Queen's School of Computing

Queen's University

# Clouds Under the Covers

KHALID ELGAZZAR
GOODWIN 531
ELGAZZAR@CS.QUEENSU.CA

# References


Understanding Full Virtualization, Paravirtualization, and Hardware Assist

White Paper, VMware Inc.

Virtualization Overview

White Paper, VMware Inc.

The Eucalyptus Open-Source Cloud-Computing System

D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov. *CCGRID 2009*.


Based on materials from:
Introduction to Virtual Machines by Carl Waldspurger
R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing Morgan Kaufmann, 2013.

# Outline

1. Virtualization
2. Cloud Infrastructure (IaaS Stacks)

# 1. Virtualization

VIRTUALIZATION AND VMS

MEMORY VIRTUALIZATION

PROCESSOR VIRTUALIZATION

I/O VIRTUALIZATION

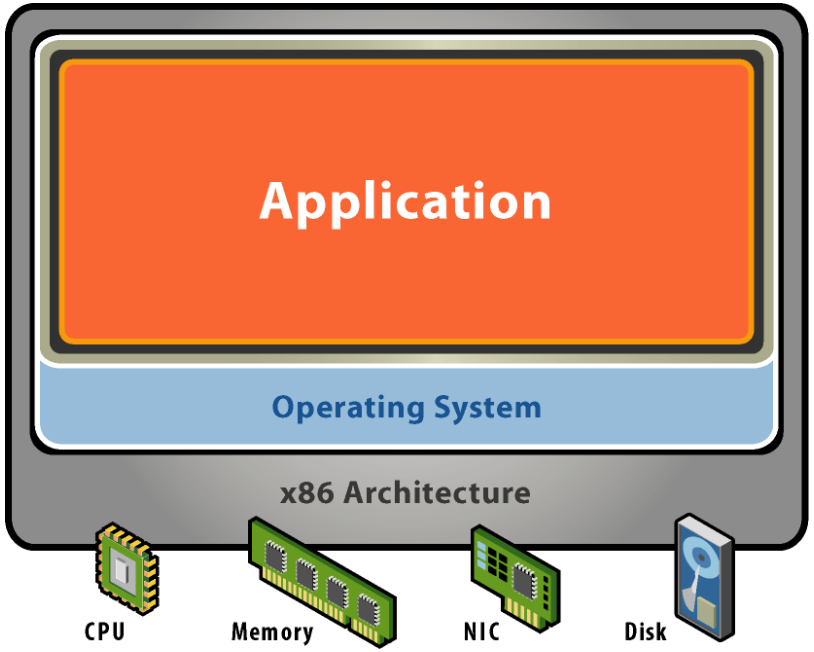
# Virtualization

***Virtualization*** broadly describes the separation of request for resources from the underlying physical delivery of that resource.

Example, *virtual memory* gives programs access to more memory than is physically installed via the background swapping of data to disk.

Virtualization techniques can be applied to other IT infrastructure layers – hardware, networks, storage, operating systems and applications.

# Starting Point: A Physical Machine




## Physical Hardware

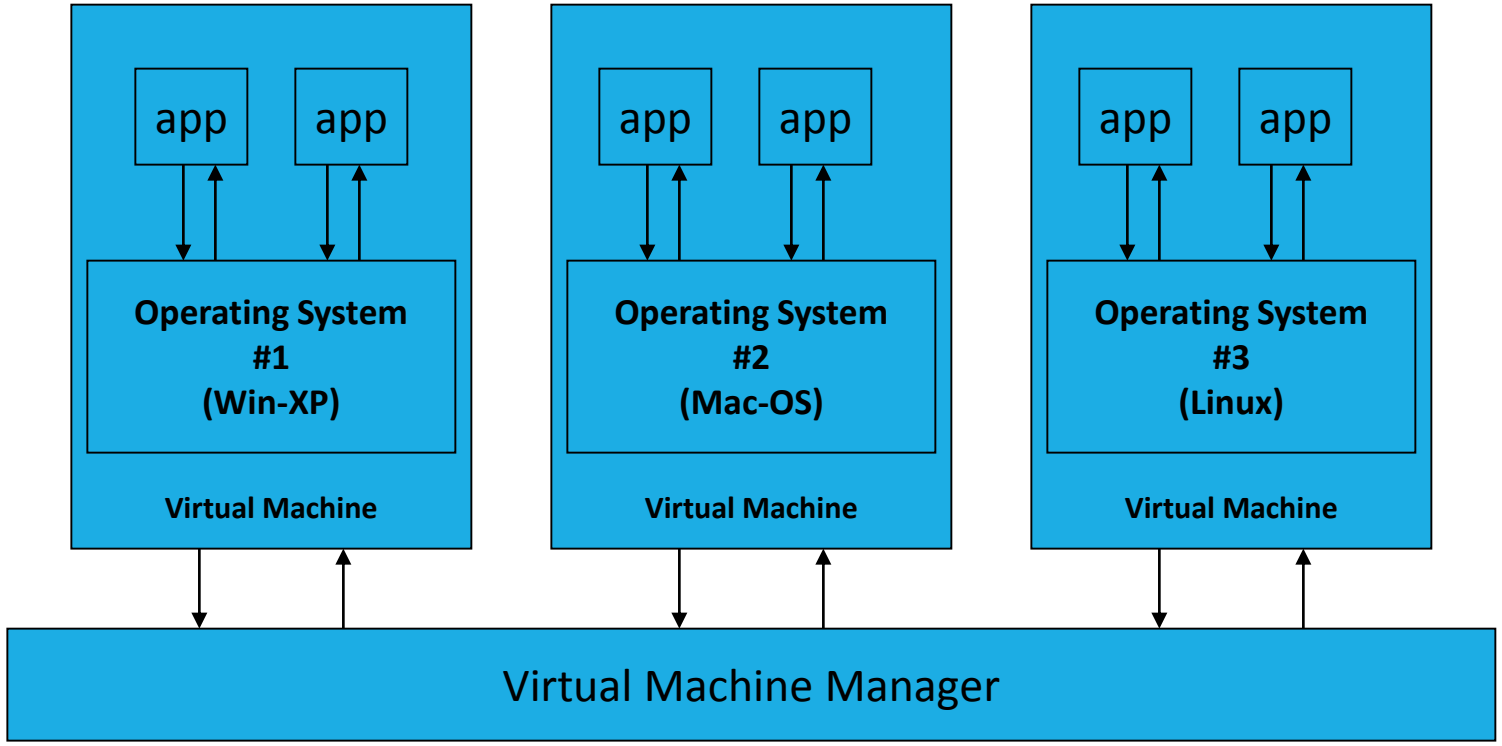
- Processors, memory, chipset, I/O devices, etc.
- Resources often underutilized

## Software

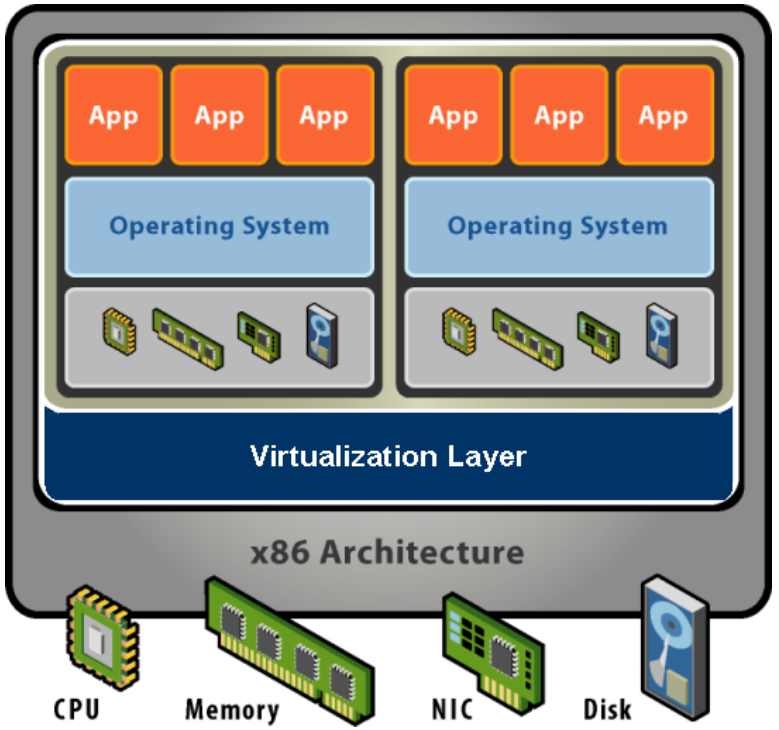
- Tightly coupled to physical hardware
- Single active OS instance
- OS controls hardware

# What's the Virtualization Layer for?

Each operating system was designed to be in total control of the system, which makes it impossible for two or more operating systems to be executing concurrently on the same platform – unless 'total control' is taken away from them by a new layer of control-software: the VMM



