Improving Search Performance for Bigdata Processing Using Machine Learning Algorithm

G.Jeeva, E.K.Subramanian

***Abstract*—** Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Data mining algorithms of increasing sophistication are being implemented in MapReduce, bringing new challenges for performance measurement and tuning. We focus on Hadoop framework which is distributed model frameworks for processing large scale data intensive applications for different purposes. Hadoop provides HDFS and Map Reduce but it still to improve the process. So it use Mahout with hadoop it perform machine learning tasks on voluminous amounts of data. These tasks have application in fields such as pattern recognition, data mining, bioinformatics, and recommendation systems. In this existing system it evaluate the performance of clustering algorithms and classification algorithms supported by Mahout within two different cloud runtimes, Hadoop and Granules. Cluster analysis is the most important data mining methods. Efficient parallel algorithms and frameworks are the key to meeting the scalability and performance requirements entailed in such scientific data analyses. In order to improve the deficiency of the long time in large-scale datasets clustering on the commodity system. In this project to propose the new k-means algorithm based on MapReduce framework. This algorithm runs on Hadoop cluster. The results show that the MapReduce framework K-Means clustering algorithm can obtain a higher performance.

Index Terms— Big data, cloud, Hadoop, Machine Learning, QOS, and Map reduce,K-Means.

1 INTRODUCTION

THE K-means is one of the most frequently used clustering algorithms. It is simple and straightforward and has been successfully applied during the last few years. Under the assumption that datasets tend to be small, research on clustering algorithms has traditionally focused on improving the quality of clustering. However, many datasets now are large and cannot fit into main memory.

Mahout/Hadoop can be a promising and inexpensive solution to solve problems with large data sets. However, there is a lack of studies regarding its performance. With this work we wanted to study the gain in runtime when using this solution and also to test if there was any loss in the clusters quality. To meet those aims we used to set up a small cluster with different configuration. We chose to study K-means since it is one of the most popular and easy algorithms.

Clustering refers to the process of grouping samples into different classes based on their similarities. Samples within a class have high similarity in comparison to one another but are very dissimilar to samples in other classes. These groups or classes are called clusters. Clustering algorithms can divide the samples into clusters automatically without any preconceptions about what kind of groupings should be found. Within the context of machine learning, clustering is considered to be a form of unsupervised learning since there is no target variable to guide the learning process. Unlike classification, clustering and unsupervised learning do not rely on predefined classes and class-labelled training examples.

- *G.Jeeva is currently pursuing masters degree program in School ofCcomputer,Information and Mathematical Sciences in B.S.AbdurRahman University,Chennai,India,jeeva10248@gmail.com.*
- *E.K.Subramanian is currently Assistant Professor in School ofCcomputer,Information and Mathematical Sciences in B.S.AbdurRahman University,Chennai,India,jeeva10248@gmail.com.*

For this reason, clustering is a form of learning by observation, rather than learning by examples . The number of clusters, size, and shape are not in general known in anticipation, and each of these parameters must be determined by either the user or the clustering algorithm . Clustering has been used in many areas, including data mining, statistics, biology, and machine learning. Clustering can also be used for outlier detection, where outliers (values that are “far away” from any cluster) may be more interesting than common cases . One useful application example is fault detection. The concept of clustering is not particularly new, but it is still an important topic of research. The Semantic Web, as originally envisioned, is a system that enables machines to "understand" and respond to complex human requests based on their meaning. Such an "understanding" requires that the relevant information sources be semantically structured.

The two major types of cluster algorithms are hierarchical and partitional. The first type produces a hierarchical decomposition of the data set into partitions. It merges or splits the partitions into clusters using a similarity criterion. In the second type, an algorithm creates an initial division of the data set, according to a given number of partitions. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion for partitioning is a combination of high similarity of the samples inside of clusters with high dissimilarity between distinct clusters.

K-means clustering is a widely used partition algorithm. It partitions the samples into clusters by minimizing a measure between the samples and the centroid of the clusters. Euclidean distance is a similarity criterion extensively used for samples in Euclidean space. Manhattan distance is a simpler measure also for Euclidean space. Cosine distance and Jaccard is often employed for documents clustering .The K-means clustering is simple but it has high time complexity when the data

sets are large. In these circumstances the memory of a single machine can be a restriction. As a consequence it has not been used in the past with large data sets.

