

An Energy Efficient Virtual Machine Allocation Policy in Cloud Environment

*Thesis submitted in partial fulfillment of the requirements for the award
of degree of*

**Master of Engineering
in
Software Engineering**

Submitted By
**Charanpreet Kaur
(801231006)**

Under the supervision of:
**Dr. Seema Bawa
Professor, Dean (Student Affairs)
CSED**



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

Certificate


Certificate


I hereby certify that the work which is being presented in the thesis entitled, "An Energy Efficient Virtual Machine Allocation policy in Cloud Environment", is partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering admitted in Computer Science and Engineering Department of Pimpri Chinchwad Education Trust, Pimpri, Maharashtra, India, is an authentic work of my own work carried out under the supervision of Dr. Anand Kulkarni and others whose names are duly listed in the reference section.

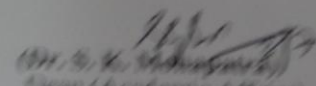
The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.


(Candidate)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.


(Dr. Anand Kulkarni)
Professor, Head (Student Affairs)
CSE Department
Pimpri Chinchwad Education Trust
Pimpri, Maharashtra, India


(Dr. Anand Kulkarni)
Head
Computer Science and Engineering Department
Pimpri Chinchwad Education Trust
Pimpri, Maharashtra, India


(Dr. S. K. Mahapatra)
Dean (Academic Affairs)
Pimpri Chinchwad Education Trust
Pimpri, Maharashtra, India

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. It is a great privilege to express my gratitude and admiration towards my respected supervisor **Dr. Seema Bawa** Professor Computer Science & Engineering Department. She has been an esteemed guide and great support behind achieving this task. This work would not have been possible without the encouragement and able guidance of her. I also thank my supervisor for her time, patience, discussions and valuable comments. Her enthusiasm and optimism made this experience both rewarding and enjoyable. I am truly grateful to her for extending her total co-operation and understanding whenever I needed help and guidance from her.

I am also heartily thankful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department and **Dr. Damandeep Kaur**, PG coordinator, for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I will be failing in my duty if I do not express my gratitude to **Dr. S. K. Mohapatra**, Senior Professor and Dean of Academic Affairs, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field. I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family for their wonderful love and encouragement, without their blessings none of this would have been possible.

Charanpreet Kaur
(801231006)

Abstract

Data centers hosting Cloud applications consume huge amounts of electrical energy, contributing to high operational costs of the system. By supporting the movement of VMs between physical nodes, it enables dynamic migration of virtual machine according to the performance requirements and thus the nodes that are idle can be switched off to eliminate power consumption. The proper VM allocation and selection policy is required to maximize the usage of host still maintaining the service level agreements. Many allocation policies such as IQR, MAD etc are there, but such policies consider only 2/3rd of whole data set. These policies also increase the computation overhead. This thesis focused on the enhancement of virtual machine (VM) allocation policies in such a way that energy consumption, number of VM migrations and SLA violation can be reduced. Central Limit theorem in conjunction with minimum migration time selection policy has been used to achieve the expected results. Proof of correctness is derived in order to conform that the proposed algorithm is a novel approach. Validation of proposed algorithm has been done by conducting a performance evaluation study using the Cloud Sim 3.0 simulator. Testing results show that the proposed algorithm is better as compared to IQR, MAD allocation policies in terms of energy efficiency, number of VM migrations and SLA violation.

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
List of Tables	vii
Chapter 1 Introduction	1
1.1. Evolution of Cloud Computing	3
1.2. Cloud Computing	4
1.2.1 Five Essential Characteristics of Cloud Computing	5
1.2.2 Deployment Models	5
1.2.3 Service Models	6
1.3 Energy Efficiency	7
1.3.1 Power Model	8
1.3.2 Energy Consumption as a Problem in DataCenters	9
1.3.3 Efficient Ways of Energy Consumption	10
1.3.4 Green Cloud Architectural Element	12
1.4 Virtualization	15
1.4.1 Need of Virtualization	16
1.4.2 Types of Virtualization	17
1.5 Virtual Machine Migration	19
1.5.1 Need of Virtual Machine Migration	19
1.5.2 Types of Virtual Machine Migration	20
1.5.3 Side Effects of Virtual Machine Migration	21
1.5.4 Why Virtual Machine Selection has been selected	21
Chapter 2 Literature Survey	22
2.1 Virtual Machine Selection	22
2.2 Virtual Machine Migration	24
2.2.1 Pre-Copy Approach	25
2.2.2 Improvement over Pre-Copy Approach	26

2.2.3 Post-Copy Approach	28
2.2.4 CR/TR Approach	28
2.3 Virtual Machine Selection	29
2.4 Energy Efficiency	30
Chapter 3 Problem Statement: VM Allocation Policy	39
3.1 Proposed Algorithm	40
Chapter 4 Implementation Details	41
4.1 Experimental Setup	41
4.2 Proof of Correctness	41
4.3 Snapshots of Implementation	48
Chapter 5 Testing and Results	54
5.1 Test Plan	54
5.2 Testing	54
Chapter 6 Conclusion and Future Scope	58
6.1 Conclusion	58
6.2 Future Scope	58
References	59
List of Publications	64

List of Figures

Figure 1.1 The worldwide data center energy consumption 2000-2010	2
Figure 1.2 Power consumption by server components	8
Figure 1.3 High level system view	12
Figure 1.4 High-level system architectural framework	13
Figure 1.5 Types of hypervisor	15
Figure 1.6 Types of Virtualization	17
Figure 1.7 Virtualized server with live migration capability	20
Figure 2.1 Pre-copy approach	26
Figure 2.2 CR/TR approach	29
Figure 4.1 Main screen	48
Figure 4.2 Entering into datacenter	48
Figure 4.3 Initial stage	49
Figure 4.4 No resource allocation	49
Figure 4.5 Task allocation	50
Figure 4.6 VM details	50
Figure 4.7 Host details	51
Figure 4.8 Selecting VM	51
Figure 4.9 Select destination host	52
Figure 4.10 Migration	52
Figure 4.11 Migration	53
Figure 5.1 Number of VM migrations	55
Figure 5.2 Energy consumption of different approaches	56
Figure 5.3 SLA violation comparison	57

List of Tables

Table 1.1 Evolution of cloud computing	3
Table1.2 Estimated average consumption per server class (W/U) 2000–2006	7
Table2.1 Comparative study of different live virtual machine migration techniques.	35
Table 2.2 Comparative study of virtual machine selection techniques	38
Table 5.1 Testing results	55

Chapter 1

Introduction

Cloud computing has become a current and latest way of providing computational resources as a service. This technology has changed the way people look at the resources. Now resources are seen as services that are available on the internet rather than products that can be purchased. The large scale data centers are addressed as clouds. Cloud computing is achieving popularity in the technical world; yet there are many concerns related with it. This technology is named on the basis of commonly used cloud-shaped symbol which represents an abstraction for complex behaviour and infrastructure behind that simple symbol. It is a pay as you go model through which user can get services, storage space and computational resources. This technology uses the various existing technologies in order to achieve the demands of users over the internet and satisfy their expectations.

Example: Experience with email account is the basic example of cloud computing, without bothering whether it is Yahoo!, Gmail or Hotmail. They take care of the software and infrastructure necessary to handle user personal email account. Whenever user want to access the account, then user just need to have an internet connection using which, web browser should be opened and then user can login. User can access the account from anywhere without bothering about carrying along the personal computer. An email client is similar to how cloud computing works. Except instead of accessing just your email, you can choose what information you have access to within the cloud.

Cloud computing has emerged as a reliable way of benefit to small and medium size businesses who want to outsource the infrastructure. This is done be relieving them from the need of arranging basic software and hardware infrastructure along with the effort and cost of its maintenance. Also the large companies who wish to provide peak resource utilization facility to their customers without investing in building large data centers are making benefits through this new promising computing model.

Large scale datacenters with thousands of compute nodes are established as a result of rapid growth of cloud computing around the world. These datacenters are expected to be dealing with diversity of applications along with flexibly allocating requested resources to the workloads, so that maximum utilization of resources can be done. In order to achieve these requested features, data centers use to consume large amount of electrical energy. This consumption results in high operating cost and high carbon dioxide emission in the environment. Carbon dioxide emission from data centers significantly contribute to the green house effect. It contributes around 2% of the global emission of carbon dioxide [1].

Statistics shows that average energy consumed by each data center is equivalent to energy consumption of 250,000 household appliances. Figure 1.1, shows that increase in energy consumption by data centers is more than double in 5 years (risen by 56% from 2005 to 2010), and energy consumption by data centers in 2010 contributes 1.5% of the total electricity use [2].

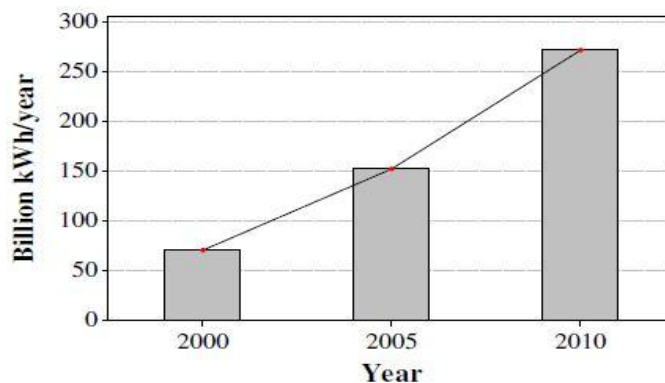


Figure 1.1: The worldwide data center energy consumption 2000-2010 [2]

According to survey of American society of heating, refrigerating and air-conditioning engineers (ASHRAE) in 2014, the infrastructure and energy consumption costs 75% of the total expenditure whereas operating a data center costs only 25% [3]. Power consumption of server is studied in [4] and results shows that it cost 7.2 billion dollars in 2005 for the amount of electricity used by servers all over the world. This also includes electricity consumption for cooling purpose and also of auxiliary equipments.

Facts also indicate that electricity consumption of this year is double of what it was in 2000. Managing resources in a energy efficient way is the biggest challenges that data

centers are facing and it will grow rapidly and continuously unless energy efficient and advance methods of data center management are developed and applied. Detail discussion of these methods is done later in report.

1.1 Evolution of Cloud Computing

Cloud computing is not that old technology as its story is linked with growth of internet and business technology. It provide a solution to the main problem that internet is facing that how internet can help users to improve and speed up the growth of their business. The end of 90s and the start of 2000s were an appropriate and right time for the cloud computing to take off. High speed bandwidth and multi-user architectures were developed in this time providing software's that are globally interoperable. In 1999, Salesforce.com comes into existence which delivers business applications using a normal trend website-this is now known as cloud computing. This time is taken as untouched brightness in technical world, which lead to dot-com boom. During this boom many internet based companies come into this business expecting that initially they can bear some loss by taking it as investment and later they can charge users for their services. Example of such companies is Amazon and Google. These companies failed to make profit in initial years as they are spending more money on marketing and for improving quality of their services. This dot-com bubble goes to its height on 10th march 2000, and then breaks down in the coming week as Dell and Cisco which are main stockholders, sold off many of their stocks. Many other reasons contribute to the burst of this bubble, which include poor online takings from the 1999 Christmas season and Y2K prediction. Only 50% of dot-com survived until 2004, so rather than just complete bursting of bubble, it either grow vigorously or simply run at normal rate. At this time companies decided that they will offer services through internet rather than just using internet as a way to communicate with customers and to place orders.

Table 1.1: Evolution of cloud computing

Company Name	Year	Technology	Concept
Amazon.com	2002	Amazon Web Services	Users can store data and multiple persons work together.

Facebook.com	2004	Facebook.com	Cloud is used as personal service. Change the way users communicate and store their data.
Amazon.com	2006	Elastic Compute Cloud (EC2)	Enable users to run their personal applications on cloud.
Amazon.com	2006	Simple Storage Service (S3)	Introduced pay-as-you-go model
Salesforce.com	2007	Force.com	Introduced platform as a service (paas). Developers build, store and run applications on cloud.
Google.com	2009	Google Apps	Allow people to create and store documents on cloud.
Salesforce.com	2010	Database.com	Cloud based database that can be used and run on any device. This can be written in any programming language.

1.2 Cloud Computing

Using Cloud Computing information can be accessed from anywhere. There is no need to carry the device along in order to access the information. An internet connection is enough to access the information that is available on the cloud space.

According to National Institute of Standards and Technology NIST, Cloud computing can be defined as “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”.

Cloud Computing is basically an on demand model “you pay as you go” this phrase represents the core concept of cloud computing i.e. users don't need to pay for whole

thing, they just pay for the part or amount of resources they use. They can increase or decrease the amount of cloud services required on demand. Cloud Computing is paradigm for large scale distributed computing that make use of existing technology such as virtualization., service orientation and grid computing. So in cloud computing someone is providing the users various types of services & users are paying according to the amount of services used by them.

1.2.1 Five Essential Characteristics of Cloud Computing

- a) **On Demand Self Service:** Users get the services only when they demand for it. Consumers get services without requiring any human interaction with the service providers.
- b) **Broad Network Access:** Services are available on internet. Just an internet connection is required to access the services.
- c) **Resource Pooling:** Different resources of different providers are pooled together to serve the multiple demands of consumers. Physical and virtual resources are assigned and reassigned based on the demand of consumer. Consumer doesn't know the exact location.
- d) **Rapid Elasticity:** Consumers can scale out and scale in the services according to the need.
- e) **Measured Services:** A service consumed must be measured in terms of the resource used. So the consumption based billing is used in this which is known as “pay per use”.

1.2.2 Deployment Models

- a) **Private Cloud:** This cloud provides services solely for an organization. The services are managed by providers and organization. Sometimes a third party apart from organization manages the private cloud in both on-premise and off-premise.
- b) **Public Cloud:** Cloud providers provide services for the general public. Any user having internet connection and authorized access to the cloud space can use the services provided by cloud provider.

- c) Community Cloud: In this, cloud is shared between several organizations having similar demand for cloud. Multiple organizations have access to same cloud infrastructure.
- d) Hybrid Cloud: Hybrid cloud is made up of two or more types of cloud discussed above. In this cloud services are available and consumer can access some of it publically and some are accessible from private or community cloud.

1.2.3 Service Models

- a) Software as a Service: Running applications are provided as service to the customers. The consumers use the provider's applications running on cloud infrastructure. Consumers can access the application through a thin client interface such as web browser but with least control. Consumers cannot manage and control any type of cloud infrastructure such as network server, OS and storage etc. SaaS allows to access the application from anywhere by just installing the software on device having internet connection. Example of SaaS providers are Google Apps, salesforce.com.
- b) Platform as a Service: In PaaS the cloud providers provide the infrastructure, programming language & tools needed by consumer to develop the application. The providers don't provide application rather they provide support in terms of tools & infrastructure to develop and deploy the application of clouds. PaaS is a level above SaaS. Here consumers have some control as compare to SaaS. Consumer cannot manage and control the network server, OS and but can control the deployed application. So PaaS provides subscriber access to the components that are required by consumers for developing & operating applications on the internet. PaaS examples include Microsoft Azure Services Platform, Google App engine and Salesforce.com internet application development platform etc.
- c) Infrastructure as a Service: Here the providers provide the infrastructure to the consumers. Infrastructure includes processing, storage, network and other fundamental computing resources such as operating systems. Consumer does not manage or control cloud infrastructure but has control over the OS, storage and deployed applications. Here the level of control is much more than SaaS and PaaS. Cloud providers have least control in IaaS. Here the

computational infrastructure is available over the internet. IaaS allow consumers to extend their infrastructure on-demand basis. One of the examples for IaaS is Amazon which offers S3 for storage and EC2 for computing power.

1.3 Energy Efficiency

Applications from scientific, computer and business domain are demanding improvements in performance which is taken care during development of computing systems. However, this improvement has started to limit due to the issues of increasing energy consumption and carbon dioxide footprints effecting global warming. Survey in [4] shows the comparison between power consume by three classes of servers.

Table 1.2: Estimated average consumption per server class (W/U) 2000–2006 [4]

ServerClass	2000	2001	2002	2003	2004	2005	2006
Volume	186	193	200	207	213	219	225
Mid-range	424	457	491	524	574	625	675
High-end	5,534	5,832	6,130	6,428	6,973	7,651	8,163

Facts indicate that if energy consumption continues to increase with this speed than one day it will be exceeding the hardware cost of server [5]. Experiments done by Intel labs shows the facts that main contribution to total energy consumption of server is done by CPU [6].

After CPU, next are memory and power supply inefficiency. In these modes, Server CPUs and desktop consumes very less amount of energy that is only 30% of what it can consume at the maximum (peak power). This leads the range of dynamic power more than 70% of the peak power [7]. On the other hand, other server components have very narrow dynamic power range. For DRAM it is less than 50%, for disk drives it is only 25% and for network switches it is only 15%. Other components consume around negligible energy [8].

