



# The Future of x86 Virtualization

Leendert van Doorn  
Sr. Fellow



# Outline

## Introduction

- What is virtualization?

## The past

- Trends that drove virtualization
- Virtualization technology back then: CPU virtualization

## The present

- Trends that are driving virtualization
- Virtualization technology today: CPU virtualization (SVM), Nested Paging, tagged TLBs, ...

## The future

- Trends that will drive virtualization
- Virtualization technology tomorrow: I/O virtualization, security, acceleration widgets, nested virtualization, high-availability ...

## Summary



# Introduction



# Virtualization

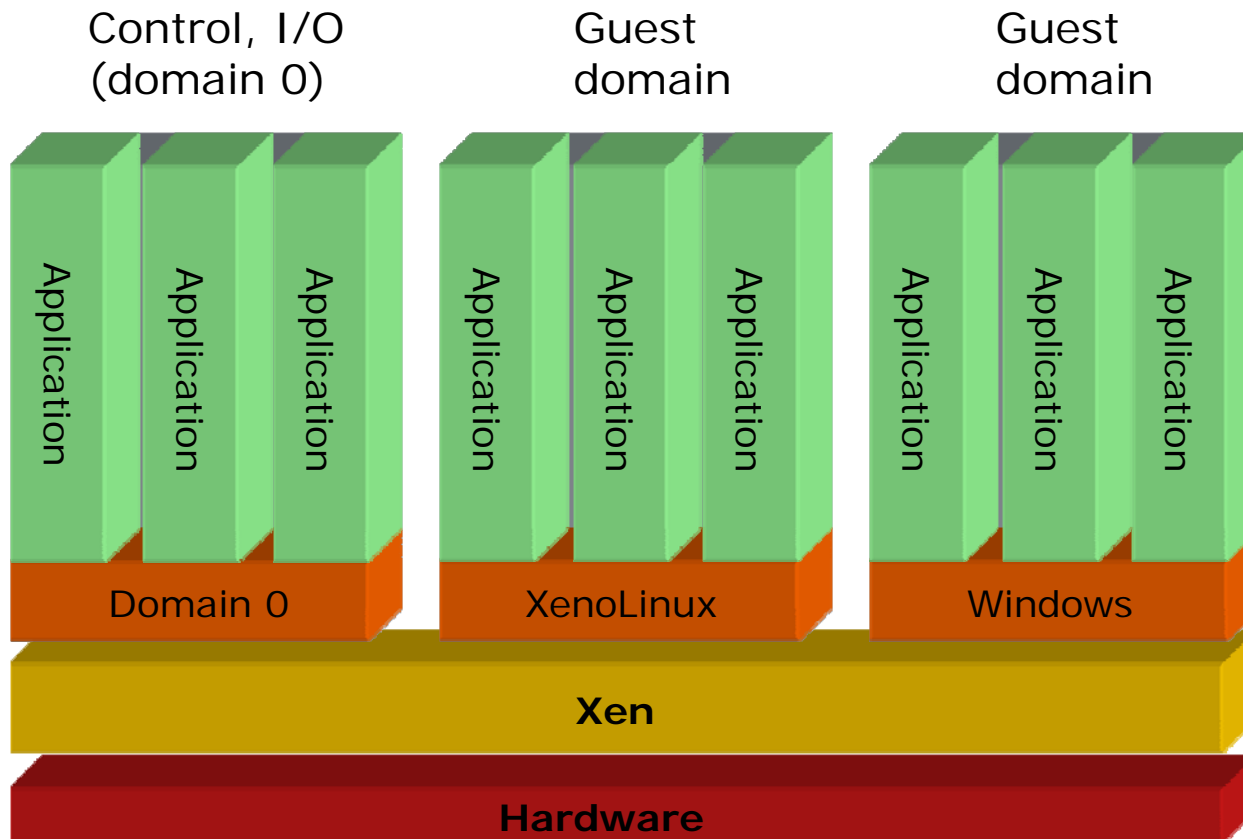
Multiple consumers share a resource while maintaining the illusion that each consumer owns the full resource

- Memory, processor(s), storage, peripherals, entire machines

Virtual Machine Monitor (VMM) or **hypervisor** is the software layer that provides one or more Virtual Machine (VM) abstractions



# Example: Xen System Architecture



**A virtual machine monitor (VMM) is a micro kernel with a slightly confused process model**



# The Past



# IBM 370 Mainframe

Virtualization has a long history

IBM's 370 architecture was self virtualizable

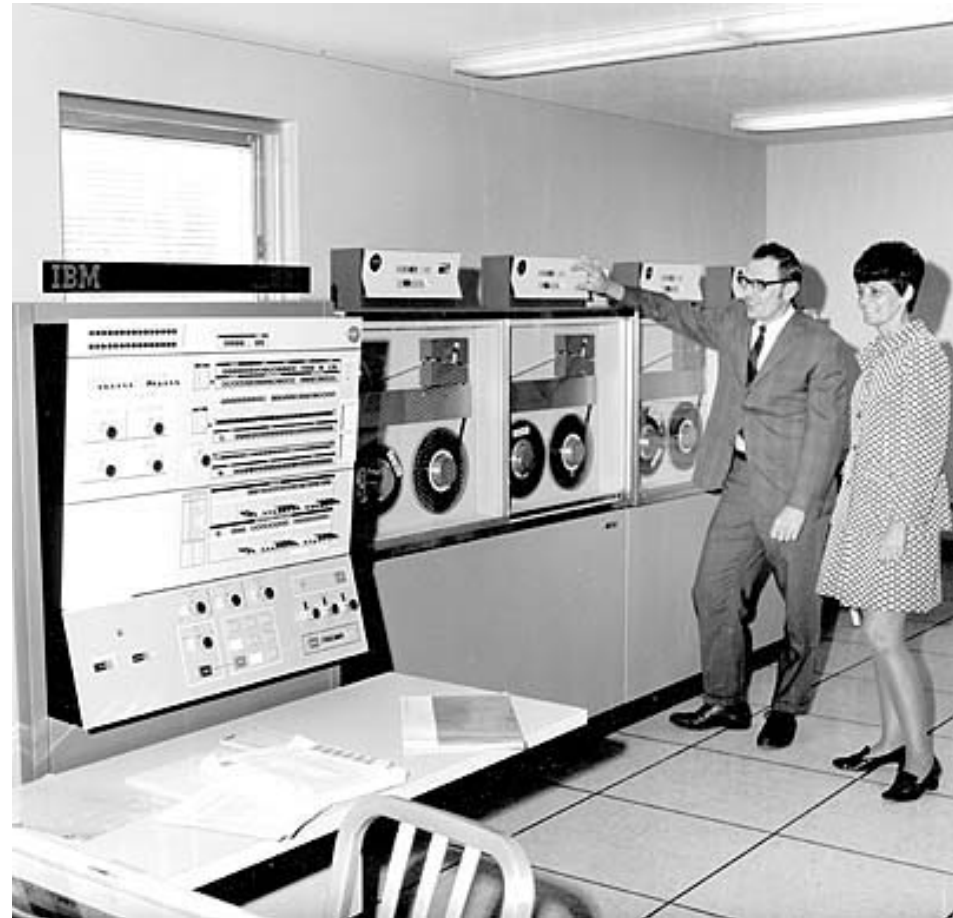
Trap and emulate model

*Popek and Goldberg [1974]*

IBM introduced the SIE instruction in 1980s in their 370/XA architecture

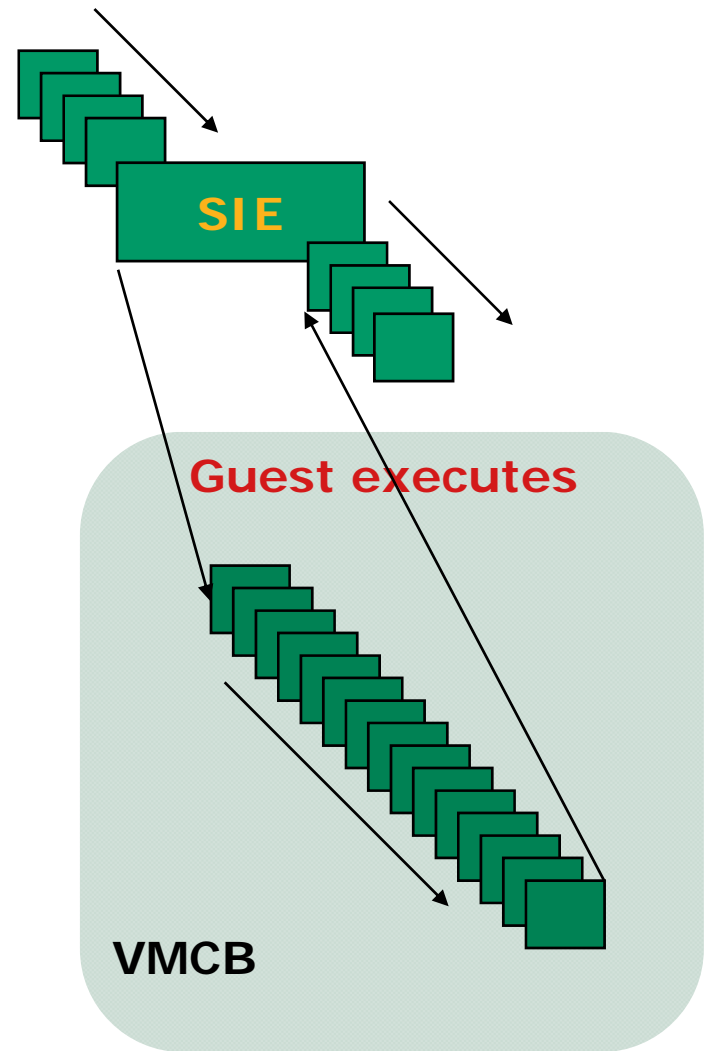
Enabled nested VMs

I/O channels



# Start Interpretive Execution (SIE)

- Virtualization based on **SIE** instruction
- SIE executed by host causes the guest to run
- Guest runs until it exits back to the host
- Host resumes at the instruction following SIE
- World-switch: host → guest → host



