جامعة الشرق الأوسط
MIDDLE EAST UNIVERSITY

Scaling cloud resources under different virtualization hypervisors types with regards to machine utilization

توسيع موارد السحابة باستخدام انواع مختلفة الادوات الافتراضية بقياس فعالية هذه الموارد

Prepared by:

Sultan Ali Abu Sharkh

Supervised by

Dr. Ahmad kayed

Master Thesis

Submitted in Partial Fulfilment of the Requirements for the Master Degree in Computer Information Systems

Department of Computer Information Systems

Faculty of Information Technology

Middle East University

Amman – Jordan

August 2014

اقرار التفويض

انـي سلطان علي ابو شرخ أفوض جامعة الشرق الاوسط بتزويد نسخ من رسالتي للمكتبات او المؤسسات او الهيئات او الافراد عند طلبها

الاسم : سلطان علي ابو شرخ

التاريخ : 17-8-2014

التوقيع

Authorization

I, Sultan Abu sharkh, authorize the **Middle East University** to provide libraries, organizations and even individuals with copies of my thesis upon request.

- **Name:** Sultan Abu sharkh

- **Signature:** ____________

- **Date: 17-8-2014**

Examination Committee Decision

This is to certify that the thesis entitled "**Scaling cloud resources under different virtualization hypervisors types with regards to machine utilization**" was successfully defended and approved on August 17, 2014.

Examination Committee Members **Signature**

Dr. Ahmad Kayed
Associate Professor
Dean Faculty of Information Technology
Middle East University

……………………………

Dr. Hebah H.O Nassereddin
Associate Professor
Faculty of Information Technology
Middle East University

……………………………

Dr. Ali Asad Al-Dahoud
Professor
Dean Faculty of Information Technology
Al-Zaytoonah Private University of Jordan

……………………………

Acknowledgments

I would like to express my deep gratitude and appreciation to my thesis supervisor Dr. Ahmad Kayed for his support, encouragement, proofreading of the thesis drafts, and for helping me throughout my study, putting me in the right step of scientific research. I would like to thank the Information Technology Faculty members at the Middle East University, and the staff of MEU for their collaboration and support especially Dr. Heba Nasereddin, I would also like to thank all of my family members specially my father who find my way to best life and my soul my mother and my brother Sofyan Abu Sharkh.

I will not forget my colleagues whom support me to finish this thesis, i would like to give a special thanks to my brother in my work in computer center of alzaytoonah university, best team in the world, Thaer Abu Sharar, Soufian Farhan, Jad Yasin, Rashad Al-banna, Saleem Ankeer, Tamer yadak, Ismael Odeh, Odai Slama, Mohammed Bdair, And all those Whom support me to, finish this thesis.

Dedication

{ قل إن صلاتي ونسكي ومحياي ومماتي لله رب العالمين لا شريك له وبذلك أمرت

وأنا أول المسلمين } 17.سورة الانعام

All praise is for Allah, the exalted, for his favors that cannot be counted, to my father who taught me how to find my way through man's most challenging hardships, to my mother who supported me during all steps of the way, to my brother my godfather Dr. Ahmad Kayed, to my brothers and sisters with whom I shared with the life, to the one who guided me toward success in my academic field Dr. Hebah Nasereddin.

Table of Contents

List of TABLE

List of Figures

Abbreviations

CC: cloud computing.

KVM: kernel based virtual machine.

BSD: Berkeley Software Distribution.

LAMP: Linux apache MySQL PHP.

DBMS: database management system.

AWS: AMAZON web services

Opstk: openstack.

Opnbu: opennebula.

توسيع موارد السحابة باستخدام انواع مختلفة الادوات الافتراضية بقياس فعالية هذه الموارد

الطالب

سلطان علي ابوشرخ – 401210058

المشرف : الدكتور أحمد الكايد

الملخص

لقد تم في هذا البحث دراسة منهجية لحوسبة السحابة واثر توسيع موارد السحابة مع الاجهزة الافتراضية المستخدمة بها , حيث ان الحوسبة السحابية احد اهم المواضيع الحديثة التي تدور حولها كثير من الدراسات ولاهميتها البالغة في تصميم معمارات الأنظمة الحديثة والشبكات.

يستطيع مستخدم الاجهزة الخاضعة لنظام الحوسبة السحابية الاستفادة من مميزات السحابة ومنها سهولة الوصول والمعالجة عن طريق الشبكة وبالامكان ايضا عمل توسيع لموارد هذه السحابةوزيادة مواصفاتها على حسب احتياجات المستخدم اما بتوسيع موارد الجهاز السحابي نفسه ويسمى التوسيع العامودي او ترقيته عن طريق ربط مجموعة اجهزة به ويسمى التوزيع الافقي ,يعتمد انشاء وحدات حوسبة سحابية بشكل رئيسي على الاجهزة الافتراضية وهناك اكثر من اسلوب لانشاء هذه الاجهزة الافتراضية تختلف عن بعضها من حيث النواة المكونة لهذه الاساليب.

لقد تم في هذا البحث طرح توصيات مقترحة لاساليب لتوسيع موارد السحابة مع كل نوع من طرق انشاء الاجهزة الافتراضيةمبنية على اساس مقارنة استهلاك هذه الموارد باستخدام نظام قواعد البيانات ودراسة اثرها على المعالج الرئيسي والذاكرة العشوائية والوقت المستهلك لتنفيذ العملية.

تمت التجربة باستخدام احدث البيئات والانظمة في مجال الحوسبة السحابية اوبن ستاك ,اوبن نبيولا ومقارنة استهلاك هذه الموارد باستخدام نظام قواعد البيانات ودراسة اثرها على المعالج الرئيسي والذاكرة العشوائية والوقت المستهلك لتنفيذ العملية, تمت هذه التجربة باستخدام احدث البيئات والانظمة في مجال الحوسبة السحابية اوبن ستاك و اوبن نيبيولا.

تم الاخذ بمجموعة معايير اثناء القيام بهذه التجارب مثل استخدام الواجهة الرسومية للنظام, استخدام الواجهة الخطية, وتفعيل الوضع الثنائي للذاكرة العشوائية, واداء المعالجة باستخدام المعالج المساعد للرسوميات, تغيير معمارية تصميم التوسيع الافقي ودراسة اثر هذه المعايير على التجارب وموارد السحابة الرئيسة مثل المعالج والرئيسي واستهلاك الذاكرة العشوائية والوقت المستهلك.

Scaling Cloud Resources under Different Virtualization Hypervisors Types with Regards to Machine Utilization

Student Name: Sultan Ali Abu Sharkh
Student Number: 401210058

Supervisor
Dr. Ahmad Kayed

Abstract

Cloud computer is very common and important nowadays, it introduce many services and advantages, user can simply use storage, computing power, or specially development environments, without having to worry how these work internally, Cloud computing is a systems architecture model for Internet-based computing.

It is the development and use of computer technology on the Internet, cloud based mainly on virtualization technology and it uses many techniques to achieve it, User can upgrade the component such that provided by cloud environment by scaling the machine it's called vertical scalability or to split the machine to many machines called horizontal scalability.

This research will find which scalable type has a better performance with certain virtual type, the challenge here is how the research can apply that in the domain of database, and we will introduce results and recommendations

that which type of scalability more suitable with particular virtualization technique

The study include the effects of types of hypervisors that fit with scalability technique with regards to machine utilizations of CPU, RAM and time consuming, also it shows that the internal effects of scale the system with other criteria's such as use standard graphical mode, use command mode, use SSH remote server, RAM with dual channels, use graphical processor unit GPU processor, and change horizontal scale architecture.

CHAPTER ONE

Introduction

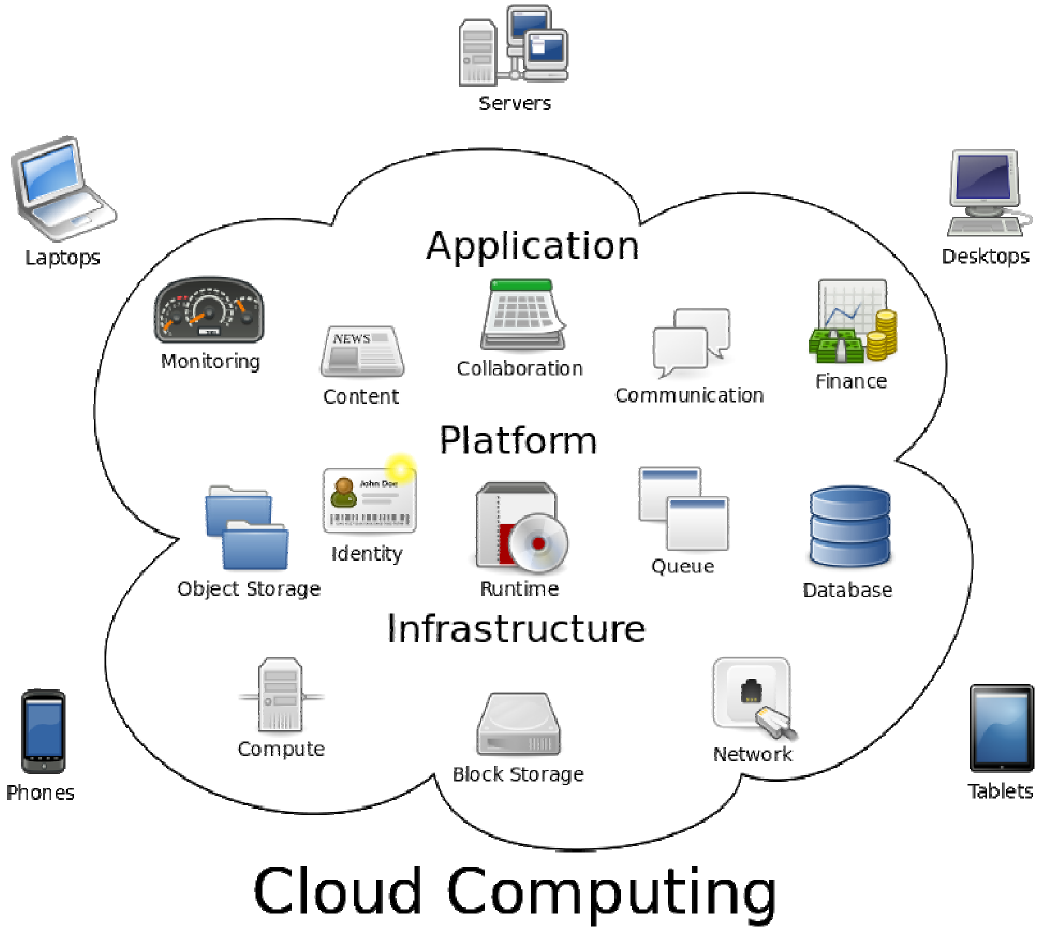
1.1 Overview

In this chapter, researcher will talk about the scope of the thesis; Cloud Computing, Hypervisors types in the Cloud, Machine Scalability Depend Utilization. Then an idea about the research problem is given and how it has been addressed, thesis questions.

1.2 Cloud Computing

In recent years, the term cloud computing has been used to identify an evolution paradigm in the computer industry. That is because of set of advanced technologies that affect the focus of the organizations and businesses on the cost. The base of the cloud computing is the evolution of three properties, virtualization, grid computing and web services. The increasing of Internet connection, mobile accessibility and portable devices has encouraged the spread of applications created for this environment and the access to available resources exclusively through the internet (Ercolani, G. 2013).

There are many definitions mentioned in papers and books, the standard definition of (CC) from NIST (National Institute of Standards and Technology) defines the cloud computing:" 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. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models" (Mell, P. et.al 2011)



Fig

ure1-1: Cloud Computing Structure

Essential characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. Three service models Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Four deployment models are private cloud, community cloud, public cloud and hybrid cloud.

As mentioned before that the base of the cloud computing is the evolution of three properties; virtualization, grid computing, and web services. Virtualization is one of the most important element that make the cloud, it helps organizations enabling much greater consolidation within private data centers, and more recently as a driving technology behind cloud computing. See figure 1-1

1.3 Cloud Services Model

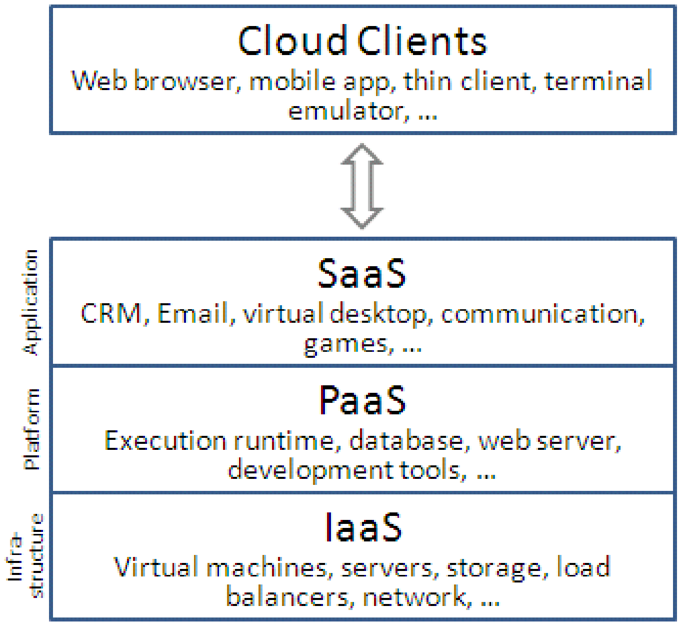

Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
Application
SaaS
CRM, Email, virtual desktop, communication, games, ...
Platform
PaaS
Execution runtime, database, web server, development tools, ...
Infra-structure
IaaS
Virtual machines, servers, storage, load balancers, network, ...

Figure1-2: Cloud Computing Models

Infrastructure as a Service (IaaS)

In the most basic cloud-service model, providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as OpenStack, XEN, Oracle VirtualBox, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. See figure 1-2

Platform as a Service (PAAS)

In the paas models, cloud providers deliver a computing platform, typically including operating system, progRAMming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some paas offers like MICROSOFT AZURE and GOOGLE, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments. See figure 1-2

Software as a Service (SAAS)

In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis. SaaS providers generally price applications using a subscription fee.See figure 1-2

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. Cloud applications are different from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand.

1.4 VIRTUALIZATION

In virtualization architecture, there are two different approaches: Type one Para-virtualization (PV), and Type two Full-virtualization (FV). Para-virtualization requires modification to the guest OS, essentially teaching the OS how to make requests to the hypervisor when it needs access to restricted resources. Full-Virtualization is designed to provide a complete simulation of the underlying physical system and creates a complete virtual system in which the guest operating systems can execute. No modification is required in the guest OS or application. This approach can be preferred because it enables complete decoupling of the software from the hardware.

Many kind of virtual platforms differ from open-source as KVM and Xen (that we used in this research) to commercial platform like VMware vSphere and Microsoft Hyper-V, their goal is one to manage the guest OS, but they differ in underlying technologies (Hwang J. et at.2013).

