International Journal of Grid Distribution Computing
Vol. 8, No.4, (2015), pp. 163-170
http://dx.doi.org/10.14257/ijgdc.2015.8.4.16

Research on Parallel Algorithm Based On Hadoop Distributed Computing Platform

Guo Weiwei[1] and Liu Feng[2]

[1,2]*Heilongjiang University of Technology, Jixi 158100, china,*
[1]*gwwguoweiwei@163.com,*[2]*Liufeng8038@163.com*

Abstract

With the rapid development of the 3G network, traditional calculation methods are unable to adapt to the data scene that telecom users' Network access behavior's data scale increase rapidly dozens of TB. The cloud techniques such as Hadoop platform are introduced to solve the data storage problem. The appropriate data mining algorithms are designed from the perspective of practical application. This paper improves the traditional decision tree SPRINT algorithms, proposes a parallel computing program and successfully applies to the Hadoop platform.

Keywords: *data mining, 3G network, parallel computing, cloud computing, Hadoop platform*

1. Introduction

In the field of telecommunications, technical difficulties sprang up regarding the storage, processing and discovery of enormous data due to rapid increase of communication services and network visiting traffic [1-2]. Right at the moment, lots of new hi-tech computing service models led by cloud computing were successively proposed. They were practically applied by many telecom enterprises home and abroad for business analysis and management activities, forming the distributed mass data architecture based on computer clustering systems [3-4]. In order to get from thousands of TB-scale massive data the business knowledge which is demanded for enterprise users, it generally requires tremendous and complicated mining operations [5-6]. To achieve that goal, the cloud computing platforms were designed with hyper-scale computational capability and super strong storage capacity. The typical one is Hadoop platform, which is particularly adopted by China Unicom in our country [7-8].

The Hadoop platform allows for the development of applications in an open distributed architecture [9-10]. For data mining systems, telecom users can choose customized service and get from tremendous data resources the demanded business knowledge. Lots of home and foreign enterprises and organizations are busy with R & D work based on Hadoop platform as to build proprietary cloud platforms [11].

Sprint algorithm belongs to a traditional decision tree method. It is mainly used for classifying mining of data sets, with satisfactory scalability. Also thanks to its parallelized features, Sprint method can perform well in Hadoop platform and meet knowledge discovery requirements from the massive telecom data [12-13].

In order to tackle large scale data mining problems, it developed the parallel mining algorithm based on Hadoop platform, ensuring to discover the required knowledge in a more efficient, accurate and practical manner. Firstly it raised a plan for designing and implementing the Hadoop platform; then based on the platform to make parallelized design of traditional Sprint and carry it out; finally, test and analysis was made about the Hadoop platform.

ISSN: 2005-4262 IJGDC

2. Design and Implementation of Hadoop Platform

2.1. Basic Design Ideas. The design ideas are shown in Fig 1. To be specific, with the use of clustering computing characteristics of Hadoop, huge data mining tasks are evenly distributed to every single computing node in the clustering system to enhance efficiency and availability of explored knowledge through parallel computation; meanwhile it gives full play of Hadoop's strong data storing and processing abilities. In the lower levels, we make use of Hadoop's strengths for data storage and analysis. In the top levels, we invoke transparently relative modules with the help of functional interfaces.

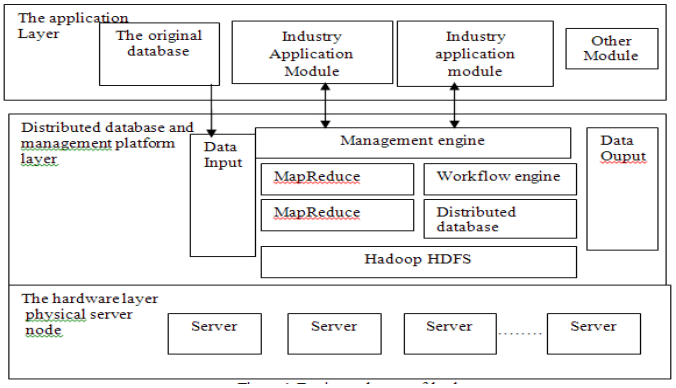

Figure 1. Design Scheme of Hadoop

2.1.1. Storage Function HDFS File System Is Chosen To Store Data Information. HDFS has API interface in multiple application levels and flexible system operation commands, providing adequate space for storing voluminous data sets, which makes it easier for data pre-processing, data computation and result output operation.

