International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.6, December 2012 – www.ijais.org

Effective Technique for VM Provisioning on PaaS Cloud Computing Platform

K S Manjunath and Anirban Basu
Department of CSE, East Point College of Engineering and Technology, Bangalore

ABSTRACT

In a Cloud Computing platform, Virtual Machine Provisioning involves instantiation of one or more Virtual Machines (VMs) that match the hardware and software requirements of an application and plays an important role in performance issues [4]. This paper discusses a VM Provisioning technique to reduce the response time in a cloud computing platform with PaaS type of deployment model when the physical servers hosting Virtual Machines in datacenters vary in their configurations. The method proposed in this paper is application specific and has been implemented on a test bed with VMware environment. Experimental results show that it can deliver better performance compared to random scheduling method for applications with repetitive computations

1. Introduction

A Cloud computing platform has a large pool of resources and provides a development platform which can be reconfigured dynamically to adjust to a variable load for better performance and for optimum resource utilization.Users submit their jobs (or requests for computing resources such as CPU, RAM, disk, application, infrastructure software, etc.) over the network. A Cloud computing platform configures its resources dynamically depending upon the service requirement. Cloud computing platforms offer three types services which are:

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

Examples of IaaS cloud include Amazon EC2, IBM Cloud. Microsoft Azure and Google AppEngine are examples of PaaS cloud. Examples of SaaS include Gmail, Google Docs. In this paper we discuss resource allocation in PaaS type of cloud computing platforms.

An important feature of cloud computer is virtualization of physical resources which enables the execution of multiple jobs on the same, shared physical environment by creating Virtual Machines (VMs). This make a cloud based service economical to both cloud service providers and cloud users. Cloud service providers can reduce their costs of service delivery by resource consolidation through virtualization

As discussed in [7] Cloud Provisioning is the process of deployment and management of applications on Cloud infrastructures. It consists of three key steps: (i) Virtual Machine Provisioning, which involves instantiation of one or more Virtual Machines (VMs) that match the specific hardware characteristics and software requirements of an application.(ii) Resource Provisioning, which is the mapping and scheduling of VMs on physical Cloud servers within a cloud and (iii) Application Provisioning, which is the deployment of specialized applications within VMs and mapping of end-user’s requests to application instances.

After the jobs or requests are provisioned in a cloud, users are typically unaware about the details of execution environment.However, the process of allocating Virtual Machines (VMs) in Clouds is complex [4], and it requires a Virtual Machine Provisioner to compute the optimal configuration of software and hardware to ensure that QoS targets of application services are achieved, while maximizing the overall system efficiency and utilization [5]. Achieving QoS targets is important for meeting Service Level Agreements (SLA) agreed with end-users and for justifying the investment in Cloud based deployments.

In this paper, we discuss a method for VM Provisioning for reducing the response time of an application. This technique is applicable in datacenters where the configuration of physical machines vary i.e., physical machines hosting the Virtual Machines vary in their hardware and software configurations and can be used for applications with repetitive computations.

2. Cloud Provisioning Approaches

Efficient VM Provisioning have been attempted by [4][5][7]. In the work reported in [4] the resources that are available are categorized into two types: opaque and transparent, where opaque resources are Virtual Machines whose details are not known to resource manager while transparent resources are Virtual Machines whose details are known to the resource manager. If the resources are opaque, the resource manager does not know which is the best resource to allocate to an incoming request. It is important to know the details of the Virtual Machines before allocating it for processing the request.

To meet the QoS deadlines the Provisioner requests the resource manager in advance for Virtual Machines [5]. The resource manager then allocates the requested Virtual Machines in the order the requests come to the resource manager. If the Virtual Machines are busy then the requests arriving waits till the Virtual Machines become free with the possibility of violation of the QoS guarantees.

