

**MINI PROJECT II REPORT**

**On**

**Performance Benchmarking  
of Type II Hypervisors**

**Submitted by:**

**Shahrukh khan (181510029)**

**Nitin Agrawal(181510021)**

**Ananya Agrawal (181510004)**

**Abhitript Gautam(171510006)**

**Shivam Singh(181510032)**

**Under the guidance of our Mentor:- Mr. Saurabh Singhal sir**

**Department of Computer Engineering & Applications  
Institute of Engineering & Technology**



**GLA University  
Mathura- 281406, INDIA  
29,June, 2021**



## **DECLARATION**

We hereby declare that the work which is being presented in the Mini Project “**Performance Benchmarking of Type II Hypervisors**”, in fulfillment of the requirements for Mini-Project, is an authentic record of our own work carried under the supervision of **Mr. Saurabh Singhal**, Assistant Professor, GLA University, Mathura.

#	University Roll. no	Section	Class Roll. No	Student's Name	Mobile Number
1	181510029	O	29	Shahrukh Khan	6386267741
2	181510021	O	22	Nitin Agrawal	9084608685
3	181510004	O	05	Ananya Agrawal	8840350822
4	171510006	O	02	Abhitript Gautam	8439205101
5	181510032	O	32	Shivam Singh	8818810031

## **CERTIFICATE**

This is to certify that the project entitled “**Performance Benchmarking of Type II Hypervisors**” carried out in Mini Project is a bonafide work done by **Shahrukh khan (181510029)**, **Nitin Agarwal (181510021)**, **Ananya Agarwal (181510004)**, **Abhitript Gautam (171510006)** and **Shivam Singh (181510032)** and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Sign\_\_\_\_\_

**Mr. Saurabh Singhal**

**(Mentor)**

Sign\_\_\_\_\_

**Mr. Saurabh Anand**

**(IBM Coordinator)**

**Date: 29/06/21**

## **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to **Prof. (Dr.) Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to **Mr. Saurabh Singhal**, Assistant Professor Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Shahrukh khan

Ananya Agarwal

Nitin Agarwal

Abhitrap Gautam

Shivam Singh

## **Abstract**

The virtualization is carried out by the software layer called as the Hypervisor or Virtual Machine Monitor (VMM). Hypervisor is widely used in cloud datacenters. Benchmark is the measurement of best practice performance. Benchmarking is very essential term for the discovery of the best performance given by the particular system. Benchmarking can provides you the external references and the best practices on which to base your evaluations and to design your system processes which can be very useful in finding the gaps in the system to achieve the desired performance. Your target hypervisors are Virtual Box, VMWare workstation as an hosted hypervisors and KVM. Performance analysis of network, cpu, ram, I/O need to carried out.

# Table of Contents

---

Declaration and certificate	ii
Acknowledgement	iii
Abstract	iv
<b>1. Introduction</b>	<b>1-5</b>
1.1 Overview of virtualization	1
1.2 How does virtualization works?	1-2
1.3 Types of virtualization	2-3
1.4 Hypervisor	3
1.5 Types of Hypervisors	3-4
1.6 Choosing the right Hypervisor	4-5
<b>2. Software Requirement Analysis</b>	<b>6-9</b>
2.1 Overview of Virtual Machine	6-7
2.2 About Linux	7-8
2.3 Brief about ISO File	8-9
<b>3. Software Design</b>	<b>10-18</b>
3.1 Stress Testing	10-14
3.2 Monitoring Linux Server by glances	14-18
<b>4. Result</b>	<b>19-20</b>
<b>5. Conclusion</b>	<b>21</b>
<b>6. References</b>	<b>22</b>

# 1. INTRODUCTION

## 1.1 OVERVIEW OF VIRTUALIZATION

Virtualization is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system. Software called a hypervisor connects directly to that hardware and allows you to split 1 system into separate, distinct, and secure environments known as virtual machines (VMs). These VMs rely on the hypervisor's ability to separate the machine's resources from the hardware and distribute them appropriately. Virtualization helps you get the most value from previous investments.

The physical hardware, equipped with a hypervisor, is called the host, while the many VMs that use its resources are guests. These guests treat computing resources—like CPU, memory, and storage—as a pool of resources that can easily be relocated. Operators can control virtual instances of CPU, memory, storage, and other resources, so guests receive the resources they need when they need them.

In more practical terms, imagine you have 3 physical servers with individual dedicated purposes. One is a mail server, another is a web server, and the last one runs internal legacy applications. Each server is being used at about 30% capacity—just a fraction of their running potential. But since the legacy apps remain important to your internal operations, you have to keep them and the third server that hosts them, right?

Traditionally, yes. It was often easier and more reliable to run individual tasks on individual servers: 1 server, 1 operating system, 1 task. It wasn't easy to give 1 server multiple brains. But with virtualization, you can split the mail server into 2 unique ones that can handle independent tasks so the legacy apps can be migrated. It's the same hardware; you're just using more of it more efficiently.

Keeping security in mind, you could split the first server again so it could handle another task—increasing its use from 30%, to 60%, to 90%. Once you do that, the now empty servers could be reused for other tasks or retired altogether to reduce cooling and maintenance costs.

## 1.2 HOW DOES VIRTUALIZATION WORKS?

Software called hypervisors separate the physical resources from the virtual environments—the things that need those resources. Hypervisors can sit on top of an operating system (like on a laptop) or be installed directly onto hardware (like a server), which is how most enterprises virtualize. Hypervisors take your physical resources and divide them up so that virtual environments can use them. Resources are partitioned as needed from the physical environment to the many virtual environments. Users interact with and run computations within the virtual environment (typically called a guest machine or virtual machine). The virtual machine functions as a single data file. And like any digital file, it can be moved from one computer to another, opened in either one, and be expected to work the same.

When the virtual environment is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor relays the request to the physical

system and caches the changes—which all happens at close to native speed (particularly if the request is sent through an open source hypervisor based on KVM, the Kernel-based Virtual Machine).