2.1.2. Computational Function. MapReduce is utilized to distribute equally data mining modules and ETL modules to all nodes in the Hadoop computational clustering system as for parallel calculation. Owing to good scalability, MapReduce is not restricted by the design of architecture in the lower levels and that users can transparently call data interfaces in top levels to complete parallelized operation. In the implementation process, we can use MapReduce to perform ETL and data mining operations.

2.2. System Structural Model. Combined with the design ideas mentioned above, we utilize hierarchical structural model to design Hadoop model from the top to the bottom in the hope of realizing the data interaction between users and cloud-based systems. Each level assumes its own task independently, which allows for extension to get good scalability. Figure 2 shows the structural model of Hadoop platform.

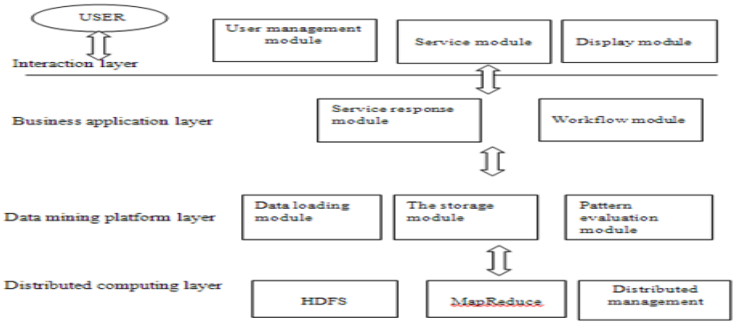

Figure 2. Architecture Model of Hadoop

Top-down analyzes the structure of the system, mainly include the following contents:

The interactive layer: providing interface of data access between system and user. Through the friendly interface, users can use a variety of customized service, and view the results feedback system.

Business application layer: defines the internal business logic, and provides the control operation and calls on all kinds of business processes. Business application layer receiving interactive layer transmits data mining tasks, and the specific mining algorithm is invoked to compute, will eventually result feedback to the interaction layer. At the same time, it is also responsible for control and call the data and close the mining flat layer of various kinds of application execution.

Data mining platform layer: providing business knowledge that the needed data mining methods for business application layer, including a variety of functional components, such as the preprocessing module, mining module, rule definition module. This layer is the core function of layer system needs to complete the operation process of the parallel algorithm, and the task allocation and then submitted to the Hadoop platform for distributed computing.

Distributed computing platform layer: in charge of platform cluster system data storage and computing, with management and distributed data processing functions of distributed file system

3. Parallelized Layout of Sprint Algorithm

As discussed before, in the hierarchical structural model of Hadoop, the kernel is data mining flat layer, which includes various parallel data mining methods. Hence, the core to building Hadoop-based data mining system is to design and implement the data mining algorithm which has parallelized computing function. Sprint is a classification mining method suitable for calculation of massive data. It has parallelizable properties. In the paper, a parallelized strategy was designed for Sprint method and was carried out.

3.1. Main Features of Sprint Method. As a decision tree classifying algorithm, Sprint is composed of two steps: tree structuring and pruning. In the first step, the method needs to perform global scanning on the database for several times. The pruning step is rather simple. The time costs less than 1% of that in the first step. Thus, more attentions should be attached to the structuring part when the method is being designed. In this regard, we chose binary tree for constructing.

In general, Sprint method describes characteristics of data sets by virtue of attribute tables and histogram. That is because when dataset is over-sized; all data can't be read into the memory. But the algorithm can store all attribute tables in hard disks and put attribute tables under process to the memory. When attribute tables are being initialized, they are arranged in order. In the subsequent breakage process, attribute lists don't need re-ordering. Therefore, the algorithm is always highly efficient. With the feature of one-time rank ordering and no limitation of memory capacity, Sprint algorithm has initial conditions for the paralleling design.

NO.	Age	Type	Risk
0	28	family	H
1	16	sports	H
2	53	sports	H
3	70	family	L
4	42	truck	L
5	19	family	H

=

NO.	Age	Risk
0	28	H
1	16	H
2	53	H
3	70	L
4	42	L
5	19	H

+

NO.	Type	Risk
0	family	H
1	sports	H
2	sports	H
3	family	L
4	truck	L
5	family	H

Figure 3. Diagram of Attribute List Generation

Figure3 shows the generation process of attribute tables. Such tables include the tree fields: attribute type, attribute value and ID record index. Different from original massive data, attribute tables can be easily stored in hard disks. When attribute tables are divided along with task allocation, the histogram can be used to make replenishment. Such bar chart depicts the distribution of category of one attribute, which has numeric type and discrete type. For the numeric attribute table, two histograms are involved, recording the distribution of category of respectively processed samples and unprocessed samples. The two histograms finally find the best split point through continuous updates and modifications. Regarding the discrete attribute table, only one histogram is needed, which is renamed statistical matrix.

