A Hybrid Approach for Efficient Clustering of Big Data

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Saurabh Arora
(Roll No. 801232023)

Under the supervision of:
Dr. Inderveer Chana
Associate Professor
Computer Science and Engineering Department

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*A Hybrid Approach for Efficient Clustering of Big Data*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Inderveer Chana* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

S. Arora

(Saurabh Arora)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Inderveer 15/07/14

(Dr. Inderveer Chana)

Associate Professor,
Computer Science and Engineering Department

Countersigned by

Deepak Garg

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First of all, I am thankful to God for his blessings and showing me the right direction. With His mercy, it has been made possible for me to reach so far.

I would like to express my deep sense of gratitude to my supervisor Dr. Inderveer Chana Associate Professor, Computer Science & Engineering Department for her invaluable guidance, support and encouragement during this thesis work. She provided me all the resources and guidance throughout thesis work.

I express my gratitude to Dr. Deepak Garg, Head, Computer Science & Engineering Department and Mr. Ashutosh Mishra, P.G. Coordinator for providing us adequate environment, facility for carrying thesis work.

I extend my thanks to Mr. Sukhpal and Dr. Damandeep Kaur for sharing their expertise and time to help me accomplish this work.

I would also like to thank all staff members and my colleagues who were always there at the need of hour and provided with all help and facilities, which I required, for the completion of my thesis work.

I would also like to thanks to all my friends for helping me in the hour of need and providing me all the help and support for completion of my thesis.

Last but not the least I am highly grateful to all my family members for their inspiration and ever encouraging moral support, which enables me to purse my studies.

Saurabh Arora

(801232023)

Abstract

In today's era data generated by scientific applications and corporate environment has grown rapidly not only in size but also in variety. This data collected is of huge amount and there is a difficulty in collecting and analyzing such big data. Data mining is the technique in which useful information and hidden relationship among data is extracted, but the traditional data mining approaches cannot be directly used for big data due to their inherent complexity.

Data clustering is an important data mining technology that plays a crucial role in numerous scientific applications. However, it is challenging to cluster big data as the size of datasets has been growing rapidly to extra-large scale in the real world. Meanwhile, MapReduce is a desirable parallel programming platform that is widely applied in data processing fields. Hadoop provides the cloud environment and is the most commonly used tool for analyzing big data. K-Means and DBSCAN are parallelized to analyze big data on cloud environment. The limitation of parallelized K-Means is that it is sensitive to noisy data, sensitive to initial condition and forms fixed shape while DBSCAN has an issue of processing time and it is more complex than K-Means.

This thesis presents a theoretical overview of some of current clustering techniques used for analyzing big data. Comprehensive analysis of these existing techniques has been carried out and appropriate clustering algorithm is provided. A hybrid approach based on parallel K-Means and parallel DBSCAN is proposed to overcome the drawbacks of both these algorithms. This approach allows combining the benefits of both the clustering techniques. Further, the proposed technique is evaluated on the MapReduce framework of Hadoop Platform. The results show that the proposed approach is an improved version of parallel K-Means clustering algorithm. This algorithm also performs better than parallel DBSCAN while handling clusters of circularly distributed data points and slightly overlapped clusters. The proposed hybrid approach is more efficient than DBSCAN-MR as it takes less computation time. Also it generates more accurate clusters than both K-Means MapReduce algorithm and DBSCAN MapReduce algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

Chapter 1
Introduction

This chapter introduces big data and its evolution including its characteristics, analytical challenges; it's analytics in cloud along with various tools, techniques and requirements. This chapter also introduces data mining, clustering application and its significance on big data followed by organization of thesis.

1.1 Evolution of Big Data

The term big data is considered as an umbrella which includes everything from digital data to health data. Big data has evolved from various stages starting from primitive and structured data to complex relational data and now very complex and unstructured data. Figure 1.1 shows evolution of big data over the years. The concept of how data became big started seventy years ago when the growth rate in volume of data was known as information explosion. In 1944 Fermont Rider, a librarian estimated that size of American Universities libraries is getting doubled every sixteen years. He estimated that by this growth rate there would be 200,000,000 volumes by 2040. In decade of 90 IBM introduced the relational database concept in which data can be stored in tables and can be analysed easily by using different analysis techniques. By the end of 2003 there was

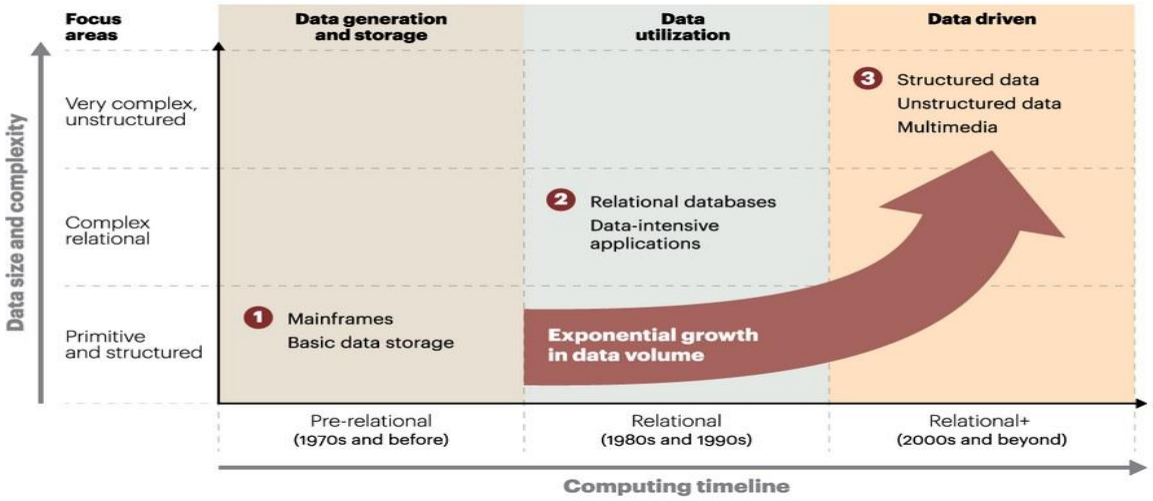

Figure 1.1: Evolution of Big Data [2]

5 exabytes of data that was created, today this amount of information is created in just two days. This data is generated from different sources and includes online transactions, sensor data, social networking, health records, census and science data and live streaming data. This digital data is of 2.72 zettabytes and is predicted to be doubled every two years. Figure 1.2 shows that how significantly data has grown from 2000 onwards [1][2][3].

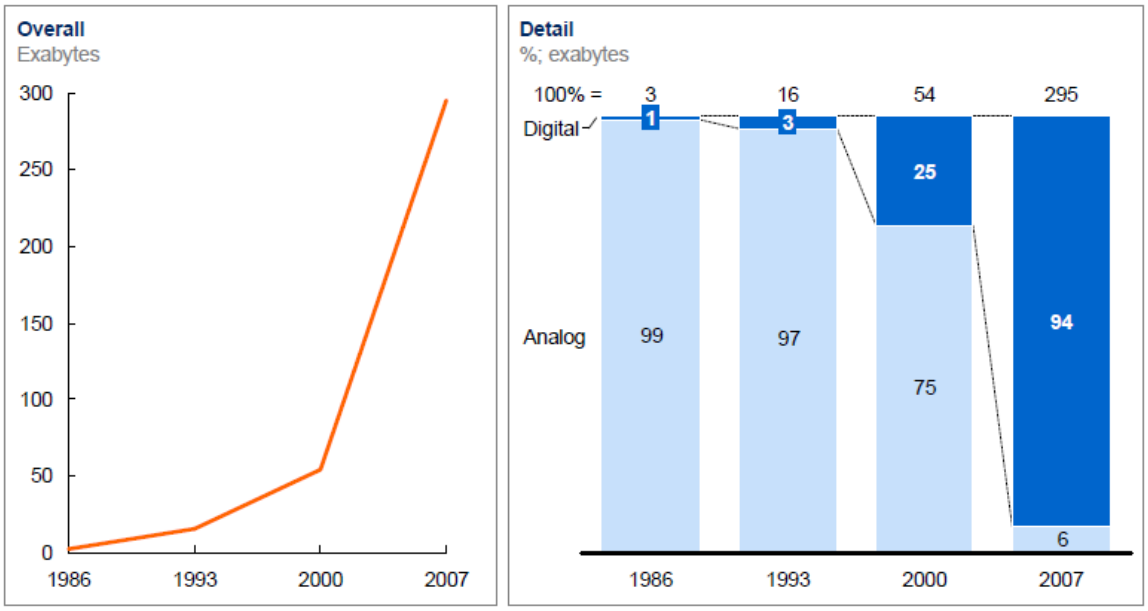

Figure 1.2 Significant Growth of Data [2]

1.2 Big Data and Data Analytics

Nowadays the data collected from different sources is not only growing in its size but is also increasing in variety and variability. The big data is defined as datasets whose size is beyond the ability of typical database tools to store, capture, manage and analyze. The best way to define big data is via three Vs which are data volume, data velocity and data variety or variability [4]. The data volume is regarding size (terabytes and petabytes), records, transactions and tables, files. The data velocity is about how frequently data is generated by an application in real time or in streaming. The data variety includes different types of data that is structured (simply RDBMS), semi-structured (XML, RSS feeds) and unstructured (text, human languages). Big data is remarkably diverse in terms data types, sources and entities represented.

Data analytics has evolved lot over the years and Big Data Analytics is the latest in this evolution of data. Big Data Analytics is the process of analyzing large amount of data having different variety to uncover unknown correlation, hidden patterns and other useful information. Such information results in business benefits, such as increased revenue and effective marketing. The primary goal of big data analytics is to help enterprises make better business decisions and other users to analyze huge volumes of transaction data as well as other data which is left by Business Intelligence (BI) programs. Big data analytics can be done with advanced analytics software tools such as data mining and predictive analytics. But big data collected from unstructured data sources is not fit in the traditional data warehouses which lead to new big data technology. These technology associated with big data analytics includes Hadoop, MapReduce and NoSQL databases. Large datasets across the clustered systems can be processed from these technologies. The pitfalls for organizations on big data analytics includes high cost for hiring professionals and challenges in integrating Hadoop and data warehouses[5].

1.2.1 Characteristics of Big Data

There are certain characteristics of big data which are listed below [6]:

- Volume – The volume is related to the size of data. At present data is in pettabytes and in near future it will be of zettabytes.
- Variety – The data is not coming from single source it includes semi structured data like web pages, log files etc, raw, and structured and unstructured data.
- Variability – The variability considers inconsistencies in data flow.
- Value – The value is importance of data used by the user. The user queries against certain data stored, obtains result, rank them and can store for future work.
- Velocity – The velocity is related to the speed of data coming from different resources. The speed of incoming data is not limited and is also not constant.
- Complexity – The data is coming from various resources in huge amount thus it is difficult to link or correlate multiple data.

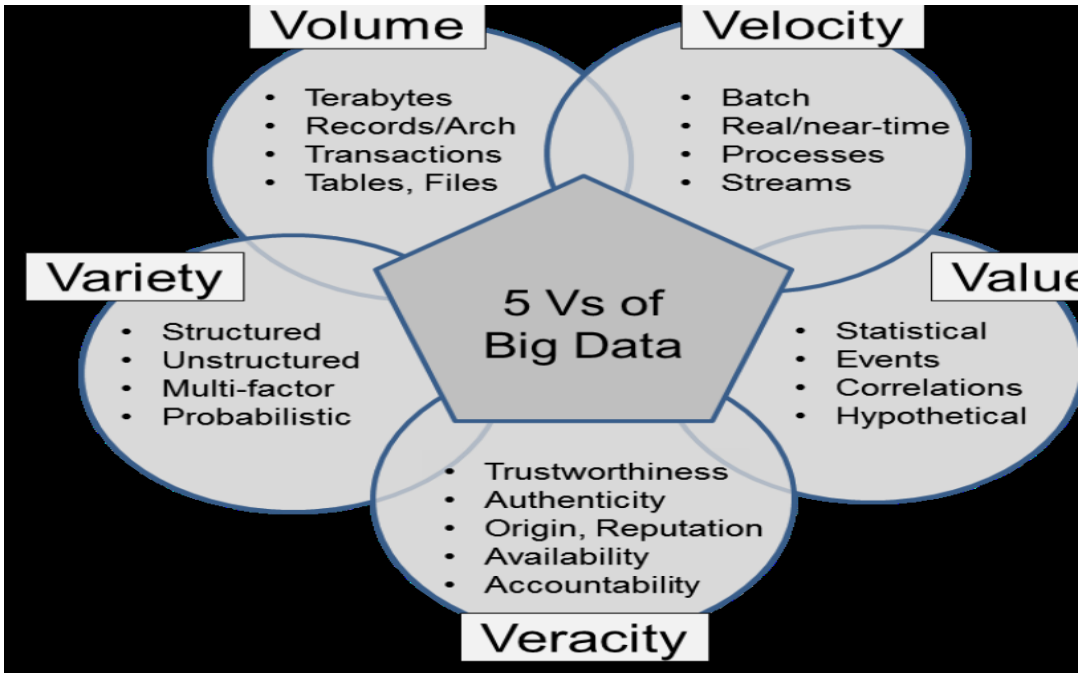

Figure 1.3: Characteristics of Big Data [7]

1.2.2 Big Data Analytics Challenges

The main challenges for big data analytics are listed below [6]:

- Volume of data is large and also varies so challenge is how to deal with it.
- Analysis of all data is required or not.
- All data needs to be stored or not.
- To analyse which data points are important and how to find them.
- How data can be used in the best way.

1.3 Data Analytics in Cloud Computing

In cloud computing deploying big-data analytics services is more than just an existing trend. Now era has begun where data is being generated from many different sources such as click-stream, sensors, log files, social media and mobile devices. Data collected now days can exceed hundreds of terabytes and is generated continuously. This big-data represents data sets which are difficult to analyze with traditional data management methods. To get information from big-data, enterprises have to deploy big-data analytics into a scalable platform. The invention of Cloud Computing is providing enterprises to

analyze such big data leveraging vast amount of computing resources available on demand with low resource usage cost.

1.3.1 Requirement for Big Data Analytics

The three significant requirements [8] for big data analytics to process large datasets are as follows:

- Minimize data movement – This is all about conserving resource. In traditional analysis when data grows it becomes less efficient and data moves around too much. It is more sensible to store and process it in the same place.
- Use existing skills – With new data sources and new data skills are required. If skills are lacking then there arises the problem in training and hiring. Sometimes existing skills can help in where analysis can and should be done. Since SQL is used more than MapReduce, it is important to support both types of processing.
- Attend Data Security – In any corporate applications data security is essential. Users not only carefully defined metrics and dimensions and attributes but also set of reliable policies and security control. In unstructured data and open source analysis tools these processes are often lacking so we need to pay attention on analysis of each project.

1.3.2 Big Data Analytics Techniques

Several techniques are drawn on disciplines such as computer science and statistics that can be used to analyze huge datasets. This section takes a brief look at some of categories of commonly used techniques for analyzing big data. All the techniques listed can be applied to diverse and larger datasets [9].

i) A/B testing – This technique is also known as bucket testing or split testing. In this technique control group is compared with a variety of test groups in order to determine what changes will improve a given objective variable. Large number of tests are analyzed and executed to ensure that group are of sufficient size to determine meaningful differences between treatment and control groups. Multivariate generalization of this technique is often called “A/B/N” testing.

ii) Network Analysis – This analysis algorithm is used to detect relationships between the nodes in a graph or in a network. It is useful for social networks where important information regarding user, his friends, relatives etc can be obtained. Central users in the network can also be discovered by social network analysis.

iii) Association rule learning – Set of techniques which finds interesting relationship among variables in large datasets. These techniques include variety of algorithms to generate and test possible rules. Used in market analysis for their benefits where retailer can determine what products are sold.

iv) Sentiment Analysis – To determine subjectivity of reviewers and opinion is the aim of sentiment analysis which is a Natural Language Processing (NLP). With increase in popularity of various websites where users can provide their opinion for other users and items that are purchased, the internet is full of comments, reviews and ratings. The main aim of sentiment analysis is to classify user opinion. To determine reaction about the product and actions of user, companies use this analysis.

v) Genetic Algorithm – It is an optimization technique that is based on the "survival of the fittest" for optimal solution. In this technique solutions are encoded as chromosomes that can be combined and mutated. The potential of chromosomes is determined by its fitness value. The best ones are copied into next generation. It is an evolutionary algorithm as it is well suited for solving non linear problems. It can be used for the improvement of scheduling of jobs in manufacturing and further optimize the performance.

vi) Machine Learning – A sub domain of the artificial intelligence and is related with the design and development of algorithms which senses the behaviours based on historical data. Machine learning basically focuses on automatic recognition of complex patterns and makes intelligent decision based on data.

vii) Cluster Analysis – This is an unsupervised learning technique which classifies user in smaller subgroup with similar properties which are not known in advance. It is different from classification. In markets it can be use to find customer with similar taste.

viii) Pattern Reorganization – This technique generates some kind of output according to a given input value according to some specific algorithm.

ix) Predictive Modelling – In this set of techniques a mathematical model is created or chosen according to the best probability of an outcome. This technique is helpful in determining customer's likelihood for the customer relation manager.

1.3.3 Big Data Analytics Tools

To analyze big data and generate insight, there are five key approaches to uncover hidden relationships [8].

i) Discovery Tools – These tools can be used for analysis along with traditional business intelligent source system. They are helpful throughout information lifecycle for rapid analysis of information from any data source. The user can draw new ideas and meaningful conclusions and can make intelligent decisions quickly.

ii) Business Intelligence (BI) Tools – These tools are important for analyzing, monitoring and performance management of data from different data sources. They help in taking the decisions for the business by providing comprehensive capabilities, ad-hoc analysis on the large scale platform.

iii) Decision Management – These tools consist of business rules, predictive modeling and self learning to perform action based on current context. They maximize customer interaction by suggestions. It can be integrated with Oracle Advanced Analytics to execute complex predictive models and make real time decision.

iv) In-Database Analytics – They consist of different techniques which help in finding relationship and pattern in data. This technique is helpful as it removes the movement of data to different locations thus reducing total cost of ownership and information cycle time.

v) Hadoop – This tool is very helpful for analyzing the big data. It uses the MapReduce programming model which makes it one of the best tools to analyze data. It helps enterprises to look for potential value in the new data using inexpensive servers.

1.4 Data Mining

Data mining is the process of extracting useful information or to find out hidden relationship among data. This information or knowledge is very helpful for business organisations to grow their business as it is helpful in decision making. Data mining technology has come across several stages [10] as *First* stage it was a single algorithm for single machine for vector data. At *second* stage it was combined with database for multiple algorithms. *Third* stage is where it has provided support for grid computing. At fourth stage data mining algorithm was distributed. Fifth stage is where parallel data mining algorithms are present for big data and cloud services. The parallel data mining model can be roughly divided into four categories as association rule mining, classification algorithms, clustering algorithms and stream data mining algorithms. The major challenge is to design and develop efficient algorithm for mining big data from sparse, uncertain and incomplete data.

