

# System Performance in Virtualization

Hiteshi  
Thapar University,  
Patiala.

V.P Singh  
Thapar University,  
Patiala.

## ABSTRACT

One thing that has been a growing challenge is keeping up with all the hardware, operating system and software upgrades. Upgrades and new releases are a constant factor presented by every system vendor. Server virtualization permits multiple virtual operating systems to run on a single physical machine, yet act logically distinct with consistent hardware profiles.

VMware is one of the leading virtualization software vendors today, provides a production ready server virtualization suite, VMware Infrastructure. It provides comprehensive performance metrics to monitor your VMware ESX/ESXi servers and their guest virtual machines, and helps you ensure they are performing well at all times. Application Manager connects with VMware ESX/ESXi servers through standard APIs and determines the health status as well as the performance of the host servers and their corresponding virtual machines. It provides a single console to monitor physical, virtual and cloud components of your IT infrastructure.

Application manager can then analyse the performance using some tools like, esxplot, Perform etc, which ensures the graphical views as well as the performance at peak time.

## General Terms

Virtualization, Hypervisor, Benchmark.

## 1. INTRODUCTION

Virtualization typically involves using special software to safely run multiple operating systems and applications simultaneously with a single computer. The technology initially allows company to consolidate an array of

servers to improve operating efficiency and reduce costs. It has since been applied to dealing with data storage as well as desktop systems. Owing to the success of tools developed by VMware Inc., the technology has become one of the most talk-about technology, and has draw attention from both IS professionals and non-IS executives in virtually all industries. Despite the potentially significant impact on company's operations, this technology has virtually been ignored by the academic researchers.

Virtualized infrastructures help simplify IT operations in many ways, from helping to shield software from hardware, to enabling secure resource sharing, to facilitating software deployment and relocation.

They also increase business agility by:

- Enabling IT staff to dynamically reallocate resources as needed to avoid planned downtime
- Enhancing the efficiency of application testing and development
- Facilitating rapid, cost-effective disaster recovery

### 1.1. Basic Terms of Virtualization

Virtualization is commonly defined as a technology that introduces a software abstraction layer between the hardware and the operating system and applications running on top of it. This abstraction layer is called virtual machine monitor (VMM) or hypervisor and

basically hides the physical resources of the computing system from the operating system (OS). Since the hardware resources are directly controlled by the VMM and not by the OS, it is possible to run multiple (possibly different) OSs in parallel on the same hardware. As a result, the hardware platform is partitioned into one or more logical units called virtual machines (VMs).

The following requirements for a VMM have been defined [1]:

- **Equivalence:** Running an application inside a virtual machine must be equivalent to running the same application on the underlying hardware.
- **Control:** The VMM must control and synchronize the access of VMs to hardware resources.
- **Isolation:** VMs must be isolated from each other with the purpose of ensuring stability (the crash of a VM should not affect another VMs), security (a possibly compromised VM shouldn't grant access to other VMs) and data consistency.
- **Performance:** The performance overhead caused by virtualization should be minimal, close to "bare metal" performance.
- **Encapsulation:** VMs must exist in form of a file or a directory of files which allows easy migration or cloning of the VM.

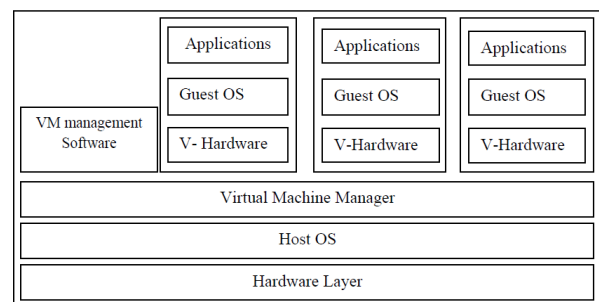
## 1.2 Virtualization Technologies

There are various virtualization technologies available in the market that helps to virtualize the environment. Depending on the needs and the goals of the organization, one virtualization technology is better than other.

Following section describes the overview of some of the existing technologies.

