ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Parallel Implementation of Fuzzy K-Means Algorithm Using Hadoop

[I]Jerril Mathson Mathew, [II]Jyothis Joseph
[I]M.Tech Scholar, College of Engineering Kidangoor, Kerala, India
[II]Assistant Professor, College of Engineering Kidangoor, Kerala, India

Abstract
Clustering is regarded as one of the momentous task in data mining which deals with primarily grouping of similar data. To cluster large data is a fact of concern. In recent years, data clustering has been studied extensively and a lot of methods and theories have been achieved. Hadoop is a software framework which deals with distributed processing of vast amount of data across groups of distributed computers using Map-Reduce programming model. The Map-Reduce computing model have two phases: map phase and reduce phase. The map phase calculates the distances between each point and each cluster and allots each point to its nearest cluster. All the points which belong to the same cluster are sent to a single reduce phase. The reduce phase calculates the new cluster centers for the next Map-Reduce job. Map-Reduce 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. This paper focuses on studying the parallel implementation of Fuzzy K-Means clustering algorithms using Map-Reduce computing model of Hadoop on different datasets and a comparison between the two clustering techniques.

Keywords
Data Mining, Data Clustering, Parallel Computing, Map-Reduce, Fuzzy K-Means Algorithm, Hadoop, HDFS, Machine Learning.

I. Introduction

With prime database expertise and worldwide 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 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 viewpoint of "Like poles attracts each other", 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 dissimilar 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 administrating massive amounts of data.

Cloud computing has got widespread attention as a promising business model. [5] Hadoop is a cloud computing platform which can more straightforwardly develop and process outsized data in parallel. [6] Its main features include strong development facility, low expenditure, high effectiveness and good quality trustworthiness. Hadoop platform consists of two parts: Hadoop Distributed File System (HDFS) and Map-Reduce computing model. [7] On the basis of cloud computing platform Hadoop, the paper depicts a parallel k-means clustering algorithm based on Map-Reduce 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 C-means clustering [1] given by Bezdek introduced that each one point has a probability of belonging to a definite cluster. A coefficient value coupled with every point gives the degree of being in the kth cluster and coefficient values should sum to one. Nowadays bigger datasets are well thought-out 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 Map-Reduce, Hadoop fires a query on the large datasets, divide it and then runs it in parallel on multiple nodes. The paper later explains about the methodologies used in Section 2. The Section 3 gives the dataset description.

II. Methodology

1. Hadoop

By Hadoop, an open source framework implementing the Map-Reduce programming model includes two components to be precise the Hadoop Distributed File System (HDFS) [4] and Map-Reduce. HDFS is used for storage of large dataset and Map-

Reduce 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 reserves the metadata and the DataNodes hoards 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 Map-Reduce model. Hadoop, a distributed file system is written in Java.

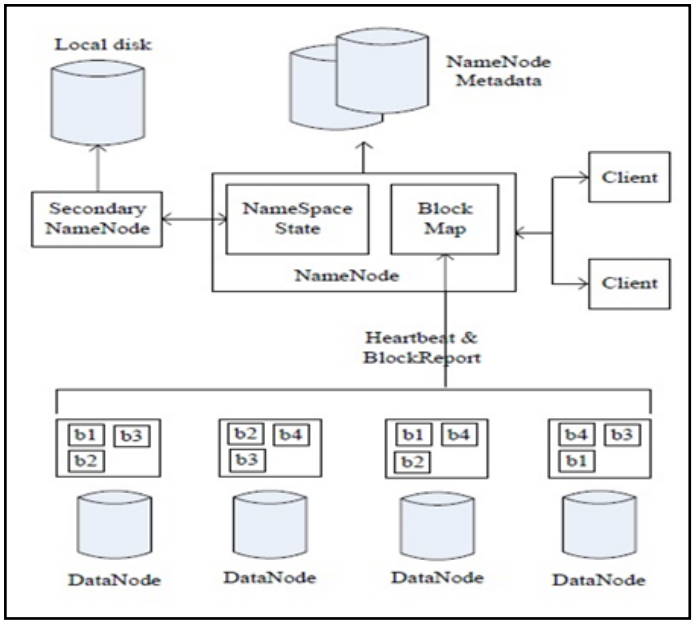

Fig 1: Architecture diagram of HDFS

2. Map-Reduce

There are mainly two programs in Map-Reduce, 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, Map-Reduce of Hadoop also adopt Master / Slave architecture, as shown in Fig 2.

