Curtin College

in association with

Curtin University

Virtualisation

Computer Systems (CS2000)

Trimester 2 2020

What is Virtualisation?

- vir•tu•al (adj):
  - existing in essence or effect, though not in actual fact
- Virtual systems
  - Abstract physical components using logical objects
  - Dynamically bind logical objects to physical configurations
  - Examples
    - Network – Virtual LAN (VLAN), Virtual Private Network (VPN)
    - Storage – Storage Area Network (SAN), LUN
    - Computer – Virtual Machine (VM), simulator

Outline

- What is virtualization?
- Virtualization classification
- Monitor Architectures

Problem

- Enterprise IT centers support many service applications
  - Microsoft Exchange
  - Oracle
  - SAP
  - Web servers
  - Citrix
  - …
- Each service application demands its own environment
  - Specific version of operating system
  - Multiple processors and disks
  - Specialized configurations
  - …

Problem (cont.)

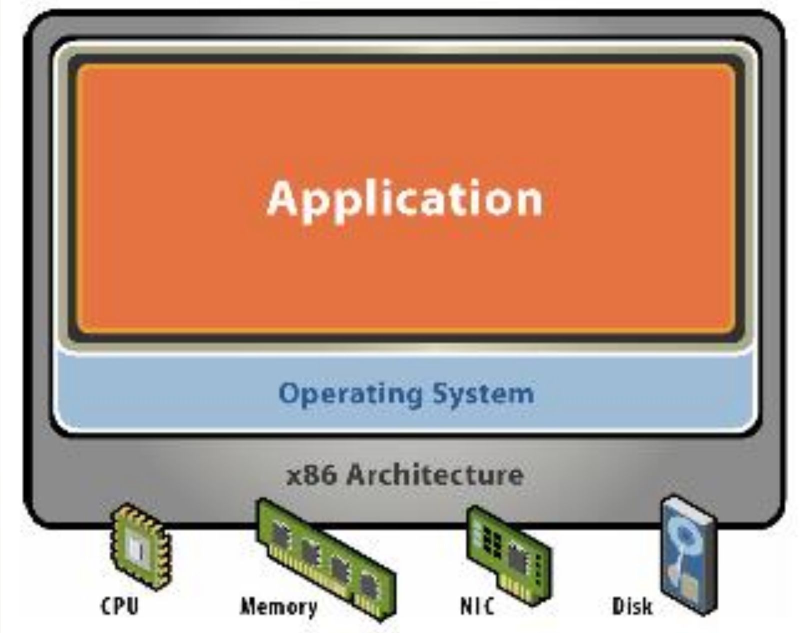
- Combining services on same server host is difficult (at best)
  - Conflicting demands
  - Incompatible loads
- Upgrading or commissioning a service is very difficult
  - Shadow server machines for debugging & testing
  - Complicated changeover tactics

Problem (cont.)

- Adding or upgrading hardware or OS is difficult
  - Testing and refitting active service
  - Complicated changeover tactics
- Load balancing is impossible
  - Services tied to own systems
  - Some underused, some overused

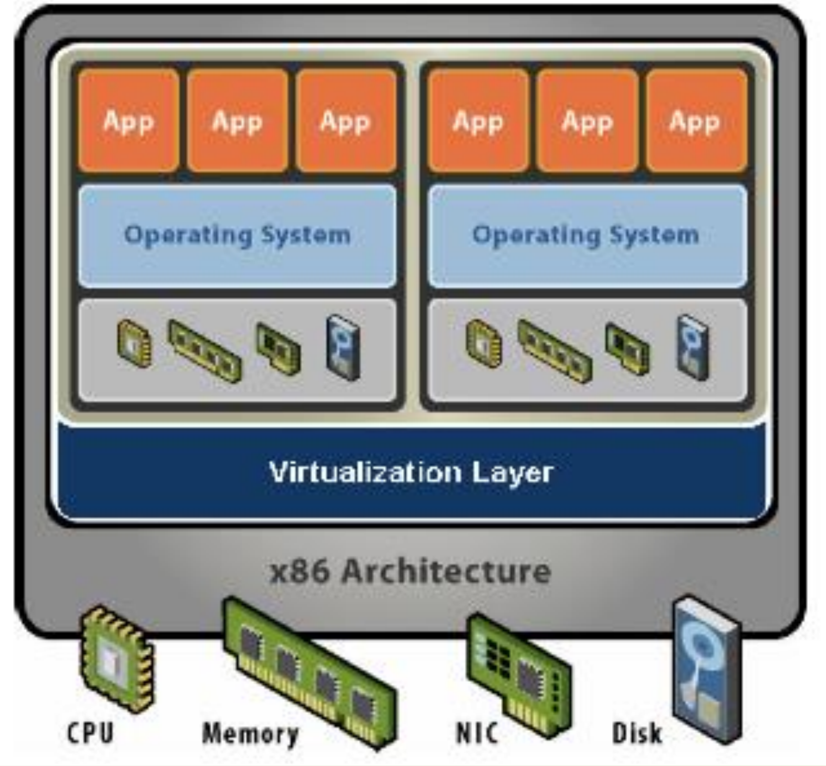
Starting Point: A Physical Machine

- Physical Hardware
  - Processors, memory, chipset, I/O bus and devices, etc.
  - Physical resources often underutilized
- Software
  - Tightly coupled to hardware
  - Single active OS image
  - OS controls hardware



What is a Virtual Machine?

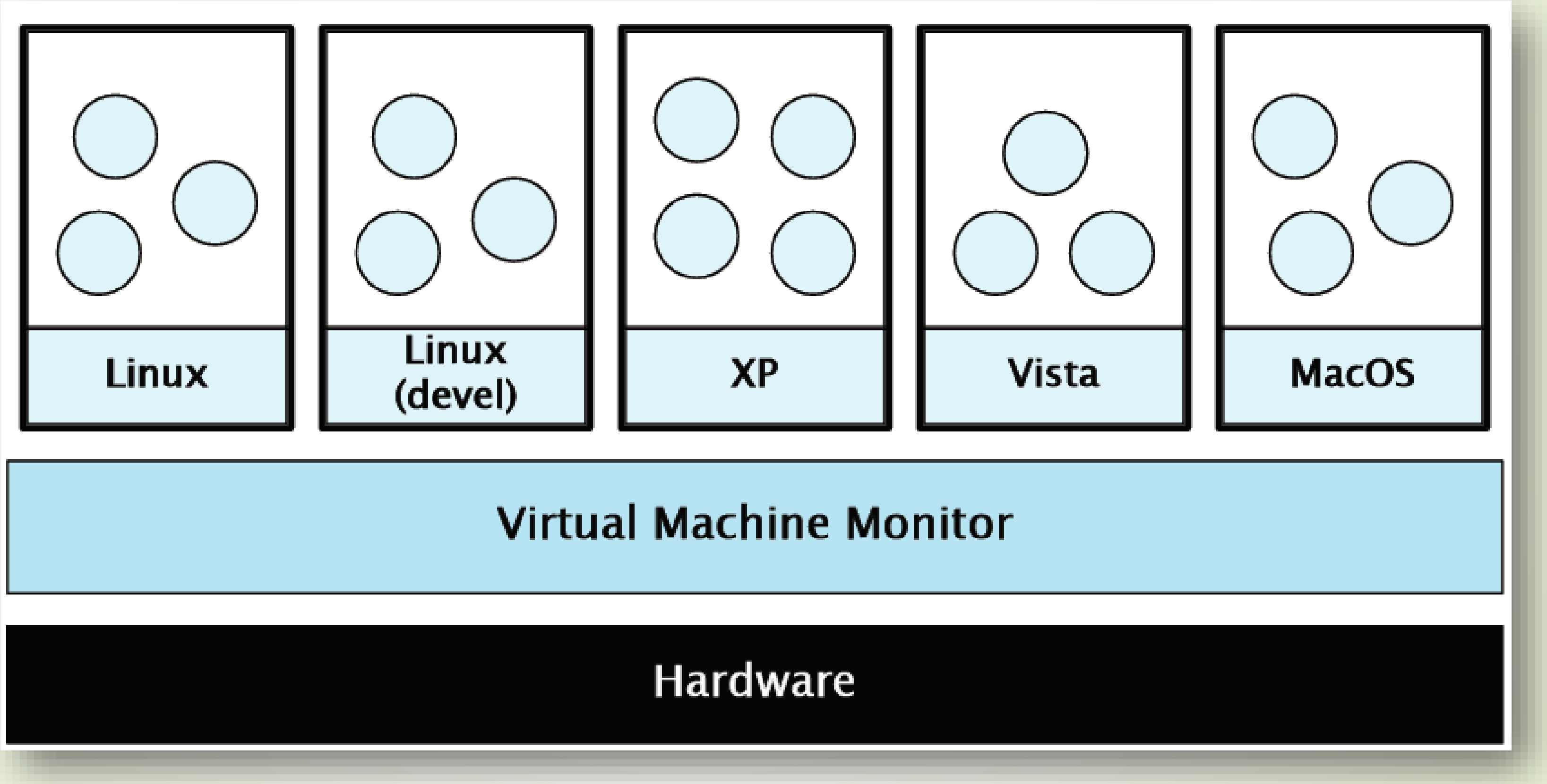
- Hardware-Level Abstraction
  - Virtual hardware: processors, memory, chipset, I/O devices, etc.
  - Encapsulates all OS and application state
- Virtualization Software
  - Extra level of indirection decouples hardware and OS
  - Multiplexes physical hardware across multiple "guest" VMs
  - Strong isolation between VMs
  - Manages physical resources, improves utilization



What is Virtualisation?

- Loosely, virtualization is the addition of a software layer (the VMM) between the hardware and the existing software that exports an interface at the same level as the underlying hardware.
- In the strictest case the exported interface is the exact same as the underlying hardware and the VMM provides no functionality except multiplexing the hardware among multiple VMs.
  - This was largely the case in the old IBM VM/360 systems.
- However the layer can export a different hardware interface
  - cross-ISA emulators.
- Virtualization is the addition of a layer of software that can run the original software with little or no changes.

What is Virtualisation?



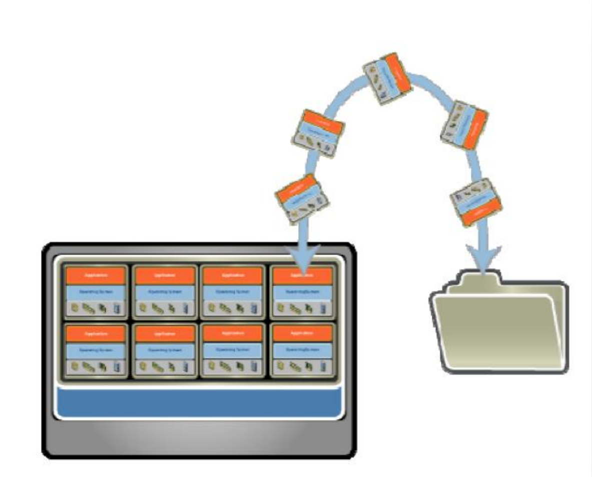
Virtualisation Properties

- Isolation
- Encapsulation
- Interposition

Isolation

- Secure Multiplexing
  - Run multiple VMs on single physical host
  - Processor hardware isolates VMs
- Strong Guarantees
  - Software bugs, crashes, viruses within one VM cannot affect other VMs
- Performance Isolation
  - Partition system resources
  - Example: VMware controls for reservation, limit, shares
  - Accomplished through scheduling and resource allocation
- Fault Isolation
  - Fundamental property of virtualization
- Software Isolation
  - Software versioning

Encapsulation



- All VM state can be captured into a file
  - Operate on VM by operating on file
  - mv, cp, rm
- Entire VM is a File
  - OS, applications, data
  - Memory and device state
- Snapshots and Clones
  - Capture VM state on the fly and restore to point-in-time
  - Rapid system provisioning, backup, remote mirroring
- Easy Content Distribution
  - Pre-configured apps, demos
  - Virtual appliances

Interposition

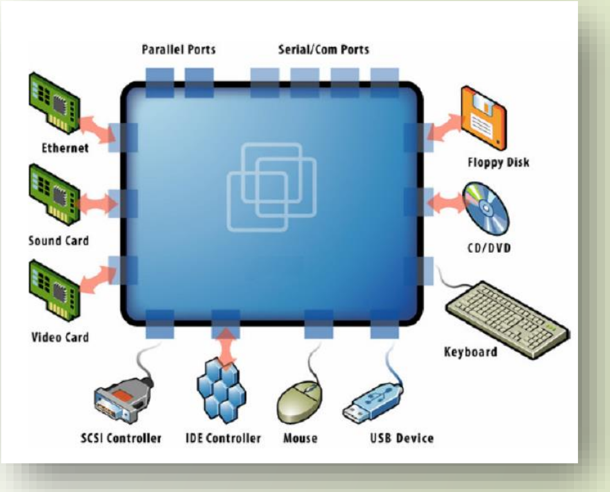
- All guest actions go through monitor
- Monitor can inspect, modify, deny operations
- Examples
  - Compression
  - Encryption
    - The advantage of this is that it does it without the knowledge of the OS.
  - Profiling
  - Translation

Why not the OS?

- It about interfaces
  - VMMs operate at the hardware interface
  - Hardware interface are typically smaller, better defined than software interfaces
- Microkernel for commodity Operating Systems
- Disadvantages of being in the monitor
  - Low visibility into what the guest is doing

VM Compatability

- Hardware-Independent
  - Physical hardware hidden by virtualization layer
  - Standard virtual hardware exposed to VM
- Create Once, Run Anywhere
  - No configuration issues
  - Migrate VMs between hosts
- Legacy VMs
  - Run ancient OS on new platform
    - E.g. DOS VM drives virtual IDE and vLance devices, mapped to modern SAN and GigE hardware



Virtualisation Applications

- Server Consolidation
  - Eliminate server sprawl by deploying systems into virtual machines that can run safely and move transparently across shared hardware
  - Google data centres use 260 million watts — about a quarter of the output of a nuclear power plant.*
- Data Center Management
  - vMotion to migrate running VMs
- High Availability
  - Automatic Restart
- Disaster Recovery
- Fault Tolerance

* http://www.nytimes.com/2011/09/09/technology/google-details-and-defends-its-use-of-electricity.html

Virtualisation Applications

- Test and Development
  - Rapidly provision test and development servers; store libraries of pre-configured test machines
- Business Continuity
  - Reduce cost and complexity by encapsulating entire systems into single files that can be replicated and restored onto any target server
- Enterprise Desktop
  - Secure unmanaged PCs without compromising end user autonomy by layering a security policy in software around desktop virtual machines

Types of Virtualisation

- Process Virtualization (user space)
  - Language construction
    - Java, .NET
  - Cross-ISA emulation
    - Apple’s 68000-PowerPC-Intel Transition
  - Application virtualization
    - Sandboxing, mobility
- Device Virtualization
  - RAID
- System Virtualization
  - VMware
  - Xen
  - Microsoft’s Viridian/Hyper-V

What is a Virtual Machine Monitor?

- VMM Characteristics
  - Fidelity • Performance • Isolation / Safety

  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). 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.*

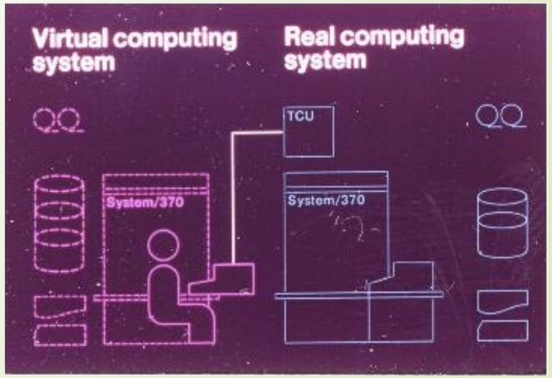
  Gerald J. Popek and Robert P. Goldberg (1974). "Formal Requirements for Virtualizable Third Generation Architectures". Communications of the ACM 17 (7): 412 –421
- An Old Concept
- IBM mainframes since 1960's

VMM Technology

- So this is just like Java, right?
  - No, a Java VM is very different from the physical machine that runs it
  - A hardware-level VM reflects underlying processor architecture
- Like a simulator or emulator that can run old Nintendo games?
  - No, they emulate the behavior of different hardware architectures
  - Simulators generally have very high overhead
  - A hardware-level VM utilizes the underlying physical processor directly

VMM’s in the Past

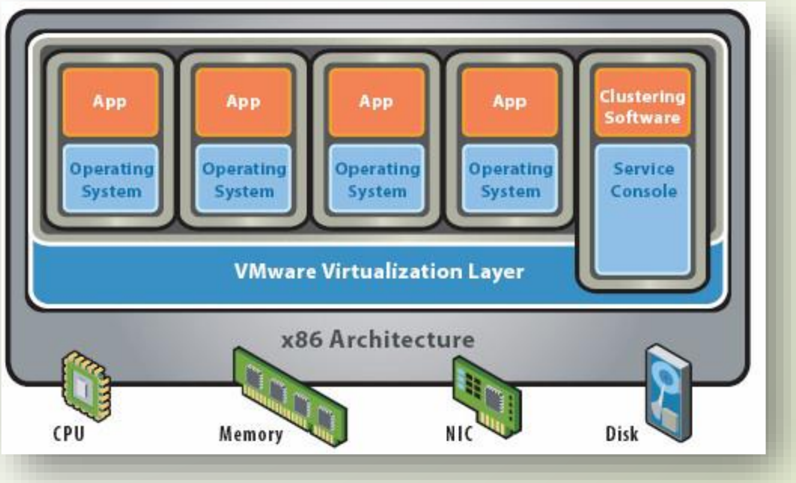
- Hardware-level VMs since’60s
  - IBM S/360, IBM VM/370 mainframe systems
  - Timeshare multiple single-user OS instances on expensive hardware
- Classical VMM
  - Run VM directly on hardware
  - “Trap and emulate” model for privileged instructions
  - Vendors had control over proprietary hardware, operating systems, VMM



From IBM VM/370 product announcement, *ca. 1972*

The Role of the Hypervisor

- Hypervisors (Virtual Machine Monitor / VMM) sit between virtual machines and the hardware.
- Divides a single physical server for multiple operating systems to interact with the underlying hardware
- Allocates and provides resources and hardware to access to the virtual machines it serves.

