Hardware-assisted Virtualization

15-612 Operating System Practicum

Carnegie Mellon University

Pratik Shah (pcshah)
Rohan Patil (rspatil)

Agenda

- Introduction to VT-x
- CPU virtualization with VT-x
  - VMX
  - VMX Transitions
  - Virtual Machine Control Structure (VMCS)
- MMU Virtualization with VT-x
  - Virtual Processor IDentifier (VPID)
  - Sidebar: Virtualizing memory in software
  - Nested / Extended Page Tables (EPT)
- References
- Q & A

VT-x

- Intel Vanderpool Technology, referred to as VT-x, represents Intel's virtualization technology on the x86 platform.

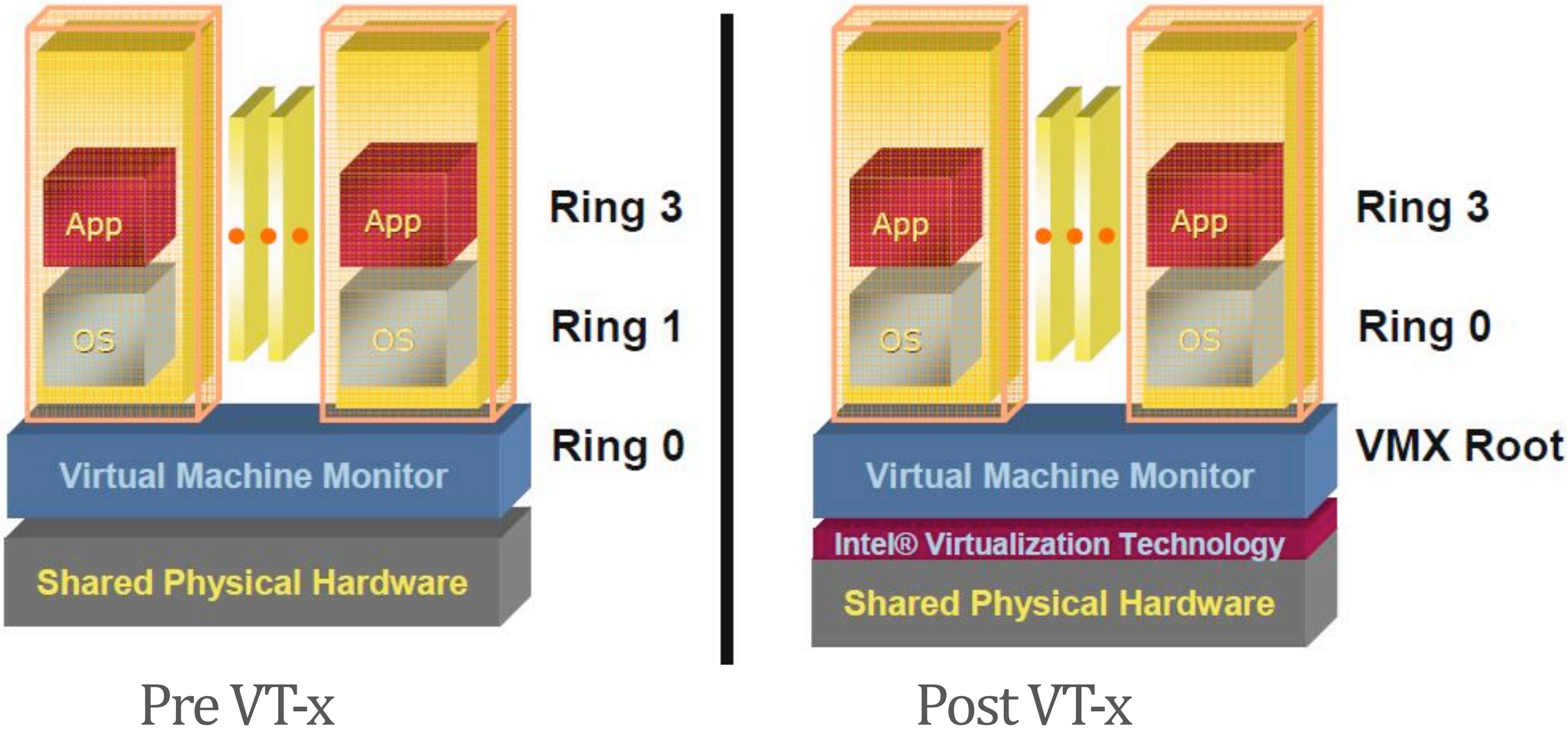
VT-x : Motivation

- To solve the problem that the x86 instructions architecture cannot be virtualized.
- Simplify VMM software by closing virtualization holes by design.
  - Ring Compression
  - Non-trapping instructions
  - Excessive trapping
- Eliminate need for software virtualization (i.e paravirtualization, binary translation).

