International Journal of Multidisciplinary Allied Research Review and Practices

IJMARRP ISSN No: 2455-1570 (Volume 3, Issue 7, July 2016)

Implementation paper on K-Means clustering using hadoop

[1]Meenakshi , [2]Bhupender Yadav

Gurgaon Institute of Technology &Management, Gurgaon

Abstract: Data mining is the field of computer's era. As we know that we are producing tons of data daily and stored in data warehouse .Data mining is used to mine the data and produce some interesting patterns and information which we want to know. There are two kind of learning supervised and unsupervised. In unsupervised learning, we have different algorithms to learn the patterns. In the previous paper we have discussed how k means clustering algorithm works using hadoop . In this paper we are discussing the implementation of k means clustering algorithm using map reduce programming coding in hadoop framework over distributed manner. The K-Means Clustering is the technique of data mining to classify objects or things into groups in an unsupervised manner.

KEYWORDS: K-Means Clustering, MapReduce, Hadoop

Clustering

Clustering is an unsupervised learning in which data are categorized according to their similarities into different groups. These groups are called clusters. So, cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Clustering algorithms are of various types. These are partition-based algorithm (K-Mean), hierarchical-based algorithm, density-based algorithm (DBSCAN) and grid-based algorithm. Its main distinctiveness is the fastest processing time. Depending on the requirements and data sets we apply the appropriate clustering algorithm to extract data or group the data according to our queries from them

K-Means Algorithm

K-means clustering is mainly used for a number of classification applications. Because k-means is run on such large data sets, and because of certain characteristics of the algorithm, it is a good candidate for parallelization. K-means clustering is a classical clustering algorithm that uses the technique to partition a number of data points into k clusters.

K-Mean MapReduce Algorithm

Step1: Initially randomly centroid is selected based on data. In our implementation we used 3 centroids

Step2: There are two Input files, one contains initial centroids and other contains data.

Step3: In Mapper class "setup" function is used to first open the file and read the

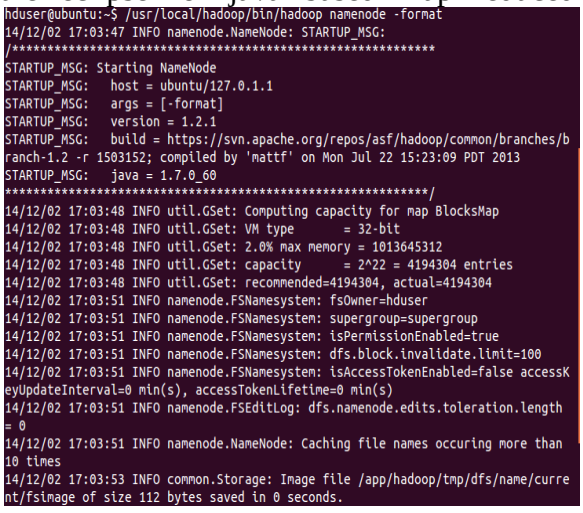
centroids and store in the data structure(like Array List)

Step4: Mapper read the data file and gives the nearest centroid with the point to the reducer.

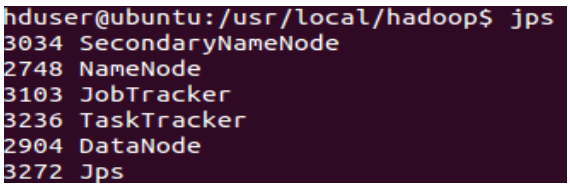
Step5: Reducer collects all this data and calculates the new corresponding centroids and emit.
Repeat step2 with new centrois.

- This is iterative method an continously updated.

we need to install the hadoop. Hadoop is installed on Linux operating system. So first we install virtual machine (VMware) .Secondly, Linux operating system on that virtual machine. Thirdly, install hadoop 1.0.4 (jar files) on the system. Finally install the eclipse for java based map reduced



Type jps to check whether all nodes are working properly.



