Bashar Ramadan El-Anani

Server Virtualization:

Para- and Full Virtualization: XenServer vs. KVM

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communications Technology

Bachelor's Thesis

3 November 2021

Abstract

Author:	Bashar Ramadan El-Anani
Title:	Server Virtualization: Para- and Full Virtualization- XenServer vs. KVM
Number of Pages:	29 pages + 0 appendices
Date:	3 November 2021
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communications Technology
Professional Major:	IoT and Cloud Computing
Instructors:	Tapio Wikström, Senior Lecturer

The purpose of this thesis was to implement testing environment through building bare metal Citrix XenServer and Kernel-based Virtual Machine (KVM) hypervisors and create one virtual machine on each hypervisor to examine the performance of full virtualization and paravirtualization technologies.

A seven-year-old tower server Fujitsu Primergy TX150 S8 with Xeon E5-2407 2.2 GHz processor, 16 GB RAM, 145 GB hard drive and one 1 Gb NIC was used to host hypervisors. Different virtual machine managers were introduced and discussed. After virtual machines creation, Phoronix test suite program was used to test processors and memory performance, while iPerf software was used for network performance tests.

Performance test results were collected and analyzed. Citrix XenServer virtual machine results showed better performance scores than KVM results.

In conclusion, the results proved that full virtualization performance is slower than paravirtualization performance. Although the bare metal hypervisors were built successfully and reflected good performance, it is not recommended to seek only free, used and old technology solutions, because much time would be wasted on deploying and maintaining the system. This project is suitable for IT learners and small companies' internal operations.

Keywords: virtualization, XenServer, KVM

Tiivistelmä

Tekijä:	Bashar Ramadan El-Anani
Otsikko:	Palvelin Virtualisointi: Para- ja Täysi Virtualisointi- XenServer vastaan KVM
Sivumäärä:	29 sivua + 0 liitettä
Aika:	3.11.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Verkot ja pilvipalvelut
Ohjaajat:	Lehtori Tapio Wikström

Tämän opinnäytetyön tarkoituksena oli toteuttaa testausympäristö asentamalla Citrix XenServer- ja Kernel-pohjainen virtuaalikoneen (KVM) hypervisorit ja luoda kumpaankin hypervisoriin yksi virtuaalikone, jonka avulla voidaan tarkastella täyden virtualisoinnin ja paravirtualisointi teknologioiden suorituskykyä.

Hypervisorien isännöintiin käytettiin seitsemän vuotta vanhaa tornipalvelinta Fujitsu Primergy TX150 S8, jossa oli Xeon E5-2407 2,2 GHz:n prosessori, 16 Gt RAM-muistia, 145 Gt:n kiintolevy ja yksi 1 Gt:n verkkokortti. Erilaisia virtuaalikoneen hallintaohjelmistoja esiteltiin ja niistä keskusteltiin. Virtuaalikoneiden luomisen jälkeen Phoronix-testiohjelmistoa käytettiin prosessorien ja muistin suorituskyvyn testaamiseen, kun taas iPerf-ohjelmistoa käytettiin verkon suorituskykytesteihin.

Suorituskykytestin tulokset kerättiin ja analysoitiin. Citrix XenServer - virtuaalikonetulokset ovat suorituskyvyltään parempia kuin KVM-tulokset.

Yhteenvetona voidaan todeta tuloksien osoittavan, että täyden virtualisoinnin suorituskyky on hitaampaa kuin paravirtualisoinnin suorituskyky. Vaikka suoraan laitteistolla ajattavat hypervisorit asennettiin onnistuneesti ja heijastivat hyvää suorituskykyä, ei ole suositeltavaa etsiä vain ilmaisia, käytettyjä ja vanhan teknologian ratkaisuja, koska järjestelmän käyttöönottoon ja ylläpitoon menisi paljon aikaa. Tämä projekti sopii IT-opiskelijoille ja pienyritysten sisäiselle toiminnalle.

Avainsanat: virtualisointi, XenServer, KVM

Contents

List of Abbreviations

KVM: Kernel Based Virtualization.

VM: Virtual Machine.

XAPI: Experience API.

QEMU: Quick Emulator.

OS: Operating System.

CPU: Central Processing Unit.

GPU: Graphics Processing Unit.

RAM: Random Access Memory.

VMM: Virtual Machine Manager.

PATA: Parallel ATA.

SATA: Serial ATA.

SCSI: Small Computer Systems Interface.

SAN: Storage Area Network.

HBA: Host Bus Adapter.

GUI: Graphical User Interface.

SSL: Secure Socket Layer.

TCP/IP: Transmission Control Protocol / Internet Protocol.

NIC:	Network Interface Controller.
XVA:	XenServer Virtual Appliance.
XOA:	Xen Orchestra Virtual Appliance.
ISO	International Organization for Standardization.
SMB	Server Message Block.
cifs	Common Internet File System.
BIOS	Basic Input/Output System.
UEFI	Unified Extensible Firmware Interface.
SSH	Secure Shell.
MB/s	Megabytes Per Second.

1 Introduction

Many virtualization solutions were introduced in the field of IT technology over the past 50 years. They turned from being thoughts into reality, because of the force driving virtualization where emerged the need to deliver agile and modern tools to customers to facilitate and faster the process of production or even to offer learners better educational environments. The diversity in virtualization resulted in Desktop, Application, Data, Storage, Network and Server virtualization.

The main frame of this project is server virtualization. So, shedding the light on virtualization world and hypervisors that play vital role within this frame is a necessity to pave the way to achieve deep comprehension of the practical part.

The core of this thesis is to conduct a comparative study between XenServer and KVM as representatives of para- and full virtualization. The two hypervisors are going to be deployed as bare-metal servers. Different management tools are going to be installed and discussed. A virtual machine is going to be created on each hypervisor to test the performance and analyze the results to find out which solution is more suitable for home, office or small size company networks that may possess moderate or humble hardware specifications, due to low budget limitation, to build these servers and run them efficiently.

2 Virtualization

2.1 Virtualization Concept

Historical Background:

Understanding the origin and essence of virtualization requires going back in history to the 1950s. This historical journey reveals that virtualization was, and still, a necessary element to support the cloud computing functionality. It started

when a shared access to large scale mainframe computers was given to schools and companies by using dumb terminals. Unfortunately, the project faced two problems that were mainframe's high cost and maintenance expenses. The solution came after twenty years, during the 1970s, when IBM created an operating system called VM to allow various operating systems to reside on the same computing hardware. This creation was the outcome of developing the idea of shared access to mainframe computers that was achieved in the 1950s. [1.]

Definition of Virtualization:

Virtualization is the technology that allows high utilization of IT infrastructure elements and resources by creating a virtual version of them. It delivers many benefits such as reduction in both IT expenses and management workload. Operations and services agility, availability, downtime, and mobility are also offered by virtualization. [2.]

2.2 Virtualization Types

Many forms of virtualization are available to serve different purposes. The following types are the most prevalent ones:

- Application Virtualization: it allows the use of applications directly without the need to install them on the operating system.
- Storage Virtualization: it facilitates the access and management of storage blocks on a network through creating one storage pool.
- Data Virtualization: it facilitates working with data through permitting any application to access the stored data regardless of how it was formatted, where it was stored and its source.
- Network Virtualization: a software that is used to create a virtual network that runs on a hypervisor. These networks are identical to real networks with all their components, but it is easier to deploy and manage them than the real or traditional networks.

- Desktop Virtualization: it lets the users to have access to multiple and varied operating systems desktops either remotely by following Virtual Desktop Infrastructure method or locally through utilizing a hypervisor to build Local Desktop Virtualization.
- Server Virtualization: It utilizes a software (hypervisor) to allow a certain server (host) to hold many neighboring but separated virtual machines. Each VM (guest) has its own distinct operating system, applications and assigned CPU, GPU, RAM, storage space and other virtualized components, with the possibility for communication among these VMs and other networks through virtual network adapters. The communication between the guest and host occurs by means of the hypervisor. [3.]

3 Hypervisors

3.1 Definition

A hypervisor is a software that allows the running of multiple and distinct virtual machines on the same physical machine by forming a layer of abstraction between the physical hardware and the virtual machines. It manages the VMs and their access to the physical hardware resources. It could be referred to as VMM. [4.] Hypervisors have two types that are going to be discussed in the following subchapter.

3.2 Hypervisor Types

- Type one hypervisor: they are also called bare-metal hypervisors, because they are installed directly to the physical hardware/ server. They are more secure and dynamic than type two hypervisors due to the direct contact between hypervisor and the physical hardware.
- Type two hypervisor: they require a running OS to operate and reside on. They are more suitable for personal PCs that aim to software testing or users who need different operating systems environments that run varied programs or applications. [4.]

4 XenServer as Paravirtualization Example

4.1 XenServer Overview

XenServer belongs to type one bare-metal hypervisors family. It was developed in England by the University of Cambridge and based on the Xen Project hypervisor. XenServer has features such as thin provisioning, high availability, resource pools, dynamic memory control and both VM and Storage live migrations. [5.]

