

PROVIDING SECURITY AND CLUSTERING FOR BIGDATA IN CLOUD

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in

Computer science and engineering

By

Dhinakaran.J, B.Dinesh (38110128, 38110132)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL
OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH - 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Dhinakaran.J, B.Dinesh (38110128, 38110132)**, who carried out the project entitled "**PROVIDING SECURITY AND CLUSTERING FOR BIG DATA IN CLOUD**" under our supervision from November 2019 to April 2020.

Internal Guide

Dr.G.Mathivanan.,M.E., Ph.D,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **GOGU JAYA SHANKAR (Reg. No. 36110397)**, hereby declare that project report entitled "**PROVIDING SECURITY AND CLUSTERING FOR BIG DATA IN CLOUD** " done under the guidance of Dr.G.Mathivanan.,M.E., Ph.D, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

A handwritten signature in black ink, appearing to read 'G. Jayashankar', with a horizontal line extending from the end of the signature.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr. S. VIGNESHWARI, M.E., Ph.D.** and **Dr.L. Lakshmanan M.E.,Ph.D., Head of the Department, Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G.Mathivanan.,M.E., Ph.D,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways For the Completion of the project.

ABSTRACT

Most cloud framework applications contain important and private information, for example, individual, exchange, or wellbeing data. Dangers on such information may put the cloud frameworks that hold these information at high hazard. This paper proposes a coordinated approach to group what is more, secure large information before executing information versatility and duplication, what is more, investigation. The need of verifying large information is a must and should process .so on behalf of that we provide the security to the large sets of data before the clustering and then send them to the cloud .The effect of information security is considered and validated on the private information in the extent of Hadoop Distributed File System. It is uncovered that the proposed approach can improve cloud and the large sets of data before the clustering are frameworks information versatility.

TABLE OF CONTENTS

CHAPTER NO	TITLE	
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF ABBREVIATIONS	IX
1	INTRODUCTION	10
	1.1 GENERAL	10
	1.2 OBJECTIVES	10
	1.3 TERMS AND TERMINOLOGIES	11
	1.4 HOMOGRAPHIC ENCRYPTION	13
2	LITERATURE SURVEY	14
3	AIM AND SCOPE OF PRESENT INVESTIGATION	23
	3.1 AIM OF THE PROJECT	23
	3.2 SCOPE AND OBJECTIVES	23
	3.3 USE CASE DIAGRAM	25
	3.4 ACTIVATE DIAGRAM	26
	3.5 SEQUENCE DIAGRAM	27
	3.6 HADOOP	27
	3.7 RISKS INVOLVED IN CLOUD COMPUTING	28
	3.7.1 TECHNICAL ISSUES	28
	3.7.2 SECURITY IN THE CLOUD	28
	3.7.3 PRONE TO ATTACK	29
4	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED	30
	4.1 MODULES	
	4.1.1 EXTRACT TRANSFORM LOAD	30
	4.1.2 SYSTEM SETUP AND DATA INTEGRATION	31
	4.1.3 WEIGHTED K-MEANS CLUSTERING	32

	4.1.4 SINGLE ROUND MAP REDUCED PRIVACY- PRESERVING CLUSTERING	33
	4.15 IMPORT/EXPORT DATA USING SQOOP TOOL ANALYSIS	36
	4.2EXISTING SYSTEM	37
	4.2PROPOSED SYSTEM	37
5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	39
6	CONCLUSION AND FUTURE WORK	40
	6.1 CONCLUSION	40
	6.2FUTURE WORK	40
	REFERENCES	41
	APPENDIX	44
	A SOURCE CODE	44
	B SCREENSHOTS	50

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
3.1	USE CASE DIAGRAM	27
3.2	ACTIVITY DIAGRAM	28
3.3	SEQUENCE DIAGRAM	29
4.1	MAP REDUCE	38
6.1	USER INTERFACE	52
6.2	SQL DATABASES	52
6.3	JAVA SCRPT CODE	53
6.4	JAVA CODE SAMPLE OUTPUT	54
6.5	DATABASES	54
6.6	JOB EXECUTION	54

LIST OF ABBREVIATIONS

ADT	: Android Development Tool
ETL	: Extract Transform Load
LWM	: Learn With Mirror
HTTP	: Hyper Text Transfer Protocol
IP	: Internet Protocol
TCP	: Transmission Control Protocol

CHAPTER 1

INTRODUCTION

1.1. GENERAL

Clustering techniques have been widely adopted in many real world data analysis applications, such as customer behavior analysis, targeted marketing, digital forensics, etc. With the explosion of data in today's big data era, a major trend to handle a clustering over large-scale datasets is outsourcing it to public cloud platforms. This is because cloud computing offers not only reliable services with performance guarantees, but also savings on in-house IT infrastructures. However, as datasets used for clustering may contain sensitive information, e.g., patient health information, commercial data and behavioural data, etc, directly outsourcing them to public cloud servers inevitably raise privacy concerns.

over a 5 million objects dataset further validates the practical performance of our scheme. In this paper, we propose a practical privacy-preserving K-means clustering scheme that can be efficiently outsourced to cloud servers. Our scheme allows cloud servers to perform clustering directly over encrypted datasets, while achieving comparable computational complexity and accuracy compared with clustering over unencrypted ones. We also investigate secure integration of MapReduce into our scheme, which makes our scheme extremely suitable for cloud computing environment. Thorough security analysis and numerical analysis carry out the performance of our scheme in terms of security and efficiency. Experimental evaluation.

1.2 OBJECTIVES

A practical privacy-preserving K-means clustering scheme is proposed that in which the cloud servers perform clustering over encrypted datasets and it is extremely suitable for parallelized processing in cloud computing environment.

1.3 TERMS AND TERMINOLOGIES

While the term “big data” is relatively new, the act of gathering and storing large amounts of information for eventual analysis is ages old. The concept gained momentum in the early 2000s when industry analyst Doug Laney articulated the now-mainstream definition of big data as the three Vs:

Volume

Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would’ve been a problem – but new technologies (such as Hadoop) have eased the burden.

Velocity

Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.

Variety

Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

K-Means Clustering on Big Data

The k-means (Lloyd) algorithm, an intuitive way to explore the structure of a data set, is a work horse in the data mining world. The idea is to view the observations in an N variable data set as a region in N dimensional space and to see if the points form themselves into clusters according to some method of measuring distance. To apply the k-means algorithm one takes a guess at the number of clusters (i.e. select

a value for k) and picks k points (maybe randomly) to be the initial centre of the clusters. The algorithm then proceeds by iterating through two steps:

1. Assign each point to the cluster to which it is closest
2. Use the points in a cluster at the m^{th} step to compute the new centre of the cluster for the $(m + 1)^{\text{th}}$ step

Eventually, the algorithm will settle on k final clusters and terminate

1.4 Homomorphic Encryption

Homomorphic encryption is a form of encryption that allows computation on cipher texts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. The purpose of homomorphic encryption is to allow computation on encrypted data.

Cloud computing platforms can perform difficult computations on homomorphically encrypted data without ever having access to the unencrypted data. Homomorphic encryption can also be used to securely chain together different services without exposing sensitive data. For example, services from different companies can calculate the tax the currency exchange rate shipping, on a transaction without exposing the unencrypted data to each of those services.