Hadoop is a software framework which allows the effortless development of cloud computing systems. It supports data intensive distributed applications on large clusters built of commodity hardware. This framework is designed to be scalable, which allows the user to add more nodes

Hadoop uses a distributed computing paradigm named MapReduce. The Map Reduce programming model consists of two user defined functions: map and reduce, specified on a job. The job usually splits the input dataset into independent blocks which are processed by the map tasks in a parallel way. Hadoop sorts the outputs of the maps, which are then the input to be processed by the reduce tasks . Hadoop uses a distributed file system HDFS, which creates multiple replicas of data blocks and distributes them on computer.

Nodes throughout a cluster to enable reliable, extremely rapid computations Mahout is a machine learning library that runs over a Hadoop system. It has a collection of algorithms to solve clustering, classification and prediction problems. It uses MapReduce paradigm which in combination with Hadoop can be used as an inexpensive solution to solve machine learning problems. Mahout contains various implementations of clustering, like K-means, fuzzy K-means, meanshift and Dirichlet among others. To input the data for Mahout clustering it is necessary to do some procedures first. If the data is not numerical it has to be first pre-processed. It is required then to create vectors. If the data set is sparse it allows the user to create sparse vectors that are much more compact. The vectors are finally converted to a specific Hadoop file format that is *Sequence File*. The K-means clustering algorithm takes the following input parameters:

A *Sequence File* containing the input vectors.
A *Sequence File* containing the initial cluster centres. If not present it will attribute them randomly.
A similarity measure to be used.

The vector implementation used for the input files. As output the user gets the centroids coordinates and the samples attributed to each cluster. The output files are in *Sequence File* format. Mahout provides the necessary tools for file conversion and creating the vectors . There are three stages in a K-means job .

Initial stage: the segmentation of the dataset into HDFS blocks; their replication and transfer to other machines; according to the number of blocks and cluster configuration it will be assigned and distributed the necessary tasks.

Map stage: calculates the distances between samples and centroids; match samples with the nearest centroid and assigns them to that specific cluster. Each map task is processed with a data block.

Reduce stage: recalculates the centroid point using the average of the coordinates of all the points in that cluster.The associated points are averaged out to produce the new location of the centroid. The controids configuration is feedback into the Mappers. The loop ends when the centroids converge. After finishing the job Mahout provides the K-means time, the centroids and the clustered sample

2 PREVIOUS RELATED WORK

Kathleen Ericson∗, Shrideep Pallickara presented "On the performance of high dimensional data clustering and classification algorithms "It include an analysis of our results for each of these algorithms like clustering and classification in a distributed setting, As well as a discussion on measures for failure recovery.

Todd D. Plantenga, Yung Ryn Choe, Ann Yoshimura presented "Using Performance Measurements to Improve MapReduce Algorithms" It bringing new challenges for performance measurement and tuning. We focus on analyzing a job after completion, utilizing information collected from Hadoop logs and machine metrics. It describes examples where measurements helped diagnose subtle issues and improve algorithm performance and provide set of tools to measure the performance of the algorithm.

Ping ZHOU , Jingsheng LEI, Wenjun YE presented "Large-Scale Data Sets Clustering Based on MapReduce and Hadoop" The propose algorithm overcome the deficiency of the large scale datasets clustering on the single machine . The advantage of the parallelism of MapReduce to design a parallel K-Means clustering algorithm based on MapReduce. This algorithm can automatically cluster the massive data, making full use of the Hadoop cluster performance. It can finish the text clustering in a relatively short period of time. Experiments show that it achieves high accuracy.

