Sistema de Virtualização

KVM - Kernel-based Virtual Machine

Objetivos

- Conhecer o hardware do HOST.
- Instalação de pacotes e módulos necessários.
- Dar uma visão geral da mecânica do KVM, usando os principais parâmetros de linha comando e o monitor.
- Explorar algumas configurações (disco, rede, memória e cpu).
- Live migration e PCI Passtrough

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Descrição do KVM

The following patchset adds a driver for Intel's hardware virtualization extensions to the x86 architecture. The driver adds a character device (/dev/kvm) that exposes the virtualization capabilities to userspace. Using this driver, a process can run a virtual machine (a "guest") in a fully virtualized PC containing its own virtual hard disks, network adapters, and display.

Using this driver, one can start multiple virtual machines on a host. Each virtual machine is a process on the host; a virtual cpu is a thread in that process. kill(1), nice(1), top(1) work as expected.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Descrição do KVM

- KVM é implementado como um módulo de kernel carregável que converte o Linux em um hypervisor bare-metal.
- Há dois princípios fundamentais de projeto que ajudaram o KVM a amadurecer rapidamente em um hypervisor estável e de alto desempenho.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

KVM em funcionamento

```
#include <linux/kvm.h>


open("/dev/kvm")
ioctl(KVM_CREATE_VM)
ioctl(KVM_CREATE_VCPU)
for (;;) {
     ioctl(KVM_RUN)
     switch (exit_reason) {
     case KVM_EXIT_IO:  /* ... */
     case KVM_EXIT_HLT: /* ... */
     }
}
```

Mais detalhes em:
http://blog.vmsplice.net/2011/03/qemu-internals-big-picture-overview.html

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

KVM em funcionamento

```
# egrep "(vmx|svm)" /proc/cpuinfo
   flags                : ... vmx ....
   flags                : ... vmx ....
   flags                : ... vmx ....
   flags                : ... vmx ....
```

```
# modprobe kvm
# modprobe kvm_intel ou kvm_amd
# qemu-img create -f qcow vm-disk.img 4G
# kvm -m 384 -cdrom guestos.iso -hda vm-disk.img -boot d
```

- KVM é implementado como um módulo de kernel carregável que converte o Linux em um hypervisor bare-metal.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Não reinvente a roda

- Há muitos componentes que um hypervisor exige, além da capacidade de virtualizar a
- CPU e memória, por exemplo
- Gerenciador de memória
- Escalonador de processos
- Pilha de I/O e rede
- Drivers de dispositivo
- Gerenciamento de segurança.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Simplificando

- Na verdade um hypervisor é realmente um sistema operacional especializado, diferindo apenas dos de uso geral pois que se executam máquinas virtuais ao invés de aplicações.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Por que o Linux?

- Uma vez que o kernel do Linux já inclui os principais recursos exigidos por um hypervisor e foi amadurecido em uma plataforma da madura e estável por mais de 15 anos de apoio e desenvolvimento, é mais eficiente de construir sobre essa base ao invés de escrever todos os componentes necessários, tais como um gerenciador de memória, escalonador, etc a partir do zero.

- Neste contexto, o KVM se beneficiou da experiência do Xen. Um dos principais desafios do arquitetura do Xen é a arquitetura da divisão de domínio0 e o hypervisor Xen.
  Desde o hypervisor Xen fornece os recursos da plataforma central dentro da pilha, ele tem a necessidade de implementar esses recursos, como escalonador e gerenciador de memória a partir do zero.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Por que o Linux?

- Por exemplo, enquanto o kernel do Linux possui um gerenciador de memória madura e comprovada, incluindo suporte para NUMA e sistemas de grande escala, o hypervisor Xen necessita construir este apoio a partir do zero. Da mesma forma recursos como gerenciamento de energia que já estão maduros e comprovado em campo no Linux tinha que ser re-implementado no hypervisor Xen.

- Outra decisão importante tomada pela equipe KVM era incorporar o KVM no kernel do Linux o quanto antes. O código KVM foi apresentado à comunidade do kernel Linux em dezembro de 2006 e foi aceito em o kernel 2.6.20, em janeiro de 2007.

