





# Virtualization of CPU, Memory, and I/O Devices

- 
- To support virtualization, processors such as the x86 employ a special running mode and instructions, known as hardware-assisted virtualization.
  - In this way, the VMM and guest OS run in different modes and all sensitive instructions of the guest OS and its applications are trapped in the VMM.
  - To save processor states, mode switching is completed by hardware.

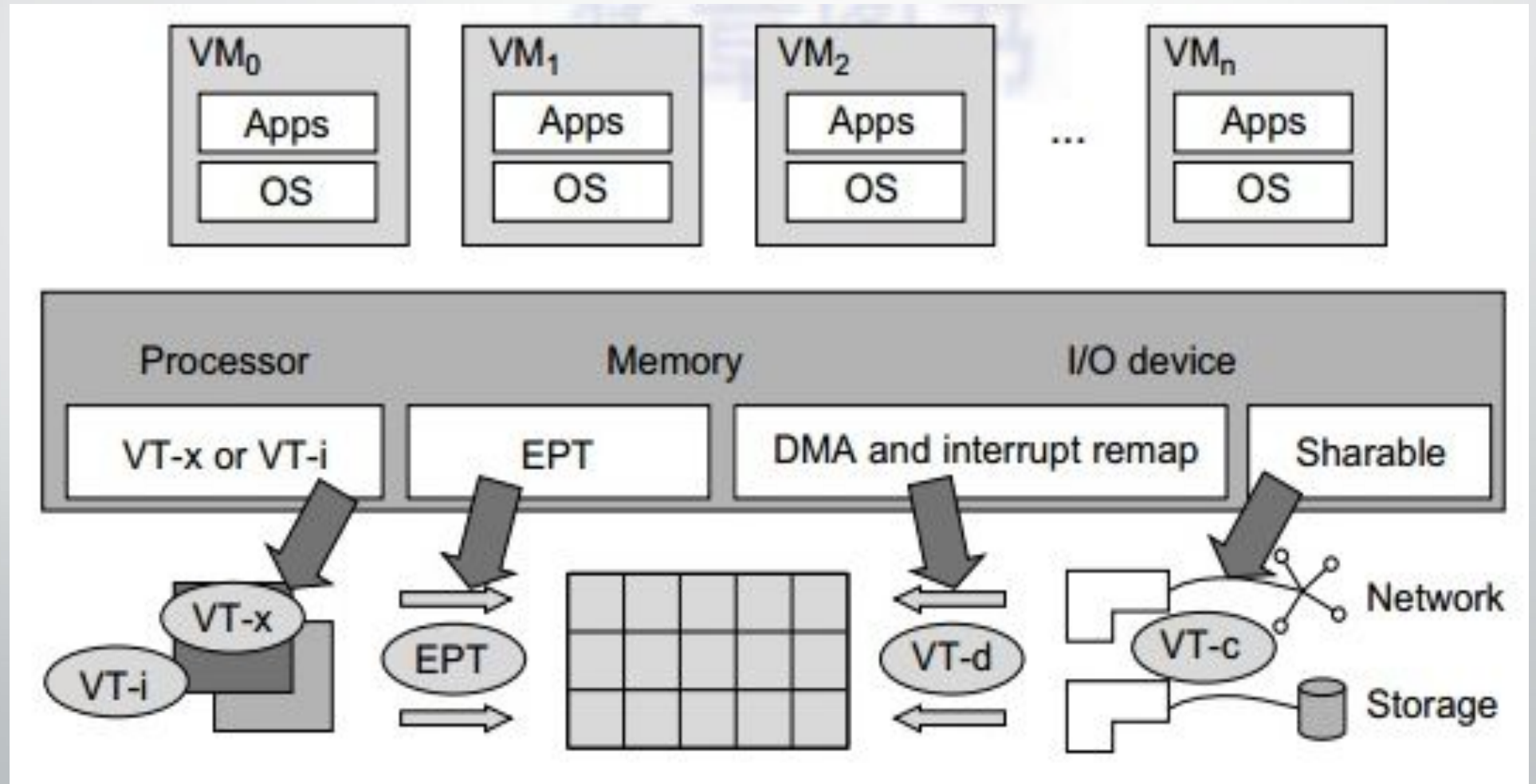



# Hardware Support for Virtualization

- Modern operating systems and processors permit multiple processes to run simultaneously.
- If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash.
- Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware.
- Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions.

