

Migração de servidores físicos para virtuais – P2V usando ferramentas OpenSource

Diego Lima Santos

Curso de Especialização em Redes e Segurança de Sistemas

Pontifícia Universidade Católica do Paraná

Curitiba, outubro de 2009

Resumo

Estudo de caso real abordando a migração (P2V) de servidores físicos Linux para máquinas virtuais, realizada no Serviço Social Autônomo Paranaidade. Utilizando para tal procedimento ferramentas OpenSource disponíveis, virtualização dos servidores utilizando o monitor de máquinas virtuais Xen. O artigo descreverá as, tecnologias utilizadas, procedimentos adotados e resultados obtidos após a migração.

1 Introdução

Em meados de 2008 o Serviço Social Autônomo Paranaidade adquiriu servidores do modelo *IBM server x3650* de alta performance, porém a migração trivial de todos os antigos servidores para novos não era possível, pois a quantidade de antigos servidores era superior. Entretanto é notável que a ociosidade dos recursos em servidores de alta performance é em média de 90%, devemos maximizar a utilização dos recursos computacionais, pois o custo unitário de cada servidor é relativamente alto.

A técnica de virtualização neste cenário é indicada para resolver o problema da subutilização dos novos recursos. Virtualização não é uma idéia nova, desde a criação da linguagem de programação *Java*, a maioria dos usuários de computadores utiliza pelo menos uma máquina virtual, talvez antigamente utilizavam o sistema *UCSD Pascal p*, uma das primeiras máquinas virtuais para *Pascal*. A virtualização na camada de sistema operacional data do final da década de 1960, a primeira máquina virtual foi o *VM/CMS* da *IBM*, essa tecnologia ainda é usada atualmente com o nome de *z/VM*, utilizado para suportar o uso eficiente do Linux nos servidores *IBM zSeries*. [1,2]

Atualmente o mercado de soluções de virtualização é disputado por diversas tecnologias e produtos, podemos então distinguir entre virtualização de servidor, memória e rede, porém essas formas frequentemente se confundem. De acordo com a finalidade da aplicação, a virtualização de servidores e de desktops se separam, porém esses limites de separação são imprecisos. Outro critério é saber se a virtualização será efetuada através de hardware ou de software. Se a virtualização for vinculada a uma determinada plataforma ou for independente de plataforma, podemos falar em virtualização *in-the-box* e *out-of-the box*. [1,2]

1.1 Técnicas de Virtualização

As máquinas virtuais de sistema podem ser distinguidas de acordo com a localização da camada de virtualização. Se a camada de virtualização estiver diretamente sobre a camada de hardware é chamada de Tipo 1 ou Nativo, portanto um sistema operacional convidado roda em um nível superior a hardware, exemplos: *CP/CMS*, *Xen*, *VMWare ESX*. Caso a camada de virtualização estiver por cima de um sistema operacional, é chamado de Tipo 2 ou *Modell Hosted Virtualization*, exemplos: *VMWare-GSX*, *Qemu*, *MS Virtual PC*, *OpenVz*, *Vserver* fazem parte desta categoria de máquinas virtuais.

Existem dois tipos básicos de virtualização de máquinas, virtualização completa e paravirtualização: Caso toda a plataforma física (CPU, memória, I/O) for virtualizada, estamos falando de virtualização completa (*full-virtualization*).

Outro tipo de virtualização requer alterações nos hóspedes, assim os mesmos sabem perfeitamente que rodam em um ambiente virtual. Essas alterações são perfeitamente possíveis de serem executadas em sistemas de código aberto. Essa técnica é conhecida como paravirtualização (*paravirtualization*), permite que máquinas virtuais específicas comuniquem-se diretamente com o hardware, ao invés de todas máquinas hóspedes comunicarem-se com o hospedeiro. A para-virtualização só é possível por meio da cooperação de hóspede e hospedeiro na manipulação de operações sensíveis, um benefício de desempenho considerável. O *Xen* e o sistema representante mais popular da para-virtualização.[1,2] A figura 1 ilustra os 3 tipos básicos de virtualização: Virtualização total, Para-virtualização e Modell Hosted Virtualization.

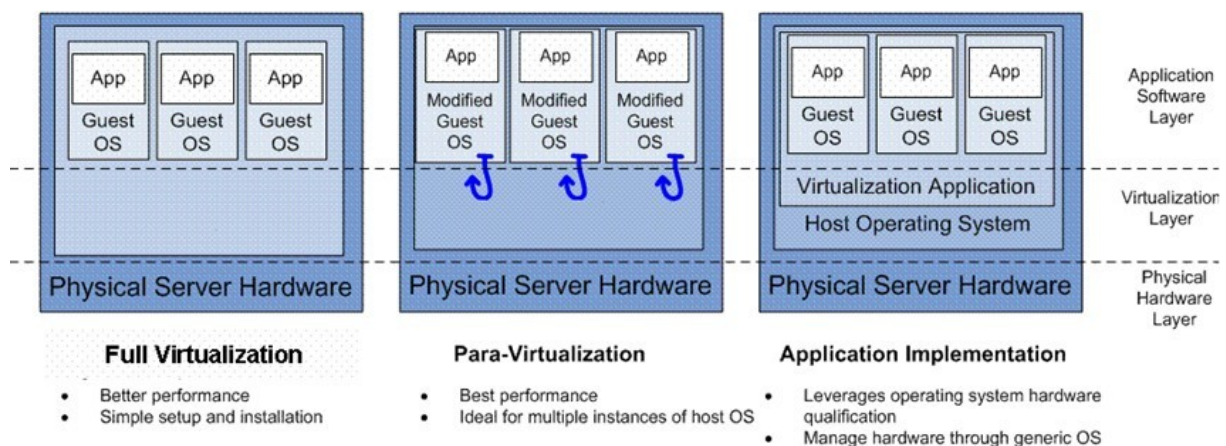


Figura 1 – Tipos de Virtualização.[3]

Nenhuma técnica de execução é necessariamente a melhor, embora a virtualização completa custe desempenho, abre a possibilidade de poder executar software de um mundo estranho. A paravirtualização tem maior performance que máquinas completamente virtualizados e alta performance em I/O, mas requer para isso sistemas operacionais hóspedes modificáveis, praticamente eliminando o sistema operacional MS Windows.