CHAPTER 2

LITERATURE SURVEY

This chapter revises the most important aspects in how computing infrastructures should be configured and intelligently managed to fulfill the most notably security aspects required by Big Data applications. One of them is privacy. It is a pertinent aspect to be addressed because users share more and more personal data and content through their devices and computers to social networks and public clouds. So a secure framework to social networks is a very hot topic research. This last topic is addressed in one of the two sections of the current chapter with case studies. In addition, the traditional mechanisms to support security such as firewalls and demilitarized zones are not suitable to be applied in computing systems to support Big Data. SDN is an emergent management solution that could become a convenient mechanism to implement security in Big Data systems, as we show through a second case study at the end of the chapter. This also discusses current relevant work and identifies open issues.

The Big Data is an emerging area applied to manage datasets whose size is beyond the ability of commonly used software tools to capture, manage, and timely analyze that amount of data. The quantity of data to be analyzed is expected to double every two years (IDC, 2012). All these data are very often unstructured and from various sources such as social media, sensors, scientific applications, surveillance, video and image archives, Internet search indexing, medical records, business transactions and system logs. Big data is gaining more and more attention since the number of devices connected to the so-called “Internet of Things” (IoT) is still increasing to unforeseen levels, producing large amounts of data which needs to be transformed into valuable information[2012][1].

Big Data Applications with Scheduling becomes an active research area in last three years. The Hadoop framework becomes very popular and most used frameworks in a distributed data processing. Hadoop is also open source software that allows the user to effectively utilize the hardware. Various scheduling algorithms of the MapReduce model using Hadoop vary with design and behavior, and are used for handling many issues like data locality, awareness with resource, energy and time. This paper gives the outline of job scheduling, classification of the scheduler and comparison of different existing algorithms with advantages, drawbacks limitations. In this paper, we discussed various tools and frameworks used for monitoring and the ways to improve the performance in MapReduce. This paper helps the beginners and researchers in understanding the scheduling mechanisms used in Big Data.

Big Data plays very important role in many industries like healthcare, automobiles IT etc. Effective utilization of energy, resource, time becomes challenging task nowadays. Big Data has become more popular in IT sector, banking, finance healthcare online purchasing, engineering and many other areas. Big Data refers wide range of datasets and hence it's difficult to manage by existing applications. The data sets are very complex and growing day by day in humongous volume. Raw data are continuously generated from social media, online transactions, etc. Due to continuous increase in volume, velocity and variety complexity increases; it induces lots of difficulties and challenges in data processing. Big Data becomes a complex process in terms of correctness, transform, match, relates, etc[2016][2].

Big data phenomenon arises from the increasing number of data collected from various sources, including the internet Big data is not only about the size or volume. Big data posses specific characteristics (volume, variety, velocity, and value 4V) that make it difficult to manage from security point of view. The evolution of data to become big data rises another important issues about data security and its management. NIST defines guide for conducting risk assessments on data, including risk management process and risk assessment. This paper looks at NIST risk management guidance and determines

whether the approach of this standard is applicable to big data by generally define the threat source, threat events, vulnerabilities, likelihood of occurrence and impact. The result of this study will be a general framework defining security management on Big data.

Big data phenomenon is triggered by the rapid growth of various social network services. User generated content is responsible for generating a huge volume of data to be analyzed for many purposes, from business to security. Machine to machine communication (M2M) and the Internet of Things also produce a vast amount of data. Data from other fields, DNA sequencing, also contribute to big data. Big data implications in data analytics is significant, such as in management business [9], the data gathered from online conversations between members in a community can be used as a consideration for marketing strategy, supply chain management, customer relationship management, competitive advantage and business intelligent of a company. In informatics a new learning system approach for artificial intelligence using big data has already been generated. In research on astronomy NASA has also use Big data to support research to map stars formation on the sky[2014][3].

The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size. We describe the architecture of HDFS and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo!.

Several distributed file systems have or are exploring truly distributed implementations of the namespace. Ceph has a cluster of namespace servers (MDS) and uses a dynamic sub-tree partitioning algorithm in order to map the namespace tree to MDSs evenly. GFS is also evolving into a distributed name-space implementation. The new GFS will have hundreds of namespace servers (masters) with 100 million files per

master. Lustre has an implementation of clustered namespace on its roadmap for tree release. The intent is to stripe a directory over multiple metadata servers (MDS), each of which contains a disjoint portion of the namespace. A file is assigned to a particular MDS using a hash function on the file name[2010][4].

Big Data has become a hot topic across several business areas requiring the storage and processing of huge volumes of data. Cloud computing leverages Big Data by providing high storage and processing capabilities and enables corporations to consume resources in a pay-as-you-go model making clouds the optimal environment for storing and processing huge quantities of data. By using virtualized resources, Cloud can scale very easily be highly available and provide massive storage capacity and processing power. This paper surveys existing databases models to store and process Big Data within a Cloud environment. Particularly, we detail the following traditional NoSQL databases: Big Table, Cassandra, DynamoDB, HBase, Hyper table, and MongoDB.

The MapReduce framework and its developments Apache Spark, Hadoop, Twister, and other alternatives such as Apache Graph, Graph Lab, Pregel and MapD – a novel platform that uses GPU processing to accelerate Big Data processing – are also analyzed. Finally, we present two case studies that demonstrate the successful use of Big Data within Cloud environments and the challenges that must be addressed in the future.

Society is becoming deeply immersed in the use of electronic devices that generate Petabytes of data, a gold mine for knowledge extraction, with different volume, velocity, and variety. Value and veracity are also two important properties that specify the need of valuable and truthfulness data. These five properties are known as the 5 V's model that supports the Big Data concept.

Big Data became a hot topic among computer researchers and business areas [60], providing organizations with a powerful tool to analyze large structured and

unstructured data and make useful decisions through it. Knowledge extraction frequently requires sophisticated analytic solutions that mine structured and unstructured data helping organizations gaining insights over the information within their private and public data[2015][5].

Clustering a large volume of data in a distributed environment is a challenging issue. Data stored across multiple machines are huge in size, and solution space is large. Genetic algorithm deals effectively with larger solution space and provides better solution. In this paper, we proposed a novel clustering algorithm for distributed datasets, using combination of genetic algorithm (GA) with Mahalanobis distance and k-means clustering algorithm. The proposed algorithm is two phased; in phase 1, GA is applied in parallel on data chunks located across different machines. Mahalanobis distance is used as fitness value in GA, which considers covariance between the data points and thus provides a better representation of initial data. K-means with K-means++ initialization is applied in phase 2 on intermediate output to get final result. The proposed algorithm is implemented on Hadoop framework, which is inherently designed to deal with distributed datasets in a fault-tolerant manner. Extensive experiments were conducted for multiple real-life and synthetic datasets to measure performance of our proposed algorithm. Results were compared with MapReduce-based algorithms, mrk means, parallel k-means and scaling GA. [2017][6].

Department of Computer Science and Engineering, IIT (ISM), Dhanbad, Dhanbad, India amount of data generated has grown markedly. Enormous amount of data generated cannot be placed at a centralized site; it needs to be stored across multiple data sites for the sake of security and easy maintenance. The sheer volume of data makes it difficult to transfer across different machines, which consumes significant bandwidth and time. technique, is effective when dealing with large volumes of data. It is an act of grouping objects based on some similarity factor. Objects in one cluster are highly similar to each other in comparison with the objects belonging to some other group. [2017][7].

Society is becoming deeply immersed in the use of electronic devices that generate

Petabytes of data, a gold mine for knowledge extraction, with different *volume*, *velocity*, and *variety*. *Value* and *veracity* are also two important properties that specify the need of valuable and truthfulness data. These five properties are known as the 5 V's model that supports the Big Data concept [10].

Big Data became a hot topic among computer researchers and business areas [60], providing organizations with a powerful tool to analyze large structured and unstructured data and make useful decisions through it. Knowledge extraction frequently requires sophisticated analytic solutions that mine structured and unstructured data helping organizations gaining insights over the information within their private and public data.

