Article can be accessed online at http://www.publishingindia.com

Resource Allocation for Cloud Based E-Learning Systems

Bhavya Deep*, Preeti Yadav, Raunak Sett*****

Abstract

Cloud computing has become one of the most promising IT solutions and business trends in recent years. As the need for cloud is increasing and the requirements are getting more diverse and complex, the need to find an effective allocation strategy is increasing. In such a virtualized shared environment the resource requirements of applications keep changing from time to time. And in such a scenario, static allocation fails to solve the purpose. It leads to the inefficient utilization of resources and might affect the quality of the application. In this paper the allocation of resources on a university cloud is discussed. Cloud computing proves to be a powerful platform for higher education. Although the adoption of cloud on the university level has various advantages, a lot of other factors like dynamically changing resource requirements, etc. need to be taken into account. Here, a dynamic allocation mechanism is presented by modeling the resource requirements using a normal distribution based on which an algorithm to solve the resource allocation problem is proposed. The algorithm ensures that a particular virtual machine is never overloaded. It also tries to minimize the difference between the resources demanded and allocated so as to maintain the quality of an application running on a virtual machine.

Keywords: Dynamic Resource Allocation, Load Balancer, Global Scheduler, Local Scheduler, Normal Distribution

Introduction

The term cloud computing has been trending in the field of information technology since the past few years. However, it is still a topic of debate and is yet to be tapped to the fullest in various fields. One such field is education. In the most simple terms, cloud computing is the sharing and accessing of data over internet instead of your own computer's hard drive. It separates the application from the underlying hardware. It allows moving services from within an organization to shared systems.

Cloud computing can bring a major change in the field of education. In a few countries, universities have shifted their Email systems to cloud or have used it for accounting purposes. But, it has not been put to use completely. Cloud computing can prove to be much more advantageous to universities. It can be used for talent management, student recruiting, collaborating with industries and other universities which will give more exposure to the students and faculty, for research and innovation. Cloud will help the universities to use their data efficiently to make better strategic decisions for the future.

Although there are a lot of advantages of using cloud, there are a few drawbacks that come along like security issues, environmental issues, dynamic resource requirements, etc. We aim at proposing a resource allocation strategy that efficiently utilizes all the resources, maintains the quality of applications running on a virtual machine, and avoids the overloading of any physical machine.

* Assistant Professor, Department of Computer Science, Bhaskaracharya College of Applied Sciences, University of Delhi, India. Email: bhavya.deep@bcas.du.ac.in

** Student, B.Tech Computer Science, Bhaskaracharya College of Applied Sciences, University of Delhi, India. E-mail: yadavpreity@yahoo.co.in

*** Student, B.Tech Computer Science, Bhaskaracharya College of Applied Sciences, University of Delhi, India. E-mail: sett.raunak@gmail.com

Resource Allocation: It is a mechanism that aims to cater to the resource requirements of all running applications effectively which also includes the consideration of the current status and availability of all the resources in the cloud environment. A typical allocation strategy decides which, when and how many resources should be allocated.

A lot of activities occur in a university throughout the year. All these activities have different resource requirements. In order to fulfill these requirements and ensure the effective usage of resources, a proper allocation strategy must be adopted. There are mainly two types of resource allocation strategies which are given as follows.

Static Resource Allocation: In the case of static allocation, the cloud user is required to convey to the provider in advance about his requirements according to which resources are allocated. Another approach to this can be deciding the priority of each activity in advance and then allotting them resources according to their priority.

Although static allocation of resources seems to be easy to implement, it leads to the overutilization and underutilization of resources as the resource requirements or the priority of a particular activity might keep changing from time to time. So, in the present scenario where the resource requirements keep changing frequently, a static resource allocation strategy will be incompetent.

Dynamic Resource Allocation: A typical dynamic resource allocation strategy provides the resources to applications as and when demanded according to the availability. A strategy that allocates resources on demand is required because static allocation results in lower resource utilization, efficiency and application quality. Some major research challenges in cloud computing are green computing, resource optimization, quick response time, etc. Dynamically allocating resources can also solve problems like green computing, resource optimization, quick response time which are some major challenges in cloud computing. [1]

Dynamic Resource Allocation can be achieved by using various Resource Allocation Strategies. Resource allocation Strategies (**RAS**) are strategies through which we can make decisions on how to go dynamically allocate resource based on the classification. Having a good resource allocation strategy ensures that none of your resources are over provisioned or scarce. We choose a resource allocation strategy according to our requirements.

