EFFICIENT TWO LEVEL SCHEDULING IN CLOUD COMPUTING ENVIRONMENT

R. Vijaya lakshmi*

Mrs. Soma Prathibha**

Abstract:

Cloud computing is an emerging IT paradigm. "Cloud computing" relies on *sharing computing resources*, rather than having local servers or personal devices to handle applications. In other words, "Cloud computing" is the general term for anything that involves delivering hosted services through the internet. A cloud has three main characteristics, viz, it is sold on demand, it is elastic and the service is managed by the provider. Cloud vendors offer various levels of services to the users. Cloud vendors who offer Infrastructure as a Service(IaaS), allow users to create Virtual Machines. These virtual machines are used to run the tasks submitted by the user. The resource allocation for each Virtual Machine is defined by the user and the Virtual machines are created on a physical machine, located at a remote place. Proper management of the available resources is essential for reducing the operating costs in a cloud. This paper focuses on a resource allocation problem that aims to minimize the operational cost in cloud computing. Total cost can be reduced by optimal allocation of Virtual Machines to hosts. An algorithm is also proposed so as to accurately map the Virtual Machines to hosts, thereby reducing the total cost and the same has been simulated using CloudSim. The results reveal that the proposed algorithm is very much cost-efficient.

Keywords: *Cloud computing, IaaS, Virtual Machine, host, resource allocation, Cloudsim.*

* Student, Dept. of Information Technology, Sri Sairam engineering college, Chennai

** Assistant Professor, Dept. of Information Technology, Sri Sairam engineering college, Chennai

Introduction:

Virtualization separates the computing functions and technology from the physical hardware. This concept of virtualization forms the backbone of cloud computing. Virtualization allows the users to access the remote storage or servers, without knowing specific server or storage details. The virtualization layer will execute user request for computing resources by accessing appropriate resources. Since cloud computing involves dynamic provisioning of IT services from third parties over a network, the concept of virtualization is used here.

Cloud computing environment:

The cloud computing environment is classified into three basic types:

- Infrastructure as a Service(IaaS)
- Platform as a Service(PaaS)
- Software as a Service(SaaS)

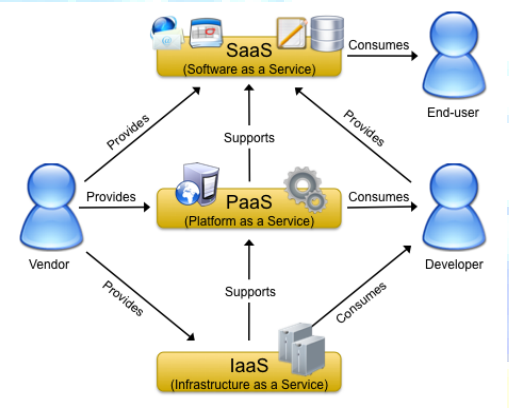

Fig 1: Cloud computing environment

1) IaaS:

IaaS vendors provide their virtual servers and storage as a service. Cloud computing allows a company to pay only for only what they use. This service is mainly used by the administrator.

Example: Amazon Web Services.

2) PaaS:

PaaS vendors provide a set of software and product development tools as a service. Developers create applications on the provider's platform over the internet. This service is mainly used by the developers.

Example: Google App Engine.

3) SaaS:

SaaS vendors provide the hardware infrastructure, the software product and interacts with the user through front-end portal. This service is mainly used by the end-users. Services range from web-based email to inventory control and database processing.

Example: Google Docs.

CloudSim:

Simulation is a technique where a program models the behavior of a system by calculating the interaction between its different entities. CloudSim offers the following novel features:

- Support for modeling and simulation of large scale cloud computing infrastructure.
- A self-contained platform for modeling datacenters, datacenter brokers, scheduling and allocation policies.
- Availability of virtualization helps in creation and management of services.

A. Classes used:

The following classes of CloudSim architecture have been used in our proposed work:

Cloudlet: Cloudlet class models all the application services. Furthermore, CloudletScheduler is used to implement the policies to determine the processing power share among many Cloudlets in a virtual machine.

