

Gustavo Prado Alkmim

Análise de Desempenho da Migração de Máquinas Virtuais utilizando Xen e iSCSI

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para a obtenção do título de Bacharel em Ciência da Computação

Orientador
Prof. Joaquim Quinteiro Uchôa

Lavras
Minas Gerais - Brasil
2009

Gustavo Prado Alkmim

Análise de Desempenho da Migração de Máquinas Virtuais utilizando Xen e iSCSI

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para a obtenção do título de Bacharel em Ciência da Computação

Aprovada em 18 de Junho de 2009

Prof. Bruno de Oliveira Schineider

Prof. Marluce Rodrigues Pereira

Prof. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil
2009

Agradecimentos

Em primeiro lugar agradeço a Deus pelo seu amor supremo e infalível que está presente em minha vida em todo tempo. Agradeço a Deus por ter me dado forças para escrever esta monografia e para superar todas as dificuldades que encontrei pelo caminho durante a graduação.

Agradeço aos meus familiares, em especial aos meus pais, Flávia Alkmim e José Luiz Alkmim, e à minha avó, Venturina Canaan, que estiveram presentes, mesmo que distantes fisicamente, orando por mim e me auxiliando sempre que eu precisei.

Agradeço também a minha namorada, Luanna Gouvêa, que esteve perto de mim, me dando apoio, mesmo quando eu estava estressado por causa deste trabalho.

Agradeço aos meus amigos, os quais não cito todos por não caber nesta página, porque eles são peças fundamentais em todos os momentos da vida.

Agradeço a todos os meus professores, em especial, à Marluce Pereira, ao Luiz Henrique, ao José Monserrat e ao Joaquim Uchôa (orientador deste trabalho), que me orientaram esclarecendo dúvidas e me ensinando a focar em um trabalho específico.

Não posso esquecer de agradecer ao guaraná em pó, ao guaraná em cápsulas e ao refrigerante de cola, que me mantiveram acordado nas madrugadas para escrever este trabalho.

Finalmente agradeço a todos que estiveram comigo durante esta etapa de minha vida. Mesmo não tendo citado o nome, vocês foram e serão muito importantes para mim. Que Deus abençoe a todos!

Sumário

1	Introdução	1
2	Referencial Teórico	5
2.1	Virtualização	5
2.2	Histórico	12
2.3	Aplicações	14
2.4	Ferramentas de Virtualização	18
2.5	Xen	19
2.6	<i>Live Migration</i>	21
2.7	O Protocolo iSCSI (Internet SCSI)	23
3	Metodologia	25
3.1	O trabalho Proposto	25
3.2	Ambiente Computacional	27
3.3	<i>Benchmarks</i>	30
4	Resultados e Análises	33
4.1	Desempenho do iSCSI Exportando o Disco da Máquina Virtual	33
4.2	Desempenho do iSCSI na Migração de Máquinas Virtuais	35
5	Conclusão	45

Lista de Figuras

2.1	VMM realiza a interface entre os recursos de <i>hardware</i> e a máquina virtual	6
2.2	Sistema de Máquinas Virtuais - Tipo 1	8
2.3	Sistema de Máquinas Virtuais - Tipo 2	9
2.4	Abordagens Híbridas - Tipo 1	9
2.5	Abordagens Híbridas - Tipo 2	10
2.6	Máquina Virtual iniciada no VirtualBox	11
2.7	Migração de máquinas virtuais com o Xen	22
3.1	O <i>storage</i> virtual	26
3.2	Ligação física do sistema de computação	27
3.3	Ligação lógica do sistema de computação	28
4.1	Resultados Obtidos	34
4.2	Execução do dbench (taxa de transferência)	35
4.3	Execução do dbench (latência)	36
4.4	Execução do SysBench (processamento)	38
4.5	Execução do SysBench (memória)	39
4.6	Execução do comando ping	40
4.7	Testes simultâneos: execução do dbench (taxa de transferência)	41
4.8	Testes simultâneos: execução do dbench (latência)	41
4.9	Testes simultâneos: execução do SysBench (processamento)	42
4.10	Testes simultâneos: execução do SysBench (memória)	42
4.11	Execução do comando ping durante a compilação do kernel do Linux	43
4.12	Execução do comando ping em um algoritmo de sobrecarga da RAM	43

Resumo

Este trabalho apresenta uma solução de baixo custo para a migração de máquinas virtuais. As ferramentas utilizadas foram o hipervisor Xen, o iSCSI *target* IET *Enterprise* e o iSCSI *initiator* *Open* iSCSI. Todas as ferramentas utilizadas, incluindo os *benchmarks*, são livres. Os resultados mostraram que utilizando uma rede de 100MBits e tendo o sistema de arquivos da máquina virtual em um *storage* virtual é possível realizar a migração com um baixo custo. O único ponto fraco da solução foi o desempenho da rede, que influenciou na latência do disco. Uma solução para isto seria a adoção de uma rede Gigabit. Como a solução envolveu a utilização do protocolo iSCSI para exportar um disco para uma máquina virtual, foi feita uma análise de desempenho deste tipo de utilização nas mesmas condições de baixo custo. O resultado mostrou que a perda de desempenho foi muito pequena, com excessão da rede.

Palavras-Chave: Virtualização; Migração de Máquinas Virtuais; Xen; iSCSI; Storage Virtual.

Capítulo 1

Introdução

No contexto deste trabalho, a virtualização será utilizada para definir a técnica de criação de computadores virtuais dentro de um computador físico, podendo cada um destes possuir diferentes arquiteturas de *hardware* e sistemas operacionais. Os recursos da máquina física são compartilhados pelas máquinas virtuais e, de modo geral, a quantidade de máquinas virtuais que um sistema pode suportar é definida pela quantidade de recursos que este possui e pela quantidade de recursos que as máquinas virtuais irão utilizar.

A virtualização não é um conceito recente. Ela teve sua origem nas décadas de 1960 e 1970 nos *mainframes* da IBM, porém continuou restrita aos computadores de grande porte devido às limitações de recursos dos computadores de pequeno porte (MEYER; SEAWRIGHT, 1970), (PARMELEE *et al.*, 1972). Com o passar do tempo, o poder de processamento, a memória e outros recursos, evoluíram até atingirem um nível que permitiu que não somente os supercomputadores, mas também os servidores de pequeno porte e os computadores pessoais tivessem condições de aplicar a virtualização, o que colaborou para o aperfeiçoamento da técnica (SOUZA, 2006) (BARHAM *et al.*, 2003).

Desde então, a utilização de máquinas virtuais tem se tornado uma alternativa para vários sistemas de computação pelas vantagens em custo, isolamento e portabilidade (ALKMIM, 2009). As vantagens em custo são obtidas no aproveitamento

dos recursos ociosos de uma máquina para a criação de outra, evitando a aquisição de novos equipamentos e, conseqüentemente, reduzindo o consumo de energia.

As máquinas virtuais podem ser completamente isoladas umas das outras, podendo ser utilizadas como ambiente para testes (WŁODARZ, 2007). Desta forma, é possível realizar vários testes simultâneos sem que um interfira no outro. Além disso, o isolamento também permite que vários usuários possam utilizar o mesmo computador, porém cada um executando um sistema operacional diferente e tendo a impressão de que está utilizando o computador sozinho (POPEK; GOLDBERG, 1974). Outra aplicação do uso da virtualização é a execução de vários serviços em um único computador físico, porém em máquinas virtuais diferentes. Desta forma, problemas como conflitos entre serviços e falhas em um serviço que podem comprometer o restante do sistema ou outros serviços são isolados.

As máquinas virtuais além de isoladas são altamente portáteis. Um computador virtual que esteja sendo executado em uma máquina física pode ser transferido para outra sem o surgimento de problemas advindos da diferença entre os dois *hardwares*. Esta técnica recebe o nome de migração de máquinas virtuais e pode ser de dois tipos: regular e *live migration*. Na migração regular, a máquina virtual é pausada, o conteúdo da memória é transferido de uma máquina física para outra e então a máquina virtual é iniciada novamente na máquina física de destino. Na *live migration*, o processo possui a mesma sequencia lógica, porém a máquina virtual não precisa ser pausada, sendo comum em servidores que não podem ter seus serviços interrompidos (XEN-MAN, 2005). Portanto, é necessário que a migração *online* seja feita de forma quase ou totalmente imperceptível para o usuário.

Em (CLARK; FRASER; HAND, 2005), é descrita uma implementação da migração de máquinas virtuais para o Xen (XEN, 2009) e são feitas análises de desempenho para avaliar a viabilidade da solução. Os testes de desempenho foram feitos utilizando o Xen e o protocolo iSCSI, porém utilizando um *storage* físico NAS (*Network Attached Storage*) robusto, cujo custo é relativamente alto, para armazenar o sistema de arquivos da máquina virtual. Os resultados do trabalho de

(CLARK; FRASER; HAND, 2005) mostraram que a migração de máquinas virtuais pode ser feita com um impacto mínimo nos serviços que estão sendo executados.

A migração de máquinas virtuais pode ser utilizada para o balanceamento de carga entre dois computadores, ao permitir que um ou mais serviços sejam transferidos 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.

O objetivo deste trabalho é o desenvolvimento de uma solução que permita realizar a migração *online* de máquinas virtuais com eficiência, confiabilidade e baixo custo. Neste trabalho foram adotadas tecnologias e ferramentas livres e que são utilizadas no mercado atualmente. As ferramentas utilizadas foram: Gentoo Linux (GENTOO, 2009), Debian Linux (DEBIAN, 2009), o hipervisor Xen (XEN, 2009), o iSCSI *target* IET (IET, 2009) e o iSCSI *initiator* openiSCSI (OPENISCSI, 2009). Uma parte dos resultados deste trabalho são apresentados em (ALKMIM; UCHÔA, 2009).

A grande diferença para a solução proposta nesta monografia com a solução proposta por *clark:05* está na utilização de um *storage* virtual ao invés de um *storage* físico. A aquisição de um *storage* físico possui um alto custo, foi utilizado um *storage* virtual, que é um computador simples simulando um dispositivo *storage*. Uma disco deste computador foi exportado para as máquinas entre as quais seria feita a migração através do protocolo iSCSI. Para certificar que a escolha do protocolo iSCSI para exportar o disco do *storage* virtual é viável, foi feita uma análise de desempenho do protocolo iSCSI exportando um disco para uma máquina virtual.

Para ambientes aonde não é necessário um ambiente computacional robusto, a utilização da migração de baixo custo adotada neste trabalho se apresentou viável, evitando assim a aquisição de um *storage* físico que possui um custo elevado. Além disso, verificar a viabilidade e os pontos fracos da utilização de um *storage* virtual permite ampliar a solução para ambientes que precisem de uma solução

mais robusta. A utilização de uma instalação local linux e de aplicações livres tornam a solução mais flexível do que a utilização de uma solução com um *storage* físico. Além disto, solução apresentou como unico ponto fraco o desempenho da rede devido a utilização de uma rede ethernet de 100 megabits. A utilização de conexões mais velozes podem tornar a solução viável inclusive para ambientes que necessitem de um alto desempenho.

Na Seção 2, é apresentado o referencial teórico desta monografia. Nesta seção são abordados os aspectos teóricos relacionados com a virtualização, suas origens, aplicações e uma visão geral das ferramentas que estão presentes no mercado voltadas para a virtualização. Também é abordado o Xen, suas principais características e alguns trabalhos relacionados. Nesta mesma seção, são discutidos a migração de máquinas virtuais e os pre-requisitos básicos para que o processo seja implementado. Por fim é definido o conceito de *storage* virtual e é mostrado a funcionamento do protocolo iSCSI. Na Seção 3 é apresentada a metodologia do trabalho. Na Seção 4 são apresentados os testes realizados e os resultados obtidos. Por fim na Seção 5 são apresentados as conclusões do trabalho e os possíveis trabalhos futuros.

