

Virtualização e kernel, vistos por dentro

Eduardo Habkost

`ehabkost@redhat.com`

Tchelinix Porto Alegre 2008

Nível: Avançado

Escopo: Código e processo de desenvolvimento do kernel

Pré-requisitos: Noções básicas sobre o kernel Linux e seu processo de desenvolvimento

Conteúdo

- 1 **Introdução**
 - Paravirtualization, full-virtualization, etc.
- 2 **Linux-based virtualization**
 - KVM
 - Mudanças no kernel
- 3 **Trabalhando upstream**
- 4 **Conclusão**

Conteúdo

- 1 **Introdução**
 - Paravirtualization, full-virtualization, etc.
- 2 Linux-based virtualization
 - KVM
 - Mudanças no kernel
- 3 Trabalhando upstream
- 4 Conclusão

Full-virtualization

Parece de verdade!

- Parece uma máquina de verdade
- VMWare, Qemu (com ou sem kqemu), Bochs, outros
- Ou com ajuda do hardware (AMD-V, Intel-VT)
 - Suportado pelo Xen
 - No Linux: KVM
- Performance ruim para I/O
 - Fácil para o hardware \neq fácil para o software
 - Operação de I/O \rightarrow pula para o *host*

Paravirtualização

“Eu sei que é de mentira”

- Kernel do guest modificado
- Xen
- lguest
- VMI (VMWare)
- User-mode Linux (por que não?)
- Comunicação com o *hypervisor* através de *hypercalls*

Paravirtualized drivers

Um agente infiltrado

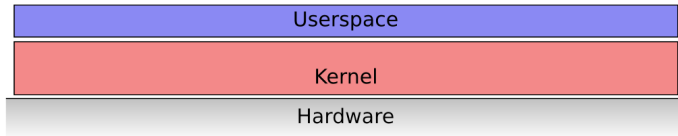
- Ainda parece máquina de verdade
- Mas com um hardware “meio diferente”
- Exemplo: “device driver disk” do VMWare
- Para o SO, é só um hardware diferente que precisa de um driver
- O driver conversa com o *hypervisor*
- O melhor dos dois mundos
- Exemplos: Xenbus (Xen), VirtIO (genérico), drivers do VMWare

Conteúdo

- 1 Introdução
 - Paravirtualization, full-virtualization, etc.
- 2 Linux-based virtualization
 - KVM
 - Mudanças no kernel
- 3 Trabalhando upstream
- 4 Conclusão

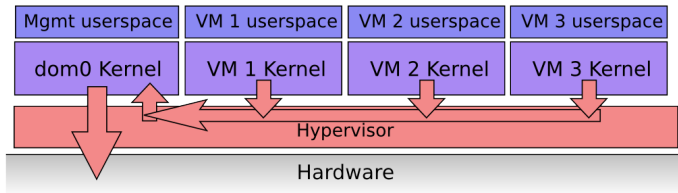
Linux-based virtualization

Bare metal:



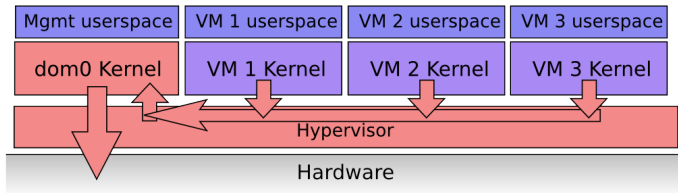
Linux-based virtualization

Xen (teoria):



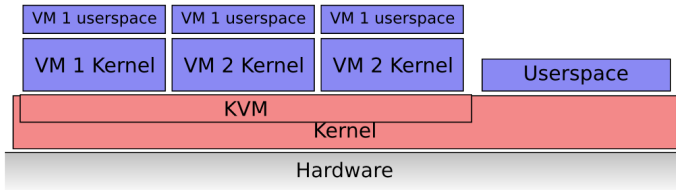
Linux-based virtualization

Xen (prática):



Linux-based virtualization

KVM:



Conteúdo

- 1 Introdução
 - Paravirtualization, full-virtualization, etc.
- 2 Linux-based virtualization
 - KVM
 - Mudanças no kernel
- 3 Trabalhando upstream
- 4 Conclusão

KVM

Kernel-based Virtual Machine

- Full-virtualization
- Precisa de suporte na CPU
- Aproveita:
 - scheduler
 - proteção entre processos; permissões
 - gerenciamento de energia
 - drivers
 - gerenciamento de memória
- *Context switches* a menos
- Kernel: virtualização da CPU, exposta via `/dev/kvm`
- Userspace (qemu modificado): I/O, UI, política