Datacenter: This class is used to model the cloud based hardware or infrastructure services. Furthermore, the DatacenterCharateristics class is used to provide information regarding the data center resources.

Host: This class is used to model physical resources such as it is used to assign the policy to share the processing power, memory and bandwidth among various virtual machines.

Virtual Machine(VM): This class is used to model a virtual machine in the cloud, which

includes allocation of all the virtual machines on the hosts, which is selected based upon the requirements for a VM deployment. This class is also responsible for allocation of processor cores among the VMs.

B. Modelling the allocation of VM to host:

Cloud computing offers virtualized datacenters to users as a service. Organizations can use cloud platforms to outsource their IT infrastructure, resulting in reduced CApital Expenditure(CAPEX) and Operating Expenditure(OPEX).Since cloud providers offer pay-per-use pricing schemes, proper allocation of resources is necessary. CloudSim supports VM scheduling at two levels:

1) At Host level:

At the host level, it is possible to specify how much of the overall processing power of each core in a host will be assigned to each VM.

2) At VM level:

At VM level, the VMs assign specific amount of available processing power to the individual task units that are hosted within it.

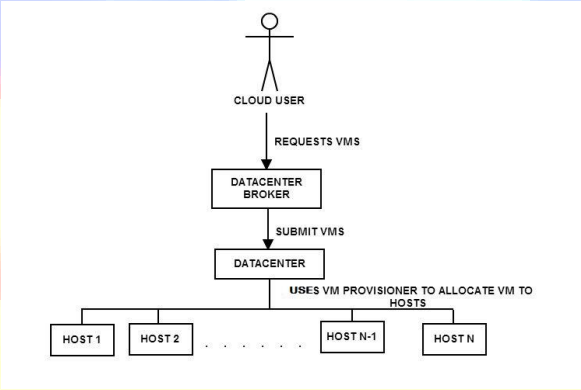

Fig 2: Architecture for allocation of VM to host

Existing model:

The default policy implemented by the VM Provisioner is a straightforward policy that allocates a VM to the Host in First-Come-First-Serve (FCFS) basis.

FCFS VM allocation policy works as follows:

- Cloud user requests for VMs.
- The VMs request list is passed on to the Datacenter Broker.

- The VM list is then passed to VM Provisioner for the placement of VM on the host.
- One by one VM in the series in the list is passed on to create VM on the host.
- The VM Provisioner determines the allocation policy for allocating VM to host.
  - If first host is free, then the VM will be created on the respective host.
    - If next host is free, VM will be created on that host.
    - Otherwise, the VM will be sent to the subsequent hosts.
  - If no host is free, VM cannot be created.

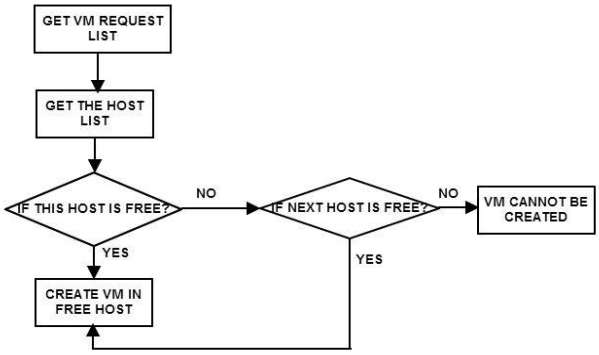

Fig 3: Flowchart describing allocation of VM to host using FCFS

The main drawback of FCFS is that the resource utilization rate is very less and the total cost also increases. Distributed resource management is one of the most challenging problems prevailing in the resource allocation field. Cloud providers have to ensure that they can be flexible in their service delivery to meet the requirements of their customers, while keeping the customers isolated from the underlying infrastructure. They must also offer the service at an affordable price. Hence there is an urgent need for an optimal resource allocation algorithm, which can reduce the cost, thereby helps in optimal allocation of VMs to hosts.

Proposed model:

In CloudSim the default allocation algorithm for allocating tasks to VMs and for allocating VMs to Hosts is FCFS(First Come First Served). Since this type of allocation does not help in optimal utilization of the available resources, the following allocations are proposed.

