Design and Development of Virtualized Hybrid Architecture for Cloud Computing

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

Master of Engineering

in

Software Engineering

by

Aarti Sharma

(800931001)

Under the supervision of:

Dr. Inderveer Chana (Assistant Professor, CSED)

Mrs. Anju Bala (Assistant Professor, CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2011

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Design and Development of Virtualized Hybrid Architecture for Cloud Computing*", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Inderveer Chana* and *Mrs. Anju Bala* refers other researcher's work which are duly listed in the reference section.

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

(Aarti Sharma)

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

(Dr. Inderveer Chana)
Assistant Professor,
Computer Science and
Engineering Department

(Mrs. Anju Bala)
Assistant Professor,
Computer Science and
Engineering Department

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

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

Acknowledgement

I would like to acknowledge everyone who supported me during my experience at Thapar University and my work on this thesis.

First of all I would like to thank my family who stayed with me and supported me all the time while I made this dream come true.

I extend my thanks to my thesis supervisor, Dr. Inderveer Chana and Mrs. Anju Bala, who helped me to materialize my ideas on this thesis. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions.

Their feedback and editorial comments were also valuable for writing this thesis.

I am also very thankful to the faculty and staff members of Computer Science and Engineering Department, for their direct – indirect help and cooperation.

I would also like to thank especially Vikash Kaushik, Aman Mehta, Harshpreet Singh, Amrita Bhandari for their valuable co-operation in this thesis works.

Aarti Sharma

(800931001)

Abstract

Cloud Computing is a buzzword for IT infrastructure. There is a shift in business from desktops towards Cloud Computing. Cloud Computing is not a completely new concept; it has intricate connection to the established Grid Computing paradigm, and other relevant technologies such as utility computing, cluster computing and distributed systems in general.

Cloud DBMS is a distributed database that delivers a query service across multiple distributed database nodes located in multiple data centres, including cloud data centers. The management of data centre is very costly and energy consuming. The cloud computing by providing the database- as- a-service helps the organizations in reduction of labour and infrastructure cost.

In this thesis, the MapReduce and Parallel DBMS are compared focusing on Cloud DBMS properties, concluding that there is a need of a hybrid architecture which can combine the properties of these traditional approaches and fulfil the requirements of current Clouds. The Hybrid Architecture is implemented in virtualized environment using Hadoop Framework, Java for MapReduce programming and Mysql for query execution in Structured Query Language (SQL). The MapReduce is extended to provide homogeneity to work on relational databases.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The underlying concept of cloud computing dates back to the mainframe days of 1960's when the idea of utility computing was coined by MIT computer scientist John McCarthy. He wrote that "computation may someday be organized as a public utility". Utility computing became a sort of business for companies such as IBM. IBM saw the potential for enormous profit to be made in this type of business and started providing computing services. Figure 1.1 [35] shows the evolution of Cloud from Grid computing.

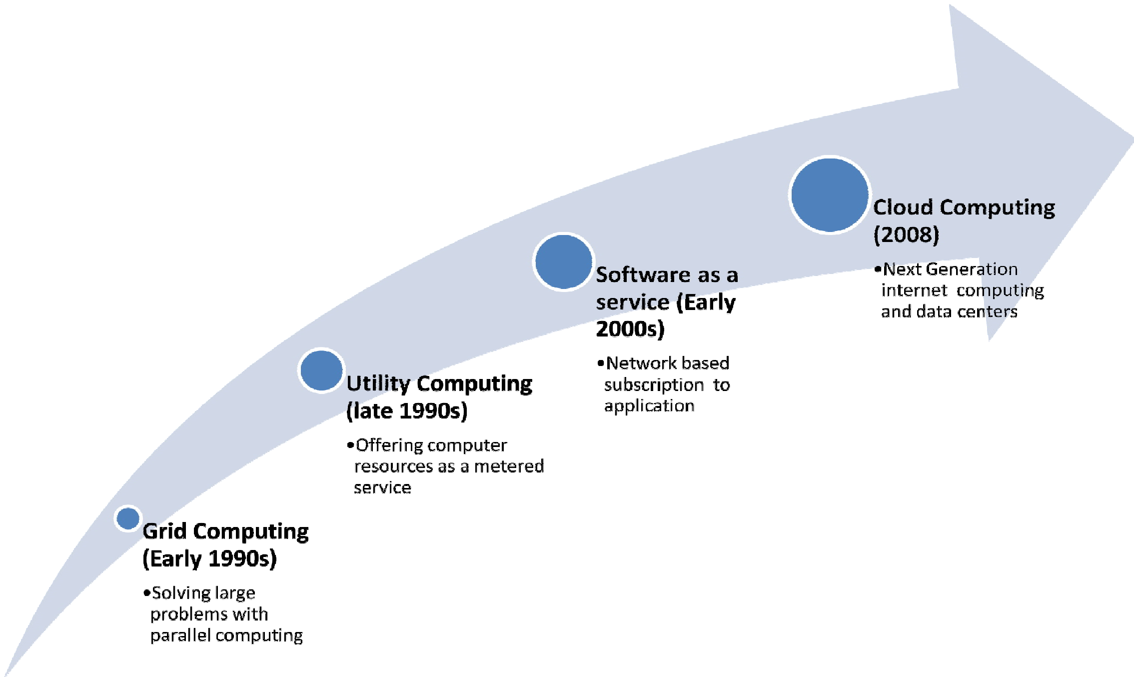


Figure 1.1 Evolution of Cloud Computing from Grid Computing [35]

Grid computing grew up with the idea of linking the number of computers to increase scalability and availability. The grid specifically refers leverage of computers for particular application while cloud computing leverages the multiple resources along with the computational resources to provide the services-to-the-end-user.

1.1 Cloud Computing

Cloud Computing is hinting at a future not to compute on local computers, but on centralized facilities operated by third-party computing and storage utilities [18]. It is more than a few applications you can access over the internet. It allows you to buy hardware resources online, take an entire platform for the software development, or setup a platform in-house.

The National Institute of Standards and Technology (NIST) puts it this way: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [26]."

NIST also categorizes cloud computing into three “as a service” offerings, namely infrastructure, platforms and software, which are broken down in more detail in stack below:

1.1.1 Cloud Computing Stack

Cloud computing is emerging as a model in support of “everything-as-a-service” (XaaS). The cloud stack [12] can be grouped basically into three categories as shown in Figure 1.2 [12].

- **Infrastructure as a Service (IaaS):** As the name implies, IaaS is a service delivery model in which an organization is given control over the different resources and applications. These resources comprise of storage, hardware, servers, networking components, etc. On the lowest level of the infrastructure closest to the hardware two types of services are distinguished: Physical Resource Set (PRS) and Virtual Resource Set (VRS) services. The PRS layer implementation is hardware dependent and therefore tied to a hardware vendor, whereas the VRS layer can be built on vendor independent hypervisor technology. Examples of PRS services are Emulab and iLO and VRS services include Amazon EC2, Eucalyptus, Tycoon, Nimbus, and OpenNebula.
- **Platform as a Service (PaaS):** This component of cloud can be defined as a set of software and product development tools that allows developers to create applications on the provider’s platform, i.e., applications can be build on the internet and executed on providers infrastructure. The services are categorized into Programming Environments and Execution Environments. Example of the former is Sun’s project Caroline and the Django framework, and examples of the latter are Google’s App Engine, Joyent’s Reasonably Smart and Microsoft’s Azure.

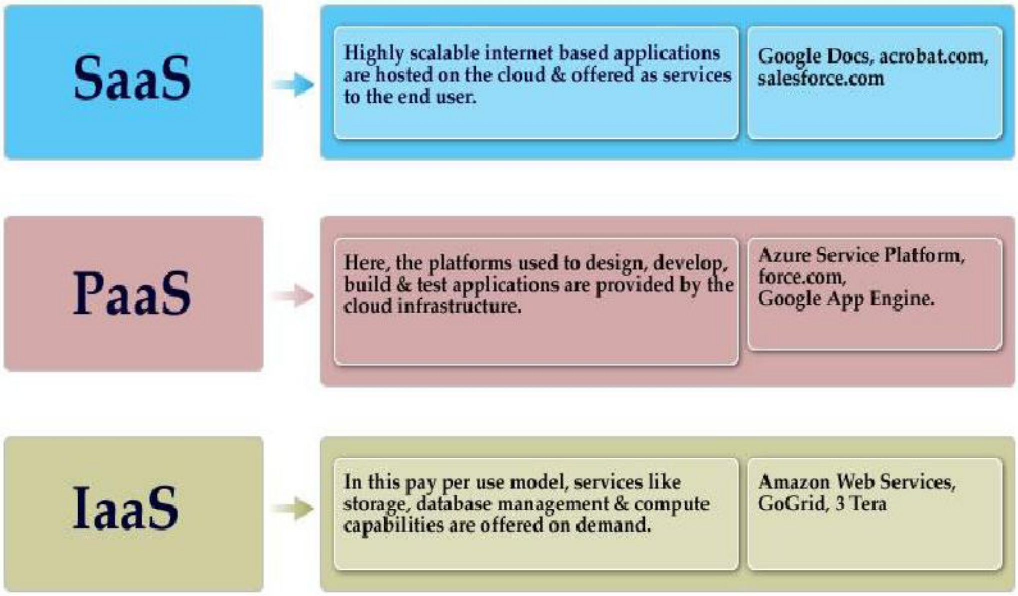


Figure 1.2 Cloud Computing Service Model [12]

- **Software as a Service (SaaS):** This is the most familiar and prolific cloud service. All the applications that run on the Cloud and provide a direct service to the customer are located in the SaaS layer. The application developers can either use the PaaS layer to develop and run their applications or directly use the IaaS infrastructure. Here, Basic Application Services (BAS) and Composite Application Services (CAS) are distinguished. Examples of Basic Application Services are the OpenId and Google Maps services. In the Composite Application Service category the mash-up support systems with Open social are the prominent example allowing entire social networks like MySpace to be used as Basic Services.

1.1.2 Cloud Deployment Models

On the basis of the location where it is deployed, clouds can be categorized as [15]:

- **Public cloud**: In Public cloud the computing infrastructure is hosted by the cloud vendor at the vendor’s premises. The customer has no visibility and control over where the computing infrastructure is hosted. The computing infrastructure is shared between any organizations.
- **Private cloud**: The computing infrastructure is dedicated to a particular organization and not shared with other organizations. Some experts consider that private clouds are not real examples of cloud computing. Private clouds are more expensive and more secure when compared to public clouds.

Private clouds are of two types: On-premise private clouds and externally used by one organization, but are hosted by a third party specializing in cloud infrastructure. Externally hosted private clouds are cheaper than On-premise private clouds.

- **Hybrid cloud:** Organizations may host critical applications on private clouds and applications with relatively less security concerns on the public cloud. The usage of both private and public clouds together is called hybrid cloud. A related term is Cloud Bursting. In Cloud bursting organization use their own computing infrastructure for normal usage, but access the cloud for high/peak load requirements. This ensures that a sudden increase in computing requirement is handled gracefully.
- **Community cloud:** involves sharing of computing infrastructure in between organizations of the same community. For example all Government organizations within the state of California may share computing infrastructure on the cloud to manage data related to citizens residing in California.

1.1.3 Benefits of Cloud Computing

Cloud computing offers various services as listed below [9]:

- **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

- **Rapid elasticity:** Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured Service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

1.2 Database Management in Cloud Computing

Cloud computing provides access to large pools of data and computational resources through a simple and unified interface between vendor and user, allowing vendors to focus more on the software itself rather than the underlying framework. Applications on the Cloud include Software as a Service system and Multi-tenant databases.

There are many ways in which computational power and data storage facilities are provided to users, ranging from a user accessing a single laptop to the allocation of thousand of compute nodes distributed around the world. Users generally locate resource-based on a variety of characteristics, including the hardware architect, memory and storage capacity, network connectivity and, occasionally, geographic location [8].

1.3 Challenges in Cloud Computing

The section provides a detailed overview about various obstacles for the growth of cloud computing along with the measures to overcome these obstacles [10].

- **Business Continuity and Availability of Services**

  Organizations worry about the whether the utility computing services will be adequately available or not. This provides the opportunity for multiple vendors to provide Cloud computing services.
- **Data Lock-In**

  The customer cannot easily extract their data and programs from one site to run on another site. The solution is to standardize the API's, so that SaaS developer could deploy services and data across multiple providers.

- **Data Confidentiality and Audit ability**

  Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. There are also requirements for audit ability. Solution is to make cloud computing environment secure by using encrypted storage, Virtual Local Area Networks, and network middle boxes. Similarly, auditability could be added as an additional layer beyond the reach of virtualized guest OS, providing the facilities arguably more secure than those built into the applications themselves and centralizing the software responsibilities related to the confidentiality and audit ability into single logic layer.

- **Data Transfer Bottlenecks**

  Cloud users and cloud providers have to think about the implications of placement and traffic at every level of the system if they want to minimize the cost. This can provide an opportunity to overcome high costs of internet transfers by FedExing disks, and Higher BW switches.

- **Performance Predictability**

  Multiple virtual machines can share the CPU's and main memory well in cloud computing, but that I/O sharing is problematic. The solution is to improve the architecture and operating systems to efficiently virtualized interrupts and I/O channels.

