Journal of The Gujarat Research Society

ISSN: 0374-8588
Volume 22 Issue 1, January 2020

A Heuristic Markova Approach to Dynamic Resource Provisioning in Cloud Computing

Mr. Nitin Kumar, Mr. Rajesh Pandey

Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut

Email Id- nitin.kumar@shobhituniversity.ac.in, rajesh@shobhituniversity.ac.in

ABSTRACT: ***Cloud computing differs from previous distributed computer paradigms in that it offers more dependable and flexible access to IT resources. The issue of providing and distributing resources on demand in response to changing workloads drives the challenge of managing applications effectively in cloud computing. The majority of studies have focused on controlling this demand at the physical layer, with just a handful at the application layer. This article focuses on an application-level resource allocation approach that assigns a sui. number of virtual machines to an application that exhibits dynamic resource needs. To the best of the authors' knowledge, this is the first completely estimation-based study in this area. The suggested method provides a more cost-effective resource provisioning approach when addressing cloud user needs, according to the findings of the experiments. One of the major distinctions from the DTMC is that global transition updates offer a short memory for the system. The goal of this research is to reduce the number of virtual machines leased from the provider the customer while taking into account application needs. We want to reduce the number of virtual machine instances for a certain incoming workload (W) while providing enough resources for the application***

KEYWORDS: ***Adaptive Resource, Cloud Computing, Dynamic Resource, Estimation Markov Chain, Provisioning.***

1. INTRODUCTION

The desire for computation as a utility paradigm has grown as a result of recent advancements in Information Technology, which have increased the need for computations whenever and wherever they are needed on the one hand, and the need of individuals and organizations for cost-effective heavy-duty computation powers on the other. Cloud computing is the most recent response to these trends, in which IT resources are provided as a service. Cloud computing also provides users with an unlimited resource pool, which distinguishes it from conventional hosting services. The fact that the average data center usage is projected to be equivalent to 25,000 homes, as the vast number of data centers across the globe, demonstrates the need of a resource provisioning strategy that optimizes resource use. Furthermore, effective resource provisioning may be used to reduce user fees by using the resources [1].

In Cloud Computing, the phrase Resource Provisioning refers to the process of bringing an application into the cloud, deploying it, and managing it. One of the fundamental concepts in resource provisioning is to provide resources to applications in such a manner that they use less power and money by maximizing and using available resources. As a result, various power management methods are being investigated in this area. There are two general approaches to resource provisioning: One is Static Resource Provisioning, which supplies the program with the peak time resource it requires all of the time. The majority of the time, resources is squandered in this kind of provisioning since the workload is not peaked in reality. The other is Dynamic Resource Provisioning, whose primary premise is to allocate resources depending on the application's requirements.

This kind of provisioning allows cloud providers to utilize a pay-as-you-go invoicing system, which is one of the most popular features of cloud computing among end users. In the current study, we created a learning-based dynamic resource provisioning method. The remainder of this work is structured in the following manner. An overview of relevant studies is included in. The suggested technique is the outcomes of the experiments. The paper comes to a close with. The aim was to optimize the profit of the resource supplierby reducing power usage and SLA violations. The Kalman filter was used to estimate the amount of incoming requests in order to forecast the system's future condition and make appropriate reallocations created an optimization technique that utilized the Constraint Satisfaction Problem to represent both the provisioning and allocation problems [2].

The allocation of resources is based on a threshold. Their algorithm uses this threshold to determine if the current counts of virtual machines allocated to an application are adequate or not, and the same is true for over provisioning. The main distinctions and benefits of our research over the latter are that, first; our work does not need any human admin intervention and is capable of estimating the future workload rather than taking a reactionary action used a queuing model and analytical performance to forecast workload and resource adaptation. As in earlier studies, there is a human control element.

By creating a Neural Network system named ECNN (Error Correction Neural Network) and applying it with a Linear Regression, Created a new machine learning method. The majority of techniques rely on load balancing and assigning physical resources to virtual resources. Only a few of them thought about the application layer. And there isn't single completely approximate-based research among these few. The authors of this article attempted to address these flaws. We looked at a number of existing studies on resource provisioning techniques, and while some of them, such as and, seemed promising; they were not feasible in a real cloud environment because they are reactive approaches that act only after the workload has arrived, whereas creating a virtual machine takes time.

