CPU Utilization in a Multitenant Cloud

Goran Velkoski, Monika Simjanoska, Sasko Ristov, and Marjan Gusev

Ss. Cyril and Methodius University, Faculty of Information Sciences and Computer Engineering,
Rugjer Boshkovik 16, PO Box 393, 1000 Skopje, Macedonia
velkoski.goran@gmail.com
m.simjanoska@gmail.com
sashko.ristov@finki.ukim.mk
marjan.gushev@finki.ukim.mk

***Abstract*—Cloud model's ability of sharing hardware resources and services among multiple tenants sets new challenges and issues. Up to now, most of the surveys mostly relate to the security issues in a multi-tenant cloud. In this paper, we measure the performance for the multi-tenant cloud model and aim to analyze the CPU utilization for applying the same load to different web services hosted in multi-tenant environments; and finally compare it to the single-tenant environment. Two parameters will be analyzed: 1) the number of concurrent messages and 2) the message size. Test cases consist of two web services, the first is the memory demand only web service ($Concat$) and the second is both computation intensive and memory demanding web service ($Sort$). The conclusions will analyze the results of the performance in terms of CPU utilization in all cloud environments using the same amount of resources, but orchestrated differently.**

We have set a hypothesis that single-tenant cloud environment achieves smaller CPU utilization than both the multi-tenant environments we will test. However, the results show that there are particular regions where multi-tenant environment achieves smaller CPU utilization than a single-tenant environment. Moreover, we have performed additional tests to make better conclusions which multi-tenant environment achieves smaller utilization in comparison one to another, and showed that at most cases, for both $Concat$ and $Sort$ web services, multi-tenant $(4x1)$ environment achieves greater utilization than multi-tenant $(2x2)$ environment, except for $Sort$ web service where multi-tenant $(2x2)$ achieves smaller utilization than multi-tenant $(4x1)$ environment when exactly one of the input parameters is small.

***Index Terms*—Cloud Computing, CPU utilization, Multi-tenancy, Performance, Resources, Web Server**

I. Introduction

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data-centers that provide those services [1]. According to the NIST definition specified in [2], cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Moreover, they refer to the on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service, as the five characteristics that compose the cloud model. The resource sharing feature describes the cloud model's ability of sharing hardware resources and services among multiple tenants. This multi-tenancy ability is one of many that make this new computing paradigm different from the traditional service computing. In contrast to a single-tenant environment, the authors in [3] give an overview to some of the key features of multi-tenancy as hardware resource sharing, high degree of reconfigurability, shared application and database instances, and list a number of benefits for companies to achieve higher utilization of hardware resources, easier and cheaper maintenance, lower overall costs, etc.

However, there are also some issues and challenges that multi-tenant environment instigates. The authors in [4] [5] [6] discuss the multi-tenancy issues in terms of security and propose novel cloud architectures, authorization system, etc., for solving the security issue in a multi-tenant cloud. Achieving a full potential of multi-tenancy requires solving of three issues: resource sharing, security isolation and customization [7].

Unlike previously mentioned multi-tenancy challenges, we observe the hardware resources utilization and performance in a multi-tenant cloud environment. One research of this kind is given by Wang et al. [8] which aims to estimate the CPU consumption in multi-tenant applications. We have previously realized experiments on web services hosted on-premise and in a cloud for different loads varying the main input factors: the message size and the number of messages [9]; and derived conclusion over the performance behavior.

In this paper we analyze the relevance of the hypothesis if the single-tenant cloud environment will provide smaller CPU utilization than multi-tenant environments. The expectation is that in a single-tenant environment we do not share the resources among several virtual machine instances and would achieve smaller CPU utilization. However, the results show that this is not always the case, i.e. special regions were identified where the opposite results are obtained.

Usually the cost for resources is proportional to the offered resources in the cloud. A natural expectation from a customer point of view is that the performance of the web services hosted in the cloud is also proportional to the availability of the rented resources. In this paper, we performed a series of experiments by changing the message size and the number of concurrent messages simulating different number of tenants to analyze the CPU utilization and thus derive conclusions about the performance. Moreover, we intent to confirm if the CPU utilization for a particular message size and number of

978-1-4673-2232-4/13/$31.00

concurrent messages provides an extra CPU utilization, which we refer to as relative increase in CPU utilization. Additionally we experiment which multi-tenant environment will provide smaller CPU utilization, and refer to as relative multi-tenant increase in CPU utilization.

The idea to analyze only the CPU Utilization in this paper is based on efficient usage of resources, especially required as essential for massive parallel servers and reducing the overall energy consumption, thus leading to the green computing paradigm.

The rest of the paper is organized as follows. In Section II we present technical details of the cloud environment, and description of the web services used for testing and implementation procedures. The experiments and the results are presented in Section III. In Section IV we analyze and discuss the results we obtained for the relative multi-tenant increase of CPU utilization. The comparison of the results is presented in Section V. The conclusion and the future work are specified in final Section VI.

II. The Methodology

This section contains a description of the cloud environment used for the testing. We present details of the web services used for testing and the implementation procedure used for obtaining reliable results.