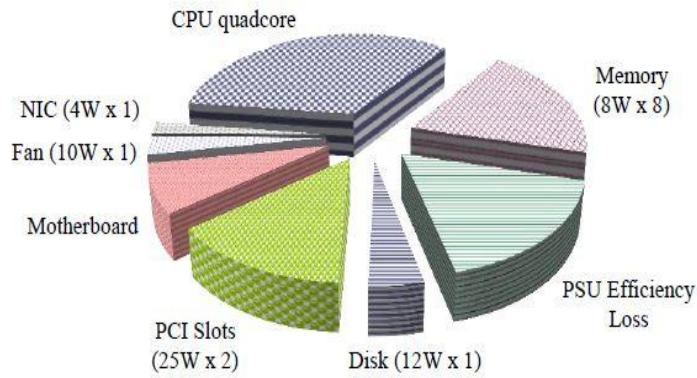


Figure 1.2: Power consumption by server components [6]

The reason behind CPU having so wider power range is low power modes are only supported by CPU whereas the other components can either completely or partially switched off. But there is a performance overhead attached with transition of CPU between the active and inactive modes. According to the facts, a disk drive consumes almost no power when it is in deep-sleep mode, but during the time of transition it bears latency which is around 1000 times higher than the latency during regular access. Due to this overall dynamic power range is narrowed (30%) further resulting in the server consuming more than 70% of its peak power even in idle state.

These loopholes shift the goal of designing the computer system to power and energy efficiency. Power consumption and energy efficiency problems are somewhat interconnected and somewhat need to be considered separately. Energy consumption determines the amount of electricity consumed by the system which further determines the cost of electricity.

1.3.1 Power Model

Power consumption in data centers is mainly calculated from consumption by CPU, disk storage and network interfaces. Energy consumption by CPU is the main biggest contributor to overall power consumption of data centers. So, this power model focuses on efficient usage of energy and managing its consumption in order to save energy.

Recent studies [9, 10, 11, 12] show that application of DVFS on CPU results in almost linear power-to-frequency relationship. This result comes due to the reason of setting limited number of states to frequency and voltage of CPU. DVFS is not

applied on all system components, only CPU is taken into account for its application. This is also the major reason for the arrived result. Normally servers are preferred to be put on idle state when not working but this approach show that an idle server consumes approximately 70% of the power consumed by the server running at full CPU speed. So, this power model justifies the decision of switching off the idle servers. Thus decision reduce overall total power consumption. Therefore, in our work we use power model defined in (1).

$$P(u) = k * P_{\max} + (1-k) * P_{\max} * u \quad (1)$$

Where P_{\max} depicts the maximum power consumed during 100% server utilization; fraction of power consumed by the idle server is represented by k ; and u shows the CPU utilization. The utilization of CPU is not fixed as CPU works according to the workload. Thus, if workload is high than CPU is fully utilized and if workload is low than it may possible that CPU is underutilized. Thus, the CPU utilization is a function of time and represented as $u(t)$. Therefore, total energy (E) consumption by a physical node can be defined as an integral of the power consumption function over a period of time (2).

$$E = \int P(u(t)) \quad (2)$$

1.3.2 Energy Consumption as a Problem in Data Centers

Energy is mainly wasted because computing resources are used inefficiently. According to the previous year's data, even when the servers are rarely at idle mode, the utilization is never 100% [7]. Servers normally use 10-15% of their peak capacity but data center owner has to pay expenses of over provisioning which further results in extra Total Cost of Acquisition (TCA) [7]. Total Cost of Ownership (TCO) is also increased due to Management and maintenance of over provisioned resources. Therefore, underutilized servers play a vital role in inefficient energy consumption. Another problem arising due to high energy consumption and increasing number of server components is the heat dissipation. Electrical power used by components is converted into heat. How much heat will be produced depends on the efficiency of component designs along with voltage and frequency at which the component operates. Overheating results in high error-proneness and decrease in lifetime of

components. Therefore, heat generated by server components need to be dissipated so that server can lie in its safe thermal state. Moreover, electrical energy is needed for cooling system operation. There are efficient cooling systems in today's world but few years back, for 1 watt of power consumed, an additional 0.5-1 W was required for cooling system [8].

High energy consumption raises the need of equipments for cooling purpose and infrastructure for delivering power. This results in boosted electricity bills. This peak energy consumption also limits the performance improvement resulting in SLA violations. Beside the overwhelming cost and electricity bills, another problem arising from this issue is 2% of global carbon dioxide is emitted by data centers. According to the estimation by the U.S. Environmental Protection Agency (EPA), the current efficiency trends led to the increase of annual CO₂ emissions from 42.8 million metric tons (MMTCO₂) in 2007 to 67.9 MMTCO₂ in 2011.

All these reasons arises the need of saving energy and power in all aspects and it becomes a first-order objective while designing modern computing systems. Concept of green and energy efficient computing started back in 1992 with energy star launched by U.S. Environmental Protection Agency. This energy star reduces emission of gases by identifying and promoting products that are energy efficient. Concept of putting the electronic devices at sleep mode is also initiated during those years.

Systems working on less than 50% utilization results in major inefficiency as resources are underutilized. Although resources are underutilized but infrastructure has to be managed according to the situation when peak load needs to be handled. So, need of handling the rare case of 100% utilization, expenses on cooling systems, PDU, generators, power delivery facilities and UPS need to be taken care of.

1.3.3 Efficient Ways of Energy Consumption

In order to eliminate the high energy consumption issue, it is essential that the wastage of electricity while delivering to computing resources should be removed. Also the inefficiencies need to be taken care of. The way of utilizing resources in order to serve application workload should be changed so that energy can be saved. This can be done by improving various things in datacenters which include resource

allocation method, physical infrastructure along with management algorithms. Energy consumption is not only determined by hardware efficiency, but also by the resource management system deployed on the infrastructure and the efficiency of applications running in the system. A basic approach followed for reducing power consumption of server processors is to switch them off or switch the server to sleep mode which is a low-power-performance mode [7].

Sleep mode consumes less energy as compare to idle mode. This approach arise the need of creating datacenters whose agenda is to use fewer physical servers for serving application workload and deactivating the idle servers, which further results in better resource utilization and less power / energy consumption. Various initiatives are taken aiming the reduction of energy consumption and carbon dioxide emissions. These initiatives are taken by Climate Savers Computing Initiative(CSCI), Green Computing Impact Organization, Inc. (GCIO), Green Electronics Council, The Green Grid, FIT4Green, ECO2Clouds, Eco4Cloud, International Professional Practice Partnership (IP3), with the membership of large companies, such as AMD, Dell, HP, IBM, Intel, Microsoft, Sun Microsystems, and VMware.

Various methods are there for accomplishing this aim, one of them is dynamic consolidation of Virtual Machines (VMs) [13, 14, 15, 16] enabled by the virtualization technology. Virtualization allows creating multiple VMs on a single host/physical machine or server. Thus virtualization helps in proper and efficient resource utilization and also in increasing the Return on Investment (ROI). Basic approach of switching idle nodes to low-power modes (i.e., sleep, hibernation) is also used under virtualization for reducing energy consumption. Virtualization also provides another efficient way of minimizing energy consumption named as live migration. Live migration means transferring the running VM from one server to another .This is done is a transparent manner (user have no idea about migration) and with a close to zero downtime. Through live migration [17], VMs are dynamically consolidated resulting in very less fluctuations in the workload and the number of active physical servers are minimum most of the times. Migration is done in two cases: first is when a host is underutilized. VMs are migrated from underutilized host so that this underutilized host can be switched off in order to minimize the number of active hosts. Second case when a VM need to be migrated is during overload of physical server.VM from overloaded host is migrated in order to minimize the load

and to avoid performance degradation Which further leads to SLA violations and violation of the QoS requirements. Idle hosts after migration are automatically switched to a low-power mode so that static power and overall energy consumption by system can be reduced. When required, hosts are reactivated to accommodate new VMs or VMs being migrated.

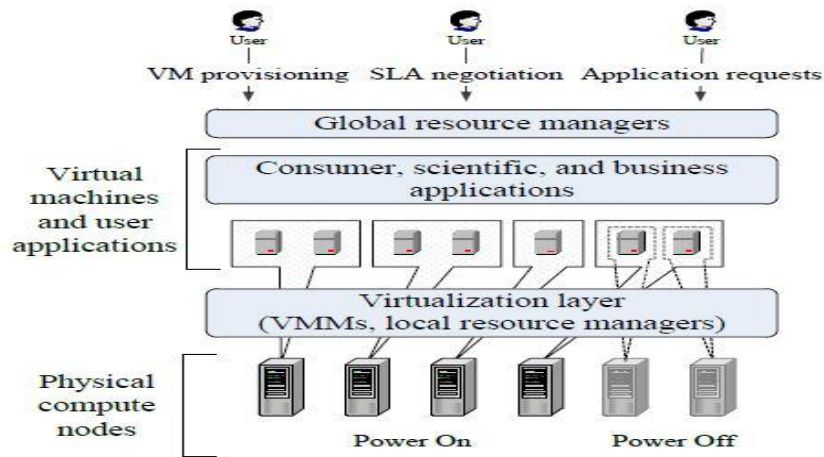


Figure 1.3: High level system view [15]

1.3.4 Green Cloud Architectural Elements

Issues related with energy efficiency and consumption in data centers of clouds leads to Green Cloud computing data centers, which are more capable of satisfying applications' demand of computational resources in an energy efficient way. Thus by using such green data centers, large amount of energy can be saved. High-level architecture, supporting energy-efficient service allocation is shown in Figure 1.4. This architecture uses the green cloud computing infrastructure [18].

Four main entities involved in this are:

a) **Consumers/Brokers:** Cloud consumers are the one who demand the cloud resources for performing their computational tasks. Brokers of cloud consumers perform the task of helping hand for cloud consumers in order to interact with the cloud world. Any of them can submit their request for resources to cloud, from anywhere by just having an internet connection. Cloud consumers and users of services deployed on cloud may not be same every time. For example, company asking for deploying a service/application is the consumer and all the employees of that company who will use or access that application are the users.

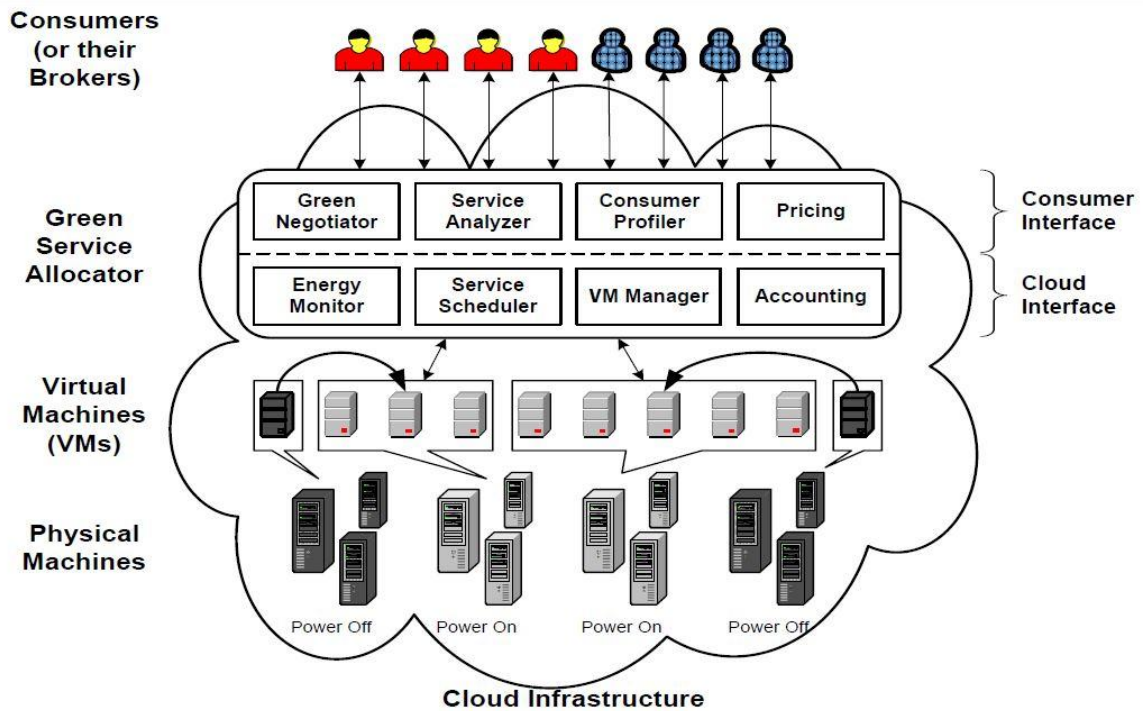


Figure 1.4: High-level system architectural framework [18]

b) Green Resource Allocator: Consumers interact with the cloud providers by accessing the cloud infrastructure provided by them. Green Resource allocator is the interface through which consumer's access cloud infrastructure. In order to perform its responsibilities, it requires the following components to interact with each other. This also play a vital role in energy-efficient resource management:

- i) Green Negotiator:** Prices and penalties for violations of SLA is specified, negotiated and finalized by green negotiator on behalf of cloud providers. Negotiation is done depending upon the QoS and energy saving schemes required by consumer's requirements and energy saving schemas. In case of Web applications, QoS metric can be 95% of requests being served in less than 3 seconds.
- ii) Service Analyzer:** It checks the feasibility of submitted request before accepting or rejecting it. The service requirements of a submitted request is interpreted and analyzed by service analyzer. Hence, for performing its tasks accurately, the information about latest load and energy is needed from VM Manager and Energy Monitor respectively.
- iii) Consumer Profiler:** It Gathers and manages the details of each and every consumer. Specific characteristics of consumers are taken into account so that

special privileges and prioritization can be granted to these consumers over other ones.

- iv) Pricing:** It decides how much should be charged for services, so that supply and demand of computational resources can be managed and service allocation should be done effectively.
- v) Energy Monitor:** It keeps track of physical machines and decides which physical machines need to be power on/off.
- vi) Service Scheduler:** It deals with VMs. Requests are assigned to VMs by service scheduler. It also decides when there is a need of new VMs or when the existing ones need to be removed in order to meet demands of consumer. It also determines laws and contracts for allocating resources to VMs.
- vii) VM Manager:** It determines whether VMs are available or not along with their resource entitlements. Decision of migrating VMs across physical machines is also taken by VM manager.
- viii) Accounting:** This keeps track of usage of resources by consumer requests, so that latterly they can be charged according to the amount of resource usage. Efficiency of service allocation decisions can be increased by using historical usage information.

c) Virtual Machines: A single physical machine can have multiple VMs running on it. These VMs can be dynamically started and stopped. These VMs are facilitated with maximum flexibility, so that they can access different partitions of resources of same physical machine on which they are running. Different applications requiring different operating system environments can be run by multiple VMs of same physical machines. These VMs can be migrated dynamically from one physical machine to another so that underutilized resources can be put to sleep mode or turned off (using DVFS).

d) Physical Machines: This is the actual underlying hardware (host/servers) which are providing infrastructure to clouds. Virtualized resources are created on these physical machines in order to meet service demands.

1.4 Virtualization

Virtualization is a technology which is being widely used in today's world and provides excellent operational and financial results. It is a concept in which access to a single underlying piece of hardware is coordinated so that multiple guest operating systems can share that single piece of hardware. No guest operating system is aware of the fact that they are sharing the physical resource.

Guest Operating System: This is the operating system which is being hosted by the specified virtualization software layer (known as hypervisor). Multiple guest operating systems can be installed on a single hypervisor without knowing that they are not directly running on physical hardware.

Hypervisor: This software is the intermediate between the guest operating system and the host operating system. Multiple applications run on guest operating systems which are running on a piece of software known as hypervisor, which in turn deals with underlying physical hardware or main operating system known as host operating system.

Type1 hypervisor or native virtual machine → process level virtualization techniques.

Type2 hypervisor or hosted virtual machine → system level virtualization techniques.

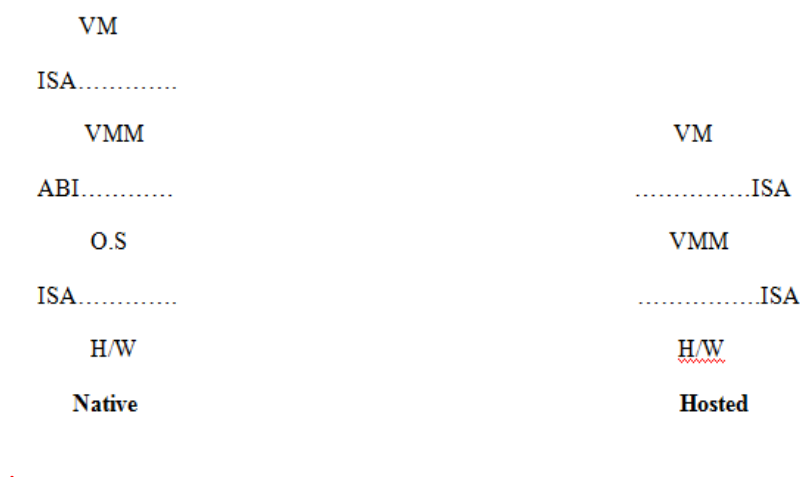


Figure 1.5 : Types of hypervisor

Host operating System: This is the main operating system installed on physical hardware. Hypervisor is installed on this host operating system.

