Dynamic Resource Provisioning in Cloud Computing: A Heuristic Markovian Approach

Hamid Reza Qavami[1(✉)], Shahram Jamali[1], Mohammad K. Akbari[2], and Bahman Javadi[3]

[1] University of Mohaghegh Ardabil, Ardabil, Iran
qavami@gmail.com, jamali@iust.ac.ir
[2] Amirkabir University of Technology, Tehran, Iran
akbarif@aut.ac.ir
[3] University of Western Sydney, Richmond, Australia
b.javadi@uws.edu.au

Abstract. Cloud computing provides more reliable and flexible access to IT resources, which differentiates it from other distributed computer paradigms. Managing the applications efficiently in cloud computing motivates the challenge of provisioning and allocating resource on demand in response to dynamic workloads. Most of investigations have been focused on managing this demand in physical layer and very few in application layer. This paper focuses on resource allocation method in application level that allocates appropriate number of virtual machines to an application which demonstrates a dynamic behavior in terms of resource requirements. By the knowledge of authors this is the first fully estimation based investigation in this field. Experimental results demonstrate that the proposed technique offers more cost effective resource provisioning approach considering cloud user demands.

Keywords: Cloud computing · Dynamic resource provisioning · Adaptive resource provisioning · Estimation · Markov chain

1 Introduction

With recent progressions in Information Technology the need for computations when ever and where ever on the one hand and also the need of individuals and organizations for cost effective heavy duty computation powers on the other hand, have increased the desire for computation as a utility paradigm. Cloud computing is the latest answer to these tendencies where IT resources are offered as services. Cloud computing also offers the user an infinite resource pool (e.g. processing capacity, Memory, Storage etc.); an intrinsic feature of cloud computing that severs it from traditional hosting services.

The fact that the average data center consumption was estimated to be something as many as 25,000 households [1] plus the huge amount of those data centers in the world, clearly shows the necessity of an optimizing resource provisioning policy. In addition an efficient resource provisioning is able to utilize the resources for reducing user payments.

V.C.M. Leung and M. Chen (Eds.): CloudComp 2013, LNICST 133, pp. 102–111, 2014.
DOI: 10.1007/978-3-319-05506-0_10,

Generally the term Resource Provisioning in Cloud Computing is used for the taking in, deploying and managing an application on Cloud infrastructure. One of the main ideas in resource provisioning is to provision resources to applications in a way that reduces power and cost by optimizing and utilizing available resource. Hence some power management techniques are considered in this field in some of investigations. As a whole there are two generic way of resource provisioning:

One is Static Resource Provisioning which usually provides the peak time needed resource all the time for the application. In this kind of provisioning most of the time the resources are wasted because the workload is not peaked in reality. The other is Dynamic Resource Provisioning which its basic fundamental idea is to provision the resources based on the application needs (Fig. 1). This type of provisioning enables cloud provisioner to use pay-as-you-go billing system which is one of the end users' favorite advantages of cloud computing. We have developed a learning based dynamic resource provisioning approach in the present investigation.

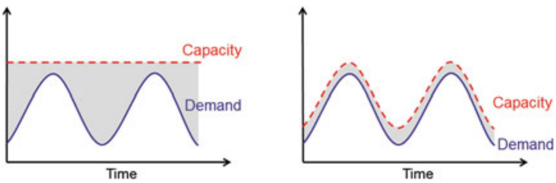

Fig. 1. Static resource provisioning vs. dynamic resource provisioning

The rest of this paper is organized as follows. Section 2 presents a review of related works. Section 3 describes the proposed methodology from approach to implementation. Section 4 discusses experimental results. Section 5 concludes the paper.

