Introduction - Virtualization

Virtualization

- Software Virtualization
- Hardware virtualization

Sources

- http://www.slideshare.net/alanmcsweeney/an-introduction-to-server-virtualisation
- www.cs.usfca.edu/~cruse/cs686s07/lesson19.ppt
- http://www.slideshare.net/ACMBangalore/virtualization-tutorial-at-acm-bangalore-compute-2009
- https://www.slideshare.net/sidhushahbaz/handout2o
- https://www.csa.iisc.ac.in/~vg/teaching/E0-253/slides/VirtualMachines.pptx

Introduction to Server Virtualization

A loose definition

Virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.

VM is an isolated duplicate of the physical machine

Without virtualization (bare metal)

Applicati
on
1

Application
2

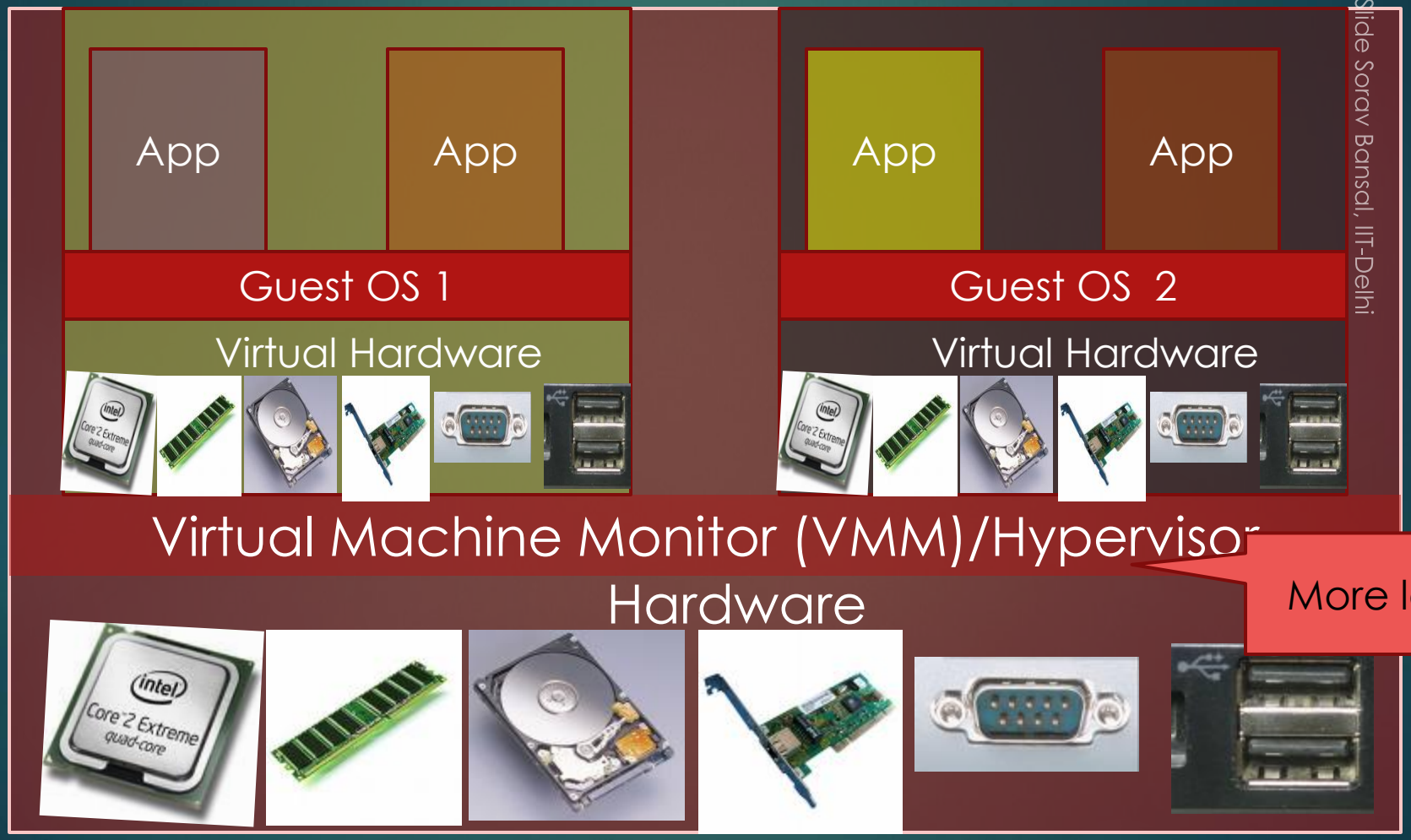
OS

Hardware





With virtualization



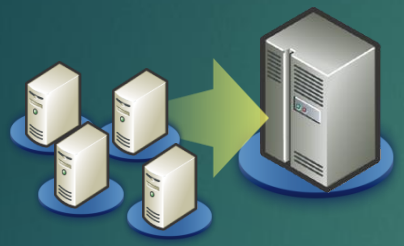
Virtualization and Cloud Computing

- A key technology in Cloud Computing
