Computing and Informatics, Vol. 38, 2019, 291–320, doi: 10.31577/cai_2019_2_291

AN EXPERIMENTAL STUDY ON VIRTUAL MACHINE LIVE MIGRATION IMPACT ON SERVICES PERFORMANCE

Petrônio Bezerra, Marcela Santos, Edlane Alves
Anderson Costa

Federal Institute of Paraíba (IFPB)
Campina Grande Campus
Tranquilino Coelho Lemos, 671 – Dinamérica
58432-300 Campina Grande, PB – Brazil
e-mail: {petroniocg, anderson}@ifpb.edu.br

Fellype Albuquerque, Gustavo Martins, Reinaldo Gomes

Federal University of Campina Grande (UFCG)
Department of Systems and Computation
Aprígio Veloso, 882 – Universitário
58429-900 Campina Grande, PB – Brazil
e-mail: gustavo.martins@copin.ufcg.edu.br, reinaldo@dsc.ufcg.edu.br

Abstract. One important benefit of servers' virtualization is the reduction of the maintenance complexity of infrastructures. A key feature is servers' live migration which allows virtual servers to be exchanged between physical machines without stopping their services. However, virtualization also has some drawbacks caused by the overhead generated. Our research evaluated live migration process overhead, on real and virtual environments, noticed from the client's side regarding two different services: web and database. YCSB and *ab* Benchmark were adopted as workloads. Almost all tests on real environment overcame those on virtual, with both benchmarks. The impact of the live migration in the services was evident, proving to be more effective on real machines than on virtual machines. We found the DB service accommodated better to the virtual environment and to migration than Web service. We also considered an environment with multiple migrations which presented a higher degradation than when only one migration is performed.

Keywords: Migration, virtualization, overhead, evaluation, Xen

Mathematics Subject Classification 2010: 62-J02

1 INTRODUCTION

Cloud computing is currently the most important technology to support on-demand allocation and delivery of computing resources. These resources can be servers, storage systems, networking, databases, etc. These cloud computing resources are organized in levels as a service aiming to better address customers' requirements. In the market view, cloud computing in recent years has had a rapid growth, as highlighted in [1] presenting the US spending projection in cloud computing between 2010–2015. An annual growth of around 40 % was predicted, reaching an investment of USD 7 billion. Around that period, analysts predicted a growth in cloud's global market estimated at USD 95 billion in addition to 12 % of software migration to the cloud market. At this point, these growths have driven the research community to investigate topics concerning cloud computing.

According to [2], it is possible to migrate a computational infrastructure to a remote location with a minimal impact on system performance. This is one of the advantages obtained with the use of cloud computing. Many companies have migrated their computing infrastructures to cloud environments because of the benefits of virtualization [3]. However, it is necessary to know the performance of services deployed in Virtual Machines (VMs) when they are running in a cloud computing environment [5]. When you migrate an IT infrastructure to a public cloud, knowledge of how your services can be impacted is crucial. It is worth noting that in cloud environments the goal is often to save energy and make the best use of available resources [4]. These goals may be antagonistic in relation to the better performance of the services deployed in VM.

A technology commonly adopted by cloud computing providers is virtualization [2, 8]. This approach involves a software layer, located between the hardware and Operating System (OS), which is called a Virtual Machine Monitor (VMM) or hypervisor [6]. It allows running different VMs with different OS on a single physical machine at the same time. There are many advantages obtained through virtualization, among them we can quote:

1. Server consolidation: The possibility of running many virtualized servers at the same time in a single physical server facilitates financial savings in acquisition and hardware maintenance.
2. Energy savings: Each instance of VM running in a server represents a physical machine, thus, as many more VMs can be running on single server, the power consumption would be less compared to installing new physical machines to run new services.

3. Load balance: The number of VMs on a single physical server should not compromise the level of services offered by virtualized servers.

4. Easy management: VMs adoptions facilitate server management, for instance, in backup procedures.

5. Migration: A feature reached by virtualization, migration allows movement of a server between different hosts [4].

This procedure can be undertaken without interference or downtime to its services, which is called live migration.

Live migration itself brings many benefits to a cloud provider's environment. For example, the reallocation VM procedure, which uses server's migration, allows better usage of a physical machine, savings in resources and maximizing profit. Nevertheless, despite the benefits of virtualization, overhead generated through virtualization layer has already been the target of many investigations. Regardless of migration, such studies show the impact on the performance of the services hosted on virtualized servers [7].

In this paper, we are interested in checking the impact perceived by a client host accessing services, such as web server and DBMS, running on a virtual server during a live virtual machine migration process, in contrast to most investigations about virtualization overhead which only adopted benchmarks that run on the server side, consequently only checking the overhead effect on server side. The goal of our research is to analyze the overhead effect on the client's side, contrary to what has been proposed by other investigations. This is important because the users through their hosts (clients) are the most interested and affected by services performance. Furthermore, these overhead effects were observed during a live migration procedure. We believe that our findings can help decision makers in cloud computing better address their requirements and to decide if/when a migration might be performed to reduce its impact.

The remainder of the article is organized as follows: Section 2 presents techniques responsible for performing the virtualization process. In Section 3 we have the related works. Then in Section 4 we describe the experiment setup and materials. The analysis of the data is in Section 5. We move on with performance evaluation based on our experiment setup in Section 6. In Section 7, we conclude this paper and discuss the future work.

2 RESOURCE VIRTUALIZATION

There are some techniques responsible for performing the virtualization process according to the way how physical resources will be accessed and allocated to the virtual machines.

2.1 Virtualization Techniques

These techniques allow the isolation and abstraction of the underlying hardware and lower level functions [11]. Different approaches are:

Full Virtualization: Here, the VMM runs on top of a host operating system similarly to other applications, in the user space. In this case, the entire physical platform (CPU, memory, disk, etc.) is being virtualized. The overhead caused by this form of virtualization can be quite significant.

OS-Layer Virtualization: In this approach, the guests' OS is being virtualized instead of the hardware. The virtualizations here occur by running more instances of the same OS in parallel. Because of that, the VMs must use the same kernel as the host OS, not being possible to virtualize a different OS. This is a restriction of this approach.

Paravirtualization: The main difference from the full virtualization is that, in paravirtualization, the guest's OS must be modified. This technique allows specific guest machines to communicate directly with the hardware, rather than communicating with the VMM. For this reason, it offers better performance but has the restriction of having to modify the guest OS.

2.2 Strategies for VM Migration

The VM migration is a very important characteristic for cloud computing environments. The two main ways to perform VM migration are:

Stop-and-copy (or non-live migration): In this approach, the VM is suspended on the source host, it migrates to the destination host through copies of the memory pages and other necessary information, and then the VM is activated again at its destination. It is a more simple way to perform the migration procedure, besides being faster than live migration. However, it presents greater downtime of the applications running on the virtualized server.

Live migration: In live migration, there are several iterations of copying the memory pages from the source host to the destination host. During this process, the services deployed in the VM are still in operation. In this way, the downtime of the applications is minimized and being more interesting in some situations. However, the total migration time is greater in this approach than in stop-and-copy.

3 RELATED WORK

The virtualization theme has been very exploited in cloud computing community, and the performance analysis is a major subject of these investigations. The most employed strategy to compare hypervisors is to apply series of benchmark software to test a wide range of system devices, for example I/O, disk, memory, networking and processors [10, 12, 13, 14]. Although being the excellent sources of research, none of the cited studies took into consideration the performance evaluation of a specific application running on VM. In those articles, the authors use benchmark software to test a device of the evaluated system.

The following two studies considered the performance evaluation of a specific application running on VM.

In [9], they evaluated two virtualization technologies, Xen and OpenVZ, comparing both technologies to a base system in terms of application performance, resource consumption and scalability, among others. The workload used in this study was RUBiS. They found that the average response time could increase over 400 % in Xen and 100 % in OpenVZ as the number of application instances grew from one to four.

Another similar study using databases running on VM, and focused on the impact of virtualization layer, can be found in [15]. In this study, the overhead of the virtualization layer was evaluated in different databases deployed in VMs, Cassandra and MongoDB, the overhead was observed to consider different virtualization techniques, full virtualization and paravirtualization, when compared to a physical host. The authors executed the YCSB benchmark to evaluate databases' performances. In the findings, virtualization technique was the factor that showed a higher influence over databases' performance compared to that obtained from a physical host. MongoDB reached better performance in most of scenarios.

None of those articles developed a performance analysis of applications running on VMs during the occurrence of a VM live migration process. The other studies in sequence did that.

Another interesting paper, similar to what has been developed, can be found in [16]. In this study, the VM migration was evaluated regarding performance during migration, performance of cloud architecture during VM migration and the energy cost of real-time migration. The goal was to understand how the cloud architecture would respond and deal with real time migrations. The results obtained showed how a virtual machine performs during a live migration. This differs from our proposal in two aspects: we are not interested in evaluating the energy cost and our study was made without the use of the cloud.

In [17], the authors demonstrated how resource consumption and latency can be substantially reduced, allowing better VMs migration performance. Initially, they experimentally studied the factors that contributed to the growth of these two variables on migration. They proposed an alternative technique of remote access memory which significantly reduces the overhead on the migration of VMs. Through simulations and experiments, the authors reduced the overhead in the migration of VMs, resulting in improvements in energy and resource efficiency over the techniques that already exist.

The main distinction of our work is that our objective is to evaluate the impact, from the client's point of view, that a live migration process cause in applications running on VM.

4 EXPERIMENT SETUP

This paper investigates the overhead caused by a live migration of a server in two common applications on the client's side. Like many other studies, we also use benchmarks tools to measure performance. Most research that uses benchmarks do this on server's side, instead, here benchmark software it was used on client's side. This is a differential from other researches.

In this experiment, aspects of performing a VM live migration were evaluated. The aspect concerned was the performance of the services, perceived by a client when a live migration is running on a server. To support machines virtualization and migration Xen hypervisor[1] was used. Xen is an open source hypervisor which enables to use full virtualization and paravirtualization techniques. In our study, only the paravirtualization technique was used, in which guest systems know they are being virtualized because their kernels need to be modified, improving the performance achieved compared to full virtualization approach [9]. Xen was chosen as a virtualization platform for our experiment because it is an open source platform, commonly adopted in investigations concerning live migration [7, 9, 10].

The services used were Apache Web Service™[2] and the Apache Cassandra™[3] a Database Management System (DBMS). Two benchmarks (*ab* and YCSB) were used in our experiments to fulfill a series of performance tests, they are described below:

Apache HTTP Server Benchmarking Tool (*ab*)[4]: Is a tool for benchmarking Apache Hypertext Transfer Protocol (HTTP) servers. It is designed to test how the Apache installation performs. In our experiments, it was used with the objective of generating workload to the VMs on Xen servers. It is important to highlight that this benchmark will simulate the requests made by a user to a web server. With *ab*, it was measured the mean number of requests per second.

Yahoo! Cloud Serving Benchmark (YCSB)[5]: This benchmark aims to generate workload for non-relational database (NoSQL). It is possible to evaluate the performance of the database through information such as average latency and throughput. In our experiments the Apache Cassandra NoSQL database was used. With YCSB the number of operations per second was measured using the Workload option, which makes 50 % of load operations and 50 % of select and update operations.

Initially, the experiments were divided into phases – with environment totally virtualized, with only real machines and with multiple migrations.

[1] `http://www.xenproject.org/`
[2] `http://www.apache.org/`
[3] `http://cassandra.apache.org/`
[4] `https://httpd.apache.org/docs/2.4/programs/ab.html`
[5] `https://github.com/brianfrankcooper/YCSB/wiki`

A. Virtualized Servers

1. *The Environment:* It consisted of one real machine which contained all the virtualized hosts. Figure 1 gives a representation of the simulation environment. It is possible to see the host machine where whole experiments of this phase took place. This host contained four VMs: XenServer1, XenServer2, Client, Network File System (NFS) Server and a Virtual Ethernet Switch. These VMs were virtualized by the VMware Workstation Player[6] free hypervisor.

   The concept of nested virtualization is not new and can be found in [18].

   The Apache Web Server and DBMS Cassandra were installed on the same server, which was called WebServer and was virtualized by Xen. The Hypervisor Xen was installed on two Xen Servers (XenServer 1 and 2) that were used for migration of the Web Server host. NFS is the server responsible for sharing virtual machines' images and virtual disks involved in the migration process. The existence of this element is one of the requirements of the Xen hypervisor [19]. The live migration process is highlighted in Figure 1.
2. *Hosts Settings:* The real machine used to install the virtual hosts in this phase had 16 GB RAM, Intel® Core™ i7-4790 CPU @ 3.60 GHz 64 bits and 680 GB of disk. This CPU was the same used at all virtualized hosts. Table 1 gives the virtual machine descriptions. All the machines (real and virtual) used on the experiment had Ubuntu 14.04 Desktop as the Operating System (OS), except the Web Server host, which had Ubuntu 14.04 Server.

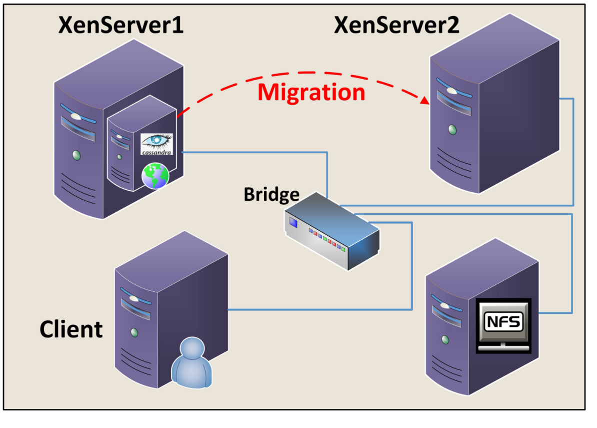


Figure 1. Environment totally with virtualized machines

[6] `http://www.vmware.com/br/products/workstation`

Hardware	XenServer1/XenServer2	WebServer	NFS	Client
Memory	6 GB	2 GB	4 GB	2 GB
Disc Size	40 GB	10 GB	25 GB	25 GB

Table 1. Virtual machines descriptions

B. Physical Servers

1. *The Environment:* it consisted of four real machines with one virtualized host that migrated between Xen Servers 1 and 2 using the same organization presented in Figure 1. The real hosts were: XenServer1, XenServer2, Client, the NFS and an Ethernet Switch. The Apache Web Server and DBMS Cassandra were installed on the same server, which was called Web Server and was virtualized by Xen. The Xen Hypervisor was installed on two Xen Servers (1 and 2) that were used for migration of the Web Server host. The real environment used the same machine's configurations as presented on virtual environment.

C. Experimental Design

Our experiments consisted of a $2^k r$ factorial design [20] to determine the effect of k factors, in our case $k = 2$ that are two factors each at two levels. We used $r = 10$, which means we made ten repetitions in each treatment. Following this design, it was made $2^k r = 2^2 10 = 40$ observations for each benchmark.

Table 2 gives the factor level combinations for each experiment made with benchmarks YCSB and *ab*. We considered two factors, Environment and Condition, each with two levels that were Real Machine or Virtual Machine in Environment factor and With Migration or Without Migration in Condition factor. These are our primary factors whose effects were quantified.

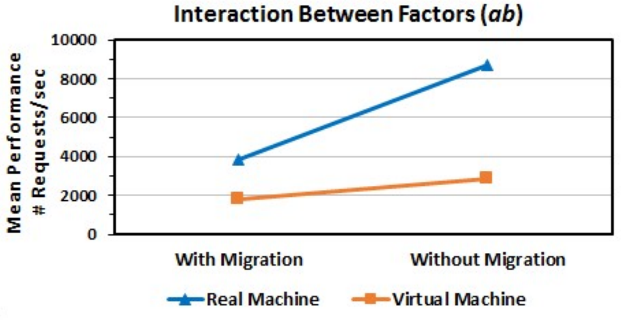
Factor	Level -1	Level 1
Environment	Real Machine	Virtual Machine
Condition	With Migration	Without Migration

Table 2. Factors and levels of the design

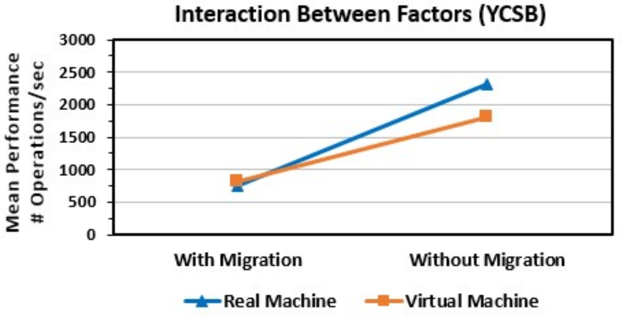
We know there are secondary factors that impact the performance but such impacts were not considered in quantifying. For example, we were not interested in determining whether performance with *ab* is better than that of with YCSB.

5 STATISTICAL ANALYSIS

To decide how to analyze the data collected, some statistical reviews were performed as follows. Two factors A and B are said to interact if the effect of one depends upon the level of the other one. Figures 2 a) and 2 b) give the interaction between Condition and Environment levels for *ab* and YCSB benchmarks. As shown in the graph for *ab*, the lines of Real and Virtual Machines are not parallel, indicating an interaction between them [20]. The same can be observed from YCSB's graph.



a) *ab* benchmark

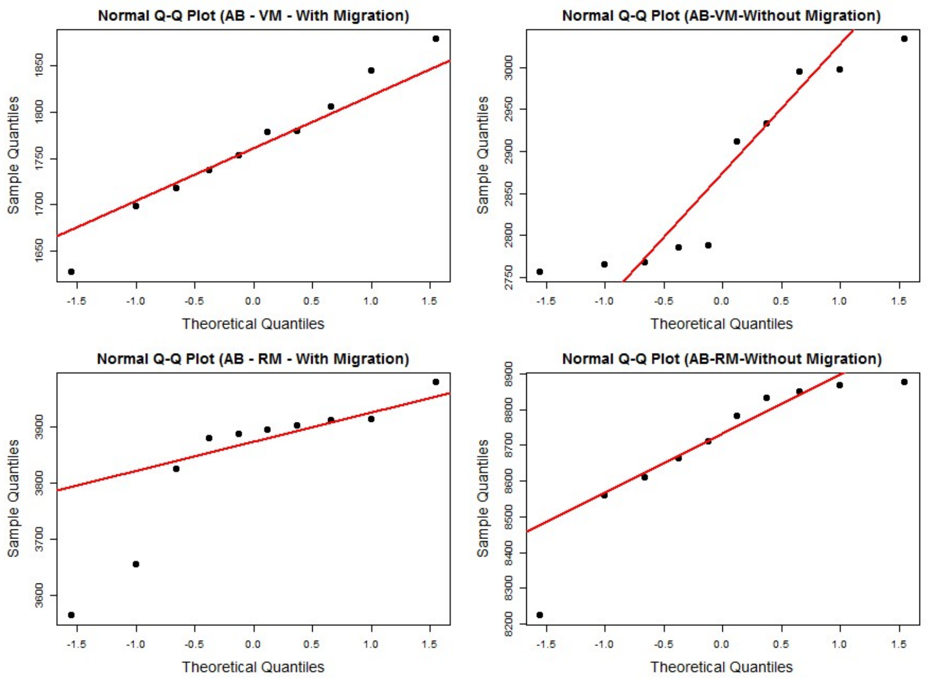


b) YCSB benchmark

Figure 2. Graphical presentation of interacting factors for a) *ab* benchmark and b) YCSB benchmark

Many statistical techniques assume the data are normally distributed, facilitating data analysis. Our data were evaluated aiming to verify whether data meets normal distribution. Some statistical tests were performed, as the Shapiro-Wilk test (whose parameter is W), and p-values less than 0.05 with $W < 1.0$ were found. In almost all results the null hypothesis was refuted (i.e., if p-value is less than the 0.05 significance level and $W < 1.0$, the null hypothesis is rejected). Only one test with a p-value = 0.9963 (> 0.05) and W very close to 1.0 ($W = 0.9896$) was that with *ab* on VM and with live migration. This result agrees with the chart on Figure 3 a) upper left where it is possible to see the most points fall along on a straight line.

As can be observed from all the others charts on Figure 3, few points fall along on a straight line. We conclude that the data samples collected come from a different distribution than normally. These results, based on graphs and in statistical test analysis lead to the decision to use non-parametric statistical tests. From the



a)

analysis made, we could conclude that the most appropriate test for these samples is the Mann-Whitney-Wilcoxon Test (or Wilcoxon rank sum test, or Mann-Whitney U-test), a non-parametric test by which we can decide whether the population distributions are identical not assuming they would follow the normal distribution [21].

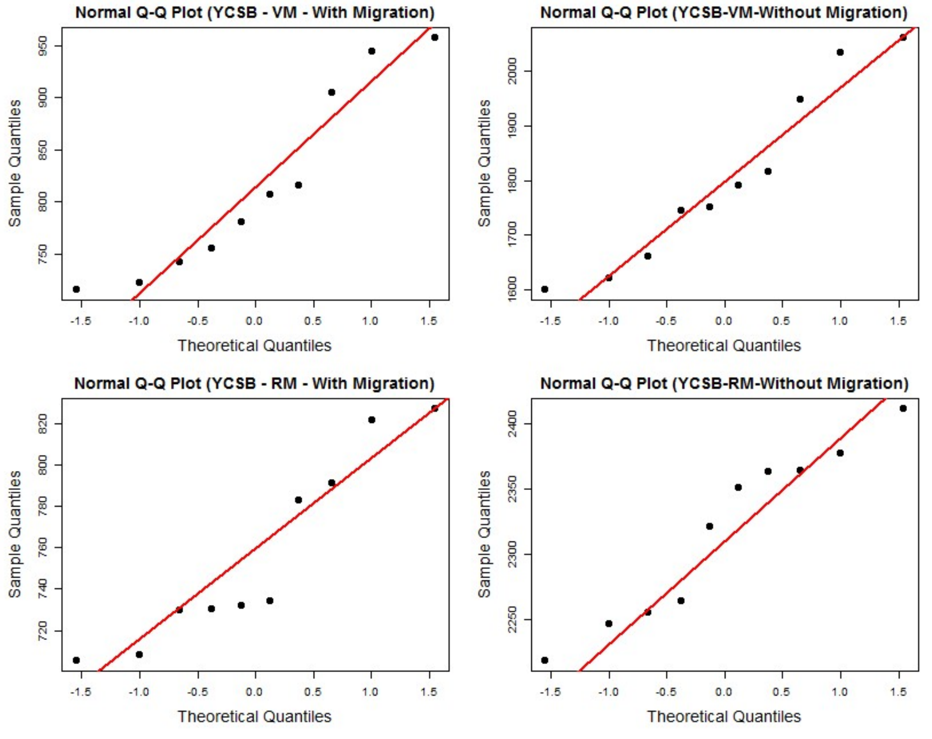
Many hypotheses could be tested with our results but, we believe the most important for statistical testing is to identify in which environment (real or virtual) a service deployed on VM provides a better performance during a migration event. This information is important to understand the client's perception of the service.

With this objective, the following hypotheses were tested:

- For the Web server tested through benchmark *ab*:

H-I$_0$: The performance obtained by the benchmark *ab* on real environment, during a migration process, ***is equal to*** the performance obtained on a virtual environment.

H-I$_1$: The performance obtained by the benchmark *ab* on the real environment, during a migration process, ***is greater than*** the performance obtained on a virtual environment.



b)

Figure 3. Normal quantile-quantile plot data for a) *ab* benchmark on Virtual and Real environments and b) YCSB benchmark on Virtual and Real environments

- For the DBMS server tested through benchmark YCSB:

  H-II$_0$: The performance obtained by the benchmark YCSB on real environment, during a migration process, **is equal to** the performance obtained on a virtual environment.

  H-II$_1$: The performance obtained by the benchmark YCSB on the real environment, during a migration process, **is greater than** the performance obtained on a virtual environment.

6 PERFORMANCE EVALUATION

To measure the impact, perceived by the host client, of the live migration of the applications running inside the virtual server, two benchmarks were used, one for each of the tested services (web server and DBMS). The benchmarks were running individually on the Client to test the web server and DBMS on remote Web Server host, which was virtualized on Xen Server1. Measurements were made on servers both with and without migration. Benchmarks were configured as follows:

Apache benchmark (*ab*): The concurrency level was fixed in 10 parallel users using services. The number of multiple requests to the URL of the web server was fixed at 10 000 000. The maximum number of seconds to spend for benchmarking was fixed at 400 seconds. And 10 repetitions were made with these configurations. The response metric was the number of answered requests per second.

Yahoo! Cloud Serving Benchmark (YCSB): The number of client threads was fixed on 5, which indicates the amount of load offered against the database. The number of records to be used on the test was fixed at 100 000. The core workload used was a mix of 50 %/50 % of load and select and update operations. The number of operations to perform was fixed at 600 000. In the same way, 10 repetitions of the test with these configurations were performed. The evaluated metric was the number of operations per second.

A. Performance with Benchmark *ab*

A performance reduction is common for a system under migration, which is a function of time (i.e., conditions are different at each stage: pre-migration, mid-migration and post-migration). It is possible to see this pattern with data collected on client, with *ab* benchmark, during a migration event in Figure 4 (the same behavior was observed with YCSB's data). However, for the purpose of this work, it is desired to establish an idea of the impact caused in a service, running in VM, during a complete migration process. To this end, the temporal variations were discarded aiming the interpretation of the process as a whole.

Given a few factors that affect the system performance, it is important to know the effects of each factor individually [20].

Table 3 gives the factor level combinations for each experiment made with benchmark *ab*. The effects were quantified using the mean of ten repetitions by treatment.

From Table 3 we can see that, on average, the virtual machine processing capacity in number of requests/seconds was equivalent to 33 % of the capacity reached by real machines in a non-migration environment. The results obtained with migration show that the virtual machines had 46 % of the capacity when compared with the real machines.

Environment	**With Migration**	**Without Migration**
Real Machine	3 840.81	8 697.43
Virtual Machine	1 761.77	2 873.10

Table 3. Performance in number of answered requests/sec with *ab*

Figure 5 demonstrates the performance of the *ab* benchmark running on virtual and real environments. The lines above each bar in the bar graphs are

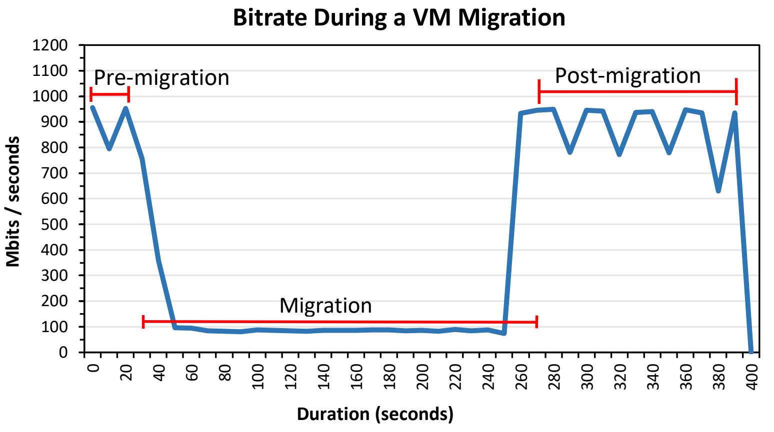


Figure 4. Throughput during a complete *ab* benchmark run on real environment

the standard errors bars. As shown in the graph, the performance of the web server was better running without migration than when a migration occurs as expected.

Comparing the individual means on the virtual environment, the results with migration reached the equivalent of 61 % of the number of requests/seconds obtained by the experiments without migration, indicating a considerable reduction in processing capacity because of the migration process.

When analyzing the results of *ab* on the real environment (Figure 5), the enhancer of performance on real machine when comparing with the results on VM is clear. Again, the performance without migration exceeded the one with live migration in all ten rounds, with a larger contrast between the conditions than that observed on VM.

In the real machine environment, experiments with migration only obtained 44 % of the processing capacity of the environment without migration. The overhead caused by the imposition of a virtualization layer in the computing environment as well as the live migration process becomes evident. This simply reflects the services offered by virtualized servers. This fact was also noticed in testing with DBMS.

Considering all the comparisons, it is possible to check that the impact of migration on the real environment was high, however, lower than on the virtual environment, since in the former we had a reduction of 44 % of the capacity without migration, while with the virtual machines a reduction of 61 % of the capacity can be observed when comparing the environment with and without migration.

Based on these results, we want to highlight that the decision makers should have a better understanding of the impact on the services offered by virtual

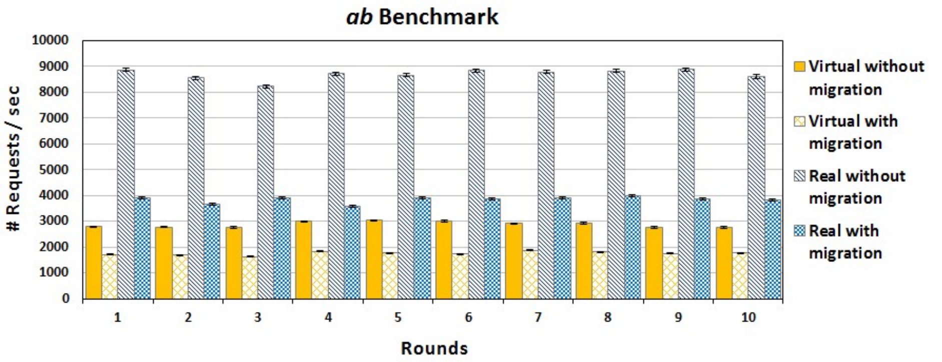


Figure 5. Performance of *ab* on virtual and real environments (higher is better)

servers. In this way, they can weigh the pros and cons and decide more safely whether to move or not their services to Cloud providers.

B. Performance with Benchmark YCSB

Table 4 gives the factor level combinations for each experiment made with benchmark YCSB. As with *ab*, the effects were quantified taken the mean of ten repetitions by treatment.

The results of the ten rounds on virtual and real environments with YCSB can be seen in Figure 6. As with the results of *ab*, YCSB on VM also prevailed in all rounds when running without migration, when compared with the environment migrating the server. Analyzing the individual means on the virtual environment, the results with migration reached the equivalent of 45 % of the performance of the ones without migration in number of operations/second.

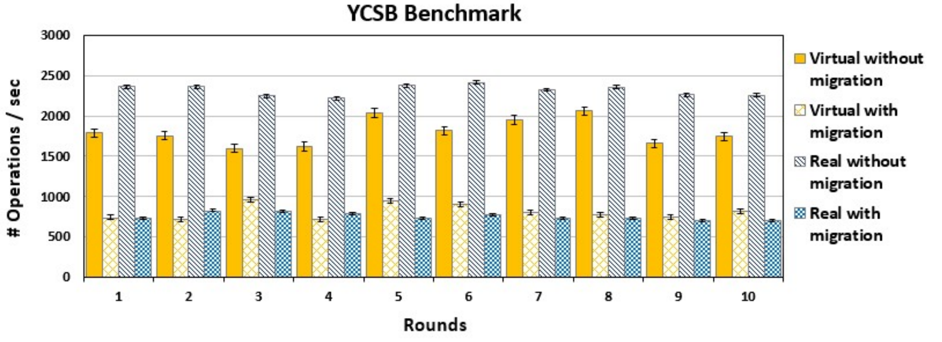


Figure 6. Performance of YCSB on virtual and real environments (higher is better)

When analyzing the performance of the YCSB, with and without migration, on the real environment, the results show that the tests without migration also

Environment	**With Migration**	**Without Migration**
Real Machine	756.26	2 316.85
Virtual Machine	814.49	1 802.68

Table 4. Performance in operations/sec with YCSB

overcame those with live migration with a significant difference, since migration condition obtained only 32 % of the performance from the non-migrating environment.

A very interesting result from YCSB evaluation is that, when migrating the machines, the virtual environment obtained better results than the real one, as presented in Table 4. Analyzing the migration logs, it was possible to see that the number of pages transferred during migration was equivalent in both environments (about to 1.4 million pages), but transferring rate was higher on the virtual environment than on the real environment. Since a considerable amount of information from the virtual machines remains only in memory, to reallocate this memory to another process in the same machines was faster than to transmit this information over the network. Because of it, the benchmark started to run without migration overhead faster on the virtual environment than on the real one (approximately 2 minutes), what was not noticed in *ab* benchmark, since it obtained the same migration time in both environments.

Also, performance without migration on real environment was much closer to the virtual environment than the result obtained using the Web Server. Using DB, the virtual environment had 78 % of the performance of the real environment, while using Web Server, the virtual machine obtained only 33 % of the capacity when running without migration. It is possible to note with these results that DB service (which demands more processing capacity) has been accommodated much better to the virtual environment and to migration than Web service (demanding more access to stored information).

To test the previously made null-hypotheses we applied the Mann-Whitney-Wilcoxon U-test, using the R[7] statistical software, all tests used a significance level of 0.05. The results were as follows:

For the Web server tested through benchmark *ab*: the p-value turns out to be 5.413e−06, and is less than the 0.05 significance level, we reject the null hypothesis $H\text{-}I_0$ and accept the alternative $H\text{-}I_1$. *That is, the performance obtained by the benchmark ab on the real environment, during a migration process, **is superior to** the performance obtained on the virtual environment.*

For the DBMS server tested through benchmark YCSB: the p-value turns out to be 0.9173, greater than the 0.05 significance level, in this case,

[7] `https://www.r-project.org/`

we can accept the hypothesis $H\text{-}II_0$ of statistical equality of the means of two groups. *That is, the performance obtained by the benchmark YCSB on the real environment, during a migration process,* ***is equal to*** *the performance obtained on the virtual environment.*

C. Allocation of Variation

As the data from the experiments suggests, there is a difference between the treatments and the interaction between factors, as shown in Figure 2. We would like to confirm those expectations. To do so, box plots were made (*ab* and YCSB benchmarks on virtual and real machine environments), in Figures 7 a) and 7 b) we have the results for *ab*, and in Figures 8 a) and 8 b) we have the results for YCSB. As can be seen, the graphs suggest differences between the treatments used. Then, it was decided to measure the allocation of variation. The percentage of variation explained by each factor is helpful in deciding whether a factor has a significant impact on the response [20]. The factors which explain a high percentage of variation are considered important.

With the factors and levels of Table 2, following the $2^k r$ factorial design described in [20], let us define two variables x_A and x_B as follows:

$$x_A = \begin{cases} -1, & \text{with migration,} \\ 1, & \text{without migration,} \end{cases} \tag{1}$$

$$x_B = \begin{cases} -1, & \text{real machine,} \\ 1, & \text{virtual machine.} \end{cases} \tag{2}$$

The performance y in number of requests/second (for *ab*) or operations/second (for YCSB) can now be regressed on x_A and x_B using a nonlinear regression model of the form:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + e. \tag{3}$$

The terms in (3) are: y is mean performance; x_A is the effect of condition; x_B is the effect of the environment; x_{AB} is the effect of interactions between environment and condition; q_0, q_A, q_B and q_{AB} are the effects; and e is the experimental error.

First, we used the sign table to analyze our $2^k r$ factorial design and compute the effects, as described in [20].

1. *Allocation of Variation for ab*: following (3) and with the data from Table 3 through the sign table, the model (4) of mean performance from *ab* was developed:

$$y = 4\,293.28 + 1\,491.99 x_A - 1\,975.84 x_B - 936.32 x_A x_B + e. \tag{4}$$

Thus, making the calculations, we found that the total variation can be divided into four parts. Factor B (environment) explains 55.58 % of the variation. Factor A (condition) explains 31.69 % of the variation and interaction AB explains 12.48 % of the variation. The remaining 0.24 % is unexplained and attributed to errors.

From these *ab* results, we can conclude that the environment had more impact in *ab* tests than the condition (using VM live migration or not).

2. *Allocation of Variation for YCSB*: following (3) and with the data from Table 4 through the sign table, the model (5) of mean performance from YCSB was developed:

$$y = 1\,422.57 + 637.19x_A - 113.98x_B - 143.10x_Ax_B + e. \tag{5}$$

Thus, making the calculations, we found that the Factor A (condition) explains 90.46 % of the variation. Factor B (environment) explains 2.89 % of the variation, it can be ignored, and interaction AB explains 4.56 % of the variation. The remaining 2.08 % is unexplained and is attributed to errors.

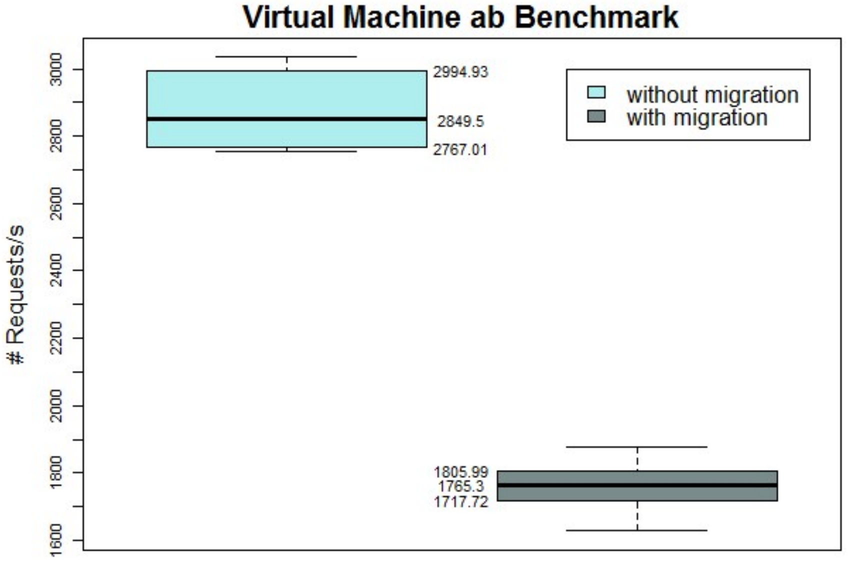
From the YCSB results, we can conclude that the condition had almost all impact in YCSB tests. Use the DB on the real or virtual machine makes no difference in our tests since each environment had an advantage in one of the tests.

In deriving the expressions for effects, we made some assumptions. These assumptions lead to the observations being independent and normally distributed with constant variance. To verify these assumptions, that were made with the regression model, it was decided to use visual tests. To do that, Figures 9 a) and 9 b) give a plot of residuals and a normal quantile-quantile plot for *ab*. As there is no trend in Figure 9 a), we can assume the errors are independently and identically distributed. In Figure 9 b) the residuals appear to be approximately normally distributed. Thus, the model appears to be valid for our experiment with *ab*.

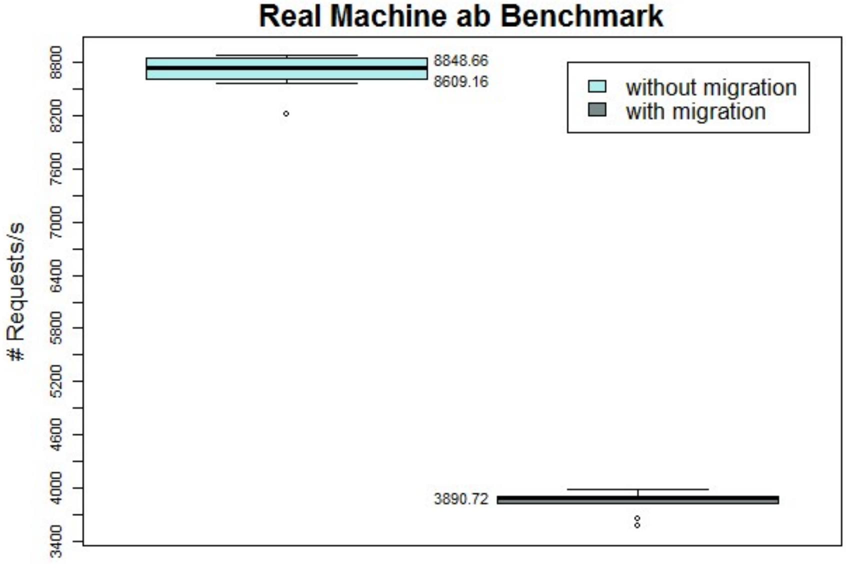
The same analysis made for *ab* can be applied to YCSB, using Figures 10 a) and 10 b). We can assume the errors are independently and identically distributed, and the residuals appear to be approximately normally distributed as well. Thereby, the model also seems to be valid for our experiment with YCSB.

D. Confidence Intervals for the Effects

As seen in Section 4, the errors were normally distributed and, as calculated, this distribution has zero mean, it was possible to discover the confidence intervals for the effects. This information makes possible to find if the effects are significant. The standard deviation of errors can be estimated from the sum of

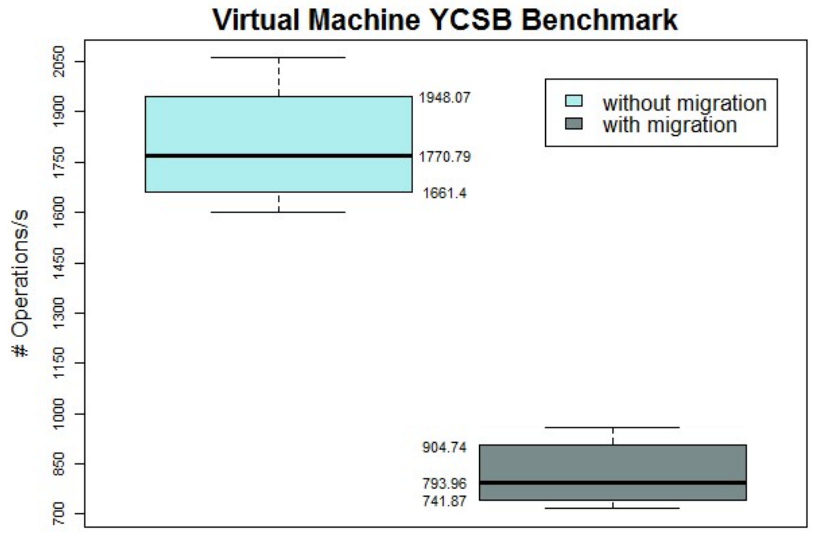


a)

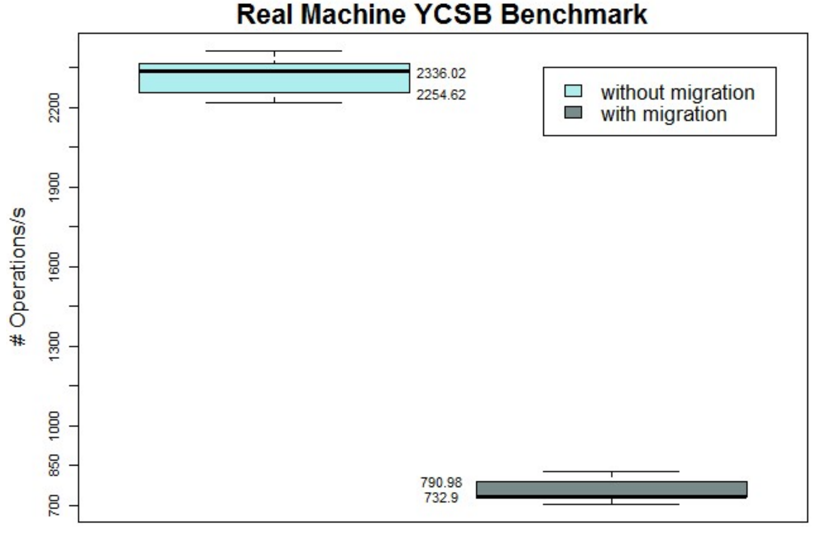


b)

Figure 7. Box plot of *ab* performance on a) VM and b) Real Machine (higher is better)



a)



b)

Figure 8. Box plot of YCSB performance on a) VM and b) Real Machine (higher is better)

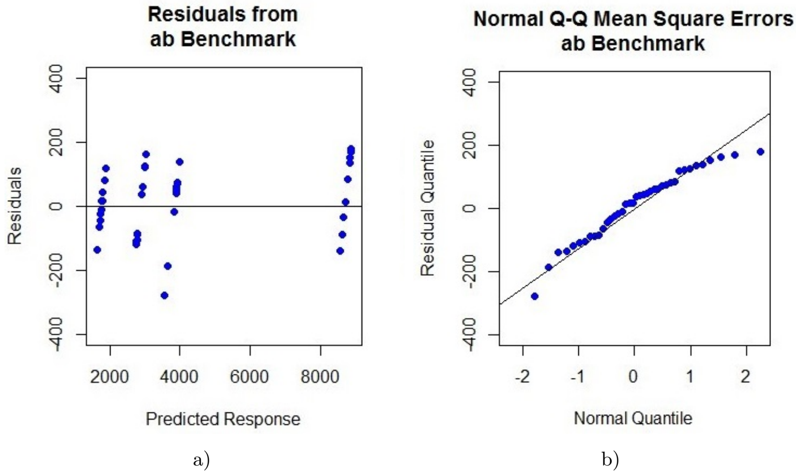


Figure 9. Plot of a) residuals versus predicted response and b) normal quantile-quantile for *ab*

the squared errors (SSE) as follows:

$$S_e = \sqrt{\frac{SSE}{2^2(r-1)}} \tag{6}$$

and the standard deviation of effects is

$$S_{q_i} = \frac{S_e}{\sqrt{2^2 r}}. \tag{7}$$

Then, the confidence intervals for the effects is

$$q_i \mp t_{\left[\frac{1-\alpha}{2};2^2(r-1)\right]} S_{q_i}. \tag{8}$$

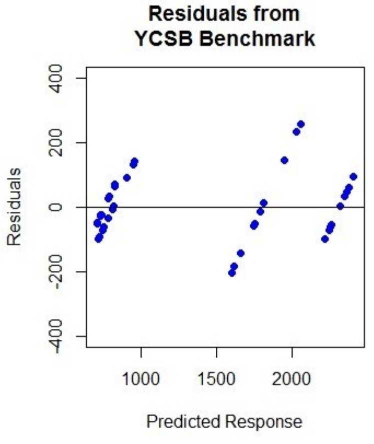
The t-value is read at $2^2(r-1)$ degrees of freedom, for our data the degrees of freedom is 36, and it was used 90 % of confidence interval. So, the t-value at 36 degrees of freedom and 90 % of confidence is 1.6883.

Table 5 has the Confidence Intervals (CI) for data from benchmarks *ab* and YCSB for q_0, q_A, q_B and q_{AB}.

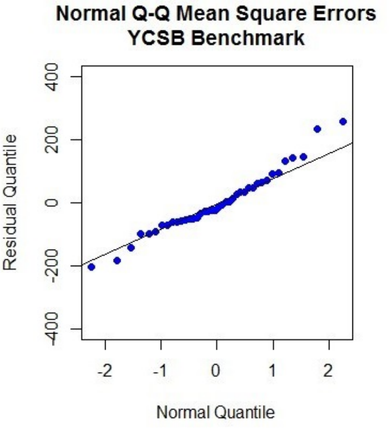
As can be seen from Table 5, none of the intervals include a zero, therefore, all the effects are significant.

E. Performance Impact with Multiple Migrations

Considering the initial evaluations, it is possible to see that the impact of the migration in virtualized environments was less representative than in real envi-



a)



b)

Figure 10. Plot of a) residuals versus predicted response and b) normal quantile-quantile for YCSB

Effects	**CI for *ab***	**CI for YCSB**
q_0	$(4\,256.769, 4\,329.785)$	$(1\,395.356, 1\,449.780)$
q_A	$(1\,455.481, 1\,528.496)$	$(609.982, 664.406)$
q_B	$(-2\,012.349, -1\,939.334)$	$(-141.194, -86.771)$
q_{AB}	$(-972.831, -899.815)$	$(-170.314, -115.890)$

Table 5. Confidence intervals for effects from *ab* and YCSB

ronments, even the virtualized services presented lower performance. Services virtualization is one of the key points in the development of many new computing paradigms during the last years.

One of them that can be highlighted is cloud computing, where thousands of services can be deployed in a shared infrastructure using Virtual Machines.

In cloud computing systems, it is usual to find the virtualized server migrating at the same time, arriving or leaving a host server. Therefore, we decided to evaluate the impact on service's performance caused when multiple migrations are made. We wanted to know, for instance, what is the Web Server performance on one VM that is migrating from a host when there is another VM arriving at the same time in the same host. Those tests were made only on the virtual environment, implemented on a server with 16 GB of memory and 500 GB of disk.

Figure 11 shows the architecture of this environment, indicating the fundamental elements for the migration process to be executed. The VM Client is responsible for generating the requests to the Apache Web server through

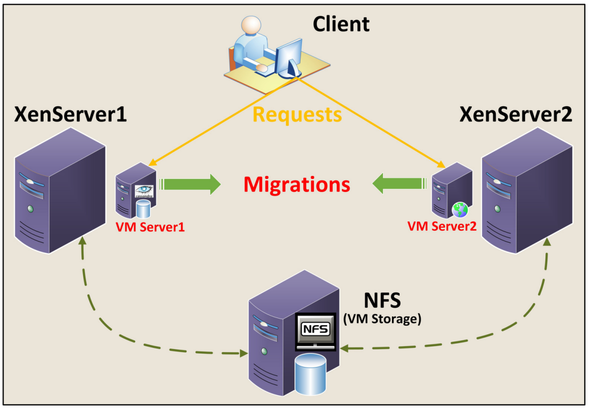


Figure 11. Environment totally with virtualized machines and with simultaneous migrations

the *ab* benchmark, as well as performing the transactions for the Cassandra database through the YCSB benchmark. It was from this machine that the benchmarks generated the triggered workload for Server1 and Server2, these two virtual servers were virtualized by Xen. The Hypervisor Xen was installed on two Xen Servers (XenServer 1 and 2) that were used for migration of the Server1 and Server2, as source and destination machines. They have the same memory, disk, and operating system settings as can be seen in Table 6.

The Server1 and Server2 VMs are hosting the Apache Web Server, as well as the DBMS Cassandra server. During the migration process, these were the servers that were migrated. The NFS was used the same way as described in previous sections.

Table 6 shows the hardware and software configurations of all VMs involved in the migration process. In all hosts, the OS used was Ubuntu 14.04.

VMs	**Memory (GB)**	**HD (GB)**
Client	2	25
XenServer1	4	40
XenServer2	4	40
Server1	2	10
Server2	2	10
Storage NFS	1	25

Table 6. Virtual machines descriptions

In the experiments, two virtual servers (Server1 and Server2) were used, one with Apache Web Server and another with Cassandra DB. Server1 was initially virtualized on XenServer1 and Server2 on XenServer2, then Server1 was migrated to XenServer2 and, at the same time, Server2 was migrated to XenServer1. Measurements were made on these virtual servers with only one migration, simultaneous migrations, and without migration. Benchmarks *ab* and YCSB used the same configurations of the initial experiment and 10 repetitions of each test were performed to guarantee statistical representativeness.

Here, we call *"with migration"* when just one VM is migrating and receiving requests from a client; and we call *"with simultaneous migration"* when the two virtual servers (Web Server and DBMS) are migrating simultaneously, and both are receiving requests from clients. The results of these simulations can be seen in Figures 12 and 13.

Analyzing the requests per second metric, Figure 12 shows that, on average, the execution of *ab* with simultaneous migration resulted in a reduction of 31.3 % when compared to the non-migration approach, and 14.6 % when compared to a single migration. In this way, it is observed that for this case the simultaneous migration was the one that resulted in a greater impact on the performance of the service.

As can be seen in Figure 13, the data collected shows the migration performance implied, on average, an approximate reduction of 61.8 % in the YCSB throughput. However, it is also possible to note that the approaches with a single and with simultaneous migrations presented approximate values, that is, a simultaneous migration execution scenario implies, in average, a reduction equivalent to the single migration approach for this service.

As a conclusion from these results, it was possible to confirm with YCSB's data that the migration process imposes a significant overhead compared to non-migration tests on the virtualized environment. In addition, the occurrence of single migration or multiple migrations apparently did not imply significant differences, which should be confirmed by statistical tests. Regarding the tests with the *ab* benchmark, it was also possible to observe the migration impact in the Web service, however, not as significantly as with the DBMS Cassandra. If we observe the events with simultaneous migrations, we can see that this situation further degrades the Web service, reaching a 31.3 % drop in performance. These results confirm the results of the initial experiment, which indicates that non-relational database service was less affected by migration process than web content service.

To confirm the results obtained in the tests with simultaneous migrations, we performed a series of statistical tests of Mann-Whitney-Wilcoxon U-test, all with a significance level of 0.05, using statistical software R. The results were:

- With respect to *ab* benchmark, comparing the data from the tests with migration and with simultaneous migration, the p-value turns out to be $1.083\mathrm{e}{-05}$, less than the 0.05 significance level, in this case, we reject the null hypothesis of statistical equality of the means of the two groups. *That is, the performance obtained by the ab benchmark with only one server migrating*

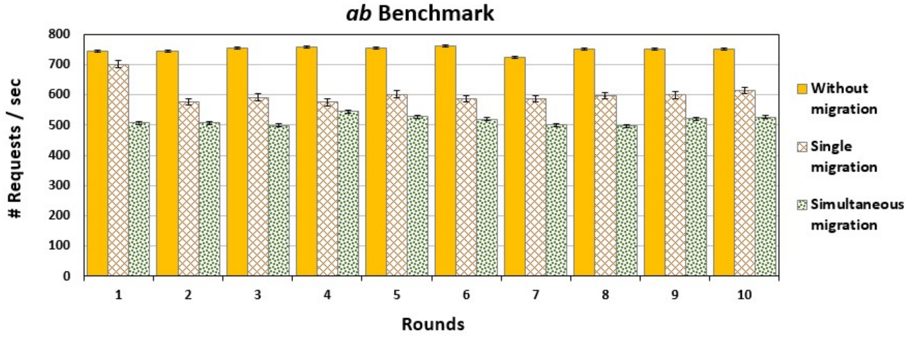


Figure 12. Performance of *ab* benchmark on virtual environment (higher is better)

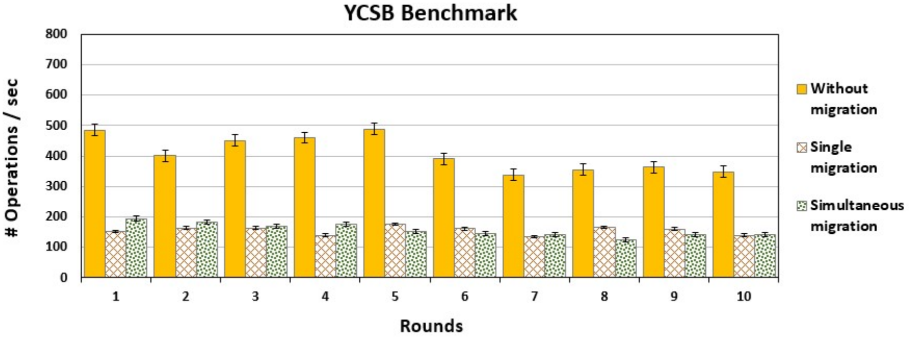
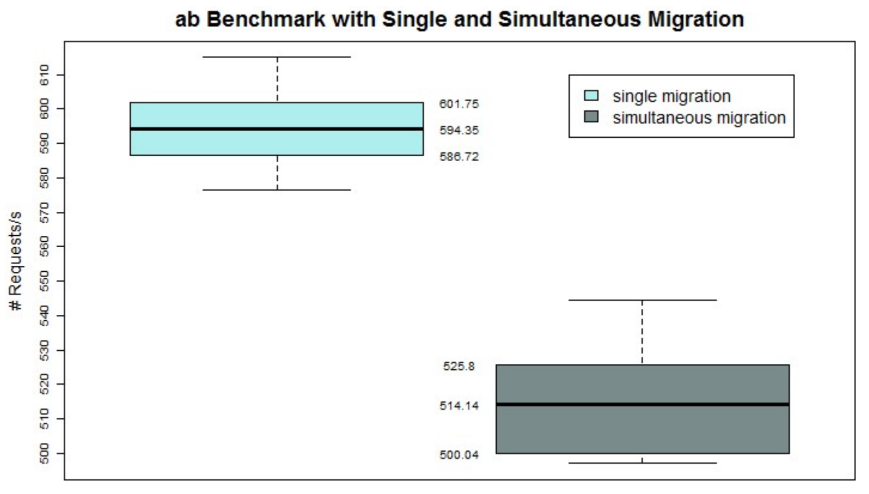


Figure 13. Performance of YCSB benchmark on virtual environment (higher is better)

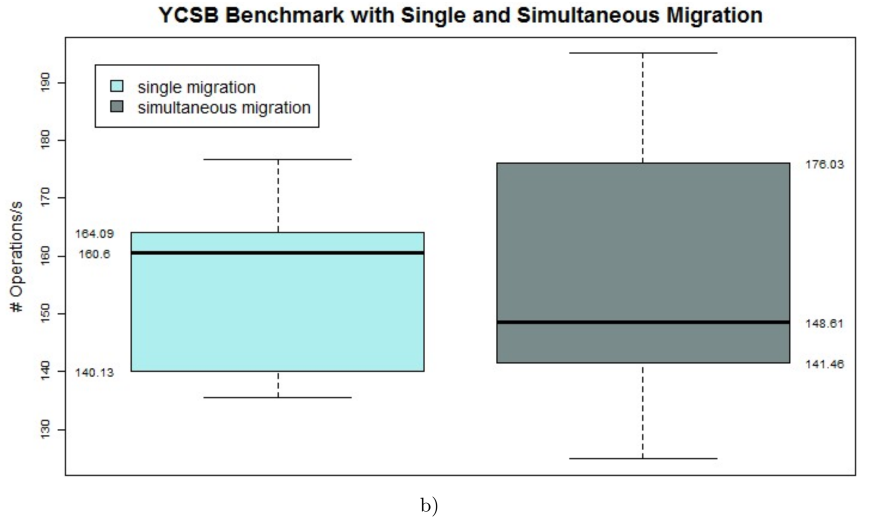
is different from *the performance obtained when simultaneous migrations were occurring.*

- The YCSB chart (Figure 13) indicated that the occurrence of single migration or multiple migrations did not imply, apparently, significant differences in the performances of the Cassandra DB. To test this hypothesis, the same statistical test was performed comparing the data of single migration and multiple migrations from YCSB benchmark. As the p-value turns out to be 0.8534, greater than the 0.05 significance level, in this case, we can accept the null hypothesis of statistical equality of the means of the two groups. *That is, it is not possible to statistically affirm, with a significance level of* 5 %*, that there are differences between the performance obtained by the YCSB benchmark when only one server is migrating and when multiple migrations are taking place.*

The results obtained with the statistical tests were confirmed with the box plot charts available in Figures 14 a) and 14 b). The difference between sim-



a)



b)

Figure 14. Box plot of the performance with single and with simultaneous migration of a) *ab* and b) YCSB (higher is better)

ple migration and multiple migrations is evident from *ab* benchmark data. Whereas, for the YCSB, it is not possible to state that there is a statistical difference.

7 CONCLUSION AND FUTURE WORK

Experiments were conducted in fully virtual and real environments migrating and not migrating the content servers. As achievements, we can highlight:

- The overhead caused by live migration process is significant, noting that on both benchmarks *ab* as YCSB, the number of operations or requests per second were significantly reduced.
- The results showed that the *ab* benchmark had superior performance when running on a fully real environment, regardless of the scenario being with or without live migration. This is because the benchmark *ab* makes much use of the network interface which eventually becomes a bottleneck for operations, since the performance of network resources with virtual machines is reduced.
- Comparing the results of the YCSB benchmark in a fully virtualized environment with real, it was observed that the fully virtualized presented lower performance without migration scenario, but exceeded the real environment when there was live migration. Here there is the intense transfer bottleneck of data between hosts since the YCSB performs operations on the host database server. Therefore, the performance without live migration was higher in the real machine. In the case of migration, we have the Ethernet bridge factor that, in the virtual scenario, favored runtime since all hosts and the bridge were on the same real machine.
- When multiple migrations were conducted in the environment two different behaviors could be observed according to the service used. When web service was in place there was a significant reduction in performance. However, running Cassandra no additional reduction was observed.

In summary, we have identified that the overhead caused by the virtual machine live migration process observed from the client's point of view is very impactful, since performance was degraded in all results of the benchmark execution on real environment when compared to virtual environment.

As future work, we are going to plan to deploy different services to check the impact perceived in each one, to extend the experiments with multiple migrations and to use a heterogeneous environment where we would find virtual and real servers used as hosts for migration process. It is also intended to include KVM as a hypervisor, as well as measuring the impact of network infrastructure on results.

REFERENCES

[1] Ramachandran, M.—Chang, V.: Towards Performance Evaluation of Cloud Service Providers for Cloud Data Security. International Journal of Information Management, Vol. 36, 2016, No. 4, pp. 618–625, doi: 10.1016/j.ijinfomgt.2016.03.005.

[2] Kim, A.—Lee, J.—Kim, M.: Resource Management Model Based on Cloud Computing Environment. International Journal of Distributed Sensor Networks, Vol. 12, 2016, doi: 10.1177/1550147716676554.

[3] Chen, W.—Shang, Z.—Tian, X.—Li, H.: Dynamic Server Cluster Load Balancing in Virtualization Environment with OpenFlow. International Journal of Distributed Sensor Networks, Vol. 11, 2015, No. 7, doi: 10.1155/2015/531538.

[4] Xing, G.—Xu, X.—Xiang, H.—Xue, S.—Ji, S.—Yang, J.: Fair Energy-Efficient Virtual Machine Scheduling for Internet of Things Applications in Cloud Environment. International Journal of Distributed Sensor Networks, Vol. 13, 2017, doi: 10.1177/1550147717694890.

[5] Zoraja, I.—Trlin, G.—Sunderam, V.: Eliciting the End-to-End Behavior of SOA Applications in Clouds. Computing and Informatics, Vol. 35, 2016, No. 2, pp. 259–281.

[6] Marshall, D.: Understanding Full Virtualization, Paravirtualization, and Hardware Assist. VMware White Paper, 2007.

[7] Magalhães, D. V.—Soares, J. M.—Gomes, D. G.: Análise do Impacto de Migração de Máquinas Virtuais em Ambiente Computacional Virtualizado. XXIX SBRC, Mato Grosso do Sul, Brazil, 2011, pp. 235–248.

[8] Bobák, M.—Hluchý, L.—Tran, V. D.: Application Performance Optimization in Multicloud Environment. Computing and Informatics, Vol. 35, 2016, No. 6, pp. 1359–1385.

[9] Padala, P.—Zhu, X.—Wang, Z.—Singhal, S.—Shin, K. G.: Performance Evaluation of Virtualization Technologies for Server Consolidation. HP Labs Technical Report HPL-2007-59R1, April 11, 2007.

[10] Li, J.—Wang, Q.—Jayasinghe, D.—Park, J.—Zhu, T.—Pu, C.: Performance Overhead among Three Hypervisors: An Experimental Study Using Hadoop Benchmarks. IEEE International Congress on Big Data, Santa Clara, USA, 2013, pp. 9–16, doi: 10.1109/BigData.Congress.2013.11.

[11] Sahoo, J.—Mohapatra, S.—Lath, R.: Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues. 2010 Second International Conference on Computer and Network Technology (ICCNT), IEEE, Bangkok, Thailand, 2010, pp. 222–226, doi: 10.1109/ICCNT.2010.49.

[12] XenSource: A Performance Comparison of Commercial Hypervisors. XenEnterprise vs. ESX Benchmark Results, XenSource, 2007.

[13] VMware: A Performance Comparison of Hypervisors. VMware White Paper. `https://www.vmware.com/pdf/hypervisor_performance.pdf`, 2007, accessed 10 October 2016.

[14] Reddy, P. V. V.—Rajamani, L.: Performance Evaluation of Hypervisors in the Private Cloud Based on System Information Using SIGAR Framework and for System

Workloads Using Passmark. International Journal of Advanced Science and Technology, Vol. 70, 2014, pp. 17–32.

[15] Martins, G.—Bezerra, P.—Gomes, R.—Albuquerque, F.—Costa, A.: Evaluating Performance Degradation in NoSQL Databases Generated by Virtualization. 2015 Latin American Network Operations and Management Symposium (LANOMS), João Pessoa, Brazil, 2015, pp. 84–91, doi: 10.1109/LANOMS.2015.7332675.

[16] Galloway, M.—Loewen, G.—Vrbsky, S.: Performance Metrics of Virtual Machine Live Migration. 2015 IEEE 8^{th} International Conference on Cloud Computing, New York, USA, 2015, pp. 637–644, doi: 10.1109/CLOUD.2015.90.

[17] Isci, C.—Liu, J.—Abali, B.—Kephart, J. O.—Kouloheris, J.: Improving Server Utilization Using Fast Virtual Machine Migration. IBM Journal of Research and Development, Vol. 55, 2011, No. 6, 12 pp., doi: 10.1147/JRD.2011.2167775.

[18] Zhang, F.—Chen, J.—Chen, H.—Zang, B.: CloudVisor: Retrofitting Protection of Virtual Machines in Multi-Tenant Cloud with Nested Virtualization. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (SOSP '11), Cascais, Portugal, 2011, pp. 203–216, doi: 10.1145/2043556.2043576.

[19] Wood, T.—Shenoy, P.—Venkataramani, A. et al.: Sandpiper: Black-Box and Gray-Box Resource Management for Virtual Machines. Computer Networks, Vol. 53, 2009, No. 17, pp. 2923–2938, doi: 10.1016/j.comnet.2009.04.014.

[20] Jain, R.: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. 1^{st} edition. John Wiley and Sons, 1990.

[21] Hollander, M.—Wolfe, D. A.—Chicken, E.: Nonparametric Statistical Methods. 3^{rd} edition. John Wiley and Sons, 2013.

Petrônio Carlos Bezerra graduated in computer science at the Federal University of Paraíba, Campina Grande, Brazil in 1999. Currently, he is Professor at the Federal Institute of Education, Science and Technology of Paraiba, Campina Grande, Brazil. He is a doctoral student at Federal University of Campina Grande. His current research interests include computer networks, software-defined networks and virtualization.

Marcela Tassyany Galdino Santos is the undergraduate student at the Federal Institute of Education, Science and Technology of Paraiba, Campina Grande, Brazil. Her current research interests include computer networks, software-defined networks, virtualization, cloud computing and optical communications.

Edlane de Oliveira G. Alves is the undergraduate student in telematics at the Federal Institute of Education, Science and Technology of Paraíba, Campina Grande Campus. She has participated in research focusing on optical communications, virtualization, technology and production with an emphasis on open source academic control systems. She is interested in cloud computing, computer networks and currently participating in projects in the area of software-defined networks, virtualization and digital inclusion with an emphasis on education.

Fellype Albuquerque graduated in computer science at the Federal University of Campina Grande, Campina Grande, Brazil with the emphasis on computer networks and cloud computing. Currently he is developing projects in the area of cloud computing in the Distributed Systems Laboratory.

Gustavo Nóbrega Martins received his Master degree in computer science from the Federal University of Campina Grande and graduation degree in computer science from the State University of Paraíba. Currently, he is the doctoral student at the Federal University of Campina Grande, Campina Grande, Brazil. His research interests include wireless sensor network, software defined radio, computer networks and virtualization technology.

Reinaldo Gomes has been Assistant Professor of the Computing and Systems Department of UFCG since 2010. Degree in telematics technology from the Federal Institute of Education, Science and Technology of Paraiba in 2004, Master in computer science from the Federal University of Pernambuco in 2005 and Ph.D. in computer science from the Federal University of Pernambuco in 2010. Since 2004 he has been actively involved in research and development projects with national and international cooperation, concerning development of communication protocols, traffic evaluation, wireless communication technologies and advanced applications for future internet.

Anderson Fabiano B. F. da Costa graduated in telematics at the Federal Institute of Education, Science and Technology in 2004, he received his Master and Doctor degrees from the Federal University of Pernambuco in 2005, and in 2011. Currently, he is Professor at the Federal Institute of Education, Science and Technology of Paraiba, Campina Grande, Brazil. He has experience in computer science, with emphasis on computer networks, acting in the following themes: software-defined networks, virtualization, wireless networks and cluster analysis.