Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(7):1152-1157

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Research on algorithms of data mining under cloud computing environment

Fei Long

Department of Business Administration, Changsha University, Changsha, China

---

ABSTRACT

With the arrival of Big Data Time, the process of large-scale data has become the bottleneck problem in many fields. Data mining as a technology combines the traditional data analysis with the data processing algorithm help people get the available information, However, it will become a very long time-consuming process with increasing size of the input data. The cloud computing technology which enforcing computing power in the meantime lowering cost, has a very high scalability, the emergence of cloud computing bringing a new way for its improvement. To improve the traditional data mining algorithm, in this paper, taking the association rules as example, the improved algorithms that can be applied to cloud computing platform were proposed, Through designing the improved algorithm, the paper demonstrates the feasibility of the improved algorithm and analyse it. Experimental results show that the improved algorithms is more efficient compared with the traditional algorithm.

Key words: Data Mining, Cloud Computing, Hadoop, MapReduce

INTRODUCTION

With the rapid development of computer technology and the Internet, the mature and widely used of Web2.0, data growth appears explosively. Modern Internet information has a very rich commercial value. Accurately picking up useful information and knowledge from these data in a high speed allows enterprises to step ahead in the highly competitive commercial so that they can gain commercial success and economic benefit. Nowadays the explosive growth of the Internet data even make the point where a single compute can hardly handle[1]. Data mining as a new technology combines the traditional data analysis method with the complex data processing algorithm to help people get the available information efficiently. Therefore, it is widely used in various fields. However, it will become a very long time-consuming process with increasing size of the input data[2].

The cloud computing technology has been relatively mature which enforcing computing power in the meantime lowering cost. Cloud computing platform which has a very high scalability is ideal for handling large-scale data, Its storage and computing power can be enhanced by dynamically increasing compute nodes[3]. Traditional data processing method for the transformation of distributed computing, and on this basis algorithm improved will have major significance for massive data processing. If traditional data mining algorithms could be transformed and deployed to the cloud computing platform, there is no doubt that the problem of large-scale data mining can be solved. That is, combining data mining algorithms with cloud computing technology is the future trend. Unfortunately, there are less research findings for certain difficult and not all the algorithm can execute on the platform[3]. In short, theory and experiments show that data mining algorithm play a very important role explore the value of data under cloud computing. It produces theory and economic significance for academic research and commercial operations[4, 5].

Based on this, this paper integrates the cloud computing technology and the typical data mining algorithm，and taking the association rules as example, selects Apriori algorithm to improve, willing enhance its efficiency when

dealing massive data. Considering above problems, this paper began with the analysis of the theories of data mining and cloud computing. Learning about the data mining theory and cloud computing technology deeply, knowing about the meaning of data mining and the basic process of the data mining, analyzing MapReduce programming model and operation mechanism, summarizing the cloud computing platform advantages, combine with related parallel computing knowledge. Combing with the typical data mining system architecture and integrating with cloud computing, the paper brings up the data mining system architecture based on cloud computing. On the basis of elaborating Apriori algorithm, aming at its bottleneck when dealing with massive data, using MapReduce programming module, the paper presents the idea of parallel improvement based on the partition. Through designing the improved algorithm, the paper demonstrates the feasibility of the improved algorithm and analyse it. Experimental results show that the improvement algorithm is more efficient compared with the traditional algorithm, which reduces the time complexity and space complexity.

The remainder of this paper is organized as follows. Section 2 describes the cloud computing and Data Mining algorithm in details. The proposed algorithm is depicted in detail in Section 3. Experiments and results analysis are presented in Section 4. We finish in Section 5 with conclusions and an outlook on future work.

BACKGROUND

Data Mining

In data mining, the association rule is a popular and well research method for discovering interesting relations between variables in large databases. Agrawal[5]introduced associations for discovering regularities between products in large scale transaction data recorded in supermarkets. This section describes some basic definitions and concepts which are used in the design of the proposed model. Apriori algorithm is to find frequent itemsets from a transaction dataset and derive association rules. To find out frequent itemsets is important. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence[6].

