

# Uma solução de baixo custo para a Migração de Máquinas Virtuais

Gustavo P. Alkmim<sup>1</sup>, Joaquim Quinteiro Uchôa<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Lavras (UFLA) – Lavras, MG – Brasil

galkmim@gmail.com, joukin@ufla.com.br

**Abstract.** *In this work, we present an evaluation of the performance of a low-cost solution for live migration of virtual machines. The research has used para-virtualization, because it has a good performance and need no special hardware support, such as in full virtualization. As hypervisor, Xen was chosen, since this has been more appropriate for the deployment of servers on virtual machines. The low cost of the solution comes from the use of free tools and a virtual storage instead of a physical storage.*

**Resumo.** *O objetivo deste trabalho é analisar o desempenho de uma solução de baixo custo para a migração de máquinas virtuais em tempo real. Para isto foi utilizada a paravirtualização por ter esta um bom desempenho e não necessitar de um hardware com um suporte especial, como é o caso da virtualização completa. Como hipervisor foi escolhido o Xen, uma vez que este tem se mostrado mais adequado para a implantação de servidores em máquinas virtuais. O baixo custo da solução é devido a utilização de ferramentas livres e a utilização de um storage virtual, ao invés de utilizar um storage físico.*

## 1. Introdução

A utilização de máquinas virtuais tem se tornado uma alternativa para sistemas de computação pelas vantagens em custo, isolamento e portabilidade [Alkmim 2009]. Além disto os computadores pessoais modernos possuem recursos suficientes para utilizar a virtualização na criação de muitas máquinas virtuais eficientes, o que colaborou para a difusão da técnica.

A migração de máquinas virtuais consiste na transferência de uma máquina virtual que está sendo executada em um servidor físico para outro servidor físico. Neste trabalho foi abordado a migração em tempo real (*Live Migration*). As ferramentas utilizadas serão o hipervisor Xen, o *Open iSCSI* e o *iSCSI Enterprise Target*. Este último é uma ferramenta que permite exportar uma partição em computador como se ela fosse um *storage* físico. O objetivo deste trabalho é propor uma solução para a migração de máquinas virtuais utilizando uma solução de baixo custo e avaliar sua eficiência e confiabilidade. O Xen foi escolhido com hipervisor porque as outras soluções livres de virtualização não possuem soluções maduras para a migração de máquinas virtuais.

A migração pode ser utilizada para balanceamento de carga entre dois computadores ao permitir que um ou mais serviços seja transferido de um computador para outro bastando migrar uma ou mais máquinas virtuais. Em servidores de alta disponibilidade a técnica também pode ser utilizada. Quando uma máquina física estiver comprometida, a

máquina virtual é migrada para outra máquina física, permitindo que o serviço continue sendo executado sem prejuízo aparente.

Um trabalho relacionado foi feito por [Clark et al. 2005], que fez a migração de máquinas virtuais utilizando o Xen e o protocolo iSCSI, porém utilizou um dispositivo *storage* para armazenar o sistema de arquivos da máquina virtual. Como a aquisição de um dispositivo *storage* pode ter um custo relativamente alto, neste trabalho foi utilizado um *storage* virtual, ou seja, uma partição de um computador irá simular um *storage*.

Na seção seguinte serão abordados os principais conceitos envolvidos na tecnologia de virtualização. Na Seção 3 serão discutidas algumas características do hipervisor Xen. Na Seção 4 será descrita de uma maneira simplificada a forma como é feita a migração de máquinas virtuais. Na Seção 5 será discutido o funcionamento da ferramenta iSCSI. Na Seção 6 será mostrada a metodologia utilizada e os testes realizados. Na Seção 7 serão mostrados os resultados obtidos e, finalmente, na Seção 8, serão mostradas as conclusões e discutidos alguns possíveis trabalhos futuros.

## 2. Virtualização

De uma forma simplificada, uma máquina virtual é uma simulação de um computador físico em um computador real. De acordo com [Popek and Goldberg 1974], uma máquina virtual é um duplicata eficiente e isolada de uma máquina real. Sendo uma duplicata, qualquer efeito produzido na máquina virtual deve ter efeitos idênticos aos executados em uma máquina física.

O monitor de máquinas virtuais (VMM - *Virtual Machine Monitor*) é o *software* executado sobre a máquina física e é o responsável por criar e gerenciar as máquinas virtuais [Popek and Goldberg 1974]. É o VMM quem divide os recursos do *hardware* físico de acordo com a configuração de cada máquina virtual [Rose 2004]. Ele é capaz de criar várias máquinas virtuais independentes, cada uma funcionando como se fosse um computador real, de tal forma que um usuário conectado não perceba que está utilizando um ambiente virtual [Rose 2004], [Huang et al. 2007].

O monitor de máquinas virtuais, também chamado de hipervisor, pode ser implementado de duas formas básicas: ele pode ser executado diretamente sobre o *hardware* da máquina real ou ser um serviço executado sobre o sistema operacional da máquina hospedeira [Laureano 2004]. No primeiro caso, todos os recursos da máquina virtual são gerenciados pelo VMM, não existe nenhuma aplicação sendo executada concomitantemente com o monitor de máquinas virtuais e todos os recursos são distribuídos entre as máquinas virtuais. No segundo, é apenas um serviço responsável por gerenciar as máquinas virtuais, como qualquer outro serviço como o Apache ou o MySQL. Várias outras aplicações podem ser executadas no sistema anfitrião junto com o monitor de máquinas virtuais [Laureano 2004]. Neste caso os recursos do *hardware* são divididos entre as máquinas virtuais e os outros serviços executados no sistema anfitrião.

Existem dois tipos principais de virtualização: a virtualização completa e a paravirtualização. Na virtualização completa, o sistema operacional hóspede pode ser executado sem nenhuma modificação, o que é necessário para sistemas operacionais como o Windows. Na paravirtualização, o *kernel* do sistema operacional hóspede precisa ser modificado para chamar o VMM sempre que executar uma instrução que possa alterar o estado do sistema, uma instrução sensível [Rose 2004].

Uma vantagem útil para a migração de máquinas virtuais proveniente da técnica de virtualização é que o *hardware* da máquina virtual é o *hardware* “simulado” pelo hipervisor. Desta forma, é possível transferir uma máquina virtual de uma máquina real para a outra, mesmo que estas possuam um *hardware* diferente, pois para a máquina virtual o *hardware* continua sendo o mesmo.

### 3. O Hipervisor Xen

O Xen, desenvolvido inicialmente pela Universidade de Cambridge, é um monitor de máquinas virtuais para a arquitetura x86, que utiliza a técnica de paravirtualização [Barham et al. 2003], [Urschei et al. 2007]. Em uma avaliação quanto ao *overhead* imposto pela presença do hipervisor em sistemas de alto desempenho (HPC – *High Performance System*), a conclusão foi a de que ele é estatisticamente insignificante, comparado ao sistema sem a utilização do Xen [Youseff et al. 2006].