The provisioning techniques used earlier did not consider the response time of Virtual Machines, but considered only the execution time of the Virtual Machines. The response time which includes time to allocate the requests to Virtual Machines, receiving the response and the waiting time for service were not considered in these methods for resource allocation. Response Time varies due to performance capability of the VM, and Traffic on the network

The proposed method allocates Virtual Machines to process the user requests based on response time on Virtual Machines that are available at the time of scheduling and therefore the proposed method gives the best possible performance.

3. Proposed System

In the VM provisioning method proposed here, the response time of each Virtual Machine is determined by running the tasks on all active Virtual Machines and the VM which provides the minimum response times found out for the tasks. This information is used for scheduling when the tasks arrive for execution. As the response time of each Virtual Machine is already known, the tasks are scheduled on the Virtual Machines to give best possible overall response time.

The VM provisioning system (Figure 1) proposed in this paper has two main sub-systems:

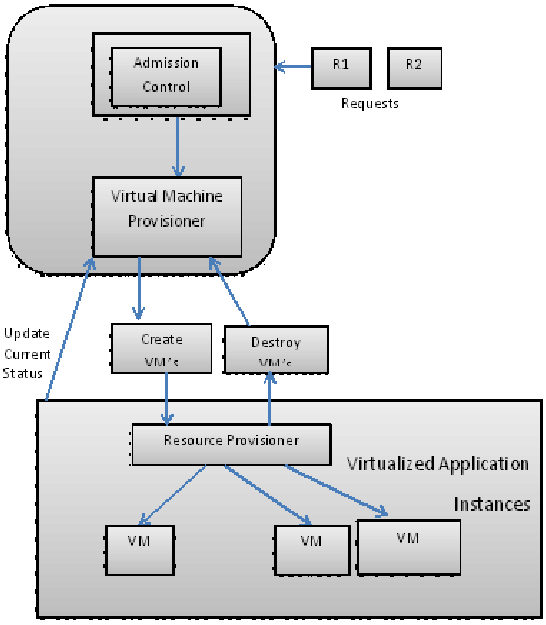

Figure 1 Proposed system for Virtual Machine provisioning

Virtual Machine Provisioner: It is the main point of contact in the system that receives incoming requests and creates VMs. The Admission Control module is responsible for admitting service requests for allocation and maintains a buffer of requests which cannot be serviced immediately on arrival

Resource Provisioner: It allocates Virtual Machines to instances of an application and follows two phases for Resource Provisioning:

Phase 1

In this phase the Response Time Matrix is constructed to maintain data on Response Times of $Task_i$ on all VMs that are active.

It proceeds as follows:

1: Start all the available Virtual Machines

2: Send task $Task_i$ to all the Virtual Machines

3: Measure the response time on all the Virtual Machines and fill up the Response Time Matrix

to find the fastest VM for that task.

Phase 2

Accept the user tasks t_1,t_2,t_3,… from Admission Control queue and schedule the tasks on the best possible VM which is free depending upon the entries in Response Time Matrix

If all the VM's are busy then wait till one of the VMs become free.

4. Performance Analysis

The performance of the proposed VM Provisioning method was measured on an experimental test bed using VMware environment using a water marking application. The technique proposed in this paper was compared with random scheduling method.

4.1 Implementation Environment

To implement this technique, three Virtual Machines were created using VMware Workstation and all the machines were connected using the Virtual LAN of the VMware environment. Provider module is run in VM_1 and Resource module is run in all the Virtual Machines and user module is run on VM_3.

There are three tasks in water marking application:

- Text Water Marking
- Image Water Marking
- Image and Text Water Marking

Before a request is accepted, the Task is sent to all the Virtual Machines that are active (i.e., the VMs that are running). Response times on all active Virtual Machines are determined for each Task and Response Time Matrix filled up based on the response times. When the requests arrive for execution, they are assigned to the Virtual Machines with the lowest response times. If for a Task, the Virtual Machine which provides the lowest response time is busy then the task is assigned to the Virtual Machine with the next best response time and so on, therefore the user tasks are run on the best possible Virtual Machines available at that point of time.