# What is a Virtual Machine?

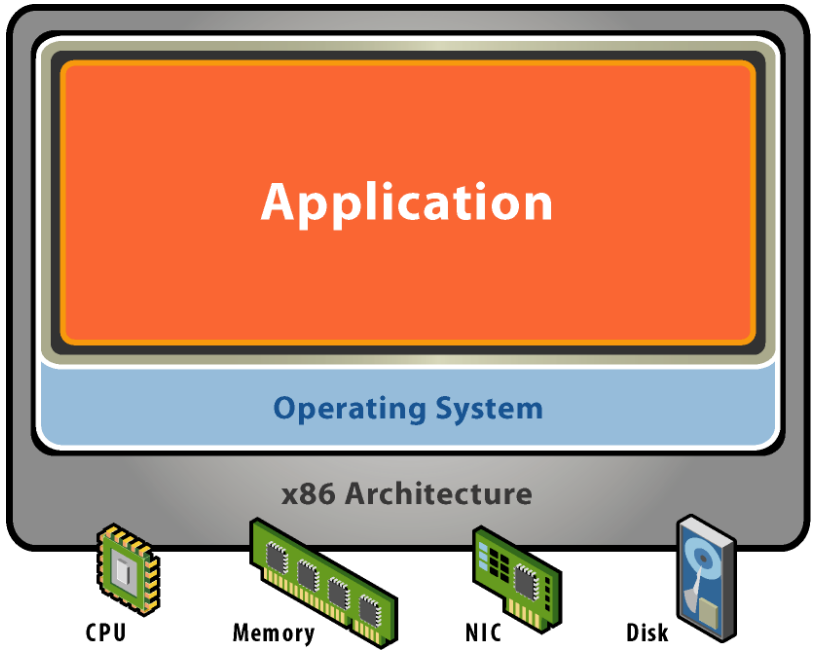



## Software Abstraction

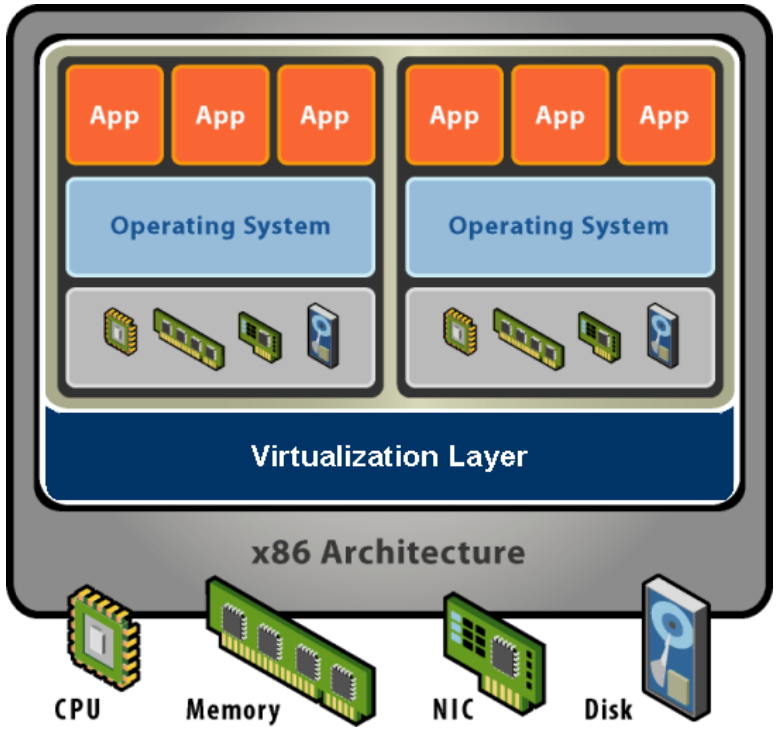
- Behaves like hardware
- Encapsulates all OS and application state

## Virtualization Layer

- Extra level of indirection
- Decouples hardware, OS
- Enforces isolation
- Multiplexes physical hardware across VMs

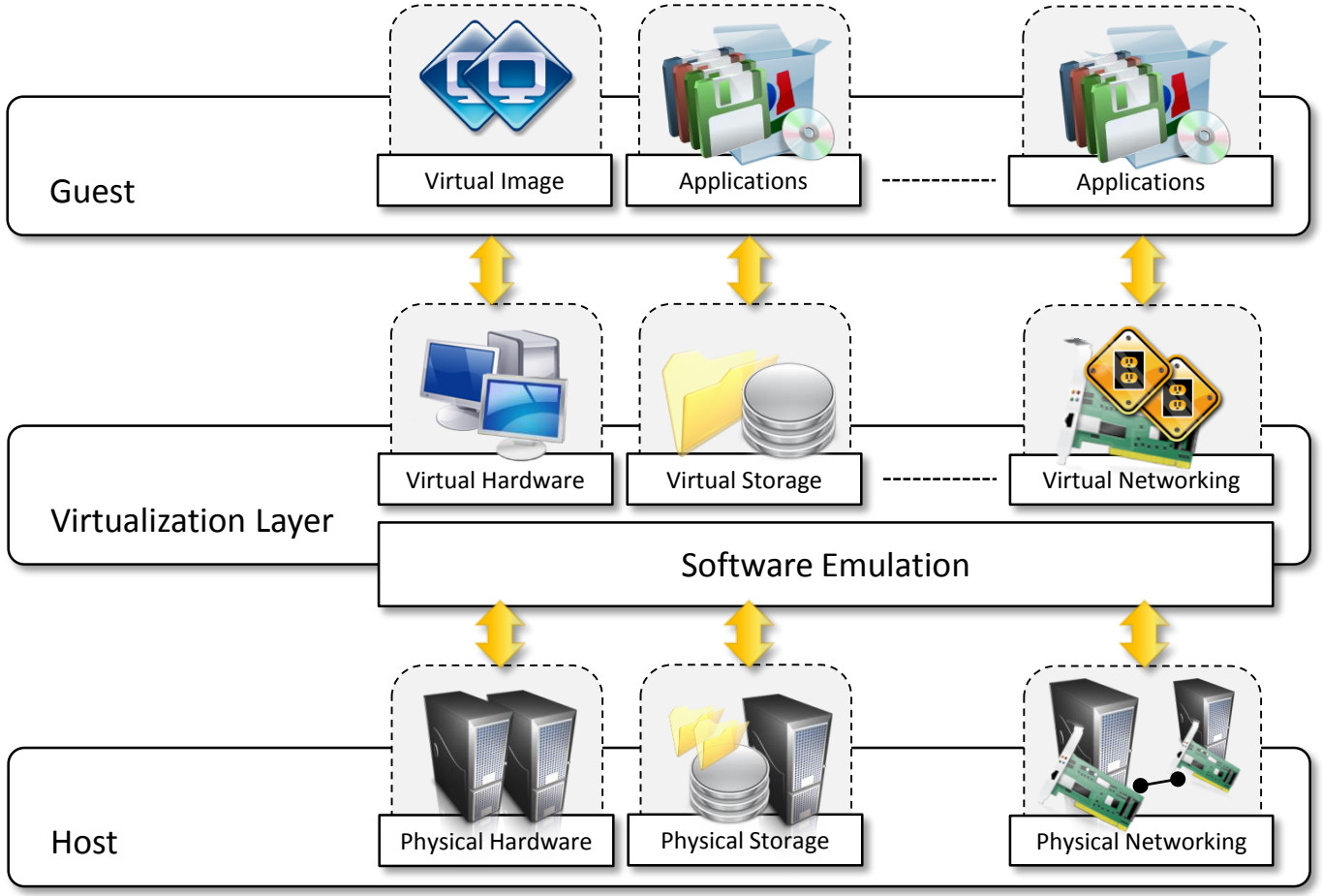



- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure




- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

# Virtualization Layers



# Virtualization Properties

- Isolation
  - Fault isolation
  - Performance isolation
- Encapsulation
  - Cleanly capture all VM state
  - Enables VM snapshots, clones
- Portability
  - Independent of physical hardware
  - Enables migration of live, running VMs
- Interposition
  - Transformations on instructions, memory, I/O
  - Enables transparent resource overcommitment, encryption, compression, replication …

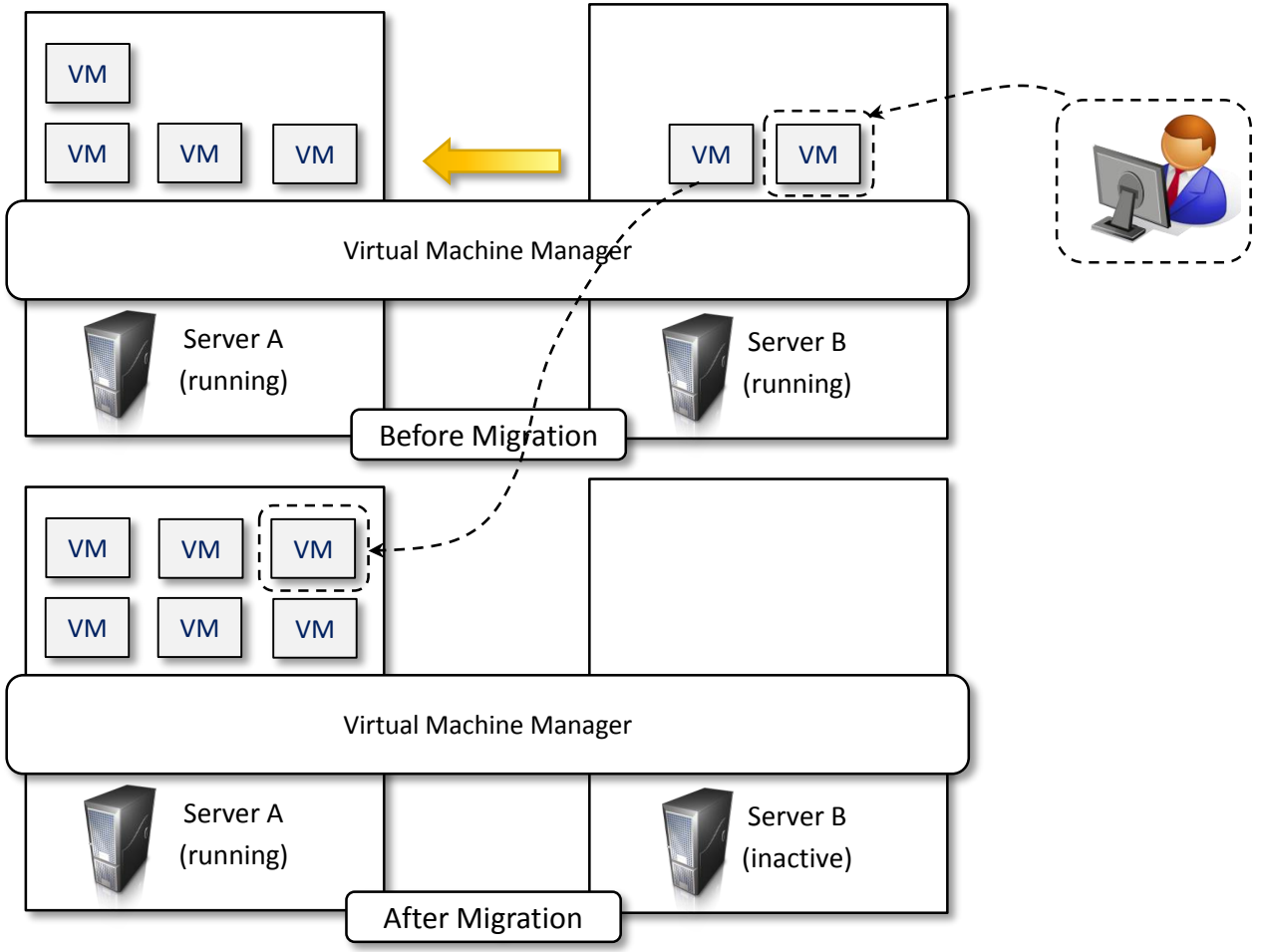
# Modern Virtualization Renaissance

- Recent Proliferation of VMs
  - Considered exotic mainframe technology in 90s
  - Now pervasive in datacenters and clouds
  - Huge commercial success
- Why?
  - Introduction on commodity x86 hardware
  - Ability to “do more with less” saves $$$
  - Innovative new capabilities
  - Extremely versatile technology

