doi:10.2498/iti.2013.0570

Does the Performance Scale the Same as the Cost in the Cloud

Monika Simjanoska, Goran Velkoski, Sasko Ristov and Marjan Gusev
Ss. Cyril and Methodius University, Skopje, Macedonia
Faculty of Information Sciences and Computer Engineering
E-mail: m.simjanoska@gmail.com, velkoski.goran@gmail.com
sashko.ristov@finki.ukim.mk, marjan.gushev@finki.ukim.mk

Abstract. *Cloud computing is a paradigm that offers on-demand scalable resources with the "pay-per-usage" model. Cloud service providers' price rises linearly as the resources scale. However, the main challenge for the cloud customers is "Does the performance scale as the price for the rented resources in the cloud"? Also, how does the performance scales for different server load? In this paper we analyze the performance and the cost of a web service that utilize both memory and CPU varying the server load with different message size and number of concurrent messages in order to determine the real cost of rented CPU resources. The results show that the web service's cost rises linearly with the resources, i.e. the lowest cost is determined while hosted on one CPU.*

Keywords. Cloud Computing, Web Services, Performance, Resources, Cost

1. Introduction

Cloud Computing is a model that enables ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Cloud computing as a new type of infinitely scalable computing, offers a pay-per-usage payment which also scales to the amount of rented resources. This usage-based pricing model offers several advantages, including reduced capital expense, a low barrier to entry, and the ability to scale up as demand requires, as well as to support brief surges in capacity [2]. Cloud service providers (CSPs) guarantee the availability of the rented resources to the customers by defining Service Level Agreements (SLAs). However, a guarantee of scalable and sustainable performance is missing in the SLAs [3]. Therefore, the problem of scalable performance is a very challenging field of study and is also beneficial for both the customers and the CSPs. In this paper we perform a research to find the trade-off between the customer's monetary costs for resources and the achieved performance.

Most of the related studies consider the CSPs' benefits, whereas the customer's expenses and privileges are not fully covered in the literature. In our research, we assume that the performance linearly depends on the amount of acquired resources, and therefore we follow the CSPs' renting model presented in Table 1 which informs of the current offers for renting the following virtual machine (VM) instances (USD per rented hour): Azure [4], Google [5] and Amazon [6].

Table 1: VM instance types and price simulation

Type	1 VM	2 VMs	4 VMs	8 VMs
Windows Azure	0,08	0,16	0,32	0,64
Google Compute	0,151	0,302	0,604	1,208
Amazon EC2	0,065	0,13	0,26	0,52
Scaling factor	1	2	4	8

In order to simulate realistic occasions of renting different amount of resources, we prepared the cloud testing environments following the offers presented in Table 1. Each cloud environment hosts the $Sort$ web service which sorts the concatenation of two input strings.

We focus our research into two directions: 1) how does the performance scale if the resources are scaled for the same server load; 2) is there a region of server load where maximum performance is achieved paying the same price.

The rest of the paper is organized as follows. In Section 2 we present a brief review of the related work. The methodology is defined in Section 3 and hence it is used in Section 4 to perform the testing. We compare the results for the $Sort$ web service with memory demanding only web service in Section 5. Eventually, we derive conclusions over the results and we present our ideas for a future research extension in the final Section 6.

2. Related Work

In this section we present some of the literature we found to be closely related.

Most of the literature that examine cloud's computing issues take into account CSPs costs for offering the cloud computing solution. Either they are investigating the influence of the cloud's energy consumption [7, 8, 9, 10], or they are discussing other issues and challenges as automated service provisioning, virtual machine migration, server consolidation, traffic management, data security, charging model, etc. [11, 12].

We are interested in the customers' benefits of the on-demand resource provisioning and the pay-per-usage pricing model. Thereto, we aim to investigate if there is a case when the customer can achieve maximum performance with minimal costs. Similar research of this kind is presented by De Assuncao et al. [13], where the authors present several scheduling strategies for balancing between performance and usage cost, and how much they improve the requests' response times. Andrzejak et al. in [14] formulated a probabilistic model that enables a user to optimize monetary costs, performance and reliability, given the user's SLA constraints as resource availability and deadline for job completion. Other authors developed a service that is able to perform the cost determination for scientific applications in cloud computing environments [15].

