Lecture 11: Virtualization Technology and Cloud Infrastructure

CSCI4180 (Fall 2013)

Patrick P. C. Lee

References

- Slides are adapted from Dr. T Y Wong's slides in 2012 spring course

- VMWare, "Understanding Full Virtualization, Paravirtualization, and Hardware Assist", 2007
  - http://www.vmware.com/resources/techresources/1008

- VMWare, Introduction to Virtualization
  - http://labs.vmware.com/academic/introduction-to-virtualization

What is Virtualization?

- **Virtualization** provides abstractions of hardware resources for multiple operating systems to share the same pool of hardware resources

OS level Virtualization

- Modern OSes contain a simplified version of virtualization:

real hardware is transparent

  - CPU virtualization: multiple processes are scheduled to consume a CPU with fair share

need a scheduler to make VMs use resource in a fair way

  - Memory virtualization: a process has its own virtual address space, which is mapped to the physical address space in memory

use a mapping table to manage memory

Why Virtualization?

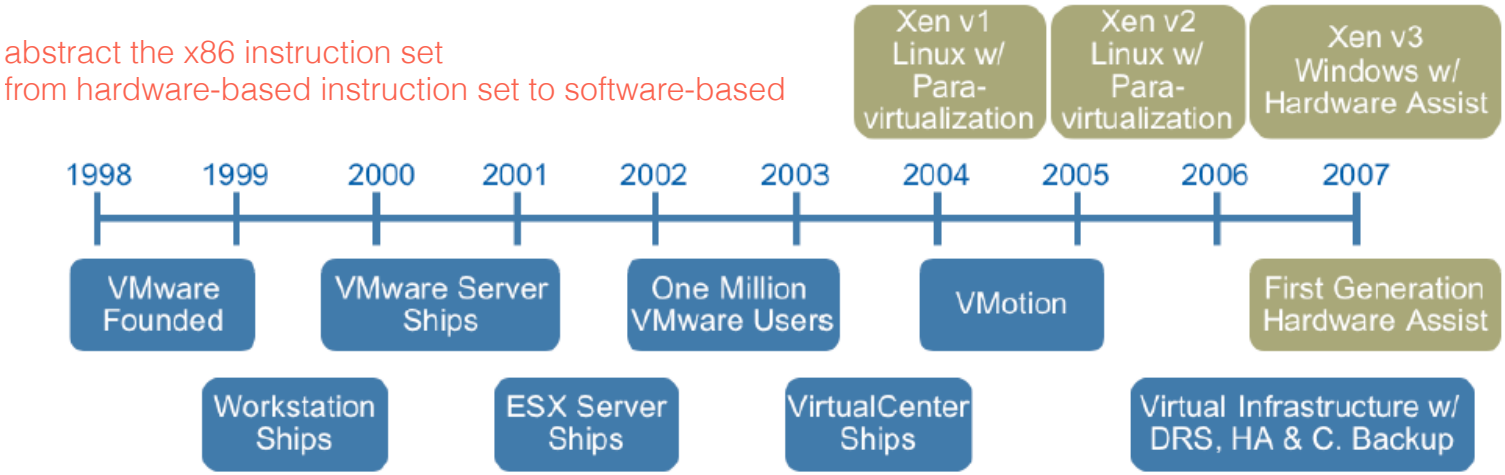
- Better resource utilization, as computers have more resources than one task needs
- Consolidation of multiple virtual servers on a physical server (i.e., reduce hardware footprints)
- Easy to migrate
- Easy to clone
- Reduced power usage
- Better isolation

Good in terms of management

Bad in performance

x86 Virtualization

- x86 architectures have complicated instruction sets that are difficult to virtualize
- In 1998, VMWare provided virtualization solutions for x86 architectures
- Timeline of virtualization development:



x86 refers to a series of computer microprocessor instruction set architectures based on the Intel 8086 CPU

x86 Virtualization

- A virtualization layer is added between the hardware and operating system
- It's a software responsible for hosting and managing all virtual machines on virtual machine monitors (VMMs) a real resource manager provide "hardware" to VMs
- Partition resources such as CPU, storage, memory, and I/O devices

