

**Vitor de Deus Miranda**

**Análise comparativa entre as ferramentas VirtualBox, VMware Workstation  
e QEMU em um ambiente Linux**

Monografia de Pós-Graduação “*Lato Sensu*”  
apresentada ao Departamento de Ciência da  
Computação para obtenção do título de Especialista  
em “Administração em Redes Linux”

Orientador  
Prof. D.Sc. Joaquim Quinteiro Uchôa

Lavras  
Minas Gerais - Brasil  
2010



**Vitor de Deus Miranda**

**Análise comparativa entre as ferramentas VirtualBox, VMware Workstation  
e QEMU em um ambiente Linux**

Monografia de Pós-Graduação “*Lato Sensu*”  
apresentada ao Departamento de Ciência da  
Computação para obtenção do título de Especialista  
em “Administração em Redes Linux”

*Aprovada em 24 de Abril de 2010*

---

Prof. Msc. Herlon Ayres de Camargo

---

Prof. Samuel Pereira Dias

---

Prof. D.Sc. Joaquim Quinteiro Uchôa  
(Orientador)

Lavras  
Minas Gerais - Brasil  
2010



*Dedico esta monografia aos meus pais.*



## **Agradecimentos**

Agradeço aos meus pais, familiares e amigos que diretamente ou indiretamente me apoiaram no desenvolvimento deste trabalho. Aos professores do curso ARL pelos ensinamentos compartilhados e em especial ao Joaquim pelo direcionamento.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Virtualização</b>	<b>3</b>
2.1	Tipos de Virtualização . . . . .	4
2.1.1	Emulação de Hardware . . . . .	4
2.1.2	Virtualização Completa . . . . .	4
2.1.3	Paravirtualização . . . . .	5
2.1.4	A camada VMM - <i>Virtual Machine Monitor</i> . . . . .	6
<b>3</b>	<b>As Ferramentas de Virtualização</b>	<b>9</b>
3.1	O QEMU . . . . .	9
3.1.1	Instalação do QEMU no Linux . . . . .	10
3.2	O VirtualBox . . . . .	11
3.2.1	Instalação do VirtualBox no Linux . . . . .	13
3.3	O VMware Workstation . . . . .	14
3.3.1	Instalação do VMware Workstation no Linux . . . . .	14
<b>4</b>	<b>Benchmark</b>	<b>19</b>
4.1	Requisitos de um Benchmark . . . . .	19
4.2	Carga de Trabalho e Métricas . . . . .	20



4.3	O Phoronix Test Suite . . . . .	21
<b>5</b>	<b>Análise entre as ferramentas</b>	<b>27</b>
5.1	Execução dos Testes . . . . .	27
5.2	Resultados dos Testes . . . . .	28
5.2.1	Resultados com o 7zip-Compress . . . . .	28
5.2.2	Resultados com o LZMA-Compress . . . . .	28
5.2.3	Resultados com o IOzone . . . . .	29
5.2.4	Resultados com o SQLite . . . . .	29
5.2.5	Resultados com o BYTE Unix Bench . . . . .	30
5.3	Discussão dos Testes . . . . .	31
<b>6</b>	<b>Conclusão</b>	<b>33</b>

# Lista de Figuras

2.1	Alocação de memória para o sistema convidado. Adaptada de (LAUREANO, 2006). . . . .	5
2.2	Relacionamento entre o VMM e as máquinas virtuais (ROSE, 2004). . . . .	6
2.3	Máquina virtual - Tipo I . . . . .	7
2.4	Máquina virtual - Tipo II . . . . .	7
3.1	Sequência de comandos para instalação do QEMU e adicionais. . . . .	11
3.2	Interface de gerenciamento de máquinas virtuais no QEMU com Qemu Launcher. . . . .	12
3.3	Interface de gerenciamento de máquinas virtuais no VirtualBox. . . . .	13
3.4	Comando para instalação do VirtualBox. . . . .	14
3.5	Nomenclatura do arquivos de instalação do VMware Workstation . . . . .	15
3.6	Comando para instalação do VMware Workstation. . . . .	15
3.7	Tela de instalação do VMware Workstation. . . . .	16
3.8	VMware Player. . . . .	17
3.9	VMware Workstation. . . . .	18
4.1	Resultado de um teste com o phoronix, modo texto. . . . .	22
4.2	Relatório do resultado de um teste com o phoronix, via navegador. . . . .	23
4.3	Comando para instalação dos testes para o Phoronix. . . . .	25

4.4	Processo de instalação dos testes para o Phoronix. . . . .	25
5.1	7-zip compress. . . . .	28
5.2	LZMA compress. . . . .	29
5.3	IOzone 512MB - Escrita. . . . .	30
5.4	IOzone 512MB - Leitura. . . . .	30
5.5	SQLite. . . . .	31
5.6	BYTE Unix Bench. . . . .	31

## Resumo

A virtualização é uma estratégia para otimizar a utilização de recursos computacionais. Diversos pesquisadores vêm produzindo trabalhos nesse campo de atividade. O objetivo deste documento é apresentar uma análise comparativa quantitativa entre três dos diversos hipervisores disponíveis, o VirtualBox, o VMware Workstation e o QEMU. Também são analisadas as principais técnicas de virtualização a emulação de *hardware*, a virtualização completa e a paravirtualização. O processo de mensuração de desempenho baseia-se na utilização de uma *benchmark suite*, o Phoronix-Test-Suite, que permite produzir resultados comparativos de maneira fácil, rápida e repetível. É compreendido a diferença de arquitetura entre os hipervisores e os efeitos nos resultados aqui apresentados, portanto servirá de apoio informal para que usuários possam tomar suas decisões na hora da escolha de um hipervisor.

**Palavras-Chave:** Virtualização; VirtualBox; VMware Workstation; QEMU; Benchmark; Phoronix Test Suite.

# Capítulo 1

## Introdução

Os computadores modernos estão suficientemente poderosos para utilizar o conceito de virtualização e apresentar a ilusão de várias máquinas virtuais sendo executadas simultaneamente, cada uma rodando um sistema operacional diferente. Isto levou ao ressurgimento do interesse na tecnologia de máquinas virtuais (BARHAM *et al.*, 2003).

Este trabalho apresenta uma análise comparativa quantitativa de três hipervisores, o VirtualBox, VMware Workstation e QEMU. Essas ferramentas foram testadas em um ambiente Linux, portanto, tanto no sistema operacional hospedeiro quanto nas máquinas virtuais foram utilizadas a distribuição Ubuntu. Juntamente com o QEMU foi aplicado o módulo acelerador KQEMU para melhorias no desempenho.

É de conhecimento do autor a existência de soluções de aceleração para o QEMU, como o KVM (Kernel-based Virtual Machine). Porém, a mesma não foi contemplada por limitações de *hardware* e ausência das extensões de virtualização (Intel VT ou AMD-V) na máquina utilizada para efetuar a comparação.

A motivação do desenvolvimento deste trabalho surgiu no período em que prestava serviços de tecnologia na Gerência de Suporte Tecnológico da Caixa Econômica Federal no estado do Espírito Santo. Nessa época, houve o primeiro contato do autor com a ferramenta de virtualização VMware ESX. Em tal ambiente optou-se pela consolidação de servidores bancários e diversas outras máquinas do departamento de tecnologia da informação. O constante crescimento e utilização do termo virtualização, juntamente com a escassez de trabalhos que trazem a

comparação entre os três hipervisores propostos aqui, despertaram o interesse pela realização desta pesquisa.

O objetivo deste trabalho é testar as ferramentas de virtualização VirtualBox, Vmware Workstation e QEMU em um ambiente Linux e realizar uma análise comparativa entre elas. Essa análise servirá como apoio informal a usuários na escolha de uma ferramenta de virtualização, seja por suas características e ou pelo desempenho apresentado. O desempenho dos diferentes hipervisores pode ser testado utilizando *benchmarking* e mensurado através de métricas e testes, que serão apresentados no decorrer dos capítulos com a utilização do Phoronix-Test-Suite<sup>1</sup>.

Quanto à organização deste trabalho, o mesmo está dividido em 6 capítulos: o Capítulo 2 apresenta uma breve descrição das principais técnicas de virtualização atualmente utilizadas, a emulação de *hardware*, a virtualização completa e a para-virtualização. O Capítulo 3 apresenta o histórico das ferramentas de virtualização (o VirtualBox, o VMware Workstation e o QEMU) e seus respectivos procedimentos de instalação. No Capítulo 4, há uma explicação sobre o *benchmark*, seus requisitos, carga de trabalho e métricas, além da descrição do Phoronix-Test-Suite. Já no Capítulo 5 dispomos a análise entre os hipervisores e os testes utilizados, assim como os resultados finais da avaliação de desempenho. E para concluir, o Capítulo 6 apresenta comentários e observações finais.

---

<sup>1</sup><http://www.phoronix-test-suite.com>.

## Capítulo 2

# Virtualização

A utilização do termo virtualização não é recente, e já vem sendo utilizado há algum tempo. Inicialmente as máquinas virtuais foram desenvolvidas para serem utilizadas nos ambientes VM/370 da IBM<sup>1</sup> com o objetivo de uma centralização dos sistemas em que onde cada máquina virtual simulava uma réplica da máquina real, causando assim a idéia de uso exclusivo pelos usuários (LAUREANO, 2006).

O avanço dos processadores e os *hardwares* em geral está permitindo que computadores pessoais sirvam como hospedeiros de máquinas virtuais, o que aumenta a flexibilidade na utilização de outras plataformas, auxiliando no conhecimento, em testes e no desenvolvimento de soluções.

