Virtual Machines

Heyi Li and Zhen Cao

(Some of the figures are from the Internet)

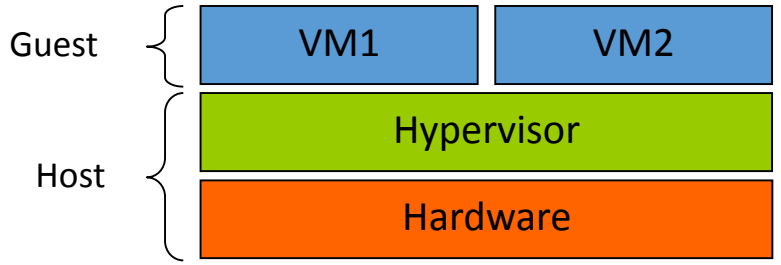
Outline

- Basic concepts
- When virtual is better
- Implementation
- When virtual is harder

Basic Concepts

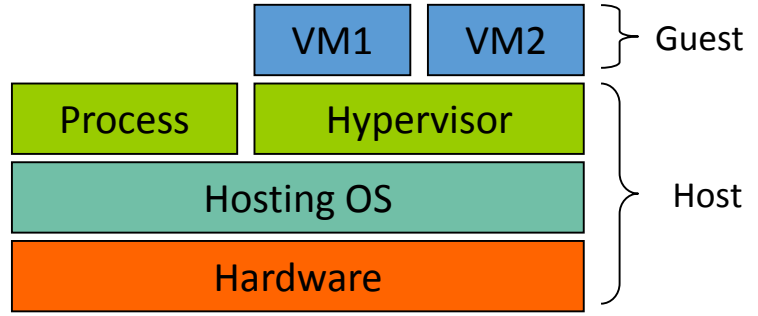
- What is a virtual machine?
  - An emulation of a particular computer system
- System VM vs. Process VM
  - System VM: supports the execution of a complete OS (Xen)
  - Process VM: supports the execution of a single process (JVM)
- Hypervisor (VMM)
  - Computer software that creates and runs VMs
- Type I & II Hypervisor

Type 1 (bare-metal)