1.4.1 Association Rule Learning

In data mining, purpose of association rule learning is to discover or extract interesting links between data items from large databases. The common algorithm is Apriori algorithm. Let us take example of supermarket where data is gathered about the purchasing habits of customer. By using association rule mining, useful information like which products are frequently bought together is obtained and is used for future marketing.

1.4.2 Classification

In data mining, purpose of classification is to give a classification function or model which identifies or maps data item to one of the given categories. Classification is the task of finding groups and structures which are similar to one another in some way without knowing structures in data. The most common algorithms are AQ Algorithm and decision tree algorithms.

1.4.3 Clustering

Clustering Analysis or clustering, one of the major techniques in data mining is the process of putting similar data in one group or cluster and dissimilar data in other group. The clustering is an unsupervised learning technique. In this each cluster contains similar data and is different from other groups. The clustering is useful technique to recognize different patterns which assists in various business applications. The advantage of clustering over classification is that clustering is adaptable to changes and easily distinguishes different clusters, also it requires less collection cost for tuples and pattern as compared to classification [11].

The clustering method is appropriate for discovering the internal relationship among datasets for preliminary assessment of sample structures. With automatic clustering one can easily performs it on one, two or three dimensional data but with higher dimensions it is difficult to construe data from high dimensional space.

Clustering methods are difficult to categorize in a crisp form as there is an overlapping of these categories which may have features of different categories. Clustering algorithms can be broadly classified into hierarchical, partition and density-based clustering. The hierarchical clustering method can be further divided into agglomerative and divisive techniques [12].

1.4.3.1 Partition Clustering

In partition based clustering techniques the dataset is partitioned into various clusters. By partitioning each data element would have the cluster index. In this approach user has to predefine the number of clusters with some criteria or parameters on the basis of which the solution will be evaluated. For example one of the parameter could be distance between the data points on the basis of which each cluster has certain number of data elements. The most popular algorithms of partitioning cluster are K-Means, PAM (Partitioning Around Mediods) and CLARA (Clustering LARge Applications). The well known distance method for this category is Euclidean distance method for K-Means.

1.4.3.2 Hierarchical Clustering

In hierarchical clustering, the hierarchy of clusters is build that is a tree of clusters also called dendrogram which represent the result of cluster analysis. In this the prior number of clusters is not to be mentioned. Hierarchical clustering can be further classified into agglomerative and divisive approach. The agglomerative is bottom-up as it starts with one cluster and merges two or more clusters. The divisive is top-down approach which each cluster is split into different number of clusters. Figure 1.4 shows the hierarchical clustering approach. The most popular algorithm of this category is CLUE.

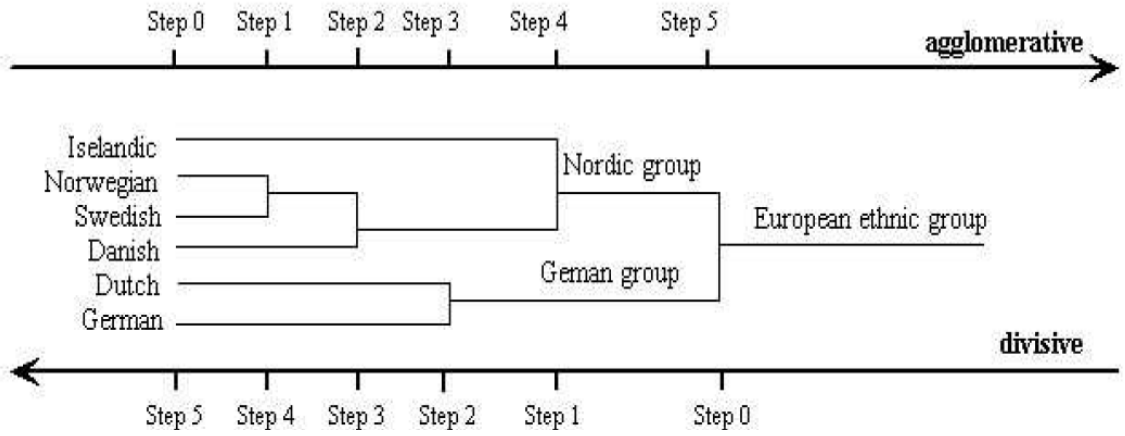

Figure 1.4 Hierarchical Clustering Approaches [12]

1.4.3.3 Density-based Clustering

The purpose of density-based clustering is to determine arbitrary-shaped clusters. In this the low density regions separate the high density region of data objects of the cluster. This algorithm is also used for determining connected graph as it checks the distance between core point with other data points and checks whether distance is not less than the defined radius. The density of each data point is calculated by detecting the number of data points in its neighborhood. The cluster is only said to be dense if it would has more than minimum points; the minimum points are the number of data points which should be present in the cluster. The most common algorithm of this category is DBSCAN which deals with noisy data as well.

1.4.4 Stream Data Mining

The stream data mining is the concept in which the dynamic data is to be analyzed and is used for business and marketing purposes. The data collected from the internet, clicking data etc is being generated day by day thus to analyze this data is one of the difficult task as the rate of flowing of data is different. There are very few techniques to analyze online streaming data.

1.5 Organization of Thesis

The rest of the thesis is organized as follows:

Chapter 2- This chapter contains exhaustive description of literature survey done to study the concept of clustering of big data, recent clustering techniques for analyzing big data and tabular comparison of these techniques.

Chapter 3- This chapter presents the problem statement along with the objectives of this research work.

Chapter 4- This chapter describes hybrid algorithm to solve the stated problem.

Chapter 5- This chapter focuses on related concepts that are cardinal in our proposed method followed by implementation details and experimental results.

Chapter 6- In this chapter conclusion, followed by possible future research work is discussed.

Chapter 2

Literature Review

This chapter presents literature survey of clustering techniques for analyzing big data and tabular comparison of these techniques is presented.

2.1 Clustering Methods for Big Data

Clustering methods are applied for both small and large datasets but the traditional clustering methods like K-means, DBSCAN and so on, cannot be applied directly on cloud environment for analyzing big data as big data is collected from different resources and is in various forms [13]. Thus the parallel version and extension of traditional algorithms are required to be designed for analyzing big data in cloud environment. Figure 2.1 shows the evolution of different clustering algorithms that are used now days for analyzing big data.

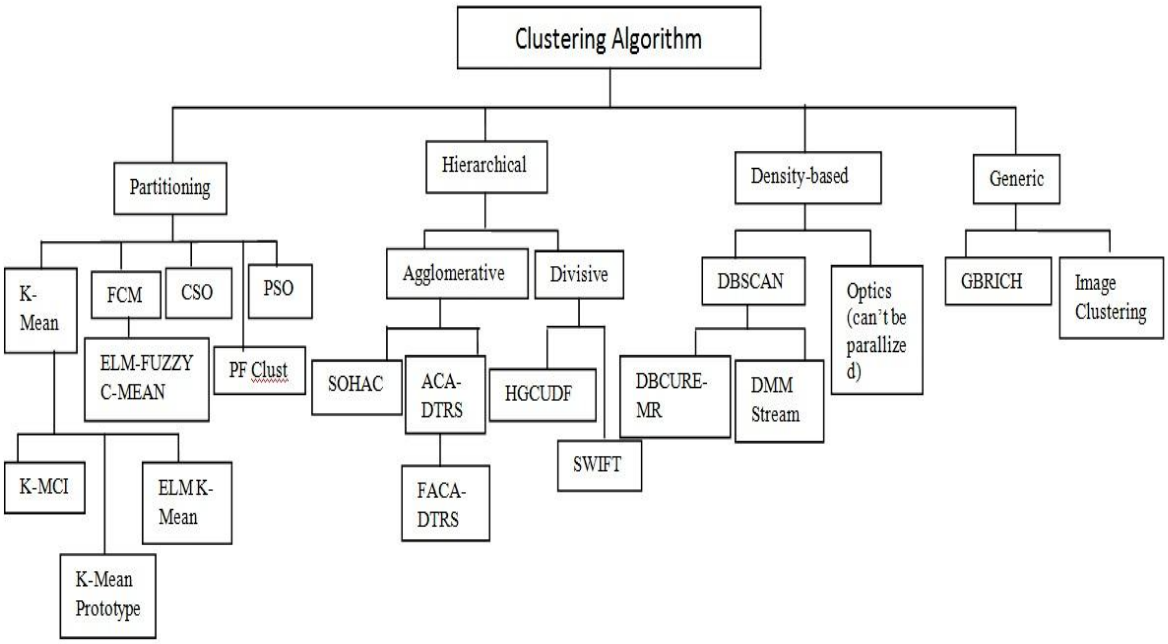

Figure 2.1: Evolution of Clustering Techniques

2.2 Recent Clustering Techniques for Big Data

Several leading research groups both in academia and industry are working in the field of analyzing big data. In this section recently developed clustering techniques for analyzing big data is discussed. These techniques are put under different categories of clustering techniques for easy reference.

2.2.1 Partitioning Clustering Techniques

T Qing He et al. [14] proposed Extreme Learning Machine (ELM) K-means and ELM NMF (Non negative Matrix Factorization) for single hidden feed forward neural network. They used ELM techniques for clustering problem as it produces better result than Mercer- Kernel based method and traditional algorithms. They used three different datasets; two from university machine repository and third was from document corpus. ELM K-means and ELM NMF both are stable for large datasets, efficient and simple to implement.

In [15] three algorithms are discussed for clustering. The author discussed incremental k-means which clusters numerical data. Modified k-modes algorithm clusters categorical data while for mixed numeric and categorical data k- prototype is defined. All three algorithms are efficient and reduce cost of clustering function. Complexity is also reduced as iteration converges and dissimilarity rate is also low.

Ganesh Krishnasamy et al. [16] presented a Modified Cohort Intelligence (MCI) by implementing mutation operator into Cohort Intelligence (CI) to overcome issue of quality and convergence speed. They presented hybrid K-MCI algorithm for data clustering which is a combination of K-Means and MCI and has an advantage of both algorithms. The presented algorithm is reliable and efficient. It also provides good quality of solutions and has better convergence speed as compared to other clustering techniques. The performance test was carried on six standard data sets from UCI Machine Learning Repository.

Ishak Saida, Nadjet and Omar [17] proposed a new metahuristic approach for data clustering which is based on Cuckoo Search Optimization (CSO) to avoid inconvenience of k-means. The basic properties of cuckoo search are that it is easy to implement and has good computational efficiency performance. The proposed algorithm improves the power

of new method to determine the best values. The experiment was carried on four data sets from UCI Machine Learning Repository.

Xue-Feng Jiang [18] has given a global optimization algorithm for large scale computational problems. The algorithm is a kind of particle swarm optimization based on parallel annealing clustering algorithm. It is a new algorithm based on method of group and is very effective for continuous variable problem; it also has ability to get parallelised. The parallel particle swarm optimization algorithm has less calculation time and provides improved quality clusters. The stronger effectiveness of algorithm is evaluated on large datasets.

Khadija Musayeva et al. [19] presented a clustering algorithm PFClust for finding optimal number of clusters automatically without any prior knowledge of number of cluster. PFClust relies on similarity matrix, immune to problems generated by high-dimensional data. The execution time of PFClust is better and resulting cluster are of good quality. The performance of PFClust is evaluated on high-dimensional dataset.

2.2.2 Hierarchical Clustering Techniques

Hong Yu, Zhanguo Liu and Guoyin Wang [20] presented an efficient automatic method to deal with problem of determining number of cluster. The method is an extension of decision-theoretic rough set model and is designed on the basis of risk calculation by loss functions and possibilities. The authors have presented a hierarchical clustering algorithm called ACA-DTRS (Automatic Clustering Algorithm based on DTRS) which automatically stops after finding the perfect number of cluster. They also proposed FACA-DTRS which is a faster version of ACA-DTRS in terms of complexity. Both algorithms are efficient in terms of time cost. The performance was evaluated on synthetic and real world datasets. Buza, Nagy and Nanopoulos [21] proposed an approach to reduce the storage for tick data which is increasing rapidly. The original tick data is decomposed into smaller data matrix by clustering attributes using a new clustering algorithm Storage-Optimizing Hierarchical Agglomerative Clustering (SOHAC). By using this technique the queries can be executed efficiently. They also present QuickSOHAC for speeding up runtime. This algorithm is based on lower bounding

technique so that it can be applied on high dimensional tick data. The performance of these algorithms is evaluated on three real-world data sets provided by investment bank.

Wang Shuliang et al. [22] presented a new clustering algorithm named Hierarchical Grid Clustering Using Data Field (HGCUDF). In this approach hierarchical grids divide and conquer large data sets in their hierarchical subsets, scope of search is limited to clustering centres and area of data space for generating data field is minimized. HGCUDF is stable and executed most rapidly, improving clustering performance on vast computerised datasets.

Naim et al. [23] present a model based clustering method, called SWIFT (Scalable Weighted Iterative Flow-clustering Technique) for high-dimensional large-sized datasets. It works in three phases comprising of iterative weighted sampling, multimodality splitting and unimodality-preserving merging to scale model based clustering of large high-dimensional dataset. This algorithm is basically for flow cytometry and finds rare cell population. It resolves small datasets and scale large datasets more effectively as compared to traditional approaches when tested over synthetic datasets. It is beneficial for task typical in immune response and scales very large FC datasets. It also has the capability to determine extremely rare population.

2.2.3 Density-based Clustering Techniques

Amini et al. [24] presented the clustering algorithm named DMM-Stream for real time stream data. It is density based clustering algorithm for evolving data streams. The concept of mini-micro cluster which is similar to micro cluster with smaller radius is introduced. The mahalanobis distance method is used to determine correct cluster as it increases quality of cluster while maintaining time complexity. They also give strategy to filter noise from real data. The experiments are performed on real and synthetic datasets.

Kim et al. [25] proposed new parallelized clustering algorithm called DBCURE to cluster big data, based on density based clustering. The proposed algorithm is robust to find clusters with varying densities and can be parallelized for MapReduce. The authors also proposed DBCURE-MR which is suitable for map reduce framework and it can find several clusters in parallel. Both algorithms proposed are efficient and find clusters accurately of various data sets. These are not sensitive to clusters containing different

densities and also perform well with MapReduce framework. The experiment is performed on synthetic datasets such as Butterfly, Window and Clover and also on the real-life data sets.

2.2.4 Generic Clustering Techniques

Chun-Wei Tsai et al. [26] presented an algorithm to solve the issue of high performance search method in big data mining. They proposed a novel spiral optimization algorithm which splits population into subpopulation to increase diversity of search for clustering. The algorithm is distributed and its effectiveness is enhanced by using k-means and oscillation methods. Novel Spiral optimization is quite promising and is based on natural phenomena such as swirl and low pressure. The results are similar to genetic k-means and spiral optimization.

Suri, Murthy and Athithan [27] have given a ranking based algorithm for detecting outliers in categorical data. This is two phase algorithm; first clustering of given data is exposed, second ranking is given to identify the similar set of outliers. This algorithm implements two different ranking methods based on value frequencies and inherent clustering which enhances its performances to detect different types of outliers. To prove the effectiveness of the algorithm numbers of experiments on different public domain categorical data sets are performed. Another benefit of algorithm is its computational complexity is not affected by number of outliers to be detected.

Jiang et al. [28] has discussed about Binary Matrix Factorization (BMF) that reduces dimensions of high-dimensional data sets with binary attributes. They present two variants of Constrained BMF (CBMF) and its relation to other dimensional reduction model. To solve problem of Binary Linear Programming (BLP) an alternating update procedure is provided. An effective 2-approximation algorithm is presented to explore relationship between clustering and BLP sub-problem. The randomized algorithm is also proposed which finds accurate solution in better time and solution obtained is accurate. The experiments are conducted with high-dimensional datasets.

Liang Chi Heish et al. [29] has given a method for improving response time of image search result using MapReduce-based image clustering. The image clustering methods are used to deal with scalability and efficiency problems. The image clusters are

precomputed at offline stage so that it can support online image search result grouping. By using MapReduce- based graph construction method and by proposed algorithm image clusters are generated for large scale datasets. The performance of this method is evaluated on image dataset of flickr which shows its response time to be faster than baseline methods.

J.Dong, F. Wang and Bo Yuan [30] presented an algorithm for clustering large scale streaming data. They discussed that how GPUs and Dynamic parallelism feature of CUDA (Compute Unified Device Architecture) platform is beneficial for Balanced Iterative Reducing and Clustering using Hierarchies (BRICH), one of the data streaming clustering technique. By using GPUs the speed can be faster by 7 to 15 times over original BRICH. The GBRICH (GpuBRICH) also reduces clustering time and can deal with scalability and dimensionality of dataset. Six datasets were used to evaluate the performance of GBRICH.

2.3 Comparative Analysis of Existing Clustering Techniques

There are several clustering techniques which have been recently developed for analyzing big data. These clustering algorithms are examined on parameters of speedup, scalability, quality and accuracy of clusters. A tabular comparison of these techniques with their pros and cons are presented below.

Table 2.1: Comparative Analysis of Clustering Techniques for Big Data

Sr. No.	Technique	Issue Addressed	Type of Clustering	Type of Dataset Used	Execution Time	Cluster Quality	Merits	Demerits
1.	ELM K-means and ELM NMF [14]	Solves the clustering problem by using ELM feature on K-means and Fuzzy C-means.	Partitioning	Datasets from UCI Machine Learning Repository and Document Corpus.	Very Good	Good	ELM features are easy to implement and ELM K-means produce better results than Mercer kernel based methods.	Number of nodes should be greater than 300 else performance is not optimal.

