جـامـعـة الـشـرق الأوسط
MIDDLE EAST UNIVERSITY

# Comparative Analysis of Database Performance for Well-known Cloud Computing Environments

**التحليل المقارن لاداء قواعد البيانات في بيئات الحوسبة السحابية المعروفة**


By

**Jasour Ahmed Obeidat**

Supervisor

**Dr. Ahmad Kayed**


This thesis is submitted to the Department of Computer Information Systems, Faculty of Information Technology, Middle East University in partial fulfillment of the Requirements for Master Degree in Computer Information Systems.


Department of Computer Information System

Faculty of Information Technology

Middle East University


May, 2014

ii

## اقرار تفويض

أنا جسور أحمد محمود عبيدات، أفوض جامعة الشرق الاوسط للدراسات العليا بتزويد نسخ من رسالتي ورقيا و الكترونيا للمكتبات أو المنظمات أو الهيئات و المؤسسات المعنية بالابحاث والدراسات العلمية عند طلبها.

**الاسم** : جسور أحمد محمود عبيدات

**التاريخ** : ٧/٥/٢٠١٤

**التوقيع** : ..............................

جسور عبيدات

7/5/2014

iii

## Authorization Statement

I, Jasour Ahmed Mahmoud Obeidat, authorize the Middle East University to supply a copy of my thesis to libraries, establishments or individuals upon their request.

**Name** : Jasour A.Obeidat

**Date** : 7.5.2014

**Signature** : ..............................

iv

## Examination Committee Decision

This is to certify that the thesis entitled "Comparative analysis of database performance for well-known cloud computing environments" was successfully defended and approved on 7th May 2014.

*(Head of the Committee)*

**Prof. Reyadh Naoum.**

Department of Computer Science ..................

Middle East University ..................

*(Supervisor)*

**Dr. Ahmad Kayed.**

Associate Professor ..................

Dean Faculty of IT ..................

*(External Examiner)*

**Dr. Hussein Al-Bahadili.**

Associate Professor ..................

Petra University ..................

# Acknowledgment

I would like to express my sincere appreciation to Dr. Ahmad Kayed for his guidance, support and motivation throughout my Master’s Thesis.

# Dedication

- To my parents for their support.
- To my wife for her support and patience.
- And to my sons "Ahmed" and "Ameer".

# Table of Contents

**Chapter Three: The Proposed Model and Experiment Design**



**Chapter Four: Experimental Results**

**Chapter Five: Conclusion and Future Works**

# List of Figures

# List of Tables

# List of Abbreviations

| Abbreviation | Meaning |
|---|---|
| **IT** | Information Technology |
| **CSP** | Cloud Service Provider |
| **SaaS** | Software as a Service |
| **PaaS** | Platform as a Service |
| **IaaS** | Infrastructure as a Service |
| **NIST** | National Institute of Science and Technology |
| **CPU** | Central Processing Unit |
| **KVM** | Kernel Virtual Machine |
| **VM** | Virtual Machine |
| **SLA** | Service Level of Agreement |
| **HPC** | High Performance Cloud |
| **I/O** | Input / Output |
| **SSH** | Secure Shell |
| **dom0** | Domain 0 |
| **dom U** | Domain U |
| **QEMU** | Quick Emulator |
| **LXC** | Linux Container |
| **GUI** | Graphical User Interface |
| **RDBMS** | Relational Database Management System |
| **KPI** | Key Performance Indicator |
| **CLI** | Command Line Interface |
| **DDL** | Data Definition Language |
| **DML** | Data Manipulation Language |
| **DQL** | Data Query Language |

| | |
|---|---|
| **DB** | Database |
| **ERD** | Entity Relationship Diagram |

# ABSTRACT

## Comparative Analysis of Database Performance for Well-known Cloud Computing Environments

Prepared By:
Jasour A. Obeidat

Supervised By:
Dr. Ahmad Kayed

Cloud computing paradigm reduces hardware costs, software costs, and provides efficient use of services in internet environment. Virtualization technology is the main component of cloud computing technology. Thus, several types of virtualization techniques (hypervisors) are available to be used in cloud environments. Cloud computing technology serves databases for private, public and hybrid clouds. Each type of hypervisor has its own interaction with database depending on the type of query that is executed.

The main aim of this thesis is to study the performance of the hypervisors with cloud managers for each type of SQL query. To be able to meet this aim it has been outlined the different types of cloud managers, hypervisors, and query types. The findings of literature review have been deployed to design several experiments for this research. Two experiments have been conducted to evaluate the response time for both CPU and memory.

The results showed that the type of hypervisors have an effect on the performance of SQL query types with regards to CPU and memory response times. Thus, The Mean (⍰)was taken to make the comparison between hypervisors and The Coefficient of Variation (CV) was taken to check the comparison results. Thus, Xen hypervisor had a positive effect based on query type. Hence, KVM hypervisor had a negative effects based on query type. Xen hypervisor affect the performance of query types with 11%, 20%, and 3% for DDL, DML, and DQL respectively. KVM hypervisor affect the performance of query types with 17%, 25%, and 4% for DDL, DML, and DQL respectively. In addition, control parameter was taken which it is dataset size; to realize the experiment results. The domain of dataset size was taken as three stages 1100, 2200, and 3300 rows respectively. Thus, the results show negative effect after doubling and tripling the size of data set.

# الملخص

## التحليل المقارن لاداء قواعد البيانات في بيئات الحوسبة السحابية المعروفة

اعداد :
جسور أحمد عبيدات

باشراف:
د. أحمد كايد الكايد

ان هرم الحوسبة السحابية يقلل من تكاليف المعدات، تكاليف البرمجيات، و يوفر أفضل استخدام للخدمات في بيئة شبكة الانترنت . تعتبر تقنية الافتراضية من أهم المكونات الاساسية لتقنية الحوسبة السحابية. وهكذا، عدة أنواع من تقنيات الافتراضية (مراقبات الاجهزة الافتراضية ) متوفرة للاستخدام دخل بيئات الحوسبة السحابية . توفر تقنية الحوسبة السحابية امكانية خدمة قواعد البيانات في السحب الخاصة، العامة، والهجينة . كل مراقب من مراقبات الاجهزة الافتراضية يتفاعل مع قاعدة البيانات اعتمادا على نوع الاستعلام المنفذ.

الهدف الرئيسي من هذا البحث هو دراسة أداء مراقب الاجهزة الافتراضية و مدراء السحابة الالكترونية لكل استعلام منفذ على قواعد البيانات . ومن أجل مقابلة هذا الهدف لقد تم تفصيل عدد من مدراء السحب الالكترونية ومراقبات الاجهزة الافتراضية و أنواع الاستعلامات المنفذة على قواعد البيانات . من النتائج البحث في ا لدراسات السابقة تم تصميم عدة تجارب من أجل هذا البحث . تم أجراء تجربتين لتقييم أداء وحدة المعالجة المركزية و الذاكرة الرئيسية من حيث زمن الاستجابة لكل منهما.

أظهرت النتائج بأن أنواع الاستعلامات و أنواع المراقبات الاجهزة الافتراضية لها تأثير مباشر وواضح على الأداء من حيث زمن الاستجابة لكل من وحدة المعالجة المركزية والذاكرة الرئيسية . وهكذا، تم اخذ معامل الاختلاف (CV) لاجراء المقارنة بين مراقبات الاجهزة الافتراضية . وبينت النتائج أن المراقب (Xen) له تأثير أجابي اعتمادا على نوع الاستعلام . حقق المراقب (Xen) معامل اختلاف بمقدار 11%، 20%، و 3% على الترتيب للمجموعات DDL، DML، و DQL. بالمقابل حقق المراقب (KVM) معامل اختلاف بمقدار 17%، 25%، و4% على الترتيب. تم أالاخذ بعين الاعتبار حجم البيانات في الجدول كمتغير تحكم من أجل التحكم بواقعية التجارب في الدراسة . وهكذا، بينت النتائج أن بعد مضاعفة حجم الجدول يتاثر زمن الاستجابة لكل من وحدة المعالجة المركزية والذاكرة الرئيسية سلبيا.

# CHAPTER ONE

# Introduction

## 1.1. Overview

This chapter explains background about cloud computing, virtualization, and performance evaluation. This chapter shows the problem statement of this research, author's contribution, and the outline of thesis chapters.

## 1.2. Cloud Computing

Recently, cloud computing is taking a significant focus in the field of Information Technology (IT) as a solution for computer services and applications that can be provided to users or business organizations (Alsmadi, I., 2013). Cloud computing brought many opportunities to IT environments. Thus, many business organizations start benefit from cloud computing technology by reducing costs, using the advanced computing resources, and the ability to work with huge amount of data (Hussain, Z. and Gummadi, A. 2013).

According to the National Institute of Science and Technology (NIST) Cloud computing is "a model for enabling ubiquitous, 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" (Mell, P. and Grance, T., 2011).

Buyya (2011) explained the promises of cloud computing. Consequently, the promises summarized in satisfying the need of a large number of users around the world, finding solutions to work with huge amount of data, providing powerful computational capabilities, and delivering services in manner to meet user's expectation.

The paradigm of cloud computing is built on the topof several technological paradigms such as grid computing, utility computing, and distributed computing. Delivering reliable services that is built on virtualized compute and storage services. Furthermore, providing users the ability to access these services from anywhere in the world and following financial model through pay as you go represent satisfied functionalities in cloud computing technology (Shawish, A. and Salama, M., 2014).

Many competitors found in the area of cloud computing, they are providing cloud computing services to customer organizations such as Google, IBM, and Microsoft. Cloud Service Provider (CSP) offer services in cloud and let customers to use them and pay as they use those services. Offering services in cloud let customers the ability to use them without the need to own highly cost hardware or buy special purpose software (Abuakibash, M. and Elleithy, K. 2012).

## 1.3. Cloud Computing Service Models

Services in cloud computing was classified into three service models, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) depending on NIST. SaaS is the gate where any customer organization can access the cloud and use software as a web based service (Karthik, B. and Sri, M. 2013).

PaaS is a service model where CSP is able to provide an active operating system images in the cloud, which give the customers the ability to design, deploy, and test their own projects. In this service model customers don't have the control on computing resources available in the cloud such as Windows Azure Platform (Madhavi, K. 2012).

IaaS is a service model where the CSP provide virtual computing resources such as network bandwidth, storage memory, or processing power, in order to give customers the ability to run operating systems and software applications. CSP provide these resources in this model as a web based services. This model based on virtualization technology such as Amazon EC2 (Al Morsy, M. et al. 2010). Figure 1.1 show the main service models in cloud computing.

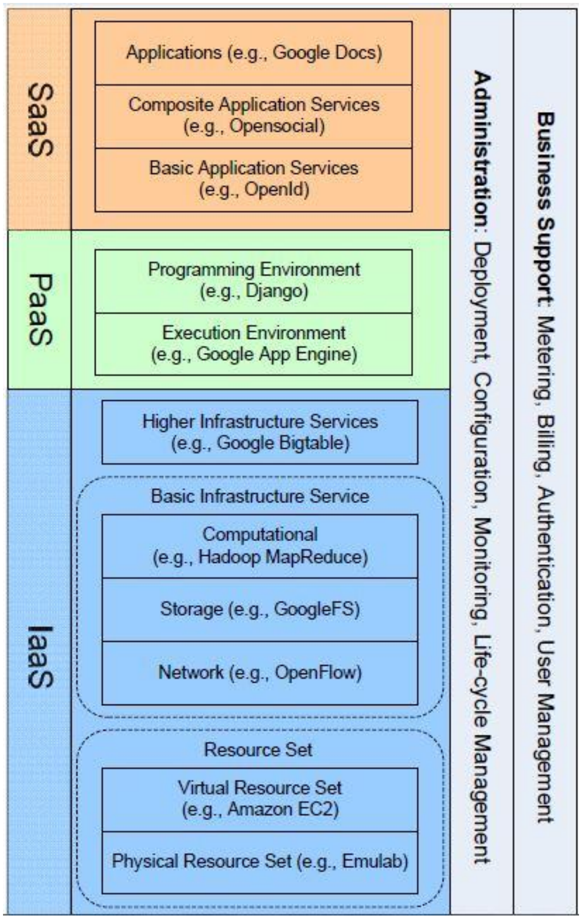


**Figure 1.1** Architecture of Cloud Computing (Xiao, Z. and Xiao, Y. 2013)

Recently, several researches mentioned a fourth cloud service model called Database as a Service (DBaaS). Thus, DBaaS is a services that is provided and managed by CSP, that support applications without the need for users administration functions. Consequently, customers in this service model don't have to think about database design, test, or maintenance (Gawande, M., and Kapse, A. 2014).Figure 1.2 shows the conceptual model that contains the core capabilities which support the delivery of DB service to an organization.

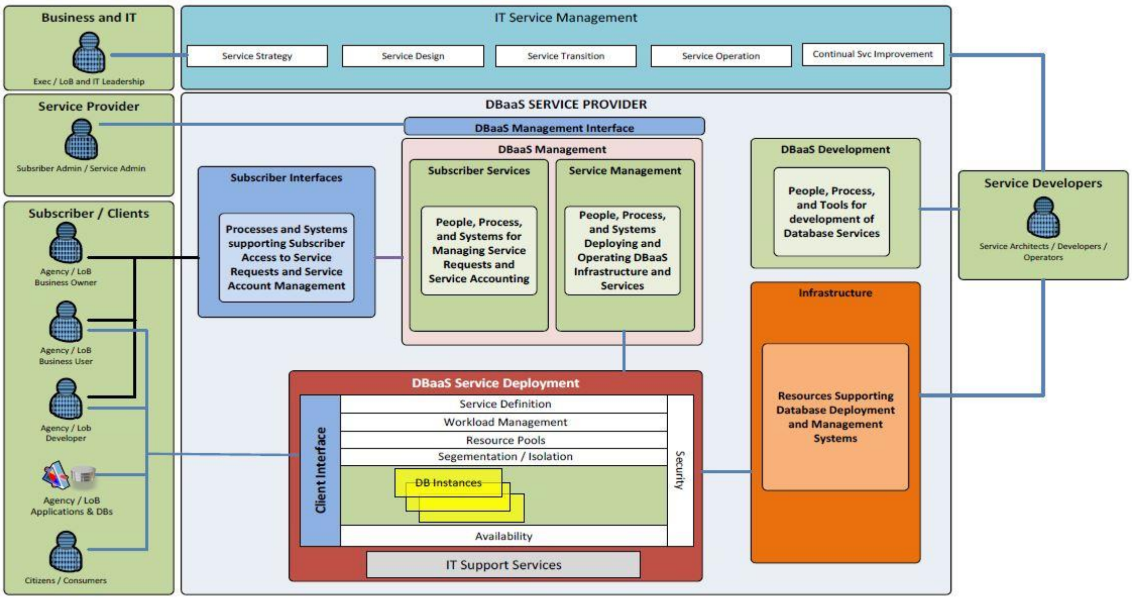


**Figure 1.2**Conceptual model of DBaaS (Oracle, 2011)

## 1.4. Types of Clouds

Four types of clouds available in the market: public, private, community, and hybrid. In public cloud, the cloud infrastructure is made available to public or large industry, and it is managed by an organization that is selling cloud services. Thus, the resources are offered as a service in the cloud. Therefore, users can set the specifications of any service and use them on demand (Goyal, S. 2014).

In private cloud, it is operated only within single organization, and it is managed by organization or third party. Hence, several aspects lead to setup a private cloud within an organization. The efficient use of in-house computing resources, saving security and privacy of an organization, reducing data transfer costs, and the full control of organization on critical activities (Dillon, T. et al. 2010).

In community cloud, the cloud infrastructure is shared by different organizations, but support specific community that has the same interests such as security, policy, requirements, and missions. Thus, it is managed by one of organization or third party (Wyld, D. 2010).

In hybrid cloud, cloud infrastructure is a composition of two or more clouds (private, public, or community), that remain a unique entities, but it is bound to others through standardize or proprietary technology, that enables applications and data portability among them (Jansen, W. and Grance, T. 2011).

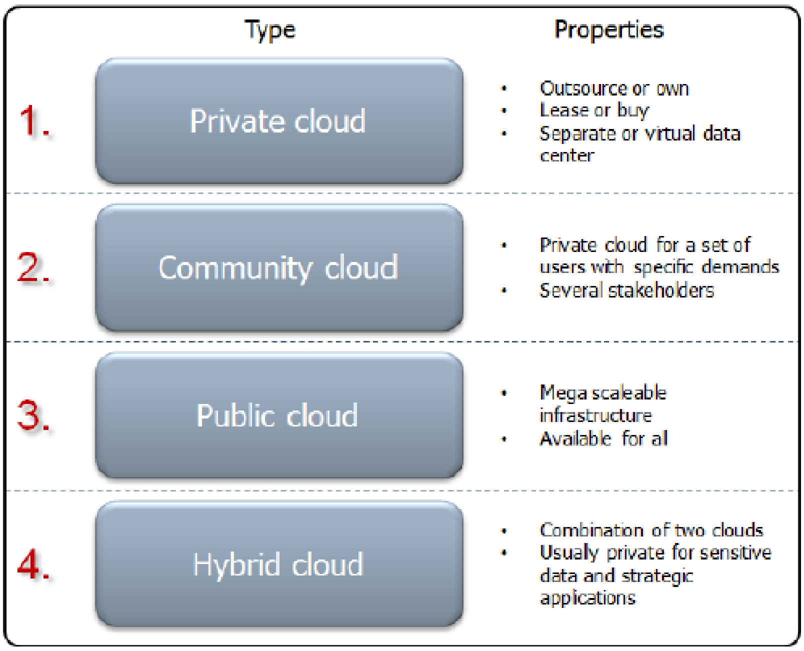


**Figure 1.3** Cloud Deployment Approach (Hashim, A. and Othman, M., 2014)

Cloud computing has eight primary attributes were represented as a drivers for the adoption of cloud computing. Hence, the attributes are availability, collaboration, elasticity, lower infrastructure costs, mobility, risk reduction, and virtualization (Lweis, G., 2010). Recently, Hashim, A. and Othman M. studied the advantages of using cloud computing. As well as, they explained the main success factors after adoption to cloud computing. Hence, minimizing IT operations budgets after centralizing software, OS, and IT infrastructure, minimizing processing time compared with traditional IT technologies, and the effective management between organization's levels (Hashim, A. and Othman, M., 2014).

## 1.5 Virtualization

The core technology in cloud computing is virtualization technology which is responsible for sharing cloud resources. Virtualization can be defined as the technology that abstract away the details of physical hardware and provides virtualized resources for high level applications (Zhang, Q. and Cheng, L., 2010). Virtualization was defined by NIST as "The simulation of software and or hardware upon which other software runs, and the simulated environment is called Virtual Machine (VM)" (Searfone, K.,Souppaya, M., and Hoffman, P.,2011). Two basic approaches of virtualization used in cloud computing, hardware virtualization and software virtualization (Zhang, L., and Zhon, Q., 2009).

Virtualization technology refers to reduce the consuming of compute resources CPU, storage, network, memory, application stack, and database capabilities, by providing them to multi-tenants subscribed in the cloud (Xing, Y., and Zhen, Y. 2012).

The idea behind virtualization is to share the capabilities of physical computers by splitting the resources among different operating systems. Virtualization in the cloud appears when customers submit their requirements, then separate virtual machine created to execute specific application (Park, J. 2012).

## Types of Virtualization

Virtualization follows three types of techniques full virtualization, paravirtualization, and software emulation (Binu, A. and Kumar, G. 2011). Full virtualization is designed to provide total abstraction of the physical system in order to create virtual machine to execute guest operating system. In this implementation no modification occurred on guest operating system. Thus, it helps to provide isolation for different applications.

Paravirtualization is designed to provide virtual machine with abstraction of hardware to each one created. In this implementation guest operating system requires modification. The guest operating system in this implementation executed on virtual machine and allowing near-native performance (ABLES, T. et al. 2005).

Software emulator is called hypervisor, and itis responsible for managing guest Operating System (OS) and control the flow of instructions between guest OSs and physical hardware such as CPU, disk storage, memory, and network interface cards (Searfone, K.,Souppaya, M., and Hoffman, P.,2011). This research used Xen and Kernel Virtual Machine (KVM) hypervisors in order to employ virtualization technology in cloud environment. This research used OpenNebula and OpenStack platforms for building two clouds in order to run computational services in both of them.

## 1.6. Cloud Computing Challenge Fields

Recently, Gary, S. et al. investigated in several challenges tackled the paradigm of cloud computing. Thus, several attributes affects cloud computing such as accountability, agility, cost, performance, assurance, security and privacy, and usability (Garg, S., 2013).

## Cloud Security

The main problems in the area of cloud computing security are: network security, interfaces, data security, virtualization governance, compliance, and legal issues (Makkar, L. and Rajput, G., 2013). Data security was represented as one of the most important security problems in cloud computing. Thus, data confidentiality, integrity, and availability were referred to cloud data security (Ajoudanian, S. and Ahmadi, M., 2012).

Gonzales N., et al. studied and analyzed the security concerns in the field of cloud computing. Therefore, the virtualization security in cloud computing was studied with regards to the isolation between VMs and hypervisors vulnerabilities. As well as, the interfaces security was studied with regard to the user's administration, controlling, and programming interfaces (Gonzales, N. et al. 2012).

## Performance Testing

Performance testing represented as one of the emerging fields in computer science. In term of software engineering, performance testing used to check the scalability of software products. Performance testing identified as a sub domain of performance engineering (Tripathi, A. et al. 2012). Performance testing defined as tests performed to determine system reaction in case of response time and satiability under a specific workload conditions. In this research, performance testing applied in term of CPU and memory response time.

## 1.7. Problem Definition

Virtualization technology offers efficient use of system resources by running multiple operating system images simultaneously on hardware. Finding the besttype of hypervisor - with regards to performance - that suits query type for a database is a challenge. Different hypervisors interact with queries in different ways; this represents a challenge in selecting hypervisor in cloud platform. The main focus is how to find the suitable hypervisor for each query type in term of performance (i.e. CPU and memory response time).This research will focus on capturing the effects of hypervisoron the applications running in VMs. This research will compare database performance by applying different types of queries on it, and measuring CPU and memory utilization for each query.

## 1.8. Research Questions

Problem will be accomplished by answering the following questions:

1. How do we select the suitable hypervisor - in term of performance -for each query type?
2. How do we select the suitable cloud manager- in term of performance – for each query type?

3. What are the issues that affect the performance of hypervisor in cloud environments?
4. What are the issues that affect the performance of cloud manager in cloud environment?