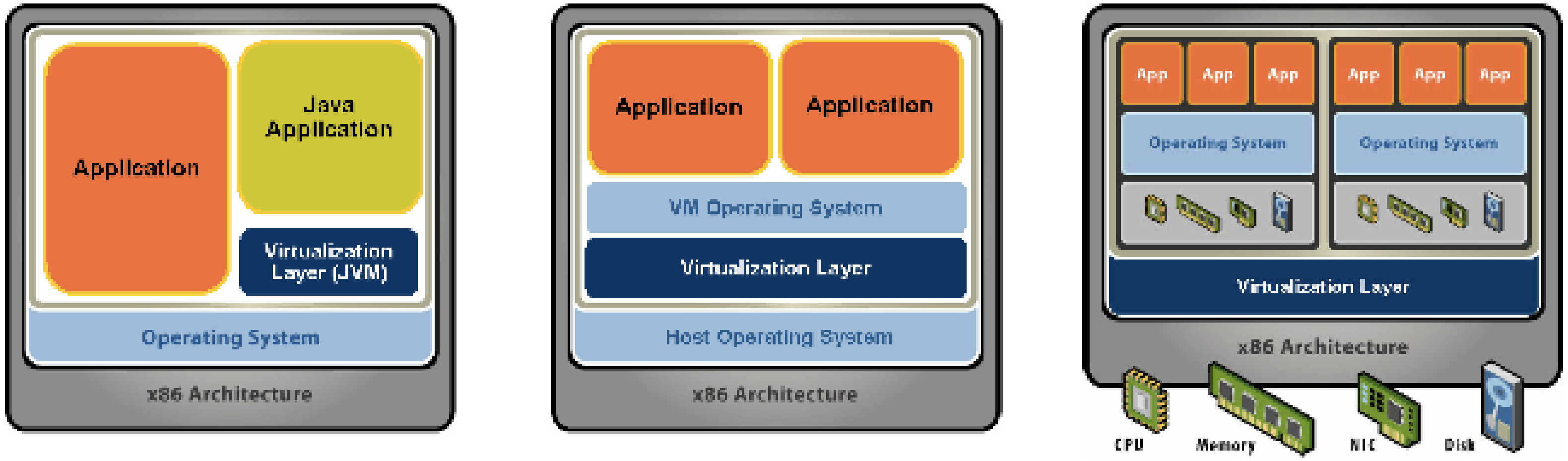


System Virtual Machine Monitor Architectures

- Traditional
- Hosted
  - Install as application on existing x86 “host” OS, e.g. Windows, Linux, OS X
  - Small context-switching driver
  - Leverage host I/O stack and resource management
    - Examples: VMware Player/Workstation/Server, Microsoft Virtual PC/Server, Parallels Desktop
- Bare-Metal Architecture
  - “Hypervisor” installs directly on hardware
  - Acknowledged as preferred architecture for high-end servers
  - Examples: VMware ESX Server, Xen, Microsoft Viridian (2008)

Virtualisation Alternatives

Virtual machines abstracted using a layer at different places

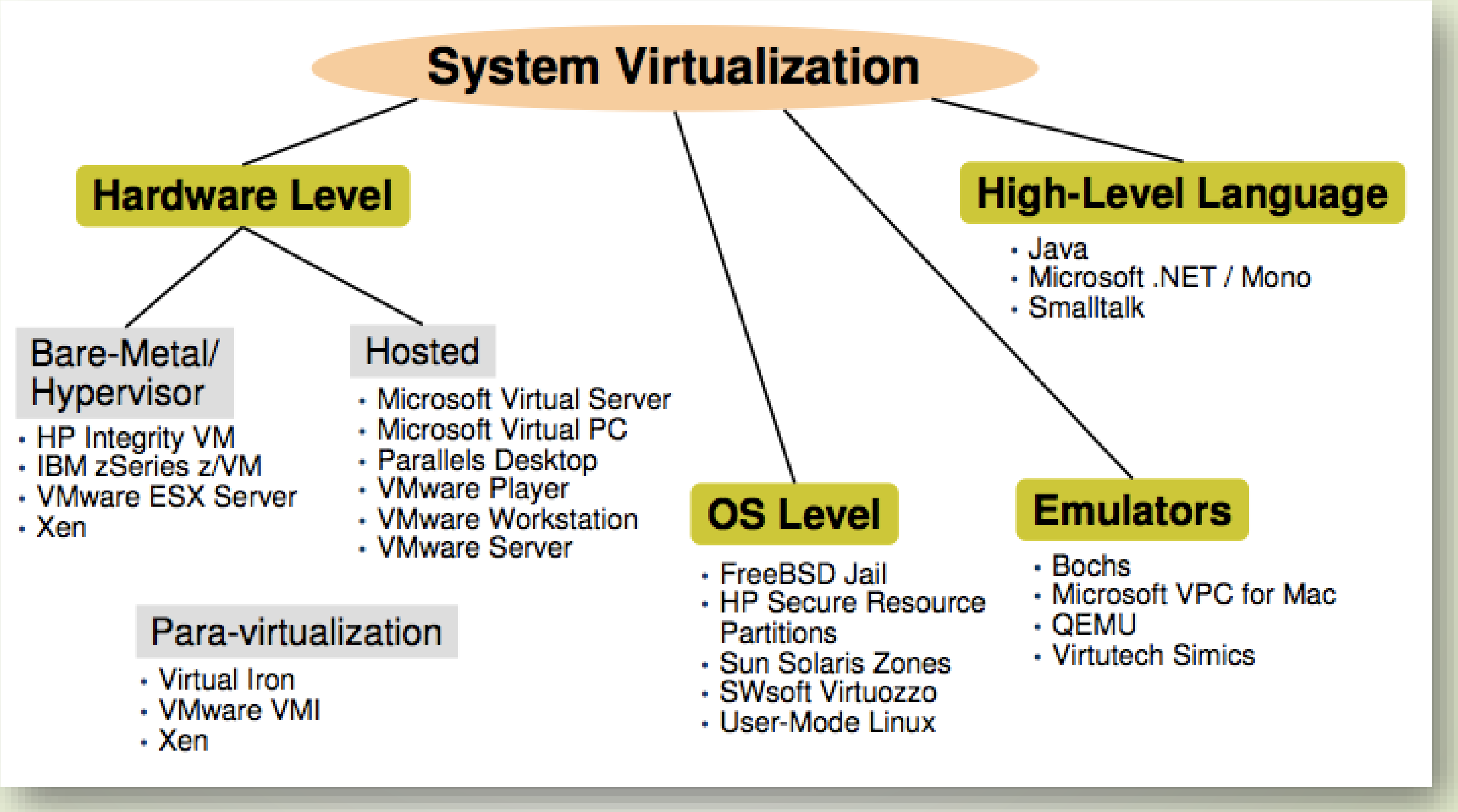


Language Level

OS Level

Hardware Level

Examples



System Virtualization

Hardware Level

Bare-Metal/ Hypervisor

- HP Integrity VM
- IBM zSeries z/VM
- VMware ESX Server
- Xen

Hosted

- Microsoft Virtual Server
- Microsoft Virtual PC
- Parallels Desktop
- VMware Player
- VMware Workstation
- VMware Server

Para-virtualization

- Virtual Iron
- VMware VMI
- Xen

OS Level

- FreeBSD Jail
- HP Secure Resource Partitions
- Sun Solaris Zones
- SWsoft Virtuozzo
- User-Mode Linux

Emulators

- Bochs
- Microsoft VPC for Mac
- QEMU
- Virtutech Simics

High-Level Language

- Java
- Microsoft .NET / Mono
- Smalltalk

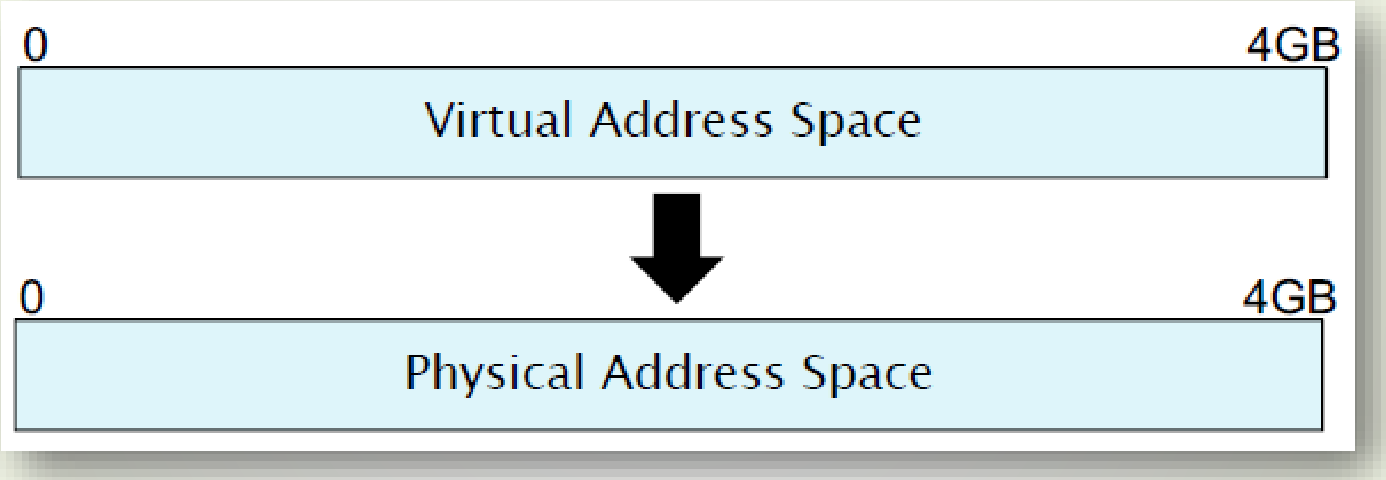
Overview

- Processor Virtualization
  - Classical techniques
  - Software x86 VMM
  - Hardware-assisted x86 VMM
  - Para-virtualization

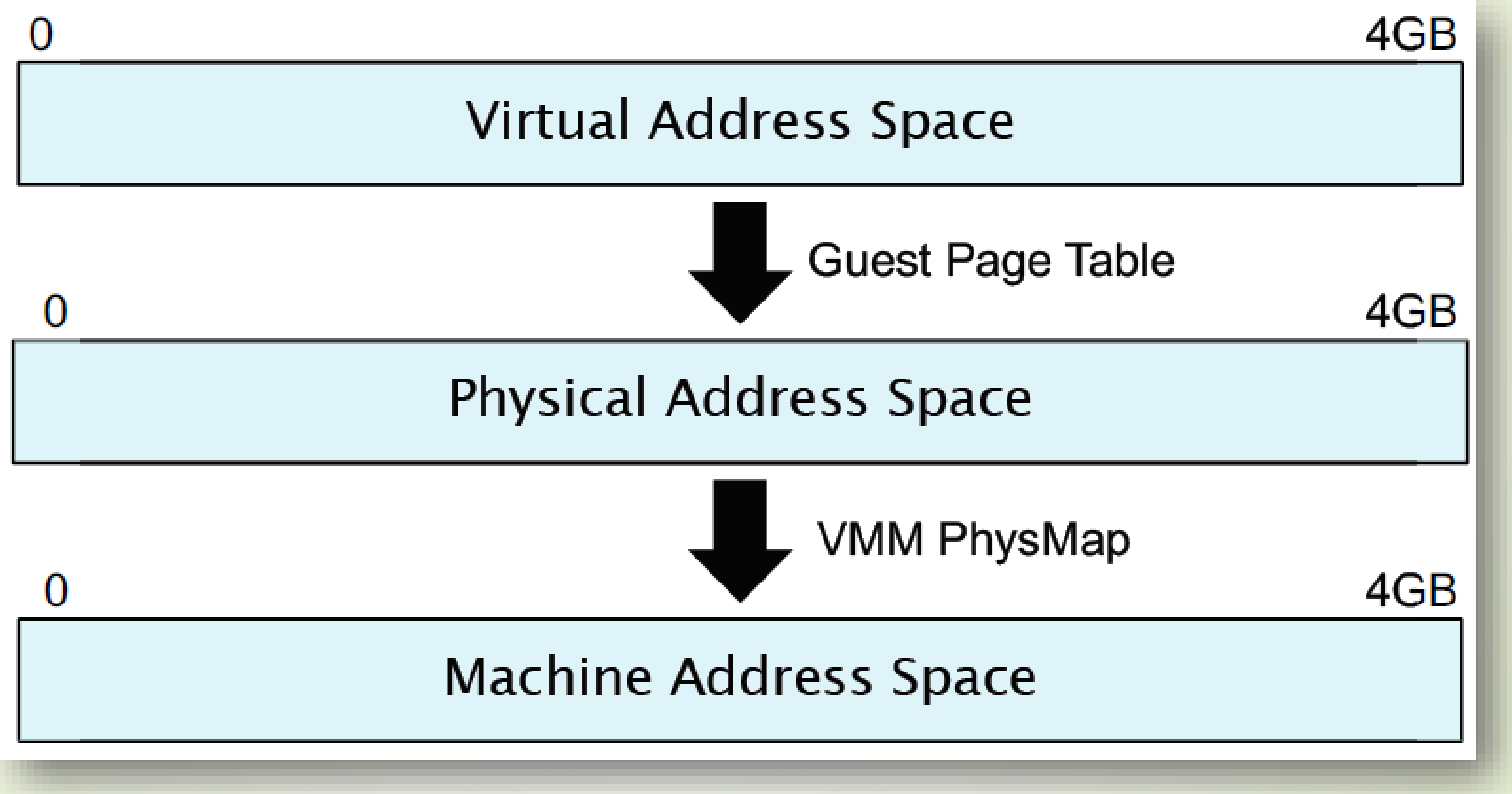
Classical Instruction Virtualisation

- Trap and Emulate
  - Run guest operating system deprivileged
  - All privileged instructions trap into VMM
  - VMM emulates instructions against virtual state e.g. disable virtual interrupts, not physical interrupts
  - Resume direct execution from next guest instruction
- Implementation Technique
  - This is just one technique
  - Popek and Goldberg criteria permit others