Today



# Trends that are driving virtualization

## Reduce total cost of ownership (TCO)

- Increased systems utilization (current servers have less than 10% average utilization, less than 50% peak utilization)
- Reduce hardware (25% of the TCO)
- Space, electricity, cooling (50% of the operating cost of a data center)

## Management simplification

- Dynamic provisioning
- Workload management/isolation
- Virtual machine migration
- Reconfiguration

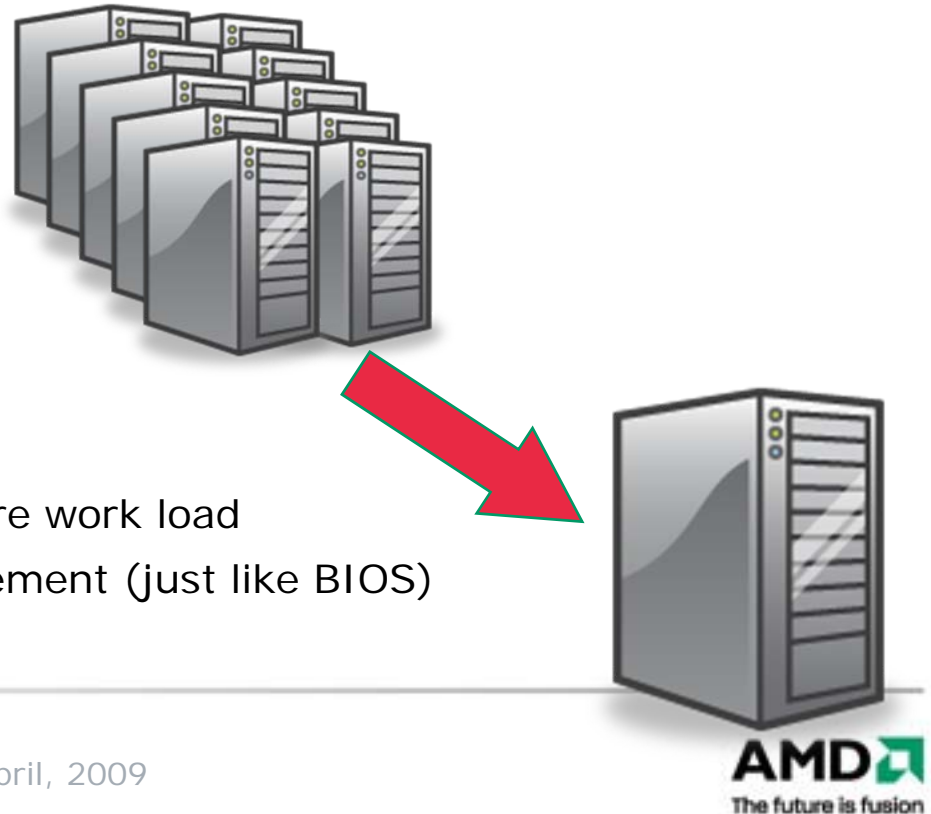
## Better security

## Legacy compatibility

## Virtualization protects IT investment

## Virtualization is a true scalable multi-core work load

## Virtualization is becoming a platform element (just like BIOS)



# Processor Virtualization Features

Both AMD and Intel defined processor extensions for their CPU architectures

AMD: Secure Virtual Machine (SVM, AMD-V), Rev F, Rev G, Barcelona, Shanghai ...

Intel: Vanderpool Technology (VT-x, VT-x2, ...)

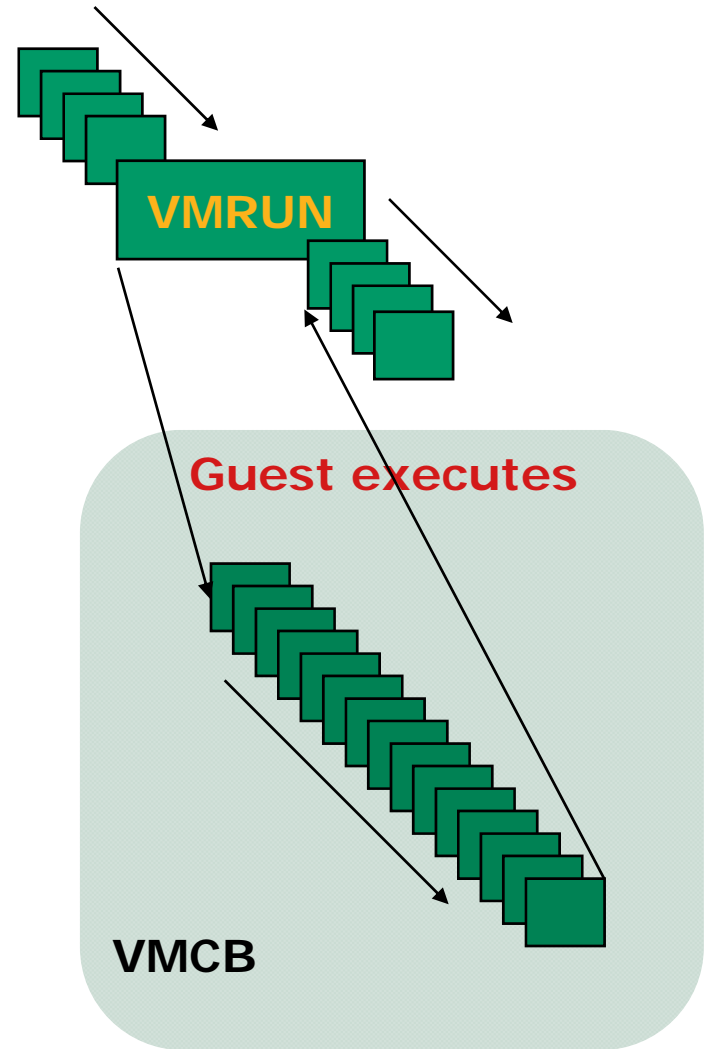
From 10,000 ft. both look very similar

- Container model (similar to mainframe SIE)
- Contrast this with Power5 RISC approach



# AMD SVM / Intel VT-x In A Nutshell

- Virtualization based on **VMRUN/VMRESUME** instruction
- VMRUN/VMRESUME executed by host causes the guest to run
- Guest runs until it exits back to the host
- Host resumes at the instruction following VMRUN/VMRESUME
- World-switch: host → guest → host
- World switches are not cheap



# Intercepts and Exits

A guest runs until

- it performs an action that causes an exit
- it executes a VMCALL/VMMCALL

Exit conditions are specified per guest

- Exceptions (e.g., page faults) and interrupts
- Instruction intercepts (CLTS, HLT, IN, OUT, INVLPG, MONITOR, MOV CR/DR, MWAIT, PAUSE, RDTSC ...)

AMD-V has paged 16-bit real-mode support

Intel VT-x has shadow registers



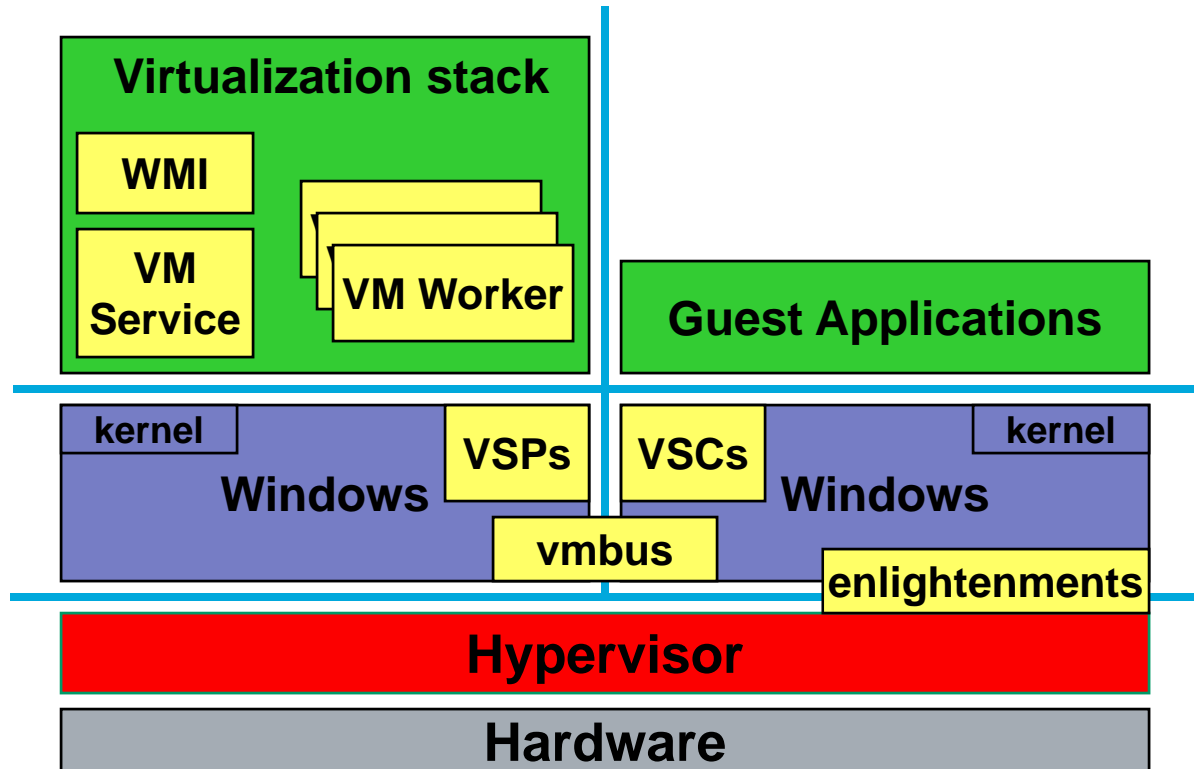
# CPU Virtualization Techniques Comparison