In this research we used the most two famous kinds of hypervisors, Xen and KVM. Xen[1] is a very famous Para-virtualization solution, originally developed at the University of Cambridge. It's the only open source with bare-metal solution, using in many cloud providers like Amazon EC2. It consists of several components that work together to deliver the virtualization environment including Xen Hypervisor, Domain 0 Guest (referred as Dom0) which represent the layer zero or hardware, and Domain U Guest (Fayyad-Kazan et al.2013).

[1] http://www.xenproject.org/

Responsibilities of the hypervisor include memory management and CPU scheduling of all virtual machines ("domains"), and for launching the most privileged domain ("dom0") - the only virtual machine which by default has direct access to hardware. From the dom0 the hypervisor can be managed and unprivileged domains ("domU") can be launched.

The dom0 domain is typically a version of Linux, or BSD (Berkeley Software Distribution Unix). User domains may either be traditional operating systems, such as Microsoft Windows under which privileged instructions are provided by hardware virtualization instructions (if the host processor supports x86 virtualization, e.g., Intel VT-x and AMD-V), or Para-virtualized operating system whereby the operating system is aware that it is running inside a virtual machine, and so makes hyper calls directly, rather than issuing privileged instructions.

KVM[2] (Kernel-based Virtual Machine) is very famous Full-virtualization solution. Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. KVM runs as a kernel module like (kvm-intel.ko/ kvm-amd.ko) and (kvm.ko) that's provides the core virtualization infrastructure, which means it uses most of the features of the Linux kernel operating system itself. For example, rather than providing its own CPU scheduler for VMs, KVM treats each VM as a process and uses the default Linux scheduler to allocate resources to them. QEMU (Quick Emulator) is added also to KVM for emulating input and output.

[2] http://www.linux-kvm.org/page/Main_Page

OpenStack[3] is a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for public and private clouds. OpenStack project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution.

This research use OpenStack and opennebula because most people use it and it's well known. Many companies produce open source cloud computing software for creating, managing, and deploying infrastructure cloud services like CloudStack, but in OpenStack project more than 200 companies have joined the project, including Arista Networks, AT&T, AMD, Canonical, Cisco, Dell, EMC, Ericsson, Go Daddy, Hewlett-Packard, IBM, Intel, NEC, NetApp, Nexenta, Red Hat, SUSE Linux, Mellanox, VMware, Oracle and Yahoo.

Many progRAMs and applications are used to set the cloud environment. Debian GNU/Linux[4] is frequently used as operating system for cloud servers; Debian GNU/Linux is a particular distribution of the Linux operating system, and numerous packages that run on it. Debian is open source and free to use has many features make it one of the best server operating system.

[3] https://www.openstack.org/
[4] https://www.debian.org/

Proxmox VE[5] is a complete open source virtualization management solution for servers. It is based on KVM virtualization and container-based virtualization and manages virtual machines, storage, virtualized networks, and high availability Clustering.

1.5 Hypervisors

A hypervisor is one of many virtualization techniques which allow multiple operating systems (guests) to run concurrently on a host computer. The hypervisor presents to the guest operating systems a virtual operating platform and monitors the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources. Hypervisor is installed on server hardware whose only task is to run guest operating systems (Sabahi, F. 2012).

1.6 Cloud Computing FRAMework

1- OpenStack

OpenStack[6] is a free and open-source software cloud computing platform. It is primarily deployed as an infrastructure as a service (IaaS) solution. The technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a data center, able to be managed or provisioned through a web-based dashboard, command-line tools, or aRESTful API. It is released under the terms of the Apache License.

[5] https://www.proxmox.com/

[6] https://www.openstack.com/

The OpenStack community collaborates around a six-month, time-based release cycle with frequent development milestones. During the planning phase of each release, the community gathers for the OpenStack Design Summit to facilitate developer working-sessions and to assemble plans

2- OpenNebula

OpenNebula[7] is a cloud computing toolkit for managing heterogeneous distributed data center infrastructures. The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. OpenNebula is free and open-source software, subject to the requirements of the Apache License.

toolkit includes features for integration, management, scalability, security and accounting. It also claims standardization,interoperability and portability, providing cloud users and administrators with a choice of several cloud interfaces (Amazon EC2 Query, OGF Open Cloud Computing Interface andvCloud) and hypervisors (XEN, KVM and VMware), and can accommodate multiple hardware and software combinations in a data center.

3- CloudStack

[7] https://www.opennebula.com/

CloudStack[8] is open source cloud computing software for creating, managing, and deploying infrastructure cloud services. It uses existing hypervisors such as KVM, vSphere, and XENServer/XCP for virtualization. In addition to its own API, CloudStack also supports the Amazon Web Services (AWS) API.and the Open Cloud Computing Interface from the Open Grid Forum.

1.7 Scalability

Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources (typically hardware) are added. An analogous meaning is implied when the word is used in an economic context, where scalability of a company implies that the underlying business model offers the potential for economic within the company. (Carol Guercio 2008)

Scale horizontally (or scale out) means to add more nodes to a system, such as adding a new computer to a distributed software application. An example might be scaling out from one Web server system to three.

[8] https://www.cloudstack.apache.com/

Scale vertically (or scale up) means to add resources to a single node in a system, typically involving the addition of CPUs or memory to a single computer. Such vertical scaling of existing systems also enables them to use virtualization technology more effectively, as it provides more resources for the hosted set of operating system and application modules to share.

1.8 Problem Statement

Nowadays many enterprises use cloud for its feature such as scalability. many enterprises need to improve and scale the machine that host Database Engine physically using upgrade of the hardware component of the machine (vertical scalability) , or to add external machines that associated with the main machine as a one unit (horizontal Scalability).

Several types of scalability are available such as vertical and horizontal scalability. At the same time several virtualization types are available such as type1 and type2. The problem here is how to find which scalable type has a better performance with certain virtual type. The challenge here is how we can apply that in the domain of database.

This research will answer the following questions:

1-How to measure the utilization of the machine in term of CPU and memory in the case of deploying database operations for both types of scalability under different hypervisors types?

2- How to select the suitable hypervisor that fit certain scalability type depends on CPU and memory utilization?

3- Is there a correlation between database operations, hypervisor, scalability, and machine utilization?

1.9 Methodology

Cloud environments use different type of hypervisors to create virtual machines in order to distribute operating system guests for its users. These OS guests represented as virtual machines has different virtual components depend on main components of physical machine itself that host the virtual environment.

In this research, we will determine and fine best way to implement hypervisor environment with each Scalability type.

This research will carry out a case study which covers building cloud environment using two types of cloud platforms, and assigning two types of hypervisors to this cloud in and scale each virtual machine with two types of scalability to monitor the benchmark during execution many operation on database engine, and make fair comparison in different cloud platforms depend on scalability type with regards to machine utilization.

CHAPTER TWO

Background and Literature Review

2.1 Introduction

In this chapter researcher will provide the literature review. The most related studies in the field of scale machine under cloud environment with use both KVM and XEN hypervisor and will discuss background of the field cloud and scalability.

2.2 Background

Many Resources want to be scaled, cloud environment such as openstack or opennebula provide virtual machine with resources depend mainly on virtualization KVM or XEN ability of implementation scalability will be tested on both types to find which one suit each other using MYSQL database as benchmark of the performance .

Selected Cloud environment openstack and opennebula are huge open source projects provide fully private cloud with all functions, and researcher can create instances from the cloud to deploy it as researcher need.

2.3 Open Stack

The OpenStack project mission is to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable. ProgRAMs can create any code repository and produce any deliverable they deem necessary to achieve their goals.

2.4 Open Nebula

OpenNebula is a cloud computing toolkit for managing heterogeneous distributed data center infrastructures. The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. OpenNebula is free and open-source software, subject to the requirements of the Apache License version 2.

2.5 MYSQL

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack, LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

2.6 Literature Review

- (RalucaPopa ET al.2011) they are group at Massachusetts Institute of Technology, introduce very important project which CryptDB, CryptDB is an open source system that provides practical and provable confidentiality in the face of attacks for applications backed by SQL databases. It works by executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes.

-(Yanbin Lu et al.2011) They present approach for coping with privacy concerns. There scheme prevents the cloud server from learning any possibly sensitive plaintext in the outsourced databases. It also allows the database owner to delegate users to conducting content-level fine-grained private search and decryption. In addition, the scheme supports private querying whereby neither the database owner nor the cloud server learns query details.

- (Hasan Kazan et al.2013) They made comparison between Full and Para Virtualization among the XEN Hypervisor, they proved a theoretical state “PV delivers higher performance than full virtualization because the operating system and hypervisor work together more efficiently, without the overhead imposed by the emulation of the system's resources” by using the experimental tests with result to achieve this state.

- (**JinhoHwanget al.2013**) they made comparison between four popular virtualization platforms, Hyper-V, KVM, vSphere and XEN. They use many bench mark tools like Bytemark, RAMspeed, Bonnie++ &FileBench, Netperf, Application Workloads, and Multi-Tenant Interference, they found that "there is no perfect hypervisor that is always the best choice; different applications will benefit from different hypervisors depending on their performance needs and the precise features they require".

-(SnehaKolhe et al. 2012) They tried to make a comparative analysis of KVM and XEN depending on various benchmarking tools. This study concentrated on measuring CPU performance, network speed, and disk access using a secure shell connection (SSH), and applying benchmark tools for finding results.

-(Chao-Tung Yang et al. 2011) They proposed a way to build KVM environment in the cloud systems and operation. This study focus 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.

-(ArshdeepBahga et al. 2013) They proposed 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.

-(Divyakant Agrawal et al. 2011) present an overview of working on instilling "cloud features" in a database system designed to support a variety of applications deployed in the cloud: designing scalable database management architectures using the concepts of data fission and data fusion, enabling lightweight elasticity using low cost live database migration, and designing intelligent and autonomic controllers for system management without human intervention.

-(Zhou Weiet al. 2011) implement approach on top of the two main families of scalable data layers: Bigtable and SimpleDB. Performance evaluation on top of HBase (an open-source version of Bigtable) in our local cluster and Amazon SimpleDB in the Amazon cloud shows that our system scales linearly at least up to 40 nodes in our local cluster and 80 nodes in the Amazon cloud.

-(**Salve Bhagyashriet al. 2013**) present two system models and demonstrate their scalability under the scale-out paradigm of Cloud com-putting platforms. In the first system model, all transaction management functions are executed in a fully decentralized manner by the application processes. The second model is based on a hybrid approach in which the conflict detection techniques are implemented by a dedicated service.

-(XianghuaXu et al. 2008) They tried to measure the performance of virtual machines running in the cloud from isolation and scalability point of view. They propose test comparison using benchmark such as CPU, memory, and disk intensive. This study focus on comparing leading hypervisors such as Kernel Virtual Machine (KVM), XEN, and VMware.

-(A Bino et al. 2011) They tried to study hypervisors such as KVM, XEN and its effects on network performance in cloud environments, by comparing full virtualizations approach with para-virtualization 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.

(SnehaKolhe et al. 2012) They tried to make a comparative analysis of KVM and XEN depending on various benchmarking tools. This study concentrated on measuring CPU performance, network speed, and disk access using a secure shell connection (SSH), and applying benchmark tools for finding results.

(Daniel Schlosser et al. 2011) They proposed a novel study to find how isolation techniques have impacts on the performance of guest systems. They study 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.

CHAPTER THREE

Design the Experiment

3.1 introduction

In this chapter researcher will describes how the experiment has been implemented. The proposed approach describe the way how will researcher answer the questions as mentioned at chapter one.

3.2 Prepare environment

In order to run the experiment, researcher installed Linux Ubuntu 12.12 LTS as host operating system which will contain the virtual machine hypervisors XEN and KVM.

There are many prerequisite components should be installed on host operating system to deploy virtual machine environment and cloud toolkit framework, such as build essentials MYSQL Server engine and general runtime platforms software. Then researcher install two different Cloud frameworks OpenStack and OpenNebula to start using the cloud features and make desired experiment. The benchmark will be MYSQL Server standard database called employee.

Servers' Description

Our cloud environment was built on servers; these server have certain specifications that are listed below with important software's installed:

- Each server has Ubutnu GNU/Linux as operating system.
- An OpenStack was used as a tool for building and managing cloud computing platforms in first private cloud.
- An OpenNebulawas used as a tool for building and managing cloud computing platforms in Second private cloud.
- Important components and libraries installed on the server to deploy and the cloud environment correctly such as Apache web server , python development run time and C++ libraries
- Glances is a software to monitor your GNU/Linux or operating system outside or inside the cloud from a text interface. Glances uses the library libstatgrab to retrieve information from your system and it is developed in Python.
- MYSQL Server, its database management system that installed on the server to handle schemas and relation.
- PHPMyAdmin is a software tool written in PHP, intended to handle the administration of MySQL over the Web.
- KVM with QEMU was used to prepare the KVM hypervisor, XENServer is based on the XENProject hypervisor.

- Horizons, it’s a dashboard provide web based interface responsible to manage cloud instances, features and settings for OpenStack and another dashboard for OpenNebula.

XEN boots from a boot loaderof Ubuntu GNU/Linux, and then usually loads a para-virtualized host operating system into the host domain (dom0) layer zero.

For each server, a single virtual machine (VM) was created.