Weizhong Yana,∗, Umang Brahmakshatriya a, Ya Xuea, Mark Gilder b, Bowden Wisec "p-PIC: Parallel power iteration clustering for big data" This paper propose p- pic algorithm and compared to the existing machine learning algorithm. Compared to traditional clustering algorithms, PIC is simple, fast and relatively scalable it requires the data and its associated similarity matrix fit into memory, which makes the algorithm infeasible for big data applications**.**

Zhuo Tang · Junqing Zhou ·Kenli Li · Ruixuan Li "A MapReduce task scheduling algorithm for deadline constraints" It proposes an extensional MapReduce Task Scheduling algorithm for Deadline constraints in Hadoop platform: MTSD. It allows user specify a job's deadline and tries to make the job be finished before the deadline. Through measuring the node's computing capacity **.**The experiments show that the node classification algorithm can improved data locality observably to compare with default scheduler and it also can improve other scheduler's locality. we calculate the task's average completion time which is based on the node level. It improves the precision of task's remaining time evaluation.

A. Espinosa · P. Hernandez · J.C. Moure ·J. Protasio · A. Ripoll "Analysis and improvement of map-reduce data distribution in read mapping applications" It describe the data distribution patterns found in current Map-Reduce read mapping bioinformatics applications and show some data decomposition principles to greatly improve their scalability and performance It suggest a method to redesign scatter to gather map-reduce applications. Also we propose a list of Hadoop

performance optimizations specific for sequence alignment map-reduce applications describing the parameters used. their It impact on the execution and some criteria to define values to regulate them

Abhishek Verma · Brian Cho · Nicolas Zea · Indranil Gupta · Roy H. Campbell "Breaking the MapReduce stagebarrier"
It develop a method to break the barrier in Map Reduce in a way that improves efficiency. Careful design of our barrier-less Map Reduce framework results in equivalent generality and retains ease of programming. We motivate our case with, and experimentally study in barrier-less techniques in, a wide variety of Map Reduce applications divided into seven classes. This experiments show that our approach can achieve better job completion times than a traditional Map Reduce framework. We achieve a reduction in job completion times that is 25% on average and 87% in the best case.

3 CONCEPT AND PROBLEM DEFINITION

The objective of this project is clustering the dataset using k-means clustering to measure and improve the performance of large scale datasets clustering on the commodity system. To propose the new k-means algorithm based on MapReduce framework. This algorithm runs on Hadoop cluster with various configuration to measure and improve the performance of the system.

*A. Definition **1**:*

The distance between data points and the cluster center. The distance formula of data point x_i and cluster center k_j defined as following

$$d_{j,i} = \sqrt{(x_{i_1} - k_{j_1})^2 + (x_{i_2} - k_{j_2})^2 + \cdots + (x_{i_w} - k_{j_w})^2}$$

B. Definition 2 The density parameter:

The number of data points which is contained by a scope defined as density parameter. The scope is a round which takes space point of not statistics x_i as the center, as the radius. The greater the density of x_i, the greater the value of the density parameter are.

C. Definition 3 The core data points:

If the -neighborhood of a data point contains at least PTS min number of data points, then the data point called the core data point

D. Definition 4 The cluster center:

The first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster center.

E. Definition 5

$$d_{j,i} = \sqrt{(x_{i_1} - k_{j_1})^2 + (x_{i_2} - k_{j_2})^2 + \cdots + (x_{i_w} - k_{j_w})^2}$$

and the cluster).

4 SYSTEM ARCHITECTURE

The proposed system architecture is shown in Fig.1, that clearly outlines every module. The module broadly classifies various sub topics within each of the modules. The input and output of the software forms the boundaries in the given figure. The proposed work consists of four main modules such as 1.Preprocessing Dataset, 2. Clustering the data set, 3. Parallelising Implementation Using Maper and Reducer 4. Monitoring the Performance of data in node.

