Scheduling and Efficient Energy Utilization in Cloud System

Abhishek Gupta
Sambeet Kar

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Scheduling and Efficient Energy Utilization in Cloud System

Thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Abhishek Gupta
(Roll: 110CS0458)

AND

Sambeet Kar
(Roll: 110CS0073)

under the guidance of

Dr. Suchismita Chinara

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

May 2014

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 10, 2013

Certificate

This is to certify that the work in the thesis entitled Feature Detection using S-Transform by Abhishek Gupta and Sambeet kar are record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. Suchismita Chinara
Assistant Professor.

Acknowledgment

"If God brings you to it, He will bring you through it."

Thank you God for showing me the path. . .

We owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, we would also like to express my gratitude towards my project advisor, Dr. Suchismita Chinara for believing in me and providing us with the opportunity to work in the challenging area of Cloud Computing. Her dedication has motivated us to work for excellence.

We would like to thank all our friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

Most importantly, none of this would have been possible without the love and support of our parents. Our parents to whom this dissertation is dedicated to, has been a constant source of love, support, strength, motivation and inspiration. We would like to express our heart-felt gratitude to them.

Abhishek Gupta

Sambeet Kar

Authors Declaration

We hereby declare that all the work contained in this report is our own work unless otherwise acknowledged. Also, all of our work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

Abhishek Gupta

Sambeet kar

NIT Rourkela

Abstract

Cloud computing is an emerging topic on software and distributed computing based on Internet, which means users can access storages and applications from remote servers by web browsers or other fixed or mobile terminals. In a cloud framework different services such as servers, storage in the form of data as well as Big data, resources etc are given to a management's computers and different devices on interest through the Internet. Multiple clients want to run their jobs or cloudlets in the cloud at a particular instant of time. The tasks are executed depending on the number of processors available and the scheduling policy of the cloud. In a cloud simulation software such as CloudSim a two level scheduling in the form of Space-shared and Time-shared can be used in collaboration with First Come First Serve(FCFS). An efficient way of job scheduling in cloud is to assign weightage or priority to the various parameters coming along with the job also taking into consideration the priority value set by the client to the task. In this thesis an attempt has been made to develop an efficient priority algorithm for the jobs running in the cloud. Also an attempt has been made to reduce the energy consumption at a particular over utilized node as well as switching idle nodes to the sleep mode thereby optimizing resource usage and reducing energy consumption.

Contents

List of Figures

Chapter 1

Introduction

Cloud computing is often used to refer to a new paradigm. People even regard it as a new technology which has evolved and adopted other existing technologies and paradigms. Cloud computing offers IT resources and services over the Internet on a pay-per-use basis.[1] When we search the word cloud or cloud computing on the internet one can find over a billion searches depicting a feeling of very high interest on the topic. Almost everybody in the IT sector speaks about cloud computing. Yet it is considered to be in its infancy and the concept is somewhat unclear to many. As of date there are few scientific contributions that have come forward to portray the actual definition of the cloud computing phenomenon.

Youseff et al. were among the first who tried to provide a complete understanding of cloud computing and all of its components and infrastructure. They regard cloud computing as a "collection of many old and few new concepts in several research fields like Service-Oriented Architectures (SOA), distributed and grid computing as well as Virtualization"[2,3]. According to Youseff et al. "cloud computing can be considered a new computing paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction" [3].

According to Armbrust et al. "Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public.

Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds".[3]

The majority of definitions of cloud computing however originate from service providers, market research companies and consulting firms. The market research company IDC for example defines cloud computing very general as "an emerging IT development, deployment and delivery model, enabling real-time delivery of products, services and solutions over the Internet" (Gens 2008). Cloud computing can be regarded as the technological basis for offering business and customer solutions in real time over the internet along with services required by the user.

The word cloud in cloud computing is used as a metaphor for "the internet". Computing can be defined as any IT activity carried out by an organization. [1]Cloud Computing is defined as an internet based computing whereby different services in the shared resources , servers, information and shared resources are provided to an organisation's computer or other devices through the internet.[14] Cloud computing is the best answer for the developing IT industry. Yet it is acknowledged to be in its earliest stages with more innovations heading up as time advances. It is membership based pay for every utilization benefit that progressively over the Internet, amplifies IT's current proficiencies. Cloud computing relies on sharing computing assets instead of having the nearby servers or particular gadgets to handle the provisions gave by the cloud.

Cloud computing makes utilization of SaaS (Software as a Service)[1].Using this, a solitary or different provisions might be conveyed to many clients utilizing a system. It can additionally be acheived through utility computing which makes utilization of virtualization or virtual machines that build the capacity competence and the exactness and pace. Platform as a Service(PaaS) gives a versatile process stage including middleware for provisions. Infrastructure as a Service (IaaS) empowers the servers, datacenters and systems to be provisioned on interest utilizing a pay for every utilization premise. These are talked about in detail afterwards.

Chapter 2

Cloud Computing

2.1 History

The development of cloud might be gone once more to the 1960's from the thoughts of pioneers like J.c.r Licklider and John Mccarthy. John McCarthy believed that cold computation can be someday be organised into a public entity. Cloud could be gotten to whenever utilizing the web. So the Internet development was the significant turning point for the cloud computing. Arpanet which was the first venture in the improvement of web was created in the year 1969. Arpanet was produced out of a task did by Advanced Research Projects Agency(Arpa)[1]. J.c.r Licklider alongside other individuals was answerable for the advancement of Arpanet. TCP/IP made it conceivable to join entire subnets to Arpanet in 1983. At the starting it was utilized for military and investigative purposes yet with the approach of World Wide Web in 1989 by Tim Berners-Lee the web got extremely well known.

The idea of grid computing began at late 1980's the point at which an extensive number of Systems were connected to a solitary issue utilizing different levels of parallel computing. In the 1990's the idea of virtualization arrived at another level past virtual servers when a virtual stage which incorporates capacity, system and provision was presented. In 1997, the expression "cloud computing" is authored by Information Systems Professor Ramnath Chellapa.[1]

In 1999,software as a Service(SaaS) was offered.[1] This made ready for the product firms to convey requisitions over the web. In the year 2000 Amazon.com understood that cloud structural engineering increments interior proficiency and started to overhaul its data centers. Amazon Web Services was then propelled that offered a mixture of cloud based administrations.

Amazon then propelled Elastic Compute Cloud(EC2) Service that permitted little organizations and people to lease machines to run their own particular provisions.

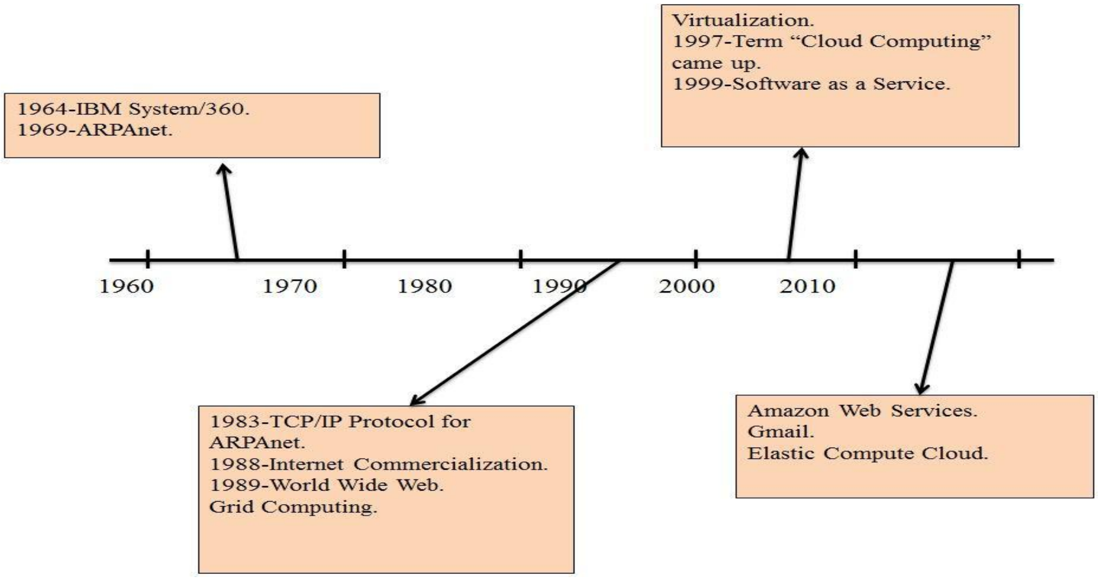


Figure 2.1:Figure depicting history of cloud

2.2 Cluster, Grid And Cloud

Cluster figuring is the cause of all processing frameworks. Cluster processing might be characterized as an arrangement of autonomous machines which are associated together to give administrations and calculation tantamount to an expansive machine.[1]