1.4.1 Need of Virtualization

- a) **Hardware is Underutilized:** The first trend that is making virtualization mandatory is exponential growth of computing power of machines or processors as explained by Moore's law. Most of the machines in data centers are running and consuming only 10 to 15% of the total processing capacity, so most of the machines are underutilized or in other words their processing power is mostly unused. Virtualization is one of the ways to match computing power with the load on the machines. Virtualization enables a single machine to be accessed by multiple systems and thus raise the utilization to 80 to 90%.
- b) **Datacenters Run Out of Space:** Data created and stored by users is increasing exponentially. This leads to the concept of Big Data. Main effect of Big Data is that large number of servers is needed to store and manage this big data. This big number is causing the space problem for data centers. So the data centers are calling for new methods of data storage which leads to Virtualization (Storage virtualization in specific).
- c) **Energy Cost Go Through the Roof:** Cost of electricity is the main parameter of project's operating cost. Running large number of underutilized machines or servers in data centers leads to the need of large amount of electricity to run those machines, which in turn leads to high cost. Virtualization has the ability to reduce the number of servers significantly by running multiple guest systems on one physical server, which in turn reduces the overall cost of energy for organizations.
- d) **Rise in System Administration Cost:** System administrators are required to ensure that servers run properly. Most of the tasks of system administrator is related to accessing server physical machines, software's running on them and monitoring its performance. The Steady increase in amount of data leads to increase in number of servers which further raise the demand of very good system administrators to manage that large number of data and servers. As the work of system administrator is critical and crucial, even mainly related with server machines, so they don't come cheap. Virtualization offers the chance to reduce the cost of system administrators by decreasing the number of servers in data centers. Virtualization only reduce the system administrator work not delete it completely.

e) Greening Initiatives: Now days, companies are trying to reduce amount of energy consumed and to reduce the carbon footprint. Data centers are the place where most of the power and energy is consumed. Energy is consumed for running servers as well as for keeping them cool. Reducing the no. of servers through server consolidation will reduce the need of energy. Virtualization provides an efficient way of server consolidation.

1.4.2 Types of Virtualization

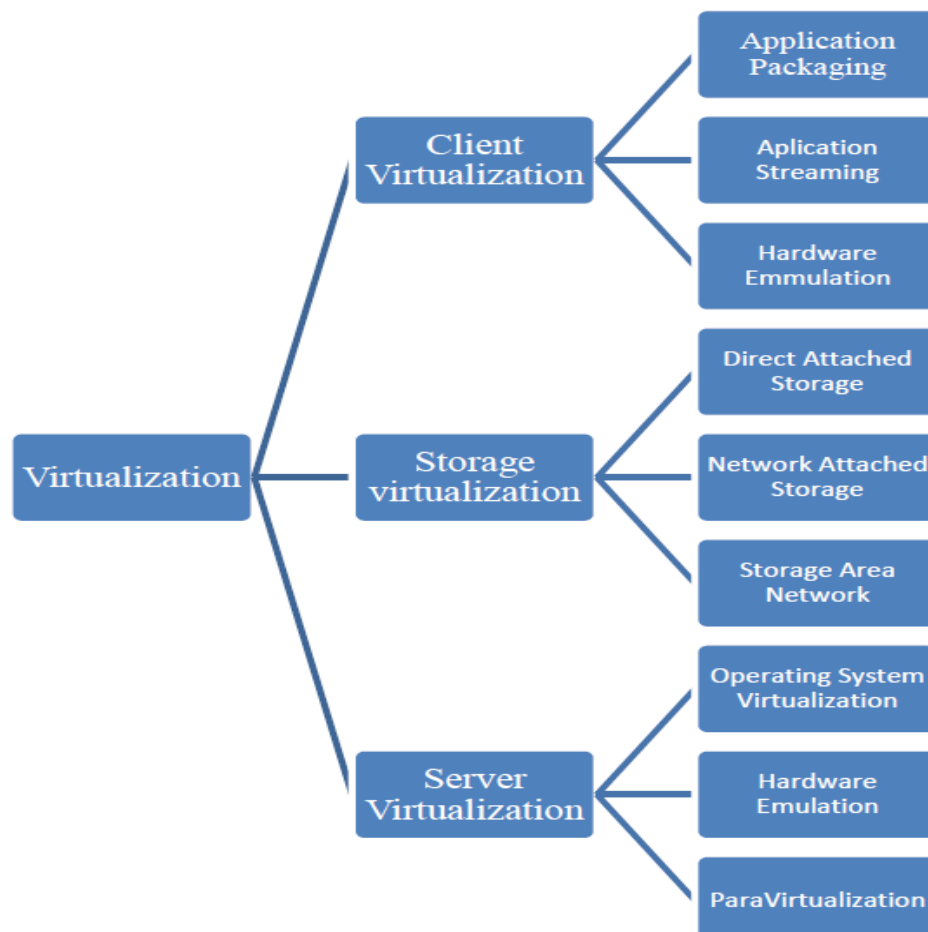


Figure 1.6: Types of Virtualization

Server Virtualization

IT Organizations are mainly working to get more control over there sprawled data centers. Server Virtualization is critical and important as data centers are running out of space. Due to lack of space data centers are unable to deliver necessary resources which is affecting the business badly. In this the concept of virtualization is applied on servers.

Server Virtualization can be done in three main ways:

- a) **Operating System Virtualization:** Operating system virtualization runs on the top of host operating system. No virtualization software is installed on host operating system to present hardware emulation. Each guest operating system is provided with set of libraries which are used by applications running on it. To be more clear, in this type multiple copies of a single guest operating system are running with in an operating system virtualization. Each copy is known as container. Applications running on any one of them will have the illusion that the machine on which they are running is only dedicated to them. One guest operating system container cannot see and interact with other container and its applications. This method is useful in case when similar operating system functionalities need to be offer to different users.
- b) **Hardware Emulation:** Virtualization software (Hypervisor) is used in this type of server virtualization. Hypervisor is installed on the guest operating system. This hypervisor create an emulated hardware environment which is referred to as Virtual Machine Manager (VMM). So, the guest operating system is installed on a virtual machine (hypervisor) that create a virtual environment of hardware and operating system (based on which type of hypervisor used) with which the guest operating system interacts. The VMM environment created by hypervisor coordinates the communication between the guest operating system (also known as virtual machine) and the actual machine hardware and operating system. Main point in this type is that the guest operating system and the virtual machine are stored in files in such a way that they together form the system image. So, this image can be migrated from hardware to another to achieve the required goals. This is known as virtual machine migration.
- c) **Para Virtualization:** The entire virtual machine is not created to host the guest operating system. The hypervisor exposes a modified version of physical hardware interface to guest O.S. It is like a traffic controller which coordinates the sharing of actual physical hardware resources among multiple guests O.S. Only one guest operating system can access the physical resource at a time.

1.5 Virtual Machine Migration

As discussed above virtualization is a technique that enables multiple operating systems to run on a single physical hardware. It has become the core aspect in cloud computing due to its various advantages such as efficiency in cost and resource sharing, flexibility, portability etc. Virtual machine (VM) migration is the key feature of virtualization. Virtual machine migration (VM migration) means copying a virtual machine (VM) from one physical machine to another in a transparent manner. In this process the storage and memory of VM, network connections of VM etc are moved from the source physical machine to the destination machine.

1.5.1 Need of Virtual Machine Migration

VM Migration is basically needed to provide benefits in virtualized environment. This technology has attracted considerable interest for datacenter management and server consolidation. VM Migration is basically done to fulfill the following needs

- a) **Load Balancing:** In this VM is migrated in order to give high throughput to users. If there is a server which is overloaded and unable to handle that load with good performance than VM from that server is migrated to underutilized server in order to balance the load. This leads to high performance and throughput. [19], [20].
- b) **Maintenance and Servicing:** VMs are migrated from one host to another by administrators or service providers in order to do maintenance and servicing of source host. In this way server is serviced with degrading the performance. Also continuity in the user access is maintained as user never came to know that the work is shifted to another host.
- c) **Improve Resource Utilization:** VM migration is done in order to achieve maximum resource utilization. Load of underutilized servers are assigned to other servers that can handle that load by migrating the virtual machines of underutilized servers. In this way the later servers can be properly utilized and the underutilized servers can be kept idle to save energy.
- d) **Power Management:** The aim of this is to power on minimum no. of servers. VMs of underutilized servers are migrated to other servers which can handle that load and then those underutilized servers are switched off or put in idle state to save the power and energy.

1.5.2 Types of Virtual Machine Migration

VM migration is done on the basis of threshold which is decided by the data center owner or decided by client in SLA. Usually, a highest resource usage value is set as upper threshold and lowest resource usage value is set as lower threshold for servers/physical machines. Migration is done in two cases, first is over-utilization and second is under-utilization. In former VM is migrated from that PM whose resource usage crosses the upper threshold to that PM whose resource usage is below upper threshold and which can bear this load. In later, the VM is migrated from PM (physical machine) whose resource usage is below lower threshold to PM whose resource usage is above lower threshold and which can handle the new coming VM and can provide resources to it. In this the source PM is sent to idle or sleep mode after migration to save energy consumption. These conditions are taken care in load balancing and server consolidation algorithms.

VM can be migrated in two ways: Static and Dynamic.

Off-Line / Static Migration: Off-line VM migration process pause the VM and transfer all the states of VM to target host then finally resume the VM in the new host. The advantage is simple procedure and disadvantage is long downtime.

Live / Dynamic Migration: Live migration of virtual machine means moving the running VM from one host to another without halting it initially and also without letting the user know about it. In live VM migration state of VM is transferred with minimum service disruption from one host to other host in a transparent manner. The Key advantage of live VM migration is user-invisible downtime with fast network.

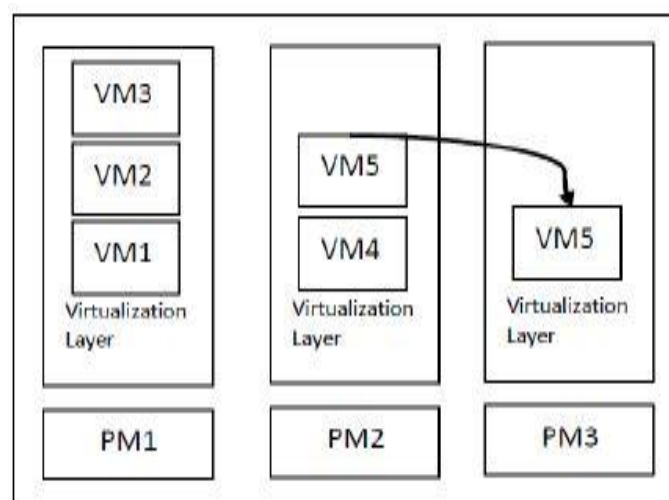


Figure 1.7: Virtualized server with live migration capability [21]

In static, the VM is suspended or power off and then it is migrated. In the later, VM is migrated while it is running on source physical machine (or host).The later is also known as live VM Migration. In order to understand the dynamic migration, the static one should be clearly understood. The static migration clears the basic concept of how VM migration is done, but its least used in cloud computing because when the VM is powered off for migration, then applications running on that VM goes down and the user came to know about this. Benefit will be there if user never get an idea that the VM on which his applications are running is being migrated or shut down even for a second. Basically this is what the main objective of virtualization in cloud computing, that user never came to know what actually is happening in the backend. User is only concerned about the performance of applications and the availability of resources for which user pay.

1.5.3 Side Effects of Virtual Machine Migration

Migration is mainly used in cloud computing for achieving various goals as discussed above. But migration has some side effects which are performance degradation of application which are running on virtual machines which are migrated as well as energy overhead[13,21]. It also causes communication cost and network load is increased during the migration. These side effects are acting as a hurdle in the use of VM migration.

1.5.4 Why Virtual Machine Allocation has been selected

As cloud computing is more beneficial if it is executed in a green environment and for this energy efficiency is the main concern, So this lead the interest to minimize the energy overhead due to VM migration in cloud computing. Energy efficiency can be achieved in terms of power consumption or time. Energy Efficiency can be increase by either reducing the power consumption for VM migration or by decreasing the time take for migration. In this thesis, VM allocation part from whole VM migration process is taken into account for reducing energy consumption of whole VM migration process. VM allocation is selected because choosing the most suitable destination host for VM migration is an important and crucial decision while considering about saving energy because if this decision is once taken wrong than rather than saving energy, more time and energy will be consumed for correcting the previous wrong decision and selecting the most appropriate host.

Chapter 2

Literature Survey

As movement of VM from one physical machine to another is possible in virtualization, it initiates dynamic or live VM migration for performance requirements. When PMs is under-utilized or over-utilized then VMs on them can be moved to another PM in order to minimize the number of active PMs, while free PMs can be switched to sleep or idle mode (sleep mode consume less energy) which further helps in decreasing total energy consumption by data centers. Complete process of VM migration is divided into three steps. In each step, various algorithms are there to perform that step. In this paper the background study is classified and carried according to two steps.

2.1 Virtual Machine Selection

VM allocation is done in two steps. First one is VM placement, whereas second step represents optimization of current allocation of VMs or we can say VM selection. Modified Best Fit Decreasing (MBFD) algorithm can be the solution to first part. In this all VMs are arranged and listed in decreasing order on the basis of their CPU utilization and then place VM on that host whose power consumption increase is less as compare to other hosts due to this placement [18].

Basic condition for VM migration is there must be need of VM migration. So, for VM migration two steps are needed. First is to decide whether there is a need of VM migration or not. If yes, then find out which VM should be migrated and to which destination PM that VM should be migrated. This is also known as optimization of current allocation of VMs. For deciding this, threshold is the basic solution. Initially one threshold is defined. This threshold is the upper utilization limit of a PM and new VMs can be placed on this PM unless and until no SLA violation occurs due to increase in CPU utilization by VMs. Reallocation of VMs is done using MBFD algorithm such that the upper utilization threshold of PM should not be violated. A more sophisticated approach is to use two fixed threshold values known as upper and lower threshold. Main concept behind this approach is to keep the utilization of PM's

resources by all VMs allocated on it, between the lower and upper threshold. If VM resource utilization of a PM violated the lower threshold (less than lower threshold) than all those VMs need to be migrated and that PM/ host is set at sleep mode in order to prevent power consumption in idle mode. On the other hand, if utilization goes beyond the upper threshold, than some of the VMs from that host have to be migrated to another host in order to decrease the utilization and prevent SLA violation. The difference between the old and new placements gives some VMs that have to be reallocated.

One more concept is considered while deciding upper and lower thresholds. A higher threshold that is approximately equal to the maximum possible CPU utilization is set as upper threshold for hosts/PMs with VMs whose behavior or we can say resource demand does not change often. In other case where hosts have VMs whose behavior changes constantly that is there resource demand scale out and scale in very frequently, has upper threshold less than the maximum possible CPU utilization to handle the sudden scale up in utilization.

Buyya et al. [22] proposed three policies for selecting VMs in VM migration. First is Minimization of migrations (MM). In this policy minimum number of VMs is selected for migration in order to get the CPU utilization less than the upper threshold, so that no violation of upper threshold exists. This policy minimizes the migration overhead as least number of VMs is migrated. In this policy firstly VMs list is sorted in descending order of their CPU utilization. From this list that VM is selected for migration that follows two conditions. First condition is that the VM's utilization should be more than the gap between the higher threshold of host and the current host utilization. So this condition means that host is not able to provide the resource demand of that VM. Second condition is that after the migration of VM to new host, the difference between the upper threshold and the current utilization of that host should be increased minimum as compare to other VMs. This means that after migration, the increase in utilization of host should be minimum. If no such VM is there that satisfies these conditions than VM with maximum CPU utilization is selected for migration. No further migrations will be done when new utilization of host/ PM is no more exceeding or crossing the upper threshold.

The second algorithm proposed is the highest potential growth policy (HPG). In order to handle the violation of upper threshold, this policy selects those VMs for migration whose CPU utilization is minimum as compare to the requested CPU capacity by other VMs in order to minimize host utilization and also to prevent SLA violation. Third policy is random Choice (RM). In this approach VMs are selected randomly for migration by using uniformly distributed random variable to decrease the resource utilization of host and to keep it below the upper threshold, so that it is no more violating the limits.

