October 2013

EFFICIENT ALLOCATION OF VIRTUAL MACHINE IN CLOUD COMPUTING ENVIRONMENT

SONAM RATHORE

MITS, Lakshmangarh, sonamrathore18@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijcsi>

Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

RATHORE, SONAM (2013) "EFFICIENT ALLOCATION OF VIRTUAL MACHINE IN CLOUD COMPUTING ENVIRONMENT," *International Journal of Computer Science and Informatics*: Vol. 3 : Iss. 2 , Article 15.

DOI: 10.47893/IJCSI.2013.1135

Available at: <https://www.interscience.in/ijcsi/vol3/iss2/15>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

EFFICIENT ALLOCATION OF VIRTUAL MACHINE IN CLOUD COMPUTING ENVIRONMENT

SONAM RATHORE

Faculty of Engineering and Technology, MITS, Lakshmangarh, India
e-mail:sonamrathore18@gmail.com

Abstract: Cloud computing is a latest new computing paradigm where applications, data and IT services are provided over the Internet. Computing service is provided on demand as a utility as other utilities. Cloud computing provides dynamic provisioning of computing services. Resource scheduling is a key process for clouds such as Infrastructure as a Service cloud. To make the most efficient use of the resources, an algorithm must be used which improves the utilization of system resources. In this work an algorithm is used which uses all the combination of allocation sequence and chooses the allocation sequence on the basis of strength of allocation. Experimental shows that the proposed algorithm gives the better utilization of resources.

Keywords: Cloud Computing; Virtual Machine; Allocation

1. INTRODUCTION

Cloud computing is a paradigm that is rising in this world of technologies. Cloud computing provides shared pool of resources on-demand over network on pay-per use. Cloud computing ensures access to virtualized IT resources that are present at the data center and are shared by others. Cloud computing can be broadly classified into three services : Infrastructure as a Services (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). These services are provided over a network and accessible across computing technologies, operations and business models.

For the Infrastructure as a Service (one of the layers of cloud stack), one of the significant issues is the scheduling of virtual resources and virtual machines (VMs). It has been widely accepted that virtual machines can be employed as computing resources for high performance computing. Thus, efficient virtual machine allocation is essential in cloud computing environment for increasing resource utilization and efficient deployment of applications in virtual machine .

There are some popular open-source cloud systems, such as Eucalyptus, Open Nebula, and Nimbus, to decide the allocation of resources. To deal with the problem of allocating VM instance request to available computing nodes , Eucalyptus uses Greedy (first fit) or Round robin algorithm, with GREEDY, the first node which can meet the IRs will be chosen. The ROUNDROBIN query all the nodes in circular order, until find the fitted node [1] [2]. The OpenNebula default scheduler provides a rank scheduling policy that places VMs on physical resources according to a ranking algorithm that is highly configurable by the administrator, and relies on real-time data from both the running VMs and available physical resources [3][7]. Nimbus uses some customizable tools like PBS and SGE. PBS is a

queuing system and SGE uses Job Scheduling Hierarchically (JOSH) [4][5]. However, all of these algorithms fail to achieve higher VMs utilization rate. Therefore an algorithm must be used which is efficient in allocating VMs instance request and hence increases the resource utilization.

2. MOTIVATION AND PROBLEM DEFINITION

IaaS layer of cloud computing serves as a foundation for the other two layers (i.e PaaS and SaaS), for their execution therefore we have focused on the IaaS. IaaS deliver computer infrastructure - typically a platform virtualization environment - as a service. Efficient scheduling of VMs instance request which meet user's requirements and improve the resource utilization increases the overall performance of the cloud computing environment. VM instance scheduling in IaaS is the one of the crucial cloud computing questions to address.

Suppose, „M“ physical machines are available and their resource capacities given along memory, CPU and hard disk dimensions. There are „N“ virtual machines to be placed. The requirements of these virtual machines are given along the dimensions of memory, CPU and hard disk. We have to find allocations of VMs on available physical machine that satisfies the VMs' resource requirements and increases overall the resource utilization. The objective of this research work is to introduce an algorithm for efficient VM scheduling in cloud computing in terms of resource utilization rate. The proposed algorithm is compared with other existing algorithm for VM instance allocation.

3. DESIGN MODEL OF VM ALLOCATION IN CLOUD COMPUTING ENVIRONMENT

Figure 1 gives an architecture of VM allocation in cloud computing. Cloud users or the Cloud consumers request for the VMs and ask for VM requirement to the Cloud broker. The Cloud broker or data center broker acts on behalf of the broker, looks for the Cloud service providers who can make VM request fulfilled by querying the Cloud Information Service (CIS). Every new resource should be register to the

CIS. Once the Cloud service provider has been chosen by Broker, it submits the VMs list to the Datacenter. Datacenter holds the physical computing servers i.e hosts and VMs place on host on the basis of allocation policies decided by the Cloud service provider.

Where, each x_{ij} indicates whether VM instance i is placed on the computing node j .

x_{ij} will be 1 if VM i is placed on computing node j , otherwise 0.

In (1) if $k=1$ it represents CPU Core, if $k=2$ it represents memory and $k=3$ represents hard disk capacity. Value of P_k depends on value of $vm_k/node_k$, so when value of P_k is „a“ then placement of instance on node is right fit and we get a maximum usage of resources. When value of P_k is „b“ then it means that the placement only get a suboptimal solution, it cannot get a maximum utilization of the resource. In both cases we assign a positive value to P_k , what's more, „a“ bigger than „b“ so that if the placement is right fit, the solution will be encouraged. While the value of $vm_k/node_k$ is bigger than one which means the placement absolutely not fit, we assign a negative value to P_k .

