MapReduce Programming Paradigm Solving Big-Data Problems by Using Data-Clustering Algorithm

Yadav Krishna R (Pursuing M.tech,) PIET, Vadodara
Ms. Purnima Singh (Assistant Professor) PIET, Vadodara

***Abstract*— Data clustering is a common technique used in data analysis and is used in fields including data mining and image analysis. It is partitioning technique of similar data and also determine similarity between data or group of data. Data clustering can be expensive and time consuming because of its iteration and repeat data clustering. Hence parallelizing and distributing becomes attractive in terms of its speed-up in computation and increase memory available in computation. Traditional distributed programming methods are sophisticated for parallel computing because they face problems like deadlocks and synchronization.**

Map-Reduce is a programming paradigm for solving certain problems of computing cluster. MapReduce having simple two steps process. In Map step a master node divides problem into number of different parts, that are forward to map tasks. Each map task processes its part and output results as key-value pairs. The Reduce step receives the outputs of maps , where particular receiver will receive only particular part of map and it will process those. Apache Hadoop provide Map-Reduce programming paradigm that allow parallel and distributed programmer to program easily for data clustering. This paper covers various data clustering algorithms using MapReduce and their benefits.

***Index Terms*—MapReduce, Data clustering, Hadoop, HDFS, Distance metrics.**

I. INTRODUCTION

Data clustering can be considered one of the most important unsupervised learning techniques and an important task in data mining. The goal of data clustering is to determine the intrinsic grouping in a set of unlabeled data. Various practical application problems can be solved with clustering method. It has been used in areas like marketing, biology, library, insurance, and world wide web and so on. Today, data set becoming larger and larger hence, classical clustering method will not work to deal with large data set. Parallel clustering method based on MapReduce is efficient model to deal with such problems. *Hadoop* is a java based computing framework which supports MapReduce paradigm. Hadoop has execute-once semantics, meaning that with iterative tasks all state information needs to be written to file and then read back in for every step of computation. *Hadoop Distributed File System* (HDFS) is used to store data for input process and store output files.

***Yadav Krishna R**, M.Tech, Parul Institute of Engineering and Ttechnology, Vadodara, Gujarat., Vadodara, India, +91-9033-493036*
***Ms. Purnima Singh**, Assistant Professor, Parul Institute of Engineering and Technology, Vadodara, Gujarat.*

II. DATA CLUSTERING

A. Overview

Data Clustering deals with partitioning of objects into groups, generally called clusters such that the similarity between members of same group is maximized and similarity between members of different groups is minimized. Various form of distance measure is used to determine similarity of objects.

B. Distance metrics

1. *Cosine Similarity*: It is a measure of distance between two vectors of dimension n, by finding the cosine of angle between them. Comparing data of text mining, it is mostly used.
   Given A and B vectors, the cosine similarity is given by formula:

$$Cos(\theta) = \frac{A.B}{||A||.\ ||B||}$$

   Result ranges from -1 to 1, closer the result to one the more similar the vectors.
2. *Mahalanobis distance*: It is used to determine the similarity of an unknown sample set to known set. It form a group of values with mean $\mu = (\mu_1,\mu_2,...\mu_N)^T$ and covariance matrix S for a multivariate vector is defined as :

$$D_M(x) = \sqrt{(x-\mu)^T S-1(x-\mu)}$$

   It is widely used in cluster analysis and other classification techniques.
3. *Euclidean distance*: It find the distance between two data points involves computing the square root of sum of the squares of the differences between corresponding values.
   The formula for this distance between a point $X(x_1,x_2,x_3,..etc.)$ and a point $Y(y_1,y_2,y_3...etc.)$ is :

$$d = (\sum_{j=1}^{n}(xj - yj)2)^{\frac{1}{2}}$$

   It is most widely use distance metrics.

ISSN: 2278 – 1323

C. Data Clustering Algorithm

1. *K-Means algorithm*: It assigns each given point to the cluster whose center is nearest (also called centroid). Here, the center is average of all the points in the cluster that is, its co-ordinates are arithmetic mean for each dimension separately over all the points in the cluster. It is popular because of its time complexity O(k + n)) and it is order independent.