	Performance	Legacy guest support	VMM complexity
Binary rewriting	medium	yes	high
paravirtualization	high	no	medium
Hardware assist (first gen)	low	yes	medium-low
Hardware assist (current gen)	medium	yes	medium-low
Future hardware assist	high	yes	low

low    medium    high  




# Typical Virtualization Software Stack Microsoft Hyper-V (a.k.a. Viridian)



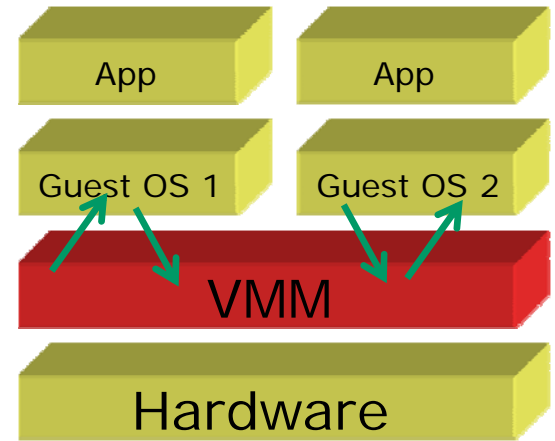
Hyper-V runs Windows and Linux guests

Uses AMD SVM, Intel VT-x and paravirtualization (enlightenments)

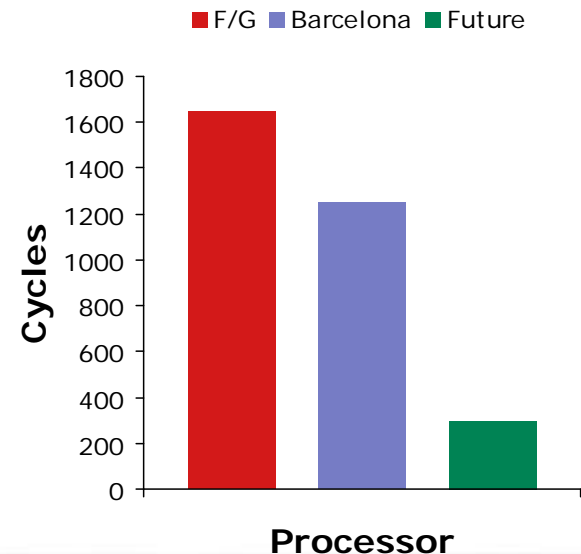


# CPU Virtualization Trends

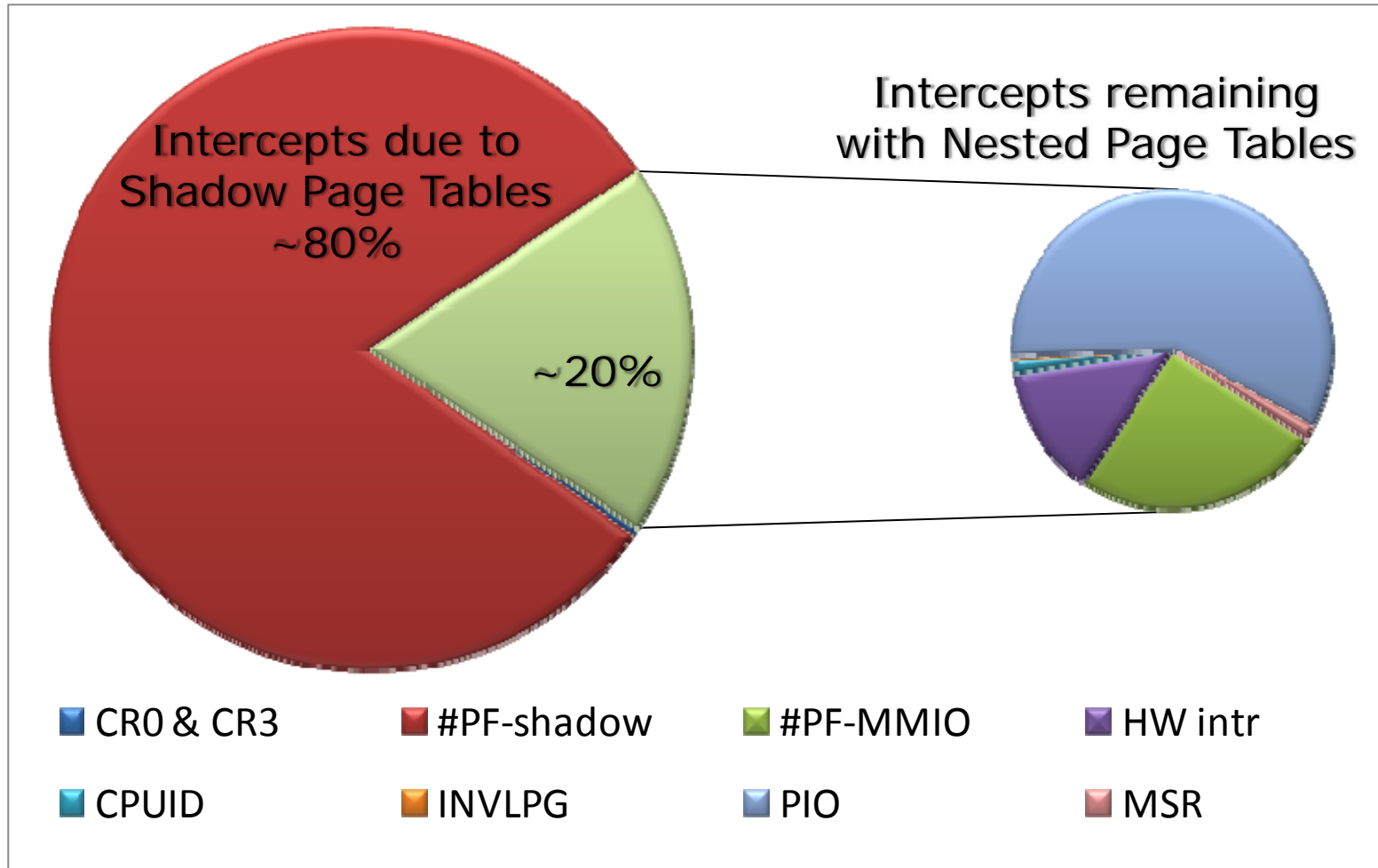
- The key trend is to eliminate the overhead of virtualization
  - Reduce overall world-switch times
  - Reduce world-switch frequencies
  - Reduce resources needed by the VMM
- Reduce world-switch times
  - Tag TLB by ASID
  - Better caching of VMCB state
- Reduce world-switch frequencies
  - Nested paging
  - Direct device assignment
  - Implement more functions in the guest OS through paravirtualization



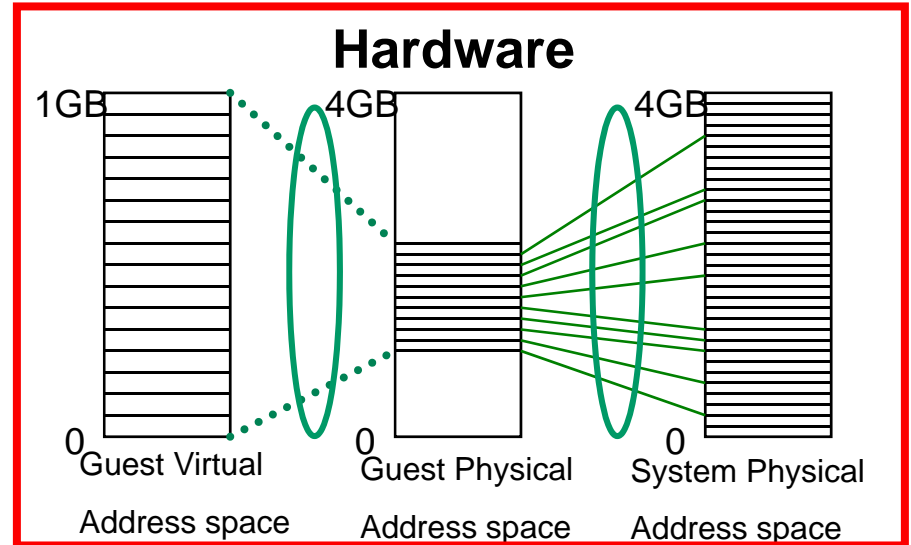
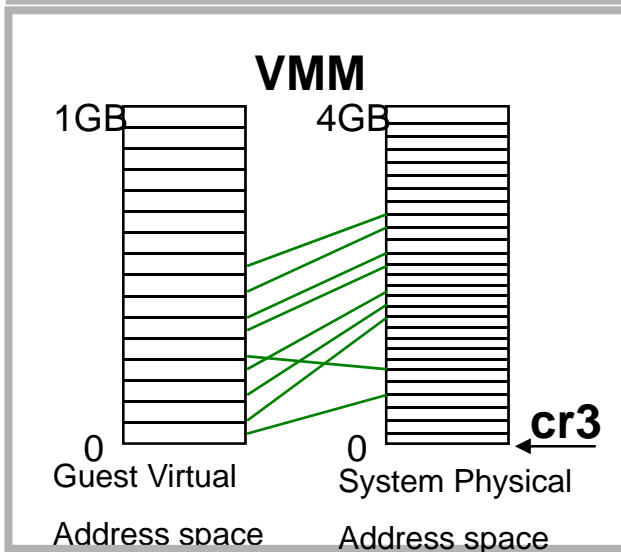
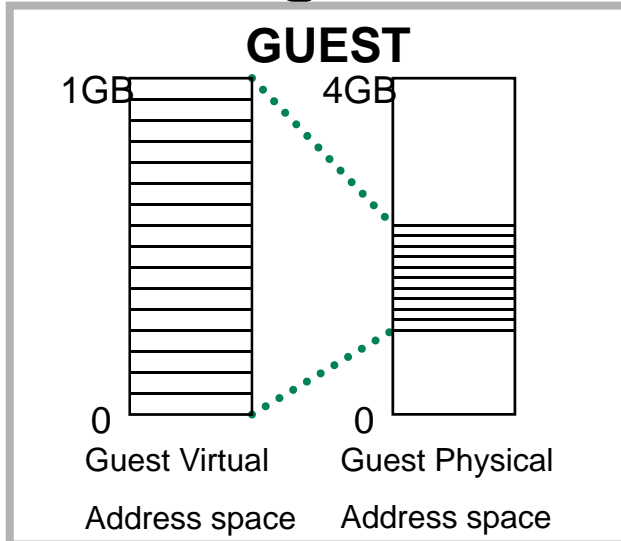
VM World-switch Times



