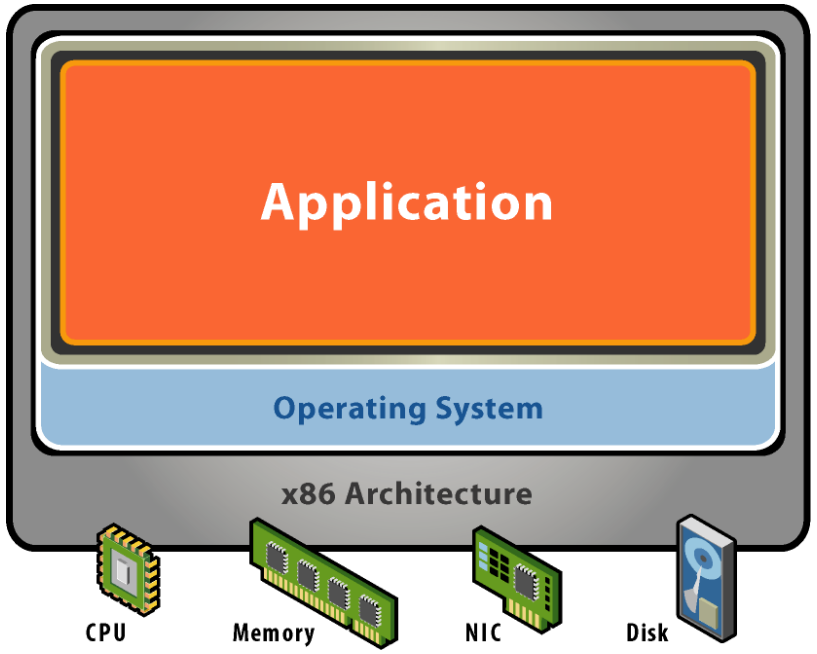
System Virtualization

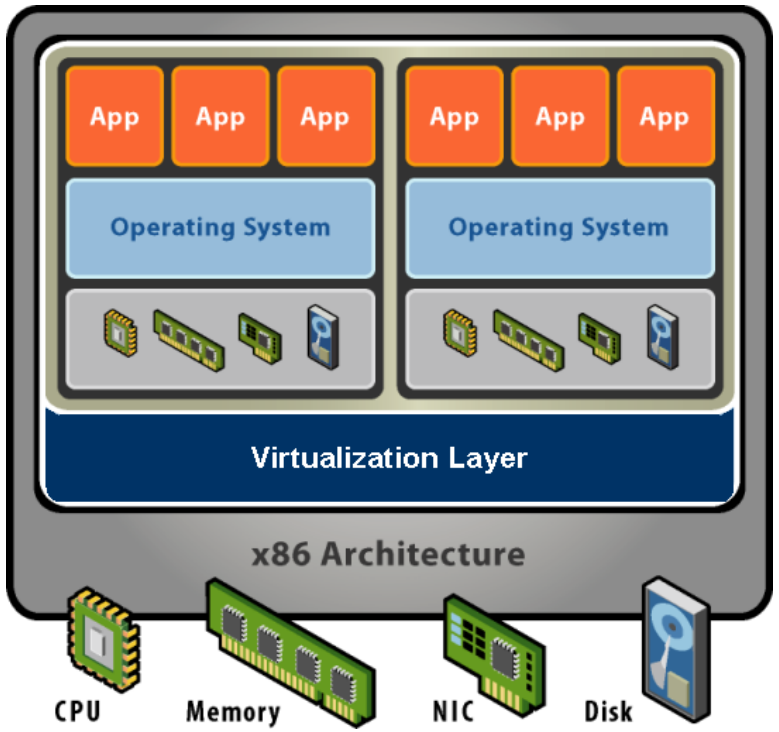
Sridhar Alagar

A Physical machine



- Physical hardware
  - Resources often under utilized
- Software - OS
  - Tightly coupled to hardware
  - Single active OS instance
  - OS controls hardware

What is a Virtual Machine?



- Software Abstraction
  - Isolated duplicate of the hardware
- Virtualization Layer
  - Another level of indirection
  - Decouples hardware
  - Multiplexes physical hardware across VMs

Virtual Machine Monitor

- VMM implements virtualization layer
- VMM must satisfy these core properties:
  - Equivalence: Virtual environment should be identical to physical
  - Safety/Isolation: VMM must have control of system resources. VMs should be isolated from on another
  - Performance: Little or no difference in performance

Virtualization History

- VMM was first implemented for classical IBM mainframes
- Time share several single-user OS
- Interests for virtualization died with multi-user OS