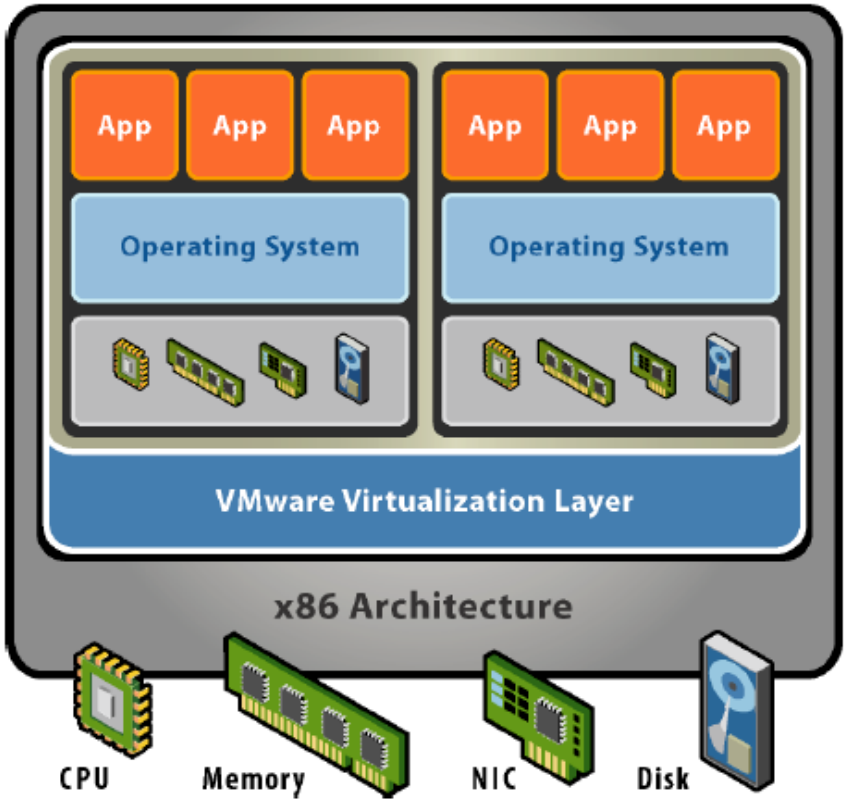


Figure 2 – x86 virtualization layer

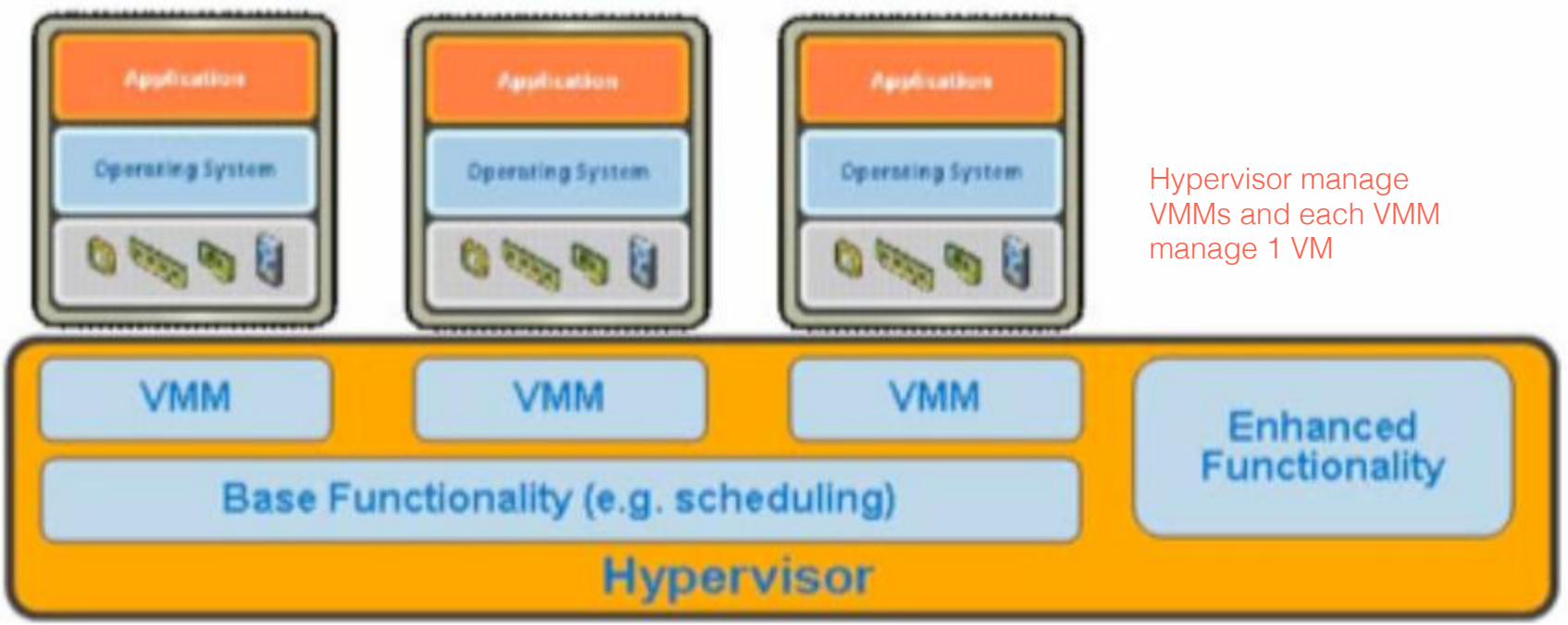
x86 Virtualization

- What is a VMM? provide a same environment to the VMs

> A virtual machine is taken to be an *efficient, isolated duplicate* of the real machine. We explain these notions through the idea of a *virtual machine monitor* (VMM). See Figure 1. As a piece of software a VMM has three essential characteristics. First, the VMM provides an environment for programs which is essentially identical with the original machine; second, programs run in this environment show at worst only minor decreases in speed; and last, the VMM is in complete control of system resources.

- VMM properties:
  - Fidelity
  - Performance
  - Safety and isolation

x86 Virtualization



Hypervisor manage VMMs and each VMM manage 1 VM

Figure 3 – The hypervisor manages virtual machine monitors that host virtual machines

x86 Virtualization

- In x86 systems, virtualization approaches use either a hosted or a hypervisor architecture.
- Hosted architecture
  - installs and runs the virtualization layer as an application on top of an operating system
  - supports the broadest range of hardware configurations
  - Examples: VMWare Player, VMWare Workstation, and VMWare Server

type 1 , type 2 virtualization

x86 Virtualization

- Hypervisor architecture

唔經 OS 直接去 control hardware
可視為一個 tailor made OS

  - The hypervisor (bare-metal) architecture installs the virtualization layer directly on a clean x86-based system.
    - Its installer is usually an ISO that installs a tailor-made OS.
  - Since it has direct access to the hardware resource rather than going through an operating system, it's more efficient than a hosted architecture and delivers greater scalability, robustness and performance.
  - Examples: VMWare ESX server.

bad: worse portability than using Hosted Architecture
because it is designed for hardware…

Challenges of x86 Virtualization

- x86 architectures offer four privilege levels: **Ring 0, 1, 2, and 3**
  - The lower the ring level, the higher privilege
  - Ring 0: where OS runs
  - Ring 3: where user-level application runs

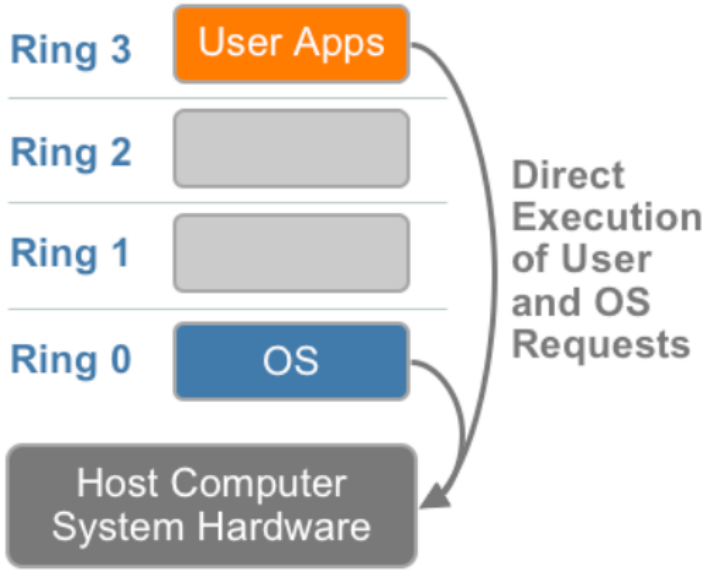


Figure 4 – x86 privilege level architecture without virtualization

Challenges of x86 Virtualization

need to implement the Virtualization below the OS i.e. RIng 0

- Virtualizing the x86 architecture requires lacing a virtualization layer under the operating system (Ring 0)

- Further complicating the situation, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0

- It's challenging to trap and translate these sensitive and privileged instruction requests at runtime

Challenges of x86 Virtualization

- There are 3 implementations:
  - **Full virtualization** with binary translation and direct execution, e.g., VirtualBox and VMWare player
  - **Para-virtualization**, e.g., Xen. below OS
  - **Hardware-assisted virtualization**, e.g., Intel VT-x and AMD-V.

