Hierarchical Clustering Algorithm for Big Data using Hadoop and Mapreduce

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

Master of Engineering
in
Software Engineering

Submitted By
Piyush Lathiya
(801431011)

Under the supervision of:
Dr. Rinkle Rani
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Hierarchical Clustering Algorithm for Big Data using Hadoop and Mapreduce*" , in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software 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. Rinkle Rani 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.

P.R Lathiya

(Piyush Lathiya)

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

Rinkle 17/06/16

(Dr. Rinkle Rani)
Associate Professor,
Computer Science and
Engineering Department

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds. With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Rinkle Rani**, Associate Professor, Computer Science and Engineering Department, Thapar University for her positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all her blessings. She has been a source of inspiration for me.

I am grateful to **Dr. Maninder Singh**, Head of Department and **Dr. Rupali Bhardwaj**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis.

I will be failing in my duty if I do not express my gratitude to **Dr. S. S. Bhatia**, Senior Professor and Dean of Academics Affairs in the University, for making provisions of infrastructure such as library facilities, computer labs equipped with internet facility, immensely useful for the learners to equip themselves with latest in the field.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

P.R. Lathiya

Piyush Lathiya

(801431011)

Abstract

Mining of massive data sets is the need of the hour in present computer science industry. The exponential growth in the number of users on internet and volume of available data force research to think about efficient approach to store data and analyze useful patterns out of it. Extracting useful information out of massive data and process them in less span of time has become crucial part of Data mining. There are many approach exist to cluster data objects based on similarity. CURE (Clustering Using Representatives) is very useful hierarchical algorithm which has ability to identify cluster of arbitrary shape and able to identify outliers. However traditional CURE algorithm is based on processing in single machine hence can't cluster large amount of data in efficient way.

In this thesis, CURE algorithm is proposed along with Distributed Environment using Hadoop. To process huge amount of data and to extract useful patterns out of it, distributed environment is the efficient solution so clustering of data objects is performed with the help of Mapreduce Programming model. One of the other advantage of CURE algorithm is to detect outlier points and removed it from further clustering process and improve quality of clusters. The major focus of this thesis has been exploring new approach to cluster data objects using CURE clustering algorithm with the help of Hadoop distributed environment and explore effect of different parameters in outlier detection.

Table of Contents

Certificate.....i

Acknowledgement..... ii

Abstract iii

Table of Contents..... iv

List of Figures..... vii

List of Tables viii

Chapter 1 – INTRODUCTION **1**

1.1 Data Mining…….. 1

1.1.1 Introduction to Data Mining…… 1

1.1.2 Data Mining Process…….. 2

1.1.3 Data Mining Techniques ……... 4

1.2 Clustering…….. 8

1.2.1 Stages of Clustering……... 8

1.2.2 Clustering Algorithm Techniques…….. 9

1.3 Tools/Platform/Language used……. 12

1.3.1 Hadoop Cluster Components……. 13

1.3.2 The Hadoop Distributed File System (HDFS)…… 13

1.3.3 Map/Reduce Programming Model……. 14

1.4 Structure of the Thesis……... 15

Chapter 2 – LITERATURE REVIEW……..……. 16

2.1 BIRCH Clustering Algorithm……..…… 17

2.2 CURE Clustering Algorithm ……... 20

2.3 ROCK Clustering Algorithm……...…… 21

2.4 CHAMELEON Clustering Algorithm…….. 24

Chapter 3 – PROBLEM STATEMENT……. 26

3.1 Research Gaps……...……... 26

3.2 Problem Statement……...……. 26

3.3 Objectives……. 27

3.4 Methodology……...…….. 27

Chapter 4 - FEASIBILITY STUDY…… 28

4.1 Existing Hierarchical Clustering Algorithms….. 28

4.1.1 Implementation of BIRCH in JAVA……... 28

4.1.2 Implementation of CURE in JAVA……. 30

4.1.3 Implementation of CHAMELEON in JAVA….. 32

4.2 Design of Proposed Algorithm……..…….. 34

Chapter 5 – IMPLEMENTATION DETAILS…….. 38

5.1 Implementation of Environment for Hadoop….. 38

5.1.1 Single Machine…… 38

5.1.2 Dedicated Hadoop Cluster……... 38

5.2 Experimental Setup……..…… 39

5.3 Results and Analysis……..…….. 43

Chapter 6 – CONCLUSION AND FUTURE SCOPE……... 50

6.1 Conclusion……..……..... 50

6.2 Summary of Contributions……..…… 50

6.3 Future Scope……..…….. 51

REFERENCES……...……. 52

LIST OF PUBLICATIONS AND VIDEO LINK..……... 56

PLAGIARISM REPORT…… 57

List of Figures

Figure No	Figure Description	Page No
1.1	Data mining steps in KDD……	3
1.2	Neural Network……	5
1.3	Decision Tree……	6
1.4	Stages of Clustering procedure…….	8
1.5	Working of HDFS…….	13
1.6	Map Reduce Workflow…….	14
2.1	Classification of Clustering Algorithm…….	16
2.2	Overview of BIRCH Algorithm……...	19
2.3	Overview of CURE Algorithm…….	21
2.4	Overview of ROCK Algorithm……	22
2.5	Merging choices for CURE……..	23
2.6	Merging choices for ROCK……..	23
2.7	Overview of CHAMELEON Algorithm……..	24
4.1	Output of BIRCH Algorithm with clustered data points..	29
4.2	Clustering Procedure in Traditional CURE Algorithm…	30
4.3	Output clusters of Dataset1(space)……...	32
4.4	Clustering of data points in CHAMELEON…….	33
4.5	Output of CHAMELEON in GNUPlot…….	34
4.6	CURE algorithm with Map Reduce……..	36
5.1	Master node processes……..	40
5.2	Slave node process……	40
5.3	Status of master node and other parameters…….	41
5.4	Status of Secondary namenode…….....	42
5.5	Data stored in Datanode……	43
5.6	cmd execution of Dataset1 in Hadoop environment…….	44

5.7	Output of dataset1…….	45
5.8	Effect on number of outliers with reducing factor……	46
5.9	Execution of Hadoop for Dataset2……...	47
5.10	File system for data file size more than 128 MB……..	49
5.11	Job done for data size more than 128 MB……	49

List of Tables

Table No	Table Description	Page No
2.1	Work done in the field of Hierarchical Algorithms …..…	16
4.1	Different Parameters for BIRCH input……...	29
4.2	Different Parameter for Traditional CURE input……...	31
5.1	Configuration of the single machine……...	38
5.2	Configuration of the machines for dedicated Hadoop……	39
5.3	Parameters for CURE algorithm in Hadoop……...	43

Chapter 1: Introduction

With the unpredictable development in internet technology and corporate sector, it is required to store an enormous amount of data in some advanced storage Technology and find some useful patterns out of it. Rapid development has occurred in different fields of human endeavor, starting from day-to-day (such as corporate transaction data, government statistics, credit card data, and commodity data) to the more exotic (such as molecular databases, astronomical images, geometric data and medical records). More interesting part is the possibility to extract useful information by using different concepts of data mining.

1.1 Data Mining

1.1.1 Introduction to Data Mining

To facilitate patterns among data, different types of data mining methods like classification, clustering, association, regression and pattern matching can be applied[1]. Data mining is an automated method of mining precious information and extracting information from large data repositories. It involves the process of analyzing data and finds some valuable information[2]. Different summaries and relations extracted through data mining are known as models or patterns like clusters, graphs linear, equations and time series. Data mining generally related to already collected data. Whatever data sets used in data mining are often large. Small datasets can be processed by using classical exploratory analysis but, it will create problems if it is large. There are many techniques exist to analyze useful information out of huge amount of data depending on the different requirement for the business problem that we are trying to solve.

Supervised and unsupervised learning are two main category of Data mining Techniques. In the first type of technique, the model is built before actual analysis is conducted using training data, then algorithms are applied on raw data to estimate patterns or parameters of the model. Classification, Association Rule, Neural Networks etc. are some examples of supervised learning. In second, model or hypothesis is not created before analysis. Algorithm is applied directly on the dataset

and result are obtained to create the model. Clustering of data points is one of the examples of unsupervised learning.

Data mining can also be described as knowledge discovery from data (KDD) as it deals with how to derive knowledge out of raw data[3].

1.1.2 Data Mining Process (Knowledge discovery process)

Steps of Knowledge Discovery Process as shown in Figure 1.1 are explained below:

1. Data cleaning:

It is an initial step to clean the row data. Smoothing noisy data, filling missing values and removing outliers are some of the operation that are performed in this step. If data is not consistent then it may cause a problem for mining procedure and the final results may be unreliable. Therefore, it is good to perform some of the data cleaning methods such as neglect the tuple, use global variables, manually fill value which are missing, use probable value, binning, regression and outlier analysis.

2. Data integration:

If data is coming from multiple sources for analysis purpose then multiple files, data sets or different types of databases need to be integrated. If redundant datasets exist then it may cause confusion in knowledge discovery process. So prior to mining process, proper integration of all relevant data is required to reduce redundancies and inconsistencies.

3. Data selection:

Sometimes it may not require an entire database for analysis purpose. instead, only some part of the database is sufficient to derive a result set by mining process, in that case, it is more reliable to reduce database size by selecting only important features. Data selection techniques can be used to obtain selected representation which is small in volume, yet data integrity is maintained. Different data mining techniques are dimensionality reduction, data compression and Numerosity reduction.

4. Data transformation

In this step, datasets are transformed into the forms which are suitable for mining process and outcomes are easier to understand. Different techniques for data transformation are Smoothing, aggregation.

5. Data mining:

This is a most critical step in the entire operation . Different types of data mining techniques are applied on large data repositories to extract useful information.

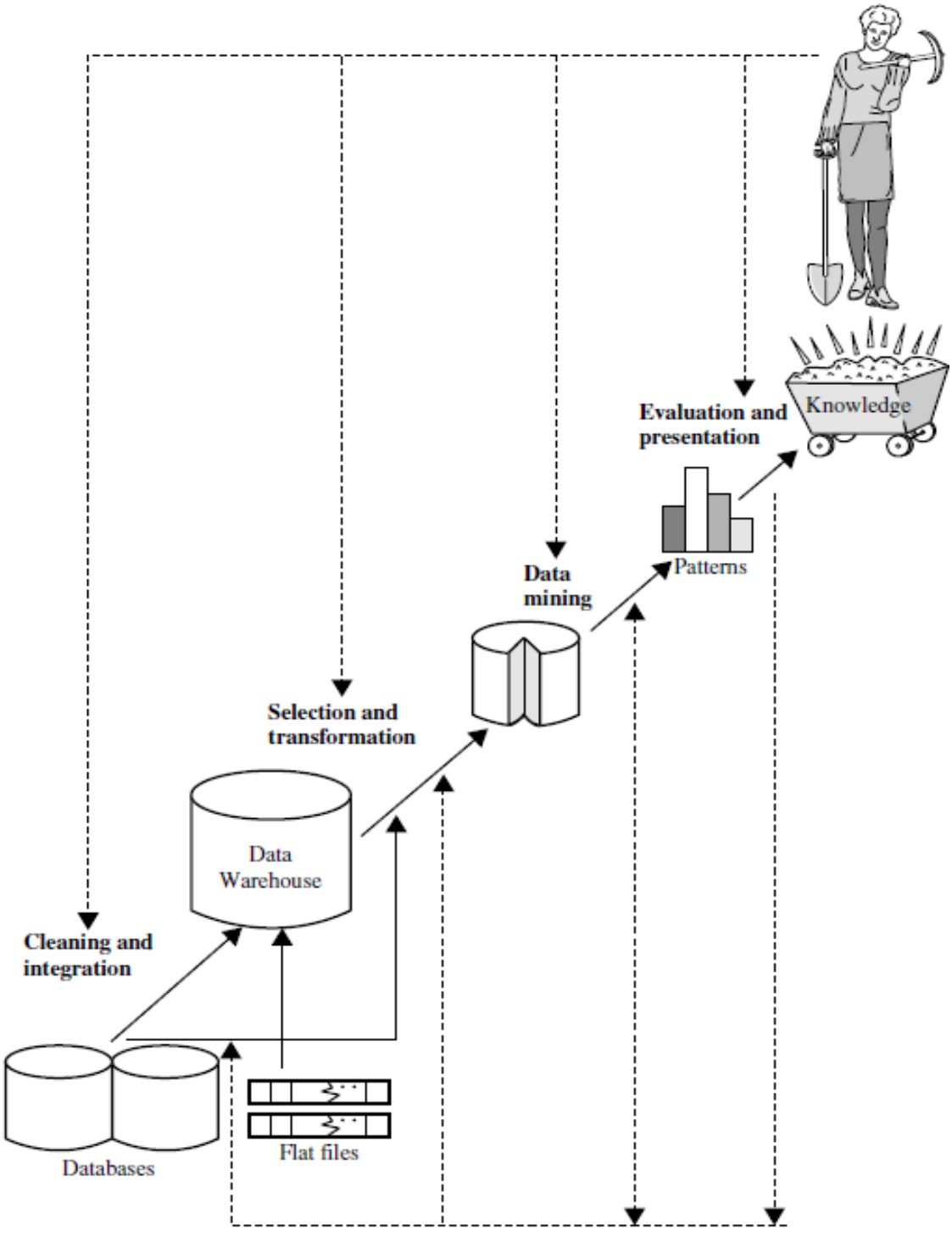

Figure.1.1 Data mining steps in KDD

6. Pattern evaluation:

Once the data is explored, identified and refined by applying data mining techniques next step is to identify interesting patterns. There are numbers of different types of patterns exist but an only small fraction of patterns are useful for given purpose.

7. Knowledge presentation

The final outcome of data mining process should be easy to understand and capable of representing useful patterns in the more specific way. For this different visualization and knowledge representation techniques are used.

1.1.3 Data Mining Techniques:

Classification:

Classification is one type of supervised learning technique. It divides the data samples into predefined groups also known as target classes. For each data point target class is predicted. For example, depending upon disease patterns a person can be classified as “low-risk” or “high-risk” patient. These classes should be defined based on attribute values. Classes are generally classified according to characteristics of the data that is known[4]. Training algorithms use these to determine the parameters for proper discrimination. Binary and multi-level are the two classification methods. In first, only two classes may be considered for example “high” or “low” risk patient. In second, more than two classes such as “low”, “medium” and “high” risk patient are considered. It has two types of datasets: Training datasets and testing datasets. Classifier algorithm is trained using training data sets and correctness of this algorithm is tested using the second dataset.

This section describes some of the classification techniques such as decision tree, K-Nearest Neighbor (K-NN), Bayesian Classification, Neural Networks.

1. Decision Tree:

   It is one type of flow chart in which every non-leaf node work as a decision maker on an attribute and each branch represents the result of decision and every leaf node represents a class. The node at top most position of the tree is

called root node.it can be easily converted to classification tree[5]. Decision tree induction algorithms are very useful for classification in various fields, such as Education, Manufacturing, Medicine[25], Financial analysis and geographical analysis. Using Decision Tree, best alternatives can be selected and while traversing from root to leaf, unique class separation can be achieved. There are many data mining algorithms such as ID3, NB tree, J48, C4.5 and REP tree.

2. Neural Network:

   It lies somewhere between the approximation algorithm and the artificial intelligence. It uses gradient descent method. It is basically biological nervous system with processing elements are known as neurons and useful to solve a specific problem. It is composed of nodes which are connected. Connections are weighed and training is performed by adjustment of these weights. Different types of rules are obtained from the trained neural network which is quite useful to improve interoperability of the learned network[6]. In Neural networks multiclass, multilayer feed forward technique is used in which they are taken in output layer instead of using one neuron. Decision tree and neural network representations are shown in Figure 1.2 and 1.3.