Como o Xen utiliza a técnica de paravirtualização, é necessário, conforme dito anteriormente, que a máquina física e as máquinas virtuais tenham o *kernel* do sistema operacional modificado. É possível também utilizar um *driver* do qemu que permite a implementação da técnica de virtualização completa, o que permite que sistemas operacionais que não podem ser modificados, como o Windows, possam ser executados não modificados com Xen.

Uma característica interessante do Xen, é que ele não é somente uma aplicação que é executada no sistema operacional da máquina hospedeira. Ele se integra ao *kernel* do sistema operacional, permitindo que o usuário opte por utilizar ou não o serviço responsável pelo monitoramento das máquinas virtuais. Portanto é possível utilizar o Xen como sistema operacional base, sem o uso do monitor de máquinas virtuais.

O Xen garante 100% de compatibilidade binária para as aplicações executadas nas máquinas virtuais. Com isto, desde que o sistema operacional da máquina virtual seja suportado pelo Xen, é possível que qualquer aplicação possa ser executada sem problemas, mesmo que a arquitetura da máquina física não ofereça suporte a ela [Rose 2004].

Vários trabalhos tem sido desenvolvidos utilizando o Xen como hipervisor. [Egi et al. 2007], por exemplo, avaliou o Xen para a virtualização de roteadores, comparando-o com o Linux nativo. O resultado obtido foi que o máquina hospedeira teve um desempenho semelhante ao do Linux nativo e a máquina hóspede obteve um desempenho muito pobre.

[Nagarajan et al. 2007] utilizou a migração em tempo real do Xen combinado com um sistema de monitoramento de estado para avaliar a viabilidade de um sistema tolerante a falhas pró-ativo para sistemas HPC. Quando um nó estivesse danificado, o sistema de monitoramento do estado detectaria isto e realizaria a migração.

### 4. A Migração de Máquinas Virtuais

A migração de máquinas virtuais move uma máquina virtual de uma máquina física para outra e deve ser feita de forma transparente para o sistema operacional hóspede e para os clientes remotos da máquina virtual [Clark et al. 2005], [Bradford et al. 2007]. As máquinas virtuais fornecem um ambiente propício para a migração por encapsular todo o estado do *hardware* e do *software* que esta sendo executado dentro da máquina virtual [Nelson et al. 2005].

Para que a migração seja realizada, devem ser considerados três estados: o estado dos dispositivos virtuais, as conexões externas com os dispositivos e a memória física da máquina virtual [Nelson et al. 2005]. Todos estes estados devem permanecer o mesmo após a migração. A cópia de toda a memória da máquina virtual enquanto ela estivesse pausada levaria a um período de inacessibilidade muito grande. Portanto uma parte da memória deve ser migrada enquanto ela ainda estiver em execução e somente as partes modificadas devem ser migradas enquanto ela estiver pausada [Clark et al. 2005], [Bradford et al. 2007].

Uma outra característica importante na migração das máquinas virtuais é a consistência do sistema de arquivos [Bradford et al. 2007]. Ao término da migração, é necessário que o sistema de arquivos no computador de destino seja idêntico ao sistema de arquivos no computador de origem. Esta característica é garantida no presente trabalho porque o sistema de arquivos utilizado é compartilhado, ou seja, as duas máquinas físicas têm acesso ao mesmo *storage*.

Quando se trabalha com a migração através de uma rede WAN (*Wide Area Network*) é necessário considerar também outros fatores para que a transparência seja mantida, como o *storage* não ser compartilhado pelos computadores e as redes locais serem diferentes, levando a máquina virtual a ter outro endereço IP. Estes problemas e outros relacionados à migração através de uma rede WAN foram abordados em [Bradford et al. 2007].

Para que a migração seja realizada com sucesso, é necessário que alguns requisitos sejam satisfeitos. Em primeiro lugar, a máquina de origem e de destino precisam estar executando o serviço Xen. Em segundo lugar, a máquina de destino precisa ter recursos suficientes para sustentar a máquina virtual que está sendo migrada. Em terceiro lugar, para este trabalho, as duas máquinas deverão estar na mesma LAN. E, por fim, a máquina de origem e a máquina de destino precisam ter acesso ao sistema de arquivos da máquina virtual.

Para que o quarto objetivo seja satisfeito, é necessário a existência de um *storage*, para que as duas máquinas físicas tenham acesso ao sistema de arquivos da máquina virtual. O grande problema desta solução é o custo para se montar um sistema de computação utilizando *storage*. Uma solução alternativa, adotada neste trabalho, é fazer com que um computador simule um *storage*. Uma partição deste foi exportada para os computadores entre os quais foi feita a migração. Isto foi feito utilizando uma ferramenta que utiliza o protocolo iSCSI, que será discutido na próxima seção.

## 5. iSCSI (Internet SCSI)

Para que seja possível a migração, é necessário a utilização de um *storage* para que a máquina de origem e a máquina de destino tenham acesso ao sistema de arquivos da máquina virtual. Para estabelecer a comunicação entre o *storage* e os computadores físicos é necessário a utilização de um protocolo que determine as regras desta comunicação. Um protocolo bastante utilizado para este fim é o SCSI (*Small Computer Systems Interface*).

A aquisição de um *storage* exige um investimento relativamente elevado, pois, além da necessidade da compra do *storage* em si, é necessário também a aquisição de cabos e conectores para fazer a comunicação com o computador. Além disto, existe a

limitação física imposta pela utilização de cabos especiais (que não são os da rede internet padrão).

Como um dos objetivos deste trabalho é implementar uma solução de baixo custo para a migração de máquinas virtuais, foi adotada uma solução alternativa que permite utilizar uma partição qualquer de um computador para fazer com que esta simule um dispositivo *storage*. Este *storage* simulado será chamado de *storage* virtual. Esta solução alternativa utiliza o protocolo iSCSI para fazer a comunicação entre o dispositivo *storage* virtual.

O iSCSI (*Internet Small Computer Systems Interface*) é um protocolo, que teve seu padrão definido pela RFC3720 [Satran et al. 2004], que define uma forma de transportar pacotes SCSI através dos protocolos TCP/IP. Desta forma, se faz desnecessário a utilização de cabos especiais, pois a própria internet pode ser utilizada para realizar a conexão dos computadores físicos ao *storage* virtual. Isto reduz os custos e a limitação física impostos pelos cabos especiais.

A arquitetura SCSI é uma arquitetura cliente-servidor. O cliente, chamado de SCSI *initiator*, envia comandos para o servidor, chamado de SCSI *target*. O servidor recebe os comandos enviados pelo *initiator* e os processa. O SCSI *target* exporta os dispositivos e é responsável por permitir conexões a eles. Como o iSCSI envia comandos SCSI através dos protocolos TCP/IP, os termos iSCSI *initiator* e iSCSI *target* são análogos ao *initiator* e ao *target* da arquitetura SCSI

A ferramenta iSCSI *target* utilizada neste trabalho permite que uma partição local seja mapeada como se fosse um disco *storage*. Desta forma o cliente ao se conectar no servidor terá a impressão de que a partição que está acessando é uma partição de um *storage*.

