

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**AUTOMATIZAÇÃO NA
CONFIGURAÇÃO DE REDE EM
VIRTUAL APPLIANCES PARA
VIRTUALBOX**

TRABALHO DE GRADUAÇÃO

Alexandro Klein dos Santos

Santa Maria, RS, Brasil

2008

AUTOMATIZAÇÃO NA CONFIGURAÇÃO DE REDE EM VIRTUAL APPLIANCES PARA VIRTUALBOX

por

Alexandro Klein dos Santos

Trabalho de Graduação apresentado ao Curso de Ciência da Computação
da Universidade Federal de Santa Maria (UFSM, RS), como requisito
parcial para a obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof^a Dr^a Andrea Schwertner Charão

Trabalho de Graduação N. 266

Santa Maria, RS, Brasil

2008

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**AUTOMATIZAÇÃO NA CONFIGURAÇÃO DE REDE EM
VIRTUAL APPLIANCES PARA VIRTUALBOX**

elaborado por
Alexandro Klein dos Santos

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:

Profª Drª Andrea Schwertner Charão
(Presidente/Orientador)

Profª Drª Iara Augustin (UFSM)

Prof. Dr. Raul Ceretta Nunes (UFSM)

Santa Maria, 17 de Dezembro de 2008.

“Trabalhamos não só para produzir, senão para dar valor ao tempo.”

— EUGENE DELACROIX

“O que mais precisamos é de alguém que nos obrigue a fazer aquilo que sabemos.”

— RALPH WALDO EMERSON

“Ninguém comete erro maior do que não fazer nada porque só pode fazer um pouco.”

— EDMUND BURKE

“A vitória mais bela que se pode alcançar é vencer a si mesmo.”

— IGNÁCIO DE LOYOLA

“As pessoas inteligentes falam por experiência. As pessoas mais inteligentes, por experiência, não falam”

— AUTOR DESCONHECIDO

“Não sabendo que era impossível, ele foi lá e fez.”

— AUTOR DESCONHECIDO

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, por estar ao meu lado sempre que precisei, não deixando-me na mão até mesmo quando acreditei estar sozinho.

Aos meus pais, que com muita paciência sempre me apoiaram, vivenciando cada preocupação, cada momento de tristeza e alegria, sempre acreditando no meu potencial.

A minha irmã Taiane, que estando na mesma condição de formanda, soube entender meus questionamentos, minhas dúvidas, e também por todos os momentos de descontração e alegria que ela me proporcionou.

Aos meus colegas de aula, de PET, de curso, em especial o João, a Suzana, o Henrique, o Vitor, nunca esquecerei o apoio, a amizade eterna que obtive com alguns, a força nos trabalhos, nas provas.

À professora Andrea, sempre disposta a ouvir, passando dicas valiosas para o bom desenrolar do trabalho, e sem a qual nada disso seria possível.

E aos demais amigos, parentes e conhecidos que de uma forma ou de outra ajudaram no decorrer dessa jornada, muito obrigado!

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

AUTOMATIZAÇÃO NA CONFIGURAÇÃO DE REDE EM VIRTUAL APPLIANCES PARA VIRTUALBOX

Autor: Alexandro Klein dos Santos

Orientador: Prof^ª Dr^ª Andrea Schwertner Charão

Local e data da defesa: Santa Maria, 17 de Dezembro de 2008.

Os avanços recentes na área de virtualização têm motivado a criação de *virtual appliances*, que são conjuntos de *software* formados por uma máquina virtual com um sistema operacional mínimo, associado a aplicativos específicos. O uso de *virtual appliances* tem várias vantagens, eliminando custos e complexidade de instalação, teste, configuração e manutenção de aplicativos. Neste trabalho, tem-se como objetivo conhecer e analisar técnicas utilizadas na configuração de rede dos *virtual appliances*, tendo por base inicial a ferramenta VirtualBox, visando automatizar algumas etapas dessa configuração. No decorrer do trabalho, tem-se a apresentação da ferramenta implementada VirtAK, desenvolvida para facilitar a manipulação da rede de diversas *virtual appliances*.

Palavras-chave: Virtualbox; virtual appliance; virtualização; redes de computadores.

ABSTRACT

Trabalho de Graduação
Undergraduate Program in Computer Science
Universidade Federal de Santa Maria

AUTOMATION IN THE NETWORK CONFIGURATION IN VIRTUAL APPLIANCES FOR VIRTUALBOX

Author: Alexandro Klein dos Santos
Advisor: Prof^a Dr^a Andrea Schwertner Charão

The recent advances in the area of virtualization have motivated the creation of *virtual appliances*, which are sets of *software* formed by a virtual machine with a minimum operating system, associated with specific applications. The use of *virtual appliances* has several advantages, eliminating cost and complexity of installation, testing, configuration and maintenance of applications. This work focuses on the process of configuring multiple virtual appliances targeted to VirtualBox virtualization technologies. Such process usually requires repetitive tasks, so our approach was to analyze configuration steps and look for tasks that could be carried out automatically. During this work, we developed a configuration tool named VirtAK, aimed to facilitate the network configuration process for multiple *virtual appliances*.

Keywords: virtualbox, virtual appliance, virtualization, networks of computers.

LISTA DE FIGURAS

Figura 2.1 – VirtualBox GUI	14
Figura 2.2 – Gerenciador de Discos Virtuais: no detalhe, imagens VDI e VMDK ..	15
Figura 2.3 – Repositório de <i>virtual appliances</i> para VirtualBox.Fonte: (HELPDES- KLIVE, 2008)	16
Figura 3.1 – Estrutura da rede com NAT no VirtualBox. Fonte: (PINHEIRO, 2008)	19
Figura 3.2 – Estrutura da rede com Interface do Hospedeiro no VirtualBox. Fonte: (PI- NHEIRO, 2008).....	20
Figura 3.3 – Estrutura usando Rede Interna no VirtualBox. Fonte: (PINHEIRO, 2008).....	22
Figura 4.1 – Função utilizada para configurar a rede com NAT.	25
Figura 4.2 – Interface Gráfica do VirtAK.....	26
Figura 4.3 – VirtAK – Escolha dos VAs.	26
Figura 4.4 – VirtAK – Configuração simples.	27
Figura 4.5 – VirtAK – Configuração composta.	27
Figura 4.6 – VA LAMP em detalhes: phpinfo e phpmyadmin	28
Figura 4.7 – Duas VAs LAMP com NAT e Rede Interna	29
Figura 4.8 – Configurações de Rede	29

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
COM	Component Object Model
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
GUI	Graphical User Interface
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
JeOS	Just Enough Operating System
LAMP	Linux Apache MySQL PHP
MMV	Monitor de Máquinas Virtuais
MV	Máquina Virtual
NAT	Network Address Translation
OSE	Open Source Edition
SOAP	Simple Object Access Protocol
SSH	Secure Shell
USB	Universal Serial Bus
VA	Virtual Appliance
VAN	Virtual Appliance Network
VDI	Virtual Disk Image
VMDK	Virtual Machine Disk
VRDP	VirtualBox Remote Desktop Protocol
WSDL	Web Service Definition Language
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO	13
2.1	Máquinas Virtuais	13
2.2	VirtualBox	13
2.3	Virtual Appliances	15
3	CRIAÇÃO DE <i>VIRTUAL APPLIANCES</i>	18
3.1	Procedimentos Gerais	18
3.2	Configuração da Rede dos <i>Virtual Appliances</i> para o VirtualBox	19
3.2.1	Rede com NAT	19
3.2.2	Rede com Interface no Hospedeiro	20
3.2.3	Rede Interna	22
4	VIRTAK: FERRAMENTA DE CONFIGURAÇÃO AUTOMÁTICA DE REDE	23
4.1	Configuração Manual	23
4.2	Configuração Automática	23
4.2.1	Os <i>Scripts</i>	24
4.2.2	A interface gráfica	24
4.3	Comparação entre VirtualBox GUI e VirtAK	28
5	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	32

1 INTRODUÇÃO

O conceito de virtualização remete a meados das décadas de 50 e 60, juntamente com o desenvolvimento dos *mainframes* da IBM e suas máquinas virtuais. Pode-se conceituar virtualização como uma camada de abstração que possibilita a execução de diferentes plataformas operacionais sobre um mesmo *hardware*.

