Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 26, Num. 1 (2019) 78-90

RESEARCH ARTICLE

A Strategy for Performance Evaluation and Modeling of Cloud Computing Services

Uma Estratégia para Avaliação de Desempenho e Modelagem de Serviços de Computação em Nuvem

Rajeev R. Yadav[1]*,Gleidson A. S. Campos [1], Erica T. G. Sousa[1], Fernando A. A. Lins[1]

Abstract: On-demand services and reduced costs made cloud computing a popular mechanism to provide scalable resources according to the user's expectations. This paradigm is an important role in business and academic organizations, supporting applications and services deployed based on virtual machines and containers, two different technologies for virtualization. Cloud environments can support workloads generated by several numbers of users, that request the cloud environment to execute transactions and its performance should be evaluated and estimated in order to achieve clients satisfactions when cloud services are offered. This work proposes a performance evaluation strategy composed of a performance model and a methodology for evaluating the performance of services configured in virtual machines and containers in cloud infrastructures. The performance model for the evaluation of virtual machines and containers in cloud infrastructures is based on stochastic Petri nets. A case study in a real public cloud is presented to illustrate the feasibility of the performance evaluation strategy. The case study experiments were performed with virtual machines and containers supporting workloads related to social networks transactions.
Keywords: Performance evaluation — Cloud Computing — Containers — Virtual machines — Performance model

Resumo: Os serviços sob demanda e os custos reduzidos tornaram a computação em nuvem um mecanismo popular para fornecer recursos escalonáveis de acordo com as expectativas do usuário. Esse paradigma é um papel importante nas organizações acadêmicas e de negócios, suportando aplicativos e serviços implantados com base em máquinas virtuais e contêineres, duas tecnologias diferentes para virtualização. Ambientes de nuvem podem suportar cargas de trabalho geradas por vários números de usuários, que solicitam o ambiente de nuvem para executar transações em que seu desempenho deve ser avaliado e estimado para atingir as satisfações dos clientes quando os serviços de nuvem são oferecidos. Este trabalho propõe uma estratégia de avaliação de desempenho composta por um modelo de desempenho e uma metodologia para avaliar o desempenho dos serviços configurados em máquinas virtuais e contêineres implantados em infraestruturas em nuvem. O modelo de desempenho para a avaliação de máquinas virtuais e contêineres em infraestruturas de nuvem é baseado em redes de Petri estocásticas. Um estudo de caso em uma nuvem pública real é apresentado para ilustrar a viabilidade da estratégia de avaliação de desempenho. Os experimentos do estudo de caso foram realizados com máquinas virtuais e contêineres suportando as cargas de trabalho.
Palavras-Chave: Desempenho de Nuvem — *Containers* — Máquinas virtuais — Modelo de desempenho

[1] *Department of Computing, Federal Rural University of Pernambuco, Brazil*
***Corresponding author**: rajeev.yadav@ufrpe.br
DOI: https://doi.org/10.22456/2175-2745.87511 • **Received:** 17/10/2018 • **Accepted:** 20/01/2019

1. Introduction

Cloud computing is depicted as a distributed mechanism in which the resources are paid based on utilization models, as presented by Erl[1]. In cloud environments, users can focus only on their activities, leaving the environment management to the cloud provider. Its flexibility allows public, private, and hybrid clouds[2], designated to different profiles of users.

In this paper, the focus is on the environment-as-a-Service. In this approach, a pool of resources supports virtual machines and containers deployed in cloud environments. Clouds allied with IaaS provides scalable and reliable services for these virtualization technologies[3] and in this scenario, its performance should be evaluated regarding metrics that can influence the perceived quality of clients requesting the cloud environment to deploy and execute services and applications. These cloud applications can be represented by stochastic Petri nets in order to evaluate metrics such as throughput, which is an important quality metric of cloud services. Evaluating and modeling throughput of cloud applications improve the performance seen by users[4].

When large amount of users request the cloud environment to execute their applications, the cloud resources can turn more constrained and result in loss of scalability and hence, users satisfaction reductions. In this context, alternatives to the traditional virtual machines such as container solution are gaining widespread attention in the academic and professional community[5]. The comparison between these two technologies contributes to measure the quality of services in cloud environments and determine which one is recommended for each situation and considering also, a mix of these technologies to compose applications in cloud environments.

For decades, hypervisors of virtual machines are responsible to deploy services and manage the layer between the physical resources and the client's layer[6]. Containers emerged presenting better performance compared to virtual machines performance[7] considering metrics such as reponse times and execution times. It does not need the hypervisor layer to deploy services and applications in cloud environments, resulting also in less overhead. Containers own all of the libraries and system archives needed to deploy and execute services. This aspect ensures interoperability of the virtualization with containerized applications without the necessity of creating a whole operating system. Relevant studies and works evaluated the performance of cloud environments, considering virtual machines and containers, but there is no presentation of a strategy composed of methodology and performance model to accomplish this.

Run containers on top of virtual machines are commonly used in the cloud computing paradigm and avoid hardware and software incompatibility between virtualization and the physical resources of host nodes in the cloud. However, if the service needs to be as fast as possible, the containers in the bare metal may be a better option. These scenarios propose the importance of evaluating containers and virtual machines in cloud environments considering different attributes, such as quality and performance metrics[7].

Based on the literature, Alkhanak et al.[8] concluded that the emerging of cloud computing incurred in cost-aware and performance challenges related to the Quality of Services (QoS), functionality requirements, and system's architectures. The management of cloud environments and its service-driven aspects are decisive to achieve quality goals. But on the other hand, this study also demonstrated that researchers have paid less attention to the performance metrics compared to its availability and reliability in the last years. Performance evaluation of cloud environments can help system administrators to make decisions regarding which virtualization technology provides the expected and desired performance when several requests are submitted to the cloud environment.

This work proposes a strategy composed of a methodology and an analytical model for evaluating the performance of virtual machines and containers in cloud environments. The performance model is based on stochastic Petri nets and can be adopted to model various scenarios, considering larger workload intensities. The performance model depicts users of cloud service submitting workloads to the virtual machines and containers deployed in the cloud environment. This performance modeling of cloud environment considering measurements performed in a real scenario.

The methodology provides the necessary activities and its logical sequence to evaluate cloud environments regarding virtual machines and containers. Practitioners and researchers can adopt this methodology to evaluate performance and estimate it under different scenarios.