Vale lembrar a diferença entre o sistema operacional hospedeiro (*Host Operating System*) e o sistema operacional convidado (*Guest Operating System*). O primeiro é o que roda diretamente sobre o *hardware* real da máquina e que servirá como base para o processo de virtualização. Já o segundo é o sistema que será executado sobre o *hardware* virtualizado, ou seja, as máquinas virtuais, o que possibilita ter vários sistemas operacionais diferentes sendo executados ao mesmo tempo.

A virtualização em um ambiente corporativo quase sempre remete à consolidação de servidores em que uma máquina de maior capacidade é responsável por hospedar diversas máquinas virtuais destinadas a atender os mais variados tipos de serviços (MATTOS, 2008).

---

<sup>1</sup><http://www.ibm.com/>.

Na Caixa Econômica Federal, este autor esteve envolvido diretamente no processo da experiência de consolidação de servidores. Nessa Empresa, servidores defasados e alocados de forma descentralizada foram convertidos em máquinas virtuais, garantindo gerenciamento centralizado, flexibilidade, redução dos custos de administração de TI, além do melhor aproveitamentos dos recursos do servidor.

## 2.1 Tipos de Virtualização

Este tópico apresenta uma breve descrição das principais técnicas de virtualização atualmente utilizadas, dentre elas a emulação de *hardware* (*hardware emulation*), a virtualização completa (*full virtualization*) e a paravirtualização (*paravirtualization*).

### 2.1.1 Emulação de Hardware

Os emuladores são considerados bastante complexos, pois tendem a simular praticamente todas as particularidades do *hardware*.

São fortes aliados aos desenvolvedores de *firmware* de *hardware*, pois permitem a validação de aplicativos sem a presença de um *hardware* real (OLIVEIRA, 2007). Por outro lado, apresentam um baixo desempenho, já que perdem a eficiência ao interpretar ou traduzir as instruções da máquina real (LAUREANO, 2006).

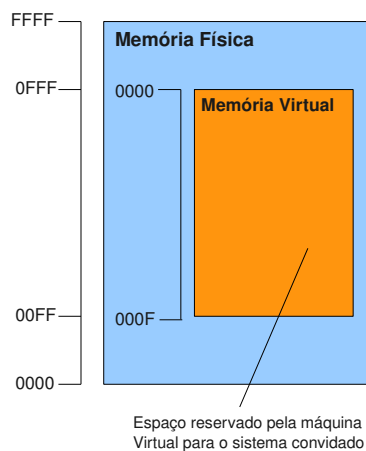
### 2.1.2 Virtualização Completa

Com a virtualização completa é fornecido uma réplica virtual do *hardware*, portanto, o sistema operacional e os *softwares* poderão ser executados na máquina virtual da mesma maneira que seriam sobre o *hardware* original (ROSE, 2004). O principal benefício deste tipo de virtualização é que o sistema convidado, aquele que será instalado na máquina virtual, não necessita de qualquer modificação em sua estrutura. Também constatado o melhor isolamento, segurança e simplificação no processo de migração de um sistema convidado para outra instância virtual ou execução nativa em um *hardware* (VMWARE, 2007).

Porém, conforme Laureano (LAUREANO, 2006), a desvantagem na virtualização completa é a perda de desempenho para o gerenciamento de memória e disco. Ao ser instanciado é reservado um espaço de memória pelo monitor para



o trabalho, no qual o sistema convidado acredita estar trabalhando com o início da memória. Observe a Figura 2.1. Aqui, a máquina real utiliza o endereço de memória compreendido entre 0000 e FFFF. Para a máquina virtual é reservado o endereço de 00FF até 0FFF, que por sua vez no sistema convidado é reservado como endereço total compreendido de 000F à 0000. Assim, para que o sistema convidado acesse a sua área de memória reservada, o monitor precisa realizar a "tradução" para acessar a memória real.



**Figura 2.1:** Alocação de memória para o sistema convidado. Adaptada de (LAUREANO, 2006).

O mesmo tipo de tradução descrito acima ocorre para o acesso a disco.

### 2.1.3 Paravirtualização

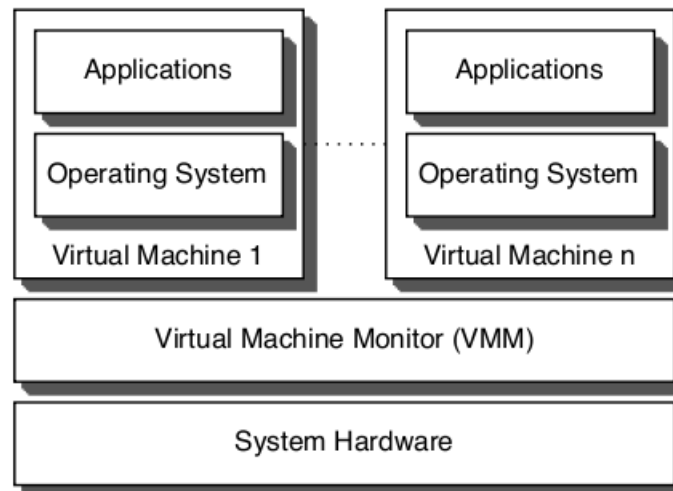
Diferente da virtualização completa, na paravirtualização, o sistema convidado necessariamente deve ser modificado para interagir com o monitor de máquinas virtuais. Apesar dessa modificação reduzir a portabilidade do sistema, o sistema convidado consegue acessar recursos do *hardware* de forma direta, sem que precise passar pelo processo de tradução anteriormente descrito. O ganho de desempenho é o principal fator para que se opte por utilizar a paravirtualização (LAUREANO, 2006).

Na paravirtualização, os dispositivos *hardware* são utilizados em sua capacidade total, pois são acessados por *drivers* da própria máquina virtual, o que dispensa a utilização de *drivers* genéricos para o seu funcionamento.

Mesmo com as vantagens descritas anteriormente, é a presença de instruções pertinentes a virtualização nos processadores da Intel<sup>2</sup> e AMD<sup>3</sup> apelidados de IVT (*Intel Virtualization Technology*), codinome Vanderpool e AMD-V (*AMD-Virtualization*), codinome Pacífica que vem representando um ganho de desempenho significativo na virtualização total (MATTOS, 2008).

#### 2.1.4 A camada VMM - *Virtual Machine Monitor*

A VMM (*Virtual Machine Monitor*) ou também conhecido como hipervisor é uma camada que abstrai todos os recursos (processadores, memória, rede, entre outros) e os aloca para as várias máquinas virtuais que rodam sobre o VMM. Por prover uma abstração, permite executar diversas máquinas virtuais no mesmo sistema (ROSE, 2004). Observa-se na Figura 2.2 o relacionamento entre o VMM e as máquinas virtuais.



**Figura 2.2:** Relacionamento entre o VMM e as máquinas virtuais (ROSE, 2004).

Em um ambiente de máquinas virtuais, o VMM roda no modo supervisor (modo onde instruções podem ser definidas e executadas) enquanto as máquinas virtuais rodam no modo usuário (MENASCE, 2005). Portanto, quando uma

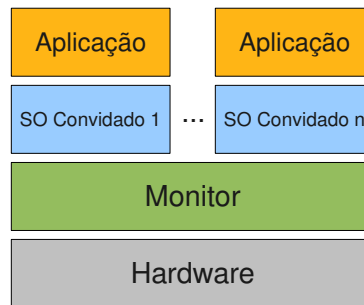
---

<sup>2</sup><http://www.intel.com/>.

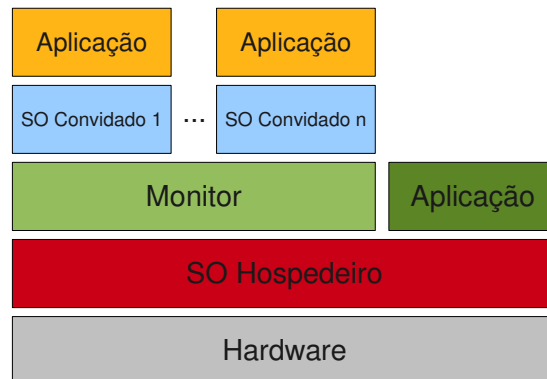
<sup>3</sup><http://www.amd.com/>.

máquina virtual executa uma instrução privilegiada é gerada uma interrupção e o VMM acaba por controlar o acesso aos recursos compartilhados.

No processo de construção de sistemas de máquinas virtuais o hipervisor pode atuar de duas maneiras. No tipo I, onde o hipervisor é implementado diretamente sobre a camada de *hardware*, tendo logo acima os sistemas convidados e suas aplicações. No tipo II o hipervisor é implementado sobre o sistema hospedeiro, funcionando como um processo e tendo subsequentemente os sistemas convidados. As Figuras 2.3 e 2.4 representam visualmente os tipos de máquinas virtuais I e II respectivamente.



**Figura 2.3:** Máquina virtual - Tipo I



**Figura 2.4:** Máquina virtual - Tipo II



## Capítulo 3

# As Ferramentas de Virtualização

Neste capítulo, apresentamos as ferramentas de virtualização, QEMU, VirtualBox e VMware Workstation, escolhidas para o processo de comparação. Após a apresentação de cada ferramenta será demonstrado o processo de instalação.

As ferramentas de virtualização foram instaladas em uma estação de trabalho com processador Intel Core 2 Duo E4500 2.20 Ghz, com 3 Gb de memória RAM e um disco rígido SATA com 250 Gb de capacidade. A distribuição Linux escolhida para ser o sistema hospedeiro é o Ubuntu 9.10 apelidado de Karmic Koala, lançado em outubro de 2009 e com suporte até abril de 2011. A versão utilizada do Kernel é a 2.6.31-17-generic-pae.

