libvirt-qemu-kvm-virtio

xiaofei

2015.05.19

Agenda

- Preview
- Libvirt Introduction
- Qemu-KVM Introduction
- VirtIO

Topic

- Preview
- Libvirt Introduction
- Qemu-KVM Introduction
- VirtIO

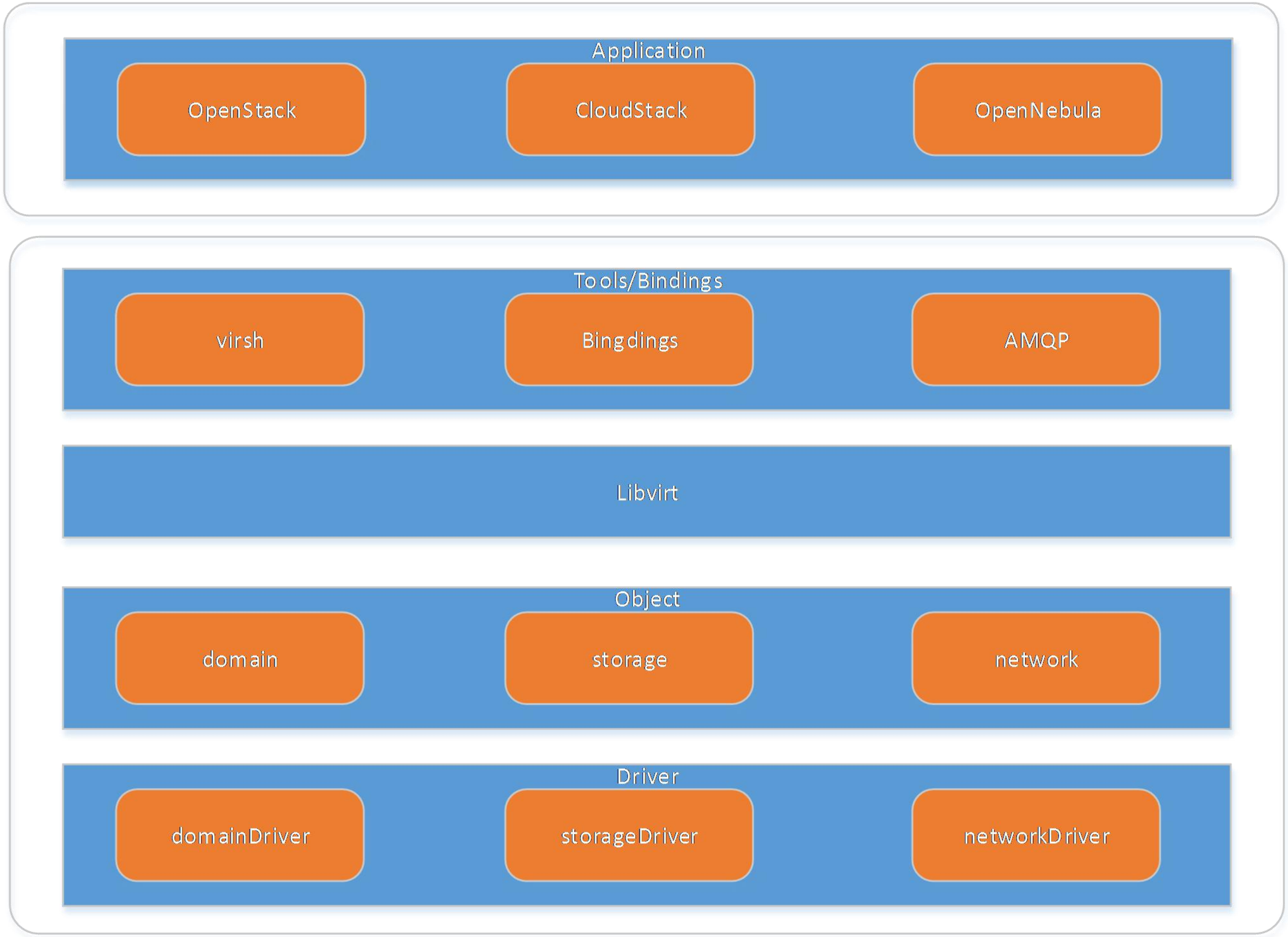
Preview

- Qemu
  - emulator, both for CPU and hardware
  - instructions relay
- KVM
  - kernel module
  - hardware assisted para-virtualization
  - translate guest CPU instructions directly
- Qemu-kvm
  - ioctl /dev/kvm
  - offload CPU instructions part to KVM
  - virtio
- Libvirt
  - virtualization tool

