

Task Scheduling in virtual machines

Enroll. No. : 10503869

Name of Student : Rishi Bhardwaj

Name of supervisor(s) : Mrs. Sangeeta Mittal



JULY – 2014

**Submitted in partial fulfillment of the Degree of
5 Year Dual Degree Programme B. Tech - M. Tech**

In

Computer Science Engineering

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING &
INFORMATION TECHNOLOGY**

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

CHAPTER NUMBER	TOPICS	Page No.
	List of figures	4
	List of tables	5
	List of symbols and acronyms	5
CHAPTER 1	Introduction	6
CHAPTER 2	Some Contemporary Challenging R & D Problems In The Chosen Focus Area	7
CHAPTER 3	Literature Survey	9
	3.1 Summary of research papers	9
	3.2 Diagrammatic / Tabular Integrated summary of related literature review with respect to point	19
	3.3 Overall Conclusion of Literature Review including Synthesis & Critique of the work reviewed	21
	3.4 Implications on your project	22
	3.5 Problem Formulation and Proposed Objective	22
CHAPTER4	Analysis, Design and Modelling	22
	4.1 Overall description of the project	22
	4.2 Functional requirements and Non Functional requirements	23
	4.3 Overall architecture with component description and dependency details	24
	4.4 Proposed Algorithm	27

	4.5	Use case and Sequence diagrams of proposed solution	31
	4.6	Risk Analysis and Mitigation Plan	33
	4.7	Test Plan	36
	4.8	Implementation	38
CHAPTER 5	Appendix		
	A	Project Plan as Gantt chart or WBS	40
	B	Details of practice with new tool/technology	41
	C	References (ACM/IEEE format only)	42

LIST OF FIGURES

S. No.	Figure Name	Page No.
1.	Console	24
2.	Xentest Process	26
3.	Figure 3	26
4.	Manager collecting information of slaves	29
5.	Use case diagram for XEN configuration	31
6.	Use case diagram for load balancing	32
7.	sequence diagram	33
8.	Xen virtual machine	39
9.	Load on VMs	39
10.	Gantt Chart	40

LIST OF TABLES

S. No.	Table Name	Page No.
1.	Integrated summary of research paper	20
2	Tabular comparison between tools and software	21
3	Risk analysis	34
4	Risk mitigation plan	36
5	Test Plan	37

List of symbols and acronyms

S. No	Symbols and Acronyms
1	VM- virtual machine
2	API- Application Programming Interface
3	CPU- Central Processing Unit
4	VMM- Virtual Machine Monitor
5	PV- Paravirtualization
6	OS- Operating System
7	Dom0- domain 0
8	DomU- Domain Unprivileged

1. Introduction

1.1 Area Description

Cloud computing is a term used to refer to a model of network computing where a program or application runs on a connected server or servers rather than on a local computing device such as a PC, tablet or smartphone. Like the traditional client-server model or older mainframe computing, a user connects with a server to perform a task. The difference with cloud computing is that the computing process may run on one or many connected computers at the same time, utilizing the concept of virtualization. With virtualization, one or more physical servers can be configured and partitioned into multiple independent "virtual" servers, all functioning independently and appearing to the user to be a single physical device. Such virtual servers are in essence disassociated from their physical server, and with this added flexibility, they can be moved around and scaled up or down on the fly without affecting the end user. The computing resources have become "granular", which provides end user and operator benefits including on-demand self-service, broad access across multiple devices, resource pooling, rapid elasticity and service metering capability.

In more detail, cloud computing refers to a computing hardware machine or group of computing hardware machines commonly referred as a server or servers connected through a communication network such as the Internet, an intranet, a local area network (LAN) or wide area network (WAN). Any individual user who has permission to access the server can use the server's processing power to run an application, store data, or perform any other computing task. Therefore, instead of using a personal computer every time to run a native application, the individual can now run the application from anywhere in the world, as the server provides the processing power to the application and the server is also connected to a network via the Internet or other connection platforms to be accessed from anywhere.

But even more importantly, we have seen new technologies evolve over the past decade that are essential to the notion of the cloud. The key technology is virtualization. In addition to some amazing cost savings and goodness for the environment, virtualization's ability to separate the OS and application from the hardware give it ideal properties to best deliver these on-demand cloud services. Charles King, Principal Analyst at Pund-IT put it succinctly: "Without virtualization there is no cloud- that's what enabled the emergence of this new, sustainable industry."

2. Some contemporary challenging R & D problems in Cloud Computing

2.1 Virtual Machine Migration

Consider the case of a server with a hypervisor and several VMs, each running an OS and applications. If you need to bring down the server for maintenance (say, adding more memory to the server), you have to shut down the software components and restart them after the maintenance window—significantly affecting application availability. VM migration allows you to move an entire VM (with its contained operating system and applications) from one machine to another and continue operation of the VM on the second machine. This advantage is unique to virtualized environments because you can take down physical servers for maintenance with minimal effect on running applications. You can perform this migration after suspending the VM on the source machine, moving its attendant information to the target machine and starting it on the target machine. To lower the downtime, you can perform this migration while the VM is running (hence the name "live migration") and resuming its operation on the target machine after all the state is migrated.

2.2 Predict Application Migration in Virtual Machine

Load balancing is a technique used to schedule task on virtual machines. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Predict application migration means if we have many virtual machine then we have to open our application on that virtual machine that machine has very low workload so that we can use our full resources. In general we generally allocate the work to that machine that is next in queue. For this we have to dynamically manage our workload on all virtual machines to check whether which machine have lowest workload.

2.3 Server Consolidation

Server consolidation is an effective approach to maximize resource utilization while minimizing energy consumption in a cloud computing environment. Live VM migration technology is often used to consolidate VMs residing on multiple underutilized servers onto a single server, so that the remaining servers can be set to an energy-saving state. The problem of optimally consolidating servers in a data center is often formulated as a variant of the vector bin-packing problem, which is an NP-hard optimization problem.

2.4 Scheduling Algorithm for Cloud

Rule-based scheduling algorithms have been widely used on many cloud computing systems because they are simple and easy to implement. However, there is plenty of room to improve the performance of these algorithms, especially by using heuristic scheduling. To find better scheduling solutions for cloud computing systems. The diversity detection and improvement detection operators are employed by the proposed algorithm to dynamically determine which low-level heuristic is to be used in finding better candidate solutions.

3. Related literature review

3.1 Summary of research paper

Title of paper	Benefits and Challenges of Three Cloud Computing Service Models
Authors	Joel Gibson, Darren Eveleigh, Robin Rondeau, Qing Tan
Year of Publication	2012
Web link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6412402
Publishing Details	2012 Fourth International Conference on Computational Aspects of Social Networks
Summary	Cloud computing can be defined as the use of new or existing computing hardware and virtualization technologies to form a shared infrastructure that enables web-based value added services. The three predominant service models are infrastructure, platform, and software-as a- service. Infrastructure-as-a-Service (IaaS) can be defined as the use of servers, storage, and virtualization to enable utility like services for users. Security is a big concern within IaaS, especially considering that the rest of the cloud service models run on top of the infrastructure and related layers. Platform-as-a-Service (PaaS) providers offer access to APIs, programming languages and development middleware which allows subscribers to develop custom applications without installing or configuring the development environment. Software-as-a-Service (SaaS) gives subscribed or pay-peruse user's access to software or services which reside in the cloud and not on the user's device.
Conclusion	Cloud computing is being adopted by large organizations and presents many benefits. It has enabled collaboration amongst disparate communities and workgroups. It has overcome challenges that have plagued existing business solutions. It has allowed for access to vast amounts of computing resources in an efficient manner.

