CompSci 401: Cloud Computing

Software Emulation

Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Approaches to virtualization

- Virtualization is key in cloud computing
  - Resource sharing
  - Isolation between cloud tenants

Approaches to virtualization

- Virtualization is key in cloud computing
  - Resource sharing
  - Isolation between cloud tenants
- **Software emulation**
- Para-virtualization
- Full virtualization

Software emulation

- Given a CPU A and another CPU B, we can virtualize A by emulating A's operation inside B
  - Read one instruction **x** from A
  - Run multiple instructions on B to perform **x**'s work
- Examples:
  - Running Super Nintendo emulators on PC
  - Running programs inside the Java Virtual Machine

Mega Man X on ZSNES

Byte-code interpreter

- Some languages compile to bytecode
  - Java, JavaScript, Python (.pyc files)
- Bytecode is then interpreted by another program
  - Java Virtual Machine, browsers, and the Python interpreter
- Easier to run application on multiple computer architectures
  - Only need to port the bytecode interpreter

Emulators and interpreters are slow

- Each instruction on CPU A may take many instructions on B to emulate
  - Consider gaming console running at 100MHz and Intel CPU at 2GHz
  - Slowdown if each instruction from the console takes 20+ instructions on the PC
- Multi-factor slowdown

Emulators and interpreters are slow

- Each instruction on CPU A may take many instructions on B to emulate
  - Consider gaming console running at 100MHz and Intel CPU at 2GHz
  - Slowdown if each instruction from the console takes 20+ instructions on the PC
- Multi-factor slowdown
- Just-In-Time compilation to improve performance
  - Java, JavaScript, some Python interpreters

CompSci 401: Cloud Computing

Privilege Levels in CPUs

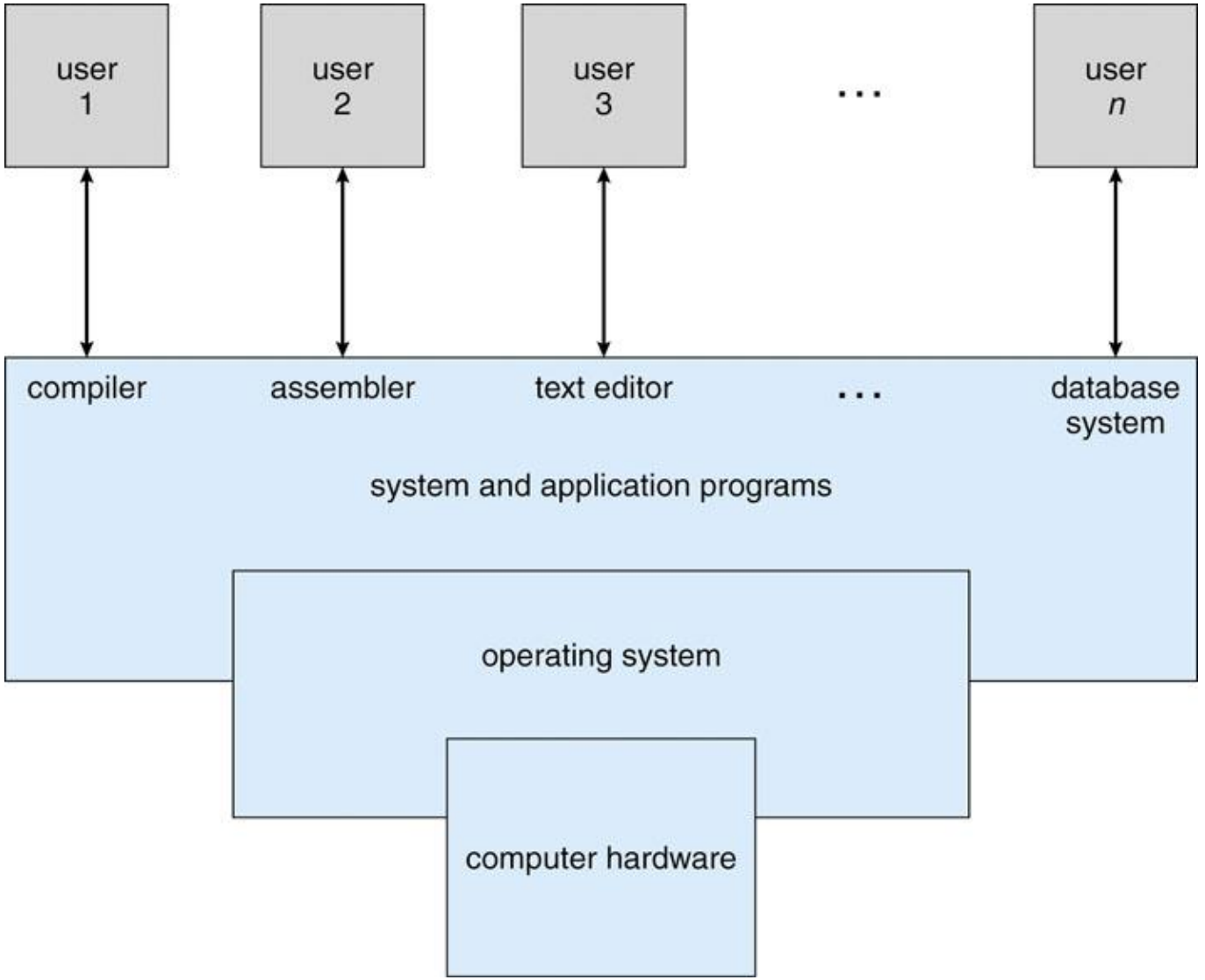
Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

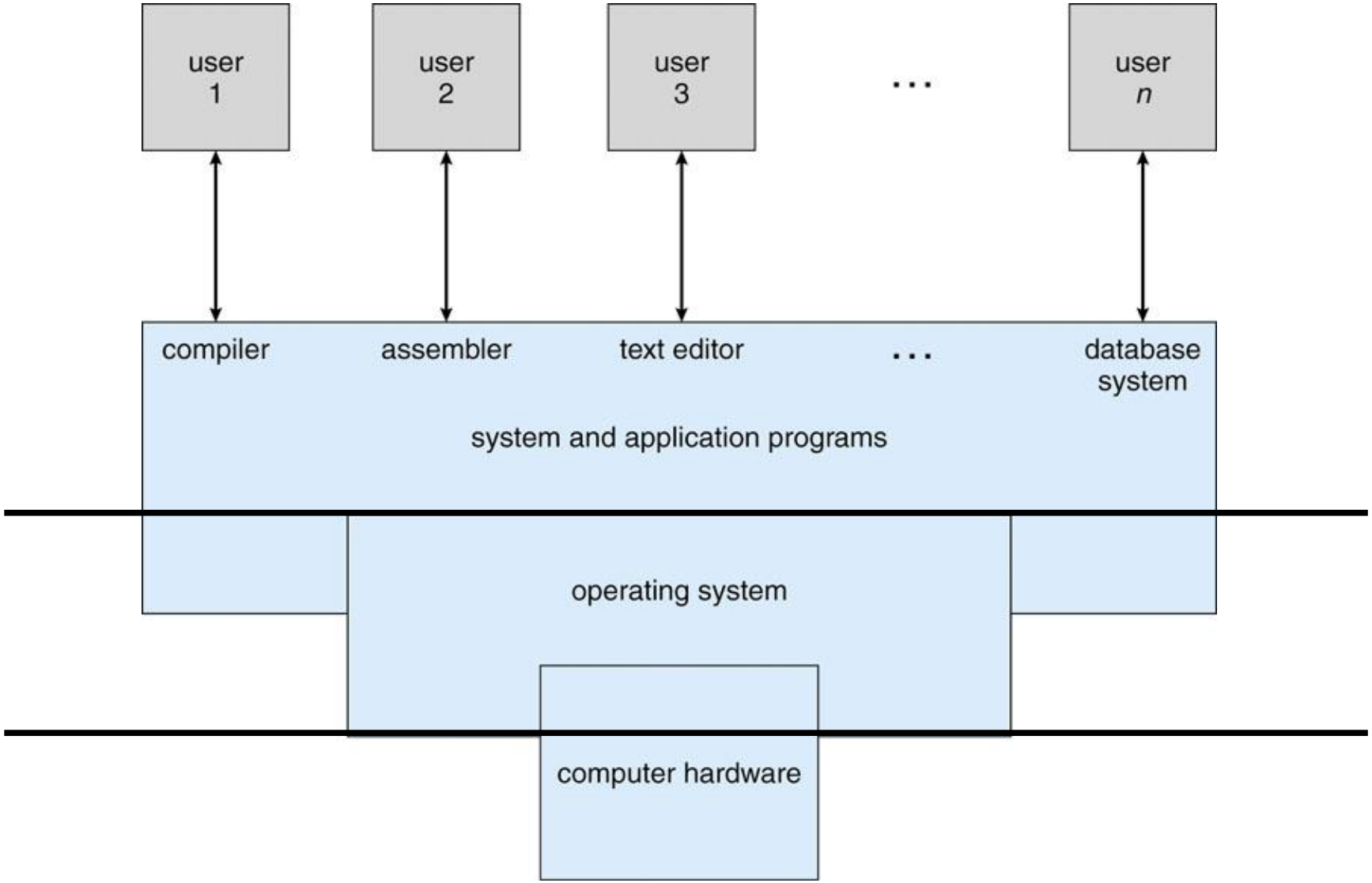
Background on operating systems

- Consider malicious programs running on a computer
  - A program trying to hog *all* CPU to mine coins
    - Program wants to run forever and not let other apps run
  - A program trying to read memory of another (sensitive) program and leak data
  - Ransomware wants to encrypt your hard drive and ask for bitcoins later

What does an OS do?

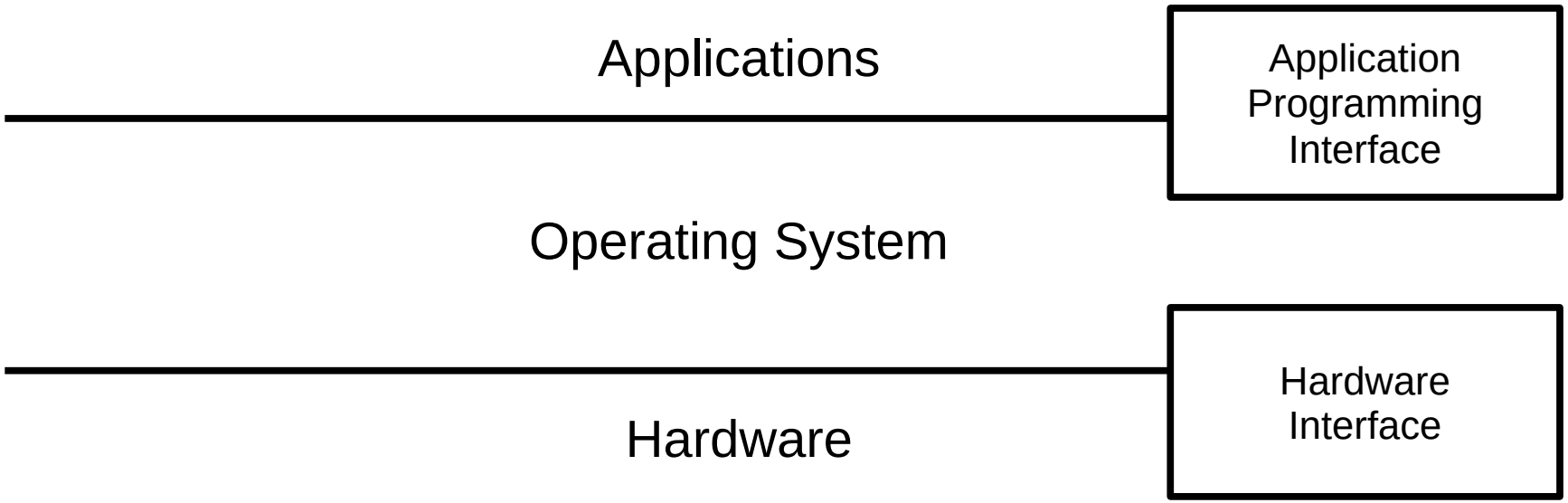


What does an OS do?



What does an OS do?

