



Intel® Virtualization Technology: A Roadmap Overview and Update

Naresh K Sehgal, Ph.D., MBA
SW Lead Architect,
Enterprise Platforms and Services Division

Acknowledgment: Thanks to Rich Uhlig

Session ID#SVTS001

Intel Developer
FORUM

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
- Copyright © 2006 Intel Corporation.

Throughout this presentation:

VT-x refers to Intel® VT for IA-32 and Intel® 64

VT-i refers to the Intel® VT for IA-64, and

VT-d refers to Intel® VT for Directed I/O

Intel® Platforms



Business
Desktop

- **Built-in Manageability**
- **Proactive Security**
- Energy Efficient Performance



Digital
Home

- Performance
- Energy Efficient
- Connectivity
- Ease of Use



Mobility

- Performance
- Battery Life
- Uncompromised Connectivity
- Innovative Form Factor



Server

- Breakthrough Performance
- Energy Efficient
- **Built for Virtualization**
- Data Intensive Computing



Intel® Virtualization Technology: A Roadmap Overview and Update

Agenda:

- IA Virtualization Today
- Intel® VT Roadmap Update
- Summary and Questions

Agenda

- Intel® Architecture (IA) Virtualization Today
 - Server and client applications of virtualization
 - Overview of VMM software challenges
- Intel® VT Roadmap Update
 - Overview of CPU, platform and I/O virtualization support
 - How Intel® VT addresses virtualization challenges
- Summary and Questions

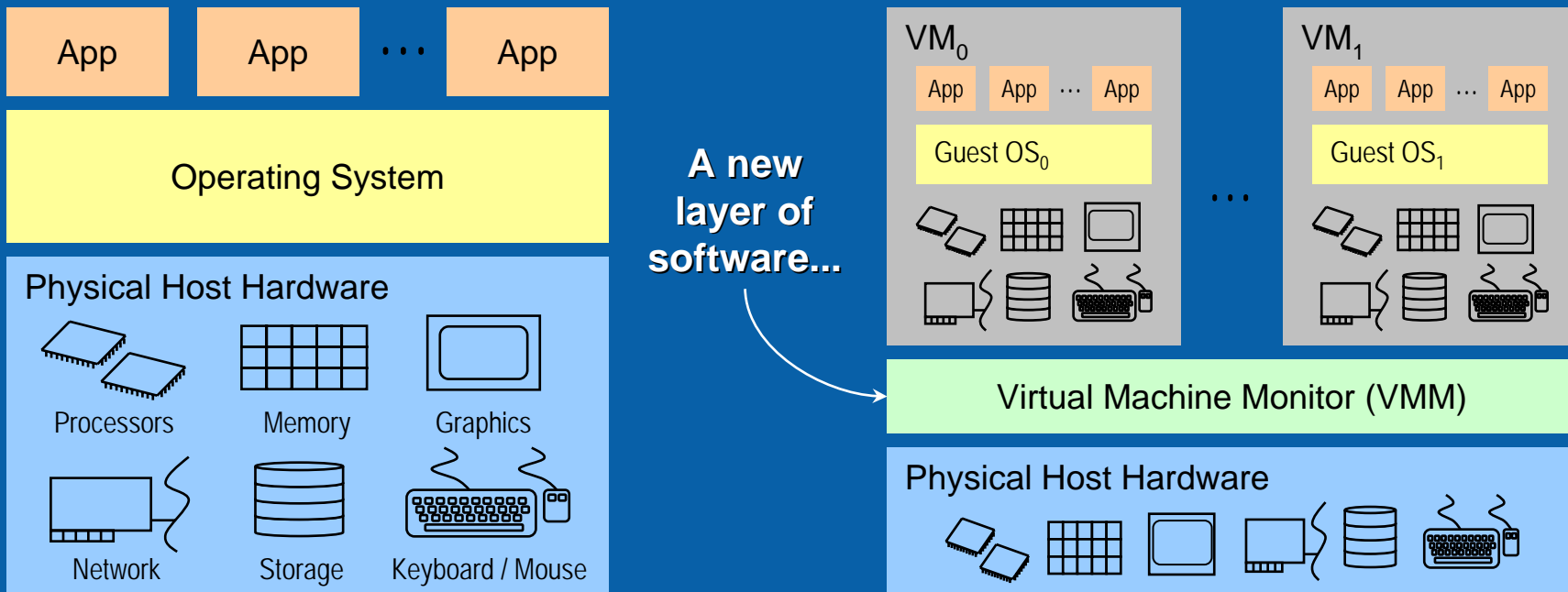


Intel® Architecture (IA) System Virtualization Today



Intel Developer
FORUM

Hardware Virtual Machines (VMs)



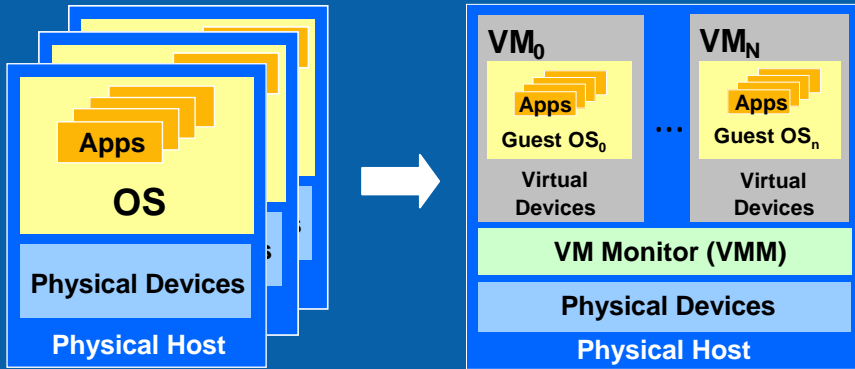
Without VMs: Single OS owns all hardware resources

With VMs: Multiple OSes share hardware resources

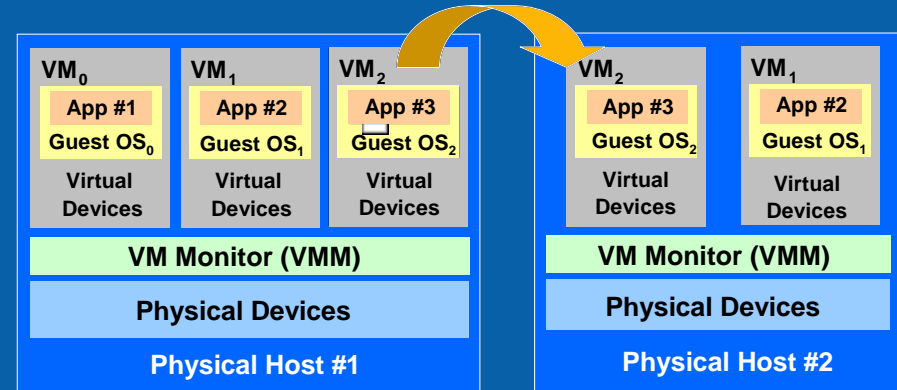
Virtualization enables multiple operating systems to run on the same physical platform

VM Usage Models

Server Consolidation

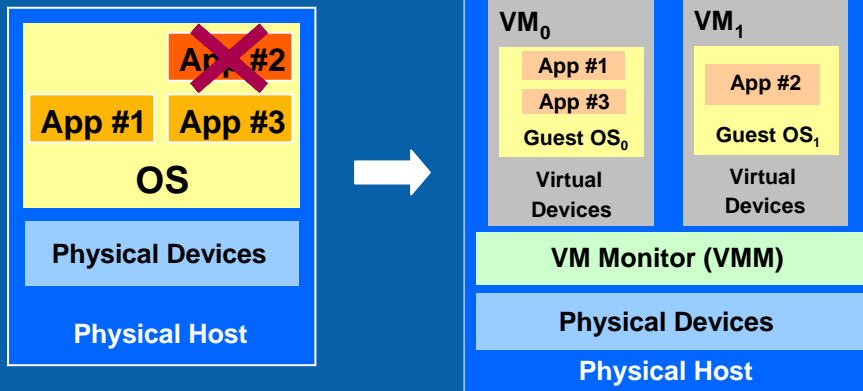


VM Migration / Load Balancing / Fail-over

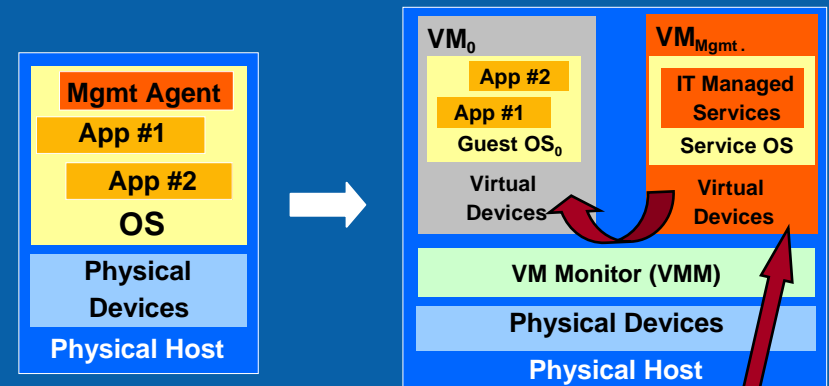


Environment Isolation

(Based on user, application, security, activity)



