

**Improved Scheduling
Algorithm in
Cloud Environment**

Nipun Madan



**Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India
May 2014**

Improved Scheduling Algorithm in Cloud Environment

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

Master of Technology

in

Computer Science and Engineering

(Specialization: Software Engineering)

by

Nipun Madan

(Roll No.- 212CS3373)

under the supervision of

Prof. P.M. Khilar



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, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled *Improved Scheduling Algorithm in Cloud Environment*, by *Nipun Madan (212cs3373)*, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Software Engineering in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: June 2, 2014

(Prof. P.M. Khilar)
Professor, CSE Department
NIT Rourkela, Odisha

Author's Declaration

I hereby declare that all work contained in this report is my own work unless otherwise acknowledged. Also, all of my work has not been submitted for any academic degree. All sources of quoted information has been acknowledged by means of appropriate reference.

Place: NIT Rourkela
Date: June 2, 2014

(**Nipun Madan**)
M.Tech, 212cs3373, CSE Department
NIT Rourkela, Odisha

Acknowledgment

Writing of thesis is a journey through gravel road, but you can make it with the help of the people and resources you get in touch through out your journey. I am thankful to all of them for their hands to frame this thesis.

First of all, I would like to express my unfeigned thanks to **Prof. P.M. Khilar** for his observations and guidance during my thesis work constantly encouraged me to build the direction to the research and to move forward with in depth analysis. Being a source of knowledge, his words make my road become straight and easier to go. I am also grateful to **all the professors** of our department for their advice and support.

I would like to thank **Prof. S.K Rath** for allowing me to sit in his research lab and motivating me for doing good research.

I would like to thank all my friends and lab-mates for their cooperation and pepping up, which can never be penned with words.

One of the most important acknowledgment is for the academic resources that I have got from NIT Rourkela from the administrative and technical staff members who have been kind enough to advise and help in their respective roles.

Despite being mentioned after everyone else, **I would like to lovingly dedicate this thesis to my Mom & Dad** for their love, patience, understanding and to being my source of inspiration.

Nipun Madan

Abstract

Scheduling in cloud computing is a technique which is used to improve the overall execution time of the job. A good scheduling algorithm can help in load balancing as well. Scheduling in cloud can be done in three areas i.e Scheduling Cloudlets within the virtual machine, Scheduling Virtual Machine on the host, and scheduling cloudlets to the Virtual Machine.

The last scheduling i.e scheduling the cloudlets to the virtual machine is implemented in this thesis. In this the cloudlets are intelligently scheduled to the best possible virtual machine so that the overall execution time can be reduced. The outcome proves that the proposed algorithm gives better result in comparison to the already used sequential algorithm.

Keywords: Cloud Computing, Scheduling, Load Balancing, Execution Time Improvement, Algorithm

Contents

Certificate	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
List of Figures	viii
List of Tables	ix
1 Introduction	2
1.1 Thesis Organization	3
2 Basics of Cloud Computing	5
2.1 Introduction	5
2.2 Cloud Computing Architecture	5
2.3 Architectural Components	9
2.4 Service Orchestration	11
3 Literature Review	15
3.1 Review of related work	15
3.2 Motivation	22
4 Proposed Technique & Implementation	25
4.1 Introduction	25
4.2 Proposed Technique	25
4.3 Assumptions	27
4.4 Problem Domain	27
4.5 Proposed Algorithm	27

4.6	Simulation Result	28
5	Conclusion	32
5.1	Achievements	32
5.2	Limitation & Future Scope	32
	Bibliography	33

List of Figures

2.1	Cloud Computing Architecture	6
2.2	Cloud Consumer	6
2.3	Cloud Provider	7
2.4	Actors	8
2.5	Public Cloud	9
2.6	Private Cloud	10
2.7	Hybrid Cloud	11
2.8	Service Orchestration	12
3.1	Cloud Layered Architecture	16
3.2	Load Balancing Algorithm Comparison	18
3.3	Simulation in CloudSim	19
3.4	Scheduling Algorithm Comparison	21
3.5	Scheduling Algorithm Comparison	22
4.1	Simplified Architecture	26
4.2	Result Graph	29
4.3	Result Graph	30

List of Tables

4.1	Comparison between existing and proposed algorithm	28
4.2	Comparison between existing and proposed algorithm	30

Chapter 1

Introduction

Thesis Organization

Chapter 1

Introduction

Nowadays Cloud Computing is the only area on which lots of companies are spending a lot of money because they feel that it is the future of technology. It is still in its early ages and needs a lot of improvement. Apart from big companies like amazon, google, microsoft, SAP etc no one is able to contribute to this field.

Cloud in IT world can be viewed as a bundle of resources which can run any application in this entire world. It provides service on demand and the user has to pay for the amount of time it has used those services. This is also an issue in cloud computing that how to do the pricing. Very Talented IIM pass-outs are working on this problem.

This service can solve many problems such as :-

- Organizations need not to buy the whole set of servers. They can just have services from the service providers.
- It's easy to handle as many responsibilities are given to cloud provider.
- They can expand their work easily as the resources are near to infinity.
- It serves best for the elastic applications (Applications which scale very fastly for ex- facebook)

Scheduling the jobs in cloud is a big challenge nowadays. As if jobs i.e. Cloudlets [1] are scheduled properly then the overall execution time can be reduced significantly. But, Scheduling is an NP Hard problem so it is very important to apply some heuristics to get best outcome many times.

This thesis consist of a method which will improve the overall cloudlet [2] execution time. It is being compared with the existing cloudlet scheduling algorithm used by the CloudSim. CloudSim is the simulation tool which is used in the overall implementation Cloudlet scheduling algorithm.

1.1 Thesis Organization

The rest of the thesis is organized as follows:

The basic informations to understand code obfuscation is given in **Chapter-2**, as Theoretical Background. It contain the definition and classification of Cloud Environment It sets up the base too understand the cloud architecture and several scheduling policies .

Chapter-3 will provide a overview of the related work and research done till now on scheduling in cloud environment. Which will lighten you up with a little deep knowledge about cloud scheduling..

Chapter-4 will take you through our proposed technique on the cloud scheduling algorithm. The results are tested using cloudSim simulator.

In **Chapter-5** the overall work with strength and drawbacks is expressed in the section Conclusion and the possible future work is described in the Future Scope section.

Chapter 2

Theoretical Background of Cloud Computing

Introduction

Cloud Computing Architecture

Architectural Components

Service Orchestration

Chapter 2

Basics of Cloud Computing

2.1 Introduction

Cloud Computing is the future of technology said by various CEO of IT industry. It works on the principle that the user must pay according to the time for which it is using the resources from various cloud providers. It is well known that it is not easy to handle large resources. It is very costly in terms of money as well. So Cloud Computing is the best answer to these challenging problems.

Cloud Computing is an answer to almost every problem faced by many industries which are directly or indirectly related to the Information Technology. This field is growing very fastly as many of the big players in the Information Technology field like Microsoft, Google, Amazon, SAP are investing a lot of money to get improved results. They feel that it is the future of technology.

2.2 Cloud Computing Architecture

The Cloud Computing architecture works on a layered approach. It has several actors like Cloud Consumer, Cloud Provider, Cloud Auditor, Cloud Broker, Cloud Carrier. The combination of all these actors create the architecture of Cloud Computing [3]. It is a very vast area that's why it is not easy to understand how everything is happening but this reference architecture gives a great overview.