Step6: In the job configuration, we are reading both files and checking

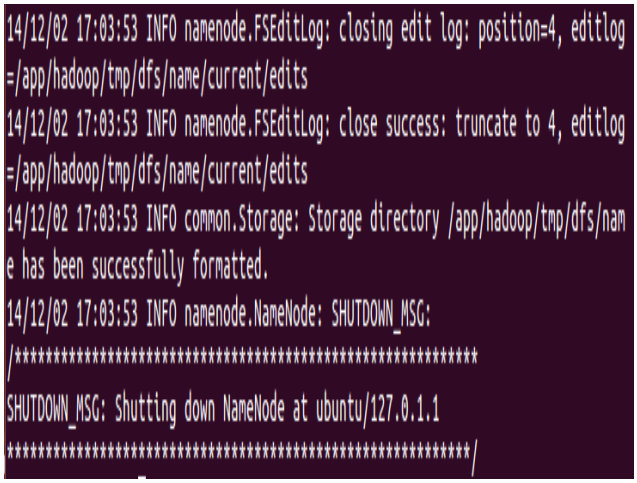
IF
Difference between old and new centroid is less than 0.1
THEN
Convergence is reached
ELSE

- **Implementation of K-means using Hadoop**

Experimental Setup

For Implementing K-Means on hadoop, first programming. Then check the system and activate the all nodes of hadoop by typing

>> start-all.sh comman on the command prompt.



K-Means Based on Map Reduce

We need some vectors. These vectors representing our data, and then we need k-centroids. These input data and centroids are in the form of vectors also, centroids file is ajust a subset of the input data vectors, but sometimes they are random points or points-of-interest to which we are going to cluster them. Since this is a Map Reduce (key, value) version. This is really simple, because we are just using a vector, a vector can be a cluster center as well. So we treat our cluster center-vectors always like keys, and the input vectors are simple values. [6]

There are three kinds of job level at Map Reduce programming.

1.driver level

2. mapper level

3. reducer level

Mapper level

The input dataset is stored on HDFS as a sequence file of pairs, each of which represents a record in the dataset. The input of the mapper is the key and value pair and key is the offset in bytes of this record to the start point of the data file, and the value is a string of the content of this record. The dataset is splitted by inputspiltter and this input is given to all mappers. As a result, the distance computations are parallel executed. For each map task, K-Means construct a global variant centers which is an array containing the information about centers of the clusters. Given the information, a mapper can compute the closest center point for each sample. The intermediate values are then composed of two parts: the index of the closest center point and the sample information. The pseudo code of map function is shown in Algorithm.

Algorithm 1. map (key, value)

Input: centers values, the offset key, the sample value

Output: pair, where the key' is the index of the closest center point and value' is a string comprise of sample information

1. Construct the sample instance from value;

2. SmallestDistance = Double.MAX VALUE;

3. index = -1;

4. For j=0 to centers.size() do

Distance = calculateDistance(instance, centers[j]);

If distance <SmallestDistance

{

SmallestDistance = distance;

index = j;

}

5. End

6. Take index as key';

7. Construct value' as a string comprise of the values of different dimensions;

8. output < key , value > p

IJMARRP ISSN No: 2455-1570 (Volume 3, Issue 7, July 2016)

```
public class KMeansMapper extends Mapper<ClusterCenter,Vector, ClusterCenter, Vector> {

    List<ClusterCenter> centers = new LinkedList<ClusterCenter>();


    protected void setup(Context context) throws IOException,InterruptedException {
    super.setup(context);
    Configuration conf = context.getConfiguration();
    Path centroids =new Path(conf.get("centroid.path"));
    FileSystem fs = FileSystem.get(conf);
    SequenceFile.Reader reader=new SequenceFile.Reader(fs,centroids,conf);
    ClusterCenter Key=new ClusterCenter();
    IntWritable value=new IntWritable();

    while (reader.next(Key,value))
    {
    centers.add(new ClusterCenter());
    }
```

Reducer Level

The input of the reduce function is the data obtained from the map function of each mapper. In reduce function, we can sum all the samples and compute the total number of samples assigned to the same cluster. Therefore, we can get the new centers which are used for next iteration.