A productive XenServer hypervisor architecture contains varied components, as illustrated in figure 1, that can be classified into two layers:

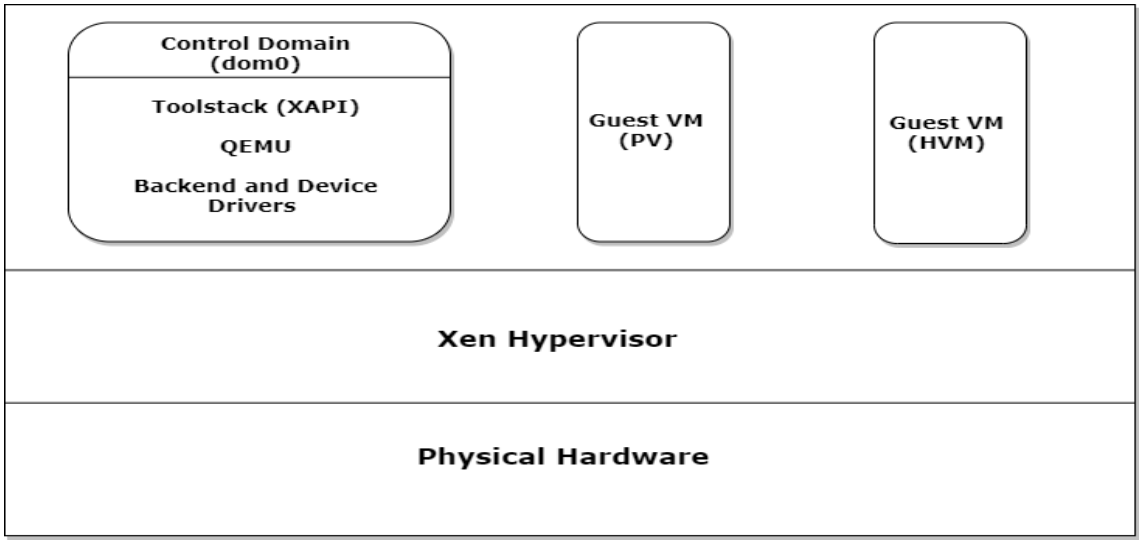

Figure 1 The XenServer Hypervisor Architecture

- Layer One: it is the physical hardware platform for XenServer hypervisor. This layer must support 64-bit x86 processors with multicore Intel VT or AMD-V.
- Layer Two: Xen Project hypervisor which is an open-source hypervisor type one that constitutes the core of XenServer hypervisor, but with less features compared to XenServer hypervisor. This layer could be divided due to their duties into the following domains:

- Domain Zero (dom0): a managerial virtual machine which is based on Centos 7.5 distribution and runs the so called XAPI or Citrix management toolstack. QEMU is also available in this domain for hardware emulation support. This domain is secure because it is a privileged area.
- Guest Domain: users' virtual machines are created here with the support of dom0 and the supervision of the operational manager XAPI and QEMU if needed. Virtualization techniques that support the guest VMs on XenServer are hardware assisted virtualization (HVM) and paravirtualization (PV). [6.]

4.2 Paravirtualization Mechanism

Paravirtualization is a method that is useful for architectures that are not classically virtualizable such as traditional x86 for high performance virtualization. It requires the modification and recompilation of the guest operating system, before installing it onto the VM, to allow calling for the hypervisor services instead of the execution of privileged hardware instructions. [7.]

Figure 2 illustrates paravirtualization mechanism by presenting the 4 levels of privilege that are offered by x86 architecture, and presented as Ring 0, 1, 2 and 3, to applications and operating systems to manage access to the computer physical side.

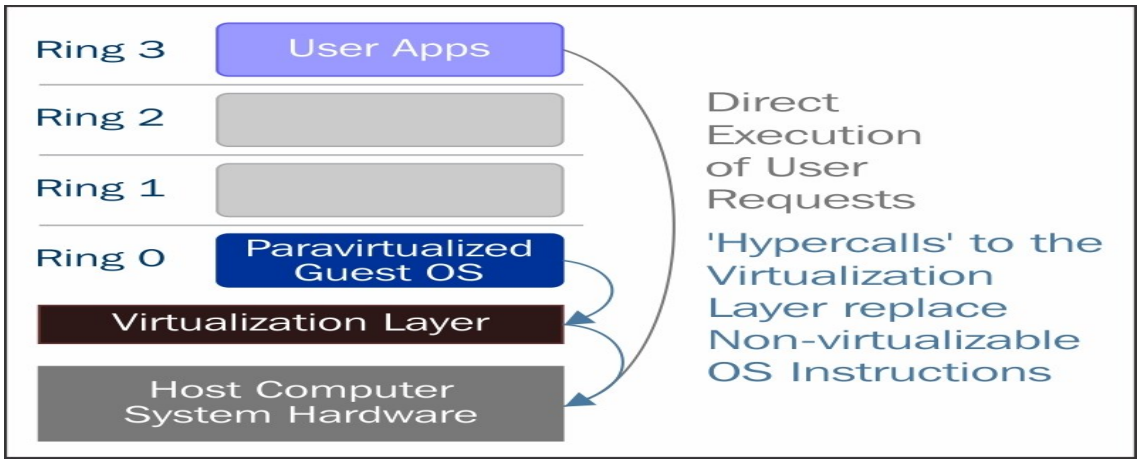

Figure 2 The Paravirtualization approach to x86 Virtualization. Reprinted from 'VMware. Understanding Full Virtualization, Paravirtualization, and Hardware Assist [online]. 2008.'

5 Implementing XenServer

5.1 Installing and Configuring XenServer

Building XenServer hypervisor bare-metal server requires to meet the following hardware specifications:

I. CPUs Requirements:
   - CPUs that support 64-bit x86 architecture with a minimal 1.5 GHZ speed and the recommended are faster multicore CPUs.
   - CPUs must also be Intel VT or AMD-V to support the running of Windows and recent distributions of Linux VMs.

II. RAM Requirements:
   - 2 GB minimum and up to 6 TB.

III. Disk Space Requirements:
   - Local storage media (PATA, SATA or SCSI) with a minimal size of 46 GB and recommended size of 70 GB, or SAN via HBA.

IV. Network Requirements:
   - One or more NICs with minimum 100 Mb, but 1 Gb or 10 Gb are recommended. [5.]

In this chapter, a tower server Fujitsu Primergy TX150 S8 with Xeon E5-2407 2.2 GHz processor, 16 GB RAM, 145 GB hard drive and one 1 Gb NIC are used to perform a fresh installation of XenServer. A copy of Citrix Hypervisor 8.0 Base Installation ISO-Express Edition file is going to be burnt to a USB flash drive. XenServer installation steps are clear and straightforward. The whole process takes from 20 to 25 minutes to be completed. After installation, XenServer is ready to be configured simply and monitored generally through the Citrix hypervisor 8.0 xsconsole screen as shown in figure 3.

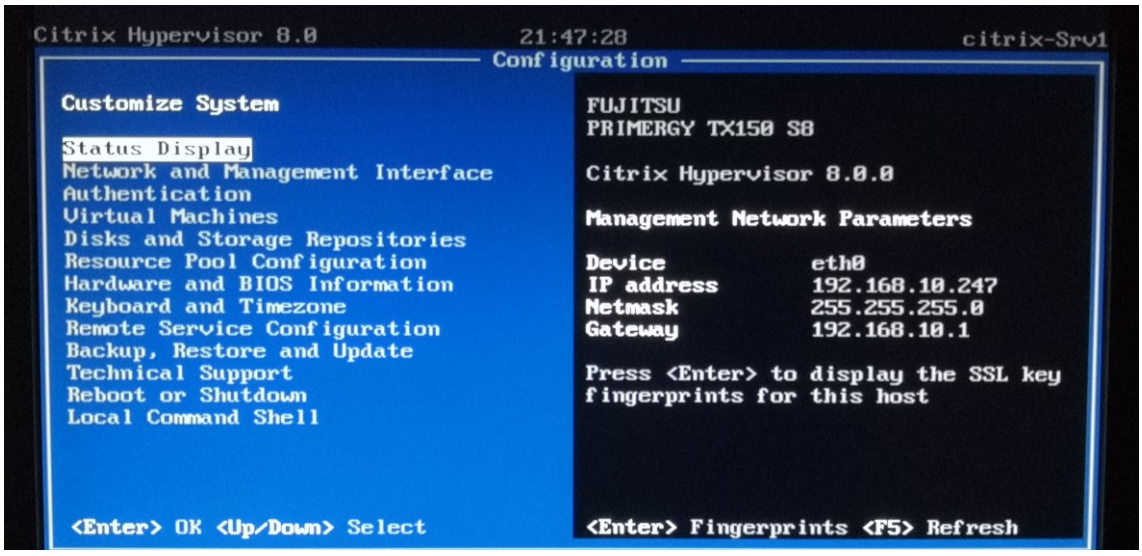

Figure 3 XenServer xsconsole

Creating VMs and performing more sophisticated configuration, management and monitoring operations are still missing until XenCenter or Xen Orchestra is used as the following chapter would explain.