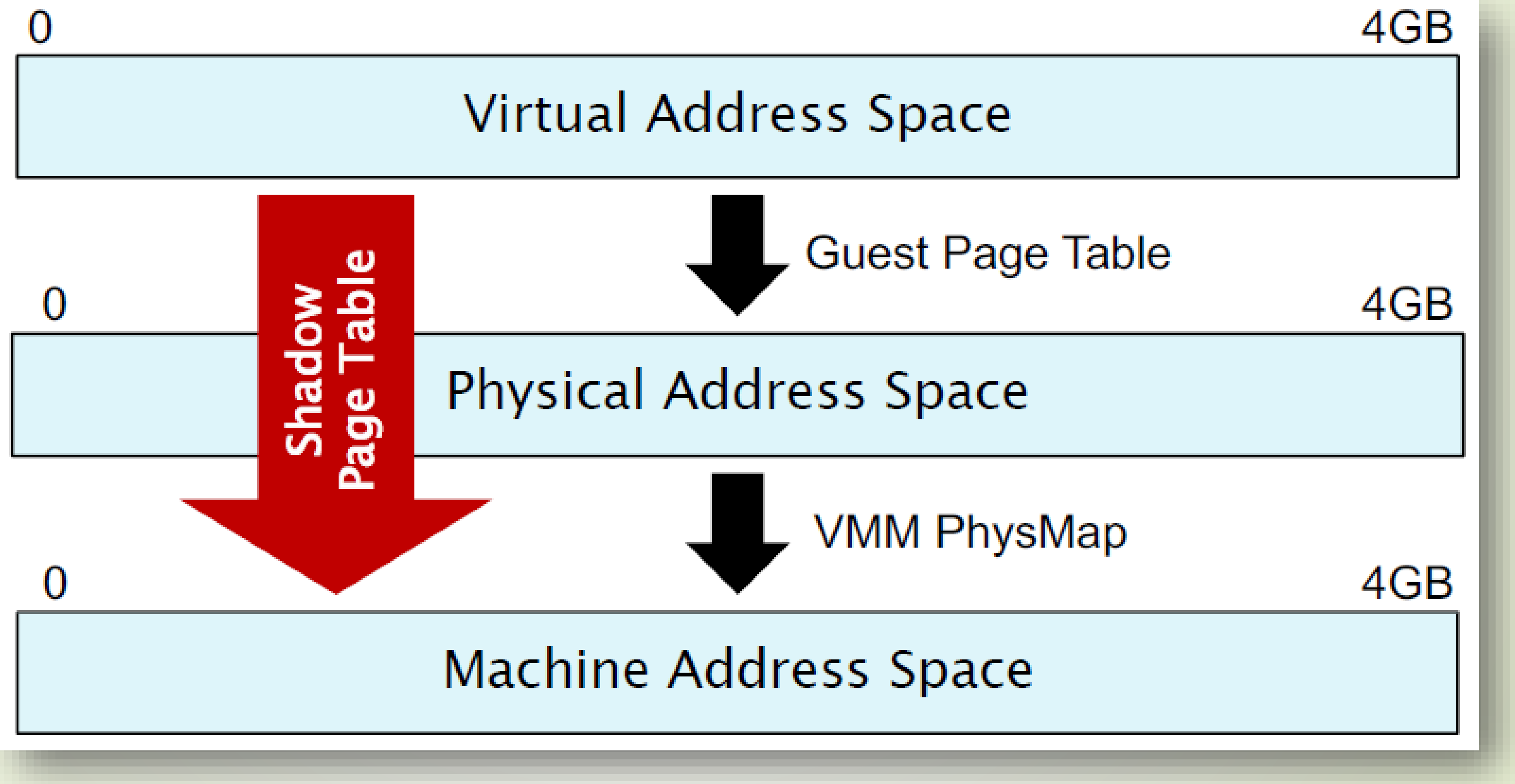
Traditional Address Spaces



Virtualised Address Space



Virtualised Address Spaces with Shadow Page Tables



0 4GB

Virtual Address Space

Shadow Page Table

Guest Page Table

0 4GB

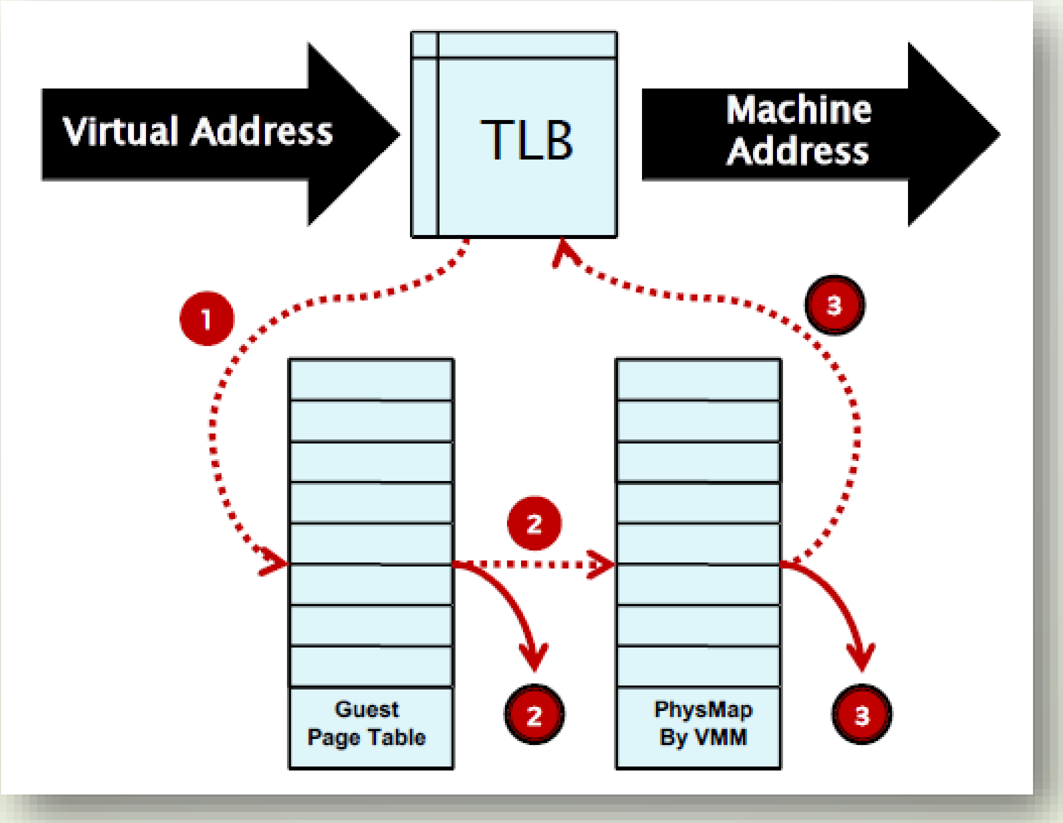
Physical Address Space

VMM PhysMap

0 4GB

Machine Address Space

Virtualised Address Translation with Nested Page Tables



Virtualised Address Translation with Nested Page Tables

1. If there is a miss in the TLB.
   - The hardware will walk the guest page table to find the mapping.
2. One of two things can happen:
   - The required mapping is not present. A page fault exception is generated to the VMM. The VMM typically passes this exception onto the guest – a true page fault.
   - The required mapping is found in the page. The hardware proceeds to walk the second page table.
3. During the hardware lookup into the PhysMap one of two things can happen:
   - The required mapping is not present. A page fault is generated to the VMM.
   - If the guest mapping is present, then the hardware places the composite mapping in the TLB and the instruction is restarted.

Issues with Nested Page Tables

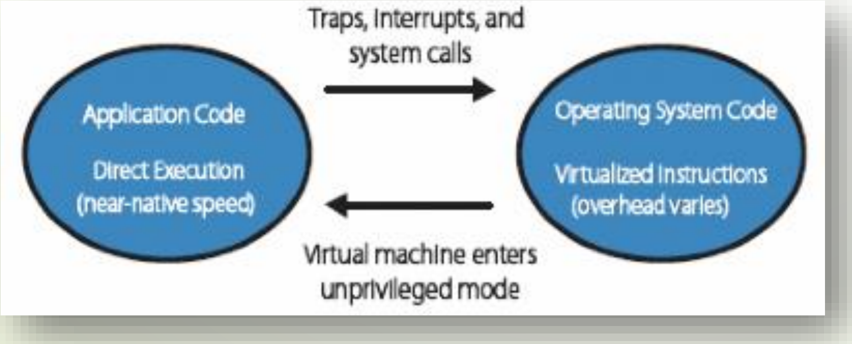
- Positives
  - Simplifies monitor design
  - No need for page protection calculus
- Negatives
  - Guest page table is in physical address space
  - Need to walk PhysMap multiple times
    - Need physical-to-machine mapping to walk guest page table
    - Need physical-to-machine mapping for original virtual address

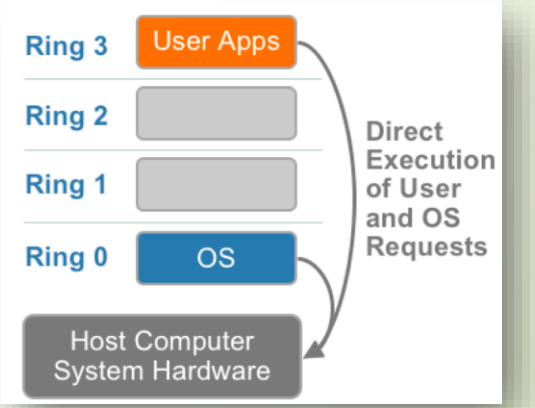
Classical VMM Performance

- Native Speed Except for Traps
  - No overhead in direct execution
  - Overhead = trap frequency × average trap cost
- Trap Sources
  - Most frequent: Guest page table traces
  - Privileged instructions
  - Memory-mapped device traces

Software VMM: Binary Translation

- Direct execute unprivileged guest application code
  - Will run at full speed until it traps, get an interrupt, etc.
- "Binary translate" all guest kernel code, run it unprivileged
  - Since x86 has non-virtualisable instructions, proactively transfer control to the VMM (no need for traps)
  - Safe instructions are emitted without change
  - For "unsafe" instructions, emit a controlled emulation sequence
  - VMM translation cache for good performance




Ring 3
User Apps
Ring 2
Ring 1
Ring 0
OS
Direct Execution of User and OS Requests
Host Computer System Hardware

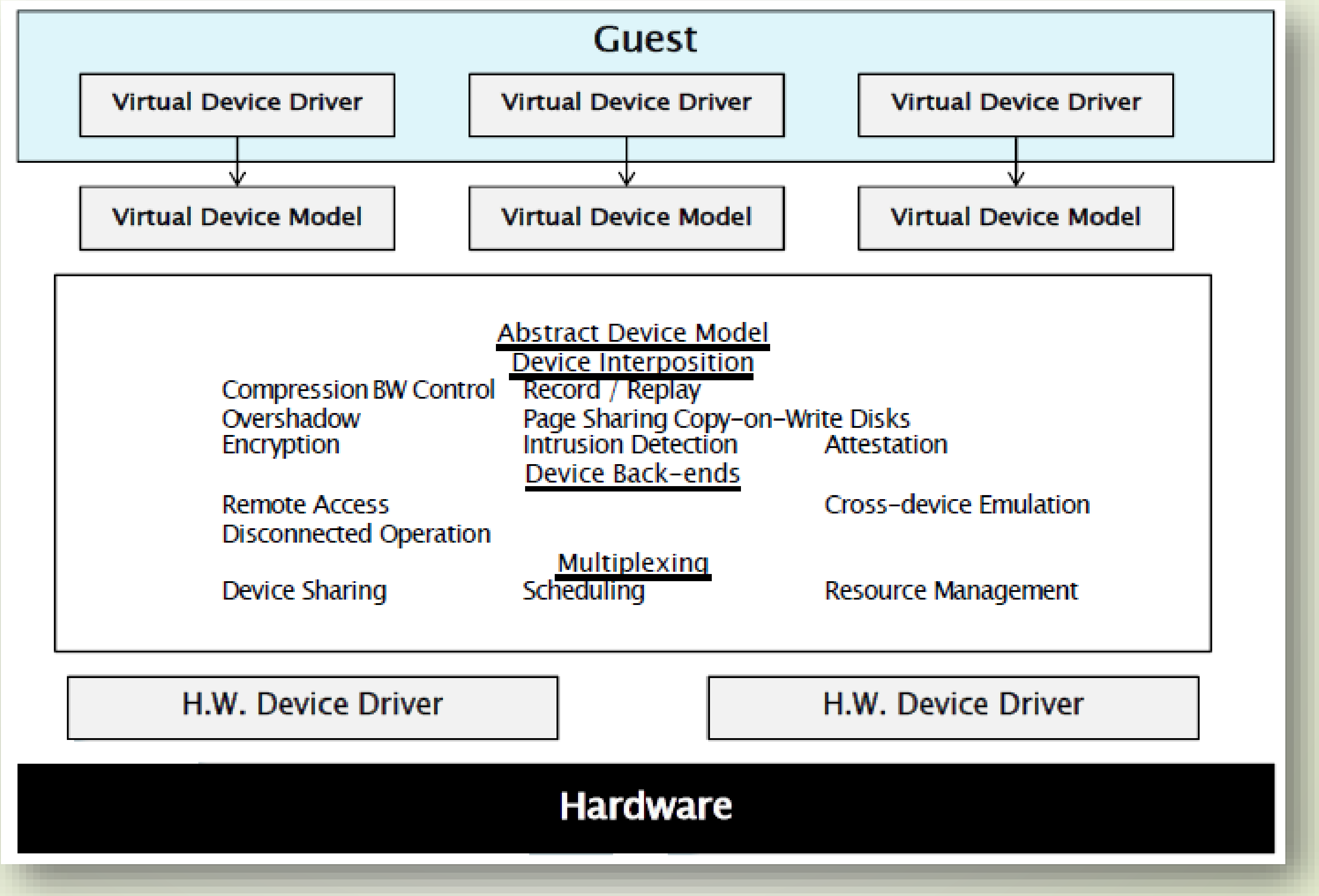
Hardware Assisted VMM

- Recent x86 Extension
  - 1998 – 2005: Software-only VMMs using binary translation
  - 2005: Intel and AMD start extending x86 to support virtualization
- First-Generation Hardware
  - Enables classical trap-and-emulate VMMs
  - Intel VT, aka "Vanderpool Technology"
  - AMD SVM, aka "Pacifica"
- Performance
  - VT/SVM help avoid BT, but not MMU ops (actually slower!)
  - Main problem is efficient virtualization of MMU and I/O, Not executing the virtual instruction stream

What is Paravirtualisation?

- Full Virtualization
  - No modifications to guest OS
  - Excellent compatibility, good performance, but complex
- Paravirtualisation Exports Simpler Architecture
  - Term coined by Denali project in ’01, popularized by Xen
  - Modify guest OS to be aware of virtualization layer
  - Remove non-virtualisable parts of architecture
  - Avoid rediscovery of knowledge in hypervisor
  - Excellent performance and simple, but poor compatibility
- Ongoing Linux Standards Work
  - “Paravirt Ops” interface between guest and hypervisor