3.3 Environment Description:-

Basic Machine: CPU 1 core – RAM 1 GB

Scaled Machine (Vertical): CPU 8 cores – RAM 8 GB

Scaled Machine (Horizontal) 4 physical Machine: CPU Dual Core – RAM 2 GB

Host Operating System: Linux Ubuntu Guest Operating System: Linux Ubuntu

Hypervisor: KVM– XEN

Cloud Environment: OpenStack, OpenNebula

Benchmark: Employees Database with MYSQL DBMS

3.4 Database benchmark:-

Researcher use standard database for benchmark from MYSQL itself called Employees database was developed by Patrick Crews and Giuseppe Maxia and provides a combination of a large base of data (approximately 160MB) spread over six separate tables and consisting of 4 million records in total. The structure is compatible with a wide range of storage engine types. Through an included data file, support for partitioned tables is also provided. See Figure3-1

3.5 Database Structure:-

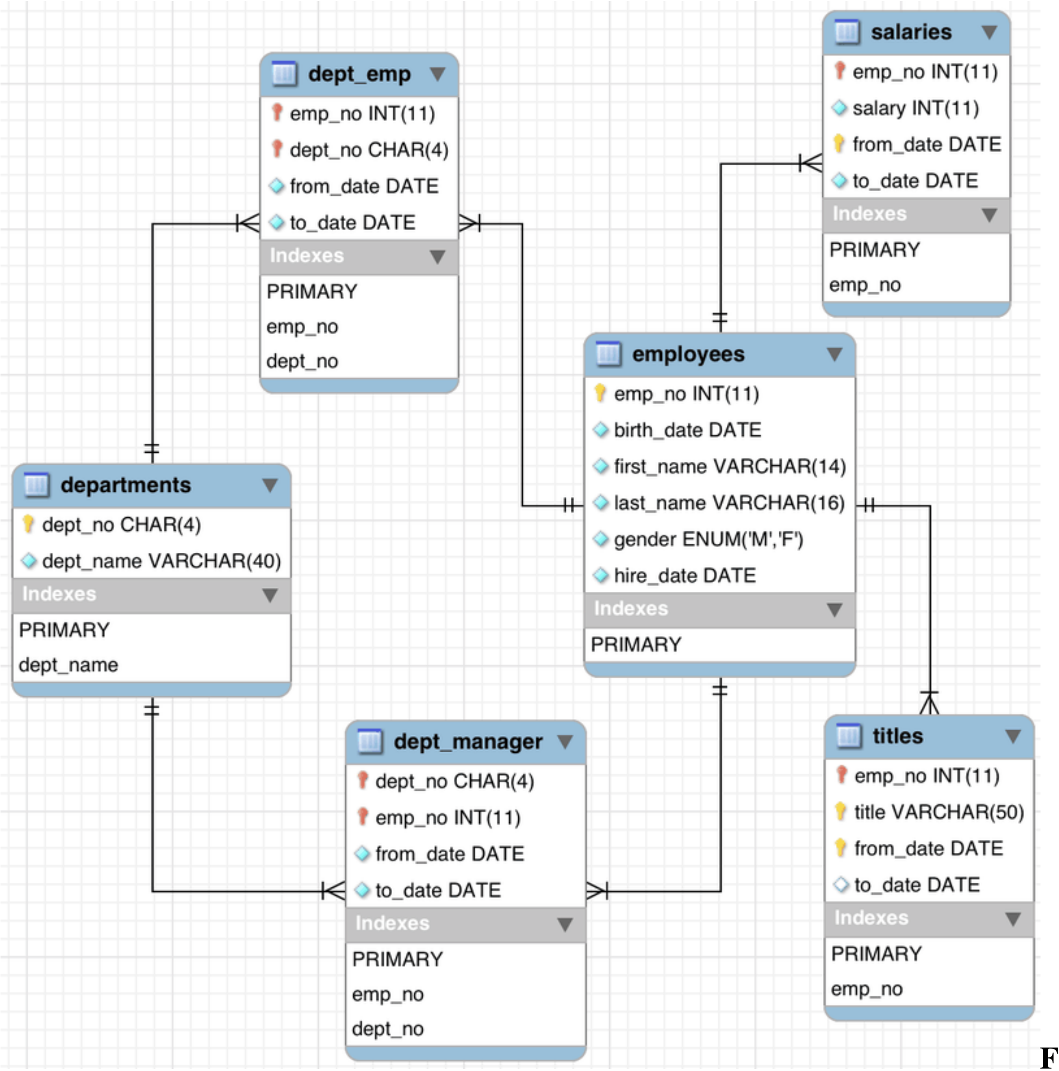


Figure3-
1: ER DiagRAM of Employees

3.6 Proposed Approach:

There are a Steps for the experiment will addressed in this research as the following below: See Figure 3-2

1- Run the cloud environment OpenStack or OpenNebula to start create cloud machine (instances).

2- Create instances with two type of virtualization hypervisors first time as KVM and second as XEN and prepare the cloud machine by install main software's as researcher mentioned before.

3- Deploy Database Management System (DBMS) in the cloud machine before scale the machine.

4- Scale cloud machine with two different type horizontal or vertical.

5- Implement database queries and operation and check the performance before and after scaling techniques.

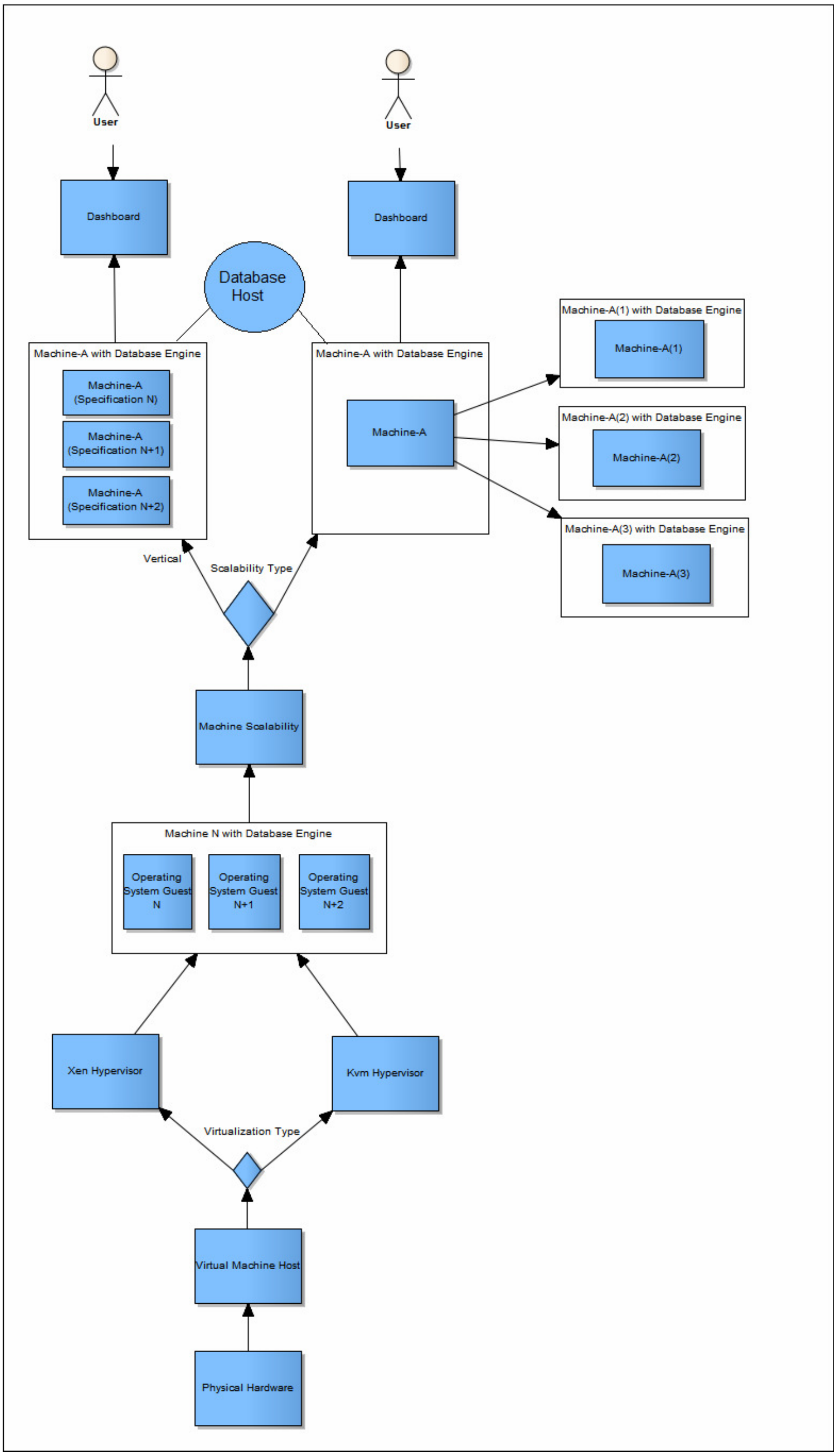
6- Change criteria's of the experiment.

7- Evaluation Benchmark depends on performance and criteria's of the components of the machine.

8- Compare Benchmark results for each hypervisor depend on scalability type in two cloud environment.

3.7 Proposed Model: -

Figure 3-2 Methodology Structure


User
User
Dashboard
Dashboard
Database Host
Machine-A with Database Engine
Machine-A (Specification N)
Machine-A (Specification N+1)
Machine-A (Specification N+2)
Machine-A with Database Engine
Machine-A
Machine-A(1) with Database Engine
Machine-A(1)
Machine-A(2) with Database Engine
Machine-A(2)
Machine-A(3) with Database Engine
Machine-A(3)
Vertical
Scalability Type
Machine Scalability
Machine N with Database Engine
Operating System Guest N
Operating System Guest N+1
Operating System Guest N+2
Xen Hypervisor
Kvm Hypervisor
Virtualization Type
Virtual Machine Host
Physical Hardware

3.8 Implementation Steps: -

Researcher will discuss the practical steps and how to interact with cloud with operating system and database, its describes with layers main layer consist of host operating system

And other layer included inside the cloud boundaries. See Figure 3-3

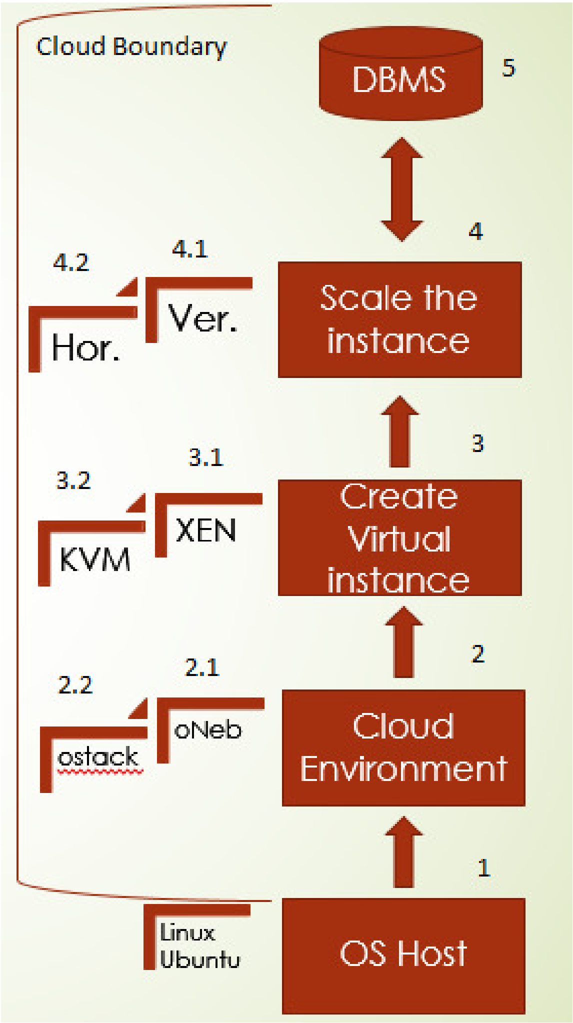


Figure 3-3 Design layers

1- First layer contain operating system as host, its contain kernel to deal with the hardware of the physical machine, researcher recommend to use Ubuntu server with some feature and tools for cloud management software such as JUJU software, this software facilitate to manage cloud nodes in the server and convert nodes and relation into graphical image with arrows to make management easy. See Figure3-3

2- Second layer contains cloud boundary, install cloud environment with two common types (See Figure3-3: 2.1 openNebula - 2.2 openStack) it's responsible to create cloud instance, management network traffic, routing IP address of the instance, maintain stability of the performance and durability of the cloud instances.

3- Creation the instance inside the cloud depend mainly of virtualization technology, so researcher use fist type KVM hypervisor, its provide full simulation of the physical utilization as CPU, RAM, HDD and network interface, and second type XEN, its provide microkernel version that can deal with physical hardware and manage CPU jobs, memory of the virtual instance. (See Figure3-3: 3.1 XEN - 3.2 KVM)

4- Scaling the instance components will be implemented with two types of scalability horizontal or vertical, in vertical technique the instance will be upgraded by add a new component such as adding CPU and RAM (virtually inside the cloud) but in horizontal technique the instance will be upgraded by adding new instances relating with main instance as a one unit, both scalability techniques will be implemented with two types of hypervisor KVM and XEN under two types of cloud environment openStack and openNebula. (See Figure3-3: 4.1 Vertical - 4.2 Horizontal)

5- Database management system will has the engine of the database to use it as benchmark MYSQL, all instances that created and scaled will run 8000 queries (insert queries) of database benchmark called employees to study the effects of the benchmark to machine utilization and the relation between them according to the type of the scalability and virtualization hypervisors technique. (See Figure3-3: 4 DBMS)

CHAPTER FOUR

Results Analysis

4.1 Introduction

In this chapter we will talking about the results of the experiment and make the analysis and compare it to find the mathematical relations, standard deviation, correlation and average between results and make answers that which type of scalability fit specific type of virtualization XEN and KVM under different type of cloud environment Openstack and Opennebula.

4.2 Experiment criteria

Test the experiment with MYSQL database employee benchmark with changing with criteria's of some feature of the operating system or physical design of the server such as:

1- Test the Experiment with standard graphical mode operating system under cloud Environment.

2- Use Command Line mode after switch off lightdm component (graphical mode) operating system under cloud Environment.

3- Enable Multi Thread Core under graphical mode & cloud Environment.

4- Use Remote Access Using SSH Server under cloud Environment.

5- Use another Hardware architecture (RAM with Dual Channel Mode) under graphical mode & cloud Environment.

6- Enable GPU processing (ATI STREAM Technology) under graphical mode & cloud Environment.

7- Test the Experiment with Changing Horizontal Scalability Design

-Scaled Machine (Horizontal) 2 physical Machine: CPU Quad Core – RAM 4 GB

4.3 Result of the Experiment

4.3.1 OpenStack Experiment

Result of implementing Scalability with **Openstack** with two types KVM and XEN with different criteria's. - Test the Experiment with standard graphical mode

First Test:-

	Scalability Type	KVM	XEN
CPU	Basic Machine	92% usage	94% usage
	Vertical Scalability	58% usage	65% usage
	Horizontal Scalability	89% usage	92% usage
RAM	Basic Machine	920MB	960MB
	Vertical Scalability	4.7GB	5.1GB
	Horizontal Scalability	1.8 GB	1.9 GB
TIME	Basic Machine	47 Minutes	50 Minutes
	Vertical Scalability	16 Minutes	15 Minutes
	Horizontal Scalability	29 Minutes	31 Minutes

Table 4-1: OpenStack – First Experiment (a)

Second Test:-

,	Scalability Type	KVM	XEN
CPU	Basic Machine	95% usage	97% usage
	Vertical Scalability	60% usage	65% usage
	Horizontal Scalability	90% usage	92% usage
RAM	Basic Machine	935MB	970MB
	Vertical Scalability	4.7GB	5.1GB
	Horizontal Scalability	1.8 GB	1.9 GB
TIME	Basic Machine	48 Minutes	52 Minutes
	Vertical Scalability	16 Minutes	16 Minutes
	Horizontal Scalability	30 Minutes	30 Minutes

Table 4-1: OpenStack – First Experiment (b)

Average Calculation depend on these equations:-

CPU usage, RAM and time in all cases for basic machine, Vertical and Horizontal Scalability for both XEN and KVM

Eq1) CPU Machine Average=$\frac{CPU\ Usage\ FT\ +CPU\ Usage\ ST}{2}$.

Eq2) RAM Machine Average=$\frac{RAM\ Usage\ FT\ +RAM\ Usage\ ST}{2}$.