### 1.2.1. Full Virtualization:

In this approach, the VMM is called the virtual machine manager that runs on the top of the operating system, commonly as an application in user space. The result is that, in the VMs, the applications and the guest operating systems run on top of a virtual hardware provided by the VMM. However, the virtual machine environment that provides enough representation of the underlying hardware to allow guest operating systems to run without modification can be considered to provide "Full Virtualization" [2]. In full virtualization setup, I/O devices are allotted to the guest machines by imitating the physical devices in the virtual machine monitor: interacting with these devices in the virtual environment is then directed to the real physical devices either by the host operating system driver or by the "hypervisor driver". The architecture can be observed as below in Fig.1.1.



**Fig.1.1: Full Virtualization**

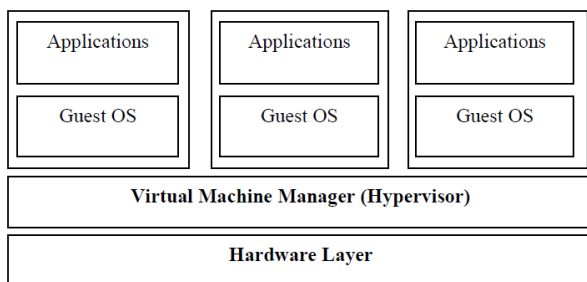
### 1.2.2. Para virtualization:

Para virtualization is the technique used by Xen which provides a virtual machine interface representing a slightly modified copy of the underlying hardware, where the non-virtualizable portions of the x86 original instruction set are replaced with their easily virtualized equivalents. Unlike full virtualization, in para virtualization the running guest OS should be modified in order to be operated in virtual environment. Para virtualization is the subset of server virtualization, which provides a thin software interface between the host hardware and the modified guest OS. An interesting fact in this

technology is that the guest machines are aware of the fact that they are running in virtualized environment. One of the main characteristics of the para virtualization technology is, the virtual machine monitor is which simply allows para virtualization to achieve performance closer to non virtualized hardware. Device interaction in para virtualized environment is very similar to the device interaction of full virtualized environment; the virtual devices in para virtualized environment also rely on physical device drivers of the underlying hardware [3].

### 1.2.3. Hardware layer virtualization:

This approach is commonly used on the server market to due to its high virtual machine isolation and performance. Here, the VMM runs directly on hardware, controlling and synchronizing the access of the guest OSs to the hardware resources. Fig.1.2 depicts the architecture.



**Fig.1.2: Hardware Technology**

### 1.2.4. Storage Virtualization:

Storage virtualization is a form of Resource Virtualization, where a logical storage is created by abstracting all the physical storage resources that are scattered over the network. First the physical storage resources are aggregated to form a storage pool which then forms the logical storage. This logical storage which is the aggregation of scattered physical resources appears to be a single monolithic storage device to the user. In storage virtualization, several physical storages are multiplexed into a single logical storage. Storage virtualization in its most basic form is used with RAID implementations, where two or more physical hard drives are combined to a

single logical hard disk to provide data redundancy. Bigger scale implementations of storage virtualization are Storage Area Network (SAN) and Network Attached Storage (NAS) technologies. With the help of storage virtualization workload of a single hardware can be uniformly distributed to multiple ones. Storage virtualization also provides fault tolerance as data can be easily replicated to multiple physical locations throughout the network. [4]

## 2. Hypervisor

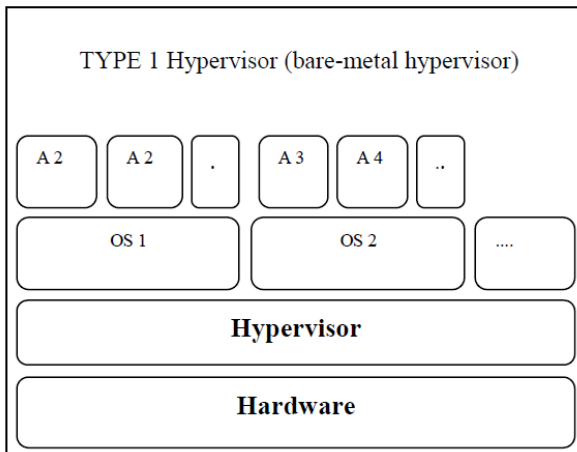
A hypervisor, also called as virtual machine manager, is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources allocating, what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other or we can say, It's the software program or part of the code in firmware that manages either multiple operating systems or multiple instances of the same operating system on a single computer system. The hypervisor's job is to manage the system's processor, memory and other resources to allocate what each operating system requires. Hypervisors provide the means to logically divide a single, physical server or blade, allowing multiple operating systems to run securely on the same CPU and increase the CPU utilization. Where hardware partitioning allows for hardware consolidation, hypervisors allow for flexibility in how the virtual resources are defined and managed, making it a more-often used system consolidation solution.