Capítulo 2

Referencial Teórico

2.1 Virtualização

Uma máquina virtual é definida como sendo uma duplicata eficiente e isolada de uma máquina real (POPEK; GOLDBERG, 1974). Uma duplicata é uma cópia essencialmente idêntica a de uma máquina original, ou seja, qualquer programa executado em uma máquina virtual deve exibir efeitos idênticos ao que exibiria se fosse executado em uma máquina física.

Por serem eficientes, é necessário que uma parte dominante das instruções seja executada diretamente no *hardware* da máquina. Isto exclui do âmbito das máquinas virtuais os tradicionais simuladores (tentativa de imitar as funções de um dispositivo) e emuladores (tentativa de imitar o desenho do *hardware* de um dispositivo), por precisarem simular quase a totalidade das instruções antes de executá-las no *hardware* (POPEK; GOLDBERG, 1974). A máquina virtual encontra-se em um nível de abstração intermediário entre a máquina real e o emulador, no que se refere à forma como os recursos de *hardware* e de controle são abstraídos e usados pelas aplicações (LAUREANO, 2004).

O termo isolamento significa que a máquina virtual trabalha como se fosse um computador independente (POPEK; GOLDBERG, 1974). Este aspecto permite que o usuário não perceba que está trabalhando em uma máquina virtual ao invés de uma

máquina real. Além disso, falhas em uma máquina virtual não serão propagadas para as outras máquinas virtuais.

Uma outra definição é a de que uma máquina virtual é um sistema de computação no qual as instruções requisitadas por um programa podem ser diferentes daquelas realmente executadas pelo *hardware* para realizar a tarefa solicitada (PARMELEE *et al.*, 1972). Desta forma, ao ser solicitada uma ação à máquina virtual, as instruções serão executadas direto no *hardware* ou podem ser simuladas. As instruções simuladas exigem um custo adicional, pois exigem um pré-processamento antes de serem efetivamente executadas, influenciando diretamente no desempenho da máquina virtual.

Outro conceito importante relacionado com a virtualização é a do monitor de máquinas virtuais (VMM – *Virtual Machine Monitor*). Ele é a camada de *software* que faz a interface entre os recursos da máquina real e a máquina virtual (ROSE, 2004). Como mostrado na Figura 2.1, o VMM é responsável por criar a máquina virtual (POPEK; GOLDBERG, 1974) e gerenciar seu funcionamento.

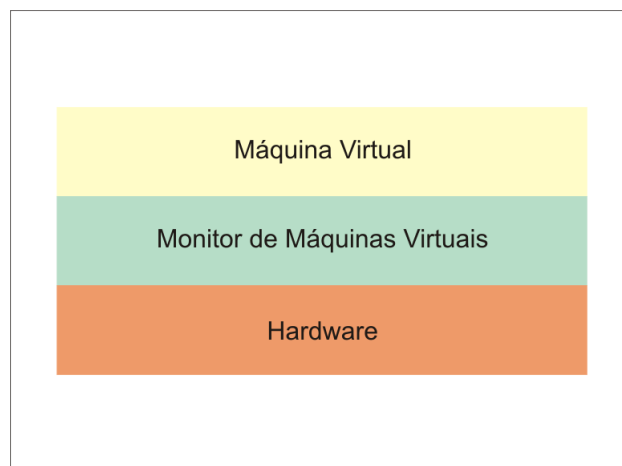


Figura 2.1: VMM realiza a interface entre os recursos de *hardware* e a máquina virtual

Devido à presença do monitor de máquinas virtuais é possível criar várias máquinas virtuais simultaneamente, cada uma contendo o seu próprio sistema opera-

cional e funcionando de forma independente uma da outra (ROSE, 2004), (HUANG *et al.*, 2007). O monitor de máquinas também fornece proteção para o sistema hospedeiro, impedindo que a máquina virtual realize alguma operação indevida no *hardware*.

O VMM é responsável por identificar as funções que precisam ser interpretadas antes de serem executadas no *hardware* da máquina física e realizar as interpretações necessárias. Desta forma, as instruções antes de serem executadas no *hardware* da máquina real, devem ser primeiramente passadas para o monitor de máquinas virtuais para que este execute as operações sobre o *hardware* e retorne o resultado para a máquina virtual.

Com o objetivo de maximizar o desempenho, o monitor de máquinas virtuais pode, sempre que possível, permitir que o sistema hóspede execute as instruções diretamente sobre o *hardware* da máquina física em modo usuário (LAUREANO, 2004). Caso não seja possível, o VMM toma o controle do sistema, realiza as interpretações necessárias, executa a instrução sobre o *hardware* e devolve o controle para a máquina virtual.

A Figura 2.2 mostra um sistema de máquinas virtuais sendo executado sobre um único *hardware* real. O monitor de máquinas virtuais é quem cria o *hardware* da máquina virtual, sobre o qual é executado o sistema operacional hóspede (*Guest OS*). O sistema operacional hóspede trabalha como se todas as instruções fossem executadas em uma máquina real fazendo com que o usuário não perceba que está trabalhando em uma estação virtual. Como os sistemas são isolados uns dos outros, se houver a necessidade de comunicação ela deverá ser feita via rede.

O monitor de máquinas virtuais 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). O primeiro caso, chamado de tipo 1, é ilustrado pela Figura 2.2, onde todos os recursos da máquina virtual são gerenciados pelo VMM. Não existe nenhuma aplicação sendo executada concomitantemente com o monitor de máqui-

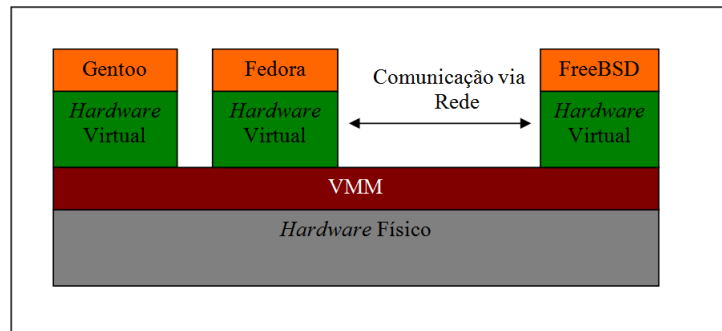


Figura 2.2: Sistema de Máquinas Virtuais - Tipo 1

nas virtuais e praticamente todos os recursos são distribuídos entre as máquinas virtuais (LAUREANO, 2004).

O segundo caso, chamado de tipo 2, é ilustrado pela Figura 2.3, onde o VMM é um processo sendo executado no sistema operacional hospedeiro, que é denominado de anfitrião. Ele é apenas o serviço responsável por gerenciar as máquinas virtuais, como qualquer outro serviço como o Apache (APACHE, 2009) ou o MySQL (MYSQL, 2009). 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 que estão sendo executadas no sistema anfitrião.

Existem ainda as abordagens híbridas que são o tipo 1 e o tipo 2 porém com algumas otimizações, para melhorar o desempenho das aplicações nas máquinas virtuais, principalmente no que se refere às operações de entrada e saída, por serem pontos cruciais no desempenho das máquinas virtuais (LAUREANO, 2004). A Figura 2.4 ilustra as otimizações inseridas nas abordagens do tipo 1 e a Figura 2.5 ilustra as otimizações inseridas nas abordagens do tipo 2. Existem outras otimizações que podem ser inseridas, mas estas são as mais comuns.

Como mostra a Figura 2.4, o sistema convidado tem acesso direto ao *hardware* (LAUREANO, 2004). Para isto, o núcleo do sistema convidado e o monitor de máquinas virtuais devem ser modificados. Com esta otimização, o *overhead* que

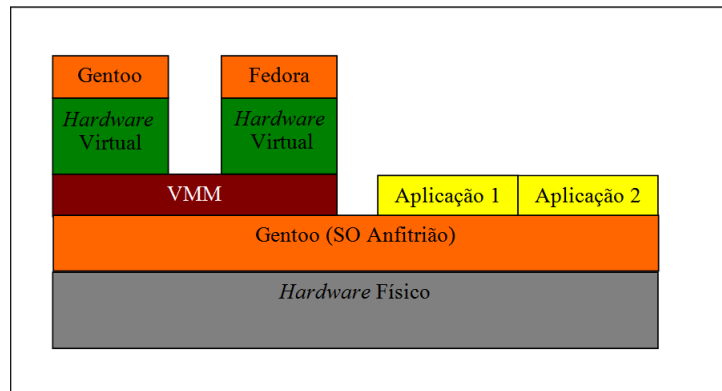


Figura 2.3: Sistema de Máquinas Virtuais - Tipo 2

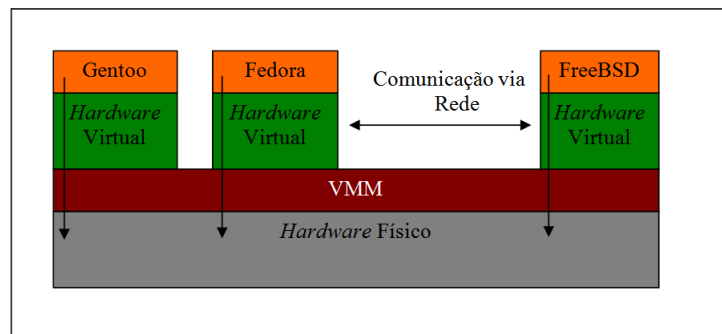


Figura 2.4: Abordagens Híbridas - Tipo 1

ocorre com a inserção da camada do VMM entre o sistema convidado e o sistema hospede é reduzido.

As otimizações inseridas nas abordagens do tipo 2 são parecidas com a descrita anteriormente para o tipo 1. (1) O sistema convidado pode acessar diretamente o sistema anfitrião. Para isto, partes da API do sistema anfitrião são oferecidas para o sistema convidado. (2) O sistema convidado tem acesso direto ao *hardware* físico. (3) O VMM pode acessar diretamente o *hardware*. Como no tipo 2 o monitor é executado como um processo no sistema hospedeiro, sem esta otimização seria ne-

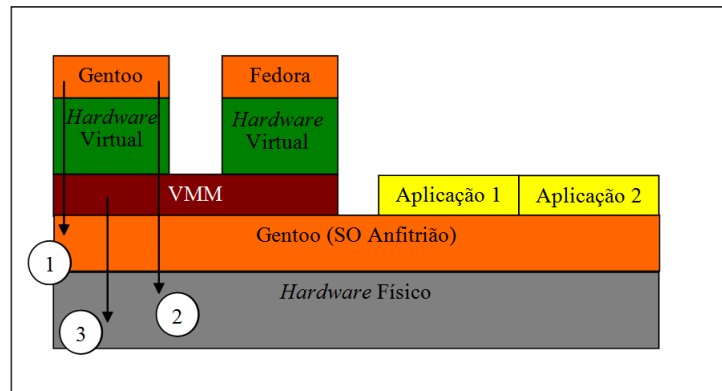


Figura 2.5: Abordagens Híbridas - Tipo 2

cessário que as informações fossem passadas para o sistema hospedeiro primeiro, para depois serem passadas para o *hardware*, o que aumentaria o overhead.

A classificação apresentada, foi quanto à forma de implementação do monitor de máquinas virtuais. É possível classificar também quanto ao tipo de virtualização, o que pode ser: virtualização completa, paravirtualização ou emulação de *hardware*.

