Aula 07 - Dispositivos de discos

Sobre

- Objetivos:
  - Aprofundar o conhecimento sobre armazenamento virtualizado.
  - Questões de cache e otimização de acesso.
  - Administração de armazenamento.
  - Escalonadores de I/O.
  - Dando diferentes prioridades para máquinas virtuais.
  - Possíveis otimizações em sistema de arquivos.

Visão geral

- O host ou hypervisor:
  - Exporta discos virtuais para os guests
  - O guest usa-os como discos reais
- Os discos virtuais são mapeamentos para dispositivos reais
  - Discos inteiros, partições, volumes lógicos ou arquivos
  - Arquivos raw em um sistema de arquivos ou formatos de imagem

Virtual storage stack

Guest Filesystem
Guest Storage Driver
Storage hw emulation
Image format
Host filesystem
Host volume manager
Host storage driver

- Temos duas pilhas de armazenamento total no host e no guest
- Um sistema de arquivos potencialmente também nos dois
- Potencialmente também um formato de imagem (mini-sistema de arquivos)

Requisitos de Armazenamento

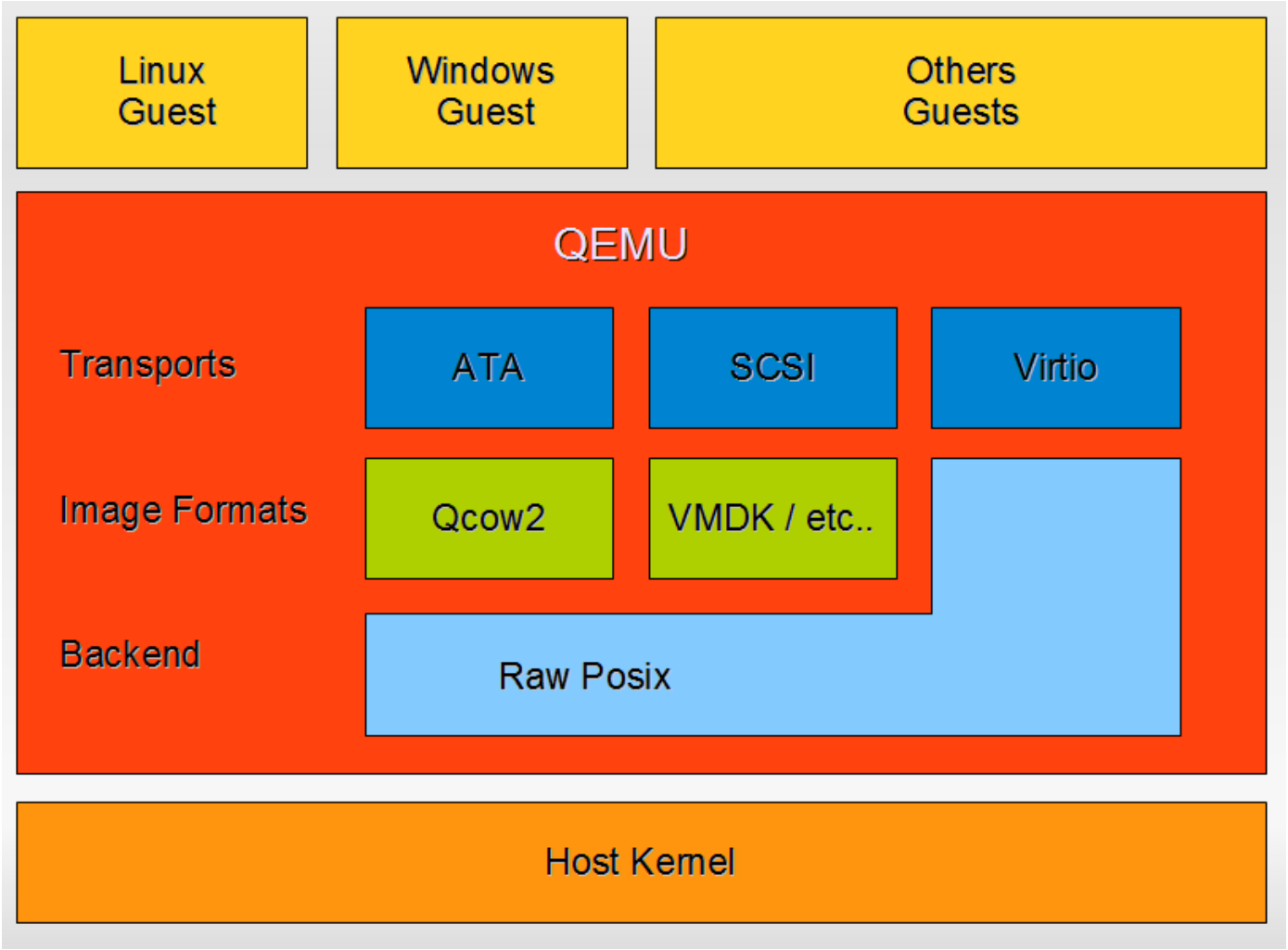
- Os requisitos de armazenamento tradicionais se aplicam:
  - A integridade dos dados: os dados devem realmente estar no disco quando o usuário/aplicação pedirem
  - Eficiência de espaço: queremos armazenar os dados do usuário/aplicação o mais eficientemente possível
  - Desempenho: fazer tudo o mais rápido possível
  - Capacidade de gestão: administrar diversos hosts

