Research and Implementation of Geography Information Query System Based on Improved B-Order Algorithm

Lin Qian\textsuperscript{1,a}, Lin Wang\textsuperscript{1,b}, Jun Yu\textsuperscript{1,c}, Guangxin Zhu\textsuperscript{1,d}, Yinghui Pei\textsuperscript{1,e}, Xin Hu\textsuperscript{2,f}, Zhu Mei\textsuperscript{1,g}, Hengmao Pang\textsuperscript{1,h}, Mingjie Xu\textsuperscript{1,i} and Haiyang Chen\textsuperscript{1,j}

\textsuperscript{1}State Grid Electric Power Research Institute (NARI Group Corporation), Nanjing, China
\textsuperscript{2}State Grid Information&Telecommunication Company of Shanxi Electric Power Corporation, Xi’an, China
\textsuperscript{a}qianlin@sgepri.sgcc.com.cn, \textsuperscript{b}wanglin18@sgepri.sgcc.com.cn, \textsuperscript{c}yujun@sgepri.sgcc.com.cn, \textsuperscript{d}zhuguangxin@sgepri.sgcc.com.cn, \textsuperscript{e}Peiyinghui@163.com, \textsuperscript{f}1484997468@qq.com, \textsuperscript{g}meizhu2016@aliyun.com, \textsuperscript{h}panghengmao@sgepri.sgcc.com.cn, \textsuperscript{i}xumingjie@sgepri.sgcc.com.cn, \textsuperscript{j}chenhaiyang@sgepri.sgcc.com.cn

Abstract. With the rise of cloud computing research, 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 relied 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. In the implementation process of Geography information query system, this thesis studied and optimized the query technology. Firstly, in the query process of NearGoos data, B-order was used to divide the two-dimensional space and combine the sort features of rowkey in the table of HBase to optimize the spatial query. By improving the B-order algorithm, this algorithm made the query efficiency increase two orders of magnitude than before.

1. Introduction
The 21st century is a century of striving for maritime rights and interests. China is a big geographic country with 3 million square kilometers of jurisdictional geographical territory [1]. In recent years, China has attached great importance to geographic economy. The National Geographic Bureau of China has been committed to the construction of "digital geography" [2]. Geographic information data processing is undoubtedly a large amount of data processing. Using cloud computing platform to process large data can effectively solve the data processing problem[3-6].In order to further develop the convenience of sharing information resources brought by digital geography, the National Geographic Bureau undertook the National Geographic Public Welfare Project of "Cloud Computing of Geographic Environment Information[7] and Framework of Cloud Service System", in which Northeast University is responsible for "Research and Construction of Cloud Computing Platform Technology". The main research contents of this sub-task include: the division and placement strategy of geographic data in...
cloud environment, data loading and update technology in cloud environment, data query processing and optimization technology in cloud environment, monitoring technology of cloud computing platform. In the research and implementation of geographic information query based on HBase [8], the author proposes a B-order spatial keyword query algorithm to improve query efficiency. But in operation, when the spatial query scope is large or the partition granularity of space is small, there will be several data block codes that are continuous, but in actual operation, the above is not well utilized the contiguous data blocks. In this way, these continuous data blocks are not fully utilized and many times of system overhead is wasted. This paper is based on the sub-task of "cloud computing platform technology research and construction", through improving B-order algorithm, cloud computing as the platform, data storage and query as the research content.

2. Related technology

2.1. HDFS
Hadoop file system, (Hadoop Distributed File System, HDFS [9, 10]) is the main distributed storage mode used by Hadoop applications. HDFS is suitable for applications that process large amounts of data. HDFS is highly fault-tolerant and can be deployed on low-cost hardware.

2.2. Reliable Coordination System for Distributed Systems Zookeeper
Zookeeper is a distributed collaborative service for distributed applications. Its motivation is to reduce the burden of redeveloping collaborative services for each distributed application development. 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 idea to solve this problem is to make the uncontrollable problem controllable [11].

3. System design

3.1. Design Idea
(1) 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. Each kind of geographic data information contains a considerable number of data items. It is not necessary to do mathematical analysis for each attribute. If we deal with all data items more, it will appear that the user interface is unusually messy and difficult for users to operate.

(2) This query system should have high availability. As a query system based on cloud platform, it can quickly resume normal use when adding or reducing nodes on cloud platform without affecting the normal operation of the whole system.

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

(4) This query system should have good scalability. When new nodes join the cloud platform, they can dynamically expand the functions without modifying the system code or when the number of query items increases, they can add a module to the original system to realize the new query function.

(5) 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.

(6) 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 more traditional C/S architecture. Data transmission between modules and between client and server is shown in Figure 1. When the client starts, it can present an operation interface to the user. In this interface, MODIS [12] 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, and MODIS image is stored in HDFS.