- OS interposes between applications and hardware



Background on operating systems

- Consider malicious programs running on a computer
  - A program trying to hog *all* CPU to mine coins
    - Program wants to run forever and not let other apps run
    - Protect CPU
  - A program trying to read memory of another (sensitive) program and leak data
    - Protect memory
  - Ransomware wants to encrypt your hard drive and ask for bitcoins later
    - Protect I/O devices

Background on operating systems

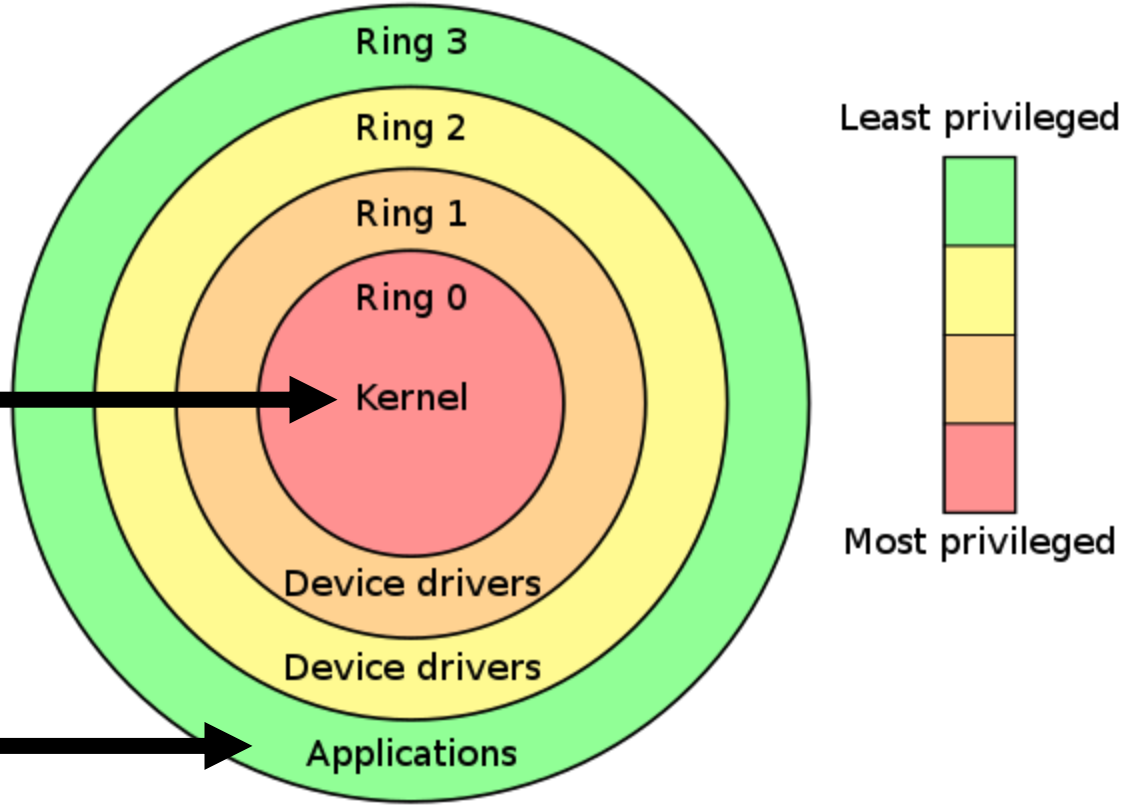
- Consider malicious programs running on a computer
  - A program trying to hog *all* CPU to mine coins
    - Program wants to run forever and not let other apps run
    - Protect CPU
  - A program trying to read memory of another (sensitive) program and leak data
    - Protect memory
  - Ransomware wants to encrypt your hard drive and ask for bitcoins later
    - Protect I/O devices

Hardware support for isolation

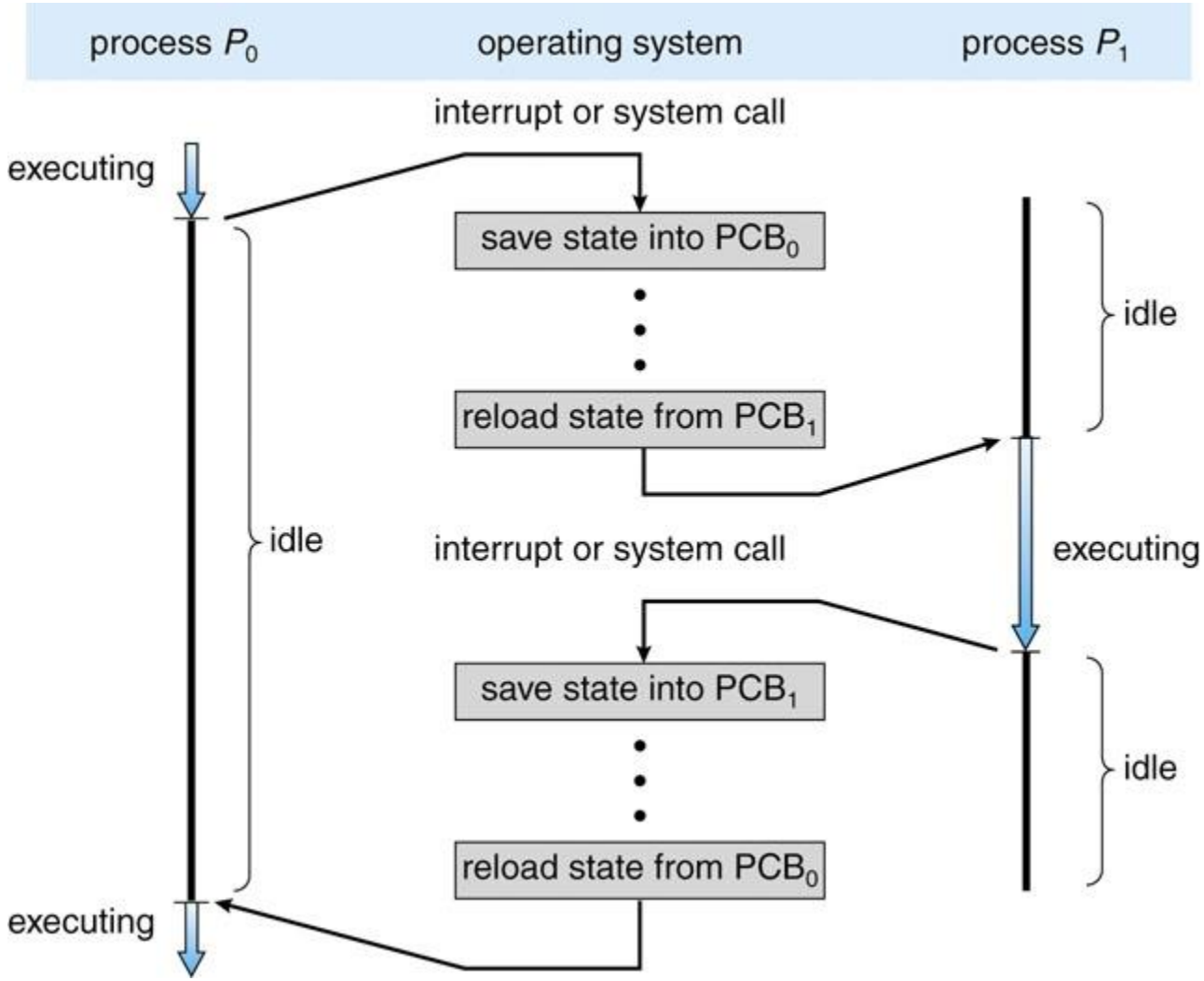
- Protection rings

Operating system can run all instructions, complete control

Applications run with less privileges, cannot run some instructions (e.g., I/O)