- Neste ponto KVM tornou-se parte do núcleo do Linux e é capaz de herdar recursos chave do kernel do Linux.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

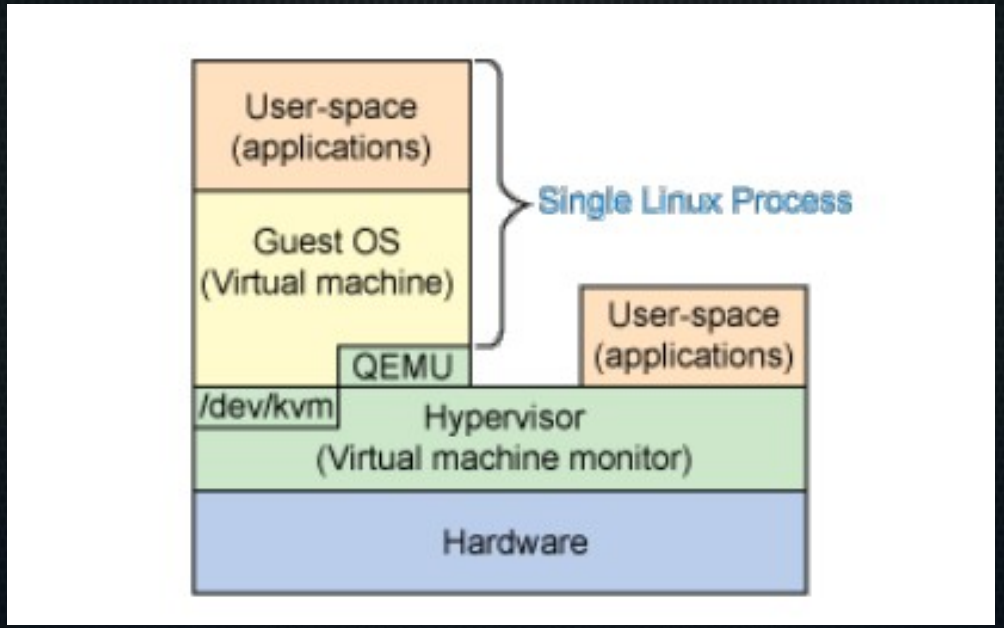
Por que o Linux?

- Em contrapartida, as correções necessárias para construir o Domínio0 Linux para o Xen ainda não faz parte do kernel Linux e obriga os fornecedores a criar e manter um fork do kernel Linux.
- Isto tem levar a um aumento dos encargos com os distribuidores de Xen, que não podem alavancar facilmente os recursos do montante kernel. Qualquer novo recurso de correção de bug ou um patch para o kernel acrescentado a montante deve ser backportados para trabalhar com o patch define Xen.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Arquitetura KVM



KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Arquitetura KVM

- I/O e dipositivos são virtualizados através de um processo QEMU levemente modificado (cara processo executa uma máquina virtual).
- Execução de I/O de um sistema operacional convidado é fornecido com o QEMU.
- QEMU é uma solução de virtualização de que permite a virtualização de um ambiente de PC inteiro (inclusive discos, placas gráficas, dispositivos de rede). Quaisquer solicitações de I / O de um sistema operacional convidado faz são interceptadas e direcionadas para o modo de usuário para ser emulado pelo processo QEMU.

KVM - Kernel-based Virtual Machine

Conhecer o hardware do host

Arquitetura KVM

- KVM permite a virtualização de memória através do dispositivo /dev/kvm.
- Cada sistema operacional convidado tem seu próprio espaço de endereço que é
- mapeado quando o convidado é instanciado.
- A memória física que é mapeada para o sistema operacional guest é a memória virtual
- mapeada para o processo.

KVM - Kernel-based Virtual Machine

Instalação de pacotes e módulos necessários

Carregando o módulo

```
# modprobe kvm kvm_intel
```

Verificando /dev/kvm:

```
# ls -l /dev/kvm
crw-rw---- 1 root kvm 10, 232 Dec 14 17:20 /dev/kvm
```

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Boot de um CD/DVD

Iniciando uma VM dando boot em uma imagem de LiveCD:

```
# qemu-kvm -drive file=/caminho/imagem.iso,media=cdrom -m 512
```

Com esse comando uma janela será aberta e a máquina virtual estará em funcionamento com 512 Mb de RAM. O boot será dado na imagem ISO. Apesar do parâmetro media se chamar cdrom, essa imagem pode ser de DVD também.

Existe também essa sintaxe:

```
# qemu-kvm -cdrom /caminho/imagem.iso -m 512
```

O parâmetro -cdrom é mantido apenas para garantir compatibilidade e seu uso não é mais recomendado.