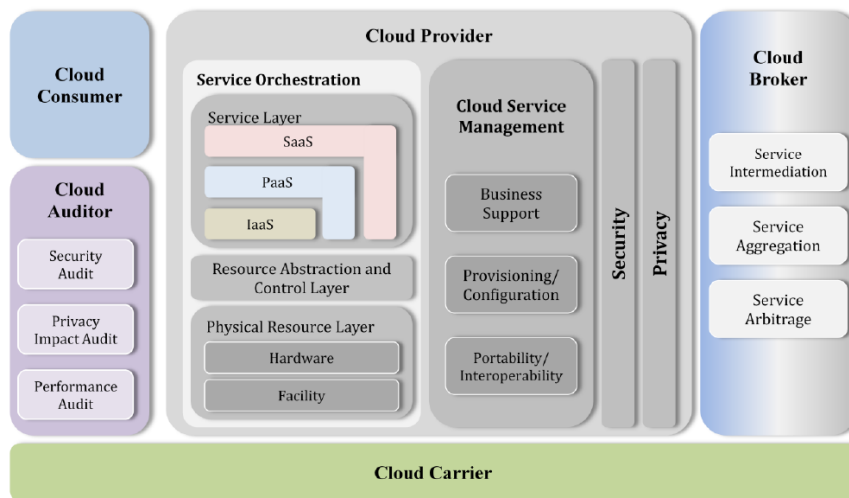


Figure 2.1: Cloud Computing Architecture

Each actor play a very big role in the cloud computing working model. They have to play their part to get the job done and satisfy the customer [1]. The different cloud consumers are defined below:-

Cloud Consumer

Cloud Consumer are the actors which consumes the services provided by the cloud provider. They can directly avail services from the providers or they can contact the cloud broker to get the services. They will pay according to the time for which they have the services from the provider [4]. They can have different services like SaaS, Paas, IaaS.

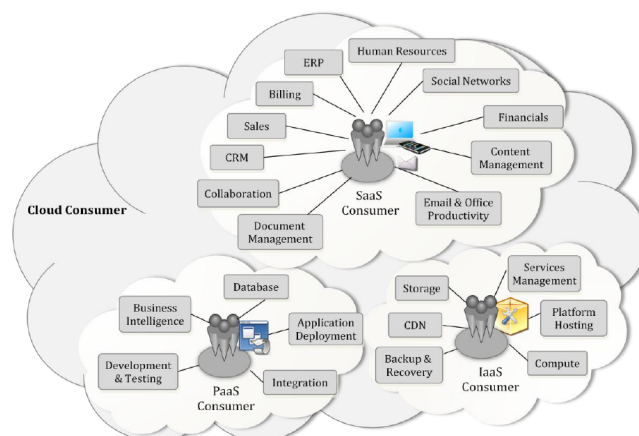


Figure 2.2: Cloud Consumer

Cloud Provider

Cloud Provider are the actors which provides the services to the consumers. The services can be SaaS, PaaS, IaaS. The big players like google, amazon etc are able to provide Infrastructure as a service. Else all other small players are only working on providing SaaS or to the some extent PaaS. The Cloud Provider also charges the money according to the time for which the services are availed by the consumers.

In SaaS service cloud provides the consumers to host the service i.e the software at their end(on their resources). This helps the consumers for not to buy the resources to run the software. All the security aspects are also handled by the providers itself.

In PaaS the provider provides a toolkit to the consumers like different IDE or SDK to compile their software and host them on the providers resources [5]. This helps the consumer to have a fair control over the resources provided by the Cloud Provider.

In IaaS the consumer has the total control over the resources provided by the Cloud Provider [6]. The low level programming can be done in IaaS. Some of the issues like response time, memory usage and Disk Access can easily be addressed using IaaS as a service.

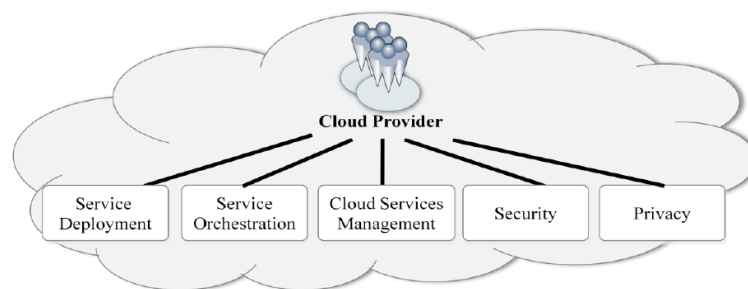


Figure 2.3: Cloud Provider

Cloud Auditor

Cloud Auditor is the actor which takes care of all the pricing and accounting aspects of the whole cloud computing scenario [7]. If any sort of conflict happens

in terms of pricing then it is resolved by Cloud Auditor [8].

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, <i>Cloud Providers</i> .
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between <i>Cloud Providers</i> and <i>Cloud Consumers</i> .
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from <i>Cloud Providers</i> to <i>Cloud Consumers</i> .

Figure 2.4: Actors

Cloud Broker

Cloud Broker is the actor which acts in between the Cloud Consumer and Cloud Provider [9]. It helps the customers to choose the efficient and the best cloud provider. It plays a very vital role in negotiating the deal between the provider and consumer. If any kind of conflict happens then that conflict is resolved by the Broker [10].

Cloud Broker provides services in the following way:-

- **Service Intermediation** It acts as a intermediate in providing services from provider to the consumer.
- **Service Aggregation** It combines the services provided by different Cloud Providers and then delivers those services to the Cloud Consumers.
- **Service Arbitrage** It enhances the services provided by the cloud provider and then provide that as a service to the different Cloud Consumers [11].

Cloud Carrier

Cloud Carrier is just a medium through which the cloud services are provided. It is basically the connection provider for example airtel, MTNL, BSNL, Vodafone etc [12].

2.3 Architectural Components

NIST defines the three architectural components of cloud which are basically defined as follows: public cloud, private cloud, community cloud, or hybrid cloud. The differences are based on how exclusive the computing resources are made to a Cloud Consumer [13].

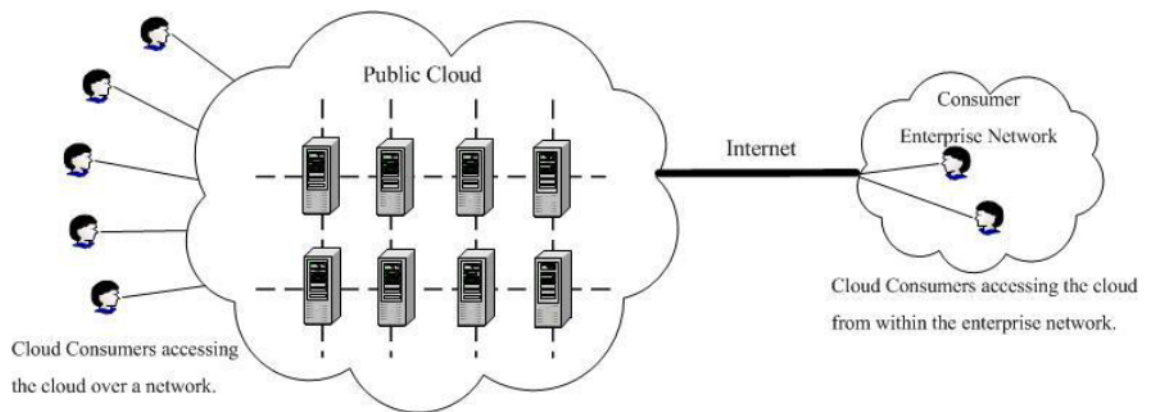


Figure 2.5: Public Cloud

A public cloud is one in which the cloud infrastructure and computing resources are made available to the general public over a public network. A public cloud is owned by an organization selling cloud services, and serves a diverse pool of clients [14].

