Study on Geography Information OLAP and Data Mining System Based On Hadoop

Study on Geography Information OLAP and Data Mining System Based On Hadoop

Jun Yu[a], Hengmao Pang[b], Zhu Mei[c], Debing Song[d], Guangxin Zhu[e], Haiyang Chen[f], Lin Wang[g]

State Grid Electric Power Research Institute (SGEPRI) Nanjing, China

[a]yujun@sgepri.sgcc.com.cn, [b]panghengmao@sgepri.sgcc.com.cn,
[c]meizhu2016@aliyun.com, [d]songdebing@sgepri.sgcc.com.cn,
[e]zhuguangxin@sgepri.sgcc.com.cn, [f]chenhaiyang@sgepri.sgcc.com.cn,
[g]wanglin18@sgepri.sgcc.com.cn

Abstract. The 21st Century is the century of ocean. Every coastal country makes important development strategies which are connected with national geography rights protection, geography economy development and geography ecological environment protection. As a part of “Digital Ocean”, geography information OLAP and data mining can find the geography laws and knowledge hidden in data, which makes great sense to geography environment protection, geography meteorological observation, geography meteorological prediction, geography disaster prevention and geography mitigation.

1. Introduction

The 21st Century is a “Big Data” century with data explosions [1]. With the continuous development of geographic information technology in China, geographic information collection is more and more fast, rich and comprehensive. In particular, since the implementation of the "digital geography" project, a large number of geographic environment information systems have been built, providing a strong technical support and digital support for the development of our country's geography. For these accumulated original geographic data, the valuable information hidden in massive geographic data is analyzed and processed by Online Analytical Processing (OLAP) and data mining [2-8]. The significance of this study is to use the GIS system based on the cloud platform to analyze and excavate the relevant hydrological information in China, and to discover the laws and knowledge contained in the massive geographic data, and to use these discoveries in the geographical environment and our life to protect the ecological environment and the sustainable development of the geographic environment. At the same time, these laws and knowledge are of great significance to the related geographical policy and the development of the coastal city development strategies, the observation and forecast of geo meteorology, disaster prevention and disaster reduction work of geography.

2. System Overview

Geographic information OLAP and data mining system based on Hadoop provide an efficient platform for geographic researchers to deal with massive geographic data. The platform is composed mainly of

two functional modules, which are geographic information OLAP and online data mining tools. Two works can be used in geographical scientific research and complement each other.

According to the analyzing for system requirement of the project, the system can be divided into two parts, one is the online analysis and processing of geographic data, and the other is an online data mining tool for geographic data. As shown in the overall structure of Figure 4, the early work of the project has completed part of the OLAP function of geographic information. In this study, it is mainly involved in the later transformation of the Hive dialect and the addition of the online data mining tools.

The overall design and implementation of the system will also be carried out around the two functions. The storage and related computing processing of the geographic information OLAP and data mining system need to use the new cloud computing technology, in order to store the massive geographic data in the traditional ordinary computer, and to parallel the data of the data. Analysis and processing can achieve low cost and high efficiency in analyzing query performance.

3. Design and Implementation of Geographic Information OLAP

3.1. Hierarchical Structure Design of Geographic OLAP

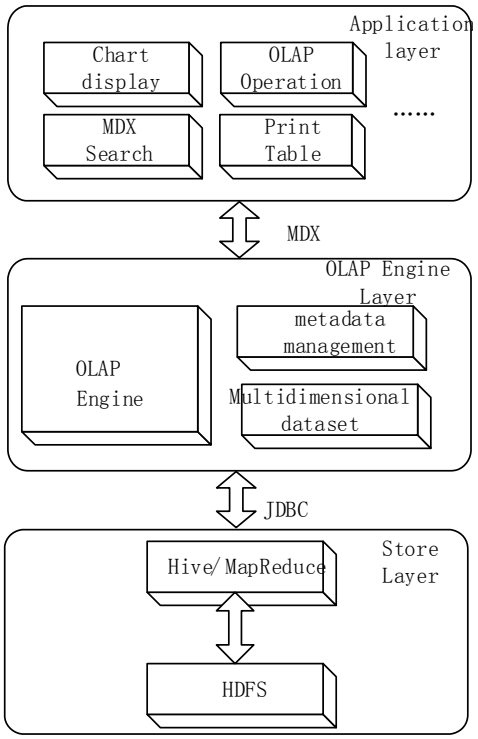

Figure 1.the structural level of Hadoop based geographic OLAP

The geographic OLAP system will be establish on the Hadoop cloud platform. The hierarchical design is shown in Figure 1. Geographic OLAP is divided into three layers from top to bottom and accord to functional modules, namely application layer, OLAP engine layer and geographic data storage layer. Among them, the application layer makes the rendering processing and two analysis operations on the result set returned by the MDX multidimensional query; the OLAP engine layer is the core of the geographic OLAP. The main function is to transform the multidimensional expression MDX statement passed by the application layer to the other query statements in the geographic data storage layer and the result of the query. The set is returned to the upper application layer, and the OLAP engine layer also manages the user defined multi-dimensional data cube, the geographic data cube and the mapping file with the storage layer, and the storage layer uses the distributed data warehouse Hive, which can

