NoHype: Virtualized Cloud Infrastructure without the Virtualization

Eric Keller, Jakub Szefer, Jennifer Rexford, Ruby Lee

Princeton University

IBM Cloud Computing Student Workshop

(ISCA 2010 + Ongoing work)

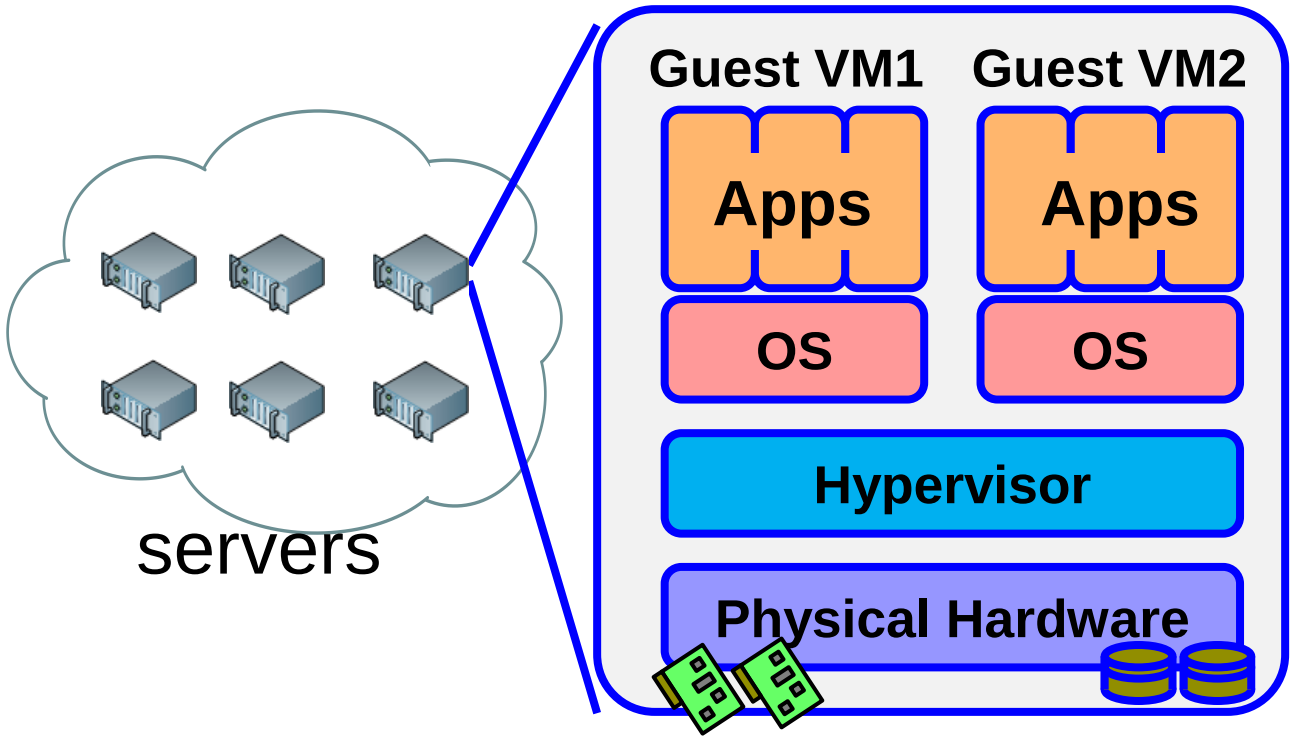
Virtualized Cloud Infrastructure

- Run virtual machines on a hosted infrastructure

- Benefits…
  - Economies of scale
  - Dynamically scale (pay for what you use)

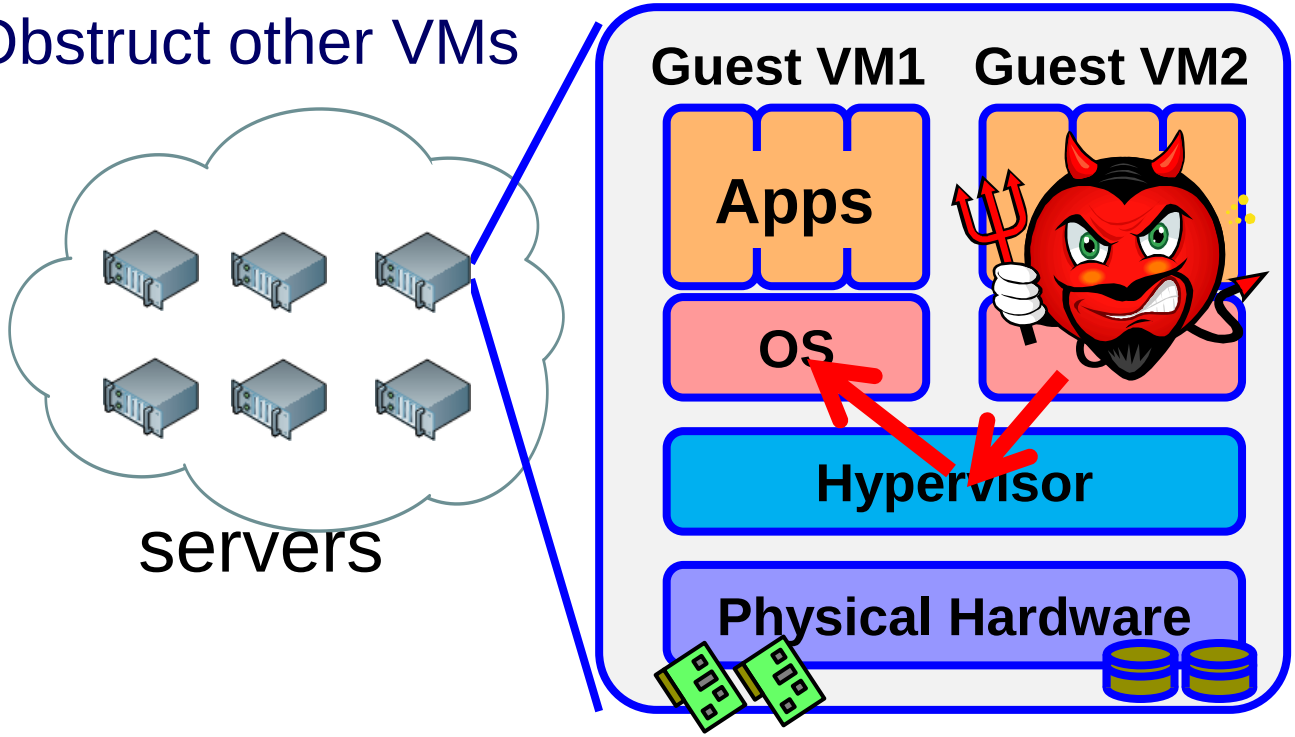
Without the Virtualization

- Virtualization used to share servers
  - Software layer running under each virtual machine



Without the Virtualization

- Virtualization used to share servers
  - Software layer running under each virtual machine
- Malicious software can run on the same server
  - Attack hypervisor
  - Access/Obstruct other VMs



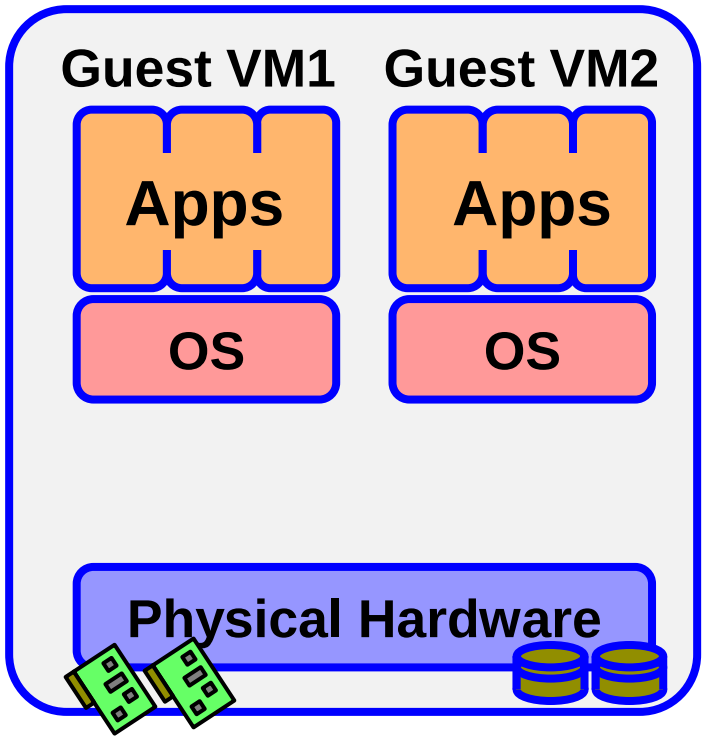
Are these vulnerabilities imagined?

