

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

**UM SISTEMA AUTOMATIZADO PARA
CRIAÇÃO DE MÁQUINAS VIRTUAIS**

TRABALHO DE GRADUAÇÃO

Márcio Parise Bouffleur

Santa Maria, RS, Brasil

2008

UM SISTEMA AUTOMATIZADO PARA CRIAÇÃO DE MÁQUINAS VIRTUAIS

por

Márcio Parise Bouffleur

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° 250
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

**UM SISTEMA AUTOMATIZADO PARA CRIAÇÃO DE
MÁQUINAS VIRTUAIS**

elaborado por
Márcio Parise Bouffleur

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

COMISSÃO EXAMINADORA:

Prof^a Dr^a Andrea Schwertner Charão
(Presidente/Orientador)

Prof. Dr. Benhur de Oliveira Stein (UFSM)

Prof. MSc. João Carlos Damasceno Lima (UFSM)

Santa Maria, 1 de fevereiro de 2008.

AGRADECIMENTOS

O desenvolvimento deste trabalho deve agradecimentos para muitas pessoas, seja pelo apoio técnico ou por motivos de ordem subjetiva. Primeiramente, agradeço à Juvi pelo apoio, carinho, dedicação e incentivo não só durante o desenvolvimento do trabalho, como durante toda a graduação e anteriormente. Obrigado por tudo.

Agradeço à professora Andrea, por ter me ensinado escrever de forma clara e objetiva. Aos outros professores do LSC e do GMOB: obrigado por propiciarem um local onde é possível criar e desenvolver coisas novas, apesar de todas as adversidades.

Aos amigos Tarta, Nareba, Cris e Tony, por todas as partidas de sinuca, Battlefield e UrbanTerror. Não fosse por eles, não estaria no caminho do bem.

Por último e não menos importante, gostaria de agradecer ao professor Pasin pela doação da cafeteira para o laboratório, utensílio este indispensável à completude desse e demais trabalhos.

“Part of the inhumanity of the computer is that, once it is competently programmed and working smoothly, it is completely honest.” — ISAAC ASIMOV

RESUMO

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

UM SISTEMA AUTOMATIZADO PARA CRIAÇÃO DE MÁQUINAS VIRTUAIS

Autor: Márcio Parise Bouffleur

Orientador: Prof^a Dr^a Andrea Schwertner Charão

Local e data da defesa: Santa Maria, 1 de fevereiro de 2008.

No ambiente computacional atual, o desperdício de recursos computacionais por meio da subutilização de processadores é muitas vezes inaceitável. Virtualização apresenta-se como uma alternativa eficaz para o melhor aproveitamento de recursos através da consolidação de diversas máquinas em único servidor. A administração de ambientes virtualizados pode ser complicada dependendo do número de máquinas virtuais que o administrador deseja criar, pois a instalação das mesmas pode consumir um tempo considerável. Para tentar sanar essa deficiência, é necessário um método de criação mais eficiente de novas máquinas virtuais. Para solucionar este problema, este trabalho apresenta a integração da ferramenta de administração de ambientes virtualizados Virtual Machine Manager (virt-manager) ao sistema de provisionamento Cobbler. A ferramenta virt-manager permite o gerenciamento de monitores de máquinas virtuais independentemente da solução de virtualização utilizada, pois é fundamentada na biblioteca libvirt, que mantém uma API consistente para gerenciamento de diversos hipervisores. Cobbler é um sistema que permite a instalação e configuração de novos sistemas operacionais de forma rápida e eficiente. Com este trabalho, pretende-se aprimorar a ferramenta virt-manager, dotando a mesma com a capacidade de criação de novas máquinas virtuais utilizando perfis do sistema Cobbler, desta forma livrando o administrador da necessidade de instalar manualmente as mesmas.

Palavras-chave: Virtualização, Xen, máquinas virtuais, administração de ambientes virtualizados, libvirt, Cobbler, provisionamento.

ABSTRACT

Graduation Work
Undergraduate Program in Computer Science
Federal University of Santa Maria

AN AUTOMATED SYSTEM FOR VIRTUAL MACHINE CREATION

Author: Márcio Parise Bouffleur

Advisor: Prof^a Dr^a Andrea Schwertner Charão

In the current computer environment, the waste of computer resources by poor utilization is often unacceptable. Virtualization shows itself as an effective alternative to a better resources usage by the consolidation of several machines into a unique server. The administration of virtualized environments can be hard depending on the number of virtual machines the administrator wants to create, because its installation can consume a serious amount of time. One needs a better installation method of virtual machines to remedy this deficiency. This work presents the integration of the virtual environment administrative tool Virtual Machine Manager (virt-manager) with the Cobbler provisioning system. The virt-manager tool allows the management of virtual machine monitors independently of the virtualization solution used, because it is built on the libvirt library, which maintains a consistent API to manage several hipervisors. Cobbler is a system that allows the rapid and efficient installation and configuration of new operating systems. This work had the intent to enhance the tool virt-manager, by giving it means to create new virtual machines using the Cobbler system's profiles, this way freeing the administrator of the necessity of manually installing them.

Keywords: virtualization, Xen, virtual machines, virtualized environment management, libvirt, Cobbler, provisioning.

LISTA DE FIGURAS

Figura 2.1 – Exemplo de código Python utilizando a libvirt	18
Figura 2.2 – Acesso remoto através do serviço libvirtd	19
Figura 2.3 – virt-manager: tela principal	20
Figura 2.4 – Visão geral dos sistemas. Fonte: (Red Hat, Inc, 2007)	22
Figura 3.1 – Ambiente virtualizado	24
Figura 3.2 – Tela de seleção da fonte de instalação e kickstarts	25
Figura 3.3 – Descoberta de um servidor Cobbler e listagem de perfis	27
Figura 3.4 – Acessando um servidor Cobbler	28
Figura 4.1 – Ambiente de testes	31
Figura 4.2 – Criação de um novo perfil no sistema Cobbler	31
Figura 4.3 – Selecionando um perfil específico	32
Figura 4.4 – Padrão de nomes criados por virt-manager	32

LISTA DE ABREVIATURAS E SIGLAS

MMV	Monitor de Máquinas Virtuais
MV	Máquina Virtual
SO	Sistema Operacional
API	Application Programming Interface
GPL	General Public License
WEB	World Wide Web
RPC	Remote Procedure Call
XML	Extensible Markup Language
SSH	Secure Shell
GTK	Gimp Toolkit

SUMÁRIO

1	INTRODUÇÃO	11
2	ADMINISTRAÇÃO DE AMBIENTES VIRTUALIZADOS	14
2.1	Monitores de Máquinas Virtuais	14
2.1.1	Soluções de Virtualização Disponíveis	14
2.1.2	Alternativas à Implementação Clássica	16
2.2	Ferramentas Administrativas	17
2.3	libvirt	17
2.3.1	Utilização	18
2.3.2	libvirtd	19
2.4	Virtual Machine Manager – virt-manager	19
2.4.1	Características Principais	19
2.4.2	Limitações	21
2.5	Cobbler	21
2.6	Interrelacionamento entre as Soluções	22
3	PROJETO E DESENVOLVIMENTO	23
3.1	Requisitos e Funcionalidades Desejadas	23
3.2	Implementação	24
3.2.1	Assistente de Criação de Máquinas Virtuais	25
3.2.2	Encontrando um Servidor Cobbler	26
3.2.3	Koan e a Arte de Descobrir Servidores	26
3.2.4	Notificando o Usuário da Presença de um Servidor Cobbler	27
4	AVALIAÇÃO	29
4.1	Utilização da Solução	29
4.1.1	Criando Novos Perfis	30
4.1.2	Configuração do Servidor DHCP	30
4.1.3	Ativando Todos os Atores	30
4.2	Fazenda de Servidores Web	30
4.2.1	Criação dos Servidores	31
4.2.2	Problemas Encontrados	32
5	CONCLUSÃO	34
	REFERÊNCIAS	36

1 INTRODUÇÃO

Avanços recentes no desenvolvimento de computadores proporcionaram a retomada da pesquisa sobre virtualização de recursos computacionais. A virtualização permite a execução concorrente de diversos sistemas operacionais sobre um mesmo computador, provendo um melhor aproveitamento de recursos.

