Chapter-VI
Evaluating Data Mining as a Service under Cloud Environment

6.1 Introduction

These days, data mining solutions are needed for most of the organizations irrespective of their size or volume of operations. However, solutions demand high end resources requiring large investments in IT infrastructure. These solutions are therefore not affordable for small organizations. With the growing size of an organization, the size of accumulated data also grows exponentially. To improve the performance, mining may require distributed computation of data. But, distributed data mining is a complex system and incorporates many components including algorithms, communication systems, resource management, task scheduling and user interface design. Since every organisation cannot afford to have costly solutions, there is a need to develop an efficient and affordable distributed data mining system that monitors mining of huge data with the use of already existing computing resources.

After analysing different distributed computing scenarios, it was found that cloud can be an affordable solution for distributed data mining. Using the concept of virtualization and cloud, existing data mining techniques can be enhanced to give better performance. Cloud computing is a type of distributed computing system that provides on-demand access to a pool of computing resources like networks, servers, storage, applications, and services and is known for its higher efficiency, usability, resource utilization with lower cost. Thus, a cloud solution for distributed data mining is proposed. Keeping in mind the security of data, a private cloud setup is created that facilitates distributed computing. In this private cloud setup, the resources are shared as a service within private network. A cost effective data mining solution is also derived using this private cloud setup that supports distributed data mining and can handle voluminous data. For distributed data mining however, there is a need to increase or decrease the number of computational resources depending on the size of data to be mined. Hadoop a free,
open source programming framework that supports distributed processing of large
data sets is therefore chosen for experimentation. This chapter presents a solution
that will help to evaluate distributed data mining in cloud computing environment.
Using this solution, data mining can be made cost effective and affordable.

6.2 Motivation

Data mining techniques can be made efficient using parallel processing on GPUs.
As specified in chapter-V, compaction and parallel processing on single GPU
resulted in multi fold improvement in the performance of data mining algorithm.
Similar approach can be implemented on a cluster of multiple GPUs also to
further improve the performance. Setup of GPUs and clusters is expensive and
requires large number of dedicated resources. So, instead of using dedicated
hardware for distributed data mining, an alternative approach can be used to
utilize existing set of resources.

Distributed computing is necessary when the data size exceeds user allocated size
memory. Cloud based tools can provide an efficient infrastructure for distributed
computing. Cloud computing is a specialized form of distributed computing where
the underlying resources, such as storage, processors and memory are completely
abstracted from the consumer. Cloud vendor is responsible for the reliability,
performance, scalability and security of the service. Limited work is carried out to
evaluate data mining in cloud environment. Therefore, cloud is chosen as a
platform in this study. An initial study was carried out to compare the cost of infra
structure for public and private cloud. It was identified that private cloud solution
for distributed data mining was more economic as compared to other solutions.

To evaluate the suitability of data mining algorithms on cloud, and to justify its
affordability, a distributed data mining solution is proposed.

6.3 Methodology

The initial study was to investigate about different products and companies
providing the facility for distributed computing in cloud environment. Then the
freeware cloud solutions were configured to enable mining in distributed
environment. This setup takes into consideration the computing and storage
resources already available in an organisation. It is more flexible and can deliver data mining at a lower cost.

The methodology followed is as below:

- Comparison of the cost of private cloud with public cloud.
- A private cloud environment to be configured using existing resources.
- Design of data storage as a service.
- Design of data mining as a service.
- Evaluating performance of data mining as a service on cloud.
- Analysis of results.

6.4 Cost Effective cloud solution for Distributed Data Mining

Private cloud setup was configured for evaluating the performance of data mining. For accumulating enterprise level data, cloud service for data storage is used and for computation of mining algorithms, instead of using high end servers, the existing system are configured so that they work as a powerful server. Service oriented architecture is also incorporated. A parallel data mining algorithm is developed and provided as a service. Here, cloud based tools and techniques are used to provide services for distributed data mining. For optimizing the performance, two services are used:

i. Data Storage as a Service

ii. Data Mining as a Service

6.4.1 Data Storage as a service

OwnCloud [8] an open source file synchronization solution is used to integrate the data from different sources on private network. Administrator has full control over the ownCloud Data. New users can be created, deleted and permissions can be granted through ownCloud. Other users can mine local data files through the interface from any of the computers on the private cloud.
It is a software system, which provides the facility of Storage as a Service. It is accessible from any place and any device like desktop, notebook, tablet and smartphone. It provides features like automatic backup, easy sharing, extended storage, customization and several other features. It helps to share several types of files over the network. Owncloud is a software system mainly used for file hosting. It has no limits on storage space and the number of connected clients. But, hard disk capacity may result into some limitation in terms of storage.