Cloud computing is currently one of the most discussed and promising topic in the information technology field and it was listed in Gartner's top ten technologies list for the last four consecutive years [9]. Cloud computing became a trend for researchers and organizations [21], allowing virtualizing resources and offering theoretically unlimited processing power and storage. In practice, cloud can easily scale up with two types of scaling: *vertical scaling*, which offers the possibility to upgrade servers; and *horizontal scaling* that allows adding new servers to a cluster [9].

capacities and high processing power (therefore, it can host and process big volumes of data.

we present Big Data in the cloud. Particularly we describe NoSQL databases; MapReduce and other variants such as: HaLoop, Twister and Apache Spark that address MapReduce's lack of interactivity; Pregel, GraphLab and Apache Giraph, that were designed to work with graphs, instead of data files; and MapD, which is a novel approach to accelerate Big Data processing by means of GPU processing power. As these systems run over distributed architectures they perfectly fit in the cloud paradigm and are often offered as Big Data as a service or used by several online applications. The paper presents two case studies showing how Big Data and Cloud computing match, the issues on moving big loads of data to the cloud and the existent solutions, disaster recovery plans, and existing Big Data challenges.

The remainder of this paper is structured as follows. Section 2 introduces Big Data and

Cloud Computing. Section 3 presents NoSQL, MapReduce, Graph databases and GPU processing. Section 4 discusses the case studies and existent problems and solutions when moving Big Data to the cloud. Section 5 discusses disaster recovery, and Section 6 presents Big Data challenges, new trends and future directions. Finally, Section 7 concludes the paper [2015][8].

Hadoop provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce [3] paradigm. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data. A Hadoop cluster scales computation capacity, storage capacity and IO bandwidth by simply adding commodity servers. Hadoop clusters at Yahoo! span 25 000 servers, and store 25 petabytes of application data, with the largest cluster being 3500 servers. One hundred other organizations worldwide report using Hadoop.

Hadoop is an Apache project; all components are available via the Apache open source license. Yahoo! has developed and contributed to 80% of the core of Hadoop (HDFS and MapReduce). HBase was originally developed at Powerset, now a department at Microsoft. Hive [15] was originated and developed at Facebook. Pig [4], ZooKeeper [6], and Chukwa were originated and developed at Yahoo! Avro was originated at Yahoo! and is being co-developed with Cloudera.

HDFS is the file system component of Hadoop. While the interface to HDFS is patterned after the UNIX file system, faithfulness to standards was sacrificed in favor of improved performance for the applications at hand.

HDFS stores file system metadata and application data separately. As in other distributed file systems, like PVFS [2][14], Lustre [7] and GFS [5][8], HDFS stores metadata on a dedicated server, called the NameNode. Application data are stored on other servers called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols.

Unlike Lustre and PVFS, the DataNodes in HDFS do not use data protection mechanisms such as RAID to make the data durable. Instead, like GFS, the file content is replicated on multiple DataNodes for reliability. While ensuring data durability, this strategy has the added advantage that data transfer bandwidth is multiplied, and there are more opportunities for locating computation near the needed data.

Several distributed file systems have or are exploring truly distributed implementations of the namespace. Ceph [17] has a cluster of namespace servers (MDS) and uses a dynamic subtree partitioning algorithm in order to map the namespace tree to MDSs evenly. GFS is also evolving into a distributed namespace implementation [8]. The new GFS will have hundreds of namespace servers (masters) with 100 million files per master. Lustre [7] has an implementation of clustered namespace on its roadmap for Lustre 2.2 release. The intent is to stripe a directory over multiple metadata servers (MDS), each of

which contains a disjoint portion of the namespace. A file is assigned to a particular MDS using a hash function on the file name[2014][9].

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

System design is the process of defining the architecture, components, modules, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis systems architecture and systems engineering. If the broader topic of product development blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to be manufactured. System design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

3.2 SCOPE AND OBJECTIVES

In our design, we consider two major entities as shown in a Dataset Owner and a Cloud Server. The owner has a collection of data objects, which will be outsourced to the cloud server for clustering after encryption. The cloud server performs the K-means clustering directly over the encrypted dataset without any decryption. During the clustering the cloud server interacts with the owner for a small amount of encrypted intermediate inputs/outputs. The clustering is finished when the clustering results do not change any more, or a predefined number of iterations is reached.

3.3 USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. An actor is represents a user or another system that will interact with the system modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

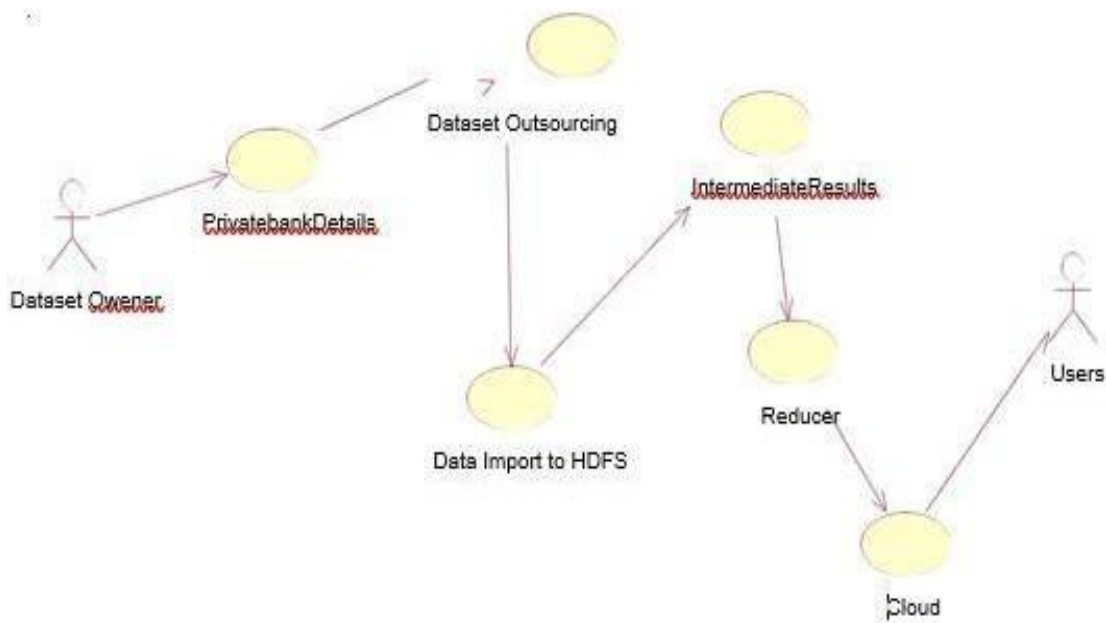


Fig 3.1 Use Case Diagram

3.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflow of components in a system. An activity diagram shows the overall flow of control.

