



Alison Vinicius Gomes da Silva

Análise de Desempenho e de Disponibilidade do Ambiente Virtual de Aprendizagem na Nuvem Privada Apache CloudStack

Recife

2019

Alison Vinicius Gomes da Silva

**Análise de Desempenho e de Disponibilidade do
Ambiente Virtual de Aprendizagem na Nuvem Privada
Apache CloudStack**

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Computação
Curso de Bacharelado em Ciências da Computação

Orientador: Prof. Dr. Gustavo Rau de Almeida Callou

Recife
2019



MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por ALISON VINICIUS GOMES DA SILVA às 14 horas do dia 05 de julho de 2019, na sala 27, 2º andar CEAGRI II, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado " **Análise de Desempenho e de Disponibilidade do Ambiente Virtual de Aprendizagem na Nuvem Privada Apache CloudStack** ", orientado por Gustavo Rau de Almeida Callou e aprovado pela seguinte banca examinadora:

Gustavo Rau de Almeida Callou
DC/UFRPE

Carlos Julian Menezes de Araujo
DC/UFRPE

*Dedico este trabalho a minha avó Denise da Silva (in memoriam) e a meu amigo
Rafael Paiva (in memoriam).*

Agradecimentos

Primeiramente queria agradecer a toda minha família. Em especial a minha filha Clara, minha esposa Stephanie, minha mãe Edilene, meu pai Reginaldo e minha irmã Letícia por sempre estar ao meu lado sempre.

Agradeço ao professor Gustavo Callou pela oportunidade e a apoio na condução deste trabalho.

Agradeço ao mestrando Claudemir Lima pela a colaboração nesse trabalho.

Agradeço aos meus amigos de turma Diogo Belém, Emanuel Leite e Tiago Falcão que estiveram comigo nesta jornada.

E por último agradeço a servidora Sandra Xavier pelos conselhos e o apoio.

*"Nobody said it was easy."
(Chris Martin)*

Resumo

A computação em nuvem é um paradigma que oferece recursos computacionais de forma dinâmica por meio da Internet. As universidades e escolas vêm adotando cada vez mais os ambientes virtuais de aprendizagem para facilitar a comunicação entre alunos e professores e, assim, existe uma necessidade de uma avaliação de desempenho do ambiente virtual de aprendizagem em *softwares* de computação em nuvem. Este trabalho realiza a avaliação de desempenho do Ambiente Virtual de Aprendizagem Moodle em nuvem privada. Foi proposta uma metodologia para a realização das medições e modelagem do sistema. Essa metodologia faz uso do modelo de desempenho proposto em rede de Petri estocástica para avaliar métricas de tempo de resposta e vazão. Além dos modelos de desempenho, modelos de disponibilidade foram feitos em diagrama de blocos de confiabilidade e em rede de Petri estocástica. Este trabalho apresenta quatro estudos de casos realizados para ilustrar a aplicabilidade da metodologia e dos modelos propostos em um ambiente real, montado em laboratório, com o Moodle configurado em uma nuvem privada com Apache CloudStack. O primeiro estudo de caso avalia a capacidade do sistema utilizando as métricas de vazão e tempo de resposta em uma máquina virtual com duas configurações distintas. O segundo estudo de caso avalia as medições das métricas de utilização de CPU, utilização de memória e taxa de escrita do HD para observação do comportamento do sistema durante os testes. O terceiro estudo de caso valida o modelo de desempenho em SPN. E o último estudo de caso avalia o impacto da disponibilidade no modelo de desempenho. Os resultados mostram que o modelo de desempenho foi prático e eficiente para quantificar as métricas de interesse.

Palavras-chave: Avaliação de Desempenho, Ambiente Virtual de Aprendizagem, Computação em Nuvem, Rede de Petri Estocástica, Bloco de Diagrama de Confiabilidade.

Abstract

Cloud computing is a paradigm that offers computational resources dynamically through the Internet. Universities and schools are increasingly adopting virtual learning environments to facilitate communication between students and teachers, and thus there is a need for a performance assessment of the virtual learning environment in cloud computing software. This work evaluates the performance of the Virtual Learning Environment Moodle in private cloud. A methodology for the measurement and modeling of the system was proposed. This methodology makes use of the performance model proposed in Stochastic Petri net to evaluate response time and flow metrics. In addition to the performance models, availability models were made in block diagram of reliability and stochastic Petri net. This paper presents four case studies to illustrate the applicability of the methodology and proposed models in a real laboratory-based environment with Moodle configured in a private cloud with Apache CloudStack. The first case study assesses system capacity using throughput and response time metrics in a virtual machine with two different configurations. The second case study evaluates the measurements of metrics CPU utilization, memory usage, and write rate of the HD to observe system behavior during the tests. The third case study validates the performance model in SPN. And the last case study assesses the impact of availability on the performance model. The results show that the performance model was practical and efficient to quantify the metrics of interest.

Keywords: Performance Evaluation, Virtual Learning Environment, Cloud Computing, Stochastic Petri Net, Reliability Block Diagram.

Lista de ilustrações

Figura 1 – Serviços da computação em nuvem.	22
Figura 2 – Plataforma aberta flexível do Apache CloudStack.	23
Figura 3 – Infraestrutura em nuvem do Apache CloudStack com múltiplas zonas.	24
Figura 4 – Arquitetura do Moodle.	25
Figura 5 – Técnicas de avaliação de desempenho.	25
Figura 6 – Arquitetura de Teste Distribuído do JMeter.	26
Figura 7 – Elementos da rede de Petri.	27
Figura 8 – Exemplo de uma rede de Petri.	28
Figura 9 – Elemento da SPN.	28
Figura 10 – Exemplo da SPN.	29
Figura 11 – Diagrama de blocos de confiabilidade.	30
Figura 12 – Gráfico regressão linear.	32
Figura 13 – Metodologia adotada.	34
Figura 14 – Visão geral do ambiente de medição	35
Figura 15 – Modelo de desempenho proposto.	37
Figura 16 – Regressão linear.	39
Figura 17 – Modelo RBD da plataforma Apache CloudStack.	40
Figura 18 – Modelo RBD da máquina virtual.	41
Figura 19 – Modelo de disponibilidade SPN.	42
Figura 20 – Comparação da vazão.	44
Figura 21 – Comparação do tempo de resposta.	45
Figura 22 – Ambiente de medição com vmstat.	45
Figura 23 – Resultado da medição da utilização da CPU.	46
Figura 24 – Resultado da medição da utilização da memória.	46
Figura 25 – Resultado da medição da escrita no HD.	47
Figura 26 – Resultado da medição e modelagem.	48
Figura 27 – Modelo com disponibilidade.	49

Lista de tabelas

Tabela 1 – Características dos Trabalhos Relacionados.	19
Tabela 2 – Exemplo para cálculo da regressão linear.	32
Tabela 3 – Descrição dos lugares do modelo.	38
Tabela 4 – Descrição das transições do modelo.	38
Tabela 5 – Parâmetros do modelo RBD da plataforma CloudStack.	40
Tabela 6 – Parâmetros do modelo RBD da VM.	41
Tabela 7 – Resultado da disponibilidade do modelo da Figura 18.	41
Tabela 8 – Parâmetros de dependabilidade dos componentes da plataforma Apache CloudStack.	42
Tabela 9 – Resultado da disponibilidade do modelo da Figura 17.	42
Tabela 10 – Transições do modelo de disponibilidade.	43
Tabela 11 – Resultado para validação do modelo com tempo de resposta.	48
Tabela 12 – Comparação do tempo de resposta com e sem o modelo de disponibilidade.	49

Lista de abreviaturas e siglas

API	Application Programming Interface
AVA	Ambiente Virtual de Aprendizagem
AWS	Amazon Web Services
CEFETES	Centro Federal de Educação Tecnológica do Espírito Santo
CSS	Cascading Style Sheets
CTMC	Continuous Time Markov Chain
EaD	Educação a Distância
FTP	File Transfer Protocol
GRASP	Greed Randomized Search Procedure
HD	Hard Disk
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
KVM	Kernel-based Virtual Machine
VLE	Virtual Learning Environment
MLB	Método de Lattice Boltzmann
Moodle	Modular Object-Oriented Dynamic Learning Environment
MTTF	Mean Time to Fail
MTTR	Mean Time to Repair
NIST	National Institute of Standards and Technology
NoSQL	Not Only Structured Query Language
OLAM	Ocean-Land-Atmosphere Model
PaaS	Platform as a Service
PHP	Hypertext Preprocessor

PN	Petri Nets
RAM	Random Access Memory
RBD	Reliability Block Diagram
SaaS	Software as a Service
SPN	Stochastic Petri Nets
VM	Virtual Machine
UFRPE	Universidade Federal Rural de Pernambuco
WebCT	Web Course Tools
YCSB	Yahoo! Cloud Serving Benchmarking

Sumário

1	INTRODUÇÃO	14
1.1	Contexto	14
1.2	Motivação	15
1.3	Objetivos	15
1.4	Organização do Trabalho	16
2	TRABALHOS RELACIONADOS	17
2.1	Considerações Finais	19
3	FUNDAMENTAÇÃO TEÓRICA	21
3.1	Computação em Nuvem	21
3.1.1	Apache CloudStack	22
3.2	Ambiente Virtual de Aprendizagem	24
3.2.1	Moodle	24
3.3	Avaliação de Desempenho	25
3.3.1	Apache JMeter	26
3.4	Redes de Petri	26
3.4.1	Redes de Petri Estocásticas	28
3.5	Diagrama de Bloco de Confiabilidade	30
3.6	Regressão Linear	31
3.7	Considerações Finais	33
4	METODOLOGIA	34
4.1	Visão Geral da Metodologia	34
4.2	Considerações Finais	36
5	MODELOS	37
5.1	Modelo de Desempenho	37
5.1.1	Parâmetros e Métricas do Modelo	38
5.1.2	Regressão Linear	39
5.2	Modelo de Disponibilidade	40
5.3	Considerações Finais	43
6	ESTUDO DE CASO	44
6.1	Análise da Capacidade do Sistema	44
6.2	Análise da Medição de CPU, Memória e Escrita do HD	45
6.3	Validação de Modelo	47

6.4	Impacto da Disponibilidade no Modelo de Desempenho	48
6.5	Considerações Finais	50
7	CONCLUSÃO	51
7.1	Contribuições	51
7.2	Trabalhos Futuros	52
	REFERÊNCIAS	53

1 Introdução

Este capítulo apresenta uma introdução sobre computação em nuvem e ambiente virtual de aprendizagem destacando aspectos de desempenho e disponibilidade. Em seguida a motivação e objetivos desse trabalho são apresentados.

1.1 Contexto

Surgindo em meados de 1961, o termo computação em nuvem apresentava um modelo de computação oferecido com um serviço parecido com a distribuição de energia elétrica (MANSUR et al., 2010). A computação em nuvem é um conjunto de *data centers* virtualizados, que oferece serviço sob demanda, também conhecido como *pay per use* (pagar pelo o que usar). Atualmente, grandes empresas utilizam a computação em nuvem para oferecer seus serviços, exemplos como a Netflix, Google, Spotify e Uber.

A computação em nuvem é um paradigma que é composto de Grid computing e Utility computing (KIM et al., 2010). A Grid computing é uma tecnologia utilizada para o desenvolvimento da computação em nuvem enquanto a Utility computing é um pacote de recursos computacional (XIONG; PERROS, 2009). As suas principais características são rápida elasticidade, alta disponibilidade, confiabilidade e baixo custo (BAUER; ADAMS, 2012).

Inicialmente usado para servir de comunicação entre aluno e professor no ensino a distância, os ambientes virtuais de aprendizagem atualmente atendem cursos presenciais e a distância e também possuem vários recursos que ajudam na educação. As vantagens dos AVAs são a possibilidade de estudo a qualquer hora e qualquer lugar só necessitando de Internet para realizar o acesso, aprendizado coletivo onde vários alunos e professores trocam ideias em fóruns e chats e podendo, inclusive, elevar o ritmo de aprendizado.

Como o ambiente virtual de aprendizagem e o ambiente de computação em nuvem tem que garantir uma alta disponibilidade e oferecer uma boa performance para seus usuários. O gestor dos ambientes tem que planejar qual ambiente será utilizado e que atenda as demandas de desempenho e disponibilidade (DANTAS, 2018). A utilização de técnicas com medição, modelagem analítica e simulação podem ajudar o gestor na hora do planejamento, pois com essas avaliações será possível uma escolha que mais se ajuste a sua necessidade.

1.2 Motivação

Com quase treze mil alunos matriculados na graduação e pós-graduação em 2017 na UFRPE (UFRPE, 2017), o uso do ambiente virtual de aprendizagem está cada vez mais presente no dia a dia dos docentes e discentes. Lançado em 2015 na UFRPE (UFRPE, 2015), o AVA aproxima a relação e a comunicação entre alunos e professores com recursos de fóruns, chats e atividades síncronos e assíncronos. O Moodle foi a ferramenta escolhida para ser adotada como AVA da UFRPE. Em época de provas o acesso ao ambiente fica maior e com isso pode haver lentidão do sistema. No caso se a prova for online, haverá um grande fluxo de acessos de usuários, *uploads* e *downloads* de arquivos, o que pode gerar uma sobrecarga no AVA da UFRPE.

