International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN : 2456-3307

Clustering of large datasets using Hadoop Ecosystem

Mounica B, Aditya Srivastava, Md. Faisal Alam

Department of Information Science, New Horizon College of Engineering, Bangalore, Karnataka, India

ABSTRACT

In today's rapid change of world along with the advancement of technology, the amount of data being generated and used is very high. The rate of data production is very rapid and is not easy to measure. The existing data processing techniques are not capable enough to process data which are so large. K-means is a traditional clustering method which is easy to implement but it converges to local minima from starting position and is sensitive to initial clusters. Hadoop or the Hadoop Distributed File System (HDFS) is a distributed file system which is highly fault tolerant and can be implemented on low cost hardware. It provides complete access to data for any operation and is suitable for applications that needs large data sets. Hadoop is used for parallel processing of large data set in less time.

Keywords: Hadoop, MapReduce, K-means.

I. INTRODUCTION

Hadoop is an open-source framework which allows to store and process big data in a distributed environment across clusters of computers. It is designed in such a way that it can scale up from single servers to thousands of machines.

Hadoop is built on 4 modules and it is written in Java language. Hadoop framework includes the following components:

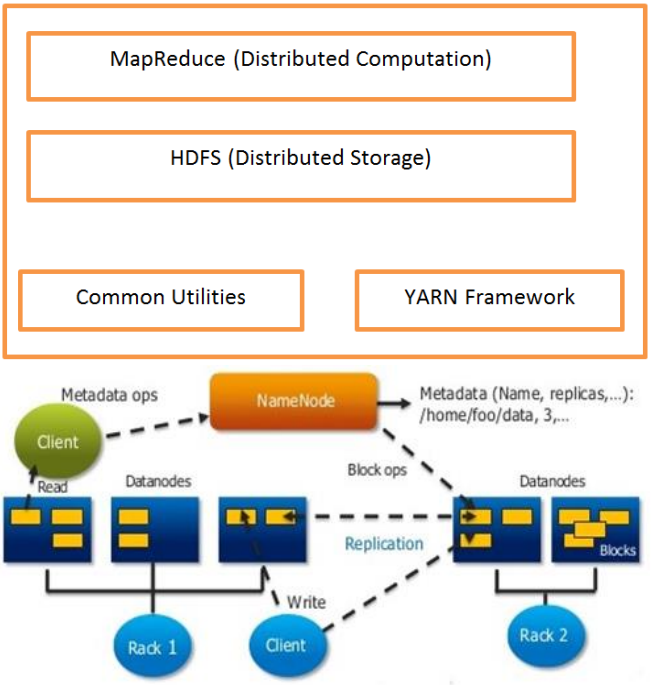
- Hadoop Common: It contains the Java libraries and utilities required by the other Hadoop modules.
- Hadoop YARN: It is a framework which is entirely responsible for cluster resource management and job scheduling.

HDFS Architecture

Hadoop has two major layers which are

1. Computation and processing layer (using MapReduce).
2. Storage layer (Hadoop distributed File System layer)

The architecture of Hadoop File System is as follows



Hadoop enitrely follows a master-slave architecture and it's components are Datanode and Namenode.

Datanode - It contains GNU/Linux operating system and datanode software. For every node in a cluster, there will be a datanode which manages the data storage on the system. As per the request of the client, the datanodes will perform read or write operation. Their functionality also include operations such as block creation, deletion and replication based on the

name node. The system with datanode software acts as a slave server.

