Virtualization and Cloud Computing

- Virtualization plays an important role in Cloud computing, since it allows for the appropriate degree of customization, security, isolation, and manageability that are fundamental for delivering IT services on demand
- Hardware and programming language virtualization are the techniques adopted in Cloud computing systems (IaaS & PaaS).

Virtualization and Cloud Computing

- Computation on demand
- Consolidation
- Isolated and controllable environments
- *Virtual machine migration*
- Live migration
- Desktop virtualization

Xen: Paravirtualization

- **Open source** initiative
- Developed at the University of Cambridge, open source community backing it
- Commercial solution, **XenSource**, by Citrix
- Provides desktop virtualization or server virtualization, and recently it has also been used to provide Cloud computing solutions by means of *Xen Cloud Platform (XCP)*.

Xen

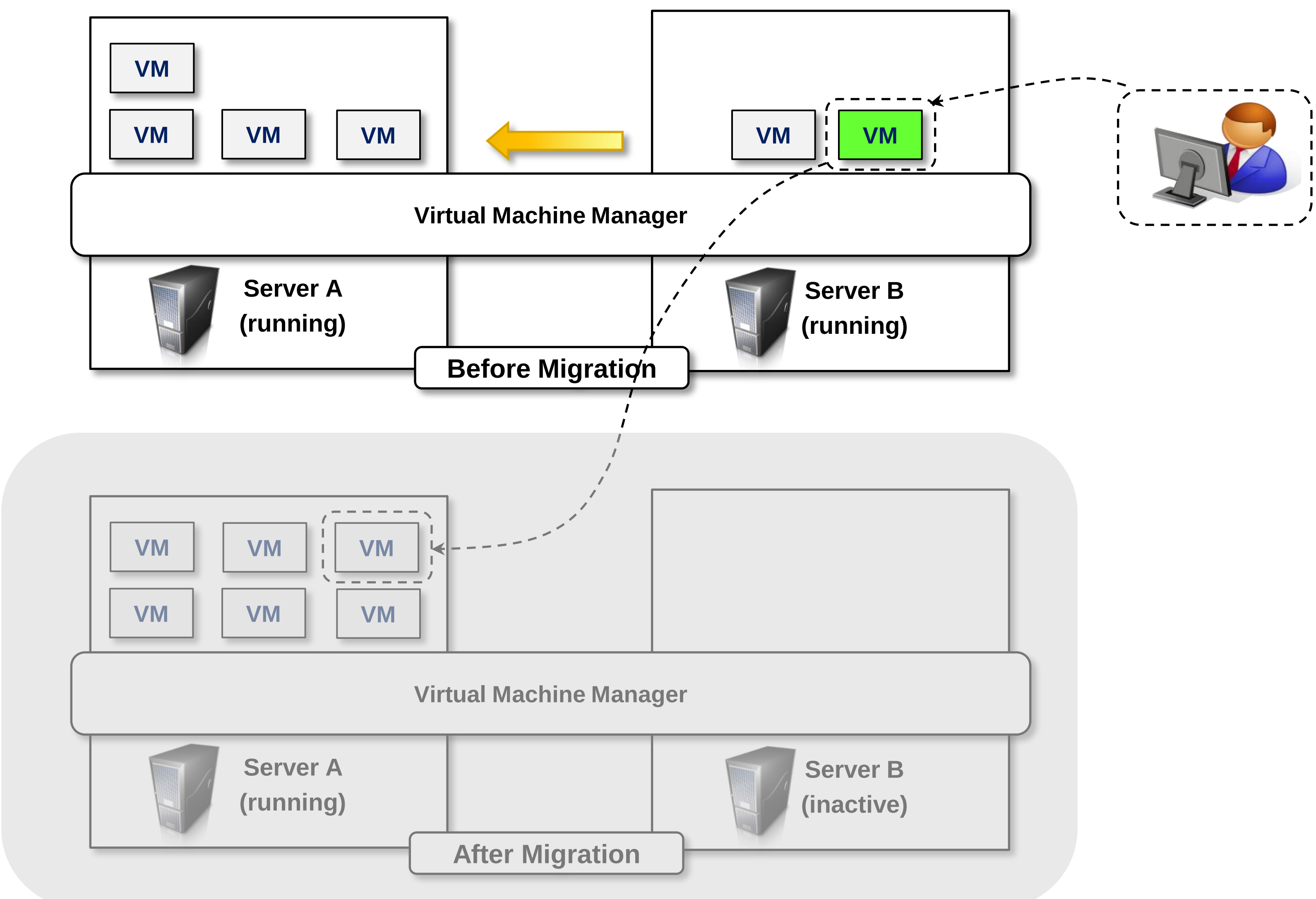
- Recently Xen has started to support full virtualization /hardware-assisted virtualization
- Xen allows high performance execution of guest operating systems
- A Xen-based system is managed by the *Xen hypervisor*, which runs in the highest privileged mode and controls the access of guest operating system to the underlying hardware.

