Utility based Resource Provisioning in Cloud Computing

Shraddha Shah[1], Dr. G.R. Kulkarni[2], Richa Sinha[3]
[1]*Computer Engineering, KIRTC, Kalol, Gujarat, India.*
[2]*Principal, KIRTC, Kalol, Gujarat, India.*
[3]*Professor, Information Technology, KIRTC, Kalol, Gujarat, India*

***Abstract*- Cloud computing has provide the innovative solution to the business development world. The mobility of retrieving and storing information enhance the usability of cloud. The two major entities in cloud are: service provider and service user. Service provider provides facilities to the user in terms of allocation of resources, based on the user requirement. There are two major ways of accessing resources: On Demand and Reservation plans. The achievement of best reservation plan is always difficult due to the uncertainty of work load and future plans.This paper discusses the development of new provisioning policies that can take advantage of more information, which can be provided by the scheduling service. When the provisioning is aware of the amount of time in which external resources are required, and also have more information about current and predicted utilization of local infrastructures, it can decide to deploy resources from the resource pool whose resources have the optimal cost in the envisioned situation. This will drive for further reduction in budget expenditure for execution of tasks.**

I. INTRODUCTION

Cloud computing establish inventive ways for business in IT infrastructure and services. It is based on the sharing computing resources, which is seen as evolution of distributed and grid computing [1]. The pool of resources provided by the service provider with charges and service user can utilize it without any installation at local resources. Provisioning of resources in cloud computing is very important for achieving business targets on time with profit. Cloud Provisioning is the process of deployment and management of applications on Cloud infrastructures. It consists of three key steps:

1. ***Virtual Machine (VM) Provisioning,*** which involves instantiation of one or more VMs that match specific hardware characteristics and software requirements of an application. Most Cloud providers offer a set of general-purpose VM classes with generic software and resource configurations. For example Amazon EC2 supports 11 types of VMs, each one with different options of processors, memory, and I/O performance.

2. **Resource Provisioning,** which is the mapping and scheduling of VMs onto physical Cloud servers within a cloud. Currently, most IaaS providers do not provide any control over resource provisioning to application providers. In other words, mapping of VMs to physical servers is completely hidden from application providers;

3. **Application Provisioning,** which is the deployment of specialized applications within Vms and mapping of end-user‘s requests to application instances. Here in this project only VM Provisioning and Application Provisioning are focused on, because these are steps that an application service provider can handle. The goal application provisioning is ensuring an efficient utilization of virtualized IT resources, which can be achieved through the use of techniques such as efficient mapping of requests, while the goal of VM provisioning is to provide applications with sufficient computational power, memory, storage and I/O performance to meet the level of QoS expected by users, this can be achieved by either increasing/ decreasing capacity of deployed VMs or by increasing/decreasing number of application and VM instances.

There are two major resource acquiring policies namely: On demand and Reservation plan. Both serve resources provisioning based on the requirement of the service user. On demand can also be called as spot instance, the service user can book resources through bidding. The resources allocate to the user who successfully win the bidding immediately. In reservation plan, user cans priory book resources based on the requirement and deadline. The uncertainty of future load and unable to fully meet demand deadline for jobs, the under provisioning problem can occur where the reserved resources are unable to fully meet the demand.

Although this problem can be solved by provisioning more resources with on demand plan to fit the extra demand, the high cost will be incurred due to more expensive price of resource provisioning with on-demand plan. On the other hand, if the reserved resources are more than the actual demand the over provisioning problem can occur in which part of a resource pool will be underutilized. The cloud user minimize the total cost of resource provisioning by reducing the on-demand cost and oversubscribed cost of under provisioning and over provisioning. To achieve this goal, the optimal computing resource management is the critical issue.

The problem of provisioning is the motivation for exploring a resource provisioning strategy for the cloud user. In this paper, discuss the algorithm for calculating requirement of extra resources, instead of demanding new resources utilize resource pool for fulfils the requirement.