- **Scalable Storage**

  How to apply cloud computing requirements to provide the persistent storage? The opportunity is to create a Storage System that would not meet these requirements but combine them with the cloud advantages of scaling arbitrarily up and down on-demand, as-well as meeting the programmer's expectations in regard to resource management for scalability, data durability, and high availability.

- **Bugs in Large Distributed System**

  One of the difficult challenges in cloud computing is removing errors in these very large scale distributed systems. A common occurrence is that these bugs cannot be reproduced in smaller configurations, so that debugging must occur at scale in the production data centres. It gives the opportunity to rely on virtual machines in cloud computing.

- **Scaling Quickly**

  Pay-as-you-go certainly applies to storage and to network bandwidth, both of which count bytes used. Computation is slightly different, depending on the virtualization level. The opportunity is then to automatically scale quickly up and down in response to load in order to save money and resource but without violating service level agreements.

- **Reputation Fate Sharing**

  One customer's bad behavior can affect the reputation of the cloud as a whole. An opportunity would be to create reputation- guarding of the cloud as a whole. An opportunity would be to create reputation-guarding services similar to the "trusted email" services currently offered (free) to services hosted on smaller ISP's, which experience a microcosm of this problem.

- **Software Licensing**

  Current software Licenses commonly restrict the computers on which the software can run. The solution is either use open source software or simply for the commercial software companies to change their licensing structure to better fit Cloud Computing.

1.4 Organisation of Thesis

This thesis is organized as follows –

Chapter 2 – This chapter describes in detail the literature survey on Cloud Database Management including Database management as a service in detail.

Chapter 3 – This chapter describes the problem statement of the thesis and requirement analysis of Cloud Database environment used in the thesis.

Chapter 4 – This chapter describes the solution of problem, design of solution, Hadoop MapReduce. It also includes the design, experimental results of the case study of Student Result Evaluation System.

Chapter 5 – This chapter describes the conclusion and future research work possible.

Chapter 2
Literature Review

This chapter provides the introduction about Cloud Database Management System, the various challenges and characteristics required for the database management on the cloud environment. The available database management applications and tools are compared in the cloud environment. The services and features offered by popular Database-as-a-Service providers are also discussed. It also discusses about virtualization technology with its benefits and need in the cloud environment.

2.1 Cloud Database Management System (CDBMS)

Cloud computing systems fundamentally provide access to large amounts of data and computational resources through a variety of interfaces [16]. Cloud DBMS (CDBMS) is a distributed database that delivers a query service across multiple distributed database nodes located in multiple geographically-distributed data centres, both corporate data centres and cloud data centres. Querying distributed data sources is precisely the problem that businesses will encounter as cloud computing grows in popularity.

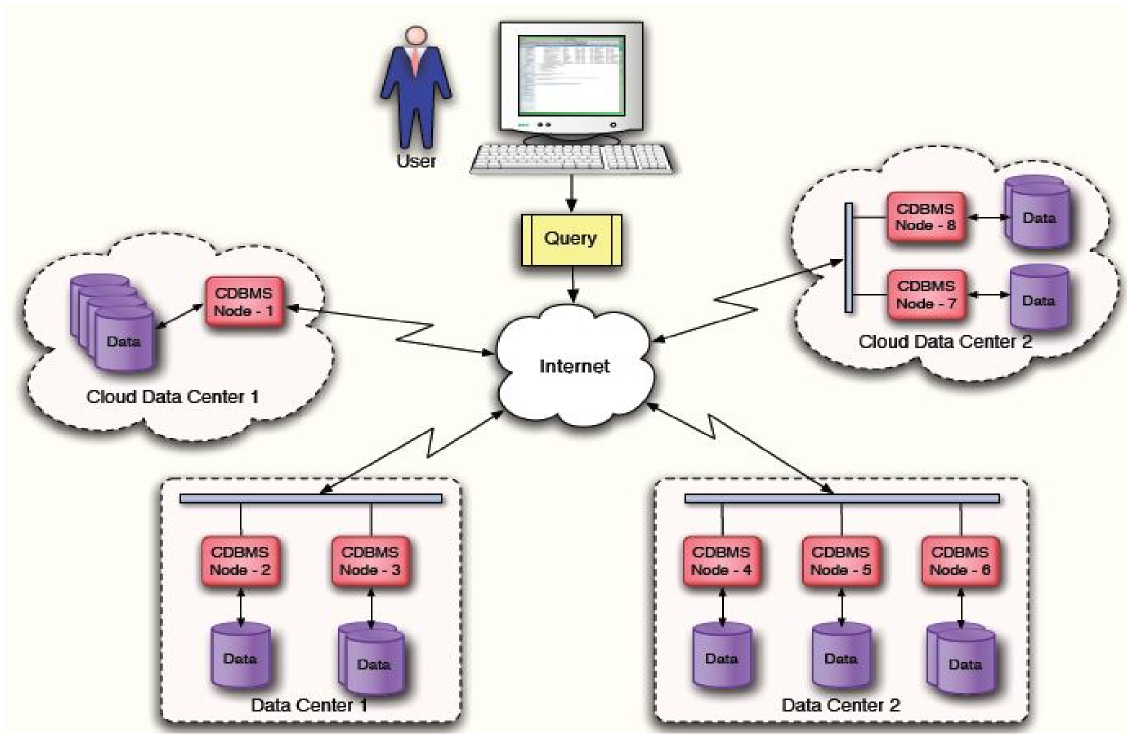


Figure 2.1: Cloud DBMS (CDBMS) [34]

The Figure 2.1 [34] above illustrates that a query is originated from anywhere through internet which is resolved by the cloud and corporate Data centres via internet which are geographically distributed.

The CDBMS will not concentrate all query traffic through a single node. A peer-to-peer architecture will be far more scalable - with any single node able to receive any query. In such an arrangement, each node needs to have a map of the data stored at every node and know the performance characteristics of every node. When a node receives a query its first task is to determine which node is best able to respond to the query. It then passes responsibility for the query to that node. That node executes the query and returns the result directly to the user [16].

2.1.1 Challenges of Managing Databases on Cloud Computing

Database systems provide an extremely attractive interface for managing and accessing data, and have proven to be wildly successful in many financial, business, and Internet applications. However, they have several serious limitations [48]:

- **Database systems are difficult to scale.** Most database systems have hard limits beyond which they do not easily scale. Once users reach these scalability limits, time consuming and expensive manual partitioning, data migration, and load balancing are the only recourse.
- **Database systems are difficult to configure and maintain.** Administrative costs can easily account for a significant fraction of the total cost of ownership of a database system. Furthermore, it is extremely difficult for untrained professionals to get good performance out of most commercial systems–for example, we found that it took several months for a senior graduate student to tune and configure a commercial parallel database system for a simple 8-query workload [19].
- **Diversification in available systems complicates selection.** The rise of specialized database systems for specific markets (e.g., main memory systems for OLTP or column-stores for OLAP) complicates system selection, especially for customers whose workloads do not neatly fall into one category.
- **Peak provisioning leads to unneeded costs.** Database workloads are often bursty in nature, and thus, provisioning for the peak often results in excess of resources during off-peak phases, and thus unneeded costs

2.1.2 Cloud DBMS Characteristics:

The following are the characteristics for the Cloud Data Management [5]:

- **Elasticity:** if data analysis software product A requires an order of magnitude more compute units than data analysis software product B to perform the same task, then product A will cost (approximately) an order of magnitude more than B. Efficient software has a direct effect on the bottom line.
- **Fault Tolerance:** A fault tolerant analytical DBMS is simply one that does not have to restart a query if one of the nodes involved in query processing fails.
- **Untrusted Host:** The storage of the data off-premises increases the number of potential security risks, and appropriate precautions must be made.
- **Heterogeneity:** A node observing degraded performance would have a disproportionate affect on total query latency. A system designed to run in a heterogeneous environment would take appropriate measures to prevent this from occurring.
- **Data Encryption:** In order to prevent unauthorized access to the sensitive data, any application running in the cloud should not have the ability to directly decrypt the data before accessing it.
- **Business Intelligence Support:** Business analysts are often not technically advanced and do not feel comfortable interfacing with the database software directly. These tools typically interface with the database using ODBC or JDBC, so database software that want to work these products must accept SQL queries over these connections.

2.1.3 Database Management Types on Cloud Computing

The huge amount of data which is needed to be stored on the clouds can be categorised as: transactional data and analytical data.

- **Transactional Database Management:**

  Transactional data management refers to the databases that back banking, airline reservation, online e-commerce, and supply chain management applications.

- **Analytical Database Management:**

  Analytical data management refers to applications that query a data store for use in business planning, problem solving, and decision support [6].

On the basis of details given in [5, 6], the Table 2.1 compares the transactional and analytical Databases as follows:

Table 2.1: Transactional vs Analytical Database Management

Properties	**Transactional**	**Analytical**
Source of data	Operational data, it is original source of data	Consolidation data, data comes from transactional databases
Purpose of the data	To control and run fundamental business task	To help with planning, problem solving and decision support
Processing speed	Typically very fast	Depends on the amount of data involved
Space requirements	Can be relatively small if historical data is archived	Larger due to existence of aggregation structures and history.
ACID guarantee	Hard to maintain, ElasTrans can be used.	Easy to maintain
Shared nothing architecture	Not a good match	Is a good match
Sensitive data	Risk to store transactional data	Can be left out of analysis

The database technology of transactional systems, such as OLTP systems, communications applications and work flow systems, does not pose a severe problem at the data level. The relational databases and column store databases have centralized architectures and such architectures encounters a scalability limit at some point, both within and between data centres.

2.2 Cloud Database-as-a-Service Providers

Database-as-a-service provides the ability to leverage the services of a remotely hosted database, sharing it with other users, and having it logically function as if the database were local. Different models are offered by different providers, but the power is to leverage database technology that would typically cost thousands of dollars in hardware and software licenses. Database-as-a-service providers include Amazon SimpleDB, and Microsoft Azure.

2.2.1 Amazon SimpleDB

Amazon SimpleDB [46] is a highly available, flexible, and scalable non-relational data store that offloads the work of database administration. Developers simply store and query data items via web services requests, and Amazon SimpleDB does the rest. It provides a simple web services interface to create and store multiple data sets, query your data easily, and return the results. Your data is automatically indexed, making it easy to quickly find the information that you need. There is no need to pre-define a schema or change a schema if new data is added later. And scale-out is as simple as creating new domains, rather than building out new servers. Amazon SimpleDB provides us below features:

- Limited or no database administration.
- Multiple geographical distributed database replicas.
- Automatic indexing.
- High availability and scalability.
- Performance tuning.
- No minimum fee. Pay only for what you use.

2.2.2 Microsoft SQL Azure

SQL Azure [47] is part of the Windows Azure platform: a suite of services providing hosted computing, infrastructure, Web services and data services. The SQL Azure component provides the full relational database functionality of SQL server, but it also provides functionality as a cloud-computing service, hosted in Microsoft datacenters around the globe. SQL Azure is generally a good fit any time you need database services. It enables you to build, host and scale applications in Microsoft datacenters. They require no up-front expenses, no long term commitment, and enable you to pay only for the resources you use. The various features of Microsoft Azure are as follows:

- Built on Microsoft SQL Server.
- High availability and scalability.
- Limited or no database administration.

- RDBMS services, like Constraints, Transactions, Stored procedures, triggers etc.
- Variety of tools to connect with SQL Azure, like Database account portal, SQL Server Management Studio, Project Houston, SQLCMS etc.
- Microsoft SQL Azure Data Sync CTP allows SQL Azure databases to be shared across multiple data centres and thereby, providing data synchronization and data management capabilities bi-directionally.
- Microsoft SQL Azure Reporting enabled us to use Azure Reporting tools instead of relying on SSRS.

The Cloud services available from Microsoft Azure and Amazon Web Services both offer message queues and data storage, combinations which enable a very simple SOA solution based on Command-Query Separation.

The figure 2.2 [44] shows that consumers and service providers communicate through the cloud message queuing service, using a pair of queues. One queue is public, where the service provider listens for request messages which can be sent by any consumer. The second queue is private to a particular consumer, where the consumer listens for responses from the provider.

