PROJETO DE REDES
www.projetoderedes.com.br

Guilherme Veloso Neves Oliveira

Solução de virtualização completa utilizando VMware e Software Livre: Um Estudo de Caso na CEF

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

Orientador
Prof. Joaquim Quintero Uchôa

Lavras
Minas Gerais - Brasil
2007

PROJETO DE REDES
www.projetoderedes.com.br

Guilherme Veloso Neves Oliveira

Solução de virtualização completa utilizando VMware e Software Livre: Um Estudo de Caso na CEF

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

Aprovada em *Abril de 2007*

Prof. Denilson Vedoveto Martins

Profa. Marluce Rodrigues Pereira

Prof. Joaquim Quintero Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

Agradecimentos

Agradeço a Deus pela oportunidade, a minha mãe pelos princípios, a minha noiva pela compreensão, aos professores pelos ensinamentos e a toda comunidade "*software* livre".

Resumo

O objetivo deste documento é apresentar um estudo de caso utilizando a ferramenta de virtualização VMware para resolver um problema de compatibilidade entre *software* e *hardware*. Este estudo de caso se baseia na realidade vivida pela Caixa Econômica Federal nos útlimos meses de 2006. Nesta ocasião, a Caixa Econômica Federal adquiriu, por meio de um processo de licitação, um novo *hardware* que deveria ser utilizado para executar a solução atual de Ponto de Vendas (PV). Porém, a solução atual de Ponto de Vendas que a Caixa Econômica Federal utiliza em suas agências bancárias foi desenvolvida sob o sistema operacional Microsoft Windows NT Workstation 4.0, o qual não é compatível com o *chipset* que equipa os novos microcomputadores adquiridos. Em virtude desta incompatibilidade, este estudo de caso mostra como uso da virtualização completa tornou possível a execução do atual sistema legado de Ponto de Vendas sob o novo *hardware*.

Gostaria de dedicar este trabalho a duas pessoas em especial. A minha noiva que sempre esteve ao meu lado, apesar da distância geográfica que nos separa atualmente, me apoiando, e dando força para continuar seguindo em frente e nunca desistir. E ao meu amigo Ronaldo Pacheco Wanzeller que foi como um irmão nas horas difícies, sabendo usar as palavras corretas para me proporcionar a paz e a tranquilidade de que precisava para continuar.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Numerosos sistemas têm sido desenhados usando a virtualização para dividir os amplos recursos dos computadores modernos (BARHAM *et al.*, 2003). O objetivo deste documento é apresentar um estudo de caso utilizando a ferramenta de virtualização VMware para solucionar problemas de compatibilidade entre *software* e *hardware*. O estudo de caso se baseia em uma realidade vivida pela Caixa Econômica Federal nos últimos meses de 2006, onde a aquisição de um novo *hardware* tornou a atual solução do sistema de Ponto de Vendas (PV) indisponível. A indisponibilidade aconteceu devido ao fato da atual solução, baseada no sistema operacional Microsoft Windows NT4 Wokstation, não ser compatível com o *chipset* do novo *hardware*. Nos capítulos seguintes serão apresentados os recursos oferecidos por esta ferramenta de virtualização e como eles foram utilizados para solucionar o problema vivido pela Caixa Econômica Federal. Essa ferramenta de virtualização não é composta de apenas um único *software*, mas uma "suite" de programas que oferecem recursos de virtualização de ambientes de *desktop* e servidores.

Computadores modernos são suficientemente poderosos para uso de virtualização de modo que, nos dias de hoje, tornou-se possível a execução de vários sistemas completos (programas básicos e programas aplicativos) sob um único *hardware* físico. A esta representação de vários sistemas completos sendo executados sob um mesmo *hardware* físico dá-se o nome de virtualização de *software*. Isto foi alcançado através da abstração de um *hardware* convencional atribuindo a maior parte, e/ou em alguns casos, toda a sua funcionalidade ao *software*. A este conceito de abstração deu-se o nome de "máquina virtual" que é um *software* representando as funcionalidades de um *hardware* (BARHAM *et al.*, 2003).

O uso da virtualização representa a ilusão de várias máquinas virtuais (VMs) independentes, cada uma rodando uma instância de um sistema operacional virtualizado (SMITH; NAIR, 2005).

O uso da tecnologia de virtualização se difundiu bastante devido ao fato da mesma propor soluções para problemas reais existentes na computação moderna. Mas a idéia de virtualização não é tão recente assim. Inclusive, a virtualização na camada do sistema operacional data de um bom tempo atrás, onde a primeira máquina virtual foi o VM/CMS da IBM que se refere a familia System/370, System/390, zSeries, System z9 IBM mainframes e sistemas compatíveis. Criado no final da década de 70 e utilizado até hoje pela IBM, os sistemas de virtualização estão cada vez mais presentes na realidade tecnológica mundial (WIKIPEDIA - VM_CMS, 2007).

Atualmente, os sistemas virtualizados estão conquistando seu espaço devido ao fato de resolverem alguns pontos que hoje são críticos em diversas empresas tais como: incompatibilidade entre *hardware* e *software* no que diz respeito a suas modificações no decorrer do tempo, ou seja, o *hardware* das empresas estão sendo atualizados a uma velocidade maior que os programas legados que devem ser executados e que são responsáveis por controlar a atividade fim das empresas; subutilização dos recursos de *hardware* pelos programas pelo mesmo motivo citado anteriormente, ou seja, os programas legados não conseguem explorar em sua totalidade a capacidade dos *hardware* atuais; dentre outros.

O sucesso da tecnologia de virtualização se baseou em alguns princípios. Primeiramente, a camada de virtualização deve isolar uma máquina virtual da outra de modo que não exista nenhuma interferência entre ambas. Não é aceitavel que o funcionamento de uma máquina virtual afete a performance de outra máquina virtual (GARFINKEL *et al.*, 2003). Segundo, é necessário suportar uma variedade diferente de sistemas operacionais para acomodar os diferentes aplicativos populares existentes (GARFINKEL *et al.*, 2003). Terceiro, o *overhead* introduzido pela camada de virtualização deve ser pequeno (GARFINKEL *et al.*, 2003).

Importante lembrar que os projetistas do VMware procuraram criar esta ferramenta de virtualização de modo que os sistemas operacionais completos pudessem ser instalados nas máquinas virtuais com o menor esforço possível. Ou seja, os projetistas procuraram não influenciar no desenvolvimento dos sistemas já existentes de modo que estes sistemas operacionais possam executar em ambientes virtuais sem a necessidade de alteração em suas estruturas de funcionamento dependentes de *hardware* (*drivers* dispositivos e algumas partes de alguns subsistemas do *kernel*).

O trabalho apresentado por esta monografia foi dividido de modo que, no capítulo 2, apresenta uma visão geral da tecnologia de virtualização com seus principais tipos e características, além da terminologia utilizada nesta área. O capítulo 3 descreve sobre a suite de software da VMware Inc. apresentando ao leitor as suas principais funcionalidades.

No capítulo 4 é mostrado o estudo de caso realizado na Caixa Econômica Federal. Neste estudo de caso sobre virtualização de software estarão sendo abordados as causas do problema bem como a solução utilizada. Além disso, este capítulo descreve as idéias utilizadas, os procedimentos necessários e os resultados obtidos.

E por último temos o capítulo 5 que descreve uma análise do impacto desta solução no ecossistema da empresa. Além disso, o apêndice A mostra o processo de criação de uma máquina virtual usando o software VMware Workstation e o apêndice B apresenta o código fonte de alguns *scripts* desenvolvidos para permitir a execução deste trabalho

Capítulo 2

Virtualização

Este capítulo apresenta uma visão geral da tecnologia de virtualização descrevendo os três principais tipos e a terminologia utilizada nesta área do conhecimento.

2.1 Tipos de Virtualização

Algumas vezes a indústria costuma utilizar-se de termos diferentes para descrever um mesmo tipo de virtualização. Basicamente, existem três tipos de virtualização: emulação de *hardware* (*hardware emulation*), virtualização completa (*full-virtualization*) e para-virtualização (*para-virtualization*).

2.1.1 Emulação de Hardware

Este tipo de virtualização é considerado o mais complexo devido ao fato de ter que emular de maneira precisa o comportamento de um *hardware*. Isso implica em emular os ciclos de *clock*, o conjunto de instruções, os estados de execução (*pipeline*) do processador e a memória *cache*. Um ponto importante que deve ser lembrado é que este tipo de virtualização se torna útil para os desenvolvedores de *firmware* e de *hardware*, uma vez que a funcionalidade de uma solução pode ser validada sem a necessidade do *hardware* real. Apesar disto, o problema deste tipo de virtualização é a lentidão com a qual a emulação acontece podendo chegar a ser até 1000 vezes mais lento que o suposto *hardware* real (IBM, 2007). Em geral, o *hardware* a ser emulado é bastante diferente do *hardware* real sobre o qual

o sistema de virtualização está sendo executado como mostrado na Figura 2.1. Neste caso temos dois *hardware* emulado: o primeiro sendo emulado pela *virtual machine* (VM) “A” e o segundo pela *virtual machine* (VM) “B”. Esta camada representada pelo *hardware VM A* e pelo *hardware VM B* compõem a camada de virtualização de *hardware*.

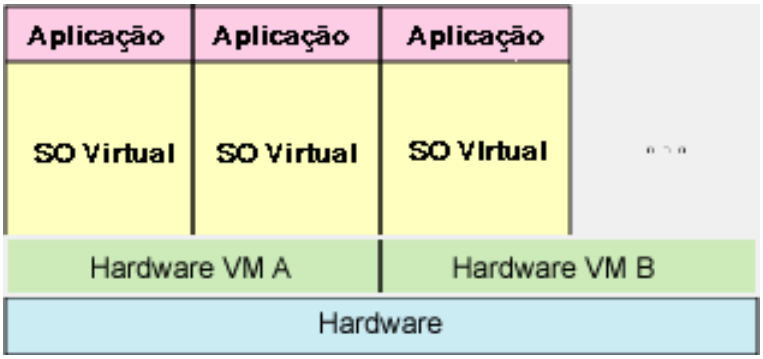

Figura 2.1: Emulação *Hardware*
(IBM, 2007)

2.1.2 Virtualização Completa

Virtualização completa é uma técnica de virtualização utilizada para permitir que qualquer *software* possa ser executado sem alterações. Para isso, esta técnica realiza uma simulação completa do *hardware* da máquina de modo que qualquer sistema operacional possa ser executado. Esta simulação implica em representar o conjunto de instruções do processador, a memória principal, interrupções, exceções e acesso aos diversos dispositivos existentes. Um ponto importante que deve ser considerado é que este tipo de virtualização necessita obrigatoriamente de um *hardware* com características específicas uma vez que instruções de execução privilegiada, como por exemplo as de acesso a *I/O*, devem ser interceptadas e somente serem executadas de acordo com os critérios definidos pela camada *Virtual Machine Monitor - VMM* (SUGERMAN; VENKITACHALAM; LIM, 2001).

Diferentemente da emulação citada anteriormente, a simulação é realizada com maior eficácia, pois não necessita representar os estados de execução do *hardware*. Importante lembrar ainda que a simulação completa do *hardware* feita por esta técnica de virtualização, geralmente, simula dispositivos padrões do mercado de modo a facilitar a instalação e configuração dos sistemas virtualizados. Por exemplo, o VMware ESX, quando instalado em um *hardware* real cuja interface

de rede seja uma placa 3Com, simula para o ambiente virtualizado uma placa de rede AMD PCNet. E esta simulação, se repete também para video, *chipset* e discos rígidos. A Figura 2.2 mostra este tipo de virtualização.

Figura 2.2: *Full-Virtualization*
(NOVELL, 2006)

2.1.3 Para-Virtualização

Para-virtualização é uma técnica de virtualização que apresenta uma *Application Programming Interface* (API) para as máquinas virtuais. Essa API é similar, mas não idêntica ao *hardware* real. Este tipo de técnica requer que o sistema operacional virtualizado seja explicitamente portado para permitir a sua execução. Quando se fala em portar um sistema operacional, logo se imagina uma alteração de grandes magnitudes. Desenvolvedores de sistemas para-virtualizados propuseram então uma solução relativamente simples, criando as *hypercalls*, que são *"system calls"* para o *hypervisor* (The Xen Source, 2006). Ou seja, em vez das "system calls" do sistema operacional virtualizado atuarem diretamente sobre o *hardware* real, elas agora atuarão sobre o *hardware* virtualizado pela *Virtual Machine Moni-*

tor (VMM) de modo que esta camada seja a responsável pela execução de todas as instruções provenientes da máquina virtual (VM) conforme Figura 2.3.