### 3.1 O QEMU

O QEMU é um emulador de processador que para obter um bom desempenho na emulação utiliza a técnica de tradução dinâmica (WIKIQEMU, 2010). O sistema operacional que será executado na máquina virtual não precisa sofrer nenhum tipo de modificação específica, sendo possível executar diversos sistemas operacionais, tais como Microsoft Windows, Linux ou Mac OS X<sup>1</sup>. Durante a execução do SO e suas aplicações, o QEMU pode ser utilizado como uma ferramenta para auxiliar no *debugging*, pois as máquinas virtuais podem ser facilmente pausadas para serem inspecionadas, salvas e restauradas. O QEMU está licenciado sob as regras da *GNU General Public License - GPL* versão 2.

---

<sup>1</sup><http://www.apple.com/br/macosex/>.

Segundo Bellard, com o QEMU é possível trabalhar em dois modos operacionais: a emulação total do sistema e a emulação no modo de usuário. No primeiro modo, é possível uma emulação de um sistema integral, incluindo o processador e periféricos (BELLARD, 2005). No modo usuário, disponível somente para o sistema Linux, é possível executar um processo contruído para uma determinada CPU em outra. Portanto, um processo Linux compilado em uma plataforma PowerPC poderá ser executado na X86, e vice-versa.

Um bom desempenho pode ser alcançado utilizando o acelerador de QEMU (o módulo Kernel QEMU - KQEMU<sup>2</sup>, se utilizado para emular um sistema x86 em um sistema hospedeiro também x86. O KVM é *open source* e trabalha na versão 2.6.20, ou superior, do *Kernel*. O desenvolvimento do KVM e QEMU é unificado, logo, ele costuma ser descrito em conjunto com o QEMU e muitas distribuições Linux disponibilizam pacotes "qemu-kvm".

### 3.1.1 Instalação do QEMU no Linux

A instalação do QEMU pode ser feita através de binários para as distribuições Linux específicas ou através da compilação dos códigos-fontes. Os arquivos de instalação podem ser encontrados no endereço eletrônico <http://wiki.qemu.org/Download><sup>3</sup>. Como o sistema hospedeiro escolhido é o Ubuntu optou-se por misturar entre a utilização binários através da ferramenta apt-get e arquivos fontes.

Na Figura 3.1, observa-se a sequência de comandos para a instalação do QEMU e auxiliares. Na primeira linha encontra-se a instalação do QEMU (versão 0.11), do build-essential (este pacote contém uma lista informativa de pacotes que são considerados essenciais para a construção de pacotes Debian) e o *headers* do *kernel* para a versão específica do *kernel* do sistema hospedeiro. Na segunda linha, há a obtenção do *kqemu-1.4* e sequentemente a descompactação, configuração, compilação, instalação e carregamento do módulo acelerador observados nas linhas de número 3, 4, 5, 6, 7 e 8.

Para facilitar a utilização do QEMU foi instalado o Qemu Launcher (versão 1.7.4)<sup>4</sup>, uma interface gráfica que permite criar, salvar, configurar e executar máquinas virtuais através do QEMU. O procedimento de instalação é simples e pode ser feito através do comando *sudo apt-get install qemu-launcher*. Observa-se na

---

<sup>2</sup>Apesar do autor ter conhecimento que o KQEMU é um módulo obsoleto em relação ao KVM (Kernel-based Virtual Machine).

<sup>3</sup>Acessado em 23 de Janeiro de 2010.

<sup>4</sup><http://gna.org/projects/qemulaunch/>.

```
1. sudo apt-get install qemu build-essential linux-headers-$(uname -r)
2. wget http://wiki.qemu.org/download/kqemu-1.4.0pre1.tar.gz
3. tar -xzf kqemu-1.4.0pre1.tar.gz
4. cd kqemu-1.4.0pre1.tar.gz
5. ./configure --prefix=/usr --kernel-path=/lib/modules/$(uname -r)/build/
6. make
7. sudo ./install.sh
8. sudo modprobe kqemu
```

**Figura 3.1:** Sequência de comandos para instalação do QEMU e adicionais.

Figura 3.2 a configuração e manipulação de máquinas virtuais através do Qemu Launcher.

## 3.2 O VirtualBox

O VirtualBox é uma ferramenta de virtualização originalmente desenvolvida pela empresa alemã Innotek. Posteriormente, foi adquirida pela Sun<sup>5</sup>, que por sua vez em 2010 tornou-se aquisição da Oracle Corporation<sup>6</sup>, recebendo assim, uma nova nomenclatura: Oracle VM VirtualBox.

Pode ter como sistemas operacionais hospedeiros o Microsoft Windows, o Mac OS, o Linux e o Solaris. Como sistema convidado cita-se Microsoft Windows, Linux, Solaris, OpenSolaris, FreeBSD, OpenBSD e OS2 (SUN MICROSYSTEMS, 2010). Da mesma forma que outras ferramentas de virtualização as máquinas virtuais podem ser inicializadas, pausadas e paradas.

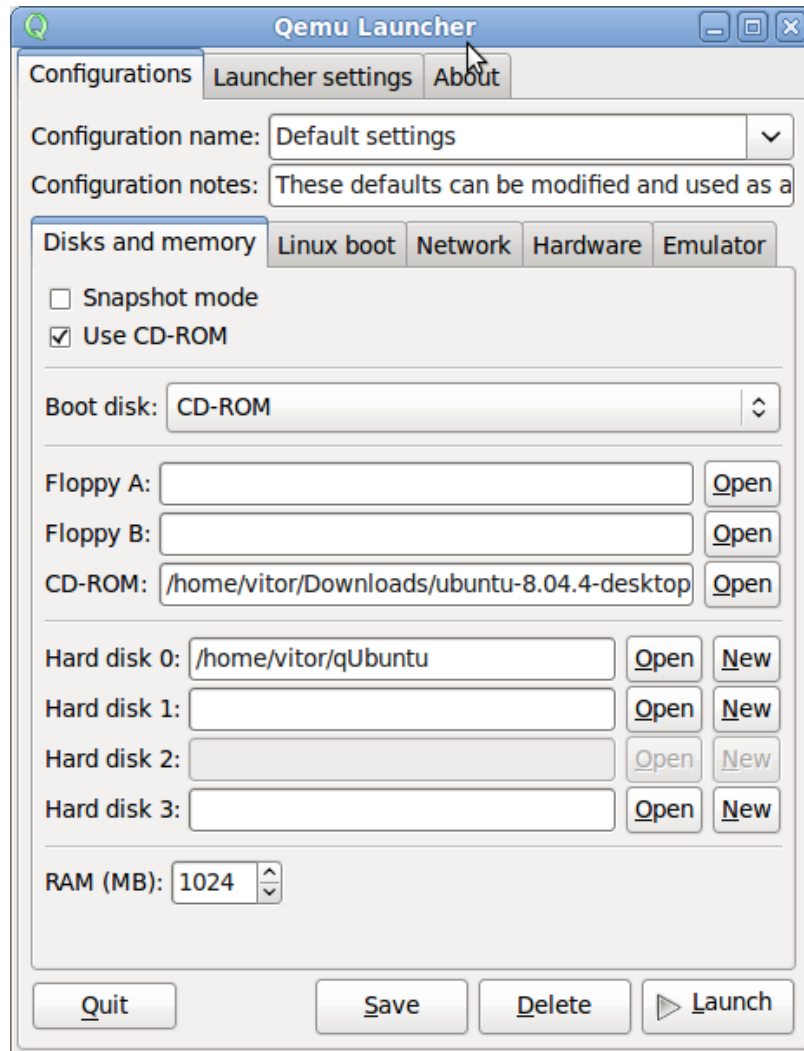
Está licenciado sob uma licença proprietária para uso comercial, caso a utilização seja pessoal e educacional será utilizada a PUEL - *Personal Use and Educational License* (PUEL, 2008). Uma outra opção de licenciamento é a *VirtualBox Open Source Edition* - OSE, um programa livre regido sobre as regras *GNU General Public License* - GPL. Neste último licenciamento duas funcionalidades serão reprimidas, o suporte ao *Remote Desktop Protocol* - RDP<sup>7</sup> e o suporte a dispositivos USB.

---

<sup>5</sup><http://br.sun.com>.

<sup>6</sup><http://www.oracle.com>.

<sup>7</sup>[http://en.wikipedia.org/wiki/Remote\\_Desktop\\_Protocol](http://en.wikipedia.org/wiki/Remote_Desktop_Protocol).



**Figura 3.2:** Interface de gerenciamento de máquinas virtuais no QEMU com Qemu Launcher.

Com o VirtualBox é possível ler discos rígidos virtuais gerados a partir de soluções de virtualização como VMware e Microsoft Virtual PC<sup>8</sup>. O seu formato de disco é conhecido como *Virtual Disk Images* (arquivos .vdi). A criação, configuração, execução e administração de máquinas virtuais podem ser feitas através de linha de comando ou de uma maneira mais simples e visual, por intermédio da interface gráfica, conforme a Figura 3.3.

<sup>8</sup><http://www.microsoft.com/windows/virtual-pc/>.