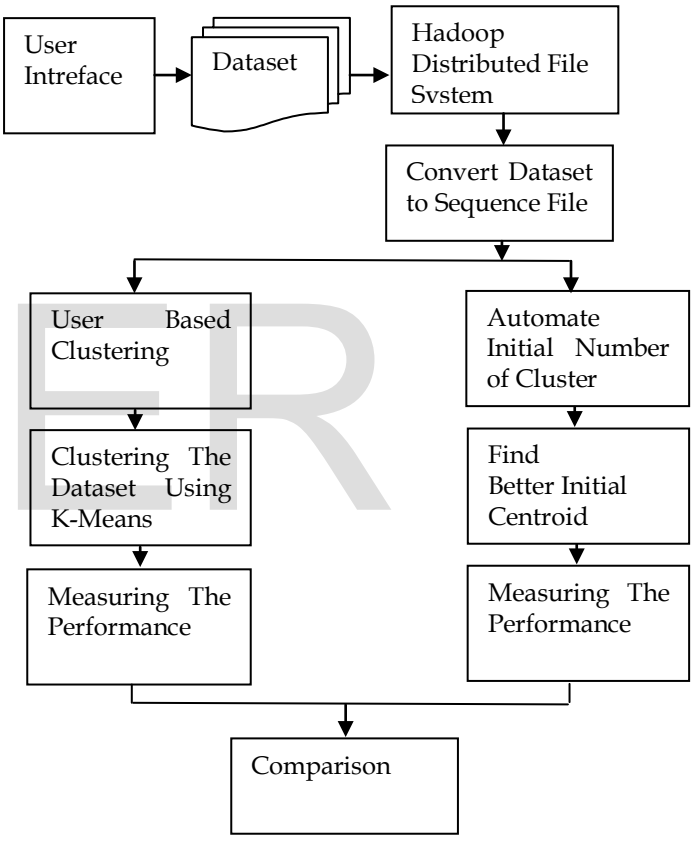

Fig.1. Improving search performance for big data processing using machine learning algorithm.

A. Preprocessing Dataset

The datasets have been pre-processed, with stop-word removal and stemming already applied. In addition, terms occurring in less than three documents have been eliminated.

B. Clustering the data set

K-means clustering is a widely used partition algorithm. It partitions the samples into clusters by minimizing a

measure between the samples and the centroid of the clusters. Euclidean distance is a similarity criterion extensively used for samples in Euclidean space. Manhattan distance is a simpler measure also for Euclidean space.

1. Choose a number of clusters k
2. Initialize centroid1 c1,… ck
3. For each data point, compute the centroid it is closest to (using some distance measure) and assign the data point to this cluster.
4. Re-compute centroids (mean of data points in cluster).

C. Clustering the dataset based on improved k-means.

In the paper the proposed algorithm have two parts one is to automate the number fo cluster and another method for finding better initial centroids followed by efficient way of assigning data points to appropriate clusters.

By using silhoutte method to automate the number of cluster The concept of silhouette width involves the difference between the within-cluster tightness and separation from the rest. Specifically, the silhouette width *s(i)* for entity *i* ε *I* is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Where *a(i)* is the average distance between *i* and all other entities of the cluster to which *i* belongs and *b(i)* is the minimum of the average distances between *i* and all the entities in each other cluster. The silhouette width values lie in the range from −1 to 1. If the silhouette width value for an entity is about zero, it means that that the entity could be assigned to another cluster as well. If the silhouette width value is close to −1, it means that the entity is misclassified. If all the silhouette width values are close to 1, it means that the set I is well clustered.

A clustering can be characterized by the average silhouette width of individual entities. The largest average silhouette width, over different K, indicates the best number of clusters.

In the proposed algorithm first we are checking, the given data set contain the negative value attributes or not. If the data set contains the negative value attributes then we are-transforming the all data points in the data set to the positive space by subtracting the each data point attribute with the minimum attribute value in the given data set. we will get the same Euclidean distance from the origin. This will result in incorrect selection of the initial centroids. To

overcome this problem all the data points are transformed to the positive space. If data set contains the all positive value attributes then the transformation is not required.

In the next step, for each data point we calculate the

distance from origin. Then, the original data points are sorted accordance with the sorted distances. After sorting partition the sorted data points into k equal sets. In each set take the middle points as the initial centroids. These initial centroids lead to the better unique clustering results. Next, for each data point the distance calculated from all the initial centroids.

The data points are assigned to the clusters having the closest centroids in the next step. ClusterId of a data point denotes the cluster to which it belongs. NearestDist of a data point denotes the present nearest distance from closest centroid.Next, for each cluster the new centroids are calculated by taking the mean of its data points. Then for each data point the distance calculated from the new centroid of its present nearest cluster. If this distance is less than or equal to the previous nearest distance, then the data point stays in the same cluster, otherwise for each data point we need to calculate the distance from all centroids. After calculated the distances, the data points are assigned to the appropriate clusters and the new ClusterId's are given and new NearestDist values are updated. This reassigning process is repeated until the convergence criterion is met.