Conteúdo

- 1 Introdução
 - Paravirtualization, full-virtualization, etc.
- 2 Linux-based virtualization
 - KVM
 - Mudanças no kernel
- 3 Trabalhando upstream
- 4 Conclusão

Mudanças no kernel

- KVM
- lguest: simples, paravirtualização
- **paravirt_ops**
- **Unificação x86**
- **VirtIO**

paravirt_ops

- Paravirtualização
- Antes: recompilação do kernel do guest para suportar paravirt
- Agora: mesma imagem do kernel, vários guests
- Truques para substituir código *on-the-fly* (parecido com SMP alternatives)
- Início: 2.6.20 (i386), lguest
- Hoje: x86 (32 e 64-bits), ia64
- Usuários hoje: lguest, VMI, Xen, KVM (paravirt. drivers)

Unificação x86

- i386, x86_64 -> x86
- Muito código parecido e duplicado
- Muito código i386 usado silenciosamente em x86_64: fácil de cometer erros
- Features novas precisavam ser “reimplementadas” (ex.: paravirt_ops)
- Ainda existem arquivos `*_32.c` e `*_64.c` a unificar

VirtIO

- Padrão *de-facto* para dispositivos virtuais
- ABI guest ↔ host
- API para drivers
- Dispositivos aparecem como um dispositivo PCI
- Drivers “comuns” (virtio-net, virtio-blk, etc.)
- “A” Solução de paravirtualized drivers para o KVM
- Também existe esforço para suportar no Xen

Conteúdo

- 1 **Introdução**
 - Paravirtualization, full-virtualization, etc.
- 2 **Linux-based virtualization**
 - KVM
 - Mudanças no kernel
- 3 **Trabalhando upstream**
- 4 **Conclusão**

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem ainda mais trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem ainda mais trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem **ainda mais** trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem **ainda mais** trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem **ainda mais** trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Trabalhando upstream

- Enviar seu código upstream dá trabalho
- O kernel é um alvo móvel
- Mas vale a pena. Senão você vai ter que escolher:
 - Fica preso a uma versão pré-histórica do kernel; ou
 - Tem **ainda mais** trabalho, podendo ser tarde demais
- O Xen é um bom exemplo didático

Uma história sobre trabalho upstream

Árvore do kernel separada com suporte ao Xen, 2.6.18



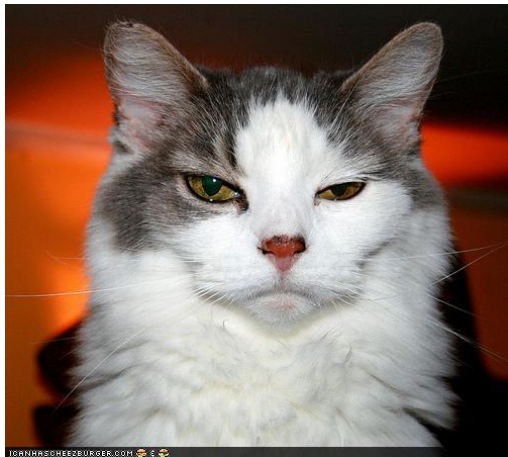
Uma história sobre trabalho upstream

Não boa o suficiente para envio upstream



Uma história sobre trabalho upstream

“E daí?”



Uma história sobre trabalho upstream

Novo kernel: 2.6.19



Uma história sobre trabalho upstream

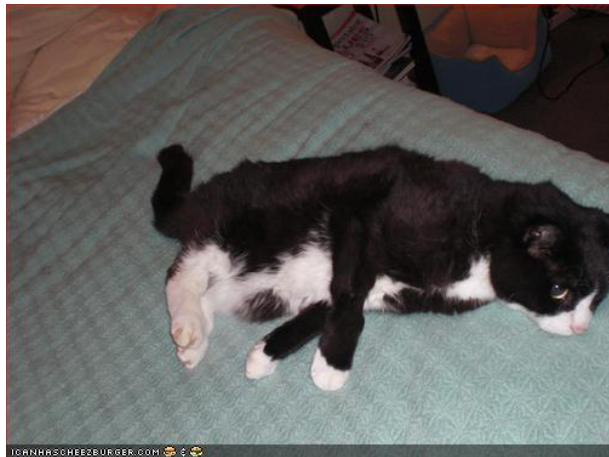
Algumas distribuições adaptam o código para o 2.6.19



É o chamado *forward-port*

Uma história sobre trabalho upstream

XenSource continua usando o 2.6.18...



