# Application and Realization of an Improved Apriori Algorism in a Hadoop Simulation Platform for Mass Data Processing

http://dx.doi.org/10.3991/ijoe.v12i02.5037

SUN Ya-ni, CHEN Xinhua
Sichuan Information Technology College, China

*Abstract*—**This paper presents the open source distributed cloud computing platform Hadoop from the Apache foundation as the basic platform. The paper introduces research about the Apriori distributed DM algorism in a Hadoop platform after analyzing the Hadoop platform structure, provides improvement and performance analysis, and finally describes the construction of and simulation research for a Hadoop mass data process platform.**

***Index Terms*—Apriori Algorism, Hadoop, Simulation Platform, Mass Data Process**

## I. INTRODUCTION

A Hadoop Distributed File System (HDFS) is a file system developed for a Hadoop project that adopted master/slave architecture [1]. HDFS consists of a NarneNode and numerous DataNodes. HDFS provides the users with corresponding file namespace for storing data in a file format. Generally, HDFS divides these files into several file blocks, which are stored in a group of data services [2-6]. Then NameNode provides fundamental functions such as opening, closing and renaming the files and directories, while being responsible for mapping the file blocks to the DataNodes. Then the DataNode is responsible for responding to the operations of read and write for concrete files in the client side, while dealing the requirements for establishing, deleting and backing-up the data blocks launched by NameNode, as shown in Figure 1.

In the typical Hadoop cluster topology structure, there is another server resident in the Secondary NameNode to prevent the server NameNode resident from cluster collapse resulting from failure. NameNode and JobTracker can be arranged in the same Master node in small-sized clusters with lower pressure for data processing, while in the large-sized clusters, it is better to arrange data in two different servers due to more frequent data exchanges [7]. DataNode and TaskTracker were arranged in every computing node as slave node responsible for the storage and computation tasks, as shown in Figure 2.

## II. IMPROVEMENT AND OPTIMIZATION OF APRIORI ALGORISM IN DISTRIBUTED PLATFORM

Correlation analysis is an important component of the collection of data process algorisms, and its typical use scenario lies in the analysis of historical transaction data. After analyzing the purchasing behavior of a group of customers, you may find that those who buy A also buy B, which may account for 80% of the total, indicating that A and B have an incidence relation of positive correlation.

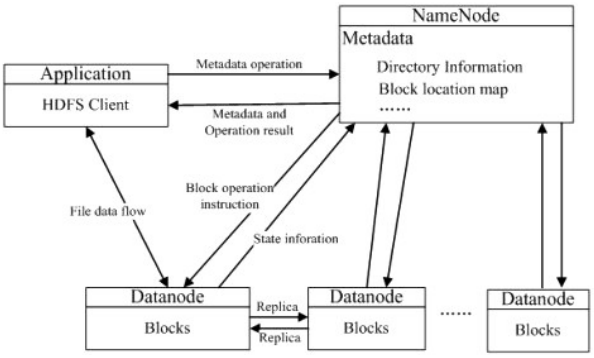

Figure 1. Structure chart of Hadoop platform

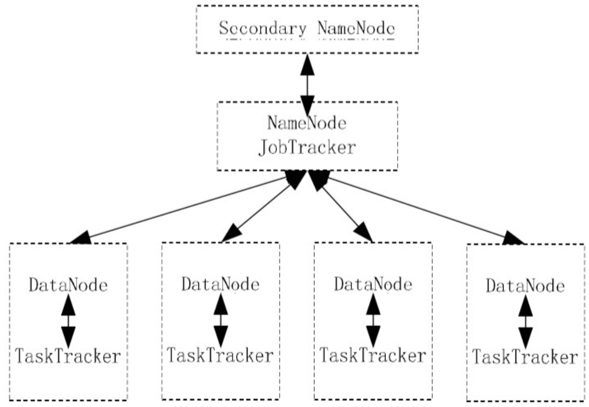

Figure 2. Topological graph of Hadoop platform

The dealers can adjust their commercial sales behaviors and add sales promotion links according to this incidence relation. Mining for incidence rules has significant significance in directing commercial behaviors of enterprises and improving sales performance.

In the traditional parallel Apriori, whether in a CD algorism or a DD algorism, some problems need further research, such as waiting time before synchronization and excessive communication traffic, etc. This paper puts forward improvement and optimization based on a traditional parallel Apriori algorism and research and explores the point of view of candidate set generating and load balancing.

### A. The improvement principal for generation algorism of candidate set

There are repeaters appearing during generation of candidate sets in the traditional Apriori algorism. For example, when the number of frequent item-sets （k-1） is n, the corresponding k, namely the number of candidate set shall reach 2, among which are repeaters and false items. The conditions above are more obvious when generating 2-candidate sets from frequent 1-item sets.