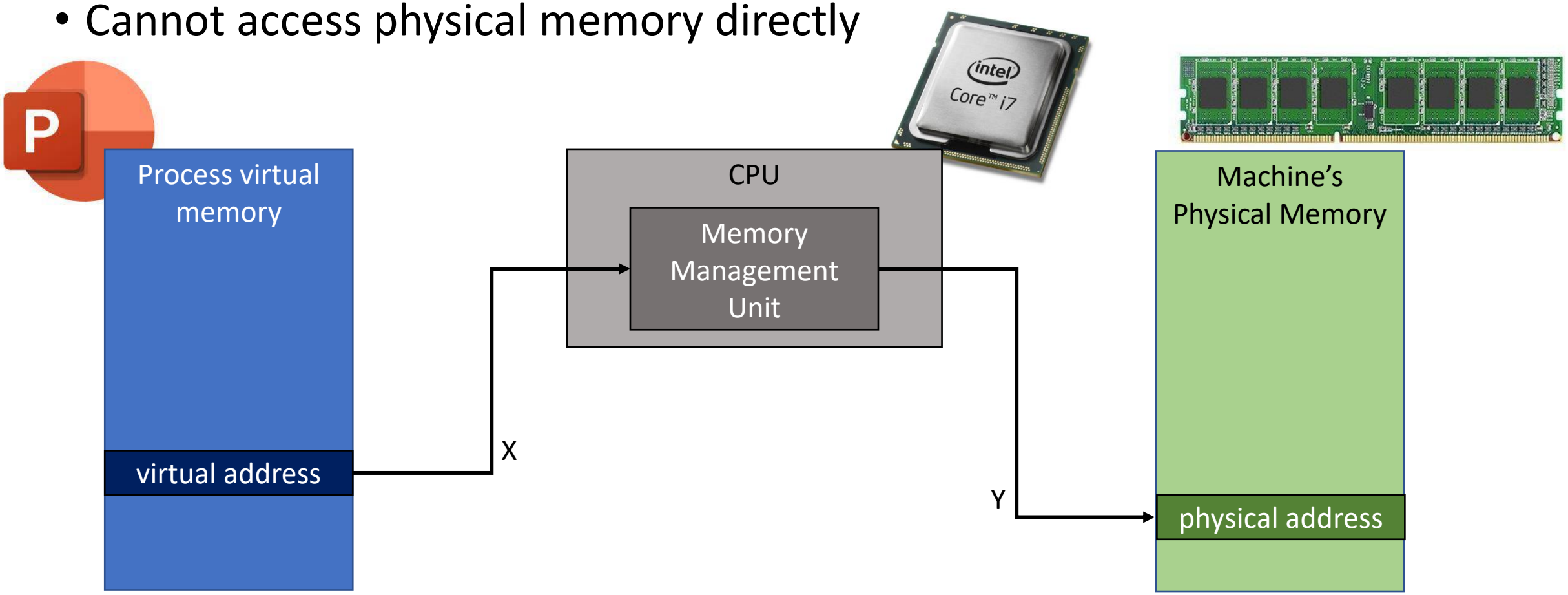


Operating system schedules CPU



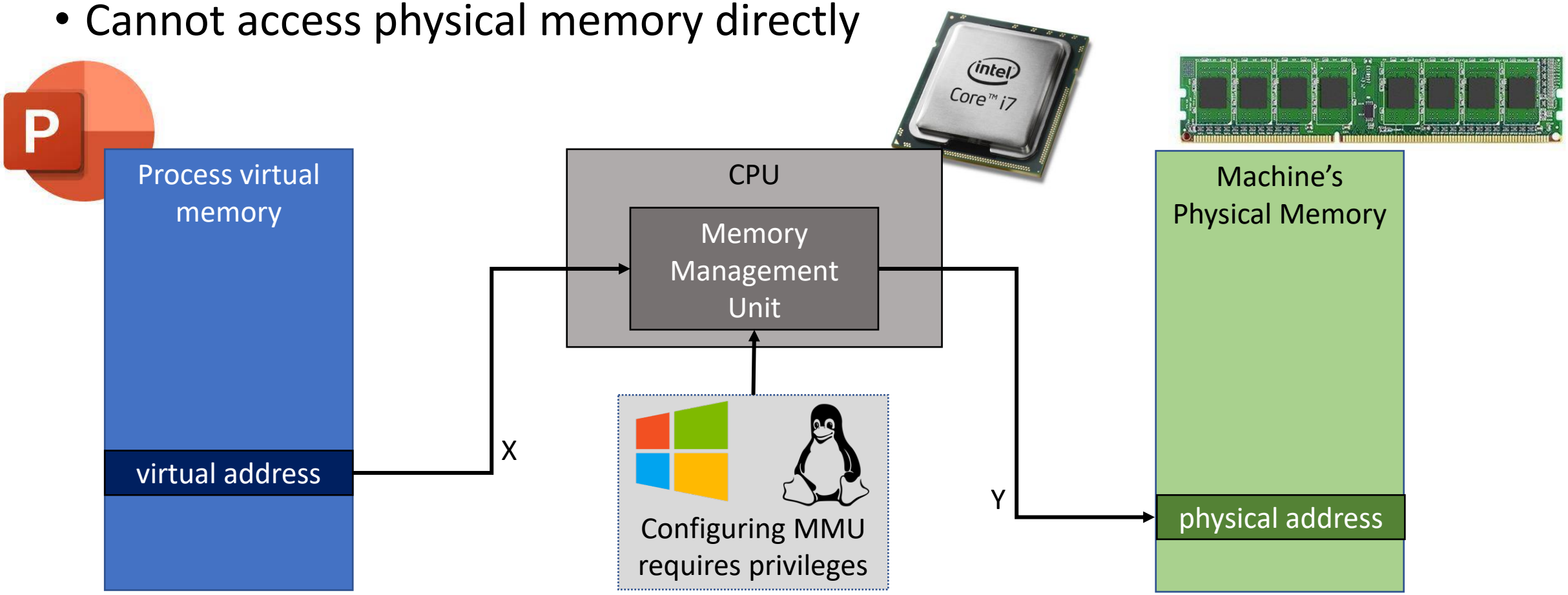
Operating system allocates memory

- Processes operate on virtual memory
- Cannot access physical memory directly

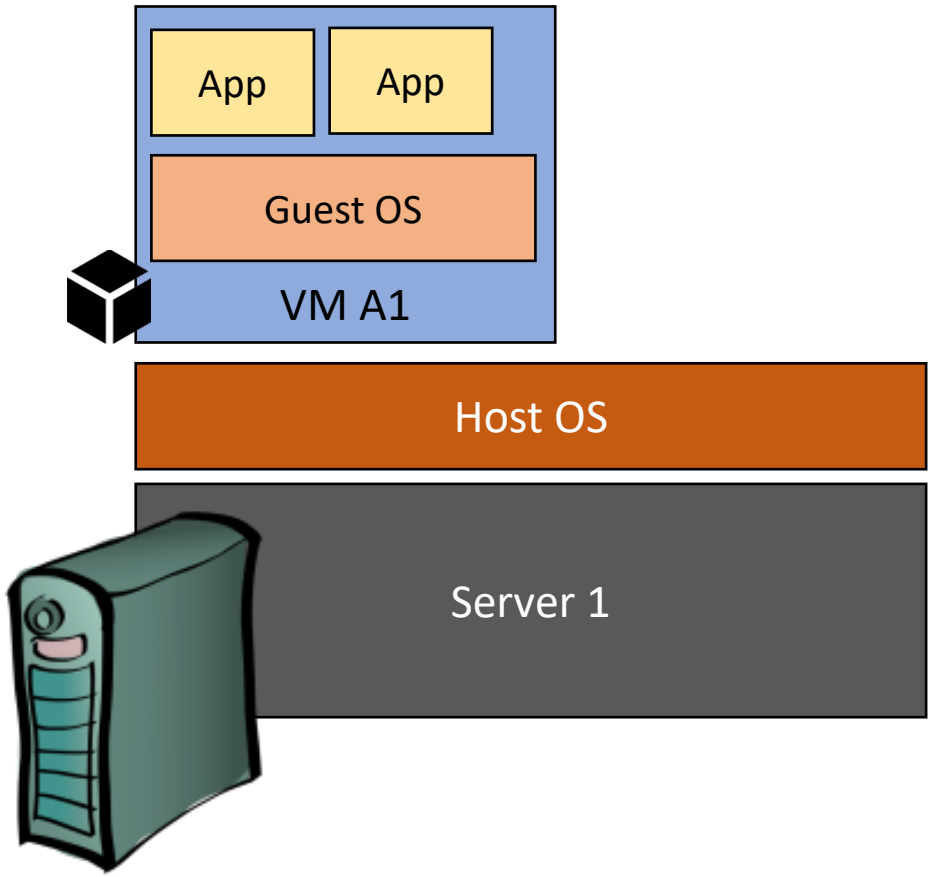


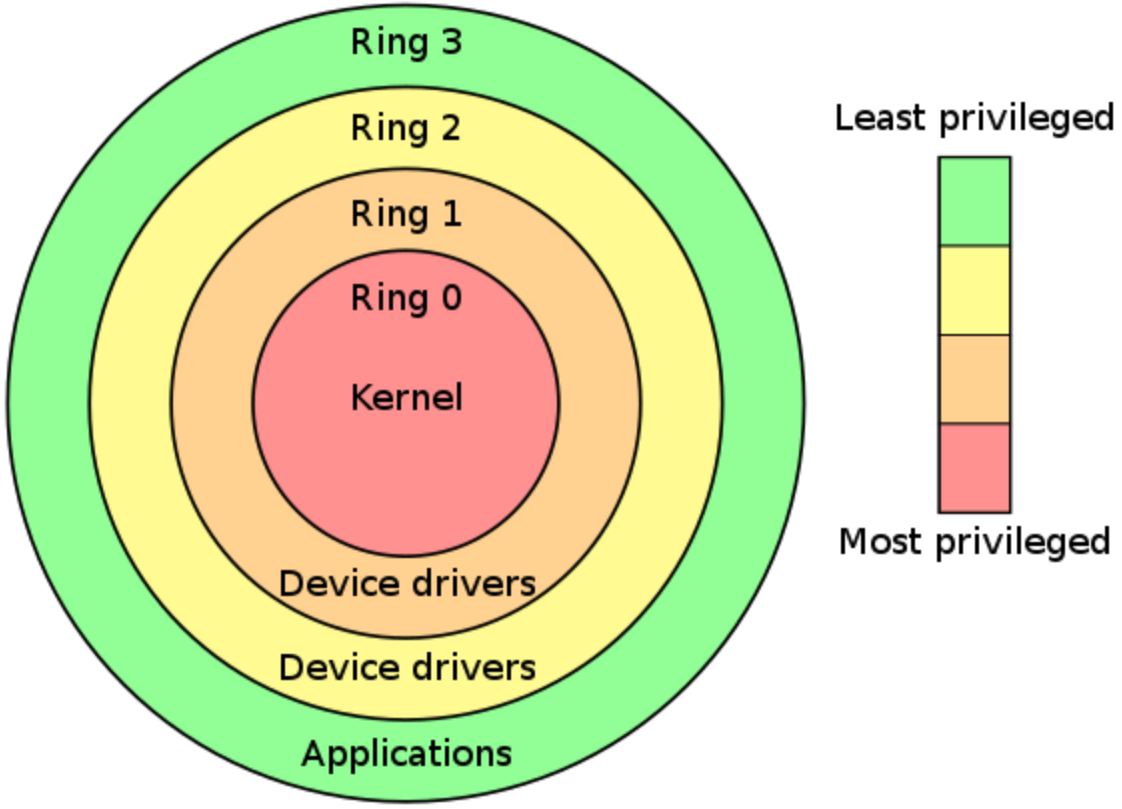
Operating system allocates memory

- Processes operate on virtual memory
- Cannot access physical memory directly

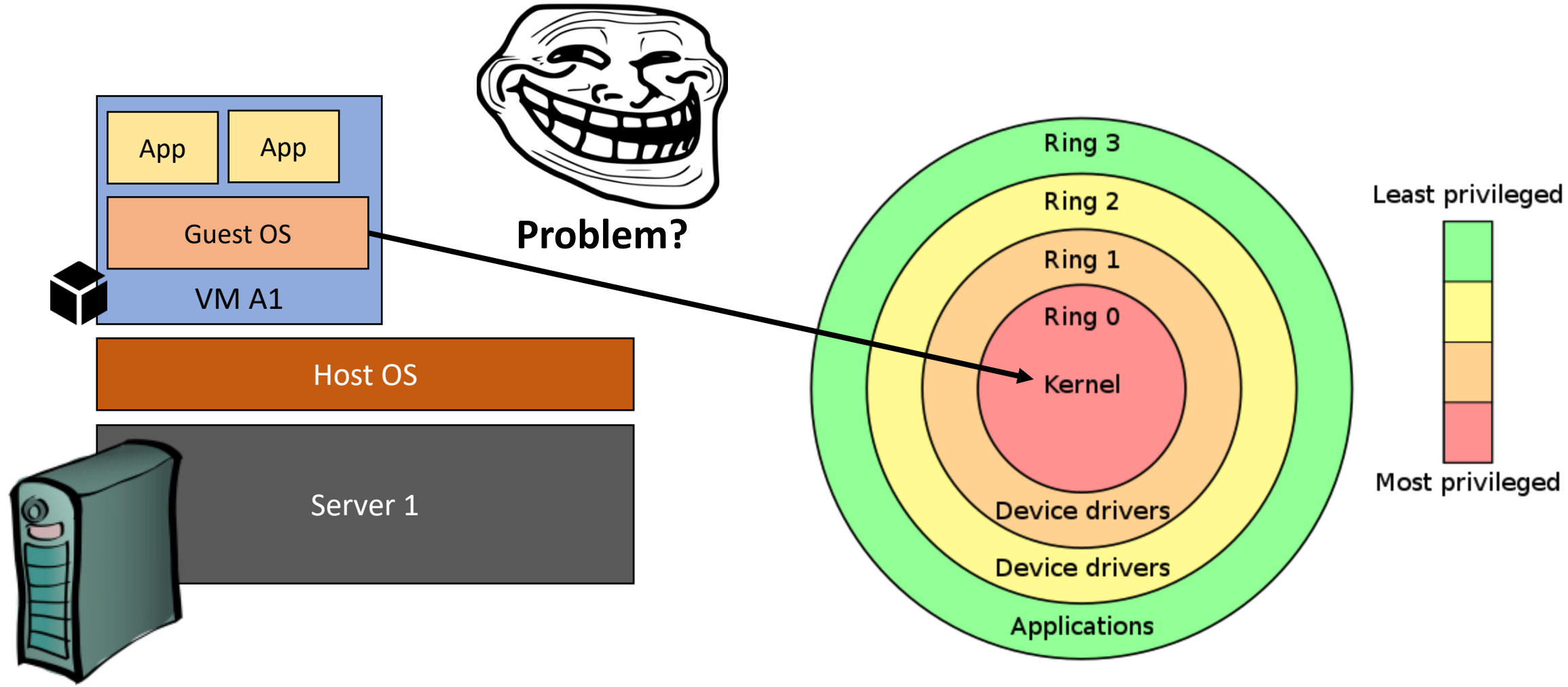


But virtual machines have kernels!





But virtual machines have kernels!



Virtualization needs to protect hardware!

- Consider malicious programs running on a computer
  - A program trying to hog *all* CPU to mine coins
  - A program trying to read memory of another (sensitive) program and leak data
  - Ransomware wants to encrypt your hard drive and ask for bitcoins later
- Consider malicious virtual machines running on a server
  - A VM kernel trying to hog *all* CPU to mine coins
  - A VM kernel trying to read memory of another (sensitive) VM and leak data
  - A VM kernel trying to encrypt the server's hard drive and ask for ransom

CompSci 401: Cloud Computing

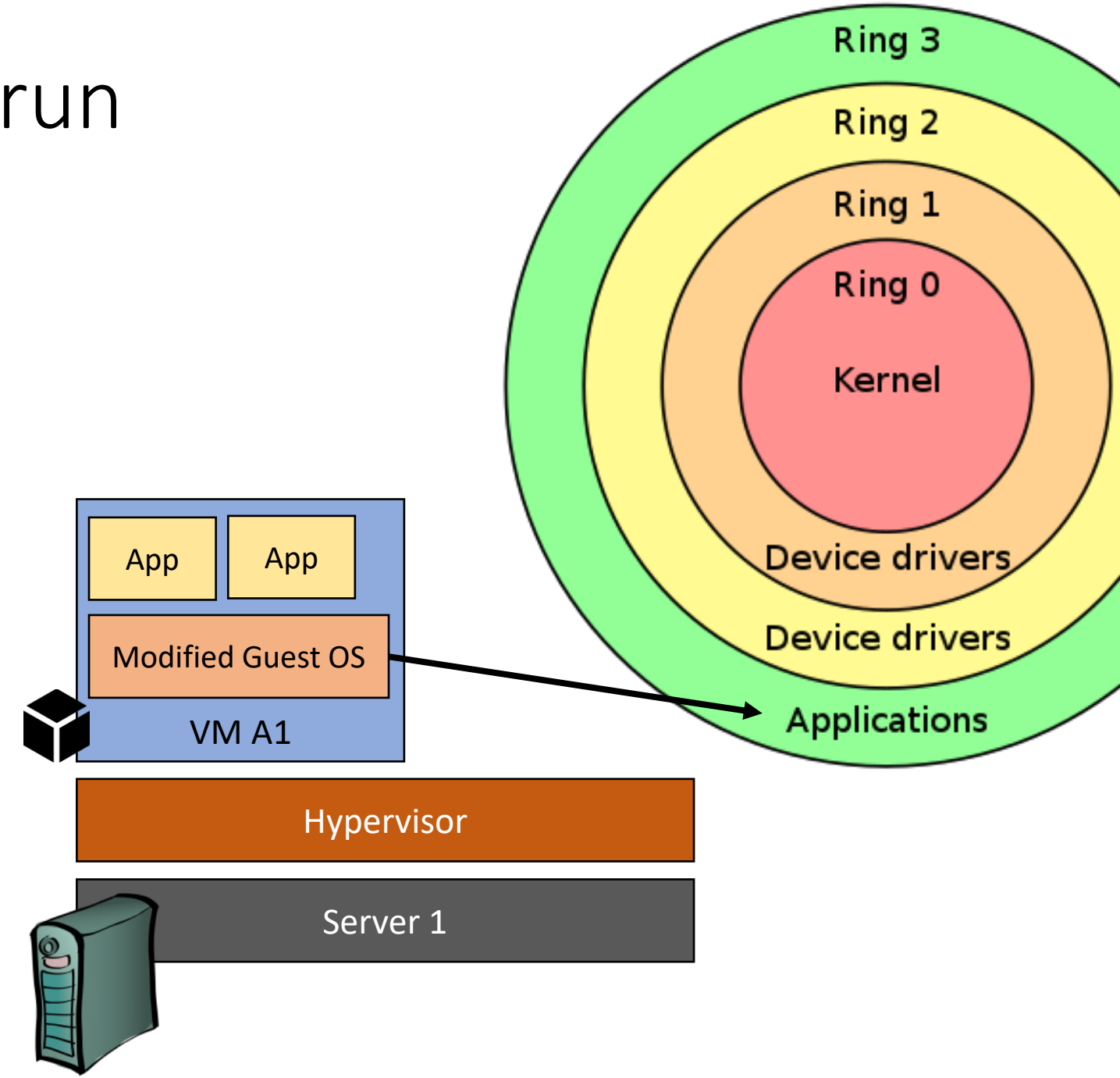
Paravirtualization

Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Modify guest OS to run without privileges

- Guest OS asks *hypervisor* whenever it needs to run a privileged operation
- Replace privileged instructions with calls to hypervisor
  - For example
    - MMU configuration
    - I/O



Native execution

- Guest OS and applications execute native code (that is, x86)
  - No emulation, high performance
- Privileges instructions in guest OS must be rewriten to call hypervisor
  - Incurs some overhead

Native execution

- Guest OS and applications execute native applications
  - No emulation, high performance
- Privileges instructions in guest OS must be rewriten to call hypervisor
  - Incurs some overhead

- Modifying the guest OS is the hardest part

CompSci 401: Cloud Computing

Full Virtualization

Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Properties of full virtualization

- Transparency
  - To a tenant, a virtual machine is just like the hardware
  - Operating systems run unmodified
- Isolation
  - Virtual machines should be isolated from one another
  - VMs and guest OSes must not be able to take control of the hardware
- Efficiency
  - Low overhead, close-to-native performance

