

Balanceamento de Serviços com Servidores Virtualizados

Vicente Lucas S. Zotti, Eugeni Dodonov

¹Departamento de Computação - Universidade Federal de São Carlos (UFSCAR)
CEP 13565-905 – São Carlos – SP – Brazil

seazotti@bol.com.br, eugeni@dodonov.net

Abstract. *According to surveys (COMPUTERWORLD, 2009) about 85 % of the computational power of an organization is idle for most of the time. Thus, the concern with the optimization of energy use, with the operating cost and better use of hardware servers are the main reasons why migrating physical servers to virtualized systems (VMWARE, 2010b). When the strategy adopted is virtualization a study on the computational resources is necessary, as well as creating a strategy to manage the resulting platform. This work aims to show a study on the migration of virtualized servers among different physical equipment, seeking the balance of services and increasing the availability of virtualized servers.*

Resumo. *Pesquisas revelam (COMPUTERWORLD, 2009) que cerca de 85% do poder computacional fica ocioso durante a maior parte do tempo. Assim, otimização do uso de energia, do custo operacional e melhor aproveitamento do hardware são os principais motivos para que haja a migração de servidores físicos para sistemas virtualizados (VMWARE, 2010b). Quando a estratégia adotada é de virtualização é necessário um estudo sobre os recursos computacionais e a criação de uma estratégia para gerenciamento da nova plataforma. Esse trabalho tem por objetivo realizar um estudo sobre a migração de servidores virtualizados para equipamentos físicos distintos, visando o balanceamento de serviços e o aumento da disponibilidade dos servidores virtualizados.*

1. Introdução

Pesquisas revelam (COMPUTERWORLD, 2009) que cerca de 85% do poder computacional fica ocioso durante a maior parte do tempo. A preocupação com a otimização do uso de energia, o custo operacional e o melhor aproveitamento do *hardware* dos servidores. Segundo pesquisa realizada, (VMWARE, 2010b) são os principais motivos para que haja a migração de servidores físicos para sistemas operacionais virtualizados.

O desenvolvedor de soluções em virtualização *VMware*, por exemplo, disponibilizou uma calculadora virtual, onde é possível simular a economia quando é adotada a solução de virtualização (VMWARE, 2010).

A utilização de máquinas virtuais vem sendo proposta desde os anos 1980 (DONALDSON, 1987); no entanto, as recentes evoluções desta área justificam o crescimento de 22,5% na comercialização de estrutura para virtualização de servidores (COMPUTERWORLD, 2009). Nesse contexto, este trabalho tem por objetivo apresentar um estudo sobre a migração de servidores virtualizados para equipamentos físicos distintos, visando o balanceamento de serviços e aumentando a disponibilidade dos servidores virtualizados. O capítulo 2 têm como foco a contextualização de virtualização e seus modos de

execução. Os capítulos 3 e 4 abordam a utilização da virtualização em ambientes de servidores e sua contextualização quanto a balanceamento de carga, e o capítulo 5 descreve o ambiente e experimentos executados. Por fim, o capítulo 6 aborda trabalhos relacionados e o capítulo 7 conclui este trabalho e apresenta um estudo sobre trabalhos futuros.

2. Virtualização

O conceito de virtualização foi amplamente estudado e utilizado no início da década de 70, abrindo portas ao compartilhamento de recursos computacionais dos *mainframes* entre as máquinas virtuais existentes (GOLDBERG, 1974).

Existem três princípios que garantem o sucesso da tecnologia de virtualização (GARFINKEL et al., 2003). Primeiro, a camada de virtualização deve garantir o isolamento de cada máquina virtual, garantindo que a performance de uma máquina virtual não afete o desempenho de outra. Segundo, é imprescindível o suporte a vários sistemas operacionais para acomodar os diversificados aplicativos existentes. Terceiro, o *overhead* adicionado às camadas de virtualização deve ser o menor possível.

Os Monitores de Máquinas Virtuais (VMM) podem ser implementados utilizando dois tipos de virtualização, a virtualização em nível de *hardware* ou a virtualização em nível de *software*. Uma das grandes vantagens de se utilizar VMM é que cada máquina virtual possui um espaço de execução isolado (ROSENBLUM, 2004).

2.1. Tipos de Virtualização

Existem três tipos de virtualização: Emulação de **hardware (hardware emulation)**, Virtualização completa (**full-virtualization**) e Para-Virtualização (**para-virtualization**).

A virtualização por emulação de *hardware* é considerada mais complexa pelo fato de ter que emular de maneira idêntica o comportamento do *hardware*. Isto implica em emular os ciclos de *clock*, o conjunto de instruções, os estados de execução (*pipeline*) do processador e até mesmo a memória *cache* (OLIVEIRA, 2007). Por estas características, esse tipo de virtualização se torna ideal para desenvolvedores de firmware, pois desta forma a solução pode ser validada sem a necessidade do *hardware* real esteja disponível. A grande desvantagem deste tipo de virtualização é a lentidão com qual a emulação acontece, podendo ser até 1000 vezes mais lenta que o suposto *hardware* real (JONES, 2006). O *hardware* a ser emulado sofre a demora pela diferença sobre o *hardware* real, sobre o qual o sistema de virtualização está hospedando, como ilustra a Figura 1. Nesta imagem há dois *hardwares* emulados: o primeiro sendo emulado pela *virtual machine* (VM) "A" e o segundo pela *virtual machine* (VM) "B". Esta camada representada pelo *hardware* VM A e pelo *hardware* VM B compõe a camada de virtualização de *hardware*. Emulação por *hardware* pode ser exemplificada quando você roda um sistema operacional modificado destinado a um *PowerPC* em um hospedeiro portando um processador *ARM*. Você pode rodar múltiplas máquinas virtuais e cada uma delas executar um processador diferente.

Virtualização completa é a técnica utilizada para que seja executado qualquer *software* sem existir nenhuma alteração. Para isso, esta técnica faz uma simulação completa do *hardware* da máquina de modo que qualquer sistema operacional possa ser executado. Esta simulação implica em representar o conjunto de instruções de um processador, memória principal, interrupções, exceções e acesso aos diversos dispositivos existentes. Uma das características deste tipo de virtualização, é que necessita-se obrigatoriamente

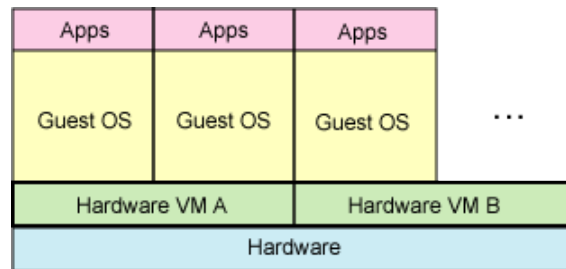


Figura 1. Emulação de Hardware
(JONES, 2006)

de um *hardware* com características específicas, uma vez que as instruções de execução privilegiada, como por exemplo o acesso a I/O, devem ser interceptadas e somente serem executadas de acordo com o que está definido pela camada do monitor de máquinas virtuais.