### 2.1. Types of hypervisor

IBM breaks hypervisors down into two different types [5]:

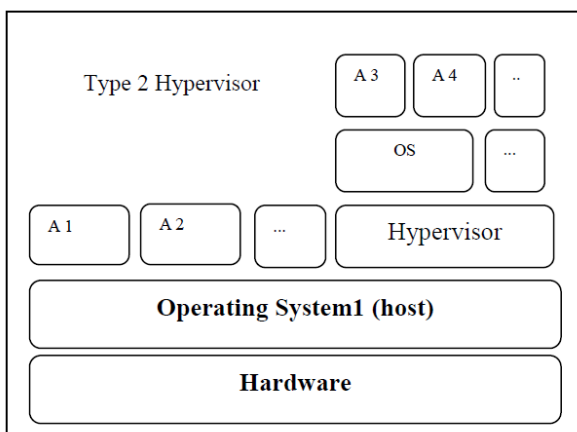
- a. Type 1 hypervisor
- b. Type 2 hypervisor

**Type 1 hypervisors** are also called Bare-metal hypervisor and are those that run directly on the system hardware and offer a higher level of virtualization and hardware security as shown in Fig. 2.1



**Fig.2.1 : Type 1 Hypervisor**

**Type 2 hypervisors** are those that run on a host operating system that provides virtualization services, such as I/O device support and memory management. These are used mainly on client systems where efficiency is less critical, and are also commonly used for systems where support for a broad range of I/O devices is needed and can be provided by the host operating system. Fig. 2.2 below shows type 2 hypervisors

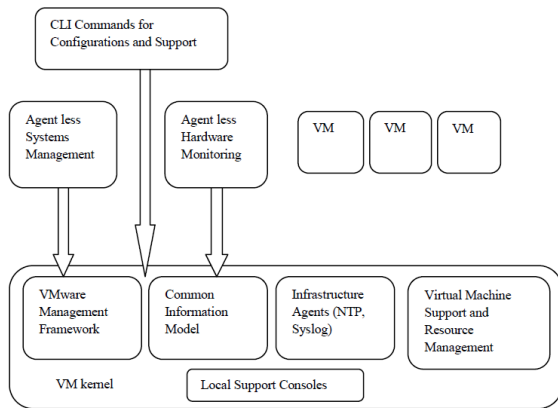


**Fig.2.2 : Type 2 Hypervisor**

## 2.2. Architecture:

**ESX Server** is a type 1 hypervisor that creates logical pools of system resources so that many virtual machines can share the same physical resources. ESX Server is an operating system that functions like a hypervisor and runs directly on the system hardware. ESX Server inserts a virtualization layer between the system hardware and the virtual machines, turning the system hardware into a pool of logical computing resources that ESX Server can dynamically allocate to any operating system or application. The guest operating systems running in virtual machines interact with the virtual resources as if they were physical resources. The hypervisor architecture of VMware vSphere 5.0 plays a critical role in the management of the virtual infrastructure. ESXi runs independently of a host operating system (OS) and improves hypervisor management in the areas of security, deployment and configuration, and ongoing administration.

In the original ESX architecture, the virtualization kernel (VMkernel) is augmented by a management partition known as the console operating system (COS) or service console. In the ESXi Architecture, the COS has been removed, and all of the VMware agents run directly on the VMkernel. Infrastructure services are provided natively through modules included in the VMkernel. Other authorized third-party modules, such as hardware drivers and hardware monitoring components, can run in the VMkernel as well. Only modules that have been digitally signed by VMware are allowed on the system, creating a tightly locked-down architecture. Preventing arbitrary code from running on the ESXi host greatly improves the security and stability of the system, As shown in below Fig. 2.3 [6].