Virtualização pode ser definida como uma abstração que se sobrepõe a um determinado *hardware*, de modo a prover uma interface consistente a camadas acima desta. Dessa forma, é possível controlar o acesso ao *hardware* real do sistema, o que permite o escalonamento entre as diversas peças de *software* que executam sobre a camada de virtualização. Tal camada é comumente denominada Monitor de Máquinas Virtuais (MMV) ou hipervisor.

Atualmente, existem diversas soluções que utilizam técnicas de virtualização. Um exemplo de solução de virtualização é a plataforma Java (ARNOLD; GOSLING, 1998), que provê uma máquina virtual que interpreta o código intermediário gerado pelo compilador e executa as instruções geradas na arquitetura destino. Dentre as soluções que abstraem todo o *hardware*, os MMV's mais populares são VMWare (SUGERMAN; VENKITACHALAM; LIM, 2001), Xen (BARHAM et al., 2003) e OpenVZ (SWsoft, 2007). Devido ao número de soluções disponíveis atualmente e suas respectivas particularidades, a administração de ambientes virtualizados que utilizam mais de uma solução de virtualização pode se tornar um sério entrave. Uma forma centralizada de gerenciamento de ambientes virtualizados que seja agnóstica em relação aos MMV's torna-se imprescindível ao administrador.

Além do problema do gerenciamento de novas máquinas virtuais, o processo de instalação de novas máquinas pode ser extremamente oneroso para o administrador. Isto ocorre principalmente pelo fato de ser necessária a intervenção por parte do administra-

dor para configurar cada nova máquina criada. Assim, um sistema que além de gerenciar hipervisores independentemente da plataforma de virtualização escolhida, possa instalar novas máquinas de maneira automática, pode facilitar consideravelmente a administração de ambientes virtualizados.

Para preencher tal lacuna, Veillard et. al. iniciaram, em 2005, o desenvolvimento da biblioteca *libvirt* para lidar com diversos hipervisores (VEILLARD et al., 2005). Esta biblioteca abstrai as idiosincrasias de cada monitor, de modo que uma ferramenta que a utilize possa gerenciar máquinas virtuais independentemente do monitor em uso em determinado servidor.

A biblioteca *libvirt* serve de base para uma ferramenta administrativa gráfica chamada *virt-manager* (BERRANGE et al., 2007). Esta ferramenta permite a administração remota de máquinas virtuais, bem como a extração de informações sobre o estado das mesmas. Tudo isso é feito através de uma interface gráfica centralizada, o que melhora sensivelmente a usabilidade da mesma.

Uma deficiência de *virt-manager* é a falta de um método centralizado para a criação de novas máquinas virtuais. Em cenários onde é necessária a criação de diversas máquinas virtuais de forma rápida e eficiente, como por exemplo um conjunto de servidores *Web* ou um aglomerado virtual de computadores, a utilização de uma ferramenta de administração de ambientes virtualizados para criar tais configurações pode tornar-se muito custosa. Um método semi-automatizado de criação de novas máquinas virtuais pode facilitar a implantação de grandes parques virtualizados de forma rápida e fácil.

Uma solução para o problema da instalação automatizada de máquinas virtuais é a ferramenta de criação de imagens *Cobbler* (DEHAAN, 2007). *Cobbler* é um sistema que permite a instalação de novas máquinas a partir de perfis de instalação que são armazenados no servidor *Cobbler*. Associada ao *Cobbler*, existe a aplicação cliente *Koan*, que faz par ao mesmo e permite a reinstalação de nós virtualizados, bem como a reinstalação de sistemas já existentes.

Contudo, o sistema *Cobbler* fornece suporte à criação de novas máquinas virtuais apenas de distribuições Linux da Redhat ou derivadas. Essa limitação compromete em parte a utilidade da solução, pois confina o administrador a utilizar somente estes produtos. Porém, é necessário realçar que trata-se de um sistema utilizado por uma grande parcela dos usuários do sistema operacional Linux e o suporte da distribuição Redhat a ambientes

virtualizados é um dos melhores disponíveis atualmente. No presente momento, não foi encontrada uma solução de provisionamento com flexibilidade, estabilidade e qualidade comparáveis à ferramenta Cobbler, fato este que levou a sua escolha para o sistema de instalação automática.

A proposta deste trabalho é o desenvolvimento de um sistema que permita a administradores de ambientes virtualizados a criação automática de novas máquinas virtuais. Para tal, será feita a integração da ferramenta administrativa gráfica virt-manager ao sistema de criação de imagens Cobbler. Dessa forma, um usuário da ferramenta virt-manager poderia criar novas máquinas virtuais de forma simples, selecionando o sistema alvo a ser criado e onde o mesmo executará. As ferramentas acima mencionadas possuem uma comunidade de desenvolvimento ativa. Assim, espera-se enviar os resultados obtidos a esta comunidade, de modo a integrar os mesmos à versão que é disponibilizada a todos os usuários destas ferramentas.

2 ADMINISTRAÇÃO DE AMBIENTES VIRTUALIZADOS

Neste capítulo será feita uma revisão sobre tecnologias de virtualização e suas diversas implementações. Será tratado também sobre a biblioteca libvirt, que permite a abstração dos monitores de máquinas virtuais subjacentes. A ferramenta virt-manager, que utiliza libvirt como fundação, é apresentada a seguir. Posteriormente, será apresentada a ferramenta Cobbler e seu aplicativo cliente Koan.

2.1 Monitores de Máquinas Virtuais

Um monitor de máquinas virtuais (MMV) é uma peça de *software* que atua sobre o *hardware* subjacente, provendo assim uma interface que gerencia e aloca recursos para os sistemas operacionais que executam sobre o MMV. Dessa forma, é criada uma abstração do *hardware* físico pelo hipervisor, garantindo isolamento no acesso concorrente aos recursos do sistema.

A forma mais comum de utilização de MMV é justamente o posicionamento de uma camada de *software* diretamente sobre o *hardware*, controlando o acesso das máquinas virtuais ao *hardware* real. A esse modelo dá-se o nome de implementação clássica (SMITH; NAIR, 2005).

2.1.1 Soluções de Virtualização Disponíveis

Implementações dessa técnica são encontradas no mercado em diversos níveis de maturação e adoção. A lista abaixo mostra algumas das soluções mais populares:

- **z/VM** é o sistema operacional de máquinas virtuais provido pela IBM. Tem suas raízes no antigo System/360 (BORNTRAGER; SCHWIDEFSKY, 2005; TURK; BAUSCH, 2005). É utilizado para prover máquinas virtuais em *mainframes*, provendo assim até milhares de instâncias executando concorrentemente. É utilizado

em grandes sistemas, como bancos e instituições governamentais.

