

# Creation and Allocation of Virtual Machines for Execution of Cloudlets in Cloud Environment

Bachelor of Technology  
In  
Computer Science & Engineering  
By  
**Durbar Show**  
**110CS0153**



Department of Computer Science and Engineering  
National Institute of Technology, Rourkela  
Rourkela, Sundargarh,  
Odisha, 769008, India  
May, 2014

# Creation and Allocation of Virtual Machines for Execution of Cloudlets in Cloud Environment

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology  
In  
Computer Science & Engineering

By  
**Durbar Show**  
**110CS0153**

Under the guidance of:  
**Prof. P.M. Khilar**



Department of Computer Science and Engineering  
National Institute of Technology, Rourkela  
Rourkela, Sundargarh,  
Odisha, 769008, India  
May, 2014



Department of Computer Science and Engineering  
National Institute of Technology, Rourkela  
Rourkela – 769008, Odisha, India

## Certificate

This is to certify that the work in the thesis entitled “**Creation and allocation of Virtual Machines for execution of Cloudlets in Cloud Environment**” submitted by **Durbar Show**, in partial fulfilments of the requirements for the award of Bachelor of Technology Degree in the department of Computer Science and Engineering at National Institute of Technology, Rourkela is a record of original and authentic research carried out by him under my supervision and guidance.

Place: NIT Rourkela

Date: May, 2014

Prof. P.M. Khilar  
Dept. of Computer Science and Engineering  
National Institute of Technology, Rourkela  
Odisha-769008

# Acknowledgement

First of all, I would like to express my earnest and sincere gratitude to my advisor Prof. P.M. Khilar for support he has provided me in my research. He has always motivated me by giving suggestion at regular intervals and thereby has been a constant source of knowledge. Being a guide he helped me in my research work and in the writing of this thesis.

I would also like to thank all the other faculty members of the Department of Computer Science and Engineering for being supportive in the evaluation of the research work.

I would like to thank my father for being there always whenever I needed him. I would like to thank my mother for providing me constant moral support during my studies. I would like to thank Aristotle, Friedrich Nietzsche, Slavoj Žižek for the philosophical advice they provided.

I would also like to thank my friends for being with me and sharing a part of their life.

Durbar Show  
110CS0153  
[durbarshow@gmail.com](mailto:durbarshow@gmail.com)

# Abstract

Cloud Computing refers to delivery of both the softwares as a service and the infrastructure and the platforms that provide those services. In other words , Cloud computing is that service that enables us to use computing resources such as processing entities, storage and applications as on-demand services over the internet. Commonly in cloud computing we refer a “Cloud” as the service, whether it is software or platform or infrastructure as a service, which is implementing the features of Cloud Computing.

Cloud Computing has affected our lives a lot. Slowly it is making a strong a firm position in our lives, and the day is not very far when it will become an integral part of our great lives. Cloud computing may make drastic changes in the business models that are currently been employed in the Technology industries. We can see that how the revolution has already started. Services like Google Cloud, Pinterest, Flickr, Facebook, Amazon Cloud Services are to name a few of the current cloud services that are available in the market. We can see how this new technology, however it existed since 1970’s, is getting a part of our daily activities.

All the service such as web hosting, social networking, data storage, and others are emerging cloud applications. All these applications have certain different requirements and specifications for their development. Analysing and testing different scheduling and allocation algorithms for the development of these applications on a real cloud environment is really challenging because most of the cloud applications show changing incoming requests and moreover testing algorithms on a real cloud can cost us a lot.

For testing the effectiveness of a particular policy that is to be implemented on a cloud we need a simulation environment that can provide us an environment that is close to the actual cloud, and can generate results that can help us in the analysis of the policies so that we can deploy them on actual Clouds. CloudSim is one such simulation toolkit that allows us to test and analyse our scheduling and allocation algorithms. In this project work we have used CloudSim version 3.0 to test and analyse our policies and modifications in the current policies.

# Contents

<b>1. INTRODUCTION</b>	<b>8</b>
1.1 Cloud Computing	8
1.2 History	8
1.3 Future	9
1.4 Project Outline	9
1.5 Motivation	10
1.6 Summary	11
<b>2. Literature Review</b>	<b>12</b>
2.1 Introduction	12
2.2 Literature Survey	12
2.3 Summary	13
<b>3. Comparison of basic approaches for cloudlet execution</b>	<b>14</b>
3.1 Cloudlets and basic cloudlet execution approaches	14
3.2 Cloudsim representation of scheduling policies	14
3.3 Simulation data	15
3.4 Simulation Plot	17
3.5 Summary	18
<b>4. Allocation of Processing Entities to Virtual Machines in Cloud Environment</b>	<b>19</b>
4.1 Processing Entity allocation to Virtual Machines	19
4.2 Approaches for allocation of Virtual Machine in CloudSim	19
4.3 Simulation Results	22
4.4 Summary	24
<b>5. Assignment of Virtual Machines to Cloudlets in Cloud Environment</b>	<b>26</b>
5.1 Assignment of Virtual Machines to Cloudlets for Execution	26
5.2 Representation of the DataCenter Broker in	26

CloudSim		
<b>5.3</b>	<b>Simulation Plots</b>	<b>28</b>
<b>5.4</b>	<b>Summary</b>	<b>30</b>
<b>6</b>	<b>Conclusion and Future Work Scope</b>	
<b>6.1</b>	<b>Conclusion</b>	<b>31</b>
<b>6.2</b>	<b>Future Work</b>	<b>31</b>
<b>6.3</b>	<b>Summary</b>	<b>32</b>
<b>7</b>	<b>References</b>	<b>33</b>

