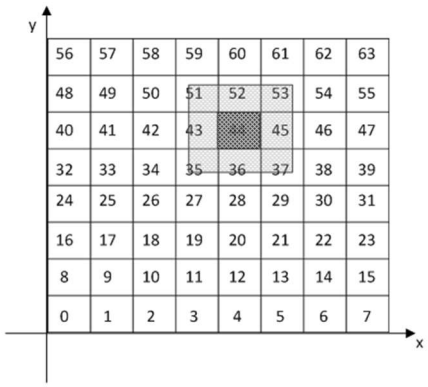


Figure 4. Spatial partition

In the application, spatial information is two-dimensional and bounded, which means the whole two-dimensional space is divided into N data blocks. The data in each data block is the data adjacent to each other or within the same range. The spatial information of the whole data block is represented by the same one-dimensional string. B-order aggregates adjacent or identical data objects in space by assigning the same B-order value to objects with similar spatial coordinate values.

For the two-dimensional spatial coordinates of NearGoos meteorological data object, we use this situation to reduce the dimension of coding, and the B-order value which is hindered by coding as the spatial information part of rowkey. In practice, the value of longitude and latitude is limited, so a string can be composed of 100 bits of longitude and 10 bits of longitude and 10 bits of latitude to represent all data falling within the range of 10*10 squares (when the amount of data is large, the granularity of data partition can be adjusted according to the actual situation to achieve the best effect). For example, if the spatial information of a data is 110.333 degrees longitude and 28.455 degrees latitude, then according to the rules, the data falls into the data block with the B-order value of 112 (in this case, if there is a lack of 100 or 10 bits in longitude and latitude, use 0 to occupy space, as in the form of 001). The generating function of Rowkey is "rowkey = B-order value + keyword". In this way, HBase can store data objects with similar space as much as possible. When querying, it can effectively use spatial information to filter data, reduce the scope of queries, and improve the efficiency of queries. As shown in Fig.4, the gray box is the scope of query. In the actual query, only nine areas, 35, 36, 37, 43, 44, 45, 51, 52 and 53, are scanned and judged. Among them, the area 44 is the area that completely meets the requirements without spatial condition judgement, and the other eight areas need conditional judgement.

Algorithm 1. Query algorithm B-order

Input: TableName: Data tables where data resides; Space: spatial constraints
Output: null
BEGIN
1. IF (The data part of block meets the requirement.)
2. SomeRowkey. add (B-order value) (SomeRowkey is a list here.);
3. IF (The data part of block meets the requirement.)
4. AllRowkey. add (B-order value)(AllRowkey is a list here.);
5. For(Take out one by one in SomeRowkey B-order value)
6. { Call the scan method of HBase to set the query range with B-order value;
7. Screening query results by using spatial constraints entered by users;}
8. For(Take out one by one in AllRowkey B-order)
9. { Call the scan method of HBase to set the query range with B-order value;
10. Store the results in ResultR;}
11. return(ResultR)
END

We focus on spatial keyword query: for a spatial keyword query Q (TableName, Space), where TableName is the data table where the data object is to be queried, Space is the space restriction condition entered by the user, and the return result is all the data objects included in the space restriction condition.For such queries, the proposed B-order-based spatial keyword query algorithm is divided into the following three steps to complete:

Step 1: The system determines the range of keywords that need to be queried according to user input conditions and B-order value generation rules.

Step 2: Select all query keywords and classify them into two types. One is that part of the data contained in the data block is qualified, the other is that the data contained in the data block is qualified.

Step 3: Use the keyword set to determine the data range scanned by Scan method of HBase and get the final result.

4.2. Optimal Processing of Temporal and Spatial Queries

In this section, we will discuss how to improve the efficiency of time and space query. In order to effectively improve the efficiency of query in HBase, we need to consider how to reduce the amount of data scanned. To reduce the scope of scanned data tables, this article should make full use of rowkey, or build an additional index table. Near Goos meteorological data can be searched in time and space by using the coding of rowkey. The range of rowkey can be obtained accurately according to the input conditions, avoiding scanning a large number of useless data, so as to improve efficiency. The rowkey encoding rule "rowkey = B-order value + time information + random number" is used in the implementation, so that multiple consecutive rowkeys can be obtained according to the time and space conditions input by users, thus achieving relatively accurate scanning of data.

