International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 9, September- 2014

Exploring Efficiency of Mapreducing Techniques using ESAMR Algorithm

Discovering Number of Clusters using Kmeans Algorithm

Neha Nayak
Department of Information Technology
V.E.S Institute of Technology
Chembur,Mumbai.

Prasad Chitnis
Department of Information technology
Datta Meghe College of Engineering
Navi Mumbai.

***Abstract*-In recent year the amount of data stored has increased exponentially so it is important to keep it properly. Our paper presents a technique which assist in processing extensive amount of data. Big data the term itself represents the type of data we are dealing with. When an employee or any trained professional have to work on data pertaining to this range it becomes essential that they use techniques which not only improve the performance of the system but also presents a better way of doing the work. In our paper we have explored the usage of map reducing techniques in the world that comprises of huge amount of data .For that we require scheduling algorithms which are one of the major factors when it comes to dealing with big data.Scheduling algorithm compresses the necessary data without causing any variation in itss outcome. In our paper we present an overview of the working of one such scheduling algorithm i.e. ESAMR using k means algorithm.**

Keyword-Bigdata , Mapreduce ,ESAMR ,K-means

I. INTRODUCTION

Big data is a term that is used to represent large amount of data and resources which we are unable to process using traditional data processing applications. Big data is a term which was unheard of 5-10 years ago. Let's put it into perspective-over 90% of the current data was created over the last two years. And data will continue its growth as it is being collected from information-sensing mobile devices, aerial sensory technologies, software log, cameras, microphones, radio frequency identifiers (RFID) readers, and wireless sensor networks. Collecting large amount of data from different sources and storing it systematically is one aspect and using this structured or unstructured data to get important information for the benefit of the company is another aspect. Big data helps its users to find required information quickly, efficiently and cost effectively. After considering that average enterprise will need to manage 50 times more information by the year 2020 while increasing IT staff by only 1.5 percent the importance of big data is understood. Now companies have to handle integrating big data models into existing infrastructure to keep a check on the efficiency and privacy of the system. The other noticeable drawback of having unstructured data was that security information and event management tools weren't designed to handle unstructured data, But now we have big data applications that are now an integral part of the security management as they help to clean, organize and solve query even in heterogeneous, incomplete and noisy data. Another problem that arises is how to deal with the ever increasing set of information in big data ?. In this paper we present one such technique that can be used when we encounter such problem which is referred to as Hadoop's MapReduce technique. It's a programming paradigm that takes a set of intensive data processes it and distributes the result across endless number of Hadoop clusters .Introduction of such techniques have been the turning point in supporting the needs of enormous processing capabilities required by bigdata.

Hadoop's MapReduce is one such technique used for processing large number of information and responsible for producing compressed dataset. . This technique is currently the talk of the town and has attracted multiple programmers working with bigdata worldwide. And to say the least Mapreduce is not over hyped, it has wide range applications including distributed pattern-based searching web access log stats, inverted index construction etc .Mapreduce can solve parallelized problems over a large data using many nodes which makes it suitable for bigdata . It can also be used in multi core or many core systems making it very useful. Mapreduce can take advantage of the locality of the data where it is stored by processing on or near the stored data so reducing the transmission time. MapReduce is said to be reliable due to the following procedure that it follows, Mapreduce assigns a number of jobs on a set of data to each node on the network and expects it to return results after a certain time period, But if a node fails to return results after a fixed time period the master node declares that node as dead and assigns its work to other nodes.

This programming model comprises of two distinct tasks. First is the map job which reads all the <key,value>pairs from the given input file and processes it into intermediate <key,value>.The reduce job merges the transformed intermediate values and associates it with their respective intermediate keys. It takes the output of the map job and compresses it into smaller set of tuples. The

performance of the MapReduce task is determined by the efficiency of the scheduler .An efficient MapReduce scheduler turns out to be extremely beneficial in cases where response time is crucial ,this is achieved by assigning the reduce operations to the node that stores the data that is being operated on or is in the same rack .It helps in avoiding all the unnecessary data transmission. In this paper we provide an overview of one such algorithm. Mapreduce is perfect for Bigdata as it has many nodes at its disposal so that it can distribute the work so that there is no pressure on the single node therefore decreasing the chances of a complete breakdown and has a master node to control [i]all the other nodes so just in case one of them fails it has many nodes to do the work. Of all the available mapreduce algorithm ESAMR is the algorithm which causes the least amount of error and finds the slow tasks accurately. ESAMR is able to do so as it divides the tasks into different clusters and then based on which cluster it is in it calculates it's remaining execution time.

