# VMware and Hardware Assist Technology (Intel VT and AMD-V)

Jack Lo

Sr. Director, R&D

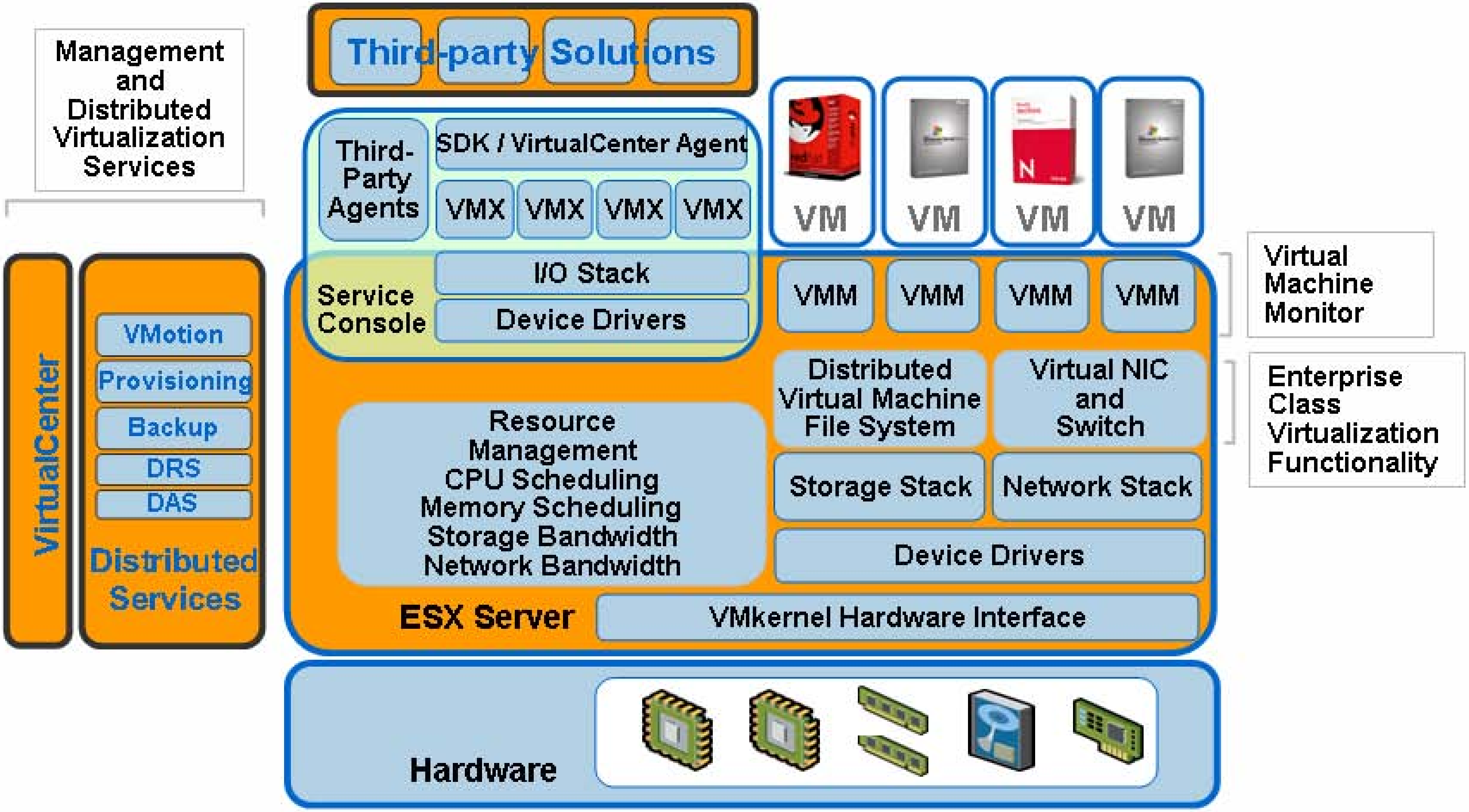
VMWORLD 2006

# Agenda

- CPU virtualization technology overview
  - Virtualizing the x86 architecture
- Hardware assist
  - First generation VT-x and AMD-V
  - Second generation HW assist
- CPU virtualization alternatives
  - VMware and paravirtualization