Requisitos - host

- É o lugar onde toda a inteligência fica
- Assegura a integridade dos dados

- Os dados estão realmente no disco quando o guest assume isso

Visão no QEMU/KVM



Storage Transports

- O QEMU fornece um simples controlador Intel ATA/IDE por padrão
- Funciona com quase todos os sistemas operacionais, porque é muito comum
- Alternativamente QEMU pode emular um controlador SCSI Symbios (ainda está melhorando)

Paravirtualização

- Fornecer interfaces melhores do que o hardware real
- Vantagem: deve ser mais rápido do que a virtualização/emulação completa
- Desvantagens: requer drivers especiais no guest

Paravirtualized storage transport

- QEMU fornece dispositivos virtualizados utilizando o framework VirtIO
- Fornece um simples driver de bloco

- Simples leitura/gravação de pedidos

Comando

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

Formatos de imagem

- Usuários querem uma recursos de gerencia de volumes em arquivos de imagem
- Snapshots
- Criptografia
- Compressão
- Snapshots também precisam armazenar metadados adicionais (memória, estado, etc)

Formatos de imagem

O QEMU suporta diversos formatos de imagem:

- cow - User Mode Linux
- vpc - Microsoft Virtual PC
- vmdk - VMware
- vdi - VirtualBox
- Bochs
- Parallels
- dmg – MacOS filesystem
- Outros.

Formatos de imagem - qcow2

- O qcow2 foi o formato de imagem principal do QEMU
- Suporta snapshots
- Suporta criptografia e compressão
- É suportado pela Red Hat e tem recebido muitas melhorias nos últimos tempos

Backends não baseados em imagem

- curl: permite a utilização de imagens a partir da Internet através de conexões HTTP e FTP.
- nbd: acesso direto a servidores nbd.
- vvfat: permite exportar sistemas de arquivos FAT
- Baseados em cluster:
  - Sheepdog
  - CEPH

qemu-img

- O canivete suíço
- Permite criar, manipular, converter e outras operações em todos os formatos de imagem suportados.

```
# qemu-img create -f qcow2 -o preallocation=metadata vdisk.qcow2 50G
Formatting 'vdisk.qcow2', fmt=qcow2 size=53687091200 encryption=off cluster_size=0
preallocation='metadata'

# qemu-img info -f qcow2 vdisk.qcow2
image: vdisk.qcow2
file format: qcow2
virtual size: 50G (53687091200 bytes)
disk size: 7.9M
cluster_size: 65536
```

- Convertendo imagem raw para qcow2

```
# qemu-img convert -O qcow2 imagem.raw novaimagem.qcow2
```

Comando

```
# kvm -drive file=vdisk.qcow2,format=qcow2,media=disk,index=0,boot=on,if=virtio
# kvm -drive file=/dev/volumes/zacarias,format=raw,media=disk,index=0,boot=on,if=ide
```

Integridade de dados no QEMU - caching

- 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

Integridade de dados - writethrough

- Este modo é o mais seguro
- Não há mais caches volátil no host
- É lento

Integridade de dados - writeback

- Quando o guest escreve dados, eles simplesmente são colocados no pagecache host
- Não há garantia de que ele realmente foi para o disco
  - O que é realmente muito semelhante à forma como os discos modernos funcionam
- O guest deverá emitir um comando flush para se certificar que os dados foram para o disco
  - Similar a discos reais modernos com cache de writeback
- E o host precisa implementar o comando flush e anunciá-lo
  - IDE e VirtIO só muito recentemente receberam suporte

A integridade dos dados - none

- Transferência direta para o disco deveria implicar que é seguro
- Só que ela não é:
  - Não se garante que caches de disco são liberados
  - Não se dá nenhuma garantia sobre metadados

- Também precisa de um flush cache explícito

Comando

```
# kvm -drive
file=vdisk.qcow2,format=qcow2,media=disk,index=0,boot=on,if=virtio,cache=writeback
```

Administração

Ferramentas utilizadas normalmente para administração de volumes, partições e sistemas de arquivos podem e devem ser utilizadas para a administração e manutenção de armazenamento virtual.