3.1 Summary of research paper

Title of paper	An Advanced Survey on Cloud Computing and State-of-the-art Research Issues
Authors	Mohiuddin Ahmed1, Abu Sina Md. Raju Chowdhury, Mustaq Ahmed, Md. Mahmudul Hasan Rafee
Year of Publication	2012
Web link	http://www.ijcsi.org/papers/IJCSI-9-1-1-201-207.pdf
Publishing Details	International Journal of Computer Science Issues
Summary	Cloud Computing is considered as one of the emerging arenas of computer science in recent times. It is providing excellent facilities to business entrepreneurs by flexible infrastructure. Although, cloud computing is facilitating the Information Technology industry, the research and development in this arena is yet to be satisfactory. This paper provides a better understanding of the cloud computing and identifies important research issues in this burgeoning area of computer science.
Conclusion	<p>Overview of cloud computing and focused on the state-of-the-art research and future issues to be handled by the research community. Recent research issues are as follows:</p> <ul style="list-style-type: none">• VM Migration• Server Consolidation• Energy Management• Information Security

3.1 Summary of research paper

Title of paper	Challenges and Opportunities of Cloud Computing
Authors	Michael Hauck, Matthias Huber, Markus Klems,
Year of Publication	2010
Web link	http://pp.info.uni-karlsruhe.de/dsci/material/TR-CC-3.pdf
Publishing Details	Karlsruhe Institute of Technology Technical Report Vol. 2010-19
Summary	Cloud Computing is a distributed computing model for enabling service-oriented, on-demand network access to rapidly scalable resources. Such resources include infrastructure as a service (IaaS), development and runtime platforms as a service (PaaS), and software and business applications as a service (SaaS). Clients do not own the resources, yet applications and data are guaranteed to be available and ubiquitously accessible by means of Web services and Web APIs “in the Cloud”.
Conclusion	<p>To predict the performance of applications running in virtualized environments during design-time, the following challenges have to be addressed:</p> <ul style="list-style-type: none">• What are the performance-relevant properties of virtualized environments that can be taken into account during design-time?• How to model software for design-time performance prediction, and how to include performance-relevant properties of virtualized environments into the model• How to integrate such performance-relevant properties into performance analyses

3.1 Summary of research paper

Title of paper	Job Scheduling Algorithm In Cloud Environment
Authors	Yihua Lan*,Xingang Zhang, Zhidu Liu, Hongbo Yu
Year of Publication	2013
Web link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6643350
Publishing Details	2013 International Conference on Computational and Information Sciences
Summary	This paper researches on the cloud computing technology systematically and comprehensively, and proposes a scheduling algorithm in the cloud environment based on the Berg model, which is a sociological justice distribution theory. This algorithm adapts to the commercialization and Virtualization features of cloud computing, differs from traditional job scheduling algorithm's character - focusing on efficiency, and establishes dual fairness constraints under the cloud environment. For tasks with multiple Qos properties, this paper gives multiple generally expect in processing tasks by applying people element analysis theory.
Conclusion	This paper research the job scheduling in cloud computing, where the dual fair constraint is established. The main element of User tasks is classified according to QoS preference. And justice in allocating sources is constrained according to the general expectance recorded in the main element of the task description. There are lots of works to do in the future which are described in the following directions, such as learning of general expected vector in job scheduling algorithms and Job scheduling algorithm based on the Berg model and so on.

3.1 Summary of research paper

Title of paper	A Model Based Task scheduling Method in IaaS Cloud
Authors	Zhenzhong Zhang, Limin Xiao, Yuan Tao
Year of Publication	2013
Web link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6687420
Publishing Details	IEEE
Summary	Many scheduling methods are based on the load forecasting model that could predict the resource requirements of the virtual machine. However, the resource requirement of virtual machine in IaaS Cloud is hard to predict because there will be variety of load types in IaaS cloud. Moreover, a Variety of heterogeneous hardware environments and virtualization technologies make it hard to predict the requirement of virtual machine based on the workloads. To address the problem, we propose a model based method to predict and calculate the resource requirement of each virtual machine and using this model to design a load balancing framework in IaaS Cloud. The contribution of this paper includes: (1) A model that forecast the load and estimate the resource requirement of virtual machines in IaaS Cloud; (2) A scalable framework for task which uses our resource requirement forecasting model. Experiments show that our method can accurately estimate the resource requirements of virtual machines, and work well in our load-balancing framework.
Conclusion	Include: (1) A model that forecast the task and estimate the resource requirement of virtual machines in IaaS Cloud; (2) A scalable framework for task scheduling using our resource requirement forecasting model. In this paper task scheduling method only considers the load of CPU and memory.

3.1 Summary of research paper

Title of paper	Load Balancing in Cloud Based on Live Migration of Virtual Machines
Authors	Raghavendra Achar, Santhi Thilagam, Nihal Soans, P. V. Vikyath
Year of Publication	2013
Web link	http://www.ijsrp.org/research_paper_jun2012/ijsrp-June-2012-29.pdf
Publishing Details	2013 Annual IEEE India Conference
Summary	Cloud computing is an upcoming trend in the field of computer science in recent years. In cloud, computing resources are provided as service in the form of virtual machine to its clients across the globe based on demand. Huge demand for cloud resources results in overutilization of servers whenever there is a heavy load. It is necessary to distribute the load across the servers in cloud by taking into consideration of allocating the right amount of resources dynamically based on the load to improve the performance of applications running in virtual machines. In this paper we present an algorithm which dynamically allocate resources based on the need and distribute the load across the servers.
Conclusion	In this paper we have presented a resource allocation algorithm to improve the performance of the applications running in virtual machine in terms of response time and distribute the load across the servers.

3.1 Summary of research paper

Title of paper	A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment
Authors	Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao
Year of Publication	2010
Web link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5715067
Publishing Details	3rd International Symposium on Parallel Architectures, Algorithms and Programming
Summary	The current virtual machine(VM) resources scheduling in cloud computing environment mainly considers the current state of the system but seldom considers system variation and historical data, which always leads to load imbalance of the system. In view of the load balancing problem in VM resources scheduling, this paper presents a scheduling strategy on load balancing of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed VM resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling.
Conclusion	In view of the current load balancing in VM resources scheduling, this paper presents a scheduling strategy on VM load balancing based on genetic algorithm. Considering the VM resources scheduling in cloud computing environment and with the advantage of genetic algorithm.