- EXAMPLE

Let's look at a simple example. Assume you have five files, and each file contains two columns (a key and a value in Hadoop terms) that represent a year and the corresponding number of students present in lecture for the various measurement days. This data might be structured or unstructured but the key principles we're covering here remain the same. Either way, in this example, year is the key and number of students is the value.

BE,20
TE,25
SE,22
FE,32
BE,4
FE,33
SE,18

Out of all the data we have collected, we want to find the maximum number of students present in lecture across all of the data files. Using Mapreduce we assign this job to five mappers who find the most number of students present in that year.Consider the following, the results produced from one mapper(Lecturer) task for the data above would look like this:

(BE, 20) (TE, 25) (SE, 22) (FE, 33)

Let's assume the other four mapper tasks produced the following intermediate results:

(BE, 18) (TE, 27) (SE, 32) (FE, 37)(BE, 32) (TE, 20)(SE, 33) (FE, 38)(BE, 22) (TE, 19) (SE, 20) (FE, 31)(BE, 31) (TE, 22) (SE, 19) (FE, 30)

All this 5 output given by the mappers is used by the reduce tasks to give us a single word output.

(BE, 32) (TE, 27) (SE, 33) (FE, 38)

II. ESAMR

ESAMR is an algorithm which improves the response time of the task in relation to the job that's been performed. It identifies all the slow tasks in a most appropriate manner. Since all the slow tasks takes up most of the execution time required for the whole job therefore an efficient scheduling algorithm is required which can re-execute all such slow tasks .Moreover ,because of the heterogeneity of the hardware the time taken for processing the task is different on every node. It uses the historical information on each set of node and differentiates it into k clusters using a k means clustering algorithm.

ESAMR uses temporary M1 weight of the node to decide how much time is remaining for that node to complete its execution. In this system as soon as a running job completes some task on a node, ESAMR record's the job's M1 on the node. So now ESAMR uses this information i.e M1 to calculate TimeToEnd(time remaining for the processing to be completed) for the node that has not completed its execution. This method also helps in finding out the slow tasks and which tasks need to be re-executed. Once the processing is done by the map job the outcome is provided to the reduce job in the second stage. To find out the slow taks accurately we divide them using historical information on every node into K clusters using k clustering algorithm. Once the complete job is performed ESAMR computes the job's stage weight on each node and this information gets saved as a part of the historical information. Once again the scheduling algorithm, using k-means algorithm classifies the newly obtained historical information into k clusters and the computed stage weight gets stored for each of the k clusters. Since ESAMR efficiently re-executes all the identified slow task thereby improving the overall execution time, it's said to be one of the widely used scheduling algorithm for MapReduce programming model.

III. K-MEANS ALGORITHM

K-means algorithm is one of the simplest learning algorithm that solves even the toughest of clustering issues. This algorithm requires the number of clusters to be specified before the processing starts. However the estimation of the number of clusters present is generally done by using trial and error process. In our paper we present the algorithm that finds out the value of k in a most efficient manner.

A. WORKING OF THE ALGORITHM:

Initially we find k centroids, one for each cluster. The election of centroid should be done smartly because epending upon the location of the centroid the result differs. So one of the option is to place them as far as possible. Then we choose specific points from each data set that we are using and associate it with the nearest centroid .After this stage we can re-calculate the k new centroids . Once we obtain the resulting new k centroids a binding needs to be done between the same data set points and the nearest centroid. Because of which a loop is generated. This loop which gets formed helps us to notice the transition in the location of the centroid slowly by slowly after which no more changes occur .i.e. the centroid stops moving. The algorithm also helps in minimizing an objective function.