- Enables different users to share the same physical resources

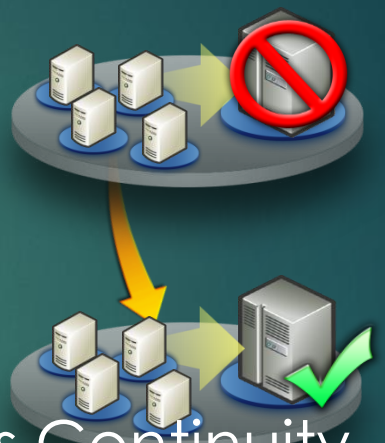
Some history

- An old concept – first virtual machines created on IBM mainframes in early ’60s
- Typically, IBM's virtual machines were identical "copies" of the underlying hardware. Each instance could run its own operating system.
- Virtualisation formed the basis of “time sharing”

Usage Scenarios for Virtualization



Consolidation



Business Continuity Management

Workload Mobility

Development and Test

Why Virtualization

- Server consolidation
- Workload mobility
- Development and test

- Increase hardware utilization by running multiple applications in isolation on same physical server
- Move applications from one server to another
- Easily provision virtual resources for test and dev

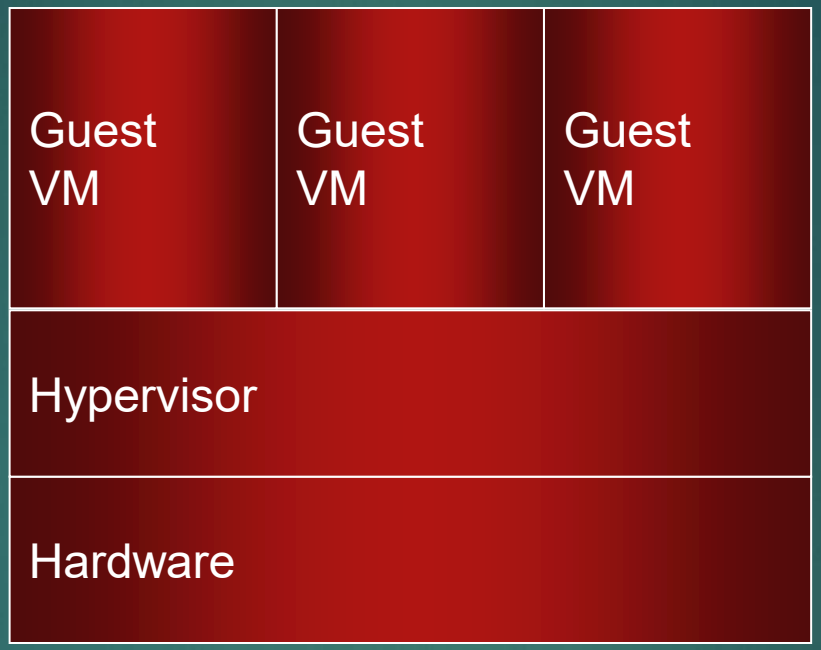
Are there other types of VMs?