Diferentemente da emulação, a virtualização completa é realizada com maior eficácia, pois não necessita representar os estados de execução do *hardware*. A virtualização completa do *hardware* feita por esta técnica de virtualização, geralmente, simula dispositivos padrões do mercado de modo a facilitar a instalação e configuração dos sistemas virtualizados. Exemplos de VMM que têm a capacidade de fazer esta virtualização são a ferramenta VMware ESX (VMWARE, 2010a) e o KVM (Kernel Virtual Machine) (RITZAU; WARNKE, 2010). Quando uma destas ferramentas é instalada, uma interface de rede 3Com, por exemplo, pode ser simulada para o ambiente virtualizado como uma placa de rede AMD PCNet. E esta simulação se repete também para placas de vídeos, *chipset* e discos rígidos. A Figura 2 ilustra este tipo de virtualização.

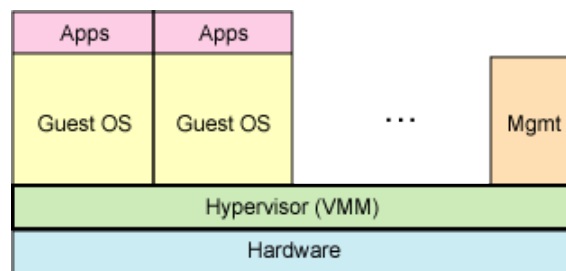


Figura 2. Virtualização Completa
(JONES, 2006)

Já na para-virtualização existe uma técnica de virtualização onde existe uma *Application Programming Interface* (API) para as máquinas virtuais. Essa API é parecida com o *hardware* real. Esta técnica de virtualização necessita que o sistema operacional virtualizado seja explicitamente capacitado a permitir sua execução. Desenvolvedores de sistemas para-virtualizados propuseram um acordo para criar os *hypercalls*, que são os "*system calls*" para o *hypervisor* (XENSOURCE, 2007). Desta forma, ao invés das "*system calls*" do sistema operacional virtualizado trabalharem diretamente no *hardware* real, elas trabalharão sobre o *hardware* virtualizado pela VMM (**Virtual Machine Monitor**) que é responsável pela execução das instruções vindas de todas as máquinas virtuais, como ilustra a Figura 3.

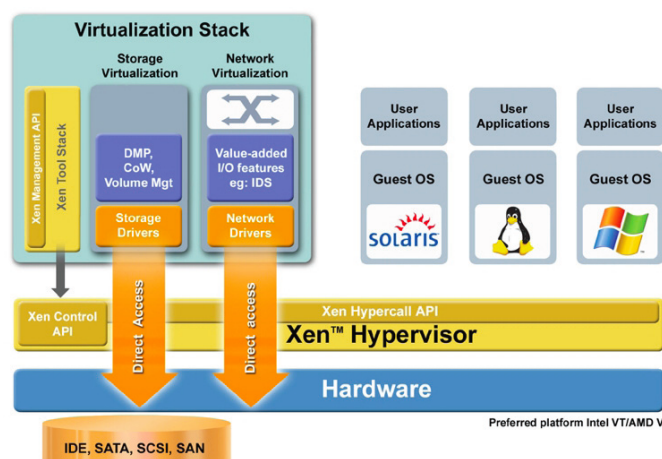


Figura 3. Para-Virtualização
(SYSTEMS, 2005)

2.2. Processadores com suporte a Virtualização

Até o primeiro semestre de 2010, existem duas arquiteturas de processadores que suportam virtualização por *hardware*: Intel VT (*Virtualization Technology*) e AMD-V (*AMD Virtualization*) (JONES, 2006). O suporte à virtualização por *hardware* não substitui a infraestrutura virtual, mas visa prover uma maior eficiência no seu funcionamento (VMWARE, 2008). A seguir, o funcionamento dessas tecnologias será detalhado.

2.2.1. Intel VT

Intel Virtualization Technology (*Intel VT*) é a tecnologia de virtualização Intel, assistida por *hardware* que, associada a um conjunto de melhorias para monitores de máquinas virtuais, acelerará o desempenho do sistema e sua aplicação em ambientes virtualizados. **Virtual Machine Device Queues** (VMDq) é uma tecnologia que faz parte do Intel VT para conectividade, orientada para a melhoria de desempenho da rede e traz também a redução na utilização do CPU (CHINNI, 2007).

Processadores com tecnologia Intel VT possuem um conjunto de instruções extra chamado Virtual Machine Extensions (*VMX*). Até o primeiro semestre de 2010 eram comercializados dois tipos de processadores com tecnologia de virtualização, sendo o VT-x para suporte a virtualização de processadores IA-32, e a tecnologia VT-i, que traz suporte à virtualização de processadores da linha *Itanium*.

Intel VT-x, traz duas formas de operação da CPU: execução VMX root e execução VMX não root. A execução em modo VMX root se destina para uso de um MMV e seu comportamento é muito semelhante ao IA-32 sem VT-x. Execução em modo não root do VMX não oferece um ambiente alternativo para um controlador MMV (NEIGER et al., 2006).

Intel VT-i expande a arquitetura *Itanium* com extensões para o hardware do processador e o *firmware Processor Abstraction Layer (PAL)*. A tecnologia VT-i permite que sistemas operacionais possam ser executados no nível de privilégio para os quais fo-

ram concebidos e cria interceptações para um MMV, necessárias para a criação de uma máquina virtual completa (NEIGER et al., 2006).

2.2.2. AMD-V

A tecnologia de virtualização que a fabricante de processadores AMD (**Advanced Micro Devices**) desenvolveu é chamada de AMD-V (**AMD Virtualization**). A tecnologia AMD-V é baseada em plataforma 64 *bits* e suas principais características são:

Extensões de virtualização para o conjunto de instruções x86. Este conjunto de instruções permite criar máquinas virtuais para que vários sistemas operacionais e suas aplicações possam ser executadas simultaneamente (DEVICES, 2010).

Translation Look-Aside Buffer (TLB), é um *buffer* que armazena traduções de endereços de memória. A cada referência de memória, o processador verifica o TLB para determinar se é necessária a tradução; caso esta tradução estiver armazenada, a instrução é simplesmente executada (DEVICES, 2008a).

Rapid Virtualization Indexing (RVI), ajuda a acelerar o desempenho de muitos aplicativos virtualizados. A tecnologia RVI permite ao MMV dar instruções ao *hardware* e assim eliminar a necessidade de páginas de sombra. Páginas de sombra armazenam informações sobre a localização física na memória dos hóspedes. Afim de deixar as páginas de sombra mais coerentes com as tabelas do convidado, o MMV causou um *overhead* na execução, para os quais o convidado tinha que atualizar frequentemente as suas estruturas de tabela de página. Afim de resolver este problema, foi desenvolvida a tecnologia RVI a qual remove a maior parte do *overhead* incorrido (BHATIA, 2009).

2.3. Monitores de Máquinas Virtuais

O VMM também pode ser chamado de *Hypervisor*. O *hypervisor* é uma plataforma de virtualização que possibilita a execução de vários sistemas operacionais em um único *hardware*. Cada instância de um sistema operacional convidado é chamado de máquina virtual (VM), já que o seu *hardware* é virtualizado, trabalhando de forma dedicada entre as máquinas virtuais.

