

Para-virtualização no Linux com o Xen 3

Virtual, veloz e leve

O sistema de virtualização Xen está rapidamente tornando-se um recurso padrão no Linux corporativo. Vamos conferir os recursos da última versão do Xen.

por Kurt Garloff

A aguardada versão 3 do Xen finalmente foi liberada no ano passado. O Xen é patrocinado pela Universidade de Cambridge, Inglaterra, e vem se tornando uma das principais soluções de virtualização para Linux. Personagens globais do mundo do Linux, incluindo Novell, Red Hat, HP, IBM, Intel e AMD, já mostraram seu apoio ao Xen. Graças a seu excelente

desempenho e baixo custo, o monitor de máquinas virtuais Xen talvez seja o catalisador que estávamos esperando para melhorar a virtualização no Linux.

A recente evolução da computação virtual gira em torno do problema de fornecer um ambiente virtualizado com a menor exigência possível do sistema. A chamada *virtualização completa* envolve

o mapeamento de chips de computador, como o processador e o controlador do teclado, em software. A virtualização completa torna a máquina virtual independente de sua infra-estrutura física, e portanto é muito flexível. Porém, a estrutura necessária para mantê-la acrescenta uma complexidade que afeta significativamente o desempenho.

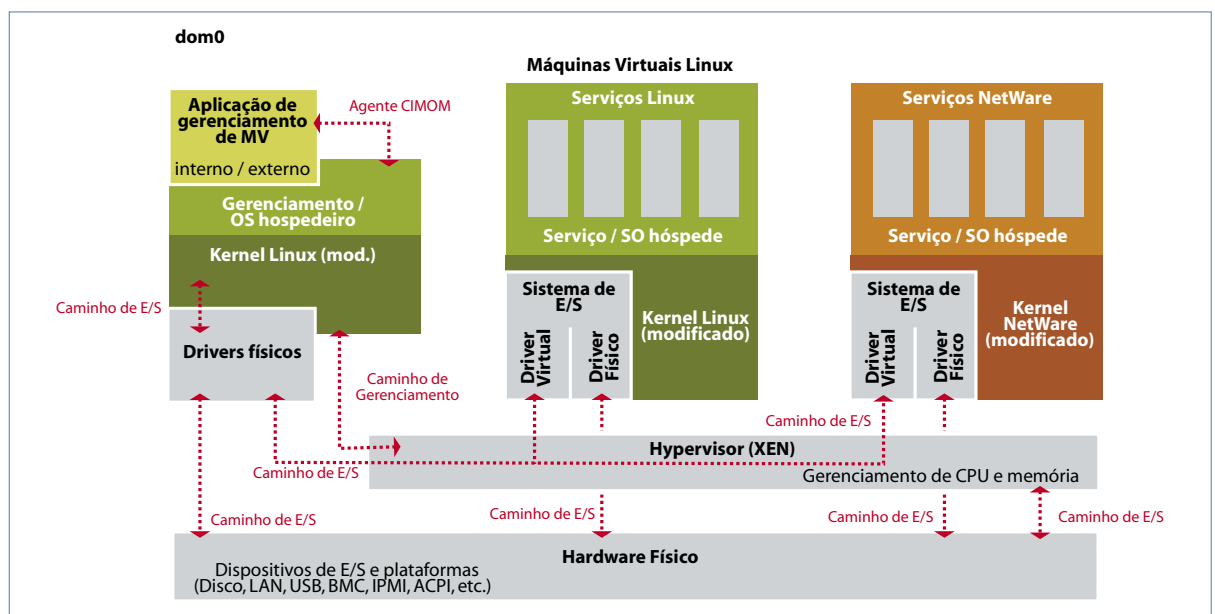


Figura 1 O hypervisor do Xen virtualiza a CPU e a memória. Enquanto o domínio privilegiado 0 pode acessar o hardware, todos os outros domínios são restritos aos discos e à rede virtuais.

O objetivo da virtualização completa é emular a CPU, que interpreta todas as instruções de máquina. Ainda que se emulem somente máquinas com a mesma arquitetura de CPU – em outras palavras, mesmo que o código de máquina possa rodar nativamente –, o emulador ainda precisará interceptar qualquer tentativa de acesso ao hardware para convencer o sistema operacional da máquina virtual de que ele dispõe de controle total e irrestrito. Os emuladores que controlam seus hóspedes dessa forma geralmente são chamados de *hypervisors*.