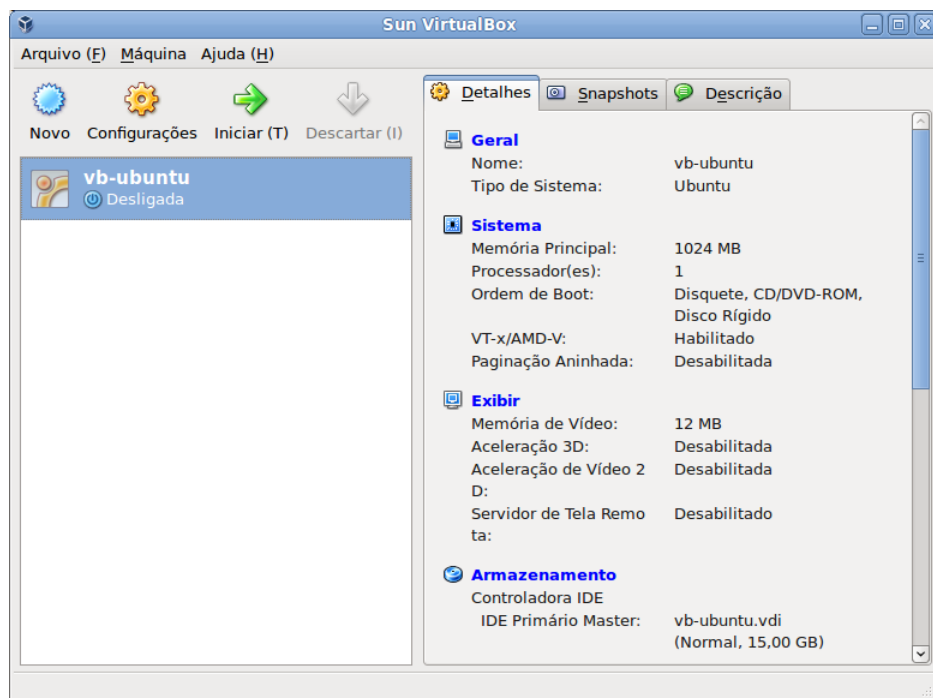


Figura 3.3: Interface de gerenciamento de máquinas virtuais no VirtualBox.

### 3.2.1 Instalação do VirtualBox no Linux

Assim como a maioria das aplicações Linux, o VirtualBox pode ser instalado a partir do arquivo fonte ou por intermédio de arquivos binários disponíveis no endereço eletrônico <http://www.virtualbox.org/wiki/Downloads><sup>9</sup>.

Antes de instalar o VirtualBox, o Qt 4.4.0 (programa que facilita tarefas de escrita ou manutenção de GUI - *Graphical User Interface*) ou superior e o SDL 1.2.7 (uma biblioteca gráfica também chamada de *libsdl*) ou superior, devem ser instalados no sistema hospedeiro como pré-requisitos. É necessário salientar que, para alguns sistemas operacionais hospedeiros, esses pacotes são instalados automaticamente ao instalar o VirtualBox. Como o sistema operacional hospedeiro adotado é o Ubuntu 9.10, após baixar o arquivo de instalação corresponde à versão 3.1.2 do VirtualBox para Linux, é necessário executar o comando de instalação conforme a Figura 3.4.

<sup>9</sup>Acessado em 26 de Janeiro de 2010.

```
sudo dpkg -i virtualbox-3.1_3.1.2-56127_Ubuntu_karmic_i386.deb
```

**Figura 3.4:** Comando para instalação do VirtualBox.

### 3.3 O VMware Workstation

Assim como o VirtualBox, o VMware Workstation fornece nativamente uma interface gráfica de configuração e gerenciamento de máquinas virtuais completa e de fácil utilização. Considerado uma referência no mercado na área de virtualização (LAUREANO, 2006), permite ser utilizado em diversos sistemas operacionais hospedeiros e por sua vez com vários tipos de sistemas operacionais convidados. Diferentemente das outras ferramentas apresentadas neste trabalho, sua instalação possui um tamanho considerável, com aproximadamente 273 MB, se comparado ao VirtualBox com 43.5 MB e o QEMU com 4.5 MB.

A VMware possui diversos produtos no quesito virtualização. Cita-se o ESXi<sup>10</sup> que é um gerenciador de máquinas virtuais do tipo I (rodando diretamente sobre o hardware) destinada ao meio corporativo e o VMware Workstation, um gerenciador de máquinas virtuais do tipo II, destinado a aplicações mais leves e testes. A versão utilizada nos testes será o VMware Workstation 7.0.1 que pode ser obtida através do site do fabricante, juntamente com a necessidade de uma licença, com validade de 30 dias, que pode ser obtida através de cadastro prévio.

O VMware Workstation trabalha isoladamente, criando máquinas virtuais seguras e com o sistema operacional e seus aplicativos de maneira encapsulada. A camada de virtualização mapeia os recursos do *hardware* físico com os recursos da máquina virtual, fazendo com que cada máquina virtual tenha a sua própria CPU, memórias, discos entre outros recursos de uma máquina x86 padrão.

#### 3.3.1 Instalação do VMware Workstation no Linux

Após cadastro prévio e *download* do VMware Workstation no endereço eletrônico <http://www.vmware.com/products/workstation/><sup>11</sup>, observa-se na Figura 3.5 a descrição da nomenclatura para o arquivo de instalação (VMware-Workstation-Full-7.0.1-227600.i386.bundle).

---

<sup>10</sup><http://www.vmware.com/products/esxi/>.

<sup>11</sup>Acessado em 29 de Janeiro de 2010.

7.0.1 - é número da versão  
227600 - corresponde ao número de compilação  
i386 - corresponde a arquitetura, podendo ser também x86\_64

**Figura 3.5:** Nomenclatura do arquivos de instalação do VMware Workstation

O processo de instalação é simples e requer direitos de usuário root. Observe na Figura 3.6 o comando para iniciar o procedimento de instalação.

```
root@core2duo:/# sh VMware-Workstation-Full-7.0.1-227600.i386.bundle
```

**Figura 3.6:** Comando para instalação do VMware Workstation.

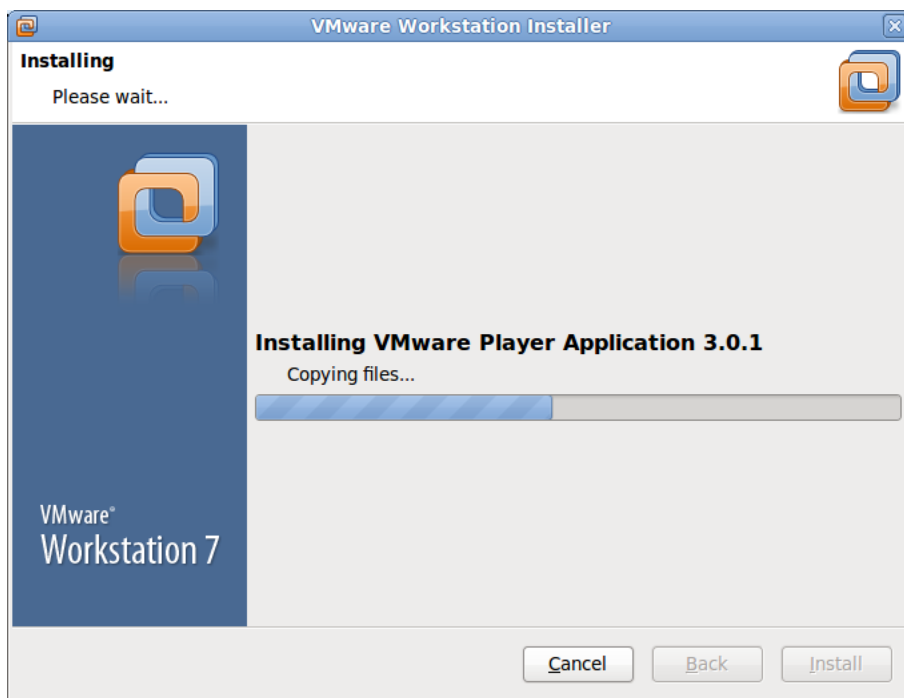
Uma tela de boas vindas será apresentada fornecendo a *End User License Agreement* - (EULA) do *software*. Após leitura, deverá ser aceito o contrato de licença para continuar com o processo de instalação. Este procedimento não requer prática, tão pouco habilidade, sendo a todo momento conduzido via interface gráfica conforme a Figura 3.7. Em seguida, é fornecido a opção de inserir o caminho de instalação do *debugger*. Caso o *Eclipse* esteja instalado, na sequência a opção de se instalar o *Eclipse C/C++ debugging plugin*. Após o fácil procedimento de pré-instalação, basta clicar em *Install* para dar continuidade ao processo.

Após a versão 5.5 do VMware Workstation, o VMware Player foi incluído no pacote de instalação. Caso queira instalá-lo separadamente, poderá ser adquirido através do endereço eletrônico <http://www.vmware.com/products/player/><sup>12</sup>. O VMware Player é uma aplicação livre que permite executar máquinas virtuais criadas através de outros produtos da VMware, bem como criar novas máquinas virtuais. Possui uma interface gráfica mais enxuta, conforme Figura 3.8, se comparada com a padrão do Workstation. Ver Figura 3.9. Por curiosidade, se executado em um hospedeiro com Microsoft Windows, é possível abrir e executar máquinas virtuais do Microsoft Virtual PC e Virtual Server<sup>13</sup>, além do Symantec LiveState Recovery e imagens de sistema.

---

<sup>12</sup>Acessado em 29 de Janeiro de 2010.

<sup>13</sup><http://www.microsoft.com/windowsserversystem/virtualserver/>.