   Let's see how it works:
   1. Choose K as number of clusters.
   2. Initialize the vectors of the K clusters
   3. For every new sample, compute the distance between the new vector and every cluster's vector and re-compute the closet vector with the new vector.

2. *K-Medoids algorithm:* Here, each cluster is represented by one of the objects located near the center of the cluster. PAM (Partitioning around medoids) was one of the first k-medoids algorithm is introduced. It is more robust than K-means in presence of noise and outlier detection.

   K-medoids works in following manner:
   1. Arbitrarily choose k objects as initial medoids.
   2. Assign each remaining object to the cluster with the nearest medoids.
   3. Randomly select a non-medoid objects $O_{random.}$
   4. Compute the total cost, S of swapping O_j with $O_{random.}$
   5. If S<0 the swap O_j with O_{random} to form new set of k-medoids.

3. *DBScan algorithm*: Here clusters are identified by looking at the density of points. Regions having high density of points depict the existence of clusters whereas regions with low density of points indicate clusters of noise or clusters of outliers. It deals with large datasets , with noise, and is able to identify clusters with different shapes and sizes.

   This algorithm needs three parameters:
   1. K, the neighbor list size.
   2. Eps, the radius that delimitate the neighbourhood area of a point (Eps neighborhood)
   3. Min pts, the minimum number of points that must exist in the Eps-neighbourhood.

III. TYPES OF CLUSTERING

Data clustering types depend on the algorithms that were discussed in previous section. Mainly it has two types hierarchical and partitional. Hierarchical algorithms find successive clusters using previously created clusters, whereas partitional algorithms determine all clusters at time. Hierarchical can be bottom-up or top-down.

The clustering algorithms can be compared according to following factors:

The size of the dataset, number of the clusters, type of dataset, and type of Software.

	Size Of Dataset	Number Of Clusters	Type Of Dataset	Type Of Software
Partitional Algorithm	Huge Dataset & Small Dataset	Large and small number of clusters	Initial Dataset and Random Dataset	LNKnet package and Cluster and TreeView
Hierarchical Algorithm	Huge Dataset and Small Datset	Large and small number of clusters	Initial Dataset and Random Dataset	LNKnet package and Cluster and TreeView

Many clustering algorithms require the specification of number of cluster to produce in input data set, prior to execution of algorithm. Barring knowledge of the proper value beforehand, the appropriate value must be determined , a problem on its own for which a number of techniques have been developed.

IV. HADOOP

Hadoop is an open source apache project written in java language[12]. It provide software framework for distributed processing of large datasets in real-time applications. It provide efficient result in distributed computing because of its simple infrastructure. It enables applications to work with thousands of node and petabytes of data and also provide reliable shared storage and analysis system. The exponential growth of individual data footprints, as well as the amount of data generated by machines called for a means to process said data. Hadoop deals with large amounts of data by splitting it into subsets and sending the data to multiple computer processor unit cores, better described as nodes. Multiple nodes used together process data at a much higher rate. Hadoop then reassembles the data into a single result set. Complex data operations shifted to clusters of computers known as clouds in which software such as Hadoop orchestrates the merging of these processes[13].

The hadoop architecture have mainly two parts*: Hadoop distributed File System (HDFS) and the MapReduce engine.*

HDFS : HDFS is a pure Java file system designed to handle very large files. Similar to traditional file systems, Hadoop has a set commands for basic tasks such as deleting files, changing directory and listing files. The files are stored on multiple machines and reliability is achieved by replicating across multiple machines[4]. However, this is not directly visible to the user it and appears as though the file system exists on a single machine with a large amount of storage. By default the replication factor is of which at least 1 node must be located on a different rack from the other two. Hadoop is designed to be able to run on commodity class hardware. The need for RAID storage on nodes is eliminated by the replication[5]. A limitation of HDFS is that it cannot be directly mounted onto existing operating systems. Hence, in order to execute a job there must be a prior procedure that involves transferring the data onto the file system and when

job is complete the data must be removed from the file system. This can be an inconvenient time consuming process. One way to get around the extra cost incurred by this process is to maintain a cluster that is always running Hadoop such that input and output data is permanently maintained on the HDFS.