- No headlines… doesn't mean it's not real
  - Not enticing enough to hackers yet? (small market size, lack of confidential data)
- Virtualization layer huge and growing
  - 100 Thousand lines of code in hypervisor
  - 1 Million lines in privileged virtual machine
- Derived from existing operating systems
  - Which have security holes

NoHype

- NoHype removes the hypervisor
  - There's nothing to attack
  - Complete systems solution
  - Still retains the needs of a virtualized cloud infrastructure

No hypervisor →



Virtualization in the Cloud

- Why does a cloud infrastructure use virtualization?
  - To support dynamically starting/stopping VMs
  - To allow servers to be shared (multi-tenancy)
- Do not need full power of modern hypervisors
  - Emulating diverse (potentially older) hardware
  - Maximizing server consolidation

Roles of the Hypervisor

- Isolating/Emulating resources
  - CPU: Scheduling virtual machines
  - Memory: Managing memory
  - I/O: Emulating I/O devices

Push to HW / Pre-allocation

- Networking

Remove

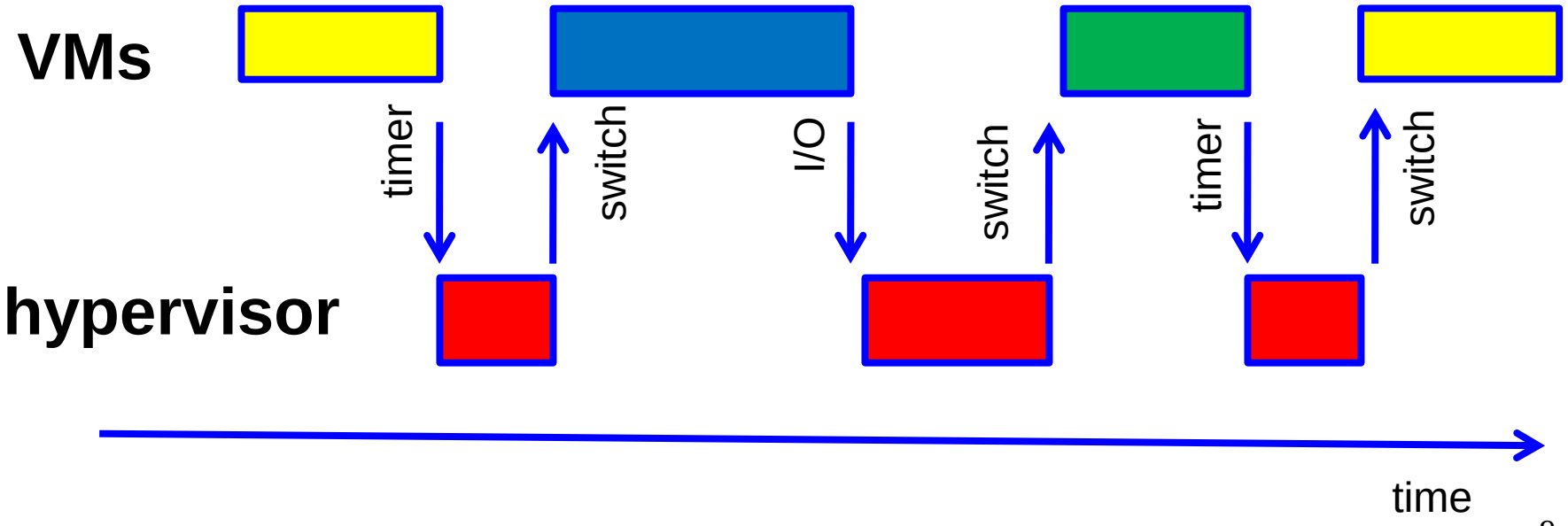
- Managing virtual machines

Push to side

NoHype has a double meaning… “no hype”

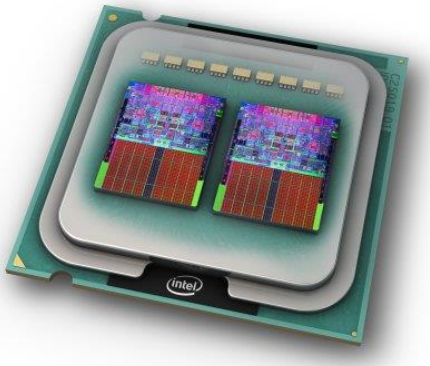
Scheduling Virtual Machines

- Scheduler called each time hypervisor runs (periodically, I/O events, etc.)
  - Chooses what to run next on given core
  - Balances load across cores



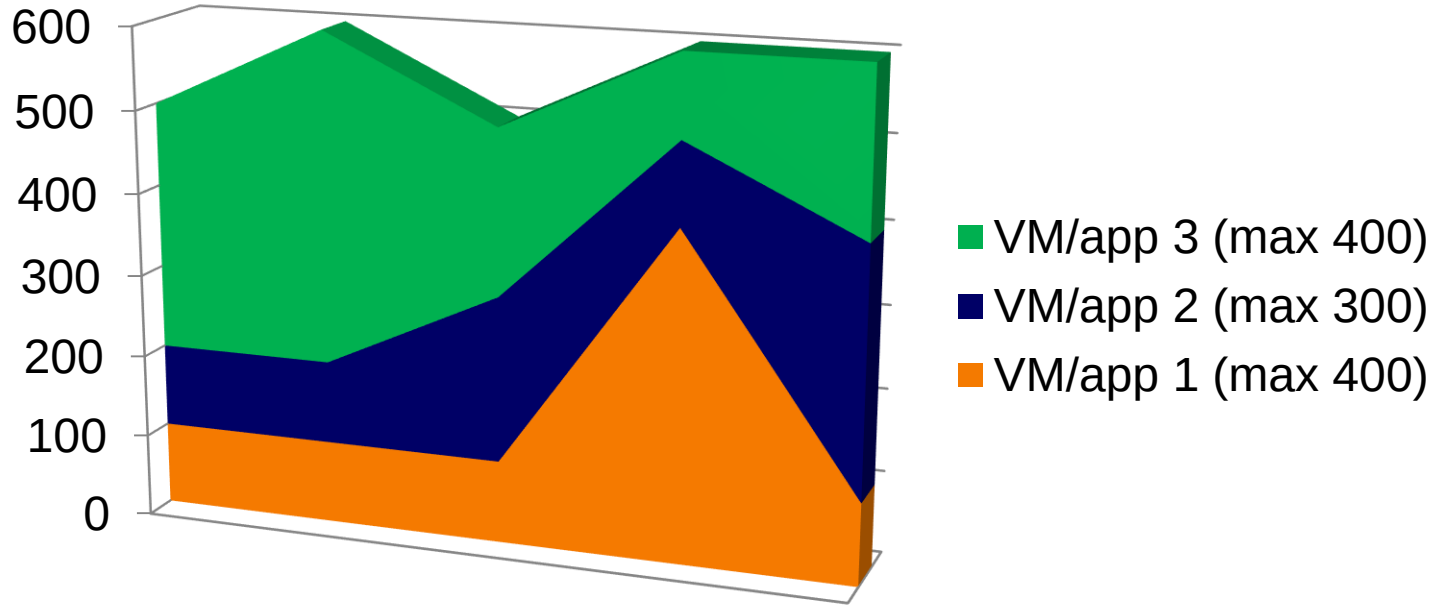
Dedicate a core to a single VM

- Ride the multi-core trend
  - 1 core on 128-core device is ~0.8% of the processor
- Cloud computing is pay-per-use
  - During high demand, spawn more VMs
  - During low demand, kill some VMs
  - Customer maximizing each VMs work, which minimizes opportunity for over-subscription