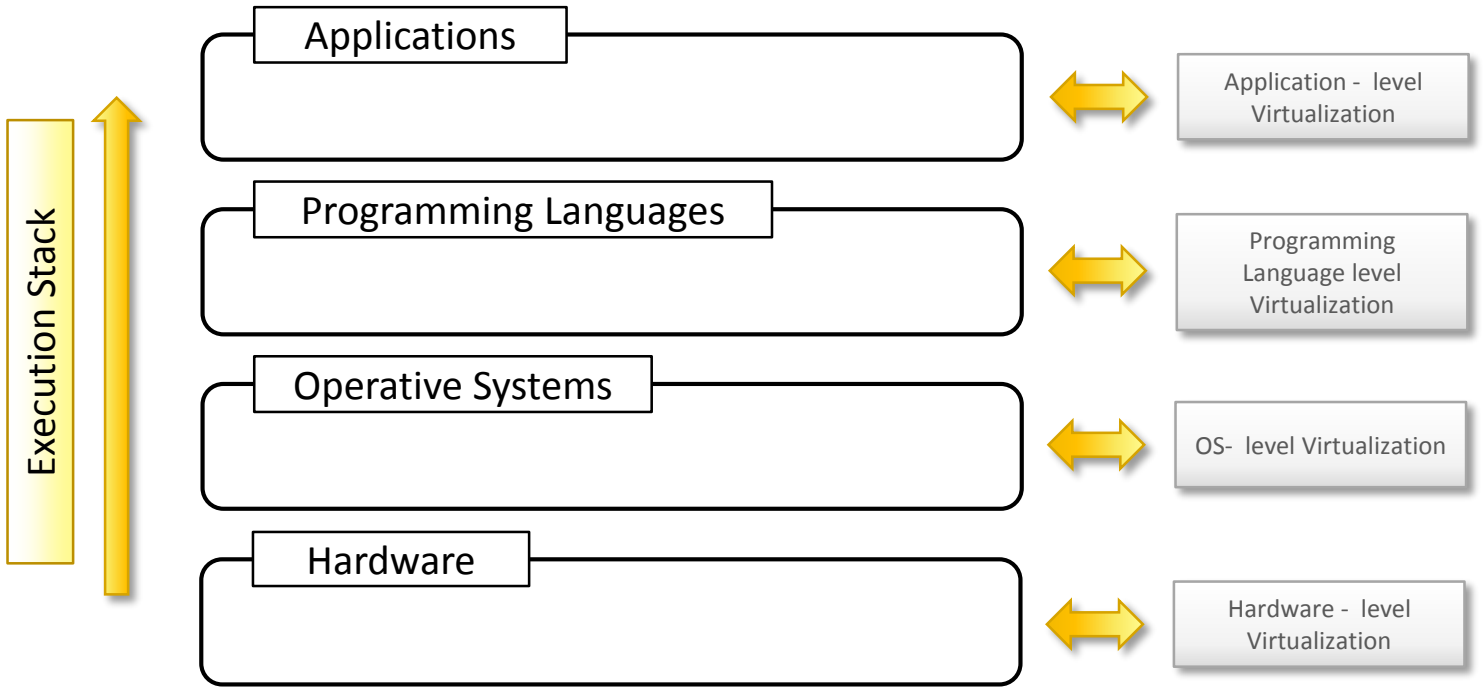
# Modern Virtualization Applications

- Server Consolidation
  - Convert underutilized servers to VMs
  - Significant cost savings (equipment, space, power)
  - Increasingly used for virtual desktops
- Improved Availability
  - Automatic restart
  - Fault tolerance
  - Disaster recovery
- Test and Development
- Simplified Management
  - Datacenter provisioning and monitoring
  - Dynamic load balancing

# Better Resource Utilization



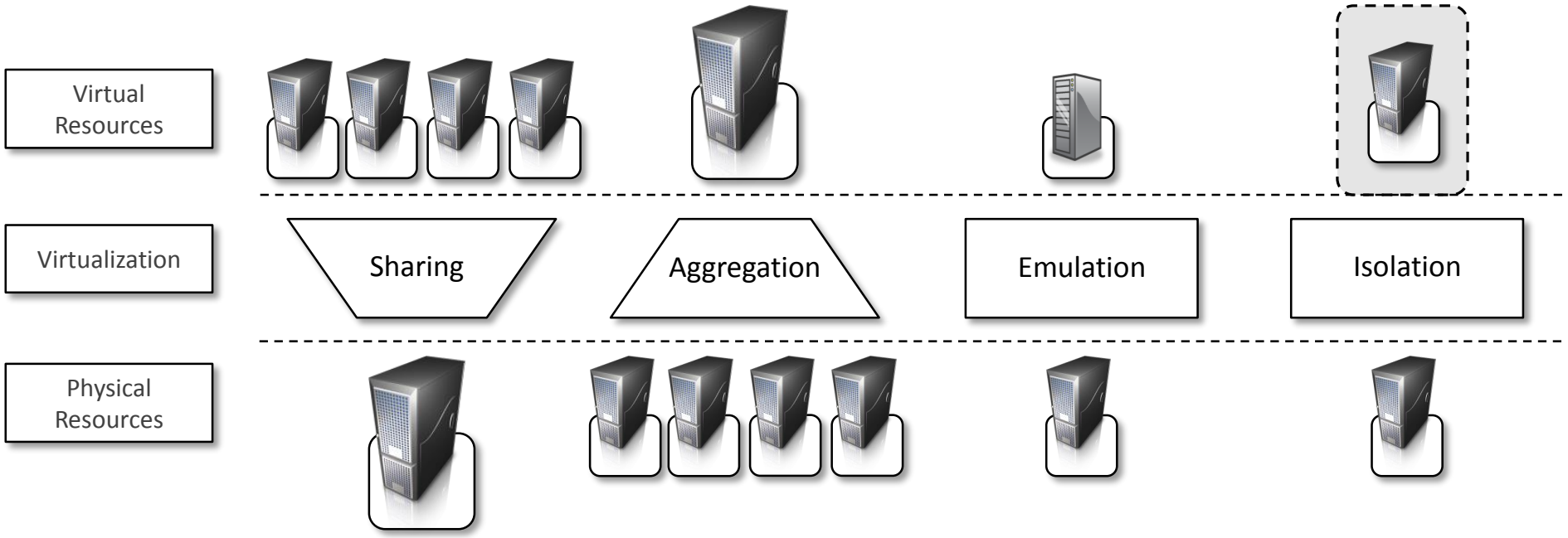
# Virtualization Execution Stack



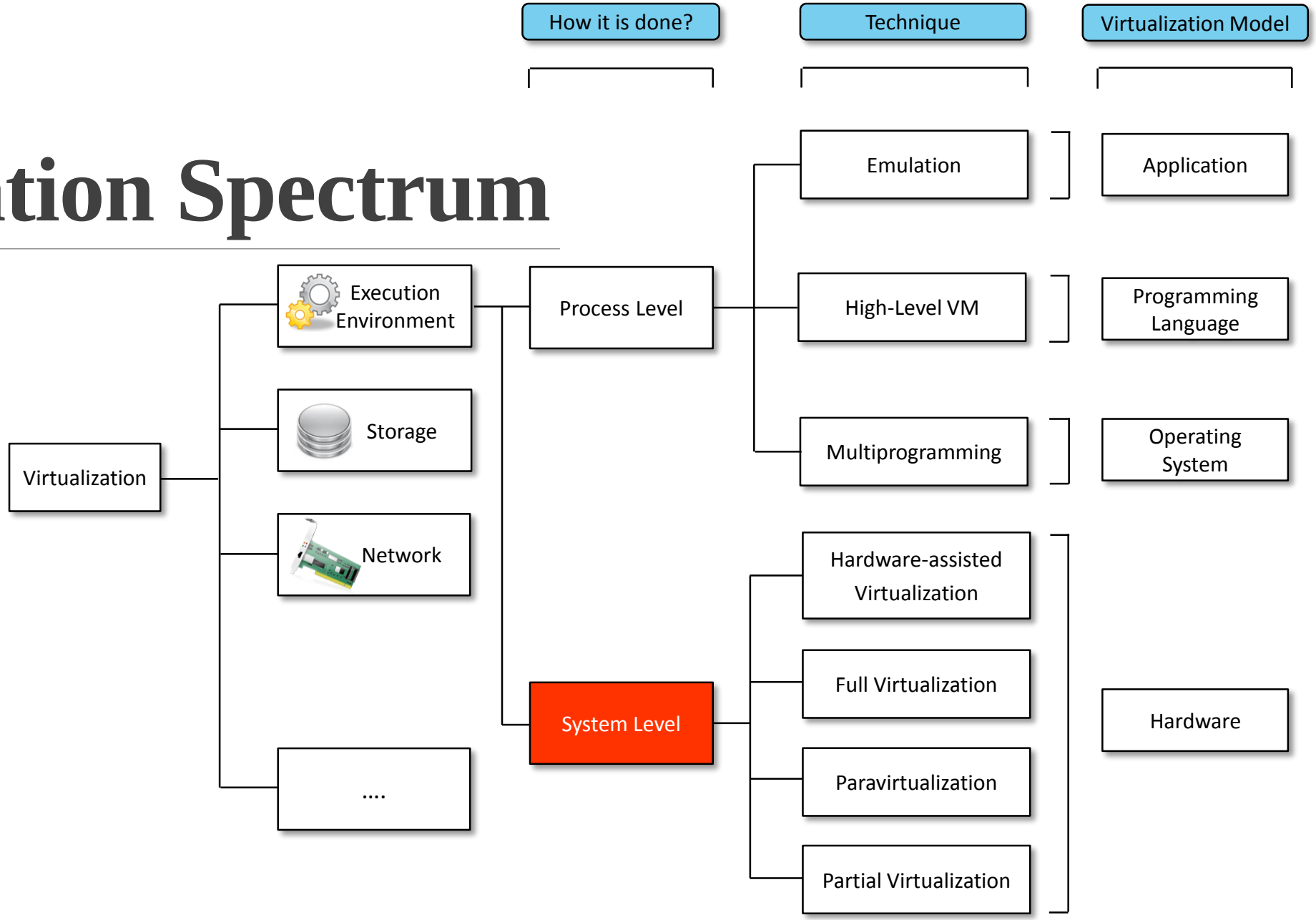
# Types of Virtualization

- Process Virtualization
  - Language-level Java, .NET, Smalltalk
  - OS-level processes, Solaris Zones, BSD Jails, Virtuozzo
  - Cross-ISA emulation Apple 68K-PPC-x86, Digital FX!32
- Device Virtualization
  - Logical vs. physical VLAN, VPN, NPIV, LUN, RAID
- **System Virtualization**
  - “Hosted” VMware Workstation, Microsoft VPC, Parallels
  - “Bare metal” VMware ESX, Xen, Microsoft Hyper-V