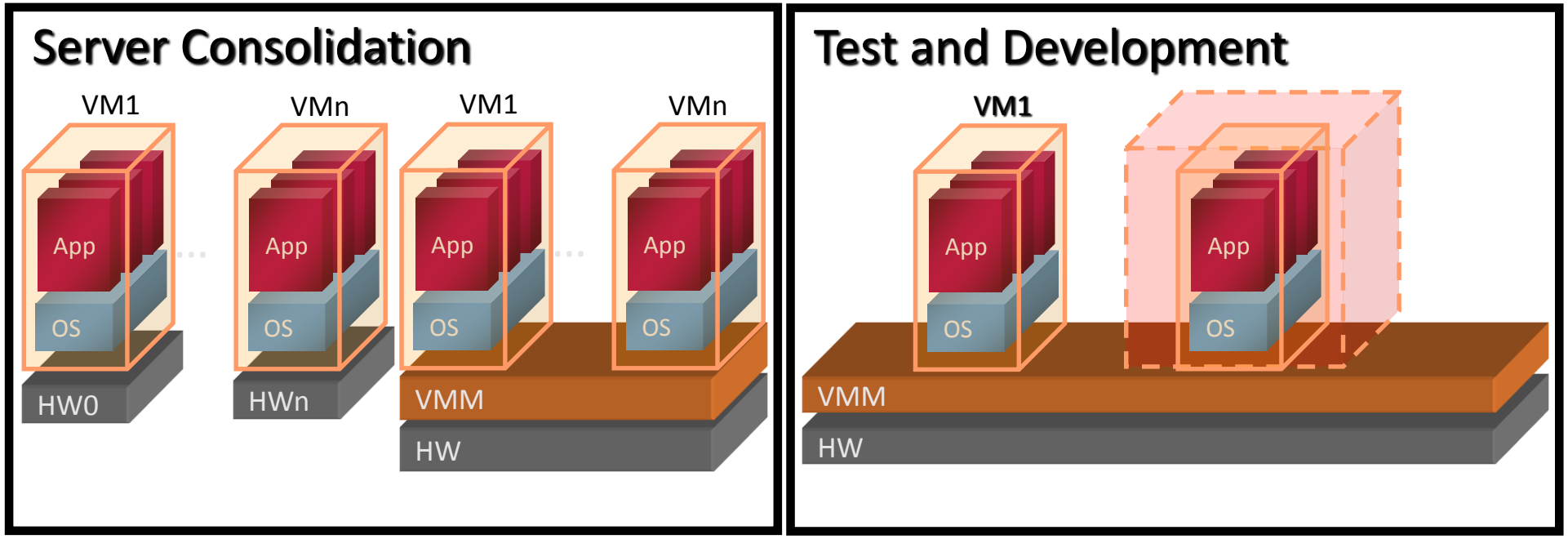
VMware ESX, Microsoft Hyper-V, Xen

Type 2 (hosted)



VMware Workstation, Microsoft Virtual PC, Sun VirtualBox, QEMU, KVM

Applications and Benefits



- Energy efficiency
- Reducing Maintenance costs
- Rapid deployment
- Security

Virtualization Requirements

- Fidelity
  - Software on the VM executes identically to its execution on hardware, barring time effects
- Performance
  - Performance overhead must be small
- Safety
  - The VMM manages all hardware resources