5.2 XenCenter and Xen Orchestra as Management Tools

XenCenter: a client-side GUI application to deploy VMs in addition to managing and monitoring XenServer environment from a Windows desktop machine. The traffic between XenServer and XenCenter is encrypted by using SSL certificate and occurs between the graphical tools and XAPI service via TCP/IP connection. [8.] The first step after installing XenCenter is to connect to

XenServer by choosing the Add New Server option from XenCenter toolbar or clicking the icon Add a Server in the middle area of XenCenter. Information related to XenServer (IP address, username, and password) are required to create the connection. After a successful connection, XenCenter would open to Server General Properties page. Many tasks are offered such as monitoring memory, storage and server performance besides creating new VMs, storage or pool as illustrated in figure 4.

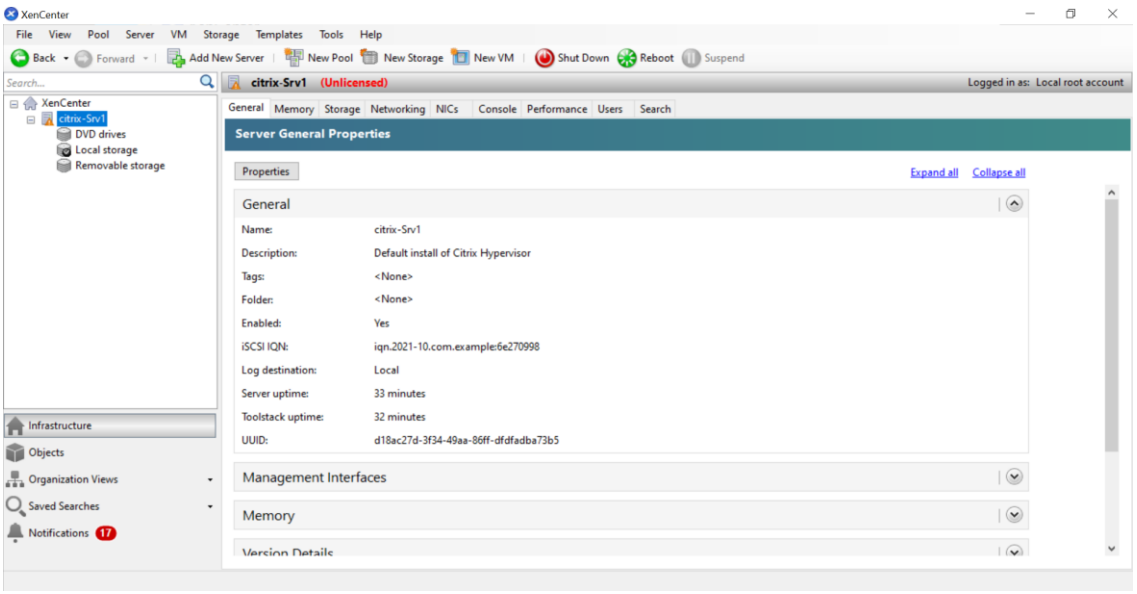

Figure 4 XenCenter Interface

Xen Orchestra: it is a web interface which is dedicated for administering XenServer from anywhere and any device. It can be downloaded for free from the Xen Orchestra official website and be registered for a premium trial period of 15 days. [9.] The downloadable file format is XVA type and must be deployed as a VM on XenCenter. These following steps are required to use Xen Orchestra:

- While XenCenter is open and XenServer is turned on, a double click should be performed on the downloaded Xen Orchestra file on Windows machine. The Import window in XenCenter opens with the path of the Xen Orchestra file as shown in figure 5.

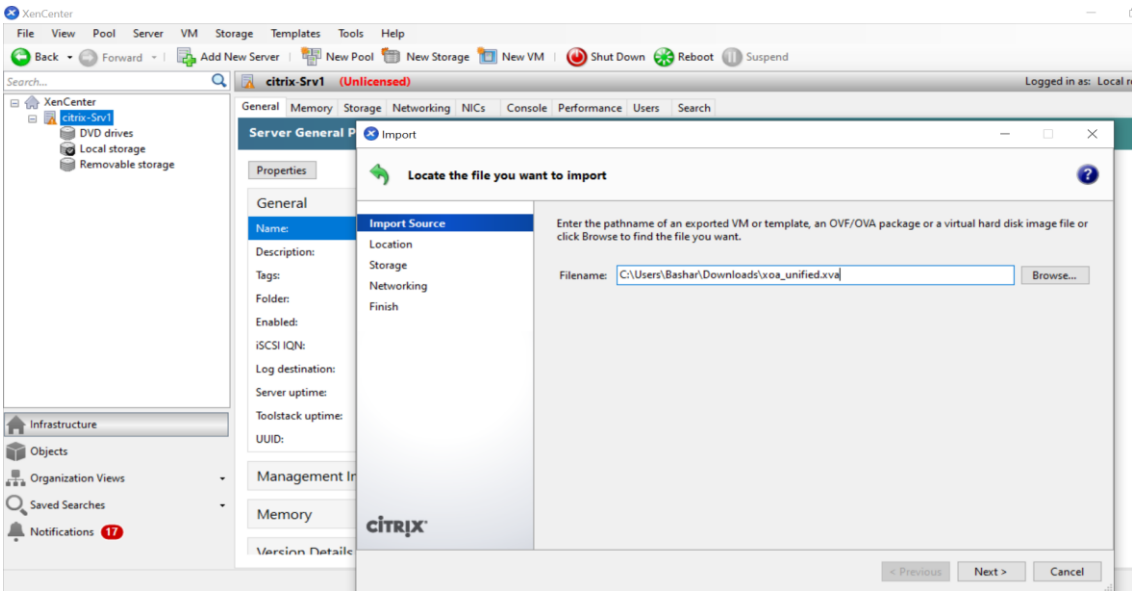

Figure 5 Xen Orchestra XVA file Import to XenCenter

- Next, the location, storage space and network interface of the imported VM location are chosen. Then click finish and the XOA VM appears under the XenServer in XenCenter.
- Right click is required on the XOA VM then selecting Start to run the VM. The yellow circle near the XOA icon turns green.
- From Network tap on the right, the IP address of the Xen Orchestra web interface is taken and can be used in a browser to reach the login page.
- The default login credentials are username: admin@admin.net and password: admin.
- The Xen Orchestra needs to be registered to receive updates and get the free premium trial period.
- To add XenServer click the icon Add Server in the middle of Xen Orchestra page. Information related to XenServer (Ip address, username, and password) are required to add it.

Now, XenServer is ready to be administered via Xen Orchestra web interface. Compared to XenCenter, Xen Orchestra is more equipped with administration utilities that facilitate management and monitoring tasks. Figure 6 illustrates Xen Orchestra XenServer Stats page.

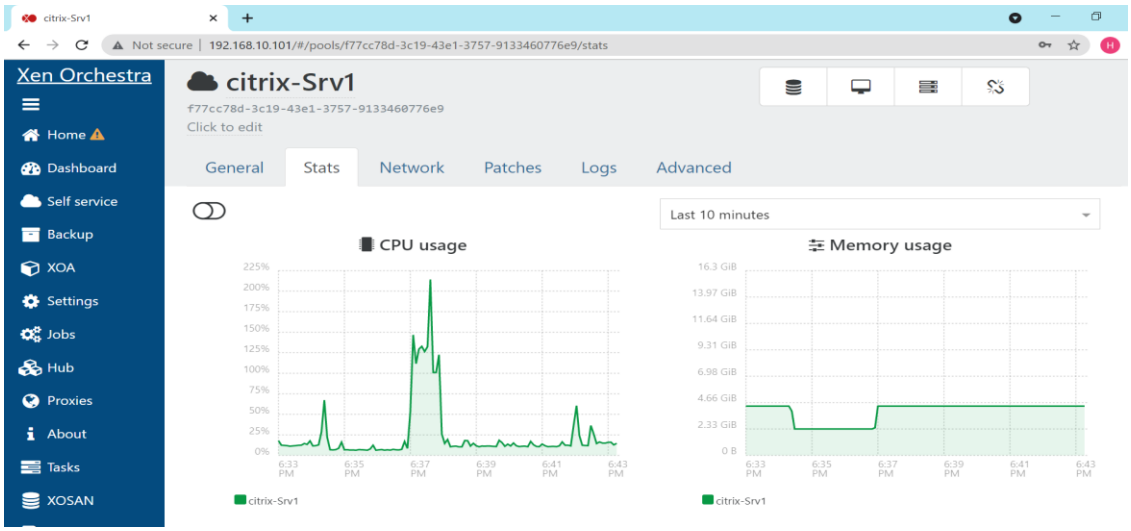

Figure 6 XenServer Stats Page in Xen Orchestra

5.3 Creating Virtual Machine on XenServer

It is possible to create VMs on XenServer by using either XenCenter or Xen Orchestra. In this work scenario, a Debian Wheezy 7 VM is going to be created by utilizing XenCenter to save on memory usage due to the 2 GB of RAM that is consumed by Xen Orchestra XOA VM on XenCenter. A Windows File Sharing (SMB/CIFS) is going to be created first to get the Debian Wheezy 7 ISO image from. A shared folder on Windows machine is prepared and given required sharing permissions. The following procedures should be executed via XenServer console before heading to final step:

1) Creating a mount point:

 # mkdir /share (this creates a folder called share under /)

2) Mounting the created directory by using the following command:

 # mount -t cifs //server/share -o username=UserName,password=myPassword /share

3) For mounting verification # *mount* command is available or # *ls -l* to list the shared content/ files.

