Scaling the Performance and Cost While Scaling the Load and Resources in the Cloud

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

***Abstract*—Cloud computing is a paradigm that offers on-demand scalable resources with the "pay-per-usage" model. Price rises linearly as the resources scale. However, the main challenge for cloud customers is whether the performance is also scaling as the price for the resources. In this paper we analyze both the performance and the cost of a memory demanding web service. The experiments are based on measuring the performance and calculating the costs of rented CPU resources for different server loads, obtained by changing the message size and the number of concurrent messages. The results show that the lowest cost is obtained while the memory demanding service is hosted on two CPUs.**

***Index Terms*—Cloud Computing, Web Services, Performance, Resources, Cost**

I. Introduction

According to the NIST definition [1], cloud computing is a model for enabling 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. From a hardware point of view, there are three aspects in cloud computing: 1) the illusion of infinite computing resources available on demand, 2) the elimination of an up-front commitment by cloud user; and 3) the ability to pay for use of computing resources on a short-term basis as needed [2]. 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 [3]. However, service providers are not able to accomplish all customers' demands for services since there is a high level of alterations which have to be negotiated via Service Level Agreements (SLAs) [4].

The service provider's "pay-per-usage" model offers a pricing scheme scalable to the amount of rented resources. A natural expectation from this kind of pricing model is that the performance gain also scales to the monetary costs, i.e. the more resources are rented, the more performance is achieved. Even though, cloud service providers guarantee the availability of the rented resources to the customers by defining SLAs, a guarantee of scalable and sustainable performance misses in the SLAs [5].

For the purpose of customers' contentment, it is very important to perform a research to answer the question if the performance proportionally scales to the price paid for renting resources. This question is not so simple as it looks like. The performance depends mostly on one or several resource parameters. Increasing the parameter which is not a performance bottleneck can even reduce the performance, while increasing a parameter that causes performance bottleneck can provide a significant speedup. For example, increasing the CPU speed will improve the matrix multiplication algorithm only several percents (the computations), but increasing the number of processors will achieve maximum linear speedup according to Gustafson's Law [6]. Using the processors with dedicated cache per core can achieve even superlinear speedup (the speedup greater than the number of processors). On the other hand, scaling is limited with the serial part of the algorithm, i.e. the Amdahl's Law [7].

The cloud service provider's renting model is usually linear with constant price / performance ratio for main computing resources: CPU, main memory and storage disk. Table I presents the details about current offers in Jan 2013 for renting the following virtual machine (VM) instances: Azure [8], Google [9] and Amazon [10]. The relative scaling factor is also presented for different resources.

TABLE I
VM instance types and price simulation

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

In this paper, we consider single user case whose monetary costs for resources are proportional to the amount of rented resources. Therefore, our performance analysis are based on the quantity of acquired resources. Hereupon, we defined three different cloud environments with virtual machine instances, each with different number of CPU cores, and thus, simulated different amount of rented resources. Furthermore, in order to obtain comparable results, we loaded each cloud environment with the same load, i.e., the same number of concurrent messages with the same message size. In order to simulate realistic demands for service, in each cloud environment we hosted a memory demanding ($Concat$) web service. This is a simple web service that returns a concatenation of two input

strings.

The rest of the paper is following the next organization. A brief review of the related work for scaling the performance and cost in the cloud is presented in Section II. In Section III we present the developed methodology used for testing and obtaining reliable results. The experiments and the outcomes are exhibited in Section IV. In the final Section V, we derive conclusions over the results and we present our ideas for future research.

II. Related Work

In this section we present a brief review of the recent research closely related to our field of interest.

The recent research related to a "cost-efficient" cloud computing mostly examine the cloud service providers' costs for offering cloud computing solution. For example, several papers have analyzed the cloud computing performance to find out if the cloud is energy-efficient and therefore, cost-efficient [11], [12], [13], [14]. Some of the most important cloud computing issues and challenges are discussed by Zhang [15] stating that with on-demand resource provisioning and utility based pricing, service providers can truly maximize resource utilization and minimize their operating costs.

Several papers refer to different behavior of performance when scaling. Lu et al. [16] discovered several pitfalls resulting in waste of active VMs idling. They examine several pitfalls in Windows Azure Cloud during several days of performing the experiments: Instance physical failure, Storage exception, System update. Windows Azure does not work well for tightly-coupled applications [17].

