Virtualization of Data Centers
Case Study on Server Virtualization

Sri Kasyap Kappagantula
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Telecommunication Systems. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Sri Kasyap Kappagantula
E-mail: srka16@student.bth.se

University advisor:
Dr. Adrian Popescu
Department of Computer Science (DIDD)

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone   : +46 455 38 50 00
Fax     : +46 455 38 50 57
Abstract

Context. Now-a-days, Data Centers [7] use Virtualization, as a technique to make use of the opportunity for extension of independent virtual resources from the available physical hardware. Virtualization technique is implemented in the Data Centers to maximize the utilization of physical hardware (which significantly reduces the energy consumption and operating costs) without affecting the QoS [13].

Objectives. The main objective of this thesis is to study, the different network topologies used in the Data Center architecture, and to compare the QoS parameters of virtual server over physical server and also to abstract the better technology exists for virtualization.

Methods. The research methodology used in this thesis is qualitative. To measure the QoS, we take the Latency, Packet loss, Throughput aspects of virtual servers under different virtualization technologies (KVM, ESXi, Hyper-V, Fusion and Virtualbox) and compare their performance over the physical server. The work also investigates the CPU, RAM Utilizations and compare the physical and virtual servers behavior under different load conditions.

Results. The Results shows us that the virtual servers has performed better in terms of resource utilization, Latency and response times when compared over the physical servers. But there are some factors like backup and recovery, VM Sprawl, capacity planning, building a private cloud, which should be addressed for growth of virtual data centers.

Conclusions. Parameters that affect the performance of the virtual servers are addressed and the trade-off between the virtual and physical servers is established in terms of qos aspects. The overall performance of virtual servers is effective when compared over the performance of physical servers.

Keywords: Internet, Servers, System Performance, Throughput, Virtualization.
Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof. Adrian Popescu. This work would not be possible without his continuous support, insights and patience. It was my great pleasure working under him.

I am grateful to my parents, for their endless encouragements, love and support at all times. They have done their best in every way possible to help me complete my studies here abroad. Without their blessings and support this achievement would not have been possible.

I am thankful to Prof. Patrik Arlos for sharing his thoughts and advices that helped me a lot for the completing the thesis work.

I would like to thank my thesis examiner Prof. Kurt Tutschku for his valuable suggestions, which helped me in improving this thesis work.

I would also like to thank my friends for encouraging and supporting me in my work.
# Contents

Abstract

Acknowledgements ii

List of Figures iv

List of Tables v

List of Abbreviations vii

1 Introduction 1

1.1 Overview of Data Center Architecture ......................... 1

1.1.1 Three-Tier DCN ..................................... 2

1.1.2 Fat-Tree DCN ...................................... 3

1.1.3 D-Cell DCN ...................................... 4

1.2 Introduction to Virtualization ................................. 5

1.2.1 Hypervisors ....................................... 6

1.2.2 Summary of existing virtualization technologies .......... 7

1.3 Problem Description ..................................... 8

1.4 Motivation ............................................ 8

1.5 Scope of Virtualization in Data Centers .................... 9

1.6 Outline: ............................................... 10

1.7 Research Questions ................................... 10

2 Related Work 11

2.1 Performance of virtual environments ....................... 11

2.2 Dynamic allocation of resources .......................... 12

2.3 Further studies on virtualization .......................... 13

3 Method 14

3.1 Overview of the Applied methods .......................... 14

3.2 Quality of Service .................................... 15

3.3 Discussing network protocols and the various types of tools used in the thesis work: ........................................... 16
List of Figures

1.1 Timeline of DCN Topologies ........................................... 2
1.2 Three-Tier Data Center Network ................................. 3
1.3 Fat-Tree Data Center Network ................................. 4
1.4 D-Cell Data Center Network ................................. 5
1.5 Type 1 and Type 2 Hypervisors ................................. 6

3.1 Experimental set-up using virtualization ......................... 18

4.1 Latency of virtualization technologies under normal and load conditions .......................................................... 24
4.2 Packet Loss of virtualization technologies under load conditions .......................................................... 25
4.3 CPU utilization of virtualization technologies under different loads .......................................................... 27
4.4 RAM utilization of virtualization technologies under different loads .......................................................... 28
4.5 Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 10 .......................................................... 28
4.6 Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 15 .......................................................... 29
4.7 Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 20 .......................................................... 29
List of Tables

4.1 Average Latency Time of Different Virtualization Technologies . 25
4.2 Average Packet Loss under Different Virtualization Technologies . 26
4.3 CPU Utilization under Increasing load . . . . . . . . . . . . . . . 26
4.4 RAM Utilization under Increasing load . . . . . . . . . . . . . . . 26
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD</td>
<td>Berkeley Software Distribution</td>
</tr>
<tr>
<td>BCN</td>
<td>Bidimensional Compound Network</td>
</tr>
<tr>
<td>CRAC</td>
<td>Computer Room Air Conditioner</td>
</tr>
<tr>
<td>DCN</td>
<td>Data Center Network</td>
</tr>
<tr>
<td>HDD</td>
<td>Hard Disk Drive</td>
</tr>
<tr>
<td>HCN</td>
<td>Hierarchical Irregular Compound Network</td>
</tr>
<tr>
<td>IDC</td>
<td>International Data Corporation</td>
</tr>
<tr>
<td>KVM</td>
<td>Kernal-Based Virtual Machine</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Cards</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RCP</td>
<td>Remote Copy</td>
</tr>
<tr>
<td>SCP</td>
<td>Secure Copy</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SSD</td>
<td>Solid State Drive</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>VMM</td>
<td>Virtual Machine Manager</td>
</tr>
<tr>
<td>ZB</td>
<td>Zetta Bytes</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview of Data Center Architecture

Data Centers are large centralized locations of computational power, storage, and applications necessary for the purpose of collecting, storing, processing, distributing or allowing access to large amounts of data.