Properties of full virtualization

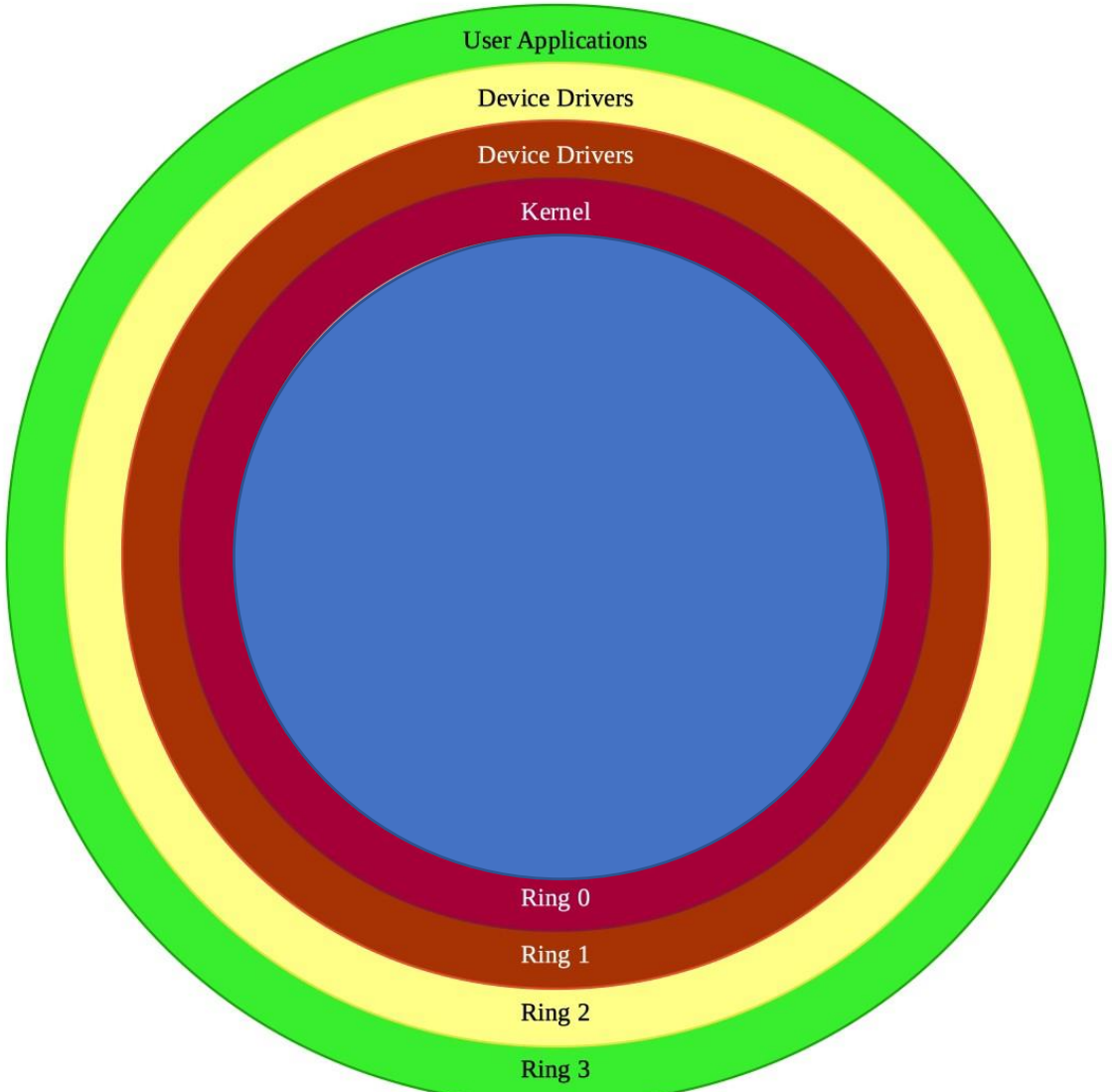
- Transparency
  - To a tenant, a virtual machine is just like the hardware
  - Operating systems run unmodified ← ~~Paravirtualization~~
- Isolation
  - Virtual machines should be isolated from one another
  - VMs and guest OSes must not be able to take control of the hardware
- Efficiency
  - Low overhead, close-to-native performance ← ~~Emulation~~

How does full virtualization achieve all goals?

- Hardware support

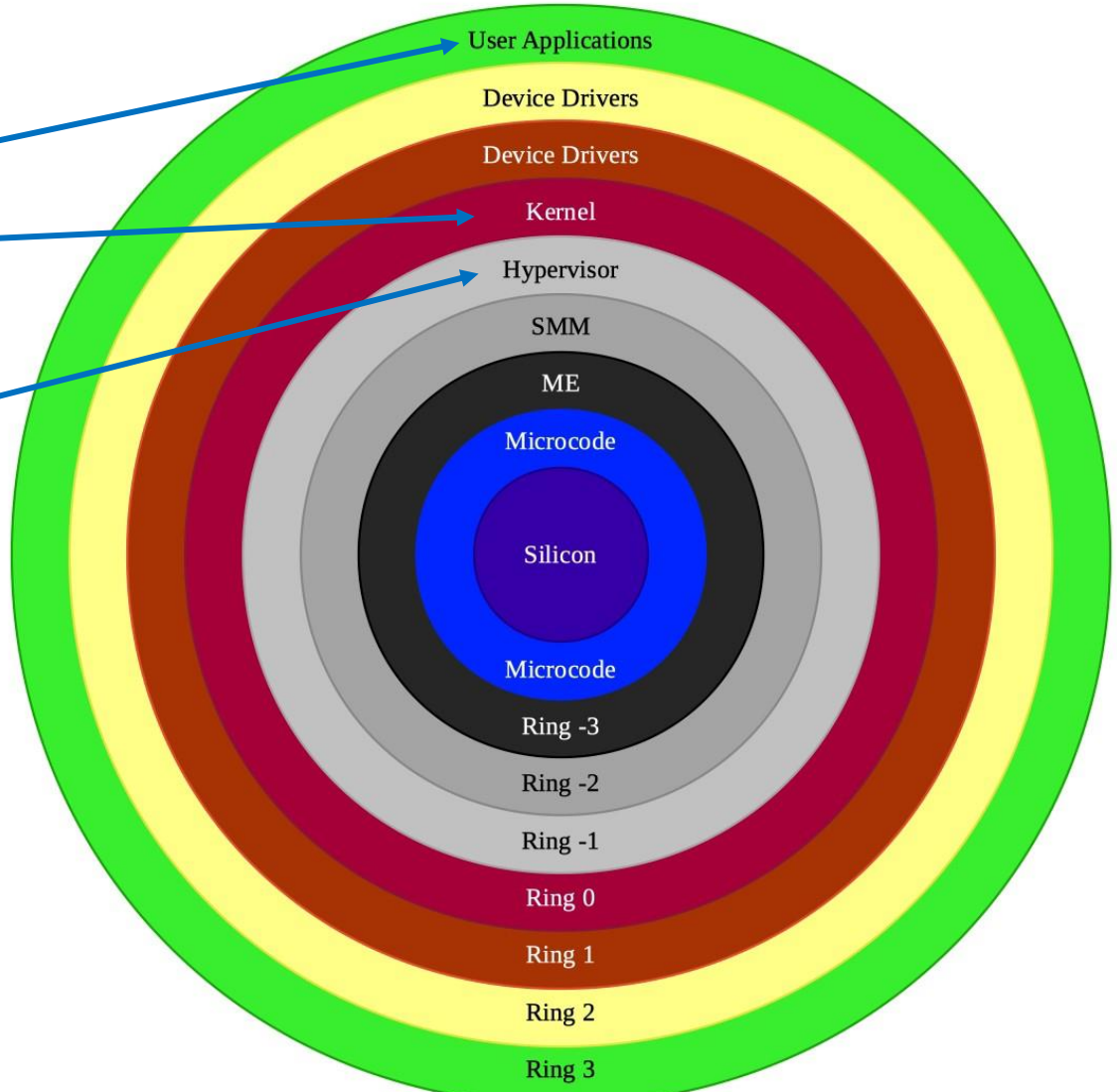
How does full virtualization achieve all goals?

- Hardware support



How does full virtualization achieve all goals?

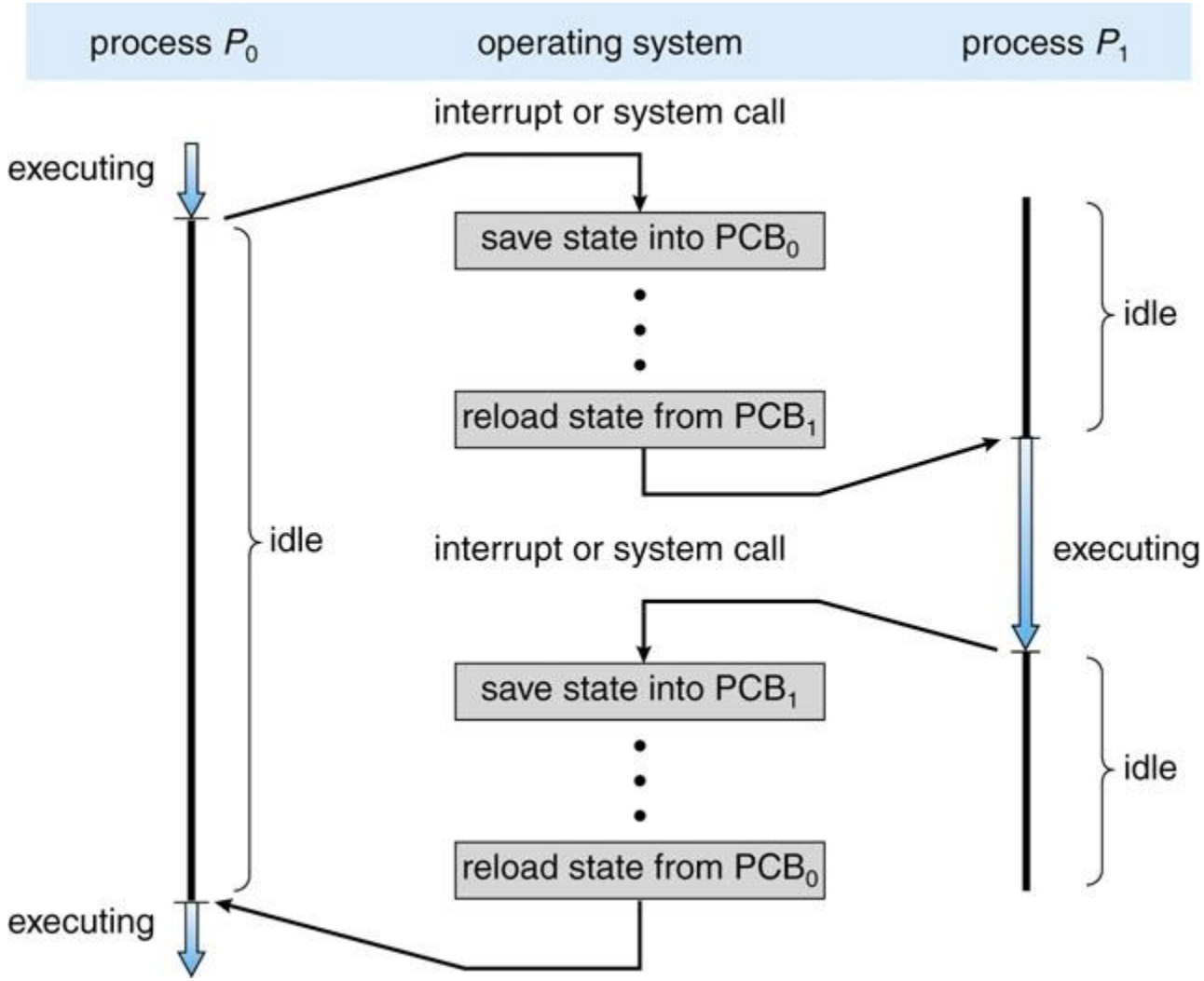
- Hardware support
  - Apps run in ring 3, unprivileged
  - Guest OS runs in ring 0
    - Some privileges
    - Manages apps
  - Hypervisor runs in ring -1
    - More privileges
    - Manages virtual machines