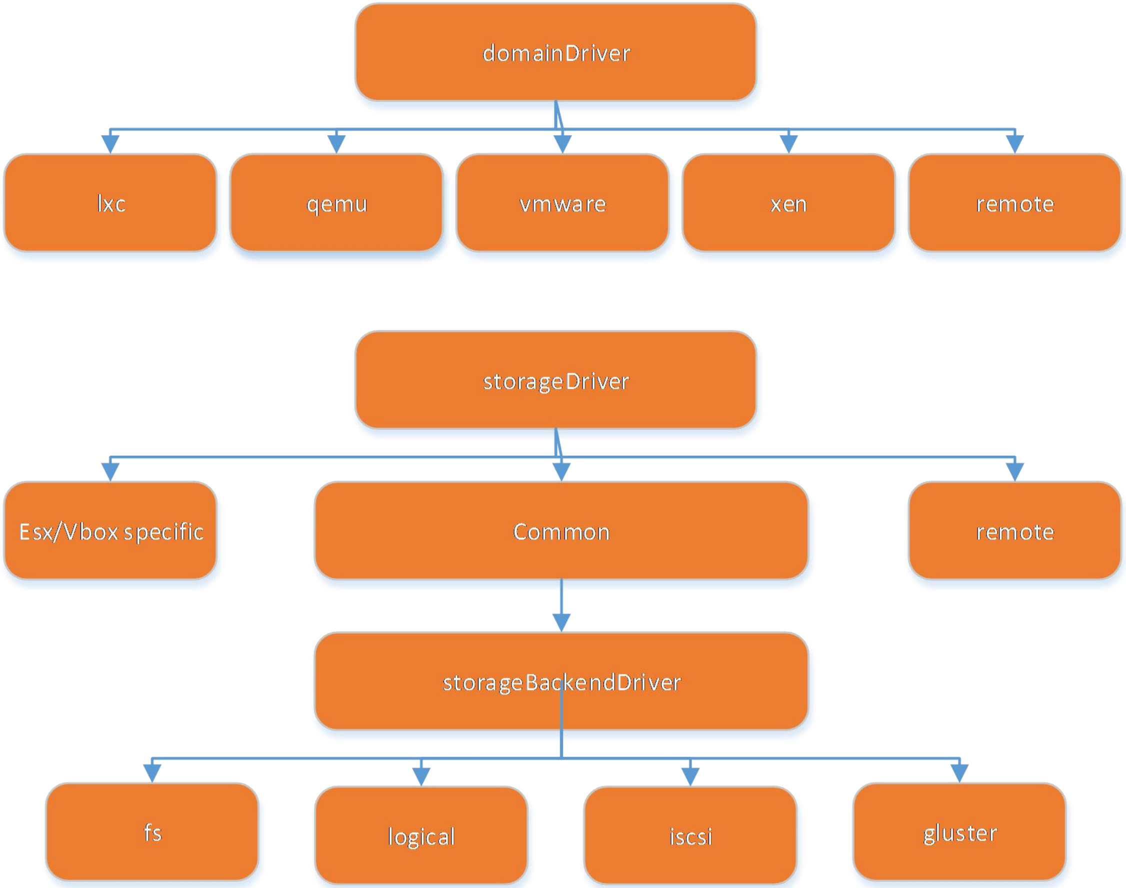
Topic

- Preview
- Libvirt Introduction
- Qemu-KVM Introduction
- VirtIO

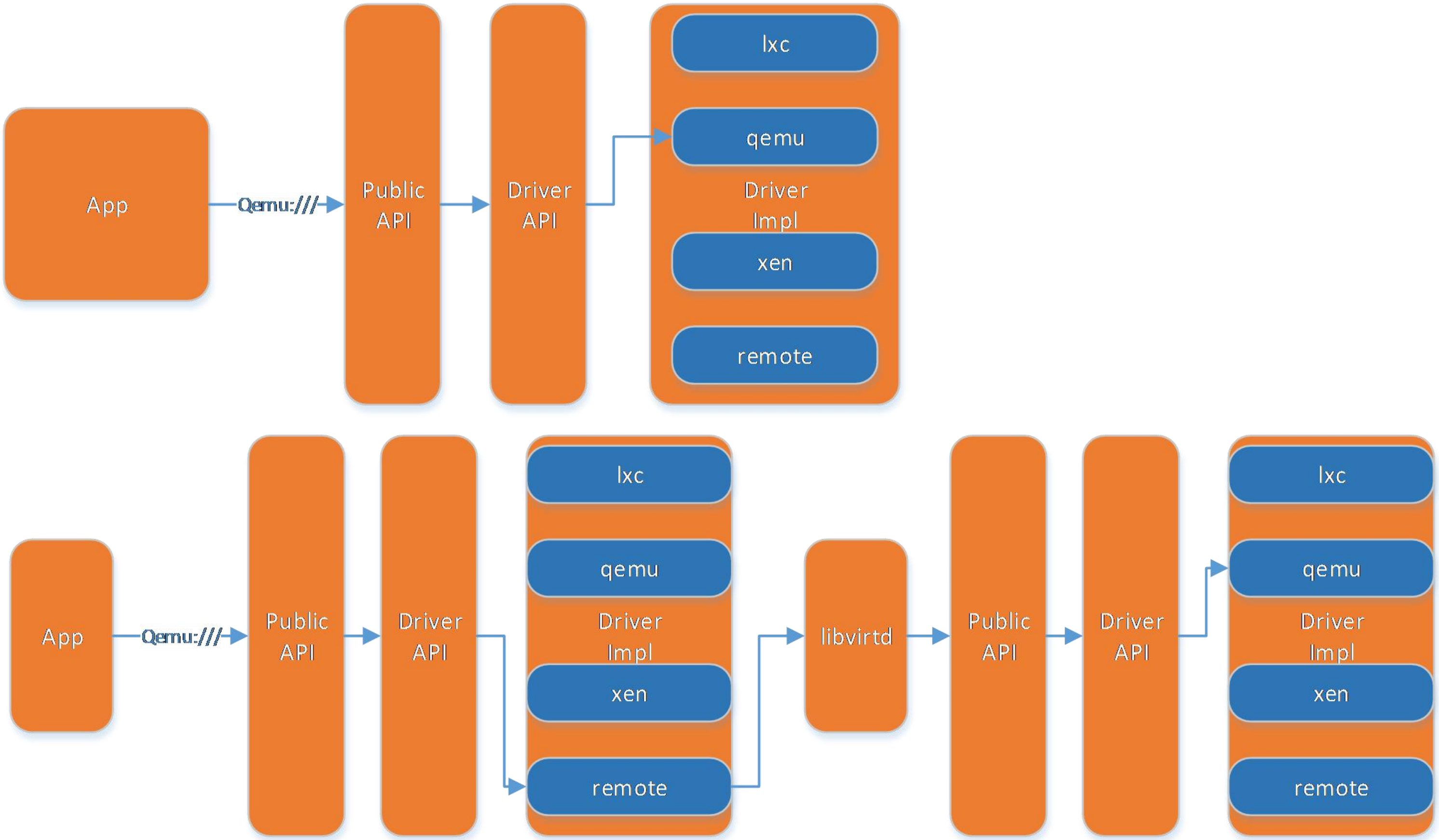
Libvirt stack



Libvirt Driver-based Architecture

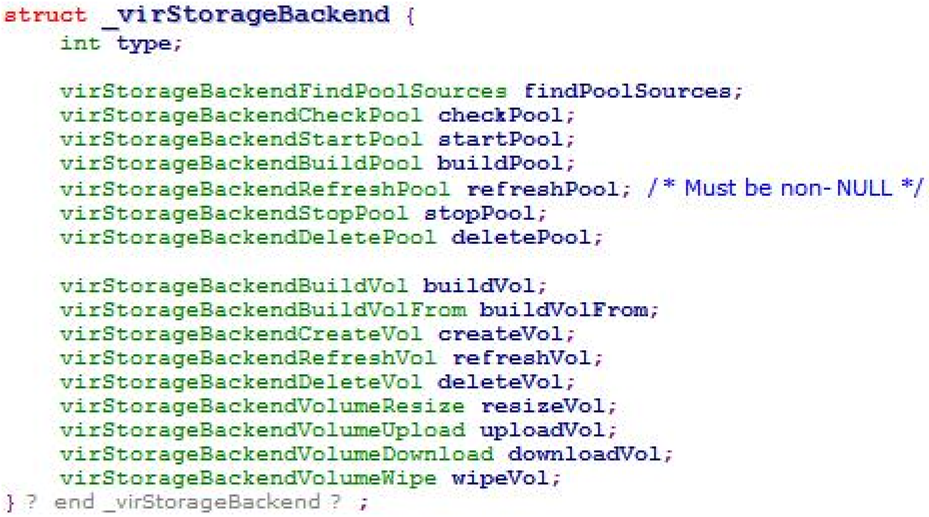


Libvirt Domain management

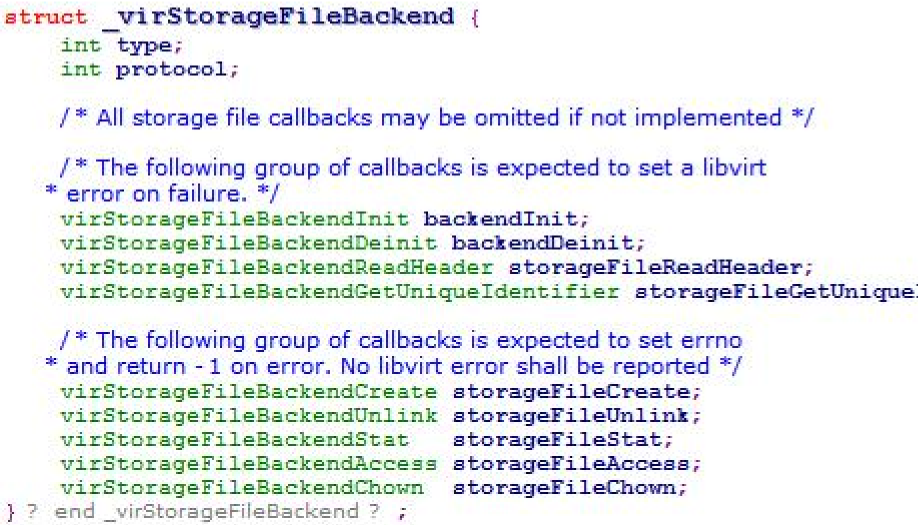


Libvirt Storage Backend

- VirStorageBackend
- VirStorageFileBackend



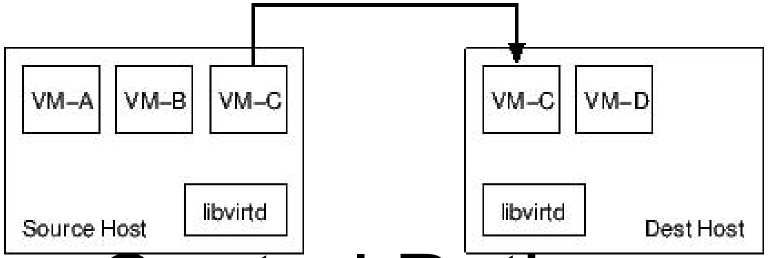
```
struct _virStorageBackend {
    int type;

    virStorageBackendFindPoolSources findPoolSources;
    virStorageBackendCheckPool checkPool;
    virStorageBackendStartPool startPool;
    virStorageBackendBuildPool buildPool;
    virStorageBackendRefreshPool refreshPool; /* Must be non-NULL */
    virStorageBackendStopPool stopPool;
    virStorageBackendDeletePool deletePool;

    virStorageBackendBuildVol buildVol;
    virStorageBackendBuildVolFrom buildVolFrom;
    virStorageBackendCreateVol createVol;
    virStorageBackendRefreshVol refreshVol;
    virStorageBackendDeleteVol deleteVol;
    virStorageBackendVolumeResize resizeVol;
    virStorageBackendVolumeUpload uploadVol;
    virStorageBackendVolumeDownload downloadVol;
    virStorageBackendVolumeWipe wipeVol;
} ? end _virStorageBackend ? ;
```

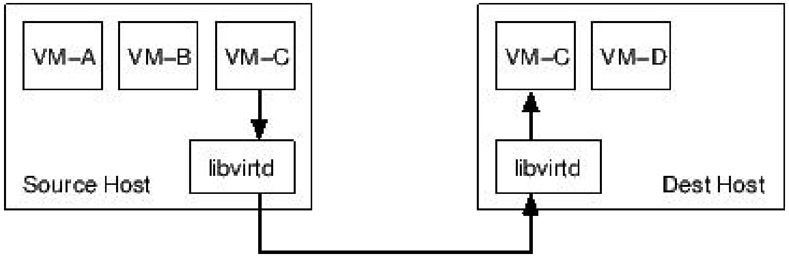


```
struct _virStorageFileBackend {
    int type;
    int protocol;

    /* All storage file callbacks may be omitted if not implemented */

    /* The following group of callbacks is expected to set a libvirt
   * error on failure. */
    virStorageFileBackendInit backendInit;
    virStorageFileBackendDeinit backendDeinit;
    virStorageFileBackendReadHeader storageFileReadHeader;
    virStorageFileBackendGetUniqueIdentifier storageFileGetUniqueI

    /* The following group of callbacks is expected to set errno
   * and return -1 on error. No libvirt error shall be reported */
    virStorageFileBackendCreate storageFileCreate;
    virStorageFileBackendUnlink storageFileUnlink;
    virStorageFileBackendStat   storageFileStat;
    virStorageFileBackendAccess storageFileAccess;
    virStorageFileBackendChown  storageFileChown;
} ? end _virStorageFileBackend ? ;
```

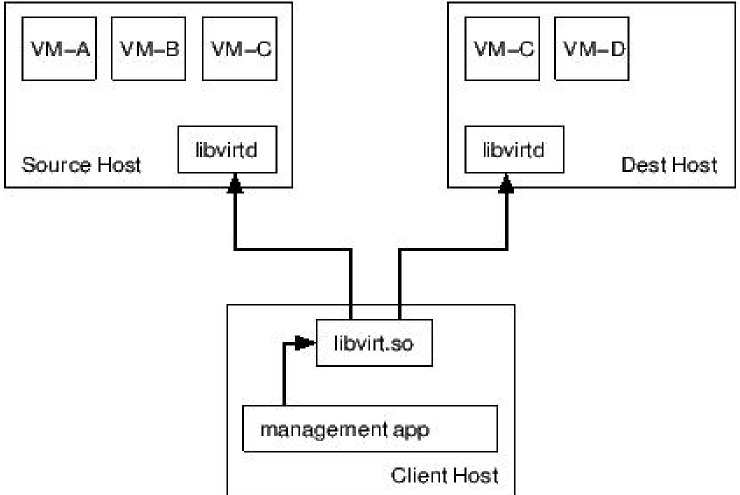
Libvirt migration

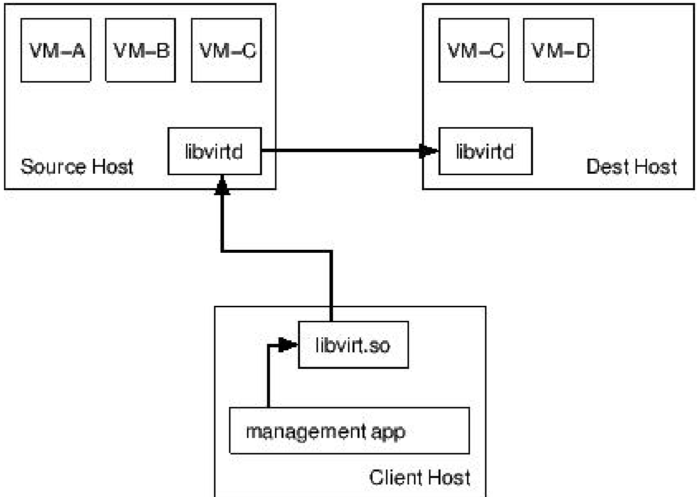
- Data Path
  - Native vs tunnelled





- Control Path
  - Direct vs P2P





Topic

- Preview
- Libvirt Introduction
- Qemu-KVM Introduction
- VirtIO

qemu-kvm hardware vitualization

- Qemu emulation
  - Architecture
  - **CPU**
  - **Network**
  - **Disk**
  - **Memory**
  - **SMP**
  - System Management BIOS
  - System Clock
  - USB
  - BUS
  - Monitor
  - Sound
  - CD-ROM

qemu-kvm hardware vitualization(2)

Openstack VM paras

```
qemu-system-x86_64
-enable-kvm
-name instance-00000024
-machine pc-i440fx-trusty,accel=kvm,usb=off
-cpu SandyBridge,+erms,+smep,+fsgsbase,+pdpe1gb,+rdrand,+f16c,+osxsave,+dca,+pcid,+pdcm,+xtpr,+tm2,+est,+smx,+vmx,+ds_cpl,+monitor,+dtes64,+pbe,+tm,+ht,+ss,
+acpi,+ds,+vme
-m 2048 -realtime mlock=off
-smp 1,sockets=1,cores=1,threads=1
-uuid 1f8e6f7e-5a70-4780-89c1-464dc0e7f308
-smbios type=1,manufacturer=OpenStack Foundation,product=OpenStack Nova,version=2014.1,serial=80590690-87d2-e311-b1b0-a0481cabdfb4,uuid=1f8e6f7e-5a70-4780-89c1-464dc0e7f308
-no-user-config
-nodefaults
-chardev socket,id=charmonitor,path=/var/lib/libvirt/qemu/instance-00000024.monitor,server,nowait
-mon chardev=charmonitor,id=monitor,mode=control
-rtc base=utc,driftfix=slew
-global kvm-pit.lost_tick_policy=discard
-no-hpet
-no-shutdown
-boot strict=on
-device piix3-usb-uhci,id=usb,bus=pci.0,addr=0x1.0x2
-drive file=/var/lib/nova/instances/1f8e6f7e-5a70-4780-89c1-464dc0e7f308/disk,if=none,id=drive-virtio-disk0,format=qcow2,cache=none
-device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=drive-virtio-disk0,id=virtio-disk0,bootindex=1
-netdev tap,fd=32,id=hostnet0,vhost=on,vhostfd=37
-device virtio-net-pci,netdev=hostnet0,id=net0,mac=fa:16:3e:d1:2d:99,bus=pci.0,addr=0x3
-chardev file,id=charserial0,path=/var/lib/nova/instances/1f8e6f7e-5a70-4780-89c1-464dc0e7f308/console.log
-device isa-serial,chardev=charserial0,id=serial0
-chardev pty,id=charserial1
-device isa-serial,chardev=charserial1,id=serial1
-device usb-tablet,id=input0
-vnc 0.0.0.0:12
-k en-us
-device cirrus-vga,id=video0,bus=pci.0,addr=0x2
-device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x5
```

qemu-kvm hardware vitualization(3)

- Architecture emulation

– PC (x86 or x86_64 processor)

– Mac99 PowerMac (PowerPC processor)

– Sun4u/Sun4v (64-bit Sparc processor)

– MIPS magnum (64-bit MIPS processor)