In distributed systems with very large data being processed, the problem has a strong impact on the execution efficiency of mining algorisms.

To solve the problem, the train of thought given by this paper is to carry out blocked jobs with unbalanced characters toward the mass objects database. The number of certain item-sets domains absolutely in each data block, which was the sole focus during the search of data in each database. According to statistics and dealing with the data occupying the dominant position, the overall properties of data can be reflected to a large extent in spite of the mass data. So, as for the generation of candidate sets in an Apriori algorism, pruning can reduce the repeated candidate sets to a large extent according to the statistics of advantageous data.

First, the definition of partition pruning shall be given.

$$\max\sup(B) = \sum_{i=1}^{n} \max\sup(B)_i$$

Divide the object database D given into n parts, and $D_i\ (1 \le i \le n)$ represents a part in them. For each candidate X, $\sup(X)$ represents its global support while $\sup(X)_i$ represents the support of X in local database $D_i$. If item-set A is the candidate set with the length of k, and $A \subset B$, then $\sup(A) \ge \sup(B)$, in that way there is a necessary being that the ULV of $\sup(B)_i$ is $\max\sup(B)_i = \min\{\sup(A)_i\, A \subset B 且 |A| = k-1\}$. Based on the fact of a blocked database, the global supports of item-set B are:

$$\max\sup(B) = \sum_{i=1}^{n} \max\sup(B)_i$$

In the case that $\max\sup(B)$ is smaller than the threshold value of the minimum global confidence coefficient, then pruning shall be given to item-set B. The content above is the definition of partition pruning based on global pruning extension.

The mass data process under a distributed system must divide the database and then store, and the theory of partition pruning is a just fit for the fast generation of candidates in parallel Apriori algorism, which we call virtual partition pruning.

### B. Validation verification of improved algorism

For example, the frequent 1-item-set generated by a search of object database D is shown in TableⅠ.

Assume the database D is $\{A, B, C, D, E, F\}$, and its minimum support is 30.

As seen in Table 1, D is divided into 3 partitions with unbalanced characters, and its support of 1-item-set has been analyzed. We can see that since all supports of a 1-item-set are larger than 30, pruning cannot be done using a traditional Apriori algorism and that $C_6^2 = 15$ times statistics computation shall be needed to scan the database for 15 times.

$$A \subset \{AB\} \cap B \subset \{AB\}$$

By using the above virtual partition pruning theory and taking {AB} item, which is concentrated with the 2-items in the object database, for example, $A \subset \{AB\}$ and $B \subset \{AB\}$, so we get:

$$\max\sup(AB)_1 = \min\{\sup(A)_1, \sup(B)_1\} = \min[26,65] = 26$$

$$\max\sup(AB)_2 = \min\{\sup(A)_2, \sup(B)_2\} = \min[2,6] = 2$$

$$\max\sup(AB)_3 = \min\{\sup(A)_3, \sup(B)_3\} = \min[3,5] = 3$$

There is

$$\max\sup(AB) = \sup(AB) + \sup(AB)_2 + \sup(AB)_3 = 26 + 2 + 3 = 31$$

Since its maximum confidence coefficient is larger than minsup, it shall be reserved.

In the same way we can establish the maximum confidence coefficient of {EF} is 30, which shall also be reserved, and beyond that other items are all smaller than 30 and need to be pruned.

In conclusion, we found the frequent 2-item-sets of object database D are {AB} and {EF}.

It can be verified that the frequent 2-item-set evaluated by a traditional Apriori algorism and evaluated by the virtual partition pruning method are the same.

In the process of virtual partition pruning to generate the frequent （k+1） item-set, scanning of the object database is not needed and generation of repeated false item-sets is fully reduced, which can largely enhance the efficiency of mining association rules in a mass object database.

According to mathematical statistic theory, the larger the data volume, the more accurate the statistics of the advantageous data are, so virtual partition pruning algorism has a high value in mass data processing.

TABLE I.
PRUNING EXAMPLE OF VIRTUAL PARTITION

| Item-set | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Support of partition 1 | 26 | 65 | 2 | 4 | 4 | 4 |
| Support of partition 2 | 2 | 6 | 24 | 60 | 28 | 8 |
| Support of partition 3 | 3 | 5 | 4 | 2 | 24 | 62 |
| Global support | 31 | 76 | 30 | 66 | 30 | 74 |

## III. THE BUILDING OF HADOOP CLOUD PLATFORM IN THE LAB ENVIRONMENT

The deployment form of a Hadoop platform can be divided into 3 classes: standalone mode, pseudo-distributed mode and fully distributed mode (standard cluster mode). We shall establish the Hadoop platform in a fully distributed mode according to the requirements from the research project. Brief introductions follow.