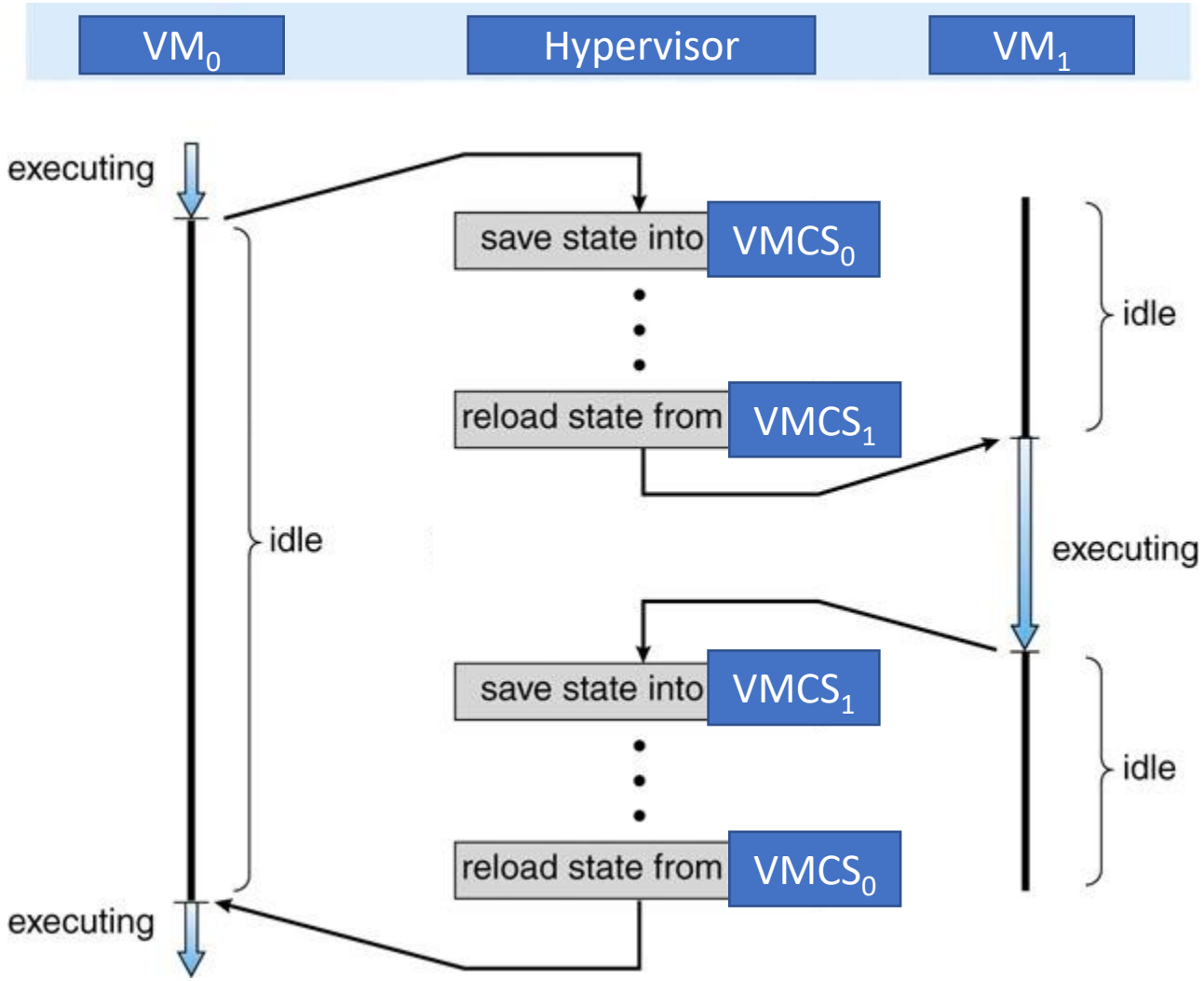
Hypervisor mediates access to hardware

- Hypervisor allocates CPU to a virtual machine
  - Virtual machine runs until its kernel needs to perform a privileged operation
  - Hypervisor can preempt virtual machine and take back control of the CPU
- Hypervisor allocates memory to a virtual machine
  - Virtual machine runs on virtual memory
  - Translation of virtual to physical addresses ultimately controlled by hypervisor
- Hypervisor mediates a virtual machine’s access to I/O devices