Eq3) Time Machine Average=$\frac{Time\ Usage\ FT\ +Time\ Usage\ ST}{2}$.

	Scalability Type	KVM	XEN
CPU	Basic Machine	93.5% usage	95.5% usage
	Vertical Scalability	59% usage	65% usage
	Horizontal Scalability	89.5% usage	92% usage
RAM	Basic Machine	925.5MB	965MB
	Vertical Scalability	4.7GB	5.1GB
	Horizontal Scalability	1.8 GB	1.9 GB
TIME	Basic Machine	47.5 Minutes	51 Minutes
	Vertical Scalability	16 Minutes	15.5 Minutes
	Horizontal Scalability	29.5 Minutes	30.5 Minutes

Table 4-2: OpenStack – Average of First Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

(-) minus means advantage for KVM in all experiment.

(+) positive means advantage XEN in all experiment.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-6%	-2.5%
RAM	-0.4 GB	- 0.1GB
TIME	0.5min	-1min

Table 4-3: OpenStack – Difference of First Experiment

-Test the Experiment with Command Line mode after switch off lightdm component (graphical mode) operating system under cloud Environment.

First Test:-

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	72% usage	75% usage
	Vertical Scalability	54% usage	58% usage
	Horizontal Scalability	69% usage	78% usage
RAM	Basic Machine	790MB	810MB
	Vertical Scalability	3.4GB	3.7GB
	Horizontal Scalability	1.4 GB	1.6 GB
TIME	Basic Machine	43 Minutes	45 Minutes
	Vertical Scalability	13 Minutes	15 Minutes
	Horizontal Scalability	23 Minutes	21 Minutes

Table 4-4: OpenStack – Second Experiment (a)

Second Test:-

	Scalability Type	KVM	XEN
CPU	Basic Machine	70% usage	76% usage
	Vertical Scalability	52% usage	60% usage
	Horizontal Scalability	67% usage	80% usage
RAM	Basic Machine	800MB	830MB
	Vertical Scalability	3.4GB	3.7GB
	Horizontal Scalability	1.38 GB	1.62 GB
TIME	Basic Machine	42 Minutes	43 Minutes
	Vertical Scalability	12 Minutes	14 Minutes
	Horizontal Scalability	22 Minutes	24 Minutes

Table 4-4: OpenStack – Second Experiment (b)

Average Calculation depend on these equations:-

CPU usage, RAM and time in all cases for basic machine, Vertical and Horizontal Scalability for both XEN and KVM

Eq1) CPU Machine Average=$\frac{CPU\ Usage\ FT + CPU\ Usage\ ST}{2}$.

Eq2) RAM Machine Average=$\frac{RAM\ Usage\ FT + RAM\ Usage\ ST}{2}$.

Eq3) Time Machine Average=$\frac{Time\ Usage\ FT + Time\ Usage\ ST}{2}$.

	Scalability Type	KVM	XEN
CPU	Basic Machine	71% usage	75.5% usage
	Vertical Scalability	53% usage	59% usage
	Horizontal Scalability	71% usage	79% usage
RAM	Basic Machine	795MB	820MB
	Vertical Scalability	3.4GB	3.7GB
	Horizontal Scalability	1.39 GB	1.61 GB
TIME	Basic Machine	42.5 Minutes	44 Minutes
	Vertical Scalability	12.3Minutes	14.3 Minutes
	Horizontal Scalability	22.5 Minutes	22.5 Minutes

Table 4-5: OpenStack – Average of Second Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-6%	-11%
RAM	-0.3 GB	-0.22GB
TIME	-2min	0min

Table 4-6: OpenStack – Difference of Second Experiment

-Test the Experiment with Multi Thread Core under graphical mode & cloud Environment.

First Test:-

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	55% usage	56% usage
	Horizontal Scalability	75% usage	82% usage
RAM	Basic Machine	-	-
	Vertical Scalability	3.1GB	3.3GB
	Horizontal Scalability	1.8 GB	1.8 GB
TIME	Basic Machine	-	-
	Vertical Scalability	14 Minutes	13 Minutes
	Horizontal Scalability	21 Minutes	23 Minutes

Table 4-7: OpenStack – Third Experiment (a)

Second Test:-

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	54% usage	57% usage
	Horizontal Scalability	73% usage	84% usage
RAM	Basic Machine	-	-
	Vertical Scalability	3.09GB	3.35GB
	Horizontal Scalability	1.78 GB	1.83 GB
TIME	Basic Machine	-	-
	Vertical Scalability	13 Minutes	15Minutes
	Horizontal Scalability	20 Minutes	25 Minutes

Table 4-7: OpenStack – Third Experiment (b)

Average Calculation depend on these equations:-

CPU usage, RAM and time in all cases for basic machine, Vertical and Horizontal Scalability for both XEN and KVM

Eq1) CPU Machine Average=$\frac{CPU\ Usage\ FT\ + CPU\ Usage\ ST}{2}$.

Eq2) RAM Machine Average=$\frac{RAM\ Usage\ FT\ + RAM\ Usage\ ST}{2}$.

Eq3) Time Machine Average=$\frac{Time\ Usage\ FT\ + Time\ Usage\ ST}{2}$.

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	54.5% usage	56.5% usage
	Horizontal Scalability	74.5% usage	83% usage
RAM	Basic Machine	-	-
	Vertical Scalability	3.095GB	3.42GB
	Horizontal Scalability	1.79 GB	1.81 GB
TIME	Basic Machine	-	-
	Vertical Scalability	13.3Minutes	14 Minutes
	Horizontal Scalability	20.5 Minutes	24 Minutes

Table 4-8: OpenStack – Average Third Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-2%	8.5%
RAM	-0.325 GB	-0.02GB
TIME	-0.5min	-3.5min

Table 4-9: OpenStack – Difference of Third Experiment

-Test the Experiment with Remote Access Using SSH Server under cloud Environment.

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	94% usage	95% usage
	Vertical Scalability	71% usage	75% usage
	Horizontal Scalability	87% usage	96% usage
RAM	Basic Machine	940MB	960MB
	Vertical Scalability	5.6GB	4.9GB
	Horizontal Scalability	1.9 GB	1.9 GB
TIME	Basic Machine	49 Minutes	55 Minutes
	Vertical Scalability	17 Minutes	18 Minutes
	Horizontal Scalability	31 Minutes	30 Minutes

Table 4-10: OpenStack –Fourth Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-11%	-9%
RAM	-0.6 GB	0
TIME	-1min	-3min

Table 4-11: OpenStack – Difference of Fourth Experiment

-Test the Experiment with another Hardware architecture (RAM with Dual Channel Mode) under graphical mode & cloud Environment.

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	65% usage	71% usage
	Horizontal Scalability	76% usage	78% usage
RAM	Basic Machine	-	-
	Vertical Scalability	4.5GB	5.2GB
	Horizontal Scalability	1.6 GB	1.8 GB
TIME	Basic Machine	-	-
	Vertical Scalability	15 Minutes	16 Minutes
	Horizontal Scalability	21 Minutes	23 Minutes

Table 4-12: OpenStack –Fifth Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-6%	-7%
RAM	-0.6 GB	-.2GB
TIME	-1min	-2min

Table 4-13: OpenStack –Difference Fifth Experiment

-Test the Experiment with Enable GPU processing (ATI STREAM Technology)Under graphical mode & cloud Environment.

	Scalability Type	KVM	XEN
CPU	Basic Machine	70% usage	79% usage
	Vertical Scalability	55% usage	65% usage
	Horizontal Scalability	84% usage	93% usage
RAM	Basic Machine	950	940
	Vertical Scalability	3.5GB	3.9GB
	Horizontal Scalability	1.4 GB	1.6 GB
TIME	Basic Machine	41 Minutes	40 Minutes
	Vertical Scalability	13 Minutes	12 Minutes
	Horizontal Scalability	17 Minutes	19 Minutes

Table 4-14: OpenStack – Sixth Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-10%	-9%
RAM	-0.4 GB	-0.2GB
TIME	0.5min	-2min

Table 4-15: OpenStack – Difference of Sixth Experiment

-Test the Experiment with Changing Horizontal Scalability Design.

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	92% usage	94% usage
	Vertical Scalability	58% usage	65% usage
	Horizontal Scalability	81% usage	90% usage
RAM	Basic Machine	920MB	960MB
	Vertical Scalability	4.7GB	5.1GB
	Horizontal Scalability	3.5 GB	3.8 GB
TIME	Basic Machine	47 Minutes	50 Minutes
	Vertical Scalability	16 Minutes	15 Minutes
	Horizontal Scalability	24Minutes	25Minutes

Table 4-16: OpenStack – Seventh Experiment

Difference between Scalability Type with Both KVM and XEN depend on these Calculations:-

1) CPU Usage for Vertical Scalability = KVM usage – XEN usage.

2) CPU Usage for Horizontal Scalability = KVM usage – XEN usage.

3) RAM Usage for Vertical Scalability = KVM usage – XEN usage.

4) RAM Usage for Horizontal Scalability = KVM usage – XEN usage.

5) Time Consuming for Horizontal Scalability = KVM– XEN.

6) Time Consuming for Vertical Scalability= KVM-XEN.

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-7%	-9%
RAM	-0.4 GB	-0.3GB
TIME	.5min	-0.5min

Table 4-17: OpenStack – Difference of Seventh Experiment

Charts show the relation between KVM and vertical Scalability with regards of CPU, RAM and Time:-

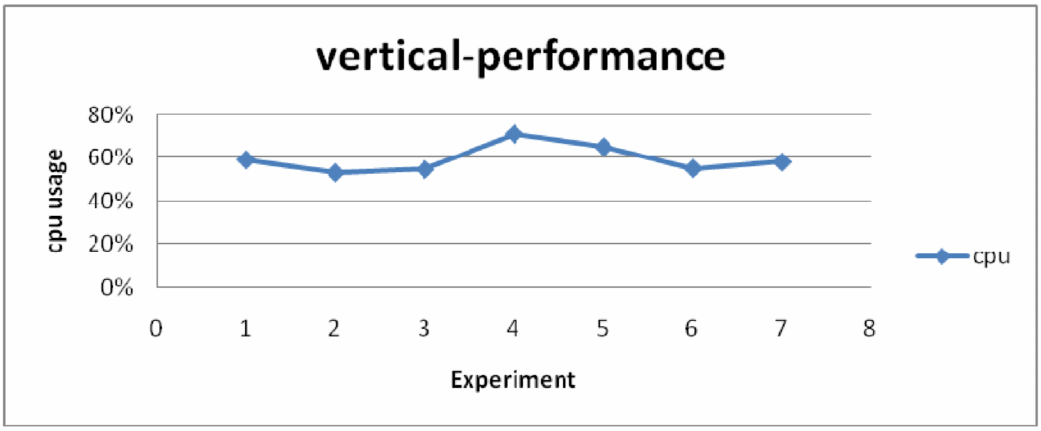


Figure4-1: KVM-Vertical Performance of CPU

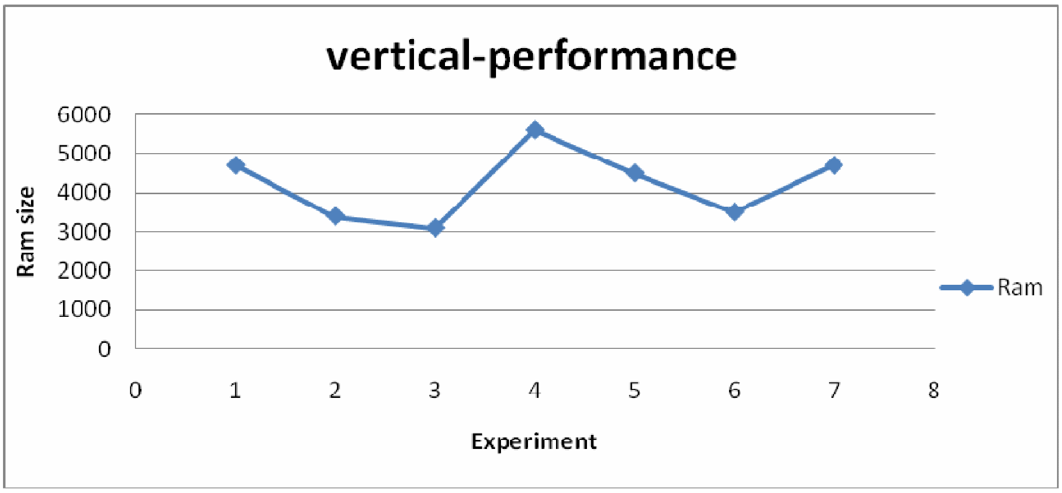


Figure4-2: KVM-Vertical Performance of RAM

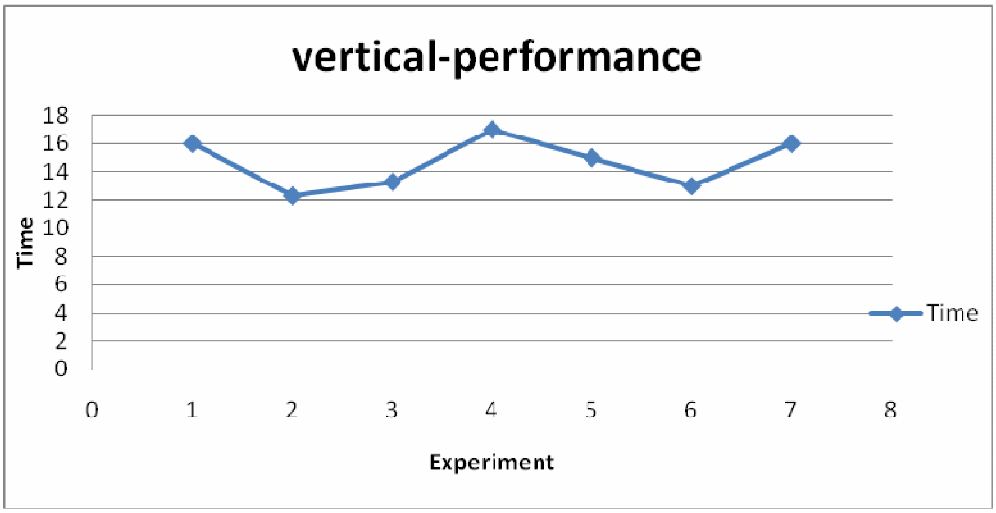


Figure4-3: KVM-Vertical Performance of Time

Charts show the relation between KVM and Horizontal Scalability with regards of CPU, RAM and Time:-