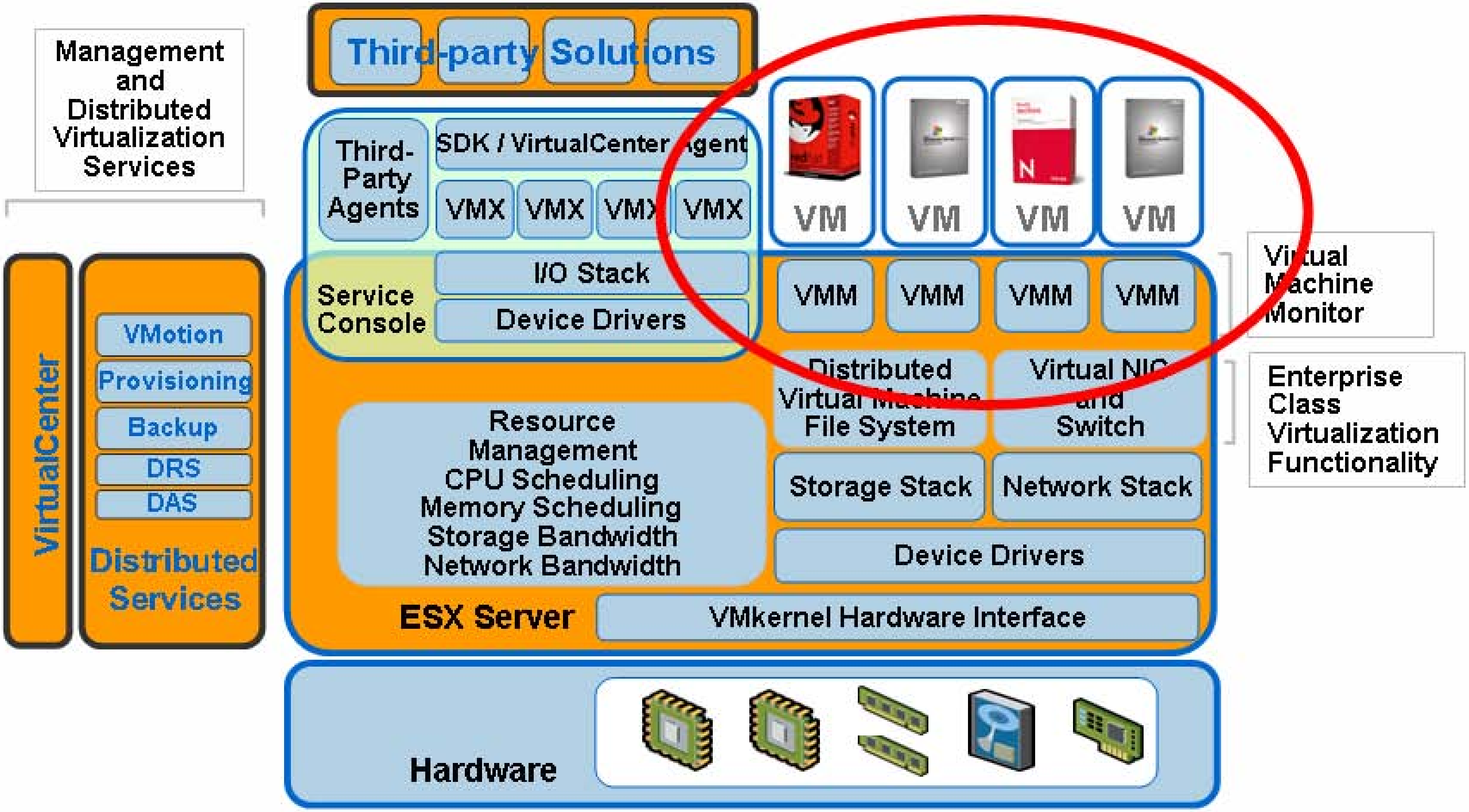
*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Full Virtualization Software Stack



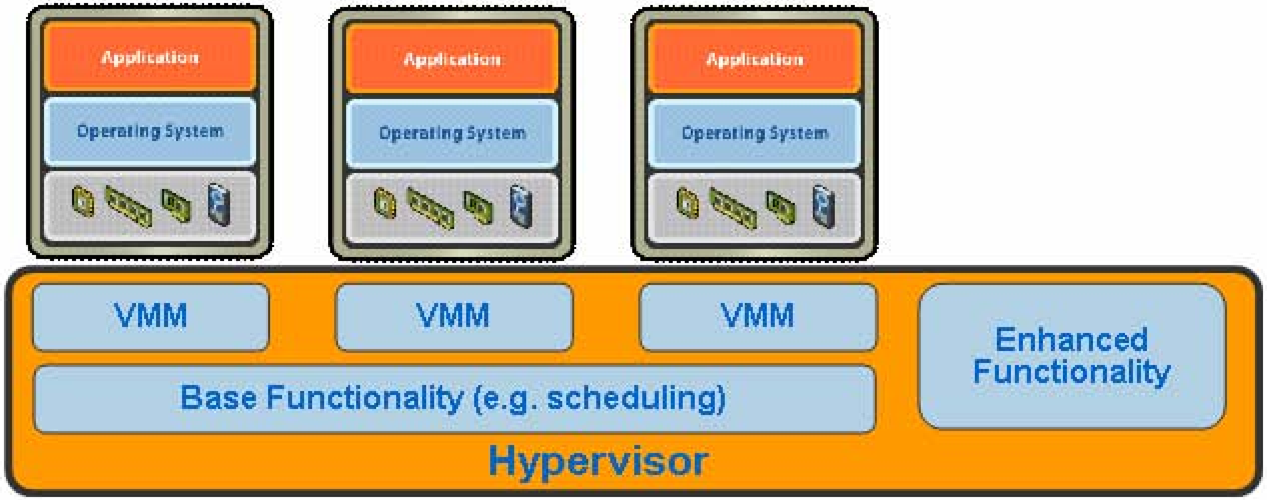
*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Full Virtualization Software Stack