A computação em nuvem surgiu de uma necessidade que as empresas de comércio tinham em épocas de datas festivas como Natal, onde ocorria o aumento de recursos computacionais para conseguir atender a demanda requerida para essas datas. O problema era quando se passava essas datas comemorativas, os recursos extras que foram adquiridos ficavam ociosos e ocorreria um gasto desnecessário. Então várias empresas atualmente, estão criando *softwares open source* para oferecer infraestrutura como serviço, com segurança, alta disponibilidade, confiabilidade e entre outros. Entre as empresas que oferece essa infraestrutura está o Apache CloudStack (APACHE, 2017), OpenStack (OPENSTACK, 2019) e Eucalyptus (EUCALYPTUS, 2019).

Então a utilização do Moodle hospedado em ambiente de nuvem privada precisa de uma de avaliação de desempenho e disponibilidade, e assim atestar a qualidade do serviço provido. Além disso, esse estudo sobre o desempenho desse tipo de sistema vai auxiliar os projetistas a estimar as configurações demandas para dar suporte ao sistema em diferentes situações e com diversos números de usuários.

1.3 Objetivos

Dada a necessidade de se realizar um estudo de desempenho sobre o AVA, o objetivo desse trabalho é propor uma estratégia baseada em modelos para a avaliação de desempenho e da disponibilidade do ambiente virtual de aprendizagem Moodle em nuvem privada.

Objetivo Específicos:

- Propor uma estratégia para medir o desempenho do Ambiente Virtual de Aprendizagem através de um conjunto métricas como vazão, tempo de resposta, utilização de CPU e memória e taxa de escrita em HD.

- Gerar uma base de dados com os valores das medições em nuvem privada montada em laboratório.
- Propor modelos de desempenho e disponibilidade para representar o Ambiente Virtual de Aprendizagem.

1.4 Organização do Trabalho

Este trabalho está dividido em 7 capítulos. O Capítulo 2 apresenta os trabalhos relacionados a essa monografia. O Capítulo 3 apresenta a fundamentação teórica abordando os seguintes assuntos: computação em nuvem, ambiente virtual de aprendizagem, avaliação de desempenho, redes de Petri, diagrama de blocos de confiabilidade e regressão linear. O Capítulo 4 apresenta a metodologia usada no trabalho. O Capítulo 5 apresenta os modelos de desempenho e disponibilidade propostos. O capítulo 6 descreve os estudos casos utilizados nesta monografia. E o último capítulo apresenta a conclusão, contribuições e trabalhos futuros dessa pesquisa.

2 Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados a análise de desempenho de AVA em computação em nuvem. O estado da arte possui várias pesquisas relacionadas com este trabalho. Alguns deles têm como foco a comparação de infraestruturas de nuvens, avaliação de desempenho do Moodle, modelagem para avaliar desempenho em nuvens privadas e avaliação de desempenho em banco de dados. Ao final desse capítulo é feito um comparativo entre os trabalhos relacionados e o trabalho proposto.

Sousa ([SOUSA, 2015](#)) realizou um planejamento para infraestruturas de nuvens privadas com uma solução integrada com metodologia, modelos de otimização utilizando meta heurística *Greedy Randomized Search Procedure* (GRASP), modelos para geração de cenários de desempenho e custo e modelos de representação. Os modelos de representação são compostos de *Reliability Block Diagram* (RBD) e *Stochastic Petri Nets* (SPN) e com eles são realizadas avaliação de desempenho e disponibilidade. Nos estudos de casos foram avaliados o impacto da variação de carga, da ocorrência de defeitos e atividades de reparo no ambiente de nuvem Eucalyptus utilizando vários cenários de configuração de software e de hardware.

Júnior e Tavares ([JUNIOR; TAVARES, 2015](#)) realizaram uma avaliação de desempenho e dependabilidade do Moodle na plataforma OpenNebula, utilizando cenários com 1 a 10 usuários simultâneos para acessar um quiz dentro do ambiente virtual de aprendizagem. Neste trabalho foram contabilizados os números de amostras, tempo médio de resposta, requisições por segundos e em kilobytes por segundos utilizando as ferramentas JMeter e Sysstat. Como conclusão foi reportada a importância do dimensionamento de recursos computacionais fornecidos pela a infraestrutura da nuvem para atender as necessidades do Moodle.

Os autores ([MARON et al., 2014](#)) avaliaram o desempenho das plataformas OpenStack e OpenNebula utilizando os benchmarks Iperf para avaliar a rede e coletar a vazão, Iometer para testar unidades de armazenamento com um arquivo de 100 MB analisando a escrita, rescrita, leitura e releitura, Stream para testes de memória, analisando as funções de Add, Scale, Triad e Copy e o Linpack para os testes de processador com uma matriz de 4000x4000 para ser executado pontos flutuantes. Ao final, foi apresentado um resumo comparativo do desempenho obtido para cada ambiente a partir dos benchmarks avaliados, onde foi mostrado que o OpenStack é recomendado para aplicações que utilizam somente o disco e OpenNebula é melhor para usar rede, memória e processador.

Marcelino et al. ([MARCELINO et al., 2008](#)) avaliaram o desempenho do Moodle

na infraestrutura do CEFETES a partir dos arquivos de log de acesso HTTP do Moodle. Os autores também simularam o impacto na infraestrutura com variação de parâmetros de tempo de disco e o link de dados com softwares WebModels e OpenQN. Assim foi constatado que se tiver um aumento na quantidade de requisições por segundo haverá gargalo no disco, se houver troca por um disco mais rápido o gargalo será do link de dados.

Fé (FÉ, 2017) propõe modelos em SPN e de otimização para auxiliar o planejamento de sistemas de transcodificação de vídeo em nuvem privada e pública. Com os modelos em SPN propostos foram definidas métricas de vazão, tempo de resposta e custo. Foram realizados testes no ambiente de nuvem Cloudstack, com o JMeter fazendo o papel de cliente para enviar as requisições de transcodificações. Foi concluído nos estudos que se diminuir o tempo de resposta de 30 para 15 segundos vai resultar em um aumento no custo de aproximadamente 300% e de 45 para 30 segundos vai ter um aumento no custo de 6%.

Dantas (DANTAS, 2018) propôs modelos de desempenho, disponibilidade, custo e um mecanismo para avaliação de espaço de projeto para infraestrutura de nuvem com um suporte a um serviço de vídeo. Foi utilizado como o gerenciador de nuvem o ambiente Eucalyptus. Em seu estudo de desempenho foi utilizado o JMeter para geração de carga e para validação do modelo proposto.

Em Rosso (ROSSO, 2015) foi avaliado o desempenho das aplicações científicas *Ocean-Land-Atmosphere Model* (OLAM) e Método de Lattice Boltzmann (MLB) em ambiente nativo e em nuvens privadas utilizando os ambientes OpenStack e OpenNebula. O autor realizou uma comparação entre eles para verificar qual tinha o melhor desempenho nos testes realizados. Os resultados apontaram uma melhor performance no ambiente nativo e mostrando que existe uma queda no desempenho na IaaS privadas.

Torres (TORRES et al., 2016) propôs modelos em *Continuous Time Markov Chain* (CTMC), SPN e RBD para uma avaliação integrada de métricas de disponibilidade e desempenho em um serviço de armazenamento em uma nuvem privada gerenciada pelo ambiente Eucalyptus. O autor utilizou para as medições as ferramentas JMeter e Apache benchmarking para validação do modelo. Em seus testes de desempenho foram utilizadas métricas de vazão e utilização do sistema.

Campos (CAMPOS, 2015) avaliou o desempenho do ambiente Eucalyptus com escalonamento automático que hospedam um serviço web de recomendação de eventos. Para isso, modelos em SPN e CTMC foram propostos. Foram utilizadas métricas de desempenho como tempo médio de execução do monitoramento de escala automática, tempo de instanciação da VM e tempo médio de resposta. Como conclusão observou-se que os parâmetros utilizados podem ter impacto no tempo de resposta e

o autores também propuseram formas para melhorar o desempenho do sistema.

Barros (BARROS, 2018) avaliou o desempenho do banco de dados NoSQL Cassandra e propôs um modelo em SPN para a inserção de dados. Foi utilizado o YCSB para medição do sistema e para validação do modelo. Nos estudos de casos foram utilizados vários cenários com taxa de operações por segundo e observou-se uma redução de aproximadamente 20% no tempo de resposta aumentando o limite da MemTable, que em consequência precisará de mais memória RAM. E por fim, foi realizado uma análise de consumo de energia para estes mesmos cenários.

Os autores (JÚNIOR et al., 2012) propõem desenvolver uma extensão para o Moodle, que visa fornecer um ambiente de aprendizagem ubíqua utilizando recomendação de objetos de aprendizagem. Foi utilizado um algoritmo genético para identificar quais objetos de aprendizagem serão recomendados. No estudo de caso o estudante realiza o *login*, responde ao um questionário para identificar suas preferências, depois escolhe qual é o seu dispositivo móvel, faz o *download* e instala o aplicativo em seu celular e quando entrar na aplicação terá mensagens com as recomendações de objetos de aprendizagem de acordo com seu perfil.

A Tabela 1 mostra as características dos onze trabalhos relacionados e do trabalho proposto. A tabela mostra que seis trabalhos utilizaram modelagens do tipo CTMC, SPN e RBD. Apenas três tem foco em ambiente virtual de aprendizagem. Oito dos onze trabalhos utilizam ferramentas de computação em nuvem.

Tabela 1 – Características dos Trabalhos Relacionados.

Artigos	Modelagem	Amb. de Nuvem	AVA
(SOUSA, 2015)	SPN, RBD	Eucalyptus	Não
(JUNIOR; TAVARES, 2015)	-	OpenNebula	Sim
(MARON et al., 2014)	-	OpenStack e OpenNebula	Não
(MARCELINO et al., 2008)	-	-	Sim
(FÉ, 2017)	SPN	CloudStack	Não
(DANTAS, 2018)	CTMC, SPN e RBD	Eucalyptus	Não
(ROSSO, 2015)	-	OpenStack e OpenNebula	Não
(TORRES et al., 2016)	CTMC, SPN e RBD	Eucalyptus	Não
(CAMPOS, 2015)	CTMC e SPN	Eucalyptus	Não
(BARROS, 2018)	SPN	-	Não
(JÚNIOR et al., 2012)	-	-	Sim
Este Trabalho	SPN e RBD	CloudStack	Sim

2.1 Considerações Finais

Este capítulo apresentou os principais trabalhos relacionados ao estudo proposto. No estado da arte vários trabalhos abordam avaliação de desempenho, com-

putação em nuvem mas poucos abordam a avaliação de desempenho em AVA sem propor modelagem. Existem também trabalhos que abordam o armazenamento de banco de dados ou da infraestrutura.

3 Fundamentação Teórica

Este capítulo apresenta os conceitos básicos para um melhor entendimento deste trabalho. Primeiramente serão apresentados os conceitos sobre computação em nuvem e sobre a IaaS Apache CloudStack. Em seguida, serão abordados os principais conceitos do AVA e a ferramenta Moodle. Logo após, serão apresentados a ideia sobre avaliação de desempenho e do software Apache JMeter utilizada para medições. Também são abordados os principais conceitos sobre rede de Petri e sua extensão, a rede de Petri estocástica. Também será introduzido o Diagrama de Bloco de Confiabilidade (RBD). E por fim, apresenta-se a regressão linear.

3.1 Computação em Nuvem

O modelo de computação em nuvem foi desenvolvido com o objetivo de fornecer serviços de fácil acesso e de baixo custo e garantir características tais como disponibilidade e escalabilidade (RUSCHEL; ZANOTTO; MOTA, 2010).

A NIST (NIST, 2014) definiu a computação em nuvem como um modelo para acesso ubíquo, conveniente e sob demanda a uma rede compartilhada de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços. Existe três tipos de serviços de computação em nuvem definidos pela NIST (NIST, 2014):

Infraestrutura como Serviço (Infrastructure as a Service - IaaS): É um modelo de serviço que fornece recursos de computação como hardware, armazenamento e rede. O consumidor não gerencia ou controla a infraestrutura de nuvem, mas tem controle sobre sistemas operacionais, armazenamento e aplicativos. Podemos citar como exemplos de IaaS são o Apache CloudStack (APACHE, 2017), Amazon EC2 (AMAZON, 2018) e Google Cloud (GOOGLE, 2018b).

Plataforma como Serviço (Platform as a Service - PaaS): É um modelo de serviço onde o usuário pode utilizar para desenvolvimento e execução de aplicações. O consumidor não gerencia ou controla a infraestrutura de nuvem, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre os aplicativos implantados. Os exemplos de PaaS são Google Cloud Platform (GOOGLE, 2018c), Microsoft Azure (MICROSOFT, 2018a) e Red Hat OpenShift (REDHAT, 2018).

Software como Serviço (Software as a Service - SaaS): É um modelo de serviço onde é fornecido aplicativos executado em uma infraestrutura de nuvem. O

consumidor não gerencia ou controla a infraestrutura de nuvem, incluindo rede, servidores, sistemas operacionais, armazenamento ou até mesmo recursos de aplicativos individuais, com a possível exceção de configurações de aplicativos. Os exemplos de SaaS são Google Apps (GOOGLE, 2018a), Dropbox (DROPBOX, 2018) e Microsoft Office 365 (MICROSOFT, 2018b). A Figura 1 ilustra os serviços da computação em nuvem detalhando onde o cliente e o administrador pode gerenciar em cada serviço.