### *A. Establishment of network topological structure*

Machinery preparations: one set of master, ten sets of slave. Install Linux (Cent OS 5.7) in advance for all 11 machines. Configure the /etc/hosts for each machine to ensure they can exchange visits according to the machine names.

Topological structure of Hadoop cluster in lab environment:

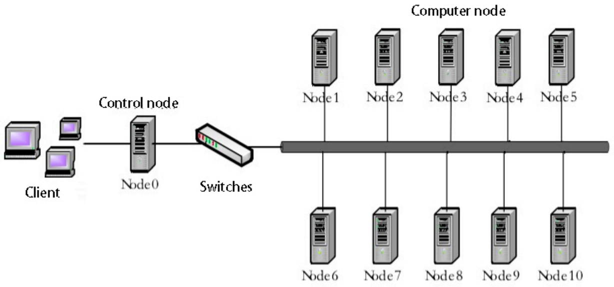

Figure 3. Hadoop cluster topology graph in experimental environment

### *B. B. Hadoop software deployment*

The machinery software deployments in Hadoop platform are as follows:

*1) Install JDK*

Install JDK for all the machines in the cluster to prepare the Java operating environment.

After installation, check the JDK edition No. for every machine to ensure the JDK edition Nos. used by all the machines are the same.

*2) Deploy users*

Configure the same directory structure for all the machines in the cluster or establish exactly the same user groups, and then carry out installation, taking the home path of users as the Hadoop installation path.

*3) Install ssh and carry out configuration*

Since after the Hadoop has started and runs normally as the Master Node, the NameNode shall control the start and stop of the daemon processes deployed in the DataNode, the ssh must be installed and configured for each machine in the cluster beforehand.

a. Install ssh: its installation order is sudo apt-get install ssh

b. Configuration: The monitoring of the NameNode on the DataNode is unconditional, so the authentication form of ssh shall be deployed to the form of a public key authentication that does not need a passport.

Check whether the configurations are successful or not after configuration. Visit other nodes using ssh in the Hadoop machine. The configuration is successful if the password is not required to enter when carrying out orders, otherwise re-configuration will be taken for the corresponding notes.

*4) Install Hadoop software*

Install and deploy the software for each server after completing the above configurations. The precondition for normal operation of a Hadoop is that the installation paths of the Hadoop software in each machine in the cluster are all in full accord.

*5) Configure the conf/hadoop-env.sh file*

Configuring the conf/hadoop-env.sh file is required after the Hadoop software has been installed and adding the JDK path in this file ensures the Java file under Hadoop can be executed smoothly.

Up to now, the experimental environment of the Hadoop distributed cluster with one NameNode and ten DataNodes has been set up. Next, we will configure the XML files.

## IV. SIMULATION EXPERIMENT AND RESULTS ANALYSIS OF IMPROVED APRIORI ALGORISM

### *A. Experimental environment and test data*

Experimental testing environment: Hadoop cloud computing environment is the same as the last part

Test the environment for comparison experiments: Standalone server with 4GB RAM and dual core 3.2 GHz processor

Experiment data: Object data output in the preprocessing modal in 5.2 part

Number of affairs: 11,354 articles

### *B. Experiment results and analysis*

Realize CD, DD in the distributed Apriori algorism and improved algorism in MapReduce in Hadoop cloud platform and deploy them in the Hadoop cluster set up in the lab.

Figure 4 represents the time consumed in finding the frequent k-item set $L_k$ in the database of CD and DD of a traditional Apriori algorism in a Hadoop platform and an improved Apriori algorism.

It can be learned by analyzing Figure 4 that the execution efficiency of an improved Apriori algorism in a Hadoop cloud platform has been effectively improved and the advantages compared with two other algorisms are more obvious when k has a larger value in finding a frequent k-item set.

Based on the experiments above, open the number of nodes in the Hadoop cluster in batches and finding a frequent 1-item set efficiency chart of chart 5 under conditions with different nodes can be obtained.

The number of nodes in the Hadoop cluster is also one of the key factors influencing the execution efficiency of the whole algorism. When the nodes in the Hadoop cluster are less, the time required by the improved algorism in a finding frequent 1 item set was longer, but with the increase of nodes in the cluster, the execution efficiency dealing with the same input data improved quickly, resulting in shorter process time. As the nodes in the cluster increase to a certain scale, due to communication consumption, etc., the growth degree of processing efficiency shall be slowed gradually, until stopped.