## 1.9. Motivation

The use of cloud computing paradigm is rising. Many businesses started using this paradigm because of many benefits that can provide such as lower of cost, flexibility, and on-demand computing. Cloud computing performance is one of the hottest topics in the scientific research. Several types of architectures were proposed to be used in cloud computing environments. Thus, these architectures affect the performance of cloud applications (e.g. Databases). Hence, this motivates to explain the differences in hypervisors performance in the cloud, and motivates to recommend in the selection of hypervisor depending on query types in term of CPU and memory response time.

## 1.10. Contribution

This thesis contributes the following issues:

- Detecting the factors that affect the performance of virtualization hypervisors in cloud environments.
- Explaining the criteria of choosing virtualization hypervisor depending on database query types.
- Explaining the criteria of selecting cloud managers with regards to CPU and memory response time.

## 1.11. Methodology

The methodology that was used to develop our model contains the following phases:

- Study and Analysis Phase.
- Design and Implementation Phase
- Evaluation Phase

### Study and Analysis Phase

In this phase the work was started based on the problem statement which was for selecting the suitable hypervisor and cloud manager depend on query type, cloud environments uses different type of hypervisors to create virtual machines in order to distribute instances for its customers. These instances had different type of specifications in CPU, memory, and disk storage, depending on the Service Level of Agreement (SLA) between providers and customers. The acquired information from this phase was as follows:

- Studying specifications of hypervisors used in cloud platforms.
- Understanding the effects of query type on hypervisor performance.
- Understanding the effect of query type on cloud manager performance.
- Studying the performance testing – with regard to CPU and memory response time.

### Design and Implementation Phase

This research was carried out a case study which covers building two cloud environments; we decide to use OpenNebula and OpenStack as cloud managers. Consequently, each cloud environment has two types of hypervisors for each cloud. We decide to use KVM and Xen as virtualization hypervisors. The experiments used to make fair comparison using the following steps:

- Instances specifications and its performance in the real world.
- Extracting CPU and memory response times of KVM and Xen hypervisors in order to address the strength and weaknesses in term of its compatibility with cloud environments.
- Extracting CPU and memory response times of OpenNebula and OpenStack as a hosted environment with KVM and Xen hypervisors.

### Evaluation Phase

We designed two experiments to evaluate the performance of cloud manager and hypervisors. The results were used to fill the comparison table. The evaluation results were for CPU and memory response times. The comparison held between hypervisors types after executing several query types in two clouds. SQL queries were divided into three categories Data Definition Language (DDL), Data Manipulation Language (DML), and Data Query Language (DQL). Both experiments were executed in different dataset sizes. The dataset size was used as a control parameter for checking performance results.

## 1.12. Organization of the Research

Chapter 1 – Introduction: This chapter provides an overview of the problem statement, contribution and objectives to meet.

Chapter 2 – Literature review and technology used: This chapter has for aim to explain the previous studies, and the differenttechnologies present in virtualization and cloud computing.

Chapter 3 – The proposed model and experiment's Design: This chapter describes the design of the tests performed tomeasure virtualization performance. It also describes the test case for design private cloud.

Chapter 4 –Experimental Results: This chapter evaluates the results obtainduring theimplementation of the different experiments.

Chapter 5 – Conclusion: The chapter summarizes the entire project. Also it gives acritical point of view and some recommendations for future researches.

# CHAPTER TWO

# Literature Review and Related Work

## 2.1. Overview

This chapter shows collection of the most relevant work in the literature that relate to the scope of this research. This literature review covers concepts that have been addressed in this research, namely, cloud computingmanagementplatforms, Virtualization, High Performance Cloud (HPC), and performance testing.Finally, section 2.4 shows the software tools that have been used in this research.

## 2.2. Literature Review

This section contains an investigation on the existing work that is relevant to this research, and it represents a context to our research.

### Cloud Computing Management Platforms

Several cloud managers are available in the field of cloud computing Peng et al.comparedthese cloud managers which were Eucalyptus, Nimbus, Abicloud, and OpenNebula from different point of view. The main of their study was to focus on cloud management system characteristics, applications, and deployment requirements. The conducted comparison was in scalability, compatibility with applications, supporting virtual machines, web interface, development languages support, and operating systems support. They summarized their findings to provide information to users in order to make deep understanding of features provided by these systems(Peng, J. et al, 2009).

Cloud managers should work with specific types of software and hardware. Sempolinski and Thainmade comparison between major cloud computing managers which was Eucalyptus, OpenNebula and Nimbus. The conducted comparison focused on the raw features of each one of them. They defined the relationship between these management systems and software that is required to make cloud computing system working. They summarized some different features for each cloud management system(Sempolinski, P. and Thain, D., 2010).

Collaborative Commerce (c-commerce) is one of e-business forms, that is responsible for servicing customers and collaborating business partners. Thus, cloud computing paradigm is represented as a suitable environment for c-commerce. Upgrading c-commerce with regards to IT infrastructure to run cloud computing have a number of challenges. Al-Bahadili, H. et al. described and evaluated the performance by applying new model of c-commerce. Hence, the model contained six primary components which were client, provider, auditor, broker, security, and privacy. They evaluated performance of new cc-commerce model by measuring the response time. Consequently, the results showed that cc-commerce achieved positive performance than equivalent c-commerce model (Al-Bahadili, H.et al., 2013).

The architecture of cloud managers is responsible for defining the way that it interacts with software applications running in the cloud to meet a suitable performance. Nagarand Sumanmade comparison between Eucalyptus, OpenNebula, Nimbus and OpenStack in term of performance. They provided a comparative study that helps users to select the best cloud manager in term of deployment strategy for users. They find out that scheduling algorithm in each cloud manager had a clear effect on cloud manager performance. They find out that OpenNebula is suitable for small and private companies, and it provides greatest level of centralization and customization(Nagar, N. and Suman, U., 2014).

## Virtualization

Virtualization follows two types which are full virtualization and paravirtualization. Binuand Kumarstudied hypervisors such as KVM, Xen and its effects on network performance in cloud environments. They compared full virtualizations approach with paravirtualization approach in term of CPU scheduling and memory management. They use scheduling algorithms of both KVM and Xen from latency dispatch point of view, in order to measure boot time for guest operating systems. In term of memory management, they proposed a technique to detect which hypervisor give its guest operating systems the permission to own application hardware page frame by using shared translation array(Binu, A. and Kumar, G. 2011).

Hypervisor performance affected due to many parameters. Scholosser et al.proposed a novel study to find how isolation techniques have impacts on the performance of guest systems. They studied how hypervisors used in cloud computing such as KVM, Xen, and VirtualBox may affect network throughput. In more details, they worked on defining the size of packets in the network and measuring virtual machines CPU and memory utilizations, which will reflect the performance of virtual machines in the network(Scholosser, D. et al. 2011).

Kolheand Dhagemade a comparative analysis of KVM and Xen depending on various benchmarking tools. They concentrated on measuring CPU performance, network speed, and disk access using a secure shell connection (SSH), and applying benchmark tools for finding results(Kolhe, S. and Dhage, S. 2012).

Hwang et al.performed performance comparison under hardware-assisted virtualization settings. They considered four virtualization platforms Hyper-V, KVM, vSphere, and Xen. The comparison depended on evaluating each resource component with specific benchmark workload. The components used were CPU, memory, Disk I/O, and network I/O. They usedBYTEmark benchmark application to stress the capability of CPU, in order to extract CPU performance, Ramspeed benchmark tool to measure cash and memory bandwidth, Bonnie++ benchmark application to measure disk throughput, and Netprefbenchmarktool to measure various aspects of network features by measuring request/response performance using TCP or UDP. They find out that there is no perfect hypervisor and to have efficient cloud environment by building heterogeneous datacenter and cloud support variety virtualization platforms(Hwang, J. et al. 2013).

The core component in search engine is the web crawler. Hence, crawling data from rapidly and changeable environment (i.e. internet) demands a large hardware resources. Al-Bahadili, H. et al. developed a new approach to speed up the performance of crawling processes. Thus, they divided the multi-core processor into a number of VMs, which can run concurrently. They extracted the average crawling rate in documents per time unit. Xen hypervisor was used as a virtualization platform in their experiment. Their findings showed that the number of VM have a positive effect in speeding up the performance of crawling documents (Al-Bahadili, H. et al.,2013).

## High Performance Cloud (HPC)

Hoffa et al.studied the use of cloud computing for scientific workflows. Their study focused in evaluating the performance of compute service for scientific application which is often requires high performance specification in traditional and local systems. They used Xen as a hypervisor for creating virtual machines. They used client cloud provided in the University of Chicago which called Sixteen-Node Tera Port, they tried to evaluate CPU, disk storage, and memory utilization as benchmark for their study. They find out that cloud instance got good results which executing scientific applications, but it faced a critical in disk storage management because of the fixed size used to create volume which attaching instances in the creating process(Hoffa, C. et al. 2009).

Youngeet al.made an analysis for some virtualization technologies by comparing them from performance point of view. They focused on the impact of hypervisor type on cloud environments. They build high performance cloud computing environment in order to test its applicability with High Performance Computing (HPC) applications. They made detailed comparison between Xen, KVM, Virtual BOX, and VMware hypervisors. In order to compare performance for such environments, they proposed to use two standard performance benchmark suits HPCC and SPECC. HPCC performance standard benchmark contains HPL LinkPack TPP benchmark to measure floating point rate of the execution of solving linear system equations, DGEM benchmark to measure floating point rate of executing real matrix multiplication, and STREAM benchmark to measure the bandwidth of memory in (GB/S). SPECC performance standard benchmark contains SPEC CPU2000 which was used to measure CPU utilization of the virtual machines. They find outa proof that KVM hypervisor is the best choice for HPC cloud environments(Younge, A. et al. 2011).

## Performance Testing

Ostermannet al.proposed evaluation technique for the performance of scientific application in cloud environment. They analyze the performance of the scientific applications in Amazon EC2 cloud environment. They used micro-benchmarking tools. The proposed experiments which they tried to apply covered various types of instances in the perspective of instance specifications provided by Amazon EC2 cloud. Their study concentrated on evaluating the total time for service creating, deployment, boot, and release(Ostermann, S. et al. 2010).

Saini et al.studied the performance and scalability of engineering applications of interest of NASA on NASA cloud computing platform called Nebula. They proposed a comparison between Nebula cloud platform applications using NUTTCP function which they used to measure network throughput between two peers in the cloud, by measuring TCP/UDP network layer throughput of transferring memory buffers between the hosts. They used HPCC function to measure processor performance in term of CPU and memory utilization(Saini S. et al. 2011).

Yang et al.proposed a way to build KVM environment in the cloud systems and operation. This study focused on building environment with respect to reduce the complexity of cloud resources access. They proposed an experiment to measure the performance of physical machine in order to calculate machine built time, start time, and computing performance. They used CPU utilization, disk usage, and memory utilization(Yang, C. et al. 2011).

Steimetzet al.studied the performance of cloud computing platform in the perspective of Information Technology (IT) management. They applied two types of tests on cloud environments which are OpenStack, and Eucalyputs. They established each environment on identical hardware. They used BYTE UNIX benchmark suit to conduct various types of performance tests on both environments. Their research concentrated on testing the launch time for virtual machine instances in both clouds. They start the experiments by launching VM in parallel using command to launch several images at one time and extracting start time for each environment(Steimetz, D. et al. 2012).

Bahgaand Madisettiproposed a methodology for performance testing of complex multitier applications. They tried to capture the work-loads of multitier cloud applications using benchmark applications. They proposed a rapid deployment prototyping methodology in order to choose the best and most cost effective deployment for multitier applications that meet specified performance requirement. The proposed benchmark model was included with attributes such as operations, workload mix, inter-request dependencies and data dependencies(Bahga, A. and Madisetti, V., 2013).

Govindand Mamathafind out a technique to evaluate the CPU usage statistics provided by KVM hypervisor for running VM. The technique was used in their research to validate the reliability and accuracy of CPU statistics. They find out after ran intensive applications in the user mode inside VM, that CPU statistics was increased. The basic idea in the experiment was by testing the CPU statistics for one VM, then retest using a specific number of VMs and compares the results(Govind, R. and Mamatha, T., 2013).

## 2.3. Software Tools Used in the Research

Many tools have been used to reach and extract some necessary results. This section shows a brief description for each one of them.

### 1. Xen Hypervisor

Xen hypervisor platform is an open source hypervisors that is used to employ virtualization technology in cloud computing environments. It contains two core components the Xen hypervisor and scheduling of virtual machines component. In Xen hypervisor, which is responsible for CPU, memory, and power management. This platform provide a privilege virtual machine called domain 0 (dom0) for managing driver of devices and allow access to hardware. The guest operating system should be modified and it is located at domain U (domU). This hypervisor categorized as "type 1" hypervisor Figure 2.1 shows illustration of Xen architecture.

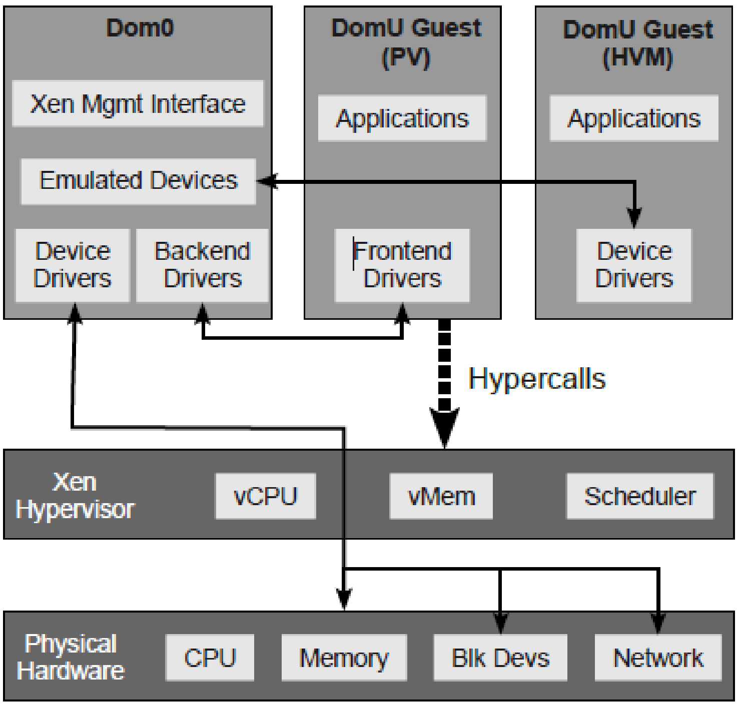


**Figure 2.1**Xen hypervisor architecture model **(**Macko, P. et al., 2011**)**

## 2. KVM hypervisor

KVM hypervisor platform represent another example of leading hypervisors that employs virtualization technology in cloud computing. This hypervisor reconstruct Linux kernel to run virtual machine as a regular process in Linux operating system, which will benefit from all features of Linux kernel. It emulates devices using Quick Emulator (QEMU) to provide emulated BIOS, PCI bus, USB bus …etc (Linux Redhat Team, 2009) Figure 2.2 shows an illustration of KVM architecture.

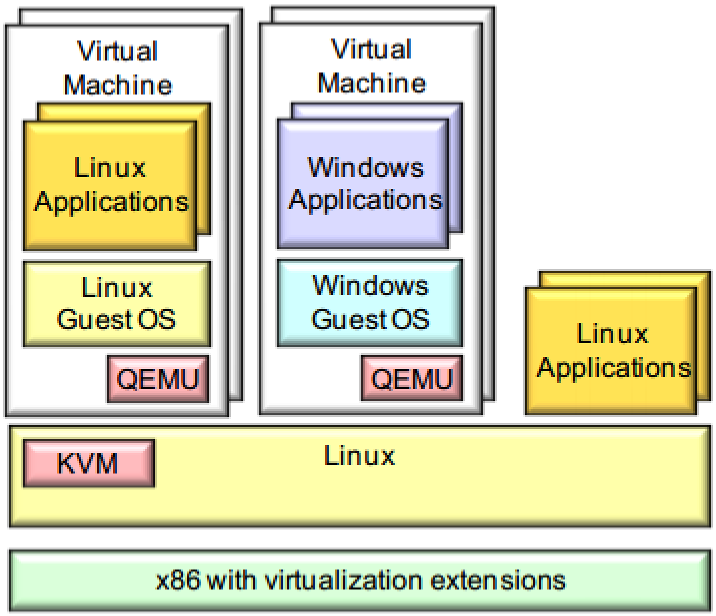


**Figure 2.2** KVM hypervisor architecture model(Chen, G. and Gillen, A., 2011)

In this research we will study hypervisors (e.g. KVM, and Xen) effects on virtual machines running in the cloud, by comparing performance of these virtual machines. We will evaluate performance depending on the utilization of computing resources: CPU utilization and memory utilization.

## 3. Ubuntu Server 12.04 LTS

Ubuntu Server 12.04 LTS provided by Canonical Company as server operating system in 2012. It was enhanced to meet newest technologies in the field of computer science. We choose Ubuntu Server 12.04 LTS in this research because of its capabilities in supporting cloud computing and virtualization technologies such as (Canonical Team. 2012):

- It's ability to support Xen and LXC. Thus, it is able to provide facility to run Ubuntu as a Xen virtualization host (dom0), and the ability to support Linux Container (LXC) in order to allow sharing of kernel resources in case of multiple operating systems.
- It's ability to support KVM and Libvirt. Thus, it is able to support CPU bandwidth limits, the ability to make tracing and debugging, and the ability to support AMD and INTELL processors.

## 4. Cloud Computing Management System – OpenNebula

OpenNebula is an open source platform in cloud computing, Offers highly scalable environment. It found in University of Madrid in 2005 as a research project, and it was released in 2008. It support Apache licensed server and virtualization technology. It provides an abstraction layer that is independent from the underlying services such as security, virtualization, networking, and storage.

OpenNebula composed of six core components, *Request Manager* which is used to pop client requests and manage them, *Virtual Machine Manager* used to control and monitor the running virtual machines, *Transfer Manager* used to manage the operating system of each virtual machine, *Virtual Network Manager* used to administer the network, *Host Manager* that is used to manage the physical resources, and *database* that is used as a storage for single data structure. Figure 2.3 shows these components.

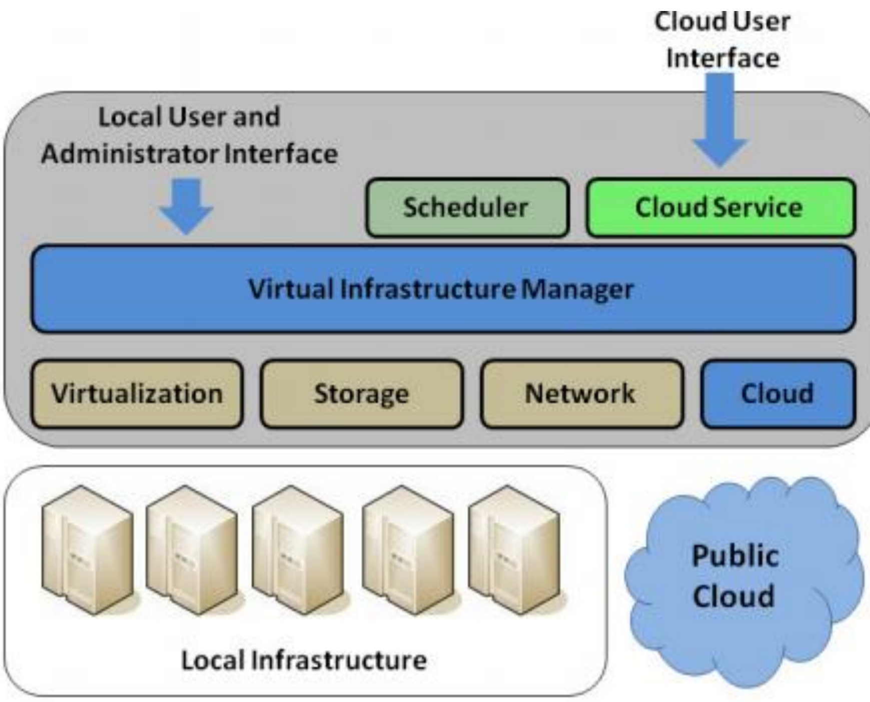


**Figure 2.3**OpenNebula Components**(**Endo, P. et al., 2010**)**

## 5. Cloud Computing Management System – OpenStack

OpenStack is cloud operating system used in public and private clouds, provided by Rackspace® Foundation. Provide free Apache-licensed server software in order to build scalable cloud environment. Three primary components found in OpenStack: OpenStack COMPUTE (called nova), OpenStack IMAGE (called glance), and OpenStack OBJECT STORAGE (called swift) (OpenStack Documentation, 2011). Figure 2.4 shows the architecture of OpenStack cloud manager.

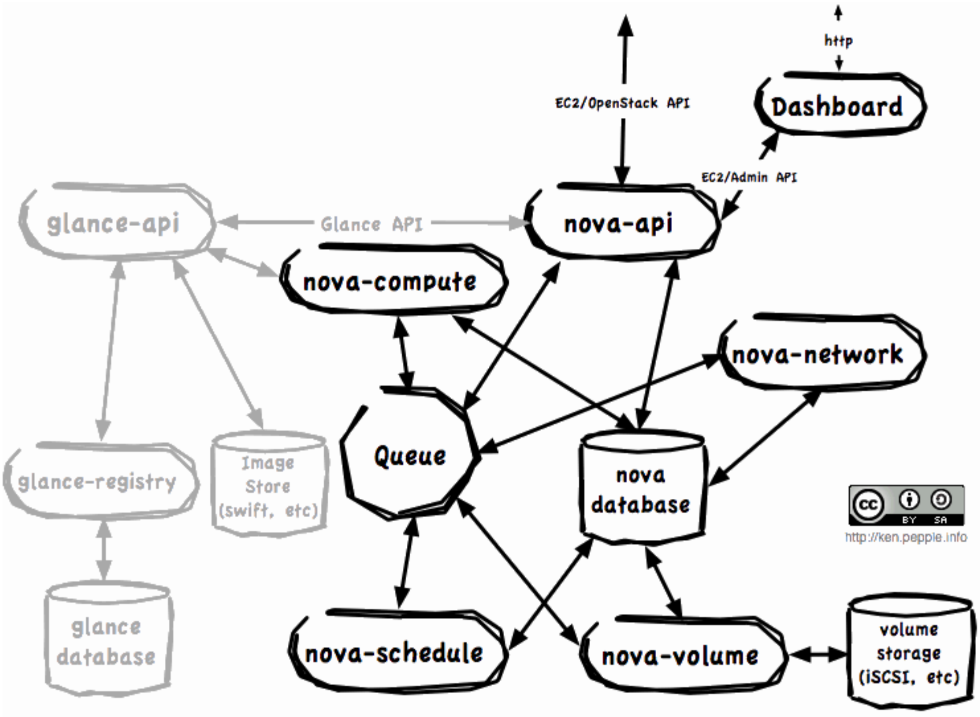


**Figure 2.4**OpenStack architecture model **(**Pepple, K., 2011**)**

Some of services provided by OpenStack summarized in Table 2.1.

| **Service** | **Code Name** | **Details** |
|---|---|---|
| Dashboard | Horizon | The interface that guarantee the interaction between users and OpenStack cloud. |
| Compute | Nova | Managing and controlling virtual machines that are running in cloud. |
| Block Storage | Cinder | Service used to provide disk space for each guest virtual machine. |
| Identity Service | Keystone | Manage authentication and authorization of cloud services. |
| Image service | Glance | Defines each instance and its image (i.e. Operating System Image). |

**Table 2.1**Some of OpenStack provided services in cloud environment.

Choosing OpenNebula and OpenStack in this research was for the most important features found in both of them such as support Apache server version 2.0 which works with Ubuntu Linux Server version 12.04 operating systems which will be our host operating system, support to all types of hypervisors such as KVM and Xen, and support to different operating system images format such as ISO, QCOW2, and RAW. In addition, both of them capable to provide Graphical User Interface (GUI) that are used to manage and control virtual machines running in the cloud using Sunstone in case of OpenNebula, and Dashboard in case of OpenStack.