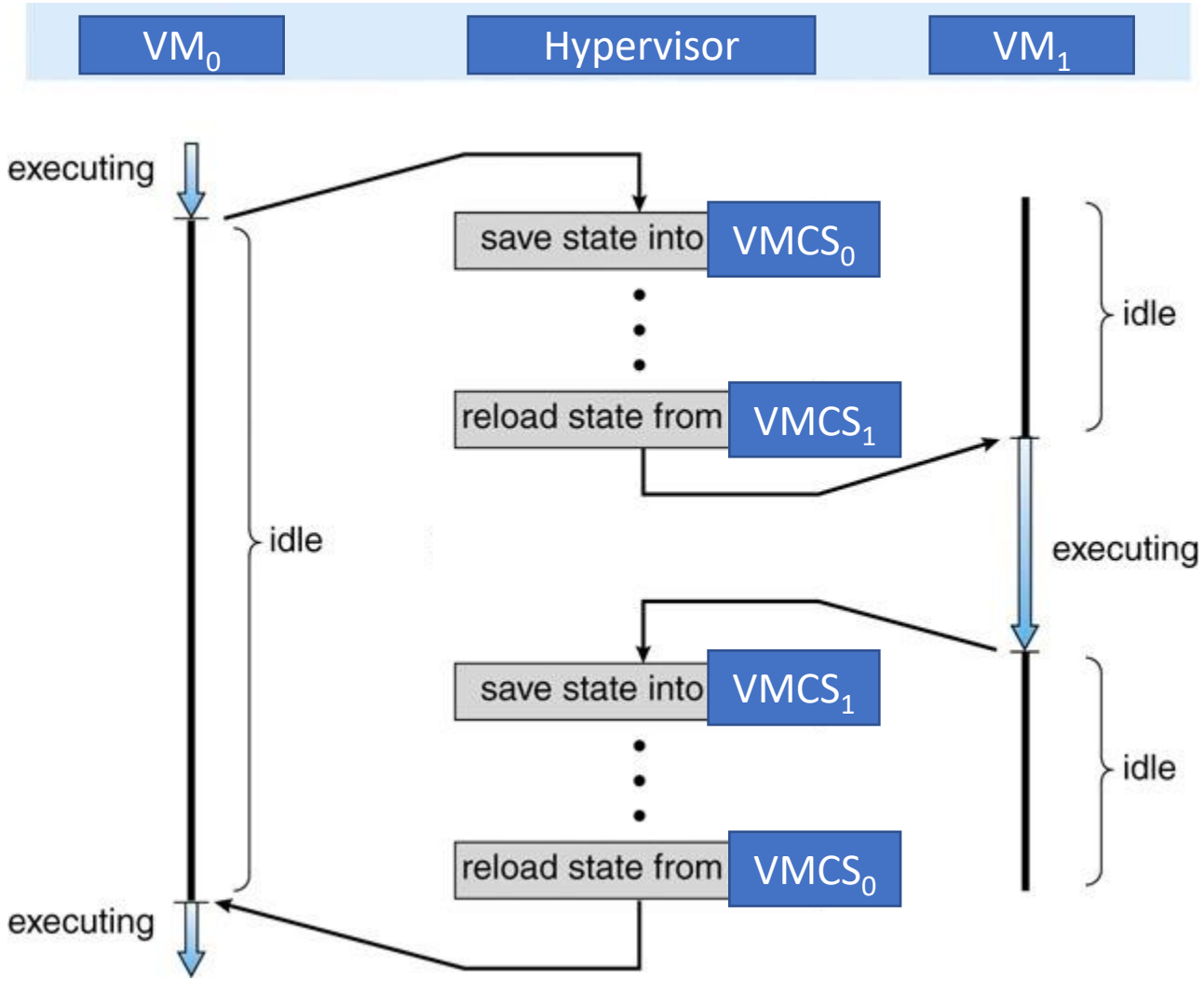
Operating system schedules CPU



Hypervisor schedules CPU

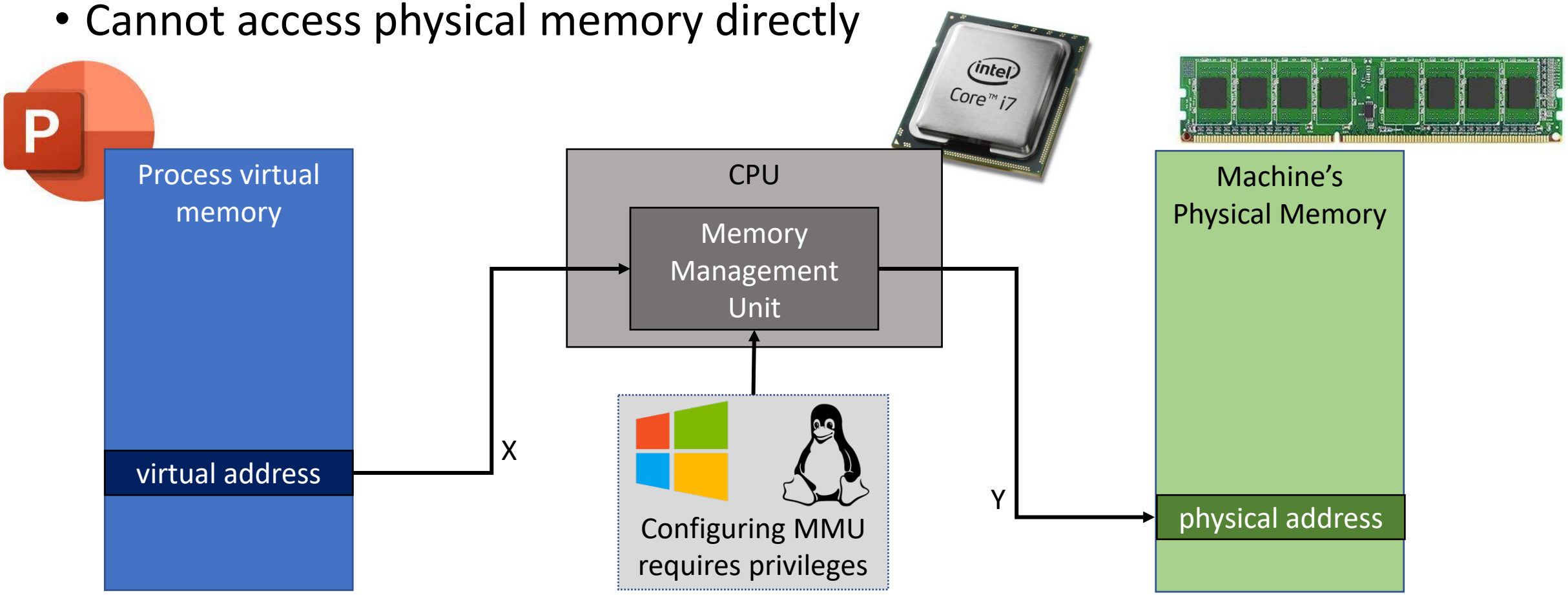


Hypervisor schedules CPU



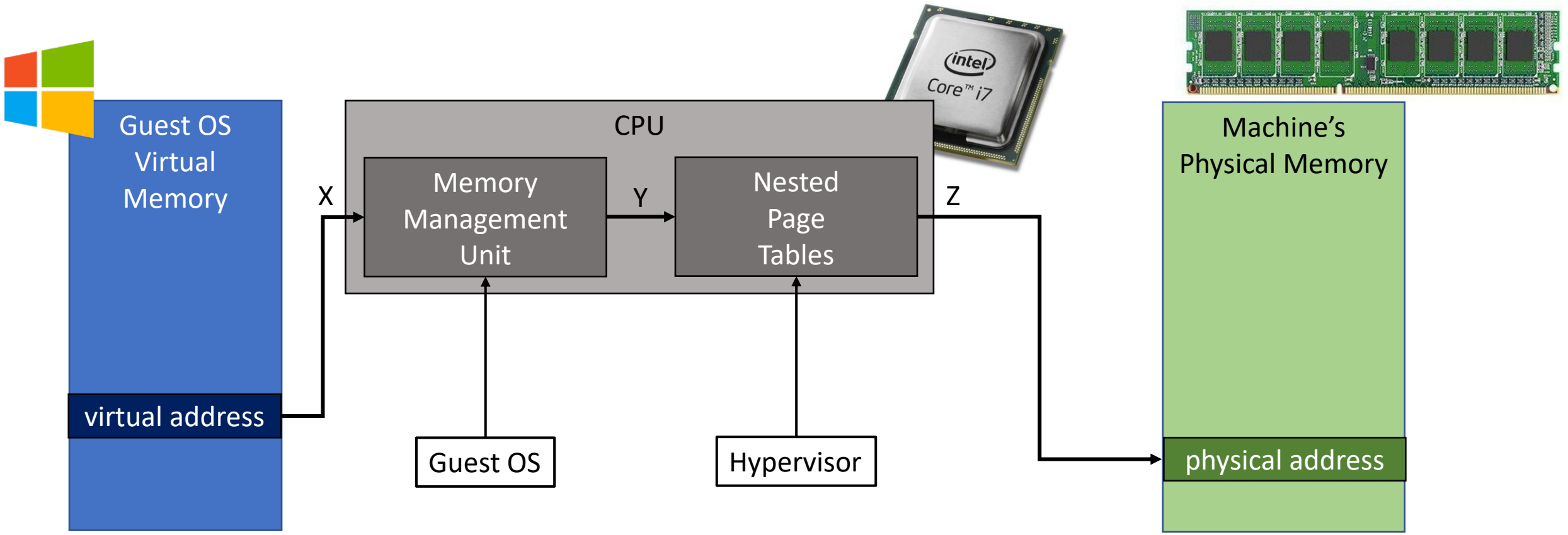
Operating system allocates memory

- Processes operate on virtual memory
- Cannot access physical memory directly



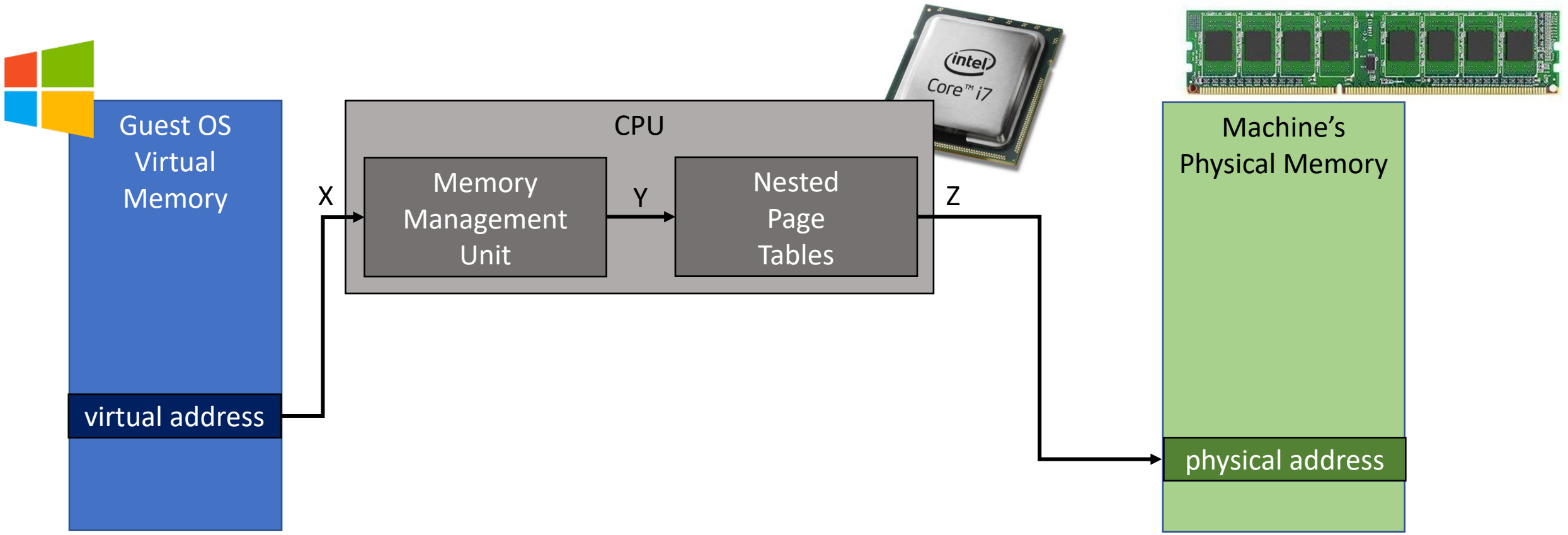
Hypervisor allocates memory

- Guest OS operates on virtual memory
- Cannot access physical memory directly



Hypervisor allocates memory

- Guest OS operates on virtual memory
- Cannot access physical memory directly



Comparison of virtualization approaches

	Emulation	Paravirtualization	Full virtualization
Idea	Emulate instructions from one architecture on another	Modify guest operating systems to run unprivileged	Hypervisor runs with more privileges than guest OS
Performance	Low	High	Native
Software	Any architecture	Modified operating systems, same arch	Unmodified, same architecture
Hardware	No requirements	No requirements	Privilege isolation
Isolation	Complete	Complete	Complete

CompSci 401: Cloud Computing

Virtual I/O

Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Unmodified OSes will access I/O hardware

- Operating system enumerates I/O devices on boot
  - Keyboard, mouse, sound cards, GPU, network cards, disks
- Operating systems have device drivers to control I/O devices

Unmodified OSes will access I/O hardware

- Operating system enumerates I/O devices on boot
  - Keyboard, mouse, sound cards, GPU, network cards, disks
- Operating systems have device drivers to control I/O devices

- How can a hypervisor let the OS access a device and still guarantee that the OS will not perform malicious operations?
  - For example, encrypt the whole disk

Unmodified OSes will access I/O hardware

- Operating system enumerates I/O devices on boot
  - Keyboard, mouse, sound cards, GPU, network cards, disks
- Operating systems have device drivers to control I/O devices

- How can a hypervisor let the OS access a device and still guarantee that the OS will not perform malicious operations?
  - For example, encrypt the whole disk
- Even if guest OSes are trusted, devices must be usable by all VMs

Virtual I/O devices

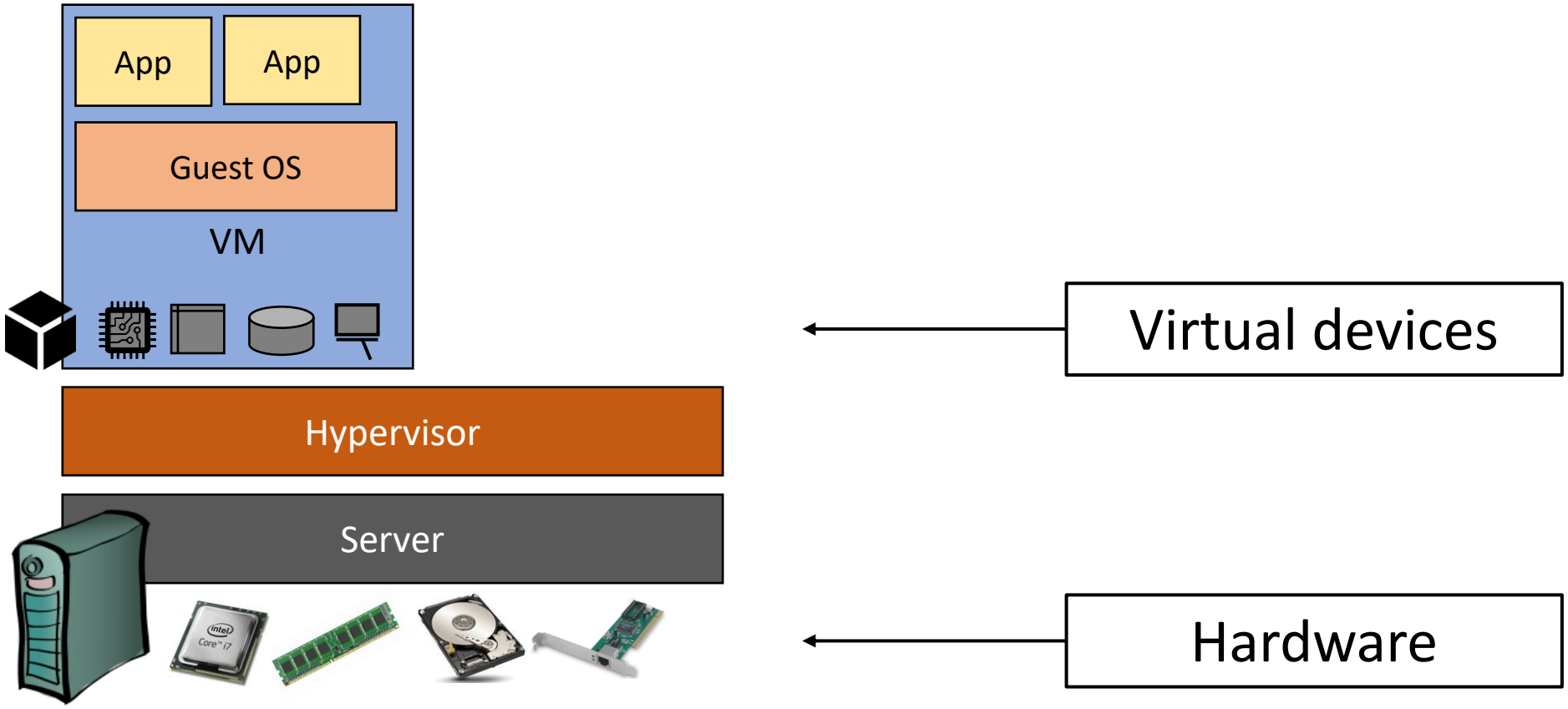
- VMs and guest OSes do not access hardware I/O devices
- VMs and guest OSes access virtual devices controlled by the hypervisor
- Virtual devices execute operations on hardware I/O devices

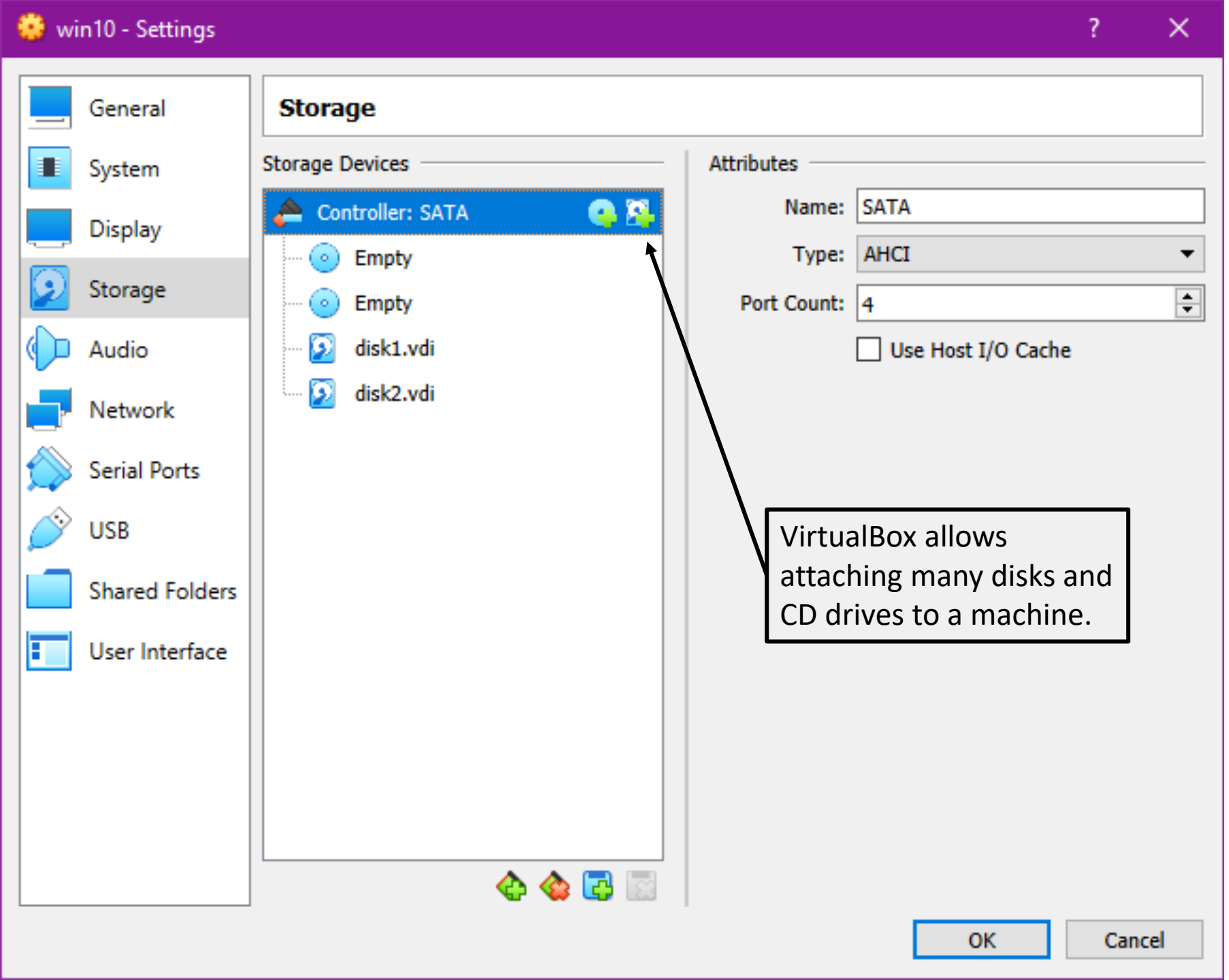
Virtual I/O devices

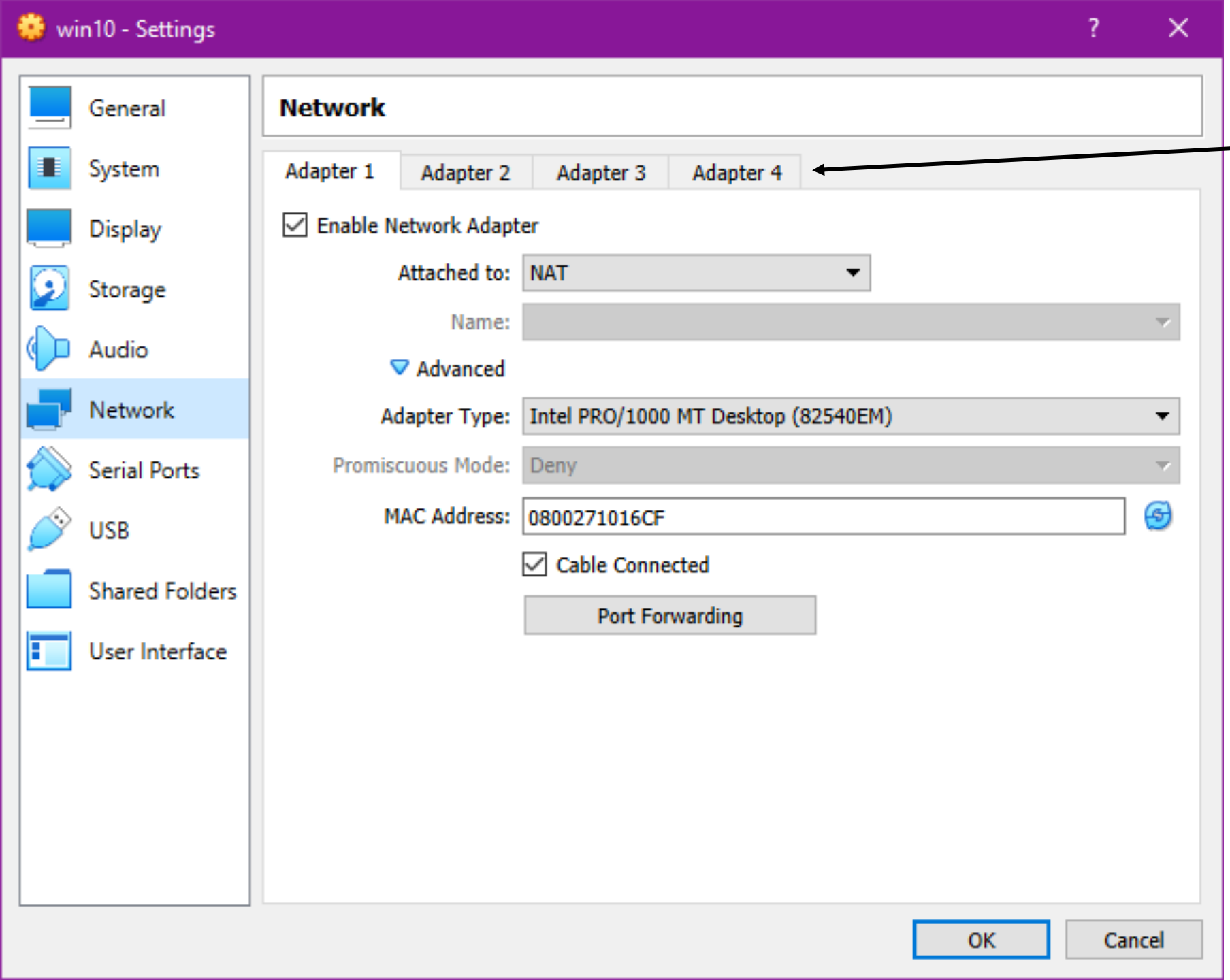
- VMs and guest OSes do not access hardware I/O devices
- VMs and guest OSes access virtual devices controlled by the hypervisor
- Virtual devices execute operations on hardware I/O devices