These Data Centers house high concentrations of servers, often stacked in racks, sometimes referred to as server farms [15], which are responsible for the wholesale crunching of zeros and ones. They are commonly used for public cloud-based services at hosts like Amazon, Microsoft, Sony, and Google. Nowadays, many companies are also moving their professional applications to cloud services to cut back on the cost of running their own centralized computing networks and servers.

Data Centers also have lots of equipment, which include fans, air handlers, filters, sensors, CRACs [9] and other equipments, to handle temperature and air quality control, although the methods and types of equipment vary from site to site.

The basic foundation of DCN [8] design, is based on a layered approach. Proper planning of the data center infrastructure design is critical, and several aspects like performance, flexibility in deploying and supporting new services, scalability and resilience need to be considered carefully. The network topologies used in the Data Center architecture are Three-Tier, Fat-Tree, D-Cell, BCube, MDCube, Scafida, Jellyfish, HCN and BCN. The common architectures such as Three-Tier, Fat-Tree and D-Cell DCN Topologies are discussed here. Figure 1.1 shows the Timeline of the DCN topologies.
1.1.1 Three-Tier DCN

The Three-Tier DCN is the most commonly used architecture in the Data Centers. The Three-Tier DCN encompasses of Core, Aggregation and Access layers. The Core layer is the lower layer in the architecture and provides high speed packet switching and connectivity to the aggregation modules. The Aggregation layer, is the middle layer in the three-tier architecture and aggregates WAN connections at the edge of the campus and provides policy-based connectivity. It also provides important functions such as service module integration, spanning tree processing and default gateway redundancy. Server-to-server multi-tier traffic flows through the aggregation layer and uses services like firewall and server load balancing, to optimize and secure applications. The Access layer is the higher layer in the Three Tier DCN topology and is used to grant user access to the network devices. In the Access Layer the servers are physically attached to the network. This layer generally consists of switched LAN devices with ports that provide connectivity to workstations, IP phones and servers. Figure 1.2 shows the architecture of the Three-Tier DCN.
Chapter 1. Introduction

1.1.2 Fat-Tree DCN

The Fat-Tree DCN design incorporates the low cost Ethernet commodity switches to form a k-ary fat-tree. The fat-tree handles the over subscription and cross section bandwidth problem faced by the legacy three-tier DCN architecture. The fat-tree network structure is composed of n pods. Each pod contains n servers and n switches organized in two layers of $n/2$ switches. Every lower layer switch is connected to $n/2$ hosts in the pod and $n/2$ upper layer switches (making aggregation layer) of pod. There are $(n/2)^2$ core switches, each connecting to one aggregation layer switch in each of n pods. The network elements in fat-tree architecture also follows hierarchical organization of network switches in access, aggregate, and core layers. However, the number of network switches is much larger than the three-tier DCN. The fat-tree architecture uses a customized routing protocol, which is based on primary prefix and secondary suffix lookup for next hop. The scalability is one of the major issues in fat-tree architecture and maximum number of pods is equal to the number of ports in each switch. Figure 1.3 shows the architecture of the Fat-Tree DCN.
1.1.3 D-Cell DCN

DCell [20], is a server-centric hybrid DCN architecture where one server is directly connected to many other servers. A server in a DCell is equipped with multiple NICs. A cell0 is the basic unit and building block of DCell topology arranged in multiple levels, where a higher level cell contains multiple lower layer cells. A cell0 contains $n$ servers and one commodity network switch. The network switch is only used to connect the server within a cell0. A cell1 contains $k = n + 1$ cell0 cells, and similarly a cell2 contains $k \times (n + 1)$ cell1. A Dcell can be built recursively resulting in more than 3.26 million servers with an average diameter of less than 10 ($k=3, n=6$). Routing in Dcell follows a divide and conquer approach. For packets to reach from a source host to destination host, it needs to traverse from source to common ancestor DCell, a link connecting the previous level DCells and finally, to the destination. The exact path can be found similarly in a recursive fashion. The protocol is further extended to implement fault tolerant routing (DFR) to cope with link or node failures. Overall, DCell is highly scalable and fault tolerant topology however, it provides low bisection bandwidth. Figure 1.4 shows the architecture of the D-Cell DCN.
1.2 Introduction to Virtualization

In today’s world, with the dawn of Internet, people are constantly online, using this Internet as a powerful tool for “Information Interchange”, for accessing the “Cloud Services” such as data storage, backup and recovery, hosting websites, Instant Messaging (IM), online Games, E-mail services and many more things. As an estimation given by IDC [19], approximately 2.8 ZB of Digital Information was created in the year 2012 and is expected that to an increase of 40 ZB by the year 2020. This massive increase of the Digital Information Exchange leads to the need for concentrations of computers and networking equipment that can process up these requests and serve the goods securely, reliably and quickly. Thus the “Data Centers” were born.