2.	K-modes and K- Prototype algorithm [15]	K-means clustering algorithm is modified as k-modes and k-prototype.	Partitioning	Mixed Numeric and Categorical data.	Good	Good	K-prototype and k-mode produce better result as compared to k-mean. Also reduces function cost.	Quality of cluster is not very good.
3.	NSO [26]	Novel spiral optimization solves issue of high performance search in data mining.	Generic	Real World Dataset	Good	Good	It is quite promising technique and is based on natural phenomena which is easy to implement.	Results are similar to genetic k-means hence need improvement.
4.	Ranking Based Algorithm [27]	The ranking based algorithm is given to deal with the issue of outliers. It works on two methods value frequency and inherent clustering.	Generic	Public domain categorical dataset	Good	Very Good	The algorithm is effective to find different numbers of outliers and computational complexity is not affected by outliers.	The limitation of this algorithm is that it can be applied to categorical datasets only.
5.	A clustering approach to constrained Binary Matrix Factorization [28]	This approach deals with reduction in dimensionality of high dimension data.	Generic	Large High Dimension Datasets.	Very Good	Very Good	The solution obtained is accurate and provided in better time.	NA
6.	DMM-Stream [24]	It resolves the issue of real-time data streaming. It uses concept of mini-micro clusters.	Density Based	Real and Synthetic Datasets	Very Good	Very Good	Determine correct number of cluster with increase quality while maintaining complexity time. Filter noise from data.	With micro-mini clustering it is little complex to implement this technique on all real world datasets.
7.	Clustering based on Cuckoo	It is a metaheuristic approach which	Partitioning	Four UCI Machine Learning	Very Good	Good	It is easy to implement and has good	NA

	Search Optimization [17]	avoids problem of k-means.		Repository Datasets			computational efficiency. It also improves method to detect best values.	
8.	DBCURE-MR [25]	It deals with issue of clustering big data problems. It finds clusters with varying densities and is parallelized with MapReduce.	Density Based	Synthetic data and Real life data.	Very Good	Very Good	It is easy to parallelize. Cluster with varying densities are found accurately. It not sensitive to cluster with varying densities.	It takes much computation time.
9.	ACA-DTRS and FACA-DTRS [20]	Extension of DTRS to find number of clusters automatically.	Hierarchical	Synthetic and Real World datasets	Very Good	Very Good	It detects accurate number of cluster without human interference without losing function quality. Also speedup execution time.	Its limitation is that it cannot work for boundary region.
10.	SOHAC [21]	It deals with the size of tick data which is growing in size rapidly.	Hierarchical	Three real world datasets by investment bank.	Very Good	Good	Queries can efficiently run. Clusters can be found in significant running time.	This algorithm is proposed for tick data only.
11.	HGCUDF [22]	Reduces scope of search and minimized data space by divide and conquer for hierarchical grids.	Hierarchical	Vast Computerised Datasets.	Very Good	Good	It can be applied on parallel platform and speed of spatial data mining is increased.	NA

12.	SWIFT [23]	Model based clustering method to deal with large high dimensional data sets via modern flow cytometry.	Hierarchical	Large FC Datasets and Synthetic Datasets	Good	Very Good	It is task typical and has capability to detect rare population in large datasets.	Limited to only a particular task for clustering.
13.	K- MCI [16]	A hybrid approach to overcome local optima problem. K-means is modified with cohort intelligence.	Partitioning	Six standard datasets from UCI Machine Learning Repository	Good	Good	Convergence speed is better than heuristic algorithms and it is efficient and reliable.	Number of clusters should be known prior.
14.	Parallel Annealing Particle Clustering Algorithm [18]	It resolves the issue of large-scale computation problem by paralleling particle swarm optimization.	Partitioning	Large Test Datasets	Very Good	Good	Computation time is reduced and clustering quality is also improved.	Does not provide the best global optimization solution.
15.	Online image search result grouping with MapReduce-based clustering and graph construction for large scale photos[29]	It deals with scalability problem of large size photos by precomputing image graphs and clusters.	Generic	Two Large Datasets of Flickr Images	Very Good.	Good	It works on MapReduce which increases upto 69 times faster than single machine while image groups are searched with 2-100 times speedup.	Limited to image clustering only cannot used for generalized datasets.
16.	Accelerating BIRCH for Clustering Large Scale Streaming Data Using CUDA Dynamic Parallelism	Latest CUDA benefits of GPUs and Dynamic Parallelism are applied on BRICH for clustering large scale streaming	Generic	Six datasets	Very Good	Good	The speed of execution time is reduced than original BRICH. Dimensionality and scalability is	It is difficult to parallelize GBRICH.

	[30]	data.					also maintained.	
17.	PFClust [19]	To find optimal clusters automatically without using prior knowledge of cluster.	Partitioning	Synthetic Datasets	Very Good	Good	It does not require any prior knowledge to find optimal clusters. It can be parallelized and executes largest dataset in minutes.	NA

The above table shows the comparison of different clustering techniques. These techniques used different large scale datasets from various resources to form clusters. Some of these techniques took less execution time to form clusters but accuracy of clusters is low. The techniques which provide high quality clusters are complex in nature, difficult to implement and took more computation time. Also some techniques are applied only to limited datasets thus not giving global solutions. The efficient techniques have to negotiate with cluster quality while techniques which give accurate clusters are complex and took more execution time.

2.4 Conclusion

In this chapter literature review and comparative analysis of the recent techniques for clustering big data is done. The various clustering techniques for analyzing big data are compared and their merits and demerits are presented in the tabular form. The next chapter provides the gap between these algorithms and the problem statement along with the objectives of this thesis work.

Chapter 3

Problem Statement

In the previous chapter various techniques of clustering for analyzing big data were discussed. This chapter focuses on problem statement taken up in the thesis.

3.1 Gap Analysis

In today's era the corporate and scientific environment produces massive amounts of data. To collect and analyze this data is a difficult task as data is increasing not in amount only but in complexity. Based on literature survey, there are various techniques which are used to analyze large datasets but these techniques are not efficient as some of them are related to particular task and do not provide the global solutions, some of them are fast but they had to compromise with the quality of clusters and vice versa. There has been lot of work done to improve efficiency of K-Means and DBSCAN algorithms to determine good quality clusters in less computation time but there are some shortcomings in both these techniques. Also there is a need to design new methodologies which can deal with the real time and online streaming data.

- The parallel k-means algorithm executes fast but it cannot handle non arbitrary shape and also does not deal with the noisy data [31].
- The parallel DBSCAN algorithm can handle noise as well as non-arbitrary shape but it takes more computation time and is more complex than K-Means [32].
- The ELM feature is applied to K-Means to find accurate clusters in less execution time, the clusters produced had to compromise in their quality also as large amount of resource is required to get an optimal solution [14].
- In another technique k-means is modified to K-mode and K-prototype which generates better result than k-means but are not able to deal with outliers [15].
- The K-Means is also applied with modified coherent intelligence which provides efficiency and reliability but in this boundary points are the problem [16].

- Another approach of parallel annealing particle swarm optimization is presented which depicts the best value in less computation time but it does not give the best global optimized solution [18].
- The density-based algorithm DMM Stream gives the best quality clusters and also filters noise for real time data but it is very difficult to understand due to its complex nature [24].
- The DBCURE-MR is extension of DBSCAN algorithm which produces good quality clusters but it takes more computation time [25].

3.2 Problem Statement

Today big data has become buzz in the market. Among various challenges in analysing big data the major issue is to design and develop the new techniques for clustering. Clustering techniques are used for analyzing big data in which cluster of similar objects are formed that is helpful for business world, weather forecasting etc. Cloud computing can be used for big data analysis but there is problem to analyze data on cloud environment as many traditional algorithms cannot be applied directly on cloud environment and also there is an issue of applying scalability on traditional algorithms, delay in result produced and accuracy of result produced. These issues can be addressed by K-Means and DBSCAN applied together. Therefore in this research work a parallel clustering algorithm is proposed and designed that helps in analysing big data in an efficient manner.

3.3 Objectives

The objectives of the thesis are as follows:

i) To study the existing clustering techniques for analyzing big data.
ii) To propose and design an efficient clustering technique for big data analysis.
iii) To implement and validate the proposed technique on the MapReduce framework.

3.4 Conclusion

The gap analysis along with the problem statement has been presented in this chapter. Also the objectives of the work to be done are described. The next chapter gives the solution design for the problem stated above.

Chapter 4

Proposed Hybrid Clustering Technique

The problem stated in above chapter is solved by a hybrid approach that is based on parallel k-means and parallel DBSCAN. It takes advantages of both algorithms and analyses big data in an efficient way. The proposed technique is considered whose purpose is to generate accurate clusters in less processing time.

4.1 Proposed Hybrid Technique

The proposed method is a hybrid technique based on parallel K-Means and parallel DBSCAN that combines the benefits of both parallel K-Means and parallel DBSCAN algorithms. The benefit of parallel K-Means is that it is not complex and forms clusters in less execution time while the advantage of parallel DBSCAN is that it forms the cluster of arbitrary shape.

Figure 4.1 shows that the main procedure of proposed method that works in three stages;

Stage 1: The first stage is the data partition stage in which data is partitioned into different regions or parts to minimize the boundary points.

Stage 2: In the second stage mapper function is performed on each node in which local clusters are formed.

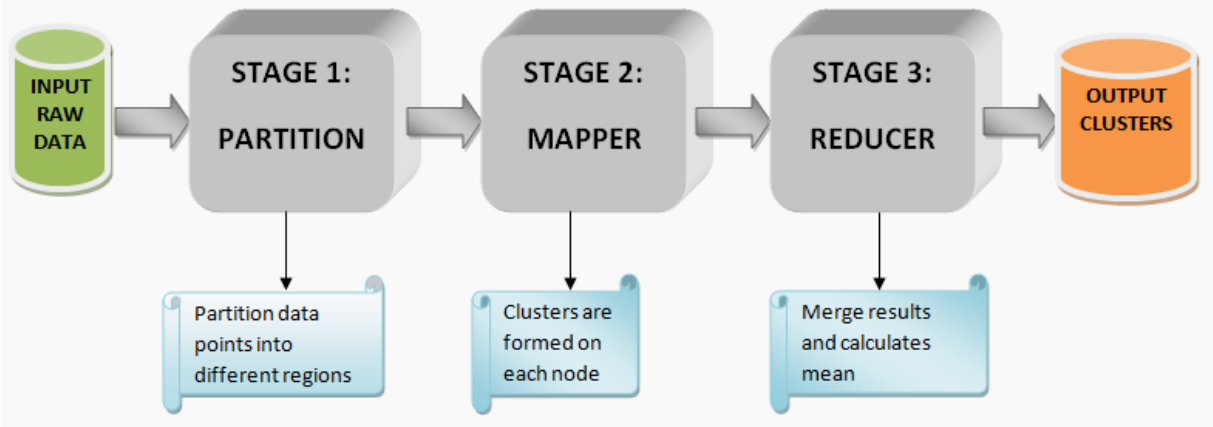

Figure 4.1 Execution of Proposed Algorithm

Stage 3: The third stage is the reducer stage in which mean of each cluster is calculated and it is returned as the cluster id of each cluster.

4.1.1 Basic K-Means Algorithm using MapReduce

Input: k: number of clusters.

D: dataset having n data points.

Output: Set of k clusters

Step 1: Iteratively improves partitioning of data into k clusters.

Step 2: Mapper Phase

i) Input in map function is in form of <key, value> pair as cluster center and data points.

ii) The map function finds the nearest center among k centers for the input point.

Step 3: Reduce Phase

i) The input of reduce function is the output of map function that is all data point and one k center which is nearest to them.

ii) The reducer phase calculates the new center using data points.

iii) The reducer will generate the final output that is cluster mean as cluster id with all data points in that cluster.

Step 4: Repeat until the centers do not change.

It uses the sum of squared distance to calculate the distance between the data points.

Formula used to calculate distance.

$$d^2(p,q) = \sum_{n=1}^{n} (p_n - q_n)^2 \quad \text{----------1}$$

where

d is distance between two data points

p and q are coordinates of data point

Figure 4.2 and figure 4.3 shows the working of K-Means on MapReduce framework.

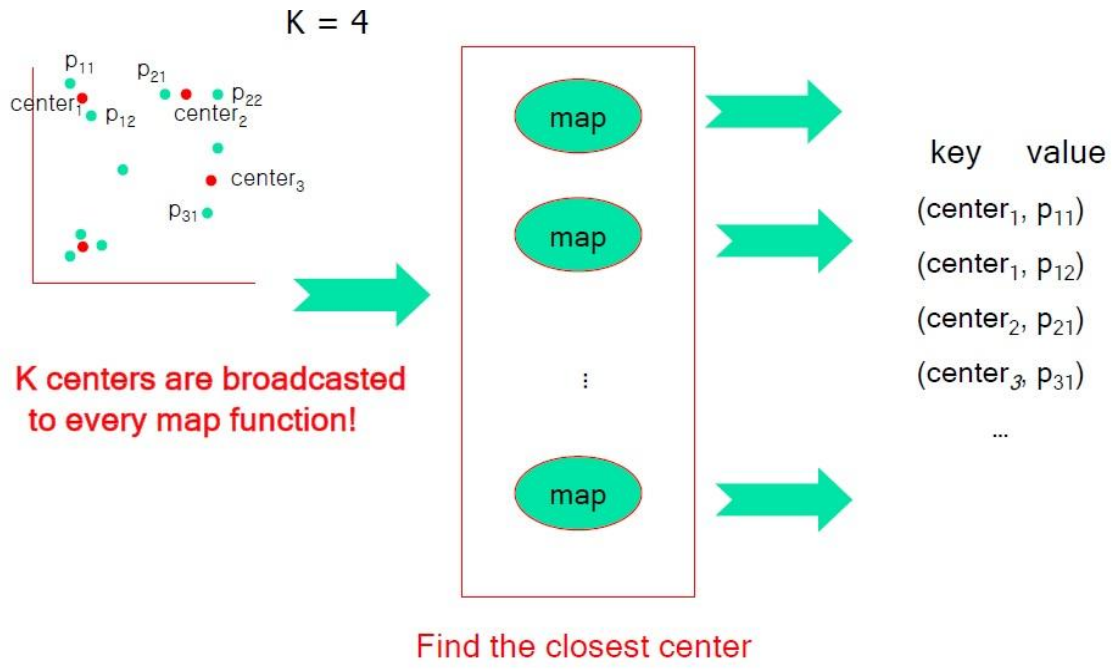

Figure 4.2 Mapper Phase of K-Means [31]

Advantage: The parallel K-Means algorithm has the following benefits

i) The strength or advantage of parallel K-Means algorithm is that it very efficient and takes less time to build the clusters.

ii) It is very easy to implement.

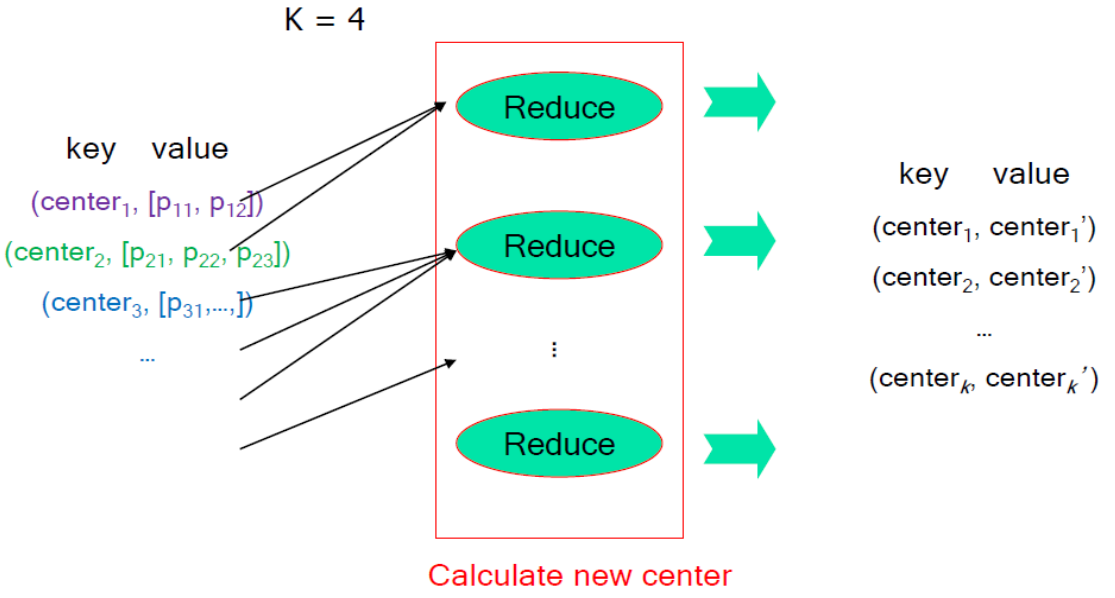

Figure 4.3 Reducer Phase of K-Means [31]

Limitation: The drawbacks or limitations of this algorithm are listed below:

i) The clusters formed by this algorithm is fixed in shape that is only convex shape clusters are formed.
ii) K-Means clustering is sensitive to noisy data or outliers.
iii) In this the initial centers are chosen randomly thus different clusters are formed during different runs for same input dataset.

4.1.2 Basic DBSCAN Algorithm using MapReduce

The DBSCAN-MR executes in the five stages [33]:

Input: e radius of neighborhood

N minimum number of points to required in the cluster

Output: Gives the m number of clusters

Step 1: The first step is preprocessing in which dataset is partitioned and is distributed evenly among different cells.

Step 2: Mapper Phase

i) The input of a map function is a partition of data points.
ii) The output of the map function is local cluster with grid id as key and local cluster id as value.

Step 3: Reducer Phase

i) The input of reduce function is the output of map function.
ii) The output produce by the reduce function is grid id and local cluster id as key and global cluster id as value which is further input for merge result phase.

Step 4: Merge Result

i) This takes input of the reducer phase.
ii) In this phase global clusters are formed on the basis of boundary points.

Step 5: Relabel Phase

i) In the relabel phase the input of merge clusters is provided and global id given to data points.

Figure 4.4 shows the working of MapReduce DBSCAN.

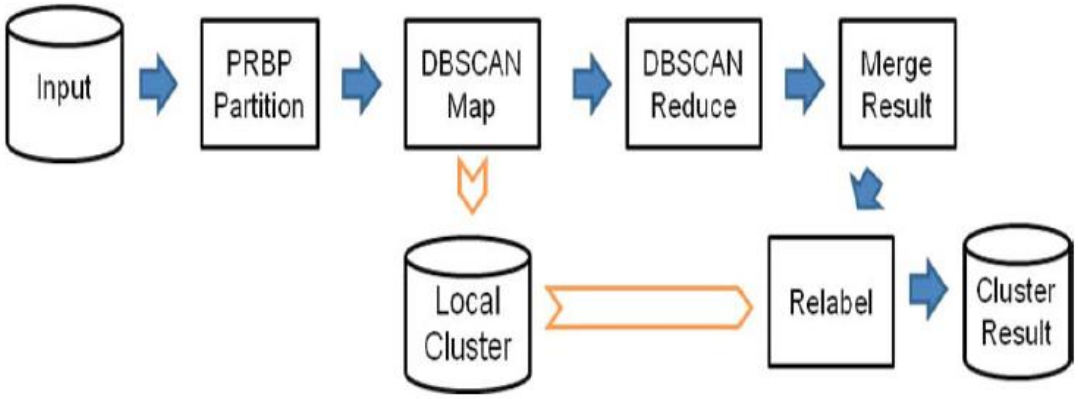

Figure 4.4 Working of MR-DBSCAN [33]

Advantage: The strength of MapReduce DBSCAN algorithm is given below:

i) The DBSCAN algorithm generates the clusters of arbitrary shapes that depend on epsilon radius and its minimum number of points in neighborhood.
ii) It has the ability to handle boundary points or noise efficiently.
iii) It does not require any prior number of clusters.