## **1.3 TYPES OF VIRTUALIZATION**

### **Data virtualization**

Data that's spread all over can be consolidated into a single source. Data virtualization allows companies to treat data as a dynamic supply—providing processing capabilities that can bring together data from multiple sources, easily accommodate new data sources, and transform data according to user needs. Data virtualization tools sit in front of multiple data sources and allows them to be treated as single source, delivering the needed data—in the required form—at the right time to any application or user.

### **Desktop virtualization**

Easily confused with operating system virtualization—which allows you to deploy multiple operating systems on a single machine—desktop virtualization allows a central administrator (or automated administration tool) to deploy simulated desktop environments to hundreds of physical machines at once. Unlike traditional desktop environments that are physically installed, configured, and updated on each machine, desktop virtualization allows admins to perform mass configurations, updates, and security checks on all virtual desktops

### **Server virtualization**

Servers are computers designed to process a high volume of specific tasks really well so other computers—like laptops and desktops—can do a variety of other tasks. Virtualizing a server lets it to do more of those specific functions and involves partitioning it so that the components can be used to serve multiple functions.

### **Operating system virtualization**

Operating system virtualization happens at the kernel—the central task managers of operating systems. It's a useful way to run Linux and Windows environments side-by-side. Enterprises can also push virtual operating systems to computers, which:

- Reduces bulk hardware costs, since the computers don't require such high out-of-the-box capabilities.
- Increases security, since all virtual instances can be monitored and isolated.
- Limits time spent on IT services like software updates.

## Network functions virtualization

Network functions virtualization (NFV) separates a network's key functions (like directory services, file sharing, and IP configuration) so they can be distributed among environments. Once software functions are independent of the physical machines they once lived on, specific functions can be packaged together into a new network and assigned to an environment. Virtualizing networks reduces the number of physical components—like switches, routers, servers, cables, and hubs—that are needed to create multiple, independent networks, and it's particularly popular in the telecommunications industry.

## 1.4 HYPERVISOR

A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware. The program which provides partitioning, isolation or abstraction is called virtualization hypervisor. The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a virtual machine manager (VMM).

## 1.5 TYPES OF HYPERVISORS

### TYPE-1 Hypervisor:

The hypervisor runs directly on the underlying host system. It is also known as “Native Hypervisor” or “Bare metal hypervisor”. It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor

### Pros & Cons of Type-1 Hypervisor:

**Pros:** Such kind of hypervisors is very efficient because they have direct access to the physical hardware resources (like CPU, Memory, Network, Physical storage). This causes the empowerment the security because there is nothing any kind of the third party resource so that attacker couldn't compromise with anything.

**Cons:** One problem with Type-1 hypervisor is that they usually need a dedicated separate machine to perform its operation and to instruct different VMs and control the host hardware resources.

### TYPE-2 Hypervisor:

A Host operating system runs on the underlying host system. It is also known as ‘Hosted Hypervisor’. Such kind of hypervisors doesn't run directly over the underlying hardware rather they run as an application in a Host system (physical machine). Basically, software installed on an operating system. Hypervisor asks the operating system to make hardware calls. Example of Type 2 hypervisor includes VMware Player or Parallels Desktop. Hosted hypervisors are often found on endpoints like PCs. The type-2 hypervisor is very useful for engineers, security analyst (for checking malware, or malicious source code and newly developed applications).



## Pros & Cons of Type-2 Hypervisor:

**Pros:** Such kind of hypervisors allows quick and easy access to a guest Operating System alongside the host machine running. These hypervisors usually come with additional useful features for guest machine. Such tools enhance the coordination between the host machine and guest machine.

**Cons:** Here there is no direct access to the physical hardware resources so the efficiency of these hypervisors lags in performance as compared to the type-1 hypervisors, and potential security risks are also there an attacker can compromise the security weakness if there is access to the host operating system so he can also access the guest operating system.

## 1.6 CHOOSING THE RIGHT HYPERVISOR

**Type 1 hypervisors offer much better performance than Type 2** ones because there's no middle layer, making them the logical choice for mission-critical applications and workloads. But that's not to say that hosted hypervisors don't have their place – they're much simpler to set up, so they're a good bet if, say, you need to deploy a test environment quickly. One of the best ways to determine which hypervisor meets your needs is to compare their performance metrics. These include CPU overhead, amount of maximum host and guest memory, and support for virtual processors. The following factors should be examined before choosing a suitable hypervisor:

1. **Understand your needs:** The company and its applications are the reason for the data centre (and your job). Besides your company's needs, you (and your co-workers in IT) also have your own needs. Needs for a virtualization hypervisor are:
  - Flexibility
  - Scalability
  - Usability
  - Availability
  - Reliability
  - Efficiency
  - Reliable support
2. **The cost of a hypervisor:** For many buyers, the toughest part of choosing a hypervisor is striking the right balance between cost and functionality. While a number of entry-level solutions are free, or practically free, the prices at the opposite end of the market can be staggering. Licensing frameworks also vary, so it's important to be aware of exactly what you're getting for your money.
3. **Virtual machine performance:** Virtual systems should meet or exceed the performance of their physical counterparts, at least in relation to the applications within each server. Everything beyond meeting this benchmark is profit.

4. **Ecosystem:** It's tempting to overlook the role of a hypervisor's ecosystem – that is, the availability of documentation, support, training, third-party developers and consultancies, and so on – in determining whether or not a solution is cost-effective in the long term.
5. **Test for yourself:** You can gain basic experience from your existing desktop or laptop. You can run both VMware vSphere and Microsoft Hyper-V in either VMware Workstation or VMware Fusion to create a nice virtual learning and testing environment.

## 2. SOFTWARE REQUIREMENT ANALYSIS

### 2.1 OVERVIEW OF VIRTUAL MACHINE

A virtual machine (VM) is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system (located off- or on-premises). Software called a hypervisor separates the machine's resources from the hardware and provisions them appropriately so they can be used by the VM.