**Figura 3.7:** Tela de instalação do VMware Workstation.

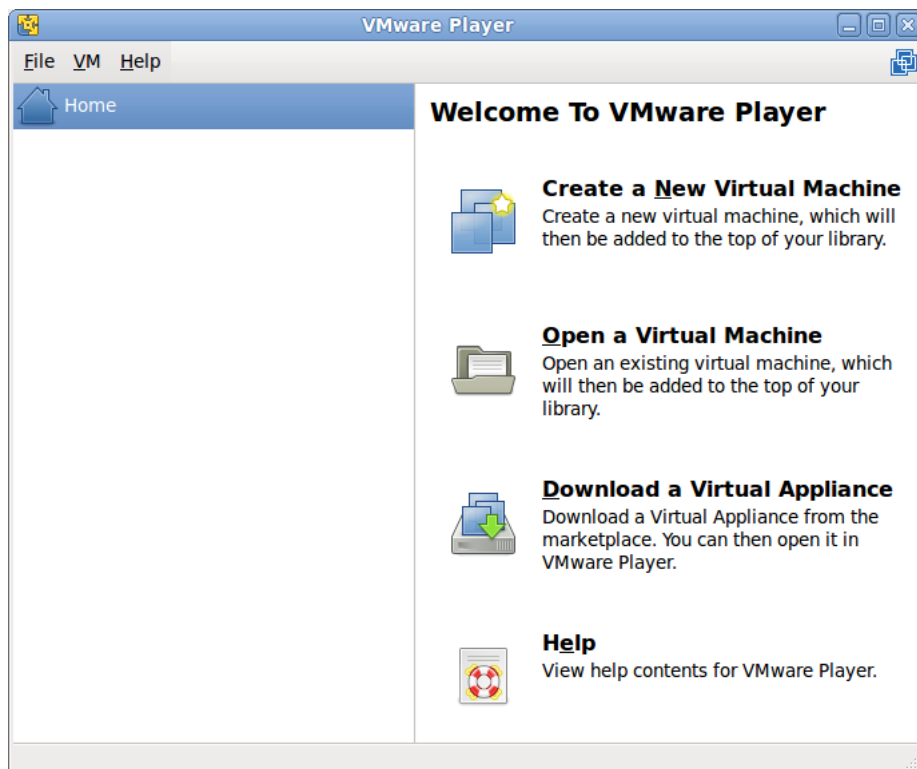


Figura 3.8: VMware Player.

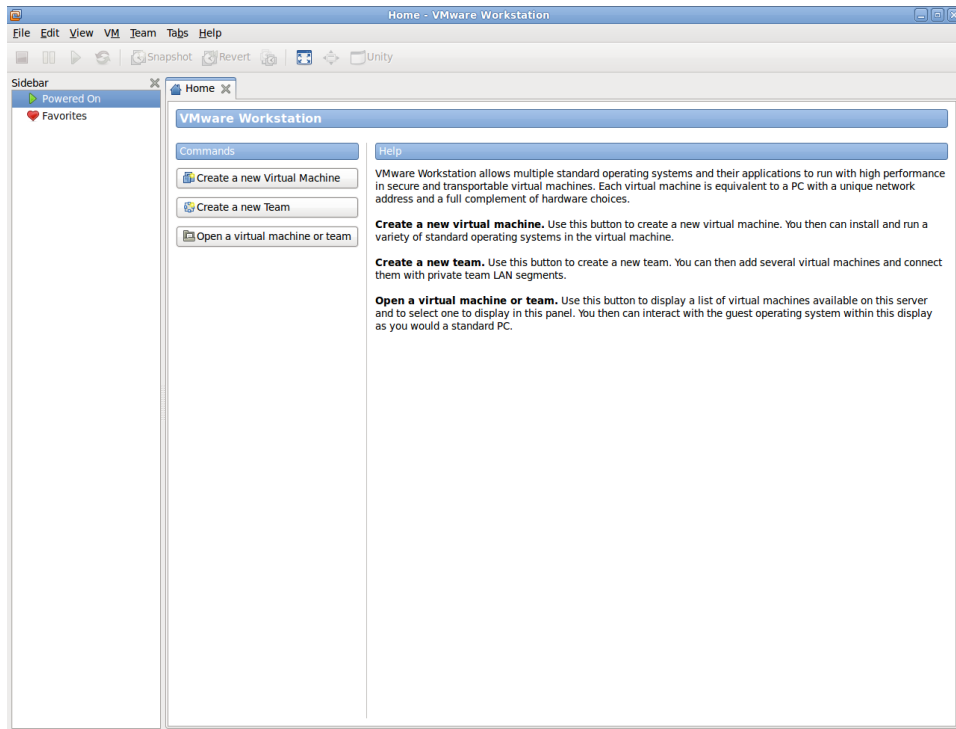


Figura 3.9: VMware Workstation.

## Capítulo 4

# Benchmark

*Benchmarking* é uma maneira de mensurar o desempenho de um sistema aplicando a ele uma carga de trabalho específica. Um *benchmark* fornece uma maneira exata de medir o desempenho para a determinada carga de trabalho (SAAVEDRA; SMITH, 1996).

O hipervisor proporciona extensibilidade e controle de recursos no ambiente virtual, definindo quais recursos estarão disponíveis para as máquinas virtuais. O isolamento fornecido permite que máquinas virtuais com diferentes sistemas operacionais sejam executados lado a lado em um mesmo hospedeiro. É evidente que todas essas vantagens tem um custo. Na medida em que a virtualização adiciona complexibilidade ao sistema, isso faz com que haja uma redução de desempenho para certas atividades.

O desempenho de diferentes hipervisores pode ser testado usando *benchmarking* e mensurado através de diferentes métricas, permitindo a comparação entre as diferentes ferramentas de virtualização e o mundo real dos sistemas (BARNETT, 2007). É importante que os usuários tenham métricas precisas e significativas, a fim de comparar efetivamente o desempenho.

### 4.1 Requisitos de um Benchmark

*Benchmarks* podem ser utilizados para atender a dois objetivos, identificar problemas de desempenho e conseqüentemente ajudar a resolvê-los e ou realizar uma comparação básica entre 2 sistemas (SELTZER *et al.*, 2007).

Para Mogul (MOGUL, 1999), um *benchmark* significativo deve apresentar 3 requisitos:

- Deve ser *repetível*, de modo que outros pesquisadores possam verificar e perpetuar os testes.
- Deve ser *comparável*, para permitir a avaliação de uma determinada tarefa.
- Deve ser *relevante para aplicações importantes*, de modo que os resultados correspondam ao compartimento da vida real.

Estes requisitos implicam que as métricas utilizadas para um *benchmarking* deverão seguir algumas propriedades:

- Um processo de pesquisa científica é imprescindível para obtenção de resultados corretos.
- Os pesquisadores devem chegar a um acordo sobre um conjunto comum de métricas que serão utilizadas.
- A métrica deve ser realista e amplamente utilizada, significa que deve ser aplicada tanto no domínio da investigação quanto no domínio do mundo real.

## 4.2 Carga de Trabalho e Métricas

*Benchmarks* podem ser classificados de acordo com o tipo de carga de trabalho que é imposto sobre o sistema testado, que pode variar de homogêneo para heterogêneo. Podem também ser classificados baseados no tipo de métrica que utilizam para caracterizar o sistema (MOLLER, 2007). Dentre as vários tipos de métricas disponíveis, destacam-se (ENDO *et al.*, 1996):

- *Latência*, que é o intervalo entre o estímulo e a resposta.
- *Throughput*, que é a quantidade de dados ou eventos em um espaço de tempo.
- *Dilatação*, que é a extensão em comprimento.



- *Utilização*, que é o percentual do tempo de computação utilizado em um processador.

As métricas representam aspectos de desempenho que podem ser mais importantes que outros, dependendo dos requisitos do sistema em teste e das características do mesmo. Existem três modelos para quantificar e compreender o desempenho de sistemas operacionais: *macro-benchmark*, *micro-benchmark* e *profiling*.

Os *macro-benchmarks* são utilizados para medir o desempenho geral de aplicações reais em condições generalizadas. De acordo com Aaron Brown and Margo Seltzer, os resultados são realistas e muito úteis para mensurar o desempenho com uma determinada carga de trabalho. Porém, estes resultados estão ligados a muitas variáveis para formar uma base de compreensão a respeito do desempenho do sistema operacional. Podem ser utilizados como carga de trabalho para *macro-benchmark*: compilação de *kernel*, *web server*, operações de banco de dados, entre outros (BROWN; SELTZER, 1997).

Conforme John Ousterhout, no geral, a maioria das ferramentas de *benchmarking* são do tipo *micro-benchmarks*, o que significa que cada um mede o desempenho de um *hardware* particular ou uma característica de um sistema operacional (como velocidade de cópia de memória para memória ou entradas e saídas no *kernel*). São muito úteis para indicar pontos fortes e fracos de sistemas, contudo, não dão uma visão geral de desempenho do sistema (OUSTERHOUT, 1989).

O *profiling* significa acompanhar o desempenho de um programa. Como exemplo, tem-se a frequência e duração de chamadas de função ou blocos básicos. Os dois principais mecanismos para gerar *profiles* são o *tracing* (o programa em análise gera rastro, dependendo da frequência de eventos pode gerar sobrecarga) e o *statistical profiling* (funciona de maneira transparente e a sobrecarga é muito baixa) (MOLLER, 2007).

### 4.3 O Phoronix Test Suite

O Phoronix Test Suite é a mais abrangente plataforma de testes e *benchmarking* disponível para o sistema operacional Linux. Com ele é possível realizar de maneira fácil e eficaz comparações tanto qualitativas como quantitativas. É baseado em vários trabalhos de *benchmarking* para Linux e ferramentas internas desenvolvidas pela Phoronix.com desde o ano de 2004, juntamente com a parceria de

fornecedores de *hardware* para computadores. É um *software open-source*, licenciado sob a GPLv3 do GNU (PHORONIX, 2010).

