Various Algorithms' Behavior in the Cloud

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

***Abstract*—Virtualization is the main technique that enables sharing the hardware resources (CPU, RAM, HDD, network) of a physical server among several users (tenants). It is the main technique that cloud computing relies on, creating multi-tenant scalable and elastic environment with theoretical unlimited resources, which can be rented on-demand. However, virtualization adds an additional layer, which reduces the performance of the hosted applications. In this paper we analyze the behavior of several different types of applications by using the same algorithm in two different situations: run at a host and virtualized in a cloud, respectively.However, the results show that there are applications, which run better in the cloud for specific problem size and load.**

***Index Terms*—Cloud Computing, Virtualization, HPC, Speed, Speedup.**

I. Introduction

The cloud computing virtual environment provides their customers with services following the modern trends by ensuring on-demand resources over Internet via pay-as-you-go model [1]. While the cloud service providers tend to provide low cost resources using pay-as-you-go model and sharing the resources among the costumers, the cloud customers expect to pay less money for rented resources compared to building their own data centers, without risking them to be over-provisioned or under-provisioned [2]. Many different applications and services are already hosted in the commercial clouds, although it is still considered that the cloud computing spreading is still in its infancy. Also, services and SOAs are broadly accepted by the public customers [3]. This means that the cloud computing expansion will be tremendous. It sets the cloud computing as an emerging IT platform, which will take place as the fifth utility after water, electricity, gas and telephony [4].

Despite the benefits that the cloud's virtual, scalable, flexible, multi-tenant environment provides, its main technique, i.e., the virtualization adds an additional layer going from bare metal hardware infrastructure to the applications and their data. It is expected that the scheduling of processes to be executed on a particular CPU core, the translation of memory addresses, the conversion of the MAC and IP addresses and so on, will degrade the performance of the application hosted in virtual machines, despite the fact of whether full or para virtualization is used..

Our hypothesis is that the cloud computing virtual environment degrades the performance of a particular application, service, or some algorithm, compared to their performance when executed on equivalent traditional bare metal environment without using virtualization. In this paper we aim to investigate the performance impact of cloud virtualization over various computation intensive, memory demanding and I/O intensive algorithms and applications: matrix-matrix and matrix-vector multiplication, (a cache intensive algorithms), as well as two web services concat and sort. The former concats two input strings, while the latter concats and sorts the concatenation. The multiplications are analyzed with different problem sizes and different number of CPU cores using parallelization, while the web services are loaded with different input parameters (message parameter size and the number of concurrent messages).

The rest of the paper is organized as follows. Section II describes the testing methodology for different experiments and test cases. In Section III we overview the results for each of the experiment test cases. Finally, Section IV concludes our work.

II. Testing Methodology

In this section, we present the testing methodology that is used for the experiments for all algorithms and web services.

A. Testing Environments

Two testing environments with the same hardware resources were set up for each experiment: Bare metal and cloud virtual environment.

1) Bare - Metal Environment: The bare-metal environment is depicted in Figure 1. All four physical server cores share the ram memory, as well as the last level L3 cache. Each CPU has its private L1 and L2 cache memory. The Linux Ubuntu server operating system is installed with the appropriate runtime environment, which consists of C++ with OpenMP for parallelization for multiplications and Apache Tomcat for hosting the web services.

2) Cloud Virtual Environment: The second environment, i.e. the cloud virtual environment is depicted in Figure 2. In this case, the platform also exists at the same Linux Ubuntu server operating system, C++ with OpenMP and Apache Tomcat. However, having this platform setup under the KVM Virtual Machine is the main difference from the bare metal environment. The KVM is installed on Linux Ubuntu operating system and cooperates with the open source cloud.