Embedded IT



Intel® VT Goals

Reduce VMM Complexity

- Close hardware “virtualization holes” by design
- Reduce need for device-specific knowledge in VMM

Enhance Reliability and Protection

- Provide new control over device DMA and interrupts

Improve Functionality

- Provide support for legacy (unmodified) guest OSes
- Enable pass-through access to I/O devices (where appropriate)

Increase Performance

- Eliminate unnecessary transitions to VMM
- New address-translation mechanisms (for CPU and devices)
- Reduce memory requirements (translated code, shadow tables)

Provide a flexible set of hardware primitives to aid VMM software

Intel® VT Roadmap: Overview

Vector 3:
I/O Focus

PCI-SIG

Standards for I/O-device sharing:

- Natively sharable I/O devices
- Endpoint DMA-translation caching



Vector 2:
Platform Focus

VT-d

Infrastructure for I/O-device virtualization:

- DMA protection and remapping
- Interrupt filtering and remapping



Vector 1:
Processor Focus

VT-x

VT-i

Establish foundation for virtualization in the Intel® 64 and Itanium® architectures...

... followed by on going evolution of support:

- Microarchitectural (e.g., lower VM entry/exit costs)
- Architectural (e.g., extended page tables – EPT)



VMM
Software
Evolution

Software-only VMMs

- Binary translation
- Paravirtualization
- Device Emulation

Simpler and more Secure VMMs through foundation of virtualizable ISAs

Improved CPU and I/O virtualization **Performance and Functionality** as VMMs exploit infrastructure provided by VT-x, VT-i, VT-d



Past
No Hardware Support

Today



VMM software evolution over time with hardware support

CPU Virtualization Challenges

Ring Deprivileging

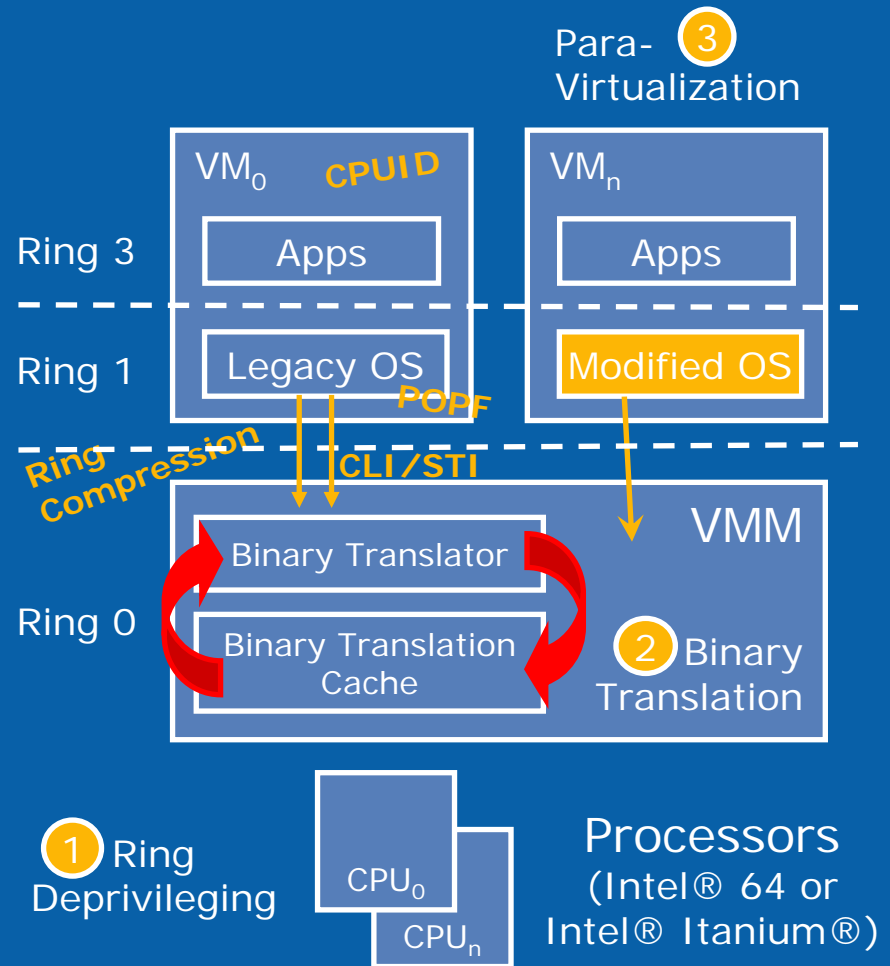
- Run guest OS above ring 0
- Control access to privileged state

Virtualization Holes

- Ring Compression
- Non-trapping operations (e.g., POPF, CPUID)
- Excessive trapping (e.g., CLI, STI)
- Context switching CPU state (e.g., "hidden" segment state)

Software Methods

- Binary Translation
- Paravirtualization



CPU Virtualization with VT-x

New CPU Operating Mode

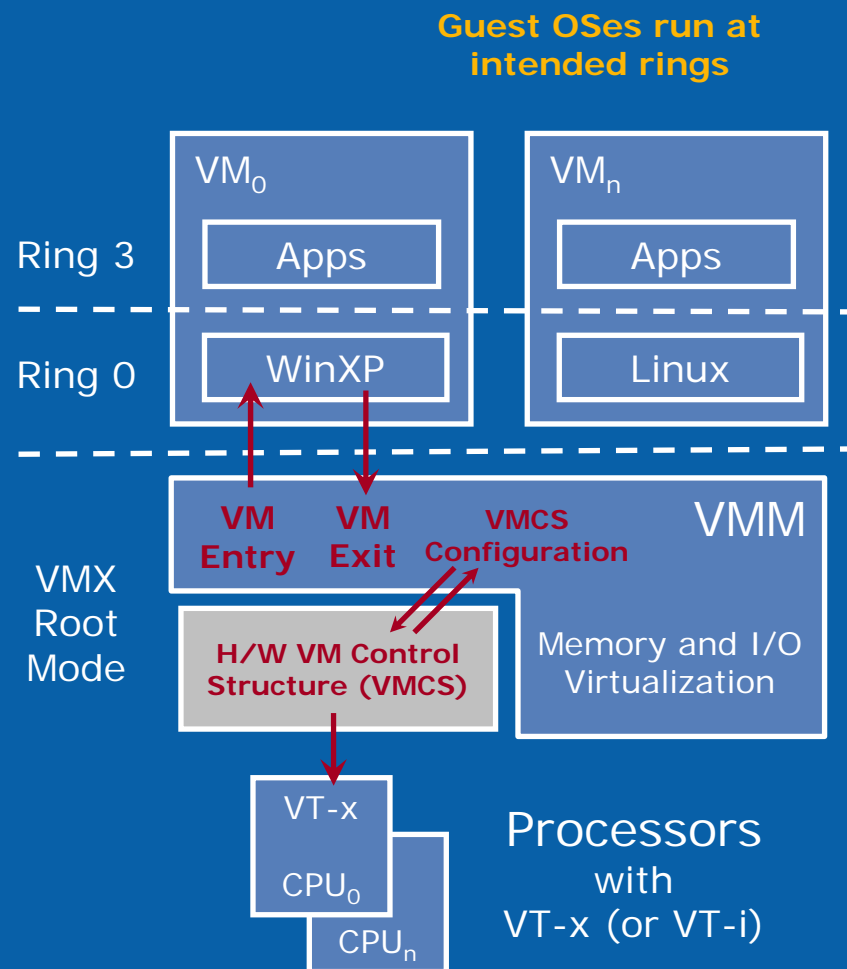
- VMX Root Operation (for VMM)
- Non-Root Operation (for Guest)
- Eliminates ring deprivileging