II. ALGORITHM

For the intent of supporting execution of applications in hybrid Desktop Grids and Clouds, a new provisioning algorithm has been developed that explores Spot Instances. Spot Instances consist of virtual machines (VMs) for which users produce a bid price that represent the maximum value they are willing to pay for using each VM. Amazon then periodically updates resource value (spot price) and, whenever the spot price is equal or smaller than the bid price, corresponding VMs are executed. When the spot price is bigger than the bid price, VMs are terminated. Users are charged hourly at the spot price, and fractions of hours are ignored in case of pre-emption. Therefore, if a VM is pre-empted after 1.5 hours, the user is charged for one hour of utilization. However, if the user terminates a VMs after 1.5 hours, user is charged for two hours of utilization. Spot Instances represent a cheaper type of Cloud resource that can be valuable to speed up applications running in the local Desktop Grid. Algorithm describes Resource Utilization Algorithm. The algorithm is executed when any of the following conditions is observed: (i) a new job is received by the system (ii) a task from a job is queued, and (iii) execution of a task completes. Basically, the algorithm checks if currently available resources are enough for completing jobs within their deadlines. The RequiredTime is the time needed to finish the cloud.The calculation considers only tasks in the waiting queue; therefore, during estimation of remaining execution time the algorithm considers that current running tasks have to complete before the waiting ones are scheduled and executed.

requiredTime ← ([tasks/resources] + 1) × executinTime

One of the key differences between the proposed provisioning algorithm and the previous one is that the former considers the deployment time in public Clouds during calculation of number of extra required resources: values α and β represent, respectively,

α = (timeLeft/ CPUUtilizationTime)

β = (deployTime/ CPUUtilizationTime)

the ratio between remaining time and task runtime and deployment time and task runtime. If $\alpha < \beta$,it is possible to complete the job before its deadline. Otherwise, new VMs will not become available before the deadline, and number of extra resources is calculated based on availability of local resources and estimated utilization of external resources. In either case, the number of external resources to be provisioned is limited to the number of waiting tasks.

