Design and Implementation of Clustering Algorithm for Big Data Analytics

Project report submitted in fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

Parminder Singh Thakur (131267)

Ankit Sharma (131263)

Under the supervision of

(Dr. Pardeep Kumar)

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Design and Implemention of Clustering algorithm for Big Data Analytics"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to May 2017 under the supervision of **Dr. Pardeep Kumar,** Assistant Professor (Senior Grade), Computer Science and Engineering & IT.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ankit Sharma, 131263

Parminder Singh Thakur, 131267

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

Dr. Pardeep Kumar

Assistant Professor (Senior Grade)

Computer Science and Engineering Department

Dated:

ACKNOWLEDGEMENT

We owe our insightful appreciation and gratitude to our project supervisor **Dr. Pardeep Kumar**, Assistant professor (senior grade), who took keen interest and guided us all along in the project work, titled **—**"**Design and Implementation of Clustering Algorithm for Big Data Analytics"**, till the completion of the project by providing all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in the domain of Big Data. We are in truth grateful to him.

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

This project is intended to implement clustering on Big Data sets. Data analysis is a crucial part of the process of formulating new policies and strategies for an organization that help the company to be successful in competitive markets, allowing them to better understand their customer base and giving solutions to so hard problems and thus gain competitive advantage. With almost every aspect of human life connected to the internet, data analysis has gained greater importance in fields like Smart Healthcare and national security and is now not just limited to business analysis. The vast and diverse nature of such Big data sets imposes new challenges as the traditional methods of knowledge discovery from databases are not equipped to handle such Big data. Distributed and Parallel computing frameworks are the key for such analysis.

The purpose of clustering algorithms is to make sense of and extract value from large sets of structured and unstructured data. The technique of clustering enables a data analytic to obtain a snapshot of data of huge volumes and complexities which can be used to form some logical structures on such huge volumes of complex data. Thus clustering can provide with some form of structure and insight to the nature of data at hand and can form the basis of further analysis. To handle Big Data clustering various limitations of present clustering techniques are needed to be mitigated by understanding the framework used to handle Big data sets and analyzing data clustering. The target is to analyze these challenges in order to redesign and implement a clustering algorithm that is suitable for Big Data sets based on today's available technologies and frameworks. Furthermore, the possible future path for more advanced algorithms is illuminated.

1 Introduction

1.1 Introduction

The rapid rate at which every day to day work is becoming automated due to the advancement of various computing technologies has given rise to a new field of computer science or more specifically a new field in data science known as Big Data analytics. Thousands of online applications, ranging from social networking applications to health tracking applications, are available to the end user that can be accessed through their desktops or handheld devices. Large volume of structured and unstructured data is generated through these applications. Moreover, with the development in communication network technology more and more people and things have connected to the internet and has given birth to concepts like Internet Of Things (IOT). These networks generate data continuously and require large and efficient storage so that the applications using such data can analyze it and use it to deliver end services to the users. Such universal data applications and business analysis generate huge volumes of structured and unstructured data which is further required for knowledge discovery. Big Data Analytics deals with this issue of efficient storage and processing of such large and complex datasets.

Knowledge discovery from data requires the application of data mining. Relevant information retrieval in the form of patterns and trends can be achieved through the application of data mining techniques on large, unstructured, complex and raw data. However, the presence of Big Data sets increases the difficulty and complexity in retrieving some useful information. The intricacies that come with Big Data demand high level of computing resources in order to perform any kind of useful and meaningful processing on such data sets. Stand-alone servers with limited availability of computing resources cannot fulfill such requirements. This kind of data processing demand high availability of computational power which can only be provided by distributed computing. Pool of machines connected to each other are configured in such way that an organization of processors, memory and storage can be achieved.

Data clustering is one of the popular topics in data mining. Clustering attempts to find the hidden patterns and relation among the data points of a data set. The various approaches of clustering the given data set have to be reconfigured in order to handle Big Data.

1.2 **Problem Statement**

The rate of generation of data and the amount of heterogeneous data has exceeded all past records in last decade or so. The reason is the advancement of technology in last twenty years and the widespread scope of the application of computing technology. New challenges have risen due to these events which are grouped together, analyzed and resolved under a common field of computer science known as Big Data Analytics. The major challenge is the storing of such huge data sets while preserving its functional dependencies and keeping the mode and method used for storing these sets transparent to the applications that use and generate this Big Data. Furthermore, the processing requirements for the analysis of Big Data are complex and cannot be met by the traditional computing architecture.

Data clustering has been a popular choice among data analysts during the data mining process. Various methods have been developed in which the data objects of a particular set can be grouped together to find the implicit classification which is otherwise difficult to observe in data with numerous attributes. However, all these methods stand inefficient and incapable in handling Big Data sets. Challenge is to devise approaches which can be used efficiently to obtain clusters in large data sets by using the existing available frameworks.

1.3 Objective

The aim of the project is to study the clustering of big data and implementing an efficient clustering algorithm which is able to handle the complexities of Big Data.

Traditional clustering techniques cannot be applied directly to Big Data so modifications are required in the classical clustering approach to incorporate data mining in Big Data. Thus, through this project we try to achieve an implementation of a clustering algorithm that performs well on big data and proves to be helpful in the knowledge discovery in data mining of big data. By studying various techniques of clustering and their shortcomings with respect to Big Data, we attempt to introduce changes in these algorithms and provide a framework to handle the task of clustering in Big Data.

1.4 **Methodology**

The first task undertaken is the study of the process of knowledge discovery from databases to establish an understanding and generate an overall picture of data mining in order to understand that where and how clustering is used in data analysis.

Second step is to traverse through the concept of Big Data and the various challenges and advantages its processing brings to the table. To study various approaches, computational platforms and frameworks and programming paradigms that have been proposed to tackle the difficulties in the processing of Big Data like Apache Hadoop, Spark, Map Reduce programming, Hive, etc.

Simultaneously analyzing the various techniques of clustering in order to find the technique suitable for the target Data sets and that are supported by the framework which will be used to implementing the algorithm.

The crucial most task is that of redesigning or adapting the chosen algorithm so that it builds on the computing platform used for Big Data. The algorithm should be scalable to handle Big Data.

The final step is to use various data sets to check the correctness of the algorithm by implementing it. Analysis of the approach is conducted to find directions in which the work can be carry forward in order to create a set of algorithms that are capable of handling any type and nature of data sets and which give the desired clusters in a more efficient way and without the much interventions and inputs by the programmer.

2 Literature Survey

2.1 Automated Process of Knowledge Discovery

Usama Fayad et al. [1] establishes a relation between data mining and knowledge discovery from data and also describes how fields like statistics and machine learning important for this purpose. The paper has become the first reference of almost every computer scientist conducting a research in the field of data mining and knowledge discovery. Observation of Usama Fayad et al. [1] about the rapidly growing data volumes and the necessity of developing frameworks and platforms that are equipped to draw information from such data in the year of 1998 is relevant in today's time as well. This has mainly happened because of different fields becoming more and more dependent on data analysis and with data being in thousand of files manual analysis to get some kind of useful information is just not possible today. As pointed in Usama Fayad and others [1] the fields like capital investment where the shares of companies and market positions of every company and every share holder is to be analyzed to make decision of investing what amount of money in which plan. Companies like LBS have developed these systems using techniques of neural networks and other machine learning techniques. Product and services marketing is another field where customer databases are exploited to categorize customers using various partition methods which may follow supervised or unsupervised learning. Healthcare has always relied on data interpretation to find new variants of diseases, viruses and their cures.

The paper defines a process known as KDD process which is used to abstract raw data into more intelligent form from which information can be retrieved. It mainly deals with the strategy of applying various data mining methods for extracting from databases. KDD is not a subset of a particular field but is derived through high performance computing, machine learning and statistics. These various fields are used to derive different components of the process and therefore any development achieved in the field of pattern recognition or machine learning will directly upshot better performance of KDD. Every application would require

specific information from different data sets so the methods and techniques used in the process cannot be made rigid and universal for every information retrieval mechanism. Some areas would require techniques like SVM, neural networks, clustering whereas other would require artificial intelligence, regression and other statistical analysis. Therefore, KDD process is an assortment different fields sort together to achieve specific desired information from the data collected and generated by a specific application.

The process encompasses all operations starting from the storage of data to its processing using scalable data mining algorithms, also the various application specific representations of data objects to forming the model on which the user can interact with the information retrieval system.

As described by Usama Fayad and others [1], the first step which is vital for getting information which is precise and effective is to understand the application which will use the retrieved information by considering the perspective of every party involved mainly its customers. Second step is to reduce the entire collection of facts and figures to the target data set by choosing the appropriate samples. Third step is to decide on what to do with the outliers or noise present in the target data and if some fields are missing then how to treat those data objects. Next step is to find an appropriate version or representation for the data keeping in mind the objective of the entire process. Choosing the technique which can help in getting the desired result from a large array of methods present in data mining like classification, etc. is the next step. After the selection of the data mining method, which algorithms and models are to be used become the next step in the process. Then the methods are implemented and results are generated. Last step is to interpret the results and retrieved models obtained by the data mining and to use the desired information.

Doing all this and doing it efficiently is the goal of KDD process. The paper also discusses the online analytical processing of data warehouses which can be used to transform and clean data to be able to be used by further extraction processes that are the part of knowledge discovery.

Usama fayad et al. [1] divide the knowledge discovery process into five macro steps namely, selection, pre-processing, transformation, data mining and finally interpretation of the results. The definitions and role of each step are clearly formulated in this research paper. Data mining step has been discussed in detail to provide some insight. Various data mining methods such as clustering, regression, Sampling etc. are defined as well.

2.2 Data Mining in Big Data Sets

Xindong Wu et al.[2] forms a basis for understanding of the concept of mining in Big Data. This research article provides a clear and detail definition of Big Data not only in terms of its characteristics but also in terms of its sources and the challenges that are needed to tackle such large data sets when extracting useful information from them and also that such a process of data mining should produce results which are relevant and can be used in a given time frame. Further, Xindong Wu and others through their work introduce a mining framework which can provide solutions to the challenges faced in the mining of such large data sets.

Xindong Wu et al.[2] provides a “HACE theorem which describes the nature of Big Data in terms of it sources, the type of relationship among each of the data objects, its size and structure.” This definition using this theorem is comprehensive and provides a clear insight to what Big Data means. The paper goes on to discuss in detail with illustrious examples about each of the facets of Big Data which include the various types of data objects in terms of their dimensions and volumes, types of sources of these data sets and the dynamics of the relationship among each data object in a particular data set of a particular application domain. Large data stes are often formed by gathering data from different data collectors and producers which leads to the phenomenon of these data sets being miscellaneous and diversified in nature. Each single data point has various number of representations and each representation includes different quality or features that define that single data point. In a

worldwide scenario part from the above diversification in the data set each source of data follows local protocols to collect ad transfer data which adds to the description of the data point in a set thereby increasing the complexity. Third major aspect in Big Data is that each data point is not independent and their correlation varies according to the application sphere of influence.