However, in this paper, we consider the customers' benefits of the on-demand resource provisioning as maximized performance with minimal costs. Similar research of this kind is presented by De Assuncao et al. [18], where the authors present several scheduling strategies for balancing between performance and usage cost, and how much they improve the requests' response times. Their results show that some of the strategies result in a higher cost under heavy load conditions, whereas some showed a good ratio of slowdown improvement to the money spent for using cloud resources. Andrzejak et al. 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 [19]. Using their model, the users can achieve largest cost savings (for considered workload types), by using the high-CPU instance types instead of standard or high-memory instance types. The authors' contribution in [20] is developing a service that is able to perform the cost determination for scientific applications in cloud computing environments. Kondo et al. compare and contrast the performance and monetary cost-benefits of clouds for computing applications, ranging in size and storage [21].

Considering the performance, using cache intensive algorithms in both single-tenant and multi-tenant cloud environments, Gusev and Ristov show how and when cloud computing can achieve even better performance than traditional environment for certain workload [22]. Another research for cloud's performance states that the cloud achieves smaller performance degradation for greater message sizes using the memory demanding web service, and also for greater message sizes and smaller number of concurrent messages for memory demanding and compute intensive web services [23].

As we covered the state of the art related to the problem of scaling performance and cost in the cloud, we proceed with research methodology to find out if the performance rises linearly to the cost.

III. Research Methodology

In order to perform realistic experiments, we developed original methodology that includes technical details, appropriate cloud environment configuring to include all three test cases with different number of resources, and the testing procedure itself. Considering the outcomes from the experiments we present relevant mathematical relations to obtain reliable results.

A. Technical Framework

As a testing environment we used client-server architecture deployed in the open source cloud platform OpenStack [24] using Kernel-based Virtual Machine (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 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 [25].

B. 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 3*: VM instance with 4 CPUs.

Each cloud environment hosts the web service $Concat$, that accepts two strings and returns their concatenation. It is only memory demanding web service, since it does not require any processing power.

C. Testing Procedure

The client uses SoapUI [26] to generate various server loads. Each instance is loaded with N messages with parameters parameter size of M kilobytes each, with 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 and avoiding saturation. The web service is loaded with $N = 12; 100; 500; 752; 1000; 1252; 1500; 1752$

and 2000 requests per second for each message parameter size $M = 0; 1; 2; 4; 5; 6; 7; 8$ and 9. In order to simulate different connections per core we divide the N concurrent messages in four groups of $N/4$ messages each.

D. Test Data

In the research experiments we measure the server's average response time for different parameter sizes M, and number of concurrent messages N. The experiments are realized for each test case, as defined by the environments in III-B. These measurements will express the cloud's performance. The equation (1) presents the scaled average response time T, where A denotes the average response time, n is the number of used cores, and 4 denotes the different connections per core.

$$T(n) = \frac{A(n)}{4} \tag{1}$$

Pursuant to our goal of expressing the performance through cost evaluation, in equation (2) we calculate the cost C, where $T(n)$ denotes the response time derived from equation (1) and n is the total number of processors used.

$$C(n) = T(n) * n \tag{2}$$

Our research problem checks whether the performance is proportional to the number of rented resources, therefore, the web services' total cost C is the real cost of rented CPU resources. For this purpose we will calculate relative performance and relative cost of the scaling problem, as expressed in (3), (4) and (5), 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.

$$R_{21} = \frac{C(2)}{C(1)} \tag{3}$$

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

$$R_{42} = \frac{C(4)}{C(2)} \tag{5}$$

Ideal expectation will be the proportional scaling, i.e. when $R_{21} = 2$, $R_{41} = 4$ and $R_{42} = 2$. Any deviation from these expectations will make new conclusions in this research.

IV. Experiments and Results

In this section we present the experiments and the results using the methodology specified in III.

A. Analysis of Response Time

The memory demanding *Concat* web service has been hosted in VM instances with 1, 2 and 4 CPU cores.