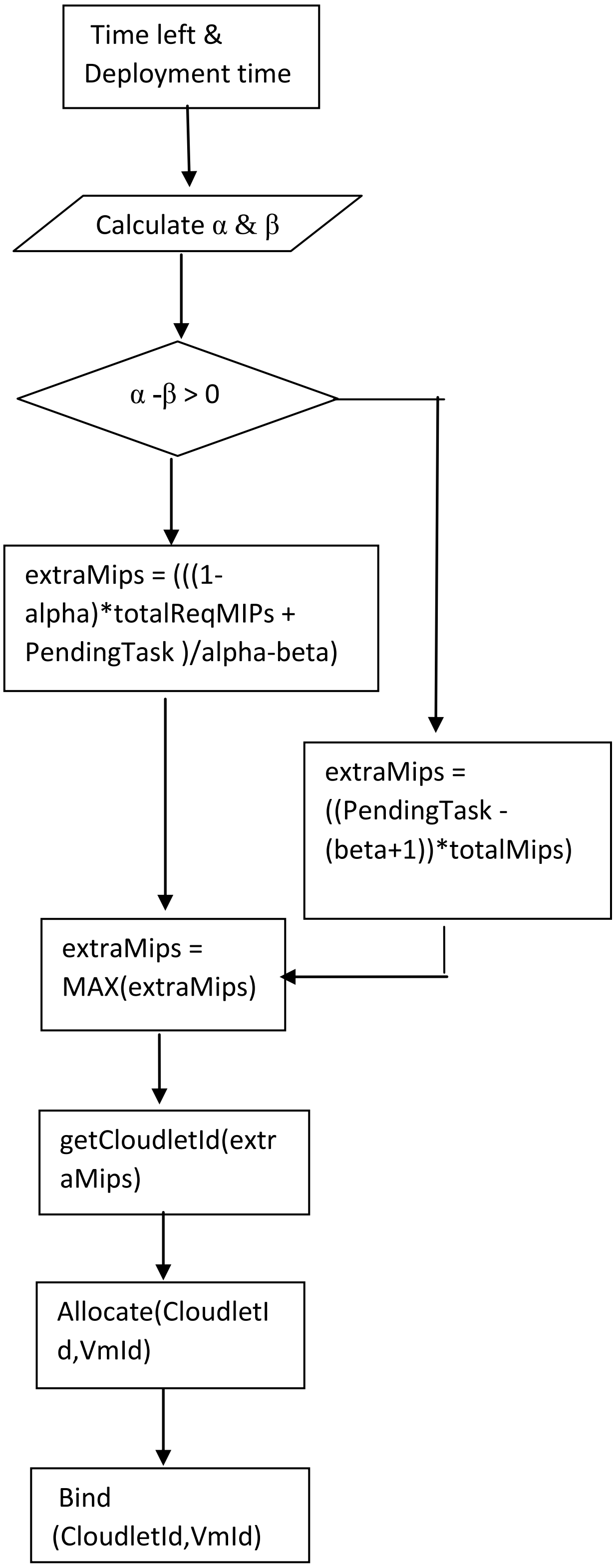
if extraResources > 0 then

requestedResources ← requestedResources + extraResources;

This strategy makes the probability of VM pre-emption small, especially if external resources are required for a small number of hours, at the same time it allows smaller prices than on demand requests. If allocation of spot instances fails (for example, if spot price is higher than the on-demand price), resources can be sought in the on-demand market or in other public

Finally, each time a task completes the algorithm also checks if the job is able to complete within its deadline with one less resource.If so, one resource is returned to the Resource Pool Manager, which can decide between releasing the resource (if the one-hour billing period is about to expire) or allocating it to another running job.

The algorithm for the finding extra Resources and allocation of resources is given in below figure.


Time left &
Deployment time
Calculate α & β
α -β > 0
extraMips = (((1-alpha)*totalReqMIPs + PendingTask)/alpha-beta)
extraMips = ((PendingTask -(beta+1))*totalMips)
extraMips = MAX(extraMips)
getCloudletId(extraMips)
Allocate(CloudletId,VmId)
Bind (CloudletId,VmId)

Flow Chart : Resource Allocation

A. Steps of Algorithm

Step1: The time has been allocated to the cloudlets at the time of initialization only. The deadline has also been allocated. The actual time left and deployment time can be calculated. Step2: The alpha and beta calculated for each cloudlet. Step3: The ExtraMips calculation depend on alpha, beta and Required Bandwidth. The bandwidth of cloudlet is equal to the length remaining so far and actual time utilized by the cloudlet. Step4: Once the ExtraMips found, need to find out the maximum value of extraMips for the cloudlet. The cloudlet having maximum ExtraMips need to allocate Virtual Machine first. Step5: The cloudlet with extraMips allocate to the Virtual Machine first and so on.

III. ANALYSIS

The experiment performed on two different scenario. First, the allocation of virtual machine to the various cloudlet based on the cloudlet Id and VM Id, which is typical way where cloudlet assign to the vm based on the creation (First come First Serve). As per the graph observe that the requirement for extra Mips is high for cloudlet No. 8. The cloudlet 8 will assign to the vm once all the previous cloudlet 0 to 7 finished their task with their vms. Hence, the cloudlet has to wait till it get the virtual machine.

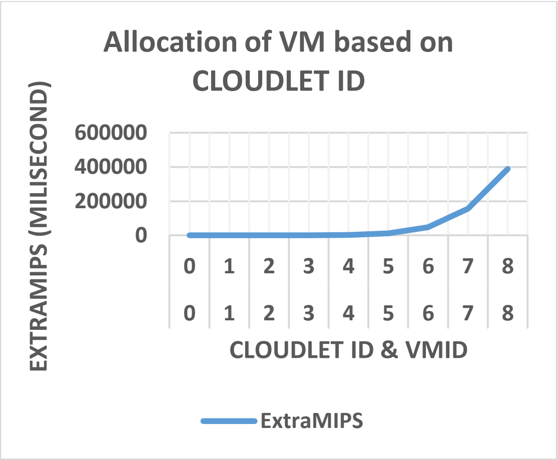


Figure 1

The observation from this is, the resources are underutilized for the cloudlets having less extra Mips. The cloudlet having requirement of extra resources are been waiting. Second experiment, The graph finding the extra Mips for the cloudlet. As it is observe that the cloudlet id 6,7,8 and 9 has maximum extra Mips.