CPU Virtualization with VT-x

VMX

- Virtual Machine Extensions define processor-level support for virtual machines on the x86 platform by a new form of operation called VMX operation.
- Kinds of VMX operation:
  - **root:** VMM runs in VMX root operation
  - **non-root:** Guest runs in VMX non-root operation
- Eliminate de-privileging of Ring for guest OS.

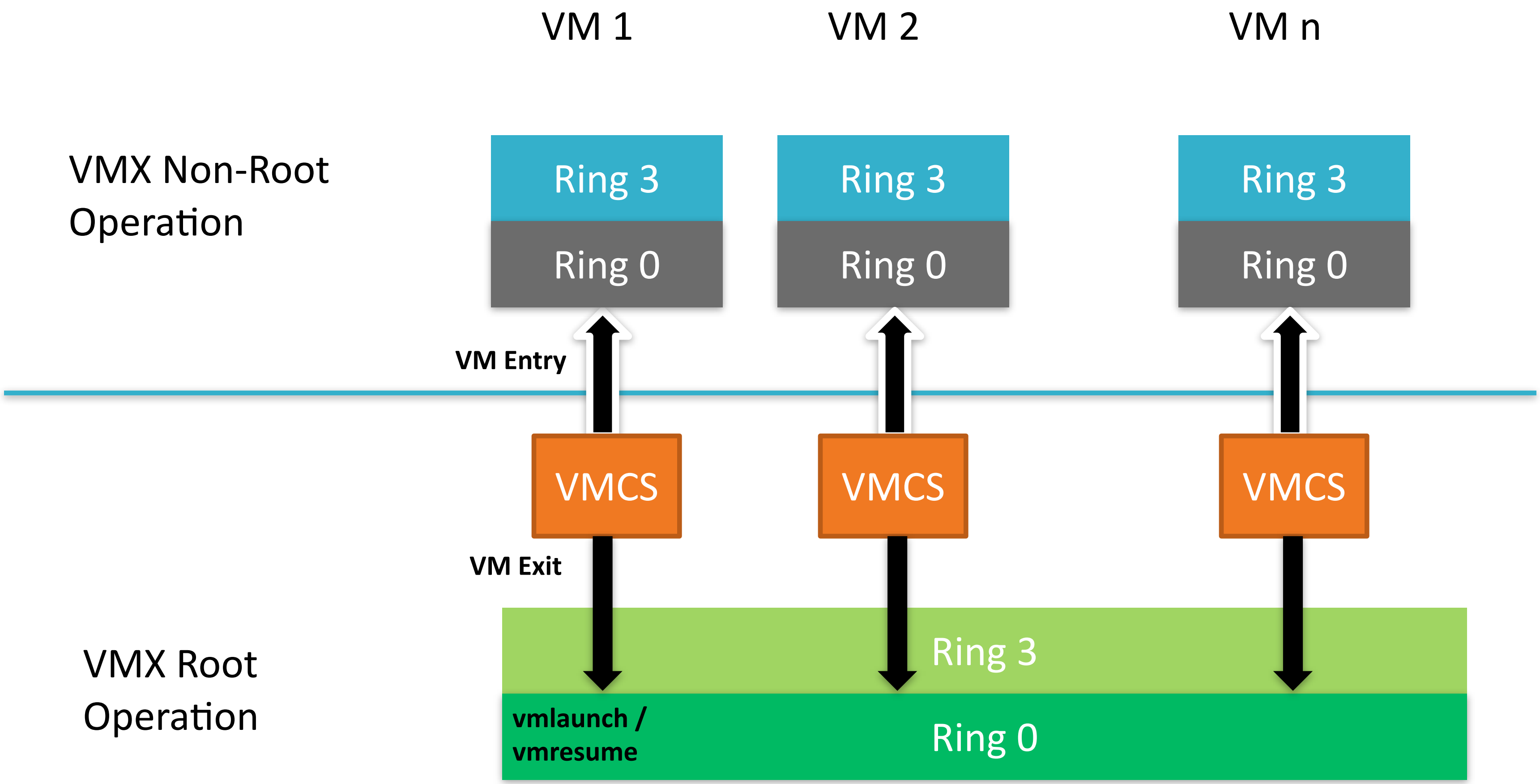


VMM ring de-privileging of guest OS	VMM executes in VMX root-mode
Guest OS aware its not at Ring 0	Guest OS de-privileging eliminated
	Guest OS runs directly on hardware

