International Journal of Grid Distribution Computing
Vol.8, No.1 (2015), pp.145-158
http://dx.doi.org/10.14257/ijgdc.2015.8.1.14

Virtual Machine Allocation Policy in Cloud Computing Using CloudSim in Java

Kushang Parikh, Nagesh Hawanna, Haleema.P.K, Jayasubalakshmi.R and N.Ch.S.N.Iyengar

School of Computing Science and ngineerng
Vellore Institute of Technology, Tamil Nadu,Vellore-632014,India
kushangm.parikh@gmail.com,nagesh11190@gmail.com,nchsniyr@vit.ac.in

Abstract

Cloud computing is a very powerful concept that can be used to enhance the next generation data center and allow service provider to use data center capability provided by cloud and develop the application based on user requirement. Data center of this cloud computing has huge number of resources and list of applications (with different architecture, configuration and requirement for deployment) want to use those resource. Cloud computing environment uses virtualization concept and provides resources to application by creating and allocating virtual machine to specific application. There for resource allocation policies and load balance policies play very vital role in allocating and managing the resources among various application in clod computing life cycle. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing environments. The model proposed by this paper for dynamic load balance policy with considering different attributes and different service level agreements in cloud computing environment helps this environment to utilize their resources and improves performance. The proposed model uses Hungarian algorithm and the result is verified by simulating this model using CloudSim.

Keywords: *virtual machine allocation policy, load balance, CloudSim, Resource allocation policy, modeling and simulation, Hungarian algorithm.*

1. Introduction

Cloud computing provide various services like IaaS (infrastructure as a service), PaaS (platform as service), and SaaS software based on a pay-as-you-use model to cloud customers and has potential to transfer a large part of the IT industry, making software even more attractive as a service. Cloud computing is the cutting edge in reckoning. Perhaps individuals can have all that they require on the cloud.

From user perspective cloud computing make them able to use and deploy their applications from anywhere on this planet and interest at focused expenses contingent upon clients QoS (Quality of Service) necessities. To provide these services continuously on demand, internally it uses many of technologies like virtualization, clustering, terminal service, application server and more. Virtualization is a foundational element of cloud computing environment. It can be defined as making of a virtual version of something, such as an operating system or servers or storage devices or network resources. [1]

2. Motivation

Cloud computing environment satisfies requirements by generating virtual machine and allocating the resources to them based on the requirements for the applications required to deploy them on the cloud computing environment. Theses application may

ISSN: 2005-4262 IJGDC

have different configuration and different architecture like social networking, web hosting, content delivery and real time data processing. Vitalization technology is the heart of cloud computing lifecycle and it is limited to physically available resources. Hence utilization of all resources is play important role in managing the cloud computing life cycle efficiently. In order to deploy these huge applications successfully and efficiently, Load Balancing is required to manage the lots of load dynamically in cloud computing environment. Hence Load balancing is also a critical part of cloud computing lifecycle.

The proposed model, provide dynamic load balance policy based on different parameter like execution time, memory required and helps cloud computing environment to improve its performance by resources utilization. For load balancing with cost optimization, proposed model internally uses Hungarian algorithm and for modelling and simulation CloudSim is used. CloudSim is an extensible simulation toolkit that enables modelling and simulation of Cloud computing environments.

3. Related Work

In [3] Author describes major challenges in the resource allocation in cloud computing environment. In this paper described challenges were related to the resource management policy but the allocation method is not described any more. In [1] author proposed architecture, using feedback control theory, with help of virtualization that do so using virtual machines, In virtual machine architecture all hardware resources are put in one place with memory sharing architecture and requested applications by cloud customer deployed as per SLA (Server lever agreement).In this paper architecture uses controllers: CPU, memory and Input output. Its objective is to control various virtualized assets usage to attain SLA of requisition by using control inputs for every virtual machine resources. Problem under the virtual machine based architecture is how to provide resources to each application with in response of time management based on workloads.