A Grid is a fittings and programming framework that clusters and coordinates high-end machines, systems, databases from various sources to structure a virtual supercomputer on which clients can work collectively inside virtual organisations.[1] Grid registering is the place more than one machine directions to tackle an issue together generally utilizing parallelism. A middleware programming is utilized that deals with the machines as a part of the grid. Likewise it catches disappointments, new workstations being included and machines being evacuated from the grid. This middleware provision could be relegated to the grid and they will utilize whatever Cpu usage and capacity the grid can accommodate their current requests. Cloud registering develops from grid processing.[8] Then again cloud processing doesn't get to assets it requires straightforwardly. It gets to them through an administration. When it obliges hard drive for capacity or a particular CPU for processing it contacts the administration that gives these assets. The administration then maps the solicitations of assets to its physical assets. It then alertly allots the assets as they are required.

In the same way if a provision requires just a little measure of some asset, say reckoning, then the administration just distributes a little sum, for instance on a solitary physical CPU. On the off chance that the provision needs a lot of some asset, then the administration apportions CPU as per the necessity, for this situation say a grid of Cpu's.

Grids are essentially on-premise,and possessed by an association while clouds are ordinarily given by sellers on a pay for every use premise by different associations.[1] Grids don't can give servers individually,including introducing a mixture of working framework and programming provisions like IaaS in a cloud does. A cloud can help a system of grids. It can additionally run on non-grid situations, for example, three-level Web construction modeling or Web 2.0 Applications. Additionally grids are generally free utilized by scholastic examination and so on inasmuch as cloud is an administration in which the administration supplier charges as indicated by the measure of work asked. Separated from their disparities, both grid and cloud registering have a few likenesses that is both are versatile, on interest register and oblige capacity.

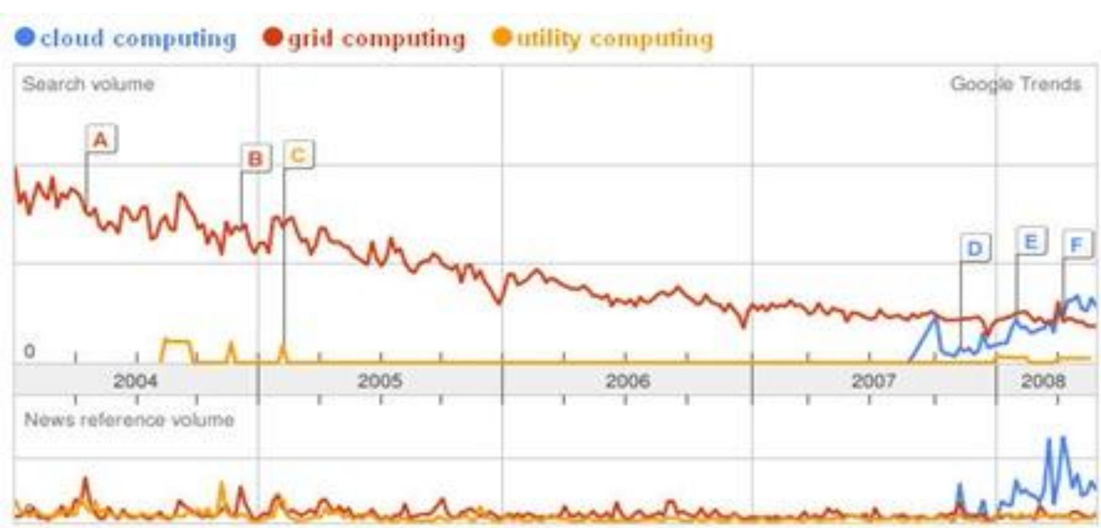


Figure 2.2:Usage of Utility, Grid & Cloud Computing

2.3 Components of a Cloud System

Clients: Clients are the gadgets that the end clients utilization to interface with the cloud as they oblige the administrations of the cloud. They could be Pcs, laptops, brilliant cell telephones and so forth. Slight clients are the machines that don't hold inside hard drives and afterward show the data from the server. Thick cloud is a typical workstation that associates with the cloud utilizing web programs like Internet Explorer, Mozilla Firefox and so on. Slender clients have turned into a mainstream result due to their decreased value and improved information security. Information security is all the more if there should arise an occurrence of slender clients as the information handling and stockpiling happens straightforwardly in the server without including a hard drive.[1]

Datacenter: It is an accumulation of servers where the provision to which you have subscribed is stored. It might be set anyplace and could be accessed through the web. A finer result is to use virtual servers through a solitary physical server. Using a software various instances of virtual servers can be run when the main server is being accessed.[1]

Databases: Databases are an integral part of every cloud where information or data is being stored.[1]

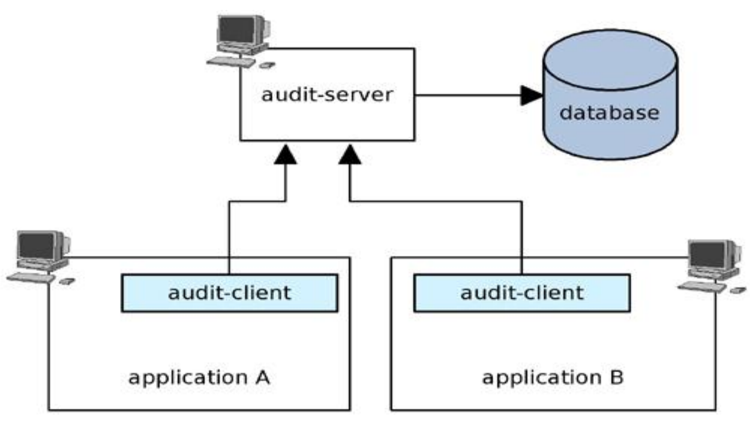


Figure 2.3: Components of Cloud

2.4 Types of Cloud

Private cloud: Private cloud is cloud infrastructure that operated solely for a single organization, which can be managed internally or by a third-party and can be hosted internally or externally. A private cloud project under taking requires a degree of engagement significant and level to virtualize the business environment. Private cloud has a significant hardware, environmental controls, requiring allocations of space and has physical footprint.[1,4]

Public cloud: A cloud is called a "public cloud" when the services are extracted over a network which is open for use of public. There is hardly any difference between private and public cloud architecture technically, however, security consideration may be substantially different for services that are made provided by a service provider for a public audience.[4]

Hybrid cloud: Hybrid cloud is a composition of two or more clouds that remain distinct entities but are bound together, offering the benefits of multiple deployment models. It cloud also mean the ability to connect, managed and dedicated services with resources of cloud.[1,4,7]

Community cloud: Community cloud shares infrastructure between several organizations from a specific community with common concerns, which are managed internally or by a third-party and hosted externally or internally. The expenses are spread over fewer clients than a public cloud (yet more than a private cloud), so only some of the cost savings prospective of cloud computing are to be took in.[4]