The physical machines, equipped with a hypervisor such as Kernel-based Virtual Machine (KVM), is called the host machine, host computer, host operating system, or simply host. The many VMs that use its resources are guest machines, guest computers, guest operating systems, or simply guests. The hypervisor treats compute resources—like CPU, memory, and storage—as a pool of resources that can easily be relocated between existing guests or to new virtual machines.

VMs are isolated from the rest of the system, and multiple VMs can exist on a single piece of hardware, like a server. They can be moved between host servers depending on demand or to use resources more efficiently.

VMs allow multiple different operating systems to run simultaneously on a single computer—like a Linux® distro on a MacOS laptop. Each operating system runs in the same way an operating system or application normally would on the host hardware, so the end user experience emulated within the VM is nearly identical to a real-time operating system experience running on a physical machine.

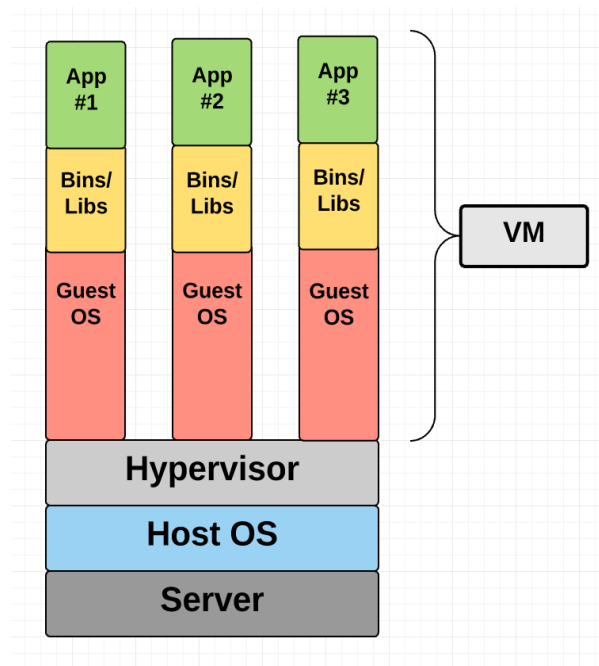


Fig:2.1.1 Virtual machine

## How do VMs work?

Virtualization technology allows you to share a system with many virtual environments. The hypervisor manages the hardware and separates the physical resources from the virtual environments. Resources are partitioned as needed from the physical environment to the VMs.

When the VM is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor schedules the request to the physical system's resources so that the virtual machine's operating system and applications can access the shared pool of physical resources.

## Why use a VM?

Server consolidation is a top reason to use VMs. Most operating system and application deployments only use a small amount of the physical resources available when deployed to bare metal. By virtualizing your servers, you can place many virtual servers onto each physical server to improve hardware utilization.

This keeps you from needing to purchase additional physical resources, like hard drives or hard disks, as well as reducing the need for power, space, and cooling in the datacentre. VMs provide additional disaster recovery options by enabling failover and redundancy that could previously only be achieved through additional hardware.

A VM provides an environment that is isolated from the rest of a system, so whatever is running inside a VM won't interfere with anything else running on the host hardware

Because VMs are isolated, they are a good option for testing new applications or setting up a production environment. You can also run a single purpose VM to support a specific process.

## 2.2 ABOUT LINUX

Linux® is an open source operating system (OS). An operating system is the software that directly manages a system's hardware and resources, like CPU, memory, and storage. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

Think about an OS like a car engine. An engine can run on its own, but it becomes a functional car when it's connected with a transmission, axles, and wheels. Without the engine running properly, the rest of the car won't work.

## How does Linux work?

Linux was designed to be similar to UNIX, but has evolved to run on a wide variety of hardware from phones to supercomputers. Every Linux-based OS involves the Linux kernel—which manages hardware resources—and a set of software packages that make up the rest of the operating system.

The OS includes some common core components, like the GNU tools, among others. These tools give the user a way to manage the resources provided by the kernel, install additional software, configure performance and security settings, and more. All of these tools bundled together make up the functional operating system. Because Linux is an open source OS, combinations of software can vary between Linux distributions.

## What's a command line?

The command line is your direct access to a computer. It's where you ask software to perform hardware actions that point-and-click graphical user interfaces (GUIs) simply can't ask.

Command lines are available on many operating systems—proprietary or open source. But it's usually associated with Linux, because both command lines and open source software, together, give users unrestricted access to their computer.

## Top 10 Flavour of Linux

1. Fedora
2. Ubuntu
3. CentOS
4. Debian
5. Kali Linux
6. Red Hat Enterprise Linux
7. Arch Linux
8. OpenSUSE
9. Gentoo
10. Linux Mini

## 2.3 BRIEF ABOUT ISO FILE

ISO stands for International Organization for Standardization but it isn't what the .iso file extension stands for. The extension was derived out standard patented by the International Organization for Standardization, for filesystems in optical (CD/DVD) disks. The file system standard is under the name ISO 9660, where the former term is used as the extension .iso for CD/DVD disk image files. But, ISO files represented with the .iso extension isn't mandatory, as ISO files with file extensions such as .img, .udf also exists.



Fig: 2.3.1 ISO Image

ISO files are also referred to as image files as they use the exact format as a disk and could be mounted on one's operating system as if it were a discrete disk. An ISO image is an archive file of an optical disc, a type of disk image composed of the data contents from every written sector on an optical disc, including the optical disc file system.

## **Characteristics of an ISO Image**

ISO image format is uncompressed, container less, and are stored in binary format. They are a sector-by-sector copy of the contents of an optical drive. Upon encountering an ISO image, the system expects the binary data of format specified under ISO 9660 or UDF (Universal Disk Format), optical media file system standard. ISO files like a regular file, aren't opened, but rather they are mounted (as if they are a volume/device). This behaviour is similar to when an operating system recognizes an optical drive. When ISO images are created from optical disks, then ISO files store only the user data from each sector on an optical disc, ignoring the control headers and error correction data, and are therefore slightly smaller than a raw disc image of optical media. The file format isn't exclusive to CD/DVD drives, as one could create an ISO image file with custom files, where the ISO would act as a Physical Disk Drive when mounted.