3.1 Summary of research paper

Title of paper:	An Overview of Xen Virtualization
Authors:	Abels,T. , Dhawan,P. , and Chandrasekaran,B
Year of Publication:	2005
Web link	http://www.dell.com/downloads/global/power/ps3q05-20050191-Abels.pdf
Publishing Details	Dell Document on virtualization
Summary:	<p>There are several ways to implement virtualization. Two leading approaches are full virtualization and para-virtualization. <i>Full virtualization</i> is designed to provide total abstraction of the underlying physical system and creates a complete virtual system in which the guest operating systems can execute. No modification is required in the guest OS or application; the guest OS or application is not aware of the virtualized environment so they have the capability to execute on the VM just as they would on a physical system. This approach can be advantageous because it enables complete decoupling of the software from the hardware. As a result, full virtualization can streamline the migration of applications and workloads between different physical systems. Full virtualization also helps provide complete isolation of different applications, which helps make this approach highly secure.</p>
Conclusion	<p>In contrast, <i>para-virtualization</i> presents each VM with an abstraction of the hardware that is similar but not identical to the underlying physical hardware. Para-virtualization techniques require modifications to the guest operating systems that are running on the VMs. As a result, the guest operating systems are aware that they are executing on a VM—allowing for near-native performance. Architecture of various Xen versions were also presented</p>

3.1 Summary of research paper

Title of paper:	Using CloudSim to Model and Simulate Cloud Computing Environment
Authors:	Wang Long, Lan Yuqing and Xia Qingxin
Year of Publication:	2013
Web link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6746411
Publishing Details	2013 Ninth International Conference on Computational Intelligence and Security
Summary:	<p>Cloud computing will be a major technology in the development of the future Internet of Services, and it delivers infrastructure, platform, and software that are made available as subscription-based services in a pay-as-you-go model to customers. Data is neither stored on the local hard drive of your computer, nor on servers that are down in the basement of your company. Instead it is out in the cloud. The infrastructure is outside of your organization and you access the applications, the infrastructure, and all those services typically through the web-based interface. Customers can accept the cloud system everywhere and receive sufficient computing environments. These cloud service are divided into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). And, cloud computing has features of the distributed computing environment such as grid computing system. Therefore, if the cloud user pays appropriate costs, it is able to create a computing environment which is nearly infinite.</p>
Conclusion	<p>Using CloudSim to simulate a cloud computing datacenter avoids spending time and effort to configure a real testing environment. This paper introduced the CloudSim simulator including its architecture, and how to use it to model the cloud environment. We will use CloudSim platform to evaluate algorithm which aim at improve the average resource utilization of the cloud datacenter and reduce the energy consumption.</p>

3.2 Diagrammatic / Tabular Integrated summary of related literature review with respect to point

<u>No.</u>	<u>Research Paper</u>	<u>Issue /Challenges</u>	<u>Implementation / Solution</u>	<u>Tool used</u>
1	Benefits and Challenges of Three Cloud Computing Service Models	Proposed issue related to IaaS, PaaS, and SaaS.	<ul style="list-style-type: none"> • Selection of cloud • Security 	None
2	An Advanced Survey on Cloud Computing and State-of-the-art Research Issues	Architectural Design of Data Centre Distributed File System over Cloud Distributed Application Framework over Clouds	<ul style="list-style-type: none"> • VMM Migration • Server consolidation • Information Security 	None
3	Challenges and Opportunities of Cloud Computing	Performance related solution in virtualized environment	Performance of applications running in virtualized environments	None
4	Job Scheduling Algorithm In Cloud Environment	Algorithm issue in cloud environment	Berg Model	Cloudsim
5	Live migration in virtual network environment	Guaranteed load balancing	Memory-compression based VM migration approach	None
6	A Model Based Task scheduling Method in IaaS Cloud	hard to predict the requirement of virtual machine based on the workloads	model that forecast the load and estimate the resource requirement	Xen

7	A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment	Seldom considers system variation and historical data for load balancing and scheduling	Genetic algorithm for load balancing	Platform ISF OpenNebula
8	Predictive VM consolidation on multiple resources: Beyond load balancing	Unpredictable system and workload delays	<ul style="list-style-type: none"> Fair load balancing scheme Queueing network analytic model 	Xen virtualization testbed
9	An Overview of Xen Virtualization	Virtualization overview		Xen
10	Using CloudSim to Model and Simulate Cloud Computing Environment		Use of cloudsim to avoid configuration of real virtualization environment	cloudsim

Table I

Tabular comparison between tools and software:-

<u>Tool</u>	<u>Benefit/Advantages</u>	<u>Drawbacks/limitation</u>
Xen	<ul style="list-style-type: none"> Open Source Mostly used Type 1 hypervisor Real system 	<ul style="list-style-type: none"> Tough to configure No GUI
CloudSim	<ul style="list-style-type: none"> Easy to setup User friendly 	<ul style="list-style-type: none"> Simulator based
Virtual Box	<ul style="list-style-type: none"> Easy setup with GUI 	<ul style="list-style-type: none"> No privilege
Citrix	<ul style="list-style-type: none"> Hardware Assisted Virtualization Paravirtualization 	<ul style="list-style-type: none"> Less privilege

Table II

3.3 Overall Conclusion of Literature Review

Cloud computing is a way of leveraging the Internet to consume software or other IT services on demand. Users share processing power, storage space, bandwidth, memory, and software. With cloud computing, the resources are shared and so are the costs. Users can pay as they go and only use what they need at any given time, keeping cost to the user down.

Load balancing is a technique used to schedule task on virtual machines. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

Xen runs in a more privileged CPU state than any other software on the machine. Responsibilities of the hypervisor include memory management and CPU scheduling of all virtual machines ("domains"), and for launching the most privileged domain ("dom0") - the only virtual machine which by default has direct access to hardware. From the dom0 the hypervisor can be managed and unprivileged domains ("DomU") can be launched.

The dom0 domain is typically a version of Linux, or BSD. User domains may either be traditional operating systems, such as Microsoft Windows under which privileged instructions are provided by hardware virtualization instructions (if the host processor supports x86 virtualization, e.g., Intel VT-x and AMD-V), or *para-virtualized* operating system whereby the operating system is aware that it is running inside a virtual machine, and so makes hypercalls directly, rather than issuing privileged instructions. Xen boots from a bootloader such as GNU GRUB, and then usually loads a paravirtualized host operating system into the host domain (dom0).

3.4 Implication on project

- Task technique can be used in many datacenters because in datacenters we have data and application that need to be computed so with the scheduling the task we can balance the load of virtual machine without manually.
- With our application client not need to select manually that in which VM our client application should run, application will be automatically run on VM according to CPU utilization.

3.5 Problem Formulation and Proposed Objective & Motivation of work

Problem Formulation:-

Task scheduling in virtual machine in now days a very big problem. My problem is to implement the preexisting Task scheduling algorithm on virtual environment and also develop new approach to balance load in virtual machine. It will be useful in many datacenters to balance load effectively on servers.

4. Analysis, Design and Modelling

4.1 Overall description of project

Task scheduling is a technique used to schedule task on virtual machines. Task scheduling balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Application Load Balancer on Xen is a very important tool in today's world as we are completing transferring our work on cloud platforms so scheduling of task properly on various servers is very important. Our tool helps the client as the application requested by client is directly transferred to the most suitable server according to its need. So by use of this the client does not need to manually select the server. In general we generally allocate the work to that machine that is next in queue. For this we have to dynamically manage our workload on all virtual machines to check weather which machine have lowest workload. And as a complete check on load, memory of various servers is maintained from time to time so we have

the complete status of all the servers all the time. By this we are able to utilize the resources completely and efficiently. Thus, the throughput and response time is improved efficiently.