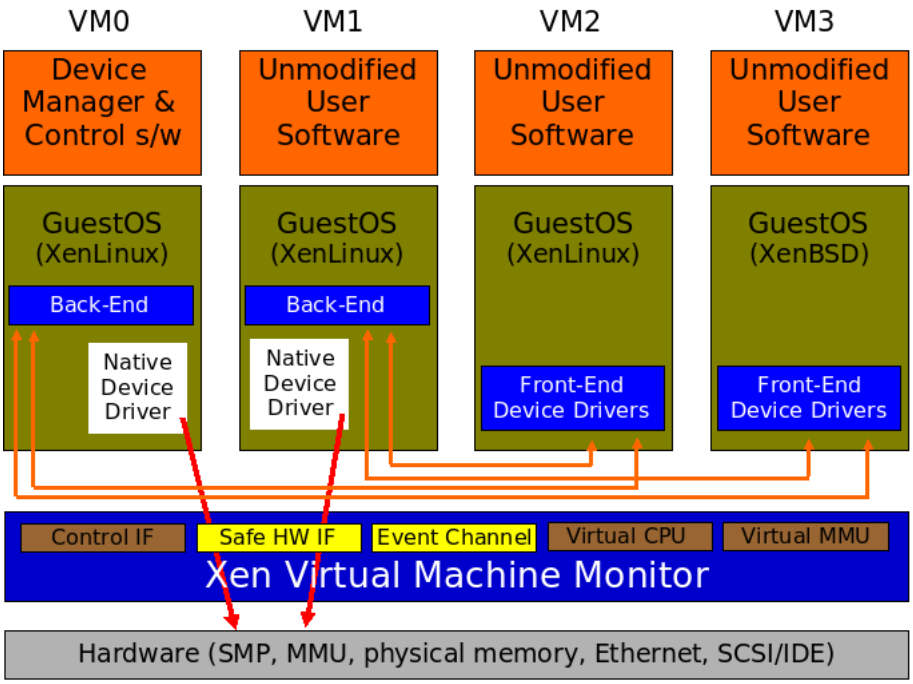

Figura 2.3: Para-Virtualização
(XEN, 2005)

2.2 A camada VMM - *Virtual Machine Monitor*

Em computação, o VMM também é chamado de *Hypervisor*. O *hypervisor* é uma plataforma de virtualização que possibilita a execução concomitante de vários sistemas operacionais em um único *hardware*, como mostra a Figura 2.4. Para realizar tudo isto, o VMM pode atuar de duas maneiras distintas para promover a virtualização:

- O tipo 1 é aquele que é executado diretamente no *hardware* da máquina, como se fosse um sistema operacional. Os sistemas operacionais virtualizados são executados em um segundo nível acima do *hardware*, logo acima

do *hypervisor*. Alguns exemplos são: VMware ESX, Xen, dentre outros. (WIKIPEDIA - Hypervisor, 2007)

- O tipo 2 é aquele que é executado sobre um sistema operacional existente em uma segunda camada. Já os sistemas operacionais virtualizados rodam em um terceiro nível acima do *hardware*, logo acima do *hypervisor*. Alguns exemplos são: VMware Workstation, VMware Player, Microsoft Virtual Server, Xen, dentre outros. (WIKIPEDIA - Hypervisor, 2007)

Segundo (The Xen Source, 2006), o Xen pode atuar como sendo tipo 1 e tipo 2. No caso do tipo 1, atuando como plataforma de virtualização, os produtos XenXource e XenEnterprise são os recomendados. No caso do tipo 2, embutido em um sistema operacional hospedeiro, as distribuições GNU/Linux presentes no mercado trazem consigo instalações já personalizadas, como é o caso do OpenSuse 10.1, Suse Linux Desktop 10, Debian Etch 4.0, RedHat RHEL 5, dentre outros. Além disso, o Xen atuando como tipo 2, sendo executado em um sistema operacional hospedeiro, oferece o recurso de para-virtualização e caso o processador sobre qual o mesmo esteja sendo executado ofereça recursos de virtualização, o Xen também oferece a virtualização completa para os sistemas virtualizados.

É importante lembrar que o *kernel* Linux modificado é também utilizado no produto VMware ESX para permtir a sua execução diretamente no hardware.

2.3 Comparação entre Virtualização Completa e Para-Virtualização

Esta seção tem o objetivo de comparar os dois processos de virtualização: Virtualização Completa e Para-Virtualização. A para-virtualização possui melhor desempenho que a virtualização completa. Isso se deve, basicamente, ao fato dos *drivers* de dispositivos que são executados nas máquinas virtuais para-virtualizadas serem os *drivers* reais para os dispositivos físicos conforme Figura 2.3, diferentemente dos *drivers* emulados da virtualização completa.

Os gráficos apresentados nas Figuras 2.5 e 2.6 mostram os resultados de desempenho obtidos de um sistema real GNU/Linux, um sistema para-virtualizado utilizando o Xen e um sistema com virtualização completa utilizando o VMware Workstation. Importante salientar que toda a análise comparativa entre a virtualização completa e a para-virtualização é feita em relação ao sistema real GNU/Linux.

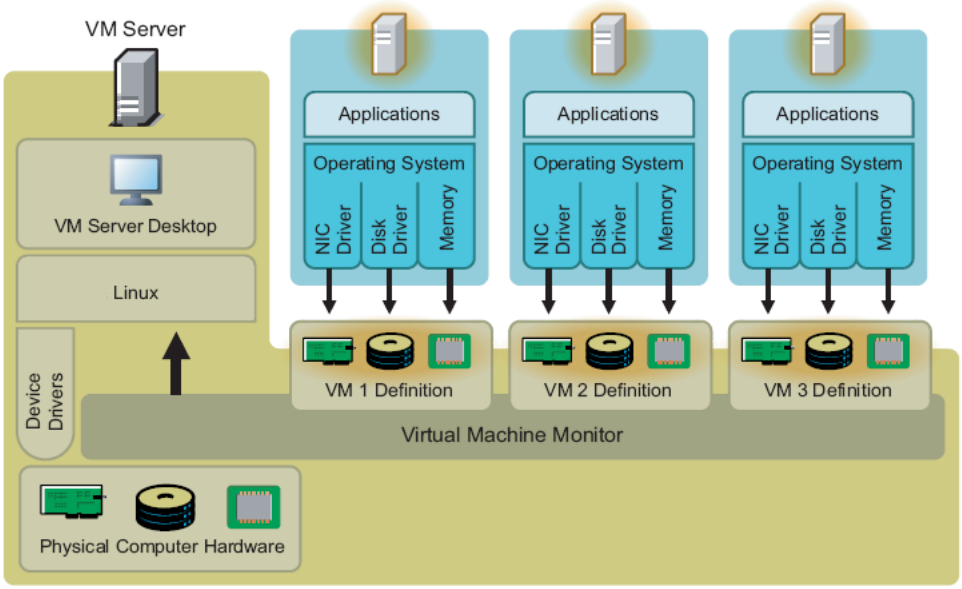

Figura 2.4: Exemplo Virtualização com Máquinas Virtuais
(NOVELL, 2006)

Na Figura 2.5 o primeiro teste de performance realizado foi o "*SPEC INT 2000*". Este teste consiste na execução de softwares que realizam diversas tarefas tais como: compressão e descompressão de dados usando Bzip2 e Gzip, compilação de códigos em "C" utilizando o GCC, dentre outros conforme (Standard Performance Evaluation Corporation, 2003). Este primeiro teste visa medir o desempenho do processador e da mémoria RAM do equipamento. Conforme mostrado na figura em questão, pode-se perceber que não existe diferença de perfomance entre o GNU/Linux e o Xen. No caso do VMware Workstation existe uma pequena perda de performance que não chega a 5%. Desta forma é possível verificar que ambas as soluções de virtualização apresentaram resultados satisfatórios.

Ainda na Figura 2.5, verifica-se também que o terceiro teste de performance não foi satisfátorio para o VMware Workstation. Isso se deve ao fato do teste "OSDB-OLTP" medir o desempenho de acesso a banco de dados, ou seja, no caso deste benchmark existe um intenso acesso ao(s) disco(s) rígido(s) da máquina. Como dito anteriormente, pode-se comprovar que o fato da virtualização completa emular os *drivers* de dispositivos traz como consequência um perda significativa de desempenho em relação a para-virtualização que utiliza o *driver* real. Conforme

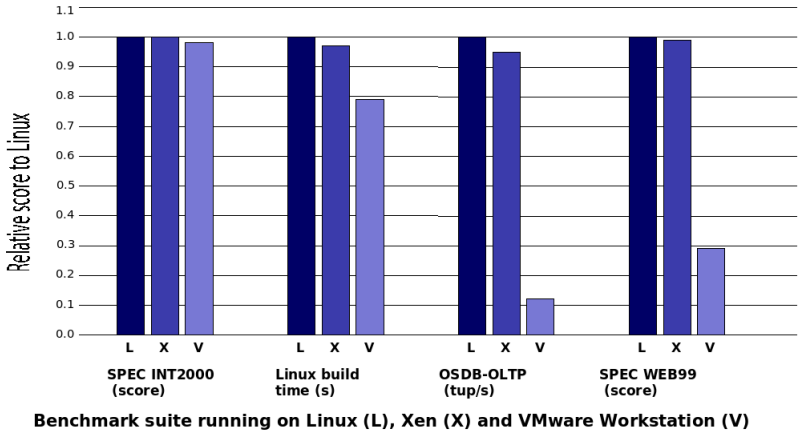

Figura 2.5: Gráfico de Performance
(XEN, 2005)

Figura 2.5 verifica-se que o VMware Workstation teve uma perda na performance de aproximadamente 80% em relação ao Xen.

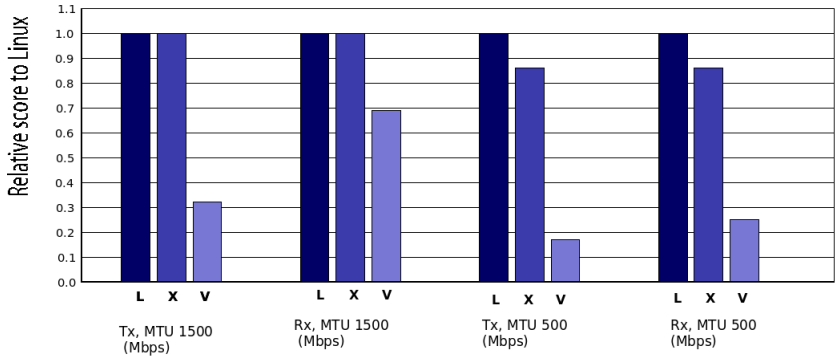

Figura 2.6: Gráfico de Performance
(XEN, 2005)

Já na Figura 2.6, é mostrado o desempenho na transmissão (TX) e na recepção (RX) de quadros (*frames*) de rede. Neste teste foram realizados transferências de dados utilizando-se tamanhos diferentes de quadros. No primeiro grupo temos *frames* de 1500 *bytes* e no segundo grupo temos *frames* de 500 *bytes*. Como estamos falando em camada de "enlace" do modelo OSI, estamos analisando de maneira direta o *driver* de rede presente em cada uma das soluções de virtualização. Con-

forme mostra a Figura 2.6, pode-se comprovar novamente que a emulação dos *drivers* de dispositivos trazem prejuízos de performance significativos para o sistema que faz uso da virtualização completa. Ou seja, que o desempenho das máquinas virtuais rodando sobre VMware Workstation foi bastante inferior ao desempenho das máquinas virtuais rodando sobre o Xen.

Mais informações sobre o desempenho de máquinas virtuais sobre o Xen podem ser vistos em (CLARK *et al.*, 2005).

Informações sobre a implementação e o funcionamento dos *drivers* virtualizados no VMware Workstation podem ser encontrados em (SUGERMAN; VENKITACHALAM; LIM, 2001).