- 
- The VMware Workstation is a VM software suite for x86 and x86-64 computers.
  - This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these VMs simultaneously with the host operating system.


# Hardware Support for Virtualization in the Intel x86 Processor




- 
- The above diagram provides an overview of Intel's full virtualization techniques. For processor virtualization, Intel offers the VT-x or VT-i technique
  - VT-x adds a privileged mode (VMX Root Mode) and some instructions to processors. This enhancement traps all sensitive instructions in the VMM automatically.
  - For memory virtualization, Intel offers the EPT, which translates the virtual address to the machine's physical addresses to improve performance.
  - For I/O virtualization, Intel implements VT-d and VT-c to support this.

# CPU Virtualization

- A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode.
- Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency.
- The critical instructions are divided into three categories: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions.

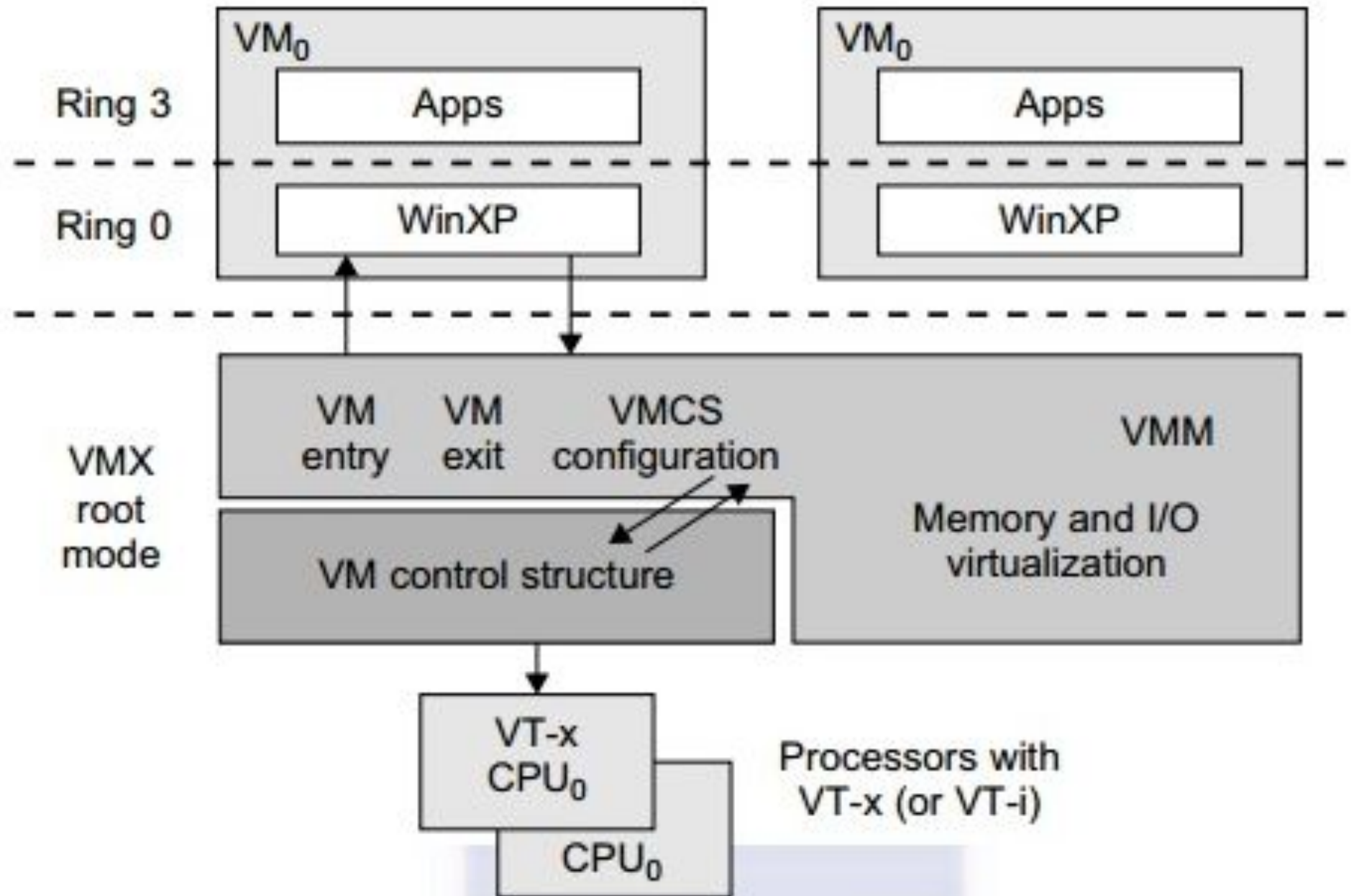
- 
- Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode.
  - Control-sensitive instructions attempt to change the configuration of resources used.
  - Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.