**Fig.2.3 : ESXi hypervisor**

The **key components** of the ESX Server architecture are:

- **ESX Server virtualization layer:** Separates the underlying physical resources from the virtual machines.
- **Resource manager:** Creates virtual machines and delivers processing units, memory, network bandwidth, and disk bandwidth to them. It efficiently maps the physical resources to the virtual resources.
- **Service console:** Controls the installation, configuration, administration, troubleshooting, and maintenance of the ESX Server. The service console resides in its own virtual machine. ESX Server automatically configures the service console virtual machine when you install ESX Server. The service console also provides a place to install systems software.
- **Hardware interface components (including device drivers):** Delivers hardware-specific services while hiding hardware differences from other parts of the system.

### 3. Problem Statement

The main objective of this paper is to investigate the system performance i.e. memory usage, CPU usage which may not

bring immediate focus among other things, as CPU resources are difficult to interpret at the very first glance, and it always look like, that there are still sufficient resources available.

But it may come at some point of utilization at “hang” state for a short time or virtual systems may “freeze”, even though the resource utilization is below 20%.

### 4. Experimental Setup

Experiments are designed to evaluate the system performance by ESXi hypervisor. Many authors had quantified a number of virtualization platforms with different experiments. Our experiments finds the system performance which is one of the major problems in the field of virtualization which includes memory usage, CPU usage, and also interpret the utilization of the virtual systems.

#### 4.1. Basic hardware requirements [7]:

To successfully install ESX/ESXi, you must have proper hardware.

- 2 GB of RAM
- One Gigabit network interface
- 4 GB of hard drive space
- 2 processors

#### 4.2. VMware ESXi server:

VMware's virtual machine (VM) approach creates a uniform hardware image -- implemented in software— on which operating systems and applications run. On top of this platform, VMware's VirtualCenter provides management and provisioning of virtual machines, continuous workload consolidation across physical servers. **VMware Server** is a hosted hypervisor that is available free of charge. Installation of the hypervisor must be done on top of either Windows or Linux host operating system. Range of supported operating systems is quite good and VMware Server also supports 64-bit guests. It is a lightweight solution that cannot compete with bare-metal hypervisor solutions in performance but is useful, for example as a testing environment or legacy hardware

elimination. VMware Server is a great product to get started with virtualization as it is free and does not set any hardware requirements [8] [4].

**VMware ESX 5.x** will be the last major version of ESX with the service console included; many are starting to look toward the future and the role of ESXi in their infrastructures. The market provides several organizations and companies that each offer their own approach to x86 server virtualization.

#### 4.2.1. Installation steps:

VMware ESXi server comes in two versions: **Embedded version** is “installed” directly on the server by the hardware vendor.

**Installable version** is installed in a similar fashion as other servers, but resulting server will not have a fully functional Service Console.

There are also three ways to run the installer:

1. Interactive Graphical (default)
2. Scripted
3. Interactive text

For this experiment, we are using “Interactive text” mode of installation.[9]

#### 4.2.2. Configuration of ESXi server:

ESXi can be configured either statically or dynamically. Configuration used for this experiment is given in below Table 4.1:

**Table 4.1: Configuring ESXi**

IP Configuration:	Static or Dynamic
IP addresses:	172.31.5.66
Subnet Mask:	255.255.255.0
Default Gateway:	172.31.5.1

It also includes DNS settings of our network.

Details for network in DNS settings used for this experiment is as shown in below Table 4.2:

**Table 4.2: Network details in DNS settings**

PrimaryDNS Server:	172.31.1.6
Secondary DNS Server:	Optional (for static)
Hostname:	localhost

#### 4.3. vSphere Client

All administrative functions are done through the vSphere Client. A subset of those functions is available through the vSphere Web Client.

There is a difference between vSphere Client[10] and vSphere Web Client.[11]

vSphere Client is used for infrastructure configuration and day to day operations. Applications installed on this client are locally and can connect to ESXi or directly to host. It also supports full range of administrative functionality. For installing vSphere client, we must have OVF template, so that it can we can install it on VMware host[12].