A. Hardware Technical Details

As a testing environment we used a client-server architecture deployed in OpenStack cloud [10]. It is an open source cloud platform. Kernel-based Virtual Machine (KVM) hypervisor is used to instantiate virtual machine instances. The client and server node are installed with Linux Ubuntu Server 12.04 operating system using. Hardware computing resources consist of Intel(R) Xeon(R) CPU X5647 @ 2.93GHz with 4 cores and 8GB RAM. The server platform in cloud, i.e. virtual machine instances consist of Linux Ubuntu Server 12.04 operating system and Apache Tomcat 6 as the application server. The client has used SoapUI [11] to test web service performance for different server loads. The client and the cloud nodes are placed in the same LAN segment to minimize network latency [12].

B. Web service description

In order to test both computationally intensive and memory demanding web services, we used two document style Java web services.

The goal of this research is to measure the CPU utilization in cloud, and not analyze behavior of other factor that impact the real speed in throughput in real web service environment. The idea is to give recommendations which environment performs better. Therefore the selection criteria for testbed web services included simple algorithms, the first which is only memory demanding, i.e. occupies the memory very fast, without any processing requirement, while the other on top of the memory demand includes algorithm which requires a lot of computations.

As a simple memory demanding web service, we used the $Concat$ web service, which accepts two strings and returns their concatenation. It is a memory demanding web service and depends only on the input parameter size M with memory complexity $O(M)$.

The $Sort$ web service is a web service which accepts two strings and returns their concatenation, but sorted alphabetically. It is both compute intensive and memory demanding web service. The memory complexity of Sort web service is $O(M)$ and its time complexity is $O(M \cdot \log_2 M)$. Sorting is realized with the merge sort algorithm.

C. Testing Environment

We defined one single-tenant and two multi-tenant environments to host both web services defined in Section II-B:

- Single-tenant cloud environment with 1 virtual machine instance with all 4 CPU cores;
- Multi-tenant cloud environment with 2 virtual machine instances, each with 2 CPU cores; and
- Multi-tenant cloud environment with 4 virtual machine instances, each with 1 CPU core.

All virtual machine instances have 512 MB RAM. Each environment uses total four cores allocated in one, two or four instances.

D. Tests Implementation

Each test case runs for 60 seconds. Web servers in virtual machine instances are loaded with N messages with parameter size of M bytes each. We use variance 0.5 to reduce the warm-up impact, i.e., the number of threads varies 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 virtual machine instances work in normal mode without replying error messages. Parameter size M is measured in KB with the following values $0, 1, 2, 4, 5, 6, 7, 8$ and 9 for Concat web service and $0, 1, 2, 4, 5$ and 6 for Sort web service. Both Concat and Sort web services are loaded with $N = 12, 100, 500, 752, 1000, 1252, 1500, 1752$ and 2000 requests per second for each parameter size M.

E. Testing Procedure

As we defined the tests and the cloud testing environments, we aim to obtain reliable tests and results for each test in a different cloud testing environment. Therefore, we divide N concurrent messages in 4 groups of $N/4$ messages each, and simulate different connections per each core apart of resource allocation in 1, 2 or 4 cores. Thus, we defined to experiment three test cases described in the following three sections.

1) Test Case 1 - Single-tenant Environment with 4 CPUs per Virtual Machine Instance: The $Concat$ and $Sort$ web services defined in Section II-B are hosted in one virtual machine instance with all four CPU cores in a single-tenant cloud environment, which is loaded with 4 different clients, each with $N/4$ concurrent messages.

2) Test Case 2 - Multi-tenant Environment with 2 CPUs per Virtual Machine Instance: The $Concat$ and $Sort$ web services are hosted in two virtual machine instances with two CPU cores each in a multi-tenant cloud environment each loaded with 2 different clients with $N/4$ concurrent messages.

3) Test Case 3 - Multi-tenant Environment with 1 CPU per Virtual Machine Instance: The $Concat$ and $Sort$ web services are hosted in four virtual machine instances with one CPU core each in a multi-tenant cloud environment each loaded with 1 client with $N/4$ concurrent messages.

F. CPU utilization Measurements

The performance of $Concat$ and $Sort$ web services in terms of CPU utilization is calculated as the average of the CPU usage U, of all virtual machine instances in each of the tests defined in Section II-E. Thus, the CPU is measured for each parameter size M, for different web service loads per second N, in each of the single-tenant and multi-tenant environments we previously defined.

The CPU utilization is measured for both $Concat$ and $Sort$ web services distinctively. We use (1) to normalize CPU usage in range from 0 up to 100%. The nominator $\sum_{i=1}^{i=k} U_i(n)$ is the sum of CPU usage of all n virtual machine instances per test case, and the denominator n denotes the number of virtual machine instances used in that case.

$$U_n = \frac{\sum_{i=1}^{i=n} U_i(n)}{n} \quad (1)$$

Furthermore, we calculate the *Relative increase of CPU utilization* $R(n)$, defined in (2), where n denotes the number of active virtual machine instances, U_1 denotes the average CPU utilization for execution of a web service hosted in single-tenant virtual machine instance with all 4 processors and a particular load, and U_n denotes the average CPU utilization for the same particular load on a web service hosted on n multi-tenant virtual machine instances with P/n processors each, where P denotes the total number of used processors.

$$R(n) = \frac{U_1}{U_n} \quad (2)$$

Our hypothesis states that single-tenant cloud environment achieves smaller CPU utilization than both of the multi-tenant environments we have defined in the experiments. Thus, we intent to test the validity of the hypothesis and find if there is a particular region where the hypothesis will not be satisfied. The regions will be defined by the values of input parameters: the parameter size M, and number of requests per second N. A survey on performance evaluation is presented in [13] where the authors show how and when cloud computing can achieve even better performance than traditional environment, both in a single-tenant and multi-tenant resource allocation for certain workload.