In [5], author proposed two layer architecture uses utility functions in resource allocation in static and also in dynamic manner with help of two agent local agent and global agent. The agent computes current work load and transfer it to global agent which is responsible for proving optimal configuration of resource allocation by analyzing work load given by local agent. For that synchronization mechanism between this two agents is perform important task, if any changes occurred in any one of them then they may have to be changed. In [2], author proposed an adaptive resource allocation algorithm in cloud computing environment. This paper used adaptive min-min scheduling and list scheduling in but it is for used in static manner.

In [4] author proposed cloud computing resource planning in multiple dimension like with respect to space or with respect to time or with respect to response time. Authors used Amazon EC2 for processing environment and describe be how this would ne benefited for cloud client and supplier.

In [5] author described the major concept in resource management in cloud computing environment with uncertainty. This uncertainty refers to parameter and policy.

In[6] author proposed the modeling in CloudSim a modeling and simulation tool kit and describes where as modeling strategy used in this simulator and benefits of this simulator. The broker strategy is described in this paper and described about the various parameters involve in it.

In [7] author describes various strategies to overcome the challenges related to configuration as per user demand in, Cloud computing environment. In this paper varies policy described for managing the virtual machine and author proposed one software framework for solving the issue discussed as major challenges. Author used CloudSim for modeling and simulation and describes varied modeling strategies for managing the virtual machine and data center resource in cloud computing environment and describe the simulation result of the CloudSim. In [11] author describe the term grid for providing high performance services for complex computing and data intensive scientific application.[8],[9],[10] described about ‘GridSim’ , ‘SimGrid’ , ‘GangSim’ simulator respectively for simulating the grid computing. SimGrid provides framework for simulating distributed application, GangSim used for modeling and simulator using virtualization concept and GridSim is event based simulator for heterogeneous Grid resource.

Scientists have additionally examined virtual machine arrangement crosswise over various crosswise over various cloud suppliers from numerous cloud suppliers under future requests with high accessibility prerequisites. In [11]-author proposed optimal virtual machine placement algorithm for minimizing the cost that cloud customer have to pay cloud provider, when they need virtual machine from cloud computing environment access as part of cloud service. In [12] author described the multi objective mechanism for scheduling applications that take various cost constrained and availability of resourced in account. In [13-15] author more focus on resource allocation strategy in selecting cloud provider, but in static manner for selecting a data center from distributed environment where global data center is available, with taking care of timing parameter.

In [16] author described briefly about the CloudSim toolkit for modeling and simulation environment. Author described usefulness of the CloudSim by various case studies, virtual machine management in CloudSim and also described about federated Cloud computing model.

4. CloudSim Architecture

This paper uses CloudSim for implanting and simulating the proposed way for load balancing in cloud computing environment and hence it is highly depends on. So it is necessary to understand basic the architecture of CloudSim described in Figure 1.