2 Related Works

One of the headmost investigations about power management was carried out by Pinheiro et al. [2] the idea was about addressing power conservation for clusters of workstations or PCs. Elnozahy et al. in [3] combined Dynamic Voltage Frequency Scaling with dynamically turning on/off method called VOVO (vary-on/vary off) to reduce power consumption. Kusic et al. [4] used Limited Lookahead Control (LLC). The goal was to maximize the resource provider's profit by minimizing both power consumption and SLA violation. Kalman filter was used to predict the number of next coming requests to predict the future state of the system and perform necessary reallocations. Verma et al. [5] solved the problem of power-aware dynamic placement of applications using Bin Packing problem. Van et al. [6] developed an optimization method and by modeling both provisioning and allocating problem they used Constraint Satisfaction Problem (CSP). Lin et al. [7] purposed a new Round Robin algorithm called Dynamic Round Robin (DRR) to allocation and migration of Virtual Machines between hosts. Lin et al. [8] introduced a dynamic Virtual Machine-Varying

Based resource allocation using a threshold. Using this threshold their algorithm decides that the current counts of virtual machines which are assigned to an application are sufficient or not, it is the same for over provisioning. The basic differences and advantages of our study as compared to the latter are that first, our work does not need any human admin interferences and is able to approximate next workload instead of a reactive action. Calheiros [9] et al. addressed workload prediction and resource adaption using a queuing model and analytical performance, like previous work, there is a human control parameter in this.

Iqbal et al. [10] aimed a bottle neck detection system for multi tier web application using a heuristic approach, this mechanism is able to detect bottle necks in every tier of system. Jeyarani et al. in [11] developed a dispatcher using a new PSO (Particle Swarm Optimization) method Called SAPSO (Self-Adaptive PSO) to dispatch virtual machine instances among physical servers efficiently. Zaman et al. [12] showed a new bid based (capital market model) approach for responding to the users' requests. Islam et al. [13] advanced a new machine learning technique by developing a Neural Network system called ECNN (Error Correction Neural Network) and using it side by side with a Linear Regression.

Most of methods relied on allocating physical resources to virtual resources and load balancing methods. Few of them considered the application layer. And among these rare studies there is not a fully approximate based study. In this paper authors tried to cover these leakages.

3 Proposed Methodology

We explored a number of existing investigations on resource provisioning techniques, some of them like [8] and [10] seemed to be good but not feasible in a real cloud environment, because they are reactive approaches and take action when the workload has already arrived, while creating a virtual machine is not instantaneous. The other problem is dependencies on parameters like [9], which is not favorable for an autonomous system. Also, more complexity imposes more overhead in a system. It will make an approach difficult to be accepted. Considering all above, we have chosen a simple learning system which is fully autonomous. The system can predict future needs of a cloud application using estimations. A quasi-DTMC[1] [14] heuristic approach which is suitable for dynamic workloads has been chosen to overcome the variety of the environment. The proposed method is not too complex so it can be implemented for each user in Cloud manager or in broker, even in the client side of the cloud system.

3.1 Approach

A basic discrete time Markov Chain (DTMC) is a memory less system with finite or countable number of states and transitions between them. The term memory less

[1] Discrete-time Markov chain.

means that the following action (state) depends only on current state and not preceding events. Each state have the probability P_i which indicates the happening chance of state i. In each state there are transitions to the other states with probability $\pi_{i,j}$ which means transition probability from state (i) to state (j). When we are in state (i) and $\pi_{i,j}$ exist, the chance of going to state (j) is $P_i \times \pi_{i,j}$ While $\Sigma_j\ \pi_{i,j} = 1$ and $\Sigma_k\ P_k = 1$.

In presented model, there is a state output diagram just like DTMC and there are transition probabilities $\pi_{i,j}$ too, although with some differences. Here we have $\Sigma_m \Sigma_n \pi_{m,n} = 1$.

There is not any probability for each state individually here but just alike the former one new state depends on the previous one. Probabilities $\pi_{i,j}$ are not fixed anymore and they vary during the time depending on the environment changes. New state is chosen based on previous state and the $\pi_{i,j}$ between them which owns the greatest chance to choose.