É possível também utilizar uma mídia real, apontando o dispositivo onde ela se encontra:

```
# qemu-kvm -drive file=/dev/sr0,media=cdrom -m 512
```

Dentro da máquina virtual, dar um lspci para listar os dispositivos padrões (vídeo, rede, etc).
Dica para teclado: adicionar -k pt-br caso tenha problemas com a tecla / ?.

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Usando e conhecendo o monitor

Clique na janela da máquina virtual para que ela receba o foco do mouse. Teclas de atalho importantes:

[Ctrl] + [Alt]: Tira o mouse do foco da janela
[Ctrl] + [Alt] + [1]: Mostra o saída de vídeo da máquina virtual.
[Ctrl] + [Alt] + [2]: Muda para o console de comando do qemu-kvm, o monitor.
[Ctrl] + [Alt] + [3]: Mostra a saída da porta serial

O monitor do qemu-kvm tem várias funções, dentre elas:

Changing or eject removable media (CD / DVD-ROMs, floppy disks).
The freezing and further run a virtual machine.
Backing up and restoring various states of the virtual machine.
Inspecting the state of a virtual machine.
The migration of a virtual machine to another host.
Changing the hardware (USB, PCI, ...).
The injection of emulated hardware failures.
Com a combinação [Ctrl] + [Alt] + [2] é aberto o monitor do QEMU.

Para conhecer mais comandos do monitor:

(Qemu) help

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Comandos básicos do monitor

The info command gives status information on the current instance. If you give info only, parameter is a list of possible output. The QEMU version is version info to determine.

(Qemu) info version
0.12.5

The kvm command info shows if the KVM hardware virtualization is enabled or not.

(Qemu) info kvm
kvm support: enabled

The quit command terminates the instance. This corresponds to switching off at a real machine and can cause data loss.

(Qemu) quit

The reset button corresponds to a real machine, the command system_reset.

(Qemu) system_reset

O daemon acpid tem quem que estar instalado no guest.

The following example sends the key combination [Ctrl] + [Alt] + [Del].

(Qemu) sendkey ctrl-alt-delete

The command sets Screendump the host system to display photos. This is useful in host systems that can not take screenshots. The screenshots are saved as PPM file (Portable Pixmap).

(Qemu) screendump imagem.ppm

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Comandos úteis do monitor

- drive_add / drive_del
- netdev_add / netdev_del
- device_add / device_del
- netdev_add / netdev_del (Antigamente: host_net_add / host_net_remove)
- info block
- info blockstats
- info network / (No futuro pode ser info netdev)
- info qtree
- info snapshots
- savevm / loadvm
- stop / cont

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Manipulando dispositivos - monitor

```
qemu-kvm -drive file=/caminho/imagem.iso,media=cdrom -m 512 \
    -monitor stdio
```

Ou:

```
qemu-kvm -drive file=/caminho/imagem.iso,media=cdrom -m 512 \
    -chardev stdio,id=mon0 -mon chardev=mon0
```

Ou múltiplos monitores, em diferentes protocolos:

```
qemu-kvm -chardev stdio,id=mon0 -mon chardev=mon0 \
    -chardev socket,id=tcpmon0,port=5000,host=localhost,server,nowait \
    -mon chardev=tcpmon0 \
    -drive file=/caminho/imagem.iso,media=cdrom -m 512
```

O monitor em Ctrl+Alt+2 não existe mais. Pode dar nc localhost 5000 que o monitor estará acessível. Consulte a manpage do kvm para conhecer todos os dispositivos de backend para chardevs.

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Fazendo a instalação de uma VM

Escolha a distribuição de sua preferência e baixe a iso:

Crie um arquivo para armazenar a VM:

```
# qemu-img create -f raw nome-do-arquivo.img 5G
```

Inicie a VM com o a ISO no CDROM:

```
# qemu-kvm -m 1024 -drive \
file=/caminho/imagem.iso,media=cdrom,index=1,boot=on \
-drive file=/caminho/arquivo.img,media=disk,index=0
```

Mate a VM e inicie ela sem CDROM:

```
# qemu-kvm -m 1024 -drive file=a.img,media=disk,index=0
```

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Saindo do ambiente gráfico - VNC

```
 # qemu-kvm -m 1024 -drive file=a.img,media=disk,index=0 -vnc :0
```

O kvm estará ouvindo na porta 5900+d a porta que foi passada na linha de comando. Pode-se usar qualquer cliente VNC para ver o display VGA.

```
 # netstat -nltp | grep kvm
tcp        0      0 0.0.0.0:5900    0.0.0.0:*         LISTEN 3589/kvm
```

Colocando senha:

```
 # qemu-kvm -m 1024 -drive file=imagem.img,media=disk,index=0 \
   -vnc :0,password -monitor stdio
QEMU 0.14.0 monitor - type 'help' for more information
(qemu) change vnc password
Password: ******
(qemu)
```

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Saindo do ambiente gráfico - porta serial

Podemos jogar um terminal GNU/Linux em uma porta serial.

