Apriori Algorithm Research Based on Map-Reduce in Cloud Computing Environments

Zhang Danping[1,*], Yu Haoran[2] and Zheng Linyu[3]

[1]*School of Economics and Management, Nanchang Hangkong University, Nanchang, 330063， China*

[2]*College of Information Science and Engineering, Northeastern University, Shengyang, 110819, China*

[3]*Avic GA Huanan Aircraft Industry Co., Ltd. Zhuhai, 5190000， China*

Abstract: Apriori algorithm is a classic data mining algorithm of association rules of data. With the help of cloud computing environment, this paper optimizes apriori algorithm according to the characteristic achieved by MapReduce model that runs in parallel. MR-Apriori algorithm, which is improved by parallelization, reduces time consumption significantly, and its strong extending ability is suitable for large-scale data analysis, processing and mining. According to the cloud computing platforms, high extension capability of MR-Apriori algorithm has been realized based on Hadoop platform. Thus, this conclusion proved the possibility of association rule mining algorithm and cloud computing technologies.

Keywords: Apriori algorithm, association rules, cloud computing, Hadoop, MapReduce model.

1. INTRODUCTION

Data mining needs to obtain large cheap calculation and storage capacity quickly and dynamically. The appearance of cloud computing makes this problem possible to address. People hope to apply cost effectiveness of cloud computing to have an increased storage capacity for data mining and achieve data mining algorithm with high expandability, so that we can address the disadvantages in traditional data mining, and reduce operation cost, as well as promote efficiency of data mining.

2. MATERIALS AND METHODOLOGY

2.1. Cloud Computing

In accordance with cloud computing specification launched by the United States National Standard and Technology Institute (NIST), cloud computing is a running mode of information technology that user can access calculation resources shared(network, servers, storage, and calculation resources and service, etc.) through network on demand conveniently. It quickly achieves features of providing and publishing resources, with minimum management cost or service suppliers intervention [1].

Cloud computing is a new computing model with its own technical characteristics for data storage, data management and programming model. In the aspect of programming model, Google, as cloud computing pioneer in practice, selected MapReduce as a framework for processing huge amounts of data for processing and generating large data sets. Follow-up of Hadoop make MapReduce win support among the users.

Hadoop is a distributed systems' infrastructure developed by the Apache Foundation [2]. The most famous and typical Hadoop are MapReduce and distributed file system (Hadoop Distributed File System, HDFS). While MapReduce finishes decomposition and summary of tasks, HDFS stores data of distributed computing.

There are two important functions in MapReduce: Map and Reduce. Map, which is wrote by the users, generate the input data for processing intermediate key/value pairs. MapReduce libraries have all the intermediate values sharing the same key value, *I*, and then passes the intermediate values to the Reduce function. Reduce function, which is also written by the users, obtain an intermediate key *I* and its collection of all the key values, and merge them into a collection of smaller value. Calling Reduce function each time only generates an output value or zero output value. These intermediate values can be passed to the Reduce function through an iterator. In this way, it can handle some data that cannot be saved by massive memorizer. The input and output of Map and Reduce functions are temporary files, which are the input and output operations' aspect that benefit's MapReduce to large amounts of data.

2.2. Apriori Algorithm and the Advantages of Transplanting to the Hadoop Framework

In association rules mining process, the production of corresponding association rules is achieved through calculating the frequent item-sets in data. There are lots of algorithms of association rules, in which the Apriori algorithm is the most classic. The core of Apriori [3], which was proposed by Agrawal and others in 1994, is that achieve algorithm through recursion of thought of two stage frequent sets. In the other word, we should find all of the candidate sets from transactions sets first, then compare candidate set with predefined minimum supports, and select the candidate set, which is greater than or equal to the minimum supports, as a frequent item-sets. Finally, strong association rules are produced by the frequent item-sets.

The advantage of Apriori algorithm is that the performances of data mining are improved by using Apriori to compress the size of frequent set [4]. However, Apriori algorithm may produce a large number of candidate sets, and may need to require repeated scanning the transaction data set.

If we should transplant the Apriori algorithm to the Hadoop framework, the efficiency of Apriori algorithm for parallel mining would be improved [5]. This combination provides the following three advantages:

(1). Apriori algorithm does not require a lot of memory to produce the intermediate data, therefore computed nodes do not need to have a strong configuration. It is suitable for cloud computing environments.