K-mean algorithm in java code is as given below:

```
import java.io.*;
import java.lang.*;
import java.util.*;
```

```
class Kmean
{
public static void main(String args[])throws IOException
{
int N=10;
int A[]={10,15,91,45,67,4,83,71,33,12};
int i,M1,M2,a,b,n=0;
boolean f=true;
float s1=0,s2=0;
a=A[0];b=A[1];
M1=a;
 M2=b;
int Cluster1[]=new int[10];
int Cluster2[]=new int[10];
for(i=0;i<10;i++)
   System.out.print(A[i]+ "  ");
System.out.println();
do
{  int p=0,q=0;
 n++;
 for(i=0;i<10;i++)
 {
   if(Math.abs(A[i]-M1)<=Math.abs(A[i]-M2))
   {   Cluster1[p]=A[i];
      p++;
   }
   else
   {   Cluster2[q]=arr[i];
      q++;
   }
 }
   System.out.println();
   for(i=0;i<10;i++)
      s1=s1+Cluster1[i];
   for(i=0;i<10;i++)
      s2=s1+Cluster2[i];
   a=M1;
   b=M2;
   M1=Math.round(s1/k);
   M2=Math.round(s2/j);
   if(M1==a && M2==b)
      fl=false;
   else
      fl=true;
    System.out.println("After iteration "+ n +" ,Cluster 1 :\n");


   for(i=0;i<10;i++)
      System.out.print(Cluster1[i]+ "\t");
   System.out.println("\n");
   System.out.println("After iteration "+ n +" , cluster 2 :\n");
   for(i=0;i<10;i++)
      System.out.print(Cluster2[i]+ "\t");

}while(f);

   System.out.println("Final cluster 1 :\n");
   for(i=0;i<10;i++)
      System.out.print(Cluster1[i]+ "\t");

   System.out.println();
   System.out.println("Final cluster 2 :\n");
   for(i=0;i<10;i++)
      System.out.print(Cluster2[i]+ "\t");
 }
}
```

B. Example

Suppose that we have several objects a1,a2,a3,a4 and each of these objects have two attributes x1,x2,x3,x4 and y1,y2,y3,y4 and we have to separate it into two clusters.

Object name	Attribute X	Attribute Y
A1	1	1
A2	2	1
A3	4	3
A4	5	4

Now consider each object as a point with its attribute's as it's (x,y) co-ordinates.
Initially assume values of centroid X and Y. For now for X take value of A1 and Y take value of A2.

Now calculate distance of all the objects from these two centroids using distance formula.

Object name	Distance from centroid X(1,1)	Distance from centroid Y(2,1)
A1	0	1
A2	1	0
A3	3.61	2.83
A4	5	4.24

Now we divide the information into two clusters based on their values,A1 is nearer to X so it is assigned to group 1,had it been nearer to Y we would have assigned it group 2.

Therefore Group 1={A1},Group 2={A2,A3,A4}.

Now to find new value of co-ordinates of X and Y take average of the co-ordinates of it's group member's.

Therefore new value of X={1,1} and Y={11/3,8/3}.

This process is done till a point where the previous group members match with the new one. So now calculating Distance from centroid

Object name	Distance from centroid X(1,1)	Distance from centroid Y(11/3,8/3)
A1	0	3.14
A2	1	2.36
A3	3.61	0.47
A4	5	1.89

Now dividing them into clusters on the basis of their distance.

Group 1={A1,A2} ,Group 2={A3,A4}.

Now calculate new values of X and Y,

X={1.5,1} and Y={4.5,3.5}.

Now calculate the distance of the objects from centroid X(1.5,1) and Y(4.5,3.5).

Object name	Distance from centroid X(1.5,1)	Distance from centroid Y(4.5,3.5)
A1	0.5	4.3
A2	0.5	3.54
A3	3.2	0.71
A4	4.61	0.71

Now dividing them into clusters on the basis of their distance.

Group 1= {A1, A2} ,Group 2= {A3, A4}.

As we can see the new group is exactly same as the group before, hence we stop this process here as we have divided the objects into respective clusters.

Therefore the final answer is as following

Object name	Distance from centroid X(1.5,1)	Distance from centroid Y(4.5,3.5)	Cluster Number
A1	0.5	4.3	1
A2	0.5	3.54	1
A3	3.2	0.71	2
A4	4.61	0.71	2

IV. CONCLUSION:

Through our paper we have tried to demonstrate the efficiency of MapReducing techniques while dealing with large amount of data.The processing of huge data through ESAMR algorithm and one of the technique of finding the number of clusters to classify historical information using k means algorithm.

V. REFERENCES

[1].http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html
[2]. http://www.ee.columbia.edu/
[3].http://www.facebook.com/l.php?u=http%3A%2F%2Fresearch.ijcaonline.org%2F&h=wAQEA0zoG
[4]. http://www.webopedia.com/