5. Experiments and Performance Analysis

5.1. Configuration of Experimental Environment

Due to the limitation of experimental conditions, this paper deploys the Geographic Information Query and Control System on a Hadoop cluster composed of nine PCs. The specific software and hardware environment of these nodes is shown in Table 3.

Table 3. Experimental Environment

Software and hardware environment	
CPU(/unit)	Intel(R) Core(TM)2 6300(1.86GHz)
Memory (/unit)	4GB
Hard disk (/unit)	1TB,7200rpm
operating system	CentOS 5.6
Programing language	Java
Switch	TP-Link 1Gbit/s

The specific deployment steps of Geographic Information Query System are as follows:

The basic configuration of the cluster: (1) the basic configuration of Hadoop cluster. Hadoop uses Cloudera release version, version 0.20.2, Hadoop block size 64 MB, and Hadoop HDFS backup number 3. (2) The basic configuration of HBase, HBase version 0.90.4, Zookeeper number 3 (in practice should be set to odd), version 3.3.4, RegionServer number 9. (3) Basic configuration of Hive, version 0.7.1.

Installing the above software on 9 nodes separately constitutes the bottom platform of the system. At runtime, the geographic information query system runs in the main node.

This experiment needs a lot of geographic information data, so most of the data in the experiment are copied from the real data of the geographic information center. But in the process of doing the experiment, we found that the original data is not enough. We generated simulation data based on the real data, and then tested the performance of the system.

5.2. Testing and Performance of B-order Value Spatial Query Technology

In this section, in order to test the effect of B-order value spatial query technology on query efficiency, the B-order value spatial query technology and the B-order spatial query technology are used to do the test experiments respectively.

Next, we use NearGoos meteorological data to test the B-order value spatial query technology. The data used in the experiment were 1 million, 2 million, 3 million, 4 million and 5 million, respectively. In this experiment, the partitioning granularity of two-dimensional space is 10*10, that is to say, the longitude of 100 bits and 10 bits and latitude of 10 bits are selected as spatial record information and added to rowkey. Spatial query conditions are 120 to 130 degrees longitude and 70 to 78 degrees latitude. The query experiments were conducted in the case of not adding spatial information to rowkey and in the case of adding spatial information to rowkey.

The relationship between the time consumed to record queries and the total amount of data is shown in Fig.5. Through the observation of image data, it can be clearly found that the query efficiency of rowkey has been significantly improved after using B-order value space filling technology to optimize rowkey. By observing the curve before optimization, we find that the time spent on querying is basically linear with the total amount of data, and there is a slight fluctuation. In Fig.5, because of the small query time after B-order value space filling technology, the curve of B-order value space filling technology is close to the coordinate axis, and the relationship between query time and quantity can not be found by curve trend. Fig.6 is made after log10 is taken from the query result. By observing the optimized curve before and after optimization, it is found that the query efficiency has been improved by two orders of magnitude.

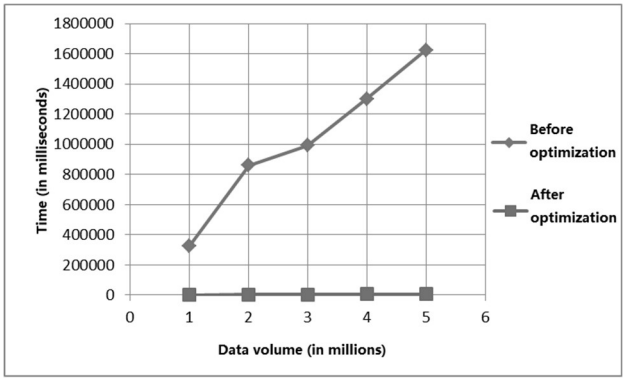


Figure 5. Relation between query time and total data

The reason for this result is that the whole data table needs to be scanned and judged before using B-order space filling technology, while after using B-order space filling technology, only the data labeled 127 can be scanned and judged. Therefore, the query time has been significantly reduced.