Uma história sobre trabalho upstream

Novo kernel: 2.6.20



Uma história sobre trabalho upstream

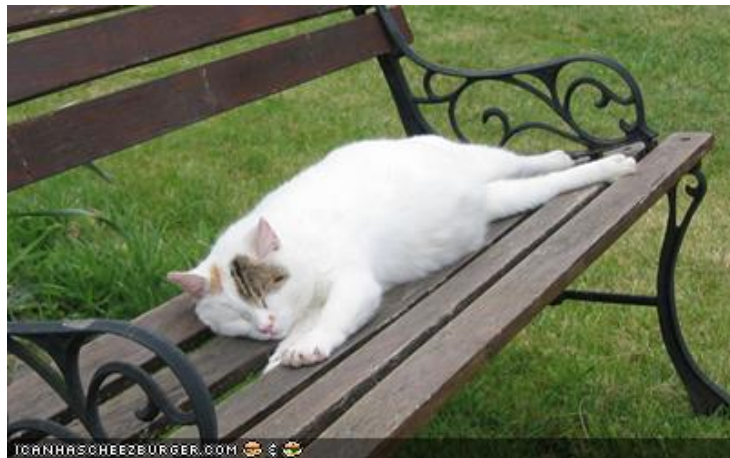
Distribuições fazem forward-port para o 2.6.20



Um pouco mais complicado: paravirt_ops começa a entrar

Uma história sobre trabalho upstream

XenSource continua usando o 2.6.18...



Uma história sobre trabalho upstream

Novo kernel: 2.6.21



ICANHASCHEEZBURGER.COM 🍷 🍷

Uma história sobre trabalho upstream

2.6.21: Forward-port realmente complicado



paravirt_ops usado em muitos pontos do código também modificados pelo patch do Xen

Uma história sobre trabalho upstream

Novo kernel: 2.6.22



Uma história sobre trabalho upstream

Quatro versões do kernel sem Xen oficial



Uma história sobre trabalho upstream

E a XenSource ainda no kernel 2.6.18



Uma história sobre trabalho upstream

Nem todo mundo chega a portar para 2.6.22



Uma história sobre trabalho upstream

2.6.23



Uma história sobre trabalho upstream

Forward-port mostra-se um trabalho sem fim



Tem que ter alguma outra solução

Uma história sobre trabalho upstream

E a XenSource continua tendo que manter sua árvore do 2.6.18



Com correções de bugs, segurança, suporte a hardware

Nem tudo está perdido

A partir do 2.6.26, algum suporte para o Xen entrou



Nem tudo está perdido

2.6.27 está um pouco melhor



Mas ainda falta muita coisa

Nem tudo está perdido

- O trabalho para incluir todas as features do Xen do 2.6.18 é grande
- Ninguém sabe quando isso vai estar pronto...
- “Emulando” o hypervisor Xen usando o KVM: Xenner

Nem tudo está perdido

- O trabalho para incluir todas as features do Xen do 2.6.18 é grande
- Ninguém sabe quando isso vai estar pronto...
- “Emulando” o hypervisor Xen usando o KVM: Xenner

Nem tudo está perdido

- O trabalho para incluir todas as features do Xen do 2.6.18 é grande
- Ninguém sabe quando isso vai estar pronto...
- “Emulando” o hypervisor Xen usando o KVM: Xenner

Nem tudo está perdido

- O trabalho para incluir todas as features do Xen do 2.6.18 é grande
- Ninguém sabe quando isso vai estar pronto...
- “Emulando” o hypervisor Xen usando o KVM: Xenner

Xenner

- Usa `/dev/kvm`
- Excelente candidato para usar um futuro “kvm-lite”
- Utiliza parte do código userspace Xen
- Mesmo que o Xen “tradicional” fique pra trás, a ABI pode demorar a morrer

Conteúdo

- 1 Introdução
 - Paravirtualization, full-virtualization, etc.
- 2 Linux-based virtualization
 - KVM
 - Mudanças no kernel
- 3 Trabalhando upstream
- 4 Conclusão

Futuro

- Mais paravirtualização
- Mais performance
 - *zero-copy I/O* (disco, rede)
 - *PCI passthrough*
 - Video: SPICE
- Integração com o qemu upstream
- kvm-lite?
- O que vai acontecer com o Xen?

Olhando para cima

O mundo não é só feito de kernel

- libvirt: API pra gerenciar VMs (Xen, KVM, outros)
- virt-manager: GUI
- oVirt: interface de gerenciamento

Referências

- Esta apresentação:
<http://raisama.net/talks/virt-2008/>
- “*virtio: towards a de-facto standard for virtual I/O devices*”,
by Rusty Russel
- KernelNewbies Virtualization Wiki:
<http://virt.kernelnewbies.org/>
- KVM: <http://kvm.qumranet.com/>
- lguest: <http://lguest.ozlabs.org/>
- Xenner:
<http://kraxel.fedorapeople.org/xenner/>
- Unificação x86:
<http://www.glommer.net/blogs/?p=265>

Perguntas?

Obrigado