Moreover, we will perform additional tests to derive a conclusion which multi-tenant environment provides greater utilization in mutual relative comparison. To achieve this we will follow (3) where R_m denotes the relative multi-tenant increase in CPU utilization, and $R(4)$ and $R(2)$ denote the relative increase in CPU utilization for multi-tenant environments with two and four virtual machine instances, respectively.

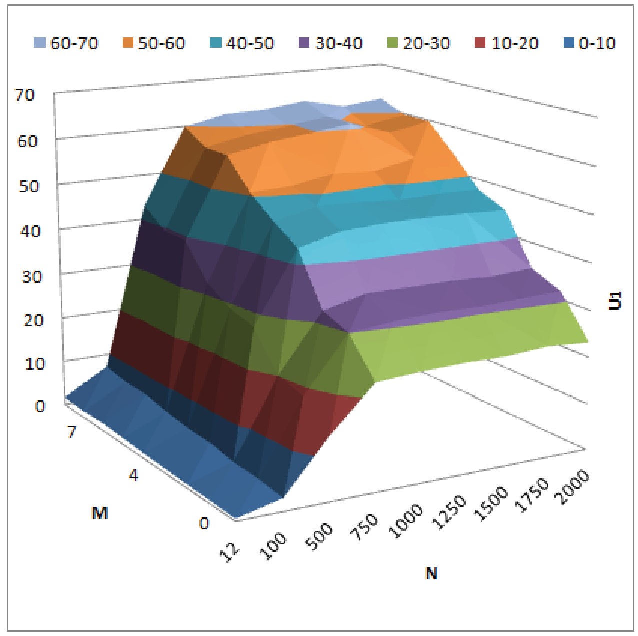

Fig. 1. Normalized CPU utilization U_1 for $Concat$ web service

$$R_m = \frac{R(2)}{R(4)} \quad (3)$$

III. Experiments and Results

In this section we present the CPU utilization results of testing the web services hosted in cloud single-tenant and multi-tenant environments. The testing is performed for different message parameter size M and number of concurrent messages N as described in sections II-D and II-B. The experiments are performed following the testing procedure specified in Section II-E. We determine the input parameters M and N where maximum and minimum CPU utilization is achieved.

A. Test 1. Single-tenant Cloud Environment (1 x 4)

This test is described with $Concat$ and $Sort$ web services hosted in a single-tenant cloud environment with 1 virtual machine instance with 4 CPU cores.

1) Test 1.1 Concat Web Service: Fig. 1 depicts the normalized CPU utilization U_1 for $Concat$ web service hosted in a single-tenant cloud environment with one virtual machine instance with four CPU cores.

The results show that the CPU utilization strongly depends on the number of concurrent messages N, and the dependence on the messages parameter size M is proportional with small increasing factor. The minimum CPU utilization is 0.54% for parameters $M = 0$ and $N = 12$. The maximum measured CPU utilization is 63.18% occurs for $M = 9$ and $N = 1500$.