- **Xen** é um monitor de máquinas virtuais de código aberto desenvolvido pela Universidade de Cambridge e atualmente mantido pela empresa XenSource. Suas principais características são o desempenho (BARHAM et al., 2003; CLARK et al., 2004; MATTHEWS et al., 2007), relocação de máquinas virtuais sem interrupção (*Live Migration*) (CLARK et al., 2005) e suporte às arquiteturas IA32, x86-64, IA64 e PowerPC.
- **VMWare** é um sistema de virtualização proprietário para a plataforma IA32 (VMWare, Inc., 2007). Desenvolvido pela VMWare Inc., é um dos sistemas mais populares, devido à sua facilidade de utilização e bom desempenho (SUGERMAN; VENKITACHALAM; LIM, 2001), tanto em sistemas domésticos como em ambientes corporativos.
- **OpenVZ** é um sistema livre desenvolvido e mantido pela empresa SWSoft (SWsoft, 2007). OpenVZ utiliza a técnica de virtualização em nível de sistema operacional, que permite somente a execução de diferentes instâncias do mesmo sistema operacional. Essa abordagem diminui a flexibilidade da solução, porém provê um ganho de desempenho considerável.
- **KVM** é um acrônimo para *Kernel Based Virtual Machine* (Kernel Based Virtual Machine, 2007). KVM consiste de um módulo de *software* que é carregado junto ao núcleo do sistema operacional Linux para prover suporte à virtualização em máquinas que utilizam processadores com suporte nativo à virtualização, como Intel VT e AMD-V.
- **Parallels Workstation** (Parallels, Inc., 2006) é um conjunto de aplicações de virtualização para a arquitetura x86-64. Possui versões para Windows, Linux e MacOS X, sendo sua utilização bem popular entre usuários do último.
- **Microsoft Virtual Server** é uma plataforma de virtualização desenvolvida pela Microsoft (ARMSTRONG, 2007). Possui versões para Windows XP e Windows Server 2003.
- **QEMU** é um emulador e uma plataforma de virtualização genérica de código aberto.

QEMU possui suporte à diversas arquiteturas e funciona também como plataforma de virtualização, utilizando-se o módulo KQEMU (BELLARD, 2005).

2.1.2 Alternativas à Implementação Clássica

Devido à ausência de suporte nativo a ambientes virtualizados na arquitetura IA32 original, três métodos de virtualização foram propostos e implementados pelas principais soluções.

- **Emulação:** a técnica de emulação simplesmente traduz as instruções executadas pelo sistema hospedado em funções que são interpretadas pelo sistema hospedeiro. Apesar da perda drástica de desempenho devido à tradução, a emulação é vantajosa pois permite a execução de sistemas onde até a arquitetura alvo do sistema hospedeiro difere do sistema hospedado. Dois sistemas que utilizam emulação são o DOSBox (DOSBox, 2007) e QEMU (BELLARD, 2005).
- **Paravirtualização:** este método de implantação consiste em criar uma arquitetura alvo levemente diferente da arquitetura do sistema hospedeiro. Dessa forma, o núcleo do sistema operacional hospedado necessita ser portado para tal arquitetura, de modo a interagir corretamente com o MMV que executa sobre o *hardware*. Cabe ressaltar que é necessária apenas a adaptação do núcleo do sistema operacional, assim a interface binária entre o sistema operacional e os programas de espaço de usuário é inalterada, garantindo a execução dos mesmos sem alterações e com desempenho semelhante à execução nativa sobre o *hardware* sem o hipervisor. Um exemplo dessa abordagem é o MMV Xen (BARHAM et al., 2003; CLARK et al., 2004).
- **Tradução binária em tempo de execução:** esta abordagem é feita por algumas versões do sistema VMWare (SUGERMAN; VENKITACHALAM; LIM, 2001). Consiste na tradução e adaptação das chamadas do sistema operacional hospedado de modo a garantir que o mesmo tenha a ilusão de estar executando solitariamente no *hardware*. Este método é uma melhoria em relação à emulação pura, pois grande parte do código executado pelas aplicações de espaço de usuário no sistema hospedado pode ser executada diretamente sobre o *hardware*, descartando a necessidade de tradução das mesmas.

2.2 Ferramentas Administrativas

Atualmente, diversas ferramentas competem entre si para ajudar administradores de sistemas nas tarefas de gerenciamento de ambientes virtualizados. VMWare possui uma ferramenta administrativa que permite a manipulação de todos os aspectos das máquinas virtuais criadas neste sistema. Contudo, a mesma não dispõe de nenhuma forma de automatização da instalação de novos sistemas hospedados, dependendo da utilização de ferramentas providas pelos fabricantes dos sistemas operacionais que serão hospedados para execução desta tarefa.

Para a arquitetura Xen, as ferramentas Xen Enterprise (Xen Source Inc., 2006) e Enomalism (Enomaly Inc, 2005) apresentam algumas funcionalidades para a criação automática de novas máquinas virtuais. Basicamente, as duas soluções fazem uso de imagens de sistemas pré-instalados que são copiadas integralmente para as novas máquinas virtuais. Após a cópia dessas novas imagens, faz-se necessária a adaptação de parâmetros como nome e endereço de rede das novas MV's.

Ainda na linha de automatização de instalação de novas máquinas virtuais em Xen, a suíte de aplicativos Xen-Tools (KEMP, 2006) permite a instalação de novas máquinas virtuais, porém a mesma é utilizada apenas em interface de linha de comando. Outra solução é o sistema MLN, que permite a criação de novas máquinas virtuais utilizando uma linguagem de descrição de ambientes virtualizados (BEGNUM, 2006). MLN utiliza a imagens pré-configuradas, porém pode integrar-se à ferramenta de gerenciamento de configurações cfengine (BURGESS; RALSTON, 1997).

2.3 libvirt

A administração de diversas soluções de virtualização muitas vezes torna-se muito custosa e enfadonha: cada MMV possui sua própria forma de gerenciar os sistemas hospedados. A necessidade de se criar uma ferramenta unificada para lidar com as idiosincrasias de cada sistema torna-se imperativa, visto que não existe uma consistência de conceitos e ferramentas entre as soluções existentes.

Com o intuito de criar uma forma centralizada e definitiva de manipular máquinas virtuais, foi criada a biblioteca libvirt (VEILLARD et al., 2005). libvirt possui uma API que permite o gerenciamento de máquinas virtuais independentemente do MMV subjacente. Atualmente, a biblioteca já possui suporte aos sistemas Xen, QEMU, KVM e OpenVZ.

Suporte a outras soluções estão atualmente em desenvolvimento.

2.3.1 Utilização

libvirt utiliza uma definição XML dos sistemas hospedados: nela são gravadas as características da máquina virtual, como discos, memória, processadores, interfaces de rede, etc.

À partir desta definição, chamadas de função padronizadas podem ser aplicadas à mesma. A figura 2.1 apresenta um exemplo de utilização da biblioteca. Note que a linguagem de programação no caso exposto é a linguagem Python. A mesma possui ligações que mapeiam suas chamadas de função à libvirt, que é implementada em C. Ligações para outras linguagens também estão disponíveis, como Perl e OCaml. Ligações para Java e Ruby também encontram-se nos planos de desenvolvimento.

```
import libvirt
import sys

conn = libvirt.openReadOnly(None)
if conn == None:
    print 'Failed to open connection to the hypervisor'
    sys.exit(1)

try:
    dom0 = conn.lookupByName("Domain-0")
except:
    print 'Failed to find the main domain'
    sys.exit(1)

print "Domain 0: id %d running %s" % (dom0.ID(), dom0.OSType())
print dom0.info()
```

Figura 2.1: Exemplo de código Python utilizando a libvirt

A figura 2.1 demonstra uma chamada ao hipervisor para recuperar informações à respeito de uma máquina virtual. Num primeiro momento, é feita a conexão com o MMV através da chamada *libvirt.openReadOnly()*. Após, é feita uma chamada para localizar uma determinada máquina virtual, no caso *Domain-0*. Por fim, são utilizados métodos de acesso para recuperar informações acerca da máquina virtual em questão.