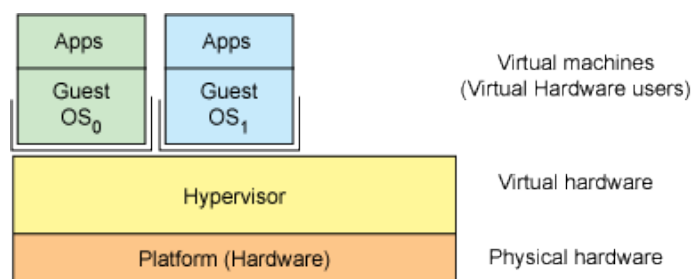


Figura 4. *Arquitetura simples que ilustra uma virtualização do hardware* (JONES, 2009)

Na Figura 4, o monitor de máquinas virtuais é a camada de software que fornece o hardware virtualizado às máquinas subjacentes. Nem todas as soluções de virtualização utilizam o mesmo estilo. Os sistemas operacionais virtualizam acesso aos recursos

subjacentes da máquina aos processos; os monitores de máquinas virtuais fazem a mesma coisa mas, em vez dos processos, eles executam essa tarefa para os sistemas operacionais convidados inteiros (JONES, 2009). A seguir são apresentados exemplos de *hypervisor*.

2.3.1. VMware

VMware é um *hypervisor* pioneiro utilizando a tecnologia x86, desenvolvido em 1998. A empresa oferece uma variedade de abordagens, baseadas em vários softwares. Os principais produtos oferecidos pela VMware Inc no primeiro semestre de 2010 são relacionados a seguir (VMWARE, 2010a).

Para plataforma de *Data Center* são oferecidas as ferramentas VMware Server 2 e VMware ESXi. A ferramenta VMware Server é uma ferramenta gratuita que pode ser copiada diretamente do site do fabricante do software. Esta ferramenta consegue aproveitar os recursos de virtualização dos processadores AMD (AMD-V) e Intel (Intel-vt) e pode ser instalada em plataformas Windows ou Linux, suportando vários sistemas operacionais virtualizados (Windows 2008 Server, Fedora 13, Windows 7 e etc...). Esta versão não tem arquitetura baseada em *hypervisor* por se tratar de uma versão mais simples.

A ferramenta VMware ESXi é uma ferramenta preparada para trabalhar com recursos superiores de *hardware* e por isso se torna uma ferramenta com maior desempenho para o suporte à virtualização. A versão gratuita do ESXi é limitada a 4-way SMP virtual e a versão *shareware* deste software dá suporte para até 8-way vias SMP virtual.

Para produtos da linha *desktop* são oferecidas as ferramentas VMware Player e VMware Fusion. A ferramenta VMware Player é um executor de máquinas virtuais que permite executar vários sistemas operacionais ao mesmo tempo em seu computador. O VMware Player pode ser usado para executar uma cópia virtual de um computador antigo permitindo desativá-lo. A ferramenta VMware Fusion é um executor de máquinas virtuais restrito a sistemas operacionais Mac OS. Com esta ferramenta é possível criar máquinas virtuais de vários sistemas operacionais (VMWARE, 2009) e é possível integrar os arquivos da máquina hospedeira com a máquina virtualizada e também os recursos de *hardware* existentes.

2.3.2. VirtualBox

VirtualBox é uma ferramenta de virtualização que trabalha com arquitetura x86, e que dá suporte à virtualização de hardware através das tecnologias dos processadores Intel VT e AMD-V. VirtualBox pode ser gerenciado através de máquinas remotas utilizando seu utilitário chamado VBoxManage, que permite controlar todos os recursos do VirtualBox. As definições das máquinas virtuais são armazenadas inteiramente em XML e são independentes das máquinas locais, podendo portanto ser facilmente migradas para outros computadores (ORACLE, 2009). O VirtualBox é uma ferramenta *OpenSource* que traz várias facilidades em suas operações. Esta ferramenta é executada com suporte a virtualização completa e faz a emulação dos periféricos necessários através do QEMU (ORACLE, 2010).

Desde o dia 30 de novembro de 2009, com o lançamento da versão VirtualBox 3.1,

o desenvolvedor da ferramenta adicionou o recurso *Teleport*. Com este recurso é possível fazer migrações de uma máquina física para outra, sem a necessidade de se parar a máquina virtualizada. É possível inclusive executar a migração para diferentes arquiteturas e sistemas operacionais (HOLWERDA, 2009).

2.3.3. QEMU

O QEMU é outro exemplo de ferramenta para virtualização que suporta dois modos de operação, emulação de modo de usuário e emulação do modo de sistema. A emulação do modo de usuário permite a um processo construído para uma CPU ser executado em outra. Isto acontece porque o QEMU faz a conversão dinâmica das instruções para a CPU de destino e conversão de chamadas do sistema Linux apropriadamente. A emulação do modo de sistema permite a emulação de um sistema integral, incluindo processador e periféricos diversos (JONES, 2007).

Quando o x86 está sendo emulado em um sistema *host x86*, o desempenho quase nativo pode ser alcançado utilizando o chamado acelerador de QEMU. Isso permite a execução do código emulado diretamente na CPU. Em sistemas operacionais Linux, isto ocorre por meio de um módulo do *kernel*. O conversor dinâmico do QEMU permite a conversão em tempo de execução de instruções para uma CPU virtualizada através do processador físico de origem assim fornecendo emulação. Isto pode acontecer de uma maneira de força bruta (mapeando instruções de uma CPU para outra), mas isso nem sempre é simples; em alguns casos, pode requerer múltiplas instruções ou mudanças no comportamento com base nas arquiteturas sendo convertidas (JONES, 2007). Utilizando o QEMU como um emulador do sistema, é fornecida uma extensa variedade de periféricos. QEMU suporta virtualização quando executando sob o *hypervisor* Xen ou usando o módulo kernel no Linux (KVM). Ao usar o *hypervisor* KVM, o QEMU pode virtualizar servidores, PowerPC e computadores S390 (BELLARD, 2010).

2.3.4. KVM

O KVM (**Kernel Virtual Machine**) é uma infraestrutura de virtualização residente no *kernel* para o *hardware* x86 Linux. Ele dá suporte à virtualização completa e pode funcionar sem modificações no *kernel*, mas requer as tecnologias de virtualização dos processadores (Intel VT e AMD-V). É constituído por módulos de *kernel*. O módulo *kvm.ko* fornece a infra-estrutura de virtualização do núcleo, um módulo específico do processador (*intel.ko* *kvm* ou *amd.ko* *kvm*, dependendo da tecnologia de virtualização do processador) é carregado e usado em conjunto com o QEMU para emular o hardware, tais como a placa de rede, disco rígido, placa gráfica e assim por diante. O KVM dá suporte a uma série de ferramentas de gerenciamento, como por exemplo o *Virsh*. Esta ferramenta é executada em linha de comando, permitindo fazer todo o gerenciamento das máquinas virtuais. Outra ferramenta é o *Virt-Manager*, que oferece uma interface gráfica para o gerenciamento das máquinas virtuais. Ambas as ferramentas são fortemente dependentes da biblioteca *libvirt* (TUXRADAR, 2010).