Map-Reduce is a programming paradigm that expresses a bulky distributed computation as a sequence of distributed operations on data sets of key/value pairs. The Map-Reduce framework of Hadoop harnesses a cluster of machines and executes user defined Map-Reduce jobs athwart the nodes in the cluster. Map-Reduce approach is composed by JobTrackers and TaskTrackers. A Map-Reduce 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').

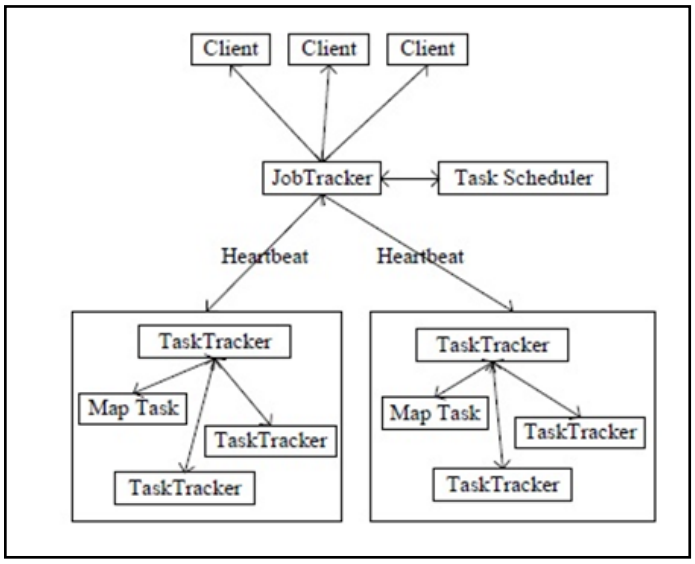

Fig 2: Architecture diagram of Map-Reduce of Hadoop

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 specific key 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 Map-Reduce computing model.

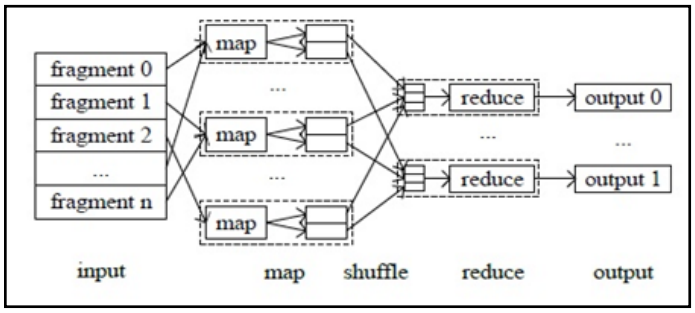

Fig 3: Map-Reduce computing model of Hadoop

3. Parallel Fuzzy K-Means Clustering

The Fuzzy K-mean clustering, also known as soft clustering is an extension of standard K- means clustering. It diminishes the intra-cluster variations. Bezdek introduced the notion of fuzziness parameter (m) in Fuzzy K-mean clustering which find out the degree of fuzziness in the clusters [1]. The algorithm of standard Fuzzy K-mean clustering algorithm is as follows:

1. Choose the number of clusters
2. Construct a distance matrix from a point x_j to every cluster heads considering the Euclidean distance between the point and the cluster head by the formula:

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

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

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

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,
$u_i(x_j)$ 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 usually greater than 1 because if it is kept equal to 1, then it look a lot likes 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 heads 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, Map-Reduce approach is used.

Map-Reduce Approach
Map-Reduce approach partitions the large datasets and then work outs 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 equation (1)
3. Compute the membership value of each vector point and create the membership matrix using equation (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 Fig 4. Total no. of reducers are less than total no. of mappers (M< N).

