

Análise do Impacto de Migração de Máquinas Virtuais em Ambiente Computacional Virtualizado*

Deborah V. Magalhães¹, José Marques Soares², Danielo G. Gomes^{1,2}

¹Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)[†]

²Departamento de Engenharia de Teleinformática (DETI)

Universidade Federal do Ceará (UFC)

Fortaleza – CE – Brasil

deborah@great.ufc.br, {marques,danielo}@ufc.br

Resumo. Migração de máquinas virtuais tem sido uma alternativa comumente adotada em clusters e datacenters devido principalmente às suas vantagens de balanceamento de carga, tolerância a falha e economia de energia. Entretanto, o processo de migração provoca degradações no desempenho, refletindo negativamente nos serviços providos pelas máquinas virtuais. Nesse trabalho, apresentamos uma avaliação do impacto de duas abordagens de migração de máquinas virtuais amplamente referenciadas na literatura e disponibilizadas pelo hipervisor Xen: stop-and-copy e pré-cópia. Essas duas abordagens foram analisadas em um ambiente de experimentação real, com 4 servidores físicos heterogêneos, a partir de 4 métricas de performance: tempo total de migração, downtime, tempo de resposta e vazão de demanda. Os resultados experimentais obtidos mostram que (i) o downtime da stop-and-copy é pelo menos 5 vezes maior que o da pré-cópia, (ii) o tempo total de migração da pré-cópia foi superior ao da stop-and-copy para todas as cargas de trabalho, (iii) a pré-cópia provocou um incremento no tempo de resposta das aplicações, caracterizando quebra de SLA para benchmarks de aplicações web, enquanto a stop-and-copy causou indisponibilidade dos serviços.

Abstract. Virtual machines migration has been widely applied in datacenter environments due to various benefits including load balancing, fault tolerance and energy saving. However, the migration process causes degradation on the services provided by the virtual machines. In this paper, we present an impact evaluation of two virtual machines migration approaches that have been widely cited in the literature: stop-and-copy and pre-copy. We present an indoor experimental testbed for evaluating these two approaches through four performance metrics: total migration time, downtime, response time and throughput. Experimental results show that (i) the stop-and-copy downtime is at least 5 times higher than the pre-copy one, (ii) the total migration time of pre-copy was higher than stop-and-copy for all workloads, (iii) the pre-copy strategy increases the applications response time, while the use of stop-and-copy led to services downtime.

* Apoio da Capes e do INCT-MACC (processo CNPq 573710/2008-2).

[†] Laboratório associado ao INCT-MACC.

1. Introdução

A utilização de tecnologias para virtualização em ambientes de *datacenters* tem aumentado consideravelmente nos últimos anos, principalmente devido a sua capacidade de isolamento, consolidação e migração da carga de trabalho [Voorsluys *et al* 2009]. Uma das principais vantagens da virtualização é a possibilidade de remapeamento de recursos físicos em servidores virtuais a fim de lidar com carga de trabalho dinâmica. Isso é especialmente útil em Computação em Nuvem, uma vez que tais ambientes necessitam de uma infraestrutura computacional dinâmica, capaz de suportar a natureza elástica da alocação de recursos conforme a demanda dos usuários.

A migração de máquinas virtuais (VMs) através de servidores físicos distintos traz diversas vantagens, tais como: (i) balanceamento de carga - as VMs podem ser rearranjadas em nós físicos do *datacenter* para aliviar a carga sobre *hospedeiros* saturados, (ii) manutenção *online* e tolerância a falhas - uma máquina física pode precisar de *upgrade* ou manutenção do sistema devido a falhas, de forma que um administrador pode migrar as VMs em execução para uma máquina alternativa, liberando a máquina original para manutenção e, (iii) gerenciamento de energia - quando as VMs podem ser consolidadas em um número reduzido de servidores, reduzindo assim o custo de energia. Tais vantagens permitem aumentar a flexibilidade e reduzir a complexidade de gerenciamento dos recursos físicos em ambientes virtualizados. Entretanto, a migração provoca uma degradação no desempenho do sistema, uma vez que para realizá-la, é necessário parar a execução da VM, provocando sua indisponibilidade e, por conseguinte, das aplicações que nela executam. Além disso, há um consumo extra de ciclos de CPU e de largura de banda no processo de transferência da VM de um hospedeiro para outro, impactando nos recursos disponibilizados para os serviços providos pela mesma. Nesse contexto, uma melhor compreensão do impacto da migração no desempenho do sistema é fundamental para concepção de estratégias aptas a utilizar os recursos de maneira eficiente sem comprometer a capacidade de resposta e disponibilidade dos serviços ofertados.

Diante deste contexto, apresentamos uma avaliação do impacto de duas abordagens de migração de máquinas virtuais amplamente referenciadas na literatura e disponibilizadas pelo hipervisor Xen [Barham *et al* 2003]: *stop-and-copy* e pré-cópia. O objetivo da nossa avaliação é compreender o impacto gerado por essas duas abordagens no desempenho de um *testbed* real dedicado, destacando as vantagens e desvantagens de cada uma delas, de maneira a contribuir com os administradores de ambientes virtualizados na escolha da técnica mais adequada às suas metas de desempenho.

Dentre os objetivos específicos do trabalho, destacam-se: (i) para cada uma das técnicas avaliadas, medir e analisar o impacto da migração no desempenho do sistema a partir de quatro métricas relevantes no contexto de migração de máquinas virtuais: tempo total de migração, *downtime*, tempo de resposta e vazão de demanda; e (ii) apontar vantagens e desvantagens de cada uma das técnicas através de seis experimentos realizados no ambiente de testes.

2. Trabalhos Relacionados

As abordagens de migração de máquinas virtuais dividem-se em: *non-live migration* ou *pure stop-and-copy* e *live migration*. A *non-live migration* [Schmidt 2000] [Osman *et al* 2002] é caracterizada pela suspensão da VM no hospedeiro de origem durante o processo de transferência das páginas de memória para o hospedeiro de destino. Essa abordagem possui vantagens em termos de simplicidade, e reduz o tempo total de

migração em relação à abordagem *live*. Em contrapartida, o *downtime* sofrido pelas aplicações que são executadas na VM é maior nesse tipo de abordagem.