2.4 Terminologia utilizada

Devido ao crescimento rápido da área de virtualização, assim como em qualquer outra área, surgem os diversos termos utilizados. Para que seja possível o entendimento pleno deste trabalho, esta seção visa esclarecer a terminologia utilizada nesta área. Importante lembrar que alguns termos, às vezes, são utilizados de maneira errada pelo simples desconhecimento de tal terminologia. Seguem-se os principais conceitos da área, adaptados de (NOVELL, 2006):

Máquina Virtual (*Virtual Machine* - VM): Refere-se a instância de um *hardware* virtualizado e um sistema operacional também virtualizado normalmente sob a forma de simulação, ou seja, uma interface com o ambiente, diferentemente da emulação que refletiria todos os estados internos do ambiente ao mesmo tempo. Uma máquina virtual pode executar qualquer tipo de *software* como um servidor, um cliente ou um *desktop*. Esta máquina virtual também é chamada de computador virtual, convidado, *domain* U, *domU* ou domínio sem privilégios.

Servidor de Máquina Virtual (*Virtual Machine Server* - VMServer): Refere-se ao *hardware* físico e ao *software* de virtualização que combinados formam a máquina virtual *host*. É o principal componente do ambiente de virtualização uma vez que é o responsável por prover os recursos necessários a criação de uma e/ou várias máquinas virtuais. Ele é também chamado de *host*, *dom0*, virtual0 ou domínio privilegiado.

Sistema Operacional Para-virtualizado (*Paravirtualized Operating Systems*): Refere-se ao sistema operacional que possui capacidades de ser executado

neste tipo de modo. Ou seja, um sistema operacional modificado para ser executado em modo para-virtual. Também chamado de *VM-aware*, *modified* ou *optimized guest*.

Sistema Operacional Nativo (*Native operating Systems*): Refere-se a um sistema operacional típico que não necessita de alteração para ser executado em um ambiente virtualizado. Este tipo de sistema operacional deve ser executado em modo *full virtualization* uma vez que o mesmo não pode ser alterado para ser executado em modo para-virtual. É também chamado de *shrink-wrapped*, *out-of-the-box*, *unmodified* ou *full virtualized guest*.

Tecnologia de Virtualização (*Virtualization Technology*): Refere-se ao *hardware* que suporta a tecnologia de virtualização como Intel VT ou AMD *Virtualization*. *Hardware* VT trabalham em conjunto com *software* de virtualização de modo a permitir uma virtualização completa dos dispositivos de *hardware*, possibilitando a execução de um sistema operacional sem alterações. Um *hardware* com este tipo de tecnologia é requerido para possibilitar o modo *full virtualization*.

Computador Padrão (*Standard Computer*): Refere-se a um *hardware* que não possui suporte especial a tecnologia de virtualização e não pode executar sistemas operacionais que requerem o modo *full virtualization* para serem executados.

Capítulo 3

Virtualização usando VMware

Este capítulo apresenta o uso da tecnologia de virtualização utilizando-se o conjunto de *software* disponibilizado pela VMware Inc. O tipo de virtualização oferecido por estas ferramentas é a virtualização completa como citado anteriormente na seção 2.1.2. Basicamente, a suite de *software* deste fabricante se divide em três produtos: VMware Workstation, VMware Player e VMware ESX Server. Os dois primeiros são utilizados na virtualização de ambientes *desktop* e o último na virtualização de ambientes servidores.

3.1 Virtualização de Ambientes *Desktop*

As ferramentas descritas nesta seção necessitam de um sistema operacional nativo instalado na máquina de modo a possibilitar a sua execução. Como dito anteriormente na seção 2.2, o VMware Workstation e o VMware Player podem ser classificados como sendo do tipo 2.

O VMware Workstation é um utilitário não gratuito, porém no site do fabricante `http://www.vmware.com`, através do preenchimento de um formulário, é possível adquirir o *software* por um período de utilização de até 90 dias.

O *software* VMware Workstation é utilizado para criar as máquinas virtuais. Criar uma máquina virtual significa definir o *hardware* desta máquina bem como o tipo de sistema operacional virtualizado que a mesma irá executar.

A criação de máquinas virtuais se divide em duas etapas. Na primeira deve-se definir o sistema operacional a ser virtualizado bem como os dispositivos de *hard-*

ware presentes. Neste caso, os dispositivos possíveis são: disco rígido, *Network Interface Card - NIC*, *USB*, processador, *driver* de disquete, adaptador de som, *mouse*, memória RAM, *DVD/CD-ROM*, portas seriais e portas paralelas. Na segunda etapa deve-se instalar o sistema operacional virtualizado seguindo os mesmos procedimentos habituais já utilizados para a instalação do mesmo. A diferença é que o sistema operacional será instalado sobre um *hardware* virtualizado. O processo de criação de uma máquina virtual está descrito no apêndice A.

Os procedimentos de instalação deste *software* no ambiente GNU/Linux são relativamente fáceis, desde que todas as suas dependências estejam instaladas. Abaixo seguem as dependências e procedimentos:

- Instalar os *headers* do *kernel* específicos para o *kernel* que está em execução na sua máquina.

- É necessário que esteja instalado na máquina o compilador *"GCC"* e o linkador *"ld"* para realizar a compilação e linkagem dos módulos "*vmnet*" e "*vmmon*". Estes dois módulos são os responsáveis por oferecer serviços de rede a máquina virtual. Os serviços oferecidos são: *"bridge", "NAT", "host only"*, roteamento e *DHCP*. Importante lembrar que estes módulos são compilados com o recurso *"version mágic"*, ou seja, somente irão funcionar no *kernel* específico em que foram compilados.

- Instalar o servidor gráfico X, bem como as bibliotecas libX11, libXrender, libXtst, libXext, libXt, libICE, libSM, para citar algumas. Todas são necessárias para execução do *software* VMware Wokstation. Importante salientar que para compilação e linkagem dos módulos "*vmnet*" e "*vmmon*" estas bibliotecas gráficas não são necessárias. Caso queira saber as bibliotecas utilizadas pelo *software* VMware Workstation basta digitar o comando: "ldd /usr/lib/vmware/bin/vmware".

- Extrair o pacote do VMware Workstation que pode ser obtido no *site* `http://www.vmware.com`. Basicamente, para sistemas GNU/Linux existem dois tipos de formato de pacote: tar.gz e rpm. Em qualquer um dos casos, deve-se extrair/instalar o pacote e logo após executar o comando "./vmware-install.pl". A partir daí, basta responder as perguntas da primeira parte do processo de instalação com as opções padrão. Esta primeira parte do processo de instalação inclui: documentação, *scripts* de inicialização, *scripts* executáveis utilizados e binários. Ao final desta etapa, automaticamente, é executado o *script* "vmware-config.pl" que realizará a segunda parte do processo de instalação. Esta segunda parte do processo solicita que seja

especificado os recursos de rede que serão oferecidos as máquinas virtuais podendo ser: *bridge*, *NAT* e/ou *host only*. Após isto, o processo de compilação e linkagem dos módulos é executado e finalmente os módulos são carregados para memória. Neste caso, os módulos "*vmnet*" e "*vmmon*" podem ser identificados como estando em execução com o uso do comando "lsmod | grep vm"

Importante lembrar que o pacote que contém o *software* VMware Workstation traz consigo o pacote do *software* VMware Player, não se fazendo necessário instalar o VMware Player separadamente.

O utilitário VMware Player pode ser adquirido de maneira independente. Ou seja, aqueles usuários que apenas desejam executar uma máquina virtual previamente configurada, podem fazer uso apenas deste *software*. Este utilitário é gratuito e seu processo de instalação é idêntico ao do VMware Workstation descrito anteriormente. Caso seja necessário realizar alguma alteração na estrutura de *hardware* da máquina virtual, deve-se utilizar o VMware Workstation. Na Figura 3.1 é apresentada a utilização deste *software* em modo gráfico.

Figura 3.1: VMware Player

3.2 Virtualização de Ambientes Servidores

A ferramenta descrita nesta seção não necessita de um sistema operacional nativo instalado na máquina de modo a possibilitar a sua execução. Diferentemente do VMware Workstation e do VMware Player, o ESX Server 3.0 é classificado como sendo do tipo 1 de acordo com a seção 2.2

Assim como o VMware Workstation, o VMware ESX 3.0 não é gratuito, apesar de utilizar o *kernel* do linux para controlar a máquina real. As alterações feitas no *kernel* do linux, pela VMware Inc., não foram, até o presente momento em que este texto estava sendo escrito, disponibilizados a comunidade de software livre, o que impossibilita este trabalho de mostrar as reais alterações realizadas. Além disso, *software* GNU também são utilizados nesta solução que a VMware INC batizou como sendo ESX Server. Esta versão do VMware ESX 3.0 foi desenvolvida com base na distribuição GNU/Linux RedHat Enterprise Linux 3.0.

As versões de *software* GNU, bem como do *kernel* do linux utilizado podem ser vistas na tabela 3.1

Tabela 3.1: Versões de Softwares GNU e Linux

Softwares	Versões
glibc	2.3.2
Linux	2.4.24
GCC	3.3.5
binutils	2.15
coreutils	4.5.3

Capítulo 4

Estudo de caso - Virtualização de ambientes *Desktop* usando VMware

Este capítulo apresenta um estudo de caso do uso da tecnologia de virtualização de *desktop* em máquinas da rede da Caixa Econômica Federal. A solução apresentada neste capítulo tem por objetivo mostrar como a virtualização de *software* possibilitou a resolução de um problema que se baseia praticamente na seguinte premissa: incompatibilidade entre *hardware* e *software*.

4.1 Histórico

Atualmente, uma grande parte da rede da Caixa Econômica Federal é baseada no ambiente da Microsoft. As máquinas da rede, em sua maioria, ainda utilizam o Microsoft Windows NT4 Workstation como sistema operacional. Neste estudo de caso, estão sendo analisadas as máquinas que compõem o que chamamos de Ponto de Venda (PV). Os pontos de venda são compostos por computadores rodando o sistema operacional Microsoft Windows NT4 Workstation e o *software* SIAPV (Sistema de Automação de Ponto de Venda) desenvolvido pela Hewllet Packard.

Em recente processo de licitação aberto pela Caixa Econômica Federal para aquisição de novas máquinas visando equipar os PV's atuais, a empresa vencedora do processo forneceu a Caixa Econômica Federal um lote de máquinas equipados

com *chipset* Intel 915i que é incompatível com a solução de *software* (NT4 + SIAPV) existente para os pontos de venda. A incompatibilidade se deve ao fato do sistema operacional NT4 workstation, já descontinuado pela Microsoft, não oferecer suporte ao *chipset* 915i da Intel.

Dentre algumas soluções apresentadas, dois itens foram considerados importantes e decidiram, dentre elas, qual seria utilizada. Os itens analisados foram: tempo hábil para funcionamento dos PV's sob o novo *hardware* e impacto da nova solução para os PV's. Mediante estes dois critérios, decidiu-se pela utilização do processo de virtualização de *software* por atender de maneira integral aos dois itens anteriores.

A opção por utilização do sistema operacional GNU/Linux como sistema operacional nativo para prover esta solução se deve basicamente a três fatores: facilidade de customização, licensa GPL e estabilidade.

4.2 *Hardware* e *Software* Utilizados

O *hardware* utilizado por esta solução possui a seguinte configuração:

- Processador Pentium IV com tecnologia *Hyper-Threading(HT)*
- 512 MB de memória RAM com tecnologia *DDR2*
- Disco rígido de 80 GB com tecnologia *SATA2*
- Placa mãe Intel com *chipset* 915i
- Placa de rede ethernet onboard *Gygabit Ethernet PCI Express*
- Uma porta paralela e duas portas seriais

Dentre as diversas distribuições GNU/Linux existentes no mercado, a distribuição escolhida foi o Debian Sarge 3.1r2. A escolha se baseou em algumas características tais como: instalação inicial mínima e bastante enxuta com apenas *software* necessário, estabilidade conhecida pelos desenvolvedores desta solução de virtualização e principalmente a facilidade de customização da distribuição.

4.3 Visão Geral do Processo

Esta seção tem o objetivo de mostrar uma visão macro de todo o processo. Desde a criação das imagens, passando pelo processo de instalação até o seu pleno funcionamento.