Full Virtualization

- The full virtualization approach can virtualize any x86 OS
- It combines:
  - **Binary execution**: VMM translates kernel code to replace non-virtualizable instructions with new instructions intended for virtualization
  - **Direct execution**: User level code is directly executed on the processor for high performance

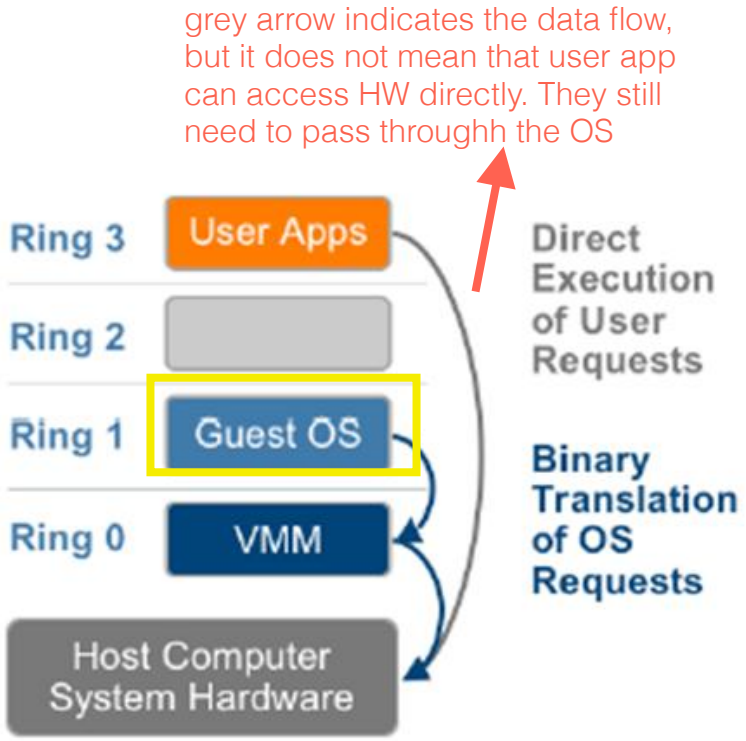
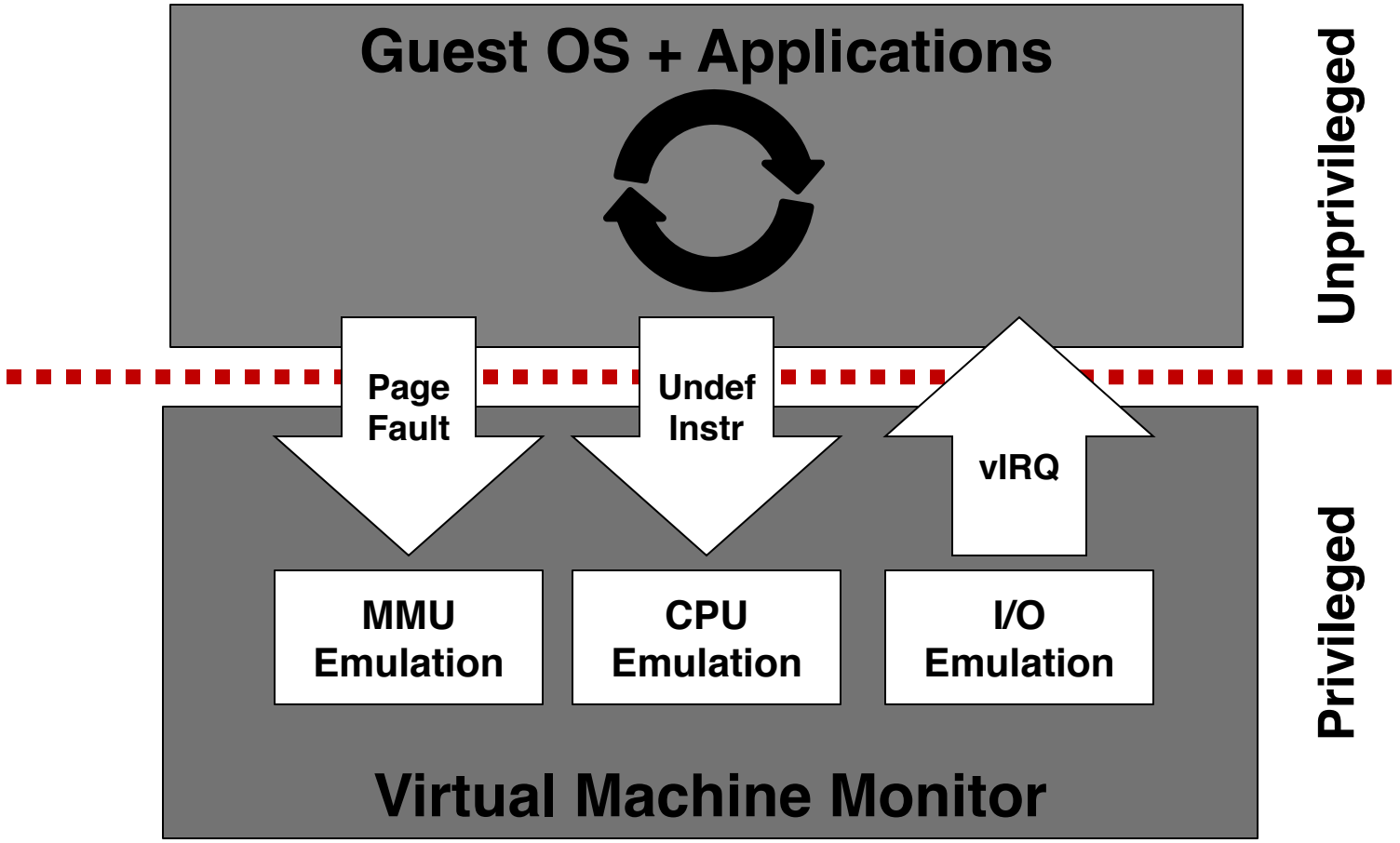


Figure 5 – The binary translation approach to x86 virtualization

Full Virtualization



Trap and Emulate

Full Virtualization

- Guest OS is fully abstracted from the underlying hardware
  - No modification of the guest OS is needed

keep guest OS unchange

- Offers the best isolation and security
- Simplifies migration as guest OS can remain unmodified
- Drawback:
  - Performance is not good.