O declínio dos *mainframes* e a disseminação dos computadores pessoais, inicialmente com baixa capacidade computacional, fizeram com que a virtualização deixasse de ser utilizada. Nos últimos anos, no entanto, o aumento do desempenho dos computadores pessoais fez as máquinas virtuais ressurgirem como uma solução vantajosa, devido às suas qualidades como portabilidade e segurança.

Com o crescimento do mercado tecnológico, os computadores vêm aumentando seu nível de complexidade. A virtualização vem para ajudar a ocultar alguns recursos, tornando mais simples a configuração e manutenção destes. Além disso, empresas e centros acadêmicos que antigamente despendiam muitos recursos financeiros, hoje com a virtualização podem investir em poucas máquinas reais, poupando também em energia elétrica, espaço físico e recursos humanos (LAUREANO, 2006). Muitas empresas investiram, então, nessa tecnologia e hoje em dia pode-se contar com muitas soluções para ambientes virtualizados, tais como Xen (BARHAM et al., 2003), OpenVZ (SWSOFT, 2008), VMware (VMWARE, INC.a, 2008), VirtualBox (INNOTEK, 2008), entre outros.

Na área das tecnologias de virtualização, surgiram novos conceitos e abordagens, tais como o conceito de *virtual appliances* (SAPUNTZAKIS et al., 2003). Um *virtual appliance* consiste em uma imagem de máquina virtual configurada com um sistema operacional mínimo e um ou mais aplicativos ou serviços de propósito específico. Os *virtual appliances* são considerados como a evolução do paradigma da virtualização, visto que unem os benefícios desta com as diversas soluções em aplicativos existentes, procurando

suprir as mais diversas demandas do mercado de forma rápida e simples (SAPUNTZAKIS et al., 2003). Um exemplo típico de *virtual appliance* destina-se a uma gama de sistemas conhecidos como LAMP(Linux+Apache+MySQL+PHP), compostos por uma distribuição de sistema operacional Linux, um servidor Web Apache, um servidor de bancos de dados MySQL e um interpretador PHP. A instalação e configuração de cada um desses componentes é uma tarefa trabalhosa, que pode ser abreviada com o uso de *virtual appliances*.

Em ambientes que produzem e utilizam múltiplos *virtual appliances*, surge um problema tecnológico referente à automatização deste processo. Os procedimentos técnicos envolvidos na geração de um *virtual appliance* são minuciosos e, geralmente, executados manualmente. Esses procedimentos, geralmente, são dependentes da tecnologia de virtualização empregada e se tornam repetitivos quando se lida com vários *appliances*. Entre as tarefas mais complexas, das quais administradores de sistemas são incumbidos de realizar, tem-se a criação e gerência da rede desses diversos *virtual appliances*.

Diante deste cenário, o presente trabalho visa investigar os passos passíveis de automatização na configuração de rede dos *virtual appliances*, através do aprimoramento das técnicas utilizadas atualmente, facilitando a manipulação de múltiplos *appliances*. Dada a diversidade de soluções de virtualização, decidiu-se limitar este trabalho aos *virtual appliances* para o VirtualBox (INNOTEK, 2008), que destaca-se por sua estabilidade, facilidade de utilização e por possuir uma versão de código aberto.

O restante deste texto está organizado da seguinte maneira: o capítulo 2 traz os conceitos utilizados como referencial teórico para o trabalho, o capítulo 3 descreve etapas para criação de *virtual appliances*, o capítulo 4 trata sobre o trabalho desenvolvido bem como do sistema implementado VirtAK e, por fim, o capítulo 5 traz as considerações finais.

2 FUNDAMENTAÇÃO

Neste capítulo estão relacionados alguns conceitos que formaram a base para o trabalho desenvolvido.

2.1 Máquinas Virtuais

Máquinas Virtuais (MV) podem ser vistas como simulações de réplicas físicas de máquinas reais. Recursos como memória, disco rígido e placa de rede são virtualizados e distribuídos entre máquinas virtuais, e estas são controladas por um Monitor de Máquinas Virtuais (MMV). A idéia é dar ao usuário das MVs a ilusão de que esses recursos são utilizados de forma exclusiva por ele. O principal problema que as MVs procuram resolver é como criar uma camada de compatibilidade entre diversas plataformas de *hardware* e *software* existentes, de forma a se beneficiar desse tipo de ambiente. A essa camada é dado o nome de virtualização (LAUREANO, 2006). Entre os principais benefícios das MVs estão a segurança e a portabilidade (WANG et al., 2007).

2.2 VirtualBox

Entre as tecnologias de virtualização disponíveis, atualmente, tem-se o VirtualBox (INNOTEK, 2008). Trata-se de uma plataforma de virtualização estável que possui uma versão de código aberto (*VirtualBox Open Source Edition*) e é de fácil utilização. Sua documentação é de fácil compreensão, bem como suas ferramentas, que são constantemente modificadas e testadas, servindo tanto para o controle como para a gerência das máquinas virtuais que ele mantém. Vale destacar algumas dessas ferramentas:

- **VirtualBox GUI** – interface gráfica administrativa (veja figura 2.1);
- **VBoxManage** – controlado pela linha de comando, possui todas as funcionalidades da VirtualBox GUI e mais algumas funções extras para controle e criação das

máquinas virtuais;

- **VBoxSDL** – interface gráfica simplificada que mostra somente o necessário para rodar a máquina virtual;
- **VBoxHeadless** – utiliza o *VirtualBox Remote Desktop Protocol* (VRDP) para ativar máquinas virtuais sem interface gráfica, com o objetivo principal de acessá-las remotamente.