- Language VM: Language runtime focused on running a single application
  - e.g., Java Virtual Machine, Microsoft Common Language runtime, Javascripts
  - Not in this **course**
- "Lightweight" VM: Does not run guest OS; But isolates applications from other
  - e.g., Docker – covered earlier
- This lecture is only on System virtual machine: presents a "copy" of whole machine

How is it implemented?

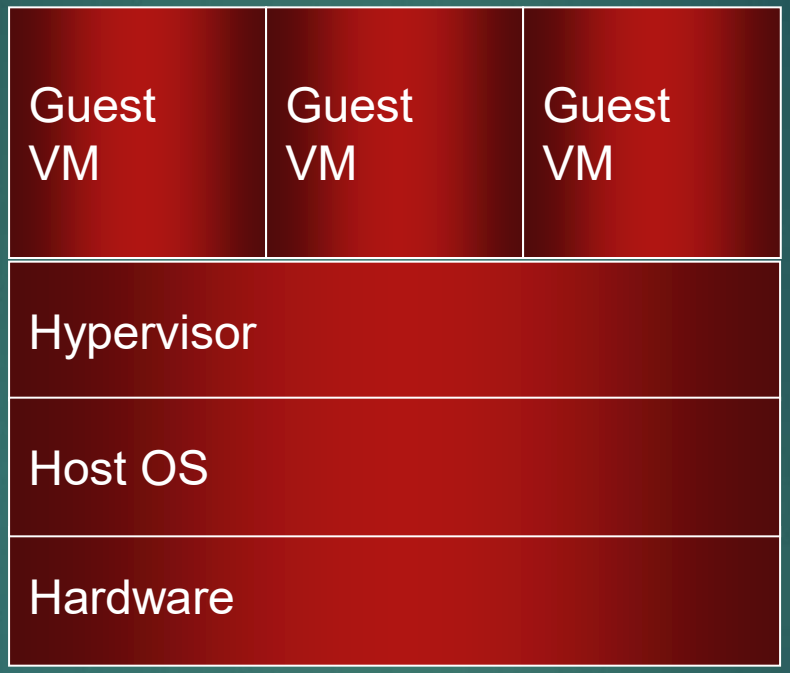
- Virtualization usually involves software called a Virtual Machine Monitor (VMM) or hypervisor that creates the illusion of real machines for multiple virtual machines.
- The hypervisor can run directly on hardware or as an application on a host operating system.

Bare Metal Hypervisors (Type 1)



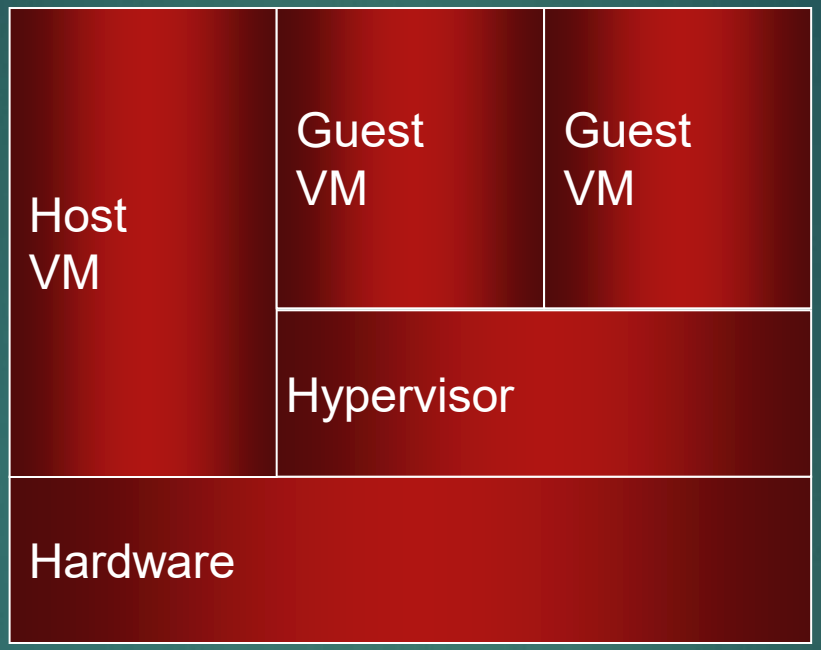
IBM CP/CMS
VMware ESX
Windows Virtualisation (2008)
Xen
Virtual Iron

