LOAD BALANCING IN AUTO SCALING-ENABLED CLOUD ENVIRONMENTS

Nguyen Hong Son [1], Nguyen Khac Chien [2]

[1]Department of Information and Communication Technology, Post and Telecommunication Institute of Technology, Ho Chi Minh City, Viet Nam
[2]University of the People's Police, Ho Chi Minh City, Viet Nam

ABSTRACT

Cloud computing is growing in popularity and it has been continuously updated with more improvements. Auto scaling is one of such improvements that help to maintain the availability of customer's subscribed cloud system. The appearance of an auto scaling mechanism in the cloud system with many existing system mechanisms is an issue that needs to be considered. Because, normally, there is no free drawbacks whenever a new part is added to a certain stable system. In this paper, we consider how existing load balancing and auto scaling impact on each other. For the purpose, we have modeled a cloud system with an auto scaler and a load balancer and implementing simulations based on the constructed model. Also based on the results from the computer simulations we proposed about choosing load balancers for subscribed cloud system with auto scaling service.

KEYWORDS

Cloud Computing, Load Balancing, Auto Scaling, Workload

1. INTRODUCTION

Cloud computing has become an attractive technology not only for service providers but also for a lot of subscribers around the world. The technology is considered as a solution that provides many key features such as on-demand self-service, global network access, distributed resource pooling, scalable, and measured service [1]. So it facilitates companies to reduce the cost for running their information system and to allow service providers to reach their target customers much more effectively. Today it is easy to get the great cloud services from brand-name cloud computing providers.

In terms of operating system, cloud computing must be implemented some major mechanisms for serving more effectively, such as scheduling, load balancing, and migration. The performance of cloud quite depends on the manner of these mechanisms. So, it makes these topics to be specially concerned by researching community. Traditionally, scheduling is referred to the job of sharing executing resources of hosts among virtual machines and allotting executing power of virtual machine to among tasks running on. Whilst schedulers are often busy with duty of resource allocation, the load balancers focuses on their own function of driving user requests to proper virtual machine. The user requests cause to generate items what often called tasks or loads on virtual machines. The main goal of a load balancing mechanism is to equitably distribute the loads into virtual machines inside a data center. It ensures that there is no virtual machine incurring much more load than others. By the way, a load balancer can maximize the throughput of clouds and minimizes the delay in serving user requests. Another special mechanism that coexists with load balancing is the virtual machine migration mechanism. It is really into energy conservation as hosts with few virtual machines will be turned off after the virtual machines migrated to proper hosts. Traditionally, the system mechanisms were designed to function in cloud habitat with definite resources. It means that definite resources are fixedly assigned to every cloud account. In the habitat, when user requests surge up and exceed the assigned executing

DOI: 10.5121/ijccsa.2017.7502

power, the leased system falls into overload state and the quality of service is degraded dramatically. To overcome the situation, today's cloud service providers have supplied to their customers a special service called auto scaling. The new service is implemented by an auto scaling mechanism which allows adding temporary resources to the subscribed service packages in case of resources being exhausted. Auto scaling mechanism is new technique added to stable cloud environment with inherent system mechanisms. The question is whether problems arise or not if cloud still runs with current version of the system mechanisms. In fact the kind of load balancer is an item which subscribers have to choose in order to configure the auto scaling service in the EC2 cloud [14]. This denotes that load balancing has a significant role in providing an auto scaling service to cloud subscribers and that there exists work that should be done with traditional load balancing for adapting to auto scaling. To clarify the matter, we look for what happen if we vary the kind of load balancing with auto scaling and also look for what things make the cloud performance to be worse. Firstly, we model the cloud which includes both load balancer and auto scaler. The load balancer is implemented by using typical load balancing algorithms such as round robin and active monitoring. The auto scaler is applied the state of art auto scaling algorithm that is based on thresholds. Then, we simulate various scenarios on the cloud model and observe special things. The results from our study help to propose solutions for improving load balancing function in auto scaling-enabled cloud environments.