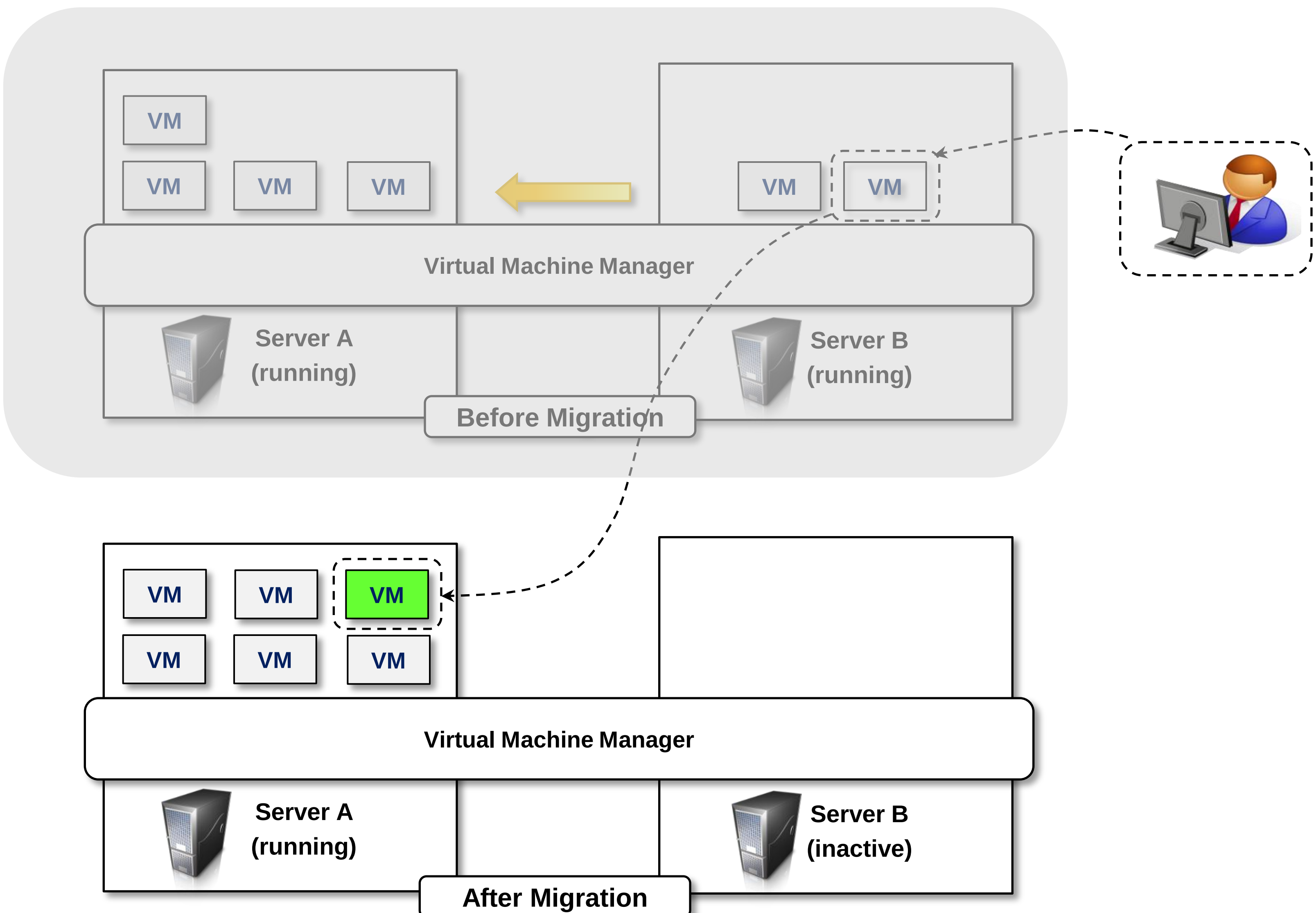
Xen Conti…

- Xen is a hypervisor providing services that allows multiple OS to execute on the same hardware concurrently.
- Xen is a powerful hypervisor capable of running many different types of guest OS like linux, netBSD, Solaris and Microsoft Windows.
- Xen is a bare-metal hypervisor which provides a layer to the Guest OS for interfacing with the hardware.