(2). The Hadoop provides HDFS storage with the good attributes of writing and reading in parallel [6]. Making full use of its performance of distributed storage system can satisfy the needs of Apriori algorithm for a large number of scanning transactions sets. It consumes less time than traditional storage systems for reading transaction sets several times. This makes using Apriori algorithm to do GB or even TB order of magnitude data mining on Hadoop possible.

(3). A large number of calculations in the Apriori algorithm are essentially a process of counting [7], and the usage of typical MapReduce model is also a process of counting. It can be said that Apriori algorithm has the natural characteristics of MapReduce.

Despite the Apriori algorithm in conjunction with Hadoop platform it has a great advantage. Due to distribution and parallelization of the Hadoop framework, it is regulated by itself, it needs to fully take into account features of MapReduce model and Hadoop framework to achieve better performance and high scalability of MR-Apriori algorithm.

2.3. Conversion Programs Based on MapReduce Model

The core content of realizing Apriori algorithm based on MapReduce model is identifying the key data of the original algorithm and then map the data to MapReduce's value of *key* and *value*. Program flowchart of achieving Apriori algorithm based on MapReduce model (Fig. **1**) are as follows.

2.3.1. The Improvement of Statistics of Frequent Item-sets

The core feature of Apriori algorithm is statistics on item-sets in the process of scanning transactions sets [8]. In the process of MapReduce computation, select item-sets as the *key* value for the stage, and the *value* is 1. Hadoop framework split data-sets into a number of subsets through the Map function, and distribute the subsets to all nodes to run and count, and then use the Reduce function to count word number and select frequent *k* item-sets, so that realize parallel improvement in the process of scanning item-sets, and reduce the scanning time significantly.

2.3.2. Grouping to Generate the k Super-set

In the process of generating the *k* super-set, we define the same *k-1* mode as main mode, frequent item with the same main mode as *key* value which is sent to the same Reduce function, and each sub-mode, which is arranged by the last pattern, as the production mode base.

We regard the production mode base as the value, then generate the production mode by Reduce function. At this point, *k* frequent set will be grouped according to the main mode and run in parallel, and the time of *k+1* super-set is greatly reduced.

2.3.3. Read Frequent Set and Accelerate Cutting Item-sets

When k+1 super-set is cut into k+1 candidate sets, we need to do the k item match on each item of k+1 super-set. In the process of MapReduce model generating k super-set, each k item super-set is combined by main mode plus produced mode. Therefore, if all super-sets, whose produced modes are the same with each other, are collected in a Reduce, it only needs to read into subsets which end up with produced mode in k items frequent set, then it can be cut. The time space required by reading k items frequent set can be greatly reduced. At the same time, because it is the subset of frequent set that is read into, the time for cutting is also significantly reduced. In the process of realizing the algorithm, we can classify frequent sets with the same pattern as a group, by using minimum memory, the mapping table of k-tier frequent set and corresponding produced mode is saved. It is easy to quickly navigate to the corresponding subset of frequent sets when cutting, to achieve the purpose of accelerating cutting of item-sets.

Apriori algorithm process based on MapReduce mode can be divided into two steps to implement: data initialization process which makes the original data sets ordered and data iteration process which calculate each layer of frequent set.

3. RESULTS

3.1. The Production of Association Rules

Association analysis is a commonly used method of data mining for discovering patterns which have strong correlated characteristics in sample data. Mining procedure is divided in two steps:

(1). Check every non-empty subset d of frequent item-set f, generate the corresponding rule d⇨(f-d);

(2). Calculate the confidence level support(f)÷support(d). When the ratio is greater than the minimum confidence, association rule is generated.

When the associated rules are produced, if association rule, which is produced by the maximum subset of frequent