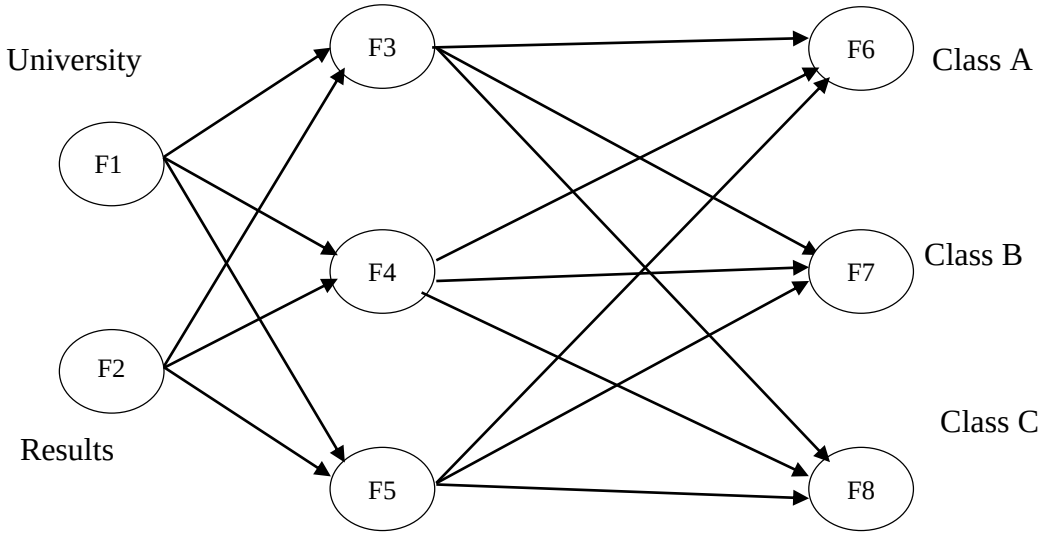

Figure 1.2 Neural Network

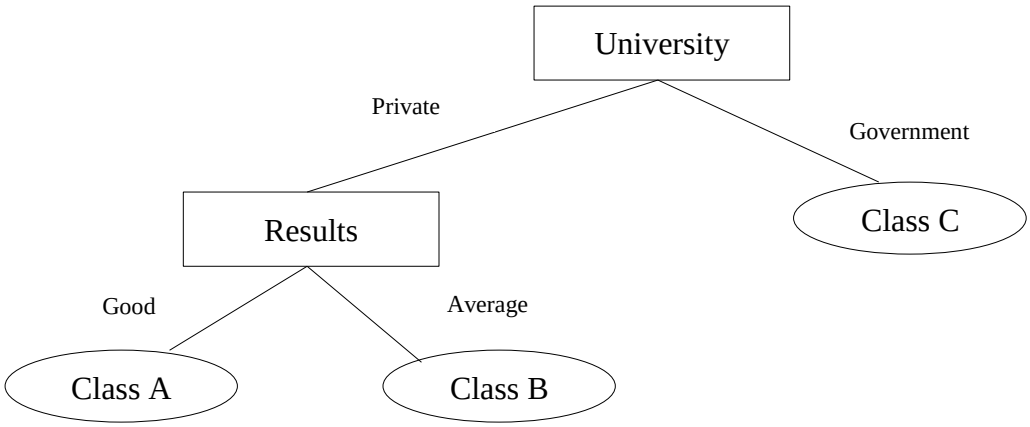

Figure 1.3 Decision Tree

3. Bayesian Classification

It is statistical classifier which can be used to predict class membership probabilities that are about tuple is belongs to given class or not. Bayes theorem is useful for Bayesian Belief Networks (BBN) and Naive Bayesian Classification. In Naïve Bayes Classifier assumes that all attributes are independent of each other. Bayes theorem concentrates discrete probability, posterior and prior distributions of data items.

Data tuple X is considered as 'evidence' and 'H' is hypothesis about X belongs to class C. P(H/X) is the probability that the H holds given the evidence or observe data tuple X. P(H/X)is the posterior probability of H conditioned on X. Suppose H is a hypothesis about the university gets A grade. Then P(H/X) is the probability that University X will get A grade. In contrast, P(H) is the prior probability of H. For example, the probability that given University will get A grade. The posterior probability P(H/X) is based on comparison of prior probability P(H) , and it is independent of X. Same way , P(X/H) is the probability of X conditioned on H. P(X), P(X/H) and P(H) may be estimated from given data. This theorem provides a useful way of calculating P (H/X) from P(X), P(X/H) and P(H). Bayes theorem is as follows:

$$P\left(\frac{H}{X}\right) = \frac{P(H)P\left(\frac{X}{H}\right)}{P(X)} \quad (1)$$

Various data structures used to represent it. Further, it discusses the benefits and applications of skip graphs.

Regression

It is used to explore different functions that can make relation among variables. Training dataset is used to construct a mathematical model. Statistically, two variables are used which are dependent and an independent variable represented using 'Y' and 'X'.

Regression method investigates the correlation between variables. Based on independent variables there are two types of regression: Linear and Non-linear. First, identifies the relation of dependent and one or more than one independent variables by using some linear function. Another type is non-linear regression which can accept categorical data. Binomial and multinomial regression are two types of logistic regression. First one predicts the output for a dependent variable when only two types of output are possible such as the person is either dead or alive. Another one is useful when more than two outcomes are there.

Association

Association is the discovery of patterns among objects in the database. It is also capable of discovering the association among sales in a transaction database. This type of analysis helps business to make future decisions and maintain supply chain scenarios such as cross marketing, customer behavior analysis and production capabilities. Algorithms of association need to be capable of generating rules with a confidence value, not more than one.

Types of association Rule:

- Multilevel
- Multidimensional
- Quantitative.

Clustering

Clustering is unsupervised learning technique in which data items are distributed into groups in such a way that similar data points lie together in one cluster[8]. These groups are not predefined so do not have any attribute with them. It distributes data points in such a way that it maximized the intra-cluster similarity and minimized the

inter-cluster similarity. Clustering of the data is a crucial problem in the various fields of computer science like machine learning, artificial intelligence, neural network and bioinformatics and is used it for data analysis. Base on different methodology it can be classified into the partition, density based, grid based and hierarchical[7].

1.2 Clustering

1.2.1 Stages of Clustering:

Steps of Clustering are shown in Figure 1.4.

1. Feature Selection:

   Basically, data samples are in collective form. To apply some clustering algorithm on them, a person has to choose specific parameter or features which can be processed further[9]. Feature selection chooses different types of features from a set of data points. Some sort of transformation is also performed to generate useful features from original row data. Strategically selection can reduce load on firther process and make the designing process easier. Unique features are those which are easy to extract and interpret and immune to noise.

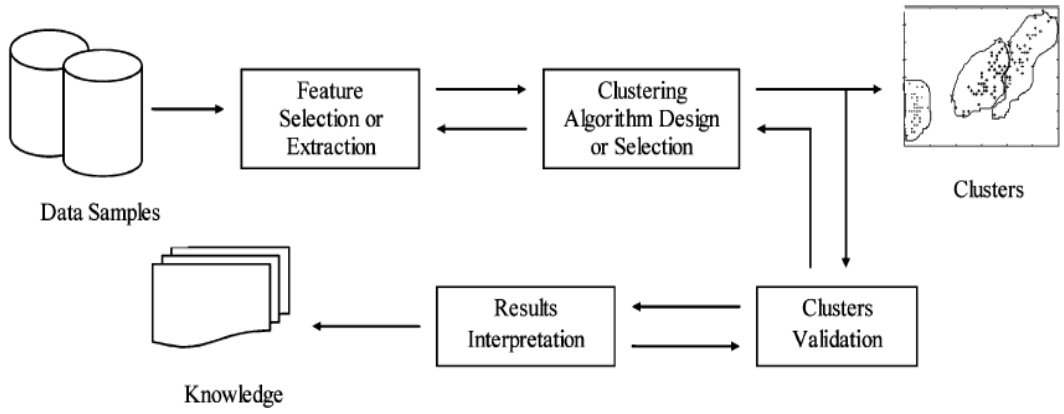

Figure 1.4 Stages of Clustering procedure

2. Clustering algorithm design or selection:

   Here, the making of a function and the selection of related proximity measure are done. Depends on whether patterns resemble or not they are grouped. Here proximity measure make changes in quality of the resultant cluster because all algorithms are connected with proximity measure. Once proximity measure is

taken, clustering criteria make this problem as an optimization problem. Many algorithms are there but no any specific algorithm can solve because no algorithms are universal for all type of problems. It is very tough to design a unified way for all types of problems. Therefore, careful investigation of characteristics of the problem results into good results.

3. Cluster validation:

   All clustering algorithms perform different processes and different approach gives different cluster and even if parameter identification or presentation order also matter. Therefore, the effective evaluation also important to ensure users about the quality of clustering by using given algorithms. During the evaluation process, different types of question are raised about clustering methodology. There are three types of testing: relative indices, external indices and internal indices. Relative place the importance on a comparison of structures, to decide about relative models, which one describe best characteristics of the object. External indices are related with pre-defined structures, which reflects prior information and are used as a validation standard for clustering solutions. Internal indices test based on internally only and are not dependent on external parameters.

4. Result interpretation:

   The main goal of clustering of given dataset is to cluster them and provide the user with patterns or analysis report from original data so that it can be useful for them to make a business decision and interpret data. Experts in their respective field analyze0 result with case studies for a guarantee of reliability of knowledge.

1.2.2 Clustering Algorithm Techniques:

Partitioned Clustering:

In partitioning , the dataset are partitioned into predefined number let's say 'k' clusters. This clustering can be based on any number of parameters. In partitioned clustering, we have to define a number of the clusters before clustering process. Partitioning method is of two: K-Means and K-Medoid. K-means method is one of the widely preferred approaches in which data points are clustered into k clusters such that in one cluster all data points have high similarity compared to another cluster. In

K-Medoid approach,1 k centroids are selected randomly and data points are assign to these on the nearest basis. During each iteration, on clustering mean is computed and data points are assigned to each newly created cluster. Clustering is done such that there is a high similarity in the cluster and high dissimilarity with another cluster. K-Means is used as beginning the process in many of the clustering problems. K-Medoid uses medoid instead of means in K-means for clustering. In K-medoid approach, initially arbitrary medoids are selected and data points are grouped into cluster depends on more similarity (lesser distance). The time complexity of partitioning algorithm is O(pcl), where p is the patterns, c is the number of clusters and l is number of iteration. Space complexity is O(c+p).

A major drawback is that this method does not suitable for data points which are closer to another cluster's center than to its own cluster's center for example when there is variation in the size of the cluster or shapes are convex.

Hierarchical Clustering:

It make grouping of the objects to make tree like structure. It is a continuous process of making groups of subsets of data points, with the condition that any two clusters are either disjoint from each other or nested. This method combines data points into clusters, then those clusters into big clusters, and thus create a hierarchy, which looks like tree structure called a dendrogram, in which root represent one cluster containing all data points and leaves represent individual observation.

The density of data set determines the relevant clusters. However once we performed the merging or splitting process, it will be very tough to reverse. Due to this, every decision needs to be created carefully because mistakes that are made during any stage are not possible to identify in the latter stage.

Basically, Hierarchical clustering algorithms is divided into two main types:

i. Agglomerative:

This is a bottom-up approach to cluster, in which all the data objects are singleton cluster at initial stage and it continues to merging all clusters based on similarity until they are combined into a single cluster. The output is represented by Tree graph. Then this one cluster is split gradually until

a number of clusters reached to the pre-defined number. Thus, it consist two steps: first is to determine exact number of clusters. Other is to split the selected cluster into new clusters.

ii. Divisive:

In this approach, as the name suggests, all data points are taken as a single cluster and then they are divided based on some dissimilarity metrics until required number of clusters is found, this is called as a top-down approach. Hierarchical clustering suffers from the one drawback that once split or more step is performed it can never be undone. It is also advantageous from the point of view that, only small computation costs are the only thing to taken into consideration and not worry about combined number of options. One way to improve the quality is to use hierarchical clustering with other clustering methods for multi-phase clustering.

Density-based Clustering:

It is connected dense component that continues to grow cluster until the density of the neighborhood becomes more than the threshold. Density based algorithms can discover not only cluster with spherical shape but also arbitrary shapes and also provide protection from outliers and noises. Data objects are divided depending on the connectivity, their region or boundary. In DBSCAN, one of the Density-based clustering algorithms, data points are either classified as a member of any cluster or as noise.

Data points are categorized as boundary points, core points or noise points.

- Core points: Points that exist inside the cluster are classified as core points. It is taken as a core point if the number of neighborhood points exceeds the threshold value.
- Border points: All those points that are not core and are located in the neighborhood of core are border points. During clustering process, a stage at which border points becomes core points for the newly generated cluster needs to be taken care.
- Noise Points: All those points which are neither border points nor core points.

DBSCAN, DENCLUE, OPTICS, DBCLASD, CHRONICLE and STDBSCAN are some of the Density-based algorithms that use to find outliers and perform clustering to generate clusters of different shapes.

Grid based clustering

Grid based clustering quantize the object space into a finite number of cells like structures which forms a grid structure. All clustering functions are performed on quantized space.. It can also be used along with other clustering algorithms STING (Statistical Information Grid) identifies statistical information lies in the grid cells. CLIQUE (An Apriori-like Subspace Clustering Method) combines density-based and grid-based for grouping in high-dimensional space.

1.3 Tools/Platforms/Languages used

A lot of computational problems can be solved by using a technique called Distributed Computing in which work is shared over interconnected nodes[33]. There are a number of frameworks are available for Distributed Computing out of which apache Hadoop[3] is one open source framework which is mainly used nowadays for the massive amount of data. Basically, Hadoop consists of distributed file system called HDFS and MapReduce. HDFS is storage component which is optimized to work fast & give high throughput [4]. Map and Reducer Programming paradigm fragment data on independently distributed node to be processed.

1.3.1 Hadoop Cluster Components

Hadoop cluster consists of components as follow:

JobTracker- It is a Master node which controls entire processing of Hadoop which includes scheduling the jobs, control submission, work assignment to the cluster components in Hadoop setup.

NameNode- Controller node for the file system of Hadoop called HDFS, it keeps information about which nodes consist which data along with given block is replicated at which nodes in the cluster.

TaskTracker- These all nodes are slave nodes and this are nodes where actual processing is done. Tasktracker controls spawning of Java virtual machines for each Map/Reduce.

DataNode: These nodes which actually act as HDFS and store data. Datanodes are also Task tracker. It exists on all slave machines.

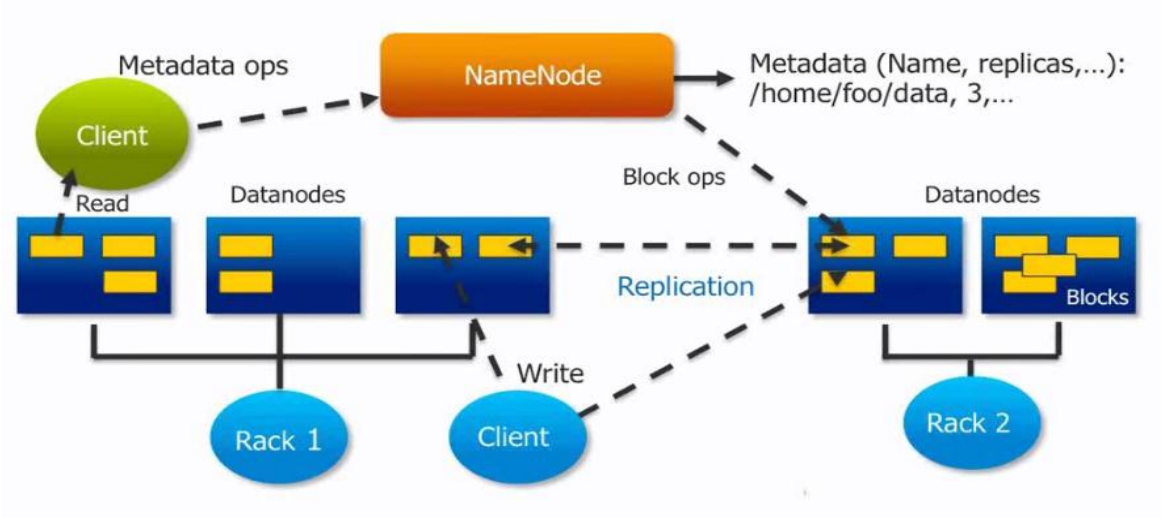

Figure1.5 Working of HDFS

1.3.2 The Hadoop Distributed File System (HDFS)

It is a File system which was originally created as Google File System (GFS). In HDFS, instead of storing data in a single entity, data or files are stored in HDFS are in the form of blocks which are 128 MB in size. The main advantage of HDFS is that each lock is replicated three times and each copy is stored on the different machine. This provides the advantage of file redundancy means if any machine fail than there will no impact on Hadoop functionality and data can be available from other replicated blocks.