Managing Memory

- Goal: system-wide optimal usage
  - i.e., maximize server consolidation



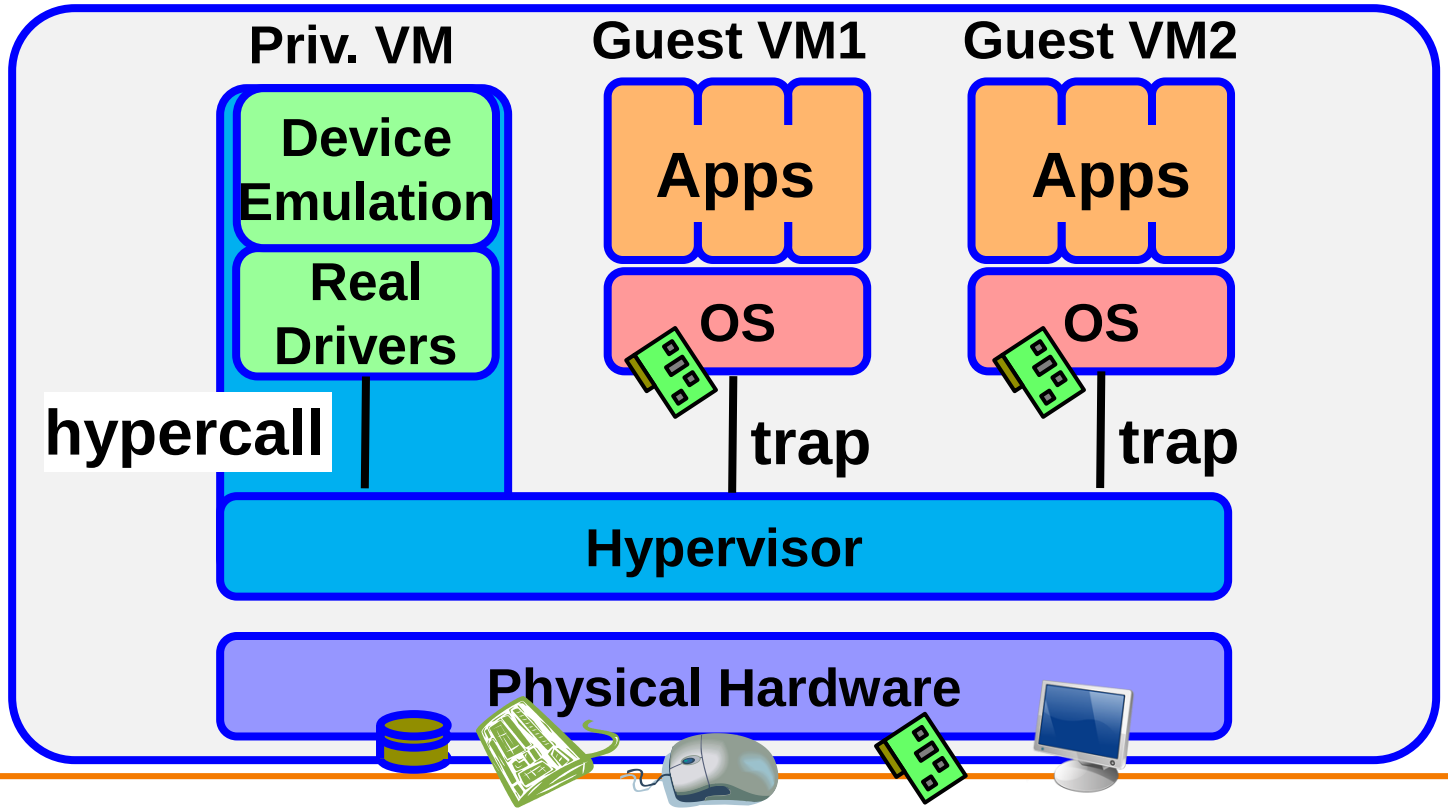
- Hypervisor controls allocation of physical memory

Pre-allocate Memory

- In cloud computing: charged per unit
  - e.g., VM with 2GB memory
- Pre-allocate a fixed amount of memory
  - Memory is fixed and guaranteed
  - Guest VM manages its own physical memory (deciding what pages to swap to disk)
- Processor support for enforcing:
  - allocation and bus utilization

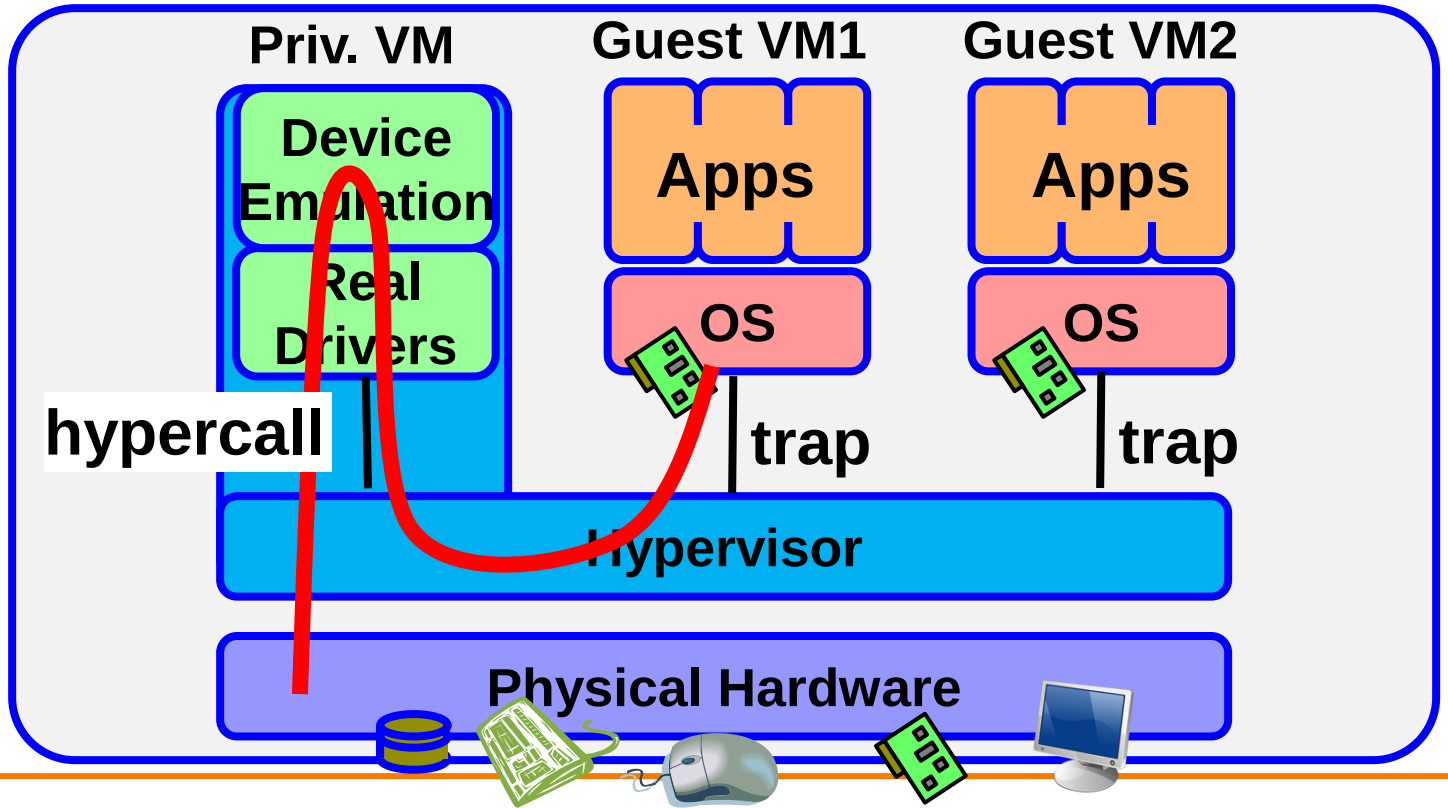
Emulate I/O Devices

- Guest sees virtual devices
  - Access to a device's memory range traps to hypervisor
  - Hypervisor handles interrupts
  - Privileged VM emulates devices and performs I/O