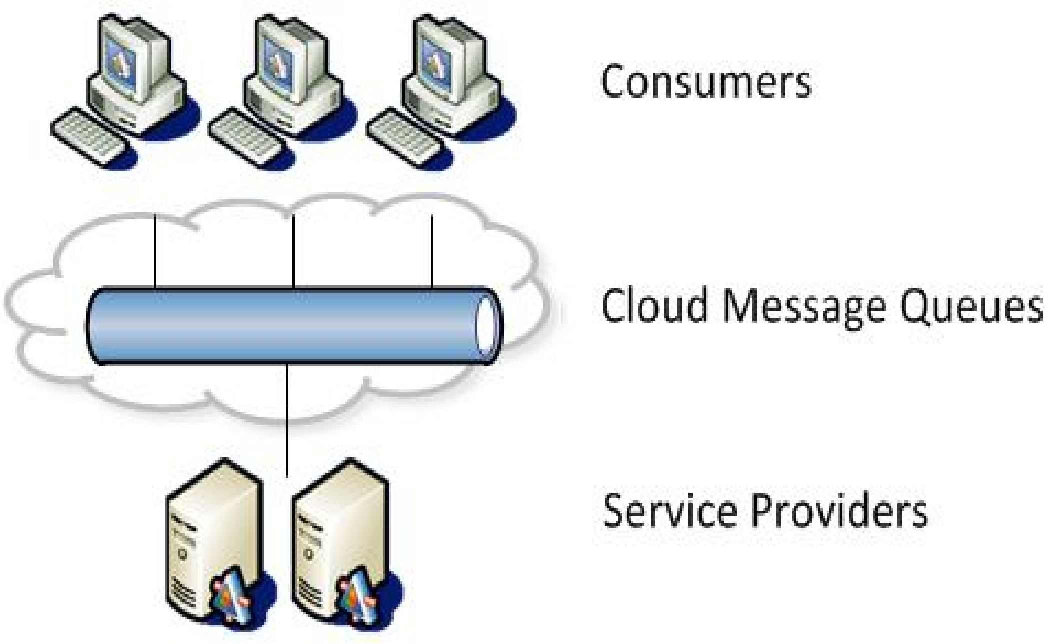


Figure 2.2 Message Queues [44]

This pattern is fully asynchronous and is all you need for Command messages - the consumer sends a command request, and continues doing what it does; the provider actions the request and sends a response, which the consumer can act on when its received.

For Query messages, the message pattern is the same, but utilises a separate service for storing and retrieving data. Figure 2.3 [44] shows that provider receives a query request message, and as part of action on it, pushes the requested data into the store. The response message sent to the consumer contains enough detail for the consumer to pull the data from the store.

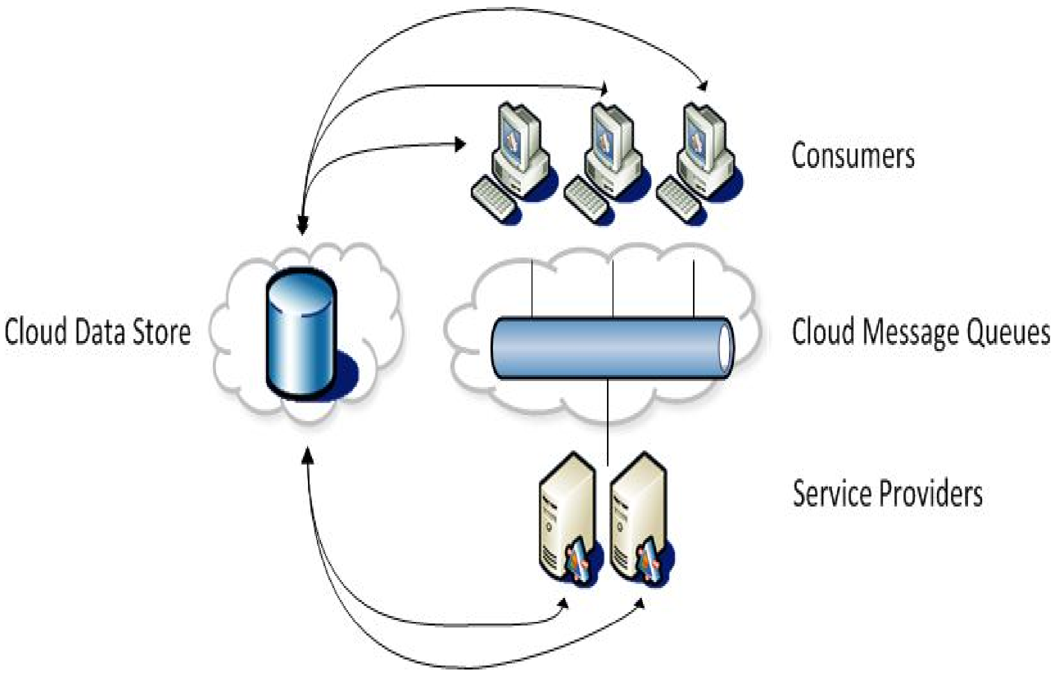


Figure 2.3 Data Storage [44]

It is important to note that the consumers and service providers are physically as well as logically separated – they can be on completely different networks with no direct link in between. This is also true of the service providers – any number of nodes can subscribe to process messages from any location. Any component can participate in the solution provided it has Internet access. The implementation of the cloud components can be left abstracted, as a third-party service the actual implementation is not relevant for the design.

The Amazon Simple DB and Microsoft Azure provide the database-as-a-service on the pay-per-usage basis. They don't provide Open Source Software which can help for research purpose. The Apache provides the Open Source Software Hadoop which helps in the research and production which is discussed as follows:

2.2.3 Apache Hadoop

Hadoop is an open-source Java-based software platform developed by the Apache Software Foundation. It lets one easily write and run distributed applications on large computer clusters to process vast amounts of data (see Figure 2.4 [39]). Hadoop implements Google's MapReduce programming model on top of a distributed file system called the Hadoop Distributed File System (HDFS).

Architecture of Hadoop:

HDFS has master/slave architecture. An HDFS cluster consists of a single NameNode – a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. The Architecture of Hadoop is shown in Figure 2.4

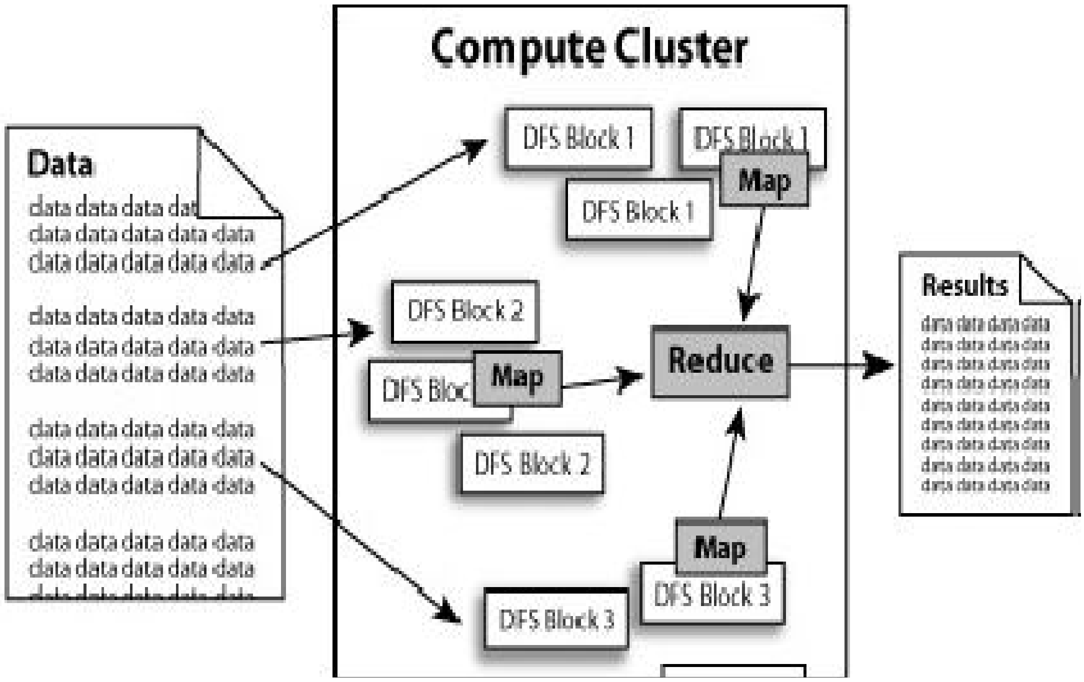


Figure 2.4 Hadoop Architecture [39]

HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

MapReduce divides applications into many small blocks of work. HDFS creates multiple replicas of data blocks for reliability, placing them on compute nodes around the cluster. MapReduce can then process the data where it is located.

- Map, written by a user of the MapReduce library, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I, and passes them to the reduce function.
- Reduce, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values.

The Map-Reduce framework consists of a single master JobTracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

2.2 Cloud DBMS Tools

The available cloud Database management tools can be categorized as Parallel DBMS tools and MapReduce based software tools as follows:

- **Parallel DBMS**: Parallel database systems stem from research performed in the late 1980s. These systems all support standard relational tables and SQL, and implement many of the performance enhancing techniques developed by the research community over the past few decades. The examples of parallel databases used in cloud are Vertica, DBMS-X, Mysql etc.
- **MapReduce**: MapReduce was introduced in 2004. It is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks [3].
- **Comparing MapReduce and Parallel DBMS on the Cloud DBMS properties**: The Cloud Data Management software should meet properties as

discussed in section 2.1.2. Based on these properties, Parallel DBMS and MapReduce are evaluated in the Table 2.2:

Table 2.2 Evaluation of Existing DBMS Tools

Properties	**Parallel DBMS**	**MapReduce**
Efficiency	Can be provided with additional complexity	Dependent upon application
Fault Tolerance	Need to restart a query upon a failure	Taken as High priority
Heterogeneity	Designed to run on homogenous environment	Designed to run in heterogeneous environment
Business Intelligence Interface	Does not easily interface	Carefully optimized and certified BI interfaces
Data Encryption Support	Works with hand-coded encryption support	Unable to operate on encrypted data

The Table 2.2 shows that Fault Tolerance, heterogeneity, BI support are provided by MapReduce software while the efficiency and Data Encryption support are provided by Parallel DBMS.

2.3 Virtualization

Virtualization [45] is a proven software technology that is rapidly transforming the IT landscape and fundamentally changing the way that people compute. The computer hardware was designed to run a single operating system and a single application, leaving most machines vastly underutilized. Virtualization lets you run multiple virtual machines on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer.

2.3.1 Working of Virtualization

The virtualization lets you transform hardware into software. It uses software to transform or “virtualize” the hardware resources of an x86-based computer—including the CPU, RAM, hard disk and network controller—to create a fully functional virtual machine that can run its own operating system and applications just like a “real” computer. Multiple virtual machines share hardware resources without interfering with each other so that you can safely run several operating systems and applications at the same time on a single computer. The Figure 2.5 [45] shows the running of an application on single hardware with multiple virtual machines.

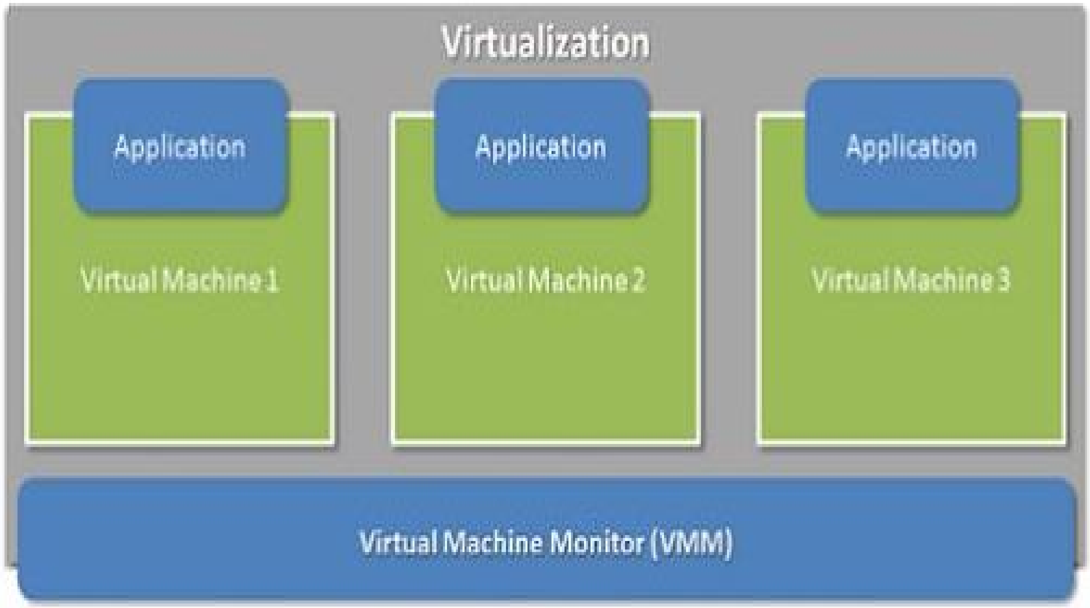


Figure 2.5 Virtualization [45]

2.3.2 Benefits of Virtualization

The virtualization provides following benefits [45]:

- Run multiple operating systems on a single computer including Windows, Linux and more.
- Reduce capital costs by increasing energy efficiency and requiring less hardware while increasing your server to admin ratio.
- Ensure your enterprise applications perform with the highest availability and performance.
- Build up business continuity through improved disaster recovery solutions and deliver high availability throughout the data centre.
- Improve enterprise desktop management & control with faster deployment of desktops and fewer support calls due to application conflicts.
- Enhance the performance of your applications by fully understanding best practices in deploying and optimizing a virtualized infrastructure. Learn more about virtualization performance.

2.3.3 Virtualization and Cloud DBMS

The Cloud provides the services in the distributed environment which requires the availability of the multiple machines. The virtualization enables us to work on multiple virtual machines on a single physical system. So, virtualization helps the researchers to analyze the cloud environment on single machine which leads to reduction in cost, increase the availability, performance and better analysis.