1.3.3 Map/Reduce Programming Model

Map/Reduce is programming model based on Google's system used in 2004. Basically, Map/Reduce consist of two phases: (1) A Mapper and (2) A Reducer.

MapReduce is an important part of Hadoop framework for computing a large amount of data in less span of time[14]. All the inputs are given to Mapper in (key, value) pairs, Mapper perform some sort of user-defined processing on this data and the

output of it is intermediate result set which is again in (key, value) pairs. All the Mappers perform this action in parallel on different machines and intermediate result set generated from all are combined according to keys and all values associated with same key[16] are sent to same reducer[15]. Reducer performs final processing and gives results in the final output in the form of key/value pairs.

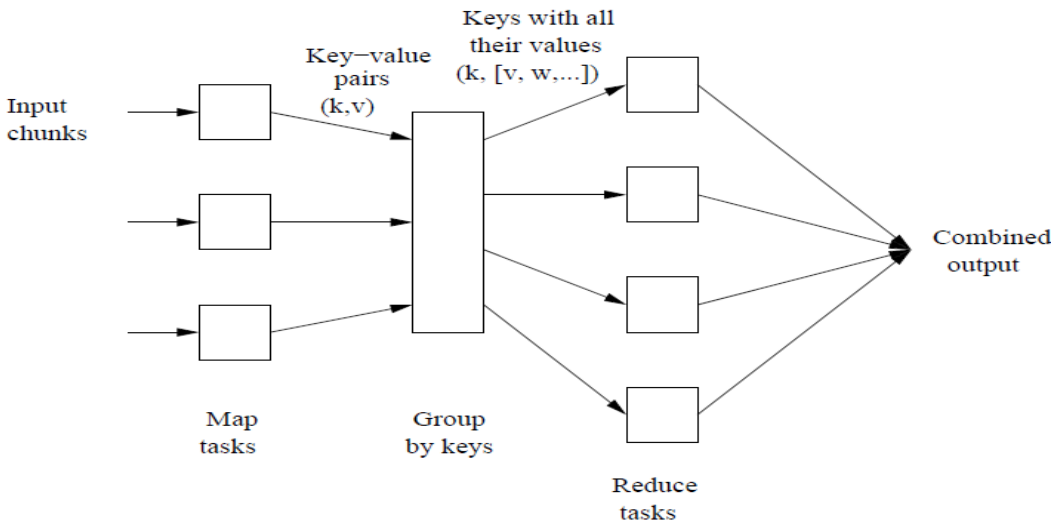

Figure 1.6 Map Reduce Workflow

Workflow of Map/Reduce is shown in Figure 1.6 Hierarchical Clustering of a large amount of data is a very crucial task and impossible to be done by single machine [5]. MapReduce provides great efficiency and flexibility to cluster data on distributed environment. Designing complex algorithms over Hadoop framework for unpredictable data is a challenging task in data mining[6]. The hierarchical algorithm considers each data set as a self-made cluster and after that iteratively these clusters are merged depends on some common features or patterns to form tree-shaped structure. One of the good characteristics of the hierarchical algorithm is each and every data points are disjoint in every cluster.

1.4 Structure of the Thesis

The rest of the thesis is organized in the following order:

- **Chapter 2 Literature Review** - It surveys the information about all the research work done Hierarchical Algorithm Field and describe the different hierarchical clustering algorithms

- **Chapter 3 Problem Statement -** It discusses the research gaps exist in current algorithm and the resulting problem statement. Further, the objectives of the problem statement and the methodology used to solve it have been discussed.
- **Chapter 4 Design of Parallel CURE algorithm -** In the first few pages, Implementation of BIRCH, CURE, CHAMELEON in Java has been discussed along with result set and after that Proposed CURE algorithm in Hadoop environment has been explained.
- **Chapter 5 Implementation Details –** In this chapter Implementation environment along with Hadoop setup has been explained and then Results of the different dataset and its analysis has been described with screenshots.
- **Chapter 6 Conclusion and Future Scope -** This chapter conclude whatever work is done in Thesis, its Contribution and future scope of this work.

Finally, references and list of publications have been appended.

Chapter 2 Literature Review

In data mining ,There are a lot of work has been done in the field of classification, regression analysis, time series analysis, anomaly detection, document analysis, text mining, predictive modeling, and semantic analysis. To understand the correlativity among the complex & huge amount of data, there are many clustering algorithms are developed and optimized by many data researchers as shown in Figure 2.1.

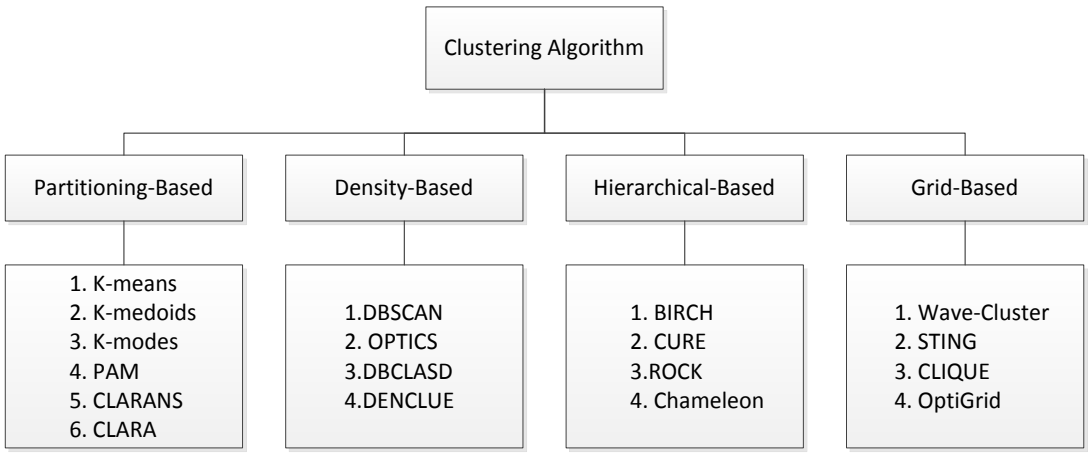

Figure 2.1 Classification of Clustering Algorithm

Out of all types of the clustering algorithm, most of Hierarchical algorithms are developed and optimized by different researchers as shown in the figure. All algorithms primarily differ from the point of view of how they update similarity between existing clusters. Some well-known algorithms are BIRCH[8], CURE[11], ROCK[40], CHAMELEON[10] etc. in all these algorithms different parameters are identified to improve the quality of resultant clusters and reduce outliers. This all hierarchical algorithms are discussed in this chapter.

Table 2.1 Work done in the field of Hierarchical Algorithms

Author	Work done in the field Hierarchical Algorithms
Zhao and Karypis [20]	Work done to explore document type data
Salvador and Chan [21]	Methods to Determine clusters in hierarchical algorithms.
Laan and Pollard. [22]	A new hybrid hierarchical clustering algorithm with the bootstrap and visualization

Mingoti and Lima [26]	Comparison of SOM neural network.
Shepitsen et al. [27]	Ways to Recommend in tag system in social media systems.
Koga et al. [28]	How to use Locality-Sensitive Hashing concept in Hierarchical algorithms
Abbas. [29]	Comparative analysis of different Data Clustering Algorithms
Jain [20]	Data clustering concepts
Murthy et al. [31]	Content based image retrieval
Horng, M. Su, Y. Chen, T.Kao [23]	Studied interuption recognize system using support vector machines
Kou and Lou [32]	Effect of parameters in large scale search engine data and web page.
A.K.,S.Balakrishnan, M Xu, A. Singh [43]	Studied different active algorithms.
Langfelder and Horvath [34]	Using of R functions in hierarchical algorithm.
Malitsky et al. [24]	Work done in the field of cost-sensitive clustering of data points
Meila and Heckerman, [36]	Initialization methods and some clustering algorithms are compared
Müllner (2013) [37]	Fast cluster is done along with R and Python.
Balcan et al. (2014) [38]	Robust hierarchical clustering
Szilágyi and Szilágyi [39]	Clustering algorithms are studied for protein sequence data.

2.1 BIRCH Clustering Algorithm (Balanced Iterative Reducing and Clustering Using Hierarchies):

BIRCH is integrated agglomerative hierarchical clustering method basically designed for clustering numerical data by using along with other clustering algorithms. When there is only limited main memory available and wants to achieve a linear I/O time with only one database scan. In Birch cluster hierarchy is represented by clustering features for summarizing a cluster, and CF-tree (clustering feature tree).

A Clustering feature consist three parameters which affect clusters of given data points. It defined as

$$CF=(n, LS, SS) \quad (2)$$

Here n is a number of data points, LS is the linear sum of points and SS is the square sum of data points. A clustering Feature is one type of summary of given cluster. Using it, we can derive many parameters like,

$$x_0 = \frac{\sum_{i=1}^{n} xi}{n} = \frac{LS}{n} \quad (3)$$

$$R=\sqrt{\frac{\sum_{i=1}^{n}(xi-x0)^2}{n}} = \sqrt{\frac{nSS-2LS^2+nLS}{n^2}} \quad (4)$$

$$D= \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(xi-x0)^2}{n(n-1)}} = \sqrt{\frac{2nSS-2LS^2}{n(n-1)}} \quad (5)$$

R is average distance from the centroid to member object, D is the average pairwise distance within the cluster. For given two cluster C1 and C2, If they are merged then resultant cluster's clustering feature is $CF_{result}= CF_1 +CF_2= (n_1 +n_2, LS_1+LS_2, SS_1+ SS_2)$

CF tree is a height-balanced tree that contains clustering features. Non-leaf node has descendants and all non-leaf node store sums of CFs of their children, there are two parameters in CF tree: threshold T and branching factor B. Here B specifies a maximum number of children per non-leaf node. The maximum diameter of subcluster at leaf nodes are specified by threshold parameter.

BIRCH use multiphase clustering. The first scan of data items is for basic and good clustering after that one or more additional scans can be used for quality improvement. While clustering an object is inserted to the closest sub-cluster. Here leaf node represents a cluster made by merging all sub-clusters. All the data points in a leaf node have to satisfy the threshold requirements compared the threshold value T, that is, the diameter of the sub-cluster must be less than T. If the diameter is larger than T, leaf node and other nodes are split. So the size of CF tree can be modified by changing T.

Phases of BIRCH clustering:

In **Phase 1** dataset is scanned and initial CF tree is built using available memory.CF tree reflects the clustering information as fine as possible. When data points are grouped, crowded data points are considered as good sub-clusters. And sparse data points are removed by considering outliers.

Phase 2 is optional as shown in Figure2.2. Whatever we obtain as the outcome of Phase 1 is CF tree having a larger size but in phase 3, global clustering applied on clusters having different size and lack and some extend if the output of phase 1 is given as input to global clustering. In phase 2 it scans all initial CF tree to build smaller CF tree and remove outliers.

In **Phase 3** global clustering is applied to all CF trees & we obtain different cluster that represents data points located in grouping manner. However there are some inaccuracies might exist.

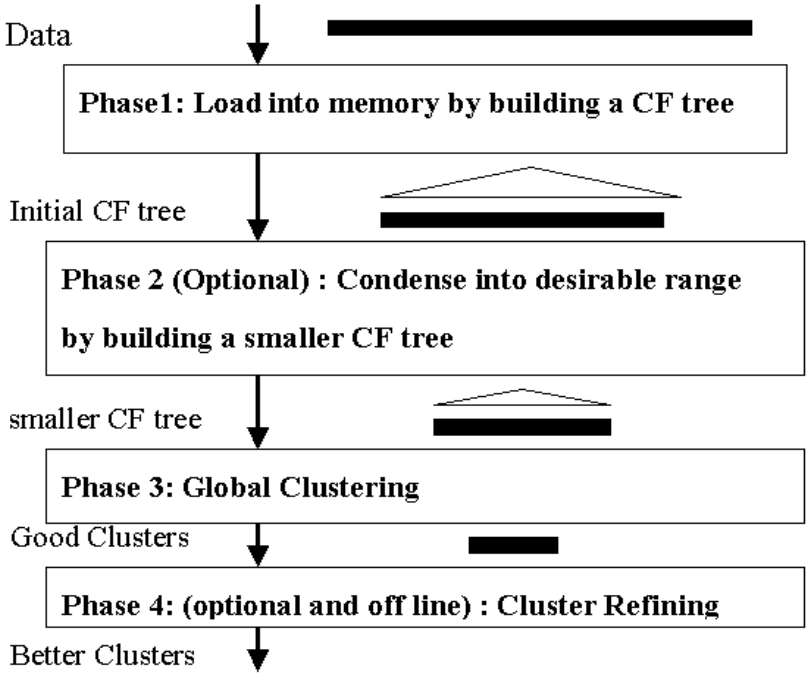

Figure 2.2 Overview of BIRCH Algorithm

Phase 4 is optional and the main purpose of it is to correct those inaccuracies. It uses centroids produced in the third phase and perform redistribution of data items to form new clusters.

The time complexity of BIRCH algorithm is O(n). Where n is a number of data points in given dataset. So it shows linear scalability of the algorithm with data points. If clusters are non-spherical shape[19], it can't perform well.

2.2 CURE Clustering Algorithm (Clustering Using REpresentatives)

In Hierarchical algorithm, There are two approaches for representing clusters, One is, centroid or medoid based approach in which centroid or medoid represent the center of the cluster and similarity between clusters are decided based on that. But it can't identify clusters of non-spherical shape. Other is using multiples scattered points to represent cluster which enables it to identify clusters of arbitrary shape and any sizes[12]. However, it is much sensitive to noises, outliers.

To overcome the above drawbacks and to use good facts of both these methods, researchers have proposed CURE (Clustering Using REpresentative points) algorithm. In Cure, Each Cluster is represented by fixed number of well-scattered points called representative points from given cluster[13]. Shrinking these points by the fraction of Shrinking Factor towards the center of the cluster enable them to be free from outliers. Due to this, not only it's possible to determine a cluster of arbitrary shape but the effect of outliers and noise can also be reduced at max level. Data points having maximum influence on given density in the class are selected as representative points so that they can reflect all data sets presence without loss of effectiveness. To improve the outcome of CURE algorithm, there are two main methods are used: random sampling and then segmentation.

CURE clustering algorithm differs from BIRCH algorithm in two ways. First, Unlike BIRCH algorithm, CURE algorithm does not use pre-clustering of all data items. To minimize the number of data points to represent a cluster, random sampling is done, which minimize the size of data set in an effective way. Although sampling takes only some data points out of entire data set, it will not impact on the efficiency of clustering. CURE use Chernoff bounds for calculating minimum sample size.

Second, for speed up entire sample data set is partitioned into a predefined number of partitions. For a given partition, after applying partial clustering number of sub-clusters is generated depends on representative points. Then due to multiple representative points, outliers are easy to identify. At the final stage, Global clustering

is applied on sub-cluster in which sub-clusters merge with each other by clustering procedure to build pre-defined number of clusters. K-d tree and heaps are used in CURE. For each cluster, information about a number of points, the mean of points , representative points and other information are stored as a unique entry in the heap.

Steps of CURE algorithm

Steps of CURE algorithm as shown in Figure.2.3 are:

1. Scan the entire database of size N data points and select a random sample that can fit in main memory.
2. Partition the given dataset into p partitions that will result in each partition of size n/p.
3. Apply partial partitioning on each partition and generate sub-clusters by clustering procedure. Here distance between clusters is represented by distance between pair of representative points belongs to their respective sub-clusters.
4. Eliminate Outliers. Due to shrinking representative points by shrinking factor α it is easy to identify outliers.
5. Apply Global Clustering on all the subclusters & Merge them till predefined number of cluster is constructed. Now all clusters and outliers are labeled to identify its shape.