Limitation: The limitations of this algorithm are as follows:

i) It takes more processing time as it has to deal with noisy data.
ii) It is more complex than K-Means clustering algorithm.

4.1.3 Proposed Algorithm

The design of proposed algorithms is as follows:

Input: D: Dataset having p data points.
K: Number of clusters to be formed.
eps: Radius to calculate neighborhood.
minPts: Number of minimum points required to be in neighborhood.

Output: K clusters having Cm center mean as cluster id.

Step 1: Partitioning of Data

Input: Data set having p data points.

Output: Partitioned Data with its id.

i) The data points are sorted in ascending order.

ii) The range of partition is defined by 2 eps.

iii) Each partition gets its id as Pi.

Step 2: Mapper Stage

Input: The input is in form of <key, value> pair where key is partition id while value is the data points.

Output: Set of Clusters of data points of each partition.

i) Set counter as 0.

ii) For each data point in partition region

iii) Calculate the neighbor set by finding distance between the data points and eps.

iv) If size of neighbor set < minPts

v) Mark data point as noise

vi) Otherwise increment counter add data point to cluster

vii) Emit combination of partition id and cluster id PiCi as key and data points as value

Step 3: Reducer Stage

Input: Takes input from mapper stage as <key, value> pair where key is PiCi and value as data points.

Output: Set of clusters with clusters global id as Cm center mean and data points of clusters as value.

i) Merge the result of mapper function and form clusters

ii) For each cluster

iii) Calculate total of data points in each cluster

iv) Calculate Cm center mean of each cluster

v) Count the total number of clusters

vi) Emit the clusters formed as result with Cm as new cluster id and data points in the clusters as value.

Step 4: If count of cluster formed is not equal to initial K clusters.

i) If count of cluster is greater than K initial cluster than merge by using mapreduce K-Means.

ii) If count of cluster is less than K initial cluster than split by using mapreduce K-Means.

Flowchart of proposed algorithm is described in Figure 4.5.

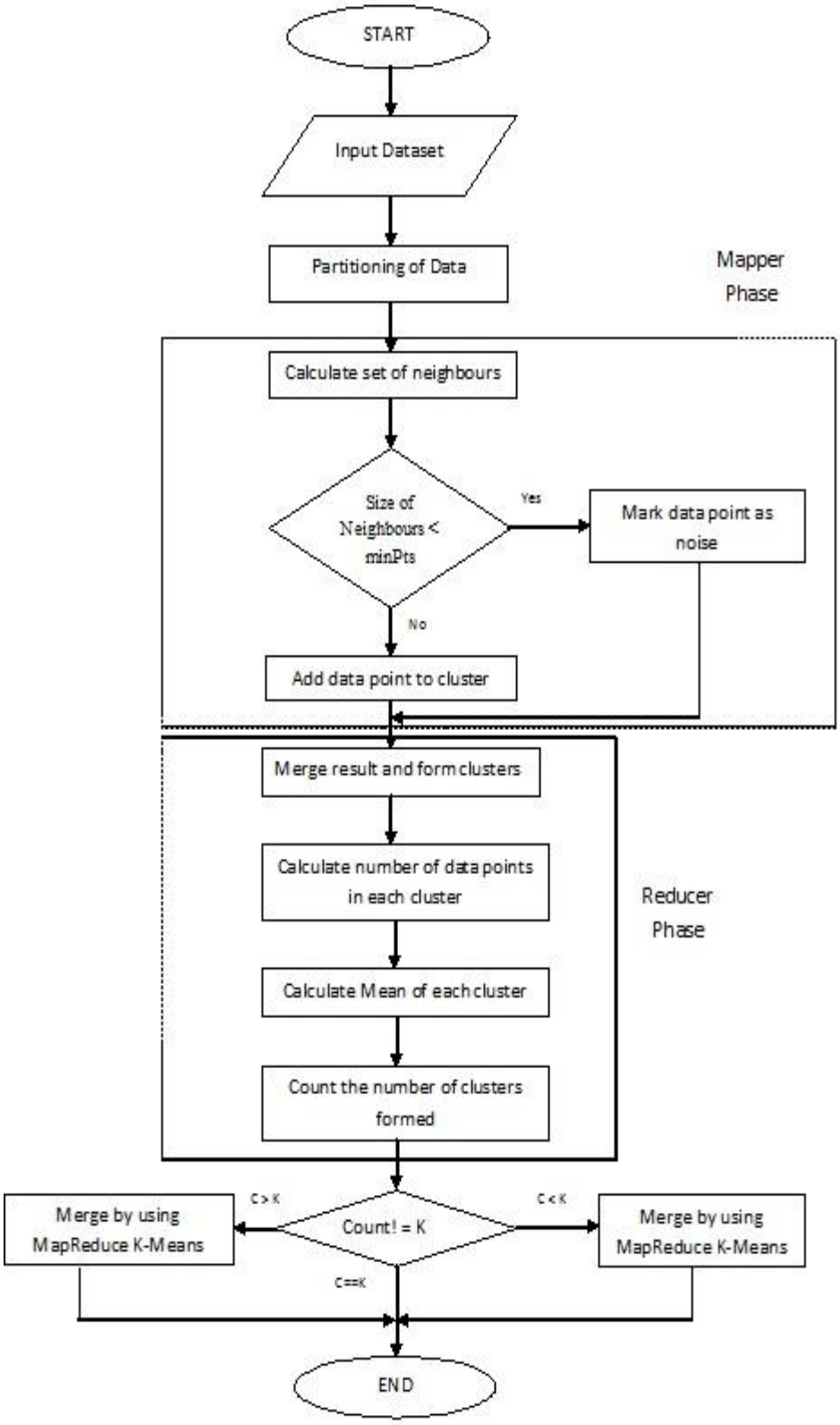

Figure 4.5 Flowchart of Proposed Algorithm

4.2 Proposed Algorithm: Execution Stages

The detailed explanation of how the proposed algorithm works is given here.

Stage 1: Partitioning of Data

The data partition is the first stage in which dataset is partitioned into different regions so that each partition can be executed simultaneously on different nodes in cloud environment. The main issue of partitioning is load balancing and shuffling cost. The raw data is needed to be partitioned so that each node gets almost equal number of data points so that their execution time can be similar. For partitioning dataset first all data points in dataset is sorted in ascending order. After sorting data points the grid cell of 2 eps is build. The reason for choosing width of 2 eps is because it minimizes the boundary points and contains enough information. The list of partitions with different data points is generated for the mapper stage.

Stage 2: Mapper Stage

The mapper stage takes the input from stage 1. The input given to mapper stage is partition region with its id as key and its data points as value. The mapper function runs on different nodes simultaneously generating the output as combination of PiCi as partition and cluster id as key and data points value. In the mapper function first the neighbor set is formed in data points that are within the radius eps are included. The distance between the data points is calculated by Euclidean distance.

Formula used to calculate distance:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad \text{.....2}$$

where

d is the distance between the centroids.

The size of each neighbor set is checked and if it is less than minPts the number of points required to be in the cluster than it is marked as noise. The main advantage of marking

noise is that it tells about the uncertain behavior and is useful in detecting fraud cases etc. If size of neighbor is greater or equal to minPts than it is added to the cluster and data point is marked as visited. The clusters are formed on each partition and each cluster gets its id as the combination of partition id Pi and cluster id Ci as PiCi. The output generated by the mapper phase is input to reducer phase.

Stage 3: Reducer Stage

The input for the reducer stage is the output of mapper stage that is PiCi as key and data points. In the reducer stage the result of the mapper stage are merged to form the global clusters. After that the data points in each cluster are counted. The mean of each cluster is calculated which represents the center of that cluster. The id for each cluster is replaced by its center as the new id of that cluster. The total number of clusters formed is counted in this phase. The output of the reducer stage is the global clusters with data points as value and center mean Cm as their id.

Note: If the clusters formed by the above stage are more than the initial clusters K than the clusters are merged by using MapReduce K-Means which is iterated until new K clusters are achieved with new center mean as their id.

If the clusters formed by reducer stage are less than K initial cluster than split the clusters by MapReduce K-Means to achieve new K clusters with new center mean.

4.3 Conclusion

In this chapter the design of proposed algorithm was presented. The detailed explanation of proposed method was given along with its execution diagram and flow chart. In the next chapter the implementation of this proposed approach is conducted on different datasets and results of the experiment would be gathered.

Chapter 5

Implementation and Experimental Results

This chapter of thesis work will focus on tools for setting Cloud environment, implementation of the proposed hybrid algorithm on MapReduce framework of Hadoop and experimental results.

5.1 Tools for Setting Up Cloud Environment

Big data analytic applications have different configuration, composition and deployment requirements. Tools required to implement the big data analytics application are given below.

- Hadoop [34] – Hadoop is an open source software framework that is suited for large datasets (can be of terabytes or petabytes) across large clusters of computers (hundreds or thousands of nodes). Hadoop is an implementation of Google's simple programming model MapReduce. The map function transforms and synthesizes input data given by user and the reduce function aggregates the output acquired by the map function. Hadoop is based on simple concept of any data will fit. The runtime environment is provided by hadoop and the developer just need to give the map and reduce which are to be executed.
- Hadoop Architecture – Figure 5.1 shows the architecture of hadoop which consists of two main components distributed file system (HDFS) and Execution engine (MapReduce). The HDFS is inspired by the Google File System (GFS) and is designed to store very large data across machines in a large cluster. Each file is stored as a sequence of blocks which are of equal size except the last block. Replication of data is done hence it provides reliability. The MapReduce [35] is a simple programming model by Google which divides application into many small blocks. The MapReduce has two phases map phase which operates on input as key / value pair. The reduce phase produces output in form of key / value pair. The MapReduce engine has the job tracker which is present on the master node (name node) whose function is to assign the jobs or tasks to the task tracker. The

job tracker decides how many and where task will run. Task tracker is present on slave node (data node), performs task of map and reduce task till their completion.

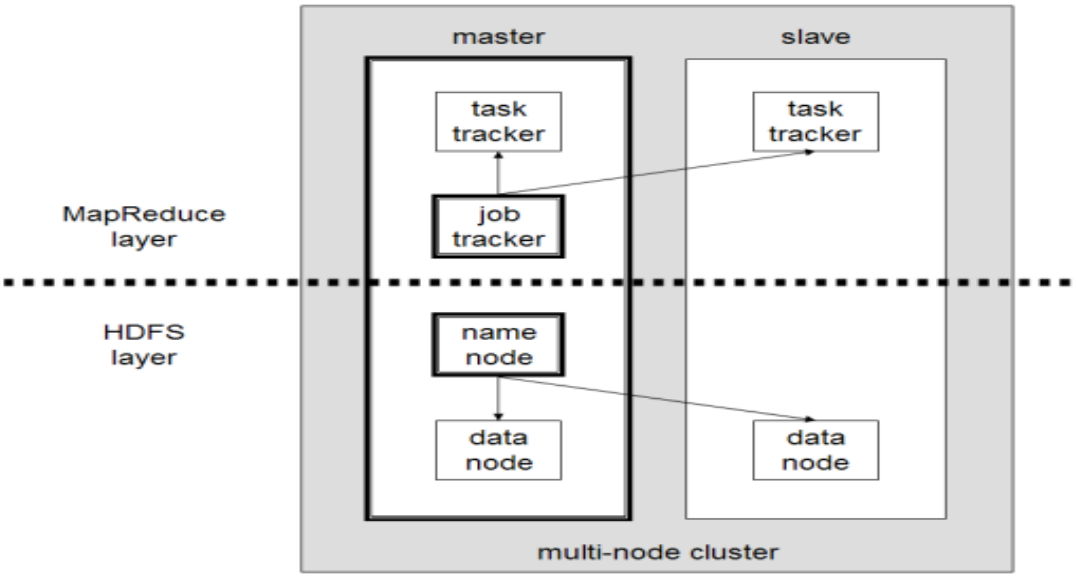

Figure 5.1 Hadoop Architecture [35].

- Eclipse – Eclipse [36] provide an integrated develop environment and extensible plug-in. It is multi-language software written in Java and is to develop Java applications, and provides different plug-ins for other languages like Ada, C, C++ etc. The in-built Eclipse SDK is for only Java developers. The users can write their own plug-ins to extend Eclipse abilities.

5.2 Implementation of Proposed Technique

To implement the proposed approach four different datasets are used. These datasets are collected from different repositories. The sample dataset is from Clustering Dataset Repository [37], US_census dataset is from UCI Machine Learning Repository [38], while iris dataset and balance dataset is from KEEL-dataset Repository [39]. All these datasets have different dimensionality. The proposed approach is implemented by Java using Eclipse on Hadoop platform under the environment of 2.3 GHz Intel(R) Core i3 processor, GB RAM and Ubuntu 12.0.4.

5.2.1 Datasets Used

i) **Sample Dataset**: The sample dataset is a three dimensional dataset which has more than ten thousand data points. This dataset has simple decimal data points. All three approaches are implanted by using this dataset to get the execution time. The snapshots below show the implementation of all three algorithms on this dataset. Figure 5.2 and 5.3 shows the clusters formed with its id and number of data points in each cluster by K-Means MapReduce.

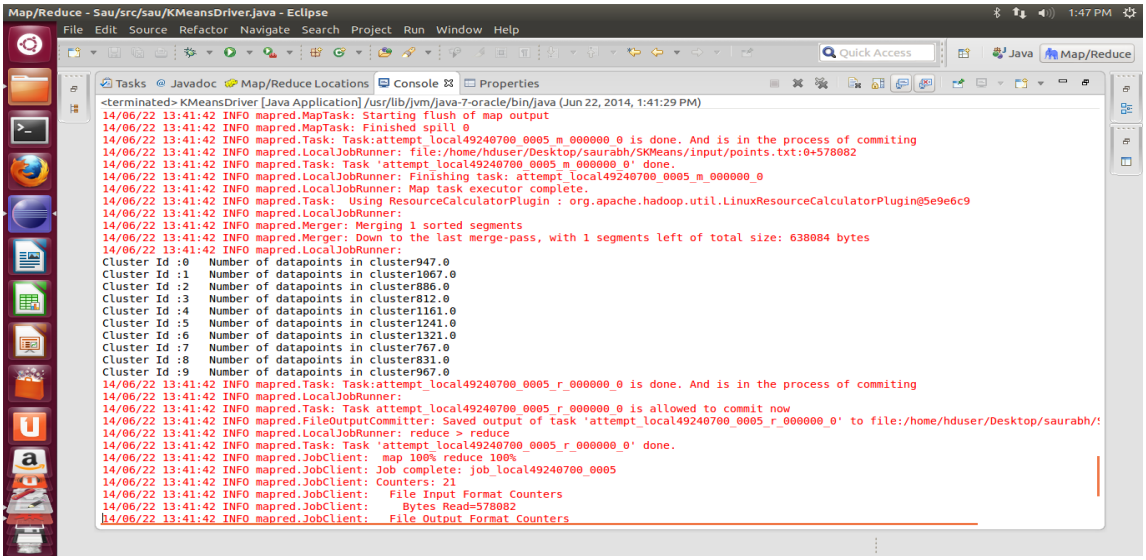

Figure 5.2 K-Means-MR Clusters with Id and Data points in Each Cluster of Sample Dataset

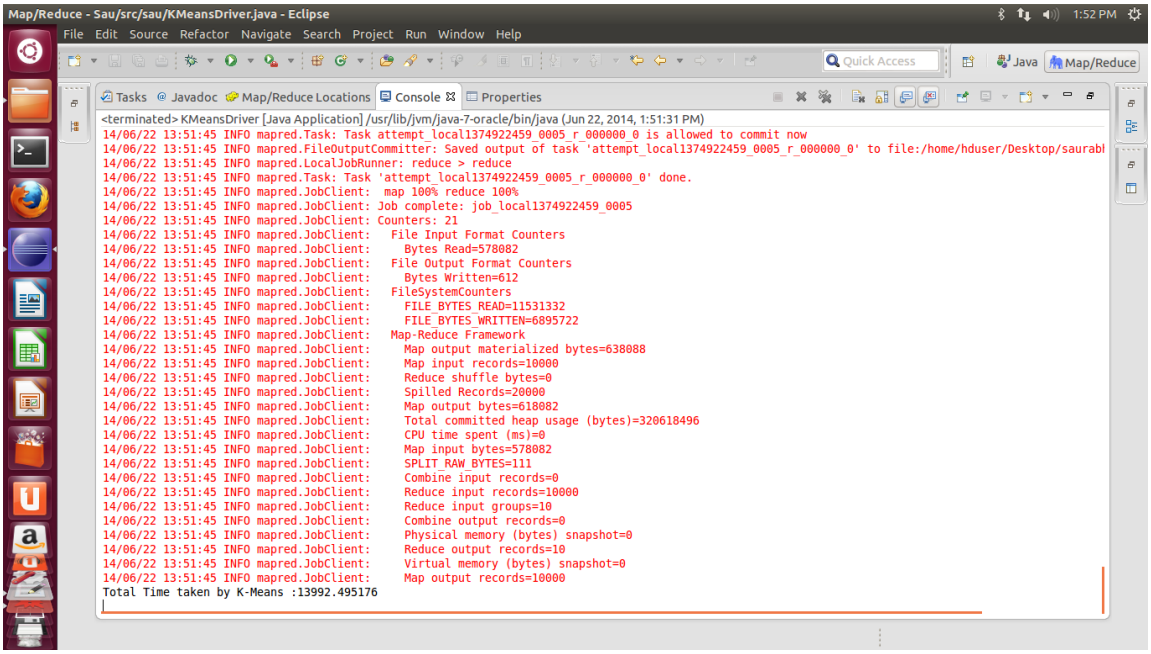

Figure 5.3 K-Means-MR Execution Time on Sample Dataset

Figure 5.4 shows the clusters formed when DBSCAN-MR is used on sample dataset while Figure 5.5 shows total execution time taken by this algorithm.