*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Virtualization SW Technology



- Virtual Machine Monitor (VMM)
  - SW component that implements virtual machine hardware abstraction
  - Responsible for running the guest OS
- Hypervisor
  - Software responsible for hosting and managing virtual machines
  - Run directly on the hardware
  - Functionality varies greatly with architecture and implementation

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# CPU Virtualization

- Three components to classical virtualization techniques
- Many virtualization technologies focus on handling privileged instructions

| | |
|---|---|
| **Privileged instruction virtualization** | De-privileging or ring compression to handle privileged instructions |
| **Memory virtualization** | Memory partitioning and allocation of physical memory |
| **Device and I/O virtualization** | Routing I/O requests between virtual devices and physical hardware |

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# CPU Virtualization

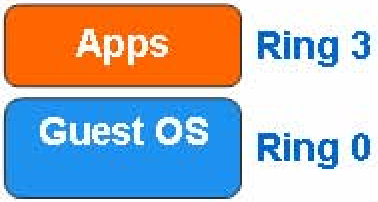
- Three components to classical virtualization techniques
- Many virtualization technologies focus on handling privileged instructions

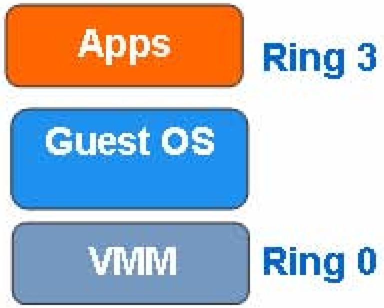
| | |
|---|---|
| **Privileged instruction virtualization** | De-privileging or ring compression to handle privileged instructions |
| **Memory virtualization** | Memory partitioning and allocation of physical memory |
| **Device and I/O virtualization** | Routing I/O requests between virtual devices and physical hardware |

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Handling Privileged Instructions

- In traditional systems
  - OS runs in privileged mode
  - OS “owns” the hardware
  - Application code has less privilege
- VMM needs highest privilege level for isolation and performance
- Traditional VMM relies on “ring compression” or “de-privileging”
  - Run privileged guest OS code at user-level
  - Privileged instructions trap, and emulated by VMM





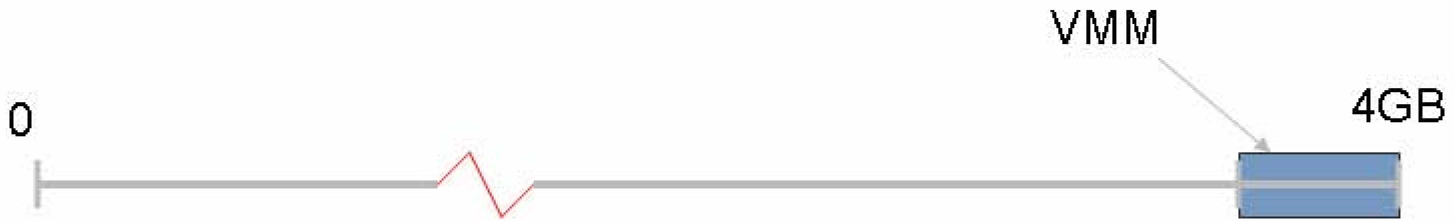
*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Handling Privileged Instructions for x86

- De-privileging not possible with x86!
  - Some privileged instructions have different semantics at user-level: “non-virtualizable instructions”
- VMware uses direct execution and binary translation (BT)
  - BT for handling privileged code
  - Direct execution of user-level code for performance
  - Any unmodified x86 OS can run in virtual machine
- Virtual machine monitor lives in the guest address space

# Protecting the VMM

- Need to protect VMM and ensure isolation
  - Protect virtual machines from each other
  - Protect VMM from virtual machines
- VMware traditionally relies on segmentation hardware to protect the VMM
  - VMM lives at top of guest address space
  - Segment limit checks catch writes to VMM area