- Edite os arquivos de configuração de sua distro para que exista um console disponível em uma porta serial.

- Dica (CentOS e Debian):
  - /etc/inittab: s1:2345:respawn:/sbin/getty 38400 ttyS0
  - No grub (/etc/grub.conf), coloque essa linha como parâmetro de boot: console=ttyS0

- Dica (Ubuntu)
  - cp /etc/init/tty1.conf /etc/init/ttyS0.conf, alterando o conteudo de tty1 para ttyS0
  - Editar /etc/default/grub
  - Na linha de GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" mudar para GRUB_CMDLINE_LINUX_DEFAULT="console=ttyS0"
  - Não esquecer de atualizar o grub: update-grub

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Saindo do ambiente gráfico - porta serial

- E com um terminal GNU/Linux jogando para um terminal podemos encaminhar para diversos lugares.
- Saida padrão

```
# qemu-kvm -m 1024 -drive file=imagem.img,media=disk,index=0 \
  -serial stdio
```

- Socket UNIX

```
# qemu-kvm -m 1024 -drive file=imagem.img,media=disk,index=0 \
  -serial unix:/tmp/portaserial,server,nowait
```

```
# socat UNIX:/tmp/portaserial STDIO,raw,echo=0,escape=0x1d
```

Sintaxe moderna:

```
# qemu-kvm -m 1024 -drive file=imagem.img,media=disk,index=0 \
-chardev socket,path=/tmp/portaserial,server,nowait,id=serial0 \
-device isa-serial,chardev=serial0
```

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Resumindo

Memoria

```
-m 1024
```

Saída de Vídeo (padrão).

```
-vnc :0
```

Monitor QEMU.

```
-chardev socket,id=tcpmon0,port=5000,host=localhost,server,nowait
-mon chardev=tcpmon0
```

```
-chardev stdio,id=mon0 -mon chardev=mon0 -monitor stdio
```

Saída Serial.

```
-chardev socket,path=/tmp/portaserial,server,nowait,id=serial0 -device
isa-serial,chardev=serial0 -serial unix:/tmp/portaserial,server,nowait
```

```
-chardev stdio,id=serial0 -device isa-serial,chardev=serial0
-serial stdio
```

KVM - Kernel-based Virtual Machine

Visão geral da mecânica do KVM

Resumindo

Mídia de CD.

```
-drive file=/isos/imagem.iso,media=cdrom,index=1,boot=on
-drive file=/dev/sr0,media=cdrom,index=1,boot=on
```

Discos.

Utilizando imagem.

```
-drive file=/diretorio/arquivoimagem.img,media=disk,index=0,boot=on
```

Utilizando partição real.

```
-drive file=/dev/sda3,media=disk,index=0,boot=on
```

Utilizando partição lvm.

```
-drive file=/dev/volumes/part01,media=disk,index=0,boot=on
```

Utilizando disco.

```
-drive file=/dev/sdb,media=disk,index=0,boot=on
```

Iniciando maquina em Background como um Daemon

```
-daemonize
```

Nome da Maquina

```
-name NOMEDAMAQUINA
```

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Disco

Tipos de controladoras (emulada ou virtual)

```
# kvm -drive file=/disco,media=disk,index=0,boot=on,if=virtio
# kvm -drive file=/disco,media=disk,index=0,boot=on,if=ide
# kvm -drive file=/disco,media=disk,index=0,boot=on,if=scsi
```

Caching de disco

`cache=none`
    usa O_DIRECT I/O que ignora o cache do host
`cache=writethrough`
    usa O_SYNC I/O que garante a confirmação da escrita no disco
`cache=writeback`
    usa o buffer de I/O do host

```
# kvm -drive file=/disco,media=disk,index=0,boot=on,if=virtio,cache=none
```

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Rede

User-space : padrão do KVM

```
    -netdev type=user,ifname=tap12,id=net0,script=no \
    -device virtio-net-pci,netdev=net0,mac=00:16:16:15:A5:50
```

TAP : Interfaces virtuais do Linux
(podendo ser incluidas em bridges Linux ou switchs virtuais)

```
    -netdev type=tap,ifname=tap12,id=net0,script=no \
    -device virtio-net-pci,netdev=net0,mac=00:16:16:15:A5:50
```

PCI Passthroung : Passar diretamente um dispositivo para o Guest

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Memoria

Tipos de overcommit

Swap
Esta é a forma clássica de apoio ao overcommit, o host escolhe algumas páginas de memória de um dos guests e as envia para o disco. Se um hóspede exige memória que tenha ido para a swap, o host trás de volta do disco.