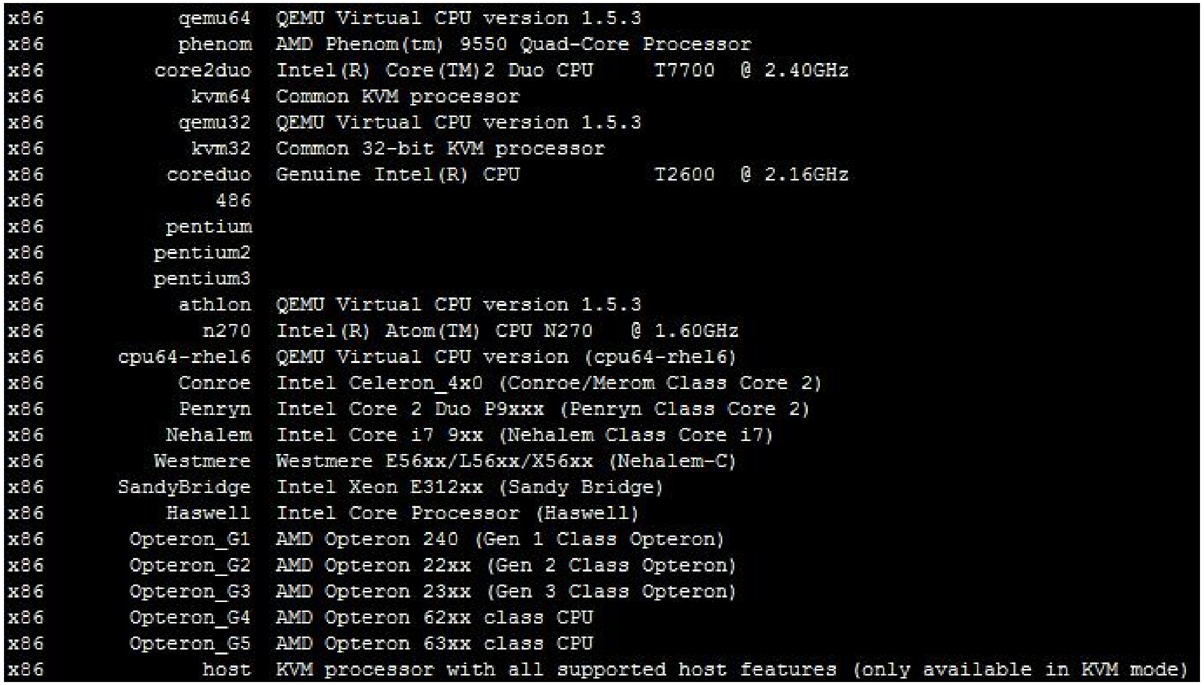
- -accel

– accel=kvm, hardware-assisted virtualization

– accel = tcg , -no-kvm, without hardware-assisted virtualization

qemu-kvm hardware vitualization(4)

- ACPI
- CPU
  - /usr/libexec/qemu-kvm -cpu help



```
x86           qemu64  QEMU Virtual CPU version 1.5.3
x86           phenom  AMD Phenom(tm) 9550 Quad-Core Processor
x86         core2duo  Intel(R) Core(TM)2 Duo CPU     T7700  @ 2.40GHz
x86            kvm64  Common KVM processor
x86           qemu32  QEMU Virtual CPU version 1.5.3
x86            kvm32  Common 32-bit KVM processor
x86          coreduo  Genuine Intel(R) CPU           T2600  @ 2.16GHz
x86              486
x86          pentium
x86         pentium2
x86         pentium3
x86           athlon  QEMU Virtual CPU version 1.5.3
x86             n270  Intel(R) Atom(TM) CPU N270   @ 1.60GHz
x86      cpu64-rhel6  QEMU Virtual CPU version (cpu64-rhel6)
x86           Conroe  Intel Celeron_4x0 (Conroe/Merom Class Core 2)
x86           Penryn  Intel Core 2 Duo P9xxx (Penryn Class Core 2)
x86          Nehalem  Intel Core i7 9xx (Nehalem Class Core i7)
x86         Westmere  Westmere E56xx/L56xx/X56xx (Nehalem-C)
x86      SandyBridge  Intel Xeon E312xx (Sandy Bridge)
x86          Haswell  Intel Core Processor (Haswell)
x86       Opteron_G1  AMD Opteron 240 (Gen 1 Class Opteron)
x86       Opteron_G2  AMD Opteron 22xx (Gen 2 Class Opteron)
x86       Opteron_G3  AMD Opteron 23xx (Gen 3 Class Opteron)
x86       Opteron_G4  AMD Opteron 62xx class CPU
x86       Opteron_G5  AMD Opteron 63xx class CPU
x86             host  KVM processor with all supported host features (only available in KVM mode)
```

qemu-kvm hardware vitualization(5)

- SMP
  - qemu-kvm supports at most 255 CPU
  - -smp 1,sockets=1,cores=1,threads=1
    - smp
    - sockets
    - cores
    - threads

- RAM
  - -m 2048
  - -device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x5
  - Memory Ballooning

qemu-kvm hardware vitualization(6)

- Network
  - -netdev tap,fd=32,id=**hostnet0**,vhost=on,vhostfd=37
  - -device virtio-net-pci,netdev=**hostnet0**,id=net0,mac=fa:16:3e:d1:2d:99,bus=pci.0,addr=0x3

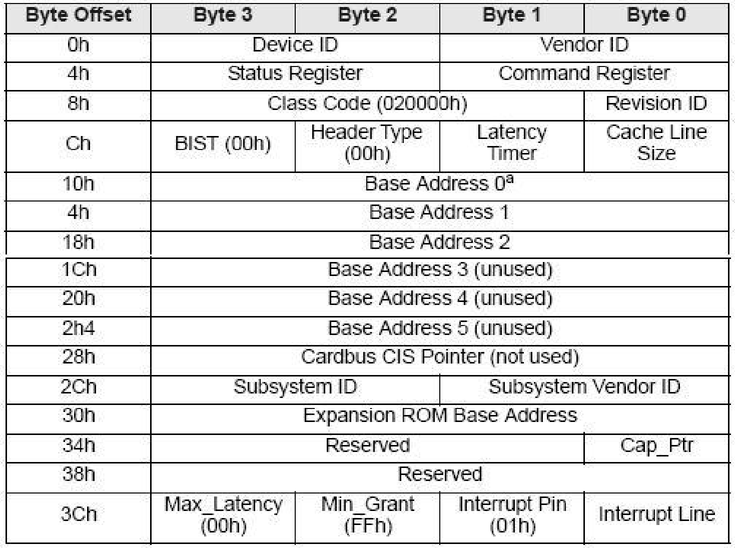
- Drive
  - -drive file=/var/lib/nova/instances/1f8e6f7e-5a70-4780-89c1-464dc0e7f308/disk,if=none,id=**drive-virtio-disk0**,format=qcow2,cache=none
  - -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=**drive-virtio-disk0**,id=virtio-disk0,bootindex=1

qemu-kvm hardware vitualization(7)

- PCI
  - PCI address
    - bus
    - slot
    - function
  - PCI configuration space
    - for PnP
  - PCI memory space and IO space
  - PCI Interrupt
    - INTx, MSI, MSI-X

qemu-kvm hardware vitualization(8)

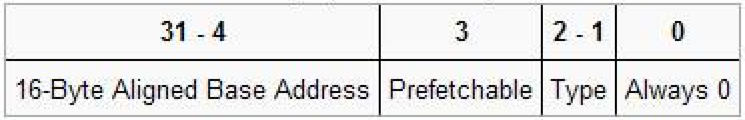
- PCI configuration space



Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0h	Device ID		Vendor ID	
4h	Status Register		Command Register	
8h	Class Code (020000h)			Revision ID
Ch	BIST (00h)	Header Type (00h)	Latency Timer	Cache Line Size
10h	Base Address 0[a]			
4h	Base Address 1			
18h	Base Address 2			
1Ch	Base Address 3 (unused)			
20h	Base Address 4 (unused)			
2h4	Base Address 5 (unused)			
28h	Cardbus CIS Pointer (not used)			
2Ch	Subsystem ID		Subsystem Vendor ID	
30h	Expansion ROM Base Address			
34h	Reserved			Cap_Ptr
38h	Reserved			
3Ch	Max_Latency (00h)	Min_Grant (FFh)	Interrupt Pin (01h)	Interrupt Line

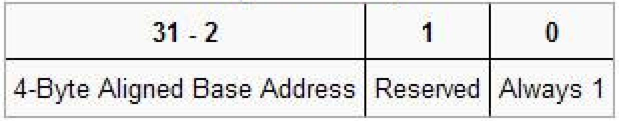
- BAR(Base Address Register)

Memory Space BAR Layout



31 - 4	3	2 - 1	0
16-Byte Aligned Base Address	Prefetchable	Type	Always 0

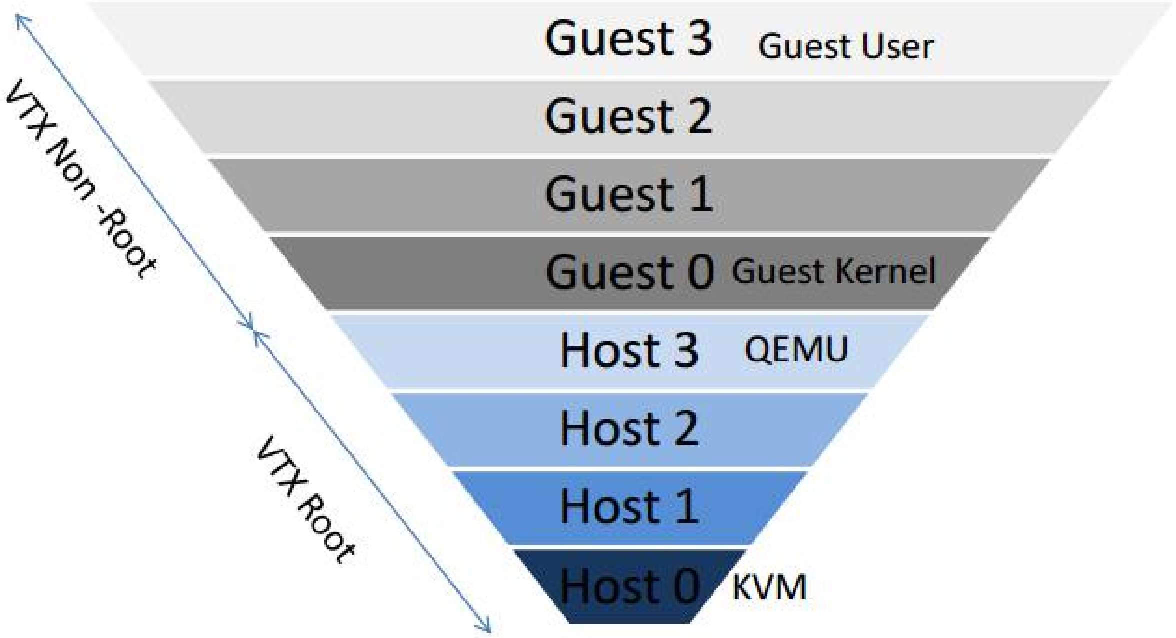
I/O Space BAR Layout



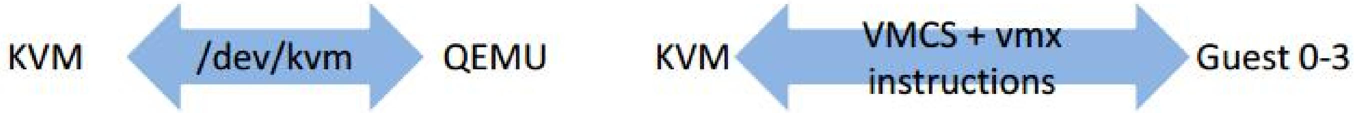
31 - 2	1	0
4-Byte Aligned Base Address	Reserved	Always 1

qemu-kvm hardware vitualization(9)

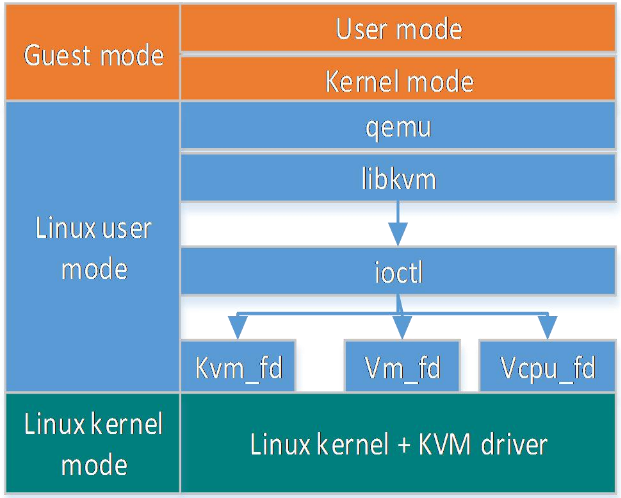
- X86 VTx support



Communication Channels



qemu-kvm hardware vitualization(10)