The above discussion shows that managing the data on the cloud environment has several challenges as the database systems are difficult to scale, configure, maintain and increases the cost for provisioning of peak time load. The transactional database applications are not a severe problem but the analytical and relational database would encounter the scalability limit at some point. The Parallel DBMS lacks in Fault tolerance, ability to work in heterogeneous environment and support for Business Intelligence and data encryption while the MapReduce are not as much efficient and does not provide data encryption support as provided by the Parallel DBMS. So, there is a need for Hybrid architecture which can combine the properties of both the tools available. The most of the Database-as-a-Service providers are not Open Source, while Apache Hadoop is open source project which can be used for research purpose to analysis of database applications in the cloud environment. Virtualization helps to create a cluster of nodes on a single machine which enhances the performance of the application by fully understanding the best practices on virtualized infrastructure.

Chapter 3
Problem Statement

Previous chapter discussed about the management of data in cloud computing and classification of tools available for managing data on the cloud. This chapter focuses on problem statement taken up in the thesis.

3.1 Gap Analysis

In today's state-of-the-art corporate and scientific environment generates enormous amounts of data. Collecting and analyzing this data is getting increasingly complex, because of the amount of data and the complexity of the analysis. The management of data center is very costly and consumes a lot of energy. Another problem in the cloud environment exits is the increasing criticality of data; all enterprises should have a disaster recovery plan in place for key applications. The cloud computing by leveraging the database management resources saves energy and cost. The literature review shows that the traditional approaches are not sufficient enough to provide all properties of cloud DBMS. The comparison of both approaches shows that the MapReduce framework can provide BI Support, Data encryption support, and does not directly support processing of related homogeneous datasets which is a common need.

3.2 Objectives

The objectives of the thesis are as follows:

i) Implementing an architecture combining the properties of MapReduce and Mysql in virtual environment.
ii) The MapReduce framework is extended for relational datasets.
iii) The designing and deployment of database application while improving homogeneity of MapReduce for the relational data sets.

3.3 Requirements for Database Systems on Cloud

Database systems provide an extremely attractive interface for managing and accessing data, and have proven to be widely successful in many financial, business, and Internet applications. The development and management for the data centre results in increase of labour cost and high energy consumption. The database system should be provided on the pay-as-per-use basis. The Cloud computing provides access to large pools of data and computational resources through a variety of interfaces. The available MapReduce and Parallel DBMS approaches are not sufficient enough to fulfil all the properties of cloud database management. An application is designed to extend the MapReduce to provide the support for the heterogonous datasets using Map-Reduce-Merge phase.

The requirements of hybrid architecture which can combine the properties of both are discussed below:

Table 3.1 shows the experimental platforms used in the thesis to implement Virtualized Hybrid Architecture using Hadoop and deploying the database application.

Table 3.1 Platform Configurations

Type	Specifications
Processor	AMD Turion (TM) 64X2 Mobile Technology
CPU Speed	2.20GHz
Memory	4.00 GB
Platform	32-bit Operating System
Operating System	Ubuntu 10.04

Table 3.2 shows the software packages used in for the architecture.

Table 3.2 Software Versions

Software Packages	Versions
VMware Workstation	6.5
Hadoop	0.20.2
OpenJDK	1.6.0_20
Eclipse	1.6.
Mysql	5.0
IBM MapReduce Plug-In	6.5

Chapter 4
Design and Implementation

This chapter discusses how the problem stated in the previous chapter can be designed and implemented.

4.1 System Design: Virtualized Hybrid Architecture

The Table 2.2 in the literature survey shows that neither the Parallel DBMS nor MapReduce like software are sufficient to manage the database applications in the cloud environment. So, a completely hybrid architecture which can combine the properties of both is required. In this work, the design of the Cloud Hybrid Architecture in virtualized environment using Hadoop which combine the properties of MapReduce and Mysql has been proposed as shown in Figure 4.1.

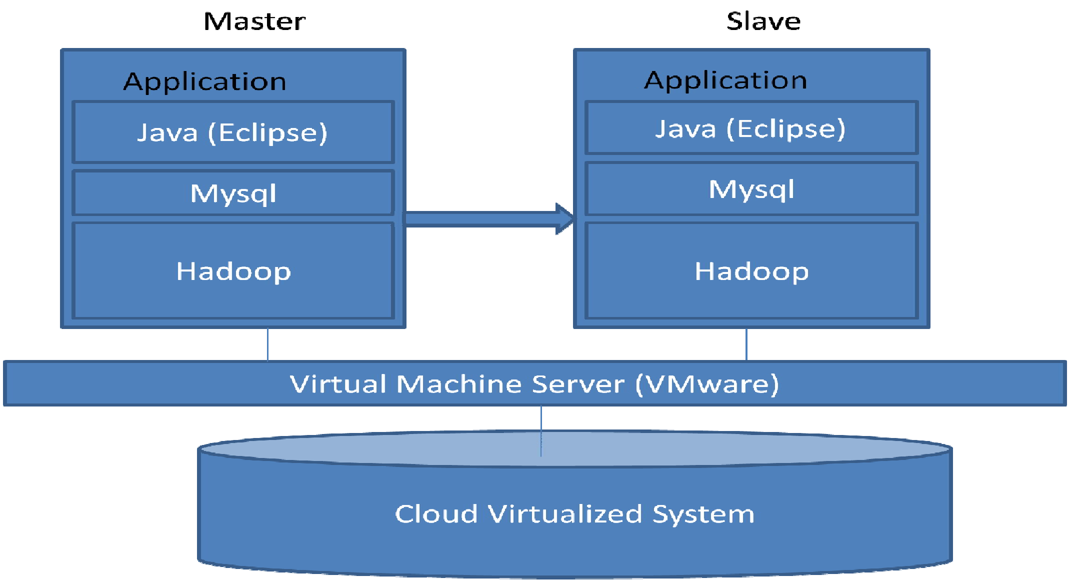


Figure 4.1 Hybrid Virtualized Architecture

The Hybrid Virtualized Architecture includes Virtual Machine Workstation (VMWare) to provide virtualization, Hadoop framework to provide MapReduce functionality, and to manage the database Mysql is used. The Eclipse is used as Java Integrated Development Environment to write function program. The virtual environment helps the researchers for analysing the cloud environment for different types of application on a single machine. This is a mater-slave architecture where the master node provides the functionality to the slave node by providing fault tolerance as if one node is failed the data can be executed on the other slave nodes. The cluster is setup between both the machines. Virtualized Hybrid Architecture consists of hosting server installing VMware and two hosted VMs (master and slave) on which an Ubuntu 10.04 OS and database application are running.

4.2 Implementation of Proposed Virtualized Hybrid Architecture

The following tools have been used for implementing the above proposed Virtualized Hybrid architecture:

- **VMware Workstation**

VMware workstation is desktop software that allows running multiple x86-compatible desktop and server operating systems simultaneously on a single PC, in fully networked, portable virtual machines with no rebooting or hard drive partitioning required [27]. The VMware workstation is shown in figure 4.2.

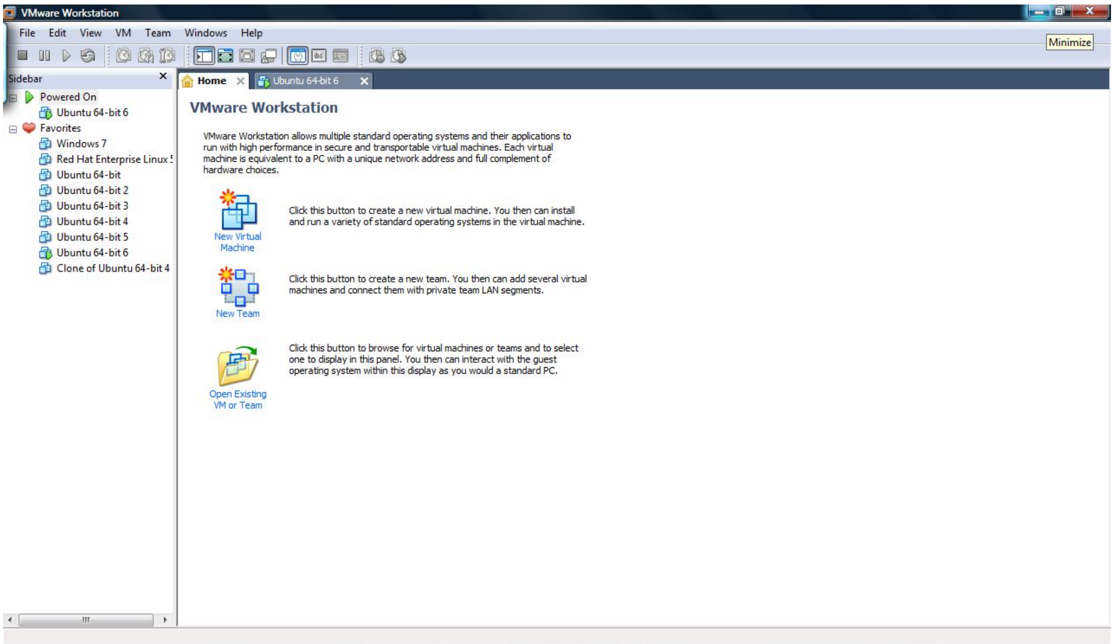


Figure 4.2 VMware Virtual Environment

Product Benefits: Workstation is used in the software development, quality assurance, training, sales, and IT fields. Easy to develop and test multiple operating systems and applications on a single PC. It is useful for testing multitier configurations. Workstation streamlines software development and testing. It also facilitates testing using multiple snapshots and debugging support. Workstation enhances productivity of IT professionals by configuring and testing desktops and servers as virtual machines before deploying them to production.

- **Ubuntu**

Ubuntu is an operating system based on the Debian GNU/Linux distribution and distributed as free and open source software. It is named after the Southern African philosophy of Ubuntu ("humanity towards others"). Ubuntu is designed primarily for desktop use although net book and server editions exist as well. Ubuntu software centre gives instant access to thousands of applications that are needed to customize the computer.

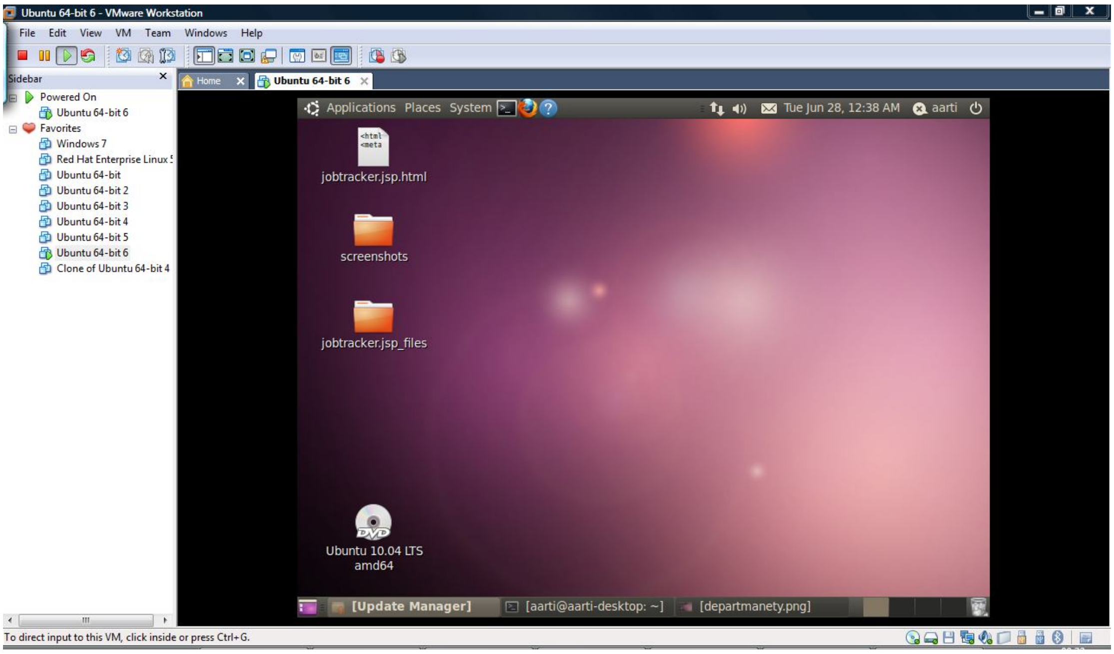


Figure 4.3 Ubuntu in Virtual Environment

In this thesis, Ubuntu 10.04 has been installed on all the two virtual machines. The ISO image of Ubuntu-10.04-desktop-i386 is downloaded and installed on the VMware. It is fast, secure, easy to install and easy to use operating system.

- **OpenJDK**

OpenJDK is a development environment for building applications, applets, and components using the Java programming language. The packages are built using the IcedTea build support and patches from the IcedTea project. It is the result of an effort Sun Microsystems began in 2006. The implementation is licensed under the GNU General Public License (GPL) with a linking exception, which exempts components of the Java class library from the GPL licensing terms.

- **Eclipse**

Eclipse as an integrated development environment (IDE) for Java. Eclipse is created by an open source community and is used in several different areas, e.g. as IDE for Java or for Android or as a platform to develop Eclipse RCP applications, etc. Eclipse requires an installed Java Runtime.

- **Mysql**

The **Mysql is** a (RDBMS) that runs as a server providing multi-user access to a number of databases. The Mysql provides the Database connectivity through SQL support.

- **Hadoop MapReduce**

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. In literature review chapter the architecture of Hadoop is discussed in detail.

The detailed description of installation of above tools on VmWare are given in the Appendix-I.