Simjanoska et al. in [16], performed cost vs performance basic research using only the $Concat$ memory demanding web service. The results showed that $Concat$ web service provides the lowest cost when hosted on two CPUs, which disproves the hypothesis of proportionality between the cost and the amount of rented resources. The results intrigued us to go deeper into the problem and widen the research introducing the $Sort$ web service, which is both memory demanding and computation intensive. Considering the performance issues, related researches were performed in [17, 18], where the authors used various web services in different cloud environments.

3. The Methodology

In this section we present the methodology used for testing in order to achieve reliable results.

3.1. Testing Environment

We used client-server architecture as a testing environment deployed in the open source cloud platform OpenStack [19] using KVM hypervisor to instantiate VM instances. The client and server node are installed with Linux Ubuntu Server 12.04 operating system. Hardware computing resources consist of Intel(R) Xeon(R) CPU X5647 @ 2.93GHz with 4 cores and 8GB RAM. The virtual machine (VM) instances consist of Linux Ubuntu Server 12.04 operating system and Apache Tomcat 6 as the application server. To minimize the network latency we placed the client and the VMs in the same LAN segment [20].

3.2. Environment Configuration

In order to simulate various number of provided resources (CPU cores), we defined three different cloud environments:

- *Test Case 1*: VM instance with 1 CPU;
- *Test Case 2*: VM instance with 2 CPUs;
- *Test Case 4*: VM instance with 4 CPUs.

Each VM hosts the $Sort$ web service.

3.3. Testing Procedure

SoapUI [21] generates different server load with N messages, each with size of M bytes, using variance 0.5. This means that the number of threads will vary by $N/2$, i.e. the number of threads will increase to $3 \cdot N/2$, then decrease to $N/2$, and finally end with N within 60 seconds, i.e. the end of the test. The range of parameters M and N is selected such that web servers in VM instances work in normal mode without replying error messages. The web service is loaded with N = 12; 100; 500; 752; 1000; 1252; 1500; 1752 and 2000 requests per second for each message size M = 0; 1; 2; 4; 5 and 6. In order to simulate different connections per core we divide the N concurrent messages in four groups of $N/4$ messages each.

3.4. Performance and Cost Measurements

We measure the server's average response time $T(n)$ for each parameter M and N in order to express the cloud's performance, where n is the total number of processors used. Then we calculate the cost $C(n)$ as defined in (1).

$$C(n) = T(n) * n \tag{1}$$

We research if the performance is proportional to the number of rented resources, and consequently, if the web service' total cost $C(n)$ is the real cost of rented CPU resources. For this purpose we introduce and calculate *Relative Cost* of the scaling, expressed in (2), (3) and (4), as ratio of test cases with VMs 2 and 1 CPU cores; VMs 4 and 1 CPU cores; and VMs with 4 and 2 CPU cores, correspondingly.

$$R_{21} = \frac{C(2)}{C(1)} \tag{2}$$

$$R_{41} = \frac{C(4)}{C(1)} \tag{3}$$

$$R_{42} = \frac{C(4)}{C(2)} \tag{4}$$

An ideal expectation will be the proportional scaling, i.e. when $R_{21} = 2$ and $R_{41} = 4$. Any deviation from these expectations will lead to new conclusions in this research.

4. The Results of the Experiments

In this section we present the results of the experiments for each cloud environment distinctively.

4.1. Analysis of Response Time

$Sort$ web service has been hosted in VM instances with 1, 2 and 4 CPU cores.