Esta plataforma de testes pode ser utilizada para fazer desde simples comparações de desempenho entre amigos (uso doméstico), passando pela utilização acadêmica e até mesmo em um ambiente corporativo. Os resultados emitidos pelo Phoronix Test Suite podem ser exibidos de diversas formas como simplesmente via texto, conforme a Figura 4.1 ou através de um relatório em XML, mais completo com informações de *hardware*, *software*, gráficos entre outras informações que serão exibidos no navegador, como mostra a Figura 4.2, além disso os resultados podem ser enviados para a Phoronix Global<sup>1</sup> para serem disponibilizados na Internet.

```
=====
Estimated Run-Time: 3 Minutes
=====

7-Zip Compression:
  compress-7zip
  Estimated Test Run-Time: 3 Minutes
  Expected Trial Run Count: 3
    Started Run 1 @ 03:07:58
    Started Run 2 @ 03:08:43
    Started Run 3 @ 03:09:29

  Test Results:
    2720
    2682
    2700

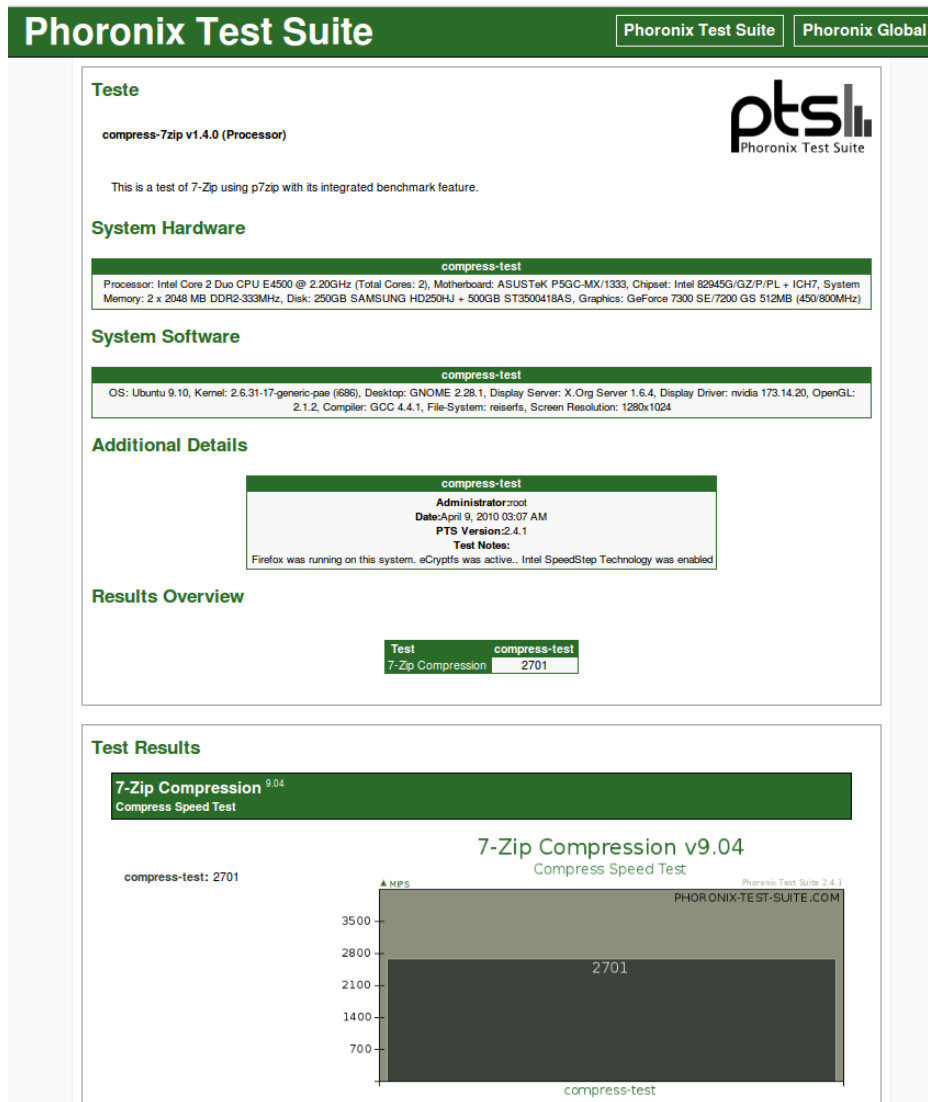
  Average: 2701 MIPS
```

**Figura 4.1:** Resultado de um teste com o phoronix, modo texto.

A instalação do Phoronix Test Suite pode ser obtida no site do fabricante na sessão de *downloads*. A versão utilizada será a 2.4.1 de 18 de Fevereiro de 2010, apelidada de "Lenvik". Sendo a distribuição utilizada nas máquinas hospedeiras será o Ubuntu 8.04 LTS é possível utilizar o pacote binário pré-destinado para Debian/Ubuntu. Após a instalação do Phoronix é necessário a instalação do PHP 5 CLI - interpretador de comandos, obtido via apt-get.

---

<sup>1</sup><http://global.phoronix.com/>.



**Figura 4.2:** Relatório do resultado de um teste com o phoronix, via navegador.

O Phoronix Test Suite possui capacidade para diversos testes, facilmente listados com o comando "phoronix-test-suit list-tests". Dentre os testes disponíveis, foram escolhidos para a análise comparativa realizada neste trabalho:

- *7zip-compress*, teste de velocidade de compressão utilizando a ferramenta 7-Zip. A unidade de medida utilizada é a MIPS - Milhões de instruções por segundo.
- *LZMA-compress*, teste de compressão de arquivo de 256 MB de tamanho, utilizando o algoritmo de compressão de dados de Lempel-Ziv-Markov<sup>2</sup>. Sua unidade de medida os segundos (s). Este algoritmo se caracteriza pela sua alta taxa de compressão (a velocidade de compressão de 1 MB/s em uma CPU de 2 GHz), requer pouca memória e suporte a *multi-threading*.
- *IOzone*, teste de disco para leitura e escrita, com opção para escolha do tamanho do arquivo. Foi utilizado 512 MB. Tem como unidade de medida MB/s (Megabytes por segundo).
- *SQLite*, teste que mede o tempo para executar um número pré-determinado (neste caso 2500) de *inserts* em um banco de dados indexado. Tem como unidade de medida os segundos.
- *BYTE Unix Bench*, este teste executa a versão ANSI C do SciMark2.0, que é um *benchmark* para a computação científica e numérica desenvolvidos por programadores do *National Institute of Standards and Technology*. Neste caso será o utilizado o modelo Dhrystone<sup>3</sup> desenvolvido em 1984 por Reinhold P. Weicker, destinado a ser representante do sistema (inteiro) de programação. Este *benchmark* não contém operações de ponto flutuante. O resultado deste teste é número de dhrystones por segundo (o número de iterações do loop do código principal por segundo.)

A instalação dos testes para o Phoronix Test Suite é feita de maneira muito simples, onde deve-se informar o comando principal "phoronix-test-suite", seguido de "install" caracterizando a instalação. Para terminar o nome do teste desejado. Observa-se na Figura 4.3 todos os comandos necessários para a instalação do 7-Zip Compression, LZMA Compression, IOzone, SQLite e BYTE Unix Benchmark.

Após solicitar a instalação de um teste específico, o mesmo será baixado da Internet para posterior instalação. Conforme a Figura 4.4, observa-se na primeira linha o nome do teste, seguido da quantidade de arquivos necessários (em alguns casos é mais que 1 arquivo), com o tamanho, o nome real do arquivo (contendo sua extensão), o tempo de *download*, o tamanho da instalação e o *status* final do procedimento.

<sup>2</sup>[http://en.wikipedia.org/wiki/Lempel-Ziv-Markov\\_chain\\_algorithm](http://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm).

<sup>3</sup><http://en.wikipedia.org/wiki/Dhrystone>.

```
root@core2duo:~# phoronix-test-suite install compress-7zip
root@core2duo:~# phoronix-test-suite install compress-lzma
root@core2duo:~# phoronix-test-suite install iozone
root@core2duo:~# phoronix-test-suite install sqlite
root@core2duo:~# phoronix-test-suite install byte
```

**Figura 4.3:** Comando para instalação dos testes para o Phoronix.

```
iozone:
  1 File Needed / 1.50 MB
  Downloading: iozone3_323.tar [1.50MB]
  Estimated Download Time: 1m .....
  Installation Size: 2.3 MB
  Installing Test
```

**Figura 4.4:** Processo de instalação dos testes para o Phoronix.



## Capítulo 5

# Análise entre as ferramentas

Neste capítulo, é apresentada a ferramenta Phoronix-Test-Suite, um conjunto de ferramentas para realizar testes de *benchmarking*. Dentre os diversos testes que a compõem, serão utilizados 7-Zip Compression, LZMA Compression, IOzone, SQLite e BYTE Unix Benchmark. Também são demonstrados os resultados da análise comparativa entre o VMware Workstation, VirtualBox e o QEMU.

### 5.1 Execução dos Testes

Em cada ambiente foi criada uma máquina virtual contendo como sistema operacional convidado o Ubuntu 8.04 LTS "Hardy Heron" com 1024MB de RAM e um disco virtual de 16 GB. Além disso, tem como versões: o Kernel 2.6.24-27-generic (i686), GNOME 2.22.3, servidor X.Org 1.4.0.90, GCC 4.2.4 e sistema de arquivos ext3. As máquinas virtuais foram criadas utilizando o arquivo de disco padrão de cada aplicativo, ou seja, VirtualBox com .vdi, o VMware Workstation com .vmdk e o QEMU com o .qcow2.

Uma vez instalados todos os testes, basta executá-los, seguindo a seguinte sintaxe: "phoronix-text-suite run <nome do teste>". A seguir, será possível observar que os gráficos contendo os resultados estão englobando as 3 ferramentas (VMware Workstation, VirtualBox e QEMU). Isso foi possível pois após executar o teste em um ambiente os arquivos de resultados, no diretório "/root/.phoronix-test-suite/test-results", os resultados eram exportados para a máquina virtual no outro ambiente de virtualização. Para manter a organização dos resultados, como

premissa deverá ser mantido o nome padrão para salvar o resultado e no nome único para o teste colocar o nome da ferramenta de virtualização que será testada.