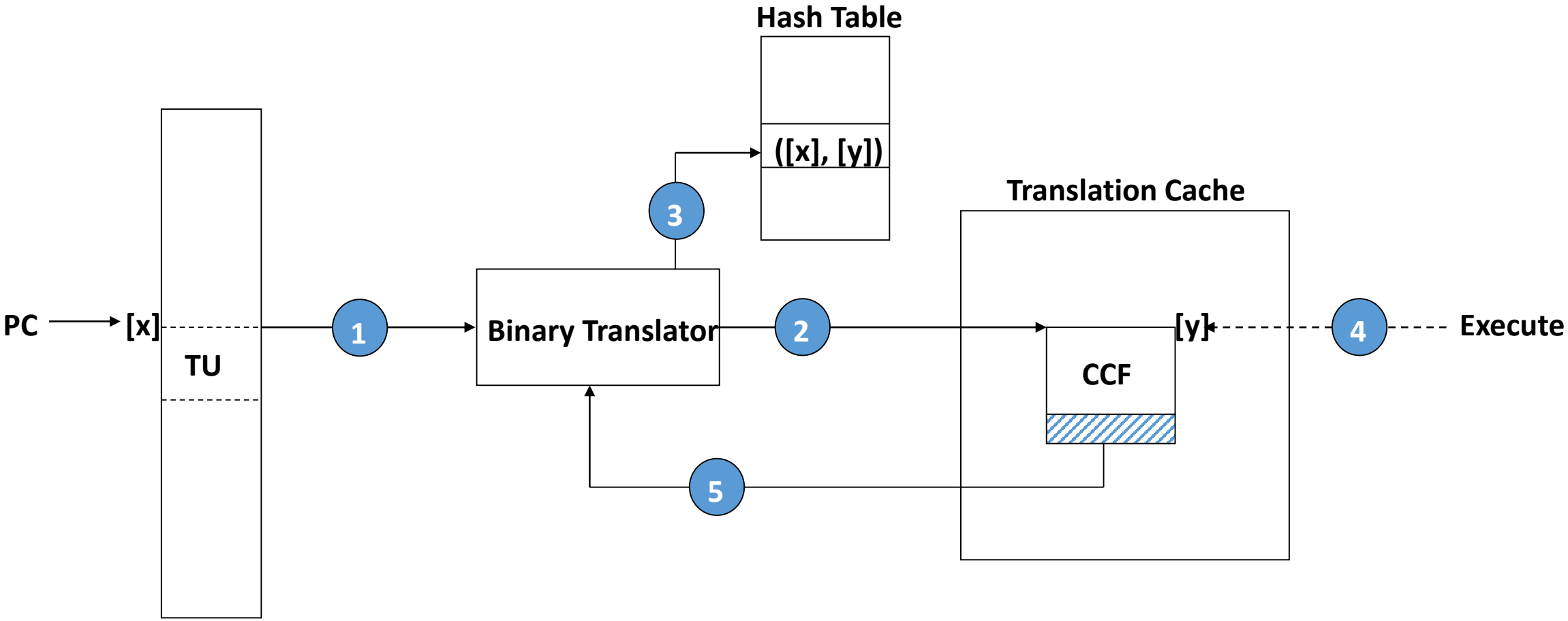
Obstacles for X86

- Trap-and-emulate
  - All virtualization-sensitive instructions are also privileged instructions
- x86 architecture once thought to be not fully virtualizable
  - Certain privileged instructions behave differently when run in unprivileged mode (POPF)
  - Certain unprivileged instructions can access privileged state (SGDT)
- Techniques to address inability to virtualize x86
  - Full virtualization w/o hardware support – Binary Translation (VMware ESX)
  - Paravirtualization (Xen)
  - Hardware-assisted virtualization

Binary Translation

Binary Translation

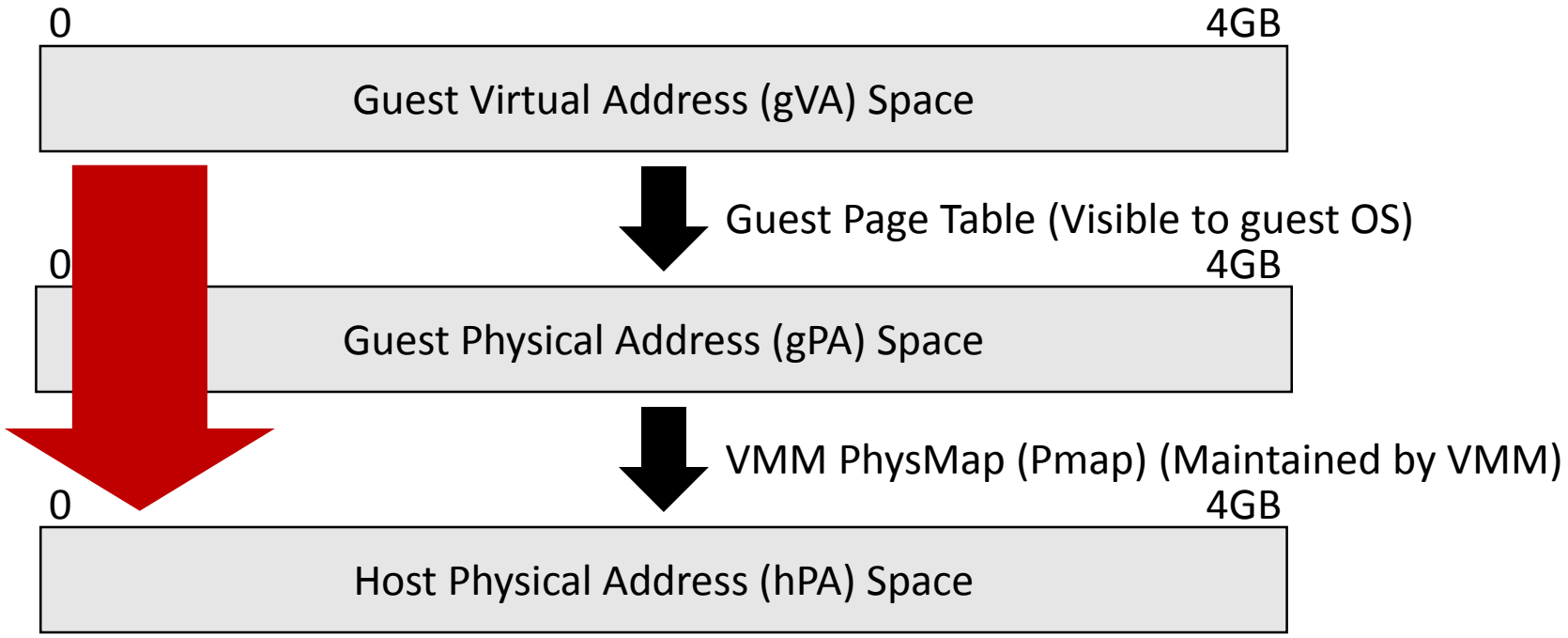
- Binary: input is binary x86 code, not source code
- On-the-fly: dynamic and on demand
- Only need to translate kernel mode code
  - User mode: direct execution
- Even for kernel mode, most instruction sequences don't change
- Instructions that **do change:**
  - Indirect control flow: call/ret, jmp
  - PC-relative addressing
  - Privileged instructions