Virtualization Renaissance

- Fundamental to cloud computing
- Pervasive in data center

- Why?
  - Availability of commodity x86 platform

Virtualization Applications

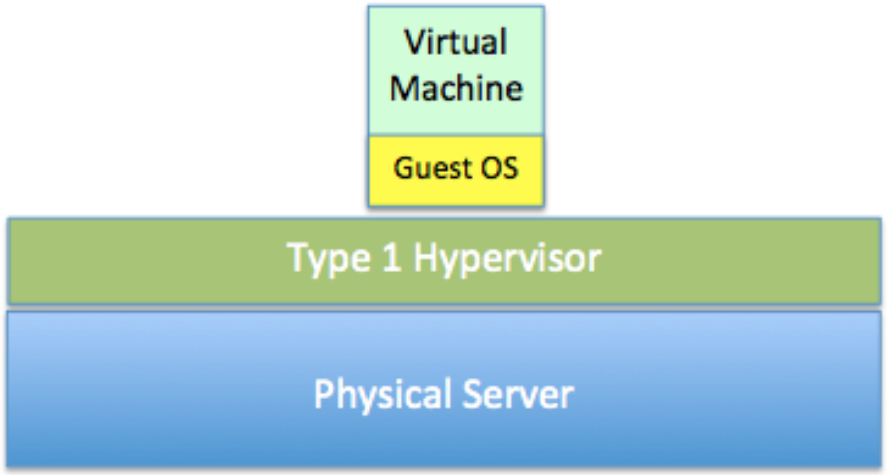
- Server consolidation
  - "One server one app" to "one server many VMs"
  - Significant cost savings due to less power, space, and equipment
- Simplified management
  - Data center provisioning
  - Dynamic load balancing
- Improved availability
  - Fault tolerance
  - Disaster recovery

Who led the resurgence?

- By a group of Professors from Stanford
  - Disco, a VMM for MIPS processor
- Later founded VMWare
  - A market leader
- First commercial "hypervisor" for x86 in 2001

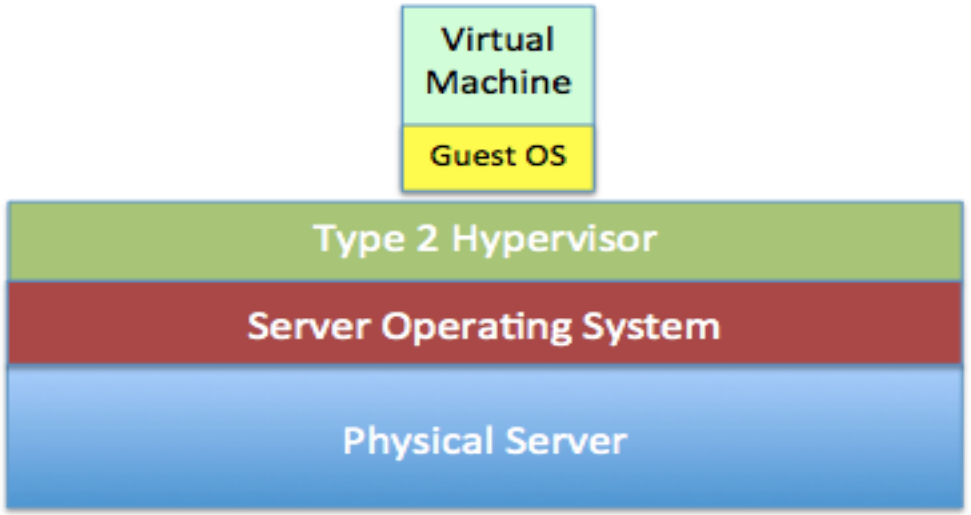
Type 1 hypervisor

- Runs directly on the hardware
- Secure and available
- Better performance



Type 2 hypervisor

- Runs on top of an OS
- Leverages OS drivers
- Application development and testing



Virtualizing CPU

- Booting a VM is easy
  - Start from the first instruction
- VMM performs "machine switch" between VMs
  - saves the machine state of one OS
- Executing a privileged operation is tricky
  - Can't let guest OS take control of the CPU

System Call Implementation

```
open:
    push    dword mode
    push    dword flags
    push    dword path
    mov     eax, 5
    push    eax
    int     80h
```

- Transfers control to a well defined interrupt (trap) handler
  - OS establishes this with the hardware during startup

Trap (int) without virtualization

Process	Hardware	Operating System
1. Execute instructions (add, load, etc.)		
2. System call: Trap to OS		
	3. Switch to kernel mode; Jump to trap handler	
		4. In kernel mode; Handle system call; Return from trap
	5. Switch to user mode; Return to user code	
6. Resume execution (@PC after trap)		