This paper is organized as follows. Section 2 presents related work regarding the performance and cost evaluation of virtual machines and containers. Section 3 brings background concepts and relevant terminologies. The methodology to evaluate the performance of cloud environments based on virtual machines and containers is presented in Section 4. Section 5 presents the performance model based on stochastic Petri nets. Section 6 presents a case study to demonstrate the feasibility of the methodology executed in a real cloud environment. Finally, Section 7 concludes this paper and propose future work.

2. Related Work

Recently, some works evaluated the performance of cloud environments, with services evaluation composed of virtual machines and containers.

Shirinbab et al.[9] addressed the problem of which virtualization technology to adopt for workloads generated by Cassandra tool. This work executed experiments in order to compare the performance of virtual machines and Docker containers in a cloud environment. Metrics such as processor utilization, disk and write rate and latency were measured. The containerized solution resulted in lower resources consumptions and presented better performance, considering the metrics evaluated. A combination of these virtualization technologies is recommended by the authors to achieve lower resources utilization and hence, reduce costs with energy consumption for instance. This work does not present a methodology demonstrating how the authors performed the performance evaluation.

Sekar et al.[10] evaluated the performance of containerized applications considering two cloud environments: Amazon EC2 e Joyent. This performance comparison considered the JMeter in order to simulate users traffic for popular applications such as Wordpress deployed in these two platforms. With the same script to generate the workloads, this work builds and deployed Docker images based on these applications and submitted workloads generated by Sysbench benchmark tool. This work concluded that Joyent platform performed 150%

better related to the processor utilization of the containers while the workloads attend the requests 3x better than the Amazon public cloud platform.

Barik et al.[11] evaluated the performance of virtual machines and containers in a cloud environment. The experiments were conducted in Ubuntu 14.04 operating system and benchmarks such as Apache Benchmark and 7-zip Benchmark were adopted to generate workloads. To accomplish the performance comparison, execution times and bootup times were measured comparing the virtual machine and the container deployed. The size of the test files determined the workload intensity for each resource evaluated and at the end of the experiments, containers presented better performance than the virtual machines. However, the containers presented trends to generate more overheads under high workloads. This work presented a performance evaluation based on measurements and presenting its results without further explanations of the reasons behind it.

Sysbench is a benchmark with customizable commands and workloads that evaluates the performance of computer systems, such as processor, memory and storage disk. In [12], this benchmark was used to compare performance between virtual machines and containers considering transactions and its metrics. The workload intensities were represented by the number of concurrent threads, in which containers presented a better performance analyzing the throughput metric regarding these transactions. Furthermore, containers provided a performance near the native performance (without any virtualization) in a stationary analysis, where the metric has a constant behavior no matter how much intense is the workload submitted to the containers. In this work, a graph is adopted to illustrate the measurements of the performance evaluation and does not explain why the containers achieved better throughput compared to the virtual machines. The authors did not demonstrate how they executed the experiments and the activities necessary to accomplish it.

Meredith & Urgaonkar[13] presented a performance model based on linear regression. In order to estimate the response time as a function of processor and memory capacity considering databases servers. In a public cloud platform, the authors performed experiments and concluded that the service offerings of the cloud platform have a significant impact on the latency of services deployed based on virtualization technologies. The throughput also presented impact when different service offerings of the cloud platform are contemplated. Although, this work presented a very specific situation in which is difficult to replicate this performance evaluation considering different scenarios and realities of practitioners and researchers over the community.

In a performance comparison of virtual machines and containers, Joy[14] demonstrated that containers present a better performance due to its high scalability and optimized resources utilization in cloud environments. The scalability was measured by the time taken to scale and process the request in a Wordpress application and measured by JMeter. This study presented some benefits of container-based virtualization such as high availability and fast deployments of services and applications based in containers. The author did not depict the activities needed to perform the evaluation and hence, cannot be replicated by practitioners and researchers.

This way, the interest of the scientific community in containers configured in cloud computing has been widely discussed as an alternative to virtual machines. These studies depicted that containers generally presents a better performance compared to virtual machines considering performance metrics.

Table 1 depicts the comparison between related work and this study, in which: Meth. = Methodology and PM = Performance Model. This table depicts the related work and the authors are identified in column 1. Related work did not present methodologies to evaluate the performance of virtualization technologies while only [13] presented a performance model to estimate performance metrics considering containers and virtual machines in cloud computing environments.

Table 1. Related work comparison

Work	Meth.	PM
Shirinbab et al.[9]	No	No
Sekar et al.[10]	No	No
Barik et al.[11]	No	No
Felter et al.[12]	No	No
Meredith & Urgaonkar[13]	No	Yes
Joy[14]	No	No
This paper	Yes	Yes

3. Background

This section describes the concepts and terminologies necessaries to a better understanding of this paper.

3.1 Containers and Virtual Machines in Cloud Computing

According to the NIST[2], clouds are managed and operated by different organizations such as academic, business, and government entities. Services and applications in cloud computing are mainly deployed based on different virtualizations technologies, including hypervisor-based virtual machines and containers, each one with its own benefits and weaknesses[7][15].

Cloud providers are deploying containers to offer services more than ever. Google Cloud Platform deploys billion of containers each week allowing mechanisms of deployment, scaling, and management of containerized applications offered to the general users. Microsoft Azure provides a solution to manage and deploy Kubernetes clusters with several numbers of containers being orchestrated. IBM Cloud Container Service is also providing a managed Kubernetes clusters as a service on its platform. In clouds, virtual machines are deployed without further concerns, but it needs a complete operating system to run services and applications.

Figure 1 depicts how the containers are layered. There is no hypervisor layer associated to the virtualization and the container engine is connected directly in the host operating system, in which the applications are self-packaged and ready-to-deploy, avoiding issues related to incompatibilities and so on.

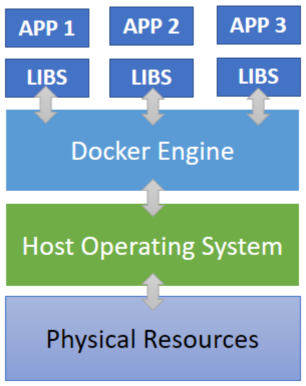

Figure 1. Layers in containers(based in Barik et al.[11])

On the other hand, Figure 2 shows the layers associated with the virtualization considering hypervisor-based virtual machines.

