RESEARCH PAPER | Engineering | Volume : 4 | Issue : 3 | Mar 2014 | ISSN - 2249-555X

User Annotated Resource Allocation in Cloud Computing Environment

KEYWORDS Resource allocation, virtual machine, Xen, hypervisor, user annotation

Krishnakumar L
Professor, Department of Computer Science and Engineering
Nehru Institute of Technology, Coimbatore

Jimy Joy
Post Graduate Scholar, Department of Computer Science and Engineering
Nehru Institute of Technology, Coimbatore

ABSTRACT *Cloud computing is the technology of next generation which combines everything into one. In Cloud computing a large number of cloud users can request multiple cloud services at the same time. So there must be a way in which all the resources are made available to the requesting user in a good manner to satisfy their needs. The mechanism of allocating available resources to the needed cloud applications through the internet is called Resource Allocation. When resource allocation is done in conjunction with user annotations as introduced in this paper can reduce the cost incurred for a user and hence it can attract more cloud users in future. User annotations like cost, deadline can be used. Users are allowed to submit the parameters during job submission. The user annotated parameters will then be considered while allocating resources to them. The main purpose of this paper is to increase information sharing among Cloud Users and Cloud Providers.*

INTRODUCTION

The reason for the drastic increase in the widespread popularity of cloud computing environment is due to its features like renting of computing resources on-demand, billing on a pay-as-you-go basis, and multiplexing of many users on the same physical infrastructure. These features of Cloud Computing systems give a deception of infinite computing resources to cloud users so that the resource consumption rate can be changed. At the same time, the cloud environment poses a number of challenges. There are two players in the cloud computing environments, cloud providers and cloud users, having different goals; providers want to increase the profit by achieving high resource utilization, while users want to decrease their expenses along with meeting their requirements. However, it is difficult to perform the resource allocation in a mutually satisfactory way due to the lack of information sharing between Cloud Users and Cloud Providers. Also due to the ever increasing challenges of cloud computing environment, both users and providers of Cloud are often put in dilemma.

Resource Allocation (RA) is the mechanism of attributing available resources to the needed cloud applications through the internet. Resource allocation craves services if the allocation is not managed precisely. In cloud computing resource allocation takes place in two situations. First, when an application is uploaded in to the cloud and second, when an application receives multiple requests at the same time [1]. When cloud computing is considered, resources are allocated to customers in terms of Virtual Machines (VM) on a demand basis. Traditionally, cloud providers specify a fixed price for each type of VM offerings [2].But, it is seen that this type of determination of price is often inappropriate due to the lack of information sharing between the users and providers in Cloud.

RESEARCH BACKGROUND

The communication among providers and users in cloud computing environment occur as shown in Figure 1. First, user sends a request for resources.

When the provider obtains the request, it checks whether the resources to satisfy the user request is available or not and then assigns the resources (if available) to the requesting user, in terms of Virtual Machines. Then the user uses the allocated resources to run applications and pays for the resources used. When the user has completed using the resources, they are returned to the provider. The Cloud Users and Cloud Providers have different goals. The goal of provider is to obtain as much profit as possible with minimum leverage. So the cloud providers try to optimize their computing resources; say by placing as many VMs as possible on each machine.

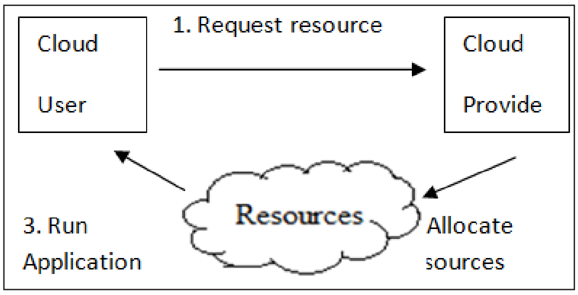

Figure 1: Cloud Usage

That is, Cloud providers want to increase their resource utilization. However, placing too many VMs on a single machine will cause VMs to interfere with each other and result in degraded and/or unpredictable performance which, in turn, frustrates the users [3]. Thus, the providers can reject existing VMs or resource requests to sustain quality of the service, but it can result in making the environment even more unpredictable. On the contrary, Cloud Users want their jobs done at little or minimal expense.

However, since these two parties do not share information with each other, an optimal allocation of resources is impossible. For example, the cloud providers do not disclose the details of machines used since that information is critical to their business. Likewise, users do not reveal their workload details. Therefore, users cannot clearly state their resource needs, because they are unaware of what is exactly available. Similarly, providers will be unable to allocate resources to suit user's requirements.

THE ASYMMETRY ALGORITHM

Asymmetry in this context is regarding the unevenness in the server utilization. The main components of the asymmetry algorithm are: load anticipation, hot spot extenuation, and green computing.