4.2 Experimental Results

4.2.1 Measurement of Response Times of all tasks

Figure 2 shows the response time of 3 VMs that are active. VM_1 is having less response time as compared to VM_2 and $VM_3$for Image and Text Water Marking and Image Water Marking. So VM_1 has got higher priority to process the request sent by user for these two kinds of tasks. For Text Water Marking VM_2 is having the least response time compared to other two Virtual Machines. If VM_1 is busy then when the request is received, then the task is given to VM_3 for processing. If all VMs are busy, then the request sent by the user is buffered till VM_1 is free.

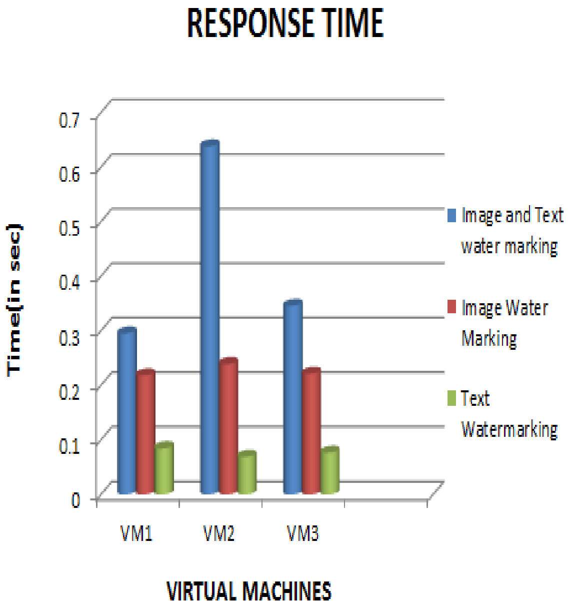

Figure 2: Response time on VM$_1$, VM$_2$and VM

4.2.2. Comparison with Random Scheduling

Proposed method assigns the tasks to Virtual Machines having lowest response time among all active VMs. But a provisioning technique based on random scheduling assigns the task to any one of the available Virtual Machines without any consideration. The performance was measured by both the methods for all the three tasks and results are shown in Figures 3, 4 and 5