2.3.2 libvirtd

De modo a garantir acesso remoto para os hipervisores, é incluído junto à biblioteca libvirt o programa libvirtd. libvirtd executa em segundo plano e provê um serviço que permite a conexão de aplicações remotas ao MMV local. A figura 2.2 ilustra a visão de alto nível da aplicação, onde um hospedeiro com diversas máquinas virtuais hospedadas é acessado remotamente por um cliente utilizando o serviço de conexão remota.

Hospedeiro + hipervisor + libvirt

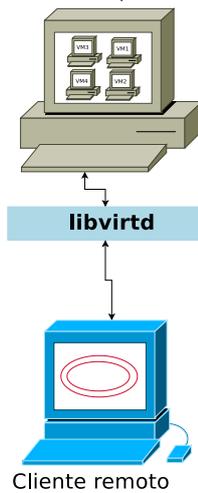


Figura 2.2: Acesso remoto através do serviço libvirtd

Dessa forma, libvirtd garante conexões remotas utilizando métodos de autenticação e criptografia, inclusive para hipervisores que não possuem serviços de conexão externa.

2.4 Virtual Machine Manager – virt-manager

Virtual Machine Manager (virt-manager), é uma aplicação gráfica que serve para a administração de ambientes virtualizados. Seu desenvolvimento teve início no Grupo de Tecnologias Emergentes da empresa RedHat Inc., responsável pela distribuição RedHat Enterprise Linux. Sua interface principal mostra ao usuário um sumário das máquinas virtuais executando sobre um determinado hipervisor, bem como estatísticas gráficas em tempo real acerca do desempenho e da utilização de recursos com o passar do tempo. A figura 2.3 ilustra a tela principal da aplicação.

2.4.1 Características Principais

virt-manager possui algumas características desejáveis para ferramentas de gerenciamento de ambientes virtualizados, a saber:

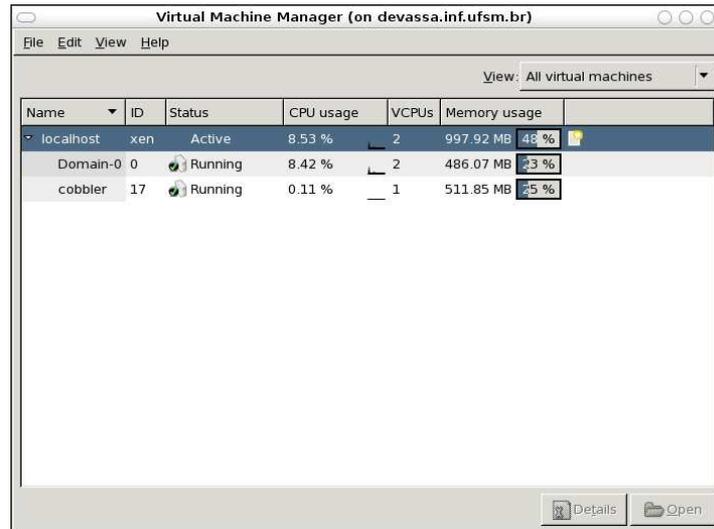


Figura 2.3: virt-manager: tela principal

- Acesso remoto a MMV's: além do gerenciamento de recursos virtualizados localmente, virt-manager pode comunicar-se com hipervisores remotos, de modo a centralizar a administração em uma só estação.
- Suporte a múltiplos hipervisores: virt-manager utiliza a biblioteca libvirt como fundação. Assim o mesmo pode ser utilizado para administrar diversas soluções de virtualização, de modo unificado e consistente.
- Assistente para criação de novas máquinas virtuais: virt-manager possui uma aplicação assistente para facilitar a criação de novas máquinas virtuais.
- Gerenciador de redes virtuais: as redes virtuais utilizadas pelos MMV's para a conexão das máquinas virtuais ao mundo externo podem ser gerenciadas de modo gráfico. Além do gerenciamento de redes existentes, virt-manager possui um assistente para a criação de novas redes virtuais.
- Gerenciador de dispositivos virtuais: todos os dispositivos utilizados pelas máquinas virtuais podem ser manipulados em um console gráfico. Algumas características da máquina virtual podem ser alteradas sem necessidade de reinicialização da mesma. Além do gerenciamento de dispositivos, há um assistente para a adição de novos dispositivos virtuais.

2.4.2 Limitações

Pelo fato de ser uma ferramenta relativamente nova, virt-manager possui suporte limitado a distribuições que não sejam baseadas na RedHat, principalmente no que diz respeito ao assistente de instalação de máquinas virtuais que utilizem a técnica de para-virtualização. Porém, o suporte a outras distribuições está previsto na base de código atual da ferramenta e depende apenas do empenho dos desenvolvedores que fazem parte destas outras comunidades ou instituições.

Além disso, ainda existem características desejáveis em uma ferramenta de gerenciamento de ambientes virtualizados que não foram implementadas em virt-manager, como o suporte à migração de máquinas virtuais.

2.5 Cobbler

Cobbler é um conjunto de serviços unificados com o intuito de prover uma forma automatizada de instalar e reinstalar sistemas operacionais em ambientes conectados. Sua utilização se dá através da configuração de um servidor de inicialização (*boot server*) e os nós da rede que necessitam de (re-)instalação inicializam-se tendo o servidor Cobbler como base. Assim os mesmos têm acesso à perfis armazenados no servidor e podem ser automaticamente instalados no seu respectivo destino, de forma simples e rápida.

Ao servidor Cobbler são importados espelhos (*mirrors*) de distribuições Linux. Com base nesses espelhos, são criados perfis de instalação para os clientes.

Os perfis armazenados no servidor Cobbler podem ser genéricos ou exclusivos para um único cliente. Estes perfis podem ser especializados a partir de perfis mais genéricos, através de um sistema de herança.

Basicamente, estes perfis são arquivos de configuração para o instalador Anaconda (Red Hat, Inc and others, 2007a). Anaconda é um instalador que trabalha tanto em modo interativo, com em modo não-interativo. Para executar o instalador em modo não-interativo, é passado como parâmetro na inicialização do instalador um arquivo especial de configuração. A esse arquivo de configuração é denominado *kickstart* (Red Hat, Inc and others, 2007b). Esse arquivo contém todas as perguntas que um instalador faz ao usuário, eliminando assim a necessidade de interação, o que permite a instalação automática do sistema.

Além de um kickstart específico para o sistema a ser instalado, o perfil armazena

informações sobre o ambiente destino, como discos, memória, processadores, etc.

Diversas funcionalidades do sistema Cobbler podem ser acessadas remotamente através de uma biblioteca de chamadas XML-RPC. Esta biblioteca permite o acesso a informações sobre os repositórios, espelhos e perfis disponíveis no servidor.

Além da recuperação de informações, existem chamadas a procedimentos que permitem a criação de novos perfis, bem como a importação de novos espelhos.