Paravirtualization

- "Para-" means "beside", "with", or "alongside"
- Paravirtualization is also known as "OS-assisted virtualization"
- It involves modifying the OS kernel to replace non-virtualizable instructions with **hypercalls** that communicate directly with the virtualization layer

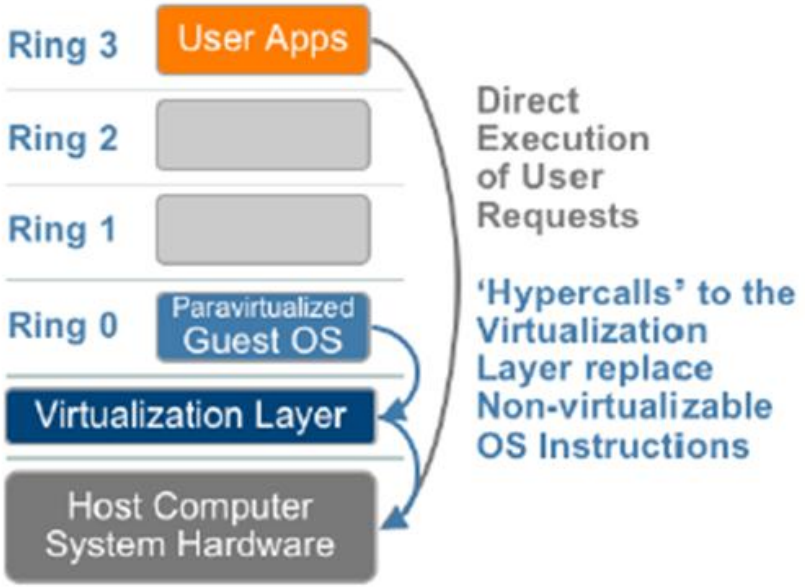


Figure 6 – The Paravirtualization approach to x86 Virtualization

hypercall: a kind of system call
OS try to issue this directly to the Virtualization layer which can "talk" to hardware

Paravirtualization

- Paravirtualization is different from full virtualization:
  - Depends on how they face non-virtualizable instructions.

Full virtualization	**Paravirtualization**
Translate those instructions during running time.	Translate those instructions during compile time.
Pros: OSes which cannot be modified (e.g., WinXP) can become guest OSes.	Pros: fast because it has a set of compiled hypercalls to speed up
Cons: Slow	Cons: only modified OSes can be used.

Hardware-Assisted Virtualization

similar to hypervisor arch.

Virtualization Technology

- Intel VT-x and AMD-V
- Enable privileged instructions with a new CPU execution feature that allows the VMM to run in a new root mode below ring 0
- All privileged instructions are set to automatically trapped to the VMM

similar to full virtualization but it adds a special laver for VMM rather than using Ring0

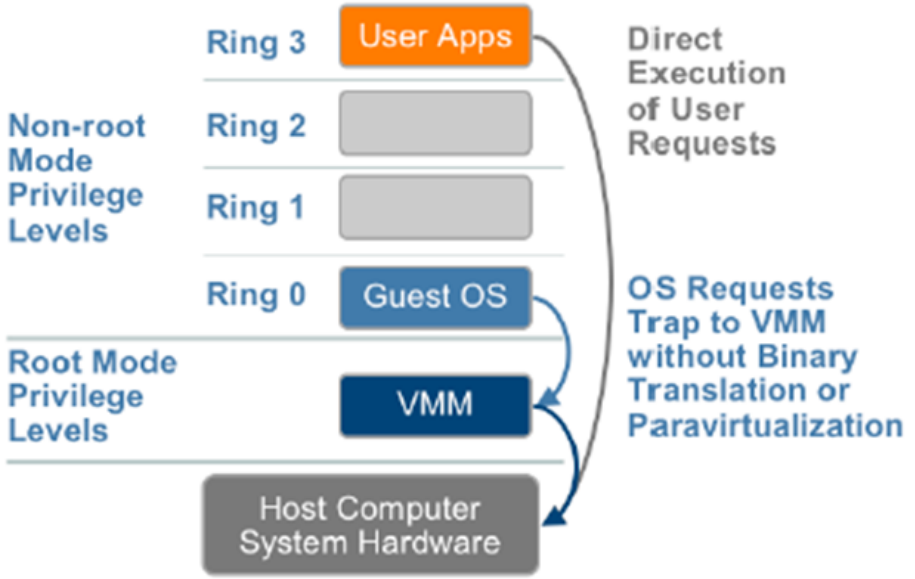


Figure 7 – The hardware assist approach to x86 virtualization

Other Types of Virtualization

- Memory virtualization
  - User programs see logical address space
  - → mapped to physical address space in guest OS
  - → mapped to physical address space in physical host
- I/O virtualization
  - The hypervisor presents each VM with a standardized set of virtual devices

Summary

	Full virtualization	Para-virtualization	Hardware-assisted Virtualization
Technique in handling privilege calls	Binary Translation.	Modified OS with hypercalls.	Switching from non-root mode to root mode automatically.
Guest modification	No.	Yes	No.
Performance	Good	Very good	Good
Used By	VMware, VirtualBox, Microsoft, Parallels.	Xen	QEMU with KVM, VMware, VirtualBox, Microsoft, Parallels, Xen.