4.2 Functional requirements and Non Functional requirements

Functional Requirements

- Multiple guest OS can run on host OS via Xen virtual machine monitor.
- The resource usage of any application to be calculated so as to profile the applications based on their resource usage.
- Live migration between virtual machines so as to migrate VM in case of any failure or allocation of an application to a specific VM.
- Different characteristics to be given to different virtual machines and then the applications to be run on these VMs accordingly.
- View all running VMs and hosts and their live performance & resource utilization statistics.
- see performance & utilization statistics for each VM

Non Functional Requirements

- Response time- This requirement describes how much time it takes from the moment a user sends a request to the system, until a complete response is provided. In web applications, this comprehends request transmission and processing, and response transmission. The factors that account for it are resource capabilities —processing power, memory, disk, network latency and bandwidth— and the load produced by other processes running in the server or the number of concurrent requests. For complex requests, this may also involve calls to external systems, or to other subsystems, in which case the host's internal network characteristics and other resources' load may be taken into account.
- Uptime- The total time the service is available. It may be expressed as a percentage. When considering this requirement, it is necessary to take into account the provider's own uptime. For example, if a provider has an uptime of 99.5%, it would be impossible to deploy an application with a higher uptime. Other factors involve the recoverability of the system (i.e., how much time it takes to restart the service after a failure happens).

- Requests per unit of time- This requirement describes the number of requests the system can handle successfully per unit of time, and can also be referred to as the system's throughput. Resource allocation and usage has an impact in this parameter. Additionally, the number of requests can have an impact in the response time requirement (i.e., a high number of requests will result in a deterioration of the overall response time). Fault tolerance. One of the system's properties is how it can withstand errors, either hardware or software-based. In the case of cloud, non-software errors can be generated either at the physical or the virtual machines hosting the service.
- Security- Security is another requirement that can be applied to the cloud provider or to the developed system.

4.3 Overall architecture with component description and dependency details

The Xen hypervisor runs directly on the hardware and is responsible for handling CPU, Memory, and interrupts. It is the first program running after exiting the bootloader. On top of Xen run a number of virtual machines, a running instance of a virtual machine in Xen is called a **domain** or **guest**. A special domain, called domain 0 contains the drivers for all the devices in the system. Domain 0 also contains a control stack to manage virtual machine creation, destruction, and configuration.

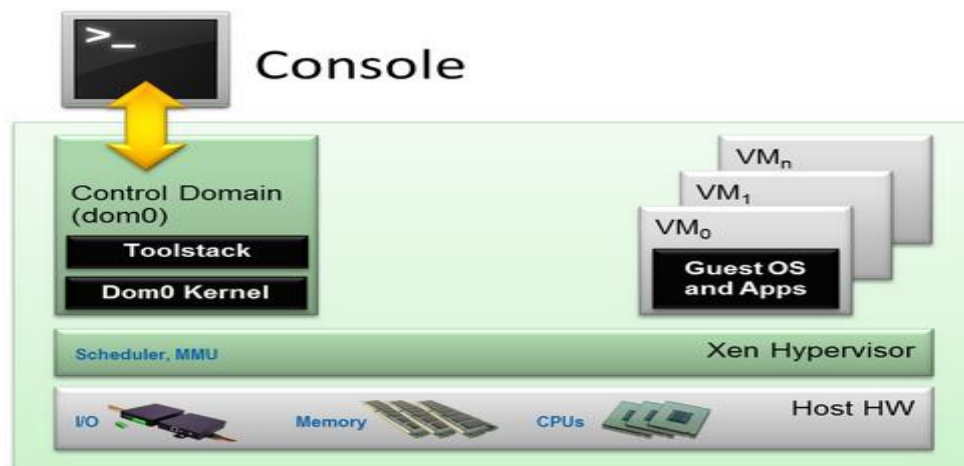
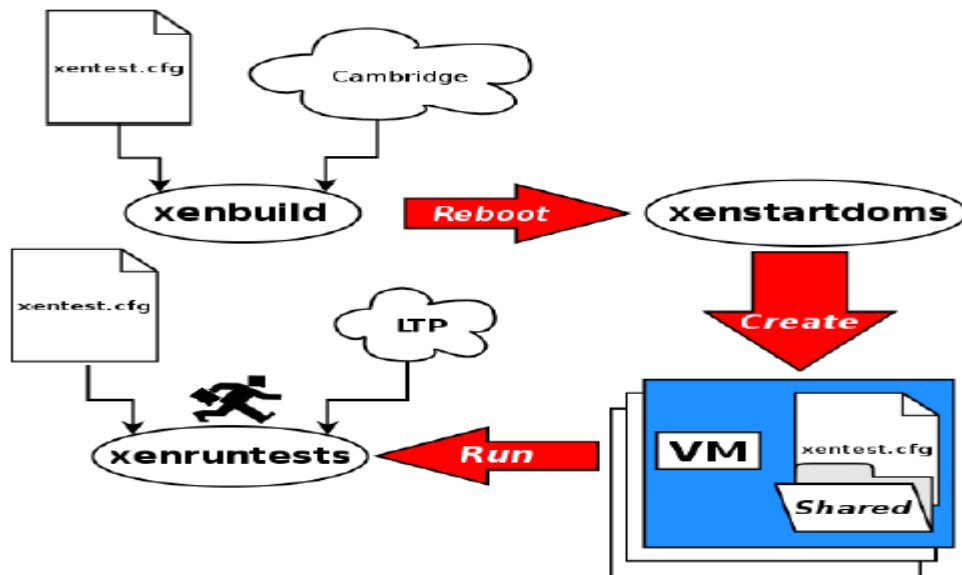


Fig. 1

- **The Xen Hypervisor** is an exceptionally lean (<150,000 lines of code) software layer that runs directly on the hardware and is responsible for managing CPU, memory, and interrupts. It is the first program running after the bootloader exits. The hypervisor itself has no knowledge of I/O functions such as networking and storage.
- **Guest Domains/Virtual Machines** are virtualized environments, each running their own operating system and applications. Xen supports two different virtualization modes: Paravirtualization (PV) and Hardware-assisted or Full Virtualization (HVM). Both guest types can be used at the same time on a single Xen system. It is also possible to use techniques used for Paravirtualization in an HVM guest: essentially creating a continuum between PV and HVM. This approach is called PV on HVM. Xen guests are totally isolated from the hardware: in other words, they have no privilege to access hardware or I/O functionality. Thus, they are also called unprivileged domain (or DomU).
- **The Control Domain (or Domain 0)** is a specialized Virtual Machine that has special privileges like the capability to access the hardware directly, handles all access to the system's I/O functions and interacts with the other Virtual Machines. It also exposes a control interface to the outside world, through which the system is controlled. The Xen hypervisor is not usable without Domain 0, which is the first VM started by the system.
- **Toolstack and Console:** Domain 0 contains a control stack (also called Toolstack) that allows a user to manage virtual machine creation, destruction, and configuration. The toolstack exposes an interface that is either driven by a command line console, by a graphical interface or by a cloud orchestration stack such as OpenStack or CloudStack.
- **Xen-enabled operating systems:** A Xen Domain 0 requires a Xen-enabled kernel. Paravirtualized guests require a PV-enabled kernel. Linux distributions that are based on recent Linux kernel are Xen-enabled and usually contain packages that contain the Xen Hypervisor and Tools Xen (the default Toolstack and Console). All but legacy Linux kernels are PV-enabled: in other words, they will run Xen PV guests.



