

AVALIAÇÃO DE DESEMPENHO DE TÉCNICAS DE MIGRAÇÃO DE MÁQUINAS VIRTUAIS PARA O AMBIENTE DE COMPUTAÇÃO EM NUVEM¹

João Batista²
Wellington Junior³
Carlos H. G. Ferreira⁴
Danilo Costa⁵
Maycon Peixoto⁶
Bruno G. Batista⁷
Bruno T. Kuehne⁸
Dionísio M. Leite⁹

RESUMO

As ferramentas de virtualização têm um papel fundamental no crescimento da utilização da Computação em Nuvem. Por meio da virtualização, é possível realizar a migração de máquinas virtuais dentro de um provedor de serviços de nuvem, proporcionando a utilização eficiente dos recursos. Entretanto, não foram evidenciadas quais técnicas são mais indicadas de acordo com os cenários distintos de carga no qual o sistema está operando. Diferentemente das avaliações de desempenhos simplesmente comparativas encontradas na literatura, este trabalho propõe a utilização de um modelo estatístico consistente para avaliação de desempenho das técnicas de migração: (i) *live migration* e (ii) *non-live migration*. O objetivo do modelo estatístico é identificar o comportamento das técnicas de migração de máquinas virtuais sob diferentes situações de cargas de trabalho. O modelo estatístico proposto para utilização é composto pela abordagem de avaliação de desempenho das técnicas de migração de máquina virtual combinado com o planejamento de experimentos de projeto fatorial 2^k .

Palavras-chave: Migração. Máquinas Virtuais. Avaliação de Desempenho.

1 Como citar este artigo:

BATISTA, J. et al. Avaliação de desempenho de técnicas de migração de máquinas virtuais para o ambiente de computação em nuvem. **ForScience**: revista científica do IFMG, Formiga, v. 4, n. 2, e00186, jul./dez. 2016.

² Mestrando em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos-SP. E-mail: joao.b@usp.br.

³ Mestrando em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos-SP, Brasil. E-mail: wellingtonjr@usp.br.

⁴ Mestre em Ciência da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos-SP, Brasil. E-mail: chgferreira@usp.br.

⁵ Mestre em Ciência da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos-SP. E-mail: dcmdancosta@usp.br.

⁶ Doutor em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos, Brasil. E-mail: mayconleone@dcc.ufba.br.

⁷ Doutor em Ciências Matemáticas e Computação pela Universidade de São Paulo (USP), São Carlos-SP, Brasil. E-mail: brunoguazzelli@unifei.edu.br.

⁸ Doutor em Ciências da Computação pela Universidade de São Paulo (USP), São Carlos-SP, Brasil. E-mail: brunokuehne@unifei.edu.br.

⁹ Doutor em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo (USP), São Carlos-SP, Brasil. E-mail: dionisio.leite@ufms.br.

1 INTRODUÇÃO

A Computação em Nuvem tem sido desenvolvida com base em muitos anos de pesquisa em diversas áreas de computação, tais como: sistemas distribuídos, virtualização, tolerância a falhas, balanceamento de carga, interoperabilidade, aplicações Web 2.0, computação utilitária e computação autônoma. Esse paradigma pode ser considerado análogo a um conjunto de recursos computacionais, que executam serviços para os usuários sob demanda. Um serviço pode ser qualquer coisa, que na computação, vai de uma aplicação simples para verificar a previsão do tempo até um ambiente de desenvolvimento integrado sob uma infraestrutura configurável. Assim, a Nuvem deve ser continuamente monitorada, avaliada e adaptada para garantir um nível adequado de Qualidade de Serviço (QoS). Além de todos esses desafios, é necessário que exista um mecanismo de gerência eficiente dos recursos, permitindo otimizar o processamento, armazenamento de dados, rede de comunicação e tecnologias fundamentais, como por exemplo, a virtualização (ZAGHLOUL, 2013).

Ao longo dos anos, os sistemas computacionais em suas diversas aplicações como jogos, serviços de hospedagem ou pesquisas no âmbito acadêmico e industrial, têm demandado cada vez mais poder computacional. Apesar do avanço tecnológico dos dispositivos de processamento e armazenamento, apenas uma fração dessa capacidade é aproveitada, deixando recursos ociosos por uma boa parte do tempo. Uma boa utilização desses recursos torna-se essencial, principalmente quando há limitações de espaço físico ou de recursos financeiros (ROSSI, 2008). A virtualização é uma solução na busca pela utilização eficiente dos recursos. Por meio de sua capacidade de executar diferentes sistemas operacionais simultaneamente sobre o mesmo *hardware*, a virtualização tem auxiliado em um melhor gerenciamento dos recursos, visando atingir um melhor desempenho e atendendo aplicações de maior custo computacional ou que demandam maior disponibilidade e segurança (SANTOS; CHARÃO, 2008; ZAGHLOUL, 2013).

Um dos benefícios da virtualização é a possibilidade de oferecer a migração de máquinas virtuais, utilizando os recursos físicos de forma eficiente (BELOGLAZOV; BUYYA, 2010). Desta forma, a técnica de migração visa reduzir a energia consumida, o que remete a uma economia de recursos, que é o foco principal da computação verde (STRUNK; DARGIE, 2013). No entanto, foi verificado que migração pode tornar-se custosa, principalmente quando ocorrem muitas requisições à memória principal da máquina virtual (VM – *Virtual Machine*) tendo que copiar todas as modificações para o *hypervisor* de destino (VOORSLUYS et al., 2009; SHAH; JAIKAR; NOH, 2015).

Dependendo da aplicação ou do ambiente no qual o sistema está operando, é aconselhável a utilização de diferentes técnicas de migração. Porém, até onde foi verificado, não foram encontradas na literatura indicações de qual técnica é a mais indicada para cada situação. Ape-

sar de ter sido detectado o problema, os estudos em [Clark et al. \(2005\)](#); [Koslovski, Bouffleur e Charão \(2006\)](#); [Neves et al. \(2007\)](#); [Voorsluys et al. \(2009\)](#); [Shah, Jaikar e Noh \(2015\)](#); [Hu et al. \(2015\)](#); [Galloway, Loewen e Vrbsky \(2015\)](#) e [Tao et al. \(2016\)](#) não apontam uma avaliação estatística que permita identificar as influências dos fatores do experimento no ambiente. O que tem sido proposto é apenas uma avaliação baseada na percepção dos usuários. Ou ainda, os estudos afirmam que as técnicas são boas, simplesmente por uma avaliação comparativa. Em todos os casos verificados não há indicação, ajustada ao caso de uso, de utilização das técnicas de migração. Os autores já referenciados utilizaram o *hypervisor Xen* em seus trabalhos além de fazerem uma avaliação superficial focada na metodologia *live migration*.