2 Hypervisor Xen

O Xen foi originalmente desenvolvido pelo Grupo de Pesquisas de Sistemas no Laboratório de Computação da Universidade de Cambridge, como parte do projeto *XenoServers* fundado pelo *UK-EPsrc*. Sua primeira aparição foi no *SOSP (Symposium on Operating System)* em 2003 e a primeira versão (1.0) foi liberada em outubro do mesmo ano. [2] Foi fundada uma empresa denominada *XenSource, Inc.*, com objetivo de gerir e promover

o desenvolvimento do projeto open source. Em outubro de 2007 a *XenSource, Inc* foi adquirida pela *Citrix System* por U\$500 milhões.[5] Atualmente participam do desenvolvimento além de membros da comunidade Xen, engenheiros pertencentes a grandes tais como: *Citrix, SGI, Sun, IBM, HP, Intel, Dell, Oracle, Novell*. O *Xen* é licenciado com a licença *GPL2* [6]. Existe a versão paga do *Xen*, desenvolvido pela *Citrix* chamada de *XenServer*.

Podemos dizer que *hypervisor Xen* é o coração do *Xen*, ele situa-se entre os domínios convidados e a plataforma física (hardware), alocando e controlando recursos, provendo proteção e isolamento. Ele define as interfaces virtuais que serão utilizadas pelos domínios convidados. Cada máquina virtual que roda no *Xen* é chamada de *domain*. Quando o servidor *Xen* inicia, ele primeiro inicia o *hypervisor*, que é responsável em iniciar o domínio chamado *Domain0* (*dom0*) que é o sistema operacional hospedeiro, o *dom0* é o domínio privilegiado, tem acesso aos recursos de hardware e contém ferramentas para gerenciar os outros domínios, domínios sem privilégios são chamados de *domainU* (*domU*), de *unprivileged*. [7,8]

A figura 2 ilustra o funcionamento do *hypervisor* do *Xen*, o *hypervisor* virtualiza a CPU e a memória. Enquanto o domínio privilegiado *dom0*, pode acessar diretamente o hardware, os domínios *domU*, sem privilégios, acessam aos dispositivos virtualizados pelo *hypervisor*. [1]

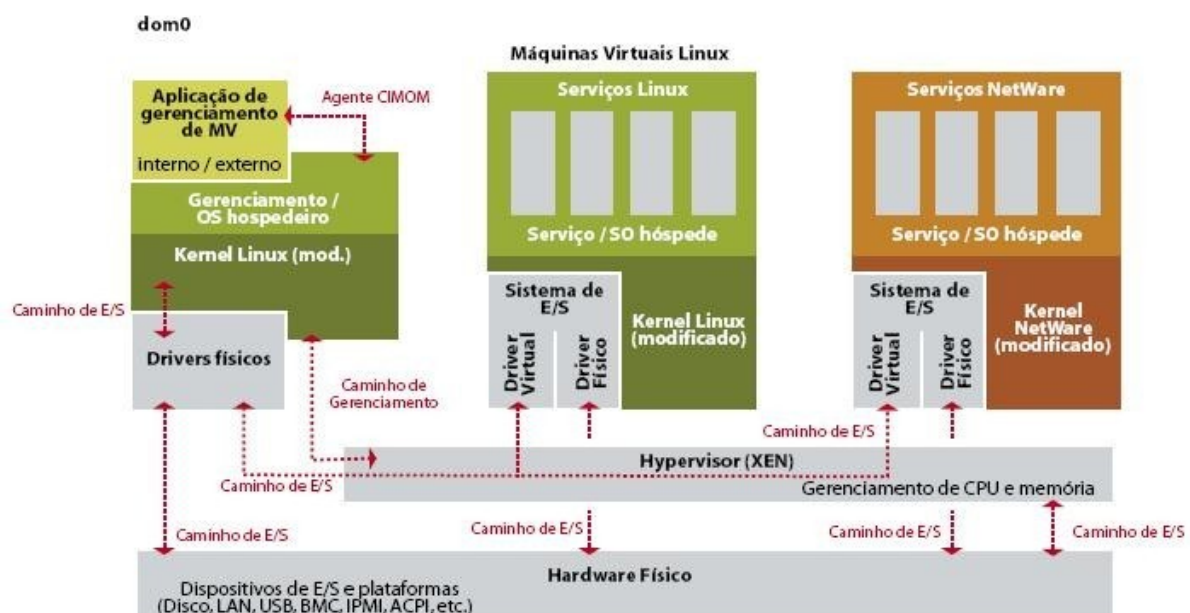


Figura 2 – *Hypervisor Xen* [1]

O *Xen* suporta dois tipos básicos de virtualização: para-virtualizado (*PVM*) e virtualizado total (*HVM*), para executar hóspedes virtualizados total conjuntamente com para-virtualizados será necessário o processador possuir em sua arquitetura tecnologia *AMD-V* (Codename *Pacífica*) ou *Intel-VT* (Codename *Vanderpool*). Máquinas para-virtualizadas apresentam melhor desempenho do que máquinas em virtualização total.[1,2]

2.1 Vantagens do Xen

Podemos relacionar como benefícios da virtualização utilizando o *Xen* como monitor de máquina virtual (*VMM*) [2,4,8]:

- **Consolidação de servidores:** Colocar vários servidores dentro de uma única máquina, reduzindo custos, necessidades de espaço físico e facilitando a administração;