2.6 Interrelacionamento entre as Soluções

De modo a clarificar o relacionamento entre os diversos atores que tomam parte no ambiente descrito pelas seções anteriores, é demonstrada na figura 2.4 uma visão geral de como todos os sistemas interagem entre si. Assim é possível de se observar a disposição em camadas dos sistemas, começando pelo nível mais inferior, onde os diversos hipervisores suportados pela biblioteca libvirt estão localizados. Diretamente acima da biblioteca libvirt, todos os outros sistemas são apresentados com suas interdependências.

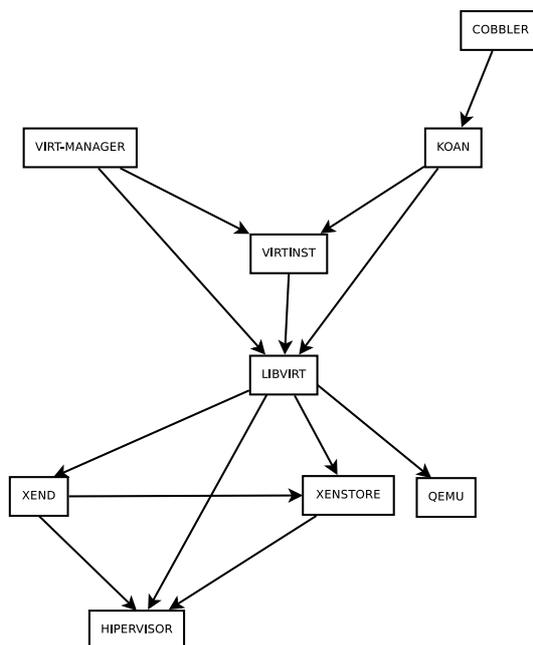


Figura 2.4: Visão geral dos sistemas. Fonte: (Red Hat, Inc, 2007)

Em nosso trabalho, será feita uma nova ligação na figura 2.4, de modo que virt-manager possa comunicar-se diretamente com a ferramenta Koan. Dessa forma, torna-se possível estabelecer uma via de comunicação entre a mesma e o sistema Cobbler.

3 PROJETO E DESENVOLVIMENTO

Neste capítulo apresenta-se o projeto e desenvolvimento da integração entre as ferramentas virt-manager e Cobbler. Inicialmente apresenta-se um ambiente típico de execução de tais ferramentas, que serve de base para a definição de requisitos e funcionalidades desejadas. A seguir relata-se aspectos importantes da implementação da solução proposta.

3.1 Requisitos e Funcionalidades Desejadas

Em ambientes que utilizam algum tipo de solução de virtualização, é muito comum a utilização de alguma ferramenta para facilitar a gerência das máquinas virtuais. Uma ferramenta bastante utilizada para tal fim é a ferramenta virt-manager.

Como a tarefa de criação de novas máquinas virtuais pode se tornar dispendiosa em termos de tempo, o sistema Cobbler pode ser utilizado para fornecer um serviço de criação automática de máquinas virtuais. Assim, temos o ambiente virtualizado que é ilustrado na figura 3.1.

Dessa forma, seria interessante para o administrador do ambiente se existisse um modo de se criar novas máquinas virtuais de forma automática utilizando a ferramenta virt-manager, provendo assim uma maneira integrada de gerenciar o ambiente.

De modo que a ferramenta modificada possa facilitar a criação de novas máquinas virtuais, estabeleceu-se os seguintes requisitos:

- **Integração transparente com o sistema Cobbler:** para um administrador que utilizará uma ferramenta gráfica como virt-manager, é necessário que toda integração entre esta e o sistema Cobbler seja automática, evitando a todo custo a interação do usuário com aspectos que não tenham relação direta com as tarefas de administração do ambiente virtualizado. Assim, todas as rotinas de descoberta, comunicação e possíveis erros que advenham de tais execuções devem ser tratadas de forma au-

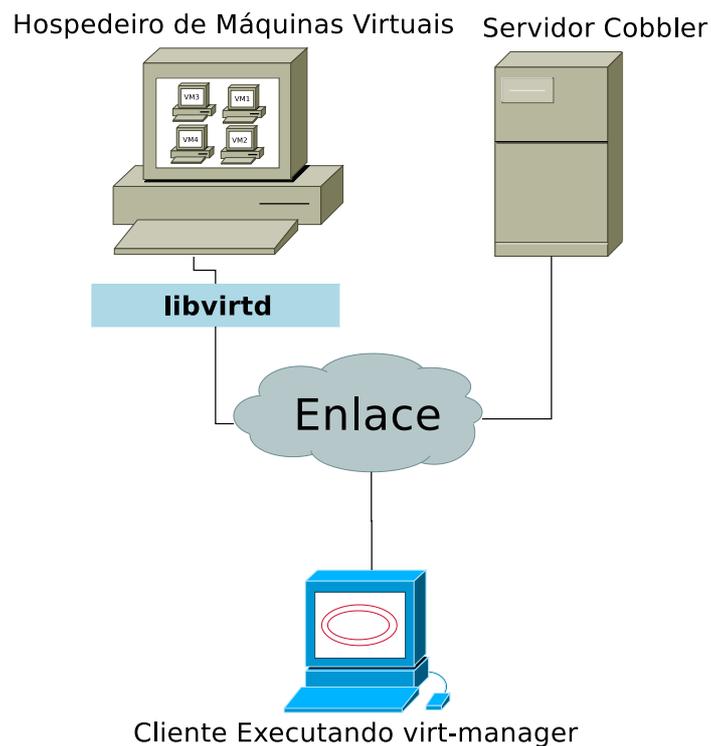


Figura 3.1: Ambiente virtualizado

tomática pela própria ferramenta. Por exemplo, no caso em que nenhum servidor Cobbler é descoberto, não é perguntado ao usuário para o mesmo informar manualmente um servidor Cobbler válido; a única diferença será o desativamento de tal opção durante a execução do assistente de criação de máquinas virtuais.

- **Redução do número de passos a serem executados:** uma vez que a seleção de um perfil é feita na terceira tela do assistente, o usuário da ferramenta é dispensado das demais etapas, pois todas as informações necessárias para a instalação serão supridas pelo perfil provido pelo sistema Cobbler. No caso do usuário optar por não utilizar nenhum perfil, a instalação segue os trâmites normais.

Assim, após seleção (ou não) de um perfil e a conseqüente conclusão do assistente, o usuário é devolvido à tela principal da ferramenta, onde o mesmo poderá se conectar à recém criada máquina virtual.

3.2 Implementação

Antes de entrar em detalhes sobre a implementação da solução propriamente dita, faz-se necessário uma breve explanação na seção 3.2.1 para contextualizar onde serão

feitas modificações na ferramenta. A seguir, nas seções 3.2.2, 3.2.3 e 3.2.4 apresenta-se o método de integração final.

3.2.1 Assistente de Criação de Máquinas Virtuais

A ferramenta virt-manager traz consigo uma aplicação para assistir o usuário na criação de novas máquinas virtuais. Este assistente consiste em um conjunto de passos com perguntas que são feitas ao usuário da aplicação de forma a criar um perfil com dados necessários à criação de uma nova máquina virtual.

Após o processo de coleta de respostas, o assistente dispara uma rotina que faz a criação de uma nova máquina virtual e a inicialização de um instalador que foi definido pelo usuário na etapa anterior. A partir desse momento, o assistente é concluído e o usuário é levado novamente à tela principal do sistema, onde poderá conectar-se à nova máquina virtual criada e executar manualmente o processo de instalação do sistema operacional.