A public cloud can be accessed from anywhere and that is the specialty of this cloud but are few issues related to the public cloud and one of them is security. It is important to handle the security issue in order to use this service of public cloud properly.

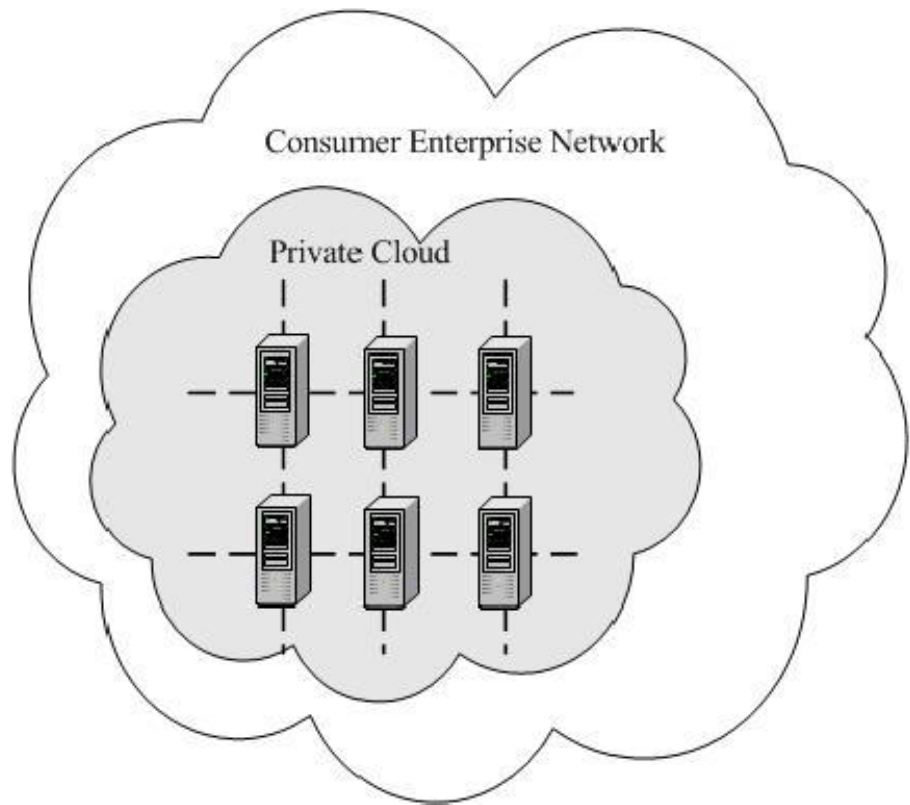


Figure 2.6: Private Cloud

Private cloud is a cloud which can only be accessed within the organization. It is basically used by large organizations in order to handle their business properly. It may be managed either by the Cloud Consumer organization or by a third party, and may be hosted on the organizations premises (i.e. on-site private clouds) or outsourced to a hosting company (i.e. outsourced private clouds) [16]. The fig above present an on-site private cloud and an outsourced private cloud, respectively.

A private cloud can only be accessed by the different by the people within the organization. This type of cloud is created to improve the safety of data in the organization. The private cloud van be created by the organization or the the third party can also create the private cloud for a particular organization [17].

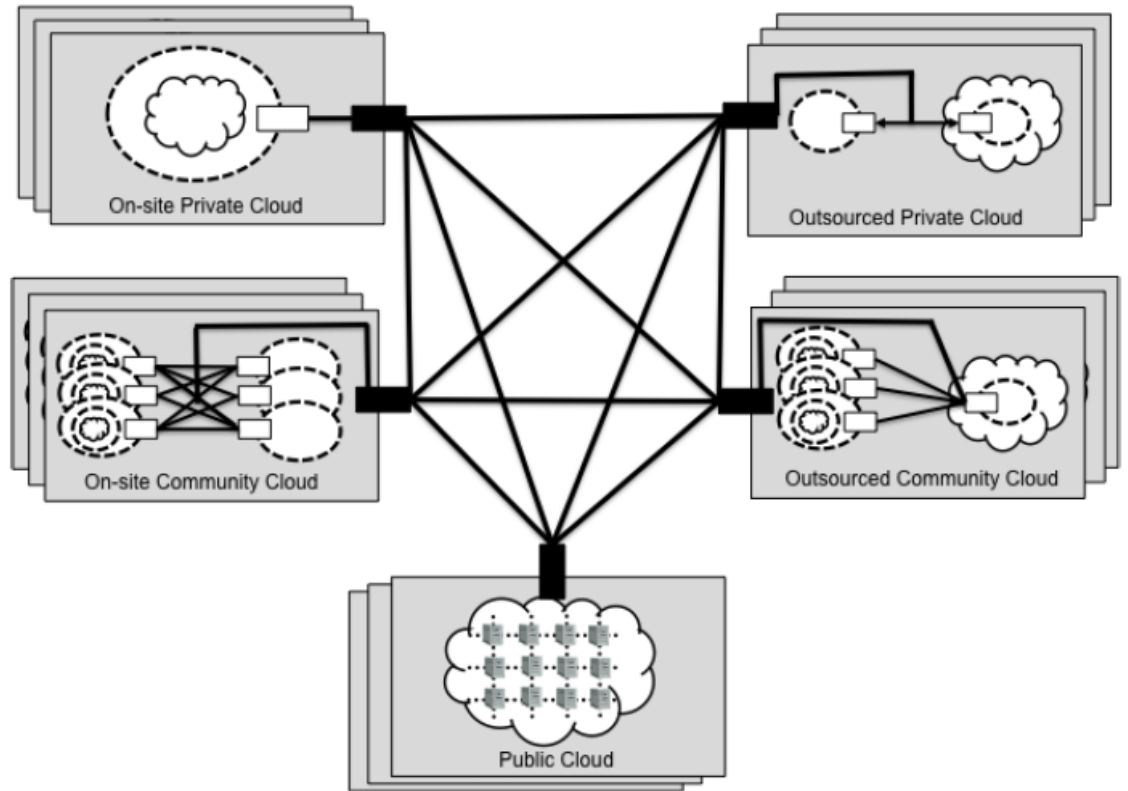


Figure 2.7: Hybrid Cloud

A hybrid cloud is a composition of two or more clouds (on-site private, on-site community, off-site private, off-site community or public) that remain as distinct entities but are bound together by standardized or proprietary technology that enables data and application portability [18]. Figure 14 presents a simple view of a hybrid cloud that could be built with a set of clouds in the five deployment model variants [19].

2.4 Service Orchestration

Service Orchestration refers to the composition of system components to support the Cloud Providers activities in arrangement, coordination and management of computing resources in order to provide cloud services to Cloud Consumers [20]. Figure below shows a generic stack diagram of this composition that underlies the provisioning of cloud services.

A three-layered model is used in this representation, representing the grouping

of three types of system components Cloud Providers need to compose to deliver their services.