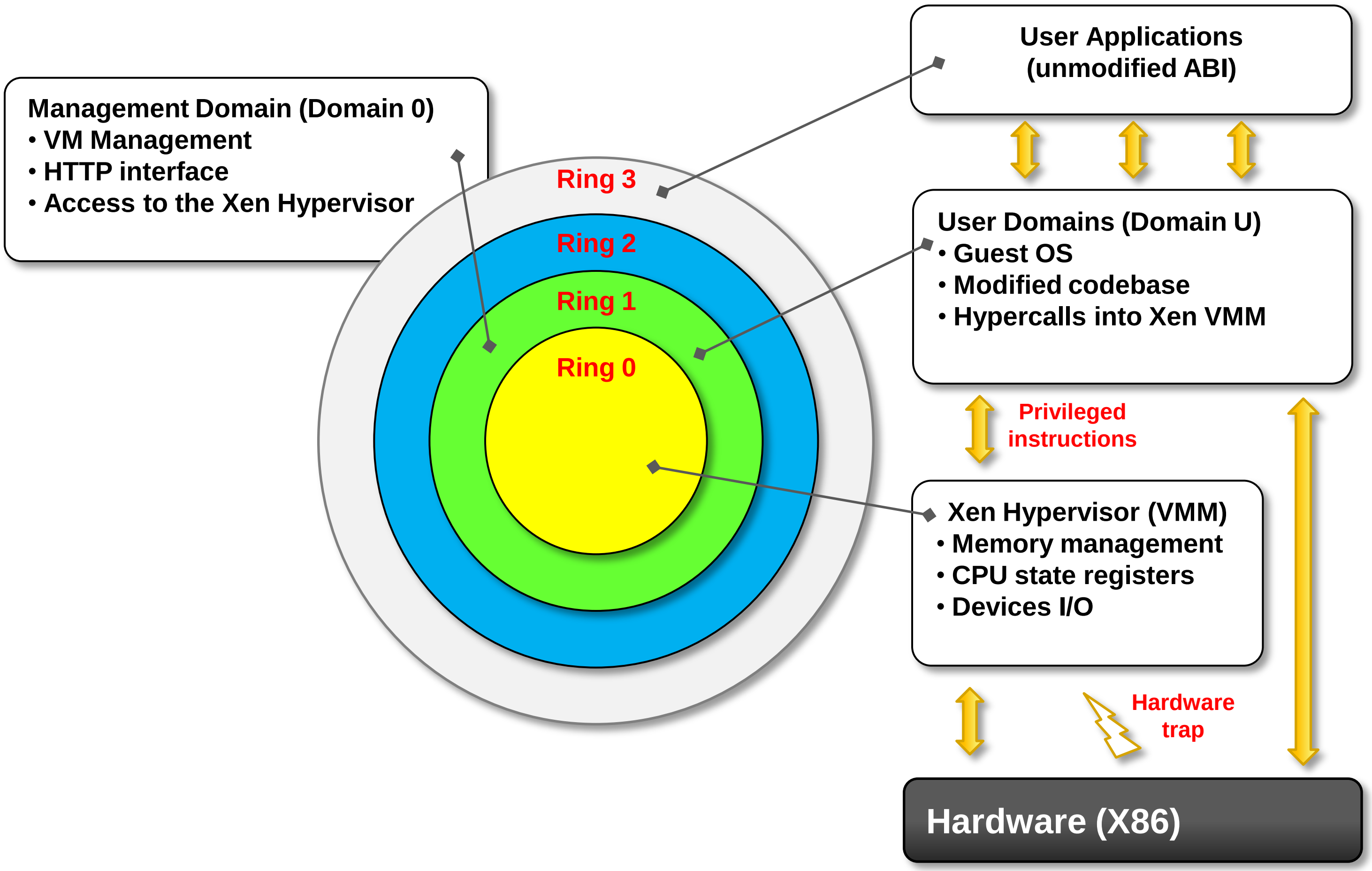
Xen Conti…

- Xen is *capable of providing both para-virtualization and full virtualization*. For Full Virtualization their should be Hardware-assisted Virtualization support.
- Xen also supports live migration from one OS to another.
- For this Xen loads all the OS address space in the memory so switching can be ultra-fast.


VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(running)
Before Migration
VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(inactive)
After Migration


VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(running)
Before Migration
VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(inactive)
After Migration

Xen Architecture and Guest OS Management



Xen Conti…

- System Requirements for Xen:
- Processor: 64-bit processor with minimum clock speed of 1.5 Ghz with Intel VT or AMD-V for Full Virtualization
- Ram: 1 GB (minimum), 2GB or more (Recommended)
- Disk Space: 16GB (minimum), 60GB or more(Recommended)