Uma variante desse modelo chama-se *para-virtualização*. A para-virtualização permite que máquinas virtuais específicas comuniquem-se diretamente com o hardware. Em vez de todas as máquinas virtuais comunicarem-se com o hospedeiro, a máquina privilegiada gerencia a interação e recebe as chamadas passadas pelos outros sistemas virtuais.

O Xen é um sistema popular de virtualização que usa o modelo de para-virtualização. Apesar de as versões 1.x utilizarem seus próprios drivers de dispositivos, a versão 2, do segundo semestre de 2004, concentrou-se em manter o hypervisor com o menor tamanho e maior segurança possíveis. O acesso ao hardware fica restrito a domínios privilegiados. O primeiro domínio a ser iniciado, o domínio 0, possui privilégios estendidos. Mais especificamente, ele pode configurar, iniciar e parar outras máquinas virtuais. O domínio 0 é iniciado imediatamente após o hypervisor, que não consegue fazer muita coisa sem ele. Diferentemente do domínio 0, ou *domo*, as máquinas sem privilégios são referidas como *domU*, de *unprivileged* (sem privilégios) (figura 1).

O Xen demonstra uma impressionante mudança de velocidade. Embora a maioria das soluções de virtualização pareçam convincentes quando se pede que realizem tarefas computacionalmente exigentes, a verdadeira qualidade de uma máquina virtual é revelada pela forma como ela lida com aplicações que fazem uso intensivo de chamadas de sistema, normalmente de entrada e saída no disco rígido ou na rede. O Xen obtém as melhores notas nesse quesito. O aumento de esforço para compilar um kernel foi de apenas 2%, e ainda ficou abaixo de 10% no *Webbench* por gigabit ethernet. Seria necessário diminuir a MTU para ver

Quadro 1: Novos recursos do Xen 3

Alguns dos novos recursos incluídos no Xen 3 são:

- ▶ **Suporte a máquinas SMP virtuais.** Apesar de o Xen já oferecer suporte a múltiplos processadores há algum tempo, os domínios antes ficavam restritos a uma única CPU. Na versão 3, isso mudou. É possível até mudar o número de CPUs virtuais em tempo de execução (supondo que o sistema operacional suporte isso, como o Linux).
- ▶ **Suporte a ACPI.** No Xen 2, o hypervisor oferecia um suporte bastante rudimentar a ACPI, que bastava para lidar com a maioria das tabelas de roteamento de IRQ. No Xen 3, o dom0 tem acesso à maioria das funcionalidades ACPI.
- ▶ **Melhor suporte a hardware.** Alguns problemas, mais especificamente com o suporte a AGP e DRM (gráficos 3D) no Linux foram eliminados no último 2.0.x e nas versões de desenvolvimento do 3.0.
- ▶ **Suporte a PAE36.** O Xen 2 ficava restrito ao espaço de endereçamento normal de 32 bits (4 GB) em máquinas x86. A extensão PAE36 agora permite acessar até 64 GB; obviamente, supondo que tanto o Xen quanto o kernel tenham sido compilados com o suporte a essa extensão.
- ▶ **Suporte a x86-64.** A variante de 64 bits da arquitetura x86 elimina todas as restrições associadas ao espaço de endereçamento de 32 bits, e acrescenta diversas otimizações. O Xen 3 agora suporta sistemas operacionais de 64 bits.
- ▶ **Mais arquiteturas.** O suporte a IA64 (Itanium) foi integrado ao Xen 3, e o suporte à arquitetura Power vem em breve.
- ▶ **Uso de novas tecnologias de processadores.** Há uma nova geração de CPUs, com extensões especiais chamadas *Vanderpool* (Intel) e *Pacífica* (AMD). Essas extensões facilitam e aceleram a implementação da virtualização completa, o que permite que o Xen 3 suporte sistemas operacionais não para-virtualizados.

o Xen com gigabit começar a cansar. De acordo com o projeto Xen, ele facilmente supera o VMware e o User-Mode Linux [1].