Buyya et al. [23] proposed a new technique for VM selection: The Maximum Correlation Policy (MC). According to this probability of overloading of a PM is directly proportional to the correlation among the VM's utilization or applications of that PM. So, according to this policy that VM is selected for migration which has the maximum correlation of CPU utilization with the sum of other VM s on that host/ PM. To find out the maximum correlation, the maximum correlation coefficient is applied. This step is repeated again and again until the host /PM is not overloaded anymore.

Cao et al. [24] described an extension of MC policy of Rajkumar Buyya and named it as MCE. In this improved policy basics MC policy is used .Initially mean and standard deviation of CPU utilization of host is determined and used to find out whether a host is overloaded or not. Secondly on the basis of knowledge of statistics, range of correlation coefficient is divided into negative correlation and positive correlation. In negative correlation two variables increase or decrease at the same time, means if one variable is decreasing than another will be increasing. On the other hand, in positive correlation both variables increase or decrease simultaneously. So, on the basis of this concept they proposed that the positive correlation is more likely to lead to overloading than negative one.

2.2 Virtual Machine Migration

Wood et al. [19] proposed a hotspot detection algorithm that detect when the VM should be migrated. Greedy algorithm used by hotspot migration determines the destination host for migration as well as evaluate the quantity of resources that need to be allotted to VM after migration. Many literature surveys are there that focus on

various migration methods with performance. The first technique of migration of VMs was stop-and-copy approach [25], which involved pausing the targeted/source VM, copying all pages of VM to the destination, and then start execution of that VM on destination PM. The advantage of stop-and-copy approach is simplicity but the service downtime is more in this approach. In the stop-and-copy approach, more the memory of VM need to be migrated more will be the service downtime. These types of approaches are not acceptable in live running scenarios. Some techniques for VM migration are:

2.2.1 Pre-Copy Approach

In a pre-copy approach all states of hardware and software running within a VM are encapsulated. While migrating a VM, three kinds of states are required to be migrated, that are virtual device states, external connections and physical memory of VM. This approach begins with transferring the memory to the destination host/machine. After this, other states such as processor states etc. are transferred to destination. The basic pre-copy approach is suggested by Nelson et al. [26]. In this approach virtual machine is migrated transparently without modifying the operating system or any application running on that operating system. They take the initiative to provide the measurement of performance in case of migrating various virtual machines which are working concurrently. They use the standard industry benchmarks in order to prove the accuracy of their results. The resources needed for VM migration and the overhead due to VM migration is also discussed by them.

Clark et al. [17] propose the Pre-copy approach for live VM migration. This algorithm works in three phases which are: Pre-copy phase, Pre-copy termination phase and Stop and Copy phase. VM continue to run on source, while its memory pages are transferred from source to destination iteratively in first phase .During this transfer as VM is running, so some memory pages get updated which are addressed with name of dirty pages in pre-copy approach. In the next iteration these dirty pages are transferred. Then in next phase VM is suspended. Then at last remaining dirty pages along with state of CPU registers is transferred to destination host. This approach balanced the down time and migration time to some extent.

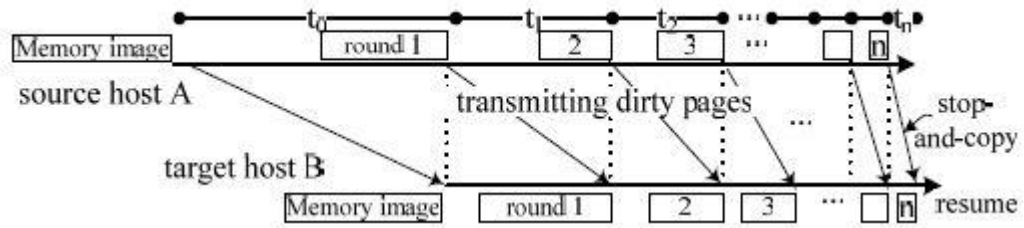


Figure 2.1: Pre-copy approach [17]

Clark et al. [28] proposed the design for live migration of services running on operating system along with some focus on data center and clustering. They use the Xen VMM to propose the designing along with implementation of operating system live migration after which evaluation of its performance is also done. Concept of writable working set is also explained and analyzed by them.

2.2.2 Improvement over Pre-Copy Approach

In order to minimize the downtime of migration of virtual machine several authors have proposed improvement over the basic pre-copy approach. Live migration of VM's downtime directly depends on the size of memory state of the VM that is going to migrate from one host to another host. To minimize the size of memory state of the VM different memory compression techniques are proposed.

Hai et al. [29] describe compression techniques and Characteristics based Compression algorithm (CBC/ MECOM) for fast, stable live migration of virtual machine data. On source side, data which is to be migrated is compressed first and then migration is done. On the destination side decompression is done after arrival of that data. This technique decrease the VM migration time but on the other hand system overhead is increased. The performance and cost of VM migration is also balanced by an adaptive zero aware compression method.

Another memory compression technique is proposed by Svard et al. [30]. Before transferring the memory pages, the delta compression technique is implemented so that downtime due to migration can be decreased. Use of delta compression also increases the migration throughput as compare to simple method of migration. This approach is considered as a modification to the KVM hypervisor. Running VMs with different workloads are migrated for evaluating the performance and the results shows that migration downtime is decreased significantly by using this approach. The

delta compression algorithm along with simple migration demonstrate successful results while migrating a large system that fails initially when migration was done using only standard algorithm. Also risk of service is low as data is saved as a change in versions. Effects of this technique and situation best suited for its use are also discussed by them.

There are some other techniques which are useful in minimizing the downtime of live migration of VMs. While transfer of memory state of the VM, pre copy algorithm marked the frequently updated pages and transfer these dirty pages at last iteration of memory transfer. Ma et al. [31] proposed an improved Pre-copy approach. Bit map page is added to Pre-copy approach, which records or mark the frequently updated pages. Those pages are then added into the page bitmap. So, the updated pages are transferred only once at the end of iterations. This approach minimizes the quantity of data for transferring which further minimize the total migration time. Number of iterations is also reduced by using bitmap page.

Lie et al. [32] proposed a new approach for virtual machine migration which is known as an improved time-series based pre-copy approach. In this technique, concept of prediction is used to find out those dirty pages that are updated very frequently in the past and a precise prediction is done that those pages are going to be updated frequently in the future also. These highly updated dirty pages are transmitted in the last and final round of iteration for migration. This technique significantly reduces the overhead due to the repeated transmission of dirty pages.

Huang et al. [33] proposed a VM migration scheme that reduce the replication of memory pages and improve the performance. RDMA (Remote Direct Memory Access) provided by InfiniBand is used to decrease the migration overhead. Ibrahim et al. [34] presented a novel algorithm which detects the memory update patterns and terminates the migration when there is no possibility of decrease in downtime. KVM is used as an environment for this experiment setup. Downtime is decreased using this approach and also impact on performance of applications is also very low. This approach is beneficial for both Ethernet and RDMA (InfiniBand) migration.

Thein et al. [35] presented a modification of pre-copy approach in which a pre-processing phase is added into pre-copy method. In pre-processing phase the working set prediction algorithm is implemented in which combination of splay tree algorithm

and LRU technique is done. This is implemented to minimize the data to be transferred with further minimize the total migration downtime.

2.2.3 Post-Copy Approach

Hines et al. [36] proposed Post-copy approach for live VM migration. In this approach processor state of VM is first transferred to destination host, start the VM on the destination host and at last the memory pages are transferred. Memory pages which are not successfully transferred are known as demand-pages, which are transferred at last from source and then VM at source is suspended. The main benefit of this approach is that no duplicate transmission of memory pages is done, thus avoiding the overhead for the same as in pre-copy approach. Downtime of Post-copy is higher as compare to Pre-copy approach.

Gopalan et al. [37] presented the designing as well as implementation of a new technique for live VM migration that uses the concept of post-copy across a Gigabit LAN. This technique is than evaluated to get the desired results. Post copy approach is used along with adaptive pre paging technique to reduce the total migration time. This approach results in decreasing the number of page faults. Dynamic self-ballooning (DSB) mechanism is applied to detect and stop the migration of free memory pages. DSB periodically finds the free memory pages in a VM; also the migration is speed up along with negligible decrease in performance of VM.

2.2.4 CR/TR Approach

Many previous approaches focused on transferring memory but there are some techniques which used different approaches to provide fast transparent live migration of VMs. Liu et al. [38] propose and implement a new technique Check pointing/Recovery and Trace/Replay also known as CR/TR Motion for Live VM migration. This technique is for fast and transparent VM migration. Rather than migrating the memory pages from source to destination host as done in Pre-Copy approach, CR/TR technique transfer the log files which results in less amount of data for transferring along with the methods to synchronize the two VM's running states. CR/TR approach minimize the migration time as well as consumption of network bandwidth as compare to Pre-C.opy approach. CR/TR approach is applicable

when the rate at which log replay at destination is higher than the growth rate of logs on source side.

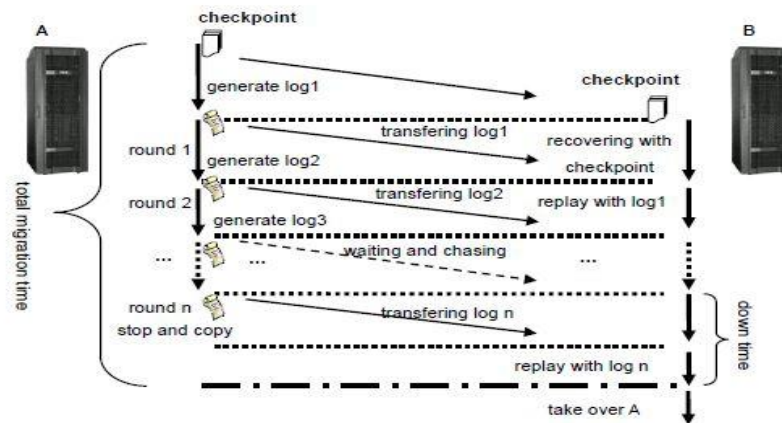


Figure 2.2: CR/TR approach [38]

Liu et al. [39] extend the CR/TR technique for VM migration in both LAN and WAN. Same as basic CR/TR approach, trace logs are migrated instead of memory pages. Synchronization algorithm between source and destination host acts as an extension to basic CR/TR approach. By using this algorithm, a consistent state is achieved between source and destination for migration. Results of this extension lead us to decreased downtime and network bandwidth consumption. The results show that the downtime for migration is almost same in both cases with a little difference of 200 milliseconds.

2.3 Virtual Machine Selection

Clark et al. [17] in his paper described the concept of *writable working set* and also stated role of virtual migration as useful tool for administrator and facilitates fault tolerance, load balancing and system maintenance.

Panchal et al. [27] described virtual machine allocation as an important feature in Cloud environment and provides information of allocated virtual machine in the datacenter. According to him allocation policies are implemented at infrastructure layer and virtual machine allocation can be made transparent to the user. Allocation policies makes cloud infrastructure highly scalable and structured. They used K-Means clustering method for grouping similar type of data set to minimize variance between these sets. Their algorithm inputs were the set of virtual machines and list of data center. Generated output was groups of similar virtual machines.

Various algorithms have been proposed in the literature for the optimal allocation of virtual machine on the host such as Randomized Rounding, Loris et al. [50]. This algorithm allocates subset of virtual machines from the pool of VM list to subset host in the datacenter with probability e_{jh} . If the virtual machine does not fit on the selected host because of hardware constraint then coefficients of probabilities were readjusted and another attempt was made.

Another popular algorithm is bin packing algorithm, Morihara et al. [51], which schedules multiple virtual machines to different hosts. It is an approximation algorithm to search for best fit.

2.4 Energy Efficiency

Energy Management for Hypervisor-based VMs

Stoess et al. [40] proposed a framework for energy management on virtualized servers. Generally energy aware OSes have the full knowledge and control over the underlying hardware and based on this, device or application based accounting is applied in order to save energy. However on other the other hand in virtualized environment, the underlying hardware is shared by multiple VMs. So, all the VMs have the device control and accounting information. This distributed information makes it infeasible for OS to control the hardware. So, in order to solve this problem author proposed new mechanism for fine grained OSes for applying energy level and accounting policies. In this approach author use the notion of energy as the base abstraction in the proposed system.

Proposed system contains the two sub-systems: a host-level resource manager and an energy-aware OS. The first subsystem handles the power limits across various VM instances. In order to meet Power constraints; the manager defines the device type that cannot be exceeded along with power limits for VMs. The second subsystem handles the application specific energy management. This handles the accounting and control for both physical and virtual devices.

The results showed that the proposed approach is capable of enforcing power limits for energy-aware and energy-unaware guest OSes.

Energy-Efficient Server Clusters

Elnozahy et al. [41] deals in homogeneous cluster. They discussed the problem of resource management in a power efficient way. They take single web application and handle the SLA in rms of response time. Main purpose of their approach is to reduce operational cost and overheating of server. Two main approaches are applied by them that are: switching server on and off and DVFS. The system proposed by author is based on mixture of five resource management policies: Independent Voltage Scaling (IVS), Coordinated Voltage Scaling (CVS), VOVO, combined policy (VOVO-IVS), and coordinated combined policy (VOVO-CVS). Authors work on the concept of threshold. Threshold decides when to power on or off the idle node. The total CPU frequency is estimated by authors who help in finding the expected response time, determine the optimal number of physical nodes, and proportionally set their frequency. Proposed IVS policy save 29% energy, VOVO policy save up to 42%, whereas CVS policy in combination with VOVO saves 18% higher energy as compare to VOVO independently.

Managing Server Energy and Operational Costs in Hosting Centers

Approach proposed by Chen et al. [42] deals with multiple server applications for minimizing energy consumption while keeping track of SLA violations. This approach works in two phases. First I allocating number of servers to applications and second is setting the DVFS parameter on server according to the application workload. After the completion of first phase that is allocation of servers. The remaining servers are switched to idle mode or switch off in order to conserve energy. The highlighted point of this approach is consideration of SLA requirements as their important objective. The main objective is to minimize the total cost which includes electricity bills and the cost of switching the server on and off. This affects the reliability of system resulting in decrease in Mean Time Between Failures (MTBF). The authors addressed the defined problem using a hybrid approach consisting of a queuing theory-based approach and control theoretic approach. First works on the concept of prediction. It predicts the workload for near future and thus adjust the allocation of server accordingly. The second is invoked at shorter time intervals. It is used to adjust the DVFS settings of the servers. The experiment shows that this approach saves energy along with meeting SLA requirements.

VirtualPower: Coordinated Power Management

Nathuji and Schwan [43, 44] takes the initiative of exploring techniques for power management for virtualized systems. Along with hardware scaling and VMs consolidation, authors proposed a technique named as “soft resource scaling”. The idea is to use combination of both hard and soft scaling, as soft scaling is useful when hard scaling provides very less power benefits.

This approach acts as a support for such operations of guest VMs which are isolated and independent. Along with this, this approach also deals with control and coordination of diverse power management policies which are applied by the VMs to resources. Proposed system interfere in ACPI calls of guest VMs in order to apply changes in power states, after this these changes are mapped on “soft” states. These soft states are further used as hints for performing real changes in hardware power state. So, system level or application level power management policies for guest VMs are supported by this system. Along with this isolation between various VMs sharing same physical machine is also maintained by this system.

The main idea behind this is to divide the management of resources into local and global policies. Power management policies of guest VMs at each host are coordinated and leveraged at local level. On the other hand multiple physical machines are managed by global policies using the knowledge of rack- or blade-level hardware characteristics and requirements.

This system reduces power consumption up to 34% without degrading the performance.

Power and Performance Management via Look ahead Control

Management of resources in a virtualized data center in a power and performance efficient way is explored by Kusic et al. [45]. Main problem which is solved in this approach is dynamic provisioning of VMs according to the number of incoming requests (workload) in multi-tiered web applications. Request processing rate is a main measure of SLAs. In case of SLA violation, penalty is to be paid by resource provider. So, power consumption and SLA violation is minimized in this proposed system. The author used the Limited Look ahead Control (LLC) to solve the problem of sequential optimization. Variables on the basis of which decisions will be taken

was calculated as the number of VMs to be provisioned for each service; the CPU share allocated to each VM; the number of servers to switch on or off; and the fraction of the incoming workload to distribute across the servers hosting.

Workload changes very quickly, so high performance of the optimization controller is required. Cost and time delays due to switching the VMs and host on/off are also considered by the authors. A power saving mechanism is used which is a combination of Dynamic VM consolidation via offline migration and switching hosts on/off. Kalman filter is used to predict the future system states on the basis of upcoming application requests and thus perform required reallocation according to that prediction. Results show that 26% is saved in power consumption cost in 24 hours by using LLC. Also only 1.6% of requests faces the SLA violation.

Shares- and Utilities-based Power Consolidation

Cardosa et al. [46] deals with the problem of allocating virtual machines in a power efficient way in a virtualized environment. A mathematical formulation of the optimization problem is proposed by the author. Author calculated the power consumption and utility gained from the execution of a VM and named the combined results as “priori”. Basic strategy proposed by author is to place all the VMs according to their maximum request of resources in a first fit manner and also spare 10% capacity to handle future request growth.