D. Parallelising Implementation Using Maper and Reducer

Each map reads the K-centroids + one block from dataset .

Assign each point to the closest centroid

Output <centroid, point>

Reduce Side

Gets all points for a given centroid

Re-compute a new centroid for this cluster

Output: <new centroid>

Creates a single output file

5 METHODOLOGY

The following algorithm shows step by step procedure of the proposed system.

- **Step 1:**Using silhoutte method to determine the number of cluster K .
- **Step 2:** In the given data set the data points contains the both positive and negative attribute values then Find the minimum attribute value in the given data set subtract with the minimum attribute value.
- **Step3:** For each data point calculate the distance from origin. Sort the distances obtained in step 2. Sort the data points accordance with the distances
- **Step4:** Partition the sorted data points into k equal sets. In each set, take the middle point as the initial centroid. Compute the distance between each data point to all the initial centroids *and* Repeat
- **Step5:** For each data point find the closest centroid and assign to cluster . Set ClusterId and Set NearestDist For each cluster recalculate the centroids. For each data point Compute its distance from the centroid of the present nearest cluster.
- **Step6:** If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster. Else For every centroid compute the distance End for. Until the convergence criteria is met.

Map Side

- Each map reads the K-centroids + one block from dataset
- Assign each point to the closest centroid
- Output <centroid, point>

Reduce Side

- Gets all points for a given centroid
- Re-compute a new centroid for this cluster
- Output: <new centroid>

Use of Combiners

- Similar to the reducer
- Computes for each centroid the local sums (and counts) of the assigned points
- Sends to the reducer <centroid, <partial sums>>

Use of Single Reducer

- Amount of data to reducers is very small
- Single reducer can tell whether any of the centers has changed or not
- Creates a single output file.

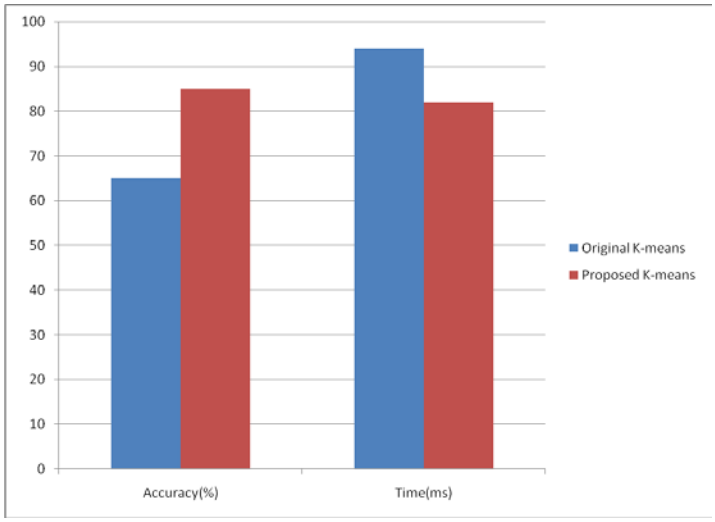

Fig 1.Performance comparison chart for Iris Dataset.

6 Experimental Results

We tested both the algorithms for the data sets with known clustering, Iris , Wine, Wine Quality,3D Road Network,Bank Marketing . The same data sets are used as an input for the original k-means algorithm. the original algorithms need number of clusters as an input. In additional set of initial centroids also required. The enhanced method finds the number of cluster and initial centroids systematically.And it does not take any additional inputs like threshold values. The original k-means algorithm is executed seven times for different sets of values of the initial centroids. In each experiment the accuracy and time was computed and taken the average accuracy and time of all experiments. The results showed with the help of bar charts in the Fig.1, 2, 3, 4, 5 .The results obtained show that the proposed algorithm is producing better unique clustering results compared to the k-means algorithm in less amount of computational time.

