ISSN:0975-9646

Parallel Implementation of Fuzzy Clustering Algorithm Based on MapReduce Computing Model of Hadoop –A Detailed Survey

Jerril Mathson Mathew
M.Tech Student
College of Engineering Kidangoor
Kerala, India

Lekshmy P Chandran
Assistant Professor
College of Engineering Kidangoor
Kerala, India

***Abstract* -Clustering is regarded as one of the significant task in data mining which deals with primarily grouping of similar data. To cluster large data is a point of apprehension. Hadoop is a software framework which deals with distributed processing of huge amount of data across clusters of distributed computers using MapReduce programming model. MapReduce allows a kind of parallelization to solve a problem that involves large datasets using computing clusters and is also a striking implication for data clustering involving large datasets. Mahout, a scalable machine learning library is an approach to Fuzzy clustering which runs on Hadoop. This paper focuses on studying the performance of using Fuzzy K-mean clustering in MapReduce on Hadoop.**

Keywords- Fuzzy C Means Clustering (FCM), MapReduce, Hadoop, HDFS, Mahout, Parallel Computing.

I. INTRODUCTION

With mature database technologies and universal data applications, business enterprises, research institutions and government departments have agglomerated a large amount of data stored in different forms. How to hoard and handle these enormous collections of data, as well as further dig out useful knowledge which can lead the applications has become a problematic issue.

Data mining is also known as knowledge discovery in databases. Data mining extracts mysterious probable valuable information or pattern from large, incomplete, noisy, blur, random data. With the hasty development of computer technology and the popularity of the network, people have more opportunities to use suitable way to barter information with the outside world. However, the influx of large amounts of data increases the difficulty of obtaining useful information. How to obtain valuable information from large amounts of data brings problems of implementing data mining structure. Due to the high complication of processing these data, the computing power of the system is difficult to meet the requirements. At this point, the limited computing resources which traditional stand-alone server can offer regularly cannot meet the desires. There need distributed computing technology to achieve large-scale parallel computing.

Data clustering is a key research area in the field of data mining. Data clustering analyses the data and finds useful information. Based on the philosophy of “Like attracts like”, the so-called data clustering is a process which divides the collection of physical or abstract objects into multiple classes or clusters. A cluster is a compilation of data objects. Data objects in the same cluster (group) are as similar as possible. However, data objects from different clusters are as different as possible. By clustering, one can identify dense and sparse areas and find an interesting correspondence between the overall allocation pattern and data attributes. The k-means algorithm belongs to a basic division technique of clustering analysis.

In the appearance of gigantic data, existing clustering algorithms have encountered the blockage in time and space complexity. This is one of the questions needed to be solved urgently in the field of clustering algorithms. A proposal to unravel this problem is to apply the parallel processing technology to data clustering, design proficient parallel clustering algorithms, and progress the performance of data clustering algorithms handling massive amounts of data.

Cloud computing has got widespread attention as an promising business model. [5] Hadoop is a cloud computing platform which can more easily develop and process large data in parallel. [6] Its main features include strong development capacity, low expenditure, high effectiveness and good trustworthiness. Hadoop platform consists of two parts: Hadoop Distributed File System (HDFS) and MapReduce computing model. [7] On the basis of cloud computing platform Hadoop, the paper depicts a parallel k-means clustering algorithm based on MapReduce computing model.

The K-mean algorithm faces a problem of giving a hard partitioning of the data which means that each point is dedicated to one and only one cluster. The data points on the edge of the cluster as well as lying near another cluster may not be as much in the cluster as the points in the center of cluster. Hence, Fuzzy K-mean clustering [1] (also known as Fuzzy C-means clustering) given by Bezdek introduced that each point has a probability of belonging to a certain cluster. A coefficient value associated with every point gives the degree of being in the k^{th} cluster and coefficient values should sum to one. Nowadays larger datasets are considered for clustering which do not even fit into main memory.

Apache Hadoop[2,4] was born to solve the problems pertaining to large datasets. With the help of MapReduce, Hadoop fires a query on the large datasets, divide it and then runs it in parallel on multiple nodes. Mahout[3] is a scalable machine learning library which is built on Hadoop and is usually written in Java.