Choosing New Storage option from XenCenter toolbar would initiate the creation of ISO files repository. Figure 7 shows the created file sharing repository.

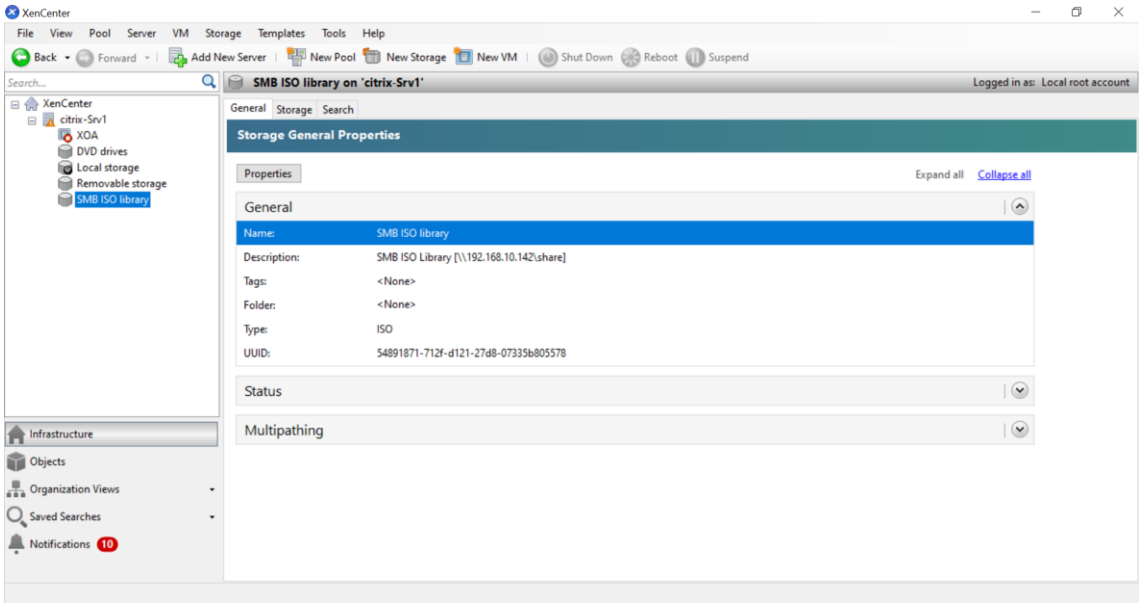

Figure 7 Created SMB/CIFS Storage

The required ISO file now is ready to create Debian Wheezy 7 VM by selecting the New VM option from XenCenter toolbar. There are seven steps to complete the VM creation process:

1) The first step is to select a template which has setting up information that suits OS requirements to create a VM. If a matching template is not available, it is possible to customize own template.
2) Selecting VM name.
3) Installation Media step comes to specify the location of the OS installation file and how the system boots (BIOS, UEFI or UEFI Secure).
4) Home Server where VM is going to resides in.
5) CPU and Memory assignments.

6) VM Storage options.
7) Networking is the final step. If another NIC is desired to hold the traffic of this VM other than management interface, this will be the right step to make the required changes.

A summary of VM configurations appears at Finish step and Create VM button is clicked to start booting the VM and OS installation. The installation of Citrix VM tools for Linux is a necessary step to enhance the performance of I/O services and avoid the traditional device emulation overhead. These tools consist of I/O paravirtualized drivers and management agent. Citrix VM tools are available for Linux and Windows (32-bit and 64-bit) operating systems. [6.] Figure 8 shows the Debian Wheezy 7 VM successful creation with Citrix VM tools for Linux.

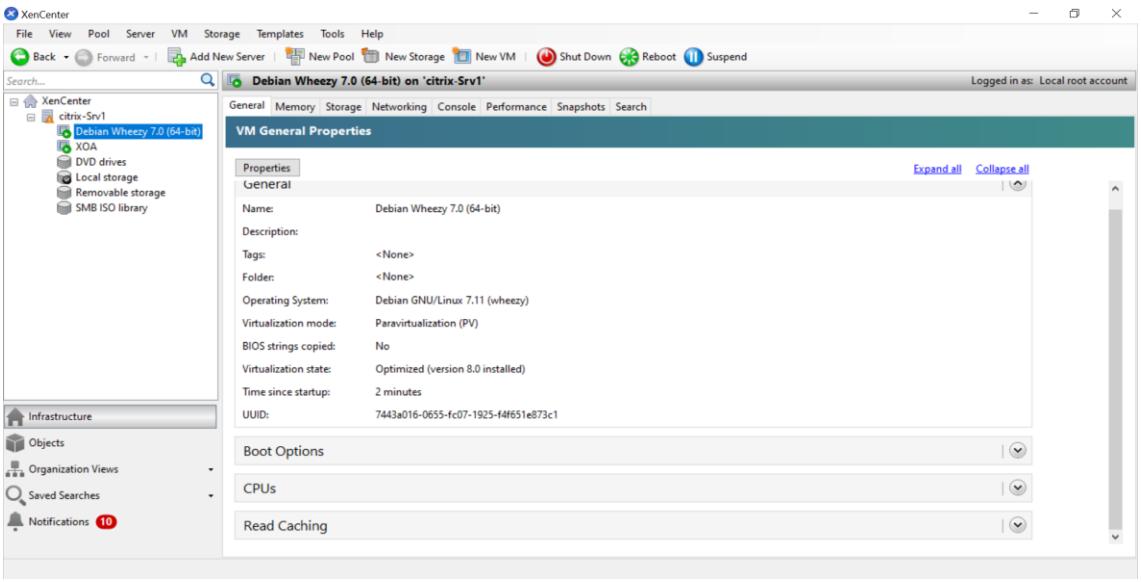

Figure 8 Created Debian Wheezy 7 VM with Citrix Guest Tools

6 KVM as Full Virtualization Example

6.1 KVM Overview

Kernel-Based Virtual Machine, or KVM as often is referred to, is a full virtualization solution that converts Linux server into type one hypervisor. The

story of KVM started in 2006 and since that time, it has become a part of the Linux Kernel version. KVM has many remarkable features such as Enhanced Security Combination (SELinux and sVirt), Live Migration, Scheduling and Resource Control, Lower Latency and Higher Prioritization. [10.]

6.2 Full Virtualization Mechanism

Although it was believed that virtualizing them was impossible, VMware found a solution to virtualize the x86 operating systems in 1998, by allowing various guest operating systems to operate on a single host operating system in complete isolation using a combination of binary translation and direct execution. That moment was the birthday of full virtualization technique. [11.] Figure 9 illustrates the VMware solution for x86 operating systems by presenting the 4 levels of privilege that are offered by x86 architecture, and presented as Ring 0, 1, 2 and 3, to applications and operating systems to manage access to the computer physical side.

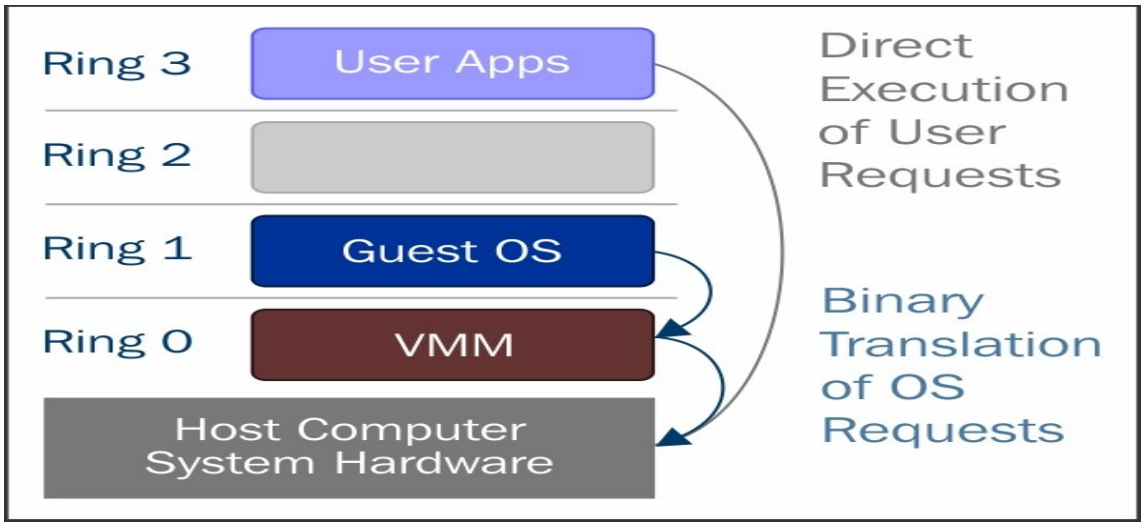

Figure 9 The Full Virtualization approach to x86 Virtualization. Reprinted from 'VMware. Understanding Full Virtualization, Paravirtualization, and Hardware Assist [online]. 2008.'