# Where Do the Intercepts Go?



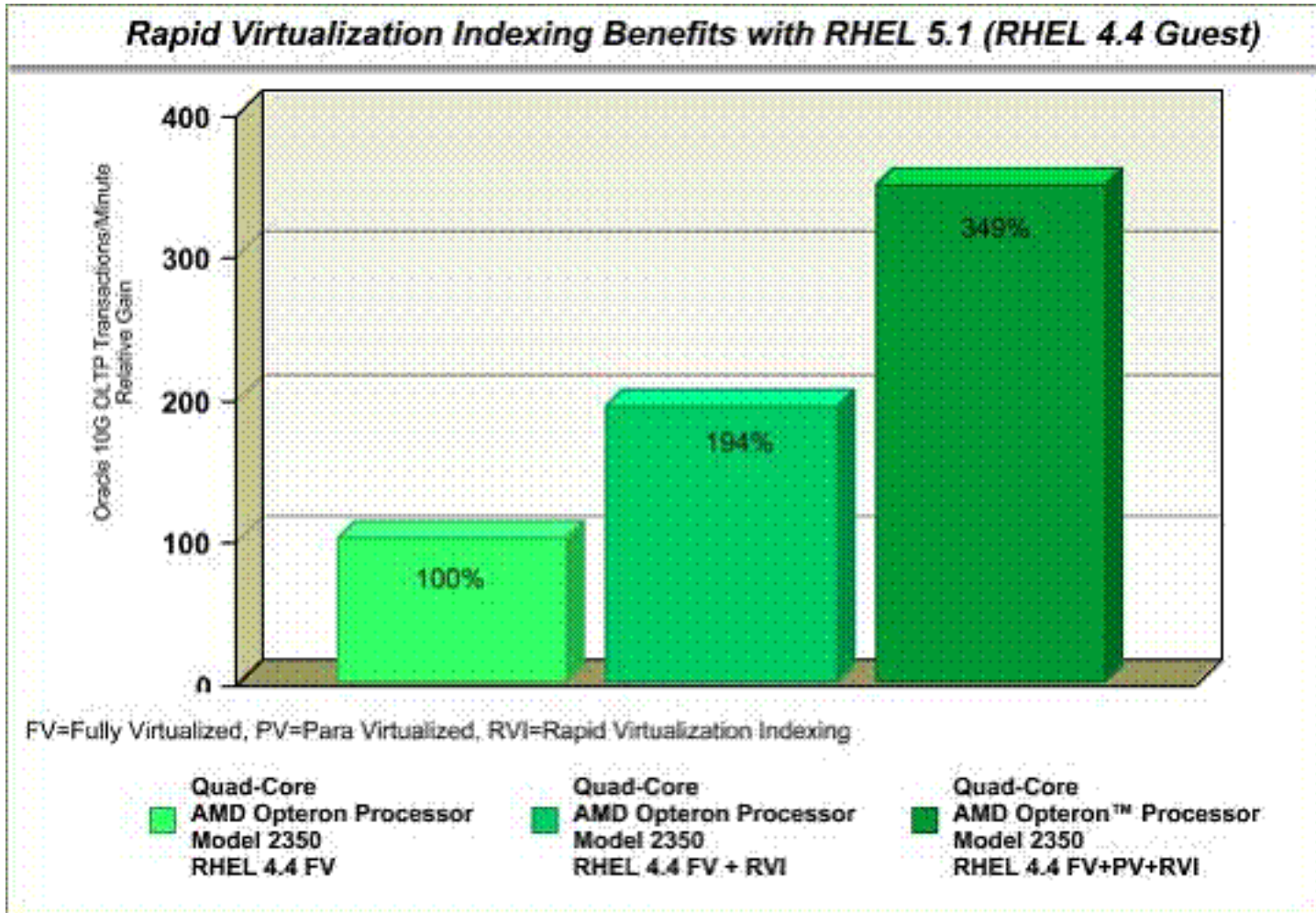
# Nested Page Tables



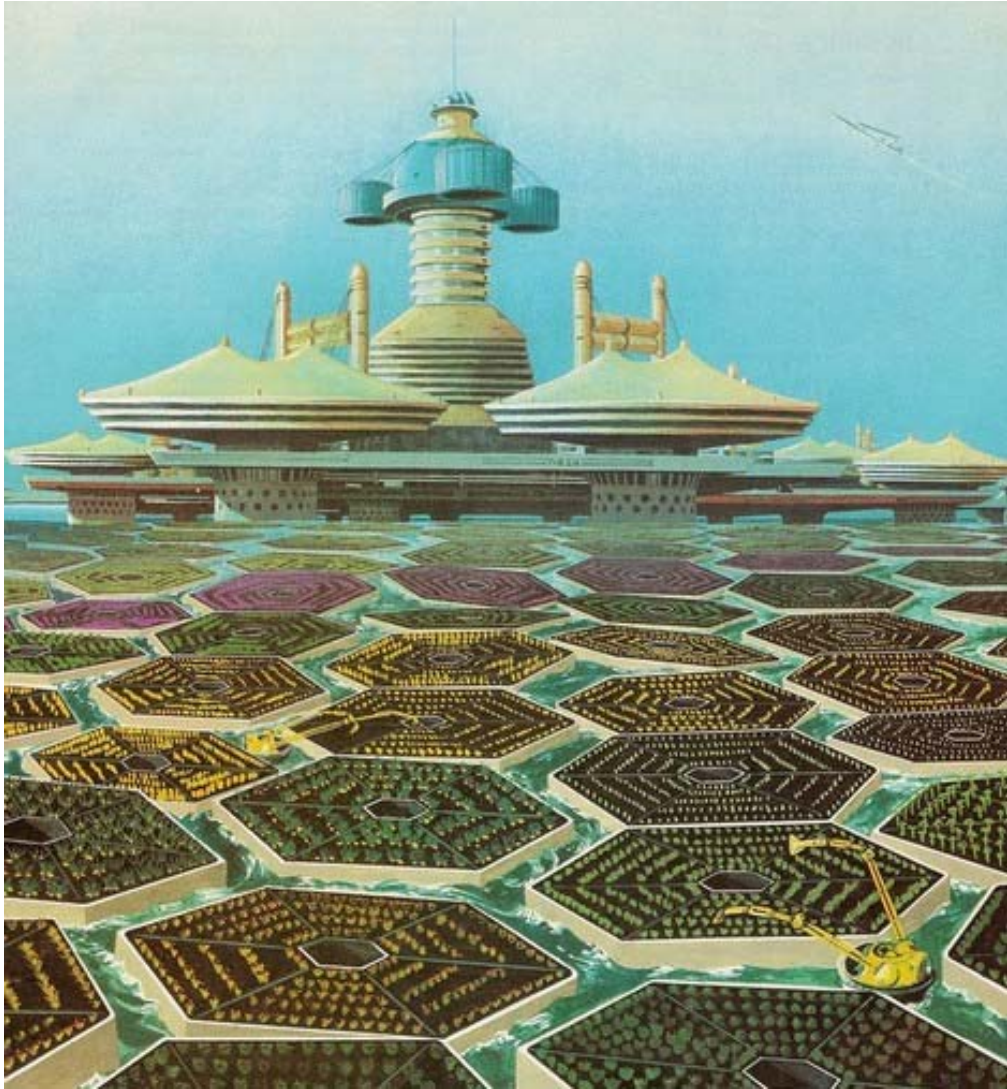
- Traditionally the hypervisor maintains shadow page tables:
  - Expensive to emulate correct behavior (accessed/modified bits)
- Nested paging eliminates this by performing a recursive walk
  - Available in Barcelona ...
  - Reduces number of #VMEXITs by 40-70%