I/O Virtualisation



Guest

Virtual Device Driver

Virtual Device Driver

Virtual Device Driver

Virtual Device Model

Virtual Device Model

Virtual Device Model

Abstract Device Model

Device Interposition

Compression BW Control
Overshadow
Encryption

Record / Replay
Page Sharing Copy-on-Write Disks
Intrusion Detection

Attestation

Device Back-ends

Remote Access
Disconnected Operation

Cross-device Emulation

Multiplexing

Device Sharing

Scheduling

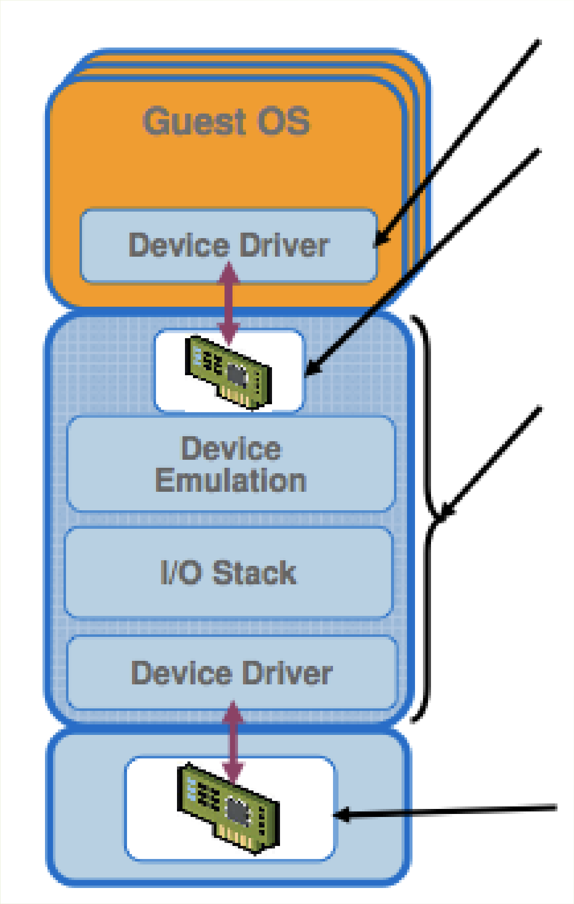
Resource Management

H.W. Device Driver

H.W. Device Driver

Hardware

I/O Virtualisation



Guest Device Driver

Virtual Device

- Model existing device, e.g. *E1000*
- Model an idealized device, e.g. vmxnet

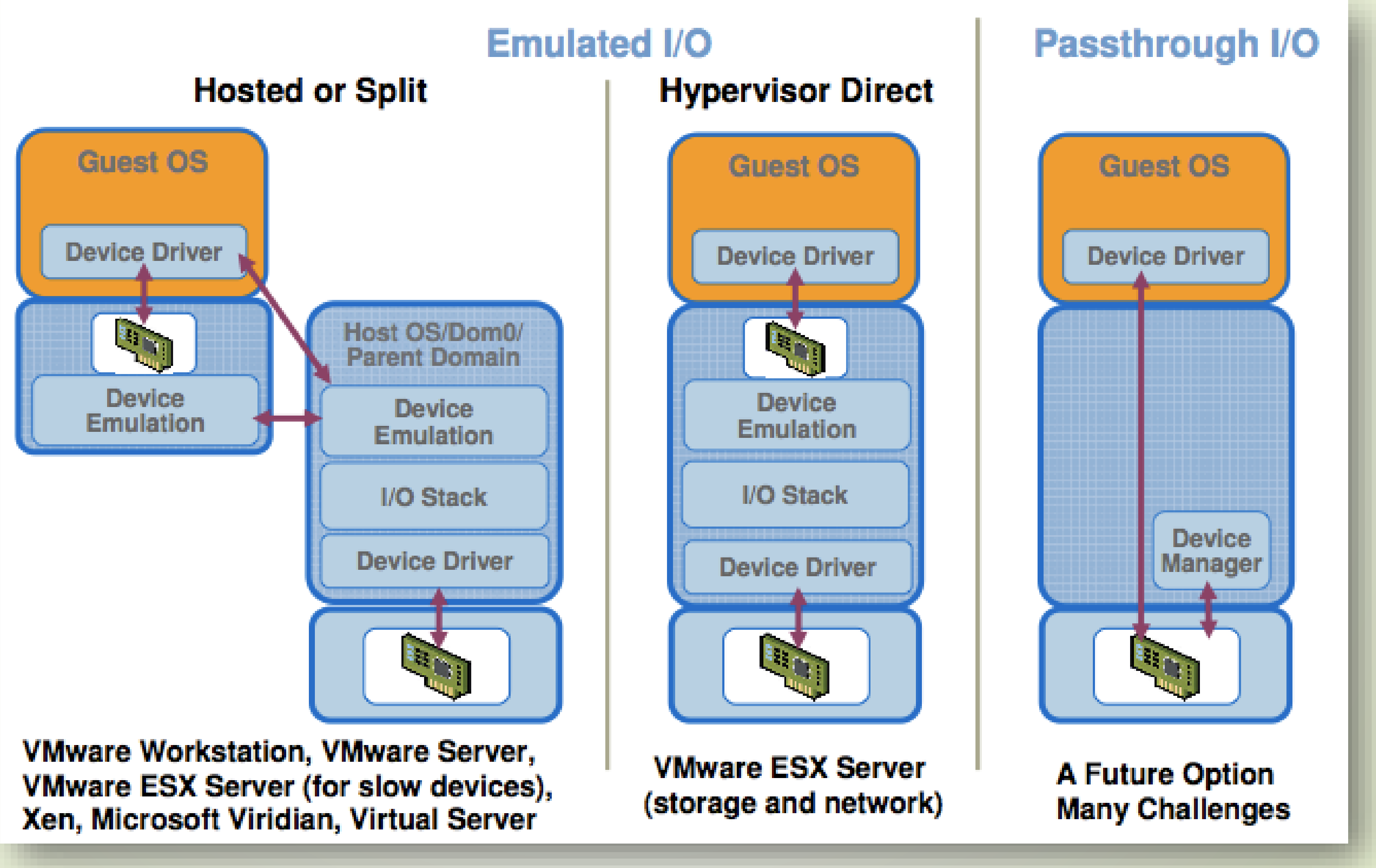
Virtualization Layer

- Emulates the virtual device
- Remaps guest and real I/O addresses
- Multiplexes and drives physical device
- Provides additional features, *e.g transparent NIC teaming*

Real Device

- Physical hardware
- *Likely to be different than virtual device*

I/O Virtualisation Implementations



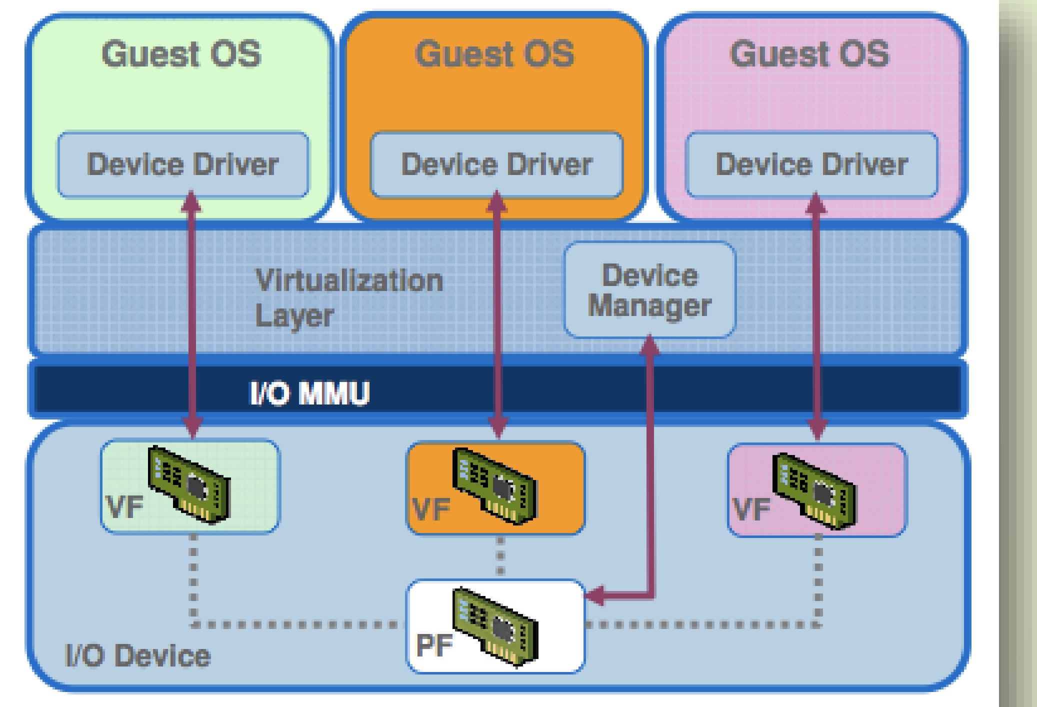
VMware Workstation, VMware Server, VMware ESX Server (for slow devices), Xen, Microsoft Viridian, Virtual Server

VMware ESX Server (storage and network)

A Future Option
Many Challenges

Passthrough I/O Virtualisation

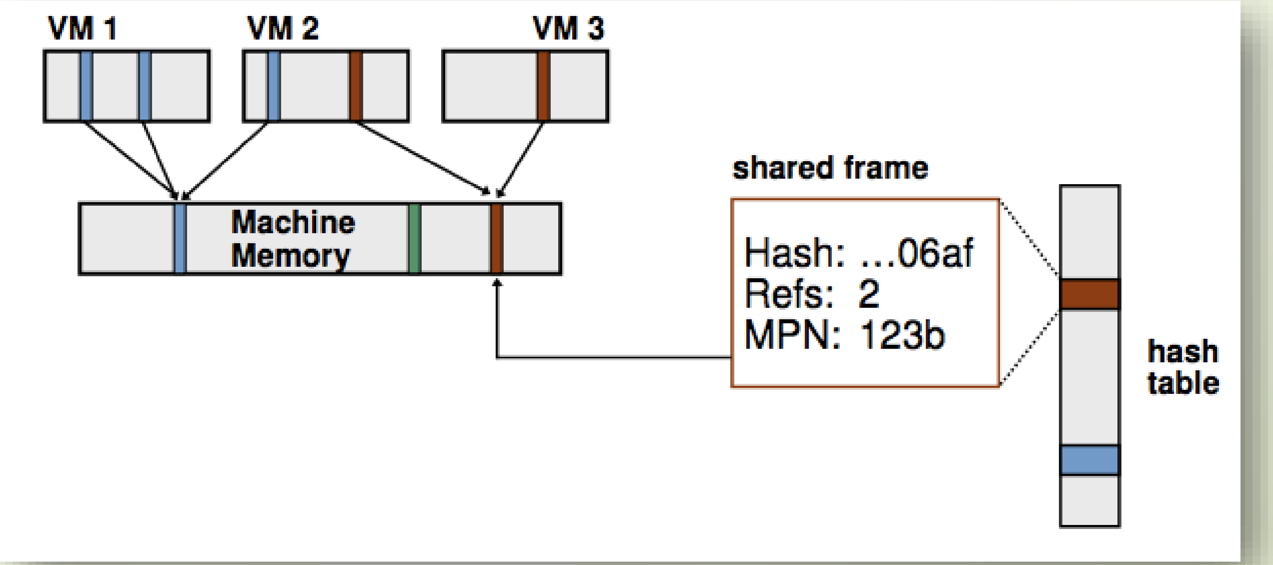
- High Performance
  - Guest drives device directly
  - Minimises CPU utilization
- Enabled by HW Assists
  - I/O-MMU for DMA isolation
- Challenges
  - Hardware independence
  - Migration, suspend/resume
  - Memory over commitment



PF = Physical Function, VF = Virtual Function

Memory Management

- Desirable capabilities
  - Efficient memory overcommitment
  - Accurate resource controls
  - Exploit sharing opportunities



Hosted Virtual Machines

- Goal:
  - Run Virtual Machines as an application on an existing Operating System
- Why
  - Application continuity
  - Reuse existing device drivers
  - Leverage OS support
    - File system
    - CPU Scheduler
  - VM management platform

Hosted Architecture Tradeoffs

- Positives
  - Installs like an application
    - No disk partitioning needed
    - Virtual disk is a file on host file system
    - No host reboot needed
  - Runs like an application
    - Uses host schedulers
- Negatives
  - I/O path is slow
    - Requires world switch
  - Relies on host scheduling
    - May not be suitable for intensive VM workloads

VMware ESXi

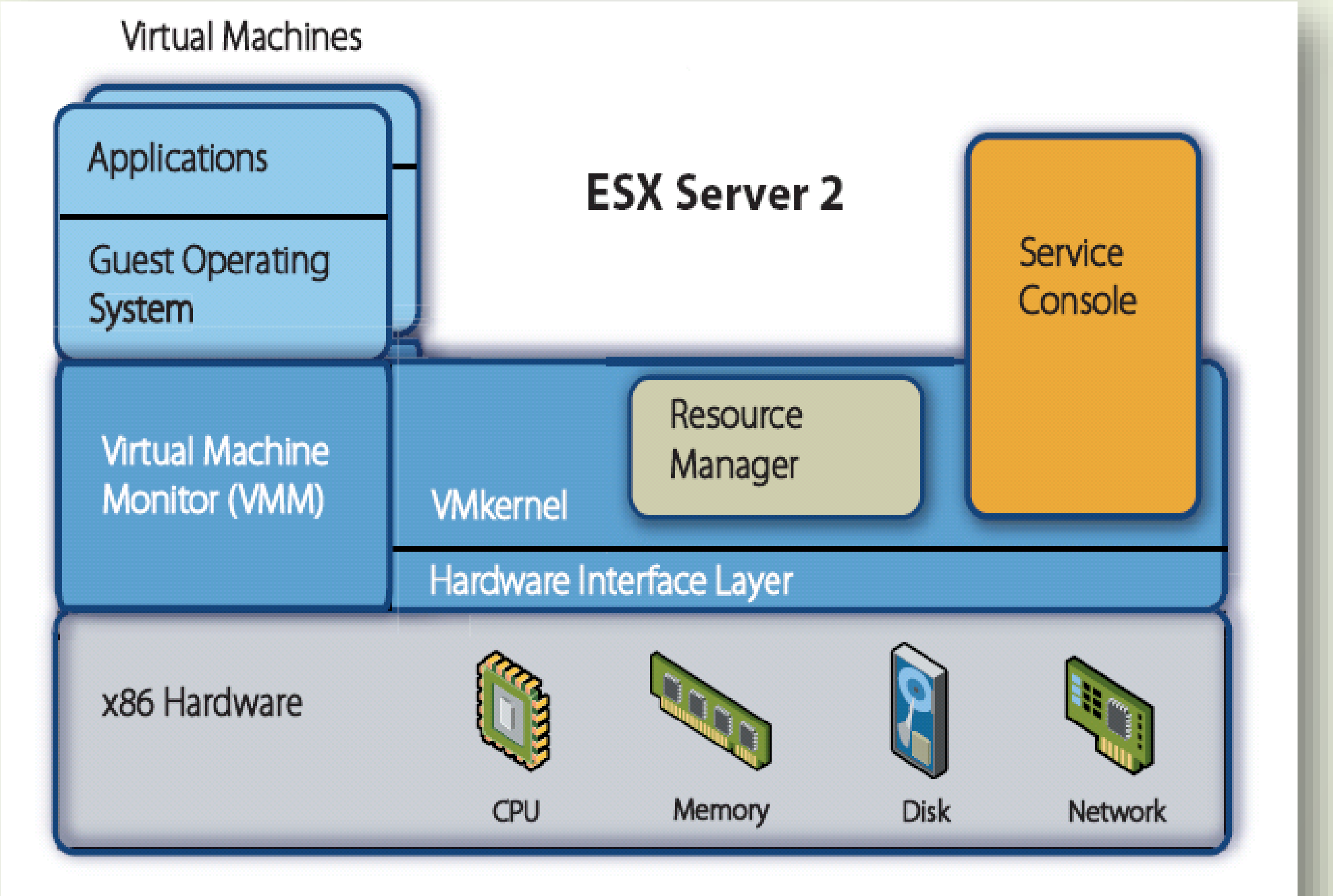
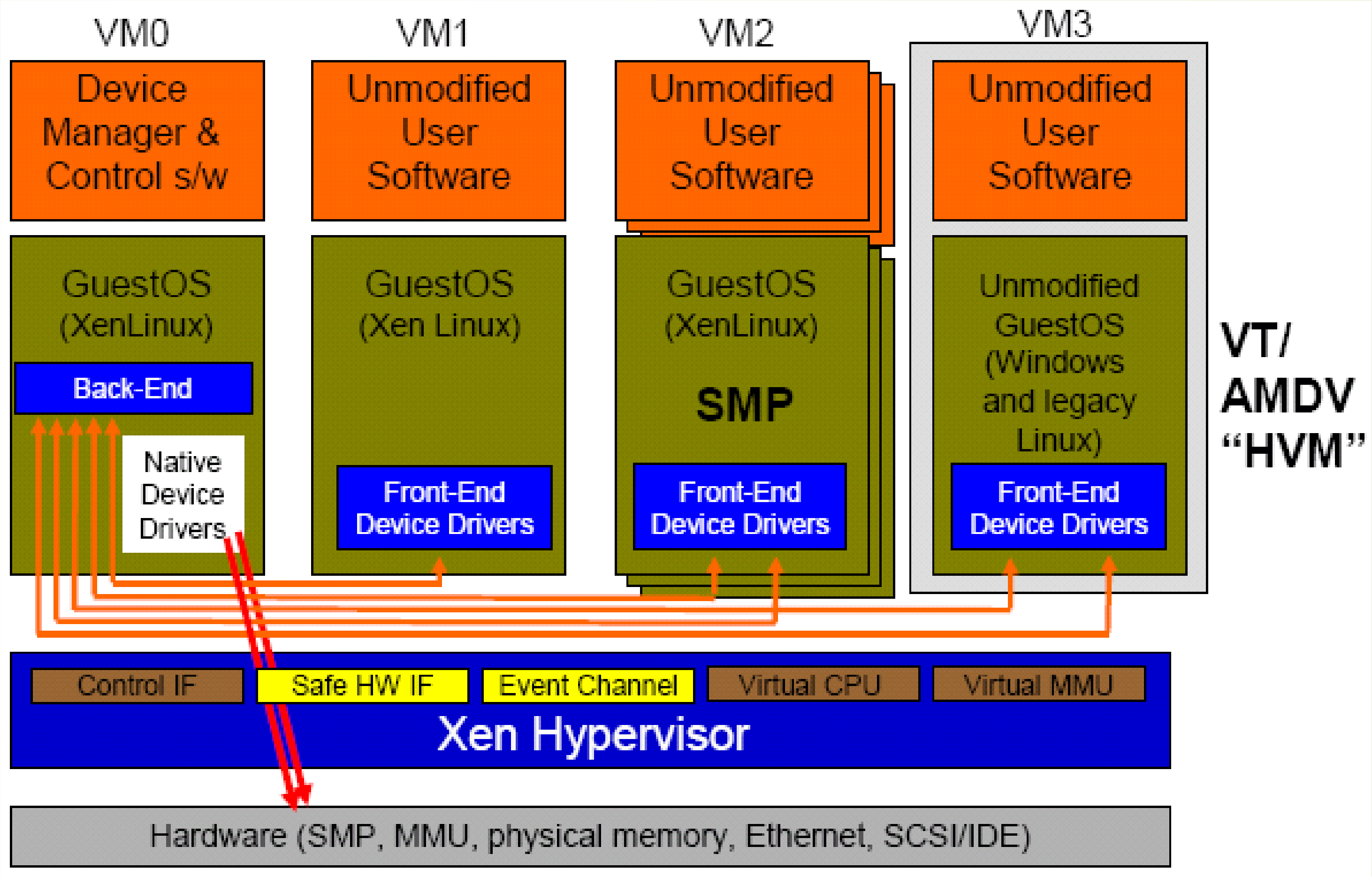