Xindong wu et al.[2] designed a "data mining framework" which works on three levels and has a three tier architecture. The tier-1 deals with the kind of computational platform that has the required data accessing and processing power which is needed to perform various type of analysis on Big Data sets. To achieve this, concepts like cluster computing and parallel computing are used. Many companies like Teradata are working towards developing such systems. These systems can harness the information present in data sets like e-mails, images and various logs which is otherwise difficult to achieve through traditional approaches like relational databases and can build on the information gained using such systems and data sets.

The tier-2 is formed by the various challenges that occur due to the distributed information sources in case of Big Data and privacy issue of the information which belong to a person or is related to the internal working of an organisation. Any Big Data system needs to take into consideration the various regulations regarding the usage and disclosure of different kind of data. Authorisation of data access and concepts like non-disclosure of the identity of the individual should be the consideration in the development of any Big Data System. Tier-2 also deals with making a comprehension about the domain of the Big Data applications which is essential in formulating the targets of the process of data mining on the data of the application under analysis. The field knowledge will always help to design goals of the data mining process in an effective manner. This includes having an understanding of what the application is intended to do and establishing the use of the information. Without the correct insight of the application field it is difficult to finalize parameters which will be used to measure the usefulness of the results.

The tier-3 of the framework deals with issues of designing and redesigning of algorithms that can facilitate the data mining process on data sets that are large, distributed, complex and in most of the cases incomplete. At individual sources a narrow mining process can be undertaken and then these local derived information can be collaborated to form the general picture of the results. As described above the data sets are not usual in Big Data hence methods have to be improved or enhanced to handle the complexities.

Xindong Wu et al.[2] have listed a few of the undergoing researches in building Big Data System for specific applications around the world and have also mentioned work done on the different levels of their framework by many of the organisations and individuals. US National Science foundation and Australia Science Council have funded projects that work on bio data and handling streams of data for real time processing, respectively.

2.3 Analysis of Clustering Techniques

Adil Fahad et al.[3] have provided a study on various clustering algorithms by dividing them into groups and have examined their performance for various data sets. The main purpose of this paper is to put the numerous clustering algorithms into groups which can be then used as a reference for future application on a specific data set as the categorization is done on the basis of the performance of algorithms on different data sets. So in future when a user needs to perform clustering on a specific data set of an application then by comparing the features of that data set with the various parameters used in this work to evaluate the different methods, one can decide on which algorithm will work better on the given data set.

Adil Fahad et al.[3] have meticulously formulated the parameters for which the clustering techniques are measured for their performance. The parameters are designed to test the effectiveness of a particular clustering algorithm in handling the various challenges that

invariably crop up whenever Big Data Analytics is discussed. The research paper forms a clear need of why clustering is so important and an in demand topic in Big Data analysis. The ability of clusters to capture the relative similarity in the member data objects and to give an abstract view by providing a model to very large and complex data sets proves to be very handy in case of Big Data. The descriptive of Big data which are obviously the amount of data and multiplicity and diversity of the data objects have been formed as the parameters by selecting such Big data sets that match the aforesaid properties of big data.

The parameters that are formulated here and the approach that has been used to study the clustering algorithms can be used to examine future algorithms which are yet to be devised. The algorithms have been categorised and from each category a representative is selected on which various data sets are used as inputs and the resulting clusters are used as input to various metrics that are designed to measure particular aspect of algorithm with respect to the data which can be as simple as the relative time taken to process the entire data set to as complex as, measuring the quality of clusters and accuracy of the results achieved.

A total of eight parameters have been considered and five algorithms of different categories of clustering have been selected to perform the analysis. By varying the size, dimension and other aspects of a data set the given eight parameters have been measured for these algorithms using various metrics. Alongside the size and dimensions in a particular data set, the other parameter used is the number of values that the algorithm requires beforehand. The smaller the number simpler it is to implement that function. Also the type of data sets the function supports is considered while evaluating and the algorithm which supports variety of data sets for example if a function supports numerical as well as categorical values and text values then that algorithm of clustering will be rated higher than the one which only supports numerical values as the attributes of data objects. Some algorithms show abnormal behaviour when the data sample has some noise or some outlier values. An appropriate algorithm for large and diverse data sets should show robustness against the noise and should be able to handle such values. The evaluation done in this paper also encompasses the above discussed aspect. Stability of a clustering algorithm can be defined as the tendency to produce same results for

data set even when the order in which the objects that occur in the input is changed. Adil Fahad and other researchers [3] have considered the stability of an algorithm as well. Last but not the least time complexity of the candidate algorithms has been taken into account because if an iterative process is adopted than algorithms that are slow to give the results will perform poorly on a large data set.

The above mentioned criteria are used to evaluate the algorithms. The work done by Adil Fahad and others divided the entire clustering into five types and an algorithm from each type was selected to represent that class of algorithms and then further analysis on each selected algorithm was performed. A comprehensive learning of each function was done and the shortcomings and strong points were formed. These algorithms were then used to perform clustering on some selected real data sets. Then for each parameter a metric was defined and the results produced by each function were used as input to get various indices and values to find which algorithm performed relatively well for which criterion.

2.4 Map Reduce Paradigm

Jeffery Dean and Sanjay Ghemawat [4] proposed a programming paradigm named as “MapReduce” which realises the dispensation of data sets that are large enough to exceed the computing resources provided by a server in terms of processors, memory and storage capacity. The approach was first implemented by these two computer scientists at Google.co where a platform was built to support this model of programming and the data at the servers of Google was processed under this scheme enabling fast processing of distributed data which would otherwise took multiple times more communication and processing time. The system was built keeping in mind that an obvious way to fasten the process of data analysis over data set distributed among several servers or nodes in a network of servers is by paralleled execution of tasks in a manner that guarantees smooth data access over a distributed environment in a robust and reliable setting where failure recovery and fault tolerance is high.

Jeffery and Sanjay made an interesting observation that the main task in an analysis can often be broken down into two or more sub-tasks where each of these sub-tasks can independently be run on different machines and output of one sub-task which is performed simultaneously on different computing machines can become the input to the other sub-task which again can be executed by different processers simultaneously, basically achieving a distributed computation over distributed data.

In Jeffery and Sanjay[4] the "MapReduce model accepts a collection of key-value pairs which are the input to phase-1 known as the map task which again gives output in similar pairs to phase-2 called the reduce task which gives the final output again in pairs".

The work of this paper also include a detailed implementation of the above approach with considerations given to functions which will generate the pairs for the two tasks which can then be performed concurrently. The implementation gives due importance to failure recovery by introducing redundant tasks and a coordinating master task which maintains the status of all the distributed tasks on different machines. The coordinating task acts like a manager and distributes each of its workers, which in this case is the set of machines on which data is present, the tasks to be performed and maintains the overall system health.

2.5 Research Work on K-Means Algorithm

Data compression and data mining often involve various clustering techniques. K-means is one of the partition based clustering approach which was first introduced by Loyd SP[5] and Macqueen, J. [6]. In the algorithm provided by Loyd[5], centres of data points belonging to a cluster are calculated and these centres are updated according to the data points in the cluster. The steps are repeated in order to fulfil a stopping condition. In other methods the process is iterated till there is no change in the centres. Macqueen,J.[6] provided the in depth analysis of k-means problem by providing mathematical proofs and lemmas about the behaviour of k-

means. This research work also high lightened the various applications and advantages of k-means.

[7] Identifies the limitation of k-means by observing the dependency of the final result on the chosen initial centres. Various approaches to improve the k-means regarding the initial selection are proposed in [8], [9]. [8] uses the concept of aligning the data points along an axis and then choosing the seeds for the k-means. The result is more optimal than home or local. [9] uses an heuristic approach to find the total solution to the k-means problem. This approach is not dependent on the starting seed points in the procedure as the data points are not selected randomly, as in common method, but are calculated in the procedure. David Arthur and Sergie Vassilvitskii [10] introduced a new concept where after choosing one data point at random other centres are chosen in accordance to their probability which is weighted by their distance from the already chosen data points as centres. Wesam and Fyfe [11] analysed the tendency of k-means algorithm to give clusters that have local maxima and proposed three improvements to the algorithm, namely " weighted, converse weighted as well as converse exponential."

The vulnerability of k-means to data points that do not follow the usual trend, in other words outlier data points, has been researched and alternatives given in V.E.Castro and J. Yang [12].

S.Gupta, K Rao [13] and Z. Huang [14] extended the application of k-means to categorical data sets. The original k-means is able to handle data points that have only numeric values as their dimensions. The work in [14] provides a way to use k-means approach to cluster data points which also have dimensions of discrete or category type and not just numeric values.

2.6 Clustering Algorithms on Data Streams

The modern applications generate data in streams and therefore a stable and universal data mining scheme should be able to identify clusters in data which is not available in whole at once but comes at a rate and is to be processed accordingly.

Work done by Aggarwal et al. [15] focuses on the fact that traditional clustering algorithms are one pass algorithms which take the entire data stream as input are not efficient and do not produce stable results in conditions where the data objects in the stream change noticeably over time. It divides the entire process in two phases. The first phase takes the data stream input and interacts with the stream and produces the so called summaries of the stream at various intervals of time and then k-means is applied by the user on the summaries at time instances where he is interested to know the clusters. The algorithm maintains two types of clusters. One are formed by taking every data point in the stream. These clusters are multiple times in number than the final partitions aimed at the entire data stream. The second type of clusters which are the final clusters are formed using the first type of clusters. The algorithm provides details of how to maintain these clusters and how to choose the time and which the summaries or the first type of clusters are to be formed and stored so that a user can infer clusters from any time period from the time when data stream originated to the present time.

Ackermann and others proposed another approach to cluster data streams in [16]. This method used the idea of Arthur [10] to build the part of the algorithm that stores the summaries of the data points in the data stream and updates the summary with every data point input. The algorithm uses a special data structure that represents the data stream to the current point which is used to approximate the clustering of data stream by performing clustering on this core-set data structure. The reason for using this approach is to make the algorithm suitable for data sets with high dimensionality. The summary data structure is maintained and updated with each incoming data point. At the end k-means clustering is performed on the maintained summary of the data stream to get the final clusters.

In both the approaches data stream is first approximated but the difference lies in how this is done. The difference is due to the target of each approach as the first one wants to capture the nature of a data stream where as the second one wants to perform clustering in a fast manner for data points which have greater number of dimensions.

These classical methods require the predefined input of number of desired clusters. In many situations it is not possible to know beforehand the number of clusters. Also these approaches generally have two phases one online which interacts with the data stream and the other is offline where the final clustering is performed.

Modern approaches have been introduced which aim to perform clustering on the data stream in one phase and which does not require the predefined input of the number of total final clusters.