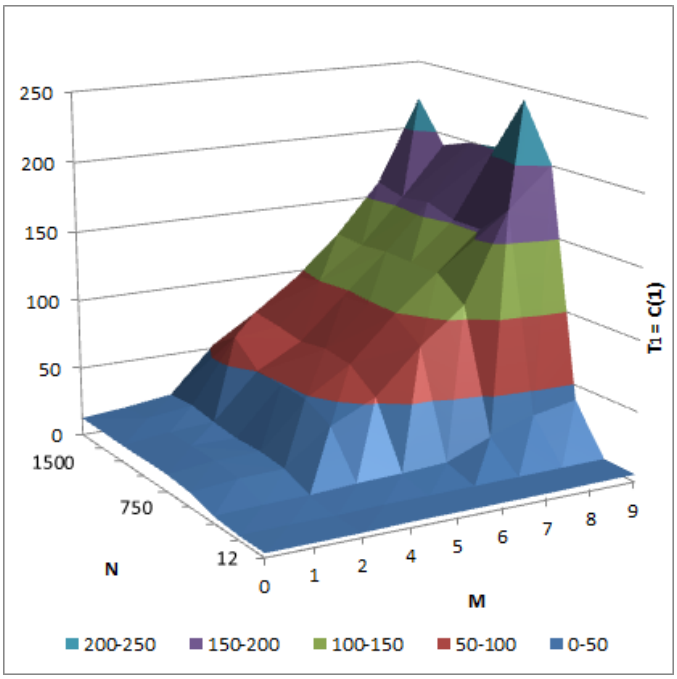

Fig. 1. Cost = response time for *Concat* web service $(1x1)$

1) Test Case 1 - VM with 1 CPU core: Figure 1 presents the results from the cloud environment with 1 VM instance with 1 CPU core. According to the results, the response time depends equally on the message parameter size, M, and number of concurrent messages, N. Figure 1 denotes that $C(1)$ is equal to $T(1)$ because the number of used CPU is 1, and if applied in equation in (2), the cost and the response time remain the same.

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 3.095 is in the point $(0, 12)$, and the maximum value of 241.35 is in the point $(9, 1000)$. Considering that the response time proportionally increases as both of the parameters M and N increase, we find the minimum value is in the expected point, but the maximum value seems to be an unexpected pick. The average value is 58.24.

2) Test Case 2 - VM with 2 CPU cores: The results from the cloud environment with 1 VM instance with 2 CPU cores are presented in Figure 2. The minimum value of 2.38 is at the point $(0, 100)$, whereas the maximum value of 59.08 is again at the point $(9, 1000)$. According to the average value of 16.24, we assume that the response time has decreased 3.6 times in comparison to the cloud with 1 VM instance with 1 CPU. In the next section we use that the performance gain of a VM with 2 cores over a VM with one core has value of 3.6.

3) Test Case 3 - VM with 4 CPU cores: Figure 3 depicts the results from the cloud environment with 1 VM instance with 4 CPU cores. The minimum value of 3.02 at $(0, 12)$ shows that for small number of concurrent messages and small message parameter size, there is minority variation in the response time