## 5.2 Resultados dos Testes

Nesta sessão são exibidos os resultados da análise comparativa entre as ferramentas de virtualização VMware Workstation, VirtualBox e QEMU.

### 5.2.1 Resultados com o 7zip-Compress

Os testes foram iniciados com o 7zip-Compress *benchmark* na versão 9.04, que mediu a velocidade de compactação em cada ambiente. Observa-se na Figura 5.1 que a máquina virtual no VMware Workstation conseguiu executar 1617 MIPS contra 1511 MIPS no VirtualBox e apenas 415 MIPS no QEMU.

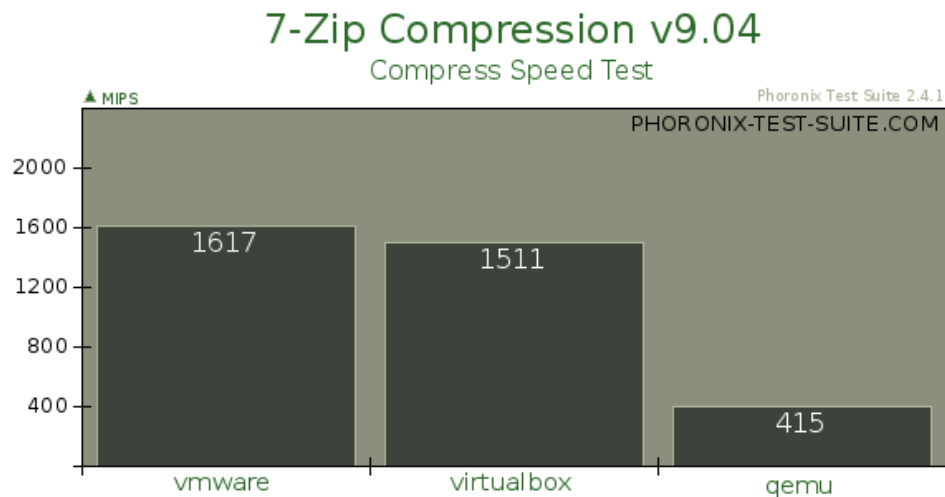


Figura 5.1: 7-zip compress.

### 5.2.2 Resultados com o LZMA-Compress

Os resultados obtidos com o LZMA se mostraram praticamente idênticos aos resultados obtidos entre o VMware Workstation com 339.72 segundos e o Virtual-Box com 340.56 segundos. Ficando o QEMU com uma desvantagem de apro-



ximadamente 995% mais lento que os concorrentes. Observa-se na Figura 5.2 os resultados obtidos.

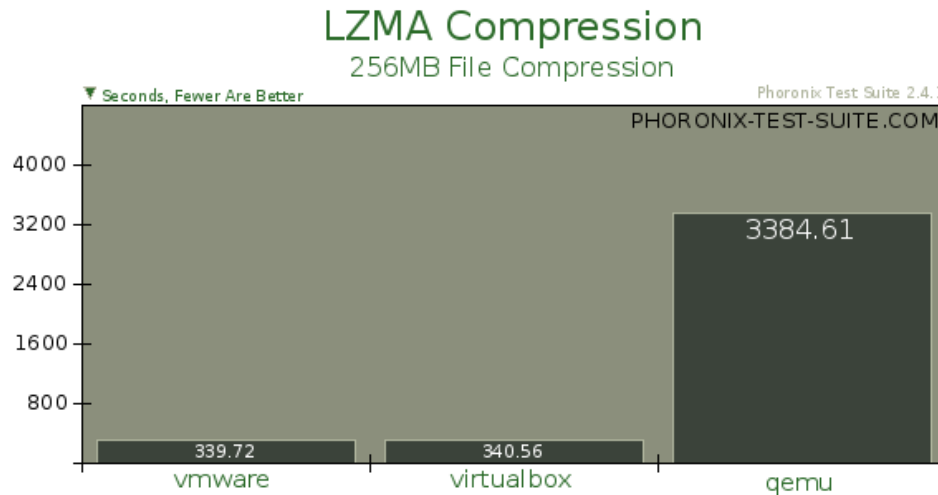


Figura 5.2: LZMA compress.

### 5.2.3 Resultados com o IOzone

O IOzone com 512 MB avaliando o desempenho de escrita mostrou, conforme a Figura 5.3 o VMware Workstation com 67.22 MB/s, o VirtualBox com 62.71 MB/s e o QEMU com 17.95 MB/s. Analisando o IOzone também com 512 MB, só que desta vez tendo como referência o desempenho de leitura, o VirtualBox reverteu o quadro e conseguiu ficar aproximadamente 8% mais veloz que o VMware Workstation e 94% a mais que o QEMU, conforme a Figura 5.4.

### 5.2.4 Resultados com o SQLite

O SQLite *benchmark* mensurou o tempo gasto para processar 2.500 inserções em um banco de dados. Conforme a Figura 5.5, o QEMU ficou aproximadamente 8 vezes mais lento que o VMware Workstation e o VirtualBox 3 vezes mais lento que o mesmo.

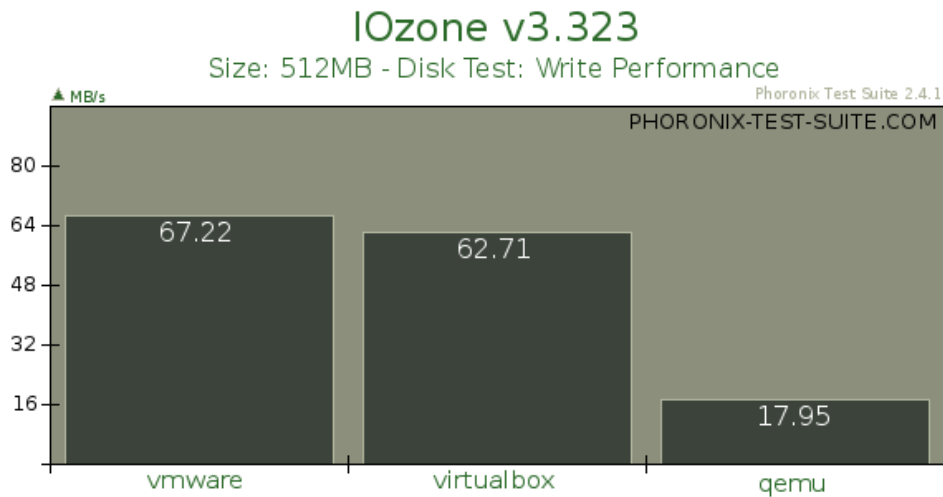


Figura 5.3: IOzone 512MB - Escrita.

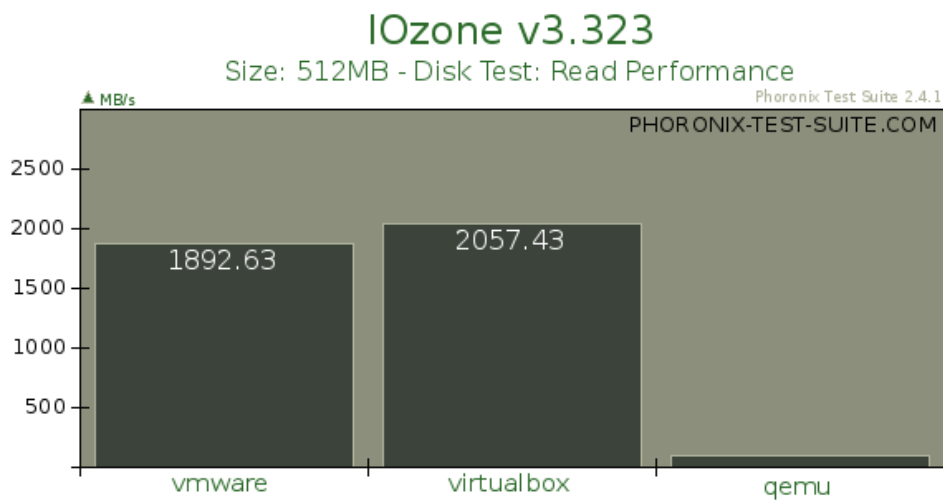


Figura 5.4: IOzone 512MB - Leitura.

### 5.2.5 Resultados com o BYTE Unix Bench

No último teste executado, conforme a Figura 5.6 o VirtualBox ficou na frente com 7336081.50 LPS contra 7210857.10 LPS do VMware Workstation. Não foi possível coletar dados através do QEMU, pois por diversas vezes o máquina virtual travou não conseguindo processar e obter os resultados.

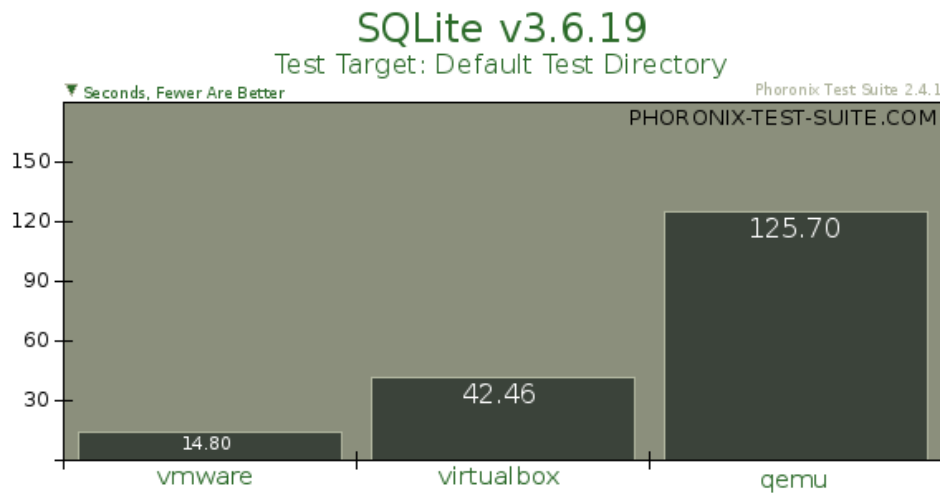


Figura 5.5: SQLite.