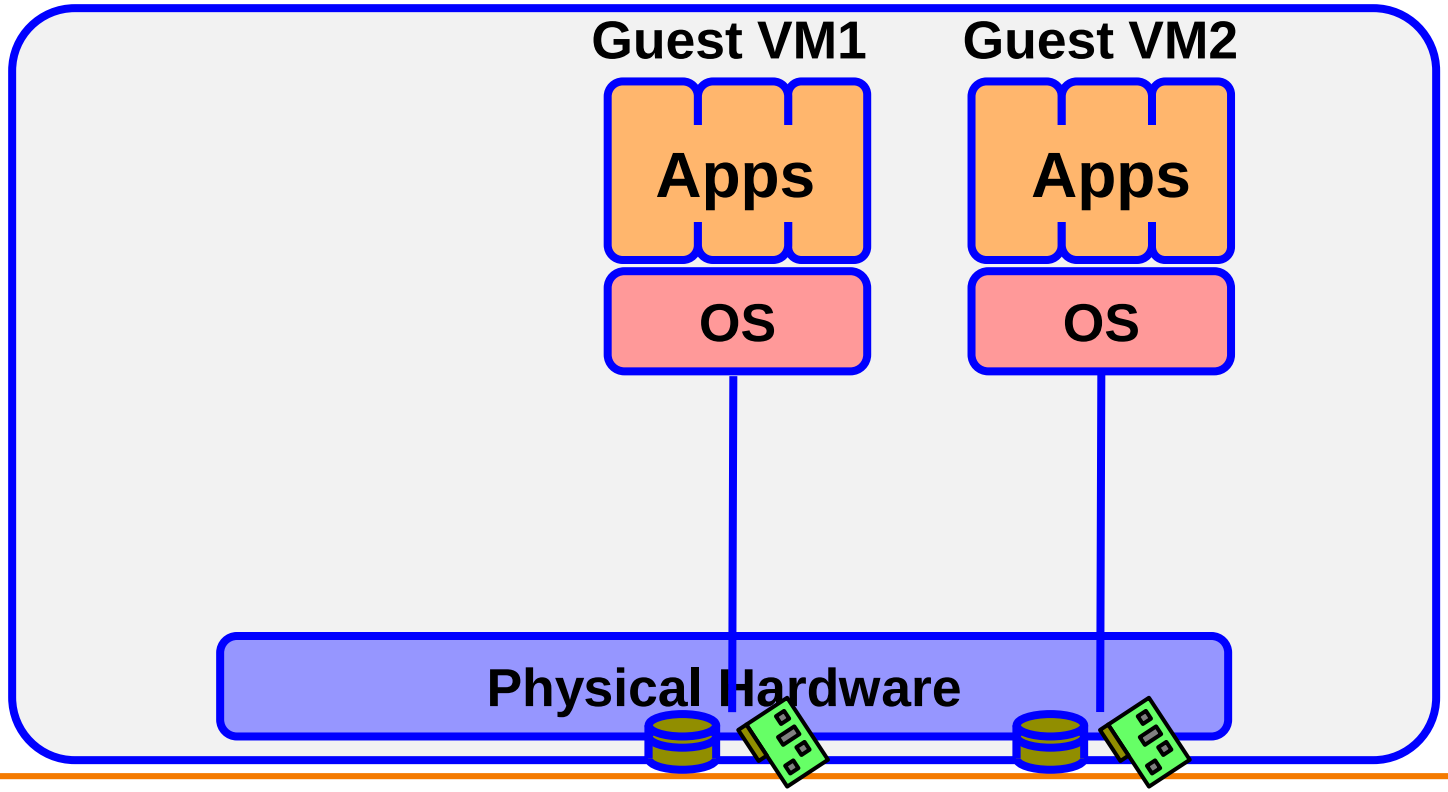
Emulate I/O Devices

- Guest sees virtual devices
  - Access to a device's memory range traps to hypervisor
  - Hypervisor handles interrupts
  - Privileged VM emulates devices and performs I/O



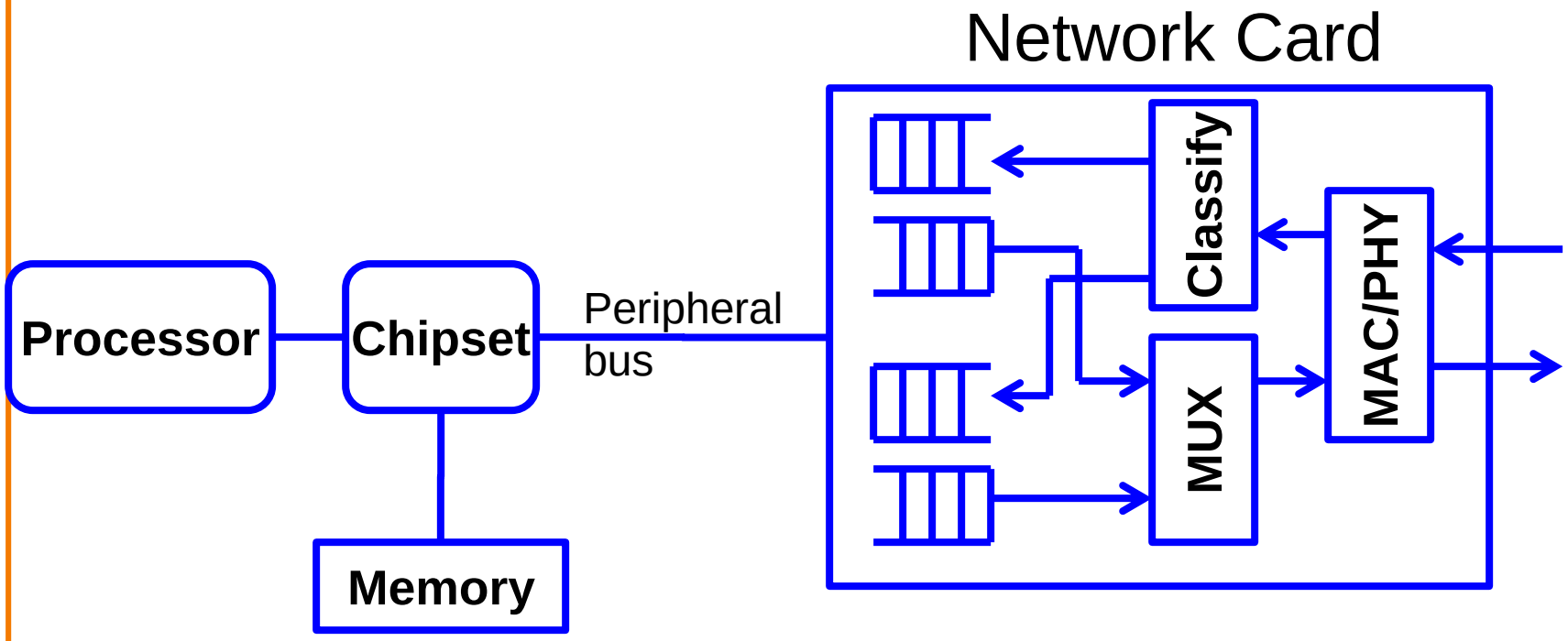
Dedicate Devices to a VM

- In cloud computing, only networking and storage
- Static memory partitioning for enforcing access
  - Processor (for to device), IOMMU (for from device)