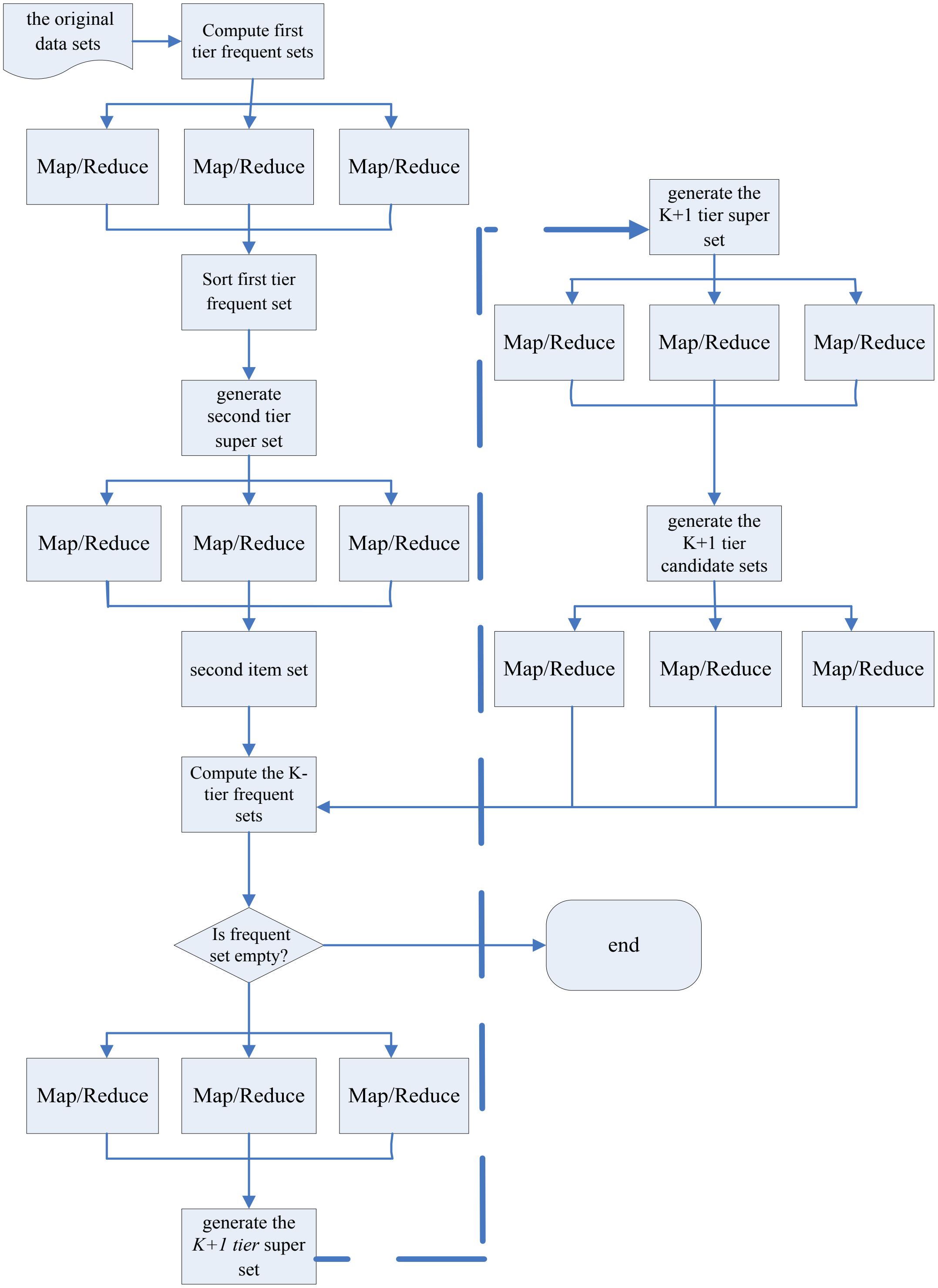
the original data sets
Compute first tier frequent sets
Map/Reduce
Map/Reduce
Map/Reduce
Sort first tier frequent set
generate second tier super set
Map/Reduce
Map/Reduce
Map/Reduce
second item set
Compute the K-tier frequent sets
Is frequent set empty?
end
Map/Reduce
Map/Reduce
Map/Reduce
generate the K+1 tier super set
generate the K+1 tier super set
Map/Reduce
Map/Reduce
Map/Reduce
generate the K+1 tier candidate sets
Map/Reduce
Map/Reduce
Map/Reduce

Fig. (1). The improved Apriori algorithm processes.

item-sets, is dissatisfied with the minimum confidence level, the other subsets are also dissatisfied with the minimum confidence level [9]. According to this characteristic, in specific calculation, we need not take into account these subsets, so that we can improve the overall operational efficiency. At the same time, in order to make the production process of association rules parallel, we can distribute each frequency item-set, which is frequent and concentrative, to different Maps to be produced simultaneously. The above is the implementation process of producing association rules in parallel based on MapReduce model:

```
Mapper{
Map(){
```