- Example: OS tries to enumerate I/O devices
  - Hypervisor will be invoked on each enumeration request
  - The hypervisor will run code from the virtual I/O device and send a response
  - Guest OS cannot distinguish responses from virtual and hardware I/O devices

Virtual I/O devices



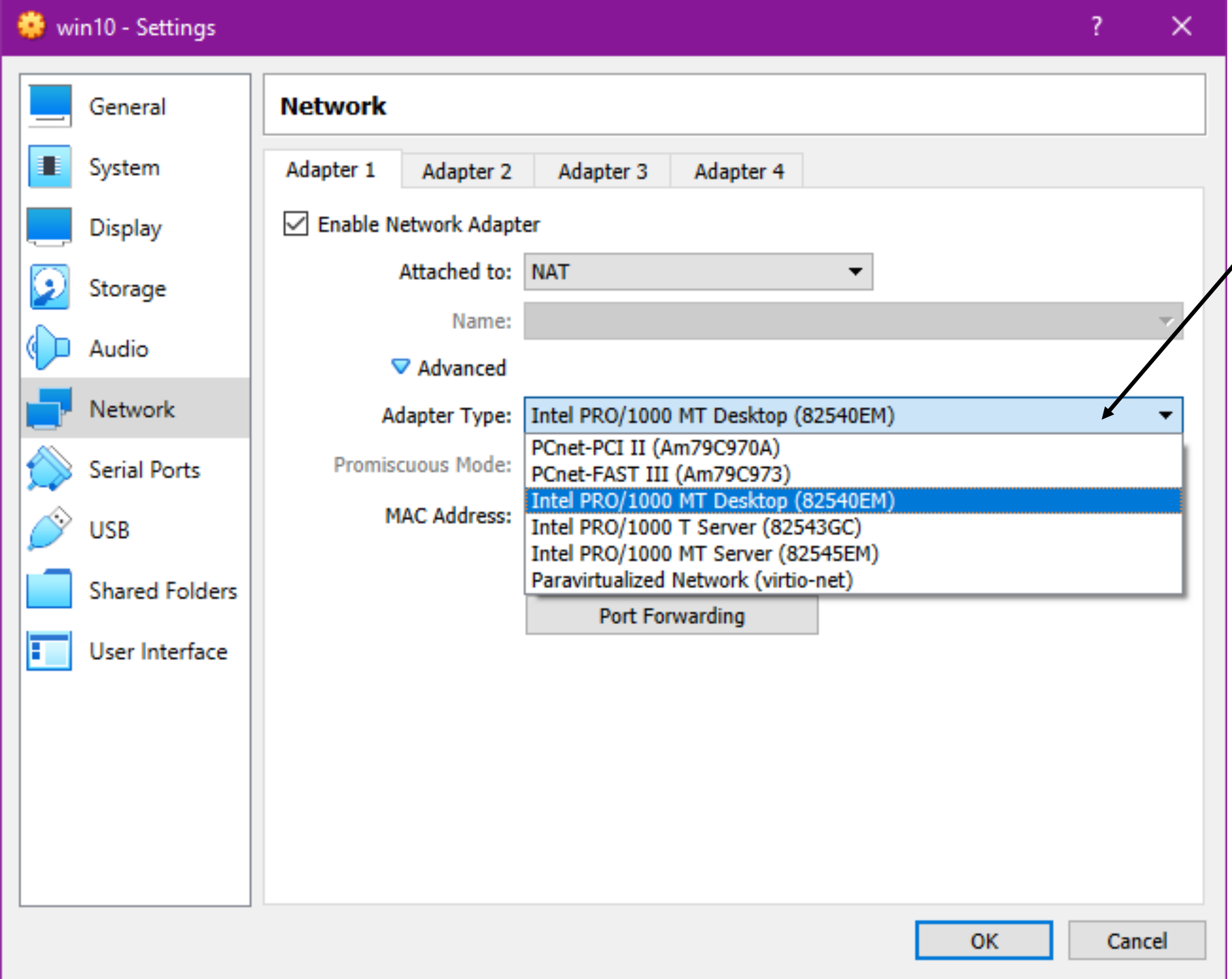

win10 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Storage
Storage Devices
Controller: SATA
Empty
Empty
disk1.vdi
disk2.vdi
Attributes
Name: SATA
Type: AHCI
Port Count: 4
Use Host I/O Cache
VirtualBox allows attaching many disks and CD drives to a machine.
OK
Cancel



VirtualBox allows attaching up to four network cards to a VM.

Implementing virtual devices

- Virtual devices can mimic existing devices
  - Guest OS will likely have drivers
  - Harder to implement in the hypervisor
    - Hardware may have many details, including bugs, which need to be replicated
- Virtual devices can implement new, imaginary devices
  - Guest OS may lack drivers
  - Can design a new, clean, and easy to implement interface



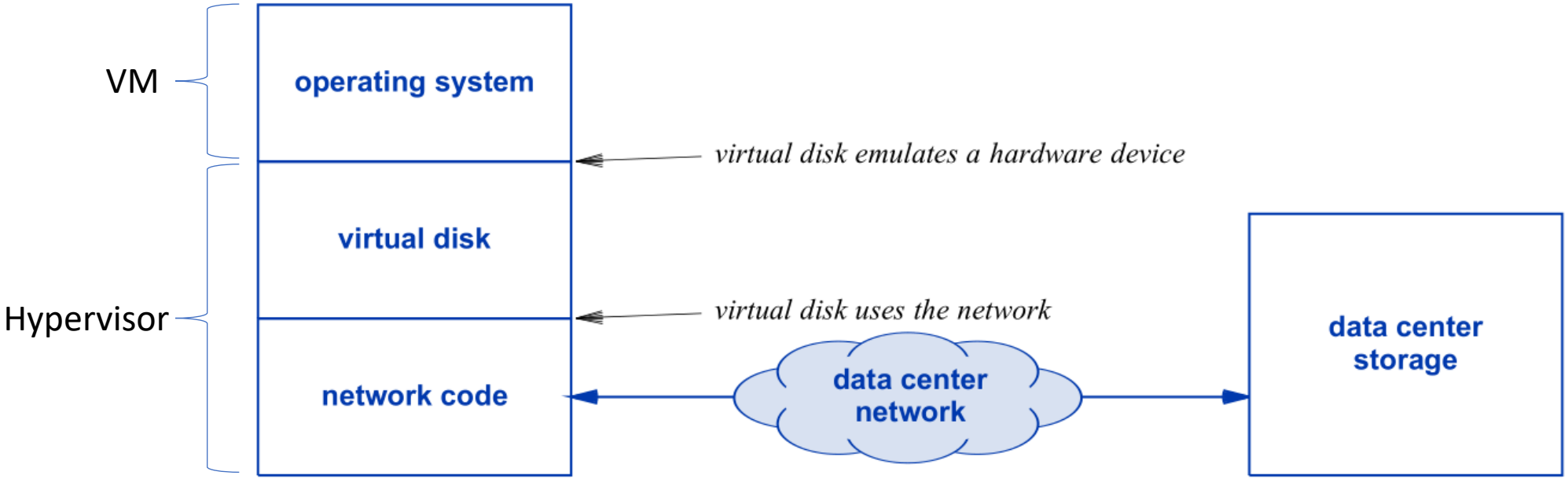
VirtualBox allows us to choose which hardware device a virtual network card will mimic.

Virtualized disks

- In VirtualBox, a virtual disk reads and writes to a file on the host

Virtualized disks

- In VirtualBox, a virtual disk reads and writes to a file on the host
- In a datacenter, a virtual disk may exchange data over the network



CompSci 401: Cloud Computing

Virtual Machines as Objects

Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Virtual machines as objects

- Virtual machine created and managed by software
- Hypervisor keeps track of VM resources
  - CPU state
  - Allocated memory
  - State of virtual I/O devices
    - Including any data in storage devices (remote or local)

Virtual machines as objects

- Virtual machines are created and managed by software
- Hypervisor keeps track of VM resources
  - CPU state
  - Allocated memory
  - State of virtual I/O devices
    - Including any data in storage devices (remote or local)
- Virtual machines can be turned into a set of bytes

Virtual machine migration

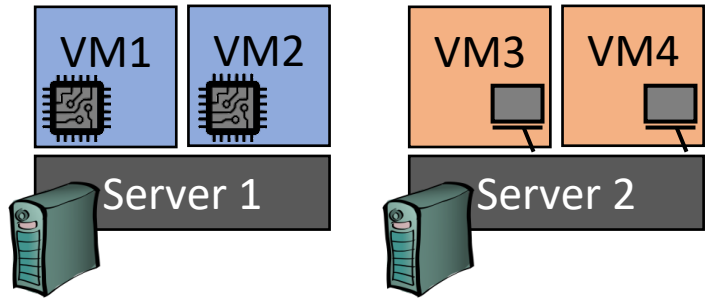
- Stop the virtual machine
- Turn the virtual machine into a set of bytes
- Move the virtual machine's data to another server
- Restart the virtual machine

- Benefits
  - Sidestep server failures
  - Control load on servers
    - Avoid server overload or underutilization
    - Load balancing

Virtual machine migration

- Stop the virtual machine
- Turn the virtual machine into a set of bytes
- Move the virtual machine's data to another server
- Restart the virtual machine

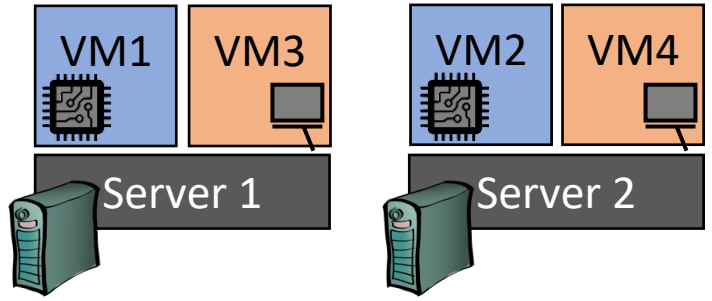
- Benefits
  - Sidestep server failures
  - Control load on servers
    - Avoid server overload or underutilization
    - Load balancing



Virtual machine migration

- Stop the virtual machine
- Turn the virtual machine into a set of bytes
- Move the virtual machine's data to another server
- Restart the virtual machine

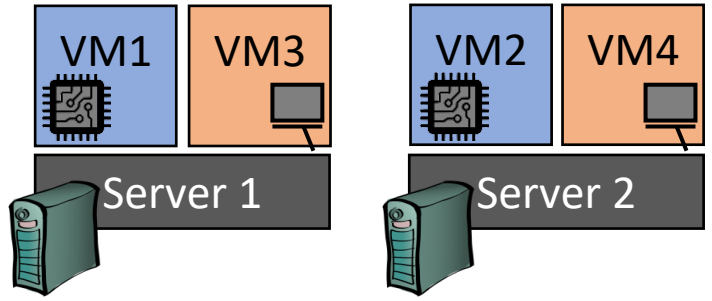
- Benefits
  - Sidestep server failures
  - Control load on servers
    - Avoid server overload or underutilization
    - Load balancing



Virtual machine migration

- Stop the virtual machine
- Turn the virtual machine into a set of bytes
- Move the virtual machine's data to another server
- Restart the virtual machine

- Benefits
  - Sidestep server failures
  - Control load on servers
    - Avoid server overload or underutilization
    - Load balancing
    - Making traffic local by gathering VMs close by (e.g., in a pod)