The proposed system also sort the physical machines in increasing order of their power cost per unit of capacity in order to leverage the heterogeneity of the infrastructure. Main limitation of this approach is that the relative priorities of different VMs is not leveraged and use only 90% of server’s capacity. This approach was considered as a bench mark and various other authors proposed improvements in the basic policy. This basic policy allows the VMs to shrink to their minimum resource requirement when needed and expand it back when there is enough spare capacity for expansion. Thus, by using this technique additional profit can be made. Limitation of this approach is that VM migration is not applied at run time that means allocation is static. Another limitation is that only CPU is taken into account in this approach.

pMapper: Power and Migration Cost Aware Application Placement

Dynamic placement of applications in virtualized environment is addressed by Verma et al. [47]. Authors also address the minimization of power consumption and meeting the SLAs. The pMapper application placement framework is proposed as a solution to the above mentioned problem. pMapper consists of three managers: performance manager; power manager; migration manager. Actions of these three managers are coordinated by an arbitrator and the allocation decisions are also taken by an arbitrator. Performance manager resizes the VMs according to the current resource requests. It also monitors the behavior of applications. Adjusting hardware power states and applying DVFS is the responsibility of power manager. To consolidate workload, live VM migration need to be done. So, instructions for VM live migration are issued by migration manager. Arbitrator takes decisions regarding new placements of VMs. It also finds the necessary VM reallocation to perform new placement. This approach saves about 25% of power as compare to static and load balanced placement algorithms.

Resource Pool Management: Reactive Versus Proactive

Gmach et al. [48] considered dynamic VM consolidation in energy efficient way as their main target to be achieved. As a solution of this problem authors proposed a new technique which is a combination of a trace-based workload placement controller and a reactive migration controller. First controller collects data about usage of resources by VMs in data centers. This historical information is further used to do more efficient allocation along with keeping track of QoS requirements. This controller place the VM in such a way that minimum number of servers will be needed to handle the workload along with limiting the number of migration need to be done for new placement. System administrator is responsible for applying limit on number of migrations according to the capacity of bearing migration overhead. Resource utilization of physical nodes is continuously monitored by reactive migration controller. It also detects when the servers are overloaded or underloaded. In contrast to first controller, this second controller works on real time data and perform the allocation after small interval of time(one minute).Reactive migration controller's main objective is to deal with fluctuation in workload and take decisions accordingly. Underload and overload condition is decided on the basis of two utilization

thresholds. Overload condition occurs when utilization of CPU or memory exceeds threshold and underload condition appears when the utilization of CPU or memory goes below the specified threshold. Through this policy CPU violation is decreased to minimum but it requires 10-20% higher CPU capacity.

vManage: Loosely Coupled Platform and Virtualization Management in Data Centers

Kumar et al. [49] deals with the concept of dynamic VM consolidation and proposed an approach that provides more stability in dynamic VM consolidation. This approach says that the VM reallocation done on the basis of their proposed idea will remain effective for some time in future. VM requirements including CPU, memory, and network constraints, as well as platform requirements, such as power budget are taken into account for taking decisions for VM placement. Time-varying probability density function is used to predict the future resource demand of applications. Authors predict that the resource utilization follows normal distribution whereas many authors showed that the resource utilization is more complex and it cannot be represents using normal distribution.

Table2.1: Comparative study of different live virtual machine migration techniques

Virtual Machine Migration Techniques	Concept	Advantages	Limitations
Stop and Copy Approach [25]	VM is firstly stopped on source, then all pages are transferred on destination physical machine and at last VM is resumed on destination host.	Simplicity.	More service downtime.
Pre Copy Approach [17]	Transfer VM's memory content first, then processor states are transferred to destination.	1) Application downtime is reduced to 1 second. 2) Migration is done in a transparent manner.	Duplicate transmission overhead.

Virtual Machine Migration Techniques		Concept	Advantages	Limitations	
Adaptive Rate Limiting Approach[28]		Use the concept of writable working set and Xen VMM is used for experiment setup.	Service downtime is low as 60ms.	High total migration time.	
Improved Pre Copy Approach	Memory Compression Techniques	Adaptive Memory Compression Approach [29]	Memory compression based VM migration approach (MECOM), that first uses memory compression.	1) It provides fast and stable VM migration. 2) Minimize the downtime and total migration time.	Compression operation includes additional overhead.
		Delta Compression [30]	The delta compression technique is implemented before basic live migration on KVM hypervisor.	1) Downtime is reduced 2) Increase total migration throughput.	Compression operation includes additional overhead.
	Frequently updated pages of memory	Page bitmap [31]	Initially identification of frequently updated pages is done and than they are added into page bitmap. Aim of decreasing transferred data is achieved by migrating these pages only once in the last round of iteration.	Total transferred data is reduced by 34% and total migration time by 32.5%.	Not suitable for wide area live VM migration.
		Time series prediction technique [32]	Prediction method is used to identify the frequently updated pages in past and future, and then transferring them at last in a single iteration.	1) Number of iterations is reduced. 2) Less downtime and migration time. 3) Fewer pages transfer.	This approach is useful only when high dirty page rate in iterations.
		Detecting memory update	Detecting memory update patterns and terminating	Downtime is reduced with least impact on	Useful for only memory intensive

Virtual Machine Migration Techniques		Concept	Advantages	Limitations
	patterns [34]	migration.	application performance.	applications.
Post Copy Approach [36]		First it migrates the VM's execution states and then target is achieved by transferring these pages at most one time in the whole process.	<p>1) Duplicate transferring of memory pages is stopped.</p> <p>2) Baseline total migration time is achieved.</p>	Downtime is more in this as compare to other approaches.
CR/TR Approach	Checkpoint/Recovery and Trace/Replay(CR/TR) Approach for LAN [38]	Execution trace logs are transferred rather than dirty pages. Log files have much less than that of dirty pages.	<p>1) There is noticeable decrease in total migration time and downtime.</p> <p>2) Less consumption of network bandwidth.</p>	CR/TR Motion is valid only when the rate with which log files are transferred is greater than the rate with which they are generated.
	CR/TR Approach for LAN+WAN [39]	Same approach is followed as in [26]. Only difference is that it is applied for LAN and WAN.	Downtime for migration is reduced by 200 milliseconds.	More cost consumption due to network devices.

Table 2.2: Comparative study of virtual machine selection techniques

Virtual Machine Selection Techniques	Total Energy Consumption	SLA Violation	No. of Virtual Machine Migration	Average SLA Violation
Highest Potential Growth Policy (HPG)	Almost same as other techniques.	High	High (if X)	Almost same as other techniques.
Minimization of Migration (MM)	<p>Almost same as other techniques.</p> <p>1)As utilization threshold increases, energy consumption decreases.</p> <p>2) Lower utilization has more impact than upper one.</p>	<p>Medium</p> <p>As utilization threshold increases, SLA violation increases.</p>	Low (Y-1495)	Almost same as other techniques.
Random Choice (RC)	Almost same as other techniques.	Low	Medium (X-1322=Y)	Almost same as other techniques.

Problem Statement: VM Allocation Policy

For most of the servers that run all time, it is very difficult to estimate the threshold limits with accuracy because to compute true threshold limit one will have to calculate entire utilization history of the host which is very difficult if the host is running since a very long time or its skew and kurtosis changes very frequently. Various methods have been proposed for the allocation of virtual machine in the host based on utilization history of the host. If the host is being underutilized it is better to turn it off to save power consumed by it. But one should avoid throttling of host because it can cause SLA violation and often lead to single point failure and also degrade performance. Various algorithms are proposed in the literature such as Inter Quartile Range Anton Beloglavoz et al. [23]. This algorithm selects hosts whose utilization history was between down and up threshold. The formula of Upper threshold is given by

$$\text{Upper Threshold} = 1 - \text{Safety Parameter} * \text{IQR (Host Utilization History)}$$

Another popular method of Host over utilization detection is Median Absolute Deviation. This method is resilient to outliers and for univariate data and formula for Median Absolute Deviation is

$$\text{MAD} = \text{median}_i (| X_i - \text{median}_j (X_j) |)$$

Threshold calculated using Median Absolute Deviation is given by

$$\text{Threshold} = 1 - \text{Safety Parameter} * \text{MAD}$$

These algorithms do not consider the whole data set and also more computational time is consumed by them. This problem of VM allocation policy is taken into account in this thesis. Dynamic VM migration consists of following steps:

- a) Deciding when to migrate a VM?
- b) Choosing which VM to migrate ?
- c) Choosing a destination where to migrate the selected VM? --> this is the area of concern of this thesis.
- d) Choosing when and which physical server to be switched on / off ?

The proposed algorithm is based on the idea that if many samples of host utilization history are taken then the mean of means of the sample will be the same as the mean of the total population. Proposed algorithm can be used to detect host over and under utilization with greater precision. The algorithm works as follows:

Step 1: Obtain Host Utilization History.

Step 2: Select Random Samples from the utilization history.

Step 3: Calculate Mean of individual samples.

Step 4: Calculate Mean of Means of this individual samples.

Step 5: if number of samples is taken sufficiently large then the Expected Value of the mean of all the individual samples will converge to the true mean.

Step 6: After obtaining Expected Value, compute threshold.

3.1 Proposed Algorithm

$$\forall y_j \in F$$

$$\text{Select } \bigcup_{i \in N} d_i \in N \leq S_0$$

$$\forall n \in N, \{N = (1, \infty)\}$$

$$\forall s, \bigcup_{K_i} S_0$$

$$G(x) \leftarrow R(X=x), x \subset F$$

$$E(x) \leftarrow \sum \frac{(x_i)}{i}$$

$$\{\forall p \in [0, \infty], \forall q \in S_0\} \sum_{p=1, q=1}^x [A \leftarrow \frac{X_{p=1, q=1}}{k_{p=1, q=1}}]$$

$$\rightarrow \sum_{i=1}^{n \in 0, \infty} \sum_{j=1}^{S_0} DS(h_i, u_i) \leftarrow A_i$$

$$\{\forall p \in [0, \infty], \forall q \in S_0\} CLT(y) \leftarrow \frac{DS(row(x_i), col(x_i))}{F_i}$$

return CLT

where,

F = Utilization History

$d = n(X)$

X = Random Variable

$E(x)$ = Expected Value

S_0 = Selection Limit

$CLT(y)$ = Central Limit Theorem

Chapter 4

Implementation Details

The solution of the problem discussed in this thesis has been proposed in the previous chapter. Focus of this thesis is on the enhancement of virtual machine (VM) allocation policies in such a way that energy consumption, number of VM migrations and SLA violation can be reduced. The Proposed algorithm can be used to detect host over and under utilization with greater precision. The proposed algorithm is based on the idea that if many samples of host utilization history are taken then the mean of means of the samples will be the same as the mean of the total population. Central Limit theorem in conjunction with minimum migration time selection policy is used to achieve the expected results.

In order to implement the proposed algorithm the following experimental setup is established.

4.1 Experimental Setup

For most of the servers that run $24 \times 7 \times 365$ it is very difficult to estimate the threshold limits with accuracy because to compute true threshold limit one will have to calculate entire utilization history of the host which is very difficult if the host is running since a very long time or its skew and kurtosis changes very frequently. To consolidate this research, CloudSim 3.0 is used as a tool for simulating the cloud. 300 hosts along with 390 virtual machines and cloudlet were modeled on which proposed CLT algorithm was applied to perform host overload and underload detection and later MMT algorithm was applied. The programming language used for this experiment was Java and Gui library used were Libgdx and swing. The platform on which this experiment was conducted was Windows 8 x64.

4.2 Proof of Correctness

To prove the correctness of the above algorithm, it is required to show that the Expected Value of all the Random variables (Sample) is a Normal Distribution. Steps followed for proof of correctness are:

- i. Find the four moments of PGF. Moments are calculated by doing derivative and putting value of $t=0$.
- ii. Find the four moments for characteristic equation of proposed algorithm. In characteristic equation pdf is taken because continuous system is considered.
- iii. Get the characteristic equation of normal distribution which is g'' in the proof.
- iv. Take relation between expected mean and true mean through Chebyshev's inequality and then prove it equal to characteristic equation of normal distribution.

The proof is stated as:

UH is Utilization history and $I(x)$ is input function. Here iid is independently identical distribution that means no correlation between random variables (Samples). PGF is probability generating function (disjoint probability distribution). In this proof t tells the order of moment. Random variables and samples are interchangeable terms.

$UH \rightarrow I(x) \wedge \forall i \in I,$

$\forall i [i \in N] I_i \rightarrow iid \{n \in \{0, \infty\}\}$

then PGF is defined as :

$$M_x(t) = E(e^{tx})$$

Expanding using Taylor series and $a = 0$

$$M_x(t) = E \left[(e^0) + (X)(t - 0) + (X^2) \frac{(t - 0)^2}{2} + \dots \right]$$

Rearranging Coefficient

$$M_x(t) = E \left[(e^0) + (X)t + (X^2) \frac{t^2}{2} + \dots \right]$$

Distributing E to the expansion

$$M_x(t) = E(1) + E(Xt) + \frac{E(X^2 t^2)}{2} + E \frac{(X^3 t^3)}{3!} \dots$$

On Taking First Derivative wrt t and evaluating t=0 we get

$$M_x(t) = \frac{d}{dt} \left\{ E[1] + E[X]t + \frac{E[X^2] t^2}{2} + E \frac{(X^3 t^3)}{3!} \dots \right\} \Big|_{t=0}$$

$$M_x(t) = 0 + E[X] + 2 E[X^2] \frac{t}{2} + 3 E[X^3] \frac{t^2}{3!} \dots$$

On Solving we get:

$$M_x(t) = E[X] \quad \text{----- [1]}$$

On Taking Derivative of (1) wrt t again and evaluating t=0 we get,

$$M_x(t) = \frac{d}{dt} \left\{ E[X] + 2 E[X^2] \frac{t}{2} + 3 E[X^3] \frac{t^2}{3!} \dots \right\} \Big|_{t=0}$$

$$\simeq 0 + 2 E \frac{[X^2]}{2} + 6 E[X^3] \frac{t}{3!} + \dots \Big|_{t=0} \simeq E[X^2] \quad \text{----- [2]}$$

On Taking Derivative of (2) wrt t again and evaluating t=0 we get,

$$\simeq 2 E \frac{[X^2]}{2} + 6 E[X^3] \frac{t}{3!} + \dots \Big|_{t=0} = 6 E \frac{[X^3]}{3!} + \dots \simeq E[X^3] \quad \text{----- [3]}$$

Similarly on taking derivative again we get

$$M_x(t) \simeq E[X^4] \quad \text{----- [4]}$$

Taking $t = 0$ without derivating we get ,

$$M_x(0) = 1 \quad \text{----- [q ']}$$

From (1) , (2) , (3) and (4) we get First , Second , Third and Fourth Moments which corresponds to μ , σ , skew and peakedness respectively

Solving Expectation of MGF for CLT Proof

Utilization History obtained from the Host $H(x): i \in N \Rightarrow \infty$

From the Definition of Moments:

$$M_x(t) = E(e^{xt}) = \int_{-\infty}^{\infty} e^{xt} * pdf \, dx$$

Because No PDF is defined for $e^{ax} < 0$

$$M_x(t) = \int_0^{\infty} e^{xt} * \lambda e^{-\lambda x} \, dx \quad [\text{since , pdf of exponential function is } \lambda e^{-\lambda x}]$$

$$M_x(t) = \lambda \int_0^{\infty} e^{xt} * \lambda e^{-\lambda x} \, dx$$

As λ is constant and using product of power rule , we get ,

$$M_x(t) = \lambda \int_0^{\infty} e^{x(t-\lambda)} \, dx$$

On Integration , we get ,

$$M_x(t) = \lambda \left(\frac{e^{x(t-\lambda)}}{(t-\lambda)} \Big|_0^{\infty} \right)$$

$\forall \lambda - t > 0$, we get ,

$$M_x(t) = \lambda \left(0 - \frac{e^0}{t-\lambda} \right)$$

$$M_x(t) = \frac{-\lambda}{t-\lambda} \quad \text{----- [5]}$$

From eq(1) we know that the first derivative of moment is μ ,

Therefore, taking first derivative of $M_x(t)$ wrt t and evaluating $t = 0$ we get ,

$$\begin{aligned} \frac{d}{dt} \left(\frac{\lambda}{\lambda-t} \right) &= \frac{d}{dt} \lambda (\lambda-t)^{-1} = (-1) \lambda (\lambda-t)^{-2} (-1) \Big|_{t=0} \\ &= \frac{\lambda}{(\lambda-t)^2} \Big|_{t=0} = \frac{\lambda}{\lambda^2} = \frac{1}{\lambda} \end{aligned}$$