New Transitions

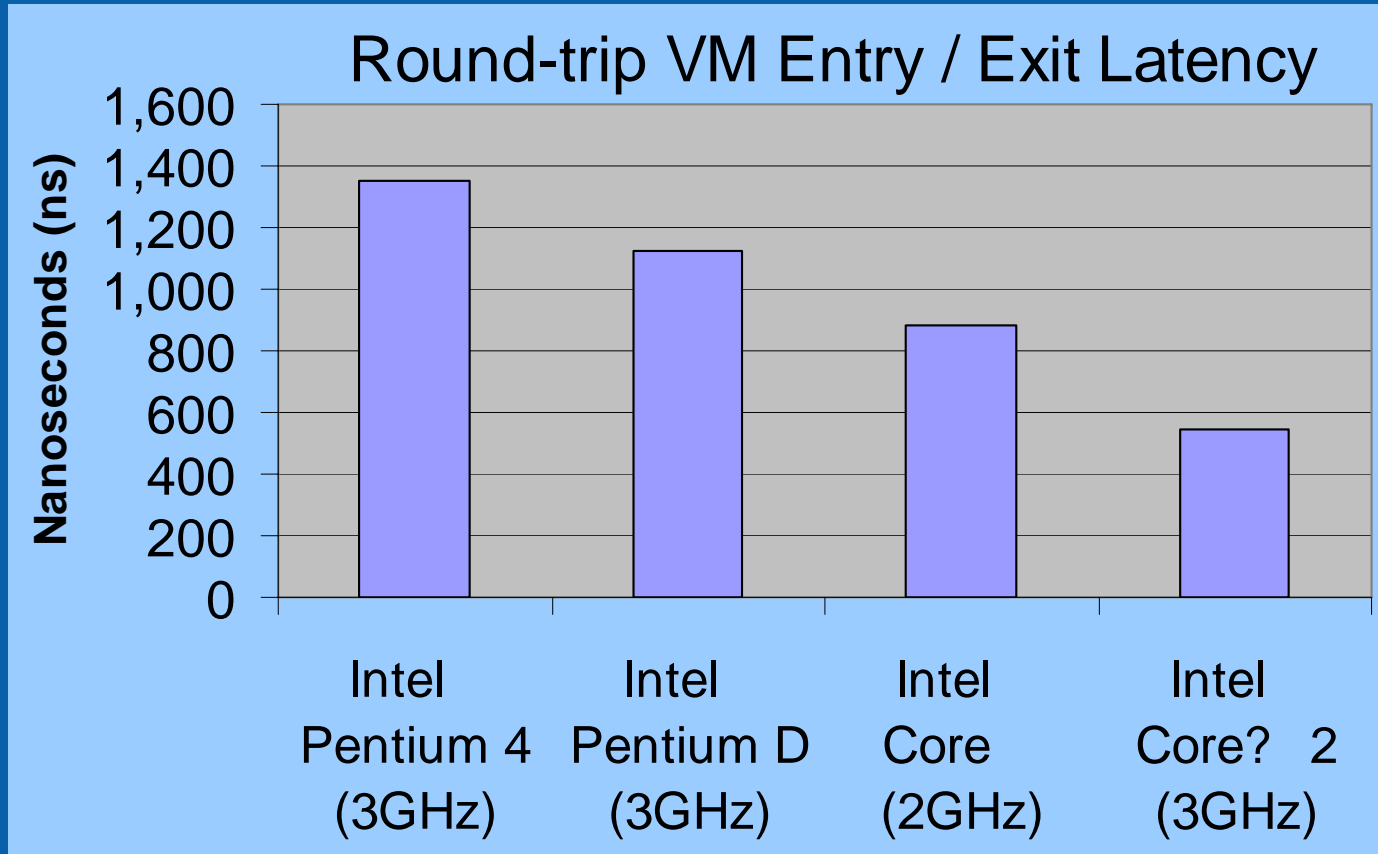
- VM entry to guest OS
- VM exit to VMM

VM Control Structure (VMCS)

- Configured by VMM software
- Specifies guest OS state
- Controls when VM exits occur (eliminates over and under exiting)
- Supports on-die CPU state caching



VT-x Microarchitecture Enhancements



- Implementations leveraging VMCS caching over time
- Further improvements planned for future implementations

Memory Virtualization Challenges

Address Translation

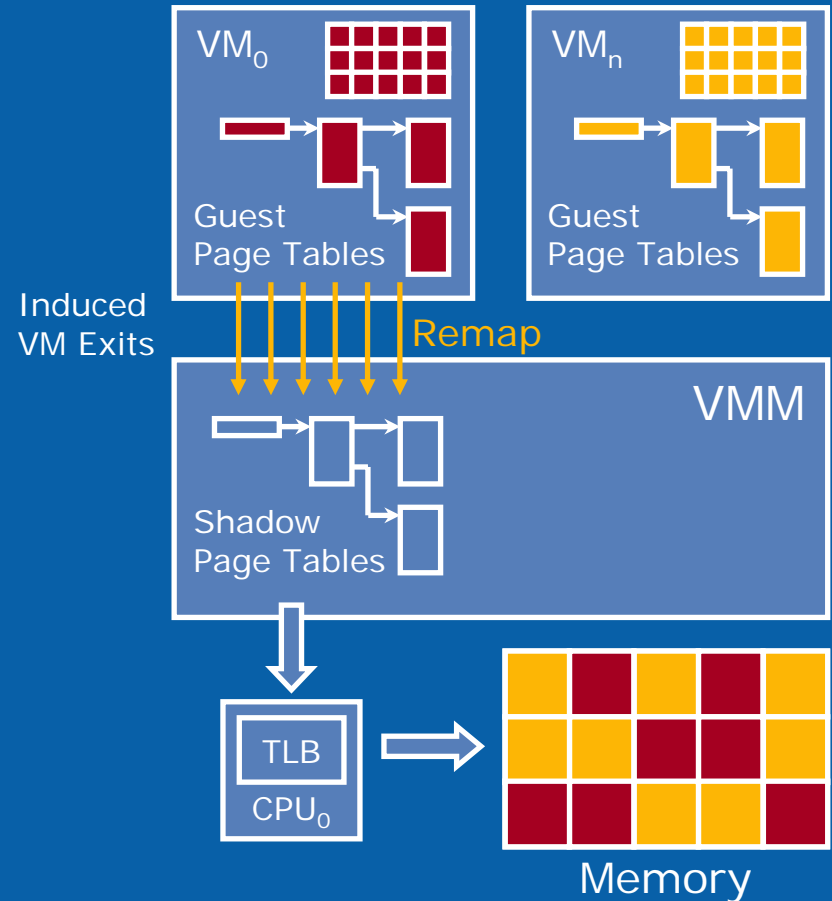
- Guest OS expects contiguous, zero-based physical memory
- VMM must preserve this illusion
- Must control both CPU and device DMA accesses to memory

Page-table Shadowing

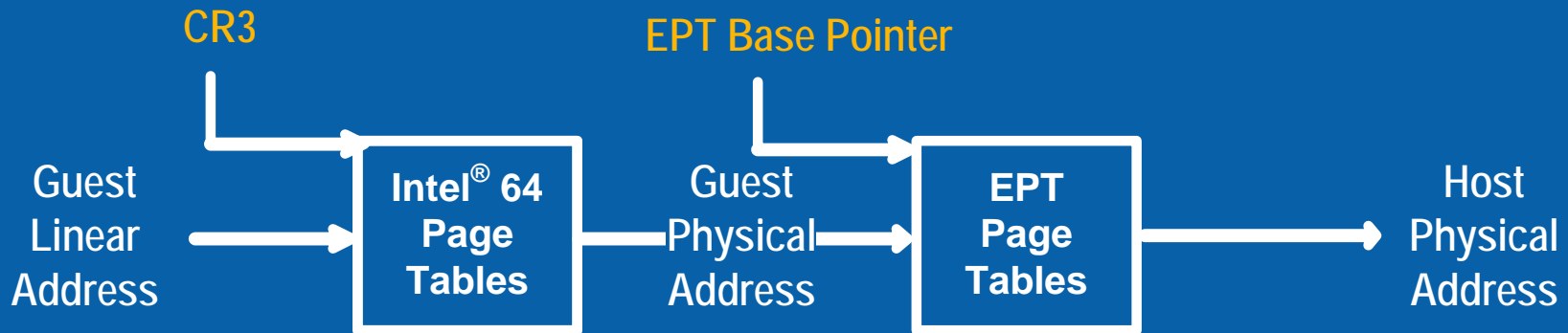
- VMM intercepts paging operations (e.g., #PF, CR3, INVLPG)
- Constructs shadow copy of page tables with address translations

Overheads

- VM exits add to execution time
- Shadow page tables consume significant host memory