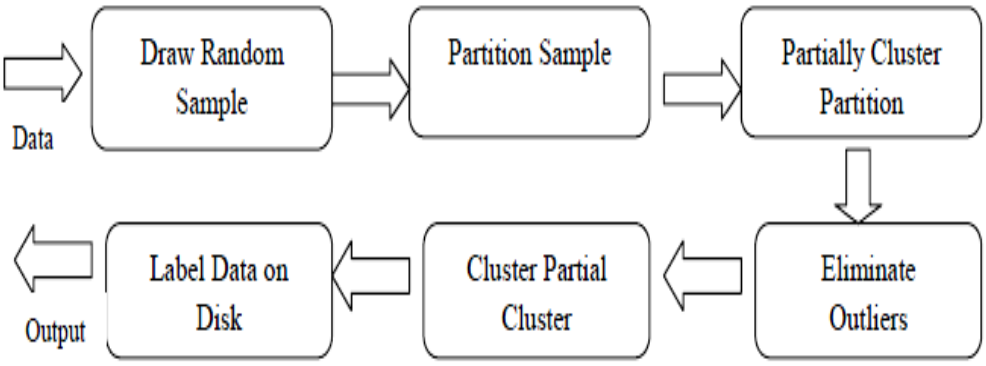

Figure 2.3 Overview of CURE Algorithm

2.3 ROCK Clustering Algorithm (Robust Clustering Using Links)

ROCK is hierarchical agglomerative algorithm. ROCK method is different from other hierarchical methods. When data objects are represented on plot then there can be link imagined among points. ROCK make clustering based on this links and Jaccard

coefficient for clustering process. Categorical and Boolean attributes are generally given as a parameter to this algorithm. A number of points from a different cluster having same neighbors are used to describe similarity of the data points in their own clusters.

Goodness measure is determined based on the criterion function:

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \quad (6)$$

Here, link[C_i,C_j] is a number of common neighbor between C_i and C_j.

At every step, the pairs of clusters are merged that maximize this function. Steps in ROCK algorithm are drawn a random sample from datasets, cluster random links and label data on disk.

Once random data points are taken from a database, links are applied to this points. At last only those points which are there in random sample are take into consider for assigning rest of the obects to the cluster. This merging is performed continuously until a threshold number of cluster or common links among clusters reduced to zero. Overview of the process is shown in Figure 2.4. The advantage of ROCK algorithm compare to the traditional algorithm is that it give better scalability. Figure 2.4 shows ROCK algorithm in a graphical manner.

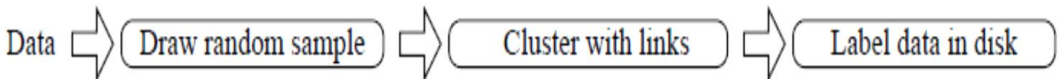

Figure 2.4 Overview of ROCK Algorithm

- **Limitation of Static Hierarchical Schemes:**

  The main drawback of above hierarchical schemes is that merging decisions are taken based upon statical modeling. Consider four sub-clusters shown in figure 2.5. Statical modeling of CURE use to do merging of (a) and (b) than merging (c) and (d), because minimum distance between representative points in (a) and (b) is less than distances in (c) and (d). But logically (c) and (d) are best choice to make merging as minimum distances between the

boundary points of (c) and (d) are of the same order as the average of the minimum distances between any points within these clusters to other points.

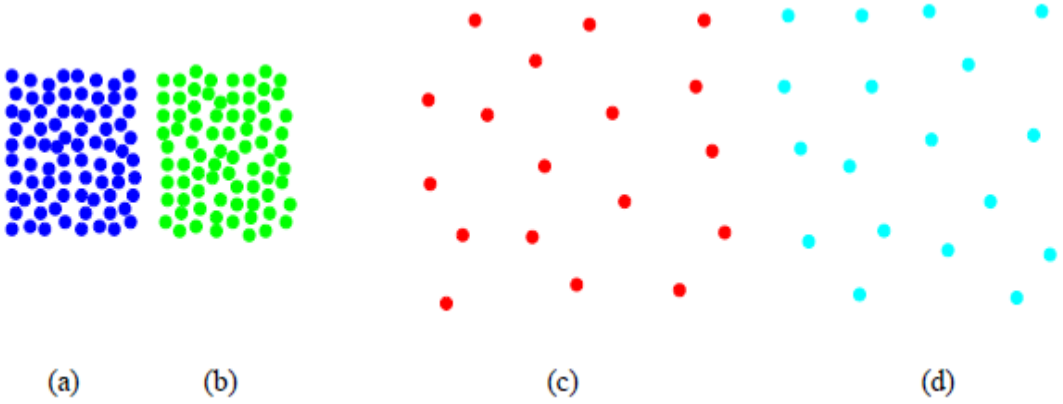

Figure 2.5 Merging choices for CURE

In ROCK, Connectivity between two clusters is measured by considering the expected connectivity. However, they assume user supplied inter-connectivity model, which can easily make the wrong decision. For example, as shown in figure 2.6, each cluster is represented by the sparse graph. If all the edges in the graph have equal weight then ROCK mechanism will merge (c) and (d). But logically (a) and (b) pair is more preferable.

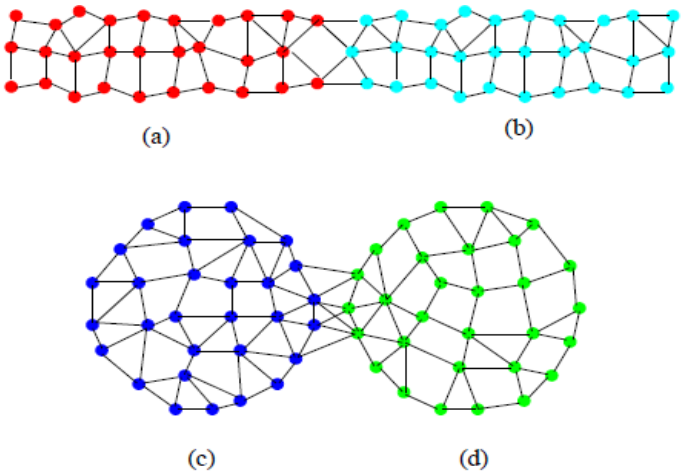

Figure 2.6 Merging choices for ROCK

Thus, CURE and its related methods ignore aggregate connectivity whereas ROCK and its related methods ignore closeness of clusters while taking a merging decision.

2.4 CHAMELEON Clustering algorithm: (Clustering Using Dynamic Modelling)

It is a hierarchical clustering algorithm that takes the use of dynamic modeling to describe similarity in clusters. Cluster similarity is evaluated based on,

a. Proximity of clusters
b. How objects in same cluster are related with each other..

In this way at any point of time two clusters are combined with each other if interconnectivity between them is at max value. So CHAMELEON is dynamic, user-supplied model and can take to merged cluster's characteristics. It performs its operation on the sparse graph in which data items are represented by nodes and similarity among them is represented by weighted edges. This sparse representation allows it to use for large datasets and to apply on data items which are available in similarity space.

As shown in Figure 2.7 it uses k-nearest- neighbor graph approach. It uses graph partitioning algorithm for partitioning this k graphs into small sub-clusters by minimizing edge cut. Cluster C is partitioned to C_i and C_j such that it minimize weight by removing edges that connect C_i and C_j.

The agglomerative hierarchical algorithm is applied on this sub-clusters to merge them on the basis of similarity. Relative interconnectivity and closeness are taken into consideration while merging.

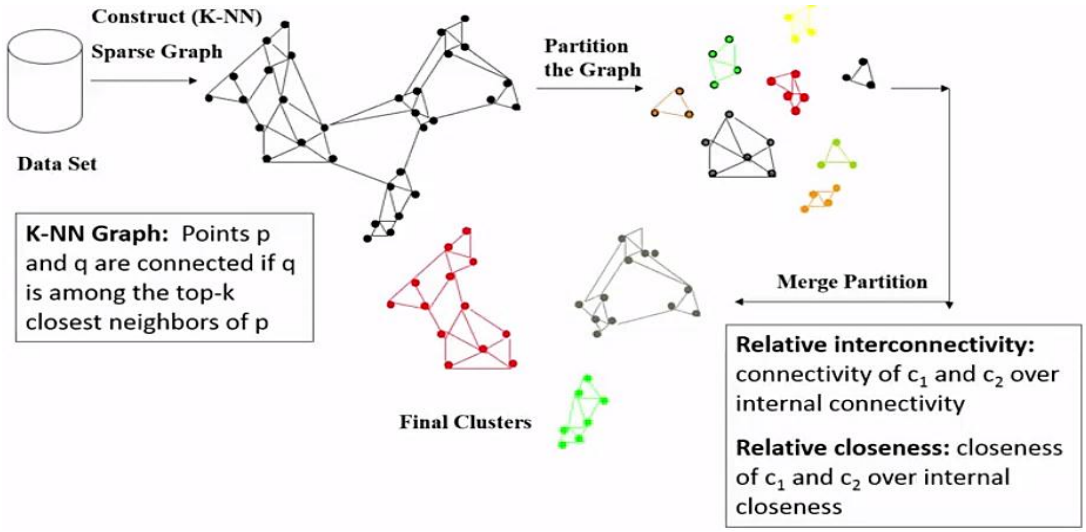

Figure 2.7 Overview of CHAMELEON Algorithm

Relative interconnectivity is defined as the absolute inter-connectivity between *Ci* and *Cj* normalized in prospective of the internal inter-connectivity of the two clusters *Ci* and *Cj*.

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}}, \tag{7}$$

Where $EC_{\{C_i, C_j\}}$ is edge cut and EC_{C_i} (or EC_{C_j}) is a minimum sum that partition it into two equal parts.

Relative closeness is the absolute closeness between C_i and C_j normalize with the internal closeness of C_i and C_j.

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}}, \tag{8}$$

Here S_{ECCi} and S_{ECCj} are the average weights of the edges that belong in the min-cut bisector of clusters *Ci* and *Cj* ,respectively, and $S_{EC\{Ci, Cj\}}$ is the average weight of the edges that connect vertices in *Ci* to vertices in *Cj*.

Thus, CHAMELEON has a higher capability to identify arbitrary shaped clusters with the removal of outliers at maximum level compare to BIRCH and DBSCAN.

Chapter 3 Problem Statement

3.1 Research Gaps

In this era of rapid development, There is a drastically growth in data generated by different fields of internet technology, social media and corporate world. So it is a basic requirement to store this data on some reliable storage medium and process data in less amount of time using latest available technology. Nowadays there are many data structures or tools are used to manage this huge data. Processing of data in distributed environment is one area in which lots of work has been done. For a given data objects, it may need to cluster them according to some similarity function. There are many hierarchical clustering techniques available to cluster data traditionally. However, they are not reliable and fast in nature to process large amount of data

With the development in technology related to distributed environment, many companies rely on Hadoop-based MapReduce Technology. For clustering of the data objects, it may convenient to migrate from traditional single machine approach to distributed environment. so there is major research gap is found for a massive amount of data as for a high number of objects traditional clustering algorithm fails to perform in a convenient way.

3.2 Problem Statement

By considering the research gaps stated above, the problem statement is to design a new way to execute CURE Hierarchical Algorithm to improve the efficiency of clustering process and implement it in distributed environment. So when Clustering is done in Hadoop environment using MapReduce programming paradigm, it reduces processing time and store data in a reliable way. There are some changes needs to be done in clustering methodology keeping in mind that these will not affect the basic concept of CURE method to cluster. Furthermore, it is also required to analyze patterns of outliers depends on different parameters and remove them most efficient way

3.3 Objectives

By looking at Research gaps and Problem need to be solved, objective of the thesis work is:

- To study, comparatively analyze and explore existing clustering techniques and look at their outcomes.
- To propose a new way to implement CURE algorithm with MapReduce concept for data objects
- To test and validate this proposed algorithm in Hadoop environment and analyze the result set for large amount of data objects
- Analyze the outlier's patterns and compare most efficient parameters for CURE to be set to remove outliers in an efficient way.

3.4 Research Methodology

- Migration from Traditional CURE clustering algorithm to CURE with parallel processing in distributed environment is the focus here. Traditional hierarchical algorithms are implemented in JAVA. The proposed algorithm is implemented in Latest version of Hadoop environment using making Mapper-reducer for that. Resultant data points are shown in GNUPlot in the form of scattered graph.

Chapter 4 Design of Parallel CURE algorithm

4.1 Existing Hierarchical Clustering Algorithms

As discussed in the literature review, in the field of data mining there are a lot of clustering algorithms are developed. Hierarchical algorithms are useful for clustering of data points in a tree structure. BIRCH, CURE, CHAMELEON are some of the hierarchical clustering algorithms. Here I have discussed some of this algorithm with its implementation.

4.1.1 **Implementation of BIRCH algorithm**

Birch hierarchical algorithm cluster the data points with the concept of cluster features and CF-trees. Other important parameters for BIRCH are threshold T and branching factor B.

Input: n number of data points to be a cluster.

Output: k clusters

Algorithm:

1. Scan the given dataset and store in memory
2. Initialize CF Tree as Empty {{}}
3. For a given Threshold value construct CF Tree out of data points or old CF-Tree and calculate number of CF-Entries, number of CF-Leaf Entries And ∑SS for CF Tree.
4. Compute new Threshold value from distance Function & given old The threshold value for good CF-Tree and removing outliers.
5. Repeat step 3 and 4 until good quality CF Tree are obtained.
6. Perform Global Clustering on all CF Trees and Construct clusters.
7. Refine Clusters.

Algorithm 4.1 BIRCH Hierarchical Algorithm in JAVA

For a given algorithm 4.1, data inputs are maximum number of data points possible in given Clustering, Initial Distance threshold, centroid distance function, margin

refinement needs to be applied, memory limit for given condition. This all parameters are given in Table 4.1.

Input:

Table 4.1 Different Parameters for BIRCH input

Parameter	Value
maximum number of data points	100
Initial Distance threshold(T)	0.5
centroid distance function	1
margin refinement	True
memory limit	100 MB
Number of data points	59
iteration	30

Output:

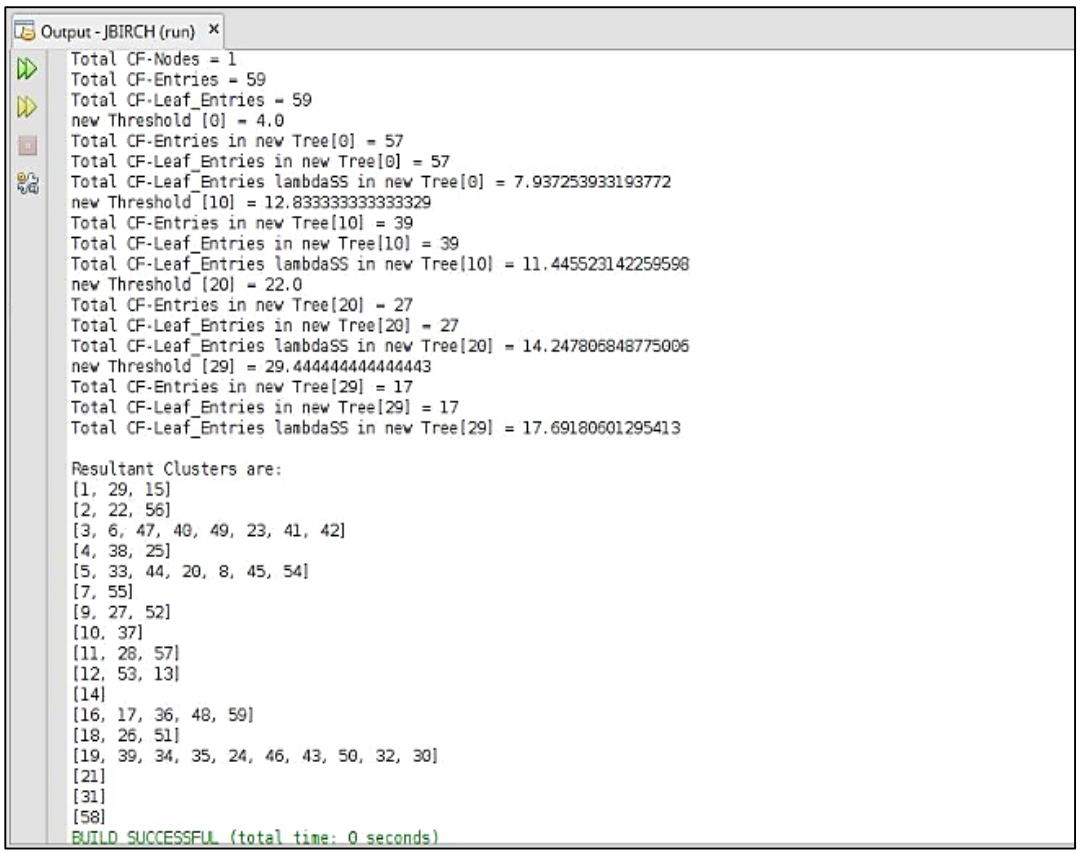

Figure 4.1 Output of BIRCH Algorithm with clustered data points

Here as shown in Figure 4.1, we have iterate Finning process of CF-Tree 30 times. Here in the output value of different parameters are shown for only some value (0, 10, 20, and 29) of Threshold. Resultant Clusters are those having two or more data points. Data points having a single point is considered as outliers. This all points make clustered form structure.

