CS 695: Virtualization and Cloud Computing

Lecture 7: I/O Virtualization Techniques

Mythili Vutukuru

IIT Bombay

Spring 2021

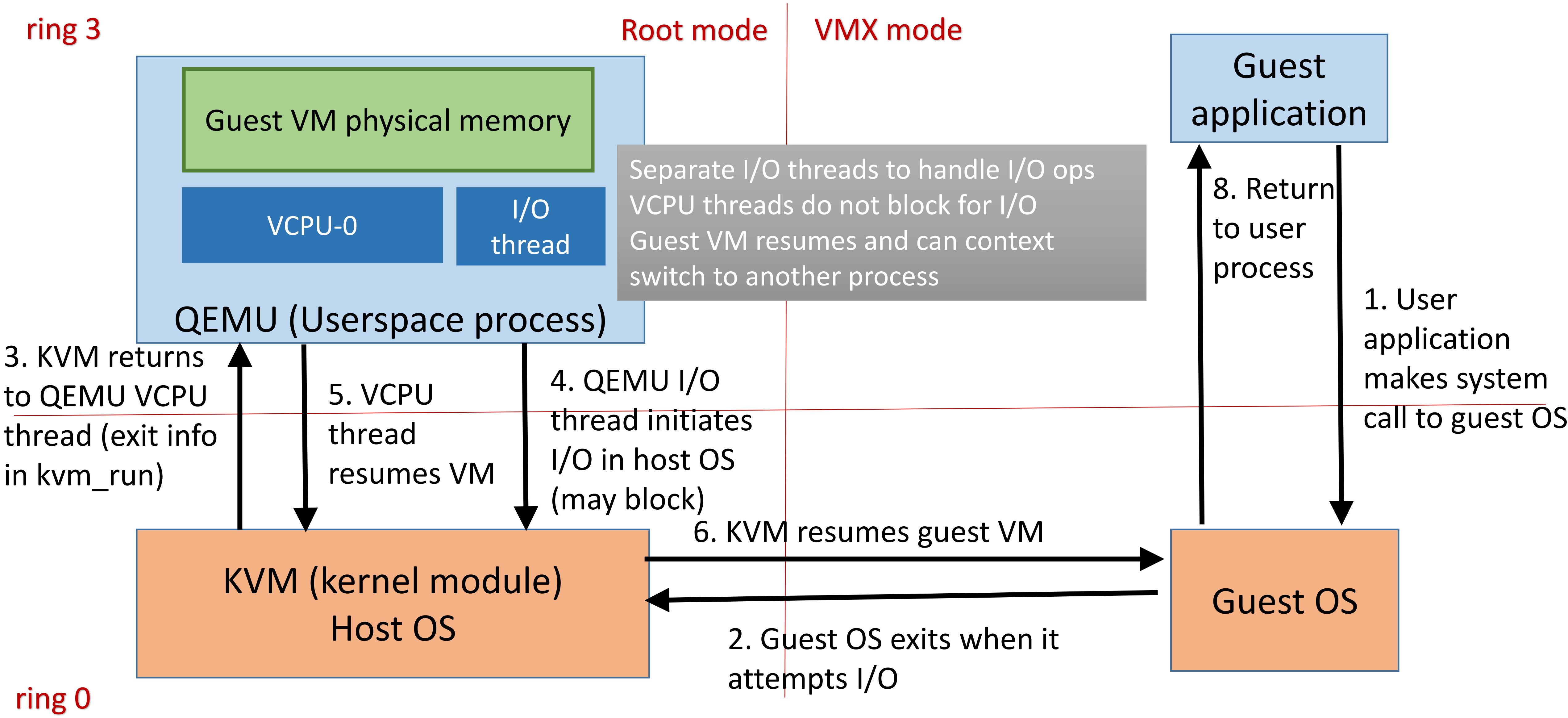
Techniques for I/O virtualization

- Guest OS cannot get full access to I/O devices
  - VMM must share I/O device access across guests
- Two ways to virtualize I/O devices:
  - Emulation: I/O access in guest traps to VMM, which performs I/O
  - Direct I/O or device passthrough: a slice of device is assigned directly to guest
- Many optimizations exist, only basics discussed here