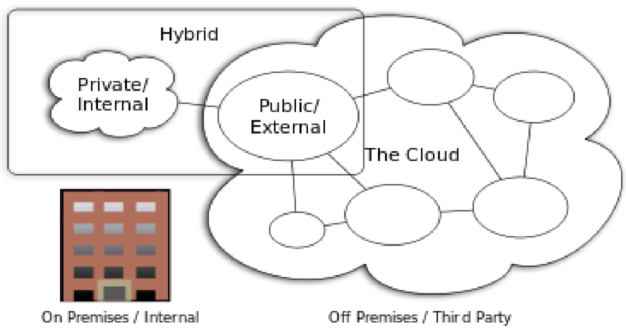


Figure 2.4:Types of Cloud

2.5 Service Deployment Models

Infrastructure Layer:It is considered as the first layer of cloud computing. It is also called Infrastructure as a Service (IaaS).It offers services in the form of data storage, communication, hardware equipments and networking components. IaaS uses virtualization technology. Technology permits datacentre provider to conform their assets on interest, accordingly creating it more productive. Storage of data in the cloud is referred to as Database as a Service. Daas permits clients to store their information on disk that might be available from anyplace. Examples of Daas are Oracle database 11g ,Microsoft SQL Server Data Services and so on.[1,5]

Software Environment Layer: It is the second layer of cloud computing. It also referred as the platform layer as it provides for developers a programming language platform to work on the cloud applications. This service is called as Platform as a Service(PaaS). It also gives API that can be exploited for many cloud services. [1,5]

Application Layer: It is also called Software-as-a-Service and it delivers complete applications to a end user of cloud. It is the final layer of service model of cloud. It is accessed by a service oriented architectures and web portal based on web service technologies.[5]

2.6 Cloudsim

Cloudsim provides extensible and generalized framework which allows experimentation, modeling and of cloud computing application and different infrastructures services.[2]

Key Terms:

Virtualization: Virtualization permits sharing of storage devices and servers in which applications can be easily moved from a server to another server**.**

Datacenters: Datacenters are collection of servers where subscribed application is stored. Application can be stored and can be accessed by means of internet from anywhere.

Host: Host performs actions linked to management of virtual machines. It can host virtual machines and is associated with datacenter.

Virtual machines: A software implementation of a machine that executes programs alike physical machine.

Cloudlet: The job or task that is to be executed in Cloudsim.

Chapter 3

Scheduling Policies

3.1 CloudSim Architecture

When we try for the planning approaches how about we observe into the structural engineering of cloudsim. The complicated configuration of Cloudsim software and its structural parts is shown by Figure 3.1. Introductory arrivals of Cloudsim utilized Sim java as the discrete occasion simulation motor that backings a few center functionalities, for example, queuing and preparing of occasions, production of Cloud framework elements (services, host, datacenter, broker , vm), correspondence between parts, and administration of the clock.

The Cloudsim simulation layer gives backing to and modelling of virtualize Cloud-based datacenter situations including devoted management interfaces for data storage, memory,VM, and transfer speed. The basic concerns, for example, allocating of hosts to VM, overseeing provision execution,and observing element framework states,are taken care of by this layer. A Coud supplier, who needs to study the proficiency of distinctive approaches in distributing it has to VM ,might need to execute his systems at this layer. Such execution is possible by automatically enlarging the center VM provisioning usefulness. There is an agreeable qualification at this layer identified with provisioning of hosts to VMs. A Cloud host might be simultaneously apportioned to a set of VMs that execute provisions focused around SaaS supplier's characterized QoS levels.[2] This layer likewise uncovered the functionalities that a Cloud provision engineer can reach out to perform complex workload profiling and requisition execution study.[2] The top-most layer in the Cloudsim stack is the User Code that uncovered essential elements for hosts (number of machines, their determination, et cetera), provisions (number of assignments and their necessities), VMs, number of clients and their requisition sorts, and merchant booking strategies. By developing the essential elements given at this layer, a Cloud requisition designer can perform the accompanying exercises: (1) create a mix of workload demand disseminations, provision arrangements; (2) model Cloud accessibility. situations and perform strong tests focused around the custom setups; and (3) actualize custom requisition provisioning methods for mists and their league.

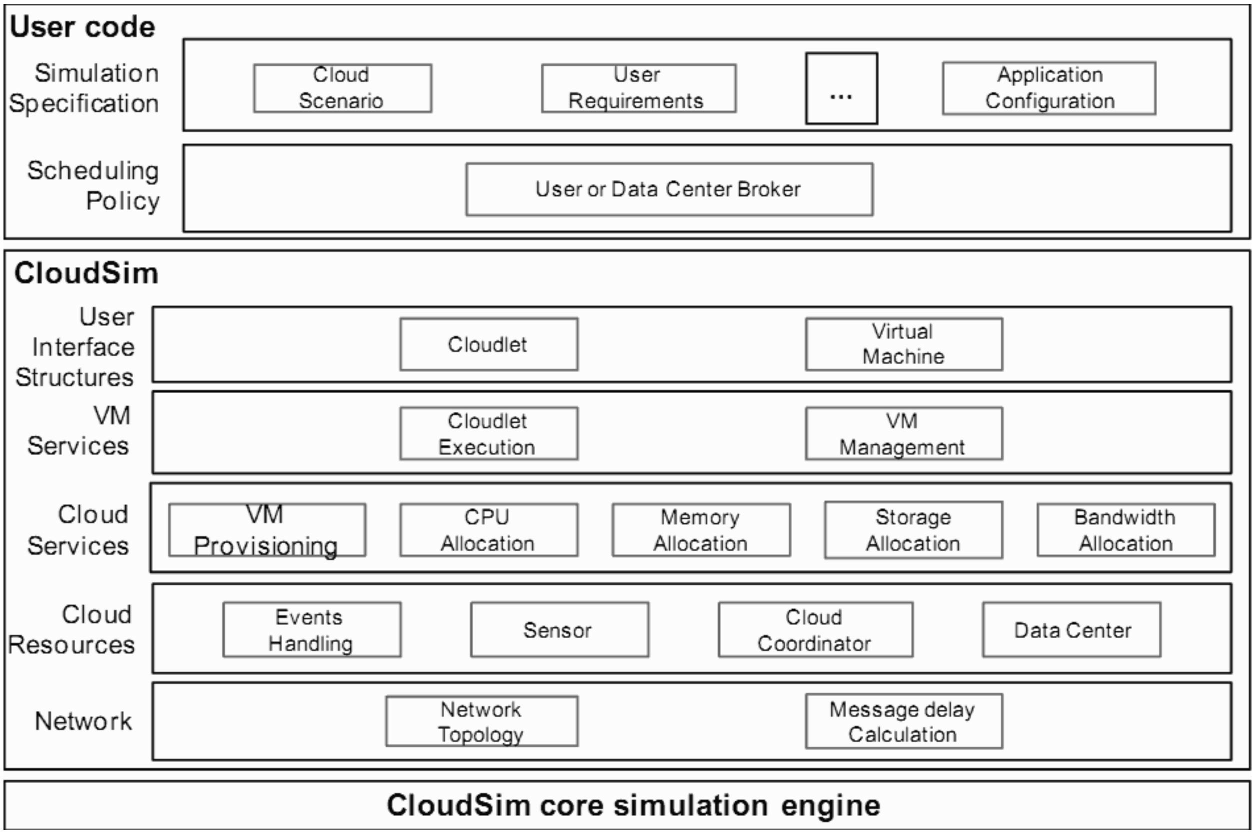


Figure 3.1:Architecture of CloudSim.