## 6. Microsoft SQL Server 2000

Microsoft – SQL Server 2000 is a Relational Database Management System (RDBMS) used to provide various types of tools in order to ease the development of database and maintenance implementation. It contains six core components *Enterprise Manager, Query Analyzer*, *SQL Profiler, Service Manager,* and *Data Transfer Service (DTS)* (Chapple, M. 2000)

In our research we will focus on using *Query Analyzer* which is used to perform queries over databases. Selecting this component based on the services which it provides such as it is ability to measure the response time to user requests, the ability of testing queries, and executing administrative tasks. The SQL queries that were executed in the experiments were divided into three categories DDL, DML, and DQL categories. Each category contained number of SQL queries in order to measure the response time of hypervisor in term of CPU and memory.

## 2.4. Hardware Used in This Research

In this research the measurement of performance shown for CPU and memory response time and discarding network bandwidth. Hardware will be used in the experiments is one SAMSUNG NP300E5V laptop with the following specifications:

- (BIOS): Phoenix BIOS SC-T v2.2 P02RBD.

- (Processor): Intel® Core ™ i3-3120M CPU @ 2.50GHz.

- (Memory): 4096 MB RAM.

In this research the VM was created in cloud using two cloud managers. In both managers the VM had the following specifications:

- Virtual Central Processing Units (VCPU) : 1.
- Random Access Memory (RAM) : 1 GB.
- Processor: 256 Ghz.
- Storage Volume: Not assigned in both managers.

# CHAPTER THREE

# The Proposed Model and Experiments Design

## 3.1. Overview

In this chapter, we present a detailed description of the proposed model, as well as, discuss the proposed experiment's design, and finally define the Key Performance Indicators (KPI's) that will be evaluated in cloud through the research.

## 3.2. The Proposed Model Architecture

The main theme of this research is to find the suitable hypervisor type that suites query types with regards to resource utilization. To achieve this goal, several types of cloud managers were used to manage and design clouds. As well as, different types of virtualization hypervisors were found in the area of cloud computing. Hence, cloud managers and hypervisors had different ways in executing SQL queries. This modelwas proposed in order to find the suitable cloud manager and hypervisor in a way that meets the best CPU and memory response time after executing different types of queries. The proposed model was consisted tothree levels which are cloud manager level, virtualization level, and VM level.

### Cloud Management Platform Level

In this level, we will discuss the cloud management platform, which is responsible for building cloud and prepare its services. The cloud services in this level are compute service which let cloud's customers to use virtual machines. Cloud management platforms provide the following facilities to manage cloud:

- Create VM's and setting up the processing features such as CPU, memory, and disk volume.
- Select the type of hypervisor used in the cloud.
- Setting up the general storage volume for cloud.
- Setting up and uploading VM's operating system images.
- Provide Hypervisor summary in term of virtual CPU, memory, and disk usage.

## OpenStack Cloud Manager Architecture

In this section, OpenStack architecture will be discussed. Figure 3.1 shows the main components of OpenStack cloud manager and its interaction with each other. OpenStack represented as cloud OS that is responsible to do multiple functions. OpenStack components designed to provide services and to work together, in order to provide full cloud infrastructure(1).

OpenStack contains Dashboard component that is responsible to provide a web gate for end users to interact with cloud facilities and it called *Horizon* service. Compute component responsible to provide saving and retrieving operations for disk images and called *Nova* service. Network component that is called *Quantum service* is responsible to provide virtual networking between cloud services. Block storage component that is responsible to provide disk volumes for compute component and it called *Keystone* service (2).

(1) Retrieved from http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html , Date Accessed: 20 Feb. 2014.

(2) Retrieved from http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html , Date Accessed: 1 Mar. 2014

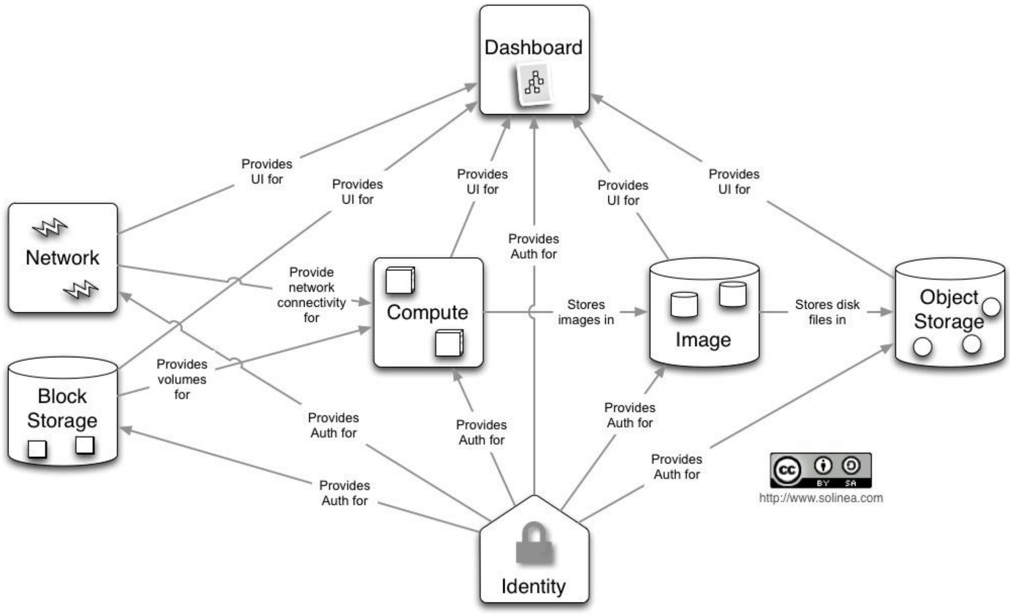


**Figure 3.1**OpenStack cloud manager architecture**(**Pepple, K., 2011**)**.

## OpenNebula Cloud Manager Architecture

In this section, OpenNebula cloud manager architecture will be discussed. OpenNebula cloud manager consists into three layers tools layer, drivers layer, and core layer. Tools layer provide Command Line Interface (CLI) service that is responsible to allow users to manipulate with cloud virtual infrastructure, and Scheduler that is responsible to invoke actions on VMs to allow definitions to several resources. Figure 3.2 shows OpenNebula tools layer components.

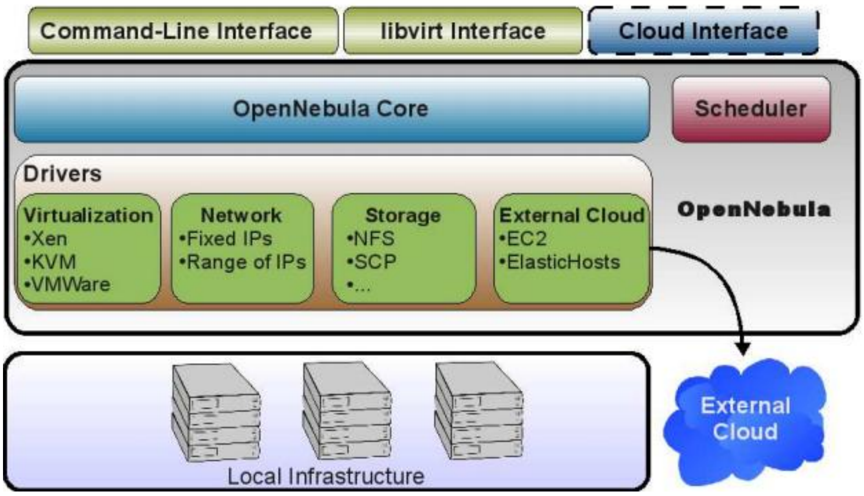


**Figure 3.2**OpenNebula tools layer architecture (Sotomayer, B. et al., 2009).

Drivers layer contain many modules that is used to interact with middleware such as Virtualization hypervisors, cloud services, and transfer operations. All of these drivers are pluggable with OpenNebula cloud. Figure 3.3 shows OpenNebula Drivers layer architecture.

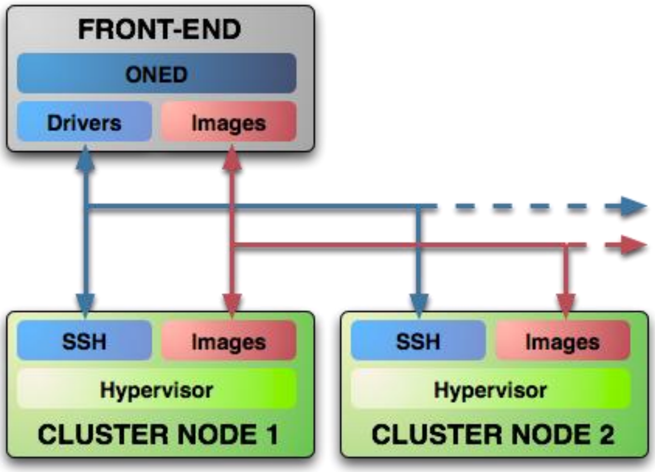


**Figure 3.3**OpenNebula drivers layer architecture (Blanco, C., and Sotomayer, B., 2010).

OpenNebula core layer contains the most important services such as request manager service that is responsible to handle client's requests, VM manager that is responsible to monitor VMs in the cloud, transfer manager that is used to manage VM images and its meta data, virtual network manager that is used to manage cloud internal networking operations, and host manager that is responsible to monitor and manage the physical resources.

## Virtualization Level

Cloud performance is affected by the type of hypervisor that has been selected. Virtualization in cloud represented in the hypervisor; which is responsible for running VM's in the cloud. Hypervisors follows different architectures which may lead for differences in the performance. Performance parameters that will be evaluated through this level are CPU, and Memory usage.

## VM's in Cloud Level

VM's is the gate between clients and cloud where they can manipulate and work on computational units. VMperformance depends on the type of hypervisor that is used in the cloud. Hypervisor technology control VMs from different points of view such as response time, throughput, and resources utilization. Database in this level will be used as a tool to predict hypervisor performance; by evaluating the performance of VMs that is varied upon the type of hypervisor.

## 3.3. The Evaluation Criteria

In this research, CPU and memory response time were extracted from two main levels; the performance evaluation of hypervisor level, and cloud manager level.The test experiment appliedon Northwind database.SQL statements which that was used to evaluate the performance of hypervisor follow three basic categories:

1. Data Definition Language (DDL) statements.

DDL statements allow users to create, alter, and omit objects in database such as tables, indexes, views, and sequences. Table 3.1 shows SQL queries in DDL category.

| No. | SQL Query | Description |
|---|---|---|
| 1 | `CREATE TABLE TestDB3 { testIDint,expNamevarchar(255),expLocationvarchar(255),expLabvarchar(255),Team varchar (255) }` | Create table in static way. |
| 2 | `SELECT Customers.CustomerID,Customers.ContactName, Customers.ContactTitle,Orders.OrderID,Orders.OrderDate INTO TestTable FROM Customers FULL OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID` | Create table using Full JOIN |
| 3 | `SELECT * INTO Customers3 FROM Customers` | Create table using SELECT |
| 4 | `CREATE TABLE TestDocNo5 seqID Int identity(1,100) PRIMARY KEY, seqVal AS 1+(seqID-1)%9)` | Create sequence |

**Table 3.1** DDL statements used in the evaluation process

2. Data Manipulation Language (DML) statements.

DML statements allow users to manipulate table's data such as insert new data to table, changing data types for specific column in tables, and removing rows from tables. Table 3.2 shows the SQL statements ofDML category.

| No. | SQL Query | Description |
|---|---|---|
| 5 | `INSERT INTO CustomerINFO4 (CUstID,CustName,CustAddress,CustCountry) VALUE ('123461','Jasour Ahmed Obeidat' ,'Irbid' ,'JORDAN')` | Insert row using INSERT INTO + VALUES |
| 6 | `INSERT INTO CustomerINFO1 (CustID, CustName, CustAddress, CustCountry) SELECT CustomerID, ContactName, Address, Country FROM Customers` | Insert new row using SELECT |
| 7 | `DELETE FROM CustomerINFO1 WHERE CustID>'2000'` | Create table using SELECT |
| 8 | `UPDATE Customers SET ContactName ='Ahmed Obeidat' , City='Amman' WHERE CustomerID='VINET'` | Static Update |
| 9 | `SELECT ContactName, ContactTitle FROM Customers WHERE CustomerID = ( SELECT CustomerID FROM Orders WHERE OrderID = ( SELECT OrderID FROM Orders WHERE EmployeeID=8 AND ShipName='Hanari Carnes'))` | Using Multiple Sub-Queries |

**Table 3.2**DML statements used in the evaluation process

3. Data Query Language (DQL) Statements

DQL allow users to retrieve data using SELECT statement, and joining tables using multiple types of SQL join. Full outer join, left outer join, right outer join, natural join, inner join, and inner join with a condition. Table 3.3 shows the SQL statements of DQL category.

| No. | SQL Query | Description |
|---|---|---|
| 10 | SELECT * FROM Customers | Retrieve using static SELECT. |
| 11 | SELECT CustomerID,ContactName, Address<br>FROM Customers<br>WHERE IN ('France','USA','United Kingdom') | Retrieve using SELECT+IN |
| 12 | SELECT DISTINICT ContactTitle<br>FROM Customers | Retrieve using SELECT + Distinct |
| 13 | SELECT CustomerID,EmployeeID FROM Orders<br>WHERE EXISTS (SELECT * FROM Orders WHERE Freight>14.5) | Retrieve using SELECT+EXISTS |
| 14 | SELECT<br>Customers.ContactName,Customers.ContactTitle,<br>Orders.OrderDate, Orders.ShipName<br>FROM Customers LEFT OUTER JOIN Orders<br>ON Customers.City = Orders.ShipCIty<br>ORDER BY Customers.CustomerID | Join tables using LEFT OUTER JOIN |
| 15 | SELECT<br>Customers.ContactName,Customers.ContactTitle,Orders.OrderDate,Orders.ShipName<br>FROM Customers RIGHT OUTER JOIN Orders ON<br>Customer.City =Orders.ShipCity<br>ORDER BY Customers.CustomerID | Join tables using RIGHT OUTER JOIN |
| 16 | SELECT Customers.ContactName,<br>Customers.ContactTitle,<br>Orders.OrderDate, Orders.ShipName<br>FROM Customers FULL OUTER JOIN Orders<br>ON Customers.City = Orders.ShipCIty<br>ORDER BY Customers.CustomerID | Join tables using FULL OUTER JOIN |
| 17 | SELECT Customers.CustomerID,<br>Customers.ContactName,Orders.*<br>FROM Customers INNER JOIN Orders<br>ON Customers.City = Orders.ShipCity | JOIN tables using NATURAL JOIN |
| 18 | SELECT *<br>FROM Customers AS a INNER JOIN Orders As p<br>ON a.City = p.City | Join tables using Equi-JOIN |
| 19 | SELECT Customers.ContactName, CUstomers.COntactTitle, Customers.Country, Orders.ShipCity FROM Customers CROSS JOIN Orders ORDER BY Customers.CustomerID DESC | Join tables using CROSS JOIN |
| 20 | SELECT Customers.ContactName,Customers.City,<br>Orders.OrderDate,Orders.ShipName,Orders.ShipCountry<br>FROM Customers INNER JOIN Orders ON Customers.Country<>Orders.ShipCountry<br>ORDER BY Customers.CustomerID | Join tables using NOT Equi-JOIN |

**Table 3.3**DQL statements used in the evaluation process

SQL statements were used as a tool to evaluate the performance of hypervisor in this level as shown in figure 3.4.

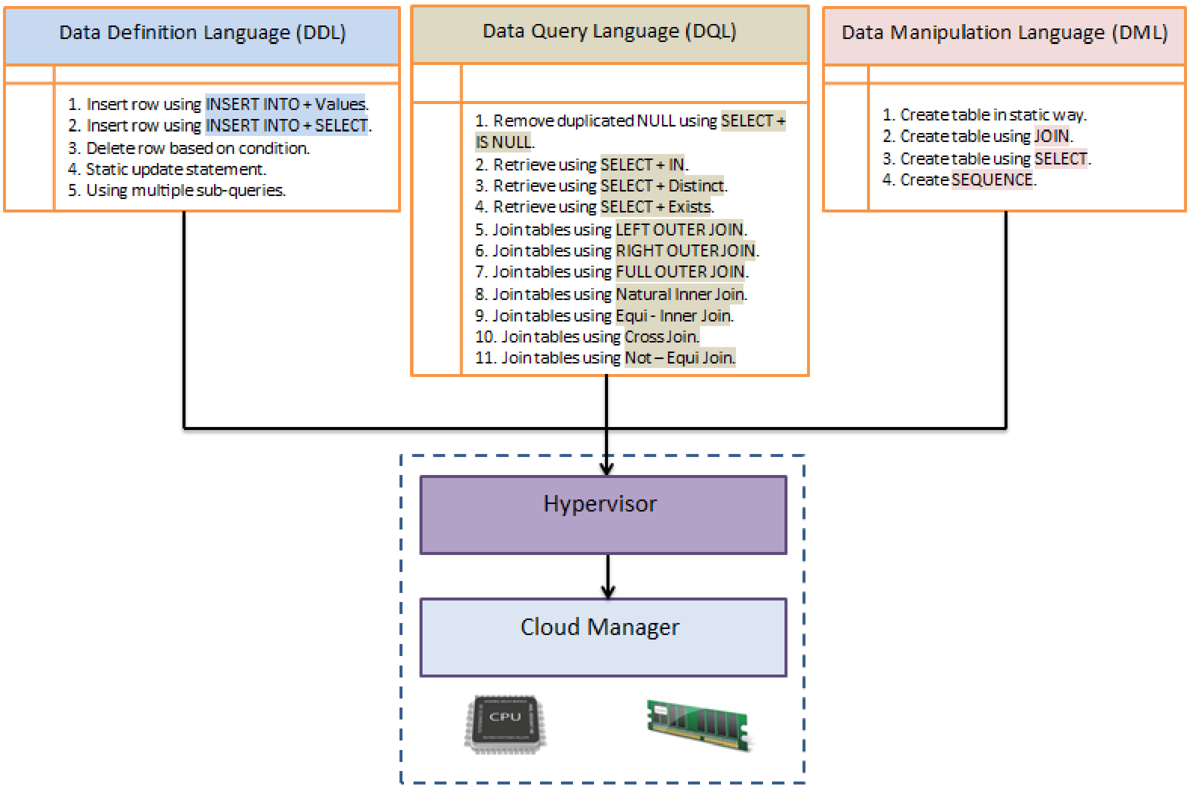


**Figure 3.4**General SQL Categories Used for Performance Evaluation

For each SQL statement in the evaluation process we added the *SET STATISTICS Time*statement to measure the performance. Table 3.4 shows the basic output for *SET STATISTICS Time* statement(1).

| Output | Description |
|---|---|
| Parse and Compile Time | The total time taken to parse, optimizes, and compile SQL query. |
| CPU Time | The total time taken by CPU to execute the query |
| Elapsed Time | The amount of time in CPU, RAM, Disk, Parse, Compile, and Monitor card to display the results. |

**Table 3.4**SET STATISITCS TIME Outputs

(1) Retrieved from http://technet.microsoft.com/en-us/library/aa259194%28v=sql.80%29.aspx , Data Accessed: 22 Mar. 2014

Memory utilization could be calculated depending on the response time of CPU, Parse & Compile time, and elapsed time. Figure 3.5 shows the computing formula.

$$\text{Memory Response Time} = \text{Elapsed Time} - (\text{CPU Response Time} + \text{Parse and Compile Time} + \text{Disk Response Time} + \text{Monitor Card Response Time})$$

**Figure 3.5**Memory response time formulas (Christoffer, H., 2010).

The total time consumed in the disk will be zero if the queries are executed three times, since it will be cached. In case of monitor card time it will not affect the comparison because of all virtual machines will use the same monitor card size. Therefore, formula used to calculate the memory utilization shown in Figure 3.6 (Donkena, K. and Gannamani, S., 2012).

$$\text{Memory Response Time} = \text{Elapsed Time} - (\text{CPU Response Time} + \text{Parse \& Compile Time})$$

**Figure 3.6**Memory response time formulas in performance evaluation (Christoffer, H., 2010).

## 3.4. The Design of Proposed Experiment

Multiple types of cloud managers were developed to let users to build their own clouds such as OpenStack and OpenNebula cloud managers. Different cloud manager architectures were proposed to provide clouds and entered the competition many areas such as performance. Cloud manager can interact with virtualization hypervisors (e.g. KVM or Xen) in different ways.  In this research, we executed two experiments to find the differences in performance for two clouds, and the criteria for the evaluation was depended on database application running in each cloud.

Database is one of the most important inventions in information technology, because of its functionalities that offer such as management, processing, and organizing information that in a structured and controlled manner. Database represented as the key for many aspects of modern business efficiency (Choo, C.W, 1995). Databases allow developers to find the performance in term of response time for any query that is running. In this research, we will use database as a tool to find the performance of a completed environment where the database are running.

Two test experiments proposed to extract the performance - with regards to CPU and Memory response times –and to find the differences between cloud managers and hypervisors after executing different query types. Figure 3.7 shows the experimental execution of the first experiment.