O primeiro trabalho a desenvolver uma estratégia *live migration* utilizando tecnologia de virtualização Xen foi proposto por Clark *et al* [Clark *et al* 2005]. Nesse trabalho, a solução conta com uma etapa de pré-cópia utilizada para reconstrução da imagem da memória da VM no *hospedeiro* de destino, realizada através da transferência das páginas de memória em várias iterações. A vantagem dessa proposta reside no fato da VM e, portanto, todas as aplicações nela hospedadas, continuarem a executar enquanto a imagem da memória é transferida, proporcionando um menor *downtime* dos serviços. No trabalho de Clark *et al* [Clark *et al* 2005], os autores pressupunham que as páginas de memória atualizadas eram transferidas muito mais rápido que a taxa de atualização - caso contrário, todo o trabalho seria ineficaz e deveria ser interrompido.

Com o intuito de reduzir o *overhead* gerado pela transferência das páginas de memória durante o processo de pré-cópia, foi proposta uma abordagem que utiliza tecnologias de *trace/replay* [Xu *et al* 2007] e *checkpointing/recovery* [Cully *et al* 2008] para recuperar o estado da VM no hospedeiro de destino de forma eficiente e, assim, minimizar o *downtime* das aplicações que nela executam e o tráfego da rede [Liu *et al* 2009]. Nessa solução, a quantidade de dados transferidos durante a etapa de pré-cópia é reduzida significativamente. Todavia, a ferramenta utilizada para recuperação do estado da VM no hospedeiro de destino, chamada ReVirt [Dunlap 2002], executa as instruções necessárias para tal, desde que o processador no hospedeiro de origem e destino sejam iguais, ou seja, a solução está limitada a cenários onde o *hardware* é homogêneo.

Em contraste às abordagens de pré-cópia, foi proposta a estratégia *live migration* baseada na pós-cópia [Hines 2009] com o propósito de reduzir o tempo de realocação das VMs durante o processo de migração. Essa abordagem é caracterizada pela execução da VM no *hospedeiro* de destino antes da transferência de suas páginas de memória para o mesmo. Apesar da pós-cópia reduzir o tempo de realocação da VM, provoca uma degradação no desempenho das aplicações, visto que ela é interrompida toda vez que uma página de memória não está disponível e somente é reinicializada quando a página de memória é recuperada a partir do hospedeiro de origem. No sentido de atenuar o *downtime* sofrido pela VM depois de sua realocação no hospedeiro de destino, os autores implementaram a técnica de pré-paginação para trabalhar juntamente com mecanismo de paginação sob demanda. Esta técnica é utilizada para esconder a latência das falhas de página ocorridas quando a VM acessa uma página de memória que ainda não está disponível. Recentemente, a combinação de paginação sob demanda com a pré-paginação também foi utilizada em [Hirofuchi *et al* 2010] para compor a estratégia de migração de máquinas virtuais baseada na abordagem pós-cópia visando reduzir o *downtime* experimentado pela VM.

3. Metodologia de Avaliação e Cenário de Experimentação

3.1. Métricas

Através da análise de trabalhos recentes [Voorsluys *et al* 2009] [Clark *et al* 2005] [Liu *et al* 2009] [Hines 2009] [Oliveira 2010], identificamos as métricas mais utilizadas e relevantes no contexto de migração de máquinas virtuais. Neste trabalho, as seguintes métricas foram escolhidas: (1) *Downtime*, refere-se ao tempo durante o qual a execução da VM em migração é interrompida. No mínimo, esse intervalo de tempo inclui a transferência do estado do processador e dispositivos de rede e disco. Na *stop-and-copy* esta transferência também inclui todas as páginas que compõem a memória. Na pré-

cópia, essa etapa inclui apenas as páginas de memória modificadas remanescentes. O *downtime* é uma métrica fundamental para avaliação de desempenho da migração, visto que a inatividade da VM impacta diretamente nos serviços por ela disponibilizados, podendo comprometer a transparência de migração em nível de usuário; (2) Tempo total de migração refere-se ao tempo gasto na realização de todas as etapas necessárias ao processo de migração. Essa métrica é de suma importância porque afeta a liberação de recursos em ambos os *hospedeiros* participantes. A memória ocupada no hospedeiro de origem pela VM fica alocada à migração e só será liberada após a conclusão do processo; (3) Tempo de resposta está relacionado com os atrasos sofridos pelas aplicações em execução dentro da VM causados pelo processo de migração. Essa métrica refere-se à responsividade das aplicações e visa medir o quanto a migração compromete a capacidade das aplicações atenderem em tempo hábil às requisições dos usuários; (4) Vazão de demanda corresponde à taxa de requisições por segundo submetidas pelos usuários durante o processo de execução da aplicação.

3.2. Cargas de trabalho

A fim de conduzir a avaliação do impacto das abordagens de migração no desempenho do sistema, serão utilizados quatro diferentes *benchmarks*:

1. O Netperf [Netperf 2010] é utilizado para medir aspectos de desempenho de rede. Ele gera um tráfego de rede e avalia o desempenho da requisição/resposta utilizando os protocolos UDP e TCP;
2. O *Sample Web Application* [The SampleWeb Application 2010] é utilizado para testar uma aplicação web simples formada pela combinação de três páginas: uma *homepage* (home.html), uma página produtos (products.html) e uma página contatos (contacts.html);
3. O *Olio* [Olio 2010] representa um *website* de eventos sociais que permite usuários executarem diversas ações como: carregar *homepage*, acessar o sistema, criar novos eventos, atender eventos e buscar eventos por data ou *tag*. Em nossos experimentos, utilizamos a implementação PHP do Olio, empregando a pilha LAMP (Linux, Apache, MySQL e PHP);
4. O Apache Benchmark (AB) [The Apache Software Foundation 2010] é um programa de linha de comando que faz parte do servidor web Apache, e é utilizado para medir o desempenho de servidores web através de requisições HTTP a URLs especificadas pelo usuário.

Utilizamos o Faban [Faban 2010] na qualidade de gerador de tráfego *open-source* para gerar a carga de trabalho, dispará-la contra o *Sample Web Application* e o *Olio* bem como monitorar seus resultados. O nível de carga disparado contra essas aplicações pode ser variado alterando o número de usuários simultâneos que as acessam. Além do *Sample Web Application*, o *Olio* foi utilizado por se tratar de uma aplicação Web 2.0 realista e representativa com padrões de comunicação cliente/servidor, padrões de escrita e carga do servidor particulares que a torna diferente de outras cargas estáticas clássicas. A combinação Olio/Faban foi originalmente proposta em [Sobel et al 2008], como parte do *benchmark* Cloudstone e, posteriormente, utilizada em [Voorsluys et al 2009].