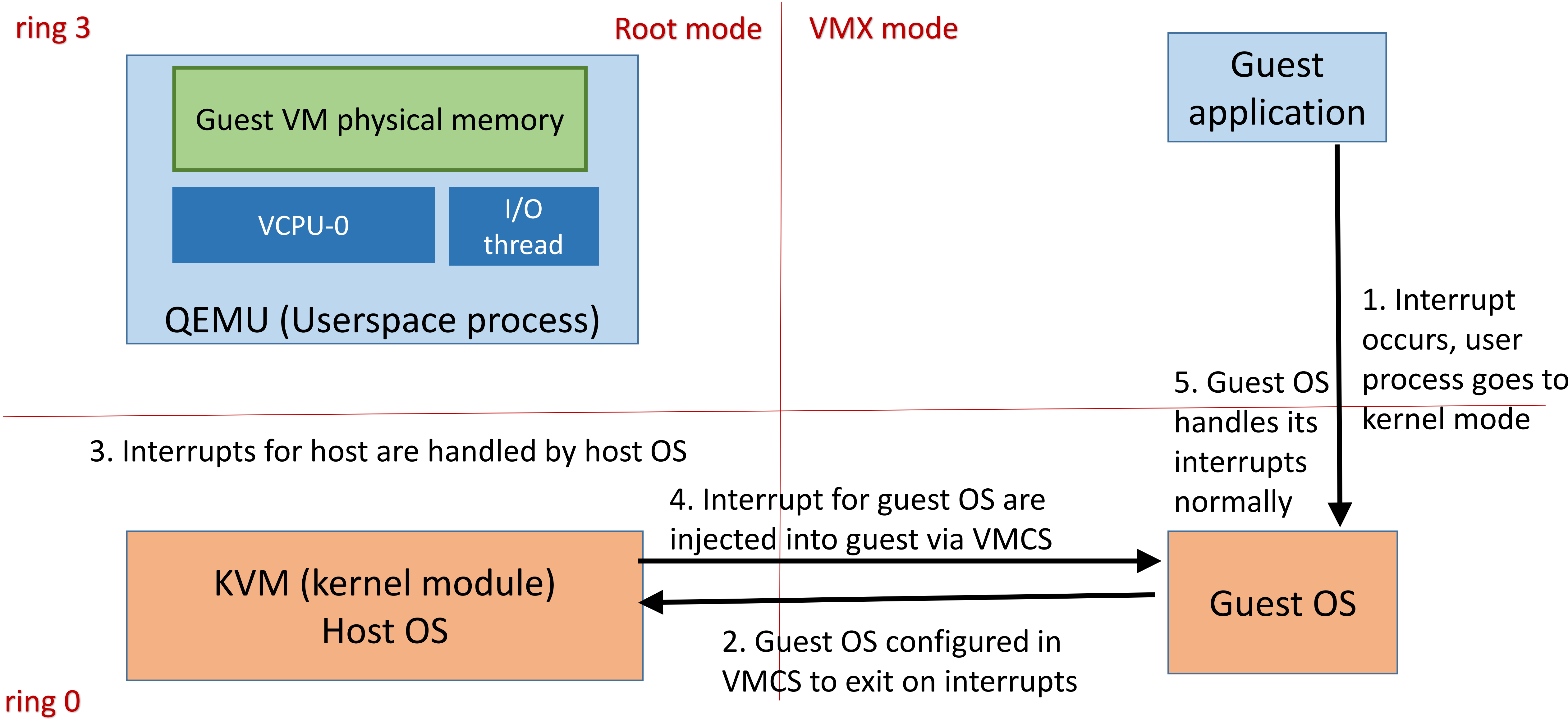
Communication between OS and device

- Device memory exposed as registers (command, status, data etc.)
  - I/O happens by reading/writing this memory
  - E.g., write command into device register to begin I/O
- OS can read/write device registers in two ways:
  - Explicit I/O: in/out instructions in x86 can write to device memory
  - Memory mapped I/O: Some memory addresses are assigned to device memory and are not to RAM. I/O happens by reading/writing this memory.
- Accessing device memory (via explicit I/O or memory mapped I/O) can be configured to trap to VMM
- Device raises interrupt when I/O completes (alternative to polling)
  - Modern I/O devices perform DMA (Direct Memory Access) and copy data from device memory to RAM before raising interrupt
  - Device driver provides physical address of DMA buffers to device