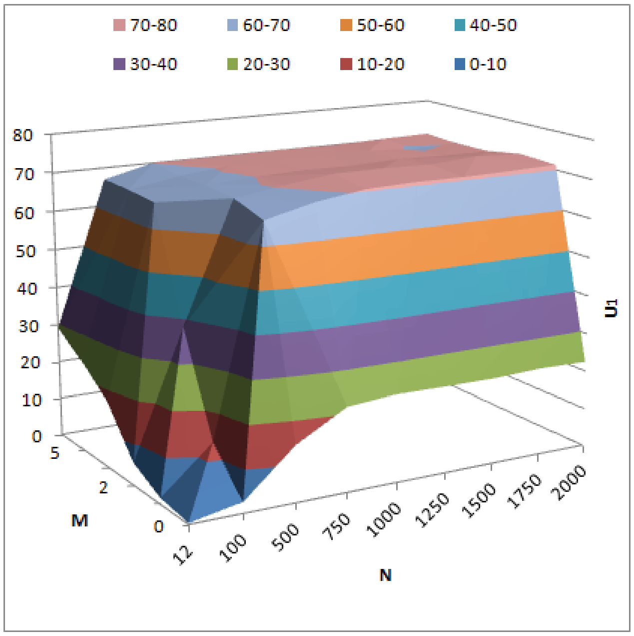

Fig. 2. Normalized CPU utilization U_1 for *Sort* web service

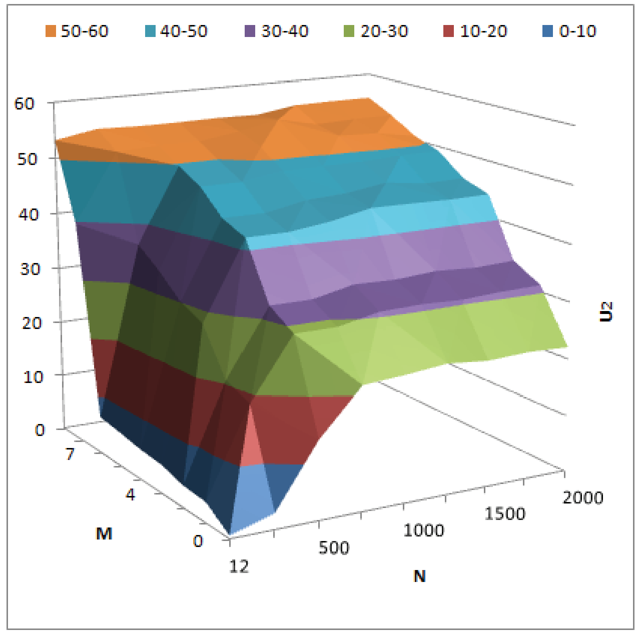

Fig. 3. Normalized CPU utilization U_2 for *Concat* web service

2) *Test 1.2 Sort Web Service:* Fig. 2 depicts the normalized CPU utilization U_1 for *Sort* web service hosted in a single-tenant cloud environment with one virtual machine instance with four CPU cores.

In comparison to *Concat* web service, the results for *Sort* web service show that the CPU utilization depends on both the messages parameter size M and the number of concurrent messages N. The dependence is expressed with huge increase factor. The minimum CPU utilization is 0,33% for input parameters $M = 0$ and $N = 12$, while the maximum CPU utilization of 72,46% is achieved for $M = 2$ and $N = 1750$.

B. *Multi-tenant Cloud Environment* ($2x2$)

The testing uses *Concat* and *Sort* web services hosted in a multi-tenant cloud environment with 2 virtual machine instances with 2 CPU cores each.

1) *Test 2.1 Concat Web Service:* Fig. 3 depicts the normalized CPU utilization U_2 for *Concat* web service hosted in a multi-tenant cloud environment with two virtual machine instances, each with two CPU cores.

The results show that the CPU utilization increases proportionally to both the messages parameter size M and the number of concurrent messages N. The minimum CPU utilization is 0,53% for input parameters $M = 0$ and $N = 12$, while the maximum CPU utilization of 55.56% occurs for $M = 9$ and $N = 2000$.

2) *Test 2.2 Sort web service:* Fig. 4 depicts the normalized CPU utilization U_2 for *Sort* web service hosted in a multi-tenant cloud environment with two virtual machine instances, each with two CPU cores.

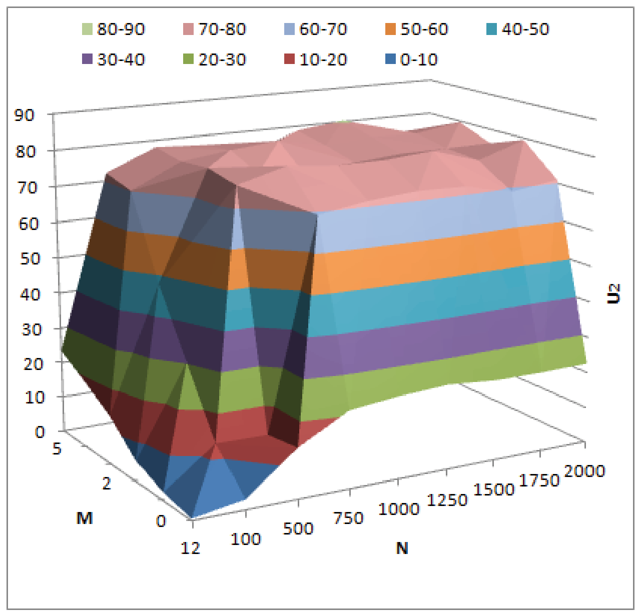

Fig. 4. Normalized CPU utilization U_2 for *Sort* web service

In this case, for a particular number of concurrent messages $N = 750; 1000; 1250$, and suitable message parameter size $M = 2; 4; 5; 6$, we measured variations in the CPU utilization, which has not been observed in comparison to the results from *Sort* web service hosted in a single-tenant cloud environment. The minimum CPU utilization is 0,52% for input parameters $M = 0$ and $N = 12$ while the maximum CPU utilization of