not only store the massive geographic data, but also use the MapReduce technology to carry out parallel data. Change the query. Among them, the most important thing is to implement the analysis of MDX multidimensional expressions and converts them into the query statements that can be identified by the storage layer Hive. Hive will parallelize the query according to the HiveQL statement of the translation, and return the multidimensional result set to the engine layer, and transfer the multidimensional result set to the use layer rendering to realize the multidimensional results [9-11].

3.2. Design and implementation of application layer

The application layer of geographic OLAP requires providing a good user interactive graphical interface. Its main function is to render multidimensional result sets returned in the middle layer, and to assist the display with multidimensional table, fold graph, bar graph and pie chart, so that the display of multidimensional results is more vivid. Through the application layer, users can perform typical OLAP operations on the results of multidimensional queries, such as drilling, rolling, turning, slicing, cutting, and so on, and can generate reports in the format of PDF or EXCEL, and provide online printing or downloading for users.

The application layer adopts the custom JSP tag library Jpivot, and defines some related operation buttons of geographic OLAP and the multi-dimensional organization of data through the JSP tag library. In addition, Jpivot also provides the connection function with the data cubes model of the middle layer. The user's multidimensional query requests are all initiated in the application layer, and the application layer is the same. The result returned by these multidimensional queries that is also processed, and rendering the returned multidimensional result to set and display it directly to the user in a multidimensional perspective.

In the application layer, a complete multidimensional query request process is shown in Figure 2. The user sends a MDX multidimensional query request from the browser. After the application layer receives the request, it calls the related program to transfer the multidimensional query request to the middle layer. After the analysis and transformation process of the OLAP engine layer, the MDX multidimensional query statements are translated to the query language that matches the data warehouse Hive. Sentence HiveQL, the result set obtained after Hive query will be returned to the application layer according to the original path, and will be organized and rendered in accordance with the multidimensional form of the application layer Jpivot, and then output to the web page and display to the user.

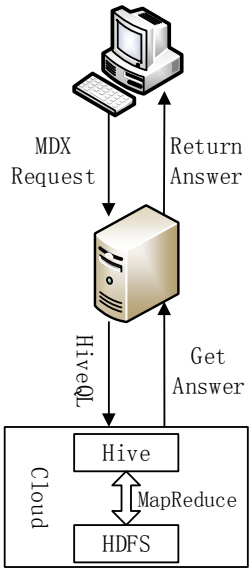

Figure 2. Multidimensional query execution process

The interaction between the application layer and the OLAP engine layer is achieved through the MondrianQuery tag of the custom tag library Jpivot. Because the OLAP engine layer in the middle layer uses open source Mondrian as a geographic information OLAP engine, users can inject the specified related connection parameters, such as JDBC, Schema file paths, and custom MDX multidimensional query statements to Mondrian by MondrianQuery, and connect them to a specified database or data warehouse. Complete the transformation of MDX multidimensional query statement and get the multidimensional result set.

3.3. Design and implementation of OLAP engine layer

The OLAP engine layer is the core of the whole geographic OLAP system and plays a connecting role. The OLAP engine layer of the system uses the open source project Mondrian, which is mainly composed of four parts, such as the Schema management, the session management, the aggregation management, and the dimension management, as shown in Figure 3.

Session manager is the most crucial part. It is mainly responsible for accepting MDX queries, parsing and returning the query consequence set to the application layer. The Schema manager is closely related to initialization, mainly some important data structures, such as the construction of the cache pool and the generation of multidimensional models. The multi-dimensional model defines the themes, dimensions, levels, metrics and computing members of the multidimensional analysis data set, which are mainly constructed from the Schema file in the storage layer. The aggregation manager implements the management of the aggregation table, mainly the management of the OLAP cache, which belongs to the performance optimization. By using the strategy of pre-calculating the summary table (aggregate table), the performance of Mondrian is effectively improved when processing the super large data set. Dimension manager is the management of dimension, and realizes the mapping of columns in dimensions and relational database tables in multidimensional models.