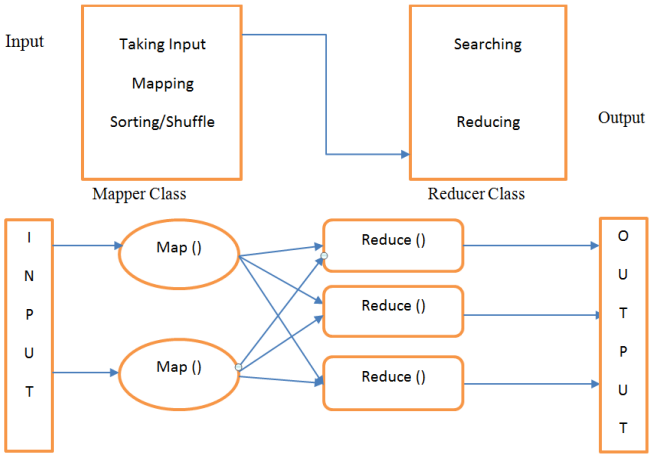
Namenode - It contains a GNU/Linux operating system and a namenode software. The system with namenode software acts as a master server and manages the file system namespace. They also perform file system operations such as opening, closing and renaming of directories

II. METHODS AND MATERIAL

MapReduce is a framework which was proposed by Google and the implementation was done by Apache. It is capable of processing large datasets in parallel mode. MapReduce has basically two major function which are

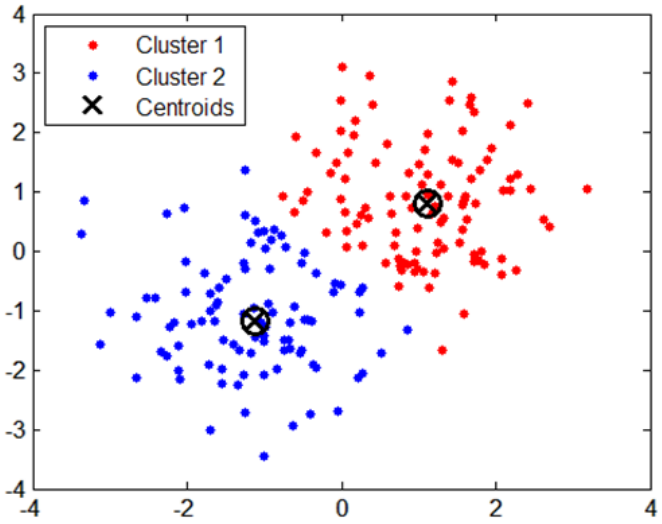
1. Map – In this task individual elements of data are broken down by Map function into key-value pairs [k1,v1….kn,vn]. It is done by the means of Mapper class.
2. Reduce- the output produced by the map function is passed as an input to the Reduce function, it combines all the key-values pair and produces a filtered data set. It is done by the means of Reducer class.

The functioning of Map and Reduce together



K-Means

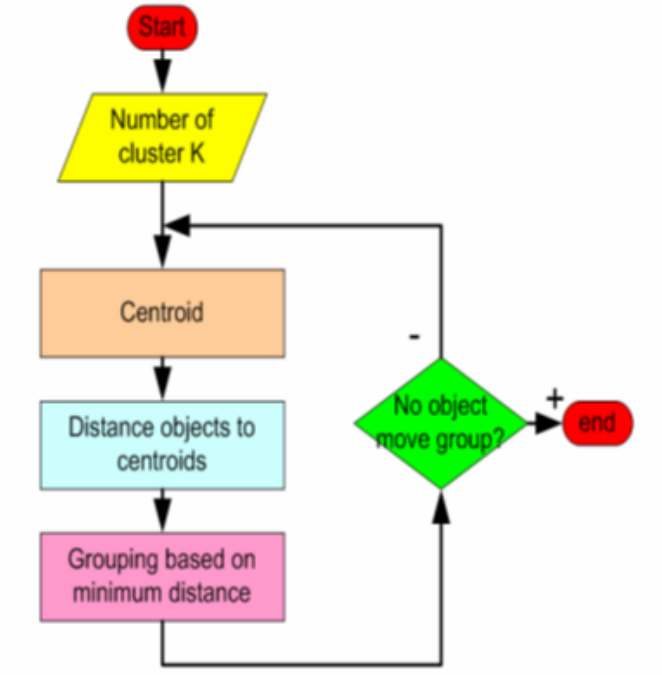
It is a traditional partitioning clustering algorithm which uses centroid of a cluster to perform the partition. Objects having similar properties are placed under one cluster. Objects within a same cluster should have intracluster similarity. The distance between an object and centroid is calculated using Euclidean's distance formula. For each object present in cluster the distance of object from its centroid is squared and the distance are summed up.



