International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 408-414
e-ISSN:2278-621X

SCALABLE K-MEANS ALGORITHM USING MAPREDUCE TECHNIQUE FOR CLUSTERING BIG DATA

Doreswamy[1], Osama A.Ghoneim[2] and B.R. Manjunatha[3]

Abstract- The era of big data is coming. But the using of traditional data analytics may not be efficient to process such huge quantities of data. Cluster analysis is one of the most significant and commonly used data analysis techniques. K-means still one of the most common clustering algorithm because of its simplicity. As the volume of data is being so huge a lot of researcher turn to MapReduce to gain high performance. In this paper, k-means clustering algorithm scaled up to be applicable for different datasets with large size using Hadoop and MapReduce platform.

Keywords – K-means, MapReduce, Big data, clustering.

I. INTRODUCTION

Data mining is the process of extracting useful information by using different techniques like clustering and classification .There are many data sources for mining purpose are available in different forms like, data warehouse database, the Web, and data which are streamed dynamically in the system[3]. Nowadays internet of things becoming one of the most important sources for data as these data may be used in a lot of application inside smart city which will help to make the life of the human more easy and comfortable. The demand of data mining methods to gain a lot of information from this valuable source becomes more vital. Data mining algorithms should be processed via using suitable computing technique like distributed computing. Distributed computing is a model used to do high computational processing over a set of connected systems. Each individual system interconnected on the network is called a node and the collection of many nodes that form a network is called a cluster[4].

Cluster analysis is one of the most common and imperative field in data mining. The propose of clustering is to discover essential data structures in data, and shape them into expressive subgroups. Each cluster is a unique subset aimed to maximize intraclass similarity and minimize interclass similarity [2]. Different clusters can be formed with same dataset using various clustering techniques. There are many well known clustering algorithms that can be

[1] *Department of Computer Science, Mangalore University, Mangalagangotri- 574 199, Karnataka, INDIA*
[2] *Assistant Lecturer, Computer Science Division, Mathematics Department, Faculty of science, Tanta University, Tanta, Egypt & Research Scholar, Deptt. of Computer Science, Mangalore University, Mangalore*
[3] *Department of Marine Geology Mangalore University, Mangalagangotri- 574 199, Karnataka, INDIA*

designed for very different research topics. Those categorized from several orthogonal aspects such as separation of clusters, partitioning criteria, similarity measures used and clustering space [7].

Apache Hadoop [1] is an open-source software platform used for distributed storage and distributed processing of very huge datasets on commodity machines. It came into our real world from Google's MapReduce and Google File systems projects. It is a framework that can be used for powerful data applications which are processed in a

distributed computing network[5]. Hadoop framework aimed to compute large number of petabytes of data. The job is distributed amongst the nodes interconnected to the network that will increase performance and efficiency of the system and network. The core of apache Hadoop consists of two parts:

1. Storage Part:-Hadoop Distributed File System (HDFS)
2. Processing Part: - MapReduce Paradigm.

Hadoop offers a well defined file system to organize processed data which is distributed; reliable and scalable is known as Hadoop Distributed File System (HDFS). Map and Reduce programming model arranges the processing by marshalling the distributed servers, running the different tasks in parallel, handling all interconections and data transfers between various parts of cluster and providing redundancy and fault tolerance. In Map and Reduce, thefragmentation of data is the basic stage and this fragmented data is fed into the distributed network for processing. At the end, the processed data is integrated as a whole.

The Hadoop framework [2, 3] takes into account the node failures and is automatically handled by itself. This makes Hadoop really elastic and adaptable platform for data serious applications. The answer to growing volumes ofdata that demand fast and effective information retrieval lies in generating the rules of data mining over a distributed environment like Hadoop.

The rest of the paper is organized as follows. Cluster analysisis explained in section II. K-Means clustering algorithm is presented in section III. Map-Reducetechnique is given in section IV.K-Means clustering using Map-Reduce technique is illustrated in section V. Results and discussion is given in section VI. Finally, the conculsion of the paper is presented in section VII.