As versões 2 e posteriores possuem a capacidade de migrar máquinas virtuais (exceto pelo domo) de uma máquina física para outra. Quando uma máquina virtual é migrada, seu conteúdo de memória é transferido pela rede – sem que seja preciso desligar a máquina, exceto quando o processo termina, para copiar as últimas páginas de memória que foram alteradas. Os desenvolvedores do Xen já demonstraram esse recurso com servidores web em uso intenso, e até com *streaming* de vídeo. As interrupções típicas são de uma magnitude aproximada de um décimo de segundo.

O lançamento da versão 3 do Xen, em dezembro de 2005, foi um grande marco. Além dos vários novos recursos da versão 3 (veja o quadro 1), a interface entre os sistemas operacionais para-virtualizados e o hypervisor foi congelada para garantir que, no futuro, todo o trabalho dedicado à para-virtualização do kernel de um sistema operacional precise ser feito apenas uma vez. Isso significa que podemos esperar que mais sistemas operacionais sejam portados para a arquitetura virtual do Xen 3 num futuro próximo, incluindo o Netware, por exemplo.

Conhecendo o Xen

A abordagem mais fácil para conhecer o Xen é converter uma instalação Linux existente em um domo do Xen, para notar a diferença entre o domo e um sistema normal, e resolver eventuais problemas. O segundo passo seria instalar mais domínios para realizar outras tarefas, ou migrar serviços que antes rodavam no domo para seus próprios domínios.

O primeiro passo do processo de conversão é instalar o hypervisor e as ferramentas que o acompanham. O procedimento exato depende da sua distribuição. Os pacotes para o Suse Linux 10.0 chamam-se *xen*, *kernel-xen* e *xen-tools*, por exemplo. Talvez seja preciso instalar mais alguns pacotes do Xen. Por exemplo, o Suse armazena a documentação num pacote separado. E definitivamente recomenda-se lê-la.

Quadro 2: Saída do comando xm

```
root@tpkurt:~ [0]# xm list
Name           ID Mem(MiB) VCPUs State  Time(s)
Domain-0      0   364      1  r----- 421.3
iSCSI0        1   128      2  ----- 111.5
```

Além do domínio 0, pode-se ver que uma segunda máquina virtual, chamada *iSCSI0*, está rodando. Ela acessa 128 MB de RAM, junto com duas CPUs virtuais. A máquina virtual consumiu 111,5 s de ciclos de CPU até o momento.

Se os pacotes forem anteriores a dezembro de 2005, você certamente precisará fazer atualizações. Na corrida para o lançamento oficial da versão 3.0, várias falhas foram corrigidas, aumentando muito a estabilidade do Xen.

Após instalar os pacotes, é bom adicionar uma entrada do carregador de inicialização para o Xen. Isso permite um sistema com *dual boot*, onde se pode escolher entrar na máquina com o kernel nativo ou carregar o hypervisor do Xen com o kernel para-virtualizado no domo. A entrada do GRUB deve ser mais ou menos assim:

```
title XEN
    root (hdX,Y)
    kernel /xen.gz PARAMS_XEN
    module /vmlinuz-xen PARAMS_KERNEL
    module /initrd-xen
```

Nesse exemplo, `hdX,Y` é a partição com o diretório `/boot`, e `PARAMS_KERNEL` representa os parâmetros normais do kernel que você passaria para o kernel nativo. O Xen também aceita parâmetros: por exemplo, é possível especificar `dom0_mem=XXX` para informar o tamanho da memória para o domo (em KB). Se você não especificar a alocação de memória,

ela será toda conferida ao domo. `lapic` permite que se ative um controlador de interrupção APIC específico, caso a BIOS o tenha desabilitado; isso pode resolver problemas de interrupção.

Isso termina a contagem regressiva: agora, o computador pode carregar o hypervisor e o domo ao invés do kernel nativo. Para ser sincero, esse evento é decepcionantemente comum: o processo de inicialização parece totalmente normal, e o sistema Linux parece inalterado à primeira vista. Mas olhando mais de perto, vê-se que o hypervisor mostra algumas mensagens de inicialização. Se o seu distribuidor ativar uma tela de *bootsplash*, a tela será desabilitada e você será apresentado com um ambiente de modo texto em VGA.