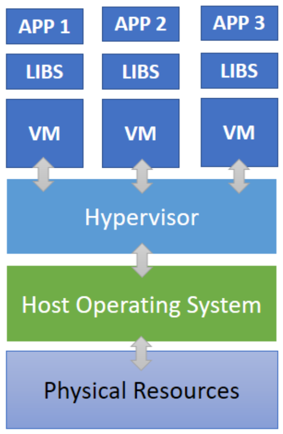

Figure 2. Layers in virtual machines(based in Barik et al.[11])

Pahl et al.[7] defines containers as a lightweight technology way to virtualize applications in cloud environments. Container owns all the files to execute complete services, including libraries and system archives. It makes this technology a self-packaged and ready-to-deploy system, acting as an alternative to virtual machines, mainly in cloud computing environments. The main benefits of containers are high software availability and rapid deployments, better productivity with low bootup times[15].

3.2 Stochastic Petri Nets

As mentioned in the introduction, stochastic Petri nets have been adopted to evaluate the performance of cloud applications and model it, providing important metrics such as throughput, which has a significant impact on the quality of cloud services perceived by clients.

Petri net is a mathematical formalism allied to graphical resources. This technique deals with different aspects of the relationship between components in a system including concurrency, synchronization and communication mechanisms. Places, represented by circles, denote local states which the evaluated system can present during its different stages of the lifecycle. Petri nets naturally represent probabilistic and deterministic models and are depicted by transitions (presented as rectangles)[3].

In order to illustrate the performance technique adopted in this paper, Figure 3 depicts a stochastic Petri net with three places, two temporized transitions, and one immediate transition. The place Client depicts users of the cloud sending requests with a mean rate of λ, represented by an exponential probability distribution(transition Sending_time. Once the request is submitted, the system enters in the state Request_submitted and requests the cloud environment through the transition Requesting_cloud_environment. The Cloud environment place represents the cloud provider, which offers the environment to support clients requests. Once a request is received, the cloud environment responds to requests, after a mean rate of μ units of time(transition Processing_time). Then the cloud environment is idle again to receive new requests from its users. Once the cloud environment processed the requests, the system enters in the state Client once again.

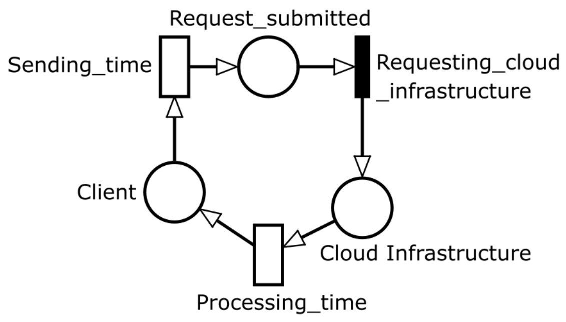

Figure 3. Stochastic Petri net model

SPNs can be composed of temporized and immediate transitions. Temporized transitions are associated with timed aspects of systems such as a machine executing a task at a certain rate of time. There is a delay time associated with the transition. On the other hand, immediate transitions consider that there is no time associated in its mechanism, where the activity passes from a state to another state immediately[3][16].

Phase approximation is a technique that adjusts SPNs in order to represent non-exponential distributions in the performance model. Based on the value of the coefficient of variance, the SPNs evaluated can have part of its composition turned into Erlang, Hyperexponential, and Hypoexponential distribution. To identify which distribution best represent the evaluated metrics, the inverse of the coefficient of variation($1/C_V$) is used as follows[17](Equation 1).

$$\frac{1}{C_v} = \frac{\mu_d}{\sigma_d} \tag{1}$$

Three possible situations regarding C_V can be adopted[17]: When C_V is a whole number and different from one, the empirical data is approximated as a Erlang distribution where the new length (γ) is calculated by Equation 2.

$$\gamma = (\frac{\mu}{\sigma})^2 \tag{2}$$

The new rate of the transition is calculated by Equation 3:

$$\lambda = \frac{\gamma}{\mu} \tag{3}$$

Figure 4 depicts an Erlang distribution adjusting a non-exponential transition into a stochastic petri net.

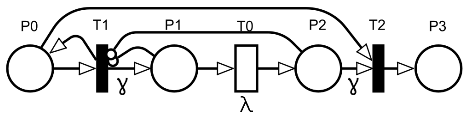

Figure 4. Erlang Distribution Net

When C_V is a number larger than one but not an integer, the empirical data should be characterized as Hypoexponential distribution, where a SPN composed of a sequence whose length (γ) is estimated as (Equation 4):

$$(\frac{\mu_1}{\sigma})^2 \leq \gamma < (\frac{\mu_2}{\sigma})^2 \tag{4}$$

And the new transition rates are calculated by Equation 5 and 6:

$$\lambda_1 = (\frac{1}{\mu_1}) \tag{5}$$

$$\lambda_2 = (\frac{1}{\mu_2}) \tag{6}$$

The average expected time, μ_1 and μ_2, are calculated by the Equations 7 and 8:

$$\mu_1 = \mu \pm \frac{\sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1} \tag{7}$$

$$\mu_2 = \gamma\mu \pm \frac{\sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1} \tag{8}$$

Figure 5 represent the adjustment of the stochastic petri net by an hypoexponential distribution.

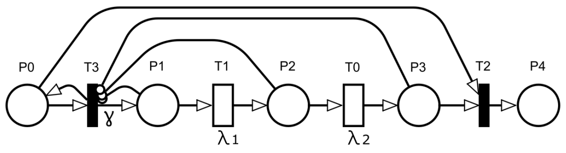

Figure 5. Hypoexponential Distribution Net

When C_V is a number smaller than one, the distribution should assume the Hyperexponential distribution

Similarly to erlang and hypoexponential distribution net, the net transition rate should be calculated by Equation 9 while its respective weights(ω_1 and ω_2) of immediate transitions should be calculated by Equations 10 and 11:

$$\lambda = \frac{2\mu^2}{\mu^2 + \sigma^2} \tag{9}$$

$$\omega_1 = \frac{2\mu^2}{\mu^2 + \sigma^2} \tag{10}$$

$$\omega_2 = 1 - \omega_1 \tag{11}$$

Figure 6 represent the hyperexponential distribution net after adjustment.