Beside the state machine a learning part works as well. The learning algorithm is a punishment/reward based one and uses the average virtual machines utilization as a feedback to understand which action among is the proper action. Also in each state, when the corresponding action must be taken, the learning algorithm would control the aggression amount of the action looking to virtual machines utilization.

For a state diagram system with (N) transitions there are these relations for initialization and carrying on the work:

In the beginning the chance of all transitions is equal and would be initiated using these equations:

$$\Pi_{m,n} = \frac{1}{N} \tag{1}$$

The learning algorithm rewards the transition chance to the proper state from current state and punishes the other transitions by decreasing their chance. Considering the punishment value as a Decrement Step, the reward (Increment Step) would be gained as below:

$$\text{IncrementStep} = (N - 1)\ \text{Decrement Step} \tag{2}$$

Using mentioned principles for this investigation, a three state machine with seven (N = 7) transitions, was developed to control the number of allocated VMs to a specific workload dynamically (Fig. 2):

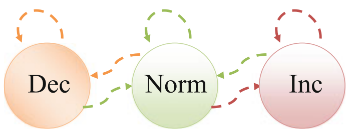

Fig. 2. Proposed state machine

Decrement state (briefly Dec) means that the workload was over provisioned, Normal state (briefly Norm) means that the workload was provisioned just enough and Increment state (briefly Inc) means that the workload was under provisioned.

Global transition updates provide a short memory for the system and this is one of the key differences to the DTMC.

3.2 Problem Formulation

The purpose of this study is to minimize the amount of rented virtual machines from the provider by the user, considering application demands. For a particular arriving workload (W) we want to minimize the number of virtual machine instances while providing appropriate amount of resources for the application. So the total processing power in MIPS[2] must be greater than total workload in MI[3] (Fig. 3):

$$\text{Object for: } \min\left(\sum_{\mathrm{n=1}}^{\mathrm{MaxVM}} \mathrm{VMlist}_{\mathrm{n}}^{\mathrm{W}}\right) \tag{3}$$

$$\text{Subject to: } \sum_{1}^{MaxOnlineVMs} MIPS_{VirtualMachines} > \sum_{1}^{CurrentCloudLetNumber} MI_{CloudLets} \tag{4}$$

Also the average utilization of virtual machines under workload (W) with allocated virtual machine number (OnlineVMs) is needed to be calculated which has been formulated as below:

$$VMs\ Avg.\ Utilization^{W} = \frac{\sum_{i=1}^{OnlineVMs} VM_i^{W}\, Utilization}{OnlineVMs} \tag{5}$$

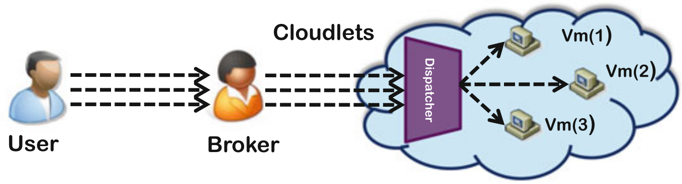

Fig. 3. Implemented architecture in Cloudsim. System contains two basic components. First is a broker allocated out of the cloud body which observes application workloads (from the cloud user's side) and communicates with cloud (provider) to adapt resources. The other is a dispatcher in cloud body which allocates VMs to workload appropriately.

4 Performance Evaluation

The proposed Learning Based algorithm (called Smart Virtual Machine Provisioner - SVMP) is implemented using Cloudsim which is suitable for resource provisioning simulating [15]. However this tool typically is not able to simulate Dynamic Virtual

[2] Million Instructions Per Second.

[3] Million Instructions.

Machine provisioning beside the disability to resource provisioning in application layer, which the authors purpose to simulate, so new components and attributes have been added to the simulator to enable it to handle Dynamic VM provisioning in application layer.