A rotina disparada pelo assistente é uma ferramenta em linha de comando, conhecida como *virt-install*. Após a coleta de todos os dados necessários à instalação, o assistente repassa as informações em questão para a ferramenta virt-install, que faz a criação da nova máquina virtual propriamente dita.

Em ambientes paravirtualizados, é oferecida ao usuário uma tela onde o mesmo escolhe o servidor remoto que possui os arquivos de inicialização, bem como o restante do instalador. Há ainda a opção de se utilizar um kickstart, sendo este também localizado em um servidor remoto. A figura 3.2 apresenta a tela em questão.



Figura 3.2: Tela de seleção da fonte de instalação e kickstarts

3.2.2 Encontrando um Servidor Cobbler

Como o servidor Cobbler é também um servidor de inicialização, novas máquinas que forem conectadas ao ambiente podem simplesmente conectar-se ao servidor Cobbler na sua inicialização para ter acesso ao seu perfil de instalação. Dessa forma, as etapas seguintes da instalação do sistema podem ser executadas sem a intervenção do usuário.

Contudo, MMV's que implementam a técnica de paravirtualização, como é o caso de Xen, não possuem uma rotina de inicialização remota. Isso se dá pelo fato da carga do sistema operacional ser iniciada no domínio administrativo do sistema, sendo que só após a carga do núcleo do sistema o controle passa a ser da nova máquina virtual. Dessa forma, é necessário a utilização de uma ferramenta auxiliar para comunicar-se com o servidor Cobbler, encontrar um perfil adequado e a partir disso inicializar a nova máquina virtual e por fim executar o instalador do resto do sistema operacional.

Um exemplo dessa abordagem é a ferramenta Koan, que permite a criação de novas máquinas virtuais à partir de perfis armazenados no servidor Cobbler.

3.2.3 Koan e a Arte de Descobrir Servidores

Aliado ao sistema Cobbler está a ferramenta Koan (*Kickstart Over a Network*). Esta ferramenta permite a instalação de sistemas onde o servidor de inicialização não se encontra disponível, ou em sistemas virtualizados.

Assim, Koan pode ser utilizado para a instalação de novas máquinas virtuais, com base nos perfis armazenados em um servidor Cobbler. Para fazer a instalação de novas máquinas virtuais, Koan utiliza como ferramenta de suporte o programa virt-install (BER-RANGE et al., 2007). Virt-install utiliza como base a biblioteca libvirt. Assim, o mesmo tem o poder de criar máquinas virtuais independente do MMV utilizado no sistema em questão.

Outra característica interessante de Koan é a descoberta automática de servidores Cobbler localizados na mesma sub-rede. Essa descoberta se dá pelo fato do servidor Cobbler utilizar o sistema Avahi (Avahi, 2007) para publicar o seu serviço de instalação remota. O sistema Avahi é uma implementação livre do sistema Zeroconf (STEINBERG; CHESHIRE, 2006). Avahi implementa uma série de técnicas para descoberta automática de serviços e autoconfiguração de sistemas.

A descoberta de um servidor Cobbler se dá pela varredura da base de serviços Avahi

publicada na rede local. O passo seguinte é estabelecer uma conexão ao servidor para a recuperação da lista de perfis disponíveis.

Um exemplo dessa descoberta de servidor implementada pela ferramenta Koan é ilustrada na figura 3.3.

```
$ koan --server=DISCOVER --list-profiles
- running: /usr/bin/avahi-browse --all --terminate --resolve
- analyzing zeroconf scan results
- connecting to: 200.18.42.94
- connected to: 200.18.42.94
FC-6-i386
FC-6-xen-i386
centos-5-i386
centos-5-xen-i386
```

Figura 3.3: Descoberta de um servidor Cobbler e listagem de perfis

Com base na implementação da função de descoberta de servidores Cobbler pela ferramenta Koan, foi feita uma implementação do método *avahi_search()* no assistente de instalação de virt-manager.

Este método basicamente utiliza a ferramenta avahi-browse (Avahi, 2007) para inquirir o sistema Avahi se existe um ou mais servidores Cobbler na rede local onde se encontra o assistente de instalação. No presente momento, este método depende da existência de avahi-browse para executar com sucesso. Futuramente, este método será reimplementado utilizando a API do sistema Avahi para localizar os servidores.

Ao final da execução do método, é retornado para o método chamador uma lista com os endereços dos servidores encontrados. Caso nenhum servidor tenha sido encontrado, retorna-se uma lista vazia.

3.2.4 Notificando o Usuário da Presença de um Servidor Cobbler

De modo a estender o poder da ferramenta gráfica de instalação de máquinas virtuais, foi feita uma modificação na interface do assistente. Assim, se um servidor Cobbler é encontrado, o usuário é notificado através da disponibilização de uma lista de servidores no assistente.

Além da notificação visual da presença de servidores Cobbler na rede, é disponibilizada ao usuário uma lista com os perfis que um determinado servidor armazena. Assim, é possível fazer a seleção de um servidor e um perfil. Como os perfis armazenados no

servidor Cobbler possuem todas as informações necessárias para a criação de uma nova máquina, o fato de selecionar um perfil no assistente desobriga o usuário de seguir adiante no processo de instalação. Para notificar o usuário de tal mudança no comportamento da aplicação, o botão de avanço do assistente (*Forward*) é escondido e, no seu lugar, o botão de conclusão do assistente (*Finish*) é exibido. A figura 3.4 ilustra esta situação.



Figura 3.4: Acessando um servidor Cobbler

Assim, foi possível uma integração entre as duas ferramentas de modo elegante do ponto de vista do usuário da ferramenta. A modificação da interface gráfica foi extremamente pontual, apenas uma tela do assistente necessitou ser modificada, o que pode reduzir a curva de aprendizagem para a utilização da nova característica.

4 AVALIAÇÃO

Para avaliar a viabilidade e utilidade da ferramenta em questão foi elaborada uma análise qualitativa da mesma. Assim, executou-se diversos testes de modo a avaliar a facilidade de utilização e se a ferramenta atinge os objetivos propostos. Uma análise quantitativa torna-se muito complicada de se executar, uma vez que determinados fatores, como o tempo de interação do administrador com o sistema de instalação, varie de forma que se torne impossível uma medição apurada.

Nas seções seguintes serão discutidas formas de se utilizar o sistema, bem como serão descritos alguns testes que avaliam a utilização do mesmo em um ambiente distribuído típico.

4.1 Utilização da Solução

O ambiente de testes foi formado por um servidor Pentium III, com 2 GB de memória e espaço suficiente de disco para a criação de novas máquinas virtuais. O hipervisor utilizado foi o Xen 3.0.3, possuindo instalado em seu domínio administrativo a distribuição CentOS Linux versão 5. Neste domínio também foram instaladas a ferramenta virt-manager, junto com a biblioteca libvirt e a ferramenta Koan.

Primeiramente, é necessário a instalação e configuração de um servidor Cobbler no ambiente de testes. Assim, criou-se uma nova máquina virtual, na qual foi instalado o sistema Cobbler e posteriormente foi importado para sua base uma distribuição Linux. A distribuição no caso foi a CentOS Linux versão 5. A instalação e importação da distribuição foram feitas utilizando como guias as páginas do manual do sistema. Após a instalação e configuração do sistema Cobbler, uma lista de perfis já encontrava-se disponível. Assim, nesse ponto já era possível a criação de uma nova máquina virtual de uso genérico.