O motivo do modo texto é que o *framebuffer* VESA não funciona no Xen. Como o hypervisor inicia o kernel no modo protegido da CPU, o kernel é incapaz de chamar as rotinas do modo real da BIOS VGA para configurar a tela.

Existem alguns outros drivers de hardware que não funcionam no Xen. O motivo geralmente é um mau mapeamento de endereços físicos para endereços de máquina. Em alguns casos, o driver, e não o hypervisor, lida com isso. Os drivers de dispositivos programados para ser portáteis não devem

causar problemas, mas ainda há uns poucos drivers que não funcionarão no Xen, incluindo aqueles para várias placas wireless, assim como os drivers 3D proprietários da Nvidia e ATI – os drivers livres funcionam. Além disso, pode-se esperar problemas com placas ISA que usam transferência DMA.

Se você tiver problemas com algum driver de hardware, na melhor das hipóteses o driver não funcionará – na pior, o kernel do domo travará. Se isso acontecer, a virtualização não poderá oferecer nenhuma ajuda, já que o hypervisor é inútil sem um kernel no domo. Na verdade, o hypervisor nem tem seus próprios drivers de teclado e monitor necessários para a depuração da instalação.

Se o seu sistema travar, o melhor a fazer é desabilitar o driver responsável – normalmente, o último driver a mostrar uma saída na tela. Os suspeitos normais são os drivers 3D para o X11, e a forma mais fácil de remediar isso é usar a opção 3 na linha de comando para iniciar o Linux em modo texto. Lembre-se de passar essa opção ao kernel, e não ao hypervisor. Para isso, deve-se sair do menu gráfico no Xen, digitar **[e]** para modificar a linha de comando do kernel, e então pressionar **[b]** para iniciar.

Se isso tudo funcionar, você agora já deve ter um sistema Linux rodando no domo, e não deve ser muito diferente do sistema nativo. Observando mais atentamente, você verá que agora tem 50 MB de memória a menos do que o normal. O Xen reserva essa memória ao ser iniciado.

Ao se iniciar o domo no Xen, os scripts de inicialização devem iniciar o *Xend*, o *daemon* que roda no domo, e permitir que você controle as outras máquinas virtuais. Se o *daemon* estiver em execução, o usuário *root* pode usar a ferramenta *xm* para exibir as informações sobre as outras máquinas virtuais, iniciar novas máquinas, parar as já existentes, migrá-las, destruí-las, e assim por diante. O *Xend* também permite que se defina as alocações de memória para as máquinas virtuais, junto com sua prioridade na CPU.

Para uma visão geral dos comandos básicos do *xm*, digite `xm help`; `xm info` dá informações sobre a versão do Xen; `xm dmesg` exibe as mensagens de inicialização do hypervisor; `xm list` é um comando que você usará com alguma frequência: ele mostra uma lista de máquinas virtuais ativas e seu consumo de memória.