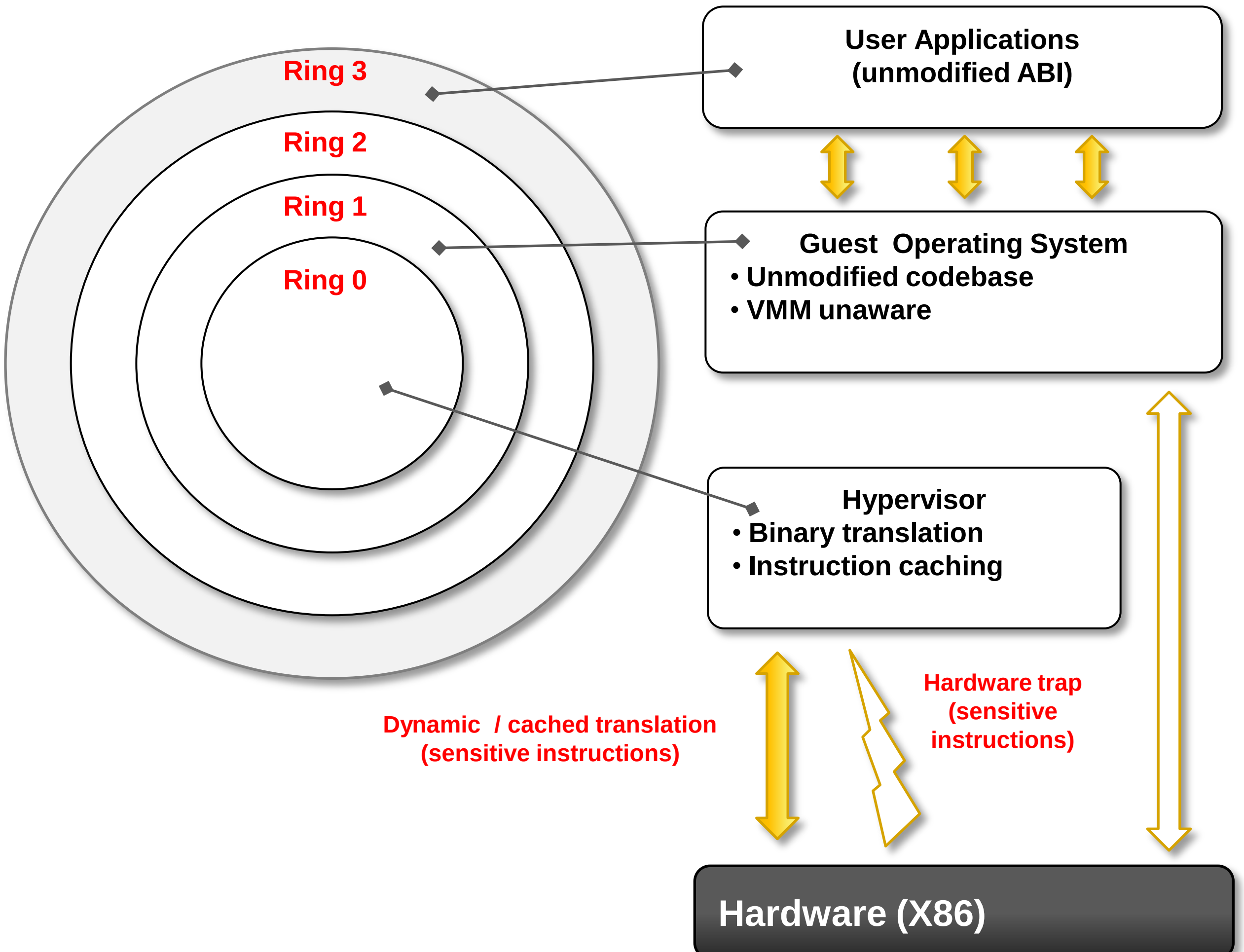
VMware: Full Virtualization

- Underlying hardware is replicated and made available to the guest operating system
- It implements full virtualization either in
  - the desktop environment by means of *Type II hypervisors*, or
  - the server environment, by means of *Type I hypervisors*
- by means of *direct execution* (for non-sensitive instructions) and *binary translation* (for sensitive instructions),

VMware: Full Virtualization

- With the new generation of hardware architectures and the introduction of *hardware assisted virtualization* (i.e. Intel VT-x and AMD V) in 2006, full virtualization is made possible with hardware support
- the use of *dynamic binary translation* was the only solution allowing running X86 guest operating systems unmodified in a virtualized environment

Full Virtualization Reference Model



Virtualization Solutions

- Several Companies offers a collection of virtualization solutions covering the entire range of market from desktop computing to enterprise computing and infrastructure virtualization.
- ***Server Virtualization***
- ***Desktop Virtualization***
- ***Infrastructure Virtualization and Cloud Computing Solutions***

Library-level virtualization

- Most applications use APIs exported by user-level libraries rather than using system calls by the OS.
- Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization.

Library-level virtualization

- Virtualization with library interfaces is possible by *controlling the communication link between applications and the rest of a system through API hooks*.
- The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.
- Another example is vCUDA that allow application running on VMs to leverage GPU's for acceleration

Library-level virtualization

- Also known as user-level Application Binary Interface (ABI) or API emulation
- It can *create execution environments for running alien programs on a platform* rather than creating a VM to run the entire operating system