From eq (2) we know that the 2nd derivative of moment is σ ,

Therefore, taking 2nd derivative of moment we get ,

$$\begin{aligned} \frac{d^2}{dt^2} \left(\frac{\lambda}{\lambda-t} \right) &= \frac{d}{dt} \left(\frac{\lambda}{(\lambda-t)^2} \right) = \frac{d}{dt} (\lambda (\lambda-t)^{-2}) \\ &= (-2) \lambda (\lambda-t)^{-3} (-1) \Big|_{t=0} = \frac{2\lambda}{(\lambda-t)^3} \Big|_{t=0} = \frac{2}{\lambda^2} \end{aligned}$$

Since , the moment derived above is centered about origin but the variance is centered at μ , therefore ,

$$\sigma^2 = E(X - \mu)^2 = E(X^2) - E(X)^2$$

$$\sigma^2 = \frac{2}{\lambda} - \frac{1}{\lambda} = \frac{1}{\lambda^2}$$

From Eq (3) and (4) we know that the 3rd and 4th moment of MGF is skew and kurtosis
Therefore , on derivating to the third and fourth power respectively we get ,

$$\frac{d^3}{dt^3} = \frac{6}{\lambda^3} \text{ and}$$

$$\frac{d^4}{dt^4} = \frac{24}{\lambda^4}$$

The above skew and kurtosis was derived about the origin.

the skew $\gamma_2 = \frac{\mu_3}{\sigma^3} = E \frac{(X - \mu)^3}{\sigma^3} = \frac{\mu_3 - 3 \mu_1 \mu_2 + 2 (\mu_1)^3}{\sigma^3}$, by converting

raw moments into central moments

$$\gamma_2 = \frac{\frac{6}{\lambda^2} - 3 \frac{1}{\lambda} \frac{2}{\lambda} + 2 \left(\frac{1}{\lambda}\right)^3}{\frac{1}{\lambda^2}} = 2 , \quad \text{On solving Algebraically}$$

similarly Kurtosis γ_2 is 6

Now we obtain MGF of standard Normal Distribution

$$M_x (t) = E [e^{tx}] = \int_{-\infty}^{\infty} e^{tx} * pdf (N (x : \mu , \sigma^2))$$

$$\text{As we know } N (x : \mu , \sigma^2) = \frac{1}{\sqrt{2 \pi \sigma^2}} e^{-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}}$$

$$\text{Therefore , } N (x : 0,1) = \frac{1}{\sqrt{2 \pi}} e^{-\frac{1}{2} (x)^2}$$

$$\text{hence , } M_x (t) = \int_{-\infty}^{\infty} e^{tx} * \frac{1}{\sqrt{2 \pi}} e^{-\frac{1}{2} (x)^2} dx$$

$$M_x (t) = \frac{1}{\sqrt{2 \pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2} (x)^2 + tx} dx$$

$$\text{we know } ax^2 + bx + c = a (x - h)^2 + k \text{ where , } h = \frac{-b}{2 a} , k = c - \frac{b^2}{4 a}$$

$$\text{Therefore , } -\frac{1}{2} x^2 + tx = -\frac{1}{2} (x - t)^2 + \frac{1}{2} x^2$$

$$M_x (t) = \frac{1}{\sqrt{2 \pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2} (x - t)^2 + \frac{1}{2} x^2} dx$$

Rearranging order of Integral components , we get

$$M_x (t) = e^{\frac{1}{2} t^2} \int_{-\infty}^{\infty} e^{-\frac{1}{\sqrt{2 \pi}} - \frac{(x - t)^2}{2}} dx$$

$$\text{And we know } \int_{-\infty}^{\infty} (\text{Density Function}) = 1,$$

$$M_x (t) = e^{\frac{1}{2} t^2} \quad \llbracket g'' \rrbracket$$

$UH \rightarrow I(x) \wedge \forall i \in I \wedge I(x)$ is iid ,

Then let $X_1, X_2 \dots X_n$ be iid and random

variables of selecting n random samples from the population space $H(x)$.

Let \hat{X} be Mean of the Random Variables from $\sum_{j=1}^n X_j$ is given by

$$\hat{X} = \frac{1}{n} \sum_{j=1}^n X_j, \text{ as } n \rightarrow \infty \Rightarrow \hat{X} \approx \mu \text{ with } p(\hat{X}) = 1$$

where , \hat{X} and μ is sample mean and true mean respectively .

$$P(|\hat{X}_n - \mu| > c) \leq \text{var}\left(\frac{\hat{X}_n}{c}\right), \text{ from Chebyshevs Inequality}$$

$$P(|\hat{X}_n - \mu| > c) \leq \frac{\sigma^2}{nc^2}$$

As $n \rightarrow \infty$ ($\frac{\sigma^2}{nc^2} \rightarrow 0$) , As σ and c are constants

Let $\mu = 0 \wedge \sigma = 1$, ie $N = N(0,1)$, because we can standardize N using

$$\frac{1}{\sqrt{n}} \frac{\sum_{j=1}^n (X_j - \mu)}{\sigma}$$

$$\text{Let } S_n = \sum_{j=1}^n X_j$$

$\therefore \forall X_i$ are uncorrelated

$$\therefore E\left(e^{\frac{tS_n}{\sqrt{n}}}\right) = E\left(e^{tX_1 \frac{1}{\sqrt{n}}}\right) * E\left(e^{tX_2 \frac{1}{\sqrt{n}}}\right) * \dots * E\left(e^{tX_n \frac{1}{\sqrt{n}}}\right)$$

$\therefore e^{tX_i}$ is Mgf and X_i are iid

$$\therefore E\left(e^{\frac{tS_n}{\sqrt{n}}}\right) = \left(M\left(\frac{t}{\sqrt{n}}\right)\right)^n$$

Now , Taking log on both side

$$\log E\left(e^{\frac{tS_n}{\sqrt{n}}}\right) = n \log\left(M\left(\frac{t}{\sqrt{n}}\right)\right)$$

Taking limit $n \rightarrow \infty$,

$$\lim_{n \rightarrow \infty} \log E\left(e^{\frac{tS_n}{\sqrt{n}}}\right) = \lim_{n \rightarrow \infty} \frac{\log M\left(\frac{t}{\sqrt{n}}\right)}{\frac{1}{n}}$$

$$\text{let } y = \frac{1}{\sqrt{n}}$$

$$\lim_{n \rightarrow \infty} \log E \left(e^{\frac{tS_n}{\sqrt{n}}} \right) = \lim_{y \rightarrow 0} \frac{\log M(yt)}{y^2}$$

Taking L'Hopital ,

$$\lim_{n \rightarrow \infty} \log E \left(e^{\frac{tS_n}{\sqrt{n}}} \right) = \lim_{y \rightarrow 0} \frac{tM'(yt)}{2y M(yt)} \quad \text{[[k'']}]$$

From equation (q'),(1),(2) we get,

$$M_X(0) = 1 \quad \text{[[6]]}$$

$$M'_X(0) = E(X) = \mu = 0 \quad \text{[[7]]}$$

$$M_X(0) = E(X^2) = \sigma = 1 \quad \text{[[8]]}$$

Substituting values of equation (6) , (7) ∈ equation k'' we get ,

$$\lim_{n \rightarrow \infty} \log E \left(e^{\frac{tS_n}{\sqrt{n}}} \right) = \frac{t}{2} \lim_{y \rightarrow 0} \frac{M'(yt)}{y}$$

Using L'Hopital Rule again we get,

$$\lim_{n \rightarrow \infty} \log E \left(e^{\frac{tS_n}{\sqrt{n}}} \right) = \frac{t^2}{2} \lim_{y \rightarrow 0} M''(yt) = \frac{t^2}{2} \quad \text{From equation (8)}$$

Taking 'e' both side we get ,

$$\lim_{n \rightarrow \infty} E \left(e^{\frac{tS_n}{\sqrt{n}}} \right) = e^{\frac{t^2}{2}} \quad \text{[[9]]}$$

From equation (g'') we can conclude that equation (9) is N(0,1)

4.3 Snapshots of Implementation

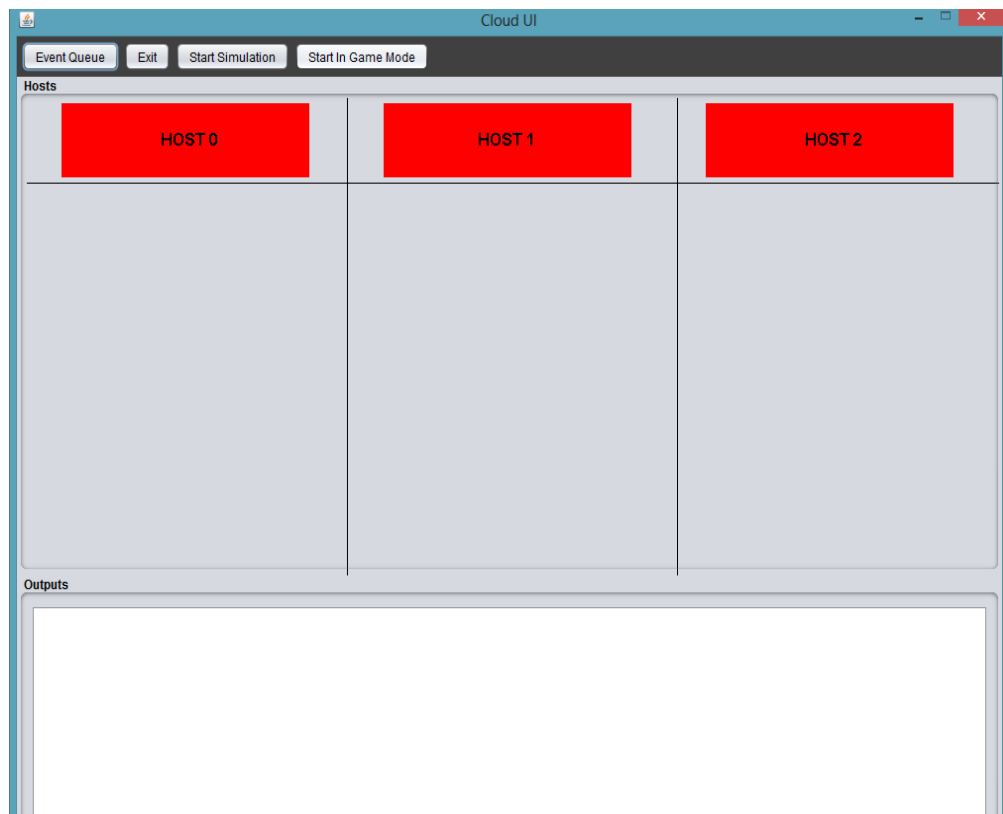


Figure 4.1: Main screen

Main screen of this implementation is shown in Figure 4.1. This screen is a default screen which appears when eclipse is opened.



Figure 4.2: Entering into datacenter

Figure 4.2 shows the screen for datacenter. This figure depicts that after starting the implementation, first thing to access is datacenter of cloud.

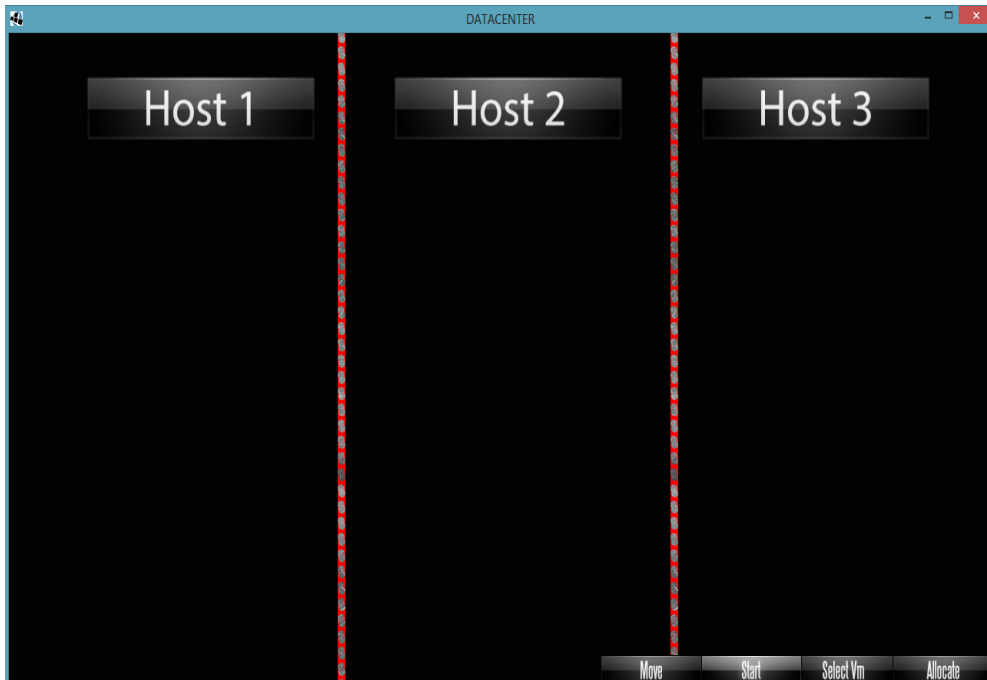


Figure 4.3: Initial stage

Figure 4.3 is showing the initial stage of cloud environment. Here just hosts are shown but they will be activated and will be allotted some resources after pressing start button. Figure 4.4 support this explanation as no resources are there for hosts.

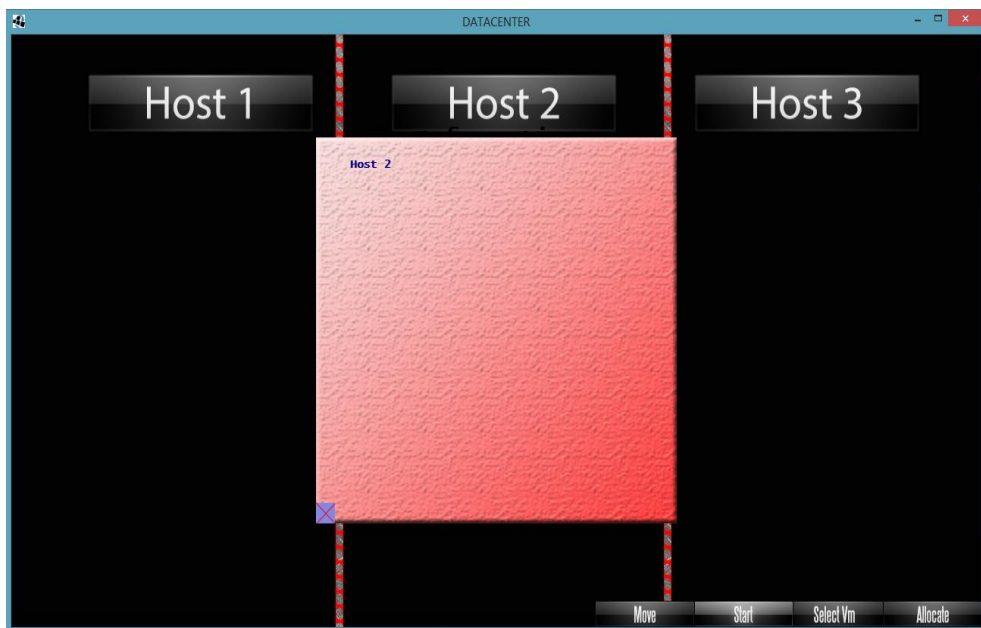


Figure 4.4: No resource allocation

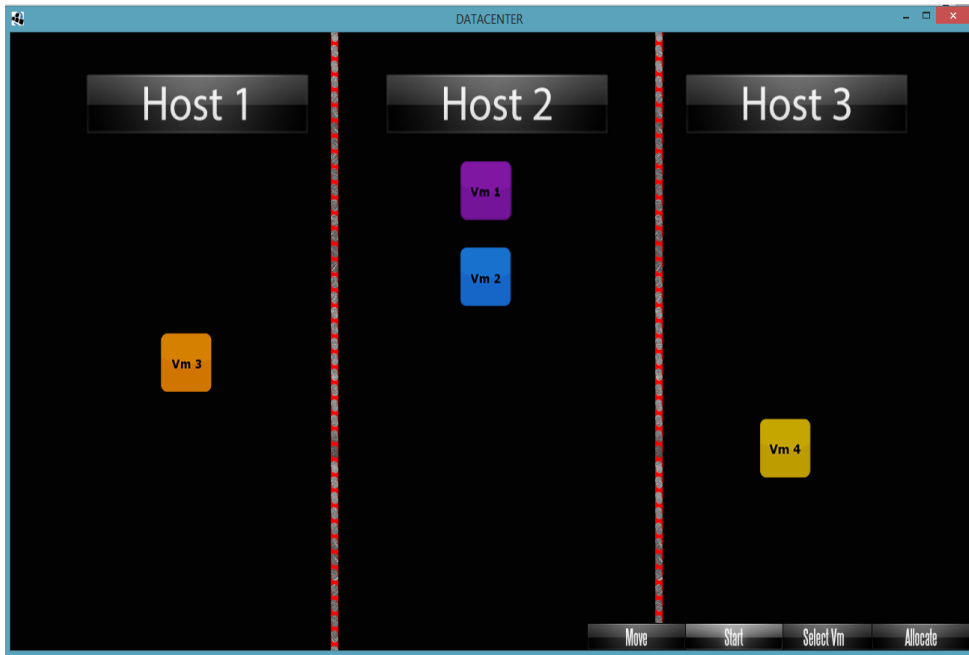


Figure 4.5: Task allocation

This figure shows the allocation of task. As it is predefined that number of tasks is equal to number of resources. So VMs are created and allocated to hosts. Each task is allocated to individual VM.