Figure 1. Architecture of VM allocation

3.1 Proposed Model

Let us consider a set of VM requests, a set of interconnected computing nodes connected by LANs. The computing nodes are different kinds of ordinary PCs,

servers, and even high performance clusters. And cloud provides all kinds of machines it possesses in forms of virtual machine that clients can visit it through Internet as a service. In this work, we take the number of CPU cores, Memory capacity and Hard Drive capacity in consideration, which most of the existing IaaS cloud systems do.

Assumed there are n VM instance requests (IRs) and m idle computing nodes available in the cloud. Now the problem is to find the allocation sequence which makes the utilization rate of the resource achieve maximum. The problem can be formulated as

$$\text{Strength of allocation} = \sum_{j=1}^m \sum_{i=1}^n \sum_{k=1}^3 P_k A_{ij} \quad (1)$$

where,

$$P_k = \begin{cases} a & \text{if } vm_k / node_k = 1 \\ b & \text{if } vm_k / node_k < 1 \\ c & \text{if } vm_k / node_k > 1 \end{cases}$$

Where $X \geq 1000$

$$\text{Strength of allocation} = \sum_{j=1}^m \sum_{i=1}^n \sum_{k=1}^3 (P_k + X) A_{ij}$$

In *Strength of allocation* any big number is added to P_k to make it positive.

So, the allocation sequence with largest *Strength of allocation* is considered for VM allocation on available hosts in datacenter.

3.2 Flowcharts

Figure 2 and Figure 3 gives the flow of FCFS and proposed algorithms for VM allocation respectively.

- FCFS VM allocation policy works as follows:
- Cloud user requests for VMs.
- Cloud Broker which acts on behalf of the Cloud user submits the VMs request list.
- An event has been generated for the submission of the VMs request list to the data center.
- One by one VM in the series in the list has been passing to create VM on the host If not found host with enough of PEs required by VM then VM cannot be created.
- If found first host with less number of PEs in use and fulfils VM PE requirement:
 - If host has enough RAM to fulfil the VM's RAM requirement then place VM on host.
 - Else, VM is not created on the host. This VM is again sent to create a new list of VMs.

VM allocation using proposed algorithm works as follows:

- Cloud user requests for VMs.
- Cloud Broker which acts on behalf of the Cloud user submits the VMs request list.
- An event has been generated for the submission of the VMs request list to the data center.

- The VM list is passed for the placement on the available host.
- Permutations of placement order of VMs on available hosts are considered.
- Strength of allocation is calculated for all the allocation sequence taken into consideration the VM's parameters.
- Allocation sequence with the highest strength of allocation value is considered for the VM allocation.

Figure 2. Flowchart of VM allocation using FCFS

Figure 3. Flowchart of proposed algorithm for VM allocation

4. EXPERIMENTAL RESULTS

The proposed model is implemented in JAVA using Netbeans IDE. The implemented algorithm is then integrated with CloudSim package [8] for simulation. The algorithm is tested for different sets of VM instance request and computing nodes. Table I and Table II gives the example of parameters (CPU cores, Memory and Harddisk) for VM instance request and computing nodes.

Table I. VM instance request parameters

VM id	CPU cores	Memory	Harddisk
1	1	512	2
2	2	512	5
3	2	1024	20
4	4	2048	20
5	2	512	10

Table II. Computing Nodes Parameters

Host id	CPU cores	Memory	Harddisk
0	2	512	20
1	2	1024	20
2	2	1024	10
3	4	2048	40
4	4	1024	10
5	4	2048	40

The figure 4 depicts the VM allocation using proposed algorithm for the VM instance request parameter and Host parameter given in table I and II. The horizontal axis represents the hosts with the host ids. Number in the line graph represents the VM_id created on the host with respective host id.

Figure 4. VM Allocation using proposed algorithm

4.1 Simulation result analysis

The figure 5 below depicts the comparison of resource utilization rate using proposed and FCFS VM allocation policy. The horizontal axis is the number of VMs requests and vertical axis represents the average resource utilization rate.

Figure 5. Comparison of resource utilization rate using proposed algorithm and FCFS

Figure 6 depicts the time taken by algorithm to execute. The horizontal axis represents the No. of VMs requests and vertical axis represents the execution time taken by the algorithm.

Figure 6. Algorithm's execution time for different number of VMs requests

5. CONCLUSION

IaaS provides provisioning of processing, storage, networks, and other fundamental computing resources over a network. The VM allocation is a major issue in IaaS service of the cloud computing as the placement of these VMs can impact application performance because the IaaS providers are unaware of the hosted application's requirements. Therefore the efficient VM allocation policy must be used. In this work the experimental results shows that proposed algorithm can improve resource utilization by efficient VM allocation.

6. ACKNOWLEDGMENT

We would like to thank Mr. P. K. Bishnoi of MITS, Lakshmangarh (Rajasthan) for his support and guidance.

REFERENCES

1. Daniel Nurmi, Rich Wolski, Chris Grzegorzczak, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov, "The Eucalyptus Open-source Cloud-computing System", 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009, pp: 124-131.
2. Eucalyptus, <http://www.eucalyptus.com>.
3. Open Nebula, <http://www.opennebula.org>.
4. openPBS, <http://pbsgridworks.com>.
5. Nimbus, <http://nimbusproject.org>.
6. Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, Warren Carithers, "Efficient resource management for Cloud computing environments," greencomp, International Conference on Green Computing, 2010, pp.357-364
7. J. Fontan, T. Vazquez, L. Gonzalez, R. S. Montero, and I. M. Llorente, "OpenNebula: The open source virtual machine manager for cluster computing," In Open Source Grid and Cluster Software Conference Book of Abstracts, San Francisco, USA, May 2008.
8. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms" SOFTWARE – PRACTICE AND EXPERIENCE Softw. Pract. Exper. 2011; 41:23-50 Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995