1. A translation unit stops at 12 instructions or a control-flow instruction
2. Translated into Compiled Code Fragments(CCF) and cached
3. Track the translation cache with a hash table
4. Execute the CCF
5. Continuation (either fall-through or taken-branch)

Memory

Shadow Page Table (Resides in hardware and maintained by VMM)

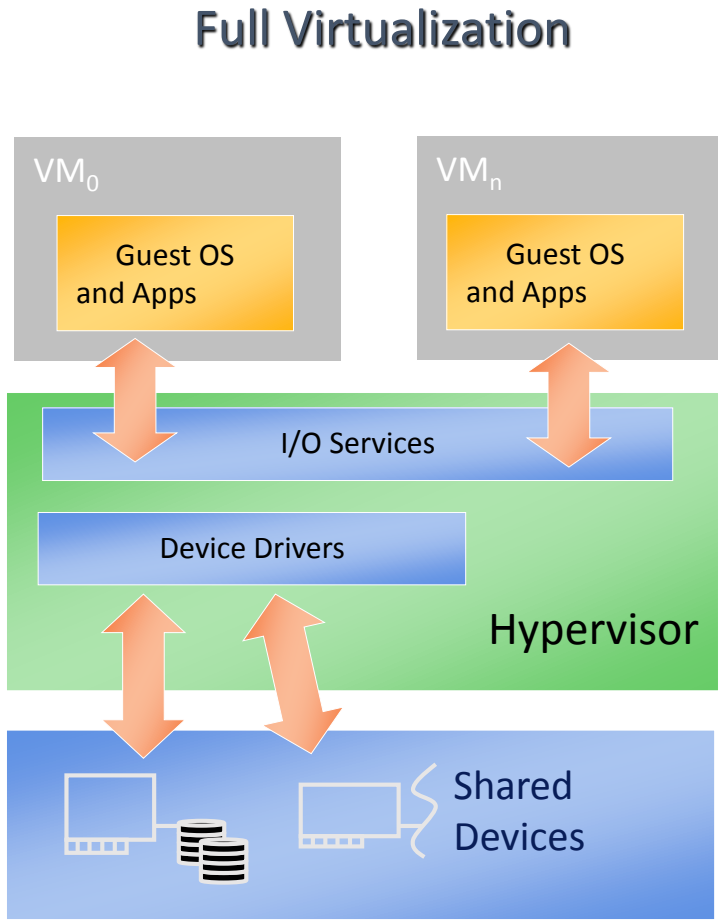


Shadow Page Tables

- Translation from gVA to hPA directly by hardware
- If not present, page fault generated by hardware
- *Hidden page fault*: the mapping present in guest page table
  - VMM walks the guest page table to determine the gPA backing that gVA
  - VMM allocates a physical page, and adds the mapping to Pmap
  - Updates the shadow page table
- *True page fault*: the mapping not present in guest page table
  - VMM generates an exception on the virtual cpu
  - Resume executing on the first instruction of the guest exception handler

I/O Virtualization – Direct I/O Model

- Place drivers for high-performance I/O devices directly into hypervisor
- *Not* attempt to have the virtual hardware match the specific underlying hardware
- Virtualize selected, canonical I/O devices
- Problems
  - Larger Hypervisor
  - Need to protect hypervisor from driver faults



Paravirtualization

CPU Virtualization

- Privilege levels in x86
  - Ring 0: Xen
  - Ring 1: guest OS
  - Ring 3: user apps
- Isolation
  - Guest user mode and guest kernel mode
    - Page table “supervisor” bit: PTE_U
  - Guest OS and VMM
    - Segmentation
  - Problem with x86-64

CPU Virtualization (cont.)