Neste contexto, este trabalho propõe a utilização de um modelo estatístico para avaliação das técnicas de migração de máquinas virtuais no contexto de computação em nuvem. *A hipótese adotada é que, por meio desse modelo estatístico, é possível quantificar estatisticamente as técnicas de migração, considerando as necessidades do usuário e a carga de trabalho imposta ao sistema virtualizado.* A abordagem adotada é submeter as técnicas de migração de máquinas virtuais a uma avaliação de desempenho com diferentes situações de cargas distintas, considerando a variação residual das variáveis de resposta como, por exemplo, o tempo total de migração e o tempo inoperante das aplicações em execução nas máquinas virtuais. Para isso, é realizada a avaliação de desempenho das técnicas de migração de máquina virtual combinada com a avaliação de desempenho por meio do planejamento de experimentos. O planejamento adotado segue o modelo de regressão a partir do projeto fatorial 2^k proposto em [Jain \(1991\)](#).

Além de fornecer a correlação entre os fatores do ambiente e as variáveis de resposta esperadas, o modelo de avaliação proposto apresenta algumas recomendações de utilização das técnicas de migração para cenários distintos por meio dos resultados obtidos, permitindo economizar recursos computacionais e financeiros.

O restante deste artigo é organizado da seguinte forma: a Seção 2 realiza um delineamento de como foi desenvolvida a análise e posteriormente, na Seção 3 são feitas as análises referentes aos resultados obtidos. A Seção 4 apresenta os trabalhos relacionados e a relação dos mesmos com o trabalho proposto, indicando o ponto onde a presente contribuição se encaixa. Na Seção 5 são apresentadas as conclusões, bem como algumas recomendações de utilização de cada técnica.

2 MIGRAÇÃO DE MÁQUINAS VIRTUAIS

Um dos benefícios da virtualização no ambiente de Computação em Nuvem é a possibilidade de realizar a migração de máquinas virtuais por meio da gestão de um *hypervisor*. De acordo com [Pchek \(2006\)](#), são funções de um *hypervisor*: (i) gerenciar tanto as máquinas virtu-

ais, quanto os recursos computacionais; (ii) modificar os parâmetros da estrutura das máquinas virtuais em execução; (iii) controlar o ciclo de vida de uma máquina virtual. É possível salvar o estado de uma máquina virtual em um computador origem e copiá-lo para outro *hypervisor* que está executando em outro computador destino, continuando o funcionamento da VM no *hypervisor* de destino.

Como representada na Figura 1, uma das técnicas de migração proposta é chamada *non-live migration*. De acordo com Magalhães, Soares e Gomes (2011), a *non-live migration* consiste em suspender o funcionamento da VM. Dessa forma, a aplicação, que está em execução na VM, fica inoperante durante todo o processo de migração e o estado da VM é copiado para o *hypervisor* de destino. Após esse procedimento, o estado da VM é retomado, diminuindo o tempo total da migração. Por outro lado, há a técnica de migração *live migration*. Na *live migration* a VM é migrada sem que o funcionamento da aplicação em execução sobre a máquina virtual seja interrompido. Porém, isto ocasiona um tempo maior para a migração, por causa da troca de páginas da memória principal.

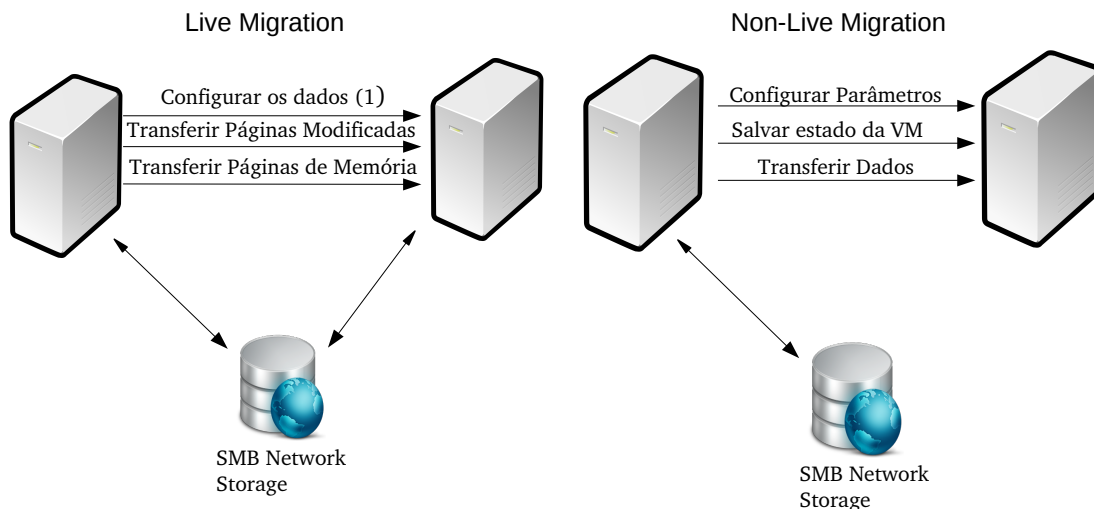


Figura 1 – Modelos de Migração de Máquinas Virtuais
Fonte: Dos autores (2015).

2.1 Modelo de Avaliação: Planejamento de Experimentos

O conjunto de dados observados $\{y_i, x_{i1}, \dots, x_{ip}\}_{i=1}^n$ representa n elementos estudados durante o planejamento de experimentos. Assume-se neste trabalho o modelo de regressão linear que é a relação entre a variável dependente y_i e o vetor de regressões lineares x_i . O

modelo geral através do erro (FREEDMAN, 2009) pode ser obtido da seguinte forma:

$$y_i = \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n, \quad (1)$$

onde T indica a transposta, de modo que $\mathbf{x}_i^T \boldsymbol{\beta}$ é o produto interno entre os vetores \mathbf{x}_i e $\boldsymbol{\beta}$. Frequentemente, essas n equações são colocadas juntas e escritas na forma vetorial como se segue:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

onde,

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Assim, por meio deste modelo de regressão, utilizou-se o planejamento de experimentos fatorial 2^k , apresentado em Jain (1991), em que são definidos dois níveis para cada fator. A Tabela 1 apresenta os fatores e seus respectivos níveis utilizados no trabalho.