Guest mode

User mode

Kernel mode

Linux user mode

qemu

libkvm

ioctl

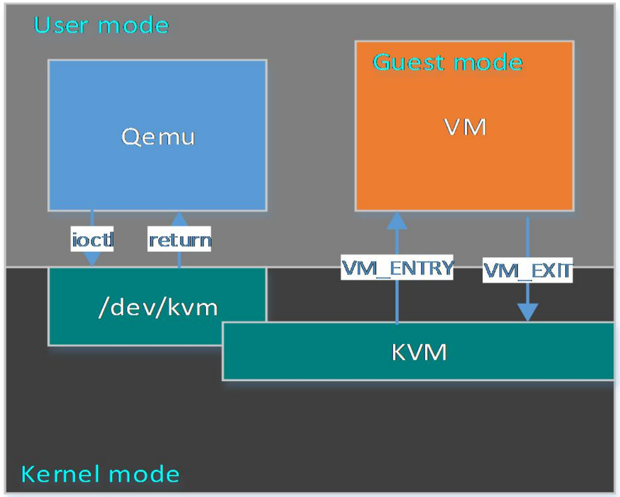
Kvm_fd

Vm_fd

Vcpu_fd

Linux kernel mode

Linux kernel + KVM driver



User mode

Qemu

Guest mode

VM

ioctl

return

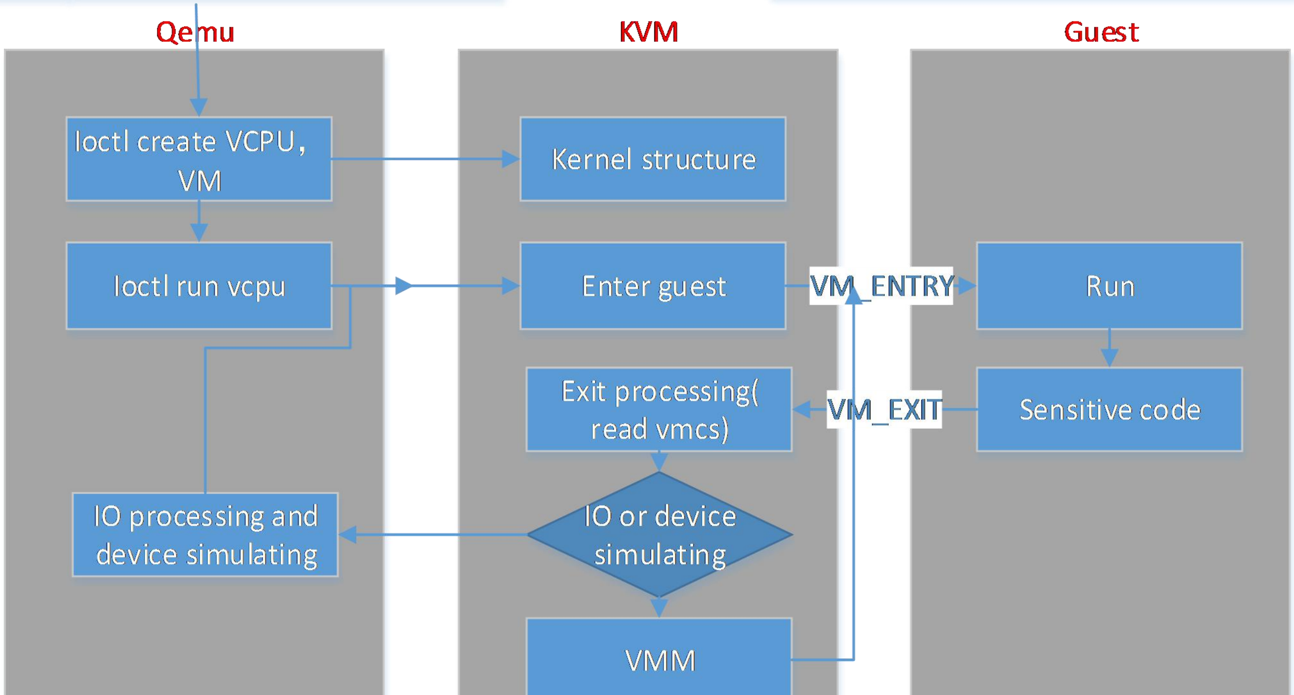
VM_ENTRY

VM_EXIT

/dev/kvm

KVM

Kernel mode



Qemu

KVM

Guest

Ioctl create VCPU, VM

Kernel structure

Ioctl run vcpu

Enter guest

VM_ENTRY

Run

Sensitive code

VM_EXIT

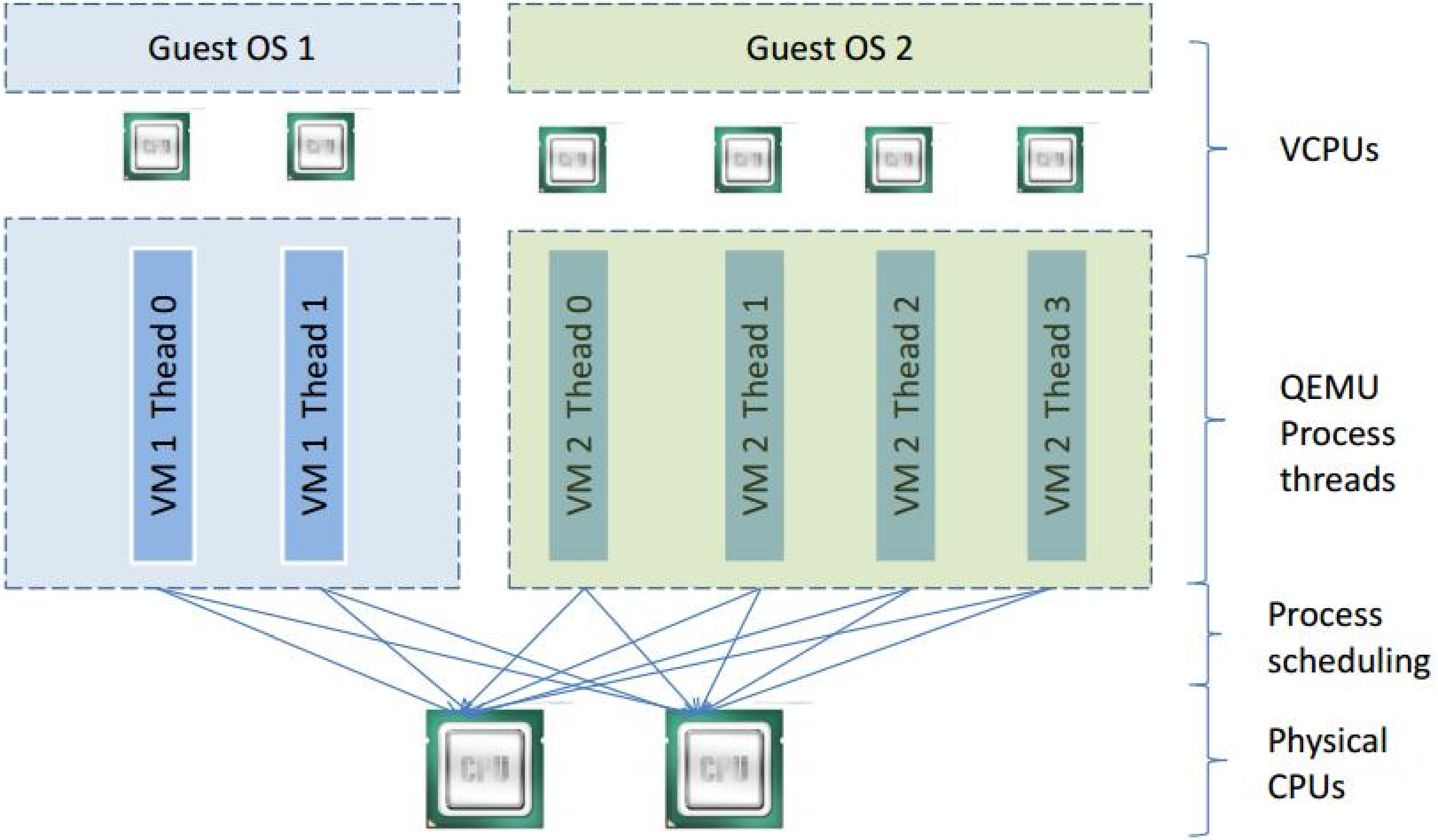
Exit processing(read vmcs)