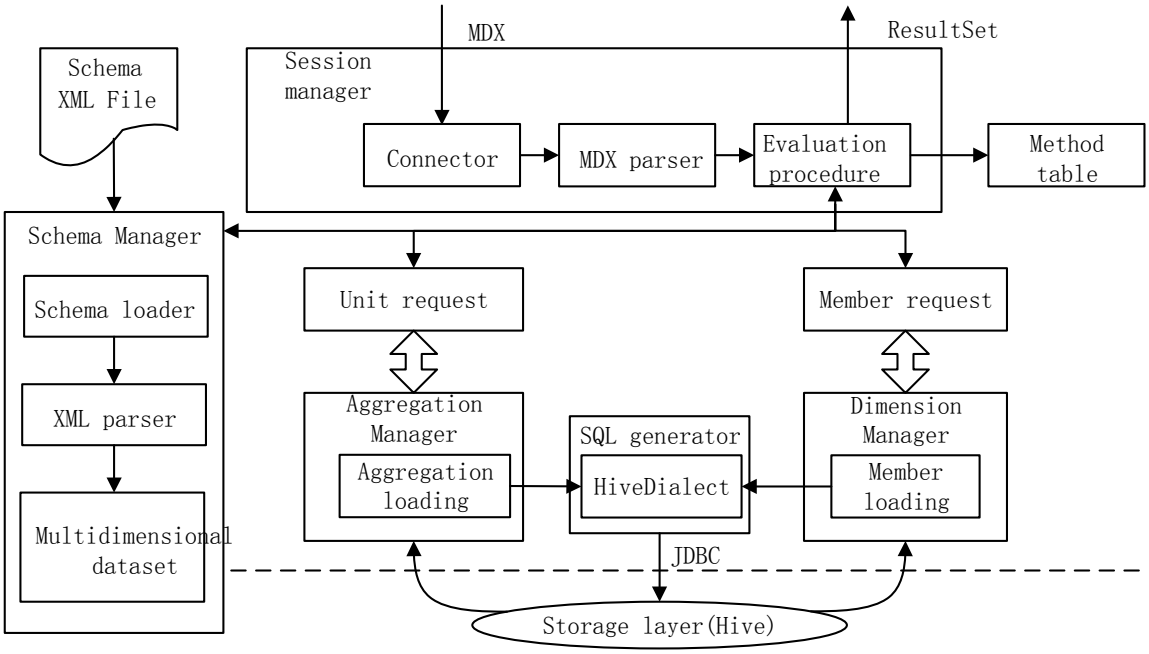

Figure 3. OLAP engine layer structure

3.4. The design and implementation of the storage layer

The storage layer of geographic information OLAP system is mainly organized in the way of ROLAP, and the multi-dimensional data model is organized by star pattern.

Compared with the traditional single OLAP, the main advantage of the geographic OLAP lies in the ability to parallelize multidimensional queries, because the storage layer uses a data warehouse Hive based on the MapReduce architecture based on Hadoop. All the data stored in Hive will eventually be distributed in the distributed file system HDFS of Hadoop, due to the logarithm of the logarithm. The data format and validity will not be checked and modified in the process of loading, but simply copy or move the data content to the corresponding HDFS directory, so Hive does not support the rewriting and adding of the data. All data are fixed at the time of loading, and this also conforms to the data of the OLAP. Features such as multiple writes, multiple reads, few writes, and frequent updates. When the OLAP operation of the application layer is transformed into a MDX multidimensional query statement, the OLAP operation is converted to one or more HiveQL by adding Mondrian's HiveDialect dialect to the MDX query statement, which is queried by Hive of the storage layer via JDBC and ThriftServer, and Hive makes these query statements run to MapReduce to run. The task is submitted to the cloud platform for parallel query, and finally the query results are returned to the upper level applications. The execution process of specific HiveQL query is shown in Figure 7 below.

Figure 4 shows the process of completing query for HiveQL query statements by the Hive of the upper application call storage layer. When the storage layer receives the HiveQL query that has been successfully translated by the application layer through the OLAP engine layer and has been successfully translated to the Hive, these query sentences are converted to multiple Job and submitted to the Hadoop cluster. The Master node is then distributed to each Slaver node for parallel query by the master node, and finally the query result is returned to the application layer. Because of the realization of parallel multidimensional query, the query speed of the system is improved and the response time is significantly reduced in the case of large amount of data.Figure 4 shows the process of completing query for HiveQL query statements by the Hive of the upper application call storage layer.