Source: [2]

VMX Transitions

- Transitions between VMX root operation and VMX non-root operation.
- Kinds of VMX transitions:
    - **VM Entry:** Transitions into VMX non-root operation.
    - **VM Exit:** Transitions from VMX non-root operation to VMX root operation.
- Registers and address space swapped in one atomic operation.



VMX Transitions

VMCS: VM Control Structure

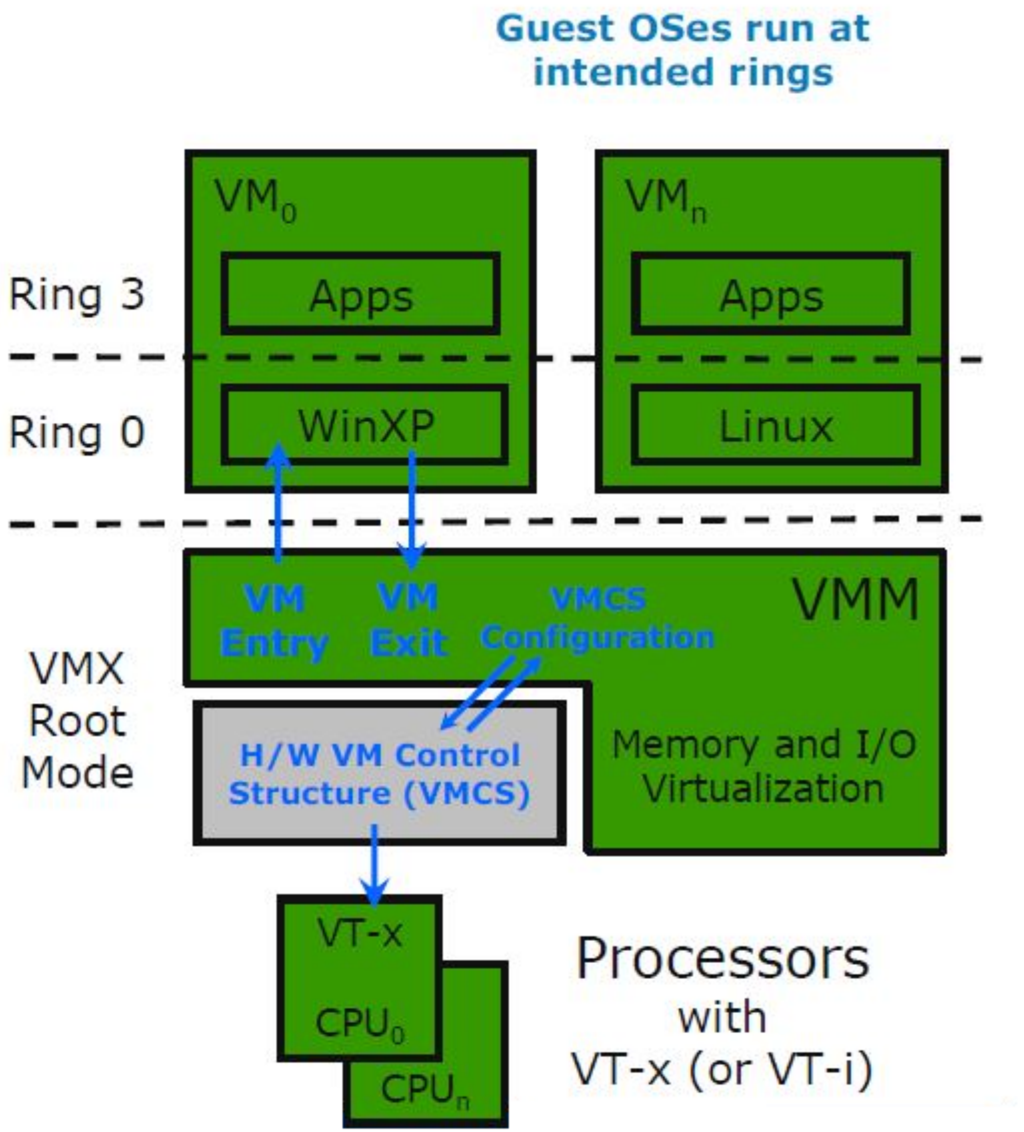
- Data structure to manage VMX non-root operation and VMX transitions.
- Specifies guest OS state.
- Configured by VMM.
- Controls when VM exits occur.