## **Advantages of ISO**

- Allows quick access of files contained within, as the file is mounted (as opposed to other similar formats where contents need to be extracted).
- Gives deception of a virtual drive, and therefore could be used as a backward compatibility option for the application that requires a disk drive
- Most operating systems provide native support for ISO image files, therefore no utility is required for using this file format

## **Disadvantages of ISO**

- The files inside an ISO image could be edited, but the whole ISO needs to be recompiled again which is time-consuming (as opposed to other formats such as ZIP, etc.)
- Performance may not be the most optimal, as the file format follows as optical drive structure which isn't much efficient
- No significant error resilience or integrity preserving protocols are enforced in the format
- Small corruption in any sector of the data, leads to unmountable ISO image files (since the file must be mounted in order to access the data inside, it consequently leads to the data becoming unreadable or not accessible)

## **Applications of ISO**

- ISO image files are used in emulating a optical drives in Video Game Console emulators such as RPCS3, PCSX2, Zenia, PPSSPP etc.
- The format is used extensively for storing copies of Operating systems like Windows, Linux, Disk Operating System (DOS) etc. Used by backup programs to create backup disks

## 3. SOFTWARE TESTING AND MONITORING

### 3.1 STRESS TESTING

Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions



Fig: 3.1.1 stress testing

In Software Engineering, Stress Testing is also known as Endurance Testing. Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity. A most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks. It also checks whether the system demonstrates effective error management under extreme conditions

The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad. Notepad is under stress and gives 'Not Responded' error message.

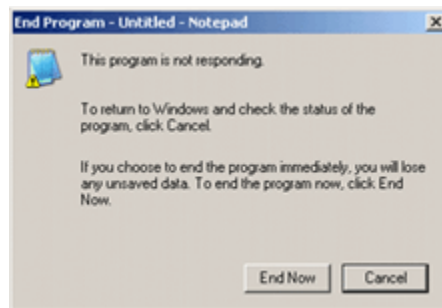


Fig:3.1.2 Stress working

In this project, we will learn-

- What is Stress Testing?
- Need for Stress Testing
- Goals of Stress Testing
- Load Testing Vs Stress Testing
- Types of Stress Testing
- How to do Stress Testing?

- Tools recommended for Stress Testing
- Metrics for Stress Testing

Consider the following scenarios -

- During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
- When a blog is mentioned in a leading newspaper, it experiences a sudden surge in traffic

It is imperative to perform Stress Testing to accommodate such abnormal traffic spikes. Failure to accommodate this sudden traffic may result in loss of revenue and repute.

Stress testing is also extremely valuable for the following reasons:

- To check whether the system works under abnormal conditions.
- Displaying appropriate error message when the system is under stress
- System failure under extreme conditions could result in enormous revenue loss
- It is better to be prepared for extreme conditions by executing Stress Testing

## Goals of Stress Testing

The goal of stress testing is to analyse the behaviour of the system after a failure. For stress testing to be successful, a system should display an appropriate error message while it is under extreme conditions

To conduct Stress Testing, sometimes, massive data sets may be used which may get lost during Stress Testing. Testers should not lose this security-related data while doing stress testing

The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.

## Load Testing Vs Stress Testing



Fig: 3.1.3 Load testing v/s Stress testing



Load Testing	Stress Testing
Load Testing is to test the system behavior under normal workload conditions, and it is just testing or simulating with the actual workload	Stress testing is to test the system behavior under extreme conditions and is carried out till the system failure
Load testing does not break the system	Stress testing tries to break the system by testing with overwhelming data or resources

## Types of Stress Testing:

Following are the types of stress testing and are explained as follows:

### Distributed Stress Testing

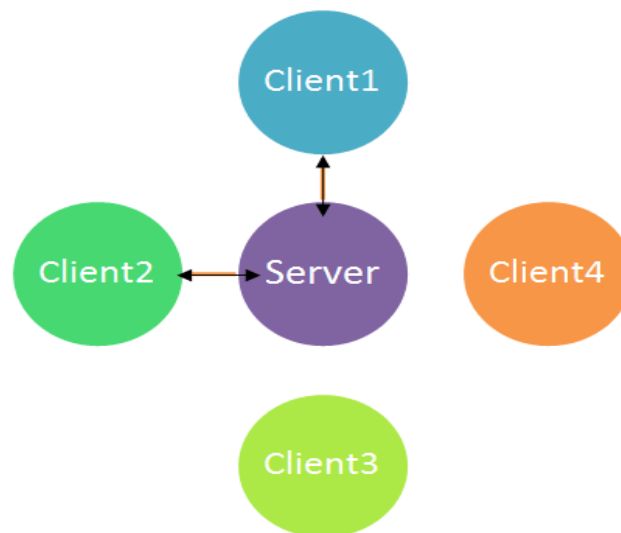


Fig: 3.1.4 Distributed Testing

In distributed client-server systems, testing is done across all clients from the server. The role of stress server is to distribute a set of stress tests to all stress clients and track on the status of the client. After the client contacts the server, the server adds the name of the client and starts sending data for testing.

Meanwhile, client machines send signal or heartbeat that it is connected with the server. If the server does not receive any signals from the client machine, it needs to be investigated further for debugging.