Hosted Hypervisors (Type 2)



VirtualBox
VMWare Workstation

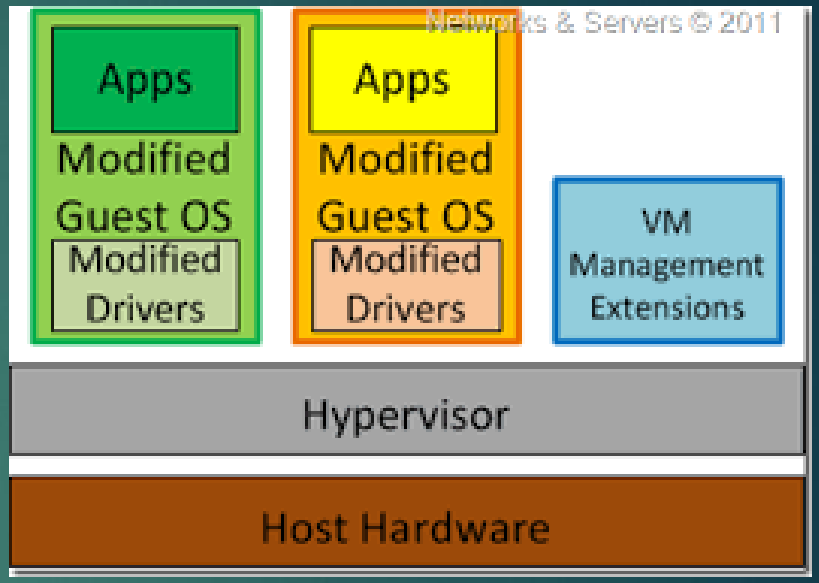
Hybrid Hypervisors



MS Virtual Server
MS Virtual PC

Types of Virtualization

- Paravirtualization
  - Modify OS to run on a hypervisor
    - Xen, Linux-kvm
  - Useful if source code of OS is modifiable
    - IBM: MVS, VM
    - Linux
    - Microsoft: Windows
- Full (transparent) virtualization
  - OS runs without modification
    - VMWare
    - kvm



Types of virtualization

- Bare-metal vs hosted
- What is paravirtualization?
- Pros and cons of paravirtualization

- Bare-metal – runs on bare machine. Hosted runs on OS
- Paravirtualization: VMM provides APIs for guest OS
- Pros: can be more efficient. Cons: Need to modify guest OS

Implementing a VMM

- In order to virtualize, we must look at following
  - Instructions -- defined by the ISA (e.g., x86, ARM)
  - Memory
  - I/O (network, disk)

- Let's first look at Instructions

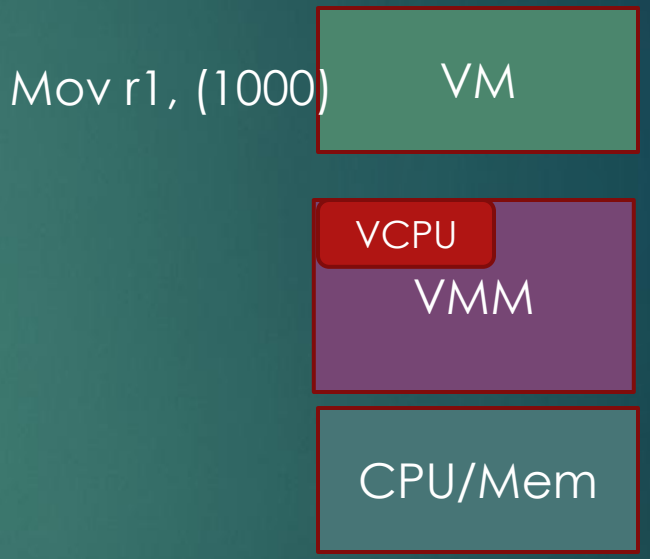
Server Virtualization Techniques: Trap and Emulate Virtualization