Sprint marks out the best break point by estimating the Gini coefficient of attributes. Generally speaking, the division leading to the smallest Gini index will bring about the biggest information gain. Gini index is reached by the equation:

Data set T consists of a total of M records, belong to N category, the $GINI$ index for T:

$$GINI(T) = 1 - \sum_{i=1}^{N} p_i^2 \qquad (1)$$

If data set T consists of two parts: T_1 and T_2 , respectively M_1 and M_2 records preservation. Then the $GINI$ index for T:

$$GINI(T) = \frac{M_1}{M} GINI(T_1) + \frac{M_2}{M} GINI(T_2) \qquad (2)$$

Numerical attribute table after sorting will calculate the adjacency properties intermediate values, as the candidate segmentation points, for each candidate segmentation points to generate the corresponding histogram. So, when the candidate node is changed or migration will continue, histogram segmentation. When determining the minimum Gene index candidate segmentation points, is the best segmentation.

3.2. Paralleling Strategy of the Algorithm. Before determining to perform parallelized operation of Sprint method, we need to know clearly the involved sub-processes for parallelization. Based on above discussion, most time of the algorithm is spent on computing the segmentation of attribute tables. So the key point should be poised to the paralleling design of segmentation operation.

3.2.1. Node Parallelism. During the implementation of Sprint algorithm, nodes are consistently attaching to attribute tables. Once such tables are partitioned, all nodes will search for new attribute tables to attach to, which will be segmented again. Likewise, the breakup goes forward for the recursive operation of programs. In the end, the decision tree is formed. In the Hadoop platform, that is completed by relying on MapReduce, which defines different indicator functions and then maps attribute tables all onto different nodes. With the mapping function, it's possible to deal concurrently with the attribute tables to which all nodes in the same level attach. With MapReduce's Map as mapping, different <key, value> can be mapped into different Reducers (representing nodes). That is the definition of rules. Attribute lists which belong to the same node can be directly distributed to a similar Reducer. With increasing number of breaking nodes, Reducer's parallel degree gets higher and higher.

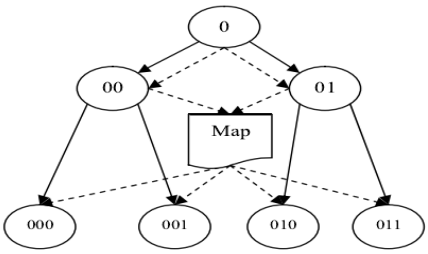

Figure 4. Diagram of Node Parallel

As indicated in Fig4, in the process of serial computation, the segmentation computation starts usually from the left to the right according to the level. That is to say 0 → 00 → 000 → 001 → 01 → 010 → 011.When designed nodes complete the parallel, the calculation mode is shifted to $0 \xrightarrow{MAP} \{00,01\} \rightarrow \{000,001,010,011\}$. That is to say, all nodes in the same level are computed concurrently, which increases greatly the speed, particularly with more in-depth level, the more nodes there're, the more obvious effects it will have.

3.2.2. Attribute Table Concurrent Design. As seen in Fig5, apart from the node parallelism, it's necessary to perform concurrent design of attribute tables. The search for split point of attribute tables can help achieve concurrency, calculating their respective Gini values. All attribute tables have a node code ID. In the paralleling step, in order to avoid attribute tables with the same node code ID from being parallelized, all such tables need dividing to one Reduce.

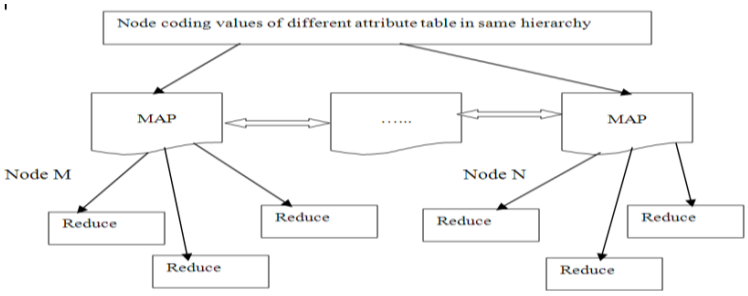

Figure 5. Diagram of Attribute List Parallel

4. Experiment Design and Discussion

4.1. Analysis of rationality. In the design process of the proposed parallel algorithm, by depending on Hadoop platform, the workflow of Sprint algorithm was designed completely and that the concurrent Sprint algorithm was successfully migrated to Hadoop platform. There are following steps:

In different phases, it's important to keep the parallelized design in the Map process as it affects directly the file processing speed and size of data blocks. Moreover, most time of the algorithm was consumed in the process. But between Reduce and separation process, there is certain inheritance relationship. The paralleling degree is not high so that it's no need to implement specific parallelization design.