7 Implementing KVM Server

7.1 Installing and Configuring KVM

Building KVM hypervisor bare-metal server requires:

I. An x86 machine with CPU that supports virtualization extensions such as Intel CPUs with VT-x or AMD CPUs with AMD-V, and a recent release of Linux operating system for hosting KVM virtual machines. [12.]

II. RAM Requirements:
   - Minimum 2 GB of RAM.

III. Disk Space Requirements:
   - Local storage media with a minimal size of 32 GB.

IV. Network Requirements:
   - One or more NICs with minimum 100 Mb, but 1 Gb or 10 Gb are recommended.

A tower server Fujitsu Primergy TX150 S8 with Xeon E5-2407 2.2 GHz processor, 16 GB RAM, 145 GB hard drive and one 1 Gb NIC were used to install KVM. A Linux operating system, Ubuntu 20.04, was installed on this server before Installing KVM. Deploying KVM hypervisor requires going through the following steps:

a. Installing KVM tools by typing the following line in the terminal:
   # apt install qemu-kvm libvirt-clients libvirt-daemon-system bridge-utils
b. Reboot server.
c. Installing the virt-manager package, if Linux system is a graphical one:
   # apt install virt-manager

The virt-manager can be opened by typing “virt” into Activities search box. Figure 10 illustrates virt-manager window.

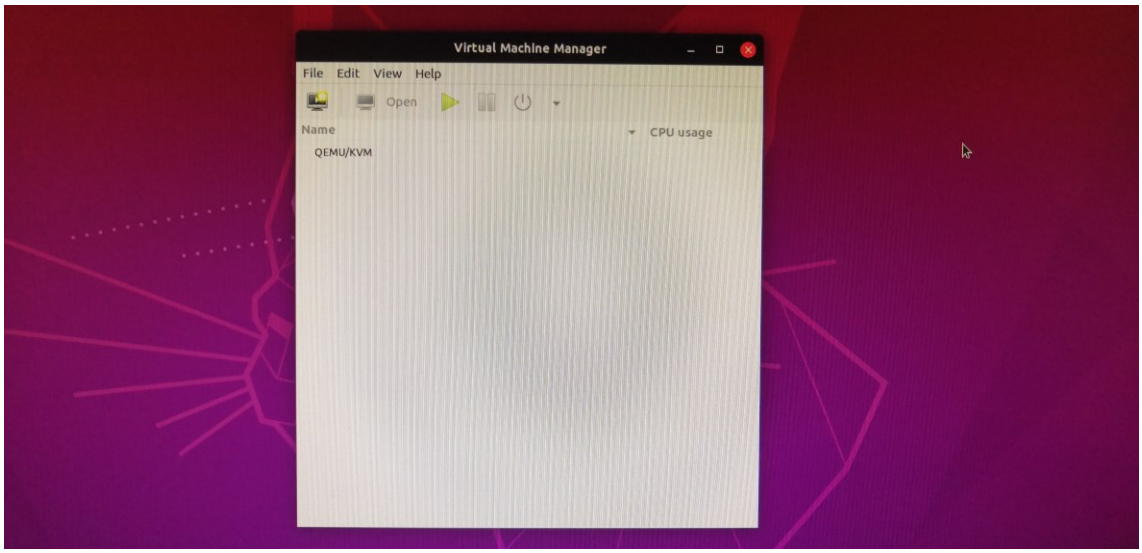

Figure 10 Virtual Machine Manager

7.2 KVM and QEMU

QEMU is an open-source software which is licensed under the GNU's General Public License. It can be an emulator or virtualizer. Its role as an emulator is dedicated to run operating systems and software that are designed to work on certain system to operate on another system. [13.]

When QEMU runs as virtualizer, it turns into hypervisor type 2 that can operate independently and runs inside user space. Its task is to supply virtual hardware emulation, but it is slow due to its emulation process that occurs entirely in software. KVM role comes here to accelerate the virtualization process through utilizing from the physical CPU virtualization extensions. [13.]

7.3 Libvirt and Management Tools

Libvirt is an open-source toolkit that is used for virtualization platforms management. It supports the functionality of many projects such as KVM, QEMU, Xen and VMWare. [14.] The following tools are the most used ones to manage KVM VMs, and they are:

1. Virt-manager: it is a desktop application that can manage virtual machines locally and remotely. It is dedicated for home and small-sized networks that may include up to 20 hosts. [14.] A Debian Wheezy 7 VM is going to be created and managed by using this tool in subchapter 7.4.

2. oVirt: it is a free open-source web-based virtualization solution that manages virtual machines in bulk within enterprise level environments such as data centers. [15.] oVirt is a centralized virtualization solution that is made up of two main parts:

   - The agents that perform the communication role with hosts.

   - The engine that has a GUI interface and offers advanced management options such as export and import of virtual machines (OVF format), Resource monitoring and Virtual-to-virtual conversion (V2V). [13.] Figure 11 illustrates oVirt interface.

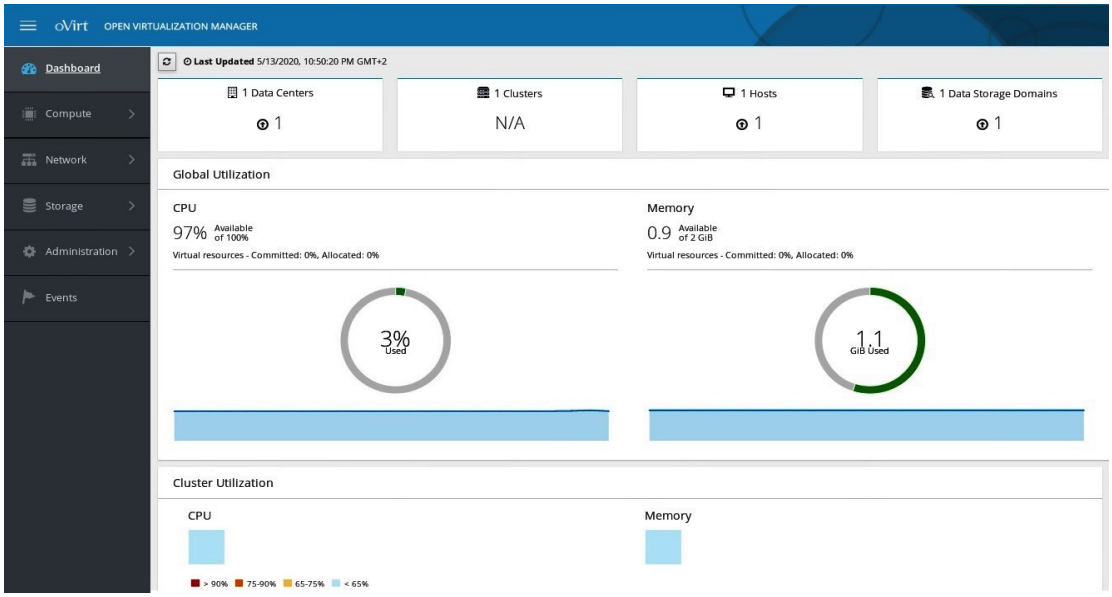

Figure 11 oVirt Dashboard

7.4 Creating Virtual Machine on KVM

A Debian Wheezy 7 VM is going to be created by following these steps:

- Opening Virtual Machine Manager.
- Clicking on the Create a new virtual machine (PC icon under File).
- Selecting the way how to install the VM operating system software.
- Clicking on Forward will open a window that asking about the location of installation media (ISO image in this creation context) and the operating system name (4 GB of memory and 1 CPU were chosen).
- Clicking on Forward again to choose memory and CPUs assigned to the VM.
- Forward button once more will lead to the creation of VM disk image (a disk image with 20 GB is assigned).
- Clicking Finish will initiate the creation of VM. [16.] Figure 12 illustrates VM creation process initiation.