Tabela 1 – Fatores e níveis

Fator	Níveis	
Técnica de Migração (TM)	<i>Live Migration</i>	<i>Non-live Migration</i>
Memória RAM (RAM)	256 MB	512 MB
Número de Máquinas Virtuais (VMs)	2	4
Carga de Trabalho (CT)	<i>SpeedRam</i>	<i>eSpeak</i>

Fonte: Dos autores (2015).

O modelo fatorial completo utiliza um arranjo de 2^4 , que é o número de níveis elevado ao número de fatores. Esse arranjo é alcançado a partir da Equação (2).

$$\begin{aligned}
 y = & q_0 + q_A x_A + q_B x_B + q_C x_C + q_D x_D + q_{AB} x_{AB} + q_{AC} x_{AC} + q_{AD} x_{AD} \\
 & + q_{BC} x_{BC} + q_{BD} x_{BD} + q_{CD} x_{CD} + q_{ABC} x_{ABC} + q_{ABD} x_{ABD} + q_{ACD} x_{ACD} \\
 & + q_{BCD} x_{BCD} + q_{ABCD} x_{ABCD}
 \end{aligned} \quad (2)$$

Substituindo-se os valores dos experimentos, obtêm-se os valores de $q_A, q_B, q_C, q_D, q_{AB}, q_{AC}, q_{AD}, q_{BC}, q_{BD}, q_{CD}, q_{ABC}, q_{ABD}, q_{ACD}, q_{BCD}, q_{ABCD}$ como mostra a Equação (3), onde é calculado o valor de q_0 .

$$\begin{aligned}
 q_0 = & 1/16 * (y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 \\
 & + y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{16})
 \end{aligned} \quad (3)$$

A partir dos valores obtidos pode-se determinar a soma dos quadrados. A variação total ou Soma Total dos Quadrados (SST) é dada pela Equação $SST = \sum_{i,j} (y_{ij} - \bar{y})$. Nesta equação, \bar{y} representa a média das respostas de todas as repetições de todos os experimentos. Com isso, a SST é dada por: $SST = 2^4 (q_A^2 + q_B^2 + q_C^2 + q_D^2 + \dots + q_{ABCD}^2)$. Por meio da utilização do modelo de regressão, a SST fornecerá a variação total das variáveis de resposta e a influência de cada fator e suas interações. Para obter a influência de um determinado fator, por exemplo o fator A, é necessário utilizar $y = SSA/SST$, onde $SSA = 2^4 * q_A^2$.

Os fatores escolhidos tem foco na influência dos recursos de memória RAM e processador. Outros fatores foram desconsiderados no artigo, tais como: a largura de banda e a utilização dinâmica da rede. Apesar de serem fatores importantes em outros cenários, não se aplicam ao caso de uma rede local estável. Foram utilizados dois *benchmarks* do *Phoronix Test Suite*¹⁰, o *SpeedRam* e o *eSpeak*, que são os responsáveis por produzir a carga de trabalho e assim permitir a análise do ambiente em situações de uso intenso do *hardware*.

O teste *SpeedRam* gera uma carga de trabalho e faz uma análise do desempenho da memória RAM de um sistema. Ele utiliza um conjunto de outros *benchmarks*, que realizam operações de leitura e escrita na memória RAM, o que possibilita mensurar o seu desempenho (HOLLANDER; BOLOTOFF, 2015). Já o *eSpeak* gera uma carga para avaliar o desempenho do processador. Ele mensura o tempo necessário para que o sintetizador de fala *eSpeak* converta o *eBook Project Gutenbergs The Outline of Science* em arquivo de áudio no formato WAV

¹⁰ <http://www.phoronix-test-suite.com/documentation/phoronix-test-suite.html>

(PHORONIX, 2015). Foi verificado que ambos os testes atingem 100% de uso do componente de hardware avaliado. O resultado do *SpeedRam* é mostrado em MBps, enquanto o *eSpeak* em segundos.

A Tabela 2 apresenta os cenários considerados no planejamento de experimentos.

Tabela 2 – Experimentos

Nº	Técnica de Migração	Memória RAM	Carga de Trabalho	Qtd de VMs
1	<i>Non-live Migration</i>	256 MB	<i>SpeedRam</i>	2
2	<i>Non-live Migration</i>	256 MB	<i>SpeedRam</i>	4
3	<i>Non-live Migration</i>	256 MB	<i>eSpeak</i>	2
4	<i>Non-live Migration</i>	256 MB	<i>eSpeak</i>	4
5	<i>Non-live Migration</i>	512 MB	<i>SpeedRam</i>	2
6	<i>Non-live Migration</i>	512 MB	<i>SpeedRam</i>	4
7	<i>Non-live Migration</i>	512 MB	<i>eSpeak</i>	2
8	<i>Non-live Migration</i>	512 MB	<i>eSpeak</i>	4
9	<i>Live Migration</i>	256 MB	<i>SpeedRam</i>	2
10	<i>Live Migration</i>	256 MB	<i>SpeedRam</i>	4
11	<i>Live Migration</i>	256 MB	<i>eSpeak</i>	2
12	<i>Live Migration</i>	256 MB	<i>eSpeak</i>	4
13	<i>Live Migration</i>	512 MB	<i>SpeedRam</i>	2
14	<i>Live Migration</i>	512 MB	<i>SpeedRam</i>	4
15	<i>Live Migration</i>	512 MB	<i>eSpeak</i>	2
16	<i>Live Migration</i>	512 MB	<i>eSpeak</i>	4

Fonte: Dos autores (2015).

Foram adotadas três variáveis de resposta para os experimentos, (i) **tempo total de migração**, (ii) **tempo total de execução dos experimentos (*downtime*)** e (iii) **resultado dos benchmarks**. A escolha dessas variáveis de resposta possibilita observar a influência dos fatores em torno não só da migração, mas também ao desempenho do teste executado.