# Virtualization Granularity

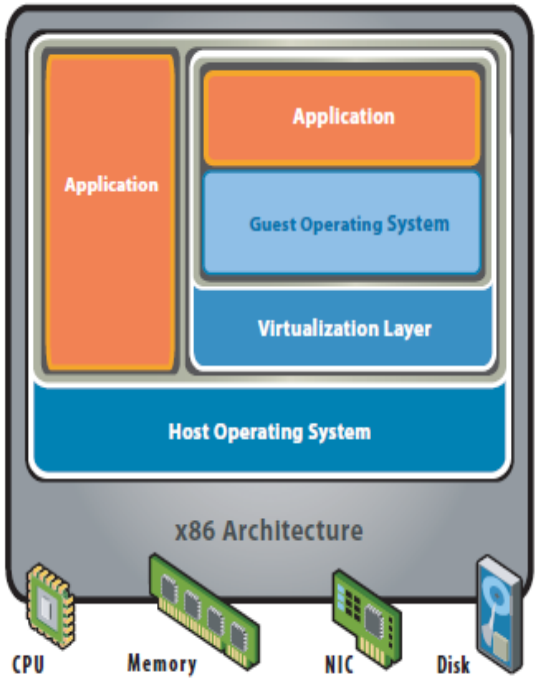


# Virtualization Spectrum



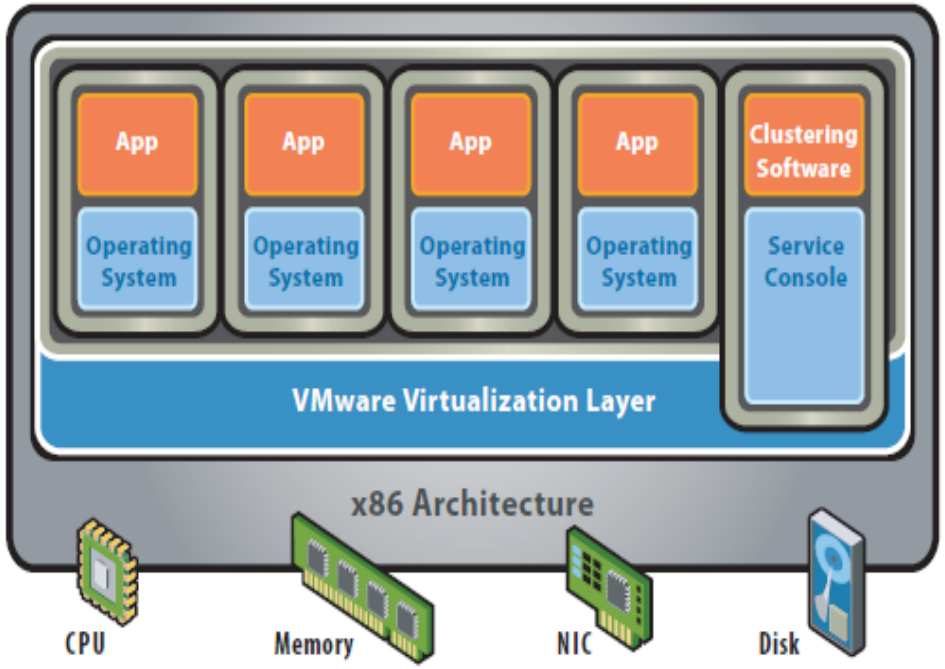
# Virtualization Architectures

## Hosted Architecture



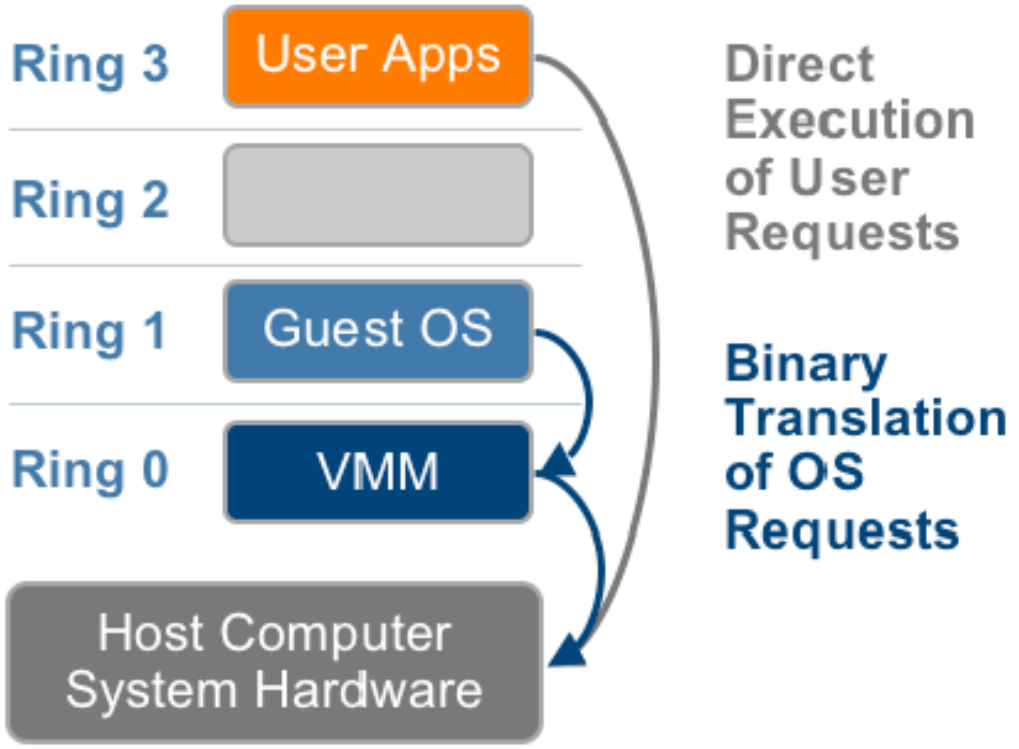

- Installs and runs as an application
- Relies on host OS for device support and physical resource management

## Bare-Metal (Hypervisor) Architecture

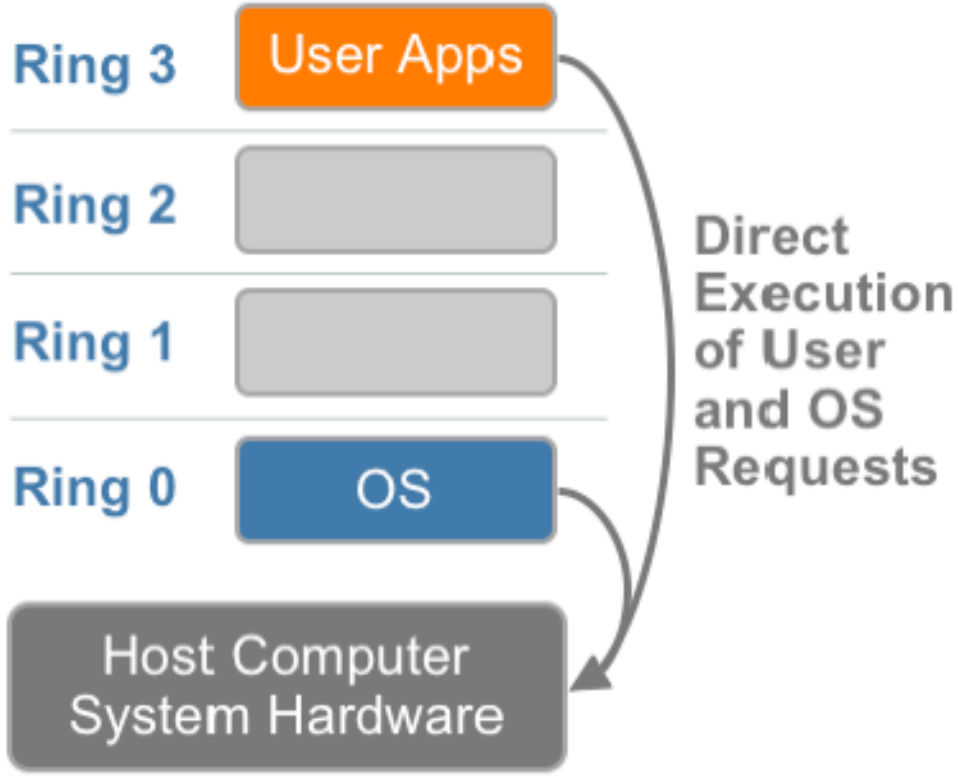



- Lean virtualization-centric kernel
- Service Console for agents and helper applications