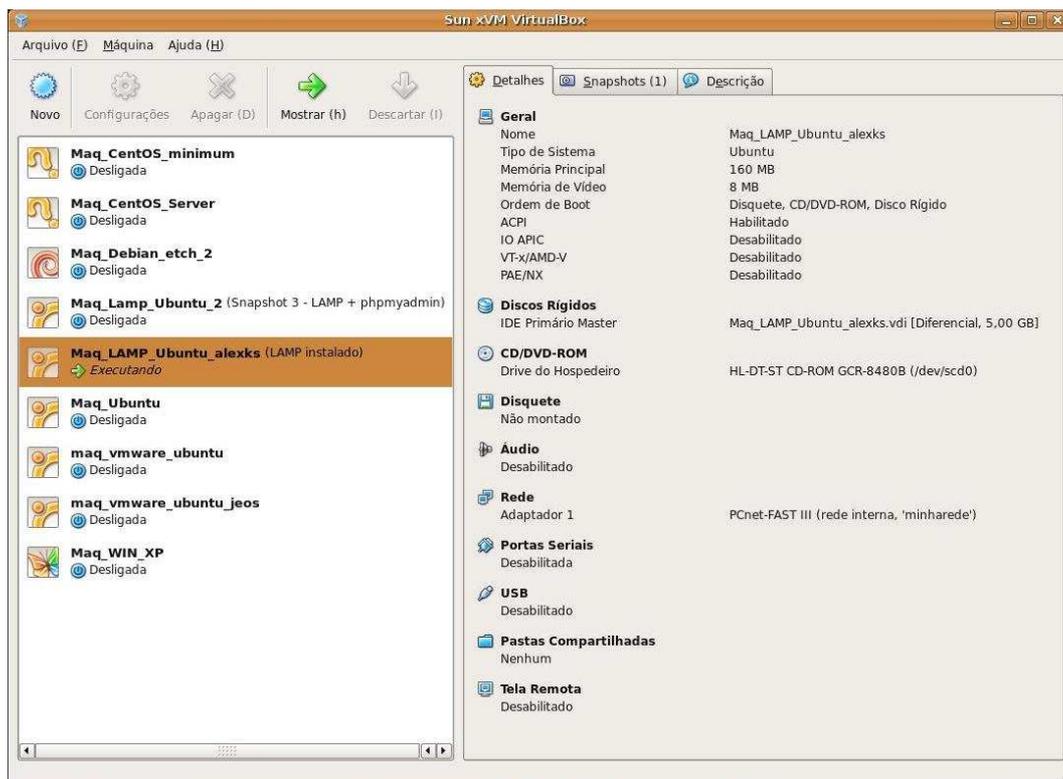


Figura 2.1: VirtualBox GUI

O VirtualBox destaca-se, ainda, pelo suporte aos tipos de arquivos de armazenamento mais usados atualmente, que são o VDI e o VMDK. Este último é utilizado pelo popular VMware (SUN MICROSYSTEMS, 2008). A figura 2.2 mostra o gerenciador de discos virtuais.

Ele é um produto criado pela empresa alemã Innotek, recentemente comprado pela Sun Microsystems, que o uniu à sua plataforma xVM (LINUXa, 2008). Desde então, o VirtualBox vem adquirindo constantes atualizações, que são disponibilizadas no *site* oficial (INNOTEK, 2008). Além disso, o VirtualBox foi criado para a família x86, seu sistema operacional hospedeiro¹ suporta Windows (XP, Server 2003, Vista), Linux (Debian,

¹O mesmo que host. SO o qual está instalado o VirtualBox. Hospeda as máquinas virtuais.

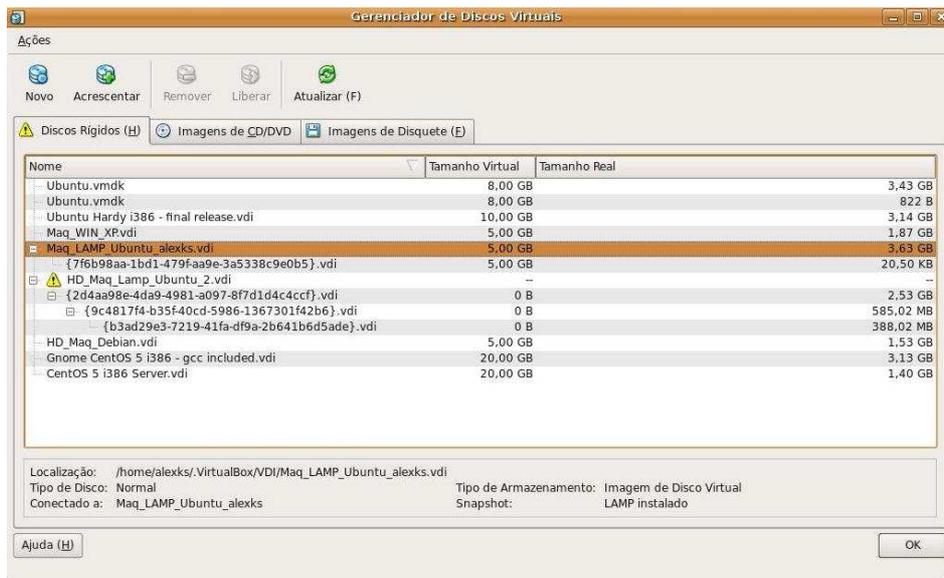


Figura 2.2: Gerenciador de Discos Virtuais: no detalhe, imagens VDI e VMDK

Fedora, Gentoo, RedHat, Suse, Ubuntu, Mandriva) e Mac OS X da Apple, utilizando-se de um diferenciado sistema de virtualização para rodar suas máquinas virtuais diretamente sobre o sistema hospedeiro, sem a necessidade de modificá-lo (porém, sabe-se que internamente o VirtualBox intercepta algumas requisições do sistema hóspede², as quais poderiam afetar o hospedeiro) (SUN MICROSYSTEMS, 2008).

Essas e outras tantas características fizeram do VirtualBox um *software* que atende bem aos objetivos desse trabalho.

2.3 Virtual Appliances

Um *virtual appliance* (VA) – também chamado de aparelho virtual, aplicativo virtual ou *software appliance* – é um conjunto formado por uma máquina virtual com um sistema operacional mínimo (*Just Enough Operating System* – JeOS) (KRISHNAMURTI, 2007) e um aplicativo, servindo tanto para um propósito específico (quando se tem um aplicativo instalado) como para facilitar a aplicação de testes em aplicativos e/ou ambientes (BR-LINUX.ORG, 2008; VIRTUALAPPLIANCES.NET, 2008). Esse conjunto vem pré-configurado e pré-instalado, com o objetivo de eliminar custos e a complexidade com instalação, configuração e manutenção, associados a grande quantidade de aplicativos de *software* existentes (WIKIPEDIAa, 2008). Como qualquer aplicativo, um VA pode ser transportado e armazenado digitalmente (A figura 2.3 mostra um repositório de *virtual*

²O mesmo que guest. SO instalado nos *virtual appliances*.

appliances encontrado na Internet). Trabalhos relacionados sugerem a divisão dessa estrutura em dois tipos: programas instalados e dados gerados/manipulados pelos VAs. Essa divisão é feita para se ter um controle maior sobre atualizações de *software* e possíveis manutenções (SAPUNTZAKIS et al., 2003).

Download / VirtualBox VDI free images				
File	Comments	Views	Added	Size
KUbuntu 7.04 Feisty - final version	0	832	2007-07-10 23:39:32	659.99 MB
Ubuntu 7.04 Feisty - final version	0	809	2007-07-10 23:40:40	963.84 MB
XUbuntu 7.04 Feisty - final version	0	233	2007-07-10 23:42:56	825.14 MB
CentOS 5 i386 server - GUI	0	582	2007-07-12 02:56:51	923.85 MB
Ubuntu Studio 7.04 Feisty - final version with Cinelerra	0	267	2007-07-12 23:42:43	946.52 MB
CentOS 5 i386 Server	0	426	2007-07-13 02:59:55	390 MB
Fedora Core 7 i386	0	449	2007-07-13 23:36:23	1750.15 MB
Gnome CentOS 5 i386 with gcc included	0	221	2007-07-13 23:37:40	936.11 MB
Ubuntu Hardy i386 - final release	0	1191	2008-05-06 15:02:39	772.5 MB
KUbuntu Hardy i386 - final release	0	610	2008-05-06 15:02:39	768.4 MB

Figura 2.3: Repositório de *virtual appliances* para VirtualBox. Fonte: (HELPDESKLIVE, 2008)

Um VA pode ser visto também como uma rede de *virtual appliances* (VAN – *Virtual Appliance Network*) composta, por exemplo, por serviços como DHCP, DNS, Web, Mail, onde cada serviço desses é oferecido também por um VA (SAPUNTZAKIS et al., 2003). Esse tipo de solução, muitas vezes, é agregada à *cloud computing*³ (e.g. Amazon EC2) e usada por empresas para se ter maior segurança, portabilidade, isolamento e demais características válidas em serviços distribuídos (LAURIANNE MCLAUGHLIN, 2008; AMAZON WEB SERVICES, 2008). Porém, para se ter todas essas facilidades, há um inconveniente: o tamanho do VA, que é relativamente maior que o aplicativo que ele mantém (LINUX.COM, 2008).

Existem várias soluções disponíveis para ambientes de *virtual appliances*, embora muitas delas ou possuem licença de alto custo, ou são de código fechado. Entre as soluções existentes pode-se citar a JumpBox (JUMPBOX, INC., 2008), a rPath (RPATHa, 2008) e a VMware (VMWARE, INC.b, 2008).