Besides of having enormous benefits with these Data centers, there also rest some factors like cost, space, equipment, power consumption, cooling facilities that pushes up the growth of Data Centers backwards. This is where the Virtualization Technology [16] came into play. Using the Virtualization technology, there can be an effective usage of power, cooling and networking equipment.
Virtualization Technology mark its impact by promise opportunities for modern data centers to host applications on shared infrastructure. With the help of Virtualization, data center operators can create large number of VMs [27] - a software contrast that mimics the behavior of physical machine, for different workloads. The operators then consolidate all the VMs into smaller number of physical servers, aiming at minimizing the total required physical server number.

### 1.2.1 Hypervisors

A hypervisor, also referred as Virtual Machine Manager (VMM), is a computer software, firmware or hardware that is used to create multiple “Vms” to share a single hardware platform. The hypervisor separates the OS from the hardware by taking the responsibility of allowing each running OS time with the underlying hardware. It acts as a traffic cop to allow time to use the CPU, memory, GPU, and other hardware. Each operating system controlled by the hypervisor is called a guest OS, and the hypervisor’s os, if any, is called the host OS. There are two different types of hypervisors - Type-1 and Type-2. Figure 1.5 shows the different types of hypervisors.

![Figure 1.5: Type 1 and Type 2 Hypervisors](image_url)

**Type-1 Hypervisors:**
Type-1 (also known as Bare-metal) hypervisors are software systems that run directly on the host’s hardware to control the hardware and to monitor guest operating-systems. A guest operating system thus runs on another level above the hypervisor. Some of the type-1 hypervisors used in this study are VMWare ESXi, Microsoft Hyper-V and KVM.
Chapter 1. Introduction

Type-2 Hypervisors:
Type-2 hypervisors are software systems that run on a conventional OS just as other computer programs do. A guest operating system runs as a process on the host. Type-2 hypervisors abstract guest operating systems from the host operating system. Some of the type-2 hypervisors used in this study are VMWare Fusion and Oracle VirtualBox.

1.2.2 Summary of existing virtualization technologies

VMWare ESXi:
VMWare ESXi [2] is a bare metal (type-1) hypervisor, that is built directly on top of an x86 hardware. It abstracts the underlying hardware and allows multiple virtual machines to use the same hardware resources. ESXi is offered in two different types: ESXi Embedded and ESXi Installable, and there is no functional difference between them. Both use the same code and provide us with the same functionality and features depending on the license used.

The two different types of ESXi are explained as follows: ESXi Embedded is available in the OEM format, and it is installed on a USB or an SD card when the hardware is being purchased. It saves the cost of purchasing additional hard drives and saves valuable time for vSphere administrators, as there is no need to install hypervisors. ESXi Installable is a traditional form of installing the hypervisor on a local disk or SAN using an ISO image.

VMWare Fusion:
VMware Fusion [6] is a type-2 hypervisor, that is used to run windows applications and PC-only devices on your Intel-based Mac. We can run multiple operating systems and applications at the same time, along with your Mac applications. The operating systems and applications are isolated in secure virtual machines. VMWare Fusion maps the physical hardware resources to the virtual machine’s resources, so each virtual machine has its own processor, memory, disks, I/O devices and so on. Each virtual machine is the full equivalent of a standard x86 computer, although it is represented in a single file package on the Mac. The operating system of the computer on which you run VMware Fusion is called the "host". Mac OS X and Mac OS X Server are the only hosts supported for VMware Fusion. The virtualized operating system you run inside VMware Fusion is called the "guest".

Virtual Box:
Virtual Box [5] is a free, cross-platform, open source, type-2 hypervisor for x86 computers currently being developed by Oracle Corporation. It creates Virtual Machines on the top of existing Operating system, which is often referred to as host OS.
Hyper-V:
Hyper-V [10], is Microsoft’s hardware virtualization product that is used to create and run a software version of a computer, called a "virtual machine". Each virtual machine acts like a complete computer, running an operating system and programs. When you need computing resources, virtual machines give you more flexibility, help save time and money, and are a more efficient way to use hardware than just running one operating system on physical hardware. Hyper-V runs each virtual machine in its own isolated space, which means we can run more than one virtual machine on the same hardware at the same time to avoid problems such as a crash affecting the other workloads, or to give different people, groups or services access to different systems.

KVM (Kernal-based Virtual Machine):  
KVM [11] is a bare-metal (Type-1), free, open-source, full virtualization solution for Linux on x86 hardware containing virtualization extensions that can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc...

1.3 Problem Description

With the vast advancements in Cloud technology, many small enterprises are moving out their professional applications to cloud services. This tremendous growth of Cloud-based services such as Drive services, online Games, Social Networking, Email services, and other applications pave the need for concentrations of servers and networking equipment (Data Centers) that handles these requests and serve up the goods securely, reliably and quickly. Several problems rises after establishing the Data centers. Most servers within these Data Centers use only a small fraction of their overall processing capabilities and these Data Centers also requires lot of physical space to establish which in turn consumes lot of power and generate large amounts of heat.

1.4 Motivation

The main aim of this thesis is to provide a solution through which there will be effective use of the underlying physical resources without affecting the Quality of Service for the users and reduces the energy consumption and operating costs of the Data Centers. As a study we took the case of "virtualization of servers" in data centers, as the servers in the Data Center alone constitutes for more than
40% of the total energy consumption.