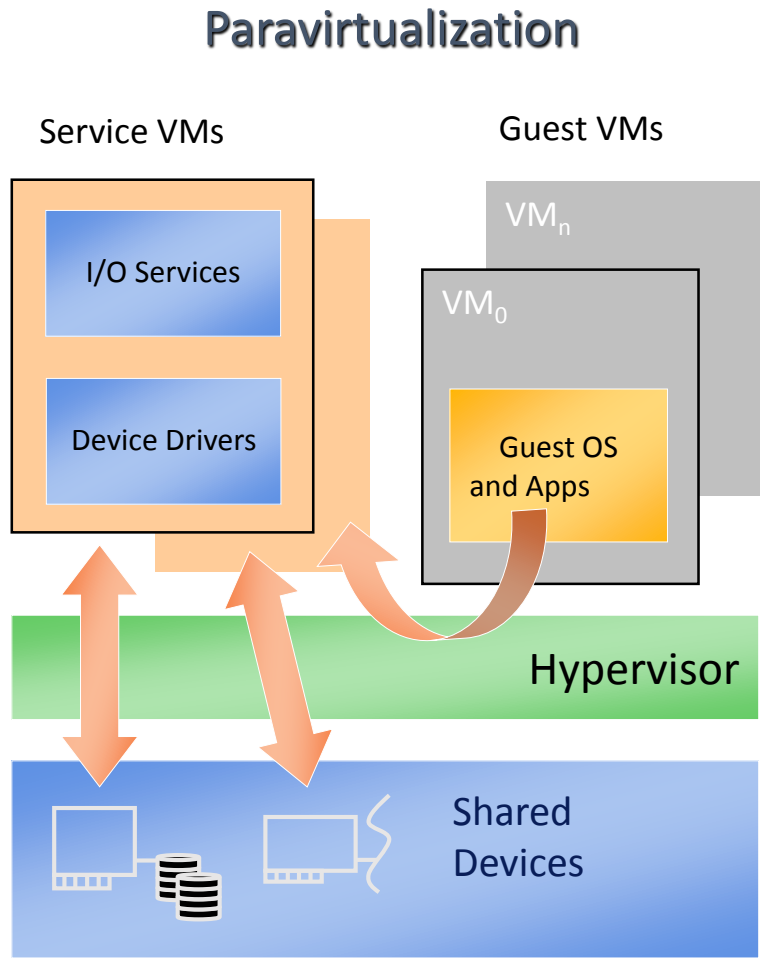
- Privileged instructions
  - Hypercalls
  - Modify source codes
  - Validated and executed by Xen (e.g., installing a new PT)
- Exceptions
  - Registered with Xen once. Accepted (validated) if don't require to execute exception handlers in ring0.
  - Called directly without Xen intervention
  - All syscalls from apps to guest OS handled this way (and executed in ring1)
- Page fault handlers are special
  - Faulting address can be read only in ring 0
  - Xen reads the faulting address and passes it via stack to the OS handler in ring1

Memory Virtualization

- Physical memory
  - At domain creation, hardware pages “reserved”
  - Domain can increase/decrease its quota
  - Xen does not guarantee that the hardware pages are contiguous
- Virtual memory
  - Register guest OS page tables directly with MMU
  - Guest OS allocates and initializes a page from its own memory reservation and registers it with Xen
    - Every guest OS has its own address space
    - Xen occupies top 64MB of every address space.
      - To save switching costs between address spaces (hypervisor calls)
  - Xen involved only in memory updates