## 6. Metodologia

### 6.1. Ambiente Computacional

Para realizar o trabalho foi montado um sistema de computação utilizando três máquinas reais conforme é mostrado na Figura 1. Ela representa as ligações físicas entre as máquinas.

As máquinas Xenserver1 e Xenserver2 possuem o Debian Linux como sistema operacional e o Xen (XEN) e o *Open* iSCSI (OPENISCSI) instalados. O Xen é o monitor de máquinas virtuais e o *Open* iSCSI é a ferramenta iSCSI *initiator*. A máquina iSCSIserver possui o Gentoo Linux como sistema operacional e o iSCSI *Enterprise Target* (IET) instalado que é a ferramenta iSCSI *target* utilizada. Todas estas ferramentas são livres. O esquema da configuração lógica das máquinas, incluindo a máquina virtual é mostrado na Figura 2.

O iSCSIserver exporta a mesma partição para o Xenserver1 e para o Xenserver2, utilizando o iSCSI *Enterprise Target*. A máquina virtual compartilha o processamento, a memória e a rede da máquina hospedeira. O disco utilizado na máquina virtual é o obtido do iSCSIserver. O objetivo é migrar a máquina virtual localizada em Xenserver1 para Xenserver2. Como as duas máquinas possuem configurações diferentes, o desempenho antes da migração foi diferente do desempenho depois da migração. Este fato não irá

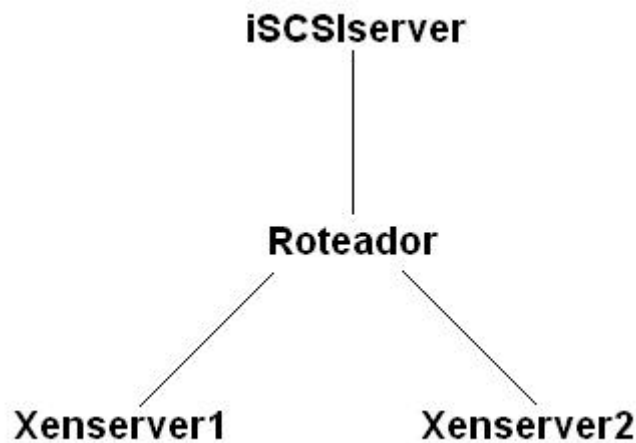


Figura 1. Ligação física do sistema de computação

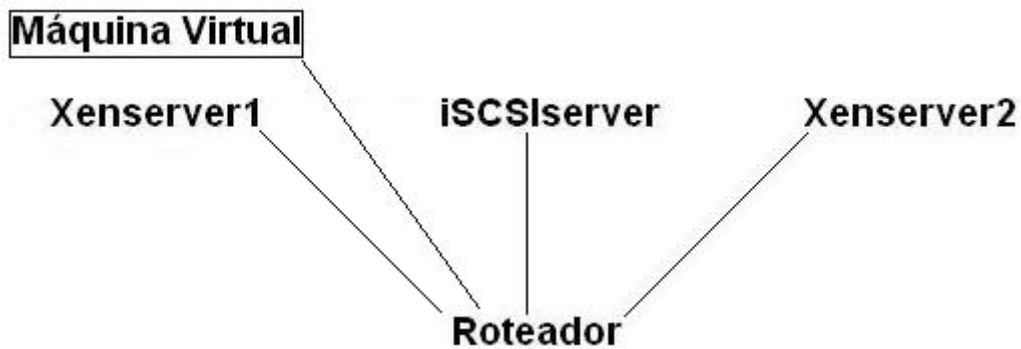


Figura 2. Ligação lógica do sistema de computação

influenciar este trabalho, uma vez que este tem como objetivo analisar o desempenho durante a migração.

A seguir são mostradas as configurações das máquinas utilizadas:

- Xenserver1
  - Processador: Pentium *Dual Core* 1.60 GHz
  - Memória RAM: DDR2 667MHz 2,5 GB
  - Disco Rígido: HD sata 7200rpm 160 GB.
  - Placa de Rede Ethernet 100Mbits Realtek
- Xenserver2
  - Processador: Pentium 4 3.06 GHz
  - Memória RAM: DDR2 533MHz 1 GB
  - Disco Rígido: HD pata 7200rpm 80 GB.
  - Placa de Rede Ethernet 100Mbits Realtek
- iSCSIserver
  - Processador: Pentium 4 4.06 GHz
  - Memória RAM: DDR 400MHz 512MB
  - Disco Rígido: HD pata 7200rpm 80 GB.
  - Placa de Rede Ethernet 100Mbits Realtek

O *switch* utilizado nos testes foi um Dlink de 100Mbps, modelo DL-604. Um fato importante é que a rede utilizada é de 100Mbps gerando um gargalo no sistema. No processo toda a memória RAM alocada para a máquina virtual deverá ser transferida via rede da máquina Xenserver1 para a Xenserver2. Ao mesmo tempo, o disco também estará sendo exportado via rede.

## 6.2. Testes

Duas ferramentas de *benchmark* foram utilizadas nos testes: o dbench [dbench 2009] e o SysBench [SysBench 2009]. O primeiro é voltado para a análise de desempenho de sistemas de arquivos e o segundo é voltado para avaliar o desempenho de vários parâmetros de um sistema. Os parâmetros avaliados pelo SysBench foram a taxa de transferência da memória RAM e a quantidade de eventos processados por segundo.