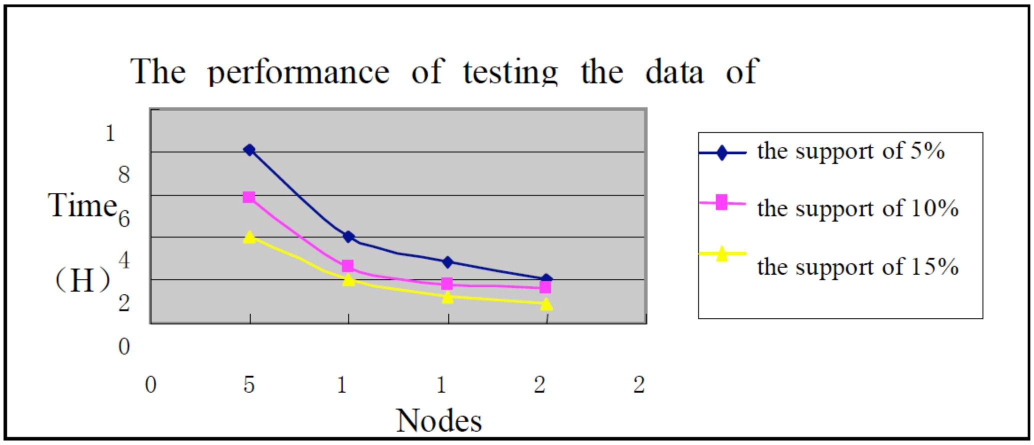

Fig. (2). The performance of testing the data of 10G.

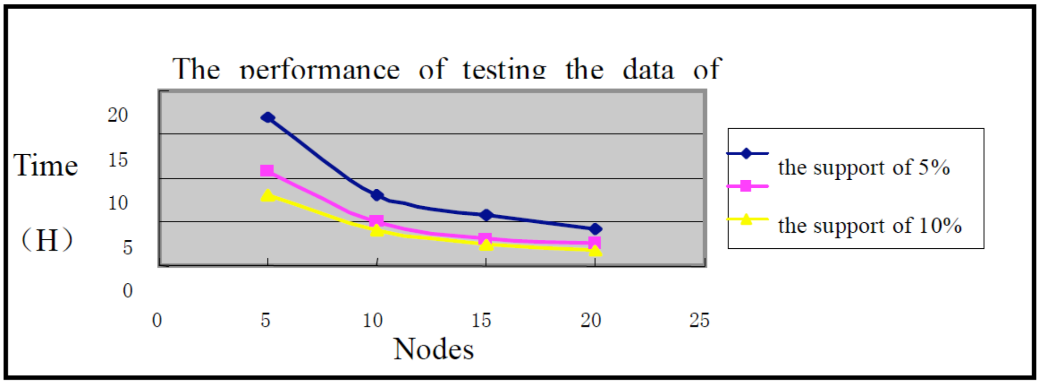

Fig. (3). The performance of testing the data of 20G.

```
//Records for frequent item set/
d=f-1;
i=1;
While(confidence (f,d)≥minisupport&&f>i+1){
i=i++;
Emit (df-d,confidence(f,d);
d=f-i
}}}
Reducer {
Reduce (){
Emit (key,value);
}}
```

3.2. Running Test of the Experiment and Results' Analysis of the Data Mining

Experiments run in the support of 5%, 10% and 15%, and compute nodes are 5, 10, 15 and 20. The capacity used is 10G, 20G and 30G. The data of 10G contains 100 million transactions approximately. The target is calculated from the two aspects of volume of data and the number of compute nodes. The results of the following three experiments can verify the calculated efficiency of MR-Apriori algorithm.

Experiment 1 validates the performance of data of 10G on the operation nodes of 5, 10, 15 and 20. Test results as shown in Fig. (**2**):

Experiment 2 validates the performance of data of 20G on the operation nodes of 5, 10, 15 and 20 respectively. Test results as shown in Fig. (**3**):

Experiment 3 validates the performance of data of 30G on the operation nodes of 5, 10, 15 and 20 respectively. Test results as shown in Fig. (**4**):

According to the results of data tests, it can be seen that if the number of data increases, the calculating time presents a linear growth trend. The increase of the number of nodes makes the operational efficiency improve significantly. It shows MR-Apriori algorithm's feature of processing data in parallel in the Hadoop framework.

On the other hand, in order to investigate the extension capability of MR-Apriori algorithm, this paper defines the linear extension index as follows for validation:

Linear extension index = obtained acceleration multiples / increased resources multiples