Figure 1: ESX Server architecture

Source: http://www.vmware.com/pdf/esx2_performance_implications.pdf

Hybrid Ex 2 – Xen 3.0

- Para – virtualization
  - Linux Guest
- Hardware-supported virtualization
  - Unmodified Windows
- Isolated Device Drivers



Source: Ottawa Linux Symposium 2006 presentation. http://www.cl.cam.ac.uk/netos/papers/

Server Virtualisation

Agenda

- **Objectives**
- Role of Server Virtualization
- Role of Hypervisor
- Types of Virtualization
- Managing Virtual Machines Resources
- Scaling up Virtual Machines
- Configuring Virtual Networking in Virtual Machines
- Summary

Objectives

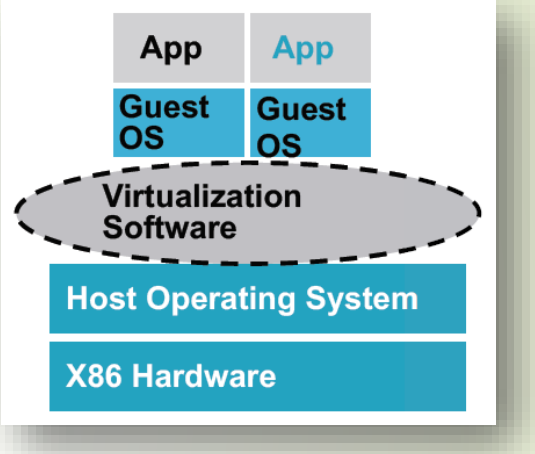
- Describe the need for virtualization on servers
- Identify the usage of server virtualization.
- Identify 2 types of hypervisors
- Identify the 3 different types of virtualization
- Describe the types of advanced virtual machine operations used in server management.
- Understand how virtual machine scalability is achieved.
- Describe how networks in virtual servers are connected

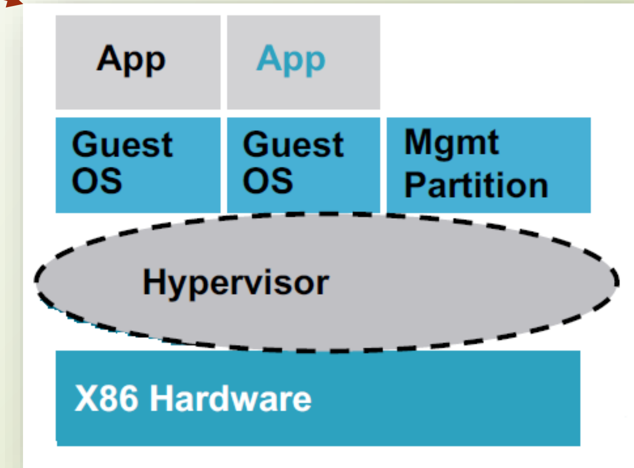
Why the need for Virtualisation?

- Effective Resource Usage
  - Resources (RAM/CPU/DISK) can be allocated wherever needed.
- Ease of Management
  - Unused machines can be kept indefinitely off until when required, and do not consume any resources other than disk space.
  - Application provisioning, maintenance, high availability and disaster recovery
- Security
  - Applications can be kept on separate servers.
  - Separate servers do not pose a risk to each other
  - Especially true when virtual machines are owned by mutually untrusting users.

Server Virtualisation Scenarios

- Hardware-based virtualization
- Software-based virtualization
  - Hosted (application virtualization)
  - Hypervisor
    - Full virtualization (binary translation)
    - Para-virtualization (OS assisted)
    - Hardware-assisted virtualization (Intel VT-x/AMD-V)

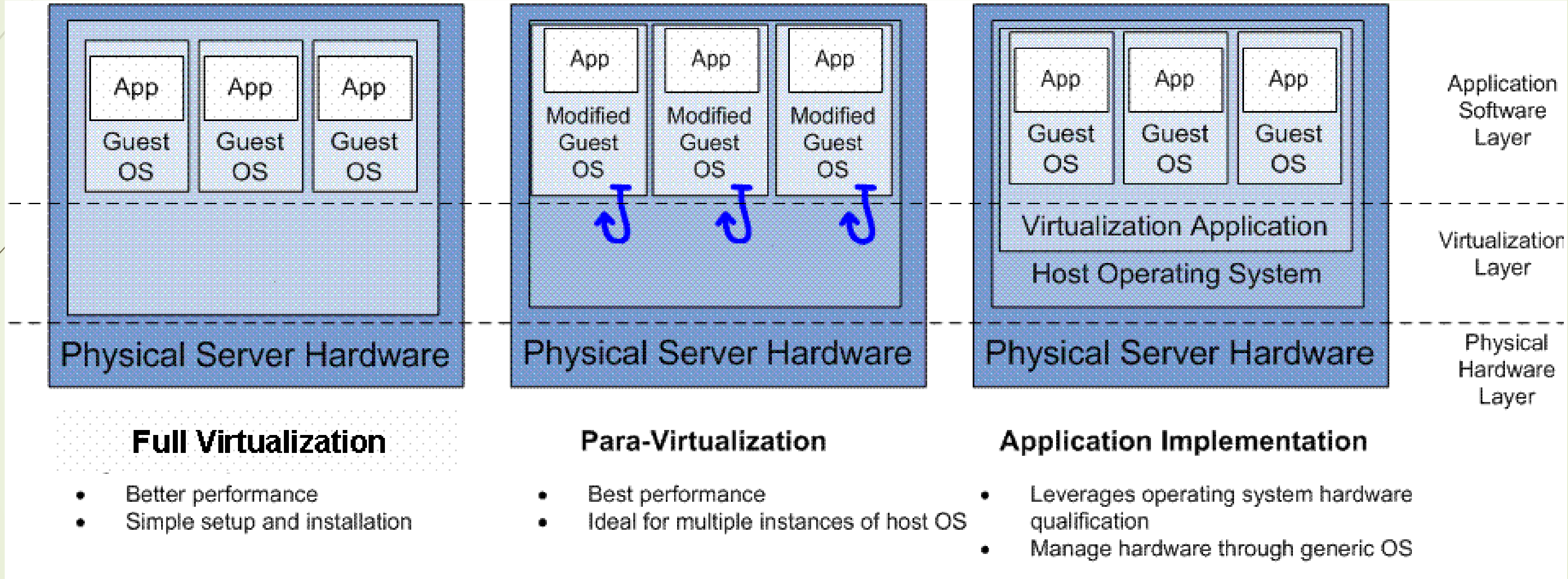




Common Hypervisors and Types of Virtualisation

- Windows 2008-2019(64 bit only)
  - Base OS includes hypervisor/ Emulated Hardware for VMs
  - Hypervisor runs on top of host Operating System.
- Linux Open Source Virtualization – Choice of 2
  - Xen (para-virtualized – same hardware as host)
  - KVM (Hardware Assisted -/ Emulated Hardware for VMs)
- Commercial Hypervisors
  - VMware vSphere
  - VMWare vSphere Hypervisor (ESXi)
  - VMWare Workstation

Software-Based Virtualisation - Examples



Full Virtualization

- Better performance
- Simple setup and installation

Para-Virtualization

- Best performance
- Ideal for multiple instances of host OS

Application Implementation

- Leverages operating system hardware qualification
- Manage hardware through generic OS

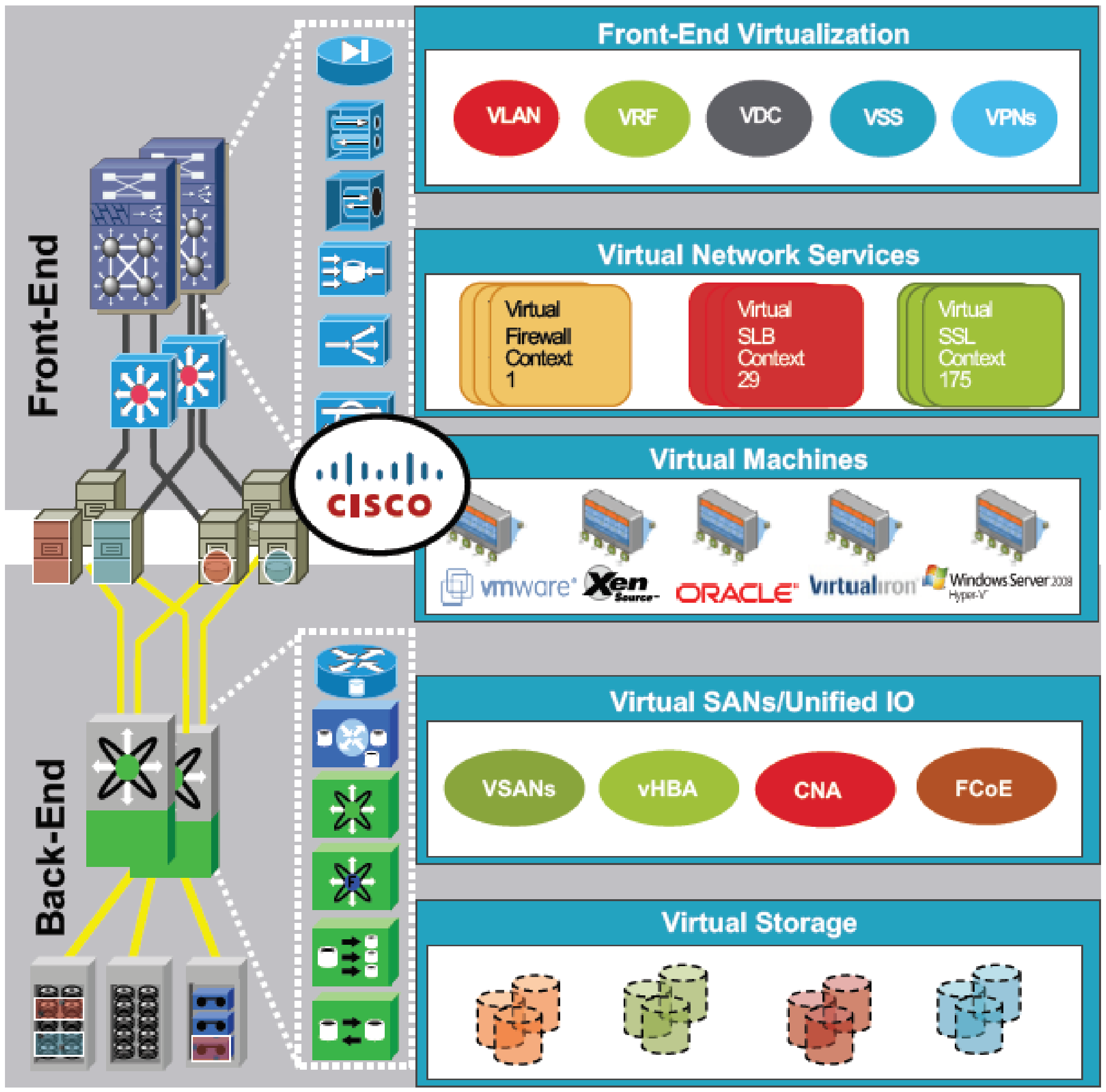
- VMware ESX server
- Microsoft HyperV
- Xen (with AMD-SVM or Intel VM-T)
- VirtualIron (hardware-assisted)

- Xen (with traditional hardware)
- Oracle VM server

- VMware server
- VMware workstation

Hardware Assisted Virtualisation

- Hardware-assisted virtualization is a virtualization approach that enables efficient full virtualization using help from hardware capabilities, primarily from the host processors.

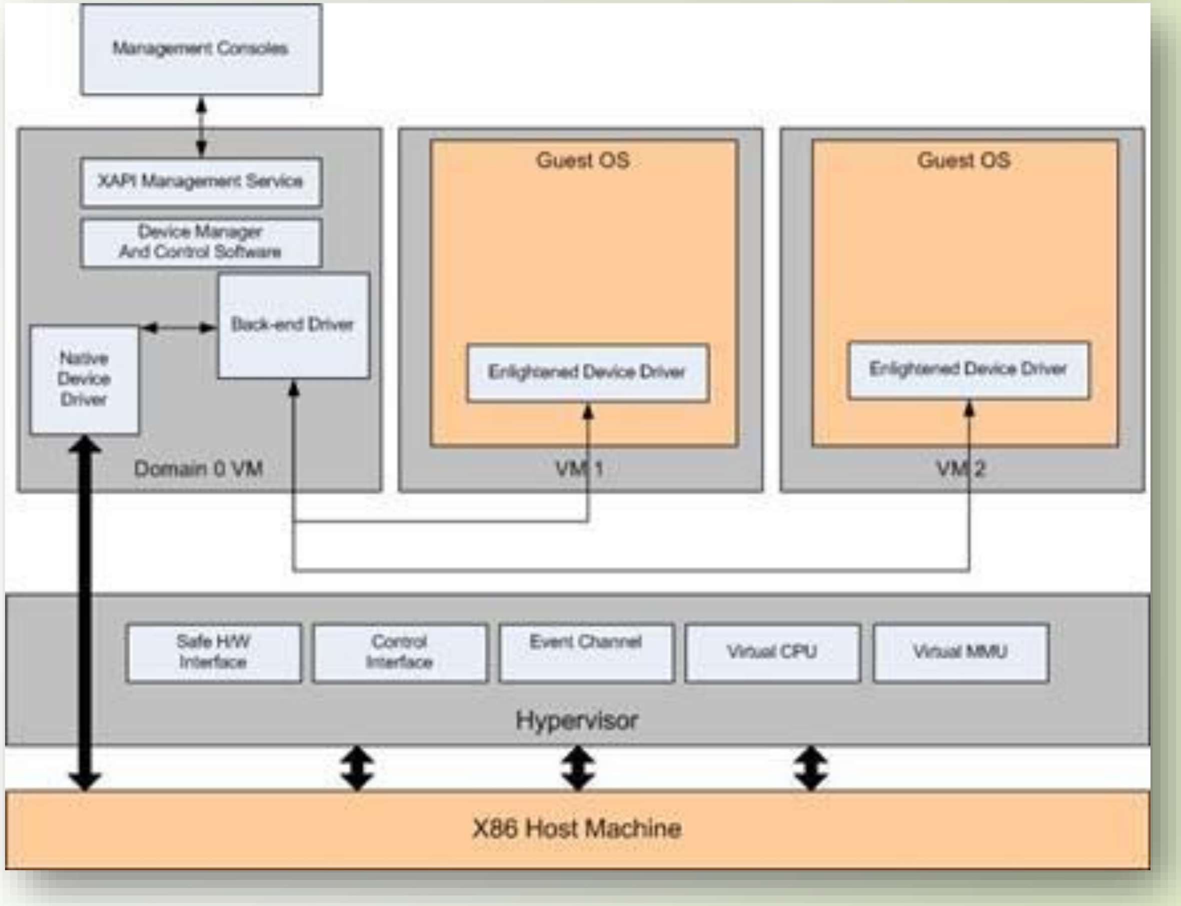


Hardware Assisted Virtualisation

- Full virtualization on x86 has significant costs in hypervisor complexity and run-time performance.
- Same as Full Virtualization, however the Guest VM can communicate directly with the hardware bypassing the hypervisor for certain machine instructions.
- Explicit hardware support in the host CPU must be present.
  - **Hardware-Assisted CPU Virtualization**
    - (Intel VT-x and AMD AMD-V)
  - **Hardware-Assisted MMU Virtualization**
    - (Intel EPT and AMD RVI)
  - **Hardware-Assisted I/O MMU Virtualization**
    - (Intel VT-d and AMD AMD-Vi)

Hardware Assisted Virtualisation

- These extensions address the parts of x86 that are difficult or inefficient to virtualize, providing additional support to the hypervisor. This enables simpler virtualization code and a higher performance for full virtualization.
- Offers some speed improvements as instructions handled in hardware as compared to software (hypervisor)



Enabling Hardware Assisted Virtualisation - CPU

- First-generation hardware support for x86 virtualization with AMD Virtualization ™ (AMD-V™) and Intel® VT-x technologies in 2006
- Requires Hardware Data Execution Prevention : prevent an application or service from executing code from a non-executable memory region. This helps prevent certain exploits that store code via a buffer overflow, for example.
- Intel VT-x (initially codenamed *Vanderpool*)
  - Intel refers to it as Execute Disable (XD). This feature must be enabled in the system BIOS.
- AMD AMD-V (also called SVM and initially codename *Pacifica*)
  - AMD refers to it as No Execute (NX). This feature must be enabled in the system BIOS.
- When you enable hardware assisted virtualization (Intel VT or AMD AMD-V) in the BIOS, you must **POWER CYCLE** the system. **NOT REBOOT.**

Enabling Hardware Assisted Virtualisation - MMU

- Intel introduced its second generation of hardware support that incorporates Memory Management Unit (MMU) virtualization, called **Extended Page Tables (EPT)**. AMD also introduced its own MMU virtualization, called **Rapid Virtualization Indexing (RVI)** or Nested Page Tables (NPT)
- EPT/RVI -enabled systems can improve performance compared to using native shadow paging for MMU virtualization.
- EPT provides performance gains of up to 48% for MMU-intensive benchmarks and up to 600% for MMU-intensive microbenchmarks
- RVI provides performance gains of up to 42% for MMU-intensive benchmarks and up to 500% for MMU-intensive microbenchmarks
- VMM can now rely on hardware to eliminate the need for shadow page tables. This removes much of the overhead otherwise incurred to keep the shadow page tables up-to-date
- Turn these features on in BIOS to enable.