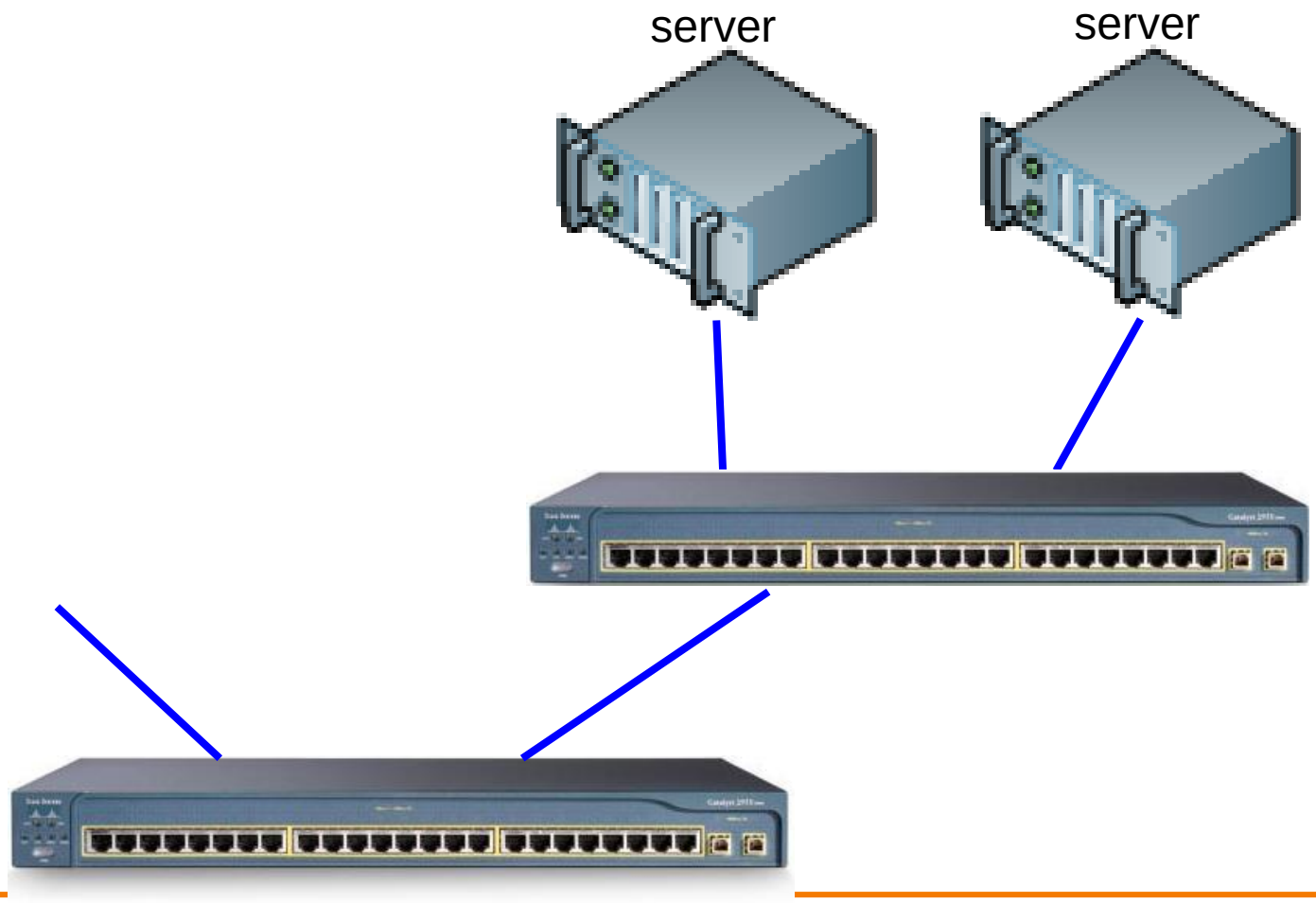
Virtualize the Devices

- Per-VM physical device doesn't scale
- Multiple queues on device
  - Multiple memory ranges mapping to different queues



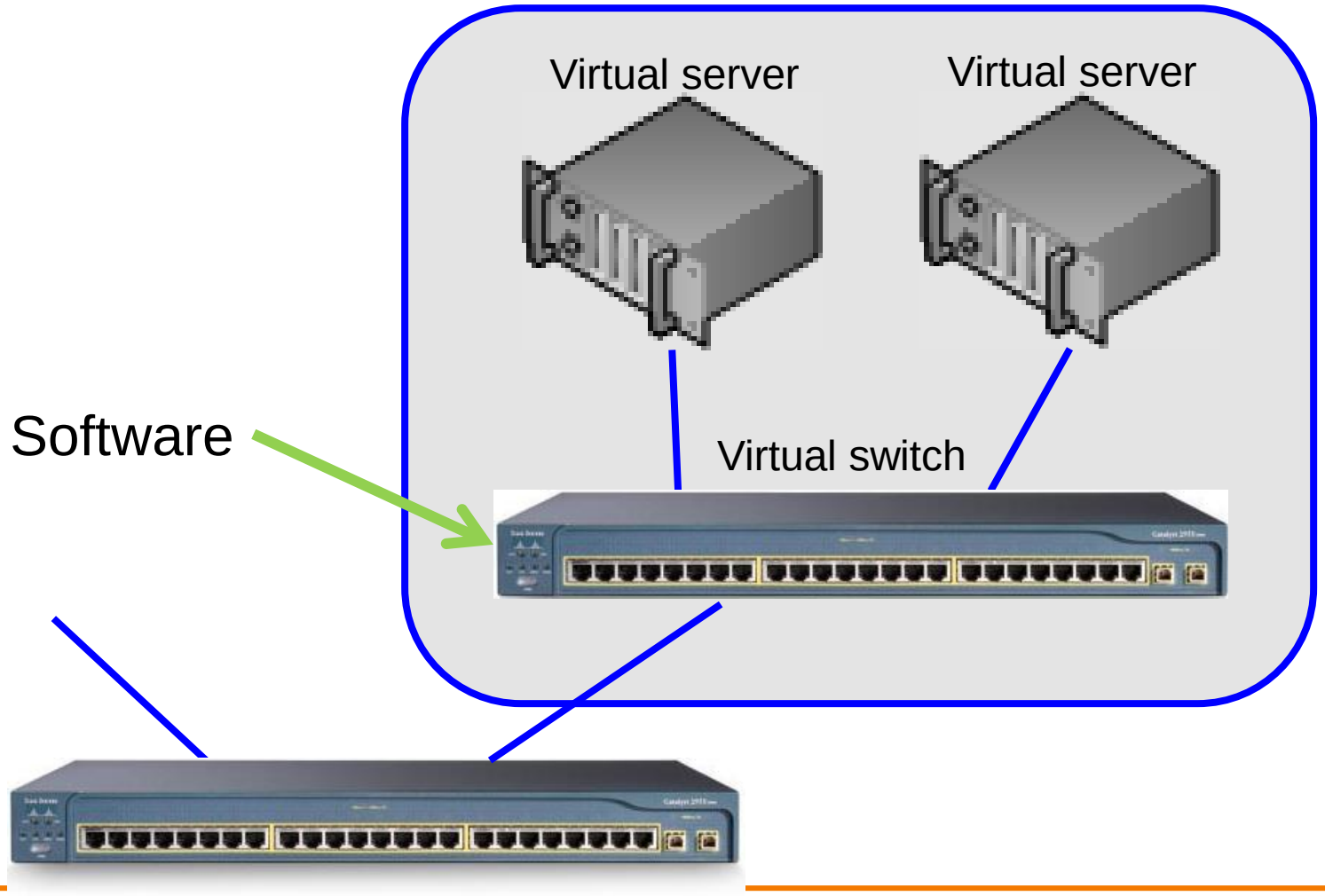
Networking

- Ethernet switches connect servers



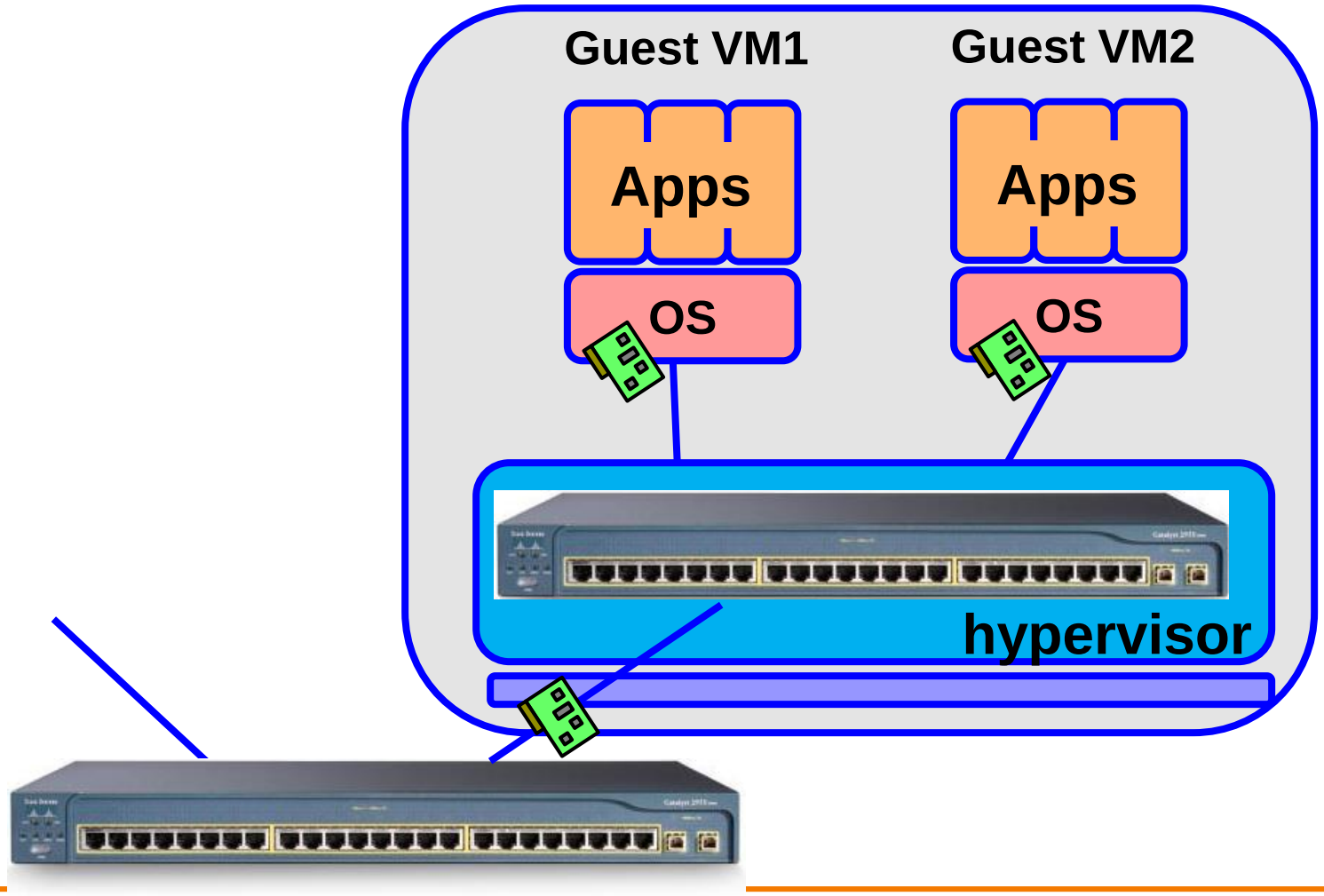
Networking (in virtualized server)