- **Hot spot Extenuation**

A server is listed as a hot spot if the resource utilization of that particular server is above a hot threshold. So, first the list of hotspots in the system needs to be sorted in decreasing order of their temperature and then migrate away the virtual machine that can reduce the server's temperature the most. All the hotspots must be eliminated if possible, or else keep their temperature as low as possible. Consider n denote the number of PMs and m denote the number of VMs present in a system. Let n_h be the number of hot spots in the system at a particular point. It will take O (n_h*log (n_h)) for sorting them based on their temperature. It is required to sort all the VMs running on each hotspot. Since the number of VMs running on a PM is limited to a small constant, the sorting also needs a constant amount of time. Scanning of all the PMs to find an exact destination for each VM takes O (n). Thus the total complexity of this phase is O (n_h* n) [5].

- **Green Computing**

If the resource utilization of physical machines is too low, some of them can be turned off in order to save energy and such servers are termed as cold spots. This is stated in the green computing algorithm. The main problem encountered here is to reduce the number of active physical machines without compromising the performance. The green computing algorithm is implemented when the average utilizations of all resources on active physical machines are below the green computing threshold. The list of cold spots in the system is sorted in the increasing order of their memory size and migrate away all its VMs before an underutilized server is turned off.

Let n_c denote the number of cold spots in the system at during a particular run. To sort them on the basis of their memory size, it takes O(n_c*log(n_c)) and it will take O(n) time to find an appropriate destination for each VM in the PM. Since it is assumed that the number of VMs on a PM is a small constant, the total complexity of this phase is O(n_c *n). The green computing phase is not invoked together with the hot spot extenuation phase at the same time.

- **Load Anticipation**

It is a difficult task for application developers to tell in advance the user load. If the future load is to be known beforehand, an offline algorithm can be used to calculate a good solution so that all application requirements are satisfied. In this case, anticipation algorithm is helpful in understanding the trend of how the load would change overtime. Load anticipation has great influence on resource allocation. If the load estimated is higher than the actual, the scheduler may allocate more resources than necessary and some of the resources will be wasted. On the other side, if the estimated load is much lesser than the actual load, the resource allocation may be insufficient. One category of widely used load anticipation algorithm is composed of variations of the Exponentially Weighted Moving Average (EWMA) algorithm. It is based on the assumption that the future value of a random variable has strong relation to its recent history. Anticipation error may result with unpleasant implications.

REDUCTION OF COST AND DEADLINE

The user is allowed to specify the parameters like cost, deadline during the job submission itself. Thus communication among the Cloud Users and Cloud Providers is enhanced. Thus through the resource allocation mechanism stated in the previous sections along with cost/deadline functions, both better resource allocation and cost/deadline optimization is obtained. In this work, we devise a mechanism that enables customers to specify favorable cost and deadline.

- **Cost Constrained Function**

Whereas consumers prefer the cheapest price for leasing a service, providers want to sell their services at the highest prices [4]. The selection of jobs to be scheduled can be based on any scheduling algorithm. Scheduling algorithm selects job to be executed and the corresponding resource where the job will be executed. In this particular case, consider only the cost specified by the user as a constraint while scheduling. As the Cloud User submits a job, the request is passed to the resource allocator. The Allocator runs the algorithm by considering the cost and then allocates the required resources to the clients that have requested it.

- **Deadline Constrained Function**

For deadline constraint, if a requested job can be executed within the specified limit, it is immediately serviced, whereas if the execution time is above the specified value, it is considered for cost constraint based allocation. Within the cost constrained allocation, if simultaneous requests come, then the one with the most cost efficiency is allocated first and so on. In the deadline constrained mechanism, compute the turnaround time of the task at each resource and select the resource with minimum turnaround time and schedule the task. The constraint based mechanism is dynamic in the sense that if cost constraint cannot be satisfied, the system will automatically look for satisfying the deadline constraint.

CONCLUSION

The design, implementation, and evaluation of a resource management system for cloud computing services along with cost/deadline optimization are presented. The resource allocation system multiplexes virtual to physical resources based on the changing demand on an adaptive basis. The asymmetry metric is used to combine VMs with different resource characteristics appropriately so that the capacities of physical machines are well utilized. The asymmetry algorithm achieves both overload avoidance and green computing. User annotations like cost, deadline when combined with the resource allocation provides satisfaction to user and hence more users will be attracted towards Cloud Computing.

REFERENCE

[1]Ronak Patel, Sanjay Patel. Survey on Resource Allocation Strategies in Cloud Computing. Gujarat Technological University, Gujarat. | [2]Qi Zhang, David R. Dynamic Resource Allocation for Spot Markets in Clouds. Cheriton School of Computer Science, University Of Waterloo, IT Convergence Engineering POSTECH, Pohang, South Korea. | [3]Gunho Lee.Resource Allocation and Scheduling in Heterogeneous Cloud Environments.Electrical Engineering and Computer Sciences, University of California at Berkeley. | [4]Seokho Son, Kwang Mong Sim. A Price- and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations. IEEE Transactions on systems, man, and cybernetics, June 2012. | [6]Zhen Xiao, Weijia Song, and Qi Chen. Supplementary File: Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment. | [5]Zhen Xiao, Peking University, China. An Infrastructureas- a-Service Cloud: On-Demand Resource Provisioning.