IO or device simulating

IO processing and device simulating

VMM

CPU Virtualization

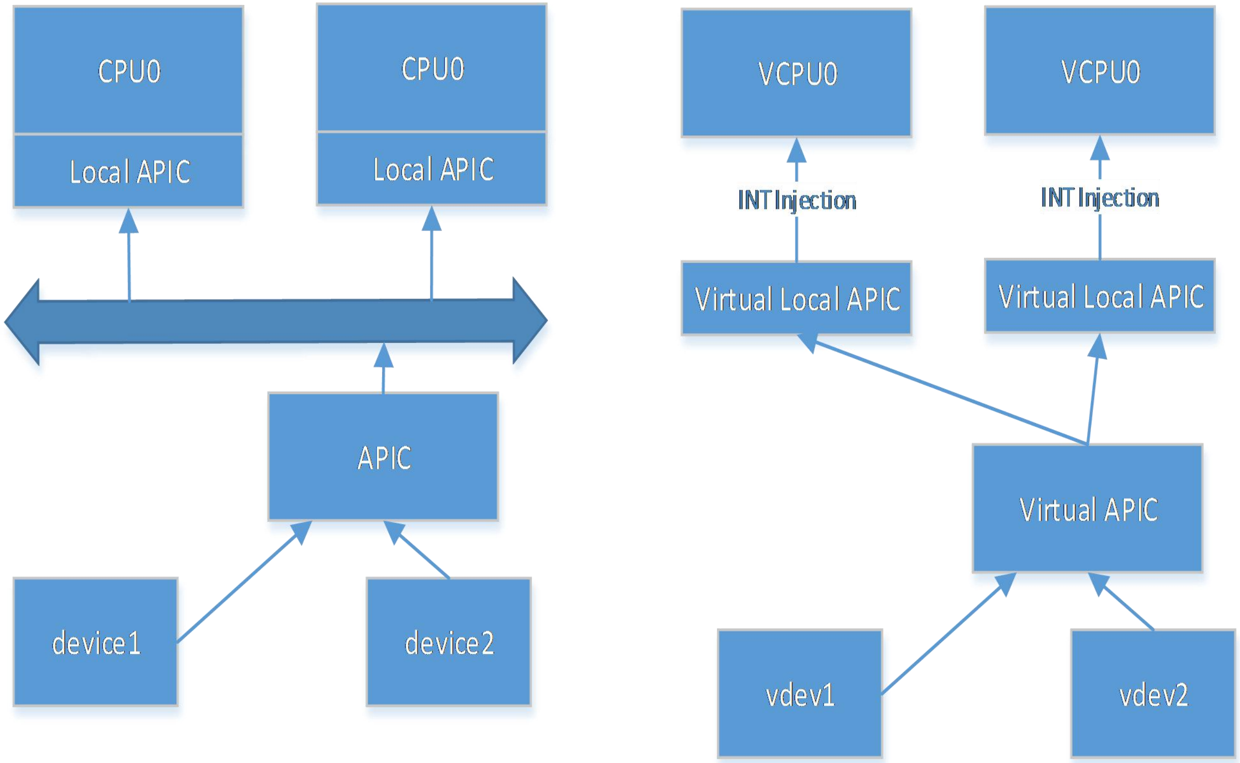


Memory Virtualization

- Problem
  - GVA -> GPA -> HVA -> HPA

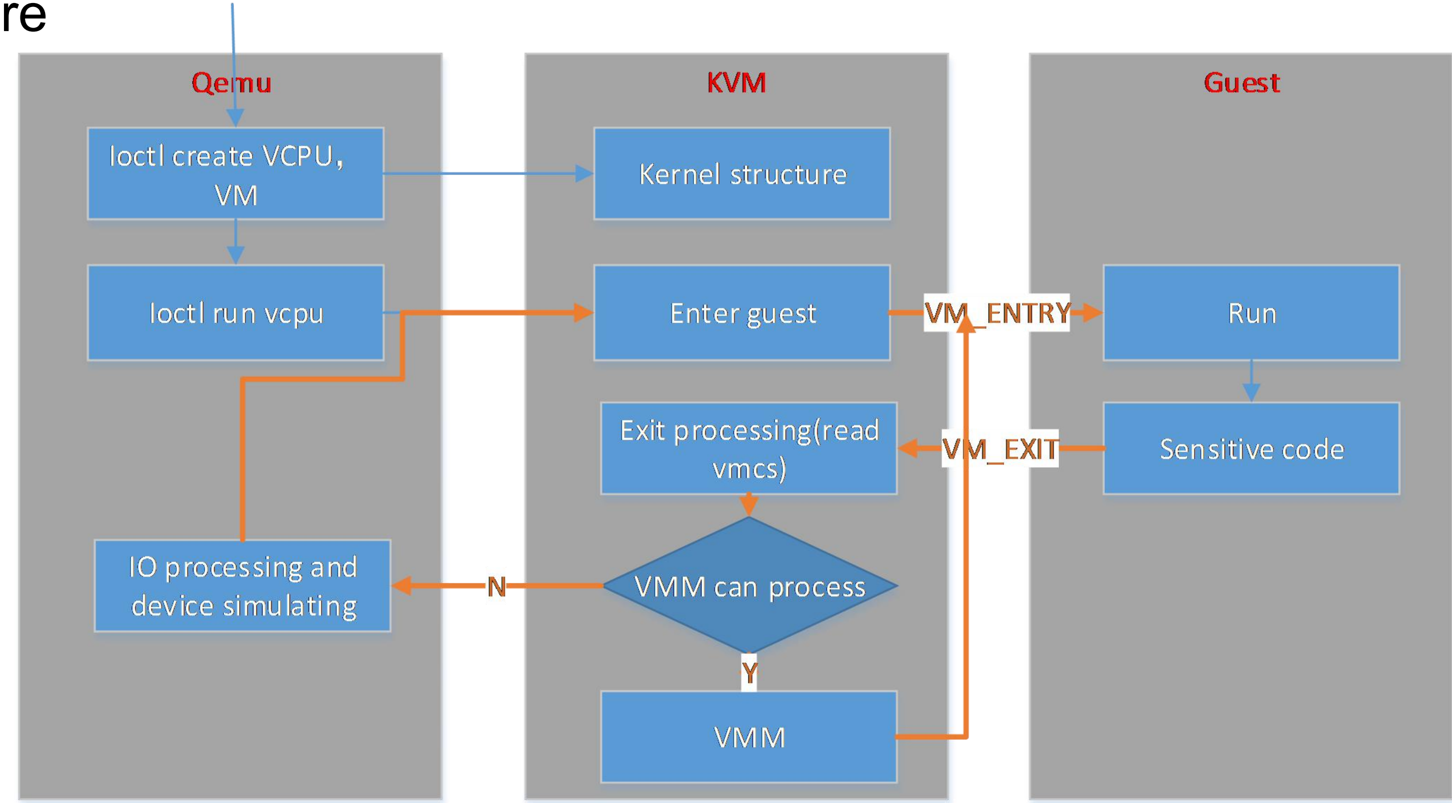
- Solution
  - Shadow page table
  - EPT/NPT hardware support

Interrupt Virtualization



IO Virtualization

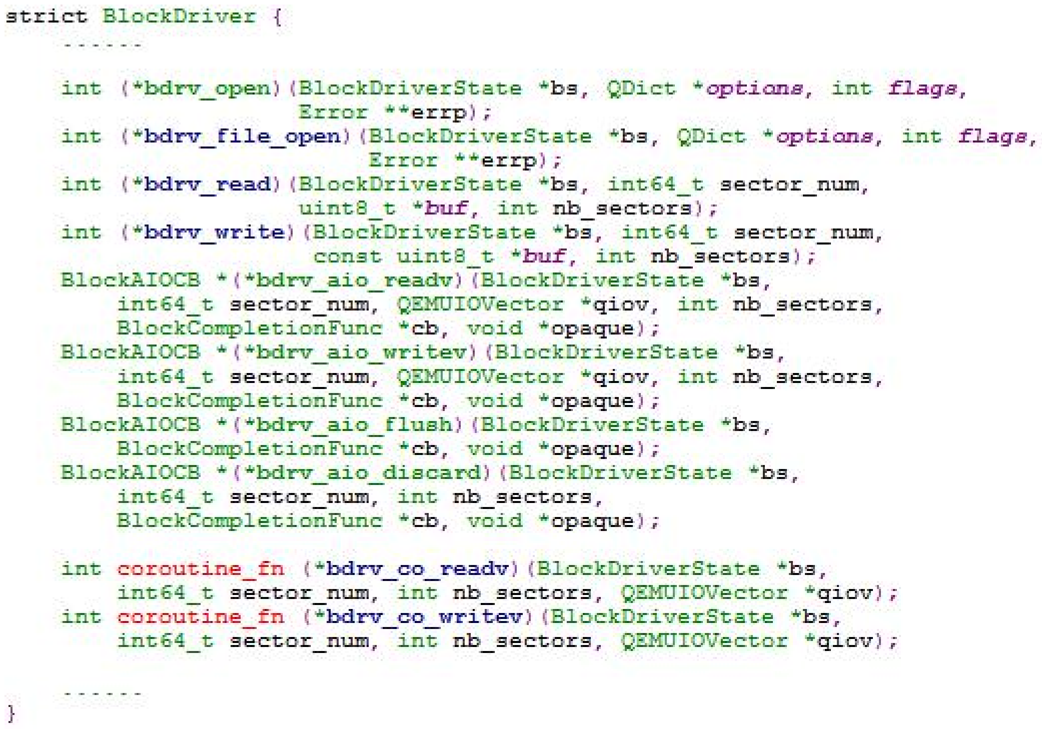
- Pure software



- Hardware based
  – PCI passthrough

Qemu Storage Backend

- BlockDriver



```
strict BlockDriver {
    ------

    int (*bdrv_open)(BlockDriverState *bs, QDict *options, int flags,
                     Error **errp);
    int (*bdrv_file_open)(BlockDriverState *bs, QDict *options, int flags,
                          Error **errp);
    int (*bdrv_read)(BlockDriverState *bs, int64_t sector_num,
                     uint8_t *buf, int nb_sectors);
    int (*bdrv_write)(BlockDriverState *bs, int64_t sector_num,
                      const uint8_t *buf, int nb_sectors);
    BlockAIOCB *(*bdrv_aio_readv)(BlockDriverState *bs,
        int64_t sector_num, QEMUIOVector *qiov, int nb_sectors,
        BlockCompletionFunc *cb, void *opaque);
    BlockAIOCB *(*bdrv_aio_writev)(BlockDriverState *bs,
        int64_t sector_num, QEMUIOVector *qiov, int nb_sectors,
        BlockCompletionFunc *cb, void *opaque);
    BlockAIOCB *(*bdrv_aio_flush)(BlockDriverState *bs,
        BlockCompletionFunc *cb, void *opaque);
    BlockAIOCB *(*bdrv_aio_discard)(BlockDriverState *bs,
        int64_t sector_num, int nb_sectors,
        BlockCompletionFunc *cb, void *opaque);

    int coroutine_fn (*bdrv_co_readv)(BlockDriverState *bs,
        int64_t sector_num, int nb_sectors, QEMUIOVector *qiov);
    int coroutine_fn (*bdrv_co_writev)(BlockDriverState *bs,
        int64_t sector_num, int nb_sectors, QEMUIOVector *qiov);

    ------
}
```