- Aumentar a utilização dos servidores: Diminuir a ociosidade na utilização de recursos computacionais em servidores de grande porte;
- Contenção de Falhas: isolamento de serviços dedicados em máquinas virtuais diferentes;
- Manter nível de serviço (SLA): Recurso de *Live migration* na realocação de domínios Xen entre plataformas físicas, para evitar a paralisação e manter acordos de nível serviço;
- Menor TCO: O *Xen open-source* além do seu reconhecido desempenho e baixo custo para data center ou sistemas empresariais;
- Economia de Energia: Economizar energia é facilmente notável, exemplo: alimentar e refrigerar 100 servidores físicos é um custo bem mais alto se analisar 1 servidor físico com 100 servidores virtuais;
- Redução de domínios de colisão e domínios de *broadcast*: O tráfego entre as máquinas virtuais é realizado dentro do *hypervisor*;
- Clonagem: Facilmente adicionar servidores por clonagem de um servidor virtual já existente;
- Computação em *cluster*: Permite a implantação de servidores em *cluster*;
- Redução de custo de implantação de novos servidores.

Podemos relacionar como desvantagens do *Xen*: A obrigatoriedade de utilizar um *kernel* modificado e um micro kernel para estabelecer a máquina real e as máquinas virtuais; A quantidade de passos para configurar e instalar uma máquina virtual.

2.2 Terminologias Xen

A infraestrutura de virtualização *Xen* funciona de maneira diferente de outras soluções de virtualização, algumas definições e termos são necessárias quando para o entendimento do sistema *Xen* de virtualização [8]:

- *VM*: Máquina virtual é o ambiente virtual que executa um sistema operacional;
- *VMM*: Monitor de máquina virtual, software que possibilita executar máquinas virtuais;
- *Domain*: Termo utilizado para referir a uma instância da máquina virtual;
- *Dom0*: Domínio privilegiado no sistema *Xen*;
- *DomU*: Todos os outros domínios no sistema *Xen*;
- *Host*: Sistema hospedeiro de máquinas virtuais;
- *Guest*: Instância de máquina virtual que é iniciada no *host*;
- *PVM*: Máquina virtual para-virtualizada, *domains Xen PVM* somente pode ser iniciada utilizando sistemas operacionais modificados;
- *HVM*: Máquina virtual totalmente virtualizada, *domains Xen* pode iniciar sistemas operacionais sem modificação. O *Xen* oferece a capacidade de virtualizar *Microsoft Windows* utilizando esse recurso.

2.3 Tipos de Xen hypervisor

O *Xen* hypervisor está disponibilizado em diferentes opções, para diferentes propósitos [2]:

- 32 bit SMP: Pacote para arquitetura 32 bit com suporte a múltiplos núcleos ou sistemas com mais de um processador, espaço de endereçamento de memória limitada a 4GB, suporta somente domínios de 32 bits.
- 32 bit PAE36 SMP: Pacote para arquitetura 32 bit com suporte a processador de

múltiplos núcleos ou sistemas com mais de um processador, a extensão PAE permite endereçar até 64GB de memória, suporta somente domínios de 32 bits.

- 64 bit SMP: Pacote para arquitetura 64 bit com suporte a processadores de múltiplos núcleos ou sistemas com mais de um processador, elimina todas as restrições de endereçamento de memória associadas ao endereçamento de 32 bits, suporta domínios de 32 bits e 64 bits.

O *source code* do *Xen hypervisor* está disponibilizado em: www.xen.org/download/, os pacotes precisam ser compilados, contém o kernel alterado para funcionar no *Xen*, ferramenta utilizadas no *dom0* para controlar o *hypervisor* e documentação do *Xen*.

Além da opção de instalar a partir do *source code* é possível instalar o *Xen* pré compilado. Atualmente principais distribuições Linux (Debian, OpenSuse, Red Hat, Fedora, CentOS) contém os pacotes necessários para implantação do *Xen Hypervisor* por meio de gerenciadores de pacotes (aptitude, apt-get, yum, etc...).

2.4 Armazenamento dos domínios

O *Xen* possibilita a opção de usar uma variedade de métodos de armazenamento para os domínios convidados, armazenamento local utilizando: arquivos de imagens, volumes lógicos (LVM) e partições comuns; Armazenamento remoto, como *iSCSI*, *ATA-over-Ethernet* (*AoE*), e *Network File System* (*NFS*). [7]

Arquivos simples podem ser utilizados como dispositivos de bloco virtual, armazenando domínios *Xen*, está é a maneira mais rápida de começar a usar o *Xen*. Utilizando o comando *dd* podemos criar o arquivo de armazenamento a ser utilizado. Embora os arquivos ser uma maneira simples e rápida de armazenar Vms, existem algumas desvantagens da utilização de arquivos como *Virtual Block Devices* VBDs [8]:

- Não são adequados para domínios virtuais que utilizam E/S intensivamente, o acesso ao dispositivo virtual é extremamente lento quando o trabalho de E/S é pesado.
- O Linux, por padrão, suporta no máximo de oito arquivos VBDs utilizado por domínios *Xen*. Este número pode ser aumentado utilizando o parâmetro de kernel *max_loop*.

O LVM fornece um nível de abstração acima dos dispositivos de bloco real (exemplo, discos). Isto permite uma flexibilidade significativa na gestão de armazenamento, entre outras coisas, tornando mais fácil redimensionar partições em tempo real. O termo LVM geralmente refere-se a execução específica de uma gerenciador de volume lógico para o kernel do Linux.

O uso de LVM para armazenamento de domínios *Xen* é provavelmente a maneira ideal para configurar e administrar os domínios. Em testes realizados o desempenho utilizando LVM como Dispositivo de Bloco Virtual VBD é próximo da utilização de dispositivos físicos diretos do tipo */dev/hdX*. [4]

O interessante recurso *Live Migration* do *Xen*, permite migrar uma máquina virtual de um servidor para outro, entretanto é necessário que o tipo de armazenamento do dispositivo de bloco virtual (*VBD*) seja remoto, sistemas de arquivos de rede ou dispositivos de armazenamento de rede (*NAS*, ou *Network Attached Storage*). O *NFS* pode ser utilizado como sistema de arquivos de rede. Uma solução de baixo custo e excelente desempenho é utilizar *iSCSI* [1].