#### 4.4. Esxtop and esxplot

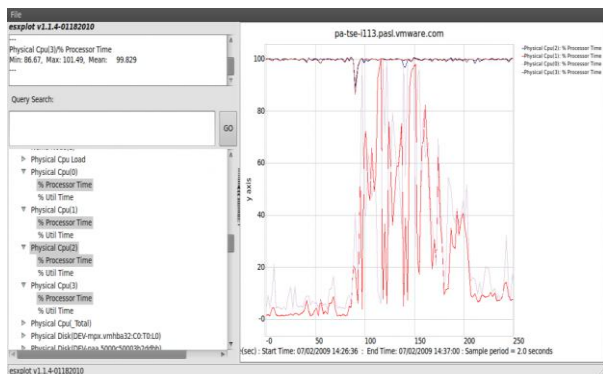
Esxtop is VMware's version of popular "top" command in Linux or Unix, that run on an ESX or ESXi server. Both top and esxtop run only at the command line of a server. You would access these commands by either going directly to a server console or by connecting to a server remotely via SSH (or telnet, if enabled).

**Esxtop**[13] is used to analyze real-time performance data from an individual ESX or ESXi server. You bring it up by going to the physical server console or remote console via SSH [14], logging in, and typing esxtop. Esxtop analyzes CPU, Memory, Disk, and Network statistics. More specifically, you'll find that esxtop has 8 different "**displays**" that show CPU, interrupt, memory, network, disk adapter, disk interface, disk VM, and power management as shown in Fig.4.1

```
Switch display:
c:cpu          i:interrupt  m:memory     n:network
d:disk adapter u:disk device v:disk VM      p:power mgmt
```

**Fig.4.1: esxplot display screen**

**Esxplot**[15] is a GUI application that lets you explore the data collected by esxtop in batch mode. The program takes a single command line argument which is the esxtop batch mode output file. You can also simply start esxplot without any arguments, and enter a dataset file via the File attribute of the menu bar. Esxplot loads the data in this file and presents the metrics as a hierarchical tree where the values are selectable in the left panel. In the right panel, a graph is plotted (value over time) of the selected metric, in this way, you can “browse” the contents of these somewhat unwieldy files as shown in Fig.4.2



**Fig.4.2 : ESXPLOT display**

#### 4.5. Benchmark tool

A benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it. There are a number of different types of bench-marking [16], which are driven by different motivating factors and thus involve different comparisons.

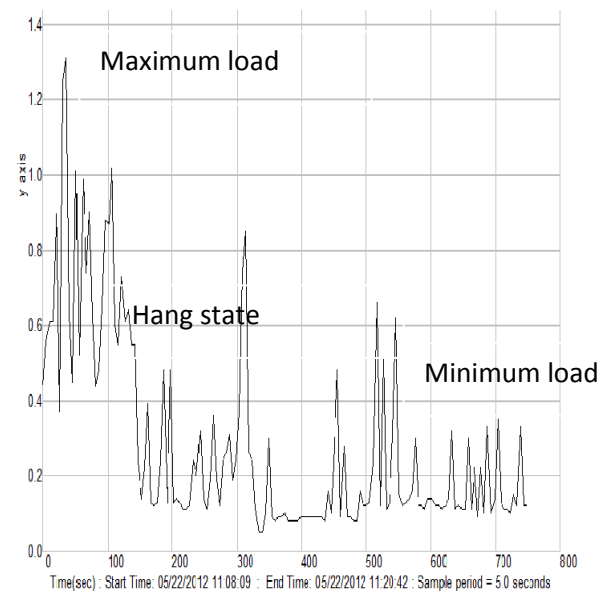
For this experiment, we are using “sysbench”[17] as benchmark tool.

## 5. RESULTS

To test the system performance in virtual environment, we generate the load by using benchmark application called “sysbench”. Side by side we also generate the load on our other virtual machines so that the effective performance can be evaluated. We generate a load on CPU with 10000 threads and also for some creating some files, simultaneously.

