Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 82-87
doi:10.4028/www.scientific.net/AEF.6-7.82
© 2012 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.

Research on Clustering Analysis of Big Data

Yuan Yuanming[1, 2, a], Wu Chanle[1, 2]

[1] Computer School of Wuhan University, Wuhan 430072, China

[2] National Engineering Research Center for Multimedia Software, Wuhan 430079, China

[a]yymwyy@sina.com

Keywords: Big Data, Clustering, MapReduce

Abstract. Data quantity of Big Data was too big to be processed with traditional clustering analysis technologies. Time consuming was long, problem of computability existed with traditional technologies. Having analyzed on k-means clustering algorithm, a new algorithm was proposed. Parallelizing part of k-means was found. The algorithm was improved with the method of redesigning flow with MapReduce framework. Problems mentioned above were solved. Experiments show that new algorithm is feasible and effective.

Introduction

In the modern network environment, data proliferates fast and its resource is wide. Therefore, a kind of special data group, called Big Data, appears. Due to the big amount of Big Data, its sustainable growth and high data latitude, traditional data process and analyzing techniques are faced with new challenges, like calculable problems. When processing massive data, traditional data process and analyzing techniques are faced with some problems. They are long time consuming, limited performance when processing on single machine, and being unable to efficiently process Big Data.

As a kind of data mining technology, clustering analysis can classify data, which is widely used in Web document classification, customer classification, biological population study, and spatial data analysis. When analyzing Big Data with clustering algorithm, calculable problem still exits.

Cloud computing has the retractable distributed computing ability. Distributed file system is suitable for distributed storage of Big Data. This paper is based on distributed file storage technology, and studies on the clustering analysis of Big Data, using Map/Reduce [1] calculation model as the solution of distributed computing.

Techniques Related to Cloud Computing

This paper is based on the Hadoop platform [2]. Hadoop is a distributed computing framework of Apache Open Resource Organization, which is a cloud computing framework inspired by MapReduce of Google and Google File System. Hadoop can run in cheap hardware cluster, and this paper is mainly based on Map its HDFS and MapReduce.

HDFS [3] is a distributed file system, and its framework is the typical master-slave mode, as shown in figure 1. There is a NameNode, more than zero Secondary NameNode, and several DataNode. NameNode manages the metadata of file system, that is, it manages how data distributes in each DataNode, which is responsible for data storage.

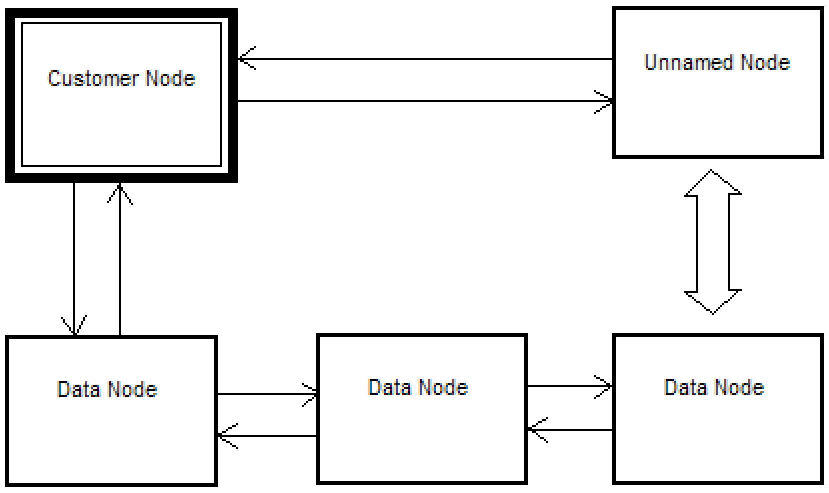

Fig. 1 HDFS framework

First, client gets file metadata from NameNode, and then interacts with DataNode to accomplish I/O. HDFS also implements the redundant backup between different DataNodes and heartbeat detection to guarantee the reliability and consistency of data storage.