Silva, Hruschka and Gama [17] have provided an evolutionary algorithm which is used to perform clustering on data streams. They focused on developing an algorithm which does not require the prerequisite of the number of clusters. The algorithm also differs from the traditional methods as it does not have two phases but produces clusters as the data stream is being processed. The algorithm maintains the clusters and checks their relevance by performing certain tests to take into the account of changing nature of the data stream over a period of time. The process starts with estimating the number of clusters form a given size of initial data points from the stream. These cluster centres are then updated according to the incoming data stream. A periodic test is performed to monitor the changes in the data stream and if the stream has changed enough then new clusters are formed using the old clusters which are still relevant and the new data points which are the part of the changing data stream. Evolutionary algorithmic approach has been used to generate clusters with varying number of partitions.

Another approach is that of Hyde, Angelov and Mackenzie [18] which is inspired by the approach where clusters of random shape are formed. It uses a graph which links the close small clusters made on the data stream using the density based partition of the data objects of the stream. The main clusters are obtained by the graph which is maintained online. Major cluster is formed by the links between the multiple small clusters. Each small cluster is updated with each new data point and then decisions are made using predefined values like the distance of the point from the centre of the mini cluster and the distances between the two

mini clusters in order to update the graph by adding an edge which means the two clusters are part of the same main cluster or to delete the edge which will signify that the two clusters are part of different main partitions or even that a new mini cluster has to be formed which is further checked for its contention of being a major cluster or just an outlier cluster. Joined nodes in the graph of the mini clusters form one major cluster.

3 System Development

3.1 Analysis

3.1.1 Platform Architecture for Big Data:-

Systems used to implement any kind of processing including data mining methods on large data sets should be capable of handling all the obstacles that crop up when processing of such large data sets is undertaken. [2] Introduces a conceptual framework which can be used to form the base of a platform of a Big Data System.

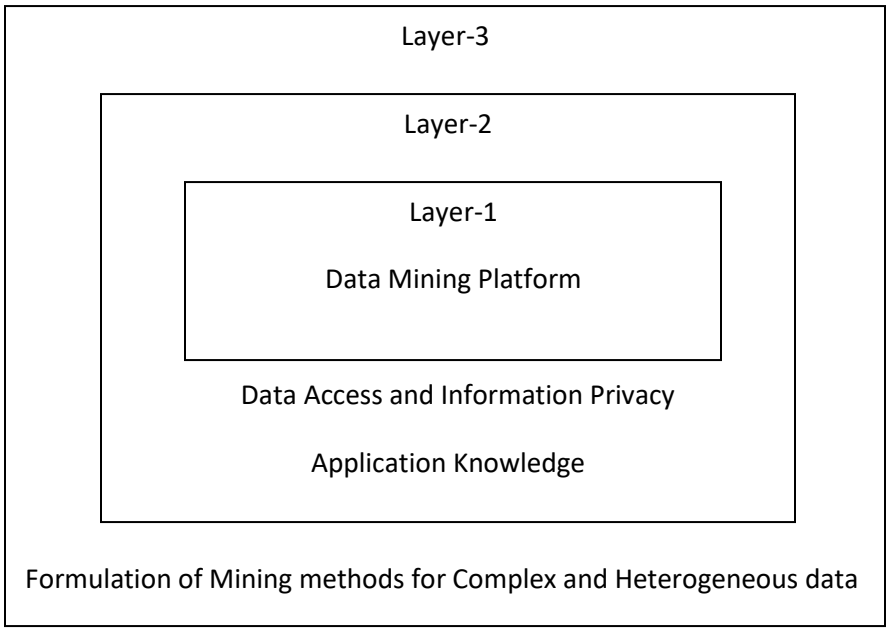


Figure-1:-Layered Architecture Framework for a Big Data System

Layer-1 of the framework deals with the computational aspect of the system. The processing power required for processing large and complex data sets cannot be matched by a single server and even if a super computer is used the delay due to the holding of data bus for the input-output operations from the disk to the memory. A parallel processing using distributed system makes sure that memory and processing power requirements are met. Hence, an efficient system would be build only by using the concept of distributed computing. Thus, at

this level of the architecture designing of a distributed environment where processing and data accessing is possible in a simple and robust fashion.

Layer-2 of the framework deals with challenges of data privacy and authorised access of the data. When dealing with distributed data a major issue is to get the authorised access to data which is at a far location from the access point. The differences in the protocols of different regions or organisation can pose different challenges to get the access to data sets which when combined are used by the system to perform certain processing. Apart from the data access another issue when inter-organisation and/or personal information are involved is of the privacy. Rules that maintain the rights of non-disclosure of some personal information or some trade policies or intellectual property or anonymity have to be strictly followed. So the system design for such Big Data applications should take in to the account all these parameters while designing the data distribution and use.

Application field knowledge should be there to design effective goals of the mining process. The success of the processing of Big data depends on how well the outcomes of the data mining process compliment the original objectives of the application.

Layer-3 focuses on designing mining methods that are able to handle the complexities of Big data. Approaches like knowledge fusion where data mining is done at local sites and then through the exchange of locally derived information to form a general model of the retrieved information.

Big Data applications often deal with data which has high rate of uncertainty and incompleteness, in other words it can be said that the data involved is vague. While designing the Big data system, the data mining methods should be selected in keeping the above fact in mind. The mining methods should be able to handle noise. For many methods if the data has incompleteness and vast dimensionality then the performance deteriorates significantly. Smart approaches of transforming the data by reducing the number of attributes and using techniques to fill the missing values should be used in the system. For uncertain values mean and variance calculations can be done to transform the data into a form where its processing is possible by most of the available methods in the field of machine learning and data mining.

Furthermore, the methods used in the system should be able to capture the relations among various data objects in the set. Largely the data is divided into two forms structured and unstructured. Big data involves both the types with varying degree of each. Some data objects will be structured and will comprise of all attributes needed for a particular data mining method like tables but there will be other objects like images which potentially have more

intrinsic information. Methods have to be devised in accordance to the data that the application is dealing with and this is handled in layer-3.

3.1.2 Hadoop Architecture:-

ApacheHadoop is an open source framework scripted and implemented in Java which allows the distributed processing of large tasks with huge memory and processing power requirements in sets of computers using models of easy and straightforward programming. An application operates in a setting that allows the storage and the computation distribution between the nodes in clusters of computing machines where the number of machines can range in hundreds and can be miles separated from each other. It is designed to be scalable to the order where it can be used from one single server to thousands of machines, each offering a computation and the local storage.

The major component modules of the entire frame work are:

• HadoopCommon: These are Java libraries and utilities required by other modules of the frame work. These libraries provide system level abstractions and contains the Java files and scripts necessary to implement and launch HadoopSystem.

• HadoopYarn: It is responsible for the scheduling of the processes in a job and the management of the resources of the machines in the cluster.

• Hadoop DistributedFileSystem (HDF): It has all the properties of a typical distributed file system. A distributed file system that is transparent for the end application during an access to application data.

• HadoopMapReduce:This the paradigm used in parallel processing of data stored in the distributed system and is tool-based..

These components should be studied in detail to gain some insight to this frame work for storing and processing big data sets.

HadoopFileSystem :- The system is developed using the Distributed File System. It directly interacts and runs on hardware. The system was designed to achieve high availability and reliability of the data stored in this file system using feasible tools and equipments.

It holds very large quantities of data and facilitates the access. To store enormous data, the files are divided in order to store on multiple machines. Redundancy is also maintained to safeguard the system from the potential of losing data in case of an undesired event such as failure of a critical process.

The main descriptive of the HDFS are :

• The concurrent access to data by storage distribution.

• The master slave model with namenode and datanodes of servers that assist users to effortlessly maintain the system and monitor its status.

• The data security feature provided by using permissions and authentication.

• Master known as namenode comprises of the middleware items which contain the base operating system and other set of programs and software to implement the functionality of the namenode. Due to the support of the system it can directly interact with the hardware.

• The machine in the distributed system which has the model of namenode is the master server and it performs the duties of managing the entire file system by keeping the mounting information of the entire directory tree. It manages the information regarding the names and paths of all the files in the systems. Any file related operation like reading, writing, creating a new file, deleting an old file or changing permission of a file are performed by the namenode.

• The slaves are known as datanodes which are the machines that are capable of storing data and performing the duties that are given by the namenode like transferring a block of data, basically input-output operations. It also comprises of the set of software that enables it to become compatible by becoming a member of the system and managing the stored data locally as per the guidelines conveyed by the master server.

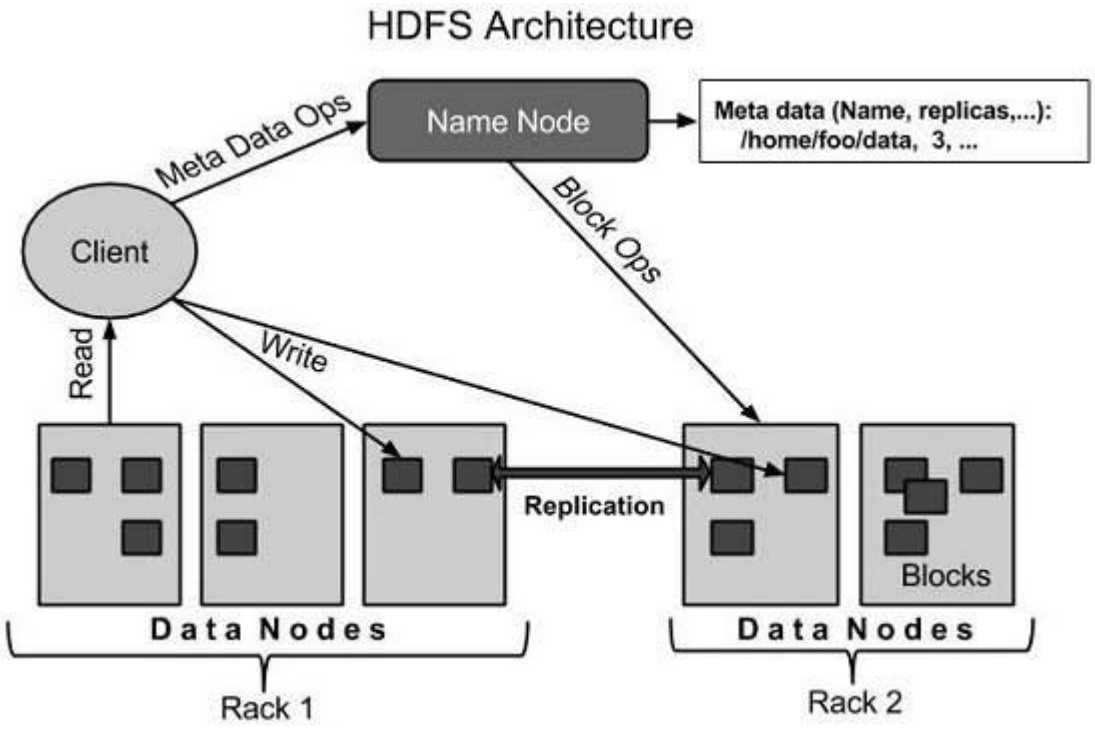


Figure-2:-Architecture Framework for HadoopFileSystem

• The application data which is stored in the files of HDFS by dividing into one or several segments or chunks of fixed sizes with each chunk being stored in nodes according to the adopted or given configuration. These are called blocks of the file. This the smallest unit in the system for all the file operations in the system as a minimum of one block can be read, write or transferred in the system. In other words, the minimum amount of data that HDFS can read or write is called a block. Default size has been set as hundred and twenty eight bytes but can be varied according to the requirement of the application.