Na virtualização completa, o sistema operacional hóspede é executado sem nenhuma modificação, o que é necessário para sistemas operacionais como o Windows. Isto é possível porque as instruções são intermediadas pelo MMV (JONES, 2006). Na paravirtualização, o *kernel* da máquina hóspede deve ser modificado para permitir que o monitor de máquinas virtuais execute tarefas protegidas. O *kernel* da máquina física também deve ser modificado devido às modificações feitas no *kernel* das máquinas hóspedes (URSCHEI *et al.*, 2007). Estas modificações no *kernel* são necessárias para suportar uma virtualização eficiente (HUANG *et al.*, 2007).

O outro método de virtualização é a emulação de *hardware*. Esta técnica consiste na simulação de um *hardware* para cada máquina virtual no sistema hospedeiro. Uma das vantagens desta técnica é a capacidade de emulação de qualquer arquitetura em uma outra (JONES, 2006). Porém, como todas as instruções

são emuladas, existe uma perda de desempenho muito alta, característica esta que contrasta com a definição de (POPEK; GOLDBERG, 1974), já discutida anteriormente, em que uma máquina virtual deve simular apenas uma pequena parte das instruções, sendo as outras executadas diretamente pelo *hardware* da máquina.

De uma forma simplificada, a máquina virtual, da forma como será abordada neste trabalho, é uma simulação de um computador físico em um computador real. O monitor de máquinas virtuais é o *software* executado sobre a máquina física e é o responsável por criar e gerenciar as máquinas virtuais. O VMM (*Virtual Machine Monitor* - Monitor de Máquinas Virtuais) divide os recursos do *hardware* físico de acordo com a configuração de cada máquina virtual. 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.

A Figura 2.6 ilustra uma máquina virtual sendo executada no VirtualBox.

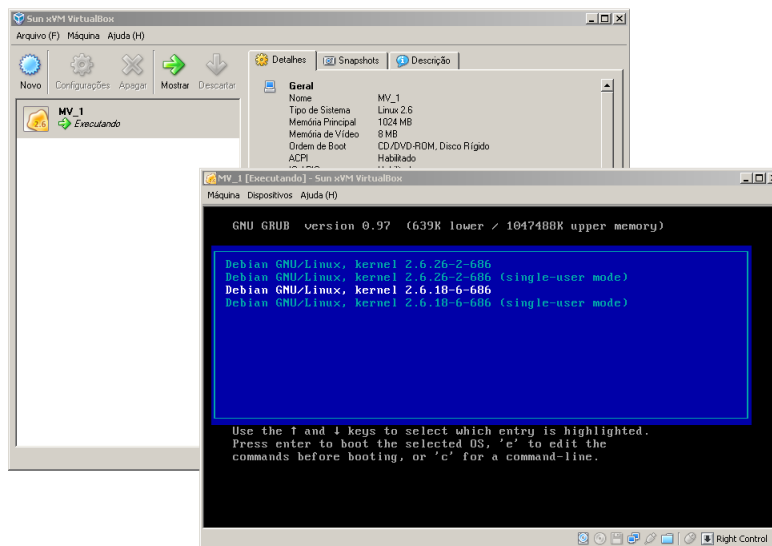


Figura 2.6: Máquina Virtual iniciada no VirtualBox

2.2 Histórico

O desenvolvimento dos sistemas de tempo compartilhado permitiu a criação de sistemas nos quais mais de uma tarefa pudesse ser executada em um único computador “simultaneamente”. Como consequência dos sistemas de tempo compartilhado mais de um usuário poderia ter acesso ao computador, o que encorajou um número maior de pessoas a utilizarem o computador (MEYER; SEAWRIGHT, 1970).

Com isto, o interesse no desenvolvimento de técnicas que aprimorassem a utilização desta nova característica dos computadores aumentou. Uma das técnicas desenvolvidas foi a virtualização e, além de outras vantagens, permitiu que cada usuário do computador selecionasse qual sistema operacional gostaria de utilizar (MEYER; SEAWRIGHT, 1970).

O primeiro sistema a implementar a virtualização de forma funcional foi o CP-40, um hipervisor desenvolvido para o computador IBM *System/360 Model 40*. Neste sistema o *software* necessário para o estabelecimento das máquinas virtuais era composto de duas partes independentes: o *Virtual Machine Control Program* (CP) e o *Cambridge Monitor System* (CMS) (MEYER; SEAWRIGHT, 1970), (PARMELEE *et al.*, 1972), (HENDRICKS; HARTMANN, 1979). O sistema CP/CMS foi concebido em 1964 como um sistema de tempo compartilhado de segunda geração (CREASY, 1981). O *Control Program* simulava várias cópias do computador e era o responsável por gerenciar o alocamento dos recursos para cada cópia, chamada de máquina virtual, as quais poderiam executar o seu próprio sistema operacional independente uma das outras (CREASY, 1981), (MEYER; SEAWRIGHT, 1970).

O *Cambridge Monitor System* era o sistema mono-usuário que provia as funções necessárias para o usuário operar o computador através de um console de texto. O CMS era capaz de funcionar independente do CP, sendo então utilizado como o sistema operacional das máquinas virtuais. O CMS era executado no IBM *System/360 Model 40* antes mesmo do CP ter sido desenvolvido (MEYER; SEAWRIGHT, 1970).

O CP-40 que foi desenvolvido em 1965-66 para o IBM *System/360 Model 40* era capaz de executar simultaneamente até 15 máquinas virtuais da família IBM *System/360*, porém não era capaz de simular sua própria arquitetura. (PARMELEE *et al.*, 1972). Este sistema foi um protótipo para o CP-67 que foi desenvolvido em 1969.

O CP-67 foi desenvolvido para o IBM *System/360 Model 67* e era um sistema multi-usuário que, como o CP-47, tinha a capacidade de simular a família IBM *System/360*, porém tinha a vantagem que podia simular sua própria arquitetura (PARMELEE *et al.*, 1972), (DOHERTY; KELISKY, 1979). Ele era capaz de endereçar 16 milhões de bytes de memória virtual para cada usuário, fornecendo um espaço maior de memória do que a memória real do *Model 67* (MEYER; SEAWRIGHT, 1970). Sem este aprimoramento, a soma da memória de todas as máquinas virtuais não poderia ultrapassar a memória real do *Model 67*. Para expandir a memória disponível para as máquinas virtuais era possível a utilização de um *storage* virtual (PARMELEE *et al.*, 1972).

Após o CP-67, foi lançado, em 1972, o *Virtual Machine Facility/370 (VM/370)*, o sistema de máquinas virtuais para o IBM *System/370*. Além de ter como componentes o CP e o CMS, foi incorporado também o *Remote Spooling Communications Subsystem (RSCS)*. Este componente é um sistema operacional de propósito específico, implementado para ser executado em um ambiente virtual (máquina virtual) com a finalidade de controlar a transmissão de dados entre as máquinas virtuais do sistema (MACKINNON, 1979), (HENDRICKS; HARTMANN, 1979), (CREASY, 1981). A impressão que os usuários tinham ao utilizar o sistema era de que o estavam usando sozinhos (ATTANASIO; MARKSTEIN; PHILLIPS, 1976).

Apesar da IBM ter sido a pioneira na área de virtualização tendo a grande maioria dos desenvolvimentos terem sido feitas por ela, existiram também máquinas virtuais para computadores de outras equipes. Em 1975, na UCLA (*Universidade of Califórnia*, Los Angeles), estava sendo desenvolvido um protótipo de uma máquina virtual para o PDP 11/45, um equipamento da DEC (*Digital Equipment*

Corporation). Neste sistema seria necessária a modificação da arquitetura do PDP 11/45 para poder suportar monitor de máquinas virtuais (POPEK; KLINE, 1975).

Conforme visto, a virtualização surgiu com a necessidade de aproveitamento dos recursos dos sistemas multi-usuários de tempo compartilhando. A evolução dos sistemas de virtualização está intimamente ligado ao desenvolvimento dos sistemas de máquinas virtuais CP/CMS e o VM/370 da IBM.

2.3 Aplicações

Nos primórdios da virtualização, sua principal utilização era no aumento do nível de compartilhamento e na melhor utilização dos recursos computacionais dos *mainframes* (SOUZA, 2006), (MEYER; SEAWRIGHT, 1970). A partir dos anos 90, os micro-computadores começaram a ser utilizados por um número maior de pessoas e estavam sendo conectados uns aos outros através de redes, gerando novas questões, como segurança, consumo de energia e confiabilidade (SOUZA, 2006).

Desde então, a virtualização vem se apresentando como uma grande alternativa para a solução destas questões, não somente para empresas de grande porte, mas também para as empresas de pequeno porte e para os usuários comuns. Algumas características da virtualização que justificam este fato são:

Isolamento: Qualquer aplicação executada dentro de uma máquina virtual não pode afetar a máquina física e nem as outras estações virtuais.

Portabilidade: O *hardware* virtual é gerado pelo monitor de máquinas virtuais, portanto suas características são definidas por ele e não pelo *hardware* físico. Se este for alterado, a máquina virtual não irá sofrer conseqüências, salvo as relacionadas ao desempenho.

Eficiência: Como a grande maioria das instruções é executada diretamente no *hardware* físico, a perda de desempenho é bem menor do que no caso dos emuladores e simuladores tradicionais, por estes tratarem todas as instruções antes de as executarem no *hardware* real.

Múltiplas instâncias: O monitor de máquinas virtuais é capaz de executar diversas instâncias de máquinas virtuais simultaneamente, permitindo a execução de sistemas operacionais diferentes em um mesmo computador.

Devido a essas características, a virtualização pode ser aplicada em diversos problemas, por permitir a criação de ambientes seguros e portáteis. Um exemplo disto é a utilização de uma máquina virtual para permitir que determinadas aplicações compiladas para uma arquitetura específica sejam executadas em outras (SOUZA, 2006).

A característica do isolamento proporcionada pela virtualização é muito utilizada em sistemas seguros e ambientes de testes. Em relação à segurança de sistemas, é possível utilizar máquinas virtuais para isolar um ambiente de outro, evitando que problemas em um dos ambientes comprometam os outros, como no caso do uso de aplicações não seguras (WLODARZ, 2007), (SOUZA, 2006). No desenvolvimento de anti-vírus e anti-*spyware*, as empresas criam máquinas virtuais sem segurança conectadas a internet para descobrir novas ameaças na rede (SOUZA, 2006).

Ainda com relação à segurança, a máquina virtual pode ser empregada para melhorar a segurança de um sistema contra ataques a seus serviços. Quando um sistema é invadido, a obtenção de dados confiáveis se torna um problema. Em (LAUREANO; MAZIERO; JAMHOUR, 2003) é apresentada uma proposta para aumentar a confiabilidade deste tipo de sistema, aplicando técnicas de detecção de intrusão em máquinas virtuais para detectar e combater ataques.

A virtualização também tem sido utilizada na educação, em cursos como banco de dados, segurança computacional e redes de computadores que exigem que os alunos tenham acesso irrestrito ao sistema e também possam utilizar uma rede para realizar seus testes (BULLERS; BURD; SEAZZU, 2006), (WLODARZ, 2007), (HU; MEINEL; SCHMITT, 2004). Isto é inviável em ambientes institucionais por serem atividades passíveis de danificar o sistema e as outras máquinas conectadas à rede. Com a virtualização é possível fornecer aos estudantes um ambiente virtual ao qual os estudantes tenham acesso administrativo sem prejuízos a ambientes exter-

nos. Um ambiente tutor que disponibiliza máquinas virtuais para os estudantes é descrito em (HU; MEINEL; SCHMITT, 2004).

As máquinas virtuais podem ser utilizadas para a criação de ambientes de testes, permitindo a criação de diferentes cenários de testes, como modificação da arquitetura simulada ou a inserção de limitações na máquina virtual (SOUZA, 2006); (WLODARZ, 2007). É possível inclusive utilizar máquinas virtuais que armazenam os estados de execução anteriores, permitindo que estes possam ser restaurados posteriormente (KING; DUNLAP; CHEN, 2005).

