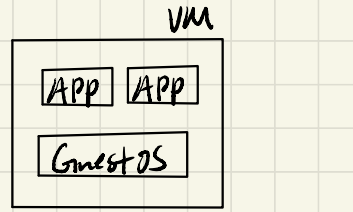
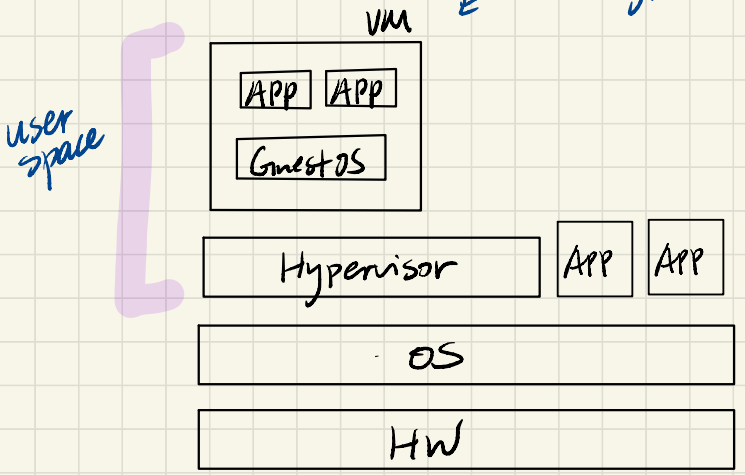
12/5 Virtualization Pt 2



Type 1 Virtualization

VM & hypervisor as a process



Type 2 virtualization

→ hypervisor fault domain = process

→ can use existing process monitoring tools for VMs & hypervisor

Qemu

→ full machine emulator

→ can emulate different architectures

? How to run one architecture on top of another?

→ different instruction set [map instructions from one arch to another]

→ different registers [map registers onto host machine's register]
(or map registers into memory)

Binary Translation

① typically translate instr into micro ops first.

② translation done in blocks, result is cached

? How to virtualize memory?

→ hypervisor needs to virtualize physical memory for Guest OS

→ mmap annonymous memory as "physical memory"

→ Guest OS then uses the mmaped region as physical memory

→ qemu needs to provide software mmu to do pt walk for GVA → GPA

→ qemu also virtualizes storage & network

☆ How's the performance now?

KVM (Kernel based virtual machine)

(VTx)

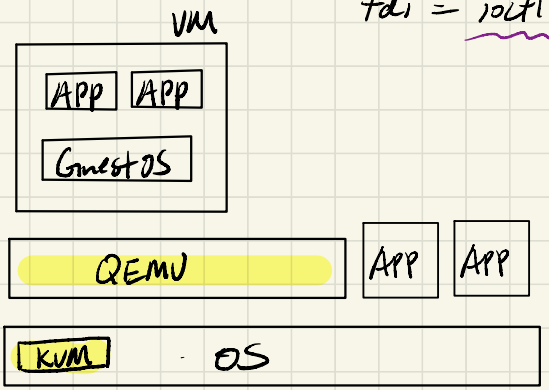
→ a char device in Linux that lets processes use hw virtualization support
(also do some in kernel device emulation)

→ /dev/kvm

→ access via `fd = open("/dev/kvm");`

existing syscalls!

`fd1 = ioctl(fd, KVM_CREATE_VM);`



Lightweight VM exit =

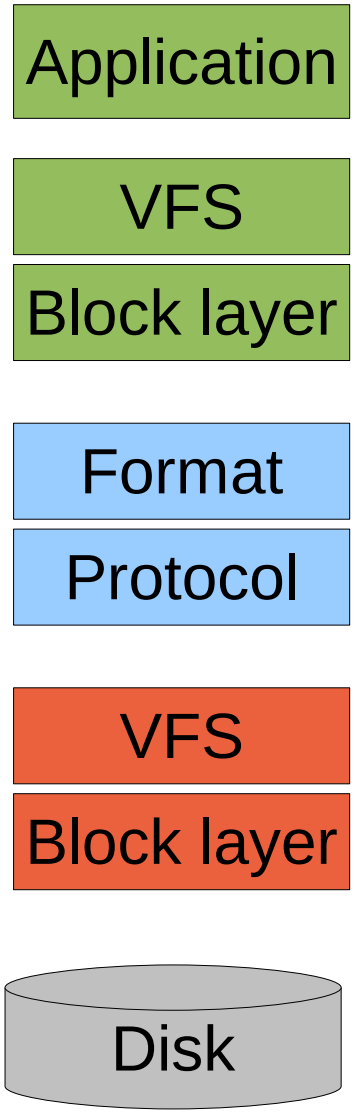
Guest OS → KVM → Guest OS

Heavyweight VM exit =

Guest OS → KVM → Qemu → KVM → Guest OS

Typically used along w/ qemu, KVM
takes care of CPU & memory virtualization
QEMU does device virtualization