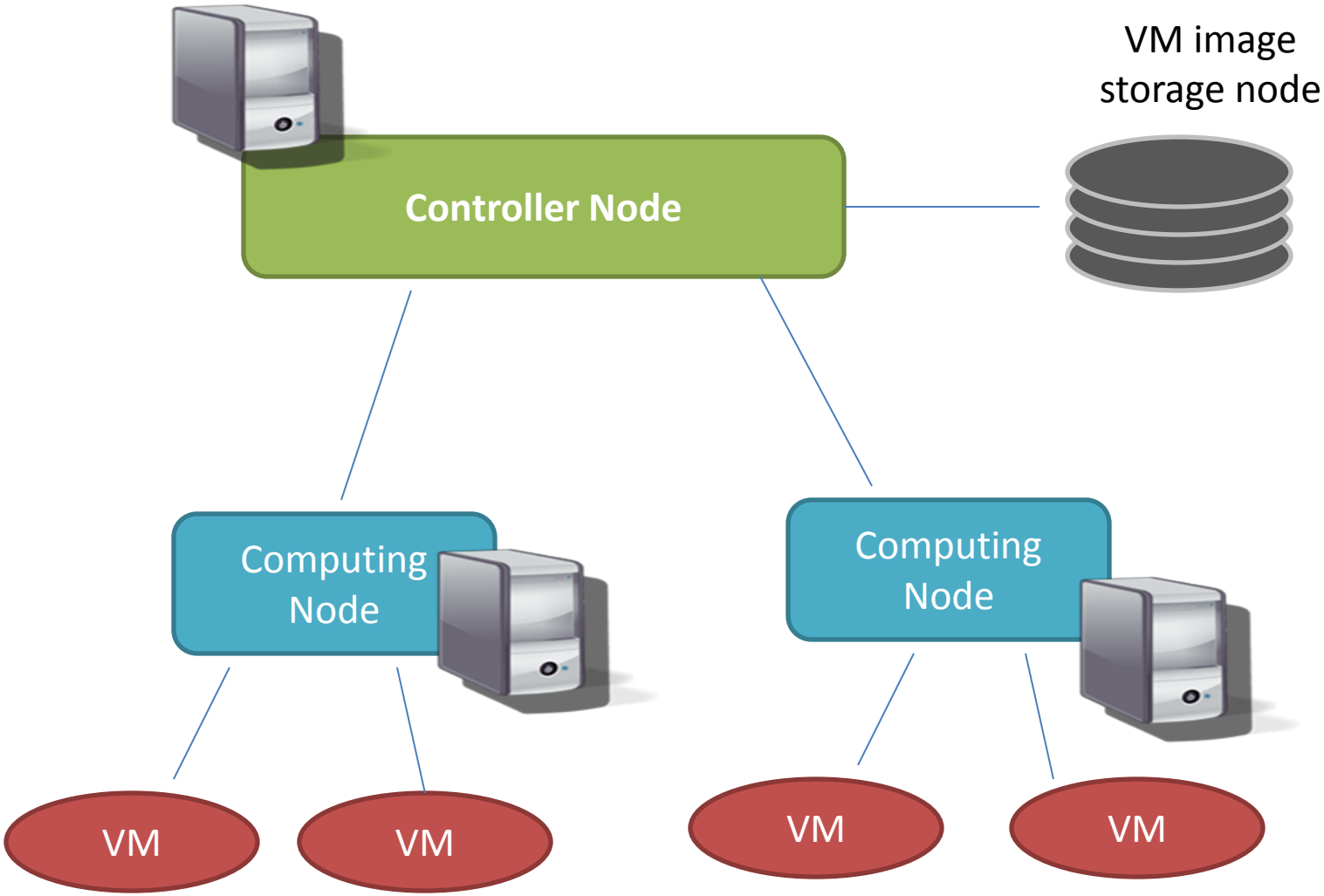
Cloud & Virtualization

- Cloud computing is usually related to virtualization. Why?
  - It is because of its elasticity.
  - Launching new machine under a virtualized environment is fast and cheap.

- Therefore, cloud infrastructure is actually a **virtual machine management infrastructure**.
  - Commercial: AWS EC2, Rackspace.
  - Open source: Eucalyptus, OpenStack.