4.3 Case Study: Student Result Evaluation System

In this section, a case study of Student Result Evaluation application has been considered for the experimental evaluation of the proposed architecture.

The Student Result Evaluation System stores the information about the students and various courses such as Roll_No, Student_name, Marks, Course_Id, Course_Name, and Course_Credits. The total marks are evaluated on the basis of marks gained by the student in mid semester and end semester terminals. The grades are also given on the basis of total marks. The total credits of the course are calculated by adding lecture, theory and practical credits. The final result is evaluated in CGPA (Commulative Grade Points) on the basis of Grades and Course_credits. The Student Result

Evaluation system is analysed and designed in the MapReduce framework as discussed below:
There are two datasets in this case study: Student and Course. Student's "key" attribute is Roll_No and the others are packed into Stud_Info i.e. "value". The "key" attribute for Course is Course_ID and the others are packed into a Couse_Info i.e. "value". The query is to join these two datasets and compute the result of Student.

The database schema of the Student Result Evaluation system is shown with help of ER diagram, and the UML diagrams and the Information Flow diagram for the application determines the application from different perspectives. The MapReduce Functions and experimental results are also shown in this section.

4.3.1 ER Diagrams

The Entity-Relationship Diagram shows the relation student with the course as below:

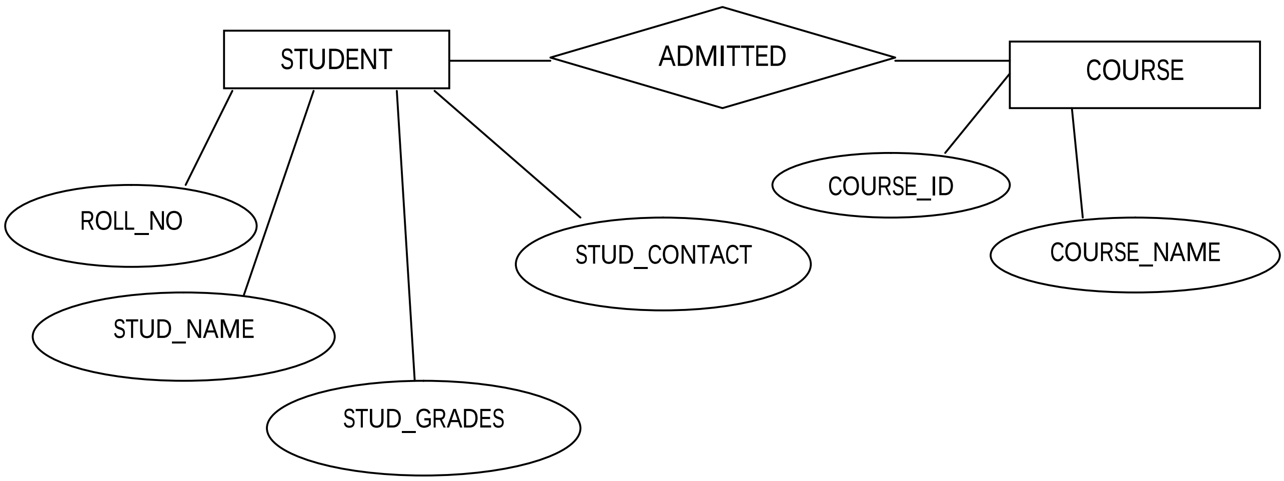


Figure 4.4 E-R Diagram of Employee Payroll System

4.3.2 UML Diagrams

The UML diagram provides the view of an application from different perspectives. This section gives the Use Case, Activity, and deployment diagram from various stakeholders' perspective.

- **Use Case Diagrams**

Use case diagrams present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. It is used to gather requirements of the system including internal and external influences. Figure 4.5 shows the use case diagram for Student

Result evaluation system from Faculty and admin view.

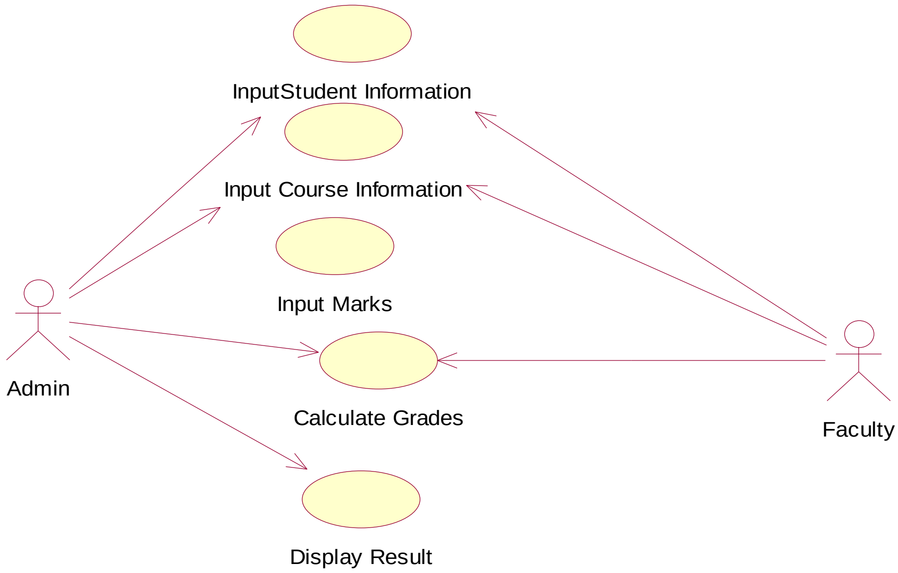


Figure 4.5 Use Case Diagram for Student Result Evaluation System

- **Activity Diagram:**

  The activity diagram shows the flow of the activities to calculate CGPA of the students.

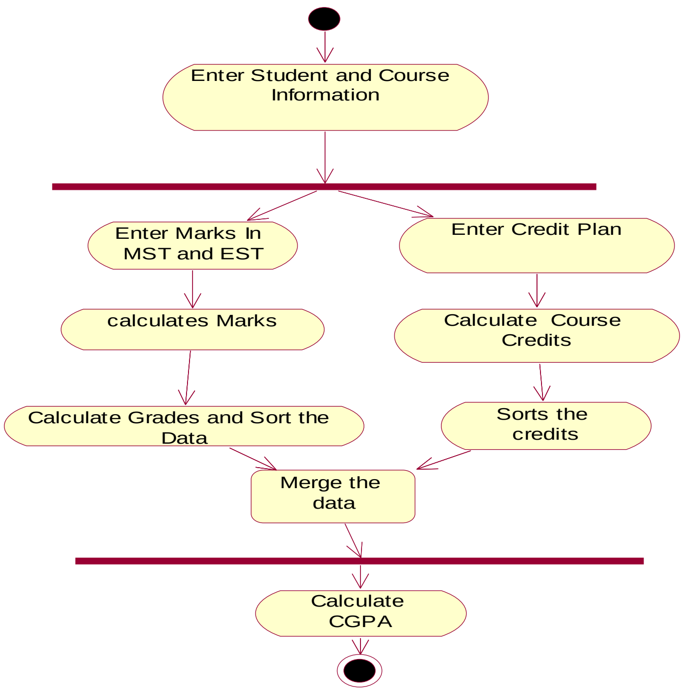


Figure 4.6 Activity Diagram

- **Deployment Diagram**

Deployment Diagram models the physical deployment of artifacts on nodes. Deployment diagram depicting hardware components how application is deployed is shown in Figure 4.7

Figure 4.7 Deployment Diagram

4.3.3 Information Flow Diagram

The Information flow diagram for the above query is shown in the Figure 4.8. On the left Hand Side, mapper reads Student entries and computes the Marks of Mid-Term and End Terminals for each entry. A reducer then sums up these Marks for every student and sorts them according to Roll_No. On the right hand side, a mapper reads Course entries and computes Course credits. A reducer then sorts these Course entries.

At the end, a merger matches the output records from the two reducers on Roll_No using the sort merge algorithm, and evaluates the CGPA of the students from Course_Credits and Grades.

Roll No	STUD_INFO: COURSE_ID	STUD_INFO: Marks
105	2001	44+22
102	1045	43+43
204	4532	34+34
303	4001	38+23

Mapper Computes Marks of Students

Roll_No	Course_ID	Sub_Total
105	2001	66
102	1045	86
204	4532	68
303	4001	61

Reducer Computes Final Grades and sorts according to Roll_No

Roll_No	Course_ID	Grades
105	2001	B
102	1045	A
204	4532	B
303	4001	B

Course_ID	Course_INFO	Credits_PLAN (LTP)
2001	SOFTWARE ENGG.	2 1 1
1045	PROJECT MANAGEMENT	2 1 1
4532	SOFTWARE ARCHITECTURE	2 2 1
4001	SOFTWARE TESTING	2 2 1

Mapper Computes Credits of Course

Course_ID	Course_INFO	Credits
2001	SOFTWARE ENGG.	4
1045	PROJECT MANAGEMENT	4
4532	SOFTWARE ARCHITECTURE	5
4001	SOFTWARE TESTING	5

Reducer Calcutes total credits and sorts according to Course ID

Course_ID	Credits
1045	4
2001	4
4001	5
4532	5

Merger Calcutes CGPA and sorts according to Roll_No

Roll_No	CGPA
102	3.2
105	4.0
204	4.0
303	4.0

Figure 4.8 Flow of information in MapReduce Frame Work

4.3.4 Map-Reduce-Merge functions for Student Result Evaluation:

The map-reduce-merge functions of Student Result Evaluation are given as follows:

- **Map Function on Student Record:**

This algorithm calculates the Mid-Term and End-Term marks of the students where student Roll_No is the primary key and Marks is the output value.

```
Map (const Key& key, /*Roll_No*/

Const Value& value /*Stud_Info*/) {

Roll_No=key;

Course_ID=value.Course_id;

/*compute Marks using Stud_Info*/

Output_key = (Course_ID, Roll_No);

Output_value = (marks);

Emit (output_key, output_value);

}
```

- **Map function for the Course Dataset.**

This algorithm calculates the credits of each course from the Credit_PLAN. The Course_ID is the Key attribute and Course_INFO contains the course name and course Duration. The Course_ID with its credits is the output.

```
Map (const Key& key, /*Course_ID*/
const Value& value /*Course_INFO){
Course_ID=key;
Credit_PLAN= value.Credit_PLAN;
Emit (Course_ID, (Credits));


}
```

- **Reduce function for Student dataset:**

This algorithm calculates the grades of the course from Marks.

reduce (const Key& key, /*(Course_ID, Roll_No) */

Const ValueIterator& value /*an iterator for calculating Marks*/)

{

Sub_total= /*sum of marks for each Roll_No*/

Emit (key, (Grades));

}

- **Reduce Function for Course Dataset:**

This algorithm sorts the course data set.

reduce (const Key& **key**, /*(Course_id)*/

const ValueIterator& **value**

/* an iterator on a Credit_PLAN and sorts Credits */

Emit (key, (Credits));

}

- **Merge Function for Student-Course Dataset**

This merge algorithm calculates the CGPA of the student by multiplying the Credits and Marks Grades.

Merge (const Leftkey**& Leftkey,** /*(Course_ID, Roll_No)*/

Const LeftValue& **leftValue** /*Grades*/

Const RightKey& **rightKey** /*course ID*/

const RightValue& rightValue /*Credits*/ {

if (leftKey. Roll_No == RightKey**) {**

CGPA= leftValue * rightValue;

Emit (leftKey.emp_id, bonus)**; }**

}

4.3.5 Experimental Results

The implementation of Virtualized Hybrid Architecture is shown in Appendix I. The screenshots of experimental results of the Student Result Evaluation application on Virtualized Hybrid architecture using MapReduce programming framework are shown below:

- The Figure 5.1 shows the Student marks evaluation. The student Roll No., name and Information about the marks are taken as the input. The map function will combine the Marks in the MST and EST and gives the total marks as an output. The Reducer computes the Grades and sorts them in the database according to the Roll_No.