• HDFS includes a large number of product hardware and because of it component breakdown is recurrent. Consequently, this file system has methods for quick look for error detection and automatic recovery with the help of redundancy maintained by the system.

HadoopMapReduce :- It is a software base that facilitates developing and writing applications that handle large amounts of data in parallel on a distributed system. The main role of this platform is to provide an application developer with an interface with basic equipment in a robust and reliable system, with fault tolerance.

The term MapReduce has the following two different tasks that the programs should have:-

• The map task: It is the first task, which takes the input data and converts it to a set of data, where the individual elements are divided into tuples.

• The reduce task: This task takes the output of map as input and combines these tuples of data in a smaller set of tuples. Reduce task is always performed after the Map task.

Commonly both the input and output are stored in the distributed file system. The framework deals with the resource management and scheduling as well as planning of tasks, supervise and watch them and managing the tasks that failed.

This software base is constituted of a singular master and numerous slaves. There are JobTracker for the system and TaskTracker per node in the system. The master is the manager of the resources, following the consumption of resources and their accessibility. The TaskTracker executes and performs the tasks as given in the directions of the master and provide information on the status of task to the master periodically.

The paradigm is based on the dispatching the program to the location of the data residence.

Stages in the model are map, shuffle, and reduce step.The mapper or map has the role of performing the charge to process and deal with the input data. The input data is obviously in the form of file or directory and is stored in the distributed file system. The file is passed to the map function object wise or line wise. Analyses of the data and creates several small pieces of data. The next step is the blend of shuffle and reduce step. The reducer has the task to process and attend to the data that originates in the Map scheme.

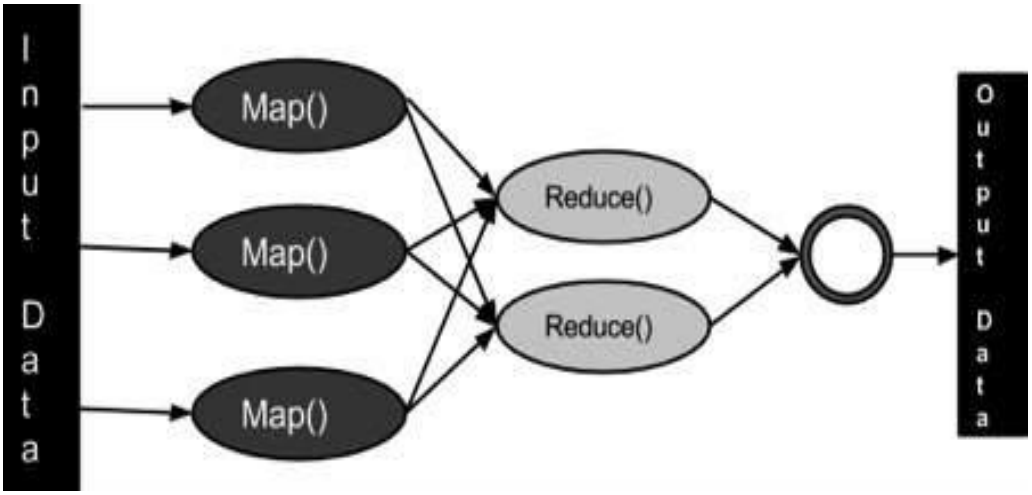


Figure-3:-Schema of MapReduce

The Framework MapReduce works on (key-value) Pairs.

	Input	**Output**
Mapper	<key1, val1>	list (<key2, val2>)
Reduce	<key2, list(val2)>	list (<key3, val3>)

It is the job of the Record reader to transform file objects into lists of pairs of key and corresponding value. Map takes these pairs as input and produces similar output. Reduce works on the these output of the map and give final pairs which is considered the final output.

3.1.3 Streaming Via Flume:-

ApacheFlume can be considered as an instrument or a means for the compilation and aggregation and shiping large quantities of flow of data such as log files, the events (etc...) from various sources to a data store centralized.

It is a steadfast resolution for a distributed system for transporting data and is programmable instrument. It is configured to direct the flow of data from connected servers which may be in a private network or connected via the internet in a particular order to the supporting distributed file system.

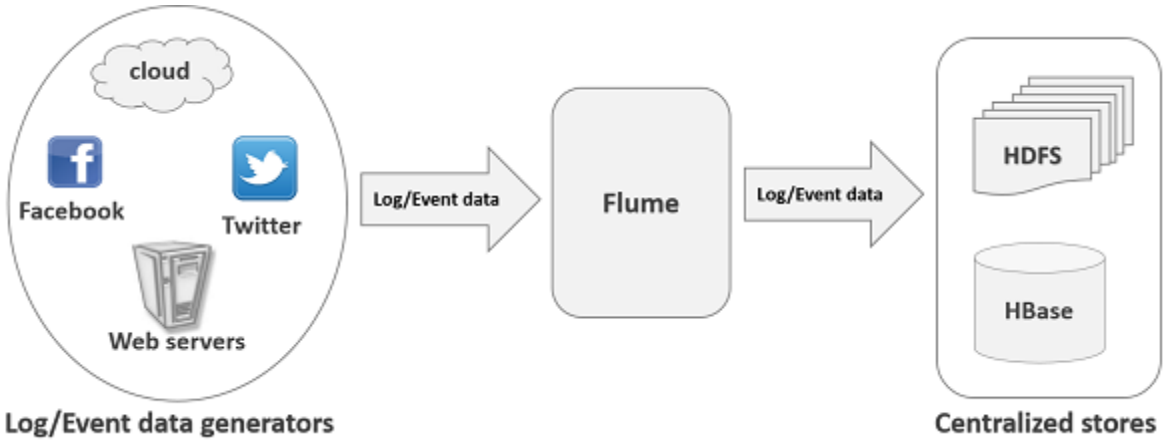


Figure-4:- Path of data

Arhitecture :- The architecture of the ApacheFlume can be understood by the figure-5. Servers on which Big data applications are run and which generates data which is then perceived by the agents of the canal that they use. Subsequently, an agent known as the data collector collects the data from the other agents which is then cumulated and driven into a the distributes system where further action is done on the data.

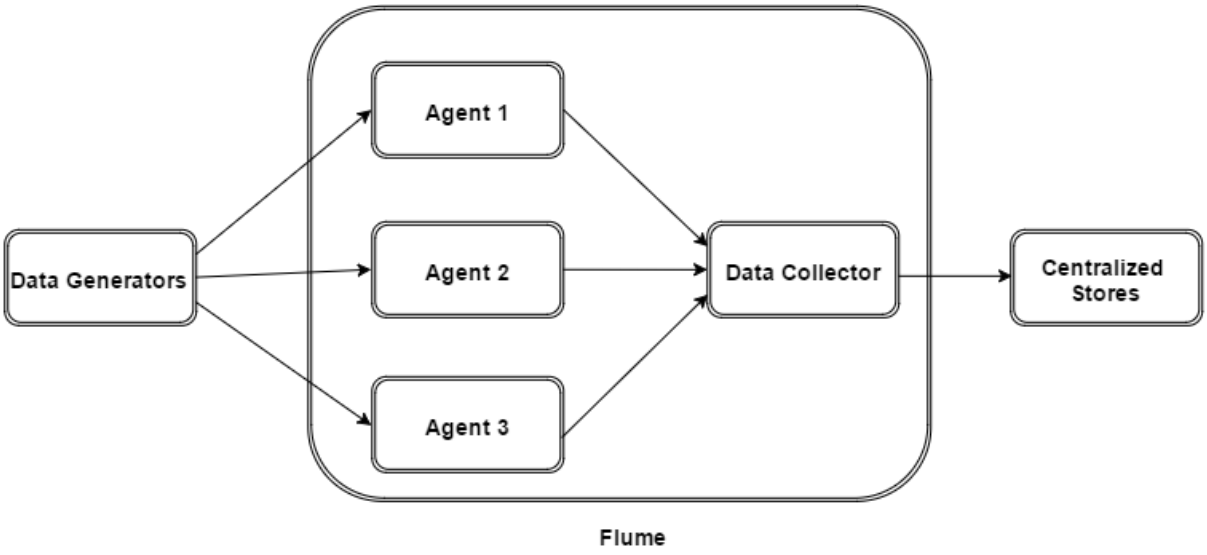


Figure-5:-Architecture of flume

Flow of Data: Usually various events and records that form the data are generated by the machines and these machines have agents of the flume which are used by them. These agents receive the data from these machines known as the data generators. The data of these agents will be collected by an intermediate node. Here is no restriction in the number of collectors being one in fact similar to agents, even multiple collectors are possible.

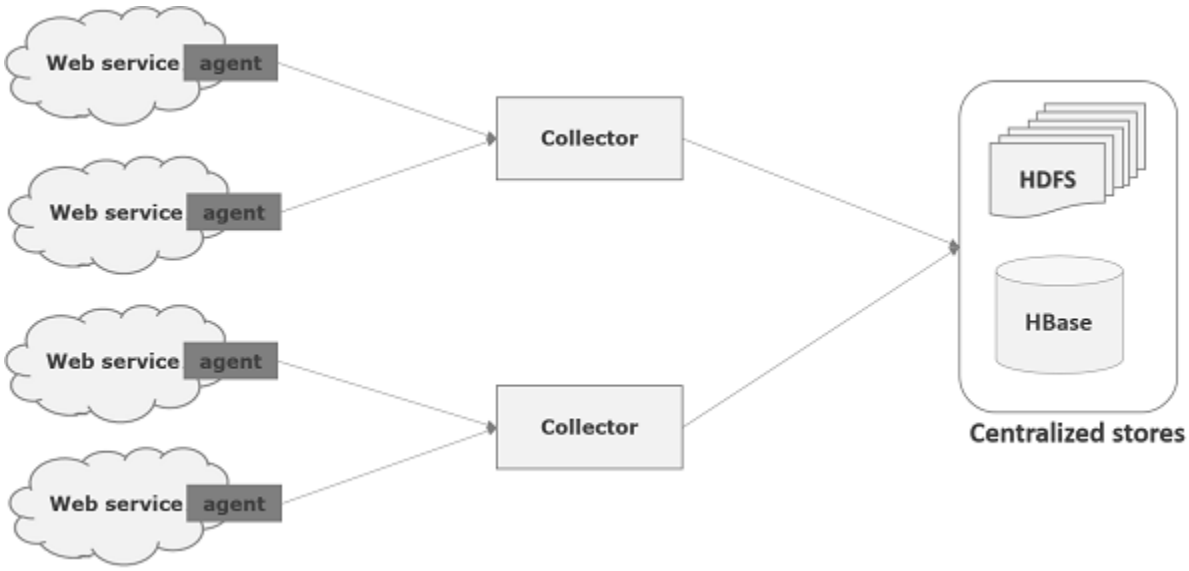


Figure-6:- Data Flow in Flume

3.1.4 Data Clustering Technique:-

Clustering is seen as an unsupervised learning technique which can be used to find the latent and implicit classification or grouping in data which would otherwise is impossible to visualize. The technique of clustering has proven to be very useful in extracting the hidden structure in big sized data sets. Clustering is sometimes used in summary making of large datasets where this summary is further used in other data mining processes. Whereas in other applications major work is to identify relations among data points and clustering achieves this by grouping the data points together on the basis of their relative distance to each other, density distributions or similarity due to other reasons like common ancestral data objects.