Another issue is reliance on factors like as, which is not ideal for an autonomous system. In addition, as a system becomes more sophisticated, it generates more overhead. It will make it tough to adopt a strategy. In light of the above, we've selected a basic, self-contained learning system. Using estimates, the system may forecast future cloud application requirements. To deal with the diversity of the environment, a quasi-[3].

DTMC1 heuristic method that is appropriate for dynamic workloads was selected. Because the suggested approach is not too complicated, it may be implemented for each user in the Cloud Manager or Broker, as well as in the cloud system's client side. CY was selected as the winner. This provisioning strategy uses the word mean because the virtual machine number is the average of the lowest allocate able virtual machine and maximum allocate able virtual machine. As a result, the processing power is always 6000 MIPS . Obviously, this is a cautious approach that is less expensive than the alternatives. However, since this method is unable to identify virtual machine saturation, the user suffers from under-provisioning when the demand is above average. In comparison to, average usage seems to be improved, but both under and over use issues persist.

Processing power adapts to the demand and is no longer constant. The first system begins in a normal state and attempts to assess the surroundings and predict the workload's future state. The times when the system is in a normal or learning condition are shown by the SVMP

curve flat regions. Figure 1 discloses the Static resource provisioning vs. dynamic resource provisioning.

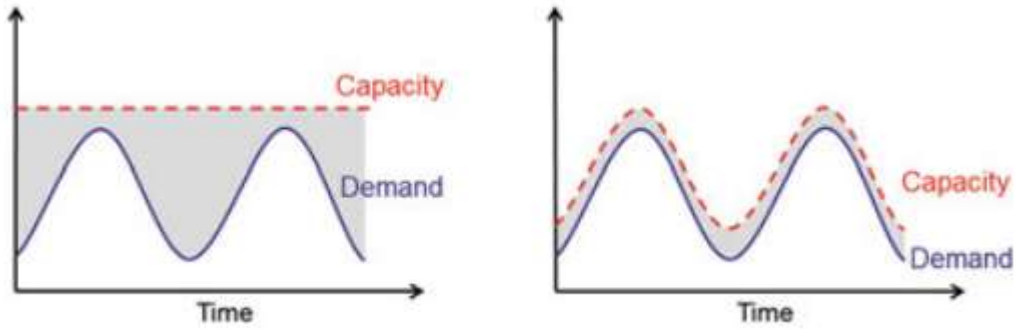

Figure 1: Static Resource Provisioning Vs. Dynamic Resource Provisioning

2. DISCUSSION

A fundamental discrete time A memory less system having a finite or count. number of states and transitions between them is known as a Markov Chain (DTMC). Memory less 1 Discrete-time Markov chain is a phrase used to describe a discrete-time Markov chain. According to H.R. the next action is solely dependent on the present state and not on previous occurrences. Each state has a probability Pi, which represents the likelihood of state I occurring. There are transitions from one state to the next with the probability which indicates the likelihood of transitioning from state I to state. When we're in state I and pi,j exists, the probability of transitioning to state, as well as transition probabilities pi,j, albeit with minor variations [4].

There is no probability for each state separately, but much as the previous one, the new one is dependent on the prior one. The probabilities pi,j are no longer set, and they fluctuate over time as the environment changes. The new state is selected based on the previous state and the between them that has the best probability of being picked.

A learning component operates alongside the state machine. The learning algorithm is built on a punishment/reward system, and it utilizes the average virtual machine usage as feedback to determine which action is the best. In addition, when the appropriate action must be performed in each state, the learning algorithm would regulate the aggressiveness level of the action in relation to the use of virtual machines. These are the initialization and continuation relations for a state diagram system with (N) transitions All transitions have an equal probability at first and would be started using the following equations [5].

The learning algorithm rewards the likelihood of transitioning to the correct state from the present state while penalizing the likelihood of alternative transitions. When the penalty is calculated as a Decrement Step, the reward is calculated as follows: For this study, a three-state machine with seven transitions was built to dynamically regulate the number of VMs assigned to a particular task The workload was over provisioned in the Decrement state the workload was provided just enough in the Normal state and the workload was under provisioned in the Increment state.