A virtualização também é utilizada no balanceamento de cargas de servidores. Dividindo os serviços dos servidores em máquinas virtuais, é possível utilizar a migração para controlar a carga entre dois servidores. À medida que um servidor se torna sobrecarregado, basta migrar uma máquina virtual que executa algum serviço para outro servidor, sem que o serviço fique fora de funcionamento por algum tempo.

A virtualização pode ter um impacto muito significativo nas empresas desde que implantada de maneira correta, colaborando para o aumento das vantagens competitivas das empresas (ALKMIM, 2009). Uma das grandes vantagens é a redução de custos com energia, aquisição de equipamentos e ainda com a manutenção de equipamentos. É importante ressaltar também, que a virtualização pode trazer ganhos relacionados à confiabilidade dos serviços. Além disto, a implantação do sistema pode ser feita a um baixo custo, pois existem ferramentas livres que possuem um desempenho muito bom e que podem ser implantadas em vários sistemas operacionais.

Muitas empresas utilizam serviços na área de TI que não podem ser interrompidos, como um servidor de e-mail ou um servidor de internet. Desta forma, são necessários servidores que permaneçam ligados em tempo integral e que sejam responsáveis por fornecer um determinado serviço. Geralmente, estes serviços não esgotam a capacidade do equipamento, fazendo com que este possua recursos ociosos, como capacidade de processamento e memória não utilizada.

Em contrapartida, estas mesmas empresas precisam manter outros servidores ligados para garantir que outros serviços permaneçam acessíveis, serviços estes que por algum motivo, como por exemplo, incompatibilidade ou segurança, não possam ser executados em uma mesma máquina que outro. Desta forma, em uma mesma empresa podem existir vários computadores com recursos ociosos, o que gera um alto consumo de energia e um alto custo com a manutenção dos equipamentos devido ao número de máquinas utilizadas.

Uma solução alternativa para este problema é utilizar a virtualização, colocando os serviços que não possam ser executados em um mesmo computador em máquinas virtuais diferentes, unindo assim vários servidores em um único, o que é chamado de consolidação de servidores (SOUZA, 2006). Ao invés de utilizar vários equipamentos cada um executando um sistema operacional diferente, utiliza-se um único equipamento com várias máquinas virtuais, cada uma executando seu próprio sistema operacional e suas próprias aplicações (LAUREANO; MAZIERO; JAMHOUR, 2003) (WLODARZ, 2007). Desta forma, os recursos ociosos são aproveitados e o número de computadores ligados é reduzido, o que reduz custos com energia, simplifica processo de manutenção e ainda pode reduzir custos com a aquisição de equipamentos (KING; DUNLAP; CHEN, 2005).

Da mesma forma como é feita a consolidação de servidores, pode ser feita a consolidação de aplicações. A consolidação de aplicações permite instalar, em um mesmo servidor, aplicações que necessitariam de um novo hardware ou sistema operacional para serem instaladas (NANDA; CHIUEH, 2005). Neste caso, o novo hardware seria oferecido pela máquina virtual. Como exemplo, é possível, utilizando máquinas virtuais, que aplicações que necessitem ser executadas em ambiente Linux, sejam executadas em um mesmo computador que exija um ambiente Windows.

A virtualização tem sido utilizada também em sistemas embarcados. Em (SU *et al.*, 2009) é apresentado o SmartVisor, uma plataforma de virtualização para sistemas embarcados. A avaliação mostrou que o desempenho da máquina hóspede utilizando o WindowsXP foi próxima a do ambiente nativo.

2.4 Ferramentas de Virtualização

O Xen (XEN, 2009), desenvolvido pela Universidade de Cambridge, é um monitor de máquinas virtuais para a arquitetura x86, que utiliza a técnica de paravirtualização (URSCHEI *et al.*, 2007), (ROSE, 2004), (BARHAM *et al.*, 2003), (HUANG *et al.*, 2007). Em *hardware* com suporte a virtualização, é possível executar o sistema convidado sem a necessidade de que este seja modificado, porém é necessário o suporte adicional de *driver* dentro do próprio Xen (XEN-MAN, 2005). O *overhead* imposto pela presença do Xen é estatisticamente insignificante comparado com ao sistema sem a utilização do Xen (YOUSEFF *et al.*, 2006), sendo portanto viável sua implantação em servidores.

O QEMU (*open source processor emulator*) é uma ferramenta de código aberto que pode funcionar como um emulador ou como um virtualizador (QEMU, 2009). Quando utilizado como um emulador ele permite que aplicativos que foram feitos para executar em um determinado sistema operacional sejam executados em outro. Apesar do processo de emulação geralmente impor um grande *overhead* no sistema, o QEMU possui um desempenho muito bom por utilizar uma técnica chamada de tradução dinâmica. Sendo utilizado como um virtualizador, o QEMU atinge um nível de desempenho muito bom ao utilizar o *driver* KQEMU (QEMU *Accelerator*), pois este permite que as instruções da máquina virtual sejam executadas diretamente sobre o processador do computador físico ao invés de serem emuladas.

O VirtualBox (VIRTUALBOX, 2009) é uma ferramenta de código aberto (*GNU General Public License*) que permite a virtualização em arquiteturas x86. Ele foi desenvolvido para ser utilizado em *desktop*, servidores ou embarcados. O VirtualBox pode ser utilizado em ambientes Linux, Windows, OpenSolaris e Macintosh. Além disto, a ferramenta possui suporte a pastas compartilhadas, uma característica que permite que pastas da máquina hospedeira sejam compartilhadas com a máquina hóspede, facilitando a troca de arquivos entre elas.

Como o Xen é uma solução livre para a virtualização que permite a *live migration* e vem sendo amplamente utilizado em ambientes computacionais, ele foi

adotado como o monitor de máquinas virtuais neste trabalho e será descrito com mais detalhes na seção seguinte.

2.5 Xen

O Xen, desenvolvido pela Universidade de Cambridge, é um monitor de máquinas virtuais para a arquitetura x86, que utiliza a técnica de paravirtualização (URSCHEI *et al.*, 2007), (ROSE, 2004), (BARHAM *et al.*, 2003), (HUANG *et al.*, 2007). Ele foi desenvolvido inicialmente para dois tipos de sistemas específicos:

- Centro de computação organizacional, no qual um grande cluster deve ser compartilhado entre unidades administrativas.
- Sevidores de alcance global, ou seja, um único sistema pode oferecer vários serviços remotamente para usuários de outras regiões.

Como o hipervisor se comporta como uma camada a mais entre as máquinas virtuais e os dispositivos de entrada e saída (URSCHEI *et al.*, 2007), é fundamental a presença de um mecanismo que realize a transferência de dados com um baixo *overhead* (BARHAM *et al.*, 2003). 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. Em *hardware* com suporte a virtualização, é possível executar o sistema convidado sem a necessidade de que este seja modificado, porém é necessário o suporte adicional de *driver* dentro do próprio Xen (XEN-MAN, 2005). Desta forma, ambientes convidados que não podem ser modificados, como o Windows, podem ser virtualizados 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.

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. Um outro trabalho interessante é o feito por (WANG *et al.*, 2008), que realizou a migração de roteadores virtuais.

(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.

No Xen, o sistema operacional que possui o monitor de máquinas virtuais é comumente chamado de Dom0 (Domínio 0) ou máquina hospedeira e as máquinas virtuais são chamadas de Dom1, DomU ou máquina hóspede.

Em relação à memória, o Xen é responsável por gerenciar a alocação de memória física para as máquinas virtuais e por garantir o uso seguro destas, evitando, por exemplo, que uma máquina virtual acesse a região de memória de outra. Uma parte da memória física é usada para o uso do hipervisor. O hipervisor aloca uma determinada quantidade de memória para uma máquina virtual e esta não pode usar além deste espaço determinado. Além da região de memória utilizada pelo Xen, uma parte da memória alocada para cada máquina virtual também é reservada para uso do hipervisor. A quantidade de memória pode ser ajustada dinamicamente através de um *balloon driver* (*driver* balão) (XEN-DOC, 2009).

Além disto, é importante observar que a alocação de memória é feita no nível de páginas, fazendo com que não haja nenhuma garantia de que as máquinas virtuais possuam todas as suas páginas reservadas em uma mesma região de memória.

Entretanto, alguns sistemas operacionais não oferecem um bom suporte para trabalhar com a memória física fragmentada. Para contornar esta situação, o Xen utiliza uma abstração que permite à máquina virtual enxergar sua memória física como sendo não fragmentada. Isto é feito utilizando uma pseudo-memória física. O hipervisor Xen possui uma tabela que mapeia a memória física da máquina real para cada máquina virtual. Somado a isto, cada máquina virtual possui uma tabela que mapeia a memória disponível na máquina virtual com a memória física da máquina real (XEN-MAN, 2005).

Além destas características existem diversas outras que podem ser encontradas de forma mais detalhada na documentação do Xen (XEN-DOC, 2009), (XEN-MAN, 2005).

2.6 *Live Migration*

A migração de máquinas virtuais transfere uma máquina virtual de uma máquina física para outra e isto deve ser feito de forma transparente para o sistema operacional hospede e para os clientes remotos da máquina virtual (NELSON; LIM; HUTCHINS, 2005), (CLARK; FRASER; HAND, 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; LIM; HUTCHINS, 2005). A migração pode ser feita de duas formas: regular ou *live migration* (XEN-MAN, 2005). A diferença entre as duas é que na primeira forma a máquina virtual é pausada durante todo o processo de migração enquanto que na segunda forma ela não é, do ponto de vista do usuário, pausada durante a migração. Como o objetivo deste trabalho é a migração *online* (*live migration*) com o xen (XEN, 2009), esta seção é voltada para este tipo de migração utilizando o hipervisor xen. A Figura 2.7 ilustra o processo de migração de máquinas virtuais com o Xen.

Para que a migração seja realizada, devem ser considerados três estados que são o estado dos dispositivos virtuais, as conexões externas com os dispositivos e

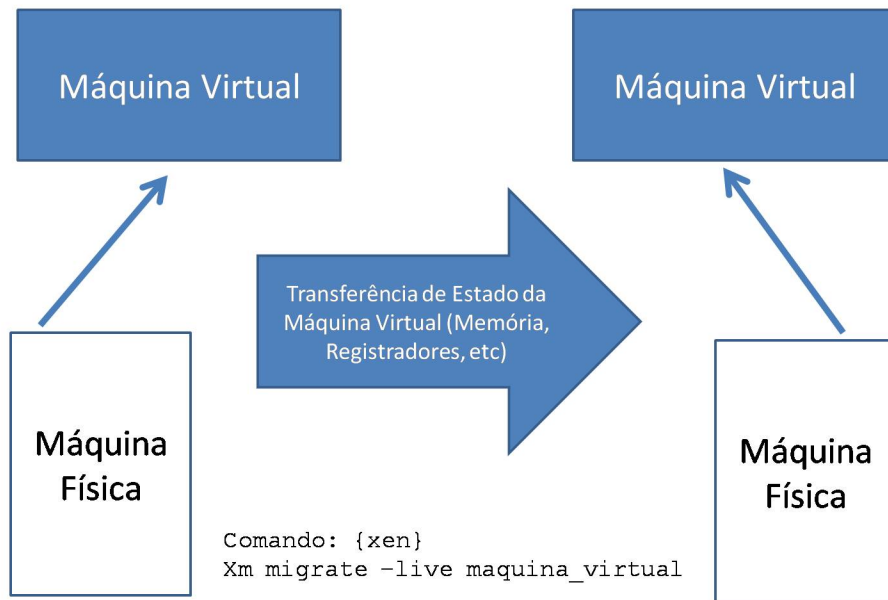


Figura 2.7: Migração de máquinas virtuais com o Xen

a memória física da máquina virtual (NELSON; LIM; HUTCHINS, 2005). Todos estes estados devem permanecer o mesmo após a migração. A cópia de toda a memória RAM da máquina virtual enquanto ela estivesse pausada levaria a um período de inacessibilidade muito grande. Portanto, uma parte da memória RAM deve ser migrada enquanto ela ainda estiver em execução e somente as partes modificadas devem ser migradas enquanto ela estiver pausada (NELSON; LIM; HUTCHINS, 2005), (CLARK; FRASER; HAND, 2005), (BRADFORD *et al.*, 2007). Uma outra característica importante na migração das máquinas virtuais é a consistência do sistema de arquivos da máquina virtual (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.