II.CLUSTER ANALYSIS

Data mining is interdisciplinary topic which can be defined in many various ways. There are a number ofdata mining methods are used to determine the types of patterns to be found in data mining task. These methods include discrimination and characterizations, frequent patterns mining, correlations and associations, classification and regression; clustering analysis, outlier analysis. Clustering is one of the most exciting topics in data mining. Clustering used in many application areas such as business intelligence, image pattern recognition, biology, security, and Web search. The objective of clustering is to explore intrinsic structures in data, and arrange them into expressive subgroups. The basic concept of cluster analysis [1] is the process of dividing large data set of objects into small subsets. Each small subset is a single cluster, such that the objects are clustered together depending on the concept of minimizing interclass and maximizing the intraclasssimilarity[7]. Similarity and dissimilarity are assessed based on the feature values describing objects and various distance measures. We measures object's similarity and dissimilarity by comparing objects with each other. These measures include distance measures such as supremum distances, Manhattan distance, and Euclidean distance, between two objects of numeric data,[1]. Cluster analysis is a vast topic and hence there are many clustering algorithms available to group datasets.

On the basis of implementation different clustering algorithm can be grouped together into

- Partitioning Method
- K-means
- K- medoids
- Hierarchical Method
- Chameleon
- BIRCH
- Density Based Clustering Method
- OPTICS
- DBSCAN
- Grid Based Clustering Method
- CLIQUE
- STING

III. K-MEANS CLUSTERING ALGORITHM

The k-means algorithm takes the input data set D and parameter k, and then divides a data set D of n objects into k groups. This partition depends upon the similarity measure so that the resulting intra cluster similarity is high but the inter cluster similarity is low. Cluster similarity is measured regarding the mean value of the objects in a cluster, which can be showed as the cluster's mean. The k-means procedure works as follows. First, it randomly chooses k of the objects, each of which initially defined as a cluster mean or center. For each of the remaining objects, an object is moved to the cluster to which it is the most similar, based on the similarity measure which is the distance between the item and the cluster average. It then calculates the new mean for each cluster. This process repeats until no change in the mean values in the clusters.

Algorithm: k-means.
Input: $E = \{e_1, e_2,..., e_n\}$ (set of objects to be clustered)
k (number of clusters)
Output: $C = \{c_1, c_2,...,c_k\}$ (set of cluster centroids)
$L=\{l(e) \mid e = 1,2,...,n\}$ (set of cluster labels of E)
Methods:
1. Randomly choose k points from the data set D as the initial cluster means (centroids);
2. Assign each object to the group to which is the most closest, based on the means values of the objects in the cluster;
3. Recalculate the mean value of the objects for each cluster;
4. Repeat the steps 2 and 3 until no change in the means values for the groups

IV.MAP-REDUCETECHNIQUE

HadoopMapReduce is a software framework for easily writing applications which process huge quantities of datalike terabytes or even petabytes of data in parallel on large clusters of commodity machines in a scalable, reliable, and fault tolerant manner[3]. Usually a MapReduce job divide the input dataset into independent parts which are processed by map tasks in a totally parallel and independent way. The framework order the outputs of maps, which are then input to

the reduce job. Both input and output of the jobs inside Hadoop are stored in HDFSfile system. Basically the compute node and storage nodes are the same, that is MapReduce and HDFS are running on the same set of nodes. This configuration makes the framework to be more effective on scheduling tasks over the nodes where data is already present [2]. The MapReduce framework operates exclusively on *(key,value)* pairs, which is the framework views input to the task as a set of *(key, value)* pairs and produces another set of (*key,value)* pairs as output of the task.

Map and Reduce stages are separate, different and full freedom is given to the programmer to layout them. Each of the Map and Reduce steps are executed in parallel manner on a set of pairs *(key,value)*data members. Therebythe program is segmented into two different and well defined steps namely Map and Reduce[4]. The Mapstage includes performance of a task on a given dataset in the form of *(key,value)*and produce the intermediate dataset. After that, the produced intermediate dataset is arranged for the implementation of the Reduceprocess. Data transfer takes place in between the Map and Reduce jobs. The Reduce task collects all the datasets similar to the particular key and this process is repeated for all the different key values. The final out put given by the Reduce job is a dataset of *(key,value)*pairs[5]. Each MapReduce Framework has job Tracker and multiple task trackers. Each node connected to the network has the right to act as a slave Task Tracker. Master node takes care of all issues such as split of data to various nodes , node failures, task scheduling, task failure management, communication of nodes, monitoring the task progress[6].