One of the major distinctions from the DTMC is that global transition updates offer a short memory for the system. The goal of this research is to reduce the number of virtual machines leased from the provider bthe customer while taking into account application needs. We want

to reduce the number of virtual machine instances for a certain incoming workload while providing enough resources for the application. As a result, MIPS2's overall processing power It is also necessary to compute the average usage of virtual machines under workload (W) with allotted virtual machine number [6].

However, this technology is usually incapable of simulating Dynamic Virtual Environments.Cloudsim architecture has been implemented. There are two fundamental components in the system. The first is a cloud broker that monitors application workloads and interacts with the cloud to adjust resources. The other is a dispatcher in the cloud body that assigns VMs to workloads as needed. Machine provisioning, in addition to the inability to mimic resource provisioning in the application layer, has been added to the simulator with additional component sand characteristics to allow it to handle Dynamic VM provisioning in the application layer.

A workload dispatcher was created in addition to SVMP to distribute user workloads across available VM instances. This dispatcher feeds each VM with incoming cloudlets until the VM usage falls below 80%, with the remaining 20% set aside for high loads in the future. VMs would be used in a fair way using this approach, and over provisioned VMs would stay load-free and simply terminated. This criterion is considered 85 percent in certain instances, such as web servers, but since SVMP is intended for more broad applications and is a learning-based system that takes time to train, we set it at 80 percent [7] .

if the algorithm was in the proper state, it would take the correct action; otherwise, it would update the probabilities to the proper state and take the action of the current state even if it is not the correct one until it reaches the correct action. The severity of actions varies, and the learning system determines how many virtual machines should be added or deleted depending on the average usage rate. We conducted three tests utilizing three distinct methods to assess our suggested system. The first and second examples use two different static provisioning methods to illustrate cloud behavior. Experiment 3 demonstrates how the proposed dynamic VM provisioner affects cloud behavior. The system has been put through its paces using a Normal Distribution Workload.

The workload begins with a modest amount of MI, rises to a high, and then begins to decrease; this pattern occurs often in real life, with varying peaks and slopes. Time’s previously stated, we used CloudComp to replicate our research using three distinct experiments. To begin, we've created the following scenario to demonstrate the studies: shows how the system behaves with a static provisioning strategy that uses the maximum virtual resource provisioning. There are 30 virtual machines in all. This figure is based on the maximum number of virtual machines that our dynamic method can supply [8].

As a result, the processing power would remain constant at 12000 MIPS .shows how many virtual machines were used on average throughout this experiment. This map has no saturation areas, indicating that there are no under provisioning. Because this is an over provisioning policy, this is perfectly understandable. However, the greatest average virtual machine usage of 66.25 percent was achieved this entails a significant level of resource waste. As a cloud user, the application extender incurs extra costs as a result of this common kind of conventional resource provisioning for apps. It was decided to use a static provisioning strategy with a mean virtual resource provisioning policy. This provisioning strategy uses the word mean because the virtual machine number is the average of the lowest allocate able virtual machine and maximum allocate able virtual machine. As a result, the

processing power is always 6000 MIPS . Obviously, this is Average usage of supplied virtual machines during Experiment 1. However, since this method is unable to identify virtual machine saturation, the user suffers from under-provisioning when the demand is above average. In comparison to Experiment 1, average usage seems to be improved [9], but both under and over use issues persist. The findings of utilizing our suggested method, the SVMP, are described in this. This is an adapt. method unless two preceding procedures are used. Processing power adapts to the demand and is no longer constant. The first system begins in a normal state and attempts to assess the surroundings and predict the workload's future state. The times when the system is in a normal or learning condition are shown by the SVMP curve flat regions in Fig. 7.On the one hand, SVMP attempts to satisfy application requirements while lowering cloud user costs on the other. The amount of provided virtual resources for a user is decreased by allocating allotted virtual machines to the user's workload. As a result, the overall usage of virtual machines becomes a highly significant feedback metric. Figure 2: implemented architecture in CloudComp. System contains two basic components. First is a broker allocated out of the cloud body which observes application workloads to adapt resources the other is a dispatcher in cloud body which allocates vms to workload appropriately.