K-means with clusters and centroids

K-means Algorithm

- ✓ Form the clusters of data.
- ✓ Select k objects from data as initial cluster centers.
- ✓ Based on the mean values of the objects present in cluster, assign each object a cluster which contains the most similar number of objects.
- ✓ Keep on calculating the mean value of the object and cluster until same values are obtained.



How is k-means related to Hadoop?

Hadoop uses MapReduce programming model for processing of data. The advantage is that MapReduce

programs are implemented in parallel. The major step in implementing the K-Means algorithm in MapReduce programming model is to manage the input and output of the application. Since the MapReduce programing model requires key/value pairs to be submitted to the MapReduce job, the input to K-Means algorithm must be prepared as key/value pairs. The cluster center is selected as the key and the value will be the vector of data set.

To implement Map and Reduce routines two files are very essential, one is that stores the initial cluster centroids and the other that stores the vector form of data set to be clustered. Having these two input files ready in our hand, clustering of data can be done. The Map and Reduce routines are designed using the steps described in the K-Means algorithm. Both of the input file must be copied into HDFS from the local filesystem before the implementation starts because Hadoop reads input from its own filesystem called HDFS, rather than any other local filesystem. The Map function takes the two files as the input, the initial cluster center file in HDFS form the key field and the other file consisting of vector form of data set forms the value. As the algorithm steps proceed the distance calculation from cluster center to each point in the data set is performed in the Map function, the suitable code is written for the distance calculation, simultaneously recording the cluster to which the given vector is nearest. After processing all the vectors, the vectors are assigned to the nearest cluster. As soon as the Mapper finishes its task the computation is recalculated until it reaches a convergent point. This recalculation part goes into the Reduce routine, it also restructures the clusters to avoid the clusters with extreme size that is the clusters with too fewer data vectors or too many data vectors. This newly created clusters are written back to the disk which will be loaded as input to the next iteration.

III. RESULTS AND DISCUSSION

Techniques Used In K-Means.

There are 3 basic techniques which is used in k-means algorithm and are as follows

1. Forgy/Lloyd Algorithm.
2. MacQueen Algorithm.
3. Hartigan and Wong Algorithm.

Forgy/Lloyd Algorithm

The Lloyd algorithm and the Forgy's algorithm are both batch centroid models. A centroid is the geometric center of a convex object and can be thought of as a generalization of the mean. Algorithms where transformative steps are applied to all cases at once are known as Batch algorithms. They are well suited to analyze large data sets, since the incremental k-means algorithms require to store the cluster membership of each case or to do two nearest-cluster computations as each case is processed, which is computationally expensive on large datasets. The difference between the Lloyd algorithm and the Forgy algorithm is that the Lloyd algorithm considers the data distribution discrete while the Forgy algorithm considers the distribution continuous.

Steps involved in Forgy/Lloyd Algorithm:

1- Choose the number of clusters
2- Choose the metric to use
3- Choose the method to pick initial centroids
4- Assign initial centroids
5- While metric (centroids, cases)>threshold
a. For i <= nb cases
i. Assign case to closest cluster according to metric
b. Recalculate centroids.

MacQueen Algorithm

The MacQueen algorithm is an iterative (also known as incremental or online) algorithm. The main difference with Forgy/Lloyd's algorithm is that the centroids are recalculated every time a case change subspace and also after each pass through all cases. The centroids are initialized the same way as in the Forgy/Lloyd algorithm and the iterations are as follow. For each case in turn, if the centroid of the subspace it currently belongs to is the nearest, no change is made. If another centroid is the closest, the case is reassigned to the other centroid and the centroids for both the old and new subspaces are recalculated as the mean of the belonging cases. The algorithm is more efficient as it updates centroids more often and usually needs to perform one complete pass through the cases to converge on a solution.