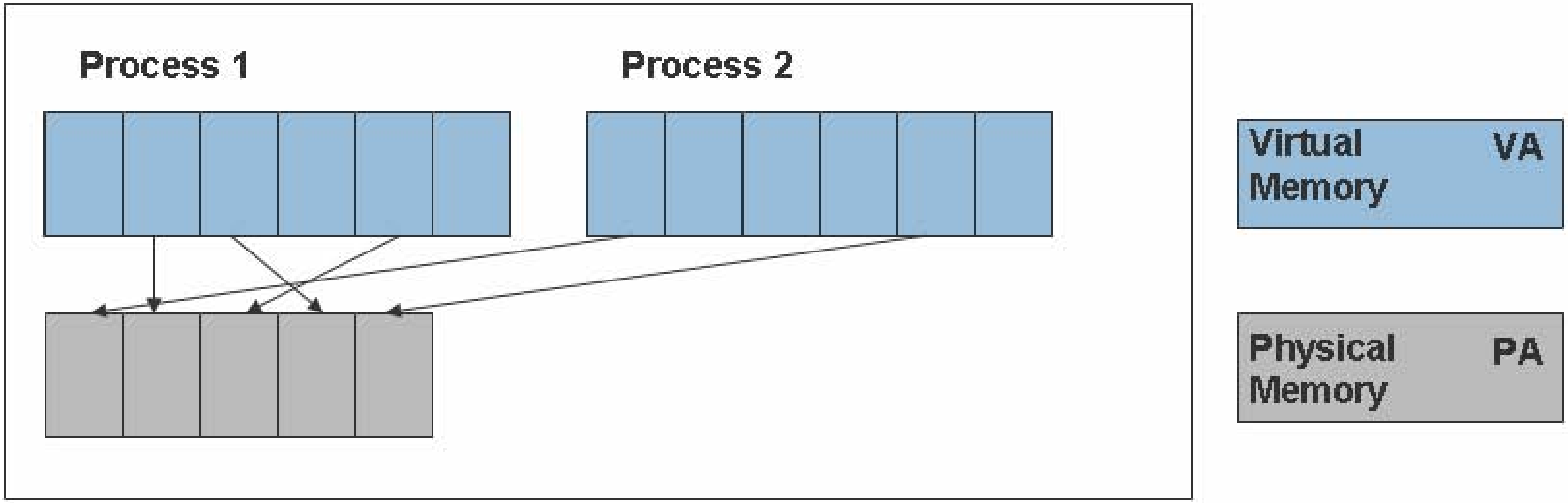
*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# CPU Virtualization

- Three components to classical virtualization techniques
- Many virtualization technologies focus on handling privileged instructions

| | |
|---|---|
| **Privileged instruction virtualization** | De-privileging or ring compression to handle privileged instructions |
| **Memory virtualization** | Memory partitioning and allocation of physical memory |
| **Device and I/O virtualization** | Routing I/O requests between virtual devices and physical hardware |