Resource allocation strategies can be based on

a. **Execution Time:** The requests execution time are estimated and the best possible resource are allocated to serve the request.

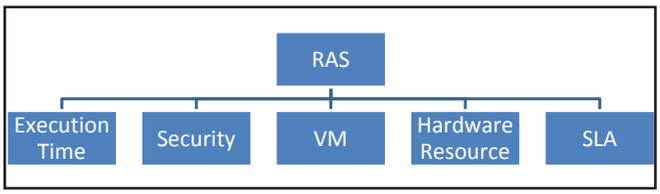
b. **Security:** This allocates the resource according to requirement of how isolated the resource should be by the request.

c. **VM (Virtual Machine):** Based on the requirement Load, Cost or Speed Virtual Machines could be started or destroyed.

d. **Hardware Resource:** Resources are allocated according to the requirement of hardware resources such as CPU Time, I/O operations, Storage requirements etc.

e. **SLA (Service Level Agreement):** Based on the agreement with the cloud vendor, the service and quality of the response to request is determined.

Figure 1. [2]



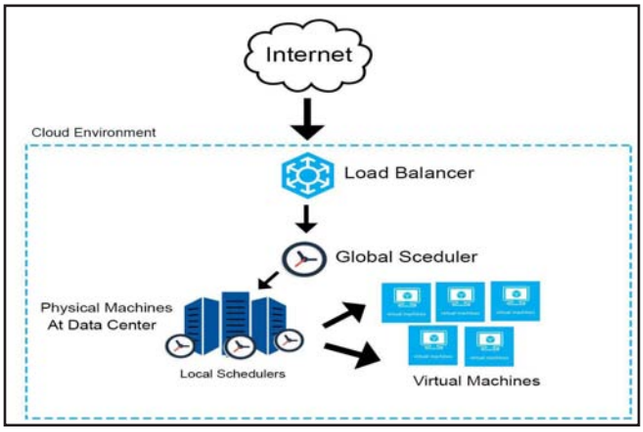
One of the major resources that any university needs during peak usage times is CPU time. For example, at the time of admissions the requests to the server time out as the server reaches its TPS (Transactions Per Second) and hence the request fails to execute. It has been observed that in a typical university system, the resources are either under provisioned or over provisioned and cannot adapt to sudden requirement changes. So even if we over reserve the resources when starting up the server, it's really hard to downscale the resource allocation. To solve this problem we are trying to propose a method through which dynamic resource allocation can be achieved and the resources can be utilized to their maximum potential.

Methodology

Load Balancer: It is a device that is used for managing requests maintaining maximum productivity of resources and prevents and overloading of resource use. It provides high availability of resources so that they are readily available. It provides redundancy and fail over

mechanisms for whenever a resource faces outage. A Load balancer will be the entry point for any request, load balancing helps handling requests appropriately and provides robust fail over mechanisms and scale up and down using a scheduler to facilitate allocation appropriate resources. The load balancer itself are implemented in pairs for high availability. The load balancer then decides what to do with the request and tells the global or local scheduler what to do next. [3]

Figure 2.



There is a global scheduler and a local scheduler. [4] The global scheduler is responsible for controlling all the physical machines in a data center. The local scheduler is responsible for controlling the virtual machines on a particular physical machine. Requirements of a university are modeled using normal distribution.

If we talk about requests, we have categorized all requests in a way that they have similar resource requirements (e.g. CPU time). Whenever a request is received the type is identified for the request (whether the request is a high CPU time, or a demanding request, etc.). Then it sends the request with the necessary parameters to the global scheduler which then allocates a physical machine. This ensures that the CPU utilization on any physical machine remains within the tolerance limit (where tolerance is given by the deviation allowed for that particular machine).