Enabling Hardware Assisted Virtualisation – I/O

- A newer processor feature is an I/O memory management unit that remaps I/O DMA transfers and device interrupts. This can allow virtual machines to have direct access to hardware I/O devices, such as network cards.
- In AMD processors this feature is called **AMD I/O Virtualization (AMD-Vi or IOMMU)** and in Intel processors the feature is called **Intel Virtualization Technology for Directed I/O (VT-d)**.
- In VMWare, VMDirectPath I/O leverages Intel VT-d and AMD-Vi hardware support to allow guest operating systems to directly access hardware devices. In the case of networking, VMDirectPath I/O allows the virtual machine to access a physical NIC directly rather than using an emulated or a para-virtualized device
- While VMDirectPath I/O has limited impact on throughput, it reduces CPU cost for networking-intensive workloads

VMware Type 1 and Type 2 Offerings

	VMware Player	VMware Workstation	VMware vSphere Hypervisor (ESXi)	VMware vSphere
Hypervisor Type	Hosted	Hosted	Bare Metal	Bare Metal
Typical Use Case	Test	Test & Dev	Production, Test & Dev	Tier 1 Apps, Production, Test & Dev
# of VMs per host	1-2	<10	<10	>10
Dedicated Server Required	No	No	Yes	Yes
Centralized Management	No	No	No	Yes

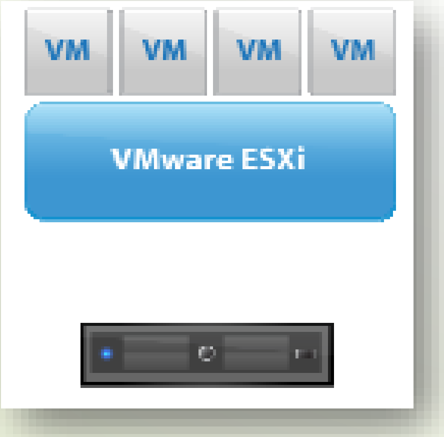
VMware Type 1 and Type 2 Offerings

	VMware Player	VMware Workstation	VMware vSphere Hypervisor (ESXi)	VMware vSphere
Remote Management	No	No	Yes	Yes
Headless Operation	No	Yes	Yes	Yes
Support Available	No	Yes	Yes	Yes
Transition Path to VMware vSphere	Easy. Convert virtual machines to run on vSphere using VMware converter.	Easy. Drag and drop to upload virtual machines to VMware vSphere.	Easiest. Virtual machines and hypervisor are compatible with paid editions of vSphere.	N/A
Pricing	Free	Paid	Free	Paid

Comparing ESXi to VMware vSphere

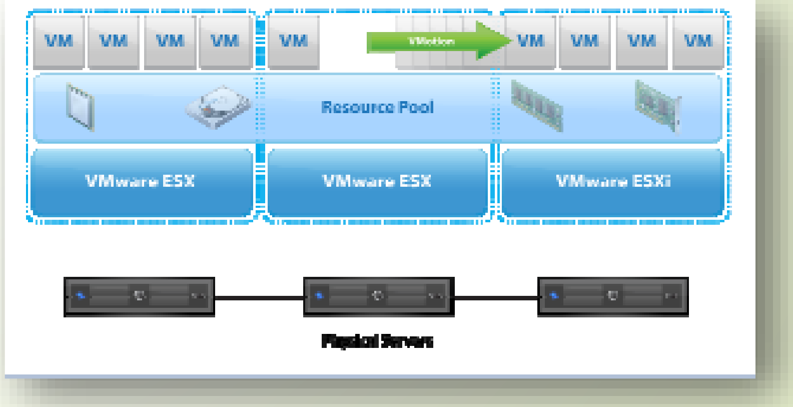
VMware vSphere Hypervisor (ESXi)

- Single server partitioning
- Production-class hypervisor
- Advanced server resource management
- FREE



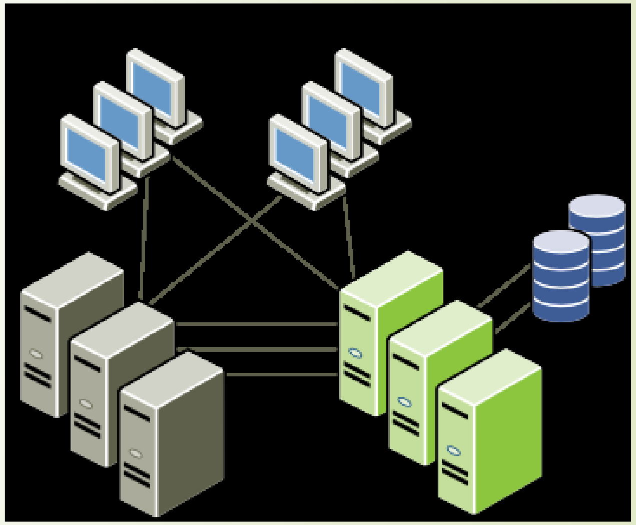
VMware vSphere (vCenter/Client/web client)

- Pools of computing resources
- Centralized management
- Built-in automation, availability and manageability
- All Editions include ESXi



VMware vSphere Components

- Each vCenter Server system manages multiple ESX hosts. You can run the vSphere Client and vSphere Web Access on multiple workstations.
- The major VMware vSphere components are:
  - **VMware ESX :** Provides a virtualization layer that abstracts the processor, memory, storage, and networking resources of the physical host into multiple virtual machines.
  - **vCenter Server:** A service that acts as a central administration point for ESX/ESXi hosts connected on a network. This service directs actions on the virtual machines and the hosts. The vCenter Server is the working core of vCenter.



VMware vSphere Components

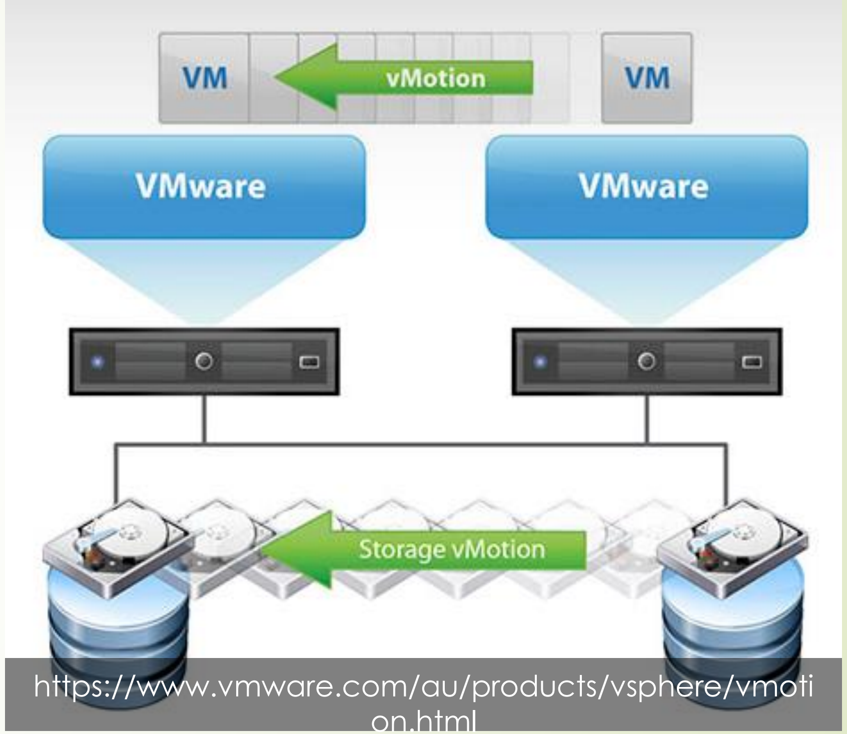
- **vCenter Server additional modules :** Provide additional capabilities eg. vCenter Update Manager, vCenter Converter, and vCenter Guided Consolidation Service and features to vCenter Server.
- **vSphere Client :** Installs on a Windows machine and is the primary method of interaction with VMware vSphere. The vSphere Client acts as a console to operate virtual machines and as an administration interface into the vCenter Server systems and ESX hosts.
- **VMware vSphere Web Access :** A browser-based interface for system administrators who need to access virtual machines remotely or without a vSphere Client.
- **Databases :** Organize all the configuration data for the Vmware vSphere environment.

VMotion

- VMotion allows you to quickly move an entire running virtual machine from one host to another without any downtime or interruption to the virtual machine This is also known as a “hot” or “live” migration.

vMotion

- The entire state of a virtual machine is encapsulated, and the VMFS file system allows both the source and the target ESX host to access the virtual machine files concurrently. The active memory and precise execution state of a virtual machine can then be rapidly transmitted over a high-speed network. The virtual machine retains its network identity and connections, ensuring a seamless migration process.



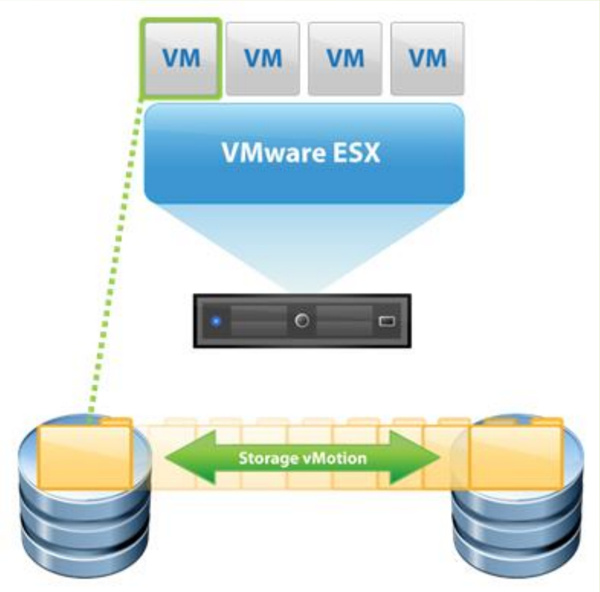
https://www.vmware.com/au/products/vsphere/vmotion.html

VMotion Operations

1. Migration request is made to move the virtual machine from ESX1 to ESX2.
2. vCenter Server verifies that the virtual machine is in a stable state on ESX1 and checks the compatibility of ESX2 (CPU, networking, etc.) to ensure that it matches that of ESX1.
3. The virtual machine is registered on ESX2.
4. The virtual machine state information (including memory, registers and network connections) is copied to ESX2. Additional changes are copied to a memory bitmap on ESX1.
5. The virtual machine is quiesced on ESX1 and the memory bitmap is copied to ESX2.
6. The virtual machine is started on ESX2 and all requests for the virtual machine are now directed to ESX2.
7. A final copy of the virtual machines memory is done from ESX1 to ESX2.
8. The virtual machine is un-registered from ESX1.
9. The virtual machine resumes operation on ESX2.

Storage vMotion

- Storage VMotion is a new feature introduced in ESX 3.5, it allows you to migrate a running virtual machine and its disk files from one datastore to another on the same ESX host .
- VMotion simply moves a virtual machine from one ESX host to another but keeps the storage location of the VM the same
- Storage vMotion changes the storage location of the virtual machine while it is running and moves it to another datastore on the same ESX host. The virtual machine can be moved to any datastore on the ESX host which includes local and shared storage.



Storage vMotion Operations

1. New virtual machine directory is created on the target datastore, virtual machine configuration files and all non-virtual disk files are copied to the target directory.
2. ESX host does a “self” VMotion to the target directory.
3. A snapshot (without memory) is taken of the virtual machine's disks in the source directory.
4. Virtual machine disk files are copied to the target directory.
5. Snapshot that is located in the source directory is consolidated into the virtual machine disk files located in the target directory.
6. Source disk files and directory are deleted.

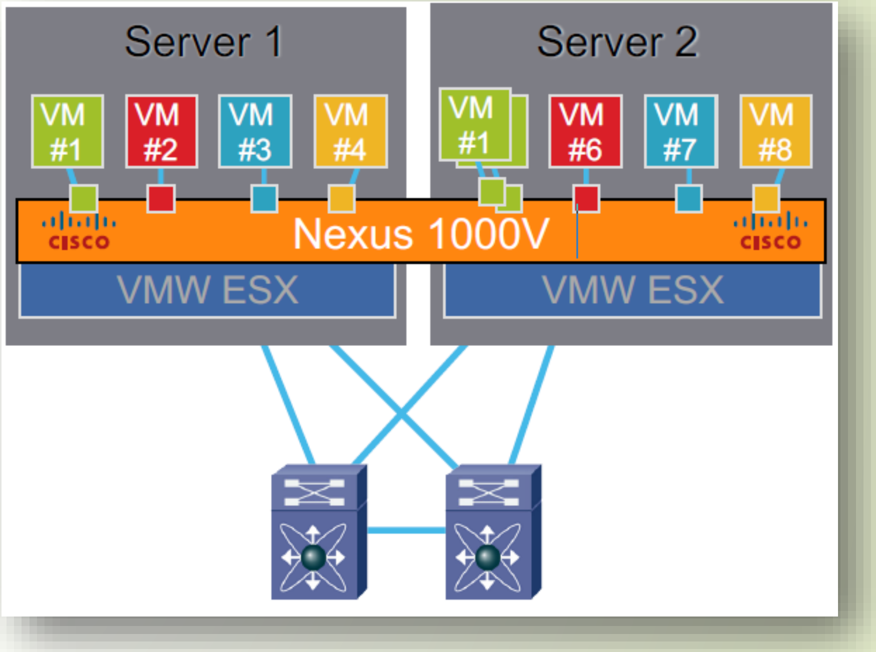
Agenda

- Objectives
- Role of Server Virtualization
- Role of Hypervisor
- Types of Virtualization
- Managing Virtual Machines Resources
- **Scaling up Virtual Machines**
- Configuring Virtual Networking in Virtual Machines
- Summary

Cisco Nexus 1000V

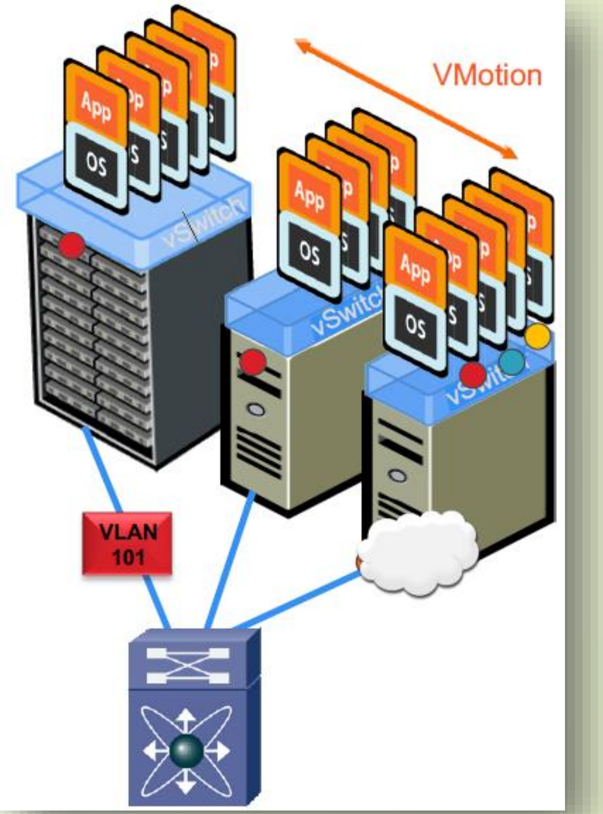
Industry First 3rd Party Virtual Distributed Switch

- **Nexus 1000V provides enhanced VM switching for VMW ESX environments**
- **Features VN-Link capabilities:**
  - Policy-based VM connectivity
  - Mobility of network and security properties
  - Non-disruptive operational model
- **Ensures visibility and continued connectivity during vMotion**
- **Enabling Acceleration of Server Virtualization Benefits**