- Software Ethernet switches connect VMs



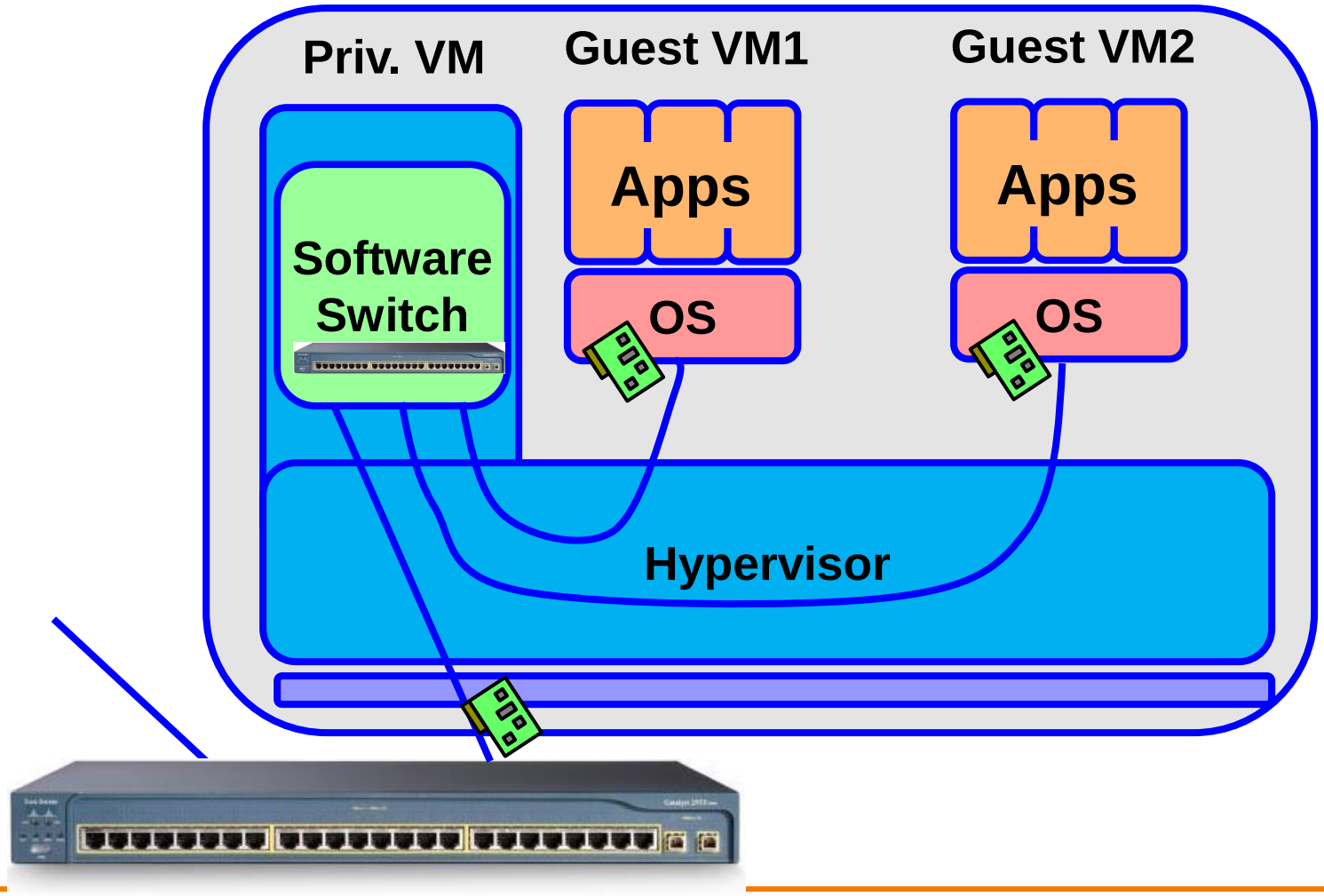
Networking (in virtualized server)

- Software Ethernet switches connect VMs



Networking (in virtualized server)

- Software Ethernet switches connect VMs



Do Networking in the Network

- Co-located VMs communicate through software
  - Performance penalty for not co-located VMs
  - Special case in cloud computing
  - Artifact of going through hypervisor anyway
- Instead: utilize hardware switches in the network
  - Modification to support hairpin turnaround

Removing the Hypervisor Summary

- Scheduling virtual machines
  - One VM per core
- Managing memory
  - Pre-allocate memory with processor support
- Emulating I/O devices
  - Direct access to virtualized devices
- Networking
  - Utilize hardware Ethernet switches
- Managing virtual machines
  - Decouple the management from operation

NoHype Double Meaning

- Means no hypervisor, also means “no hype”

- Multi-core processors
- Extended Page Tables
- SR-IOV and Directed I/O (VT-d)
- Virtual Ethernet Port Aggregator (VEPA)

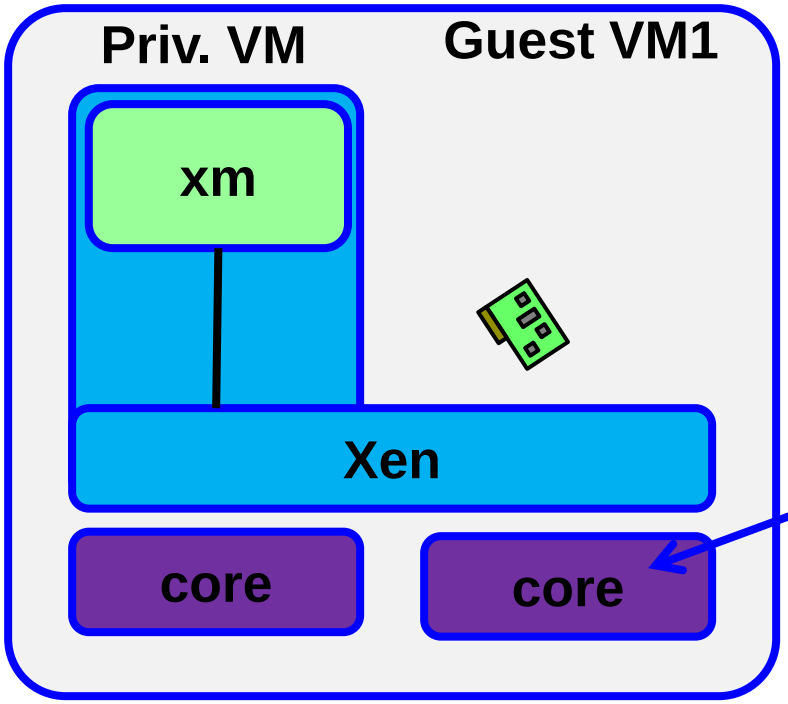
NoHype Double Meaning

- Means no hypervisor, also means "no hype"

- Multi-core processors
- Extended Page Tables
- SR-IOV and Directed I/O (VT-d)
- Virtual Ethernet Port Aggregator (VEPA)