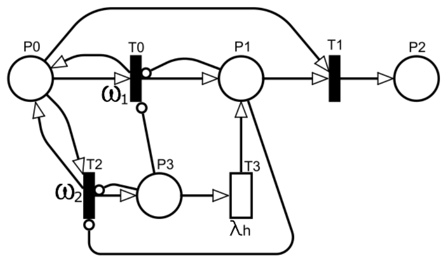

Figure 6. Hyperexponential Distribution Net

4. Methodology for the Performance Evaluation of Virtual Machines and Containers in Cloud Environments

In order to evaluate containers and virtual machines deployed in cloud environment, a sequence of logical activities must be planned to achieve the goals of the performance evaluation. Practitioners and researchers can adopt the methodology proposed in this paper to evaluate several scenarios, such as private, public, and hybrid clouds. Following this methodology system administrators and researchers are able to:

- Understand the cloud environment that is under evaluation
- Understand which performance metrics are important to their organizations

- Evaluate the performance of the cloud environment based on the results provided by the measurements considering these metrics
- Generate and validate a performance model to estimate metrics in further scenarios
- Analyze scenarios not covered in the measurements, in which robust scenarios can be difficult to set up in a real scenario

The methodology proposed to evaluate the performance of virtual machines and containers in cloud environments is presented in Figure 8.

- **Cloud Environment Definition and Configuration**: In this activity, the practitioner must understand how virtual machines and containers can be deployed in cloud environments and also select the cloud platform to be adopted in the experiments. All the settings of the virtual machine and the container should be configured in this activity and dimensioned according to the experiments complexity. Service offerings of cloud environments represent an important factor that can impact the performance of cloud services. Once defined the cloud platform and the capacities of the virtual machine and the container deployed in the cloud environment, the practitioner can deploy them. According to Barik et al.[11], there are many ways to deploy virtual machines and it is widely explained in the academic community. On the other hand, containers can be easily deployed pulling the container image of the application in a repository and executing it through the Docker engine, for instance. After its executions, the container is destroyed and the resources are released again to receive further workloads.
- **Cloud Environment Performance Metrics Selection**: The practitioner must decide which metrics represents and impacts the performance of service deployed in cloud environments, such as response time(time between the moment that the client sends a request and the moment which the client receives a response, completing the request cycle) and throughput(number of requests completed per unit of time)[4][18]. These metrics are related to the quality of services in cloud environments and its monitoring and analysis enable system administrators to check levels of quality that are offered to clients.
- **Workload Generation**: Once the cloud environment is defined and configured, the practitioner is able to plan which type of workload will be submitted to the cloud environment. Benchmarks are models constructed out of special-purpose programs and descriptive parameters, where a suite of applications are specially written to experiment particular aspect of a computational system[4][18]. Each parameter denotes an aspect of the system such as the number of clients requests and hardware capacity. The workload generated by the benchmark can be varied in order to evaluate the behavior of the system under different workload intensities. For instance, the number of clients in a system can be adopted in different intensities in order to observe the cloud service performance. In this activity, benchmark applications are executed in the virtual machine and the container solution, with the same parameters and environments to provide the performance evaluation of these two technologies for virtualization.
- **Performance Measurements and Statistical Analysis**: The practitioner needs to collect the metrics and analyze its statistics in order to evaluate the performance and further, modeling of the cloud service based on measurements of the cloud environment under evaluation. Execution times, response times, and latency play significant roles regarding clients satisfactions in cloud-based services and if there are no monitoring and analysis of these metrics, the cloud service can present degraded performance, incurring in unsatisfied users and hence, profit loss. Measurement interval also must be defined in this activity, observing the amount of time demanded by each application and hence, defining the time interval that each sample is collected (also known as sample interval). Statistics such as Averages and standard deviations are measured and metrics are collected to depict the behavior of the system under different workload intensities. However, the experiments are subjected to minor errors. The experiments can be subjected to errors such as incomplete isolation of the environment to be evaluated and restarting virtual machines and containers after each experiment. In order to avoid these errors, outliers must be removed from the data series containing the performance metrics[13].
- **Cloud Environment Performance Model**: Once the performance metrics and its statistics are collected, the practitioner can generate a model in order to represent the client and the cloud service. This activity consists of generating a model that represents the system components and its relationships. The practitioner depicts key elements of the cloud service in which users submit requests to a cloud environment which attend these requests in a certain amount of time. As the Linkbench social network benchmark submit requests based on an exponential probability distribution and with a high rate(several requests per second), the transition sending_time is inputted in the performance model with a large value for its delay time and therefore, representing the high rate of requests being submitted through an exponential distribution to the cloud environment.
- **Properties Analysis of the Cloud Environment Performance Model**: The performance model generated in the previous step, can present qualitative problems,