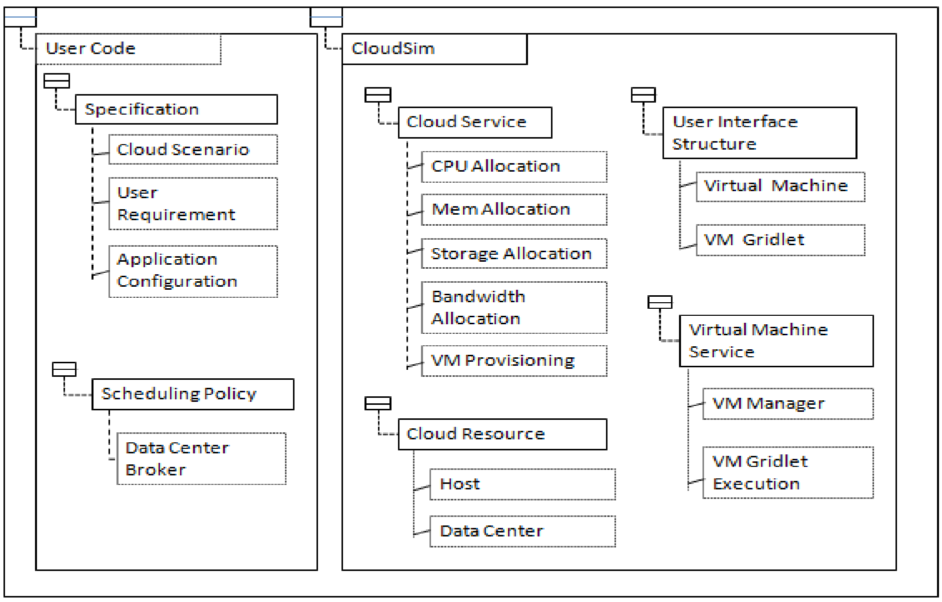

Figure 1. Describes the Layer Architecture of CloudSim

CloudSim: “An extensible simulation toolkit” that provide a way for modeling and simulation of Cloud computing environments. It enables to create virtual machine VMs and manages the virtual machines. It also provides way to simulate the data center. It provides the mapping and much other way for managing and utilizing the resources.

In CloudSim environment three major entities that perform most of the work are Broker, MIS (Management Information System) and Datacenter.

In this architecture Cloud user give request to cloud provider with their different configuration and requirements. Different configuration refers to Number of processing elements, amount of RAM, Space, number of instruction, file length etc. This all combines as task contained in cloudlet.

In cloud computing actual physical resources are available in single pool with shared memory. The Datacenter has actual resource. This entity contains list of host with different hardware configuration. Data centers registered them self to MIS for allowing the broker to access them. CloudSim provides services by using vitalization mechanism. It generates number of virtual machines and allocates them to job and simulates of a Data Center. In order to perform deploy requested application in terms of cloudlet with list of specification, broker map the cloudlet with virtual machines and requested to MIS for the knowing the information about actual physical hardware in data center in order to deploy of virtual machines in data center.

CloudSim allows to modeling various policies for binding the cloudlet to virtual machines, for allocating the virtual machines to host in data center, for scheduling the virtual machines, for power consumption, for creating topology in distributed networking and many more in order to utilize the resources and the performance in static or dynamic environment. This all policies take lots of policy in to the account like response time, space, memory, workload and many more.

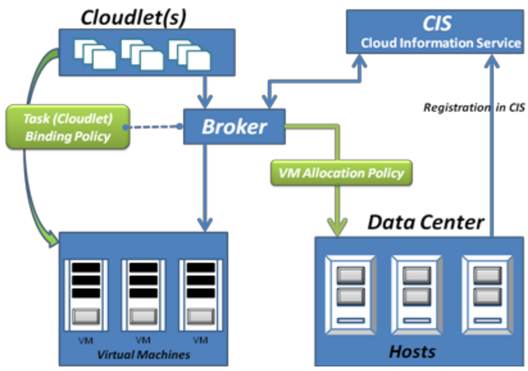

Figure 2. Proposed Policies in CloudSim Environment

This paper focused on mapping between the cloudlet to virtual machine in order to balance the workload and also on the virtual machine allocation policy. In cloud computing environment lots of workload needs to be maintained and through the virtualization is actually depends or limited by actual hard ware resources vm allocation policy play a vital role in cloud computing life cycle. We proposed a way for binding the task to vm with objective of providing the high speed or less execution time to the entire task. So each task can performed well. For that Hungarian method is used [17]. And to utilize the resources of data center this paper proposed a way that decides allocating the virtual machines to host with both having different configuration.