4.1.1 Criando Novos Perfis

A criação de novos perfis no sistema Cobbler pode ser feito de diversas formas. Todas as formas se dão por variação de parâmetros que são passados pela ferramenta de linha de comando `cobbler`. A forma mais flexível é a criação de um novo perfil através da especificação de todos os parâmetros e fornecendo um kickstart para o novo perfil. Outra forma mais simples se dá através do mecanismo de herança da ferramenta, onde somente alguns atributos de algum perfil pré-existente são modificados, de modo a facilitar a criação de perfis que possuem diversas características em comum.

4.1.2 Configuração do Servidor DHCP

De modo a garantir que as novas máquinas virtuais possam ser configuradas de modo automatizado, faz-se necessário a utilização de um sistema de configuração automática de rede. Um sistema de configuração muito popular atualmente é o *Dynamic Host Configuration Protocol* (DHCP) (DROMS et al., 1997). Assim, na máquina virtual que executa o servidor Cobbler instalou-se também o servidor DHCP. Cobbler pode ser configurado para gerenciar o servidor DHCP, se este existir na mesma máquina. Assim, torna-se dispensável por parte do administrador gerenciar mais esse serviço, uma vez que ele pode ser feito pelo próprio Cobbler.

4.1.3 Ativando Todos os Atores

Assim, com os serviços de DHCP e distribuição de perfis instalados, a criação de novas máquinas virtuais através do assistente de virt-manager já pode ser utilizada. Para tal é executada a ferramenta virt-manager no domínio administrativo do ambiente de testes.

A partir deste momento, já é possível a criação de novas máquinas virtuais de modo automático utilizando a ferramenta adaptada desenvolvida.

4.2 Fazenda de Servidores Web

Um caso típico de utilização da solução desenvolvida é a criação de um conjunto de servidores Web. Nessa situação, um servidor provendo acesso a um conjunto de dados é dividido em diversas máquinas virtuais, de modo a dividir o acesso concorrente aos dados através de vários servidores, permitindo assim que as máquinas virtuais que servirão os dados possam ser migradas para outro hipervisor, garantindo o acesso aos dados de forma

mais confiável.

Além dos servidores Web, é necessário a configuração de um serviço de nomes, que possa designar uma conexão de um cliente aos diversos servidores Web, permitindo assim a ilusão de que todos os clientes estão acessando o mesmo servidor. A figura 4.1 ilustra a configuração proposta.

O serviço de nomes não foi configurado em uma máquina virtual. Para tal tarefa, um servidor de nomes que é utilizado pelo ambiente de testes foi reconfigurado para refletir o esquema acima descrito.

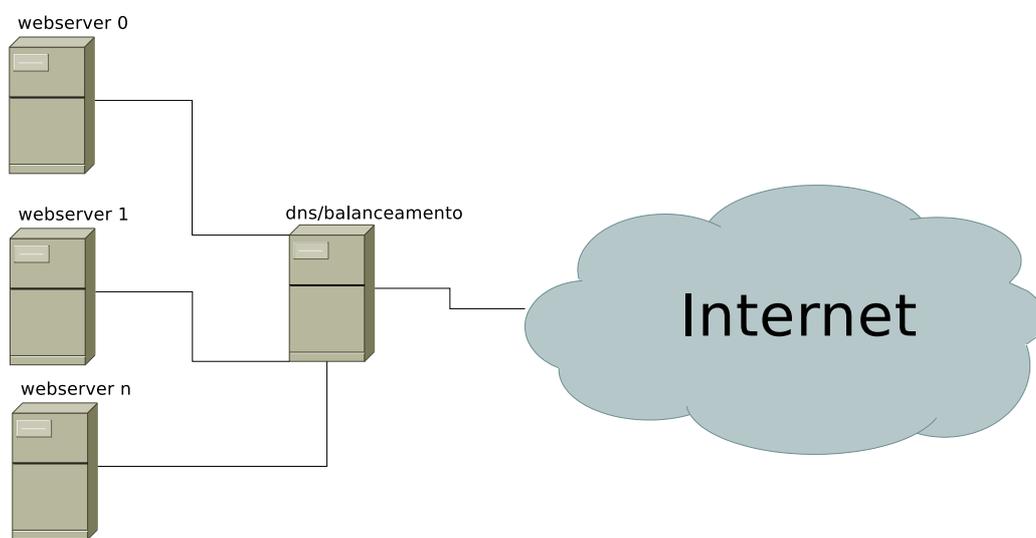


Figura 4.1: Ambiente de testes

4.2.1 Criação dos Servidores

Para a criação dos servidores Web, foi necessário a criação de um novo perfil no sistema Cobbler. Para criar tal perfil, foi utilizado um kickstart que garantisse a instalação do servidor Web na nova máquina virtual. Devido à grande semelhança entre o perfil padrão que é criado na hora da importação de uma nova distribuição ao sistema, optou-se por criar um perfil derivado deste. Assim, o perfil "webservice-xen" foi adicionado ao sistema. A figura 4.2 ilustra o comando utilizado.

```
$ cobbler profile add --profile=centos-5-xen-i386 \
  --name=webservice-xen \
  --kickstart=/var/lib/cobbler/ks/webservice.ks
$ cobbler sync
```

Figura 4.2: Criação de um novo perfil no sistema Cobbler

Após a adição do novo perfil, executou-se o assistente de criação de máquinas virtuais modificado da ferramenta virt-manager, que imediatamente inclui o novo perfil em sua lista de perfis disponíveis. Na figura 4.3 é apresentada a tela do assistente com o perfil "webserver-xen" selecionado.



Figura 4.3: Selecionando um perfil específico

4.2.2 Problemas Encontrados

Apesar da utilização da ferramenta ter transcorrido sem erros, surgiram alguns problemas não previstos inicialmente.

Durante a criação de uma nova máquina virtual, o processo de atribuição de nome para a mesma é feito automaticamente: cada máquina criada possui como nome uma concatenação da data em que a mesma é criada, conforme pode ser observado na figura 4.4. Para renomear a nova máquina, é necessário a utilização de ferramentas de linha de comando, o que pode ser bastante complicado.

Name	ID	Status	CPU usage	VCPUs	Memory usage
localhost	xen	Active	17.63 %	2	1.97 GB 98 %
Domain-0	0	Running	15.98 %	2	997.05 MB 48 %
Thu_Jan_24_17_03_17_2008_11	11	Running	1.50 %	1	511.78 MB 25 %
cobbler	5	Running	0.14 %	1	511.80 MB 25 %

Figura 4.4: Padrão de nomes criados por virt-manager

Outro problema é a falta de liberdade oferecida pelo sistema Cobbler, que só permite a manipulação de perfis baseados em distribuições RedHat. Seria mais interessante a

integração com um sistema que fosse mais flexível, mas no presente momento tal solução parece ainda não existir.

Uma outra questão não resolvida é a criação de novas máquinas virtuais em hipervisores remotos. Apesar de existir suporte à conexão remota por parte da biblioteca libvirt, no presente momento o assistente de criação de máquinas virtuais ainda não possui implementado um método para criação remota de novas máquinas. Assim, em cada servidor que possua um hipervisor instalado, é necessário possuir também a ferramenta virt-manager instalada localmente.