To provide the better solution exists in the Virtualization, we investigate the QoS parameters like Latency, Packet Loss, Throughput, of the servers under different virtualization technologies (KVM, ESXi, Hyper-V, Fusion, VirtualBox) and compare their performance over the physical servers. We also investigate the RAM and CPU utilizations of the virtualized servers and compare them over the physical server under different work load conditions.

1.5 Scope of Virtualization in Data Centers

Virtualization technologies are helping many businesses cut costs, regain control, and allow for greater growth with their infrastructure. The data center industry has moved way beyond server virtualization, and is exploring new avenues to make virtualization an even more powerful platform. Some of the virtualization technologies in which Data Centers is exploring new avenues are

1. Application Delivery
2. Hosted/Virtual/Cloud Desktops
3. Network Virtualization (SDN and NFV)
4. Security Abstraction
5. User Virtualization
6. Storage Virtualization
Chapter 1. Introduction

1.6 Outline:

This thesis paper is structured as follows.

Chapter 1: outlines the important concepts helpful in understanding this thesis paper.

Chapter 2: presents the related work done in the field of Virtualization.

Chapter 3: describes in detail, the methodology of implementation and evaluation of the solution.

Chapter 4: will present you the results obtained through the performance comparisons done in this thesis.

Chapter 5: provide the results of the evaluation along with analysis and discussions.

Chapter 6: provides the conclusion and future work of this thesis work.

1.7 Research Questions

1. What is the impact on Latency and Packet Loss through virtualization of servers when compared to the physical servers?

2. Is there any impact on Disk throughput and response time when using SSD in virtual servers against using HDD in physical server?

3. Is there any impact on CPU utilization and RAM Utilization through Virtualization of servers when compared over physical servers?
2.1 Performance of virtual environments

Ahmadi and Maleki [17], Evaluated the performance of virtualized versus non-virtualized servers in data center applications. They have presented a tile based evaluation based on heterogeneous workloads to compare several key parameters and demonstrated the effectiveness of virtualization techniques. The experimental results that are obtained by tile-based model clearly stated that server virtualization significantly reduced the resource consumption while improved the system performance.

Che et al. [18], measured and analyzed the performance of two open source virtual machine managers - Xen and KVM using LINPACK, LM-bench and IOzone, and provided a quantitative and qualitative comparison of both virtual machine managers. These experimental results shows that context switch, process creating and executing and I/O operation’s virtualization puts mainsource of the total virtualization overhead.

Lee et al. [22], analyzed the network performance of containers-based virtualization on actual IoT devices depending on the network mode. They also evaluated the performance when multiple containers run concurrently and investigated a bottleneck through system-level profiling. The study results clearly identified the performance issues of containers for better IoT device management.

Shirinbad et al. [26], compared the performance of KVM, VMWare and XenServer with different parameters like CPU Utilization, Disk Utilization, average response time and downtime of a large real-time telecommunication application under two different scenarios. The results of these performance tests show that ESXi and KVM has performed better in terms of CPU utilization, application response time when compared with XEN.
Umeh et al. [29], evaluated the throughput and delay performance of IEEE 802.11 WLANs using different performance monitoring tools (NetStress, Wireshark and Jperf) and rooted the effects of varying the number of work stations over the network performance in the real WLAN environments. The study results proved that maximum throughput and minimum delay can only be achieved in the non-saturated case and is clearly dependent on the number of active nodes.

Wang and Wang [30], introduced "a performance controlled power optimization solution for virtual data center to achieve power efficiency and application-level performance assurance". At the application level, a MIMO controller is proposed to achieve the required reaction time for the applications crossing numerous VMs, on a short-lived span scale by reallocating the CPUs and other assets.

### 2.2 Dynamic allocation of resources

Shirinbad et al. [24], have done performance measurement on different distributed storage solutions during I/O operations and also compared the recovery time of the solutions in case of storage server failures. By re-allocating the number of virtual CPUs, performance of different hypervisors has been calculated in terms of downtime and total migration time. They proposed an overhead model which considers the effect of overhead during live migration from one machine to another.

Shirinbad and Lundberg [25], successfully built a testbed using VMWare virtualization technology and compared the performance Implications of Over-allocation of virtual CPUs under different scenarios. They have used a large industrial telecommunication application and measured the performance of the application under different conditions. The results help virtual environment service providers to decide the allocation of virtual CPUs in a proper way to gain high performance and high resource utilization within a cloud environment.

Wei et al. [31], discussed about the dynamic allocation of server CPU, Memory and other resources based on application load. As virtualized servers can be initiated, and can be restarted inside a constrained time to make a critical server, virtualization server in a short span has rapidly become one of the most financially savvy arrangements of disaster recovery and administration abilities.
2.3 Further studies on virtualization

Junhao and Aili. [21], Studied the particular applications of virtualization technology and the difficulties it will confront in realistic High Performance Computing (HPC). The study results shows that the probabilities and challenges of virtualization that might confront in HPC are rising effectively.

Liu et al. [23], investigated the "current trends of Green IT awareness and how the deployment of IT equipment optimization techniques can offer a solution to the global issue by reducing the carbon emissions". They also compared the power consumption of physical servers under different processing loads and also observed the implication of virtual servers on power consumption under different load conditions.