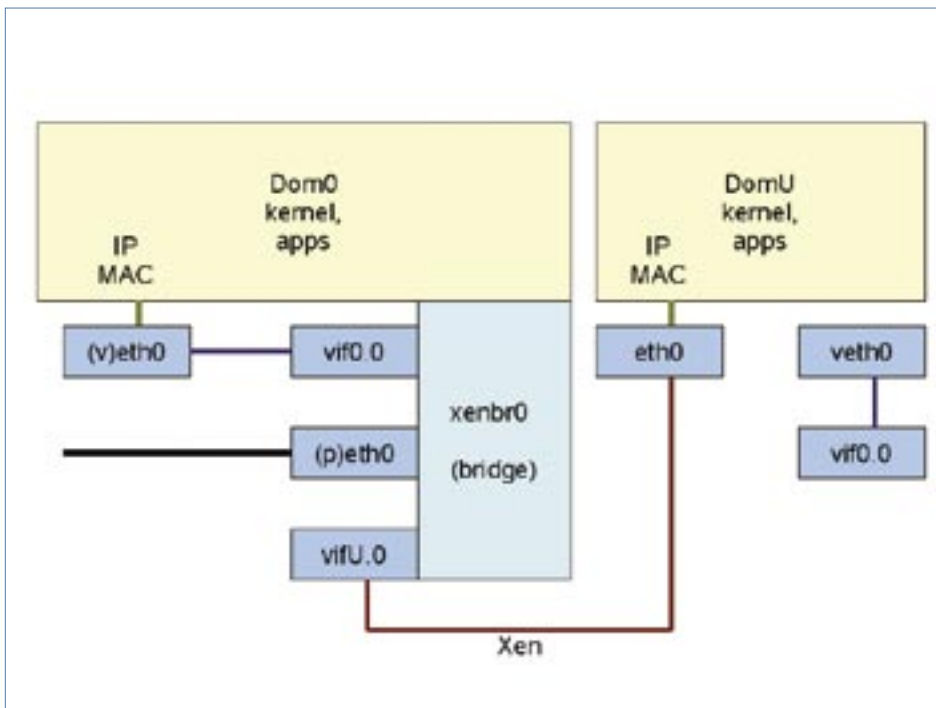


Figura 2 As interfaces `(v)eth0` e `vif0.X` são conectadas internamente como um *pipe*. Os pacotes vindos dos *domUs* podem acessar tanto a rede externa (via `(p)eth0`) quanto outros domínios.

Máquinas virtuais

A configuração do Xen fica guardada no arquivo `/etc/xen/xend-config.sxp`. As configurações de rede padrão nesse arquivo usam *bridging*, com uma *bridge* em domo. A *bridge xenbr0* simplesmente repassa os pacotes de rede que chegam por uma interface para todas as outras interfaces de rede. A configuração é dificultada pelo fato de o Xen renomear as interfaces de rede na inicialização. A interface `eth0` vira `peth0`; os endereços de hardware (MAC) e IP são copiados para a interface `veth0`, que então é renomeada para `eth0`. Internamente, `veth0` é vinculada a `vif0.0`, que se conecta à *bridge*, assim como faz a `peth0`. Os pacotes vindos de aplicações no domo agora são roteados através da nova `veth0` (virtual), e pela `vif0.0` para a *bridge xenbr0*, e de lá para a interface física `peth0` e a rede externa.

Você deve estar se perguntando se existe algum motivo para essa configuração complicada. Assim que outros domínios são iniciados, suas interfaces virtuais apenas se conectam à *bridge*, o que lhes dá acesso tanto à rede física quanto ao domo. A **figura 2** mostra uma visão geral desse processo.

O *bridging* é bastante complexo – mas pode-se configurar o Xen para usar roteamento e evitar que as interfaces de rede sejam renomeadas e recorram a um *bridge*. Isso também faz sentido se você utiliza uma interface de rede sem fio – não é possível enviar pacotes Ethernet através da *bridge* em pacotes sem fio, pois o formato é diferente.

A **figura 2** mostra outro domínio que iniciaremos em breve. Como o domínio inicia seu próprio kernel, precisa também de seu próprio sistema de arquivos, que também será inicializado virtualmente. Há algumas soluções para oferecer um sistema de arquivos para o domínio:

- ◆ Reutilize uma instalação pré-existente. Simplesmente instale o kernel para-virtualizado, que permitirá carregar os módulos do kernel respectivo; depois você pode inicializar com o Xen uma partição Linux pré-existente.
- ◆ Baixe uma imagem existente. Vá ao *XenWiki* se quiser imagens já feitas, ou acesse [\[2\]](#) para downloads.

- ◆ Use métodos de instalação específicos para a sua distribuição, como o módulo *YaST* para a instalação do Xen.
- ◆ Instale numa partição separada ou num diretório que você possa converter manualmente para uma instalação do Xen rodando `kernel-xen`.

Apenas uma máquina virtual deve acessar as imagens do sistema de arquivos em qualquer momento. Se mais máquinas montarem um sistema de arquivos (sem que esse seja feito para clusters), os dados podem ser corrompidos, exatamente como ocorreria com máquinas físicas.

Configurar as máquinas virtuais é um processo simples (veja o **quadro 3**).

No Xen, você encontra alguns arquivos de configuração de exemplo bem documentados, no diretório `/etc/xen`. No arquivo de exemplo, primeiro o kernel é configurado, seguido de um *initrd* (opcional). Esses arquivos são carregados do sistema de arquivos do domo quando o domU é iniciado. Isso pode ser um problema, mas vamos discutir isso em maior profundidade mais adiante. A máquina virtual receberá 128 MB de memória; uma interface virtual de rede será criada com o endereço MAC especificado, e a interface será conectada à *bridge*

`xenbr0`. A partição lógica `/dev/VG_new/LV_iSCSI1` é vinculada à máquina virtual como o disco `hda1`. Os parâmetros de inicialização passados para ela são `root=` e a máquina virtual. O nome da MV é `iSCSI1`.