3 Migração P2V – físico para virtual.

O processo de alcançar a consolidação de servidor virtualizado depende do ponto de partida, caso esteja projetando seu sistema desde o início, a solução é implementar a virtualização a partir do desenvolvimento. Caso o sistema já está fisicamente instalado, a

solução é migrar do ambiente físico para virtual [9].

O termo migração *P2V* vem do inglês, *Physical to Virtual*, é o procedimento para conversão e migração de sistema operacional, aplicativos e dados instalados em um servidor físico, para um máquina virtual convidada de uma plataforma de virtualização. Existem diversos métodos para executar a migração *P2V*: manual, semi-automático e automático [9]:

- Método *P2V* automático: Método de migração que utiliza ferramentas que migra o servidor físico pela rede, sem a assistência do usuário. Exemplos de ferramentas: *PlatSpin*, *AutoVirt*, *VizionCore*, *LeoStream*.
- Método *P2V* semi-automático: Método de migração que utiliza ferramentas, que necessita da assistência do usuário para migrar o servidor físico. Exemplos de ferramentas: *VMWare Converter*, *MS Virtual Server Migration Toolkit*, *Virt-P2V*.
- Método *P2V* manual: Método *P2V* onde o usuário manualmente cria a máquina virtual no ambiente de virtualização e copia todos os arquivos do sistema operacional, aplicativos e dados da máquina física.

Algumas Ferramentas de migração também podem solucionar diversos problemas em potencial trazidos pela incompatibilidade de hardware entre o servidor físico e a máquina virtual, passando os drivers necessários ao kernel do sistema operacional e inicializando os drivers corretamente durante a fase de inicialização do sistema. A maioria dessas ferramentas *P2V* são proprietárias e fazem mais do que é preciso para migração de uma única máquina [9].

Especificamente para o *XenServer* a *Citrix* desenvolveu a ferramenta proprietária *XenConvert*, a ferramenta permite a migração *P2V* semi-automática, permite migração *P2V* de qualquer sistema operacional suportado pelo *XenServer* (*Windows*, *Linux*, etc).

A ferramenta livre, porém ainda experimental, *Virt-P2V* [12] é promissora, funciona como um *live CD* que realiza a migração *P2V* via rede, incluindo a instalação da VM na máquina hospedeira, o requisito para migração é a plataforma de virtualização suportar a API de virtualização *Libvirt* entre as plataformas que suportam estão *Xen*, *KVM*, *Qemu* [13].

Existe diversos métodos de se fazer migração *P2V* manualmente, um desse métodos utiliza ferramentas *open source* de clonagem tais como *Mondo Rescue*, *Partimage*, *Clonezilla Livre*, com algumas configurações podemos virtualizar em modo paravirtualizado ou totalmente virtualizado conforme for o caso, esse método suporta a migração *P2V* de servidores *MS Windows* e *Linux* [9,14].

O método mais interessante para migrar servidores físicos Linux para virtuais, utiliza das ferramentas e comandos do próprio Linux para efetuar a migração *P2V* manual. Devido a grande variedade de comandos e ferramentas disponíveis no Linux existem muitos métodos descritos, tais como:

- Utilizando o comando *netcat - nc* conjuntamente com o utilitário *dd* [15,16] e efetuar a clonagem via rede.
- Utilizando o *ssh* com o comando *tar* [17].
- Utilizando a ferramenta de cópia *rsync* [18,19,20].

Em todos esses métodos descritos caso o sistema migrado for Linux, é necessário vários ajustes para que o sistema operacional possa se acomodar bem em máquina virtual paravirtualizada ou totalmente virtualizada.

4 Projeto de migração

O Serviço Social Autônomo PARANACIDADE adquiriu novos servidores do modelo *IBM server x3650*, havia a necessidade de colocar em produção os novos servidores visto que estações de trabalho e servidores legados faziam o papel de servidores em produção. No parque de servidores da organização a política de software é preferencial para utilização de

software livre, todos os servidores utilizam como sistema operacional *Debian Linux*. Para a migração dos servidores legados existia duas possibilidades: migrar para máquinas físicas ou migrar para máquinas virtuais.

Migrar para máquinas físicas o sistema operacional ocupa toda a plataforma física e não utiliza com eficiência todos os recursos computacionais do servidor (E/S, memória, CPU), cada servidor físico legado necessita de um novo servidor para permitir a migração, fazendo necessário a aquisição de um maior número de servidores, gerando custos exagerados.

A outra solução possível, migrar para máquinas virtuais através de alguma plataforma de virtualização, possibilitaria melhor utilização dos recursos dos servidores, reduzindo a subutilização, compartilhando e alocando os recursos (memória, CPU, disco) de forma customizada, oferecendo também segurança e performance.

Rapidamente vemos que a migração utilizando máquinas virtuais é a melhor solução, sendo possível “multiplicar” a quantidade de servidores dedicados por plataforma física.

Dentre as soluções de virtualização disponíveis a plataforma escolhida foi o Xen, escolha foi pelo Xen ser de código aberto – GPL 2, estar disponível em praticamente todas as atuais distribuições Linux (*Debian, Fedora, CentOS, Red Hat, OpenSuse...etc*), reconhecido alto desempenho em máquinas virtuais do tipo paravirtualizadas.

4.1 Planejamento da migração

O estudo de caso em questão contempla o projeto de migração realizado no Paranacidade, o caso apresentado será migração P2V de 3 servidores físicos para 3 máquinas virtuais. O servidor de nome *uru* é onde está instalado o *hypervisor Xen*, este servidor é da marca/modelo *IBM System x3650*.