VMCS: VM Control Structure

The VMCS consists of six logical groups:

- **Guest-state area:** Processor state saved into the guest-state area on VM exits and loaded on VM entries.
- **Host-state area:** Processor state loaded from the host-state area on VM exits.
- **VM-execution control fields:** Fields controlling processor operation in VMX non-root operation.
- **VM-exit control fields:** Fields that control VM exits.
- **VM-entry control fields:** Fields that control VM entries.
- **VM-exit information fields:** Read-only fields to receive information on VM exits describing the cause and the nature of the VM exit.

CPU Virtualization with VT-x



Source: [2]

MMU Virtualization with VT-x

VPID: Motivation

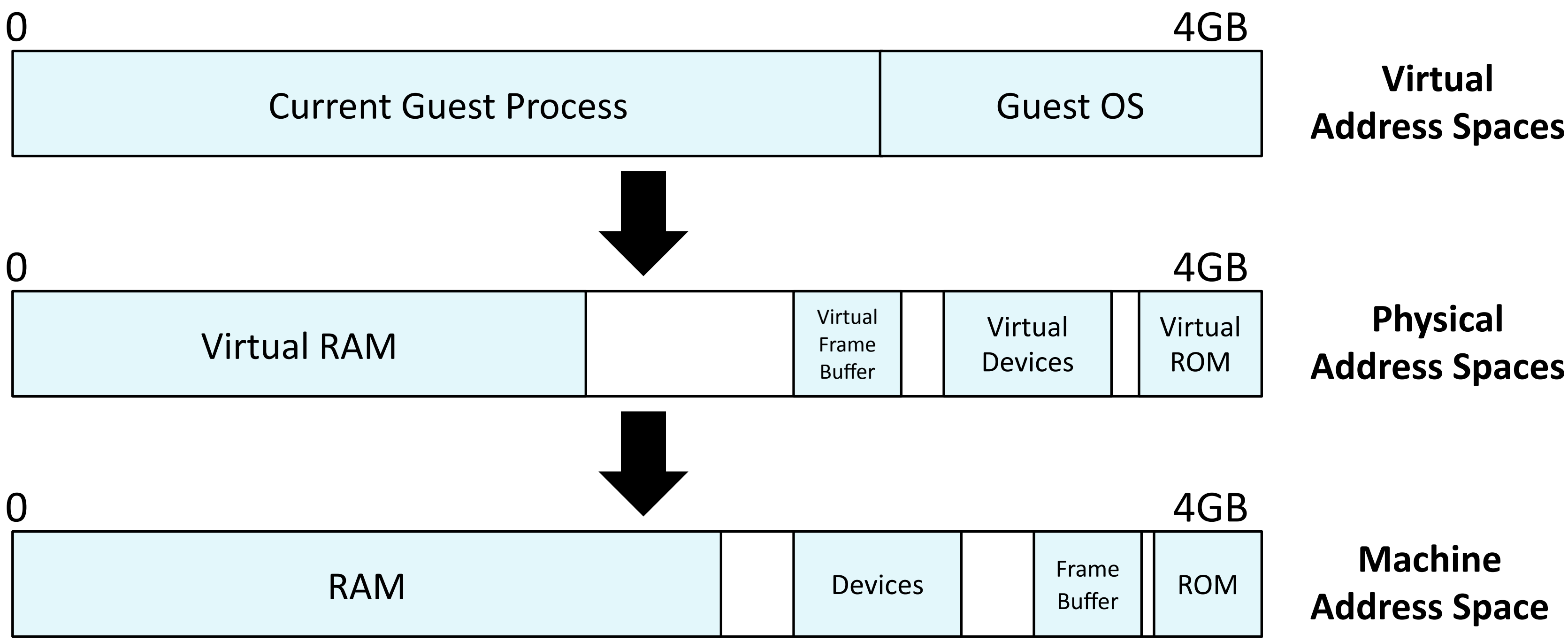
- First generation VT-x forces TLB flush on each VMX transition.
- Performance loss on all VM exits.
- Performance loss on most VM entries
  - Guest page tables not modified always
- Better VMM software control of TLB flushes is beneficial.