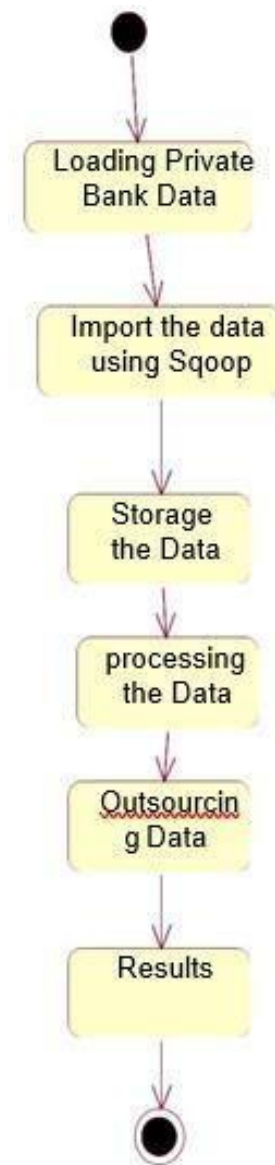


Fig 3.2 Activity Diagram

3.4 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

SEQUENCE DIAGRAM:

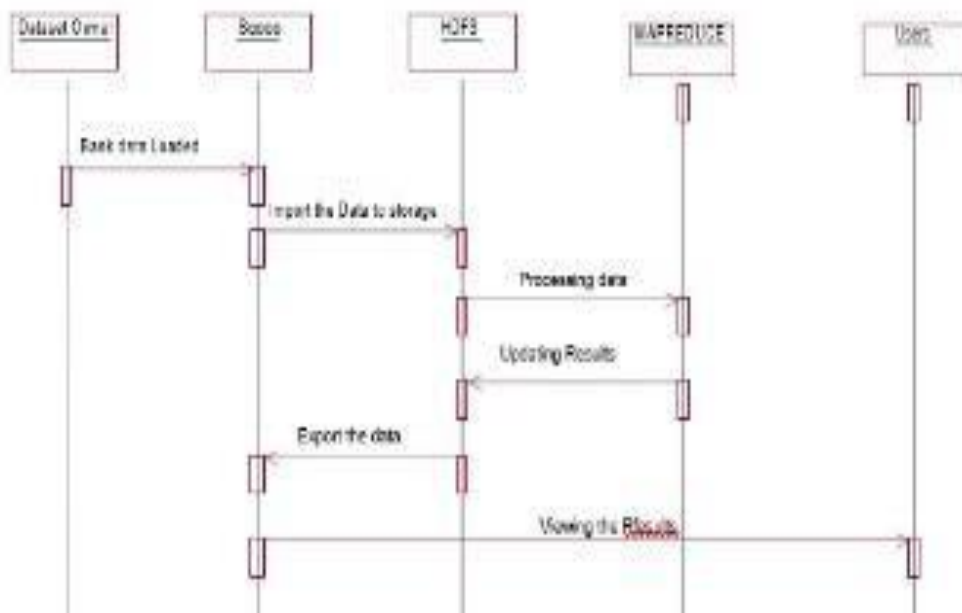


Fig 3.3 SEQUENCE DIAGRAM

3.6 HADOOP

Apache Hadoop offers a scalable, flexible and reliable distributed computing big data framework for a cluster of systems with storage capacity and local computing power by leveraging commodity hardware. Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets using Hadoop MapReduce paradigm.

Hadoop follows a master slave architecture design for data storage and distributed data processing using HDFS and MapReduce respectively. The master node for data storage is hadoop HDFS is the NameNode and the master node for parallel processing of data using Hadoop MapReduce is the Job Tracker. The slave nodes in the hadoop architecture are the other machines in the Hadoop cluster which store data and perform complex computations. Every slave node has a Task Tracker daemon and a DataNode that synchronizes the processes with the Job Tracker and NameNode respectively. In Hadoop architectural implementation the master or slave systems can be setup in the cloud or on-premise.

Hadoop Distributed File System (HDFS) stores the application data and file system metadata separately on dedicated servers. NameNode and DataNode are the two critical components of the Hadoop HDFS architecture. Application data is stored on servers referred to as DataNodes and file system metadata is stored on servers referred to as NameNode. HDFS replicates the file content on multiple DataNodes based on the replication factor to ensure reliability of data. The NameNode and DataNode communicate with each other using TCP based protocols. For the Hadoop architecture to be performance efficient, HDFS must satisfy certain pre-requisites.

The execution of a Map Reduce job begins when the client submits the job configuration to the Job Tracker that specifies the map, combine and reduce functions along with the location for input and output data. On receiving the job configuration, the

job tracker identifies the number of splits based on the input path and select Task Trackers based on their network vicinity to the data sources. Job Tracker sends a request to the selected Task Trackers.

The processing of the Map phase begins where the Task Tracker extracts the input data from the splits. Map function is invoked for each record parsed by the “InputFormat” which produces key-value pairs in the memory buffer. The memory buffer is then sorted to different reducer nodes by invoking the combine function. On completion of the map task, Task Tracker notifies the Job Tracker. When all Task Trackers are done, the Job Tracker notifies the selected Task Trackers to begin the reduce phase. Task Tracker reads the region files and sorts the key-value pairs for each key. The reduce function is then invoked which collects the aggregated values into the output file.

3.7 RISKS INVOLVED IN CLOUD COMPUTING

3.7.1 Technical Issues

Though it is true that information and data on the cloud can be accessed anytime and from anywhere at all, there are times when this system can have some serious dysfunction. You should be aware of the fact that this technology is always prone to outages and other technical issues. Even the best cloud service providers run into this kind of trouble, in spite of keeping up high standards of maintenance. Besides, you will need a very good Internet connection to be logged onto the server at all times. You will invariably be stuck in case of network and connectivity problems.

3.7.2 Security in the Cloud

The other major issue while in the cloud is that of security issues. Before adopting this technology, you should know that you will be surrendering all your company's sensitive information to a third-party cloud service provider. This could potentially put your company to great risk. Hence, you need to make absolutely sure that you choose the most reliable service provider, who will keep your information totally secure. When all

Task Trackers are done, the Job Tracker notifies the selected Task Trackers to begin the reduce phase. Task Tracker reads the region files and sorts the key-value pairs for each key. The reduce function is then invoked which collects the aggregated values into the output file.

3.7.3 Prone to Attack

Storing information in the cloud could make your company vulnerable to external hack attacks and threats. As you are well aware, nothing on the Internet is completely secure and hence, there is always the lurking possibility of stealth of sensitive data.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1 MODULES

ETL WORKFLOW FOR HDFS

SYSTEM SETUP AND DATA INTEGRATION

WEIGHTED K-MEANS CLUSTERING

SINGLE ROUND MAP REDUCED PRIVACY-PRESERVING CLUSTERING

IMPORT/ EXPORT DATA USING SQOOP TOOL ANALYSIS

4.1.1 Extract Transform Load (ETL)

Extract Transform Load (ETL) systems commonly integrate data from multiple applications (systems), typically developed and supported by different vendors or hosted on separate computer hardware. The separate systems containing the original data are frequently managed and operated by different patient records. This represents the most important aspect of ETL, since extracting data correctly sets the stage for the success of subsequent processes.

A properly designed ETL system extracts data from the source systems, enforces data quality and consistency standards, conforms data so that separate sources can be used together, and finally delivers data in a presentation-ready format so that application developers can build applications and end users can make decisions.

Since the data extraction takes time, it is common to execute the three phases in parallel. While the data is being extracted, another transformation process executes while processing the data already received and prepares it for loading while the data loading begins without waiting for the completion of the previous phases.

ETL systems commonly integrate data from multiple applications (systems), typically developed and supported by different vendors or hosted on separate computer hardware. The separate systems containing the original data are frequently managed and operated by different employees. For example, a cost accounting system may combine data from payroll, sales, and purchasing.

4.1.2 SYSTEM SETUP AND DATA INTEGRATION

A Dataset Owner and a Cloud Server: the owner has a collection of data objects, which will be outsourced to the cloud server for clustering after encryption. The cloud server performs the K-means clustering directly over the encrypted dataset without any decryption. During the clustering, the cloud server interacts with the owner for a small amount of encrypted intermediate inputs/outputs. The clustering is finished when the clustering results do not change any more, or a predefined number of iterations is reached.