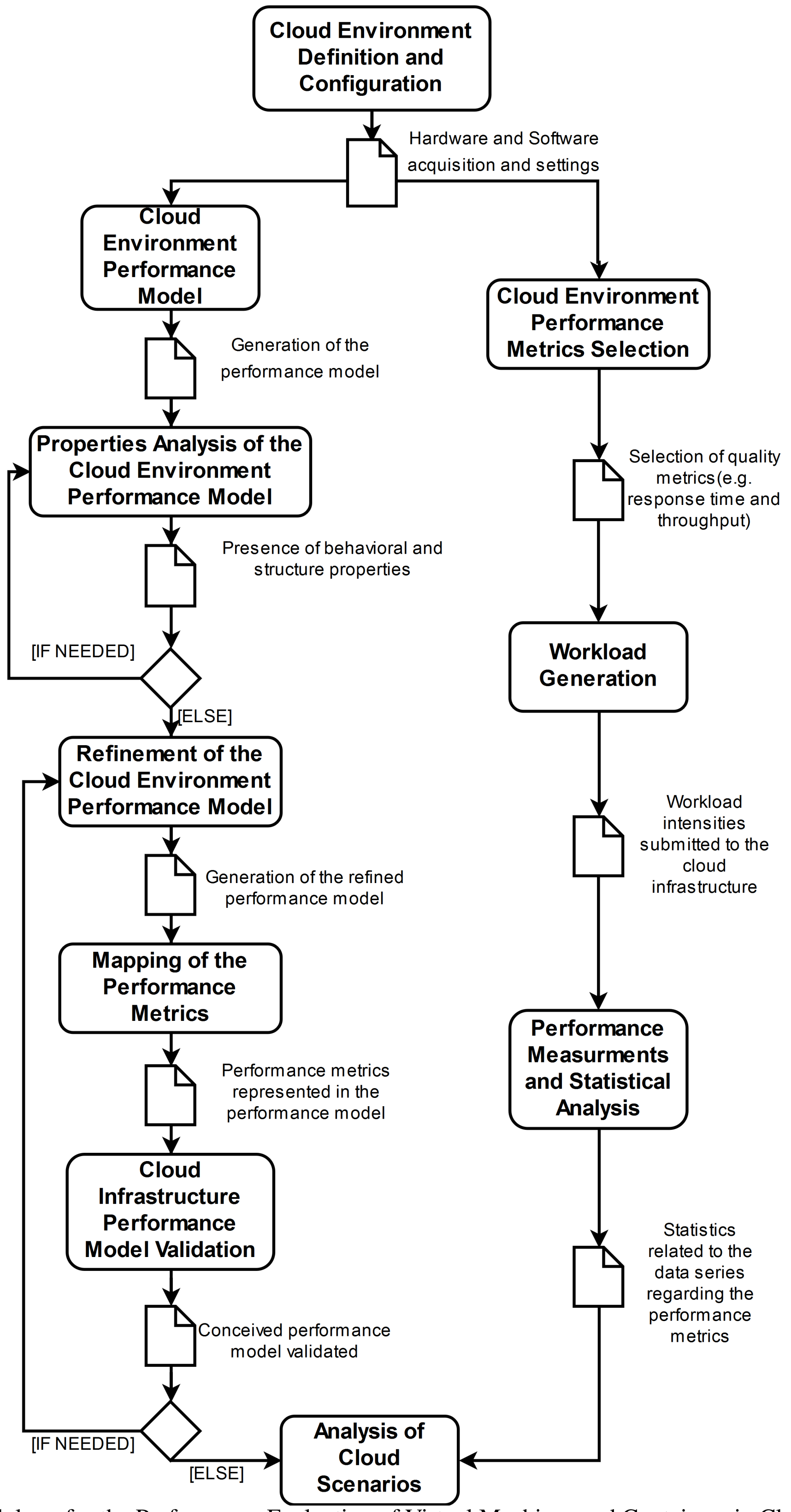
Cloud Environment Definition and Configuration
Hardware and Software acquisition and settings
Cloud Environment Performance Model
Generation of the performance model
Properties Analysis of the Cloud Environment Performance Model
Presence of behavioral and structure properties
[IF NEEDED]
[ELSE]
Refinement of the Cloud Environment Performance Model
Generation of the refined performance model
Mapping of the Performance Metrics
Performance metrics represented in the performance model
Cloud Infrastructure Performance Model Validation
Conceived performance model validated
[IF NEEDED]
[ELSE]
Analysis of Cloud Scenarios
Cloud Environment Performance Metrics Selection
Selection of quality metrics(e.g. response time and throughput)
Workload Generation
Workload intensities submitted to the cloud infrastructure
Performance Measurments and Statistical Analysis
Statistics related to the data series regarding the performance metrics

Figure 7. Methodology for the Performance Evaluation of Virtual Machines and Containers in Cloud Environments

such as deadlock and boundaries, and it must be investigated. This activity represents the qualitative analysis of the performance model[17], evaluating if exists deadlocks, boundaries, and other issues in the performance model. The HiPS tool [19] evaluates if the refined performance model is deadlock free, for example. According to Murata[19], the main properties considered are reachability, where it is verified if all the states can be achieved from another state, and liveness, that verify the presence of deadlocks in the model.

- **Refinement of the Cloud Environment Performance Model**: Stochastic Petri nets models generally exponential transitions. However, some transitions are not exponentially characterized according to the measurements. The practitioner should adapt this non-exponential transition. This activity provides the refinement of the performance model considering the statistics obtained in the measurements. These statistics suggest the polynomial-exponential distribution that best fits the empirical distribution (collected data). This adaptation is performed by the phase approximation technique [17], which computes the first and second moments of the empirical distribution, the average (μD) and standard deviation (σD), and associates to the first and second moments of the s-transition in the performance model.

- **Mapping of the Performance Metrics**: When the practitioner identified the poly-exponential distribution that best approximates the non-exponential transition which represents the processing time of the cloud service, the next activity consists of translating the performance metrics into mathematical equations in the refined performance model. In case of SPNs, quality metrics such as throughput(requests attended per second) and response time(time to attend one single request) are well represented by formulas consisting of elements of the SPNs. For instance, the number of tokens and the times associated with the transitions composes these formulas in order to obtain metrics in stationary analysis.

- **Cloud Environment Performance Model Validation**: The practitioner now can obtain results in the refined performance model and compare them with the results obtained in measurements. It is important to evaluate the approximation between the metrics given by the performance model and the real scenario. In this activity, the validation is performed to evaluate if the performance model represents the cloud environment with a significant degree of reliability. T-paired tests can be adopted to perform the validation[13], in which the data series that represents the metrics measured in a real scenarios is compared against the metrics obtained in the model.

- **Analysis of New Cloud Scenarios**: Some scenarios are difficult to set up and measure in real cloud environments, such as several real users sending requests or workload intensities that demand large amount of time. For instance, it is difficult for practitioners to call 1000 or more real users to send requests to the cloud environment at the same time. Therefore, the performance model after refinement and validation can be useful and is adopted to perform the stationary analysis considering larger workload intensities than considered in the measurements. Based on these restrictions, the practitioner can evaluate the cloud service under larger workload intensities, considering more users sending requests through the performance model. The validation guarantee that the model results in metrics that represent the real scenario.

5. Performance Model for Cloud Computing Services

Stochastic Petri nets allow practitioners to support performance analysis of cloud environments, in which cloud applications are well modeled based on its workflow[3]. Workflow patterns are represented in the performance model in order to represent the cloud applications into mathematical formalisms allied with graphical resources.

This section presents the proposed performance model based on Stochastic Petri Nets (SPNs). This model is composed of Client subnet and Cloud Infrastructure subnet.

The Client represents clients submitting requests with a certain rate (λ) to the cloud environment, which supports and attend these requests with a processing time(μ) as represented by the Cloud Environment Subnet.