- 
- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
  - When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM.


# Intel Hardware-Assisted CPU Virtualization

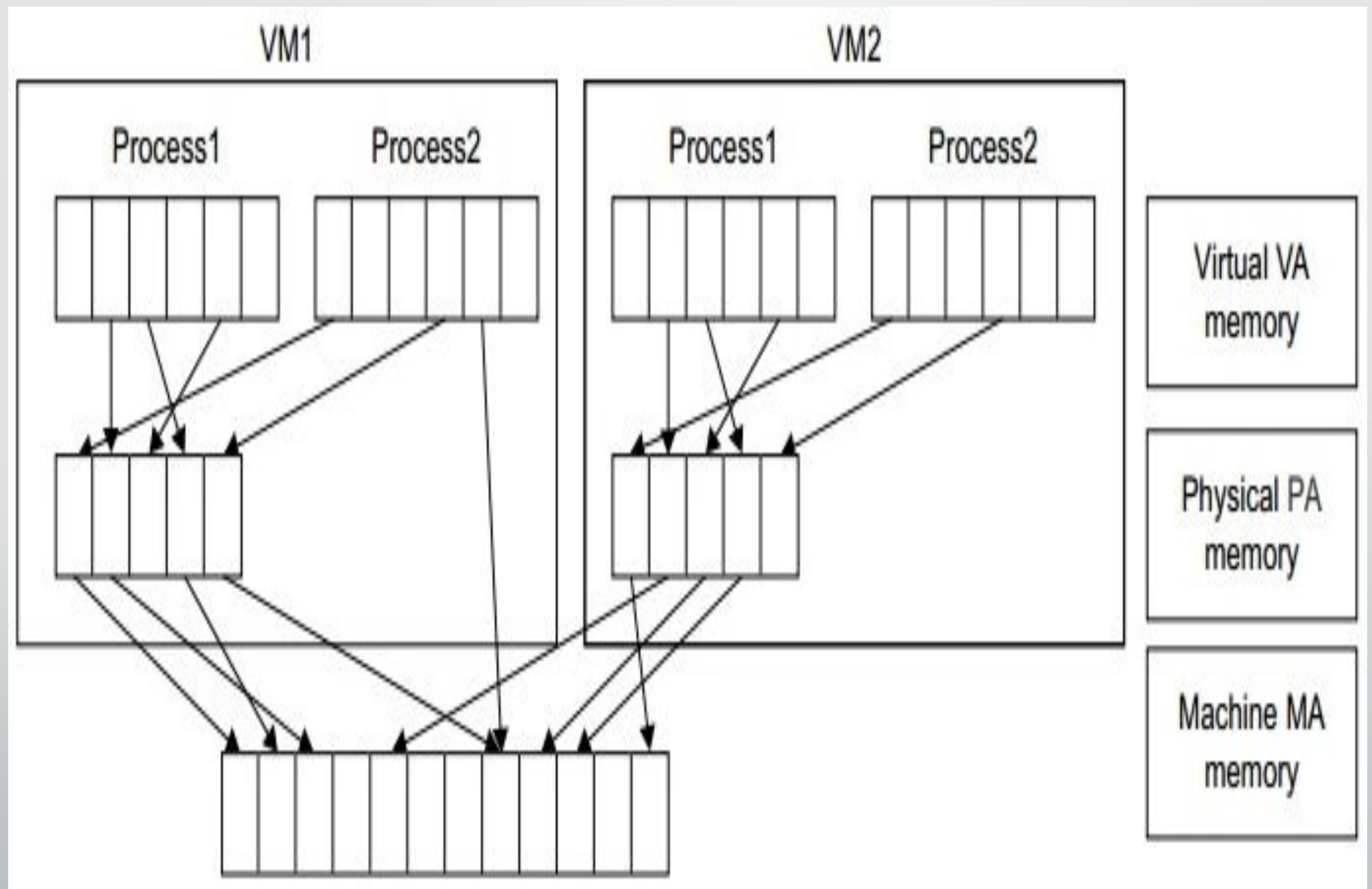
- Intel's VT-x technology is an example of hardware-assisted virtualization
- Intel calls the privilege level of x86 processors the VMX Root Mode.