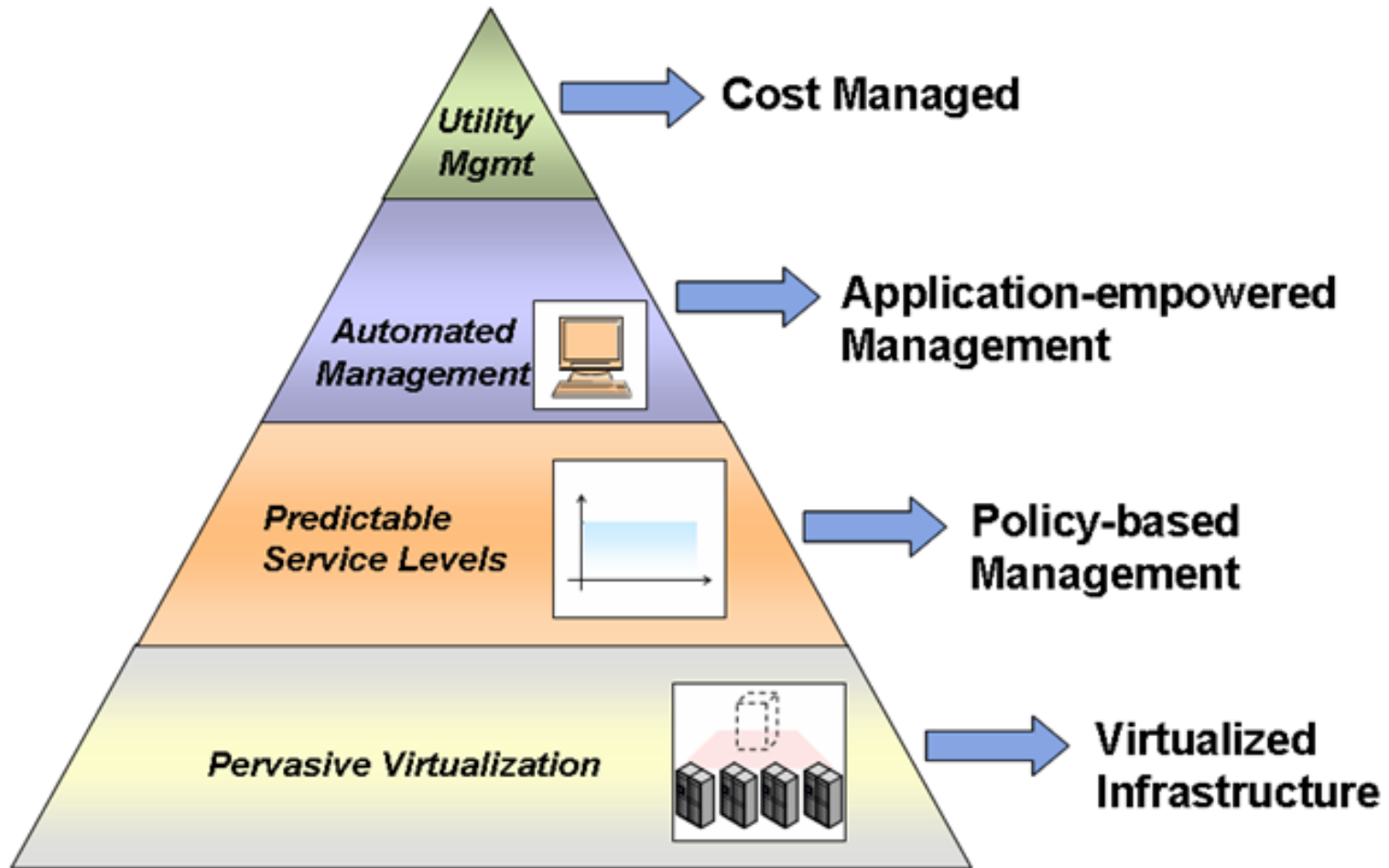
# Nested Page Table Performance



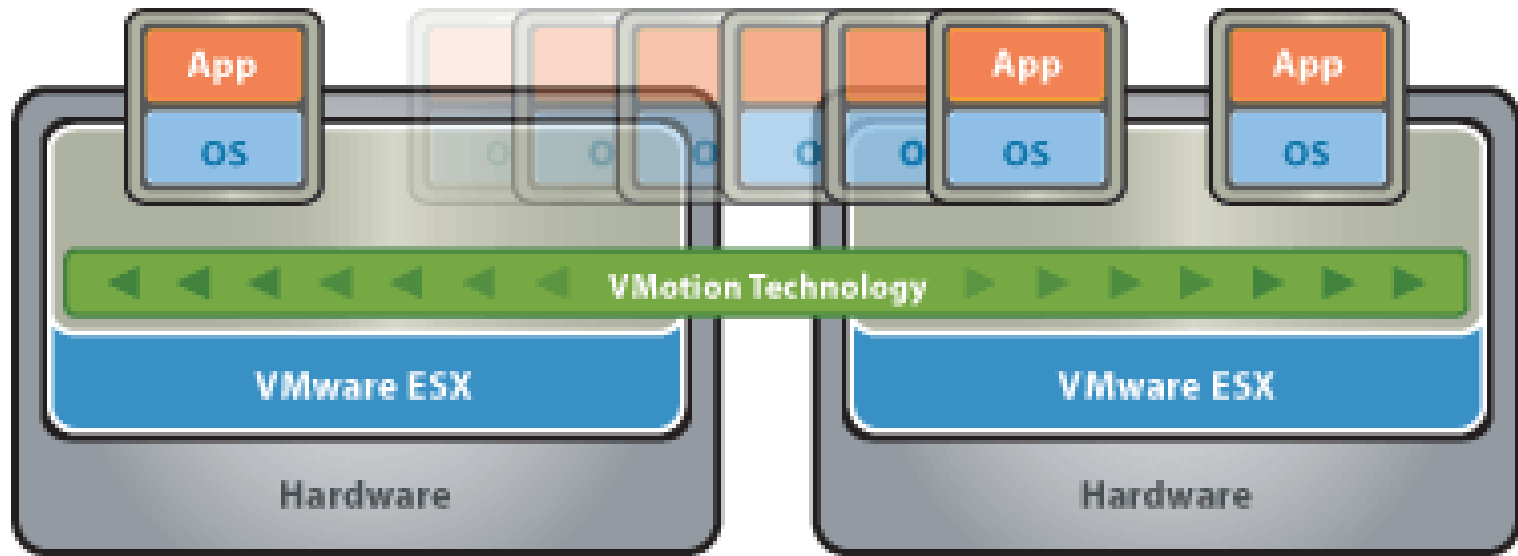
# The Future



# The Dynamic Data Center



# Live Virtual Machine Migration



Live migration of VM is the ability to move **running** VMs from one machine to another

Compare this with checkpoint/restart

Vmware introduced Vmotion (live migration) in ESX server 3.0 (2005)

Hyper-V (MS, 2008) has checkpoint/restart in its first version

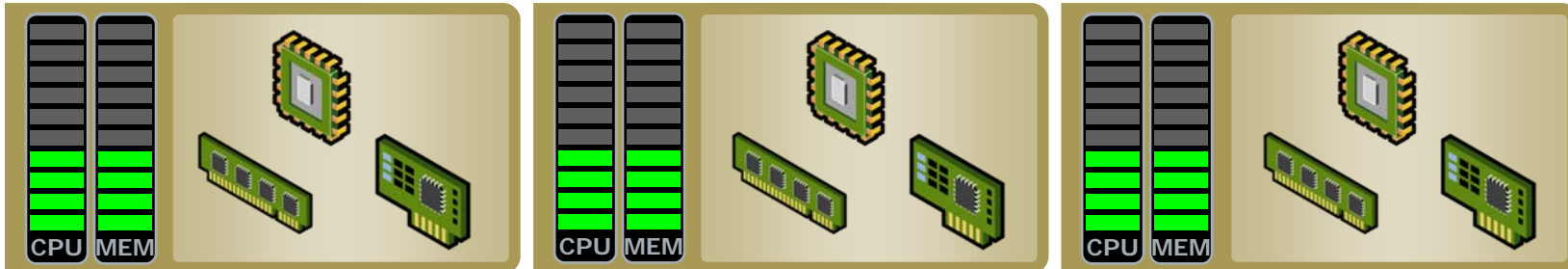
Enabling the dynamic datacenter



# VMWare VMotion: Load Balancing



*Blade Chassis*



**Blade #1**

**Blade #2**

**Blade #3**



# VM Migration Challenges

Most hypervisors allows inter-vendor migration of certain processor types

- AMD: Rev E (2005) ... Istanbul (2009)
- Define common denominator platform to deal with new and old instructions

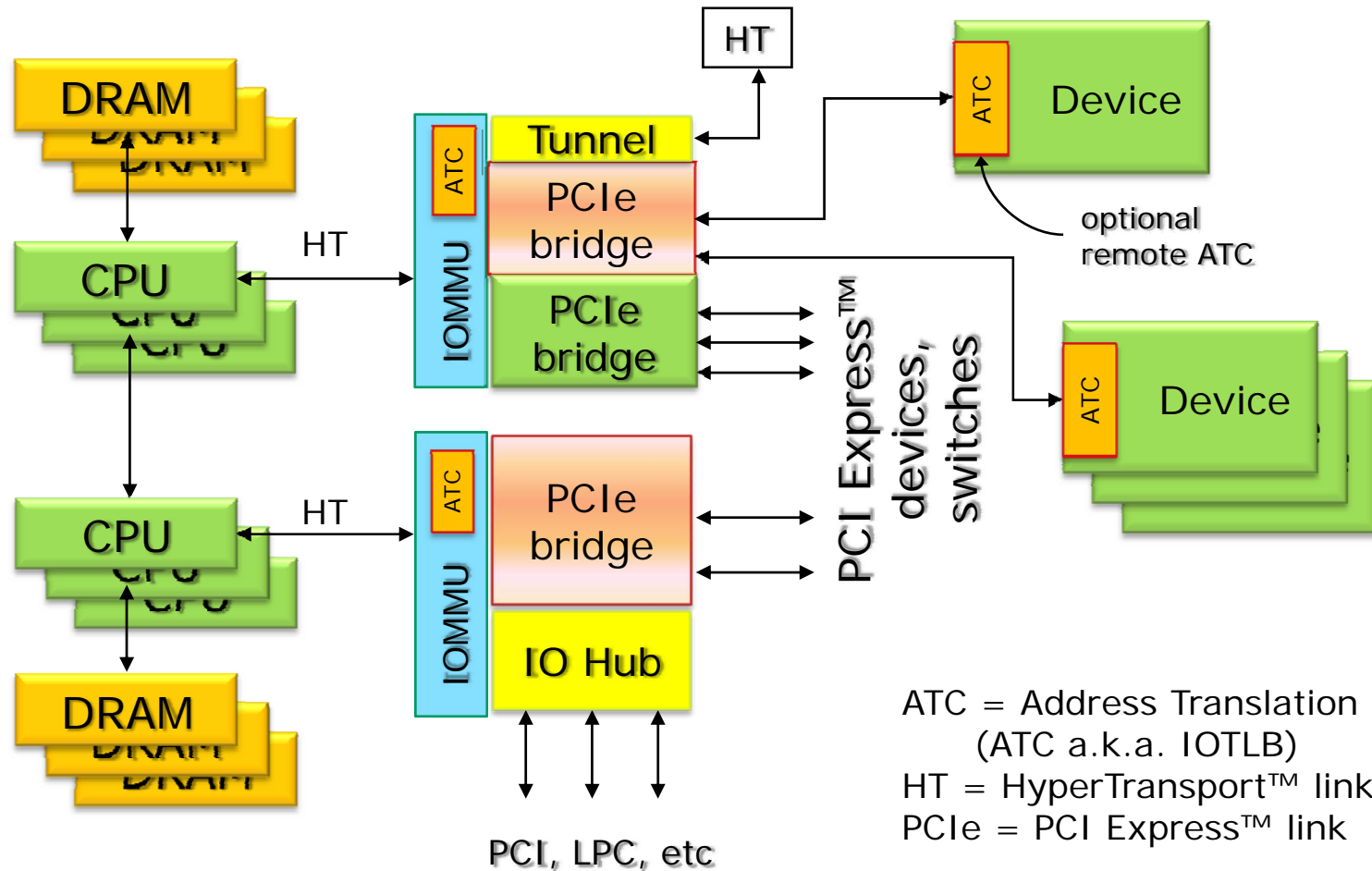
Next challenge is cross vendor migration