2.2 Ambiente de Experimentos

O ambiente no qual os experimentos foram realizados é composto por dois computadores estruturados como apresentado na Tabela 3 e conectados por uma rede local com interfaces de 100 Mbps.

As máquinas virtuais foram criadas conforme apresentado na Tabela 4, utilizando a versão 4.1 do *hypervisor Xen* para a arquitetura *amd64*. Optou-se por utilizar o *Xen*¹¹ devido ao fato do *Xen* da grande utilização dele no meio científico, como pode ser visto na Seção

¹¹ <<http://www.xenproject.org/>>

de trabalhos relacionados. Foi utilizado o sistema de arquivos *Network File System* de forma a compartilhar os arquivos da imagem das máquinas virtuais, assim apenas a memória RAM precisa ser migrada.

Tabela 3 – Configuração das máquinas físicas

Configuração das Máquinas Físicas	
Memória RAM	12 GiB (3x4 GiB) 1333 MHz DDR3
Processador	Intel Core i5-3470 3.20 GHz
Quantidade de Núcleos	4
Quantidade de Threads	4
Cache	6 MB
Disco Rígido	SATA 1 TiB 3.5 7.200 rpm
Sistema Operacional	Ubuntu 12.04 LTS x64

Fonte: Dos autores (2015).

Tabela 4 – Configuração das máquinas virtuais

Configuração das Máquinas Virtuais	
Tamanho do Disco	5 GB
Sistema Operacional	Ubuntu 12.04 LTS x64
Quantidade de Núcleos Virtuais	1
<i>Swap</i>	128 MB

Fonte: Dos autores (2015).

3 DISCUSSÃO E ANÁLISE DOS RESULTADOS

Após o planejamento dos experimentos e sua execução, foi verificado que os valores das 10 replicações iniciais tiveram pouca variação. Dessa forma, foi considerado desnecessário o incremento na quantidade de replicações. Foram utilizados, para cada variável de resposta, os intervalos de confiança de 95% obtidos através da distribuição *T-Student*. A partir dos valores obtidos para cada variável de resposta, foram calculadas as suas variações totais, além da influência de cada um dos fatores em seus valores, como descrito em [Jain \(1991\)](#).

Como pode ser visualizado na Figura 2, a influência dos fatores carga de trabalho (26,75%), técnicas de migração (26,76%) e os dois juntos (técnicas de migração + carga de trabalho - 26,66%) são os valores com maior porcentagem. O que mostra que a técnica de migração e a carga de trabalho são os fatores que mais influenciam no tempo total de migração. A memória RAM é outro fator que também influencia (15,35%) diretamente no tempo de mi-

gração, uma vez que quanto maior a memória, maior será a quantidade de dados a ser copiada através da rede até o *hypervisor* de destino.

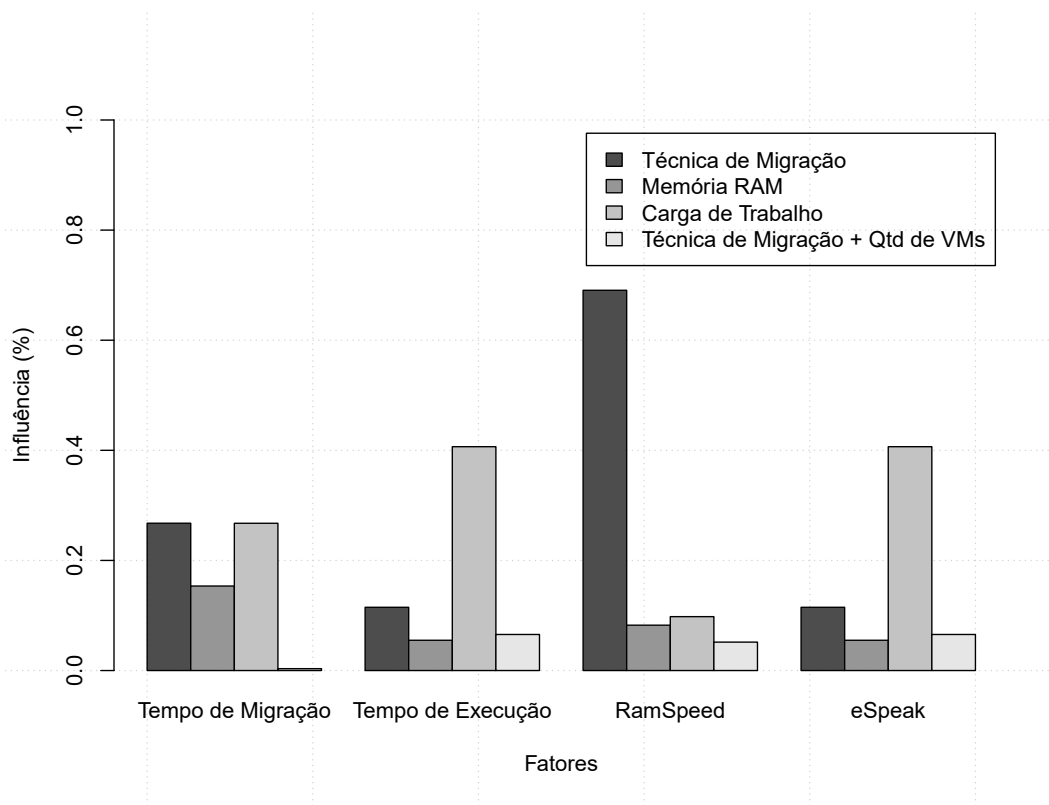


Figura 2 – Influência dos fatores
Fonte: Dos autores (2015).

Além disso, a Figura 2 apresenta a carga de trabalho como o fator que apresentou maior influência (55,63%), devido ao fato de que o teste *SpeedRam* leva um pouco mais de tempo para ser executado que o *eSpeak*. O segundo fator foi a técnica de migração (14,14%), em que a execução do teste leva um tempo maior utilizando a *non-live migration*, em alguns casos muito mais demorado, conforme a Figura 5. A técnica de migração aliada a carga de trabalho reflete o peso destes fatores (7,97%), que juntos também influenciam no tempo total. A quantidade de máquinas virtuais influencia (7,55%) o tempo total do teste, por sobrecarregar o *hardware*, levando uma maior concorrência de recursos. Os demais fatores obtiveram valores de influência mais baixos, sendo a carga de trabalho + a quantidade de máquinas virtuais igual 4,93%, a técnica de migração + a quantidade de VMs 3,11%.