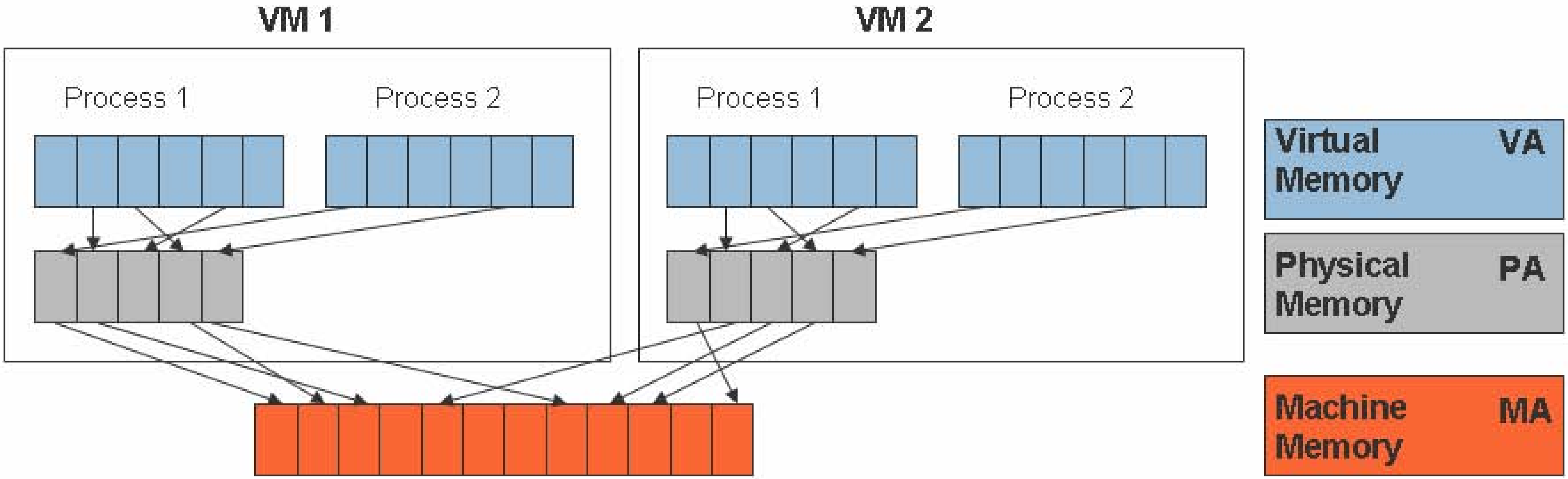
# Virtual Memory



- Modern operating systems provide virtual memory support
  - Applications see a contiguous address space that is not necessarily tied to underlying physical memory in the system
  - OS keeps mappings of virtual page numbers to physical page numbers
  - Mappings are stored in page tables
- CPU includes memory management unit (MMU) and TLB for virtual memory support

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

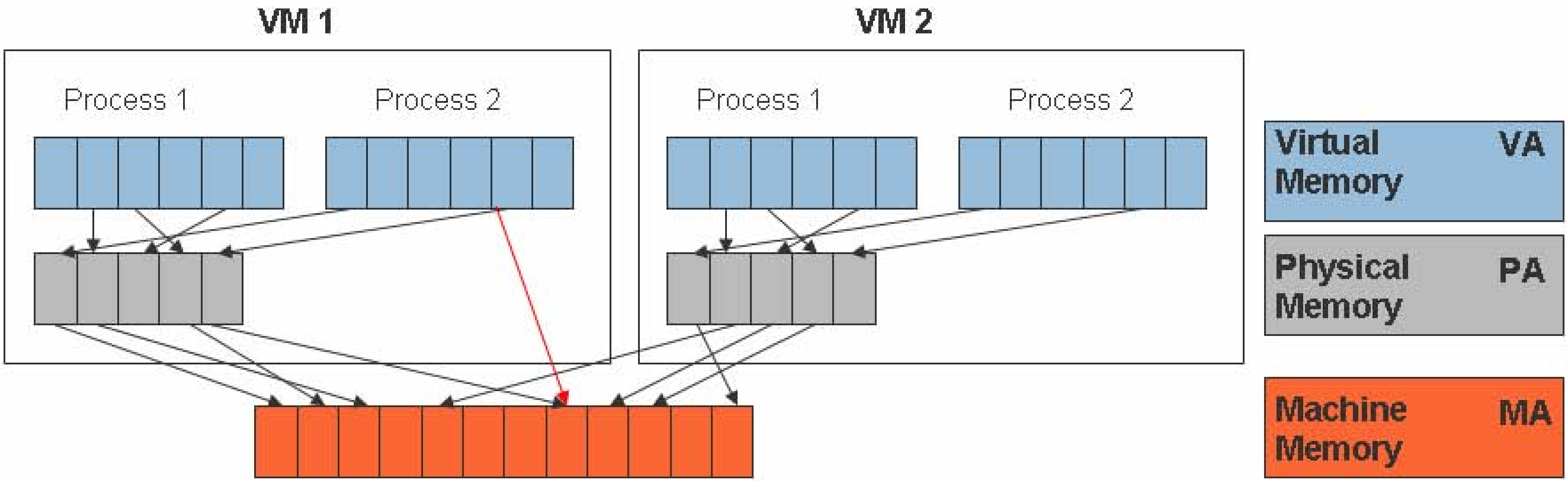
# Virtualizing Virtual Memory



- In order to run multiple virtual machines on a single system, another level of memory virtualization must be done
  - Guest OS still controls mapping of virtual address to physical address: **VA -> PA**
  - In virtualized world, guest OS cannot have direct access to machine memory
  - Each guest's physical memory is no longer the actual machine memory in system
- VMM maps guest physical memory to the actual machine memory: **PA -> MA**

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Virtualizing Virtual Memory: Shadow Page Tables



- VMM uses “shadow page tables” to accelerate the mappings
  - Directly map VA -> MA
  - Can avoid the two levels of translation on every access
  - Leverage TLB hardware for this VA -> MA mapping
  - When guest OS changes VA -> PA, the VMM updates the shadow page tables