3.3. Cenário de Experimentação

O cenário utilizado consiste em quatro servidores físicos com especificações de *hardware* e *software* heterogêneas (vide Tabela 1), definidas de acordo com o papel desempenhado por cada máquina no sistema. A arquitetura do ambiente de

experimentação é ilustrada na Figura 1, na qual podemos destacar 3 diferentes funções: (1) o *Front-end* é responsável por gerenciar o conjunto de recursos formados pelo *XCP-storage*, *XCP-resource01* e *XCP-resource02*, através da ferramenta *Open Xen Manager* [OpenXenManager 2010]; (2) os servidores *XCP-resource01* e *XCP-resource02* representam os hospedeiros onde serão instanciadas as VMs e suas respectivas aplicações. Para tanto, é utilizado o *XCP Server* que faz parte da infraestrutura de virtualização para Nuvens XCP (*Xen Cloud Platform*) [Xen 2010]. Essa infraestrutura é baseada no *hipervisor Xen* [Xen, 2010] e foi escolhida por permitir agregar recursos físicos heterogêneos em um mesmo conjunto de recursos. Todas as VMs instanciadas contam com a pilha LAMP (Linux Ubuntu 10.04, Apache, MySQL e PHP) para a execução dos *benchmarks Sample Wev Application*, *Olio* e *Apache benchmark*; (3) o *XCP-storage* representa o servidor NFS (*Network File System*) e também hospeda o gerador de carga responsável por gerar a carga que será disparada contra as aplicações que são executadas nas VMs. Os quatro nós compartilham esse dispositivo de armazenamento, onde residem as imagens das VMs e os discos virtuais. Para realizar a migração, o Xen exige que as imagens estejam hospedadas em um dispositivo de armazenamento compartilhado [Wood et al 2009]. Todas as máquinas estão conectadas através de uma rede LAN *Fast Ethernet*.

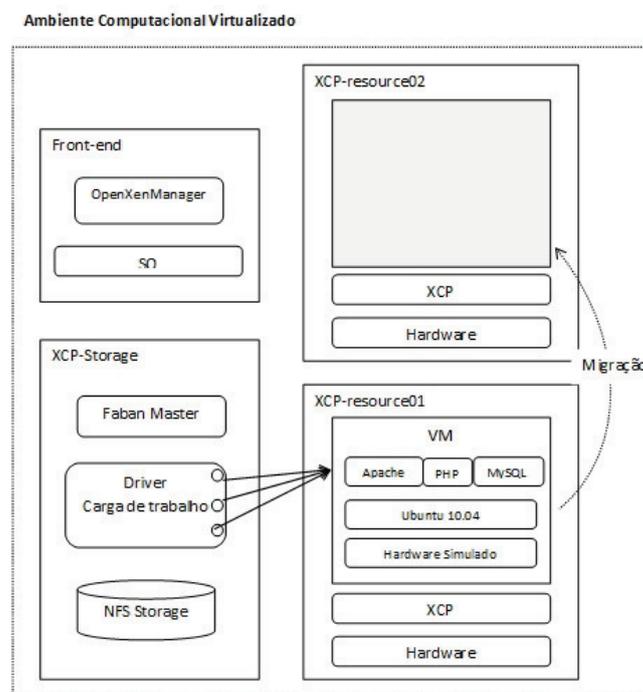


Figura 1. Arquitetura do ambiente de experimentação

Tabela 1. Hardwares e softwares utilizados para compor o ambiente de experimentação

Hardware	Software	Papel no ambiente
Intel® Core™2 Duo Processor . Memória DDR2 2GB Disco 120 GB	Windows Vista Home, OpenXenManager	Frontend - gerente do conjunto de recursos
Processador Intel® Core™ Duo Memória DDR2 2GB. Disco 80GB	Ubuntu Server 10.04	Storage NFS Gerador de carga (mestre)
Processador Intel® Core™2 Duo Processor. Memória DDR2 1.5GB.	XCP 0.5	Hospedeiro 1
Intel® Pentium® Processor Memória 2GB. Disco 500GB	XCP 0.5	Hospedeiro 2

4. Resultados

Nesta seção descrevemos seis experimentos realizados para avaliar as abordagens de migração pré-cópia e *stop-and-copy*. Na avaliação, as abordagens foram submetidas a cargas de trabalho com diferentes características e o impacto no desempenho do sistema associado a cada uma delas é quantificado a partir de quatro métricas: tempo total de migração, *downtime*, tempo de resposta e vazão de demanda. Em todos os experimentos, os nós do *tesbed* e as conexões de rede são dedicados. As migrações de todos os experimentos (exceto o teste de estresse) acontecem entre os nós físicos *XCP-resource01* e *XCP-resource02* (Figura 1).

4.1. Experimento #1

O primeiro experimento consistiu na migração de 3 VMs sem o uso de qualquer tipo de carga de trabalho. Cada VM, com um tamanho de memória RAM definido (256 MB, 512 MB e 1024 MB, respectivamente) foi submetida à pré-cópia e à *stop-and-copy*. O objetivo desse teste é avaliar a influência do tamanho da imagem de memória da máquina virtual sobre o *downtime* e o tempo total de migração.

Podemos observar na Figura 2(a) que a abordagem *stop-and-copy* obteve os valores de *downtime* igual a 21 s, 38 s e 71 s, respectivamente para as VMs de 256 MB, 512 MB e 1024 MB, enquanto a pré-cópia manteve seu *downtime* em 4 s para os três tamanhos de memória. Isso é justificado devido ao fato do *downtime* na *stop-and-copy* ser determinado pelo tempo necessário para transferir o contexto (informações referentes aos registradores de CPU, disco, interface de rede e páginas de memória) da VM para o *hospedeiro* de destino. Em contrapartida, na abordagem pré-cópia, o *downtime* é calculado a partir de um tamanho mínimo pré-estabelecido de páginas remanescentes a serem transferidas, ou seja, independe do tamanho da memória da VM. Portanto, o incremento no tamanho da memória da VM impacta negativamente na abordagem *stop-and-copy*. Além do *downtime* apresentado pela *stop-and-copy* ser pelo menos cinco vezes maior que o da pré-cópia, ele varia de acordo com o tamanho de memória das VMs, ou seja, para VM com memória muito grande o *downtime* pode atingir valores inaceitáveis, comprometendo a capacidade de resposta dos serviços disponibilizados pelas VMs.