As características principais de hardware do servidor *uru*: 2 processadores *Intel Xeon E5320* de 4 núcleos cada um, 8 GB de memória principal e 2 discos SAS de 300GB cada, suporte a RAID. O *IBM System x3650* possui os recursos da tecnologia de virtualização *Intel-VT*, necessários para o *Xen* virtualização total (*full-virtualization*).

A situação inicial dos servidores físicos em produção antes da migração P2V era utilização de servidores legados e estações de trabalhos, servidores migrados estão descritos abaixo:

- Servidor *beijaflor*: Genérico, processador *Pentium III* 1 GHz, disco IDE de 4GB, 256 MB de memória. *Debian Etch*, serviços DNS *master* do domínio e NTP.
- Servidor *tucano*: Genérico, processador *Pentium IV* 1,8 GHz, disco IDE 40GB, 1 GB de memória. *Debian Etch*, serviço WEB (Apache e Postgresql) de projetos.
- Servidor *colibri*: *Compaq Proliant ML370 server*, processador *Pentium III - Copermine* 1 Ghz, disco SCSI de 40 GB, 512 MB de memória. *Debian Etch*, serviço WEB (Apache e Mysql), portal ambiente *XOOPS*. Aproximadamente 80.000 horas em produção.

A condição dos servidores migrados não era a ideal para servidores em produção, porém apenas o servidor *colibri* apresentava problemas de hardware, blocos defeituosos no disco rígido ocorrendo erros de leitura e escrita.

Migrar os servidores físicos para máquinas virtuais foi a melhor solução encontrada, pois utilizar um novo servidor para cada máquina migrada subutilizaria recursos computacionais, com a virtualização há diversos benefícios, consolidação de servidores, melhor aproveitamento de recursos, economia de energia, redução de custos operacionais, dentre outros já citados.

4.2 Instalando o Xen

Existe diversas maneiras de instalar o Xen em servidores, o primeiro passo é definir qual distribuição Linux instalar para ser a base do hospedeiro, em nosso caso o data-center é padronizado com a utilização do sistema operacional Debian Linux.

O servidor *uru* tem 2 discos de 300GB optou-se por utilizar os discos em RAID nível 1, aumentando a segurança dos dados, o particionamento foi definido da seguinte forma:

- / - Sistema de arquivos raiz. - 15GB;
- Swap – Área de troca - 1 GB;
- LVM – Logical Volume Manager - 280 GB, partição LVM onde são criados os discos das máquinas virtuais.

A versão instalada do sistema operacional no servidor *uru*, foi a versão *stable* disponível, *Debian Lenny* – arquitetura *i386*, em servidores Xen é aconselhado que não sejam instalados serviços extras desnecessários (*WEB*, *DHCP*, *SAMBA*, etc), podendo comprometer o desempenho. No momento da instalação foi selecionado somente o conjunto de pacotes “Sistema Básico”.

Alguns pacotes são necessários após a instalação básica, através do gerenciador de pacotes *aptitude*, podemos instalar facilmente dependências e os pacotes.

- Com o comando `# aptitude install rsync ssh postfix ntp`, instalamos pacotes úteis.
- *ssh*: servidor e cliente de ssh;
- *Rsync*: aplicativo que sincroniza arquivos e diretórios remotos;
- *postfix*: Servidor de e-mail.
- *NTP*: Serviço de sincronização de relógio entre servidores, utilizado para manter a hora correta no servidor.

Para fins de monitoramento básico do servidor *uru*, instalar os pacotes *logwatch*, *fcheck* e *cron-apt*. Estes aplicativos podem enviar e-mail para uma determinada conta habilitando previamente o MTA *postfix*:

- `# aptitude install logwatch fcheck cron-apt`

Para instalar o *Xen System* em servidores *Debian Linux* existe duas possibilidades relatadas [8, 21]; Compilar e instalar os fonte obtidos do site do projeto Xen <http://www.xen.org/downloads>. Outra possibilidade é instalar através do formato de pacotes da distribuição, no caso do *Debian* formato *.deb*, instalar usando pacotes da *distro* apresenta inúmeras vantagens, são pacotes homologados e altamente testados pela comunidade e desenvolvedores, podem ser gerenciados facilmente através do *aptitude*, *apt-get*, etc , não precisam do trabalhoso e demorado processo de compilação dos fontes, porém frequentemente os fontes obtidos do site do projeto Xen são versões mais recentes.

Pela facilidade e comodidade no gerenciamento de pacotes utilizar os pacotes da distribuição *Linux* é a melhor solução em casos normais.

No *Debian Lenny* podemos instalar o *Xen System* e todas suas dependências com apenas um comando:

- `# aptitude install xen-linux-system-2.6.26-2-xen-686`.

Este metapacote instala o *Xen hypervisor* de 32 bits *PAE*. Para instalar o *Xen hypervisor* 64 bits, pode-se instalar o sistema operacional Debian arquitetura *amd64* ou obter o pacote *xen-hypervisor-amd64* e instalar manualmente:

- `dpkg -x xen-hypervisor-3.2.1-amd64_3.2.1-2_amd64.deb xen64`
- `cp xen64/boot/xen-3.2.1-amd64.gz /boot`

Utilizando o *aptitude* não é possível ter um *hypervisor* 64 e *dom0* de 32 bits ou tudo é 64 bits ou 32 bits, Desta maneira manual temos o *hypervisor* 64 bits e a *dom0* de 32 bits, esse arranjo é utilizado no Xen Live CD [22]. Após é necessário configurar o gerenciador de boot

grub para utilizar o *microkernel Xen* de 64 bits e o *kernel* 32 bits na *dom0*, é conveniente também limitar a memória da *dom0* (parâmetro *dom0_mem*) técnica conhecida como *memory balloon*. Exemplo de entrada no menu *grub*:

```
title Xen Hypervisor 3.2-1 64 bits / Debian GNU/Linux, kernel 2.6.26-2-xen-686
kernel /xen-3.2-1-amd64.gz dom0_mem=512M
module /vmlinuz-2.6.26-2-xen-686 root=/dev/mapper/vg01--dom0--root ro
console=tty0
module /initrd.img-2.6.26-2-xen-686
quiet
```