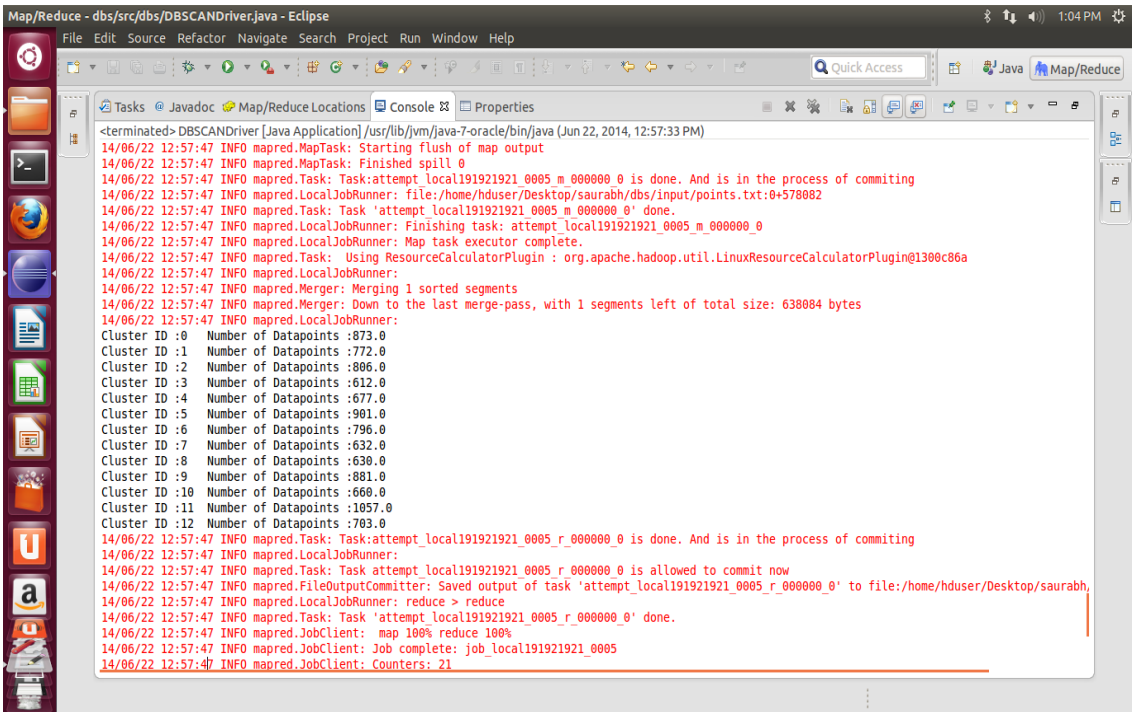

Figure 5.4 DBSCAN-MR Clusters Formed on Sample Dataset

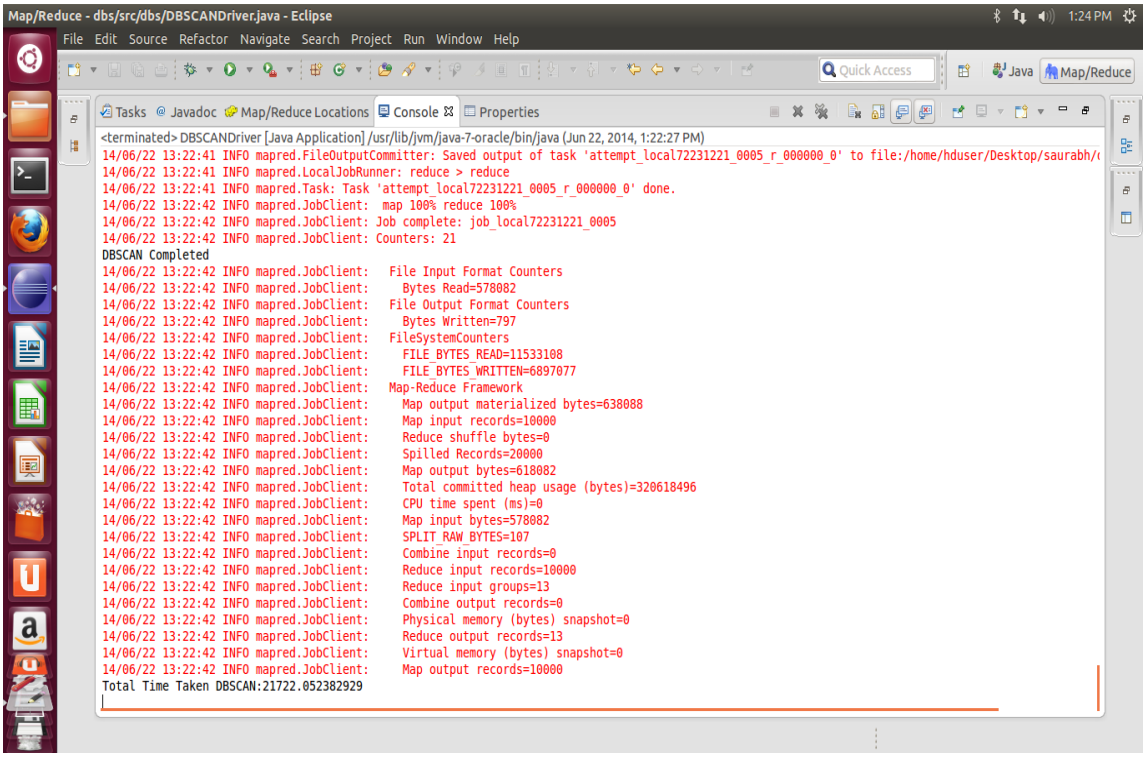

Figure 5.5 DBSCAN-MR Execution Time on Sample Dataset

Figure 5.6 and 5.7 shows the clusters each with total number of data points and their cluster id and the total time taken by the proposed approach.

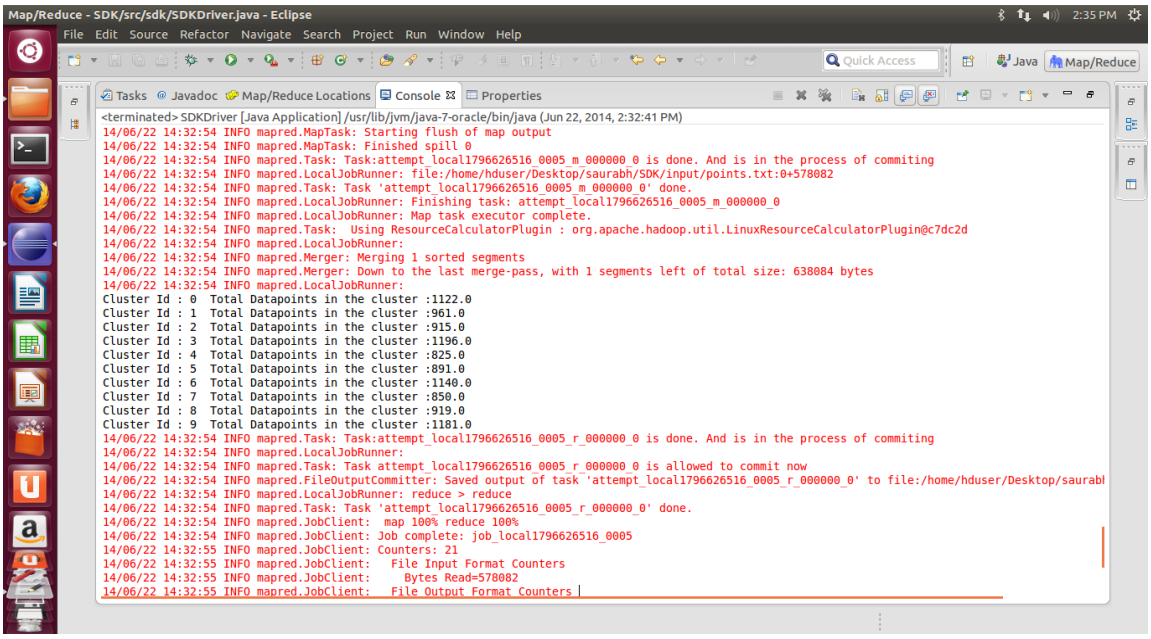

Figure 5.6 Clusters Formed by Proposed Method on Sample Dataset

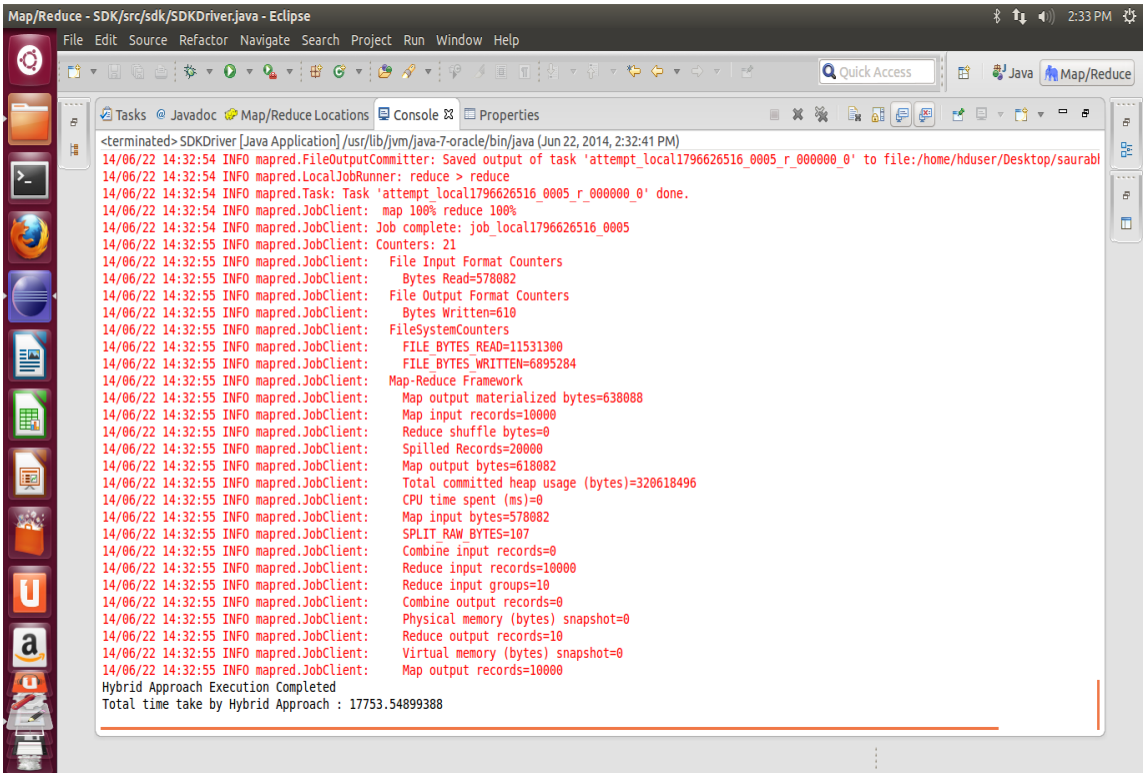

Figure 5.7 Proposed Method Execution Time on Sample Dataset

ii) **US Census 1990 Dataset**: The US census dataset is n dimensional dataset. This dataset contains information regarding the citizens of US with their age, id, cases, cases pending etc. US census has more than twenty thousand data points. This dataset is used to calculate the execution time of all three algorithms. Figure 5.8 and 5.9 shows the clusters formed and the execution time taken by K-Means-MR algorithm.

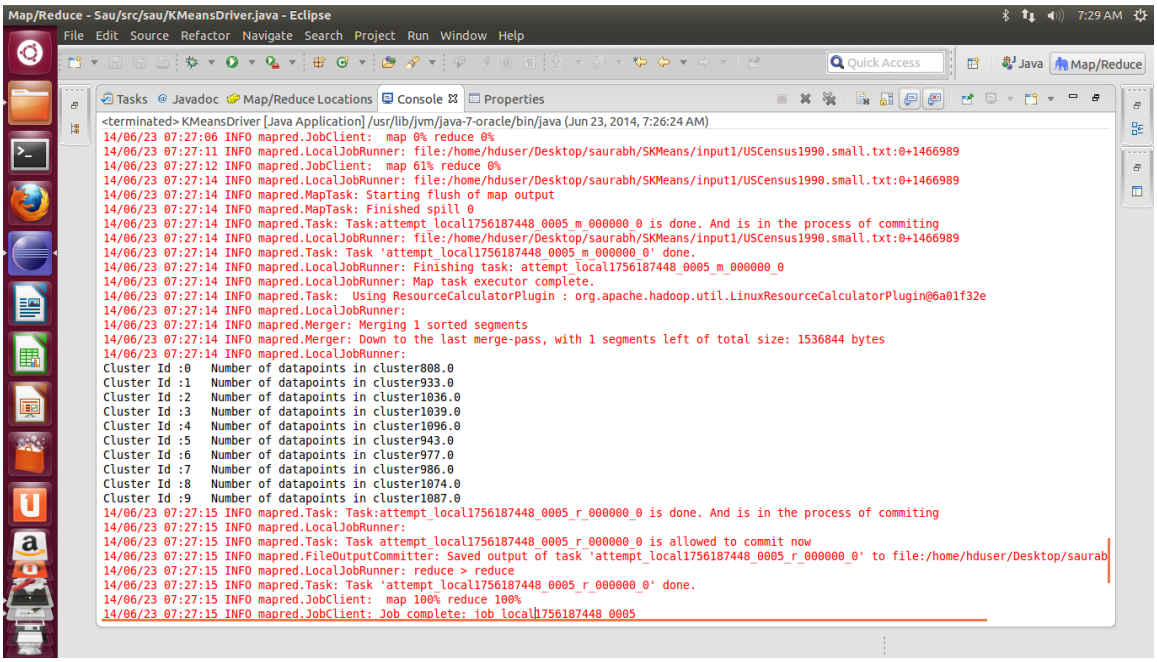

Figure 5.8 K-Means-MR Clusters of US Census Dataset

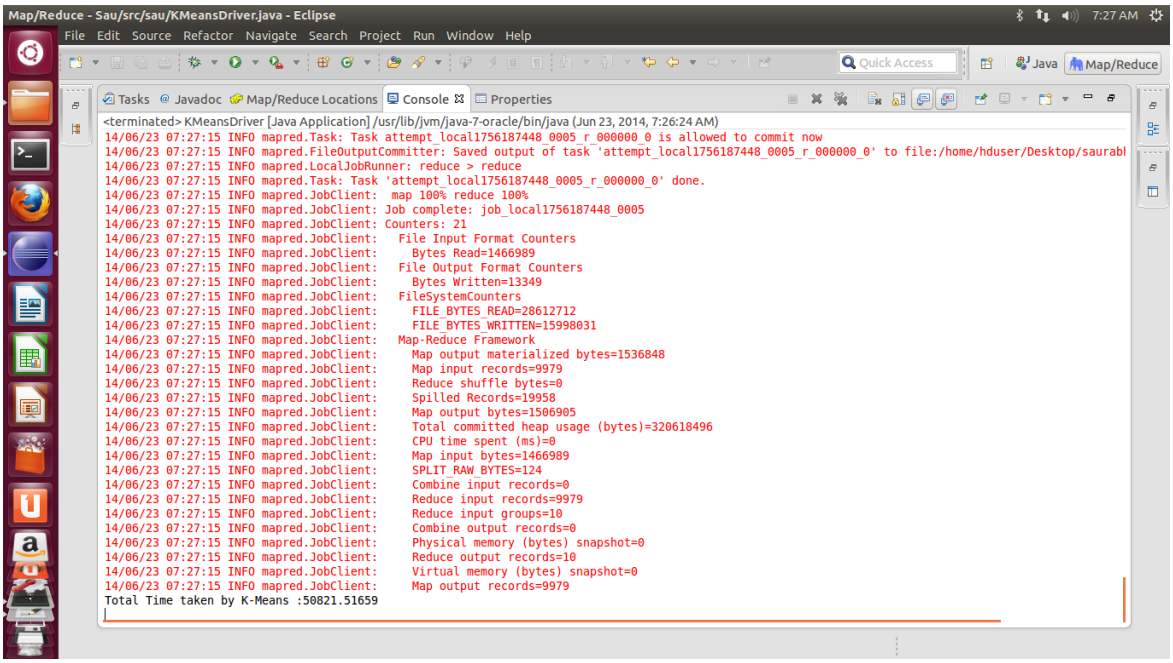

Figure 5.9 K-Means-MR Execution Time on US Census Dataset

Figure 5.10 and 5.11 shows the US census datasets clusters each with total number of data points and their cluster id and the total time taken by DBSCAN-MR.

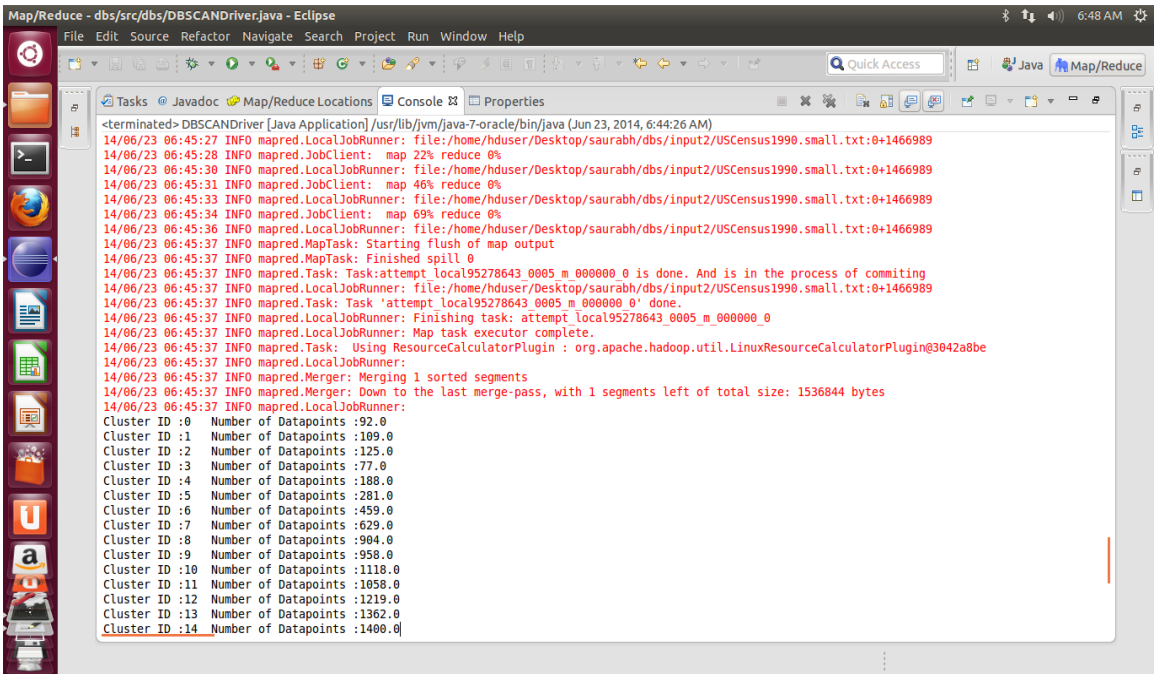

Figure 5.10 DBSCAN-MR Clusters of US Census Dataset.