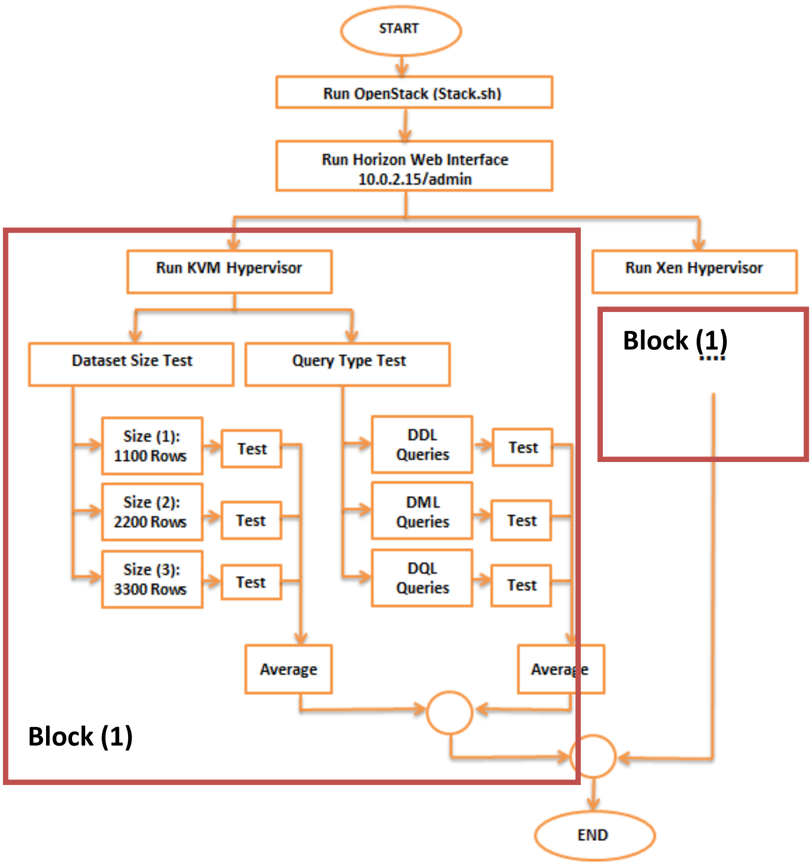


**Figure 3.7**First Experimental Execution

In the first experiment, we used OpenStack as a cloud manager, and we applied two types of hypervisors KVM and Xen. The second experiment OpenNebula was used as cloud manager with KVM and Xen hypervisors. Figure 3.8 shows the experimental execution of the second experiment, and the followed procedures to run it.

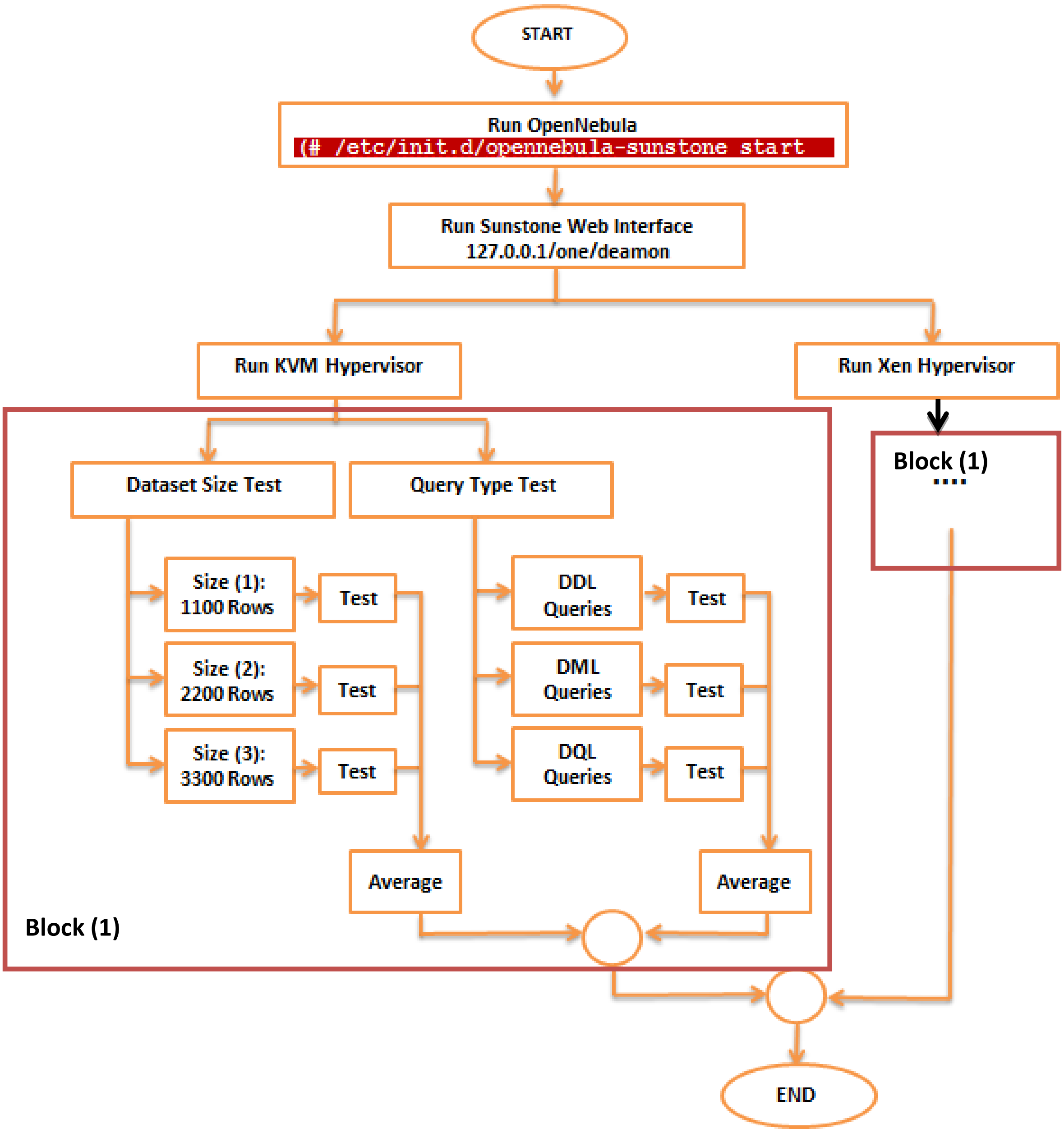


**Figure 3.8**The experimental execution flow chart for the second experiment.

In the second experiment, we used OpenStack as a cloud manager, and we applied two types of hypervisors KVM and Xen. In both experiments, we collected the performance in term of CPU and memory utilization for each query. The proposed comparative table was taken into consideration the types of cloud management platform that used – OpenStack vs. OpenNebula – and the type of hypervisor in each one of them – KVM vs. Xen – in case of performance evaluation.

The comparison built for two cloud environments that have the same specifications for the following components:

- The CPU specification for VMs offered by cloud management platforms.
- The size of memory and disk volume assigned to VMs.

The results will be used to compare the performance – with regard to CPU and memory response time – for hypervisors (e.g. KVM, Xen) used in the cloud, and cloud managers used to build the cloud (e.g. OpenStack, OpenNebula).

## 3.5. Case Study for Building OpenStack Cloud with KVM Hypervisor

We build cloud environment using OpenStack which is capable to provide compute service. We prepare all cloud services required to assign KVM hypervisor. The designed cloud will provide one VM that is able to run Northwind database through using SQL Server 2000. Performance will be evaluated using Query Analyzer component of SQL Server 2000.

## Creation of VM

The process of creation VM in cloud consists into three tasks:

- Uploading VM operating system image as shown in Figure 3.9.

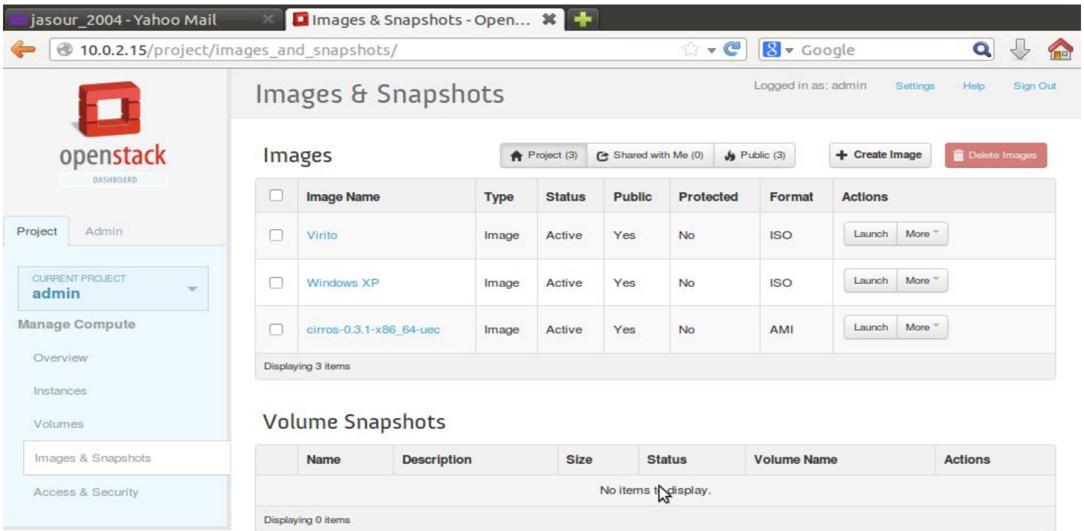


**Figure 3.9**Operating System Image Uploading into OpenStack

- Setting up VM specifications as shown in Figure 3.10.

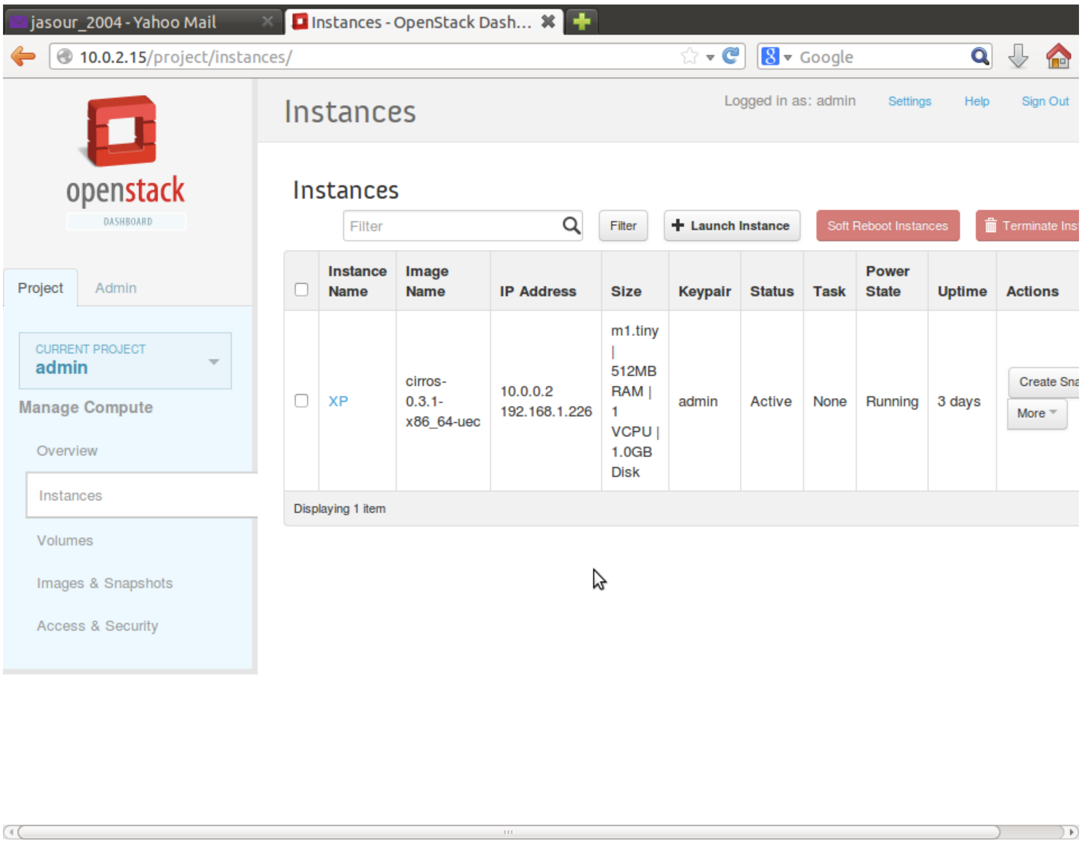


**Figure 3.10**Setting Up the specifications of VM running in the cloud.

- Assign the created VM to KVM as shown in Figure 3.11.

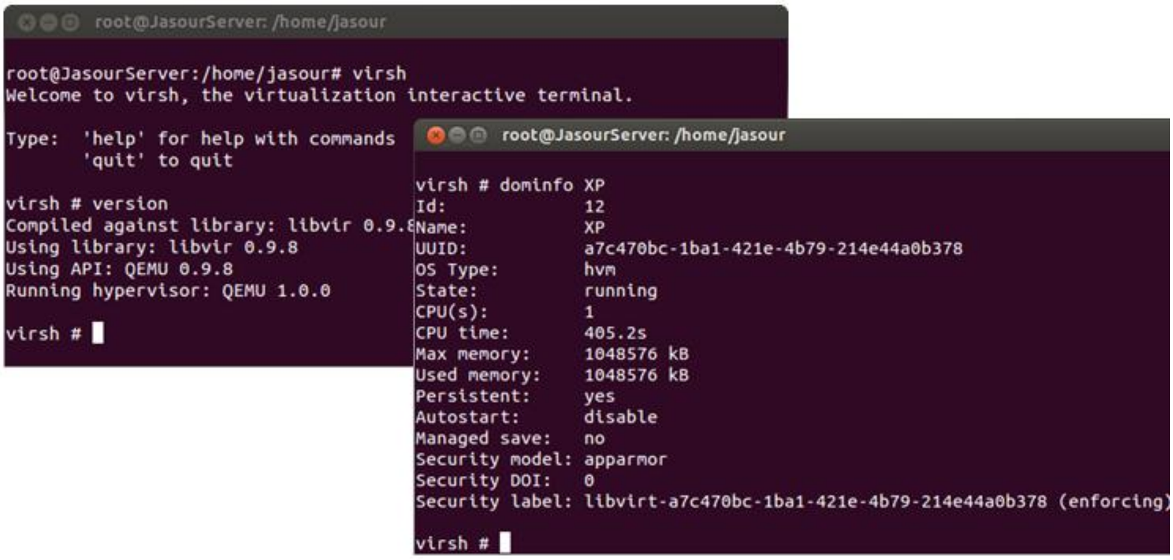


**Figure 3.11**Cloud VM running in KVM hypervisor.

## Running VMs in Cloud

OpenStack provide dashboard in order to control VMs that are running in the cloud as shown in Figure 3.12.

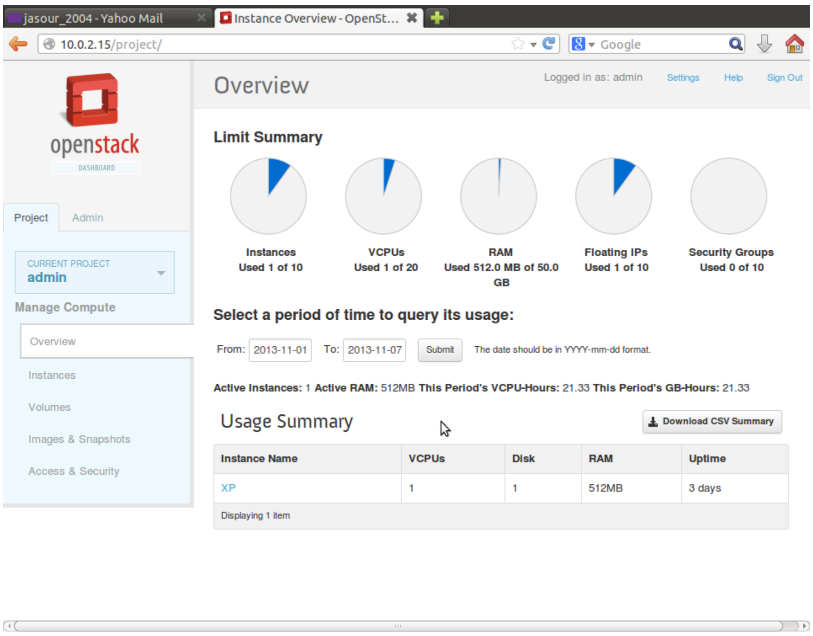


**Figure 3.12**OpenStack Dashboard VMs Overview Window.

KVM hypervisor monitor the performance - in term of CPU, memory, I/O, and network usage - of VMs which are running in the cloud as shown in Figure 3.13.

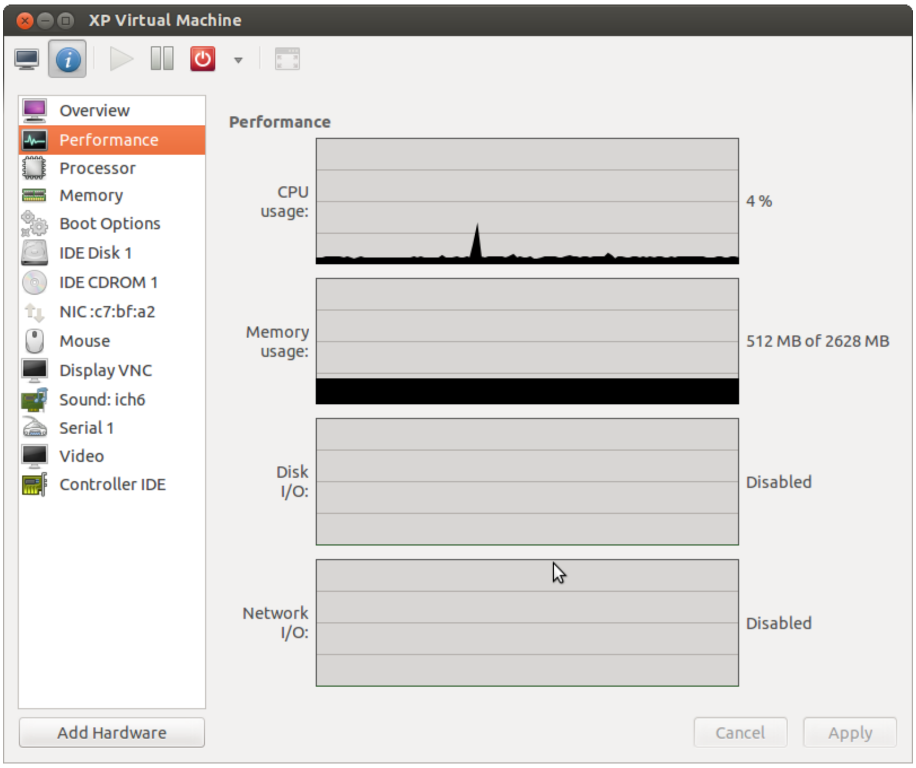


**Figure 3.13**KVM hypervisor monitoring VM's Performance.

## Northwind DB and SQL Server 2000

Northwind DB is a sample database which is included with MS-SQL Server 2000 and it represents the transactions occurred between company called Northwind Traders and its customers, as well as, the company and its suppliers (1).

(1) Retrieved from http://technet.microsoft.com/en-us/library/aa276825%28v=sql-.80%29.aspx ,

*Date Accessed: 21 Feb. 2014*

# Chapter Four

# Experimental Results

## 4.1. Overview

This chapter will discuss the results of the proposed experiment that is discussed in Chapter Three. In this chapter, performance of hypervisors and cloud managers will be discussed by analyzing and collecting the observations for each query in the experiment.

## 4.2. Experimental Results

In order to find the performance for virtualization hypervisors and cloud managers, database that is running on a VM was taken. The response time of CPU and memory was taken while executing queries over a database tables. The queries were categorized into DDL, DML, and DQL categories. The results that had been extracted represent the average of ten times of iteration. The queries were executed over a *Customers*, *Orders*, and *Employee* tables in Northwind database.

The flow of results in this chapter will be discussed as the following:

- Discuss the CPU and RAM response times of hypervisor and cloud manager in term of query category.

- Discuss the CPU and RAM response times of hypervisor and cloud manager in term of table data size.

## 4.3. Performance of Cloud Managers and Hypervisors over Different Query Types.

In this section, we discussed the effect of query type on CPU response time. The SQL queries have been divided into three categories. At first category, it is called DDL that is responsible for building new objects into database. At second category, it is called DML which is responsible for data manipulation operations, such as adding new row to table, or deleting one from it. At third category, it is called DQL which is responsible for retrieving data and joining tables.

### CPU Response Time of Cloud Managers and Hypervisors in DDL Category

From CPU point of view, the main aim behind testing this category; is to find the effect of creating new object in database on CPU response time. For convenience, the average of results in different table's sizes was taken for each query in the category. Table 4.1 shows the results of DDL category in term of CPU response time.

| Category | Data Definition Language | | | |
|---|---|---|---|---|
| Data | CPU Response Time | | | |
| Query | OpenStack | | OpenNebula | |
| | KVM | Xen | KVM | Xen |
| Query 1 | 48 | 30.3 | 49.3 | 31.3 |
| Query 2 | 1227.3 | 1196 | 1350 | 1247.3 |
| Query 3 | 950 | 861 | 1186.6 | 988.3 |
| Query 4 | 56.3 | 31.3 | 104.6 | 45 |
| Average | 570 | 529 | 672 | 578 |

**Table 4.1** Data definition language category results in term of CPU response time.

These results show drastic gaps in CPU response time after executing queries in this category. At query one and four, both cloud managers and their hypervisors was performing well. At query two and three, the CPU response time affected clearly. Figure 4.1 shows the chart of CPU response time results for cloud managers and hypervisors in DDL category.

From these results, we can extract that creating new table by using join operation has affected the performance of CPU such as in query 2. Another factor has direct effect in CPU response time was through copy table's data to new one such as in query 3.