Apart from SVMP, a workload dispatcher was developed to dispatch user workloads (called cloudlets in the Cloudsim) among available VM instances. This dispatcher fulfills each VM with incoming cloudlets until VM utilization is under 80 %, the remaining 20 % is reserved for eventual heavy loads. With this method VMs would be utilized in a reasonable manner, moreover over provisioned VMs remain empty of load and can be easily shutdown. For some cases like web servers this threshold is considered 85 % [10, 16], but SVMP is designed for more general applications besides it is a learning based system and take some time to learn, so our threshold was set to 80 %.

Based on decision table (Table 1), if the algorithm was in the proper state it would do the right action, else it would update the probabilities into the proper state and do the action of the current state (although it is not the proper one) until it reaches (so Learns) the right action. The aggression of actions differs and learning system decides how many virtual machine(s) must be added or to be removed based on the average utilization amount.

Table 1. Decision table of the algorithm.

	Average utilization > 80	$80 \geq$ Average utilization≥ 70	Average utilization < 70
Probably condition	Under provisioned	Normal	Over provisioned
Proper state to be select	Inc	Norm	Dec
Action of the state	Add VM(s)	Do nothing	Remove VM(s)

To evaluate our proposed system, we performed three experiments using three different approaches. 1st and 2nd experiments demonstrate the cloud behavior using two specific static provisioning approaches. Experiment 3 shows the cloud behavior using the proposed dynamic VM provisioner.

The system has been tested under Normal Distribution Workload (Fig. 4). The workload starts from small amount of MI, then goes to a peak and then starts falling; this pattern happens a lot in real word with different peaks and slopes.

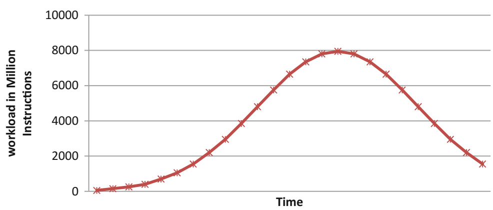

Fig. 4. Workload generation profile for all experiments.

As mentioned before, we have simulated our study using Cloudsim with three different experiments. First of all for demonstrating the studies we have defined a scenario as below (Table 2):

Table 2. SLA parameters for running the scenario

SLA parameters				
Workload type	Max VMs	VMs CPU core(s)	Core processing power	VMs RAM
CPU intensive	30	1	250 MIPS	512 MB

Cloud user deploys an application in the cloud on several virtual machines (Web server e.g.).

4.1 Experiment 1: Static Max Provisioning (Over Provisioning)

Experiment 1 describes behavior of the system under a static provision policy using the maximal virtual resource provisioning. Number of virtual machines is 30 instances. We have obtained this number considering the maximum virtual machine number provisioned by our dynamic approach. So the processing power is would be 12000 MIPS (Fig. 7), constantly. Figure 5 demonstrates average VMs utilization during this Experiment.

There is not any saturation area in this plot which means there is not any under provisioning. This is completely expectable because this is an over provisioning policy. However the highest average utilization of virtual machines that was obtained is 66.25 % (Fig. 5). This means high amount of waste in resources. This usual type of traditional resource provisioning for applications inflicts unnecessary cost to the application extender as a cloud user.

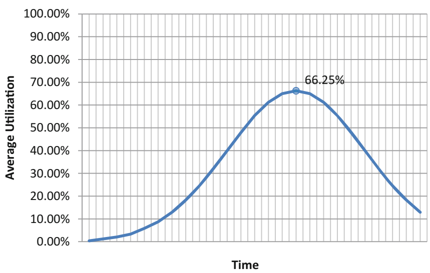

Fig. 5. Average utilization of provisioned virtual machines during Experiment 1.

4.2 Experiment 2: Static Mean provisioning

This section describes the obtained results from Experiment 2. A static provisioning using mean virtual resource provisioning policy was chosen. The term mean is referred to this provisioning policy because here the virtual machine number is the mean of minimum allocate able virtual machine (1) and maximum allocate able virtual machine (used in previous experiment). So the processing power is 6000 MIPS (Fig. 7), constantly. Obviously this is a conservative strategy which is cheaper than latter strategy. However this approach is not able to detect saturation in virtual machines so user suffers from under provisioning in time that workload is above average. The average utilization generally seems better (Fig. 6) comparing to Experiment 1, but both under and over utilization problems are still remained.