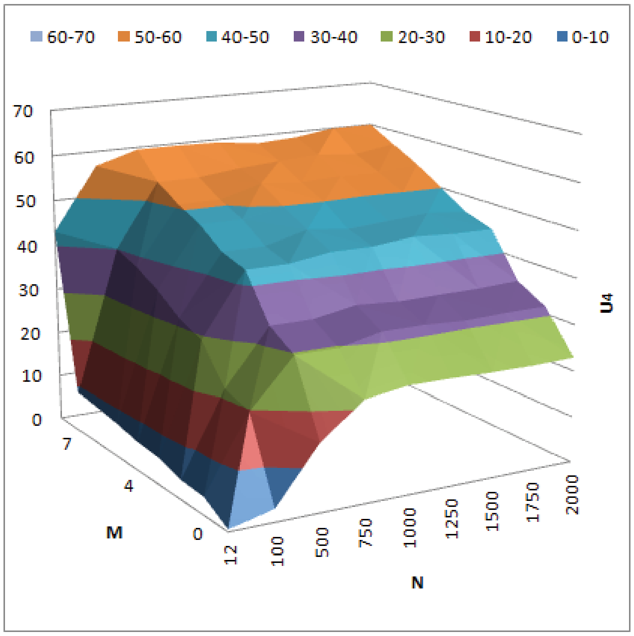

Fig. 5. Normalized CPU utilization U_4 for *Concat* web service

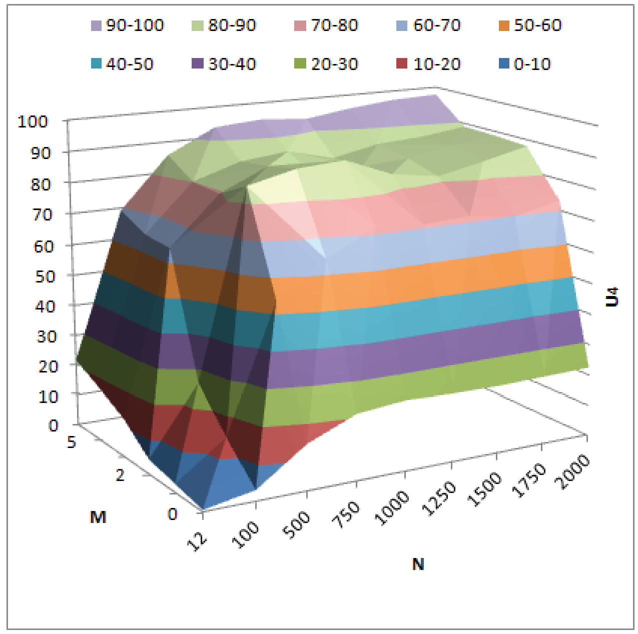

Fig. 6. Normalized CPU utilization U_4 for *Sort* web service

80.37% occurs for $M = 6$ and $N = 1500$.

C. Multi-tenant Cloud Environment $(4x1)$

This test is performed with execution of *Concat* and *Sort* web services hosted in a multi-tenant cloud environment with 4 virtual machine instances, each with 1 CPU core.

1) Test 3.1 Concat web service: Fig. 5 depicts the normalized CPU utilization U_4 for *Concat* web service hosted in a multi-tenant cloud environment with four virtual machine instances, each with one CPU core.

These results for *Concat* web service do not differ a lot from the results presented in Section III-A1, i.e. the CPU utilization depends on the number of concurrent messages with greater increasing factor than on the message size. The minimum CPU utilization of 0.50% is achieved for input parameters $M = 0$ and $N = 12$ while the maximum CPU utilization of 60,06% occurs for $M = 9$ and $N = 2000$.

2) Test 3.2 Sort web service: Fig. 6 depicts the normalized CPU utilization U_4 for *Sort* web service hosted in a multi-tenant cloud environment with four virtual machine instances, each with one CPU core.

We can conclude that for particular number of concurrent messages $N = 1000; 1250; 1500$ and message parameter size $M = 4$, the CPU utilization deviate from the increasing affinity and briefly decreases. The minimum CPU utilization of 0.54% is achieved for input parameters $M = 0$ and $N = 12$ while the maximum CPU utilization of 97.48% occurs for $M = 6$ and $N = 2000$.

IV. Relative Multi-tenant Increase of CPU Utilization

In this section we analyze the relative increase of CPU utilization achieved by the web services hosted in multi-tenant cloud environment in comparison to single-tenant cloud environment. The analysis is performed for both *Concat* and *Sort* web services distinctively.

A. Analysis of Relative Increase of CPU Utilization for Concat Web Service

Using (2) we calculated the relative increase of CPU utilization $R(2)$ for *Concat* web service hosted in a multi-tenant environment with 2 VM instances, each with 2 cores. Fig. 7 shows the relative increase of CPU utilization for *Concat* web service hosted in a single-tenant environment with 1 VM instance with 4 cores versus the same web service hosted in a multi-tenant environment.

We observe that the single-tenant environment performs much smaller CPU utilization for each $N = 12; 100$ and 500 regardless of the values of parameter M. Another important results is that the value of $R(2)$ is near 1 for $M = 0$ regardless of N.

We can conclude that single-tenant cloud environment provides smaller CPU utilization only for smaller number of concurrent messages ($N \leq 500$) for *Concat* web service. However, opposite to our hypothesis, single-tenant cloud environment provides greater CPU utilization for greater number of concurrent messages for *Concat* web service.

Fig. 8 represents the relative increase of CPU utilization $R(4)$ for the *Concat* web service hosted in a single-tenant environment with 1 VM instance with 4 cores versus *Concat*