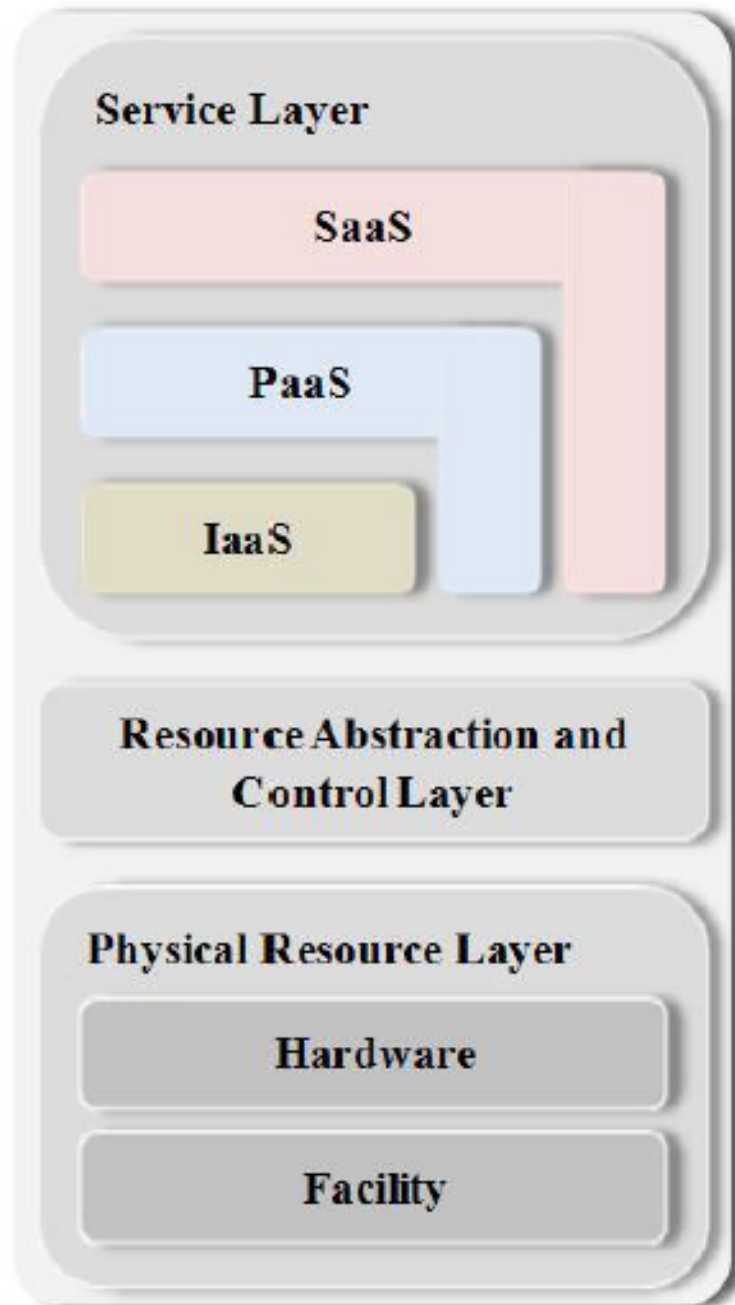


Figure 2.8: Service Orchestration

In the model shown in FigureService Orchestrationthe top is the service layer, this is where Cloud Providers define interfaces for Cloud Consumers to access the computing services. Access interfaces of each of the three service models are

provided in this layer. It is possible, though not necessary, that SaaS applications can be built on top of PaaS components and PaaS components can be built on top of IaaS components. The optional dependency relationships among SaaS, PaaS, and IaaS components are represented graphically as components stacking on each other; while the angling of the components represents that each of the service component can stand by itself [21]. For example, a SaaS application can be implemented and hosted on virtual machines from an IaaS cloud or it can be implemented directly on top of cloud resources without using IaaS virtual machines.

Chapter 3

Literature Review

Review of related work

Motivation

Chapter 3

Literature Review

Cloud Computing is still a new area of research so not much has been done in this field upto now. But still i have reviewed some of the research papers and books to gain some background and improve the already developed scheduling algorithms till now. This field is growing very fastly in the recent past so the amount of research going on is intensive.

3.1 Review of related work

CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms - 2011

In this paper [22] authors described almost everything which is required to develop and test any cloud related policy. The authors gave a brief knowledge about the tool cloudSim and basic knowledge of CloudComputing. They defined that cloudSim supports almost all the functionality required to define the policies in Cloud Environment.

According to them cloud computing is a parallel and distributed system which consists of a collection of inter-connected and virtualized computer that are dynamically provisioned. Some of the cloud computing platforms are Microsoft Azure, Amazon EC2, Aneka, Google App. It also defined the layered design architecture of cloud computing.

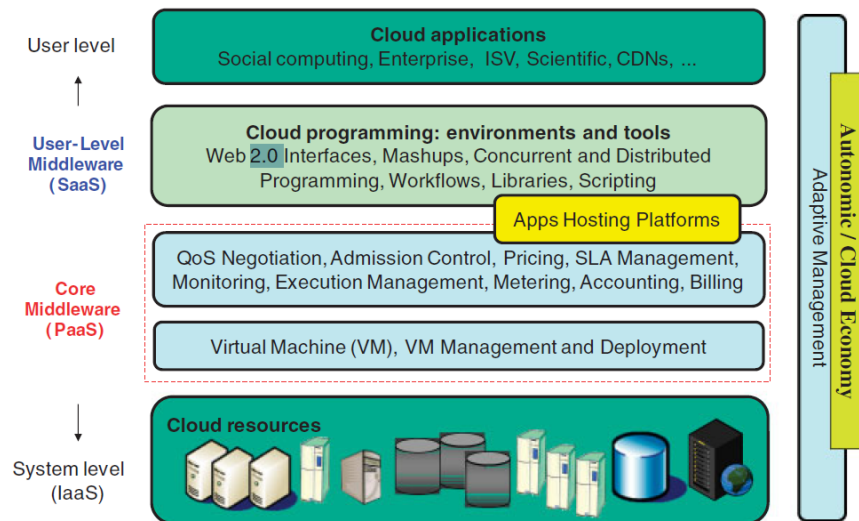


Figure 3.1: Cloud Layered Architecture

In this Figure Cloud Architecture the layered architecture of cloud is defined. It is divided into four layers user level layer, user level middleware layer, core middleware layer and system layer. Each layer is independent of each other and only dependent upon the inputs from the upper layer and gives the output to the lower layer. The User Level consist of the cloud application GUI. It will run on the client side where the inputs will be taken and then forwarded to the next level. The Core Middleware consist of the system of some System Software like NetBeans, Eclipse or some sort of compiler. This layer helps the client to write down there own software using the system softwares and then deploy them on the cloud provider side. When these software are deployed on the provider side the overall performance improvement doesn't happen. To have a control over the performance the System Level control is required. In this layer the user has the control over the overall resources of the provider. Because of this the users can make as many changes as they want. Because they are having the low level control over the resources of the provider. Due to this low level control the overall improvement can be done very easily.

So as the conclusion the above helped me gain the complete overview of the Cloud Computing Environment. This overview helped me to understand the other factors about the cloud computing. This paper also built the strong foundation

about the basics of cloud computing. Cloud is a new technology so fast improvements are expected therefore the terms and the way it is used will be changed in future very quickly.

A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms-2012

The researchers of this paper [23] did a survey on the already developed and defined load balancing algorithms. They gave a detailed overview of the already defined load balancing algorithms. As we know that load balancing and scheduling are somehow related in the field of cloud computing therefore i am defining this paper as a part of my thesis in the Literature Overview section. According to the authors of this paper the load balancing algorithms are of two types i.e static load balancing algorithm and dynamic load balancing algorithm. The static load balancing algorithms are those which do not modify or change states at runtime but dynamic load balancing algorithms definitely do that at run time. As we all know that the in Cloud Environment new resources keep on adding and the overall structure of the cloud keep on changing therefore static load balancing algorithms do not play a big role. To cope up with the changing cloud environment the dyanamic load balancing algorithms play a big role.