Modelos de armazenamento

- LVM
  - Cada LV possui um disco virtual de uma VM
- Partições
  - Pode-se exportar uma partição toda para uma VM
- Arquivos de imagem

Backend de armazenamento

- Centralizado em um storage
  - iSCSI
  - Fibre Channel
  - AoE
- Qualquer dispositivo de bloco que esteja acessível no host

Backend de armazenamento

- Filesystem compartilhado
  - NFS
  - GFS
  - OCFS

Ferramentas

- kpartx, fdisk, gparted, lvscan, fsck, etc.
- dstat, iotop, htop

Schedullers

- O Linux oferece quatro schedulers de I/O, também conhecido como elevators:
  - Noop
  - Anticipatory
  - Completely Fair Queuing (CFQ)
    - Divide a banda de I/O disponível, mantendo filas de requisições por processo.
  - Deadline
    - Submete requisições de I/O baseando-se no tempo em que estão esperando na fila, garantido um

início de serviço.
- Cada agendador é eficaz em um cenário

Trocando o scheduller

```
 # cat /sys/block/sda/queue/scheduler
noop deadline [cfq]

# echo deadline > /sys/block/sda/queue/scheduler

# cat /sys/block/sdb/queue/scheduler
noop [deadline] cfq
```

- Colocar como parâmetro de boot elevator=deadline
- Quando CFQ, usar ionice para dar prioridade diferenciada

Material de referência

- The KVM/gemu Storage Stack: http://events.linuxfoundation.org/images/stories/slides/jls09/jls09_hellwig.odp
- I/O Schedullers: http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=/liaat/liaatbpscheduleroverview.htm
- http://duartes.org/gustavo/blog/post/page-cache-the-affair-between-memory-and-files
- http://www.ibm.com/developerworks/linux/library/l-virtio/

Base images e rebasing

- http://www.linux-kvm.com/content/how-you-can-use-qemukvm-base-images-be-more-productive-part-1
- http://www.linux-kvm.com/content/be-more-productive-base-images-part-2
- http://www.linux-kvm.com/content/be-more-productive-base-images-part-3
- http://outlyer.net/howtos-linux/kvm-qemu-notes/
- http://kacper.blog.redpill-linpro.com/archives/82

Benchmarks

- KVM I/O slowness on RHEL 6: http://www.ilsistemista.net/index.php/virtualization/11-kvm-io-slowness-on-rhel-6.html?showall=1
- Benchmark de HDs: http://techreport.com/articles.x/20562
- IO Containment: http://www.kernel.org/doc/ols/2008/ols2008v1-pages-151-162.pdf
- VM I/O benchmarks: http://learnitwithme.com/?p=198
- Rackspace Cloud Servers versus Amazon EC2: Performance Analysis: http://www.thebitsource.com/featured-posts/rackspace-cloud-servers-versus-amazon-ec2-performance-analysis/

Material para se aprofundar

- Which I/O controller is the fairest of them all?: http://lwn.net/Articles/332839/
- I/O Topology: http://www.kernel.org/doc/ols/2009/ols2009-pages-235-238.pdf
- The 2010 Linux Storage and Filesystem Summit, day 1: http://lwn.net/Articles/399148/

- The 2010 Linux Storage and Filesystem Summit, day 2: http://lwn.net/Articles/399313/
- I Will Keep Saying It: Align Your Partitions: http://lonesysadmin.net/2010/03/30/i-will-keep-saying-it-align-your-partitions/
- VMware I/O Problems: http://lonesysadmin.net/2006/05/20/vmware-io-problems/
- Controle de I/O com cgroups e libvirt: http://comments.gmane.org/gmane.comp.emulators.libvirt/32332
- High Performance VMM-Bypass I/O in Virtual Machines: http://www.cc.gatech.edu/classes/AY2007/cs8803hpc_fall/papers/dk-vmmio.pdf