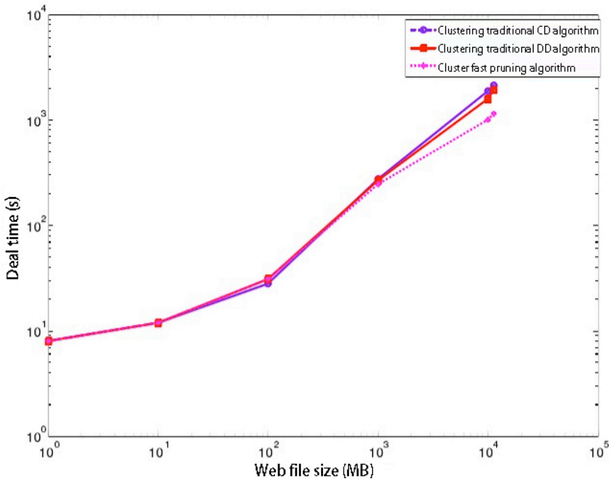

Figure 4. The efficiency comparison chart of traditional CD, DD algorism and improved Apriori algorism in Hadoop cluster

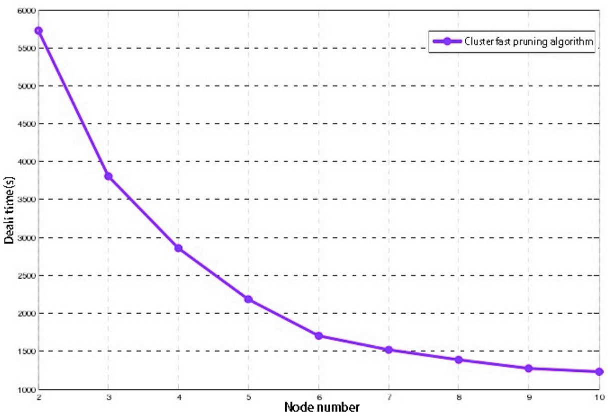

Figure 5. Efficiency chart of finding frequent 1-item set under conditions with different nodes.

### C. *Experimental conclusions*

It was found that compared with the traditional CD and DD algorisms, the improved distributed Apriori algorism this paper described improved performance in a Hadoop platform and the improvement in performance is more obvious when k in the finding frequent k-item set is larger.

During the processing of mass data, the number of nodes in a Hadoop cluster is also a significant factor influencing mining efficiency. Within limits, when the data volume is larger, the more nodes there are and the stronger the computing capacity of the cluster and the higher the process efficiency are.

## V. CONCLUSIONS

The research direction of this paper comes from the requirements of a mass data process in actual project demands. This paper mainly studied the Apriori parallel hierarchical search method for mass data processing based on a Hadoop cloud platform, picked an improved parallel Apriori algorism, ran it in MapReduce, and then carried out simulation analysis under the Hadoop platform. The simulation results show that in case of mass data, the capacity of mass data processing of an improved distributed preprocessing model combined with an Apriori algorism in the Hadoop cloud platform was notably improved. It was also noticed that the characteristics of the distributed system determined that when facing scattered small data, there is a limited improvement space for the computing performance of a cluster distributed process over the traditional standalone mode.

## REFERENCES

[1] Chen, Z. (2014). An improved ant algorithm qos routing on hadoop platform. Fire Control Radar Technology.

[2] Chen, Y., Zhejiang Industry amp, & Trade Vocational College. (2014). The construction of flipped classroom model based on the cloud platform of the worlduc.com for design courses in higher vocational college. Journal of Zhejiang Industry & Trade Vocational College.

[3] Dede, E., Govindaraju, M., Gunter, D., Canon, R. S., & Ramakrishnan, L. (2013). Performance evaluation of a MongoDB and hadoop platform for scientific data analysis. Proceedings of the 4th ACM workshop on Scientific cloud computing. ACM. http://dx.doi.org/10.1145/2465848.2465849

[4] He, Y. (2013). A survey of improved apriori algorithm. Microcomputer & Its Applications.

[5] Hong, H. X., Zhang, W. Q., Shen, J., Zhenqiang, S. U., Ning, B. T., & Han, T., et al. (2013). Critical role of bioinformatics in translating huge amounts of next-generation sequencing data into personalized medicine.. Science China-life Sciences, 56(2), 110-8. http://dx.doi.org/10.1007/s11427-013-4439-7

[6] JI Huai-meng, Sunshine College, & Fuzhou University. (2013). Improved apriori algorithm based on frequency 2-item set support matrix. Computer Engineering, 39(11), 183-186.

[7] Liu, S. J., Tian-Rui, L. I., Jia, Z., & Zhu, J. (2014). Research on parallel chinese syntactic analysis based on hadoop platform. Computer Science.

## AUTHORS

**SUN Ya-ni** is with the Sichuan Information Technology College, CO 628040, China.

**CHEN Xinhua** is with the Sichuan Information Technology College, CO 628040, China.

Submitted 17 September 2015. Published as resubmitted by the authors 23 January 2016.