Dr. TY Wong's Slide in Spring 2012

IaaS – Infrastructure as a Service

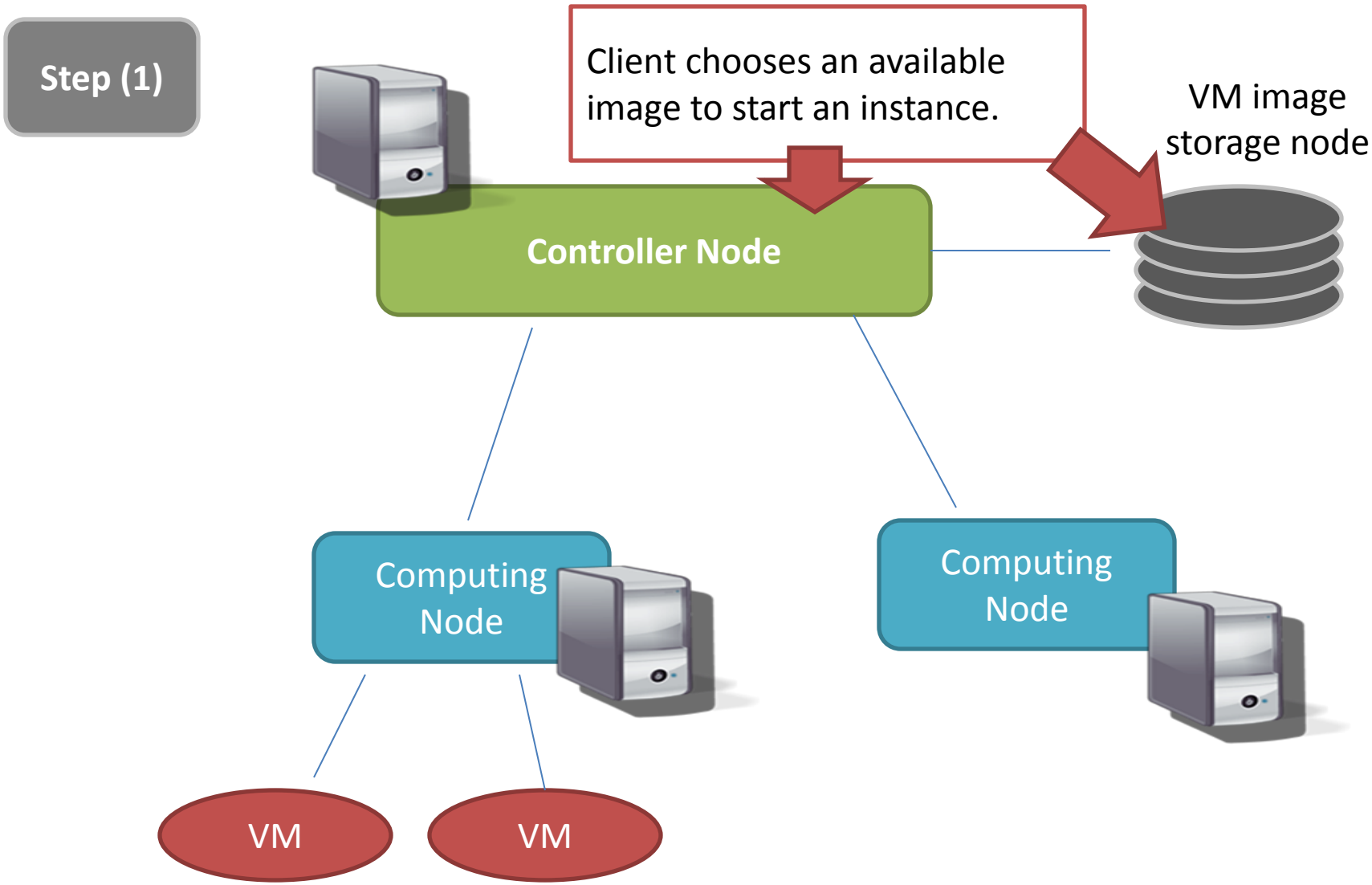


IaaS – Infrastructure as a Service

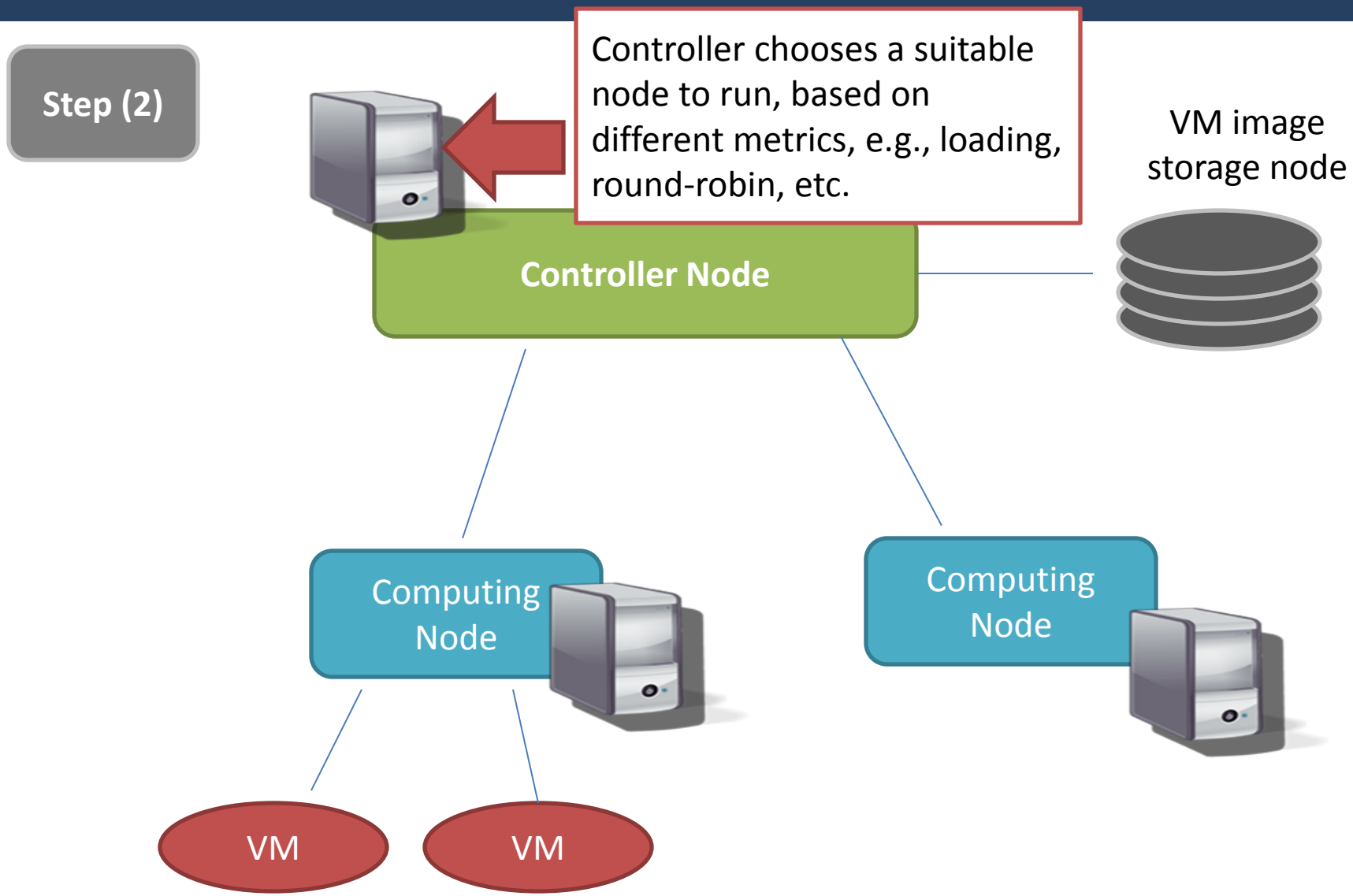
- Eucalyptus VS OpenStack

	Eucalyptus	OpenStack
Controller Node	Cloud Controller (CC)	Compute (Nova)
Computing Node	Node Controller (NC)	Computing Node
Image Storage	Storage Controller (SC)	Image Service (Glance)
Object Storage	Walrus	Object Storage (Swift)
Origin	Emulate AWS EC2	Open source version of RackSpace and NASA's Nebula.