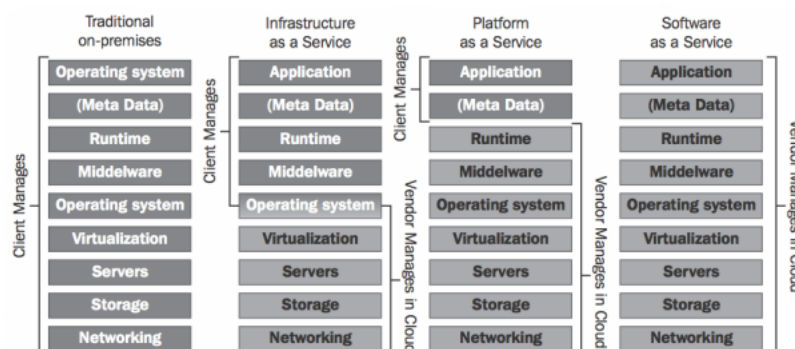


Figura 1 – Serviços da computação em nuvem. (SCHOUTEN, 2014)

Além disso, também existe quatro tipos de modelos de computação em nuvem (NIST, 2014):

Nuvem pública: a infraestrutura em nuvem é disponibilizada ao público geral.

Nuvem privada: a infraestrutura em nuvem é operada exclusivamente para uma organização.

Nuvem comunitária: a infraestrutura da nuvem é compartilhada por várias organizações.

Nuvem híbrida: a infraestrutura da nuvem por uma composição de duas ou mais nuvens (privada, pública e comunitária) .

Este trabalho vai abordar o uso da IaaS com o software Apache CloudStack que será introduzido logo a seguir. E será usado para o gerenciamento dos recursos da nuvem privada.

3.1.1 Apache CloudStack

O Apache CloudStack é um software de código aberto projetado para implantar e gerenciar grandes redes de máquinas virtuais com uma IaaS altamente escalável e disponível (APACHE, 2017). Usada por grandes empresas como Dell, Huawei e Walt Disney. Foi criada em 2008 por uma start-up que deu o seu primeiro nome como VMOPs. Em 2010 adquiriu o nome Cloud.com e foi lançado o CloudStack. Em

2011 foi vendida para Citrix Systems que no ano seguinte doou para Apache Software Foundation.

O Apache CloudStack gerencia a computação, rede e recurso de armazenamento, utiliza vários hypervisors como KVM, vSphere e XenServer para virtualização e também suporta a API do AWS, além de suas próprias APIs (KUMAR et al., 2014). A Figura 2 ilustra os hypervisors, armazenamento e rede que podem ser utilizados pelo ambiente.

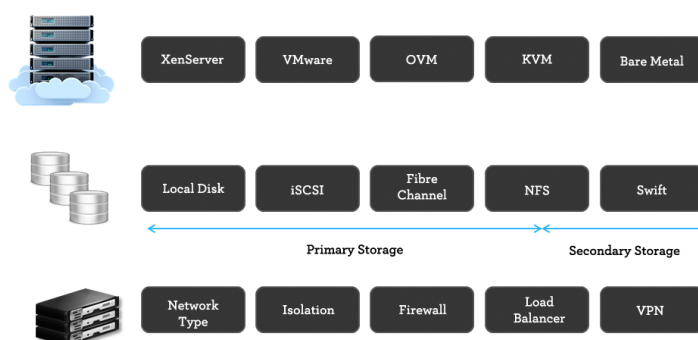


Figura 2 – Plataforma aberta flexível do Apache CloudStack.
(BLUE, 2013)

Segundo o Apache (APACHE, 2017), os recursos dentro da infraestrutura em nuvem do Apache CloudStack são regiões, zonas, pods, clusters, hosts, armazenamento primário e secundário (APACHE, 2017). A Figura 3 ilustra a infraestrutura do Apache CloudStack com várias zonas sendo utilizadas.

Regiões: uma coleção de uma ou mais zonas geograficamente próximas gerenciadas por um ou mais servidores de gerenciamento (APACHE, 2017).

Zonas: é equivalente a um único *datacenter*. Uma zona consiste em uma ou mais pods e armazenamento secundário (APACHE, 2017).

Pods: Um pod geralmente é um rack ou fileira de racks que inclui um switch de camada 2 e um ou mais clusters (APACHE, 2017).

Clusters: um *cluster* consiste em um ou mais hosts homogêneos e armazenamento primário (APACHE, 2017).

Hosts: um único nó de computação dentro de um cluster; muitas vezes um hipervisor (APACHE, 2017).

Armazenamento Primário: um recurso de armazenamento fornecido a um único cluster para a execução real de imagens de disco da instância (APACHE, 2017).

Armazenamento Secundário: Um recurso de toda a zona que armazena modelos de disco, imagens ISO e instantâneos (APACHE, 2017).

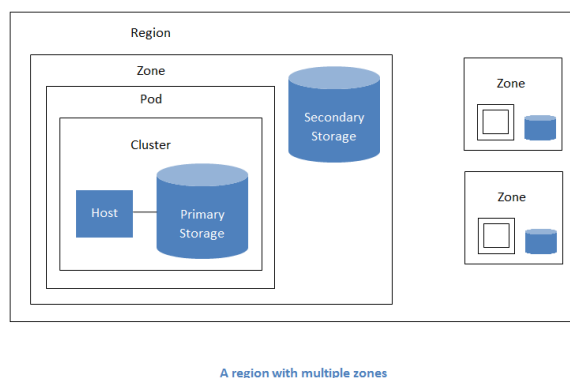


Figura 3 – Infraestrutura em nuvem do Apache CloudStack com múltiplas zonas. (APACHE, 2017)

3.2 Ambiente Virtual de Aprendizagem

Durante a década 90, com expansão e implementação de programas educacionais em meio online, os AVAs começaram a ganhar espaço e reconhecimento no âmbito educacional (BELUCE; OLIVEIRA, 2012). O AVA é uma ferramenta com dimensão pedagógica em que o professor disponibiliza vários recursos para o aluno, como textos, aulas, cronogramas e exercícios (HAGUENAUER; LIMA; FILHO, 2010).

Os AVAs possuem recursos de interação, comunicação e locais onde os participantes podem armazenar conteúdos utilizando várias mídias (SILVA, 2012). As principais características do AVA são a sua interatividade, hipertextualidade e conectividade (KENSKI, 2007).

Neste trabalho vai ser abordado, a seguir, o AVA Moodle que é uma das ferramentas mais usadas atualmente e também é utilizada pela UFRPE tanto para os cursos presenciais como também para os cursos a distância (EaD).

3.2.1 Moodle

O Moodle foi criado em 2001 por Martin Dougiamas (MOODLE, 2018), logo após a experiência vivenciada com o AVA WebCT na Curtin University com o objetivo de ser um método alternativo de ensino online. Em 2002 foi lançado a primeira versão do Moodle, e vários usuários em um fórum já estavam traduzindo para outros idiomas e criando temas. Com mais de cem milhões de usuários registrados, o Moodle é um dos AVA mais utilizado.

O Moodle é uma plataforma de aprendizado em código aberto projetada para fornecer aos educadores, administradores e alunos um sistema robusto, seguro e integrado para criar ambientes de aprendizado personalizados (MOODLE, 2018). A ferramenta disponibiliza os seguintes recursos para o desenvolvimento de atividades como

fóruns, chat, materiais, questionário e entres outros.

A Figura 4 mostra a arquitetura do Moodle que é composta por um servidor web PHP, um banco de dados (pode ser MySQL) e uma interface gráfica para usuário em HTML com CSS e JavaScript.

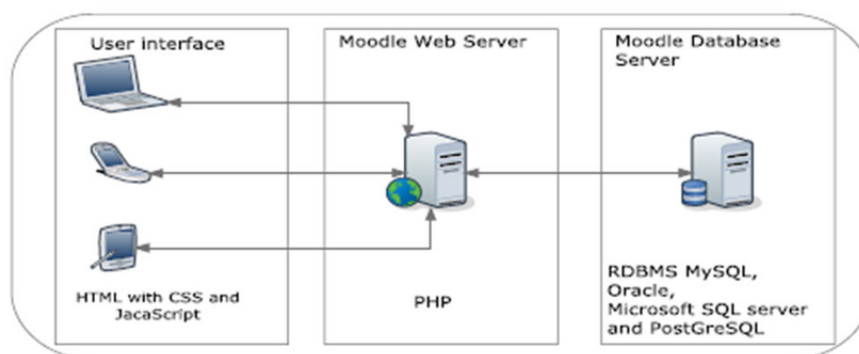


Figura 4 – Arquitetura do Moodle. (AYYAD, 2016)

3.3 Avaliação de Desempenho

A avaliação de desempenho pode ser classificada com medição e modelagem (CALLOU et al., 2011; JAIN, 1991). Um dos principais propósito para o desenvolvedor é fornecer alto desempenho com baixo custo (BARROS, 2018). A seguir, é ilustrado na Figura 5 as técnicas de avaliação de desempenho.

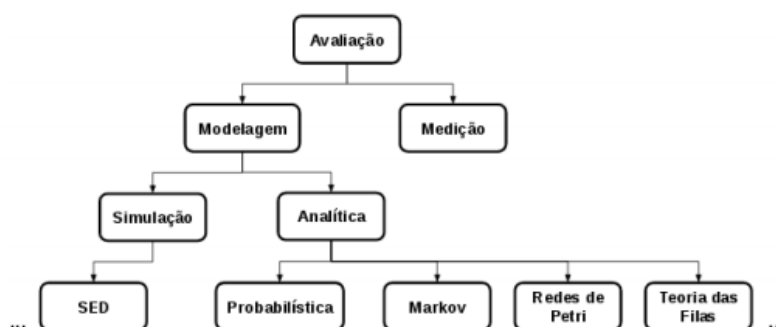


Figura 5 – Técnicas de avaliação de desempenho. (CALLOU et al., 2011)

A medição 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 se um novo sistema estiver sendo feito (TORRES et al., 2016). Já a modelagem analítica é utilizado para abstração do sistema. O modelo deve conter detalhes do sistema que são essenciais para o seu

comportamento. E a simulação do modelo dar resultados não tão precisos quanto os fornecidos pela medição, mas é possível calcular as estimativas (CALLOU et al., 2011).

Neste trabalho será utilizado a ferramenta Apache JMeter, que será introduzida a seguir, para a realização das medições do Moodle hospedado na nuvem privada gerenciada pelo Apache CloudStack. Também será utilizada a ferramenta Mercury para modelagem em SPN e simulação dos modelos.

3.3.1 Apache JMeter

O Apache JMeter é um software de código aberto, desenvolvido em Java e é utilizado para teste de carga e medição de desempenho. Ela tinha sido criada originalmente para testes de aplicativos Web, mas já foi expandida para outras funções de testes (APACHE, 2019).

O Apache JMeter pode ser utilizado para testes em recursos estáticos e dinâmicos como arquivos, servidores FTP, banco de dados, objetos Java e entre outros (CHANDEL; PATIAL; GULERIA, 2013). A Figura 6 ilustra a arquitetura do teste distribuído do Apache JMeter, onde o JMeter gera uma carga com vários usuários e realiza várias medições enquanto manda as requisições e recebe as respostas do servidor. As métricas utilizadas pelo Apache JMeter são, por exemplo, vazão que é dada como requisições por segundo, quantidade de kilobytes enviados e recebido por segundo e tempo de resposta.

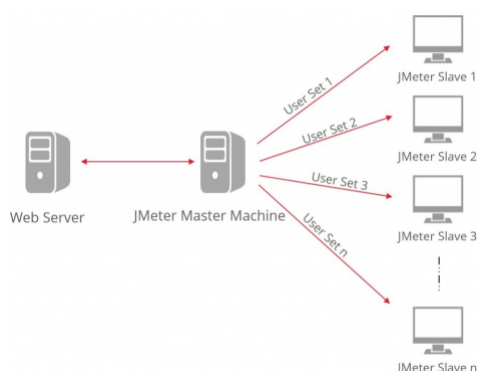


Figura 6 – Arquitetura de Teste Distribuído do JMeter.
(NIGAM, 2014)

3.4 Redes de Petri

A origem do conceito da rede de Petri ocorreu em 1962 na tese de doutorado de Carl Adam Petri na faculdade de Matemática e Física da Universidade Darmstadt na Alemanha (MURATA, 1989). A rede de Petri é uma técnica de especificação de sistemas que permite uma representação matemática e que é usada para modelar

sistemas paralelos, concorrentes, assíncronos e não-determinísticos (MACIEL; LINS; CUNHA, 1996).

A rede de Petri é composta por 4 elementos: lugares, transições, arcos e tokens. Os lugares representam o estado do sistema. As transições são ações realizadas para alterar o estado do sistema. Para uma transição está habilitada, é preciso que as pré-condições sejam satisfeitas. Os arcos representam o fluxo de token pela rede e os tokens representam o estado que o sistema se encontra. Graficamente, os lugares são representados por círculos, as transições, por barras, os arcos, por setas e os tokens, por pontos. A Figura 7 ilustra os elementos da rede de Petri.