The paper later explains about the literature survey in the Section II. The Section III gives the descriptions of the methodology. Finally we conclude with Section IV.

II. LITERATURE SURVEY

Data set X={x_1, x_2, …, x_i, …, x_n} contains n d-dimensional data points. Each data point belongs to d-dimensional data space, which is defined as $x_i \in R_d$. The variable k is the number of data subsets to be generated. The k-means clustering algorithm arranges the data objects into k divisions, which is defined as C={c_k, k=1, 2, …,K}. Each division represents a cluster c_k. Each cluster has a cluster center μ_k. Select the Euclidean distance as the similarity criterion, and calculate the square of the distance between the points in the cluster and the cluster center. The square of the distance is defined as:

$$J(c_k) = \sum_{x \epsilon c_k} \|x_i - \mu_k\|^2$$

The goal of clustering makes the minimum of the total distance of all clusters. The entire distance of all clusters is defined as:

$$J(C) = \sum_{k=1}^{K} J(c_k)$$

$$J(C) = \sum_{k=1}^{K} J(c_k) = \sum_{k=1}^{K} \sum_{x_i \epsilon c_k} \|x_i - \mu_k\|^2$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{n} d_{ki} \|x_i - \mu_k\|^2$$

$$d_{ki} = \begin{cases} 1, if\ x_i \in c_i \\ 0, if\ x_i \notin c_i \end{cases}$$

Apparently according to the least squares method and the Lagrange principle, the cluster center μ_k is the average of data points in the cluster c_k. The k-means clustering algorithm begins from initial k categories, and assigns each data point to each category in order to diminish the total sum of squared distances. With the increase of the number of the categories, the total sum of squared distances tends to decrease. When k is equal to n, J(C) is 0. Therefore, the total sum of squared distances obtains the least only under the determined number k.

Paper [9] presented a clustering algorithm based on MPI (Message Passing Interface). Because MPI uses the way of inter-process communication to harmonize parallel computing, this leads to lower parallel efficiency, memory overhead. Paper [10] presented the parallelization of a k-means algorithm based on PVM (Parallel Virtual Machine). However the algorithm is limited by the system and lacks flexibility. Paper [11] used multi-core CPU platform to improve the clustering speed. Paper [12] presented a fast clustering algorithm based on GPU (Graphics Processing Unit). Paper [13] presented a capable parallel clustering algorithm in a top-performance cluster environment. These solutions which paper [11-13] presented are based on expensive high-performance hardware. These solutions cannot make large-scale promotion. Paper [14] presented a clustering algorithm by means of data and task parallelism. The disadvantage is that communication overhead between nodes is high.

III. METHODOLOGY

Clustering, an unsupervised learning technique groups the samples having similarity in different classes. The groups or classes are referred to as clusters. Samples within a class are of high similarity as compared to the samples of other classes. It is learning by observation process which is mainly used in the areas like data mining, machine learning and statistics. Fuzzy K-mean clustering is based on centroid based clustering technique.

A. *Hadoop Platform*

By Hadoop, an open source framework implementing the MapReduce programming model includes two components namely the Hadoop Distributed File System (HDFS)[4] and MapReduce. HDFS is used for storage of large dataset and MapReduce is used for processing the datasets. In HDFS, the file is split in contiguous chunks each of size 64MB (default block size)[4] and each of these chunk is replicated in different racks. The NameNode in HDFS stores the metadata and the DataNodes stores the blocks from files. Associated with the NameNode and the DataNode is the daemon known as the JobTracker and the TaskTracker respectively. It is the duty of the JobTracker to assign the jobs to the TaskTracker which then processes each of the jobs assigned to it using the MapReduce model. Hadoop, a distributed file system is written in Java.