O arquivo de configuração do *Xen* fica localizado em */etc/xen/xend-config.sxp*, existe diversos parâmetros para torna a virtualização possível, segue abaixo alguns parâmetros básicos configurados[7]:

- *network-script* : Determina o script que será usado para criar o ambiente de rede;
- *vif-script*: Especifica a interface virtual, o script é localizado em */etc/xen/script*;
- *mim-mem*: Define o tamanho mínimo de memória reservado para a *Dom0*, padrão 196MB;
- *xend-unix-server*: Habilita o gerenciamento do servidor *Xen* via *Unix Domain Socket Server*;
- *xend-http-server*: Habilita o gerenciamento do servidor *Xen* via *http stream packet*;
- *xend-port*: Define a porta utilizada para gerenciar o servidor via *http*, padrão porta 8000;
- *xend-relocation-server*: Habilita o recurso de *live migration*;
- *xend-relocation-port*: Defina a porta utilizada para o *live migration*, padrão porta 8002.

4.2.1 Gerenciando os domínios Xen

Para gerenciar as máquinas virtuais convidadas do servidor *uru*, utilizamos a ferramenta linha de comando *xm* – *Xen Management User Interface*, essa ferramenta é instalada e acessada na *domain0*, *xm* se comunica diretamente com o *daemon* do *Xen* (*xend*), através do comando *xm* podemos criar, ativar, desligar e alterar domínios virtuais.

Os principais subcomandos para gerenciar domínios virtuais através do *xm* são [7]:

- *create*: Cria um domínio virtual baseado num arquivo de configuração;
- *destroy*: Encerra um domínio imediatamente;
- *list*: Lista e provê informações dos domínios;
- *reboot*: Reinicia o domínio;
- *shutdown*: desliga o domínio.

Para o gerenciamento remoto das máquinas virtuais sem precisar estar na console do servidor *uru*, utilizamos a biblioteca desenvolvida *C* pela *Red Hat*, *Libvirt*, para instalar a *libvirt* em servidores *Debian*, utilizamos o seguinte comando: *#aptitude install libvirt-bin*, através da biblioteca *libvirt* podemos acessar remotamente o servidor *uru*, utilizando a ferramenta linha de comando *virsh* ou utilizando a ferramenta gráfica *virt-manager*; nesse caso via túnel *ssh* seguro (figura 3).

Existe diversas outras ferramentas de gerenciamento de domínios *Xen* e algumas também são gráficas, porém foram testadas, exemplos de outras ferramentas são: *HyperVM* (Baseado *WEB*), *Ganeti*, *MLN*, *Enomalism*, *ConVirt*.



Figura 3 – Ferramenta *virt-manager* acessando servidor URU

4.2.2 Segurança no domínio privilegiado - Domain0

Quando administramos uma máquina física, umas das tarefas mais importantes é proteger o sistema, esta tarefa pode assumir diversas formas, incluindo a aplicação dos *patches* de segurança mais recentes, fechar todas as portas desnecessárias, auditar qualquer comportamento anormal ou suspeito. Essas atividades são importantes também quando se utiliza máquinas virtuais *Xen*. Além disso, o *Xen* introduz novas oportunidades e desafios em segurança, o *Xen* introduz uma camada abaixo das máquinas virtuais, qualquer comprometimento do próprio *Xen* ou do domínio privilegiado *Domain0* pode causar problemas em todas as máquinas virtuais. É fundamental procurar isolar serviços em máquinas virtuais diferentes, podemos também criar segmentos de redes distintos [7].

Se o domínio privilegiado *dom0* estiver comprometido, todos os domínios *domUs* também estão, por esse motivo é vital que o sistema operacional *dom0* esteja bem protegido contra ataques. Segundo [7] remover softwares e serviços desnecessários é a maneira mais simples de proteger um *domain0*, pois qualquer serviço em execução representa um ponto de entrada para um possível ataque, é importante limitar acesso de usuários na *dom0*, somente permitir o login do usuário administrador, seguimos estas premissas na instalação e configuração do servidor *uru*.

A estrutura de rede no *Xen* é flexível, é possível monitorar toda atividade de rede das máquinas virtuais com facilidade. Podemos utilizar um monitor de tráfego de rede como o *Snort* e bloquear todos os pacotes de entrada para o *domain0* utilizando um firewall como o *iptables netfilter* para *domain0s* baseados em Linux ou *ipf* para baseados em UNIX [7].

Optamos por utilizar regras de *iptables* na *dom0* do servidor *uru*. Um script *shell* é executado ao final do processo de *boot* do servidor. Apenas permitimos o acesso ao serviço *ssh* na *dom0* necessário para o gerenciamento remoto do servidor, um projeto futuro podemos implementar regras para os domínios convidados *domUs*. Em linhas gerais o script habilita as seguintes regras:

- `#iptables -a input -s 0.0.0.0 -d uru.paranacidade.org.br -p tcp --dport 22 -j accept;`
- `#iptables -a input -d uru.paranacidade.org.br -j reject.`

4.3 Migrando P2V

Existem diversas ferramentas e procedimentos para migração *P2V*, algumas ferramentas são proprietárias e há relatos que funcionam muito bem, porém nosso objetivo é sempre que possível utilizar de ferramentas *Open-Source*, das ferramentas testadas a

ferramenta *Virt-P2V* é a mais promissora, entretanto a ferramenta ainda é experimental e esta em desenvolvimento. De todos os métodos pesquisados e considerando o cenário: servidores físicos *Debian Linux*, máquina virtual do tipo paravirtualizada e ambiente de virtualização *Xen* utilizando como base *Debian Linux*, o método que consideramos mais adequado descrevemos abaixo.