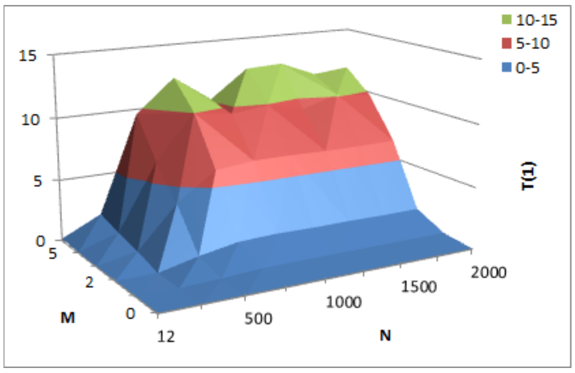

Figure 1: Response time $T(1)$

4.1.1. Test Case 1 - 1 Core

Fig. 1 depicts the results for the response time $T(1)$, i.e. VM instance with 1 CPU core. Since the number of cores is 1, $T(1)$ is equal to $C(1)$. The response time (cost) proportionally increases with the load, and it depends more on the message size M than the number of concurrent messages N. This can be explained by the fact that $Sort$ is computation intensive and spends more CPU time when processing large messages. For a simplified presentation we denote the points in the format (M, N), where M and N refer to both the parameters we previously defined. Thus, the minimum value of $0.003s$ is in the point $(0, 100)$, and the maximum value of $12.56s$ is in the point $(6, 1500)$. Considering the fact there is a little variability in the response time for the values of the parameter N, we find the minimum and maximum points to be within the limits of the expected. Unexpected peaks are detected in $(5, 750)$ and $(6, 750)$. The average value is $T(1) = 3.75s$.

4.1.2. Test Case 2 - 2 Cores

The results for $T(2)$ are presented in Figure 2. The maximum value detected in this case is $8.7s$, again at the point $(6, 1500)$, and the minimum value is $0.002s$ at $(0, 12)$. Considering the average value of $2.2s$, we assume that the $T(2)$ has decreased **1.7 times** in comparison to $T(1)$. Otherwise, it also proportionally increases with the load, more depending on M, than on the parameter N.

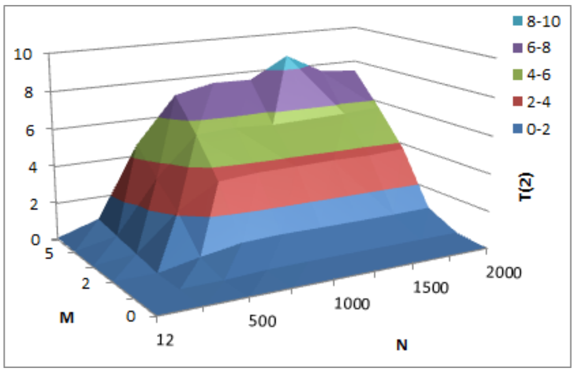

Figure 2: Response time $T(2)$

4.1.3. Test Case 3 - 4 Cores

Figure 3 presents the response time $T(4)$ results. The minimum value of $3.02s$ at $(0, 12)$

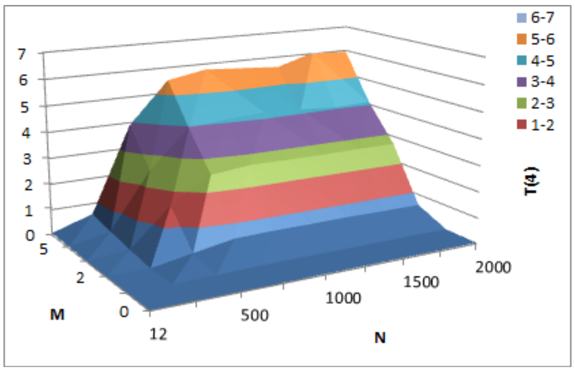

Figure 3: Response time $T(4)$

disproves the scalability property of response time decrease as the number of cores used increases. However, this phenomenon occurs only for $M = 0KB$. The maximum value of $6.04s$ is at the point $(6, 1750)$, and the average value is $1,64s$ which means that $T(4)$ decreased **2.3 times** compared to $T(1)$, and **1.35 times** compared to $T(2)$.

4.2. Cost Analysis

Using the values for response time in (1) we calculate the customer's cost for rented resources and realize the most sufficient trade-off between the cost and the gained performance.

Considering the average response time decrease in Section 4.1, we concluded that scaling up the resources n times does not provide equal performance scale. Hereupon, we aim to investigate if the cost for resources is nearly equal to the performance gain, precisely, if the customers pay as much as the gained performance.

To compare the scaling with factor 2, we analyze the results for relative cost R_{21} depicted in Figure 4.