The fundamental of the Apriori algorithm is “if an itemset is not frequent, any of its superset is never frequent”. Let the set of frequent itemsets of size k be Fk and their candidates be Ck. Apriori algorithm scans the database at first and searches for frequent itemsets of size 1 by accumulating the count for each item and collecting those itemsets that satisfy the minimum support requirement[7].The Apriori algorithm iterates on the following three steps and extracts all the frequent itemsets. Generate Ck+1, the candidates of frequent itemsets of size k+1, from the frequent itemsets of size k. Scan the database for calculating the support of each candidate of frequent itemsets. Add those that satisfy the minimum support requirement to Fk+1. The Apriori algorithm is shown in Fig.1.

```
F1=(Frequent itemsets of cardinality 1);
for(k=1;Fk≠Φ;k++) do begin
    Ck+1=aprio-gen(Fk);
    for all transactions t∈Database do begin
      Ct' =subset(Ck+1, t)
         for all candidate c∈Ct' do
           c.count++;
         end
      Fk+1={C∈Ck+1 | c.count≧minimum support}
    end
 end
 Answer ∪kFk;
```

Fig. 1: Apriori Algorithm

Apriori algorithm is to find frequent itemsets from a transaction dataset and derive association rules. Because of the combinatorial explosion, To Find out frequent itemsets is important. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence[8]. As one of the most influential algorithms of data mining，the Apriori algorithm discovers frequent itemset through an iteration method which called layer by layer search. The Apriori algorithm is simple and easy to understand and easy to implement. In this algorithm, the k-itemset is used to generate the (k+1)-itemset, the frequent k-itemsets are extracted from the candidate k-itemsets. However, the Apriori algorithm has some disadvantages. The Apriori Algorithm scans the database too many times. The Apriori algorithm need repeatedly scan the transaction database which may be very large and thus the scale of the database is the important factor to the Apriori performance[9].The Apriori algorithm may generate vast intermediate itemsets. The Apriori algorithm generates Ck+1, candidates of frequent itemsets of size k+1, from the frequent itemsets of size k. The number of the Ck+1 may be very vase. The Apriori algorithm then scans the database to calculate the support of each candidate of frequent itemsets[10]. Because of the huge number of the candidates frequent itemsets, the process of confirming the

candidates will take a lot of time.

Cloud Computing

Cloud computing is basically services-on-demand over the Internet. It is a natural evolution of the widespread adoption of virtualization, service-oriented architecture and utility computing. Cloud computing is a computing capability that provides an abstraction between the computing resource and its underlying technical architecture, enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. There is no formal definition commonly shared in industry, unlike for Web 2.0, and it is very broadly defined as on-demand provisioning of application, resources, and services that allow resources to be scaled up and down. Clouds are a large pool of easily usable and accessible virtualized resources[11]. These resources can be dynamically reconfigured to adjust to a variable load, allowing also for an optimum resource utilization.

Cloud computing includes a number of technologies such as Virtualization, Web services, Service Oriented Architecture, Web 2.0, Web Mashup. One of the key new technologies used in the cloud are scalable batch processing systems. The main reference implementation here is Google MapReduce and its open source implementation Apache Hadoop originally developed at Yahoo. Distribution and scalability are playing important roles in the cloud, and for that Hadoop can be used. Hadoop is a cloud computing program created to deal with the growing demands of modern computing and storage of massive amounts of data. Many companies, especially ones that operate through demanding websites, e.g. Amazon, Facebook, use Hadoop.

Hadoop is an open-source project of the Apache Software Foundation, and the concept was inspired from the search technology proposed by Google Inc. Hadoop is written in Java language; any machine that supports Java can run the Hadoop software. It has its own distributed file system called Hadoop Distributed File System. Hadoop is an Apache project. Hadoop enables the development of scalable, efficient and distributed computing using very simple Java interfaces. Using the Hadoop platform, programs can be developed that facilitate the processing of large amounts of data. HDFS based database used mainly for batch processing is HBase which is heavily inspired by Google Bigtable. The main applications for Hadoop seem to be log analysis, Web indexing, and various data mining and customer analysis applications.

PROPOSED ALGORITHM

Although there exist some improved algorithms and have the ability to reduce the times of scanning original dataset, but almost all of those are depending on sacrificing disk space, the traditional improved algorithm are not suitable for deals with large-scale data. We proposed a new algorithm of Apriori algorithm under cloud computing environment, called MRApriori algorithm. It use Hadoop components to perform job execution and information storage. It deploy MapReduce to parallel the first step and have the data stored in the file replace the database in order to deal with in the Hadoop Distributed File System (HDFS).

HDFS is a master/slave architecture that consists mainly of Namenode software and multiple copies of Datanodes software to compose a cluster. In the files each line can be seen as a transaction, each item is separated with white space. The workflow of HDFS is shown in Fig. 2.