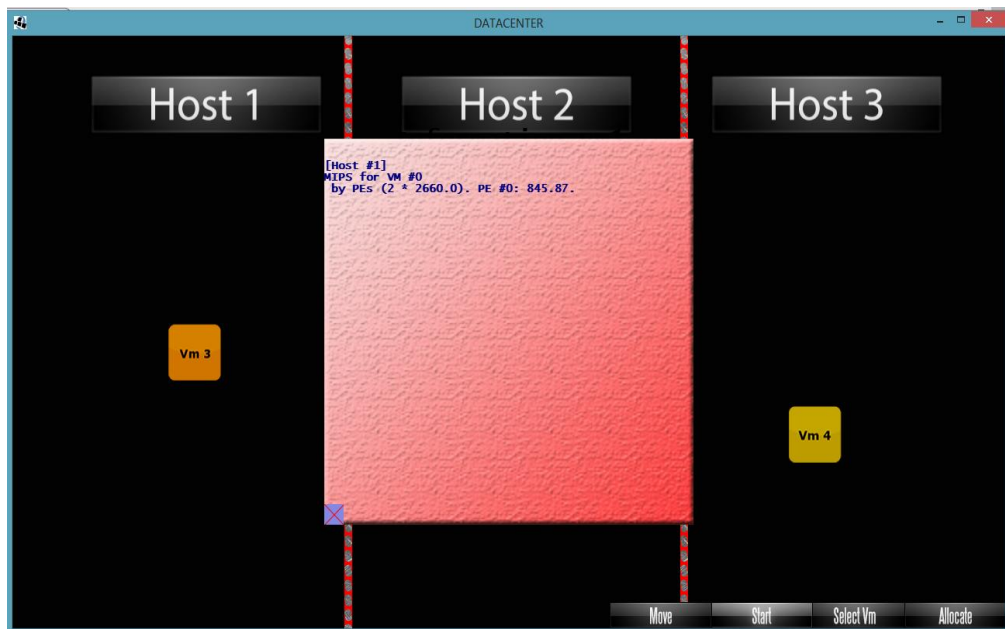


Figure 4.6: VM details

Figure 4.6 shows the details of resources allocated to VMs. This figure depicts that this VM is allocated to host id 1 and has utilization of 2660 Mips.

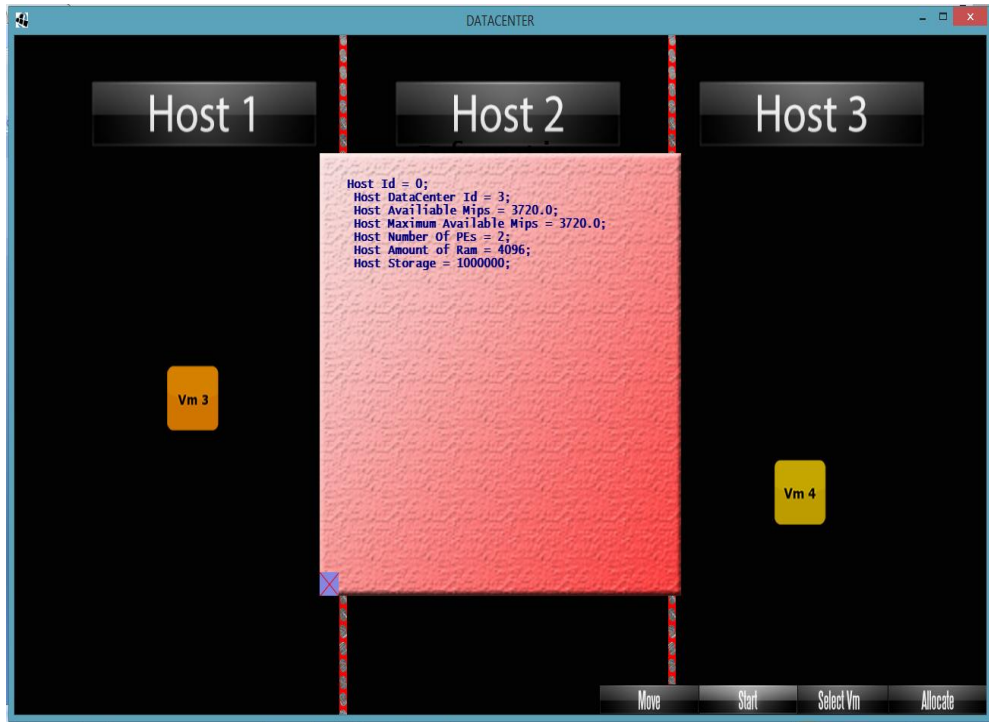


Figure 4.7: Host details

Figure 4.7 shows the resources allocated to host. Details of resources available to each host can be seen from this figure.

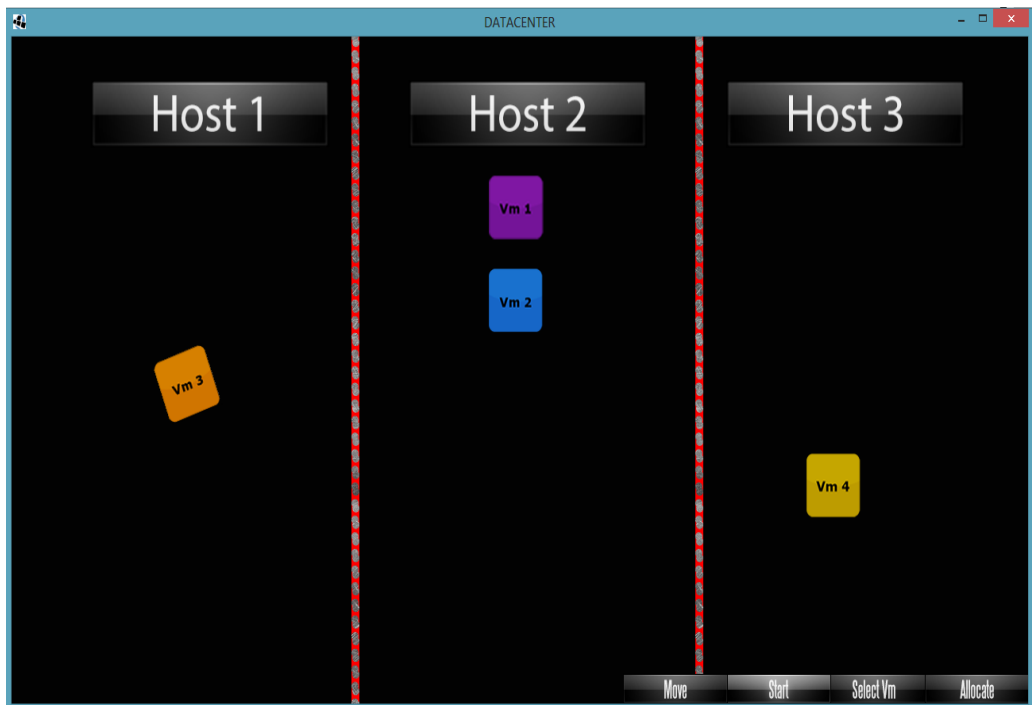


Figure 4.8: Selecting VM

The VM is selected for migration in Figure 4.8. Here minimum migration time policy is implemented at the simulator to select the VM.

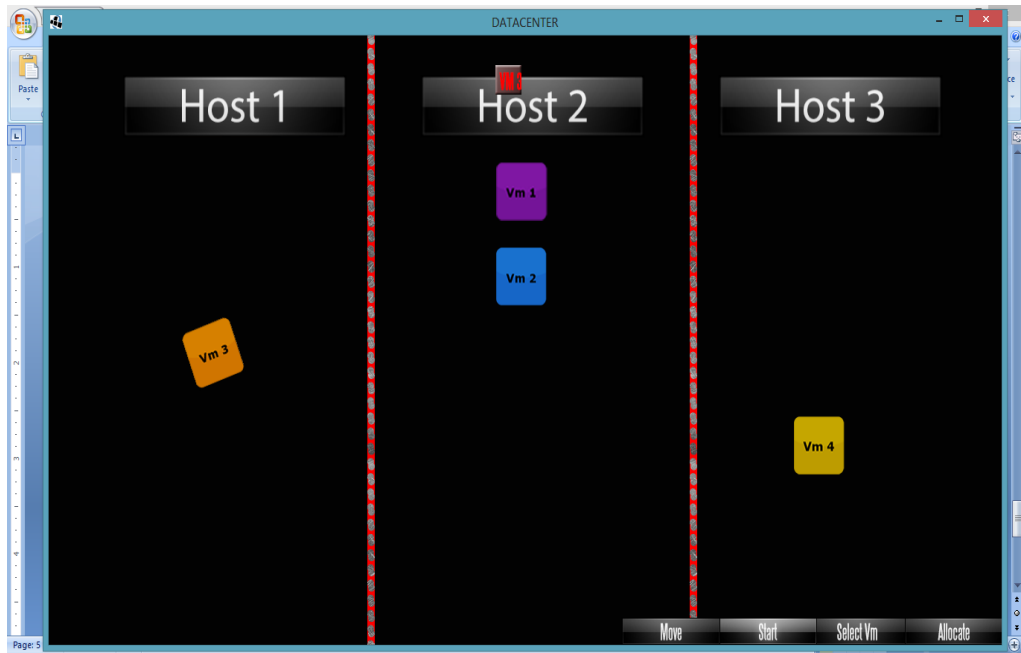


Figure 4.9: Select destination host

This Figure shows the screen where a host is selected as destination for migration of VM selected in previous step. In order to select the host, proposed allocation policy, which used CMT is executed on simulator.

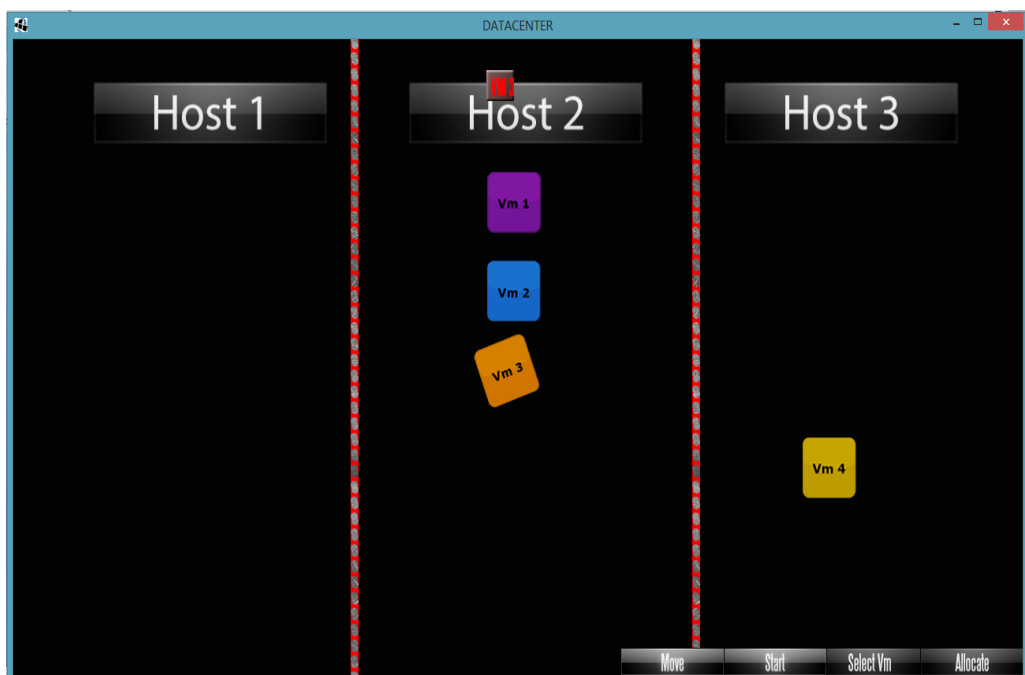


Figure 4.10: Migration

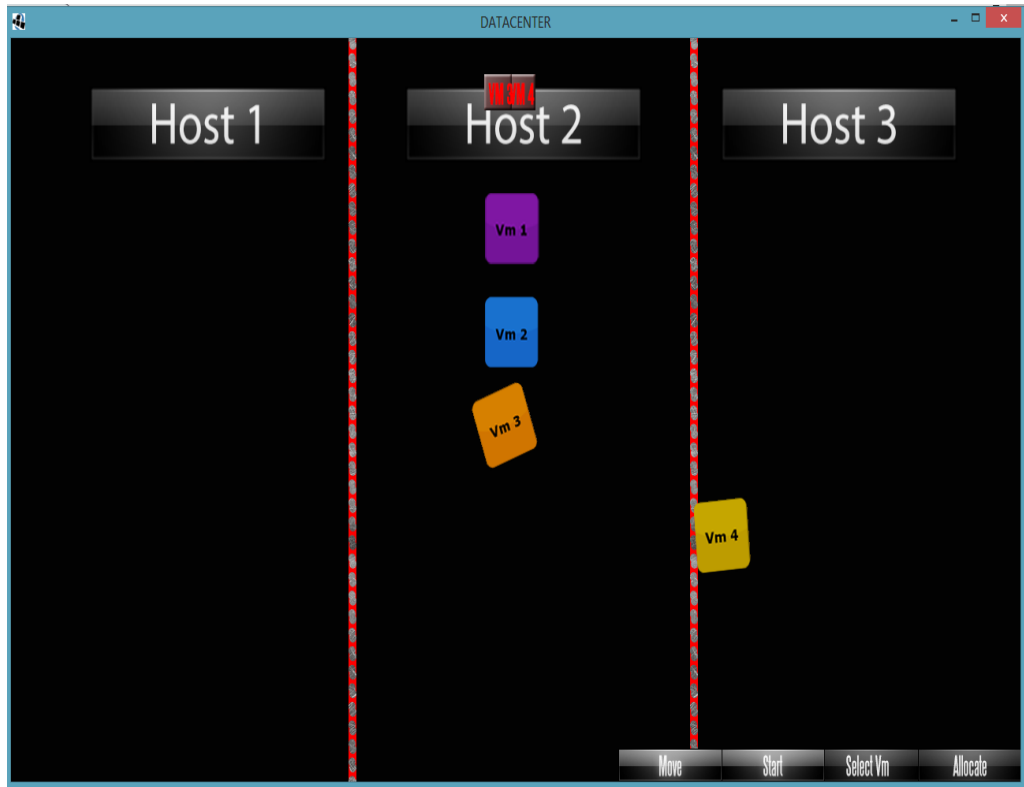


Figure 4.11: Migration

Migration of VM from one host to another is shown in Figure 4.10 and Figure 4.11.

Chapter 5

Testing and Results

5.1 Test Plan

Accuracy of proposed algorithm “Central Limit Theorem (CLT)” can be validated by comparing it with other algorithms performing the same functionality. Comparison will be done between few popular approaches and this proposed algorithm. Two approaches with which proposed algorithm will be compared are:

Inter Quartile Range (IQR) – The IQR method selects the 2/3 of the host utilization history and removes the rest outliers.

Median Absolute Deviation (MAD) – The median absolute deviation is a measure of variance of the data. The median absolute deviation measures dispersion by finding the value where half of the data is closer to the median and half of the data is farther from the median than that value.

Same input can be given multiple times and then the results of these two algorithms and the proposed algorithm can be compared. It is expected that the energy consumption, number of VM migrations and SLA violations in case of CLT will be less than the energy consume By IQR, MAD. If the testing results conforms the above statement than it will conclude that the proposed algorithm is an energy efficient allocation policy. Input set can also be changed to check the behavior of algorithms when the workload changes. Changing input set validates that the proposed algorithm can perform better in most of the scenarios. Comparison will be done in terms of 3 parameters which are: Number of Virtual Machine Migration, Energy Consumption in kWh, Overall SLA violation.

5.2 Testing

In order to test the accuracy of proposed algorithm, different input sets (which is tasks in this thesis work) is given to the system. Each input folder contains 390 tasks. In each task there are 282 entries which represent the utilization of that task at various time instances. For each task set, every algorithm is executed ten times in order to test

the accuracy of results. After execution of algorithm, the results are noted in terms of energy consumption, number of VM migrations and overall SLA violation.