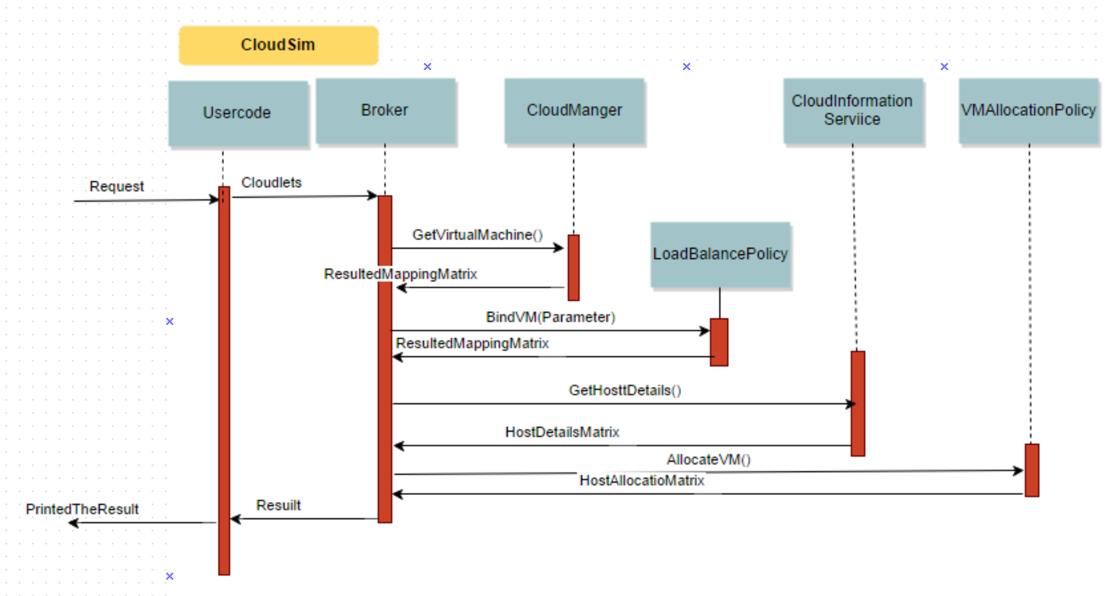

Figure 3. Flow Diagram of the Project

4.1. Task Binding Policy

4.1.1 Objective

Objective of this policy is distributing the task among virtual machines such that all can get the high performance or less execution time. This helps to balance load.

4.1.2 Notation

This paper uses some abstract nation that helps for describing algorithm. Let CloudMatric is the of the size m × n where m is number of virtual machine and n is the number of cloudlet. CloudMatric [p, q] represent the time required to complete task of cloudlet-q by virtual machine–p. CloudMatric is obtains by performing following operation on CloudMatric in sequential order.Operation-1: find minimum value from each row and subtract it from all the element present in that row.Operation-2: find minimum value from each column and subtract it from all element present in that column. LineMatric is one dimensional metric that describes lines that cover all the zeros in reduceMatric. A crossedElement is element in the reduceMatric covered by lines of LineMatric and an uncrossedElement is element in the reduceMatric not cover by lines of LineMatric. Mapping is one dimensional metric that describes which task allocated to which virtual machine for example if value of Mapping[i] is p denotes virtual machine-I assign task-p.

4.1.3 Algorithm: Hungarian Algorithm

Step 1: Initialize CloudMatric.

CloudMatric[i][j] = cloudletParameter / VMparameter

If number of cloudlets and number of number of virtual machines are not equal then add the dummy cloudlets with zero execution time value with any virtual machine in order to make CloudMatric as square metric

Step2: Compute the reduceMatric.

```
For each row
        j=1 to n
        CloudMatric[row][j] = CloudMatric[row][j] – MinimumElement_row[i];

for each column
    I = 1 to n
   CloudMatric[i][column] = CloudMatric[i][column] – MinimumElement_column[i];
```

Step3: Compute the LineMatric

```
line =  compute the _minimum_number_Line()
if( line < VM )
     for I = 1 to n
        for j = 1 to n
          if elements are uncovered
             CloudMatric[i][j] = CloudMatric[i][j] – min;
         if element are covered by two line
             CloudMatric[i][j] = CloudMatric[i][j] + min;
```