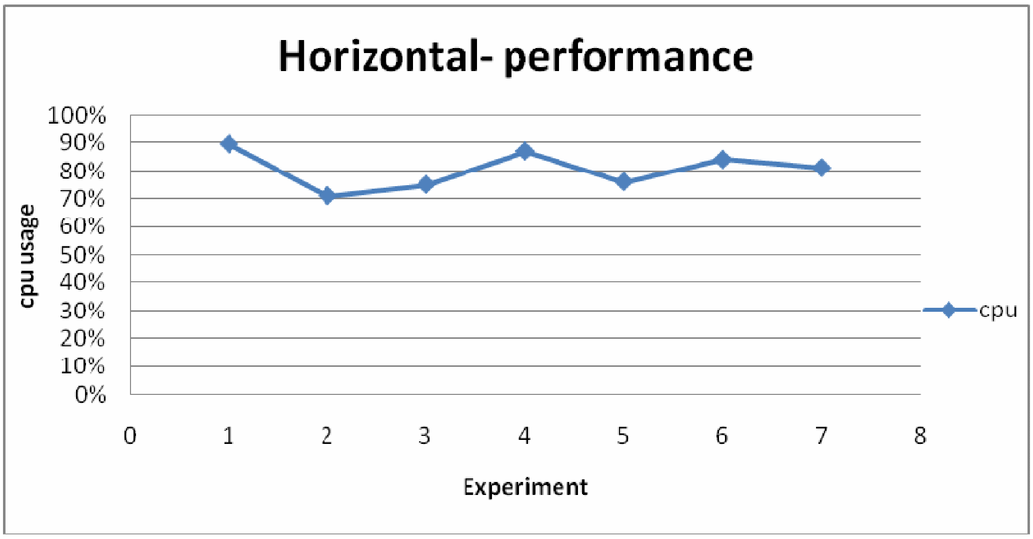


Figure4-4: KVM-Horizontal Performance of CPU

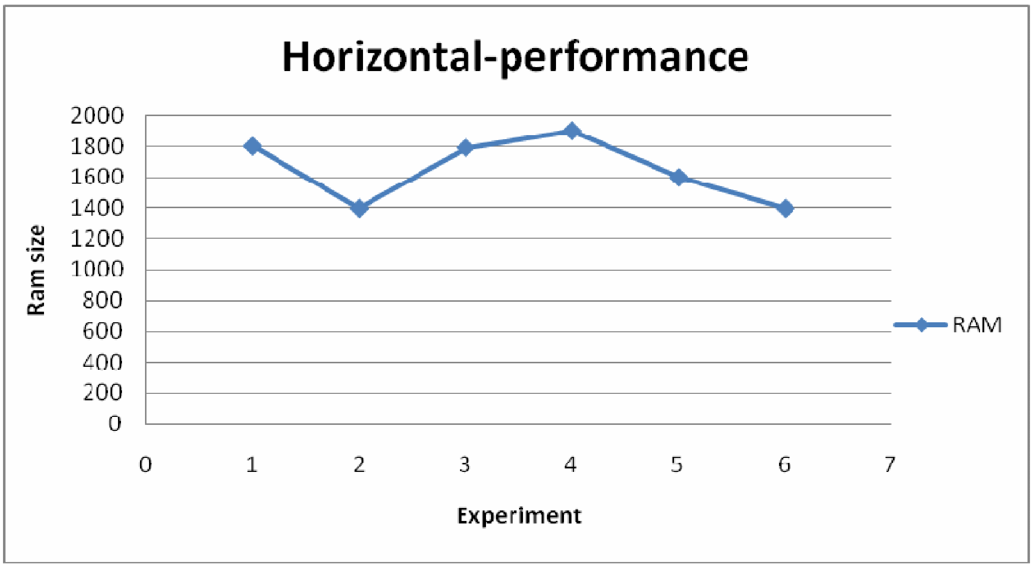


Figure4-5: KVM-Horizontal Performance of RAM

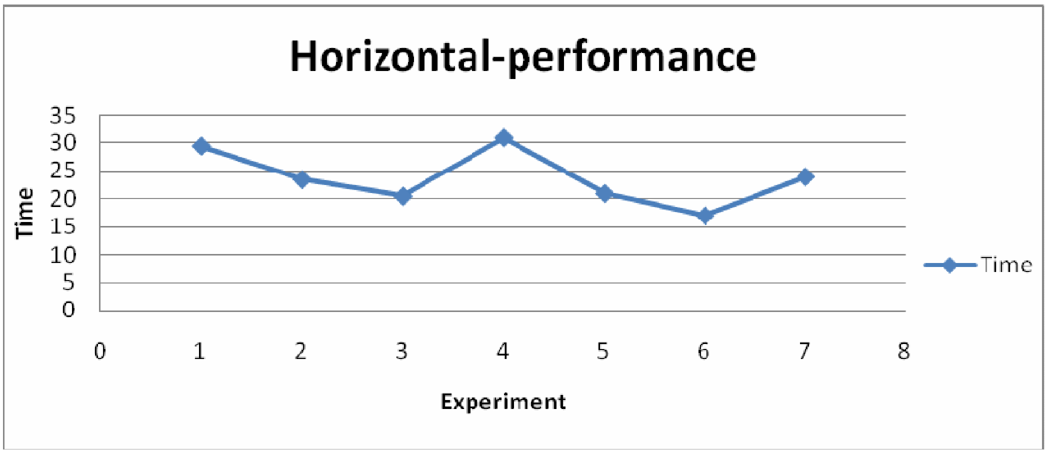


Figure4-6: KVM-Horizontal Performance of Time

Charts show the relation between XEN and vertical Scalability with regards of CPU, RAM and Time:-

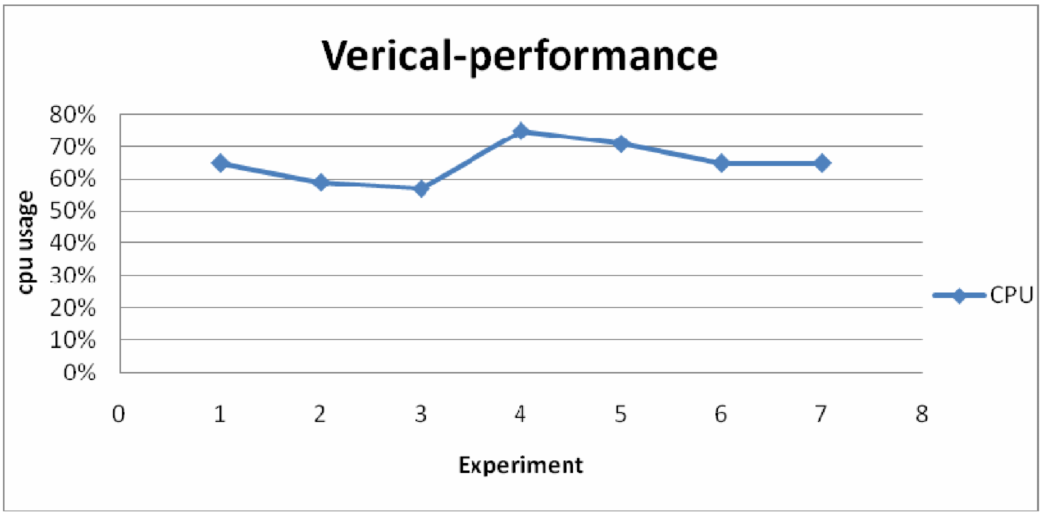


Figure4-7: XEN-Vertical Performance of CPU

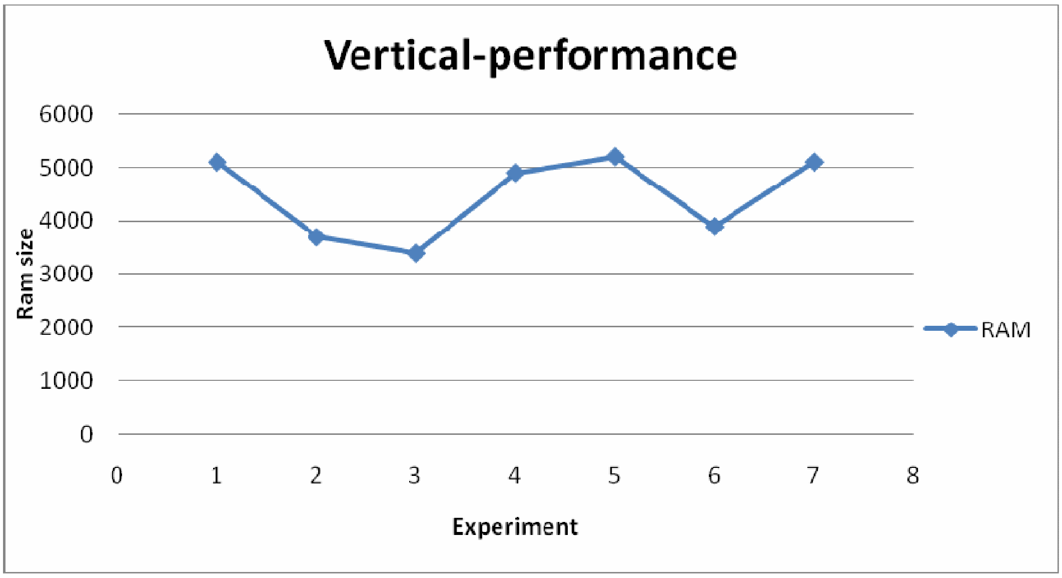


Figure4-8: XEN-Vertical Performance of RAM

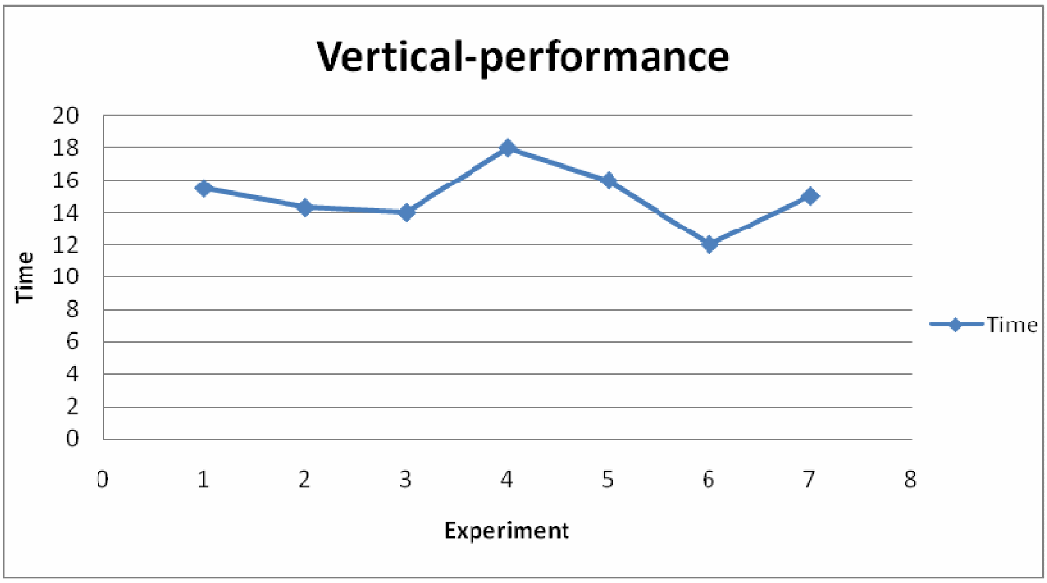


Figure4-9: XEN-Vertical Performance of Time

Charts show the relation between XEN and Horizontal Scalability with regards of CPU, RAM and Time:-

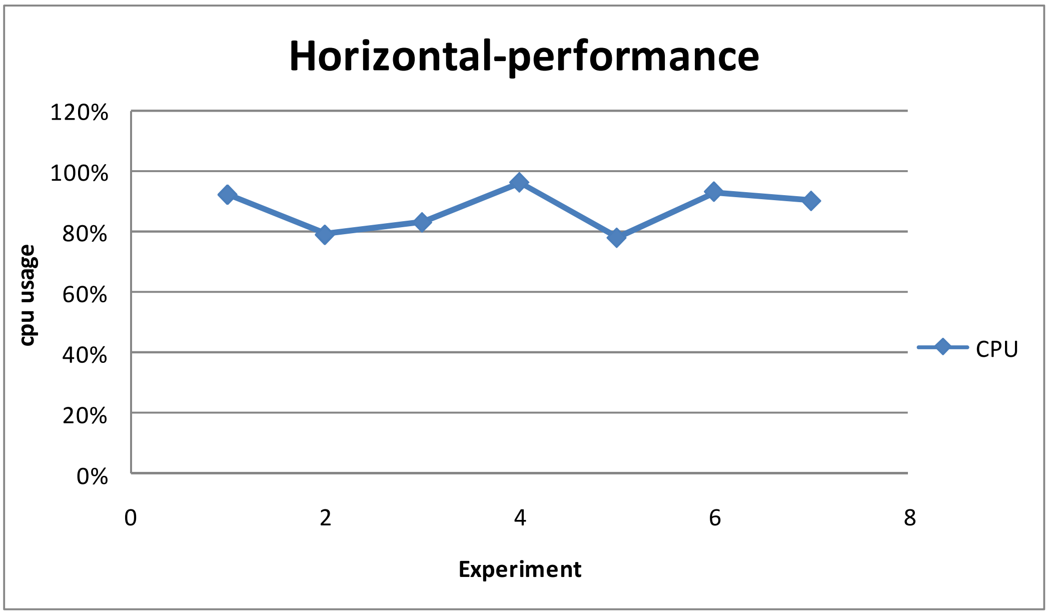


Figure4-10: XEN-Horizontal Performance of CPU

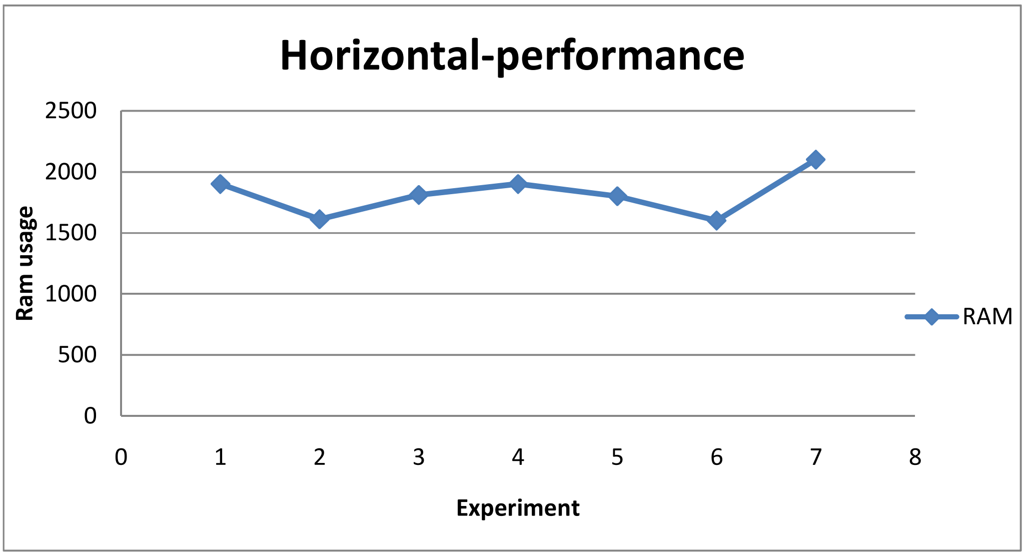


Figure4-11: XEN-Horizontal Performance of RAM

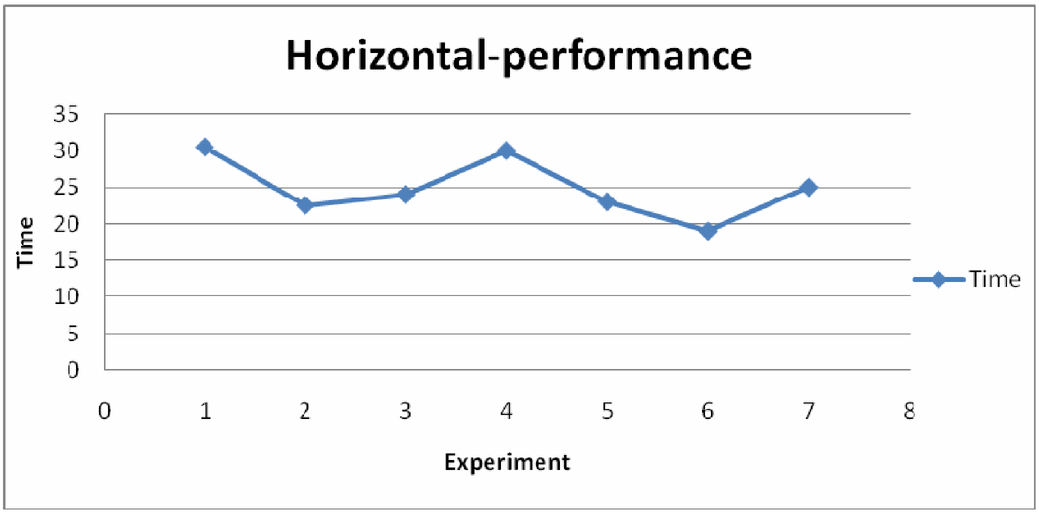


Figure4-12: XEN-Horizontal Performance of Time

4.3.2 OpenNebula Experiment

Result of implementing Scalability with **OpenNebula** with two types KVM and XEN with different criteria's. - Test the Experiment with standard graphical mode