Em vez de um dispositivo único de rede, poderíamos ter configurado múltiplos dispositivos. O mesmo se aplica às partições do disco. Também é possível emular um disco rígido inteiro ao invés de apenas uma partição. Naturalmente, a partição usada para esse fim deve conter uma tabela de partições. Especificar `file:` permite que se exportem arquivos com imagens de sistemas de arquivos como partições, ou arquivos com imagens de discos rígidos como discos virtuais, para a máquina virtual.

Ao iniciar a máquina virtual, os dispositivos no domo são ativados por *hotplugging*. O domo recebe um evento por cada rede e disco descobertos e ativados. O evento é parte do subsistema de *back-end* do Xen, e usa os mesmos nomes `vif*` que o kernel para a rede, e `vbd*` para o disco rígido. O *Udev* então chama os scripts apropriados em `/etc/xen/scripts`. Depois de os scripts terminarem e o dispositivo aparecer no banco de dados *Xenstore*, um evento de *hotplug* é disparado no domU para tornar o dispositivo visível, seguindo o mesmo procedimento

Quadro 3: Exemplo de configuração de um domínio Xen

```
01 # Kernel e initrd a serem carregados
02 kernel = "/boot/vmlinuz-2.6.13-15.8-xen"
03 ramdisk = "/boot/initrd-2.6.13-15.8-xen"
04 memory = 128
05 # Nome, CPUs virtuais, discos e rede virtuais
06 name = "iSCSI1"
07 #vcpus = 2
08 vif = [ 'mac=aa:cc:10:00:00:a1, bridge=xenbr0' ]
09 disk = [ 'phy:VG_new/LV_iSCSI1,hda1,w' ]
10 # Parâmetros a passar para a MV
11 dhcp = "dhcp"
12 hostname = "iSCSI1"
13 root = "/dev/hda1 ro"
14 # extra = ""
```

usado para o hardware físico durante a inicialização.

O hotplug é um dos entraves ao uso desimpedido do Xen. Distribuições mais antigas usam o velho mecanismo de hotplug ao invés do Udev. Nesses casos, o arquivo `/etc/hotplug/xen-backend.agent` é usado no lugar do `/etc/udev/rules.d/40-xen.rules`. Se o hotplug não estiver corretamente configurado, uma mensagem de erro pode ser mostrada. Mas tenha em mente que, se você usar a versão do Xen que veio com a sua distribuição, esse problema não deve ocorrer.

Um `xm create -c arquivo_de_conf` inicia a máquina virtual. A opção `-c` associa um console virtual à máquina virtual para você conseguir assistir à sua inicialização. Pressionar **[Ctrl] + [I]** separa o terminal da MV, mas é possível reconectá-lo digitando `xm console`.

O processo de inicialização provavelmente falhará com uma mensagem dizendo que não existe memória suficiente, pois o domU usa toda a memória disponível na inicialização. Você pode reduzir a alocação de memória para o domU ou mandar o `Xend` fazer isso automaticamente.

Use `xm mem-set 0 XXX` para o primeiro, e uma configuração de `dom0-min-mem` em `/etc/xen/xend-config.sxp` para o segundo. Após transpor esse obstáculo, já deve ser possível inicializar o domU.

Nas versões mais velhas do kernel Linux, um driver “balão” especial era usado para a manipulação de memória. O driver ocupava a memória conforme necessário, entregando ou devolvendo-a ao Xen. A memória era marcada como reservada no Linux. Os kernels 2.6 mais recentes usam uma infra-estrutura de hotplug de memória para que o `free` retorne um tamanho de memória diferente após uma operação de `mem-set`.

Nesse estágio, faz sentido um *tour* pela ferramenta `xm`. Você pode parar, fazer backup e reiniciar a máquina virtual, modificar o tamanho e a prioridade da CPU, e várias outras coisas.