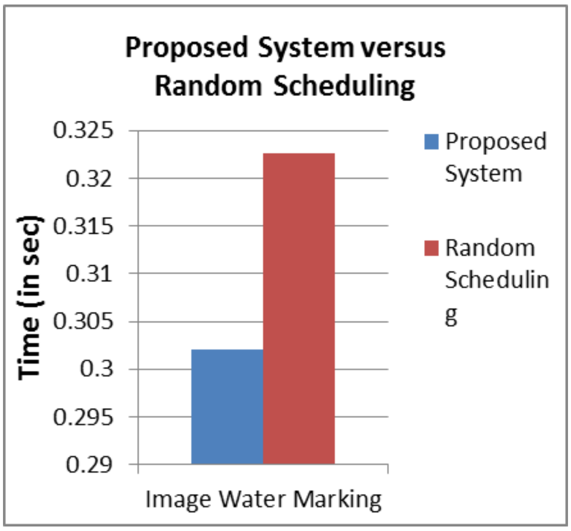

Figure 3: Comparison of Time Taken for Image Water Marking

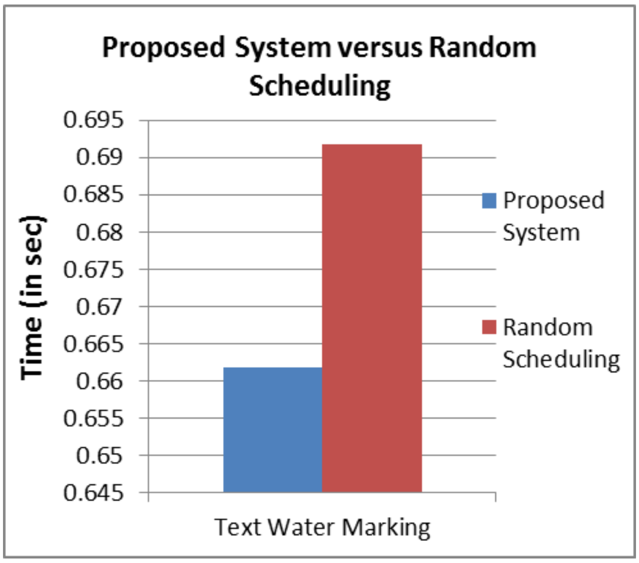

Figure 4: Comparison of Time Taken for Text Water Marking

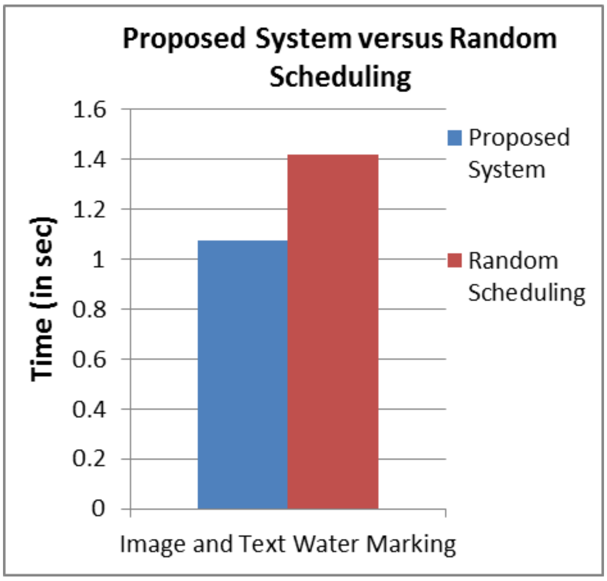

Figure 5: Comparison of Time Taken for Image and Text Water Marking

The above results show the effectiveness of the proposed method compared to random scheduling for all types of water marking application

5. Conclusion

Cloud computing environment offers large pool of resources to the consumer using the services. Virtual Machine provisioning is a decisive factor about how well the resources are utilized. Allocation of resources to PaaS applications is normally done at random without considering the Virtual Machines' capabilities. This results in underutilization of resources and increase in response times.

This paper presents a technique for Virtual Machine provisioning which is deterministic and all attempts are made to schedule Tasks on the VMs to give the best possible performance. Further, in this method requests are never rejected if the Virtual Machines are busy. They are buffered till one of the Virtual Machines is free thereby guaranteeing service to the user.

This method is specific to particular kind of application as the response time is calculated based on the task related to the same application. This approach can be used where the Virtual Machines are allocated to perform repetitive computations as in image processing, performing complex mathematical calculations etc.

6. REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "*Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*," Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[2] Michael Armbrust, Armando Fox, "*Above the Clouds: A View of Cloud Computing,*" Berkeley Reliable Adaptive Distributed systems Laboratory (RADLab).

[3] A. Quiroz, H. Kim, M. Parashar, N.Gnanasambandam, and N. Sharma, "Towards autonomic workload provisioning for enterprise grids and clouds," in Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (GRID'09), 2009.

[4] Shikharesh Majumdar, "*Resource Management in Clouds: Handling Uncertainties in Parameters and Policies*," CSI communications May 2011

[5] Majumdar. S, *"The Any-Schedulability Criterion for Providing QoS Guarantees Through advance Reservation Requests,"* Proceedings of the Cluster Computing and the Grid (International Workshop on Cloud Computing), Shanghai (China), May2009, pp. 490-495

[6] Tharam Dillon, Chen Wu and Elizabeth Chang, *"Cloud Computing: Issues and Challenges,"* 2010 24th IEEE International Conference on Advanced Information Networking and Applications.

[7] Rodrigo N. Calheiros, Rajiv Ranjan and Rajkumar Buyya "Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments" 2011 ICPP