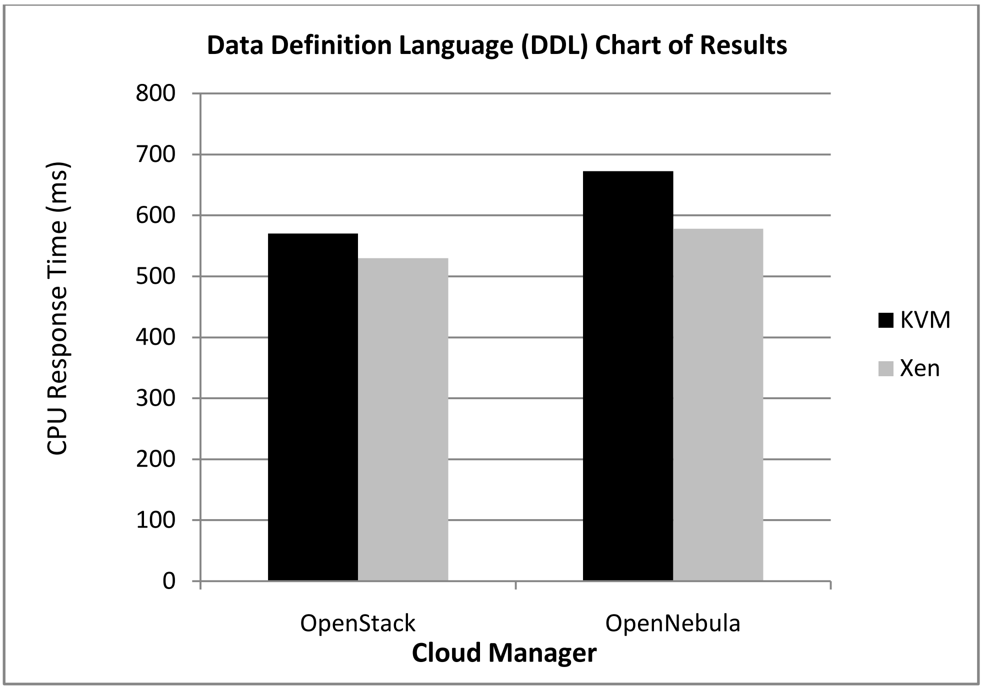


**Figure 4.1**CPU response time chart for OpenStack and OpenNebula with virtualization hypervisors in DDL queries.

From Figure 4.1, OpenStack cloud manager achieved better results than OpenNebula in term of CPU response time in DDL queries. OpenStack with Xen hypervisor achieved the best CPU response time in DDL queries. OpenNebula with KVM hypervisor achieved the highest CPU response time in DDL queries.

Coefficient of variance (CV)*was taken to make fair comparison between hypervisor types based on query type. Table 4.2 shows the results of computing standard deviation (S.D) and Coefficient of variance.

| Query | KVM | | | Xen | | |
|---|---|---|---|---|---|---|
| | Mean (?) | S.D (σ) | (CV) % | Mean (?) | S.D (σ) | (CV) % |
| 1 | 48.7 | 0.9 | 1.90% | 31 | 0.71 | 2.30% |
| 2 | 1289 | 87 | 6.70% | 1222 | 36.3 | 3% |
| 3 | 1068 | 167 | 16% | 1068 | 167.3 | 16% |
| 4 | 44 | 18 | 40% | 38 | 9.9 | 26% |

**Table 4.2** Coefficient of variation and standard deviation results for DDL queries in term of CPU response time.

The results show that KVM hypervisor achieved higher CV results than Xen hypervisor for a majority of DDL queries. The CV results in case of using KVM for DDL queries was 16%, while by using Xen hypervisor for the same SQL queries the CV was 11%. Thus, KVM hypervisor had a negative effect on CPU response time for DDL queries. On the other hand, Xen hypervisor had a positive effect on CPU response time based on DDL queries with regard to CPU response time. Figure 4.2 shows a chart of CV results for each SQL query in DDL category.

*CV is a statistical measurement; it represents the ration between standard deviation and the mean. It is used to find the percent for comparing two data sets, and it is more accurate than mean or S.D. Michael (Zeltkevic, M., 1998)

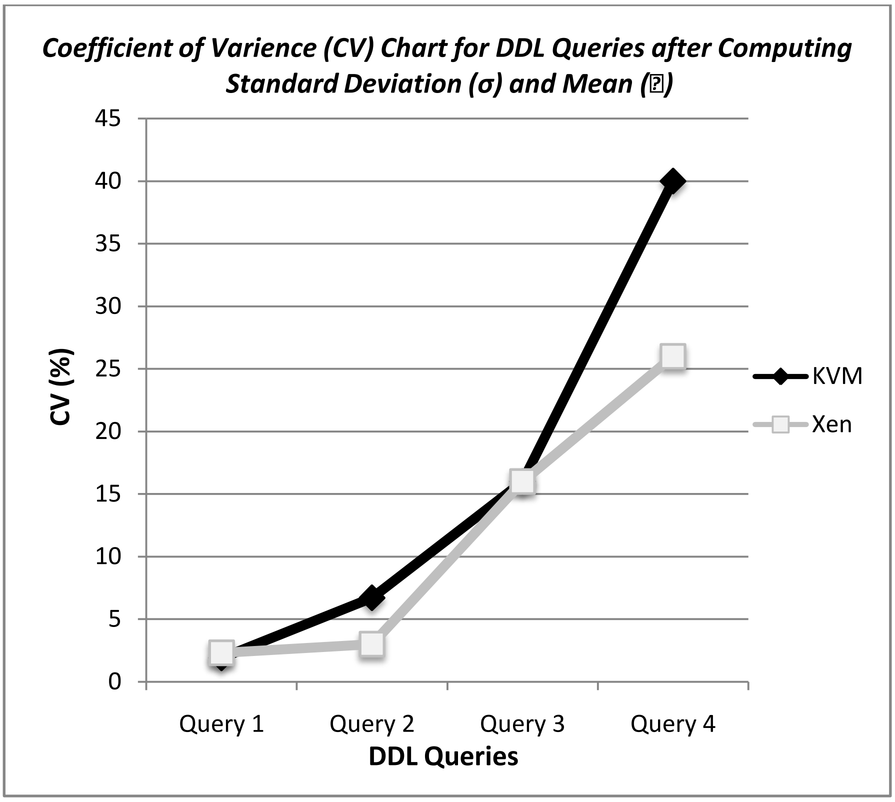


**Figure 4.2** CV, mean and S.D chart of results for DDL queries.

From memory point of view, the main aim of test is to find the effects of query types on both cloud manager and hypervisor. Evaluating the memory response time on DDL queries will be discussed in this section. Table 4.3 shows the calculated results of memory response time after executing DDL queries.

| **Category** | Data Definition Language (DDL) | | | |
|---|---|---|---|---|
| **Data** | Memory Response Time | | | |
| **Hypervisor** | **OpenStack** | | **OpenNebula** | |
| **Query #** | **KVM** | **Xen** | **KVM** | **Xen** |
| Query 1 | **156** | **184** | **250.3** | **214** |
| Query 2 | **2365.6** | **2741.6** | **2923** | **3074.3** |
| Query 3 | **5020.3** | **3522.3** | **4871** | **4708.3** |
| Query 4 | **153** | **174.3** | **131.3** | **156.6** |
| **Average** | **1924** | **1656** | **2044** | **2038** |

**Table 4.3** Data definition language category results in term of memory response time.

These results show that the memory had multiple behaviors for each query in DDL. Drastic gaps existed between the response times of memory for each query. Executing queries that was responsible for copying or creating new table from existed one achieved long memory response time. Simple DDL queries were performing well in term of memory response time. Figure 4.3 shows a graph that represents the averages of results at each stage in the cloud for DDL category.

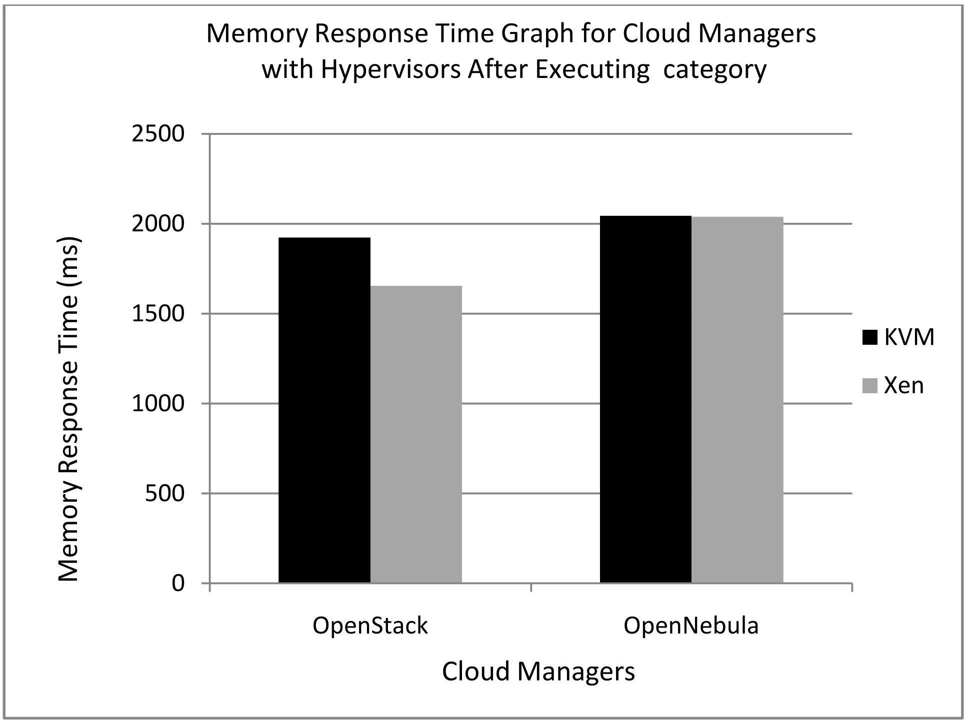


**Figure 4.3** Memory response time chart for OpenStack and OpenNebula with virtualization hypervisors in DDL queries.

From Figure 4.3, OpenStack cloud manager was performing well after executing DDL queries. OpenNebula cloud manager with both types of hypervisors achieved long memory response time. Xen hypervisor with OpenStack achieved the shortest memory response time compared with others. A drastic gap found between KVM and Xen while running in OpenStack cloud manager.

In addition, we applied the CV and S.D to compare the behavior of KVM and Xen hypervisors for each query in DDL queries with regard to memory response time. Table 4.4 shows the results were computed based on CV and S.D.

| Statistical Measurements for Memory Response Time Data after executing DDL Queries | | | | | | |
|---|---|---|---|---|---|---|
| Query | KVM | | | Xen | | |
| | Mean (?) | S.D (σ) | (CV) % | Mean (?) | S.D (σ) | (CV) % |
| 1 | 203 | 67 | 32 | 199 | 21 | 11 |
| 2 | 2644 | 394 | 15 | 2908 | 235 | 8 |
| 3 | 4956 | 106 | 2 | 4115 | 838 | 20 |
| 4 | 142 | 15 | 11 | 165 | 13 | 8 |

**Table 4.4** The computed CV, mean, and S.D results in term of memory response time for DDL queries.

The results of CV for Xen hypervisor exceeded the CV results of KVM. Thus, Xen hypervisor have a positive effect on DDL queries with regard to memory response time. However, KVM hypervisor faced a negative effect one DDL queries in term of memory response time compared with the CV in case of using Xen hypervisor. KVM hypervisor registered 15% as a CV, while Xen hypervisor achieved 12% as CV. Figure 4.4 shows the CV results of both types of hypervisors for each SQL query in DDL category.

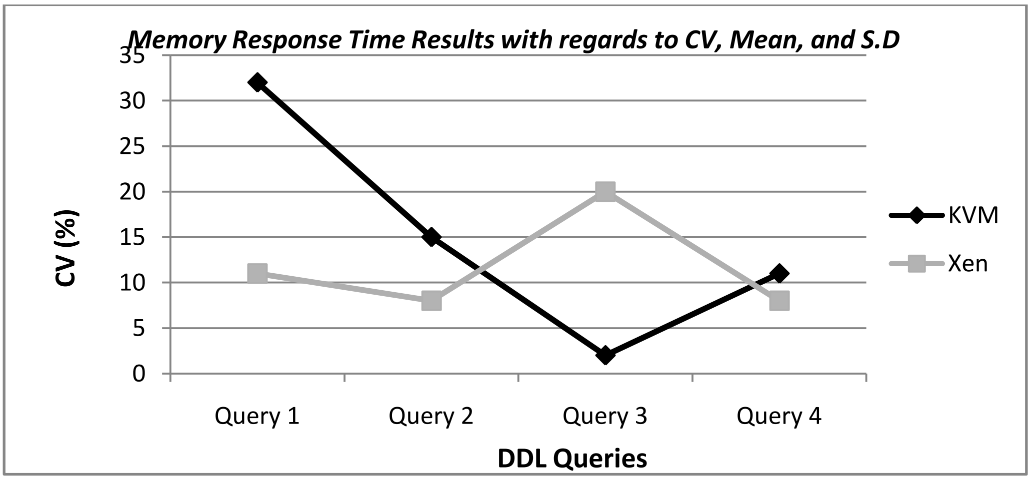


**Figure 4.4** The computed CV, mean, and S.D chart in term of memory response time for DDL queries.

## Performance of Cloud Managers and Hypervisors in DML Category

From CPU point of view, the main aim for testing this category is to find the effect of inserting new row to the table, delete data from the table, editing data in the table, and retrieving data from the table using complex sub conditions. The test was conducted over different sizes of tables. Thus, the average of results will be taken to ease the comparison. Table 4.5 shows the results of CPU response time for DML queries.

| **Category** | Data Manipulation Language (DML) | | | |
|---|---|---|---|---|
| **Data** | CPU Response Time | | | |
| **Query** | **OpenStack** | | **OpenNebula** | |
| | **KVM** | **Xen** | **KVM** | **Xen** |
| Query 5 | **32.6** | **37.6** | **29.3** | **30** |
| Query 6 | **1169.6** | **1019** | **1099** | **1069.3** |
| Query 7 | **707** | **565.6** | **614.6** | **516.6** |
| Query 8 | **44** | **49.3** | **73.3** | **36.6** |
| Query 9 | **328.6** | **228.6** | **222.3** | **210.3** |
| Average | **456** | **380** | **408** | **373** |

**Table 4.5**DML category results in term of CPU response time for OpenStack and OpenNebula.

These results show that query type have a direct effect on CPU response time. At query five and eight, CPU response time was short. At query seven and nine, CPU response time was increased forming the first gap. At query six, Long CPU response time existed compared with other queries forming the second gap.

Consequently, the queries that was responsible for inserting or updating one row in the table was performing well in both cloud managers, with some differences while using two types of hypervisors. Queries that was responsible to delete or retrieve from table depending on complex conditions, achieved long CPU response time. Queries that were responsible to insert whole table to new one achieved the longest CPU response time.

Figure 4.5 show results chart of DML queries in term of CPU response time for OpenStack and OpenNebula with hypervisors.

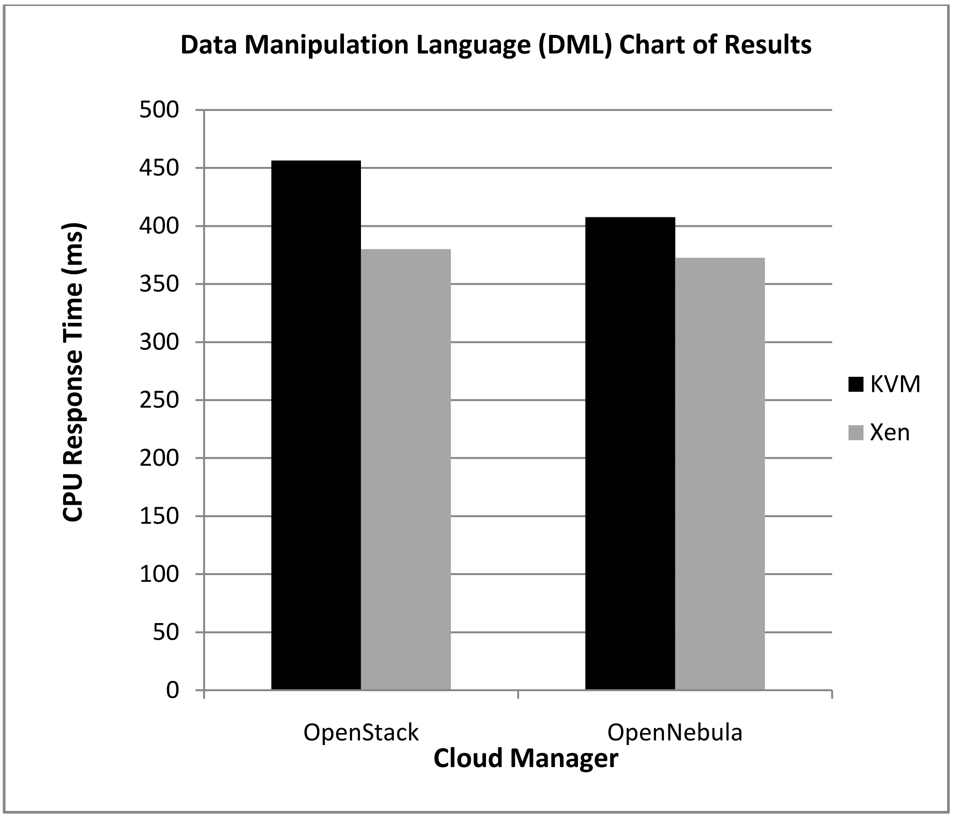


**Figure 4.5**CPU response time chart for OpenStack and OpenNebula with virtualization hypervisors in DML queries.

From Figure 4.5, the chart shows that OpenNebula cloud manager was performing better than OpenStack in term of CPU response time in DML queries. OpenNebula cloud manager with Xen hypervisor achieved the shortest CPU response time in DML queries compared with others. OpenStack cloud manager with KVM hypervisor achieved the longest CPU response time in DML queries. A drastic gap existed between KVM in OpenStack and KVM in OpenNebula. Thus OpenNebula had a direct effect to decrease the CPU response time in DML queries.

Furthermore, we worked on another evidence to compare hypervisor interactions for DML queries. Thus, KVM hypervisor registered high CV, which leads to negative implications on the queries of DML category in term of CPU response time. Table 4.6 shows the computed CV results of both hypervisors for DML queries with regards to CPU response time.

| Query | The Computed CV for KVM and Xen after executing DML queries | | | | | |
|---|---|---|---|---|---|---|
| | KVM | | | Xen | | |
| | Mean (□) | S.D (σ) | (CV) % | Mean (□) | S.D (σ) | (CV) % |
| 5 | 31 | 2.3 | 7.5 | 64 | 5.3 | 16 |
| 6 | 1134 | 50 | 4.4 | 1044 | 36 | 3.4 |
| 7 | 661 | 65 | 10 | 541 | 35 | 6.4 |
| 8 | 59 | 21 | 35.3 | 43 | 9 | 21 |
| 9 | 275 | 75 | 27 | 219 | 13 | 6 |

**Table 4.6** The computed CV, mean, and S.D results in term of CPU response time for DML queries.

The CV results achieved by KVM hypervisor was 7.5%, 4.4%, 10%, 35.3%, and 27% respectively for queries 5 - 9. Hence, the average of CV results in case of using KVM approximately equals 17% which reflects a negative effect on executing DML queries with KVM hypervisor. On the other hand, Xen hypervisor achieved better CV results with 10% as CV average. Thus, using Xen hypervisor for DML queries has positive implications on the performance with regards to CPU response time. Figure 4.6 shows the chart of CV results after executing DML queries in both environments.

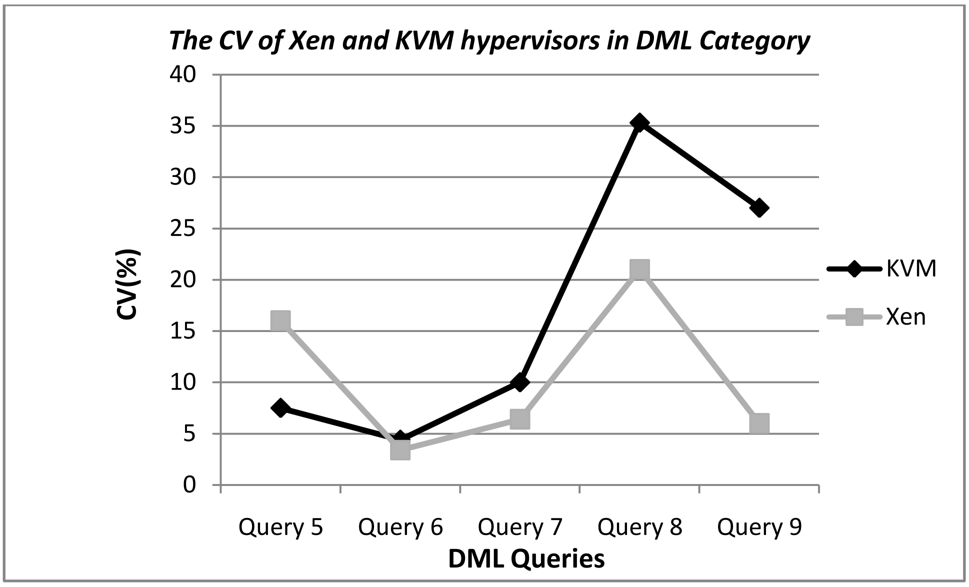


**Figure 4.6** CV, mean, and S.D chart for OpenStack and OpenNebula with virtualization hypervisors in DML queries.

From memory point of view, the main aim of testing DML queries; is to find the impacts of updating, inserting, and deleting rows on Hypervisors and cloud managers. The effects of DML queries can be extracted, if there are differences in memory response time for each query on known cloud manager with known hypervisor. Table 4.7 shows the results of executing DML queries in cloud environment in term of memory response time.

| Category | Data Manipulation Language | | | |
|---|---|---|---|---|
| Data | Memory Response Time | | | |
| Hypervisor | OpenStack | | OpenNebula | |
| Query # | KVM | Xen | KVM | Xen |
| Query 5 | 24.3 | 38.6 | 40 | 26 |
| Query 6 | 3087.3 | 2551.3 | 3244.3 | 2939.6 |
| Query 7 | 281.6 | 348.3 | 883.6 | 595.6 |
| Query 8 | 70.3 | 74.3 | 85 | 115.3 |
| Query 9 | 1681 | 1907 | 1642.6 | 1897.6 |
| **Average** | **1029** | **984** | **1179** | **1115** |

**Table 4.7**DML category results in term of memory response time for OpenStack and OpenNebula.

For convenience, we had been taken the average of the results for each query in multiple table sizes. Inserting one row to the table was performing well such as in query five, with some differences between hypervisor types in cloud managers. Updating one row in the table was performing well such as in query eight, with some differences between hypervisor types in cloud managers. Tangible increase in memory response time existed, while using multiple querying with complex search conditions such as in query nine. A drastic increase in memory response time existed, while copying one table's data to new one. Figure 4.7 shows a graph of memory response time results for OpenStack and OpenNebula using KVM and Xen hypervisors.