Integrating data into an integration-platform-as-a-service (ipaaS) cloud database can sound like a scary process, however it really comes down to five simple steps. Business owners who may be tentative to make the transition into big data and the improved performance that comes with that should fear not - it's easier and less risky than ever to use a data integration platform.

Once you have determined essentially how you want your business data to sync, you can now set the criteria for data, choose your platform, and enable the integration. When choosing a platform to use, there are certain features to pay attention to, including price, bi-directional syncing (does the data flow into and out of each system?), number of system you can connect and how many records you're limited to sync in a given time. We would like to think that our system works for businesses of all sizes (feel free to try it for free), but if you need a simple, one-way integration there are other systems out there that may work for you as well.

By integrating data across multiple platforms, an organization that is new to the data management process can quickly gain value from their integration efforts, sometimes in as little as a few days. Other companies that have invested years of resources into old-

school, hard to use integrations can create relevant fields and tune systems to utilize their exiting analytical data as a short-cut to a working big-data system.

4.1.3 WEIGHTED K-MEANS CLUSTERING

A popular extension of the original K-means clustering is the weighted K-means clustering. In weighted K-means clustering, every data element is associated with a real valued weight. This is because different elements in an object can have different level of importance. To support weighted K-means clustering in a privacy-preserving manner, our design should enable the privacy-preserving weighted distance comparison.

Clustering is a task of grouping data based on similarity. A popular k-means algorithm groups data by firstly assigning all data points to the closest clusters, then determining the cluster means. The algorithm repeats these two steps until it has converged. We propose a variation called weighted k-means to improve the clustering scalability. To speed up the clustering process, we develop the reservoir-biased sampling as an efficient data reduction technique since it performs a single scan over a data set. Our algorithm has been designed to group data of mixture models. We present an experimental evaluation of the proposed method.

When clustering massive data set, data reduction is an effective technique to speed up the algorithm. Sampling is a powerful data reduction paradigm to remedy the inherent complexity of clustering. Uniform random sampling in which every data point has the same probability of being selected has been used extensively in data mining and databases. In the case of data sets with large variation in cluster sizes, density biased sampling tends to be a better scheme. In density biased sampling, the probability that a data point will be included in the sample is varied by the density of a cluster. Recent researches propose several techniques to density biased sampling. Our work also follows this path with a step further on extending the k-means algorithm to work with a weighted sample. We propose an algorithm on density biased sampling based on the reservoir technique and a weighted k-means algorithm to cluster a data sample augmented with weights. The proposed algorithms are explained in Sections and, respectively. We

present the experimental results in Section. The conclusion and our future work are discussed in Section.

On scalable popular and successful clustering methods such as k-means to work against large data sets, many algorithms like BIRCH and CLARANS employ the sampling technique to minimize data sets. In BIRCH, a CF-tree structure is built after an initial random sampling step. The CF-tree is used as a summarized data structure with statistical representations of space regions stored on leaf nodes. After the phase of CF-tree building, any clustering algorithm can be applied to the leaf nodes. CLARANS also uses uniform sampling to derive initial representative objects for the clusters. The sampling technique used in these algorithms is uniform random sampling, which assigns every object the same probability of being included in the sample. But many data sets in real life do not follow the uniform distribution scheme. It instead seems to follow the Zipfs distribution, for instance, income and population distribution. In these data sets, some areas such as large metropolitan area have much higher population density than the small cities. If all the populations have equal opportunity of being selected as a representative, sparse areas may be missed and not be included in the sample.

Clustering is a task of grouping data based on similarity. A popular k-means algorithm groups data by firstly assigning all data points to the closest clusters, then determining the cluster means. The algorithm repeats these two steps until it has converged. We propose a variation called weighted k-means to improve the clustering scalability. To speed up the clustering process, we develop the reservoir-biased sampling as an efficient data reduction technique since it performs a single scan over a data set. Our algorithm has been designed to group data of mixture models. We present an experimental evaluation of the proposed method.

4.1.4 SINGLE ROUND MAP REDUCED PRIVACY-PRESERVING CLUSTERING

The clustering purpose for a data object is to find the clustering centre that has the minimum Euclidean distance to it. As different data objects are independent with each other in a single round of clustering, we set the clustering for an object as the Map Reduce job in our scheme. Now, the first task is to achieve Euclidean distance comparison directly

over encrypted data objects and clustering centres.

Privacy and security concerns can prevent sharing of data, derailing data mining projects. Distributed knowledge discovery, if done correctly, can alleviate this problem. The key is to obtain valid results, while providing guarantees on the

(non)disclosure of data. We present a method for k -means clustering when different sites contain different attributes for a common set of entities. Each site learns the cluster of each entity, but learns nothing about the attributes at other sites.

Privacy preserving data mining has become increasingly popular because it allows sharing of privacy-sensitive data for analysis purposes. However, existing techniques such as random perturbation do not fare well for simple yet widely used and efficient Euclidean distance-based mining algorithms. Although original data distributions can be pretty accurately reconstructed from the perturbed data, distances between individual data points are not preserved, leading to poor accuracy for the distance-based mining methods. Besides, they do not generally focus on data reduction. Other studies on secure multi-party computation often concentrate on techniques useful to very specific mining algorithms and scenarios such that they require modification of the mining algorithms and are often difficult to generalize to other mining algorithms or scenarios. This paper proposes a novel generalized approach using the well-known energy compaction power of Fourier-related transforms to hide sensitive data values and to approximately preserve Euclidean distances in centralized and distributed scenarios to a great degree of accuracy. Three algorithms to select the most important transform coefficients are presented, one for a centralized database case, the second one for a horizontally partitioned, and the third one for a vertically partitioned database case. Experimental results demonstrate the effectiveness of the proposed approach.

4.1.5 IMPORT/EXPORT DATA USING SQOOP TOOL ANALYSIS

By using Sqoop the owner updates clustering centers based on information from the cloud server and his/her own secret keys. These new centers are sent to the cloud server in encrypted format for the next round of clustering. Will be iteratively executed until the clustering result does not change any more or the predefined number of iterations is reached.

Map Reduce is the programming framework for processing large scales datasets in utter distributed manner. As shown to process a task with massive amounts of data, MapReduce divides the task into two phases: map and reduce. These two phases are expressed with map and reduce functions, which take It key, value gt; pairs as input and output data format. In a cluster nodes that are responsible for map and reduce functions are called mappers and reducers respectively.

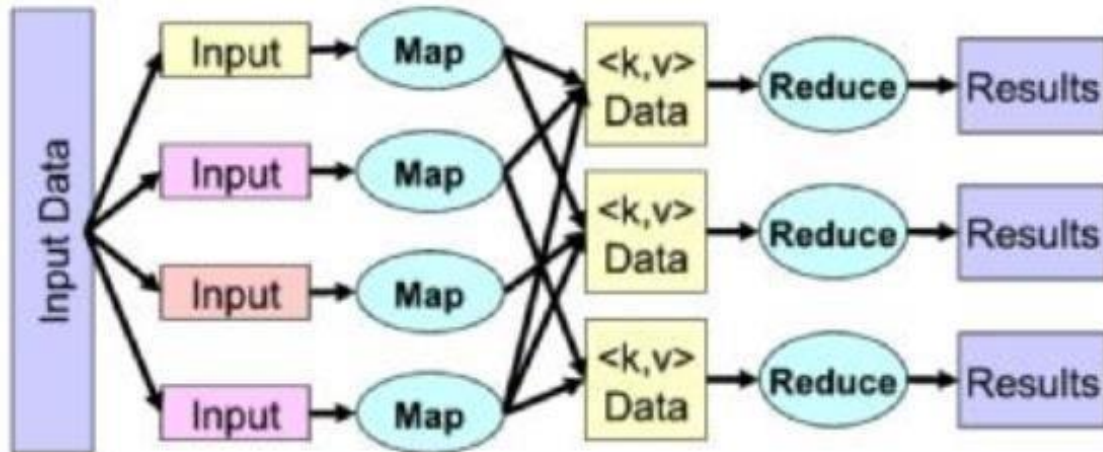


Fig 4.1 Map reduce

4.2 EXISTING SYSTEM

The rapid growth of big data involved in today's data mining and analysis also introduces challenges for clustering over them in terms of volume, variety, and velocity. To efficiently manage large-scale datasets and support clustering over them, public cloud infrastructure is acting the major role for both performance and economic consideration. The problem of privacy-preserving K-means clustering has been investigated under the multi-party secure computation model in which owners of distributed datasets interact for clustering without disclosing their own datasets to each other.