Este *hypervisor* fornece suporte a multiprocessamento simétrico (SMP) e suporta recursos de nível corporativo, como migração ativa. Essa tecnologia é implementada com

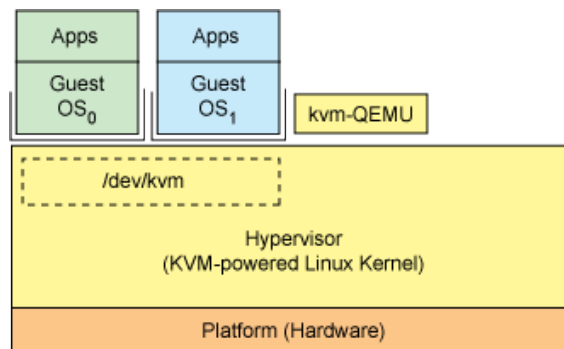


Figura 5. Virtualização de alto nível com KVM
(JONES, 2009)

dois componentes. O primeiro módulo carregável pelo KVM fornece o gerenciamento de hardware de virtualização com o kernel do Linux, expondo os recursos através do sistema de arquivos `/proc`. O segundo componente fornece a emulação de plataforma PC, que é fornecida por uma versão modificada do QEMU. O QEMU é executado como um processo de espaço do usuário, coordenando junto com o kernel os pedidos do sistema operacional convidado, como ilustra a Figura 5 (JONES, 2009).

2.3.5. XEN

Xen é uma plataforma de virtualização de servidores que oferece uma estrutura muito parecida com a ferramenta VMware ESXi. XenServer utiliza o *hypervisor* para utilizar cada servidor no qual está instalada esta solução, permitindo que cada *host* possa trabalhar com múltiplas máquinas virtuais simultaneamente. Com a solução XenServer, se estende a ideia de vários servidores de virtualização, criando assim um *pool* de recursos, cujo armazenamento, memória, CPU e recursos de rede podem ser controlados de uma forma dinâmica, oferecendo o aumento de confiabilidade e disponibilidade de suas máquinas virtuais (XENSOURCE, 2007).

A ferramenta XenServer permite atribuir até 16 servidores x86 em um único *pool* de recursos. Múltiplos *pools* podem ser gerenciados através do console de gerenciamento XenCenter. Um *pool* é uma coleção dos recursos físicos de cada servidor XEN, cujos recursos são virtualizados para criar um suporte de *hardware* para as máquinas virtuais. Desta forma, caso um equipamento falhe, isso pode rapidamente ser solucionado substituindo o nó que falhou (XENSOURCE, 2007).

Em outubro de 2007 a XenSource foi comprada pela Citrix e, desde então o desenvolvimento de novas versões e suporte ao XenServer é feito por este desenvolvedor de soluções.

2.4. Gestão de soluções de virtualização: libvirt

Embora muitas soluções de virtualização sejam livres, ferramentas de gestão apropriadas são geralmente caras. Além disso, muitas vezes é necessário para cada solução de virtualização uma ferramenta de gestão diferente. Para resolver estes problemas, a biblioteca *libvirt* foi desenvolvida. *Libvirt* é uma API desenvolvida em linguagem C, que é execu-

tada entre o software de virtualização e a ferramenta de gestão, que oferece um conjunto de ferramentas de gerenciamento. Exemplos destas ferramentas são *Virt-Manager* e *virsh* (RITZAU; WARNKE, 2010).

Até o primeiro semestre de 2010, o *libvirt* oferece suporte aos seguintes monitores de máquinas virtuais (LIBVIRT, 2010a):

- Xen, hypervisor linux e Solaris.
- QEMU, emulador e KVM, hypervisor Linux.
- LXC, sistema de *container* Linux.
- OpenVZ, sistema de *container* Linux.
- User Mode Linux, kernel para-virtualizados.
- VirtualBox, hypervisor com suporte a vários Sistemas Operacionais.
- VMware ESX e GSX, hypervisor.

Na Figura 6, é ilustrado o funcionamento da biblioteca *libvirt* juntamente aos monitores de máquinas virtuais. Nesta figura temos o *libvirt* trabalhando com três monitores de máquinas virtuais distintos, sendo gerenciado por três ferramentas de gerenciamento distintas, porém todos compatibilizados pelo *libvirt*.

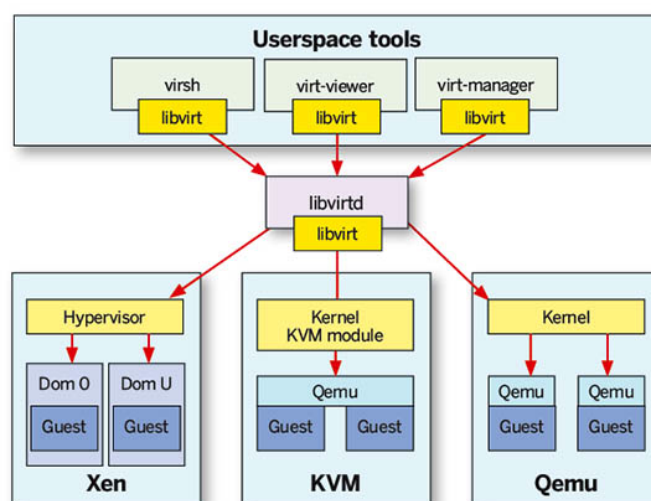


Figura 6. Arquitetura de trabalho da biblioteca libvirt
(TUXRADAR, 2010)

A biblioteca *libvirt* permite o acesso a monitores de máquinas virtuais que estiverem remotos, através de conexões devidamente autenticadas e encriptadas.

Na máquina remota, o *daemon libvirtd* deve estar em execução, e é necessário informar à ferramenta de gerenciamento local que você deseja acessar um recurso remoto. Para isso, deve-se informar o nome do *host* ou IP a ser acessado, que em seguida é passado ao *virConnectOpen* ou (*virsh -c ...*). De um ponto de vista da API, além da mudança de onde se estará acessando as informações, é necessário que as chamadas comuns sejam encaminhadas através da conexão remota de forma transparente e que os valores ou erros do lado remoto sejam devolvidos como se estivessem acontecido localmente (LIBVIRT, 2010b).

3. Virtualização em Ambiente de Servidores

A virtualização em ambientes de infraestrutura de TI traz consigo uma série de benefícios, embora muitas vezes associamos virtualização apenas a economia de energia. Outros benefícios são descritos a seguir.

Economia de Energia: Segundo (COMPUTERWORLD, 2009), a economia dos recursos de empresas é um dos principais benefícios da virtualização. Quando pensamos em virtualização, a economia de energia é um dos benefícios principais, pois traz diretamente economia dos recursos da empresa. Se imaginarmos que cerca de 85% do poder computacional fica ocioso durante a maior parte do tempo e calcularmos que em um ambiente de *backbones* temos centenas de servidores, podemos ver o quanto de energia elétrica está sendo desperdiçada. Este desperdício não se resume a energia elétrica e sim a um conjunto de equipamentos necessários para que o *datacenter* funcione, como por exemplo, o sistema de refrigeração (COMPUTERWORLD, 2009). Economizar significa hoje ter capacidade para modernizar o ambiente de tecnologia e, portanto, vantagem competitiva no mercado.

Otimização de Espaço: Segundo a (COMPUTERWORLD, 2009), a taxa de redução proporcionada no ambiente virtualizado é cerca de 90%, portanto para um parque de cem servidores físicos, pode-se convertê-los em máquinas virtuais e reduzir este número a 10 equipamentos.