One technique of clustering is not suitable for every data set. Different algorithms support different type and size and nature of data sets. Therefore, it is very important to have the knowledge of all types of clustering and to select the appropriate clustering for the given data.

We have selected k-means algorithm as it is suitable for large numeric data provided some improvements are undertaken. The data sets adopted by us are all numeric.

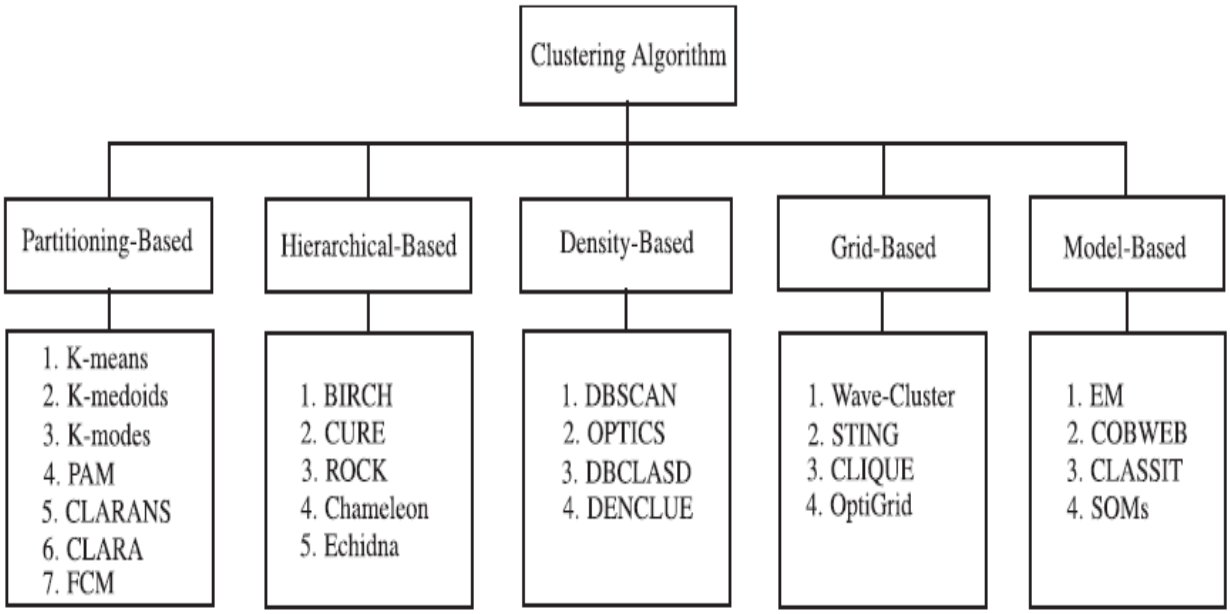
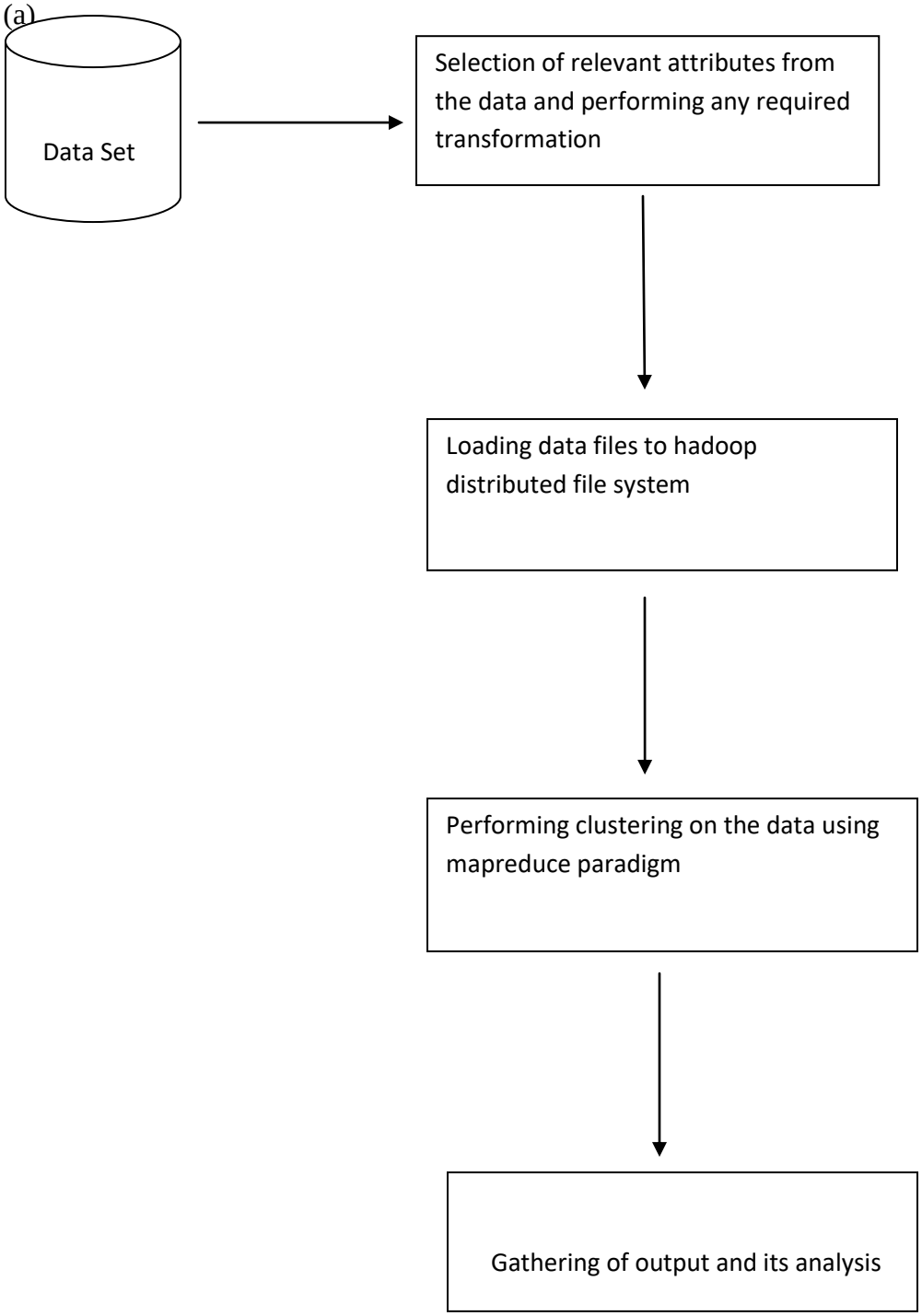


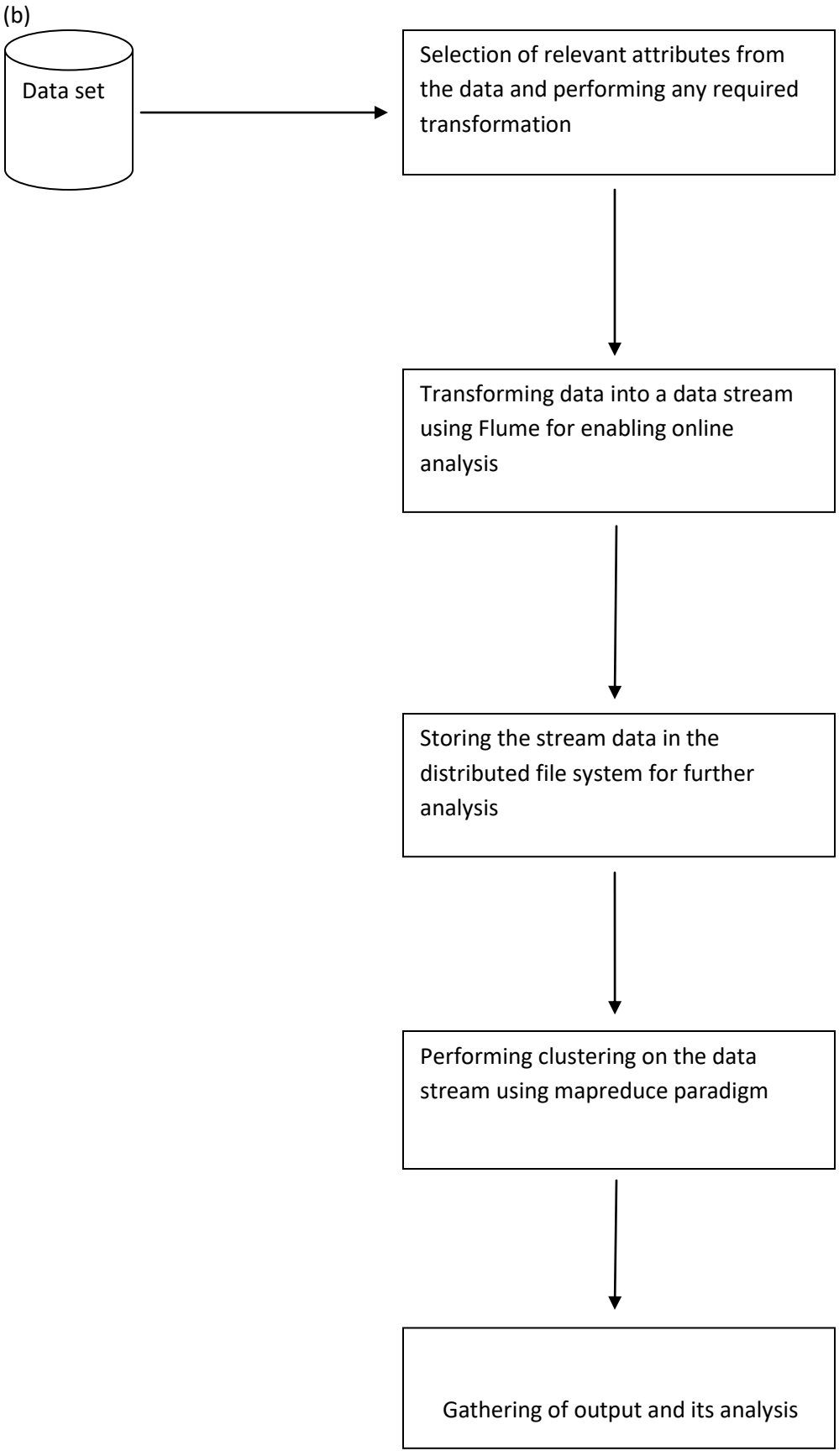
Figure-7:-Clustering Cataloguing

3.1.5 System Model:- First one to handle normal data and next for data stream

(a)



(b)


Data set
Selection of relevant attributes from the data and performing any required transformation
Transforming data into a data stream using Flume for enabling online analysis
Storing the stream data in the distributed file system for further analysis
Performing clustering on the data stream using mapreduce paradigm
Gathering of output and its analysis

3.2 Algorithm

3.2.1 K-Means Algorithm:-

K-means algorithm solves the problem of clustering by partitioning the data space into k number of regions where k is an input parameter used by the algorithm. In a space where each point, say p belongs to the data sample where each point has, say d number of dimensions. The algorithm takes k which is the number of clusters to be performed as an input and then associates each data point to a cluster using a distance metric which is usually Euclidean Distance. The following is the description of each step of the algorithm:-

Step-1:- Input the value k, which is the number of desired clusters.