Sukmana et al. [28], demonstrated the use for the "Implementation of Server Consolidation Method on a Data Center by using the Virtualization Technique". The work was done in two methods, "the data collection" method and "the Server Consolidation" method. The results show us that with the use of server consolidation technique in the Data Centers, the energy consumption is reduced by more than 35 percent, the man power and time for managing the Data Center has reduced significantly.
Chapter 3

Method

3.1 Overview of the Applied methods

The methodology followed in this thesis work are listed below and will be described in detail in the following sections.

- Explaining the QoS parameters that have been used in this thesis work for the performance comparison of the virtual servers over the physical server.

- Discussing the various types of tools and network protocols used in the thesis work.

- Overview of the small telecommunication project used in this thesis study.

- Overview of the experimental setup used in this thesis work.

- Explaining the method for calculating the latency, Loss characteristics and Load Testing which are used in this thesis to compare the performance of the physical and virtual servers.

- Explaining the method for calculating the CPU and RAM Utilization of the servers.
3.2 Quality of Service

Nowadays, one of the most important goals of data center management is to maximize their profit by minimizing power consumption and service-level agreement violations of hosted applications. While power consumption must be minimized, one of the important requirements for a data center is to provide reliable "QoS" - which is defined as the measurement of the overall performance of a service, such as a telephony or computer network or a cloud computing service, required by customers. It can be defined in terms of SLAs that are part of customer commitments and describe such the key performance metrics as minimal throughput, maximal response time. Although modern virtualization technologies can ensure performance isolation between VMs on the same physical nodes, due to aggressive consolidation and variability of workloads some VMs may not get the required amount of resource when requested. This will lead to performance loss in terms of increased response time, time outs or failures in the worst case. Therefore, it is important to guarantee the application QoS while minimizing the power consumption of data centers. This thesis work investigates the following performance metrics of virtual servers and compare them over the physical server.

**Latency:**
In a computer network, Latency (also referred as Round Trip Time) is the amount of time, required for a packet to be return back to its sender. Latency depends on the speed of the transmission medium and the delays in the transmission by devices along the way. A low latency indicates a high network efficiency.

**Packet Loss:**
Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is typically caused by network congestion. Packet loss is measured as a percentage of packets lost with respect to packets sent.

**Load Testing:**
Load testing (also referred to as stress testing) [12], is the process of putting demand on a software system or computing device and measuring its response. Load testing is performed to determine a system’s behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation. When the load placed on the system is raised beyond normal usage patterns to test the system's response at unusually high or peak loads.

Load testing let us measure web-server quality of service (QoS) performance based on actual users behavior.
3.3 Discussing network protocols and the various types of tools used in the thesis work:

SNMP:
Simple Network Management Protocol (SNMP) is a popular protocol for network management. It is used for collecting information from, and configuring, network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP) network.

PING:
Packet Internet Grouper (ping) is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network. It measures the round-trip time (or Latency) for messages sent from the originating host to a destination computer that are echoed back to the source.

Apache Bench:
ApacheBench [1] is free, open source, single-threaded command line computer program for measuring the performance of HTTP web servers. Originally designed to test the Apache HTTP Server, it is generic enough to test any web server.

IPerf:
Iperf [3] is a open source, cross-platform, widely-used tool for network performance measurement and tuning. It can produce standardized performance measurements for any network. Iperf has client and server functionality, and can create data streams to measure the throughput between the two ends in one or both directions.

SCP:
The SCP [14] is a network protocol, based on the BSD RCP protocol, which supports file transfers between hosts on a network. SCP uses Secure Shell (SSH) for data transfer and uses the same mechanisms for authentication, thereby ensuring the authenticity and confidentiality of the data in transit. A client can send (upload) files to a server, optionally including their basic attributes (permissions, timestamps). Clients can also request files or directories from a server (download). SCP runs over TCP port 22 by default.
3.4 Testbed Implementation

Two servers of 64 bit Architecture has been used to test the performance of KVM, Hyper-V, ESXi, Fusion and VirtualBox. On the top of each virtualization technologies, we have deployed Ubuntu 16.04 server version as a guest OS. All devices are located in a same Wireless Local Area Network (WLAN) as shown in the figure 3.1. Each vm shown in the figure 3.1 has been installed with LAMP [4], SNMP, OpenSSH, Iperf and Apache Bench (ab). Figure 3.1 shows the Experimental Setup of this thesis study.

VM-1 (has the static IP address of "192.168.0.117") is equipped with

- 4 GB RAM
- 4-core CPU
- 40 GB disk (SSD)

VM-2 (has the static IP address of "192.168.0.109") is equipped with

- 1 GB RAM
- 1-core CPU
- 20 GB disk (SSD)

User Server (has the static IP address of "192.168.0.9") is configured with

- 5 GB RAM
- 2-core processor
- 100 GB disk (HDD)

Physical Server (has the static IP address of "192.168.0.118") is configured with

- 4 GB RAM
- 4-core CPU
- 100 GB disk (HDD)
3.5 Brief overview of the small telecommunication application

In order to collect real time data, we have set up a small industrial telecommunication application, SDrive, which is a LAN based file storage service on the servers. With SDrive users can make use of the service functionalities like upload, view, download and deletion of files. SDrive provides High Availability of service to its customers using distributed file servers, and with the use of Self Signed SSL certificates SDrive ensures that all the communication between the user and SDrive is encrypted.