Storage stack



Guest – application plus full file system and block layer

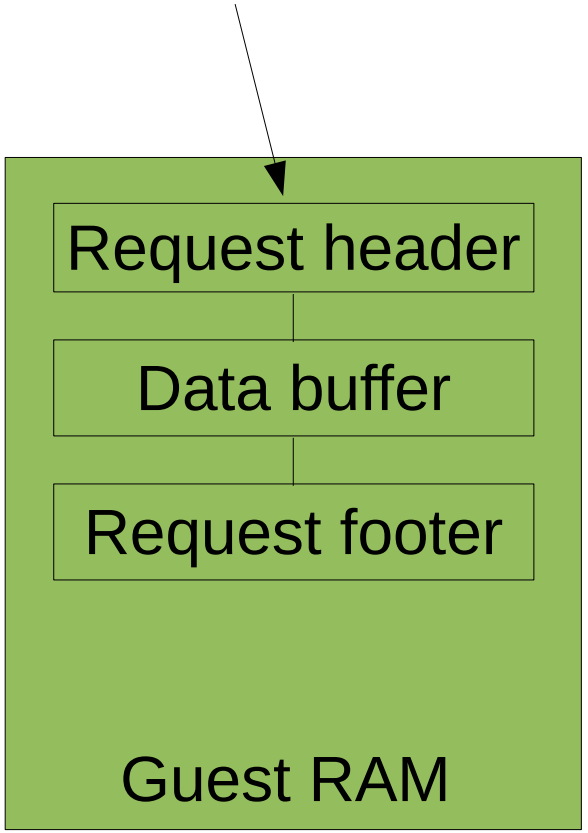
QEMU – image format, storage migration, I/O throttling

Host – full file system and block layer

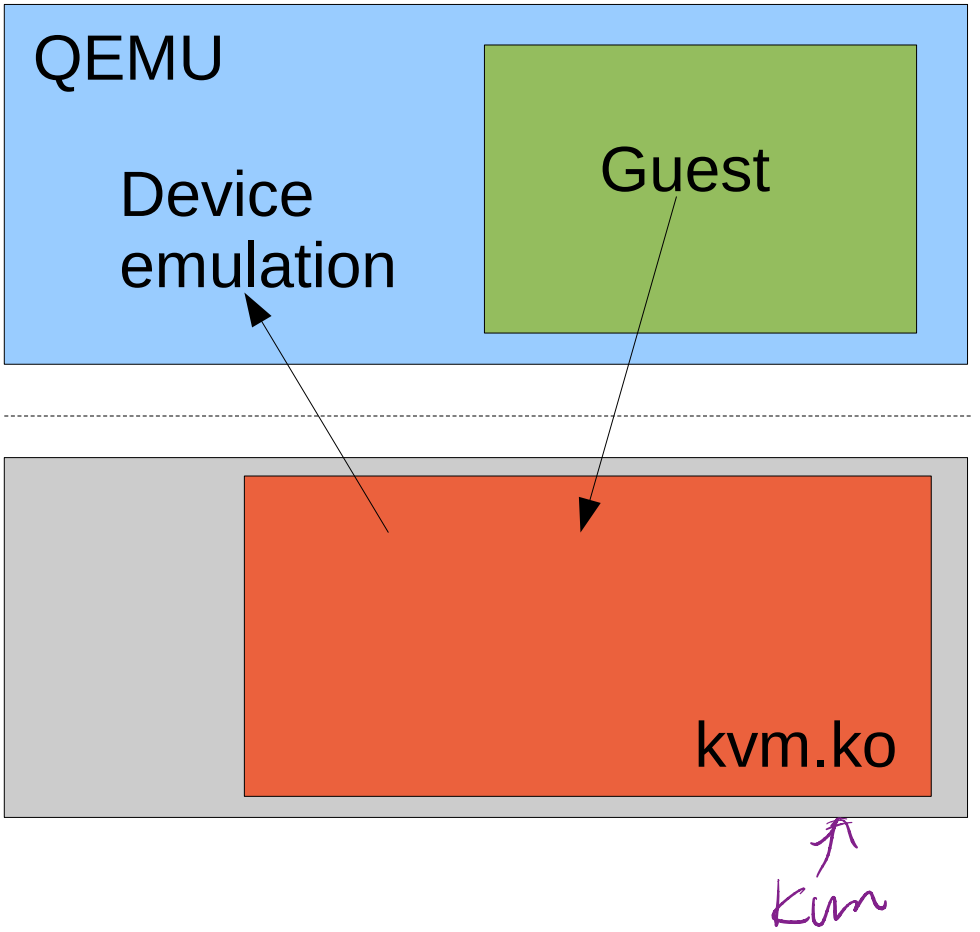
Beware double caching and anticipatory scheduling delays!