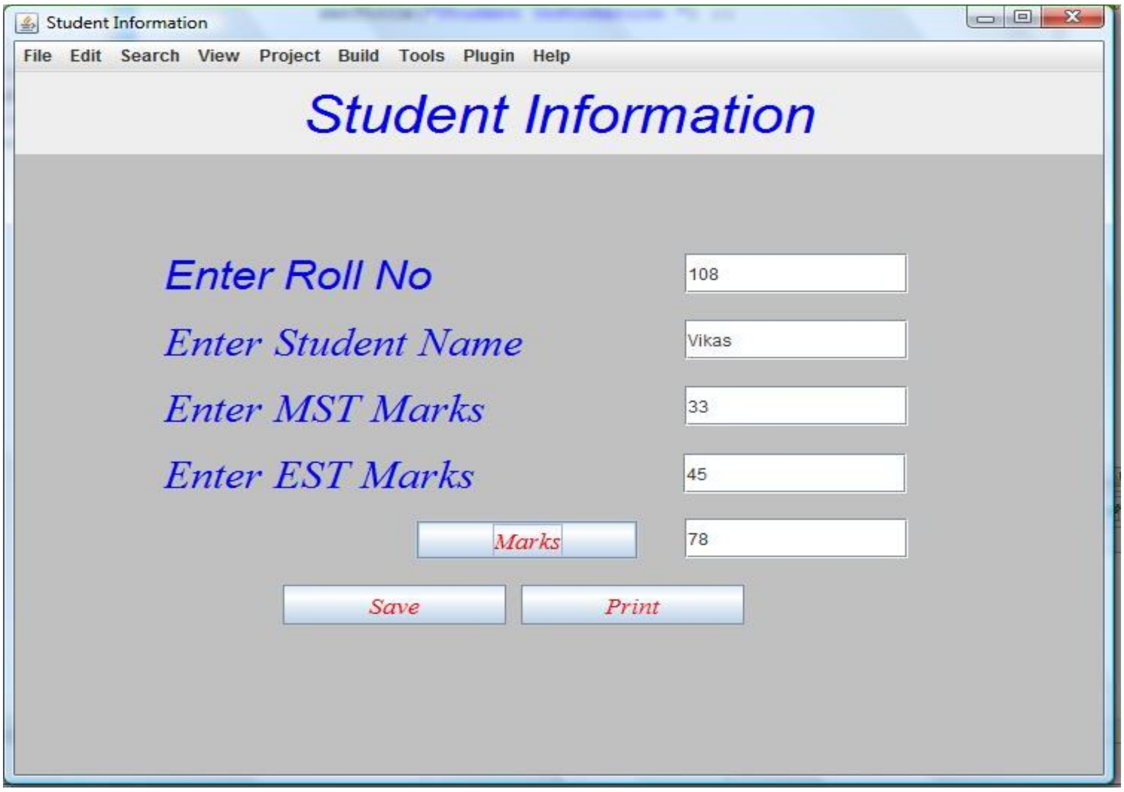


Figure 5.1 Student Information and Marks

- The Figure 5.2 shows the Course information with final credits. The Course ID, name of the course, and its credits in lecture, theory and practical are taken as input. The map function combines the credits while reducer sorts the data according to Course_id and stores it in database.

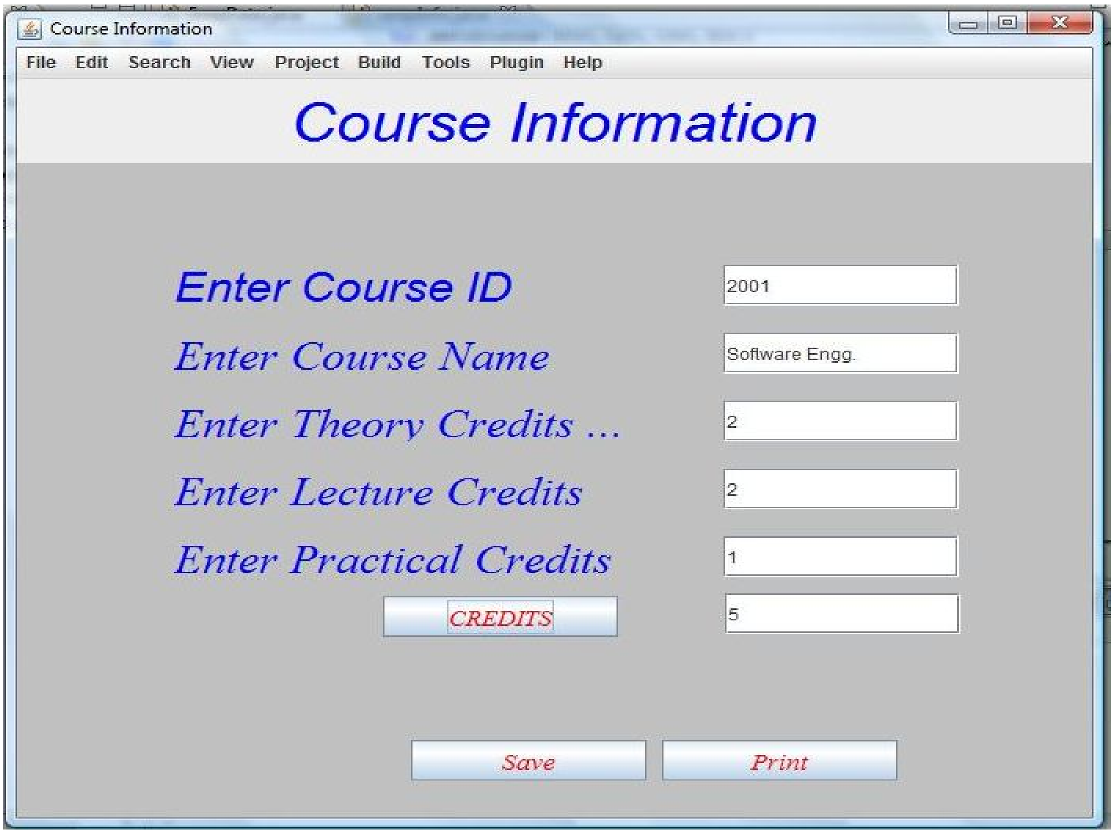


Figure 5.2 Course Information

- The 5.3 evaluates the result of the students. The Roll_No is taken as input. The merger function will calculate the CGPA by combining the course credits and Grades and sorts the data according to Roll_No.

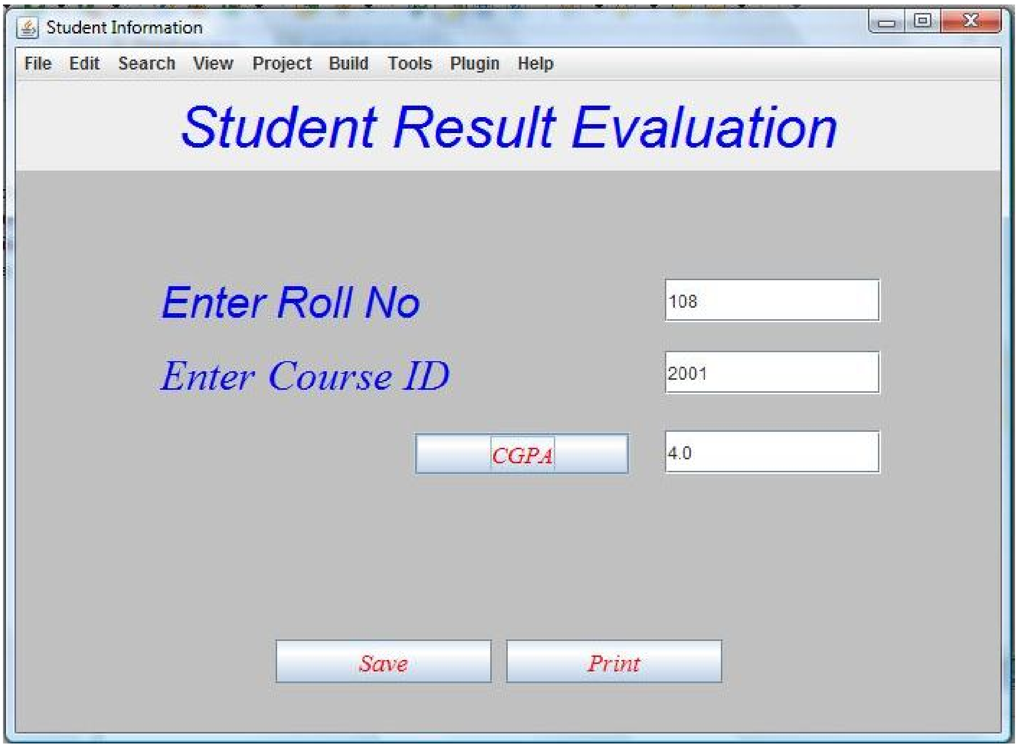


Figure 5.3 Student Result Evaluations

Chapter 5
Conclusion and Future Scope

This chapter discusses the conclusions of work presented in this thesis. The chapter ends with a discussion of the future direction which thesis work can take.

6.1 Conclusion

This thesis discusses about the evolution of cloud computing form grid computing, introduction of cloud computing, its service and deployment models, benefits and challenges of cloud computing. It focuses on Cloud Database management (C-DBMS), challenges, types and tools for managing data on the cloud. The virtualization and need for virtualization in the cloud database management is also discussed. The CDBMS tools are compared against Cloud DBMS properties concluding that there is a need for hybrid architecture which can combine the properties of traditional DBMS tools. The virtualized cloud Hybrid architecture is proposed and implemented. The MapReduce framework is not able to work in the homogeneous environment, which is improved through Map-Reduce-Merge framework. This extension is shown with the help of analysis, design and implementation of Student Result Evaluation System. In conclusion, the thesis work shows that how the relational database application can be deployed in the cloud environment using MapReduce framework.

6.2 Thesis Contribution

- Parallel and MapReduce software are compared on the basis of Cloud DBMS properties.
- The Virtualized Hybrid Cloud Architecture is proposed implemented using Hadoop framework.
- The MapReduce functionality is extended to work in Homogenous environment.
- The Student Result Evaluation Application is analysed, designed and deployed on Virtualized Hybrid Cloud Architecture.

6.3 Future Scope

- The Virtualized Hybrid Architecture can be used to analyze and design the other application which can support Data Encryption.
- The thesis extends the Homogeneity feature of MapReduce, other features can also be considered for research purpose such as Business intelligence support and heterogeneity.
- The Fault Tolerance of Mysql can be improved with MapReduce Framework.
- This architecture can be used to analyze complex analytical application also.

References

[1]. Lenk A., Klems A., Nimis J., Tai S., and Sandholm T., "What's inside the cloud? An architectural map of the cloud landscape". In CLOUD '09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, Washington, DC, USA. IEEE Computer Society, 2009.

[2]. Caceres J., Vaquero L.M., Merino L.R. and Lindner M.," A break in the clouds: towards a cloud definition". SIGCOMM Computing, Commnun Rev, 2009.

[3]. Dean J., Ghemawat S., "MapReduce: Simplified data processing on large clusters", in: Proc. 6th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2004, San Francisco, USA, Dec. 2004.

[4]. Abouzeid A., K. Bajda-Pawlikowski, D. J. Abadi, Silberschatz A., and Rasin A.. "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads". In VLDB, 2009.

[5]. Pavlo A., Rasin A., Madden S., Stonebraker M., D. DeWitt, E. Paulson, L. Shrinivas, and D. J. Abadi. "A Comparison of Approaches to Large Scale Data Analysis". In Proc. Of SIGMOD, 2009.

[6]. Liladhar R. Rewatkar and Ujwal A. Lanjewar. "Data management in market oriented cloud". Advances in Computational Sciences and Technology ISSN 0973-6107 Volume-3, 2009.

[7]. Abouzied A., Kamil Bajda-Pawlikowski, Huang J., Daniel J. Abadi, Silberschatz A., Yale University. "HadoopDB in Action: Building Real World Applications". SIGMOD'10, Indianapolis, Indiana, USA. , June 6–11, 2010.

[8]. Tuan Viet – DINH, "Cloud Data Management". ENS de Cachan, IFSIC, IRISA, KerData Project-Team. Jan, 2010.

[9]. INFOSYS, "Cloud Computing". Business Innovations through Technology, SETLabs Briefings Volume No 7, 2009.

[10]. DeWitt D. and Stonebraker M., "MapReduce: A major step backwards". DatabaseColumn Blog. http://www.databasecolumn.com/2008/01/mapreduce-a-major-step-back.html.

[11]. Armbrust M., Fox A., Griffith A., Joseph A. et al, "Above the Clouds: A Berkeley View of Cloud". EECS, University of California, Berkeley, Technical Report No. UCB/EECS-2009-28, February 10, 2009.

[12]. Buyya R., “Market-Oriented Cloud Computing: Vision, Hype and Reality for Delivering IT Services as Computing Utilities” 2008.

[13]. Landis C. and Blacharski D., “Cloud Computing Made Easy, Version0.3, www.cloudpedia.com.

[14]. Jiang D., Chin B. O., Shi L., Wu S., “The Performance of MapReduce: An Indepth Study”. Proc. The 36th International Conference on VLDB, 1317, Singapore. September, 2010.

[15]. http://www.thecloudtutorial.com/cloudtypes.html, cited on 15 Nov, 2010

[16]. Chris Grzegorczyk Graziano Obertelli Sunil Soman Lamia Youself Daniel Nurmi,Rich Wolski and Dmitrii Zagorodnov. “The Eucalyptus open source cloud computing system. Cluster Computing and the Grid”, IEEE International Symposium on, 2009.

[17]. Foster I., Yong Zhao, Raicu I., and S. Lu. “Cloud computing and grid computing 360-degree compared”. In Grid Computing Environments Workshop, 2008. GCE '08, pages 1-10, Nov. 2008.

[18]. Yuefa D., Wu Bo, Yaqiang G., Quan Z., Chaojing T.. “Data Security Model for Cloud Computing”, Proce. of the 2009 IWSA, China, November 21-22, 2009.

[19]. Abouzeid A., K. Bajda-Pawlikowski, D. J. Abadi, Silberschatz A., and Rasin A.. “HadoopDB: Need for speed in analytics”. Yale University,Hadoop NYC Meet up, 4 May, 2010.

[20]. Daniel J. Abadi, Yale University, New Haven, CT, USA. “Data Management in the Cloud: Limitations and Opportunities”, 2009.

[21]. Ubuntu Server Guide, 2010 is available at “https://help.ubuntu.com/10.04/serverguide/C/serverguide.pdf”.

[22]. VMware user’s manual is available at “http://www.vmware.com/pdf/ws32_manual.pdf”.