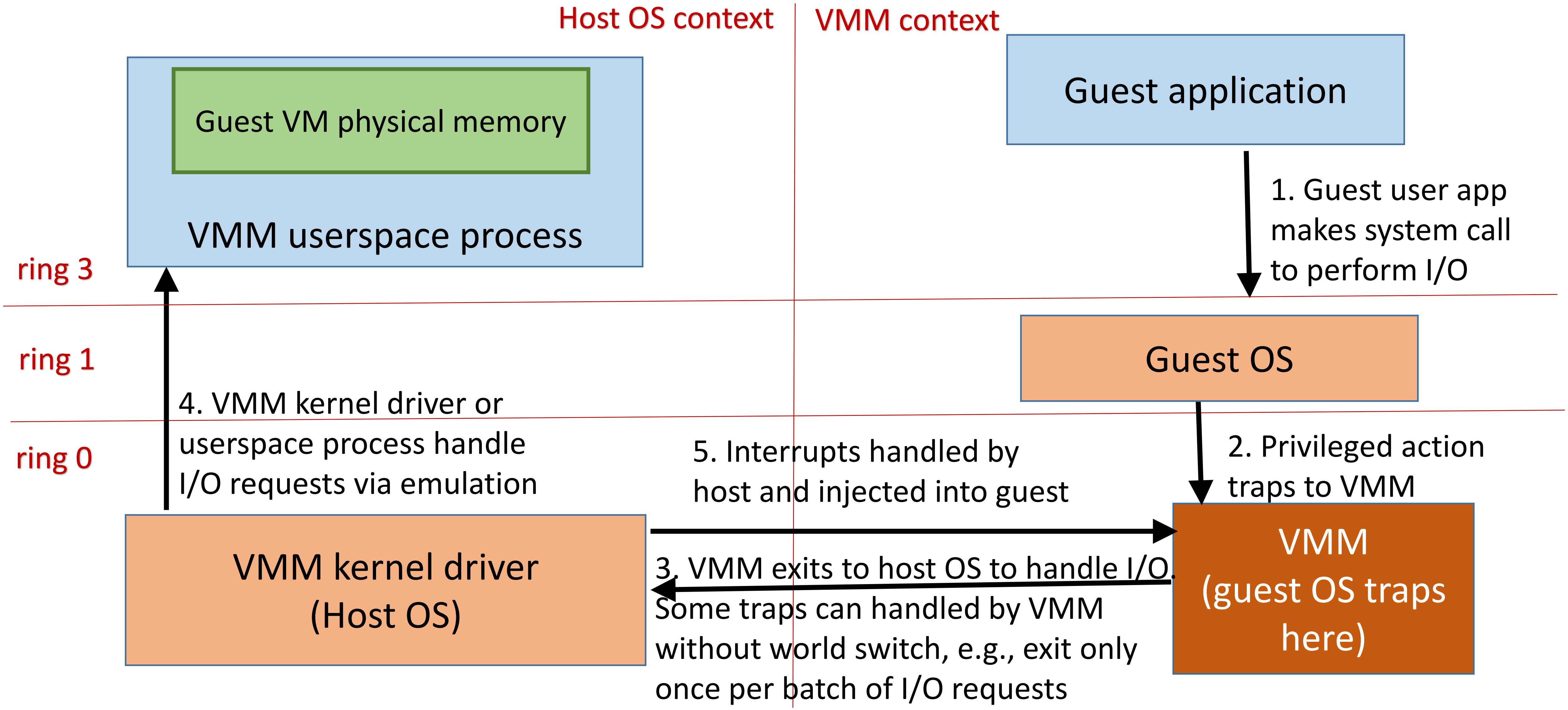
QEMU/KVM I/O handling



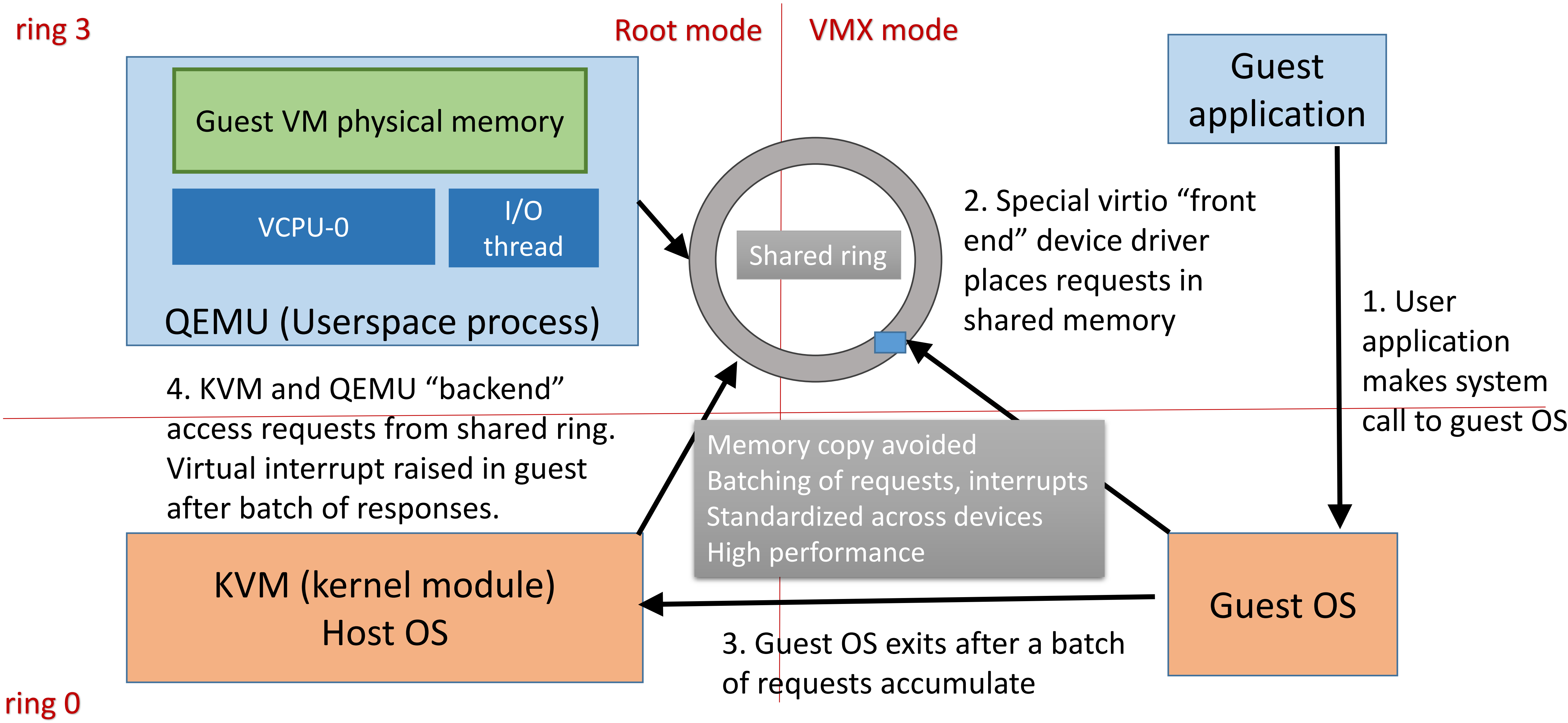
QEMU/KVM interrupt handling



Full virtualization VMM architecture

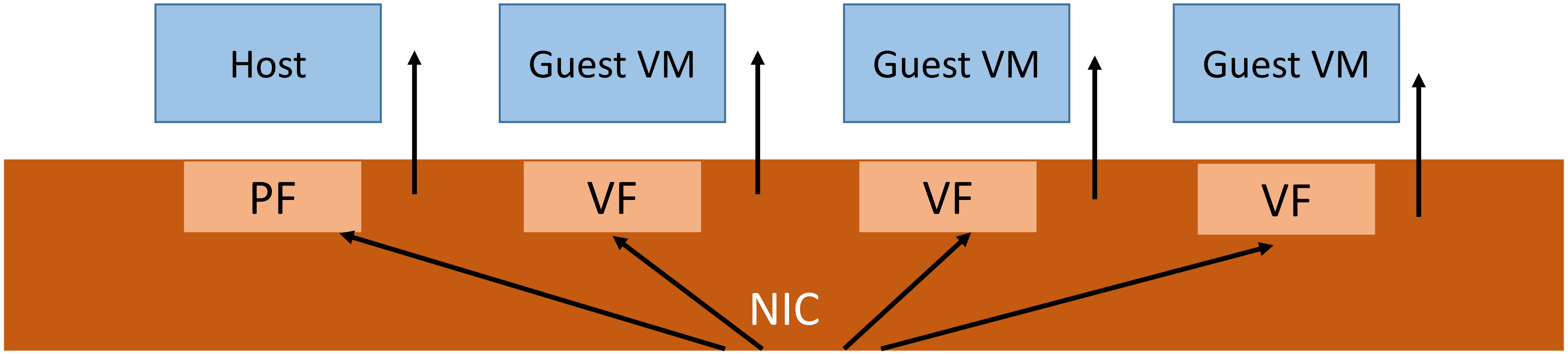


QEMU/KVM virtio optimization



Device passthrough or Direct I/O

- More efficient than device emulation
- Example: SR-IOV (Single Root IO Virtualization) in network devices
  - Network card has one physical function (PF) and many virtual functions (VFs)
  - PF managed by host OS, each VF assigned to one guest VM
  - Each VF is like a separate NIC, and is bound to a guest VM
  - Packets destined to the MAC address of VM are switched to corresponding VF



SR-IOV

- SR-IOV NIC communicates directly with device driver in guest OS
  - Packets do not go to the host OS stack at all
  - Packets switched at Layer-2 using VM virtual device's MAC address
  - Packets DMA'ed directly into guest VM memory, host OS not involved
  - But, interrupts may still cause VM exit (interrupt can be for host too)
- Challenge: when guest device driver provides DMA buffers to VF, it can only provide guest physical addresses (GPA) of the buffer
  - NIC cannot access the DMA buffer memory using GPA alone
- SR-IOV capable NICs have an inbuilt MMU (IOMMU) to translate from GPA to HPA

Summary

- Techniques for I/O virtualization
  - Device emulation
  - Virtio optimization
  - Device passthrough or direct I/O (SR-IOV)