The place Request_submitted indicates that the system is ready to receive the workloads generated by the users. The immediate transition submit_request consider that the request passes immediately to the cloud environment. Once received, the request is in the state Processing_request which is processed with $1/\mu$ associated with the temporized transition processing_time. After completing the request, the system enters in the state Request_completed and release the virtual machine or the container which is providing the environment to attend the requests.

Figure 8 depicts the conceived performance model composed of two subnets:(I) Client subnet; (II) Cloud environment subnet.

We adopt the statement E{#Processing_request})/μ for estimating the throughput metric in the performance model and Eexp indicates the average number of the inner expression(exp), where exp is #Processing request and $1/\mu$ is the service time. The response time [18] is calculated according to Equation 12, in which N represents the number of nodes in the social network while X_0 depicts the average throughput of the social network transactions considering these amount of number of nodes.

$$R = \frac{N}{X_0} \qquad (12)$$

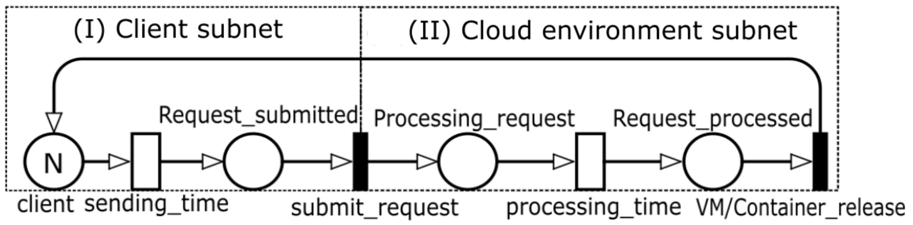

Figure 8. Cloud environment performance model

6. Case Study

This section describes the applicability of the methodology proposed in this paper. The case study is conducted in order to achieve the goals of the strategy, which is composed of a performance evaluation and modeling for cloud services.

- **Cloud Environment Definition and Configuration**: Microsoft Azure is selected to be the cloud platform for some reasons. The advantages offered by this cloud platform, in terms of managing huge volumes of data and the cost details provided are important reasons to choose Microsoft's cloud platform to deploy the virtual machines and the containers with the specifications according to Table 2. Debian Linux is selected to deploy both these virtualizations due to its lightweight utilization of resources when executing applications.

Table 2. Systems specifications

Component	CPU	Mem	OS	HD
VM	2 cores	8 GB	Debian	50 GB
Container	2 cores	8 GB	Debian	50 GB

Figure 9 depicts how the Linkbench benchmark[20] is deployed in the cloud environment. In the cloud platform, virtual machines are deployed in order to provide the resources to support the generated workloads from social networks transactions generated by the benchmark. In the first virtual machine, a Linkbench application is downloaded and installed as a normal Linux application while in the other virtual machine, a container with the Linkbench benchmark is pulled from a public repository and executed as an isolated application, with the characteristics of a container solution. The image containing the Linkbench application own every packages, runtime, and libraries to deploy the benchmark application in an isolated mechanism. This containerized image is builded automatically reading instruction from the Docker file, which contains the workflow of the applications and its dynamics.

Linkbench allows defining the complexity of the social network that generates the workloads. It is represented by the number of nodes in the social network and adjusted according to the desired workload intensity by the practitioner.

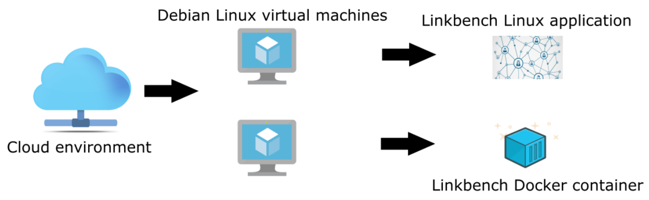
Debian Linux virtual machines
Linkbench Linux application
Cloud environment
Linkbench Docker container

Figure 9. Measurement environment configured in the cloud

- **Cloud Environment Performance Metrics Selection**: The metrics evaluated in this study are related to the quality of services in cloud computing. Throughput is one of these metrics and represents the number of requests completed per second. This metrics is shown at the end of the benchmark execution[20]. The response time is also evaluated in this paper, which represents the mean delay time of a unique social network request among all the requests generated and sent by millions of users.

- **Workload Generation**: In order to choose the benchmark purpose, 607 related papers were founded in IEEE Xplore repository, searching by terms related to big data and cloud computing, such as social media, agriculture, healthcare, finance, and biometrics. The phrase ”social media” appears as one of the most cited words in big data and cloud computing related papers alongside ”healthcare”. Social networks such as Facebook can be modeled as a social graph with several numbers of nodes connected by associations [20]. Nodes represent all the objects including users, pages, posts, and comments, while associations represent the relationships between the nodes. The intensity of the workload generations is defined by the number of nodes and the number of requests that should be sent to the cloud environment.

  Linkbench benchmark simulates social networks and operates in two main phases: Load phase and Request phase. In a first moment, the Load Phase generates the initial state of the social graph according to the configuration edited previously[Linkbench]. In a second moment, the Request phase generates and access concurrently the large social network based on MySQL mechanisms. Statistics such as throughput and response times are provided, collected, and reported after the benchmark execution.

- **Performance Measurements and Statistical Analysis**: After analyzing the virtual machine and the container executing the Linkbench benchmark with different levels of workloads, the results regarding throughput and response time must be analyzed. With 100 nodes in the social network(users, pages, posts), the virtual machine environment completes almost 430 requests per second while, with a better result, the container completes almost 462 requests per second. This pattern continues for 200, 300, and 400 nodes. It demonstrates that the container solution presented a better performance considering quality metrics compared to the virtual machine in the same conditions(Table 3).

Table 3. Throughput(requests/second)

Workload	Virtual Machine	Container
100 nodes	429.18	461.58
200 nodes	511.11	595.13
300 nodes	569.99	611.55
400 nodes	603.59	629.97

In order to test if the throughput metric of the two virtualization technologies is similar and hence, does not present significative difference, the t paired test with 95% level of confidence is performed considering the values of Table 3. This technique tests if the averages of two or more populations are equal or suggest significative differences[13]. As a result of this test, a p-value of 0,038 is stated. Once the p-value is lower than the confidence level(0,05), the throughputs of the virtual machine and container present significative difference, indicating that the container presented a better throughput compared to the virtual machine.