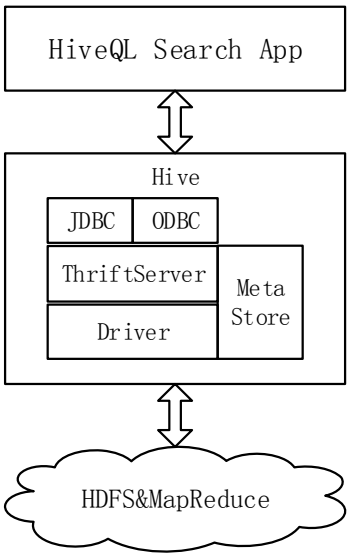

Figure 4. HiveQL query execution process

4. Design and implementation of data mining tools

4.1. Hierarchical structure and function design of data mining tools

Online data mining tools are divided into management layer, computing level and storage layer according to their functions, as shown in Figure 5. The management layer is the interaction interface of the system and the user. The user can implement the related operations to the data set file through the

management layer, including the uploading, downloading, browsing and deleting of the data files. The usage of the mining algorithm in the tool is also selected by the user in the layer. In the computing layer, many commonly used data mining analysis algorithms are provided. Users can select related mining algorithms for mining and analysis of geographic data. These algorithms are composed of two parts. One part is the related clustering, classification, association rules and collaborative filtering related to the open source data mining algorithm library Mahout. The algorithm is part of the parallel sequential learning algorithm (POS-ELM), which is complementary to the Mahout classification algorithm [10-12]. The storage layer uses HDFS, a special distributed file system on cloud platform, to store relevant data files, and do a good job of redundancy backup and other disaster recovery security measures.

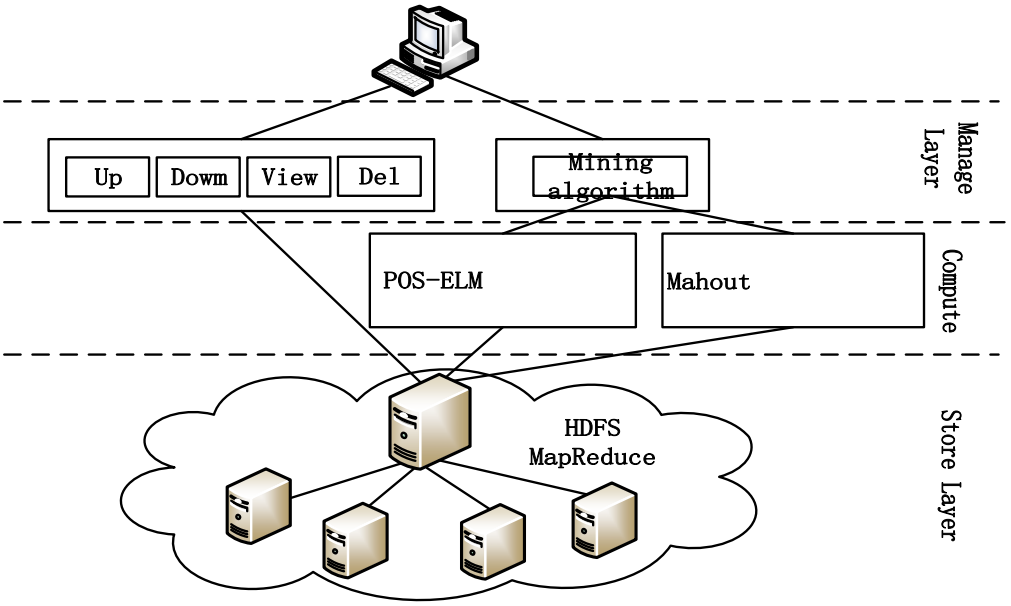

Figure 5. data mining tool architecture

4.2. The design and implementation of the storage layer

The storage layer of the data mining tool mainly completes the storage of the data, including the original data set and the results of the data file after analysis and processing through the related data mining algorithm.

Before the data mining algorithm is used, there are no relevant data files in the Hadoop cloud platform. Users need to upload the data files to the cloud platform through the upload function provided by the mining tools. It is saved in the distributed file system HDFS of cloud platform. The data file will be distributed in the file system HDFS of the cluster in a distributed manner with the configured backup number of the Hadoop cluster, as shown in Figure 6.