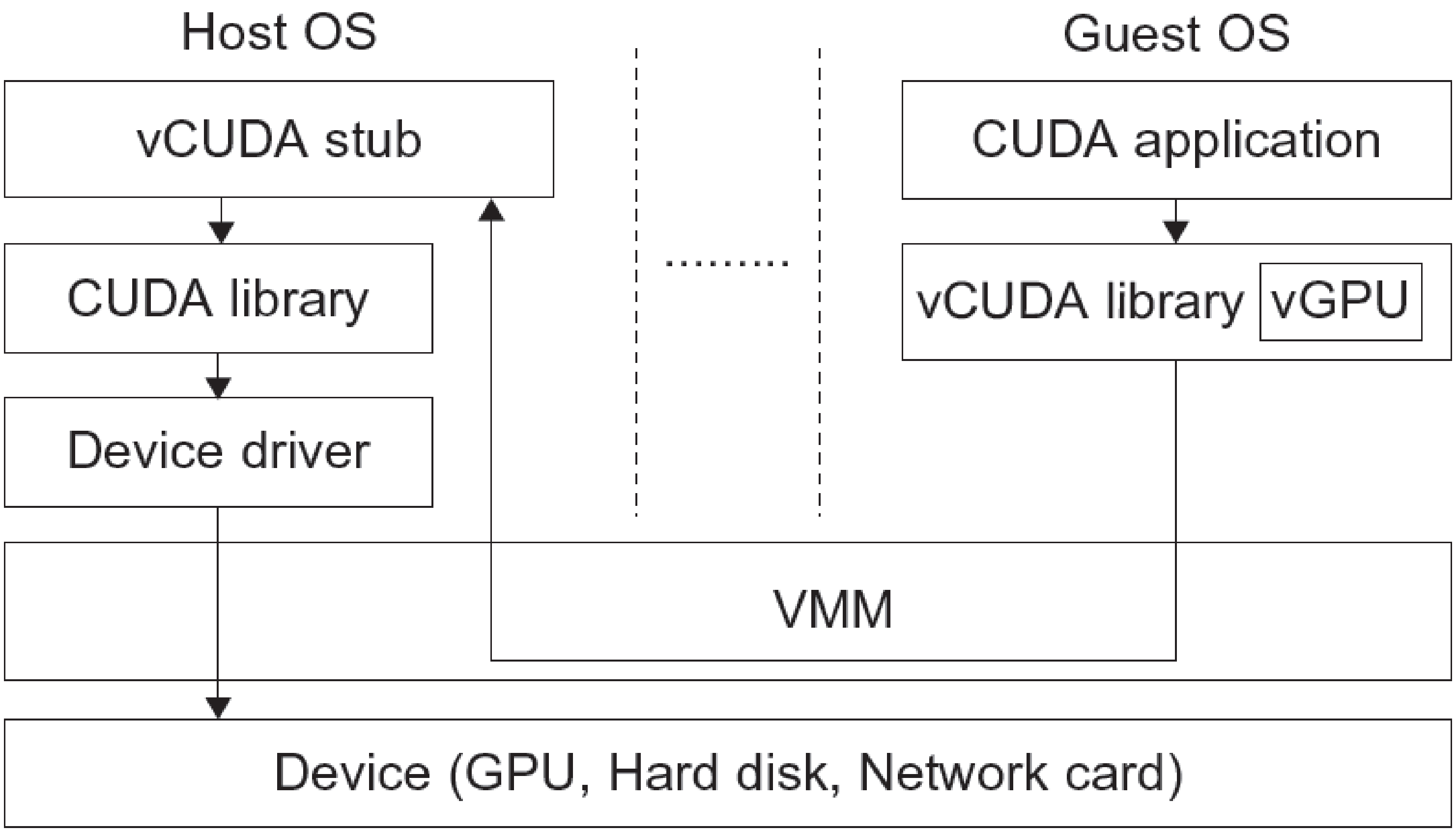
The vCUDA for Virtualization of General-Purpose GPUs

- CUDA is a programming model and library for general-purpose GPUs
- It is difficult to run CUDA applications on hardware-level VMs directly
- *vCUDA virtualizes the CUDA library and can be installed on guest OSes*
- When CUDA applications run on a guest OS and issue a call to the CUDA API, vCUDA intercepts the call and redirects it to the CUDA API running on the host OS

vCUDA

- The vCUDA employs a client-server model to implement CUDA virtualization.
- It consists of three user space components: the vCUDA library, a virtual GPU in the guest OS (which acts as a client), and the vCUDA stub in the host OS (which acts as a server).
- The vCUDA library resides in the guest OS as a substitute for the standard CUDA library. It is responsible for intercepting and redirecting API calls from the client to the stub. Besides, vCUDA also creates vGPUs and manages them.