This paper also defined the challenges behind the load balancing in cloud environment. Some of the main reasons are:-

- **Spatial Distribution of the Cloud Nodes** As we know that cloud is a distributed system therefore the cloud nodes are spatially distributed over the geographical locations therefore it becomes a great challenge for the designers of load balancing algorithms to design the algorithms keeping in mind the geographical location of the nodes. The Fault Tolerance has to be taken into account.
- **Storage/ Replication** To improve the fault tolerance of the system data replication is the way we to go. It is not easy to manage such a huge amount of data. This huge amount of data storing is not easy and fetching it also very

difficult. Therefore, it is very important to manage this huge amount of data properly to provide adequate amount of service.

- **Algorithm Complexity** If the complexity of the algorithm is more than the overall turnaround time will definitely increase. Therefore it is very important for the load balancing algorithm designers to do a proper trade-off between the algorithm complexity and the overall complexity of the application.
- **Point of Failure** If Cloud is using the centralized approach then one point of failure becomes a big challenge therefore the solution to this problem is that the architecture of cloud must become decentralized.

The comparison of several load balancing algorithms is given below

	<i>Pros</i>	<i>Cons</i>
INS	<ul style="list-style-type: none"> •Initially proved to handle some sort of dynamic load balancing 	<ul style="list-style-type: none"> •No forecasting algorithm to identify the future behavior of the nodes. •Complicated in terms of implementation. •Only certain parameters are considered such as distance and time
ESWLC	<ul style="list-style-type: none"> •More accurate results than WLC 	<ul style="list-style-type: none"> •Complicated •Prediction algorithm requires existing data and has long processing time
CLDBM	<ul style="list-style-type: none"> •Solves issues of Round Robin Algorithm •Automated tasks forwarding reduces the need for a human administrator 	<ul style="list-style-type: none"> •Inherits Round Robin issues such as not taking into consideration node Capabilities •Single point of failure (if CLBDM fails, the whole process fails) •The threshold might not be applied to all cases.
ANT COLONY	<ul style="list-style-type: none"> •Best case scenario is that the underloaded node is found at beginning of the search •Decentralized, no single point of failure •Ants can collect the information faster 	<ul style="list-style-type: none"> •Network overhead because of the large number of ants •Points of initiation of ants and number of ants are not clear •Nodes status change after ants visits to them is not taken into account •Only availability of node is being considered, while there are other factors that should be taken into consideration
Enhanced MapReduce	<ul style="list-style-type: none"> •Less overhead for the reduce tasks 	<ul style="list-style-type: none"> •High processing time •Reduce tasks capabilities are not taken into consideration
VM Mapping	<ul style="list-style-type: none"> •Reliable calculation method 	<ul style="list-style-type: none"> •Single Point of failure •Does not take into account network load, and node capabilities
DDFTP	<ul style="list-style-type: none"> •Fast •Reliable download of files 	<ul style="list-style-type: none"> •Full replication of data files that requires high storage in all nodes.
LBMM	<ul style="list-style-type: none"> •Reliable tasks assignment to nodes 	<ul style="list-style-type: none"> •Slower than other algorithms because Work must pass through three layers to be processed.

Figure 3.2: Load Balancing Algorithm Comparison

The above FigureLoad Balancing Algorithm shows a good comparison between the several load balancing algorithms.

DARTCSIM: AN ENHANCED USER-FRIENDLY CLOUD SIMULATION SYSTEM BASED ON CLOUDSIM WITH BETTER PERFORMANCE-2012

Here [3] the researchers also give a new simulator which has improved features in comparison to the cloudSim. In this paper the researchers worked on improving the already available cloud simulator cloudSim. They diminished some of its drawbacks by introducing some new methods.

The way in which simulation happens in cloudSim is shown below

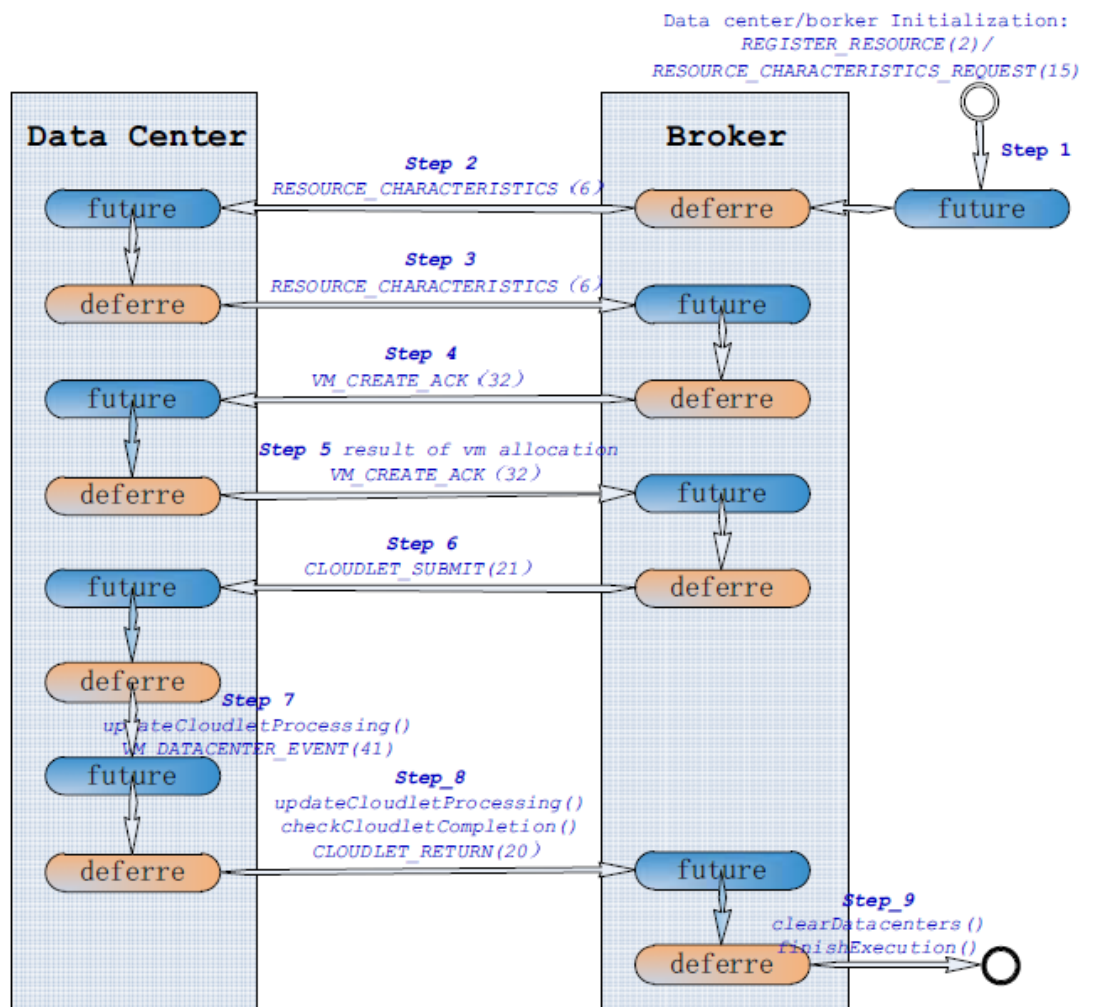


Figure 3.3: Simulation in CloudSim

The dataflow in cloudSim happens in the way it is shown above. The re-

searchers of this paper found some problems in the way the data is flowing in cloudSim and they worked upon that to show some sort of improvement in that.