3.2 Cloud Modelling

The infrastructure-level services can be simulated using cloudsim by extension of cloudsim data centre entity. The data centre element deals with various host substances. One or more VM are allocated to host according to VM allocation policy. Here VMs policy remains for the operations control approaches identified with life cycle of VM, for example, provisioning of a host to a VM,[2] VM creation, VM pulverization, and VM movement. So also, one or more provision administrations might be provisioned inside a solitary VM example, alluded to as requisition provisioning in the connection of Cloud registering. In the connection of Cloudsim, a substance is an occurrence of a segment. A part of cloudsim could be a class (conceptual or finish) or set of classes that speak to one cloudsim model. A data center can deal with a few has that thus oversees VMs throughout their life cycles.[2] Host is a part of cloudsim that speaks to a physical registering server in a Cloud: it is relegated preconfigured handling proficiencies (communicated in a large number of directions for every second—MIPS), memory, storage, and a provisioning approach for allotting preparing centers to VMs.[2] The Host segment executes interfaces that help displaying and reproduction of both single-center and multi-center hubs.

A two level booking is utilized: initial one is for assigning VM's to the host in the datacenter and the second is for distributing the cloudlets to the virtual machines. In time imparted booking transforming force is simultaneously imparted by the VMs or undertaking relying upon which time planning strategy is connected. In space imparted planning the transforming force is allotted to VMs or undertaking in a consecutive manner.

The process of creating instances of VMs on host which satisfied the basic qualities (like memory and storage), requirements and configuration of services as software provider is called as VM allocation .It helps advancement of conventional application models that could be conveyed inside a VM case and its clients are obliged to enlarge the center Cloudlet object for executing their provision administrations. Moreover, it does not implement any confinement on the administration models or provisioning strategies that engineers need to actualize and perform tests with. Once a provision administration is characterized and demonstrated, it is appointed to one or more pre instantiated VMs through an administration-particular allotment arrangement. Allotment of provision-particular VMs to have in a Cloud-based data center is the obligation of a VM Allocation controller segment (called VM allocation policy). This part uncovered various custom strategies for analysts and engineers who help in the usage of new arrangements focused around advancement objectives (client driven, framework driven, or both). Of course, VM allocation policy executes a clear strategy that allots VMs to the Host on a First-Come-First-Serve (FCFS) premise.[2] Equipment prerequisites, for example, the amount of transforming centers, memory, and capacity, structure the premise for such provisioning. Different strategies, including the ones liable to be communicated by Cloud suppliers, can additionally be effortlessly reproduced and displayed in it. Nonetheless, strategies utilized by open Cloud suppliers are not openly accessible, and subsequently a pre implemented form of these calculations is not furnished with it.

VM are allocated using host allocation policy for each hosts. Host allocation policy considers a few equipment aspects, for example, number of cores of CPU, share of CPU, and memory capacity, that are apportioned to a given instant of VM . Thus can be done in 2 different way firstly, assign specific CPU cores to specific VM (space-share), Secondly, distributing the capacity of a cores among VM dynamically (time-shared), or VMs are assigned with cores when required.[2]

A VM scheduler component is instantiated by each host which cloud either uses the space-share or time-share policy for assigning cpu cores to VM .Details related to time and space share is mentioned in next sections. Central programming and hardware arrangement parameters identified for VM is characterized in the VM class.

If for both allocating cloudlet to VM and for allocating VM to hosts,space shared is used.[2,9]
Then the evaluated time is given by the formula

$$eft(c) = est + rl / (limit \times cores(c))$$

where eft(c) is the evaluated completion time, est is the assessed beginning time, and cores(c) is the no of cores needed by the Cloudlet.

If for allocating cloudlet to vm ,time shared is used and for allocating vm to hosts ,space shared is used.
Then the evaluated time is given by formula

$$eft(c) = ct + rl/(limit \times cores(c))$$

Where eft(c) is the evaluated completion time, cores(c) is the no. of cores (pes) needed by Cloudlet and ct is the current time.

If time-shared distribution is connected to both Vms and job units. Thus, the handling force is simultaneously shared by the Vm and the shares of every VM are all the concurrently divided among its job unit.

Algorithm:

Step 1: Create the datacenter.

Step 2: Create VMs and allocate them to the host in datacenters according to the space shared schedule or timeshared schedule.

Step 3: Create the cloudlets and allocate to the VMs according to the space shared schedule or timeshared schedule.

Step 4: Create the broker and submit the cloudlet list to it.

Step 5: Start and print the result after the .

Step 6: End

3.3 Basis Of the Proposed Work: Priority Scheduling

The Proposed model has been created by acknowledging different parameters, for example, expense, benefit, client, time, Number of processor solicitation, resource allocated, resource accessibility, asset determination criteria and so on. In Resource designation Model, customers will be clients or clients of cloud which sends administration ask for that is customer sends occupation ask for that is to be executed or run in cloud server[10]. Server in distributed computing environment is the cloud administration supplier which will run the errand or employment presented by customer. The cloud head assumes key part in effective asset portion on the grounds that he chooses the necessity around the distinctive client demand. This necessity based asset assignment acknowledges the parameters examined previously.

Virtualization is an alternate critical subject in distributed computing.[12] It is a registering engineering that empowers a solitary client to get to numerous physical gadgets. An alternate approach to take a gander at it is a solitary machine controlling numerous machines, or one working framework using various workstations to examine a database. With distributed computing, the product programs that are utilized aren't run from your PC, yet rather are Stored on servers housed somewhere else and got to through the Internet. The asset designation demonstrates that chooses necessity around distinctive client appeal is indicated in figure.

Each one appeal comprises of distinctive task. For each one task distinctive parameters are viewed as, for example, time, Processor demand, Priority and so on. Time alludes to calculation time required to finish the specific task, Processor appeal alludes to number of processors required to run the task. More the amount of processor, speedier will be the fulfillment of task. Necessity alludes to the necessity of the task or occupation doled out by the customer.

Based on all the parameters acknowledged above and additionally focused around some limit parameters, necessity calculation chooses necessity around diverse task put together by distinctive clients. The client's task with higher necessity will be given first opportunity to run. The client's task with next higher necessity will be given additional opportunity etc. The tasks which surpass edge will be prematurely ended. Cloud administrator can likewise weigh the status to know which the running tasks are and which are in queue. Along these lines by utilizing necessity calculation, cloud chairman can effectively designate the assets around the clients with least wastage and gives greatest benefit.

In a distributed nature, different clients are submitting employment demand with conceivable obligations that is numerous clients are asking for same asset. Case in point in an elite computational environment which mostly manages experimental reenactments, for example, climate forecast, precipitation simulation, storm expectation and tornado reenactment and so forth which oblige colossal measure of figuring assets, for example, processors, servers, stockpiling and so on. Numerous clients are asking for these computational assets to run their model which is utilized for investigative expectations. So at this circumstance it will be issue for cloud director to choose how to apportion the accessible assets around the asked for clients.

The proposed necessity calculation helps cloud administrator to choose necessity around the clients and assign assets productively as per necessity. This asset portion system is more productive than matrix and utility figuring in light of the fact that in those frameworks there is no necessity around the client ask for and cloud manager is arbitrarily taking choice and he is offering necessity to those client who have submitted their occupation first that is focused around first started things out serve technique.[13] In any case with the appearance of distributed computing and by utilizing this actualized necessity calculation, the cloud administrator can undoubtedly take choice focused around distinctive parameters examined prior to choose necessity around diverse client ask for so administrator can effectively dispense the accessible assets and with expense-adequacy and also fulfillment from clients.

Proposed Algorithm(Pseudo Code):

- For each cloudlet that arrives at the virtual machine consider the following parameters associated with the cloudlet.
- Estimated time of running of the cloudlet(e).

  e=(length of the cloudlet/ mips)

  where mips =millions of instructions per second.
- Number of processors required(c).
- Priority(p).