MapReduce Engine : MapReduce is a framework that allows certain kinds of problems particularly those involving large data sets to be computed using many computers. Problems suitable for processing with MapReduce must usually be easily split into independent subtasks that can be processed in parallel. In parallel computing such problems as known as embarrassingly parallel and are ideally suited to distributed programming[12]. MapReduce was introduced by Google and is inspired by the map and reduce functions in functional programming languages such as Lisp. In order to write a MapReduce pro-gram, one must normally specify a mapping function and a reducing function. The map and reduce functions are both specified in terms of data that is structured in key-value pairs. The power of MapReduce is from the execution of many map tasks which run in parallel on a data set and these output the processed data in inter-mediate key-value pairs. Each reduce task only receives and processes data for one particular key at a time and outputs the data it processes as key-value pairs. Hence, MapReduce in its most basic usage is a distributed sorting framework. MapReduce is attractive because it allows a programmer to write software for execution on a com-puting cluster with little knowledge of parallel or distributed computing[12].

V. DATA CLUSTERING ALGORITHMS USING MAP REDUCE STRATEGY

A. Overview

The greatest challenge to processing and solving a problem using Hadoop's MapReduce system is designing the inputs and outputs. Complex problems must often be performed in multiple MapReduce steps. Each step takes as input the output from a previous step. The main advantages is that map and reduce functions are allowed for distributed processing. Therefore, all map operations can be performed in parallel, and set of reduce operations can be performed in parallel.

B. Data Preparation

By default Hadoop mappers process input text files line by line, it is logical to transform each file into a single line such that each files can be processed by single map. Hence the final output may be a collection of text files.

C. K-means with MapReduce

The K-means algorithm main task is to compute the distance between datasets and centroids and it takes most of the execution time. If there is large datasets, iteration steps use to increase and hence computation becomes overload. However the process is important to calculate the algorithm and in that way efficiency of algorithm can be improved. Therefore, to enhance distance calculation method is the most important things to improve the time calculation of the algorithm. In that case execution orders of distance calculation of objects will not affect result of clustering. Therefore, distance between datasets and centroids can be execute in parallel for better result using MapReduce Framework.

The k-means with MapReduce works as follows :

1. Let's take k points into the space which represent objects that are being clustered. These points represent the centroids.
2. Consider each object that are most closet to the clusters and they are assign to them.
3. Finally, when all the distance from the centroids are calculated, then objects assign to them and at last recalculate the centroids position based on results.

Repeat Steps 2 and 3 until the centroids got the repeated values. This produces a separation of the objects into clusters from which the metric to be minimized can be calculated.

D. Function K-means in Map

The map function of the K-means algorithm using MapReduce and its steps is shown in Figure 1, and this function is used to runs on each mapper in the cluster. This function results in assigns the object to the nearest centroid after calculation. Here the function uses a distance method to calculate the distance between the given object and each centroid in clusters in step 3. After calculating the nearest centroid to the object, this function puts the object in to the cluster which contains the centroid in step 8.

Function K-means_in_Map

Input: *centroids*, *input*. // The *key* of the *input* is the offset of the *input* segment in the raw text (data set), and the *value* of the input is an object for clustering. The *centroids* are raw centroids, and are given to the *map* function by the JobConf of MapReduce.

Output: *output*. // The *key* of *output* is the nearest cluster to the *object*, and the *value* of *output* is the *object*.

```
1: nstCentroid ← null, nstDist ← ∞
2: for each c∈ centroids do
3:    dist ← Distance (input. value, c);
4:    if nstCentroid == null || dist < nstDist then
5:       nstCentroid← c, nstDist ← dist;
6:    end if
7: end for
8: output. collect(nstCentroid, object);
```

Figure 1: K-means Map Function[11]

E. Function K-means in reduce

The reduce function of the K-means algorithm using MapReduce is shown in Figure 2, and this function runs on each given reducer in the given cluster. This function helps to

ISSN: 2278 – 1323

calculates the new centroid of the cluster for next step. This function first gets all objects in this cluster in step 3, and then uses a Centroid function to re-calculate the centroid with the objects in the cluster in step 5. After getting the new centroid of the cluster, this function returns the centroid in step 6.