B. The Algorithms

Four different algorithms are used in the experiments: matrix-matrix multiplication, matrix-vector multiplication,

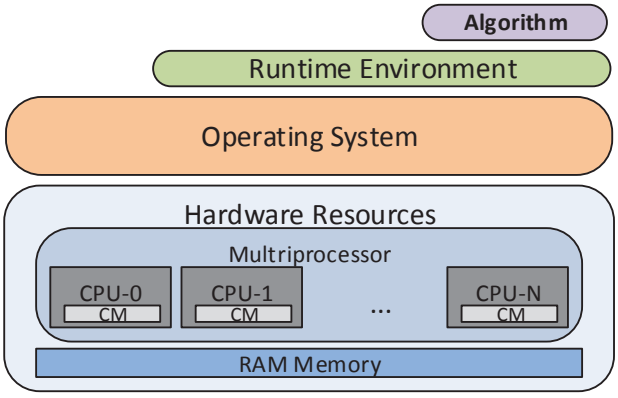

Fig. 1. Bare-metal testing environment

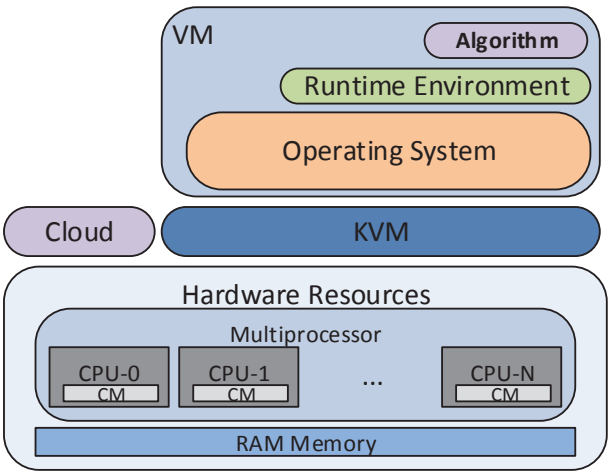

Fig. 2. Cloud computing testing environment

concat web service and sort web service. The following sections describe their implementation.

1) Matrix-matrix Multiplication: A matrix-matrix multiplication algorithm is used with squared matrices with dimension N. The result product matrix $C_{N \cdot N} = [c_{ij}]$ is defined in (1) by multiplying the multiplier matrix $A_{N \cdot N} = [a_{ij}]$ and the multiplicand matrix $B_{N \cdot N} = [b_{ij}]$, for all $i, j = 0, 1, \ldots, N-1$.

$$c_{ij} = \sum_{k=0}^{N-1} a_{ik} \cdot b_{kj} \tag{1}$$

The matrix A is partitioned in row blocks and each block C_i of matrix C is calculated according to (2), which is depicted in Figure 3. The number of blocks depends on the number of used CPU cores and the matrix size N, in order to achieve maximum efficiency. Each block of matrix A is sent for execution on a particular CPU core. All test cases are executed using different matrix size N.

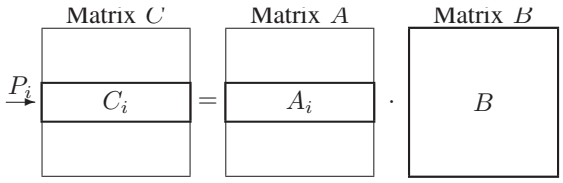

Fig. 3. Parallel Matrix Multiplication [5]

$$C_i = A_i \cdot B \tag{2}$$

2) Matrix-vector Multiplication: Another cache intensive algorithm that is used in the experiments is the matrix-vector multiplication algorithm (3). Each element c_i of the result vector $C_{N \cdot 1}$ is calculated as the inner product of row i of the matrix $A_{N \cdot N}$ and the vector $B_{N \cdot 1}$. Although this is a cache intensive algorithm as the matrix-matrix multiplication, we choose it since only the elements of the vector $B_{N \cdot 1}$ are reused, while in matrix-matrix multiplication the elements of both matrices $A_{N \cdot N}$ and $B_{N \cdot N}$ are reused N times.

$$C_{N \cdot 1} = A_{N \cdot N} \cdot B_{N \cdot 1} \tag{3}$$

Also this algorithm is used both in sequential and parallel implementation, similar as matrix-matrix multiplication.

3) Web Services: We use two different computation intensive and memory demanding algorithms implemented as web services: Concat and Sort. The Concat is a web service that accepts two input strings and returns their concatenation, while the Sort web service returns alphabetically sorted concatenation using sort function.

The experiments are repeated for different input strings with length M and the server is loaded with various number of messages (requests) N, such that to retain server normal workload mode. All test cases are performed in both environments.

Each test case runs for 60 seconds, N messages are sent with M bytes each, with variance 0.5. The accent is on the server response time in regular mode, and neither burst nor overload mode.

III. Behavior Comparison Between Both Platforms

In this section we present the achieved speed and speedup of each algorithm, i.e., matrix-matrix multiplication, matrix-vector multiplication, and both sort and concat web services. The results are presented as relative speed among cloud virtual and bare-metal environments.

A. Matrix-matrix Multiplication Behavior

Figure 4 depicts the results of the experiments for matrix-matrix multiplication algorithm behavior. $V(c)$ denotes the achieved speed for particular matrix size N when the algorithm is executed using c CPU cores.

Different behaviors are observed in each cache region [5]. Despite the hypothesis, the cloud virtual environment provides much better performance in L1 and L2 regions compared

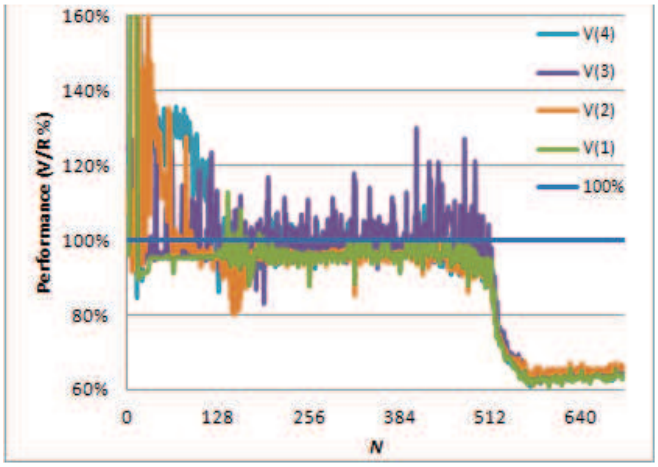

Fig. 4. Measured relative environment speed

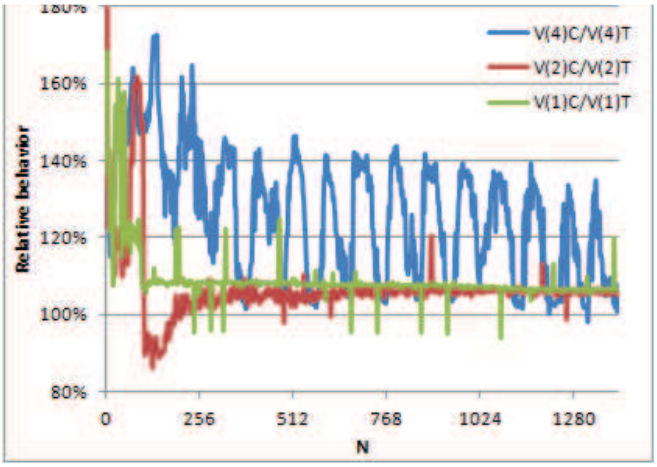

Fig. 5. Measured relative environment speed [7]

L1-L3 Regions	L4 Region
100.68%	65.73%

TABLE I
VIRTUAL VS TRADITIONAL AVERAGE SPEED PERFORMANCE

to traditional. Both environments are similar In L3 region. However, the cloud virtual environment suffers in L4 region compared to bare-metal.

Table I presents the average performance ratio between the cloud virtual and bare-metal environments. L1, L2 and L3 cache regions are groupd in a single group, and L4 region as another. We observe that the cloud virtual environment provides better performance if the matrices can be placed in the L3 cache, while virtualization degrade the algorithm performance in L4 region, i.e., if the matrices cannot be stored in the cache memories and a lot of cache misses are generated. More detailed results are presented by Gusev and Ristov [6].

B. Matrix-vector Multiplication Behavior

The results of the experiments for matrix-vector multiplication algorithm behavior are presented in Figure 5. It is obvious that the cloud virtual environment multiplies a matrix with vector much better compared to its counterpart, since the three curves are above the 100% threshold.

C. Web Services Behavior

This section compares the performance of both web services, hosted in the cloud virtual and bare-metal environments.

1) Concat Web Service: Figure 6 depicts the comparison of the concat web service behaviors. $V(c)$ denotes the achieved speed for particular N when the algorithm is executed using c CPU cores. Indexes C and T denote the cloud virtual and bare-metal environments.

The results show that the concat web service behaves better in the bare-metal, rather than the cloud virtual environment, for each server load. More detailed results are presented by Ristov et al. [8].

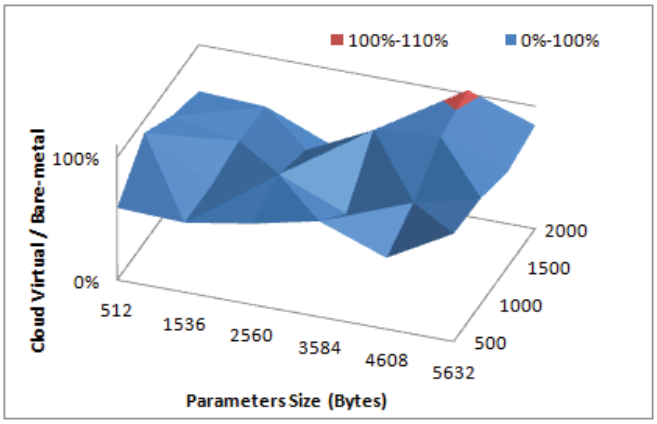

Fig. 6. Relative behavior of concat web service in Cloud virtual and bare-metal environments

2) Sort Web Service: The relative behavior for Sort web service is depicted in Figure 7. The results show very similar behavior such as for concat web service, i.e., the cloud virtual environment is worse than bare-metal environment for each server load.

IV. Conclusion

This paper compares the cloud virtual and bare-metal environments using the same hardware infrastructure for both platforms. Four different algorithms are used: matrix-matrix multiplication and matrix-vector multiplication algorithms (cache intensive algorithms), and concat and sort web services. The experiments disproved the hypothesis that the virtualization will always degrade the performance. That is, we found that cache intensive algorithms is better to be executed in the cloud virtual environment, rather than on bare-metal. However, this holds only if the problem size does not exceed the cache size. Otherwise, the cache intensive algorithms will suffer in a cloud virtual environment. Similarly, the bare-metal environment is much better for both of the tested web services.

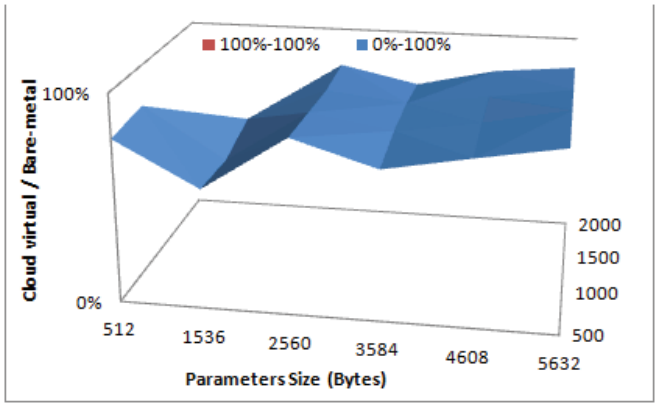

Fig. 7. Relative behavior of sort web service in Cloud virtual and bare-metal environments

ACKNOWLEDGMENT

This work was partially financed by the Faculty of Computer Science and Engineering at the "Ss. Cyril and Methodius University", Skopje, Macedonia.

REFERENCES

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*. IEEE, 2008, pp. 5–13.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] 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.

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.

[5] S. Ristov and M. Gusev, "Superlinear speedup for matrix multiplication," in *Information Technology Interfaces, Proceedings of the ITI 2012 34th International Conference on*, 2012, pp. 499–504.

[6] M. Gusev and S. Ristov, "Matrix multiplication performance analysis in virtualized shared memory multiprocessor," in *MIPRO, 2012 Proceedings of the 35th International Convention*. IEEE, 2012, pp. 251–256.

[7] S. Ristov, G. Velkoski, and M. Gusev, "Can cloud virtual environment achieve better performance?" in *International Conference on Applied Internet and Information Technologies (AIIT)*, 2013, pp. 92–97.

[8] S. Ristov, G. Velkoski, M. Gusev, and K. Kjiroski, "Compute and memory intensive web service performance in the cloud," in *ICT Innovations 2012*, ser. Advances in Intelligent Systems and Computing, S. Markovski and M. Gusev, Eds. Springer Berlin Heidelberg, 2013, vol. 207, pp. 215–224. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37169-1_21