Otimização do domU

Se você seguiu os passos até aqui, agora já pode inicializar um domU do Xen em modo texto. Você notará que alguns scripts do sistema falharam: qualquer tentativa de acesso direto ao hardware, como o carregamento de módulos do kernel espe-

cíficos para o hardware ou o acesso ao relógio da CMOS (`hwclock`), está fadado a falhar.

Na maioria dos casos, você pode simplesmente desativar os scripts. O acesso ao hardware não faz nenhum sentido, e a hora na máquina virtual é automaticamente sincronizada com o relógio físico.

Para iniciar um ambiente gráfico, a solução mais fácil é iniciar o `Xvnc` ao invés de um servidor X normal. A seguinte entrada em `/etc/X11/xdm/Xservers` cuidará disso:

```
:0 local /usr/X11R6/bin/Xvnc -br
  -geometry 960x720 -depth 24 -rfbwait
  120000 -rfbauth /root/.vnc/passwd
  -rfbport 5900 -httpd
  /usr/share/vnc/classes`
```

A entrada inicia um servidor X virtual com profundidade de cor de 24 bits e resolução de 960x720 pixels. O servidor escuta na porta 5900, e você pode usar um cliente `VNCviewer` para conversar com ele. Por motivos de segurança, você deverá fornecer uma senha, pedida pelo `vncpasswd`.

Mudar as entradas nos arquivos correspondentes de configuração permite que você use o KDM ou o GDM no lugar do XDM. A [figura 3](#) mostra uma captura de tela de uma máquina virtual gráfica – essa máquina virtual é uma configuração gráfica mínima do Suse Linux 10.0 que usa o gerenciador de janelas `openbox`. Pode-se baixar uma imagem pré-compilada dessa configuração em [\[2\]](#). A máquina é mantida mínima por um motivo: você pode usá-la como uma instalação básica e instalar por cima todos os pacotes que quiser.

Armazenamento e migração

A possibilidade de migrar uma máquina virtual de um hospedeiro físico para outro é um dos recursos mais úteis do Xen. Ele dá aos administradores a possibilidade de agendar a manutenção planejada do hardware, ou transferir a carga de uma máquina sobrecarregada, ou apenas mudar um serviço específico para uma nova máquina. Há outros usos possíveis em ambientes de tes-

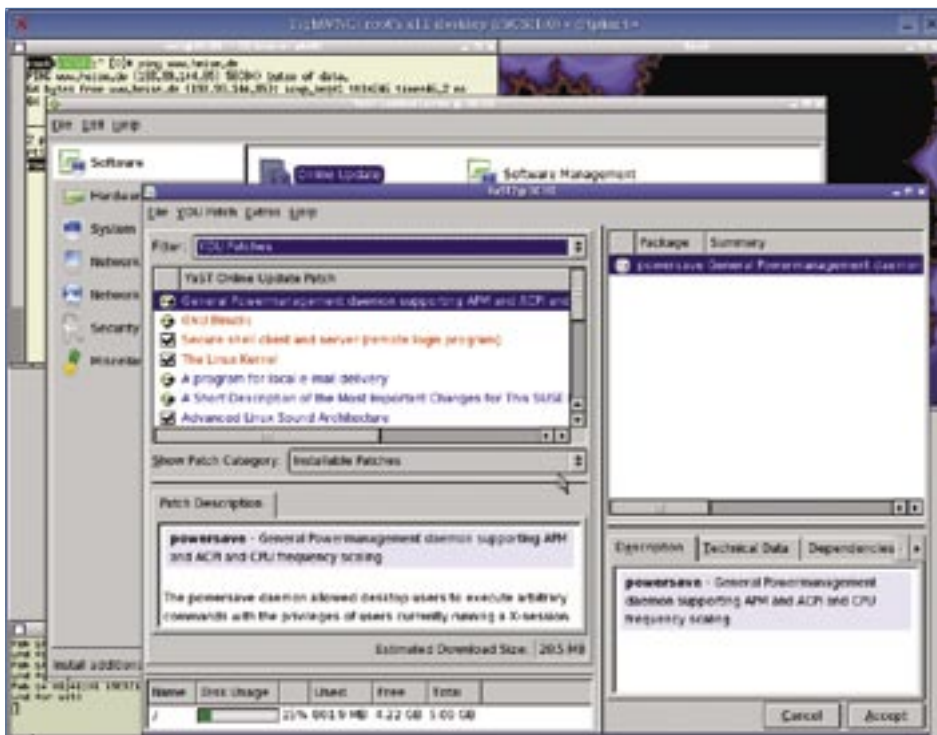


Figura 3 Captura da tela de uma máquina virtual usando `Xvnc`. O usuário `root` está ativo e rodando o `YaST online update` para aplicar as últimas atualizações de segurança à máquina virtual.

te e desenvolvimento, simulações de redes e clusters, ou ambientes de execução separados por motivos de segurança.