Based on the throughput metric and the experiments, each one of with its own workload intensity and mean-time observation of 24.96 hours, the response times for each virtualization technology are presented in Table 4. The container-solution presented a better performance considering the response time metric, which represents the time taken to attend one single request.

The response times are almost equal. Therefore a t-paired test is applied with the response times provided by each virtualization technology in order to verify it there are significative differences between. The p-value calculated is 0.030. Once the p-value is lower than 0.050, related to the confidence level of 95%, it suggests that there are statistical evidences which suggests a better performance of the container solution compared to the virtualized solution.

- **Cloud Environment Performance Model**: In this phase, the performance model proposed in Section 5 is adopted to represent the big data benchmark configured on virtual machine and container in a cloud.

Table 4. Response time comparison between the virtual machine and the container

Workload	Virtual Machine	Container
100	0.233	0.221
200	0.392	0.343
300	0.534	0.492
400	0.661	0.635

- **Properties Analysis of the Cloud Environment Performance Model**: HiPS tool[19] performed a properties analysis of the SPN proposed in this paper. This analysis concluded that there are no deadlocks in the model structure as well as it is structurally unbounded, which means complete reachability of all states.

- **Refinement of the Cloud Environment Performance Model**: Once verified the properties of the performance model, it is time to choose the polynomial-exponential distribution which best represents as well as adjusts the execution time of the requests from the social network benchmark. Through the average(μ) and the standard deviation(σ)(Table 5) of the execution times provided by the Linkbench, the inverse of the variation coefficient($1/C_v$) was calculated, as demonstrated in Section 3. The appropriate phase approximation of the performance model is depicted in Table 5. In both virtualization technologies, the Hypo-exponential probability distribution is the best choice to represent the execution times where: μ is the average of the processing time(seconds), σ represents the standard deviation of the execution time(seconds), Prob. Dist. is the probability distribution which best adjusts the execution times, Hypo depicts the Hypoexponential probability distribution.

Table 5. Probability Distribution Characterization

System	μ(seconds)	σ(seconds)	$1/C_v$	Prob. Dist
VM	92311.5	6566.21	14.05	Hypo
Container	88344.00	9981.12	8.85	Hypo

Once defined the most appropriate polynomial exponential distribution for the execution times, the parameters of the Hypo-exponential probability distribution(Table 6) are calculated according to the conditions and equations presented in Section 3.

Table 6. Hypo-exponential Parameters

System	μ_1(seconds)	μ_2(seconds)	γ
Virtual machine	792	91512	197
Container	1908	86472	77

- **Mapping of the Performance Metrics**: The performance metrics considered in this paper are represented in the model through the statements presented in Section 5 and depicted in the case study as the equations below. The throughput(T) is calculated through the statement $T = \#Processing\_request/\mu$, where μ is the average processing time of the benchmark execution in the cloud environment while E#Processing_request depicts the mean number of tokens in the place #Processing_request in the performance model.

- **Cloud Environment Performance Model Validation**: The performance model for the virtual machine and the container is validated based on the values presented in Table 7 and Table 8, respectively. The comparison between the results provided by the performance model and the results obtained in the measurements is depicted in these tables.

Table 7. Validation of the Throuhgput(requests/seconds) for the Virtual Machine

Number of nodes	Measurement	Model
100	429.18	468.55
200	511.11	514.67
300	569.99	529.53
400	603.59	580.78

Table 8. Validation of the Throuhgput(requests/seconds) for the Container

Number of nodes	Measurement	Model
100	461.58	594.43
200	595.13	558.23
300	611.55	566.41
400	603.59	575.26

In order to verify statistically if the throughput provided by the performance model is similar to the results obtained in the measurements, the t-paired test[13] is adopted. Considering the virtual machine, the p-value calculated by the test is 0,789, which is higher than 0,05(related to the confidence level of 95%). It means that there are no significant differences between the results provided by the model and the measurements regarding the virtual machine. On the other hand, the same test applied with the throughput of the container resulted in a p-value of 0.372, which is also higher than 0.05. Based on these tests, we conclude that the model represents the real scenario with the confidence of 95%.

- **Analysis of Cloud Scenarios**: This activity provides a cost and performance evaluation of different scenarios. Regarding new scenarios considering more nodes in the social network, instead of 100, 200, 300, and 400 nodes in the social network, the increase of the number of nodes are now considered with 1000, 2000, 3000, and 4000 nodes(N of nodes). The performance model provided the throughput metric when a larger number of users, pages, and posts submit requests to the virtual machine and the container. Table 9 shows that the virtual machine achieved 3% of better throughput for 1000 and 2000 nodes in the social network. When the number of nodes increases, the difference decrease, indicating that for more than 4000 nodes in the social network, the throughput metric for both virtual machine and the container can be very similar.

Table 9. Throughput(requests/second) under higher workload intensities

N of nodes	VM	Container
1000	555.55	540.54
2000	561.79	544.96
3000	569.26	555.55
4000	575.54	579.71

Once validated and refined the performance model, the throughput is predicted by stationary simulation under workload intensities of 1000, 2000, 3000, and 4000 nodes. In order to verify if there are significant differences between the hypervisor solution(VM) and the container solution, a t-paired test is applied with the mean throughput considering each number of nodes in the social network. The p-value of 0,124 calculated in this statistical test suggests that the mean throughput metrics considered these two solution present significative differences and hence, depicts a better performance of the container solution. Based on the results provided in the performance model with the supporting of the t-paired test, the virtual machine presented better results considering more complex social networks.

The response times are calculated dividing the number of nodes by its throughput that each node in the social network would perceive, including users expecting for quality in the social network. For 2000 nodes, the response time for each request in the social network is higher than 3 seconds. The maximum limit to guarantee the client's satisfaction is 1 second[21]. It is an indication that the virtual machine and the container are not providing the quality necessary to support these social network transactions. The results provided by stationary analysis are depicted in Table 10, considering each virtualization technology.

Table 10. Response times(seconds) for higher number of nodes in the social network

N of nodes	VM	Container
1000	1.80	1.85
2000	3.56	3.67
3000	5.27	5.40
4000	6.95	6.90

A t-paired test with the mean response times provided by the virtual machine and the container is performed and with a p-value of 0.234, the test resulted that there are significant differences between these two virtualization technologies, indicating better performance of the hypervisor solution against the container solution considering the response time metric. The container-solution presented a degraded performance against the virtual machine considering high workload intensities. Therefore, for high workload intensities(number of nodes in the social network in this case) the virtual machines would provide a better quality performance compared to the container.