Steps involved in MacQueen Algorithm:

1- Choose the number of clusters
2- Choose the metric to use
3- Choose the method to pick initial centroids
4- Assign initial centroids
5- While metric (centroids, cases)>threshold
  a. For i <= cases
    i. Assign case i to closest cluster according to metric
    ii. Recalculate centroids for the two affected clusters

b. Recalculate centroids.

Hartigan & Wong algorithm

This algorithm searches for the partition of data space with locally optimal within-cluster sum of squares of errors (SSE). It means that it may assign a case to another subspace, even if it currently belong to the subspace of the closest centroid, if doing so minimizes the total within-cluster sum of square .The cluster centers are initialized the same way as in the Forgy/Lloyd algorithm. The cases are then assigned to the centroid nearest them and the centroids are calculated as the mean of the assigned data points. The iterations are as follows. If the centroid has been updated in the last step, for each data point included, the within-cluster sum of squares for each data point if included in another cluster is calculated. If one of the cluster sum of square is smaller than the current one (SSE1), the case is assigned to this new cluster.

Steps involved in Hartigan & Wong Algorithm

1- Choose the number of clusters
2- Choose the metric to use
3- Choose the method to pick initial centroids
4- Assign initial centroids
5- Assign cases to closest centroid
6- Calculate centroids
7- For j <= nb clusters, if centroid j was updated last iteration
a. Calculate SSE within cluster
b. For i <= nb cases in cluster
  i. Compute SSE for cluster k!= j if case included.
  ii. If SSE cluster k < SSE cluster j, case change cluster.

Advantages of Hadoop

- ✓ Efficiency- It is highly efficient as it automatically distributes the data and work across the machines and utilizes the parallel processing capability of the CPU.
- ✓ Flexibility- It provides flexibility by adding or removing servers dynamically from the cluster without any interruption.
- ✓ Compatibility- As Hadoop is open source framework and is written in Java language it is compatible with most of the systems.
- ✓ Resilient to Failure- Hadoop does not depend on the hardware to provide fault tolerance, it has its own library than can detect and handle failures.
- ✓ Scalable- Hadoop is highly scalable platform because it can handle large amount of data across multiple machines simultaneously.
- ✓ Disadvantages of Hadoop.
- ✓ Not fit for small data – Hadoop is designed for high capacity due to which it lacks the ability to efficiently perform on small data sets.
- ✓ Security Concerns- At the storage and network level encryption is not provided by Hadoop which puts crucial data at huge risk.
- ✓ Stability Issues- It is an open source platform so it is created by ideas and contributions of many developers. Improvements are made constantly and companies are recommended to work on latest stable version of Hadoop.
- ✓ Inability to process random data- Since Hadoop uses MapReduce, it has limitations in batch-orientation which restricts it to access, process and serve real time data or random data.

IV. REFERENCES

[1]. The k-means clustering technique: General considerations and implementation in Mathematica, Laurence Morissette and Sylvain Chartier, Université d'Ottawa.

[2]. Implementation of K-Means Clustering Algorithm in Hadoop Framework Uday Kumar Sr, Naveen D Chandavarkar, PG Scholar, Assistant professor, Dept. of CSE, NMAMIT, Nitte, India.

[3]. K-Means Clustering Tutorial, By Kardi Teknomo, Teknomo, Kardi. K-Means Clustering Tutorials. http:\\people.revoledu.com\kardi\ tutorial\kMean\.

[4]. Parallel Clustering of large data set on Hadoop using Data mining techniques, Kaustubh S. Chaturbhuj, Dept. of Computer Science and Engineering, YCCE Nagpur, India, , Mrs. Gauri Chaudhary, Dept. of Computer Science and Engineering, YCCE, Nagpur, India.
[5]. Apache documentation on Hadoop.