³Um tipo de sistema paralelo e distribuído que consiste em uma coleção de computadores interconectados e virtualizados, apresentados como um ou mais recursos unificados de computação, baseados em acordos de níveis de serviços estabelecidos através de negociação entre o prestador de serviços e os consumidores. (BUYA; YEO; VENUGOPAL, 2008).

Entre os *White Papers* da VMware e rPath (VMWARE, INC.c, 2008; RPATHb, 2008) encontra-se propostas de algumas características desejáveis ao se criar *virtual appliances*:

- **Encapsulamento completo** – todos os componentes necessários para executar o *virtual appliance* sobre uma camada de virtualização devem estar presentes;
- **Sistema operacional otimizado (JeOS)** – utilizando pouco espaço em disco, e somente os recursos necessários para rodar uma aplicação específica;
- **Configuração inicial** – uma senha para uma conta administrativa, um endereço IP para acesso a Web, etc;
- **Componentes padrões** – interface para linha de comando, suporte a SSH, etc;
- **Configuração da máquina virtual** – dispositivos externos que poderão ser acessados (CD Player, USB, etc.), kit de ferramentas incorporado, etc.

Dependendo da criatividade e da necessidade do desenvolvedor, ainda se tem outras características que são comumente agregadas aos *virtual appliances*, por exemplo, configuração de um DHCP visando a gerência das aplicações via Web (VMWARE, INC.c, 2008).

3 CRIAÇÃO DE *VIRTUAL APPLIANCES*

O objetivo deste capítulo é mostrar os procedimentos gerais para criação e gerenciamento de *virtual appliances*, bem como os tipos de configuração de rede possíveis de serem aplicados no VirtualBox.

3.1 Procedimentos Gerais

Analisando-se as ferramentas utilizadas, atualmente, para desenvolver *virtual appliances* nas diversas tecnologias de virtualização existentes, notou-se que há certa dificuldade, tanto para a criação quanto para a configuração desses VAs. Essa análise mostrou que alguns passos geralmente são seguidos:

- Instalação de uma tecnologia de virtualização – que envolve determinadas configurações no sistema hospedeiro;
- Criação da máquina virtual – que envolve a criação do disco rígido (ou imagem) e a configuração geral dessa máquina (memória principal, memória de vídeo, quais dispositivos de E/S essa máquina terá acesso, etc);
- Instalação de um sistema operacional;
- Configuração da rede – DHCP, DNS, rede interna ou externa, interface de rede, etc;
- Instalação de uma aplicação de propósito específico.

Geralmente, considera-se que o sistema já possua alguma tecnologia de virtualização instalada. Logo, essa etapa não seria contada como uma das etapas de criação de VAs (SUN et al., 2008).

3.2 Configuração da Rede dos *Virtual Appliances* para o VirtualBox

Entre as etapas descritas anteriormente, tem-se a configuração da rede dos *virtual appliances*, que em ambientes de grande porte (empresariais, acadêmicos, ...) pode trazer muito trabalho para administradores desses sistemas. Para se avançar no trabalho, pretendeu-se, então, automatizar essa etapa. Um exemplo em que essa automatização ajudaria seria em uma rede de *virtual appliances*, onde cada VA estaria conectado numa mesma rede de alguns outros VAs, e também pudesse acessar a Internet. Analisando as ferramentas do VirtualBox, notou-se que há três tipos de configurações de rede possíveis para serem aplicadas nas máquinas.

3.2.1 Rede com NAT

Por padrão é aplicado **Network Address Translation (NAT)**, que por roteamento¹, e utilizando um servidor DHCP integrado ao VirtualBox, gera um endereço IP para a máquina hospedeira numa rede privada e traduz esse endereço em um endereço real da máquina hospedeira, para enviar e receber pacotes pela Internet (ver figura 3.1).

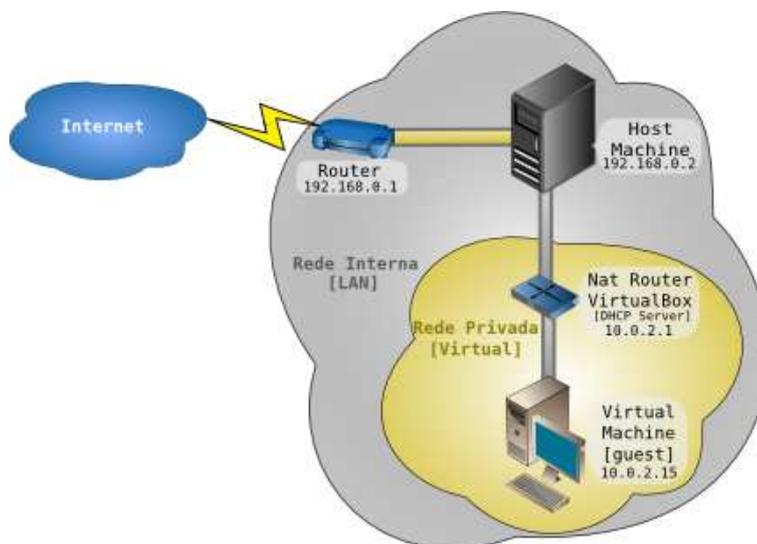


Figura 3.1: Estrutura da rede com NAT no VirtualBox. Fonte: (PINHEIRO, 2008)

Essa alternativa, porém, possui algumas limitações, tais como: a impossibilidade de acessar os VAs pela Internet, desabilitando o acesso a um servidor Web instalado nessas máquinas, por exemplo; o *Internet Control Message Protocol (ICMP)* – utilizado por depuradores de rede, como o aplicativo ping – não pode ser utilizado diretamente pelos

¹Encaminhamento de pacotes. Caminho percorrido por uma mensagem desde o ponto de origem até ao seu ponto de destino. (WIKIPEDIAc, 2008)

aplicativos do VirtualBox (HUNT, 2004; SUN MICROSYSTEMS, 2008).

No VirtualBox, a configuração do NAT pode ser de duas formas. Pela VirtualBox GUI: seleccionar a máquina virtual; ir em **Configurações -> Rede -> Conectado a**, e escolher NAT. Pela linha de comando, pode-se usar o VBoxManage da seguinte forma:

```
>VBoxManage modifyvm "nome da máquina" -nicX nat
```

onde **X** é o numero do adaptador de rede (de 1 a 4).

3.2.2 Rede com Interface no Hospedeiro

A segunda opção é configurar uma **interface no hospedeiro**. Essa interface é usada pela máquina hóspede para se conectar a Internet através de um *bridge*², criado no hospedeiro (ver figura 3.2). Nesse caso, cada VA precisa ter sua própria interface gerada no hospedeiro. A vantagem desse mecanismo é que o hóspede passa a fazer parte de uma rede acessível pela Internet. Um inconveniente desta técnica é a complexidade das configurações necessárias para aplicá-la na prática. Em (LINUXb, 2008; SUN MICROSYSTEMS, 2008) é mostrado uma possível solução para linux baseada na interface TAP (*Virtual Ethernet Network Device*):