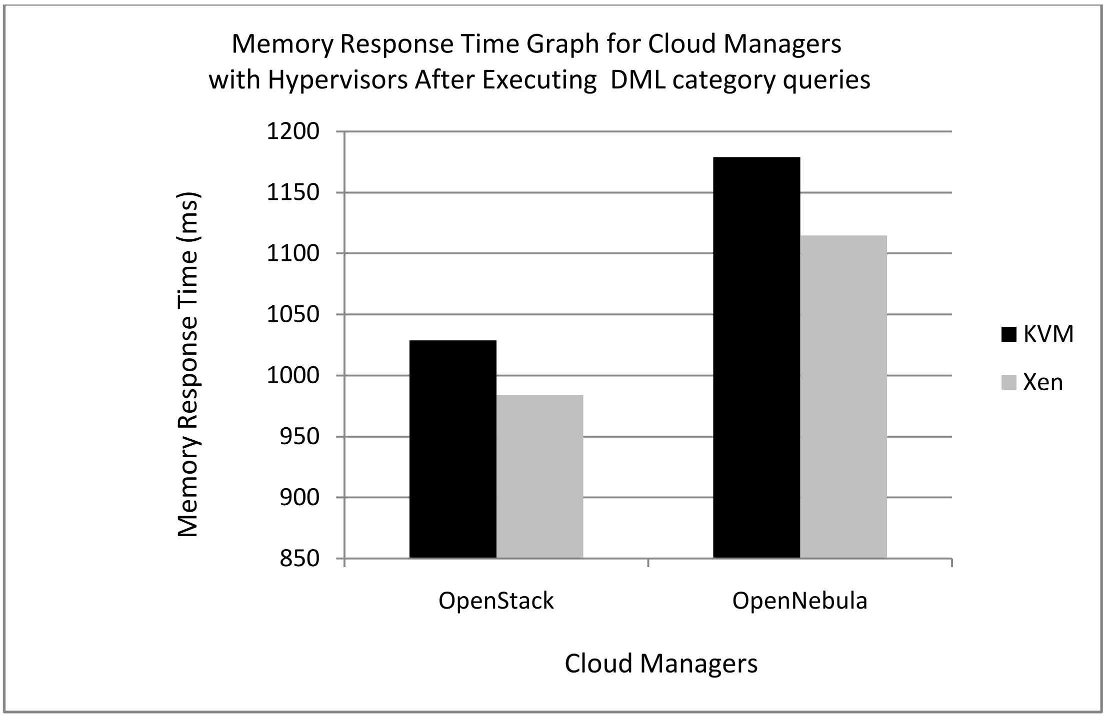


**Figure 4.7**Memory response time graph for OpenStack and OpenNebula with virtualization hypervisors in DML queries.

From Figure 4.7, OpenStack with both types of hypervisors (i.e. KVM and Xen) was performing well in term of memory response time. Tangible difference in memory response time found between OpenStack and OpenNebula. OpenStack with Xen hypervisor achieved the shortest memory response time. KVM hypervisor in OpenStack achieved better results than use it with OpenNebula. OpenNebula with KVM achieved the longest memory response time. Table 4.8 shows the computed CV results for both hypervisors in term of memory response time after executing DML queries.

| *The Computed CV for KVM and Xen after executing DML queries* | | | | | | |
|---|---|---|---|---|---|---|
| Query | KVM | | | Xen | | |
| | Mean (?) | S.D (σ) | (CV) % | Mean (?) | S.D (σ) | (CV) % |
| 5 | 32.15 | 11 | 35 | 32.3 | 8.9 | 27.6 |
| 6 | 3165.8 | 111 | 3.5 | 2745.4 | 274.5 | 10 |
| 7 | 582.6 | 426 | 73 | 472 | 175 | 37 |
| 8 | 77.65 | 10.4 | 13.3 | 94.8 | 29 | 30.5 |
| 9 | 1661.8 | 27 | 1.7 | 1902.3 | 7 | 0.34 |

**Table 4.8** The computed CV, mean, and S.D results in term of memory response time for DML queries.

From Table 4.8, the CV results of Xen hypervisor show positive effect on the performance of memory – with regards to memory response time – for each query in DML category with average 20%. On the other hand, KVM hypervisor registered higher CV results than in case of Xen, which represents that KVM had negative effect for DML queries. Figure 4.8 shows chart for CV computed results for each query in DML in term of memory response time.

For each query in DML category, KVM hypervisor achieved better results in queries 6 and 8. Thus, the CV percent results for both of them were 3.5%, and 13.3% respectively. However, Xen hypervisor achieved better results in other DML queries (i.e. query 5,7, and 9). Xen hypervisor registered 20% as a CV result for all DML queries. On the other hand, KVM hypervisor achieved 26% as a CV result for DML queries.

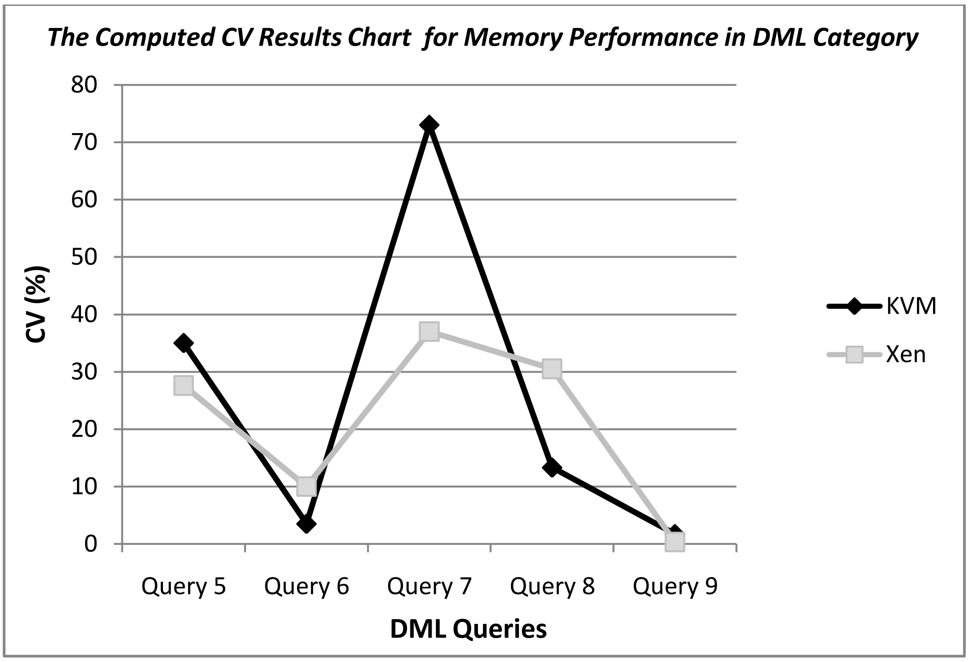


**Figure 4.8** Memory response time graph for CV results for both KVM and Xen hypervisors in DML queries.

## Performance of Cloud Managers and Hypervisors in DQL Category

From CPU point of view, The main aim of testing CPU performance after executing DQL queries; is to find the effects of retrieving data from tables using simple SELECT statement, retrieving data from tables using complex search conditions, and retrieving data from more than one table using different types of JOIN operations. Table 4.9 shows the CPU response time after executing DQL queries in cloud two cloud environments.

| Category | Data Query Language (DQL) | | | |
|---|---|---|---|---|
| Data | CPU Response Time | | | |
| Hypervisor | OpenStack | | OpenNebula | |
| Query # | KVM | Xen | KVM | Xen |
| Query 10 | 453.6 | 459.3 | 395 | 476 |
| Query 11 | 286.6 | 283.3 | 276.3 | 291.6 |
| Query 12 | 481 | 494.3 | 515.3 | 493.3 |
| Query 13 | 627.6 | 671.3 | 609.6 | 659.6 |
| Query 14 | 5182.3 | 5271 | 4619 | 4665.6 |
| Query 15 | 4567.3 | 4574.6 | 4485.3 | 5136 |
| Query 16 | 5499 | 4990 | 5003.6 | 4574.6 |
| Query 17 | 4036.3 | 3660 | 3953.6 | 3656.6 |
| Query 18 | 4803.6 | 4661.3 | 4683.6 | 4450.3 |
| Query 19 | 57855.3 | 57638.3 | 56573 | 56217 |
| Query 20 | 34948.3 | 34470.3 | 39866.3 | 34579 |
| **Average** | **10795** | **10652** | **10998** | **10473** |

**Table 4.9**DQL category results in term of CPU response time for OpenStack and OpenNebula.

These results show that retrieving data using JOIN operations was consuming the CPU response time such as in queries 13 till 18. A drastic increase in CPU response time existed after inserting complex condition with JOIN operation such as in queries 19 and 20. Thus, these results show that query types in DQL category have a clear effect on CPU response time.

For convenience, the average for each hypervisor used in cloud manager will be taken. Figure 4.9 shows the chart of average for all queries results for each hypervisor in cloud managers.

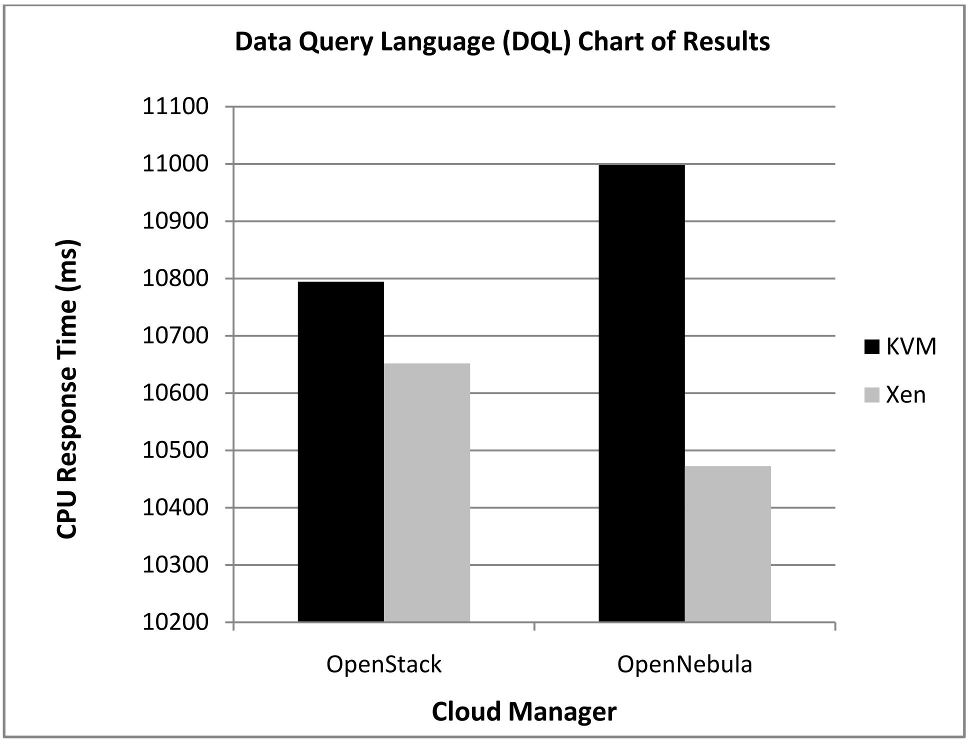


**Figure 4.9**CPU response time chart for OpenStack and OpenNebula with virtualization hypervisors in DQL queries.

From Figure 4.9, OpenNebula cloud manager with Xen hypervisor had the shortest CPU response time in DQL queries. KVM hypervisor with OpenStack cloud manager achieved better results than with OpenNebula in term of CPU response time in DQL queries. Using KVM hypervisor with OpenNebula cloud manager achieved the longest CPU response time for DQL category. Xen hypervisor achieved better results than KVM hypervisor in OpenStack cloud manager. Table 4.10 shows the computed CV results for both Xen and KVM hypervisors after executing DQL queries with regards to CPU response time.

The CV was used for comparing the behavior of Xen and KVM hypervisors in DQL queries in term of CPU response time, in order to find other evidence and make fair comparison between them.

| Query | KVM | | | Xen | | |
|---|---|---|---|---|---|---|
| | Mean | S.D | CV | Mean | S.D | CV |
| 10 | 424.3 | 41.43646 | 9.76584 | 467.65 | 11.80868 | 2.5251 |
| 11 | 281.45 | 7.2832 | 2.587742 | 287.45 | 5.868986 | 2.0417 |
| 12 | 498.15 | 24.25376 | 4.868767 | 493.8 | 0.707107 | 0.1432 |
| 13 | 618.6 | 12.72792 | 2.057537 | 665.45 | 8.273149 | 1.2432 |
| 14 | 4900.65 | 398.3132 | 8.127764 | 4968.3 | 428.0824 | 8.6163 |
| 15 | 4526.3 | 57.98276 | 1.281019 | 4855.3 | 396.9697 | 8.176 |
| 16 | 5251.3 | 350.3007 | 6.670742 | 4782.3 | 293.7322 | 6.1421 |
| 17 | 3994.95 | 58.47773 | 1.463791 | 3658.3 | 2.404163 | 0.0657 |
| 18 | 4743.6 | 84.85281 | 1.788785 | 4555.8 | 149.1995 | 3.2749 |
| 19 | 57214.15 | 906.723 | 1.584788 | 56927.65 | 1005.011 | 1.7654 |
| 20 | 37407.3 | 3477.551 | 9.296451 | 34524.65 | 76.86251 | 0.2226 |

**Table 4.10**CV results for DQL category in term of CPU response time for Xen and KVM hypervisors.

The CV results show differences in CPU response time by using both hypervisors. Thus, we get that KVM hypervisor had positive effect in DQL queries that is responsible for right outer join (i.e. query 15) and inner join (i.e. query 18). Furthermore, Xen hypervisor had positive effect on the CPU response time for all other DQL queries. As an average, we can say that Xen hypervisor had 3% of CPU response time, while in KVM the average of CV was 4.1%. Thus, the difference between KVM and Xen hypervisors was too short. Figure 4.10 shows a chart of CV results for both hypervisors in DQL category.

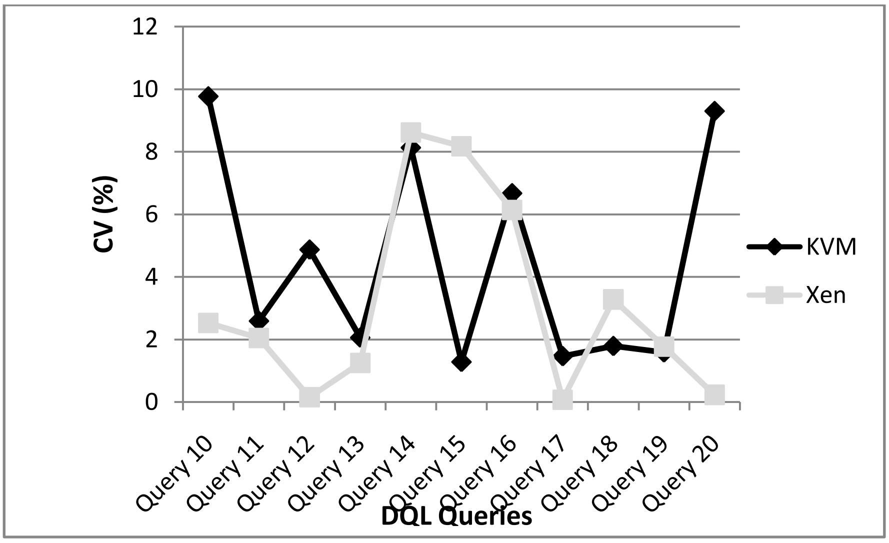


**Figure 4.10**CV results chart in term of CPU response time for Xen and KVM in DQL queries.

From memory point of view, the main aim of testing the DQL queries, is to find the impacts of retrieving data of tables using simple search condition, retrieving data of tables using complex search conditions, retrieving data from two or more tables using different types of JOIN operations, and retrieving data from more than one table using complex search conditions. Table 4.11 shows the results of memory response time after executing DQL queries in cloud environment.

| **Category** | Data Query Language (DQL) | | | |
|---|---|---|---|---|
| **Data** | Memory Response Time | | | |
| **Hypervisor** | **OpenStack** | | **OpenNebula** | |
| **Query #** | **KVM** | **Xen** | **KVM** | **Xen** |
| Query 10 | **1843** | 1731 | **1808.6** | 1647.3 |
| Query 11 | **633.3** | 565.6 | **652** | 604 |
| Query 12 | **2094** | 1657 | **1797.3** | 1770 |
| Query 13 | **4646.6** | 4390.6 | **4063** | 4747 |
| Query 14 | **6385** | 5928.6 | **6267** | 6304.3 |
| Query 15 | **6450** | 5804.6 | **6681** | 6188.3 |
| Query 16 | **8759.3** | 4995.6 | **8693** | 8036 |
| Query 17 | **14743.3** | 13933.6 | **15090** | 15354.3 |
| Query 18 | **19861** | 18526.3 | **18835.3** | 18176.6 |
| Query 19 | **161789** | 275471 | **165142.7** | 167539 |
| Query 20 | **189066.3** | 149524 | **212829.7** | 165605.7 |
| **Average** | **37843** | **43866** | **40169** | **35997** |

**Table 4.11**DQL category results in term of memory response time for OpenStack and OpenNebula.

From Table 4.11, the results show that memory response time was consumed drastically for queries that contain a condition with JOIN operations, such as in query 19 and query 20. Similar behavior in consuming memory response time existed in left and right outer joins such as in queries 14 and 15. Tangible increase in memory response time existed by using full outer join such as in query 16. Using natural join affect the memory's response time drastically such as in query 17. Both types of hypervisors (i.e. KVM and

Xen) achieved different results in OpenStack and OpenNebula. Figure 4.11 shows a graph of memory response time for OpenStack and OpenNebula with KVM and Xen hypervisors after executing DQL queries.

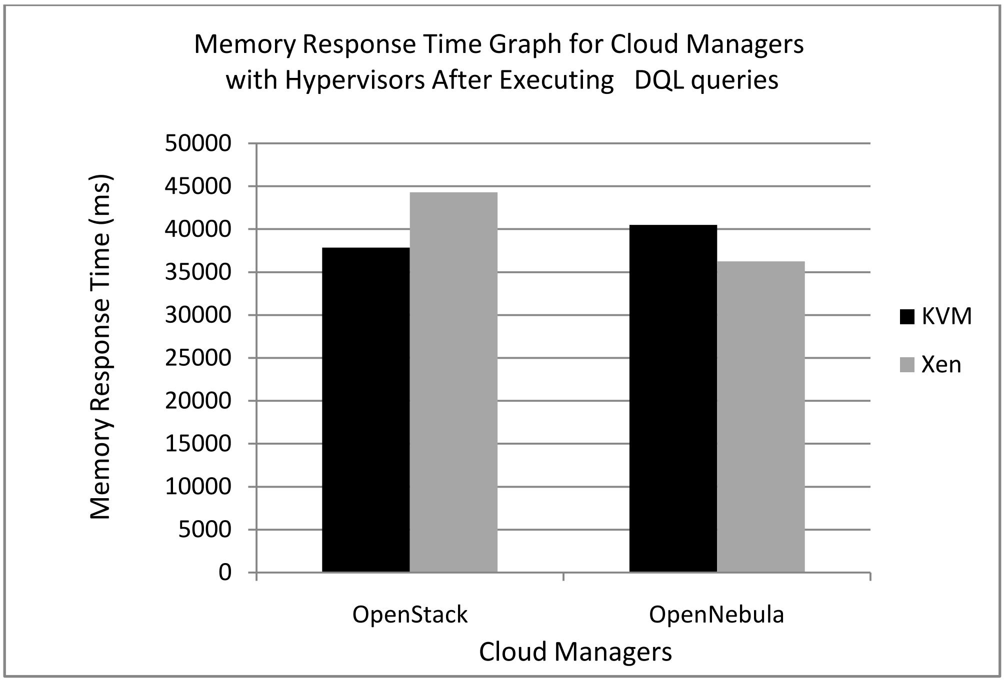


**Figure 4.11**Memory response time graph for OpenStack and OpenNebula with virtualization hypervisors in DQL queries.

From Figure 4.11, OpenNebula cloud manager with Xen hypervisor achieved the shortest memory's response time. OpenStack cloud manager with Xen hypervisor achieved the longest memory's response time. Thus, using Xen hypervisor with OpenStack had negative effects for DQL queries in term of memory response time. The use of KVM hypervisor in OpenNebula had negative effects for DQL queries in term of memory response time.

## 4.4. The CPU Response Time of Hypervisor and Cloud Manager over Multiple Data Sizes

It has been noted that the relation between the data size and response time of CPU and memory in linear (i.e. when we increase the data size the response time will be increased). For that, we decide to use this factor (i.e. Data size) as a control parameter in our experiment. Thus, to focus on the results were extracted after executing queries form the same category in the same database table with different table's size. The results extracted for three stages 1100 rows for the first stage, 2200 rows for the second stage, and 3300 rows for the third stage.

### CPU Response Time Results of the First Experiment overMultiple Data Sizes

In this section, we discussed the results of CPU response time in the first experiment. The main aim of testing CPU response time of hypervisor and cloud manager after executing queries is to find the effects occurred, while changing table's sizes in database. Table 4.12 shows the results of the first experiment by using OpenStack with KVM and Xen Hypervisors in term of CPU response time for all SQL queries executed in database tables over the three stages of data sizes.

For convenience, the comparison in this section will measure the effect of data size variable on cloud manager with virtualization hypervisor in term of CPU response time. The first experiment represented with cloud manager (i.e. OpenStack) and virtualization hypervisor (i.e. KVM and Xen).