Function K-means_in_Reduce

Input: *input.* // The *key* of *input* is the centroid of a cluster, and the *value* of *input* is a list of objects who are assigned to the cluster.

Output: *output.* // The *key* of *output* is the old *centroid* of the cluster, and the *value* of *output* is the new *centroid* of the cluster.

```
1: v ← Φ;
2: for each obj∈ input. value do
3:       v ← v ∪ {obj};
4: end for
5: centroid ← ReCalCentroid(v);
6: output. collect(input. key, centroid);
```

Figure 2 : K-Means Reduce function[11]

F. Conclusion

The volume of information exchange in today's world engages huge quantity of data processing and relating to Big datasets. The K-Means Clustering Algorithm over a distributed network, not only does this algorithm provide a robust and efficient system for grouping of data with similar characteristics. Also it helps in reducing the implementation costs of processing such huge volumes of data. Data Mining being one of the most important tools in information retrieval and provide technique for data processing. The rate of information exchange today is growing spectacularly fast and so there arises a need of processing huge volume of data. To respond to this need vital algorithms used for data mining on a distributed or a parallel environment to reduce the operational resources and increase the speed of operation.

MapReduce is a feasible solution to processing problems involving large amounts of data. Especially for problems that can easily be partitioned into independent sub tasks that can be solved in parallel. Hadoop is an open source MapReduce imple-mentation featured in this study. Hadoop should be considered only for non-time sensitive tasks that can be batch processed. The most challenging part with designing MapReduce programs is usually determining inputs, outputs and expressing a problem in MapReduce terms. In the future it is feasible that the demand for cluster computing will grow as problem sets become larger and more interest develops in the field. MapReduce will undoubtedly become a more popular solution and much used paradigm.

ACKNOWLEDGMENT

I am very grateful to Ms. Purnima Singh (Assistant Professor), Parul Institute of Engineering and Technology, Vadodara for her continuous help and support.

REFERENCES

[1] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. "The Google File System." Proceedings of the nineteenth ACM symposium on Operating systems principles. , pp. 29-43, ACM, New York, NY, USA, 2003.

[2] Jeffrey Dean, and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters." Proceedings of the 6th conference on Symposium on OperatingSystems Design & Implementation., pp. 137-150, USENIX Association, Berkley,CA, USA, 2004.

[3] C C Chang, B He, Z Zhang. Mining semantics for large scale integration on the web: evidences, insights, and challenges. SIGKDD Explorations, 2004: 6(2):67-76

[4] J. Coldberger, S. Gordon, H. Greenspan, "Unsupervised image-set clustering using an information theoretic framework," IEEE transactions on image processing, vol.15, no. 2, pp. 449-457, 2006

[5] A. McCallum, K. Nigam, and L. H. Ungar. Efficient Clustering of High Di-mensional Data Sets with Application to Reference Matching. Association for Computing Machinery (ACM), New York, NY, USA, 2000.

[6] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implemen-tation. IEEE Computer Society, Washington, DC, USA, 2002

[7] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wal-lach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber.

[8] "Bigtable: A Distributed Storage System for Structured Data." ACM Transac-tions on Computer Systems, vol. 26(4), ACM, New York, NY, USA, 2008.

[9] R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart The Mahalanobis distance. Texas A&M University, College Station, TX, USA, 2000.

[10] So What is Hadoop? (2010). Retrieved February 2011, from Atbrox: http://atbrox.com/2010/02/17/hadoop/

[11] How Map and Reduce Operations are Actually Carried Out. (2009).Retrieved February 2011, from Hadoop Wiki: http://wiki.apache.org/hadoop/HadoopMapReduce

[12] Google and IBM Announce University Initiative to Address Internet-Scale Computing Challenges. (2007). Retrieved February 2011, from IBM: http://www-03.ibm.com/press/us/en/pressrelease/22414

[13] Apache Hadoop. http://hadoop.apache.org/.

[14] T. White, "Hadoop, the Definitive Guide," O'Reilly Media, May 2009, p.