[23]. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic,” Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems”, Elsevier Science, Amsterdam, The Netherlands, June 2009.

[24]. Junjie Peng, Xuejun Zhang, Zhou Lei, Bofeng Zhang, Wu Zhang, Qing Li, “Comparison of Several Cloud Computing Platforms”, Second International Symposium on Information Science and Engineering.

[25]. Qi Zhang, Lu Cheng, Raouf Boutaba, “Cloud computing: state-of-the-art and research challenges”, Springer, pp 7-18, Apr 2010.

[26]. P.Mell and T.Grance, “The NIST definition of Cloud Computing” , National Institute of Standards and Technology, Information Technology Laboratory, Version 15, July 2009.

[27]. B.Botelho, “VMware Entering Final Phase of Virtualization Evolution: Cloud Computing”, IT Knowledge Exchange, May 2008.

[28]. Carlo Curino, Evan Jones, Yang Zhang, Eugene Wu, and Samuel Madden,” Relational Cloud: The Case for a Database Service”, Computer Science and Artificial Intelligence Laboratory Technical Report, March14, 2010.

[29]. D. B. Lomet, A. Fekete, G. Weikum, and M. J. Zwilling, “Unbundling transaction services in the cloud”, CIDR Perspectives, 2009.

[30]. Robert L. Grossman and Yunhong Gu, “On the Varieties of Clouds for Data Intensive Computing”, IEEE Data Engineering Bulletin, vol. 32 no. 1, March 2009.

[31]. Borthakur D., “The Hadoop Distributed File System: Architecture and Design”, 2007.

[32]. Das S., Elmore A., Wang S., Agrawal D., Abbadi A. E., ”Autonomic, Elastic, Fault-tolerant, Scalable, and Secure Data Management in the Cloud”, University of California at Santa Barbara, Santa Barbara, CA 93106-5110, USA.

[33]. Agrawal D., Abbadi A. E., Das S.,”Big Data and Cloud Computing: Current State and Future Opportunities”. University of California at Santa Barbara, Santa Barbara, CA 93106-5110, USA.

[34]. Robin B., “What is Cloud Database? Suitability of Algebraix Data’s Technology to Cloud Computing”, The Bloor Group, White Paper, 2011.

[35]. David C. Wyld, “Moving to the cloud: An Introduction to Cloud Computing in Government”, IBM centre for The Business of Government e-Goverment Series, 2009.

[36]. Branter M., Graf David, Florescu D., et al, “Building Database in the cloud”, Systems Group, ETH, ZURICH.

[37]. NCHC “Programming Hadoop with eclipse” [Online Document] Available: http://trac.nchc.org.tw./cloud, 2008.

[38]. Hadoop. http://hadoop.apache.org/.

[39]. Hadoop MapReduce Tutorial. http://hadoop.apache.org/core/docs/current/mapred tutorial.html.

[40]. MapReduce. http://labs.google.com/papers/mapreduce.html.

[41]. The Hadoop Distributed File System: Architecture and Design. http://hadoop.apache.org/core/docs/current/hdfs design.html.

[42]. D. Jeff and G. Sanjay. "MapReduce: Simplified Data Processing on Large Clusters." Communications of the ACM, 51, 2008.

[43]. Yang H., Dasdan Ali, Parker D.S. and Hsiao R. , "Map-Reduce-Merge: Simplified Relational Data Processing on large clusters". SIGMOD'07, China, 2007.

[44]. http://geekswithblogs.net/EltonStoneman/archive/2010/07/24/cloud-services-and-command-query-separation-part-1.aspx, [Online], 14 Jul, 2011.

[45]. AaSys, "Virtualization Basics", Issue 9, Volume 6, September 2008.

[46]. http://aws.amazon.com/simpledb/ cited on 28th October, 2010.

[47]. http://www.microsoft.com/windowsazure/ cited on 15 November, 2010.

Appendix I
Environmental Setup of Virtualized Hybrid Architecture

The experimental set up of Virtualized Hybrid Architecture given as follows:

A.1 Cluster Setup in VMware:

To set up the cluster on the VMware, install two or more Ubuntu virtual machines with same configuration (name) and assign IP address as follows:

a) Go to VPN configurations>Network Connection> Wired connection> Add.

A pop up window will appear for editing the wired connections. Select IPv4 settings tab and choose the method as manual as shown in the screenshot below.

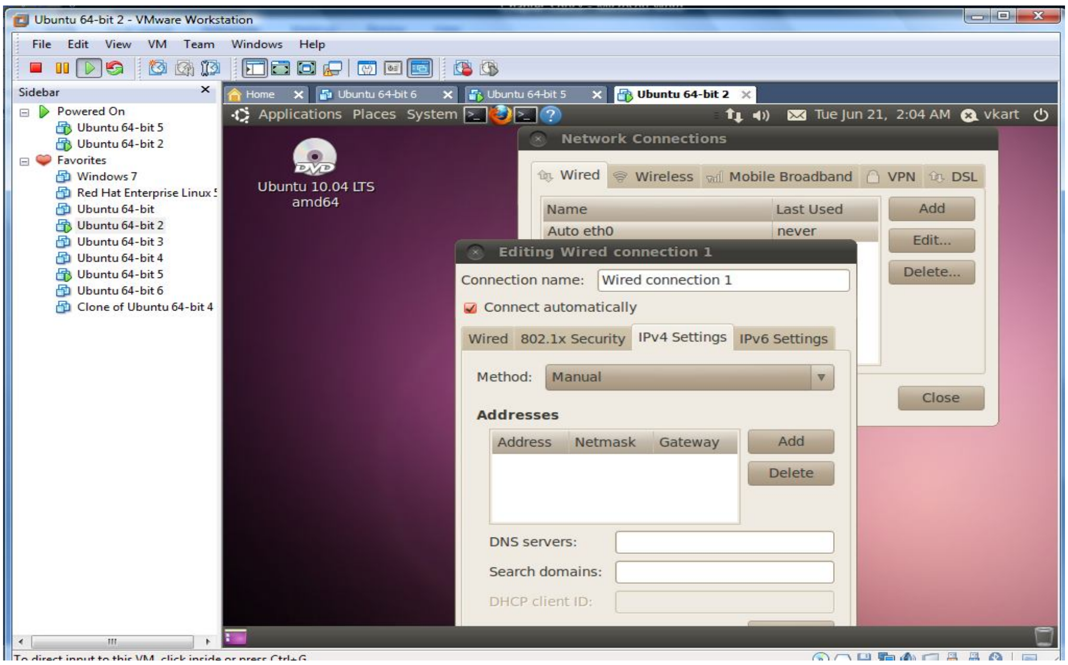


Figure A.1 VPN configuration

b) Provide the IP address, Netmask and gateway as 192.168.220.150, 255.255.255.0 and 0.0.0.0 respectively.

c) Update the contents of /etc/hosts file on master node with IP addresses of Master and slave nodes as: (Master Node Only)

192.168.220.150	master
192.168.220.149	slave

d) Go to terminal and ping the machines form master to master and mater to slave to check if they are connected or not.

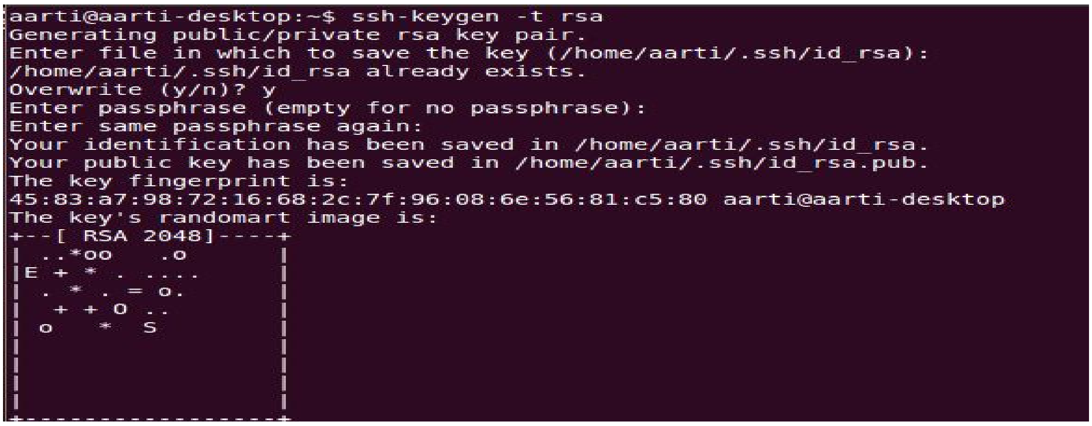
A.2 Implementation of Hadoop:

a) Download Hadoop 0.20.2 and copy the .zar file at a location such as /home/aarti/hadoop

b) Generate an SSH key for the user: SSH, which is an acronym for Secure Shell, was designed and created to provide the best security when accessing another computer remotely. Not only does it encrypt the session, it also provides better authentication facilities, as well as features like secure file transfer, X session forwarding, port forwarding and more so that you can increase the security of other protocols. The ssh key is generated as shown below:

$ ssh -keygen –t rsa - P ""

OUTPUT:



```
aarti@aarti-desktop:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aarti/.ssh/id_rsa):
/home/aarti/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aarti/.ssh/id_rsa.
Your public key has been saved in /home/aarti/.ssh/id_rsa.pub.
The key fingerprint is:
45:83:a7:98:72:16:68:2c:7f:96:08:6e:56:81:c5:80 aarti@aarti-desktop
The key's randomart image is:
+--[ RSA 2048]----+
| ..*oo    .o     |
|E + * . ....     |
| . * . = o.      |
|  + + O ..       |
| o   *   S       |
|                 |
|                 |
|                 |
|                 |
+-----------------+
```

Figure A.2 Key Generation

c) Create Password less ssh: The hadoop user on the master (aka hadoop@master) must be able to connect a) to its own user account on the master – i.e. ssh master in this context and not necessarily ssh localhost – and b) to the hadoop user account on the slave (aka hadoop@slave) via a password-less SSH login.

$ scat ~/. ssh /id_r sa.pub > > ~.ssh/authorized_keys

$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hadoop@slave

$ ssh master

OUTPUT: Connecting master to master:

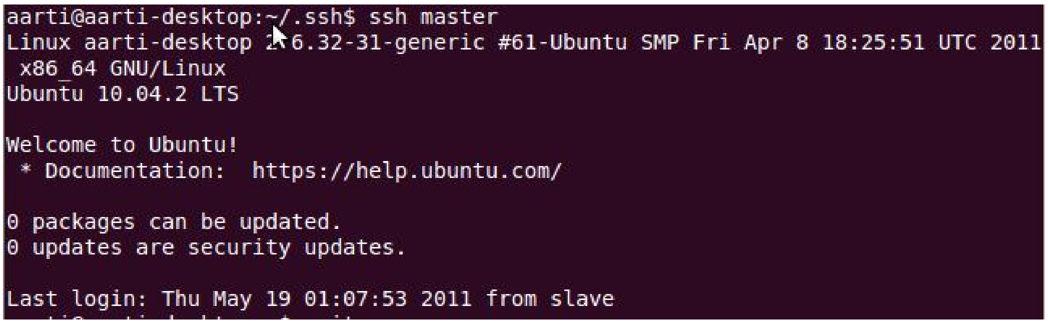


Figure A.3 ssh master

$ ssh slave

OUTPUT: Connecting master to slave.