If the number of lines is not equal to number of virtual machine than finding minimum from all uncrossedElement subtract it all uncrossedElement and increment by one in execution at all intersection point in reduceMatric. After this go to step3

***Step4**. Find the Mapping*

Find_the_AllPossibleArrangements

Mapping is arrangement for binding cloudlet to virtual machine such that all virtual machine bound with different cloudlet. Suppose three virtual machines are created in cloud computing environment with following specification

	Virtual machine 1	Virtual machine 2	Virtual machine 3
Vmid	0	1	2
MIPS	250	500	250
Size	10000	10000	10000
Ram	2048	2048	2048
Bandwidth	1000	1000	1000
pesNmuber	2	3	2

Suppose three cloudlets are in the queue with broker in cloud computing environment with following specification.

	Cloudlet 1	Cloudlet 2	Cloudlet 3
Id	0	1	2
Length	40000	80000	120000
File Size	300	10000	10000
Output Size	300	2048	2048

Step 1: Initialize CloudMatric.

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	160	320	480
VM 2	80	160	240
VM 3	160	320	480

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	160	320
VM 2	0	80	160
VM 3	0	160	320

Step2: Compute the reduceMatric.

In row number 0,1.2 the minimum values are reduceMatric[0][0], reduceMatric[1][0], reduceMatric[2] [0] respectively. Hence after applying operation 1 the result is:
In column number 0, 1.2 the minimum values are reduceMatric[0][0], reduceMatric[1][1],reduceMatric[2][1] respectively. Hence after applying operation 1 the result is:

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	80	160
VM 2	0	0	0
VM 3	0	80	160

Step3: Compute the LineMatric

LineMatric is one dimensional metric that describes lines that cover all the zeros in reduceMatric.Here two Lines are there, first on cover column-0 element i.e. reduceMatric [0,0], reduceMatric[0][1] reduceMatric[0][2] and second one cover row-1 i.e. reduceMatric[1][0],reduceMatric[1][1]reduceMatric[1][0].

Line number	
Line 1	Column 0
Line 2	Row 1

Number of lines is not equal to number of virtual machines and the minimum of all uncrossed is 80 and intersection positions is ReduceMatric [1][0].After finding minimum from all uncrossedElement subtracting it from all uncrossedElement and incrementing by one in execution at all intersection point in reduceMatric following result will be generated.

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	0	80
VM 2	1	0	0
VM 3	0	0	80

Again this procedure computes the reduceMatric at step 2 and generates following result.

	Cloudlet 1	Cloudlet 2	Cloudlet 3
VM 1	0	0	80
VM 2	1	0	0
VM 3	0	0	80

Step4. Find the Mapping

There are more than one mapping arrangements possibility is there and again finding all possibility is NP complete class .To reduce time complexity this algorithm find randomly any one of all possibility by performing row wise selection.
After performing step 4 the following result will be produced.

Line number	
Virtual machine 1	Cloudlet 2
Virtual machine 2	Cloudlet 3
Virtual machine 3	Cloudlet 1

Hence the final result is after Binding Policy is described by Fig3:

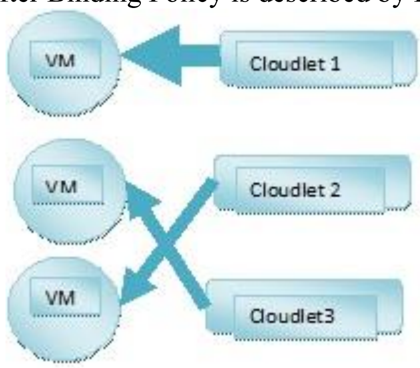

Figure 4. Mapping between Cloudlets and VM

VM Allocation Policy

Objective;The objective of this policy is to utilize the resource by making all processing element for particular host busy.