VM requirements

- Guest OS (and process on guest OS) needs to execute on the processor
- One Guest OS should not interfere with another – in memory and I/O

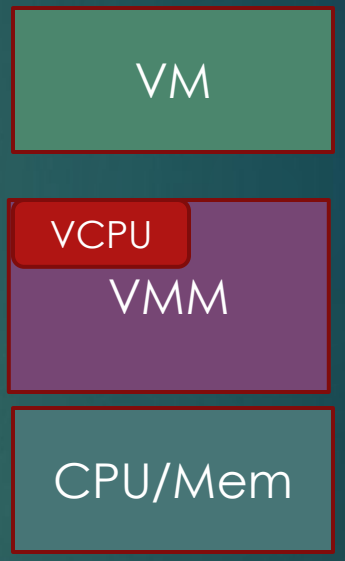
How to run the VM?

- The VMM abstracts a CPU
  - Just like CPU reads every instruction, interprets and executes it
  - VMM is responsible for doing the same
- One way to do it is to
  - Simulate every instruction
  - But this would be very slow
  - Examples: Bochs (http://bochs.sourceforge.net/)



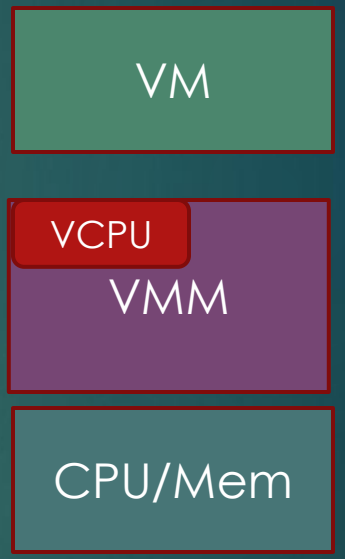
How to run the VM?

- The VM is also code
- Why can we not run the VM on the CPU
- Example
  - addl %eax, %ecx
- Should be able to run fine on the hardware
- Can you think of examples of code that cannot run on the hardware?



What cannot run directly?

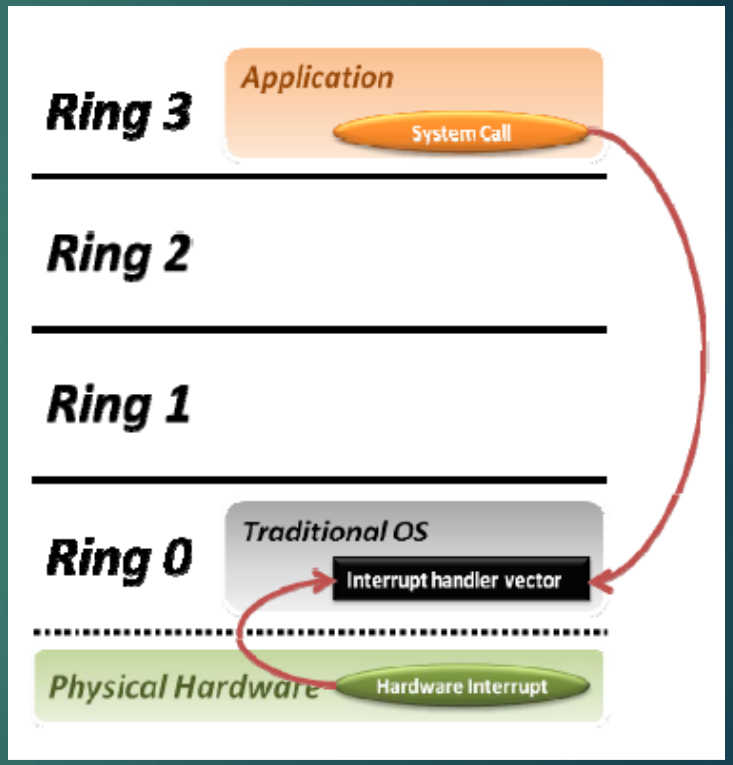
- Any code that requires higher privileges to run
- For example
  - syscall
  - Movl something, %cr3
  - I/O



CR3 is the page table.