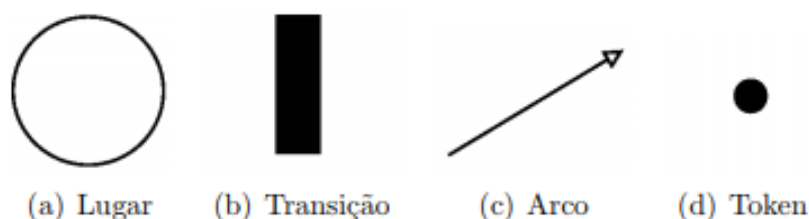


Figura 7 – Elementos da rede de Petri.

A representação formal de uma rede de Petri é uma 5-tupla $PN = \{P, T, F, W, M_0\}$ (MURATA, 1989), onde:

- $P = \{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_1, \dots, t_n\}$ é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos;
- $W : F \rightarrow \{1, 2, 3, \dots\}$ é uma função de peso do arco;
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$ é a marcação inicial, $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$

A seguir, a Figura 8 mostra um exemplo de PN. O modelo apresenta o funcionamento de um sistema, onde os lugares representam o estado do sistema (on ou off) e as transições representam as ações que altera o estado do sistema (ligar e desligar). O modelo na Figura 8 (a) mostra que o sistema está ligado, pois possui token no lugar on. A única transição que pode ser disparada é desligar. Depois do disparo dessa transição, o modelo passará para o estado desligado e irá possuir um token no lugar off (Figura 8 (b)). Após isso, a transição habilitada para disparar é a Ligar, e o disparo desta fará o sistema voltar ao estado de ligado como na Figura 8 (a).

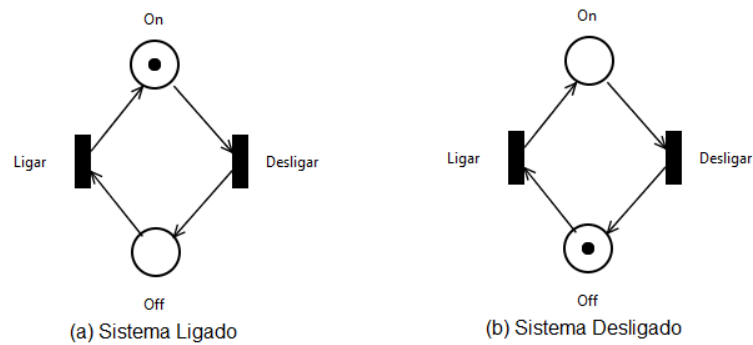


Figura 8 – Exemplo de uma rede de Petri.

3.4.1 Redes de Petri Estocásticas

As redes de Petri estocástica são uma extensão da rede de Petri utilizada para a modelagem de desempenho e dependabilidade (TORRES et al., 2016). Nas SPN são adicionados dois elementos, como pode ser visto na Figura 9, que são as transições estocástica Figura 9 (a) e o arco inibidor Figura 9 (b). A transição estocástica utiliza tempo associados, de modo que o período de habilitação da transição corresponde ao período de execução da atividade. Já o arco inibidor desativa a transição se houver tokens no lugar de origem do arco inibidor, e vai ser ativada quando não houver tokens (SILVA, 2016).

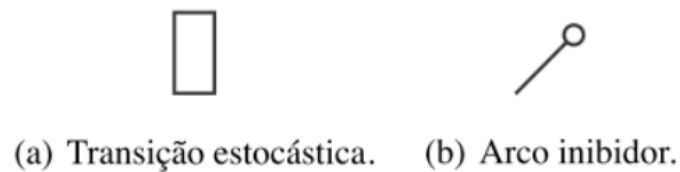


Figura 9 – Elemento da SPN.

A definição de uma rede de Petri estocástica é uma 9-tupla $SPN = \{P, T, I, O, H, \Pi, G, M_0, Atts\}$ (GERMAN, 2000), onde:

- $P = \{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{m \times n}$ é a matriz que representa a multiplicidade dos arcos de entrada (que podem ser dependentes de marcações);
- $O \in (\mathbb{N}^n \rightarrow \mathbb{N})^{m \times n}$ é a matriz que representa a multiplicidade dos arcos de saída (que podem ser dependentes de marcações);

- $H \in (\mathbb{N}^n \rightarrow \mathbb{N})^{m \times n}$ é a matriz que representa os arcos inibidores (que podem ser dependentes de marcações).
- $\Pi \in \mathbb{N}^m$ é um vetor vincula o nível o nível de prioridade para cada transição;
- $G \in (\mathbb{N}^n \rightarrow \{true, false\})^m$ é o vetor que associa uma condição de guarda relacionada à marcação do lugar a cada transição;
- $M_0 \in \mathbb{N}^n$ é o vetor que associa uma marcação inicial de cada lugar (estado inicial);
- $Atts = (Dist, W, Markdep, Policy, Concurrency)^m$ é o vetor que associa uma marcação inicial de cada lugar (estado inicial), onde:
 - $Dist \in \mathbb{N}^m \rightarrow f$ é uma função de distribuição de probabilidade associada ao tempo de uma transição, onde o domínio de $f[0, \infty)$ (esta distribuição pode ser dependente de marcação);
 - $W \in \mathbb{N}^m \rightarrow \mathbb{R}^+$ é uma função de peso (esta distribuição pode ser dependente de marcação);
 - $Markdep \in \{constant, enabdep\}$ é definido como a distribuição de probabilidade associada ao tempo de uma transição constante (*constant*) ou dependente de marcação (*enabdep*);
 - $Policy \in \{prd, prs\}$ é a política de memória adotada pela transição (*prd* - *preemptive repeat different*, corresponde a *race enabling policy*; *prs* - *preemptive resume*, corresponde a *age memory policy*).
 - $Concurrency \in \{ss, is\}$ é o grau de concorrência das transições, onde *ss* representa a semântica de *single server* e *is* representa a semântica de *infinity server*.

A Figura 10 mostra um exemplo de SPN, onde é representado um sistema de cliente-servidor. A transição TE0 representa a chegada de requisição para o servidor. A presença de ao menos um token no lugar P0 mostra que tem requisição para ser tratada no servidor. A transição TE1 representa a execução da resposta do servidor. Um token no lugar P1 representa que tem resposta para ser enviada para o cliente e o disparo da transição TE2 representa o envio da requisição para o cliente.

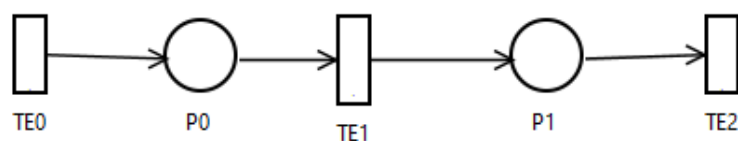


Figura 10 – Exemplo da SPN.

Com os modelos em SPN é possível realizar cálculos de métricas de desempenho. Um exemplo dela é o tempo de resposta, que é o tempo entre o pedido de realização de um serviço pelo usuário e a chegada de resposta do pedido ao usuário (TRIVEDI et al., 1996). Usando a lei de Little é possível calcular o tempo de resposta, conforme a equação 3.1.

$$T = \frac{N}{\lambda} \quad (3.1)$$

Onde T representa o tempo de resposta, N indica o número de requisições no sistema e λ representa a taxa de chegada do sistema (ZIMMERMANN et al., 2006). Então considerando o modelo representado pela Figura 10, $N = E\{\#P0\} + E\{\#P1\}$ que corresponde ao número de requisições para serem atendidas pelo sistema e $\lambda = W(TE0)$ representa a taxa de chegada de requisições para o servidor (ZIMMERMANN et al., 2006). Onde E representa o números de tokens do lugar indicado entre chave e W(TE0) representa o tempo de chegada de requisição da transição TE0.

3.5 Diagrama de Bloco de Confiabilidade

O Diagrama de Bloco de Confiabilidade (RBD) foi inicialmente proposto como uma técnica para calcular a confiabilidade de um sistema, mas atualmente também é utilizado para cálculo de disponibilidade. Em um diagrama de blocos, os componentes são representados por blocos, interligados com outros blocos e podem ter as seguintes composições: série, paralelo ou combinações dessas estruturas. (TRIVEDI et al., 1996).

A Figura 11 ilustra dois modelos de RBD, onde na Figura 11 (a) os blocos são organizados em série e na Figura 11 (b) os blocos são organizados em paralelo. No sistema em série, quando um componente falhar, o sistema todo irá parar. Já no sistema em paralelo, o sistema só irá falhar se em ambos os blocos venham a falhar.

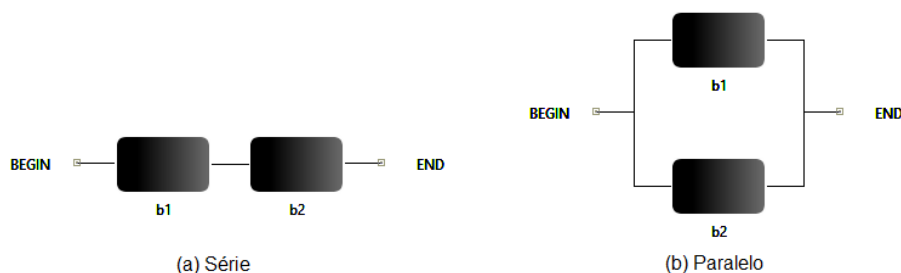


Figura 11 – Diagrama de blocos de confiabilidade.

A confiabilidade de dois blocos conectados em série (R_s), considerando n componentes, é obtida pela Equação 3.2 :

$$R_s = \prod_{i=1}^n R_i(t) \quad (3.2)$$

Onde $R_i(t)$ é a confiabilidade do bloco i . A confiabilidade de dois blocos conectados em paralelo (R_p), considerando n componentes, é obtida pela Equação 3.3:

$$R_p = 1 - \prod_{i=1}^n (1 - R_i(t)) \quad (3.3)$$

Onde $R_i(t)$ é a confiabilidade do bloco i .

3.6 Regressão Linear

A regressão linear é uma técnica estatística para relacionar uma variável x que é independente com uma variável y que é dependente (MATOS, 1995). A regressão linear simples é dada pela seguinte Equação 3.4.

$$y = a + bx \quad (3.4)$$

Onde y é a resposta de 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 dada pela seguinte Equação 3.5.

$$b = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (3.5)$$

Onde n é quantidade de pontos (x_i, y_i) . Já o cálculo do parâmetro de regressão a é dada pela seguinte Equação 3.6.

$$a = \bar{y} - b\bar{x} \quad (3.6)$$

Onde \bar{y} e \bar{x} é a média aritmética da variável de saída e de entrada respectivamente. Existe um grau de erro associado a regressão que é chamado de coeficiente de determinação e é representado por R^2 (LILJA, 2000). 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 dada pela seguinte Equação 3.7.

$$R^2 = \frac{b \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.7)$$

A Tabela 2 apresenta um exemplo com vários tamanhos de arquivos (variável independente) e os tempos de leitura (variável dependente) para realizar o cálculo da regressão linear.

Tabela 2 – Exemplo para cálculo da regressão linear.

Tamanho do Arquivo (bytes)	Tempo de Leitura (μs)
10	3,8
50	8,1
100	11,9
500	55,6
1000	99,6
5000	500,2
10000	1006,1

Com esses dados é possível calcular as Equações 3.5, 3.6 e 3.7. Os resultados dos parâmetros de regressão obtidos foram os seguintes: $b = 0,1002$ e $a = 2,24$. O resultado para o coeficiente de regressão é $R^2 = 0,9998$ e a equação da regressão linear obtida para esse exemplo é $y = 2,24 + 0,1002x$. A Figura 12 ilustra a regressão linear do exemplo da Tabela 2 e pode se notar que os valores previstos estão sobre os valores reais e R^2 próximo a 1 indica a validade da regressão.

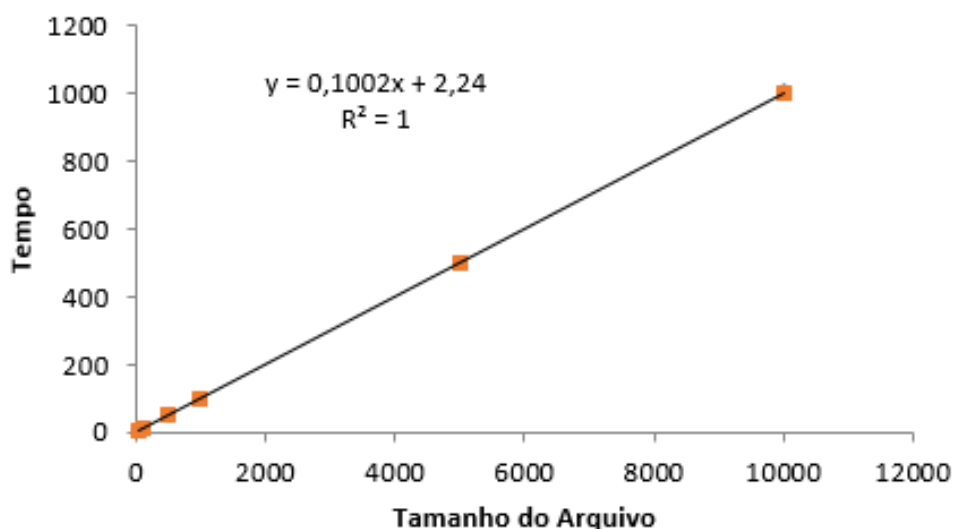


Figura 12 – Gráfico regressão linear.