- Consider a weightage factor 'W' and a base value . The base value for every parameter is calculated based on the average values of the parameters of the cloudlets.
- Compute Base estimated time value(B) as $(\Sigma ti)/n$,

  where ti represents estimated time of job.
- Compute Base priority value(P) as$(\Sigma pi)/n$, where pi represents priority set by the user.
- Compute Base processor value(C) as $(\Sigma ci)/n$, where ci represents no. of processors for the job.
- For each cloudlet do:

  If ti> B

  Compute w1 as B/2

  else

  Compute w1 as B

  If pi> P

  Compute w2 as P/2

  else

  Compute w2 as P

  If ci> C

  Compute w3 as C

else

Compute w3 as C/2

- We use three normalizing constants a, b and c to compute the final weightage(W) of each cloudlet.
- Compute Wi as aw1+bw2+cw3 .
- Sort the Wi values in descending order.
- Allocate the cloudlets to the virtual machines on the basis of higher weightage values.

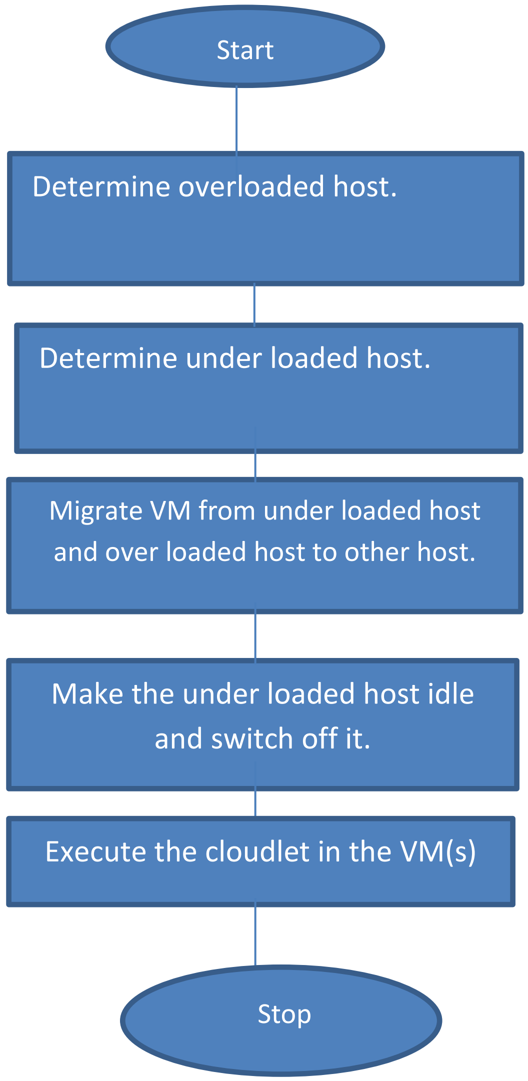
3.4 Reducing Energy Consumption

In our proposed calculation we discovered the host are not used legitimately. A portion of the host are underutilized and some are over used. Such data centers expend gigantic measures of vitality bringing about high working expenses. So we attempted to utilize Dynamic combination of virtual machines (VMs) which is a successful approach to enhance the use of assets and vitality productivity in cloud data centers.[15] Deciding when it is best to reallocate VMs from an overloaded host is a part of element VM combination that straightforwardly impacts the asset use. Dynamic solidification of virtual machines (VMs) utilizing movement of VM and exchanging unmoving hubs to the slumber mode permit Cloud suppliers to advance asset utilization and diminish vitality utilization.

An large-scale data center comprising of N heterogeneous physical hubs. Every hub i is portrayed by the CPU execution characterized in Millions Instructions per Second (MIPS), measure of RAM and system data transmission. Various autonomous clients submit demands for provisioning of M heterogeneous VMs described by necessities to transforming force characterized in MIPS, measure of RAM and system transfer speed. In our model, physical servers are furnished with multi-center Cpus. We demonstrate a multi-center CPU with n centers each one having m MIPS as a solitary-center CPU with the aggregate limit of nm MIPS. These are defended as requisitions, and additionally VMs, are not secured to transforming centers and could be executed on a self-assertive center utilizing a period-imparted booking calculation. The main restriction is that the CPU limit needed for a VM must be less or equivalent to the limit of a solitary center. The reason is that if the CPU limit needed for a VM higher than the limit of a solitary center, then a VM must be executed on more than one center in parallel. Then again, we don't expect that VMs don't run parallel, it is a complex issue.

Dynamic VM merging into four parts: (1) deciding when a host is acknowledged as being overloaded obliging relocation of one or more VMs from this host;(2) deciding when a host is recognized as being under loaded prompting a choice to move all VMs from this host and switch the host to the slumber mode; (3) determination of VMs that ought to be moved from an overloaded host; and (4) discovering another position of the VMs chose for movement from the overloaded and under loaded hosts.[15] In the first place, the looks through the rundown of hosts and by applying the overloading discovery calculation checks whether a host is overloaded. On the off chance that the host is overloaded the VM choice strategy is connected to select VMs that need to be moved from the host. When the rundown of VMs to be moved from the overloaded hosts is fabricated, then new situation for the VMs is to be observed where it is going to relocate. At that point find under loaded hosts and a situation of the VMs from these hosts making the host sit still and exchanging the node off.

Flow Chart


Start
Determine overloaded host.
Determine under loaded host.
Migrate VM from under loaded host and over loaded host to other host.
Make the under loaded host idle and switch off it.
Execute the cloudlet in the VM(s)
Stop

Chapter 4

Simulation & Results

4.1 Time Shared & Space Shared Scheduling

As we already know a two level scheduling is used: first one is for allocating VM's to the host in the datacenter and the second is for allocating the cloudlets to the virtual machines. In time shared scheduling processing power is concurrently shared by the VMs or task depending on which time scheduling policy is applied. In space shared scheduling the processing power is allocated to VMs or task in a sequential way. Here are the results of the simulation.

- Fig 4.1 & Fig 4.2 are results showing space shared scheduling in both the levels.
- Fig 4.3 & Fig 4.4 are results showing time shared scheduling in both the levels.
- Fig 4.5 & Fig 4.6 are results showing space shared scheduling in allocating VMs and time shared for allocating cloudlets.

The no. of datacenters, VMs and the no. of cloudlets are taken as input. The cloudlets are received in the VMs according to Poisson Distribution which is inbuilt in CloudSim.

4.2 Cloud Priority Scheduling

As we have already discuss the proposed algorithm. Here are the result and simulation of the result

- Fig 4.7 shows the time taken by cloudlets to complete the execution in proposed algorithm.
- Fig 4.8 is graph showing no. of cloudlets in a datacenter over time according to proposed algorithm scheduling in VM.

4.3 Analysis of Reduction in Energy Consumption

On implementing the Flowchart in 3.4, we get the following graph

- Fig 4.9 is graph showing the Energy consumption comparison in accordance to the proposed algorithm and after migration in VM.