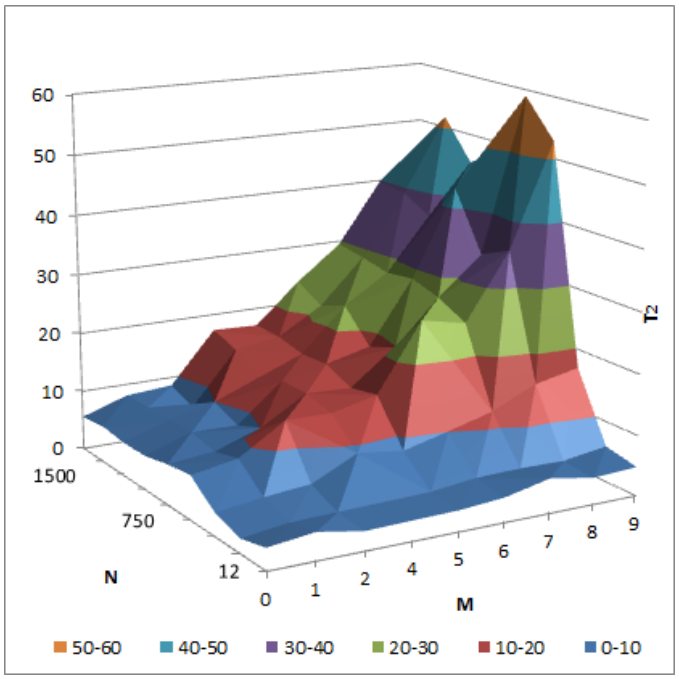

Fig. 2. Response time for $Concat$ web service $(1x2)$

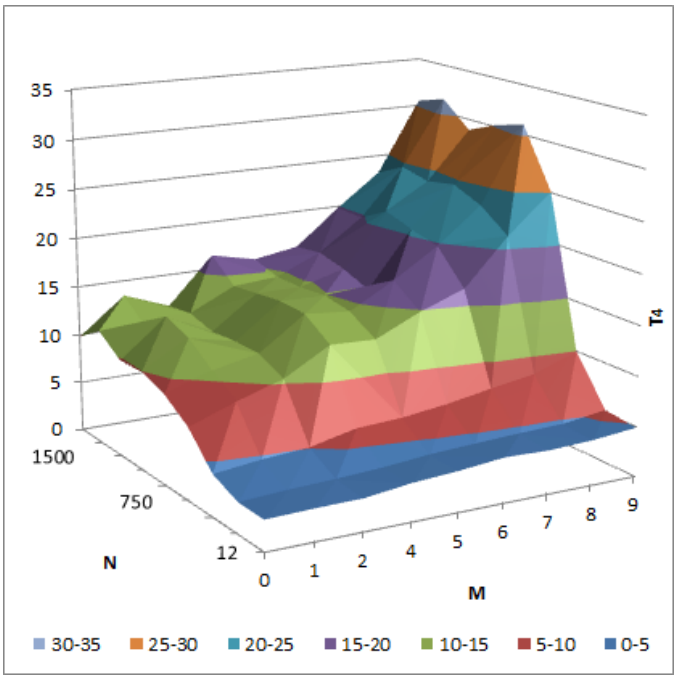

Fig. 3. Response time for $Concat$ web service $(1x4)$

despite of the increased number of resources. The maximum value of 31.39 is at the point $(9, 1750)$, and the average value is 12.78 which means that the response time decreased 4.6 times compared to the cloud VM with 1 core, i.e. the performance gain of a VM with 4 CPU cores compared to a VM with 1 CPU core is 4.6. We have also calculated the performance gain of a VM with 4 CPU cores over a VM with 2 CPU cores to be 1.3.

B. Cost Analysis

In order to derive conclusions about the sufficient trade-off between the cost and the gained performance, we used the equation in (2) to calculate the customer's cost for resources using the average response time calculated when using the equation in (1).

Observing the average response time decrease in IV-A, we have concluded that scaling up the resources n times, where n is the number of cores, provides even more that n times performance gain. Hereupon, we aim to find out answer for our research problem, i.e. whether the cost for resources is equal to the performance gain, moreover, whether there is an occasion where customers pay less than they gain.

The default cost per core is the one core cloud environment, i.e. the results from the cloud environment with 1 VM instance with 1 CPU as presented in Figure 1.

The relative ratio between the costs for the cloud VM with 2 CPUs and cloud VM with 1 CPU, calculated by (3) is depicted in Figure 4. As a cost threshold we used the average performance decrease of 3.6, thus, if the cost transcend this value, a customer is considered to pay more than he gets. Interestingly, we observe that the customer's costs for

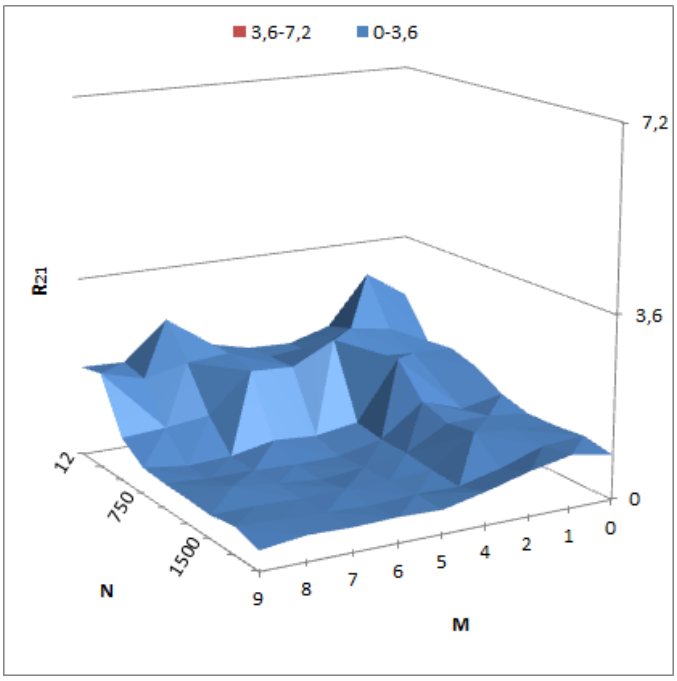

Fig. 4. Cost analysis for $Concat$ web service R_{21}

resources remain far beyond the threshold value.