Basic concept of the vCUDA architecture



Ref.: Distributed and Cloud Computing: From Parallel Processing to the Internet of Things Kai Hwang , Jack Dongarra , Geoffrey C. Fox.

Virtualization of Virtual Memory

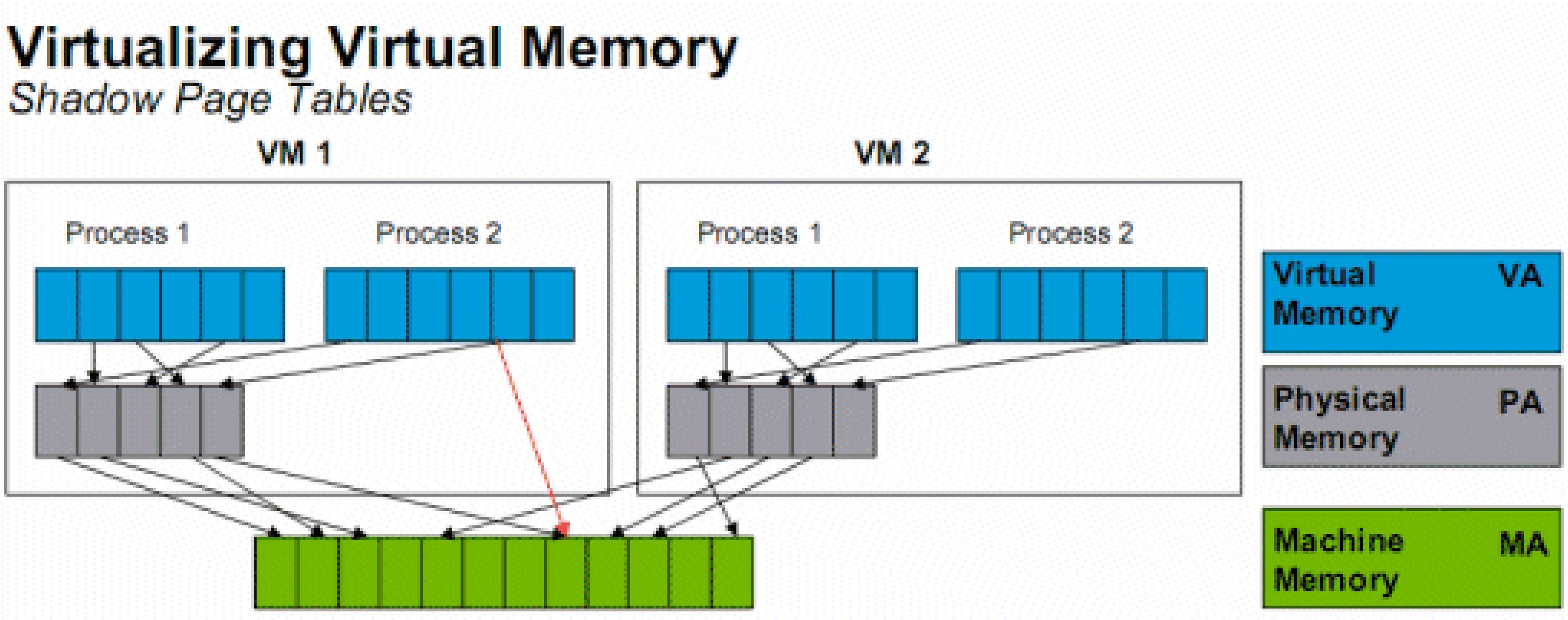
Memory Virtualization

- In a traditional environment, the OS 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, a hardware unit) and a Translation Lookaside Buffer (TLB, an associative cache of page table entries) to optimize virtual memory performance

Memory Virtualization

- But, 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.
- So a two-stage mapping process should be maintained by the guest OS and the VMM, respectively:
  - virtual memory to physical memory &
  - physical memory to machine memory.

Memory Virtualization



Ref.: Distributed and Cloud Computing: From Parallel Processing to the Internet of Things Kai Hwang , Jack Dongarra , Geoffrey C. Fox.