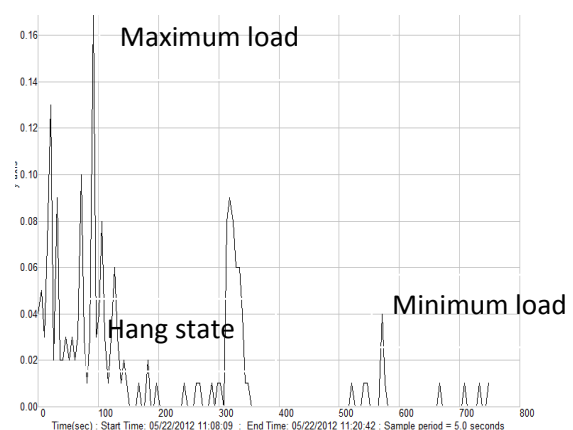
As a result, we capture the CPU performances differently as used, overlapping, running state, total system performance and can be shown graphically in following section 5.1.

### 5.1 CPU latency graph



**Fig.5.1 : CPU Graph**

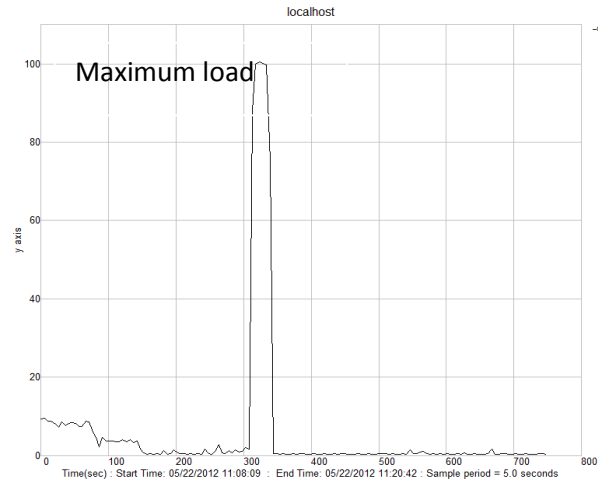
### 5.2. Overlapping



**Fig.5.2 : % overlap graph**

This graph shows the time spent by the system services on behalf of other worlds (processes).

### 5.3. Total usage

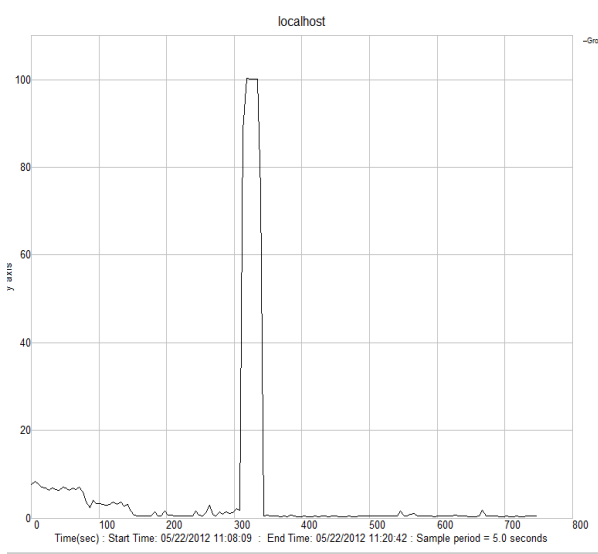


**Fig.5.3 : %used graph**

This graph shows percentage physical CPU time accounted to the world i.e. group of statistics. If a system service runs on behalf of this world, the time spent by that service (i.e. %SYS) should be charged to this world. If not, the time spent (i.e. %OVRLP) should not be charged against this world.