Notation;UsedPes stores the number of used processing elements for particular host.differenceMatric is one dimensional metric that store different between required pes and free pes for every host in data center.
This paper uses some abstract nation that helps for describing algorithm. VM table is table for storing host with key value .for example <VIT, host1> will mapped string name VIT as key to host 1.freePes is the list that stores the free space or more specifically free pes (Processing Elements) for particular host.

Algorithm for Virtual Machine Allocation

Input: VM Table, Host Table
Output: Updated Host Table and VM table
Step1 Initialize the freePes and usedPes
FreePes, usedPes ← *calculate(host)*
For every VM request do step 4 and 5
Allocate the virtual machine which host satisfying required configuration and host has least difference value in differenceMatric

Step2 find the differenceMatric
for i ← *1 to total number of host*
DifferenceMatric[i] ← *freePes[i]-requiredPes*

End for

Step3 Allocate Vm to host which have least difference Metric

hostId = minimum(DifferenceMatric)

host[hostId] = VM[VMId]

Step4 Update all the metric

Minus required pes from free pes of allocated host and add it too usedPes and compute again freePes, usedPes, and differenceMatric.

for i ← 1 to total number of host

freePes[i] = freePes[i]-requiredPes

End for

Sample code:

```
//This function selects the host with number of free pes very closed
// to required pes for particular vm

 for(int i=0;i<freePesTmp.size();i++){
   if(freePesTmp.get(i)-requiredPes >= 0){
     if (freePesTmp.get(i)-requiredPes < moreFree1){
          moreFree1 = freePesTmp.get(i)-requiredPes;
          idx = i;
     }}        }
```

Let five virtual machine creation requests are came to broker and five data center are available with following configuration. (Let assume that Round Robin scheduling policy)

	VM 1	VM 2	VM3	VM 4	VM 5
Vmid	0	1	2	3	4
MIPS	250	250	250	500	250
Size	10000	10000	10000	10000	10000
Ram	2048	2048	2048	2048	2048
Bandwidth	1000	1000	1000	1000	1000
pesNmuber	20	5	10	20	10

	Host 1	Host2	Host3	Host 4	Host 5
Id	0	1	2	3	4
MIPS	250	250	250	500	250
Ram	2048	2048	2048	2048	2048
Storage	1M	1M	1M	1M	1M
Bandwidth	1000	1000	1000	1000	1000
pesNmuber	20	5	10	20	10

(*M means 1 million)

For vm1

Step1 Initialize the freePes and usedPes

	Host1	Host2	Host3	Host4	Host5
freePes	30	50	15	20	10

	Host1	Host2	Host3	Host4	Host5
UsedPes	0	0	0	0	0

Step2 Find the differenceMatric

	Host1	Host2	Host3	Host4	Host5
differenceMatric	10	30	0	10	0

Step3 Allocate Vm to host which have least
Hence vm1 will allocated to host 3
Step4 Update all the metric

	Host1	Host2	Host3	Host4	Host5
freePes	30	50	5	20	10

	Host1	Host2	Host3	Host4	Host5
UsedPes	0	0	10	0	0

Likewise VM-2 will allocated to host-3 and VM-3 will allocated to host-5 and VM-4 will allocated to host-4 and VM-5 will allocated to host-1.
The final result will be:

	Host1	Host2	Host3	Host4	Host5
freePes	10	50	0	0	0

	Host1	Host2	Host3	Host4	Host5
UsedPes	10	0	15	20	10

Utilization: It the ration of used pes to total pes. The following graph shows resource utilization for all hosts. This is described in Fig 4.

5. Experiment

This load balance policy and vm allocation policy are modeled and simulated in CloudSim with machine containing window 7 os, 2 gb ram and 160gb hard disk and with dual core processor and netbean IDE 7.4 used . With configuration described above in this paper, for vm, cloudlet and hosts the output of the CloudSim is described in Fig4.