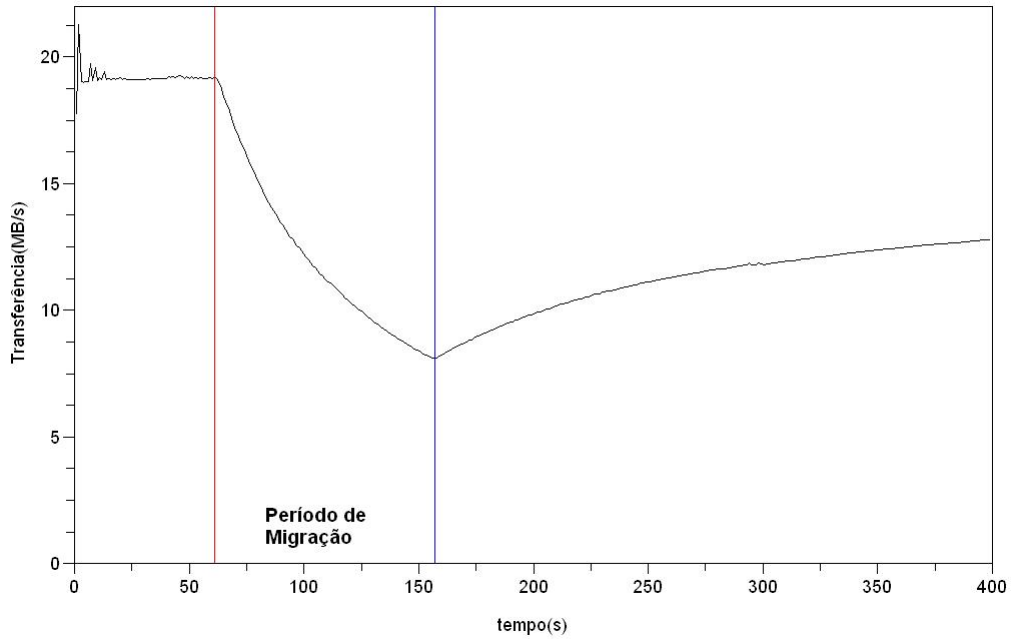
O dbench simula uma carga de operações no sistema de arquivos maior do que a normalmente utilizada em computadores comuns. Cada cliente realiza cerca de 90000 operações. Neste trabalho o dbench foi executado com 2 clientes. Com objetivo de simular uma utilização real, o disco é “aquecido” durante 80 segundos pela própria ferramenta. No teste de processamento do SysBench, cada evento consiste em calcular os números primos até o número especificado na linha de comando, que para o trabalho foi 1000. O SysBench foi executado utilizando 20 *threads*, valor medido experimentalmente para obter um melhor desempenho. O teste de memória do SysBench consiste em realizar leituras ou escritas sequenciais na memória.

Foram feitos sete tipos de testes diferentes com objetivo de avaliar o desempenho de vários aspectos da máquina virtual antes e durante a migração. O primeiro teste, com o objetivo de avaliar o desempenho do sistema de arquivos, consistiu na execução do dbench. Ele foi executado sozinho na máquina virtual antes, durante e após a migração. A migração foi iniciada 60 segundos após o término do aquecimento. O período de migração foi de 1 minuto e 36 segundos, com resultados apresentados nas figuras 3 e 4. A Figura 3 mostra a taxa de transferência do sistema de arquivos por segundo e o Figura 4 mostra a latência do sistema de arquivos. Os gráficos não mostram o período de aquecimento.

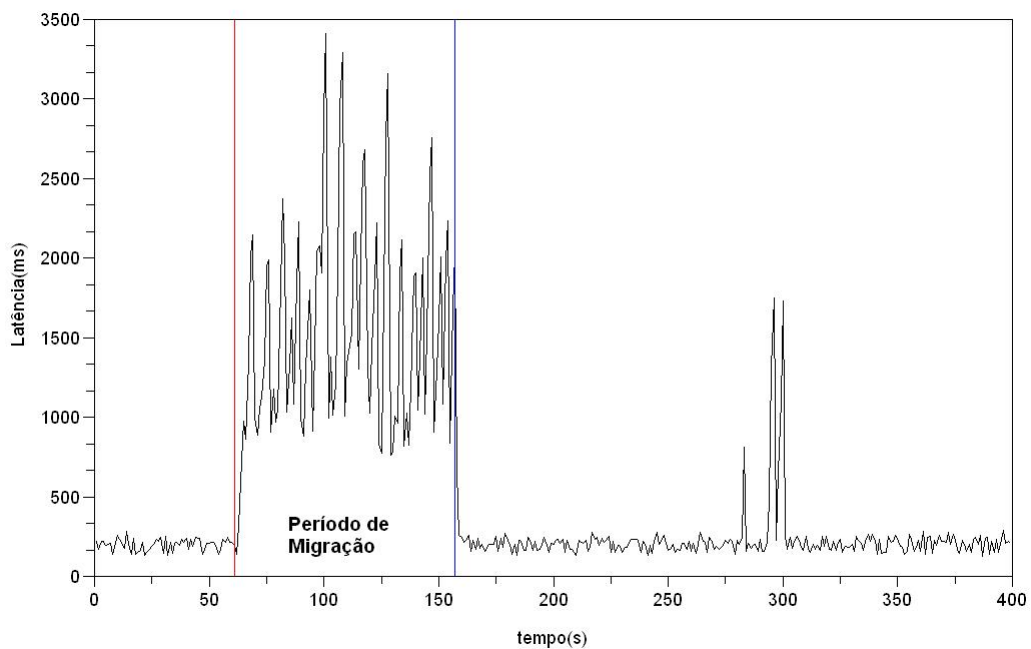
A partir das figuras 3 e 4, pode-se perceber que durante o período de migração, a redução do desempenho foi de aproximadamente 55% no ponto mais baixo, um valor alto para algumas aplicações. A latência também alcançou valores elevados e foi muito inconstante durante a migração.

Com o objetivo de avaliar o desempenho do processamento, o segundo teste foi a execução do benchmark SysBench, avaliando a quantidade de eventos processados por segundo. Ele foi executado na máquina virtual antes do início da migração. Após 60 segundos, a migração foi iniciada. Os resultados são mostrados na Figura 5. A redução de processamento foi de aproximadamente 16% durante a migração, um valor considerável, porém para aplicações normais, não causaria problemas.

Com o objetivo de avaliar o desempenho da memória, o terceiro teste foi a execução do *benchmark* SysBench, avaliando a quantidade de escritas por segundo. Ele foi executado na máquina virtual antes do início da migração. Após 60 segundos, a migração foi iniciada. Os resultados são mostrados na Figura 6. A redução da taxa de transferência foi praticamente imperceptível.