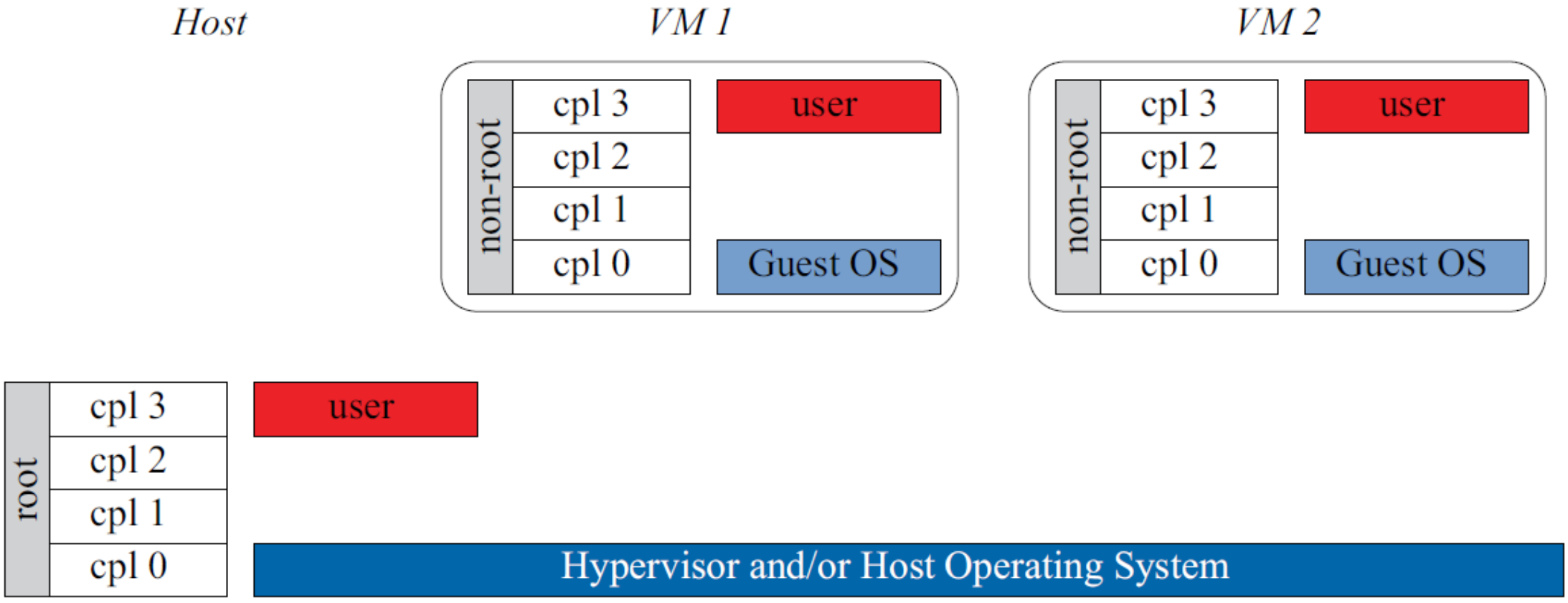
Trap and Emulate with virtualization

Process	Operating System	VMM
1. System call: Trap to OS		
		2. Process trapped: Call OS trap handler (at reduced privilege)
	3. OS trap handler: Decode trap and execute syscall; When done: issue return-from-trap	
		4. OS tried return from trap: Do real return from trap
5. Resume execution (@PC after trap)		

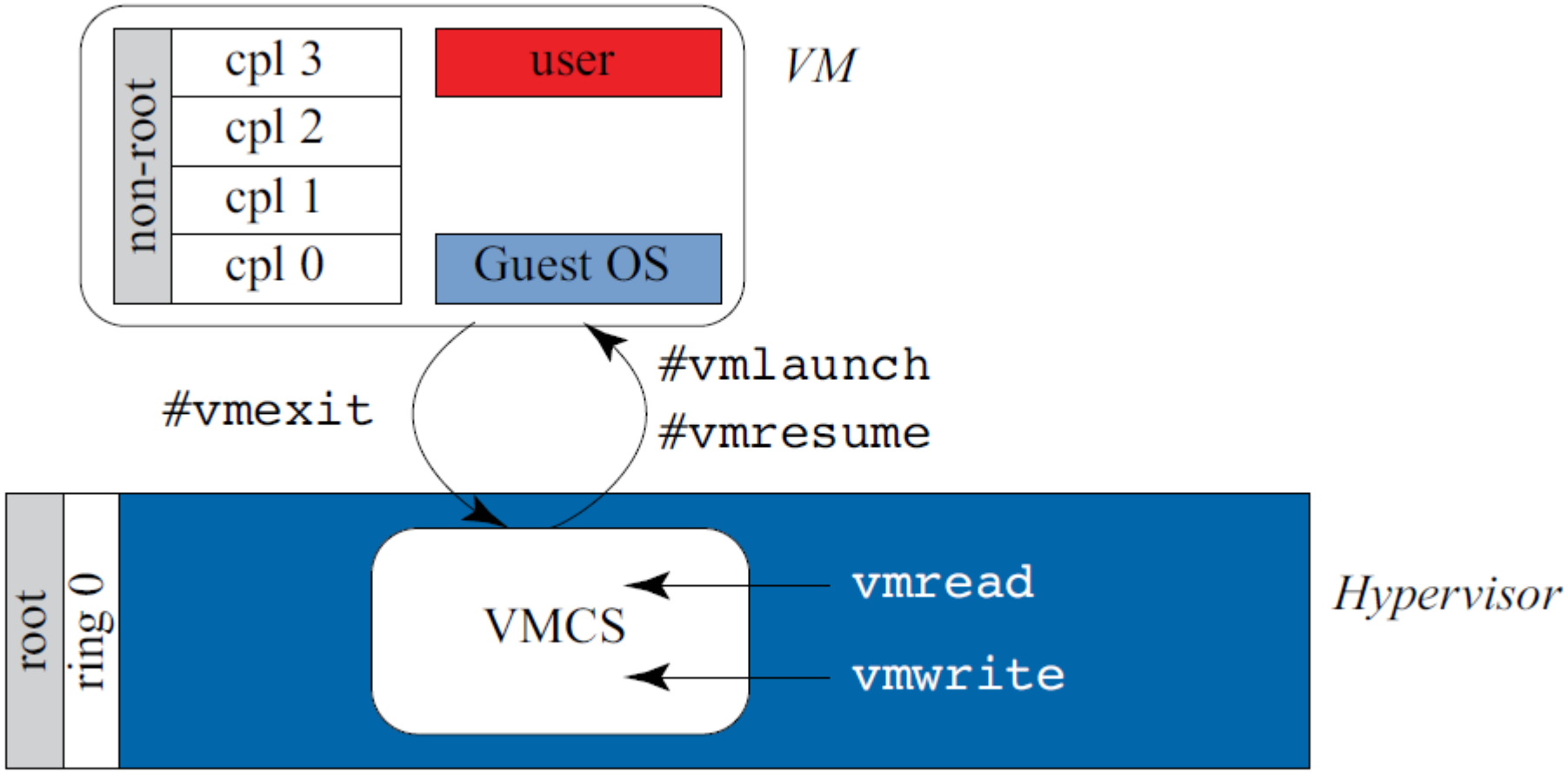
In what mode should the OS run?