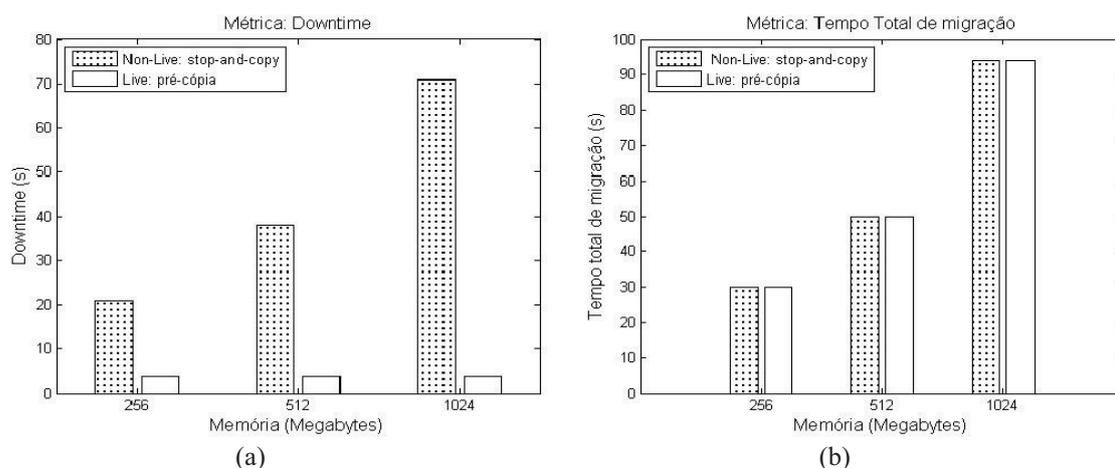


Figura 2: Experimento #1 - impacto da migração de VMs com diferentes tamanhos de memória utilizando as abordagens *stop-and-copy* e pré-cópia (a) sobre o *downtime*; (b) sobre o tempo total de migração

A Figura 2(b) ilustra o comportamento da métrica tempo total de migração versus tamanho da memória das VMs. Observamos que o incremento no tamanho da memória da máquina virtual impacta negativamente (e por igual) no tempo total de migração em ambas as abordagens. Isso é esperado, visto que é necessário copiar todas as páginas de memória em algumas das etapas do processo de migração nas duas abordagens. Como a taxa de transmissão da rede permanece constante, uma vez ocorrendo incremento do tamanho da memória na VM mais páginas serão transmitidas e maior será o tempo total de migração. As duas abordagens obtiveram os mesmos tempos totais de migração, a saber, 30 s, 51 s e 94 s, respectivamente para as VMs de 256 MB, 512 MB e 1024 MB. Essa igualdade dá-se pelo fato das VMs encontrarem-se ociosas neste primeiro experimento, portanto, sem operação de escrita na memória.

4.2. Experimento #2

O segundo experimento consistiu na migração de uma VM de 256 MB (RAM). Os resultados desse experimento correspondem a uma média de 10 execuções e apresentam um intervalo de confiança de 95%. Para cada grupo de 10 execuções, a VM foi submetida a uma carga de trabalho diferente (*benchmarks* Netperf, SampleWeb e Olio). Nesse experimento, o nível de carga foi fixado em 200 usuários simultâneos para o *benchmark* SampleWeb e 25 usuários simultâneos para o Olio. Na Figura 3(a), o *downtime* apresentado pela *stop-and-copy* é igual a 21 s para todas as cargas, valor este próximo ao esperado para transmitir uma VM de 256 MB de RAM a uma taxa efetiva 100 Mb/s. Na pré-cópia, o *downtime* permanece igual a 4 s para todas as cargas. A partir dos valores apresentados, constatamos que a variação da carga de trabalho não impacta no *downtime* em nenhuma das duas abordagens. Esse resultado deve-se ao fato da definição do *downtime* em ambas não depender do tipo de carga de trabalho. O resultado ilustrado na Figura 3(a) é positivo do ponto de vista de gerenciamento de recursos, pois o valor atribuído ao *downtime* é conhecido independentemente da carga de trabalho que esteja sendo executada.

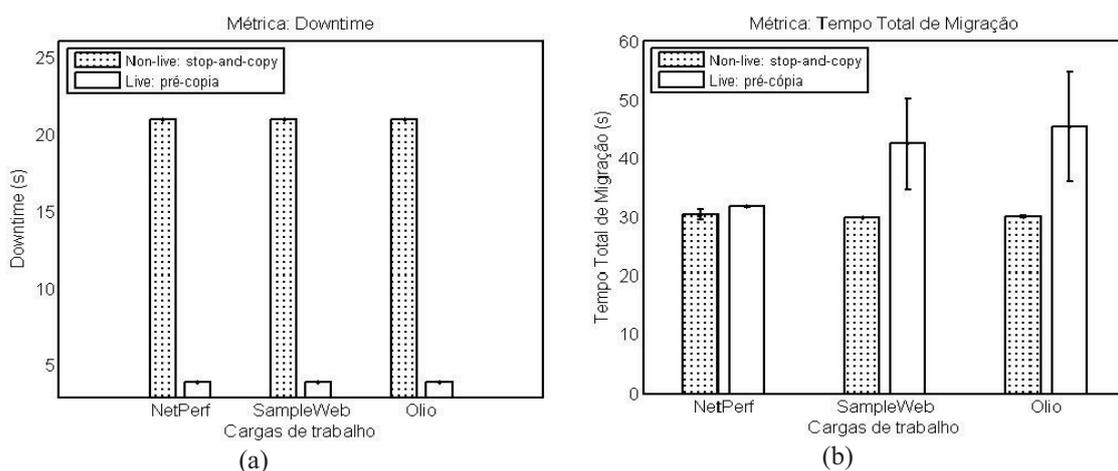


Figura 3: Experimento #2 - impacto da *stop-and-copy* e pré-cópia com VM de 256 MB (RAM) submetida a diferentes cargas de trabalho (a) sobre o *downtime* (b) sobre o tempo total de migração

A Figura 3(b) ilustra o comportamento da métrica tempo total de migração versus cargas de trabalho. A *stop-and-copy* obteve os tempos totais de migração 30,609 s, 30,100 s e 30,155 s, respectivamente, para os *benchmarks* NetPerf, SampleWeb e Olio, enquanto a pré-cópia obteve, para as mesmas cargas, os tempos de 31,943 s, 42,650 s e 45,540 s, respectivamente. Percebemos, portanto, que independentemente do