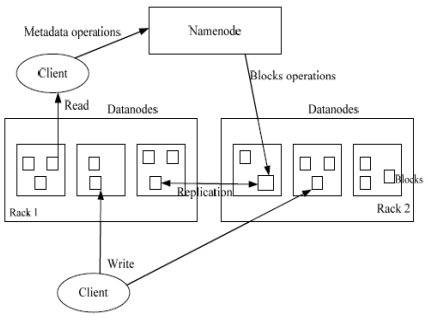

Fig. 2: The workflow of HDFS

A HDFS cluster consists of a single NameNode and a number of DataNodes. The master server is responsible for managing the namespace and file operations of Datanode software. The NameNode and DataNodes are designed to run on machines. The usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNodes software. A file is divided into multiple blocks and is stored on a group of Datanode software programs. Datanode software is responsible meeting users' needs of

HDFS, The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment one machine usually runs one DataNode.

The corn idea of cloud computing technology is MapReduce which all the strategy of changing for data mining algorithm under cloud computing has map and reduce two different parts. The workflow of MapReduce is shown in Fig.3.

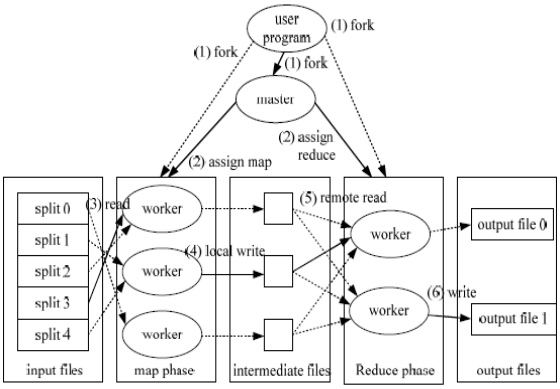

Fig. 3: The workflow of MapReduce

The map function receives input pairs and produces a set of intermediate key/value pairs, then sends intermediate key/value pairs to reduce nodes. The datasets in files are split into smaller segments automatically after stored in HDFS and the map function is executed on each of these data segments. The reduce function accepts an intermediate key and a set of values for that key and merges these values together to form a possibly smaller set of values[15]. All algorithms must separate into those two functions, if not that will difficult or unable to run on the platform. It only needs two MapReduce phases to find frequent itemsets.

The MRApriori algorithm consists of two steps: the first is generating all frequent itemsets, the second is generating confident association rule from the frequent itemsets.

In first step, The task of is to scan each record of the input item subset and generating candidate itemsets, then the frequent itemsets are found. The original transaction data is automatically divided into N data subset by the HDFS, and then the subsets are sent to N nodes, Formatting the N data subsets, each of the N nodes processes the data subset independently, and then generating <key,value> pairs where key is transaction id and the input of the Map function, and their formats are <transaction_id, list>, and then generates candidate itemsets, setting the support count of candidate itemsets to 1. Each of the N nodes separately merges the support count by using Combiner function, if each candidate itemsets is same. The same candidate itemsets in N nodes are sent to the same node by using the Hash function.

What differ our map function from the previous related algorithms is the modification of its value parameter to take the whole split as an input, instead of one line at a time, and then we apply the traditional Apriori algorithm on that split with partial minimum support count equal the number of transactions in the split multiply by the minimum support threshold. The definition of <key,value> is the first step to implement the function of MapReduce. The pairs of key/value are shown in Fig.1, , we can design the map function and reduce function using the <key,value>pairs.

Input/Output	Input: <Key,Value> pairs	Output: <Key/Value> pairs
Map function	Key: transaction_id Value: one row of data (Key)	Key: candidate items Value: 1
Reduce function	Key: candidate items Value: 1	Key: candidate item Value: support

Fig. 4: Key/Value pairs for map and reduce function

The map function mainly collects the count of every item in candidate itemsets and the reduce function prunes the candidate itemsets which have an infrequent sub pattern. Each frequent item is generated through one execution of map and reduce function. After stored in HDFS, the datasets are split into smaller segments and then transformed to datanodes. Map function is executed on these data segments and produces <key,value> pairs. The framework groups all the pairs, which have the same item and invokes the reduce function passing the list of values for candidate items. The map's output is a list of intermediate <key,value> pairs: grouped by the key via combiner, and stored in the

map worker, where the key is an element of partial frequent k-itemsets and the value is its partial count. When all map tasks are finished, the reduce task is started. The map's output are shuffled to the reduce worker that calls a reduce function.

The reduce function adds up all the values and produce a count for the candidate item as a one-time synchronization. This algorithm's advantage is that it doesn't exchange data between data nodes, it only exchanges the counts. In every scan, each map function generates its local candidate items, then the reduce function gets global counts by adding local counts. The output of reduce function is a list <key,value> pairs , where the key is an element of partial frequent k-itemsets and the value equal one, stored in HDFS.