VPID: Virtual Processor Identifier

- 16-bit virtual-processor-ID field in the VMCS.
- Cached linear translations tagged with VPID value.
- No flush of TLBs on VM entry or VM exit if VPID active.
- TLB entries of different virtual machines can all co-exist in the TLB.

Virtualizing Memory in Software

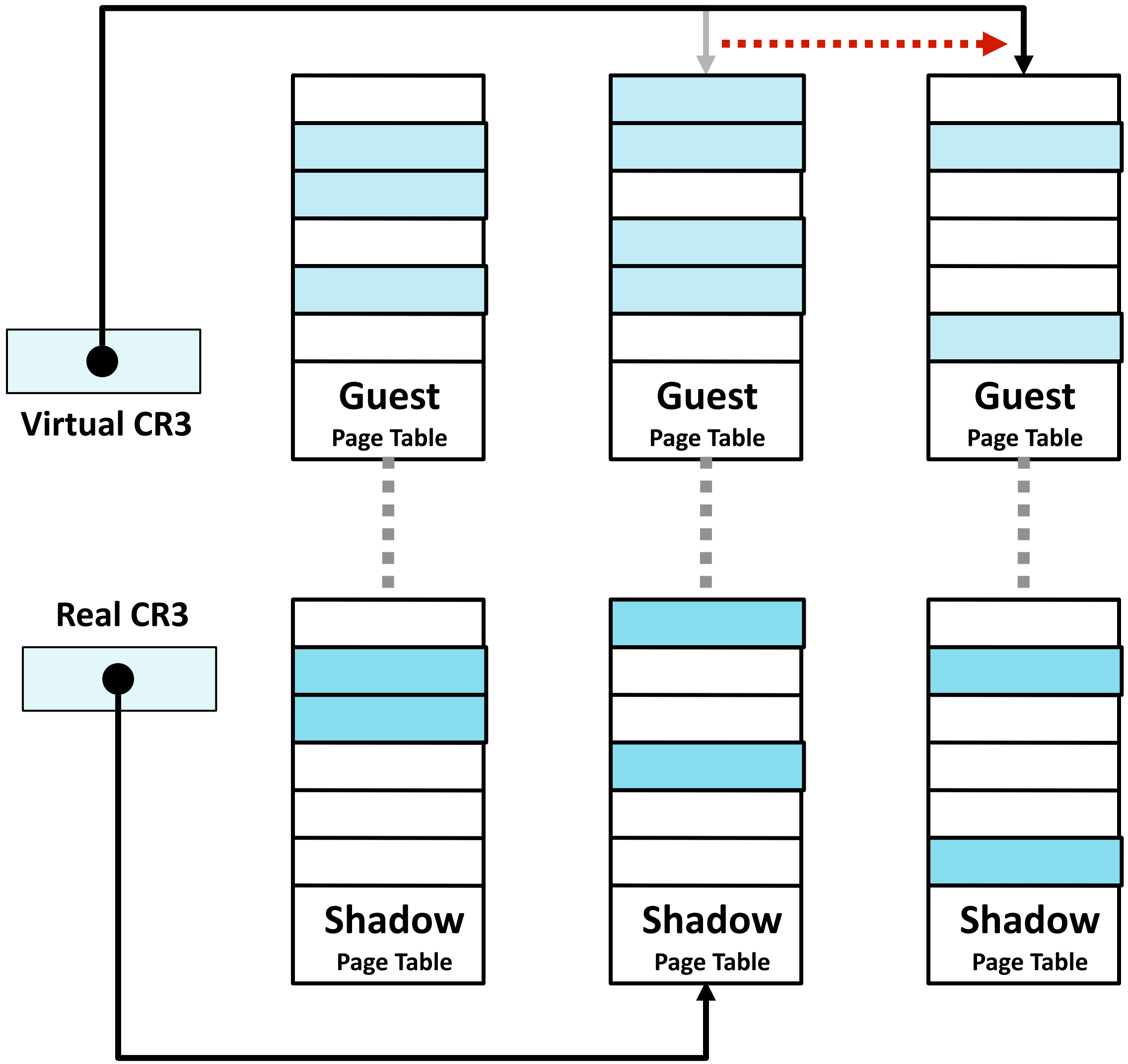
- Three abstractions of memory:



Source: [6]

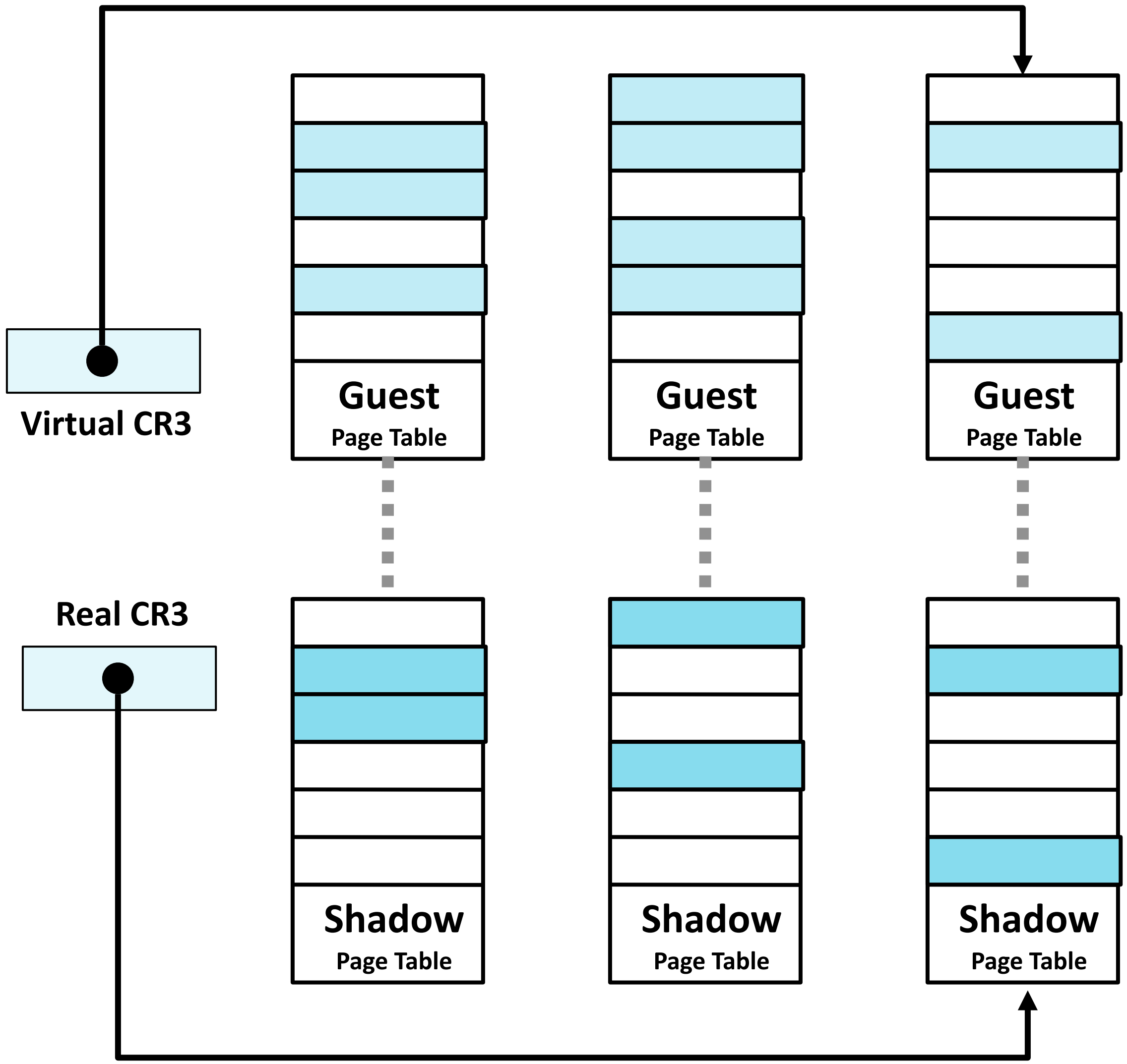
Shadow Page Tables

- VMM maintains shadow page tables that map guest-virtual pages directly to machine pages.
- Guest modifications to V->P tables synced to VMM V->M shadow page tables.
  - Guest OS page tables marked as read-only.
  - Modifications of page tables by guest OS -> trapped to VMM.
  - Shadow page tables synced to the guest OS tables