De maneira mais específica, os experimentos realizados com a carga de trabalho *SpeedRam* apresentam um aumento significativo nos tempos de migração e de execução quando foi alterada a técnica de migração e a quantidade de memória RAM, sem uma variação significativa quando se aumenta a quantidade de máquinas virtuais. Isso ocorre porque a técnica

live-migration copia iterativamente as páginas utilizadas da memória, chamadas de páginas sujas, antes de finalizar o processo de migração. Pode ser observado que a técnica de migração (69,07%) é o fator que mais influencia o resultado. De forma geral, a técnica de migração *live migration* tem melhores resultados do que a *non-live migration*, mostrando o efeito da degradação do desempenho em função do *downtime*.

Também é possível verificar a interação dos fatores no teste *eSpeak*, na Figura 3, em que a quantidade de máquinas virtuais (40,66%) influencia diretamente no resultado do teste, afinal quanto maior for a quantidade de máquinas virtuais, maior será a concorrência de recursos e a degradação do desempenho. O conjunto TM + RAM + VMs (técnica de migração + memória RAM + quantidade de máquinas virtuais - 33,38 %) também tem influência nas variáveis de resposta.

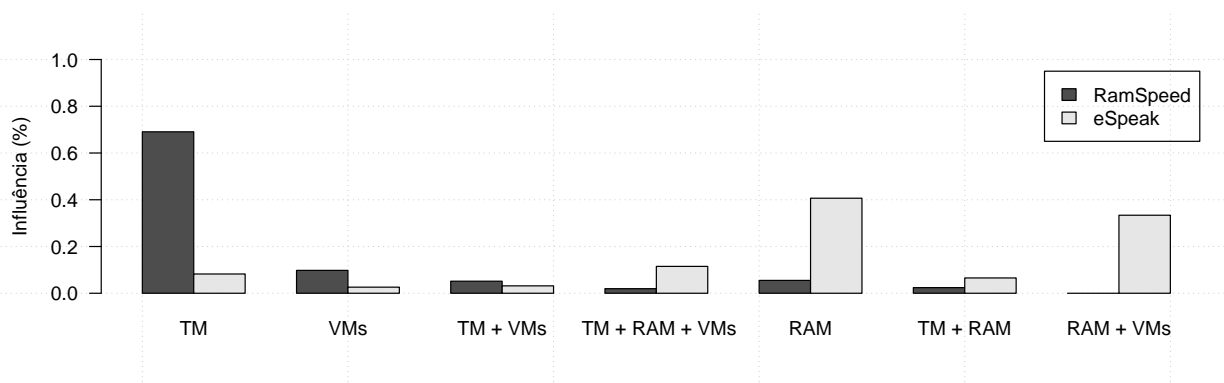


Figura 3 – Influência *SpeedRam* e *eSpeak*
 Fonte: Dos autores (2015).

A Figura 4 apresenta a relação das médias dos tempo totais de migração das máquinas virtuais. Nota-se que os valores obtidos para os experimentos executados com a carga de trabalho gerada pelo teste *eSpeak* apresentam uma variação ao aumentar (dobrar) a quantidade de memória RAM solicitada para utilização nos experimentos pelas máquinas virtuais. Embora para as técnicas de migração os tempos foram praticamente iguais, na outra análise do teste *SpeedRam*, a técnica de migração *Live* obteve tempos superiores ao da técnica *Non-Live*, decorrente do aumento das cópias das páginas da memória RAM dinamicamente.

Na Figura 5 podem ser observadas as médias dos tempos de execução dos experimentos. Nota-se que em todos os casos em que foi utilizada a técnica *live migration*, os tempos observados foram menores comparados aos tempos da técnica *non-live migration*.

Com relação à carga de trabalho, as duas cargas (*SpeedRam* e *eSpeak*) apresentaram comportamentos distintos. A Figura 5 mostra que o resultado do teste *SpeedRam* foi pior para a *live migration* em todos os casos, indicando que a técnica de migração influencia de forma

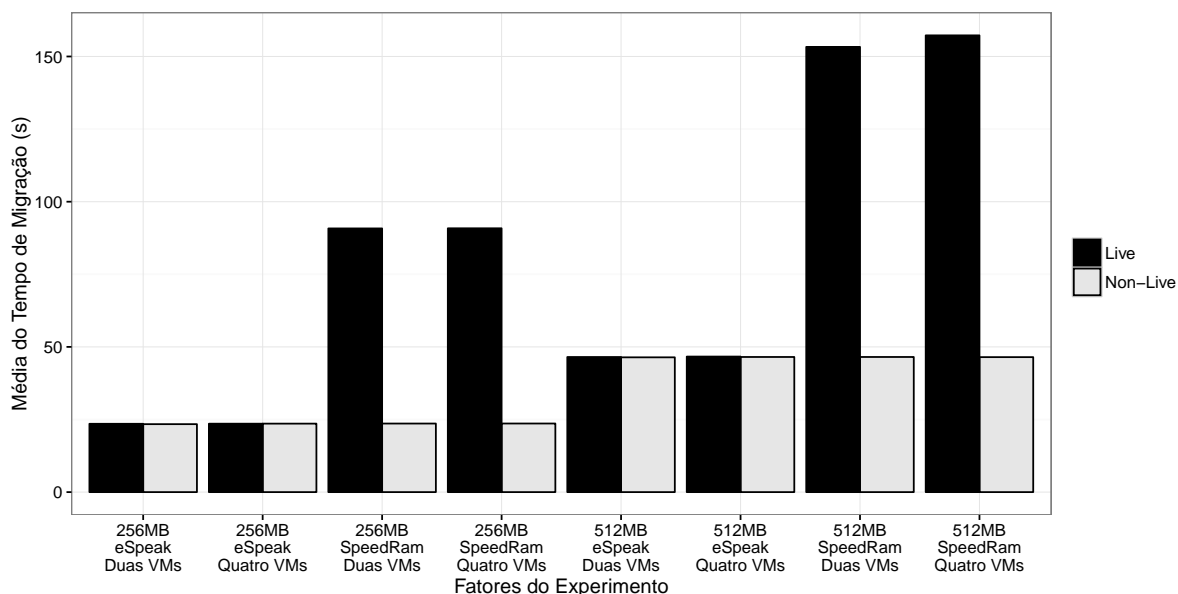


Figura 4 – Média em segundos do tempo total de migração
Fonte: Dos autores (2015).