IaaS – Flow

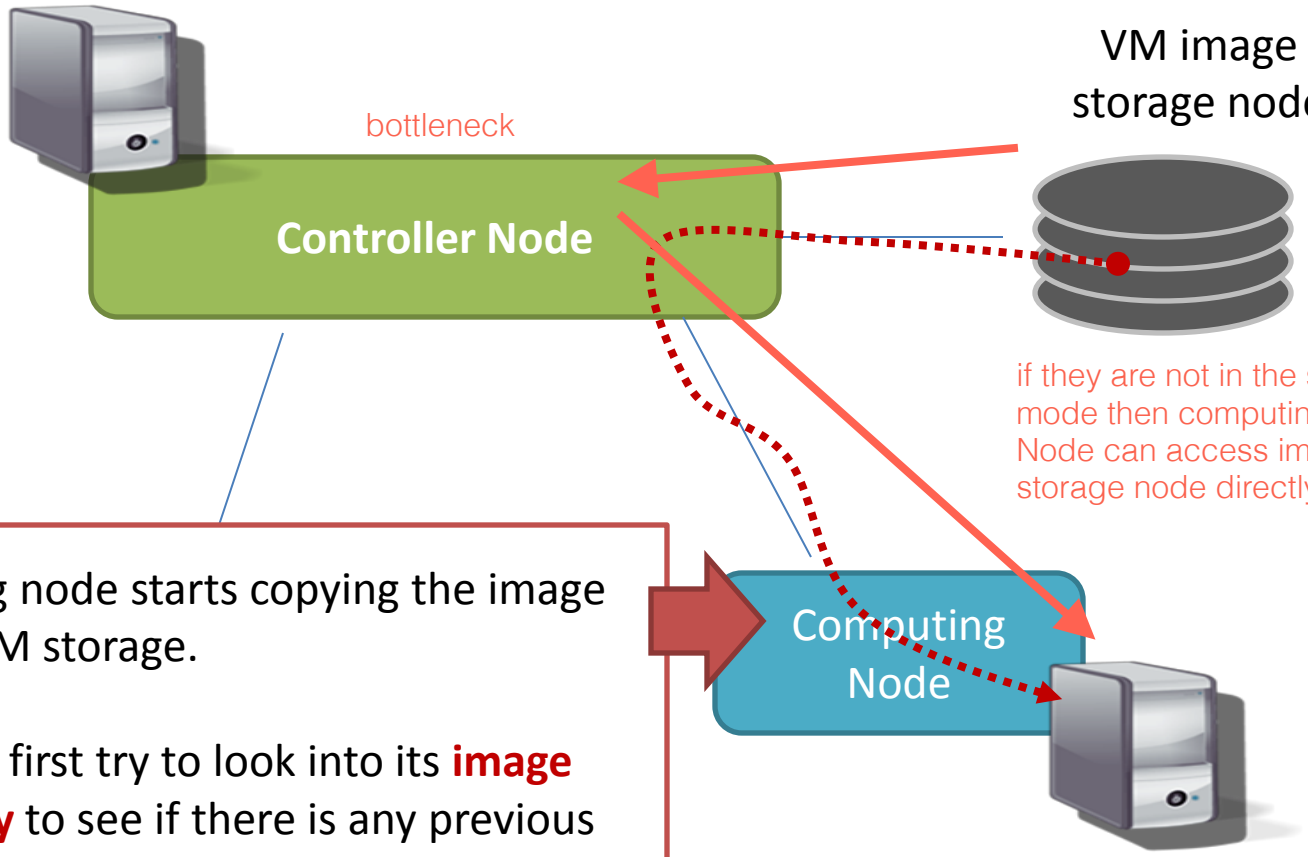

Step (1)
Client chooses an available image to start an instance.
VM image storage node
Controller Node
Computing Node
Computing Node
VM
VM

IaaS – Flow


Step (2)
Controller chooses a suitable node to run, based on different metrics, e.g., loading, round-robin, etc.
VM image storage node
Controller Node
Computing Node
Computing Node
VM
VM

IaaS – Flow

Step (3)



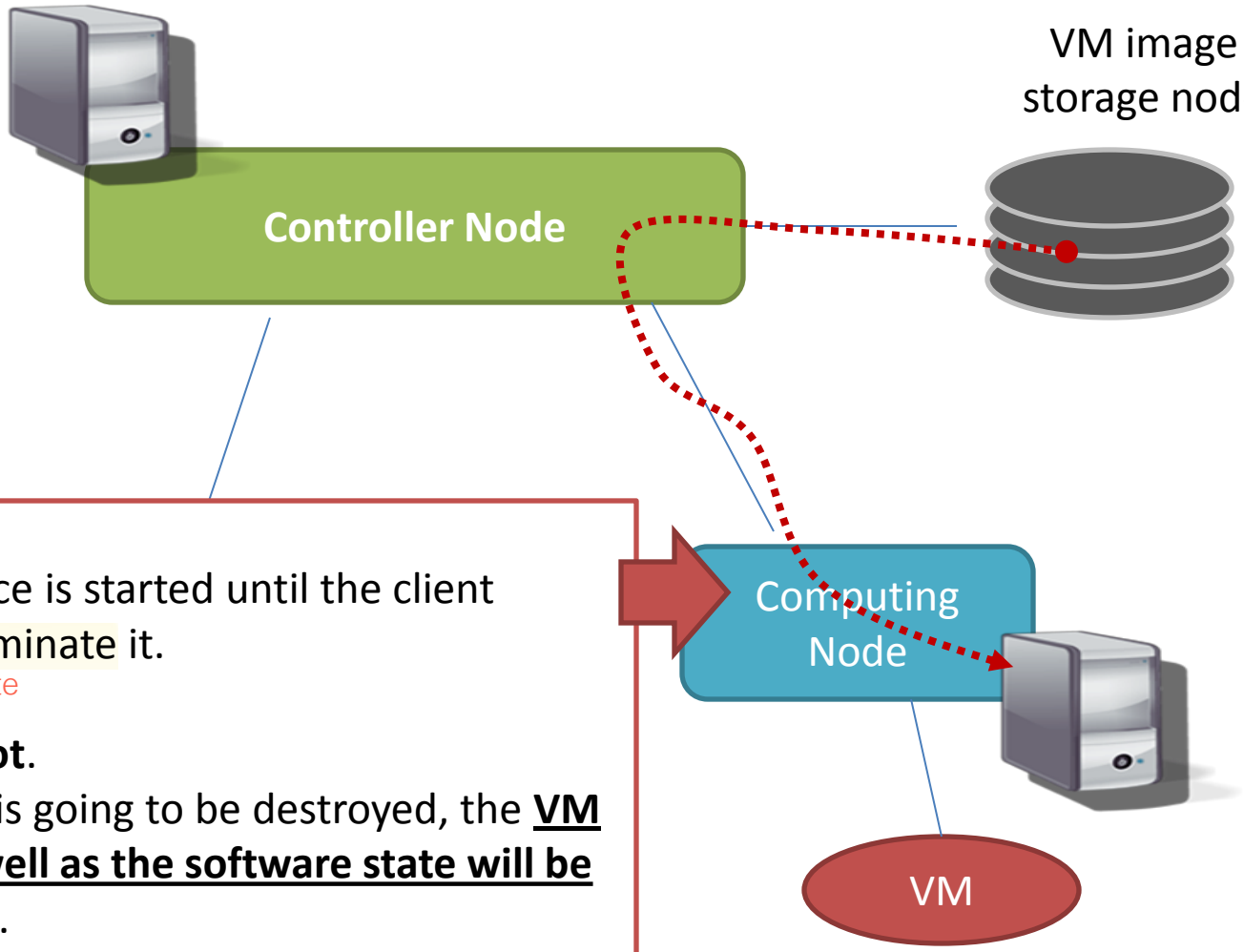
This computing node starts copying the image file from the VM storage.

Eucalyptus will first try to look into its **image cache directory** to see if there is any previous copy of the image resided in the machine.

If yes, the copy is spared. Else, copy.