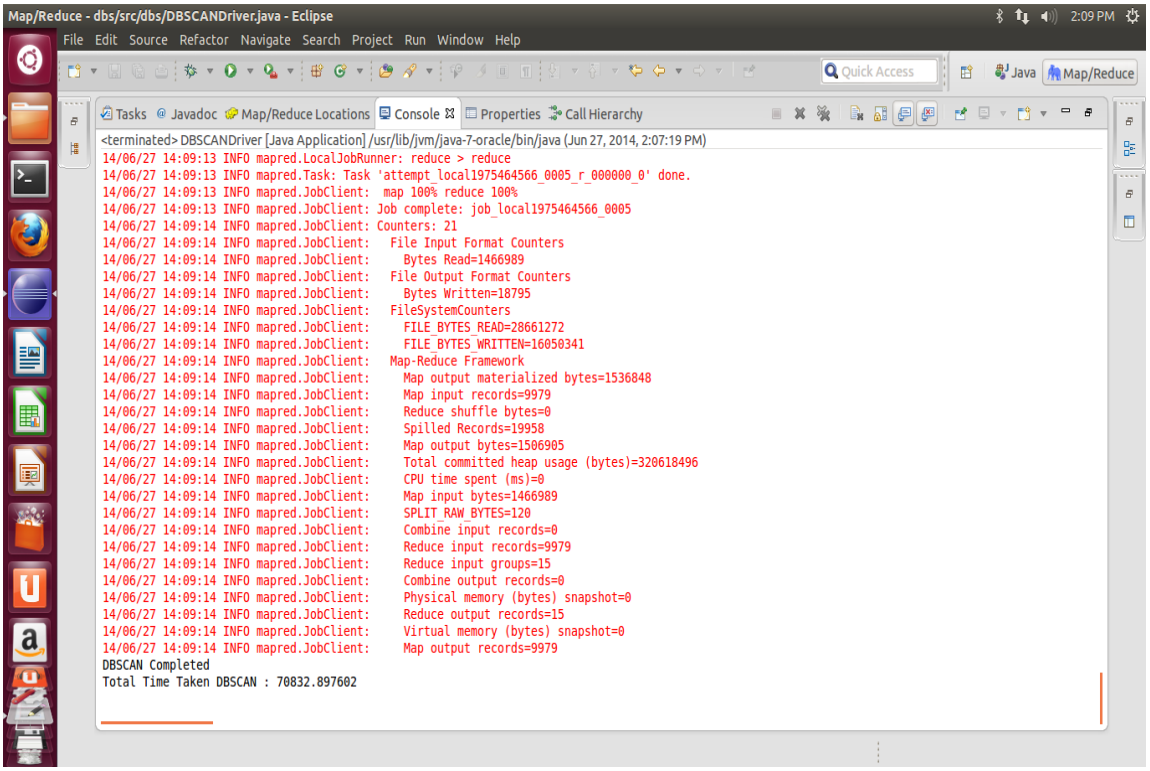

Figure 5.11 DBSCAN-MR Execution Time on US Census Dataset.

Figure 5.12 shows the clusters formed when proposed approach is used on US census dataset while Figure 5.13 shows total execution time.

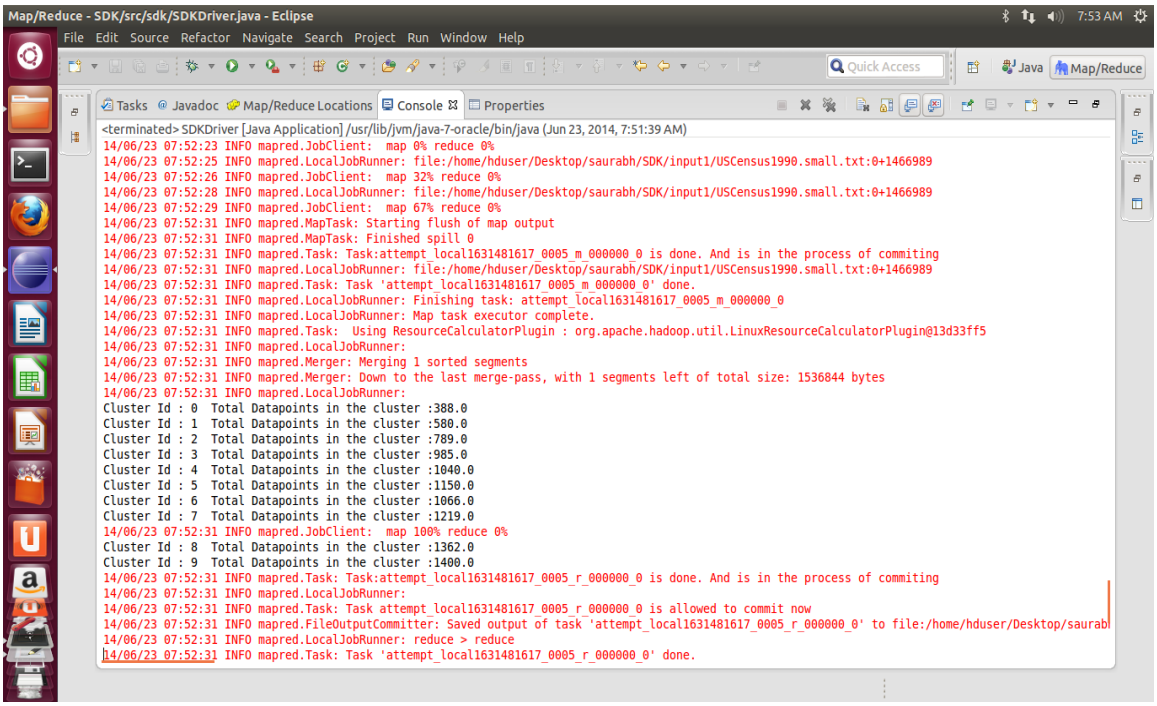

Figure 5.12 Proposed Method Forming Clusters of US Census Dataset.

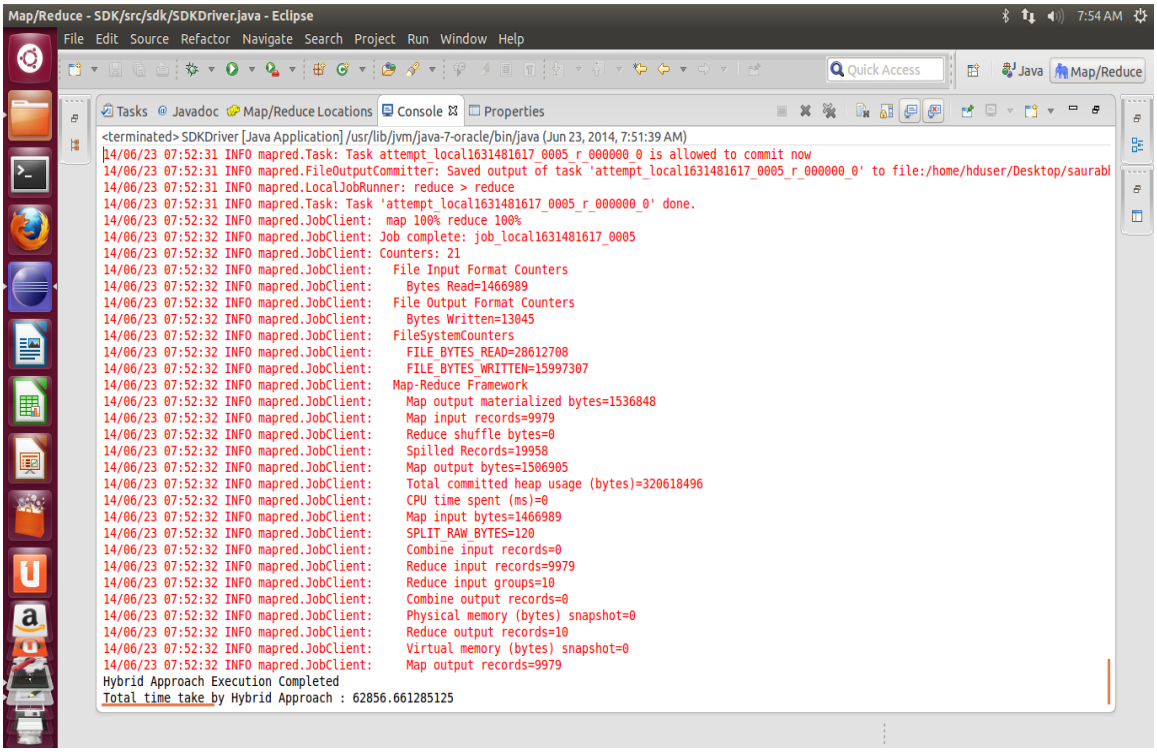

Figure 5.13 Execution Time Taken by Proposed Method on US Census Dataset.

iii) **IRIS Dataset**: The iris dataset is a very famous dataset which is used for evaluating the performance of the clustering techniques. The iris dataset is four dimensional which has 3 class labels. This dataset has 150 points which are divided into three categories (setosa, versicolour, vegenica) of each flower and each category has 50 points. The iris dataset has attributes petal length and width and sepal length and width of each flower. Figure 5.14 shows the clusters formed each with its id and total number of points in it while figure 5.15 shows the execution time and the accuracy performance of the K-Means-MR algorithm.

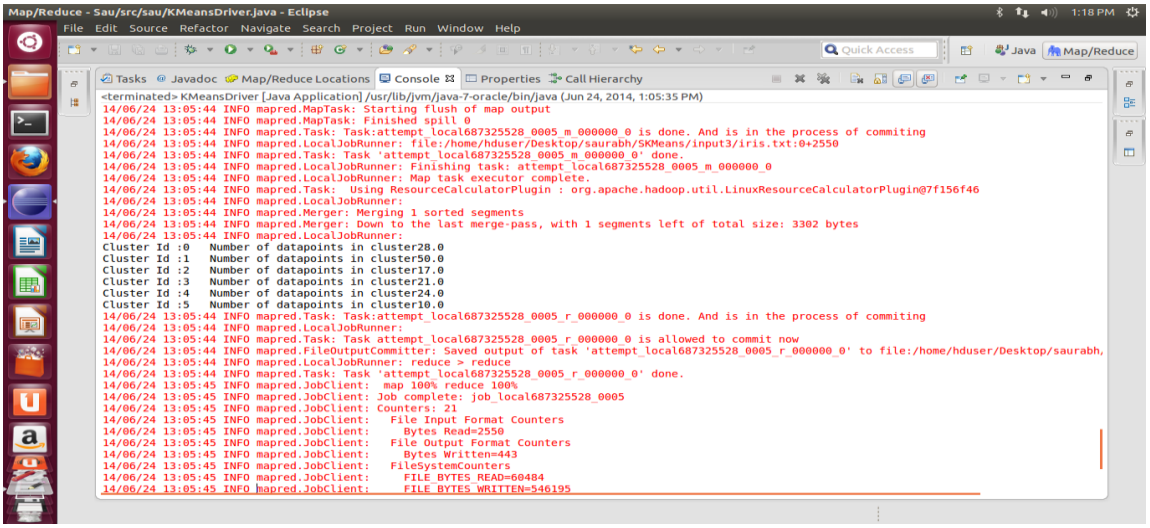

Figure 5.14 K-Means-MR Clusters with Id and Total Data points of Iris Dataset.

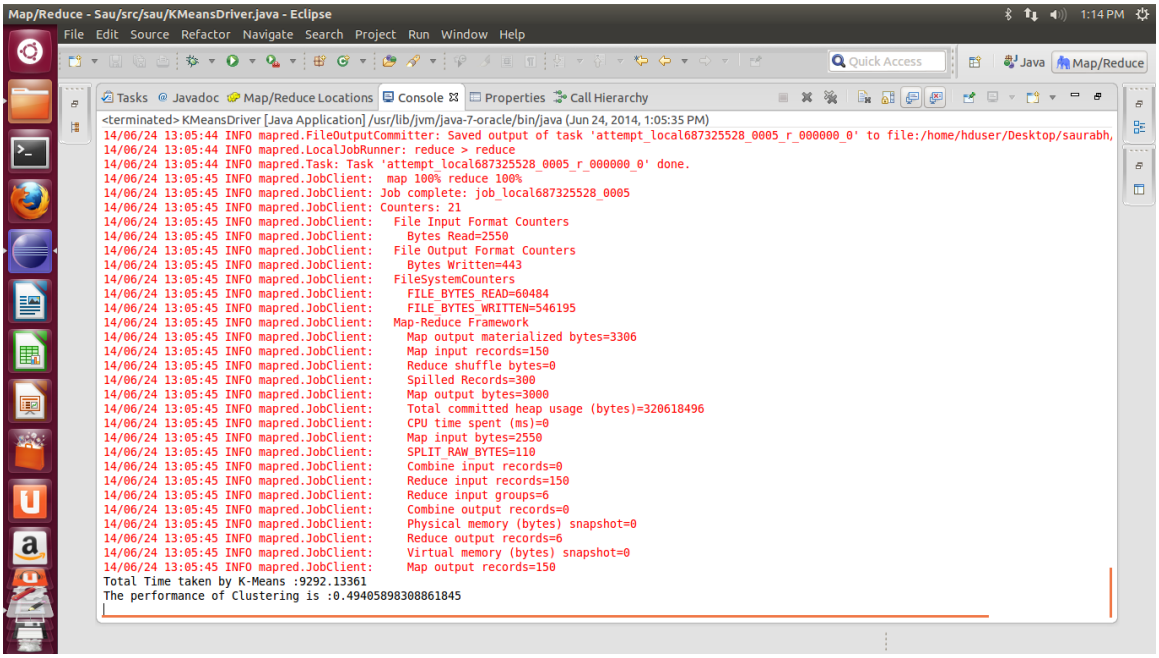

Figure 5.15 K-Means-MR Execution Time and Accuracy Performance on Iris Dataset

Figure 5.16 and 5.17 shows the clusters each with total number of data points and their cluster id and the total time taken by DBSCAN-MR on IRIS dataset.

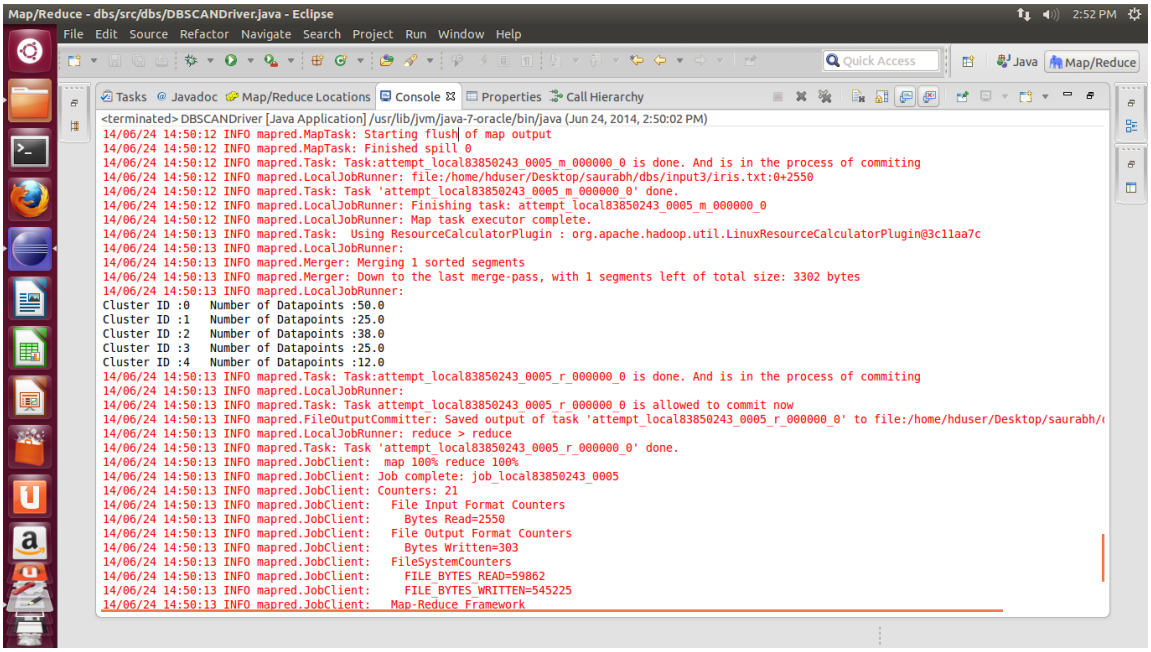

Figure 5.16 DBSCAN-MR Cluster's Formed Using Iris Dataset.

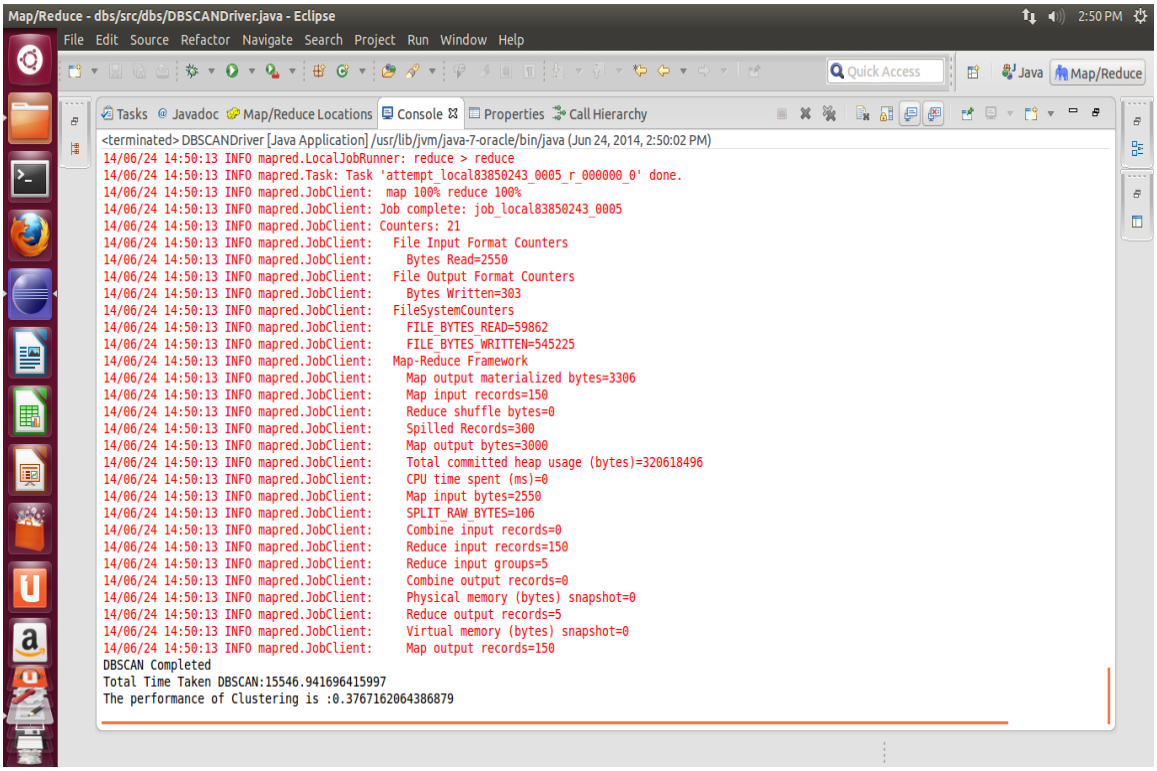

Figure 5.17 DBSCAN-MR Execution Time and Accuracy Performance on Iris Dataset.

The proposed algorithm when applied to IRIS dataset its performance is shown in the figure 5.18 which shows the clusters formed and figure 5.19 which shows the accuracy performance of proposed approach.