Current Work: Implement it on today's HW

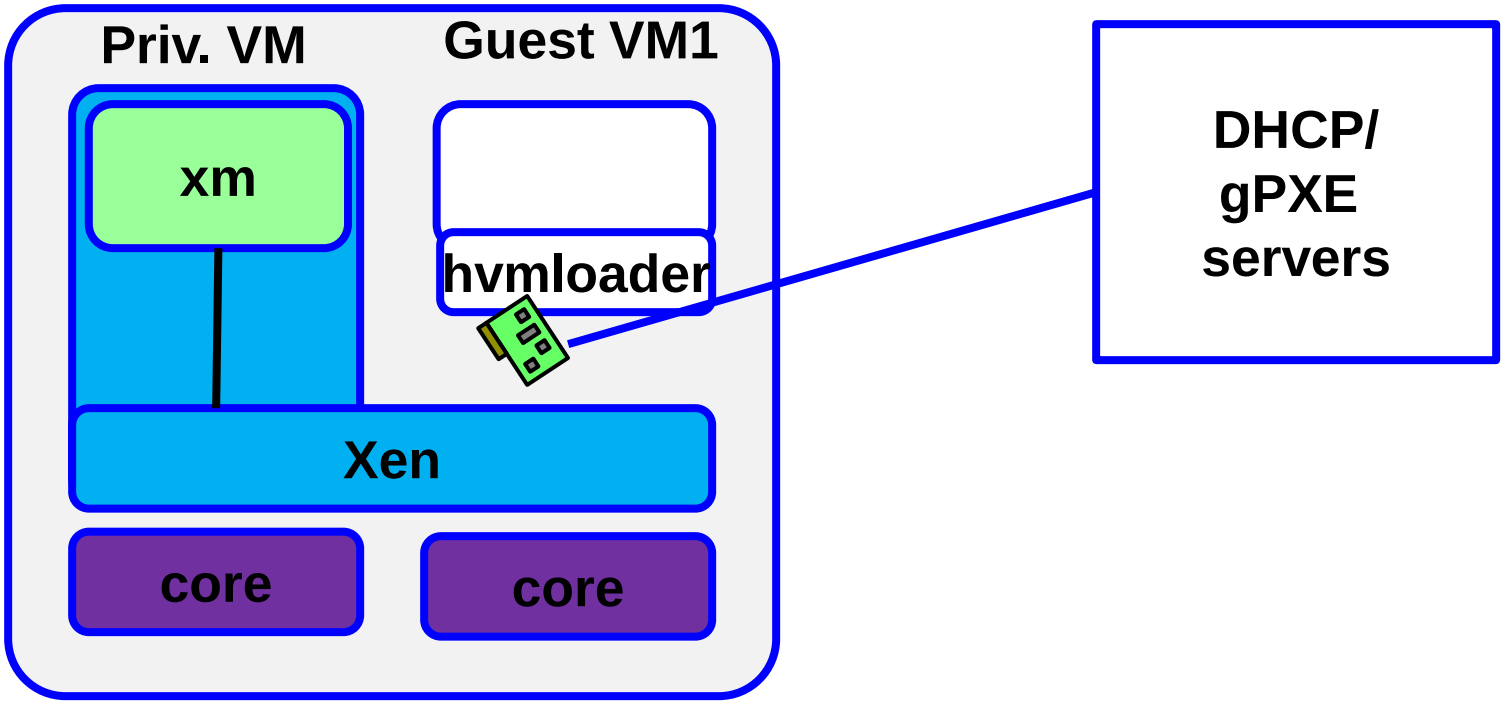
Xen as a Starting Point



Pre fill EPT mapping
to partition memory

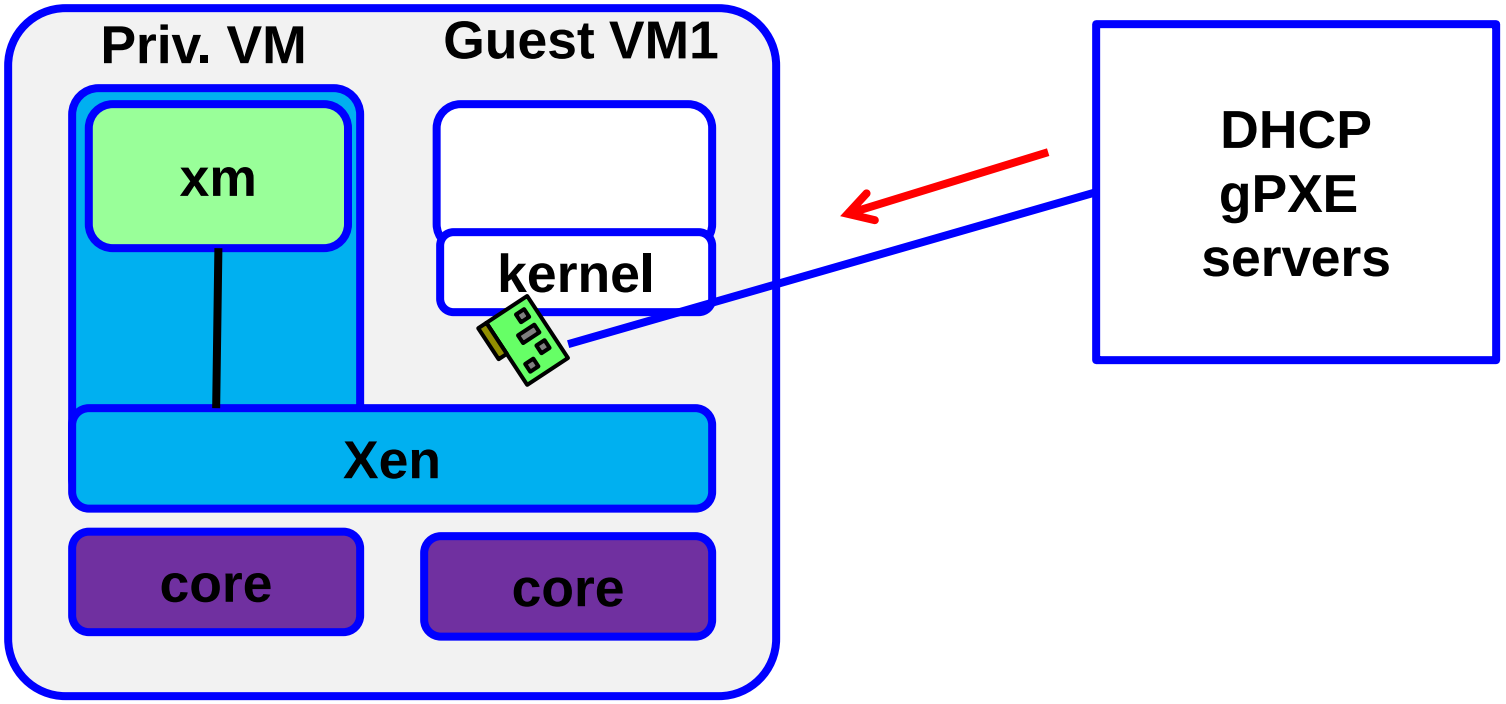
- Management tools
- Pre-allocate resources
  - i.e., configure virtualized hardware
- Launch VM

Network Boot



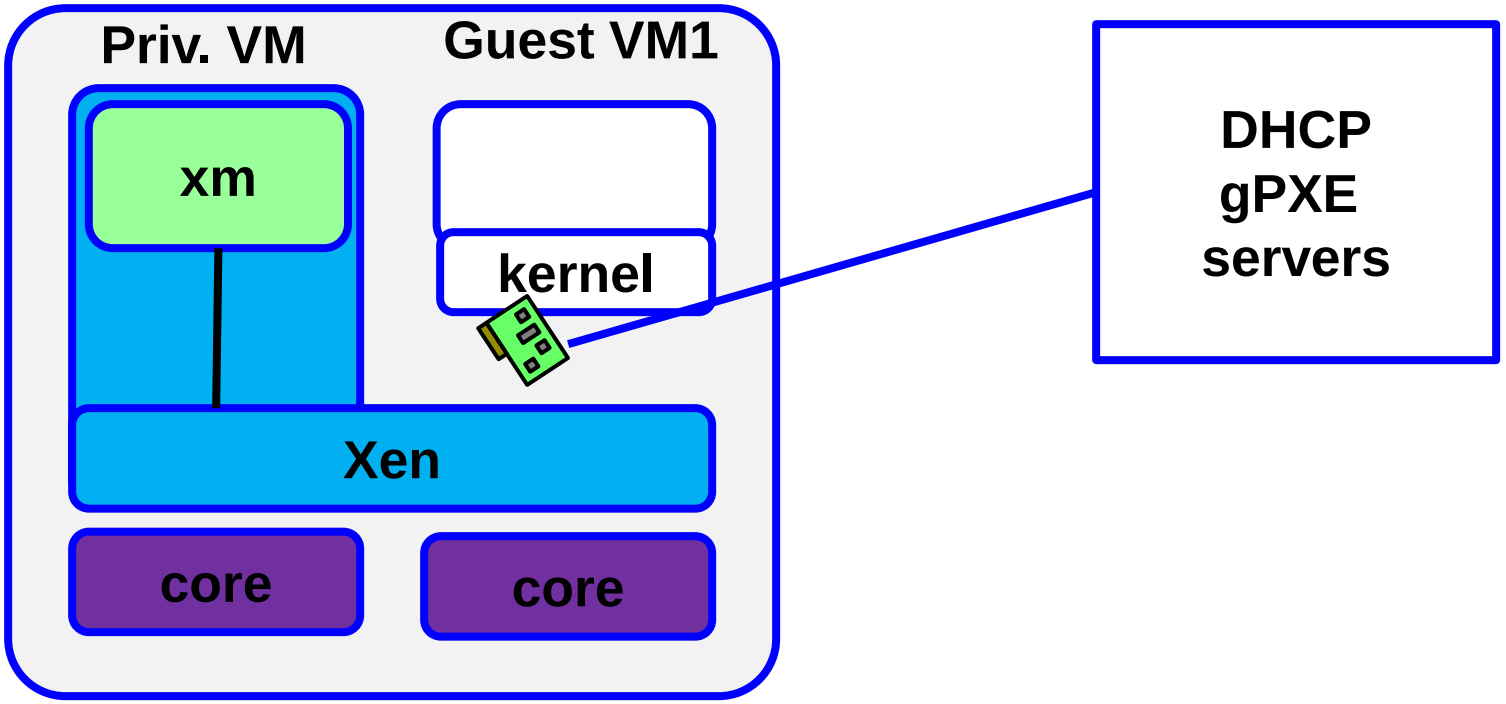
- gPXE in Hvmloader
  - Added support for igbvf (Intel 82576)
- Allows us to remove disk
  - Which are not virtualized yet