## List of Figures

<b>3.4</b>	<b>Simulation Data Set</b>	<b>17</b>
<b>3.4</b>	<b>Simulation Plot</b>	<b>18</b>
<b>4.3</b>	<b>Output Screen 1</b>	<b>23</b>
<b>4.1</b>	<b>Output Screen 2</b>	<b>24</b>
<b>5.2</b>	<b>CloudSim architecture</b>	<b>27</b>
<b>5.2</b>	<b>Data Center Broker</b>	<b>27</b>
<b>5.3</b>	<b>Simulation Plot</b>	<b>30</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 Cloud Computing

In plain English Cloud Computing can be defined as the ability to use computing resources whether it be applications, or storage or processing entities, over the internet. The National Institute of Standards and Technology defines cloud computing as “Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources, for example networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”[1]. The term “cloud” has been popularized by the marketers, which refers to a service such as the applications or the infrastructure or the platforms that underlies cloud computing technology beneath them. A virtual machine or a VM is a software enactment of a machine (e.g. a computer) that can execute applications like a physical machine.

### 1.2 History

The history of cloud computing is dated back to 1950’s when large scale mainframes were available in the academic institutions and corporations. Then in the 1960’s John McCarthy got the idea that someday computing resources can be organized as public utility.

In the 2000s Amazon played an important role in the development of cloud and bringing it in for commercial purposes.



### **1.3 Future**

Cloud computing, today is in the spotlight. It is getting the attention of large Technology Giants. The features of this new technology are explored, the main issues are being addressed and now it is ready for exploitation in the market. Still all the problems associated with cloud are not solved yet. That's why it is curiously being examined by the researchers as it is getting carefully speculated by the businessmen.

The cloud computing technology is still deficient of the proper standards, tools for analysis, and proper approaches that can handle the complexities at infrastructure and application level. That's why, in future there will be lots of research work going on both in the academic institutions and industry towards defining core algorithms, policies, and application benchmarking based on execution contexts.

CloudSim is one such toolkit which is covering the basic functions of a cloud. We have used CloudSim to perform analysis of our project, to put a step forward in the development of Cloud Computing.

### **1.4 PROJECT OUTLINE**

Implementing allocation and scheduling policies on a real cloud, just for the purpose of analysis is not a good idea. This can be imagined by the power consumptions of the Datacenters which are an integral part of a cloud. A Datacenter at CERN needs 2.7 megawatts of power. So we can imagine the need of cost reduction while developing and testing of various allocation and scheduling policies. Our project too is developed and tested with the help of

CloudSim, which is a simulation toolkit as we may have mentioned earlier. Our project focuses on the following topics:-

- i) Comparison of the existing cloudlet execution algorithms.
- ii) Allocation of Processing Entities to the Virtual Machines that are being created inside the Datacenters.
- iii) Proper Assignment of the Created Virtual Machines to the Cloudlets for achieving maximum number of Cloudlets execution.

## **1.5 MOTIVATION**

After analysing the existing policies of allocation of Processing Entities that are being implemented in the CloudSim, we decided to modify them and enhance the existing approaches. First of all we have compared the existing approaches of executing the cloudlets. Cloudlets are the jobs (tasks) that are assigned to a Virtual machine which is running inside a Host. Host is present inside the Datacenters. Then we go deeper in the assignment process and observe that Cloudlet execution depend on the creation of Virtual machines. The current approach assumes that all the Processing entities inside a Host are of the same MIPS(Million Instructions Per Second) rating. We, in our project have implemented a new approach which abolishes the assumption that all the Processing entities are of the same MIPS rating. We have assumed that processing entities of different MIPS rating maybe present inside a Datacenter and thus creating a more generalized Cloud environment.

## **1.6 SUMMARY**

Cloud Computing is the ability to use computing resources such as processing power, storage, and applications as a service, over the internet. The Cloud

technology is being curiously examined by the researchers and the business people. Although all the issues of this new technology are addressed but still some are yet to be addressed. The implementations and analysis of the scheduling and allocation policies in a real cloud environment can prove costly, that's why there's a need of a simulated environment. The Cloudsim, cloud simulation toolkit has implemented the basic features of a cloud. We have used CloudSim for testing and analysing new approaches for creation and allocation of virtual machines to Cloudlets for execution.

# CHAPTER 2

## LITERATURE REVIEW

### 2.1 Introduction

In this section we briefly discuss the research work done in the area of the cloudlet execution policies and the creation and allocation of virtual machines in the Cloud environment. The two basic approaches of Cloudlet execution are time shared and space shared. In “time-shared”, each cloudlet is allowed to run inside the virtual machine for a certain period of time, as the time limit finishes, another cloudlet is allowed to run and the same cycle repeats for all the cloudlets until all cloudlets have finished execution. In “space shared” approach the cloudlets runs inside the virtual machine until and unless it has finished its execution. Another cloudlet is allowed to run inside the virtual machine only if the current cloudlet has finished its execution. [2]

The approaches that are being followed while allocating Processing entities to virtual machines.

### 2.2 Literature Survey

Not much work has been done on the implementation and analysis of scheduling and allocation policies using CloudSim, because it has been recently released. Although there is some research work that exist like [3] which discusses about the energy efficient Virtual machine allocation in cloud. Some of the papers are proposing modification in the approaches that are being implemented inside cloudsim by the authors. The basic implementation methods in the Cloudsim java API are mostly authored by Rodrigo N. Calheiros,

Anton Beloglazov and Rajkumar Buyya. Any modification in the current API is welcome. One such paper is [4] written by S.Sadhasivam, N.Nagaveni, R.Jayarani, R.V. Ram, which discusses about the implementation of a two level scheduler for Cloud Computing. They have used Cloudsim as a toolkit.