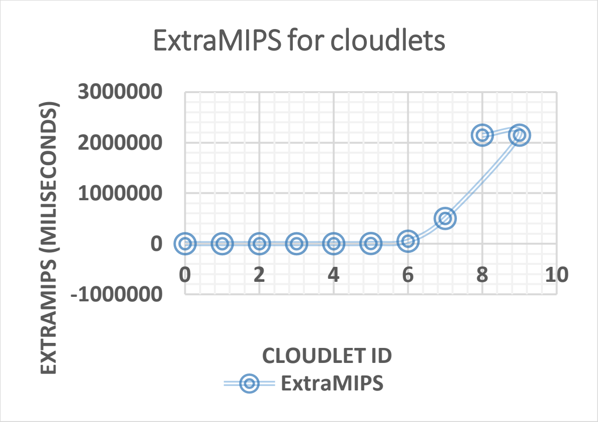


Figure 2

Here, the cloudlets allocated to the VM based on the extra Mips. So, the cloudlet id 0 having less extra Mips assign to the Virtual Machine 9 and cloudlet Id 8 having the maximum extra Mips is allocated to the VM 0.Hence, the allocation of the VM based on the Extra Mips.

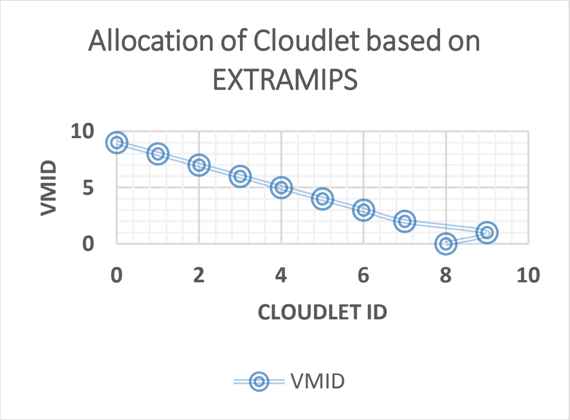


Figure 3

Due to the strategic calculation and allocation of VM, the advantage is can get optimum utilization of the local resources.

IV. CONCLUSION

This paper discuss the policies that take advantage of more information that can be provided by the scheduling service to solve resource provisioning problem from consumer's perspectives. The solution also leads to the optimum utilization of local resources. The benefits will drive for further reduction in budget expenditure for execution of applications in hybrid Clouds, what will further motivate adoption of hybrid Clouds as a platform for execution of application.

REFERENCES

[1] Amazon elastic couputing. http://aws.amazon.com/ec2/.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering IT services as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599{616.

[3] S. Yi, D. Kondo, and A. Andrejak, "Reducing Costs of Spot Instances via Checkpointing in the Amazon Elastic Compute Cloud," IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010.

[4] S. Chaisiri, R. Kaewpuang, B. Lee and D. Niyato, "Cost Minimization for Provisioning Virtual Servers in Amazon Elastic Compute Cloud,"19th Annual IEEE Int. Symposium on Modelling,z Analysis, and Simulation of Computer and Telecommunication Systems,2011

[5] C. Vecchiola, R. N. Calheiros, D. Karunamoorthya, R. Buyya, Deadline- driven provisioning of resources for scienti_c applications in hybrid clouds with Aneka, Future Generation Computer Systems, 2011, doi:10.1016/j.future.2011.05.008.

[6] A. Andrzejak, D. Kondo, and S. Yi, "Decision model for cloud computing under SLA constraints," Proc. of the 3rd IEEE Int'l Symp. on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 257 - 266, 2010.

[7] S. Chaisiri, R. Kaewpuang, B. Lee and D. Niyato, "Cost Minimization for Provisioning Virtual Servers in Amazon Elastic Compute Cloud," in 19th Annual IEEE Int. Symposium on Modelling,z Analysis, and Simulation of Computer and Telecommunication Systems, 2011.