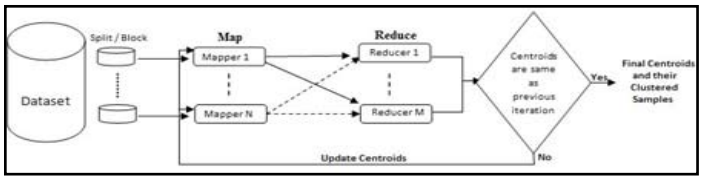

Fig 4: Fuzzy K-Means Clustering in Map-Reduce

III. Dataset Description

For the purpose of testing the performance of Fuzzy K-Means clustering on Hadoop using Map-Reduce, datasets from well-known UCI Machine Learning Repository are used namely Iris dataset, KDD Cup Dataset and Ecoli Dataset.

1. Iris Dataset

The Iris flower data set is a multivariate data set introduced by Ronald Fisher. The use of multiple measurements in taxonomic problems is as an example of linear discriminant analysis. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species.
Iris dataset, of size 8KB has five attributes. Out of which four attributes (Sepal length, sepal width, petal length, petal width) are numeric and fifth attribute has three classes (Iris Setosa, Iris Versicolour, and Iris Virginica) are there for this one nonnumeric attribute. It is created by R.A. Fisher. There are total 150 samples in this dataset. Based on Fisher's linear discriminant model, this data set became a typical test case for many statistical classification techniques in machine learning such as support vector machines.

2. KDD Cup Dataset

This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD Cup 99 -The Fifth International Conference on Knowledge Discovery and Data Mining. KDD cup 1999 dataset (10% of full) of size 75 MB is classified as labelled and unlabelled records. There are 41 attributes in each labelled record along with one target value indicating the category name of the attack.The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

3. Ecoli Dataset

The Ecoli data set is a protein localization sites dataset introduced in September 1996. Ecoli dataset, of size 20KB has nine attributes. Out of which seven attributes (mcg: McGeoch's method for signal sequence recognition, gvh: von Heijne's method for signal sequence recognition, lip: von Heijne's Signal Peptidase II consensus sequence score Binary attribute, chg: Presence of charge on N-terminus of predicted lipoproteins Binary attribute, aac: score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins, alm1: score of the ALOM membrane spanning region prediction program and alm2: score of ALOM program after excluding putative cleavable signal regions from the sequence) are numeric and two attributes(Sequence

Name: Accession number for the SWISS-PROT database and class name) are nonnumeric. The ninth attribute has eight classes (cytoplasm, inner membrane without signal sequence, perisplasm, inner membrane uncleavable signal sequence, outer membrane, outer membrane lipoprotein, inner membrane lipoprotein and inner membrane cleavable signal sequence) are there for this one nonnumeric attribute. There are total 336 samples in this dataset.

IV. Conclusions

This paper conducts in-depth research on the parallel Fuzzy K-Means algorithm based on Map-Reduce 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 Map-Reduce. 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.

V. AcknowledgEment

The author would like to thank Mrs. Anitha R. and Mrs. Rekha K.S. (Asst. Prof., Dept. of CSE, CE Kidangoor) and Mrs. Lekshmy P Chandran (Asst. Prof., Dept. of IT, CE Kidangoor) for their valuable suggestions and guidance in improving the paper's quality. The author would also like to thank TEQIP Phase-II for providing the economic support for this project.

References

[1] "FCM: THE FUZZY c-MEANS CLUSTERING ALGORITHM", vol. 10, pp191-203, 1984.
[2] Hadoop official site, http://hadoop.apache.org/core/.
[3] 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
[4] Armbrust M, Fox A.(2009) Above the clouds: a Berkeley view of cloud computing. University of California at Berkeley.
[5] Tom White.(2012) Hadoop: The Definitive Guide Third Edition. O'Reilly Media.
[6] Dean J, Ghemawat S.(2008) MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), p.p.07-113.
[7] 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
[8] Garg, Dweepna, Parth Gohil, and Khushboo Trivedi. "Modified Fuzzy K-mean clustering using MapReduce in Hadoop and cloud," IEEE International Conference on Electrical Computer and Communication Technologies (ICECCT), 2014.