Balões (ballooning)
Com o balão, o host e o guest cooperam em que páginas serão liberadas. É de responsabilidade do guest escolher a página e liberá-la, se necessário.

Compartilhamento de páginas
O hypervisor olha para páginas de memória que possuem dados idênticos, estas páginas são fundidas em uma única página, que é marcada apenas para leitura. Se um cliente escreve em uma página compartilhada, ela é "descompartilhada" antes de conceder ao guest a gravação.

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Memoria

Swap

Parâmetros de ajuste de swap:

```
# sysctl -w vm.swapiness=100
```

(quanto maior, mais agressivamente o kernel fará swap)

```
# sysctl -w vm.overcommit_memory=2
```

(0 heurístico, 1 overcommit sem limit, 2 não fazer overcommit)

```
# sysctl -w vm.overcommit_ratio=50
```

(quando overcommit_memory=2)

Quando overcommit_memory=2:
memória_total = swap + (memória_física * (overcommit_ratio / 100))

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Memoria

Balões (ballooning)

```
# kvm -balloon virtio
```

ou

```
# kvm -device virtio-balloon-pci
```

No monitor:

```
(qemu) info balloon
balloon: actual 1024
(qemu) balloon 512
(qemu) info balloon
balloon: actual 512
```

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

Memoria

Compartilhamento de páginas

KSM - Kernel Samepage Merging

Usando no host:

```
# echo 1 > /sys/kernel/mm/ksm/run
```

(demora alguns segundos para os primeiros resultados)

```
# cat /sys/kernel/mm/ksm/pages_sharing
5487
```

(multiplicar por 4KB para total compartilhado)

KVM - Kernel-based Virtual Machine

Explorando algumas configurações

CPU

```
# kvm -cpu ?
x86           [n270]
x86         [athlon]
x86        [pentium3]
x86        [pentium2]
x86         [pentium]
x86             [486]
x86         [coreduo]
x86           [kvm32]
x86         [qemu32]
x86           [kvm64]
x86        [core2duo]
x86         [phenom]
x86         [qemu64]
```

```
# kvm -cpu host
```

```
# kvm -smp 2
```

KVM - Kernel-based Virtual Machine

Live migration e PCI Passtrough

PCI passthrough

Passar um dispositivo diretamente para o guest.
Apenas um guest pode ter acesso.
Pode ser útil em casos de muita sensibilidade à latência como VoIP, ou hardware especializado.

```
# Parâmetro intel_iommu=on no boot
# lspci (identificar endereço no barramento)
# lspci -n (pegar o ID do dispositivo)
# echo "10ec 8136" > /sys/bus/pci/drivers/pci-stub/new_id
# echo "0000:04:00.0" > /sys/bus/pci/devices/0000\:04\:00.0/driver/unbind
# echo "0000:04:00.0" > /sys/bus/pci/drivers/pci-stub/bind
# qemu-kvm -device pci-assign,host=04:00.0
```

KVM - Kernel-based Virtual Machine

Live migration e PCI Passtrough

Live migration

No host A, inicie o kvm normalmente:

```
# qemu-kvm -drive file=/caminho/imagem-vm.img, ...
```

No host B, inicie o kvm usando o mesmo comando do host A, porém com o parâmetro -incoming:

```
# qemu-kvm -drive file=/caminho/imagem-vm.img, ... -incoming tcp:0:4444
```

Agora no monitor da máquina virtual em execução na host A, executar o comando migrate:

```
(qemu) migrate -d tcp:host_b:4444
(qemu) info migrate
Migration status: active
transferred ram: 17343 kbytes
remaining ram: 1023952 kbytes
total ram: 1057216 kbytes
```

Quando terminar, a máquina continua sua execução no host B.

Cuidados
As imagens/disposivos de discos virtuais devem estar acessíveis no host de destino.
Interfaces tap devem estar na mesma rede local, geralmente, dependendo da topologia da rede.
As versões do KVM devem ser exatamente a mesma.

Mais dicas de live migration: http://www.linux-kvm.org/page/Migration