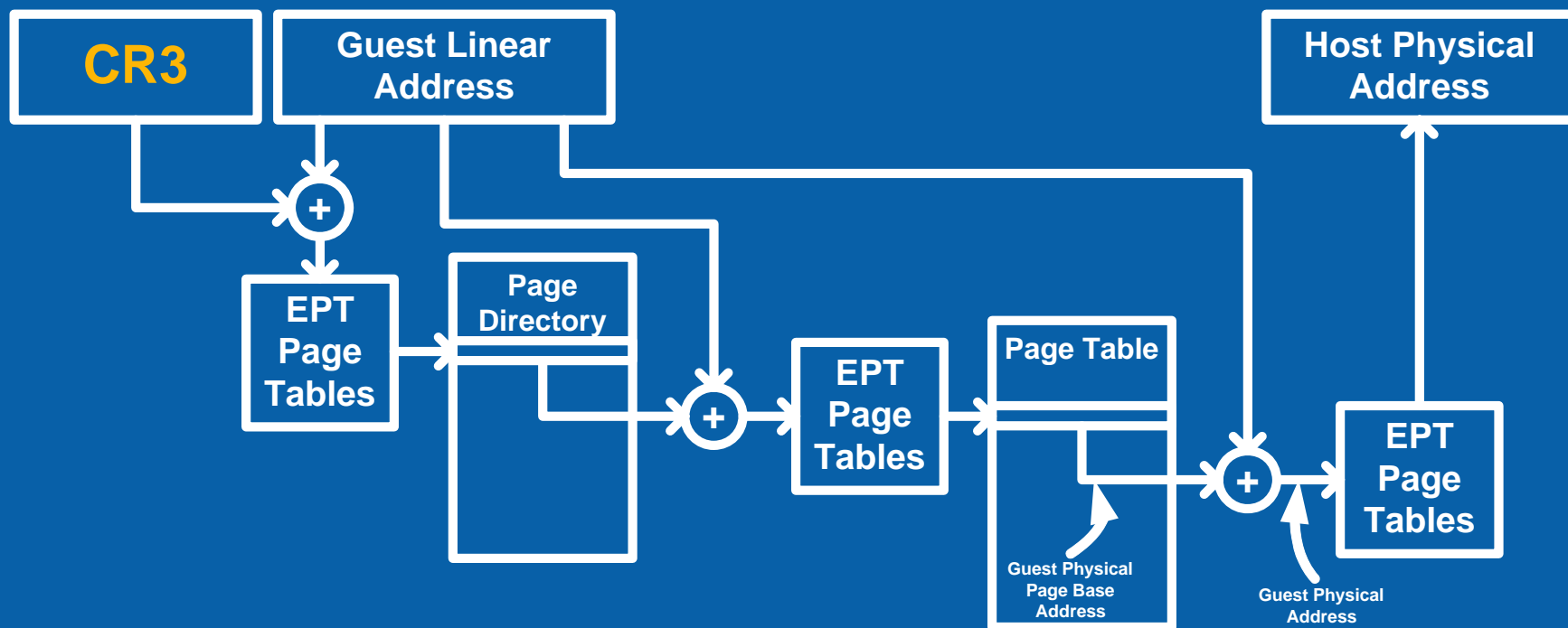


Extended Page Tables (EPT)



- Intel® 64 page tables
 - Map **guest-linear** to **guest-physical** (translated again)
 - Can be read and written by guest
- New EPT page tables under VMM control
 - Map **guest-physical** to **host-physical** (accesses memory)
 - Referenced by new **EPT base pointer**
- No VM exits due to **page faults**, **INVLPG**, or **CR3** accesses

EPT Translation: Details



- All guest-physical addresses translated by EPT
 - CR3, PDE, PTE
 - Includes PDPTRs and 64-bit paging structures

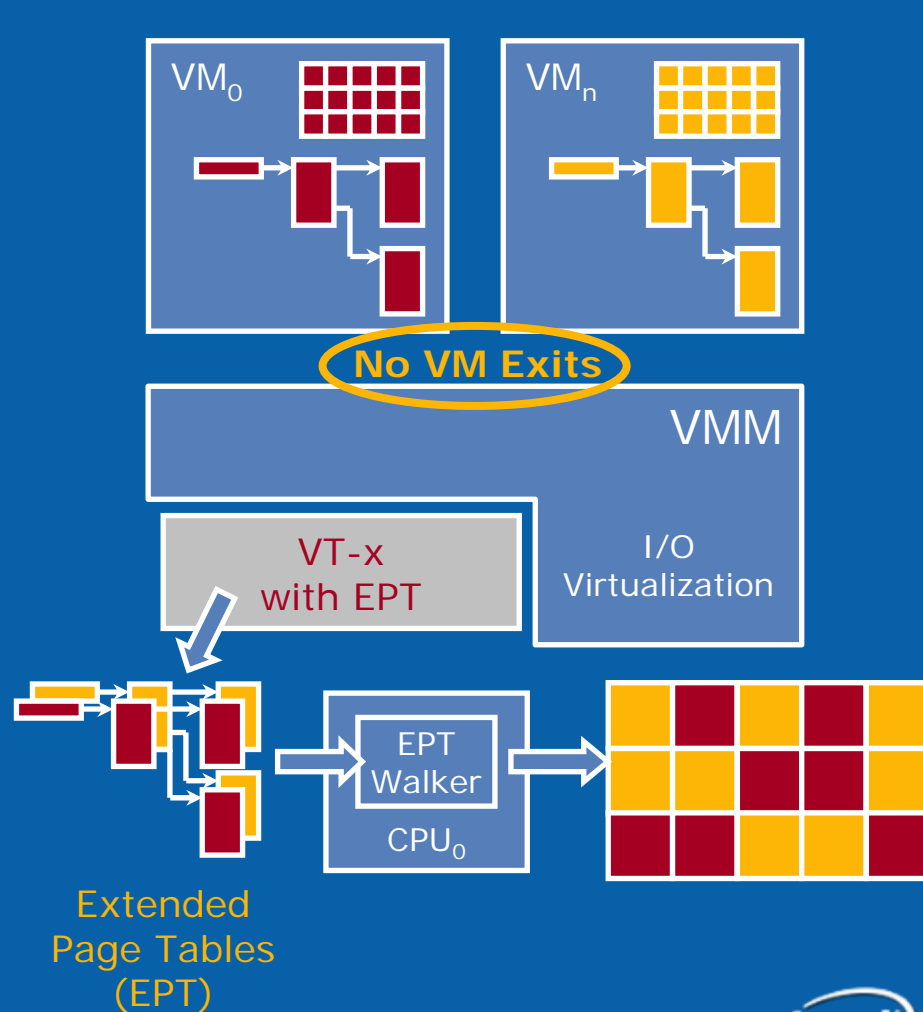
Memory Virtualization with Intel® VT

Performance Benefit

- Guest OS able to modify its own page tables
- Eliminates VM exits

Memory Savings

- Shadow page tables required for each guest user process (w/o EPT)
- Many MB of memory needed for shadow page tables
- A single EPT supports entire VM



I/O Virtualization Challenges

Physical Device Driver

- Leverage drivers of a hosting OS
- Or, build drivers into the VMM

Virtual Device Interface

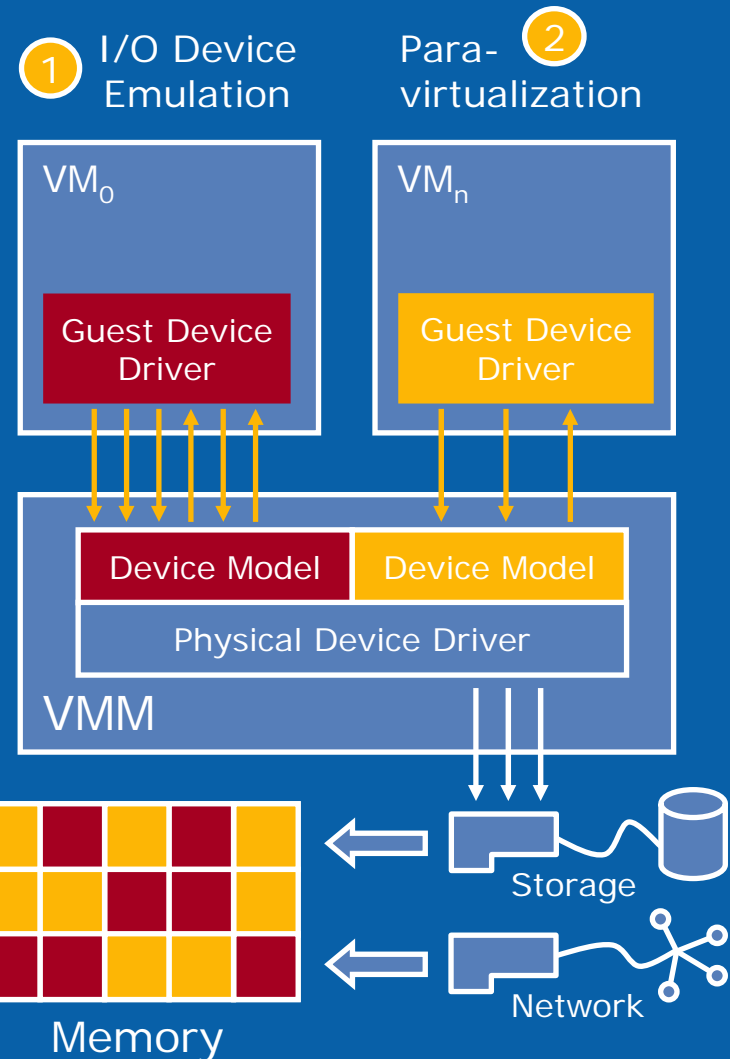
- Trap device commands
- Virtual DMA translation
- Injecting virtual interrupts