Este processo será dividido em quatro partes:

- Processo de criação da imagem do sistema operacional hospedeiro (GNU/-Linux)
- Processo de criação da imagem do sistema operacional virtualizado (NT4 + SIAPV)
- Processo de instalação do sistema hospedeiro (GNU/Linux)
- Processo de instalação do sistema virtualizado (NT4 + SIAPV)

O "Processo de criação da imagem do sistema operacional hospedeiro (GNU/-Linux)" é feito por meio da instalação do sistema em um *hardware* real, customização do mesmo e compactação desta imagem em um arquivo denominado imagem_linux_debian.tar.gz.

O "Processo de criação da imagem do sistema operacional virtualizado (NT4 + SIAPV)" é feito por meio da instalação do sistema em um *hardware* virtualizado. Neste caso será utilizado o Vmware Workstation para criação das máquinas virtuais conforme apêndice A. Após a instalação desta máquina virtual, os arquivos gerados são compactados em um arquivo denominado windows_nt_et.tar.gz.

O "Processo de instalação do sistema hospedeiro (GNU/Linux)" se dá por meio de um LiveCD customizado para tal finalidade. Este LiveCD possui um sistema básico que permite a instalação do sistema hospedeiro pré-configurado (imagem_linux_debian.tar.gz). Após a finalização deste processo, o sistema é reiniciado e então o sistema hospedeiro instalado é inicializado.

Em seu primeiro carregamento será dado continuidade ao processo de instalação da solução. Passamos então ao "Processo de instalação do sistema virtualizado(NT4 + SIAPV)". A partir deste ponto é solicitado um outro CD de instalação, agora contendo a imagem do sistema virtualizado. Esta etapa consiste na extração do sistema virtualizado (windows_nt_et.tar.gz) para um diretório local presente no sistema de arquivos do sistema hospedeiro.

4.4 Criação da imagem do Sistema Operacional Hospedeiro (GNU/Linux)

O processo de criação desta imagem consiste em instalar o sistema operacional Debian Sarge em um *hardware* real, configurá-lo e customizá-lo de maneira adequada e então compactá-lo em um arquivo denominado imagem_linux_debian.tar.gz.

O processo de instalação do Debian Sarge 3.1r2 foi feito como "instalação mínima" de acordo com o processo de instalação da distribuição que pode ser encontrado em (Equipe instalador Debian, 2005). Após a instalação mínima, alguns *software* foram instalados/alterados:

- *Kernel* Linux teve que ser alterado, da vesrão 2.6.8-2-386 sem suporte a *SMP* para o *kernel* 2.6.11 com suporte *SMP* visando aproveitar ao máximo os recursos do processador Pentium IV (*Hyper Threading*) que equipa os computadores em questão conforme descrito anteriormente na seção 4.2. O processo de instalação deste pacote foi realizado através do comando "aptitude install kernel-image-2.6.11-smp". É importante lembrar que este pacote do *kernel* não existe nos repositórios oficiais da distribuição. Neste caso em específico, está sendo utilizado um repositório interno presente na rede da Caixa Econômica Federal que foi instalado e configurado para tal finalidade.

- O programa "mingetty" foi instalado para permitir que o procedimento de autologin pudesse acontecer sem a necessidade de se digitar na linha de comando uma senha para validação da seção de um usuário. O processo de instalação deste pacote foi realizado através do comando "aptitude install mingetty".

- O *software* VMware Player foi instalado para permitir a execução da máquina virtual. O processo de instalação deste pacote foi realizado através do comando "aptitude install vmplayer". Lembrando que este pacote somente existe no repositório interno presente na rede da Caixa Econômica Federal e é especifico para ser executado com o *kernel* 2.6.11 customizado para está solução.

- O servidor gráfico X foi instalado para permitir a execução do *software* VMware Player bem como do sistema operacional virtualizado. O processo de instalação deste servidor gráfico foi realizado através do seguinte comando: "aptitude install xserver-xfree86 xutils xbase-clients xfonts-base"

Além da instalação e configuração dos *software* mencionandos acima, é necessário a criação de um usuário local. O nome que este trabalho apresenta é o de um usuário imaginário chamado "fulano" que será o responsável por executar a máquina virtual. A criação deste usuário se fez através da utilização do comando "useradd -m fulano". O nome real do usuário que executa a máquina virtual esta sendo omitido por questões de segurança.

4.4.1 Customização do Sistema Operacional

A customização do sistema operacional se baseou em algumas premissas:

1. O sistema operacional GNU/Linux deve ficar inacessível para o usuário final.

2. O sistema operacional virtualizado deve inicializar de maneira automática. Ou seja, o próprio sistema hospedeiro deve inicializar o sistema operacional virtualizado sem a intervenção do usuário.

3. O sistema hospedeiro deve possuir uma única porta de acesso para que, em caso de problemas seja possível acessar a máquina remotamente e analisá-la.

As premissas 1 e, parcialmente a 2, foram atendidas através da alteração do arquivo "/etc/inittab" que é lido pelo processo "init". Este processo é o primeiro a ser carregado pelo *kernel* Linux e o responsável por executar os procedimentos de inicialização do sistema. A alteração visa garantir que nenhum *tty*, além do *tty2* seja inicializado. Além disso, o *software* "mingetty" está sendo utilizado para permitir que o usuário fulano efetue *logon* na máquina sem necessitar de digitar qualquer tipo de senha de maneira interativa. A Figura 4.1 exibe apenas as linhas alteradas deste arquivo.

A premissa 2 é definitivamente atendida com alterações/criações em outros arquivos de configuração tais como:

- /home/fulano/.bash_profile
- /usr/bin/controle_X
- /usr/bin/X11/inicia_vm

```
# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
#  <id>:<runlevels>:<action>:<process>
#
# Note that on most Debian systems tty7 is used by the X Window
# System, so if you want to add more getty's go ahead but skip
# tty7 if you run X.
#
#1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/mingetty --autologin fulano tty2
#3:23:respawn:/sbin/getty 38400 tty3
#4:23:respawn:/sbin/getty 38400 tty4
#5:23:respawn:/sbin/getty 38400 tty5
#6:23:respawn:/sbin/getty 38400 tty6
```

Figura 4.1: Arquivo de configuração "/etc/inittab"

```
#!/bin/bash

startx /usr/bin/X11/inicia_vm -- -br &

OPCAO=1

sleep 25

while [ $OPCAO -eq 1 ]
do
   PID=$(ps ax | grep vmplayer | cut -d " " -f2)
   PID=$(echo $PID | cut -d " " -f1)

   if [ -z $PID ]
   then
      OPCAO=0
   fi

done
sudo nohup /sbin/shutdown -h now &
```

Figura 4.2: *Script* de configuração "/usr/bin/controle_X"

O arquivo ".bash_profile" é lido no momento em que o usuário "fulano" efetuar o seu *login* no sistema. Este arquivo possui uma chamada para o *script* criado

denominado de “controle_X”. Este *script* tem a finalidade de iniciar a execução do servidor gráfico através da chamada ao comando “startx” passando como parâmetro o que o mesmo deverá executar, conforme pode ser observado na Figura 4.2. Além disso, este *script* tem a finalidade de monitorar o funcionamento do *software* VMware Player. Ou seja, o sistema virtualizado será executado enquanto o VMware Player estiver em execução. No momento em que o VMware Player não estiver mais sendo executado, é porque o usuário solicitou o desligamento da máquina virtual e consequentemente do sistema hospedeiro.

Uma vez chamado o servidor gráfico através da execução do comando “startx”, o mesmo irá carregar o *script* “inicia_vm”. Este *script* tem a finalidade de realizar o ajuste do volume do sistema operacional hospedeiro, remover arquivos temporários provenientes de desligamento inadequado desde a última utilização e então executar o VMware Player conforme Figura 4.3. É possível verificar que o parâmetro passado para o VMware Player é o caminho completo do arquivo de configuração da máquina virtual que possui a extensão “.vmx”.

```
#!/bin/bash
aumix -v -100
aumix -c -100
aumix -v +70
aumix -c +70
rm -f /home/fulano/vmware/000C29A47B76/Windows_NT_ET/*.vmss
rm -f /home/fulano/vmware/000C29A47B76/Windows_NT_ET/*.WRITELOCK
vmplayer \
/home/fulano/vmware/000C29A47B76/Windows_NT_ET/Windows_NT_ET.vmx
```

Figura 4.3: *Script* de configuração “/usr/bin/X11/inicia_vm”

A premissa de que o sistema operacional hospedeiro fosse acessado remotamente foi atendida através da instalação e configuração do servidor *OpenSSH*. Para isto basta digitar o comando “aptitude install sshd” e proceder com a configuração padrão.

Após todos os procedimentos descritos acima é hora de compactar a imagem em um único arquivo. Para isto deve-se utilizar o comando “tar”. Para tal procedimento desloque-se para a partição “/” com o comando “cd /” e crie um arquivo texto denominado “exclude” conforme Figura 4.4. Depois basta digitar o comando “tar -cvzf imagem_linux_debian.tar.gz -X exclude .”. Este arquivo exclude tem a finalidade de fazer com que o comando “tar” não inclua no arquivo “tar.gz” os diretórios e/ou arquivos listados pelo arquivo “exclude”

```
./proc/*
./tmp/*
./sys/*
./var/*
./exclude
./imagem_linux_debian.tar.gz
```

Figura 4.4: Arquivo “exclude”

4.5 Criação do Sistema Operacional Virtualizado (NT4 + SIAPV)

Este processo ocorre em uma máquina separada que esteja com o *software* VMware Workstation instalado e configurado. Após a criação da máquina virtual conforme apêndice A deve-se proceder com a instalação do sistema operacional dentro da máquina virtual de acordo com os procedimentos específicos do sistema operacional em questão. O procedimento de instalação do sistema operacional virtualizado ocorre de maneira trivial na máquina virtual. Uma vez instalado este sistema na máquina virtual, os arquivos gerados por esta instalação são comprimidos em um arquivo denominado “windows_nt_et.tar.gz”.

Neste caso o arquivo gerado ultrapassa o tamanho de um CD, se fazendo necessário dividí-lo para permitir a sua gravação em CD’s. Então, para dividí-lo é necessário a utilização do comando “split”. Para isto deve-se utilizar o comando da seguinte forma: “split -b 695m windows_nt_et.tar.gz windows_nt_et.tar.gz”. O resultado será a criação de dois arquivos cujos nomes são: windows_nt_et.tar.gzaa e windows_nt_et.tar.gzab. Estes aqruivos contém o sistema operacional virtualizado NT4 + SIAPV que serão instalados no sistema operacional hospedeiro na seção 4.7. No caso desta solução, o arquivo “windows_nt_et.tar.gzaa” foi gravado em um CD de dados individual pelo fato de estar com o tamanho de 695 MB. Já o outro arquivo, “windows_nt_et.tar.gzab” foi inserido no LiveCD na seção 4.6 juntamente com o sistema hospedeiro GNU/Linux (imagem_linux_debian.tar.gz) uma vez que o seu tamanho é de apenas 117 MB.

4.6 Processo de instalação do Sistema Hospedeiro GNU/-Linux

Esta etapa consiste em mostrar o que foi feito para permitir a instalação do sistema hospedeiro GNU/Linux pré-configurado na seção 4.4.

Nesta etapa fez-se necessário a criação de um LiveCD customizado para permitir a instalação do sistema operacional GNU/Linux pré-configurado. Este LiveCD também é composto de um outro sistema operacional GNU/Linux customizado para tal finalidade. O processo de criação deste LiveCD foi feito utilizando o *software* "bootcdwrite". O "bootcdwrite" tem a função de criar um LiveCD de um sistema previamente instalado e configurado em um *"hard disk"*. Portanto o procedimento adotado para geração deste LiveCD foi instalar um sistema Debian, que não é o sistema hospedeiro definitivo, em um disco rígido e customizá-lo para que realizasse o processo de instalação do sistema hospedeiro GNU/Linux. A customização deste LiveCD se fez com os seguintes passos:

- Instalação de um sistema básico Debian.
- Criação de um usuário chamado "caixa" com a utilização do comando "useradd -m caixa".
- Instalação do *software* "bootcdwrite" com a utilização do comando "aptitude install bootcdwrite".
- Instalação do *software* "dialog" com a utilização do comando "aptitude install dialog". Este *software* se faz necessário, uma vez que toda a interação, durante o processo de instalação realizada pelo *script* "script_instalacao.sh", é feita por meias de caixas de diálogo.
- Alteração do arquivo ".bash_profile" presente no diretório "home" do usuário "caixa" adicionando uma chamada na última linha deste arquivo para o *script* "script_instalação.sh". O código deste *script* pode ser visto na seção B.1.
- Copiar o *script* "script_instalação.sh" para o diretório "home" do usuário "caixa".
- Copiar o arquivo (imagem_linux_debian.tar.gz) que contém a imagem do sistema hospedeiro GNU/Linux para o diretório *home* do usuário caixa.