First Test:-

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	91% usage	96% usage
	Vertical Scalability	59% usage	69% usage
	Horizontal Scalability	92% usage	92% usage
RAM	Basic Machine	930MB	980MB
	Vertical Scalability	4.7GB	5.3GB
	Horizontal Scalability	1.7 GB	1.9 GB
TIME	Basic Machine	45 Minutes	52 Minutes
	Vertical Scalability	15 Minutes	18 Minutes
	Horizontal Scalability	28 Minutes	32 Minutes

Table 4-18: OpenNebula – First Experiment (a)

Second Test:-

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	93% usage	96% usage
	Vertical Scalability	59% usage	69% usage
	Horizontal Scalability	94% usage	93% usage
RAM	Basic Machine	950MB	980MB
	Vertical Scalability	4.7GB	5.3GB
	Horizontal Scalability	1.7 GB	1.9 GB
TIME	Basic Machine	44 Minutes	51 Minutes
	Vertical Scalability	15 Minutes	17 Minutes
	Horizontal Scalability	29 Minutes	31 Minutes

Table 4-18: OpenNebula – First Experiment (b)

Average Calculation depend on these equations:-

CPU usage, RAM and time in all cases for basic machine, Vertical and Horizontal Scalability for both XEN and KVM

Eq1) CPU Machine Average=$\frac{CPU\ Usage\ FT + CPU\ Usage\ ST}{2}$.

Eq2) RAM Machine Average=$\frac{RAM\ Usage\ FT + RAM\ Usage\ ST}{2}$.

Eq3) Time Machine Average=$\frac{Time\ Usage\ FT + Time\ Usage\ ST}{2}$.

	Scalability Type	KVM	XEN
CPU	Basic Machine	92% usage	96% usage
	Vertical Scalability	59% usage	69% usage
	Horizontal Scalability	93% usage	92.5% usage
RAM	Basic Machine	940MB	980MB
	Vertical Scalability	4.7GB	5.3GB
	Horizontal Scalability	1.7 GB	1.9 GB
TIME	Basic Machine	44.5 Minutes	51.5 Minutes
	Vertical Scalability	15 Minutes	17.5 Minutes
	Horizontal Scalability	28.5 Minutes	31.5 Minutes

Table 4-19: OpenNebula – Average of First Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-10%	5%
RAM	-0.6 GB	- 0.2 GB
TIME	-2.5min	-3min

Table 4-20: OpenNebula – Difference of First Experiment

-Test the Experiment with Command Line mode after switch off lightdm component (graphical mode) operating system under cloud Environment.

First Test:-

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	69% usage	80% usage
	Vertical Scalability	56% usage	62% usage
	Horizontal Scalability	72% usage	81% usage
RAM	Basic Machine	820MB	840MB
	Vertical Scalability	3.5GB	3.8GB
	Horizontal Scalability	1.5 GB	1.7 GB
TIME	Basic Machine	42 Minutes	45 Minutes
	Vertical Scalability	12 Minutes	16 Minutes
	Horizontal Scalability	22 Minutes	23 Minutes

Table 4-21: OpenNebula –Second Experiment (a)

Second Test:-

	Scalability Type	KVM	XEN
CPU	Basic Machine	71% usage	84% usage
	Vertical Scalability	55% usage	65% usage
	Horizontal Scalability	78% usage	86% usage
RAM	Basic Machine	835MB	850MB
	Vertical Scalability	3.51GB	3.83GB
	Horizontal Scalability	1.52 GB	1.71 GB
TIME	Basic Machine	41 Minutes	45 Minutes
	Vertical Scalability	11 Minutes	16 Minutes
	Horizontal Scalability	22 Minutes	24 Minutes

Table 4-21: OpenNebula –Second Experiment (b)

Average Calculation depend on these equations:-

CPU usage, RAM and time in all cases for basic machine, Vertical and Horizontal Scalability for both XEN and KVM

Eq1) CPU Machine Average=$\frac{CPU\ Usage\ FT + CPU\ Usage\ ST}{2}$.

Eq2) RAM Machine Average=$\frac{RAM\ Usage\ FT + RAM\ Usage\ ST}{2}$.

Eq3) Time Machine Average=$\frac{Time\ Usage\ FT + Time\ Usage\ ST}{2}$.

	Scalability Type	KVM	XEN
CPU	Basic Machine	70% usage	82% usage
	Vertical Scalability	55.5% usage	63% usage
	Horizontal Scalability	75% usage	83.5% usage
RAM	Basic Machine	827.5MB	845MB
	Vertical Scalability	3.505GB	3.815GB
	Horizontal Scalability	1.51 GB	1.705 GB
TIME	Basic Machine	43 Minutes	45 Minutes
	Vertical Scalability	11.3 Minutes	16 Minutes
	Horizontal Scalability	22 Minutes	23.3 Minutes

Table 4-22: OpenNebula –Average of Second Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-7.5%	-8.5%
RAM	-.31 GB	- .195 GB
TIME	-4.3min	-1.3min

Table 4-23: OpenNebula –Difference of Second Experiment

-Test the Experiment with Multi Thread Core under graphical mode & cloud Environment.

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	52% usage	52% usage
	Horizontal Scalability	78% usage	88% usage
RAM	Basic Machine	-	-
	Vertical Scalability	3.0GB	3.4GB
	Horizontal Scalability	1.7 GB	1.8 GB
TIME	Basic Machine	-	-
	Vertical Scalability	12 Minutes	15 Minutes
	Horizontal Scalability	20 Minutes	25 Minutes

Table 4-24: OpenNebula –Third Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	0%	-10%
RAM	-0.4 GB	-0.1 GB
TIME	-3min	-5min

Table 4-25: OpenNebula –Difference of Third Experiment

-Test the Experiment with Remote Access Using SSH Server under cloud Environment.

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	96% usage	97% usage
	Vertical Scalability	73% usage	83% usage
	Horizontal Scalability	85% usage	97% usage
RAM	Basic Machine	960MB	980MB
	Vertical Scalability	5.8GB	6.1GB
	Horizontal Scalability	1.9 GB	1.9 GB
TIME	Basic Machine	51 Minutes	59 Minutes
	Vertical Scalability	15 Minutes	19 Minutes
	Horizontal Scalability	25 Minutes	31 Minutes

Table 4-26: OpenNebula – Fourth Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-19%	-12%
RAM	-0.8 GB	0 GB
TIME	-4min	-7min

Table 4-27: OpenNebula – Difference of Fourth Experiment

-Test the Experiment with another Hardware architecture (RAM with Dual Channel Mode) under graphical mode & cloud Environment.

	Scalability Type	KVM	XEN
CPU	Basic Machine	-	-
	Vertical Scalability	68% usage	75% usage
	Horizontal Scalability	74% usage	81% usage
RAM	Basic Machine	-	-
	Vertical Scalability	4.6GB	5.4GB
	Horizontal Scalability	1.5 GB	1.9 GB
TIME	Basic Machine	-	-
	Vertical Scalability	17 Minutes	17 Minutes
	Horizontal Scalability	19 Minutes	25 Minutes

Table 4-28: OpenNebula – Fifth Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-7%	-7%
RAM	-0.8 GB	-0.4 GB
TIME	0min	-6min

Table 4-29: OpenNebula – Difference of of Fifth Experiment

-Test the Experiment with Enable GPU processing (ATI STREAM Technology) under graphical mode & cloud Environment.

	Scalability Type	KVM	XEN
CPU	Basic Machine	68% usage	85% usage
	Vertical Scalability	54% usage	66% usage
	Horizontal Scalability	86% usage	92% usage
RAM	Basic Machine	930	960
	Vertical Scalability	3.5GB	4.0GB
	Horizontal Scalability	1.5 GB	1.8 GB
TIME	Basic Machine	39 Minutes	42 Minutes
	Vertical Scalability	12 Minutes	13 Minutes
	Horizontal Scalability	15 Minutes	19 Minutes

Table 4-30: OpenNebula – Sixth Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-12%	-6%
RAM	-0.5 GB	-0.3 GB
TIME	-1min	-4min

Table 4-31: OpenNebula – Difference of Sixth Experiment

-Test the Experiment with Changing Horizontal Scalability Design

Scaled Machine (Horizontal) 2 physical Machine: CPU Quad Core – RAM 4 GB

	Scalability Type	**KVM**	**XEN**
CPU	Basic Machine	91% usage	96% usage
	Vertical Scalability	59% usage	76% usage
	Horizontal Scalability	58% usage	69% usage
RAM	Basic Machine	930MB	980MB
	Vertical Scalability	4.7GB	5.3GB
	Horizontal Scalability	3.6GB	3.8 GB
TIME	Basic Machine	45 Minutes	52 Minutes
	Vertical Scalability	15 Minutes	18 Minutes
	Horizontal Scalability	22 Minutes	28 Minutes

Table 4-32: OpenNebula –Seventh Experiment

Difference between Scalability Type with Both KVM and XEN

CPU Usage for Vertical Scalability = KVM usage – XEN usage

CPU Usage for Horizontal Scalability = KVM usage – XEN usage

RAM Usage for Vertical Scalability = KVM usage – XEN usage

RAM Usage for Horizontal Scalability = KVM usage – XEN usage

Time Consuming for Horizontal Scalability = KVM– XEN

Time Consuming for Vertical Scalability= KVM-XEN

	Difference KVM and XEN Vertical Scalability	Difference KVM and XEN Horizontal Scalability
CPU	-17%	-11%
RAM	-0.6 GB	-0.2 GB
TIME	-3min	-6min

Table 4-33: OpenNebula –Difference of Seventh Experiment

Charts show the relation between KVM and vertical Scalability with regards of CPU, RAM and Time:-