In order to synchronize files on desktop machines with their ownCloud server, desktop clients are also available. These clients are available for PCs running on Windows, OS X, or Linux. Main features of ownCloud include file storage, cryptography, synchronization of clients, task scheduling, user and group administration, sharing of content across groups or public URLs and connecting external storages.

6.4.2 Data Mining as a Service

Hadoop [86] supports MapReduce which is a distributed programming model intended for processing massive amounts of data in large clusters. MapReduce is mainly intended for large clusters of systems that can work in parallel on a large dataset. It has simplified the implementation of parallel data mining applications. MapReduce can be applied to various data mining applications.

An extension of K-Means is developed so as to increase its applicability in distributed applications. Hadoop MapReduce can be extended for the analysis of large datasets. In this solution, Hadoop is used to run parallel K-Means algorithm which is provided as a service on private multinode setup.

Map Reduce [87] is implemented as two functions, Map ( ) which applies a function to cluster the data on and returns local results based on local nodes. Reduce ( ), collects the results from multiple Maps and gives consolidated final clusters. All Map ( ) can run in parallel, on multiple machines at the same time. To manage storage resources across the cluster, Hadoop uses a distributed file system. Distributed data mining is implemented by leveraging the advantages of Hadoop.
6.5 Software Requirements

- Operating System: Ubuntu 12.04 or later
- Runtime Environment: JDK 1.6 or later
- Open Source: Hadoop 1.2.1
- Other: SSH Server and Client

6.6 Hardware Requirements

- Processor: P IV or higher
- RAM: 2GB or more
- Space on Disk: minimum 10 GB

6.7 Parallel K-Means Clustering Algorithm

K-Means follows a simple way to classify a data set and assign them to K clusters. Here, K is fixed a priori. The main idea is to define k centroids, one for each cluster. Actual K-Means algorithm runs in following sequence [113]:

- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- Assign each object to the group that has the closest centroid.
- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups.

The parallel version of k-Means Clustering algorithm is based upon Hadoop MapReduce. From the data set, the parallel k-Means clustering algorithm reads some sample and computes the centroids of the cluster. Each mapper reads these centroids via a centroid server running on master node. Each reducer can write the
computed centroid via the same server. Parallel K-Means Clustering algorithm consists of five classes:

i. **KMeansDriver**: The class **KMeansDriver** is the main class and drives the whole execution. It is the main class that iterates the execution no of times specified by user. It also checks for the convergence. If convergence occurs it breaks the loop and prints the total time taken to execute the program. When initialized, it takes sample values from the input file and does the centroids computation task. These centroids are passed to the centroid server as initial centroids so that each mapper can take these centroids and perform mapping. Reduce task computes the new centroid and writes to the centroid server. By using the reference of centroid server main class checks for the convergence and decides the flow of execution.

ii. **KMeansMapper**: The split files are received by **KMeansMapper** and the computes local clusters. It performs mapping on specific splits. It reads initial centroids from centroid server and computes the Euclidean distance. On the basis of Euclidean distance it decides nearest centroid from the input record and writes output accordingly.

iii. **KMeansReducer**: This class has a reduce function that computes the new centroid for a group of records. After computing the new centroid, it connects to the centroid server and writes the new centroid to the output steam of the server. Also it forms the cluster of the group assigned to it.

iv. **DoubleArrayWritable**: The input record is in the form of text. Since parallel k-Means Clustering algorithm is executed on three dimensional points so, numerical value is needed to compute centroid. This class has a vector that stores the dimensions of the point. It has two methods readFields() for reacing the dimensions from any input stream and write() for writing it to any output stream.
v. **CentroidServer**: KMeansDriver starts the Centroid server by passing initial centroids. After starting the server, control is transferred to KMeansDriver. It has three methods namely Compare( ), Read( ) and Write( ). Compare( ) method compares the new centroids with old centroids and if convergence occurs it returns true otherwise returns false. Read( ) and Write( ) methods read from reducer and writes to the mapper. Before termination of the application, it is stopped by the KMeansDriver.

### 6.8 Configuring the cluster

Hadoop cluster composed of one master and 3 slave nodes is set up for experimentation. The steps to configure the cluster are specified below:

1. First of all generate public key on master node and replicate it on slave nodes so that communication between master and slave nodes can happen.
2. Configure Hadoop on each node by giving the path of JAVA_HOME environment variable in hadoop-env.sh. Make necessary changes in hdfs-site.xml and core-site.xml file residing in conf directory of Hadoop.
3. Run the script start-dfs.sh found in the directory HADOOP_HOME/bin. It will start the NameNode on master and DataNode on each slave.