4.1.2 Implementation of CURE algorithm

CURE(data points, k)

Input: A set of data points S.

Output: k clusters.

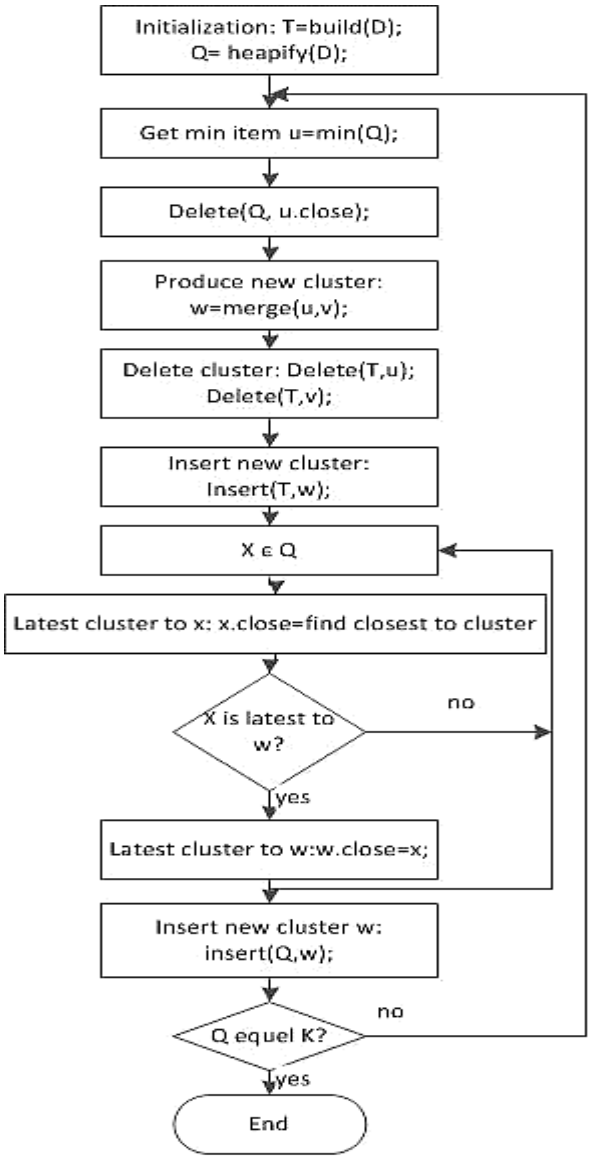

Figure 4.2 Clustering Procedure in Traditional CURE Algorithm

CURE algorithm uses middle way between centroid based and approach based on multiple scattered points by using more than one representative points. For a given data set, all Steps are described in Literature survey. Detailed Clustering procedure is given in Figure 4.2.

CURE Algorithm is implemented in JAVA using eclipse. Different parameters for given data sets are given in below Table 4.2.

Table 4.2 Different Parameter for Traditional CURE input

	Explanation	Dataset1 (space)	Dataset2(Europe outlets)
No of data points (n)	Total number of points	59	788
Sample size(s)	Sample size(based on Chertoff bounds)	≥ 2.5 % of datasets	≥ 2.5 % of datasets
Shrink factor (α)	Shrinking factor towards centroid or mean	0.5	0.5
Number of Cluster (K)	Predefined number of clusters	5	5
Representative Points	No of Representative points from cluster	6	6
Number of Partition	Total number of partition	2	2
Reducing Factor for each Partition	It use to define outlier removal strategy	3	3
Required Representation Probability	Decide samples to be selected for clustering	0.1	0.1

Dataset1 is about particular points from space and dataset2 is about Europe outlets of the company. Here we have given output for data set1 in below GNUplot. Here points in different color and shape represent different cluster in scattered plot. We have taken reducing factor as 3 which is quite enough to identify outlier points which are located at some distance and form separate cluster so they are removed from clustering process and not shown in scattered graph shown in Figure 4.3.

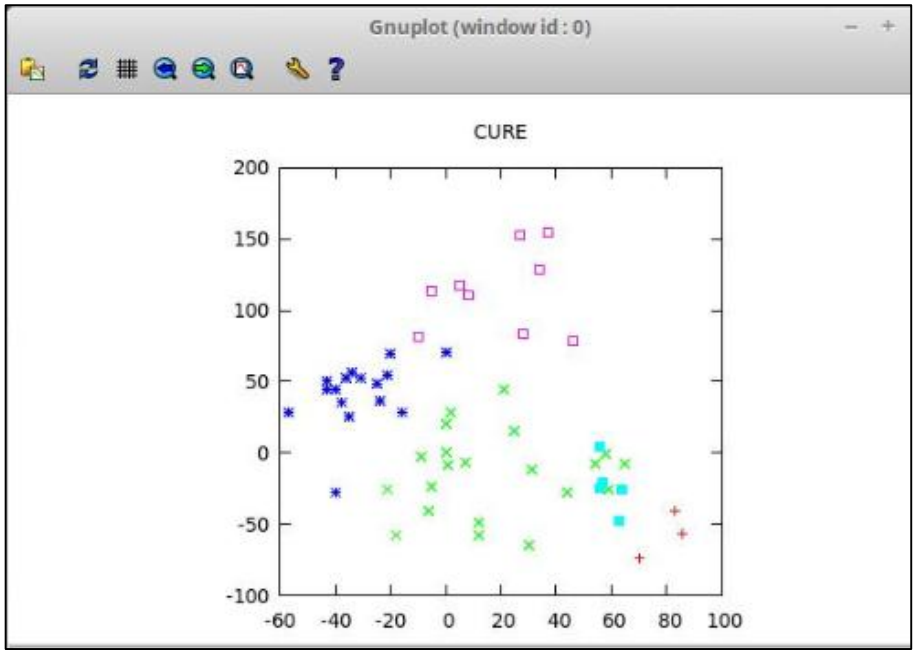

Figure 4.3 Output clusters of Dataset1 (space)

4.1.3 Implementation of CHAMELEON algorithm

CHAMELEON algorithm uses dynamic approach for clustering.in which given clusters are merged only if relative interconnectivity is high. Algorithm for it is given in Algorithm 4.2.

CHAMELEON (data points, k)

Input: A set of data points S.

Output: k clusters

Algorithm

1. Scan the database and set indexes
2. Check capability of clusterer, whether it can handle given data or not.
3. Filter data points to check if data points are missing or not
4. Merge instances to nearest neighbor and construct sparse graph Using K-NN graph approach
5. Partition the sparse graph into small sub-clusters such that it minimize edge cut.
6. Sub-clusters are combined using Agglomerative hierarchical algorithms with the help of relative closeness and interconnectivity.

Algorithm 4.2 CHAMELEON Hierarchical Algorithm in JAVA

To implement the CHAMELEON algorithm in java, I have used some APIs of WEKA for some of the operations like capability check, clusterer and others. Dataset1 having size 59 points are clustered to form 5 clusters.

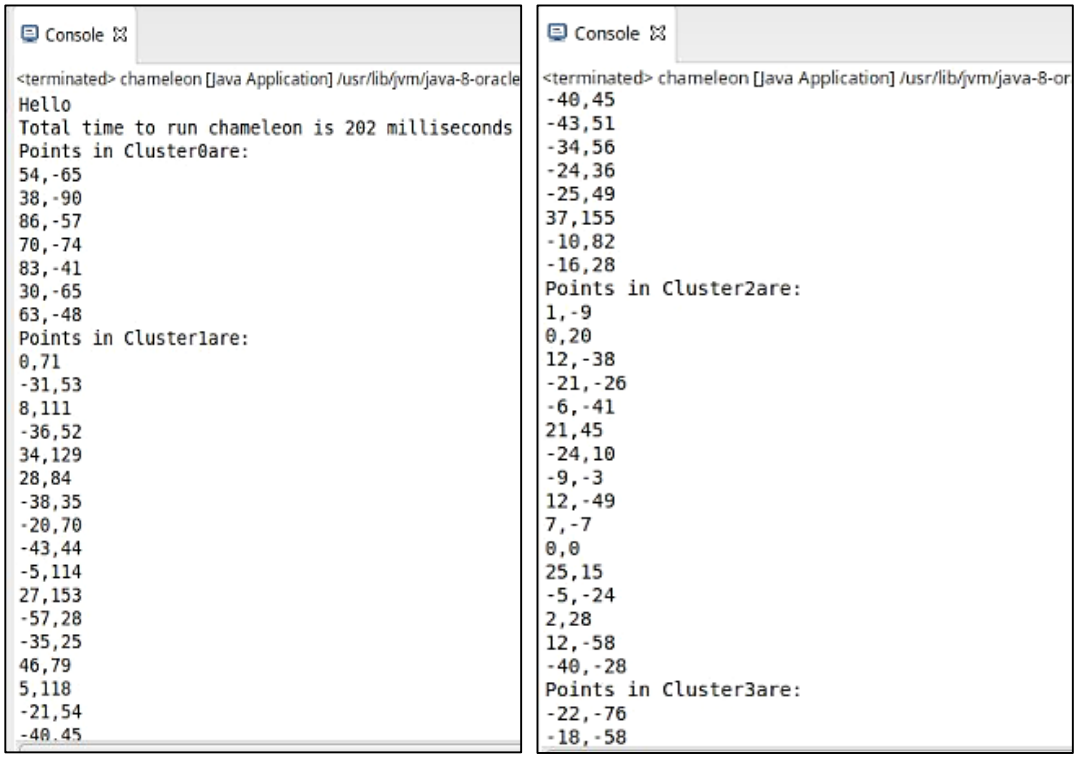

Figure 4.4 (a), (b) Clustering of data points in CHAMELEON

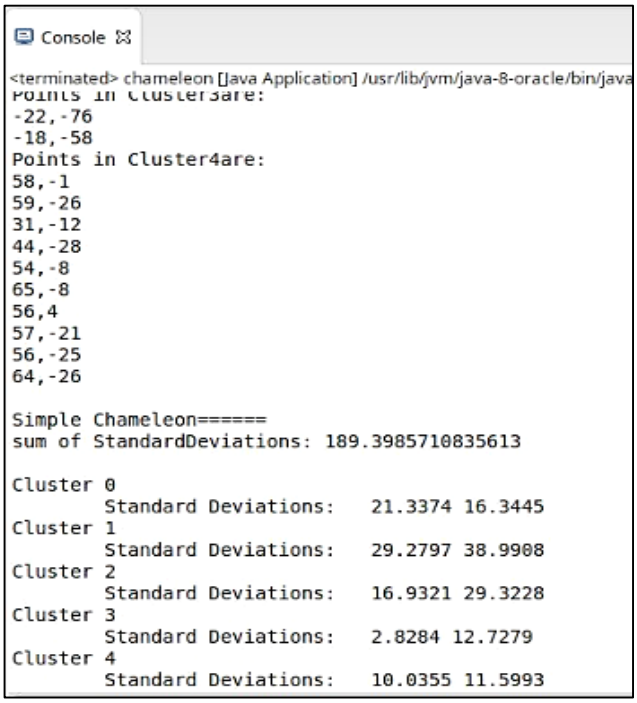

Figure 4.4 (c) Clustering in CHAMELEON

Execution time and Different data points in all five clusters are shown in Figure 4.4(a),(b) and (c). Here data points having the same cluster are decided dynamically so it is more reliable approach than CURE and BIRCH. This all five Clusters are plotted on GNUplot as shown in below Figure 4.5.

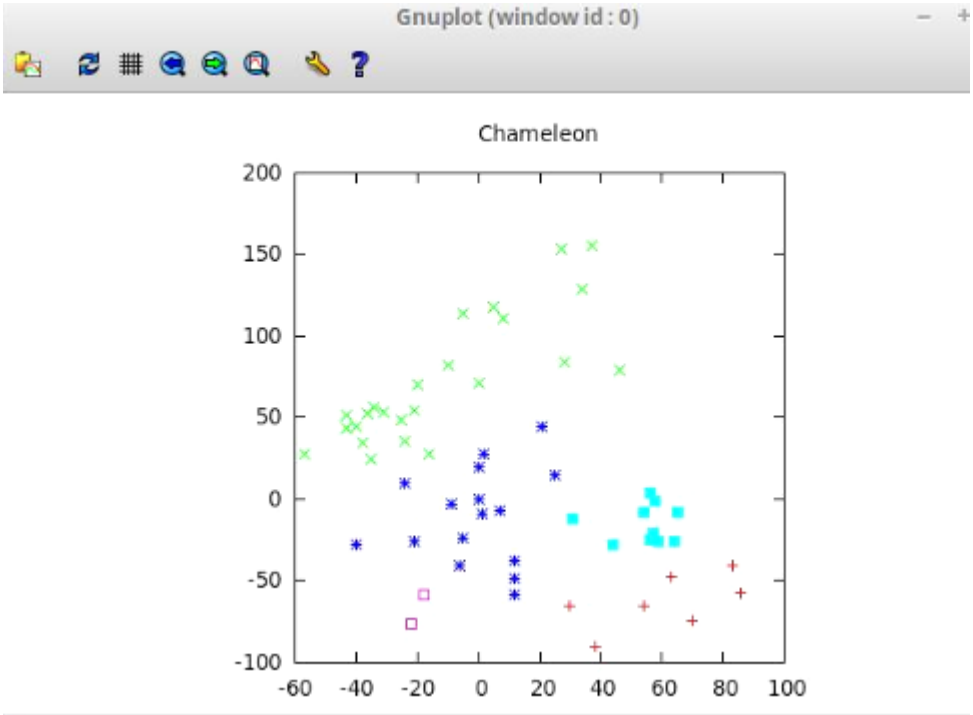

Figure 4.5 Output of CHAMELEON in GNUPlot

4.2 Design of Proposed Algorithm

We have discussed Performance of Traditional CURE algorithm in a simple way by implementing it on Java platform for given set of dataset1 and dataset2 having a small amount of data. However in the Real world, there are so many areas which contain a large amount of data need to cluster them according to some feature are a similarity. For example, Data generated from geo-stations located in space are huge in amount and it may generate TBs of data per day. So we need some efficient method to store this all data at reliable data store which remain available all the time and more importantly we need some efficient way to analyze this data and find useful patterns out of it. For example, this data may need to cluster according to its distribution on given area of the earth. This may in the form of (x, y) coordinates or with some other information.

If we perform mining clustering algorithm on this large amount of data then it will lack in performance or may be unable to handle it on single java platform. So Traditional CURE algorithm in incapable to cluster huge amount of data. The performance of CURE algorithm can be improved by implementing it on Hadoop environment as execution time to cluster given dataset reduce if it is done in parallel systems. Traditional CURE algorithm we scan the entire database in a single machine and then apply random sampling. After that further partitioning process is done. In this all process we process only data in KBs.

However for large data set it is tough to scan entire database from the local machine and cluster it accordingly. Instead of this if this work is done by MapReduce method on more than one machine (mapper) than it will take less execution time and cluster dataset in more efficient way.

Steps Of proposed CURE Clustering Algorithm:

Overall steps which are followed in improved CURE algorithm are:

1. To implement CURE algorithm, prerequisites is to have a list of datasets which is to be a cluster in (x, y) coordinate form in a text file. Initially text file having a set of data points is stored in the HDFS (Hadoop Distributed File System). For that, we configure Driver file of MapReduce and job is created which is to be submitted to all mappers configured as part of MapReduce.
2. Data points are read from Text file in the form of (key, value) pair and feed to mapper by using map function. Here index of each data point is considered as key and data points, which are in the form of (x, y) coordinate are taken as a value in mappers.

   However, coordinates x and y are of double data type so not suitable to be used in distributed environment so it must be converted to writable form. Data in sequential form are easy to pass from mapper to reducer. The output of mapper is also in the form of (key, value) pair where key are longwritable and values are data points which are in the form of DoubleArrayWritable.
3. The output of Map function is taken by reducer in the form (longwritable, DoubleArrayWritable). Reducer takes a random sample out of it and partition

data set into p partition and apply partial partitioning on each partition as per traditional CURE algorithm.

4. Now there are a number of sub-clusters are there and each cluster may have a different number of data points depends on Euclidean distance. Another advantage of this algorithm is to have the precise technique to detect outliers. If the cluster is growing slowly are considered as outliers. In this paper, sub-cluster having only 1 data point is identified as an outlier and removed from the clustering process.
5. Finally, All the cluster are combined in reducer function until data points reduces to specified the number of clusters.