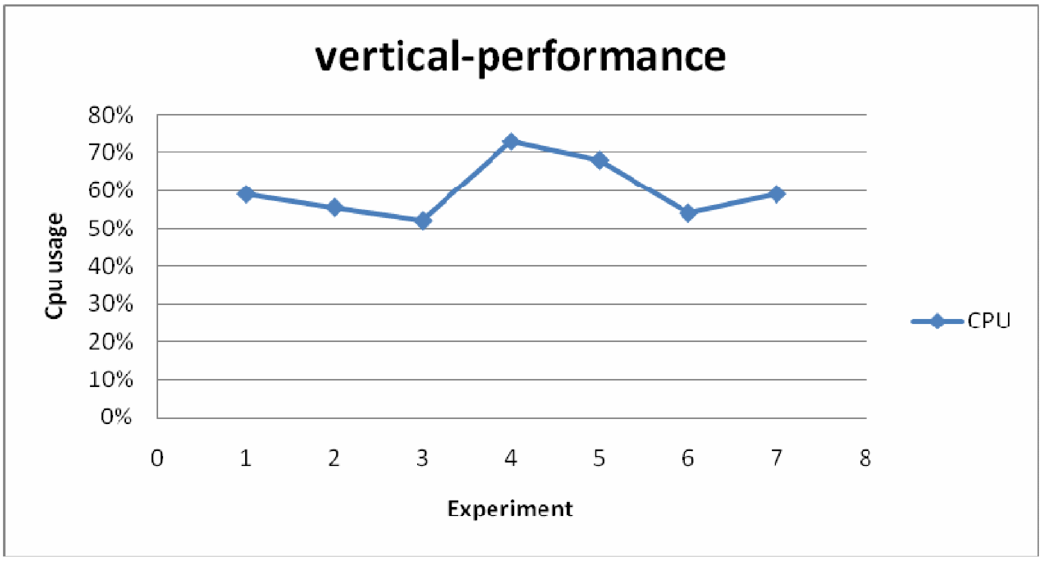


Figure4-13: KVM-Vertical Performance of CPU

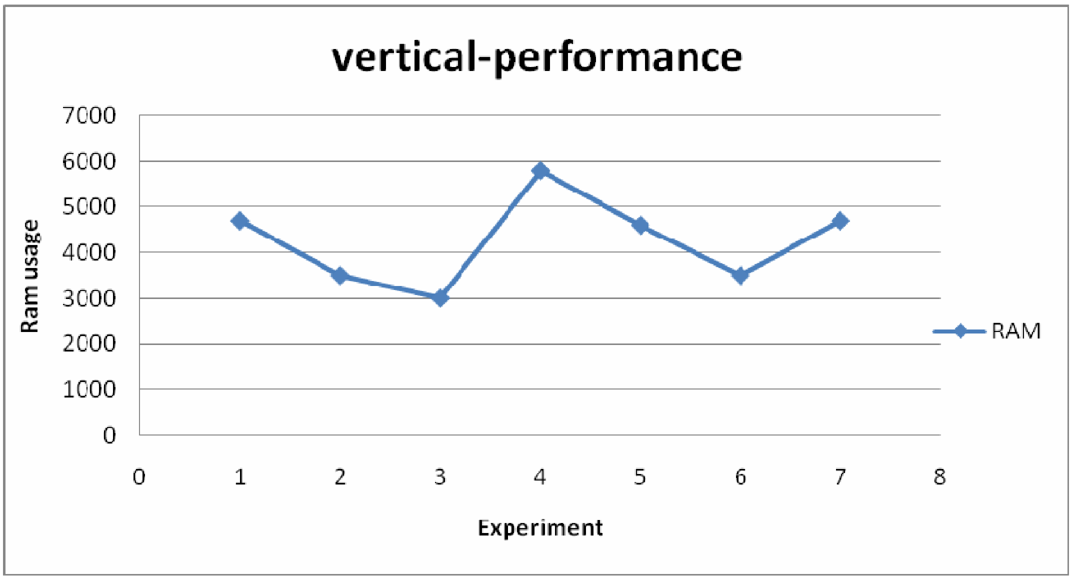


Figure4-14: KVM-Vertical Performance of RAM

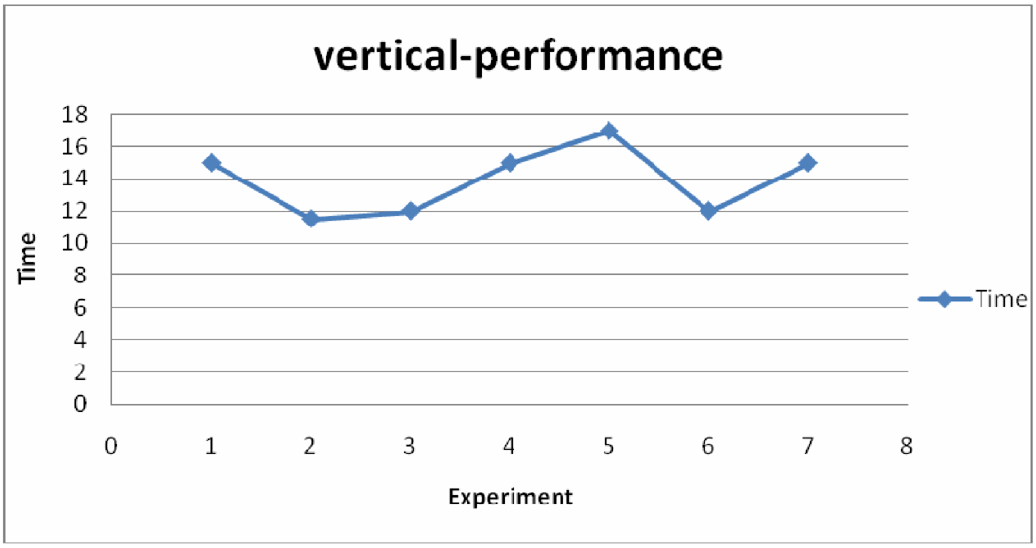


Figure4-15: KVM-Vertical Performance of Time

Charts show the relation between KVM and Horizontal Scalability with regards of CPU, RAM and Time:-

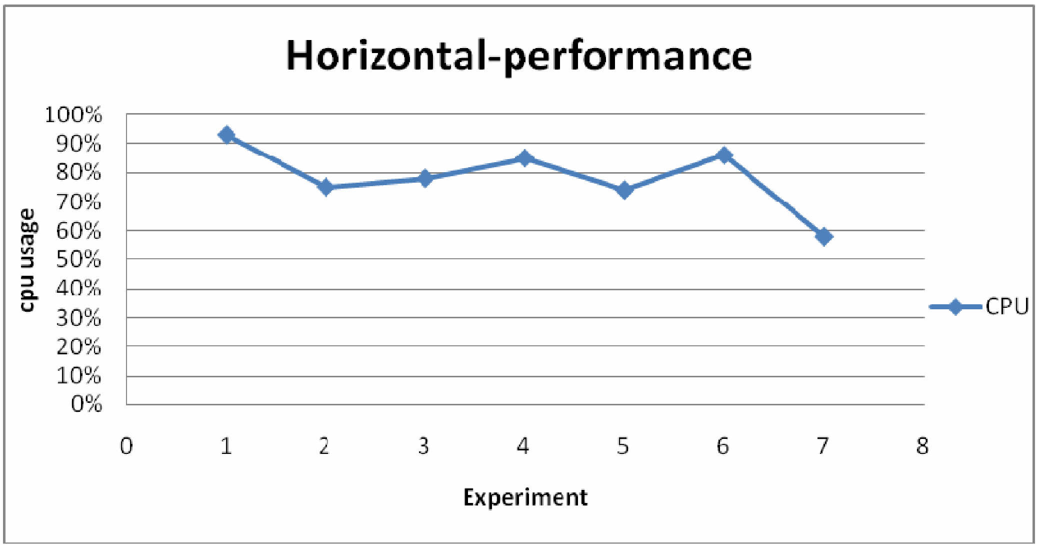


Figure4-16: KVM-Horizontal Performance of CPU

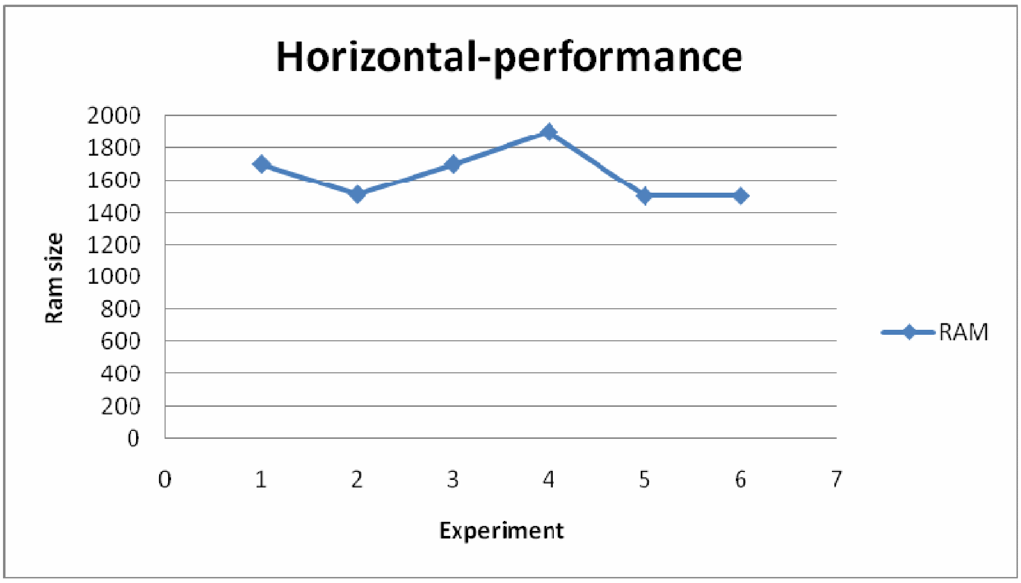


Figure4-17: KVM-Horizontal Performance of RAM

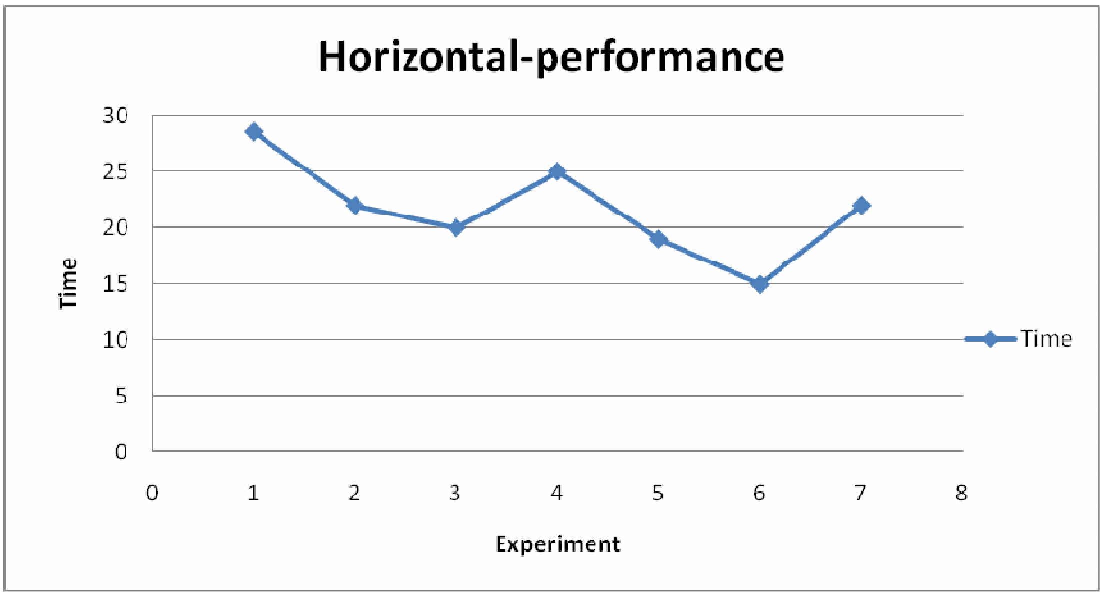


Figure4-18: KVM-Horizontal Performance of Time

Charts show the relation between XEN and vertical Scalability with regards of CPU, RAM and Time:-

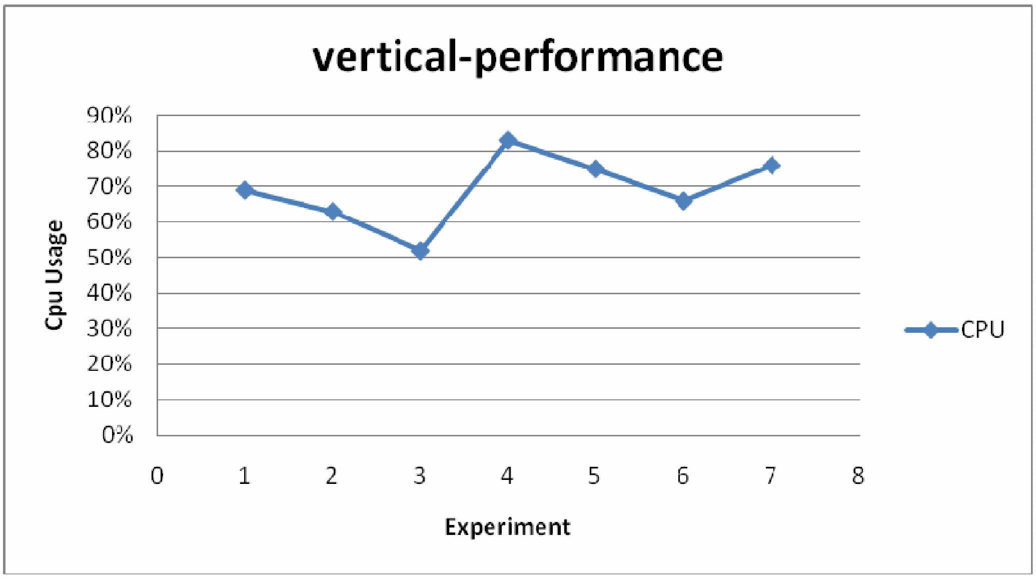


Figure4-19: XEN-Vertical Performance of CPU

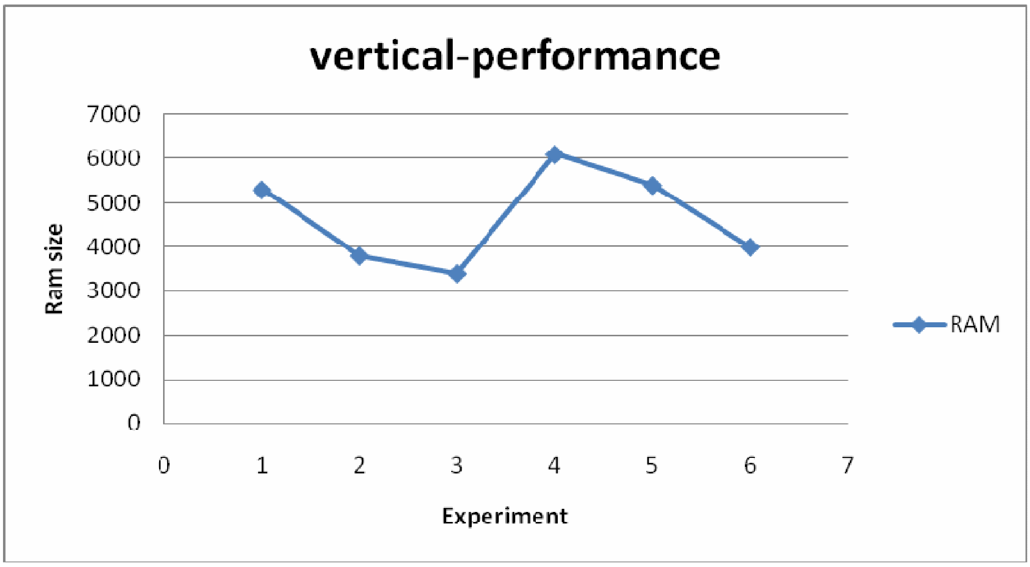


Figure4-20: XEN-Vertical Performance of RAM

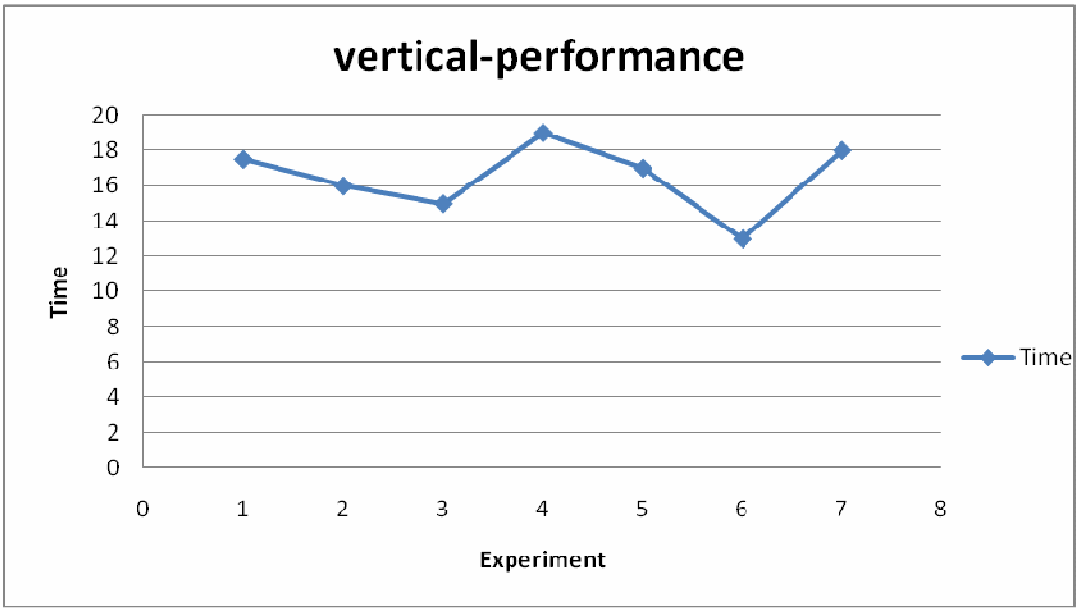


Figure4-21: XEN-Vertical Performance of Time

Charts show the relation between XEN and Horizontal Scalability with regards of CPU, RAM and Time:-