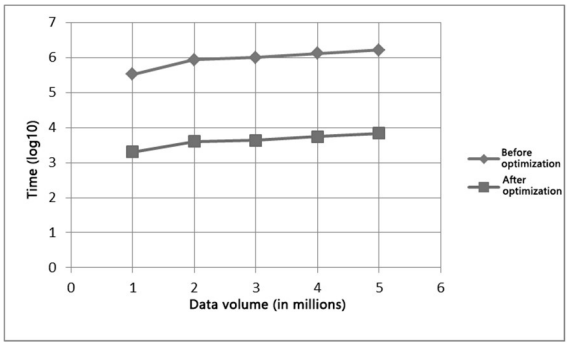


Figure 6. Relation between query time and total data

In order to better understand the relationship between query time and data volume after adopting B-order value space filling technology, further tests and analyses are carried out on the condition that the data in the data block after space partitioning fully and partially meet the requirements.

The data used in the experiment are NearGoos meteorological data. Query condition 1, longitude range 120 degrees to 125 degrees, latitude range 70 degrees to 78 degrees; query condition 2, longitude range 120 degrees to 130 degrees, latitude range 70 degrees to 80 degrees.

Record the relationship between query results and total data as shown in Fig.7. Through the observation of the image, it is found that under the condition of query condition 1, the time increases linearly with the amount of data. In the case of query condition 2, the query time does not change significantly with the increase of data volume, so it is not difficult to find that it basically maintains a straight line.

The reason for the above experimental results is that in the case of query condition 1, the data contained in block 127 does not satisfy all the conditions, and the data in the block need to be judged one by one. In the case of query condition 2, the data labeled 127 are all qualified data, so long as all rowkeys in the data block can be obtained, the data in the data block need not be filtered one by one, so the query time does not increase significantly with the increase of data volume.

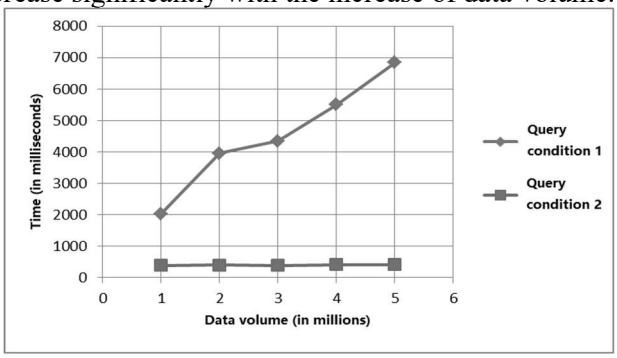


Figure 7. The relationship between query time and query number after improvement

5.3. Performance Testing of Spatial and Temporal Queries

When the system is in use, sometimes it is not only to deal with the user's limited conditions in space, but also to meet the user's limited conditions such as time. In this section, we will further optimize rowkey in HBase table to add time information on the basis of spatial information. The overall structure of rowkey is rowkey = B-order value + time information + random number.

Next, we use NearGoos meteorological data to test and record the query time and data volume before and after adding time information to rowkey. Spatial query conditions are longitude 110 to 120 degrees, latitude 30 to 40 degrees, and time limit conditions are 2003 2001 to 31212 (i.e., starting time is January 1, 2003, ending time is December 12, 2003). The relationship between query time and total data is shown in Fig.8.

By observing the data results, we can find that the query time of rowkey maintains on a straight line after adding the time attribute, and does not change with the increase of data volume. In the absence of time information added to rowkey, although the data block in which the data is located is known according to the spatial constraints, the data in the data block does not meet all the conditions. At this time, the data in the data block need to be filtered one by one according to the time limit again. At this time, as the amount of data increases, the number of judgments that need to be made increases, and the time spent on queries also increases.