**Figura 3. Execução do dbench (taxa de transferência)**

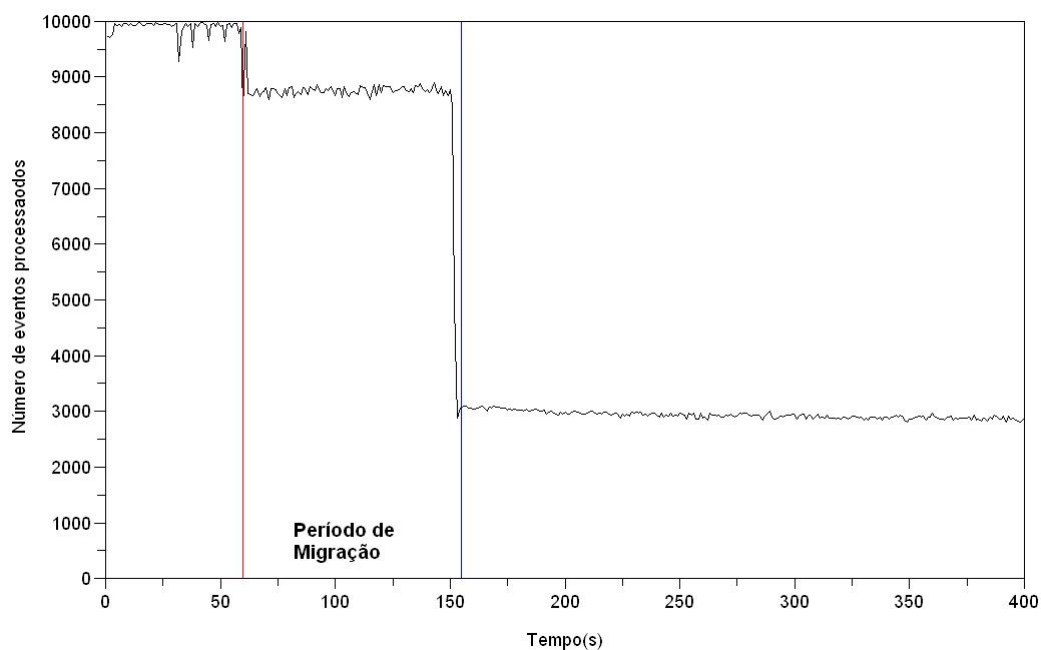


**Figura 4. Execução do dbench (latência)**

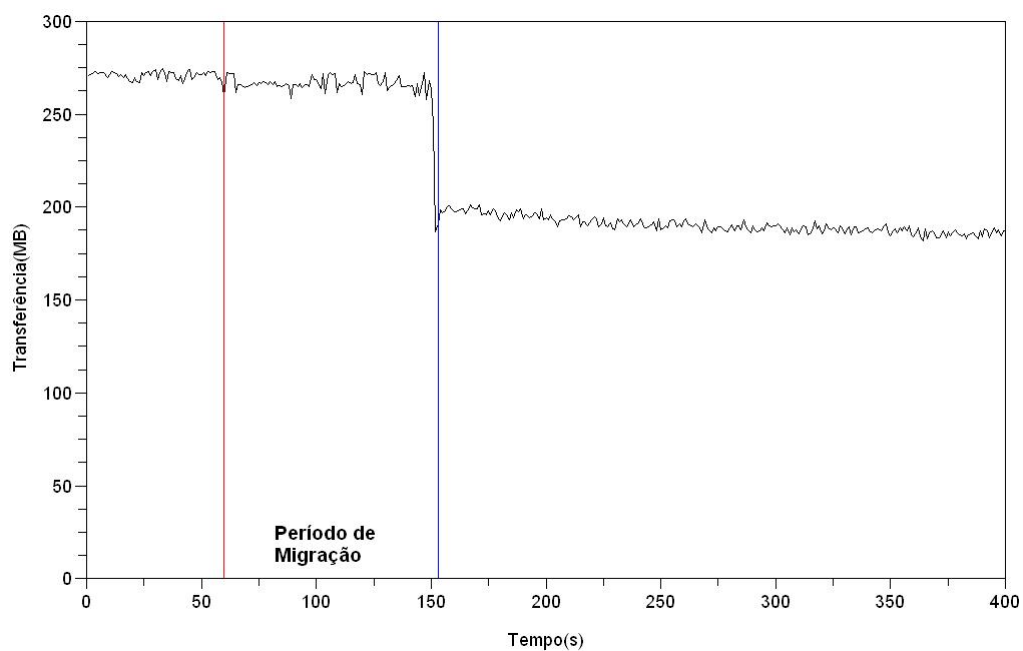
O quarto teste foi a execução de um comando ping na máquina virtual para avaliar qual seria o período em que esta permaneceria inacessível durante a migração. Os resultados são mostrados na Figura 7. Nota-se que em nenhum momento a máquina virtual ficou inacessível. Durante o período de migração a latência aumentou muito pouco.

Os quatro primeiros testes explicados anteriormente permitiram avaliar o desempenho isolado dos principais fatores que influenciam no desempenho da máquina virtual





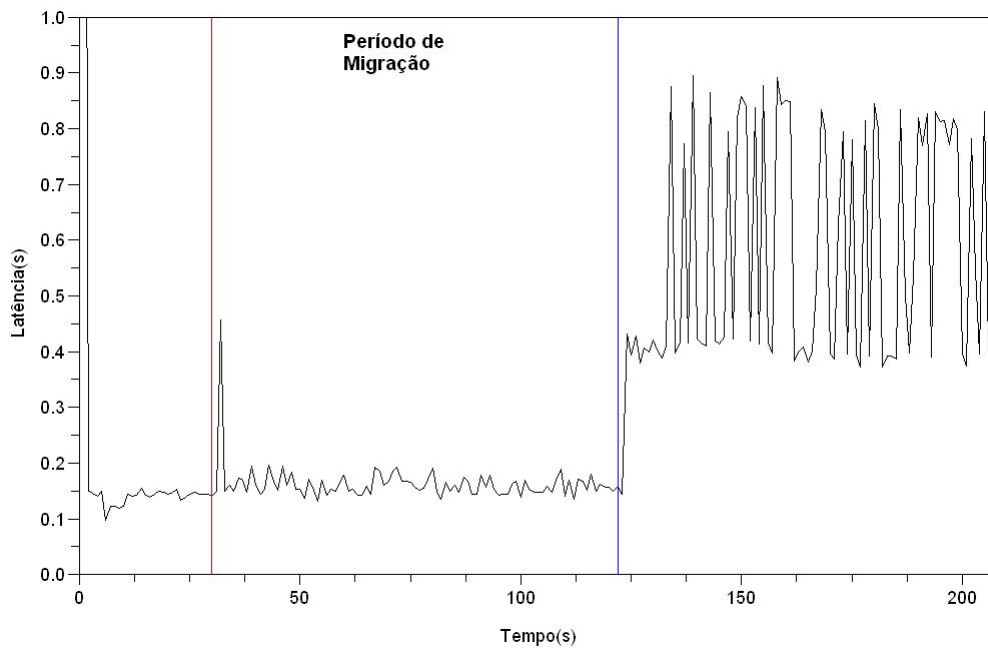
**Figura 5. Execução do SysBench (processamento)**