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

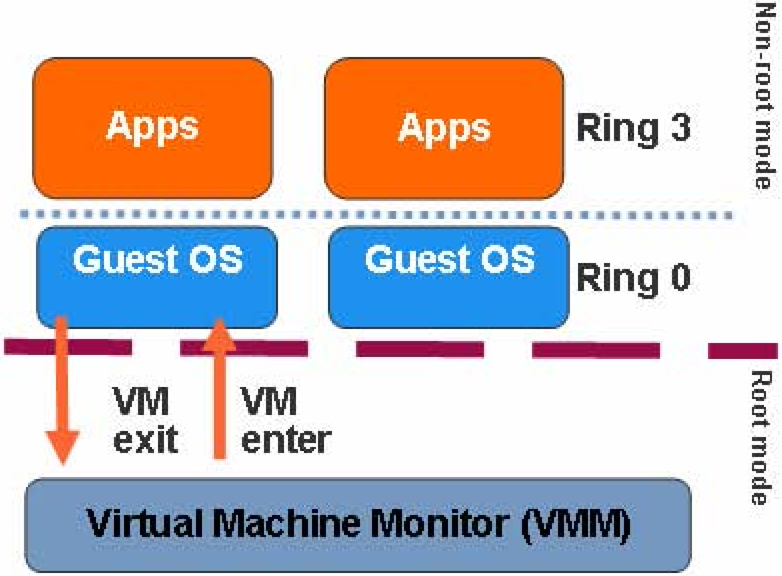
# Agenda

- CPU virtualization technology overview
  - Virtualizing the x86 architecture
- Hardware assist
  - First generation VT-x and AMD-V
  - Second generation HW assist
- CPU virtualization alternatives
  - VMware and paravirtualization

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Intel VT-x / AMD-V Overview

- CPU vendors are embracing virtualization
  - Intel Virtualization Technology (VT-x)
  - AMD-V
- Key feature is new CPU execution mode (root mode)
  - VMM executes in root mode
  - Allows x86 virtualization without binary translation or paravirtualization
  - Guest state stored in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Block (AMD-V)



*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# 1st Generation Hardware Assist

- Initial VT-x/AMD-V hardware targets privileged instructions
  - HW is an enabling technology that makes it easier to write a functional VMM
  - Alternative to using binary translation
- Initial hardware does not guarantee highest performance virtualization
  - VMware binary translation outperforms VT-x/AMD-V

| | Current VT-x/AMD-V |
|---|---|
| **Privileged instructions** | Yes |
| **Memory virtualization** | No |
| **Device and I/O virtualization** | No |

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Challenges of Virtualizing x86-64

- Initial AMD64 architecture did not include segmentation in 64-bit mode
  - Segmentation also missing from EM64T

***How do we protect the VMM?***

- 64-bit guest support requires additional hardware assistance
  - Segment limit checks available in 64-bit mode on newer AMD processors
  - VT-x can be used to protect the VMM on EM64T
    - Requires trap-and-emulate approach instead of BT

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

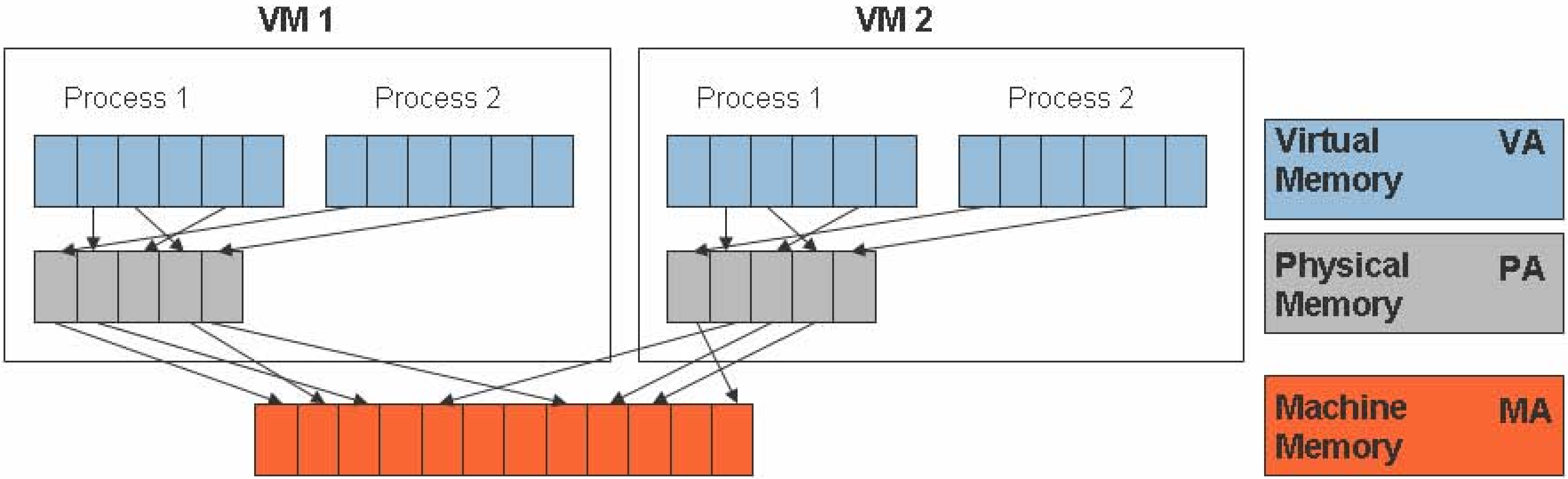
# Future Hardware Assist

- Both AMD and Intel have announced roadmap of additional hardware support
  - Memory virtualization (Nested paging, Extended Page Tables)
  - Device and I/O virtualization (VT-d, IOMMU)

| | **HW Solution** |
|---|---|
| **Privileged instructions** | VT-X / AMD-V |
| **Memory virtualization** | EPT / NPT |
| **Device and I/O virtualization** | Intelligent devices, IOMMU / VT-d |