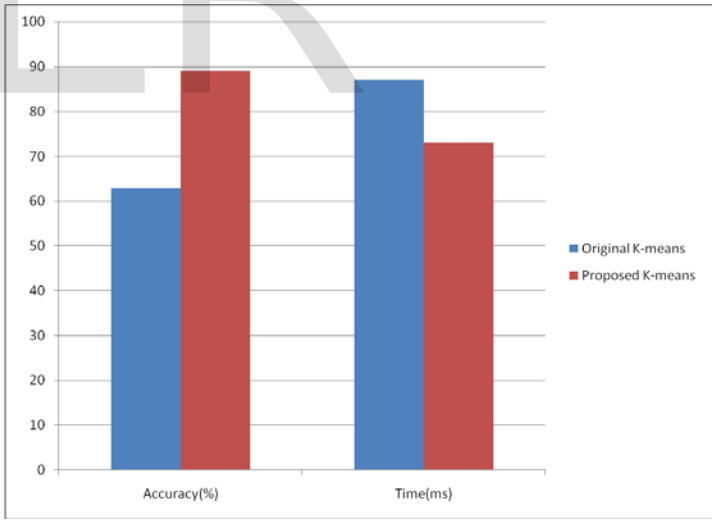

Fig 2. Performance comparison chart for Wine Dataset

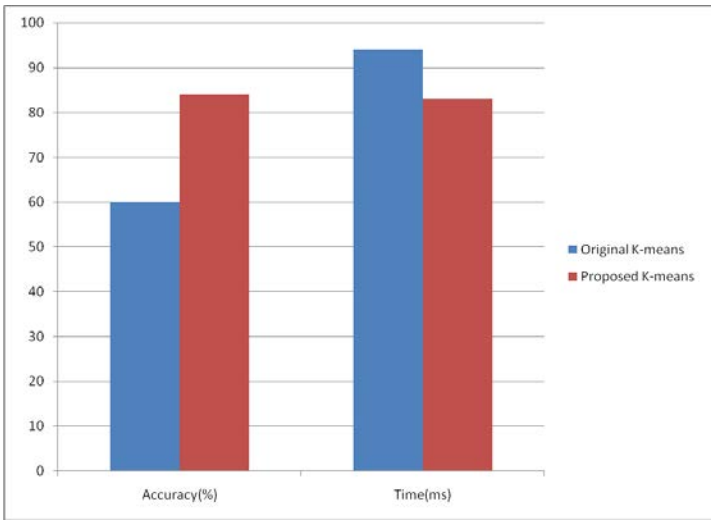

Fig 3. Performance comparison chart for Wine Quality

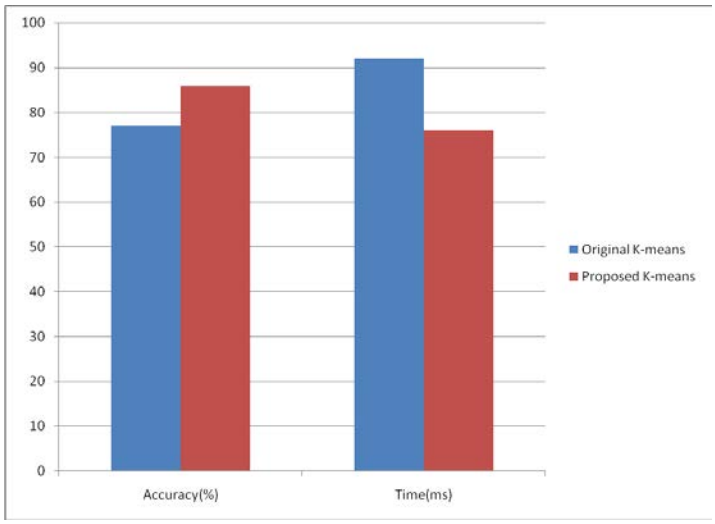

Fig 5. Performance comparison chart for Bank Marketing

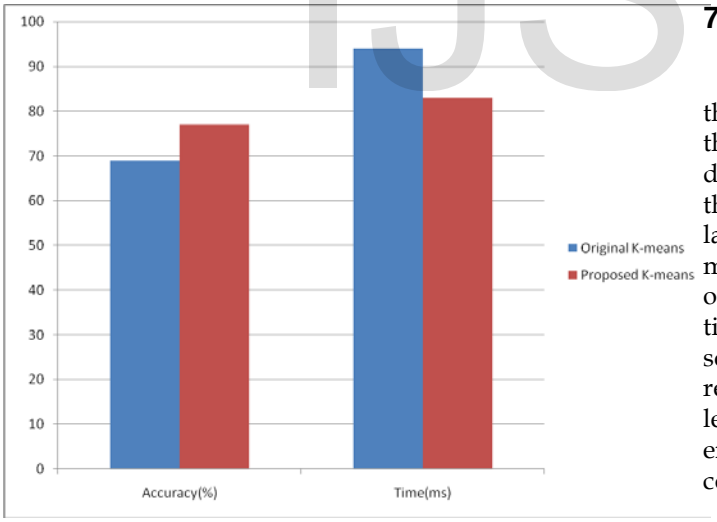

Fig 4. Performance comparison chart for 3D Road Network

7 Conclusion

As In this paper to tested Mahout K-means scale. For that purpose several experiments were conducted by varying the dataset size and the number of nodes. The influence of two different distance measures was also tested. It is concluded that Mahout can be a promising tool for Kmeans clustering for large datasets, which is scalable and gains significant performance. Using Mahout with small files is not always the best option. There is an overhead due to replication and distribution of the data blocks. The overhead is largely compensated as soon as the dataset grows. Increasing the number of nodes reduces the execution times. However, for small files it can lead to an underutilization of each machine's resources As expected the quality of Mahout K-means clustering will be consistent between the several experiments.

References

[1]. Kathleen Ericson∗, Shrideep Pallickara"On the performance of high dimensional data clustering and classification algorithms" Elsevier Publica-

tion,ScienceDirect.com,13 june 2012,pg.no:1024-1034.

[2]. Todd D. Plantenga, Yung Ryn Choe, Ann Yoshimura "Using Performance Measurements to Improve MapReduce Algorithms" Elsevier Publication,ScienceDirect.com,4 may 2012,pg.no:1920-1929.