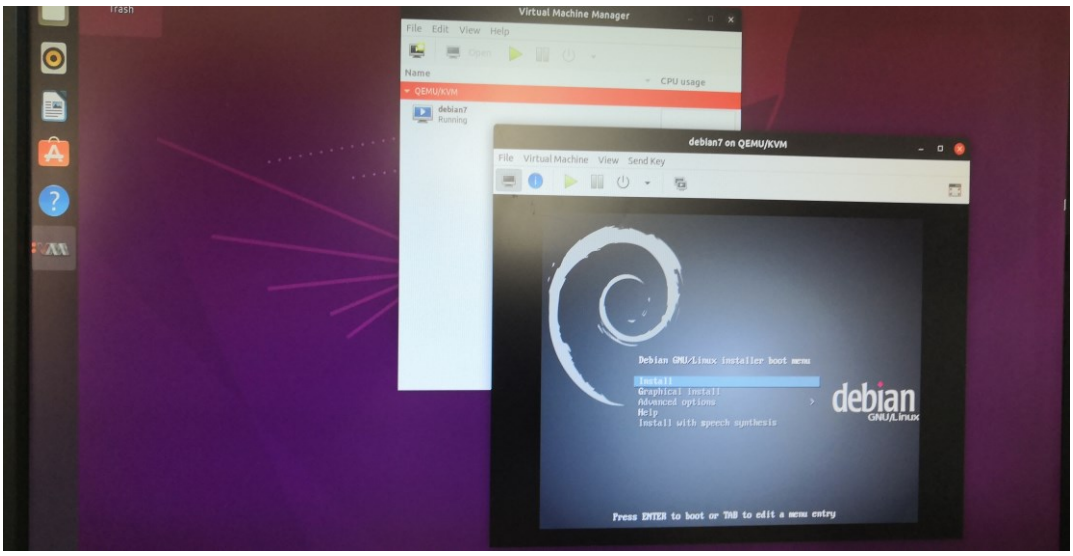

Figure 12 Virt-Manager VM Creation Process Initiation

The default network type (NAT), through which guest operating systems communicate, limits the network communication capabilities, because the virtual network is mapped to host network to provide internet connectivity. So, it is important to create a network bridge to allow guest operating systems to communicate with the outer world such as creating SSH sessions. [16.]

8 Testing VMs on XenServer and KVM

8.1 Phoronix Test Suite

Phoronix is an open-source performance test software that can be used on different operating systems. It is free and licensed under GNU GPLv3. The Phoronix Test Suite offers over 400 test profiles and more than 100 test suites via OpenBenchmarking.org, that contains private and public storage areas for testing results to compare them to available sets and share them with other users. Testing results are displayed in a web-based viewer with various options for uploading them to OpenBenchmarking.org. Some of the other good options to be mentioned here are exporting results to PDF and running side-by-side performance comparisons. [15.]

8.1.1 Testing Debian Wheezy 7 VM Using Phoronix on XenServer

Phoronix Test Suite gives the possibility to store the testing results directly to openbenchmarking.org. So, it is advised to register for the free account and login to the website by typing inside the Wheezy 7 terminal, before performing the tests, the following command:

 # phoronix-test-suite openbenchmarking-login

Processor Test: many processor tests are available by phoronix to check the performance depending on different techniques. The SciMark is one of the best popular processor tests, because it contains six scientific and numerical computing benchmarks, they are: Jacobi Successive Over-relaxation, Monte Carlo, Sparse Matrix Multiply, Fast Fourier Transform and dense LU matrix factorization.

The SciMark test was initiated by typing the command: *phoronix-test-suite benchmark scimark2* and was performed three times. The best results group was considered. Figure 13 shows the SciMark test results (to be analyzed later in chapter 9).

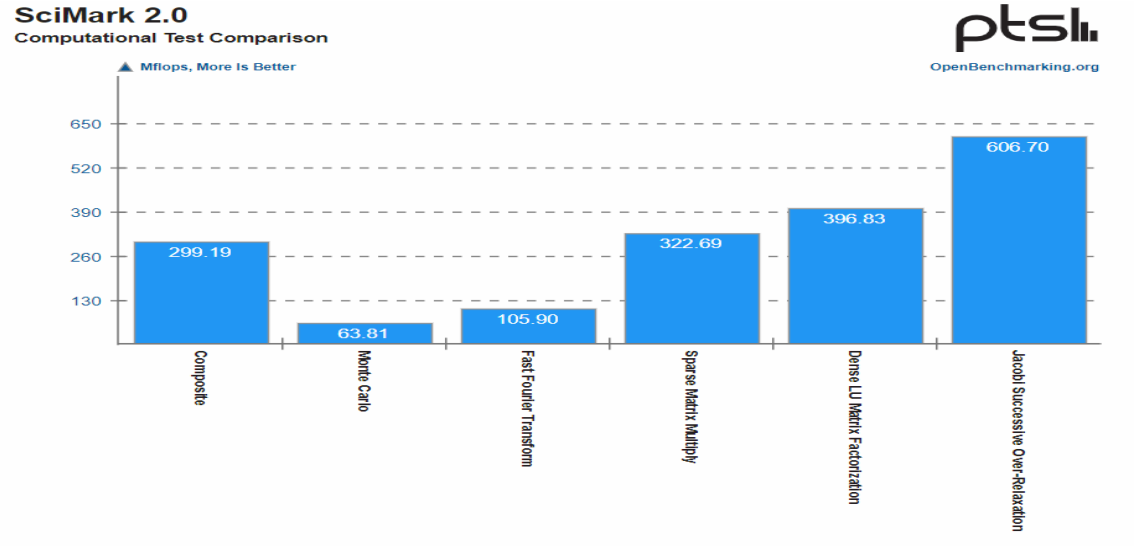

Figure 13 Processor Testing Results on XenServer VM

Memory Test: the RAMspeed SMP is one of the best memory tests offered by phoronix test suite, because it checks the memory performance through applying integer and floating-point benchmarks on the system memory and observe how it operates within five operations: Add, Copy, Scale, Triad and Average.

The RAMspeed SMP test was initiated by typing the command: phoronix-test-suite benchmark ramspeed and was performed three times. The best results group was considered. Figures 14 (integer test) and 15 (floating-point test) show the RAMspeed SMP testing results (to be analyzed later in chapter 9).

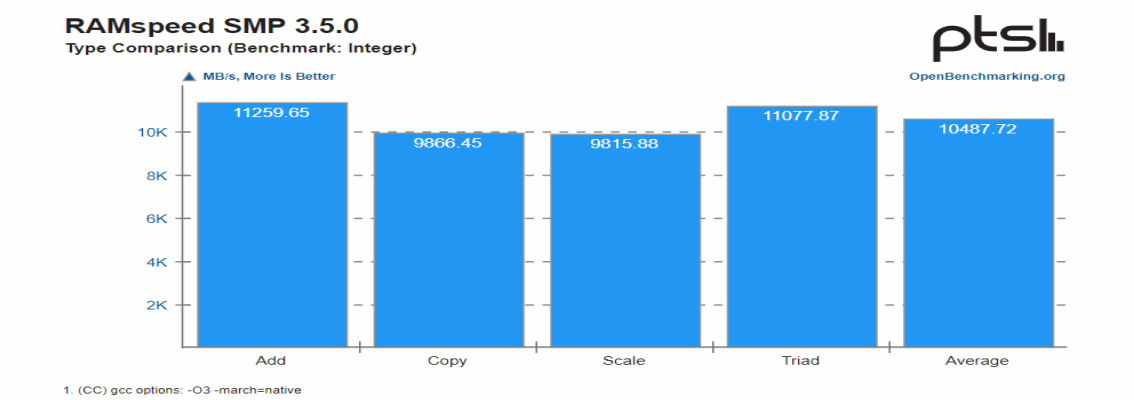

Figure 14 Memory Testing Results on XenServer VM (Integer)

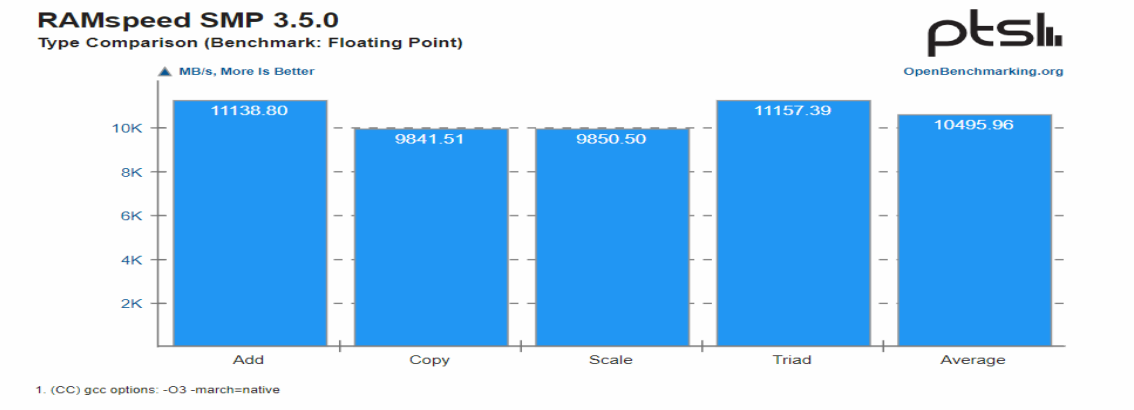

Figure 15 Memory Testing Results on XenServer VM (Floating-Point)

8.1.2 Testing Debian Wheezy 7 VM Using Phoronix on KVM

Processor Test: the same method, steps and tools as in subchapter 8.1.1 were applied again to test Debian Wheezy 7 VM processor performance using Phoronix on KVM in this subchapter. Figure 16 shows the CPU testing results.