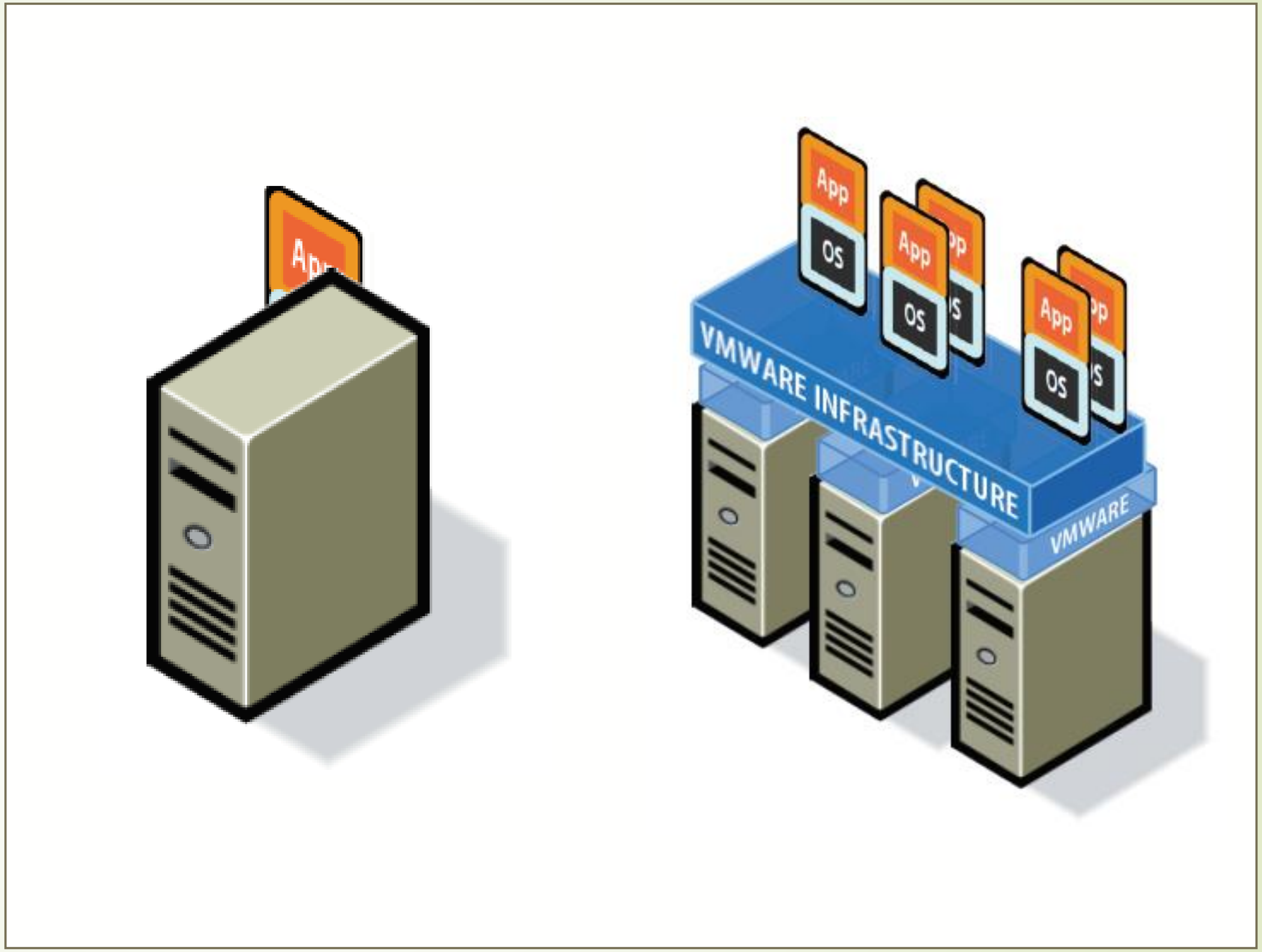
VN-Link Brings VM Level Granularity

- Problems:
  - VMotion may move VMs across physical ports—policy must follow
  - Impossible to view or apply policy to locally switched traffic
  - Cannot correlate traffic on physical links—from multiple VMs
- **VN-Link:**
  - Extends network to the VM
  - Consistent services
  - Coordinated, coherent management



Easy Migration from Physical to Virtual

- Consolidation Management with the VMware Infrastructure software will automate the migration from physical to virtual machines.



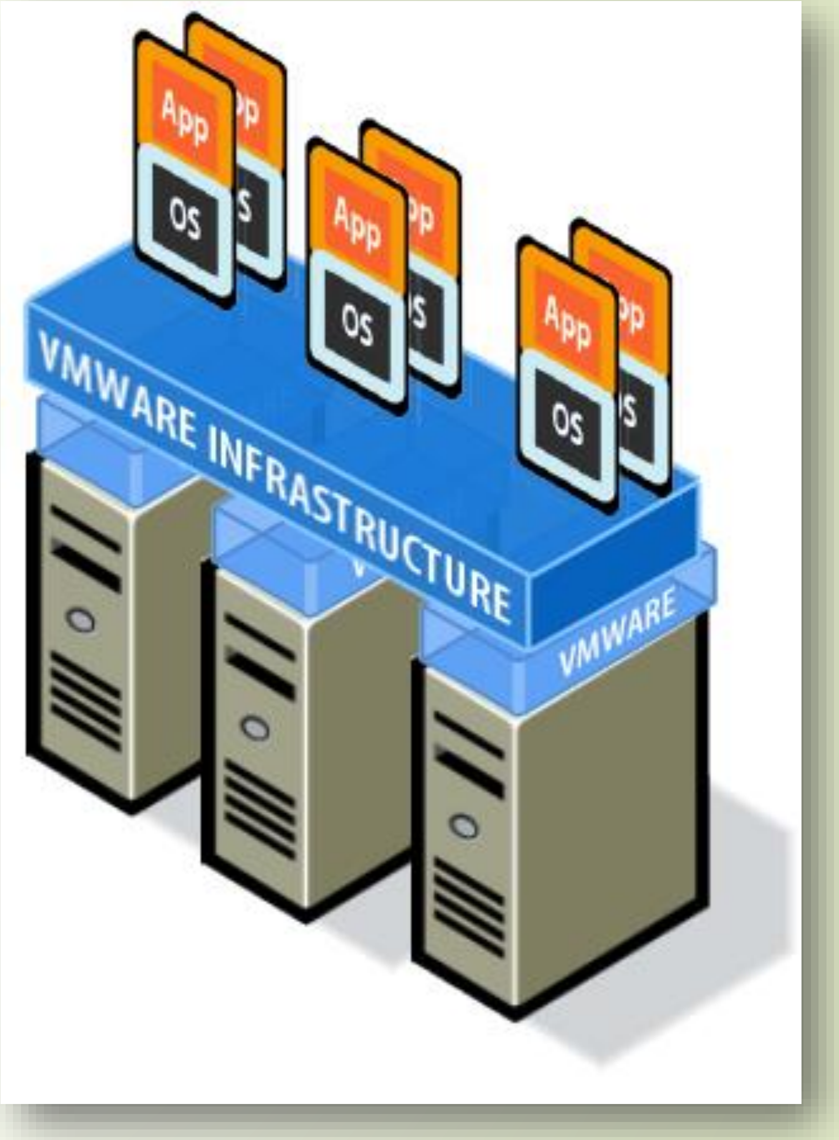
Easy Workflow Management for New Virtual Machines

- Provisioning with approval processes with Life Cycle Manager and Stage Manager

App
OS
OK
VMWARE INFRASTRUCTURE
VMWARE

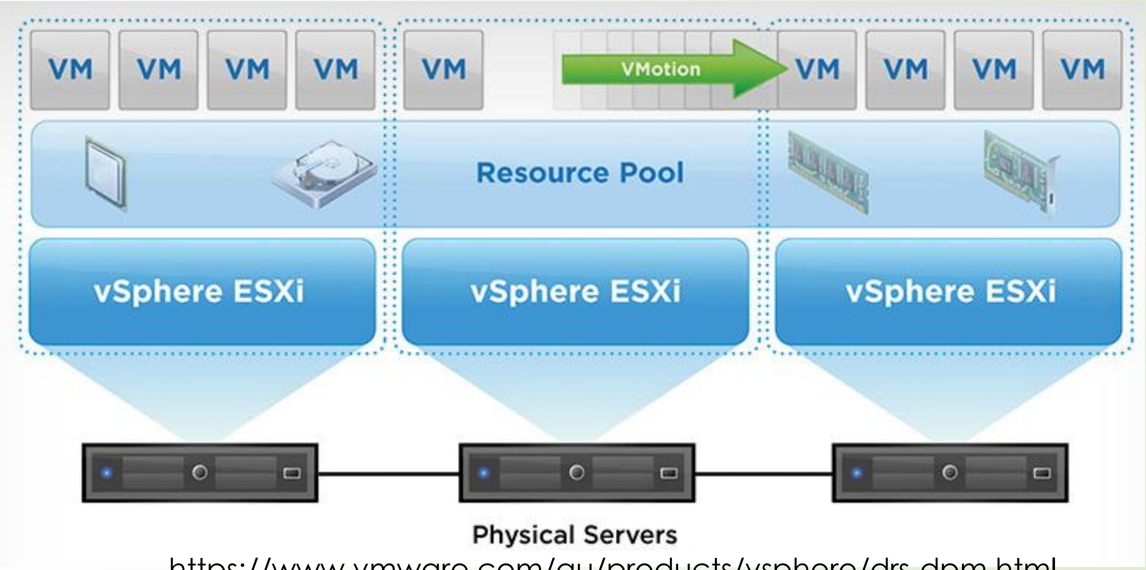
Scalability vServices

- VMware vMotion, makes it possible to move Virtual Machines, without interrupting the applications running inside.



Automatic Scalability vServices

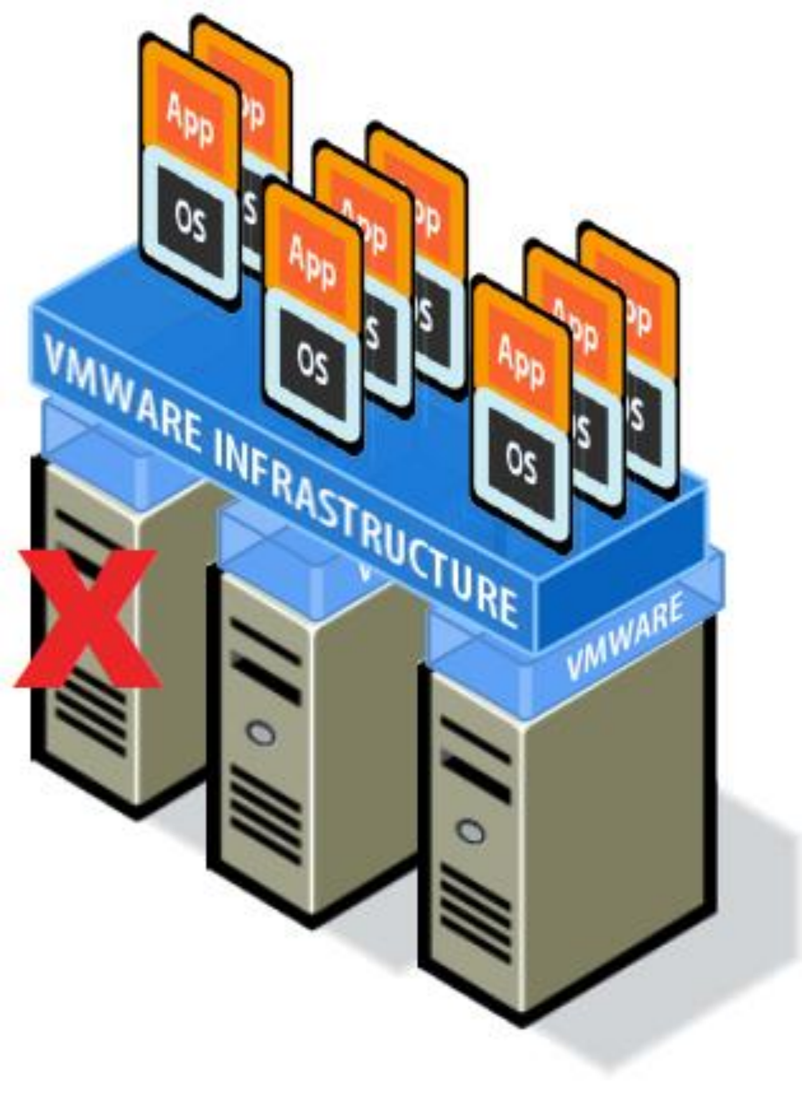
- VMware Distributed Resource Scheduler automatically balances the Workloads according to set limits and guarantees, removing the need to predict resource assignment.



https://www.vmware.com/au/products/vsphere/drs-dpm.html

High Availability vServices

- VMware High Availability makes all Servers and Applications protected against component and complete system failure.
- Only One-Click to configure!

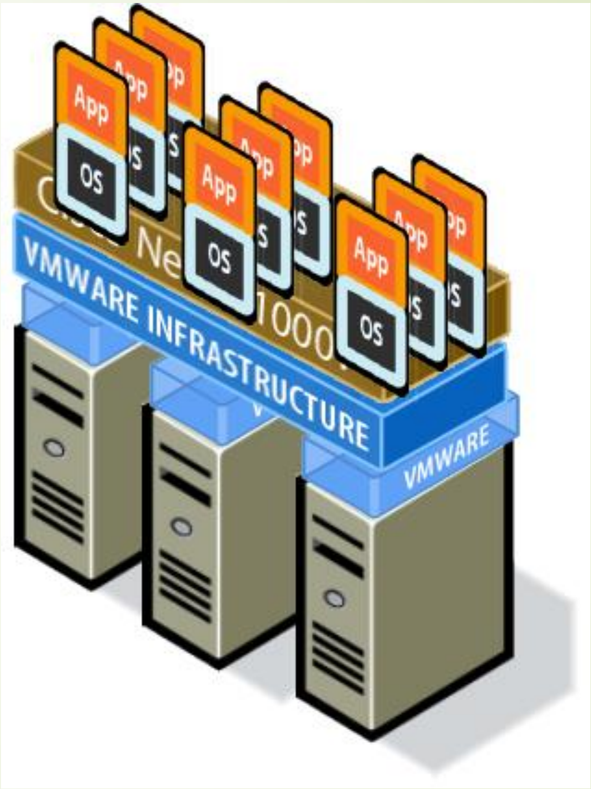


Disaster Recovery vServices

- VMware Site Recovery Manager enables an easy transition from a production site to a Disaster Recovery site.
- Easy Execution for real Disaster
- Easy Testing for good night sleep

Adding Infrastructure vServices

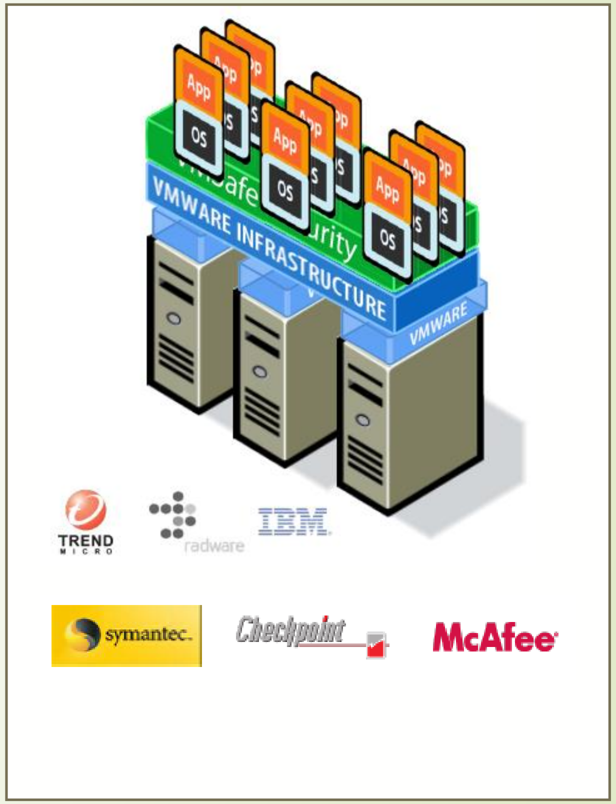
- VMware and Cisco are collaborating to enhance workload mobility and simpler management with virtualization-aware networks