Whenever the global scheduler hands over a task to the local scheduler of a physical machine, it allocates the most suitable virtual machine to that task. If, an activity needs resources that go beyond the tolerance limit (sum of mean and deviation) of a machine, the local scheduler again hands it over to the global scheduler which in turn allocates the best suited machine to it. Also in the case where the resources required by an activity are much below the mean of that machine, the activity is moved to another machine by the global scheduler so that it can be used for some other tasks.

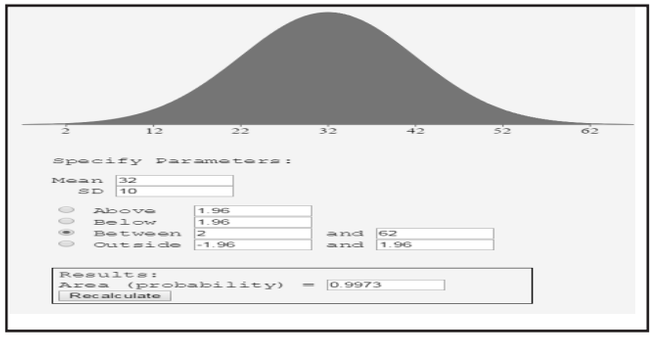
Not all machines in the data center shall have the same resource capacity. Some will have a lesser capacity than others. The activities that have lesser resource requirements will be allocated to a machine with lesser capacity. Since a high resource demand is not observed often, the machines with higher capacity can be turned off at those times to save energy. This is identified by the load balancer according to the requests that are being sent to it.

In this way, underutilization and overutilization of resources can be avoided. This also maintains the quality of application running on virtual machines by minimizing the difference between the resources desired and allocated. Also, when the resource demand is high the load balancer ensures that there are sufficient fail over nodes to be highly available for any kind of unexpected resource peak.

Results

All tasks/activities are categorized according to their nature and resource requirements as per the data collected during past months. The categorization is done by the global scheduler. For each activity, there is mean resource (e.g. CPU) utilization and a deviation value which indicates how much an activity's resource utilization can shift from the mean value. This deviation value is called the tolerance limit. If the resources demanded by a virtual machine exceed this tolerance limit, the activity is either stopped or migrated according to the availability of resources.

Figure 3. [5]



All the major activities occurring on a university system closely follow a normal distribution [6]. High deviations in the resource requirements are majorly noticed at the time of admissions and result declaration which is a very little percentage. At the other time of the year, the resource requirements remain nearly stable. These are the properties of a normal curve. In figure III, the CPU utilization has been given in percentage (%). The probability of area between 2% and 62% is 0.9973 (so the normal distribution is being followed closely). (Calculations have been done using the following source: http://onlinestatbook.com/2/calculators/normal_dist.html).

Discussion

A normal distribution [7] of a random variable X with mean μ and variance σ^2 is a statistic distribution with probability density function

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

on the domain $x \in (-\infty, \infty)$. The normal distribution is sometimes also called the bell curve which is shown in figure IV.