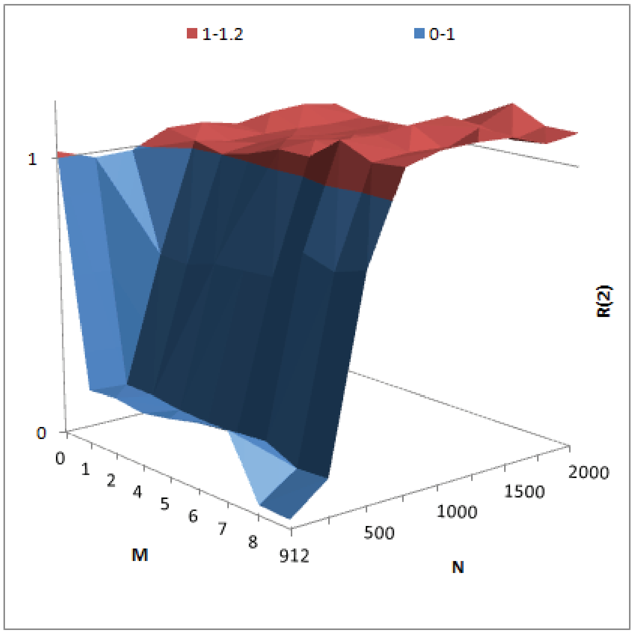

Fig. 7. Increase of CPU utilization $R(2)$ for *Concat* web service

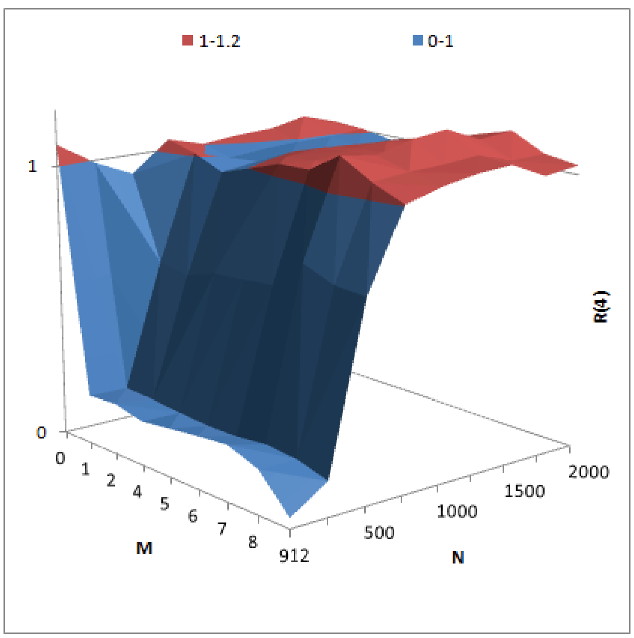

Fig. 8. Increase of CPU utilization $R(4)$ for *Concat* web service

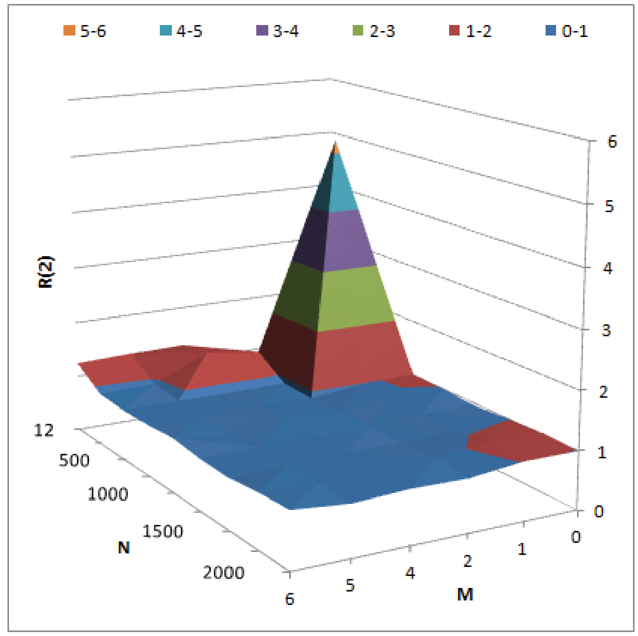

Fig. 9. Increase of CPU utilization $R(2)$ for *Sort* web service

web service hosted in a multi-tenant cloud environment with 4 VM instances each with 1 CPU core.

The results show similar performance as the relative increase of CPU utilization $R(2)$ for *Concat* web service. It means that the single-tenant cloud environment also provides smaller CPU utilization only for smaller number of concurrent messages ($N \leq 500$) for *Concat* web service. Also, opposite to our hypothesis, single-tenant cloud environment provides smaller CPU utilization for greater number of concurrent messages for *Concat* web service. Another important results is that the value of $R(4)$ is near 1 for $M = 0$ regardless of N.

B. Analysis of Relative Increase of CPU Utilization for Sort Web Service

Using the equation in (2) we calculated the relative increase of CPU utilization $R(2)$ for *Sort* web service hosted in a single-tenant environment with 1 VM instance with 4 cores versus *Sort* web service hosted in a multi-tenant environment with 2 VM instances, each with 2 cores. Fig. 9 represents the relative increase of CPU utilization for *Sort* web service versus the same web service hosted in a single-tenant environment.

The results show that the single-tenant cloud environment provides totally opposite results for *Sort* web service compared to *Concat*. We observe that single-tenant cloud environment provides smaller CPU utilization for each $N \geq 500$ and $M \geq 2$. However, although $R(2) < 1$, its value is near 1. Opposite to our hypothesis, there is a region where single-tenant cloud environment achieves greater CPU utilization, i.e. for $N \leq 100$ regardless of the values of M.

Another interesting result, is observation of local extreme for relative increase of CPU utilization of 5.23 achieved for $N = 500$ and $M = 1$. This will be analyzed in our future research and reported in another paper, giving more comprehensive analysis and explanation of cache behavior and its correlation to the CPU utilization. Here, we will analyze only those results related to our hypothesis.