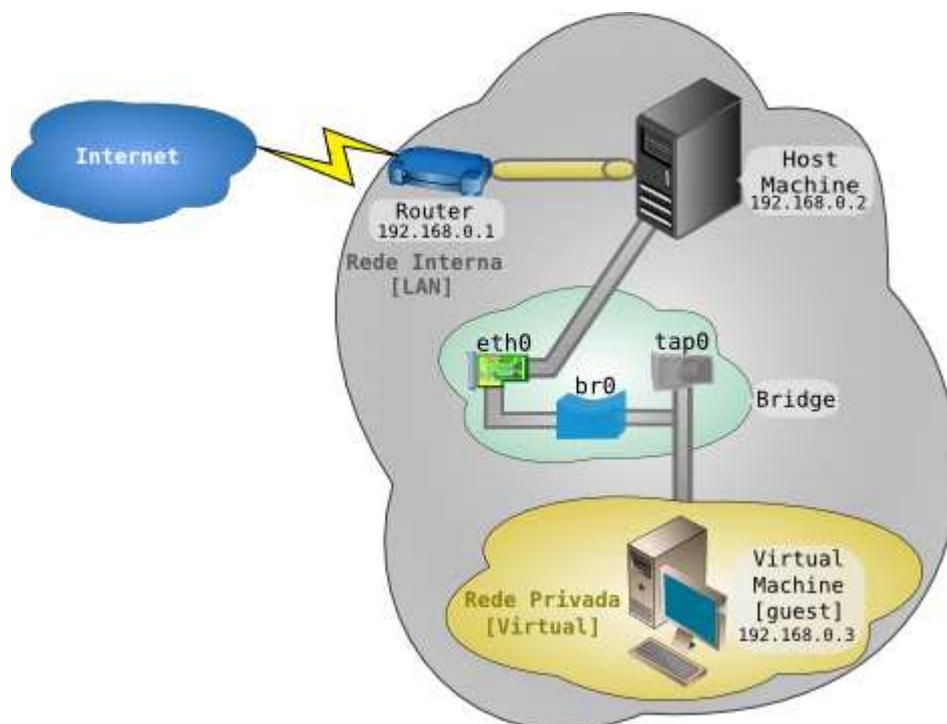


Figura 3.2: Estrutura da rede com Interface do Hospedeiro no VirtualBox. Fonte: (PINHEIRO, 2008)

²Usado para interligar duas redes.

1. Instalar os pacotes *bridge-utils* (para usar o comando **brctl**) e *uml-utilities* (para o comando **tunctl**). O comando **brctl** é utilizado para criar e gerenciar *bridges* ethernet e o comando **tunctl** é capaz de pré-configurar interfaces TAP para um determinado usuário;
2. Liberar o acesso ao dispositivo */dev/net/tun* para o usuário que irá rodar o VirtualBox. Quando o VirtualBox usa uma interface TAP, esse dispositivo é responsável pela criação e registro dessa interface no sistema hospedeiro.
 - **>chown usuario:usuario /dev/net/tun** ou
 - **>gpasswd -a usuario uml-net**
3. Criar um *bridge* (**br0**) e vinculá-lo à interface do hospedeiro (geralmente a **eth0**);
 - **>brctl addbr br0**
 - **>ifconfig eth0 0.0.0.0**
 - **>brctl addif br0 eth0**
4. Atualizar o endereço IP (neste caso está sendo utilizado DHCP);
 - **>dhclient br0**
5. Nesse ponto foi utilizado o *script*³ **VBoxAddIf** – que vem integrado ao VirtualBox – que cria uma interface TAP, vincula ao bridge criado e libera acesso ao usuário fornecido;
 - **>VBoxAddIf tap0 usuario br0**
6. Vincular a interface TAP criada à máquina virtual;
 - Pela VirtualBox GUI: **Configurações -> Rede -> Conectado a**, e escolher **Interface do Hospedeiro**; e em **Opções de Placa de Rede do Hospedeiro -> Nome da Placa de Rede**, digitar **tap0**;
 - Ou pela linha de comando: **>VBoxManage modifyvm “nome da máquina” -nic1 hostif -hostifdev1 tap0**

³Arquivo de texto com comandos que podem ser executados em seqüência.

3.2.3 Rede Interna

A última opção é utilizar uma rede interna (ver figura 3.3), a qual é fechada (sem conexão com a Internet) e pode ser acessada somente pelas máquinas pertencentes à rede (LINUXb, 2008). Para configuração desta, tem-se:

- Pela interface gráfica: **Configurações -> Rede -> Conectado a**, e escolher **Rede Interna**; e em **Nome da Rede**, inserir um nome, cuidando para ser o mesmo nome usado nas outras máquinas virtuais;
- Pela linha de comando: `>VBoxManage modifyvm “nome da máquina” -nic1 intnet -intnet1 “nome da rede”`

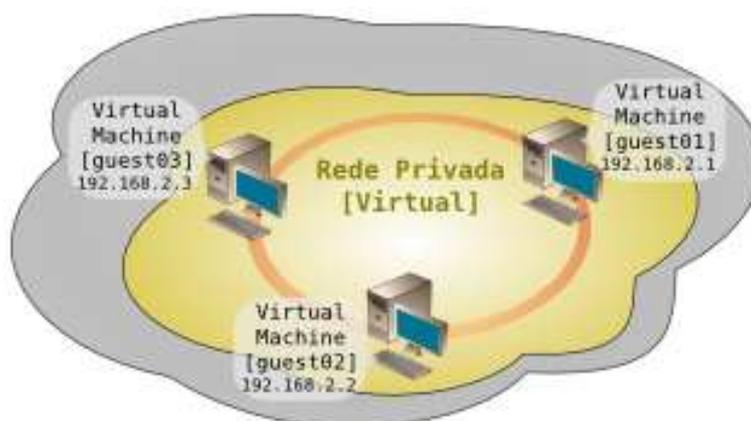


Figura 3.3: Estrutura usando Rede Interna no VirtualBox. Fonte: (PINHEIRO, 2008)

4 VIRTAK: FERRAMENTA DE CONFIGURAÇÃO AUTOMÁTICA DE REDE

Um dos ambientes desejados, e que foi utilizado como estudo de caso deste trabalho, foi o de uma rede de *virtual appliances* (VAN). Para isso utilizou-se a ferramenta desenvolvida VirtAK, capaz de configurar a rede desse ambiente. Duas abordagens foram utilizadas. Em ambas as abordagens, a idéia foi utilizar duas placas de rede – ou adaptadores – virtuais, oferecidas pela própria estrutura do VirtualBox (cada máquina suporta até quatro adaptadores de redes virtuais), cada qual utilizando um dos mecanismos de rede descritos anteriormente. Roda-se, então, um *script* na inicialização de cada VA, contendo os ajustes necessários para se habilitar os adaptadores de rede (WANG et al., 2007). Antes de desenvolver os *scripts*, bem como o sistema VirtAK, analisou-se e aplicou-se manualmente os procedimentos utilizados atualmente para se criar esse tipo de ambiente.

4.1 Configuração Manual

Na primeira abordagem, um dos adaptadores foi configurado com NAT – para conexão com a Internet – e o outro adaptador foi conectado a uma rede interna, que recebeu um nome, o qual foi transmitido para todos os VAs criados.

Na segunda abordagem, o primeiro adaptador recebeu uma interface do hospedeiro – possibilitando a conexão com a Internet – e o segundo adaptador, como na abordagem anterior, foi configurado com uma rede interna.

4.2 Configuração Automática

O trabalho desenvolvido buscou, então, automatizar, ou seja, reduzir o número de intervenções manuais em ações repetitivas, através de *scripts* e uma interface gráfica, os quais comporam o sistema VirtAK.

4.2.1 Os *Scripts*

Após a análise e configuração manual das abordagens utilizadas, partiu-se para a criação dos *scripts*, que servem para automatizar esse processo. Os *scripts* foram feitos em **Bash Script**¹, pois a maior parte das configurações da rede eram chamadas a *scripts* e comandos dessa linguagem.

Bash é um interpretador de comandos que foi criado em 1987, no projeto GNU, por Brian Fox. É um *software* livre, distribuído por padrão na maioria dos sistemas operacionais Unix-like (WIKIPEDIA, 2008).