3.6 Method for calculating the Latency, Packet Loss and Load testing the servers:

Latency Test and Packet loss:
we have calculated and compared the Latency for the servers under different virtualization technologies over physical server using PING utility. For testing the performance of servers under virtualization, we have pinged from the user to the server1 (vm1) continuously and took the latency average for every 15 minutes interval. Simultaneously for every 60 minutes interval we have generated load on the server2 (vm2) in order to monitor the latency of server1(vm1) under load condition. Similarly the latency tests has been carried out on the physical server. The whole test held for a day and the results are then monitored and tabled. Then a graph is plotted against the number of packets sent to the server versus round trip time of the packets for each of the virtualization technologies tested and the physical server. The Virtualization technologies used for testing of the servers are KVM, Hyper-V, ESXi, Fusion and Virtual Box.
We have calculated the Packet Loss based on the ratio of number of packets sent and the number of packets received. Packet Loss has been calculated (in percent) by using the formula shown below for every 15 minutes interval. The script shown in listing 3.1 is used for conducting the Latency and Packet Loss tests on the servers.

\[
Loss = (1 - (\frac{\text{Number of Packets Received}}{\text{Number of Packets Sent}})) \times 100
\]

Load testing:
We have done the load testing to compare the performance of Apache HTTP web server in the virtual servers over the physical server. We have used the Apache Bench (ab), a benchmarking tool for conducting the load test of different virtualization technologies under different load conditions. We have changed the concurrent number of users from 10, 15, 20 and measured the web server response for both the virtual servers and the physical. After conducting the load test, we tabled the results and graphs are plotted against the "number of requests served" by the server versus "Total Time". Here the "Total Time" is the summation of both "Connection Time" and the "Delay Time". We have repeated this load test for some series of times and compared the performance of servers under different virtualization technologies over the physical server. We also compared the throughput of the servers with the results obtained from the load test. Throughput is the amount of data successfully sent (or received) over a period of time and has been measured in Megabits per second (Mbps). The script shown in listing 3.2 is used for conducting the load test of the Apache HTTP web server under different virtualization technologies.

\[
Throughput = \frac{\text{Data Received successfully}}{\text{Time taken to complete the requests}}
\]

3.7 Method for calculating the CPU and RAM utilizations of the servers:
By using virtualization technology, we have deployed two virtual machines (vm1 and vm2) on the same server as shown in figure 3.1, in order to maximize the utilization of the underlying hardware resources. We have compared the CPU and RAM utilizations of the virtual server when both virtual machines are running, using different virtualization technologies over the physical server. The CPU and RAM utilizations are noted under normal and increasing load conditions. The script shown in listings 3.4 and 3.3 are used to get the CPU and RAM utilizations of the virtual and physical servers. We have used the SNMP protocol for getting the information about the CPU and RAM of the servers and calculated the RAM and CPU utilizations using the formula shown below. The Management Information Bases (MIB) that corresponds to Virtual Server Memory, CPU are
used to retrieve the information and to calculate the CPU and RAM utilizations of the Virtual server.

\[
RAM\ (Free) = \frac{RAM\ (Available + inBuffer + inCache)}{RAM\ (Total)}
\]

\[
RAM\ (Utilized) = (1 - \frac{RAM\ (free)}{RAM\ (Total)}) \times 100
\]

\[
CPU\ (Total) = CPU\ (System + User + Idle)
\]

\[
CPU\ (Utilized) = \frac{CPU\ (System + User)}{CPU\ (Total)} \times 100
\]
Listing 3.1: Latency.sh

```bash
#!/bin/bash

Num=1
r=100000
LoadInterval=4
User='whoami'
SerIP="192.168.0.117"  ## server 1 (vm1)
VirtualizationUsed=$1
path="/home/$User/Desktop/thesis/$VirtualizationUsed"
Lfile="$path/latency.csv"
Pfile="$path/PacketLoss.csv"
mkdir $path

###### Latency test ######

while [ $Num -lt 100 ]; do
  check=$(( Num % LoadInterval ))
  if [ $check -eq 0 ]; then
    ## generating load on server2 (vm2)
    ab -t 300 -n $r -c 10 http://192.168.0.109/testing/testfile.txt &
  fi
  printf " Test $Num " >> $path/latency.csv
  ping -c 900 $SerIP >> $path/PingLog.txt
  cat $path/PingLog.txt | grep 'avg' | cut -d'/' -f5 >> $Lfile

###### Packet Loss calculations ######

T='cat $path/PingLog.txt | grep 'trans' | cut -d' ' -f1'
R='cat $path/PingLog.txt | grep 'recei' | cut -d' ' -f4'
PL=$(( 1 - ( R / T ) ))
PLoss=$(( PL * 100 ))
printf " Test $Num Loss $PLoss" >> $Pfile
Num=$(( Num + 1 ))
rm $path/PingLog.txt

done
```
Chapter 3. Method

Listing 3.2: LoadTest.sh

#!/bin/bash

User='whoami'
SerIP="192.168.0.117"
r=100000
VirtualizationUsed=$1
path="/home/$User/Desktop/thesis/$VirtualizationUsed"
url="http://$SerIP/testing/file10.txt"

ab -t 300 -n $r -c 5 -g $path/5.data $url >> $path/load.txt
printf "\n*** \n" >> $path/load.txt

ab -t 300 -n $r -c 10 -g $path/10.data $url
printf "\n *** \n" >> $path/load.txt

ab -t 300 -n $r -c 15 -g $path/15.data $url >> $path/load.txt
printf "\n *** \n" >> $path/load.txt

ab -t 300 -n $r -c 20 -g $path/20.data $url >> $path/load.txt
printf "\n *** \n" >> $path/load.txt