- Executar o comando “bootcdwrite -s” para geração de um arquivo ISO da referida instalação.
- Gravar um CD, que já é o LiveCD customizado, com o arquivo ISO produzido no passo anterior.

Basicamente a funcionalidade deste LiveCD é formatar o disco rigido da máquina, extrair o sistema hospedeiro pré-configurado (imagem_linux_debian.tar.gz), criar os arquivos fstab e mtab, instalar o *grub* e alterar uma linha do arquivo inittab para permitir que no primeiro *boot* da máquina seja dado prosseguimento ao processo de instalação. Para isto tornou-se necessário a criação de um *script*, citado anteriormente, que realizasse este processo de maneira automática. O *script* em questão é o “script_instalação.sh” que pode ser visualizado na seção B.1

4.7 Processo de instalação do Sistema Virtualizado

Uma vez instalado o sistema hospedeiro, no seu primeiro carregamento existe a execução de quatro *scripts*.

1. inicializa.sh
2. install_win_nt.sh
3. renomeia.sh
4. inservlandesk.sh
5. landesk.sh
6. hora.sh

O primeiro a ser chamado é o *script* “inicializa.sh” conforme Figura 4.5. Este *script* tem apenas a finalidade de chamar cada um dos outros três *scripts*.

O próximo a ser chamado é o “install_win_nt.sh” conforme seção B.2. A sua função é instalar o sistema operacional virtualizado. O processo de instalação se dá através da extração do arquivo “windows_nt_et.tar.gz” no diretório apropriado. Um ponto importante que deve ser observado neste *script* é a funcionalidade da variável “$DIRVM”. Esta variável contém o endereço *MAC* da placa de rede do equipamento real. Isto se fez necessário porque o VMware gera um *MAC* para as

máquinas virtualizadas com base no caminho completo do diretório de instalação das mesmas. Portanto se os caminhos de instalação de duas máquinas virtuais fossem os mesmos, consequentemente ambas as máquinas teriam o mesmo endereço *MAC*. Caso estas máquinas estivessem no mesmo segmento de rede, o funcionamento das mesmas estaria indisponível. Em virtude disto, optou-se por criar um diretório com base no endereço *MAC* da placa real de modo a fazer com que o caminho completo de instalação de cada máquina virtual seja diferente do caminho completo de instalação de qualquer outra máquina virtual.

O próximo *script* da lista a ser executado é o "renomeia.sh" conforme seção B.3. Este *script* tem a funcionalidade de atribuir um nome válido à máquina hospedeira GNU/Linux conforme indicado pelo técnico que estará realizando o processo de instalação da imagem. Neste caso foi feito o uso de expressões regulares para garantir que o nome de máquina informado pelo técnico estivesse de acordo com os normativos internos da Caixa Econômica Federal no que diz respeito a nomes de computadores.

E então o último *script* é executado conforme seção B.4. Neste *script*, os endereços *IP* referentes aos servidores da rede da Caixa Econômica Federal foram omitidos. Em substituição foram adicionados marcações como "x.x.x.x". O "inservlandesk.sh" é executado e tem a responsabilidade de criar um *script* chamado "landesk.sh" que será responsável por realizar o inventariado da máquina hospedeira GNU/Linux. Este *script* deve ser criado no momento do processo de instalação, uma vez que não é possível determinar com precisão o endereço do servidor *landesk* previamente. Além disso, cada máquina de cada estado, deve reportar ao servidor *landesk* do seu estado. Neste *script* "landesk.sh" existe a chamada ao comando "ldiscnux" passando como parâmetro o endereço do servidor para o qual as informações deverão ser reportadas. Além disso, para permitir a execução deste *script* "landesk.sh" de maneira aleatória, foi criado um outro *script* chamado de "hora.sh" que é executado toda vez que a máquina hospedeira é ligada e que tem a responsabilidade de realizar um agendamento na *cron* informando em que momento o inventariado da máquina acontecerá conforme seção B.5. A janela de horário disponível para realização do inventariado das máquinas ficou definida para acontecer no período das 8:00 às 19:00h. A necessidade deste último *script* surgiu para evitar que todas as máquinas se reportassem ao servidor no mesmo instante de tempo. Os dados referentes a este equipamento são enviados para um servidor regional para permitir um controle por parte da gerência responsável.

Ao final do *script* "inicializa.sh" o arquivo "/etc/inittab" é novamente alterado para permitir que nos próximos *"boots"* o *runlevel* de execução padrão seja inicializado.

Após todo o processo descrito acima, a nova solução do PV é inicializada deixando disponível para o usuário o sistema legado de ponto de venda constituido por NT4 mais SIAPV.

```
#!/bin/bash

dmesg -n1

#Script de instalação Windows Virtualizado
/etc/init.d/install_win_nt

#Script para renomear a máquina GNU/Linux
/etc/init.d/renomeia

#Script para inventariar a máquina GNU/Linux
/etc/init.d/inservlandesk

#Alterando o arquivo /etc/inittab para inicialização do \
# ambiente virtualizado
sed -i "s|inicializa|rc 2|" /etc/inittab

clear
confirma=1
while [ $confirma != 0 ]; do

   confirma=$(dialog --stdout --msgbox "CAIXA ECONÔMICA FEDERAL \n \
   Processo Finalizado com sucesso !!!!" 7 60; echo $?);
done
sleep 1
reboot
```

Figura 4.5: *Script* do Primeiro Boot "inicializa.sh"

4.8 Resultados Obtidos

O processo de instalação desta solução começou a ser feito em dezembro de 2006 e atualmente existem aproximadamente 10.000 máquinas utilizando esta solução. Nesses quase cinco meses de utilização a solução de virtualização não apresentou nenhum problema considerado de alto risco para o funcionamento dos PV's. Pelo contrário, a solução se mostrou estável e completamente funcional não apresentando, até o presente momento, nenhum tipo de anomalia.

A partir desta solução, considerada um caso de sucesso dentro da Caixa Econômica Federal, outra que segue a mesma idéia foi desenvolvida para melhorar o uso dos novos equipamentos adquiridos. Outro caso de sucesso é a solução para as estações de retaguarda de ponto de venda (RETPV) que possui duas máquinas virtuais instaladas em uma mesma máquina física. A idéia principal neste caso foi proporcionar melhorias como: facilidade de acesso a dois sistemas diferentes permitindo maior produtividade por parte do funcionário, diminuição de gastos com aquisição de novos equipamentos uma vez que apenas uma máquina física é capaz de executar ambos os sistemas legados. No caso da solução RETPV, uma das máquinas virtuais é composta pelo conjunto NT4 + SIAPV, apresentado neste trabalho, enquanto que a outra máquina virtualizada é composta de NT4 + Office 97 configurando uma estação de trabalho para realização de trabalhos diversos como acesso a internet, escrita de documentos, planilhas, apresentações eletrônicas, dentre outros. Antes da solução de virtualização, a Caixa Econômica Federal tinha de manter duas máquinas separadas para executar cada um dos sistemas legados.

Desta forma, atualmente, as soluções de virtualização correspondem com aproximadamente 13.000 (8.6%) máquinas de um total de 150.000 mil computadores, aproximadamente, presentes em toda a rede.

Capítulo 5

Conclusão

Este trabalho teve a finalidade de mostrar como os sistemas de virtualização existentes atualmente no mercado podem ajudar as empresas a resolverem problemas de compatibilidade entre *software* legado e *hardware* moderno. Além disso, o uso do sistema operacional GNU/Linux como sistema base para a solução mostra o seu grau de maturidade, bem como a visão das grandes empresas de tecnologia mundial com relação a este sistema.

Importante lembrar que esta solução de virtualização somente foi possível devido ao fato de utilizarmos um *software* de código aberto e de fácil customização. Sem estas características seria bastante difícil ter desenvolvido este trabalho em tão pouco tempo.

No que diz respeito ao tempo hábil, toda a solução desenvolvida por mim (Guilherme Veloso Neves Oliveira) e por meu colega Ronaldo Pacheco Wanzeller ficou pronta e operacional em 4 meses.

No que diz respeito ao impacto da nova solução para os PV's, podemos considerar que o mesmo foi desprezível, uma vez que se manteve toda a solução legada (NT4 + SIAPV) funcionando sem qualquer alteração. Este impacto pode ser analisado sob alguns pontos de vista. No que se refere a usabilidade por parte do usuário podemos considerar que o impacto foi zero, uma vez que a maneira com a qual o usuário interage com o sistema permaneceu idêntica a antiga solução. No que se refere a funcionalidade do ecossistema, algumas alterações se fizeram necessárias, porém todas de maneira transparente para o usuário final. Dentre as alterações necessárias podemos citar:

- Alteração do *range* dos servidores *DHCP*, uma vez que o modo de funcionamento utilizado pela rede da máquina virtual é o modo *"bridge"*. Ou seja, cada máquina necessita de dois endereços IP, sendo um para a máquina GNU/Linux e outro para o sistema virtualizado NT4.

- Instalação de servidores "*Landesk*" para permitir o inventariado das máquinas GNU/Linux.

Além disso deve-se atentar para os requisitos de *hardware* (processador, memória, periféricos) que compõem a máquina física a ser utilizada, uma vez que o processo de virtualização de software, principalmente o *full-virtualization* exige um processador com tecnologia apropriada.

A partir deste trabalho, outras soluções que envolvem processos de virtualização podem ser desenvolvidas. Este trabalho foi desenvolvido utilizando-se os *software* da VMware, pois naquela época o Xen ainda não oferecia suporte a virtualização completa. A idéia agora é que toda esta mesma solução seja feita utilizando-se o Xen como software de virtualização para permitir a execução do software legado NT4 + SIAPV e verificar a sua estabilidade para depois propor que os grandes ambientes virtualizados, os servidores das agências bancárias, que hoje utilizam o ESX 3.0, possam a vir utilizar o Xen como plataforma de virtualização.

Referências Bibliográficas

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEUBAUER, R.; PRATT, I.; WARFIELD, A. Xen and the Art of Virtualization. 2003. University of Cambridge Computer Laboratory.

CLARK, C.; FRASER, K.; HAND, S.; HANSEN, J. G.; JUL, E.; LIMPACH, C.; PRATT, I.; WARFIELD, A. Live Migration of Virtual Machines. 2005. - University of Cambridge Computer Laboratory - Departament of Computer Science University of Copenhagem, Denmark.

Equipe instalador Debian. Guia de Instalação de Debian GNU/Linux. *http://www.debian.org/releases/stable/i386/*, 2005. Acesso em: 15/02/2007.

GARFINKEL, T.; PFAFF, B.; CHOW, J.; ROSENBLUM, M.; BONEH, D. Terra: A Virtual Machine-Based Platform for Trusted Computing. 2003.

IBM. http://www-128.ibm.com/developerworks/linux/library/l-linuxvirt/. 2007. Acesso em: 20/03/2007.

NOVELL. Novell Virtualization on Suse Linux Enterprise Server - Virtualization Tecnology. p. 53, 7 ago. 2006.

SMITH, J. E.; NAIR, R. The Architecture of Virtual Machines. p. 32–38, maio 2005.

Standard Performance Evaluation Corporation. Spec INT 2000. *http://www.spec.org/cpu/CINT2000/*, 26 set. 2003. Acesso em: 15/04/2007.

SUGERMAN, J.; VENKITACHALAM, G.; LIM, B.-H. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. 30 jun. 2001.

The Xen Source. Volume Virtualization via the Next Generation of Server Virtualization. *Introducing the Xen Server Product Family: A Xen Source Withe Paper V12102006*, 12 out. 2006.

WIKIPEDIA - Hypervisor. http://en.wikipedia.org/wiki/Hypervisor. 19 fev. 2007. Acesso em: 20/02/2007.

WIKIPEDIA - VM_CMS. http://en.wikipedia.org/wiki/VM/CMS. 13 jan. 2007. Acesso em: 25/03/2007.