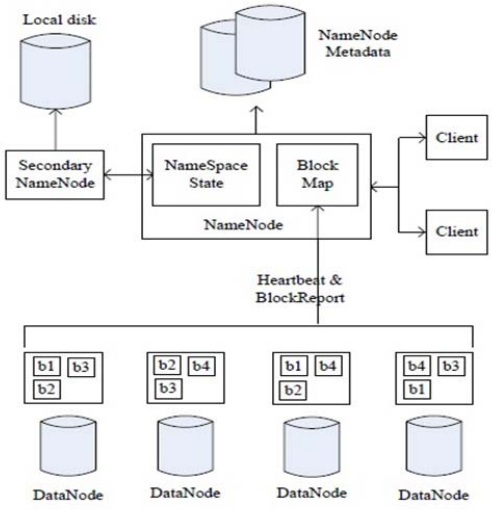

Fig.1. Architecture diagram of HDFS

B. *MapReduce Technique*

There are mainly two programs in MapReduce[8], one is the Map and another is Reduce. Dataset is split according to block size of Hadoop. Map( ) function is associated with each block and considers the input pair in the form of a key and value and then processes the input pair thereby generating an intermediate set of <key, value> pairs. The function of Reduce( ) aggregates the intermediate results and generates the final output. Like HDFS, MapReduce of Hadoop also adopts Master / Slave architecture, particularly as shown in Fig 2.

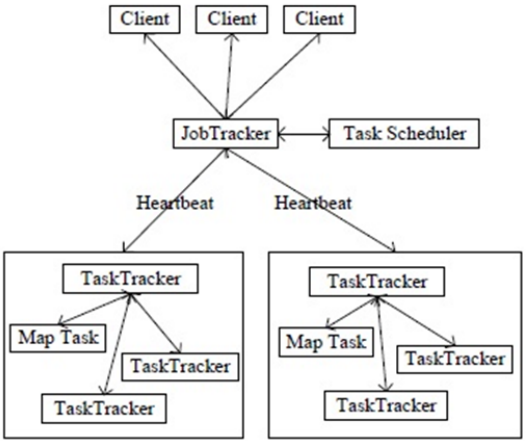

Fig.2. Architecture diagram of MapReduce of Hadoop

MapReduce is a programming paradigm that expresses a bulky distributed computation as a sequence of distributed operations on data sets of key/value pairs. The MapReduce framework of Hadoop harnesses a cluster of machines and executes user defined MapReduce jobs athwart the nodes in the cluster. MapReduce is composed by JobTrackers and TaskTrackers. A MapReduce working out has two phases: a map phase and a reduce phase. The input to the computation is a data set of key/value pairs.

In the map phase, the architecture splits the input data set into a large number of fragments and allocates each fragment to a map task. The framework also distributes the many map tasks across the cluster of nodes on which it operates. Each map task guzzles key/value pairs from its assigned fragment and produces a set of intermediate key/value pairs. For each input key/value pair (k, v), the map task invokes a user defined map function that transmutes the input into a diverse key/value pair (k', v').

Following the map phase the skeleton arranges the intermediate data set by key and produces a set of (k', v') tuples so that all the values coupled with a specifickey appear together. It also partitions the set of tuples into a number of fragments equivalent to the number of reduce tasks. In the reduce phase, each reduce task consumes the fragment of (k', v') tuples assigned to it. For each such tuple it calls a user-defined reduce function that transmutes the tuple into an output key/value pair (k, v). Once again, the skeleton distributes the many reduce tasks across the cluster of nodes and deals with distributing the appropriate fragment of intermediate data to each reduce task.

Tasks in each phase are put to death in a fault-tolerant manner. If node(s) fail in the middle of a computation the tasks allocated to them are re-distributed among the outstanding nodes. Having many map and reduce tasks enables good load balancing and permits failed tasks to be re-run with small runtime overhead. Fig. 3 shows MapReduce computing model.

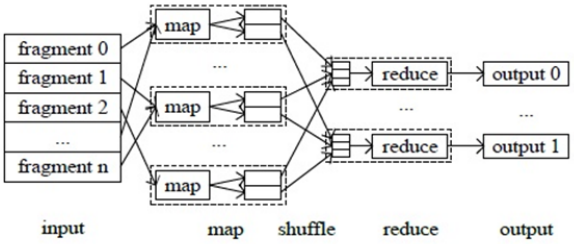

Fig.3. MapReduce Computing Model

C. *Mahout*

Apache Mahout [3] is a scalable machine learning library which involves clustering, classification and collaborative filtering employed on top of Apache Hadoop using the MapReduce paradigm and is also written in Java. There are various subcategories of machine learning such as unsupervised learning, supervised learning, semi-supervised learning, learning to learn and reinforcement learning.