- Intel to AMD and visa versa
- Instruction sets are not fully compatible (syscall), floating point significance

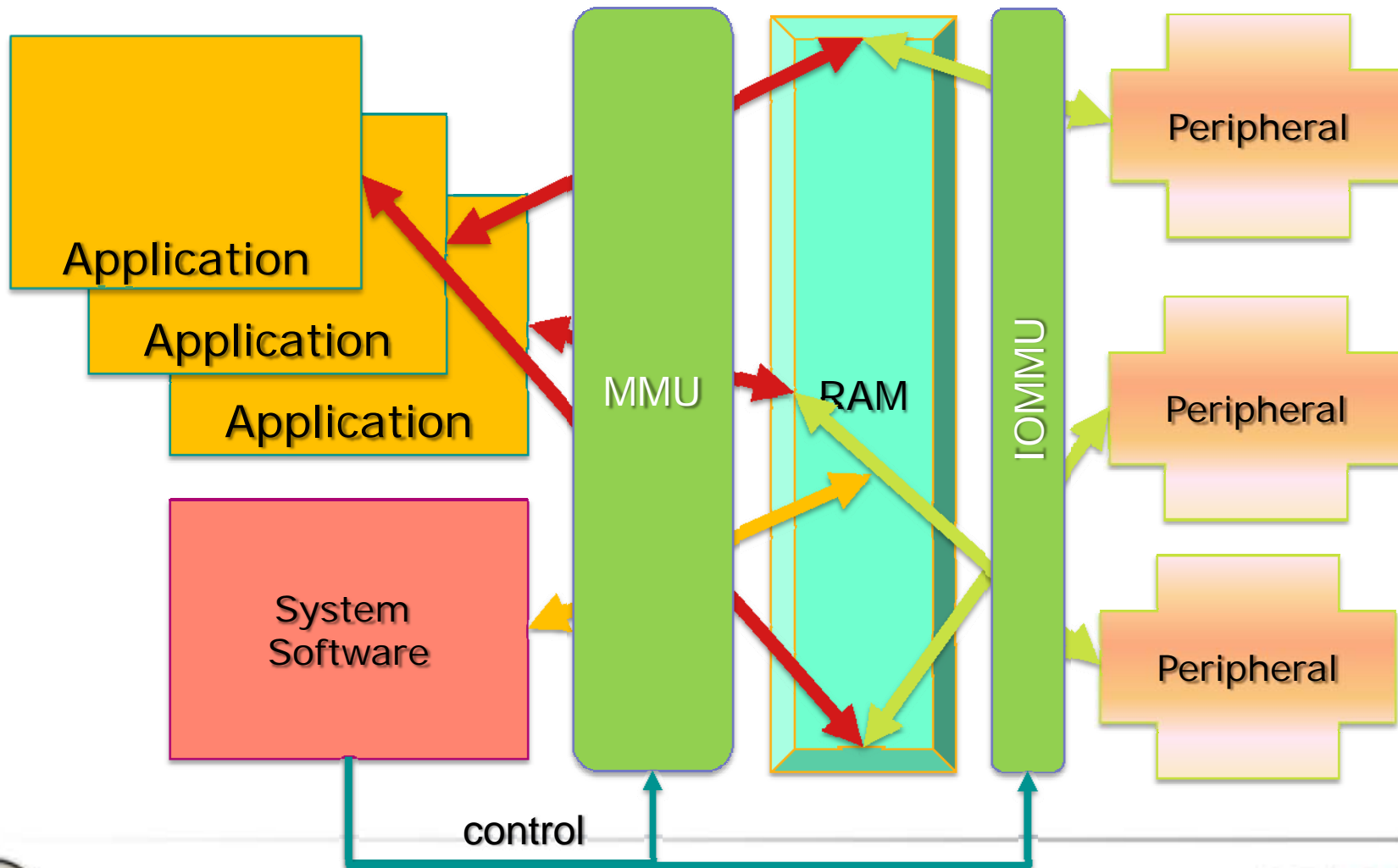
I/O virtualization breaks migration



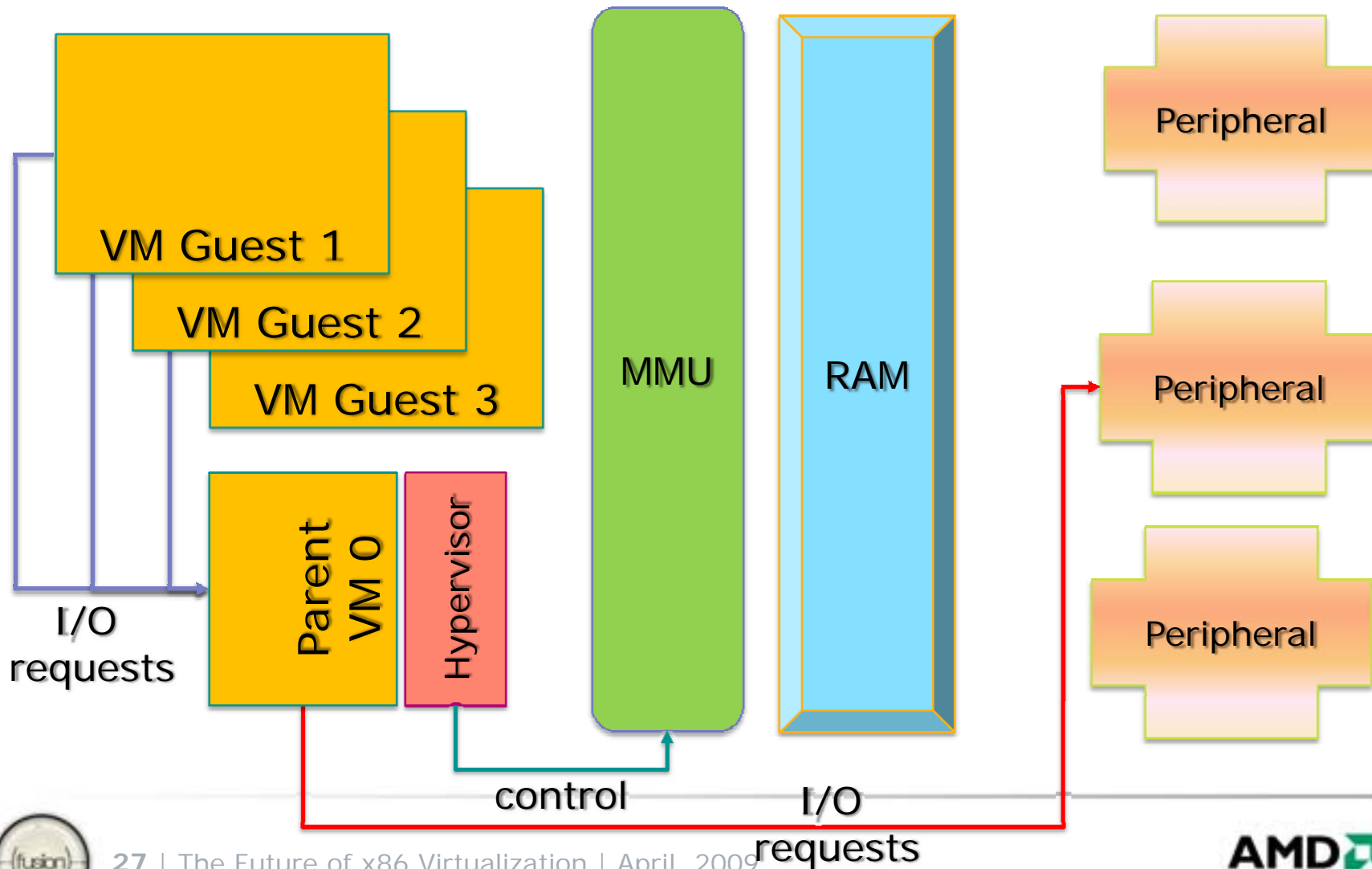
# Virtualizing The Platform



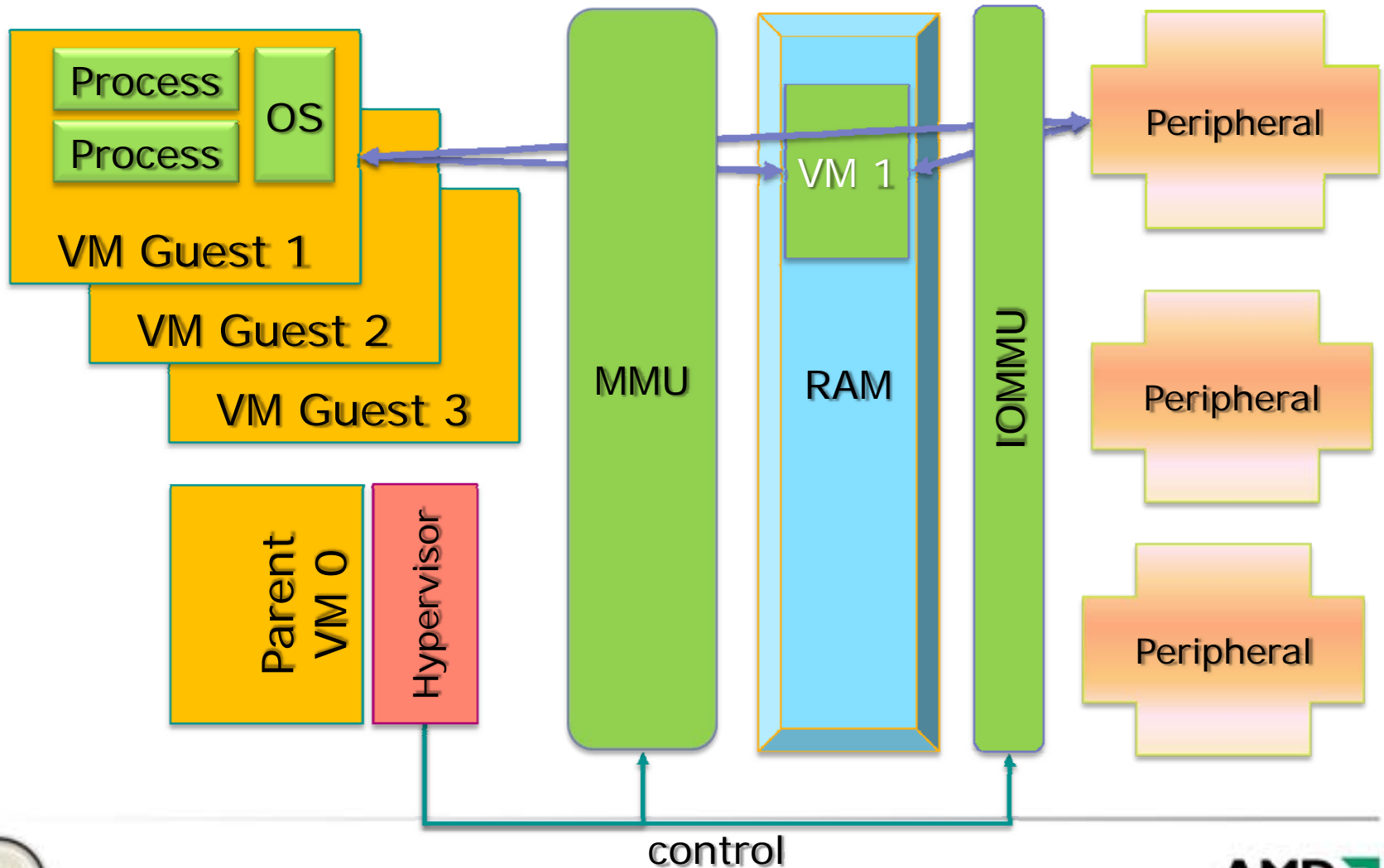
# IOMMU Role In System



# I/O bottleneck illustrated

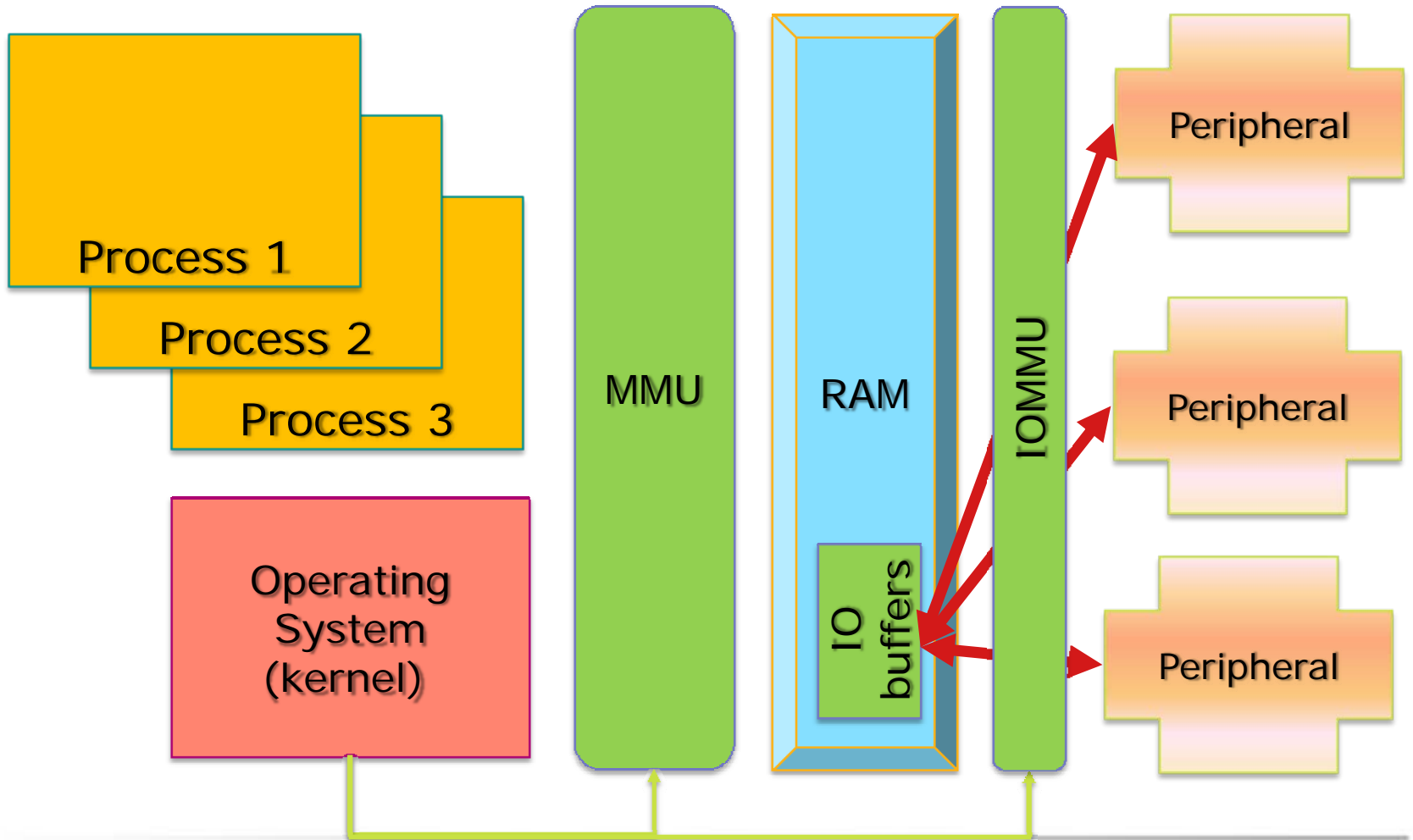


# I/O Device Assignment

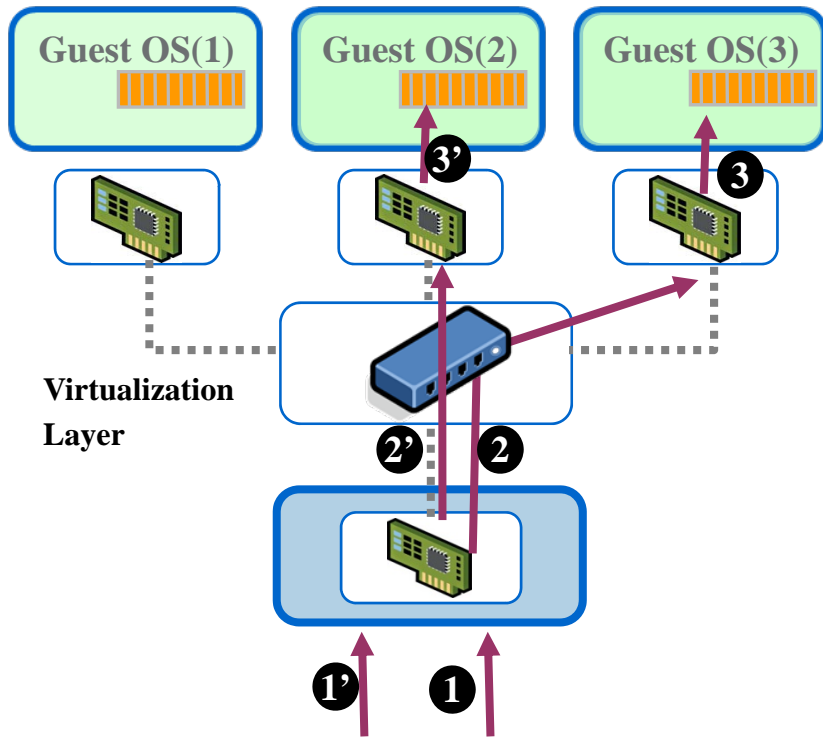


# Device Protection

## No virtualization



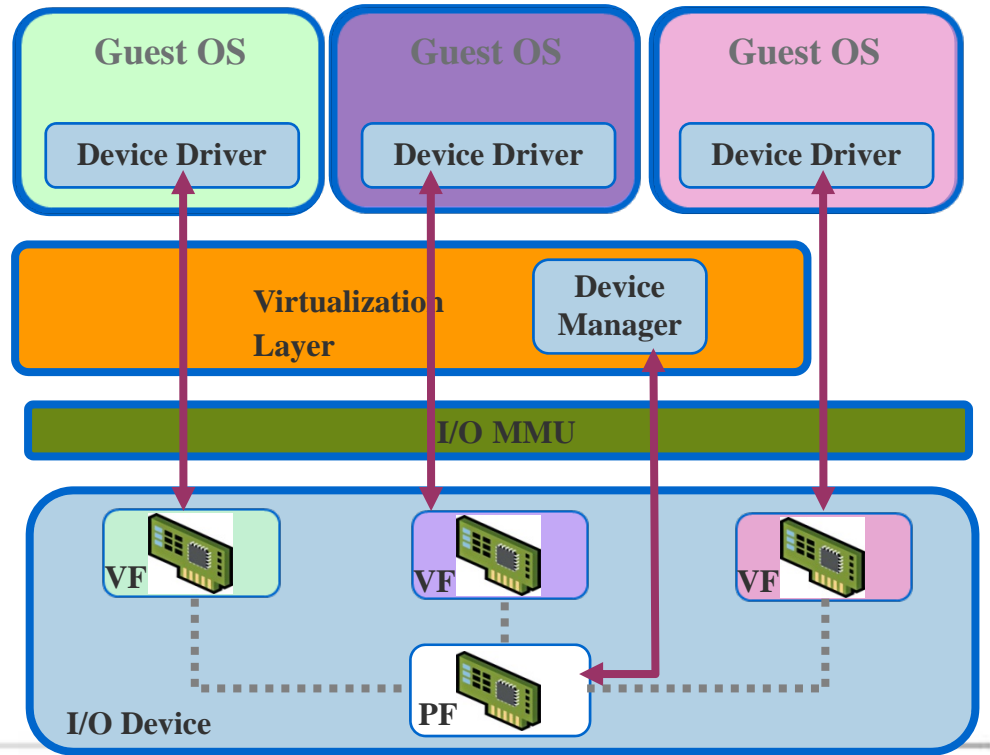
# VMware NIC Example



Packets for guest OS 2 and 3

## NetQueue

## Fixed Pass through I/O



PF = Physical Function, VF = Virtual Function

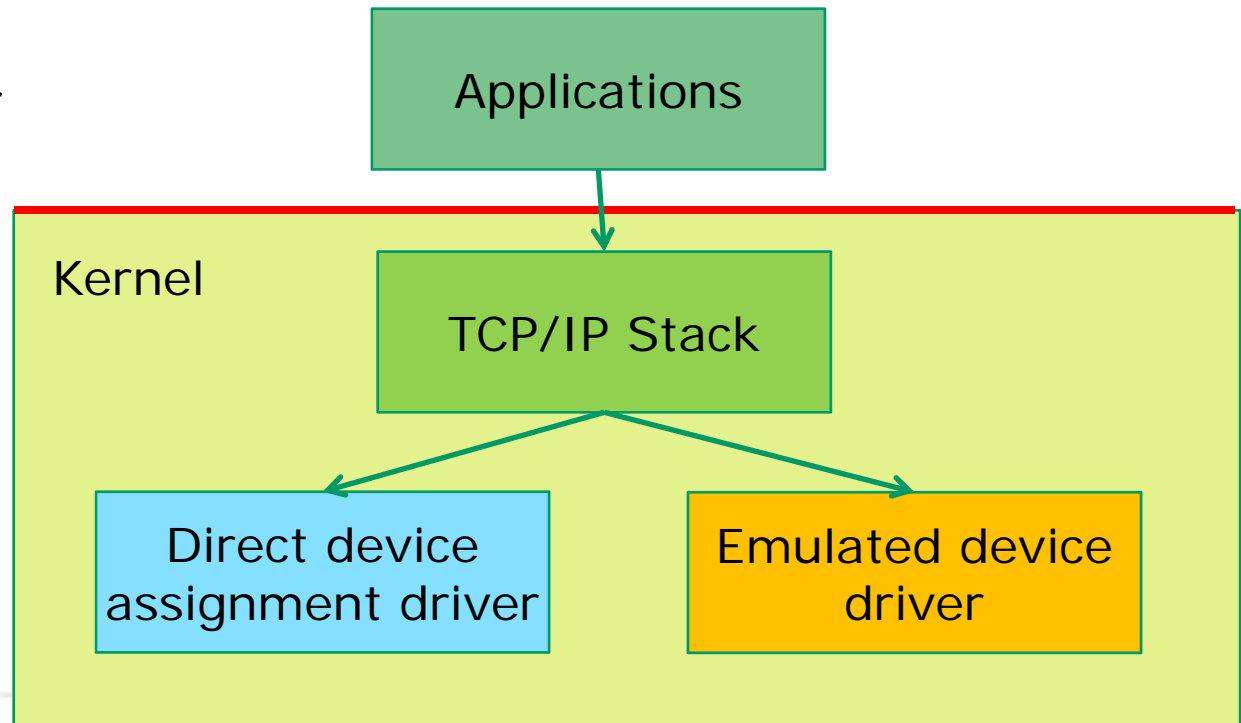


# OS Multi-Pathing

Multiple paths to the destination and the system can automatically fall back between different paths

Originally designed as a fault tolerant feature

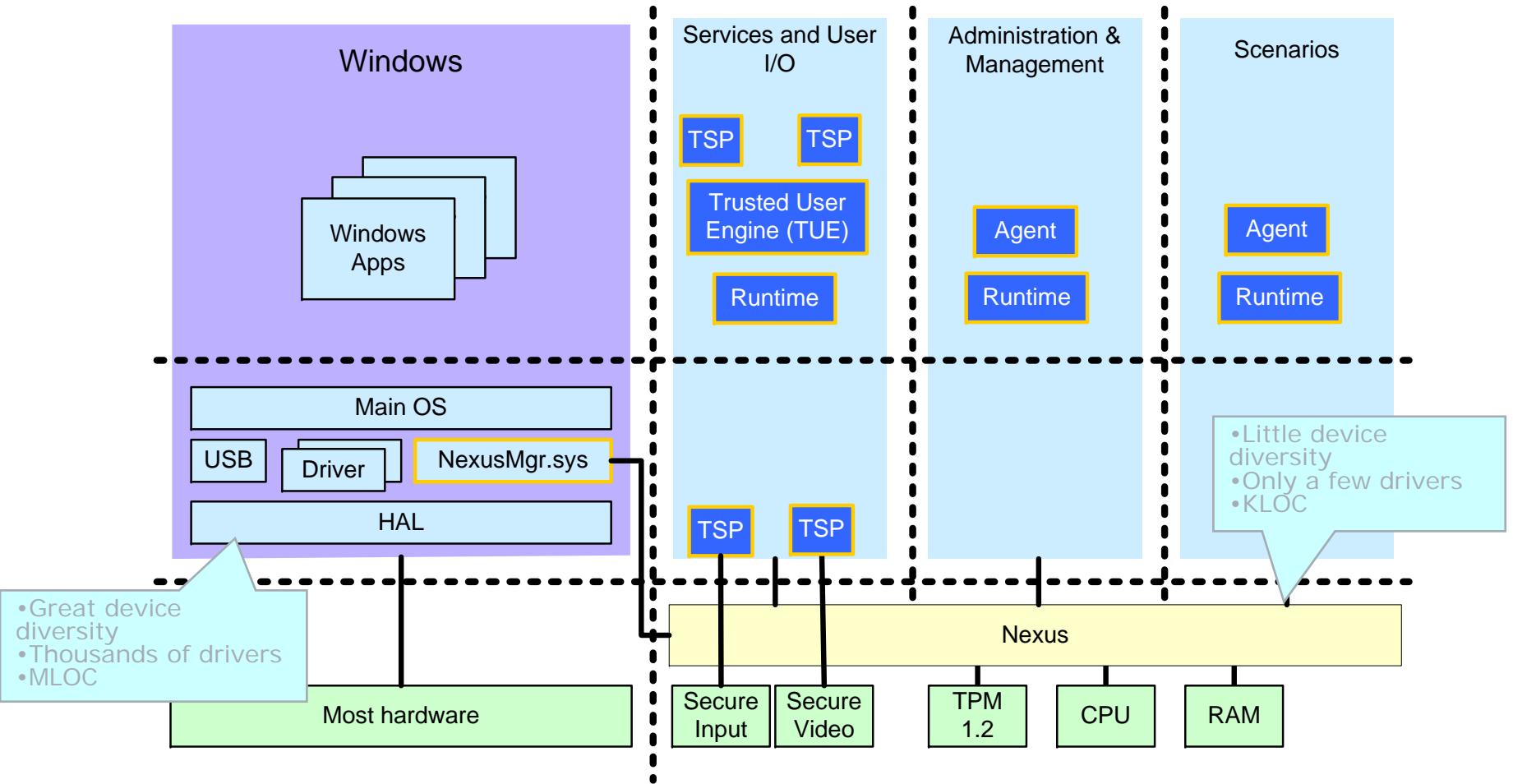
- State transfer
- Windows
- Solaris



# Security at Work



# Microsoft's 2004 NGSCB Version



Source: Microsoft's WinHEC 2004 presentation