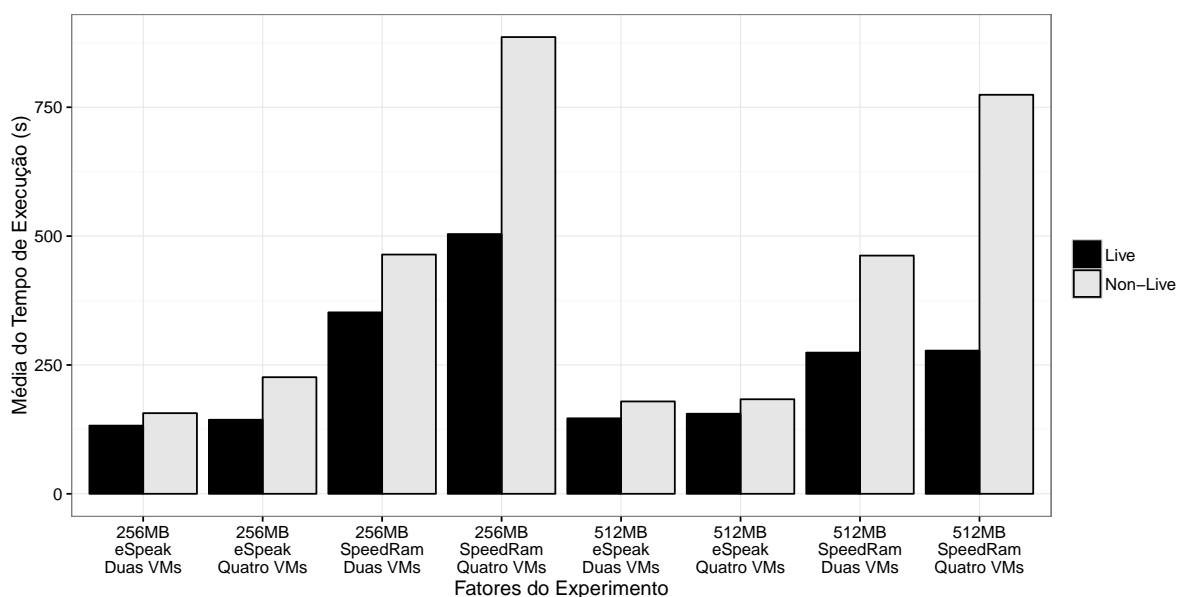


Figura 5 – Média em segundos do tempo total de execução
Fonte: Dos autores (2015).

significativamente em aplicações com uso intenso da memória RAM. A *live migration* obteve os melhores resultados em todos os casos. Quando a técnica *non-live migration* é utilizada em situações de aumento da quantidade de máquinas virtuais, os resultados obtidos são inferiores. Além disso, a Figura 5 mostra que em todos os casos os resultados do *eSpeak* foram seme-

lhantes, todos entre 41 e 45 segundos. O foco do teste é gerar uma carga de trabalho para o processador, enquanto que são feitos poucos acessos à memória em relação ao *SpeedRam*.

4 TRABALHOS RELACIONADOS

Koslovski, Bouffleur e Charão (2006) fizeram uma avaliação de desempenho da migração de máquinas virtuais utilizando o *hypervisor Xen*. Na avaliação realizada, os autores consideraram como métricas: tempo de indisponibilidade de um serviço, tempo de resposta de um servidor virtualizado e tempo total de migração de VMs. Os experimentos possuíam um serviço de um servidor Web, utilizando o método de migração *live migration*. Os resultados mostram que na presença de muitas requisições à memória da máquina virtual, a migração tornou-se mais demorada, sendo necessário copiar todas as modificações para o *hypervisor* de destino. Na análise da latência em uma máquina virtual, utilizando um servidor de jogos Quake2, não foi verificada a perda de nenhum pacote, e do ponto de vista dos jogadores, nenhuma alteração foi percebida na usabilidade. Desta forma, os autores afirmam que a migração é uma boa ferramenta para administradores de sistemas, onde a sobrecarga imposta para efetuar a migração e manter os serviços ativos é baixa.

O trabalho de Clark et al. (2005) foi o primeiro a considerar a *live migration* utilizando o *hypervisor Xen*. Nele foi realizada a análise do tempo de resposta dos pacotes enviados pela rede local e o *downtime*. Foi observado que o tempo de resposta dos pacotes sofria uma alteração durante a migração, aumentando no início e no fim da migração, contudo sem aparente perda de desempenho. Em relação ao *downtime*, utilizando a ferramenta de *benchmark SPECweb99* para produzir a carga de trabalho, é afirmado que a migração pode ser feita com *downtime* de 20 milissegundos. Já com o servidor de jogo Quake3, a migração pode ser realizada com 60 milissegundos de *downtime*.

Magalhães, Soares e Gomes (2011) buscaram avaliar o impacto de duas abordagens de migração de máquinas virtuais no *hypervisor Xen*. Os autores avaliaram a abordagem *non-live migration* e a *live migration*, tendo como métricas principais o tempo total de migração, o *downtime* e o tempo de resposta. Com os resultados, os autores afirmam que o *downtime* da *non-live migration* é pelo menos 5 vezes maior que o da *live migration*. Além disso, o tempo total de migração da *live migration* foi superior ao da *non-live migration*. Com a *live migration* houve um aumento no tempo de resposta das aplicações, caracterizando quebra de acordo de nível de serviço (SLA - *Service Level Agreement*) para *benchmarks* de aplicações Web, enquanto a *non-live migration* causou indisponibilidade dos serviços.