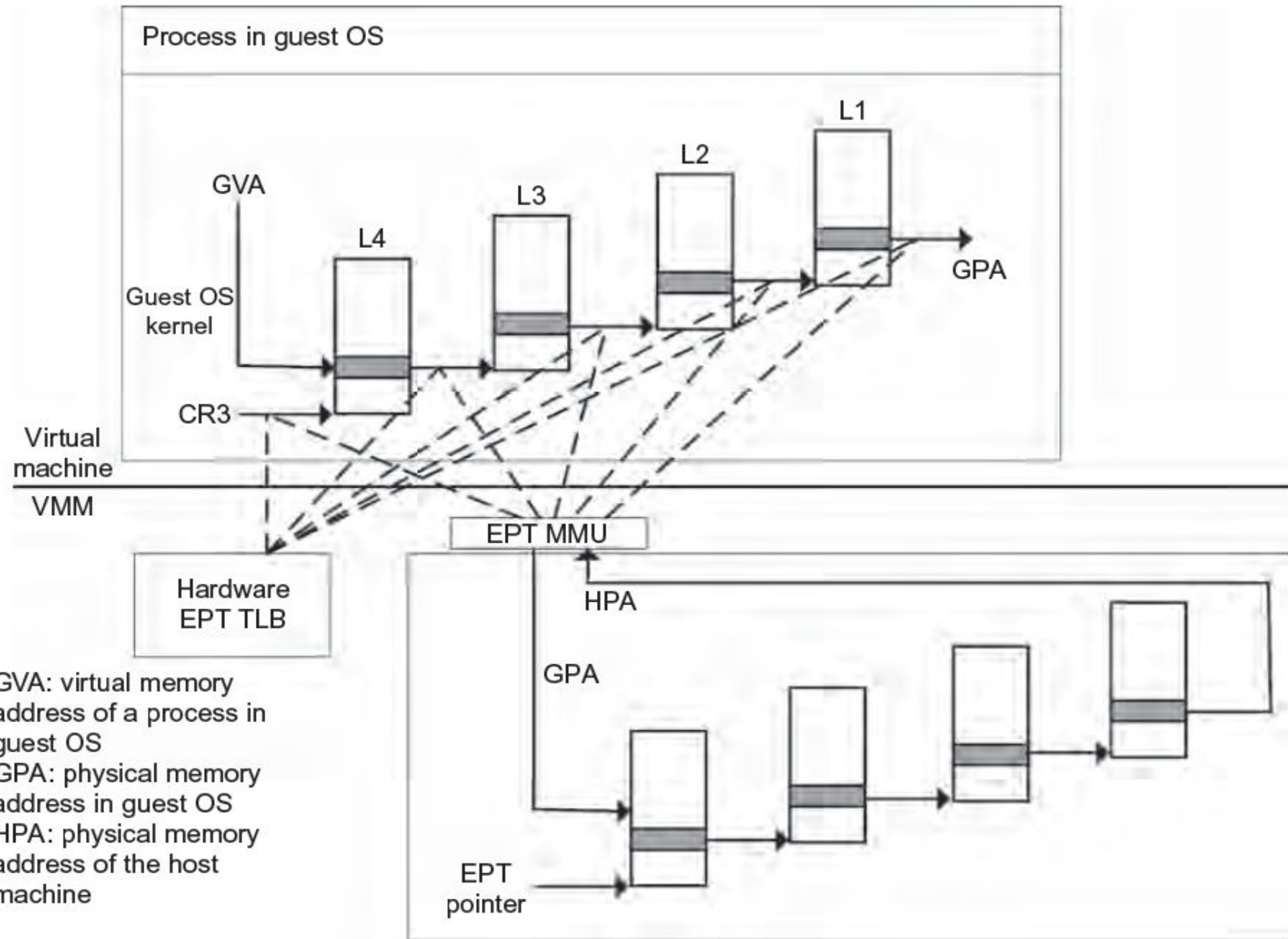


- 
- Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems.
  - In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory.
  - All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.


- 
- However, in a virtual execution environment, virtual memory virtualization involves **sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.**
  - That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively: **virtual memory to physical memory and physical memory to machine memory.**
  - Furthermore, MMU virtualization should be supported, which is transparent to the guest OS.
  - The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs.


- 
- But the guest OS cannot directly access the actual machine memory. The VMM is responsible for mapping the guest physical memory to the actual machine memory.









- 
- When a virtual address needs to be translated, the CPU will first look for the L4 page table pointed to by Guest CR3.
  - Since the address in Guest CR3 is a physical address in the guest OS, the CPU needs to convert the Guest CR3 GPA to the host physical address (HPA) using EPT.
  - In this procedure, the CPU will check the EPT TLB to see if the translation is there. If there is no required translation in the EPT TLB, the CPU will look for it in the EPT.
  - If the CPU cannot find the translation in the EPT, an EPT violation exception will be raised.