Walkthrough: virtio-blk disk read request (Part 1)

1. Guest fills in request descriptors

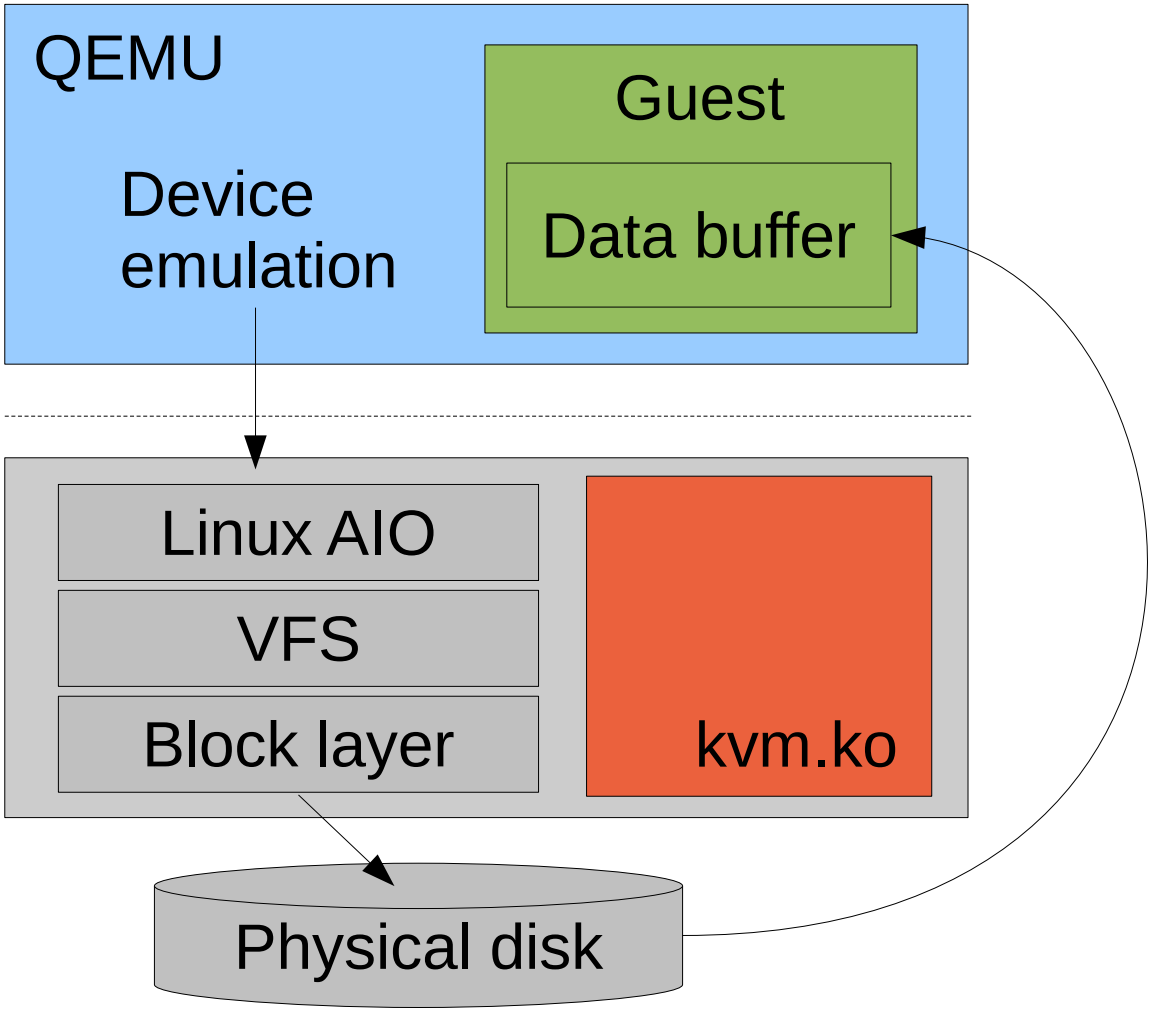


2. Guest writes to virtio-blk virtqueue notify register



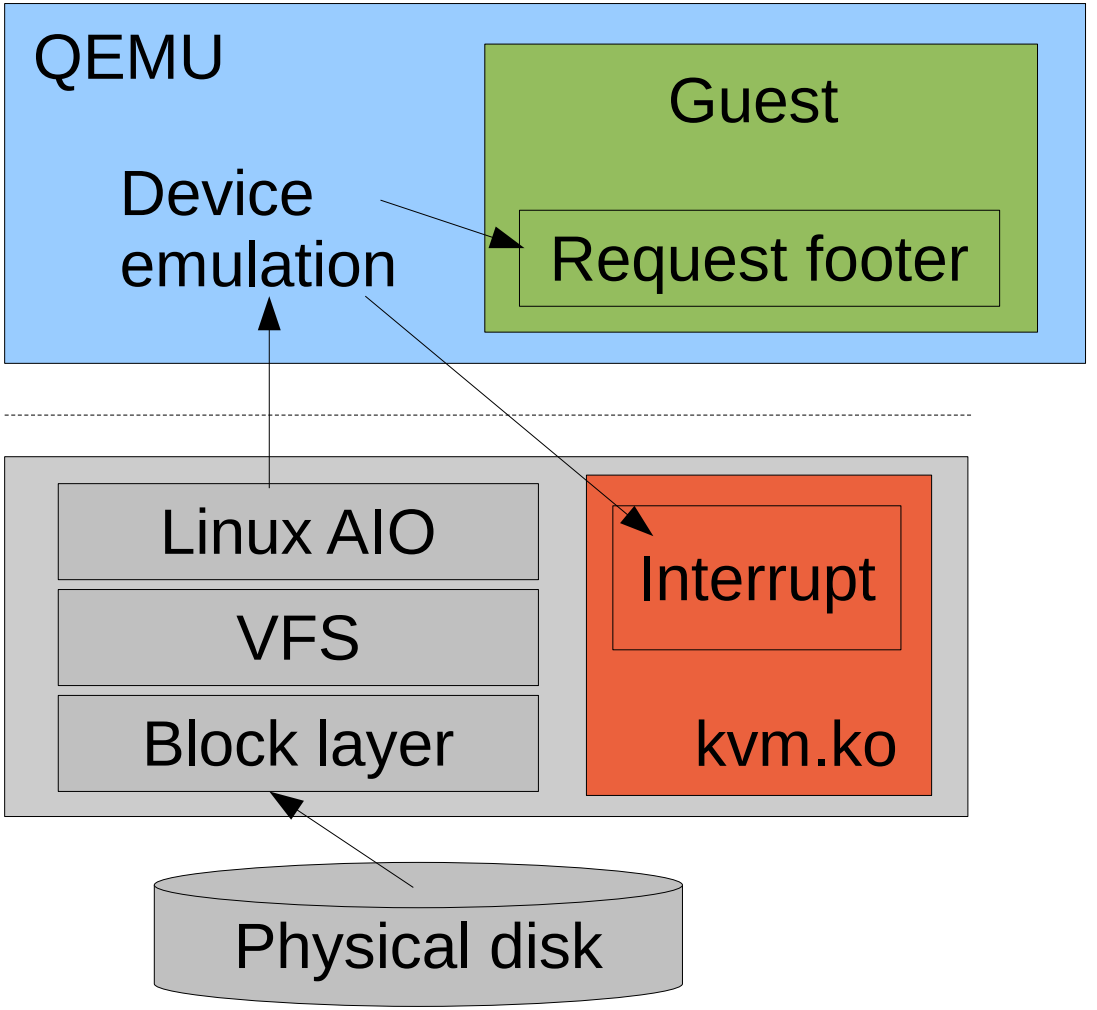
Walkthrough: virtio-blk disk read request (Part 2)

3. QEMU issues I/O request on behalf of guest



Walkthrough: virtio-blk disk read request (Part 3)

4. QEMU fills in request footer and injects completion interrupt



Walkthrough: virtio-blk disk read request (Part 4)

5. Guest receives interrupt and executes handler

6. Guest reads data from buffer