In the end the problem with cloudSim is that it does not provide any sort of graphical user interface and the researchers need to have good command over the JAVA programming language in order to work upon that. The DartcSim overcomes this shortcoming and created a good GUI for the researchers to work on but the low level control goes away. If you want to do low level work then you must go for cloudSim.

Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment-2009

In this paper [24] the authors introduced a new type of scheduling algorithm which is known as meta scheduling algorithm. In this paper they divided the jobs into four different categories i.e short narrow, short wide, long narrow and long wide. With this division they set the priority more to the shorter jobs in such a way that they do not have to compromise with the longest end time of the larger job.

If we see in a different way this works on the principle of slag time. They will slag the longest job to the maximum extent and then they will delay it's processing accordingly. It is very important that starvation does not and that is the main goal of the algorithm therefore they do not delay the job more than there slag time.

A SURVEY OF VARIOUS SCHEDULING ALGORITHM IN CLOUD COMPUTING ENVIRONMENT-2013

This research work [25] the researchers surveyed the already developed and used scheduling algorithms. As we know that cloud computing uses the already defined and used protocols and algorithms but in a entirely different way which indeed a different architecture therefore many of these scheduling algorithms which are used here are already being used in operation system.

Scheduling Algorithm	Scheduling Method	Scheduling Parameter	Scheduling Factor	Findings	Environment
Resource-Aware-Scheduling algorithm (RASA)	Batch Mode	Make Span	Grouped task	1. It is used to reduce makespan	Grid environment
RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)	Batch Mode	processing time	Grouped task	1. It is used to reduce processing time. 2. It is efficient for load balancing.	Cloud environment
An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing	Batch Mode	Quality of Service, Service request time	An array of workflow instances	1. High QoS 2.High throughput	Cloud environment

Figure 3.4: Scheduling Algorithm Comparison

The above figure shows the difference among the several scheduling algorithms. It distinguishes them on the basis of scheduling algorithm, scheduling method, scheduling parameter, scheduling factor, findings and the environment.

As we know that cloud computing uses the already defined and used protocols and algorithms but in an entirely different way which indeed is a different architecture therefore many of these scheduling algorithms which are used here are already being used in operation system.

According to the authors of this paper the scheduling algorithms are of two types i.e static scheduling algorithm and dynamic scheduling algorithm. The static scheduling algorithms are those which do not modify or change states at runtime but scheduling algorithms definitely do that at run time. As we all know that in Cloud Environment new resources keep on adding and the overall structure of the cloud keep on changing therefore static load balancing algorithms do not play a big role. To cope up with the changing cloud environment the static scheduling algorithms play a big role.

Some more comparisons are shown in the figure below

Environment					
A Priority based Job Scheduling Algorithm in Cloud Computing	Dependency mode	Priority to each queue	An array of job queue	1. Less finish time	Cloud environment
Extended Max-Min Scheduling Using Petri Net and Load Balancing	Batch Mode	Load balancing, finish time	Grouped Task	1. It is used for efficient load balancing. 2. Petrin net is used to remove limitation of max-min algorithm.	Cloud environment
An Optimistic Differentiated Job Scheduling System for Cloud Computing	Dependency mode	Quality of service, Maximum profit	Single Job with multiple user	1The Qos requirements of the cloud computing user and the maximum profits of the cloud computing service provider are achieved.	Cloud environment
Improved Cost-Based Algorithm for Task Scheduling	Batch mode	Cost, Performance	Unscheduled task group	1.Measures both resource cost and computation performance 2. Improves the computation /communication ratio	Cloud environment
Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling	Batch mode	Performance, Cost,	Workflow with large number of job	1. The application of migrations and starvation handling had a significant effect on the model. 2. It improves performance.	Cloud environment

Figure 3.5: Scheduling Algorithm Comparison

The above figure shows the difference among the several scheduling algorithms. It distinguishes them on the basis of scheduling algorithm, scheduling method, scheduling parameter, scheduling factor, findings and the environment.

If we closely notice the above defined scheduling we came across a conclusion that every algorithm performs better in a specified environment.

3.2 Motivation

As we have seen above that many of the scheduling algorithms are working on scheduling the cloudlets to the computational cores of datacenter or some of them are working on scheduling the virtual machine within the computational cores of

the datacenter.

I as a researcher exploited the another area where i concentrated on distributing/scheduling the cloudlets to several Virtual Machines which are available. based on this i implemented my a new algorithm which schedules the cloudlets to the appropriate virtual machine so that the overall turnaround time of the cloudlet is reduced.

Chapter 4

Proposed Technique & Implementation

Introduction

Assumption

Proposed Technique

Implementation

Chapter 4

Proposed Technique & Implementation

4.1 Introduction

The main goal for many algorithms is to develop algorithms which indeed decrease the execution time of the job in hand. My proposed technique also does that in the field of cloud computing by giving an improved cloudlet scheduling algorithm.

4.2 Proposed Technique

As we have already seen the architecture of cloud computing in the above chapters so from we can see that there are three areas where the scheduling is required and those three areas are as follows

- **Cloudlet in VM** The cloudlets submitted to the VM need to be scheduled. So, there is a need of scheduling in order to improve some of the time parameters like response time, turnaround time etc.
- **VM in Datacenter** As we know that in cloud the virtualization takes place. Therefore many virtual machine runs on several computational cores. So scheduling those virtual machines is also a tedious task. So we can introduce a good scheduling there.
- **Scheduling the task to VM** The other area is that we can intelligently submit the task to the virtual machine so that the parameters like response

time, turnaround time etc can be improved.