A máquina virtual carrega tudo embora junto com seu kernel, o que significa levar consigo o endereço IP e as configurações do kernel. Processos ativos do servidor continuarão rodando (após uma breve interrupção) e continuarão a usar o mesmo endereço IP.

A máquina virtual precisa de acesso a seu sistema de arquivos antes e depois da migração. Se a máquina não reside num disco RAM, será necessário usar um sistema de arquivos de rede ou um dispositivo de armazenamento de rede (NAS, ou *Network Attached Storage*). Você pode usar NFS como um sistema de arquivos de rede. Em aplicações de servidor, o armazenamento em rede geralmente baseia-se em *FibreChannel*, ao qual, naturalmente, ambas as máquinas físicas precisam ter acesso.

Uma solução de baixo custo é o iSCSI. *UNH* e *Enterprise Linux Target* oferecem dois softwares servido-

res iSCSI com suporte à emulação de dispositivos de rede.

Usar dispositivos de dados de rede (NBD, ou *Network Block Devices*) é ainda mais simples. Digitar `nbd-server 20001` ARQUIVO exporta o arquivo (com uma imagem de sistema de arquivos) para o `nbd`; a entrada respectiva no cliente é:

```
disk = [ 'nbd:IP 20001,hda1,w' ]
```

O desempenho é bom o suficiente se você tiver uma rede de 100 Mbps, como uma *FastEthernet*, por exemplo, e a migração leva poucos segundos numa Gigabit Ethernet.

Conclusões

O Xen 3 leva à virtualização em PCs a versatilidade antes restrita aos *mainframes*. Durante a escrita deste artigo, a instalação da maioria das distribuições envolvia algumas alterações manuais. Mas conforme a integração progride, todos nós podemos esperar uma explosão na virtualização, graças à flexibilidade que ela oferece às aplicações. ■

Mais Informações

- [1] Benchmarks do Xen:
<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/performance.html>
- [2] Imagens para máquinas hóspedes:
<http://www.suse.de/~garloff/linux/xen/images/>
- [3] Xen: <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>
- [4] Últimas notícias sobre o Xen em seu Wiki: <http://wiki.xensource.com/>

O autor

O Linux atraiu a atenção de **Kurt Garloff** quando ele estava na universidade. Quando um driver SCSI que deveria funcionar não o fez, a carreira de Kurt como desenvolvedor do kernel começou. Após sua formatura no curso de Física, Kurt entrou na Suse Linux GmbH, e – depois de tirar o ano sabático para terminar seu doutorado em Eindhoven (Holanda) – ele agora é chefe dos laboratórios de Suse da Novell.

Os Thin Clients da ThinNetworks

Melhor relação custo benefício do mercado e funcionalidades exclusivas

O TC-NET revoluciona o mercado de ThinClients ao apresentar um sistema operacional baseado em Linux e funcionalidades exclusivas, o TC-OS. Compatível com a maioria dos servidores de terminal existentes, essa solução agrega redução de custos de aquisição e atualização, facilidade de instalação e manutenção, maior estabilidade e confiabilidade, maior segurança, baixo consumo de energia e tamanho compacto.



O sistema Operacional TC-OS



- Recursos de administração centralizados;
- Servidor VNC para permitir o controle remoto de cada ThinClient;
- Disponível com Flash (64mb ou 128mb) ou na versão PXE para uso com LTSP;
- Clientes para CITRIX-ICA, Microsoft RDP 5.1, XDMCP e VNC e Tarantella;
- Diversos aplicativos para uso local (navegador, Java, Skype, protetor de tela e outros).



ThinNetworks
Soluções Thin Client

Fone: (61) 3366-1333
Skype: ThinNetworks
vendas@thinnet.com.br
www.thinnetworks.com.br