XEN. xen-lwe2005-short.ppt. *http://www.cl.cam.ac.uk/research/srg/netos/xen/architecture.html*, 2005. Acesso em: 15/04/2007.

Apêndice A

Procedimento de Criação de Máquinas Virtuais usando VMware Workstation

Este apêndice tem o objetivo de ilustrar o procedimento de criação de máquinas virtuais usando o software VMware Workstation 5.5.2.

O primeiro passo consiste selecionar a opção “New Virtual Machine”.

O segundo passo é apenas uma tela de boas vindas para a criação da nova máquina virtual como mostrado na Figura A.2

O terceiro passo consiste em definir que tipo de configuração o usuário irá fazer nas telas sub-sequentes a esta. A configuração tipica (que será mostrada neste exemplo) cria uma nova máquina virtual com os dispositivos de hardware mais comuns presentes atualmente. Já a configuração personalizada permite escolher quais dispositivos de hardware farão parte da máquina virtual. Lembrando que independente da escolha que se faça, ao final do processo é possível alterar todas as configurações definidas durante este processo inicial.

A etapa seguinte define o tipo de sistema operacional a ser virtualizado. A lista de SO’s suportados é extensa como pode-se observar na Figura A.4.

O próximo passo define onde serão armazenados os arquivos da máquina virtual como mostra a Figura A.5

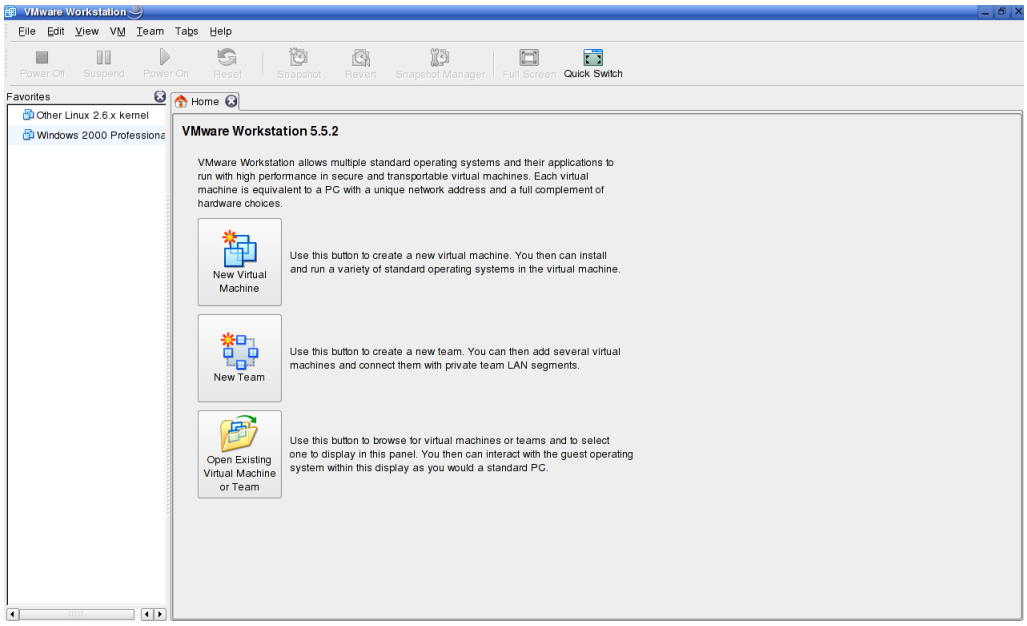

Figura A.1: Vmware Workstation - Tela inicial

O passo seguinte, Figura A.6, define o funcionamento da rede para a máquina virtual. Neste caso alguns pontos devem ser observados, uma vez que eles impactam diretamente no funcionamento da rede.

O modo de funcionamento “bridge” garante a máquina virtual um IP exclusivo e válido, diferente do IP atribuído à máquina hospedeira. Além disso, a placa de rede da máquina hospedeira é ativada para funcionar em modo promíscuo, pois é a única forma que existe atualmente para que os pacotes da rede cheguem até a máquina virtual.

Já o modo de funcionamento “NAT” garante a máquina virtualizada um IP reservado, caso nenhum seja especificado. Além disso, o controle sob este funcionamento fica a cargo da interface vmnet* criada com o objetivo de servir a máquina virtualizada. Neste modo de funcionamento, a interface vmnet* funciona mo servidor DHCP e roteador para a máquina virtual

O modo de funcionamento “host-only” isola a máquina virtual em uma rede privada criada na máquina hospedeira. É um modo util pra quando se deseja criar um ambiente virtualizado isolado da sua rede local.

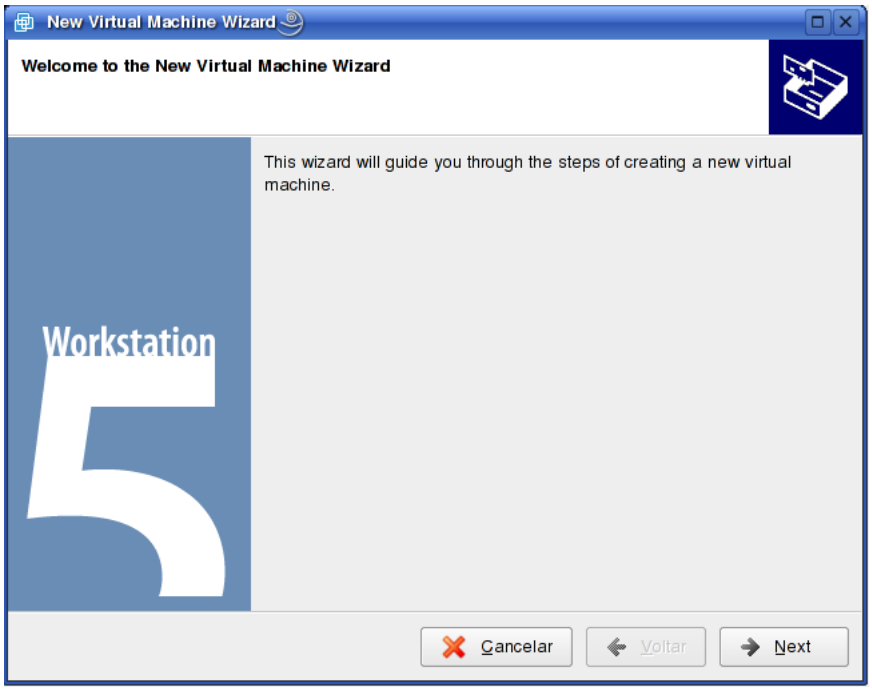

Figura A.2: Primeira tela para criação da Máquina Virtual

O último passo é a definição do tamanho do disco rígido da máquina virtual. Apesar desta configuração poder ser alterada no futuro, um ponto importante é que a opção “Allocate all disk space now” esteja desmarcada de modo que os arquivos gerados pela máquina virtual ocupem apenas o espaço inicial necessário. A medida que as informações forem sendo armazenadas no disco virtual, o arquivo correspondente irá aumentar o tamanho necessário até o limite pré-definido. Já a opção “split disk into 2GB files” garante que o(s) arquivo(s) que irá(ão) compor o disco rígido virtual não ultrapassem o tamanho real de 2.0 gygabytes.

A Figura A.8 exibe o console de administração do VMware Workstation que é utilizado para gerenciar as máquinas virtuais.

Uma vez criado o hardware virtualizado, é necessário que se adicione o CD-ROM do sistema operacional a ser instalado no driver correspondente e ligue a máquina virtual. Deste ponto em diante o procedimento de instalação deve seguir o manual do SO em questão.