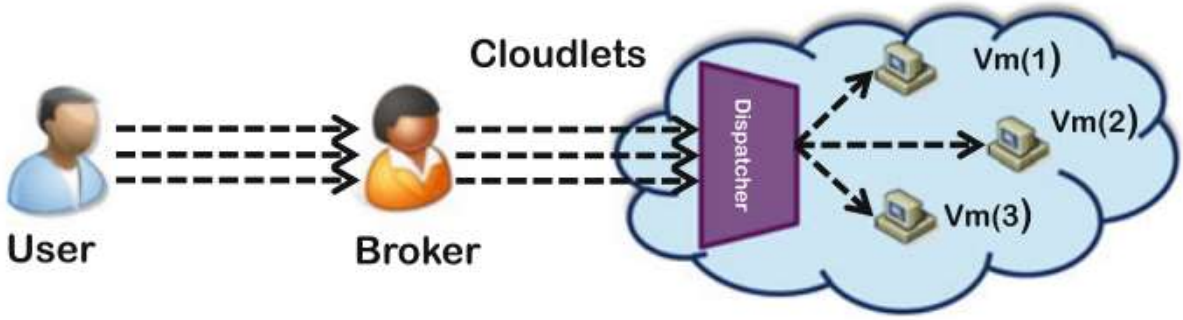

Figure 2: Implemented Architecture In CloudComp. System Contains Two Basic Components. First Is A Broker Allocated Out Of The Cloud Body Which Observes Application Workloads (From The Cloud User's Side) And Communicates With Cloud (Provider) To Adapt Resources The Other Is A Dispatcher In Cloud Body Which Allocates Vms To Workload Appropriately

Machine provisioning, in addition to the authors' goal of simulating resource provisioning at the application layer, has resulted in additional components and characteristics. The simulator has been updated to support dynamic VM provisioning inlayer of the application workload dispatcher was created in addition to SVMP to distribute user workloads (called cloudlets in CloudComp) across available VM instances. Until VM usage falls below 80%, this dispatcher feeds each VM with incoming cloudlets. The remaining 20% is set aside for large loads in the future. Virtual Machines (VMs) may be created using this technique [10].

Would be used in a fair way, and over provisioned VMs would still be available. It's free of burden and may be turned off quickly. This is true in certain instances, such as web servers.SVMP is intended for more broad use, and the threshold is regarded 85 percent .Apart from that, it's a learning-based system that takes some time to master, thus the cutoff point was set at 80%.According to the decision. , if the algorithm was in good working order, it

would Otherwise, it would update the probabilities to the correct condition and perform the right thing.the current state's action until it reaches how to take the appropriate action. The level of aggressiveness in acts varies, and the learning system determines this. Depending on the average, how many virtual machines must be added or deleted level of usage.

3. CONCLUSION

Dynamic resource provisioning allows cloud users to complete activities while maintaining QoS in a more cost-effective manner. Because providing resources in this layer is not instantaneous, and a virtual machine instance takes some time to start and become functional, a true application layer dynamic resource provisioner must predict the users' workload and provide the resources before the workload arrives. The experiment described in this article was aimed towards

Average Time of Utilization the average usage of supplied virtual machines throughout the course of the experiment. The SVMP monitors virtual machine usage to identify application needs and scales up or down when resources are under or over supplied, accordingly. In tested methods, total virtual machine usage averages (left) and total virtual machine cost for the cloud user (right). Dynamic resource provisioning using a learning-based system called SVMP to decrease costs while maintaining cloud user application needs to enhance the performance of the resource provisioning method; we are presently expanding our system to use additional learning algorithms.

REFERENCES

[1] S. Shilpashree, R. R. Patil, and C. Parvathi, "'Cloud computing an overview,'" *Int. J. Eng. Technol.*, 2018.

[2] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of IoT and Cloud Computing," *Futur. Gener. Comput. Syst.*, 2018.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, 2010.

[4] B. de Bruin and L. Floridi, "The Ethics of Cloud Computing," *Sci. Eng. Ethics*, 2017.

[5] J. Lee, "A view of cloud computing," *Int. J. Networked Distrib. Comput.*, 2013.

[6] V. Pushpalatha, K. B. Sudeepa, and H. N. Mahendra, "A survey on security issues in cloud computing," *Int. J. Eng. Technol.*, 2018.

[7] N. Subramanian and A. Jeyaraj, "Recent security challenges in cloud computing," *Comput. Electr. Eng.*, 2018.

[8] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Futur. Gener. Comput. Syst.*, 2013.

[9] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Applied Computing and Informatics*. 2018.

[10] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Trans. Serv. Comput.*, 2018.