$$\%USED = \%RUN + \%SYS - \%OVRLP$$

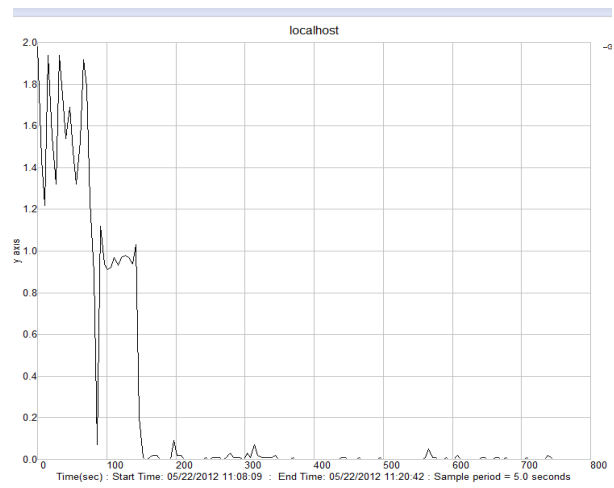
### 5.4. Running state



**Fig.5.4 : %run**

This graph shows the total scheduled time for the processes to run.

### 5.5. Total system performance



**Fig.5.5 : % system**

This graph shows the total cpu performance while maximum load, shows the maximum utilization of the system .

## 6. Conclusion and Future work

It has been shown that virtualization can play different roles in a computing system, be it for security, consolidation, etc. Once the role is known a proper VM mechanism can be chosen and evaluated allowing minimal performance impact to be achieved. Each of the virtualization methodologies presents its own benefits. In cases where the guest operating system cannot be modified for enhanced virtualization performance, only native virtualization and emulation methods may be implemented. In such cases where modification to the guest operating systems is available, the enhancements afforded by para-virtualization can and should be taken advantage of. All the mechanisms and methods must still be secured, both for each virtual machine on its own as well as the physical host. The final analysis must be then that the selection of a virtualization method for a single



physical host must be based on application based policies and reassessed as new technologies become available.

There are also many hypervisors available in market which is new to virtualization and can be used for further research. Some of the hypervisor security products can be as:

**HookSafe** is hypervisor based kernel root-kit protection system. **sHype** is a secure hypervisor system developed by IBM research

## 7. REFERENCES

- [1] Popek, J. Gerald ; Goldberg, P. Robert : “Formal requirements for virtualizable third generation architectures.” In: Commun. ACM, Nr. 7, 412-421, July 17, 1974.
- [2] J. Krich, “Virtual Machine security guidelines, the center for internet security,” September 2007.
- [3] A. Mann, “The pros and cons of virtualization,” BTQ, 2007.
- [4] Rule, David-Dittner, Rogier, “The Best Damn Virtualization Book Period. Burlington, Massachusetts. Syngress Publishing, Inc.” 2007.
- [5] “Basics of Hypervisor.”  
<http://pubs.vmware.com/vsphere-50/>
- [12] “Host System Requirements- VMware.”  
[http://www.vmware.com/support/ws5/doc/intro\\_hostreq\\_ws.html](http://www.vmware.com/support/ws5/doc/intro_hostreq_ws.html)
- [13] “ESXTOP.”  
<http://www.Yellowbricks.com>.
- [14] “SSH.”  
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- [15] “ESXPLOT.”  
<http://labs.vmware.com/flings/esxplot>
- [16] “Bechmark Tools.”  
<http://majorgeeks.com/downloads4.html>
- <http://losangelescomputerhelp.com/2010/04/th-e-different-types-of-virtualization/>
- [6] “ESXi Architecture.”  
[http://www.vmware.com/files/pdf/vmware\\_es\\_xi\\_architecture\\_wp.pdf](http://www.vmware.com/files/pdf/vmware_es_xi_architecture_wp.pdf)
- [7] “VMware Compatibility Guide.”  
<http://www.vmware.com/resources/compatibility>
- [8] S. Stephen David, Marshall, Beaver, McCarthy, W. Jason, “VMware ESX Essentials in the Virtual Data Center. Boca Raton, Florida. Taylor & Francis Group, LLC,” 2009.
- [9] VMFS, “VMware vStorage Virtual Machine File System,” A VMware Technical White Paper for VMware vSphere.
- [10] “Using the vSphere Client.”  
[http://pubs.vmware.com/vsp40\\_e/admin/c\\_using\\_the\\_vi\\_client.html](http://pubs.vmware.com/vsp40_e/admin/c_using_the_vi_client.html)
- [11] “Using the vSphere Web Client.”

[17] "Sysbench download."  
<http://sf.net/projects/sysbench/>