Figure 5 depicts the proportion of the cost for the cloud VM with 4 CPUs and the cost for a cloud VM with 1 core, calculated by (4). The results also show that for any message parameter size and number of concurrent messages, the customer's costs are much lower than the performance advantage.

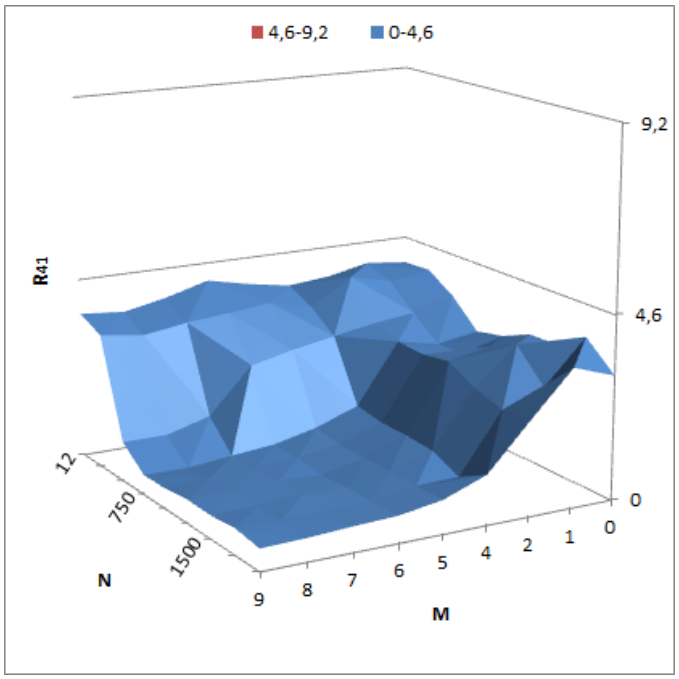

Fig. 5. Cost analysis for $Concat$ web service R_{41}

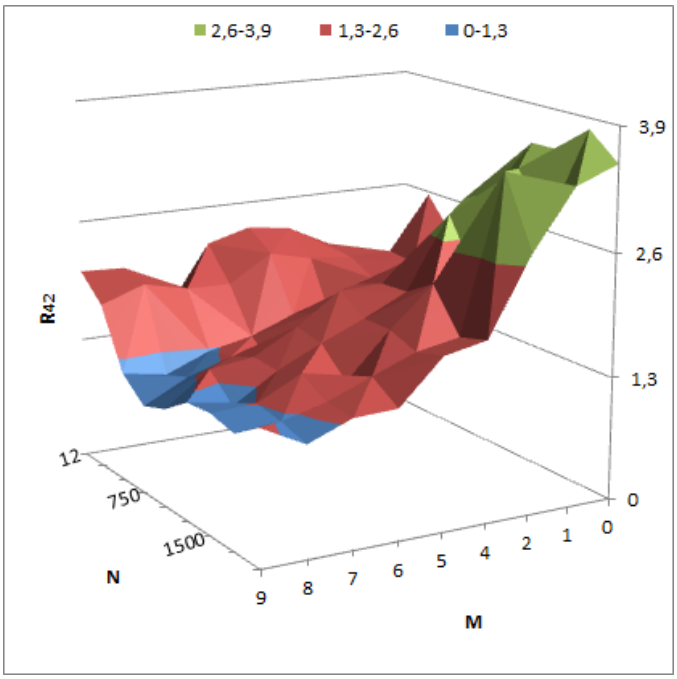

Fig. 6. Cost analysis for $Concat$ web service R_{42}

Interestingly, incrementing the number of concurrent messages, N, and the message size, M, the cost decreases at both scenarios.

Eventually we compared the costs between the cloud VM instance with 4 CPUs and cloud VM instance with 2 CPUs, calculated by (5). The reason for this analysis is realizing if the customer will make a good decision demanding 2 more CPUs once he used 2 cores. When presenting the results on Figure 6 we used 1.3 value as a threshold which we obtained from the results in IV-A where we concluded that the performance gain is 1.3 times for VM with 4 cores in comparison to VM with 2 cores. However, the results show that there are small regions where the customer pays less than he gets, thus, it is not worthwhile renting 2 more cores once the customer has 2 cores.

V. Conclusion and Future Work

Cloud computing pricing pay-per-usage model makes the paying be linearly scalable to the amount of rented resources. The most important question from a customers point of view is whether the performance also scales to quantity of resources acquired and thus, whether it proportionally varies with the customer's monetary costs.