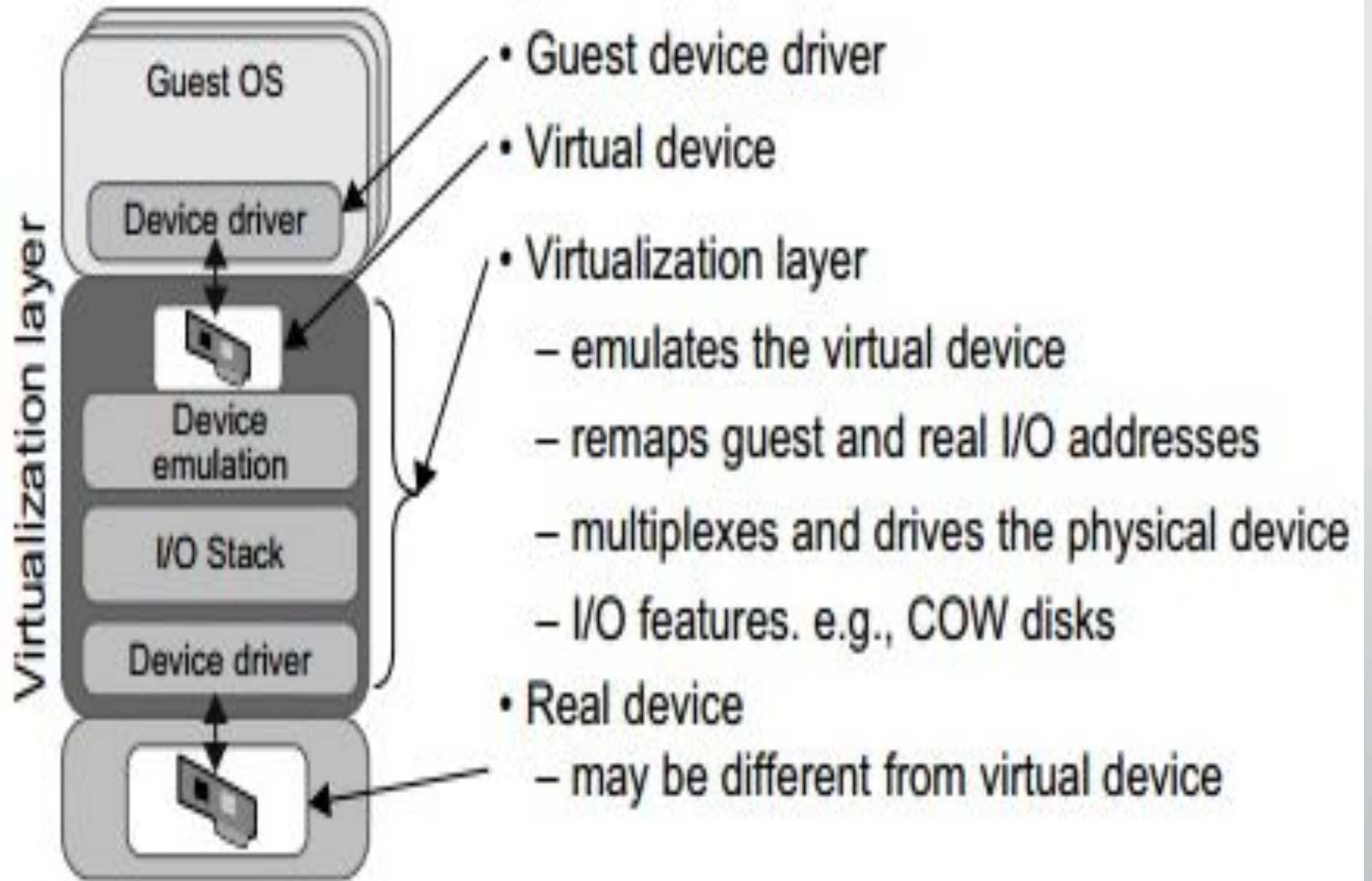
- 
- When the GPA of the L4 page table is obtained, the CPU will calculate the GPA of the L3 page table by using the GVA and the content of the L4 page table.
  - If the entry corresponding to the GVA in the L4 page table is a page fault, the CPU will generate a page fault interrupt and will let the guest OS kernel handle the interrupt.
  - When the PGA of the L3 page table is obtained, the CPU will look for the EPT to get the HPA of the L3 page table
  - To get the HPA corresponding to a GVA, the CPU needs to look for the EPT five times, and each time, the memory needs to be accessed four times.


- 
- There-fore, there are 20 memory accesses in the worst case, which is still very slow. To overcome this short-coming, Intel increased the size of the EPT TLB to decrease the number of memory accesses


# I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
- There are three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O.
- Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices.

- 
- All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software.
  - This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.



- 
- A single hardware device can be shared by multiple VMs that run concurrently.
  - The para-virtualization method of I/O virtualization is typically used in Xen.
  - It is also known as the split driver model consisting of a frontend driver and a backend driver.
  - The frontend driver is running in Domain U and the backend driver is running in Domain 0.
  - They interact with each other via a block of shared memory.
  - The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.

- 
- Direct I/O virtualization lets the VM access devices directly.
  - Intel VT-d supports the remapping of I/O DMA transfers and device-generated interrupts.