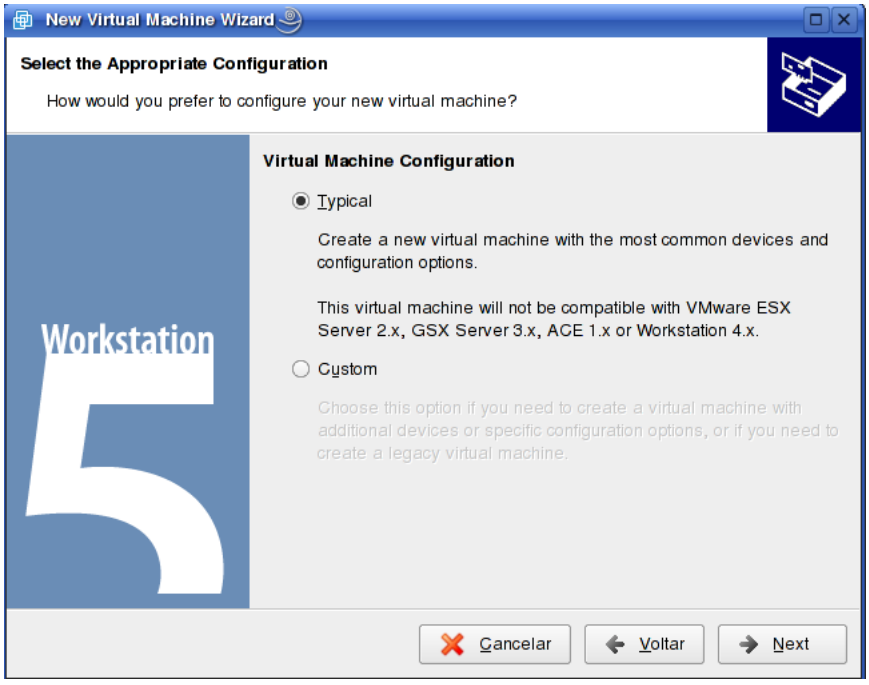

Figura A.3: Procedimento de Customização

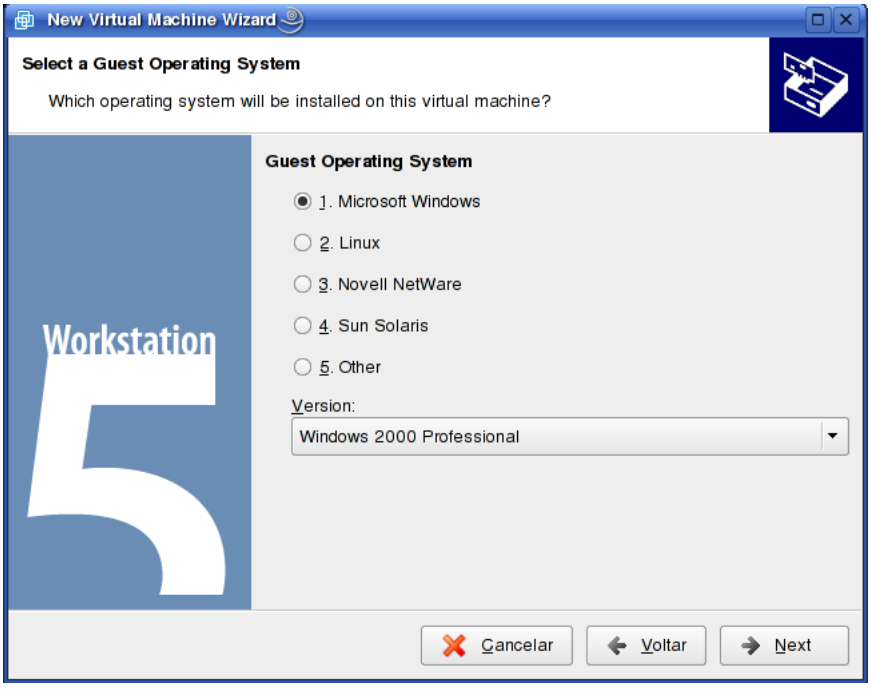

Figura A.4: Seleção do SO Virtualizado

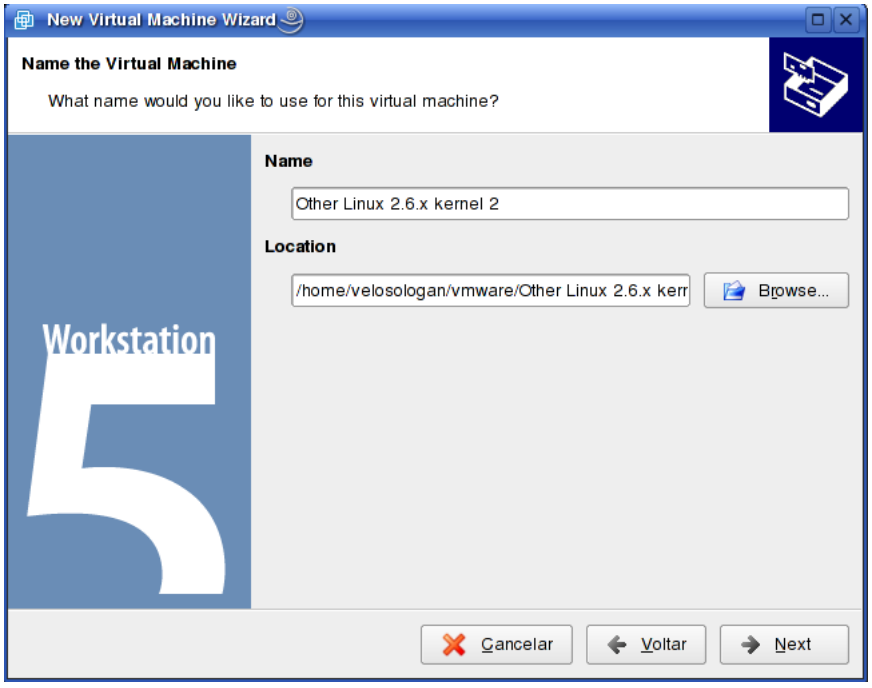

Figura A.5: Local Armazenamento

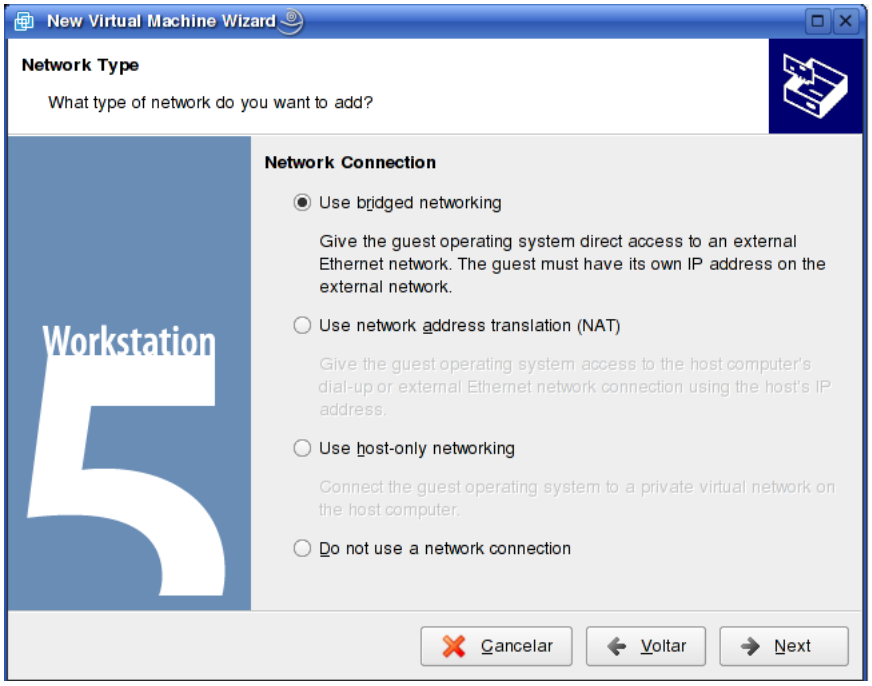

Figura A.6: Tipos de configuração da rede

Figura A.7: Tamanho do disco rígido

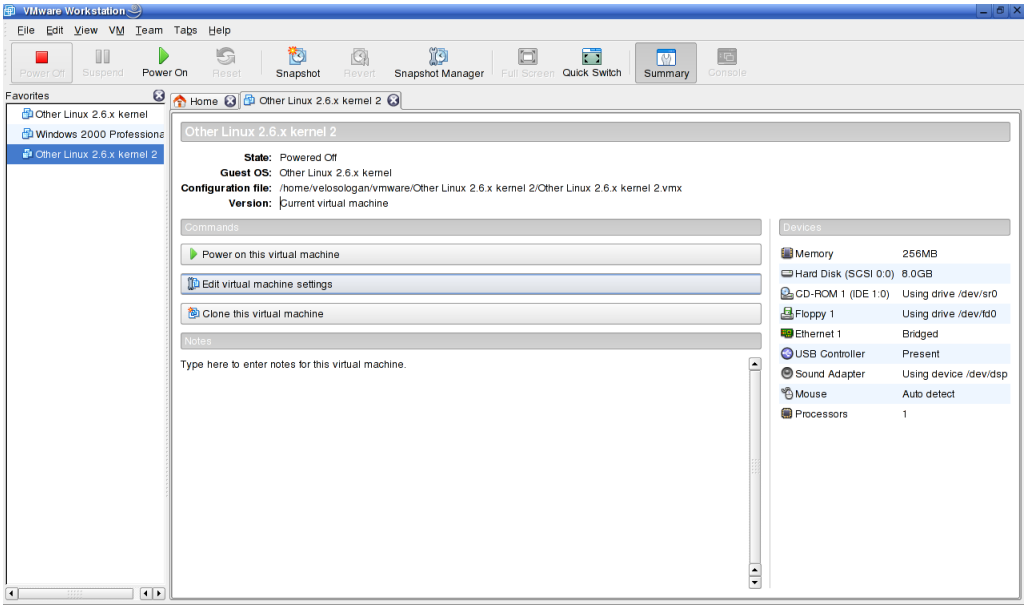

Figura A.8: Console de Administração das Máquinas Virtuais

Apêndice B

Código Fonte de Alguns *Scripts* utilizados

B.1 *Script* de Instalação - "script_instalacao.sh"

```
#!/bin/bash

DIR_INSTALL=$PWD
IMAGEM_TGZ=/home/caixa/imagem_linux_debian.tar.gz
PONTO_MONTAGEM=/mnt/destino
PARTICAO_ROOT=1
HABILITAR_PARTICAO_SWAP=1
PARTICAO_SWAP=2
TAMANHO_SWAP=1024 #MB
SIST_ARQUIVO=ext3
DISCO=sda;
DISCO_DESTINO=/dev/$DISCO

#APAGANDO A TABELA DE PARTIÇAO DO $DISCO_DESTINO
sudo dmesg -n1
dialog --title ' CAIXA ECONOMICA FEDERAL ' --infobox \
    " \n Apagando tabela de particoes do disco rigido " 5 55
sudo dd if=/dev/zero of=$DISCO_DESTINO bs=1024 count=1024

#CRIANDO TABELA DE PARTIÇOES VAZIA
sleep 1
dialog --title ' CAIXA ECONOMICA FEDERAL ' --infobox \
   " \n Criando a tabela de particoes do disco rigido " 5 55
sudo parted --script $DISCO_DESTINO mklabel msdos
```

```
#RECUPERANDO O TAMANHO MAXIMO POSSIVEL DO $DISCO_DESTINO
TAMANHO_MAXIMO=$(sudo parted --script $DISCO_DESTINO print|grep \
      geometry|cut -f 2 -d-|cut -f 1 -d.)

#CRIANDO PARTICAO NO $DISCO_DESTINO
if [ ${HABILITAR_PARTICAO_SWAP} -eq 1 ]; then

   POSICAO_SWAP=`expr ${TAMANHO_MAXIMO} - ${TAMANHO_SWAP}`
   sudo parted --script $DISCO_DESTINO mkpart primary  \
      0 $POSICAO_SWAP #cria root

   sudo parted --script $DISCO_DESTINO mkpartfs primary \
      linux-swap $POSICAO_SWAP $TAMANHO_MAXIMO #cria swap

else

   sudo parted --script $DISCO_DESTINO mkpart primary \
      0 $TAMANHO_MAXIMO #cria root

fi;

sudo parted --script $DISCO_DESTINO set $PARTICAO_ROOT boot on

#Formatando a particao como ext3
if [ ${HABILITAR_PARTICAO_SWAP} -eq 1 ]; then

   sudo mkswap -v1 ${DISCO_DESTINO}${PARTICAO_SWAP} || shellout
   sudo swapon ${DISCO_DESTINO}${PARTICAO_SWAP} || shellout

fi;

dialog --title ' CAIXA ECONOMICA FEDERAL ' --infobox \
     " \n Formatando particoes do disco rigido, aguarde! " 5 55

# Formata com o sistema de arquivos especificado
sudo /sbin/mkfs.ext3 -q ${DISCO_DESTINO}${PARTICAO_ROOT} &> \
               /dev/null

#MONTANDO A PARTICAO
dialog --title ' CAIXA ECONOMICA FEDERAL ' --infobox \
         " \n Montando disco rigido, aguarde! " 5 55

sudo mount ${DISCO_DESTINO}${PARTICAO_ROOT} ${PONTO_MONTAGEM} &> \
               /dev/null

#COPIANDO ARQUIVOS DA IMAGEM PARA A PARTICAO
cd $PONTO_MONTAGEM
sudo tar zxf ${IMAGEM_TGZ} &> /dev/null &
```

```
sleep 1
pidtar=$(pidof tar)
Tam=0
Tmax=652344

while [ -n "$pidtar" ]
do
   sleep 1;
   Tam=$(sudo du -s $PONTO_MONTAGEM | cut -f1);
   percent=$(($Tam*100/$Tmax));
   pidtar=$(pidof tar)
   echo $percent

done |   dialog --gauge " CAIXA ECONOMICA FEDERAL \n\n\n   \
   Instalando imagem linux no disco rigido. " 10 60

echo 100 | dialog --gauge "CAIXA ECONOMICA FEDERAL \n\n\n  \
    Instalando imagem linux no disco rigido. " 10 60

sleep 1
cd

#CONFIGURAÇAO DE INSTALAÇAO
dialog --title ' CAIXA ECONOMICA FEDERAL ' --infobox \
   " \n Finalizando instalacao da imagem no disco rigido, \
            aguarde. " 5 80

cd $PONTO_MONTAGEM;
sudo cp ${DIR_INSTALL}/device.map.template boot/grub/device.map
#sudo cp ${DIR_INSTALL}/menu.lst.template boot/grub/menu.lst
sudo cp ${DIR_INSTALL}/mtab.template etc/mtab
sudo cp ${DIR_INSTALL}/fstab.template etc/fstab
sudo sed -i -e "s#DISCO_DESTINO#$DISCO_DESTINO#g" \
      $PONTO_MONTAGEM/boot/grub/device.map;

sudo sed -i "s#/dev/.* ro#${DISCO_DESTINO}${PARTICAO_ROOT}\
       ro#" $PONTO_MONTAGEM/boot/grub/menu.lst;

## ALTERA O FSTAB
echo "${DISCO_DESTINO}${PARTICAO_ROOT} / ${SIST_ARQUIVO}\
           defaults        0       1" > /tmp/fstab;

if [ ${HABILITAR_PARTICAO_SWAP} -eq 1 ]; then
   echo "${DISCO_DESTINO}${PARTICAO_SWAP} none swap sw \
   0       0" >> /tmp/fstab;
fi;

sudo chmod a+w ${PONTO_MONTAGEM}/etc/fstab;
```

```
sudo cat /tmp/fstab >> ${PONTO_MONTAGEM}/etc/fstab;
sudo chmod 644 ${PONTO_MONTAGEM}/etc/fstab;

## ALTERA O MTAB
sudo chmod a+w ${PONTO_MONTAGEM}/etc/mtab;
sudo echo "${DISCO_DESTINO}${PARTICAO_ROOT}  /  ${SIST_ARQUIVO} \
   rw,errors=remount-ro 0 0" >> ${PONTO_MONTAGEM}/etc/mtab;

sudo chmod 644 ${PONTO_MONTAGEM}/etc/mtab;
cd
cd $PONTO_MONTAGEM;

#Entrando na imagem e fazendo o a instalacao do grub
cd
sudo chroot ${PONTO_MONTAGEM} grub-install --no-floppy \
         ${DISCO_DESTINO}

cd

#Alterando inittab para primeiro boot
sudo sed -i "s|rc 2|inicializa|" ${PONTO_MONTAGEM}/etc/inittab

#Desmontando a particao
sudo umount ${PONTO_MONTAGEM}

#Reiniciando
sudo clear
sudo mount /dev/sda1 /mnt 2> /dev/null
sudo mount -o bind /proc /mnt/proc 2> /dev/null
sudo chroot /mnt /home/caixa/ejecta 2> /dev/null
```