- Certainly not privileged
- Not user mode either
- Special mode called "supervisor mode"
  - Provided by MIPS
- On hardware that does not offer supervisor mode
  - OS runs in user mode
  - VMM uses Binary Translation and memory protection using segmentation

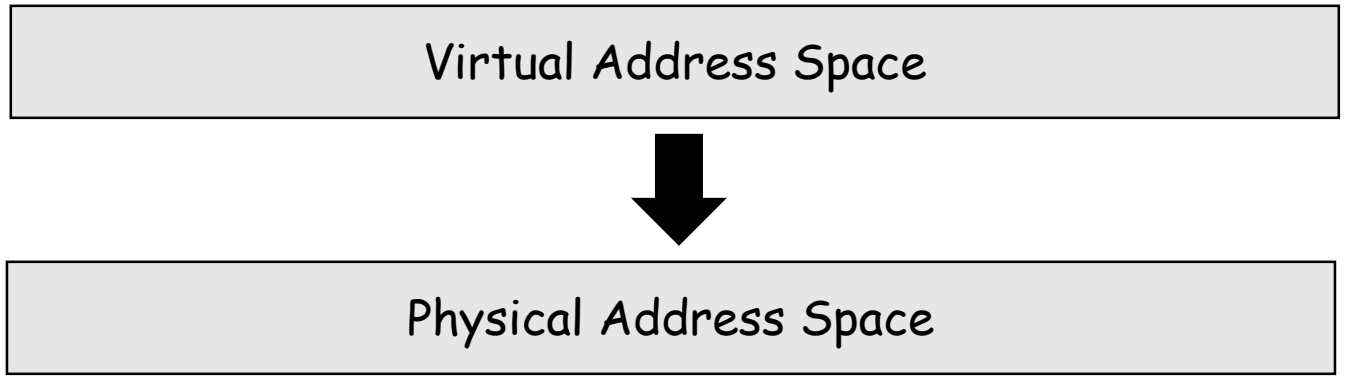
VT-x: Intel's technology for Virtualization