![Figure 1. Query System Architecture Diagram](image.png)

3.3. Design and Implementation of Data Storage Layer
In this level, the main task is to complete the storage and management of data. This section will introduce the storage of MODIS data and Geo-Meteorological data. The overall implementation process is shown in Figure 2.

(1) Design and Implementation of Data Storage
MODIS data mainly consists of two parts: one part is the collected image information as shown in Figure 3, the other part is the text data describing the image attribute information. Samples of text information are shown in Table 1:

| Data name | Date     | Spatial resolution | Description                                           | File name            |
|-----------|----------|--------------------|-------------------------------------------------------|----------------------|
| Data      | 20040401 | 250/500/1000       | Less cloudiness in the Bohai Sea, Yellow Sea and South China Sea | 20040401.355.jpg    |

| Data name | SceneID | Begin time | End time | Coverage area                                                                 |
|-----------|---------|------------|----------|-------------------------------------------------------------------------------|
| Data      | 355     | 3: 55: 52  | 4: 07: 26| 91.67050934, 127.62878418, 108.31600952, 87.07913208, 57.51202393, 52.31939316, 13.10152149, 16.19052124 |
Figure 2. Storage Layer Diagram

![Storage Layer Diagram](image)

Figure 3. MODIS Data Sample

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

1) Calling get functions through a single rowkey;
2) Call the scan function by specifying the Rowkey range;
3) Perform a full table scan to get the desired data.

Table 2. MODIS data table structure

| Column family name | Time | Describe |
|-------------------|------|----------|
| Column name       |      |          |
| Time              |      | 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 are analyzed, and the space filling algorithm of B-order value is improved 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; secondly, the ray method is used to judge the location relationship between spatial query conditions and MODIS images, and the space querying of MODIS data is completed.

4.1. Spatial Query Optimization

In this paper, 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. In this section, we need to find an effective dimension reduction
method to make the existing two-dimensional data can be effectively transformed into one-dimensional data, which is conducive to adding rowkey to the HBase data table, and optimize the B-order algorithm to improve query efficiency. In this paper, 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.

(1) Spatial Filling Curve

The characteristic of this kind of index structure is that we hope to find a way to sort the data in multi-dimensional space approximately, so that the data that are close to each other in space can be ranked together with a higher probability. Then they can be indexed with one-dimensional data. This method can achieve good results in point query operation, but it will be more troublesome in range query. According to this idea, several methods of mapping and sorting point data in multi-dimensional space into one-dimensional space are proposed. Four methods are listed in Figure 4. For a detailed description, see the article [13] for a detailed description.

![Figure 4. Common Spatial Filling Curves](image)

All space filling curves have an important advantage that any dimension data can be processed, provided that the projected one-dimensional space keys are arbitrarily large. However, this method also has an obvious disadvantage, when the index of two different regions is combined together, at least one of them should be re-coded.

(2) Rowkey Merging Technology

In the actual operation of the system, it is found that after using the algorithm in the previous section, when the scope of spatial query is large, or when the partition granularity of space is small. There will be several data block encoding is continuous, but in the actual operation, the above does not make good use of these continuous data blocks, but one by one calls to complete the data query. In this way, these continuous data blocks are not fully utilized and many times of system overhead is wasted. The query rowkey generated in the previous section will be further filtered in this section. Scope lookup with continuous coded data blocks as a whole can effectively reduce the number of calls to scan method of HBase and improve query efficiency. In the process of implementation, it draws lessons from the nature of row-wise space filling curve, that is, continuous space is formed into a whole to make a call. The implementation process is shown in Figure 5. The B-order values based on the user's input qualifications are sixteen regions including 26, 27, 28, 29, 34, 35, 36, 37, 42, and 43, 44, 45, 50, 51, 52, 53. When no measures are taken, the scan function of HBase will be invoked 16 times, but this query can be completed eight times after the B-order values are joined together consecutively in this paper, which reduces the number of calls to the query function and improves the efficiency. The improved query algorithm flow is shown in algorithm 2.
Algorithm 2 Improvement of Query Algorithms 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 the B-order value in SomeRowkey one by one)
6.  {
7.    While(The next B-order value is continuous with the current B-order value)
8.      {Read the next B-order value;}
9.    Set scan.setStartRow() and scan.setStopRow() based on the starting and ending B-order values
10.   Call the scan method of HBase to set the query range with B-order value;
11.   Screening query results by using spatial constraints entered by users;}
12. For(Take out the B-order value in AllRowkey one by one)
13.  {
14.    While(The next B-order value is continuous with the current B-order value)
15.      {Read the next B-order value;}
16.    Set scan.setStartRow() and scan.setStopRow() based on the starting and ending B-order values
17.    Call the scan method of HBase to set the query range with B-order value;
18.    Store the results in ResultR;}
19. return(ResultR)
END