### 6.9 Input File Format

Currently the experiments are performed on numeric data. The file assumed is of the marks of students. The K Means clustering will group the students into K clusters each having nearest objects. Different file sizes ranging from 50,000 to 2,50,000 were used for experimentation.

```
223 37 222
111 159 198
414 440 271
430 203 325
278 274 194
```
6.10 Interface Design for Distributed Data Mining

The interface is designed in Netbeans-8 with Java 7. Apache Hadoop 2.3.0 is used for creating multi node setup and for running mapreduce jobs. Single system configuration is:

<table>
<thead>
<tr>
<th>CPU</th>
<th>Core2duo 2.93GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>2 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu 64-bit</td>
</tr>
</tbody>
</table>

Multi node setup of computer systems is created. The setup includes Single node setup, 2-node setup and 3-node setup through which the system resources re increased. This setup can be accessed from any other node in the network.
Analyst can apply clustering on the data that is available through i. Owncloud and ii. User’s own System files. This distributed data mining solution can be accessed from any node on the private network.

The interface works as follows:

i. Enter into the interface of Data Mining as a Service.

ii. Two types of user can access the interface: Admin and other users. Admin can access data from Owncloud.

iii. Admin can select the file to mine from Owncloud whereas other users can upload a file from their system.

iv. Input file size to get a prediction about appropriate number of nodes to select for mining.

v. As per the prediction, choose the number of nodes on which you want to process the data.

vi. Press the process key and wait for results.

vii. Output will generate clusters on the basis of K Means algorithm. It will also display the time taken for clustering.

Hadoop needs lot of manual configuration to create a multi node setup. A user friendly interface is therefore developed to execute data mining on specified number of nodes that doesn’t require any user level configuration and handles the setup process at backend automatically. Hadoop is configured over private servers and clients can execute parallel K- Means through common interface from anywhere in private network.

For experimentation of Data storage as a service, academic institute’s data files are accumulated on ownCloud. Data to be mined can either be chosen from cloud data server or can be uploaded from private computers on the network. Folders of users were synchronized with ownCloud. Files to be mined are automatically stored on ownCloud since they are synchronized. Required files can be selected and can be provided to K- Means algorithm which runs in a distributed fashion.
through Map Reduce. If the data size is manageable and the results to be obtained are of departmental level, user can directly mine departmental level data from individual nodes using the interface. But when the data size is very large and is of whole enterprise, ownCloud service can be used for storage.

6.11 Experimental Results

Data for clustering was selected from files of different sizes and the execution time was noted for single and multinode setup. For small sized files with number of records below 50,000, the performance of parallel k-Means Clustering algorithm on single node was best. But as the number of record grows, the performance of parallel algorithm gets degraded on single node and it is observed that for very large datasets, the performance on multi node hadoop setup keeps improving.

Table 6.1 Execution time for different no. of nodes

<table>
<thead>
<tr>
<th>Time taken for clustering using K-Means(in secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Records / No. of Nodes</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>2- Nodes</td>
</tr>
<tr>
<td>3- Nodes</td>
</tr>
</tbody>
</table>
Figure 6.1 Time plot (in secs.) parallel k-Means Clustering algorithm on single node, 2-nodes and 3-nodes

6.12 Proposed Solution

The proposed solution can be used to optimize the mining performance for different large size inputs. It also ensures the security of data since the data remains on private network only. For the mining of organizational level data, ownCloud can be used whereas for mining personal data, individual files can be uploaded from anywhere on the private network. The proposed solution is helpful in processing large size data in less time and is less costly as compared to other data mining systems.
Figure 6.2 FDMPC: Data Mining as a Service on Private Cloud
6.13 Summary

An organisation can avail the benefits of cloud by installing a private cloud using the existing resources. It gives benefit of huge availability of resources in low cost. The integration of data mining techniques with cloud computing can allow the users to extract and mine useful information from a cloud based storage and mining service. Public cloud is accessible through internet and brings more threats to the security of the organization's data. So, instead of public cloud, a secure private cloud solution for mining data is chosen. This work also proves that when the data size is very large and results in performance bottlenecks, a parallel data mining algorithm is an improved alternative. K-Means algorithm needs lot of iterations and its time complexity is very high. Therefore, a parallel version of K-Means algorithm is proposed that outperforms in clustering large data volumes.

On the basis of experimentation, a generalized solution that supports Distributed Data Mining and Storage as a service on private network is proposed. The main challenge handled by this solution is mining of voluminous amount of data in a heterogeneous distributed environment. Execution of distributed data mining technique using private cloud setup can allow the stakeholders to retrieve meaningful information from the large pool of data. The proposed solution presented a scalable distributed platform to manage large number of users working under private cloud environment. It is equipped with the necessary modules that support the fundamental features of database as a service and data mining as a service. Currently, the interface is limited to parallel K-Means algorithm only but can be extended for other data mining algorithms also.