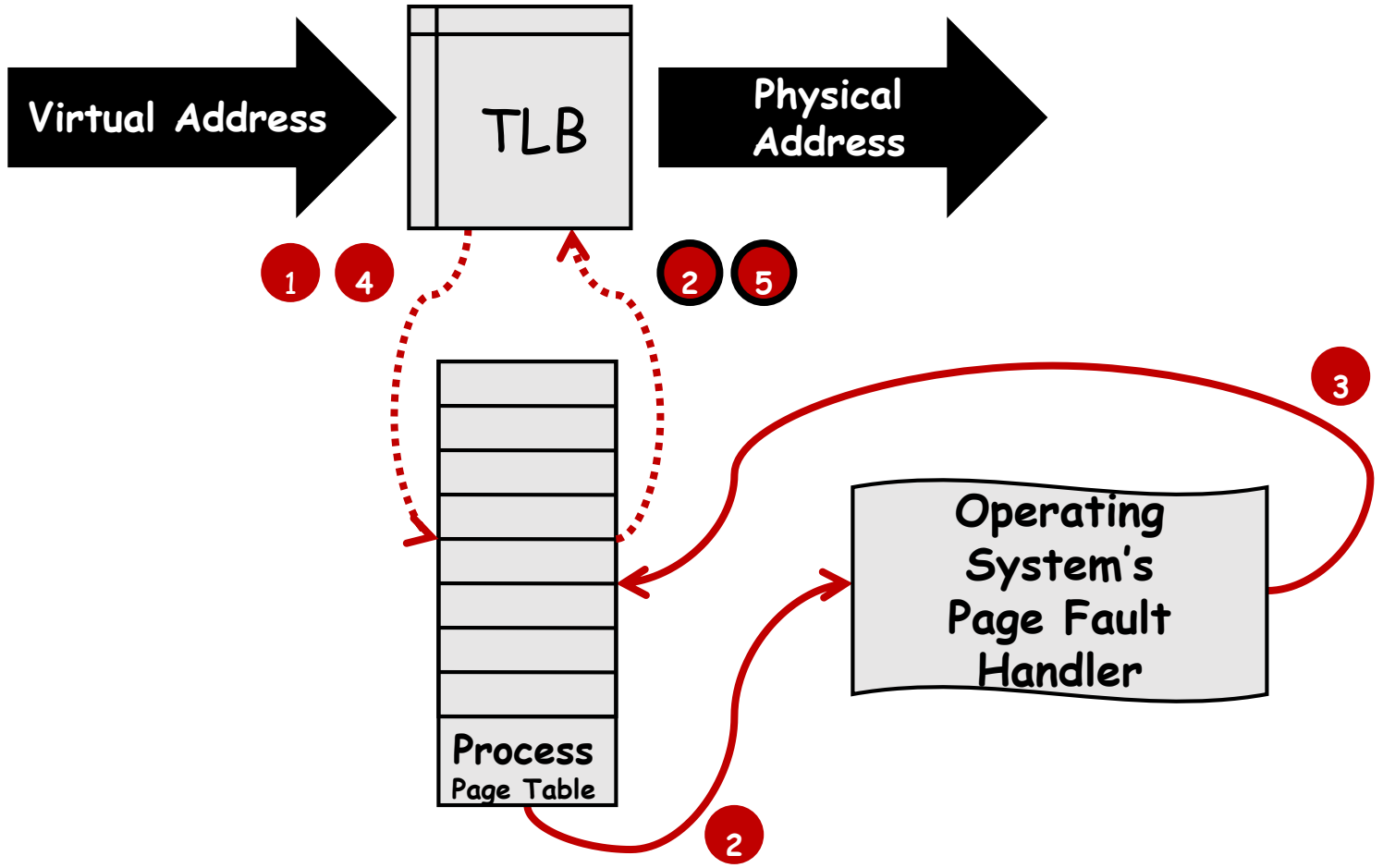
VT-x Transitions and Control Structure



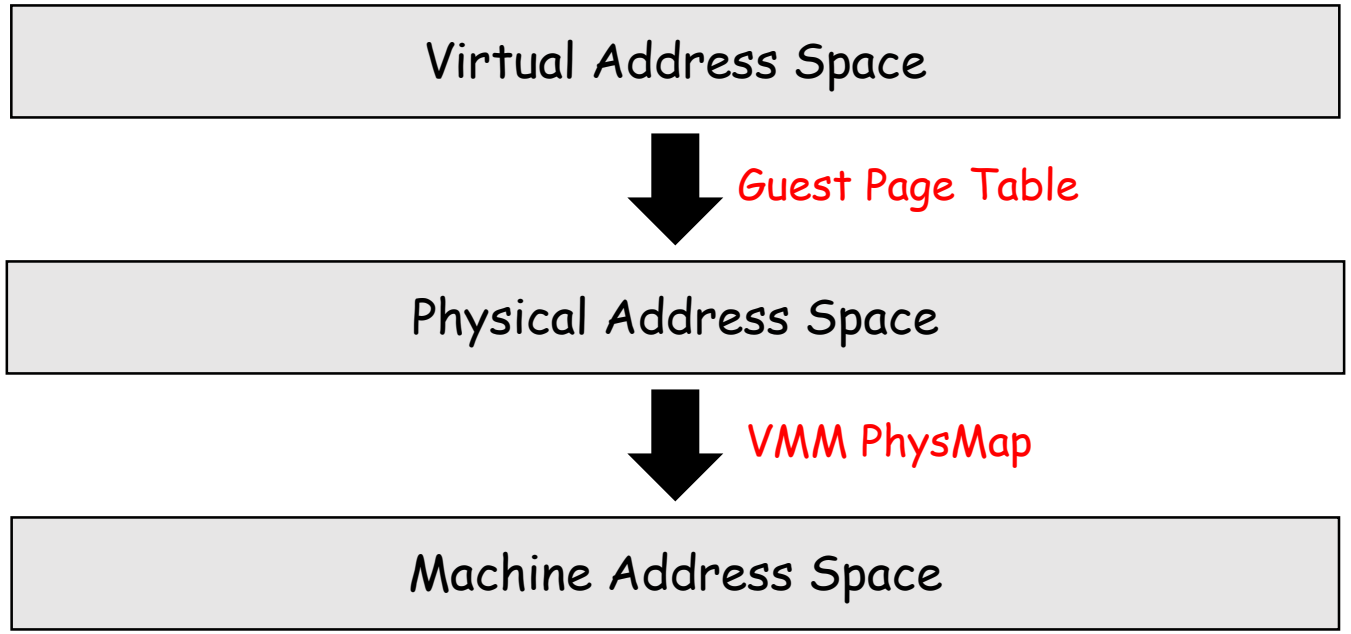
Traditional Memory Mapping



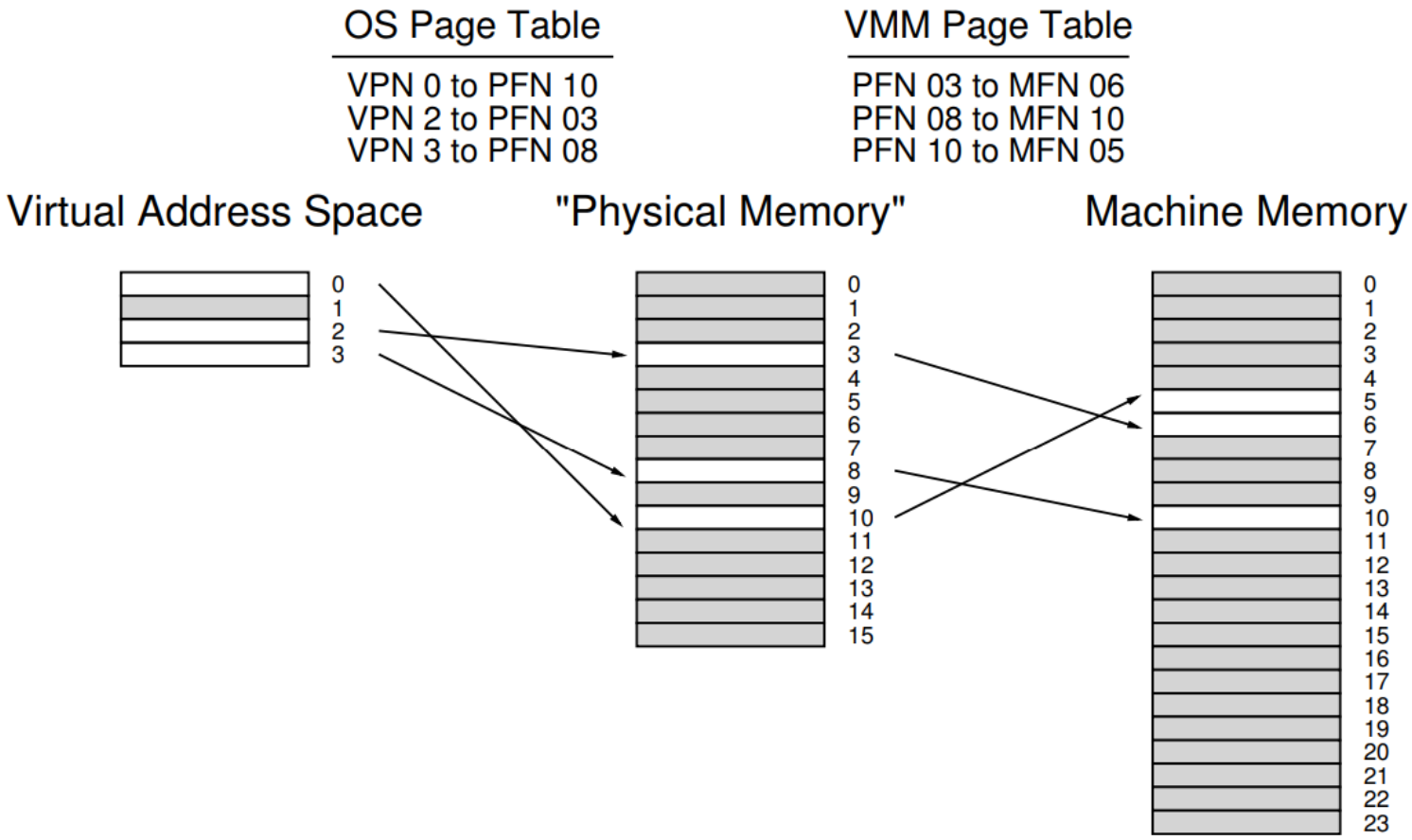
Traditional Address Translation