tipo de carga de trabalho utilizada, a *stop-and-copy* apresentou um tempo total de migração próximo a 30 s, justamente o tempo total de migração encontrado na Figura 2(b) para a mesma VM de 256 MB. Este resultado é esperado, visto que o tempo total de migração na *stop-and-copy* depende diretamente do tamanho da imagem de memória da VM a ser migrada e não do tipo de carga executada. Em contrapartida, a pré-cópia é vulnerável ao tipo de carga submetida à VM. O *benchmark* NetPerf gera tráfego na rede, sendo, portanto, I/O de rede intensivo, mas não causa impacto na memória. Logo, o tempo total de migração da abordagem pré-cópia para esse *benchmark* é muito próximo ao da *stop-and-copy*. No entanto, o SampleWeb gera carga que mistura operações orientadas à CPU e orientadas à memória, de modo que percebemos, pela Figura 3(b), que a diferença entre as duas abordagens aumenta para aproximadamente 12 segundos. Já no *benchmark* Olio, que realiza operações do tipo memória intensiva, a diferença aumenta para aproximadamente 15 segundos. Esses resultados são justificados pela fase de pré-cópia iterativa, durante a qual na primeira iteração todas as páginas são transferidas da origem para o destino. Nas iterações subsequentes, são copiadas apenas as páginas que foram modificadas durante a fase de transferência anterior, ou seja, para cargas de trabalho na quais as páginas de memória são atualizadas com muita frequência, a sobrecarga causada pela transferência de páginas de memória que serão posteriormente modificadas será grande. Nesse experimento, a abordagem pré-cópia teve um desempenho inferior a *stop-and-copy*, tendo em vista que quanto maior a fase de pré-cópia iterativa mais tempo os recursos de CPU e rede são utilizados para a transferência das páginas de memória. Estes recursos poderiam estar sendo disponibilizados às aplicações que executam na VM. Além disso, as páginas de memória ocupadas no hospedeiro de origem são liberadas apenas após a conclusão da migração, ocupando assim espaço de memória que poderia estar sendo utilizado por outras VMs.

4.3. Experimento #3 – teste de estresse

Antes de realizar o quarto experimento, executamos um teste de estresse com a combinação Faban/*Sample Web* em uma VM de 512 MB de memória sem efetuar qualquer migração. O objetivo desse teste é detectar o limite máximo de usuários simultâneos atendidos pela aplicação sem comprometer o SLA (*Service Level Agreement*). A carga de trabalho foi submetida à aplicação web e aumentamos gradualmente o número de usuários simultâneos entre as execuções, em incrementos de 100 usuários (vide Figura 4). O tempo total de execução do *benchmark* para cada grupo de usuários é configurado através do ajuste de 3 parâmetros distintos: *ramp-up*, *steady state* e *ramp-down*. Os valores ilustrados nas figuras levam em consideração apenas o *steady state* (regime permanente), configurado para durar 10 minutos. O tempo de resposta de 250 ms atende o SLA previamente definido no *Sample Web* para a ação do usuário de carregar sua *homepage*. Caso esse tempo seja superior a 250 ms, é caracterizada a quebra do SLA. Ao analisarmos os valores obtidos, descobrimos que 900 é o número máximo de usuários simultâneos que podem ser servidos pelo nosso ambiente de experimentação sem a quebra do SLA (Figura 4).

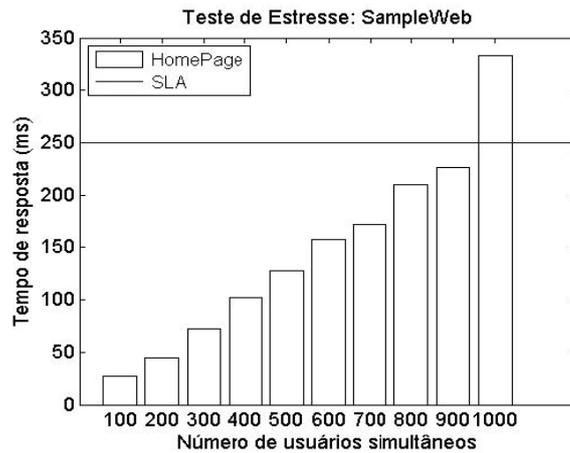


Figura 4: Experimento #3 - impacto da variação do número de usuários simultâneos no tempo de resposta do *Sample Web*

4.4. Experimento #4

O quarto experimento consistiu na migração, utilizando a abordagem pré-cópia, de uma única VM de 512 MB (RAM) executando o *workload* Faban/*Sample Web*. O objetivo desse teste é avaliar o impacto da abordagem pré-cópia no tempo de resposta e na vazão das requisições da aplicação para ação do usuário de carregar uma *homepage*. Durante a migração, a aplicação foi submetida a 300 usuários simultâneos. Apesar do teste de estresse (vide Figura 4) ter mostrado que nosso ambiente de experimentação comporta 900 usuários simultâneos sem provocar a quebra do SLA, a partir de 300 usuários, algumas requisições falham e esse número vai crescendo com o aumento do número de usuários simultâneos. Neste experimento, a migração ocorreu durante o *steady state*, cuja duração foi de 10 minutos. A Figura 5(a) ilustra que, durante o processo de migração, i.e. entre 250 s e 350 s, ocorre um incremento substancial no tempo de resposta da aplicação, provocando a quebra do SLA. Esse incremento representa um crescimento de 55,7% no tempo de resposta da aplicação.

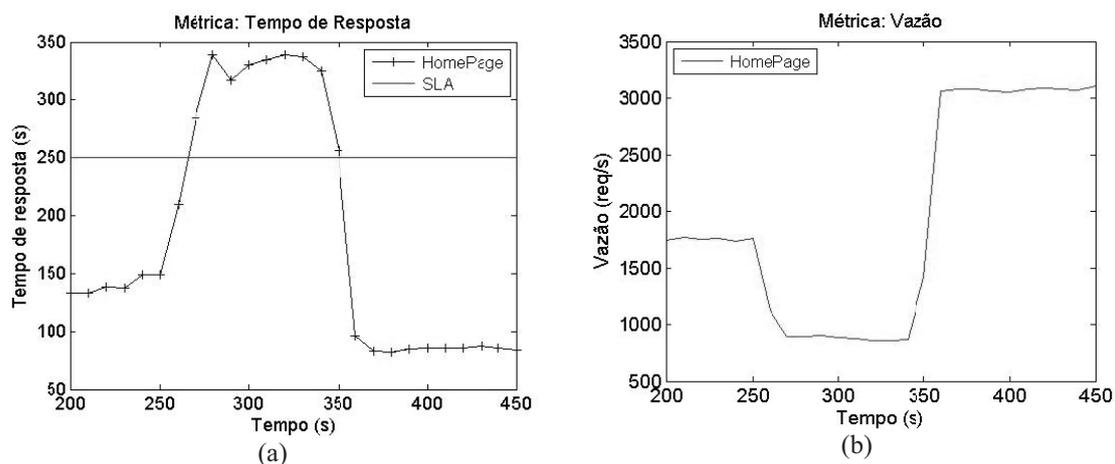


Figura 5: Experimento #4 - impacto da pré-cópia para a aplicação de carregamento de uma *homepage Sample Web* (a) no tempo de resposta (b) na vazão de demanda

Na Figura 5(b), observamos que durante o processo de migração ocorre uma redução no número de requisições por segundo. A redução na vazão de requisições foi de 23,8%. Depois do término da migração (instante 350 s), percebemos que existe um

aumento na vazão das requisições, chegando a alcançar a taxa de 3105 requisições por segundo. Essa taxa é maior que o valor encontrado antes da migração. Esse aumento é justificado pela heterogeneidade do ambiente de experimentação, visto que o *hospedeiro* de destino possui um *hardware* com desempenho melhor que o *hospedeiro* de origem.