libvirt-qemu iotune

- vish comman
  - blkdeviotune

- Throttle initialization

- Throttle during IO

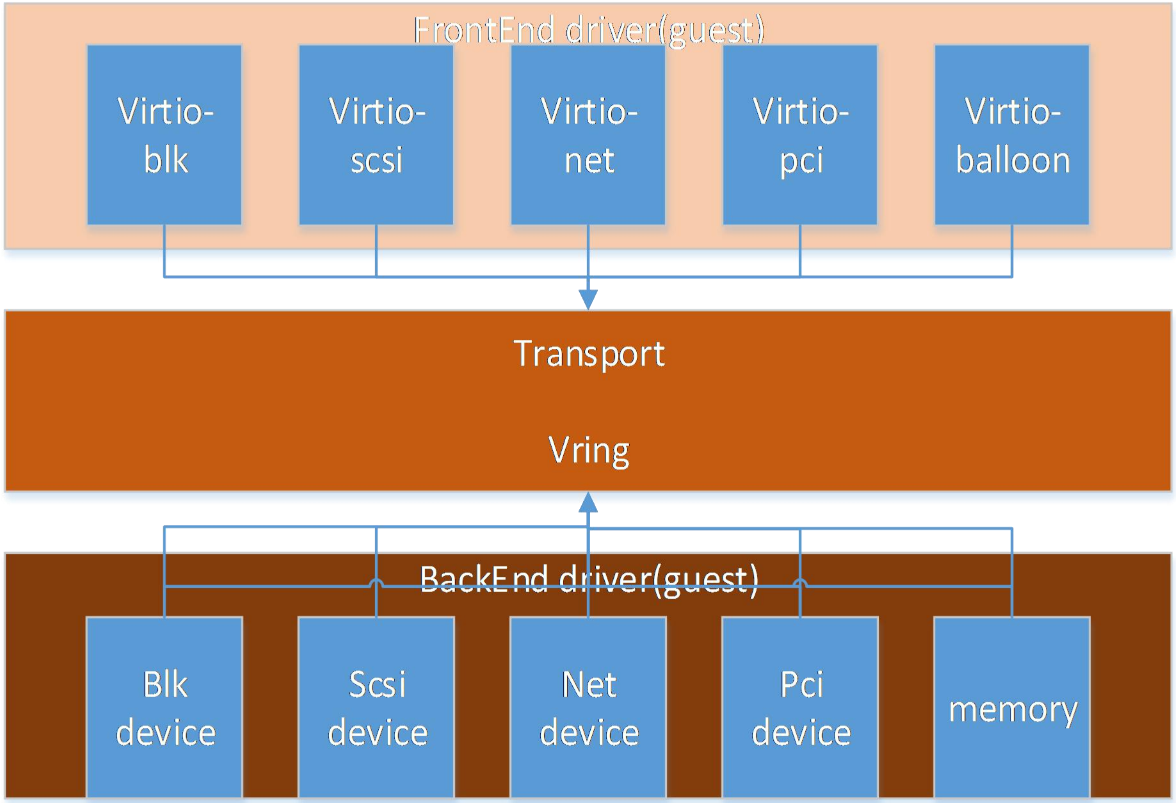
- Timer

- Leaky bucket

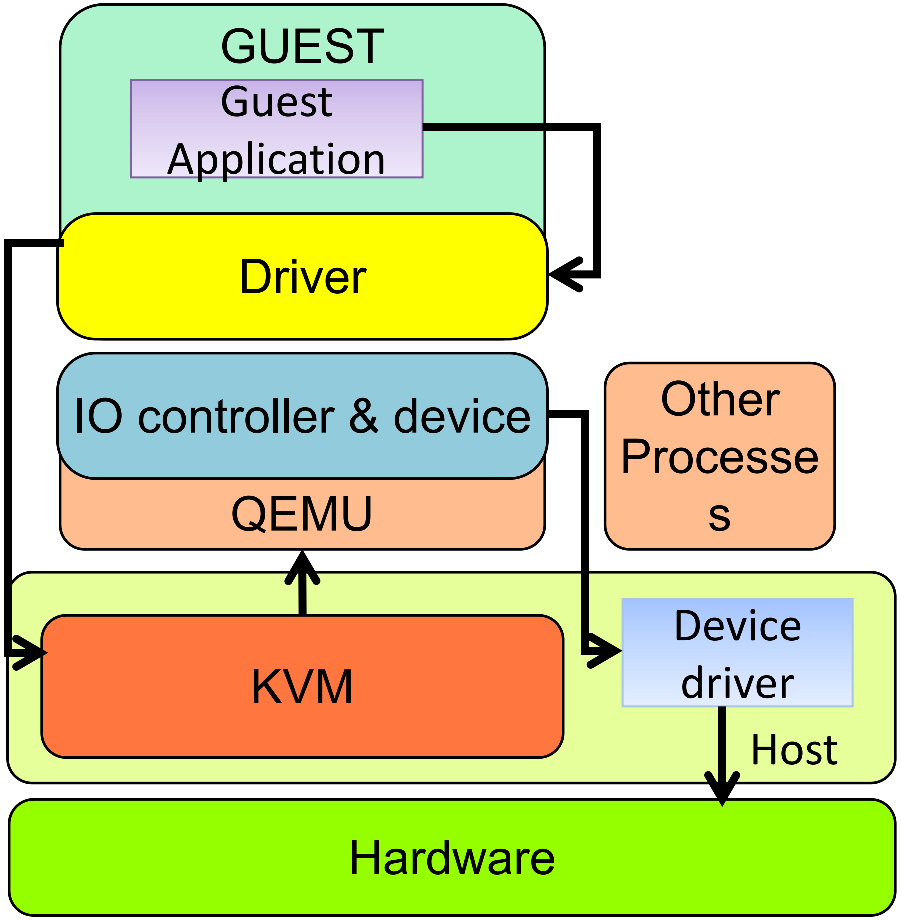
Topic

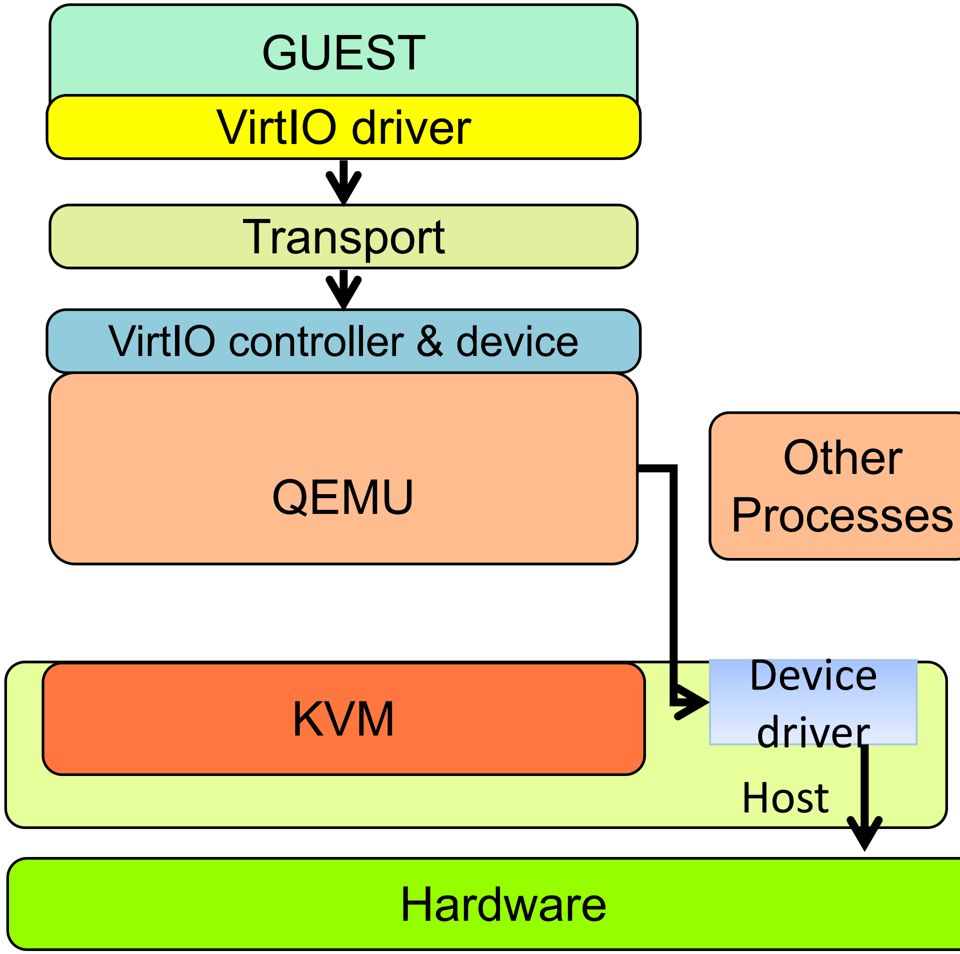
- Preview
- Libvirt Introduction
- Qemu-KVM Introduction
- VirtIO

VirtIO Architecture



KVM without virtio *vs* with virtio

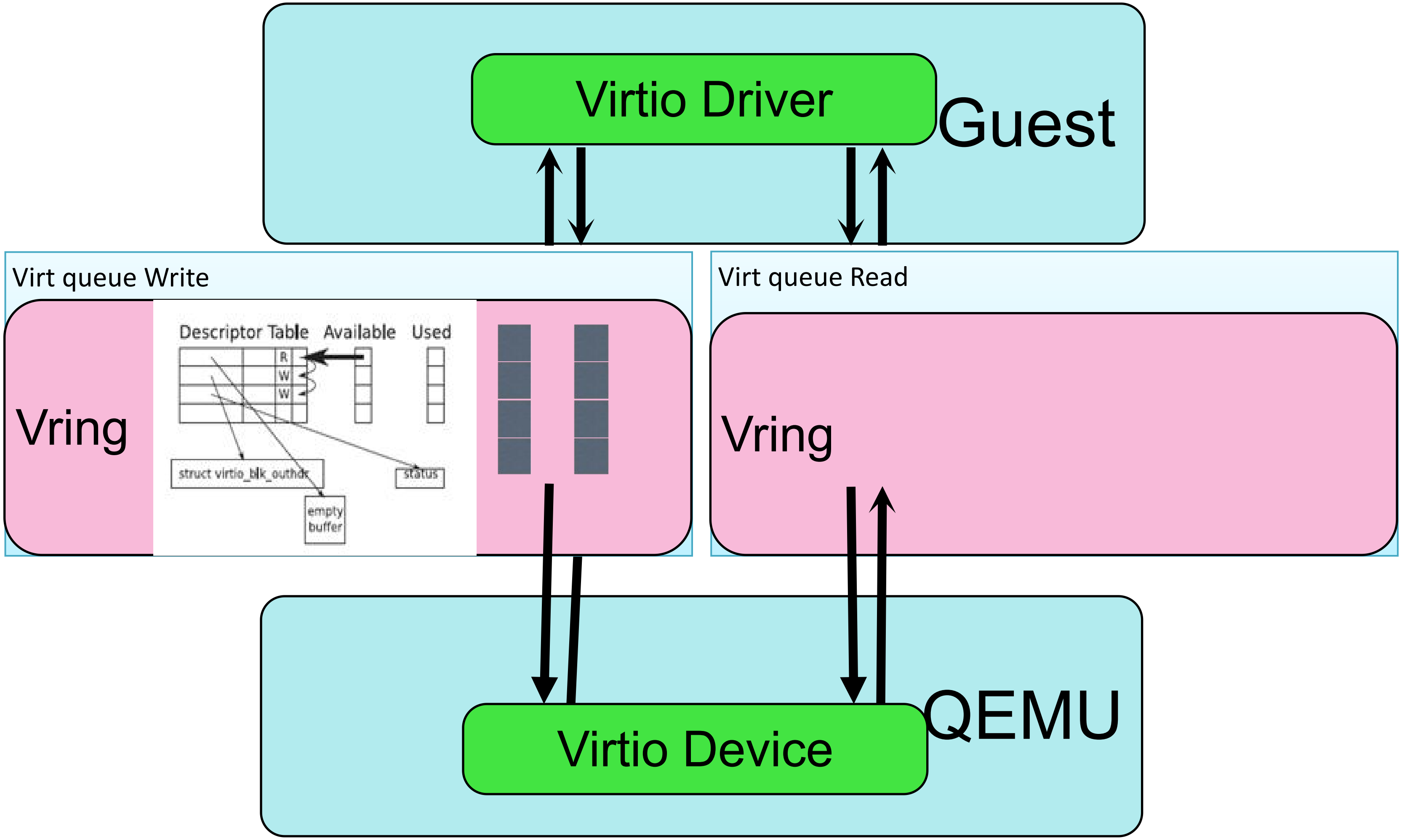




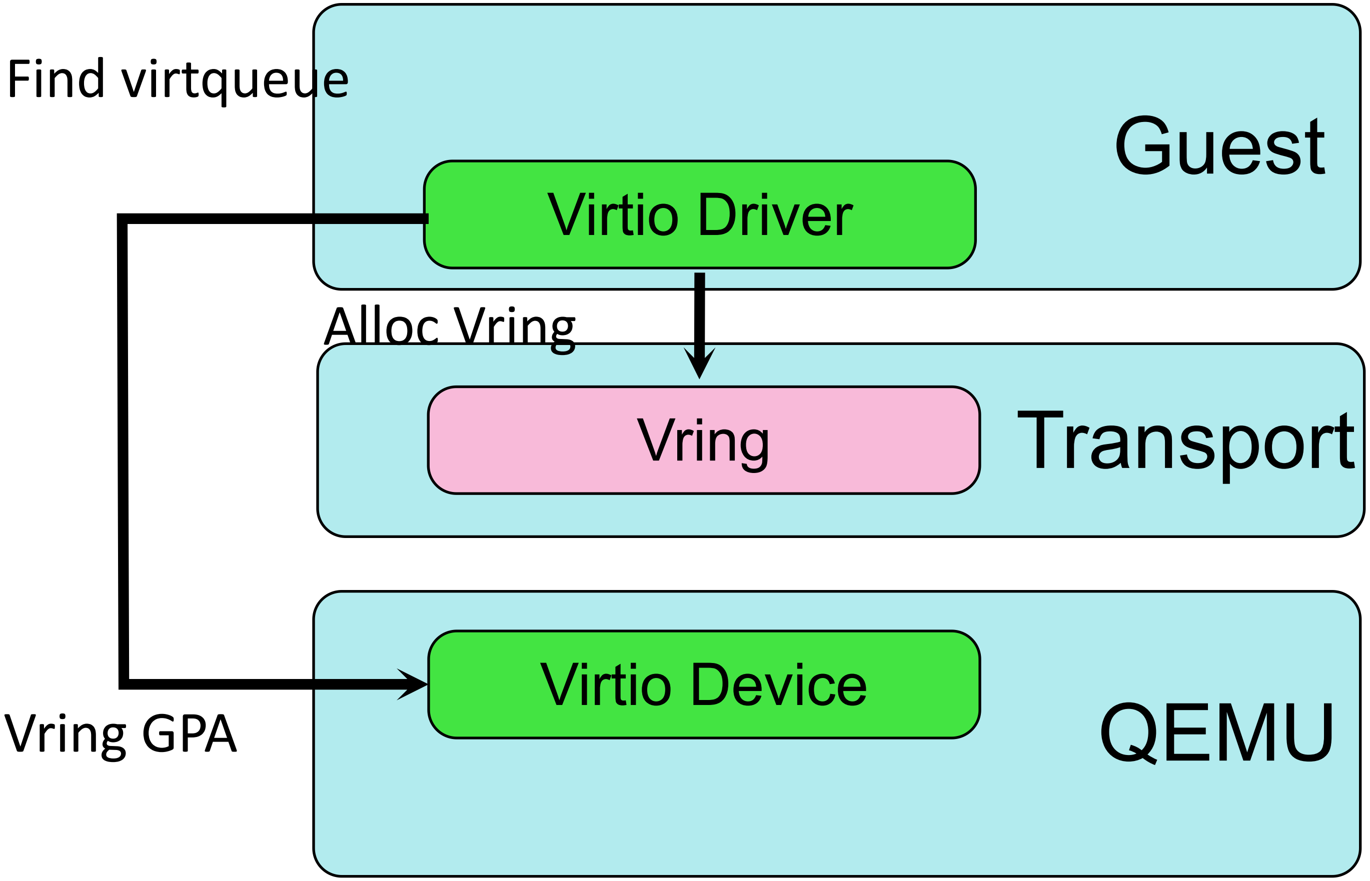
Driver

- Front-end driver
  - A kernel module in guest OS.
  - Accepts I/O requests from user process.
  - Transfer I/O requests to back-end driver.
- Back-end driver
  - A device in QEMU.
  - Accepts I/O requests from front-end driver.
  - Perform I/O operation via physical device.