Software Methods

- I/O Device Emulation
- Paravirtualize Device Interface

Challenges

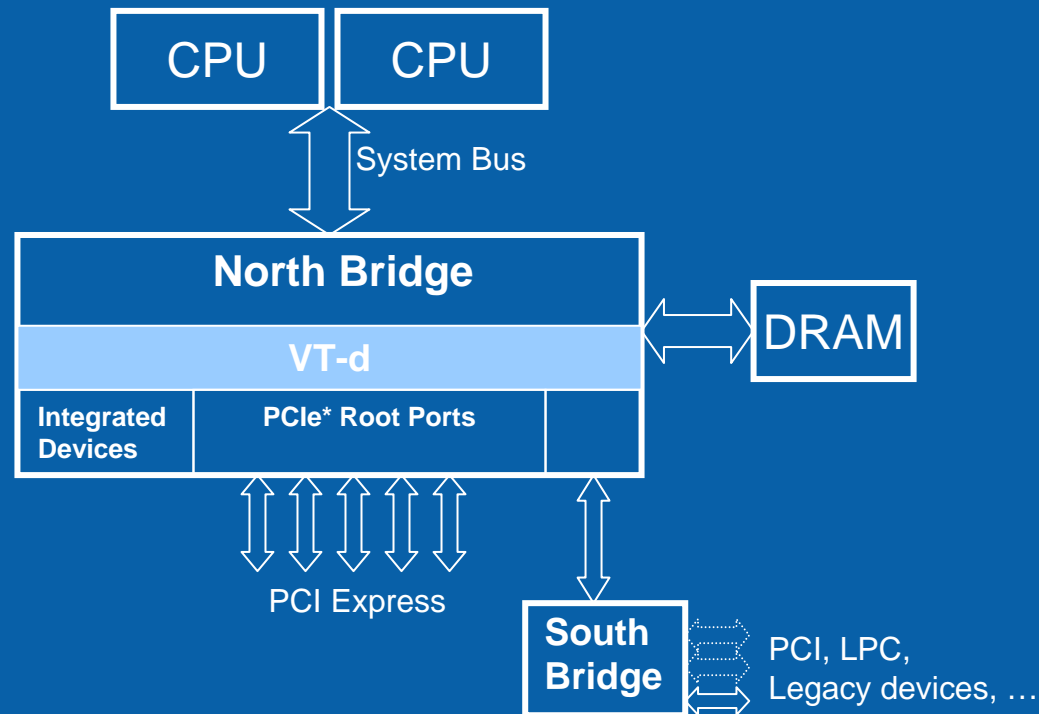
- Controlling DMA and interrupts
- Rich I/O device functionality



VT-d Overview

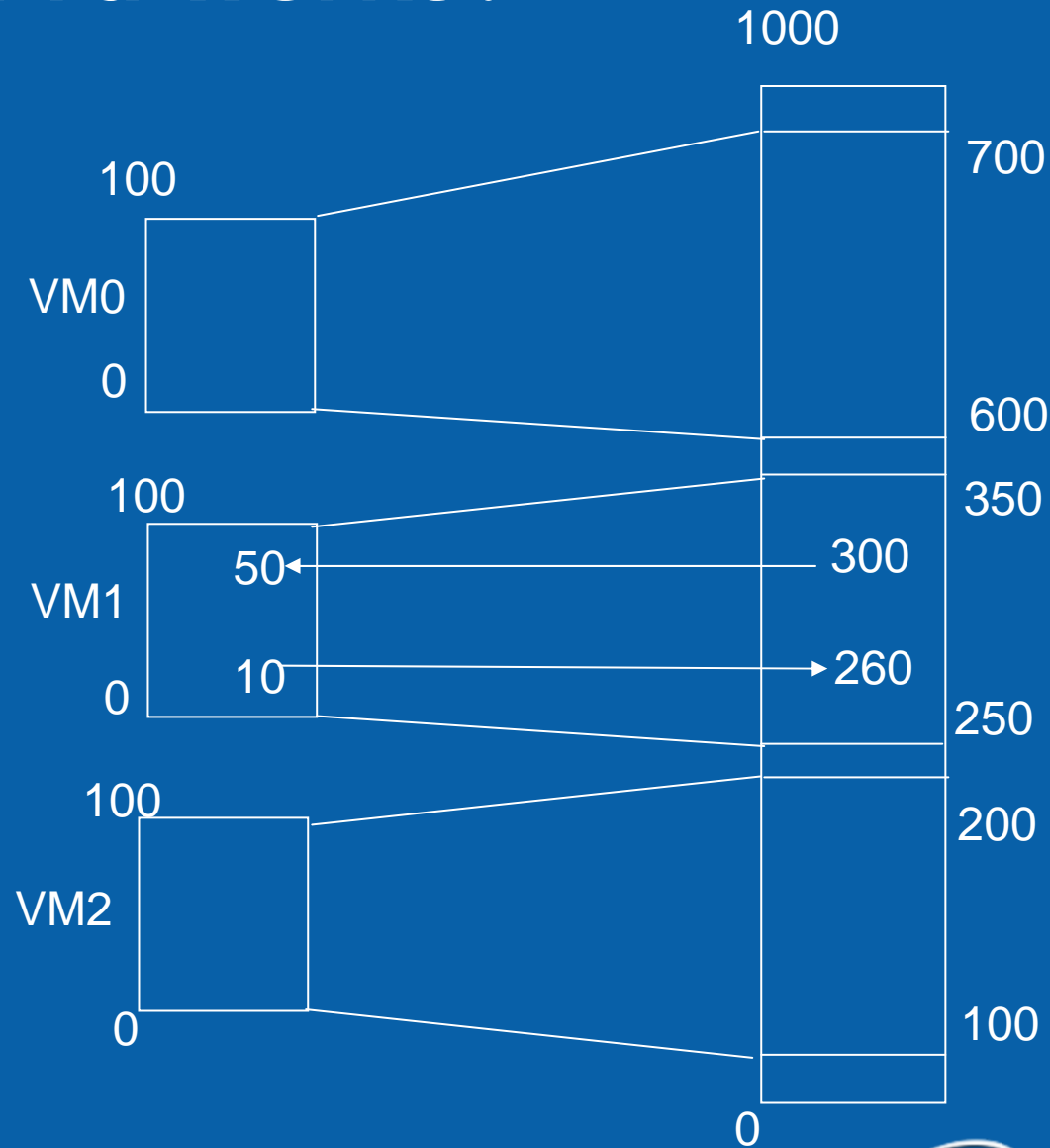
VT-d is platform infrastructure for I/O virtualization

- Defines an architecture for DMA and interrupt remapping
- Implemented as part of core logic chipset
- Will be supported broadly in Intel server and client chipsets