Step-2:- Randomly select k number of data points and label them as c_1, c_2,…..,c_k.

Step-3:- These selected points are the initial centers.

Step-4:- For each data point in the sample calculate the distance of each center using a distance metric. The common one is Euclidean distance. For two points a and b, each of d dimensions there distance can be measured as-

$$D = \sqrt{((a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \ldots + (a_d - b_d)^2)}$$

Step-5:- For each of the distances calculated for each single data point find the minimum distance from a center and add the point to the set of points which is closest to this center.

Step-6:- After every data point's distances have been measured for each of the centers and every point has been allocated a centre calculate the new centre for each set b taking the mean of all the values of points present in center.

Step-7 :- Step-4 to Step-6 are repeated until a stopping condition is not met. Usually the condition is to get the minimum sum of squares. The sum of squares is the sum of squares of distance of each data point in a cluster from its respective center. If n is the total number of points and C_i is the set of all points belonging to a cluster with center c_i , then the sum of the squares of the entire data set is calculated as-

$$S = \sum_{i=1}^{i=k} \sum_{\forall p \in C_i} | D(p - c_i) |^2$$

Where $D(p - c_i)$ calculates the distance between p and c_i.

3.2.2 Advanced K-Means Algorithm:-

The previously mentioned k-means algorithm works fine when data is limited to small sizes but when large datasets are used the time complexity increases manifold as the number of data points increase and the times when distance computation is needed. Thus for large data sets some changes need to be performed on the algorithm. During the implementation of this algorithm on a platform which supports map reduce paradigm of computing mapper and reduccer functions are to be designed.

The important concept is to locate the steps in the k-means algorithm which are independent and can be executed simultaneously to speed up the processing in of the data set. The data is distributed in the distributed file system in the cluster of nodes capable of processing the stored data and the clustering job is submitted to the master node and the sub-job of mapping and reducing are given to the slave nodes. The step where for each point distance is calculated fro each center can be made parallel as the data is distributed in various nodes, the map job will be to calculate the distances and that of the reducer will then calculate the new centers.

The mapper and reducer functions are described below:

(A) KmeansMapper(input_key, input_value, output_key, output_value)

```
Point p = input_value
Double min= Max_value_of_Double
for each center, c from the shared list of centers

    d = distance between p and c

    if d < min then

        output_key  = c

        min = d

    end if

  end for

 outputCollecter ( output_key, input_value )

end function K-meansMapper
```

The generation of input key and the value is the function of the record reader from the input file present in the distributed system. The input key is the offset value of a particular data point with respect to the start point of the file. The contents associated with one key in the file is the input value. Every framework provides with a system to collect the results from the map function at each local site in the distributed setting. Also, methods to transform data from one type to another.

The result from the first phase is shuffled and sorted according to the keys and then this is fed to the various trackers. The final combination of keys and values is the output of the reduccer phase of the algorithm.

(B) KmeansReducer(input_key, input_value, output_key, output_value)

```
Double sum = 0
Integer i=0
    for each point p in the list L from input_value
        sum = sum+ p
        i++
    end for

 point new_c = sum/i // where i are the number of values.

  Output_key  = new_c

  Output_value = Input_value

  Update  the list of centers by adding the new centers, new_c

 outputCollecter ( output_key, output_value )

end function K-meansReducer
```

The output of the reducer is the new center and list of points that form the cluster. After reducer tasks the closing criterion is checked and if the condition is met then the process is terminated. If the condition is not fulfilled then the map and reduce work in an iterative manner till the condition is met. This algorithm divides the most heavy computational task in k-means that is calculating distances for each point multiple times into sets of concurrent tasks. Hence speed up is achieved when slave nodes work only on a part of data and not on its entirety.

4 Performance Analysis

4.1 Analysis on K-means

Experimental runs on k-means algorithm were conducted by performing clustering on real data sets with varying size and dimensionality.

The purpose of these tests is to verify the accuracy of k-means for numerical data sets. Also, to observe how different methods of initializing the starting centers known as seeds perform on data sets with varying sizes in order to propose a method that is suitable to handle large data sets.

For this task we implemented clustering by k-means on the Matlab (Matrix Laboratory) platform which is the product of The Math Works Inc.[19]. The input data is converted into matrix form where each row is new data point and each column represents a data field. Matlab has functions to perform k-means algorithm and also offers some flexibility by providing various options with it with respect to the distance metrics, number of iterations, ways of choosing the initial data points, etc.

4.1.1 Checking Effectiveness:-

The first task was to test the effectiveness of k-means in producing clusters that are relevant to the intrinsic nature of the data set. For this purpose we choose Iris plant data[20] set from UCI machine learning data repository which contains data about the readings of three different types of iris plant, namely Setosa, Versicolor, Verginica. Each data point had four attributes, Sepal length, Sepal width, Petal length, Petal Width . All were numeric.

The purpose is to find the optimal number of clusters using k-means and verify it with nature of the data set. To check which particular formations of clusters has good quality we use the silhouette value of each of the distribution as the metric to test it.

Silhouette value ranges from -1 to 0 to 1. Closer to 1 means that point has best cluster fit and a value closing to 0 means that the point can belong to multiple clusters. A negative value means that the point has been allocated to a wrong center and should be in another cluster. A more efficient way is to take the average of silhouette values of the entire distribution. This way comparison among different distributions is easier.