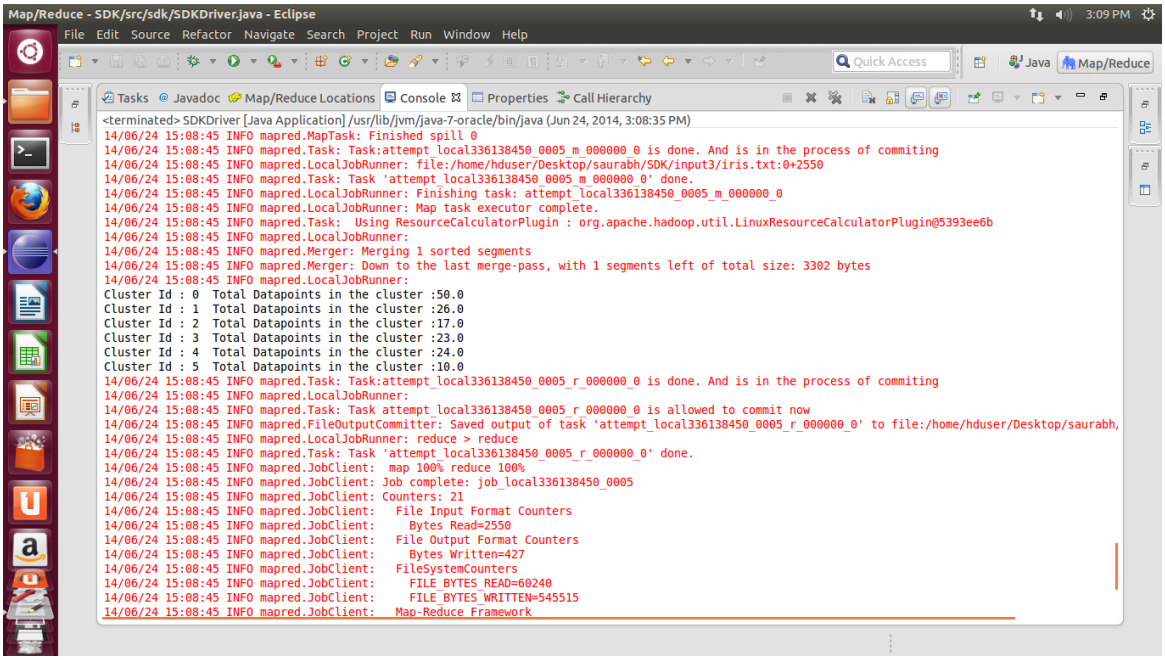

Figure 5.18 Cluster's Formed by Proposed Method Using Iris Dataset.

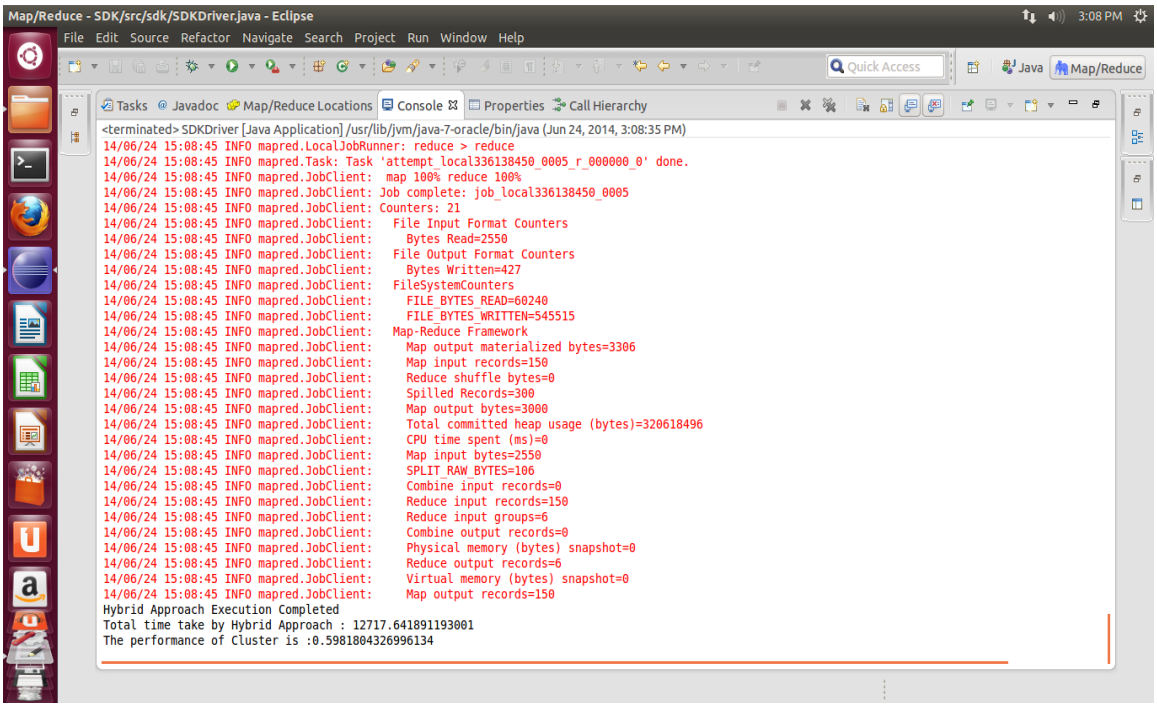

Figure 5.19 Execution Time and Performance of Accuracy of Proposed Method on Iris Dataset.

iv) **Balance Scale Dataset**: The balance scale dataset is four dimensional. It has attributes left weight, left distance, right weight and right distance. Its three labels are balanced, right balanced and left balanced. This dataset is also used to evaluate the accuracy performance and execution time of all three algorithms. First K-Means-MR algorithm is implemented on this dataset than DBSCAN-MR followed by proposed approach. Figure 5.20 and Figure 5.21 shows cluster formed with its id and number of data points in it and the performance of K-Means-MR while figure 5.22 and 5.23 gives DBSCAN-MR clusters formed and its performance. Figure 5.24 shows the performance in terms of accuracy and execution time taken by proposed approach along with cluster formed with their id and total number of data points in each cluster.

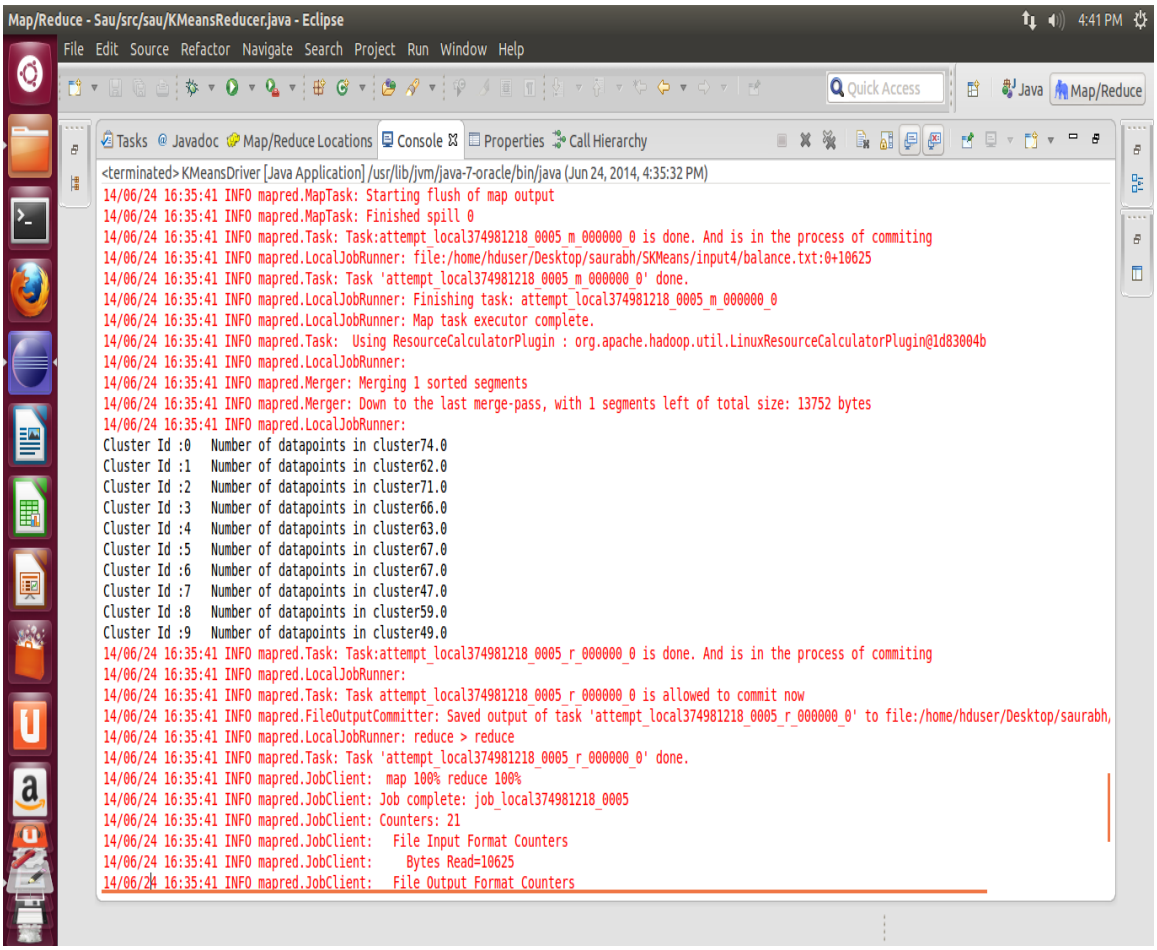

Figure 5.20 K-Means-MR Clusters with Id and Data points in Each Cluster of Balance Scale Dataset.

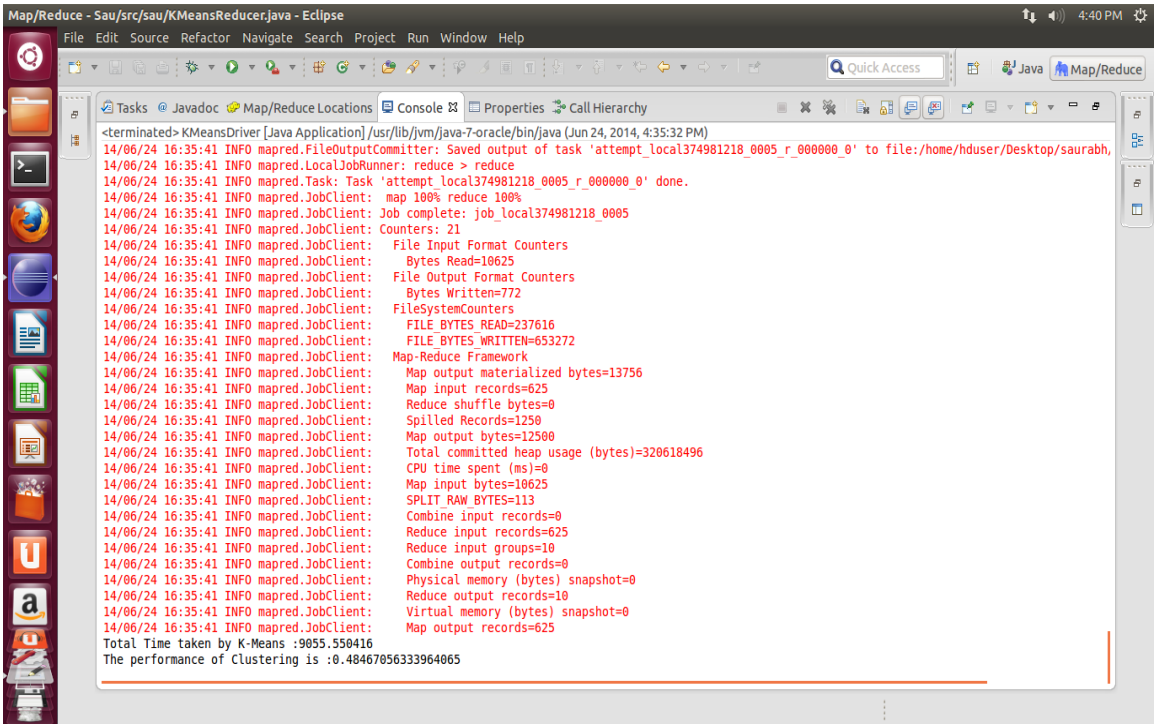

Figure 5.21 K-Means-MR Execution Time and Accuracy Performance on Balance Scale Dataset.

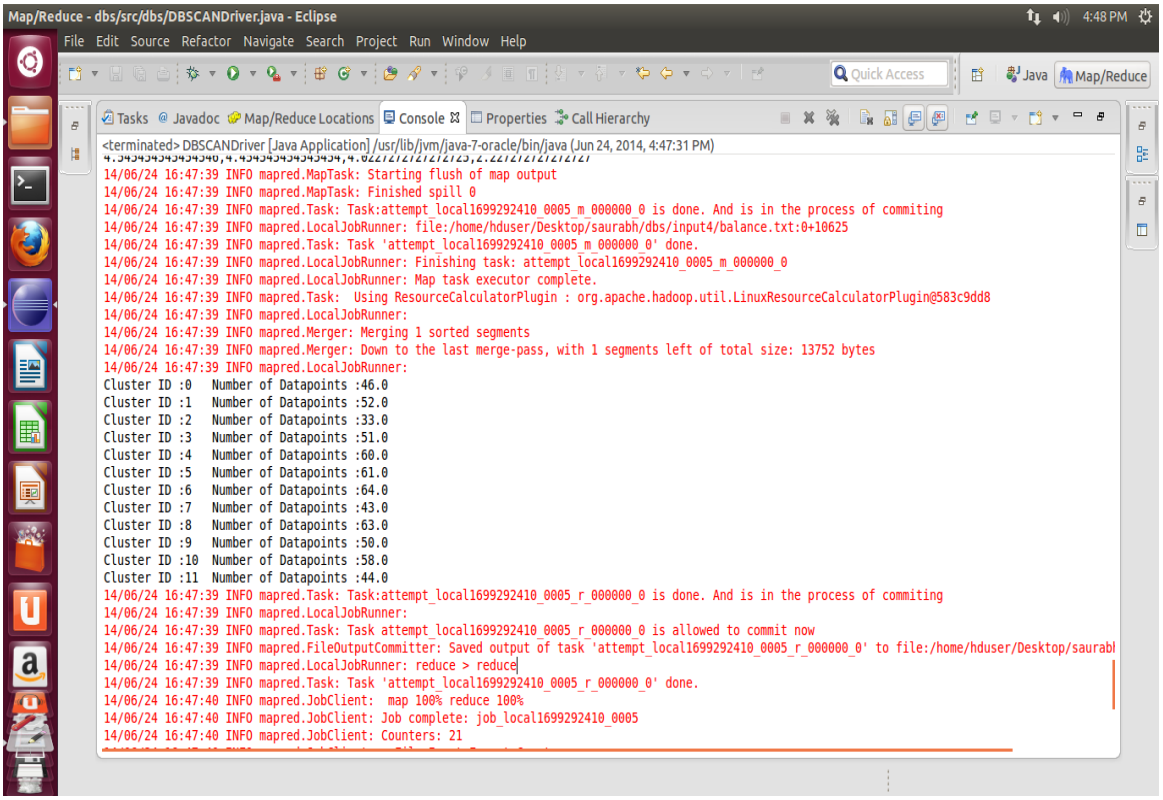

Figure 5.22 Cluster's Formed by DBSCAN-MR on Balance Scale Dataset.

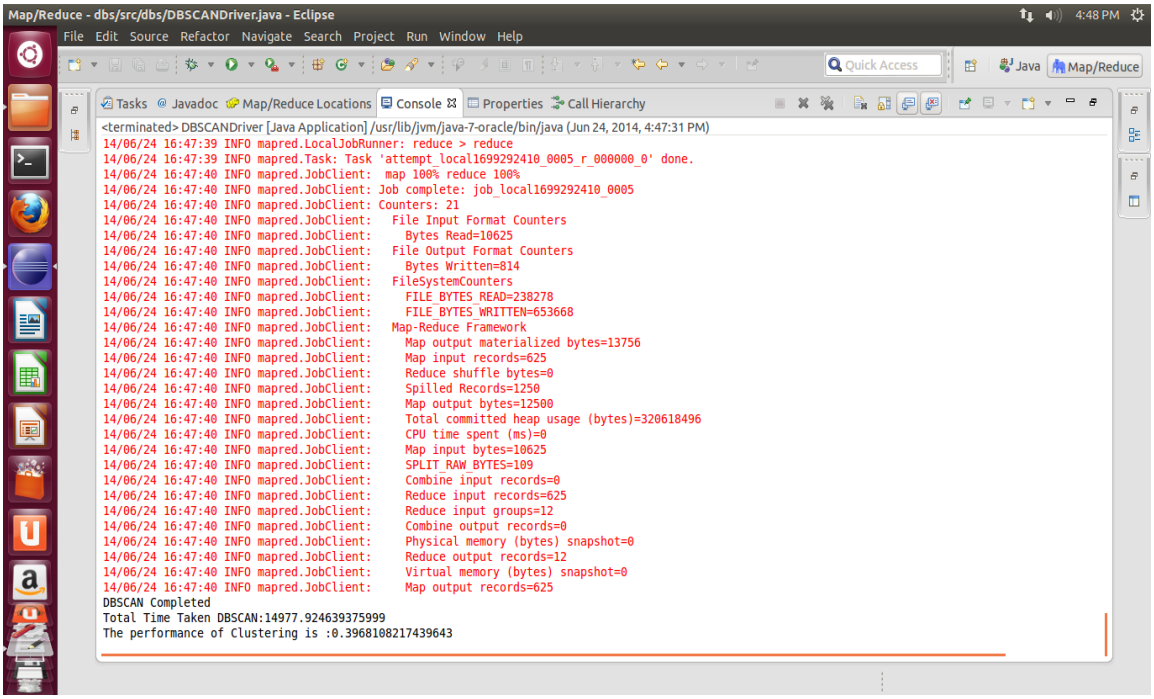

Figure 5.23 DBSCAN-MR Performance and Execution Time on Balance Scale Dataset.