Set CR3 by guest OS (1)

Source: [6]



Set CR3 by guest OS (2)

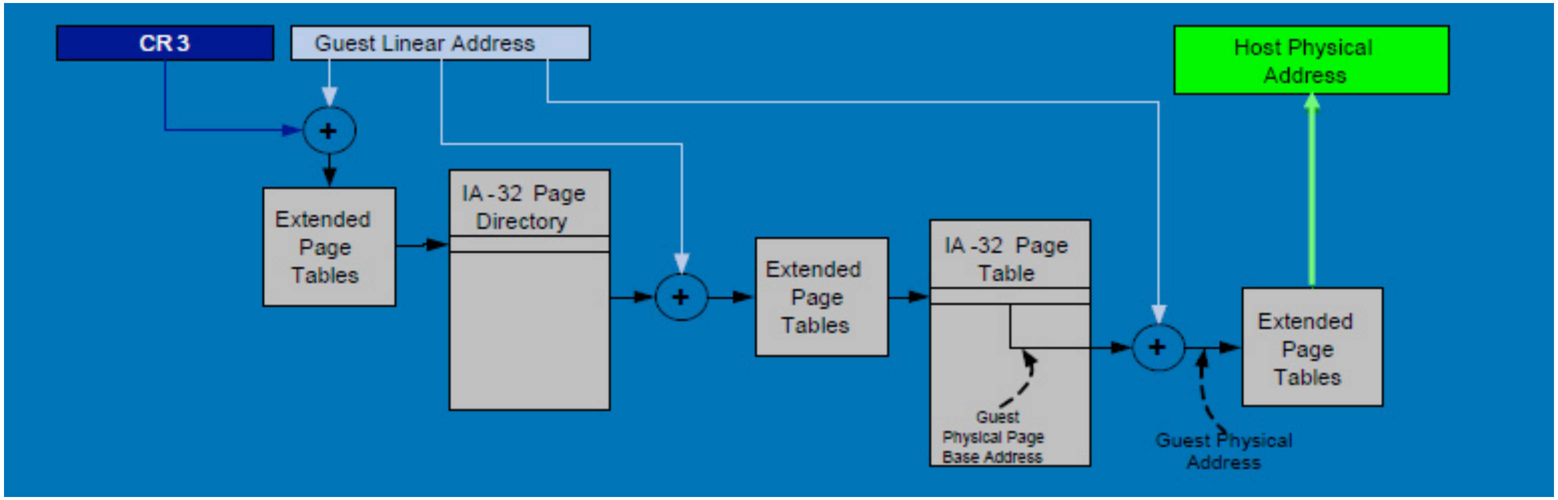
Source: [6]

Drawbacks: Shadow Page Tables

- Maintaining consistency between guest page tables and shadow page tables leads to an overhead: VMM traps
- Loss of performance due to TLB flush on every “world-switch”.
- Memory overhead due to shadow copying of guest page tables.