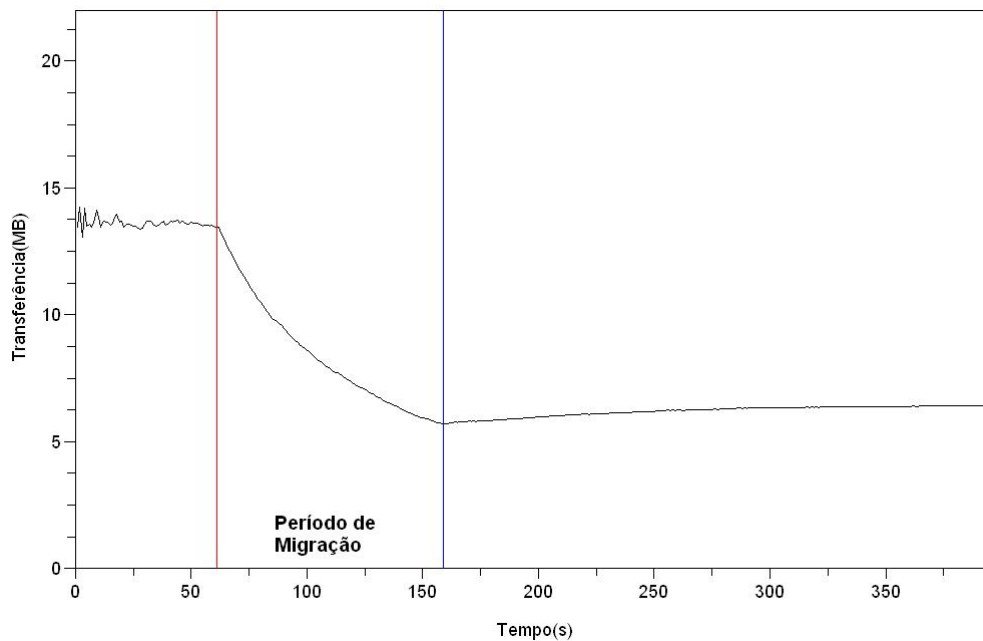
**Figura 6. Execução do SysBench (memória)**

durante a migração: o sistema de arquivos, a memória, o processamento e a acessibilidade. O quinto teste é a execução dos três primeiros testes simultaneamente, com o objetivo de simular uma carga maior no sistema. Os resultados são mostrados nas figuras 8, 9, 10 e 11. É interessante observar que em nenhum momento a máquina virtual ficou fora do ar. Confirmando o resultado obtido no teste 1, o acesso a disco ficou comprometido.

O sexto teste, com o objetivo de simular uma execução real que utilizasse uma



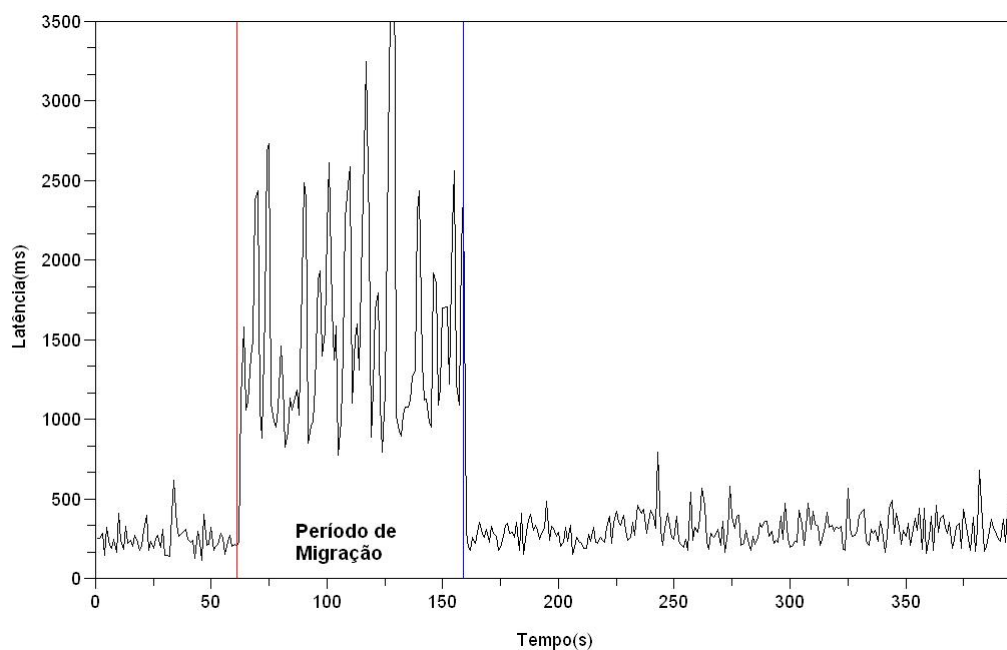
**Figura 7. Execução do comando ping**



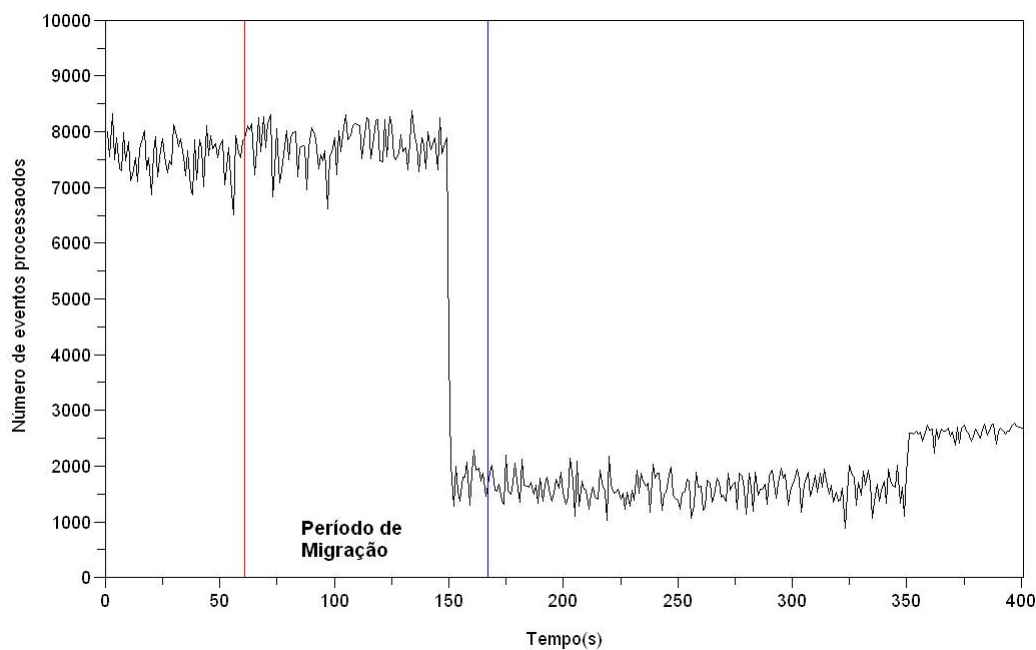
**Figura 8. Testes simultâneos: execução do dbench (taxa de transferência)**

quantidade maior de recursos computacionais, a migração foi executada durante a compilação do *kernel* do Linux. Para avaliar o desempenho neste ambiente foi executado um comando ping para a máquina virtual. Os resultados são mostrados na Figura 12. Após a migração a máquina virtual ficou *offline* durante 29 segundos.