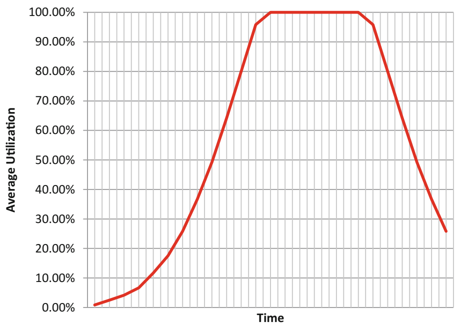

Fig. 6. Average utilization of provisioned virtual machines during Experiment 2.

4.3 Experiment 3: Dynamic Provisioning Using Proposed System

This section describes the results of Experiment 3 using our proposed algorithm, the SVMP (Smart Virtual Machine Provisioner). Unless two previous approaches this is an adaptive approach. The processing power is adapting to the workload and is not constant any more (Fig. 7). First system starts from normal state and tries to evaluate the environment and guesses the next state of the workload. The SVMP curve flat areas in Fig. 7 are the times that system is in normal or learning state.

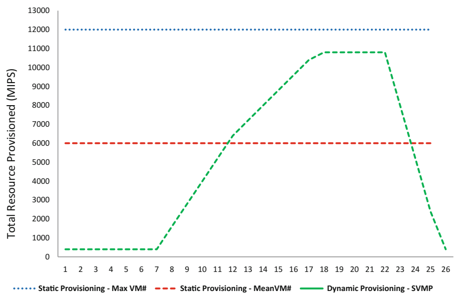

Fig. 7. Comparing total MIPS estimated and provisioned by different approaches.

SVMP tries to meet application needs in one hand and reducing cloud user cost in the other. By utilizing allocated virtual machines to user's workload, number of provisioned virtual resources to user is reduced. So here total utilization of virtual machines is the feedback parameter so becomes very important. As Fig. 8 demonstrates the SVMP tries to keep this utilization up and where ever this amount drops, it will bring it back up (Fig. 8. The Point **B**) by removing unnecessary allocated virtual machines. Besides that, SVMP detects under provisions– when utilization grows and goes to 100 % – and pulls it down by adding extra virtual machines up (Fig. 8. The Point **A**), so keeps Qos parameters.

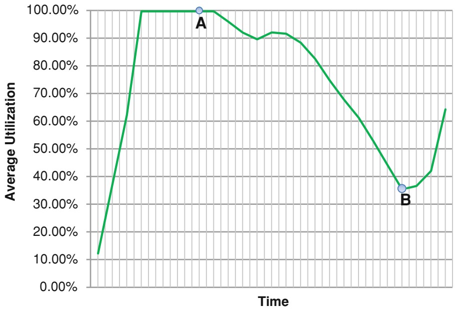

Fig. 8. Average utilization of provisioned virtual machines during Experiment 3. The SVMP detects application demands by monitoring utilization of virtual machines and scales up or down when resources are under or over provisioned, respectively.

From experimental result it can be extracted that SVMP owns not only the best average utilization and price (Fig. 9) even the least saturation time among the all (Figs. 6, 8). Both results show advantages of proposed dynamic provisioning approach (SVMP) comparing to the static provisioning method.