- Fig 4.10 is graph showing Utilization comparison initially and after migration.

```
Initialising...
6
18
Starting CloudSim version 3.0
Datacenter_2 is starting...
Datacenter_3 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_2
0.0: Broker: Trying to Create VM #1 in Datacenter_2
0.0: Broker: Trying to Create VM #2 in Datacenter_2
0.0: Broker: Trying to Create VM #3 in Datacenter_2
0.0: Broker: Trying to Create VM #4 in Datacenter_2
0.0: Broker: Trying to Create VM #5 in Datacenter_2
[VmScheduler.vmCreate] Allocation of VM #5 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #5 to Host #1 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #0
0.1: Broker: VM #4 has been created in Datacenter #2, Host #1
0.1: Broker: Creation of VM #5 failed in Datacenter #2
0.1: Broker: Trying to Create VM #5 in Datacenter_3
0.2: Broker: VM #5 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
0.2: Broker: Sending cloudlet 2 to VM #2
0.2: Broker: Sending cloudlet 3 to VM #3
0.2: Broker: Sending cloudlet 4 to VM #4
0.2: Broker: Sending cloudlet 5 to VM #5
0.2: Broker: Sending cloudlet 6 to VM #0
0.2: Broker: Sending cloudlet 7 to VM #1
0.2: Broker: Sending cloudlet 8 to VM #2
0.2: Broker: Sending cloudlet 9 to VM #3
0.2: Broker: Sending cloudlet 10 to VM #4
0.2: Broker: Sending cloudlet 11 to VM #5
```

Figure 4.1

```
29.954943949318647: Broker: Cloudlet 12 received
29.954943949318647: Broker: All Cloudlets executed. Finishing...
29.954943949318647: Broker: Destroying VM #0
29.954943949318647: Broker: Destroying VM #1
29.954943949318647: Broker: Destroying VM #2
29.954943949318647: Broker: Destroying VM #3
29.954943949318647: Broker: Destroying VM #4
29.954943949318647: Broker: Destroying VM #5
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_2 is shutting down...
Datacenter_3 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time
    5         SUCCESS        3               5           9.43      0.2           9.63
    4         SUCCESS        2               4           9.52      0.2           9.72
    3         SUCCESS        2               3           9.63      0.2           9.83
    2         SUCCESS        2               2           9.74      0.2           9.94
    1         SUCCESS        2               1           9.85      0.2           10.05
    0         SUCCESS        2               0           9.96      0.2           10.16
    11         SUCCESS        3               5           9.43      9.63          19.07
    10         SUCCESS        2               4           9.52      9.72          19.24
    9         SUCCESS        2               3           9.61      9.83          19.44
    8         SUCCESS        2               2           9.7      9.94          19.65
    7         SUCCESS        2               1           9.8      10.05          19.85
    6         SUCCESS        2               0           9.9      10.16          20.06
    17         SUCCESS        3               5           9.43      19.07          28.5
    16         SUCCESS        2               4           9.51      19.24          28.75
    15         SUCCESS        2               3           9.61      19.44          29.05
    14         SUCCESS        2               2           9.71      19.65          29.35
    13         SUCCESS        2               1           9.8      19.85          29.65
    12         SUCCESS        2               0           9.9      20.06          29.95
```

Figure 4.2

```
Initialising...
6
18
Starting CloudSim version 3.0
Datacenter_2 is starting...
Datacenter_3 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_2
0.0: Broker: Trying to Create VM #1 in Datacenter_2
0.0: Broker: Trying to Create VM #2 in Datacenter_2
0.0: Broker: Trying to Create VM #3 in Datacenter_2
0.0: Broker: Trying to Create VM #4 in Datacenter_2
0.0: Broker: Trying to Create VM #5 in Datacenter_2
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #0
0.1: Broker: VM #4 has been created in Datacenter #2, Host #1
0.1: Broker: VM #5 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #2
0.1: Broker: Sending cloudlet 3 to VM #3
0.1: Broker: Sending cloudlet 4 to VM #4
0.1: Broker: Sending cloudlet 5 to VM #5
0.1: Broker: Sending cloudlet 6 to VM #0
0.1: Broker: Sending cloudlet 7 to VM #1
0.1: Broker: Sending cloudlet 8 to VM #2
0.1: Broker: Sending cloudlet 9 to VM #3
0.1: Broker: Sending cloudlet 10 to VM #4
0.1: Broker: Sending cloudlet 11 to VM #5
0.1: Broker: Sending cloudlet 12 to VM #0
0.1: Broker: Sending cloudlet 13 to VM #1
0.1: Broker: Sending cloudlet 14 to VM #2
0.1: Broker: Sending cloudlet 15 to VM #3
```

Figure 4.3

```
29.801773298239787: Broker: Destroying VM #0
29.801773298239787: Broker: Destroying VM #1
29.801773298239787: Broker: Destroying VM #2
29.801773298239787: Broker: Destroying VM #3
29.801773298239787: Broker: Destroying VM #4
29.801773298239787: Broker: Destroying VM #5
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_2 is shutting down...
Datacenter_3 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    5          SUCCESS      2              5        28.27    0.1           28.37
    11         SUCCESS      2              5        28.27    0.1           28.37
    17         SUCCESS      2              5        28.27    0.1           28.37
    4          SUCCESS      2              4        28.56    0.1           28.66
    10         SUCCESS      2              4        28.56    0.1           28.66
    16         SUCCESS      2              4        28.56    0.1           28.66
    3          SUCCESS      2              3        28.82    0.1           28.92
    9          SUCCESS      2              3        28.82    0.1           28.92
    15         SUCCESS      2              3        28.82    0.1           28.92
    2          SUCCESS      2              2        29.11    0.1           29.21
    8          SUCCESS      2              2        29.11    0.1           29.21
    14         SUCCESS      2              2        29.11    0.1           29.21
    1          SUCCESS      2              1        29.4     0.1           29.5
    7          SUCCESS      2              1        29.4     0.1           29.5
    13         SUCCESS      2              1        29.4     0.1           29.5
    0          SUCCESS      2              0        29.7     0.1           29.8
    6          SUCCESS      2              0        29.7     0.1           29.8
    12         SUCCESS      2              0        29.7     0.1           29.8
 finished!
```

Figure 4.4

```
Initialising...
15
30
Starting CloudSim version 3.0
Datacenter_2 is starting...
Datacenter_3 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_2
0.0: Broker: Trying to Create VM #1 in Datacenter_2
0.0: Broker: Trying to Create VM #2 in Datacenter_2
0.0: Broker: Trying to Create VM #3 in Datacenter_2
0.0: Broker: Trying to Create VM #4 in Datacenter_2
0.0: Broker: Trying to Create VM #5 in Datacenter_2
0.0: Broker: Trying to Create VM #6 in Datacenter_2
0.0: Broker: Trying to Create VM #7 in Datacenter_2
0.0: Broker: Trying to Create VM #8 in Datacenter_2
0.0: Broker: Trying to Create VM #9 in Datacenter_2
0.0: Broker: Trying to Create VM #10 in Datacenter_2
0.0: Broker: Trying to Create VM #11 in Datacenter_2
0.0: Broker: Trying to Create VM #12 in Datacenter_2
0.0: Broker: Trying to Create VM #13 in Datacenter_2
0.0: Broker: Trying to Create VM #14 in Datacenter_2
[VmScheduler.vmCreate] Allocation of VM #5 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #5 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #6 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #6 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #9 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #9 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #10 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #10 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #11 to Host #0 failed by MIPS
```

Figure 4.5

Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
9	SUCCESS	3	9	27.25	0.2	27.45
19	SUCCESS	3	9	27.25	0.2	27.45
29	SUCCESS	3	9	27.25	0.2	27.45
8	SUCCESS	3	8	27.52	0.2	27.72
18	SUCCESS	3	8	27.52	0.2	27.72
28	SUCCESS	3	8	27.52	0.2	27.72
7	SUCCESS	3	7	27.77	0.2	27.97
17	SUCCESS	3	7	27.77	0.2	27.97
27	SUCCESS	3	7	27.77	0.2	27.97
6	SUCCESS	3	6	28.02	0.2	28.22
16	SUCCESS	3	6	28.02	0.2	28.22
26	SUCCESS	3	6	28.02	0.2	28.22
5	SUCCESS	3	5	28.28	0.2	28.48
15	SUCCESS	3	5	28.28	0.2	28.48
25	SUCCESS	3	5	28.28	0.2	28.48
4	SUCCESS	2	4	28.54	0.2	28.74
14	SUCCESS	2	4	28.54	0.2	28.74
24	SUCCESS	2	4	28.54	0.2	28.74
3	SUCCESS	2	3	28.83	0.2	29.03
13	SUCCESS	2	3	28.83	0.2	29.03
23	SUCCESS	2	3	28.83	0.2	29.03
2	SUCCESS	2	2	29.12	0.2	29.32
12	SUCCESS	2	2	29.12	0.2	29.32
22	SUCCESS	2	2	29.12	0.2	29.32
1	SUCCESS	2	1	29.39	0.2	29.59
11	SUCCESS	2	1	29.39	0.2	29.59
21	SUCCESS	2	1	29.39	0.2	29.59
0	SUCCESS	2	0	29.69	0.2	29.89
10	SUCCESS	2	0	29.69	0.2	29.89
20	SUCCESS	2	0	29.69	0.2	29.89

finished!