The pseudo code for reduce function is shown in Algorithm

Algorithm 3. Reduce (key, V)

Input : key is the string nameof center

: V is the list of the partial sums from different host

Output: < key, value > pair, where the key' is the index of the cluster, value' is a string representing the new center

1. Initialize one array record the sum of value of each dimensions of the samples contained in the same cluster, e.g. the samples in the list V;

2. Initialize a counter NUM as 0 to record the sum of sample number in the same cluster;

3. While (V.hasNext ())

{

Construct the sample instance from V.next ();

Add the values of different dimensions of instance to the array NUM += num;

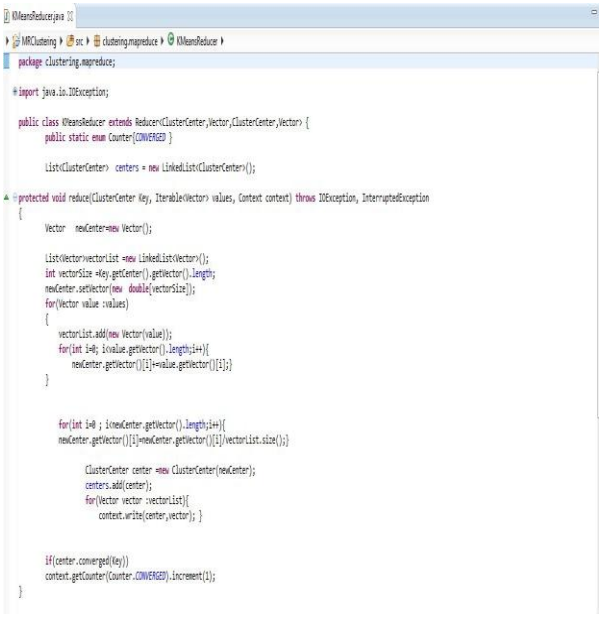
4. }

5. Divide the entries of the array by NUM to get the new center's coordinates;

6. Take key as key';

7. Construct value' as a string comprise of the center's coordinates;

8. Output < key, value > pair;



Result

In our implementation, we found that the data is divided into no. of clusters which we have given. The data is divided into different groups or clusters using k means algorithm. In k-means, it is clear that the calculation of the Euclidean distance can be done between the input values in the form of vectors and the centroids that are stored in centroid fileis used for splitting the data into individual subgroups and clustering the datasets in each subgroup separately done by the map function defined in a mapper. In recalculating new centroid vectors, we divide the input data vectors into subgroups and compute the sum of vectors in each subgroup in parallel, and finally the reducer will add up the results of all mappers and calculate the sum and compute the new centroids.

Conclusion and Future Work

The conclusion on this topic is that K-Means clustering using Hadoop was required just to improve the time efficiency with large datasets and and multi attributes. while doing this thesis work, we realized that clustering has got a many applications and its useis increasing with the increase of the use and applications of web. Also, we found that it'snot just the number of uses but the ways in which clustering is used in variousapplications is changing and this motivated us to think of some new algorithm or to improve our coding for K-Means Map Reduce programming for improving the time efficiency to complete the clustering work.

References:

1. Meenakshi Gupta and Poonam Yadav ,IJRAET-Feb2016, Vol-4-I-2(NCRISTM), A Survey Paper on K-Means using Hadoop
2. D. Venkatavara Prasad, Sathya Madhusudanan and Suresh Jaganathan, ARPN Journal of Engineering and Applied Sciences, Vol. 10, No- 5, march2015, uCLUST – A new algorithm for clustering unstructured data.
3. T.Nelson Gnanaraj, Dr.K.Ramesh Kumar, N.Monica, International Journal of Advances in Computer Science and Technology,Volume 3, No.2, Feb 2014, structured and unstructured data
4. Apache Hadoop. http://hadoop.apache.org/
5. Clustring_mapreduce.pdf by suny Buffalo
6. Thomas.blogspot.in/2011/05/k-means-clustering-with-mapreduce