D. *Fuzzy K-Mean Clustering*

The fuzzy K-mean clustering, also known as soft clustering is an extension of K-mean clustering. It minimizes the intra-cluster variance. Bezdek introduced the concept of fuzziness parameter (m) in Fuzzy K-mean clustering which determines the degree of fuzziness in the clusters [1]. The algorithm of standard Fuzzy K-mean clustering algorithm is as follows:

1. Choose a number of clusters
2. Create distance matrix from a point x_j to each of the cluster centers considering the Euclidean distance between the point and the cluster center using the formula:

$$d_{ij} = \sqrt{\sum(x_j - c_i)^2} \quad (1)$$

where,

d_{ij}= Euclidean distance between the j^{th} data point and the i^{th} cluster center

3. The membership matrix is created using:

$$\mu_i(x_j) = \frac{\left(\frac{1}{d_{ij}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{p}\left(\frac{1}{d_{kj}}\right)^{\frac{1}{m-1}}} \quad (2)$$

where,

$\mu_i(x_j)$ = is the membership of x_j in the i^{th} cluster

m= fuzziness parameter

p= number of specified clusters
d_{kj}= distance of x_j in cluster C_k

For a point in a sample, the total membership must sum to 1.The value of m is kept generally greater than 1 because if it is kept equal to 1, then it resembles K-mean clustering algorithm.

4. The new centroid for each cluster is generated as:

$$C_i = \frac{\sum_i [\mu_i(x_j)]^m x_j}{\sum_i [\mu_i(x_j)]^m} \quad (3)$$

Stopping criteria: - The algorithm continues until any centers of the clusters do not change beyond the convergence threshold and neither the points change in the assigned cluster.

The limitation of this iterative algorithm is that number of iterations is increased for forming overlapping clusters thereby increasing the execution time and if large dataset is used then it becomes difficult to handle in main memory. Hence to overcome this problem, MapReduce approach is used.

MapReduce Approach

MapReduce approach partitions the large datasets and then computes on the partitioned dataset (known as jobs) in a parallel manner where the individual jobs are processed by the maps and then the sorted output from the maps are processed by the reduce.

Input: Data points, randomly selected centroid points, number of clusters.

Output: Final centroids and their clustered points.

Algorithm of Map:

1. The randomly selected centroid point is considered as key and vector points as values.
2. Calculate the Euclidean distance between centroid point and the vector point using (1)
3. Compute the membership value of each vector point and create the membership matrix using (2)
4. Clusters are generated using nearest centroid and the data points assigned to that particular cluster
5. Maintains a cache holding the detail about which vector point is in which cluster.

Algorithm of Reduce:

1. Recalculates the centroid for each cluster

The recalculated centroid would go serially to Map and after that as it iterates, the work would be done in parallel until the centroid converged as depicted in figure 1. Total no. of reducers are less than total no. of mappers(M< N).

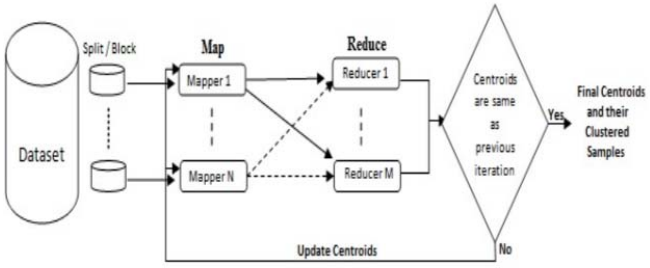

Fig.4. Fuzzy K-Mean Clustering Algorithm in MapReduce

IV. CONCLUSION

This paper conducts in-depth research on the parallel fuzzy algorithm based on MapReduce computing platform of Hadoop. First, briefly describe the basic components of Hadoop platform including structural relationships of HDFS framework and the workflow of all stages of MapReduce. Then, consider the main issues, the main processes in the design of the parallel fuzzy k-means algorithm based on Hadoop..

With the rise of cloud computing concepts, the research on data mining and clustering algorithms based on cloud computing platform gradually becomes a hot topic of scholars. Future research directions include the following:

Study general law of the parallelization of clustering algorithms. Find the relation between data scale and the number of nodes. Find influencing factors of speedup and scalability to design highly efficient parallel clustering algorithm. Study information security and privacy issues based on data mining applications in the cloud computing platform. Solving the problem will play a key role in cloud computing applications in the actual business.