## **2.7 SUMMARY**

After the literature survey we came to conclude that implementation of the basic cloud policies using Cloudsim is a challenging task but not impossible and needs focus. This work proposes modification in the current approach for Virtual machine allocation and creation for the execution of cloudlets that are being implemented in CloudSim, and the approaches are being authored by N. Calheiros, Anton Beloglazov.

# CHAPTER 3

## Comparison of the basic approaches for Cloudlet Execution

### 3.1 Cloudlets and basic cloudlet execution approaches

Cloudlets are jobs or tasks that are being assigned to a virtual machine for execution. The virtual machine implements two kinds of policies for cloudlet execution. The two basic policies are:-

- i) Time Shared
- ii) Space Shared

In Time-Shared scheduling policy the resources are being shared among the cloudlets. Each cloudlet gets the resources for execution for a certain period of time. After that period of time the resources are taken away from that cloudlet and are assigned to another cloudlet. In Space Shared the resources are not shared among cloudlets. A cloudlet owns the resources until it gets executed. We have compared these two scheduling policies

The same policies are applied for scheduling of Virtual machines which are running inside a host. In time shared the resources of the hosts are being shared by the virtual machine for a timely basis, but in space shared a virtual machine can run on a Host only if free processing entities are available.

### 3.2 CloudSim Representation of Scheduling Policies

CloudSim API which is a java based API, has represented the two basic cloudlet scheduling algorithms with Classes named as “CloudletSchedulerTimeShared”

and “CloudletSchedulerSpaceShared”. Similarly the two basic classes for implementing the two basic scheduling algorithms for virtual machines, the two classes are “VmSchedulerSpaceShared” and “VmSchedulerTimeShared”. By specifying the type of scheduling policy we want to implement to cloudsim we can get the desired results.

### **3.3 Simulation Data**

The performances based on the average waiting time of Cloudlets in completely time shared and completely space shared environment were measured. The data was generated by conducting simulations with 2 datacenters, with 1 host each and 1 virtual machine running inside each host. The scheduling policy implemented by the Host for the scheduling of Virtual machine is Space Shared. As there is only one virtual machine it doesn't matter which virtual machine scheduling policy is implemented. The scheduling policy is significant only if there two or more VMs. The details of the configuration of the DataCenters, the Hosts, The virtual machines and the Cloudlets are detailed below:-

#### **DataCenter Characteristics :-**

Architecture: x86

Operating System: Linux

Hypervisor: Xen

#### **Characteristics of Host (Each datacentre has a host.):-**

Ram: 1024 MB

Storage: 500000 MB

Number of Processing Entities: 1

MIPS rating of Processing Entity: 1000

**Virtual Machine Requirements** (Each host has a virtual machine):-

MIPS rating: 250

Image Size on Disk: 10000 MB

RAM: 512 MB

Number of Required PEs: 1

Hypervisor: Xen

**Characteristics of the cloudlets:-**

Length (in terms of instructions): 40000

Input File Size: 300

Output File Size: 300

The virtual machines are created first. Then the cloudlets are assigned to them in a sequential manner. After each cloudlet has been assigned a virtual machine the processing is started by the Virtual Machine. We have submitted cloudlets ranging from 1 to 20 and noted down the total time taken by them to execute, with the virtual machine implementing the policies time shared and space shared one by one.



### 3.4 SIMULATION PLOTS

The simulation data is shown below.

CloudLEts	Time Shared	Space Shared
1	160	160
2	160	160
3	266.666	213.333
4	320	240
5	415.995	288
6	479.992	320
7	571.425	365.714
8	640	400
9	728.888	444.444
10	800	480
11	887.264	523.636
12	959.984	560
13	1046.133	603.076
14	1119.976	640
15	1205.322	682.666
16	1280	720
17	1364.688	762.352
18	1439.968	800
19	1524.195	842.105
20	1600	880

Figure 1

The resultant graph is shown below.