Xentest process

Figure 2

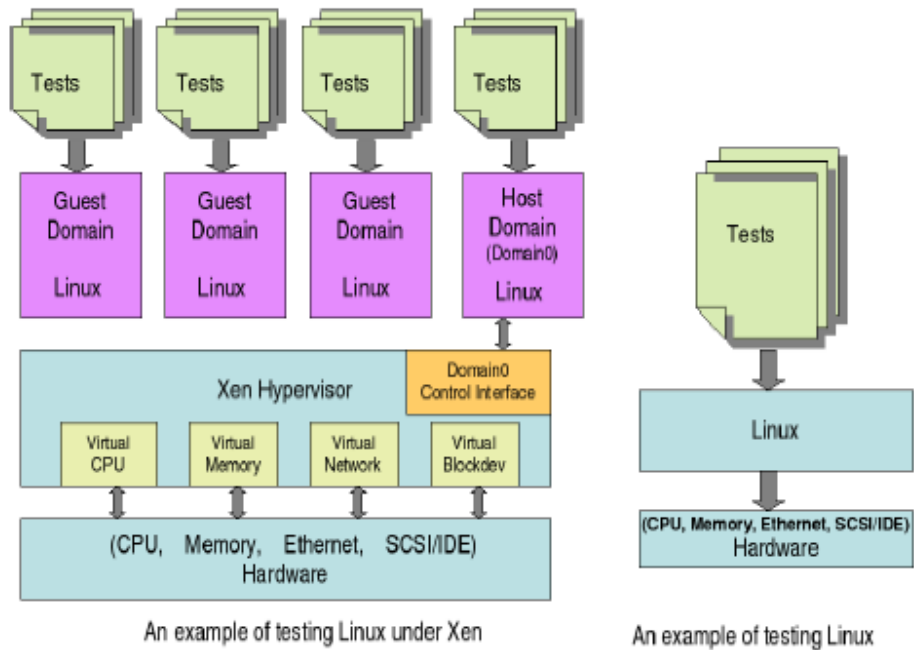


Figure 3

Guest type

Xen Project Paravirtualization (PV)

Paravirtualization is an efficient and lightweight virtualization technique originally introduced by Xen Project, later adopted by other virtualization platforms. PV does not require virtualization extensions from the host CPU. However, paravirtualized guests require a PV-enabled kernel and PV drivers, so the guests are aware of the hypervisor and can run efficiently without emulation or virtual emulated hardware. PV-enabled kernels exist for Linux, NetBSD, FreeBSD and OpenSolaris. Linux kernels have been PV-enabled from 2.6.24 using the Linux pvops framework. In practice this means that PV will work with most Linux distributions (with the exception of very old versions of distros).

Xen Project Full Virtualization (HVM)

Full Virtualization or Hardware-assisted virtualization uses virtualization extensions from the host CPU to virtualize guests. HVM requires Intel VT or AMD-V hardware extensions. The Xen Project software uses Qemu to emulate PC hardware, including BIOS, IDE disk controller, VGA graphic adapter, USB controller, network adapter etc. Virtualization hardware extensions are used to boost performance of the emulation. Fully virtualized guests do not require any kernel support. This means that Windows operating systems can be used as a Xen Project HVM guest. Fully virtualized guests are usually slower than paravirtualized guests, because of the required emulation.

4.4 Proposed Algorithm

Task scheduling Algorithm

Static Scheduling

This scheduling algorithm that distributes the load based strictly on a fixed set of rules relating to the characteristics of input load. It does not consider which node is receiving more or less load. In all static algorithms final selection of the virtual machine is done as soon as the application is created and cannot be altered during execution to change the load of that virtual machine. These static load balancing techniques are suitable for a system in which load is limited and request of the clients is also limited. But nowadays load on cloud servers is also increasing and also the load is not static hence we need more efficient algorithms than static load balancing algorithms.

Round Robin Algorithms

Whenever a new application comes it is assigned a virtual machine in a round robin fashion. In general, basic idea for Round Robin is to reduce message passing between various virtual machines and reduce communication delay. Thus it is independent of the state of the system. When coming applications are of similar load then Round Robin works very well as it reduces the communication delay due to inter-process communication. Thus Round Robin has best performance for this special purpose application of similar load, but does not give a good performance for general cases.

Randomized Algorithm

Random numbers are distributed on a basis of a statistical distribution and assigned to virtual machines. Incoming applications are distributed according to these randomly generated numbers. This algorithm is applicable when we have many virtual machines as compared to processes

Central Manager Algorithm

In this algorithm, we have a central virtual machine and others are slave virtual machines which are assigned applications to be executed. Central virtual machine's task is to gather load information of all the slave virtual machines and assign the coming application to the least loaded slave virtual machine.

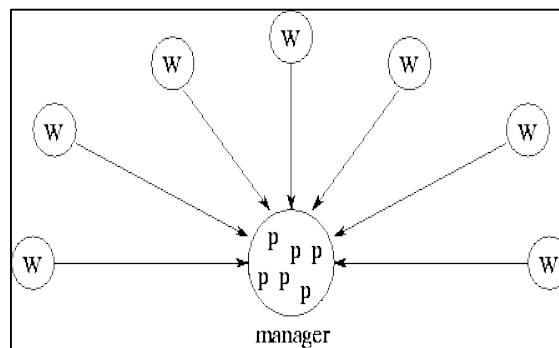


Figure 4: Manager collecting information of slaves

Dynamic Scheduling

This techniques provide a method to dynamically allocate task based on self-adapting distribution and intelligent distribution. Here, it is distributed at runtime based on the new information collected. Mainly these techniques are based on greedy algorithmic approaches.

Central Queue Algorithm

The host virtual machine maintains a central queue of all the applications. This queue is shared by all the processes. New applications are added and pending applications are maintained in a cyclic FIFO order in the queue. When a virtual machine is free it will request for application for executing to the host and host assigns the application next in queue to the virtual machine which is demanding the request. If there is no application for execution in the queue then the request is buffered in queue form. And request is answered when new application arrives.

Load Queue Algorithm

Here the applications are assigned virtual machines similar to the static algorithm but a virtual machine here can initiate application migration. Initially all under loaded virtual machines are assigned the applications. The applications are assigned following some static algorithm. Then if a virtual machine's load goes below the lower threshold value then it asks for load from other machines and initiates migration process.

Proposed algorithm

This algorithm repeats itself after an interval of time 'k'.

- 1) After time k the dom0 as updated load status of each VM
- 2) Dom0 machine will read the load
- 3) $A := \text{load of vm1}$
- 4) $B := \text{load of vm2}$
- 5) $C := \text{load of vm3}$
- 6) if($A < B \ \&\& \ A < C$)
- 7) then
- 8) allocate task to VM A
- 9) if ($B < A \ \&\& \ B < C$)
- 10) then
- 11) allocate task to VM B
- 12) else
- 13) allocate load to VM C
- 14) end if
- 15) end if

This algorithm is proposed for in case we have small number of virtual machine, if we have large number of virtual machine then we will try to optimize it.