Para que uma máquina virtual seja migrada, é necessário que toda a RAM alocada para ela seja transferida de uma máquina física para outra. O sétimo teste foi a execução um algoritmo, criado por nossa equipe, que realizava um escrita intensa na memória RAM



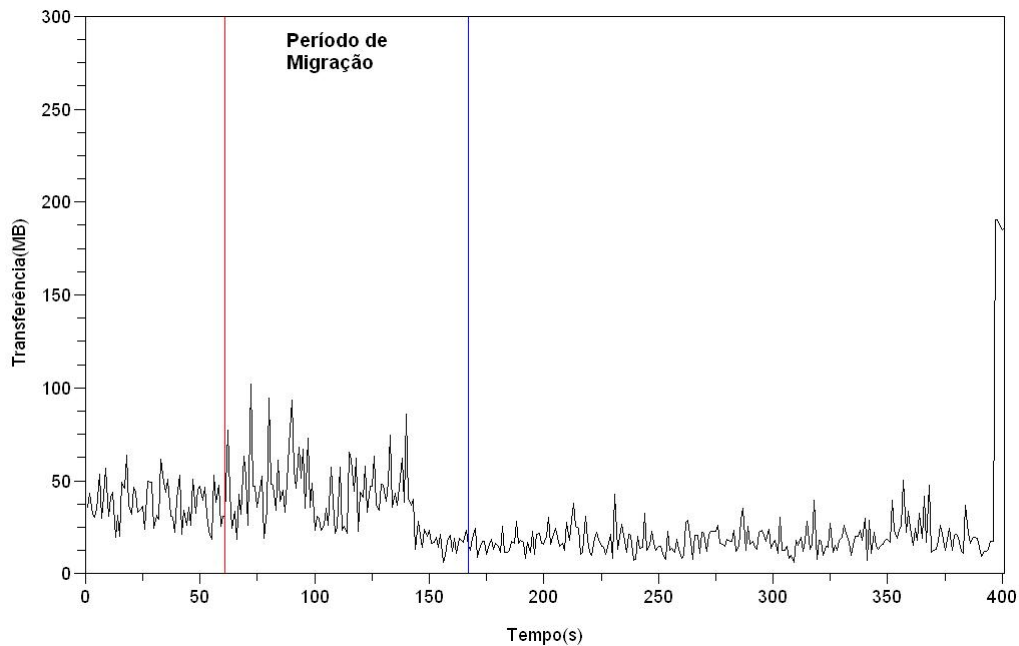
**Figura 9. Testes simultâneos: execução do dbench (latência)**



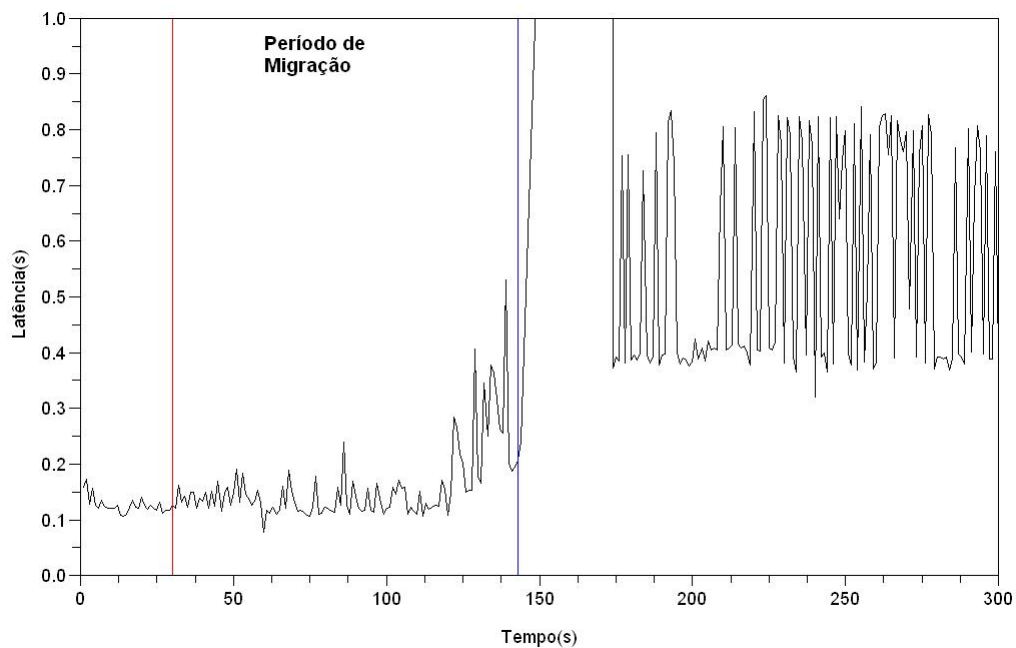
**Figura 10. Testes simultâneos: execução do SysBench (processamento)**

da máquina virtual enquanto esta era migrada. Este algoritmo foi baseado na criação de uma partição na memória RAM na qual eram feitas escritas. A taxa de escrita por segundo era maior do que a quantidade de memória que era migrada. Um teste semelhante foi efetuado por [Clark et al. 2005]. Para avaliar o desempenho foi executado um comando ping na máquina virtual. Os resultados são mostrados na Figura 13.

A partir do gráfico é possível perceber que a máquina virtual ficou inacessível



**Figura 11. Testes simultâneos: execução do SysBench (memória)**

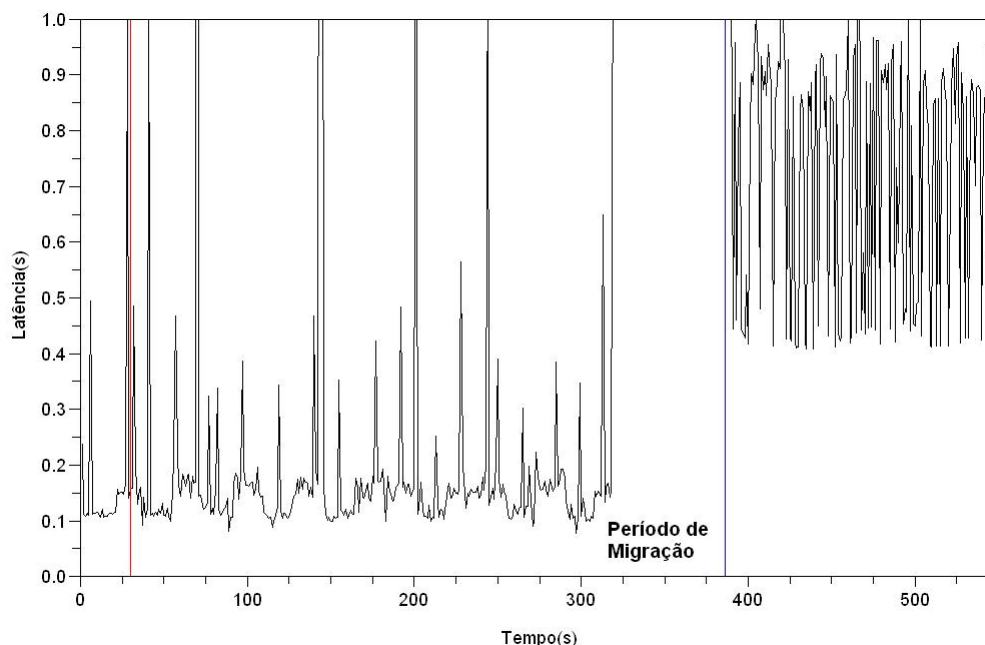


**Figura 12. Execução do comando ping durante a compilação do *kernel* do Linux**

durante um tempo considerável. A migração durou 5 minutos e 56 segundos e o período de inacessibilidade foi de 70 segundos. Apesar deste alto valor, o teste não representa um ambiente real.