# Execution Privilege levels

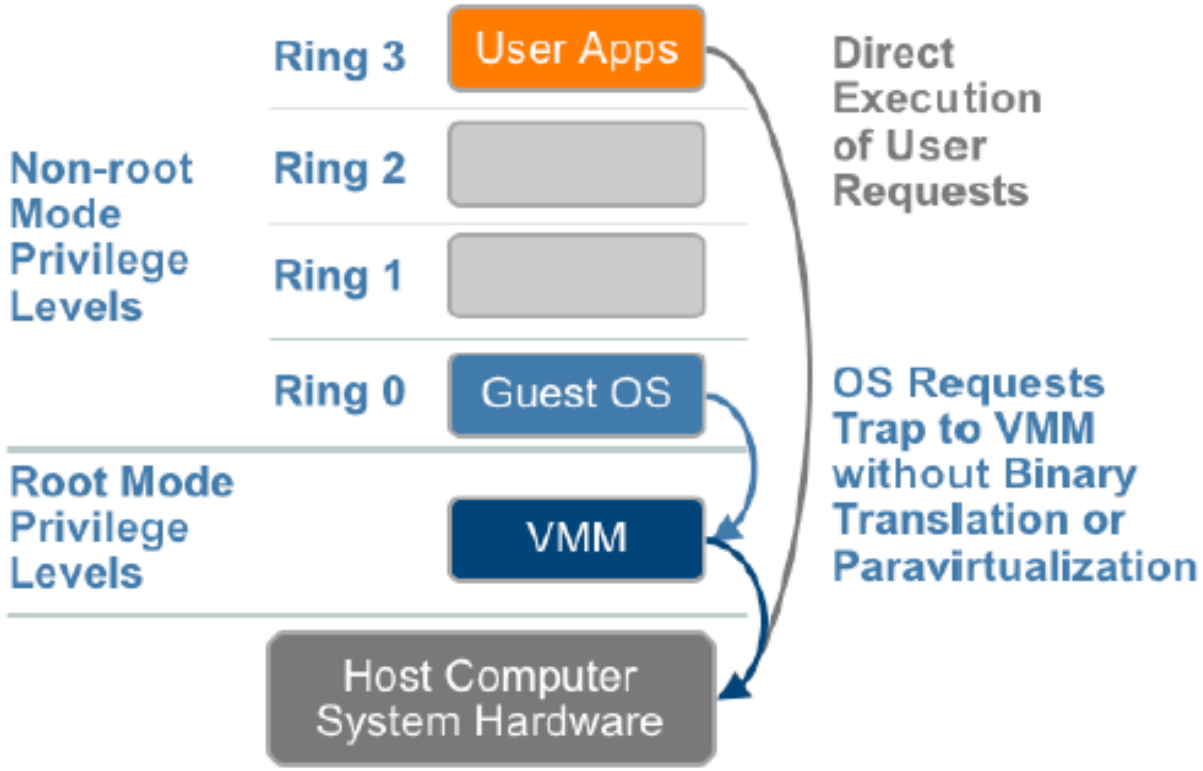



Full Virtualization using Binary Translation

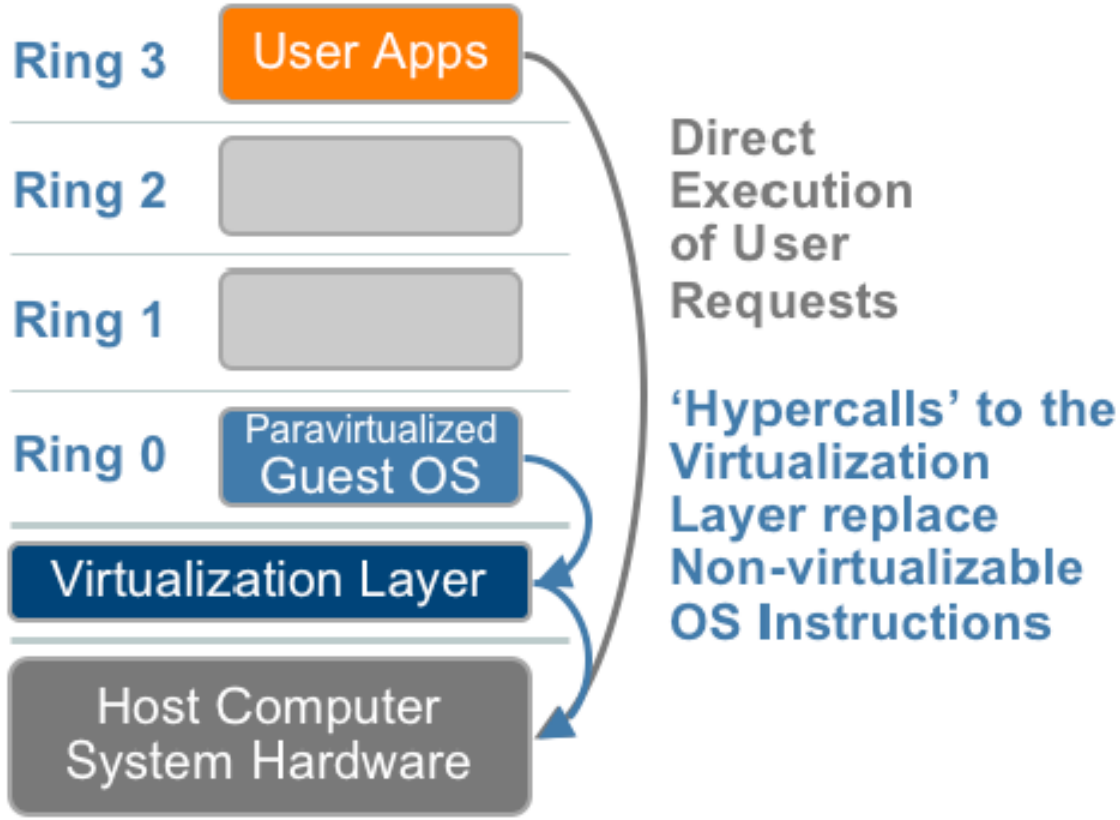



Privilege level architecture without virtualization

# Execution Privilege levels (cont.)




Hardware-assisted Virtualization




Paravirtualization Virtualization (OS-assisted)

# What type of virtualization does Amazon EC2 use?

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html

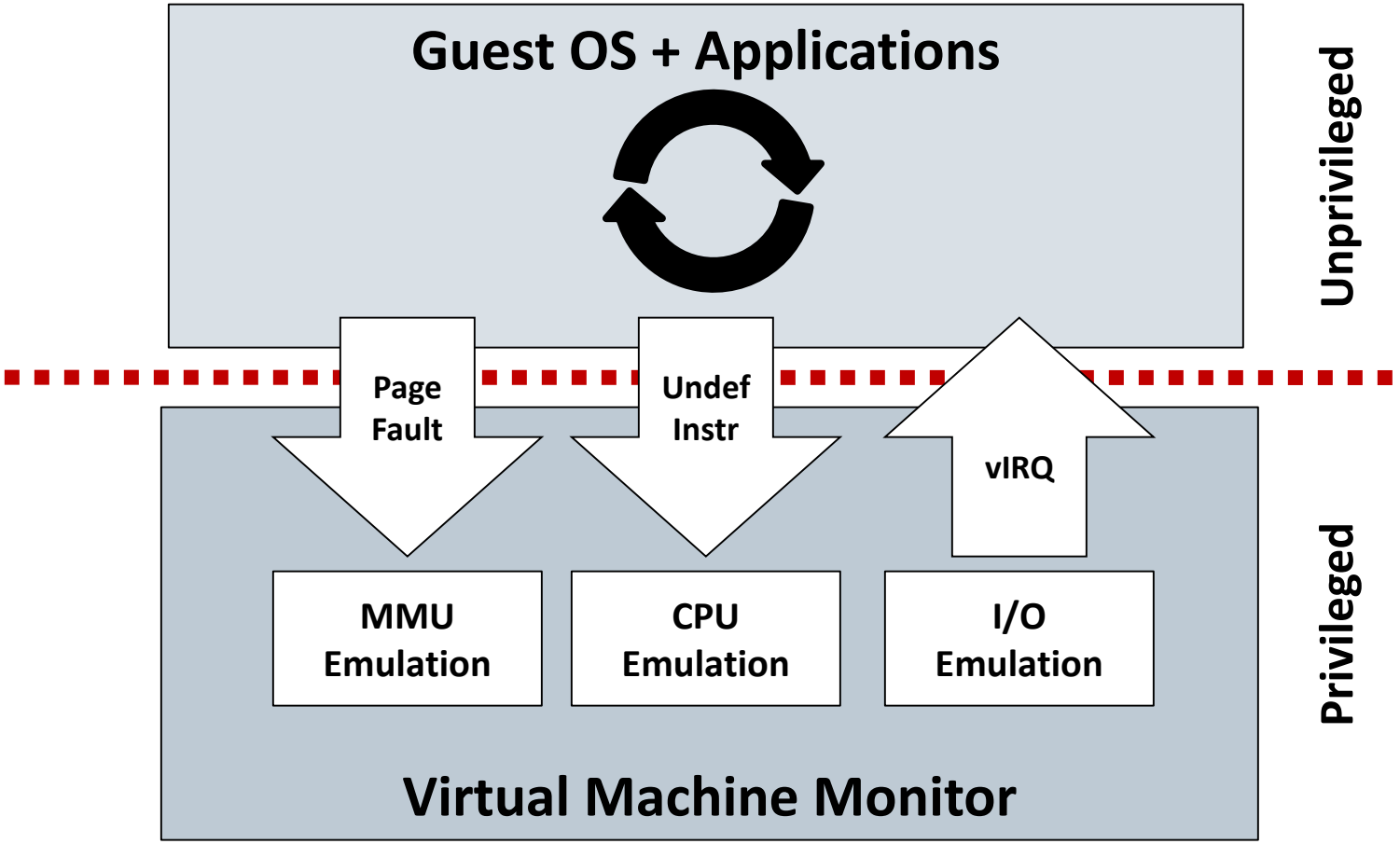
# Components of System-Level Virtualization

- Processor virtualization
- Memory virtualization
- I/O virtualization

# Processor Virtualization

- Trap and Emulate
- Binary Translation

# Trap and Emulate

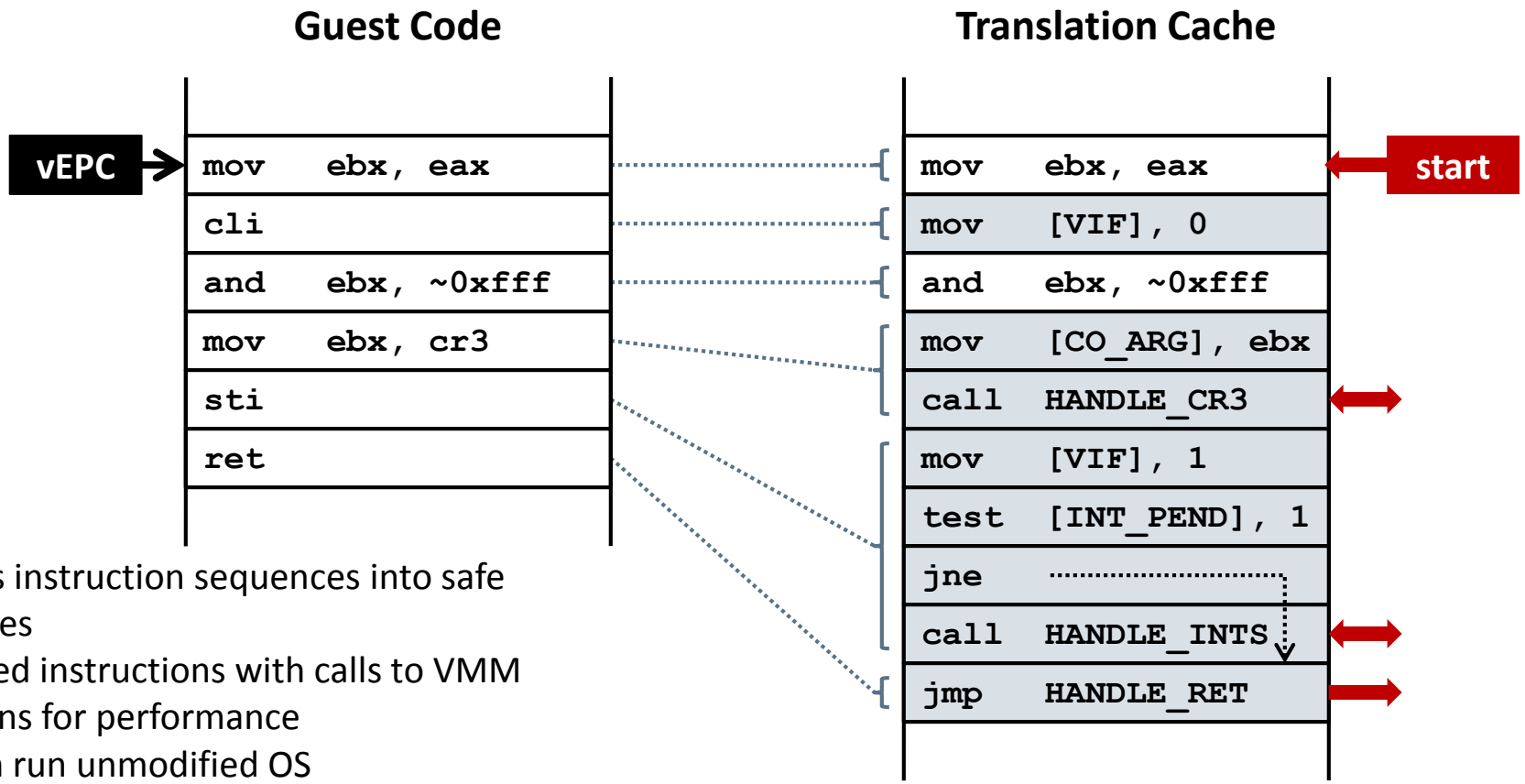


# Control Register CR0

The CPU can readily access register CR0 when it is executing in ‘real-mode’ (ring0), but it lacks the necessary privileges for executing ‘mov %cr0, %eax’ when it is running in Virtual-8086 Mode (ring3)