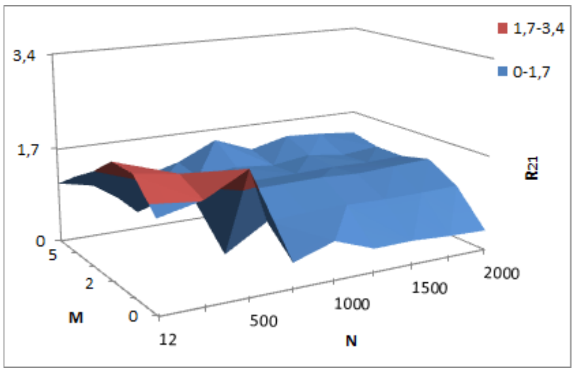

Figure 4: Relative cost R_{21}

As a cost threshold we used the average performance decrease of 1.7, i.e. if the cost transcends this value, a customer is considered to pay more than he gets. We observe that the cost is within the limits except for the load with small values of M and N. We explain this with the fact that web server needs more time to schedule the small number of small tasks instead of executing them.

The relative cost R_{41} is depicted in Figure 5.

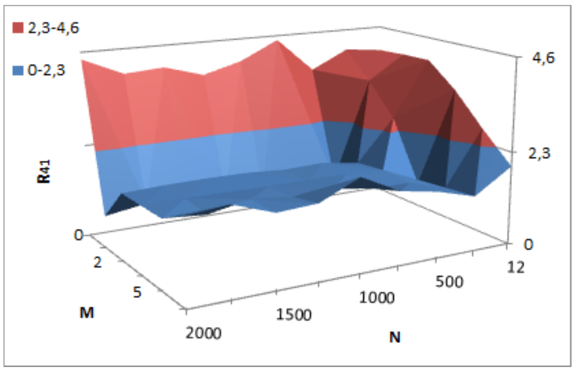

Figure 5: Relative cost R_{41}

The results show that the customer pays more than he gets when one or both parameters M or N are small. For greater load the customer will obtain the performance of the resources he pays.

The results for the relative cost R_{42} is depicted in Figure 6.

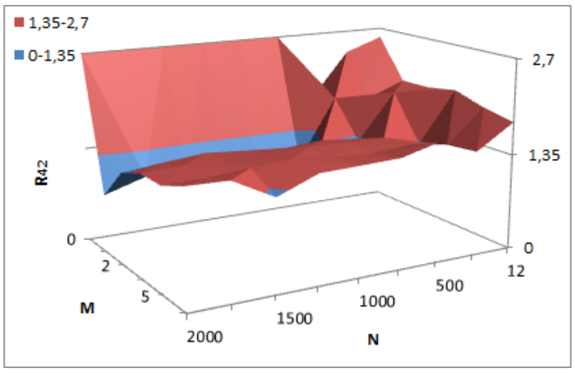

Figure 6: Relative cost R_{42}

The threshold value for scaling the resources from scaling factor 2 to 4 is 1.35. But the relative cost R_{42} is much greater than the threshold for almost all values of parameters M and N. Although there is a small region where the relative cost is smaller the threshold, its value is near the threshold.

5. Discussion

In this section we compare the results from this paper with the results from the research [16] where we performed cost and performance analysis for the memory demanding $Concat$ web service.

5.1. Response Time Comparison

When discussing the performance disproportion to the scaled resources, the $Concat$ web service performs better for both the cloud environments with VM instance with 2 and 4 CPUs defined in 3.2. Therefore, when the $Concat$ web service is hosted in cloud VM instance with 2 cores, the response time decreases 3.6 times in comparison to when hosted in cloud VM instance with 1 core. When the same web service is hosted in cloud environment with VM instance with 4 cores, according to the average response time values, the performance gets 4.6 times better than when hosted in cloud VM with 1 core. On the other hand, for the $Sort$ web service the response time decreases 1.7 times when the CPU scales to 2 cores, and 2.3 times if the number of CPUs is 4. Obviously the performance improvement depends on the type of the web service used. For a memory demanding web service, customers will gain higher performance when renting more resources, whereas for computation intensive web service, the increasing performance factor is much smaller but it also has the affinity to rise.

5.2. Cost Analysis Comparison

Once we discussed the performance improvements, in this section we proceed to discuss the trade-off between the cost and the performance. We observe that the Concat web service cost, while hosted on cloud VM instance with 2 CPUs, never transcends the 3.6 threshold. The cost analysis of the $Concat$ web service hosted in cloud VM instance with 4 cores, using the average response time decrease value of 4.6, also show that pay-per-usage model is convenient for the customers. Even though, the computation intensive $Sort$ web service shows less performance gain when scaling the resources, there are only few regions when the cost transcends the performance.

The overall comparison results show that the compromise between the cost and the performance is positive, which means the customers will gain as much performance as they pay with negligible exceptions.