Night run is the best option to run these stress testing scenarios. Large server farms need a more efficient method for determining which computers have had stress failures that need to be investigated

## **Application Stress Testing**

This testing concentrate on finding defects related to data locking and blocking, network issues and performance bottlenecks in an application

## **Transactional Stress Testing**

It does stress testing on one or more transactions between two or more applications. It is used for fine-tuning & optimizing the system

## **Systemic Stress Testing**

This is integrated stress testing which can be tested across multiple systems running on the same server. It is used to find defects where one application data blocks another application

## **Exploratory Stress Testing**

This is one of the types of stress testing which is used to test the system with unusual parameters or conditions that are unlikely to occur in a real scenario. It is used to find defects in unexpected scenarios like

1. A large number of users logged at the same time
2. If a virus scanner started in all machines simultaneously
3. If Database has gone offline when it is accessed from a website
4. When a large volume of data is inserted to the database simultaneously

## **How to do Stress Testing?**

Stress Testing process can be done in 5 major steps:

1. Planning the Stress Test. Here you gather the system data, analyze the system, define the stress test goals
2. Create Automation Scripts: In this phase, you create the Stress testing automation scripts, generate the test data for the stress scenarios
3. Script Execution: In this stage, you run the Stress testing automation scripts and store the stress results
4. Results Analysis: In this stage, you analyze the Stress Test results and identify bottlenecks
5. Tweaking and Optimization: In this stage, you fine-tune the system, change configurations, optimize the code with goal meet the desired benchmark.

Lastly, you again run the entire cycle to determine that the tweaks have produced the desired results. For example, it's not unusual to have to 3 to 4 cycles of the Stress Testing process to achieve the performance goals

## Tools recommended for Stress Testing

### LoadRunner

LoadRunner from HP is a widely-used Load Testing tool. Load Test Results shaped by LoadRunner are considered as a benchmark

### JMeter

JMeter is an Open Source testing tool. It is a pure Java application for stress and Performance Testing. JMeter is intended to cover types of tests like load, functional, stress, etc. It needs JDK 5 or higher to function

### Stress

This tool provides extensive analysis of the web application performance, provides results in graphical format, and it is extremely easy to use. No high-level scripting is required and gives a good return on investment

### Neo load

This is a popular tool available in the market to test the web and Mobile applications. This tool can simulate thousands of users in order to evaluate the application performance under load and analyse the response times. It also supports Cloud-integrated - performance, load and stress testing. It is easy to use, cost-effective and provides good scalability.

## 3.2 MONITORING LINUX SERVER BY GLANCES

If you're a Linux administrator, you know that information is valuable. Without the ability to quickly gain insight into data centre servers, troubleshooting becomes exponentially more difficult. Because of that, you need that information at a glance. But when your Linux servers are sans a desktop environment, how do you get the information you need?

There are many tools available that provide various bits of information. For many admins, the first tool to use is the top command. But for some, top misses out on some crucial information. What if there is a terminal app that can provide you with more information than top? That tool does exist, and it's called Glances

### GLANCES

Glances is a cross-platform monitoring tool which aims to present a maximum of information in a minimum of space through a curses or Web based interface. It can adapt dynamically the displayed information depending on the terminal size.

Glances is a text-based monitor that provides information about:

- CPU
- Memory
- Load
- Process list

- Network interface
- Disk I/O
- IRQ / Raid
- Sensors
- Filesystem (and folders)
- Docker
- Monitor
- Alert
- System info
- Uptime
- Quick look (CPU, MEM, LOAD)

Glances is a free tool and licensed under GPL to monitor GNU/Linux and FreeBSD operating systems. There are lots of interesting options available in Glances as well. One of the main features we have seen in Glances is that we can set thresholds (careful, warning and critical) in configuration file and information's will be shown in colors which indicates the bottleneck in the system

Stats can also be exported to external time/value databases

It can also work in client/server mode. Remote monitoring could be done via terminal, Web interface or API (XMLRPC and RESTful).

Glances is written in Python and uses the psutil library to get information from your system

## Glances Color Codes

Meaning of Glances color code:

1. GREEN : OK (everything is fine)
2. BLUE : CAREFUL (need attention)
3. VIOLET: WARNING (alert)
4. RED : CRITICAL (critical)

We can set thresholds in configuration file. By default thresholds set is (careful=50, warning=70 and critical=90), we can customized as per our needs. The default configuration file is located at `‘/etc/glances/glances.conf’`.

## Glances Options

Besides, several command line options, glances provides many more hot keys to find output information while glances is running. Below are the list of several hot keys.

1. a – Sort processes automatically
2. c – Sort processes by CPU%
3. m – Sort processes by MEM%
4. p – Sort processes by name
5. i – Sort processes by I/O rate
6. d – Show/hide disk I/O stats
7. f – Show/hide file system stats
8. n – Show/hide network stats
9. s – Show/hide sensors stats
10. y – Show/hide hddtemp stats
11. l – Show/hide logs
12. b – Bytes or bits for network I/O
13. w – Delete warning logs
14. x – Delete warning and critical logs
15. 1 – Global CPU or per-CPU stats
16. h – Show/hide this help screen
17. t – View network I/O as combination
18. u – View cumulative network I/O
19. q – Quit (Esc and Ctrl-C also work)

## Features of Glances

Let's have a quick glance over the main features Glances provides (pun intended).

- Can monitor 15+ metrics on a system (including docker containers).
- Flexible usage modes: standalone, client-server, over SSH and web mode.
- Versatile REST API and XML-RPC available for integrations.
- Readily supports exporting data to different services and databases.
- Highly configurable and adaptable to different needs.
- Very comprehensive Documentation.

## Installing Glances on Ubuntu and other Linux distributions

Glances is available in the official repositories of many Linux distributions. This means you can use your distribution's package manager to install it easily.

On Debian/Ubuntu based distributions, you can use the following command:

**sudo apt install glances**

You can also install the latest Glances using snap package:

**sudo snap install glances**

## Using Glances to monitor Linux system resources on local system (standalone mode)

You can easily launch Glances to monitor your local machine by running this command in the terminal.

You can instantly observe that it integrates a lot of different information in one single screen. I like that it shows the public and private IP's of the computer right at the top.