Nexus 1000V

Security vServices

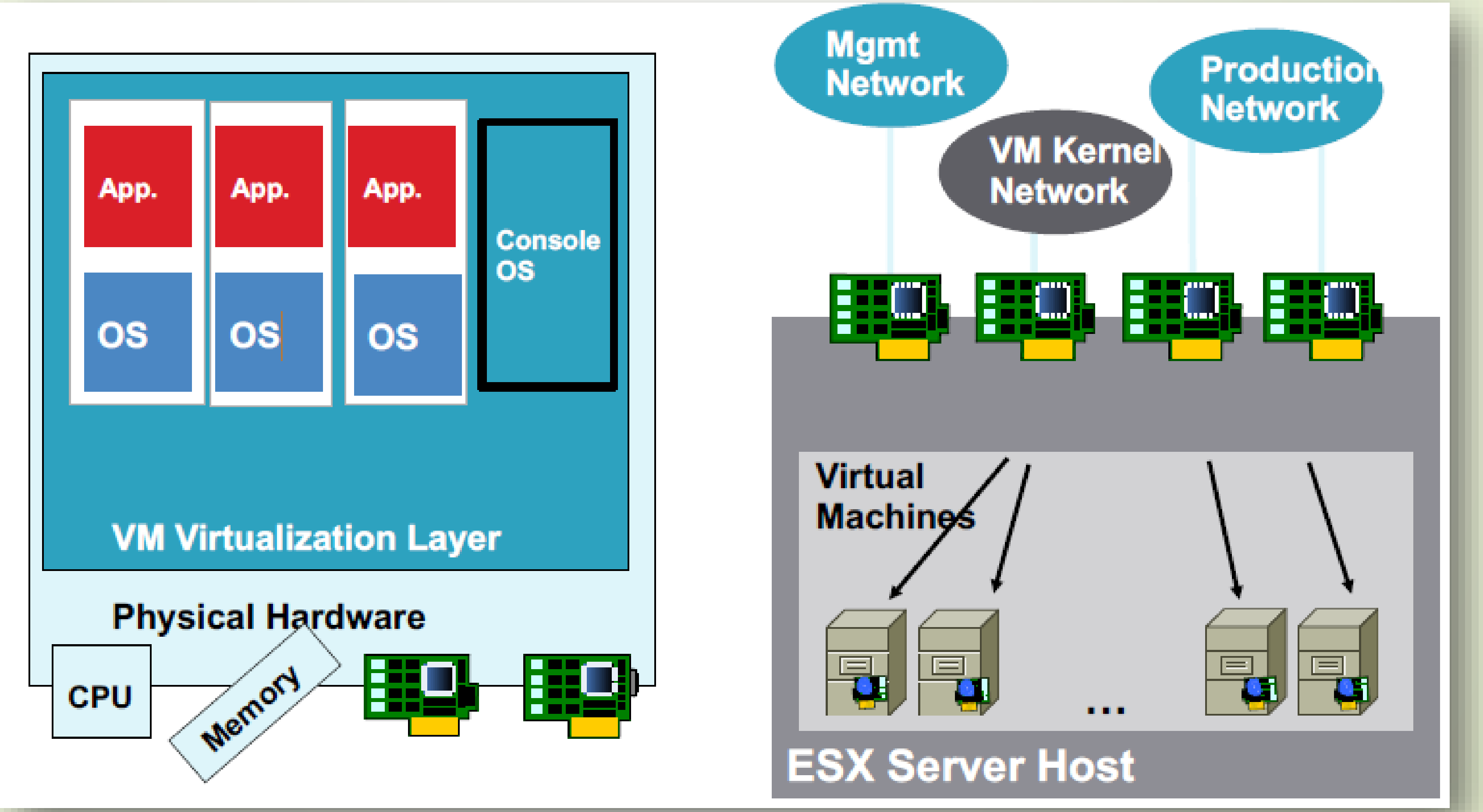
- The Application vService VMSafe allows security vendors to add superior security solutions inside the VMware Infrastructure.



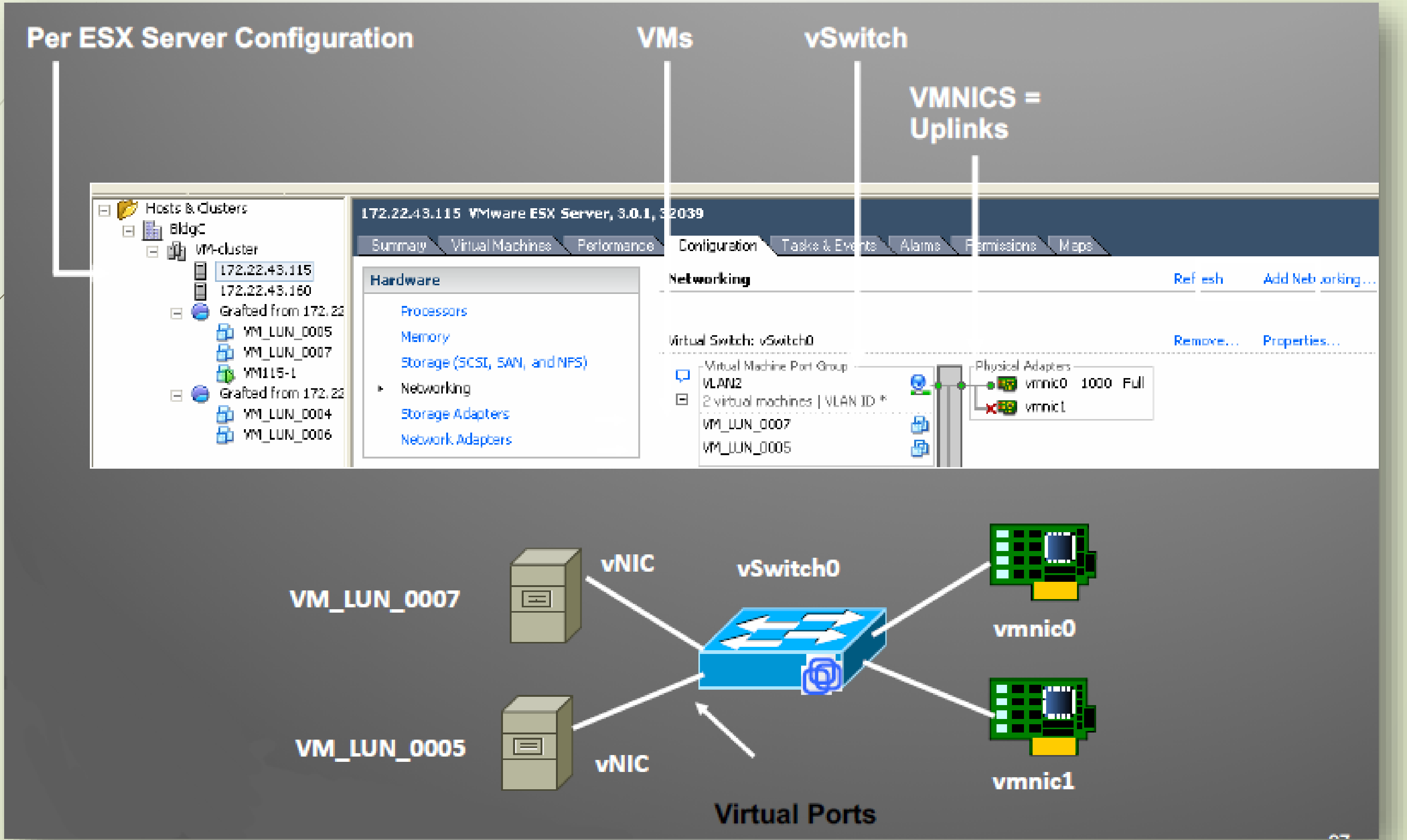
Agenda

- Objectives
- Role of Server Virtualization
- Role of Hypervisor
- Types of Virtualization
- Managing Virtual Machines Resources
- Scaling up Virtual Machines
- **Configuring Virtual Networking in Virtual Machines**
- Summary

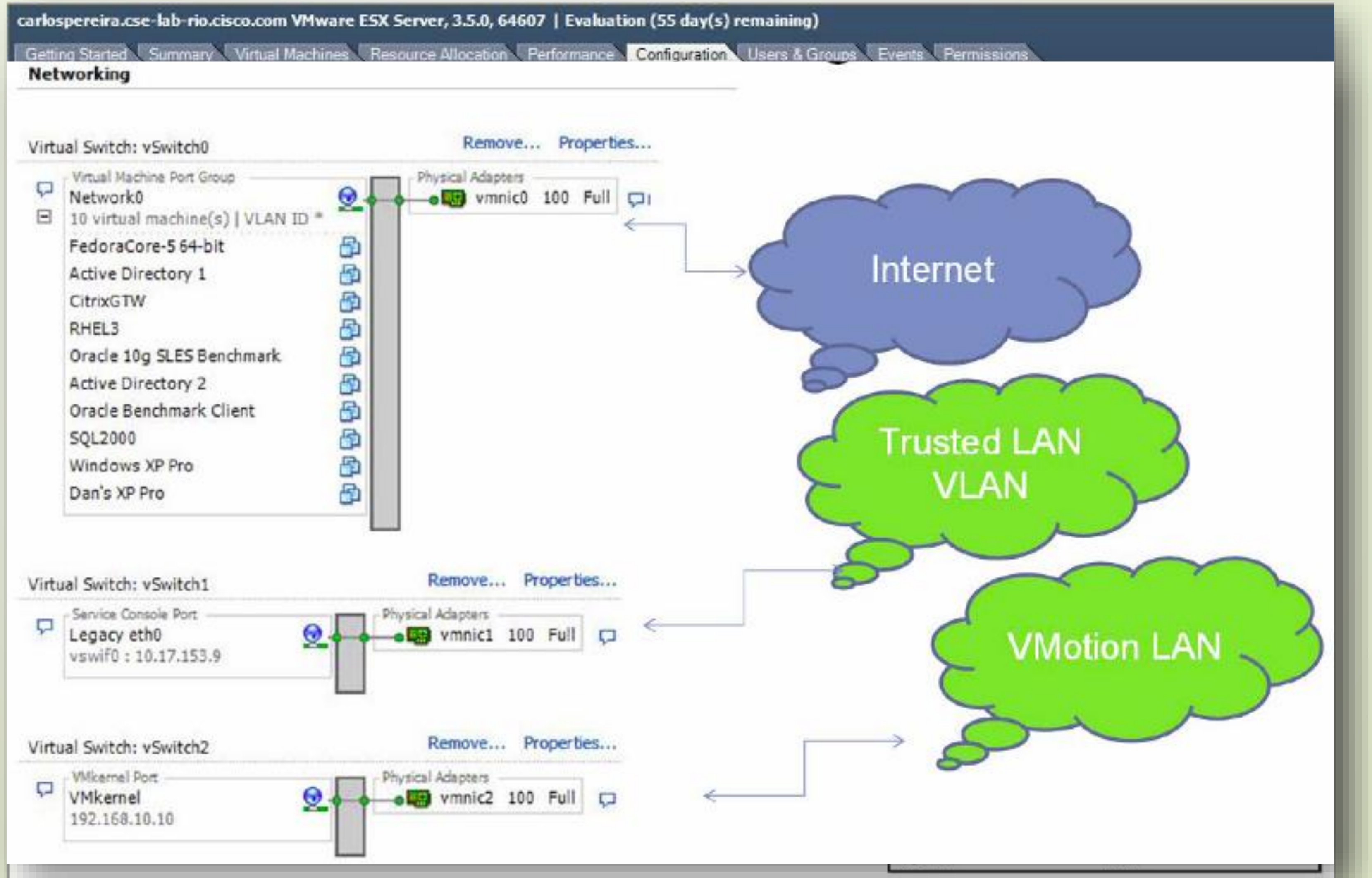
VMware ESX Network Architecture in a Nutshell



VMware Networking Components



VMware Networking Components



carlospereira.cse-lab-rio.cisco.com VMware ESX Server, 3.5.0, 64607 | Evaluation (55 day(s) remaining)

Networking

Virtual Switch: vSwitch0

Virtual Machine Port Group
Network0
10 virtual machine(s) | VLAN ID *
FedoraCore-5 64-bit
Active Directory 1
CitrixGTW
RHEL3
Oracle 10g SLES Benchmark
Active Directory 2
Oracle Benchmark Client
SQL2000
Windows XP Pro
Dan's XP Pro

Physical Adapters
vmnic0 100 Full

Virtual Switch: vSwitch1

Service Console Port
Legacy eth0
vswif0 : 10.17.153.9

Physical Adapters
vmnic1 100 Full

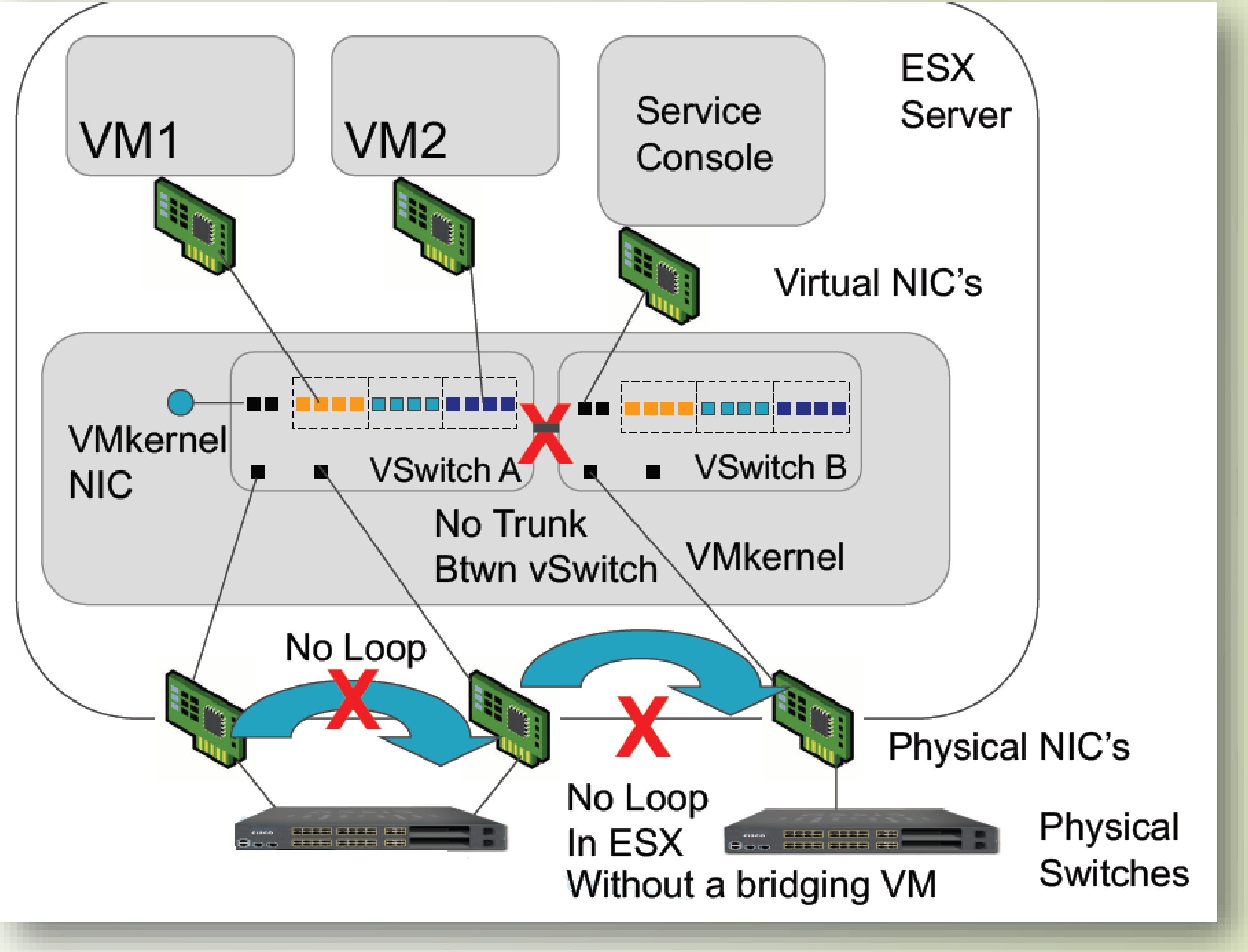
Virtual Switch: vSwitch2

VMkernel Port
VMkernel
192.168.10.10

Physical Adapters
vmnic2 100 Full

vSwitch Overview

- **Software implementation of an Ethernet switch within ESX**
- **How is it like a switch:**
  - MAC addr forwarding VLAN segmentation
- **How is it different:**
  - No need to learn MAC addresses – it knows the address of the connecting NIC's
  - No participation in spanning tree



vSwitch Forwarding Characteristics

- Forwarding based on MAC address (no learning): If traffic doesn't match a VM MAC is sent out to vmnic
- VM-to-VM traffic stays local
- Vswitches TAG traffic with 802.1q VLAN ID
- vSwitches are 802.1q-capable
- vSwitches can create EtherChannels

Resources

- http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf
- http://www.vmware.com/pdf/RVI_performance.pdf
- http://www.vmware.com/pdf/Perf_Best_Practices_vSphere4.1.pdf
- Cisco Server Virtualization with Vmware http://www.cisco.com/en/US/netsol/ns1148/index.html