In second step, The map function of this phase counts occurrence of each element of partial frequent k-itemset in the split and outputs a list of key/value pairs, where the key is an element of partial frequent k-itemset and the value is the total occurrence of this key in the split. The reduce function outputs a list of key/value pairs, where the key is an element of global frequent k-itemsets and the value is its occurrence in the whole data set.

Performance Evaluation

In this section, the experiments and performance evaluation results analysis are presented. All of the algorithms were implemented in Java: JDK version is 1.6.31. In our experiments, we used Hadoop version 0.19.2, running on a cluster with 6 machines(1 master, 5 slaves). Each machine has duplicate-core processors (running at 2.60GH) and 2GB memory.

This experiment is based on real datasets, to calculate the correlation of products. Dataset mainly determines the performance of the parallel algorithm. If experiment with the small datasets, its performance turned out to be lower for the reasons, extra communication time occupying a large proportion compared to the total execution time. The experiment compares the traditional Apriori with the new MRApriori algorithm.

In order to test the performance of the two algorithms, we use the execution time as criterion to measure our algorithm's superiority. All experiments were executed two times and the average was taken. We have evaluated the performance of our proposed MRApriori algorithm by comparing its execution time with the execution time of the existed Apriori algorithms. Figure 1 show the running time of Apriori algorithm and MRApriori algorithm.

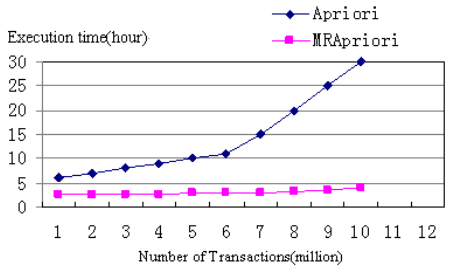

Fig. 5. Experimental results.

The results show that our proposed MRApriori algorithm outperforms the traditional Apriori algorithm and it will continue outperforms them as the datasets size is increased. When the processing datasets are proportionally increased, the running time of the MRApriori algorithm has little change before and after, it shows that the MRApriori algorithm have good scalability in the MapReduce environment. This is easily predicted from experiments where we noticed that the more data a node processes, the less significant proportion becomes the communication time. The opposite effect is simply seen here, larger datasets would have shown even better speed up characteristics. In the case of more than one data node, speedup increases with the increase of number of data in certain circumstances, which can prove the high efficiency of the parallel Apriori based on MapReduce. The results show that under the cloud computing environment, the improved algorithm can effectively mine the frequent itemsets from mass data, and the dataset partition method and distribution method can improve the efficiency of the improved algorithm in the heterogeneous cluster environment. In the MapReduce environment, MRApriori algorithm eliminate the need to iterative scanning of the data to find all frequent items. AprioriMR repeats scanning other intermediate data that usually keep shrinking per iteration. when Apriori and MRApriori algorithm deal with the different sizes datasets, with the increase of Datanode in the cluster, the running time is proportionally decreased.

CONCLUSION

Cloud computing broughts new ways for data mining algorithms, and combining data mining algorithms with cloud computing technology will become the future research trend. This paper taking the association rules as example, provides a reference for other algorithms' improvement of data mining. More and more algorithms to be parallelized and transplant to the cloud computing platform. In short, theory and experiments show that data mining algorithm play a very important role mining the valuable data under cloud computing environment. It produces theory and economic significance for academic research and commercial operations.

REFERENCES

[1]Liu Xiao-lan. *China Sport Science and Technology*. **1984**, 29(13), 46-49.
[2]Luo Yang-chun. *Journal of Shanghai Physical Education Institute*. **1994**, 23(12), 46-47.
[3]Wan Hua-zhe. *journal Of Nanchang Junior Colle*ge. **2010**, 3, 154-156.
[4]Li Ke. *Journal of Shenyang Sport University.* **2012**, 31(2), 111-113.
[5]Zhang Shu-xue. *Journal of Nanjing Institute of Physical Education.* **1995**, 31(2), 25-27.
[6]Pan Li. *Journal of nanjing institute of physical education(natural science).* **2004**, 19(1), 54-55.
[7]Li Yu-he; Ling Wen-tao. *Journal of Guangzhou Physical Education Institute*. **1997**, 17(3), 27-31.
[8] Xu Guo-qin. *Journal Of Hebei Institute Of Physical Education.* **2008**, 22(2), 70-72.
[9] Chen Qing-hong. *China Sport Science and Technology*. **1990**, 21(10), 63-65
[10] Tian Jun-ning. *Journal of Nanjing Institute of Physical Education*. **2000**, 14(4), 149-150.
[11] Bing Zhang. *Journal of Chemical and Pharmaceutical Research*, **2014**, 5(2), 649-659.