V. K-MEANS CLUSTERING USING MAP-REDUCE TECHNIQUE

The first step of designing MapReduce code Kmeans algorithm is to express and investigate the input and output of the implementation. Input is given as <key,value> pair, where "key" is the cluster mean and "value" is the serializable implementation of a vector in the dataset[2]. The prerequisite to implement Map routine and Reduce routine is to have two files. The first one should involve clusters with their centroids values and the other one should have objects to be clustered[7]. Chosen of centroids and the objects to be clustered are arranged in two spillited files is the initial step to cluster data by K-means algorithm using MapReduce method of Apache Hadoop. It can be done by following the algorithm to implement MapReduceroutines for K-means clustering[8]. The initial set of centroid is stored in the input directory of HDFS prior to Map routine call and they form the "key" field in the <key, value> pair. The instructions required to compute the distance between the given data set and cluster centroid fed as a <key, value> pair is coded in the Mapper routine. The Mapper function calculates the distance between the object value and each of the cluster centroidreferred in the cluster set and jointly keeping track of the cluster to which the given object is closest[9].Once the computation of distances is complete the object should be assigned to the closest cluster.

Once Mapper is invoked, the given object is assigned to the cluster that it is nearest related to. After the assignment of all objects to their associated clusters is done the centroid of each cluster is recomputed[10].

The recalculation is done by the Reduce routine and also it restructures the cluster to avoid generation of clusters with extreme sizes. At the end, once the centroid of the given cluster is revised, the new set of objects and clusters is re-written to the memory and is ready for the next iteration.

Algorithm 1: Mapper Design for K-means Clustering

```
1: procedure KmeansMapDesign
2: Load Cluster file
3: fp= Mapclusterfile
4: Create two list
```

Algorithm 2: Reducer Design for K-means Clustering

```
1: procedure KmeansReduceDesign
2:   NEW ListofClusters
3:  COMBINE resultant clusters from MAP CLASS
4:  if cluster size too high or too low then RESIZE cluster
5:  CMax=find MaxSize(ListofClusters)
6:  CMin=findMinSize(ListofClusters)
7:  if  CMax>(1/20)totalSize then Resize(cluster)
14: end procedure = 0
```

Algorithm 3: Implementation of K-means Function

```
1: procedure K-means Function
2: if Initial Iteration then LOAD cluster file from DIRECTORY
3:  else READ cluster fie from previous iteration
4:  Create new JOB
5:  SET MAPPER to map class defined
6:  SET REDUCER to reduce class define
8:  path for output DIRECTORY
9:  SUBMIT JOB
10: end procedure = 0
```

VI. RESULTS AND DISCUSSION

Experimental setup

To implement the k-means algorithm we installed cluster composed of four nodes in aws,

1- One Master Node (instance) of typem4.2xlarge having Ubuntu 14.04, 64 bit.
2- Three slave Nodes (instance) of type m4.large having Ubuntu 14.04, 64 bit.
3- Hadoop 2.4.1
4- JDK 1.7

This distributed environment of four instances in AWS used to implement, perform the k-means clustering algorithm and to save the results.

Data set description

To scale k-means clustering algorithm one of smart city dataset used [11]. We used the pollution data set which consists of 449 file. Each file contains around 17500 observation of the pollutants ratio of five attributes.

Evaluation

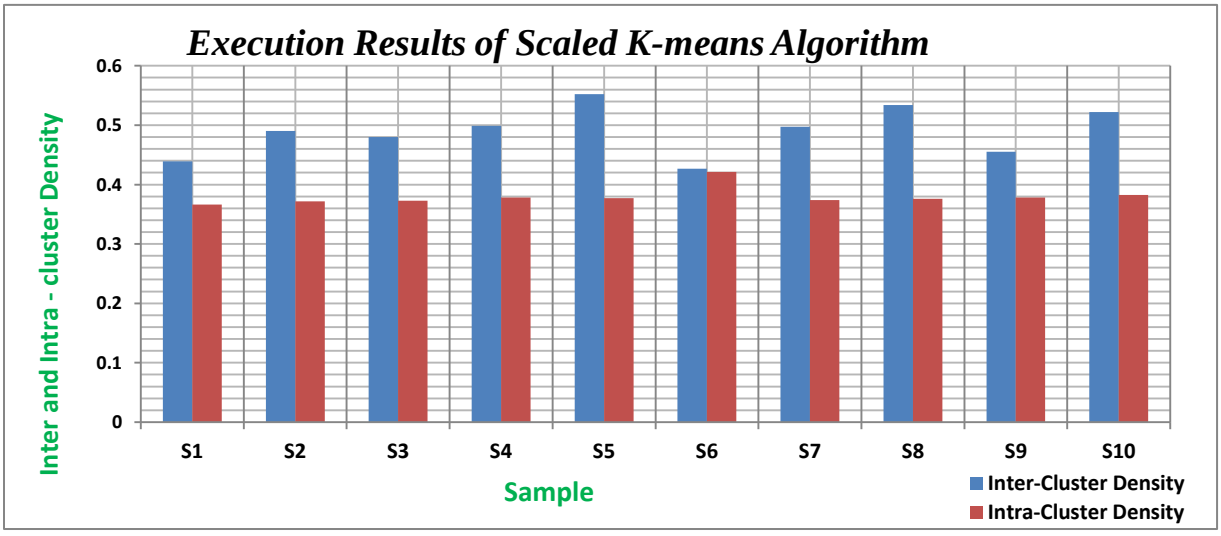
To measure the performance of the scaled k-means algorithms using HadoopMapReduce, we have executed thealgorithms on 10 different samples of data. After execution of the algorithm, we have calculated and measure the inter-cluster and intra-cluster similarity measure.