O método P2V é manual e utiliza o aplicativo *rsync* [18,19,20] para efetuar a cópia dos dados do servidor físico para o disco da máquina virtual, alguns ajustes nos arquivos de configuração precisam ser feitos para máquina física acomodar bem na máquina virtual, utilizaremos como exemplo a migração P2V do servidor físico *beijaflor*.

Primeiramente criamos os volumes lógicos (LVM) que servirão de discos virtual. O esquema de particionamento pode variar ou ser semelhante ao do servidor físico, criamos 2 volumes lógicos, 1GB para partição de *swap* e 5GB para partição *root* com sistema de arquivos *ext3*:

- `#lvcreate -L1G -n beijaflor.swap vg01`
- `#lvcreate -L5G -n beijaflor.root vg01`
- `#mkfs.ext3 /dev/vg01/beijaflor.root`
- `#mkswap /dev/vg01/beijaflor.swap`

Cada máquina virtual *Xen* utiliza um arquivo de configuração, este arquivo deve ficar localizado na *dom0* em */etc/xen/auto/*, criamos o arquivo *beijaflor-vm.cfg*:

```
#Arquivo de configuração Xen – máquina virtual beijaflor-vm
import commands
krn_vers = commands.getoutput('uname -r')
name      = 'beijaflor-vm'
builder   = 'linux' #builder PVM se a VM for HVM builder = 'HVM'
#vfb e extra configura console, habilita acesso VNC
vfb       = [ 'type=vnc,vncdisplay=1' ]
extra     = 'console=hvc0 clocksource=jiffies guestname=beijaflor-vm'
kernel    = '/boot/vmlinuz-' + krn_vers #Define o kernel utilizado pela VM
ramdisk    = '/boot/initrd.img-' + krn_vers
memory    = '512' # Quantidade de memória em megabytes
vcpus     = 1 # Número de CPUs
# Define os discos virtuais e a partição root
root       = '/dev/xvda1 ro'
disk      = [ 'phy:/dev/vg01/beijaflor.root,xvda1,w', \
              'phy:/dev/vg01/beijaflor.swap,xvda2,w' ]
# Configura o ambiente de rede
vif        = [ 'bridge=xenbr0' ]
```

A próxima etapa é copiar todos os arquivos e diretórios do servidor físico *beijaflor*, para que não haja inconsistência dos dados copiados é recomendado parar os serviços do servidor migrado, especialmente os serviços que utilizem bases de dados como *openldap*, *mysql*, *pgsql*, etc, deixando somente os serviços básicos para a cópia remota.

No caso do servidor *beijaflor* paramos os processos *ntpd* - NTP e *bind9* - DNS. Alguns diretórios não precisam ser copiados, em especial os */sys* e */proc*. Na console da *dom0* executamos o seguinte comando:

- `#rsync -vaH -e 'ssh' --numeric-ids --stats --progress --exclude "/mnt/*" --exclude "/proc/*" --exclude "/sys/*" --exclude "/tmp/*" --exclude "/var/tmp/*" --exclude "/var/run/*.pid" --exclude "/var/run/dbus/system_bus_socket" beijaflor.paranacidade.org.br:/ /mnt/beijaflor.root/`

É preciso ajustar os pontos de montagens da máquina virtual no arquivo `/mnt/beijaflor.root/etc/fstab` e incluir a console padrão `hvc0` nos arquivos `/etc/inittab` e `/etc/securetty`.

Utilizando esse método não é preciso instalar gerenciador de *boot* (*grub*, *lilo*, etc) no disco root da máquina virtual, porém é possível utilizar o script *pygrub* para tal finalidade. Utilizamos a chamada do kernel Linux da *dom0* fora da máquina virtual, porém é necessário copiar os módulos do kernel da *dom0* para dentro da máquina virtual.

- `#cp -a /lib/modules/$(uname -r) /mnt/beijaflor.root/lib/modules/`

Verificamos e revisamos todas as configurações antes de desligar o servidor físico, feito isso concluímos a migração P2V inicializando a máquina virtual criada, utilizando o aplicativo *xm*:

- `#xm create -c /etc/xen/auto/beijaflor-vm.cfg`

4.3.1 Domínios virtuais - DomUs

A tabela abaixo mostra os *domUs* criados no servidor *uru* utilizando o procedimento P2V descrito. Na criação das máquinas virtuais somente os recursos mínimos necessários foram alocados.

DomUs	S.O.	Serviços	Disco	RAM	Processador
<i>Beijaflor-vm</i>	<i>Debian Lenny</i>	<i>DNS, Ntp</i>	<i>LVM – 8GB</i>	<i>512MB</i>	<i>1</i>
<i>Colibri-vm</i>	<i>Debian Etch</i>	<i>Web</i>	<i>LVM - 10G</i>	<i>2048MB</i>	<i>2</i>
<i>Tucano-vm</i>	<i>Debian Lenny</i>	<i>Web, Pgsq1</i>	<i>LVM - 15G</i>	<i>1024MB</i>	<i>2</i>

4.4 Problemas encontrados

Diagnosticamos alguns problemas em migração P2V, após a migração as máquinas virtuais – *domUs* congelavam, se a máquina hospedeira fosse reiniciada “a quente”, o kernel gerava a log de erro com a seguinte expressão: *'clocksource/0: Time went backwards'*, esse problema foi resolvido adicionado no arquivo de configuração da máquina virtual a *flag extra= clocksource=jiffies* [23].

Outro problema constatado após a migração P2V, foi a inacessibilidade ao serviço SSH das máquinas virtuais, pesquisando sobre o problema, constatamos que o pacote *udev* não estava instalado nas máquinas virtuais [24].

O procedimento que utilizamos não pode ser realizado em sistemas operacionais muito antigos, visto que não existe kernel Linux modificado *Xen* para Linux antigos (ex: *Debian Potato*, *Red Hat 6.2*, etc) impossibilitando de utilizar a paravirtualização, somente sendo possível a migração para máquinas *full*-virtualizadas.