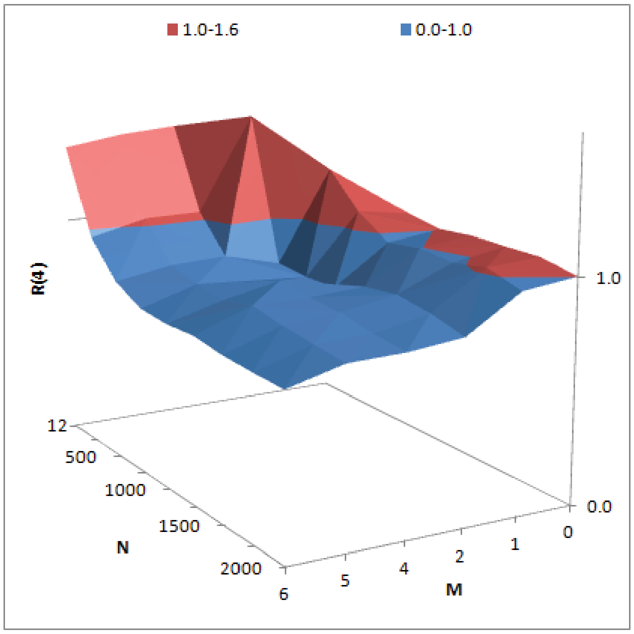

Fig. 10. Relative increase of CPU utilization $R(4)$ for $Sort$ web service

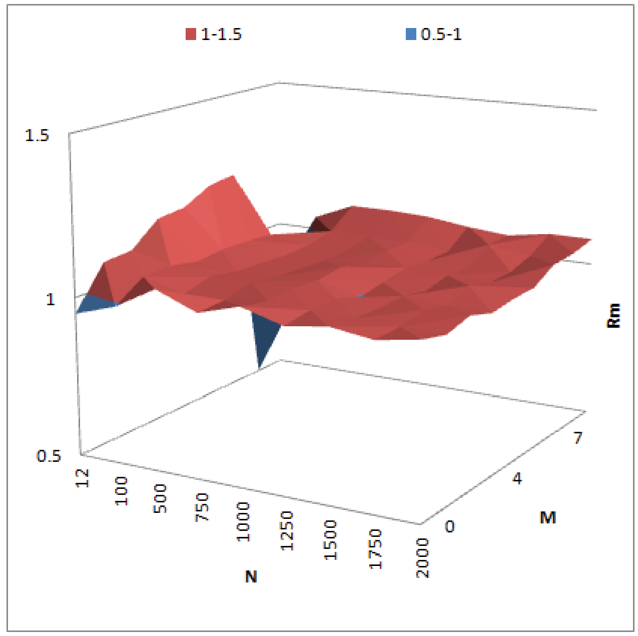

Fig. 11. Relative increase of CPU utilization R_m for $Concat$ web service

Fig. 10 represents the relative increase of CPU utilization $R(4)$ for the $Sort$ web service hosted in a single-tenant cloud environment with 1 VM instance with 4 cores versus $Sort$ web service hosted in a multi-tenant cloud environment with 4 VM instances each with 1 CPU core.

The results show that single-tenant cloud environment provides smaller CPU utilization for each $N \geq 500$ and $M \geq 2$. However, opposite to our hypothesis, single-tenant cloud environment achieves greater CPU utilization when one of the parameters has small value, especially for $N = 12$. We explain this phenomenon with the fact that for small number of messages the OS need smaller time to schedule the tasks on smaller number of concurrent messages.

C. Analysis of Relative Multi-tenant Increase of CPU Utilization

This section presents a comparison of both multi-tenant environments analyzed previously. Further on, we will make conclusions which environment achieves better increase of CPU utilizations.

Fig. 11 shows the relative multi-tenant increase in CPU utilization R_m for $Concat$ web service.

The results show that the multi-tenant environment with 4 virtual machine instances each with 1 core provides greater CPU utilization for each input parameter M and N, excluding some points for $N = 12$.

Fig. 12 shows the relative multi-tenant increase in CPU utilization R_m for $Sort$ web service.

The results show that for Sort web service hosted in multi-tenant cloud environment (4x1) also provides greater CPU utilization for each input parameter $N \geq 500$ and $M \geq 2$.

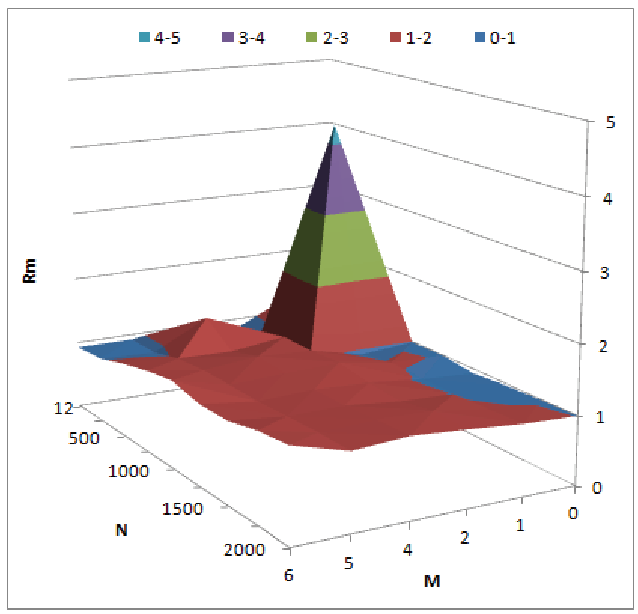

Fig. 12. Relative increase of CPU utilization R_m for $Sort$ web service

We also observe a local extreme $R_m = 4.28$ for the same $N = 500$ and $M = 1$ that appeared in the multi-tenant cloud environment $2x2$. Analysis of this phenomenon and Its explanation is beyond the limits of this paper and is foreseen for future work.