I/O Virtualization – Indirect I/O Model

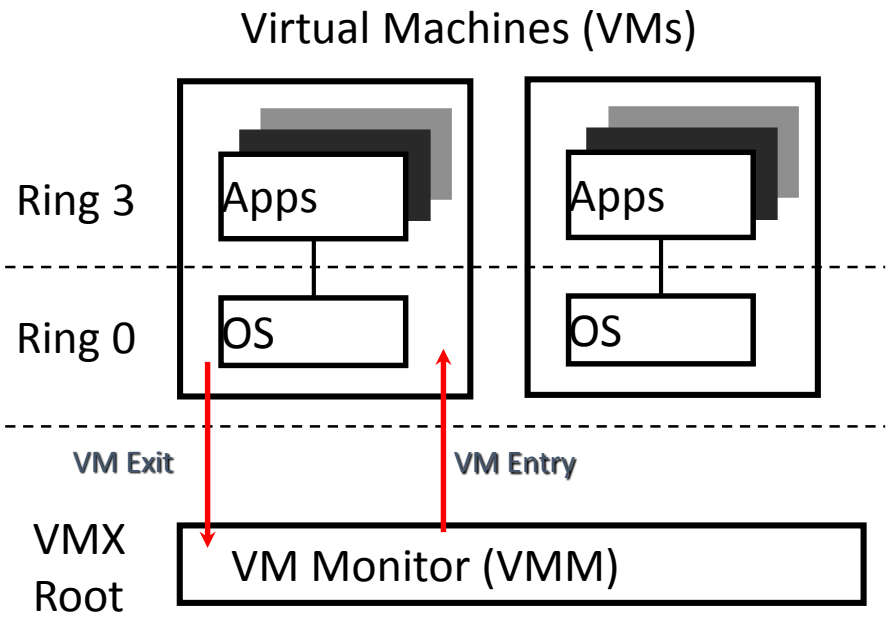
- Uses a privileged virtual machine (Domain0) for all device drivers
- Simple interfaces for guest OSes
- Pros
  - higher security
- Cons
  - lower performance



Hardware-assist Virtualization (HVM)

Intel's VT-x

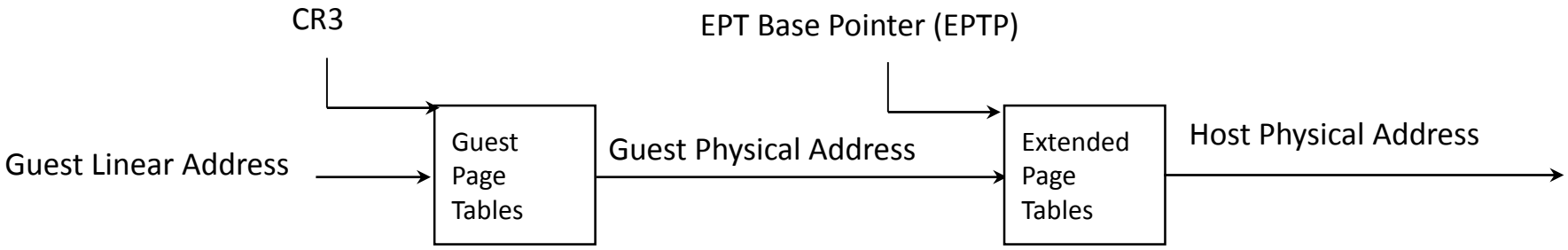
- More-privileged mode for VMM
- Less-privileged mode for guest OS
- Eliminate de-privileging of Ring for guest OS



VM Control Structure(VMCS)