Table 5.1: Testing results

Workload	Number of Attempt	Energy Consumption (kWh)			Number of VM Migrations			Overall SLA Violations %		
		CLT	IQR	MAD	CLT	IQR	MAD	CLT	IQR	MAD
20110303	10	4.41	4.58	4.58	764	792	792	0.07	0.14	0.14
20110306	10	4.27	4.44	4.44	728	747	747	0.04	0.13	0.13
20110309	10	4.28	4.45	4.45	750	774	774	0.07	0.14	0.14
20110322	10	4.14	4.31	4.31	744	767	767	0.05	0.14	0.14

Testing results shows that the proposed algorithm performs better for all types of workload taken in testing. The proposed algorithm CLT consumes less energy as compare to IQR AND MAD. The results also show that the number of VM migrations and overall SLA violations in case of proposed algorithm is much less than the other two algorithms.

Virtual Machine Migration incurs cost; it will be most suitable if it could be minimized. The main source of unnecessary virtual machine migration is sloppy allocation of virtual machine to the host. Our proposed algorithm improves it and reduces the number of virtual machine migration and thus reduces the cost associated with the migration process.

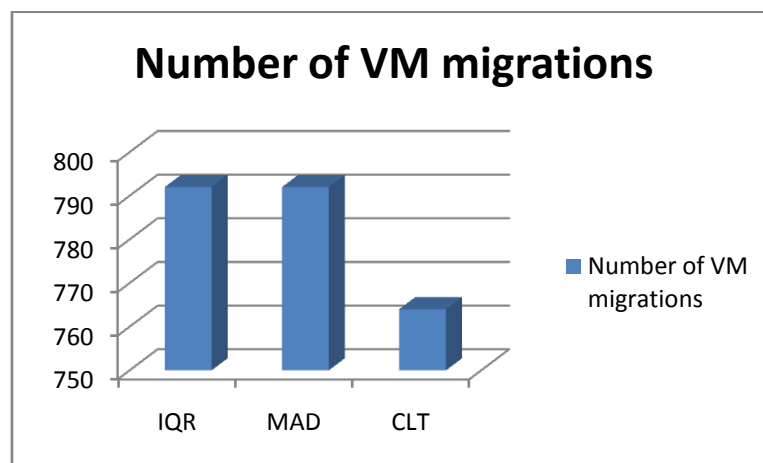


Figure 5.1: Number of VM migrations

Alone Google's data center uses 260 million watts of electricity enough to power a small city. Energy Consumption drains money and wastage of energy should be minimized. The emphasis is on migrating virtual machines from underutilized host to another host without disturbing equilibrium or unnecessary migration of caused due to inappropriate choice of hosts. The incorrect choice of host while migration process can cause chain reaction if the host is at the boundary of upper threshold limit and this should be avoided.

Though choosing the best host for the virtual machine is a bin-packing problem and considered to be np-hard. This kind of problem takes exponential time to generate solution and the size of time grows exponentially with increase in domain. But certain heuristics and rule can help in solving such kind of problem.

It is evident that our algorithm also aids to improve the management of electricity. The next table and chart shows the improvement in energy consumption.

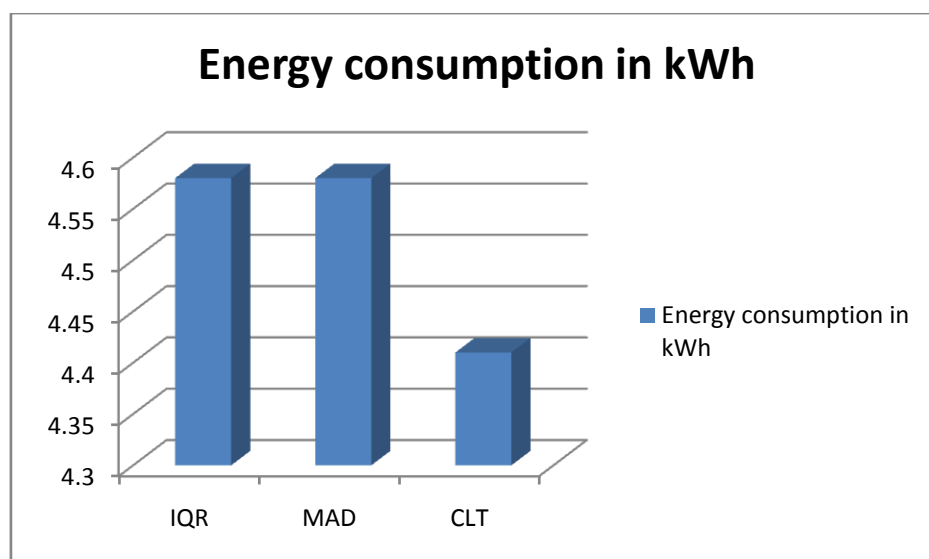


Figure 5.2: Energy consumption of different approaches

Most businesses are mission critical and cannot afford downtime but with the advent of virtual machine migration technology users experience downtime. To avoid such inconvenience an agreement between the service provider and the customer is made this agreement defines the service contents such as uptime is offered with a tool to measure the service delivery to the customer.

So the service provider such as cloud administrator needs to minimize the SLA violations. The proposed algorithm helps in reduction of overall SLA violations.

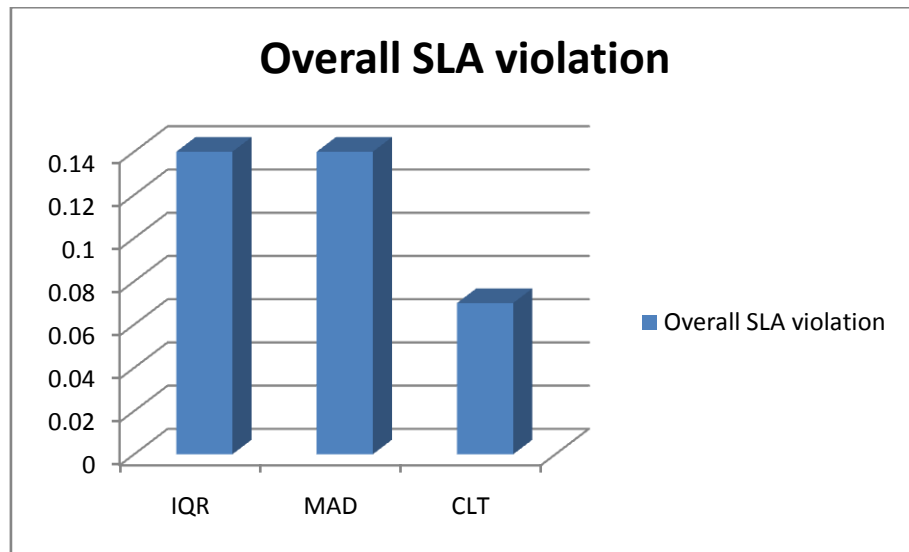


Figure 5.3: SLA violation comparison

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

This thesis focuses on enhancement of virtual machine (VM) allocation policies in such a way that energy consumption, number of VM migrations and SLA violation can be reduced. Above described enhancements are achieved in this thesis work by applying Central Limit theorem for selecting host for VM migration in conjunction with minimum migration time selection policy. In order to prove the correctness of this idea, a mathematical proof is derived in which the proposed algorithm is proved to have the normal distribution. Various performance evaluation tests are done using Cloud Sim 3.0 simulator for validating the effectiveness and accuracy of results. To conclude, the results demonstrate that Proposed Algorithm has immense potential as it offers significant energy saving with comparatively less VM migrations and SLA violations under dynamic workload scenarios.

6.2 Future Scope

In order to consider threshold, a fixed quantity is used in this thesis work. Threshold value in this thesis work is true mean plus a constant. In future, deviation can be considered in case of finalizing the threshold. As, deviation change from input or it is not wrong saying this that deviation depends on input. So, threshold can be dynamic in case of deviation. Only 68% of data is in first deviation of normal distribution while second and third deviation contains 95% and 99% of data respectively. So, in future real value of deviation can be taken rather than integer value. Finding the optimum real value can be the main challenge.

Energy consumption and SLA violations can be further minimized through various other approaches.

In this thesis testing is done for four different workload folders. More rigorous testing can be done for checking the accuracy of results for all type of scenarios.

References

- [1] C. Pettey, "Industry Accounts for 2 Percent of Global CO₂ Emissions," [Online]. Available: <http://www.gartner.com/it/page.jsp?id=503867>, 2007.
- [2] J. Koomey, "Growth in Data Center Electricity Use 2005 to 2010," Analytics Press, Tech. Rep., 2011.
- [3] C.L. Belady, "In the Data Center, Power and Cooling Costs More Than the IT Equipment it Supports," *Electronics Cooling*, vol. 13, no. 1, pp. 24, 2007.
- [4] J. G. Koomey, "Estimating total power consumption by servers in the US and the world," *Lawrence Berkeley National Laboratory, Tech. Rep.*, 2007.
- [5] L.A. Barroso, "The Price of Performance: An Economic Case for Chip Multiprocessing," *ACM Queue*, vol. 3, no. 7, pp. 48-53, Sept. 2005.
- [6] L. Minas and B. Ellison, "Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers," *Intel Press*, 2009.
- [7] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.
- [8] X. Fan, W. D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA)*, pp. 13–23, 2007.
- [9] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [10] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [11] Canonical Ltd., "Ubuntu 12.04 (Precise Pangolin) Cloud images," (accessed on 22/11/2012). [Online]. Available: <http://uec-images.ubuntu.com/precise/current/>
- [12] M. Cardosa, M. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *Proceedings of the 11th*

IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 327–334, 2009.

[13] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, “Power and performance management of virtualized computing environments via lookahead control,” *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.

[14] R. Nathuji and K. Schwan, “VirtualPower: Coordinated power management in virtualized enterprise systems,” *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 265–278, 2007.

[15] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, “No ”power” struggles: Coordinated multi-level power management for the data center,” *SIGARCH Computer Architecture News*, vol. 36, no. 1, pp. 48–59, 2008.

[16] A. Verma, P. Ahuja, and A. Neogi, “pMapper: power and migration cost aware application placement in virtualized systems,” in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 243–264, 2008.

[17] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in *Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 273–286, 2005.

[18] R. Buyya, A. Beloglazov and J. Abawajy, “Energy-efficient management of datacenter resources for cloud computing: a vision, architectural elements, and open challenges,” in *Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2010, Las Vegas, USA, 2010*.

[19] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Blackbox and gray-box strategies for virtual machine migration,” in *Proceedings of the 4th USENIX conference on Networked systems design and implementation*, ser. NSDI’07. Berkeley, CA, USA: USENIX Association, pp. 229-242, April 2007.

[20] B. Gerofi, H. Fujita, and Y. Ishikawa, “An efficient process live migration mechanism for load balanced distributed virtual environments,” in *IEEE International Conference in Cluster Computing (CLUSTER)*, pp. 197 –206, Sept. 2010.

[21] Bernard Golden, *Virtualization for Dummies*, Wiley India, 2008.

[22] A. Beloglazov, J. Abawajy and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,” *Future Generation Computer Systems*, pp.755-768, May. 2012.

- [23] A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," *Concurrency and Computation: Practice and Experience (CCPE)*, Wiley Press, New York, USA, pp. 1397–1420, Sep. 2012.
- [24] Z. Cao and S. Dong, "Dynamic VM Consolidation for Energy-Aware and SLA Violation Reduction in Cloud Computing," in *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2012 13th International Conference on*, pp. 363-369, IEEE, 2012.
- [25] K. Yang, J. Gu, T. Zhao and G. Sun, "An Optimized Control Strategy for Load Balancing based on Live Migration of Virtual Machine," in *Processing of 2011 Sixth Annual Chinagrid Conference (ChinaGrid)*, pp. 141-146, 2011.
- [26] M. Nelson, B. H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," in *Proceedings of USENIX Annual Technical Conference (USENIX'05)*, Marriott Anaheim, Anaheim, CA, USA, pp.391-394, April 10-15, 2005.
- [27] B. Panchal and R. K. Kapoor, "Dynamic virtual machine allocation algorithm using clustering in cloud computing," in *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, 2013.
- [28] A. Kochut and K. Beaty, "On strategies for dynamic resource management in virtualized server environments," *15th International Symposium in Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp.193–200, IEEE, 2008.
- [29] J. Hai, D. Li, W. Song, S. Xuanhua, and P. Xiaodong, "Live virtual machine migration with adaptive memory compression," in *Proceedings of the 2009 IEEE International Conference on Cluster Computing (Cluster 2009)*, pp. 1-10, 2009.
- [30] P. Svard, B. Hudzia, J. Tordsson, and E. Elmroth, "Evaluation of delta compression techniques for efficient live migration of large virtual machines," in *Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, ser. VEE '11. New York, NY, USA: ACM, pp. 111–120, 2011.
- [31] F. Ma, F. Liu, and Z. Liu, "Live virtual machine migration based on improved pre-copy approach," in *IEEE International Conference on Software Engineering & Service Sciences ICSESS*, pp. 230-233, 2010.

- [32] B. Hu, Z. Lei, Y. Lei, D. Xu, and J. Li, "A time-series based precopy approach for live migration of virtual machines," in *2011 IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 947-952, Dec. 2011.
- [33] W. Huang, Q. Gao, J. Liu, and D. K. Panda, "High Performance Virtual Machine Migration with RDMA over Modern Interconnects," in *Proceedings of IEEE International Conference on Cluster Computing (Cluster'07)*, pp. 11-20, September 2007.
- [34] K. Z. Ibrahim, S. Hofmeyr, C. Iancu, and E. Roman, "Optimized precopy live migration for memory intensive applications," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1-11, 2011.
- [35] E. P. Zaw and N. L. Thein, "Improved Live VM Migration using LRU and Splay Tree Algorithm," *International Journal of Computer Science and Telecommunications*, Volume 3, Issue 3, pp. 1-7, March 2012.
- [36] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proceedings of the ACM/Usenix International Conference on Virtual Execution Environments (VEE'09)*, pp.51-60, ACM, 2009.
- [37] M. R. Hines, U. Deshpande and K. Gopalan, "Post-copy live migration of virtual machines," *SIGOPS Oper. Syst. Rev.*, vol. 43, pp. 14-26, July 2009.
- [38] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in *Proceedings of the 18th International Symposium on High Performance Distributed Computing (HPDC'09)*, pp.101–110, 2009.
- [39] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in *Proceedings of the 18th ACM international symposium on High performance distributed computing*, pp. 101-110, ACM, 2009.
- [40] J. Stoess, C. Lang, and F. Bellosa, "Energy management for hypervisor-based virtual machines," in *Proceedings of the USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, pp. 1–14, 2007.
- [41] E. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *Power-Aware Computer Systems*, ser. Lecture Notes in Computer Science, B. Falsafi and T. Vijaykumar, Eds. Springer Berlin Heidelberg, vol. 2325, pp. 179–197, 2003.

- [42] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 303–314, 2005.
- [43] R. Nathuji, C. Isci, and E. Gorbato, "Exploiting platform heterogeneity for power efficient data centers," in *Proceedings of the 4th International Conference on Autonomic Computing (ICAC)*, pp. 5–5, 2007.
- [44] R. Nathuji and K. Schwan, "VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems," in *Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles (SOSP'07)*, pp. 265-278, October 2007.
- [45] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.
- [46] M. Cardoso, M. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *Proceedings of the 11th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 327–334, 2009.
- [47] A. Verma, P. Ahuja, and A. Neogi, "pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware'08)*, Belgium, pp.243-264, Springer, December 1-5, 2008.
- [48] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," *Computer Networks*, vol. 53, no. 17, pp. 2905–2922, 2009.
- [49] S. Kumar, V. Talwar, V. Kumar, P. Ranganathan, and K. Schwan, "vManage: Loosely coupled platform and virtualization management in data centers," in *Proceedings of the 6th International Conference on Autonomic Computing (ICAC)*, pp. 127–136, 2009.
- [50] M. Loris, Y. Yang, H. Casanova, and Y. Robert, "Steady-state scheduling of multiple divisible load applications on wide-area distributed computing platforms," in *International Journal of High Performance Computing Applications* 20, Volume 3, pp. 365-381, 2006.
- [51] I. Morihara, T. Ibaraki and T. Hasegawa, "Bin packing and multiprocessor scheduling problems with side constraint on job types," *Discrete applied mathematics* 6, Volume 2, pp. 173-191, 1983.

List of Publications

- [1] C. Kaur and S. Bawa, “Review of virtual machine selection and migration techniques,” in *Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA-14)*, Elsevier, Conference on 1-2 Aug, 2014. [Accepted]
- [2] C. Kaur and S. Bawa, “An energy efficient virtual machine allocation policy in cloud environment,” in *Journal of Future Generation Computer Systems*, Elsevier. [Communicated]