Vring structure



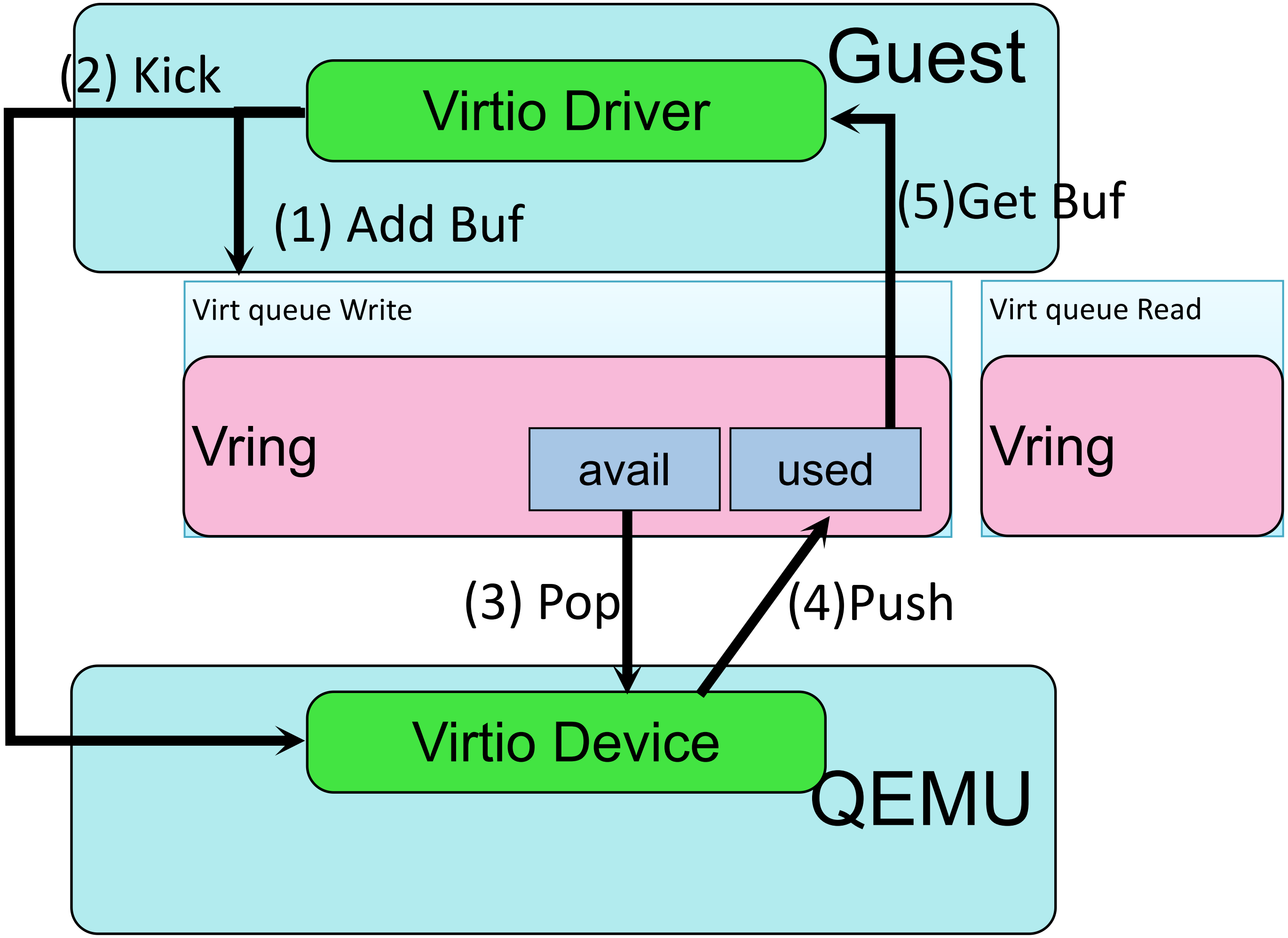
Virtqueue Initialization



VirtIo data exchange API

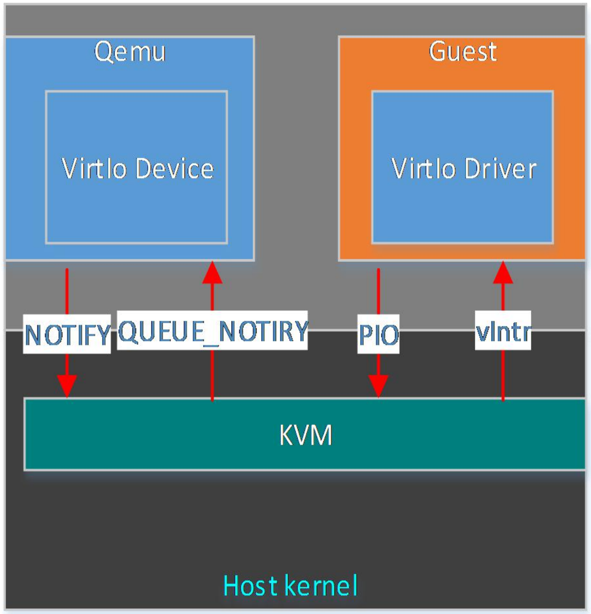
- In guest
  - *virtqueue_add_buf*
    - Expose virtio-buffer to other end
  - *virtqueue_get_buf*
    - Get the results from virtqueue
  - *virtqueue_kick*
    - Update virtqueue after add_buf
    - Notify QEMU to deal with the data
- In QEMU
  - virtqueue_pop
    - Pop the data from virtqueue
  - virtqueue_push
    - Put data back to virtqueue

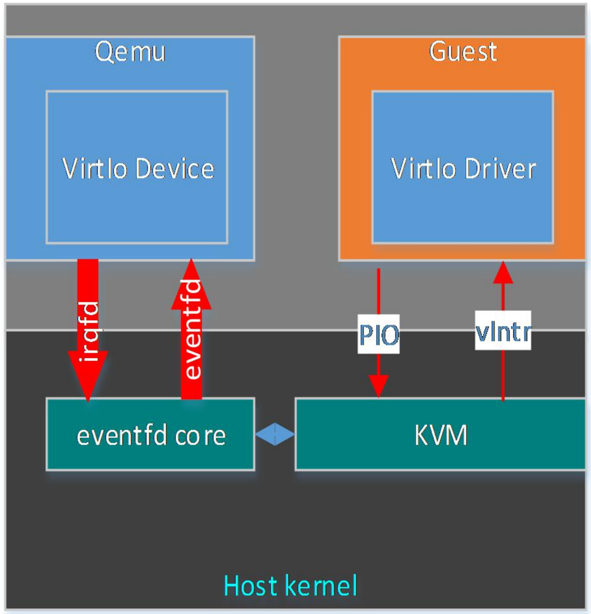
Vring data exchange flow



Notification

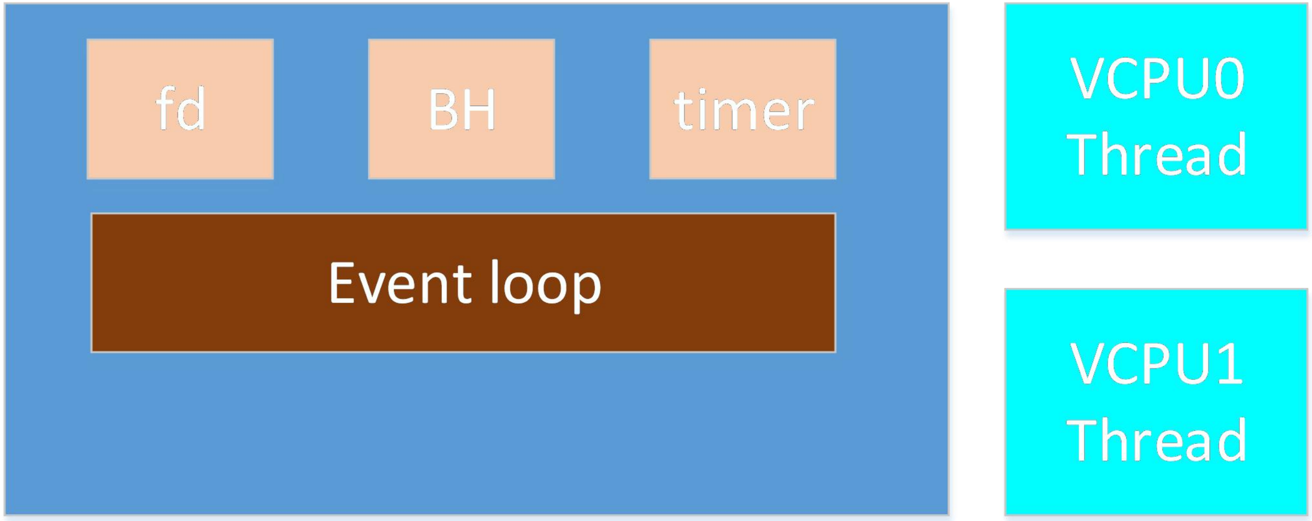
- Without ioeventfd and irqfd
- With ioeventfd and irqfd





Thread Model

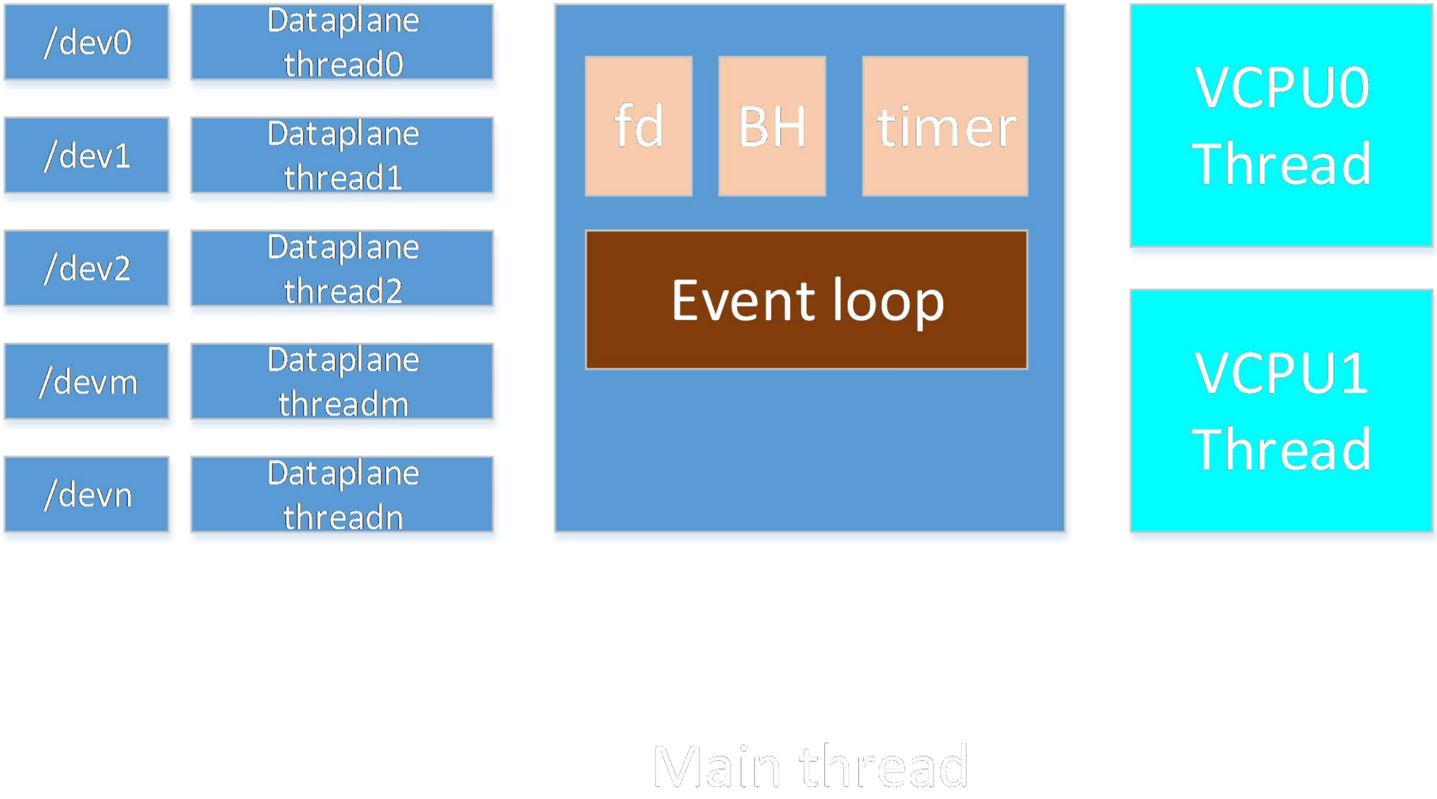
- Without IoThread/Dataplane



Main thread

Thread Model(cont 1)