3.7 Considerações Finais

Este capítulo apresentou os conceitos básicos sobre computação em nuvem e mostrou o ambiente Apache CloudStack que será utilizado neste trabalho. Em seguida foi apresentado o AVA e a ferramenta Moodle. Também foi introduzido a avaliação de desempenho e da ferramenta Apache JMeter e os principais conceitos sobre redes de Petri e sua extensão SPN. Foram descritas a definição do RBD e suas organizações. E finalmente foi descrito a regressão linear e um exemplo de como realizar os cálculos para obtenção da equação.

4 Metodologia

Este capítulo apresenta a metodologia usada para a avaliação de desempenho no ambiente virtual de aprendizagem em nuvem privada. Também são contextualizados as atividades realizadas e os objetos adquiridos em cada uma das atividades.

4.1 Visão Geral da Metodologia

A Figura 13 ilustra a metodologia adotada para este trabalho, destacando as suas atividades que são: estudar o sistema, instalar e configurar a nuvem, instalar e configura o Moodle, criar plano de teste, realizar medição, analisar resultados, criar modelo, calcular regressão linear, simular o modelo e validar o modelo e os objetos adquiridos ao final de cada atividade. A seguir, cada fase da metodologia adotada será detalhada.

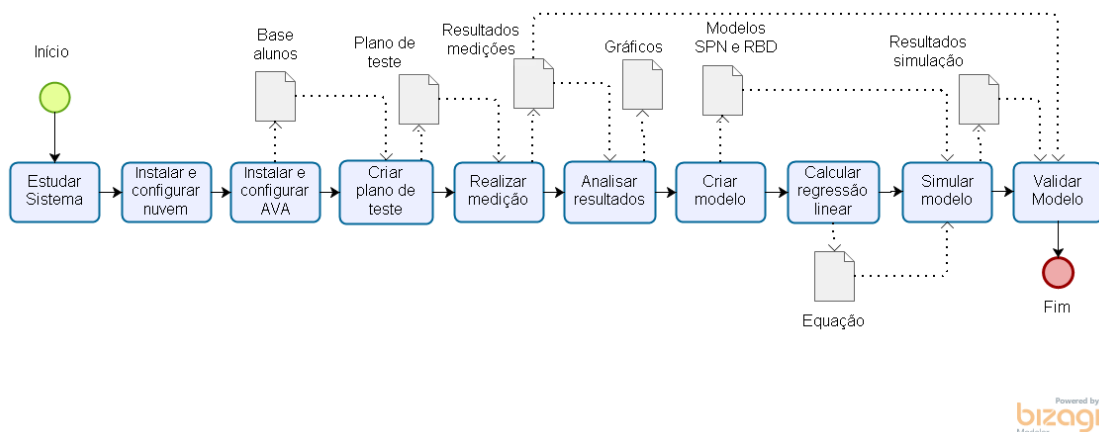


Figura 13 – Metodologia adotada.

Estudar o sistema: O objetivo desta fase foi realizar um estudo com a finalidade de se obter um entendimento da infraestrutura em nuvem Apache CloudStack e seus componentes, entendimento do funcionamento do ambiente virtual de aprendizagem Moodle que é utilizado na UFRPE, definição de métricas que serão utilizadas para avaliação do desempenho.

Instalar e configurar a nuvem: Esta atividade compreende a instalação e configuração da infraestrutura em nuvem com o Apache CloudStack. O ambiente foi montado em computador com a seguinte configuração: processador AMD A8-550B 3.2 GHz com 4 cores, 8 GB de memória RAM, 500 GB de armazenamento e sistema operacional CentOS 7. Também foi instanciado uma máquina virtual com a seguinte

configuração: processador com 2 GHz e 2 cores, 4 GB de memória RAM, 20 GB de armazenamento e sistema operacional CentOS 7.

Instalar e configurar o AVA: Esta fase corresponde a instalação e configuração do ambiente virtual de aprendizagem Moodle na máquina virtual. Além disso, nesta atividade ainda ocorre a criação de uma base de alunos, professores, curso, turma e atividade. A base de alunos foi criada a partir do modelo de cadastro do Moodle, onde contém informações obrigatórias como: nome, E-mail e senha. Todos esses dados inserido são fictícios.

Criar plano de teste: Um plano de teste na ferramenta de medição JMeter foi criado com a finalidade de se realizar experimentos no ambiente montado em laboratório. Nesta atividade será utilizada a base de dados de alunos que servirá como carga para o teste. Aqui foram definidas as seis atividades realizadas no ambiente que são: entrar no Moodle, realizar login, entrar no curso, entrar na atividade, realizar o upload de um arquivo de tamanho de 1 MB na atividade e realizar logout. Também foram definidos a quantidade de usuários virtuais e o tempo de inicialização de cada um.

Realizar medição: Esta atividade corresponde a medição utilizando a ferramenta JMeter e a ferramenta Vmstat, onde foi aplicado o plano de teste criado na atividade anterior. O JMeter foi responsável pela coleta de métricas como vazão e tempo de resposta, já o Vmstat mediu utilização da CPU, utilização da memória e taxa de escrita no HD. Os testes ocorreu trinta vezes para cada quantidade de clientes, que foi de um até quarenta clientes simultâneos e com chegada entre eles de 0,1 segundos. Essas repetições de testes é para se obter um padrão nos resultados. A Figura 14 ilustra a visão geral do ambiente de medição. O ambiente é composto de um cliente com a ferramenta JMeter para realizar as medições, uma máquina que será utilizada para ser o ambiente de nuvem que foi gerenciado pelo Apache CloudStack, uma VM onde foi instalado o AVA Moodle e a ferramenta de medição Vmstat. E por fim, um switch Gigabit que foi responsável por fazer a ligação entre as máquinas.

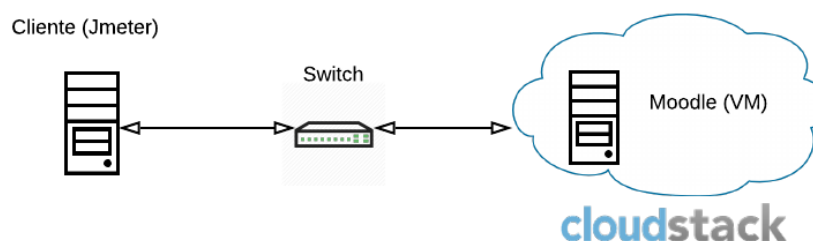


Figura 14 – Visão geral do ambiente de medição

Analisar resultados: A análise dos resultados foi feita com cálculos estatísticos de média, desvio padrão, intervalo de confiança. E ao final são plotados os gráficos de comparação dos resultados com tempo de resposta, vazão, utilização de CPU, utilização da memória e taxa de escrita no HD.

Criar modelo: Esta fase corresponde a criação de modelos SPN de desempenho e de disponibilidade e também a criação de RBDs para cálculo da disponibilidade e do MTTF e MTTR de todo o sistema. Vale ressaltar que este modelo de desempenho em SPN é refinado na fase seguinte com a adoção de técnicas de regressão linear. O objetivo é conseguir representar o tempo diferente que o sistema demanda quando se trabalha com uma diferente quantidade de usuários no sistema. Sendo assim, o tempo de algumas atividades pode ser representado através de uma equação linear.

Calcular regressão linear: Realização do cálculo da regressão linear entre a quantidade de clientes no sistema e o tempo de atendimento. Ao final foi gerado uma equação que será utilizado pelo modelo de desempenho.

Simular o modelo: Será realizado a simulação com os mesmos cenários da medição no modelo SPN de desempenho e de disponibilidade. Sendo assim, o mesmo plano de testes realizado na medição foi adotado para a realização da simulação dos modelos propostos.

Validar o modelo: Para se realizar a validação do modelo proposto foi adotada a seguinte estratégia:

- (i) Deve-se computar o intervalo de confiança para cada métrica de interesse tanto nos resultados do modelo, como também nos resultados da medição;
- (ii) Compara-se intervalos de confiança e, se estiverem sobresscrevendo, pode-se dizer que o modelo consegue representar o sistema modelado.

4.2 Considerações Finais

Este capítulo apresentou as atividades e os objetos que compõem a metodologia para avaliação de desempenho do ambiente virtual de aprendizagem em nuvem privada. Com as atividades apresentadas é possível criar modelos SPN e RBD para a avaliação de desempenho e de disponibilidade do sistema.

5 Modelos

Este capítulo apresenta o modelo de desempenho para o Moodle em nuvem privada, mostrando as métricas, parâmetros e a regressão linear utilizada. Também são apresentados os dois modelos RBDs para realizar o cálculo da disponibilidade do sistema e um modelo SPN para representação da disponibilidade.

5.1 Modelo de Desempenho

A Figura 15 ilustra o modelo SPN de desempenho proposto para representar o comportamento do Moodle nos cenários analisados. O modelo representa o funcionamento da chegada de clientes até a resposta do AVA. Esse cenário mostra que um cliente que chega vai fazer seis requisições que são: acessar o Moodle, realizar login, entrar no curso, entrar na atividade, realizar o upload de arquivo e realizar *logout*. E, ao final, o sistema envia a resposta ao cliente.

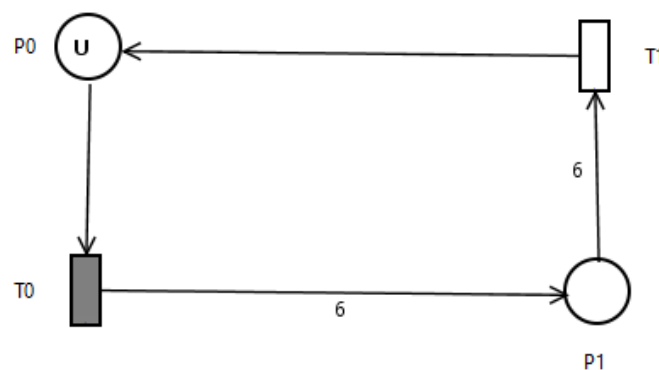


Figura 15 – Modelo de desempenho proposto.

O número de *tokens* presente no P0 representa a quantidade de clientes (U) que serão atendidos pelo sistema. A transição determinística T0 representa o tempo de chegada de usuário. Ao ser disparada a transição T0, um *token* é consumido de P0 e seis *tokens* são gerados em P1. O lugar P1 representa a fila de requisições a serem atendidas pelo sistema. A transição T1 representa o tempo demandado para atender os seis passos do cenário analisado (acessar o Moodle, se logar, selecionar um curso e uma atividade, fazer o *upload* de um arquivo e sair do sistema). O disparo desta transição T1 faz com que seis *tokens* sejam consumidos do lugar P1 e gera um *token* no lugar P0. Como pode ser visto na Figura 15, o arco que liga P0 a T0 tem peso um, onde representa a chegada de um cliente, já o arco que liga T0 a P1 tem peso

seis, que representa que o cliente vai realizar seis atividades no Moodle, o arco que une P1 a T1 também possui peso seis, pois o sistema realizará o atendimento dessas atividades e o arco que liga T1 a P0 possui peso 1 que significa que o sistema atendeu um cliente.

A Tabela 3 apresenta a descrição dos lugares do modelo e a Tabela 4 descreve as transições do modelo. Os parâmetros utilizado no *token* inicial do lugar P0 e nos tempos da transição serão explicados na subseção a seguir.

Tabela 3 – Descrição dos lugares do modelo.

Lugar	Nº Token Inicial	Descrição
P0	U	Quantidade de clientes que serão atendidos pelo sistema
P1	0	Fila de requisições a serem atendidas pelo sistema

Tabela 4 – Descrição das transições do modelo.

Transição	Tipo	Tempo	Tipo Serv.	Descrição
T0	Determinística	t_{Cheg}	Single Server	Chegada de 1 cliente no sistema
T1	Exponencial	t_{Aten}	Infinite Server	Atendimento de 1 cliente

5.1.1 Parâmetros e Métricas do Modelo

O modelo possui três parâmetros que devem ser especificados para realizar a análise do modelo de desempenho. O modelo também possui duas métricas, uma é responsável por calcular o tempo de resposta, e a outra é destinada para o cálculo da vazão do sistema.

O primeiro parâmetro é t_{Cheg} da ferramenta que representa o tempo de chegada de clientes ao sistema. Este tempo é definido no ambiente da ferramenta JMeter na construção do plano de teste. O tempo utilizado para este parâmetro é de 0,1 segundos e será inserido na transição determinística T0. Com o tempo de chegada é possível calcular a taxa de chegada de clientes que é dado por $1/t_{Cheg}$. O segundo parâmetro é U que representa a quantidade de clientes que serão atendidos pelo sistema. O terceiro e último parâmetro é t_{Aten} que é o tempo de atendimento de um cliente e é definido por uma equação de regressão linear que será apresentado na subseção a seguir.

A métrica do modelo $tempResp$ é responsável pelo cálculo do tempo de resposta. O tempo de resposta representa o período entre o envio da requisição para o sistema até a chegada da resposta para o cliente (BARROS, 2018). Seguindo os conceitos da

Lei de Little, a fórmula do tempo de resposta é definida na Equação 5.1.

$$tempResp = \frac{E\{\#P1\}}{P\{\#P0 > 0\} \times W(T0)} \quad (5.1)$$

Onde $E\{\#P1\}$ indica o número médio de tokens no lugar P1. O $P\{\#P0 > 0\}$ indica a probabilidade de ter token no lugar P0 multiplicado pela taxa de chegada de clientes (W). Este tempo de resposta pode ser dado, por exemplo, em segundos.