4.5 Use case and Sequence diagrams of proposed solution

Use case diagram for XEN configuration and virtual machine installation

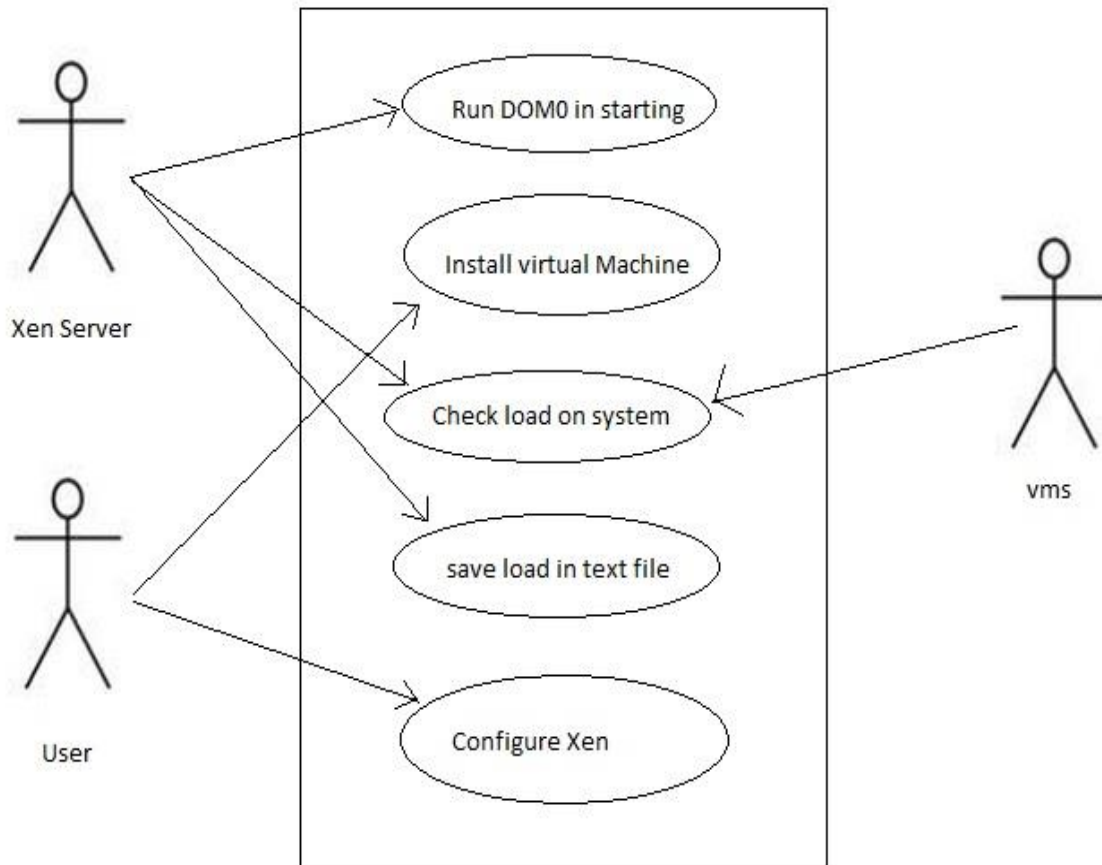


Fig. 5 Use case diagram for XEN configuration

- XEN server start at the time of booting Ubuntu it starts the Dom0 which is a preconfigured virtual machine and automatically start at the time of boot.
- For installing virtual machine on server user can use VMM.
- Check load on system function can be used to check the complete load on system in server.
- Save load to txt file function can be used to save the load in text file.

Use case diagram for scheduling

This diagram shows the all the functions that will our task scheduler will do.

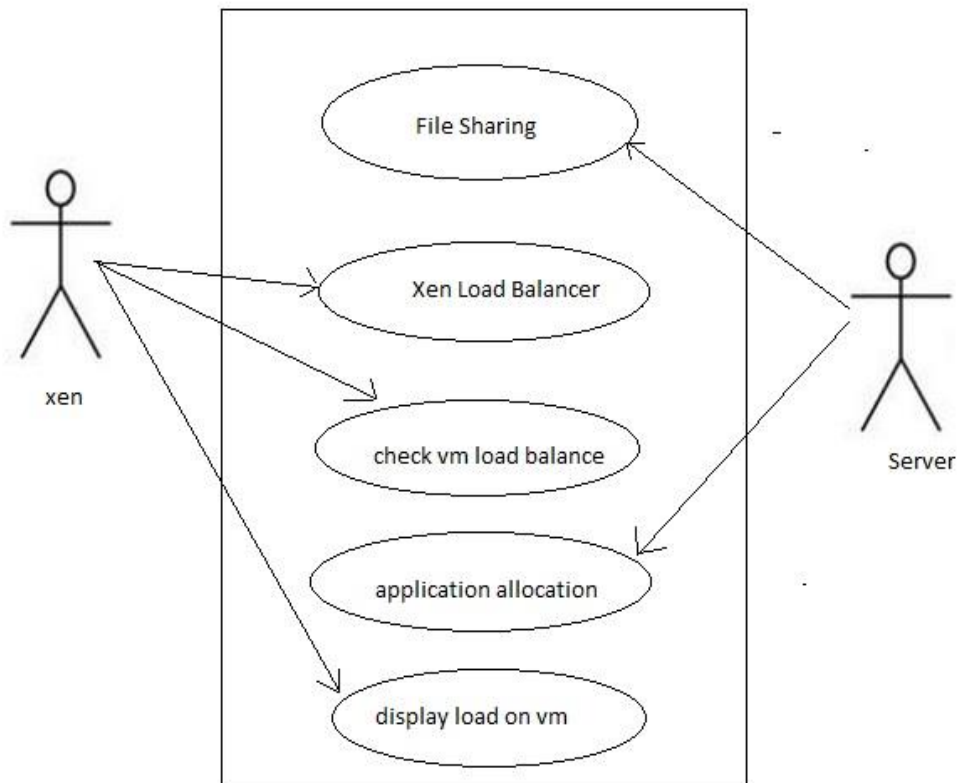


Fig. 6 Use case diagram for task scheduling

- File sharing will be used to create a network supported shared file system in this all the VMs will be able to access a common space.
- Xen task scheduler will be used to balance the load between the virtual machines.
- Application allocation function will allocate the desired application to that VM which has less resources.