Figure 4. Normal Distribution Curve

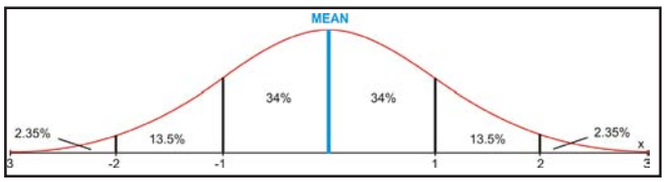


Figure 5. Resource Utilization in a Typical University System

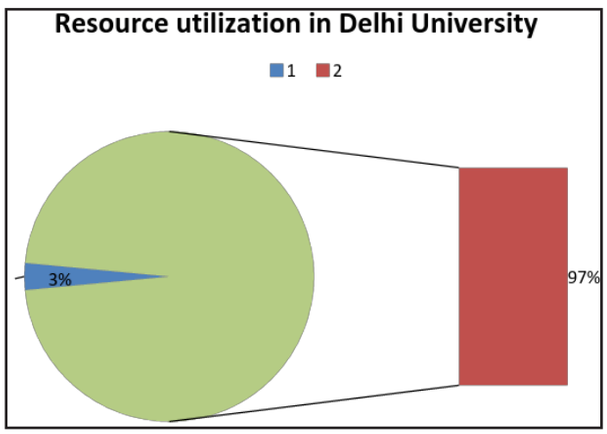


Figure-V Here, 3% indicates the probability that the resource utilization deviates thrice or more than its deviation value. Such a scenario is observed mostly during the time of admissions when the hits on the server are a lot more than the rest of the year. On calculating the percentage of the time during which the resource utilization is at a peak (i.e. thrice or more than the standard deviation) the value was approximately 3%, which is also the case in a normal distribution.

Conclusion

Cloud computing will prove to be a game changer for higher education institutions. It will help universities to focus more on the teaching and learning process rather than spending a tremendous amount of their time and money on maintaining the IT infrastructure, libraries, etc. The major requirements of a university were gathered after which the pattern of their resource requirements throughout the year was studied and modeled using the normal distribution curve. With the help of the data collected, a strategy for allocating resources to these activities has been presented.

Acknowledgements

We would like to extend a vote of thanks to University of Delhi, for providing us with an opportunity to be a part of this Research paper. Our heartfelt gratitude to our mentor Prof. Punam Bedi for her unparalleled support and continuous guidance throughout our research work which was based on the Dynamic Resource allocation for cloud computing.

References

1. Patil, S. S., & Bhavani, K. (2016). Dynamic resource allocation using virtual machines for cloud computing environment. *International Journal of Engineering and Advanced Technology*, August, 3(6), 583-588.
2. Vinothina, V., Sridaran, R., & Ganapathi, P. (2012). A survey on resource allocation strategies in cloud computing. *International Journal of Advanced Computer Science and Applications*, 3(6), 97-104.
3. Randles, M., & Lamb, D., & Taleb-Bendiab, A. (2016). *A comparative study into distributed load balancing algorithms for cloud computing*.

4. Song, Y., Sun, Y., & Shi, W. (2015). *A two-tiered on-demand resource allocation mechanism for VM-based data centers*. IEEE Transactions on Services Computing, January-March, 6(1).
5. *Normal Distribution*, (2016). Retrieved from http://onlinestatbook.com/2/calculators/normal_dist.html.
6. Sembiring, K. (2016). *Dynamic Resource Allocation for Cloud-based Media Processing*. NEC Labs Europe Ltd. and Andreas Beyer, NEC Labs Europe Ltd.
7. http://mathworld.wolfram.com/NormalDistribution.html
8. Gonçalves, G. E., Endo, P. T., Cordeiro, T. D., Palhares, A. V. A., Sadok, D., Kelner, J., Melander, B., & Mångs, J. E. (2016). *Resource Allocation in Clouds: Concepts, Tools and Research Challenges*.
9. Wolke, A., Bichler, M., & Setzer, T. (2016). *Planning vs. dynamic control: Resource allocation in corporate clouds.*
10. Vinothina, V., Sridaran, R., & Ganapathi, P. (2015). A Survey on resource allocation strategies in cloud computing. *International Journal of Advanced Computer Science and Applications*, 3(6), 97-104.
11. "Sudeepa, R., & Guruprasad, H. S. (2014). Resource allocation in cloud computing. *International Journal of Modern Communication Technologies & Research*, April, 2(4), 19-21.
12. Patil, S. D., & Mehrotra, S. C. (2012). Resource allocation and scheduling in the cloud. *International Journal of Emerging Trends & Technology in Computer Science*, May-June, 1(1), 47-52.
13. Hadji, M., & Zeghlache, D. (2016). *Minimum cost maximum flow algorithm for dynamic resource allocation in clouds.*
14. Kumar, K. P., Kumar, S. A., & Jagadeeshan. (2013). Effective load balancing for dynamic resource allocation in cloud computing. *International Journal of Innovative Research in Computer and Communication Engineering*, July, 2(3), 3427-3431.
15. Sharma, N., & Dhanda, M. (2014). A review of cloud systems and resource allocation strategies. *International Journal of Advanced Research in Computer Science and Software Engineering*, April, 4(4), 62-66.