6. Conclusion and Future Work

CSPs pay-per-usage model offers linearly scalable charging to the acquired resources. We assume that the performance also scales to the pay-per-usage model and in this paper we performed series of experiments to investigate in what case the customer would make the best trade-off between the performance and the expenses.

Performing analyses of both web services, we confirmed that the cost-performance relation still depends on the web services' characteristics. Overall results show that for a memory demanding web service the customer will gain maximum performance for the particular investment, but when adding more characteristics to the web service this performance gain decreases. However, generally the performance achievement stays positive.

This intrigues us to extend our research in the future and to include more web services that depend on different characteristics. Furthermore, we will perform the same analysis in a multi-tenant cloud, since we expect the performance to be interfered as a result of a shared infrastructure among many users.

References

[1] Mell P, Grance T. "The NIST Definition of Cloud Computing," *Nat. Inst. of Stand. and Tech., Inform. Tech. Lab.*, 2011.

[2] Grossman R L. "The case for cloud computing," *IT prof.*, 2009, 11, 2: 23–27, IEEE.

[3] Durkee D. "Why cloud computing will never be free," *Queue*, 8, 4, 20, 2010, ACM.

[4] Microsoft. "Windows Azure," *http://www.windowsazure.com/pricing/*, May, 2013.

[5] Google. "Compute Engine," *http://cloud.google.com/pricing/*, May, 2013.

[6] Amazon. "EC2," *http://aws.amazon.com/ec2/*, May, 2013.

[7] Berl A, Gelenbe E, Di Girolamo M, Giuliani G, De Meer H, Dang M Q, Pentikousis K. "Energy-efficient cloud computing," *The Computer Journal*, 2010, 53, 7: 1045–1051, Br Computer Soc.

[8] Baliga J, Ayre R W A, Hinton K, Tucker R S."Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. of the IEEE*, 99, 1, 149–167, 2011.

[9] Beloglazov A, Buyya R."Energy Efficient Allocation of Virtual Machines in Cloud Data Centers," *CCGrid, 2010 10th IEEE/ACM Inter. Conf. on.*

[10] Lee Y C, Zomaya A Y. "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, 60, 2, 268–280, 2012, Springer.

[11] Zhang Q, Cheng L, Boutaba, R. "Cloud computing: state-of-the-art and research challenges," *J. of Internet Services and App.*, 2010, 1, 1: 7–18, Springer.

[12] Dillon T, Wu C, Chang E. "Cloud computing: Issues and challenges," *Advanced Inform. Network. and App. (AINA), 24th IEEE Intern. Conf. on*, 27–33, 2010.

[13] De Assunção M D, Di Costanzo A, Buyya R."Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters," *Proc. of the 18th ACM inter. symp. on High perf. distr. comp.*, 141–150, 2009, ACM.

[14] Andrzejak A, Kondo D, Yi S."Decision model for cloud computing under SLA constraints," *in Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, 257–266, 2010.

[15] Truong H, Dustdar S. "Composable cost estimation and monitoring for computational applications in cloud computing environments," *Procedia Computer Science*, vol. 1, no. 1, pp. 2175–2184, 2010.

[16] Simjanoska M, Ristov S, Velkoski G, Gusev M. "Scaling the Performance and Cost While Scaling the Load and Resources in the Cloud," *MIPRO, 2013 Proceedings of the 36th International Convention*, 2013.

[17] Gusev M, Ristov S. "The optimal resource allocation among virtual machines in cloud computing," *in CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs and Virtualization*, 36–42, 2012.

[18] Ristov S, Velkoski G, Gusev M, Kjiroski K. "Compute and memory intensive web service performance in the cloud," *in ICT Innovations 2012*, S. Markovski and M. Gusev, Eds. Springer Berlin / Berlin Heidelberg, vol. AISC 257, pp. 215–224, 2012.

[19] OpenStack."Openstack cloud," *www. openstack.org*, 2013.

[20] Juric M B, Rozman I, Brumen B, Colnaric M, Hericko M."Comparison of performance of Web services, WS-Security, RMI, and RMI–SSL," *Journal of Systems and Soft.*, 79, 5:689–700, 2006, Elsevier.

[21] SoapUI."Functional testing tool for web service testing," *www.soapui.org*, 2013.