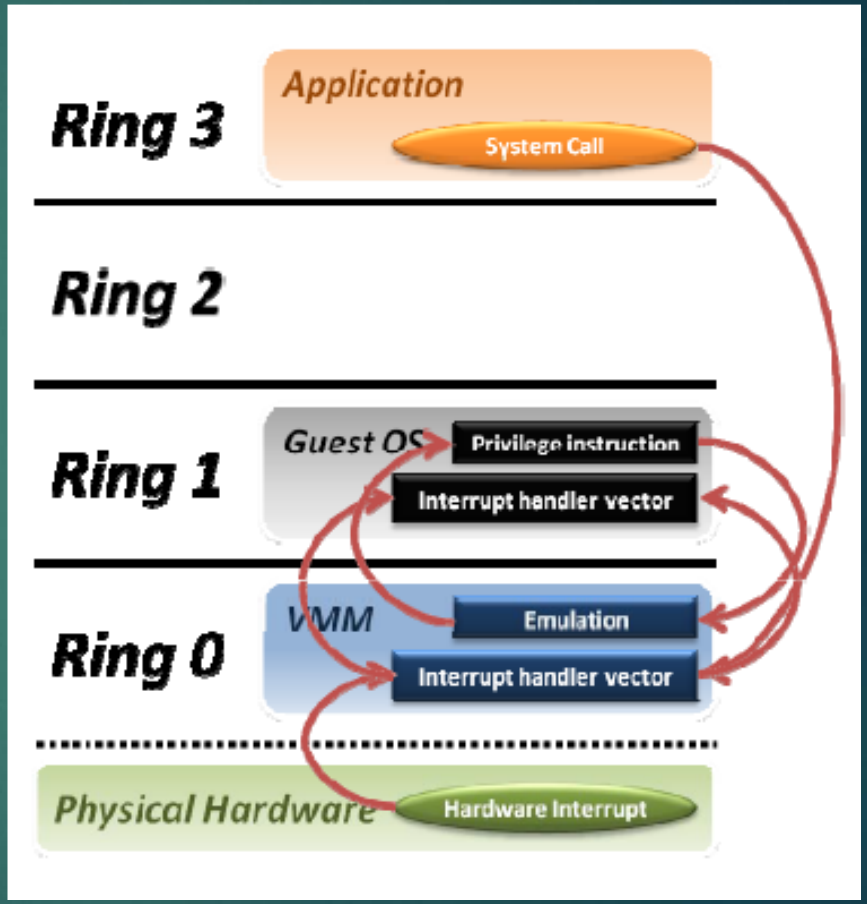
Why cannot it run directly?

- Privilege Rings in Hardware
- Typically OS runs in high privilege mode (Ring 0)
- And application in Ring 3 (less privilege)
- But with a VMM, the
  - VMM runs in ring 0
  - Which ring does Guest OS run in?



Trap and Emulate

- Run the Guest OS in Ring 1
- On executing a privileged instruction in Ring 1
  - Trap into the VMM
  - VMM emulates the instruction
    - Does everything that the original instruction was supposed to do but without executing it
  - Then go back to the Guest OS