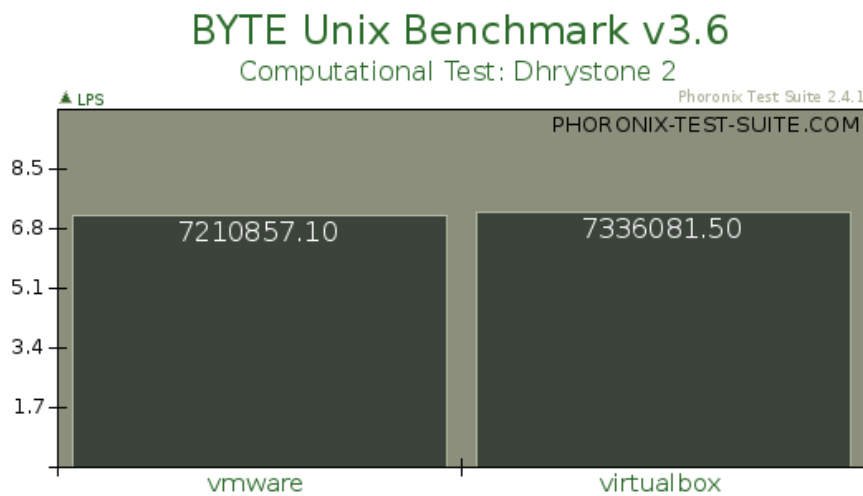


Figura 5.6: BYTE Unix Bench.

### 5.3 Discussão dos Testes

Após execução dos testes observou-se na maioria dos resultados uma equiparação de desempenho entre o VirtualBox e o VMware Workstation, o que comprova a maturidade e capacidade do VirtualBox em relação ao líder de mercado a VMware. O VirtualBox chegou a apresentar desempenho melhor que o VMware Workstation

em alguns quesitos, mas teve uma redução significativa na atividade de inserção em um banco de dados, chegando a ser aproximadamente 2.8 vezes mais lento.

O QEMU por sua vez apresentou baixo desempenho em relação ao Virtual-Box e VMware Workstation. Além do mais, a máquina virtual executada em seu ambiente não foi capaz de concluir com sucesso o último teste, o BYTE Unix. O autor tem conhecimento de que talvez a heterogeneidade no processo de instalação do QEMU, aplicativos auxiliares e o módulo acelerador possam ter contribuído para o baixo desempenho. Contudo, ressalta que inicialmente a mistura entre arquivos binários e compiláveis pareceram ser o caminho mais promissor para o desenvolvimento deste trabalho. Portanto, o impacto no desempenho do QEMU pode estar diretamente ligado à maneira como foi introduzido no sistema hospedeiro e até mesmo como foi utilizado, neste caso com o módulo KQEMU e não com o KVM por limitação do *hardware* utilizado.

## Capítulo 6

# Conclusão

Como observado, a utilização de virtualização no nível do *hardware* é uma maneira bastante específica de virtualização, que em conjunto com alto desempenho do atuais computadores, vem se tornando cada vez mais popular. Apesar de toda popularidade de hoje, constatou-se que o termo virtualização já existia no passado e já utilizava parte do conceito nas estações IBM na década de 60.

Foram examinados o VirtualBox, VMware Workstation e QEMU entre os diversos hipervisores disponíveis no mercado, cada um deles com suas particularidades, seja pela licença, facilidade de instalação ou pela capacidade de desempenho. Observa-se que a opção de utilização em alguns casos poderá não levar em conta desempenho e fatores técnicos e sim a virtuosidade que o aplicativo apresenta ou proporciona ao usuário. Mesmo com os resultados da análise de desempenho entre VirtualBox e o VMware Workstation serem equiparados, destaca-se a preferência pela utilização do VirtualBox pela estabilidade apresentada, interface gráfica agradável e de fácil utilização, além facilidade de aquisição e instalação. O QEMU pode ter tido uma desvantagem na comparação de desempenho pela maneira como foi instalado, ou seja, a utilização do módulo KQEMU que não está mais recebendo o foco no desenvolvimento e que por sua vez foi transferido para o KVM.

O desempenho é uma importante consideração para o desenvolvimento e utilização de máquinas virtuais. A análise de desempenho, o *benchmarking*, é uma opção utilizada para determinar o desempenho entre os diversos hipervisores ou entre uma máquina física e uma virtual. Existem diversos trabalhos e pesquisas que realizam este tipo de comparação, dentre eles destacam-se Barnett (BARNETT, 2007), Deshane (DESHANE *et al.*, 2008), Walters (WALTERS *et al.*, 2008), en-

tre outros que servem como apoio e opções diferenciadas de comparação. Como sugestão de trabalhos futuros a possibilidade de comparação das mesmas ferramentas com a utilização do KVM, utilização de outras sistemas operacionais tanto hospedeiro como convidado e até mesmo ferramentas de *benchmarking* correlatas.

# Referências Bibliográficas

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEBAUER†, R.; PRATT, R.; WARFIELD, A. Xen and the art of virtualization. *University of Cambridge Computer Laboratory*, 2003.

BARNETT, R. J. Virtualized systems and their performance a literature review. *Rhodes University*, 2007. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.4788&rep=rep1&type=pdf>>.

BELLARD, F. Qemu, a fast and portable dynamic translator. *USENIX Annual Technical Conference*, 2005. Disponível em: <[http://www.usenix.org/publications/library/proceedings/usenix05/tech/freenix/full\\_papers/bellard-bellard.pdf](http://www.usenix.org/publications/library/proceedings/usenix05/tech/freenix/full_papers/bellard-bellard.pdf)>.

BROWN, A. B.; SELTZER, M. Operating system benchmarking in the wake of lmbench: A case study of the performance of netbsd on the intel x86 architecture. *Harvard University*, 1997. Disponível em: <<http://www.eecs.harvard.edu/vino/perf/hbench/sigmetrics/hbench.html>>.

DESHANE, T.; SHEPHERD, Z.; MATTHEWS, J. N.; BEN-YEHUDA, M.; SHAH, A.; RAO, B. Quantitative comparison of xen and kvm. *Xen Summit*, 2008.

ENDO, Y.; WANG, Z.; CHEN, J. B.; SELTZER, M. Using latency to evaluate interactive system performance. *SIGOPS Operating Systems Review*, 1996.

LAUREANO, M. *Máquinas virtuais e emuladores : conceitos, técnicas e aplicações*. São Paulo: Novatec, 2006.

MATTOS, D. M. F. Virtualização: Vmware e xen. *UFRJ*, 2008. Disponível em: <[http://www.gta.ufrj.br/grad/08\\_1/virtual/artigo.pdf](http://www.gta.ufrj.br/grad/08_1/virtual/artigo.pdf)>.

MENASCE, D. A. Virtualization: Concepts, applications, and performance modeling. *George Mason University*, 2005.

MOGUL, J. C. Brittle metrics in operating systems research. *Compaq Computer Corporation*, 1999. Disponível em: <<http://www.computer.org/portal/web/csdl%2Fdoi/10.1109/HOTOS.1999.798383>>.

MOLLER, K.-T. Virtual machine benchmarking. *Universitat Karlsruhe*, 2007.

OLIVEIRA, G. V. N. Solução de virtualização completa utilizando vmware e software livre: Um estudo de caso na cef. *UFLA*, abr. 2007. Disponível em: <<http://www.ginux.ufla.br/files/mono-GuilhermeOliveira.pdf>>.

OUSTERHOUT, J. Why aren't operating systems getting faster as fast as hardware? *Western Research Laboratory*, 1989. Disponível em: <<http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-11.pdf>>.

PHORONIX. Sitio acessado em janeiro. *Phoronix Test Suite*, 2010. Disponível em: <<http://www.phoronix-test-suite.com/>>.

PUEL. *VirtualBox Personal Use and Evaluation License (PUEL)*. [S.l.], set. 2008. Disponível em: <<http://www.virtualbox.org/wiki/VirtualBox%20PUEL>>.

ROSE, R. Survey of system virtualization techniques. *UFRJ*, 2004. Disponível em: <<http://www.robertwrose.com/vita/rose-virtualization.pdf>>.

SAAVEDRA, R. H.; SMITH, A. Analysis of benchmark characteristics and benchmark performance prediction. *ACM Trans. Comput.*, 1996. Disponível em: <<http://portal.acm.org/citation.cfm?id=235545>>.

SELTZER, M.; KRINSKY, D.; SMITH, K.; ZHANG, X. The case for application-specific benchmarking. *Harvard University*, 2007. Disponível em: <<http://www.eecs.harvard.edu/~keith/papers/hotos99.pdf>>.

SUN MICROSYSTEMS. *Sun VirtualBox User Manual*. [S.l.], 2010. Disponível em: <<http://download.virtualbox.org/virtualbox/3.1.6/UserManual.pdf>>.

VMWARE. *Understanding Full Virtualization, Paravirtualization, and Hardware Assist*. California, 2007. Disponível em: <<http://www.vmware.com>>.

WALTERS, J. P.; CHAUDHARY, V.; CHA, M.; SALVATORE, G. J.; GALLO, S. A comparison of virtualization technologies for hpc. *22nd International Conference on Advanced Information Networking and Applications*, 2008.

WIKIQEMU. *QEMU Emulator User Documentation*. [S.l.], jan. 2010. Disponível em: <<http://wiki.qemu.org/download/qemu-doc.html>>.