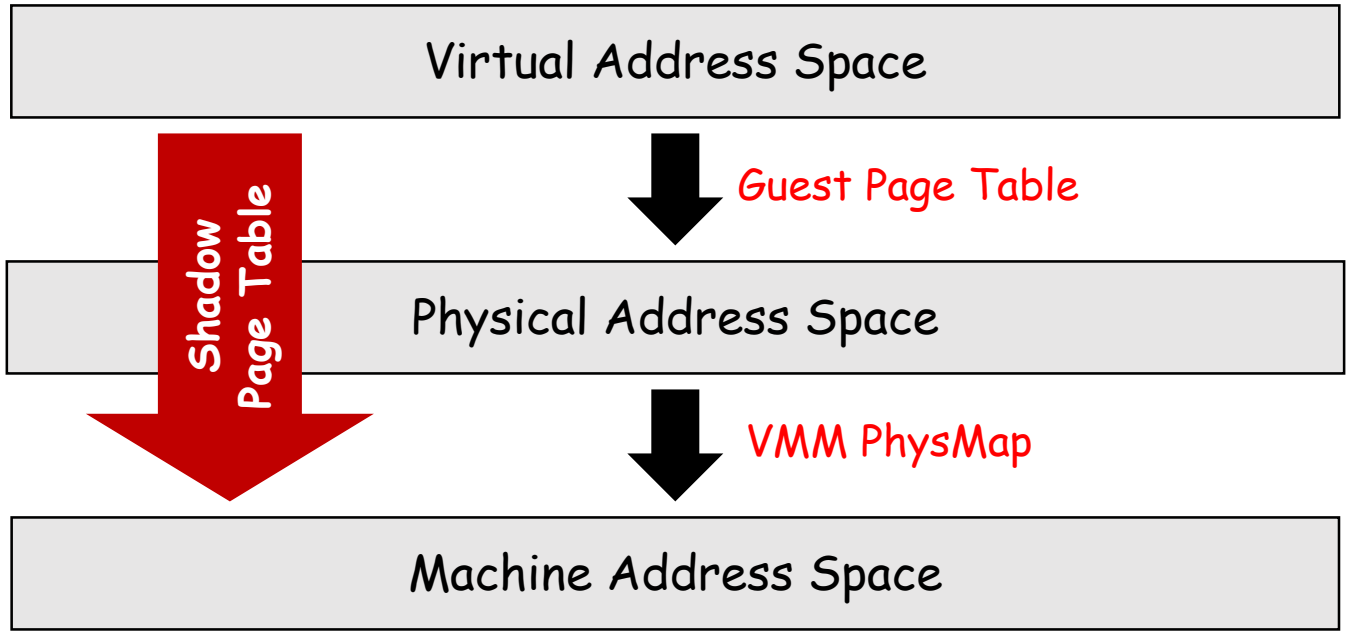
Virtualized Memory Mapping



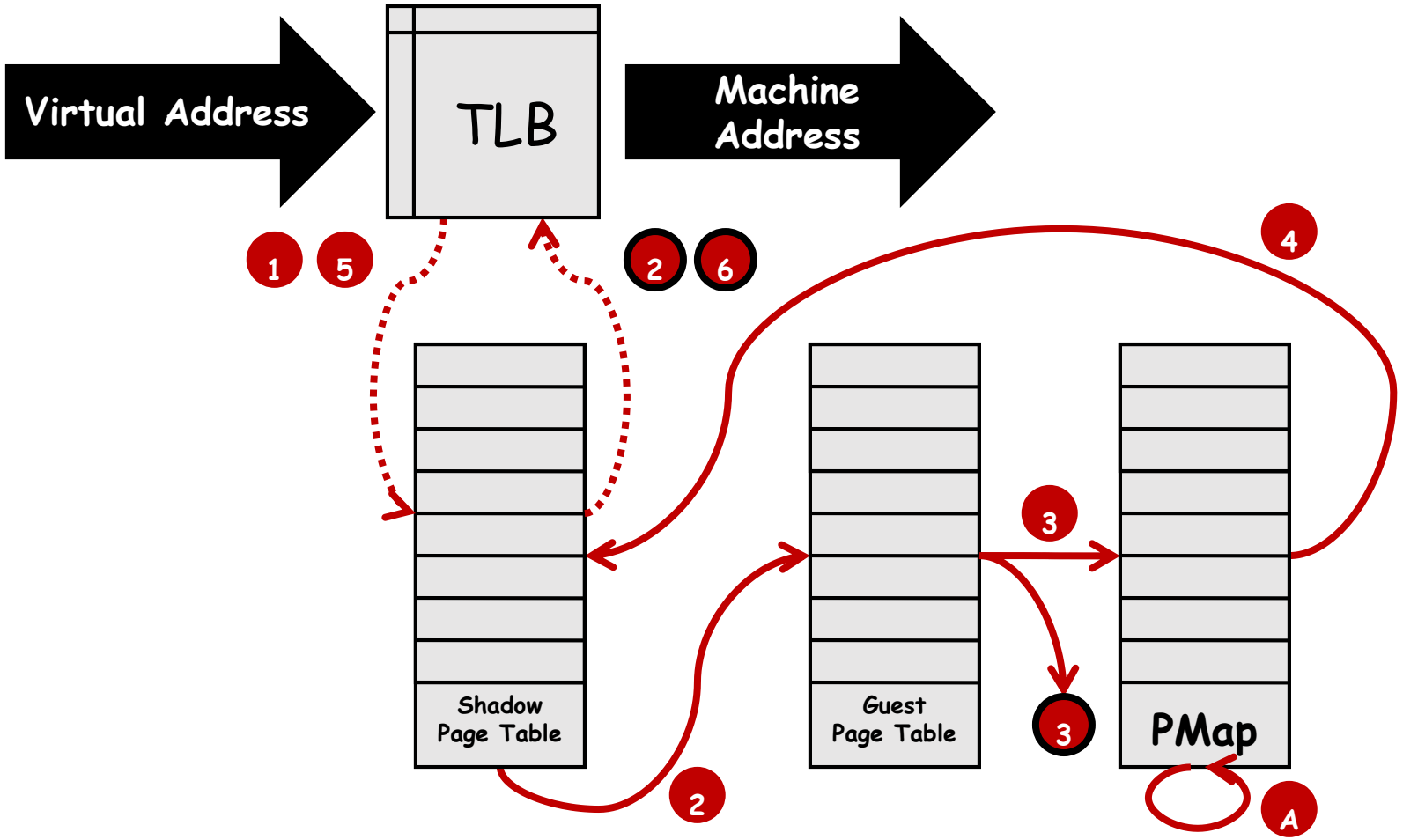
Virtualized Memory mapping



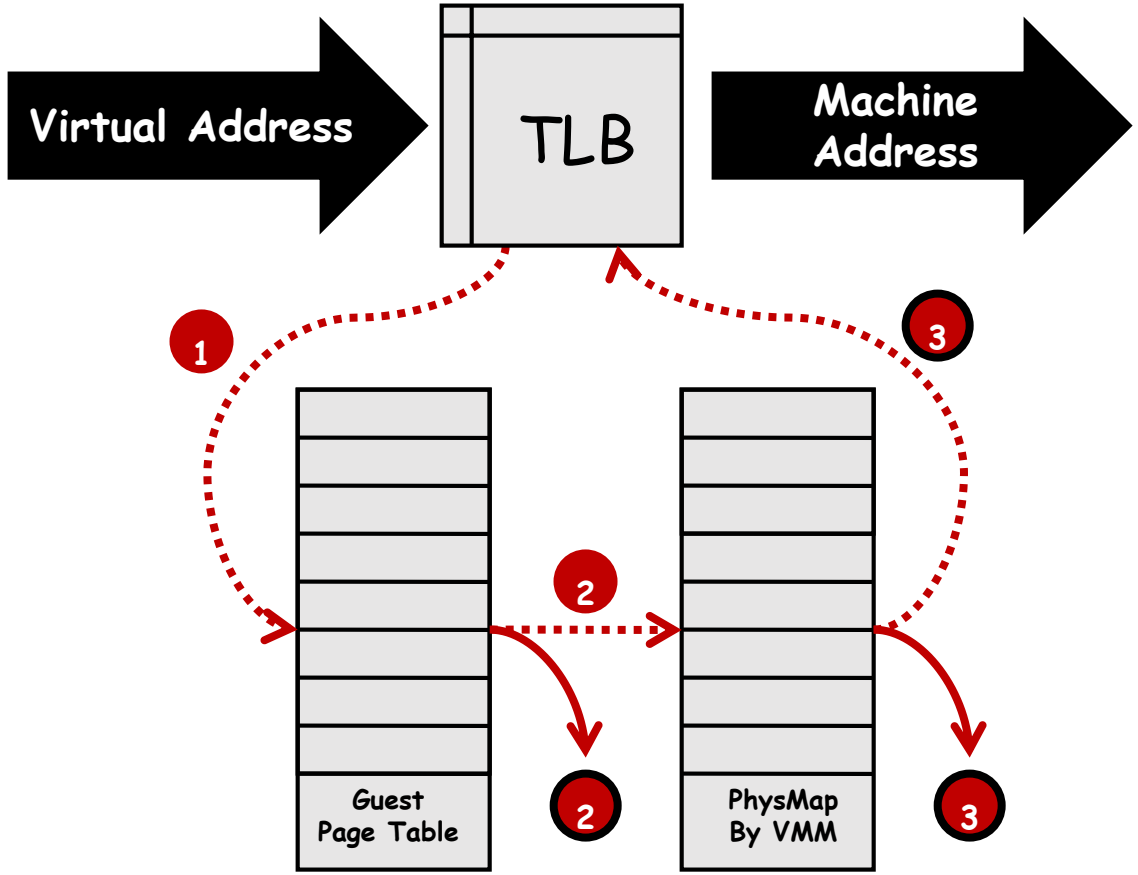
Virtualized Memory with Shadow PT



Virtual Address Translation with Shadow PT



Virtual Address Translation with Nested PT



Full/para/nested Virtualization

Disclaimer

- Some of the materials in this lecture slides are from the lecture slides by Prof. Arpaci, Prof. Youjip, and other educators. Thanks to all of them.

Additional Sources:

- Hardware and Software Support for Virtualization, Edouard Bugnion, Jason Nieh, and Dan Tsafrir, Morgan & Claypool Publishers
- https://www.vmware.com/pdf/virtualization.pdf