Trap and Emulate: Earliest Virtualization Technique

Transparent virtualization

Hypervisor traps privileged instructions and emulates

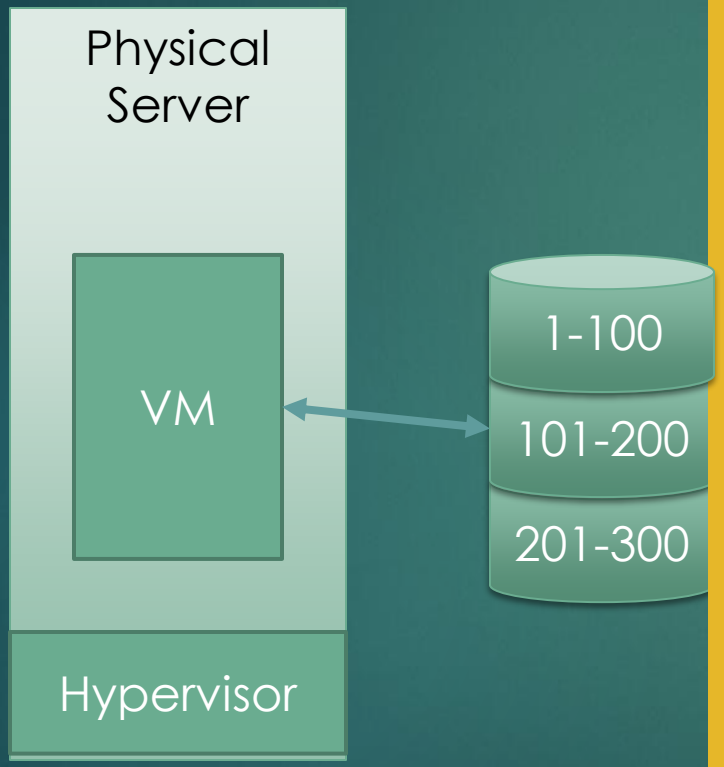
- Access to physical pages
- Physical I/O devices
- Control registers

Not trapping privileged instructions could lead to security breach

Handling privileges

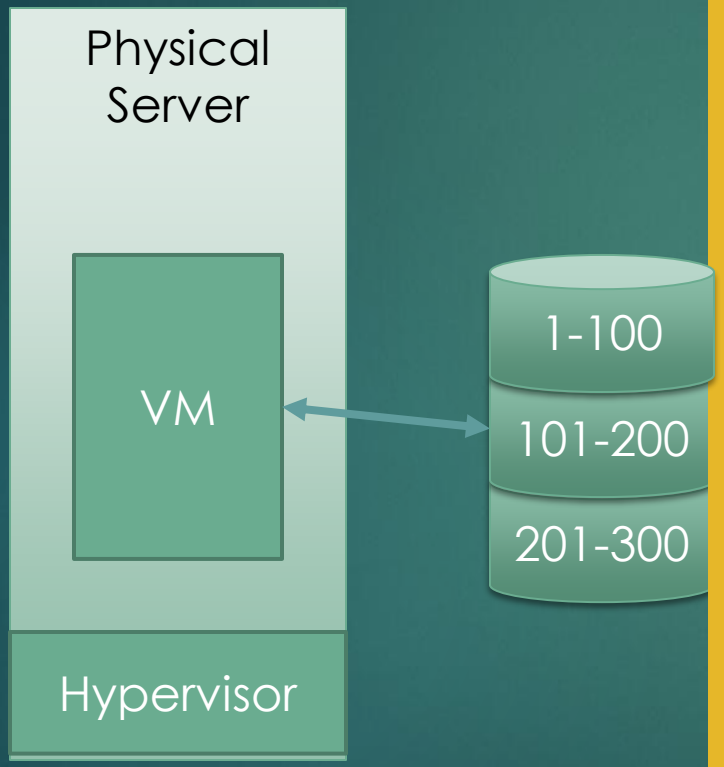
- All processors have rings of privilege
- Run hypervisor in highest privilege ring (ring 0)
- Run guest in lower ring

Exercise 1 (10 minutes)