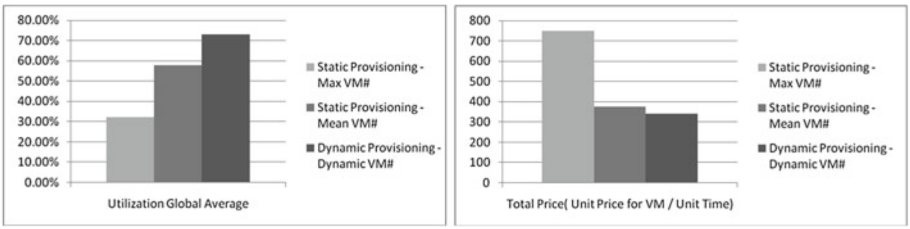

Fig. 9. Comparing total virtual machines utilization averages (Left side) and total virtual machines cost for the cloud user (Right side) in experimented approaches.

5 Conclusion

Dynamic resource provisioning enables cloud users to execute their tasks considering QoS elements in more cost effective way. A true application layer dynamic resource provisioner must predict the users' workload and provide the resources before workload arrival; this is important because providing resources in this layer (virtual machines) is not instantaneous and for a virtual machine instance it takes some time to start and become functional. The experiment presented in this paper addressed dynamic resource provisioning with a learning based system called SVMP to reduce the cost while keeping requirements of cloud user application. We are currently extending our system to employ other learning algorithm to improve performance of resource provisioning scheme.

References

1. Kaplan, J., Forrest, W., Kindler, N.: Revolutionizing Data Center Energy Efficiency. McKinsey & Company (2008)
2. Pinheiro, E., et al.: Load balancing and unbalancing for power and performance in cluster-based systems. In: Proceedings of the Workshop on Compilers and Operating *Systems* for Low Power (2001)
3. Elnozahy, E.N.M., Kistler, J.J., Rajamony, R.: Energy-efficient server clusters. In: Falsafi, B., VijayKumar, T.N. (eds.) PACS 2002. LNCS, vol. 2325, pp. 179–196. Springer, Heidelberg (2003)
4. Kusic, D., et al.: Power and performance management of virtualized computing environments via lookahead control. In: Proceedings of the 2008 International Conference on Autonomic Computing, pp. 3–12. IEEE Computer Society (2008)
5. Verma, A., Ahuja, P., Neogi, A.: pMapper: power and migration cost aware application placement in virtualized systems. In: Issarny, V., Schantz, R. (eds.) Middleware 2008. LNCS, vol. 5346, pp. 243–264. Springer, Heidelberg (2008)
6. Van, H.N., Tran, F.D., Menaud, J.M.: Performance and power management for cloud infrastructures. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD) (2010)
7. Lin, C.-C., Liu, P., Wu, J.-J.: Energy-aware virtual machine dynamic provision and scheduling for cloud computing. In: 2011 IEEE International Conference on Cloud Computing (CLOUD) (2011)
8. Lin, W., et al.: A threshold-based dynamic resource allocation scheme for cloud computing. Procedia Eng. **23**, 695–703 (2011)
9. Calheiros, R.N., Ranjan, R., Buyya, R.: Virtual machine provisioning based on analytical performance and QoS in cloud computing environments. In: 2011 International Conference on Parallel Processing (ICPP) (2011)
10. Iqbal, W., et al.: Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Future Gener. Comput. Syst. **27**(6), 871–879 (2011)
11. Jeyarani, R., Nagaveni, N., Vasanth Ram, R.: Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence. Future Gener. Comput. Syst. **28**(5), 811–821 (2012)
12. Zaman, S., Grosu, D.: An online mechanism for dynamic VM provisioning and allocation in clouds. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD) (2012)
13. Islam, S., et al.: Empirical prediction models for adaptive resource provisioning in the cloud. Future Gener. Comput. Syst. **28**(1), 155–162 (2012)
14. Papoulis, A., Pillai, S.U.: Probability, Random Variables and Stochastic Processes, vol. 1, 4th edn., 852 p. McGraw-Hill, Boston (2002)
15. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: International Conference on High Performance Computing & Simulation 2009 (HPCS '09), (2009)
16. Allspaw, J.: The Art of Capacity Planning: Scaling Web Resources. O'Reilly Media, Sebastopol (2008)