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 pré-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 (*Local Area Network*). 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 no próximo Capítulo.

2.7 O Protocolo iSCSI (Internet SCSI)

Para que seja possível a migração, é necessária 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 disso, existe a limitação física imposta pela utilização de cabos especiais, diferentes dos 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 seja exportada para um outro computador, simulando assim um dispositivo *storage*. Este *storage* simulado é chamado de *storage* virtual. Esta solução alternativa utiliza o protocolo iSCSI para fazer a comunicação entre o dispositivo *storage* virtual e os computadores físicos.

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 desnecessária 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 cliente e os processa. O SCSI *target* mapea 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*.

Capítulo 3

Metodologia

3.1 O trabalho Proposto

O objetivo deste trabalho é o desenvolvimento uma solução que permita realizar a migração *online* de máquinas virtuais com eficiência, confiabilidade e baixo custo. Para garantir o baixo custo da solução foi utilizado um *storage* virtual ao invés de um *storage* físico que tem um custo elevado.

A Figura 3.1 ilustra a utilização do *storage* virtual na solução proposta. O *storage* virtual é um computador que irá exportar uma partição para as máquinas que estão executando o Xen, permitindo que estas iniciem uma máquina virtual utilizando a mesma partição como disco para a máquina virtual. Como os dois servidores Xen têm acesso a mesma partição, quando a migração acontecer o disco da máquina virtual será o mesmo antes e depois da migração. Neste trabalho foram adotadas tecnologias e ferramentas livres e que são utilizadas no mercado atualmente. As ferramentas utilizadas foram:

Xen (XEN, 2009): O Xen é o hipervisor que foi utilizado para a criação e o gerenciamento das máquinas virtuais. Ele foi escolhido por ser o único hipervisor popular, que seja livre e que possua suporte para a *live migration*. Além

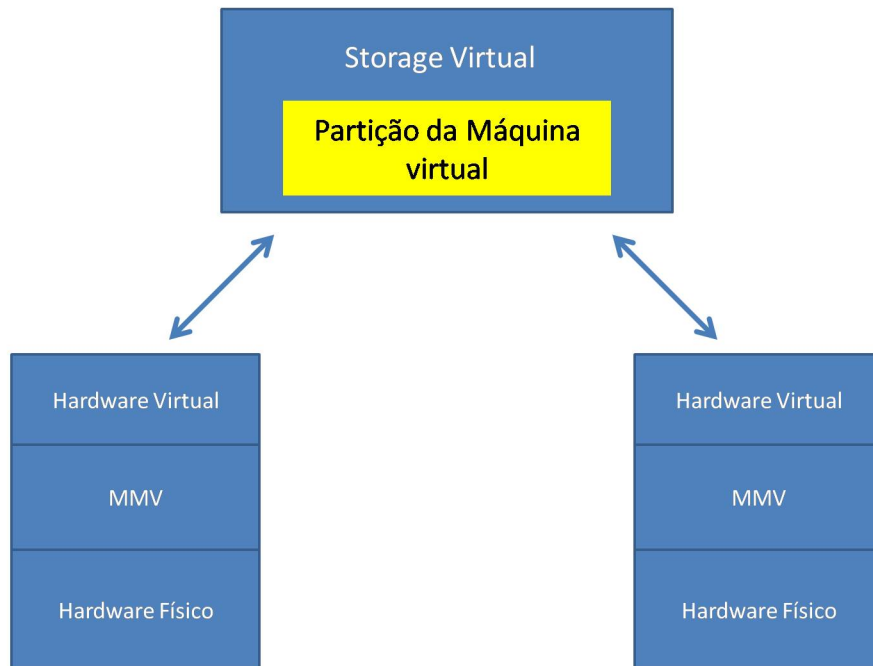


Figura 3.1: O *storage* virtual

disto, o *overhead* imposto pela presença do Xen é muito pequeno (YOUSEFF *et al.*, 2006).

iSCSI target enterprise (IET) (IET, 2009): O IET é a ferramenta iSCSI *target* responsável por exportar o dispositivo que será utilizado como disco na máquina virtual. Ela foi escolhida por ser uma ferramenta livre disponível nas distribuições linux.

Open-iSCSI (OPENISCSI, 2009): O Open-iSCSI é a ferramenta iSCSI *initiator* e foi instalada nas máquinas físicas entre as quais foi feita a migração. Ela importa o dispositivo exportado pelo IET para que o xen possa iniciar a máquina virtual utilizando este dispositivo como disco. O Open-iSCSI foi escolhido.

Neste trabalho foram feitas duas análises de desempenho. A primeira análise avaliou o desempenho de uma máquina virtual que tinha seu sistema de arquivos localizado em um *storage* virtual. O resultado foi comparado com o desempenho da mesma máquina virtual utilizando um sistema de arquivos localizado na própria máquina hospedeira. Esta análise de desempenho é importante uma vez que na solução proposta a máquina virtual não possui um disco local. A segunda análise avaliou a perda de desempenho de uma máquina virtual durante uma migração *live migration* comparado com a mesma máquina virtual antes de iniciar a migração. Nesta última análise em nenhuma situação a máquina virtual possui um disco local.

3.2 Ambiente Computacional

Para avaliar o desempenho da migração de máquinas virtuais foi montado um sistema de computação utilizando três máquinas reais conforme é mostrado na Figura 3.2. Ela representa as ligações físicas entre as máquinas. As máquinas Xenserver1 e Xenserver2 possuem o Debian Linux como sistema operacional, e Xen (XEN, 2009) e o *Open-iSCSI* (OPENISCSI, 2009) instalados. A máquina iSCSIserver possui o Gentoo Linux como sistema operacional e o iSCSI *Enterprise Target* (IET) instalado. O esquema da configuração lógica das máquinas, incluindo a máquina virtual é mostrada na figura 3.3.

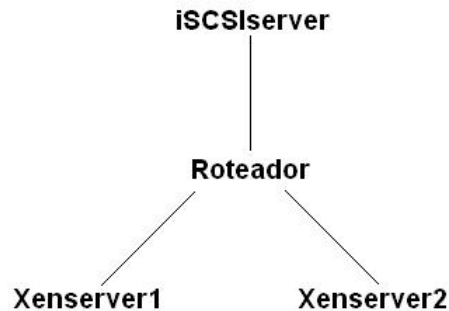


Figura 3.2: Ligação física do sistema de computação

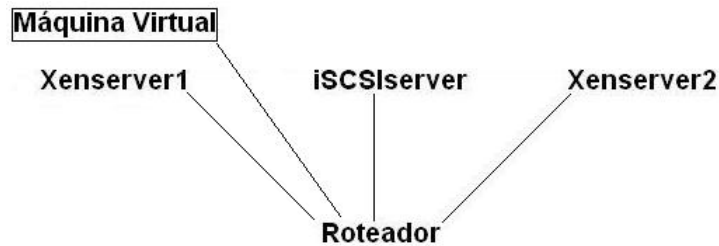


Figura 3.3: Ligação lógica do sistema de computação

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 *hardware* e sistemas operacionais diferentes, o desempenho antes da migração foi diferente do desempenho depois da migração. Este fato não irá influenciar nos resultados deste trabalho, uma vez que este tem como objetivo analisar o desempenho apenas durante a migração.

Com o IET é possível exportar uma partição de um computador para um outro. Desta forma será possível iniciar a máquina virtual tendo como disco desta a partição exportada. O *Open-iSCSI* é uma ferramenta iSCSI *initiator* que se conecta ao iSCSI *target* para receber o disco exportado. Utilizando estas duas ferramentas é possível fazer a máquina hospedeira enxergar um disco vindo de um outro computador como se ele fosse um disco local e assim executar a máquina virtual o utilizando.

As duas máquinas virtuais têm acesso a partição da máquina virtual, o que permite que não seja necessário copiar todo sistema de arquivos da máquina virtual, apenas o seu estado é copiado entre as duas máquinas físicas. Durante o período em que é copiado o estado, é necessário que a máquina virtual seja congelada, porém, este tempo de congelamento pode ser pequeno o suficiente para que o usuário não perceba (CLARK, 2005; NELSON 2005).

O roteador utilizado que estava funcionando como um *switch* 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.

A configuração dos computadores físicos utilizados na avaliação de desempenho da migração de máquinas virtuais foram as seguintes:

- 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 100Mbps 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 100Mbps Realtek

- iSCSIserver
 - Processador: Pentium 4 3.06 GHz
 - Memória RAM: DDR 400MHz 512MB
 - Disco Rígido: HD pata 7200rpm 80 GB.
 - Placa de Rede Ethernet 100Mbps Realtek

Conforme dito na Seção 3.1, além da análise de desempenho da máquina virtual durante o período de migração foi analisado o desempenho de uma máquina

virtual que possuía o sistema de arquivos proveniente de um *storage* virtual. O ambiente computacional utilizado para esta análise foi o mesmo do utilizado na migração da máquina virtual. A única diferença é que não foi utilizada a máquina Xenserver2, uma vez que não era necessário duas máquinas utilizando o serviço Xen. A configuração das máquinas físicas foi a mesma utilizada na migração de máquinas virtual. Em ambas as análises a máquina virtual foi executada com duas CPUs virtuais e 1GB de memória RAM. Foram utilizadas duas CPUs virtuais porque uma das máquinas era *dual core*.

3.3 *Benchmarks*

Nas duas análises de desempenho realizada foram analisados quatro fatores importantes de um sistema: o processamento, a memória, a rede e o acesso ao disco. Para avaliar o desempenho da rede foi utilizado o comando ping do Linux e analisados a latência e se houve alguma perda de pacote.

Para avaliar o acesso ao disco foi utilizada a ferramenta de *benchmark* dbench (DBENCH, 2009). 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 devido a uma das máquinas serem *dual core*. Se fosse utilizado apenas um cliente um dos núcleos ficaria ocioso. Com objetivo de simular uma utilização real, o disco é "aquecido" durante 80 segundos pela própria ferramenta.

Para avaliar a memória e o processamento da máquina virtual foi utilizada a ferramenta de *benchmark* SysBench (SYSBENCH, 2009). Esta ferramenta é voltada 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. 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 valor padrão da ferramenta. O SysBench foi executado utilizando 20 *threads*, valor obtido experi-

mentalmente para obter um melhor desempenho. O teste de memória do SysBench consiste em realizar leituras ou escritas sequenciais na memória.

Capítulo 4

Resultados e Análises

4.1 Desempenho do iSCSI Exportando o Disco da Máquina Virtual

Para avaliar o desempenho de um *storage* virtual exportando um disco para a máquina virtual através do protocolo iSCSI foram analisados quatro aspectos: processamento, memória, disco e rede. Conforme mencionado no Capítulo anterior, estes aspectos foram avaliados utilizando os *benchmarks* dbench (DBENCH, 2009), SysBench (SYSBENCH, 2009) e o comando ping.