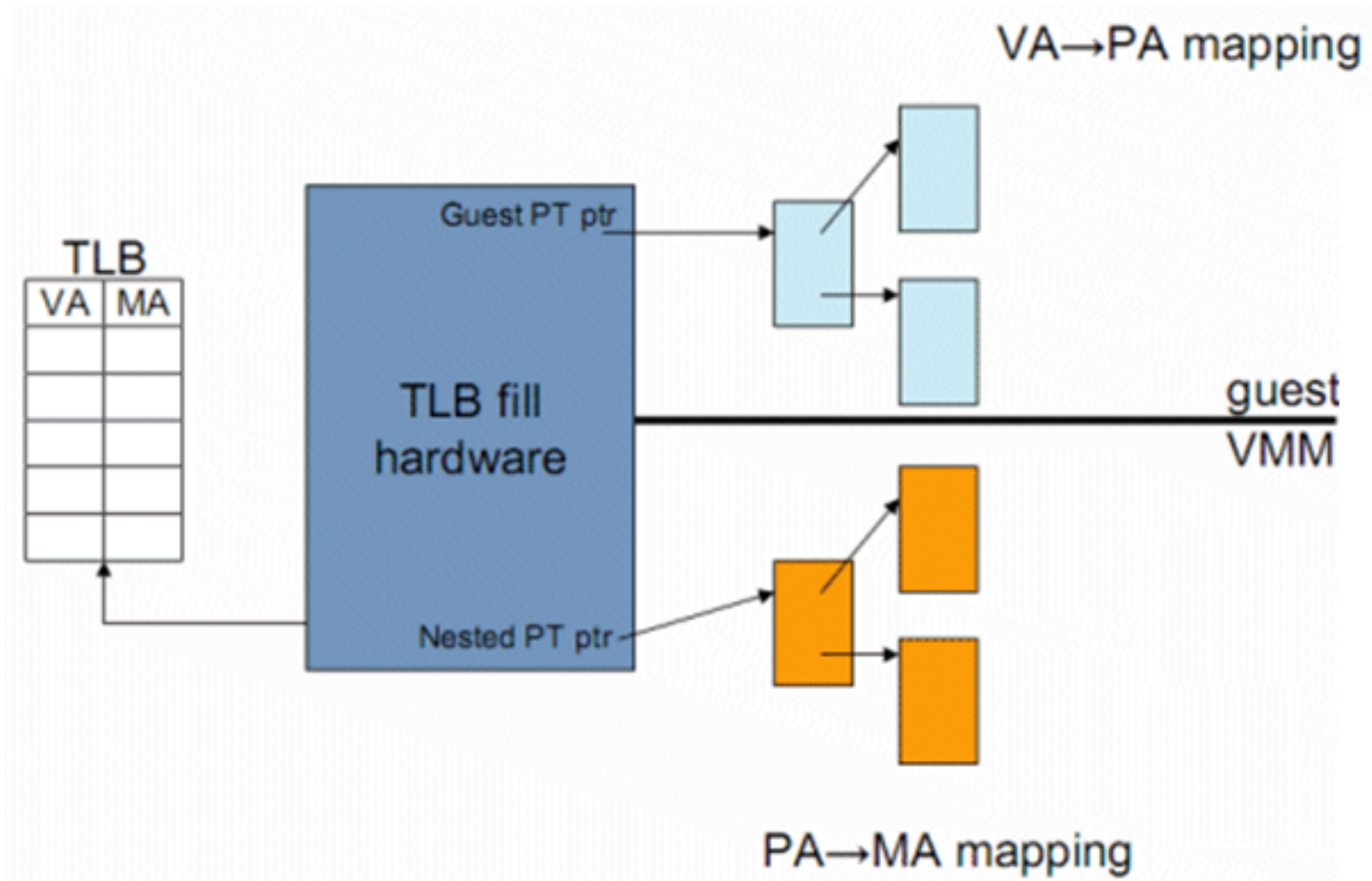
Nested / Extended Page Tables

- Extended page-table mechanism (EPT) used to support the virtualization of physical memory.
- Translates the guest-physical addresses used in VMX non-root operation.
- Guest-physical addresses are translated by traversing a set of EPT paging structures to produce physical addresses that are used to access memory.



Nested / Extended Page Tables

Source: [2]



Nested / Extended Page Tables

Source: [4]

Advantages: EPT

- Simplified VMM design.
- Guest page table modifications need not be trapped, hence VM exits reduced.
- Reduced memory footprint compared to shadow page table algorithms.

Disadvantages: EPT

- TLB miss is very costly since guest-physical address to machine address needs an extra EPT walk for each stage of guest-virtual address translation.

References

1. Intel 64 and IA-32 Architectures Software Developer's Manual (Volume 3C, Part 3) http://download.intel.com/products/processor/manual/326019.pdf
2. Intel Virtualization Technology Processor Virtualization Extensions and Intel Trusted execution Technology repo.meh.or.id/Todo/virtualization.pdf
3. Intel Virtualization Technology Primer by Rich Uhlig software.intel.com/file/26677
4. Hardware Virtualization blogs by Johan De Gelas at AnandTech http://www.anandtech.com/show/2480/9 http://www.anandtech.com/show/2480/10
5. Performance Evaluation of Intel EPT Hardware Assist http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf
6. Memory Virtualization by Scott Devine, VMware Labs labs.vmware.com/download/46/

Questions

Thank You