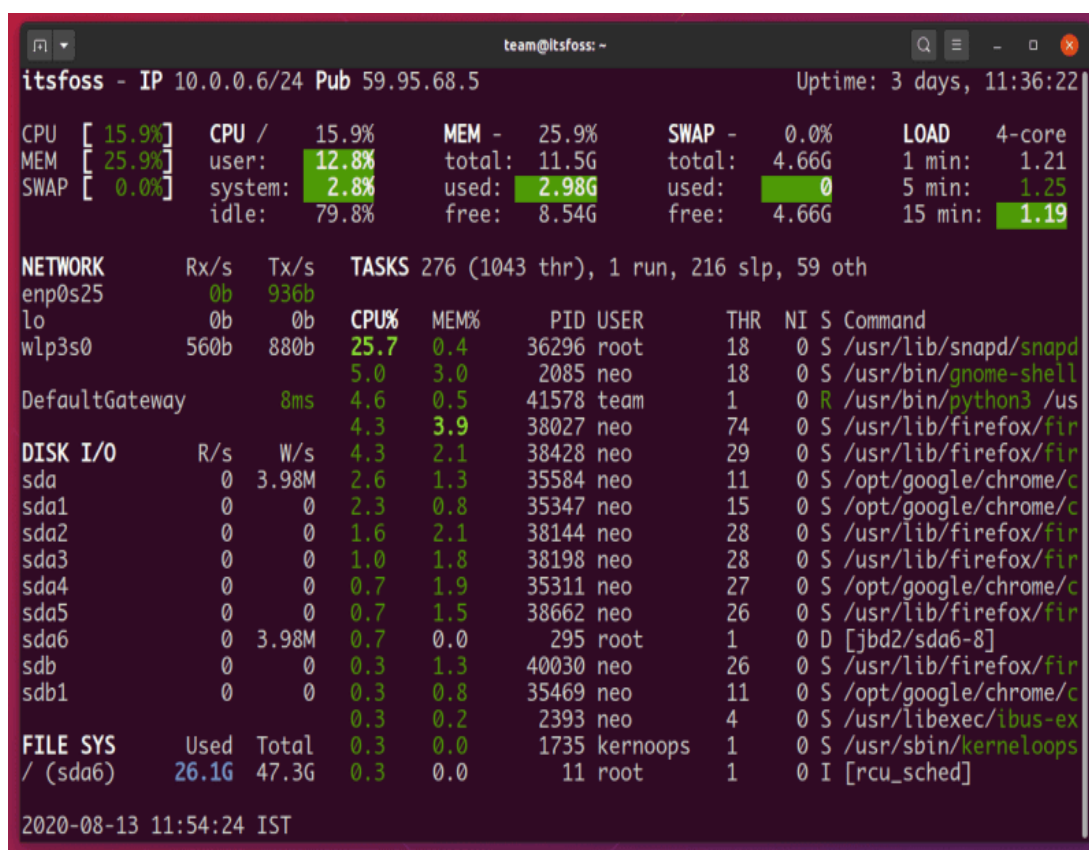


Fig : 3.2.1 Glances Monitorization

Glances is also interactive, meaning you can use commands to interact with it while it is running. You can press “s” to bring up the sensors onto the screen; “k” to bring up the TCP connections list; “l” to expand the CPU stats to show individual threads.

You can also use the arrow keys to move around the list of processes and sort the table by different metrics.

You can launch Glances with various command line options. There are plenty more interactive commands as well. **Press Ctrl+C to exit Glances.**

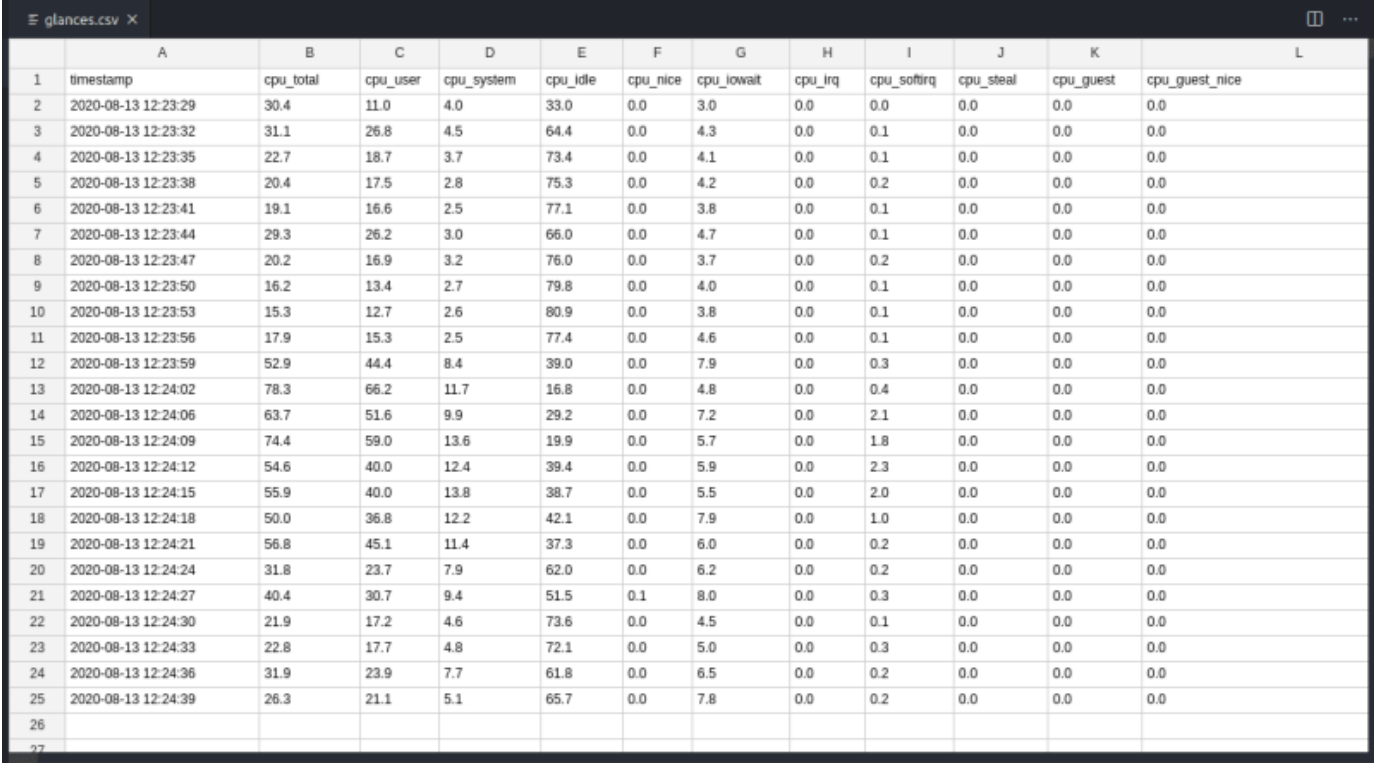
## Export Glances data to different services