Os testes foram realizados em uma máquina virtual iniciada em um hospedeiro cuja configuração está descrita no Capítulo anterior. Os resultados obtidos são mostrados na Figura 4.1.

Analisando a Figura 4.1, percebe-se que o processamento e a memória não tiveram uma redução considerável no desempenho. Os valores são menores que 1%. Isto ocorre porque as informações são processadas no *hardware* da máquina hospedeira sem a necessidade de ocorrer uma transferência considerável de dados via rede. O mesmo ocorre com a memória. A memória RAM está localizada no *hardware* da máquina hospedeira, não havendo a necessidade de transferência de dados via rede.

Teste	Desempenho com disco local	Desempenho com <i>storage</i> virtual	Porcentagem de Desempenho	Influência
Disco (Transferência) MB/s	36,0246	18,9599	- 47,3695%	Alta
Disco (latência) Milissegundos	154,092	207,913	+ 34,9278%	Considerável
Memória MB/s	271,73	270,66	- 0,3938%	Desprezível
Processamento Eventos	9952,45	9899,59	- 0,5311%	Desprezível
Rede segundos	0,1109	0,1867	+ 68,3497%	Alta

Figura 4.1: Resultados Obtidos

Os fatores analisados que mostraram uma redução de desempenho mais considerável foram os que envolviam uma maior transferência de dados via rede. Como a rede utilizada foi de 100 Mbits, a transferência de dados via rede ficou muito limitada. A taxa de transferência de disco foi reduzida em 47%, um valor que pode comprometer alguns sistemas que necessitem de um acesso a disco alto. Ocorreu um aumento na latência do disco e da rede que foram respectivamente de 35% e de 68%. Ambos valores consideravelmente altos. Não houve nenhuma perda de pacotes.

Uma possível solução para este problema é a utilização de uma rede ethernet Gigabit ao invés de 100Mbits. Caso a taxa de transferência máxima dos discos seja muito alta, é possível ainda utilizar duas ou mais placas de rede para suprir este problema.

No teste de desempenho com o *storage* virtual, a taxa de transferência do disco apresentou o valor obtido, 18,95 MB/s, foi superior a capacidade máxima de transferência da rede, 12,5 MB/s. Uma possível explicação para este fato é a utilização de algum tipo de *cache* ou *buffer* durante o teste pela ferramenta de *benchmark* ou pelo sistema operacional.

4.2 Desempenho do iSCSI na Migração de Máquinas Virtuais

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 a 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 4.2 e 4.3. A Figura 4.2 mostra a taxa de transferência do sistema de arquivos por segundo e a Figura 4.3 mostra a latência do sistema de arquivos. As figuras não mostram o período de aquecimento.

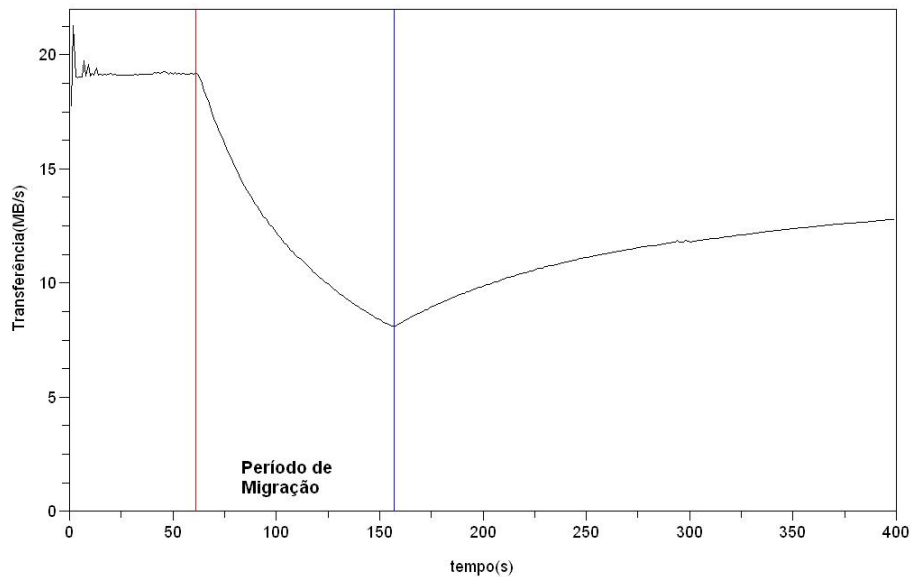


Figura 4.2: Execução do dbench (taxa de transferência)

A partir das figuras 4.2 e 4.3, 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. Na figura 4.3, no tempo próximo ao 300 segundos, é possível perceber um aumento da latência, porém este fato não influencia neste trabalho, uma vez que o interesse deste trabalho é avaliar o desempenho da máquina virtual durante o período com o período antes da migração. Uma possível explicação para este aumento inesperado é a ocorrência de uma interferência qualquer na rede.

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 4.4. A redução de processamento foi de aproximadamente 16% durante a migração, um valor considerável, porém para aplicações normais, não causaria problemas.

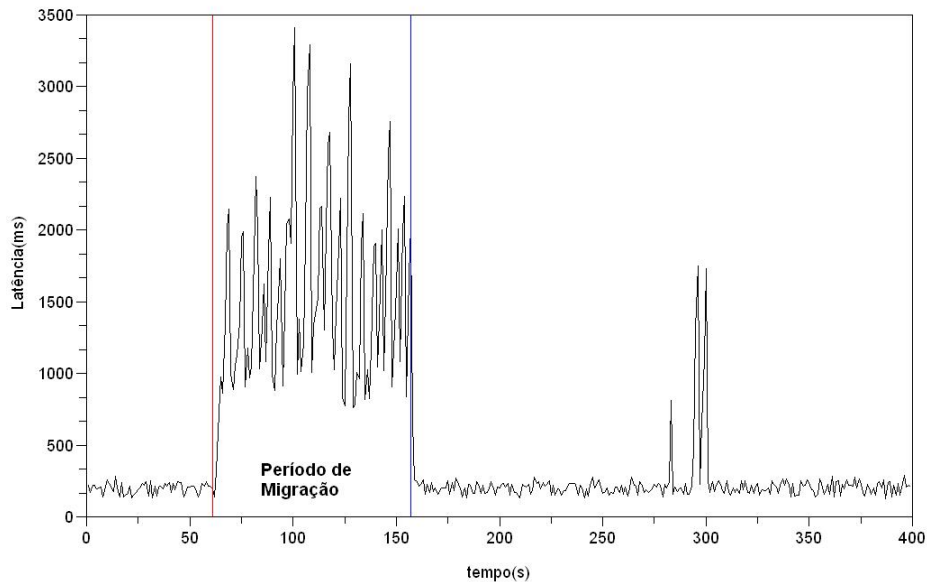


Figura 4.3: Execução do dbench (latência)

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 4.5. A redução da taxa de transferência foi imperceptível.

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 4.6. 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 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 4.7, 4.8, 4.9 e 4.10. É interessante observar que em nenhum momento a máquina virtual ficou fora do ar, mesmo com o período de sobrecarga. 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 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 4.11. Após a migração a máquina virtual ficou *offline* durante 29 segundos. Uma possível causa para a máquina virtual ter ficado inacessível após a migração ao invés de durante a migração é a troca dos endereços MAC das placas físicas, sendo necessário que o roteador precisa atualizar a tabela de endereços.

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 de um algoritmo, criado por nossa equipe, que realizava uma escrita

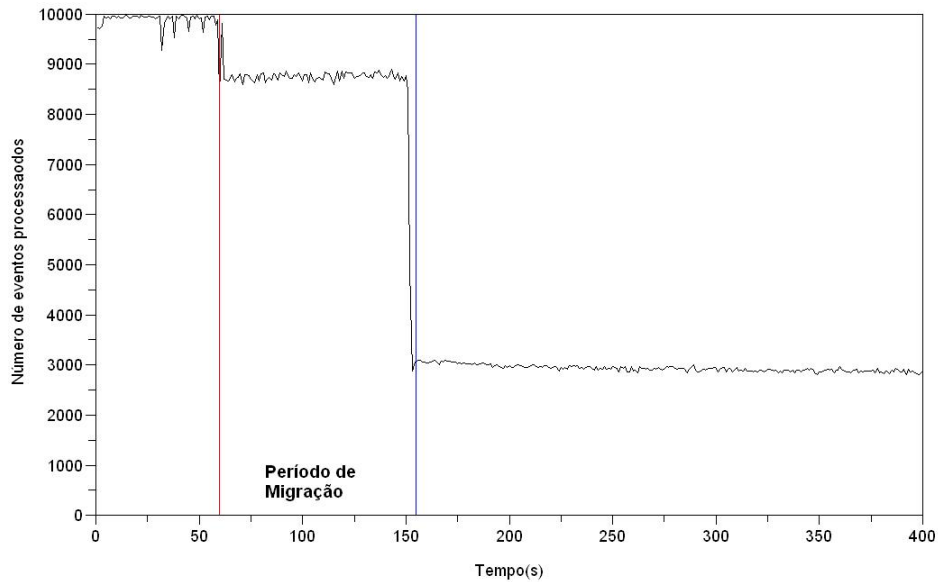


Figura 4.4: Execução do SysBench (processamento)

intensa na memória RAM 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. Para avaliar o desempenho foi executado um comando ping na máquina virtual. Os resultados são mostrados na Figura 4.12. Um teste semelhante a este foi efetuado no trabalho de (CLARK; FRASER; HAND, 2005) que obteve como resultado um *downtime* na máquina virtual de 3,5 segundos.

A partir da Figura 4.12 é possível perceber que a máquina virtual ficou inacessível 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.

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 este estava limitado a largura da banda. Durante a migração,

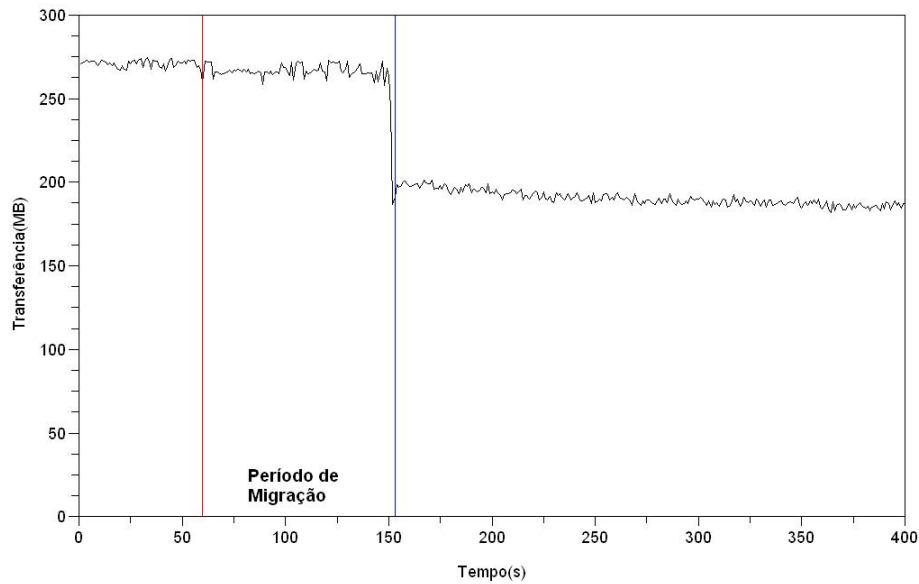


Figura 4.5: Execução do SysBench (memória)

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 1 Gbits/s ou de 10 Gbits/s é recomendado. Neste trabalho não utilizamos redes mais valozes por não serem encontradas com tanta frequência como as redes 100 Mbits/s, colaborando assim com o baixo custo da solução, uma vez que não é necessário adquirir novos equipamentos.