5 CONCLUSÃO

Neste trabalho, foi proposto e implementado um sistema que habilita um administrador de ambientes virtualizados a instalar novas máquinas virtuais de modo automatizado. Para atingir tal objetivo, foi projetado e desenvolvido um sistema que integra a ferramenta de administração de ambientes virtualizados virt-manager e o sistema de instalação de novas máquinas Cobbler. virt-manager é uma solução de administração vantajosa por ser construída sobre a biblioteca libvirt, que possui uma API de comunicação consistente para a manipulação de diversos monitores de máquinas virtuais. Cobbler é um sistema que permite a instalação automática de novas máquinas, através da seleção de perfis disponibilizados pelo mesmo. Junto ao sistema Cobbler, a ferramenta Koan é utilizada a comunicar-se com o servidor Cobbler e iniciar a instalação de um novo sistema. Koan também possui suporte à criação de máquinas virtuais, fazendo uso da ferramenta virt-install. Outra importante característica de Koan é a descoberta automática de servidores Cobbler na rede local, utilizando o sistema de descoberta de serviços Avahi.

A implementação se deu pela modificação do assistente de criação de máquinas virtuais provido pela ferramenta virt-manager. A ela foi adicionado a opção de se utilizar um perfil disponibilizado pelo sistema Cobbler, permitindo assim a criação automática de uma nova máquina virtual.

Para avaliar a eficácia da ferramenta, propôs-se a criação de uma fazenda de servidores Web virtualizados. Tal ambiente é típico em ambientes distribuídos e sua implantação é semelhante a diversas outras situações enfrentadas por um administrador de sistemas. A implantação desse ambiente comprovou a facilidade e a agilidade do sistema proposto.

Como trabalho futuro, seria desejável a criação de novos perfis no sistema Cobbler através dos dados colhidos durante uma instalação normal de uma nova máquina virtual, de modo que fosse possível replicar uma instalação à partir da criação de um modelo bá-

sico. Assim, a tarefa de criação de novos ambientes virtualizados seria consideravelmente facilitada.

REFERÊNCIAS

ARMSTRONG, B. **Professional Microsoft Virtual Server 2005**. [S.l.]: Wrox Press Ltd. Birmingham, UK, UK, 2007.

ARNOLD, K.; GOSLING, J. **The Java programming language**. [S.l.]: ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1998.

Avahi. Disponível em: <http://avahi.org>. Acesso em: agosto de 2007.

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEBAUER, R.; PRATT, I.; WARFIELD, A. Xen and the art of virtualization. In: ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES (SOSP '03), 19., 2003, Bolton Landing, USA. **Proceedings...** ACM, 2003. p.164–177.

BEGNUM, K. Managing large networks of virtual machines. **Proceedings of the 20th conference on Large Installation System Administration Conference-Volume 20 table of contents**, [S.l.], p.16–16, 2006.

BELLARD, F. QEMU, a Fast and Portable Dynamic Translator. **Proceedings of the USE-NIX Annual Technical Conference, FREENIX Track**, [S.l.], p.41–46, 2005.

BERRANGE, D.; BROCK, H. O.; KATZ, J.; DUFFY, M. et al. **Virtual Machine Manager: home**. Disponível em: <http://virt-manager.org/>. Acesso em: agosto de 2007.

BORNTRAGER, C.; SCHWIDEFSKY, M. Providing Linux 2.6 support for the zSeries platform. **IBM Systems Journal**, [S.l.], v.44, n.2, p.331–340, 2005.

BURGESS, M.; RALSTON, R. Distributed resource administration using cfengine. **Software Practice and Experience**, [S.l.], v.27, n.9, p.1083–1101, 1997.

CLARK, B.; DESHANE, T.; DOW, E.; EVANCHIK, S.; FINLAYSON, M.; HERNE, J.; MATTHEWS, J. N. Xen and the Art of Repeated Research. In: USENIX ANNUAL TECHNICAL CONFERENCE, FREENIX TRACK, 2004. **Anais...** USENIX, 2004. p.135–144. Disponível em: <http://www.usenix.org/publications/library/proceedings/usenix04/tech/freenix/clark.html>.

CLARK, C.; FRASER, K.; HAND, S.; HANSEN, J. G.; JUL, E.; LIMPACH, C.; PRATT, I.; WARFIELD, A. Live Migration of Virtual Machines. In: SYMPOSIUM ON NETWORKED SYSTEMS DESIGN AND IMPLEMENTATION (NSDI '05), 2., 2005, Boston, USA. **Proceedings...** Usenix, 2005.

DEHAAN, M. **Cobbler: provisioning made simple**. Disponível em: <http://cobbler.et.redhat.com/>. Acesso em: agosto de 2007.

DOSBox. **DOSBox, a x86 emulator with DOS**. Disponível em: <http://dosbox.sourceforge.net/>. Acesso em: outubro de 2007.

DROMS, R. et al. **Dynamic Host Configuration Protocol**. [S.l.]: RFC 2131, March 1997, 1997.

Enomaly Inc. **Enomalism - Virtualized Management Console**. Disponível em: <http://www.enomalism.com>.

KEMP, S. **Xen Tools**. Disponível em: <http://xen-tools.org/software/xen-tools/>.

Kernel Based Virtual Machine. Disponível em: <http://kvm.qumranet.com/kvmwiki>. Acesso em: novembro de 2007.

MATTHEWS, J.; HU, W.; HAPUARACHCHI, M.; DESHANE, T.; DIMATOS, D.; HAMILTON, G.; MCCABE, M.; OWENS, J. Quantifying the performance isolation properties of virtualization systems. **Proceedings of the 2007 workshop on Experimental computer science**, [S.l.], 2007.

Parallels, Inc. **Parallels Workstation**. Disponível em: <http://www.parallels.com/>. Acesso em: novembro de 2007.

Red Hat, Inc. **RedHat Emerging Technology Group**. Disponível em: <http://et.redhat.com/>. Acesso em: novembro de 2007.

Red Hat, Inc and others. **Anaconda**. Disponível em: <http://rhlinux.redhat.com/anaconda/>. Acesso em: novembro de 2007.

Red Hat, Inc and others. **The Official Red Hat Linux Customization Guide**. Disponível em: <http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/custom-guide/part-install-info.html>. Acesso em: novembro de 2007.

SMITH, J. E.; NAIR, R. The Architecture of Virtual Machines. **IEEE Computer**, [S.l.], v.38, n.5, p.32–38, 2005. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/MC.2005.173>.

STEINBERG, D.; CHESHIRE, S. **Zero Configuration Networking: the definitive guide**. [S.l.]: O'Reilly, 2006.

SUGERMAN, J.; VENKITACHALAM, G.; LIM, B.-H. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. In: USENIX ANNUAL TECHNICAL CONFERENCE, 2001., 2001. **Proceedings...** Usenix Assoc., 2001. p.1–14.

SWsoft. **Server Virtualization Open Source Project**. Disponível em: <http://openvz.org/>. Acesso em: agosto de 2007.

TURK, D.; BAUSCH, J. Virtual Linux servers under z/VM: security, performance, and administration issues. **IBM Systems Journal**, [S.l.], v.44, n.2, p.341–351, 2005.

VEILLARD, D.; ZAK, K.; BERRANGE, D.; MCLOUGHLIN, M.; JONES, R. W. **the virtualization API**. Disponível em: <http://libvirt.org/>. Acesso em: agosto de 2007.

VMWare, Inc. **VMware: virtualization, virtual machine & virtual server consolidation - vmware**. <http://www.vmware.com/>.

Xen Source Inc. **Xen Enterprise Data Sheet V11022006, version 3.1**. [S.l.]: Xen Source Inc., 2006. (02-11-2006).