- With IoThread/Dataplane



/dev0

Dataplane thread0

/dev1

Dataplane thread1

/dev2

Dataplane thread2

/devm

Dataplane threadm

/devn

Dataplane threadn

fd

BH

timer

Event loop

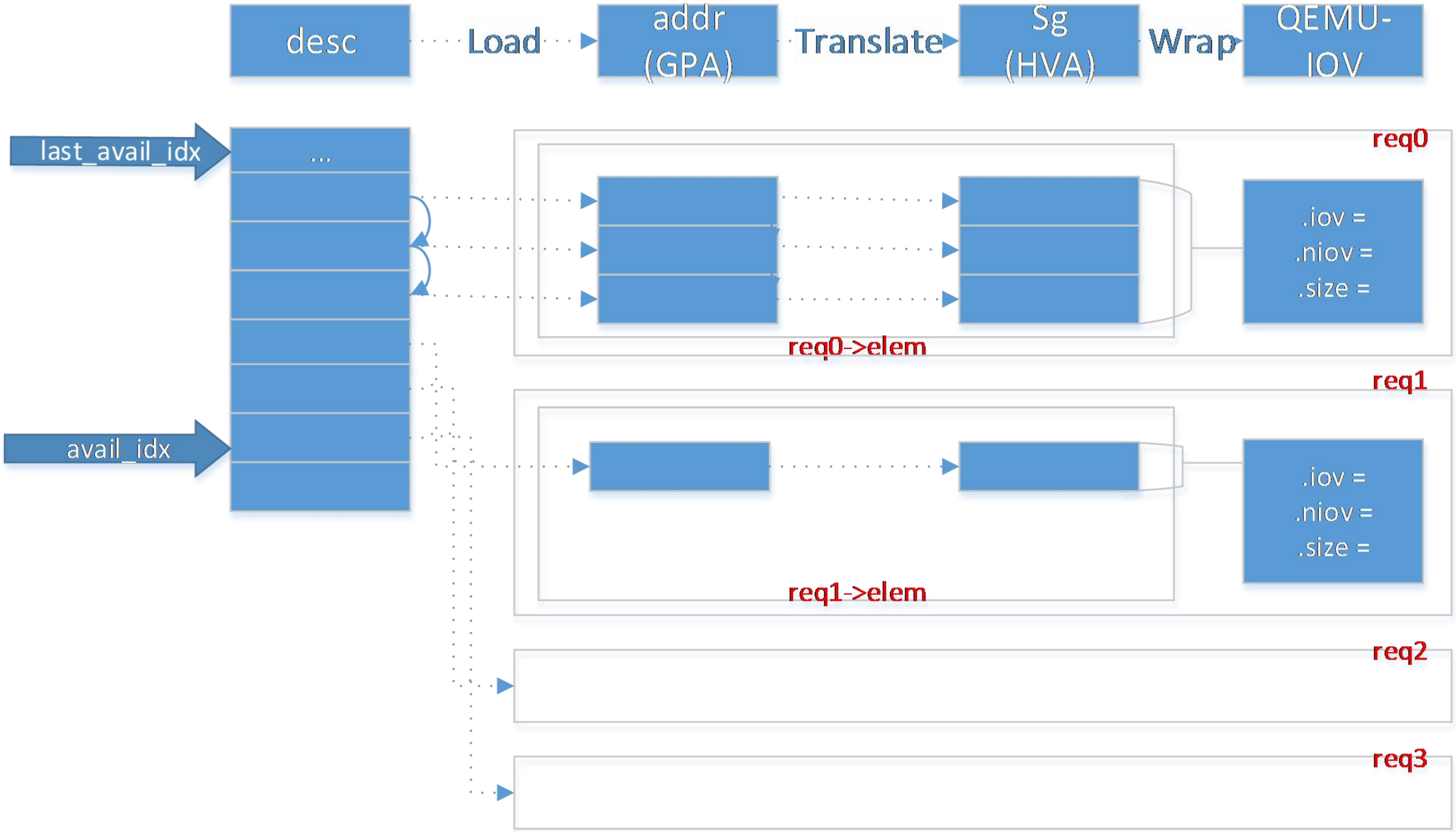
Main thread

VCPU0 Thread

VCPU1 Thread

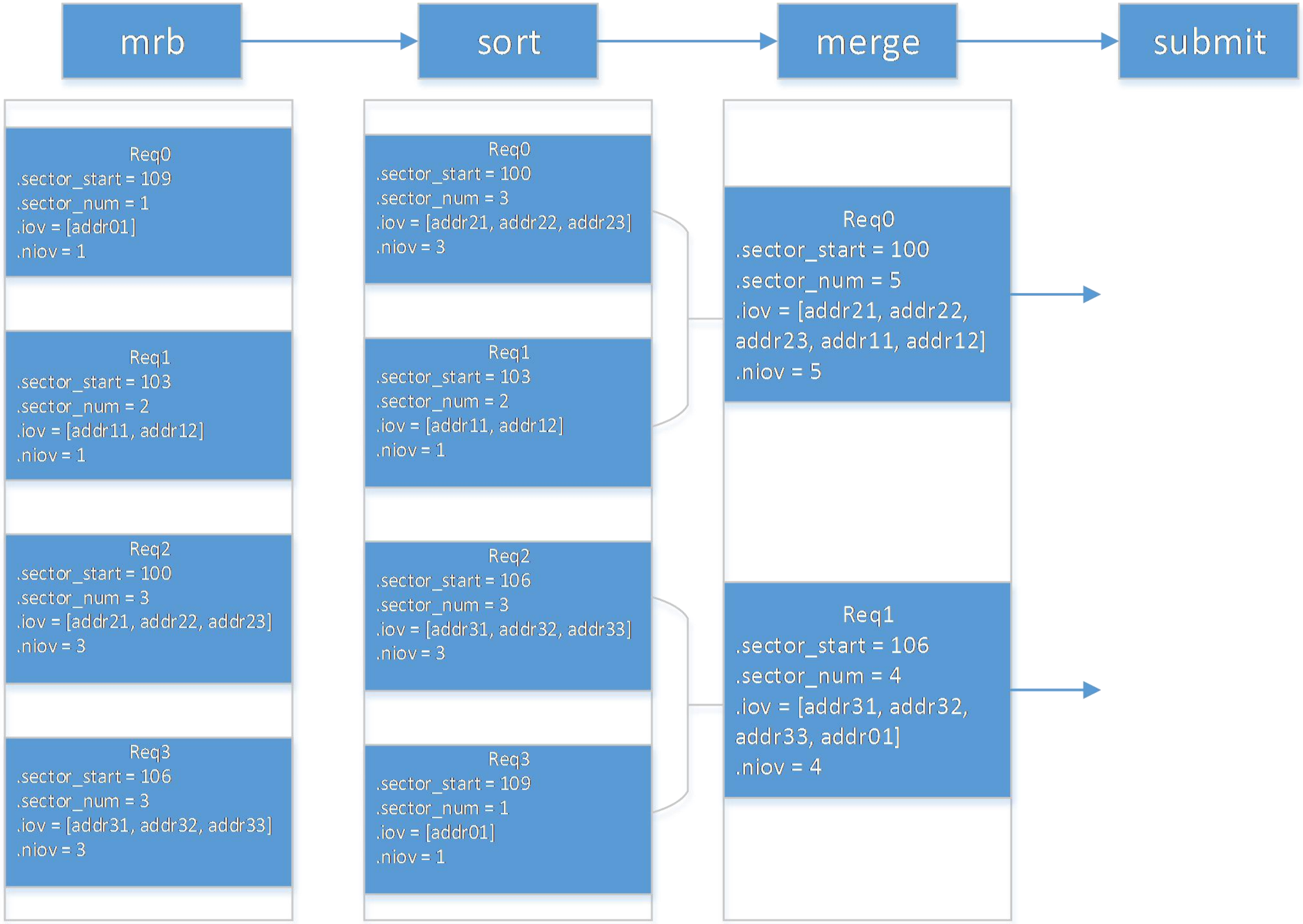
IO

- Get requests



IO(cont 1)

- Pre-processing



mrb → sort → merge → submit

mrb:

Req0
.sector_start = 109
.sector_num = 1
.iov = [addr01]
.niov = 1

Req1
.sector_start = 103
.sector_num = 2
.iov = [addr11, addr12]
.niov = 1

Req2
.sector_start = 100
.sector_num = 3
.iov = [addr21, addr22, addr23]
.niov = 3

Req3
.sector_start = 106
.sector_num = 3
.iov = [addr31, addr32, addr33]
.niov = 3

sort:

Req0
.sector_start = 100
.sector_num = 3
.iov = [addr21, addr22, addr23]
.niov = 3

Req1
.sector_start = 103
.sector_num = 2
.iov = [addr11, addr12]
.niov = 1

Req2
.sector_start = 106
.sector_num = 3
.iov = [addr31, addr32, addr33]
.niov = 3

Req3
.sector_start = 109
.sector_num = 1
.iov = [addr01]
.niov = 1

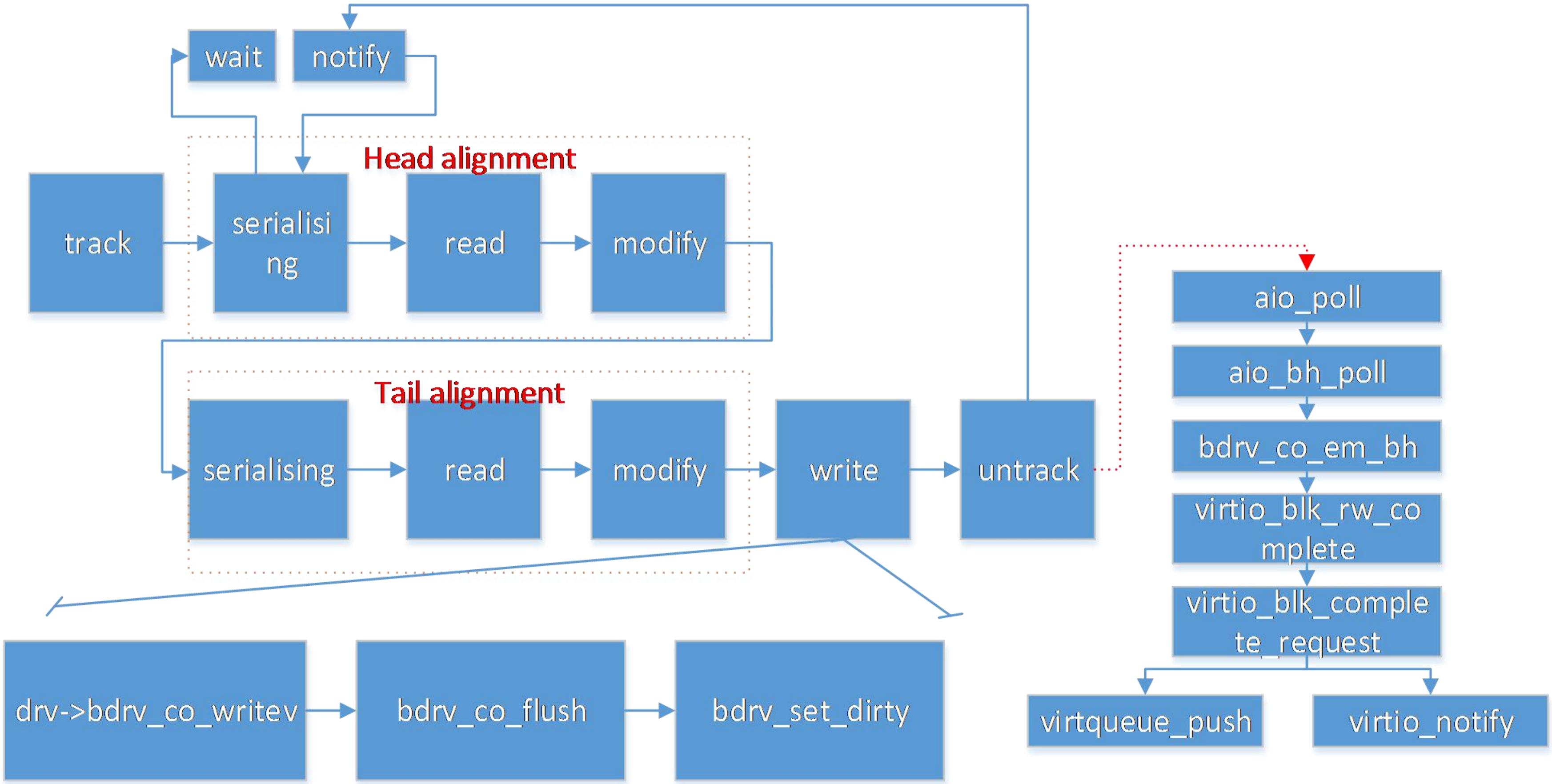
merge:

Req0
.sector_start = 100
.sector_num = 5
.iov = [addr21, addr22, addr23, addr11, addr12]
.niov = 5

Req1
.sector_start = 106
.sector_num = 4
.iov = [addr31, addr32, addr33, addr01]
.niov = 4

IO(cont 2)

- IO processing



Cache

- Double page cache
- Disk write cache

cache mode	semantics	host page cache	disk write cache	comment
writethroug	O_DSYNC	enable	disable	
writeback		enable	enable	
none	O_DIRECT	disable	enable	
directsync	O_DSYNC and O_DIRECT		diable	
unsafe		enable	enable	ignore guest flush