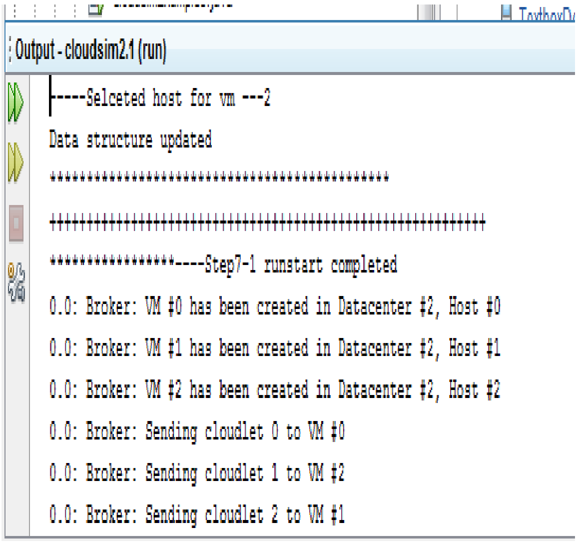

Figure 5. Result after Modeling Policy in CloudSim

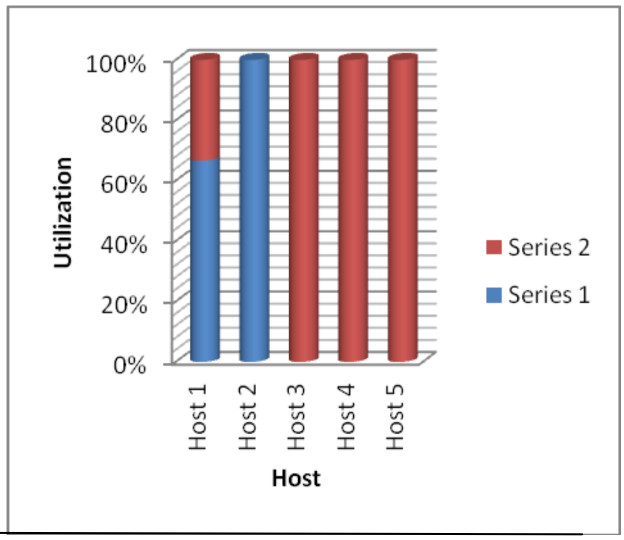

Figure 6. It Describes Utilization of Different Hosts in CloudSim

6. Conclusion and Future Work

CloudSim is simulation toolkit provides modeling and simulation in cloud computing environment by providing way to modeled and to simulate varies modeling policies in this environment. In cloud computing environment, all the resources are pooled in single place with shared memory architecture called Data Center. Broker has list of requested cloud application with high workload by cloud consumer, In order to deploy them cloud computing uses virtualization as the focus element of this environment. Because cloud computing is worked on principal "pay as you used" and the virtualization is limited the actual physical hardware, Utilization of the cloud resources plays ,allocation of virtual machines and proper load balance policy play major role in cloud computing environment. VM allocation algorithm allocates the virtual machines to the host of the data center which have amount of free pes close to the amount of pes required by virtual machine. The task allocation algorithm act as load balance policy and provides way to bind cloudlets (task) to virtual machines such that each task gets high speed of less execution time to complete the task. This algorithm provides fair distribution of cloudlets among the virtual machines. In these both algorithms are modeled and simulated in CloudSim toolkit and analyzed the output.

In future proposed policy can be implemented with consideration of various parameters like space in more dynamic fashion with respect to load generated in cloud environment.

References

[1] "Virtualization and Cloud Computing: Optimized Power, Cooling, and Management Maximizes Benefits Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control", First International Conference on Information Science and Engineering, **(2010)**.

[2] J. Y. Li, M. K. Qieu, J. W. Nieu, Y. Chen and Z. Meing, "Adeptive Resorce Alelocation for Preemptble Jobs in Cloud Systems", 10th International Conference on Intelligent System Design and Application, **(2011)**.

[3] P. T. Endo, "Resourc alocation for distrbuted cloud: Concept and Research challenges", IEE, pp. 42-46.