***

Listing 3.3: wrapper.sh

#!/bin/bash

ip="192.168.0.117"
User='whoami'
VirtualizationUsed=$1
path="/home/$User/Desktop/thesis/$VirtualizationUsed"

while :
do
execute='perl CPUandRAM.pl $ip' > output.txt
mem='echo output.txt | cut -d'\t' -f1 | tr -d '[:space:]'
cpu='echo output.txt | cut -d'\t' -f2 | tr -d '[:space:]'
printf "memory = $mem cpu = $cpu\n" >> $Path/system.csv
sleep 30
done
Listing 3.4: CPUandRAM.pl

```perl
#!/usr/bin/perl

use Net::SNMP;

$ip = $ARGV[0];

@list = ("1.3.6.1.4.1.2021.11.9.0", "1.3.6.1.4.1.2021.11.10.0", "1.3.6.1.4.1.2021.11.11.0", "1.3.6.1.4.1.2021.11.12.0", "1.3.6.1.4.1.2021.11.13.0", "1.3.6.1.4.1.2021.11.14.0", "1.3.6.1.4.1.2021.11.15.0");

my ($session, $error) = Net::SNMP->session(
    -hostname => $ip,
    -community=> "public",
    -port => "161",
    -version => 'snmpv2c',
);

$result = $session->get_request(
    -varbindlist => \@list,
);

if (!defined($result)){
    printf("Timeout: %s.\n", $session->error());}

### calculating CPU Utilization ###

$User = $result->{"1.3.6.1.4.1.2021.11.9.0"};
$System = $result->{"1.3.6.1.4.1.2021.11.10.0"};
$Idle = $result->{"1.3.6.1.4.1.2021.11.11.0"};

$cpuUsed = sprintf("%.5f",($User+$System)/($User+$System+$Idle)*100);

### calculating RAM Utilization ###

$memTotal = $result->{"1.3.6.1.4.1.2021.14.1.2021.4.5.0"};
$memAvailable = $result->{"1.3.6.1.4.1.2021.14.1.2021.4.6.0"};
$memCache = $result->{"1.3.6.1.4.1.2021.14.1.2021.4.15.0"};

$memUsage = 1-((memAvailable+memBuffer+memCache)/memTotal);
$memUsed = sprintf("%.5f",($memUsage*100));

print "$memUsed\n$cpuUsed\n";
```
Chapter 4

Results

4.1 Latency:

Figure 4.1 shows the latency of the servers in different virtualization technologies and compared against the physical server. From the figure we can clearly observe the peak values occurred in the latency (round trip time) for VM-1, when we generated load on VM-2. When load is generated on the servers, network congestion occurred, which increased the latency (round trip time) of the packet. The results obtained for the latency test are tabled and shown here "Latency results". Table 4.1 shows the average packet loss of different virtualization technologies considered in this work.

![Latency test](image)

Figure 4.1: Latency of virtualization technologies under normal and load conditions
Table 4.1: Average Latency Time of Different Virtualization Technologies

<table>
<thead>
<tr>
<th>Virtualization Technology</th>
<th>Average Latency Time (in milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>17.10</td>
</tr>
<tr>
<td>KVM</td>
<td>12.86</td>
</tr>
<tr>
<td>ESXi</td>
<td>13.12</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>14.50</td>
</tr>
<tr>
<td>Fusion</td>
<td>18.76</td>
</tr>
<tr>
<td>Virtual Box</td>
<td>16.80</td>
</tr>
</tbody>
</table>

4.2 Packet Loss:

Figure 4.2 shows the packet loss occurred while testing the latency for the physical servers and virtual servers under load conditions. The servers are virtualized using different technologies. To observe the behavior of servers we have generated load on VM-2 for every 60 minutes of time interval. The results obtained are tabled and shown here "Packet Loss results". Table 4.2 shows the average packet loss of different virtualization technologies considered in this work.

Figure 4.2: Packet Loss of virtualization technologies under load conditions
Table 4.2: Average Packet Loss under Different Virtualization Technologies

<table>
<thead>
<tr>
<th>Virtualization Technology</th>
<th>Average Packet Loss (in percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>1.00</td>
</tr>
<tr>
<td>KVM</td>
<td>1.15</td>
</tr>
<tr>
<td>ESXi</td>
<td>2.25</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>2.95</td>
</tr>
<tr>
<td>Fusion</td>
<td>2.95</td>
</tr>
<tr>
<td>Virtual Box</td>
<td>3.75</td>
</tr>
</tbody>
</table>

4.3 Resource Utilization

The resource utilization has been calculated for the servers under different virtualization technologies and compared them against the physical server. Figures 4.3, 4.4 shows the RAM and CPU utilizations of the physical and virtual servers under increasing load conditions. Tables 4.3, 4.4 shows the average CPU and RAM Utilizations under increasing load conditions. The results obtained while calculating the server utilization are tabled and shown here "Resouse utilization results".

