Elastic VM: A Fine-grained Scalability Architecture for Virtualized Environment

Wesam Dawoud, Ibrahim Takouna, and Christoph Meinel
Hasso Plattner Institute, Potsdam University, Potsdam, Germany

Abstract

The rapid growth of E-Business and the frequent changes in sites contents pose the need for rapid and dynamic scaling of resources. Currently, cloud computing infrastructure enables agile and dynamic scalability of resources. However, current implementation of clouds' scalability is based on the Virtual Machine (coarse-grained) as a scaling unit which often leads to over-provisioning of resources. Hence, we propose Elastic VM (fine-grained) scaling architecture. It implements the scalability into the VM resources level. So, instead of scaling-out by running more VMs instances, our architecture scales-up the VM's resources themselves (e.g., number of cores and memory size) to cope with the workload demand.

Experimental results show that the Elastic VM scaling architecture is able to reduce scaling overhead, maintain a high throughput, mitigate Service Level Objectives (SLOs) violation, and enable a simple scalability to broader range of applications including databases.

Introduction

To maintain a good quality of service (QoS), system administrators should provision adequate resources to cope with workload demand fluctuations. However, over provisioning implies extra cost and decreases the business profit. On the other hand, under-provisioning degrades the QoS, irritates the clients, and consequently retreats the E-business durability.

The later advance in virtualization technology software, e.g. Xen and VMware , enables cloud computing environment to deliver agile, scalable, and low cost infrastructures. However, as seen in Figure 1, there is a gap between the actual demand and the automated resources allocation implemented by current scalability architectures in the clouds. Our goal in this research is to reduce this gap. From the provider side reduction the gap will lead to running less hardware, consuming less power, and contributing less in CO2 emissions [1]. From the client side this gap reduction will be translated into reduction in the price.

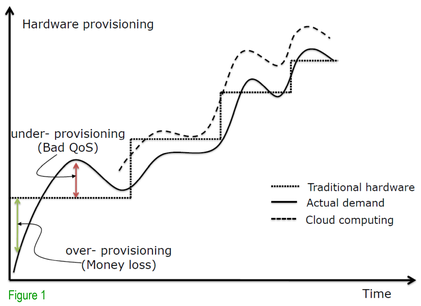

Figure 1

Multi-instances Scaling Architecture

Current implementation of the scalability by Amazon EC2 and GoGrid is abstracted in Figure 2. Users' requests are directed to a load-balancer which forwards them to available VM instances (i.e., VM1 to VMn). To maintain a determined QoS, a controller monitors performance metrics (e.g., CPU utilization) and scale-out VMs to cope with workload demand surge or scale-down VMs to reduce the cost.

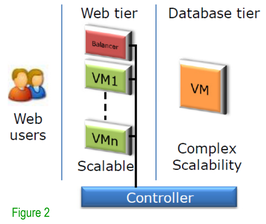

Figure 2

Elastic VM Scaling Architecture

Elastic VM is a VM that supports scaling resources on-the-fly without interrupting the service or rebooting the system. To cope with workload demand, as shown in Figure 3, a controller monitors tiers performacnce. If the monitored performance metric (e.g., CPU utilization) exceed a specified threshold, the controller scale up the virtual machine resources (e.g., number of cores) to maintain an acceptable system performance.

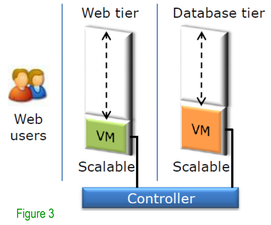

Figure 3

Multi-instances vs. Ealstic VM Scaling Architectures

Multi-instances architecture	Elastic VM architecture
Implies running load-balancer (i.e., additional consumption of resources)	No need for running additional VM instance as a Load-balancer
Limited to specific applications	Applicable to any tier
Applies VM as a scaling unit (coarse-grained scale)	Fine-grained scaling while it implements scaling into VM resources
Scaling-down can interrupt sessions based web connections	Supports sessions based web connections
Booting time of VMs, to scale-out, increases the overhead	Eliminates the overhead caused by booting VMs
Scale-out overhead causes SLO violation and decreases the throughput	Scale-up vertically reduces SLO's violation and maintains higher Throughput
Both software and hardware load-balancer can be a single point of failure	Elastic VM itself could be a single point of failure
It supports business at all scales: small, medium, and big	Elastic VM is limited to one physical host, which limits it to

Experiment Setup: Testing Web-tier Scalability

The goal of the following experiments is to compare web-tier scalability using Multi-instances with web-tier scalability using Elastic VM scalability architecture. The comparison includes total throughput and SLO violation. For this experiment we assume (20 ms maximum response time) as a SLO. It is measured as the difference between the moment of packet arrival to network interface of physical host until the moment of getting out with the result.

To examine web-tier scalability without any influence from database-tier, we ran RuBBoS benchmark [2] with browse only no search transitions traffic pattern. This traffic pattern is selected because it casts high workload to web-tier and a low workload to database-tier which prevents database-tier from being a bottleneck at any stage of the experiment.