B.2 *Script* Instalação do Sistema Virtualizado "install_win_nt.sh"

```
#!/bin/bash
DIRVM=$(ifconfig eth0 | grep HWaddr | cut -d " " -f11 | tr -d :)
[ -z $DIRVM ] && DIRVM=$(ifconfig eth0 | grep HW | cut -d " " \
            -f16 | tr -d :)

LOCAL="/home/caixa/vmware/$DIRVM"
HOME="/home/caixa/vmware"
ARQUIVO1="windows_nt_et.tar.gzaa"
ARQUIVO2="windows_nt_et.tar.gzab"
ARQUIVO_FINAL="windows_nt_et.tar.gz"
PTMONT="/mnt"

echo '#!/bin/bash' > /usr/X11R6/bin/inicia_vm
echo aumix -v -100 >> /usr/X11R6/bin/inicia_vm
```

```
echo aumix -c -100 >> /usr/X11R6/bin/inicia_vm
echo aumix -v +70 >> /usr/X11R6/bin/inicia_vm
echo aumix -c +70 >> /usr/X11R6/bin/inicia_vm
echo rm -f $LOCAL/Windows_NT_ET/*.vmss >> /usr/X11R6/bin/inicia_vm
echo rm -f $LOCAL/Windows_NT_ET/*.WRITELOCK >> \
         /usr/X11R6/bin/inicia_vm
echo vmplayer $LOCAL/Windows_NT_ET/Windows_NT_ET.vmx >> \
         /usr/X11R6/bin/inicia_vm

if [ ! -e $LOCAL/Windows_NT_ET/Windows_NT_ET.vmx ]; then

   clear
   eject
   dialog --stdout --msgbox "CAIXA ECONÔMICA FEDERAL \n\n \
       Insira o CD com o arquivo: \n\n   $ARQUIVO1 " 10 35

   eject -t
   sleep 1
   mount -t iso9660 /dev/hda $PTMONT 2> /dev/null
   mkdir -p $LOCAL
   TARQ1=$(ls $PTMONT|cut -f1 -d " " |head -n 1)
   TARQ=$(ls $PTMONT|grep $ARQUIVO1)
   while [ "$TARQ" != "$ARQUIVO1" ]
         do
      eject
      dialog --stdout --msgbox "CAIXA ECONÔMICA FEDERAL \n \
         Arquivo esperado: $ARQUIVO1 \n Diferente do  \
         arquivo atual do CD: $TARQ1.\n Coloque o CDROM \
         que contenha o arquivo correto e pressione Enter." 18 85

      eject -t
                  sleep 1
      hdparm -E 40 /dev/hda &> /dev/null
      mount -t iso9660 /dev/hda $PTMONT > /dev/null 2>&1
      sleep 2
      TARQ=$(ls $PTMONT|grep $ARQUIVO1)

   done

   Tam=0
   Tmax=$(ls -s $PTMONT/$ARQUIVO1 | cut -f1 -d " ")
   cp $PTMONT/$ARQUIVO1 $HOME  &

   for ((i=0; i<$Tmax; i=$Tam))
   do
      sleep 1;
      Tam=$(ls -s $HOME/$ARQUIVO1 | cut -f1 -d " ");
      percent=$(($Tam*100/$Tmax));
```

```
      echo $percent
   done |   dialog --gauge "CAIXA ECONÔMICA FEDERAL \n\n\n  \
          Copiando arquivo, aguarde. " 10 60

   Tam=0
   Tamarq1=$(ls -s $HOME/$ARQUIVO1 | cut -f1 -d " ")
   Tamarq2=$(ls -s $HOME/$ARQUIVO2 | cut -f1 -d " ")
   Tmax=$(($Tamarq1+$Tamarq2-8))
   cat $HOME/$ARQUIVO1 $HOME/$ARQUIVO2 > $HOME/$ARQUIVO_FINAL &

   for ((i=0; i<$Tmax; i=$Tam))
   do
      sleep 1;
      Tam=$(ls -s $HOME/$ARQUIVO_FINAL | cut -f1 -d " ");
      percent=$(($Tam*100/$Tmax));
   echo $percent
   done |   dialog --gauge "CAIXA ECONÔMICA FEDERAL \n\n\n   \
         Preparando instalação, aguarde. " 10 60

   Tam=0
   Tmax=1930208

   tar -mxzf $HOME/$ARQUIVO_FINAL -C $LOCAL 1> /dev/null &
   sleep 1

   while [ $(pidof tar) ]
   do
      sleep 1;
      Tam=$(ls -s $LOCAL/Windows_NT_ET | grep total | cut -f2 -d " ");
      percent=$(($Tam*100/$Tmax));
      echo $percent

   done |   dialog --gauge "CAIXA ECONÔMICA FEDERAL \n\n\n \
          Instalando Windows NT na Máquina virtual. " 10 60

   echo 100 | dialog --gauge "CAIXA ECONÔMICA FEDERAL \n\n\n  \
          Instalando Windows NT na Máquina virtual. " 10 60
   chown -R caixa:caixa $LOCAL
   eject  &

   dialog --msgbox " CAIXA ECONÔMICA FEDERAL \n\n\n Instalação finalizada \
      com sucesso.\n\nRemova o CD e pressione Enter para continuar. " 10 60

   eject -t
fi
```

B.3 *Script* para Renomear o Sistema Hospedeiro GNU/-Linux “renomeia.sh”

```
#!/bin/bash

ARQ_TEMPORARIO="/tmp/aux.txt";
INTERFACE="eth0";
TIMEZONE_TEMP="/tmp/timezone.teste";

rm -rf ${ARQ_TEMPORARIO};
touch ${ARQ_TEMPORARIO};
touch ${TIMEZONE_TEMP};

confirma=0;

while [ $confirma -eq 0 ]; do
  # --- Pergunta qual o nome da máquina ---

    nome=$(dialog --stdout --title ' CAIXA ECONOMICA FEDERAL '\
    --inputbox "\n Digite o nome da máquina: " 0 0);


    nome=`echo $nome | tr [:lower:] [:upper:]`;
    estado_inf=`echo $nome | cut -c1-2`;
    cpf_inf=`echo $nome | cut -c3-6`;
    eh_inf=`echo $nome | cut -c7-8`;
    sequencial_inf=`echo $nome | cut -c9-11`;
    estado=`echo $nome | cut -c1-2 | grep '[A-Z][A-Z]'`;
    cpf=`echo $nome | cut -c3-6 | grep '[0-9][0-9][0-9][0-9]'`;
    eh=`echo $nome | cut -c7-8 | grep 'EH'`;
    sequencial=`echo $nome | cut -c9-11 | egrep '[0-9]{2}'`;

  # -- Valida se o estado da federacao eh letra, cpf eh numero, \
  #      a string "EH" e os dois ultimos digitos

    if test ! -z $estado
    then
      if test ! -z $cpf
   then
       if test ! -z $eh
       then
         if test ! -z $sequencial
      then
        # -- Pede confirmacao --

        confere=$(dialog --stdout --title ' CAIXA ECONOMICA FEDERAL ' \
         --yesno "\n As informações estão corretas ? \n\n \
```

```
            Nome da Máquina -> $nome" 9 40; echo $?);
      else

         dialog --stdout --msgbox " Valor Informado -> $sequencial_inf \
         \n\n Os últimos três números informados não são válidos! \
         \n\n Apenas informe uma sequência de 3 algarismos!" 10 60;

      fi
       else
         dialog --stdout --msgbox " Valor Informado -> $eh_inf \n\n \
         Vc deve obrigatoriamente informar a sequência de letras \
          \"EH\" após o CGC da Unidade" 10 80;

       fi
   else
         dialog --stdout --msgbox " Valor Informado -> $cpf_inf \n\n \
          O identificador do CGC da Unidade não é válido! \n\n  \
         Informe uma sequência de 4 algarismos válidos!" 10 60;

   fi
     else
         dialog --stdout --msgbox " Valor Informado -> $estado_inf \n \
         \n O estado da federação informado não é válido! \n\n  \
         Informe uma sequência de 2 letras válidas! \n\n Por exemplo: \
         \n\n Distrito Federal - DF \n\n Sao Paulo - SP \n\n \
         Rio de janeiro - RJ" 20 60;
     fi

  case $confere in
      0) echo $nome > ${ARQ_TEMPORARIO};
        cp ${ARQ_TEMPORARIO} /etc/hostname;
        sed -i "/\b[A-Z][A-Z][0-9].*\b/s/\b[A-Z][A-Z][0-9].*\b/${nome}/" \
                   /etc/hosts;

           confirma=1;
        hostname $nome
           ;;
  esac;

done;

echo "America/Sao_Paulo" > ${TIMEZONE_TEMP};

cp -f ${TIMEZONE_TEMP} /etc/timezone;

clear;
```

B.4 *Script* de Inventariado da Máquina "inservlandesk.sh"

```
#!/bin/bash
trap " " 1 2 3 15
loop1=1
clear

while [ $loop1 != 0 ]; do
IPLANDESK=$( dialog \
   --stdout \
   --menu "CAIXA ECONÔMICA FEDERAL\n\n Escolha o IP do Servidor \
       LanDesk da sua região. \n Ou - Entrar - para entrada \
       manual do IP.\n\n Use as setas do teclado p/ \
      selecionar sua opção. " \
   0 0 0 \
   Entrar       ' - Servidor - Não listado ' \
   x.x.x.x       ' - Servidor - MG ' \
   x.x.x.x       ' - Servidor - DF ' \
   x.x.x.x      ' - Servidor - SP ' \
   x.x.x.x      ' - Servidor - MS ' \
   x.x.x.x      ' - Servidor - SP ' \
   x.x.x.x      ' - Servidor - PR ' \
   x.x.x.x      ' - Servidor - ES ' )

   if [ $? != 0 ]; then
      loop1=1
   else
      while [ "$IPLANDESK" = "Entrar" ]; do
         IPLANDESK=$( dialog \
         --stdout --inputbox " Digite o IP do \n
         Servidor LanDesk : " 0 0 );
         echo "$IPLANDESK" > /tmp/$$;

         IPLANDESK=$(egrep "10\.((1[0-9]|[1-9]?)[0-9]| \
         2([0-4][0-9]|5[0-5]))\.((1[0-9]|[1-9]?)[0-9]|2( \
         [0-4][0-9]|5[0-5]))\.((1[0-9]|[1-9]?)[1-9]|2( \
         [0-4][0-9]|5[0-4]))$" /tmp/$$)

         A=$(dialog --stdout --yesno "IP informado -> $IPLANDESK")
      done

   [ -z "$IPLANDESK" ] && loop1=1 && dialog --msgbox " IP inválido ! \
      ou cancelado \n Tente novamente." 0 0 || loop1=$(dialog \
      --stdout --yesno "Confirma IP ? \n\n $IPLANDESK" 7 30 ; echo $?)
   fi
done
echo "#!/bin/bash" > /usr/bin/landesk.sh;
echo "ldiscnux -ntt=$IPLANDESK -f" >> /usr/bin/landesk.sh;
```

B.5 *Script* de Agendamento de Inventário "hora.sh"

```
#!/bin/bash

CURENTH=$(date +%H)
HOR=$(expr $CURENTH + 1)

if [ $HOR -ge 19 ]; then

   /usr/bin/landesk &

else
   if [ $HOR -le 7 ]; then
      HOR=8
   fi

   TIMERUN=$(/usr/bin/randomic -bt $HOR 19)
   TIMEMIN=$(/usr/bin/randomic -bt 1 59)

   echo "SHELL=/bin/sh" > /etc/crontab
   echo "PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin \
         :/usr/bin">>/etc/crontab

   echo "17 *  * * * root    run-parts --report \
         /etc/cron.hourly" >> /etc/crontab

   echo "25 6  * * * root    test -x /usr/sbin/anacron || \
         run-parts --report /etc/cron.daily" >> /etc/crontab

   echo "47 6  * * 7 root    test -x /usr/sbin/anacron || \
         run-parts --report /etc/cron.weekly" >> /etc/crontab

   echo "52 6  1 * * root    test -x /usr/sbin/anacron || \
         run-parts --report /etc/cron.monthly" >> /etc/crontab

   echo "$TIMEMIN $TIMERUN * * *  root  /usr/bin/landesk.sh" >> \
                     /etc/crontab

   invoke-rc.d cron restart 1> /dev/null
fi
```