A. Allocation of tasks to VMs:

In order to allocate the task to the VMs efficiently, we use the concept of sending Shortest Cloudlet to Shortest Processor(SCFP). Here,

- The cloudlets are sorted in increasing order of length.

- The processors are sorted in the decreasing order of processing power.
- Map the cloudlets from the sorted list of processors one by one.

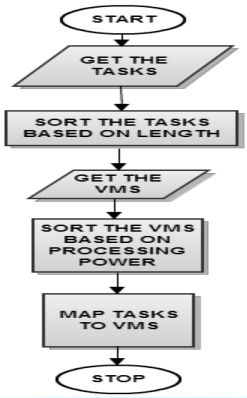

Fig 4: Flowchart describing SCFP

Algorithm:

if this.cloudletlength<next.cloudletlength then

store this.cloudlength in a temporary variable

swap the cloudletlength

end

if this.propower>next.propower then

store this.propower in a temporary variable

swap propower

end

Map the cloudlets to corresponding processors.

B. Allocation of VMs to hosts:

To allocate VMs to hosts appropriately, the following steps are followed:

- Cloud user requests for VMs.
- The VMs request list is then passed on to the Datacenter Broker.
- The VM list is then passed to VM Provisioner for the placement of VM on the host.
- The VM Scheduler determines the allocation policy for allocating VM to host.

➢ Get the VM list requested by the user.

- Compare the number of processing elements (pes) in the VM list with that of the Host list.
- Create VM by assigning the VM to the host, which has the same number of free processing elements.
- If the host with nearly exact number of processing elements is not found, then allocate the VM to the host which best fits the processing element.
- Otherwise, VM will not be created on the host.

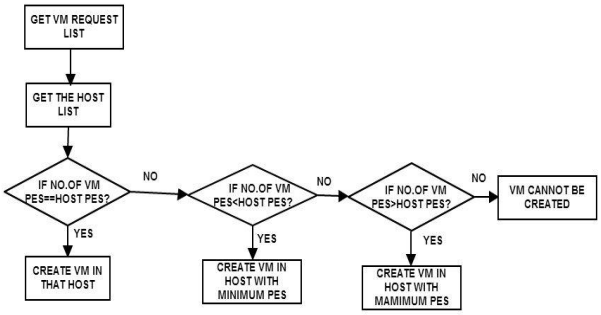

Fig 5: Flowchart describing allocation of VM to host using proposed model

Algorithm:

Input: User requests VMs

Get the user request as VM list in the variable vmlist.

Submit the vmlist to the datacenter broker for creating VMs using the function submitvmlist().

Extend the class VmAllocationPolicy to compare the pes of VM with host and for optimal allocation.

for each VM in the vmlist do

begin

if numberofpesrequested == numberoffreepesinhost **then**

allocate that host to the VM

elseif numberofpesrequested<numberoffreepesinhost **then**

allocate the host to the VM which best fits the requested pes(a host with minimum free pes).

elseif numberofpesrequested>numberoffreepesinhost **then**

allocate the host to the VM which best fits the requested pes(a host with maximum free pes).

if no host is found free **then**

VM cannot be created

end

end

Output: The VM will be allocated to the host which best fits its number of processing elements.

Experimental results:

The proposed model is implemented using CloudSim using Java programming language. The algorithm is tested by creating 16VMs with different processing elements. Five hosts are created with different processing elements. Two datacenters are used in which the hosts are created. 40 cloudlets(tasks) of varying length were to be assigned to these VMs and the VMs must run on appropriate hosts. The cloudlets and VMs are sorted in such a way that the smallest cloudlet is sent to the fastest processor.