(A) Number of clusters = 2

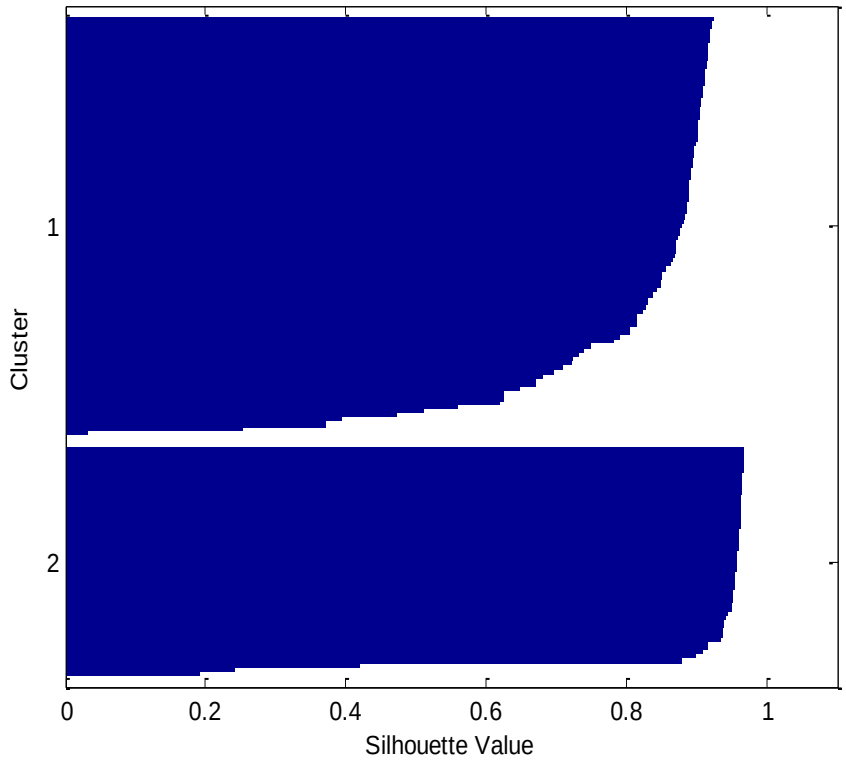


Figure-8: Silhouette plot for iris data with two clusters

Average number of Iterations = 4.8

Best total sum of distances of each point to its center = 152.369

Mean of Silhouette value = .8502

(B) Number of clusters = 3

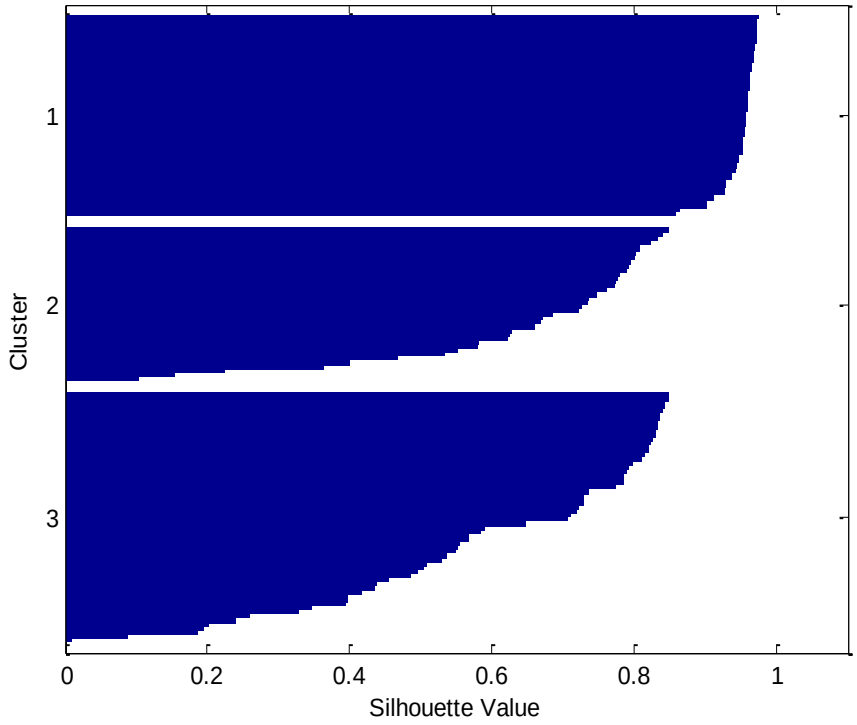


Figure-9: Silhouette plot for iris data with three clusters

Average number of iterations = 5.6

Best total sum of distances of each point to its center = 78.9408

Mean of Silhouette value = .7355

(C) Number of clusters = 4

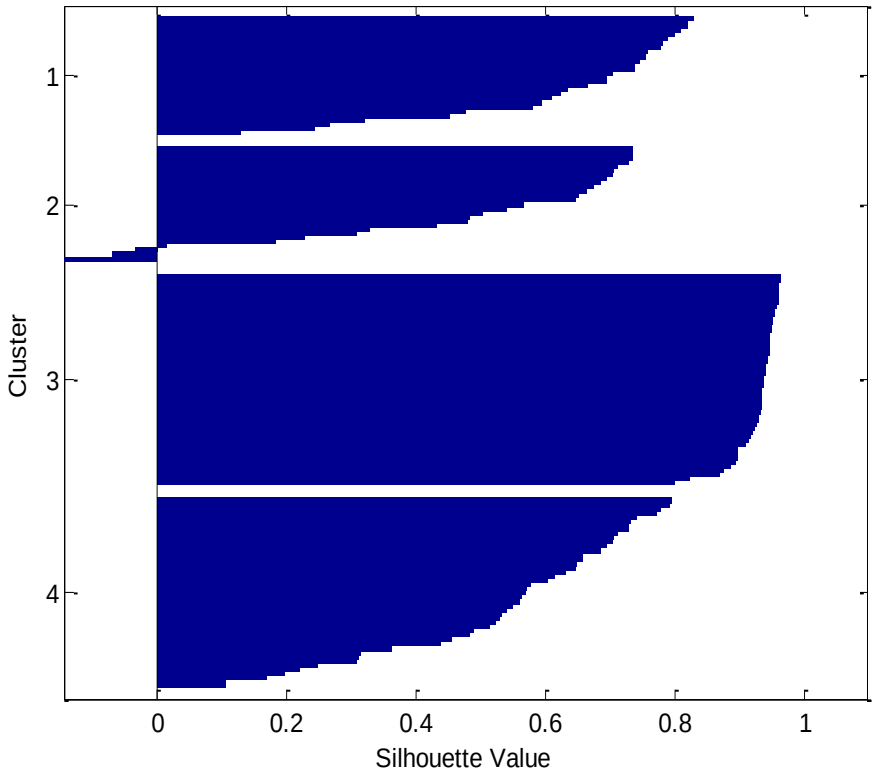


Figure-10: Silhouette plot for iris data with four clusters

Average number of Iterations = 10.8

Best total sum of distances of each point to its center= 57.355

Mean of Silhouette value= .6766

(D) Number of Clusters = 5

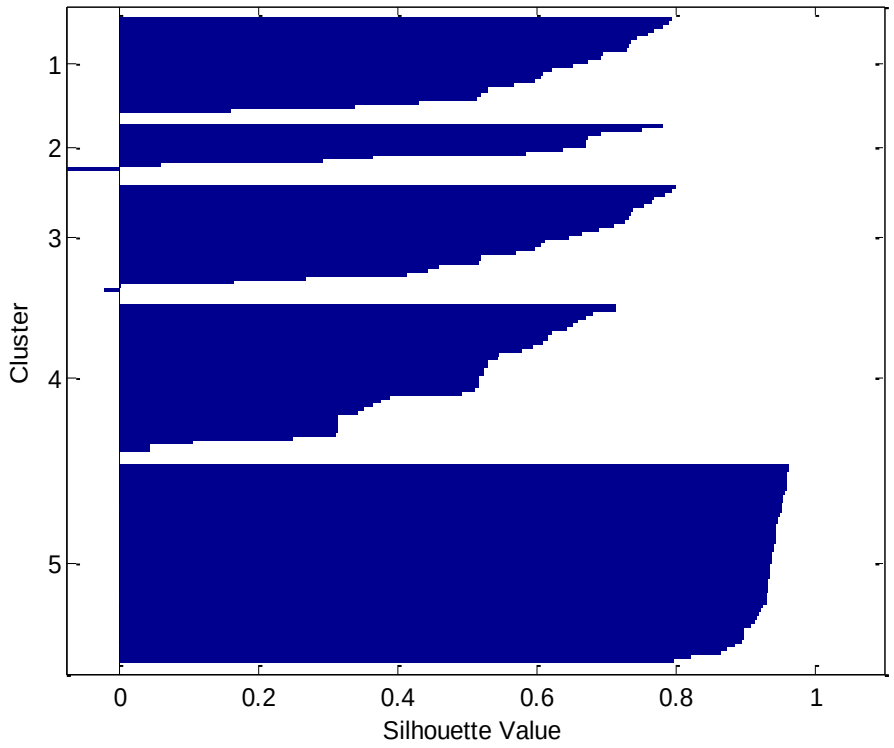


Figure-11: Silhouette plot for iris data with five clusters

Average number Of Iterations = 12.2

Best total sum of distances of each point to its center = 46.5506

Average Silhouette value = 0.6700

Result:-

From the above analyses it is clear that for the iris data set the optimum number of clusters are two to three. This result is supported by the intrinsic nature of the iris data as one kind of iris plant is linearly distinguished from the other two types, hence clustering gives best result with

two clusters as the solution. Later, with three clusters as the aim one cluster remain almost unchanged and the larger clusters gets split into two, thus indicating that the data has certain level of classification in it. With increasing the number of clusters, the silhouette values of the solutions start to decrease indicating excessive clustering.

The average number of iterations increases with increase in number of clusters. This is obvious because with increase in number of clusters the processing is increased and finer level of clusters can be achieved by higher number of iterations of the kmeans on the data.

The total sum distances of the points to their centers decreases with increase in number of clusters because the points get closer to their centers.

4.1.2 Analyzing Seeding Methods:-

The next task is to analyze the methods used to initialize the centers in the k-means clustering algorithm. Many researches have found that the results produced by k-means are sensitive to the initial centers selected. The problem of local minima occurs because results show different minimum sum of squares of distances with different initial centers. Therefore, it is important to choose that method for initialization which gives result that are close to the absolute solution.

For this task we have selected the modified glass identification data set [21] from the UCI repository. The set has nine attributes which are continuous numeric values. Attributes are the amount of Sodium, Magnesium, Iron, Aluminium, Silicon, Potassium, Calcium and Barium. It also has the value of the refractive index of the glass samples.

To analyze the sensitivity of the K-means algorithm towards the initial starting centers, variations in the initialization of cluster centers was implemented on the glass identification data.

(A) K-means with random initialization:

(centers are chosen randomly in each replication)

Replicate 1, 7 iterations, total sum of distances = 0.647805.

Replicate 2, 6 iterations, total sum of distances = 0.995446.

Replicate 3, 8 iterations, total sum of distances = 0.995446.

Replicate 4, 7 iterations, total sum of distances = 0.630937.

Replicate 5, 7 iterations, total sum of distances = 1.01399.

Replicate 6, 7 iterations, total sum of distances = 0.705404.

Best total sum of distances = 0.630937

Average Number of Iterations = 7

Average Total Sum Distance= .8315046

Elapsed time is 2.5322 seconds.

Mean Silhouette value = .6007

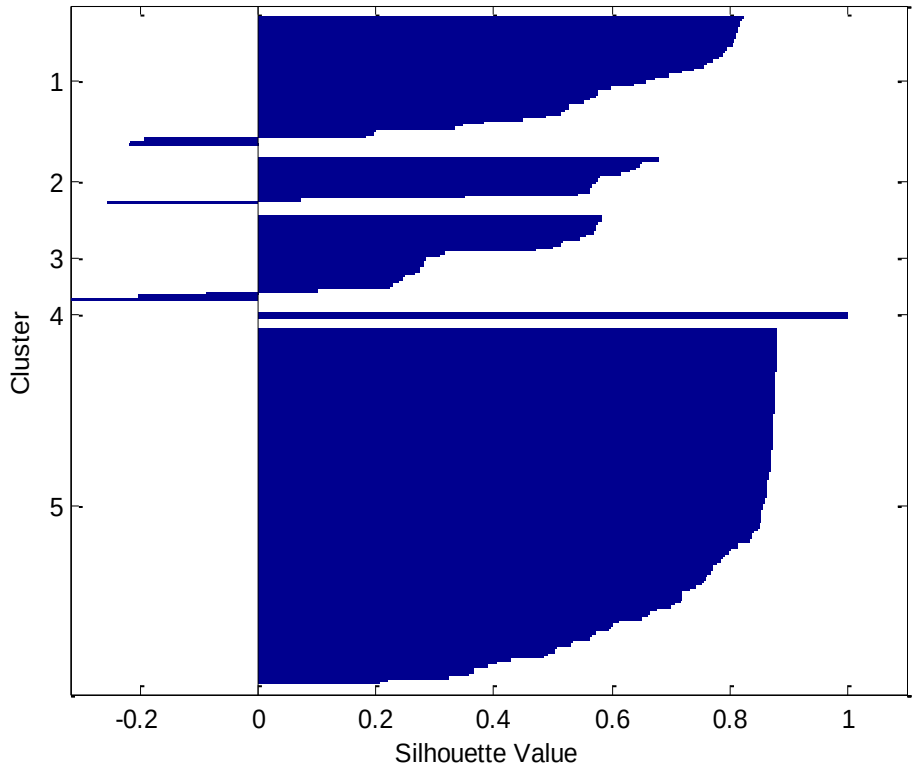


Figure-12: Silhouette plot for glass data with random initialization

(B) K-means with K-means++ initialization:

Replicate 1, 7 iterations, total sum of distances = 0.616404.

Replicate 2, 11 iterations, total sum of distances = 0.705404.

Replicate 3, 10 iterations, total sum of distances = 0.705404.

Replicate 4, 7 iterations, total sum of distances = 0.705404.

Replicate 5, 15 iterations, total sum of distances = 0.61655.

Replicate 6, 8 iterations, total sum of distances = 0.995446.

Best total sum of distances = 0.616404

Average Number of Iterations = 9.66

Average Total Sum Distance= .724102

Elapsed time is 1.831596 seconds.