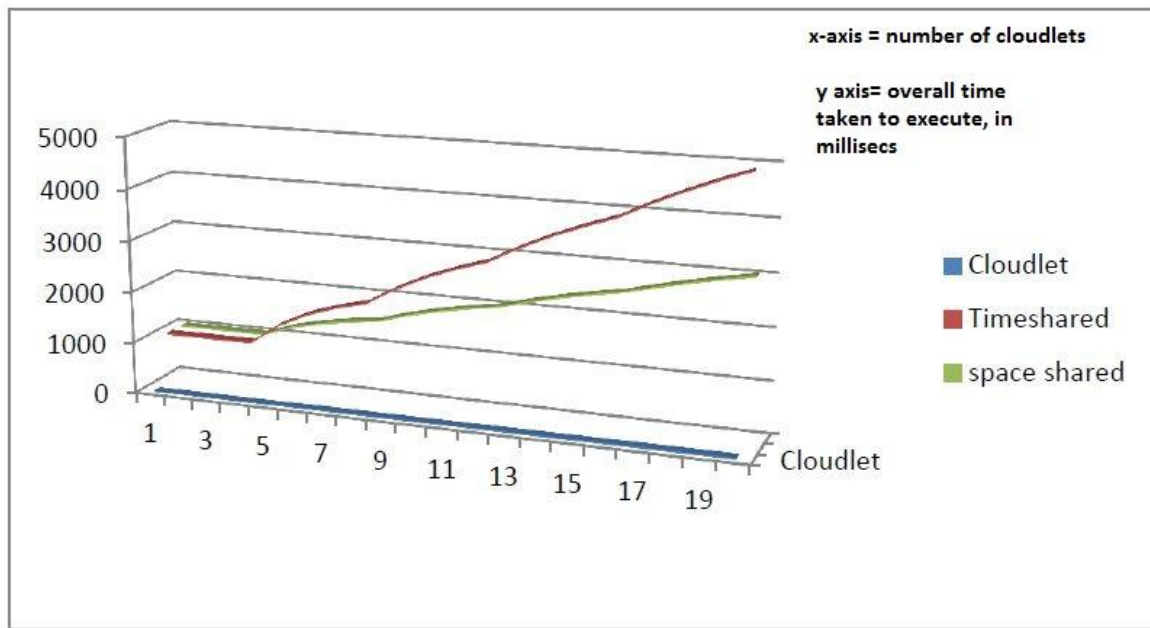


Figure 2

### 3.5 SUMMARY

The results generated show that Space Shared scheduling policy is outperforming the time shared scheduling policy in terms of average execution time taken by a cloudlet for complete execution. This is because in time shared there is the process of removal of a cloudlet from the current execution state to the waiting state, and then bringing back the cloudlet from the waiting queue to the execution state. This results in degradation of performance in terms of overall execution period. The theory has been confirmed with the results that we have got.

# CHAPTER 4

## Allocation of Processing Entities to Virtual Machines in Cloud Environment

### 4.1 Processing Entity allocation to Virtual Machines

The processing entity allocation to virtual machine is pretty simple. The numbers of processing entities are allocated to a particular virtual machine only if the numbers of processing entities required by the virtual machines are available in the host. The allocation also depends on the scheduling policy implemented by the “VmScheduler”. “VmScheduler” is the class representation of the scheduling policy for the virtual machines. “VmShedulerTimeShared” and “VmSharedSpaceShared” implements “VmScherduler”. If the “VmSharedSpaceShared” then each time a virtual machine is allocated the number of processing entities reduces by the amount of processing entities required by the virtual machine. But in case of time shared policy for virtual machine scheduling, a virtual machine creation is denied only if the number of processing entities required by the virtual machine exceed the actual number of processing entities that the host has on which the virtual machine is to be hosted.

### 4.2 Approaches for allocation of Virtual Machine in CloudSim

Cloudsim provides a virtual machine approach which assumes that the Processing entities available inside a Host are of the same MIPS rating. This assumption has drawbacks. This assumption bounds us to assume that we can

have only Processing entities of same MIPS rating inside a Host. We, in our work has modified the basic approach that CloudSim has implemented. The following code fragment is from the “VmSchedulerSpaceShared.java” which is present in the CloudSim API.

```
protected boolean allocatePesForVm(String vmUid, List<Double>  
mipsShareRequested)  
{  
    /*function code*/  
    for (Double mips : mipsShareRequested) {  
        // each virtual PE of a VM must require not more than the  
capacity of a physical PE  
        if (mips > peMips) {  
            return false;  
        }  
        totalRequestedMips += mips;  
    }  
    /*function code*/  
}
```

The code modification done is shown in the following code snippet.

```
public boolean newnewallocatePesForVm(Vm vm, List<Double> mipsShare)  
{  
    // if there is no enough free PEs, fails  
    List<Pe> freepes=getFreePes();  
    if (freepes.size() < mipsShare.size())  
{  
        return false;  
}
```

```

    }
    quicksortPeList(freepes,0,freepes.size()-1);
    quicksort(mipsShare,0,mipsShare.size()-1);
        List<Pe> selectedPes = new ArrayList<Pe>();
        double totalMips = 0;
        int low=0,high=freepes.size()-
1,current=0,size=mipsShare.size(),left;//assign this to avoid noted mistake in
next steps
        left=size;
        boolean bin=true,found;
        for(Double mips:mipsShare)
        {
            found=false;
            current++;
            for(int i=low;i<=high;i++)
            {
                if(mips<=freepes.get(i).getMips())
                {
                    low=i+1;
                    selectedPes.add(freepes.get(i));
                    totalMips += mips;
                    found=true;
                    left=size-current;
                    if(left>(high-low)+1)
                    {
                        return false;
                    }
                }
            }
        }

```

```
        break;
    }
}
if(!found)
{
    return false;
}
}
```

The code modification done, discards the assumption that all the processing entities present inside a Host are of the same MIPS rating. The modified approach first of all sorts the available Processing entities list, and then starts allocation of Processing entities to virtual machines. The search space is reduced with each allocation of processing entities to a certain Virtual machine.

### **4.3 Simulation Results**

The modifications done are the modifications to the current space shared strategy in virtual machine scheduling. We have not taken time shared policy because of the migration approaches that are associated with that. So the current modifications are based on the Space Shared policy for virtual machine allocation.

The characteristics of Virtual machine taken while conducting the simulation is detailed below:-

MIPS requirement: 5000

Hypervisor: "xen"

Number of Processing Entities required: 7

RAM required: 1024 MB

Image size: 1000 MB

List of Available Processing Entities available with respective MIPS rating:  
[10000, 9000, 8000, 11000, 2000, 3000]

The first simulation shows the failure of the current approach implemented in Cloudsim.