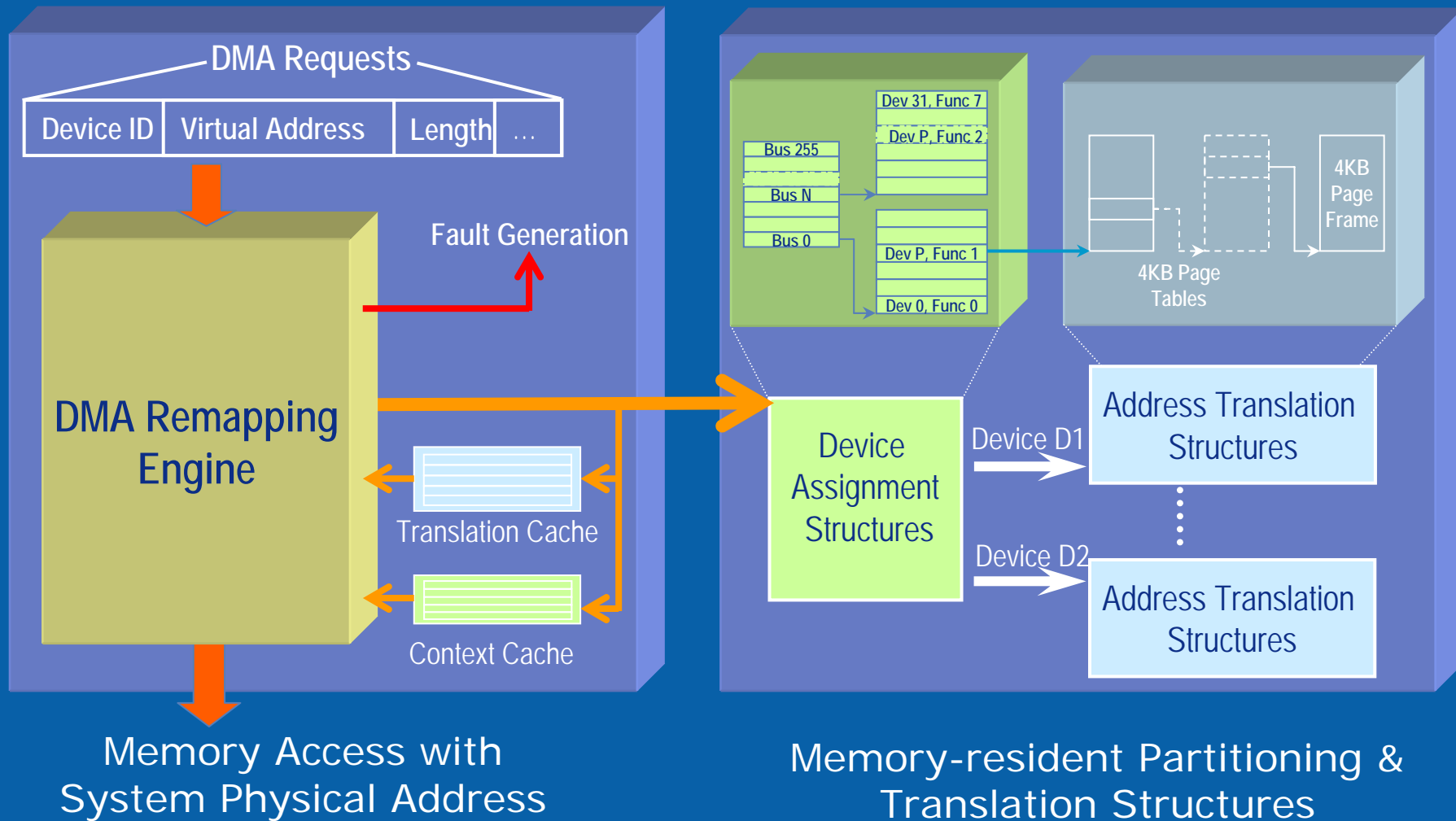


How VTd works?

- Each VM thinks it is 0 address based
 - GPA (Guest Physical Address)
- But mapped to a different address in the system memory
 - HPA (Host Physical Address)
- VTd does the address mapping between GPA and HPA
- Catches any DMA attempt to cross VM memory boundary

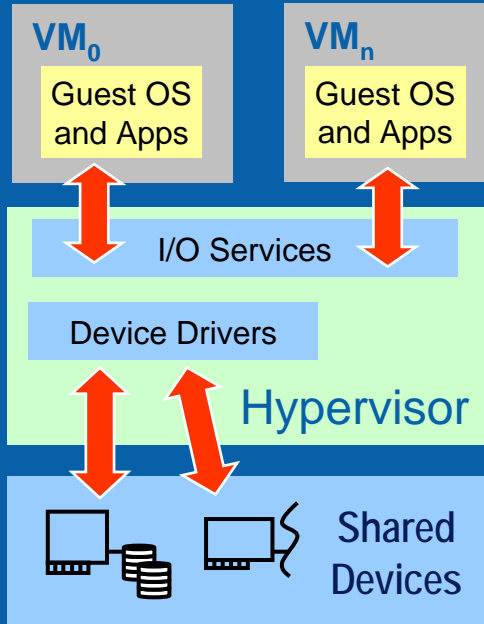


VT-d Architecture: Detail



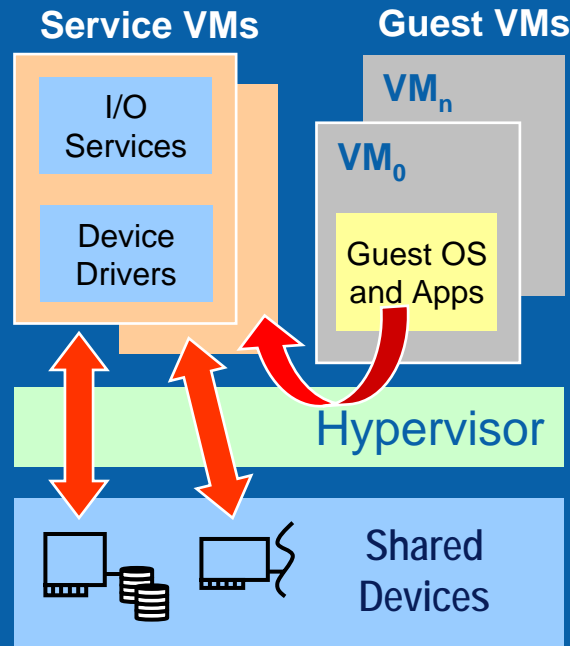
VTd Applications for I/O Virtualization

Hypervisor Model



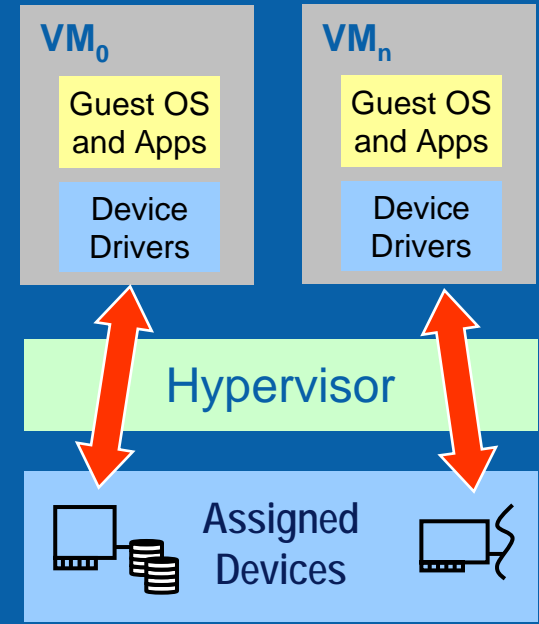
- Pro: High Performance
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Large Hypervisor

Service VM Model



- Pro: Higher Security
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Lower Performance