Tarefas administrativas de TI: Segundo (VMWARE, 2010b), após a virtualização de seu ambiente de servidores, são otimizadas as tarefas de rotinas administrativas do departamento de TI em cerca de 73%. Esta mesma pesquisa aponta que cerca de 90% dos departamentos de TI gastam mais da metade do tempo em tarefas administrativas de rotina e que 73% das empresas que usam virtualização classificam suas áreas de TI como responsivas ou bastante responsivas. A recuperação de desastres é uma das causas em que os departamentos de TI gastam a maior parte de seu tempo. Com um ambiente virtualizado, contudo, o tempo gasto com a resolução de desastres é muito otimizado, pois dentro das soluções de virtualização existem recursos disponíveis, como os *Snapshots* dos sistemas, que possibilitam restaurar o estado das máquinas virtualizadas antes do desastre em pouco tempo, diminuindo assim os tempos de parada.

Agrupamento de recursos dispersos: Cada vez mais os departamentos de TI sofrem com a missão de fazer sempre mais com menos, à medida que novos sistemas exigem mais do *hardware* e quase sempre o investimento nos equipamentos não acompanha a mesma evolução. Desta forma, a virtualização ajuda criando *pools* compartilháveis, aos quais se pode agregar os servidores já existentes nos departamentos de TI e, quando necessário, agregar novos servidores ao mesmo *pool*. Migrar as máquinas virtuais entre os servidores e otimizar o desempenho das máquinas virtuais eventualmente é uma necessidade de todos os departamentos de TI. Com a estrutura tradicional de servidores físicos essa tarefa é praticamente impossível de ser realizada, pois cada servidor possui um *hardware* diferente, o que dificulta tal migração. Neste sentido, a virtualização vem trazendo avanços importantes ao balancemaneto de serviços nos *datacenters*.

Transformar ambientes de estrutura tradicional em ambientes de estrutura virtualizada não é uma tarefa fácil. É necessário um projeto e análise, onde sejam classificados os candidatos a virtualização, identifiquem-se os *hosts* que vão hospedar os sistemas virtu-

ais convidados e transformem-se os *hard drives* (discos rígidos) em discos virtualizados. Exemplos destas ferramentas de conversão são WinImage e VMware Converter.

4. Balanceamento de serviços em servidores virtualizados

Com o ambiente já virtualizado, podem existir picos de uso da máquina virtualizada, algumas vezes por falhas no escalonamento do projeto de virtualização, de forma que a máquina física não suporte toda a carga e cause lentidão. Ao mesmo tempo, é possível que existam outras máquinas físicas disponíveis e ociosas. Assim, para um melhor aproveitamento da capacidade dos servidores, os operadores de TI podem efetuar migrações de máquinas virtualizadas para outras máquinas físicas, melhorando a velocidade de todos os processos e aproveitando ao máximo os benefícios da virtualização. Migrações podem ocorrer de três formas (DEVICES, 2008b):

Static Migration: Quando a máquina virtual é desligada usando os métodos de seu sistema operacional e logo em seguida é movida para outra máquina física.

Cold Migration: É quando a máquina virtual é suspensa (exemplo: Hibernar do sistema Windows), usando métodos de seu sistema operacional ou acionada a função de *Snapshot* que salva o estado atual da máquina virtualizada. Após acionado um dos dois procedimentos citados, a máquina virtualizada é movida para outra máquina física e iniciado o sistema virtualizado em seu estado anterior.

Live Migration: É quando o próprio monitor de máquinas virtuais, que executa a máquina virtualizada se incumbem de fazer a migração sem paradas. *Live Migration* permite o equilíbrio de carga dinâmica de associações virtualizadas do recurso, a manutenção do *hardware* sem tempo ocioso da máquina virtualizada e o apoio dinâmico contra falhas do sistema.

O maior desafio da migração de máquinas virtuais é que, após a migração da máquina virtualizada entre os servidores físicos ocorrer, a máquina virtualizada deve se comportar da mesma forma como antes da migração [Devices 2008].

4.1. *Live Migration* para migração de máquinas virtuais

Visando avaliar a eficiência dos mecanismos de *live migration* adequados para uso em *datacenters*, este artigo apresenta uma avaliação dos mecanismos existentes para permitir que servidores fiquem disponíveis a maior parte do tempo possível. Este método de migração traz consigo benefícios como, não necessitar de parada, não necessitar de reinicialização do sistema virtualizado e ser de fácil operação (HOLWERDA, 2009).

Esta seção descreve dois monitores de máquinas virtuais que dão suporte ao método de migração *live migration*, o KVM/QEMU, que é um *hypervisor* livre e robusto, e o VirtualBox, que também é de uso livre, porém, ainda não conta com um número grande de utilizadores.

Existe um gerenciador gráfico para o *hypervisor* KVM/QEMU chamado Virt-Manager. Com ele é possível executar o processo de migração com apenas alguns cliques mas, para que esta migração ocorra, são necessários alguns requisitos independentemente da ferramenta de gerenciamento utilizada (KVM, 2010):

- É obrigatório que o disco virtualizado esteja acessível entre a origem e destino. (Exemplo: NFS (*Network File System*) armazenado em pasta compartilhada).

- As máquinas de origem e destino devem estar acessíveis via rede.
- As plataformas dos sistemas operacionais de origem e destino devem ser compatíveis.
- No caso de processadores com tecnologia de virtualização sendo usada pela máquina virtual, esta não poderá ser migrada para outro servidor que não disponha desta tecnologia.

Após o início do processo de migração utilizando o KVM/QEMU o algoritmo a ser seguido é o seguinte (KVM, 2010):

1. Inicialização da máquina virtual no destino, ativando-se uma página de log.
2. Transferência de memória. A máquina virtualizada continua a funcionar. Primeiramente é transferida uma cópia de toda a memória e depois são transferidas as páginas utilizadas enquanto era realizada a transferência de memória.
3. A máquina virtualizada aparece para o gerenciador como se estivesse em pausa. Neste momento é ativada a sincronização dos discos rígidos e a transferência de estado. É executado o mais rápido possível, sem limitação de largura de banda, uma vez que informações de todos os dispositivos e a página de estados ainda estão sendo transferidos.

O VirtualBox implementou o suporte ao *live migration* apenas a partir do dia 30 de novembro de 2009, no lançamento de sua versão 3.1 com o nome de *teleporting* (HOLWERDA, 2009). Para o *teleporting* acontecer no VirtualBox, os seguintes requisitos devem estar alinhados (CORPORATION, 2010):

- No equipamento de destino, a máquina virtualizada deve estar configurada exatamente como a máquina de origem, o que se refere à quantidade de memória e outras configurações de *hardware*.
- Os equipamentos de origem e destino necessariamente devem compartilhar o mesmo disco virtualizado (compatilhamento via NFS ou SMB).

5. Experimentos realizados

O objetivo dos experimentos realizados, é demonstrar que não ocorre nenhuma parada enquanto a máquina virtual é migrada e que após a conclusão da migração, a máquina virtual continua em execução sem ser afetada.