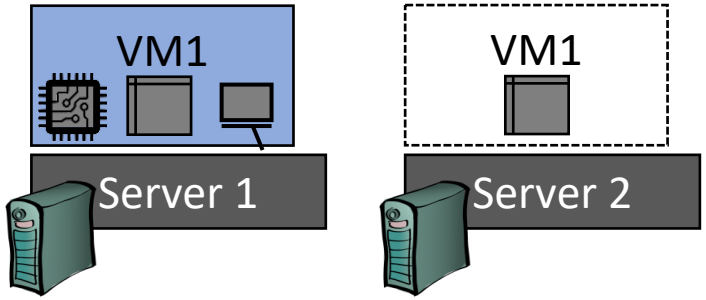


Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations

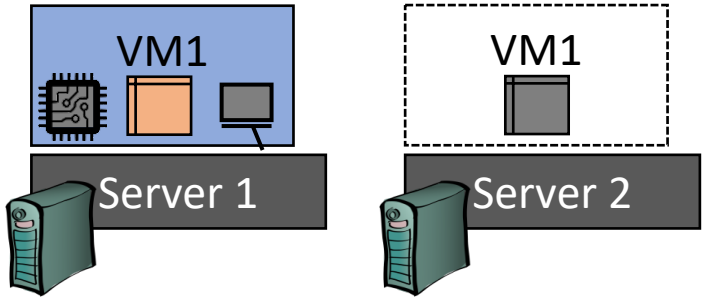
Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. **Pre-copy: Copy all the memory allocated to a VM**
  2. Stop-and-copy: Copy again any memory areas that have been changed
  3. Post-copy: Synchronize volatile state (CPU, I/O devices)



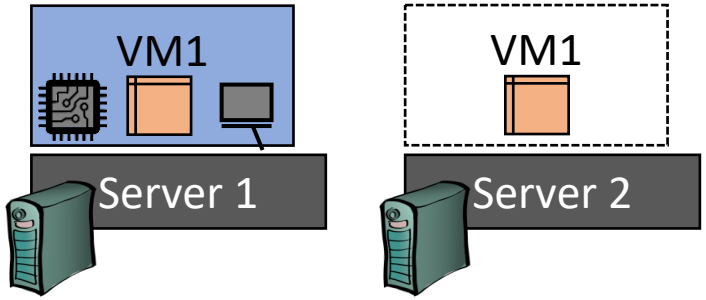
Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. Pre-copy: Copy all the memory allocated to a VM
  2. **Stop-and-copy: Copy again any memory areas that have been changed**
  3. Post-copy: Synchronize volatile state (CPU, I/O devices)



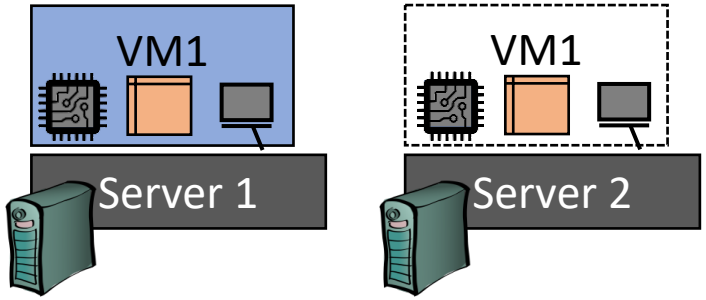
Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. Pre-copy: Copy all the memory allocated to a VM
  2. **Stop-and-copy: Copy again any memory areas that have been changed**
  3. Post-copy: Synchronize volatile state (CPU, I/O devices)



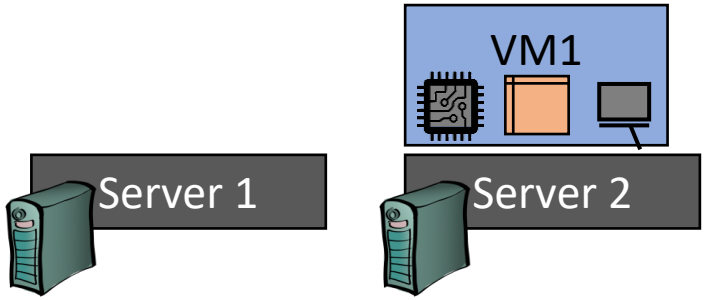
Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. Pre-copy: Copy all the memory allocated to a VM
  2. Stop-and-copy: Copy again any memory areas that have been changed
  3. **Post-copy: Synchronize volatile state (CPU, I/O devices)**



Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. Pre-copy: Copy all the memory allocated to a VM
  2. Stop-and-copy: Copy again any memory areas that have been changed
  3. Post-copy: Synchronize volatile state (CPU, I/O devices)



Live migration

- Stop-migrate-restart would interrupt network communication
  - Remote clients could give up on the connection and experience a failure
  - Tenants would be able to perceive migrations
- Live migration keeps the VM running during most of the migration
  1. Pre-copy: Copy all the memory allocated to a VM
  2. Stop-and-copy: Copy again any memory areas that have been changed
  3. Post-copy: Synchronize volatile state (CPU, I/O devices)

- Second and third stages are fast!

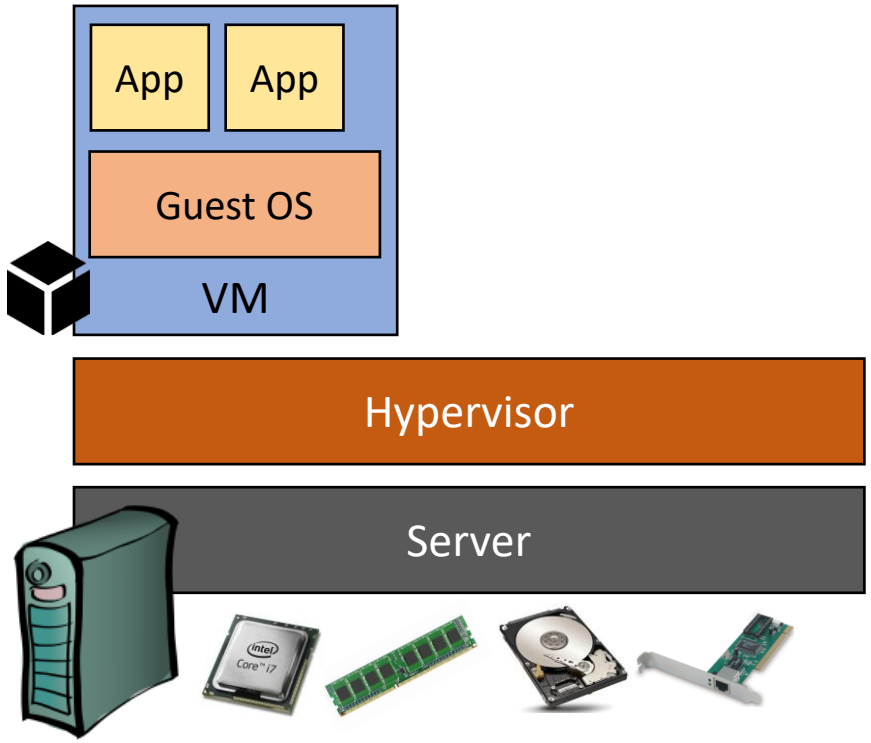
CompSci 401: Cloud Computing

Hosted Hypervisors

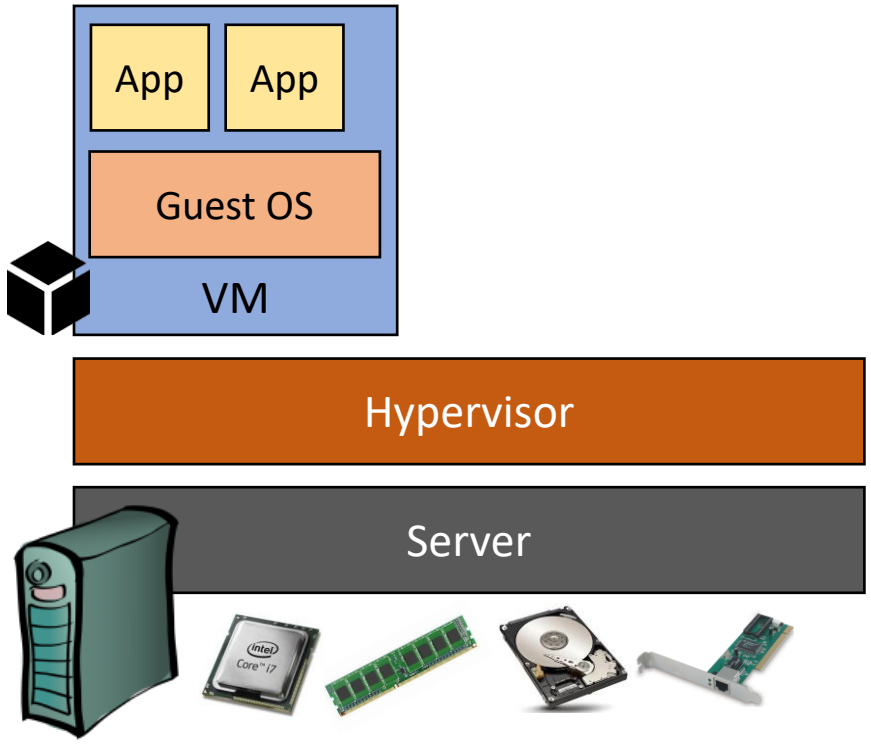
Prof. Ítalo Cunha

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

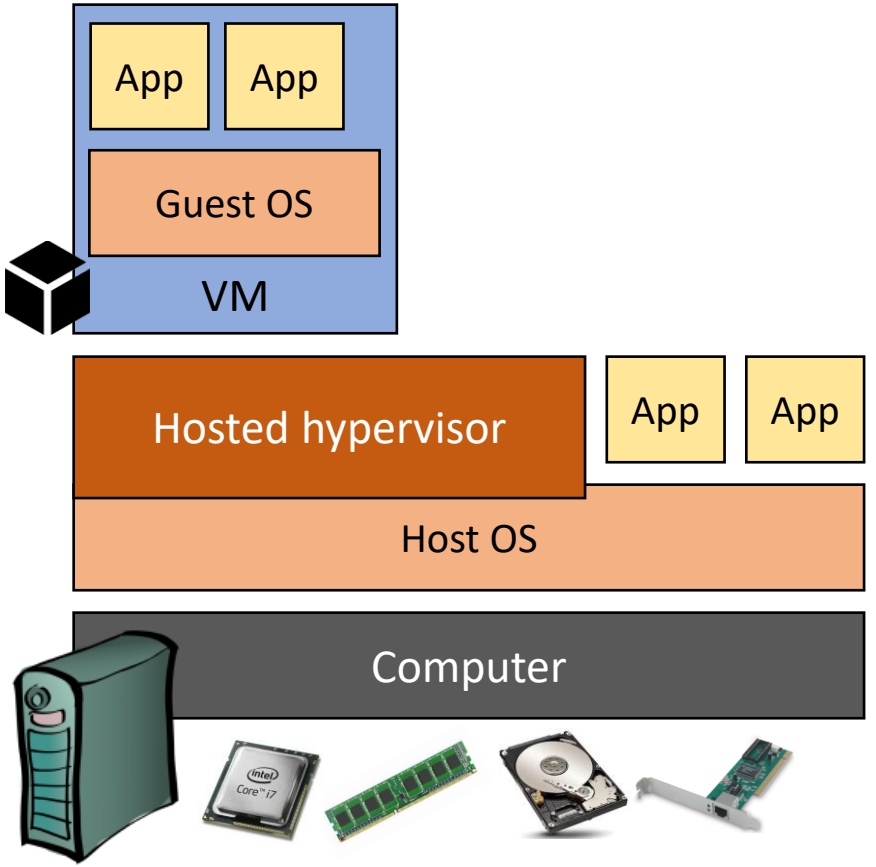
Type-1 Hypervisor



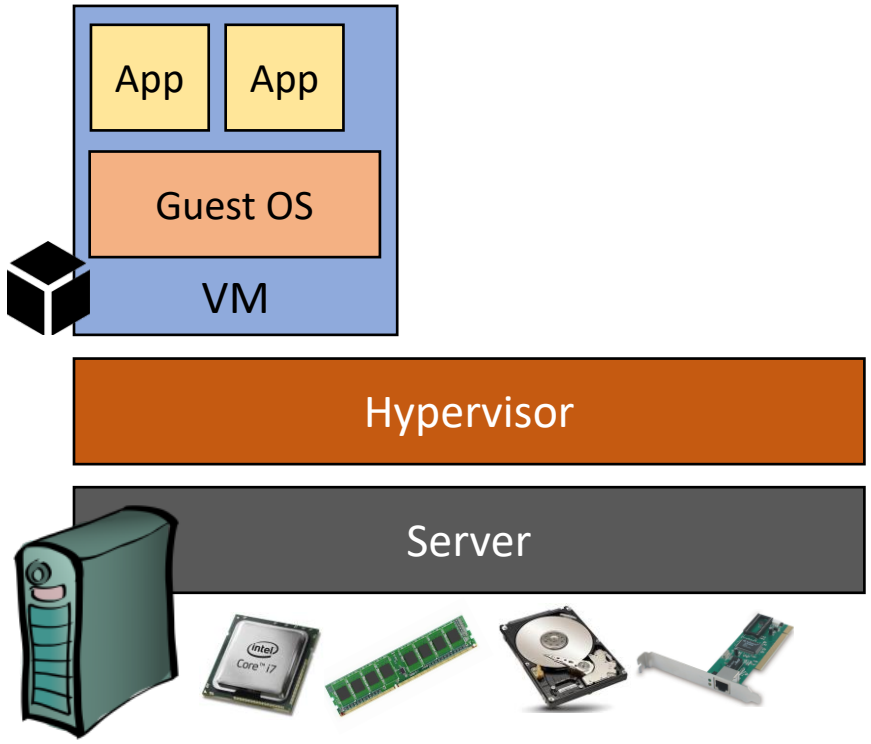
Type-1 Hypervisor



Type-2 Hypervisor



Type-1 Hypervisor



Type-2 Hypervisor