# Secure Kernel Initialization

A lot can happen before the Kernel gets to run

- BIOS, extension BIOSes, bootstrap loaders, etc.
- Do we really have to trust this?

Dynamic Root of Trust is used to alleviate this

- It enables a secure **late launch** of the operating system
- Unknown state → known secure state

AMD has been shipping this capability since 2006 as part of its virtualization extensions

Intel has been shipping this since 2008 as part of their Trusted Execution Technology

Adoption is likely to happen in 2010/2011



# New AMD64 Instruction: SKINIT

**Objective:** Known code executing in known (quiet) environment

**Result:** Forms the basis for establishing trust in the platform environment

SKINIT instruction behavior:

- Performs an “INIT” of the CPU
- Resets the architecturally visible CPU state to known values
- Removes Microcode patches
- Activates special DMA protection over the Secure Loader (SL) code
- Multi-Processor “Safety Check”
- SL code is copied to the TPM using SKINIT-only special cycles
- Unconditional Jump to entry point in the SL code



# What Else Can You Expect?

A few more virtualization acceleration widgets

- Virtualized interrupt controller (interrupts, IPIs)

Additional hardware RAS capabilities

- *Putting all your eggs in one basket*

Quality-of-Service guarantees

- Hard to give performance guarantees in a virtualized environment

High-availability / Disaster recovery / VM failover

- SMP support

Nested/recursive virtualization

- Embedded hypervisors are driving this

Virtualization aware devices

- NICs, storage, graphics adapters



# Summary



# Summary

Virtualization is becoming pervasive

- Server consolidation is still the primary reason
- Dynamic (next gen) data center is quickly becoming another
- Client virtualization

I/O virtualization is driving platform, adapter and software stack changes

Platform virtualization capabilities are rounded out and exceed the capabilities of mainframes

Novell uses of virtualization (security, high-availability, manageability)



## Trademark Attribution

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2009 Advanced Micro Devices, Inc. All rights reserved.