Memory Virtualization

- 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

Memory Virtualization

- Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the **shadow page table**
- The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor

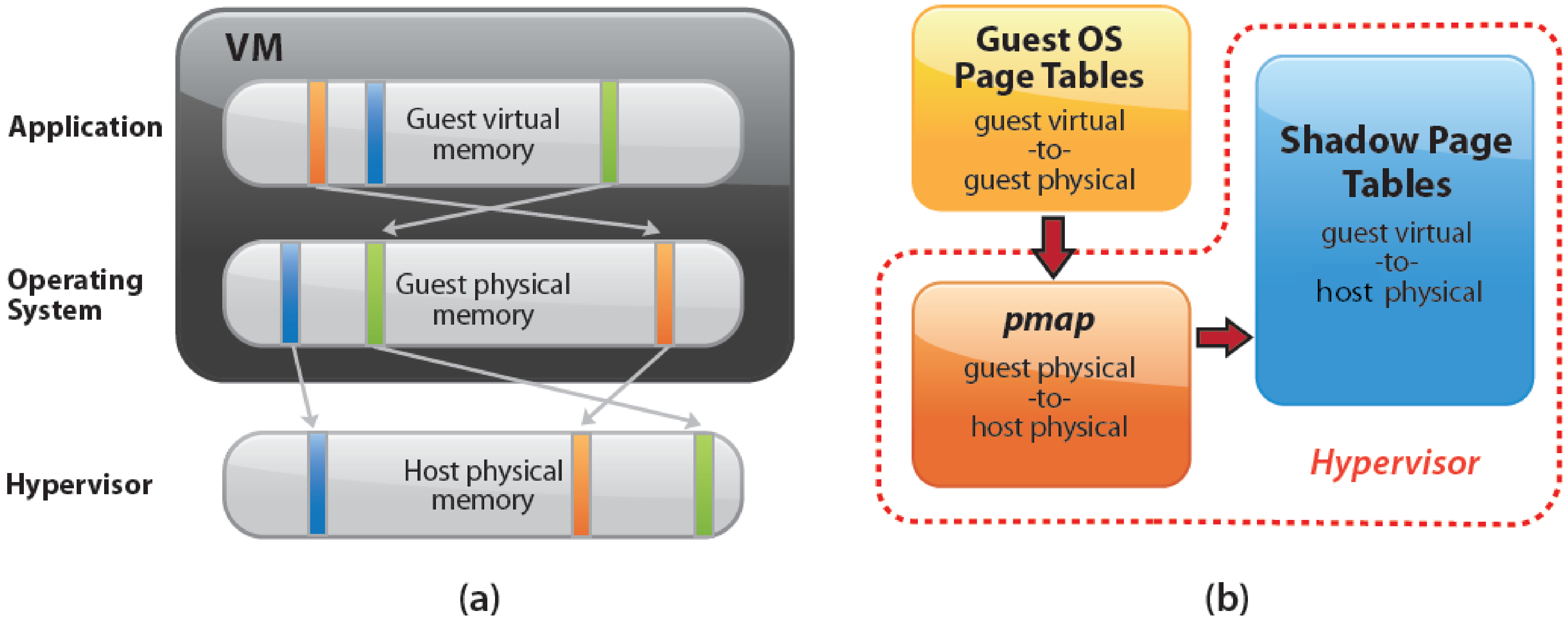
Memory Virtualization

- Since modern operating systems maintain a set of page tables for every process, *the shadow page tables will get flooded.*
- VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.

Memory Virtualization

- Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access.
- When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

More on it can be found in
http://www.vmware.com/files/pdf/mem_mgmt_perf_vsphere5.pdf


VM
Application
Guest virtual memory
Operating System
Guest physical memory
Hypervisor
Host physical memory
(a)
Guest OS Page Tables
guest virtual -to- guest physical
pmap
guest physical -to- host physical
Shadow Page Tables
guest virtual -to- host physical
Hypervisor
(b)

Summary comparison of X86 processor virtualization techniques

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic