

Avaliação de desempenho e consumo de energia do ambiente MOODLE

Performance and power consumption evaluation of the MOODLE environment

Evaluación del rendimiento y el consumo de energía del entorno MOODLE

Recebido: 18/04/2021 | Revisado: 26/04/2021 | Aceito: 04/05/2021 | Publicado: 16/05/2021

Claudemir Jeremias de Lima

ORCID: <https://orcid.org/0000-0001-9497-4348>
Universidade Federal Rural de Pernambuco, Brasil
E-mail: Claudemir.jlima@ufrpe.br

Alison Vinicius Gomes da Silva

ORCID: <https://orcid.org/0000-0001-9978-2070>
Universidade Federal Rural de Pernambuco, Brasil
E-mail: alison.gsilva@ufrpe.br

Ermeson Carneiro de Andrade

ORCID: <https://orcid.org/0000-0002-9614-4492>
Universidade Federal Rural de Pernambuco, Brasil
E-mail: ermeson.andrade@ufrpe.br

Gustavo Rau de Almeida Callou

ORCID: <https://orcid.org/0000-0002-7997-374X>
Universidade Federal Rural de Pernambuco, Brasil
E-mail: gustavo.callou@ufrpe.br

Resumo

A computação em nuvem, a avaliação de desempenho, os ambientes de aprendizagem virtual de energia (SVA) estão cada vez mais presentes nas instituições educacionais devido ao fato de que tais sistemas representam outro recurso para melhorar a comunicação entre estudantes e professores. Embora estes sistemas não sejam novos, há poucos estudos avaliando seu desempenho, assim como o consumo decrescente de energia. Avaliar a relação entre o desempenho da AVA e o consumo de energia é fundamental para resolver problemas nas fases iniciais de novos projetos ou durante a atualização da infraestrutura existente. Portanto, este trabalho visa avaliar o desempenho e o consumo de energia do ambiente Moodle, uma plataforma que fornece suporte a AVAs, utilizando dois sistemas de nuvem privada, Apache CloudStack e OpenStack. Para isso, adotou-se uma metodologia para apoiar a avaliação de tais experimentos. Além disso, foram realizados estudos de caso para ilustrar a aplicabilidade da metodologia assumindo as nuvens privadas Apache CloudStack e OpenStack. Os resultados revelaram que o desempenho e também o consumo de energia são diferentes entre os dois sistemas. Por exemplo, nos experimentos realizados, o OpenStack mostrou um maior consumo de energia quando comparado com o Apache CloudStack porque exige um tempo de resposta mais longo.

Palavras-chave: Computação em nuvem; Avaliação de desempenho; Consumo de energia; Ensino; Ambiente virtual de aprendizagem.

Abstract

Virtual Learning Environments (VAS) are increasingly present in educational institutions due to the fact that such systems represent another resource for improving communication between students and professors. Although these systems are not new, there are few studies evaluating their performance as well as the decurrent energy consumption. Assessing the relationship between AVA's performance and energy consumption is critical to solving problems in the initial phases of new projects or during the upgrading of existing infrastructure. Therefore, this work aims to evaluate the performance and power consumption of the Moodle environment, a platform that provides AVAs support, using two private cloud systems, Apache CloudStack and OpenStack. In order to accomplish this, a methodology is adopted to support the evaluation of such experiments. Additionally, case studies have been conducted to illustrate the applicability of the methodology assuming the private clouds Apache CloudStack and OpenStack. The results revealed that performance and also energy consumption are different between the two systems. For example, in the experiments performed, OpenStack showed a higher power consumption when compared to the Apache CloudStack because it demands a longer response time.

Keywords: Cloud computing; Performance evaluation; Energy consumption; Teaching; Virtual learning environment.

Resumen

Computación en la nube, evaluación del rendimiento, energía Los entornos virtuales de aprendizaje (EVA) están cada vez más presentes en las instituciones educativas debido a que estos sistemas representan un recurso más para mejorar la comunicación entre estudiantes y profesores. Aunque estos sistemas no son nuevos, existen pocos estudios que

evalúen su rendimiento así como el consumo energético decurrente. Evaluar la relación entre el rendimiento del AVA y el consumo de energía es fundamental para resolver los problemas en las fases iniciales de los nuevos proyectos o durante la actualización de la infraestructura existente. Por lo tanto, este trabajo tiene como objetivo evaluar el rendimiento y el consumo de energía del entorno Moodle, una plataforma que proporciona soporte a AVAs, utilizando dos sistemas de nube privada, Apache CloudStack y OpenStack. Para ello, se adopta una metodología de apoyo a la evaluación de dichos experimentos. Además, se han realizado estudios de caso para ilustrar la aplicabilidad de la metodología asumiendo las nubes privadas Apache CloudStack y OpenStack. Los resultados revelaron que el rendimiento y también el consumo de energía son diferentes entre los dos sistemas. Por ejemplo, en los experimentos realizados, OpenStack mostró un mayor consumo de energía en comparación con Apache CloudStack porque exige un mayor tiempo de respuesta.

Palabras clave: Computación en nube; Evaluación del rendimiento; Consumo de energía; Ensenãza; Entorno de aprendizaje virtual.

1. Introdução

Surgido em 1961, o termo computação em nuvem propôs um modelo de computação que fornece serviços semelhantes à distribuição de energia (Mansur, et al., 2010). A computação em nuvem está cada vez mais presente na sociedade, em diversas áreas (ex.: educacional, comercial e financeira) devido às suas vantagens, como alocação dinâmica de recursos e baixo custo. Existem diversos exemplos de serviços de *e-commerce*, plataforma de *streaming* e ambientes virtuais de aprendizagem que necessitam de recursos dinâmicos e que são facilmente providos pelo paradigma de computação em nuvem.

A computação em nuvem oferece vantagens como: serviços compartilhados, sistemas operacionais e aplicativos de rede, facilitando o gerenciamento de negócios sem se preocupar com detalhes de infraestrutura (Machida, et al., 2011). Por estes motivos, a computação em nuvem passa a representar uma opção para a configuração do ambiente virtual de aprendizagem (AVA), que foi originalmente utilizado como meio de comunicação entre alunos e professores na Educação a Distância (EAD). Atualmente, os AVAs vêm sendo utilizados em salas de aula e cursos a distância, e contam com alguns recursos úteis para a educação. A vantagem do AVA para os discentes é que se pode estudar a qualquer hora, em qualquer lugar, sendo preciso apenas uma conexão com acesso à Internet. Esse tipo de ambiente permite que alunos e professores troquem ideias em fóruns e salas de *chat*, podendo inclusive aumentar a velocidade de aprendizagem.

O AVA é uma plataforma colaborativa que é utilizada no ensino a distância, podendo ser utilizada também no ensino presencial, além de um acompanhamento de grupos de trabalho. No entanto, tal ambiente demanda elevado desempenho, o qual está relacionado diretamente à infraestrutura de Tecnologia da Informação e Comunicação (TICs), que é composta por dispositivos de *hardware* e *software*. Esses ambientes também demandam tecnologias para se realizar a gestão dos dados e para permitir a comunicação entre os seus componentes que podem ser distribuídos. Sendo assim, podem existir múltiplos fatores que venham a influenciar tanto no desempenho como no consumo de energia de uma determinada aplicação hospedada em um sistema em nuvem. Existem vários AVAs utilizados como um sistema de gestão do ensino e aprendizagem. Por exemplo, o Amadeus e Moodle são plataformas de ensino a distância baseadas em *software* livre (Melo Filho, et al., 2014). O Moodle é um *software* desenvolvido para auxiliar os professores tanto em cursos *online* como nos cursos presenciais, e conta com uma variedade de recursos disponíveis. Com uma comunidade composta por centenas de programadores em todo o mundo, de contínuo desenvolvimento, e possui um grupo de apoio aos usuários para prover novas funcionalidades. É importante destacar que os projetistas desses ambientes devem planejar qual sistema e recursos que serão demandados a fim de poder atender aos requisitos de desempenho estabelecidos para tais sistemas (Dantas, 2018).

Este artigo tem como objetivo a proposição de uma estratégia de avaliação de desempenho e consumo de energia que visa a comparação de um Ambiente Virtual de Aprendizagem hospedado em diferentes nuvens privadas. De forma mais específica, as nuvens OpenStack e Apache CloudStack serão comparadas quanto a métricas de desempenho e de consumo de

energia. Para se atingir tal objetivo, a metodologia proposta faz uso de redes de Petri estocásticas (SPN) para quantificar tanto o desempenho como o consumo de energia desses sistemas computacionais.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados a essa pesquisa presentes na literatura e relacionados à medição, modelagem de sistema em computação nas nuvens e de AVAs. A seção 3 introduz os conceitos fundamentais para um melhor entendimento deste trabalho. Em seguida, a Seção 4 apresenta a metodologia adotada nesta pesquisa para avaliar consumo de energia e desempenho. O modelo em SPN proposto é apresentado na Seção 5. A Seção 6 mostra a aplicabilidade do modelo proposto e da estratégia adotada através de estudos de casos aplicados. Por fim, a Seção 7 conclui o artigo e apresenta proposições de futuros direcionamentos.

2. Trabalhos Relacionados

Para posicionar nosso artigo e indicar as suas contribuições, esta seção apresenta os trabalhos relacionados à análise de desempenho de AVAs em computação em nuvem, considerando as metodologias adotadas e as ferramentas utilizadas. Em primeiro lugar são apresentados os trabalhos relacionados à computação em nuvem. Vale destacar que, inicialmente, são destacados os trabalhos que fazem uso de estratégias de medição. Posteriormente, serão apresentados os trabalhos que utilizam estratégias de modelagem e os trabalhos que fazem uso do Moodle. Por fim, é feita uma comparação dos trabalhos presentes na literatura com o que é proposto nesta pesquisa, ressaltando as contribuições.

2.1 Medição

Os autores em (Bruschi, 2016) apresentaram um estudo de caso com o sistema Apache CloudStack utilizando o *hypervisor* XenServer e armazenamento dos dados no sistema Openfiler. Foram utilizados testes de desempenho a partir de três diferentes tipos de perfil de instâncias em uma nuvem computacional privada. O objetivo era o de medir o nível de utilização da CPU, E/S em disco e memória. Como resultado, foi observado um alto consumo de disco na camada de armazenamento de dados, em particular na gravação de dados de E/S. Também foi registrado um alto consumo de memória na camada *hypervisor*, o que pode ser explicado pela utilização do próprio *hypervisor* na alocação de VMs sendo empregadas e usadas no processo.

Em (Vogel, et al., 2017) realizaram uma pesquisa com experimentos em instâncias utilizando *hypervisor* KVM e contêiner, em um sistema de nuvens privadas de IaaS com o sistema Apache CloudStack. Para realizar as medições, foi utilizado NetPIPE, obtendo-se as seguintes métricas: taxa de transferência e a latência da rede. Também foi utilizado o Iperf, que é uma ferramenta de *benchmarking*, sendo uma ferramenta multiprotocolo utilizada para avaliar o desempenho de redes. Como resultado, observou-se que as instâncias de nuvem baseadas em KVM apresentaram pequena degradação do desempenho da rede em relação à taxa de transferência quando comparado com as instâncias baseadas em contêiner.

Em (Vogel Adriano, al., 2016), foi apresentado um trabalho que fez um estudo sobre as ferramentas de gerenciamento de IaaS, discutindo e comparando suas diferenças e desafios. Esse trabalho também analisou a eficiência e o desempenho da computação em nuvem levando em consideração os modelos de implantação e suporte, a flexibilidade, e realizou a avaliação do OpenNebula, CloudStack e OpenStack. O estudo indicou que a vantagem do CloudStack é que ele é mais flexível para implantar um serviço privado IaaS na nuvem, ao passo que o ponto positivo do OpenStack é que ele é mais confiável.

Os autores em (dos Santos, et al., 2020) avaliaram o desempenho dos seguintes serviços: OwnCloud, NextCloud e Pydio. O objetivo do trabalho foi o de ilustrar a aplicabilidade desses serviços sendo executados em nuvens privadas hospedadas em infraestruturas de baixo custo. Um estudo de caso foi realizado para mostrar a estratégia proposta para quantificar o desempenho da CPU, o uso da memória e a taxa de gravação do disco desses serviços. É importante destacar que o trabalho não teve como foco realizar a análise do consumo de energia, por exemplo.

2.2 Modelagem

Para (Fe, et al., 2017) realizaram a proposição de modelos em SPN para auxiliar no planejamento de sistemas de transcodificação de vídeo em nuvem privada e pública. Modelos em SPN foram definidos para se estimar as seguintes métricas: vazão, tempo de resposta e custo. Para validar os resultados obtidos através dos modelos SPN, testes foram realizados no ambiente de nuvem Apache CloudStack que fizeram uso do Apache JMeter, na máquina cliente, para enviar as requisições. De acordo com os resultados, observou-se que o tempo de resposta diminuiu de 30 para 15 segundos, e o custo teve um reajuste de cerca de 300%.

(Torres, et al., 2018) propuseram um método de avaliação para o planejamento de infraestruturas em nuvem privadas, utilizando a plataforma Eucalyptus e o ambiente de armazenamento Pydio. Ele também propôs modelos (em RBD e SPN) para representar e quantificar a disponibilidade e o custo das arquiteturas analisadas. Além disso, também foi realizada análise de sensibilidade para identificar os componentes mais sensíveis. Um estudo de caso foi apresentado para analisar o método de avaliação do planejamento de infraestrutura em nuvem privada.

Em (da Silva, et al., 2019), os autores propuseram uma estratégia baseada em modelagem com redes de Petri estocásticas para avaliar o desempenho do ambiente virtual de aprendizagem Moodle hospedado na nuvem privada Apache CloudStack.

Esse trabalho teve como objetivo a análise do desempenho do Moodle quando, por exemplo, estudantes vão fazer o upload das respostas de uma prova no sistema de forma simultânea (ex., próximo ao horário do término da prova). Este trabalho foi realizado com VMs contendo 1GB e 2GB de RAM e um único núcleo de CPU de 1GHz. Com esta configuração, foi possível realizar experimentos considerando apenas as medidas de desempenho de até 40 usuários simultâneos no sistema. Portanto, ao contrário do trabalho anterior, este trabalho tem como objetivo analisar o desempenho e o consumo de energia entre duas nuvens privadas (OpenStack e Apache CloudStack). Além disso, esse trabalho faz uso de VMs com maior capacidade de memória RAM (4GB) e, também, maior poder computacional (2 núcleos de 2 GHz).

Li et al. (Li, et al., 2019) propuseram uma plataforma de avaliação inteligente para avaliar a qualidade e confiabilidade de salas de aula. Uma estratégia baseada em redes neurais e em modelos de redes de Petri coloridas foi proposta para analisar estudantes e professores em sala de aula. Mesmo tendo proposto modelos para se analisar ambientes de ensino, os autores não tiveram como objetivo analisar o desempenho do Moodle, por exemplo.

2.3 Moodle

Em (Sousa, et al., 2017), foi realizado um planejamento para infraestruturas de nuvens privadas. Os autores propuseram modelos e uma técnica de otimização com GRASP para otimizar custo e desempenho. Os modelos em RBD e SPN propostos foram capazes de realizar a estimativa de desempenho e disponibilidade. Nos estudos de caso, a autora avaliou o impacto da variação de carga, ocorrência de falhas e atividades de recuperação no ambiente de nuvem Eucalyptus. Para isso, diversos cenários foram utilizados para se configurar os *softwares* e o *hardware* utilizados. Na validação do modelo de desempenho, foi utilizado o AVA Moodle configurado em uma máquina virtual na plataforma Eucalyptus.

Os autores (Rakic, et al., 2019) realizaram um estudo da taxa de sucesso de alunos utilizando a plataforma de aprendizagem online Moodle. Esse estudo propôs um método de análise de dados baseado na análise dos dados de redes sociais, na utilização de agrupamento (K-means clustering) e de regressão linear. Essa pesquisa demonstrou existir uma forte correlação entre o sucesso dos alunos e o aumento na utilização de recursos providos através de ambientes como o Moodle.

Dados obtidos através do Moodle também foram analisados em (Assavakamhaenghan, et al., 2020) com o objetivo de poder recomendar os membros de equipe de desenvolvimento de software. Embora esses trabalhos adotaram o Moodle como ambiente de análise, nenhum dos dois trabalhos teve como foco a análise de desempenho desse sistema.

Diferentemente dos estudos anteriores, este trabalho propõe uma estratégia integrada para avaliação tanto do desempenho como do consumo de energia do AVA hospedado em diferentes infraestruturas de nuvens privadas (ex., Apache CloudStack e OpenStack). A Tabela 1 apresenta uma visão geral dos trabalhos relacionados previamente explicados. Pode-se perceber que não foi encontrado um estudo com uma abordagem integrada de desempenho e consumo de energia de AVAs hospedados em diferentes nuvens privadas.

Tabela1. Características dos Trabalhos Relacionados.

Trabalho	Estratégia	Métricas	Sistema em Nuvem	AVA
(Bruschi, 2016)	Medição	Memória, CPU, leitura em disco	Apache CloudStack	Não
(Vogel, et al., 2017)	Medição	Vazão, latência	Apache CloudStack	Não
(Barros, et al., 2017)	Medição	Latência e Consumo de energia	-	Não
(Fe, et al., 2017)	Medição/ Modelagem	Vazão, Temp. Resp. Consumo energia	Cloudstack	Não
(Torres, et al., 2018)	Medição/ Modelagem	Vazão, Número de falhas, número de atendimentos	Eucalyptus	Não
(Sousa, et al., 2017)	Medição/ Modelagem	Memória, CPU	Eucalyptus	Moodle
(Maron, et al., 2014)	Medição	Rede, memória e CPU	OpenStack e OpenNebula	Não
(Ratna , et al., 2017)	Medição	CPU, RAM	OpenStack	Não
(Wang Qiang, al., 2017)	Medição	CPU	Não	Não
(Borba, 2017)	Medição	Armazenamento SSD e HD	OpenStack	Não
(dos Santos, et al., 2020)	Medição	Utilização de CPU, Memória RAM	ownCloud, NextCloud	-
(da Silva, et al., 2019)	Medição/ Modelagem	Vazão, tempo de resposta utilização de CPU, taxa de escrita em HD	Apache CloudStack	Moodle
(Rakic, et al., 2019)	Medição	Taxa de sucesso e nível de utilização	-	Moodle
(Assavakamhaenghan, et al., 2020)	Medição	-	-	Moodle
(Li, et al., 2019)	Modelagem	Acessibilidade	-	Classroom
Este Trabalho	Medição e Modelagem	Memória, CPU, Consumo energia, vazão e tempo de resposta	CloudStack e OpenStack	Moodle

Fonte: Autores.

3. Fundamentação Teórica

Essa seção apresenta os conceitos para uma melhor compreensão deste trabalho. Inicialmente, são apresentados os conceitos sobre computação em nuvem. Em seguida, são mostradas as características tanto do Apache CloudStack como do OpenStack, e são apresentados alguns conceitos básicos sobre avaliação de desempenho. Posteriormente, definições sobre redes de Petri são ilustradas. Por fim, conceitos sobre regressão linear e do Ambiente Virtual de Aprendizagem Moodle são mostrados.

3.1 Computação na nuvem

O *data center* é o elemento central de processamento e armazenamento na nuvem. A virtualização é a tecnologia central do data center, pode-se dizer que a virtualização é uma das principais tecnologias da computação em nuvem. Graças a esta tecnologia, a abstração de recursos físicos pode ser realizada de forma com que vários *hosts* virtuais possam ser criados como um único *host* físico. Os *hosts* virtuais possuem características semelhantes ao dos *hosts* físicos. Sendo assim, essas características de computação em nuvem podem otimizar os recursos de computação, de modo com que os recursos dos *hosts* físicos possam ser particionados para fornecerem serviços para vários clientes (Peixoto, 2012).

A computação em nuvem pode ser vista como um modelo para acesso ubíquo, conveniente e sob demanda. Ela representa uma rede compartilhada de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento,

aplicações e serviços) que podem ser rapidamente adquiridos e liberados com o mínimo esforço gerencial ou a mínima interação com o provedor dos serviços (NIST, 2014). A computação em nuvem oferece, então, vantagens como compartilhamento de serviços, sistemas operacionais e aplicativos em rede, facilitando o gerenciamento dos modelos de negócio sem se preocupar com detalhes da infraestrutura (Machida, et al., 2011).

Os três modelos de serviços em nuvem computacional podem ser classificados de acordo com o seu modelo de negócio (Zota, al., 2014):

- **Infraestrutura como Serviço (*Infrastructure as a Service IaaS*):** É um modelo de serviço que fornece recursos de computação como hardware, armazenamento e rede.
- **Plataforma como Serviço (*Platform as a Service PaaS*):** É um modelo de serviço que o usuário pode utilizar para o desenvolvimento e execução de aplicações.
- **Software como Serviço (*Software as a Service SaaS*):** É um modelo de serviço onde são fornecidos aplicativos executados em uma infraestrutura de nuvem.

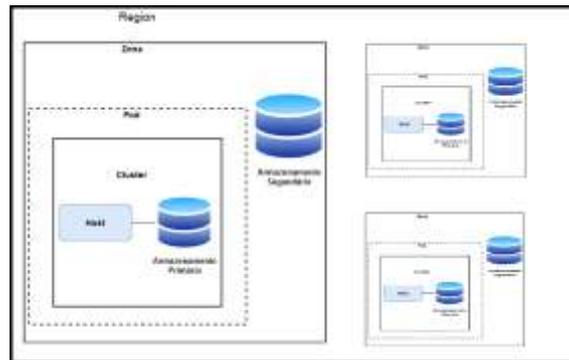
A grande disponibilidade de recursos e serviços, além do crescente acesso de usuários à computação em nuvem, vêm favorecendo o aumento acelerado de hospedagem em diversas soluções de TI. No entanto, existe uma variedade de configurações possíveis e, assim, é importante avaliar tais configurações.

Esses serviços compartilhados são divididos em: nuvem pública, privada, comunitária e híbrida (Zhang, al., 2010, Nist, 2014). A nuvem pública é aplicada quando sua finalidade é compartilhar serviços e recursos pela internet. Já a nuvem privada é, geralmente, utilizada quando sua infraestrutura está totalmente (ou maior parte) dentro de uma estrutura privada, com serviços disponíveis e que tenham a finalidade somente para uma única organização. A nuvem comunitária representa a infraestrutura da nuvem que é compartilhada por várias organizações. Neste modelo, um grupo de pessoas ou empresas, que geralmente têm interesses comuns, se reúnem para criar uma nuvem que é compartilhada apenas entre os membros dessa comunidade. Por fim, a nuvem híbrida é aplicada quando sua infraestrutura está disponível agregando modelos de nuvens públicas e privadas.

3.2 Apache CloudStack

O Apache CloudStack é um *software* de código aberto, projetado para implantar e gerenciar amplas redes de máquinas virtuais, com uma IaaS altamente escalável e disponível (Pauro, 2016). O Apache CloudStack gerencia a computação, a rede e os recursos de armazenamento. O CloudStack utiliza diversos *hypervisors* como KVM, vSphere e XenServer para virtualização, e também tolera a API do AWS, além de suas próprias APIs (Kumar, et al., 2014). Os recursos dentro da infraestrutura em nuvem do Apache CloudStack são denominados por: regiões, zonas, *pods*, clusters, *hosts*, armazenamento (Sabharwal, 2013). A Figura 1 ilustra a relação entre cada um desses itens. A seguir, uma explicação mais detalhada também é fornecida.

Figura 1. Infraestrutura em nuvem do Apache CloudStack.



Fonte: Autores.

- Regiões: uma coletânea de uma ou mais zonas geograficamente próximas gerenciadas por um ou mais servidores. Cada região é controlada pelo seu próprio conjunto de servidores de gerenciamento, que precisam executar em uma das zonas. Regiões representam uma tecnologia útil para fornecer tolerância a falhas e recuperação de dessas três;
- Zonas: é análogo a um único *data center*. Uma zona consiste em uma ou mais *Pods* de armazenamento secundário. A vantagem de organizar a infraestrutura em zonas é fornecer isolamento físico e redundância. Zonas são visíveis para o usuário final e podem ser privadas ou públicas. Zona pública é visível para todos os usuários na nuvem, enquanto que a zona privada é visível apenas para os usuários de um determinado domínio;
- *Pods*: um *pod* de regra é um rack ou fileira de racks que inclui um *switch* de camada 2 e um ou mais clusters. Ele possui um ou mais *hosts*, e um ou mais servidores de armazenamento primários. Os *Pods* estão inseridos dentro de uma zona, e cada zona pode conter um ou mais *Pods*;
- Clusters: um cluster incide em um ou mais *hosts* homogêneos e com armazenamento primário. Assumindo que os nós de computação estão no mesmo cluster, eles possuem o mesmo *hardware* e *hypervisor*, e se estiverem na mesma sub-rede, poderão acessar o armazenamento primário compartilhado;
- Hosts: um único nó de computação dentro de um cluster; muitas vezes um *hypervisor*. O host físico é virtualizado pelo *hipervisor*. CloudStack oferece suporte a vários *hypervisors*, para gerenciar máquinas virtuais entre eles: Xen, KVM, vSphere, o Hyper-V, VMWare;
- Armazenamento Primário: é uma solução de armazenamento fornecido a um único cluster para a execução real de imagens de disco da instância.
- Armazenamento Secundário: sendo um recurso de toda a zona que armazena imagens de disco (ex., imagens ISO).

3.3 OpenStack

O OpenStack é uma plataforma com conjunto crescente de soluções de *software* colaborativo. Foi criado pela NASA para gerenciar e armazenar dados em uma infraestrutura virtualizada tanto em nuvem privada como em nuvem pública. Esses serviços oferecidos pelo OpenStack permitem a virtualização do processamento (computação) e da rede como Serviço (IaaS), formando um conjunto de serviços de *software* distribuídos e interconectados.

O OpenStack apresenta as seguintes características: escalabilidade, flexibilidade, e é *open source*. Dessa forma, é possível redefinir os patamares de velocidade de implementação, custo e escalabilidade dos serviços, oferecendo flexibilidade em suas decisões tecnológicas. A Figura 2 apresenta a infraestrutura dos principais componentes do sistema em nuvem OpenStack (Bruschi, 2016). A seguir, apresentamos mais detalhes sobre os seus componentes.

- *Keystone* (Serviço de Autenticação): disponibiliza a autenticação e autorização para todos os serviços;
- *Glance* (serviço de imagem): garantindo um catálogo e repositório de imagens de discos virtuais, que são utilizados pelas VM;
- Nova (serviço de computação): disponibiliza máquinas virtuais à medida que são demandadas pelos clientes;
- *Neutron* (serviço de rede): proporciona conectividade de rede como um serviço, permite acesso a uma API para requerer e configurar redes virtuais;
- *Horizon* (serviço de painel-Dashboard): é uma interface para visualizar todos os serviços;
- *Cinder* (bloco de armazenamento): fornece armazenamento em nível de bloco, que pode ser montado como volume pelas instâncias (servidor virtual);
- *Swift* (Objeto de armazenamento): serviço de armazenamento de objetos, por onde os clientes podem armazenar ou buscar seus arquivos (Vogel, et al., 2016).

Figura 2. Infraestrutura em nuvem do OpenStack.



Fonte: Autores.

3.4 Avaliação de Desempenho

A avaliação de desempenho pode ser realizada via medições, modelagem e simulações (Callou, et al., 2011). A mensuração de um sistema é o método mais direto para a avaliação de desempenho. As técnicas de medição, apesar de fornecerem respostas exatas sobre o desempenho do sistema, não podem ser utilizadas (ou são muito caras) se um sistema estiver em operação. Além disso, embora a carga de trabalho real seja uma boa escolha por representar fielmente o sistema, às vezes essa opção não é viável. Isso pode acontecer, por exemplo, quando o tamanho da carga não é representativo com os dados coletados sujeitos a muitas interferências externas. A fim de facilitar a vida dos projetistas que realizam medições, existem alguns mecanismos usados para auxiliar na geração da carga de trabalho (Lilja, 2005, Sousa, 2015).

A escolha da carga de trabalho é tão importante quanto a definição da estratégia de medição a ser seguida. Diferentes estratégias de medição são baseadas no conceito de eventos, que são mudanças no estado do sistema. A definição exata do evento depende de métricas. Por exemplo, os eventos podem ser definidos como acessos a disco (*upload* de um arquivo), acesso de memória ou comunicação de rede de computadores. Assim, as ferramentas utilizadas para avaliação de desempenho de sistemas computacionais via medição podem modificar o comportamento do objeto em teste. Quanto maior a informação e resolução que as ferramentas de medição podem fornecer, maior pode ser a interferência introduzida por esta ferramenta de medição. Essa interferência introduzida pela ferramenta de medição pode reduzir a qualidade dos dados aferidos (Lilja, 2005, Sousa, 2015). No entanto, também existe a possibilidade de se realizarem medições via *hardware* externo ao ambiente que está sendo analisado. E, assim, adotando essa estratégia, o impacto sob o sistema analisado não ocorre.

Já a modelagem analítica é utilizada para a abstração do sistema. Assim, o modelo deve conter detalhes do sistema que são essenciais para o seu comportamento. Por fim, a simulação do modelo poderá fornecer resultados não tão precisos quanto os fornecidos pela medição, mas é possível calcular as estimativas considerando as precisões dos erros desejados. Vale ressaltar que existem diversos tipos de modelos formais que podem ser utilizados para quantificar as métricas de desempenho desejadas.

A simulação pode ser usada para avaliação de desempenho e verificação de modelos analíticos. Ao contrário das medições, as simulações são baseadas em modelos abstratos do sistema, portanto, podem ser aplicadas sem a necessidade de implantar totalmente o sistema. Portanto, o modelo utilizado no processo de simulação é concluído abstraindo as características básicas do sistema, e sua complexidade, e o grau de abstração adotado pode variar de um sistema para outro. No processo de simulação, os valores assumidos dos parâmetros do sistema podem ser controlados. Isso facilita obter informações relacionadas à avaliação de desempenho (Lilja, 2005).

Dependendo das condições de tempo e recursos, o uso de mais de uma técnica pode permitir a compreensão melhor do sistema e fornecer informações mais precisas para auxiliar nas tomadas de decisões. Com a ajuda de ferramentas de modelagem, simulação e medição, informações detalhadas sobre o sistema podem ser obtidas. Usando a tecnologia de medição, os resultados mais precisos podem ser obtidos porque é a tecnologia que melhor corresponde ao ambiente real sendo analisado. No entanto, a modelagem fornece vários métodos alternativos de avaliação de desempenho, que podem ser obtidos através de simulação ou modelagem analítica, onde os modelos analíticos lidam com métodos probabilísticos, teoria de filas, modelos de Markov ou redes de Petri (Callou, et al., 2011). Vale destacar que a modelagem e simulação permitem a avaliação do sistema para diferentes cenários, ainda não disponíveis no sistema em análise.

3.5 Redes de Petri

O conceito de redes de Petri foi introduzido por Carl Adam Petri, em 1962, como uma técnica gráfica e formal para representar e simular o comportamento de sistemas (Petri, 1962). Desde então, tem sido utilizada para modelagem e análise dos sistemas em várias áreas do conhecimento. Essas redes permitem descrever e analisar sistemas de processamento de informações que são caracterizados como sistemas concorrentes, assíncronos, distribuídos, paralelos, não determinísticos ou estocásticos. As redes de Petri representam uma técnica visual bastante utilizada para auxiliar no entendimento do comportamento de sistemas, e é capaz de simular a dinâmica e as atividades concorrentes de sistemas (Murata, 1989).

A rede de Petri é uma técnica de análise e modelagem de sistemas complexos amplamente difundida. Esta técnica possui representação matemática e mecanismos de análise que permitem a verificação de características dos sistemas. As redes de Petri básicas são compostas por lugares, transições, arcos e tokens (ver Figura 3). Contudo, a necessidade de atender a outras áreas do conhecimento fez com que outras extensões surgissem ao longo do tempo, como as redes de Petri temporizadas (Merlin, et al., 2020, Mendonça, 2015).

Figura 3. Elementos da Redes de Petri.

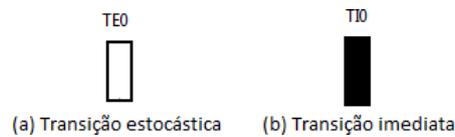


Fonte: Autores.

3.6 Redes de Petri Estocástica

As redes de Petri estocásticas (SPN) são uma extensão das redes de Petri, e podem ser empregadas, por exemplo, para a modelagem de desempenho e dependabilidade (Torres, et al., 2016). As SPN passam a ter duas representações distintas para as transições, como pode ser visto na Figura 4. Existe a transição estocástica, Figura 4 (a), onde um tempo é associado à transição para representar o período de execução da atividade. A Figura 4 (b) mostra a transição imediata que não demanda tempo para o disparo (tempo igual a zero), ou seja, o disparo é instantâneo (Silva, 2012). A definição formal das SPNs pode ser encontrada em (German, 2000).

Figura 4. Elemento da SPN.

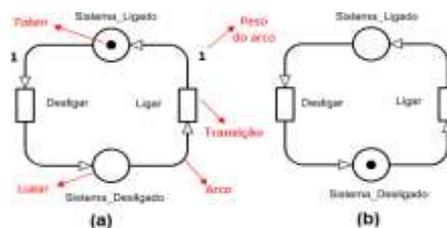


Fonte: Autores.

A Figura 5 mostra um exemplo de uma SPN que representa o funcionamento de um sistema operacional de um computador. Para isso, definimos dois lugares que representam os estados que o sistema pode ter (*Sistema Ligado* ou *Sistema Desligado*), e duas transições que modelam as ações para alterar o estado do sistema (*Ligar* e *Desligar*).

Inicialmente, temos o sistema em funcionamento, o que é representado por um Token no lugar *Sistema Ligado*, conforme pode ser visto na Figura 5 (a). No estado atual do modelo, a única transição que pode ser disparada é *Desligar*. Assim que esta transição for disparada, o modelo passará do estado *Ligado*, como representado na Figura 5 (a), para o estado *Desligado*, conforme pode ser visto na Figura 5 (b). Vale destacar que as transições *Ligar* e *Desligar* tem um tempo associados, em outras palavras, esse tempo representa a duração de cada atividade.

Figura 5. Exemplo de uma Rede de Petri (RdP)



Fonte: Autores.

3.7 Regressão Linear

A regressão linear é uma técnica estatística usada para relacionar uma variável x , independente, com uma variável y que é dependente (Matos, 1995). Este trabalho fará uso da regressão linear no modelo SPN proposto para a simulação do consumo de energia, vazão e tempo de resposta, por exemplo. A regressão linear simples é dada pela seguinte Equação 1.

$$y = a + bx \quad (1)$$

Onde y é o resultado da saída prevista, x é a variável de entrada e a e b são os parâmetros de regressão.

O cálculo do parâmetro de regressão b é dado pela seguinte Equação 2.

$$b = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (2)$$

Onde n é quantidade de pontos $(x_i; y_i)$.

Já o cálculo do parâmetro de regressão a é dado pela seguinte Equação 3.

$$a = y - bx \quad (3)$$

Onde y e x é a média aritmética da variável de saída e de entrada, respectivamente.

Existe um grau de erro associado à regressão que é chamado de coeficiente de determinação e é representado por R^2 (Lilja, 2005). Esse grau de erro varia entre zero e um. Quanto mais próximo o valor do coeficiente de determinação estiver de um, maior será a validade da regressão. O cálculo do coeficiente de determinação é dado pela seguinte Equação 4.

$$R^2 = \frac{b \sum_{i=1}^n (x_i - x)(y_i - y)}{\sum_{i=1}^n (y_i - y)^2} \quad (4)$$

3.8 Moodle

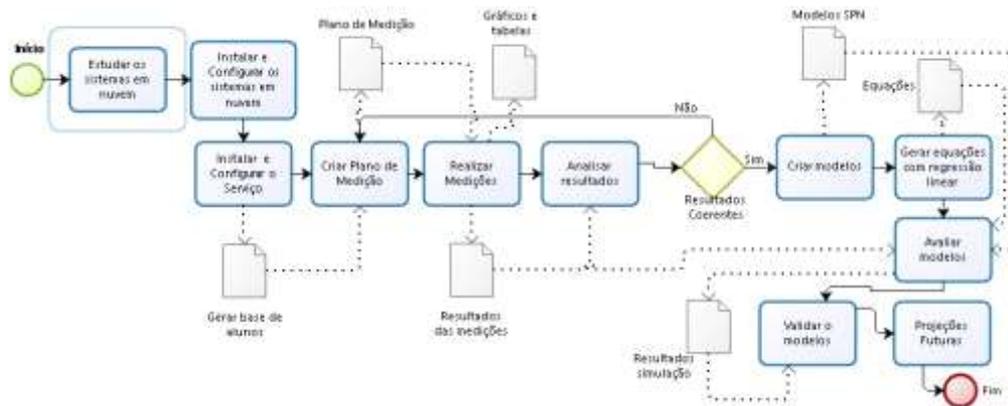
Um dos exemplos do crescimento acelerado de sistemas hospedados na nuvem é o Moodle, ambiente virtual de aprendizagem colaborativo, que facilita o acompanhamento e permite a gestão do ensino e aprendizagem para alunos e professores. Muitas instituições e pessoas estão utilizando o AVA Moodle tanto em cursos online como em cursos presenciais, seja em universidades ou em cursos profissionalizantes (Rakic, et al., 2019). Esse crescimento na utilização pode ser explicado pela busca frequente de flexibilidade de horário e espaço para realização dos cursos, tornando-se cada vez mais popular entre educadores e estudantes.

Em (SOUSA, 2015), é apresentada uma descrição sobre o Moodle explicando a função do sistema aberto de gerenciamento de aprendizagem, bem como suas ferramentas. O Moodle é um sistema *open source* de gerenciamento de aprendizagem e de trabalho colaborativo em ambiente virtual que permite a criação e administração de cursos *online*, grupo de trabalho e comunidades de aprendizagem. Esse AVA também fornece ferramentas de avaliação específicas, como, por exemplo: discussões de fórum, questionários, entradas de glossário, *chat* e Wiki (MOODLE, 2020).

4. Metodologia

Esta seção apresenta a metodologia adotada para avaliar o desempenho do AVA hospedado em nuvens privadas. A metodologia descreve as atividades realizadas, os objetos gerados em cada uma das atividades e, também, provê detalhe de como foram realizadas as coletas de dados. Por fim, é apresentada a infraestrutura utilizada como cenário dos experimentos realizados nesta pesquisa. A Figura 6 ilustra a metodologia. A seguir, cada uma das atividades é detalhada.

Figura 6. Metodologia adotada.



Fonte: Autores.

- **Estudar os sistemas em nuvem:** o propósito desta fase é fazer um estudo com a finalidade de se obter uma compreensão da infraestrutura em nuvem a ser analisada e seus componentes. Além disso, nessa fase também devem ser analisados os serviços que serão configurados na nuvem (ex., Moodle) e as métricas que serão usadas para avaliar o desempenho.
- **Instalar e configurar os sistemas em nuvem:** nesta atividade são realizadas a instalação e configuração da infraestrutura em nuvem (ex., Apache CloudStack e o OpenStack).
- **Instalar e configurar o serviço:** esta fase corresponde à atividade de instalação e configuração de um serviço na máquina virtual de cada um dos sistemas em nuvem a serem comparados.
- **Criar o plano de medição:** nesta atividade, um plano de medição é criado, na ferramenta Apache JMeter, com o objetivo de realizar os experimentos no ambiente em análise. Essa atividade é responsável, ainda, pela criação das bases de dados que serão utilizadas; por exemplo, pode-se criar a base de alunos, professores, cursos e turmas. Vale destacar que todos os dados inseridos são hipotéticos. O plano de medição é usado para definir as atividades a serem executadas, como, por exemplo, acessar o Moodle, realizar *login*, acessar o curso, acessar a atividade, realizar *upload* de arquivos com 1Mb e sair do AVA. É nesse plano que deve ser definido o número de usuários que irão acessar o sistema e o tempo de inicialização para cada um.
- **Realizar medições:** esta atividade corresponde às medições utilizando JMeter e/ou NMON, de acordo com o plano de medição criado na atividade anterior. O Apache JMeter é o responsável pela coleta de métricas como vazão e tempo de resposta. Já o NMON pode ser utilizado para quantificar a utilização da CPU e da memória RAM, por exemplo. As medições devem ser repetidas, no mínimo, trinta vezes para aumentar a confiabilidade dos resultados obtidos.
- **Analisar resultados:** nesta etapa, devem ser feitos os cálculos estatísticos da média, desvio padrão e intervalo de confiança. E, ao final, são gerados gráficos de comparação dos resultados, por exemplo, do tempo de resposta, vazão, utilização de CPU e da memória RAM, e consumo de energia.
- **Resultados coerentes:** caso os resultados não se encontrem conforme o esperado, ou seja, não estejam coerentes, a metodologia faz com que se volte para a fase de Criar Plano de Medição. Caso os resultados estejam coerentes, segue-se para a fase de criar modelos.
- **Criar modelos:** esta fase refere-se à criação de modelos de desempenho. Vale salientar que estes modelos podem ser feitos, por exemplo, utilizando o formalismo SPN. Vale destacar que esse modelo poderá ser refinado na fase seguinte, com o emprego de técnicas de regressão linear. As equações criadas podem ser utilizadas para se conseguir

representar o tempo demandado para diferentes atividades no sistema. Sendo assim, a transição do modelo SPN que representa tal atividade irá possuir uma equação que foi o resultado da regressão linear realizada.

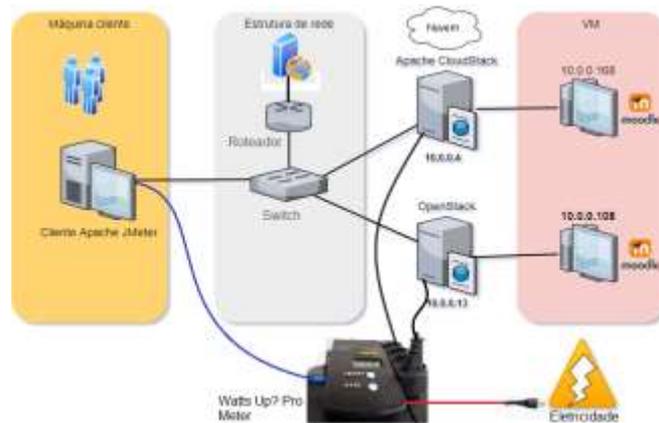
- **Gerar equações com regressão linear:** é nesta fase que as equações de regressão linear são geradas. Por exemplo, pode-se fazer o cálculo da regressão linear entre a quantidade de clientes no sistema e o tempo de atendimento, ou ainda, entre o nível de utilização da CPU e o consumo de energia decorrente. Então, é nesta fase que são geradas as equações que serão utilizadas para refinar o modelo de desempenho em SPN.
- **Avaliação do modelo:** esta etapa é responsável por realizar a avaliação do modelo proposto, considerando os cenários adotados no plano de medição.
- **Validar o modelo:** para se realizar a validação do modelo proposto, é adotada a seguinte estratégia: (i) deve-se observar o intervalo de confiança para cada métrica de interesse, tanto nos resultados do modelo quanto nos resultados das medições; (ii) comparam-se os intervalos de confiança e, se estes estiverem se sobrepondo, pode-se confirmar que não existe diferença estatística significativa, com 95% de confiança entre os resultados do modelo e os resultados dos experimentos.
- **Resultados coerentes:** se o modelo for validade, segue-se para a fase Projeções futuras. Caso contrário, a metodologia retorna o processo à fase de Criação de Modelos.
- **Projeções futuras:** uma vez validado o modelo, pode-se utilizá-lo para fazer estimativas inviáveis de serem feitas em medições. Assim, o modelo pode ser utilizado para análises mais complexas. Esta é uma das grandes vantagens de se utilizar a modelagem.

4.1 Infraestrutura

A Figura 7 ilustra a visão geral do ambiente montado para a realização dos experimentos. O ambiente é composto por três máquinas, sendo (i) uma cliente, com Apache JMeter para gerar a carga de trabalho e quantificar as métricas de desempenho; (ii) uma máquina com o ambiente de nuvem gerenciado pelo Apache CloudStack; e (iii) uma máquina com o ambiente de nuvem gerenciado pelo OpenStack. Em cada máquina com sistema de nuvem foi criada uma VM, configurado o Moodle e a ferramenta de medição NMON. É importante destacar que a comunicação entre essas máquinas é feita por um *switch* Gigabit. Além disso, pode-se observar a existência de um medidor para quantificar o consumo de energia das máquinas com o ambiente de nuvem privada.

Todas as máquinas utilizadas são equivalentes e são compostas por CPU Intel Core i5 de 3,4 GHz, 8 MB Cache, 8 GB RAM, NIC 1 Gigabit Ethernet 10/100/1000, 500 HD. O sistema operacional adotado foi o CentOS 7. Para ambas as nuvens foi instalado o hypervisor KVM. A VM utilizada foi configurada com processador de 2 GHz, com 2 CPUs e 4 GB de memória RAM, 20 GB de espaço em disco e com o sistema operacional Linux CentOS 7 e banco de dados MySQL. A seguir, será apresentado o modelo de desempenho proposto por este trabalho para avaliar as nuvens privadas Apache CloudStack e o OpenStack.

Figura 7. Infraestrutura montada para os experimentos.



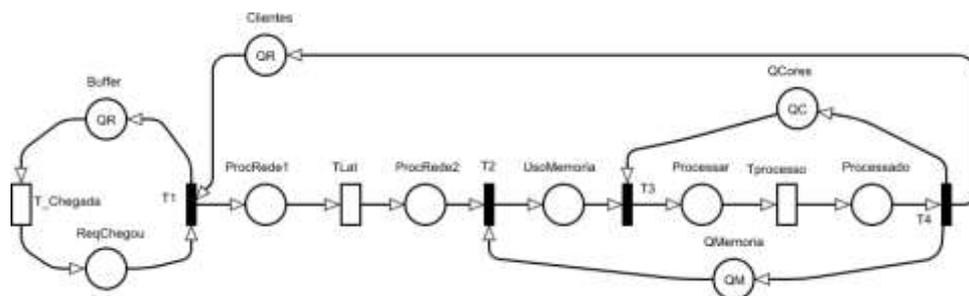
Fonte: Autores.

5. Modelo de Desempenho

Esta seção inicia com a apresentação do modelo SPN proposto para a quantificação do desempenho dos sistemas Apache CloudStack e OpenStack com o Moodle hospedado. Na descrição do modelo, são apresentados os lugares e as descrições das transições. Em seguida, são mostradas as regressões lineares e as métricas utilizadas.

A Figura 8 mostra o modelo SPN de desempenho proposto para representar o comportamento do Moodle hospedado de acordo com a infraestrutura apresentada na Figura 7. O modelo representa o funcionamento da chegada de requisições dos usuários até a resposta do sistema. Dessa forma, usuários enviam requisições, as quais passam pela rede, acessam a memória e são processadas pela CPU.

Figura 8. Modelo de desempenho proposto.



Fonte: Autores.

O número de tokens (QR) presente no lugar *Clientes* representa a quantidade de clientes utilizando o sistema Moodle, simultaneamente. O lugar *Buffer* armazena os clientes, a transição *T_Chegada* possui o tempo de chegada e, ao ser disparada, consome um token de *Buffer* e dispara um token para *ReqChegou*. A transição imediata *T1*, ao ser disparada, consome um token de *Clientes* e um token de *ReqChegou* e gera um token no lugar *ProcRede1*. Os lugares *ProcRede1* e *ProcRede2* representam a infraestrutura de rede. O tempo da latência para cada requisição dentro do sistema é associado à transição exponencial *TLat*, valor que pode variar de acordo com a quantidade de clientes (QR). O disparo da transição *TLat* consome um token de *ProcRede1* e gera um token no lugar *ProcRede2*. A transição imediata *T2*, ao ser disparada, consome um token de *ProcRede2* e também um token de *QMemoria*, gerando um token no lugar *UsoMemoria* que indica que a requisição já foi carregada na memória para poder ser executada. A seguir, o disparo da transição *T3* consome um token de *UsoMemoria* e

outro de *QCores*, e deposita um token no lugar *Processar*, indicando que o processamento será iniciado. O tempo de processamento na CPU para cada requisição é associado à transição exponencial *TProcesso*, e o disparo desta transição consome um token do lugar *Processar* e gera um token em *Processado*, que indica o término do processamento. Por fim, o disparo da transição imediata *T4* consome um token de *Processado* e libera os recursos consumidos (tokens) para *QCores*, *QMemoria* e *Clientes*. O label *QC*, associado ao lugar *QCores*, representa o número de Cores de CPU que o sistema em análise possui. Já o label *QM*, associado ao lugar *QMemoria*, representa a quantidade de memória RAM disponível para ser utilizada pelo sistema através das requisições. As Tabelas 2 e 3 apresentam, respectivamente, um resumo das descrições dos lugares e das transições do modelo SPN proposto.

Tabela 2. Descrição dos lugares do modelo.

Lugar	Descrição
Buffer (QR)	Quantidade de clientes para atendimento no sistema
ReqChegou	Quantidade esperando atendimento no sistema
Clientes (QR)	Quantidade de clientes sendo atendidos no sistema
ProcRede1	Requisições na rede antes de serem enviadas ao sistema
ProcRede2	Solicitações foram enviadas ao sistema e se encontram esperando.
UsoMemoria	Requisições na memória aguardando serem escalonados para o processador.
Processar	Requisições em processamento pelo sistema.
Processado	Finalização do processamento.
QCpu (QC)	Quantidade de CPUs disponíveis para serem usados.
QMemoria (QM)	Quantidade de Memória RAM disponível para ser utilizada na VM.

Fonte: Autores.

É importante destacar que, com esse modelo, é possível fazer a variação de alguns parâmetros, como, por exemplo, pode-se variar a quantidade de clientes *QR* no sistema e também o *TLat*, que é o tempo de latência da rede e corresponde ao tempo necessário para que a carga de trabalho seja enviada pela rede ao sistema em análise. Essa latência pode variar de acordo com a quantidade de requisições, gerando impacto diretamente no tempo total de processamento. Outro parâmetro é o *TProcesso*, que corresponde ao tempo de processamento das requisições na CPU, e o seu valor pode variar conforme a quantidade de clientes. Existe ainda o parâmetro *QC*, que representa a quantidade de Cores que a CPU possui. Por exemplo, o valor 2 significa que a VM faz uso de duas cores de CPUs.

Tabela 3. Descrição das transições do modelo.

Transição	Tipo	Tempo	Tipo Serv.	Descrição
TChegada	Exponencial	-	Single Server	Chegada de 1 cliente
T1	Imediata	-	-	Chegada de 1 cliente
TLat	Exponencial	TLatencia	Single Server	Latência da rede.
T2	Imediata	-	-	Alocação de Memória
T3	Imediata	-	-	Alocação de Processador
TProcesso	Exponencial	TProc	Infinite Server	Tempo de processamento
T4	Imediata	-	-	Liberação dos recursos

Fonte: Autores.

O último parâmetro é a quantidade de memória no sistema, sendo representado por *QM*. Por exemplo, o valor 4 pode representar que a memória RAM da VM possui 4 Gigabytes. O modelo da Figura 8 faz uso de três métricas: (i) utilização da

CPU, (ii) quantidade de memória RAM utilizada e, por fim, (iii) consumo de energia em Watts. Mais detalhes dessas métricas são apresentados na Seção 5.2.

5.1 Regressão Linear

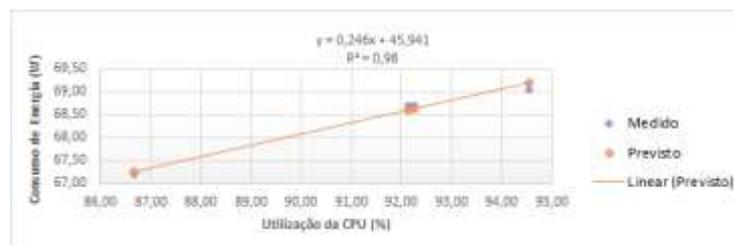
Este trabalho faz uso de regressões lineares com o objetivo de propor equações para serem utilizadas no modelo SPN proposto. Por exemplo, os resultados das regressões podem ser utilizados como o *delay* de transições ou, ainda, na definição de métricas dentro do próprio modelo SPN.

5.1.1 Consumo de Energia e Nível de Utilização da CPU

Existe uma relação entre o consumo de energia e o nível de utilização da CPU. Sendo assim, resolvemos fazer uma regressão linear entre esses dois itens para determinar uma equação que possa ser utilizada no modelo SPN proposto acima. A Figura 9 apresenta a regressão linear utilizada para definir essa relação entre o consumo de energia e o nível de utilização da CPU para o Apache CloudStack. Nessa figura, o eixo Y representa o consumo de energia (*W*) e o X nível de utilização da CPU (*UC*). Os resultados dessas regressões serão utilizados e melhor ilustrados na Seção 5.2.

Ainda sobre a Figura 9, pode-se observar que a regressão praticamente se sobrepõe aos pontos. Vale destacar que a cor azul representa os dados consumidos de fato, e a cor vermelha representa a previsão obtida através da regressão. Cada um dos pontos consumidos representam o resultado da média de 30 medições realizadas no ambiente em laboratório. A qualidade da regressão pode ser comprovada pelo valor de R^2 que foi 0,98 para o Apache CloudStack. R^2 bem próximo ao valor um (1), indica a validade da regressão. A equação obtida pela regressão linear entre o consumo de energia e a utilização da CPU, na nuvem Apache CloudStack, é mostrada a seguir.

Figura 9. Gráfico da Regressão linear Consumo Energia Apache CloudStack.

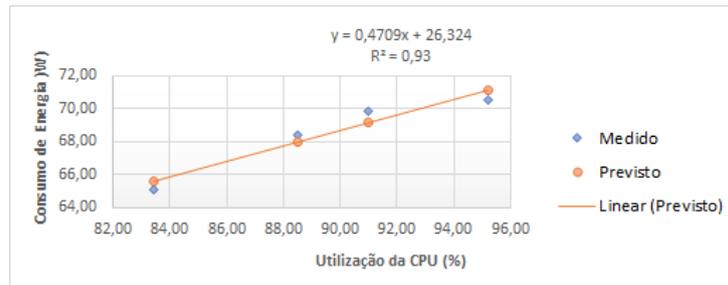


Fonte: Autores.

$$y = (0,246 \times UC) + 45,941 \quad (5)$$

Onde y é a previsão do consumo de energia em Watts e UC é a utilização da CPU. A partir dessa equação, é possível estimar o consumo de energia do *controller*, que corresponde à máquina com a nuvem instalada. Essa equação é utilizada no consumo de energia do modelo de desempenho (Figura 8).

Figura 10. Gráfico da Regressão linear Consumo Energia OpenStack.



Fonte: Autores.

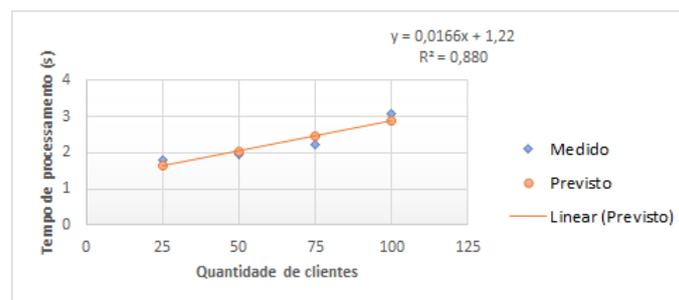
Já para o OpenStack, a Figura 10 apresenta a regressão linear utilizada para definir a relação do consumo de energia e a utilização da CPU para o OpenStack. O resultado dessa regressão é mostrado na Equação 6. Essa equação é utilizada no consumo de energia do modelo de desempenho (Figura 8) para a nuvem OpenStack. Sobre essa regressão (ver Figura 10), pode-se observar que a regressão obteve um valor de R^2 de 0,93, indicando assim a Validade da regressão. É importante destacar que cada ponto da regressão representa um valor médio aferido no sistema para amostras de tamanho 30.

$$y = (0.4709 \times UC) + 26.324 \quad (6)$$

5.1.2 Regressão do tempo de processamento e quantidade de clientes

Outra relação importante e que também foi analisada através de regressão linear é a do tempo de processamento com a quantidade de clientes fazendo requisições no sistema com Apache CloudStack e OpenStack. Esta regressão linear representa a quantidade de clientes (x , independente) e o tempo de processamento para atender a *requisições* por cliente (y , dependente).

Figura 11. Regressão linear Tempo de processamento Apache CloudStack



Fonte: Autores.

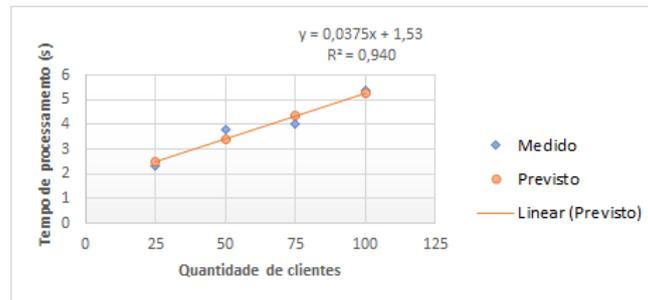
A Figura 11 mostra que os valores para y previstos pela regressão são próximos aos valores obtidos via medição. Destacamos que cada ponto da medição representa a média de uma amostra com tamanho 30. A qualidade da regressão pode ser confirmada com o valor computado para R^2 , que foi de 0,880. Essa regressão para o Apache CloudStack é dada pela Equação 7.

$$y = (0.0166 \times QR) + 1.22 \quad (7)$$

Onde Y é a previsão do tempo de resposta e QR é a quantidade de clientes fazendo solicitações no sistema.

Similarmente, foi desenvolvida uma regressão linear para o OpenStack. A Figura 12 mostra os valores de y previstos pela regressão para o OpenStack. A qualidade da regressão pode ser confirmada com o valor computado para R^2 , que foi de 0.94. A regressão linear da quantidade clientes e do tempo de processamento, para o OpenStack, é dada pela Equação 8.

Figura 12. Regressão linear Tempo de processamento OpenStack.



Fonte: Autores.

$$y = (0.0375 \times QR) + 1.53 \quad (8)$$

5.2 Métricas

Esta seção apresenta as métricas adotadas no modelo SPN proposto (Figura 8) e que são aplicadas para os dois sistemas de nuvem, Apache CloudStack e OpenStack.

5.2.1 Utilização de CPU

A primeira métrica, *UtilizacaoCPU*, é responsável pelo cálculo da utilização da CPU na VM. A fórmula da utilização da CPU é definida pela Equação 9.

$$UtilizacaoCPU = \frac{(E\{\#Processar\})}{QC} \times 100 \quad (9)$$

Onde $(E\{\#Processar\})$ indica o número médio de tokens no lugar *Processar* e QC corresponde a quantidade de CPUs.

5.2.2 Utilização da Memória

A métrica do modelo *UtilizacaoMemoria* é responsável pelo cálculo da utilização da memória RAM na VM. A utilização da memória é representada em % conforme definida na Equação 10.

$$UtilizacaoMemoria = \frac{QM - (E\{\#QMemoria\})}{QM} \times 100 \quad (10)$$

Onde QM indica a quantidade de memória disponível na VM e $(E\{\#QMemoria\})$ corresponde ao número médio de tokens no lugar *QMemoria*.

5.2.3 Consumo de Energia

Utilizando o modelo, a variável *ConsumoEnergia* é responsável por calcular o consumo de energia da máquina com a nuvem que hospeda o Moodle. Esse cálculo consiste em verificar, através do ambiente de medição, o consumo elétrico pelo uso do *Controller* com instâncias ativas e com N clientes demandando solicitações à VM. As Equações 11 e 12 definem como é feito o cálculo do consumo de energia para o Apache CloudStack e OpenStack. É importante destacar que, em ambos os casos, foram utilizadas as regressões lineares mostradas anteriormente. Sendo assim, o consumo de energia do Apache CloudStack faz uso da Equação 5, e o CloudStack utiliza a regressão mostrada na Equação 6. Mais detalhes sobre essas regressões estão disponíveis na Seção 5.1. A regressão linear simples é dada pela seguinte Equação $y = bx + a$.

Consumo de energia do Apache CloudStack

$$\text{ConsumoEnergia} = \left(0.246 \times \frac{(E\{\#Processar\})}{QC} \times 100 \right) + 45.941 \quad (11)$$

Onde $b = 0.246$, $(\{E\ \#Processar\})$ indica o número médio de tokens no lugar *Processar* e QC corresponde a quantidade de CPUs disponíveis e $a = 45.941$.

Consumo de energia do OpenStack

$$\text{ConsumoEnergia} = \left(0.4709 \times \frac{(E\{\#Processar\})}{QC} \times 100 \right) + 26.324 \quad (12)$$

Onde $b = 0.4709$, e $a = 26.324$.

5.3 Vazão

A métrica do modelo *VZ* é responsável pelo cálculo da vazão do sistema. A vazão representa a quantidade de requisições que o sistema atende durante um período de tempo. A fórmula da vazão é definida nas Equações 13 e 14 para o OpenStack e Apache CloudStack, respectivamente.

Vazão OpenStack:

$$vz = \left(\frac{(E\{\#Processar\})}{(0.0375 \times QR) + 1.53} \right) \quad (13)$$

Onde $(\{E\ \#Processar\})$ indica o número médio de tokens no lugar *Processar*. O tempo que leva para processar a requisição (*delay* da transição *TProcesso*) foi substituído pela Equação 8 da regressão linear, que indica o tempo que o sistema leva para atender as requisições. $b = 0.0375$, QR Quantidade de usuários e $a = 1.53$.

Vazão Apache CloudStack:

$$vz = \left(\frac{(E\{\#Processar\})}{(0.0166 \times QR) + 1.22} \right) \quad (14)$$

Onde o denominador indica o tempo que o sistema leva para atender as requisições (Equação 7). $b = 0.0166$, e $a = 1.22$.

5.4 Tempo de Resposta

A métrica do modelo TR é responsável pelo cálculo do tempo de resposta que representa o período entre o envio da requisição para o sistema até a chegada da resposta para o cliente (Barros, et al., 2017). A Tabela 4 apresenta as equações adotadas para cada uma das duas nuvens analisadas.

Tabela 4. Tempo de resposta.

Nuvem	Equação
Apache CloudStack	$TR = \frac{(E\{\#ProcRede1\}) + (E\{\#ProcRede2\}) + (E\{\#UsoMemoria\}) + (E\{\#Processar\}) + (E\{\#Processado\})}{((E\{\#Processar\}) / ((0.0166 \times QR) + 1.22))}$
OpenStack	$TR = \frac{(E\{\#ProcRede1\}) + (E\{\#ProcRede2\}) + (E\{\#UsoMemoria\}) + (E\{\#Processar\}) + (E\{\#Processado\})}{((E\{\#Processar\}) / ((0.0375 \times QR) + 1.53))}$

Fonte: Autores.

Tabela 5. Descrição das equações do Tempo de resposta.

Equação	Descrição
$E\{\#ProcRede1\}$	Número médio de tokens no lugar ProcRede1
$E\{\#ProcRede2\}$	Número médio de tokens no lugar ProcRede2
$(E\{\#UsoMemoria\})$	Número médio de tokens no lugar UsoMemoria
$(\{E\#Processar\})$	Número médio de tokens no lugar Processar
$(\{E\#Processado\})$	Número médio de tokens no lugar Processado
$(0.0166 QR) + 1.22$	Regressão linear do TProcesso para o Apache CloudStack
$(0.0375 QR) + 1.53$	Regressão linear do TProcesso para o OpenStack

Fonte: Autores.

A Tabela 5 apresenta uma descrição das equações adotadas para cada uma das duas nuvens analisadas.

6. Estudo de Caso

Essa seção apresenta os estudos realizados com a finalidade de comparar tanto o desempenho como o consumo de energia do Moodle hospedado no OpenStack e no CloudStack. O primeiro estudo de caso tem como objetivo a validação do modelo SPN proposto. A fim de se atingir essa validação, foram adotadas as seguintes métricas: utilização da CPU, utilização da memória RAM, e, por fim, do consumo de energia para os dois sistemas de nuvem. O estudo de caso 2 apresenta os resultados obtidos da vazão e do tempo de resposta através do modelo proposto. O estudo de caso seguinte tem como objetivo analisar cenários que não foram possíveis de medir experimentalmente. Sendo assim, foi realizado um aumento da quantidade de CPUs (QC) disponíveis ao sistema e também da quantidade de memória RAM (QM). Por fim, no último estudo de caso o objetivo é o de mostrar os resultados obtidos, via modelo, para a utilização da CPU e memória RAM com a variação do tempo de chegada e da quantidade de clientes (QR) no sistema.

6.1 Estudo de Caso I

Esse primeiro estudo de caso tem como objetivo validar o modelo SPN proposto na Seção 5. Para isso, foram coletados dados via medições da utilização da CPU e memória RAM, e do consumo de energia nos dois sistemas de nuvem avaliados. Vale ressaltar que a coleta foi realizada utilizando os *softwares* (JMeter, NMON) e também através de *scripts* de medição. O modelo SPN utilizou como dado de entrada os resultados obtidos nas medições para cada sistema de nuvem (Apache CloudStack e OpenStack). Na avaliação do modelo SPN, foi adotada simulação estacionária, com o nível de confiança de 95% e com o erro relativo da simulação em 10%.

Tabela 6. Parâmetros de entrada do modelo.

QR	Apache CloudStack		OpenStack	
	T _{Lat} (s)	T _{processo} (s)	T _{Lat} (s)	T _{processo} (s)
25	1,01	1,78	1,41	2,33
50	1,08	1,95	2,22	3,78
75	1,21	2,23	2,28	4,01
100	1,58	3,07	2,81	5,38

Fonte: Autores.

A Tabela 5 apresenta os parâmetros de entrada para o modelo SPN proposto. Esse modelo representa o comportamento do Apache CloudStack e do OpenStack, e em ambos os casos, foram adotados 2 CPUs ($QC=2$), 4GB de memória RAM ($QM=4$) e o tempo de chegada entre um cliente e outro ao sistema em 0.1 segundos ($T_{Chegada}=0.1$).

6.1.1 Resultados

Utilização da CPU

A Tabela 6 apresenta um comparativo dos resultados obtidos na medição e os obtidos via simulação do modelo proposto. Essa tabela mostra além dos resultados da medição e da modelagem, os seus respectivos intervalos de confiança, para a utilização da CPU da VM com Apache CloudStack. É possível perceber que os intervalos de confiança dos dados medidos se sobrepõem aos valores estimados pelo modelo. Por exemplo, assumindo 25 clientes, o intervalo de confiança medido foi [83.03; 90.30], e o obtido com o modelo foi [85.39; 88.45]. Sendo assim, o valor estimado pelo modelo se encontra dentro do quantificado no sistema com a aferição. É importante destacar que todas as medições foram executadas, no mínimo, 30 vezes a fim de aumentar a confiabilidade dos dados coletados.

Tabela 7. Resultados para validação do modelo com a utilização da CPU da VM com Apache CloudStack.

QR	Medição		Modelagem	
	CPU	Int. Conf.	CPU	Int. Conf.
25	86,67	[83,03; 90,30]	87,46	[85,39; 88,45]
50	92,13	[91,49; 92,17]	92,11	[90,60; 93,78]
75	92,26	[90,66; 93,66]	93,40	[90,47; 93,57]
100	94,55	[93,84;95,26]	95,18	[89,88; 96,33]

Fonte: Autores.

Similarmente, a Tabela 7 mostra os resultados para a utilização da CPU da VM com OpenStack. Também é possível observar nessa tabela que os intervalos de confiança dos valores medidos se sobrepõem aos valores dos dados estimados pelo modelo.

Tabela 8. Resultado para validação do modelo com uso da CPU da VM OpenStack.

QR	Medição		Modelagem	
	CPU	Int. Conf.	CPU	Int. Conf.
25	83,45	[81,02; 85,87]	82,51	[81,69; 83,52]
50	88,49	[87,24; 89,74]	88,10	[87,76; 88,45]
75	91,01	[89,15; 92,92]	90,37	[88,74; 92,00]
100	95,21	[94,48; 95,95]	95,08	[92,62; 97,33]

Fonte: Autores.

Utilização da memória RAM

A Tabela 8 apresenta o mesmo comparativo anterior levando em consideração a utilização da memória RAM no Apache CloudStack. É possível perceber que os intervalos de confiança se sobrepõem, ou seja, também foi possível realizar a validação para este cenário da utilização da memória RAM no CloudStack.

Tabela 9. Resultado para validação do modelo com utilização da memória RAM Apache CloudStack.

QR	Medição		Modelagem	
	Memória RAM (%)	Int. Conf.	Memória RAM (%)	Int. Conf.
25	61,20	[57,62; 64,78]	65,46	[62,42; 71,38]
50	66,70	[64,11; 67,34]	72,04	[65,93; 76,52]
75	70,67	[68,48; 72,91]	79,71	[71,77; 79,73]
100	80,14	[76,16; 84,11]	89,52	[80,89; 89,74]

Fonte: Autores.

Esse mesmo procedimento foi repetido levando em consideração o OpenStack. Conforme pode ser visto na Tabela 9, os intervalos de confiança dos dados medidos e estimados pelo modelo também se sobrepõem. Sendo assim, pode-se afirmar que o modelo também representa o sistema para o cenário de utilização da memória RAM.

Tabela 10. Resultado para validação do modelo com a utilização da memória RAM OpenStack.

QR	Medição		Modelagem	
	Memória RAM (%)	Int. Conf.	Memória RAM (%)	Int. Conf.
25	49,20	[45,53; 52,15]	54,44	[50,86; 56,97]
50	53,66	[52,28; 59,03]	63,01	[58,97; 64,92]
75	69,71	[66,70; 71,63]	73,45	[70,65; 75,93]
100	94,26	[90,21; 94,50]	90,03	[87,03; 91,16]

Fonte: Autores.

Consumo de energia

Os resultados comparativos das duas nuvens privadas para o consumo de energia entre os dados obtidos via modelo e os auferidos no ambiente real podem ser vistos nas Tabelas 10 e 11. Na primeira tabela, os resultados são do CloudStack e é possível verificar que os dados medidos e estimados também se sobrepõem. O mesmo comportamento acontece com os dados obtidos para o OpenStack (ver Tabela 11).

Tabela 11. Resultado para validação do modelo com o consumo de energia (W) Apache CloudStack.

QR	Medição		Modelagem	
	Cons. Energia (W)	Int. Conf.	Cons. Energia (W)	Int. Conf.
25	67,20	[57,62; 67,78]	65,58	[65,52; 67,87]
50	68,71	[68,12; 68,74]	68,19	[67,38; 68,99]
75	68,72	[68,53; 68,95]	68,76	[66,53; 68,80]
100	69,06	[68,32; 69,06]	69,70	[69,02; 70,20]

Fonte: Autores.

Tabela 12. Resultado para validação do modelo com o consumo em de energia (W) OpenStack.

QR	Medição		Modelagem	
	Cons. Energia (W)	Int. Conf.	Cons. Energia (W)	Int. Conf.
25	65,14	[60,86; 67,28]	67,22	[64,08; 67,36]
50	68,43	[66,80; 69,72]	68,18	[68,12; 68,84]
75	69,85	[65,64; 70,98]	68,57	[68,29; 68,85]
100	70,55	[69,35; 71,16]	69,36	[68,96; 69,77]

Fonte: Autores.

Sendo assim, pode-se destacar que esse estudo de caso comprovou que o modelo SPN proposto representa o sistema analisado quanto ao consumo de energia, utilização de CPU e memória RAM.

6.2 Estudo de Caso II

Uma vez realizada a validação do modelo SPN, os estudos apresentados a seguir têm como objetivo ilustrar a aplicabilidade desse modelo proposto. Em particular, essa seção apresenta os resultados obtidos para a vazão e o tempo de resposta. Sendo assim, comparações do tempo de resposta e da vazão obtidas são utilizadas para comparar o desempenho dos dois ambientes, Apache CloudStack e OpenStack.

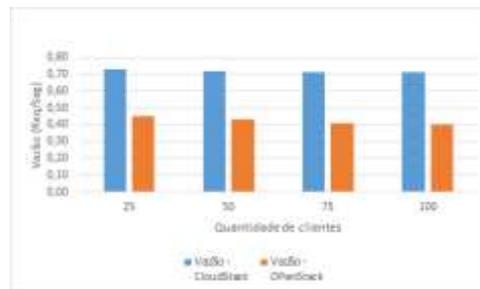
A configuração utilizada na VM é a mesma do estudo de caso anterior e esse estudo também adota os mesmos parâmetros de entrada para o modelo proposto (ver Tabela 5). A configuração adotada no modelo SPN (ver Seção 5) foi a seguinte 2 CPUs ($QC=2$), 4 GB de memória RAM ($QM=4$) e o tempo de chegada entre um cliente e outro ao sistema foi de 0.1 segundos ($TChegada=0.1$).

6.2.1 Resultados

Vazão

A Figura 13 mostra um gráfico comparativo dos resultados para a vazão entre o Apache CloudStack e OpenStack que foram obtidos via simulação do modelo proposto. Nessa figura, é possível observar que a vazão do Apache CloudStack apresenta um desempenho superior. Além disso, é possível observar que a vazão de cada um desses sistemas permaneceu praticamente constante. Por exemplo, assumindo o CloudStack, pode-se dizer que a vazão foi praticamente 0.70 requisições (ou clientes) atendidos por segundo, ao passo que para o OpenStack foi de 0.42 requisições por segundo. Logo, estes resultados mostram que o Apache CloudStack possui uma vazão superior quando comparado com o outro sistema em estudo. É importante destacar, ainda, que esse comportamento da vazão praticamente constante ocorreu em virtude do limite do sistema. No entanto, pode-se perceber, no tempo de resposta mostrado a seguir, que com mais clientes, o tempo demandado para fazer as mesmas operações foi bem superior.

Figura 13. Resultado da simulação da vazão Apache CloudStack e OpenStack.

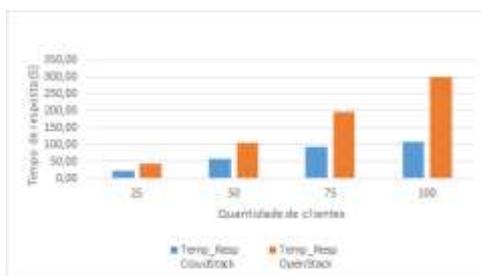


Fonte: Autores.

Tempo de resposta

A Figura 14 mostra o resultado da simulação e do tempo de resposta para os sistemas Apache CloudStack e OpenStack utilizando o modelo proposto. Conforme esperado, o tempo de resposta no sistema acaba aumentando à medida que são adicionados mais clientes ao sistema. A partir desse gráfico, é possível perceber também que o sistema OpenStack apresenta um tempo de resposta significativamente maior quando comparado com o tempo de resposta do CloudStack. Esse comportamento se deve pelo fato de que o OpenStack possui uma vazão média bem inferior, conforme mostrado anteriormente na Figura 13.

Figura 14. Resultado da simulação do tempo de resposta Apache CloudStack e OpenStack.



Fonte: Autores.

6.3 Estudo de Caso III

Esse terceiro estudo tem como objetivo mostrar o comportamento do desempenho dos sistemas ao serem realizadas variações da quantidade de recursos presentes para a CPU e memória RAM. As simulações foram realizadas a partir do modelo SPN da Figura 8.

Vários cenários foram definidos com a variação do número de CPUs (QC) e da quantidade de memória RAM disponível no sistema (QM). A Tabela 12 mostra os cenários que foram adotados. Por exemplo, com $QC=2$ foram realizados experimentos com a memória variando de 4, 8, 16, 32 e 64 GB. Esses mesmos cenários foram repetidos assumindo um maior número de CPUs, valores de $QC= 4, 6, e 8$.

Tabela 13. Parâmetros de variação da CPU (QC) e RAM (QM).

QC	QM				
	4	8	16	32	64
2	4	8	16	32	64
4	4	8	16	32	64
6	4	8	16	32	64
8	4	8	16	32	64

Fonte: Autores.

A Tabela 5 mostra outros parâmetros de entrada adotados nesse estudo. Além desses, foram considerados como tempo de chegada entre um cliente e outro ao sistema 0.1 segundos ($T_{Chegada}=0.1$) e a quantidade de clientes ($QR = 100$).

6.3.1 Resultados

A Tabela 13 apresenta um quadro resumo dos resultados obtidos para todos os cenários analisados neste estudo. Nessa tabela é possível perceber que a adição de memória RAM não impacta, significativamente, na utilização da CPU. No entanto, à medida que foi disponibilizado mais recurso de memória (QM), ocorreu uma redução do percentual da memória RAM utilizada. Essas constatações valem para todos os cenários analisados.

Assumindo $QC = 4$, pode-se perceber que ocorreu uma redução representativa na utilização da CPU em relação a $QC = 2$. Por exemplo, considerando o cenário com CloudStack e com $QM = 4$, de $QC = 2$ para $QC = 4$ houve uma redução de 95.18% para 48.52%. Assumindo os mesmos cenários, mas considerando o OpenStack, a redução da utilização da CPU foi de 96.35% para 47.53%. Redução similar também ocorreu em relação a memória RAM, onde assumindo $QM = 4$, o aumento do número de $QC = 2$ para $QC = 4$ fez com que a memória RAM tivesse uma redução de 89.52% para 48.52% no CloudStack, e

de 91.75% para 48.43% no OpenStack. Comparando $QC = 4$ e $QC = 6$, pode-se observar que a redução em relação a utilização da memória RAM ocorrido de $QC = 2$ para $QC = 4$ não foi mais observada.

Tabela 14 Resumo dos resultados obtidos.

QC	QM	CloudStack		OpenStack	
		CPU (%)	Memória (%)	CPU (%)	Memória (%)
2	4	95,18	89,52	96,35	91,75
2	8	97,43	89,94	95,32	83,77
2	16	96,90	78,23	95,41	71,96
2	32	97,26	65,94	95,21	52,00
2	64	96,98	42,68	95,38	31,42
4	4	48,52	48,52	47,53	48,43
4	8	48,37	25,99	47,62	24,41
4	16	48,56	13,10	47,72	12,85
4	32	48,63	6,55	47,85	6,48
4	64	48,68	3,28	47,84	3,18
6	4	32,28	48,42	32,10	48,23
6	8	32,18	24,21	32,20	24,24
6	16	32,40	13,48	32,00	12,00
6	32	32,45	6,10	32,12	06,05
6	64	32,41	03,05	32,05	03,03
8	4	24,36	48,73	23,79	47,50
8	8	24,07	24,07	23,80	23,80
8	16	24,81	12,40	23,81	11,97
8	32	24,18	06,04	23,80	5,93
8	64	24,10	03,01	24,10	2,95

Fonte: Autores.

Esse mesmo comportamento ocorreu também para $QC = 8$. Por exemplo, assumindo $QC = 8$ e $QM = 4$, para o CloudStack, obteve-se 48.73% de utilização da memória RAM; considerando $QC = 4$ e $QM = 4$, obteve-se 48.52% de utilização da memória, mostrando assim que não ocorreu mais redução na utilização da memória ao se incrementar mais o número de CPUs. No entanto, o aumento no número de CPUs faz com que se reduza, em média, o nível de utilização das CPUs. Por exemplo, assumindo $QM = 4$ e comparando $QC = 2$ e $QC = 8$, para o CloudStack, pode-se observar uma redução na utilização da CPU de 95.18% para 24.36%.

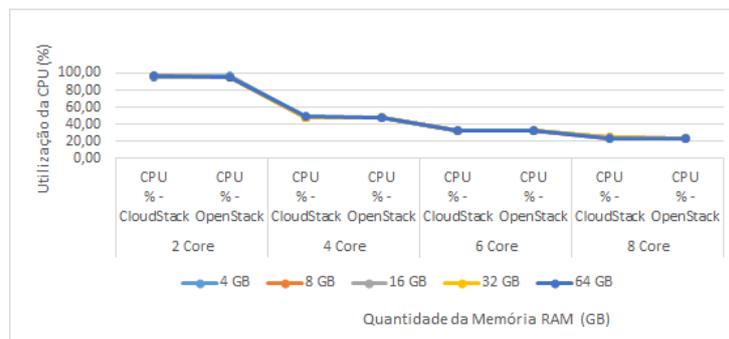
Comparando os Resultados do OpenStack vs Apache CloudStack

A Figura 15 mostra uma comparação da quantidade de CPUs utilizadas e o seu impacto na utilização média das CPUs tanto no OpenStack como no Apache CloudStack. Pode-se perceber que o aumento da quantidade de CPUs (cores) disponíveis, nos dois sistemas em nuvem, ocasionou uma redução significativa na utilização média das CPUs. Por exemplo, considerando o cenário com 2 CPUs ($QC = 2$) e 64GB de memória RAM ($QM = 64$), em comparação com o de 8 CPUs, obteve-se um resultado para o OpenStack saindo de 95.38% para 24.10% de utilização e, assumindo o CloudStack, a redução foi de 96.98% para 24.10%.

É importante ressaltar que a variação da memória RAM e o impacto no nível de utilização da CPU não foram tão representativos. Sendo assim, os valores acabaram se sobrepondo a este gráfico da Figura 15.

Outro fator que também é relevante é a comparação da média de utilização das CPUs considerando os diferentes cenários. Por exemplo, assumindo o cenário com 2 CPUs ($QC = 2$), para o Apache CloudStack, a média de utilização das CPUs foi de 96.75%.

Figura 15. Resultado da simulação da utilização da CPU Apache CloudStack e OpenStack.

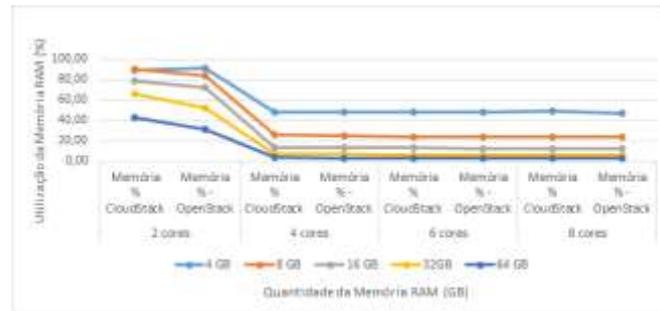


Fonte: Autores.

Esse valor médio leva em consideração os experimentos com diferentes quantidades de memória RAM (4, 8, 16, 32 e 64GB). Já no cenário com 8 CPUs, a utilização média da CPU foi de 24,30%. Assumindo o OpenStack, o valor médio de utilização das CPUs para o cenário com 2 CPUs ($QC = 2$) foi de 95.93%. Com 8 CPUs, a utilização média das CPUs reduziu para 23.86%.

A Figura 16 mostra o resultado do impacto da variação da quantidade de CPUs na utilização da memória RAM, tanto para o OpenStack como para o CloudStack. É possível perceber que o aumento da quantidade de CPUs ocasionou uma redução na utilização da memória RAM apenas no aumento de 2 para 4 cores de CPU. Nas demais situações, o aumento da quantidade de cores no sistema acabou não reduzindo o percentual de utilização da memória RAM. Ainda neste gráfico, é possível destacar que à medida que se aumenta a quantidade de memória RAM, nos dois sistemas em nuvem, o nível de utilização da memória RAM teve uma redução.

Figura 16. Resultado da simulação da utilização da Memória RAM Apache CloudStack e OpenStack



Fonte: Autores.

É importante evidenciar que a média de utilização da memória RAM com 2 CPUs, para o Apache CloudStack, foi de 73,26% e a média da utilização das CPUs com 8 cores de CPU foi de 24,30%. Esse valor médio se refere aos valores com o número de cores fixo, variando somente a memória RAM (4, 8, 16, 32 e 64GB). Já com o OpenStack, assumindo 8 CPUs, a média foi de 18,43% de utilização de memória RAM. Para o OpenStack, a média de utilização da memória RAM com 2 CPUs foi de 66,18% e a média para 8 CPUs foi de 18,43%. Com base nos dados apresentados, pode-se concluir que o sistema OpenStack acabou apresentando, para os cenários analisados, um menor nível de utilização tanto para a memória RAM quanto para a CPU.

6.4 Estudo de Caso IV

Este último estudo tem como objetivo mostrar o impacto da variação da taxa de chegada de requisições (clientes) ao sistema nos resultados obtidos para a utilização da CPU e memória RAM. Para isso, foi utilizado o modelo SPN de desempenho proposto, onde foram feitas comparações aplicando variações no tempo de chegada dos clientes.

Os cenários adotados neste experimento consideram 25, 50, 75 e 100 clientes, ao mesmo tempo, realizando requisições ao sistema. Para cada um desses cenários, adotou-se tempos de chegada diferentes (0,01, 0,05, 0,1, 0,2, 0,3, 0,4, 0,5, 1,0 s). Esses experimentos foram realizados tanto no Apache CloudStack, como no OpenStack. Para cada simulação, utilizamos o modelo SPN com os parâmetros de latência e tempo de processamento para cada sistema em nuvem, conforme mostrado na Tabela 5.

6.4.1 Resultados

A Tabela 14 apresenta um resumo comparativo dos resultados obtidos em que foi evidenciado que a variável Tempo de Chegada (T_1) gera impacto na utilização da CPU. Ao realizar a comparação da média de utilização da CPU em relação ao tempo de chegada, pode-se visualizar que, com um tempo $T_1=0,01s$, obtém-se uma média de utilização da CPU em 91,91% para o CloudStack, e de 88,79% para o OpenStack. Já assumindo um $T_1=1,0s$, tem-se a utilização da CPU média em 90,43% para o CloudStack, e de 86,14% para o OpenStack. Comportamento similar também é obtido a esses considerando as demais quantidades de clientes enviando requisições ao sistema simultaneamente.

Na Tabela 14 é possível observar que, para ambos os sistemas em nuvem, quando ocorre um aumento da frequência de chegada de clientes ao sistema, mesmo limitando em 25, 50, 75, ou 100 clientes, acaba por também aumentar a utilização da CPU. Por exemplo, considerando 25 clientes e o tempo de chegada $T_1=0,01s$, tem-se a utilização da CPU em 88% para o CloudStack e em 82,78% para o OpenStack. Considerando o mesmo tempo de chegada, e ao aumentar o número máximo de clientes simultâneos realizando requisições ao sistema para 100, temos a utilização da CPU em 96,92% para o CloudStack e em 97,24% para o OpenStack.

Tabela 15. Resultados da utilização da CPU. Op=OpenStack e Cl=CloudStack.

T1(s)	25 Clientes		50 Clientes		75 Clientes		100 Clientes		Média	
	Op (%)	Cl (%)	Op(%)	Cl (%)	Op (%)	Cl (%)	Op (%)	Cl (%)	Op (%)	Cl (%)
0,01	82,78	88,00	85,74	89,79	89,41	92,91	97,24	96,92	88,79	91,91
0,05	81,62	87,76	85,02	89,87	88,22	92,71	96,15	96,98	87,75	91,83
0,10	81,39	87,65	84,87	89,80	87,69	92,55	95,70	96,25	87,41	91,56
0,20	81,37	87,74	84,90	89,70	87,66	91,91	95,66	96,21	87,40	91,39
0,30	81,34	87,72	84,74	89,66	87,65	91,35	95,35	95,78	87,27	91,13
0,40	81,19	87,51	84,35	89,64	87,49	91,29	95,21	95,66	87,06	91,03
0,50	81,15	87,43	84,27	89,55	86,84	91,17	95,02	95,64	86,82	90,95
1,00	80,10	86,24	83,93	89,07	85,80	90,55	94,73	95,86	86,14	90,43
Média	81,37	87,51	84,73	89,64	87,60	91,81	95,63	96,16	87,33	91,28

Fonte: Autores.

Assumindo a utilização da memória RAM, a Tabela 15 apresenta os resultados desse mesmo experimento ressaltando os impactos na utilização da memória. Ao se comparar a média da utilização em relação ao tempo de chegada, é possível visualizar que, para um menor tempo $T_1=0.01s$, a utilização média da memória RAM foi 84.84 para o CloudStack e de 79.58% para o OpenStack. Já assumindo o $T_1=1.0s$, temos a utilização em 82.46% para CloudStack e de 76.10% para o OpenStack. Sendo assim, pode-se perceber que, ao se espaçar a chegada de clientes ao sistema, tem-se uma redução na utilização tanto da CPU como da memória RAM.

Tabela 16. Resultados da utilização da memória RAM. Op=OpenStack e Cl=CloudStack.

T1(s)	25 Clientes		50 Clientes		75 Clientes		100 Clientes		Média	
	Op (%)	Cl (%)	Op (%)	Cl (%)	Op (%)	Cl (%)	Op (%)	Cl (%)	Op (%)	Cl (%)
0,01	71,21	77,91	73,27	81,00	79,56	86,30	94,26	94,14	79,58	84,84
0,05	70,00	77,56	72,79	80,50	78,41	85,80	92,33	94,08	78,38	84,49
0,1	69,10	77,50	73,13	80,45	77,84	85,05	91,62	92,85	77,92	83,96
0,2	68,65	76,42	72,53	80,36	77,76	85,25	91,51	92,00	77,61	83,51
0,3	67,55	77,25	72,38	80,27	77,64	84,77	90,90	91,64	77,12	83,48
0,4	67,53	77,24	72,08	80,17	77,34	83,86	90,88	91,62	76,96	83,22
0,5	67,25	77,01	71,51	80,10	76,21	83,40	90,17	91,54	76,29	83,01
1,0	67,12	75,22	71,21	79,74	76,06	83,00	90,00	91,86	76,10	82,46
Média	68,55	77,01	72,36	80,32	77,6	84,68	91,46	92,47	77,49	83,62

Fonte: Autores.

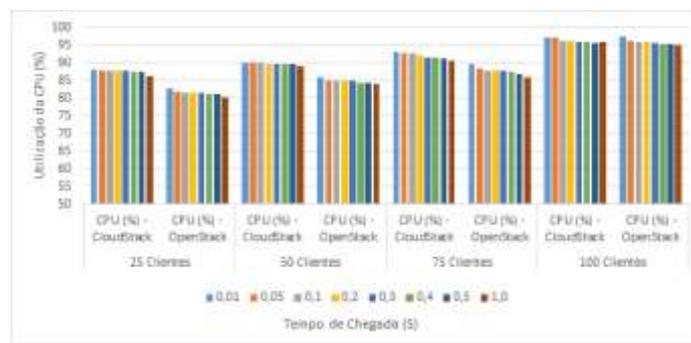
Similarmente à utilização da CPU, um espaçamento maior do tempo de chegada de clientes ao sistema faz com que se reduza o nível de utilização da memória RAM. Por exemplo, assumindo 100 clientes, com o $T_1 = 0.01$ se tem uma utilização da memória acima de 94% para ambas as nuvens. Já para o $T_1=1.0$ e com o mesmo número de clientes no sistema, se obteve uma redução para 91.86% para o CloudStack e 90% para o OpenStack na utilização da memória RAM.

É possível notar que uma redução da utilização da memória ocorre sempre que se espaça a chegada de clientes ao sistema. Além disso, o aumento da quantidade de clientes no sistema realizando requisições simultaneamente também acaba por impactar em um aumento na utilização da memória RAM.

Comparação Apache CloudStack vs OpenStack

A Figura 17 mostra um comparativo dos resultados obtidos para utilização da CPU no ambiente Apache CloudStack vs OpenStack. O aumento do tempo de chegada (T_1) e da quantidade de clientes (QR) no sistema acabam por impactar na utilização da CPU. Ficou evidenciado que, com o aumento da quantidade de clientes, há uma maior utilização da CPU. Já quando se aumenta o espaçamento entre a chegada de clientes, ocorre uma redução da utilização da CPU. Esse mesmo comportamento ocorre para os dois sistemas de nuvem.

Figura 17. Resultado da simulação da utilização da CPU do Apache CloudStack e OpenStack.

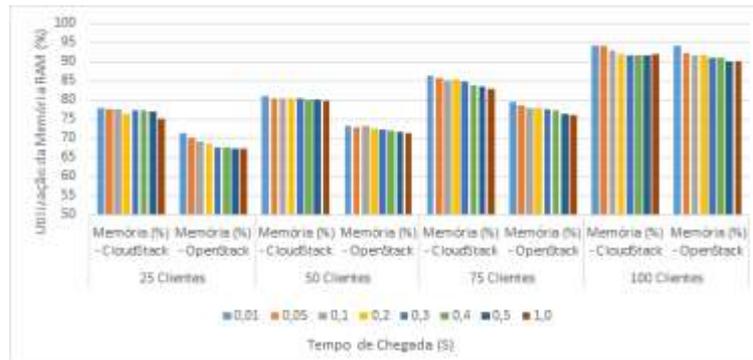


Fonte: Autores.

A Figura 18 ilustra os resultados obtidos comparando o Apache CloudStack e o OpenStack para a utilização da memória RAM. O aumento do tempo de chegada (T_1) e da quantidade de clientes (QR) impactam na utilização da memória RAM. Por exemplo, nessa figura é possível perceber que o aumento da quantidade de clientes no sistema ocasiona uma maior utilização da memória RAM. Além disso, assumindo uma redução da frequência da chegada de clientes ao sistema, ocorre uma redução no nível de utilização da memória. Esse mesmo padrão de comportamento ocorreu nos dois sistemas em nuvem analisados.

A partir dos resultados obtidos, ficou evidente que a utilização da CPU para o sistema Apache CloudStack foi em média 3,95% maior que a do OpenStack. Considerando a utilização da memória RAM, o OpenStack apresentou um menor consumo, 6,13% em média inferior quando comparado ao CloudStack. Sendo assim, nos experimentos realizados, o OpenStack apresentou melhor eficiência na utilização da CPU e utilização da memória RAM.

Figura 18 Resultado da simulação da utilização da Memória RAM do Apache CloudStack e OpenStack.



Fonte: Autores.

7. Conclusão

A crescente demanda pelos ambientes para suporte ao ensino e aprendizagem online acaba por fomentar a necessidade de se avaliar tanto o desempenho como o consumo de energia decorrente de tais sistemas. Essas análises têm como objetivo o de que esses sistemas sejam capazes de atender as demandas de forma satisfatória e sem um alto impacto no custo energético.

Nesse contexto, o presente trabalho propôs uma solução integrada composta de uma metodologia e modelo em SPN para auxiliar na escolha entre nuvens privadas distintas, por exemplo, entre o Apache CloudStack e o OpenStack. Sendo assim, o Moodle foi hospedado em cada uma dessas nuvens a fim de se poder avaliar o desempenho e o consumo de energia a partir de diversos cenários analisados.

A estratégia proposta mostrou que pode auxiliar na escolha do sistema em nuvem privada. Ficou evidenciado, por exemplo, que o sistema Apache CloudStack obteve um melhor desempenho em relação ao OpenStack para as métricas analisadas. Por exemplo, o Apache CloudStack demandou um tempo 49.46% inferior, e o seu consumo de energia foi 28.96% inferior. Assumindo a utilização da CPU, houve um ganho de 1.79%; e, por fim, a vazão foi superior em 59.06%. Como futuro direcionamento desta pesquisa, tem-se que um estudo integrado de desempenho e disponibilidade pode vir a ser realizado para complementar os resultados já obtidos.

Agradecimentos

Os autores agradecem ao CNPq, CAPES, UFRPE e à FACEPE pelo apoio na realização desta pesquisa.

Referências

- Assavakamhaenghan, N., Suwanworaboon, P., Tanaphantaruk, W., Tuarob, S., & Choetkiertikul, M. (2020). Towards Team Formation in Software Development: A Case Study of Moodle. In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 157-160). IEEE.
- Barros, J., Callou, G., & Gonçalves, G. (2017). Análise Integrada de Desempenho e Consumo de Energia em Sistemas de Armazenamento de Dados Distribuídos. In Anais do XV Workshop em Clouds e Aplicações. SBC.
- Borba, E. R. (2017). Modelagem de desempenho e disponibilidade para sistemas de armazenamento híbridos (Master's thesis, Universidade Federal de Pernambuco).
- Bruschi, G. C., Spolon, R., Pauro, L. L., Lobato, R. S., Manacero, A., & Cavenaghi, M. A. (2016). Stackact: Performance evaluation in an iaaS cloud multilayer. In 2016 15th International Symposium on Parallel and Distributed Computing (ISPD) (pp. 149-156). Ieee.
- Callou, G., Maciel, P., Tavares, E., Andrade, E., Nogueira, B., Araujo, C., & Cunha, P. (2011). Energy consumption and execution time estimation of embedded system applications. *Microprocessors and Microsystems*, 35(4), 426-440.

- da Silva, A. V. G., de Lima, C. J., & de Almeida Callou, G. R. (2019). Análise de desempenho do ambiente virtual de aprendizagem na nuvem privada apache cloudstack. *GESTÃO. Org*, 17(8), 120-133.
- dos Santos, D. M. L., Vale, K. M. A. C., & de Alencar, F. M. R. (2020). Nextcloud e Pydio. *Brazilian Journal of Development*, 6(6), 40549-40566.
- Dantas, J. R. (2018). Planejamento de infraestrutura de nuvens computacionais para serviço de VoD streaming considerando desempenho, disponibilidade e custo.
- Fe, I., Matos, R., Dantas, J., Melo, C., & Maciel, P. (2017). Stochastic model of performance and cost for auto-scaling planning in public cloud. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2081-2086). IEEE. DOI 10.1109/SMC.2017.8122926
- German, R. (2000). Performance analysis of communication systems with non-Markovian stochastic Petri nets. John Wiley & Sons, Inc.
- Kumar, R., Jain, K., Maharwal, H., Jain, N., & Dadhich, A. (2014). Apache cloudstack: Open source infrastructure as a service cloud computing platform. *Proceedings of the International Journal of advancement in Engineering technology, Management and Applied Science*, 111, 116.
- Li, Y., Yu, M., & Hu, Y. (2019). Modeling and Analysis of Classroom Teaching Evaluation Platform Based on Colored Petri Nets. In 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC) (pp. 442-445). IEEE.
- Lilja, D. J. (2005). *Measuring computer performance: a practitioner's guide*. Cambridge university press.
- Machida, F., Andrade, E., Kim, D. S., & Trivedi, K. S. (2011). Candy: Component-based availability modeling framework for cloud service management using sysml. In 2011 IEEE 30th International Symposium on Reliable Distributed Systems (pp. 209-218). IEEE.
- Mansur, A. F. U., Gomes, S. S., Lopes, A. D. A., & Biazus, M. C. B. (2010). Novos rumos para a Informática na Educação pelo uso da Computação em Nuvem (Cloud Education): Um estudo de Caso do Google Apps. In Foz do Iguaçu: Anais do XVI Congresso Internacional ABED de Educação a Distância (p. 35). sn.
- Maron, C. A., Griebler, D., Vogel, A., & Schepke, C. (2014). Avaliação e Comparação do Desempenho das Ferramentas OpenStack e OpenNebula. 12th Escola Regional de Redes de Computadores (ERRC), 1-5.
- Matos, M. A. (1995). Manual operacional para a regressão linear. Faculdade de Engenharia da Universidade do Porto, 63.
- Melo Filho, I. J., Carvalho, R. S., Gomes, A. S., da Rocha Perris, P. A., Rodrigues, R. L., & de Oliveira Feliciano, F. D. (2014). Análise comparativa da usabilidade dos ambientes de gestão da aprendizagem Amadeus e Moodle. *Revista Brasileira de Informática na Educação*, 22(01), 107.
- Mendonça Neto, J. R. D. (2015). Modelagem e análise de desempenho e consumo de energia em aplicações móveis (Master's thesis, Universidade Federal de Pernambuco).
- Merlin, P., & Farber, D. (1976). Recoverability of communication protocols-implications of a theoretical study. *IEEE transactions on Communications*, 24(9), 1036-1043.
- Moodle. (2020). moodle.org. <https://moodle.org/>
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 541-580.
- Nist. (2014). Nist cloud computing program. <http://www.nsti.gov/itl/cloud/index.cfm>
- Pauro, L. L. (2016). Auditoria e monitoramento de eventos inconsistentes em instâncias de máquinas virtuais em IaaS no Orquestrador Apache CloudStack.
- Peixoto, M. L. M. (2012). Oferecimento de QoS para computação em nuvens por meio de metaescalonamento (Doctoral dissertation, Universidade de São Paulo).
- Petri, C. A. (1962). Kommunikation mit automaten.
- Rakic, S., Pavlovic, M., Softic, S., Lalic, B., & Marjanovic, U. (2019). An evaluation of student performance at e-learning platform. In 2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA) (pp. 681-686). IEEE.
- Ratna, A. A. P., Wirianata, T., Ekadiyanto, F. A., Ibrahim, I., Husna, D., & Purnamasari, P. D. (2017). Cloud computing network design for high performance computing implementation on openstack platform. In Proceedings of the 3rd International Conference on Communication and Information Processing (pp. 356-360).
- Sabharwal, N., & Shankar, R. (2013). Apache CloudStack cloud computing: leverage the power of CloudStack and learn to extend the CloudStack environment. Packt Publ..
- Silva, S. (2012). Acessibilidade digital em ambientes virtuais de aprendizagem. *Revista GEINTEC-Gestão, Inovação e Tecnologias*, 2(3), 245-254.
- Sousa, E., Lins, F., Tavares, E., & Maciel, P. (2017). Cloud infrastructure planning considering different redundancy mechanisms. *Computing*, 99(9), 841-864.
- SOUSA, E. T. G. D. (2015). Modelagem de desempenho, dependabilidade e custo para o planejamento de infraestruturas de nuvens privadas.
- Torres, E., Callou, G., Alves, G., Accioly, J., & Gustavo, H. (2016). Storage services in private clouds: Analysis, performance and availability modeling. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 003288-003293). IEEE.
- Torres, E., Callou, G., & Andrade, E. (2018). A hierarchical approach for availability and performance analysis of private cloud storage services. *Computing*, 100(6), 621-644. <https://doi.org/10.1007/s00607-018-0588-7>

Vogel, A., Griebler, D., Maron, C. A., Schepke, C., & Fernandes, L. G. (2016). Private IaaS clouds: a comparative analysis of OpenNebula, CloudStack and OpenStack. In 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP) (pp. 672-679). IEEE.

Vogel, A., Griebler, D., Schepke, C., & Fernandes, L. G. (2017). An intra-cloud networking performance evaluation on cloudstack environment. In 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP) (pp. 468-472). IEEE.

Vogel, A., Griebler, D., Maron, C. A., Schepke, C., & Fernandes, L. G. (2016). Private IaaS clouds: a comparative analysis of OpenNebula, CloudStack and OpenStack. In 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP) (pp. 672-679). IEEE.

Wang, Q., Xu, P., Zhang, Y., & Chu, X. (2017). EPPMiner: An extended benchmark suite for energy, power and performance characterization of heterogeneous architecture. In Proceedings of the Eighth International Conference on Future Energy Systems (pp. 23-33).

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1), 7-18.

Zota, R., & PETRE, I. A. (2014). An Overview of the Most Important Reference Architectures for Cloud Computing. *Informatica Economica*, 18(4).