Sequence Diagram:-

Sequence diagram for load balancing

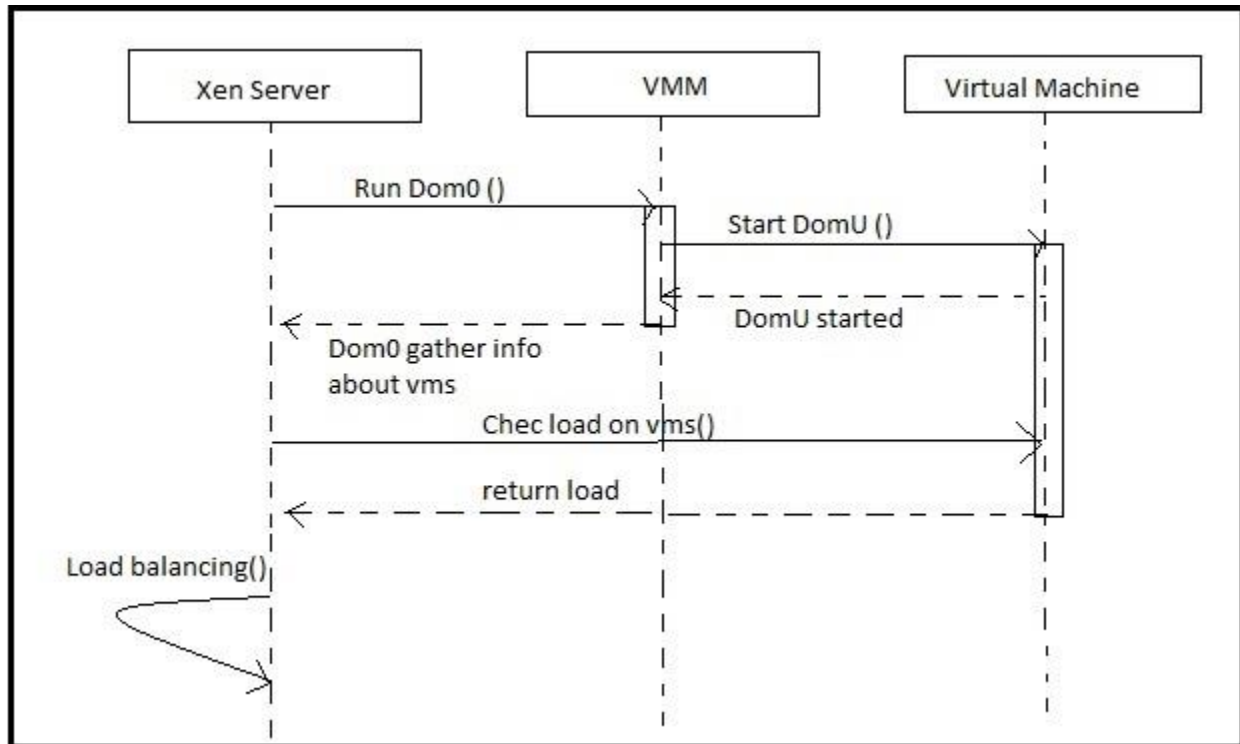


Fig. 7 sequence diagram

4.6 Risk Analysis and Mitigation Plan

Risk Id	Description of Risk	Risk Area (Identify Risk Areas for your project)	Probability (P)	Impact (I)	RE (P*I)
1	Excessive recourse requirement by underlying operating system- Working on XEN software for management of resources, heavy computations are	Requirement Risk	Medium(3)	High(5)	15

	required. But, underlying operating should never take more than reasonable amount of resource otherwise system will crash.				
2	Lack of similar platform- As this project will be on XEN platform, there will be problems, if it's used on an hypervisor other than XEN	Hardware	Low(1)	Low(1)	1
3	High CPU requirement by various applications- There is complex and big requirement for the application. Due to unavailability of resources or server, we can't use the application.	Hardware	Medium(3)	Medium(3)	9
4	Patching- We will never be able to check if patch is correct for large applications or not, for large inputs, as these applications would require high CPU resources. Although, this risk can be reduced by running the application many number of times.	Performance	Low(1)	High(5)	5

5	Crashing- Any time the system can crash and the data can be lost	Performance and Hardware	High(5)	High(5)	25
6	Excessive evolution of problem statement → delay in project deliverable	Project Scope	High(25)	High(25)	25
7	Wide scope of functionality of project → unrealistic expectations from the project	Project Scope	Low(1)	High(5)	5
8	Irregularity → schedule will be affected	Personnel Related	High(5)	High(5)	25
9	Incorrect system requirements → program doesn't run successfully	Requirement Risk			
10	Software packages integration → part of code doesn't run	Development Environment Risk	High(5)	High(5)	25

Table III

RISK MITIGATION PLAN

S. No	Risk	Mitigation Approach	Additional Resources needed for mitigation
1.	Risk of using a new language and technology (Xen)	Choosing a language and interface which is known	Install Ubuntu, cloudsim
2.	Risk of change in Project plan	Program should follow modularity	
3.	Risk of non-availability hardware support	Explore ways of hypervisor implementation	Intel Virtualisation Technology explored
4.	Risk of variable output		
5.	Risk of changing task scheduling algorithm	Implementing pre-existing algorithm	Experimentation

Table IV

4.7 Test Plan

Type Of Test	Test Performed?	Comments	Software Component
Unit Testing	Yes	The application is mainly divided into the main components – Xen Hypervisor, Virtual machines, migrating environment, profiling.	Profiling patch, hypervisor, virtual machine after running different operating systems on them.
Integration Testing	Yes	The application's modules are inter- Dependent. 'Dom u' host	Integration of hypervisor with all

		depends upon 'Dom0' host And eventually Dom0 depends on the underlying hardware	the virtual machines, profilation patch and global manager.
Performance Testing	Yes	All the virtual machines should be test separately and performance on correspondent domains should checked.	Performance of every virtual machine should be done.
Stress testing	No	At this stage, we have not identified stress points for this application as feature development is higher priority.	
Security Testing	Yes	As hypervisor is the underlying managing body under virtual machines. Hence any hacker can enter hypervisor and hence can cause huge damage to the system	Several components were tested due to high risk involved.
Load Testing	Yes	Load Testing is important at this stage As we are dealing with a variety of virtual machines and multiple operating systems hence load testing in very important.	Underling hardware was tested along with load on virtual machines.

Table V

4.8 Implementation:-

XEN Server Configuration:-

For project propose I have configured the Xen server in Ubuntu 12.04 LTS version. Xen is a native (bare-metal) hypervisor providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

Install Xen Package

```
$ sudo apt-cache search xen
```

```
$ sudo apt-get install xen-hypervisor-4.1-amd64 xen-utils-4.1 xen-tools xen-docs-4.1
```

Install libvirt and Virtual Manager Tools

```
$ sudo apt-cache search virt-install
```

```
$ sudo apt-get install virtinst python-libvirt virt-viewer virt-manager
```

Create Xen Virtual Machine Images

```
$ sudo dd if=/dev/zero of=/etc/xen/vm01.img bs=1M count=3072
```

```
3072+0 records in
```

```
3072+0 records out
```

```
3221225472 bytes (3.2 GB) copied, 39.5561 s, 81.4 MB/s
```

View Dom0 Current Status

```
$ sudo xm list
```

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	945	2	r-----	73.1

```
$ sudo mv /etc/grub.d/10_linux /etc/grub.d/50_linux
```

```
$ sudo update-grub2
```

```
$ sudo reboot
```

Modify the Configuration File

```
$ sudo vi /etc/xen/xend-config.sxp
```

Modify #(xend-unix-server no) to (xend-unix-server yes), save the file and then restart the xend service

```
$ sudo xend restart
```

XEN Virtual Machine

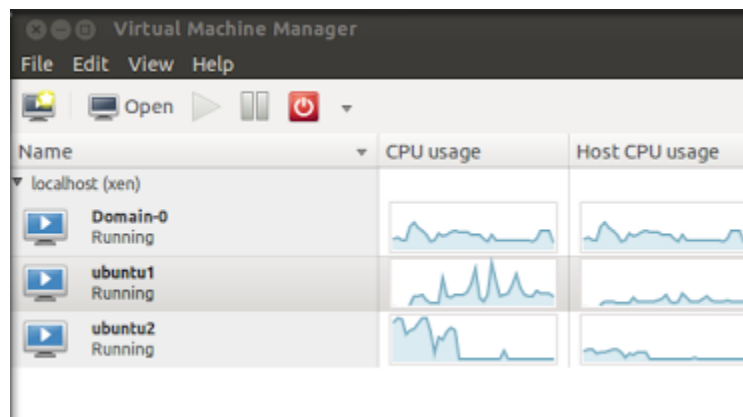


Figure 8

Load on each virtual machine:-

	NAME	STATE	CPU(sec)	CPU(%)	MEM(k)	MEM(%)	MAXMEM(k)	MAXMEM(%)	VCPUS
NETS	NETTX(k)	NETRX(k)	VBDS	VBD_OO	VBD_RD	VBD_WR	VBD_RSECT	VBD_WSECT	SS
ID									
0	Domain-0	-----r	220	0.0	2928648	72.2	no limit	n/a	4
0	0	0	0	0	0	0	0	0	0
0	ubuntu1	--b---	7	0.0	528336	13.0	528384	13.0	1
0	1	6	1	0	5553	146	143718	3824	1
0	ubuntu2	--b---	86	0.0	528336	13.0	528384	13.0	1
0	1	506	43	0	139560	5693	4034950	392752	1
0									