A métrica do modelo ST 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 na Equação 5.2.

$$ST = \frac{E\{\#P1\}}{T1} \quad (5.2)$$

Onde $E\{\#P1\}$ indica o número médio de tokens no lugar P1. O $T1$ indica o tempo que o sistema leva para atender as requisições. A vazão é dada, por exemplo, por requisições por segundo.

5.1.2 Regressão Linear

A Figura 16 apresenta a regressão linear entre a quantidade de clientes (x , independente) e o tempo de atendimento de um cliente (y , dependente).

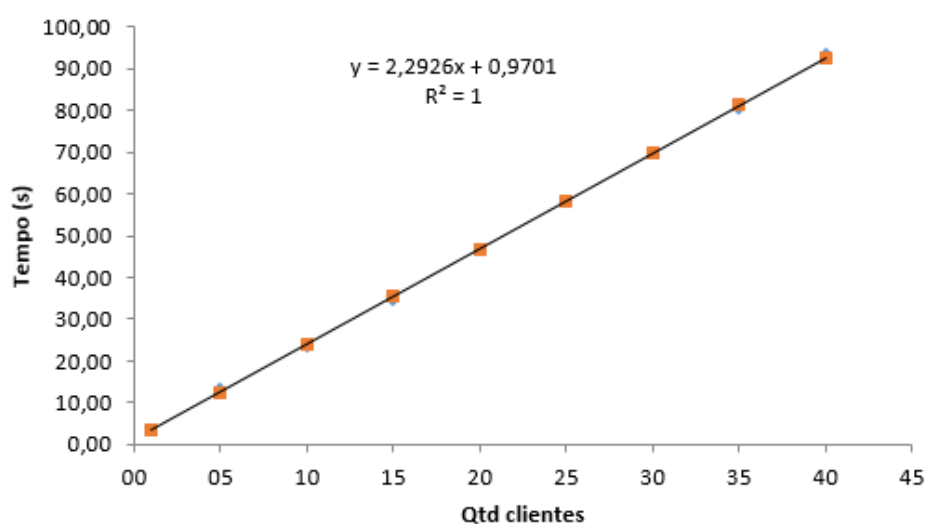


Figura 16 – Regressão linear.

Pode-se notar que os valores de y previsto pela regressão está praticamente se sobrepondo os valores de tempo de atendimento feito pela medição e o valor da R^2 foi de 0,9994. Assim, R^2 bem próximo ao valor um, indica a validade da regressão.

A regressão linear entre a quantidade de clientes e o tempo de atendimento de um cliente é dada pela Equação 5.3.

$$y = 2,2926 \times U + 0,9701 \quad (5.3)$$

Onde y é a previsão do tempo de atendimento de um cliente e U é a quantidade de clientes no sistema. A Equação 5.3 será utilizado na transição T1 do modelo de desempenho (Ver Figura 15).

5.2 Modelo de Disponibilidade

A Figura 17 mostra o modelo RBD proposto para calcular a disponibilidade, o MTTF e o MTTR do ambiente de nuvem Apache CloudStack. O modelo é composto dos componentes: hardware (HW), sistema operacional (SO), gerenciador da nuvem (CONTROLLER) e máquina virtual (VM). O modelo operacional do sistema é $OP_Sys = HW \wedge SO \wedge CONTROLLER \wedge VM$. A Tabela 5 apresenta os parâmetros utilizados no modelo RBD da Figura 17 e suas descrições.

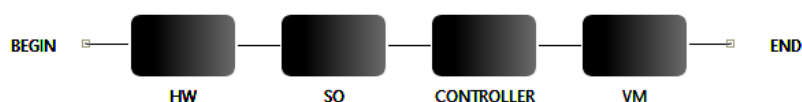


Figura 17 – Modelo RBD da plataforma Apache CloudStack.

Tabela 5 – Parâmetros do modelo RBD da plataforma CloudStack.

Parâmetro	Descrição
$MTTF_{HW}$	Tempo médio para defeito do hardware
$MTTR_{HW}$	Tempo médio para reparo do hardware
$MTTF_{SO}$	Tempo médio para defeito do sistema operacional
$MTTR_{SO}$	Tempo médio para reparo do sistema operacional
$MTTF_{CO}$	Tempo médio para defeito do gerenciador da nuvem
$MTTR_{CO}$	Tempo médio para reparo do gerenciador da nuvem
$MTTF_{VM}$	Tempo médio para defeito do máquina virtual
$MTTR_{VM}$	Tempo médio para reparo do máquina virtual

A Figura 18 ilustra o modelo RBD da máquina virtual (VM). A VM é composta dos componentes hypervisor (KVM), sistema operacional (SO), servidor web (WEB) e serviço (SERVIÇO). O software usado para o servidor web e o serviço são respectivamente, Apache e o Moodle. O modelo operacional da VM é $OP_Vm = KVM \wedge SO \wedge WEB \wedge SERVIÇO$. A Tabela 6 apresenta os parâmetros utilizados no modelo RBD da Figura 18 e suas descrições.

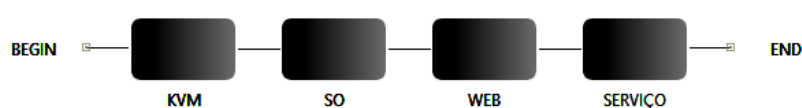


Figura 18 – Modelo RBD da máquina virtual.

Tabela 6 – Parâmetros do modelo RBD da VM.

Parâmetro	Descrição
$MTTF_{KVM}$	Tempo médio para defeito do hypervisor
$MTTR_{KVM}$	Tempo médio para reparo do hypervisor
$MTTF_{SO}$	Tempo médio para defeito do sistema operacional
$MTTR_{SO}$	Tempo médio para reparo do sistema operacional
$MTTF_{WEB}$	Tempo médio para defeito do servidor web
$MTTR_{WEB}$	Tempo médio para reparo do servidor web
$MTTF_{SER}$	Tempo médio para defeito do serviço
$MTTR_{SER}$	Tempo médio para reparo do serviço

A Tabela 7 apresenta o resultado da avaliação da disponibilidade do modelo RBD da máquina virtual (ver Figura 18). O tempo médio de falha (MTTF) desse modelo ficou em 720 horas, enquanto o tempo médio de reparo (MTTR) ficou em 8 horas.

Tabela 7 – Resultado da disponibilidade do modelo da Figura 18.

Parâmetro	Valor
MTTF	2592000s
MTTR	28800s

A Tabela 8 a seguir, mostra os parâmetros utilizado para realização do cálculo da disponibilidade da plataforma Apache CloudStack. Os tempos utilizados nesses

parâmetros foram retirados de (TORRES et al., 2016; MACIEL, 2016; SOUSA, 2015).

Tabela 8 – Parâmetros de dependabilidade dos componentes da plataforma Apache CloudStack.

Componente	MTTF (s)	MTTR (s)
HW	31.536.000	5.976
SO	10.414.800	900
Controller	529.272	3.528
VM	2.592.000	28.800

A Tabela 9 mostra o resultado da avaliação da disponibilidade, onde é apresentado o tempo médio de falha (MTTF), o tempo médio de reparo (MTTR) e a porcentagem da disponibilidade da plataforma. Com os parâmetros utilizado a disponibilidade do Apache CloudStack ficou em 98,24%, o tempo médio de defeito ficou em 115,60 horas e o tempo médio de reparo ficou em 2,07 horas.

Tabela 9 – Resultado da disponibilidade do modelo da Figura 17.

Parâmetro	Valor
MTTF	416160,80s
MTTR	7465,54s
Disponibilidade	98,24%

A Figura 19 apresenta o modelo em SPN que representa os componentes da plataforma Apache CloudStack (ver Figura 17). O lugar CloudStack_On representa que o sistema está em funcionamento. A transição exponencial MTTF representa o tempo médio de falha do sistema. O lugar CloudStack_Off representa que o sistema não está em funcionamento. E por fim a transição MTTR indica o tempo médio de reparo do sistema. A Tabela 10 apresenta os tempos associados a cada transição do modelo de disponibilidade. Vale destacar que este mesmo modelo vai ser associado ao modelo de desempenho (ver Figura 15) para se quantificar o impacto da disponibilidade no desempenho do sistema (ver Seção 6.4).

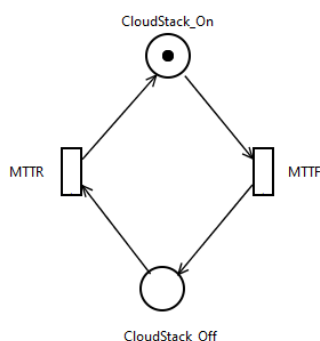


Figura 19 – Modelo de disponibilidade SPN.

Tabela 10 – Transições do modelo de disponibilidade.

Transição	Tipo	Tipo Serv.	Tempo (s)
MTTF	Exponencial	Single Server	416160,80
MTTR	Exponencial	Single Server	7465,54

5.3 Considerações Finais

Este capítulo apresentou o modelo de desempenho, os dois parâmetros e as métricas de tempo de resposta e de vazão que foram utilizadas. Também foram mostrados os modelos em RBD para o sistema e para a máquina virtual. Além disso, também foi apresentado o modelo SPN para representar a plataforma do Apache CloudStack.

6 Estudo de Caso

Este capítulo apresenta quatro estudos de caso. O primeiro estudo de caso apresenta a avaliação da capacidade do sistema utilizando métricas de vazão e tempo de resposta. O segundo estudo de caso mostram-se os resultados das medições das métricas de utilização de CPU, utilização de memória e taxa de escrita no HD. O terceiro estudo de caso apresenta a validação do modelo de desempenho. No quarto e último estudo de caso, mostra-se o impacto causado pela disponibilidade no desempenho.

6.1 Análise da Capacidade do Sistema

O estudo de caso apresentado nesta seção avalia a capacidade do sistema Moodle em uma máquina virtual com duas configurações distintas. A primeira configuração tem um processador com 1 GHz com 1 core e 1 GB de memória RAM, já a outra possui o mesmo processador da configuração anterior mas altera o tamanho da memória RAM para 2GB. Os testes realizados nessa etapa foi de vários clientes tentando somente acessar o Moodle. A Figura 20 mostra o resultado da comparação entre as duas configurações com a métrica da vazão.

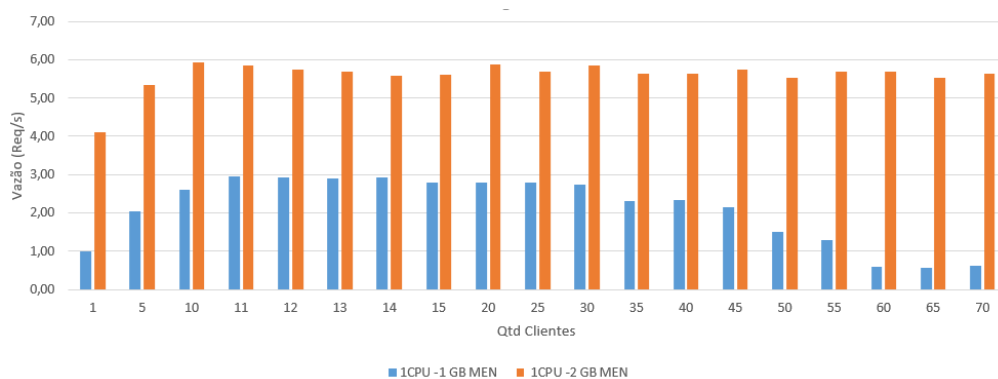


Figura 20 – Comparação da vazão.

Como pode-se notar na Figura 20, a máquina virtual que possui 1 GB de memória RAM teve um crescimento até onze clientes, manteve-se estável a vazão até trinta clientes e a partir de trinta e cinco clientes começou ter queda. Enquanto a máquina virtual com 2 GB de memória RAM também teve um crescimento até dez clientes e durante o aumento da quantidade de clientes se manteve praticamente com a mesma vazão. A Figura 21 apresenta a comparação entre as duas configurações, mas utilizando a métrica do tempo de resposta.

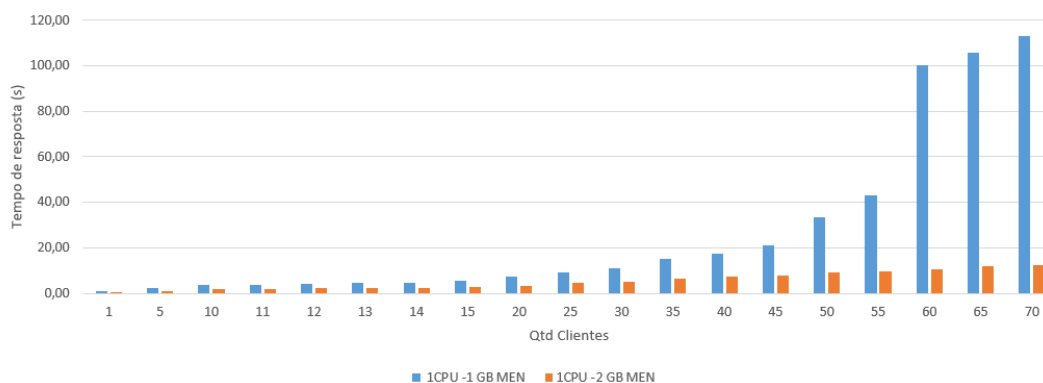


Figura 21 – Comparação do tempo de resposta.