A tabela 1 detalha as especificações de *hardware*, *software* e as configurações de rede dos equipamentos utilizados para os experimentos:

O Ambiente dos experimentos continha máquinas físicas tendo acesso ao mesmo disco virtualizado através de um compartilhamento NFS (**Network File System**). Foi adotado o sistema operacional Fedora 13 por este ser *Open Source*, por ter vários fóruns de auxílio e prover suporte nativo à virtualização. O suporte a virtualização foi melhorado a partir da versão 12 deste sistema operacional.

Para que o **live migration** funcione, são necessárias uma série de pré-requisitos, como descrito anteriormente na seção *Live Migration*. Através dos comandos `grep -color vmx /proc/cpuinfo` e `grep -color svm /proc/cpuinfo`, é possível saber se o processador tem suporte a virtualização. Após a execução dos comandos, caso seja obtido algum resultado em vermelho, isso indica que o processador tem suporte a virtualização, possibilitando utilizar o KVM como *hypervisor*.

Comparativo	Máquina 1	Máquina 2
Equipamento	Servidor Dell PowerEdge T110	Notebook Dell Inspiron 1525
CPU	Intel Pentium G6950 – 2.8GHZ	Intel Core 2 Duo P8600 – 2.4 GHZ
Memória	4 GB Memória DDR 2	3 GB Memória DDR 2
Disco Rígido	250 GB 7.2k RPM	320 GB 5.4k RPM
Sistema Operacional	Fedora 13 - 64 Bits	Fedora 13 - 64 Bits
Libvirt	0.8.1 de Abril/2010	0.8.1 de Abril/2010
QEMU / KVM	0.12.4 de Maio/2010	0.12.4 de Maio/2010
Nome DNS Local	base.localdomain	pos.localdomain
Endereço IP	192.168.0.3	192.168.0.6
Conectividade	Cabo de Rede Cat5 – 10/100 Mbps	Wireles 802.11g – 54Mbps
Acesso ao Disco Virtual	Servidor NFS neste equipamento	Acesso ao NFS da máquina 1

Tabela 1. Tabela comparativa

Neste primeiro experimento foi utilizada a ferramenta de gerenciamento *Virsh*. Com ela é possível gerenciar todas as outras soluções de virtualização às quais a biblioteca *libvirt* dá suporte, como *virsh*, *Virt-Manager* e *Virt-Viewer*. Para acessar a ferramenta *virsh*, basta digitar *virsh* no terminal ou em modo console e depois inserir o comando desejado.

A Figura 7 ilustra a ferramenta *Virt-Manager* integrando vários servidores com suas máquinas virtuais.

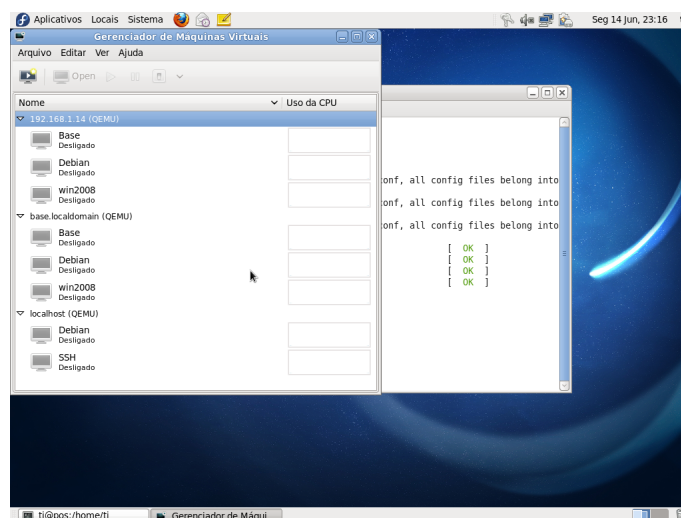


Figura 7. Virt-Manager

Os experimentos foram conduzidos no seguinte contexto: no ambiente considerado, após às 18 horas o nível de acesso aos servidores de E-Mail e FTP era baixo, com ociosidade; porém, as máquinas continuavam ligadas 24 horas. Desta forma, uma solução para economizar energia, para melhorar a utilização do *hardware* e reduzir a dissipação de calor foi a virtualização. Para tanto, após às 18 horas, as máquinas virtuais passaram a ser migradas para uma mesma máquina física e, após a migração concluída, a máquina original era desligada. Com isso, os serviços providos não sofriam nenhuma parada e obtinha-se um melhor aproveitamento dos recursos disponíveis. Como o volume de informações após as 7 horas cresce, os servidores de E-mail e FTP são migrados novamente

para equipamentos distintos. Desta forma, obtém-se uma solução simples para um balanceamento de serviços com servidores virtualizados. Obviamente, esta solução pode ser adotada para outros servidores desejados, bastando apenas os mesmos serem convertidos em máquinas virtuais, como mencionado no tópico Virtualização em Ambientes Virtualizados. Para executar as migrações foram executados os seguintes comandos:

1. `virsh connect qemu:///system` (comando executado na máquina 1)
2. `list -all` (apenas para verificar o estado atual das máquinas virtuais)
3. `migrate Base qemu+ssh://maquina2/system`
4. `migrate Debian qemu+ssh://maquina2/system`

Para que que o *Live Migration* fosse concluído com sucesso, foi necessário editar o arquivo *hosts* das máquinas 1 e 2, onde foi inserido o nome e IP de ambos, para que a tradução dos nomes fosse feita por este modo. Também foi necessária a liberação no firewall do seguinte *range* de portas 49152 a 49215 que utilizam o protocolo TCP, para as transmissões de dados associadas à migração.

A Figura 8 ilustra o momento em que a migração se inicia entre as máquinas físicas. Pode-se visualizar também que em um equipamento a máquina virtual Base está em execução e no outro equipamento a mesma máquina virtual se encontra em pausa.

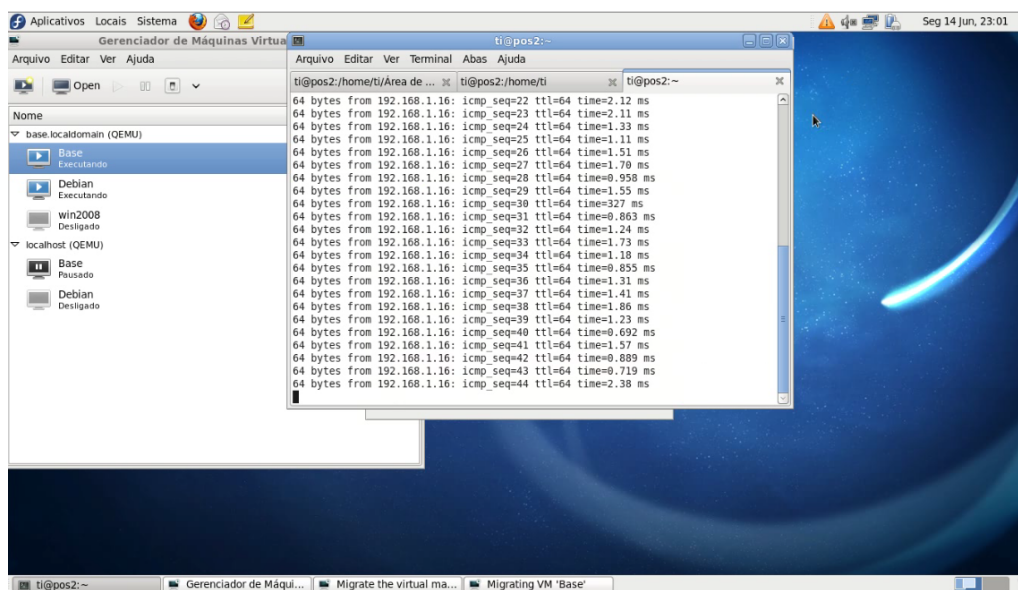


Figura 8. Momento da migração

Foi testada também a migração através da ferramenta VirtualBox, no intuito de demonstrar que novas ferramentas estão optando por suportar o método *live migration*. Para esta solução de virtualização é possível também utilizar a ferramenta *virsh* para fazer todo o gerenciamento, porém a ferramenta utilizada neste experimento foi a *VBoxManage*. Para executar a migração entre as máquinas físicas foram utilizados os seguintes comandos:

1. `VBoxManage modifyvm Fedora -teleporter on`
`-teleporterport 6000` (Executado na Máquina 2)

2. `VboxManage start Fedora` (Executado na Máquina 2)
3. `VboxManage start Fedora` (Executado na Máquina 1)
4. `VboxManage controlvm Fedora teleport -maquina2 -port 6000` (Executado na Máquina 1)

Através dos experimentos executados foi mensurado o tempo médio de migração de uma máquina virtual para outra máquina física, que ocorreu em 30 segundos, e a migração simultânea de duas máquinas virtuais para outra máquina física, que concluiu-se após 45 segundos em média. Estes resultados comprovam que a migração das máquinas virtuais é totalmente possível em um curto espaço de tempo (ZOTTI, 2010b).

6. Trabalhos Relacionados

Existem vários trabalhos na literatura onde são apresentadas experiências de soluções em virtualização. Abaixo é apresentada uma análise crítica e comparação das características entre alguns trabalhos que se assemelham ao estudo apresentado.

Em (NABHEN, 2006), são apresentados experimentos que comprovam a possibilidade de se criar um ambiente virtualizado para a prática de exercícios em laboratório, sem adequação da topologia de rede e acesso privilegiado às configurações do sistema operacional de cada computador. Para a solução do problema levantado pelo autor, foi adotada a solução de virtualização User-Mode-Linux (UML). Não foram executados experimentos onde as máquinas virtualizadas fossem migradas por exemplo para um notebook de um aluno onde o mesmo pudesse levar sua VM para continuar seus estudos. Porém, o trabalho mostra que é possível criar um ambiente de rede virtualizado com facilidade utilizando a tecnologia de virtualização. O trabalho (KOSLOVSKI et al., 2006) analisa duas ferramentas de virtualização, VMware e Xen, a fim de disponibilizar maiores informações para os administradores de rede. Em vários momentos este trabalho levantou a possibilidade do uso de migração através do modo *Live Migration*, porém não fez a aplicação de nenhuma ferramenta para comprovar o relatado. Uma abordagem mais parecida com o problema aqui tratado foi desenvolvido por (MORA, 2008). Neste trabalho é tratado o balanceamento de carga de servidores virtualizados onde o foco se resume ao balanceamento automático das máquinas virtualizadas sem a intervenção de um administrador do sistema. Este trabalho realizou experimentos com a ferramenta XEN que possui de forma nativa a migração em *Live Migration*. Para se balancear as máquinas virtuais foi desenvolvido um *script* pelo autor, nominado *VMbal*, que coleta informações dos servidores através dos *daemons* que são executados através da biblioteca *libvirt* em cada servidor e faz comparações para saber para qual servidor a máquina virtual será migrada. Os testes realizados duraram pouco tempo o que deixa os testes com uma margem de erro, já que o autor ressaltou que através dos testes foi detectada uma falha não identificada no *hypervisor Xen*, onde o mesmo não consegue concluir a migração após um certo tempo.

Os experimentos realizados por este trabalho trazem outra abordagem, pois foram desenvolvidos utilizando a ferramenta KVM/QEMU, que tem seu uso simples e que pode estender o gerenciamento a um nível avançado. Foram realizados também experimentos que comprovam a funcionalidade da função *Live Migration* em sistemas menos complexos, que é o caso da ferramenta VirtualBox. De maneira igual ao último trabalho citado, foram executados experimentos utilizando para gerenciamento do *hypervisor* a biblioteca *libvirt*, que disponibiliza e facilita todo o gerenciamento para o administrador do sistema. O diferencial deste trabalho é que a migração ocorreu a partir de uma ação de um VMM.

7. Conclusões

Os experimentos comprovam que é possível realizar a migração através do modo *live migration* sem nenhuma parada dos serviços executados pela máquina virtualizada. Com isso, o balanceamento dos serviços virtualizados pode ocorrer, mantendo suas disponibilidades enquanto a migração ocorre e após sua conclusão. Um vídeo realizado durante a migração comprova a viabilidade dessa abordagem (ZOTTI, 2010a).

Foram realizados experimentos que comprovaram a facilidade de se balancear e aumentar a disponibilidade de serviços em servidores virtualizados, através da função *Migrate* do KVM/QEMU, onde com apenas dois servidores físicos, foi possível criar um pool de servidores virtualizados capaz de criar e gerenciar várias máquinas virtuais.

Este trabalho concluiu que o balanceamento de serviços em servidores virtualizados é uma boa opção para solucionar problemas relacionados à carga de serviços virtualizados, sendo que todos os experimentos realizados se mostraram efetivos.

Todos os passos que os experimentos descrevem foram filmados, para que, a partir da divulgação deste conteúdo, outras pessoas possam desenvolver projetos ou trabalhos com esta tecnologia de migração (ZOTTI, 2010a).

Este trabalho traz alguns experimentos básicos de balanceamento de serviços em servidores virtualizados, porém possibilita futuras expansões das funcionalidades desta solução. Algumas propostas para trabalhos futuros incluem:

- Migração automática das VMs quando a máquina física estiver com poucos recursos disponíveis;
- Implementar algoritmos para análise de tráfego de rede, visando ao balanceamento dos serviços quando algum caminho estiver sobrecarregado;
- Implementar algoritmos para análise de uso de memória, CPU e disco rígido;
- Implementar algoritmos para comparar o desempenho entre as soluções de virtualização;
- Implementar algoritmos para comparar o desempenho entre soluções de virtualização completa e para-virtualização;

Referências

BELLARD, F. *QEMU, Open Source Processor Emulator*. 2010. Acessado em: 11/08/2010. Disponível em: <http://wiki.qemu.org/Main_Page>.

BHATIA, N. *Performance Evaluation of AMD RVI Hardware Assist*. 2009. VMware, Inc. Acessado em: 11/08/2010. Disponível em: <<http://www.vmware.com/pdf/RVIperformance.pdf>>.

CHINNI, R. H. S. *Virtual Machine Device Queues*. [S.l.], 2007. Acessado em: 03/06/2010. Disponível em: <http://www.intel.com/technology/platform-technology/virtualization/VMDq_whitepaper.pdf>.