4.5. Experimento #5

O quinto experimento consistiu na migração de uma única VM de 512 MB com Apache Benchmark (AB) para gerar as requisições disparadas contra a *homepage* da aplicação *Sample Web*. O experimento teve duração de 400 segundos. Seu objetivo consiste em avaliar o impacto da pré-cópia e da *stop-and-copy* no tempo de resposta e no número máximo de requisições que a máquina física consegue suportar. A Figura 6 ilustra os resultados coletados a partir desse experimento. As Figuras 6(a) e 6(b) apresentam os valores de tempo de resposta e vazão, respectivamente, coletados durante o experimento utilizando a pré-cópia. O processo de migração aconteceu entre 168 e 242 segundos. Na Figura 6(a), observamos que o tempo de resposta sofre um incremento durante o processo de migração. O pico mais alto ocorre imediatamente após a execução da VM no hospedeiro de destino, e sua duração corresponde ao tempo necessário para o sistema servir todas as requisições que haviam sido iniciadas durante o downtime experimentado pela aplicação. Na Figura 6(b), observamos que a vazão sofre um decremento durante o processo de migração, correspondendo a uma redução de aproximadamente 69% da taxa de requisições por segundo. Esses resultados mostram que a abordagem pré-cópia impacta negativamente no desempenho da aplicação *Sample Web*. Isso se deve à divisão dos recursos de rede e processamento entre a aplicação e o processo de migração, recursos esses anteriormente destinados apenas à aplicação.

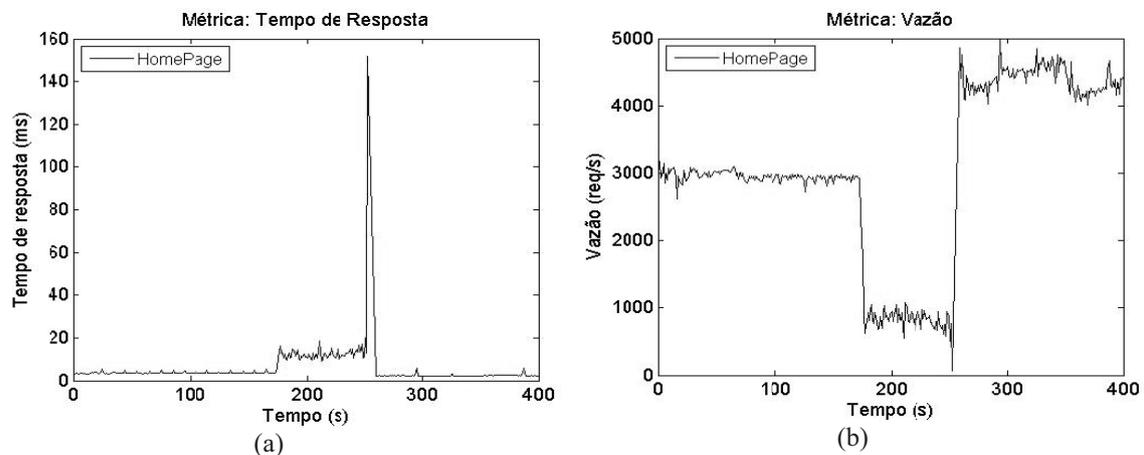


Figura 6: Experimento #5 - impacto da abordagem de migração pré-cópia para carregar uma *homepage* Sample Web (a) no tempo de resposta (b) na vazão de demanda

As Figuras 7(a) e 7(b) apresentam os valores de tempo de resposta e vazão, respectivamente, coletados durante o experimento utilizando a abordagem de migração *stop-and-copy*. O processo de migração aconteceu entre 185 s e 233 s. Na Figura 7(a), observamos que o tempo de resposta durante todo o processo de migração cai para zero. O mesmo comportamento pode ser observado com a vazão na Figura 7(b). Esses resultados se devem ao fato da execução da VM ser suspensa durante o processo de migração, causando indisponibilidade do serviço, o que torna esse tipo de migração

inapropriada para aplicações com rígidas restrições de tempo, como, por exemplo, aplicações de tempo real.

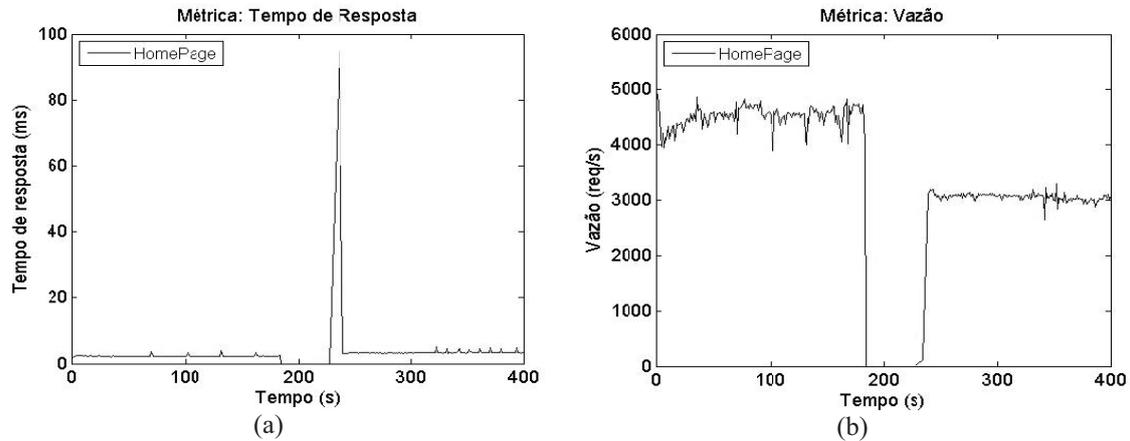


Figura 7: Experimento #5 - Impacto da abordagem de migração *stop-and-copy* para carregar uma *homepage* Sample Web (a) no tempo de resposta (b) na vazão de demanda

4.6. Experimento #6

O sexto experimento consistiu na migração, utilizando a pré-cópia, de uma única VM de 512 MB executando o *workload* Faban/Olio. Durante a migração, a aplicação foi submetida a 150 usuários simultâneos. Nesse experimento, a migração ocorreu durante o *steady state* que foi configurado para durar 10 minutos. O objetivo desse teste foi avaliar o impacto da pré-cópia no tempo de resposta e na vazão das requisições da aplicação para ação do usuário de carregar uma *homepage* do Olio. Caso a aplicação demore mais que a 1 s para reagir à ação do usuário, é caracterizada a quebra do SLA. A Figura 8(a) ilustra o processo de migração (intervalo 293 s a 429 s). Pouco antes do término da migração (~ 428 s), observamos um pico em seu tempo de resposta próximo provocando a quebra de SLA.