As the general results provided by the modeling of cloud services considered in this work, in the measurement scenarios the containerized solution presented better results while considering larger amount of nodes in the social network, the hypervisor solution presented better results. It means that a composition of virtual machines and containers can be recommended to attend requests generated by social networks of different range of nodes, in which the cloud infrastructure can allocate containers to support social network transactions with fewer nodes and on the other hand, allocate hypervisor solutions in scenarios with larger amount of nodes.

7. Conclusions

This work proposed a strategy for the performance evaluation of virtual machines and containers in cloud environments, when cloud services are deployed in order to support high transactions intensity. These virtualization technologies are compared considering quality metrics such as response time and throughput, which are important to achieve clients fidelity. A benchmark that simulates requests based on the Facebook social network is adopted to generate and submit the workloads, where a case study illustrated the applicability of the proposed performance evaluation strategy in a real cloud environment.

Additional relevant contributions of this work is the performance model to evaluate different environments, with higher workload intensities. The performance model evaluated the throughput and response time metrics when 1000, 2000, 3000, and 4000 nodes compose the social network, generating higher workload intensities. This model showed that for lower workload intensities, the container solution presented a better performance regarding quality metrics. For higher workload intensities, the difference between the virtual machine and the container decrease, indicating that a composition of virtual machines and containers can be a hybrid solution to support social networks transactions.

In future works, we will propose a strategy for performance and dependability evaluation of containers and virtual machines in cloud environments. It means a more detailed study of how big data applications, such as social networks transactions, are processed in cloud environments and how its advantages can improve the quality of cloud services and applications. We intend to evaluate containers and virtual machines considering different big data applications with distinct approaches, such as data related to healthcare and agriculture. In order to enlarge the study of quality services based on virtual machines and containers, other clouds such as AWS and Google Cloud Platform can be evaluated based on the proposed methodology of this work.

Contribuições dos Autores

The authors contributed equally to this work.

References

[1] ERL, T.; MAHMOOD, Z.; PUTTINI, R. *Cloud Computing. Concepts, Techonology and Architecture*. 1. ed. New Jersey, USA: Prentice Hall Press, 2013. v. 1. (The Prentice Hall Service Technology Series from Thomas Erl, v. 1).

[2] NIST - National Institute of Standards and Technology. *The NIST Definition of Cloud Computing*. Online, https://csrc.nist.gov/publications/detail/sp/800-145/final.

[3] MARINESCU, D. C. *Cloud Computing: Theory and Practice*. 1. ed. London, UK: Elsevier, 2017. v. 1.

[4] JAIN, R. *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*. 1. ed. Hoboken, USA: John Wiley & Sons, 1991. v. 1.

[5] KOZHIRBAYEVA, Z.; SINNOTT, R. O. A performance comparison of container-based technologies for the cloud. *Future. Gener. Comput. Syst.*, v. 68, n. 1, p. 175–182.

[6] ARMBRUST, M. et al. *Above the clouds: A Berkeley view of cloud computing*. Berkelet, USA, 2010.

[7] PAHL, C. et al. Cloud container technologies: A state-of-the-art review. *IEEE Trans. Cloud Comput.*, v. 1, n. 1, p. 1.

[8] ALKHANAK, R. et al. Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Gener. Comput. Syst.*, v. 50, n. 1, p. 3–21, 2015.

[9] SHIRINBAB, S.; LUNDBERG, L.; CASALICCHIO, E. Performance evaluation of container and virtual machine running cassandra workload. In: *3rd International Conference of Cloud Computing Technologies and Applications*. Rabat, Morocco: IEEE, 2017. (CloudTech, v. 3), p. 1–8.

[10] SEKAR, V. B. et al. AWS EC2 vs. Joyent's Triton: A Comparison of Docker Container-hosting Platforms. In: *8th Workshop on Scientific Cloud Computing*. washington, USA: ACM, 2017. (ScienceCloud, '17).

[11] BARIK, R. K. et al. Performance analysis of virtual machines and containers in cloud computing. In: *International Conference on Computing, Communication and Automation*. Noida, INDIA: IEEE, 2016. (ICCCA, '16), p. 1204–1210.

[12] FELTER, W. et al. An update performance comparison of virtual machines and linux containers. In: *2015 IEEE International Symposium on Performance Analysis of Systems and Software*. Philadelphia,USA: IEEE, 2015. (ISPASS, '15), p. 171–172.

[13] GUPTA, B. C.; GUTTMAN, I. *Statistics and Probability with Applications for Engineers and Scientists*. 1. ed. Hoboken, USA: John Wiley & Sons, 2017. v. 1.

[14] MEREDITH, M.; URGAONKAR, B. On exploiting resource diversity in the public cloud for modeling application performance. In: *The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*. San Francisco, USA: IEEE, 2017. (CLOUD, '17), p. 171–172.

[15] JOY, A. M. Performance comparison between linux containers and virtual machines. In: *International Conference on Advances in Computer Engineering and Applications*. Ghaziabad, India: IEEE, 2015. '15, p. 342–346.

[16] MARSAN, M. A. et al. *Modelling with Generalized Nets*. 1. ed. New York, USA: John Wiley & Sons, 1994. v. 1.

[17] DESROCHERS, A. A.; AL-JAAR, R. Y. *Applications of Petri Nets in Manufacturing Systems - Modeling, Control, and performance Analysis*. 1. ed. New York, USA: IEEE Control Systems Society, 19945. v. 1.

[18] MENASCE, D.; DOWDY, L.; ALMEIDA, V. A. F. *Performance by Design: Computer Capacity Planning by Example*. 1. ed. New Jersey, USA: Prentice Hall, 2004. v. 1.

[19] HIPS TOOL. *Hierarchical Petri net Simulator*. Online, https://sourceforge.net/projects/hips-tools/.

[20] ARMSTRONG, T. G. et al. Linkbench: a database benchmark based on the facebook social graph. In: . New York, USA: ACM, 2013. (SIGMOD, '13), p. 1185–1196.

[21] Nielsen Norman Group. *Response Times: The 3 Important Limits*. Online, https://www.nngroup.com/articles/response-times-3-important-limits/.