One of the biggest advantage of using Glances is the out of the box support to export data to various databases, services and integration into various data pipelines seamlessly.

You can export to CSV while monitoring with this command.

**glances --export csv --export-csv-file /tmp/glances.csv**

/tmp/glances.csv' is the location of the file. The data is filled in neatly as time series.



	A	B	C	D	E	F	G	H	I	J	K	L
1	timestamp	cpu_total	cpu_user	cpu_system	cpu_idle	cpu_nice	cpu_iowait	cpu_irq	cpu_softirq	cpu_steal	cpu_guest	cpu_guest_nice
2	2020-08-13 12:23:29	30.4	11.0	4.0	33.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
3	2020-08-13 12:23:32	31.1	26.8	4.5	64.4	0.0	4.3	0.0	0.1	0.0	0.0	0.0
4	2020-08-13 12:23:35	22.7	18.7	3.7	73.4	0.0	4.1	0.0	0.1	0.0	0.0	0.0
5	2020-08-13 12:23:38	20.4	17.5	2.8	75.3	0.0	4.2	0.0	0.2	0.0	0.0	0.0
6	2020-08-13 12:23:41	19.1	16.6	2.5	77.1	0.0	3.8	0.0	0.1	0.0	0.0	0.0
7	2020-08-13 12:23:44	29.3	26.2	3.0	66.0	0.0	4.7	0.0	0.1	0.0	0.0	0.0
8	2020-08-13 12:23:47	20.2	16.9	3.2	76.0	0.0	3.7	0.0	0.2	0.0	0.0	0.0
9	2020-08-13 12:23:50	16.2	13.4	2.7	79.8	0.0	4.0	0.0	0.1	0.0	0.0	0.0
10	2020-08-13 12:23:53	15.3	12.7	2.6	80.9	0.0	3.8	0.0	0.1	0.0	0.0	0.0
11	2020-08-13 12:23:56	17.9	15.3	2.5	77.4	0.0	4.6	0.0	0.1	0.0	0.0	0.0
12	2020-08-13 12:23:59	52.9	44.4	8.4	39.0	0.0	7.9	0.0	0.3	0.0	0.0	0.0
13	2020-08-13 12:24:02	78.3	66.2	11.7	16.8	0.0	4.8	0.0	0.4	0.0	0.0	0.0
14	2020-08-13 12:24:06	63.7	51.6	9.9	29.2	0.0	7.2	0.0	2.1	0.0	0.0	0.0
15	2020-08-13 12:24:09	74.4	59.0	13.6	19.9	0.0	5.7	0.0	1.8	0.0	0.0	0.0
16	2020-08-13 12:24:12	54.6	40.0	12.4	39.4	0.0	5.9	0.0	2.3	0.0	0.0	0.0
17	2020-08-13 12:24:15	55.9	40.0	13.8	38.7	0.0	5.5	0.0	2.0	0.0	0.0	0.0
18	2020-08-13 12:24:18	50.0	36.8	12.2	42.1	0.0	7.9	0.0	1.0	0.0	0.0	0.0
19	2020-08-13 12:24:21	56.8	45.1	11.4	37.3	0.0	6.0	0.0	0.2	0.0	0.0	0.0
20	2020-08-13 12:24:24	31.8	23.7	7.9	62.0	0.0	6.2	0.0	0.2	0.0	0.0	0.0
21	2020-08-13 12:24:27	40.4	30.7	9.4	51.5	0.1	8.0	0.0	0.3	0.0	0.0	0.0
22	2020-08-13 12:24:30	21.9	17.2	4.6	73.6	0.0	4.5	0.0	0.1	0.0	0.0	0.0
23	2020-08-13 12:24:33	22.8	17.7	4.8	72.1	0.0	5.0	0.0	0.3	0.0	0.0	0.0
24	2020-08-13 12:24:36	31.9	23.9	7.7	61.8	0.0	6.5	0.0	0.2	0.0	0.0	0.0
25	2020-08-13 12:24:39	26.3	21.1	5.1	65.7	0.0	7.8	0.0	0.2	0.0	0.0	0.0
26												
27												

Fig:3.2.2 Glances exportation

## 4. RESULT

Here is the result obtained on performing the benchmarking on two different Hypervisors that is VMware and VirtualBox of two VM's that is Ubuntu VM and Fedora VM . They are graphically shown as below:

For Ubuntu VM :-

Ubuntu VM	Vmware	Virtual Box
CPU usage	76	71
Memory usage	42.8	38.6

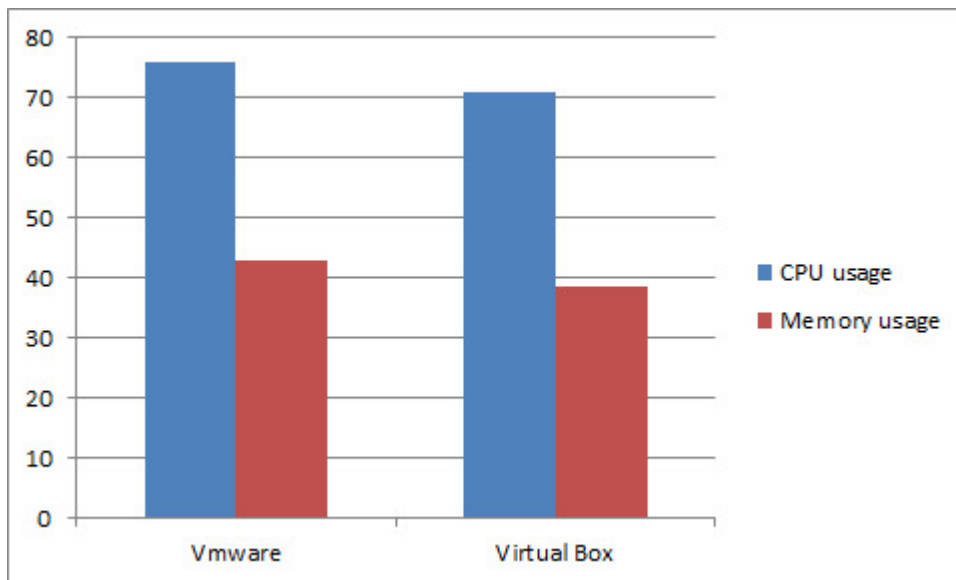


fig: 4.1 Graphical representation for Ubuntu VM



For Fedora VM :-

Fedora VM	Vmware	Virtual Box
CPU usage	78.7	66
Memory usage	71.8	64.7

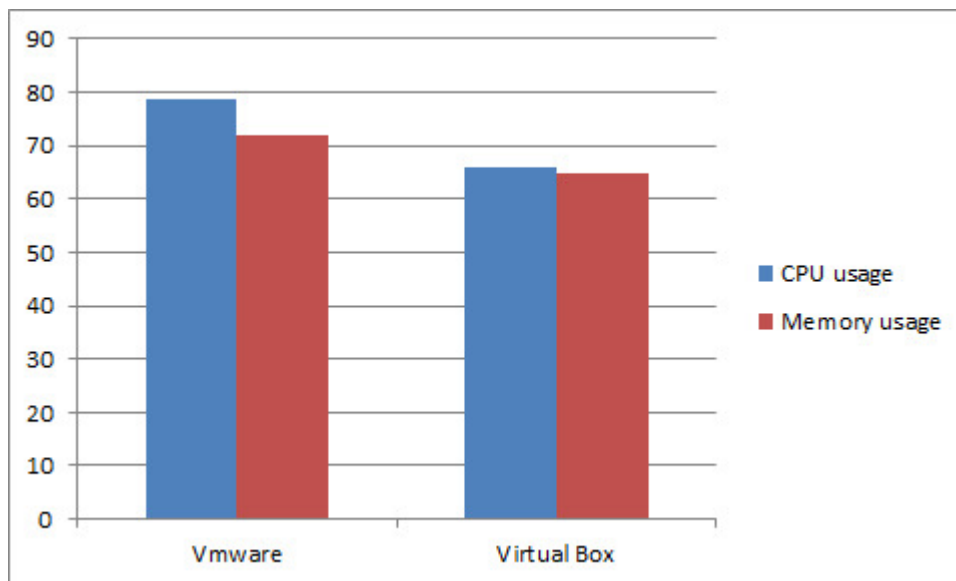


Fig: 4.2 Graphical Representation for Fedora VM

## 5. CONCLUSION

Analysing the performance of the different hypervisors with the similar hardware configuration and operating system. We focused on study of Virtualization, Hypervisor and Benchmarking. We have carried out experiment to benchmark different hypervisor performance with respect to CPU use, RAM use, Disk Speed and Network speed. From the experiment carried out we may conclude that Best CPU performance is given by Xenserver 6.5. While Best RAM performance is given by VMware. DISK and N/W Performance of VirtualBox is much better than other two. The Performance of hypervisor may vary for different perspectives. The performance of different hypervisors may get influenced by the environment and the resources. The considered case studies in the report are with the different research papers regarding the benchmarking the performance of the hypervisors in different environment. The performance of different hypervisors may get influenced by the environment and the resources. Further study can be carried out in discovering the new benchmarking techniques for benchmarking the hypervisors.

## 6. REFERENCES

- [1] <https://github.com/nitinagrawalup/glau-cloud-projects/tree/main/mini-project-2>
- [2] <https://www.freecodecamp.org/news/how-to-install-ubuntu-with-oracle-virtualbox/>
- [3] <https://youtu.be/zADj0k0waFY>
- [4] <https://youtu.be/yWrlVSfVCI>
- [5] <https://youtu.be/E3Ioopzt8ko>
- [6] <https://www.semanticscholar.org/paper/Performance-Comparison-of-the-type-2-hypervisor-and-Vojnak-%C3%90or%C4%91evi%C4%87/b102dbea4d8ba047d09838624cc6d696313832ca>
- [7] Lakshmanan L, Gayathri P. Enhancing network I/O virtualization for cloud using finite multi server queuing model. Indian Journal of Science and Technology. 2015; 8(32):1–4. Crossruff.
- [8] Buyya R, Selvi T, Vecchiola C. Mastering cloud computing. 1st ed. McGraw Hill Education; 2013.
- [9] Benchmarking. Available from: [https://en.wikipedia.org/wiki/Benchmark\\_\(computing\)](https://en.wikipedia.org/wiki/Benchmark_(computing))
- [10] Patni JC, Sharma A, Mishra P, Kumar A. Datacentre virtualization with optimization and customization. Indian Journal of Science and Technology. 2016; 9(44):1–6. Crossruff.
- [11] Li J, Wang Q, Jayasinghe D, Park J, Zhu T, Pu C. Performance overhead among three hypervisors an ex-perimental study using Hadoop benchmarks. 2013 IEEE International Congress Big Data (Bigdata Congress); 2013.