Como pode-se notar na Figura 21 a máquina virtual que possui 1 GB de memória RAM apresentou valores inferiores a 20 segundos no tempo de resposta até 40 clientes e a partir dos 45 clientes percebe-se um aumento, onde ocorreu que o tempo mais que duplicou de 55 para 60 clientes. Já na segunda configuração o tempo de resposta tem valores inferiores a 20 segundos em todas as quantidades de clientes.

6.2 Análise da Medição de CPU, Memória e Escrita do HD

O estudo de caso apresentado nesta seção avalia os resultados da medição das métricas de CPU, memória e escrita no HD. O objetivo de estudo foi analisar o comportamento do sistema simulando uma turma de quarenta alunos realizando as atividades proposta na Seção 4.1. A Figura 22 ilustra a visão geral do ambiente de medição adicionando a ferramenta de medição vmstat na máquina virtual, que será utilizada para a coleta das métricas propostas. A máquina virtual possui a mesma configuração que é mostrada na seção 4.1.

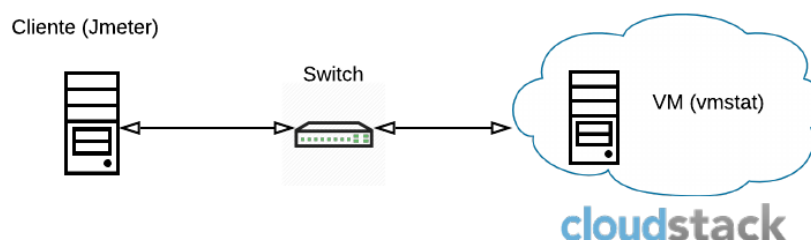


Figura 22 – Ambiente de medição com vmstat.

A Figura 23 mostra os resultados da medição da utilização da CPU. Como pode ser notado, um cliente utiliza quase 50% do processador. Nos outros casos o nível de

utilização se estabiliza em torno dos 95%, pois o sistema entra em saturação. Com um maior número de clientes, o processador pode virar um gargalo para o sistema, e assim prejudicar o desempenho.

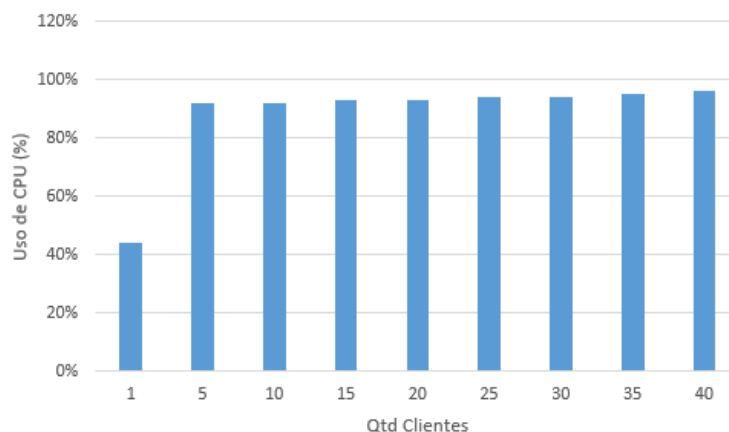


Figura 23 – Resultado da medição da utilização da CPU.

A Figura 24 mostra os resultados da medição da utilização da memória. A medida que a quantidade de clientes aumenta, a utilização da memória cresce junto. Mesmo com uma grande quantidade de clientes, o máximo de utilização de memória foi de 52%.

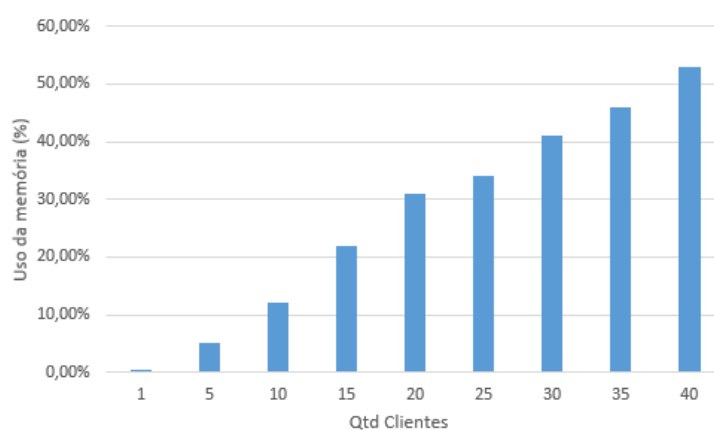


Figura 24 – Resultado da medição da utilização da memória.

A Figura 25 apresenta os resultados da medição da escrita do HD. A ferramenta vmstat utiliza a unidade de medida de blocos para representar a quantidade de escrita do HD. Um bloco é que corresponde a 1024 bytes. Do mesmo jeito que acontece

na utilização de memória, a quantidade de blocos escritos aumenta de acordo com a quantidade de clientes.

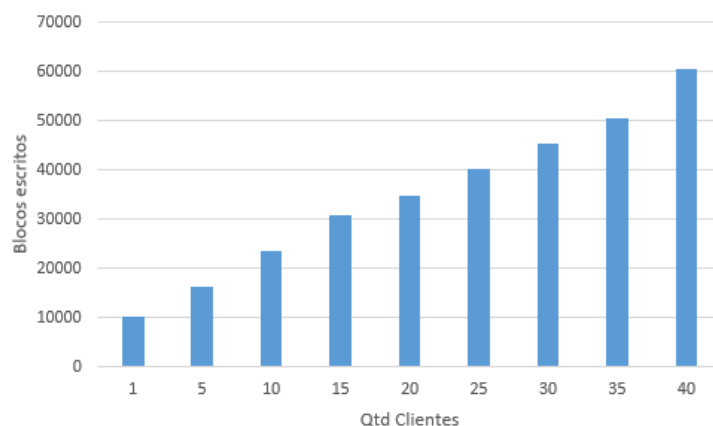


Figura 25 – Resultado da medição da escrita no HD.

6.3 Validação de Modelo

Esta seção apresenta o estudo de caso para a validação do modelo de desempenho (ver Figura 15). A validação é um procedimento utilizado para certificar que os valores gerados pelo modelo apresentam coerência com os gerados pelo sistema real. Para ocorrer a validação do modelo, o mesmo deve ser executado nas mesmas condições impostas ao sistema. A validação desse modelo foi feita com a comparação do tempo de resposta da medição com o tempo de resposta da simulação estacionária do modelo. A configuração utilizada na máquina virtual é a mesma apresentada na seção 4.1.

A Figura 26 apresenta os resultados da medição e da modelagem. O nível de confiança dos resultados é de 95% e o erro relativo da simulação 10%. A Tabela 11 mostra em maior detalhes os resultados da medição e da modelagem e seus respectivos intervalo de confiança. Para ocorrer a validação do modelo, os intervalos de confiança devem se sobrepor, como acontece na Tabela 11.

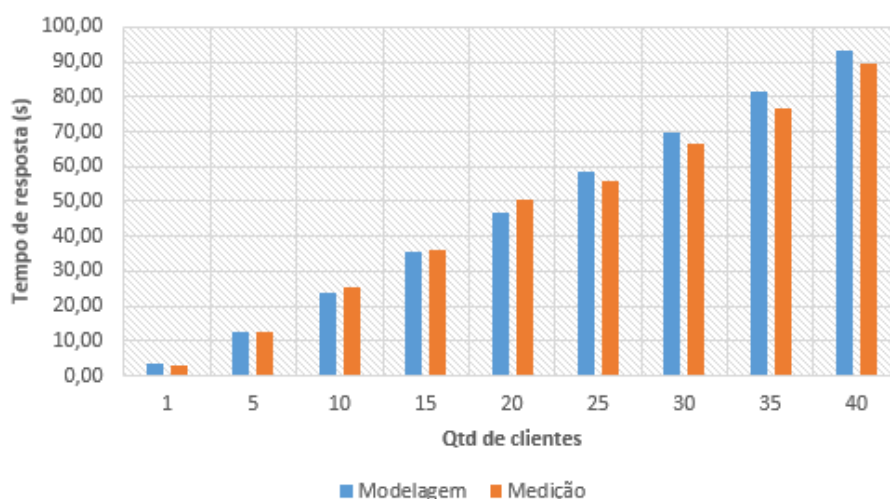


Figura 26 – Resultado da medição e modelagem.

Tabela 11 – Resultado para validação do modelo com tempo de resposta.

Qtd. Clientes	Medição		Modelagem	
	Tempo (s)	Int. Conf.	Tempo (s)	Int. Conf.
1	3,12	[2,73; 3,50]	3,27	[3,27; 3,28]
5	12,37	[10,29; 14,45]	12,39	[12,35; 12,44]
10	25,55	[22,54; 28,56]	23,97	[23,90; 24,04]
15	36,04	[35,20; 36,89]	35,47	[35,45; 35,48]
20	50,54	[46,96; 54,13]	46,78	[46,43; 47,13]
25	55,70	[53,25; 58,15]	58,52	[57,79; 59,25]
30	66,43	[62,80; 70,05]	69,55	[69,41; 69,68]
35	76,92	[73,27; 81,57]	81,51	[80,46; 82,26]
40	89,31	[79,95; 98,67]	93,01	[92,07; 93,96]

6.4 Impacto da Disponibilidade no Modelo de Desempenho

Esta seção apresenta o estudo de caso que tem como objetivo mostrar o impacto causado no modelo de desempenho com a inserção do modelo de disponibilidade. A Figura 27 ilustra o modelo proposto, onde existe uma ligação entre os dois modelos com um arco inibidor.

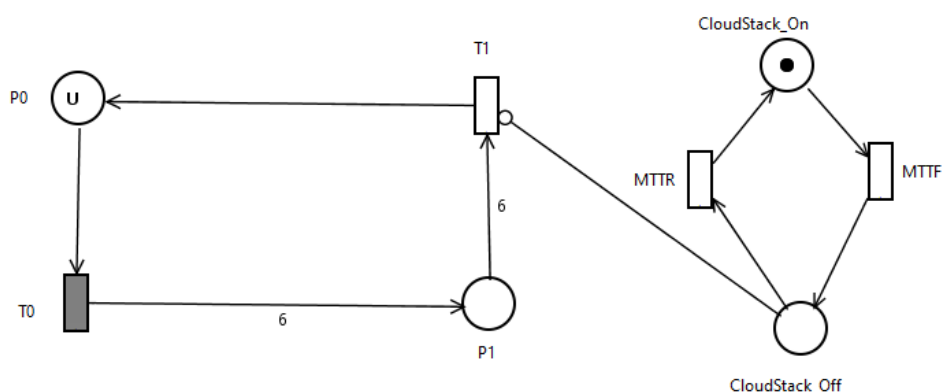


Figura 27 – Modelo com disponibilidade.

A transição T1 só estará ativa quando não houve token no lugar CloudStack_Off, quando a transição MTF disparar o token irá para o lugar CloudStack_Off e o sistema ficará desativado. Quando a transição MTR disparar o token voltará para o lugar CloudStack_On e o sistema irá voltar a funcionar. A Tabela 12 mostra a comparação com os resultados do tempo de resposta das simulação feita sem o modelo de disponibilidade realizada na validação, com a simulação feita com o modelo de disponibilidade.

Tabela 12 – Comparação do tempo de resposta com e sem o modelo de disponibilidade.

Qtd. Clientes	T. de Resp. sem Disp.(s)	T. de Resp. com Disp.(s)	Impacto (%)
1	3,27	3,31	1,22%
5	12,39	12,77	3,07%
10	23,97	24,31	1,42%
15	35,47	36,23	2,14%
20	46,78	47,52	1,52 %
25	58,52	60,44	3,28%
30	69,55	70,95	2,01%
35	81,51	82,87	1,67%
40	93,01	93,76	0,81%

O nível de confiança dos resultados continua sendo de 95% e com o erro relativo da simulação de 10%. Como pode-se observar nos resultados na Tabela 12, ocorreu um aumento médio de quase 2% no tempo de resposta. Comprovando, assim, que a disponibilidade afeta o desempenho do sistema.

6.5 Considerações Finais

Este capítulo apresentou os quatro estudos de casos realizados para este trabalho. No primeiro estudo de caso foi feita uma análise da capacidade do sistema com duas configurações diferentes para a máquina virtual e foi feita a comparação utilizando as métricas de vazão e tempo de resposta. No segundo estudo de caso foi feita uma análise das medições das métricas da utilização da CPU, utilização de memória e a quantidade de blocos escritos no HD. No terceiro estudo de caso foi realizado a validação do modelo de desempenho proposto para o Moodle em ambiente de nuvem privada. E no último estudo de caso foi realizado a junção dos modelos de desempenho e disponibilidade para verificar o impacto da disponibilidade no desempenho do sistema.

7 Conclusão

Com o crescimento da computação em nuvem, que fornece compartilhamento de recursos computacionais através da Internet, o investimento do mercado nesta tecnologia era avaliada em 67 bilhões de dólares em 2015, e já esta cotada para 2020 em 162 bilhões de dólares (EXAME, 2018). O Brasil tende a acompanhar esse crescimento mundial, onde em 2016 da vigésima segunda posição passou para décimo oitavo em 2018 (BSA, 2018).