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

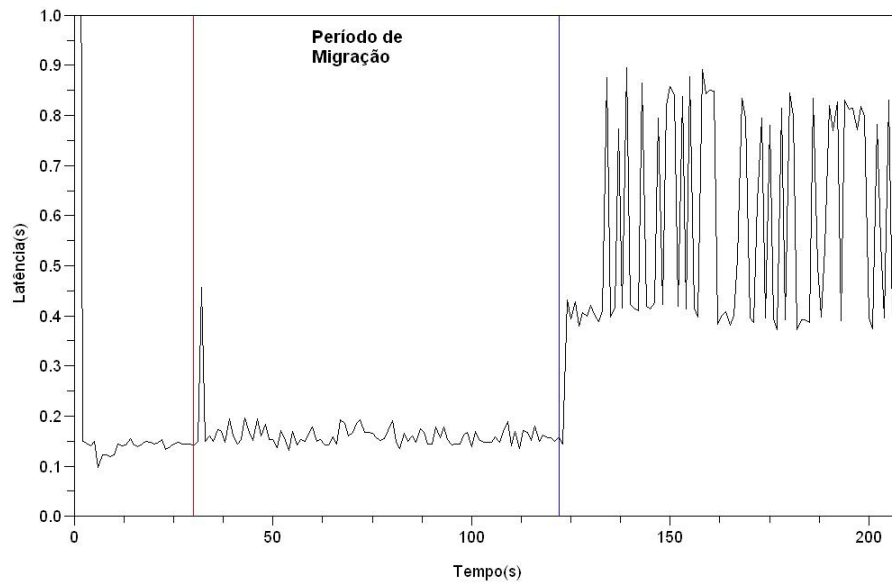


Figura 4.6: Execução do comando ping

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.

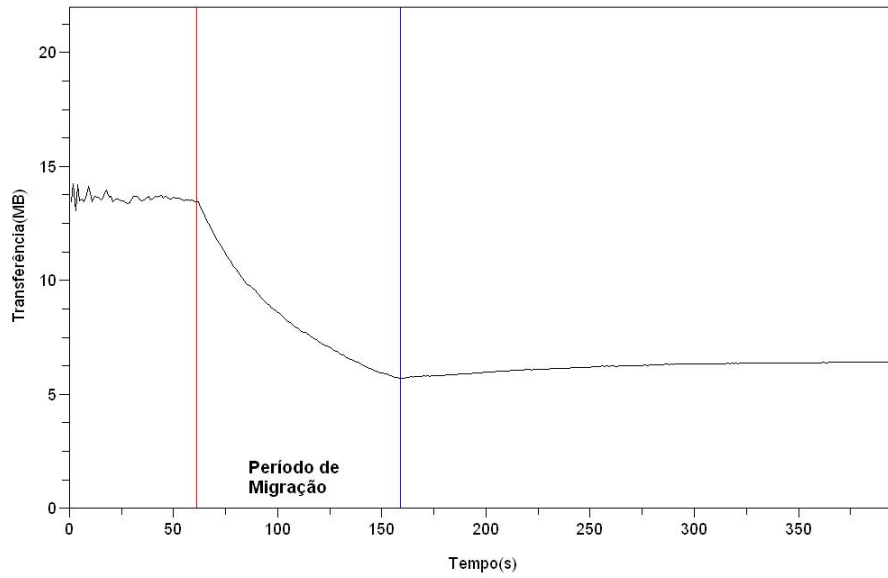


Figura 4.7: Testes simultâneos: execução do dbench (taxa de transferência)

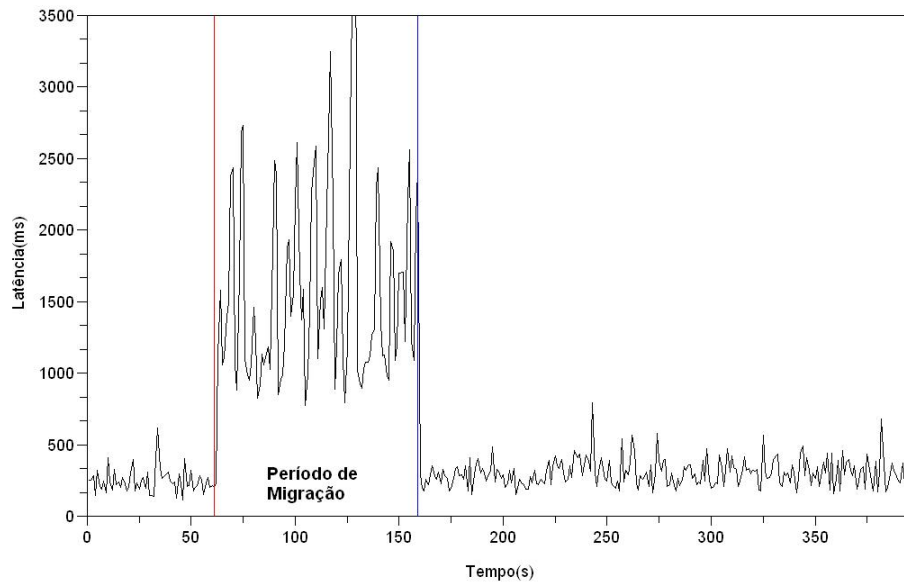


Figura 4.8: Testes simultâneos: execução do dbench (latência)

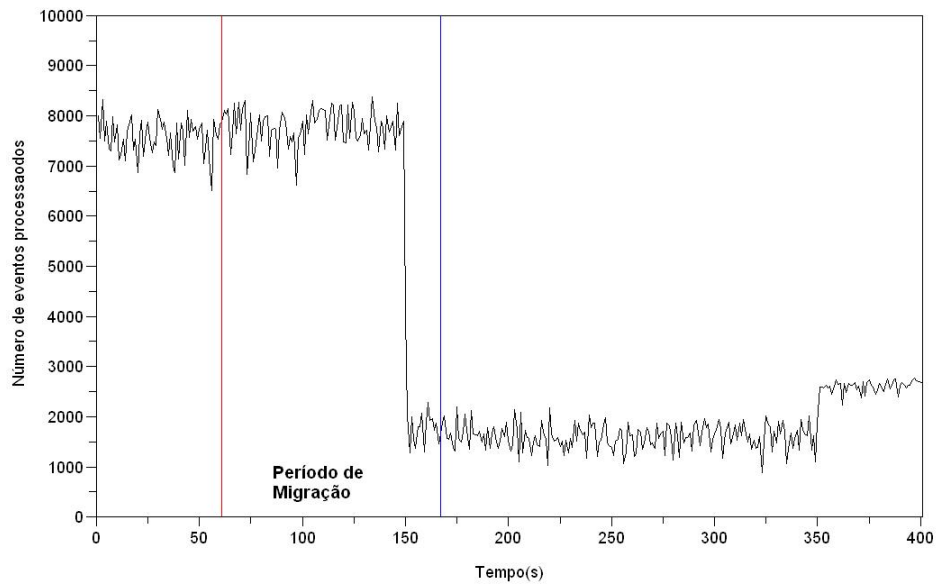


Figura 4.9: Testes simultâneos: execução do SysBench (processamento)

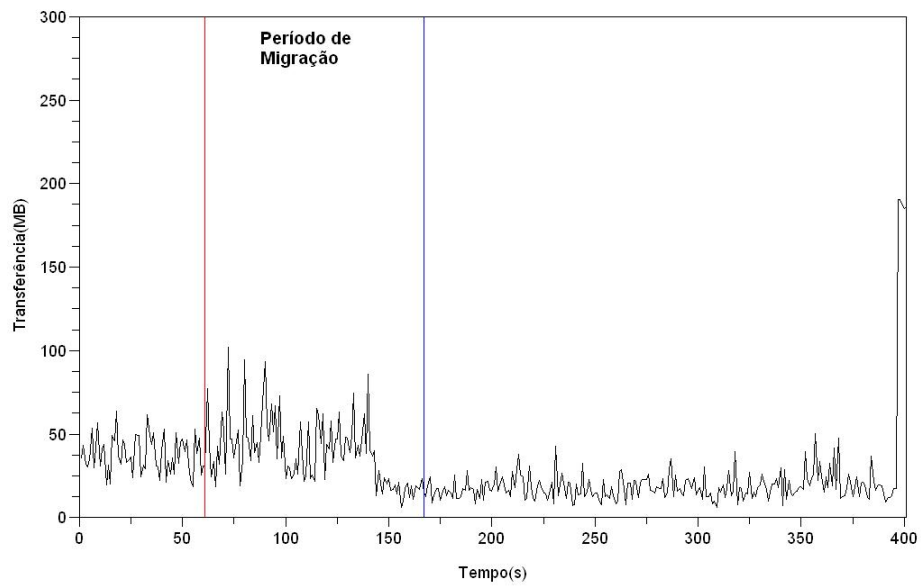


Figura 4.10: Testes simultâneos: execução do SysBench (memória)

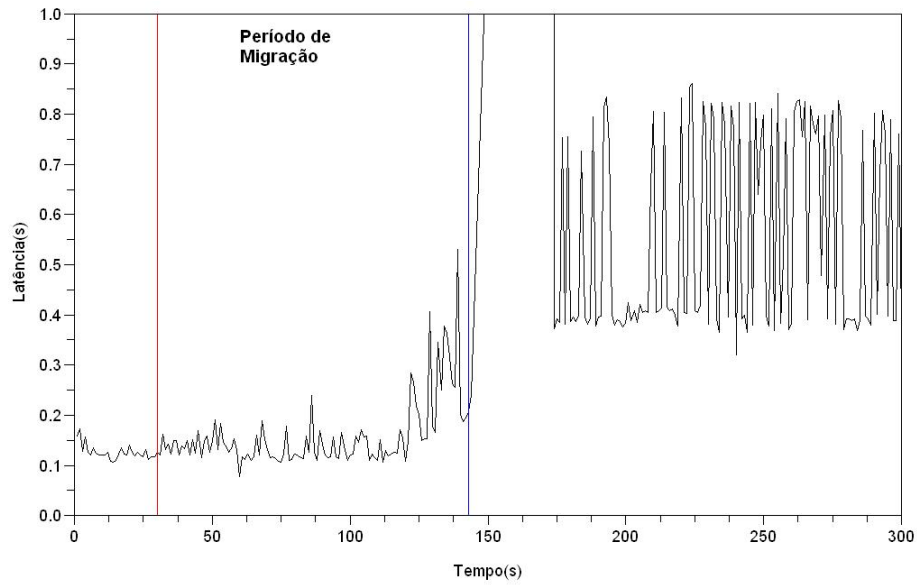


Figura 4.11: Execução do comando ping durante a compilação do kernel do Linux

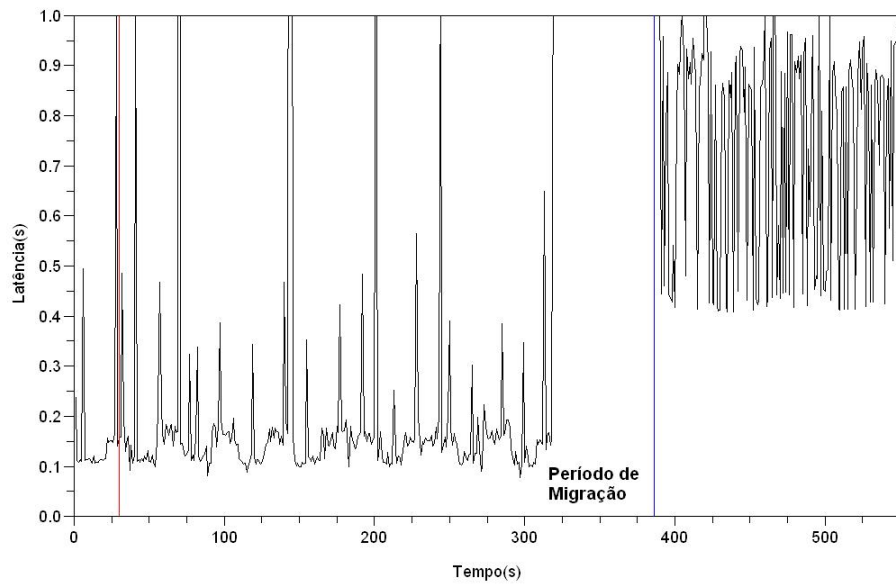


Figura 4.12: Execução do comando ping em um algoritmo de sobrecarga da RAM

Capítulo 5

Conclusão

O objetivo deste trabalho é o desenvolvimento de uma solução que permita realizar a migração *online* de máquinas virtuais com eficiência, confiabilidade e baixo custo. Neste trabalho foram adotadas tecnologias e ferramentas livres e que são utilizadas no mercado atualmente. As ferramentas utilizadas foram: Gentoo Linux (GENTOO, 2009), Debian Linux (DEBIAN, 2009), o hipervisor Xen (XEN, 2009), o iSCSI *target* IET (IET, 2009) e o iSCSI *initiator* openiSCSI (OPENISCSI, 2009). Uma parte dos resultados deste trabalho são apresentados em (ALKMIM; UCHôA, 2009).