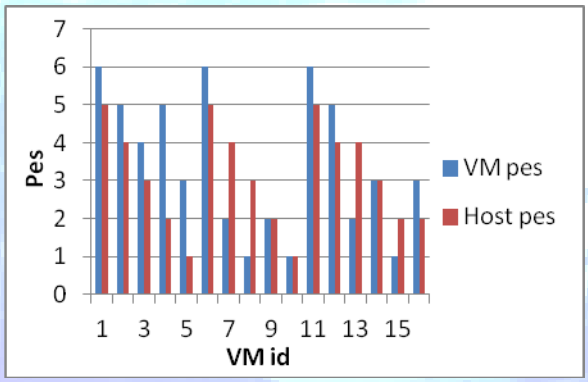

Fig 6: Allocation of VMs to Hosts using FCFS

From Fig 6, we could find that VMs 2,3,4,5,16 requires more processing elements but the existing FCFS algorithm allocates these VMs to hosts with lesser number of processing elements. VMs 7,8,13,15 requires only less processing elements, but this algorithm allocates these VMs to hosts with more number of processing elements, which is not needed for them.

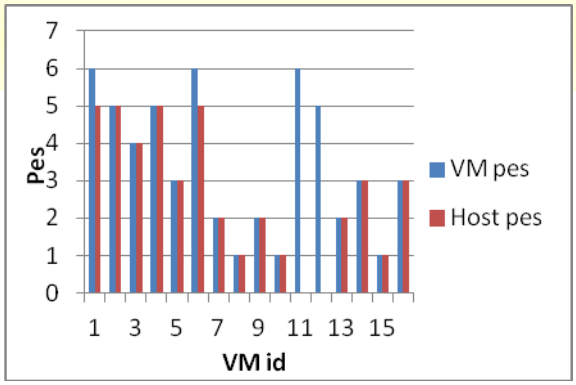

Fig 7: Allocation of VMs to Hosts using the proposed algorithm

From Fig 7, we could find that the VMs are allocated to the hosts which has the same processing elements as they have. If the host with same number of processing elements is not found, then the host with nearly same number of processing elements is allocated to the VM. For example, VMs 1,6 requires 6 processing elements. But the maximum number of processing elements available with the host is only 5. Hence these VMs are allocated to the host with 5 processing elements.

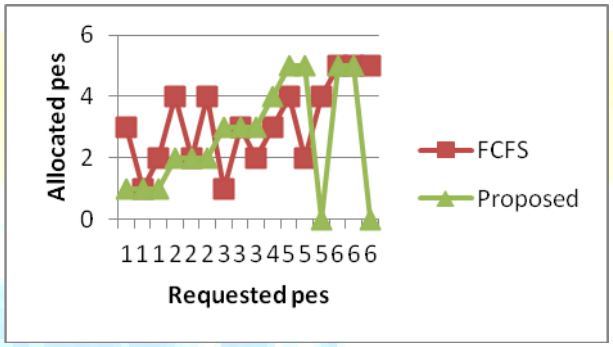

Fig 8: Comparison of existing and proposed models

From Fig 8, we could find the difference in allocation between the existing system and the proposed system. Though the user requests for creating 16VMs, only 14VMs are created and all the tasks are executed using those 14VMs in a very efficient manner. The proposed algorithm allocates VMs to hosts based on the load, thereby avoiding unnecessary creation of VMs. Since unnecessary VM creation is avoided, there is some reduction in the cost the cloud user has to pay for the service offered by the cloud service provider.