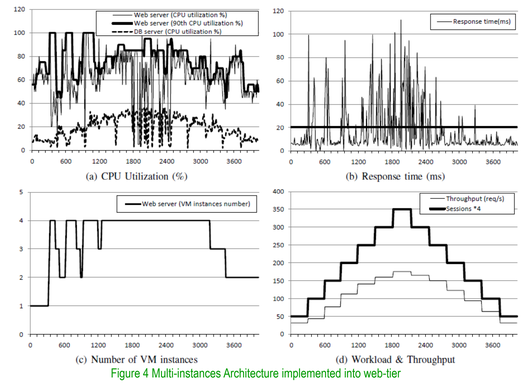

Figure 4 Multi-instances Architecture implemented into web-tier

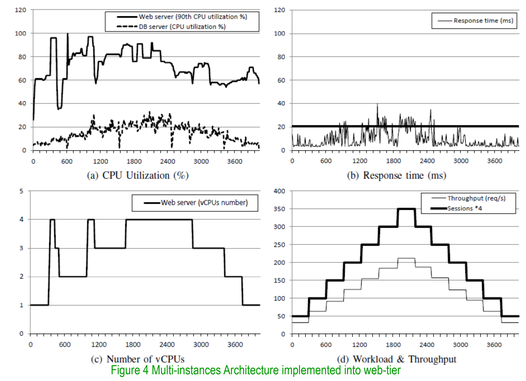

Figure 4 Multi-instances Architecture implemented into web-tier

Sessions number	Elastic-VM throughput (req/sec)	Multi-instances throughput (req/sec)	Multi-instances throughput degradation (%)
400	64	44	31
600	92	77	16
800	125	113	10
1000	155	141	9
1200	184	160	13
1400	212	176	17
1200	187	165	12
1000	157	150	4
800	124	123	1
600	95	94	1
400	64	64	0

Challenges

In spite of the fact that Elastic VM scaling architecture offers many advantages compared with Multi-instances scaling architecture, it posses many challenges that should be dissolved:

1- Some applications could be unaware of dynamic resources scaling, especially memory scaling. Fortunately, many researches are directed to optimizing applications parameters according to available resources like [3], [4], [5], and [6]. Such approaches can be incorporated into our architecture to tune applications parameters for optimum performance after each scaling.

2- Currently, Elastic VM is limited to one host. However, If a global policy is implemented to reallocate VMs according to host utilization, it will be possible to move VMs with lower load into another hosts to make more room for the overloaded Elastic VM to scale-up. Moreover, integrating both Elastic VM and Multi-instances architecture together can come up with fine-grained scaling architecture which is unlimited to one physical host.

Conclusion & Future work

The presented work suggests Elastic VM scaling architecture as dynamic and fast scaling architecture. Experiment results proved that Elastic VM reduces the provisioning overhead, mitigates SLOs violations, and maintains a higher throughput. Moreover, it enables scaling applications, such as databases, with lower cost and complexity.

Our immediate future work includes developing a global management policy for VMs management. It does not only consider scaling VMs in place but also relocation of VMs to physical hosts with less utilization. We also study integrating both Multi-instances and Elastic VM scaling architecture to enable rapid and fine-grained scaling architecture which is unlimited to one physical host.

References

[1] P. Johnson and T. Marker, "Data Center Energy Efficiency Report Product Profile," 2009.
[2] C. Amza, E. Cecchet, A. Ch, A. L. Cox, S. Elnikety, R. Gil, J. Marguerite, K. Rajamani, and W. Zwaenepoel, "Bottleneck Characterization of Dynamic Web Site Benchmarks," 2002.
[3] W. Dawoud, I. Takouna, and C. Meinel, Elastic VM for Cloud Resources Provisioning Optimization, ser. Communications in Computer and Information Science, A. Abraham, J. Lloret Mauri, J. F. Buford, J. Suzuki, and S. M. Thampi, Eds. Springer Berlin Heidelberg, 2011, vol. 190
[4] X. Liu, L. Sha, Y. Diao, S. Froehlich, J. L. Hellerstein, and S. Parekh, "Online Response Time Optimization of Apache Web Server," pp. 461–478, 2003.
[5] D. Wiese, G. Rabinovitch, M. Reichert, and S. Arenswald, Autonomic tuning expert, ser. CASCON '08. New York, New York, USA: ACM Press, 2008.
[6] D. N. Tran, P. C. Huynh, Y. C. Tay, and A. K. H. Tung, "A new approach to dynamic self-tuning of database buffers," ACM Transactions on Storage, vol. 4, no. 1, pp. 1–25, May 2008.

Contact information

Wesam Dawoud
Hasso Plattner Institute
Potsdam University. Potsdam, Germany

Telephone: +493315509532
Email(s): wesam.dawoud@hpi.uni-potsdam.de