Para este trabalho foram criados os seguintes *scripts*:

- **Principal** – responsável por toda a execução. Chama os outros *scripts* (Verificador, Rede e Desfaz Interface com Hospedeiro);
- **Verificador** – primeiro *script* chamado pelo Principal. Verifica a existência de *scripts* do VirtualBox e comandos necessários para o sistema de *scripts* funcionar. Dentre eles tem-se o VBoxManage, VBoxAddIF, dhclient, brctl e tuncctl;
- **Rede** – o “coração” do sistema de *scripts*. Responsável por chamar comandos de configuração da rede dos *virtual appliances*. Ele foi composto basicamente por funções feitas em bash, cada qual correspondendo a uma das possíveis configurações de rede do VirtualBox ou combinações destas. A figura 4.1 mostra uma das funções usadas para configurar a rede utilizando NAT;
- **Desfaz Interface com Hospedeiro** – como a interface com o hospedeiro gerada pelos *scripts* é permanente (até mesmo quando se reinicia a máquina hospedeira), esse *script* foi criado para desfazê-la, quando necessário. Basicamente ele desvincula as interfaces da máquina real e do *virtual appliance* do bridge criado, logo em seguida remove o bridge e atualiza o IP da máquina real.

Após a criação dos *scripts*, optou-se por gerenciá-los usando uma interface gráfica. Essa interface será descrita na próxima seção.

4.2.2 A interface gráfica

Trabalhos correlatos sugerem a integração de *scripts* em bash com alguma interface para gerenciamento destes. Esse tipo de integração é utilizada, por exemplo, quando se

¹Acrônimo de *Bourn Again Shell Script*

```
#####
##### NAT #####
#####
func_nat(){
echo
echo ">>Alguns dados para configuração do 'NAT'<<"
echo ">>Obs.: VAs não podem estar rodando."
echo

echo "Deseja configurar faixa IP do NAT?(Sim ou Não)(default \"Nao\")"
read IP_NAT_sim_nao

echo ">>>>>>INICIO DO LOG<<<<<<" >> log_nat.dat

while true
do
case "$IP_NAT_sim_nao" in
n* | N*)   VBoxManage modifyvm $VA_nome -nic1 nat >> log_nat.dat
           echo "Ok! NAT configurada para esse VA."
           echo ">>>>>>FIM DO LOG<<<<<<" >> log_nat.dat
           return 0;;
s* | S* )  echo "Digite a faixa IP desejada(formato X.X.X.X/X):"
           read VA_IP_NAT
           VBoxManage modifyvm $VA_nome -nic1 nat -natnet1 $VA_IP_NAT >> log_nat.dat
           echo "Ok! NAT configurada para esse VA."
           echo ">>>>>>FIM DO LOG<<<<<<" >> log_nat.dat
           return 0;;
* )       echo "Digite sim ou não!"
           read IP_NAT_sim_nao;;
esac
done
}
```

Figura 4.1: Função utilizada para configurar a rede com NAT.

quer implementar, em um laboratório de informática, um ambiente educacional para ensino de redes, sistemas operacionais, etc, mas deseja-se que o tempo gasto com configuração desse ambiente seja o mínimo possível (WALKER; MINYARD, 2004; SPINELLIS, 2005).

Dentre linguagens atuais, Java se destaca por possuir uma vasta API de suporte a programadores, bem como diversos tipos de ambientes de desenvolvimento integrados (IDE), que facilitam a geração de código, principalmente quando se trata de interfaces gráficas. Devido a essas qualidades, escolheu-se Java para a implementação da interface gráfica.

A interface foi estruturada conforme os parâmetros necessários pelos *scripts* em bash. Porém, algumas modificações nos *scripts* originais foram necessárias, pois além dos parâmetros serem agora recebidos pela interface, as funções foram transformadas em novos *scripts* onde para cada tipo de configuração de rede requisitada pela interface, um *script* foi criado. A figura 4.2 mostra a interface gráfica do **VirtAK**.

O VirtAK funciona da seguinte forma:

- Ao iniciar, ele roda o *script* **Verificador**. Se algum erro ocorrer na verificação, é mostrado uma mensagem informando o usuário, e a execução termina;
- Em seguida, o *script* **MaqXml** varre o arquivo **VirtualBox.xml** do VirtualBox que contém, entre outras coisas, o nome dos *virtual appliances* registrados, para o usuá-



Figura 4.2: Interface Gráfica do VirtAK.

rio poder escolher os que sofrerão a modificação da rede. Há a possibilidade de se configurar até quatro VAs com o mesmo tipo de rede, como mostra a figura 4.3;



Figura 4.3: VirtAK – Escolha dos VAs.

- Então, o usuário pode escolher, pelas abas da interface, qual configuração de rede os VAs terão. Essa configuração foi dividida em dois tipos:
 - **Simples** – NAT, Rede Interna ou Interface com Hospedeiro (como mostrado na figura 4.4);

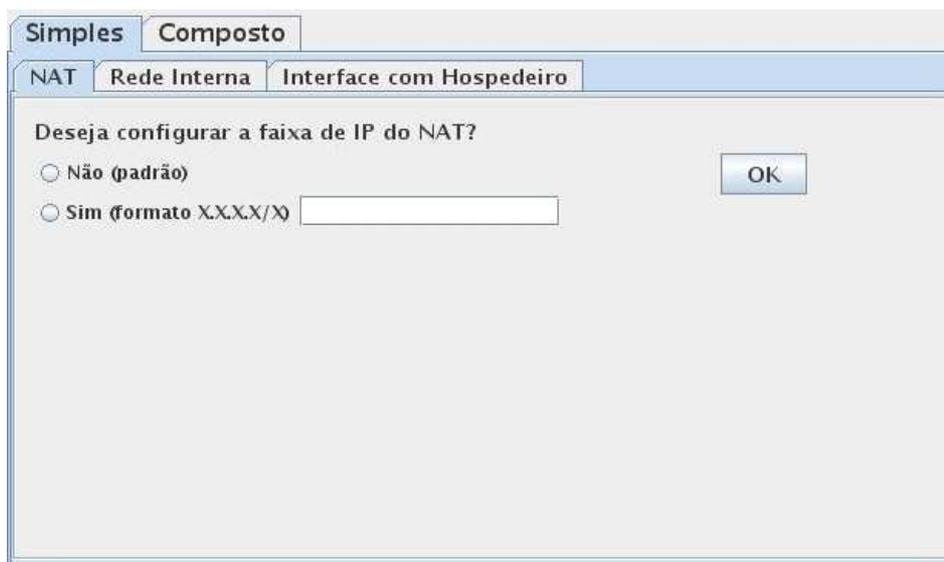


Figura 4.4: VirtAK – Configuração simples.

- **Composta** – NAT + Rede Interna ou Interface com Hospedeiro + Rede Interna (como mostra a figura 4.5);

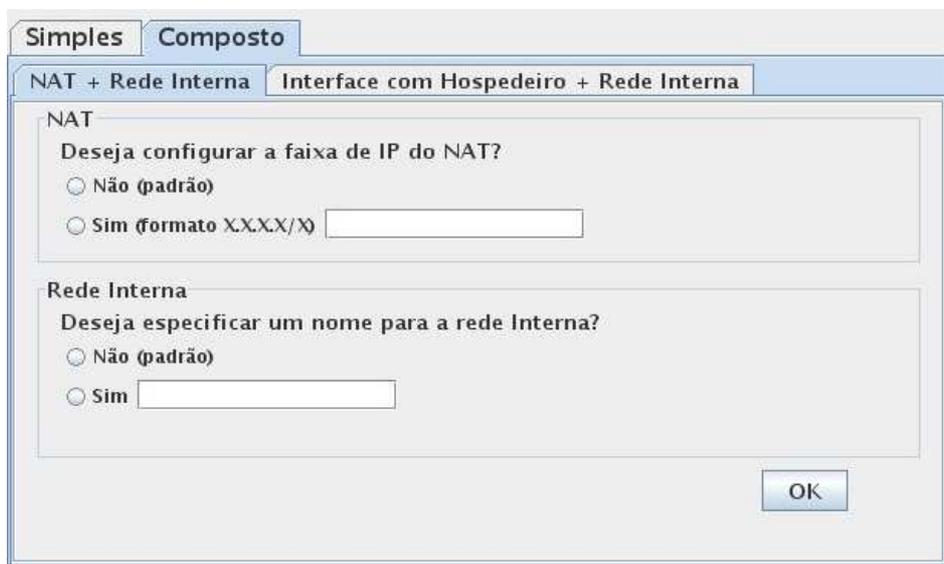


Figura 4.5: VirtAK – Configuração composta.