Figure 4.6

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	weightage(w)	Time	Start Time	Finish Time
0	SUCCESS	2	0	358	67.99	0.5	68.49
1	SUCCESS	2	1	353	73.99	0.5	74.49
2	SUCCESS	2	2	347	80	0.5	80.5
3	SUCCESS	2	3	342	86	0.5	86.5
4	SUCCESS	2	4	337	92	0.5	92.5
5	SUCCESS	2	5	332	98	0.5	98.5
6	SUCCESS	2	6	327	104	0.5	104.5
7	SUCCESS	3	7	322	110	0.5	110.5
8	SUCCESS	3	8	317	116	0.5	116.5
9	SUCCESS	3	9	313	122	0.5	122.5
10	SUCCESS	3	10	308	128	0.5	128.5
11	SUCCESS	3	11	304	134	0.5	134.5
12	SUCCESS	4	12	300	140	0.5	140.5
13	SUCCESS	4	13	296	146	0.5	146.5
14	SUCCESS	4	14	292	152	0.5	152.5
15	SUCCESS	4	15	289	158	0.5	158.5
16	SUCCESS	5	16	286	164	0.5	164.5
17	SUCCESS	5	17	283	170	0.5	170.5
18	SUCCESS	5	18	281	176	0.5	176.5
19	SUCCESS	6	19	280	182	0.5	182.5
20	SUCCESS	2	0	277	184.36	0.5	184.86
21	SUCCESS	2	1	271	185.67	0.5	186.17
22	SUCCESS	2	2	282	187.69	0.5	188.19
23	SUCCESS	2	3	276	190.29	0.5	190.79
24	SUCCESS	2	4	270	193.33	0.5	193.83
25	SUCCESS	2	5	264	196.75	0.5	197.25
26	SUCCESS	2	6	258	200.47	0.5	200.97
27	SUCCESS	3	7	252	204.44	0.5	204.94
28	SUCCESS	3	8	246	208.63	0.5	209.13
29	SUCCESS	3	9	240	213	0.5	213.5
30	SUCCESS	3	10	234	217.52	0.5	218.02
31	SUCCESS	3	11	228	222.18	0.5	222.68
32	SUCCESS	4	12	222	226.96	0.5	227.46

Figure 4.7

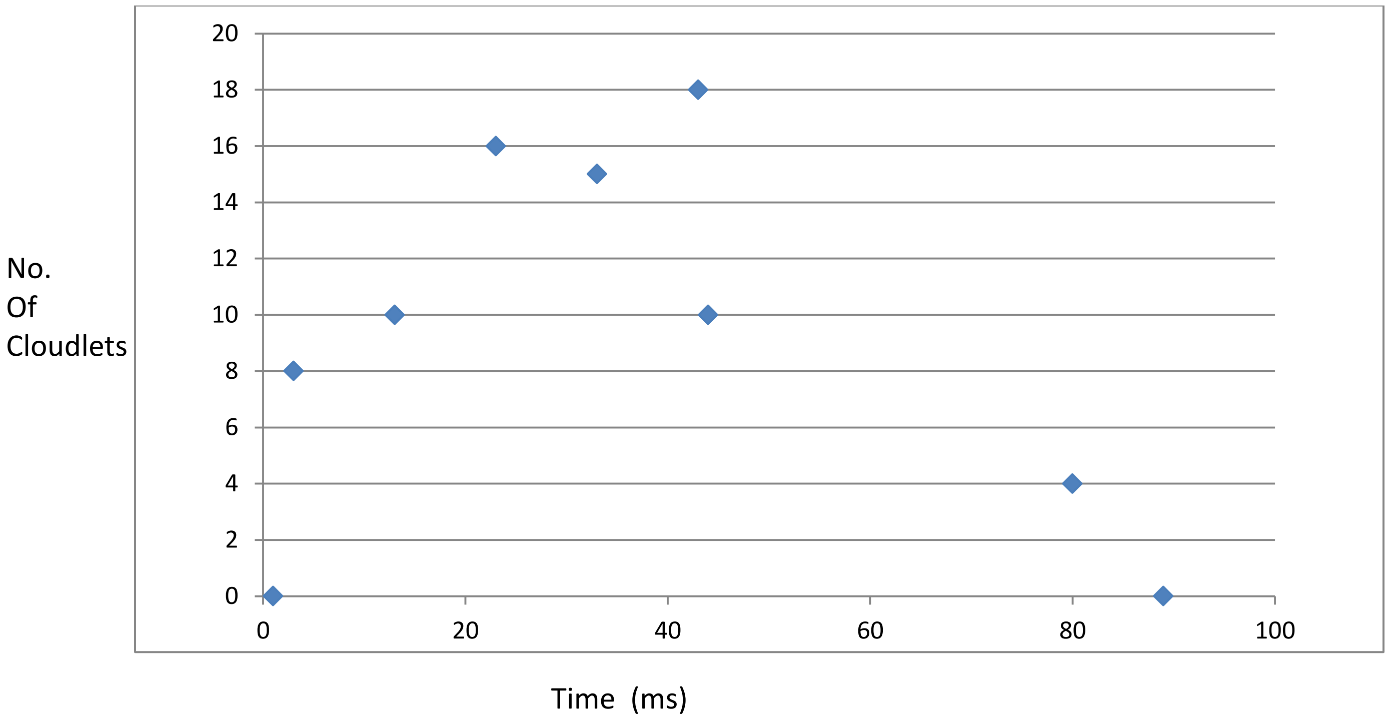
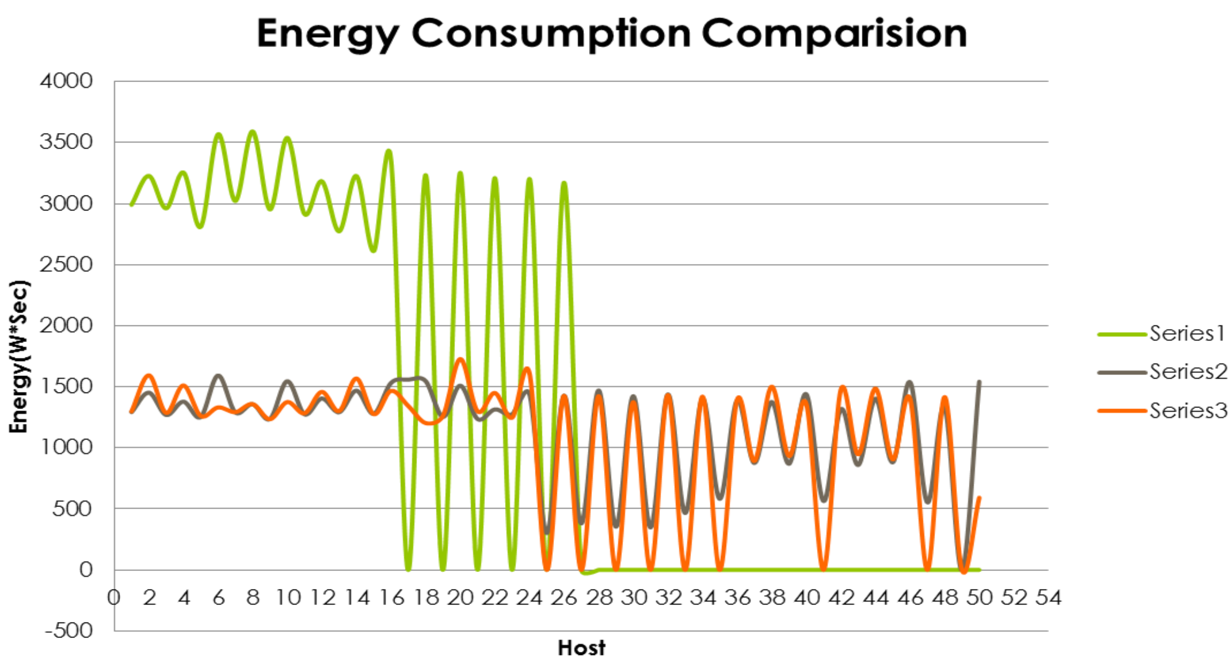