In this paper we performed a series of experiments to answer this question and to derive conclusions for which cloud configuration, if any, the customer gets more performance than the amount paid.

Observing all scenarios, we conclude that the $Concat$ web service while hosted on 2 CPUs provides 3.6 time better performance than when hosted on 1 CPU, which is the best occasion from all test cases. Moreover, it also provides the lowest cost in comparison to the gained performance.

On the other hand, the worst results are obtained when the customer migrates from 2 CPUs to 4 CPUs, i.e. the cost is much above the performance benefit.

However, the results showed satisfying trade-off between the cost and the performance for the memory demanding web service. In our future work we aim to extend the research including a web service which is not only memory demanding, but also depends on other characteristics like computation or I/O intensive. Furthermore, we aim to perform the same analysis in a multi-tenant cloud since we expect the performance to be interfered because large number of users share the same infrastructure.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Information Technology Laboratory, Sep. 2011.

[2] A. Fox, R. Griffith *et al.*, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Tech. Rep. UCB/EECS*, vol. 28, 2009.

[3] R. Grossman, "The case for cloud computing," *IT professional*, vol. 11, no. 2, pp. 23–27, 2009.

[4] P. Patel, A. Ranabahu, and A. Sheth, "Service level agreement in cloud computing," in *Cloud Workshops at OOPSLA*, 2009.

[5] D. Durkee, "Why cloud computing will never be free," *Queue*, vol. 8, no. 4, p. 20, 2010.

[6] J. L. Gustafson, "Reevaluating Amdahl's law," *Communication of ACM*, vol. 31, no. 5, pp. 532–533, May 1988.

[7] G. M. Amdahl, "Validity of the single-processor approach to achieving large scale computing capabilities," in *AFIPS Conference Proceedings*, vol. 30. AFIPS Press, Reston. Va., Atlantic City, N.J., Apr. 18-20 1967, pp. 483–485.

[8] Microsoft, "Windows azure," July 2012. [Online]. Available: http://www.windowsazure.com/pricing/

[9] Google, "Compute engine," July 2012. [Online]. Available: http://cloud.google.com/pricing/
[10] Amazon, "Ec2," 2012. [Online]. Available: http://aws.amazon.com/ec2/
[11] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *The Computer Journal*, vol. 53, no. 7, pp. 1045–1051, 2010.
[12] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.
[13] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10, 2010, pp. 826–831.
[14] Y. Lee and A. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268–280, 2012.
[15] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.
[16] W. Lu, J. Jackson, J. Ekanayake, R. S. Barga, and N. Araujo, "Performing large science experiments on azure: Pitfalls and solutions," in *CloudCom'10*, 2010, pp. 209–217.
[17] V. Subramanian, H. Ma, L. Wang, E.-J. Lee, and P. Chen, "Rapid 3d seismic source inversion using windows azure and amazon ec2," in *Proc. of IEEE*, ser. SERVICES '11. IEEE Comp. Soc., 2011, pp. 602–606.
[18] M. De Assunção, A. Di Costanzo, and R. Buyya, "Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters," in *Proceedings of the 18th ACM international symposium on High performance distributed computing*. ACM, 2009, pp. 141–150.
[19] A. Andrzejak, D. Kondo, and S. Yi, "Decision model for cloud computing under sla constraints," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 257–266.
[20] H. Truong and S. Dustdar, "Composable cost estimation and monitoring for computational applications in cloud computing environments," *Procedia Computer Science*, vol. 1, no. 1, pp. 2175–2184, 2010.
[21] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. Anderson, "Cost-benefit analysis of cloud computing versus desktop grids," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1–12.
[22] M. Gusev and S. Ristov, "The optimal resource allocation among virtual machines in cloud computing," in *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization*, 2012, pp. 36–42.
[23] S. Ristov, G. Velkoski, M. Gusev, and K. Kjiroski, "Compute and memory intensive web service performance in the cloud," in *to be published in ICT Innovations 2012*. Springer Berlin / Heidelberg, 2012.
[24] OpenStack, "Openstack cloud software," Jan. 2013. [Online]. Available: http://openstack.org
[25] M. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko, "Comparison of performance of web services, ws-security, rmi, and rmi–ssl," *Journal of Systems and Software*, vol. 79, no. 5, pp. 689–700, 2006.
[26] SoapUI, "Functional testing tool for web service testing," Jan. 2013. [Online]. Available: http://www.soapui.org/