Neves et al. (2007) avaliaram o impacto da virtualização e da migração de máquinas virtuais no *hypervisor Xen* executando aplicações científicas escritas com a interface MPI (*Mes-*

sage Passing Interface). Comparando o ambiente virtualizado com o sistema nativo, o primeiro teve resultados bem próximos em relação à vazão, ao tempo de execução da aplicação e ao desempenho. Nos experimentos com migração, foi utilizada a *live migration*, possibilitando notar a correlação entre a quantidade de acessos à memória e o tempo de migração. Com aplicações de alto desempenho, o tempo de migração pode se estender por mais de 100 segundos. Já nos testes de comunicação entre processos executando no ambiente, os autores notaram uma degradação do desempenho ao se colocar mais de uma máquina virtual em um nó físico.

Voorsluys et al. (2009) ressaltam a utilização da migração como balanceador de carga. Foram realizados experimentos analisando os efeitos sobre o desempenho ao migrar máquinas virtuais, utilizando o *hypervisor Xen*, focando no SLA. Em um cenário de aplicações Web utilizando a *live migration*, 600 usuários recorrentes de *downtime* de mais de 3 segundos foram atendidos, violando os acordos de SLA. Entretanto, os autores afirmam que o SLA pode ser satisfeito diminuindo a carga do sistema (500 usuários).

Em Tao et al. (2016), os autores focaram em resolver o problema de migração dinâmica considerando simultaneamente três objetivos diferentes: energia, comunicação e custo. Assim sendo, os autores propuseram um modelo multiobjetivo triplo para o problema de migração dinâmica de VMs, dividido em duas partes: i) solucionável por meio de um método determinístico e ii) por meio de um algoritmo não-determinístico. No entanto, eles não consideraram diferentes técnicas de migração, mantendo o foco apenas na alocação de recursos.

Xiao, Song e Chen (2013) propuseram um sistema de alocação de VMs em um ambiente em nuvem dinâmico com migração que considera as demandas das aplicações e com suporte à computação verde. Para tal, foi desenvolvido um sistema de alocação que, segundo os autores, evita a sobrecarga e minimiza o número de máquinas físicas utilizadas. Tal sistema é baseado no conceito de “*skewness*”, que representa a utilização dos servidores de VMs. A técnica *non-live migration* não foi considerada.

Em Baruchi, Midorikawa e Sato (2015), os autores argumentam que a *live migration* degrada severamente o desempenho das VMs, e que saber o momento certo para fazer a migração pode resultar em menores perdas. Com o histórico e a identificação das cargas foi possível obter a Transformada de *Fourier* de tais informações, o que permitiu extrair o tamanho do ciclo das requisições, possibilitando que a migração aconteça nos melhores instantes. Apesar dos esforços, nenhuma comparação direta entre as técnicas de *live migration* e *non-live migration* foi realizada.

Kukral et al. (2015) desenvolveram um *framework* com alto grau de modularização que tem o objetivo de quantificar a disponibilidade de VMs durante a migração entre as estações, chamando esta métrica de disponibilidade do serviço. Os autores averiguaram, por meio de experimentações, que aproximadamente 87% das migrações são realizadas de forma satisfatória.

Assim como os outros trabalhos elencados, tal trabalho também não comparou a técnica *live migration* com a *non-live migration*.

5 CONCLUSÃO

Este trabalho utilizou o modelo estatístico de planejamento de experimentos para a avaliação de desempenho das técnicas de migração de máquinas virtuais *live migration* e *non-live migration*. Por meio de uma análise dos resultados obtidos para as variáveis de resposta adotadas, foi possível verificar o comportamento do sistema computacional em estudo.

A técnica *live migration* apresentou os melhores resultados comparados ao da técnica *non-live*. Em situações de alta utilização da memória RAM, a técnica *non-live migration* apresentou um tempo total de migração elevado. Os valores obtidos para os experimentos executados com a carga de trabalho gerada pelo teste *eSpeak* mostraram que o uso intenso do processador tem pouca influência sobre o tempo total de migração, ao passo que experimentos realizados com o teste *SpeedRam* mostraram que o uso da memória RAM pode aumentar drasticamente esse tempo. Isso ocorre por causa da sobrecarga na quantidade de páginas de memória que devem ser migradas dinamicamente. Observou-se também que quanto maior o número de máquinas virtuais executando sobre o mesmo *hardware* simultaneamente, maior será o tempo total de migração. O mesmo acontece quando se aumenta a memória RAM disponível para cada máquina virtual. Tudo isto confirma a memória RAM como fator determinante sobre o tempo de migração.

Em geral, as principais contribuições deste artigo são: i) identificação de um modelo estatístico que pode ser aplicado para avaliação das principais técnicas de migração de máquinas virtuais utilizadas principalmente no ambiente de Computação em Nuvem; ii) correlação entre as técnicas de migração de máquinas virtuais e os fatores envolvidos no ambiente virtualizado; iii) identificação de quais fatores influenciam as variáveis de resposta e como essa informação pode colaborar para a tomada de decisão eficiente.

Para os trabalhos futuros será proposta a utilização do modelo estatístico para a avaliação de desempenho da estratégia de *live migration* utilizando as técnicas: i) *precopy*, ii) *postcopy* e iii) *hybrid-live* no *hypervisor Xen*. Espera-se que o modelo estatístico possa levantar mais informações sobre quais abordagens permitem uma redução significativa no tempo total de migração e o número total de páginas transferidas pela rede, reduzindo a sobrecarga da mesma. Outra expectativa sobre os trabalhos futuros será identificar o que ocorre com o desempenho do ambiente quando a máquina virtual é suspensa por causa da falta de páginas de memória.

Agradecimentos: Os autores agradecem o apoio financeiro da FAPESB, CAPES e CNPq para a conclusão deste trabalho.

PERFORMANCE EVALUATION OF *VIRTUAL MACHINE MIGRATION APPROACHES* ON CLOUD COMPUTING ENVIRONMENT

ABSTRACT

Virtualization tools play a key role in increasing the use of Cloud Computing. Through virtualization, it is possible to migrate virtual machines within a cloud service provider, providing efficient resource utilization. However, there are not evident which techniques work better in according to the different load scenarios in which the system is operating. This work proposes the use of a consistent statistical model to evaluate the performance of migration techniques: (i) live migration and (ii) non-live migration. The objective of the statistical model is to identify the behavior of virtual machine migration techniques under different workload situations. The proposed statistical model for use is based on the performance evaluation approach of the virtual machine migration techniques combined with the 2^k factorial design.

Keywords: Migration. Performance Evaluation. Virtual Machine.