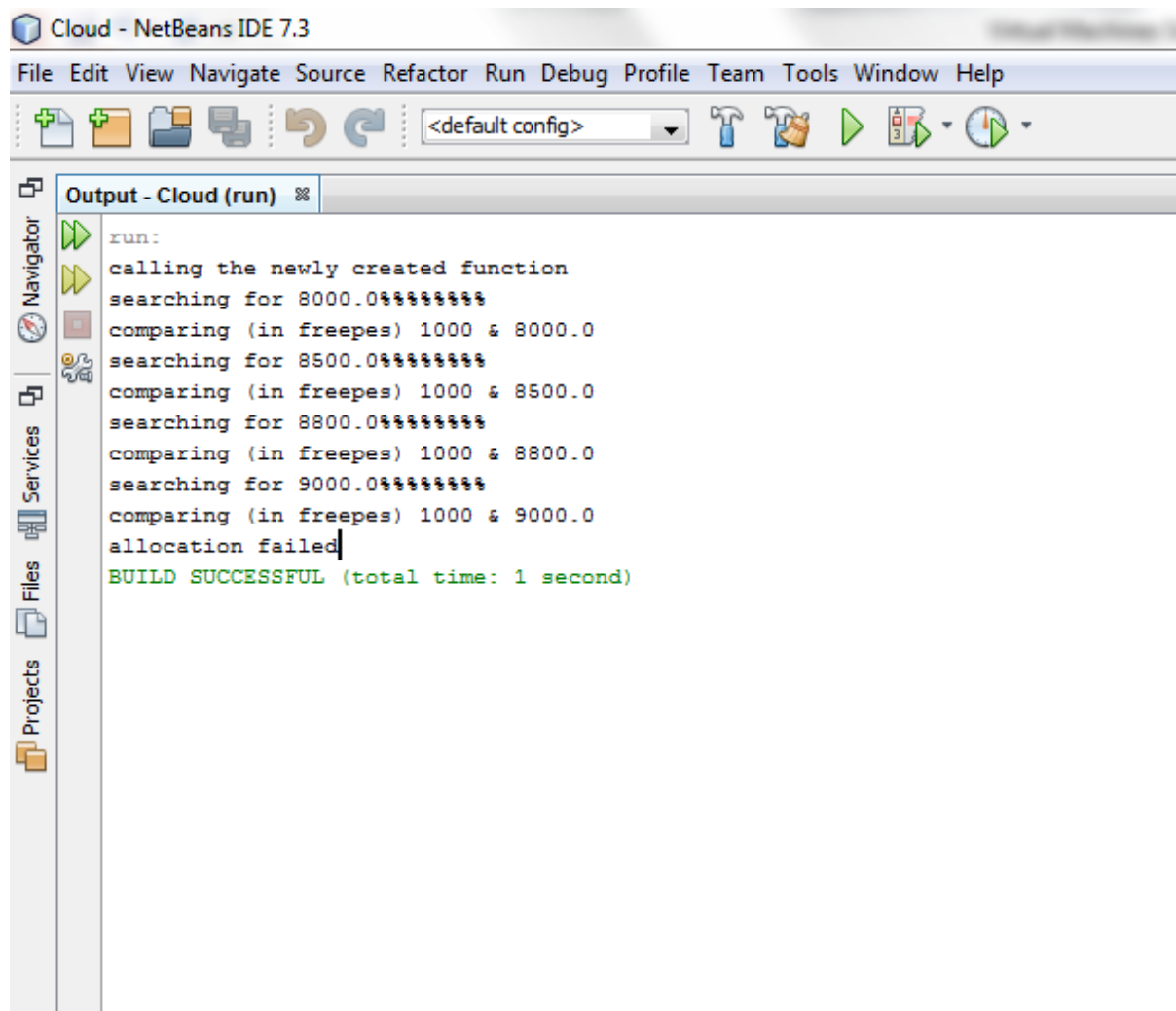
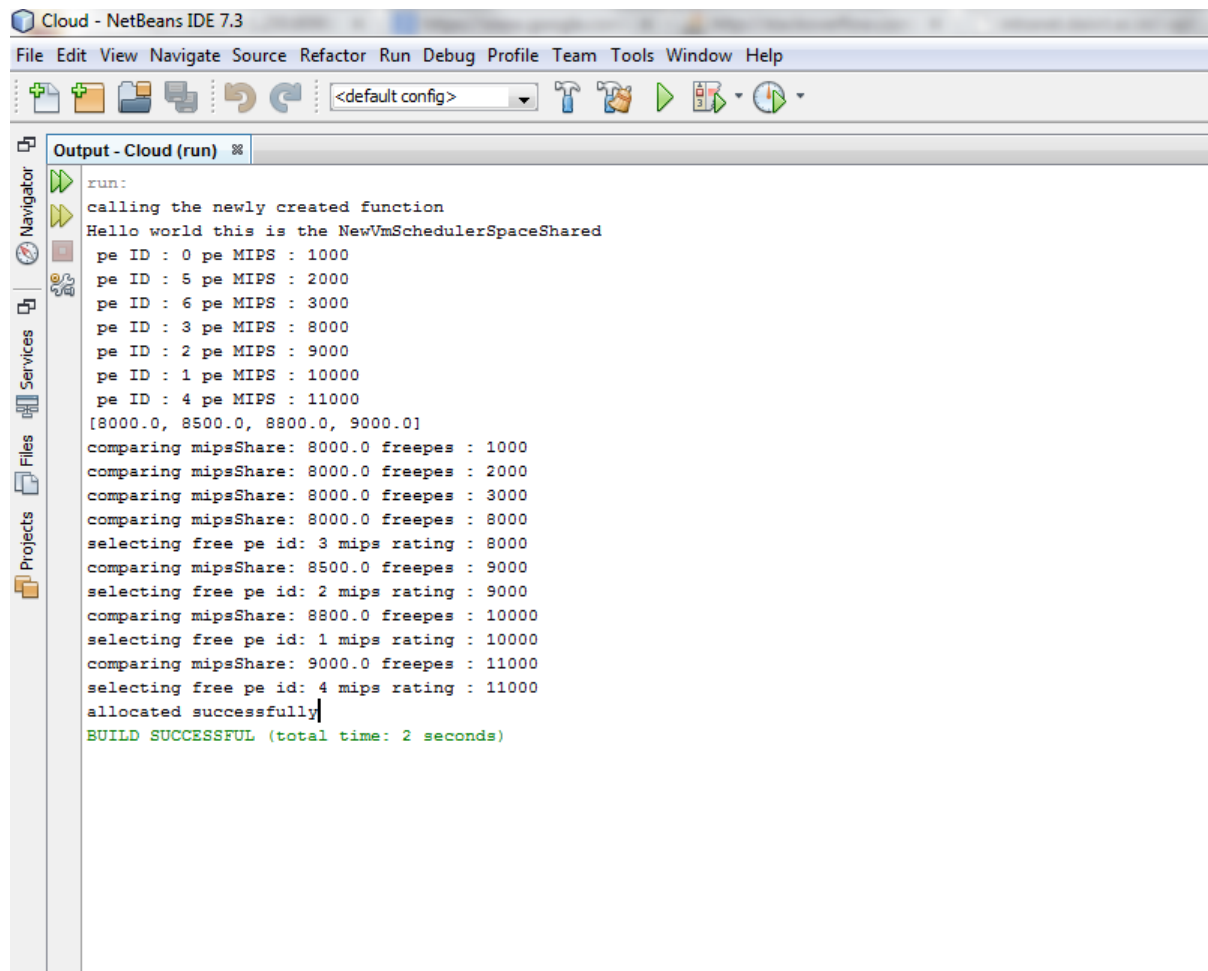