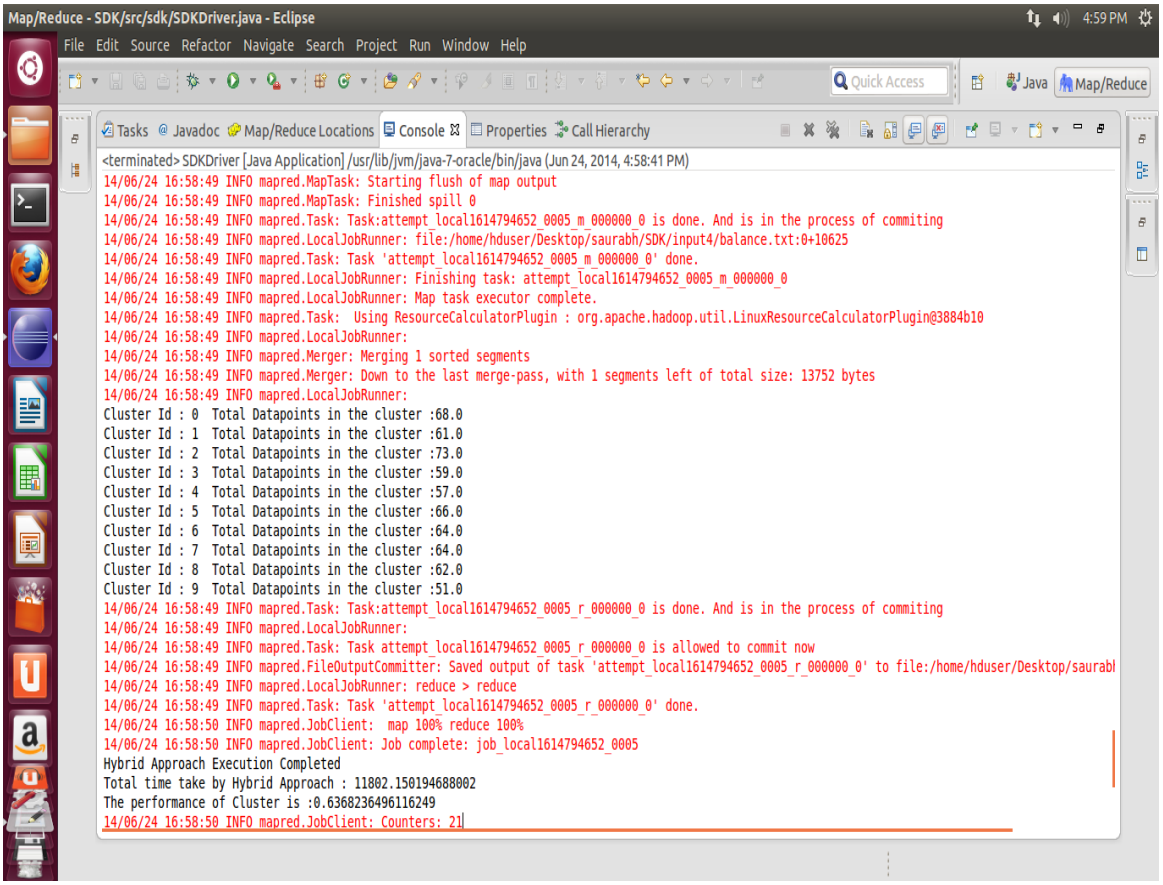

Figure 5.24 Proposed Method's Clusters, Execution Time and Performance on Balance Scale Dataset.

5.3 Experimental Results

The experiments performed on these algorithms are on the basis of two parameters that is the execution time and accuracy performance.

i) Test for Execution Time

All the three algorithms that is K-Means-MR, DBSCAN-MR and proposed approach are tested on the same datasets to calculate their execution time. Table 5.1 shows the comparison of the execution time taken by each algorithm on four different datasets that includes sample dataset, US census dataset, iris dataset and balance scale dataset. Figure 5.25 shows the comparison graph of these algorithms for the all four datasets.

Table 5.1: Comparison of Execution Time (in seconds)

Datasets / Algorithms	DBSCAN-MR	K-MEANS-MR	PROPOSED METHOD
SAMPLE DS	21.72	13.99	17.75
US_CENSUS	70.83	50.82	62.85
IRIS	15.54	9.29	12.71
BALANCE SCALE	14.927	9.05	11.8

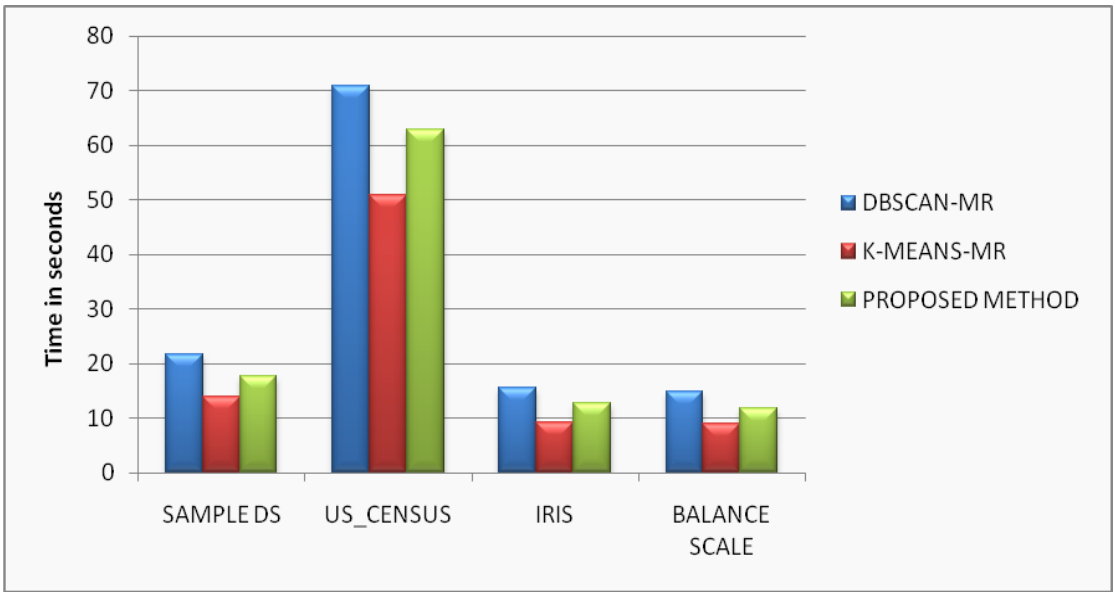

Figure 5.25 Time Comparison Graph

It is seen from graph that the proposed algorithm takes less execution time than DBSCAN-MR algorithm but when it compared to K-Means-MR it takes more time to

execute because proposed algorithm has to deal with the noise while K-Means algorithm does not dealt with noise or boundary points. Noise is an important factor as it shows the uncertain behavior of the commodity which is helpful in detecting frauds cases. Thus proposed algorithm takes more execution time than K-Means-MR.

ii) Test for Performance of Accuracy

All the three algorithms that is K-Means-MR, DBSCAN-MR and proposed approach are tested on two datasets that includes iris dataset and balance scale dataset to calculate the accuracy performance. Accuracy evaluation is needed to maintain the quality of clusters. The Random Index [40] method is used to evaluate the performance of given algorithms. This is a commonly used method to measure similarity of two data clusters.

The formula for Random Index is given below:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \text{ ----------3}$$

Where R is random index.

The Rand Index has a value between 0 and 1 where 0 indicates that two data clusters do not agree on any point of pairs while 1 indicate that clusters are exactly same.

Table 5.2 shows the performance comparison of the three algorithms. Figure 5.26 shows the comparison graph for accuracy performance of these algorithms for the two datasets.

Table 5.2 Performance Comparison of Accuracy

Datasets / Algorithms	DBSCAN-MR	K-MEANS-MR	PROPOSED METHOD
IRIS	37%	49%	59%
BALANCE SCALE	39%	48%	63%

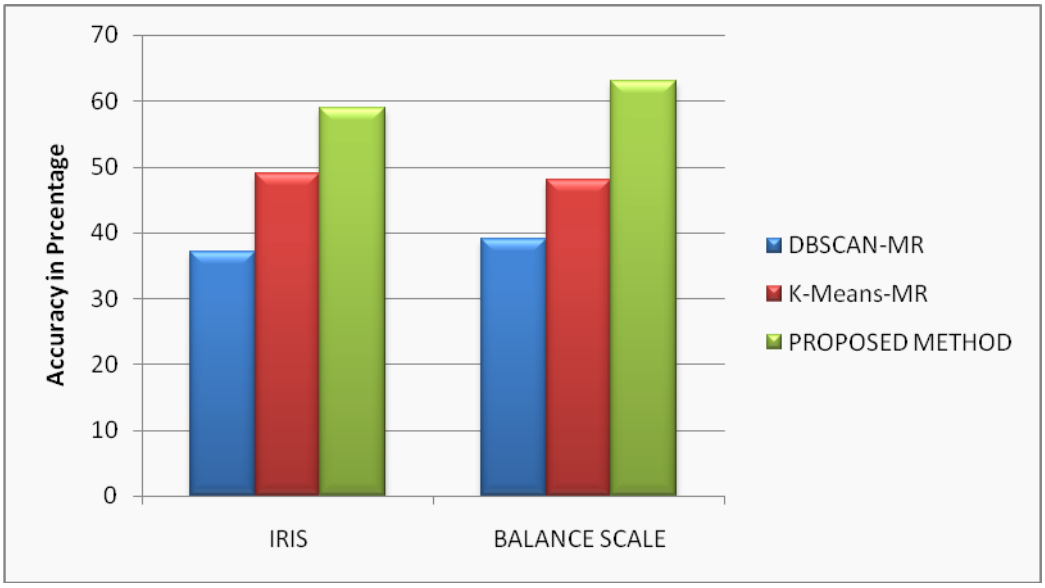

Figure 5.26 Performance Comparison Graph

When these algorithms are tested for the accuracy the proposed algorithm out performs both K-Means-MR and DBSCAN-MR. The performance graph and table clearly demonstrate that proposed algorithm performs better than both algorithms.

5.4 Conclusion

This chapter gives the details of implementation with the experimental results. The results obtained shows that proposed algorithm is better by K-Means-MR and DBSCAN-MR. In the next chapter the conclusion and future directions of this work is presented with thesis contribution.

Chapter 6

Conclusion and Future Scope

This chapter discusses the conclusion of the work done in thesis and ends with a clear vision of future direction which can be taken further.

6.1 Conclusion

This thesis introduces big data and provides background of various clustering techniques used to analyze big data. In this work comparative analysis of these techniques is done. A hybrid approach based on parallel K-Means and parallel DBSCAN for efficient clustering of big data is proposed. This approach is developed in java, deployed in MapReduce framework of Hadoop. The experimental results have been gathered which shows that the proposed approach is more accurate as compared to MapReduce K-Means and MapReduce DBSCAN when tested on the four different datasets with different dimensions.

6.2 Limitations

The proposed approach still has some of the following limitations.

i) The proposed approach still requires the value of K, the number of initial or desired clusters as input, though data points has been distributed.

ii) This approach can be applied only for those data sets which have numerical values or attributes.

6.3 Thesis Contribution

i) In this thesis various current clustering techniques for analyzing big data have been analyzed and are compared according to some parameters.

ii) A hybrid approach based on parallel K-Means and parallel DBSCAN has been designed.

iii) The proposed design is implemented and deployed on Hadoop's MapReduce framework by using Eclipse IDE.

iv) Experimental results demonstrate that the proposed technique is more accurate then K-Means-MR and DBSCAN-MR.

v) Proposed technique takes less time as compared to DBSCAN-MR but more than K-Means-MR as it filter out noise also.

6.4 Future Scope

i) The proposed approach shows that initial value of clusters is needed as input, in future the approach can be enhanced such that automatic number of desired clusters is formed.

ii) In future, this approach can be applied for a particular application area by addressing issues involved and can be applied for data sets with categorical attributes.

References

[1] Picciano, Anthony G, "The Evolution of Big Data and Learning Analytics in American Higher Education," *Journal of Asynchronous Learning Networks*, Vol. 16, No.3, pp. 9-20, 2012.

[2] "Evolution of Big Data and Significant Growth of Data" [online]. Available: http://www.atkearney.com/strategic-it/article/-/asset_publisher/content/big-data-business-models. [3 February 2014].

[3] "Evolution of Big Data" [online]. Available: http://www.forbes.com/sites/gilpress/a-very-short-history-of-big-data. [3 February 2014].

[4] P. Russom, "Big Data Analytics," *TDWI Best Practices Report*, 4th Quarter 2011, 2011.

[5] "Big Data Analytics" [online]. Available: http://techtarget.com/definition/big-data-analytics. [12 November 2013].

[6] A. Katal, M. Wazid and R.H. Goudar, "Big data: Issues, challenges, tools and good practices," *Contemporary Computing (IC3), 2013 Sixth International Conference on*, IEEE, 2013.

[7] Demchenko, Yuri, P. Grosso, C. Laat and P. Membrey, "Addressing big data issues in scientific data infrastructure," *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pp. 8-55. IEEE, 2013.

[8] An Oracle White Paper, "Big Data Analytics, Advanced Analytics in Oracle Databases," March 2013.

[9] J. Manyika, M. Chui, B. Brown, J. Bughhin, R. Dobbs, C. Roxburgh and A.H. Byers "Big data: The next frontier for innovation, competition and productivity," The McKinsey Global Institute, Tech. Rep., May 2011.

[10] T. Hu, H. Chen, L. Huang and X. Zhu "A survey of mass data mining based on cloud-computing," *Anti-Counterfeiting, Security and Identi_cation (ASID), 2012 International Conference*, 2012.

[11] Vats, Prashant, M. Mandot and A. Gosain, "A Comparative Analysis of Various Cluster Detection Techniques for Data Mining," *Electronic Systems, Signal*

Processing and Computing Technologies (ICESC), 2014 International Conference on. IEEE, 2014.

[12] Jain, Anil K., M. N. Murty and P.J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, Vol. 31, No.3, pp. 264-323, 1999.

[13] J. Dong, F. Wang and B. Yuan, "Accelerating birch for clusteringlarge scale streaming data using cuda dynamic parallelism," *Intelligent Data Engineering and Automated Learning{IDEAL 2013*, pp. 09-16.Springer, 2013.

[14] Q. He, X. Jin, C. Du, F. Zhuang and Z. Shi, "Clustering in extreme learning machine feature space," *Neurocomputing*, Vol.128, pp. 88-95, 2014.

[15] R Madhuri, M R Murty, JVR Murthy, PVGD Prasad Reddy and S.C Satapathy "Cluster analysis on di_erent data sets using k-modes and k-prototype algorithms," *ICT and Critical Infrastructure: Proceedings of the 8th Annual Convention of Computer Society of India*, Vol. II, pp 137-144. Springer, 2014.

[16] G. Krishnasamy, A.J. Kulkarni and R. Paramesran "A hybrid approach for data clustering based on modified cohort intelligence and k-means," *Expert Systems with Applications*, 2014.

[17] I.B Saida, K. Nadjet and B. Omar "A new algorithm for data clustering based on cuckoo search optimization," *Genetic and Evolutionary Computing*, pp. 55-64. Springer, 2014.

[18] X.F Jiang "Application of parallel annealing particle clustering algorithm in data mining," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 12, No. 3, pp. 2118-2126, 2014.

[19] K. Musayeva, T. Henderson, J.B Mitchell and L. Mavridis "Pfclust: an optimised implementation of a parameter-free clustering algorithm," *Source code for biology and medicine*, 9(1):5, 2014.

[20] H. Yu, Z. Liu and G. Wang, "An automatic method to determine the number of clusters using decision-theoretic rough set," *International Journal of Approximate Reasoning*, Vol. 55, No. 1, pp. 101-115, 2014.

[21] K. Buza, G.I. Nagy and A. Nanopoulos, "Storage-optimizing clustering algorithms for high-dimensional tick data," *ExpertSystems with Applications*, 2014.

[22] S. WANG, J. FAN, M. FANG and H. YUAN, "Hgcudf: Hierarchical grid clustering using data," *Chinese Journal of Electronics*, 2014.

[23] I. Naim, S. Datta, J. Rebhahn, J.S Cavenaugh, T.R. Mosmann and G. Sharma, "Swiftscalable clustering for automated identi_cation of rare cell populations in large, high-dimensional flowcytometry datasets," part 1: Algorithm design. *Cytometry Part A*, 2014.

[24] A. Amini, H. Saboohi, T.Y. Wah and T. Herawan, "Dmm-stream: A density mini-micro clustering algorithm for evolving datastreams," *Proceedings of the First International Conference on AdvancedData and Information Engineering (DaEng-2013)*, pp- 675-682, Springer, 2014.

[25] Y. Kim, K. Shim, M.S Kim and J.S Lee, "Dbcure-mr: An efficient density-based clustering algorithm for large data using mapreduce," *Information Systems*, Vol. 2, pp. 15-35, 2014.

[26] C.W Tsai, B.C Huang, M. Chiang, "A novel spiral optimization for clustering," *Mobile, Ubiquitous, and Intelligent Computing*, pp. 621-628. Springer, 2014.

[27] NNR Suri, M.N Murty, G Athithan, "A ranking-based algorithm for detection of outliers in categorical data", *International Journal of Hybrid Intelligent Systems*, Vol. 11, No.1, pp. 1-11, 2014.

[28] P. Jiang, J. Peng, M. Heath and R. Yang, "A clustering approach to constrained binary matrix factorization", *Data Mining and Knowledge Discovery for Big Data*, pp. 281-303. Springer, 2014.

[29] LC Hsieh, GL Wu, YM Hsu and W. Hsu, "Online image search result grouping with mapreduce-based image clustering and graph construction for large-scale photos," *Journal of Visual Communication and Image Representation*, Vol. 25, No. 2, pp. 384-395, 2014.

[30] X. Wu, X. Zhu, GQ Wu and W. Ding, "Data mining with big data," *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 26, No. 1, pp. 97-107, 2014.

[31] Zhao, Weizhong, Huifang Ma, and Q. He, "Parallel k-means clustering based on mapreduce," *Cloud Computing*, Springer Berlin Heidelberg, pp-674-679, 2009.

[32] He and Yaobin, "Mr-dbscan: An efficient parallel density-based clustering algorithm using MapReduce," *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*. IEEE, 2011.

[33] Dai, Bi-Ru and IC Lin, "Efficient map/reduce-based dbscan algorithm with optimized data partition," *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012.

[34] "Hadoop" [online]. Available: http://hadoop.apache.org/. [6 October 2013].

[35] "Hadoop Architecture" [online]. Available: http://web.cs.wpi.edu/~cs525/513-MYE/lecture/1/hadoop.pptx. [6 October 2013].

[36] "Eclipse" [online]. Available: http://www.eclipse.org. [8 October 2013].

[37] "Clustering Dataset Repository" [online]. Available: http://cs.joensuu.fi/sipu/dataset. [10 March 2014].

[38] "UCI Machine Repository" [online]. Available: https://archive.ics.uci.edu/ml/dataset. [13 March 2014].

[39] "KEEL-Dataset Repository" [online]. Available: http://sci2s.ugr.es/keel/datasets.php. [20 March 2014].

[40] Rand, M. William, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association,* Vol. 66, No. 336, pp. 846-850, 1971.

List of Publications

1. Saurabh Arora and Dr. Inderveer Chana, “A Survey of Clustering Techniques for Big Data Analysis”, 5th International Conference- Confluence 2014 on Cloud Security and Big Data”, IEEE, 2014.[Accepted]