The attempt by ring3 code to access any of the Control Registers will be ‘trapped’ (raise an exception) by the CPU (a ‘General Protection’ Fault)

# Binary Translation




Translate dangerous instruction sequences into safe instruction sequences

- Replace privileged instructions with calls to VMM
- Cache translations for performance
- Advantages: Can run unmodified OS
- Disadvantages: Frequent traps to VMM

# Memory Virtualization

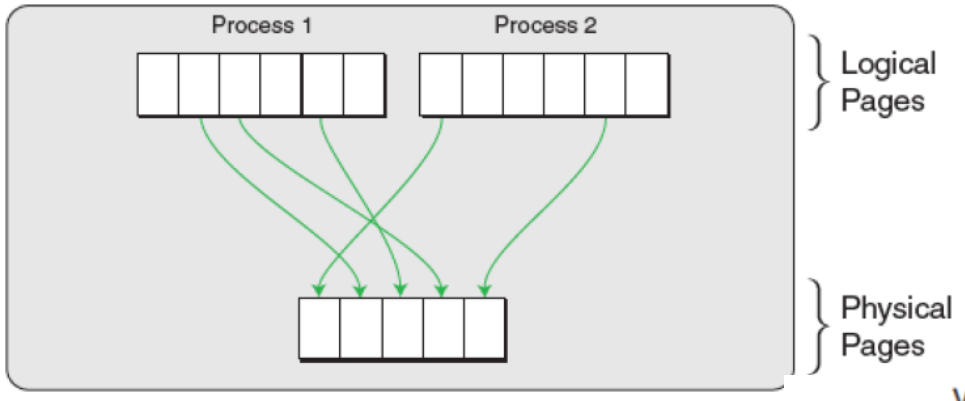
- Shadow Page Tables
- Nested Page Tables

# Memory Virtualization

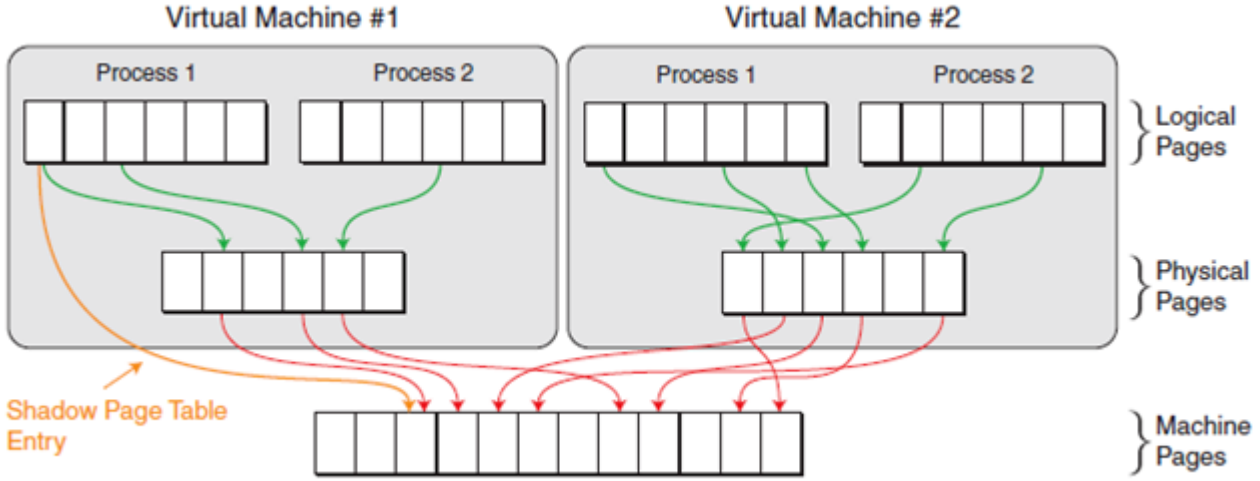
- The operating system keeps mappings of virtual page numbers to physical page numbers stored in page tables. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.
- The guest OS continues to control the mapping of virtual addresses to the guest memory physical addresses, but the guest OS cannot have direct access to the actual machine memory.
- The VMM is responsible for mapping guest physical memory to the actual machine memory.