Na Figura 8(b) (*zoom* da Figura 8(a)), observamos no instante 425 s que o tempo de resposta da aplicação chega a zero e permanece durante 3 segundos. Em seguida, o pico mais alto nos tempos de resposta é observado. Esse período de 3 segundos corresponde ao *downtime* sofrido pela aplicação durante o processo de migração da pré-cópia, na qual a execução da VM é suspensa para copiar as páginas de memória remanescentes e o estado da CPU para o *hospedeiro* de destino. O pico mais alto ocorre imediatamente após a execução da VM no hospedeiro de destino e sua duração corresponde ao tempo necessário para o sistema servir todas as requisições que haviam sido iniciadas durante o *downtime* experimentado pela aplicação. Aqui vale ressaltar mais um resultado interessante. O tempo de inatividade experimentado pelo *benchmark* Olio servindo a 150 usuários simultâneos é bem acima do nível esperado em milésimos de segundo, previamente relatados na literatura para uma série de cargas conhecidas [Clark et al 2005] [Oliveira 2010]. Entretanto, a combinação Olio/Faban foi utilizada em [Voorsluys et al 2009] para avaliar o impacto da *live migration* no desempenho do sistema e o mesmo *downtime* e comportamento do tempo de resposta foi encontrado, reforçando os resultados alcançados nesse trabalho. Na Figura 8(c), notamos um comportamento semelhante da vazão em relação ao tempo de resposta. No instante 425 s o número de requisições por segundo chega a zero e, em seguida, um pico é observado alcançando um valor de 121 requisições/s.

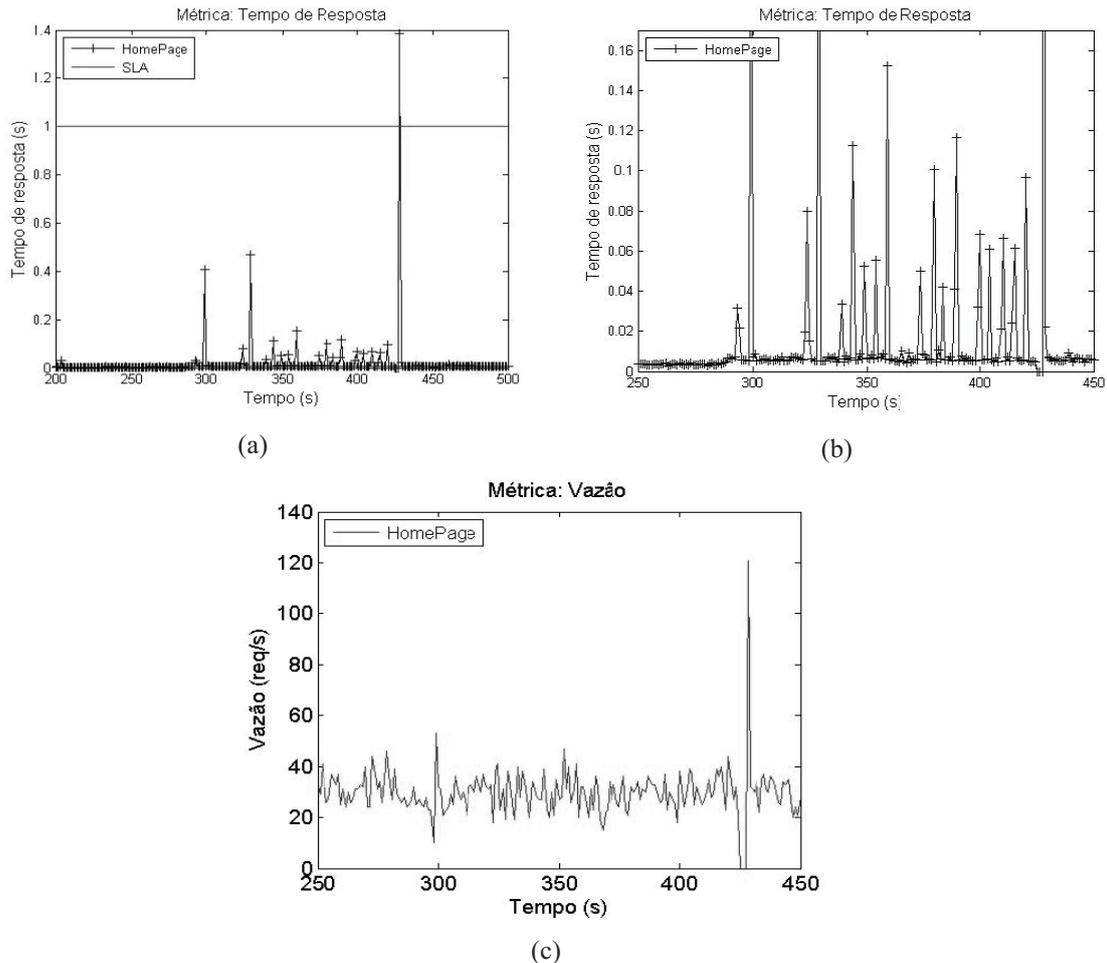


Figura 8: Experimento #6 - impacto da abordagem de migração pré-cópia em carregamento de *homepage* Olio (a) no tempo de resposta (b) no tempo de resposta (zoom destacando o *downtime*); (c) na vazão de demanda

4. Conclusão

Neste trabalho, desenvolvemos um ambiente computacional virtualizado para alocação dinâmica de máquinas virtuais através de duas abordagens de migração amplamente utilizadas na literatura e disponibilizadas, por padrão, pelo hipervisor Xen: *stop-and-copy* e pré-cópia.