Figure 3

We can see how the current approach is comparing the demanded Processing entities with the available Processing entities list. This approach discards the search of the apt Processing entity if the required MIPS rating of the Processing entity is greater than the first element in the list of the available Processing entities.

The diagram below shows the successful allocation of a virtual machine with the demanded processing entity request and with the available Processing entity request.



```
Cloud - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - Cloud (run)
run:
calling the newly created function
Hello world this is the NewVmSchedulerSpaceShared
pe ID : 0 pe MIPS : 1000
pe ID : 5 pe MIPS : 2000
pe ID : 6 pe MIPS : 3000
pe ID : 3 pe MIPS : 8000
pe ID : 2 pe MIPS : 9000
pe ID : 1 pe MIPS : 10000
pe ID : 4 pe MIPS : 11000
[8000.0, 8500.0, 8800.0, 9000.0]
comparing mipsShare: 8000.0 freepes : 1000
comparing mipsShare: 8000.0 freepes : 2000
comparing mipsShare: 8000.0 freepes : 3000
comparing mipsShare: 8000.0 freepes : 8000
selecting free pe id: 3 mips rating : 8000
comparing mipsShare: 8500.0 freepes : 9000
selecting free pe id: 2 mips rating : 9000
comparing mipsShare: 8800.0 freepes : 10000
selecting free pe id: 1 mips rating : 10000
comparing mipsShare: 9000.0 freepes : 11000
selecting free pe id: 4 mips rating : 11000
allocated successfully
BUILD SUCCESSFUL (total time: 2 seconds)
```

Figure 4

We can see how modified approach is successfully sorting the available Processing entity list and then with each allocation it is reducing the search space and thus successfully allocating the right Processing entities to the Virtual machines.

#### 4.4 SUMMARY

We conclude with the thought that the current assumption that all the Processing Entities are of the same MIPS rating does not generalizes the cloud



environment. The modified shows a successful allocation of processing entities to a certain virtual machine.

# CHAPTER 5

## Assignment of Virtual Machines to Cloudlets in Cloud Environment

### 5.1 Assignment of Virtual Machines to Cloudlets for Execution

Here comes a new actor into the picture that we are going to discuss. It is the Datacenter Broker. The Datacenter broker acts on behalf of the user. It hides virtual machine management, like virtual machine creation, submission of cloudlets to the virtual machines created and destroying of virtual machines.

### 5.2 Representation of the DataCenter Broker in CloudSim

In CloudSim the DataCenter Broker is represented by a class with the same name “DatacenterBroker”. The diagram below gives us a view of the CloudSim architecture.