# Memory Virtualization

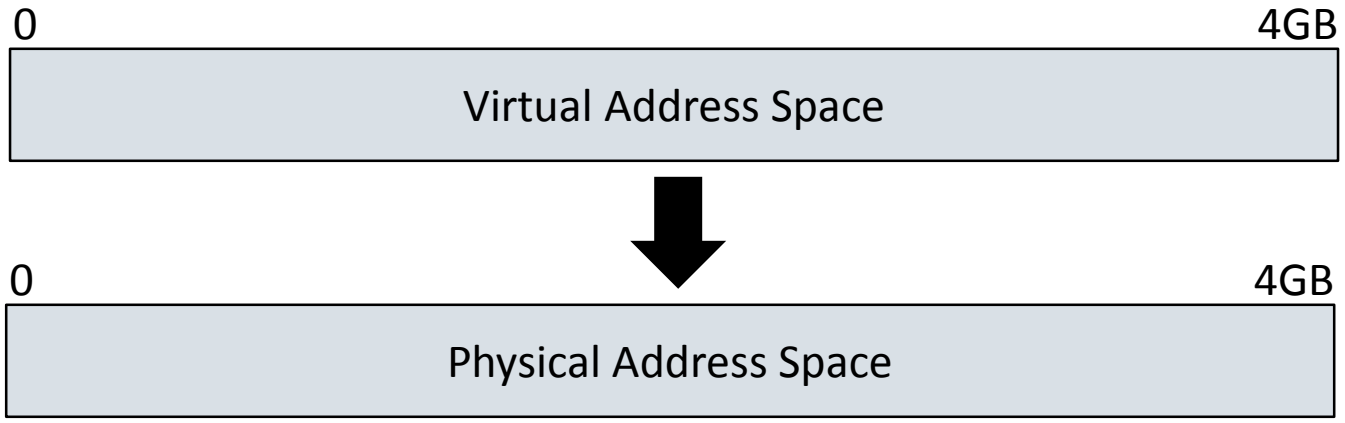
- Native



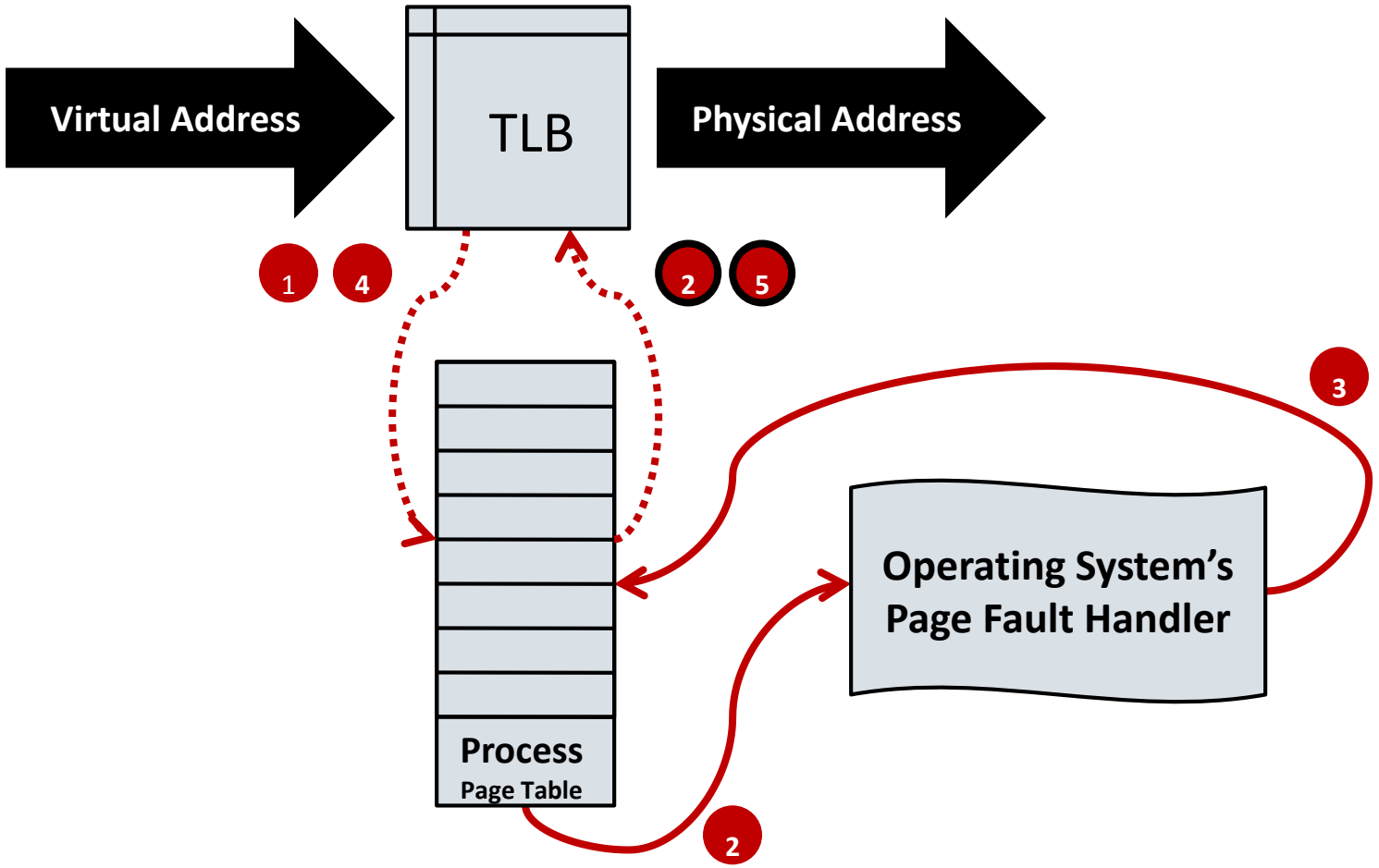

- Virtualized



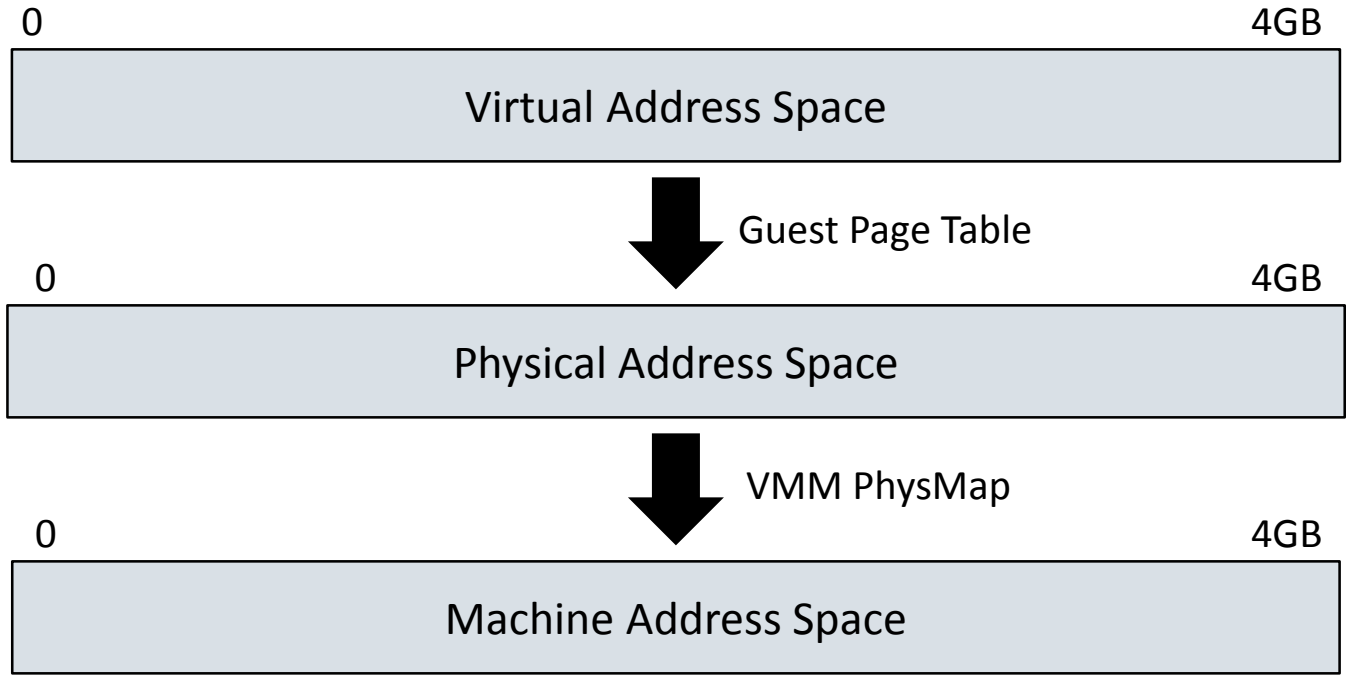
# Traditional Address Spaces



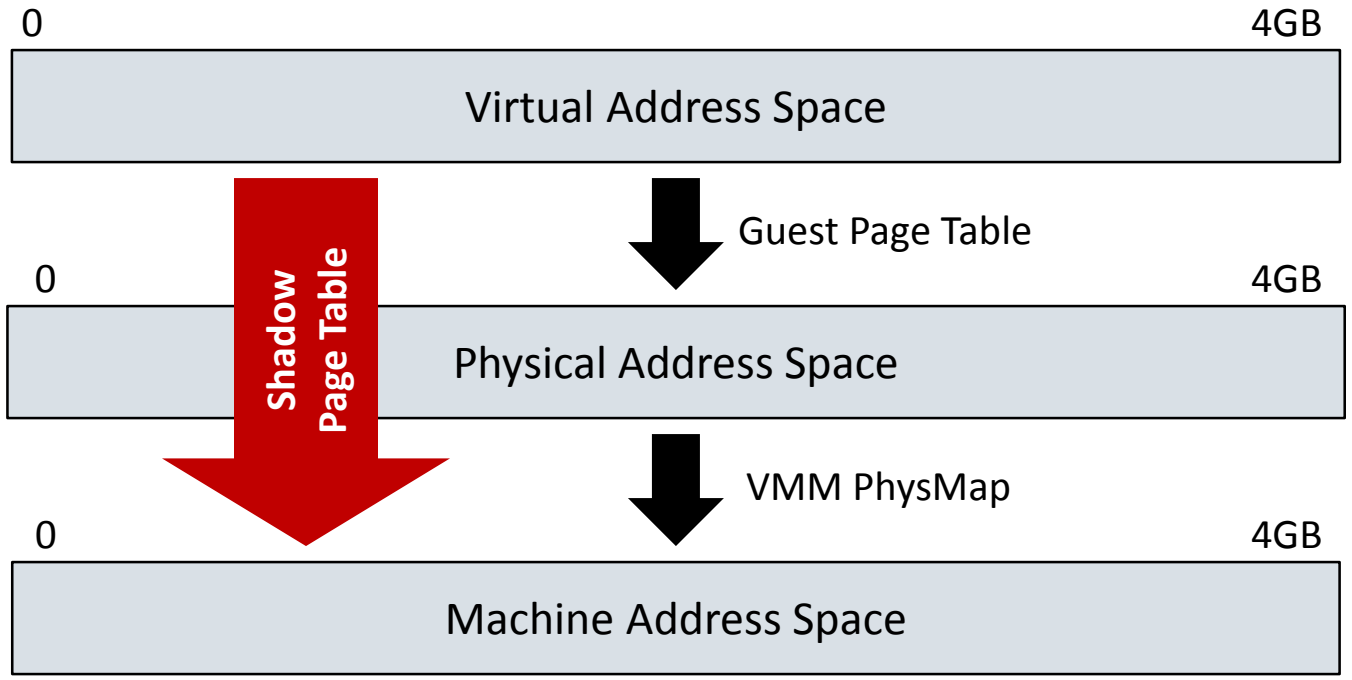
# Traditional Address Translation



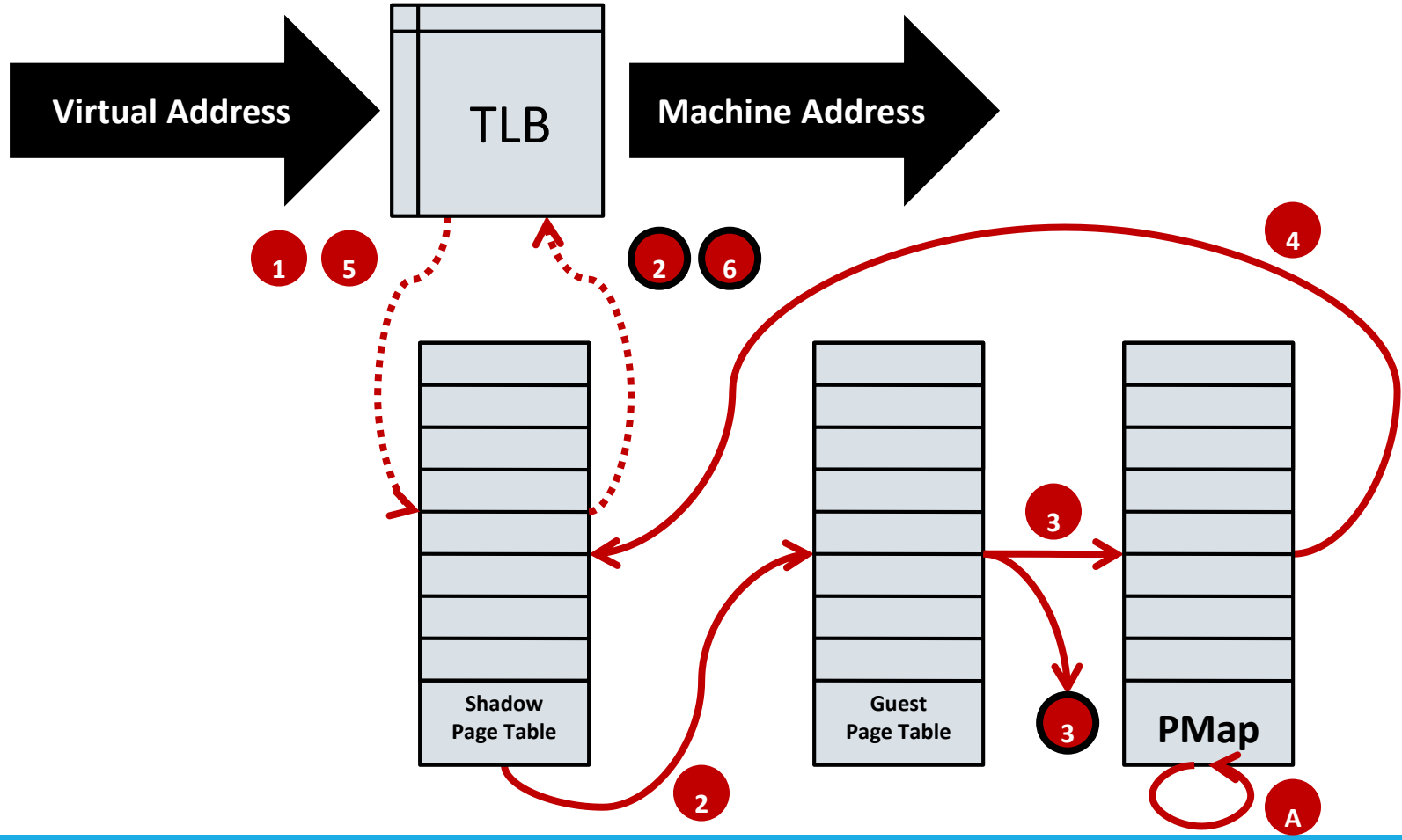
# Virtualized Address Spaces



# Virtualized Address Spaces w/ Shadow Page Tables



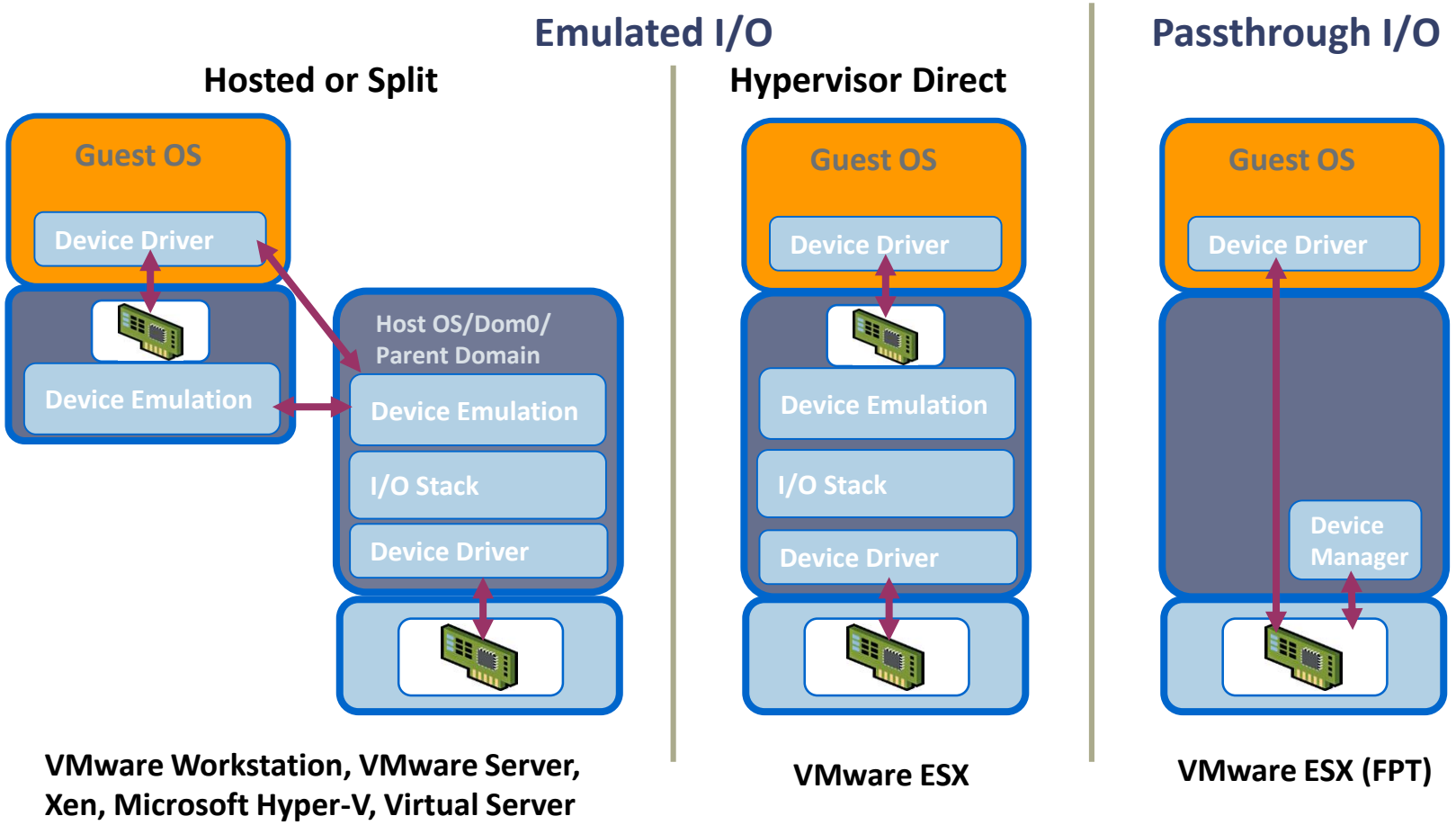
# Virtualized Address Translation w/ Shadow Page Tables