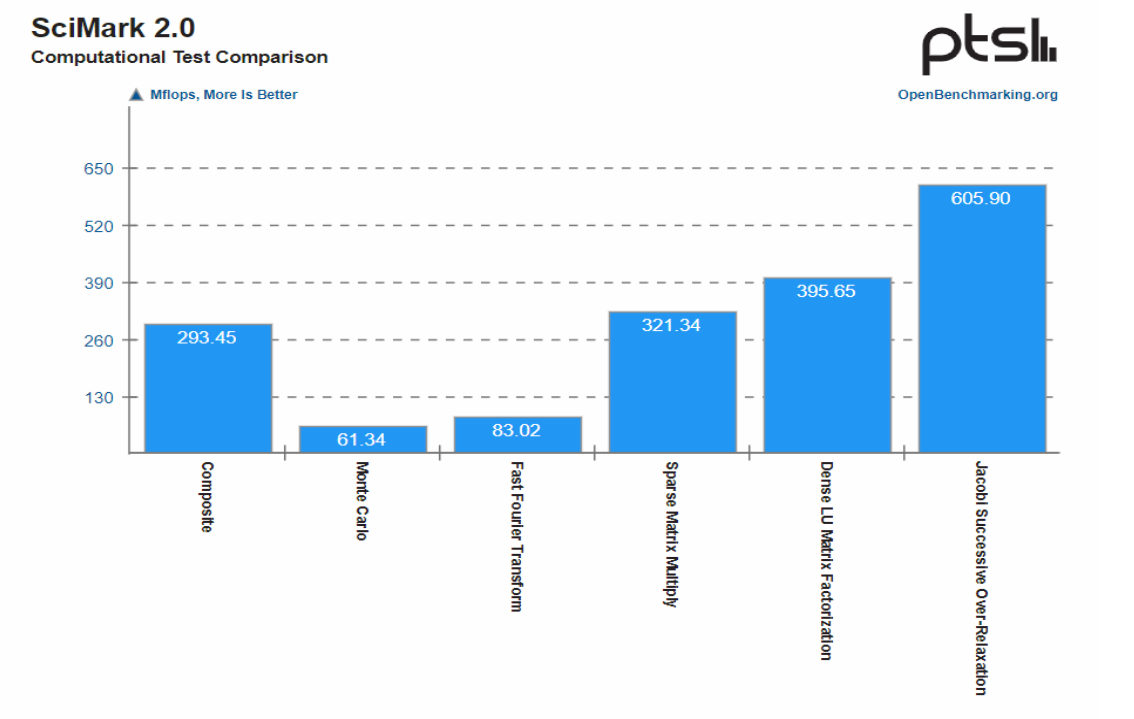

Figure 16 Processor Performance Test Results on KVM Server VM

Memory Test: the same method, steps and tools as in subchapter 8.1.1 were applied again to test Debian Wheezy 7 VM memory performance using Phoronix on KVM in this subchapter. Figure 17 (integer test) and Figure 18 (floating-point test) show testing results.

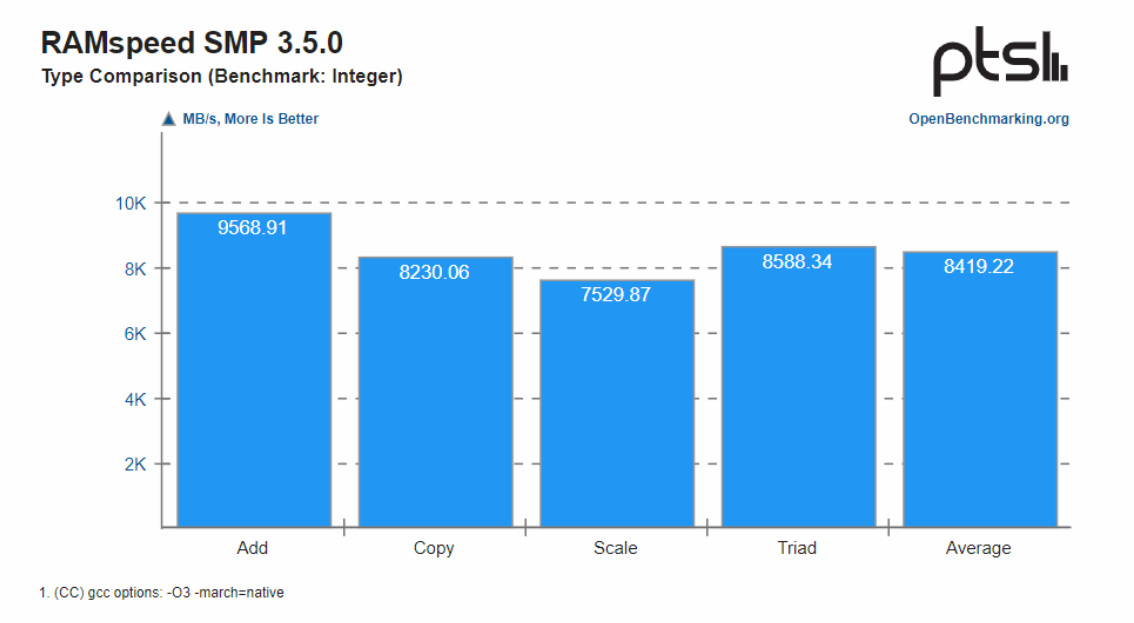

Figure 17 Memory Testing Results on KVM Server VM (Integer)

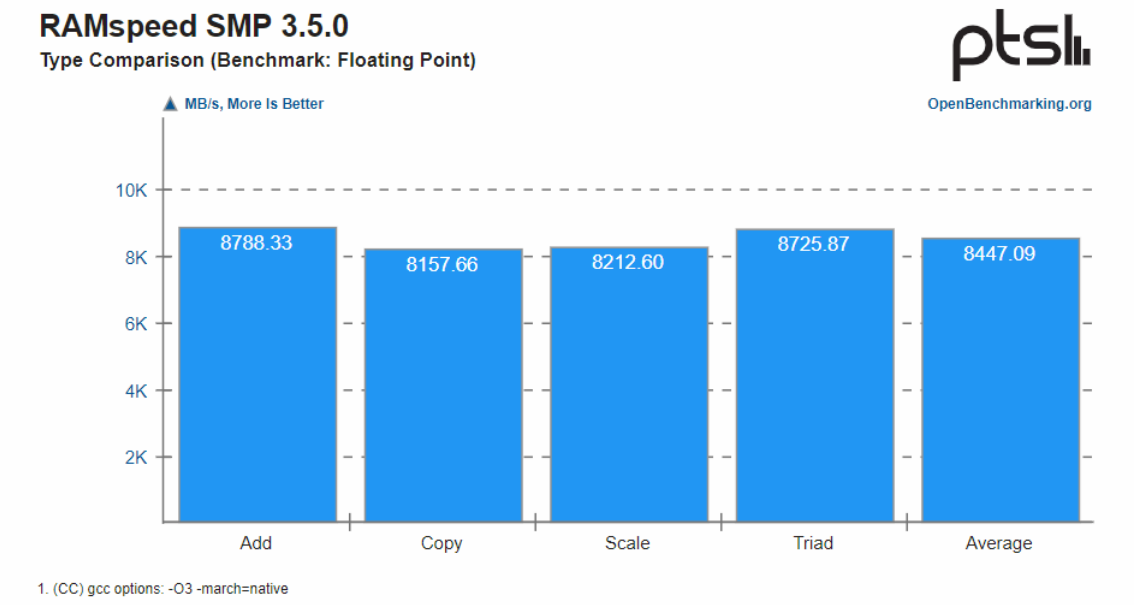

Figure 18 Memory Testing Results on KVM Server VM (Floating-Point)

8.2 iPerf

iPerf is a software that is used for testing the performance of networks. It is supported by many operating systems. iPerf does not have a GUI, but it is easy to be used through the command line interfaces/ terminals. It has the functionality of both client and server. iPerf measures the throughput, loss and latency between two ends of a network by creating data streams in one or two directions. This software is distributed under the 3-Clause BSD License. [16.]

8.2.1 Testing Debian Wheezy 7 VM Using iPerf on XenServer

An Intel Core i5-8265U CPU 1.60GHz with 16 GB RAM, Intel(R) Wireless-AC 9461 and Windows 10 operating system laptop was used to be the client side in this test. Two network performance tests were carried out and each one was repeated five times on both client and server sides (the average value of the results was considered), they are:

1. A TCP performance test was initiated by using the following commands:
   # iperf --server → on Wheezy 7 VM as shown in figure 19.

Figure 19 TCP Performance Test Results on XenServer VM

> iperf --client [server IP address] --time 10 → on client laptop as shown in figure 20.

```
C:\Users\Bashar\Desktop\iperf-2.0.5-win32>iperf --client 192.168.10.242 --time 10
------------------------------------------------------------
Client connecting to 192.168.10.242, TCP port 5001
TCP window size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.10.142 port 63699 connected with 192.168.10.242 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  45.0 MBytes  37.6 Mbits/sec
```

Figure 20 TCP Performance Test Results on Client Laptop

The average results were transfer value = 41.64 Mbytes and bandwidth value = 34.8 Mbits/sec.