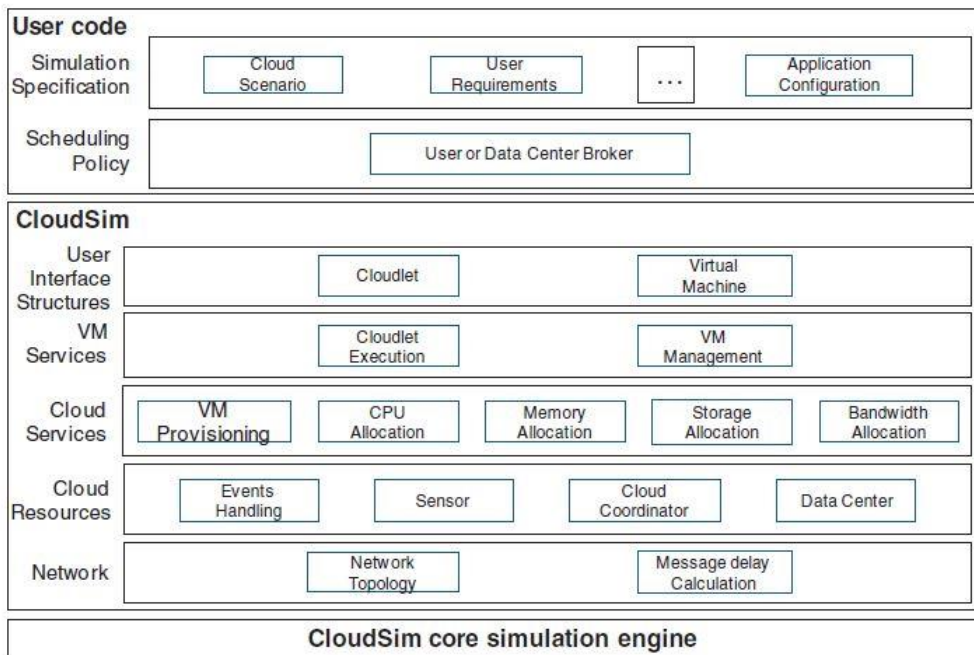


Figure 5 : Taken from R. Buyya, C. S. Yeo, and S. Venugopal. Marketoriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008

Another diagram which is shown below, gives us a picture of the place and position of the Datacenter Broker.

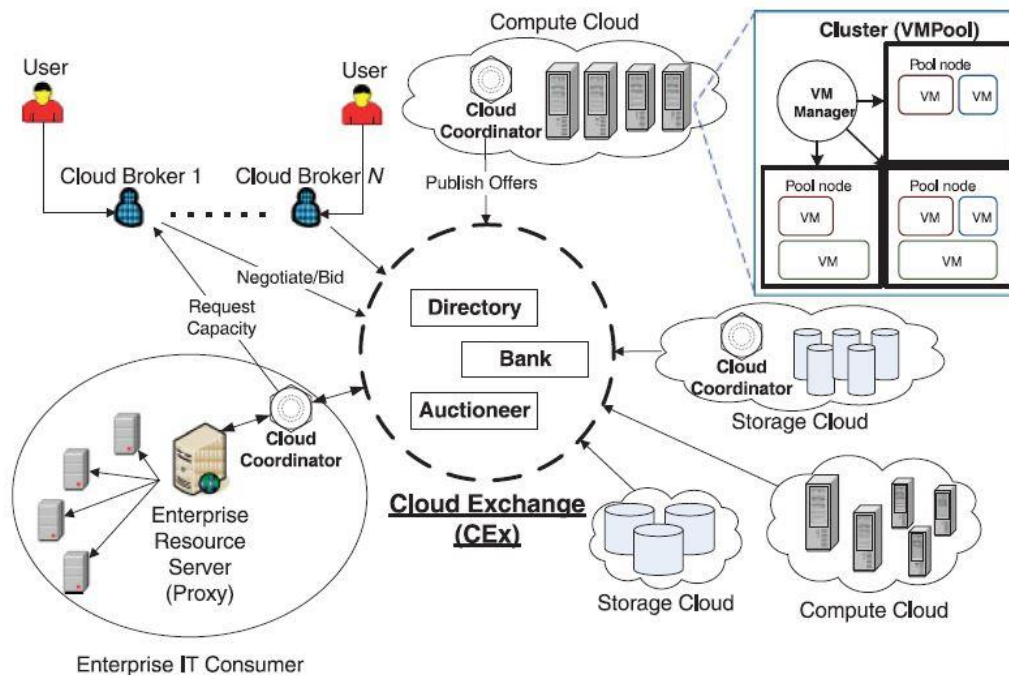


Figure 6 : R. Buyya, C. S. Yeo, and S. Venugopal. Marketoriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008

So as the Datacenter Broker is responsible for the submission of the cloudlets to virtual machines, we must take care of the fact that a cloudlet must be submitted to a virtual machine which is capable of completing it. In the current implemented approaches in Cloudsim the jobs are being assigned to Virtual Machines which are not capable of completing them they get queued, and thus never gets executed. Because the existing virtual machines are the only virtual machines which will run inside the Host until the end. This leads to lesser number of Cloudlets getting executed. In our proposed approach using the new Virtual machine allocation strategy that we discussed in the previous chapter along with the basic space shared technique to execute cloudlets. The new approach proposes that instead of randomly assigning of cloudlets to virtual machines which are not capable of completing the execution of the cloudlets we should first search a proper Virtual machine which has the capability of executing and completing the execution of the cloudlet.

### **5.3 SIMULATION PLOTS**

The characteristics of the virtual machine that used in the simulation to prove the point are detailed below:-

Two of the VMs have the following requirements

**Virtual Machine Requirements** (Each host has a virtual machine):-

MIPS rating: 2500

Image Size on Disk: 10000 MB

RAM: 512 MB

Number of Required PEs: 1

Hypervisor: Xen

One of the virtual machines has the following

**Virtual Machine Requirements** (Each host has a virtual machine):-

MIPS rating: 250

Image Size on Disk: 10000 MB

RAM: 512 MB

Number of Required PEs: 2

Hypervisor: Xen

**Characteristics of Host (Each datacentre has a host.):-**

Ram: 2048 MB

Storage: 500000 MB

Number of Processing Entities: 4

MIPS rating of Processing Entity: 1000

**Characteristics of the cloudlets:-**

Length (in terms of instructions): 40000

Input File Size: 300

Output File Size: 300

The Simulation was done with 1 Datacenter and 1 host inside it with 3 virtual machines running inside the host. Varying number of Cloudlets were sent to the virtual machines and the status of the number of cloudlets which were successfully completed were plotted on a graph against the number of cloudlets that were successfully completed with the current approach that is already present in CloudSim. The graph below depicts the success of the new proposed approach. Each category has a certain number of cloudlets which were supplied to the virtual machine for execution.