From table 4.12 we can find that there is a tangible increase in CPU response time following the increase of data size in table. A drastic increase found in CPU response time for queries that contain joining operations. Thus, queries that was responsible for retrieving data from two or more tables, caused a high CPU response time too.

| Exp. | First Experiment (OpenStack + ( KVM \| \| Xen ) ) | | | | | |
|---|---|---|---|---|---|---|
| Data | CPU Time | | | | | |
| Query # | OpenStack | | | | | |
| | KVM | | | Xen | | |
| | 1100 rows (Entries) | 2200 rows (Entries) | 3300 rows (Entries) | 1100 rows (Entries) | 2200 rows (Entries) | 3300 row (Entries) |
| Query 1 | 47 | 47 | 50 | 16 | 24 | 51 |
| Query 2 | 1094 | 1287 | 1301 | 1047 | 1244 | 1297 |
| Query 3 | 985 | 1024 | 841 | 875 | 908 | 800 |
| Query 4 | 15 | 94 | 60 | 16 | 31 | 47 |
| Query 5 | 16 | 23 | 59 | 18 | 27 | 68 |
| Query 6 | 672 | 1482 | 1355 | 532 | 1225 | 1300 |
| Query 7 | 328 | 825 | 970 | 219 | 604 | 874 |
| Query 8 | 32 | 38 | 62 | 47 | 46 | 55 |
| Query 9 | 266 | 321 | 399 | 93 | 219 | 374 |
| Query 10 | 328 | 451 | 582 | 344 | 473 | 561 |
| Query 11 | 203 | 313 | 344 | 219 | 272 | 359 |
| Query 12 | 140 | 587 | 716 | 156 | 622 | 705 |
| Query 13 | 219 | 635 | 1029 | 266 | 767 | 981 |
| Query 14 | 1797 | 6750 | 7000 | 1781 | 6750 | 7282 |
| Query 15 | 1797 | 5687 | 6218 | 1750 | 5781 | 6193 |
| Query 16 | 1968 | 7063 | 7466 | 1860 | 5985 | 7125 |
| Query 17 | 2297 | 4875 | 4937 | 1313 | 4750 | 4917 |
| Query 18 | 2187 | 5938 | 6286 | 2063 | 5766 | 6155 |
| Query 19 | 15875 | 70078 | 87613 | 15453 | 69782 | 87680 |
| Query 20 | 28469 | 35266 | 41110 | 27656 | 35154 | 40601 |
| Average | 2937 | 7139 | 8420 | 2786 | 7022 | 8371 |

**Table 4.12**The results of OpenStack with KVM and Xen in term of CPU response time over multiple data sizes.

These results show that OpenStack cloud manager with both hypervisors (i.e. KVM and Xen) were affected due the change of table's data size. At (1100 rows),the differences in

CPU response time between KVM and Xen were found clearly. At (2200 rows), the results were doubled for majority of queries with clear differences between KVM and Xen. At (3300 rows), tangible rise in the results were found, tripled in some cases, and a drastic gaps found between KVM and Xen.

For convenience, we will take the average of CPU response time to make the comparison between hypervisors inOpenStack cloud manager in term of data size. Figure 4.12 shows the chart of CPU response time for multiple data sizes in OpenStack cloud.

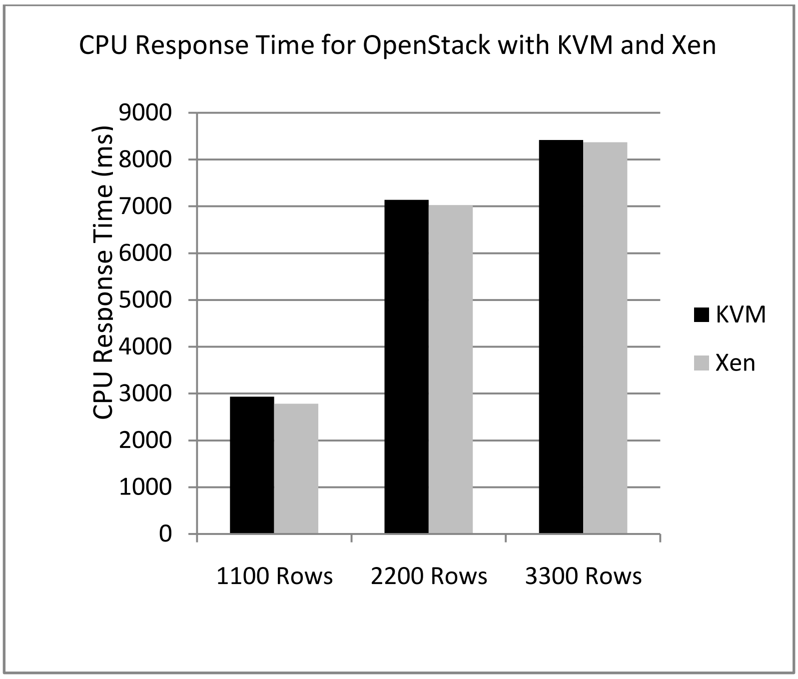


**Figure 4.12**CPU response time chart in the first experiment over multiple data sizes.

From Figure 4.12, it shows that Xen hypervisor in OpenStack were performing better than KVM in OpenStack. Both KVM and Xen hypervisors in OpenStack cloud were affected from the change in table's data size. OpenStack cloud was performing well at low data size (i.e. 1100 rows). OpenStack cloud with both hypervisors faced rise in CPU response time. OpenStack cloud faced a drastic gap after doubling the size of table (i.e. 2200 rows). OpenStack cloud with both hypervisors continued to increase in CPU response time after tripling the number of rows in table.

## CPU Response Time Results in Second Experiment for Multiple Data Sizes

In this section, we will discuss the results of CPU response time in the second experiment. The second experiment represented in OpenNebula as a cloud manager with KVM and Xen for virtualization hypervisor. In this section, we will measure the effect of table's data size on CPU response time in the second experiment. Table 4.13 shows the results of CPU response time after executing the proposed queries from different categories in variable number of rows in table.

| Data | Second Experiment (OpenNebula + ( KVM \| \| Xen ) ) (CPU Time) | | | | | |
|---|---|---|---|---|---|---|
| Query # | OpenNebula | | | | | |
| | KVM | | | Xen | | |
| | 1100 rows (Entries) | 2200 rows (Entries) | 3300 rows (Entries) | 1100 rows (Entries) | 2200 rows (Entries) | 3300 row (Entries) |
| Query 1 | 47 | 47 | 54 | 16 | 24 | 54 |
| Query 2 | 1094 | 1287 | 1669 | 1047 | 1244 | 1532 |
| Query 3 | 985 | 1024 | 1551 | 875 | 908 | 1182 |
| Query 4 | 15 | 94 | 205 | 16 | 31 | 88 |
| Query 5 | 15 | 23 | 50 | 16 | 23 | 51 |
| Query 6 | 641 | 1355 | 1301 | 625 | 1286 | 1297 |
| Query 7 | 281 | 722 | 841 | 203 | 547 | 800 |
| Query 8 | 94 | 66 | 60 | 31 | 32 | 47 |
| Query 9 | 157 | 178 | 332 | 125 | 156 | 350 |
| Query 10 | 203 | 410 | 572 | 438 | 421 | 569 |
| Query 11 | 203 | 297 | 329 | 218 | 315 | 342 |
| Query 12 | 125 | 690 | 731 | 172 | 600 | 708 |
| Query 13 | 125 | 600 | 1104 | 297 | 692 | 990 |
| Query 14 | 1610 | 5922 | 6325 | 1526 | 6125 | 6346 |
| Query 15 | 1625 | 5812 | 6019 | 1391 | 6875 | 7142 |
| Query 16 | 1891 | 5781 | 7339 | 1875 | 4516 | 7333 |
| Query 17 | 2203 | 4765 | 4893 | 1844 | 4329 | 4797 |
| Query 18 | 1891 | 5813 | 6347 | 1719 | 5532 | 6100 |
| Query 19 | 18375 | 71750 | 79594 | 17563 | 70829 | 80259 |
| Query 20 | 37781 | 39818 | 42000 | 25593 | 37123 | 41021 |
| Average | 3468 | 7323 | 8066 | 2779 | 7080 | 8050 |

**Table 4.13**OpenNebula with KVM and Xen in term of CPU response time over multiple data sizes.

The results show that table's data size had a direct effect on CPU response time. OpenNebula cloud was performing well at 1100 rows. Virtualization hypervisors used in OpenNebula (i.e. KVM and Xen) had a drastic gap in CPU response time especially at 1100 rows. At 2200 rows, the CPU response time doubled after doubling the number of rows in table. At 3300 rows, the CPU response time continued to rise.

For convenience, the average of CPU response time will be taken to make the comparison with table's data size. Figure 4.13 shows a chart of CPU response time at different number of rows in table for the second experiment.

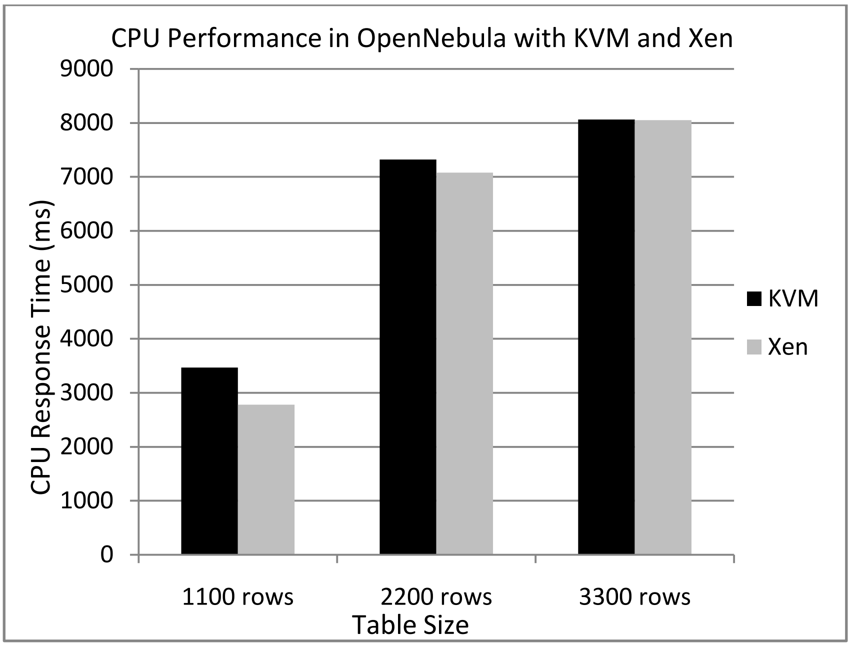


**Figure 4.13**Chart of OpenNebula with KVM and Xen in term of CPU response time over multiple data sizes.

From Figure 4.13, the OpenNebula cloud manager with Xen hypervisor was performing well in all table sizes. OpenNebula with KVM hypervisor achieved high CPU response time. New indicator existed in this experiment, that both virtualization hypervisors (i.e. KVM and Xen) achieved small differences in CPU response time, after doubling and tripling number of rows in table.

## Discussion

In this section, the test has been divided for two environments.The main aim for test was to find the effects of table's data sizes on CPU response time. At first part, it is tested the OpenStack cloud that uses either KVM or Xen hypervisors, in term of CPU response time. In the second part, the test was for OpenNebula cloud with either KVM or Xen hypervisors, in Term of CPU response time with table's data sizes. In both parts of test, the tables data size affect the CPU response time directly. KVM hypervisor in both environments achieved high CPU response time. Xen hypervisor in both environments was performing well compared with KVM. Using OpenStack cloud manager with KVM hypervisor was performing better than using KVM in OpenNebula. Using Xen hypervisor with OpenNebula cloud manager was performing better than using Xen in OpenStack. Increasing the number of rows in tables has a direct effect on increasing the CPU response time. Figure 4.14 shows comparison chart between OpenStack and OpenNebula in term of CPU response time.

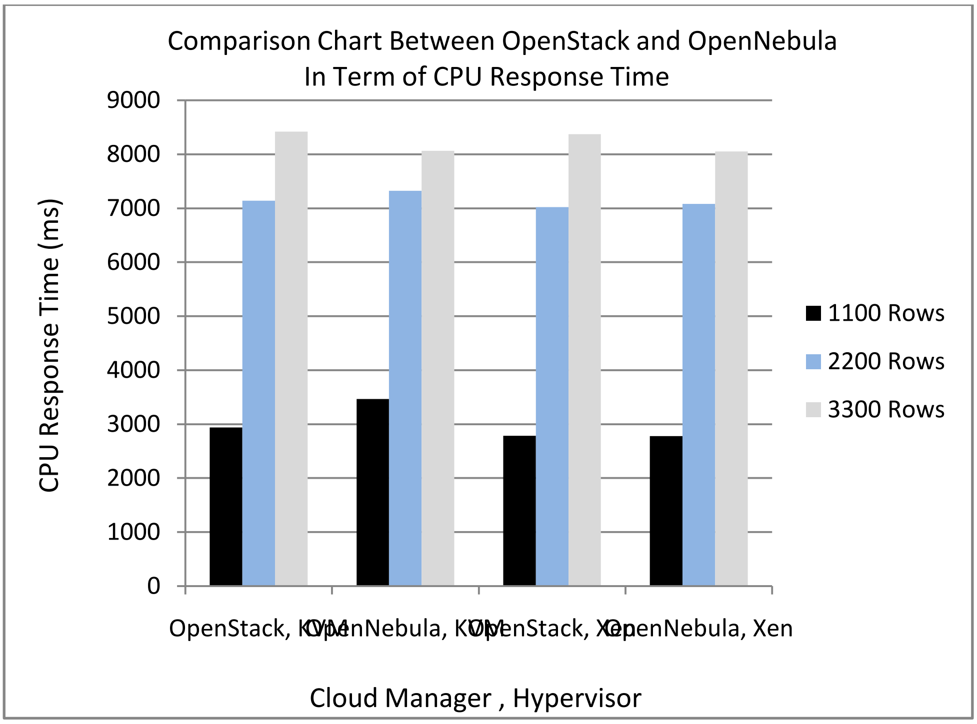


**Figure 4.14** Comparison chart between OpenStack and OpenNebula in term of CPU response time.

## 4.5. Memory Response Time of Hypervisor and Cloud Manager Over Multiple Data Sizes

In this section, we will discuss the effects of changing the number of rows in table over hypervisors and cloud manager in term of memory response time. The test results were for three stages (i.e. table's data sizes). At first stage, number of rows was 1100 rows. At second stage, number of rows was doubled to be 2200 rows. At third stage, the number of rows was tripled to be 3300 rows. Memory response time will be evaluated in two experiments. At the first experiment, OpenStack will be used as cloud manager with two types of hypervisors KVM and Xen. At the second experiment, OpenNebula will be used as cloud manager with two types of hypervisors KVM and Xen.

### Memory Response Time Results in the First Experiment for Multiple Data Sizes

In this section, the performance of cloud manager and virtualization hypervisors will be discussed. The performance will measured in term of memory response time. The aim of this test is to find the effect of table's data size on memory response time. This experiment represented by OpenNebula cloud manager with KVM and Xen hypervisors. Table 4.14 shows the results of this experiment over a multiple table's data sizes.

These results show that the size of data in table had a direct effect on memory response time. The increasing behavior in memory response time existed in both hypervisor (i.e. KVM and Xen) and for all queries in the experiment.

| Exp. | First Experiment (OpenStack + ( KVM \| \| Xen ) ) | | | | | |
|---|---|---|---|---|---|---|
| Data | RAM Response Time | | | | | |
| Query # | OpenStack | | | | | |
| | KVM | | | Xen | | |
| | 1100 rows (Entries) | 2200 rows (Entries) | 3300 rows (Entries) | 1100 rows (Entries) | 2200 rows (Entries) | 3300 row (Entries) |
| Query 1 | 23 | 130 | 315 | 49 | 82 | 421 |
| Query 2 | 1227 | 1830 | 4040 | 2284 | 2032 | 3909 |
| Query 3 | 2652 | 4456 | 7953 | 316 | 2918 | 7333 |
| Query 4 | 54 | 157 | 248 | 90 | 255 | 178 |
| Query 5 | 27 | 16 | 30 | 17 | 6 | 93 |
| Query 6 | 246 | 1329 | 7687 | 181 | 1206 | 6267 |
| Query 7 | 160 | 408 | 277 | 182 | 533 | 330 |
| Query 8 | 50 | 24 | 137 | 96 | 10 | 117 |
| Query 9 | 242 | 1305 | 3496 | 37 | 1924 | 3670 |
| Query 10 | 1714 | 1833 | 1982 | 1580 | 1795 | 1998 |
| Query 11 | 201 | 836 | 863 | 356 | 751 | 805 |
| Query 12 | 170 | 2692 | 3420 | 197 | 2843 | 3182 |
| Query 13 | 3081 | 5294 | 5565 | 3084 | 5089 | 5551 |
| Query 14 | 3519 | 6853 | 8783 | 3912 | 7434 | 8523 |
| Query 15 | 2534 | 7446 | 9370 | 2664 | 7213 | 8819 |
| Query 16 | 3315 | 6265 | 16698 | 2912 | 6542 | 17351 |
| Query 17 | 8577 | 16885 | 18768 | 7418 | 17436 | 18549 |
| Query 18 | 8841 | 23141 | 25601 | 9448 | 23041 | 27609 |
| Query 19 | 18240 | 36422 | 168705 | 22549 | 37417 | 151809 |
| Query 20 | 157623 | 186938 | 205675 | 159911 | 209946 | 216930 |
| Average | 10624 | 15213 | 24480 | 10864 | 16423 | 24172 |

**Table 4.14**Memory Response Time Results of the First Experiment over different table's sizes.

These results show that OpenStack cloud manager with KVM and Xen hypervisor was performing well in small data size of table. After doubling the size of data in table, the response time of memory was doubled. Thus, the rise in memory response time continued after tripling the size of table. Figure 4.15 shows a chart of memory response time for this experiment.

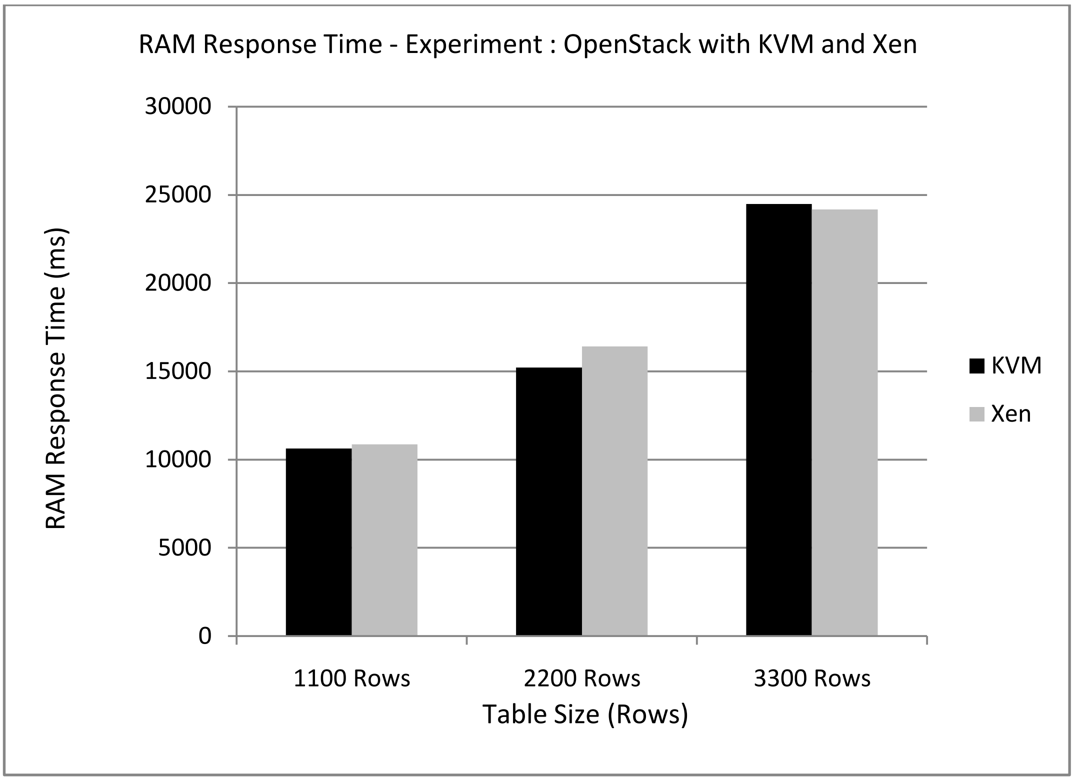


**Figure 4.15** Chart of memory response time results in the first experiment over different table's sizes.

From Figure 4.15, the results show that OpenStack with KVM was performing better than OpenStack with Xen at 1100, and 2200 rows. At 3300 rows, small difference in memory response time existed. A drastic gap in memory response time existed after increasing the size of tables.

## Memory Response Time Results in the Second Experiment for Multiple Data Sizes

In this section, queries will be executed in the second experiment. This experiment represented in OpenNebula as cloud manager with KVM and Xen hypervisors. Memory response time will be measured in order to show the performance while changing the size of table in database. Table 4.15 shows the results that were collected from the second experiment.

| Exp. | Second Experiment (OpenNebula + ( KVM \| \| Xen ) ) | | | | | |
|---|---|---|---|---|---|---|
| Data | RAM Response Time | | | | | |
| Query # | OpenNebula | | | | | |
| | KVM | | | Xen | | |
| | 1100 rows (Entries) | 2200 rows (Entries) | 3300 rows (Entries) | 1100 rows (Entries) | 2200 rows (Entries) | 3300 row (Entries) |
| Query 1 | 105 | 211 | 435 | 3 | 162 | 477 |
| Query 2 | 2850 | 2152 | 3767 | 2698 | 2435 | 4090 |
| Query 3 | 2802 | 4278 | 7533 | 2669 | 4720 | 6736 |
| Query 4 | 85 | 161 | 148 | 73 | 146 | 251 |
| Query 5 | 45 | 35 | 40 | 32 | 23 | 23 |
| Query 6 | 534 | 1778 | 7421 | 377 | 1441 | 7001 |
| Query 7 | 212 | 1269 | 1170 | 249 | 886 | 652 |
| Query 8 | 151 | 17 | 87 | 195 | 39 | 112 |
| Query 9 | 31 | 1409 | 3488 | 76 | 1845 | 3772 |
| Query 10 | 1621 | 1838 | 1967 | 1170 | 1818 | 1954 |
| Query 11 | 472 | 596 | 888 | 354 | 590 | 868 |
| Query 12 | 66 | 2233 | 3093 | 175 | 1961 | 3174 |
| Query 13 | 642 | 5727 | 5820 | 3112 | 4999 | 6130 |
| Query 14 | 3085 | 7201 | 8515 | 3065 | 7140 | 8708 |
| Query 15 | 2192 | 7150 | 10701 | 2277 | 7537 | 8751 |
| Query 16 | 2896 | 5958 | 17225 | 2311 | 6433 | 15364 |
| Query 17 | 6087 | 17533 | 18650 | 8415 | 18947 | 18701 |
| Query 18 | 7200 | 23509 | 23797 | 8030 | 24190 | 23410 |
| Query 19 | 7401 | 37225 | 39174 | 8936 | 37966 | 40014 |
| Query 20 | 149068 | 151903 | 218518 | 156546 | 152015 | 224256 |
| Average | 9377 | 13609 | 18621 | 10038 | 13764 | 18722 |

**Table 4.15**Memory response time results of the second experiment over different table's sizes.

These results show that OpenNebula with both hypervisors types was performing well in small table's size. The rise in memory response time found clearly after increasing the number of rows in table. Consequently, the size of table had a direct effect on memory response time.

For convenience, the average of memory response time result will be taken to ease the comparison between KVM and Xen in case of OpenNebula cloud manager. Figure 4.16 shows the chart of the averages for both hypervisors in different table's sizes.