***These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.***

# Nested Paging / Extended Page Tables



- Hardware support for memory virtualization is on the way
  - AMD: Nested Paging / Nested Page Tables (NPT)
  - Intel: Extended Page Tables (EPT)
- Conceptually, NPT and EPT are identical
  - Two sets of page tables exist: VA -> PA and PA -> MA
  - Processor HW does page walk for both VA -> PA and PA -> MA

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

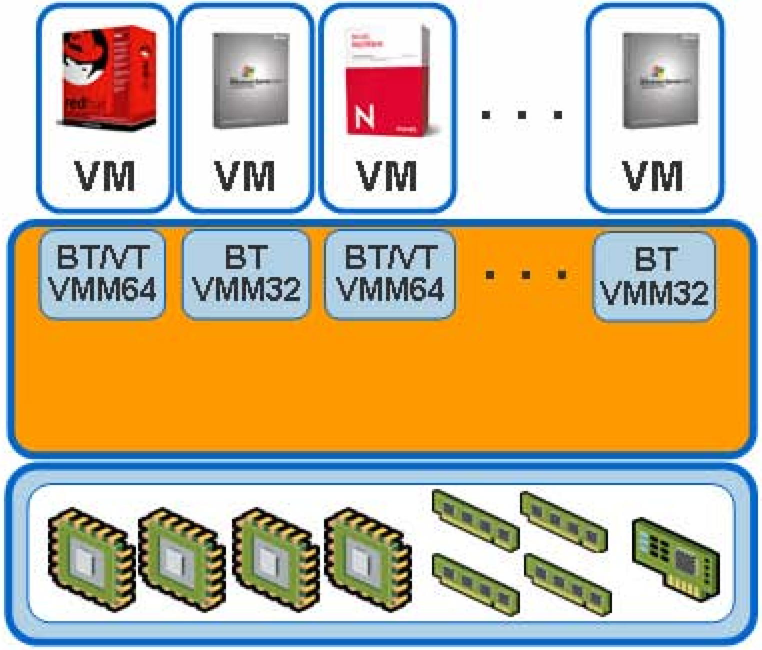
# Benefits of NPT/EPT

- Performance
  - Compute-intensive workloads already run well with binary translation/direct execution
  - NPT/EPT will provide noticeable performance improvement for workloads with MMU overheads
  - Hardware addresses the performance overheads due to virtualizing the page tables
  - With NPT/EPT, even more workloads become candidates for virtualization
- Reducing memory consumption
  - Shadow page tables consume additional system memory
  - Use of NPT/EPT will reduce “overhead memory”
- Today, VMware uses HW assist in very limited cases
  - NPT/EPT provide motivation to use HW assist much more broadly

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Flexible VMM Architecture

- Flexible "multi-mode" VMM architecture
  - Separate VMM per virtual machine
- Select mode that achieves best workload-specific performance based on CPU support
- Today
  - 32-bit: BT VMM
  - 64-bit: BT or VT-x
- Tomorrow
  - 32-bit: BT VMM or AMD-V + NPT or VT-x + EPT
  - 64-bit: BT or VT-x or AMD-V + NPT or VT-x + EPT
- Same VMM architecture for ESX Server, Player, Server, Workstation and ACE



*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Agenda

- CPU virtualization technology overview
  - Virtualizing the x86 architecture
- Hardware assist
  - First generation VT-x and AMD-V
  - Second generation HW assist
- CPU virtualization alternatives
  - VMware and paravirtualization

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# CPU Virtualization Alternatives: OS Assist

- Three alternatives for handling non-virtualizable instructions
  - Binary translation
  - Hardware assist (first generation)
  - OS assist or paravirtualization

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | |
| Performance | **Good** | **Average** | |
| VMM sophistication | **High** | **Average** | |

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Paravirtualization

- Paravirtualization can also address CPU virtualization
  - Modify the guest OS to remove non-virtualizable instructions
  - Export a simpler architecture to OS

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | |
| Performance | **Good** | **Average** | |
| VMM sophistication | **High** | **Average** | |

***These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.***

# Paravirtualization

- Paravirtualization can also address CPU virtualization
  - Modify the guest OS to remove non-virtualizable instructions
  - Export a simpler architecture to OS
  - Cannot support unmodified guest OSes (e.g., Windows 2000/XP)