2. A UDP performance test was initiated by using the following commands: *# iperf --server --udp* (on Wheezy 7 VM as shown in figure 21).

```
root@debian:~# iperf --server --udp
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  224 KByte (default)
------------------------------------------------------------
[  3] local 192.168.10.242 port 5001 connected with 192.168.10.142 port 53432
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  47.3 MBytes  39.6 Mbits/sec   1.257 ms    0/33711 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
```

Figure 21 UDP Performance Test Results on XenServer VM

> iperf --client [server IP address] --udp --time 10 --bandwidth 100M (on client laptop as shown in figure 22).

```
C:\Users\Bashar\Desktop\iperf-2.0.5-win32>iperf --client 192.168.10.242 --udp --time 10 --bandwidth 100MM
------------------------------------------------------------
Client connecting to 192.168.10.242, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.10.142 port 53432 connected with 192.168.10.242 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  47.3 MBytes  39.6 Mbits/sec
[  3] Sent 33712 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  47.3 MBytes  39.6 Mbits/sec   1.256 ms    0/33711 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
```

Figure 22 UDP Performance Test Results on Client Laptop

The average results were transfer value = 46.48 Mbytes, bandwidth value = 38.96 Mbits/sec, latency = 1.028 ms and packet loss = 0.006%.

8.2.2 Testing Debian Wheezy 7 VM Using iPerf on KVM

The same testing procedures, tools and methods were applied again here to test Debian Wheezy 7 VM network performance on KVM server. The tests and their results were as the following:

1. A TCP performance test returned in the following results:

   The average results were transfer value = 38.81 Mbytes and bandwidth value = 32.3 Mbits/sec.

2. A UDP performance test returned in the following results:

   The average results were transfer value = 40.23 Mbytes, bandwidth value = 35.14 Mbits/sec, latency = 1.513 ms and packet loss = 0.013%.

9 Testing Results Analysis

Processor Performance Test Results Analysis: the results of tests on both XenServer VM and KVM VM CPUs, performed by Phoronix Test Suite software, revealed that paravirtualization represented by XenServer VM had better CPU computational results compared to Full Virtualization represented by KVM VM, though the difference between those results wasn't wide. The measurement unit in this type of tests is MFLOPs which is a common measure of computer performance (computation speed), and it stands for Mega Floating-Point Operations Per Second. Table 1 shows the CPU performance results on XenServer and KVM VMs.

Table 1 Comparison of CPUs Performance Results

Test Part	XenServer VM	KVM VM

Composite	299.19	293.45
Monte Carlo	63.81	61.34
Fast Fourier Transform	105.90	83.02
Sparse Matrix Multiply	322.69	321.34
Dense LU Matrix Factorization	396.83	395.65
Jacobi Successive Over-Relaxation	606.70	605.90

Table 1 reveals that only one test part (Fast Fourier Transform) showed bigger difference in result than other tests whereas the other parts reflected slight difference.

Memory Performance Test Results Analysis: XenServer VM showed higher results regarding memory performance test that performed by Phoronix test suite program. The difference reached in some readings more than 2000 megabytes per second. Table 2 shows the integer testing part results within five operations related to memory functionality.

Table 2 Comparison of Memory Performance Test Results (Integer)

Integer Testing Part		
Test Part	XenServer VM	KVM VM
Add	11259.65	9568.91
Copy	9866.45	8230.06
Scale	9815.88	7529.87
Triad	11077.87	8588.34
Average	10487.72	8419.22

As can be noticed from table 2, memory performance on paravirtualized VM is good remarkably.

The floating-point testing part on XenServer VM is also better than that related to KVM VM. Table 3 shows the difference between the two VMs memory performances regarding floating-point test.

Table 3 Comparison of Memory Performance Test Results (Floating-Point)

Floating-Point Testing Part		
Test Part	XenServer VM	KVM VM
Add	11138.80	8788.33
Copy	9841.51	8157.66
Scale	9850.50	8212.60
Triad	11157.39	8725.87
Average	10495.95	8447.09

The results reached again a difference of more than 2000 MB/s in memory performance between XenServer VM and KVM VM memories as table 3 illustrates.

Network Performance Test Results Analysis: the iPerf network performance test program revealed that network performance on XenServer VM achieved relatively higher scores than the network performance on KVM VM as illustrated in table 4.

Table 4 Network Performance Test Results

	Testing Part	XenServer	KVM
TCP	Transfer	41.64 MB	38.81 MB
	Bandwidth	34.8 Mbits/sec	32.3 Mbits/sec
UDP	Transfer	46.48 MB	40.23 MB
	Bandwidth	38.96 Mbits/sec	35.14 Mbits/sec
	Latency	1.028 ms	1.513 ms
	Packet Loss	0.006%	0.013%

The noticeable differences in network performance testing results, as shown in table 4, are apparent in latency and packet loss values on KVM VM where they are higher than the results on XenServer VM. Consequently, network performance on XenServer VM is better than on KVM VM.

10 Conclusion

Virtualization technology will keep growing and developing due to the ongoing competitions between developers and companies trying to offer perfect IT solutions that meet the requirements of building smart civilization. Being part of this chain, hypervisors gained a great attention and care, because they played a vital role in virtualization world. Different virtualization techniques are available, but this project shed the light on paravirtualization and full virtualization, because they kept the competition on.

This comparative study aimed to examine those two technologies through testing the performance of processors, memories and networks on each virtual machine that was created and managed by different virtual machine manager to collect the results for analysis. Paravirtualization, represented by Citrix XenServer hypervisor, showed better performance results than full virtualization, represented by KVM hypervisor. However, some results were close to each other, and this reveals that KVM community could improve soon the slow performance of full virtualized virtual machines when compared with XenServer paravirtualized ones.

Although the infrastructure of this project worked well, it is recommended to avoid budget squeezing not to waste time and efforts while trying to find free tools and solutions to manage the hypervisors and their virtual machines. This project is suitable for IT learners and internal operations within small companies.

References

1 IBM Cloud Team. A Brief History of Cloud Computing [online]. IBM, 2017.
URL: https://www.ibm.com/cloud/blog/cloud-computing-history
Accessed 13 October 2021.

2 VMware. Virtualization [online]. VMware, 2021.
URL: https://www.vmware.com/solutions/virtualization.html
Accessed 13 October 2021.

3 IBM Cloud Education. Virtualization [online]. IBM, 2019.
URL: https://www.ibm.com/cloud/learn/virtualization-a-complete-guide#toc-
Accessed 16 October 2021.

4 IBM Cloud Education. Hypervisors [online]. IBM, 2019.
URL: https://www.ibm.com/cloud/learn/hypervisors
Accessed 16 October 2021.

5 Reed, Martez. Mastering Citrix XenServer [e-book]. Packet Publishing, 2014.
URL: https://learning.oreilly.com/library/view/mastering-citrix-xenserver/9781783287390/ch01.html
Accessed 16 October 2021.

6 Citrix Staff. Citrix Hypervisor 8.2 Product Documentation [online]. Citrix, 2021.
URL: https://docs.citrix.com/en-us/citrix-hypervisor/technical-overview.html
Accessed 17 October 2021.

7 Vora, Zeal. Enterprise Cloud Security and Governances [e-book]. Packet Publishing, 2017.
URL: https://learning.oreilly.com/library/view/enterprise-cloud-security/9781788299558/431f9eb3-8034-4b20-95b7-23ac505b7bd3.xhtml
Accessed 17 October 2021.

8 Crawford, Luke S; Takemura, Chris. The Book of Xen [e-book]. No Starch Press, 2009.
URL: https://learning.oreilly.com/library/view/the-book-of/9781593271862/.
Accessed 18 October 2021.

9 Xen Orchestra. XO Documentation [online]. 2021.
URL: https://xen-orchestra.com/docs/installation.html#xoa.
Accessed 18 October 2021.

10 RedHat. What is KVM? [online]. 2021.
URL: https://www.redhat.com/en/topics/virtualization/what-is-KVM.
Accessed 21 October 2021.

11 VMware. Understanding Full Virtualization, Paravirtualization, and Hardware Assist [online]. 2008.
URL:https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf
Accessed 21 October 2021.

12 KVM contributors, "FAQ," KVM. 2020.
URL:https://www.linux-kvm.org/index.php?title=FAQ&oldid=174019
Accessed 24 October 2021.

13 Dakic, Vedran; Devassy Chirammal, Humble; Mukhedkar, Prasad; Vettathu, Anil. Mastering KVM Virtualization - Second Edition [online-book]. Packt Publishing, 2020.
URL:https://learning.oreilly.com/library/view/mastering-kvm-virtualization/9781838828714/
Accessed 26 October 2021.

14 Libvirt. Libvirt virtualization API [online]. 2021
URL: https://libvirt.org/
Accessed 26 October 2021.

15 oVirt. Official Website [online]. 2021
URL: https://www.ovirt.org/
Accessed 26 October 2021.

16 Smyth, Neil. Ubuntu 20.04 Essentials [e-book]. Packt Publishing, 2020.
URL:https://learning.oreilly.com/library/view/ubuntu-20-04-essentials/9781800568525/
Accessed 26 October 2021.

17 Phoronix Test Suite. Phoronix Media [online]. 2021
URL: https://www.phoronix-test-suite.com/
Accessed 28 October 2021.

18 iPerf. Official Website [online]. 2021
URL: https://iperf.fr/
Accessed 28 October 2021.