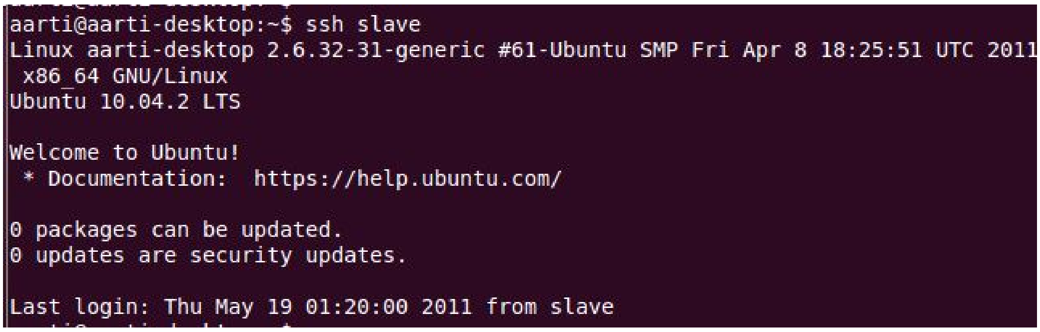


Figure A.4 ssh slave

a) Configuration of Hadoop: The Hadoop is configured by updating the some files in Hadoop/conf folder on all machines as follows:

- Hadoop/conf/env.sh

```
# The java implementation to use.  Required.
# export JAVA_HOME= /usr/ lib/j2sdk1.5 - sun

To

# The java implementation to use. Required.

export JAVA_HOME= /usr /lib/jvm/openjdk-6-java

export HADOOP_HOME= /home /aarti/hadoop

export HADOOP_LOG_DIR=$HADOOP_HOME/logs

export HADOOP_SLAVES=$HADOOP_HOME/conf / slaves
```

- Update /hadoop/conf/core-site.xml with the contents below:

```
<configuration>
 <property>
 <name>hadoop.tmp.dir</name>
 <value>/opt/hadoop-data/tmp-base</value>
 <description>A base for other temporary directories</description>
 </property>
 <property>
 <name>fs.default.name</name>
 <value>localhost:54311</value>
 <description>
 The name of the default file system.  A URI whose scheme and authority determine
the FileSystem implementation. The url's scheme determines the config property
(fs.SCHEME.impl) naming the FileSystem implementation class. The url's authority
is used to determine the host, port, etc. for a filesystem.
 </description>
 </property>
```

- Change /hadoop/conf/mapred-site.sh

```
<configuration>
 <property>
 <name>mapred.job.tracker</name>
 <value>localhost:54310</value>
 <description>
 The host and port that the MapReduce job tracker runs.  If "local", then jobs are run
in-process as a single map and reduce task.
 </description>
</property>
</configuration>
```

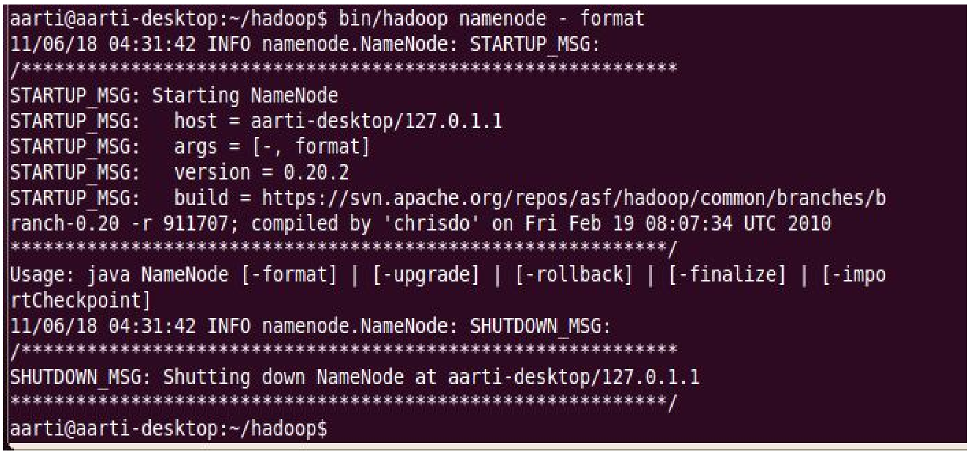
- Update / hadoop/conf/hdfs-site.sh with contents below:

```
<configuration>
 <property>
 <name>dfs.replication</name>
 <value>1</value>
  <description>
  Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
 </description>
 </property>
</configuration>
```

d) Formatting namenode: Before starting new multi-node cluster, Hadoop's distributed file System (HDFS) has to be formatted for the namenode. A running Hadoop namenode should not be formatted as this will erase all the data in the HDFS file system.

```
$ bin/hadoop namenode  -  format
```

OUTPUT:



```
aarti@aarti-desktop:~/hadoop$ bin/hadoop namenode - format
11/06/18 04:31:42 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = aarti-desktop/127.0.1.1
STARTUP_MSG:   args = [-, format]
STARTUP_MSG:   version = 0.20.2
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/b
ranch-0.20 -r 911707; compiled by 'chrisdo' on Fri Feb 19 08:07:34 UTC 2010
************************************************************/
Usage: java NameNode [-format] | [-upgrade] | [-rollback] | [-finalize] | [-impo
rtCheckpoint]
11/06/18 04:31:42 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at aarti-desktop/127.0.1.1
************************************************************/
aarti@aarti-desktop:~/hadoop$
```

Figure A.5 Format namenode

- **Start up Hadoop**

Starting the cluster is done in two steps. First, the HDFS daemons are started: the namenode daemon is started on master, and datanode daemons are started on all slaves (here: master and slave). Second, the MapReduce daemons are started: the jobtracker is started on master, and tasktracker daemons are started on all slaves.

$ ./start-all.sh

OUTPUT:

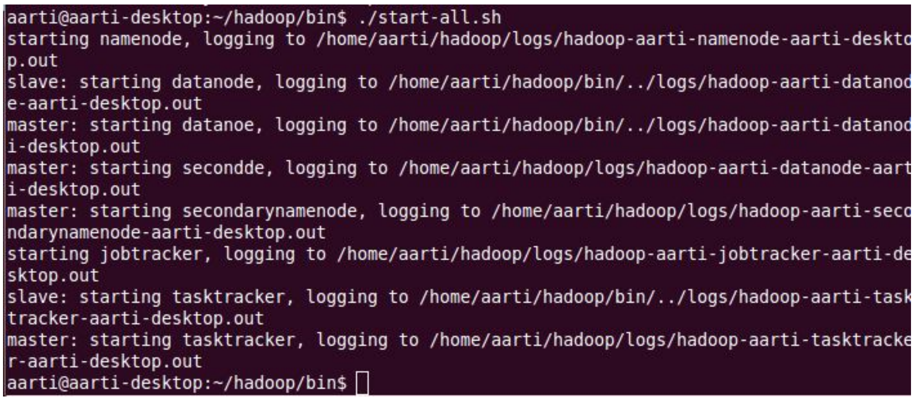


Figure A.6 Start Hadoop

- **Stop Hadoop**

  This will shut down HDFS by stopping the namenode daemon running on the machine you ran the previous command on, and datanodes on the machines listed in the conf/slaves file.

$ ./stop-all.sh

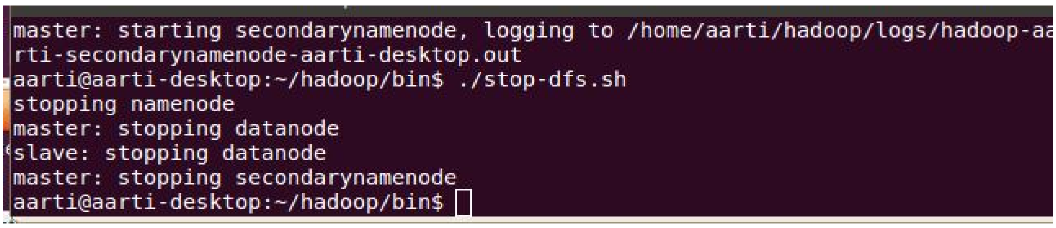


Figure A.7 Stop Hadoop

The Hadoop is installed, now the eclipse should be configured with Hadoop.

A.3 Installation and configuration of Mysql

The Mysql provides the Database connectivity through SQL support. To install Mysql type the command written below:

$ sudo apt-get install mysql-server

A prompt will appear asking password for root as shown in the fig. below. It will also ask to retype the password. Fill the password and remember it for further usage.

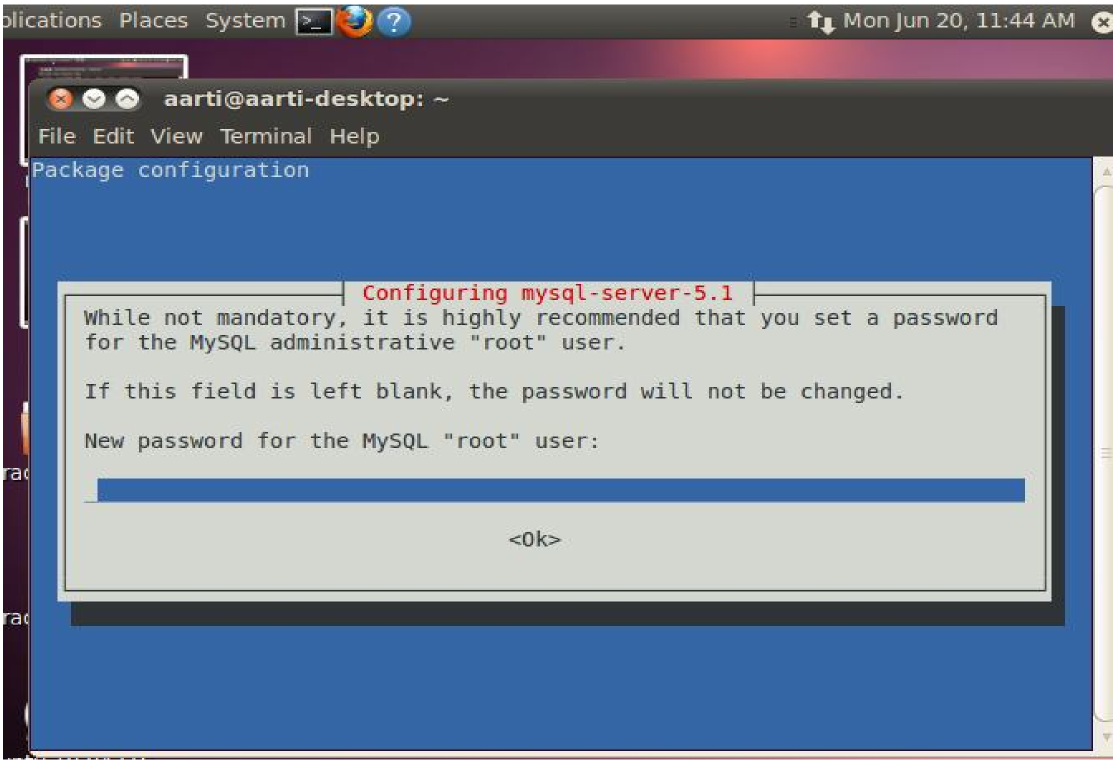


Figure 4.4 Input Password

To configure Mysql with Hadoop download the JDBC connector from the link: "http://www.mysql.com/downloads/connector/j/" and copy the Jar file in the Hadoop's lib directory.

A.4 Setting up Eclipse with Hadoop

a) The eclipse should be installed and set the following Java Class Path in the etc/bash.bashrc file.

```
$ export JAVA_HOME= /usr/ lib/jvm/java-6-sun

$ export HADOOP_HOME= / home/aarti/hadoop/

$ export CLASSPATH=.:$JAVA_HOME/lib/d t.jar:$JAVA_HOME/lib/tools.jar
```

b) Download the IBM MapReduce Tools zip file from http://www.alphaworks.ibm.com/tech/mapreducetools and extract to /tmp/.

```
$ cd / tmp /
$ unzip mapreduce_tools.zip
$ mv plugins/com.ibm.hipods.mapreduce* /usr/ lib/eclipse/plugins/
```

c) Restart the eclipse. Now, Eclipse can run the Mapper and Reducer programs.

The Student Result Evaluation application is a sample application to be run on the hybrid architecture. To run application, connection to the Hadoop file system should be established. Hadoop should be working on the nodes. The steps to connect the Hadoop file system are given below:

a) Go to Window>Show view> Other>Mapreduce Tools>Mapreduce Servers as shown in the fig 5.8.
b) Now, Go to file> New>Project>Mapreduce Project. Fill the project name and write the Mapper and reducer functions in java.

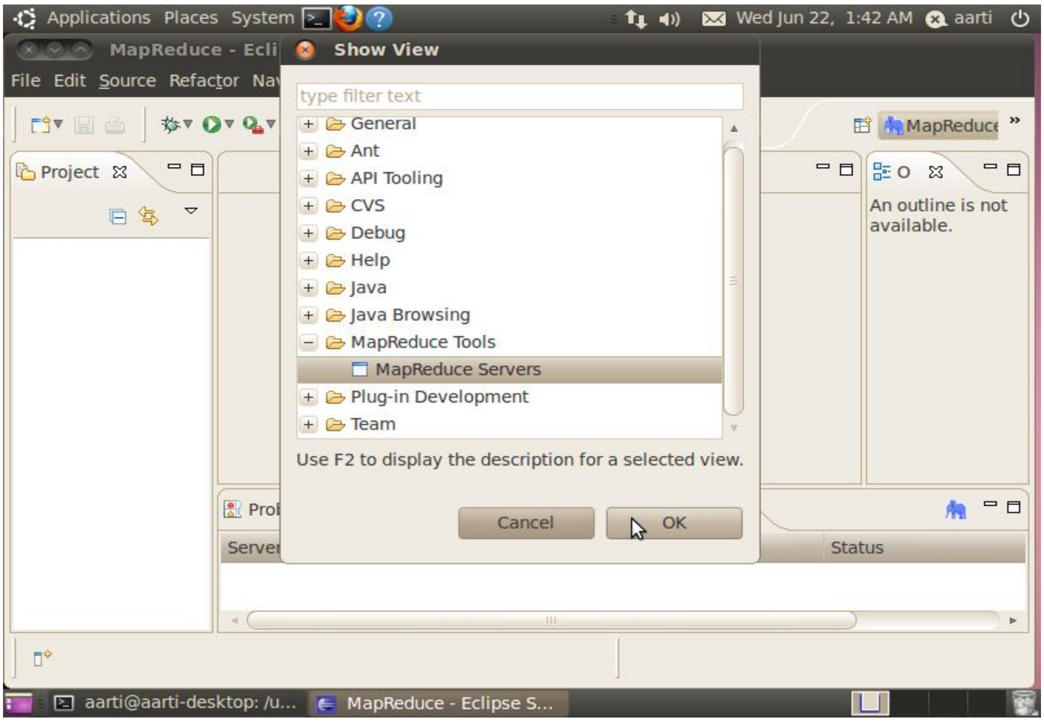


Figure A.9 Enable MapReduce Servers

c) To execute the application, go to Run As> Run on Hadoop>Choose the Hadoop server.