[4] J. Y. Shei, M. Taiefi and A. Khreisheah, "Resource Planing for Paralel Procesing in the Cloud", IEEE 13th International Conference on High Performance and Computing, **(2011)**.

[5] S. Majumdar, "Resource Management on cloud: Handling uncertainties in Parameters and Policies", CSI Communication, **(2011)**, pp.16-19.

[6] B. Oiza, "A Proposed Serviece Broker Stratagy in CloudAnalyst for Cost-Effactive Data Centre Selection Dhavael Limbeani", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622www.ijera.com, vol. 2, no. 1, **(2012)**, pp. 793-797 793.

[7] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, "CloudSim: A Toolkit for the Modeling and Simulation of Cloud Resource Management and Application Provisioning Techniques".

[8] C. E. L. Duemitresecu and I. R. Fostera, "GangSim: a simulator for grid scheduling studies", Proceedings of the IEEE International Symposium on Cluster Computing and the Grid, **(2005)**.

[9] "Scheduling distributed applications: the SimGrid simulation framework", Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, **(2003)**.

[10] R. Buyya and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing.Concurrency and Computation", Practice and Experience, Wiley Press, vol. 14, no. 13-15, **(2002)**.

[11] I. Foster, C. Kesselman and M. Kaufmann, "The Grid: Blueprint for a New Computing Infrastructure", **(1999)**.

[12] S. Chaisiri, B. S. Lee and D. Niyato, "Optimal virtual machine placement across multiple cloud Providers", In: Services computing conference, APSCC, IEEE Asia-Pacific, **(2009)**.

[13] M. E. Frincu and C. Craciun, "Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments", Fourth IEEE international conference on utility and cloud computing, **(2011)**.

[14] J. Tordsson, R. S. Montero, R. M. Vozmediano and I. M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers", Future Gener Comput Syst., vol. 28, no. 2, **(2011)**, pp. 358–367.

[15] R. N. Calheiros, R. Ranjan and A. Belo, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms".

[16] R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities".

[17] H. W. Kunh, "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, vol.1, pp. 83 -97 1955.

Authors

Kushang Parikh is currently persuing his master degree in computing science and engineering from VIT University vellore 632014.He has completed his Bachelor in field of information technology. His area of interest is in field of algorithm and cloud computing. He had published paper in IEEE conference on cloud computing .He had done projects on security in cloud computing, data mining and on web services.

Nagesh Hawanna is currently persuing his master degree in computing science and engineering from VIT University vellore 632014.He has completed his Becholore in field of computer science.His area of interest is in field of web service and cloud computing.He has done project on antology, pervasive coputing and on security in computer network.

Haleema (M.C.A., M.Phil,, M.Tech) is an Assistant Professor (Senior) in the School of Social Sciences and Languages and pursuing her Ph.D. research work in the School of Computing Science and Engineering. Her area of research is "Software Agent based computing".

R. Jaya Subalakshmi is an Assistant Professor in the School of Computing Science and Engineering at VIT University, Vellore-632014, Tamil Nadu, India. She did M.S.(By Research) in VIT University. Her research area is Cryptography, Data Privacy and Agent based Distributed Computing.

Dr.N.Ch.S.N. Iyengar (b 1961) currently Senior Professor at the School of Computing Science and Engineering , VIT University, Vellore-632014, Tamil Nadu, India .He had 30 yrs of teaching experience. His research interests include Agent-Based Distributed secure Computing, Intelligent Computing , Network Security, Cloud Computing and Fluid Mechanics. He has authored several textbooks and had nearly 172 research publications in reputed peer reviewed International Journals. He delivered many keynote /invited lectures and served as PCM//TCM/reviewer for many International Conferences. He is Editor in Chief for International Journal of Software Engineering and Application (IJSEA) of AIRCC, Guest Editor for SI on Cloud Computing and Services of *Int'l J. of Communications, Network and System Sciences* and Editorial Board member for International Journals like **IJAST of SERSC, IJConvC of Inderscience** and many more.