Table 4.3: CPU Utilization under Increasing load

<table>
<thead>
<tr>
<th>Virtualization Technology</th>
<th>Average CPU Utilization (in percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>2.4</td>
</tr>
<tr>
<td>KVM</td>
<td>4.7</td>
</tr>
<tr>
<td>ESXi</td>
<td>6.9</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>6.3</td>
</tr>
<tr>
<td>Fusion</td>
<td>7.4</td>
</tr>
<tr>
<td>Virtual Box</td>
<td>8.9</td>
</tr>
</tbody>
</table>

Table 4.4: RAM Utilization under Increasing load

<table>
<thead>
<tr>
<th>Virtualization Technology</th>
<th>Average RAM Utilization (in percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>24.6</td>
</tr>
<tr>
<td>KVM</td>
<td>9.4</td>
</tr>
<tr>
<td>ESXi</td>
<td>15.2</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>19.0</td>
</tr>
<tr>
<td>Fusion</td>
<td>26.2</td>
</tr>
<tr>
<td>Virtual Box</td>
<td>21.0</td>
</tr>
</tbody>
</table>
Figure 4.3: CPU utilization of virtualization technologies under different loads

4.4 Load Testing of Apache HTTP Web Server:

Figures 4.5, 4.6, 4.7 shows the response of Apache HTTP Web Server under physical and different virtualization technologies. Graphs have been plotted between "Number of requests served" and the "Total Time". Here the Total Time is the summation of the Connection and the Delay Times. The response times are tabled and shown here "Load results - 1" and here Load results - 2".
Figure 4.4: RAM utilization of virtualization technologies under different loads

Figure 4.5: Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 10
Figure 4.6: Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 15

Figure 4.7: Behavior of Apache HTTP web server under different virtualization technologies when concurrent level = 20
Chapter 5

Analysis and Discussion

5.1 What is the impact on Latency and Packet Loss through virtualization of servers when compared to the physical servers?

By virtualizing the servers using technologies like KVM, Hyper-V, ESXi, Virtual Box, Fusion and comparing the latency of these virtual servers over the physical servers, we have shown that the average latency (round trip time) of the packets is lower in case of KVM when compared with other technologies and physical server. KVM also possesses lower latency time when compared under load conditions. KVM leads in the virtualization technologies compared in this thesis study and also has low Packet Loss when compared against the physical server. we can clearly state from the results that, virtualizing the servers in a Data Center with KVM technology, the users will experience better quality of service when compared over other technologies used in this study.

5.2 Is there any impact on Disk throughput and response time when using SSD in virtual servers against using HDD in physical server?

when the hard disk drive in a physical server is replaced with solid state drive in virtual server, we can observe that the disk throughput (amount of data successfully read/write into the disk in a period of time) has been increased comparatively. From the results, it can be shown that while replacing the HDD with SDD in virtual servers, the disk throughput of 10.3 Mbps (physical server) has increased to 15.4 Mbps (Virtual Server under KVM technology). In case of response times, Virtual server has performed better, serving total of 236 requests in less than 4 seconds while physical servers served in 6 seconds with a concurrency level of 10. While increasing the number of concurrent Users from 10 to 15, 15 to 20, the virtual servers has served more number of requests in less time when
compared with physical servers. In the higher load conditions Hyper-V has higher disk throughput when compared over the physical server. By replacing the HDD with SDD in the virtual servers we have achieved high disk throughput and lower response times when compared to traditional HDD in physical servers.

5.3  Is there any impact on CPU utilization and RAM Utilization through virtualization of servers in Data Centers when compared over physical servers in Data Centers?

When compared with physical servers and other virtualization technologies, KVM has efficient resource utilizations. with the results shown in the result section, we can clearly observe that the KVM has comparatively better resource utilization than the Virtual Box, ESXi, Fusion, Hyper-V and physical server. From the experiments we performed in this thesis, we can clearly state that with Virtualization, we can maximize the utilization of underlying physical resources by deploying virtual servers on the existing physical hardware and by using server virtualization, the need for deploying more number of physical servers can be reduced comparatively and thereby reducing the power consumption operation and maintenance costs of the Data Centers.
Chapter 6

Conclusions and Future Work

This thesis work investigated and compared the qos aspects of the virtual servers over the physical server using a small telecommunication application to determine the better solution exists for virtualization of servers in the Data Centers. Performance issues such as Latency, Packet Loss, response time, throughput of the servers are compared under different conditions. Also the investigation is done on CPU and RAM Utilization of the servers. This work has been limited to some of the common virtualization technologies like KVM, ESXi, Hyper-V, VMWare Fusion and Virtual Box.

The results of this thesis lead us to the conclusion that with the use of the virtualization technique in the data centers, the utilization of the underlying physical resources can be maximized surpassingly without compromising the Quality of Service assured to the users. The results have shown us that KVM has performed better in terms of the CPU utilization, Packet Loss, Latency, Throughput and web server load testing when compared with Fusion, ESXi, Workstation and Hyper-V.

This thesis work can be further extended by investigating the other qos aspects like bit rate, transmission delay, availability, downtime of the virtual servers and comparing them over the physical servers. We can also deploy different types of applications and deploy them on to the servers in order to compare the performance of real time applications. Other virtualization technologies such as Open-Stack, XEN, LXC and Docker containers can also be taken into consideration and compare their performance to extract the better technology that exists for virtualization in the Data Center Industry.
   https://httpd.apache.org/docs/2.4/programs/ab.html.

[2] ESXi | Bare Metal Hypervisor.

   https://iperf.fr/.

[4] LAMP.

   https://www.virtualbox.org/.


[12] Load testing, December 2017.