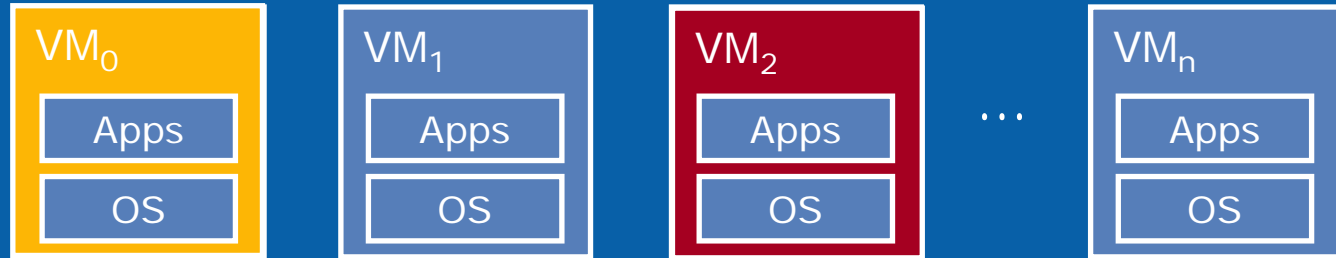
Pass-through Model



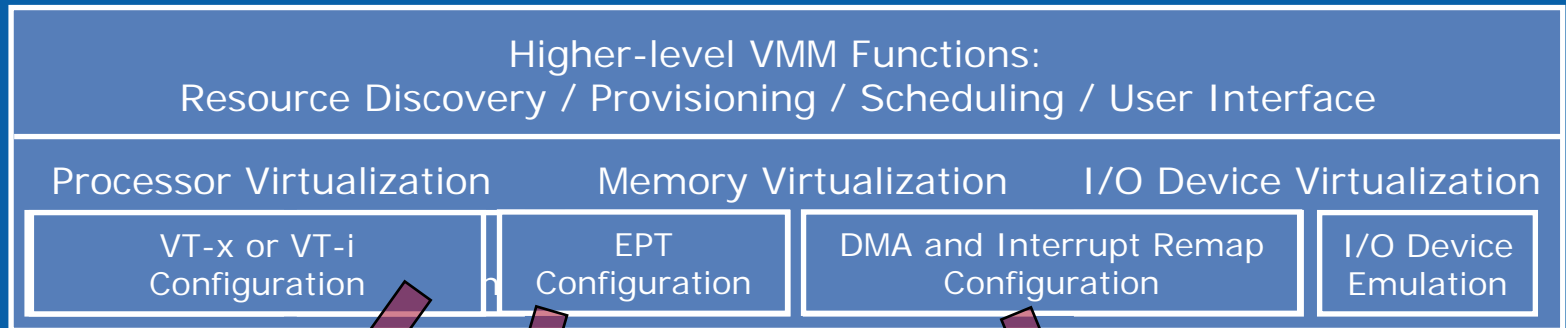
- Pro: Higher Performance
- Pro: Rich Device Features
- Con: Limited Sharing
- Con: VM Migration Limits

Putting it all together...

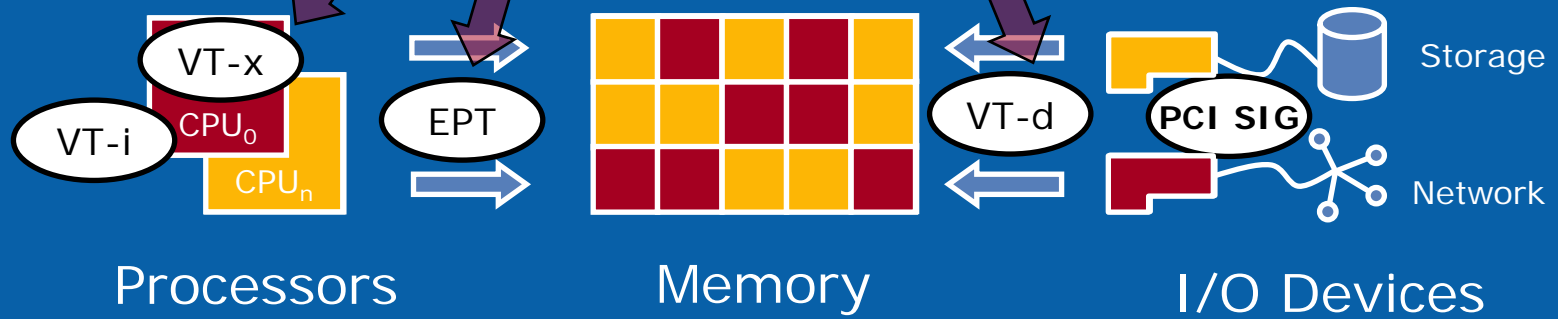
Virtual Machines (VMs)



VMM (a.k.a., hypervisor)



Physical Platform Resources



Delivering Intel® VT

- **Intel® VT Specifications** available
 - For the IA-32 and Intel® 64 Architecture VT-x
 - For the Intel® Itanium® Architecture VT-i
 - For Directed I/O Architecture VT-d
 - See: <http://www.intel.com/technology/virtualization>
- **Shipping** platforms enabled with Intel® VT since 2005
 - **VT-x**: Available in all Intel-based processors (server, desktop, mobile)
 - **VT-i**: Now available in Intel® Itanium® processor-based servers
 - **VT-d**: Working with VMM vendors to deliver software support with systems in 2007
- **Broad VT Ecosystem Support** from VMM Vendors
 - Visit the demo showcase for VT-enabled VMMs

Summary

- Intel® Virtualization Technology (Intel® VT)
 - A comprehensive roadmap to address virtualization challenges
 - Support for CPU and I/O virtualization
 - Strong VMM ecosystem support
- To learn more:
Intel® VT Web Site: <http://www.intel.com/technology/virtualization>
- Intel Technology Journal
Special issue on virtualization technology, Volume 10, Issue 03
See: <http://www.intel.com/technology/itj/>
- Technical book from Intel Press:
Applied Virtualization Technology
by Sean Campbell and Michael Jeronimo
For more info: <http://www.intel.com/intelpress/>

Please fill out the Session Evaluation Form.

Session presentation will be available on IDF web
site – www.prcidf.com.cn

Thank You!

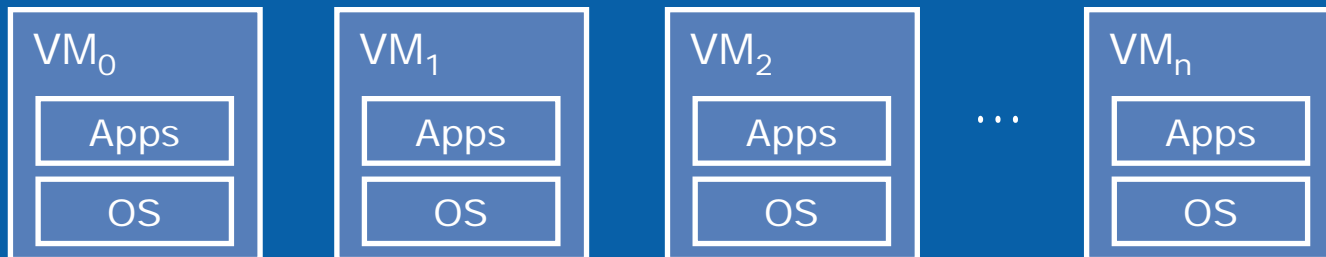


Intel Developer
FORUM

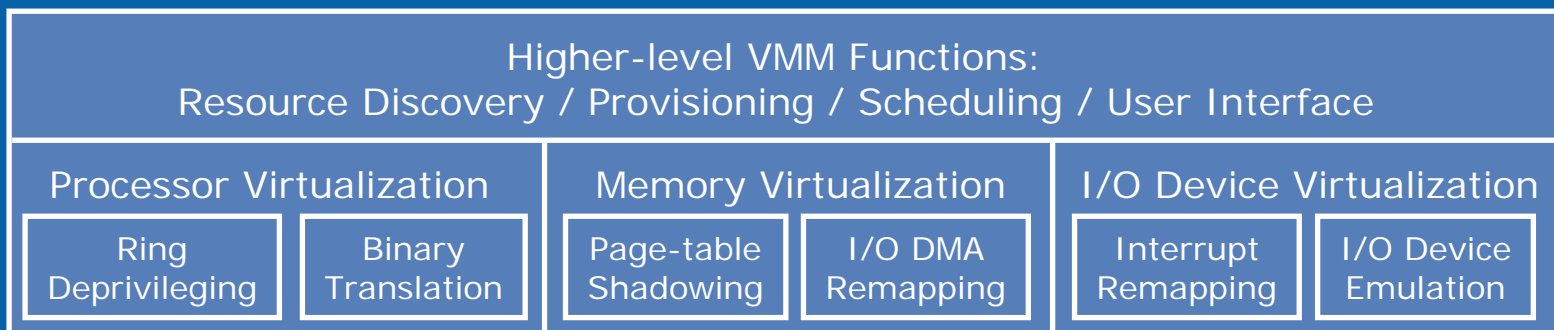


Virtual Machine Monitor (VMM)

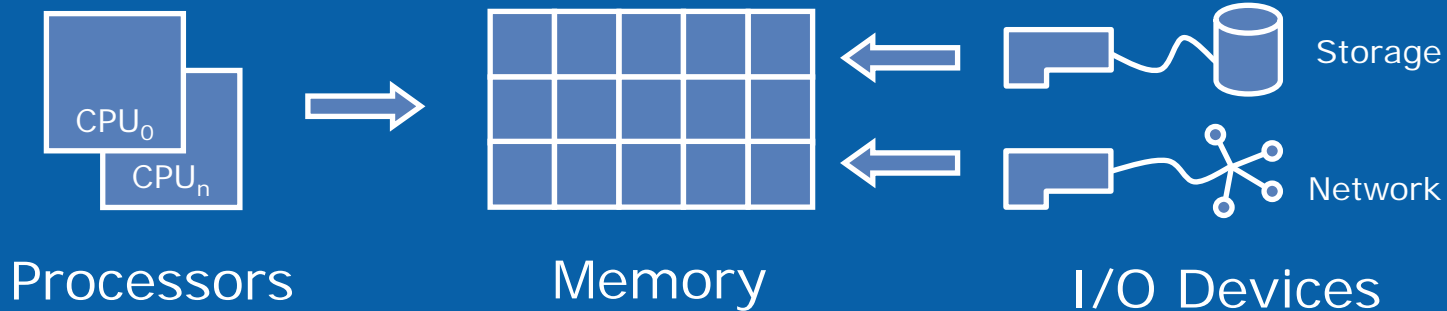
Virtual Machines (VMs)