In the parallelized design process, a few Maps are concurrently read in. Then, data are passed to Reduce. For the attribute of continuous values, which are all sequenced for processing, it's no need for Sprint algorithm to make global ordering at big costs. For giant data volume, the method's paralleling advantage will more observable.

The design of parallel algorithm takes into full account the own features and superiority of enormous computation of Hadoop platform. Together with the computational power of the framework, the scalability and efficiency of the proposed method are both ensured to the maximum limit.

4.2. Analysis of Effectiveness. From the experimentation above, we can transfer all data sets to attribute lists according to the expression pattern of trees. New attribute tables can be created by binding such tables with corresponding nodes. Then according to the hierarchical structure of trees, mutually separate folders are formed for storing attribute tables and node information in each level.

This experiment shares some nodes, it as shown in figure6.

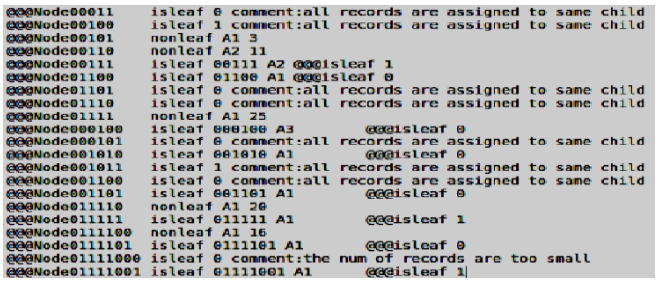

Figure 6. Diagram of Some Nodes Storage

As mentioned previously, numbers after "@@@" refer to node's code; "isleaf" and "noleaf" mean separately leaf nodes and non-leaf nodes; the marker "0" or "1" after leaf nodes represents that: the customer will or will not have overdue bills in May 2015; "A1, A2, A3, A4" corresponds respectively to the pre-defined variable R, F, M and T. As a result, the decision tree is constructed regarding the arrearage problem.

Here we use multi-set sample crossing way to test the accuracy of the method. In the experiment, we divide randomly the initial data set into five subsets of equal size, with no overlapping contents of those sets. For the training each time, one set is selected randomly as test set. All results are listed in Table 1.

Table 1. Descriptions of Data Crossing Tests

	Total number of elements	Number of correct	Number of error
Subset 1	92004	70301	21703
Subset 2	111800	86030	25770
Subset 3	96735	74047	22688
Subset 4	181722	156032	25690
Subset 5	134120	104853	29267

According to the above data, the accuracy can be obtained for the algorithm:

$$\frac{82301+102830+89947+171032+124853}{92004+111800+96735+181722+134120}=\frac{570963}{616381}=79.7\%$$

From the experimental results, the model of decision tree generation algorithm is effective, an acceptable accuracy was obtained.

4.3. Analysis of accuracy. With view to validating more effectively the accuracy of the parallel algorithm, we compare the proposed method and the data mining method in [14-15]. When choosing test data sets, we use the same crossing way. The data source is in itself objective and true, extracted from user defaulting behavior records data in the actual telecom business operation process.

The fundamental ideas for implementing the two methods are described as follows:

The literature [14] proposed a method based on rule sets, i.e. creating rule classes of multiple attributes. Before the data mining method is carried out, all data are extensively incorporated into each rule; then, based on the difference and importance of each rule, the processing sequence, parallel mining pattern and specific mining workflow are determined.

The work [15] suggested a data mining method based on rough sets. The point is such method is not feasible. To highlight various features of the proposed parallel algorithm, we intentionally chose it for the accuracy comparison test. With the introduction of rough sets, the method showed desirable fitness for the discovery of huge data. That is accepted by scholars in the field. Besides, from the actual data test, the method proved higher accuracy. The accuracy comparison results are seen in Table2.

Table 2. Results of Accuracy Tests for Three Algorithms

	Total number of elements	Correct number of this algorithm	Number correct of[14]	Number correct of[15]
Subset 1	119383	83892	88343	77598
Subset 2	149302	120933	107496	119441
Subset 3	160492	131603	117159	110739
Subset 4	93820	70365	73179	63797
Subset 5	93384	73773	65368	63501

From the above data, we can conclude that the accuracy rate of three methods is 77.96%, 73.25% and 70.59% respectively.

In terms of the accuracy rate, the proposed method demonstrated stability of concurrent work and excellent comprehensive process performance owing to the parallelized features and the multilayer-oriented scheduling and working mechanism of Hadoop platform. More importantly, the proposed method considered fully all influencing factors in each processing stages of the whole parallelized process. For data processing, segmentation of attribute tables and the final result output, it designed and improved the paralleling strategy, enabling it to acquire a higher accuracy ratio.

The method based on rule sets has some universality, which may require for further test. Only judged from the test result and its implementation way, the ideas based on rule features and clustering is the future concern about improving the proposed method. The method showed superior adaptability to the subset (2). After analysis of data in subset (2), we found if there are more elements used as data of leaf nodes, its attribute data and classifying information become relatively evident, meeting quite well the computational requirements of methods based on rule sets.

The method based on rough sets performed pretty well. As a kind of method with no support of parallelized strategy, to achieve higher accuracy, it must rely on its own enough strong processing mechanism and fault tolerance. In this case, the own features of rough sets make the method maintain certain error rate. From the statistical results, in the five crossing test process, we observed a higher error rate of the method in the subset (2). We believe with rapid growing data volume and deeper mining operations, its fault tolerant mechanism will degrade a lot and the error rate hoists remarkably.

5. Conclusion

This paper firstly analyzes and introduces the framework and designs model of Hadoop platform, expounds the characteristics of cloud data mining platform, on the basis of this, the traditional SPRINT parallelize algorithms have been successfully designed. The algorithms are carried out experimental verification and performance analysis on the Hadoop platform. The algorithm laid the foundation for data traditional parallel mining algorithm.

References

[1] Z. Yanan, "Research on parallel clustering algorithm Hadoop based on cloud computing platform", Inner Mongolia University of Science and Technology, **(2013)**.
[2] G. J. Guang, "Research on key technology of data mining for telecom field", Harbin Engineering University, **(2012)**.
[3] C. Aiping, "Analysis and Application Research on parallel clustering algorithm based on Hadoop", The University of Electronic Science and technology, **(2012)**.
[4] X. Jun, G. Yang, S. Lin and Y. Yubin, "Parallel text classification based on Hadoop platform", Computer science, vol. 10, **(2011)**, pp. 184-188.
[5] L. Shile,." Realization of Hadoop platform and the random forest algorithm research based on image classification system", Xiamen University, **(2014)**.
[6] H. Li and G. Jun, "parallel data classification algorithm based on Hadoop platform", the manufacturing industry automation, vol. 14, **(2014)**, pp. 5-9.
[7] C. Yong, "Design and implementation of algorithm of distributed data query communication platform based on Hadoop", **(2009)**, Beijing Jiaotong University.
[8] H. Xiaofei, T. Yuesheng and W. Jingyu, "On the platform of Hadoop parallel algorithm research and implementation of Apriori", Computer and modernization, vol. 03, **(2013)**, pp. 1-4.
[9] W. Qiuwen, "Design and implementation of system for whole genome association studies based on Hadoop", **(2012)**, Tianjin University.
[10] Z. Tao, "Research and implementation of network text analysis based on Hadoop", **(2014)**, Huazhong Normal University.
[11] Z. Caihui, "Research on the platform of Hadoop adaptive local hyperplane based on K nearest neighbor algorithm", South China University of Technology, **(2011)**.
[12] P. Tianming, "Research on parallel decision tree algorithm based on the Hadoop platform", **(2012)**, East China Normal University.
[13] S. Yuan, "Huang Gang. Analysis and research of C4.5 algorithm based on Hadoop platform", Computer technology and development, vol. 11, **(2014)**, pp. 83-86.
[14] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases", Proceedings of the ACM SIGMOD Conference on Management of data, **(1993)**, pp. 207-216.
[15] I. Foster, C. Kesselman, "The Grid, Blueprint for New Computing Infrastructure", San Francisco, CA, USA, Morgan Kaufmann Publishers, **(1999)**, pp. 290-292.

Authors

Guo WeiWei, She received her B.S degree in computer Science from Heilongjiang University of Science and Technology. She received her M.S degree in computer Science from Liaoning Technical University. She is an Associate professor in Heilongjiang University of Technology. Her research interests include database and algorithm.

Liu Feng, He received his B.S degree in computer Science from Heilongjiang University of Science and Technology. He received his M.S degree in computer Science from Liaoning Technical University. He is an Associate professor in Heilongjiang University of Technology. His research interests include database and algorithm.