Figure 4.8



Series1 Represent energy consumption according to proposed algorithm.
Series 2 Represent energy consumption when Vm are allocated to host intilally.
Series 3 represent energy consumption after migration of Vm

Figure 4.9

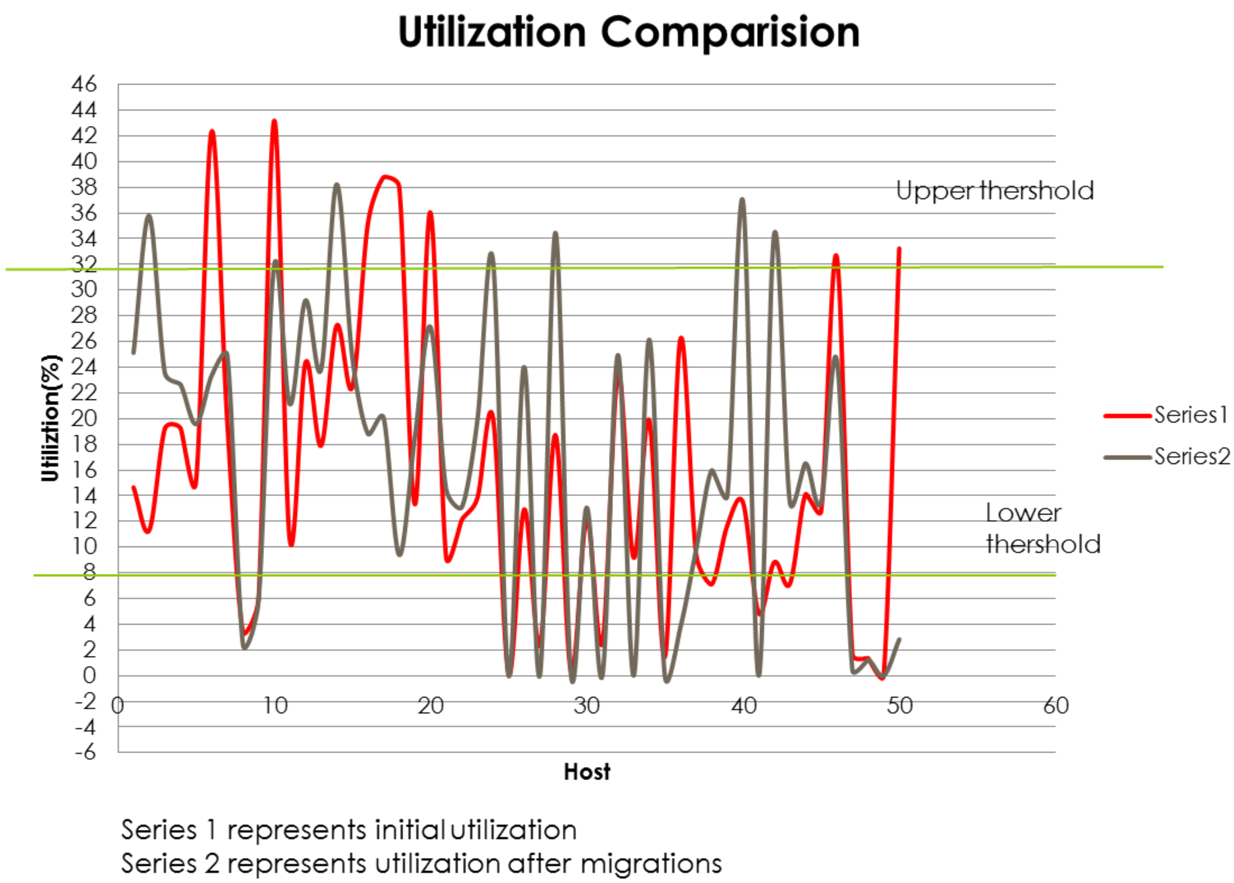

Utilization Comparision
46
44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
-2
-4
-6
0
10
20
30
40
50
60
Utiliztion(%)
Host
Upper thershold
Lower thershold
Series1
Series2
Series 1 represents initial utilization
Series 2 represents utilization after migrations

Figure 4.10

Conclusion & Future Work

This thesis depicts various scheduling policies in the cloud and proposes an efficient priority scheduling algorithm pertaining to certain parameters associated with the job. Using this implemented priority algorithm efficient allocation of available resources along with cost effectiveness can be obtained. Finally cloud admin will decide how much profit he can gain by allocating the available resources and prioritizing among the different user requests. The reduction in energy of over utilized nodes in the cloud system has been achieved by transferring the Virtual Machines to another node. Switching of idle nodes to sleep mode in underutilized nodes has also been done that optimizes resource usage and reduces energy consumption. Future work may include extending the parameters of the priority scheduling algorithm like including cost, software or application to be used, history or previous interaction with the cloud etc that extends the capability of the algorithm. Also work could be done on the selection of Virtual Machines that are to be migrated from the over utilized or underutilized nodes.

Bibliography

[1] K C Gouda, Sambeet Kar, Rohan Patra. “Cloud Computing-What,How and Why.” International Journal of Engineering Trends and Technology,Volume 4 Issue 6,Pg 2453-2457.

[2] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and Rajkumar Buyya. “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.”

[3] Marcus Bohm, Stefanie Leimeister, Christopher Riedl, Helmut Krcmar “Cloud Computing and computing evolution” Technische Universität München (TUM), Germany

[4] Josh Ames. ― “Types of Cloud computing.”

[5] Microsoft Perspective:Issues that matter .“The Three layers of Cloud computing.”

[6] Priya Vishwanathan. “Cloud Computing-Is it really that Beneficial?”

[7] Raul F Chong. “Smarter Cloud Solutions-Changing the World:Big Data and the Cloud.”

[8] Raul F Chong. “Big data analytics on IBM SmartCloud Enterprise.”

[9] University of BRISTOL. ―re:search High Performance Computing Issue.

[10] Gouda, K. C., and Dwaipayan Acharjee. "An Outlook of cloud computing." International Journal of Science, Engineering and Technology Research 2.6 (2013): pp-1403-1406.

[11] Vouk, Mladen A. 2008. “Cloud computing - Issues, research and implementations. Proceedings of the International Conference on Information Technology Interfaces (ITI’08), June 2008, 31-40.”

[12] Zhongyuan Lee, Ying Wang, Wen Zhou. ― “A dynamic priority scheduling algorithm on service request scheduling in cloud computing.” International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011, Harbin, Heilongjiang, China, 12-14 August, 2011.

[13]Luqun Li. "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers" The Fifth Annual China Grid Conference.

[14]Wenying Zeng, Yuelong Zhao, Junwei Zeng. ― “Cloud Service and Service Selection Algorithm Research.”

[15]Anton Beloglazov, Rajkumar Buyya.―“Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers Concurrency Computation.: Pract. Exper. 2012; 24:1397–1420.