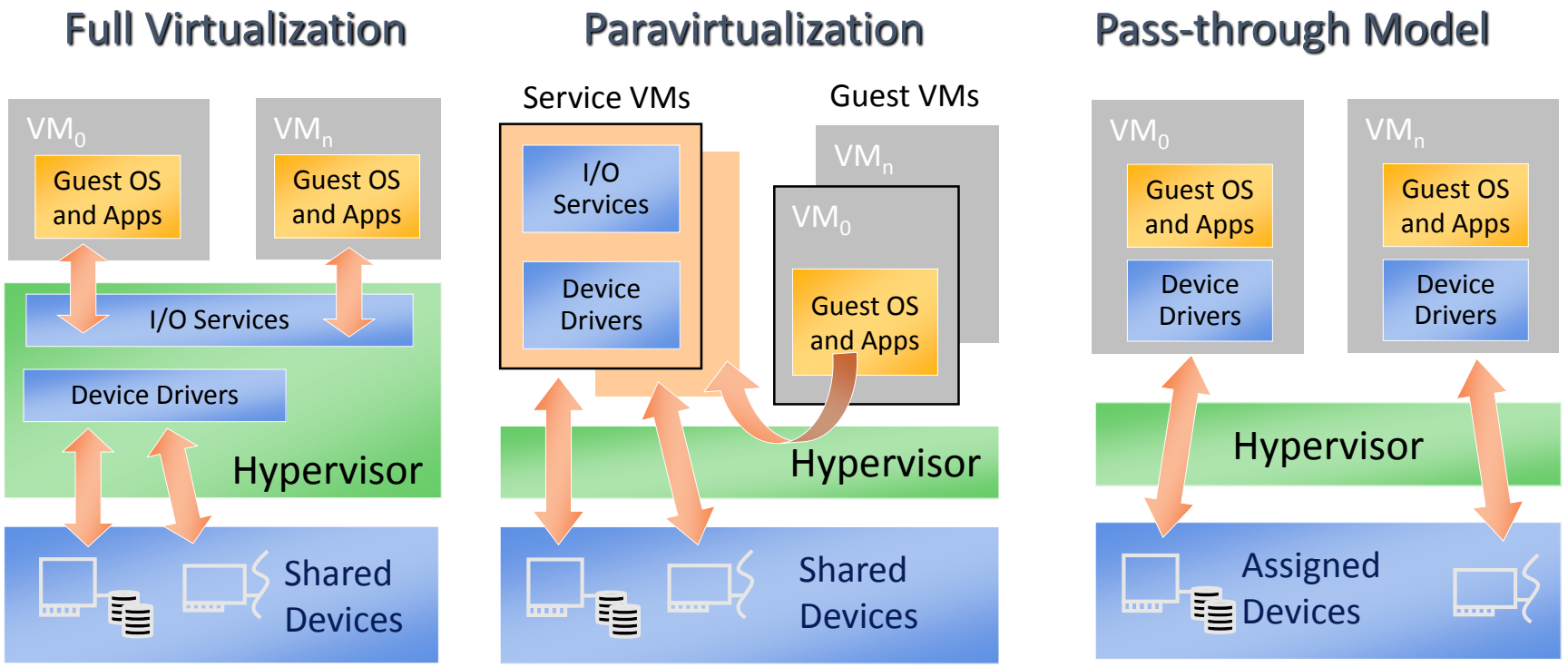
- Execution controls determine when exits occur
  - Access to privileged state, occurrence of exceptions, etc.
  - Flexibility provided to avoid unwanted exits
- Guest-state area
  - Processor state saved into the guest-state area on VM exits and loaded on VM entries
- Host-state area
  - Processor state loaded from the host-state area on VM exits
- Other

Extended Page Table(EPT)



- A new page-table structure, under the control of the VMM
  - Defines mapping between GPA & HPA
  - EPT base pointer (new VMCS field) points to the EPT page tables
  - EPT (optionally) activated on VM entry, deactivated on VM exit
- Guest has full control over its own IA-32 page tables
  - No VM exits due to guest page faults, INVLPG, or CR3 changes