ACKNOWLEDGMENT

I would like to extend my sincere gratitude to my guide Ms. Lekshmy P Chandran, Assistant Professor, Information Technology Department, CEK and Ms. Anitha R, HOD, Computer Science Department, CEK for their valuable suggestions and guidance which helped to improve the paper's quality.

REFERENCES

[1] Bezdek, James C, "FCM : THE FUZZY c-MEANS CLUSTERING ALGORITHM", vol. 10, pp191-203, 1984
[2] Hadoop official site, http://hadoop.apache.org/core/.
[3] https://mahout.apache.org/
[4] Shvachko, K.; Hairong Kuang; Radia, S.; Chansler, R., "The Hadoop Distributed File System," *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on* , vol., no., pp.1,10, 3-7 May 2010
[5] Armbrust M, Fox A.(2009) Above the clouds: a Berkeley view of cloud computing. *University of California at Berkeley.*
[6] Tom White.(2012) Hadoop: The Definitive Guide Third Edition. *O'Reilly Media.*

[7] Dean J, Ghemawat S.(2008) MapReduce: simplified data processing on large clusters. *Communications of the ACM,* 51(1*)*, p.p.07-113.

[8] Ghemawat, H. Gobioff, S. Leung. "The Google file system," *In Proc.of ACM Symposium on Operating Systems Principles, Lake George, NY* , pp 29–43, Oct 2003

[9] Zhao Zongtang, Sun Shenli, Fan Ji (2005) Parallel clustering algorithm based on MPI. *Journal of Zhengzhou Institute of Aeronautical Industry Management*, 24(3), p.p.160-171.

[10] Mao Jiali (2003) *K-means Algorithm and Parallelization.* Chongqing: College of Computer Science and Engineering Chongqing University.

[11] Li Jingbin, Yang Liu, Hua Bei (2011) Research on parallel K-Medoids algorithm based on multi-core platform. *Application Research of Computers*, 28(2), p.p.498-505.

[12] Cao Feng, Zhou Aoying (2007) Fast Clustering of Data Streams Using Graphics Processors. *Journal of Software*, 18(2), p.p.291-302.

[13] Zhou Bing, Feng Zhonghui, Wang Hexing (2007) The Study of Parallel Clustering Algorithm for Cluster System. *Computer Science*, 34(10),p.p.4-16.

[14] Boutsinas B, Gnardellis T.(2002) On distributing the clustering process. *Pattern Recognition Letters*, 23(8), p.p. 999-1008.

[15] Wegener D, Mock M, Adranale D.(2009) Toolkit-based high-performance data mining of large data on MapReduce clusters. *IEEE International Conference on Data Mining, Washington: IEEE*, p.p.296-301.

[16] Chen Hui-ping, Lin Li-li, Wang Jian-dong, Miao Xinrui (2008) Data mining platform WEKA and secondary development on WEKA. *Computer Engineering and Applications*, 44(19), p.p.76-79.

[17] X. Zhang, H. Li, and C. Qi, "Spatially constrained fuzzy-clustering-based sensor placement for spatiotemporal fuzzy-control system," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 946–957, Oct. 2010.

[18] L. F. S. Coletta, L. Vendramin, E. R. Hruschka, R. J. G. B. Campello, and W. Pedrycz, "Collaborative fuzzy clustering algorithms: Some refinements and design guidelines," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 3, pp. 444– 462, Jun. 2012.

[19] E. Hullermeier, M. Rifqi, S. Henzgen, and R. Senge, "Comparing fuzzy partitions: A generalization of the rand index and related measures," *IEEETrans. Fuzzy Syst.*, vol. 20, no. 3, pp. 546–556, Jun. 2012.

[20] J. P. Mei and L. H. Chen, "A fuzzy approach for multitype relational data clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, pp. 358–371, Apr. 2012.

[21] H. C. Huang, Y. Y. Chuang, and C. S. Chen, "Multiple kernel fuzzy clustering,"*IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, Feb. 2012.