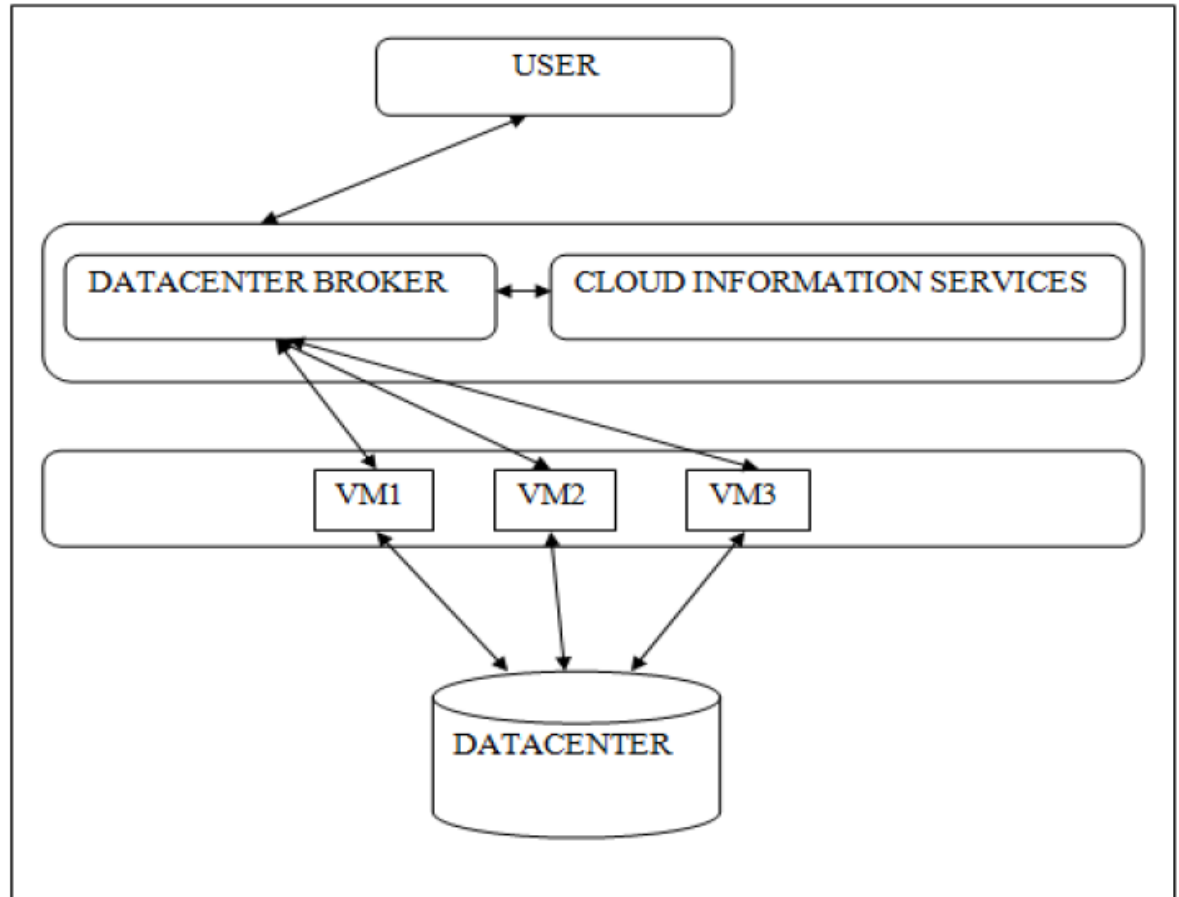


Figure 4.1: Simplified Architecture

The above shows how the main area where the scheduling in the cloud happens. If we observe the above figure closely we can see that hierarchy in which the things are placed is

- Datacenter
- Processing Cores
- Virtual Machines

The datacenter is a place where all the processing elements are placed. While processing cores are the real resources on which Virtual Machines are running.

4.3 Assumptions

The Assumptions behind my proposed techniques are

- All cloudlets which have to be processed are available.
- At runtime no more cloudlets are added.
- The environment is also static i.e no more resources are added at runtime.

4.4 Problem Domain

The problem is to schedule or map the cloudlets to the Virtual Machine so that the overall turnaround can be least.

- Let there be set of cloudlets(jobs)
 $C=t_1,t_2,t_3,\dots$
- Let there be a set of Virtual Machines
 $V=v_1,v_2,v_3,\dots$
- It is assumed that the environment is static and all the jobs are independent.
- Output that which Cloudlet must be executed to which VM to have the least Turnaround time.

4.5 Proposed Algorithm

- **Input** The set of cloudlets, and virtual machines.
- **output** Pairs of Cloudlet and Virtual Machine.
- For every Cloudlet 1....i
- for every Vm 1..j
- Calculate the turnaround time which is $\text{turnaround time} = \text{waiting time} + \text{cloudlet length} / \text{Power of VM}$

- Update the VM waiting time and schedule the cloudlet on that VM which results the least turnaround time.
- Loop stops when all cloudlets are traversed.

4.6 Simulation Result

The cloud simulation environment created is as follows:-

- No. of host=5
- No. of pes in each host=4
- The MIPS of each host=2000
- No. of cloudlet=100
- Half of total cloudlet length=400000 MI
- Rest cloudlet length=100000 MI
- Virtual Machine varies

The output of the simulation result is

Table 4.1: Comparison between existing and proposed algorithm

No. of VM	Existing Algo	Proposed Algo
4	5000.1	4250.09
8	2600.09	2200.1
16	1400.1	1200.1
32	800.1	650.1

The above algorithm gives the least turnaround time of every cloudlet. By using this algorithm we can reduce the execution time of jobs. Else job matching is an NP hard problem but by using these heuristics we can match in an polynomial complexity.

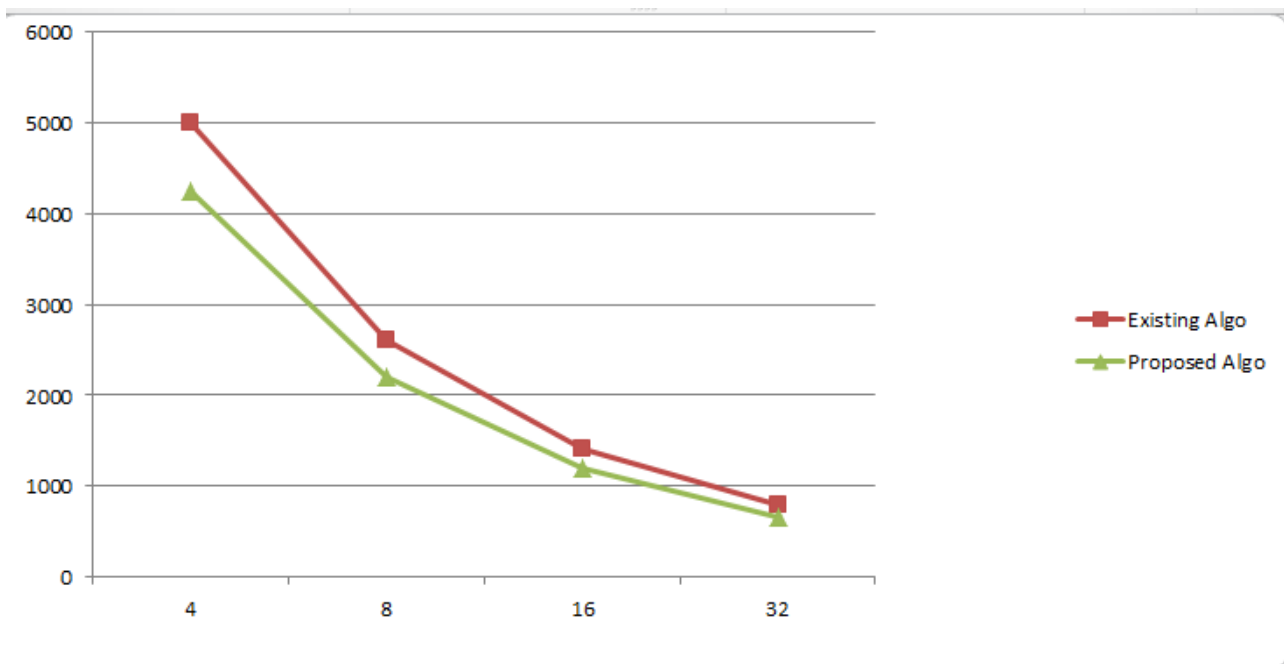


Figure 4.2: Result Graph

In the graph above it is very clear that the proposed algorithm outperforms the existing algorithm.

Now let us perform some more simulations to see upto to see that in how many cases our proposed technique performs better than the existing techniques.

The description of the new cloud environment is shown below.

- No. of host=5
- No. of pes in each host=4
- The MIPS of each host=2000
- No. of VM=8
- Half of total VM Power=1000 MIPS
- Rest VM Power=2000 MIPS
- Cloudlet varies

The results of simulation is shown below

Table 4.2: Comparison between existing and proposed algorithm

No. of Cloudlets	Existing Algo	Proposed Algo
50	1400.1	1200.1
100	2600.09	2200.1
150	3800.1	3250.09
200	5000.1	650.1

Form the above result it is very clear that if we vary the number of cloudlets and keep the number of virtual machine constant in that case also our algorithm outperforms the existing algorithm.