I/O Virtualization

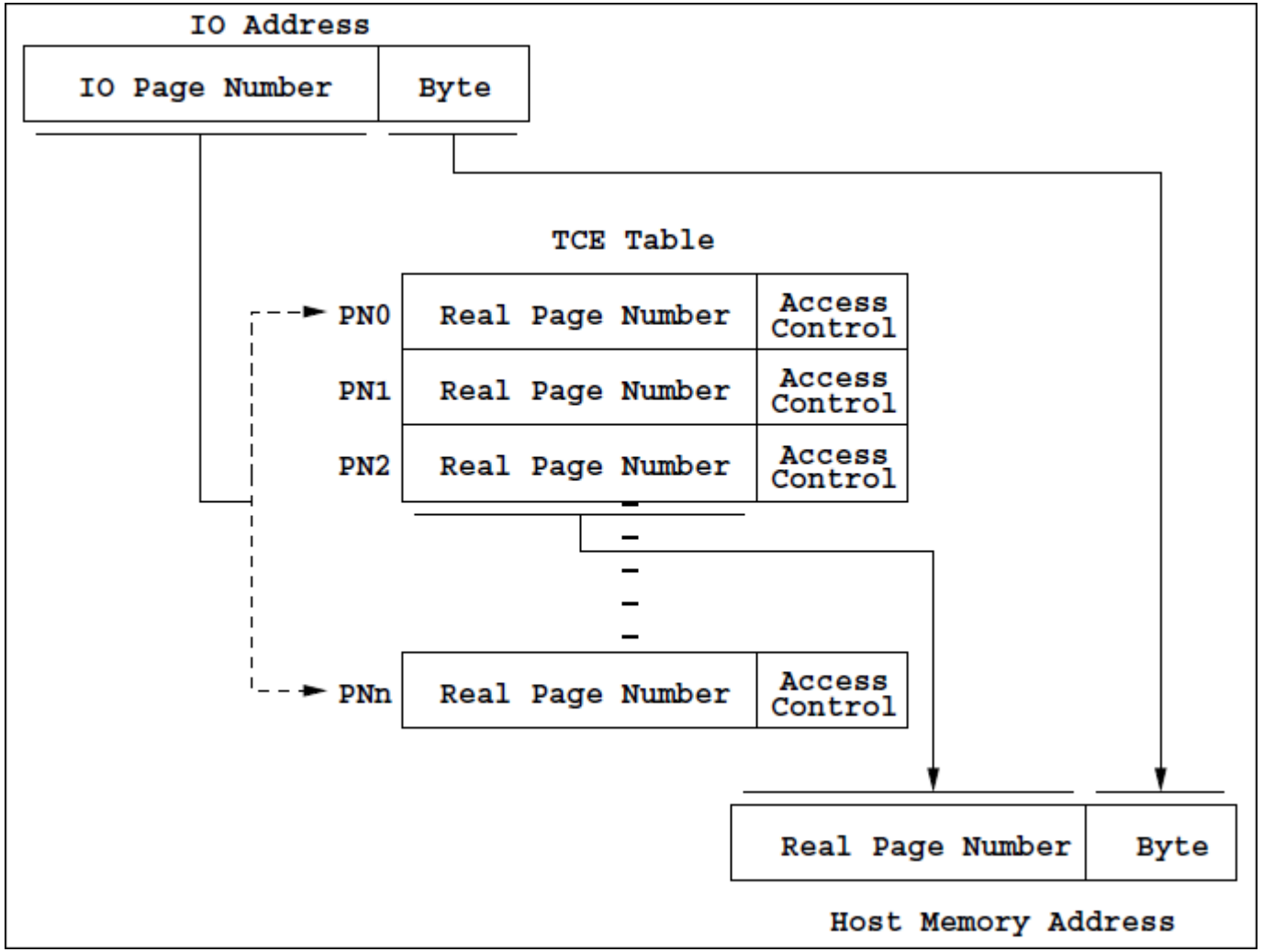


IOMMU

- Device pass through
  - Directly assign a physical device to a particular guest OS
  - Address space translation handled transparently
- Device isolation
  - Safely map a device to a particular guest without risking the integrity of other guests

IOMMU

- Translation Control Entry
  - Translation from a DMA address to a host memory address



Security Problems

- Transience
  - Large numbers of machines appear and disappear from the network sporadically
- Diversity
  - Long and painful upgrade cycles
- Identity
  - Difficult to establish who owns a VM running on a particular physical host
- Mobility
  - Can be easily copied over a network or carried on portable storage media

Discussion

Thanks!