This entire process is shown in Figure 4.6 in diagram form.

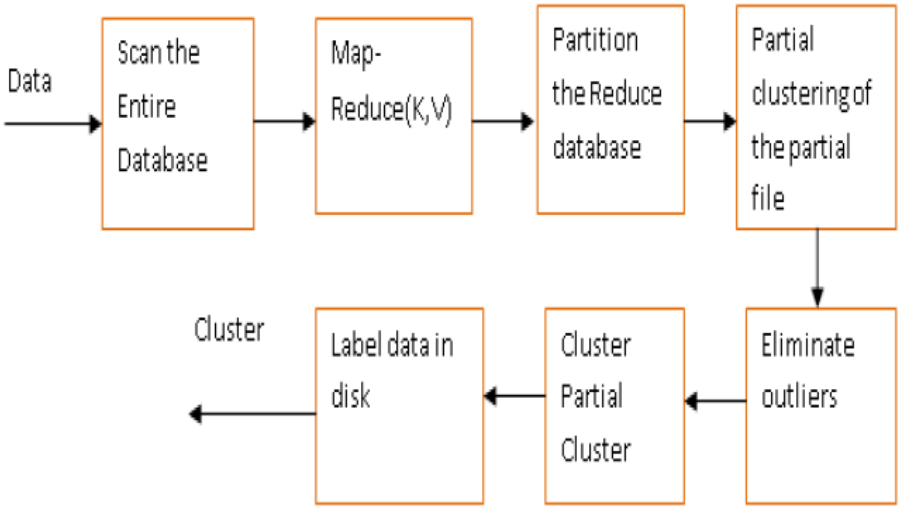

Figure 4.6 CURE algorithm with Map Reduce

So basic Functionality of Proposed algorithm is to execute CURE algorithm in a parallel environment so that processing time can be reduced and data is stored at a replicate location which is HDFS file system so that data cannot be lost.

Detailed processing of operations which are performed on mapper and reducer are described as follow:

- For Mapper input is all the data points which are in the form of (key, value) pair which are (`LongWritable, Text`) .
- Here the key is use just for indexing and key is given data point which is taken in Text format.Mapper processes this data points which are read in text form

and convert them to DoubleArrayWrittable form. So the output of Mapper is in the form of (`LongWritable, DoubleArrayWritable`).

- Reducer selects random data points out of it and process it so input for reducer is also in the form of (`LongWritable, DoubleArrayWritable`) which is the output of all the mappers.
- After processing the all the data points finally we have different data points clustered in different clusters which are written as a text file in (Double, Double) form.
- This output file is stored in HDFS output and can be retrieved in local machine for evaluation purpose. We can draw this cluster points as scattered points using GNUplot or another plotting tool.

Chapter 5 Implementation Details

Different types of Hierarchical Clustering algorithms like BIRCH, traditional CURE and CHAMELEON has been implemented in Java as shown in Chapter 4. As described in chapter 4, there is clear need to create a new way to implement CURE algorithm in new framework Proposed algorithm for CURE in distributed environment using Hadoop has been described in this chapter along with result set and diagrams.

5.1 Implementation of Environment for Hadoop

Cure algorithm is tested in two different environments: a single machine and Hadoop cluster of more than one nodes.

5.1.1 Single Machine

Before implementing it on the dedicated multimode framework, it is tested on a single machine by installing it. Software installation done for it are Linux Mint 17.3, Java OpenJDK 1.8, and Hadoop 2.6.0. Hardware Configurations of this machine are given in Table 5.1.

Table 5.1 Configuration of the single machine

Component	Configuration
CPU	Intel i5-3470
Hard Disk	500 GB
RAM	8GB DDR3

In this single node, we have install Hadoop and this run in a pseudo-distributed manner means name node and data node are run on the same machine. Here we can directly run CURE algorithm on Hadoop in localhost and no need to configure a connection with another machine. Although there will be no any performance improvement on a single machine.

5.1.2 Dedicated Hadoop Cluster

After successfully run on single machine scalability of the algorithm is tested on a dedicated Hadoop cluster this all setup consist of a name node along with three data

nodes. The configuration of all this is describing is fig. All the machines have Linux Mint/Ubuntu, Java OpenJDK 1.8 and Hadoop 2.6.0. All the machine communicates with each other using Gigabit switch/Wi-Fi connection. . Hardware Configurations of this setup are given in Table 5.2

Table 5.2 Configuration of the machines for dedicated Hadoop

	Node1 (Master)	Node 2(Slave)	Node3(Slave)	Node4(Slave)
CPU	Intel(R) Core(TM) i5	Intel(R) Core(TM) i5	Intel(R) Core™i3-i2120	Intel(R) Core(TM)i3-2120
Hard Disk	500 GB	1024GB	320GB	320GB
RAM	16 GB	8 GB	3GB	3 GB
OD	Linux Mint 17.3	Ubuntu 15.6	Linux Mint 17.3	Ubuntu 15.6
Java JDK	1.8	1.8	1.8	1.8
Ethernet	1 Gbps	1 Gbps	1 Gbps	1 Gbps

Here we have Master node with high configuration compare to slave nodes. Master node act as a name node and all other three nodes are act as jobnode while executing the job.

5.2 Experimental Setup

To run Job in Hadoop environment, slave nodes are connected to the master node via a communication medium. After that Hadoop environment can start by “start-all.sh” which start all the process in master node and slave node. So in master node process need to start are NodeManager, Namenode, SecondaryNameNode, ResourceManager. Here Name node act as Master and all the processing of Hadoop is controlled by it. Secondary Namenode acts as a backup name node for Hadoop. The resource manager is a new concept of yarn which is newly introduce in

Hadoop2.0. Along with this Datanode is also start in all slave node which consists Datanode, the Node manager.

Here one data node is created per slave machine in setup. NodeManager is responsible for managing all the activity of that job node and manage data stored in this. Screenshot for Master node is Figure 5.1.

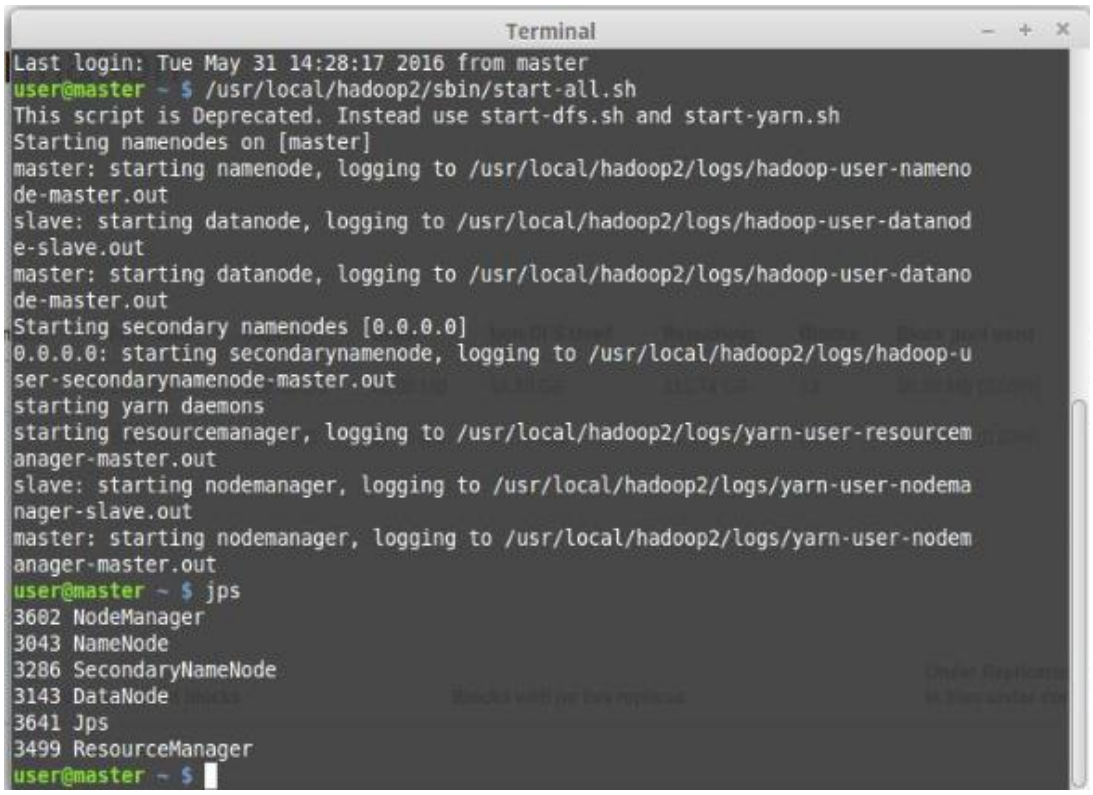

Figure 5.1. Master node processes

Screenshot for one of the Slave node is given in Figure 5.2.

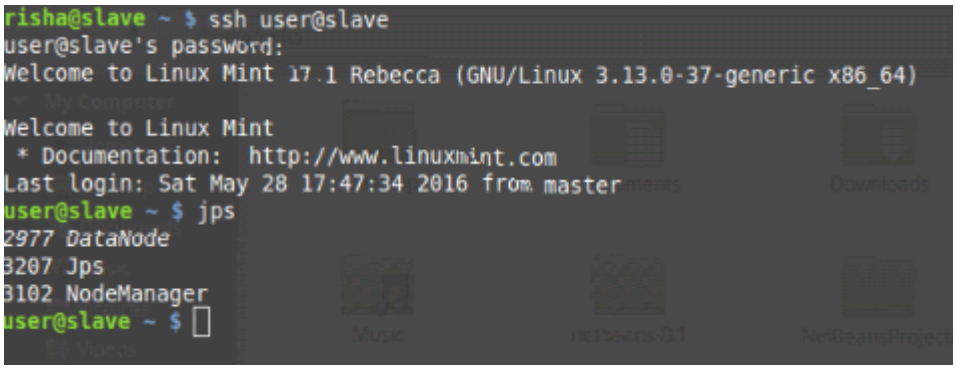

Figure 5.2. Slave node process

So after starting with the Hadoop setup, it is possible to check its status on browser using http://localhost:50070/ which shows status of master node along with information about all the Datanodes and File system along with all the required parameters and IDs as shown in below Figure 5.3 (a)

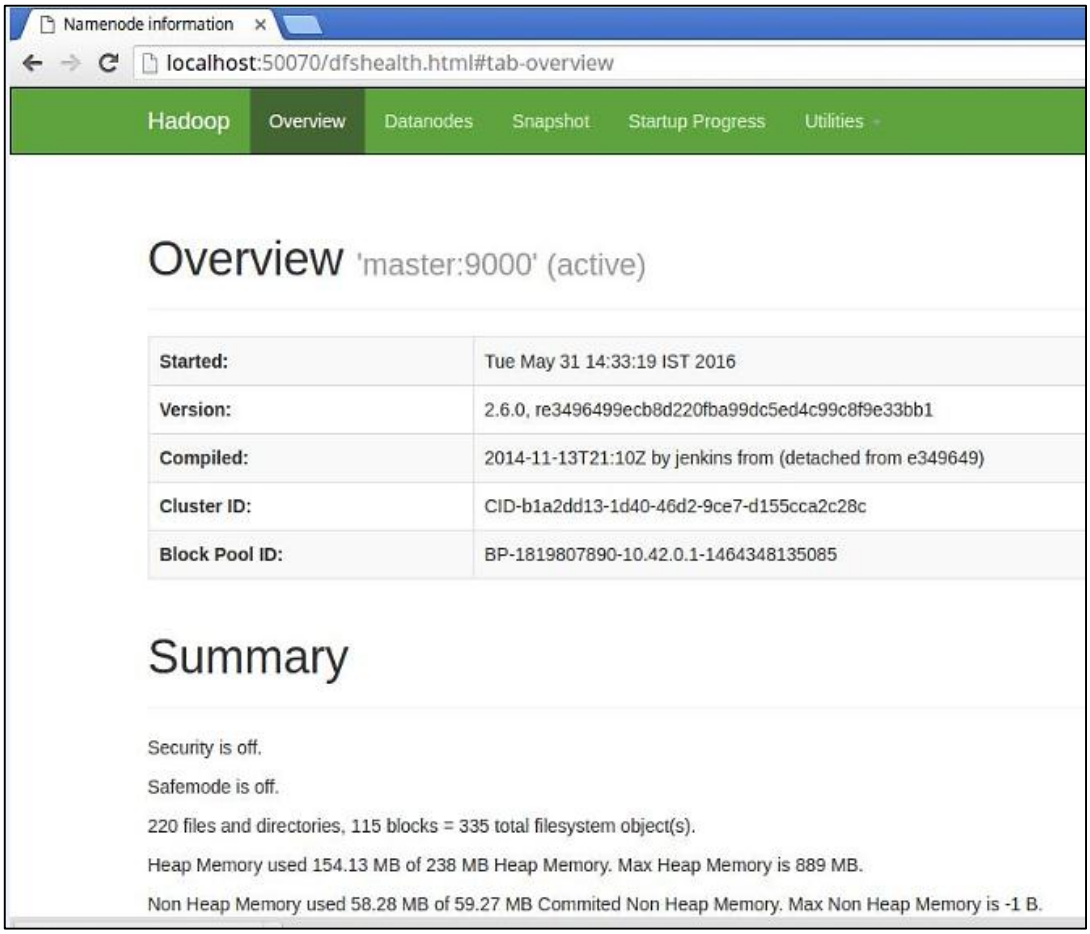

Figure 5.3(a) Status of master node and other parameters

Figure 5.3(b) is taken when two slaves are connected with master and it shows memory available in DFS along with its capacity and status of blocks.

Configured Capacity:	167.93 GB
DFS Used:	133.31 MB
Non DFS Used:	34.64 GB
DFS Remaining:	133.16 GB
DFS Used%:	0.08%
DFS Remaining%:	79.3%
Block Pool Used:	133.31 MB
Block Pool Used%:	0.08%
DataNodes usages% (Min/Median/Max/stdDev):	0.05% / 0.17% / 0.17% / 0.06%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	5/31/2016, 2:33:19 PM

Figure 5.3(b) Status of master node and other parameters

Figure 5.4 shows status of Secondary name node:

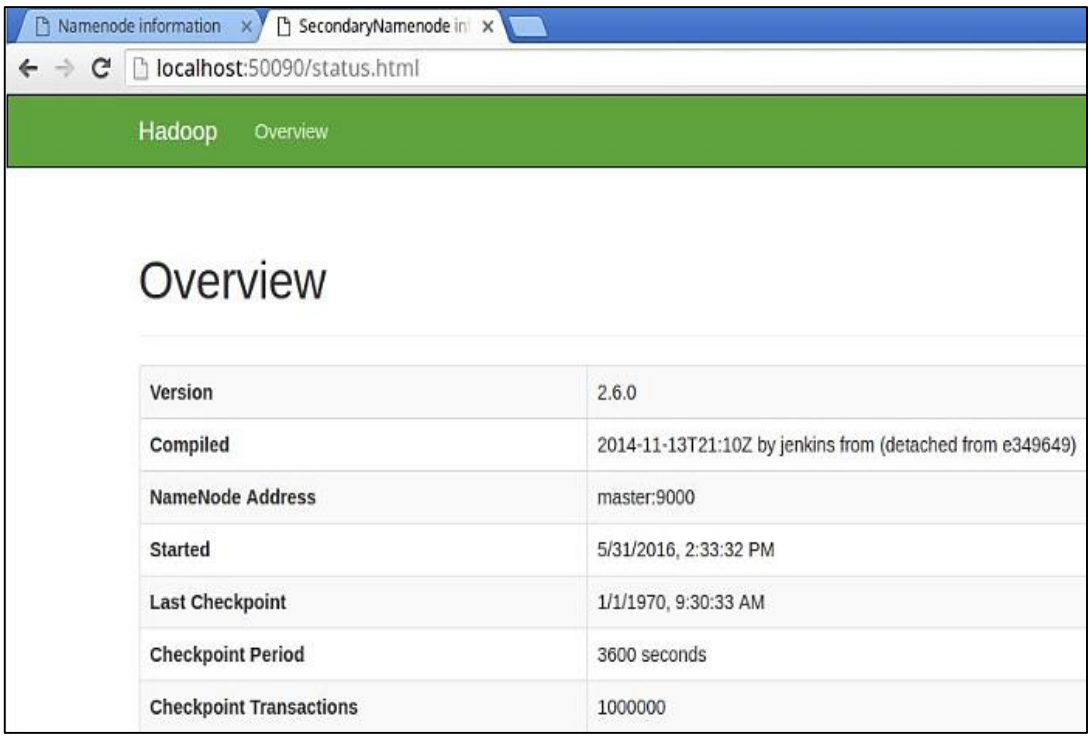

Figure 5.4. Status of Secondary namenode

Here When Hadoop Setup is created HDFS is also created in all the slaves connected in connection. Before processing any source data it needs to be stored in HDFS. When

data is stored in HDFS it is replicated on maximum three slave machines. In Figure 5.5, Screenshot for HDFS is given when one slave and master node is taken as HDFS storage medium:

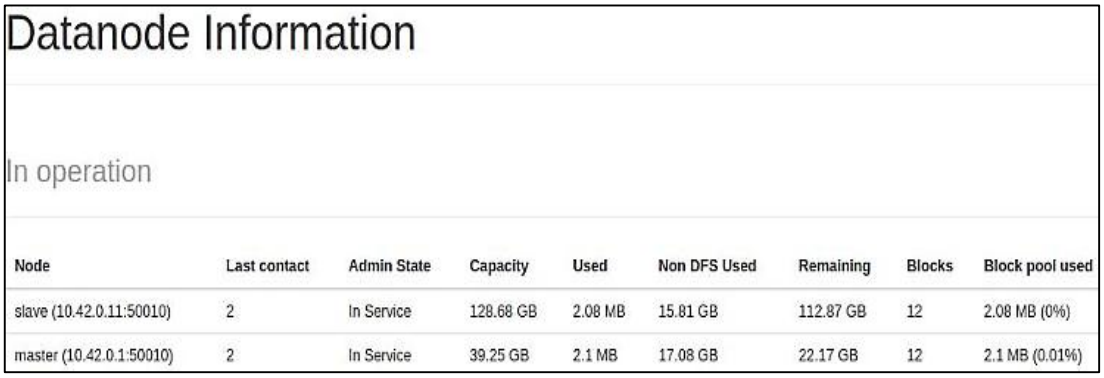

Figure 5.5 Data stored in Datanode

5.3 Results and Analysis

After successfully implementing setup of Hadoop environment, to execute CURE algorithm on Hadoop, the algorithm is needed to be in runnable JAR format. First of all, we run CURE algorithm on Dataset1 (geo-station data) having 1000 data points which are very large for a single machine to be a process. This 1.9 MB data file is uploaded and then processed with parameters given as shown in Table 5.3.

Table 5.3 Parameters for CURE algorithm in Hadoop

	Dataset 1	Dataset 2
Data file size		26.2 MB
Sample Size	1000	1248842
Shrink factor	0.5	0.5
Number of Cluster	5	5
Representative Points	6	6
Number of Partition	2	2
Reducing Factor for each Partition	3	3
Required Representation Probability	0.1	0.1

When Dataset 1 is executed on Hadoop using JAR file for CURE algorithm. Screenshots of it are shown in Figure 5.6.

```
user@master /home/piyush/Desktop $ /usr/local/hadoop2/bin/hdfs dfs -put /home/piyush/Desktop/curedata.txt /usr/local/had
nput
user@master /home/piyush/Desktop $ /usr/local/hadoop2/bin/hadoop jar test2.jar /usr/local/hadoop2/input/curedata.txt /us
l/hadoop2/output22
hiiiphello
16/05/27 19:01:15 INFO client.RMProxy: Connecting to ResourceManager at master/10.42.0.1:8032
16/05/27 19:01:16 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool inte
and execute your application with ToolRunner to remedy this.
16/05/27 19:09:04 WARN hdfs.DFSClient: Slow waitForAckedSeqno took 55264ms (threshold=30000ms)
16/05/27 19:09:04 INFO input.FileInputFormat: Total input paths to process : 1
16/05/27 19:09:04 INFO mapreduce.JobSubmitter: number of splits:1
16/05/27 19:09:06 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1464348161271_0006
16/05/27 19:09:06 INFO impl.YarnClientImpl: Submitted application application_1464348161271_0006
16/05/27 19:09:06 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1464348161271_0006/
16/05/27 19:09:06 INFO mapreduce.Job: Running job: job_1464348161271_0006
16/05/27 19:09:13 INFO mapreduce.Job: Job job_1464348161271_0006 running in uber mode : false
16/05/27 19:09:13 INFO mapreduce.Job:  map 0% reduce 0%
16/05/27 19:09:18 INFO mapreduce.Job:  map 100% reduce 0%
16/05/27 19:09:24 INFO mapreduce.Job:  map 100% reduce 100%
16/05/27 19:09:24 INFO mapreduce.Job: Job job_1464348161271_0006 completed successfully
16/05/27 19:09:24 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=30006
                FILE: Number of bytes written=271797
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=22362
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3116
                Total time spent by all reduces in occupied slots (ms)=3060
                Total time spent by all map tasks (ms)=3116
                Total time spent by all reduce tasks (ms)=3060
                Total vcore-seconds taken by all map tasks=3116
```

Figure 5.6(a) cmd execution of Dataset1 in Hadoop environment

```
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3116
                Total time spent by all reduces in occupied slots (ms)=3060
                Total time spent by all map tasks (ms)=3116
                Total time spent by all reduce tasks (ms)=3060
                Total vcore-seconds taken by all map tasks=3116
                Total vcore-seconds taken by all reduce tasks=3060
                Total megabyte-seconds taken by all map tasks=3190784
                Total megabyte-seconds taken by all reduce tasks=3133440
        Map-Reduce Framework
                Map input records=1000
                Map output records=1000
                Map output bytes=28000
                Map output materialized bytes=30006
                Input split bytes=120
                Combine input records=0
                Combine output records=0
                Reduce input groups=1000
                Reduce shuffle bytes=30006
                Reduce input records=1000
                Reduce output records=0
                Spilled Records=2000
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=115
                CPU time spent (ms)=1820
                Physical memory (bytes) snapshot=411947008
                Virtual memory (bytes) snapshot=3823009792
                Total committed heap usage (bytes)=308281344
```

Figure 5.6(b) cmd execution of Dataset1 in Hadoop environment

Here we can see in the screenshot that only one split has done and the process is run in only one machine due to less size of data points. “Job completed successfully” shown at the end of mapping reducing indicate that clustering is properly done. The result set is stored in the output folder and from there it can retrieve in the local machine. When dataset having 1000 data points from HDFS file system, are given to CURE algorithm implemented in MapReduce, it performs Random Sampling, Partitioning, Partial Clustering, Outliers Removing and global clustering on it and draws this clusters on any plotting tool like GNU plot than output scattered plot is as shown in Figure 5.7.

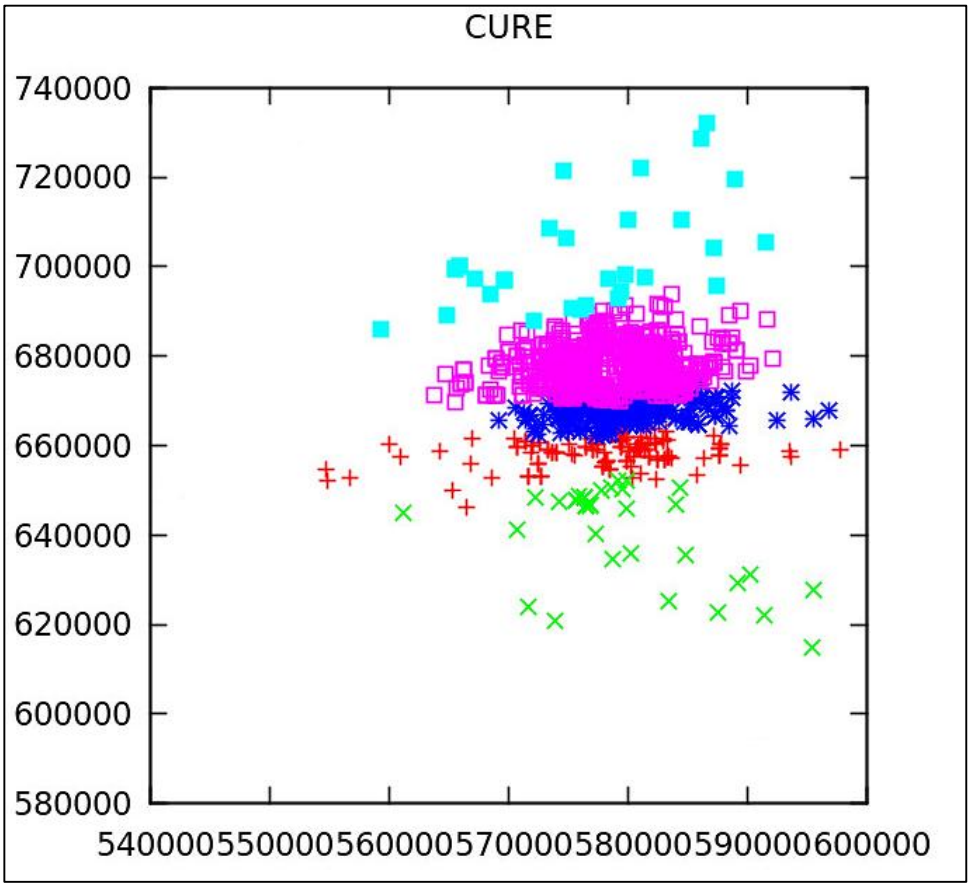

Figure 5.7 Output of dataset1

In this Plot outliers are not shown because it is removed from further clustering process. Here Shrinking Factor can be any value between 0 and 1. If it is 0 than it indicates all point based approach and if it is 1 than indicate centroid based approach. We have taken a=0.5 to take more than one representative in consideration. Reducing Factor define at which extend clustering is to be done. It is set to 3 means clustering

is done till a number of the cluster in each partition becomes 1/3 of the number of points at the initial level.

Reducing Factor plays a major role in Outliers detection. If reducing factor exceeds predefine value then number of the cluster in each partition (total number of points/ (the number of partition*Reducing Factor)) reduce. Because of this, data points in different clusters may merge. But it impacts the overall quality of cluster so needs to be avoided. So in this experiment if reducing factor exceeds a Threshold value which is 3, than all the sub-cluster having only one data point are considered as outliers and removed from clustering. However with increment in value of reducing Factor after Threshold limit, the number of Outliers decrease (as shown in the graph in Figure5.8) and that will result in only identifying cluster at maximum distance.

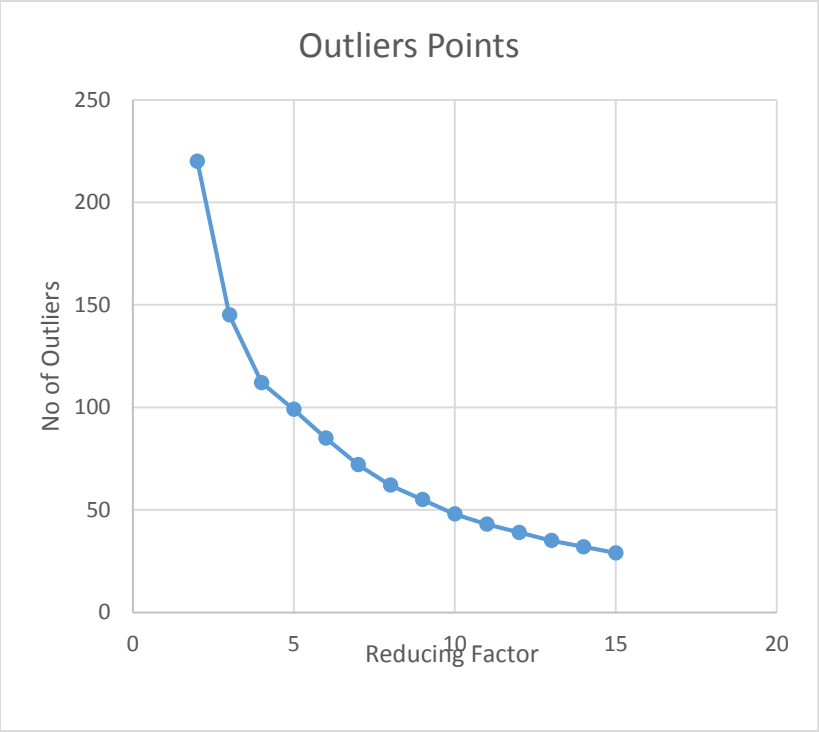

Fig. 5.8. Effect on number of outliers with reducing factor

If we execute a large amount of data or a large number of data files than they are executed parallel in multiple mappers and it may possible to have more than one split. In second, if dataset 2, which is data about Europe continental analysis, having 1248842 number of records is run on given CURE algorithm as shown in Figure 5.9 (a), (b) and (c).

```
user@master /home/piyush/Desktop $ /usr/local/hadoop2/bin/hadoop jar curewithmr.jar /usr
/local/hadoop2/curedatasets /usr/local/hadoop2/cureoutputdata1
hiiiphello
16/05/31 15:16:59 INFO client.RMProxy: Connecting to ResourceManager at master/10.42.0.1
:8032
16/05/31 15:17:01 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not pe
rformed. Implement the Tool interface and execute your application with ToolRunner to re
medy this.
16/05/31 15:26:14 WARN hdfs.DFSClient: Slow waitForAckedSeqno took 62113ms (threshold=30
000ms)
16/05/31 15:26:14 INFO input.FileInputFormat: Total input paths to process : 4
16/05/31 15:26:14 INFO mapreduce.JobSubmitter: number of splits:4
16/05/31 15:26:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1464685439
889_0002
16/05/31 15:26:17 INFO impl.YarnClientImpl: Submitted application application_1464685439
889_0002
16/05/31 15:26:17 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy
/application_1464685439889_0002/
16/05/31 15:26:17 INFO mapreduce.Job: Running job: job_1464685439889_0002
16/05/31 15:26:25 INFO mapreduce.Job: Job job_1464685439889_0002 running in uber mode :
false
16/05/31 15:26:25 INFO mapreduce.Job:  map 0% reduce 0%
16/05/31 15:27:10 INFO mapreduce.Job:  map 25% reduce 0%
16/05/31 15:27:13 INFO mapreduce.Job:  map 36% reduce 0%
16/05/31 15:27:14 INFO mapreduce.Job:  map 38% reduce 0%
16/05/31 15:27:18 INFO mapreduce.Job:  map 45% reduce 0%
16/05/31 15:27:20 INFO mapreduce.Job:  map 53% reduce 0%
16/05/31 15:27:21 INFO mapreduce.Job:  map 55% reduce 0%
16/05/31 15:27:25 INFO mapreduce.Job:  map 59% reduce 0%
16/05/31 15:27:28 INFO mapreduce.Job:  map 62% reduce 0%
16/05/31 15:27:31 INFO mapreduce.Job:  map 66% reduce 0%
16/05/31 15:27:35 INFO mapreduce.Job:  map 77% reduce 0%
16/05/31 15:27:48 INFO mapreduce.Job:  map 78% reduce 0%
16/05/31 15:27:50 INFO mapreduce.Job:  map 78% reduce 17%
16/05/31 15:27:52 INFO mapreduce.Job:  map 78% reduce 25%
16/05/31 15:27:54 INFO mapreduce.Job:  map 79% reduce 25%
16/05/31 15:28:04 INFO mapreduce.Job:  map 80% reduce 25%
16/05/31 15:28:10 INFO mapreduce.Job:  map 81% reduce 25%
16/05/31 15:28:16 INFO mapreduce.Job:  map 82% reduce 25%
16/05/31 15:28:23 INFO mapreduce.Job:  map 83% reduce 25%
16/05/31 15:28:32 INFO mapreduce.Job:  map 84% reduce 25%
16/05/31 15:28:42 INFO mapreduce.Job:  map 85% reduce 25%
```