[3]. Ping ZHOU †, Jingsheng LEI, Wenjun YE"Large-Scale Data Sets Clustering Based on MapReduce and Hadoop" Journal of Computational Information Systems 7: 16 (2011) 5956-5963

[4]. Weizhong Yana,∗, Umang Brahmakshatriya a, Ya Xuea, Mark Gilder b, Bowden Wisec "p-PIC: Parallel power iteration clustering for big data" Elsevier Publication ScienceDirect.com, 73 (2013) 352–35 .

[5]. Zhuo Tang · Junqing Zhou ·Kenli Li · Ruixuan Li "A MapReduce task scheduling algorithm for deadline constraints" Springer Science+Business Media New York 2012.

[6]. . A. Espinosa · P. Hernandez · J.C. Moure ·J. Protasio · A. Ripoll "Analysis and improvement of map-reduce data distribution in read mapping applications" Springer Science+Business Media, LLC 2012, J Supercomput (2012) 62:1305–1317

[7]. Abhishek Verma · Brian Cho · Nicolas Zea ·Indranil Gupta · Roy H. Campbell "Breaking the MapReduce stage barrier" Springer Science+Business Media, LLC 2011, Cluster Comput (2013) 16:191–206.

[8]. [8]. Chunming Rong - "Using Mahout for clustering Wikipedia's latest

[9]. articles" Third IEEE International Conference on Coud Computing Technology and Science, 2011

[10]. [9]. Chien-Liang Liua,∗, Tao-HsingChangb, Hsuan-HsunLic, "Clustering documents with labeled and unlabeled documents using fuzzy semi-Kmeans ",Elsevier Publication,ScienceDirect.com, *221(2013)48–64.*

[11]. [10].Vivek Kumar Singh, Nisha Tiwari, Shekhar Garg, "Document Clustering using K-means, Heuristic K-means and Fuzzy C-means" , 978-0-7695-4587-5/11 $26.00 © 2011 IEEE.