root@Dell:/home/rishi/Desktop#

Figure 9

Appendix

A. Project Plan as Gantt chart or WBS

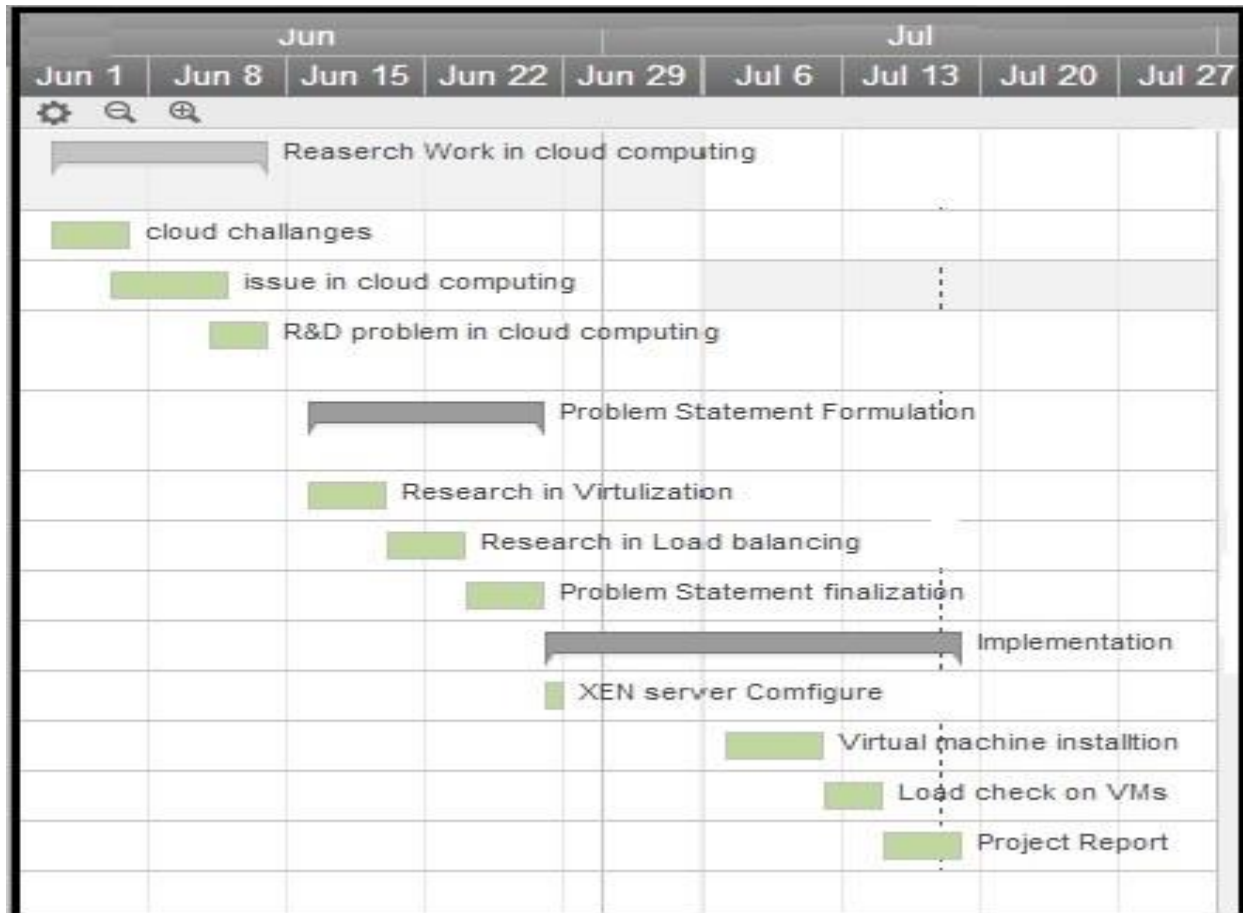


Figure 10

B. Details of practice with new tool/technology

XEN:-

The Xen Project hypervisor is an open-source type-1 or baremetal hypervisor, which makes it possible to run many instances of an operating system or indeed different operating systems in parallel on a single machine (or host). The Xen Project hypervisor is the only type-1 hypervisor that is available as open source. It is used as the basis for a number of different commercial and open source applications, such as: server virtualization, Infrastructure as a Service (IaaS), desktop virtualization, security applications, embedded and hardware appliances. The Xen Project hypervisor is powering the largest clouds in production today.

Here are some of the Xen Project hypervisor's key features:

- Small footprint and interface (is around 1MB in size). Because it uses a microkernel design, with a small memory footprint and limited interface to the guest, it is more robust and secure than other hypervisors.
- Operating system agnostic: Most installations run with Linux as the main control stack (aka "domain 0"). But a number of other operating systems can be used instead, including NetBSD and OpenSolaris.
- Paravirtualization: Fully paravirtualized guests have been optimized to run as a virtual machine. This allows the guests to run much faster than with hardware extensions (HVM). Additionally, the hypervisor can run on hardware that doesn't support virtualization extensions.

Python:-

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large standard library.

C. References

- [1] Joel Gibson, Darren Eveleigh, Robin Rondeau, Qing Tan, “Benefits and Challenges of Three Cloud Computing Service Models”, 2012 Fourth International Conference on Computational Aspects of Social Networks.
- [2] Mohiuddin Ahmed¹, Abu Sina Md. Raju Chowdhury, Mustaq Ahmed, Md. Mahmudul Hasan Rafee, “An Advanced Survey on Cloud Computing and State-of-the-art Research Issues”, International Journal of Computer Science Issues.
- [3] Michael Hauck, Matthias Huber, Markus Klems, “Challenges and Opportunities of Cloud Computing”, Karlsruhe Institute of Technology Technical Report Vol. 2010-19.
- [4] Wang Xiaojing, Tong Wei, Zhao Wei, Liu Jingning, “Evaluation on network load balancing in Xen”, 2012.
- [5] Maha Jebalia, Asma Ben letaifa, Sami tabbane, “Live migration in virtual network environment”, 2010, IEEE.
- [6] Raghavendra Achar, Santhi Thilagam, Nihal Soans, P. V. Vikyath, “Load Balancing in Cloud Based on Live Migration of Virtual Machines”, 2013, 2013 Annual IEEE India Conference.
- [7] Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, 2010, “3rd International Symposium on Parallel Architectures, Algorithms and Programming”.
- [8] Lei Lu, Hui Zhang, Smirni, E, Guofei Jiang, Yoshihira, K., “Predictive VM consolidation on multiple resources: Beyond load balancing”, 2013, Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium.
- [9] Abels,T. , Dhawan,P. , and Chandrasekaran,B , “An Overview of Xen Virtualization”, 2005, “Dell Document on virtualization”.

[10] Wang Long, Lan Yuqing and Xia Qingxin,” Using CloudSim to Model and Simulate Cloud Computing Environment”, 2013, 2013 Ninth International Conference on Computational Intelligence and Security.