5 Conclusão

As migrações P2V realizada no data-center do Serviço Social Autônomo Paranaidade foram bem sucedidas, os já citados benefícios da virtualização de servidores utilizando o *Xen* foram alcançados, consideramos que a consolidação de servidores é de vital importância para organização, pois maximiza a utilização dos recursos computacionais existentes e conseqüentemente diminui custos.

Softwares de virtualização 100% livre integrados a sistemas operacionais livres, como é o caso do *Xen* em relação às distribuições Linux, tornam o futuro pertencentes a

virtualização [2].

Constatamos que o desempenho das máquinas virtuais é superior aos antigos servidores físicos, até o momento as máquinas virtuais em produção não apresentaram sinais de falta de recurso (memória, processador), caso ocorra podemos alocar os recursos necessários utilizando as interfaces de gerenciamento (CLI ou GUI).

Como projetos futuros podemos: implementar regras de *firewall* na *dom0* para cada *domU*, está sendo estudada a aquisição de um *Storage Area Network* - *SAN*, possibilitando assim utilizar do recurso *Live Migration*, implantar Controle de Acesso Obrigatório (*MAC - Mandatory Access Control*) das permissões de acesso de cada máquina virtual, utilizando a biblioteca *Svirt* [25], ainda experimental.

Bibliografia

- [1] GARLOFF, KURT. “Para-virtualização com o Xen 3”. Linux MAGAZINE. v..24, p. 38-43, outubro de 2006.
- [2] SIQUEIRA, LUCIANO; BRENDDEL, JENS-CHRISTOPH. *Virtualização*. São Paulo, Linux Pocket PRO. ed. 04. 2007.
- [3] HARA, FÁBIO. *O que é virtualização ?*. Acessado em outubro de 2009. Disponível em: <http://superdownloads.uol.com.br/materias/que-virtualizacao-parte-1.html>
- [4] IBRAHIM, HORÁCIO. *Virtualização de Servidores com Xen*. Brasília: SERPRO, 2008.
- [5] *Xen Citrix Acquisition*. Acessado em outubro de 2009. Disponível em: <http://www.xensource.com/about/Pages/CitrixAcquisition.aspx>
- [6] *Xen Hypervisor*. Acessado em outubro de 2009. Disponível em: <http://www.xen.org/products/xenhyp.html>
- [7] MATTEWS, JEANNA; DOW, ELIS; DESHANE, TODD, HU WENJIN; ET AL. *Running Xen: A Hands-On Guide to the Art of Virtualization*. Editora Prentice Hall. Abril de 2008.
- [8] WILLIAMS, DAVID. *Virtualization with Xen*. Editora Syngress. 2008.
- [9] LI, TING. *Migrando para um ambiente Linux Virtual com Clonezilla*. Acessado em outubro de 2009. Disponível em: <http://www.ibm.com/developerworks/br/library/l-clonezilla/>
- [10] Wikipédia. Acessado em outubro de 2009. Disponível em: <http://en.wikipedia.org/wiki/Physical-to-Virtual>
- [11] *Xen Convert*. Acessado em outubro de 2009. Disponível em: <http://www.citrix.com/English/ss/downloads/details.asp?downloadId=1857892&productId=683148#top>
- [12] *Virt-P2V*. Acessado em outubro de 2009. Disponível em: <http://et.redhat.com/~rjones/virt-p2v/>
- [13] *LibVirt Virtualization API*. Acesado em outubro de 2009. Disponível em: <http://libvirt.org/index.html>

[14] *Mondo Rescue Home Page*. Acessado em outubro de 2009. Disponível em: <http://www.mondorescue.org/>

[15] *Wonders of dd and netcat :: Cloning Operating Systems*. Acessado em outubro de 2009. Disponível em: http://www.rajeevnet.com/hacks_hints/os_clone/os_cloning.html

[16] *Linux P2V*. Acessado em outubro de 2009. Disponível em: http://conshell.net/wiki/index.php/Linux_P2V

[17] LARCHERI, PAOLO. *Xen P2V: how to import a Linux Physical Machine as a Xen DomU*. Acesado em outubro de 2009. Disponível em: <http://tutdeb.debian.blogspot.com/2008/05/xen-p2v-how-to-import-linux-physical.html>

[18] Xen-br wiki. *Migrando um servidor físico Linux para Virtual (p2v)*. Acesado em outubro de 2009. Disponível em: <http://wiki.xen-br.org/index.php?title=P2v-howto>

[19] XenSource. *Xen Manual P2V Process*. Acessado em outubro de 2009. Disponível em: <http://wiki.xensource.com/xenwiki/XenManualPtoVProcess>

[20] ORMIERES, RICARDO. *Migração P2V de RHEL4 com RedHat/Xen*. Acessado em outubro de 2009. Disponível em: <http://blog.seatecnologia.com.br/2009/06/10/migraao-p2v-de-rhel4-com-redhat-xen>

[21] VON HAGEN, WILLIAM. *Professional Xen Virtualization*. Editora Wiley, 2008.

[22] MARTINS, THIAGO. *LiveCd Xen*. Acessado em outubro de 2009. Disponível em: <http://wiki.xensource.com/xenwiki/LiveCD>

[23] Debian Wiki. *Xen HowTo*. Acessado em outubro de 2009. Disponível em: <http://wiki.debian.org/Xen#A.27clocksource.2BAC8-0.3ATimewentbackwards.27>

[24] *Virtualization in Debian Etch*. Acessado em outubro de 2009. Disponível em: http://www.punknix.com/virtualization_uml

[25] *SELinux Project, Svirt*. Acessado em outubro de 2009. Disponível em: <http://selinuxproject.org/page/SVirt>

OUTROS TRABALHOS EM:

www.projetoederedes.com.br