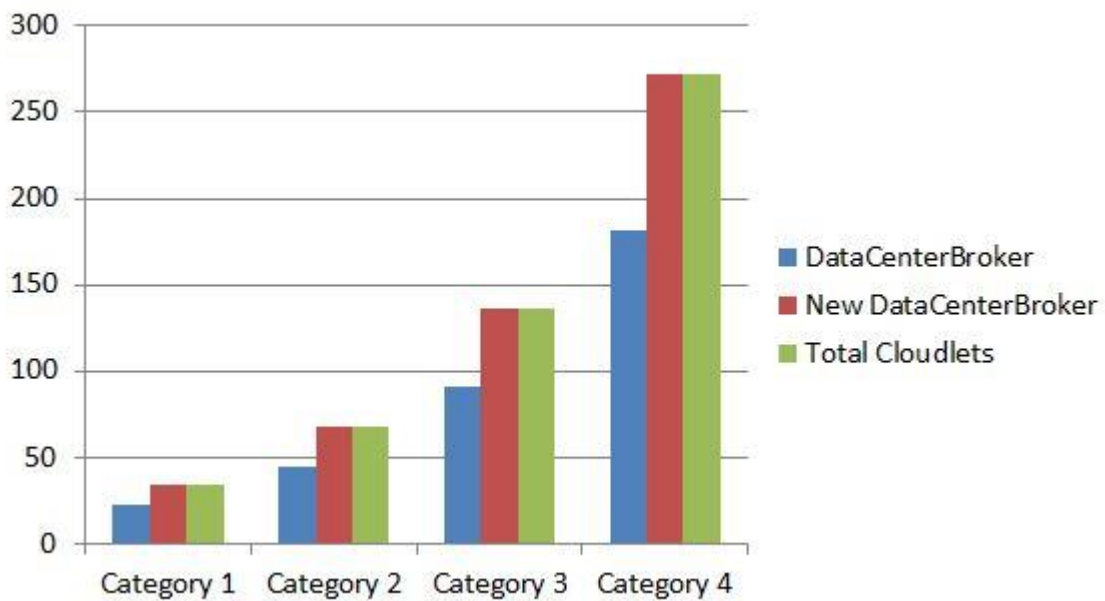


Figure 7 Red bar is for the number of cloudlets which were successfully executed using the new proposed approach. Blue bar represents the number of cloudlets that were successfully executed using the existing approach that is being implemented in CloudSim. Each category has a certain number of Cloudlets. We increased the number of cloudlets with each category.

## 5.4 SUMMARY

The above results validates that the number of cloudlets that are getting executed completely, when there are virtual machines that are capable of completing the execution of cloudlets, is more in the new proposed approach.

# CHAPTER 6

## Conclusion and Future Work Scope

### 4.1 Conclusion

In this project we have successfully compared the two basic cloudlet scheduling algorithms the “cloudlet scheduler space shared” and the other one the “cloudlet scheduler time shared” and saw that time has performance degradation in cases in which we compare the overall execution time taken by the cloudlet.

Finally in the end we conclude that we have successfully simulated a heterogeneous cloud environment in which we discarded the assumption that all the processing entities inside a host are of the same MIPS rating and have successfully allocated processing entities with different MIPS rating to virtual machines which are need of processing entities. We have also allocated the apt virtual machine which will be able to complete the execution a cloudlet successfully and plotted the total number of cloudlets executed successfully and have seen that the new proposed approach surpasses the existing approach in terms of the total number of cloudlets successfully executed.

### 4.2 Future Work

Due to increased number of cloudlet execution, in some situation it may happen that most of the virtual machines inside the host are idle and only a few are handling the task of cloudlet execution. These idle virtual machines are

acquiring resources which too are idle. These idle resources acquired by the virtual machines are processing entities. The future work proposes to free these idle processing entities. Creation of new virtual machines is an option. Another situation is that not sufficient virtual machines are created inside the host. We should be able to create virtual machines with the required characteristics which are needed to handle the current traffic of cloudlets so that cloudlets are not handled by just a few virtual machines while other are resting in peace.

#### **4.4 SUMMARY**

Cloud Computing is still a developing field. CloudSim API gives us the power to simulate various scheduling and allocation policies and analyse them on the basis of their performances. We in this project have done total 2 modifications and have shown that those modifications are better than the current existing approach.



# REFERENCES

1. The NIST Definition of Cloud Computing National Institute of Standards and Technology Special Publication 800-145 (Draft)7 pages (January. 2011) writers: Peter Mell ,Timothy Grance.
2. <http://www.cloudbus.org/cloudsim/doc/api/>
3. Green Computing Conference and Workshops (IGCC), 2011 International, Issue Date: 25-28 July 2011, Written by: Jansen, R.; Brenner, P.R.
4. Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on, Issue Date: 27-28 Oct. 2009, Written by: Sadhasivam, S.; Nagaveni, N.; Jayarani, R.; Ram, R.V.
5. R. Buyya, C. S. Yeo, and S. Venugopal. Marketoriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008.*
6. R. Buyya, C. S. Yeo, and S. Venugopal. Marketoriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008.*
7. Skewness-Based Min-Min Max-Min Heuristic for Grid Task Scheduling  
Sanjaya Kumar Panda, Pratik Agrawal, Pabitra Mohan Khilar, Durga Prasad Mohapatra: Proceedings of the 2014 Fourth International Conference on Advanced Computing & Communication Technologies; 02/2014