COMPUTERWORLD, E. B. Virtualização, eficiência sob medida. *COMPUTERWORLD*, v. 1, 2009. Acessado em: 11/08/2010. Disponível em: <<http://computerworld.uol.com.br/gestao/2009/03/25/virtualizacao-eficiencia-sob-medida>>.

CORPORATION, O. *User Manual*. [S.l.], 2010. Acessado em: 11/08/2010. Disponível em: <<http://www.virtualbox.org/manual/ch07.html>>.

- DEVICES, I. A. M. *AMD-V Nested Paging*. 2008. Acessado em: 11/08/2010. Disponível em: <<http://developer.amd.com/assets/NPT-WP-1%201-final-TM.pdf>>.
- DEVICES, I. A. M. Live migration with amd-v™ extended migration technology. *AMD Developer*, v. 1, 2008. Acessado em: 11/08/2010. Disponível em: <http://developer.amd.com/assets/43781-3.00-PUB_Live-Virtual-Machine-Migration-on-AMD-processors.pdf>.
- DEVICES, I. A. M. *AMD Virtualization (AMD-V) Technology*. 2010. Acessado em: 11/08/2010. Disponível em: <<http://sites.amd.com/us/business/it-solutions/virtualization/Pages/amd-v.aspx>>.
- DONALDSON, J. L. *Teaching operating systems in a virtual machine environment*. [S.l.], 1987.
- GARFINKEL, T. et al. *Terra: a virtual machine-based platform for trusted computing*. [S.l.]: ACM New York, NY, USA, 2003.
- GOLDBERG, R. P. Survey of virtual machine research. *IEEE Computer*, v. 1, 1974.
- HOLWERDA, T. *Sun Releases VirtualBox 3.1, Adds Teleportation Feature*. [S.l.], 2009. Acessado em: 11/08/2010. Disponível em: <<http://www.osnews.com/story/22557/>>.
- JONES, M. T. *An overview of virtualization methods, architectures, and implementations*. 2006. Acessado em: 24/06/2010. Disponível em: <<http://www.ibm.com/developerworks/linux/library/l-linuxvirt/>>.
- JONES, M. T. Emulação do sistema com o qemu. *DeveloperWorks Brasil*, v. 1, 2007. Acessado em: 11/08/2010. Disponível em: <<http://www.ibm.com/developerworks/br/library/l-qemu/>>.
- JONES, M. T. Anatomia de um hypervisor linux. *DeveloperWorks Brasil*, 2009. Acessado em: 11/08/2010. Disponível em: <<http://www.ibm.com/developerworks/br/library/l-hypervisor/>>.
- KOSLOVSKI, G. P.; BOUFLEUR, M. P.; CHARÃO, A. S. *Uso de Virtualização de Recursos Computacionais na Administração de Redes*. 2006.
- KVM. *Migration using Kernel Base Virtual Machine (KVM)*. 2010. Acessado em: 14/06/2010. Disponível em: <<http://www.linux-kvm.org/page/Migration>>.
- LIBVIRT. *libvirt Virtualization API*. 2010. 03/06/2010. Disponível em: <<http://libvirt.org/goals.html>>.
- LIBVIRT. *Remote support*. 2010. Acessado em: 11/08/2010. Disponível em: <<http://libvirt.org/remote.html>>.
- MORA, G. F. *BALANCEAMENTO DE CARGA DE SERVIDORES VIRTUALIZADOS*. 12 2008. UNIVERSIDADE FEDERAL DE SANTA MARIA.
- NABHEN, C. M. R. Experiências no uso de máquinas virtuais para o ensino de redes de computadores. In: *XIV Workshop sobre Educação em Computação*. [S.l.: s.n.], 2006.
- NEIGER, G. et al. Intel® virtualization technology. *Intel® Technology Journal*, v. 1, 2006. Acessado em: 11/08/2010. Disponível em: <http://www.intel.com/technology/itj-/2006/v10i3/1-hardware/1-abstract.htm?iid=tech_vt_tech+itj>.

- OLIVEIRA, G. V. N. *Solução de virtualização completa utilizando VMware e Software Livre: Um Estudo de Caso na CEF*. 2007. Acessado em: 11/08/2010.
- ORACLE. *The VirtualBox architecture*. 2009. Oracle Magazine. Acessado em: 11/08/2010. Disponível em: <<http://www.virtualbox.org/wiki/VirtualBox>>.
- ORACLE. *The VirtualBox architecture*. 2010. Oracle Magazine. Acessado em: 11/08/2010. Disponível em: <<http://www.virtualbox.org/wiki/VirtualBox%20architecture>>.
- RITZAU, T.; WARNKE, R. *QEMU-KVM and libvirt*. [S.l.]: Books on Demand GmbH, Norderstedt, 2010.
- ROSENBLUM, M. *The Reincarnation of Virtual Machines*. [S.l.]: ACM New York, NY, USA, 2004.
- SYSTEMS, I. C. *The XenServer Product Family*. 2005. Acessado em: 04/07/2010. Disponível em: <http://www.intersales.de/fileadmin/usr/xen/docs/citrix_xenserver_whitepaper.pdf>.
- TUXRADAR. *Howto: Linux and Windows virtualization with KVM and Qemu*. [S.l.], 2010. Acessado em: 04/07/2010. Disponível em: <<http://www.tuxradar.com/content/howto-linux-and-windows-virtualization-kvm-and-qemu>>.
- VMWARE. *VMware Savings Calculator*. 2010. VMware Inc. Acessado em: 11/08/2010. Disponível em: <http://www.vmwareyourtime.com/nam/?src=WWW_09Q3_VMW_SMBT_CAMPAIGN-HOME-ASSESS>.
- VMWARE, I. *Virtualization Overview*. 2008. Acessado em: 11/08/2010. Disponível em: <<http://www.vmware.com/pdf/virtualization.pdf>>.
- VMWARE, I. *Getting Started with VMware Fusion*. [S.l.], 2009. Acessado em: 11/08/2010. Disponível em: <http://www.vmware.com/pdf/fusion_getting_started_30.pdf>.
- VMWARE, I. *BRING EFFICIENCY, CONTROL AND FLEXIBILITY TO YOUR IT INFRASTRUCTURE*. 2010. Acessado em: 11/08/2010. Disponível em: <<http://www.vmware.com/products/>>.
- VMWARE, I. *Proteção e eficácia - benefícios da Virtualização em PMEs*. 01 2010. Acessado em: 03/06/2010. Disponível em: <http://www.timaster.com.br/revista/materias/main_materia.asp?codigo=1718>.
- XENSOURCE, I. *XenServer Installation Guide*. Release 4.0.1. [S.l.], 2007. Acessado em: 11/08/2010. Disponível em: <<http://docs.vmd.citrix.com/XenServer/4.0.1/installation/ch01s03.html>>.
- ZOTTI, V. L. S. *Migrate KVM/QEMU - Visão externa*. 2010. Youtube. Acessado em: 11/08/2010. Disponível em: <<http://www.youtube.com/watch?v=ItX2AQ3Ad5o>>.
- ZOTTI, V. L. S. *Migrate KVM/QEMU via Console*. 2010. Youtube. Acessado em: 26/09/2010. Disponível em: <<http://www.youtube.com/watch?v=HdyAyMuRAn0>>.