V. Discussion

In this section we will discuss and compare the CPU utilization results for both $Concat$ and $Sort$ web services hosted in particular cloud environments described in Section II-C.

A. Concat vs Sort Web Service in a Single-tenant Cloud Environment

Both of the web services are hosted in a single-tenant cloud environment with 1 VM instance and 4 CPU cores. The results show that both web services achieve increase of CPU utilization while increasing message parameter size M and increasing the number of concurrent messages N. A different behavior is notable for $Sort$ web service, i.e. there is a region where the CPU sharply rises for $M = 1KB$ in comparison to $M = 0KB$ and no further eminent increase is notable for $M > 1$. This steep increase is evidently lower for $Concat$ web service. In this case the CPU utilization is continuously increasing as M rises.

B. Concat vs Sort Web Service in a Multi-tenant (2x2) Cloud Environment

This comparison uses both web services hosted in a multi-tenant cloud environment with 2 VM instances and 2 CPU cores each. Analysis of these results shows that the nature of increasing the CPU utilization for different values for parameters M and N is the same for both $Concat$ and $Sort$ web services. The difference is in the almost two times higher increasing step for $Sort$ web service for $M > 1$.

C. Concat vs Sort Web Service in Multi-tenant (4x1) Cloud Environment

The analysis is performed for both web services hosted in a multi-tenant cloud environment with 4 VM instances and 1 CPU core. In this case, both web services show similar behavior of continuous increase for different parameters M and N.

VI. Conclusion and Future Work

In this paper, we exhibit the performance analysis in terms of CPU utilization while testing two web services hosted in different cloud environments. $Concat$ and $Sort$ web services were tested with different load determined with various message parameter size and number of concurrent messages. The analyses were accomplished for three different cloud environments: single-tenant ($1x4$), multi-tenant ($2x2$) and multi-tenant ($4x1$) cloud environments, i.e. the same runtime environment and total number of CPU cores, but orchestrated differently.

However, opposite to our hypothesis, the results show that there are regions where single-tenant cloud environment provides greater CPU utilization compared to other multi-tenant cloud environments for both web services. $Concat$ web service provides greater CPU utilization than multi-tenant $2x2$ cloud environment for each $N \geq 750$ regardless of parameter size M. It also provides greater CPU utilization than multi-tenant $1x4$ cloud environment for each $N \geq 750$ and for each $M \geq 4$. There are regions where single-tenant cloud environment provides greater CPU utilization for $Sort$ web service either, i.e. when one of the input parameters is small.

The results show that multi-tenant ($2x2$) cloud environment provides smaller CPU utilization for each parameter M and N, except when exactly one of the parameters is small and multi-tenant ($4x1$) cloud environment provides smaller CPU utilization.

In our future research we aim to perform the same analysis on different infrastructure and cloud platforms, and thus compare them with our previous results in order to confirm the presented methodology as an accurate way of testing and deriving conclusions. In addition, we also plan to conduct a more comprehensive research with metrics including both CPU and memory utilization.

The future work also consists of elaboration of local extremes observed in Fig. 9 and Fig. 12, with analysis of correlation among cache behavior and CPU utilization by giving more comprehensive theoretical and experimental explanation.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

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

[3] C.-P. Bezemer and A. Zaidman, "Multi-tenant saas applications: maintenance dream or nightmare?" in *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, ser. IWPSE-EVOL '10. New York, NY, USA: ACM, 2010, pp. 88–92.

[4] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. IEEE, 2010, pp. 27–33.

[5] T. Takahashi, G. Blanc, Y. Kadobayashi, D. Fall, H. Hazeyama, and S. Matsuo, "Enabling secure multitenancy in cloud computing: Challenges and approaches," in *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on*. IEEE, 2012, pp. 72–79.

[6] J. Calero, N. Edwards, J. Kirschnick, L. Wilcock, and M. Wray, "Toward a multi-tenancy authorization system for cloud services," *Security & Privacy, IEEE*, vol. 8, no. 6, pp. 48–55, 2010.

[7] W. Tsai, X. Sun, and J. Balasooriya, "Service-oriented cloud computing architecture," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE, 2010, pp. 684–689.

[8] W. Wang, X. Huang, X. Qin, W. Zhang, J. Wei, and H. Zhong, "Application-level cpu consumption estimation: Towards performance isolation of multi-tenancy web applications," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 439–446.

[9] S. Ristov, G. Velkoski, M. Gusev, and K. Kjiroski, "Compute and memory intensive web service performance in the cloud," in *ICT Innovations 2012*, S. Markovski and M. Gusev, Eds. Springer Berlin / Berlin Heidelberg, 2013, vol. AISC 257, pp. 215–224.

[10] OpenStack, "Openstack cloud software," Oct. 2012. [Online]. Available: http://openstack.org

[11] SoapUI, "Functional testing tool for web service testing," Sep. 2012. [Online]. Available: http://www.soapui.org/

[12] 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.

[13] 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.