- Para cada configuração, alguns parâmetros são necessários: Qual a interface de rede utilizada pelo host, se utiliza DHCP ou IP fixo, etc;

Exemplos de cenários possíveis de serem criados utilizando o VirtAK são mostrados na figura 4.6, onde se tem o cenário Simples, e na figura 4.7 o Composto. Em ambos os cenários foi utilizado LAMP com a seguinte configuração: sistema operacional Linux

(distribuição Ubuntu 8.04), servidor Apache 2.0, banco de dados MySQL 5.0 e interpretador PHP 5.

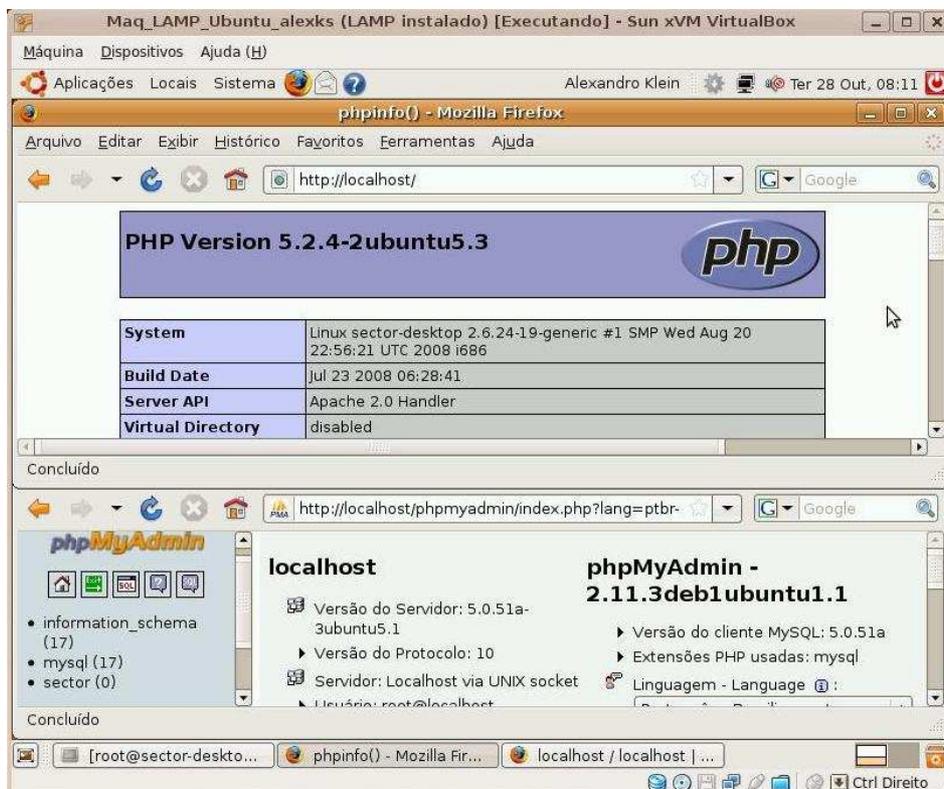


Figura 4.6: VA LAMP em detalhes: phpinfo¹ e phpmyadmin²

Optou-se pela criação de LAMP como caso de teste, pois esse tipo de ambiente requer uma configuração inicial de rede para funcionar, e também porque esses aplicativos, de forma integrada, são geralmente utilizados para soluções em ambientes de servidores Web (SUN et al., 2008).

4.3 Comparação entre VirtualBox GUI e VirtAK

Esta seção tem por objetivo discutir as maiores diferenças encontradas ao se comparar a ferramenta utilizada pela VirtualBox GUI para configuração da rede (figura 4.8) com o VirtAK, desenvolvido neste trabalho.

Nota-se que ambas possuem uma interface amigável para configuração da rede por NAT e Rede Interna, pois com poucos cliques tem-se a configuração desejada. Porém, no VirtAK, quando cria-se a rede por NAT, tem-se a possibilidade de mudar a faixa IP dos VAs. Além disso, todas as modificações podem ser refletidas em até quatro *virtual*

¹Função em PHP que mostra informações sobre o sistema.

²Interface gráfica para gerência do banco de dados MySQL.

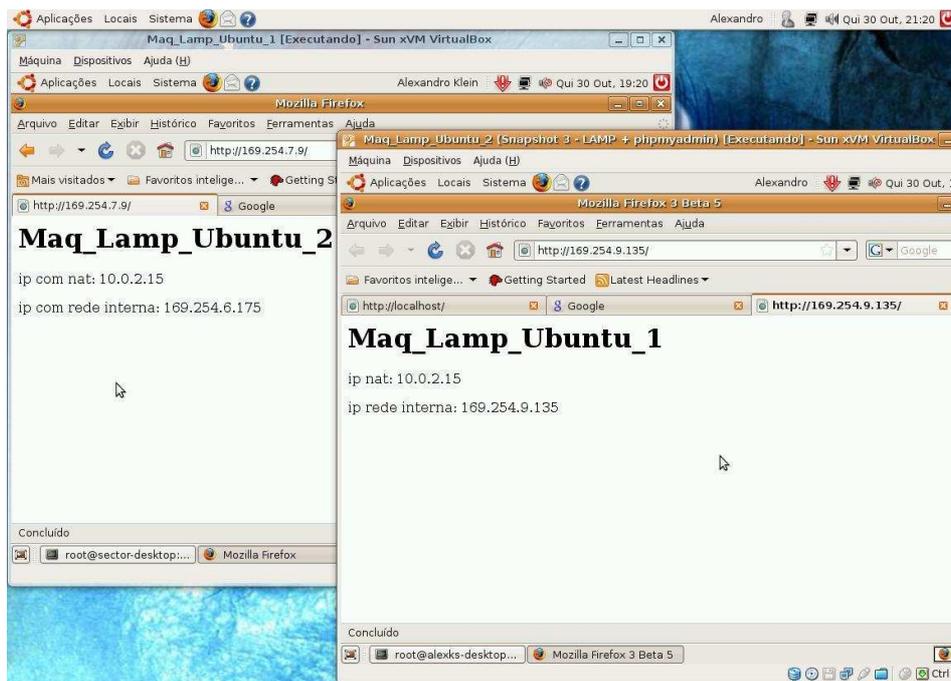


Figura 4.7: Duas VAs LAMP com NAT e Rede Interna



Figura 4.8: Configurações de Rede

appliances, o que na VirtualBox GUI só são possíveis de serem feitas num VA por vez. Essa última característica se deve ao fato de que o VirtAK foi projetado com o objetivo principal de configurar diversos VAs de forma facilitada.

Já no caso de configuração da rede por Interface com o Hospedeiro, a VirtAK se sobressai, pois os únicos parâmetros necessários são: a interface utilizada pelo *host*, e se

será utilizado DHCP ou IP fixo para esse *host*. Para o VirtualBox GUI, as configurações são baseadas no fato de já existir uma interface no hospedeiro, na qual o VA poderia utilizar. Então o único parâmetro a ser passado seria o nome da interface. Porém, na maioria das vezes, deve-se criar a interface que será utilizada pelo hóspede, e essa criação envolve todos os passos descritos anteriormente (seção 3.2.2). No VirtAK essas informações são abstraídas.