Assim como a computação em nuvem, a EaD também está em constante crescimento. A expansão de alunos matriculados em graduação a distância foi de quase 18% entre 2016 e 2017 (G1, 2018). Com isso os ambientes virtuais de aprendizagem, que representam o recurso utilizado para interação entre alunos e professores, sofreram um forte aumento em sua demanda.

Na avaliação de desempenho, técnicas de modelagem e simulação são utilizadas para ajudar nesse processo. Este trabalho propôs modelos SPN e RBD para avaliação de desempenho e disponibilidade do ambiente virtual de aprendizagem Moodle em nuvem privada. Também foi avaliado o impacto que o modelo de disponibilidade causa no tempo de resposta do modelo de desempenho. E, por fim, foi feita uma avaliação das métricas de utilização de CPU, utilização da memória e a quantidade de blocos escritos no HD.

7.1 Contribuições

Esta seção apresenta as contribuições deste trabalho:

- **Metodologia:** Foi proposta uma metodologia para a realização da avaliação de desempenho e disponibilidade, e proposição de modelos do Moodle em nuvem privada.
- **Avaliação de desempenho:** Foi avaliado o desempenho do ambiente virtual de aprendizagem Moodle em ambiente de nuvem privada verificando as seguintes métricas: tempo de resposta, vazão, utilização da CPU, utilização da memória e taxa de escrita do HD.
- **Modelo de desempenho:** Foi proposto um modelo de desempenho em SPN que simula o funcionamento do sistema Moodle e que calcula métricas de tempo de resposta e vazão.

- **Modelo de disponibilidade:** Foi proposto dois modelos RBD para avaliar a disponibilidade do sistema em questão e calcular seu tempo médio de falha e de reparo. Com os valores do MTTF e MTTR, foi proposto também um modelo de disponibilidade em SPN.

7.2 Trabalhos Futuros

Como trabalhos futuros, pode-se avaliar o Moodle com um número maior de clientes, já que neste trabalho foi considerado uma turma de quarenta alunos. Pode ser utilizado outros softwares de computação em nuvem como OpenStack ou OpenNebula e realizar a comparação de desempenho entre eles, e assim definir qual ambiente de nuvem é mais propício ao uso. Também pode ser utilizados outros tipos de nuvens como nuvem híbrida. A modelagem também pode ser feita com extensões de rede de Petri, como por exemplo, rede de Petri colorida. Também pode-se inserir no contexto o consumo de energia e ver qual entrega o menor consumo para ao usuário.

A avaliação de desempenho comparando uma máquina virtual com uma configuração computacionalmente grande, com várias máquinas virtuais com configurações pequenas. Também pode ser realizado uma avaliação de instâncias com elasticidade.

Referências

- AMAZON. *Amazon EC2*. 2018. Disponível em: <<https://aws.amazon.com/pt/ec2/>>. Citado na página 21.
- APACHE. *Apache CloudStack*. 2017. Disponível em: <<https://cloudstack.apache.org/>>. Citado 5 vezes nas páginas 15, 21, 22, 23 e 24.
- APACHE. *Jmeter*. 2019. Disponível em: <<https://jmeter.apache.org/index.html>>. Citado na página 26.
- AYYAD, Y. Security concerns in a web-based e-learning platform. 02 2016. Citado na página 25.
- BARROS, J. R. A. de. *Modelagem de Desempenho do Banco de Dados Cassandra*. Dissertação (Mestrado) — Universidade Federal Rural de Pernambuco, 2018. Citado 3 vezes nas páginas 19, 25 e 38.
- BAUER, E.; ADAMS, R. *Reliability and availability of cloud computing*. [S.l.]: John Wiley & Sons, 2012. Citado na página 14.
- BELUCE, A. C.; OLIVEIRA, K. L. d. Ambientes virtuais de aprendizagem: Das estratégias de ensino às estratégias de aprendizagem. 2012. Citado na página 24.
- BLUE, S. *Apache CloudStack*. 2013. Disponível em: <<https://www.shapeblue.com/pt-br/cloudstack-101-2/>>. Citado na página 23.
- BSA. *BSA Global Cloud Computing Scorecard*. 2018. Disponível em: <https://cloudscorecard.bsa.org/2018/?sc_lang=pt-BR>. Citado na página 51.
- CALLOU, G. et al. Energy consumption and execution time estimation of embedded system applications. *Microprocessors and Microsystems*, Elsevier, v. 35, n. 4, p. 426–440, 2011. Citado 2 vezes nas páginas 25 e 26.
- CAMPOS, E. G. *Performance evaluation of auto scaling mechanisms in private clouds for supporting a web service application*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2015. Citado 2 vezes nas páginas 18 e 19.
- CHANDEL, V.; PATIAL, S.; GULERIA, S. Comparative study of testing tools: Apache jmeter and load runner. *International Journal of Computing and Corporate Research*, v. 3, n. 3, 2013. Citado na página 26.
- DANTAS, J. R. Planejamento de infraestrutura de nuvens computacionais para serviço de vod streaming considerando desempenho, disponibilidade e custo. Universidade Federal de Pernambuco, 2018. Citado 3 vezes nas páginas 14, 18 e 19.
- DROPBOX. *Dropbox*. 2018. Disponível em: <https://www.dropbox.com/pt_BR/>. Citado na página 22.
- EUCALYPTUS. *Eucalyptus Cloud*. 2019. Disponível em: <<https://www.eucalyptus.cloud/>>. Citado na página 15.

EXAME. *Mercado da computação em nuvem deve chegar a 162 bilhões de dólares em 2020: qual o impacto da Cloud Act?* 2018. Disponível em: <<https://exame.abril.com.br/negocios/dino/mercado-da-computacao-em-nuvem-deve-chegar-a-162-bilhoes-de-dolares-em-2020-qual-o-imp>>. Citado na página 51.

FÉ, I. d. S. *Planejamento de transcodificação de vídeo em nuvem elástica*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017. Citado 2 vezes nas páginas 18 e 19.

G1. *Aumento das matrículas na graduação a distância é o maior desde 2008, aponta censo*. 2018. Disponível em: <<https://g1.globo.com/educacao/guia-de-carreiras/noticia/2018/09/20/aumento-das-matriculas-na-graduacao-a-distancia-e-o-maior-desde-2008-aponta-censo.ghtml>>. Citado na página 51.

GERMAN, R. *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. New York, NY, USA: John Wiley & Sons, Inc., 2000. ISBN 0471492582. Citado na página 28.

GOOGLE. *Google Apps*. 2018. Disponível em: <<https://gsuite.google.com.br/intl/pt-BR/>>. Citado na página 22.

GOOGLE. *Google Cloud*. 2018. Disponível em: <<https://cloud.google.com/?hl=pt-br>>. Citado na página 21.

GOOGLE. *Google Cloud Platform*. 2018. Disponível em: <<https://edu.google.com/intl/pt-BR/products/google-cloud-platform/>>. Citado na página 21.

HAGUENAUER, C. J.; LIMA, L. G. R.; FILHO, F. C. Comunicação e interação em ambientes virtuais de aprendizagem. In: *Anais do Congresso Internacional de Educação a Distância (CIED)*. Foz do Iguaçu-PR. [S.l.: s.n.], 2010. Citado na página 24.

JAIN, R. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. [S.l.]: Wiley, 1991. I-XXVII, 1-685 p. (Wiley professional computing). ISBN 978-0-471-50336-1. Citado na página 25.

JUNIOR, D. d. A. B. d. S.; TAVARES, E. A. G. Avaliação de desempenho e dependabilidade de infraestruturas da computação em nuvem. 2015. Citado 2 vezes nas páginas 17 e 19.

JÚNIOR, L. J. et al. Uma extensão do moodle para recomendação ubíqua de objetos de aprendizagem. *RENOTE*, v. 10, n. 3, 2012. Citado na página 19.

KENSKI, V. M. *Educação e tecnologias*. [S.l.]: Papirus editora, 2007. Citado na página 24.

KIM, H. et al. A trust evaluation model for qos guarantee in cloud systems. *International Journal of Grid and Distributed Computing*, v. 3, n. 1, p. 1–10, 2010. Citado na página 14.

KUMAR, R. et al. 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*, p. 111–116, 2014. Citado na página 23.

LILJA, D. J. *Measuring Computer Performance: A Practitioner's Guide*. [S.l.]: Cambridge University Press, 2000. Citado na página 31.

MACIEL, P.; LINS, R. D.; CUNHA, P. R. F. Introdução às redes de petri e aplicações. 05 1996. Citado na página 27.

MACIEL, P. R. M. Modeling availability impact in cloud computing. In: *Principles of Performance and Reliability Modeling and Evaluation*. [S.l.]: Springer, 2016. p. 287–320. Citado na página 42.

MANSUR, A. F. U. et al. 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*. [S.l.: s.n.], 2010. p. 35. Citado na página 14.

MARCELINO, C. et al. Avaliação de desempenho da infra-estrutura do ambiente virtual de aprendizagem do cead/cefetes. *Simpósio Brasileiro de Informática na Educação*, 2008. Citado 2 vezes nas páginas 17 e 19.

MARON, C. A. et al. Avaliação e comparação do desempenho das ferramentas openstack e opennebula. *12th Escola Regional de Redes de Computadores (ERRC)*, p. 1–5, 2014. Citado 2 vezes nas páginas 17 e 19.

MATOS, M. A. Manual operacional para a regressão linear. *Faculdade de Engenharia da Universidade do Porto*, p. 63, 1995. Citado na página 31.

MICROSOFT. *Microsoft Azure*. 2018. Disponível em: <<https://azure.microsoft.com/pt-br/>>. Citado na página 21.

MICROSOFT. *Microsoft Office 365*. 2018. Disponível em: <<https://www.office.com/>>. Citado na página 22.

MOODLE. *Moodle*. 2018. Disponível em: <https://docs.moodle.org/36/en/About_Moodle>. Citado na página 24.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, April 1989. ISSN 0018-9219. Citado 2 vezes nas páginas 26 e 27.

NIGAM, R. *Performance Test in Distributed Environment*. 2014. Disponível em: <<https://www.infobeans.com/performance-test-distributed-environment>>. Citado na página 26.

NIST. *NIST Cloud Computing Program*. 2014. Disponível em: <<http://www.nsti.gov/itl/cloud/index.cfm>>. Citado 2 vezes nas páginas 21 e 22.

OPENSTACK. *OpenStack*. 2019. Disponível em: <<https://www.openstack.org/>>. Citado na página 15.

- REDHAT. *Red Hat OpenShift*. 2018. Disponível em: <<https://www.redhat.com/pt-br/technologies/cloud-computing/openshift>>. Citado na página 21.
- ROSSO, J. P. Análise de desempenho de aplicações científicas em ambiente de nuvem privada. Universidade Federal do Pampa, 2015. Citado 2 vezes nas páginas 18 e 19.
- RUSCHEL, H.; ZANOTTO, M. S.; MOTA, W. d. Computação em nuvem. *Curitiba, abr*, p. 1–3, 2010. Citado na página 21.
- SCHOUTEN, E. *Cloud computing defined: Characteristics service levels*. 2014. Disponível em: <<https://www.ibm.com/blogs/cloud-computing/2014/01/31/cloud-computing-defined-characteristics-service-levels/>>. Citado na página 22.
- SILVA, B. A framework for availability, performance and survivability evaluation of disaster tolerant cloud computing systems. Universidade Federal de Pernambuco, 2016. Citado na página 28.
- SILVA, S. Acessibilidade digital em ambientes virtuais de aprendizagem. *Revista GEINTEC-Gestão, Inovação e Tecnologias*, v. 2, n. 3, p. 245–254, 2012. Citado na página 24.
- SOUSA, E. T. G. *Modelagem de desempenho, dependabilidade e custo para o planejamento de infraestruturas de nuvens privadas*. Tese (Doutorado) — Federal University of Pernambuco, Recife, Brazil, 2015. Citado 3 vezes nas páginas 17, 19 e 42.
- TORRES, E. et al. Performance and availability evaluation of storage services in private cloud. In: *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.: s.n.], 2016. p. 1–6. Citado 5 vezes nas páginas 18, 19, 25, 28 e 42.
- TRIVEDI, K. S. et al. *Reliability Analysis Techniques Explored Through a Communication Network Example*. 1996. Citado na página 30.
- UFRPE. *UFRPE lança Ambiente Virtual de Suporte à Aprendizagem*. 2015. Disponível em: <<http://ufrpe.br/br/content/ufrpe-lanca-ambiente-virtual-de-suporte-a-aprendizagem>>. Citado na página 15.
- UFRPE. *Relatório de Gestão*. 2017. Disponível em: <http://www.proplan.ufrpe.br/sites/www.proplan.ufrpe.br/files/relatoriogestao_ufrpe_2017.pdf>. Citado na página 15.
- XIONG, K.; PERROS, H. Service performance and analysis in cloud computing. In: *2009 Congress on Services - I*. [S.l.: s.n.], 2009. p. 693–700. ISSN 2378-3818. Citado na página 14.
- ZIMMERMANN, A. et al. Towards version 4.0 of timenet. In: *MMB*. [S.l.: s.n.], 2006. Citado na página 30.