IaaS – Flow

Step (4)



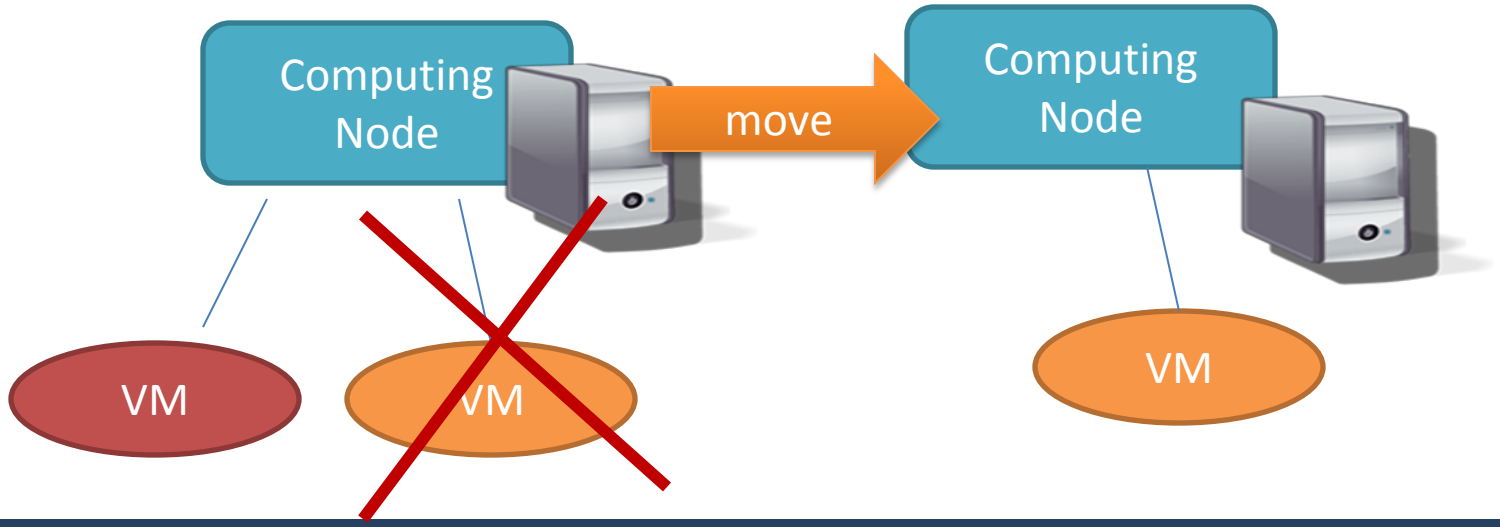
The VM instance is started until the client chooses to terminate it.
delete

Believe it or not.
When the VM is going to be destroyed, the **VM image file as well as the software state will be destroyed**, too.

Subtleties about IaaS

- VM migration.
  - How to do that?
    - Suspend the VM.
    - Extract and move CPU status as well as page frames,
    - Resume the VM on another machine.
  - Anything missing?

only keeps memory and CPU states



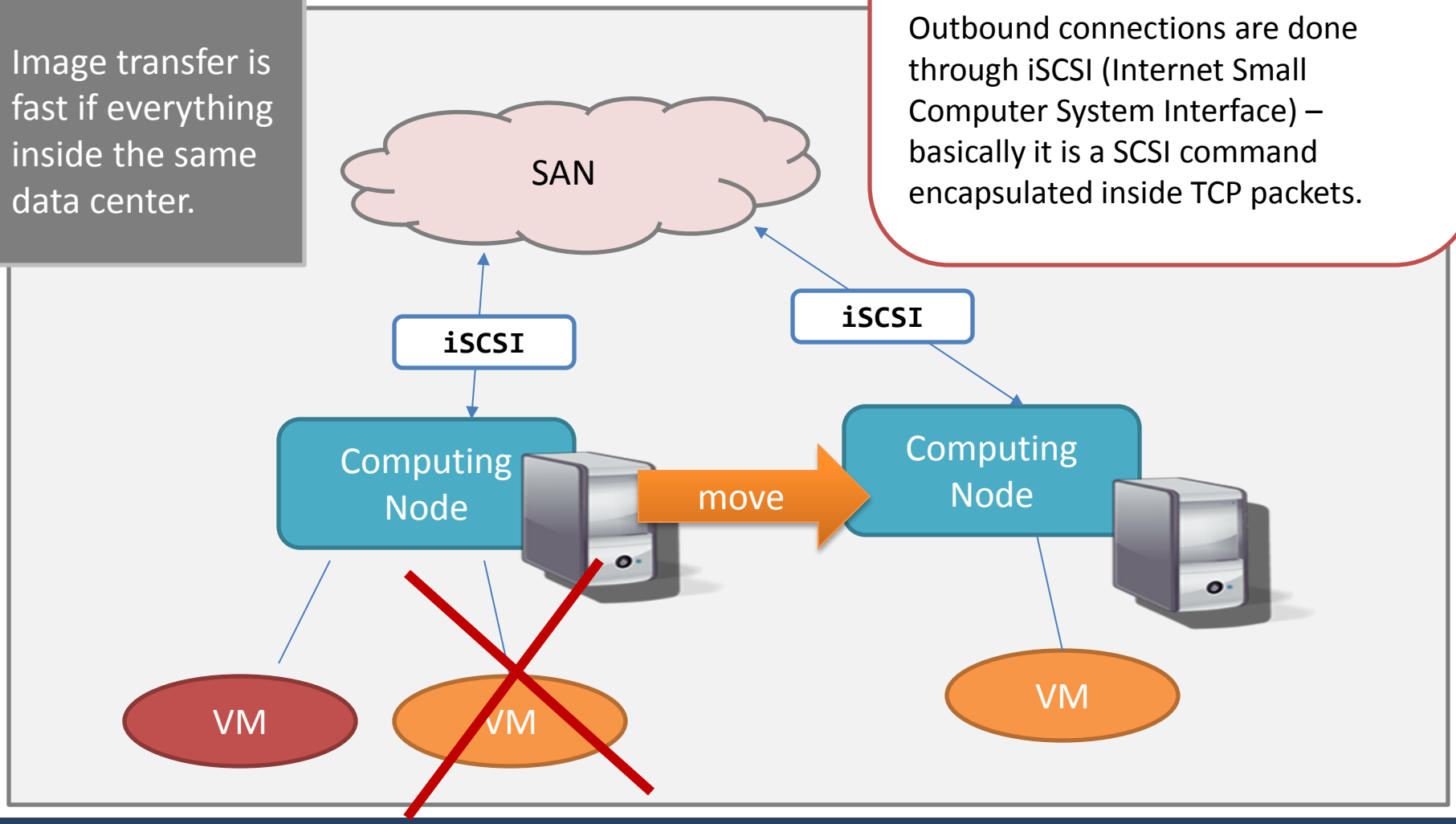
Subtleties about IaaS

- SAN – storage area network.

Image transfer is fast if everything inside the same data center.

It is usually a network of storage servers connected using fabric channel.

Outbound connections are done through iSCSI (Internet Small Computer System Interface) – basically it is a SCSI command encapsulated inside TCP packets.



Subtleties about IaaS

- Virtualization technique used?
  - It depends…
  - If **Windows** is involved, Full or Hardware-assisted virtualization is needed.

- E.g., AWS EC2:
  - Backend is Xen.
  - Xen provides both para-virtualization and hardware-assisted one (through QEMU).

Subtleties about IaaS

Domain 0 (Dom0) manages emulated devices.

Bypassed if KVM is used.

Domain Unprivileged (DomU)