Além disso, para se ter ambientes equivalentes aos Compostos do VirtAK na VirtualBox GUI, deve-se habilitar dois adaptadores de rede virtuais e configurá-los para cada tipo de rede desejada, e isso tudo uma máquina por vez. No VirtAK, com no máximo três parâmetros tem-se esse ambiente configurado para até quatro *virtual appliances*.

5 CONSIDERAÇÕES FINAIS

Com o aumento do poder tecnológico dos computadores pessoais, surgiram novas idéias para se utilizar melhor desses recursos. A virtualização trouxe alguns benefícios como segurança e portabilidade, aliada à melhor utilização dos recursos computacionais através da abstração e gerência destes, realizada pelas suas máquinas virtuais.

Unindo a virtualização aos mais diversos aplicativos existentes, surgiram os *virtual appliances*, que se mostraram capazes de diminuir problemas relacionados a administração de sistemas, redirecionando o foco do trabalho para etapas mais relevantes, como a utilização dos aplicativos.

Entre os problemas mais frequentes e complexos enfrentados por esses administradores, tem-se a manipulação da rede. Embora com a utilização da virtualização esse problema seja minimizado, ainda sim, quando se pensa em gerenciar um número relativamente grande de *virtual appliances*, o problema pode ser maximizado.

Pensando nisso, esse trabalho propôs estudar e elaborar uma solução relacionada e gerência de diversos *virtual appliances*. Dessa forma, foi possível criar um sistema capaz de configurar a rede desses VAs, o VirtAK, que se utiliza de *scripts* de configuração, bem como uma interface gráfica para melhor manipulá-los.

Acredita-se que o trabalho tenha contribuído para a pesquisa científica, e que futuramente o VirtAK possa ser melhorado, e/ou até mesmo incorporado ao próprio VirtualBox.

REFERÊNCIAS

AMAZON WEB SERVICES. **Amazon web services**. Disponível em: <http://aws.amazon.com/>. Acesso em: dezembro de 2008.

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEBAUER, R.; PRATT, I.; WARFIELD, A. Xen and the art of virtualization. **SIGOPS Oper. Syst. Rev.**, New York, NY, USA, v.37, n.5, p.164–177, 2003.

BR-LINUX.ORG. **O que são virtual appliances**. Disponível em: <http://br-linux.org/2008/o-que-sao-virtual-appliances/>. Acesso em: dezembro de 2008.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-Oriented Cloud Computing: vision, hype, and reality for delivering it services as computing utilities. **High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on**, [S.l.], p.5–13, Sept. 2008.

HELPDESKLIVE. **helpdesklive**. Disponível em: <http://helpdesklive.info/download/VirtualBox VDI free images.html>. Acesso em: dezembro de 2008.

HUNT, C. **Linux**: servidores de rede. Rio de Janeiro: Ciência Moderna: [s.n.], 2004.

INNOTEK. **Innotek - VirtualBox**. Disponível em: <http://www.virtualbox.org/>. Acesso em: dezembro de 2008.

JUMPBOX, INC. **JumpBox**. Disponível em: <http://www.jumpbox.com/>. Acesso em: dezembro de 2008.

KRISHNAMURTI, S. **Get Juiced!** Disponível em: <http://blogs.vmware.com/console/2007/07/get-juiced.html>. Acesso em: dezembro de 2008.

LAUREANO, M. **Máquinas Virtuais e Emuladores**: conceitos, técnicas e aplicações. São Paulo: Novatec Editora: [s.n.], 2006.

LAURIANNE MCLAUGHLIN, C. **Virtual Software Appliances**: why they could end deployment hell. Disponível em: <http://www.cio.com/article/print/170051>. Acesso em: dezembro de 2008.

LINUXa. **Sun acquire Innotek**. Disponível em: http://www.linuxnewmedia.com.br/noticia/sun_acquire_innotek. Acesso em: dezembro de 2008.

LINUXb. **Virtualização em 2008**. Disponível em: http://www.linuxmagazine.com.br/issue/lm_40_virtualizacao. Acesso em: dezembro de 2008.

LINUX.COM. **A virtual appliance primer**. Disponível em: <http://www.linux.com/feature/138166>. Acesso em: dezembro de 2008.

PINHEIRO, K. **VirtualBox, Configurações de rede**. Disponível em: <http://even.archlinux-br.org/blog/virtualbox-configuracoes-de-rede>. Acesso em: dezembro de 2008.

RPATHa. **rPath**. Disponível em: <http://www.rpath.com/corp/>. Acesso em: dezembro de 2008.

RPATHb. **White Paper - Best Practices for Building Virtual Appliances**. Disponível em: http://www.rpath.com/corp/images/stories/white_papers/WP_BestPractices.pdf. Acesso em: dezembro de 2008.

SAPUNTZAKIS, C.; BRUMLEY, D.; CHANDRA, R.; ZELDOVICH, N.; CHOW, J.; LAM, M. S.; ROSENBLUM, M. Virtual Appliances for Deploying and Maintaining Software. In: IN PROCEEDINGS OF THE SEVENTEENTH LARGE INSTALLATION SYSTEMS ADMINISTRATION CONFERENCE (LISA 2003), 2003. **Anais...** [S.l.: s.n.], 2003. p.181–194.

SPINELLIS, D. Java makes scripting languages irrelevant? **Software, IEEE**, [S.l.], v.22, n.3, p.70–71, May-June 2005.

SUN, C.; HE, L.; WANG, Q.; WILLENBORG, R. Simplifying Service Deployment with Virtual Appliances. In: SCC '08: PROCEEDINGS OF THE 2008 IEEE INTERNATIONAL CONFERENCE ON SERVICES COMPUTING, 2008, Washington, DC, USA. **Anais...** IEEE Computer Society, 2008. p.265–272.

SUN MICROSYSTEMS. **Sun xVM VirtualBox**. [S.l.: s.n.], 2008. Disponível em: <http://www.virtualbox.org/download/1.6.2/UserManual.pdf>. Acesso em: dezembro de 2008.

SWSOFT. **Server Virtualization Open Source Project**. Disponível em: <http://openvz.org/>. Acesso em: dezembro de 2008.

VIRTUALAPPLIANCES.NET. **Virtual Appliances**. Disponível em: <http://virtualappliances.net/>. Acesso em: dezembro de 2008.

VMWARE, INC.a. **VMware: virtualization, virtual machine & virtual server consolidation - vmware**. Disponível em: <http://www.vmware.com/>. Acesso em: dezembro de 2008.

VMWARE, INC.b. **Virtual Appliance Marketplace, Virtual Appliances & VMware Appliance - VMware**. Disponível em: <http://www.vmware.com/appliances/>. Acesso em: dezembro de 2008.

VMWARE, INC.c. **White Paper - Best Practices for Building Virtual Appliances**. Disponível em: http://www.vmware.com/files/pdf/vam_best_practices_building.pdf. Acesso em: dezembro de 2008.

WALKER, E.; MINYARD, T. Orchestrating and coordinating scientific/engineering workflows using GridShell. **High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on**, [S.l.], p.270–271, June 2004.

WANG, X.; LAN, D.; WANG, G.; FANG, X.; YE, M.; CHEN, Y.; WANG, Q. Appliance-Based Autonomic Provisioning Framework for Virtualized Outsourcing Data Center. In: ICAC '07: FOURTH INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING, 2007, Beijing, China. **Anais...** IEEE Computer Society, 2007. p.29–29.

WIKIPEDIAa. **Virtual Appliance**. Disponível em: http://en.wikipedia.org/wiki/Virtual_appliance. Acesso em: dezembro de 2008.

WIKIPEDIAc. **Roteamento**. Disponível em: <http://pt.wikipedia.org/wiki/Roteamento>. Acesso em: dezembro de 2008.

WIKIPEDIAc. **Bash**. Disponível em: <http://en.wikipedia.org/wiki/Bash>. Acesso em: dezembro de 2008.