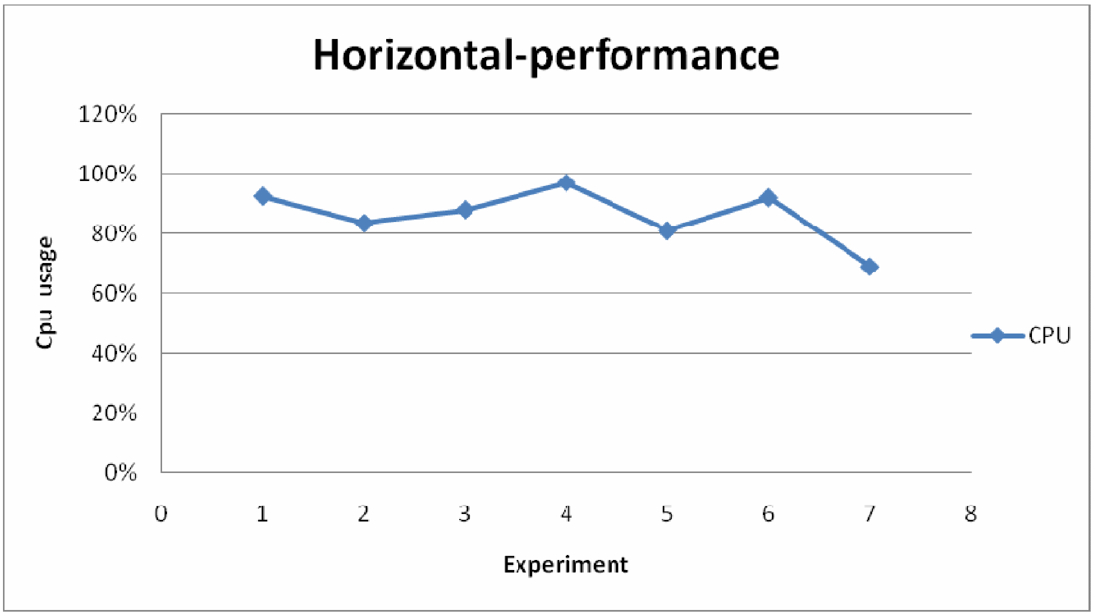


Figure4-22: XEN-Horizontal Performance of CPU

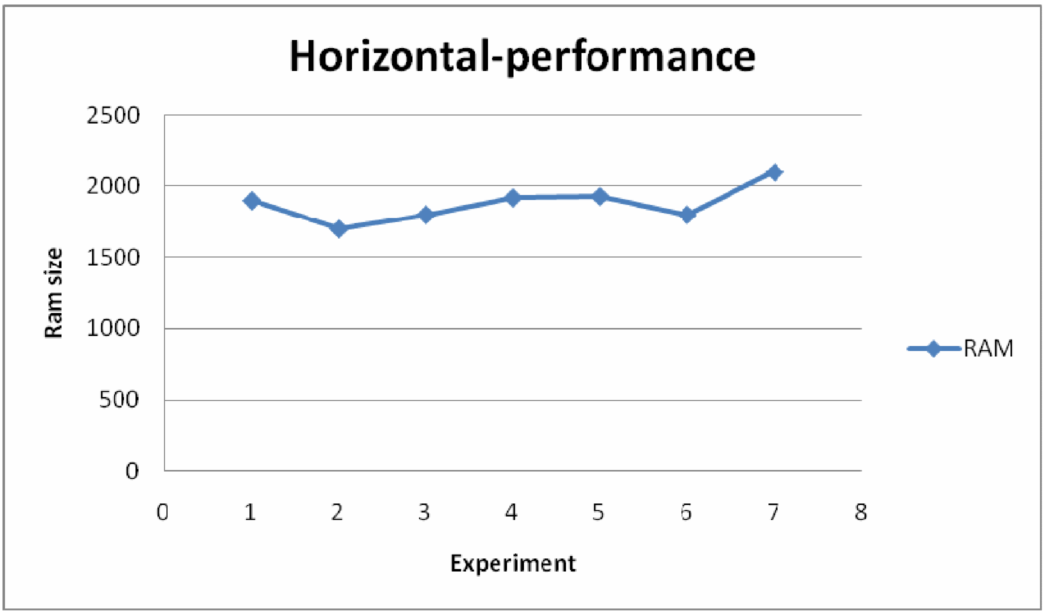


Figure4-23: XEN-Horizontal Performance of RAM

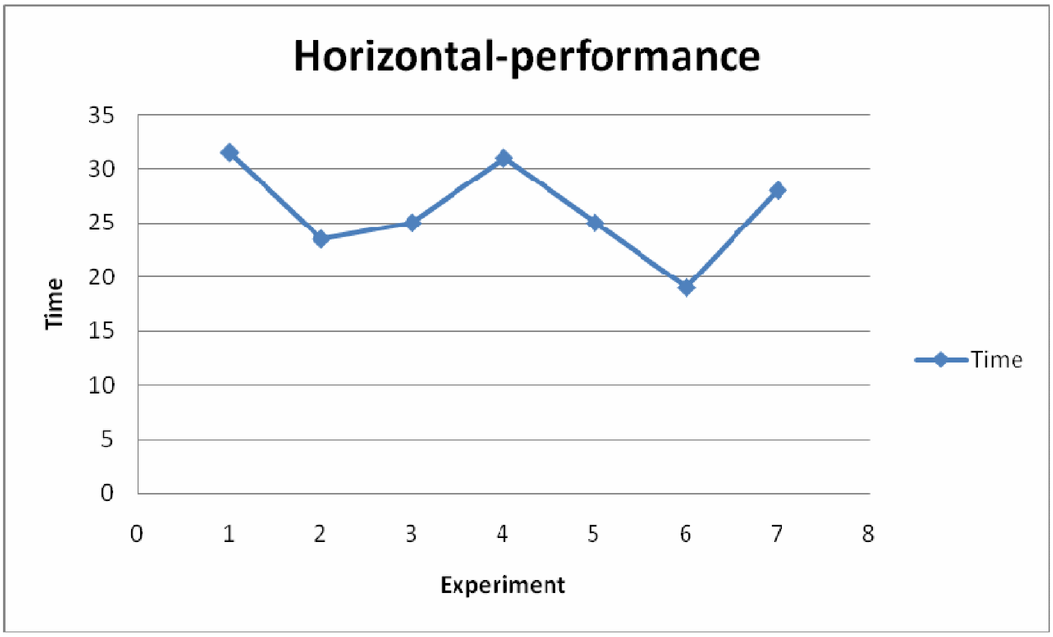


Figure4-24: XEN-Horizontal Performance of Time

CHAPTER FIVE

Conclusion and Future Work

5.1 Conclusion

This chapter concludes about the Scalability of the cloud resources with different type of hypervisors XEN and KVM with two types of cloud environments Openstack and OpenNebula using database as benchmark, the performance in this research takes into consideration CPU, RAM and time consuming utilization.

5.1.1 Openstack experiment

1- in first case results show that XEN is better with 1 minutes execution time.

Over KVM when researcher use vertical scalability at the ratio of .93 but KVM shows better at horizontal scalability with 2 minutes difference over XEN.

2- KVM usage of CPU and RAM less than XEN because the framework of XEN is more heavy than KVM.

3- in second case KVM shows better result than XEN with difference of 2 minutes at execution time with less usage of CPU and RAM.

Vertical scalability take 13 minutes execution time better horizontal scalability it’s reduce .56 in KVM and XEN reduce .62 at the same case.

5.1.2 OpenNebula experiment

1- in first case results show that KVM is better with 3 minutes execution time.

Over XEN when researcher use vertical scalability at the ratio of .83 and KVM shows better at horizontal scalability with 4 minutes difference over XEN.

2- using vertical scalability reduce up .53 of time instead of using horizontal scalability

3- in second case KVM shows better result than XEN with difference of 4 minutes at execution time with less usage of CPU and RAM.

Vertical scalability take 12 minutes execution time better horizontal scalability it's reduce .54 in KVM and XEN reduce .69 at the same case.

5.1.3 Time conclusion:

- KVM is better than XEN at most of execution time cases at both Scalability types.
- KVM is better than XEN at most of RAM and CPU usage.
- XEN is better KVM at some of execution time such as under multi thread CPU mode, Enable GPU processing and changing Horizontal Design.
- KVM with Vertical Scalability suitable more than Horizontal Scalability.
- Horizontal Scalability may cause high CPU temperature when the process take long time and that make it get slow in case of small number of cores.

5.2 Research Contribution

- KVM shows best at CPU utilization at all results, this is very important for servers, providers, customers, electric consumer, and Green Cloud interested people.
- Xen Framework is Heavy that’s case more RAM usage and CPU while processing and that’s case more time consuming
- KVM time consuming better than XEN.
- Vertical Scalability very suitable more than horizontal scalability in most time.
- Vertical scalability with KVM show very good performance than XEN with same scalability technique.
- Horizontal Scalability get more heating when divided to 4 units and that’s make more time consuming and more CPU usage.
- XEN show better results than KVM with Horizontal Scalability in some cases
- OpenStack environment performance better than OpenNebula in most cases
- As a result, KVM with vertical Scalability is the optimal choice for cloud infrastructure especially OpenStack, which it’s agreed to the literature review.

5.3 Future work

Through conducting this research, many ideas and issues were unfolded but not accomplished yet because of time, resources, and other constraints.

1- Study other Hypervisors performance like VMWARE and Hyper-V.

2- Study hybrid scalability technique.

3- Implement other database management system like oracle.

4- Study other cloud environment such as cloud Stack.

References:

1- Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007, March). Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review* (Vol. 41, No. 3, pp. 275-287). ACM.

2- Deshane, T., Shepherd, Z., Matthews, J., Ben-Yehuda, M., Shah, A., & Rao, B. (2008). Quantitative comparison of XEN and KVM. *XEN Summit, Boston, MA, USA*,

3- Gavrilovska, A., Kumar, S., Raj, H., Schwan, K., Gupta, V., Nathuji, R., ...&Saraiya, P. (2007, March). High-performance hypervisor architectures: Virtualization in hpc systems. In *Workshop on System-level Virtualization for HPC (HPCVirt).*

4- Ekanayake, J., & Fox, G. (2010). High performance parallel computing with clouds and cloud technologies. In *Cloud Computing* (pp. 20-38). Springer Berlin Heidelberg.

5- Vecchiola, C., Pandey, S., &Buyya, R. (2009, December). High-performance cloud computing: A view of scientific applications. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on* (pp. 4-16). IEEE.

6- Huber, N., von Quast, M., Hauck, M., &Kounev, S. (2011, May). Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In*CLOSER* (pp. 563-573).

7- Mei, Y., Liu, L., Pu, X., &Sivathanu, S. (2010, July). Performance measurements and analysis of network i/o applications in virtualized cloud. In*Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* (pp. 59-66). IEEE.

8- Dillon, T., Wu, C., & Chang, E. (2010, April). Cloud computing: Issues and challenges. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on* (pp. 27-33). Ieee.

9- Tsugawa, M., & Fortes, J. A. (2010). Characterizing user-level network virtualization: performance, overheads and limits. *International Journal of Network Management*, *20*(3), 149-166.

10- RAMakrishnan, L., Canon, R. S., Muriki, K., Sakrejda, I., & Wright, N. J. (2012). Evaluating Interconnect and Virtualization Performance forHigh Performance Computing. *ACM SIGMETRICS Performance Evaluation Review*, *40*(2), 55-60.

11- Dede, E., Govindaraju, M., Gunter, D., Canon, R. S., &RAMakrishna, L. (2013, June). Performance evaluation of a MongoDB and hadoop platform for scientific data analysis. In *Proceedings of the 4th ACM workshop on Scientific cloud computing* (pp. 13-20). ACM.

12- Periasamy, R. (2012, October). Performance Optimization in Cloud Computing Environment. In *Cloud Computing in Emerging Markets (CCEM),* *2012 IEEE International Conference on* (pp. 1-6). IEEE.

13- Wong, P. C., Shen, H. W., Johnson, C. R., Chen, C., & Ross, R. B. (2012). The top 10 challenges in extreme-scale visual analytics. *Computer Graphics and Applications, IEEE*, *32*(4), 63-67.

14- Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure User Data iCloud Computing Using Encryption Algorithms. *International Journal of Engineering Research and Applications (IJERA)*, *3*(4), 1922-1926.

15-Chierici, A., &Veraldi, R. (2010, April). A quantitative comparison between XEN and KVM. In *Journal of Physics: Conference Series* (Vol. 219, No. 4, p. 042005). IOP Publishing.

16-Curran, K., Carlin, S., & Adams, M. (2011). Security issues in cloud computing.*Elixir*, *38*, 4069-72.

17-Ercolani, G. (2013). Cloud Computing Services Potential Analysis. An integrated model for evaluating Software as a Service. *Cloud Computing*, 77-80.

18-Hwang, J., Zeng, S., & Wood, T. (2013, May). A component-based performance comparison of four hypervisors. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on* (pp. 269-276). IEEE.

19-Kolhe, S., &Dhage, S. (2012, October). Comparative study on Virtual Machine Monitors for cloud. In *Information and Communication Technologies (WICT), 2012 World Congress on* (pp. 425-430). IEEE.

20-Mell, P., &Grance, T. (2011). The NIST definition of cloud computing (draft).*NIST special publication*, *800*(145), 7.

21-Sabahi, F. (2012). Secure Virtualization for Cloud Environment Using Hypervisor-based Technology. *Int. Journal of Machine Learning and Computing*,*2*(1).

22-Schlosser, D., Duelli, M., &Goll, S. (2011). Performance comparison of hardware virtualization platforms. In *NETWORKING 2011* (pp. 393-405). Springer Berlin Heidelberg.

23-Xu, X., Zhou, F., Wan, J., & Jiang, Y. (2008, December). Quantifying performance properties of virtual machine. In *Information Science and Engineering, 2008. ISISE'08. International Symposium on* (Vol. 1, pp. 24-28). IEEE.

24-Yang, C. T., Wang, S. F., Huang, K. L., & Liu, J. C. (2012). On construction of cloud iaas for VM live migration using KVM and opennebula. In *Algorithms and Architectures for Parallel Processing* (pp. 225-234). Springer Berlin Heidelberg.

25-Younge, A. J., Henschel, R., Brown, J. T., von Laszewski, G., Qiu, J., & Fox, G.C. (2011, July). Analysis of virtualization technologies for high performance computing environments. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on* (pp. 9-16). IEEE.

Appendix

Links

- http://www.linux-kvm.org/page/Main_Page
- http://www.techterms.com/definition/
- http://www.ubuntu.com/
- http://www.xenproject.org/
- https://www.debian.org/
- https://www.proxmox.com/
- http://www.advanceresearchlibrary.com/