The reason is that the starting and ending rowkeys can be precisely set up from the query condition 1 in the scan method of HBase, and the data in this range are all satisfied, so long as the keywords (rowkeys) are obtained, the data can not be scanned. Therefore, the query time does not change significantly with the increase of data volume. In the case of query condition 2, the scope space is determined, but the data information should be judged. With the increasing amount of data, the number of judgments is also increasing, and the query time spent increases naturally.

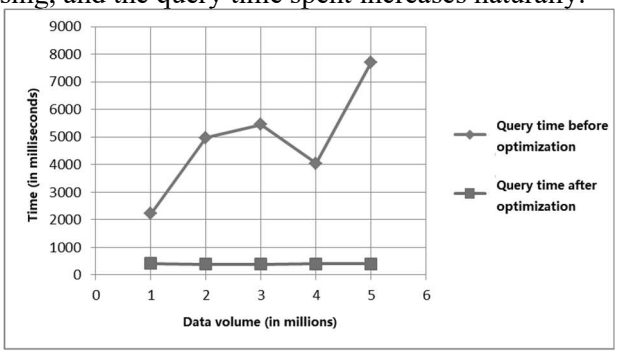


Figure 8. Comparison of query efficiency

6. Conclusion

In recent years, with the continuous development of detection technology, increasing investment and other factors, the type and scale of geographic information data is growing at an unprecedented rate. The large-scale effect of large data brings great challenges to data storage, management and data analysis. On this basis, this paper analyses the basic framework of mainstream big data processing and the effect of cloud computing technology on data management in the era of big data. According to the characteristics of basic geographic information, a geographic information query system based on HBase is designed and implemented, and a B-order value space filling algorithm is proposed to optimize the query processing of the system.

Firstly, a geographic information query system based on HBase is designed and implemented. Then, the B-order value space filling algorithm is proposed to optimize the query processing. The method of B-order value is used to partition the data in two-dimensional space, and the B-order value is added to rowkey to improve the query efficiency. Finally, the geographic information query system is deployed on a real cluster, and the running process and results of the system are shown with specific data. The B-order spatial filling algorithm used in the geographic information query system is tested and analyzed. The results show that the algorithm can effectively improve the query efficiency and reduce the space query time by two orders of magnitude compared with the original time.

Acknowledgments

This work was financially supported by the science and technology projects of headquarters of State Grid Corporation (Research on key technologies of secure network coding and data protection for secure renewable storage System).

References

[1] Wengfeng, H. Tentative ideas on the development of "Digital Ocean" in China [J]. Marine Science Bulletin, 2000, 12(1):25-37.
[2] Wang Shicheng. Geographical Strategy for the 21st Century (II) [J], Qilu Fisheries, 1997, 14 (6): 1-4.
[3] Chen k, Zheng WM. Cloud computing: System Instances and Current Research [J], Journal of Software, 2009: 1337-1348.
[4] Liu Peng. Cloud Computing (2nd Edition) [M], Beijing: Electronic Industry Press, 2011, 38(4):32-37.
[5] Li Qiao, ZHENG Xiao. Research Survey of Cloud Computing [J]. Computer Science, 2000, 12(1):25-37.
[6] Xingwang Z, Chenhui L , Xiaozhu Q . Research and Initial Implementation of Large-scale Data Processing Based on Cloud Computing [J]. New Technology of Library & Information Service, 2011.
[7] Niemenmaa M, Kallio A, Schumacher A, et al. Hadoop-BAM: directly manipulating next generation sequencing data in the cloud [J]. Bioinformatics, 2012, 28(6): 876-877.
[8] Hadoop [EB/OL],http://developer.teradata.com/extensibility/articles/Hadoop-dfs-to-terad ata, Teradata Dev- eloper Exchange, 2010. (The date of the visit is December 2012)
[9] Hunt P, Konar M, Junqueira F P, et al. ZooKeeper: wait-free coordination for internet-scale systems [A], Proceedings of the 2010 USENIX conference on USENIX annual technical conference[C], 2010, 8: 11-12.
[10] MODIS [EB/OL], http://zh.wikipedia.org/wiki/, 2013. (The date of the visit is January 2013)
[11] HBase [EB/OL], http://baike.baidu.com/view/1993870.htm, 2013. (The date of the visit is December 2012)