The rest of the paper is organized as follow: Section 2 relates published researches about load balancing and auto scaling in cloud computing. A model of cloud system with load balancer and auto scaler is presented in section 3. Computer simulations based on the model in section 3 is included in section 4. The paper is closed by some key conclusions in section 5.

2. RELATED WORKS

Load balancing has exposed its crucial role and a complex issue in cloud computing. There are many challenges for developing an efficient load balancer. Load balancing schemes have to control incoming loads so that all processing cores in the system or every host in the cloud execute approximately an equal amount of workloads at any instant of time [2]. Many endeavors relating to the topic have been made by researchers, as mentioned in [3]. Researches in [4][5] proposed active monitoring-based algorithms. The algorithms keep track of the current load on all machines in cloud. On arriving of load, it looks for what machine has the lowest load and allocates the load to it. Thereby, the stronger machines will be assigned more load than weaker machines. Therefore the goal of load balancing may be achieved. However, in cloud environment load balancing strategies should be considered in two levels of scheduling, the host level and the virtual machine level. They also depend on scheduling policy chosen in each level, which is the time-shared scheduling or the space-shared scheduling, as result of estimating the surplus capacity of machine is different between the scheduling styles. So as to concern the scheduling commodity, the research in [6] proposed a load balancing scheme that treated the scheduling cases separately in its calculations. For the sake of improving cloud performance, [7] also proposed a novel mechanism to allot tasks to virtual machine and a novel algorithm based on bio-inspired techniques such as Particle Swarm Optimization (PSO), Cat Swarm Optimization (CSO) for allocating resources effectively to tasks.

Similar to load balancing, auto scaling raises many challenges for endeavors of achieving the given goals of the function. Therefore, the topic also caught more attention of researchers, as reviewed in [8][9]. The first problem of auto scaling is how to lease the right amount of resources that cost on pay as you go basic. Moreover, how to return the leased amount of resources punctually in order to keep the cost low is the second problem of the function. Up to now, to solve both the problems has not been an easy task. In doing with the problems, in [10], authors have proposed a new auto scaling mechanism called BATS, which satisfies the limitation of

budget while minimizes service delay. Also, research in [11] proposed an auto scaling system based on time-series prediction algorithms. So as to achieve the prediction accuracy of the auto scaling system, a special key was made called self-adaptive prediction suite which can automatically select the most suitable prediction algorithm based on the incoming workload pattern.

The relationship between load balancing and auto scaling was mentioned in [12]. It described the way how brand-name cloud systems such as EC2, Azure, and RackSpace provide their subscribers with auto scaling service. Especially, EC2 uses a monitor called Cloudwatch to emit metrics alarm which in turn starts or stop virtual machine instances. Related to both key functions, [13] also gave a comparison between various cloud providers on features of the functions. However, it also did not clarify how load balancing and auto scaling influence each other.

3. MODELING CLOUD DATA CENTER WITH AN AUTO SCALER AND A LOAD BALANCER

Let data center (DC) have N virtual machines (VMs), called VM_1, VM_2,….,VM_N.

$$DC(VM_1, VM_2,\ldots,VM_N)$$

Every virtual machine has a processing power of P_i and a workload $L_i(t)$ at any time t.

$$VM_1(L_1(t),P_1)$$
$$VM_2(L_2(t), P_2)$$
$$\ldots\ldots$$
$$VM_N(L_N(t), P_N)$$

We assume that the data center applies the time-share scheduling policy for its scheduling levels, host level and virtual machine level. Also, it just creates a limited number of VMs on each host in order to ensure enough processing power for VMs. Let l(t) be the workload at time t that is generated by incoming request to load balancer and the load balancer allotted it to a certain VM based on specific load balancing algorithm. So if $L'_i(t)$ is the workload of a VM_i after the load balancer has distributed the load l(t), which is computed by following expression:

$$L'_i(t) = L_i(t) + a_i(t).l(t) \; ; \; i\epsilon 1..N$$

with $a_i(t)$ taking values of 1 or 0, depending of whether VM_i matches with the right condition of the load balancing algorithm or not. It has value 1 if the condition is true and vice versa.

$$a_i(t) = \begin{cases} 0 & VM_i \text{ is not assigned} \\ 1 & VM_i \text{ is assigned} \end{cases}$$

If the load balancer implements the round robin algorithm, incoming workloads are assigned to each VM in circular order. With the active-monitoring load balancing algorithm, the load balancer has to find the VM with the least workload in order to allocate new load to it. In this case, $a_i(t)$ is as follow:

$$a_i(t) = \begin{cases} 1 & VM_i \text{ has the lowest load in} \\ 0 & Other \end{cases}$$

The major purpose of load balancers is to allocate fairly workloads among VMs inside a data center. It should take a way that there is no VM to incur more loads than others. Therefore, we also consider the equitable allotment as one of criterions for comparing between load balancers. A load balancer with the smallest workload deviation between among VMs inside a data center is

the best load balancer. It also results in minimum delay when the workload deviation gets minimal value. In the research, we specially concern the service time left of the VM which means necessary amount of time for executing completely current workload in the VM. The service time left of VM, called $T_{left,i}$, is calculated by formulation below:

$$T_{left,i}(t) = \frac{L_i(t)}{P_i}$$

Let Dev(t) is the difference between the lowest service time left and the highest service time left among the VMs inside the considered system. The item will be checked in various simulation scenarios later and calculated such as below.

$$Dev(t) = \left(\frac{L_i(t)}{P_i}\right)_{max} - \left(\frac{L_i(t)}{P_i}\right)_{min}$$
$$Dev(t) \to 0, \quad t \to \infty$$

In general, auto scaler frequently monitors a data center and gathers information about the system health. When the subscribed system is going to fall into overload situation it orders to supply power by adding a new machine to the system. Using preset thresholds for making decisions is common method that has been applied in today's auto scalers. The system metric that is set in thresholds could denote the status of system. The metrics are different between various auto scaling algorithms. For the research, we use the average of service time left in the leased system of subscribers as a metric for auto scaling. Let $T_{av}(t)$ be the average of the service time left at time t, it is calculated as follow:

$$T_{av}(t) = \frac{\sum_i^N L_i(t)}{\sum_i^N P_i}$$

If Let N'(t) be the number of machines in the considered system at time t and it is expressed as below:

$$N'(t) = N(t) + n(t)$$

n(t) is the number of machines that are temporarily suplemented the considered system at time t. The auto scaler obeys the following rule:

$$n(t) = \begin{cases} 0 & Threshold_{low} \le T_{av}(t) \le Threshold_{high} \\ 1 & T_{av}(t) > Threshold_{high} \\ -1 & T_{av}(t) < Threshold_{low} \end{cases}$$

Normally, the load balancer works with a fixed number of machines in a subscribed system but it will have to work with altering the number of machines in the system with auto scaller. How to keep load balancer knowing right number of machines is an issue in this context. It also raises

another issue that the number of machine may rapidly change due to fluctuation of workloads. While the issues may take place, the research is inspired to see if there are any impacts of the issues on performance of load balancers and efficiency of cloud systems.

4. COMPUTER SIMULATIONS

We developed a cloud simulation program that based on the model in the section 3. The simulation program also includes all key components of a cloud system such as host, virtual machine, load balancer, auto scaler, etc. In this simulation, we implement two load balancing algorithms: round robin and active monitoring. The simulation includes an auto scaler that works based on thresholds as mentioned above.

In the simulation, we construct a subscribed cloud system with four VMs which has 500MIPS CPU, 2Gbyte RAM and 1Gbps bandwidth. The four VMs are minimum number of VMs in the considered system that they belong to customer's cloud service package. The auto scaler will supplement provisional VMs when the cloud system is going to be overloaded and it also removes the VMs when the cloud system goes back to normal state. So it will shift the number of VMs forth and back in the range of greater than number of four. In the simulation scenario, we set the upper threshold equals to 6 seconds and the lower threshold equals to 2 seconds. We submit the same workloads, as described in figure 1, to load balancer in both cases of round robin and active monitoring algorithm. The workloads have fluctuated more erratically. There were some sharp increases in loads, before reaching a peak of approximately 4000MI, but then the workloads dropped markedly again the following time.

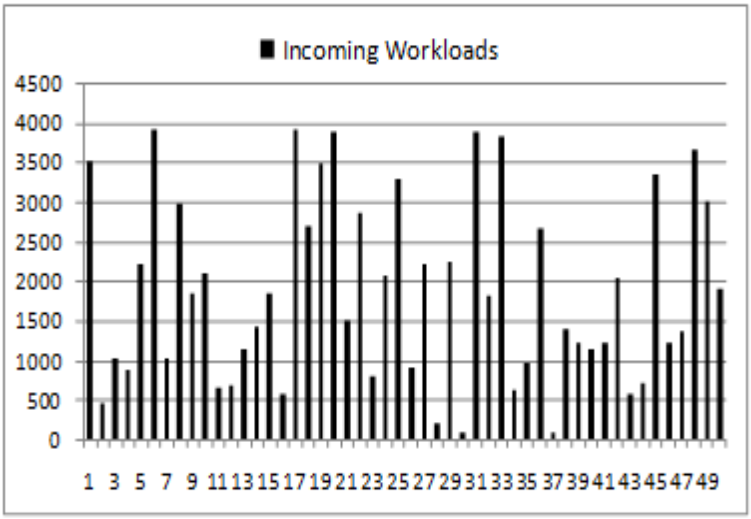

Figure 1. Incoming workloads are distributed by load balancer

Firstly, we would like to see how auto scaler supplementing VMs to subscribed cloud system with round robin load balancer and with active monitoring load balancer. The results of running with round robin load balancer are described in figure 2 and the results from running with active monitoring load balancer is described in figure 3.

The figure 2 shows that auto scaler had to supplement VMs to the cloud system several times, from 1 to 2 VMs, and the VMs make a long stay in the system. While it is just one time to add one VM to the system done by auto scaler in case of using active monitoring load balancer, as

showed in figure 3. Moreover, the provisional VM makes no long stay in the subscribed cloud system.

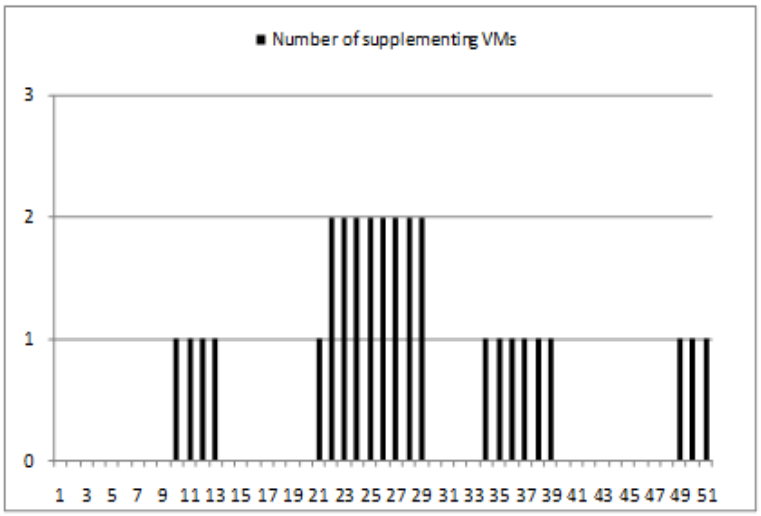

Figure 2. Number of supplementing VMs in case of using round robin load balancer.

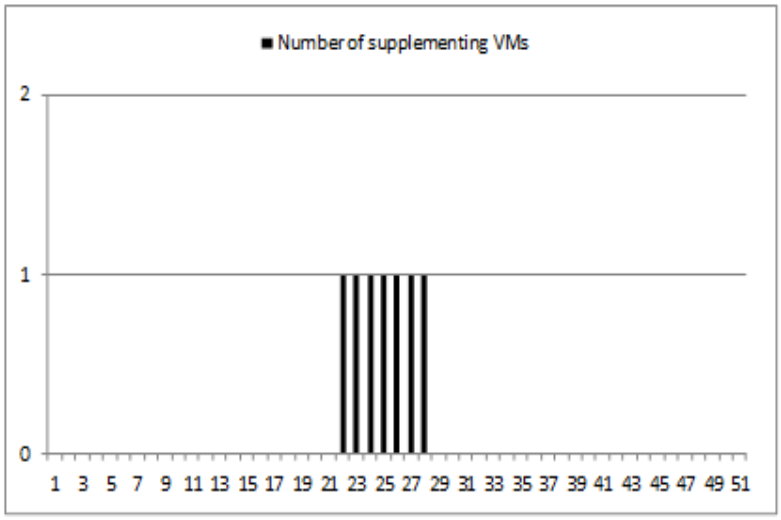

Figure 3. Number of supplementing VMs in case of using active monitoring load balancer.

Secondly, we focus on how different in deviation Dev(t), as defined in section 3, between round robin load balancer and active monitoring load balancer. The results from both cases are depicted in figure 4. The figure shows that deviations from round robin (RR) load balancer are always greater than those from active monitoring (AM) load balancer. The item can reach value of 25.2 with round robin algorithm while the greatest value in active monitoring algorithm is about 15.5. In addition, the figure also denotes surges in deviations corresponding to every time supplementing VMs to the cloud system. It also shows that the surge from AM load balancer is not more greater than average value whilst those is quite high and long with RR load balancer.

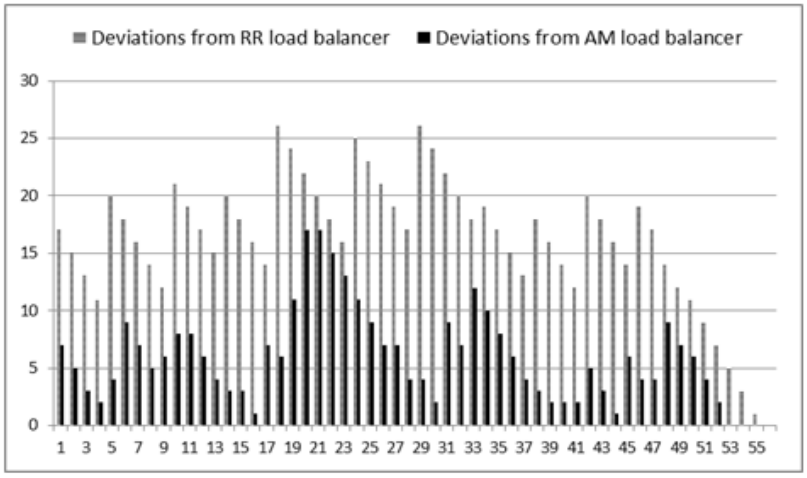

Figure 4. Deviations, Dev(t), from RR load balancer and AM load balancer.

5. CONCLUSIONS

Load balancing in cloud environments with auto scaling has been just considered. There are kinds of load balancers that may cause auto scaler adding more provisional VMs to subscribed cloud system than others, round robin load balancer, for example. This results in using cloud resources ineffectively and subscribers have to paid more money. Our research also denotes that load balancers applying algorithms based on current workload of VMs in cloud system have well adapted to cloud environment with auto scaling. For instance, active monitoring load balancer helps not only to keep unbalance level between VMs in small, but also to use cloud resources effectively. So, we should consider choosing load balancers based on workload of VMs in auto scaling-enabled cloud environment, especially auto scalers using algorithms based on thresholds.