In addition, existing multi-party designs always rely on powerful but expensive cryptographic primitives (e.g., secure circuit evaluation, homomorphic encryption and oblivious transfer) to achieve collaborative secure computation among multiple parties, and are inefficient for large-scale datasets.

Utilizing data perturbation and data transformation for privacy-preserving clustering may not achieve enough privacy and accuracy guarantee. For example adversaries who get a few unencrypted data records in the dataset will be able to recover rest records protected by data transformation. Adversaries who get a few unencrypted data records in the dataset will be able to recover rest records protected by data transformation.

4.3 PROPOSED SYSTEM

We proposed a practical privacy-preserving K-means clustering scheme large scale datasets, which can be efficiently outsourced to public cloud servers. Our proposed scheme simultaneously meets the privacy, efficiency, and accuracy requirements as discussed above. In particular, we propose a novel encryption scheme based on the Learn with Error (LWE) hard problem, which achieves privacy preserving similarity

measurement of data objects directly over ciphertexts. Based on our encryption scheme, we further construct the whole K-means clustering process in a privacy preserving manner, in which cloud servers only have access to encrypted datasets, and will perform all operations without any decryption.

Moreover, we uniquely incorporate MapReduce into our scheme with privacy protection, and thus significantly improving the clustering performance in cloud computing environment. We provide thorough analysis for our scheme in terms of security and efficiency.

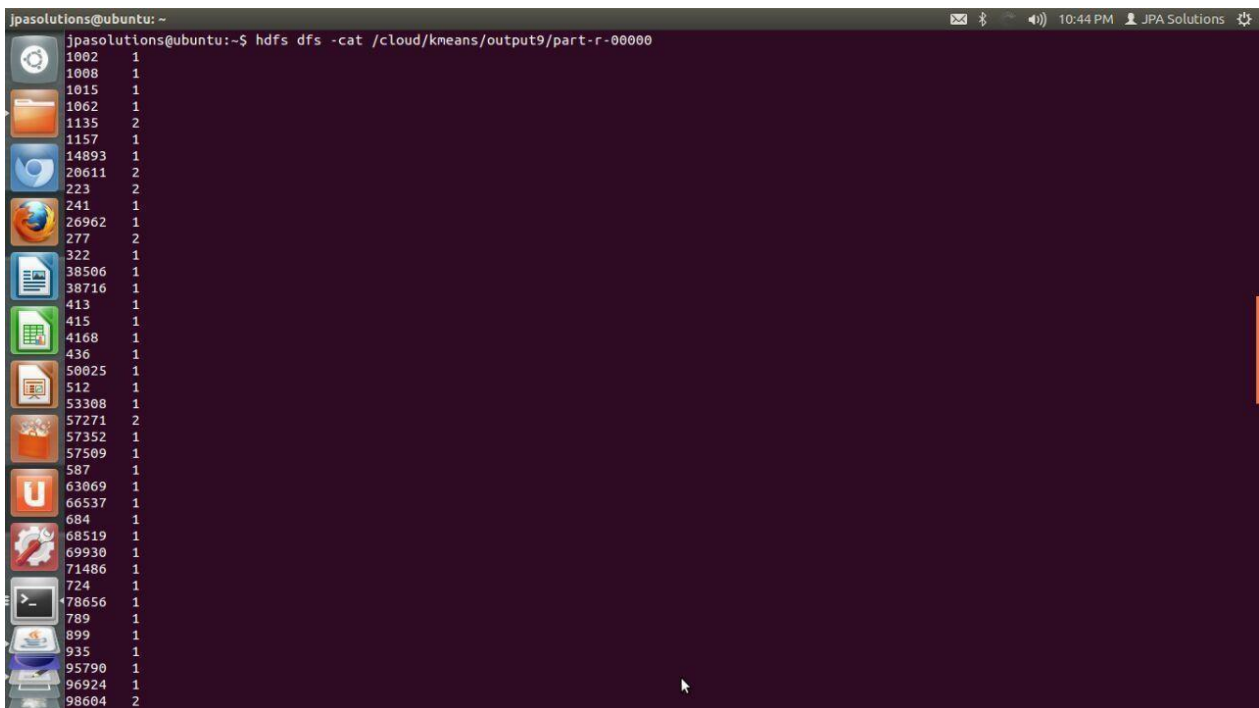
Our extensive evaluation results over 5 million objects show that our privacy-preserving clustering is efficient, scalable, and accurate. Specifically, compared with the K-means clustering over unencrypted datasets, our scheme achieves the same accuracy as well as comparable computational performance and scalability.

CHAPTER-5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

We propose a practical privacy-preserving K-means clustering scheme that can be efficiently outsourced to cloud servers. Our scheme allows cloud servers to perform clustering directly over encrypted datasets, while achieving comparable computational complexity and accuracy compared with clustering over unencrypted ones. We also investigate secure integration of MapReduce into our scheme, which makes our scheme extremely suitable for cloud computing environment. Thorough security analysis and numerical analysis carry out the performance of our scheme in terms of security and efficiency. Experimental evaluation.

Thus after the big data comes as the encrypted form and as it reaches the local data bases like mysql etc, It undergoes clustering using map reduce algorithm and this data is made meaningful and shortened and prioritized as important data as we liked and it gives the final required output and it is stored in the cloud data base.



A terminal window titled 'jpasolutions@ubuntu: ~' showing the output of the command 'hdfs dfs -cat /cloud/kmeans/output9/part-r-00000'. The output is a list of numbers and their corresponding cluster assignments (1 or 2). The window has a dark purple background and a sidebar with various application icons on the left. The top status bar shows the time as 10:44 PM and the user as JPA Solutions.

```
jpasolutions@ubuntu: ~  
jpasolutions@ubuntu:~$ hdfs dfs -cat /cloud/kmeans/output9/part-r-00000  
1002 1  
1008 1  
1015 1  
1062 1  
1135 2  
1157 1  
14893 1  
20611 2  
223 2  
241 1  
26962 1  
277 2  
322 1  
38506 1  
38716 1  
413 1  
415 1  
4168 1  
436 1  
50025 1  
512 1  
53308 1  
57271 2  
57352 1  
57509 1  
587 1  
63069 1  
66537 1  
684 1  
68519 1  
69930 1  
71486 1  
724 1  
78656 1  
789 1  
899 1  
935 1  
95790 1  
96924 1  
98604 2
```

CHAPTER-6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this work, we proposed a privacy-preserving MapReduce based K-means clustering scheme in cloud computing. Thanks to our light-weight encryption design based on the LWE hard problem, our scheme achieves clustering speed and accuracy that are comparable to the K-means clustering without privacy protection. Considering the support of large-scale dataset, we securely integrated MapReduce framework into our design, and make it extremely suitable for parallelized processing in cloud computing environment. In addition, the privacy preserving Euclidean distance comparison component proposed in our design can also be used as an independent tool for distance based applications. We provide thorough analysis to show the security and efficiency of our scheme. Our prototype implementation over 5 million data objects demonstrates that our scheme is efficient, scalable, and accurate for K-means clustering over large-scale dataset.