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | **Poor** |
| Performance | **Good** | **Average** | |
| VMM sophistication | **High** | **Average** | |

***These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.***

# Paravirtualization

- Paravirtualization can also address CPU virtualization
  - Modify the guest OS to remove non-virtualizable instructions
  - Export a simpler architecture to OS
  - Cannot support unmodified guest OSes (e.g., Windows 2000/XP)
  - Higher performance possible
  - Paravirtualization not limited to CPU virtualization

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | **Poor** |
| Performance | **Good** | **Average** | **Excellent** |
| VMM sophistication | **High** | **Average** | |

# Paravirtualization

- Paravirtualization can also address CPU virtualization
  - Modify the guest OS to remove non-virtualizable instructions
  - Export a simpler architecture to OS
  - Cannot support unmodified guest OSes (e.g., Windows 2000/XP)
  - Higher performance possible
  - Paravirtualization not limited to CPU virtualization
  - Relatively easy to add paravirtualization support; very difficult to add binary translation

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | **Poor** |
| Performance | **Good** | **Average** | **Excellent** |
| VMM sophistication | **High** | **Average** | **Average** |

# Paravirtualization Challenges

- XenLinux paravirtualization approach unsuitable for enterprise use
  - Relies on separate kernel for native and in virtual machine
  - Guest OS and hypervisor tightly coupled (data structure dependencies)
  - Tight coupling inhibits compatibility
  - Changes to the guest OS are invasive

- VMware's proposal: Virtual Machine Interface API
  - Proof-of-concept that high-performance paravirtualization possible with a maintainable interface
  - VMI provides maintainability & stability
  - API supports low-level and higher-level interfaces
  - Allows same kernel to run natively and in a paravirtualized virtual machine: "transparent paravirtualization"
  - Allows for replacement of hypervisors without a guest recompile
  - Preserve key virtualization functionality: page sharing, VMotion, etc.

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

## Paravirt Ops

- Great progress is happening in the Linux kernel community
  - Paravirtualization interfaces were discussed at the recent kernel summit and Ottawa Linux Symposium
  - Agreement to have a team of developers work on a paravirtualization interface for Linux
- Paravirt ops
  - Source-level paravirtualization interface that supports multiple hypervisors
  - Spearheaded by Rusty Russell from IBM Linux Technology Center
  - Developers from VMware, IBM LTC, XenSource, Red Hat
  - http://ozlabs.org/~rusty/paravirt/

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Improved Paravirtualization

- Great progress is happening in the Linux kernel community
  - Paravirtualization interfaces were discussed at the recent kernel summit and Ottawa Linux Symposium
  - Agreement to have a team of developers work on a paravirtualization interface Source-level paravirtualization interface that supports multiple hypervisors
- Paravirt ops
  - Developers from VMware, IBM LTC, XenSource, Red Hat
  - http://ozlabs.org/~rusty/paravirt/
  - Improves compatibility

| | Binary Translation | Current HW Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | **Good** |
| Performance | **Good** | **Average** | **Excellent** |
| VMM sophistication | **High** | **Average** | **Average** |

***These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.***

# Impact of NPT/EPT

- Role of paravirtualization changes when NPT/EPT is available
  - NPT/EPT addresses MMU virtualization overheads
  - Paravirtualization becomes more I/O-focused

| | Binary Translation | Hardware Assist | Paravirtualization |
|---|---|---|---|
| Compatibility | **Excellent** | **Excellent** | **Good** |
| Performance | **Good** | **Excellent** | **Excellent** |
| VMM sophistication | **High** | **Average** | **Average** |

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

## Related Information

- Other talks
  - TAC9727: VMware VMI Paravirtualization
    Wednesday, 4:45 – 5:45, Room 411
  - TAC0080: I/O Architectures for Virtualization
    Tuesday, 4:45 – 5:45, Room 411
- Other references
  - A Comparison of Software and Hardware Techniques for x86 Virtualization, ASPLOS XII, Oct. 2006, Adams and Agesen
    http://www.vmware.com/pdf/asplos235_adams.pdf

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Summary

- VMware provides flexible architecture to support emerging virtualization technologies
  - > Multi-mode VMM utilizes binary translation, hardware assist and paravirtualization
  - > Select best operating mode for the workload
- Hardware assist technology will continue to mature
  - > Continue to broaden the set of workloads that can be virtualized

*These features are representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.*

# Presentation Download

Please remember to complete your
**session evaluation form**
and return it to the room monitors
as you exit the session

The presentation for this session can be downloaded at
**http://www.vmware.com/vmtn/vmworld/sessions/**

Enter the following to download (case-sensitive):

**Username: cbv_rep**
**Password: cbvfor9v9r**

Some or all of the features in this document may be representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.

VMWORLD 2006