## 7. Análise dos Resultados

A solução de baixo custo adotada possui como principal fraqueza a rede utilizada, que era de 100 Mbits/s. Com isso, a taxa de transmissão do disco foi baixa, uma vez que



**Figura 13. Execução do comando ping em um algoritmo de sobrecarga da RAM**

este estava limitado a largura da banda. Durante a migração, a redução de desempenho foi muito alta, a largura da banda foi dividida com a transmissão da memória RAM do computador de origem.

A migração de máquinas virtuais utilizando a solução proposta é viável para ambientes que não necessitem de muito acesso a disco, que corresponde a maioria das atividades mais comuns. Para aplicações que necessitem de um acesso elevado do disco durante um tempo constante, a adoção da técnica se torna inviável, uma vez que o disco não tem um bom desempenho. Para resolver este problema, a utilização de uma rede de 1Gbits/s ou de 10Gbits/s seria suficiente, porém contribuiria para o aumento do custo da solução. Além disso, a utilização de redes Gigabit não é tão comum quanto a utilização de redes de 100Mbits.

O período em que a máquina virtual ficou fora do ar foi de 70 segundos no pior caso, ao fim do qual ela voltou ao seu funcionamento normal. Apesar do alto valor, o teste representa um ambiente de intensa escrita na memória, sendo que a quantidade de dados enviados da memória era menor do que a quantidade de dados que era escrita. Uma possível solução para este problema seria a adoção de uma placa de rede com maior taxa de transmissão de dados. Uma outra solução seria a adoção de duas placas de redes separadas, uma para a migração (transferência da memória) e a outra responsável pela transferência do disco.

## 8. Conclusão

A solução proposta possui um baixo custo devido as ferramentas utilizadas serem livres e pela presença de um *storage* virtual ao invés de um *storage* físico. A utilização de equipamentos comuns e a adoção de uma rede de 100Mbits também colaboraram para o baixo custo da solução. Como a rede utilizada nos testes foi de 100Mbits, a taxa de transferência do sistema de arquivos ficou bastante comprometida. Uma possível solução

para este problema seria utilizar rede Gigabit ou então utilizar redes diferentes para a migração e para o *storage* virtual. A solução proposta é adequada para sistemas aonde não sejam executadas tarefas de acesso intenso ao disco simultaneamente com a migração, caso contrário, será necessário uma adaptação utilizando placas de rede mais rápidas.

Apesar do conhecimento prévio de que a rede de 100Mbits seria um gargalo, este valor foi adotado por ser o cenário mais comumente encontrado nos dias de hoje, não exigindo a aquisição de novas placas de rede. Além disso, o custo de placas de rede gigabit é mais elevado do que o de placa de redes de 100Mbits. Sendo assim, a utilização de uma rede gigabit iria contra os objetivos do trabalho. Porém, a análise do desempenho da solução proposta utilizando uma rede Gigabit ou duas placas de redes pode ser adotada em um trabalho futuro.

Durante o processo de migração, há interferência do protocolo ARP pois o MAC virtual passa a ser respondido pelo outro servidor físico. Como havia tráfego concorrente proveniente do disco e da memória no mesmo canal, é provável que switch tenha influenciado nos resultados. A investigação do impacto desta interferência poderá ser feita em um trabalho futuro.

Outro trabalho futuro seria a utilização do Ata-Over-Ethernet (AOE), ao invés do protocolo iSCSI e realizar uma comparação entre os dois.

## Referências

- Alkmim, G. P. (2009). Aplicações da virtualização em empresas. In *II WIA - Workshop de Informática Aplicada*.
- 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 *nineteenth ACM symposium on Operating systems principles*, pag 164-177.
- Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H. (2007). Livewide-area migration of virtual machines including local persistent state. In *3rd international conference on Virtual execution environments*, pages 179–189.
- Clark, C., Fraser, K., and Hand, S. (2005). Live migration of virtual machines. In *2nd conference on Symposium on Networked Systems Design and Implementation, Vol 2*, pag 273-286.
- dbench (2009). disponível em <http://samba.org/ftp/tridge/dbench/>, acessado em 04/2009.
- Egi, N., Greenhalgh, A., Handley, M., Hoerd, M., Mathy, L., and Schooley, T. (2007). Evaluation xen for router virtualization. In *16th International Conference on Computer Communications and Networks*.
- Huang, W., Liu, J., Abali, B., Koop, M., and Panda, D. (2007). Nomad: Migrating os-bypass networks in virtual machines. In *3rd international conference on Virtual execution environments*, pag 158-168.
- Laureano, M. A. P. (2004). Uma abordagem para a proteção de detectores de intrusão baseada em máquinas virtuais. Ph. D. Thesis, Pontifícia Universidade Católica do Paraná.

- Nagarajan, A. B., Mueller, F., Engelmann, C., and Scott, S. L. (2007). Proactive fault tolerance for hpc with xen virtualization. In *21st annual international conference on Supercomputing*.
- Nelson, M., Lim, B., and Hutchins, G. (2005). Fast transparent migration for virtual machines. In *USENIX Annual Technical Conference*, pages 391–394.
- Popek, G. J. and Goldberg, R. P. (1974). Formal requirements for virtualizable third generation architectures. In *Communications of the ACM, Vol 17, n 7, pag 412-421*.
- Rose, R. (2004). Survey of system virtualization techniques. University of California.
- Satran, J., Meth, K., Sapuntzakis, C., Systems, C., Chadalapaka, M., Co., H.-P., and Zeidner, E. (2004). *RFC3720 - Internet Small Computer Systems Interface (iSCSI)*. IETF - The Internet Engineering Task Force, <http://www.rfceditor.org/rfc/rfc3720>, acessado em 04/2009.
- SysBench (2009). disponível em <http://sysbench.sourceforge.net/>, acessado em 04/2009.
- Urschei, F., Pelegriani, E. J., Silva, M. A. L., Midorikawa, E. T., and Carvalho, T. C. M. B. (2007). Análise multiparamétrica do overhead de rede em máquinas virtuais. In *Proceedings of the IV Workshop de Sistemas Operacionais*, pag 816-827.
- Youseff, L., Wolski, R., Gorda, B., and Krintz, C. (2006). Paravirtualization for hpc systems. In *Technical Report Technical Report Number 200610, Computer Science Department, University of California, Santa Barbara*.