Para ambientes aonde não é necessário um ambiente computacional robusto, a utilização da migração de baixo custo adotada neste trabalho se apresentou viável, evitando assim a aquisição de um *storage* físico que possui um custo elevado. Além disso, verificar a viabilidade e os pontos fracos da utilização de um *storage* virtual permite ampliar a solução para ambientes que precisem de uma solução mais robusta. A utilização de uma instalação local linux e de aplicações livres tornam a solução mais flexível do que a utilização de uma solução com um *storage* físico. Além disto, solução apresentou como unico ponto fraco o desempenho da rede devido a utilização de uma rede ethernet de 100 megabits. A utilização de conexões mais velozes podem tornar a solução viável inclusive para ambientes que necessitem de um alto desempenho.

A análise de desempenho da migração de máquinas virtuais utilizando 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 onde 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.

Neste trabalho também foram apresentados os resultados de uma análise de desempenho da utilização de um *storage* virtual exportando um disco para uma máquina virtual. Todas as ferramentas utilizadas são livres, inclusive as de *benchmark*. Os resultados mostram que os fatores que precisam de uma transferência de dados via rede apresentaram uma redução de desempenho considerável. O motivo para isto foi a utilização de uma rede de apenas 100Mbits. Os resultados mostraram que a utilização de um *storage* virtual em máquinas virtual é viável, porém a capacidade da rede deve ser escolhida de acordo com a capacidade de transferência dos discos, para que não ocorra sobrecarga da rede. O protocolo utilizado para exportar o disco foi o iSCSI. Um possível trabalho futuro seria a utilização do protocolo AoE (Ata over Ethernet) e comparar os resultados com os obtidos neste trabalho.

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, pode ter ocorrido 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.

Um possível trabalho futuro seria a utilização do Ata-Over-Ethernet (AOE), ao invés do protocolo iSCSI e consequente realizar uma comparação entre os dois. A análise do desempenho da solução proposta utilizando uma rede Gigabit ou duas placas de redes também é interessante.

Referências Bibliográficas

ALKMIM, G. P. Aplicações da virtualização em empresas. *II WIA - Workshop de Informática Aplicada*, p. 124–129, 2009.

ALKMIM, G. P.; UCHÔA, J. Q. Uma solução de baixo custo para a migração de máquinas virtuais. *VIII WPerformance - Workshop em Desempenho de Sistemas Computacionais e de Comunicação - XXIX CSBC - Congresso da Sociedade Brasileira de Computação*, 2009.

APACHE. *The Apache Software Foundation*. Disponível em <http://www.apache.org/>, acessado em 06/2009, 2009.

ATTANASIO, C. R.; MARKSTEIN, P. W.; PHILLIPS, R. J. Penetrating an operating system: a study of vm/370 integrity. *IBM System Journal*, v. 15, n. 1, p. 102–116, 1976.

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEBAUER, R.; PRATT, I.; WARFIELD, A. Xen and the art of virtualization. *nineteenth ACM symposium on Operating systems principles*, p. 164–177, 2003.

BRADFORD, R.; KOTSOVINOS, E.; FELDMANN, A.; SCHIÖBERG, H. Livewide-area migration of virtual machines including local persistent state. *3rd international conference on Virtual execution environments*, p. 179–189, 2007.

BULLERS, W. I.; BURD, S.; SEAZZU, A. F. Virtual machines – an idea whose time has returned: application to network, security, and database courses. *SIGCSE technical symposium on computer science Education*, p. 102–106, 2006.

CLARK, C.; FRASER, K.; HAND, S. Live migration of virtual machines. *2nd conference on Symposium on Networked Systems Design and Implementation*, p. 273–286, 2005.

CREASY, R. J. The origin of the vm/370 time-sharing system. *IBM Journal of Research and Development*, v. 25, n. 5, p. 483–480, 1981.

DBENCH. *dbench benchmark*. Disponível em <http://samba.org/ftp/tridge/dbench/>, acessado em 05/2009, 2009.

DEBIAN. *Debian Linux*. Disponível em <http://www.debian.org/>, acessado em 05/2009, 2009.

DOHERTY, W. J.; KELISKY, R. P. Managing vm/cms systems for user effectiveness. *IBM System Journal*, v. 18, n. 1, p. 143–163, 1979.

EGI, N.; GREENHALGH, A.; HANDLEY, M.; HOERDT, M.; MATHY, L.; SCHOOLEY, T. Evaluation xen for router virtualization. *16th International Conference on Computer Communications and Networks*, p. 1256–1261, 2007.

GENTOO. *Gentoo Linux*. Disponível em <http://www.gentoo.org/>, acessado em 05/2009, 2009.

HENDRICKS, E. C.; HARTMANN, T. C. Evolution of a virtual machine subsystem. *IBM System Journal*, v. 18, n. 1, p. 111–142, 1979.

HU, J.; MEINEL, C.; SCHMITT, M. Tele-lab it security: An architecture for interactive lessons for security education. *Special Interest Group on Computer Science Education*, v. 36, p. 412–416, 2004.

HUANG, W.; J., L.; ABALI, B.; KOOP, M.; PANDA, D. Nomad: Migrating os-bypass networks in virtual machines. *3rd international conference on Virtual execution environments*, p. 158–168, 2007.

IET. *iSCSI Enterprise Target*. Disponível em <http://iscsitarget.sourceforge.net/>, acessado em 05/2009, 2009.

JONES, M. T. *Virtual Linux: An overview of virtualization methods, architectures, and implementations*. Disponível em <http://www.ibm.com/developerworks/library/l-linuxvirt/>, acessado em 05/2009, 2006.

KING, S. T.; DUNLAP, G. W.; CHEN, P. M. Debugging operating systems with time-traveling virtual machines. *USENIX Annual Technical Conference*, p. 1–15, 2005.

LAUREANO, M.; MAZIERO, C.; JAMHOUR, E. *Detecção de Intrusão em Máquinas Virtuais*. 5º Simpósio de Segurança em Informática – SSI. ITA – São José dos Campos, 2003.

LAUREANO, M. A. P. *Uma Abordagem para a Proteção de Detectores de Intrusão Baseada em Máquinas Virtuais*. Ph. D. Thesis, Pontifícia Universidade Católica do Paraná, 2004.

MACKINNON, R. A. The changing virtual machine environment: Interfaces to real hardware, virtual hardware, and other virtual machines. *IBM System Journal*, v. 18, n. 1, p. 18–46, 1979.

MEYER, R. A.; SEAWRIGHT, L. H. A virtual machine time-sharing system. *IBM System Journal*, v. 9, n. 3, p. 199–218, 1970.

MYSQL. *MySQL*. Disponível em <http://www.mysql.com/>, acessado em 06/2009, 2009.

NAGARAJAN, A. B.; MUELLER, F.; ENGELMANN, C.; SCOTT, S. L. Proactive fault tolerance for hpc with xen virtualization. *21st annual international conference on Supercomputing*, p. 23–32, 2007.

NANDA, S.; CHIUEH, T. *A Survey on Virtualization Technologies*. RPE Report, 2005.

NELSON, M.; LIM, B.; HUTCHINS, G. Fast transparent migration for virtual machines. *USENIX Annual Technical Conference*, p. 391–394, 2005.

OPENISCSI. *Open iSCSI*. Disponível em <http://www.open-iscsi.org/>, acessado em 05/2009, 2009.

PARMELEE, R. P.; PETERSON, T. I.; TILLMAN, C. C.; HATFIELDS, D. J. Virtual storage and machine concepts. *IBM System Journal*, v. 11, n. 2, p. 99–130, 1972.

POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, v. 17, n. 7, p. 412–421, july 1974.

POPEK, G. J.; KLINE, C. S. The pdp-11 virtual machine architecture: A case study. *ACM SIGOPS Operating Systems Review*, v. 9, p. 97–105, 1975.

QEMU. *QEMU Processor Emulator*. Disponível em <http://www.nongnu.org/qemu/>, acessado em 05/2009, 2009.

ROSE, R. *Survey of System Virtualization Techniques*. University of California, 2004.

SATRAN, J.; METH, K.; SAPUNTZAKIS, C.; SYSTEMS, C.; CHADALAPAKA, M.; CO., H.-P.; ZEIDNER, E. *RFC3720 - Internet Small Computer Systems Interface (iSCSI)*. IETF - The Internet Engineering Task Force, 2004.

SOUZA, F. B. *Uma arquitetura para Monitoramento e Medição de Desempenho para Ambientes Virtuais*. Ph. D. Thesis, Universidade Federal de Minas Gerais, 2006.

SU, D.; CHEN, W.; HUANG, W.; SHAN, H.; JIANG, Y. Smartvisor: towards an efficient and compatible virtualization platform for embedded system. *Second Workshop on Isolation and Integration in Embedded Systems*, p. 37–41, 2009.

SYSBENCH. *SysBench benchmark*. Disponível em <http://sysbench.sourceforge.net/>, acessado em 05/2009, 2009.

URSCHEI, F.; PELEGRINI, E. J.; SILVA, M. A. L.; MIDORIKAWA, E. T.; CARVALHO, T. C. M. B. Análise multiparamétrica do overhead de rede em máquinas virtuais. *IV Workshop de Sistemas Operacionais*, p. 816–827, 2007.

VIRTUALBOX. *VirtualBox*. Disponível em <http://www.virtualbox.org/>, acessado em 05/2009, 2009.

WANG, Y.; KELLER, E.; BISKEBORN, B.; MERWE, J. van der; REXFORD, J. Virtual routers on the move: live router migration as a network-management primitive. *ACM SIGCOMM 2008 conference on Data communication*, p. 231–242, 2008.

WLODARZ, J. J. *Virtualization: A double-edged sword*. arXiv:0705.2786v1 [cs.OS]. Disponível em <http://arxiv.org/pdf/0705.2786v1>, acessado em 05/2009, 2007.

XEN. *Xen Hypervisor*. Disponível em <http://www.xen.org/>, acessado em 05/2009, 2009.

XEN-DOC. *Documentação do Xen*. Incluso no Código Fonte do Xen, 2009.

XEN-MAN. *Xen Interface Manual - Xen v3.0 for x86*. University of Cambridge, UK, 2005.

YOUSEFF, L.; WOLSKI, R.; GORDA, B.; KRINTZ, C. *Paravirtualization For HPC Systems*. Santa Barbara, 2006.