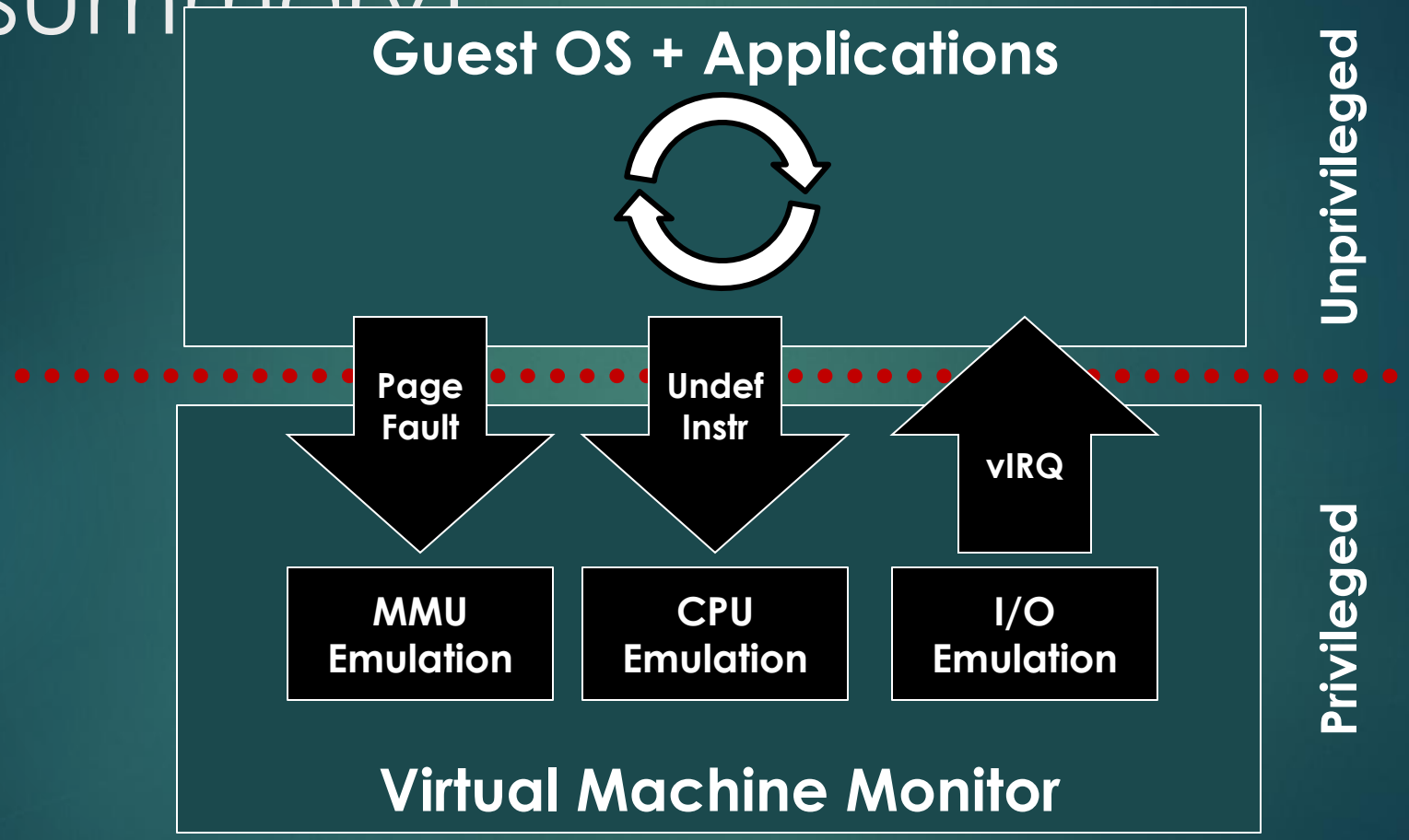
- Disk has 3 virtual disks – cylinders: (1) 1-100, (2) 101-200, (3) 201-300
- VM is connected to virtual disk (2)
- VM executes an instruction to read cylinder 1, block 5 of virtual disk (2)
- Why is this a privileged instruction?
- What happens when VM executes instruction?

Exercise 1 solution



- VM executes an instruction to read cylinder 1, block 5 of virtual disk (2)
- Why is this a privileged instruction?
  - Otherwise, VM could read virtual disk (1) or (3). These could belong to other VMs
- What happens when VM executes instruction?
  - Hypervisor traps instruction, replaces it with an instruction to read cylinder 101 block 5 of physical disk

Trap and Emulate (summary)



Source: https://labs.vmware.com/download/45, Scott Devine, VMWare

Limitations of Trap and Emulate Virtualization

- Performance Overhead
  - E.g., trapping privileged instructions
- Some architectures cannot be virtualized by use of trap and emulate virtualization
  - Depends on the fact that executing an instruction at a lower privilege than required will cause a **trap** to the VMM.

True or False?

- Trap and emulate virtualization does not need any special support
- Trap and emulate virtualization is very efficient

- False: Trap and emulate virtualization requires CPU to support access rings.
- False: for many operations, there is a significant overhead