# Device and I/O Virtualization

- Software based I/O virtualization and management, in contrast to a direct pass-through to the hardware, enables simplified management.
- The key to effective I/O virtualization is to preserve these virtualization benefits while keeping the added CPU utilization to a minimum.
- The hypervisor virtualizes the physical hardware and presents each virtual machine with a standardized set of virtual devices. Virtual devices effectively emulate well-known hardware and translate the virtual machine requests to the system hardware.

# I/O Virtualization Implementations



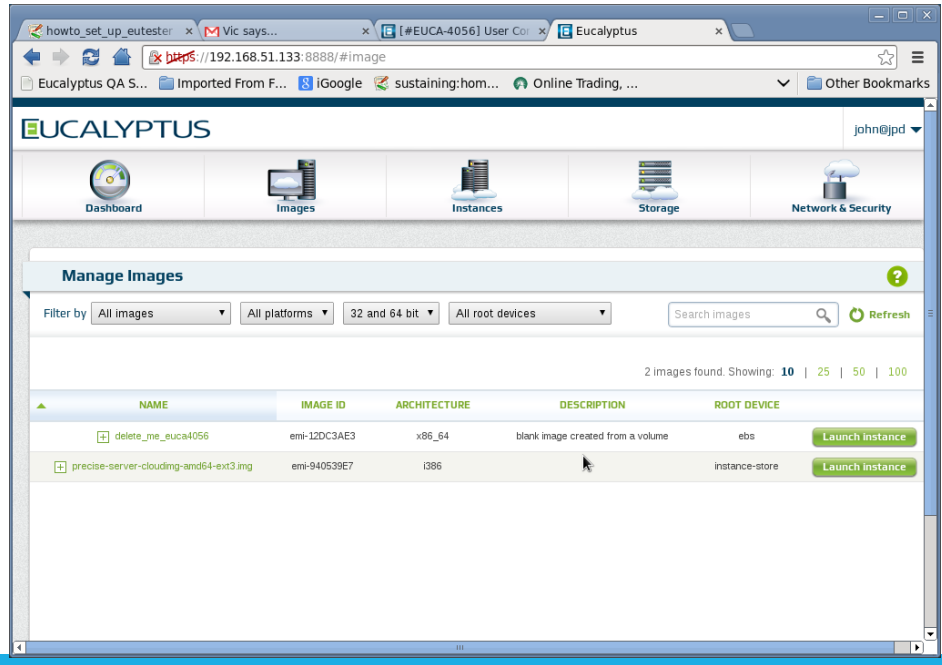
# 2. Cloud Infrastructure

# IaaS Stacks

- CloudStack, a 100 percent Java open source IaaS, backed by Citrix
- Eucalyptus, the academic IaaS
- OpenStack, the best-known open source IaaS, backed by Rackspace
- Amazon VPC, a closed-source option (but AWS is the "invisible hand" that guides the others)

# Eucalyptus

- EUCALYPTUS – an open-source software  framework for cloud computing
- Implements Infrastructure as a Service (IaaS)
- Emulates Amazon cloud – EC2, S3 and Elastic Block Storage (EBS)
- Intended as a tool for cloud computing researchers



# Eucalyptus (Cont.)

Original work  addressed several cloud computing questions:

- VM instance scheduling,
- VM and user data storage,
- Cloud computing administrative interfaces,
- Construction of virtual networks,
- Definition and execution of service level agreements (cloud/user and cloud/cloud)
- Cloud computing user interfaces.

# Eucalyptus Main Components

**Node Controller**

- controls the execution, inspection, and terminating of VM instances on the host where it runs.

**Cluster Controller**

- gathers information about and schedules VM execution on specific node controllers
- manages virtual instance network.

**Storage Controller (Walrus)**

- put/get storage service that implements Amazon's S3 interface,
- provides a mechanism for storing and accessing virtual machine images and user data.

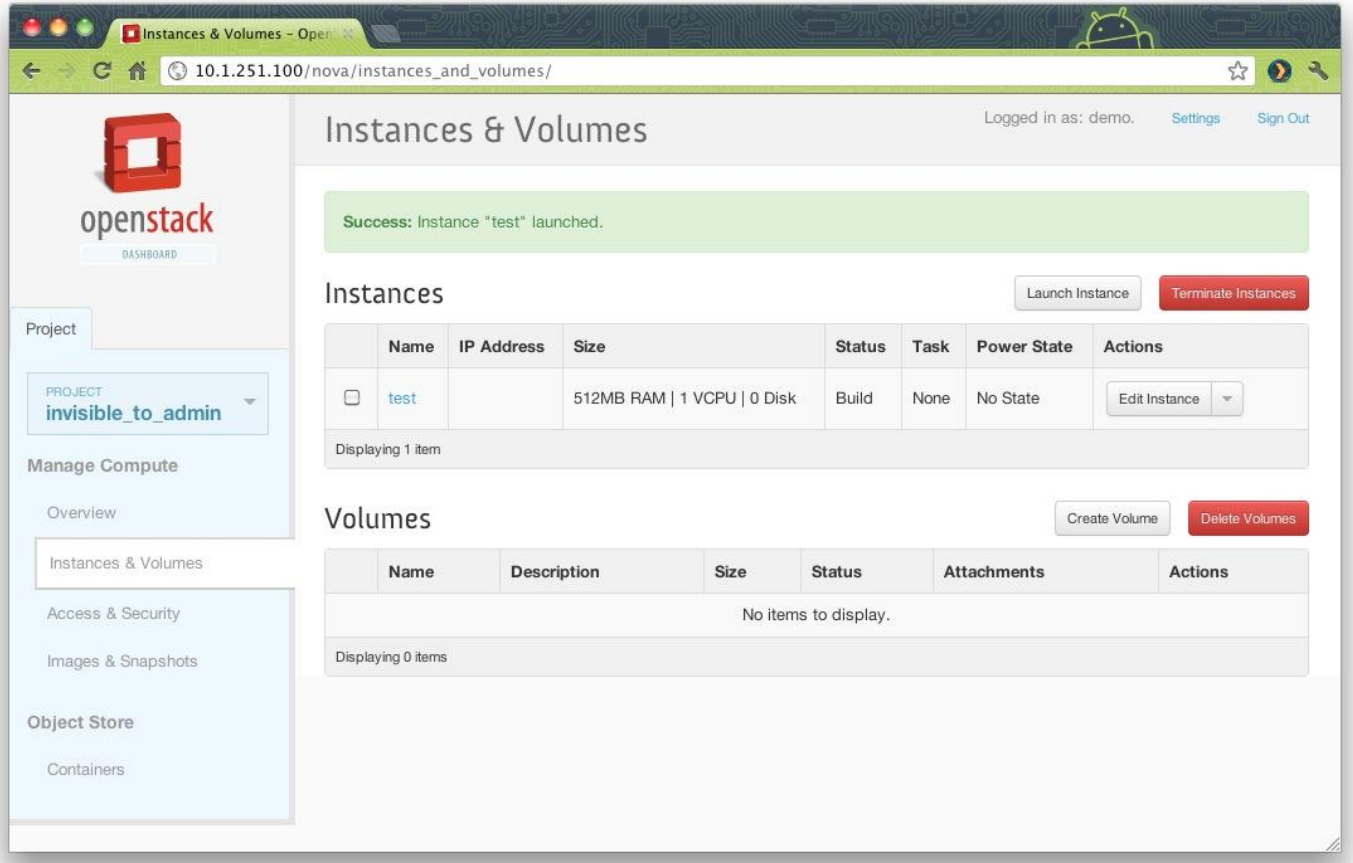
**Cloud Controller**

- the entry-point into the cloud for users and administrators
- queries node managers for information about resources
- makes high-level scheduling decisions, and implements them by making requests to cluster controllers.

# Goals for Eucalyptus

- Foster research in elastic/cloud/utility computing
  - models of service provisioning, scheduling, SLA formulation, hypervisor portability and feature enhancement, etc.
- Experimentation vehicle prior to buying commercial services
  - “Tech Preview” using local machines with local system administration support
- Provide a debugging and development platform for EC2 (and other clouds)
  - Allow the environment to be set up and tested before it is instantiated in a for-fee environment
- Provide a basic software development platform for the open source community
  - E.g. the “Linux Experience”
- Not a designed as a replacement technology for EC2 or any other cloud service

# OpenStack


Instances & Volumes - Open
10.1.251.100/nova/instances_and_volumes/
openstack
DASHBOARD
Project
PROJECT
invisible_to_admin
Manage Compute
Overview
Instances & Volumes
Access & Security
Images & Snapshots
Object Store
Containers
Instances & Volumes
Logged in as: demo.
Settings
Sign Out
Success: Instance "test" launched.
Instances
Launch Instance
Terminate Instances
Name
IP Address
Size
Status
Task
Power State
Actions
test
512MB RAM | 1 VCPU | 0 Disk
Build
None
No State
Edit Instance
Displaying 1 item
Volumes
Create Volume
Delete Volumes
Name
Description
Size
Status
Attachments
Actions
No items to display.
Displaying 0 items