Mean Silhoutte value = .7405

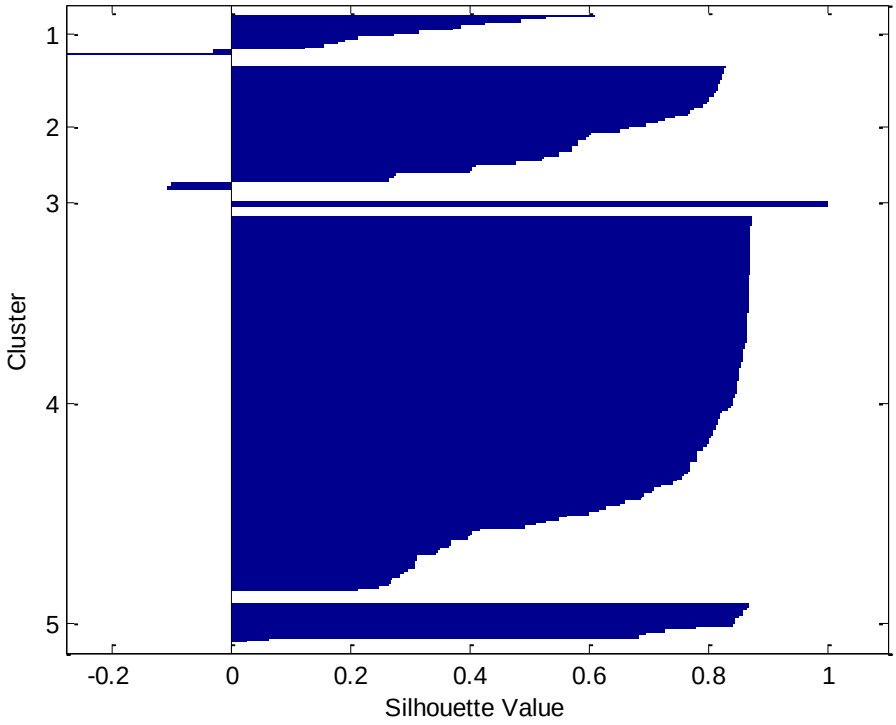


Figure-13: Silhouette plot for glass data with K-means++ initialization

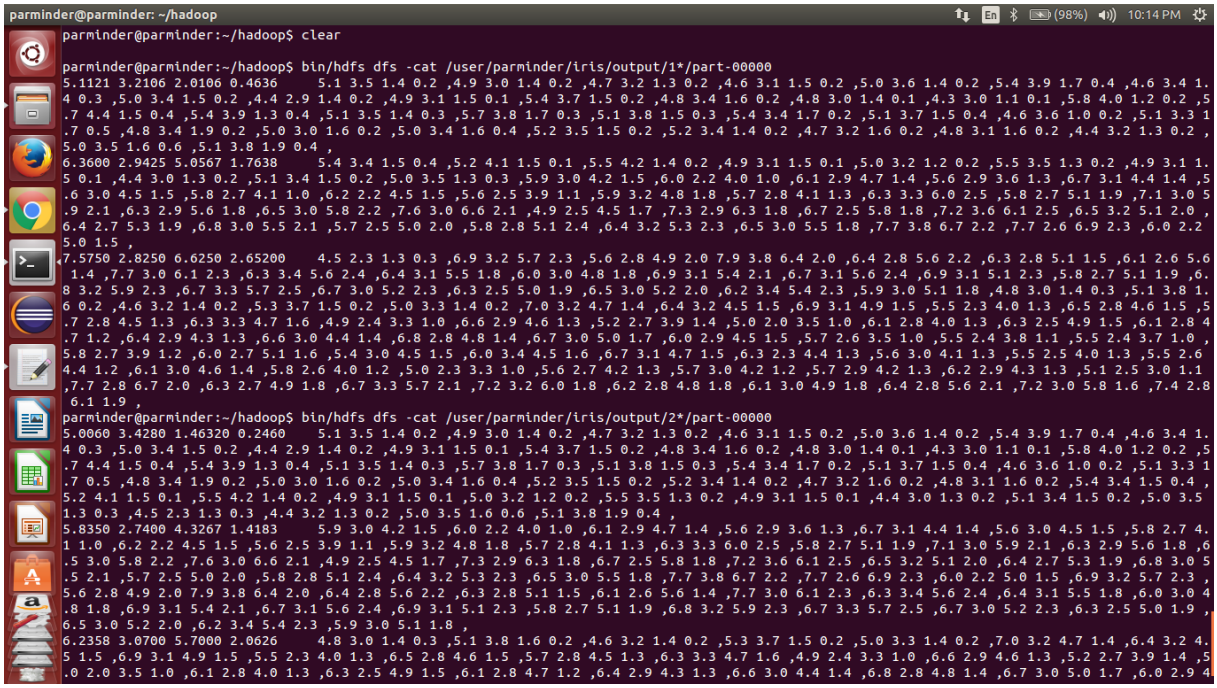
Result:-

The plus method outperforms the random initialization for data sets with large values. But for small data sets random initialization is the preferred method. Hence, for future use of k-means for large data sets alternative methods like k-means++ should be used when the cluster quality to be achieved is more and overhead due to extra computation is not an issue but where fast results are required random initialization can be adopted or these alternatives have to be designed to be more efficient.

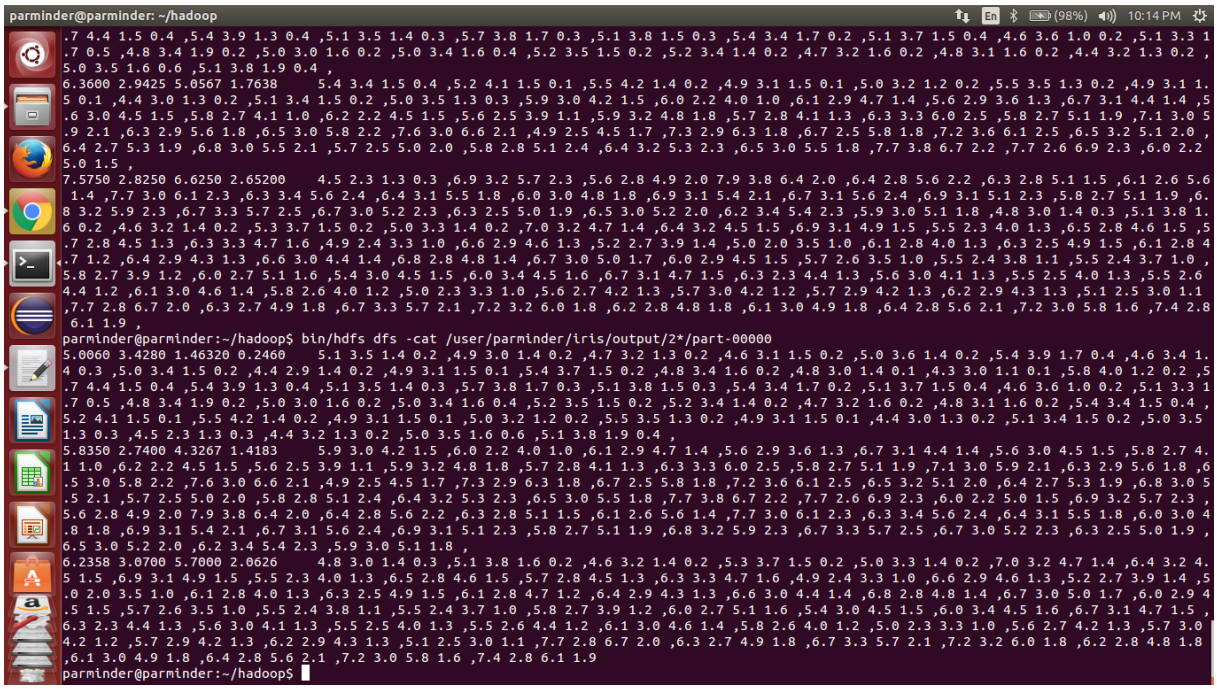
4.1.3 Advanced K-means Implementation:-

K-means implemented using map reduce was run on data sets. Due to the concurrent calculation of distances there will be some reduction in the overall time in case of the advanced version. Here are the screen shots for the output of the one data set run using MapReduce model.

(a) Intermediate Output:-



(b) Final Output:



5 Conclusion

We have analyzed the process of information retrieval from data and how this approach differs for big data because of the associated challenges with it. It has been realized that traditional methods cannot deliver while dealing with Big data and therefore techniques like clustering become more important which can help in making some sense out of the large distributed data.

We focused on data sets which have continuous numerical valued attributes and K-means algorithm was realized to be a potentially good algorithm because of its simple implementation. Various features of the algorithm are analyzed by running it on data sets and finding the implicit facts about the algorithm.

A parallel version of the clustering algorithm is developed and implemented on the hadoop frame work and was tested with some real data sets.

Not a lot has been done in the field of clustering data streams using mapreduce model. We build a system that could handle a data stream and can store it in the distributed file system where programs supporting mapreduce can be run on the stream producing analysis of data streams that are too large to be handled by a single stand-alone machine. By doing this we can apply various data mining methods which were earlier applied only to the bulk data. We implemented clustering on data stream by using the developed parallel version of k-means algorithm.

There is an immense scope of developing a clustering algorithm which can handle categorical data also and supports the mapreduce model in future. Observation of alternative seeding method in k-means performing better for large data sets produces an opportunity to further improve the algorithm and extend its applicability.

References

[1] Fayyad, U.M., Piatetsky-Shanpiro, G., Smyth P., Uthurusamy, R.: "Advances in Knowledge Discovery and Data Mining." AAAI/MIT Press, 1996.

[2]Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding:" Data Mining with Big Data", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 1, JANUARY 2014.

[3] Adil Fahad, Najlaa, Zahir , Abdulaah , Ibrahim, and Bouras: "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis", IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, Date of publication 11 June 2014.

[4] Jeffrey Dean and Sanjay Ghemawat: "MapReduce: Simpli_ed Data Processing on Large Clusters", Google.Inc, OSDI 2004

[5] Lloyd SP "Least squares quantization in PCM."IEEE Trans. inform, Theory (Special Issue on Quantization), vol. IT-28, pp 129 – 137 Mach 1982. 24

[6]MacQueen, J.: "Some Methods for Classification and Analysis of Multivariate Observations." In Proceedings Fifth Berkeley Symposium Mathematics Statistics and Probability. Vol. 1. Berkeley, CA (1967) 281-297.

[7] Xu, Rui and Donald Wunsch II. Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, Vol., 16, No. 3, May 2005

[11] Wesan, Barbakh And Colin Fyfe. "Local vs global interactions in clustering algorithms: Advances over K-means." International Journal of knowledge-based and Intelilligent Engineering Systems 12 (2008).83 – 99

[12] Estivill-Castro V. and J. Yang, "A fast and robust general purpose clustering algorithm." In Proc. 6 th Pacific Rim Int. Conf. Artificial Intelligence (PRICAI`00), R. Mizoguchi and J. Slaney, Eds., Melbourne, Australia, 2000, pp, 208 – 218

[13] Gupata S., K. Rao, and V. Bhatnagar, "K-means clustering algorithm for categorical attributes", in Proc. 1st Int. Conf. Data Werehousing and Knowledge Discovery (DaWak`99). Florence, Italy, 1999, pp. 203 – 208.

[14] Huang, Z., "Extensions to the k-means algorithm for clustering large data sets with categorical values.". Data Mining Knowl. Discov., vol. 2, pp. 283 – 304, 1998.

[15] Aggarwal, C.C., Han, J.,Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases, vol. 29. pp. 81–92. VLDB '03, VLDB Endowment (2003).

[16] Ackerman, Martens, Raupach, “StreamKM++: A Clustering Algorithm for Data Streams” ACM Journal of Experimental Algorithmics, Vol. 17, No. 2, Article 2.4, Publication date: May 2012.

[17] Silva, Hruschka, Gama :- “ Evolutionary algorithm for clustering data streams ” Expert Systems With Applications.

[18] Hyde, Angelov, Mckenzee : “Fully Online Clustering of Evolving Data Streams into Arbitrarily Shaped Clusters” information Sciences volumes 282-283 march 2017 pages 96- 114, 2016

[19] https://archive.ics.uci.edu/ml/datasets/Iris

[20] https://archive.ics.uci.edu/ml/datasets/Glass+Identification