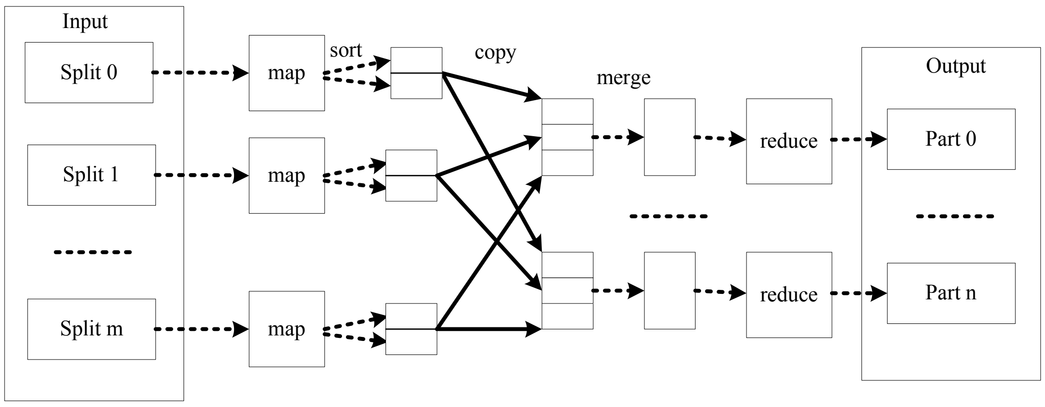

Fig. 2 MapReduce computing model

MapReduce is a distributed computing model, as shown in figure 2. MapReduce accomplishes the internal control process of the model, and abstracts any complicated parallel computing process into the task chain constituted of two functions of Map and Reduce. Map function is responsible for the processing of the key value pairs input and the intermediate results are also in form of key value pairs. Reduce function is responsible for combining all the intermediate results by key, protocol treatment and the final result. The computing model is simple so that it is unnecessary to care about the implementation details of distributed computing, but to implement the process of Map and Reduce to handle complicated problems.

Hadoop firstly break the dataset input by tasks of MapReduce into data blocks in moderate sizes. Hadoop will create a map task, which will run a map program written to solve specific problem in a data block, for every data block.

MapReduce computing model has another function, the combiner function, which is designed to reduce the data quantity transmitted from map task to reduce task. The combiner function can accomplish partial tasks similar to reduce tasks at the map point, and then transmit the middle data. Then combiner function absolutely cannot replace the reduce function, because it can only do some partial computing, and the global computing must be completed by the reduce function receiving different results produced by map function.

Clustering Analysis

Clustering analysis is to divide the given dataset into k clusters. Inside every cluster, data have close relation. And the relation between clusters is relatively loose.

K-means algorithm [4] is a widely used clustering algorithm. The algorithm divides n objects into k clusters, so that there is high similarity inside a cluster and high dissimilarity degree between clusters. Similarity is the result of calculating the average value of objects in every cluster.

The algorithm firstly randomly selects or appoints k objects as the default average value or center of a cluster. For the left objects, everyone will be classified into the cluster, which has the minimum distance to the object. Then the average value of every cluster will be recalculated. The process will be repeated until the criterion function converges. And the criterion function is:

$$E=\sum_{i=1}^{k}\sum_{x\in C_i}\left|x-\overline{x_i}\right|^2$$

The E in the function is the sum of the square error of all objects in the dataset. The X is the point in space, which stands for the given data object. $\overline{x}_i$ is the average value of the c_i cluster, which is also called cluster center.

The process of the K-means algorithm is as follows.

(1) Randomly select or appoint k objects from n objects as the cluster center of the k cluster;

(2) Calculate the distances of every point to the cluster center, and add the point, which has the minimum distance, to the Cluster center;

(3) Recalculate every cluster center;

(4) Repeat step 2 and step 3 until every cluster center is stable in a special range, or it reaches the maximum iteration number.

The time complexity of the algorithm is O(nkt), in which n stands for the number of the cluster points, k stands for the number of clusters, and t stands for the iteration number.

Good locality of K-means makes it easier for parallelization. In the first stage, the process of producing cluster can be parallelized, and each local node reads the dataset stored locally. Use the K-means algorithm to generate cluster set, and use several cluster sets to generate the global cluster set for the first iteration. Repeat the above steps until it satisfies the termination condition. In the second stage, cluster the clusters formed from the first stage.

Distributed K-Means Alogrithm

Based on the analysis of K-means, this paper bases on HDFS and improve the algorithm, using the MapReduce technology. Therefore, the algorithm has the distributed computing ability.

When distributed improving the algorithm, the following factors must be considered.

(1) Try to improve the complicated computing task with distributed method. There are three tasks taking more time. The first one is calculating the distances from every object to every cluster center in the sample. The second one is updating the average value of clusters. And the third one is calculating the criterion function. The time complexity of the first and third is O(nkd), and the time complexity of the second is O(nd). The improvement of the three computing tasks will improve the performance of the entire computing task.

(2) Reduce the quantity of middle data transmission. In most cases, the time consumption for mapper to transmit data to reducer is much larger than it for computing. We can use the combiner function to parallelize the output of mapper, and sort and transmit inside. Therefore, the quantity of data transmission is reduced.

(3) Data partitioning. Big Data is stored in HDFS. And special rules are installed inside HDFS to divide data. Data blocks are distributed to data nodes. The default size of the database is 64k. It needs adjustment in real operation according to the amount of data.

To adapt to MapReduce model, we adjust the K-means algorithm as follows:

(1) pick k objects from n objects as cluster centers;
(2) calculate the distances from each object to all c luster centers, and assign the object to the nearest cluster;
(3) merge intermediate data;
(4) record all objects of each cluster, and compute each cluster centers;
(5) calculate squares of difference between each object and cluster centers;
(6) calculate criterion function E;
(7) repeat steps 2 ~ 6 until Ei+1 – Ei <ε(ε is a very small positive number).

Steps 2-6 are implemented by using MapReduce model through two iterations. Steps 2-4 are implemented in the first iteration.

In the first iteration, each Mapper reads native local object. Then it calculates the distances from the object to all cluster centers and assigns the object to the nearest cluster. The format of output data is <key, value>. Key represents cluster ID, and value represents object ID.

Combiner merges and computes the output data of Mapper, and then outputs <key, value_list>. Key is cluster ID, and value_list is a list of objects in the cluster. After having been processed by Combiner, the amount of data transferring among nodes is reduced. Combiner implements step 3.

The number of Reducers is K. Each Reducer corresponds to a Cluster. Reducer gets the intact objects of the Cluster, then computes the averages to form new center. It implements step 4. The format of output data is <key, <cluster, b>>. Key is object ID, cluster is cluster ID, and b is cluster center.

After the first iteration, we get 4 divisions and compute their center. After the second iteration, we calculate criterion function.

In the second iteration, Mapper reads the output of previous Reducer, then calculates square of difference between each object and cluster centers. The format of output data is <key, value >. Key is object ID, value is square of difference between each object and cluster centers. It implements step 5.

There is only one Reducer in the second iteration. Result of every object's calculation is brought here to sum, which come to criterion function. It implements step 6.

During the two iterations, we achieve distributed computing according to data distribution.

Experiment

The experimental environment is composed of 50 PCs installed Hadoop 0.20.2 in Linux.

Our experimental data comes from UCI (University of California, Irvine Machine Learning Repository)[5]. We choose dataset Synthetic Control Chart Time Series (D1 for short), Iris (D2 for short) and Glass Identification (D3 for short) from UCI. D1 includes 6000 examples, divided into 6 groups, 1000 examples each group. Each example has 60 attributes. D2 includes 1500 examples, divided into 3 groups, 500 examples each group. Each example has 4 attributes. D3 includes 2140 examples, divided into 6 groups. Each example has 9 attributes.

First we set termination condition of iteration to $E_c(i+1) - E_c(i) < 1e-6$.

Then we use k-means to test many times on single and more nodes, and pick accuracy rate, minimum square error and average value. The results are shown in Table 1, Table 2 and Table3.

Table 1 Running result on D1

Number of Nodes	Accuracy	Emin
10	80%	889310
30	79%	852261
50	81%	879273

Table 2 D2Running result on D2

Number of Nodes	Accuracy	Emin
10	63%	79
30	62%	77
50	66%	81

Table 3 Running result on D3

Number of Nodes	Accuracy	Emin
10	55%	423
30	54%	436
50	57%	433

In Table 1, Table 2 and Table 3, Emin is the minimum sum of square errors from clusters through iterations. Accuracy rate is the percentage of the correctly recognized objects from the total objects. It is seen in Table 1, Table 2 and Table 3: the cluster results' qualities on single and more nodes are the same in terms of accuracy rate and minimum square error.

Speedup is an important indicator of verifying performance of parallel algorithms, which describes the ratio of the run time of single node algorithm to the run time of parallel algorithm. Speedup $S=T_S/T_P$. T_S is the time needed of the algorithm running on single node. T_P is the time needed of the algorithm parallel running on p nodes. To demonstrate the advantage of parallel algorithm k-means based on MapReduce in processing Big Data, we randomly generate four Big Data sets referring to D1. Their sample sizes are 5000, 10000, 200000 and 50000 and they are called D11, D12, D13 and D14. Seen in Figure 3, parallel algorithm k-means based on MapReduce has a higher speedup. The rate of growth of speedup tends to decline as the number of nodes grows. The reason is that the amount of data transferred grows as the number of nodes grows.

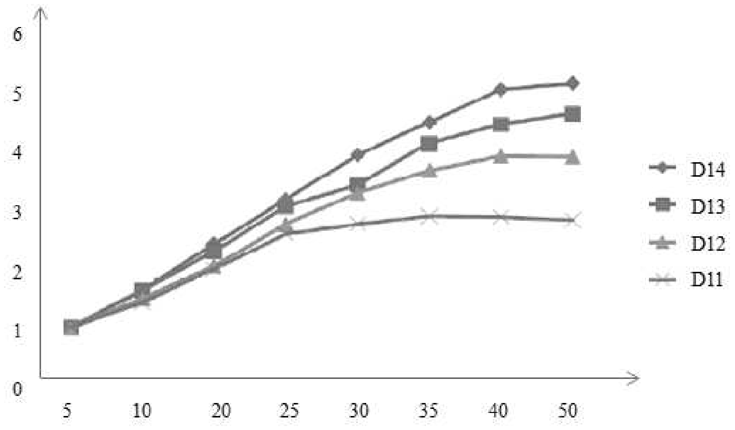

Fig. 3 Curve of speedup

Efficiency of parallel algorithm is the ratio of speedup to number of child nodes, i.e. E=S/P. S is speedup and P is number of nodes. As seen in Figure 4, the efficiency tends to decline. The larger the data set is, the smoother the curve of efficiency will be. When the data set is smaller, the efficiency descends quickly beyond a certain number of nodes. Therefore, this algorithm's scalability is better on large data set.

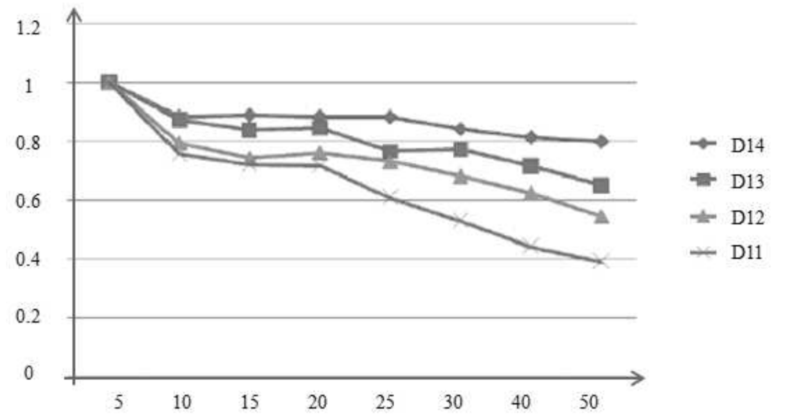

Fig. 4 Curve of algorithm efficiency

Conclusion

We have implemented division and distributed storage of Big Data in Hadoop environment. We have rewritten algorithm k-means and implemented distributed computation of cluster by using MapReduce computing framework. First of all, we have solved the computability problems of data. So it will not be incalculable because of hardware limitations on single machine. Hadoop has good scalability. It can scale computing resources up based on demand and compute no matter how large the data is. In the second place, the efficiency is higher. With complicated task having been divided, every computing node only needs to complete simple computing task. It is efficient and requirement for hardware of every node is low.

In general, it takes much time to do a cluster computing on Big Data. If the data keeps increasing, recalculating is needed as long as new data comes, which is high frequent and high pressured. In the future, the idea of incremental classification can be taken into account. When new data comes, classification is only done on new data, which reduces data size and then saves computing time.

References

[1] Ralf Lammel, Data Programmability Team.Google's MapReduce Programmig Model-Revisited. Redmond, WA, USA: Microsoft Corp. 2007.
[2] Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters, Communications of the ACM, vol.51, no .1(2008), pp .107-113.
[3] Hadoop Community. Hadoop Distributed File System, http://hadoop.apache.org/hdfs 2010.
[4] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28, No. 1 (1979), pp. 100-108.