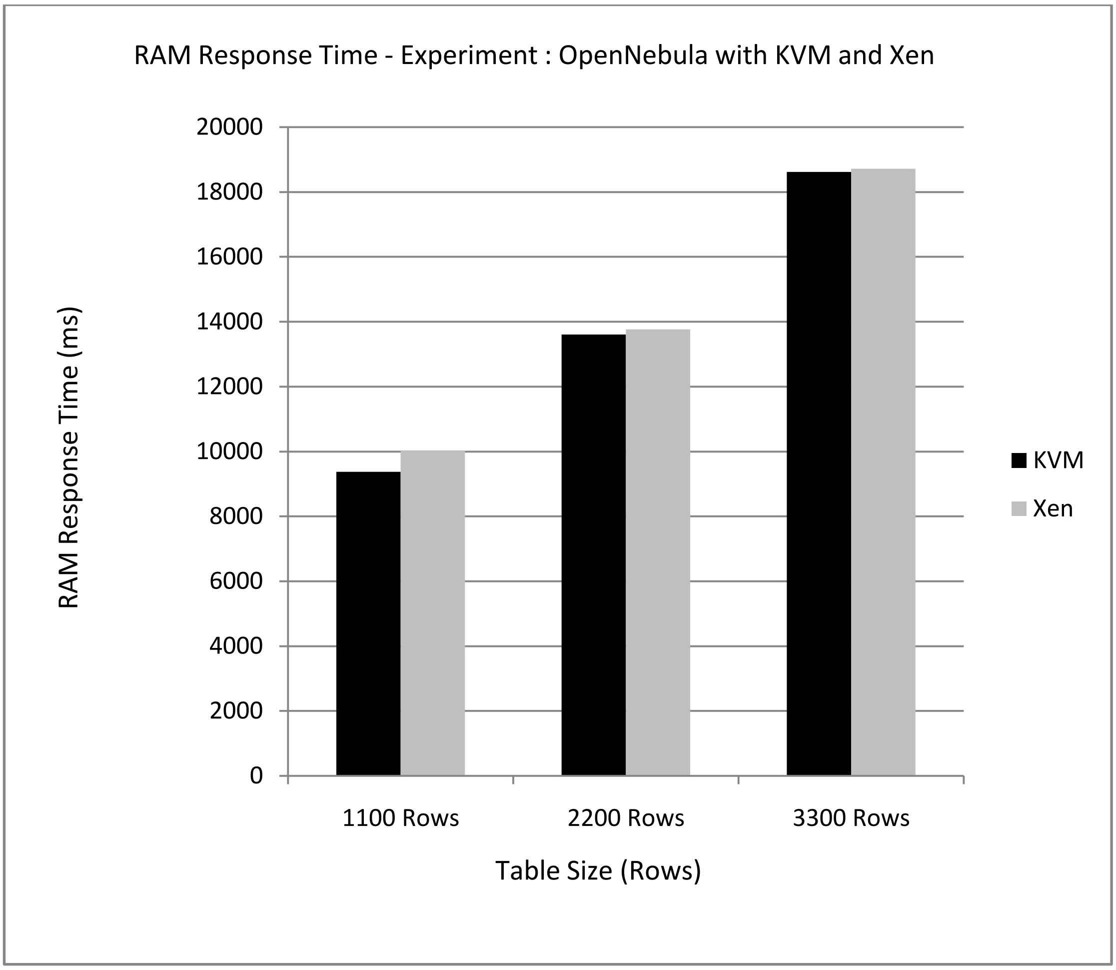


**Figure 4.16**Chart of memory response time results in the second experiment over different table's sizes.

From Figure 4.16, we can find a tangible rise in memory response time after increasing the number of rows in table. The use of KVM hypervisor with OpenNebula cloud manger achieved better results than using Xen hypervisor. At 1100 rows, the gap in result between KVM and Xen existed clearly. At 2200 and 3300 of rows, KVM hypervisor with OpenNebula decreases the gap.

## Discussion

The test has been divided in two parts. At first part, testing memory response time for OpenStack cloud manager with KVM and Xen hypervisors. The results show that increasing table's size had a tangible effect in increasing memory response time. A drastic gap existed after doubling and tripling number of rows. Xen hypervisor with OpenStack cloud manager achieved the best results in the first part of test. At the second part, testing memory response time for OpenNebula cloud manager with KVM and Xen hypervisors. The results show that increasing table's size had a tangible effect on memory response time. Figure 4.17 shows comparison chart between OpenStack and OpenNebula in term of memory response time over different table's sizes.

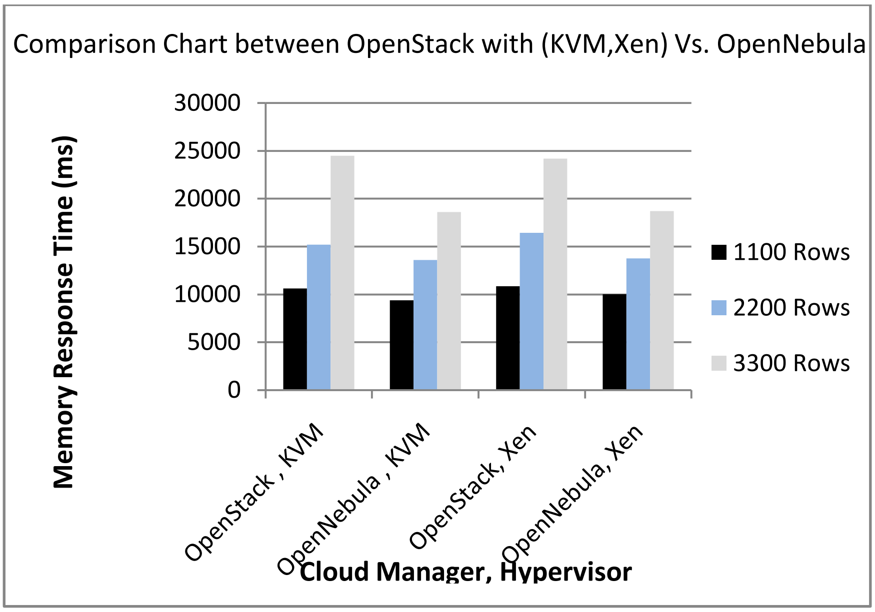


**Figure 4.17**Comparison chart between OpenStack and OpenNebula in term of memory response time.

From data size point of view, OpenNebula cloud manager was performing better than OpenStack in term of memory response time. The use of KVM hypervisor with OpenNebula achieved better results than using Xen hypervisor in term of response time.

## 4.6 Technical Recommendations

The benefit from this research is to compare the performance of different types of hypervisors that were working in different types of cloud managers in term of CPU and memory response time. Hence, findings of this research were summarized as recommendations in order to select the suitable cloud manager and hypervisor. The recommendations were as the following:

- From CPU response time and query type perspectives, it is recommended to selectOpenNebula with Xen for DML and DQL queries as well as selects OpenStack with Xen for DDL queries.

- From Memory response time and query type perspective, it is recommended to select OpenStack with Xen for DDL and DQL queries as well as selecting OpenNebula with KVM for DQL queries.

- From CPU response time and dataset size perspectives, it is recommended to select OpenStack with Xen for first dataset size (i.e. 1100 rows) as well as selects OpenNebula with Xen for third data size (3300 rows), after doubling and tripling dataset sizes Xen hypervisor showed good behavior in performance.

- From memory response time and dataset sizes perspectives, it is recommended to select OpenNebula with KVM for first dataset size (i.e. 1100 rows) as well as selects OpenNebula with Xen for third data size (3300 rows), after doubling and tripling dataset sizes Xen hypervisor showed good behavior in performance.

.

# Chapter Five

# Conclusions and Future Work

## 5.1. Overview

This chapter summarizes the conclusions of our work, and suggested recommendations for using the suitable cloud manager with virtualization hypervisor – with regard to performance – for query type in database. In section 5.2, the main conclusion is presented. In section 5.3, future works.

## 5.2. Conclusions

According to the goals and experimental results, we can find that two basic parameters influencedthe performance of cloud manager and hypervisor clearly. The first parameter is the type of query that was executed on DB running in the cloud. Different types of queries were executed in our experiments. The test query types were DDL, DML, and DQL queries. Hence, the queries achieved different performance results while changing the type of cloud manager and hypervisor. The second parameter is the dataset size (i.e. Table size in DB). The queries were executed at different dataset sizes. Hence, the performance influenced clearly after doubling and tripling the size of datasets. Our experiments succeed in finding the best performance according to query type and dataset size.

The results of the first experiment showed that the CPU and memory response time has been increased based on using OpenStack with KVM hypervisor. As well as, using Xen hypervisor achieved better results based on all query types that had been executed.

The results of the second experiment showed that the CPU and Memory response time has been decreased due using the OpenNebula cloud manager. Xen hypervisor in this experiment achieved better results than KVM for DDL and DML queries. Change in Memory response time existed while using KVM hypervisor for DQL queries.

The control parameter (i.e. dataset size) reflects that our experiments were realized. Thus, by increasing the dataset size both CPU and memory response times were increased.

## 5.3. Future Works

This research focused on evaluating the performance of cloud manager and hypervisor,and opens the door for finding the suitable cloud manager and hypervisor for each query in SQL query categories. This research presents a way to make performance comparison between other leading hypervisors and cloud managers. Covering more domains using the proposed test experiments, will enable more and more domains to be evaluated. By applying the proposed test experiments for comparing commercial cloud package will handle to realize the results. Also, achievingmore accurate results is still a topicof continuous and constant research.

# References


- ABLES, T., DAHWAN, P. and CHADRASEKRAN, B. (2005, Augest). *An Overview of Xen Virtualization. DELL Power Solutions*. (pp. 109-111).

- Abuakibash, M. and Elleithy, K. (2012, July). *Cloud Computing: The Future of IT Industry*. International Journal of Distributed and Parallel System (IJDPS). Volume 3, Number 4.

- Ajoudanian, S. and Ahmadi, M. (2012). *A Novel Data Security Model for Cloud Computing*. International Journal of Engineering and Technology. Vol. 4, Issue 3. ( June 2012).

- Al Morsy, M., Grundy, J. and Muller, I. (2010, November). *An Analysis of the Cloud Computing Security Problem*. In Proceeding of APSEC 2010 Cloud Workshop, Sydney, Australia.

- Al-Bahadili, H., Al-Sabbah, A., and Abu Arqoub, M. (2013). *Modeling and Analysis of Cloud Collaborative Commerce.* International Journal on Cloud Computing and Architecture (IJCCSA). Vol. 3, Issue 1. DOI: 10.5121/ijccsa.2013.3.101. (Feb. 2013).

- Al-Bahadili, H., Qtishat, H., and Naoum, R. (2013). *Speeding Up the Web Crawling Process on A Multi-core Processor Using Virtualization*.

International Journal on Web Service Computing (IJWSC). Vol. 4, Issue 1, Mar.2013.

- Alsmadi, I., (2013). *Software Development Methodologies for Cloud Computing*. Software Development Techniques for Construction Information System Design. DOI:10.41018/1978-1-4666-3679-8 Ch:006, Book Business Strategies and Software Development Technologies for Constructive Information System Design, IGI 2013. (pp. 110-117).

- Ashktorab, V., and Taghizadeh, S. (2012, October). *Security Threats and Countermeasures in Cloud Computing*. International Journal of Application or Innovation in Engineering of Management (IJAIEM). Volume 1, Issue 2.

- Bahga, A. and Madisetti, V. (2013, February). *Performance Evaluation for Multitier Cloud Applications*. Journal of Software Engineering and Application. Volume 6, Number 2. (pp 74-83). JSEA.

- Binu, A. and Kumar, G. (2011). *Virtualization Techniques: A Methodical Review of Xen and KVM*. Communication in Computer Science and Information Science, Volume 190, DOI: 10.1007/978-3-642-22709-7-40. (pp. 399-410). Springer.

- Blanco, C. and Sotomayer, B. (2010). *OpenNebula Tutorial*. Cloudcom 2010, Indianapolis, USA. Nov. 30 – Dec. 20. Page (15).

- Buyya, R., Garg, S., and Calheiros, R. (2011). *SLA-Oriented Resource Provisioning for Cloud Computing:Challenges, Architecture, and Solutions*. Cloud and Service Computing (CSC), 2011, International Conference on, Vol., no., PP. 1, 10, 12-14 Dec. 2011, DOI:10.1109/CSC.2011.6138522. IEEE

- Chen, G., and Gillen, A. (2011). *KVM for Server Virtualization: An Open Source Solutions Comes of Age*. White Paper Sponsored By IBM. International Data Corporation IDC. No. 228115.

- Dillon, T., Wu, C., and Chang, E. (2010). *Cloud Computing: Issues and Challenges*. IEEE International Conference on Advanced Information Networking and Applications. PP. 27-33. ISBN: 978- 0 -7695- 4018.- 4. DOI: 10.1109/AINA.2010.187. (Apr. 2010).

- Donekena, K. and Gannamani, S., *Performance Evaluation of Cloud Database and Traditional Database*, Sep. 2012.

- Endo, P., Goncalves, G., Kelner, J., and Sadek, D. (2010). *A Survey on OpenSource Computing Solutions*. VIII Workshop em Clouds, Grids e Aplicações.

- Garg, S., Versteeg, S., and Buyya, R. (2013). *A Framework for Ranking of Cloud Computing Services*. Elsevier Future Generation Computer Systems. Vol. 29, Issue 4. PP. 1012-1023. DOI: 10.1016/j.Future.2012.06.006.

- Gawande, M. and Kapse, A. (2014). *Analysis of Data Confidentiality Techniques in Cloud Computing.* International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue 3, PP. 169-175, ISSN: 2320-088X, Mar. 2014.

- Gonzales, N., Miers, C., Redigolo, F., Simplicio, M., and Carvalho, T. (2012). *A quantitive Analysis of Current Security Concerns and Solution for Cloud Computing.* Journal of Cloud Computing Advance, Systems, and Applications - (Open Access). Springer.

- Govind, R. and Mamatha, T., (2013). *Validating the CPU Using Statistics Provided by Linux KVM Hypervisor.* International Journal of Engineering Research & Technology (IJERT), ISSN:2278-0181, Vol. 2, Issue 4, April 2013.

- Goyal, S. (2014). *Public vs Private vs Hybrid vs Community – Cloud Computing: A Critical Review*. International Journal for Computer Networks and Information Security. Vol. 3, Issue 3. PP. 20-29. DOI: 10.5815/ijcnis.2014.03.03.

- Hoffa, C., Mehta, G., Freeman, T., Declman, E., Keahey, K. and Beriman, J. (2008). *On the Use of Cloud Computing for Scientific Workflows*. Computing Research Association's Distributed Research Expenses Program.

IEEE Fourth International Conference on, vol., no., pp. 640,645, 7-12 Dec. 2008, DOI: 10.1109/eScience.2008.167

- Huber, N. and Quast, M. (2013). *Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments*. CLOSER 2011 – Proceeding of the 1st International Conference on Cloud Computing and Services Sciences, Noorwijkerhout, Netherlands, 7-9, May 2011, ISBN: 978-989-8425-52-2, pp. 17-25.

- Hussain, Z., and Gummadi, A. (2013). *Vulnerabilities in Cloud Computing*. George Mason University. INFS 612.

- Hwang, J., Zeng, S., Wu, F., and Wood, T. (2013). *A Component Based Performance Comparison of Four Hypervisors*.IFIP/IEEE International Symposium of Integrated Network Management 27-31 May 2013.

- Jansen, W. and Grance, T. (2011). *Guidelines on Security and Privacy in Public Cloud Computing*. NIST Special Publication 800-144, 2011.

- Karthik, B. and Sri, M. (2013, April). *Cloud Computing Services and Application*. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). Volume 3, Issue 4.

- Kolhe, S. and Dhage, S. (2012). *Comparative Study on Virtual Machine Monitor for Cloud*. World Congress on Information and Communication

Technology. Information and Communication Technologies (WICT), on., vol., no., PP. 425,430, Oct. 30.2012 – Nov 2.2012, DOI: 10.1109/WICT.2012.6409115. IEEE.

- LIorente, I. (2010, November). *OpenNebula: Leading Innovation in Cloud Computing Management*. OW2 Annual Conference 2010, Paris, France.

- Macko, P., Chiarini, M., and Seltzer, M. (2011).*Collecting Provenance via the Xen Hypervisor*. Harvard University.

- Madhavi, K. (2012, September). *Cloud Computing: Security Threats and Counter Measures*. International Journal of Research in Computer and Communication Technology (IJRCCT). Volume 1, Issue 4.

- Makkar, L., and Rajput, G. (2013). *Architecture and Security Functions of Cloud Computing*. International Journal of Computer Sciences and Management Researches. Vol. 2, Issue 1. PP. 1261-1264.

- Mell, P. and Grance, T., (2011). *The NIST Definition of Cloud Computing*. NIST Special Publication 800-145. National Institute of Standards and Technology NIST, Gaithersburg, MD, (September 2011).

- Nagar, N. and Suman, U., (2014). *Architectural Comparison and Implementation of Cloud Tools and Technologies*. International Journal for Future Computer and Communications, Vol 3, No. 3, June, 2014.

- Oracle. (Dec. 2011). *White Paper – Database as a Service Reference Architecture – An Overview*. Redwood Shores, CA 94065, USA.
- Ostermann, S. Losup, A., Yigitbasi, N., Prdan, R., Fahringer, T. and Epeman, D. (2010). *A Performance Analysis of EC2 Cloud Computing fir Scientific Computing*. Lecture Notes of Institute for Computer Sciences, Social Informatics, and Telecommunication Engineering Volume 34,2010 PP. 115,131.

- Park, J. (2012). *A Virtualization Security Framework for Public Cloud Computing*. Computer Sciences and Its Applications, Lecture Notes in Electrical Engineering, Volume 203, 2012, PP. 421,428. Springer.

- Saini S., Heistand, S., Jin, H., Chang, J., Hood, R. Mehrotra, P. and Biswas, R. (2011). *An Application Based Performance Evaluation of NASA's Nebula Cloud Computing Platform*. High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and System (HPCC-ICESS). 2012 IEEE 14th International Conference on, vol., PP. 336,343, 25-27 Jun. 2012, DOI: 10.1109/HPCC.2012.52

- Scarfone, K., Souppaya, M., and Hoffman, P., (2011). *Guide to Security for Full Virtualization Technologies*, NIST Special Publication 800-125.

- Scholosser, D. Duelli, M and Goll, S. (2011). *Performance Comparison of Hardware Virtualization Platforms*. International Federation for Information Processing. (PP 393-405). IFIP.
- Sharma, M. and Sharma, P. (2012). *Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm*. International Journal of Advanced Computer Science and Application. Volume 3, Number 2. (pp. 86-88). IJACSA.

- Shawish, A. and Salama, M., (2014). *Cloud Computing: Paradigm and Technologies*. Inter-Cooperative Collective Intelligence: Techniques and Applications Studies in Computational Intelligence Volume 495, 2014, PP. 39, 67. ISBN: 978-3-642-35015-3.

- Sotomayer, B., Montero, R., Liorente, M., and Foster, I. (2009). *An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Cloud.* IEEE Internet Computing, Special Issue on Cloud Computing.

- Srivstava, A., and Yadav, D. (2012, December). *TaaS: An Evolution of Testing Services Using Cloud Computing*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 1, Issue 10.

- Steimetz, D., Perraut, B., Nordeen, R., Wilson, J. and Wang, X. (2012). *Cloud Computing Performance Benchmarking and Virtual Machine*

*LanuchTime*. SIGITE '12 Proceeding of the 13th Annual Conference on Information Technology Education, PP. 89, 90, Jan. 2012, ISBN: 978-1-4503-1464-0. ACM.

- Tripathi, A., Updhyay, S., and Dwived, S. (2012, May). *Scalability Performance of Software Testing by Using Review Technique*. International Journal of Scientific & Engineering Research. Volume 3, Issue 5.
- Wyld, D. (2010). *The Cloudy Future of Government IT: Cloud Computing and Public Sector Around The World*. International Journal of Web & Semantic Technology (IJWEST). Vol. 1, Issue 1.

- Xiao, Z., and Xiao, Y. (2013). *Security and Privacy in Cloud Computing*. IEEE Communications Surveys & Tutorials, Vol. 15, Issue. 2, Second Quarter 2013. PP. 843-859.

- Xing, Y., and Zhen, Y. (2012). *Virtualization and Cloud Computing. Future Wireless Network and Information Systems*. Lecture Notes in Electrical Engineering Volume 143, 2012, PP. 305, 312.Springer.

- Xu, X., Zhou, F., and Jiang, J.(2008). *Quantifying Performance Proporties of Virtual Machine*. ISISE '08 International Symposium on Information Science and Engineering, vol. 1, no., PP. 24,28, 20-22 Dec. 2008. DOI: 10.1109/ISISE.2008.221. IEEE.

- Yang, C., Chen, B. and Chen, W. (2011*). On Implementation of KVM IaaS with Monitoring System on Cloud Environments*. National Science Council, Taiwan. (pp 300-309). Communication and Networking Communication in Computer and Information Science Volume 265, 2012, PP.300, 309. Springer.

- Zhang, Q. and Cheng, L., (2010). *Cloud Computing: State of the Art and Research Challenges*. Journal of Internet Services And Applications, May 2010, Volume 1, Issue 1, PP. 7, 18. Springer.

## Online References


- Christoffer, H. "Measuring SQL Performance", SQLserverCentral.com. N.p., n.d. web.

  ***Date Accessed 9 Mar. 2014.***

- "OpenStack Architecture.", Document ATOM, N.p., n.d Web.

  ***Date Accessed 14 Feb. 2014.***

- "Cloud Management Platforms." Cloud Computing Competence Center for Security RSS, N.p., n.d Web.

  ***Date Accessed 1 Mar. 2014.***

- Pepple, K., OpenStack Nova Architecture. Retrieved from http://ken.pepple.info/openstack/2011/04/22/openstack-nova-architecture.

  ***Date Accessed 11 Mar. 2014.***

- "OpenNebula Architecture", Document OpenNebula Team, archives.opennebula.org. N.p., n.d. Web.

  ***Date Accessed: 15 Mar. 2014.***

- "SET STATISTICS TIME", Microsoft SQL Server. N.p., n.d. Web.

  ***Date Accessed: 22 Mar. 2014.***

- Younge, A., Henschel, R., Brown, J., Laszwski, G., Qiu,J, and Fox, G. (2011). Analysis of Virtualization Technology for High Performance Computing Environment. National Science Foundation (NSF).
- Chapple, M. (2001). *Introduction to Microsoft SQL Server 2000*. Online Article.
- Zeltkevic, M. (1998). *Variance, Standard Deviation and Coefficient of Variation*. Retrieved from: http://web.mit.edu/10.001/Web/Course-_Notes/Statistics_Notes/Visualization/node4.html