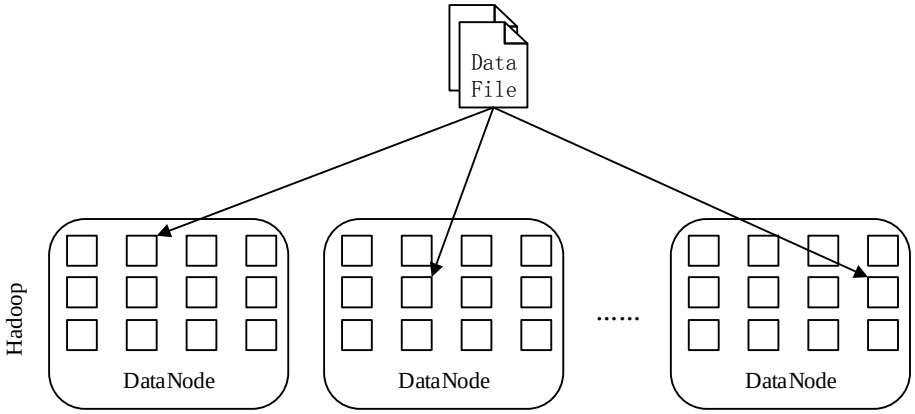

Figure 6. Storage of data files in a cloud platform

5. Conclusion

The OS-ELM classification algorithm in mining tools often has a low classification precision in the face of high dimension and noisy data sets. In this paper, an improved integration strategy, OS-ELM integrated classification algorithm based on random subspace (RSEOS-ELM), is proposed. The algorithm first passes to the original high dimensional data set. In order to reduce the dimension and sampling, many subsets of training data are generated, and then multiple OS-ELM based classifiers are constructed to train the sub data sets. Finally, the majority voting strategies are used to classify these base classifiers. The algorithm improves the classification accuracy of OS-ELM effectively, but because these base classifiers are trained serial, the training time increases linearly with the increase of the number of base classifiers. On this basis, this paper also proposes a parallel integration strategy based on MapReduce. The OS-ELM integrated classification algorithm (PRSEOS-ELM) is used to analyze the matrix operation dependence of RSEOS-ELM in the integrated classification training process, and the multiple OS-ELM based classifier is trained by the MapReduce programming framework. The speed of the integrated classification training is improved and the integrated training is reduced. At the same time, PRSEOS-ELM also has good scalability for large scale data. For example, for data sets with 640 thousand training samples with 40960 dimensions (attributes) and data sets with 409 million 600 thousand training samples with 64 dimensions, if the training subset for each base classifier contains 640 thousand training samples and 64 dimensions, the training time of PRSEOS-ELM is more than that of RSEOS-ELM in the case of constructing the same number base classifier. The training time is 2 orders of magnitude lower. With the increase of the kernel number, the acceleration ratio of PRSEOS-ELM is also increased, and when the kernel number is 80, the acceleration ratio is up to 40 times. From the experimental results, we can see that PRSEOS-ELM algorithm is an optimized ensemble classification algorithm which has both classification speed and accuracy for large-scale datasets.

Acknowledgments

This work was financially supported by the State Grid Corporation of Science and Technology (WBS number: 521104170019).

References

[1] Li Guojie. The Scientific Value of Large Data Research [J]. Communication of the CCF, 2012, 8 (9): 8-15.
[2] Nadim W.Alkharouf, D.Curtis Jamison, Benjamin F.Mathhews. Online Analytical Processing (OLAP): A fast and effective data mining tool for gene expression databases [J], Journal of Biomedicine and Biotechnology, 2005, 2005 (2): 181-188.
[3] Han J, Kamber M, Pei J. Data mining: concepts and techniques [M], Morgan kaufmann, 2006, 20-20.

[4] Witten I P, Frank E. Data Mining: Practical Machine Learning Tools and Techniques [M], Beijing: China Machine Press, 2005, 117-119.
[5] Lam C. Hadoop in action [M]. Manning Publications Co., 2010, 1-30.
[6] Liu Peng. Cloud Computing [M], Beijing: Publishing House of Electronics Industry, 2010.3, 149-151.
[7] Chen k, Zheng WM. Cloud computing: System instances and current research [J], Journal of Software, 2009, 2009: 1337-1348.
[8] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters [C], In the Proceedings of the 6th Symposium on Operating System Design and Implementation, 2004: 137-150.
[9] Chu C, Kim S K, Lin Y A, et al. Map-reduce for machine learning on multicore [J], Advances in neural information processing systems, 2007, 19: 281.
[10] Mondrian [EB/OL], http://mondrian.pentaho.org, 2012-12-18.
[11] Da Silva J, Times V C, Salgado A C, et al. A set of aggregation functions for spatial measures [C], Proceedings of the ACM 11th international workshop on Data warehousing and OLAP. ACM, 2008: 25-32.
[12] Sarawagi S, Agrawal R, Gupta A. On computing the data cube [M], IBM Research Division, 1996, 21