REFERÊNCIAS

BARUCHI, A.; MIDORIKAWA, E. T.; SATO, L. M. Reducing virtual machine live migration overhead via workload analysis. **IEEE Latin America Transactions**, IEEE, v. 13, n. 4, p. 1178–1186, 2015.

BELOGLAZOV, A.; BUYYA, R. Energy efficient allocation of virtual machines in cloud data centers. In: INTERNATIONAL CONFERENCE ON CLUSTER, CLOUD AND GRID COMPUTING, 10., 2010, Washington, DC. **Proceedings...** Piscataway: IEEE Computer Society, 2010. p. 577–578. Disponível em: <<http://dx.doi.org/10.1109/CCGRID.2010.45>>. Acesso em: 10 out. 2015.

CLARK, C. et al. Live migration of virtual machines. In: CONFERENCE ON SYMPOSIUM ON NETWORKED SYSTEMS DESIGN & IMPLEMENTATION: volume 2, 2., 2005, Berkeley, CA, USA. **Proceedings...** Berkeley, CA, USA: USENIX Association, 2005. p. 273–286. Disponível em: <<http://dl.acm.org/citation.cfm?id=1251203.1251223>>. Acesso em: 11 out. 2015.

FREEDMAN, D. A. **Statistical models:** theory and practice. Cambridge, GB: Cambridge university press, 2009. 442 p.

GALLOWAY, M.; LOEWEN, G.; VRBSKY, S. Performance metrics of virtual machine live migration. In: IEEE INTERNATIONAL CONFERENCE CLOUD COMPUTING , 8., 2015. **Proceeding...** New York: Piscataway: IEEE, 2015. p. 637–644.

HOLLANDER, R. M.; BOLOTOFF, P. V. **RAMspeed**. 2015. Disponível em: <<http://alasir.com/software/ramspeed>>. Acesso em: 10 out. 2015.

HU, Y. et al. Performance analysis of encryption in securing the live migration of virtual machines. In: IEEE INTERNATIONAL CONFERENCE CLOUD COMPUTING, 8., 2015, New York. **Proceeding...** New York, Piscataway: IEEE, 2015. p. 613–620.

JAIN, R. **The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling**. [S.l.]: John Wiley & Sons Chichester, 1991. (Wiley Professional Computing).

KOSLOVSKI, G. P.; BOUFLEUR, M. P.; CHARÃO, A. S. Avaliação do desempenho da migração de máquinas virtuais em xen. In: SIMPÓSIO DE INFORMÁTICA DA REGIÃO CENTRO DO RIO GRANDE DO SUL, 5., 2006, Santa Maria, RS. **Anais...** Santa Maria, RS: Centro Universitário Franciscano, 2006.

KUKRAL, T. et al. Vm migration measurement and failure detection. In: IEEE INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS AND SIGNAL PROCESSING, 38., 2015, Prague. **Proceeding...** Prague, Piscataway: IEEE, 2015. p. 285–288.

MAGALHÃES, D. V.; SOARES, J. M.; GOMES, D. G. Análise do impacto de migração de máquinas virtuais em ambiente computacional virtualizado. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 29., 2011, Campo Grande, MS. **Anais...** Campo Grande, MS: Universidade Federal de Mato Grosso do Sul, 2011. p. 235–248.

NEVES, M. V. et al. Impacto da migração de máquinas virtuais de xen na execução de programas mpi. In: WORKSHOP EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO, 8., 2007, Gramado, RS. **Anais...** Gramado, RS: Sociedade Brasileira de Computação, 2007.

PCHEK, M. L. **Máquinas Virtuais e Emuladores: conceitos, técnicas e aplicações**. [S.l.]: Novatec, 2006.

PHORONIX. **Open-Source Benchmarking**. 2015. Disponível em: <<http://www.phoronix-test-suite.com>>. Acesso em: 02 julho 2015.

ROSSI, F. D. **Alocação Dinâmica de Recursos no Xen**. 2008. 70 f. Dissertação (Mestrado em Ciência da Computação) — Universidade Católica do Rio Grande do Sul, Porto Alegre, BR, 2008.

SANTOS, R.; CHARÃO, A. S. **Análise Comparativa de Desempenho do Hipervisor Xen: paravirtualização versus virtualização total**. UFSM, Santa Maria, RS, 2008.

SHAH, S. A. R.; JAIKAR, A. H.; NOH, S.-Y. A performance analysis of precopy, postcopy and hybrid live vm migration algorithms in scientific cloud computing environment. In: IEEE INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING & SIMULATION, 13., 2015, Amsterdam. **Proceeding...** Amsterdam, Piscataway: IEEE, 2015. p. 229–236.

STRUNK, A.; DARGIE, W. Does live migration of virtual machines cost energy? In: INTERNACIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 27., 2013, Barcelona. **Proceeding...** Barcelona, Piscataway: IEEE, 2013. p. 514–521.

TAO, F. et al. BGM-BLA: A new algorithm for dynamic migration of virtual machines in cloud computing. **IEEE Transactions on Services Computing**, v. 9, n. 6, p. 910–925, Nov. - Dez. 2016. Disponível em: <https://www.researchgate.net/publication/277564168_BGM-BLA_A_new_algorithm_for_dynamic_migration_of_virtual_machines_in_cloud_computing>. Acesso em: 10 out. 2015.

VOORSLUYS, W. et al. Cost of virtual machine live migration in clouds: a performance evaluation. In: JAATUN, M.; ZHAO, G.; RONG, C. (Ed.). **Cloud Computing**. Heidelberg: Springer, 2009, (Lecture Notes in Computer Science, v. 5931). p. 254–265. Disponível em: <http://dx.doi.org/10.1007/978-3-642-10665-1_23>. Acesso em: 09 out. 2015.

XIAO, Z.; SONG, W.; CHEN, Q. Dynamic resource allocation using virtual machines for cloud computing environment. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 24, n. 6, p. 1107–1117, 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6311403/>>. Acesso em: 10 out. 2015.

ZAGHLOUL, S. The mutual effect of virtualization and parallelism in a cloud environment. In: AFRICON, 11., 2013, Maurice. **Proceeding...** Maurice, Piscataway: IEEE, 2013. p. 1–5.

Recebido em: 18/05/2016

Aprovado em: 29/11/2016

Publicado em: 20/01/2017