Figure5.9 (a) Execution of Hadoop for Dataset2

```
16/05/31 15:28:42 INFO mapreduce.Job:  map 85% reduce 25%
16/05/31 15:28:51 INFO mapreduce.Job:  map 86% reduce 25%
16/05/31 15:28:57 INFO mapreduce.Job:  map 87% reduce 25%
16/05/31 15:29:04 INFO mapreduce.Job:  map 88% reduce 25%
16/05/31 15:29:13 INFO mapreduce.Job:  map 89% reduce 25%
16/05/31 15:29:22 INFO mapreduce.Job:  map 90% reduce 25%
16/05/31 15:29:32 INFO mapreduce.Job:  map 91% reduce 25%
16/05/31 15:29:41 INFO mapreduce.Job:  map 92% reduce 25%
16/05/31 15:30:00 INFO mapreduce.Job:  map 100% reduce 25%
16/05/31 15:30:03 INFO mapreduce.Job:  map 100% reduce 67%
16/05/31 15:30:09 INFO mapreduce.Job:  map 100% reduce 69%
16/05/31 15:30:12 INFO mapreduce.Job:  map 100% reduce 72%
16/05/31 15:30:15 INFO mapreduce.Job:  map 100% reduce 74%
16/05/31 15:30:18 INFO mapreduce.Job:  map 100% reduce 77%
16/05/31 15:30:21 INFO mapreduce.Job:  map 100% reduce 79%
16/05/31 15:30:25 INFO mapreduce.Job:  map 100% reduce 81%
16/05/31 15:30:28 INFO mapreduce.Job:  map 100% reduce 84%
16/05/31 15:30:31 INFO mapreduce.Job:  map 100% reduce 86%
16/05/31 15:30:34 INFO mapreduce.Job:  map 100% reduce 88%
16/05/31 15:30:37 INFO mapreduce.Job:  map 100% reduce 90%
16/05/31 15:30:40 INFO mapreduce.Job:  map 100% reduce 93%
```

Figure5.9 (b) Execution of Hadoop for Dataset2

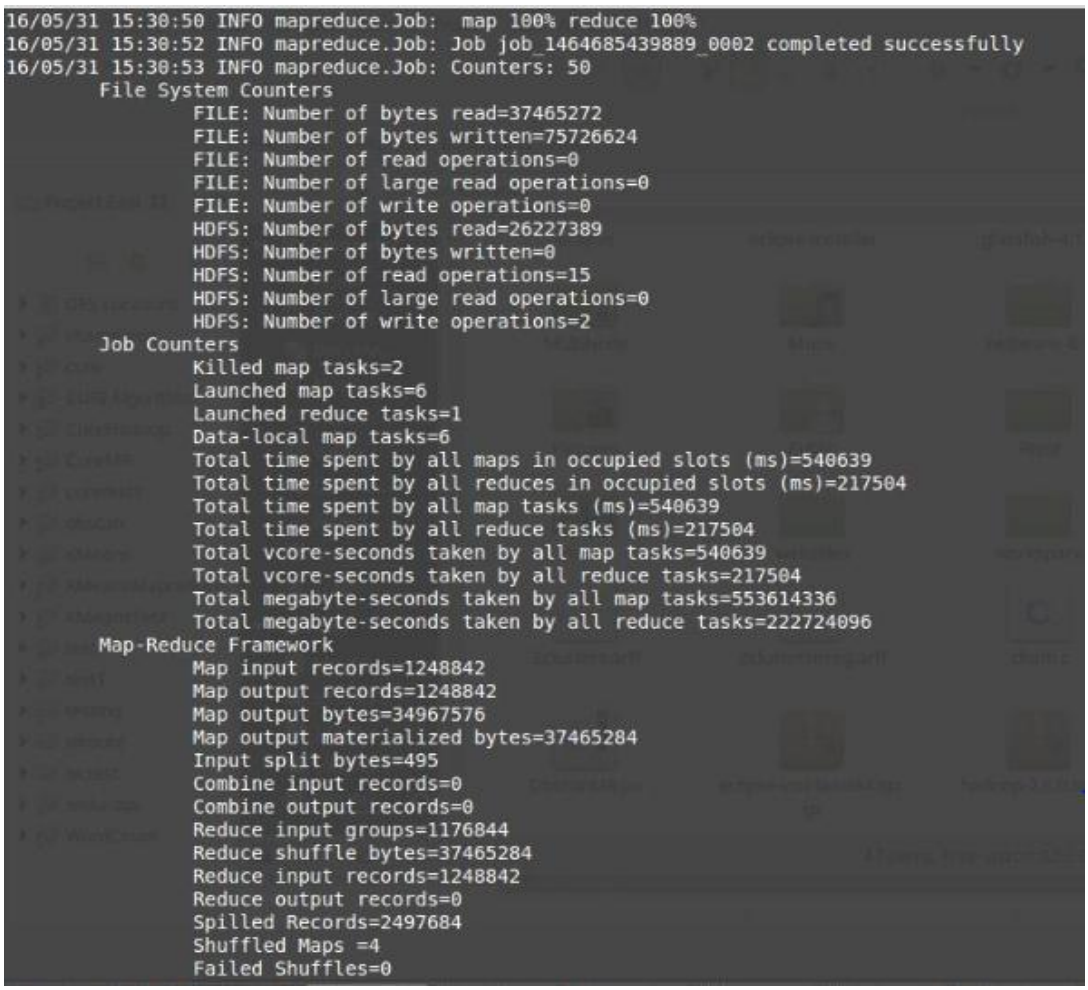

Figure 5.9(c) Execution of Hadoop for Dataset2

As shown in Figure 5.9 (a) there are 4 splits has been done and four mapper functions will execute with each split and then reduce function start to work. After processing the algorithm clusters of data points are created.

Here whatever data files are processed is of small size in the range of 26.2 MB. So there will be only 1 block will be generated in HDFS. But if the data file is more than 128 MB than, each block of size 128 MB are generated.

As shown in figure 5.10(a) and (b), data file is 143.4 MB so two blocks are generated

Figure 5.10(a) and (b) File system for data file size more than 128 MB

```
16/05/31 18:37:46 INFO mapreduce.Job:  map 100% reduce 98%
16/05/31 18:37:52 INFO mapreduce.Job:  map 100% reduce 99%
16/05/31 18:37:59 INFO mapreduce.Job:  map 100% reduce 100%
16/05/31 18:38:04 INFO mapreduce.Job: Job job_1464685439889_0008 completed successfully
16/05/31 18:38:04 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=410400030
                FILE: Number of bytes written=617414762
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=143639931
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

Fig5.11 Job done for data size more than 128 MB

When CURE algorithm is applied successfully on data file having size more than 128 MB which is stored in two data blocks and all are replicated on multiple slave file systems.

Chapter 6 Conclusion and Future Scope

6.1 Conclusion

Clustering is one type of grouping method of given objects in such a way that inter-cluster similarity is minimum and the intra-cluster similarity is maximum. When Clustering algorithms are used for a large amount of data, majority of them has efficiency and scalability problems and incapable of recognizing non-spherical shape clusters. CURE algorithm can recognize the non-spherical shape and also can handle the massive amount of data. Due to more than one representative point, CURE is able to identify outliers and eliminate it from clustering process. However, traditional CURE algorithm can cannot cluster massive amount of data. In this thesis CURE algorithm is implemented with the help of Hadoop framework which is more convenient way to cluster large data. It is implemented on Distributed environment using MapReduce programming model so that it can handle the massive amount of data and can cluster in less amount of time and space[17].

6.2 Summary of Contributions

In this thesis, the contributions done by this research work are summarized as follows:

- Survey of Existing Hierarchical clustering methods and implemention using Java programming language.
- Methodology to implement CURE algorithm in Hadoop using Mapper-Reducer files and try to achieve parallelism in processing.
- Identify outliers existing in clustering process and try to find useful patterns in selection of given parameters to minimize the final clusters as well as to maximize quality of clusters created.

6.3 Future Scope

Currently, CURE algorithm is implemented on Hadoop environment by making setup of an only limited number of nodes check its performance. In future, it can be expanded to implement in dedicated Hadoop setup having more nodes to achieve higher efficiency and speed. In the Current algorithm, we have used Random sampling to select only some data objects to represent all data objects in clustering activity. In Future instead of random sampling, some other approach can be used to select data objects by using the concept of nesting of mapreduce in which data points are selected as separate mapreduce activity. One more important thing to be done is to deal with outliers, although it is not much important to ignore some points but still there should some way exist to handle outliers in a logical way. CURE clustering is one type of static modeling process which not cares about relative distance. There is a need to have some dynamic clustering algorithm which takes relative connectivity and closeness into consideration while clustering.

References

[1] M. Brown, "Data mining techniques", Ibm.com, 2016. [Online]. Available: http://www.ibm.com/developerworks/library/ba-data-mining-techniques/.

[2] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "Clustering algorithms and validity measures", In Proceedings of IEEE 13th International Conference on Scientific and Statistical Database Management, 2001, pp. 3-22.

[3] "Welcome to Apache™ Hadoop®!", Hadoop.apache.org, 2016. [Online]. Available: http://hadoop.apache.org/.

[4] S. Ibrahim, H. Jin, L. Lu, L. Qi, X. Shi and S. Wu, "Evaluating mapreduce on virtual machines: The hadoop case", Cloud Computing, pp. 519-528, 2009.

[5] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis", IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 3, pp. 267-279, 2014.

[6] Z. He, S. Deng and X. Xu, "An optimization model for outlier detection in categorical data", Advances in Intelligent Computing, pp. 400-409, 2005.

[7] R. Ng and Jiawei Han, "CLARANS: A method for clustering objects for spatial data mining", IEEE Transaction on Knowledge and Data Engineering, vol. 14, no. 5, pp. 1003-1016, 2002.

[8] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH", ACM SIGMOD Record, vol. 25, no. 2, pp. 103-114, 1996.

[9] M. Ester, H. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", Knowledge Discovery and Data Mining ,vol.96, no.34, pp. 226-231, 1996.

[10] G. Karypis, Eui-Hong Han and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling", IEEE Computer, vol. 32, no. 8, pp. 68-75, 1999.

[11] S. Guha, R. Rastogi and K. Shim, "CURE: An efficient clustering algorithm for large databases", ACM SIGMOD Record, vol. 27, no. 2, pp. 73-84, 1998.

[12] R. Xu and D. WunschII, "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, 2005.

[13] M. Henzinger, "Finding near-duplicate web pages: a large-scale evaluation of algorithms", in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 284-291, 2006.

[14] J. Dean and S. Ghemawat, "MapReduce", Communications of the ACM, vol. 51, no. 1, p. 107-113, 2008.

[15] P. Anchalia and K. Roy, "The k-Nearest Neighbor Algorithm Using MapReduce Paradigm", 2014 5th International Conference on Intelligent Systems, Modelling and Simulation, pp. 513-518, 2014.

[16] S. Blanas, J. Patel, V. Ercegovac, J. Rao, E. Shekita and Y. Tian, "A comparison of join algorithms for log processing in mapreduce",In Proceedings of the ACM SIGMOD International Conference on Management of data, pp. 975-986, 2010.

[17] K. Lee, Y. Lee, H. Choi, Y. Chung and B. Moon, "Parallel data processing with MapReduce", ACM SIGMOD Record, vol. 40, no. 4, pp. 11-20, 2012.

[18] P. Berkhin, "A survey of clustering data mining techniques", Grouping multidimensional data, pp. 25-71, 2006.

[19] H. Huang, Y. Gao, K. Chiew, L. Chen and Q. He, "Towards effective and efficient mining of arbitrary shaped clusters", IEEE 30th International Conference on Data Engineering (ICDE), pp. 28-39, 2014.

[20] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets", In Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 515-524, 2002.

[21] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms", 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 576-584, 2004.

[22] M. J. van der Laan and K. Pollard, "A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap", Journal of Statistical Planning and Inference, vol. 117, no. 2, pp. 275-303, 2003.

[23] S. Horng, M. Su, Y. Chen, T. Kao, R. Chen, J. Lai and C. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, vol. 38, no. 1, pp. 306-313, 2011.

[24] Y. Malitsky, A. Sabharwal, H. Samulowitz and M. Sellmann, "Algorithm Portfolios Based on Cost-Sensitive Hierarchical Clustering", International Joint Conference on Artificial Intelligence, vol. 13, pp. 608-614, 2013.

[25] K. Kaur and R. Rani, "Managing Data in Healthcare Information Systems: Many Models, One Solution", Computer, vol. 48, no. 3, pp. 52-59, 2015.

[26] S. Mingoti and J. Lima, "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms", European Journal of Operational Research, vol. 174, no. 3, pp. 1742-1759, 2006.

[27] A. Shepitsen, J. Gemmell, B. Mobasher and R. Burke, "Personalized recommendation in social tagging systems using hierarchical clustering", pp. 259-266, 2008.

[28] H. Koga, T. Ishibashi and T. Watanabe, "Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing", Knowledge and Information Systems, vol. 12, no. 1, pp. 25-53, 2006.

[29] O. Abbas, "Comparisons Between Data Clustering Algorithms", The International Arab Journal of Information Technology, vol. 5, no. 3, pp. 320-325, 2008.

[30] A. Jain, "Data clustering: 50 years beyond K-means", Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.

[31] V. Murthy, E. Vamsidhar, J. Kumar and P. Rao, "Content based image retrieval using Hierarchical and K-means clustering techniques", International Journal of Engineering Science and Technology, vol. 2, no. 3, pp. 209-212, 2010.

[32] G. Kou and C. Lou, "Multiple factor hierarchical clustering algorithm for large scale web page and search engine clickstream data", Annals of Operations Research, vol. 197, no. 1, pp. 123-134, 2010.

[33] Baljit Kaur and Rinkle Rani, "Designing URL Access Counter using MapReduce with HBase," in Proceedings of Elsevier Second International Conference on Emerging Research In Computing Information, Communication and Applications - (ERCICA-14), 01-02 Aug. 2014.

[34] P. Langfelder and S. Horvath, "Fast R Functions for Robust Correlations and Hierarchical Clustering", Journal of Statistical Software, vol. 46, no. 11, pp. 1-17, 2012.

[35] Y. Malitsky, A. Sabharwal, H. Samulowitz and M. Sellmann, "Algorithm Portfolios Based on Cost-Sensitive Hierarchical Clustering", International Joint Conference on Artificial Intelligence, vol. 13, pp. 608-614, 2013.

[36] M. Meila and D. Heckerman, "An experimental comparison of several clustering and initialization methods",In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 386-395, 1998.

[37] D. Müllner, "fastcluster : Fast Hierarchical, Agglomerative Clustering Routines for R and Python", Journal of Statistical Software, vol. 53, no. 9, pp. 1-18, 2013.

[38] M. Balcan, Y. Liang and P. Gupta, "Robust hierarchical clustering", The Journal of Machine Learning Research, vol. 15, no. 1, pp. 3831-3871, 2014.

[39] S. Szilágyi and L. Szilágyi, "A fast hierarchical clustering algorithm for large-scale protein sequence data sets", Computers in Biology and Medicine, vol. 48, pp. 94-101, 2014.

[40] S. Guha, R. Rastogi and K. Shim, "Rock: A robust clustering algorithm for categorical attributes", Information Systems, vol. 25, no. 5, pp. 345-366, 2000.

[41] A. Jain, M. Murty and P. Flynn, "Data clustering: a review", ACM Computing Surveys (CSUR), vol. 31, no. 3, pp. 264-323, 1999.

[42] Z. Abdulla and A. Hamdan, "Hierarchical Clustering Algorithms in Data Mining", International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 9, no. 10, pp. 1921-1926, 2015.

[43] A. Krishnamurthy, S. Balakrishnan, M. Xu and A. Singh, "Efficient Active Algorithms for Hierarchical Clustering", in Proceedings of the 29th International Conference on Machine Learning, pp. 887-894, 2012

List of Publications and Video Link

Publications

P. Lathiya, R. Rani, *"Improved CURE Clustering for Big Data using Hadoop and Mapreduce"*, IEEE International Conference on Inventive Computation Technologies (ICICT 2016). Date-26-27 August-16 [**Accepted**]

Video Link

https://youtu.be/5-ZOTUq2rt4

Plagiarism Report

ppaper

ORIGINALITY REPORT

9%	7%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijcsit.com Internet Source	1%
2	waset.org Internet Source	1%
3	www-users.cs.umn.edu Internet Source	1%
4	Tomar, Divya and Agarwal, Sonali. "A survey on Data Mining approaches for Healthcare", International Journal of Bio-Science & Bio-Technology, 2013. Publication	1%
5	www.cs.uiuc.edu Internet Source	<1%
6	dspace.thapar.edu:8080 Internet Source	<1%
7	Han, Jiawei, Micheline Kamber, and Jian Pei. "Cluster Analysis", Data Mining, 2012. Publication	<1%

Sachin Kulkarni. "Scalable Clustering for Large