The traditional data mining approaches could not be directly implanted on big data as it faces difficulties to analyze big data. Clustering is one of the major techniques used for data mining in which mining is performed by finding out clusters having similar group of data. In this paper we have discussed some of the current big data mining clustering techniques. Comprehensive analysis of these techniques is carried out and appropriate clustering algorithm is provided.

6.2 FUTURE WORK

Apache Spark is a fast and general-purpose cluster computing systems. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, Graph X for graph processing, and Spark Streaming.

In future, Spark tool can be used to improve the performance as Spark is 100 times faster when compared to Map Reduce.

REFERENCES

Mikhail J. Atallah and Keith B. Frikken (2010), Securely outsourcing linear algebra computations. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS '10, pages 48–59, New York, NY, USA.

J. Huang, R. Zhang, R. Buyya, and J. Chen. Melody-join (2014), Efficient earth mover's distance similarity joins using map reduce. In 2014 IEEE 30 th International Conference on Data Engineering, pages 808–819.

Rakesh Agrawal and Ramakrishnan Srikant (2000), Privacy-preserving data mining. SIGMOD Rec., 29(2):439–450.

Xun Yi and Yanchun Zhang (2013), Equally contributory privacy-preserving k-means clustering over vertically partitioned data. Inf. Syst., 38(1):97– 107.

Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy (2013), The effectiveness of lloyd-type methods for the k-means problem. J. ACM, 59(6):28:1–28:22.

Ackerman Margareta, Ben-David Shai, Branzei Simina, and Loker David(2012), Weighted clustering. pages 858–863.

Jiawei Yuan and Shucheng Yu. Efficient Privacy-Preserving biometric

Identification in cloud computing. In 2013 Proceedings IEEE INFOCOM (INFOCOM'2013), pages 2652–2660, Turin, Italy.

Jiawei Han and Micheline Kamber. Chapter 7, Data Mining(2006), Concepts and Techniques, Second Edition. Morgan Kaufmann.

K.Srilatha and V.Ulagamuthalvi "Support Vector Machine and Particle Swarm Optimization Based Classification of Ovarian Tumour "BIOSCIENCE BIOTECHNOLOGY RESEARCH COMMUNICATIONS- vol 12(3): 714-719 (2019).

K.Srilatha and V.Ulagamuthalvi "A Comparative Study on Tumour Classification" *Research J. Pharm. and Tech* 2019; 12(1): 407-411.

S. R. Khonde & V. Ulagamuthalvi (2019): Ensemble-based semi-supervised learning approach for a distributed intrusion detection system, *Journal of Cyber Security Technology*, DOI: 10.1080/23742917.2019.1623475 , Taylor and Francis.

Khonde S & Ulagamuthalvi Venugopal (2019), Hybrid Architecture for Intrusion detection system, *Ingenierie des Systemes d'Information* Volume 24, No. 1 , PP – 19-28 February 2019.

Shraddha Khonde and V. Ulagamuthalvi(2019), Fusion of feature selection and Random Forest for an Anomaly based intrusion detection system , *Journal of Computational and Therotical Nanoscience* , Volume 16, PP 3603-3607.

Ajay, M. D. V., N. Adithya, and A. Mary Posonia. "TECHNICAL ERA: An Online Web Application." In *IOP Conference Series: Materials Science and Engineering*, vol. 590, no. 1, p. 012002. IOP Publishing, 2019.

Mary Posonia, A., Vigneshwari, S., Albert Mayan, J., Jamunarani, D. "Service direct: Platform that incorporates service providers and consumers directly ." *International Journal of Engineering and Advanced Technology(IJEAT)*, ISSN: 2249 – 8958, Volume- 8 Issue-6, August 2019

Keerthana, K. T. E., S. Karpagavalli, and A. Mary Posonia. "Smart System Monitoring Agricultural Land Using Iot." In 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), pp. 1-7. IEEE, 2018.

Ahkila, A., Arva Kiranmai, and A. Mary Posonia. "Aggregate tracker of invisible web with checkbox interfaces." In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1-4. IEEE, 2016.

Abhishek, P.S.B.Ravipati, P.Vigneswari, S. Posonia, A. " Product Management System".,Journal of Computational and Theritical Nano Science,Vol.16 no.8(2019),pp.3311-3315

APPENDIX

A SOURCE CODE

```
package com.cloud.encryption1;

import java.io.BufferedReader;

public class DataCluster {

    public static void main(String args[])

    {

        //Connection

        con=null; Statement

        st; ResultSet rs;

    try

    {

        Connection

        con=DatabaseConnection.getConnection();

        st=con.createStatement();

        StringBuffer buffer=new StringBuffer();

        rs=st.executeQuery("select * from
```

```

patients");

if(rs.next()){

    System.out.println("inside");

    String values=(String)rs.getString(1)+"
    "+(String)rs.getString(2); buffer.append(values+"@");

}

System.out.println("buffer values--"+buffer);

String

encdata=TestProtector.encrypt(buffer.toString(),"H1");

System.out.println("encdata"+encdata);

st.close();

BufferedWriter bufWriter =

null; try {
    bufWriter = new BufferedWriter(new
FileWriter("/home/jpasolutions/sathish/destination.txt
"));

    if (buffer != null) {

        System.out.println(bufe

r);

```

```

        bufWriter.write(encdata)

        ;

    }

    } catch (FileNotFoundException
    e) { e.printStackTrace();
    } finally {

        if (bufWriter != null)

            bufWriter.close();

    }

}

catch (Exception e)

{

    e.printStackTrace();

}

String decryptD=decryptData();

    System.out.println("Decrypted Data-

    "+decryptD);

}

```

```

public static String decryptData(){

    BufferedReader br =

    null; String

    contentLine=null;

    String data=null;

    try{

        br = new BufferedReader(new
FileReader("/home/jpasolutions/sathish/destination.t
xt"));

        System.out.println("Reading the file using readLine()
method:"); contentLine = br.readLine();

        /*while (contentLine != null) {

            System.out.println(contentLin

            e);

            contentLine = br.readLine();

        }*/

        data=TestProtector.decrypt(contentLine,"H

1"); System.out.println("Decrypted Data--

```

```

        "+data);

    }

    catch(Exception

    e){

        e.printStackTrace

        ();

    }

    return data;

    }
}
package com.cloud.encryption;
import

org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

/**

*/

```

```
public class WCReducer extends
```

```
    Reducer<Text,LongWritable,Text,LongWritable> { public void reduce(Text  
    key,Iterable<LongWritable> values,Context ctx) throws
```

```
IOException, InterruptedException {
```

```
    // initialize the variable
```

```
    sum long sum=0;
```

```
    // Iterate the list of values for the key and  
    sum it for(LongWritable value:values){
```

```
        sum = sum + value.get();
```

```
    }
```

```
    // In the end you got the word(as key) and corresponding count(as sum). Write to ctx  
    object
```

```
    .
```

```
    ctx.write(key,new LongWritable(sum));
```

```
}
```

```
}
```

B SCREENSHOTS

The screenshot shows a web application titled "User Data Collection" with a light blue header and a green gradient background. The form is titled "User ..." and contains the following fields and controls:

- First Name: Last Name:
- Gender: ☐ Male ☒ Female
- Customer ID: Account:
- Employee Type: IP Address: Experience:
- Account Type: Balance Amount: Annual Income:
- Port: Nationality:
- Caste: Religion: Location:
- Zip Code: Mobile No:

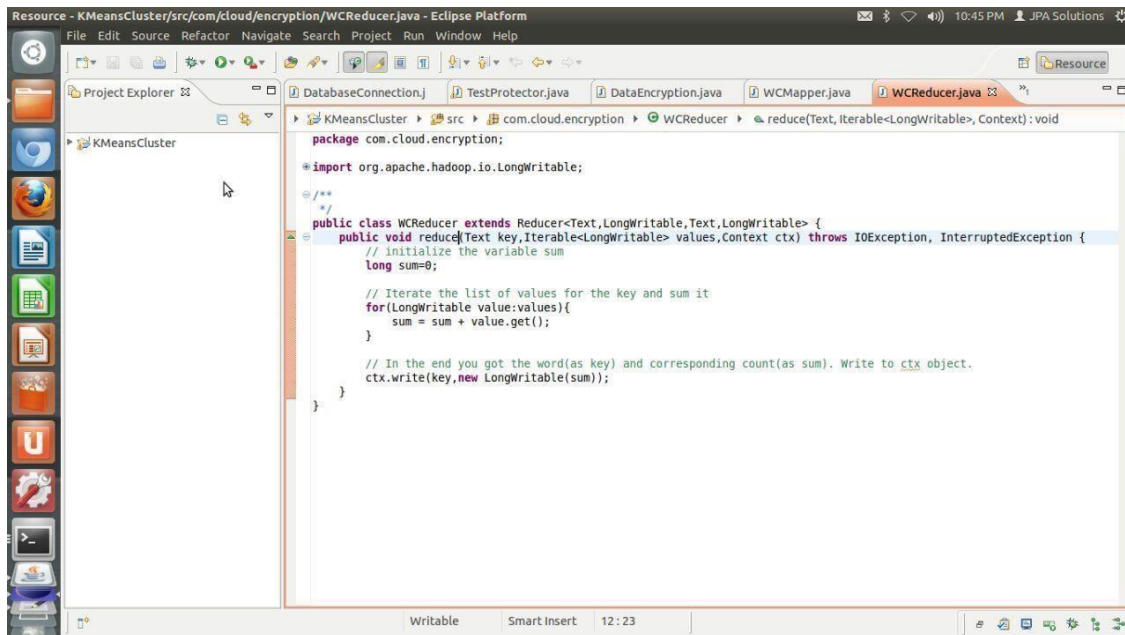
Buttons include "Get ID", "Get P...", "Reset", "Submit", and "Exit".

6.1 USER INTERFACE

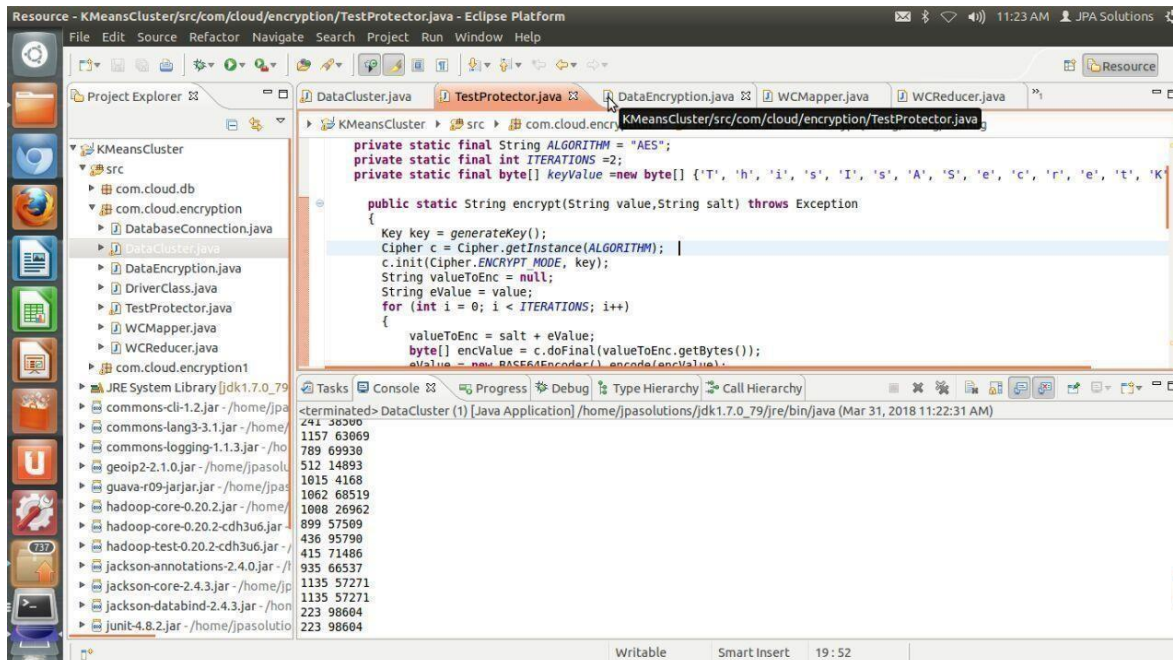
The screenshot shows the SQL Developer interface with a table named "loan_details" selected. The table has the following columns: loan_id, name, and loan_type. The data is as follows:

loan_id	name	loan_type
557	chraen	Personal Loan
512	isell	Personal Loan
2062	sanatya	Car Loan
2062	sanatya	Car Loan
2062	sanatya	Car Loan
557	chraen	Agricultural Loan
557	chraen	Personal Loan
557	chraen	Personal Loan
241	vaga	Personal Loan
2008	vaga	Personal Loan
2008	vaga	Personal Loan

6.2 SQL DATA BASES



6.3 JAVA SOURCE CODE



6.4 JAVA SOURCE CODE

```
Terminal
jpasolutions@ubuntu: ~/sathish/code
jpasolutions@ubuntu:~$ cd sathish/
jpasolutions@ubuntu:~/sathish$ cd code/
jpasolutions@ubuntu:~/sathish/code$ java Userdatacollection

^7
jpasolutions@ubuntu:~$
j ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
E corresponds to your MySQL server version for the right syntax to use near 'dtaba
S ses
j show databases' at line 1
e mysql> show database
T -> ;
j ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'datab
ase' at line 1
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| kmeans |
| mysql |
| performance_schema |
| privatebank |
| test |
+-----+
6 rows in set (0.27 sec)

mysql>
```

6.5 DATABASES

```
jpasolutions@ubuntu: ~
FILE: Number of bytes read=0
FILE: Number of bytes written=540852
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=1381
HDFS: Number of bytes written=0
HDFS: Number of read operations=19
HDFS: Number of large read operations=0
HDFS: Number of write operations=0

Job Counters
  Launched map tasks=4
  Data-local map tasks=4
  Total time spent by all maps in occupied slots (ms)=94299
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=94299
  Total vcore-seconds taken by all map tasks=94299
  Total megabyte-seconds taken by all map tasks=96562176

Map-Reduce Framework
  Map input records=40
  Map output records=40
  Input split bytes=616
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=6864
  CPU time spent (ms)=10050
  Physical memory (bytes) snapshot=600518656
  Virtual memory (bytes) snapshot=8998125568
  Total committed heap usage (bytes)=372768768

File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=0
18/03/31 22:43:44 INFO mapreduce.ExportJobBase: Transferred 1.3486 KB in 46.7501
seconds (29.5401 bytes/sec)
18/03/31 22:43:44 INFO mapreduce.ExportJobBase: Exported 40 records.
jpasolutions@ubuntu:~$
```

6.6 JOB EXECUTION