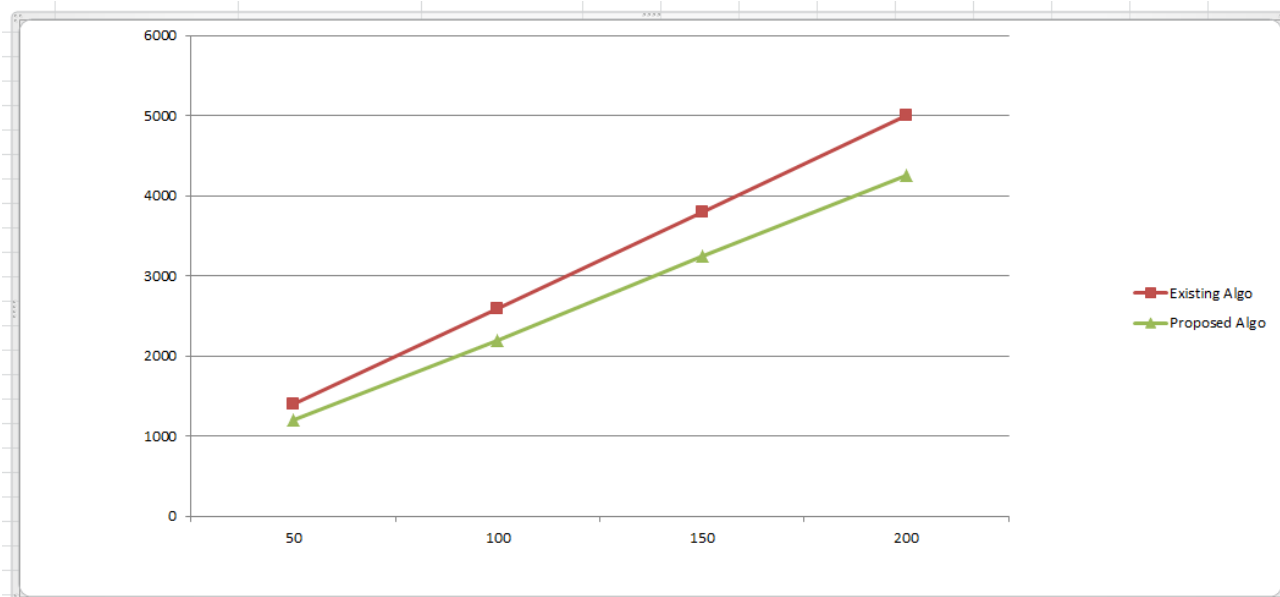


Figure 4.3: Result Graph

From the graph above it becomes very clear that our proposed algorithm has outperformed the existing algorithm.

Chapter 5

Conclusion

Achievements

Limitations & Future Scope

Chapter 5

Conclusion

5.1 Achievements

Above all descriptions, results make you feel that a good scheduling algorithm can really help to reduce the overall execution time of the cloudlets which indeed will help the cloud providers to attract more and more customers. Our results above clearly proves that if cloudlets are given intelligently to the viirtual machines than the overall execution can be improved very easily.

5.2 Limitation & Future Scope

As we seen that in our simulation we have kept the cloud environment to be static which means that no virtual machines can be created at runtime and no new hosts and datacenter can add themself to the environment. But situation never exist in real cloud environment as in reality the cloud environment will expand and contract it's size during the runtime. Therefore, it is very important to change this algorithm to the dynamic cloud environment.

————- END ———-

Bibliography

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.
- [3] X. Li, X. Jiang, P. Huang, and K. Ye, “Dartcsim: An enhanced user-friendly cloud simulation system based on cloudsim with better performance,” in *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, vol. 1, pp. 392–396, IEEE, 2012.
- [4] R. Buyya, C. S. Yeo, and S. Venugopal, “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,” in *High Performance Computing and Communications, 2008. HPCC’08. 10th IEEE International Conference on*, pp. 5–13, Ieee, 2008.
- [5] B. T OGRAPH and Y. R. MORGENS, “Cloud computing,” *Communications of the ACM*, vol. 51, no. 7, 2008.
- [6] T. Velte, A. Velte, and R. Elsenpeter, *Cloud computing, a practical approach*. McGraw-Hill, Inc., 2009.
- [7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in *Computer Security–ESORICS 2009*, pp. 355–370, Springer, 2009.

- [8] M. Hogan, F. Liu, A. Sokol, and J. Tong, “Nist cloud computing standards roadmap,” *NIST Special Publication*, vol. 35, 2011.
- [9] S. G. Grivas, T. U. Kumar, and H. Wache, “Cloud broker: Bringing intelligence into the cloud,” in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 544–545, IEEE, 2010.
- [10] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, “Introducing stratos: A cloud broker service.,” in *IEEE CLOUD*, pp. 891–898, 2012.
- [11] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. Masoud Sadjadi, and M. Parashar, “Cloud federation in a layered service model,” *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012.
- [12] L. Carrier, G. Cato, and K. Von Essen, “The backscattering and extinction of visible and infrared radiation by selected major cloud models,” *Applied Optics*, vol. 6, no. 7, pp. 1209–1216, 1967.
- [13] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities,” in *High Performance Computing & Simulation, 2009. HPCS’09. International Conference on*, pp. 1–11, IEEE, 2009.
- [14] A. Li, X. Yang, S. Kandula, and M. Zhang, “Cloudcmp: comparing public cloud providers,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 1–14, ACM, 2010.
- [15] P. Hofmann and D. Woods, “Cloud computing: the limits of public clouds for business applications,” *Internet Computing, IEEE*, vol. 14, no. 6, pp. 90–93, 2010.
- [16] F. Doelitzscher, A. Sulistio, C. Reich, H. Kuijs, and D. Wolf, “Private cloud for collaboration and e-learning services: from iaas to saas,” *Computing*, vol. 91, no. 1, pp. 23–42, 2011.

- [17] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pp. 27–33, Ieee, 2010.
- [18] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *Internet Computing, IEEE*, vol. 13, no. 5, pp. 14–22, 2009.
- [19] J. Qiu, J. Ekanayake, T. Gunarathne, J. Y. Choi, S.-H. Bae, H. Li, B. Zhang, T.-L. Wu, Y. Ruan, S. Ekanayake, *et al.*, "Hybrid cloud and cluster computing paradigms for life science applications," *BMC bioinformatics*, vol. 11, no. Suppl 12, p. S3, 2010.
- [20] M. A Vouk, "Cloud computing—issues, research and implementations," *CIT. Journal of Computing and Information Technology*, vol. 16, no. 4, pp. 235–246, 2008.
- [21] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, "Portable cloud services using toasca.," *IEEE Internet Computing*, vol. 16, no. 3, 2012.
- [22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De 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.
- [23] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*, pp. 137–142, IEEE, 2012.
- [24] S. Sadhasivam, N. Nagaveni, R. Jayarani, and R. V. Ram, "Design and implementation of an efficient two-level scheduler for cloud computing environment," in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*, pp. 884–886, IEEE, 2009.

- [25] P. Salot, “A survey of various scheduling algorithm in cloud computing environment,” *International Journal of research and engineering Technology (IJRET)*, ISSN, pp. 2319–1163, 2013.