REFERENCES

[1] P. Mell and T. Grance, (2009) "The NIST definition of Cloud Computing" version 15. National Institute of Standards and Technology (NIST), Information Technology Laboratory

[2] Rodrigo, N. C., Rajiv, R., Anton, B., Cesar, A. F. D. R., and Rajkumar, B. (2011), CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, *Software: Practice and Experience,* Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA

[3] Minxian Xu, Wenhong Tian, Rajkumar Buy, (2017) A survey on load balancing algorithms for virtual machines placement in cloud computing, Concurrency and Computation: Practice and Experience Volume 29, Issue 12

[4] Soumya, R. J., Zulfikhar, A., (2013) "Response Time Minimization of Different Load Balancing Algorithms in Cloud Computing Environment", International Journal of Computer Applications (0975–8887), Volume 69, No. 17

[5] Jasmin, J., Bhupendra, V., (2012) "Efficient VM load balancing algorithm for a cloud computing environment", International Journal on Computer Science and Engineering (IJCSE)

[6] Nguyen Khac Chien, Nguyen Hong Son, Ho Dac Loc, (2016) Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing, The IEEE ICACT, Phonix, Korea

[7] Rajeev Kumar; Tanya Prashar, (2016) A bio-inspired hybrid algorithm for effective load balancing in cloud computing, Int. J. of Cloud Computing, Vol.5, No.3, pp.218 – 246

[8] Tania Lorido-BotranEmail authorJose Miguel-AlonsoJose A. Lozano, (2014) A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments, Journal of Grid Computing, December 2014, Volume 12, Issue 4, pp 559–592, Springer

[9] Hanieh Alipour, Yan Liu, Abdelwahab Hamou-Lhadj, (2014) Analyzing Auto-scaling Issues in Cloud Environments, Proceeding CASCON '14 Proceedings of 24th Annual International Conference on Computer Science and Software Engineering pp 75-89

[10] A. Hasan Mahmud, Yuxiong He, Shaolei Ren, (2015) BATS: Budget-Constrained Autoscaling for Cloud Performance Optimization, 2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems

[11] Ali Yadavar Nikravesh, Samuel A. Ajila and Chung-Horng Lung, (2017) An autonomic prediction suite for cloud resource provisioning, Journal of Cloud Computing Advances, Systems and Applications 6:3 DOI 10.1186/s13677-017-0073-4, SpingerOpen

[12] Caron, Eddy & Rodero-Merino, Luis & Desprez, Frédéric & Muresan, Adrian. (2012). Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds. Cloud computing: methodology, systems, and applications. 10.1201/b11149-17

[13] Ashalatha R and Jayashree Agarkhed, (2015) Article: Evaluation of Auto Scaling and Load Balancing Features in Cloud. International Journal of Computer Applications 117(6):30-33, May 2015

[14] Michael Pleshakov (2017), Load Balancing AWS Auto Scaling Groups with NGINX Plus, Blog Tech, NGINX Inc.

[15] Naidila Sadashiv and Dilip Kumar S M, (2017) A Baye's Theorem Based Node Selection for Load Balancing in Cloud Environment, International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 7, No. 1

AUTHORS

Nguyen Hong Son, received his B.Sc. in Computer Engineering from the University of Technology in HCM city, his M.Sc. and PhD in Communication Engineering from the Post and Telecommunication Institute of Technology Hanoi. His current research interests include communication engineering, network security, computer engineering and cloud computing.

Nguyen Khac Chien, received his master degree in Computer Science from the University of Natural Sciences in HCM City in 2008. He is currently a lecturer at the University of the People's Police, and is doing a PhD candidate in Computer Engineering at the PTIT, Hanoi. His research interests include Auto-Scaling, VM Migration and Load balancing in cloud computing.