TABLE I: COST USING FCFS

Datacenter id	Cost
1	12307.2
2	4102.4

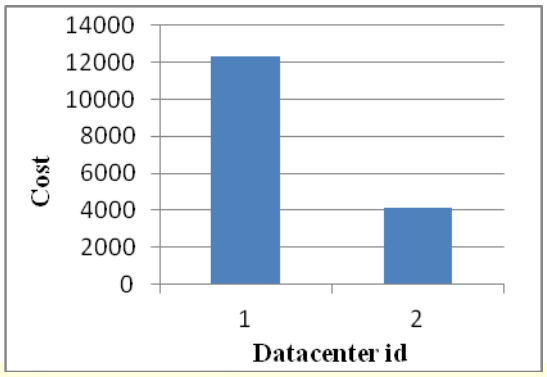

Fig 9: Cost analysis- using FCFS

TABLE II: COST USING THE PROPOSED ALGORITHM

Datacenter id	Cost
1	11281.6
2	3076.8

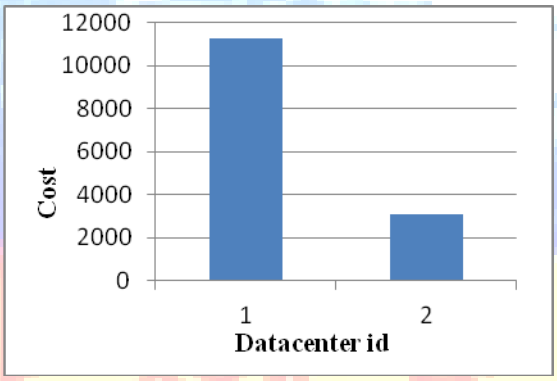

Fig 10: Cost analysis- using the proposed algorithm

From the cost analysis graphs, we could find that the cost incurred by the cloud user has reduced by Rs.2051.2.

Conclusion:

Cloud enables the users to pay for the utility services. Users are required to pay for access to the services based on the service they have utilized. VM allocation is considered as an important issue in IaaS environment of the cloud computing as the placement of the VMs on the hosts can have an impact on the utilization cost. In this paper, we have considered that the VM may be placed on the host, in such a way that the number of processing elements requested by the host is same as the number of free processing elements available in the host. The experimental results

show that by using this policy, the cost incurred also reduces. End users will benefit from the decreased prices for their resource usage.

References:

- Flavio Lombardi , Roberto Di Pietro, "Secure virtualization for cloud computing", Journal of Network and Computer Applications Appl (2010), doi:10.1016/j.jnca.2010.06.008.
- Brian de Haaff, "Cloud computing -- the jargon is back!", Cloud Computing Journal, August 2008.
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No. UCB/EECS-2009-28, February 10, 2009.
- Archie Hendryx, "Cloudy Concepts: IaaS, PaaS, SaaS, MaaS, CaaS & XaaS, Clarifying the meaning of IaaS, PaaS, SaaS, MaaS, CaaS & XaaS",The cloud computing journal on November 1, 2011.
- Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.
- Jerry Gao ,Xiaoying Bai, and Wei-Tek Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice" , Software engineering : an international Journal (SeiJ), Vol. 1, no. 1, September 2011.
- Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros,"Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities", Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE Press, New York, USA), Leipzig, Germany, June 21-24, 2009.
- Anne M. Holler ,"The Green Cloud: How Cloud Computing Can Reduce Datacenter Power Consumption", SustainIT'10 workshop, VMware.

- Qi Zhang, Lu Cheng, Raouf Boutaba, " Cloud computing: sate-of-art- and research challenges", Published online: 20th April 2010, Copyright : The Brazillian Computer Society 2010.
- Hadi Salimi, Mahsa Najafzadeh, and Mohsen Sharifi. , "Advantages, Challenges and Optimizations of Virtual Machine Scheduling in Cloud Computing Environments",International Journal of Computer Theory and Engineering Vol. 4, No. 2, April 2012.
- Shailesh S. Deore, A. N. Patil, Ph.D, Ruchira Bhargava , "Energy-Efficient Job Scheduling and Allocation Scheme for Virtual Machines in Private Clouds", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.1, January 2013 .
- K. Mills, J. Filliben and C. Dabrowski,"Comparing VM-Placement Algorithms for On-Demand Clouds", Information Technology Laboratory,NIST,USA.
- Albert Greenberg, James Hamilton, David A. Maltz, Parveen Patel ,"The Cost of a Cloud: Research Problems in Data Center Networks", Microsoft Research, Redmond, WA, USA.
- K L Giridas and Shajin A Nargunam, "Optimal Resource Allocation Technique (ORAT) for Green Cloud Computing",International Journal of Computer Applications 55(5):20-26, October 2012, Published by Foundation of Computer Science, New York, USA.
- Shaminder Kaur,Amandeep Verma, " An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment", published in I.J. Information Technology and Computer Science, 2012, 10, 74-79 Published Online September 2012 in MECS DOI: 10.5815/ijitcs.2012.10.09.