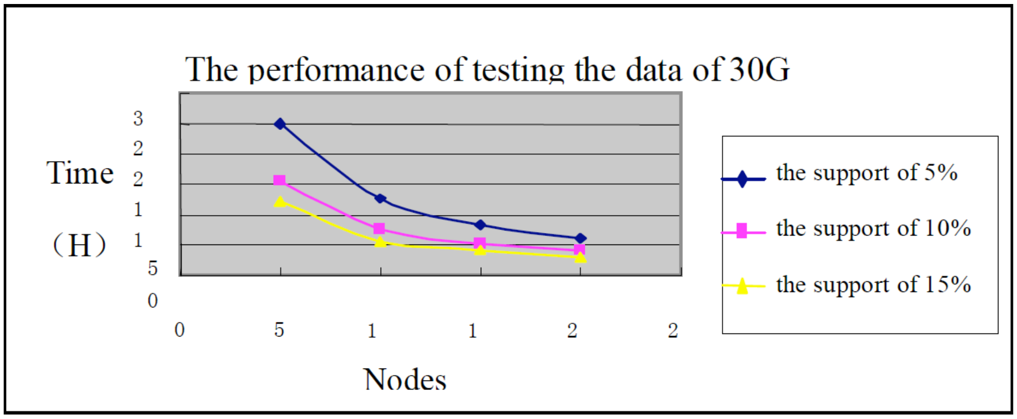

Fig. (4). The performance of testing the data of 30G.

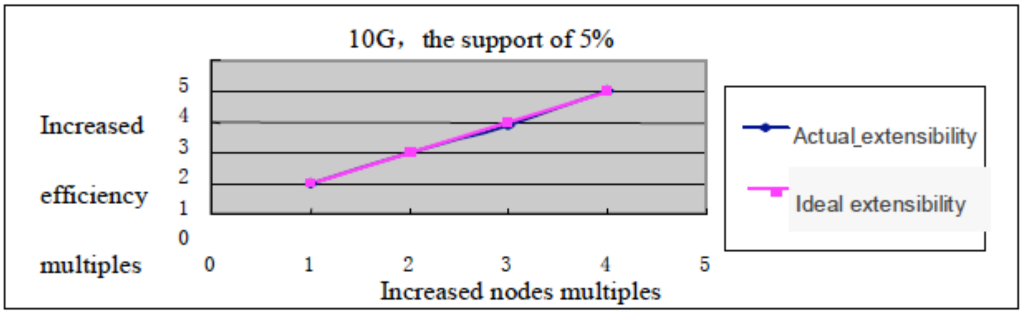

Fig. (5). Linear scalability.

As shown in Fig. (**5**) is the MR-Apriori algorithm's extension capability in the support of 5% with the data size of 10GB. It can be seen from the graph that as the number of nodes increases, MR-Apriori algorithm always maintains a high degree of extension capability, which illustrates that the calculation is extended to computing resources effectively by the MR-Apriori algorithm in the Hadoop framework, adapting to applications of cloud computing.

CONCLUSION

In recent years, because of technological development there has been an increase to the prospects of cloud computing, traditional data mining algorithms are diverted to cloud computing platforms to combine cloud computing with the research and application of existing data mining algorithm, which have become a hot research topic in every industry. On the basis of an in-depth study of Apriori algorithm, this paper has improved the Apriori algorithm according to the cloud computing environment to solve traditional problems encountered in the traditional Apriori data mining. This paper achieved high extension capability of MR-Apriori algorithm based on Hadoop platform, and proved the possibility of association rule mining algorithm and cloud computing technologies.

CONFLICT OF INTEREST

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

ACKNOWLEDGEMENTS

This research was financially supported by the project of science and technology of Department of Education in Jiangxi province (DB200909122) and the project of "Twelve-Five" planning of social science's research in Jiangxi province (11YJ68).

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", *National Institute of Standards and Technology* 2009.

[2] Hadoop, The Apache Software Foundation. http://hadoop.apache.org/core/, 2012

[3] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of the 1993 ACM SIGMOD Conference*, 1993, pp. 207-216.

[4] L. Chen and L. Wei. The Hot Research Topics and the Research Fronts in the Field of Web Data Mining(WDM) Based on Web of Science. *The 5th International Conference on Computer Science & Education Hefei, China*. August 24-27, 2010.

[5] N. Nicholas Carr. *IT is No Longer Important: Converting the Commanding Heights of the Internet—Cloud Computing*. Beijing: China CITIC press, 2008

[6] Cloud Computing Architecture White Paper. *SUN(version 1)*, June 2009.

[7] W. Lin and J. Liu, 'Performance analysis of mapreduce program in heterogeneous cloud computing", *Journal of Networks*, vol. 8 (8), pp.1734-1741, 2013.

[8] S. Sebastian and F. Lukas. Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds. *BMC Bioinformatics*. vol. 13, p. 200, 2012.

[9] R. Gu, X. Yang and J. Yan. SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters. *Journal of Parallel and Distributed Computing*, vol. 74 no, 3, 2014.

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014