VMM (a.k.a., hypervisor)



Physical Platform Resources



**Key VMM Challenge:
Controlling Physical Platform Resources**

How VT Addresses Virtualization Challenges

Reduced VMM Complexity

- Closes hardware “virtualization holes” by design
- Reduces need for device-specific knowledge in VMM

Enhanced Reliability and Protection

- Provides new control over device DMA and interrupts

Improved Functionality

- Provides support for legacy (unmodified) guest OSes
- Enables pass-through access to I/O devices (where appropriate)

Increased Performance

- New address translation mechanisms (for CPU and devices)
- Reduces memory requirements (translated code, shadow tables)

VT provides a flexible set of hardware primitives to aid VMM software

Next-generation VT-x Features

1st generation of Intel® VT shipped last year

- Discussed in detail at past IDF presentations
- Several extensions planned for VT-x in future

Microarchitectural enhancements

- Example: reductions in VM entry/exit latencies
- No VMM software changes required

Architectural extensions

- Example: New VM execution controls
- Require VMM software updates to exploit these features

New VT-x Architecture Features

MSR Bitmaps

- Eliminates VM exits on non-sensitive MSRs (e.g., KernelGSbase)

Virtual-processor Identifiers (VPIDs)

- Supports retention of TLB entries across VM switches

NMI-window Exiting

- Enables timely delivery of NMIs to guest OS

Preemption Timer

- Allows VMM to bound guest-OS execution time

Descriptor-table Exiting

- Enables VMM to protect IDT, GDT, etc. from attack in guest OS

See Session IVTS002 for more details on these new VT-x features

VT-d Applied to Hypervisor Model

Improved Reliability and Protection

- VMM hypervisor programs remap tables
- Errant DMA is detected by hardware and reported to hypervisor / device driver

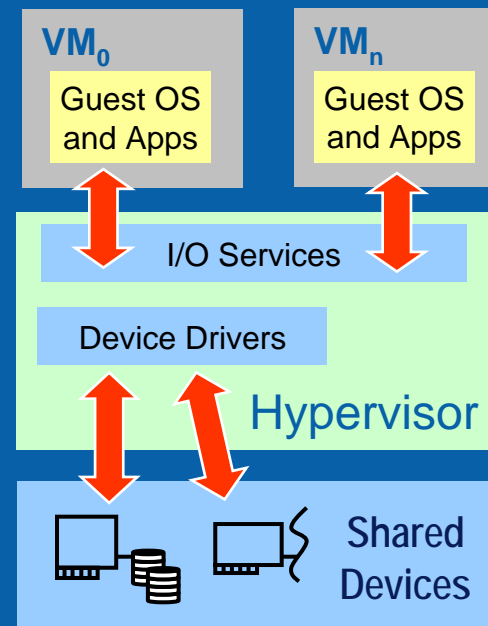
Bounce Buffer Support

- Limited DMA addressability in I/O devices limits access to high memory
- “Bounce buffers” are a software technique to copy I/O buffers into high memory
- VT-d eliminates need for “bounce buffers”

Above equally useful for standard OSes

- VT-d does not require a VMM to function

Hypervisor Model



- Pro: High Performance
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Large Hypervisor

VT-d Applied to Service VM Model

Device Driver Deprivileging

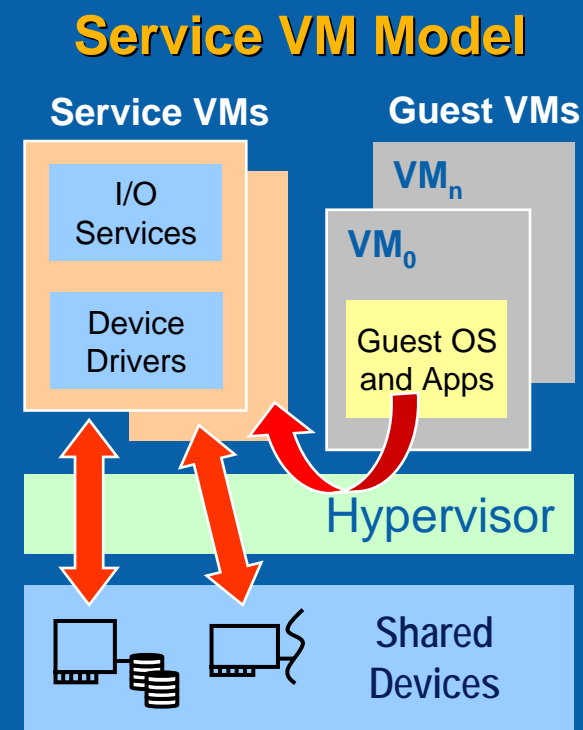
- Device drivers run above hypervisor as part of a “Service OS”
- Guest device drivers program devices in DMA-virtual address space

Service VM

- Forwards DMA API calls to hypervisor
- Hypervisor sets up DMA-virtual to host-physical translation

Further Improvements in Protection

- Guest device driver unable to compromise hypervisor code or data either through DMA or through CPU-initiated memory accesses



Pro: Higher Security

Pro: I/O Device Sharing

Pro: VM Migration

Con: Lower Performance

VT-d Applied to Pass-through Model

Direct Device Assignment to Guest OS

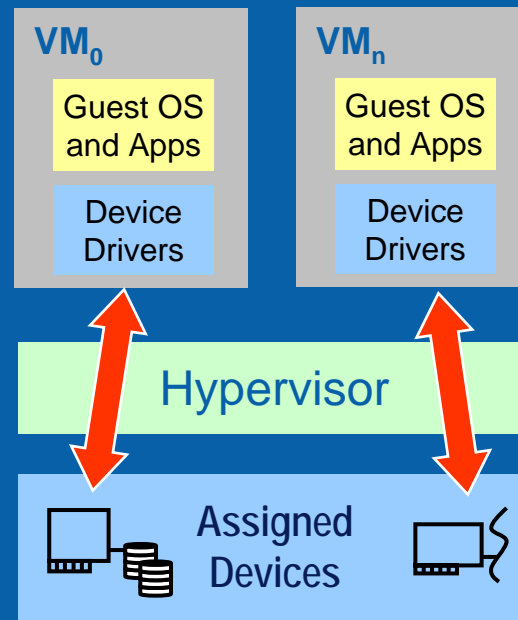
- Guest OS directly programs physical device
- For legacy guests, hypervisor sets up guest- to host-physical DMA mapping
- For remapping-aware guests, hypervisor involved in map/unmap of DMA buffers

PCI-SIG I/O Virtualization Working Group

- Activity towards standardizing natively sharable I/O devices
- IOV devices provide virtual interfaces, each independently assignable to VMs
- Standards for caching DMA translations at PCIe* device endpoints

*Other names and brands may be claimed as the property of others

Pass-through Model



- Pro: Higher Performance
- Pro: Rich Device Features
- Con: Limited Sharing
- Con: VM Migration Limits

See Session IVTS004 for more details on natively shareable I/O devices

Intel® VT: Learning More

Vector 3:
I/O Focus

Vector 2:
Platform Focus

Vector 1:
Processor Focus

VMM
Software
Evolution

Focus of session IVTS004

Standards for I/O-device sharing:

- Natively sharable I/O devices
- Endpoint DMA-translation caching

PCI-SIG

Focus of
Session
IVTS003

Infrastructure for I/O-device virtualization:

- DMA protection and remapping
- Interrupt filtering and remapping

VT-d

Focus of
Session
IVTS002

VT-x

VT-i

Establish foundation for virtualization in the Intel® 64 and Itanium® architectures...

- ... followed by on going evolution of support:
- Microarchitectural (e.g., lower VM entry/exit costs)
 - Architectural (e.g., extended page tables – EPT)

Software-only VMMs

- Binary translation
- Paravirtualization
- Device Emulation

Simpler and more **Secure** VMMs through foundation of virtualizable ISAs

Improved CPU and I/O virtualization **Performance** and **Functionality** as VMMs exploit infrastructure provided by VT-x, VT-i, VT-d

Past
No Hardware Support

Today

VMM software evolution over time with hardware support