Além do *downtime* apresentado pela *stop-and-copy* ter sido pelo menos cinco vezes maior que o da pré-cópia, ele varia de acordo com o tamanho de memória das VMs. Portanto, para VMs com memória muito grande o *downtime* pode atingir valores inaceitáveis, comprometendo a capacidade de resposta dos serviços disponibilizados pelas VMs. Apesar no *downtime* na pré-cópia ser bem inferior ao da *stop-and-copy*, seu valor é bem acima do nível esperado em milésimos de segundo, previamente relatados na literatura para uma série de cargas [Clark et al 2005] [Oliveira 2010]. Entretanto, a combinação Olio/Faban foi utilizada em outros trabalhos para avaliar o impacto da *live migration* no desempenho do sistema [Voorsluys et al 2009] e o mesmo *downtime* e comportamento do tempo de resposta foi encontrado, reforçando os resultados alcançados neste trabalho.

Notamos ainda que a pré-cópia provocou um incremento no tempo de resposta das aplicações, caracterizando quebra de SLA para os *workloads* *Faban/SampleWeb* e

Faban/Olio, enquanto a *stop-and-copy* causou indisponibilidade dos serviços. Vale a pena ressaltar que a pré-cópia impacta negativamente no desempenho das aplicações *Sample Web*, visto que o processo de migração consome recursos de rede e processamento destinados à aplicação, resultando em um incremento no tempo de resposta e uma redução de 69% da vazão das requisições durante o processo de migração, enquanto a *stop-and-copy* provoca queda de 100% no tempo de resposta e na vazão da aplicação, causando indisponibilidade do serviço, o que torna essa abordagem inapropriada para aplicações com rígidas restrições de tempo, como, por exemplo, aplicações de tempo real.

Dentre as perspectivas de trabalhos futuros, pretendemos realizar experimentos com novas configurações de hardware (memória e CPU) e utilizar a técnica Dynamic Self-Ballooning (DSB) para reduzir o número de páginas livres transferidas durante a migração e, assim, melhorar o desempenho da pré-cópia e pós-cópia [Hines 2009]. Além disso, planejamos utilizar a abordagem de migração pós-cópia na avaliação, que tem como objetivo a redução do tempo de realocação das VMs durante o processo de migração em comparação com as abordagens *stop-and-copy* e pré-cópia.

Referências

- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003) “Xen and the art of virtualization”. In *SOSP '03: Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 164-177, New York, NY, USA. ACM.
- Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., and Warfield, A. (2005) “Live migration of virtual machines”. In *NSDI'05: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, pages 273-286, Berkeley, CA, USA. USENIX Association.
- Cully B., Lefebvre G., Meyer D. T., Karollil A., Feeley M. J., Hutchinson N. C., and Warfield A. (2008) “Remus: High Availability via Asynchronous Virtual Machine Replication”. In *NSDI'08: Proceedings of 5th Symposium on Networked Systems Design and Implementation*, San Francisco, CA, USA.
- Dunlap G. W., S. King T., Cinar S., Basrai M., and Chen P. M. (2002) “ReVirt: Enabling Intrusion Analysis through Virtual-Machine Logging and Replay”. In *OSDI'02: Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, pages 211-224, Boston, MA, USA. ACM Press.
- Faban (2010). Faban Harness and Benchmark Framework. Disponível em: <http://java.net/projects/faban/>
- Hines, M. R. and Gopalan, K. (2009) “Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning”. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Washington, DC, USA.
- Hirofuchi, T., Nakada, H., Itoh, S., and Sekiguchi, S. (2010) “Enabling Instantaneous Relocation of Virtual Machines with a Lightweight VMM Extension”. In *Proceedings of the 2010 10th IEEE/ACM international Conference on Cluster, Cloud and Grid Computing*, pages 73-83, Washington, DC.
- Liu, H., Jin, H., Liao, X., Hu, L., and Yu, C. (2009). “Live migration of virtual machine based on full system trace and replay”. In *HPDC '09: Proceedings of the 18th ACM*

- international Symposium on High Performance Distributed Computing*, pages 101-110, Garching, Germany. ACM.
- Netperf Homepage (2010). Disponível em: <http://www.netperf.org/netperf/>. Acesso: Novembro de 2010.
- Olio (2010). Olio Index. Disponível em: <http://incubator.apache.org/olio>
- Oliveira, C., Petrucci, V., and Loques, O. (2010) “Impact of server dynamic allocation on the response time for energy-efficient virtualized web clusters”. In: *XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - 12th Brazillian Workshop on Real-Time and Embedded Systems (WTR)*, Gramado, RS.
- OpenXenManager (2010). Disponível em: <http://www.openxenmanager.com/>. Acesso: Outubro de 2010.
- Osman, S., Subhraveti, D., Su, G., and Nieh, J. (2002) “The design and implementation of Zap: A system for migrating computing environments”. In *OSDI'02: Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, pages 361–376.
- Schmidt, B. K. (2000) “Supporting Ubiquitous Computing with Stateless Consoles and Computation Caches”. PhD thesis, Computer Science Dept., Stanford University.
- Sobel, W., Subramanyam, S., Sucharitakul, A., Nguyen, J., Wong, H., Patil, S., Fox, A., and Patterson, D. (2008) “Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0”. In *CCA'08: Proceedings of the 1st Workshop on Cloud Computing*.
- The Apache Software Foundation (2010). Apache HTTP server version 2.2. Disponível em: <http://httpd.apache.org/docs/2.2/>
- The SampleWeb Application (2010). Creating your first Workload in Faban. Disponível em: <http://testnscale.com/docs/CreatingWorkloadTutorial.html>
- Voorsluys, W., Broberg, J., Venugopal, S., and Buyya, R. (2009) “Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation”. In *Proceedings of the 1st international Conference on Cloud Computing*, pages 254-265. Lecture Notes In Computer Science, vol. 5931. Springer-Verlag, Berlin, Heidelberg. DOI= http://dx.doi.org/10.1007/978-3-642-10665-1_23.
- Wood, T., Shenoy, P., Venkataramani, A., and Yousif, M. (2009) “Sandpiper: Black-box and gray-box resource management for virtual machines”. In *Comput. Netw.*, pages 2923-2938. DOI= <http://dx.doi.org/10.1016/j.comnet.2009.04.014>.
- Xen (2010). Xen Cloud Platform - Advanced Virtualization Infrastructure for the Clouds. Disponível em: <http://www.xen.org/products/cloudxen.html>. Acesso: Outubro de 2010.
- Xen (2010). What is Xen?. Disponível em: <http://www.xen.org/>. Acesso: Outubro de 2010.
- Xu M., Malyugin, V., Sheldon, J., Venkitachalam, G., and Weissman, B. (2007) “ReTrace: Collecting Execution Trace with Virtual Machine Deterministic Replay”. In *MoBS'07: Proceedings of the Third Annual Workshop on Modeling, Benchmarking and Simulation*.