- **The inter-cluster distance:** distance$d(i,j)$ between two clusters is measuredas the distance between the centroids of the clusters.
- **The intra-cluster** distance measured between the all pair of objects within a cluster.

The following table and figure represent the experimental results of K-means algorithm on different data samples where k=3.

Table 1: Execution results of scaled K-means algorithm

Sample	Sample size	Inter-Cluster Density	Intra-Cluster Density
S_1	78290	0.439142	0.366309
S_2	1576718	0.490337	0.371887
S_3	2368512	0.480140	0.372767
S_4	3153530	0.498691	0.378400
S_5	3942470	0.552399	0.377079
S_6	4732842	0.426724	0.421366
S_7	5522887	0.497220	0.373842
S_8	6312932	0.533907	0.375958
S_9	7099392	0.454998	0.377970
S_{10}	7887974	0.521926	0.382288



From the results it is clear the sample s_5 shows the maximum inter-cluster density of 0.552399 which indicates well separation of different cluster. Similarly, the inter-cluster density for sample s_8 is calculated as 0.533907, separating data clusters very well. Also the results of Intra-cluster

density for sample s_1 show minimum value, which gives a clear indication of having the similar objects in the same cluster.

VII. CONCLUSION

In this paper k-means clustering algorithm scaled up to be applied to huge dataset which contain around 8millionobjects. Each object is a vector of five attributes. Inter and intra cluster measurements computed to find the maximum value of inter-cluster density and the minimum value of intra-cluster measurements. This work done using Hadoop and MapReduce framework which gives high performance in big data analysis.

ACKNOWLEDGEMENT

The authors thank for financial assistance provided by the SERB-DST, New Delhi vide SB/EMEQ-137/2014, dated 21-03-2016, the ICCR fellowship by the Governments of India, and Egypt for carrying the project.

REFERENCES

[1] Cui, Xiaoli, et al. "Optimized big data K-means clustering using MapReduce." *The Journal of Supercomputing* 70.3 (2014): 1249-1259.

[2] Akthar, Nadeem, MohdVasimAhamad, and Shahbaz Khan. "Clustering on Big Data Using HadoopMapReduce." *Computational Intelligence and Communication Networks (CICN), 2015 International Conference on*. IEEE, 2015.

[3] Patil, Yaminee S., and M. B. Vaidya. "K-means Clustering with MapReduce Technique."

[4] Moertini, Veronica S., and LiptiaVenica. "Enhancing parallel k-means using map reduce for discovering knowledge from big data." *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*. IEEE, 2016.

[5] Eluri, Venkateswara Reddy, et al. "A comparative study of various clustering techniques on big data sets using Apache Mahout." *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*. IEEE, 2016.

[6] Anchalia, Prajesh P., Anjan K. Koundinya, and N. K. Srinath. "MapReduce design of K-means clustering algorithm." *2013 International Conference on Information Science and Applications (ICISA)*. IEEE, 2013.

[7] Akthar, Nadeem, MohdVasimAhamad, and Shahbaaz Ahmad. "MapReduce Model of Improved K-Means Clustering Algorithm Using HadoopMapReduce." *Computational Intelligence & Communication Technology (CICT), 2016 Second International Conference on*. IEEE, 2016.

[8] Wang, Shuguang, and Chao Jiang. "K-means Parallelization Algorithm Based on MapReduce." *International Journal of Database Theory and Application* 9.8 (2016): 21-30.

[9] Xu, Hongbo, et al. "Parallel implementation of K-Means clustering algorithm based on mapReduce computing model of hadoop." (2015).

[10] Zhao, Weizhong, Huifang Ma, and Qing He. "Parallel k-means clustering based on mapreduce." *IEEE International Conference on Cloud Computing*. Springer Berlin Heidelberg, 2009.

[11] http://iot.ee.surrey.ac.uk:8080/