Figure 5. The Merge Schematic of Rowkey

5. Experiments and performance analysis
The basic principle of the algorithm is creating a database which conforms to the CIM model.
This chapter combines real data to test the geographic information query system based on HBase.
Firstly, the geographic information query system is deployed on the real node to show the running
example and interface of the system. Secondly, the HBase query optimization technology used in the
geographic information query system is tested and analyzed. Finally, the performance of HDFS in
Hadoop platform is tested and analyzed to verify the scalability of the system.
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.

| 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:
Basic configuration of cluster. The basic configuration of Hadoop cluster. Hadoop uses Cloudera release version, version 0.20.2, Hadoop block size 64 MB, Hadoop HDFS backup number 3. 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. Hive's basic configuration, 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. Performance Test of Rowkey Merging Algorithm
In this section, in order to understand the impact of rowkey merging on query efficiency, we test the time spent on queries before and after optimization.
Next, it is tested using NearGoos meteorological data. The data used in the experiment were 1 million, 2 million, 3 million, 4 million and 5 million, respectively. In this experiment, the partition granularity of two-dimensional space is 10*10, and the spatial query conditions are 115 to 120 degrees longitude and 35 to 65 degrees latitude. According to the spatial query conditions, we can get the data blocks in 113, 114, 115 and 116, respectively. Query experiments were conducted before and after rowkey optimization.
The relationship between the time consumed to record queries and the total amount of data is shown in Figure 6. By observing the image, we can find that the query time keeps a linear relationship with the amount of data in two cases. There are fluctuations when the amount of data is 1 million and 2 million, respectively. The query time after optimization is less than that before optimization.
The reason for the above experimental results is that the most time-consuming part of the query is the comparison operation, and the change of the proportion of the data distributed in the query block causes the fluctuation of the query time. In the image, we can see that the time spent after optimization is less than that before optimization. The reason is that before optimization, we need to call a query function once for each data block. In this experiment, we need to call it four times. After optimization, we only need to call a query function once, which saves three times. And when the amount of data changes, this time is basically unchanged, always maintained in about 2 seconds. Moreover, the smaller the amount of data is, the more obvious the optimization effect is and the larger the proportion of time saving is. Therefore, in practical application, it should be noted that the amount of data contained in each data block should not be too much.
6. Conclusion
In recent years, with the continuous development of detection technology, the types and scales of geographic information data are increasing at an unprecedented rate. This paper analyses the basic framework of mainstream large data processing and the effect of cloud computing technology on data management in the era of large data. According to the characteristics of basic geographic information, a geographic information query system based on HBase is designed and implemented. The B-order value space filling algorithm is improved, and the query processing of the system is optimized. Finally, the running process and results of the system are shown with specific data, and the improved 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. Combining with MODIS image information, the HDFS performance of Hadoop platform used in the system is tested and analyzed. The results show that the more the number of DataNodes is, the higher the upload rate is, which proves that the system has good scalability.

Acknowledgments
This work was financially supported by the science and technology projects of headquarters of State Grid Corporation (Research and application of non-stop state Adaptive upgrade technology for information system)

References
[1] Wang Shicheng. Geographical Strategy for the 21st Century (II) [J], Qilu Fisheries, 1997, 14 (6): 1-4.
[2] Hou WF. Tentative Ideas on the Development of "Digital Ocean" in China [J]. Marine Science Bulletin, 2000, 12 (1): 25-37.
[3] Liu Peng. Cloud Computing (2nd Edition) [M], Beijing: Electronic Industry Press, 2011, 38 (4): 32-37.
[4] Li Qiao, ZHENG Xiao. Research Survey of Cloud Computing [J]. Computer Science, 2000, 12 (1): 25-37.
[5] 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.

[6] CHEN Chongcheng, LIN Jianfeng, WU Xiaozhu. Massive Geo-spatial Data Cloud Storage and Services Based on NoSQL Database Technique [J]. Journal of Geo-Information Science, 2013, 166-174.

[7] Chen K, Zheng WM. Cloud computing: System instances and current research [J]. Journal of Software, 2009: 1337-1348.

[8] Yu Z, You-Zhong M A, Xiao-Feng M. Efficient Processing of Spatial Keyword Queries on Hbase [J]. Journal of Chinese Computer Systems, 2012, 33 (10): 2141-2146.

[9] Liu X, Han J, Zhong Y, et al. Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS[A], International Conference on IEEE [C], 2009: 1-8.

[10] Wei-Long X U, Yan Z, Hong-Liang Z, et al. The Design and Implementation of a Data Backup System based on HDFS [J]. Netinfo Security, 2012.

[11] Rui L I, Gang L, Shuwei Y, et al. Research and Design of ZooKeeper and ACE Based Distributed Computing Framework [J]. Electric Power Information & Communication Technology, 2015.

[12] MODIS [EB/OL], http://zh.wikipedia.org/wiki/, 2013. (The date of the visit is January 2013)

[13] Harbin. Nearest-neighbor Query Algorithm Based on Grid Partition of Space-filling Curve [J]. Computer Science, 2010, 37 (1): 184-188.