Allow Legacy Bootup Functionality



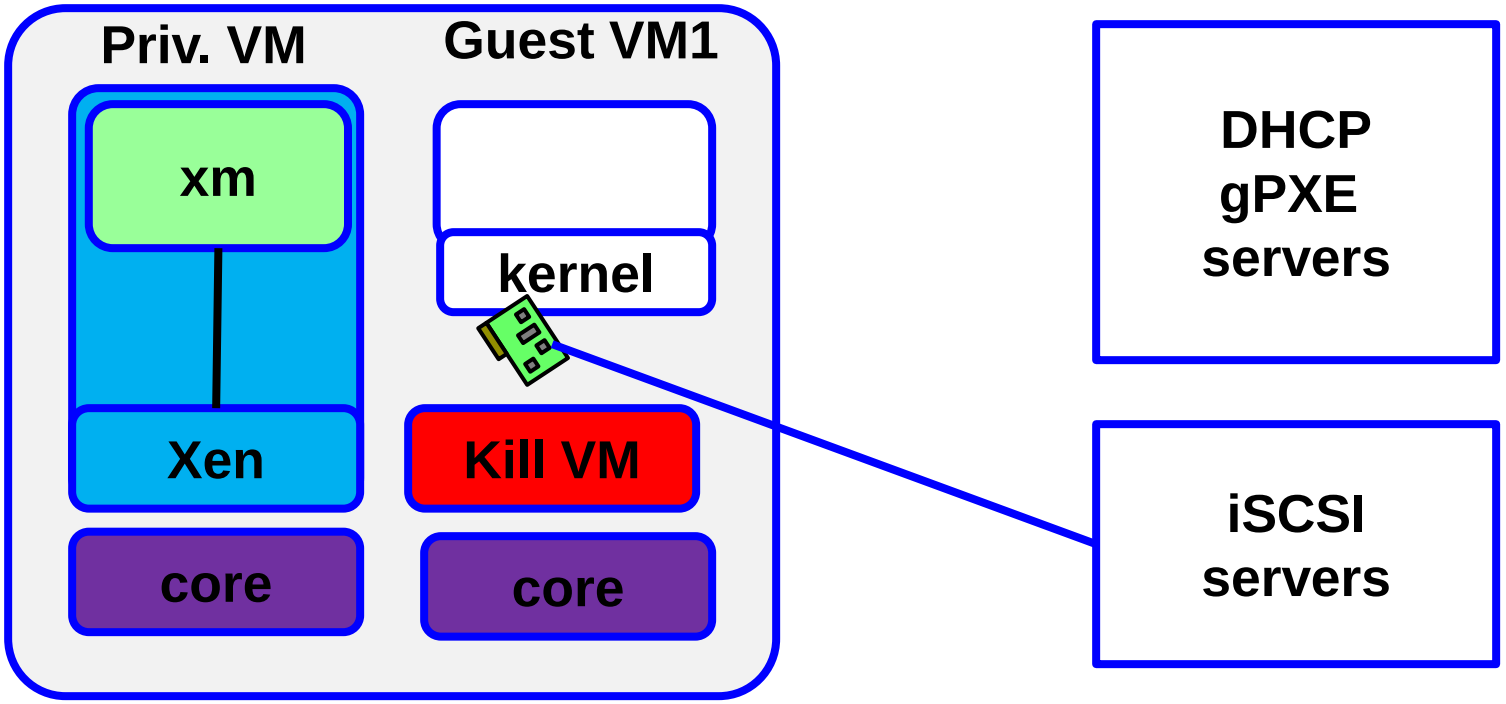
- Known good kernel + initrd (our code)
  - PCI reads return “no device” except for NIC
  - HPET reads to determine clock freq.

Use Device Level Virtualization



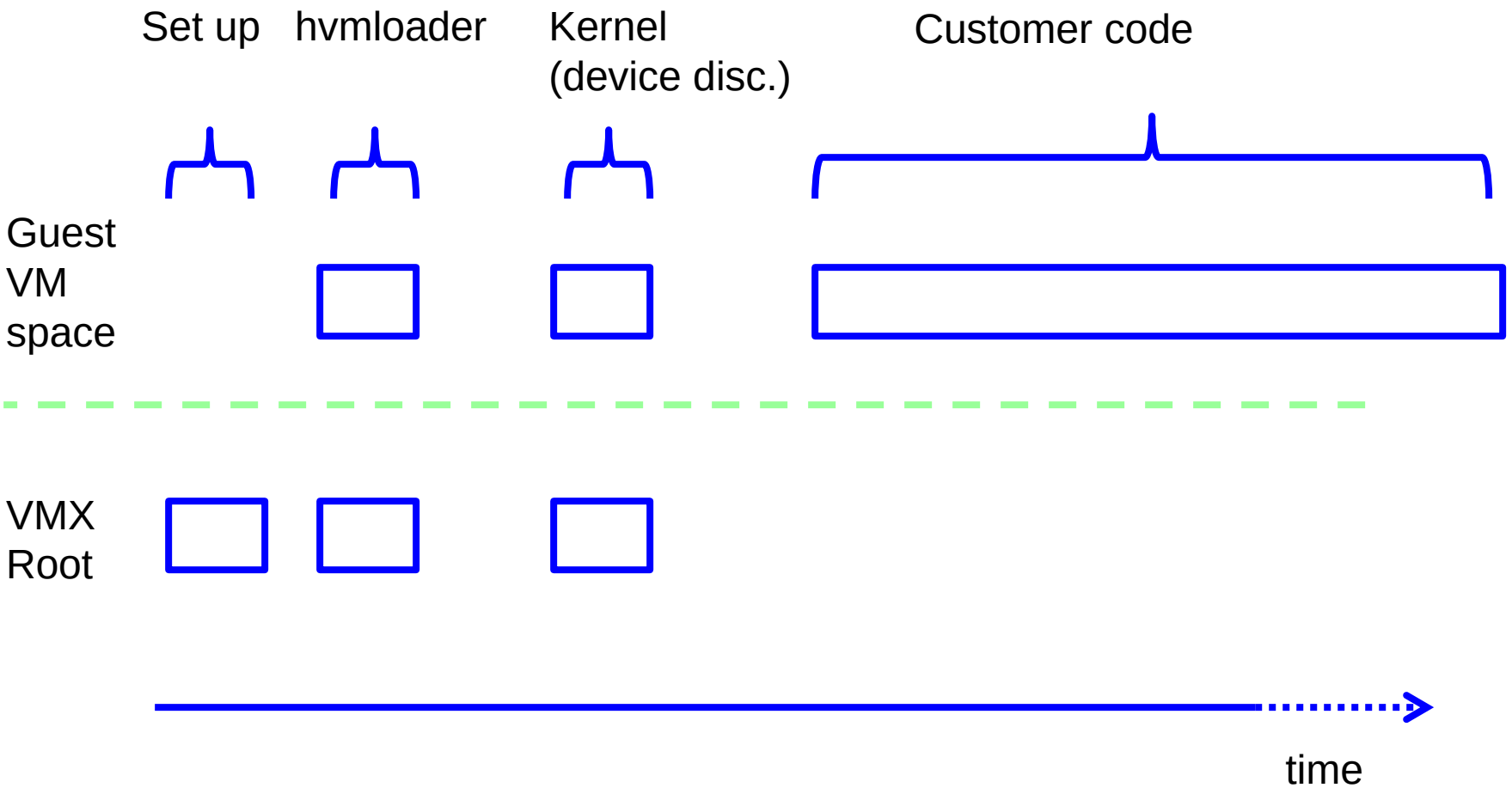
- Pass through Virtualized NIC
- Pass through Local APIC (for timer)

Block All Hypervisor Access



- Mount iSCSI drive for user disk
- Before jumping to user code, switch off hypervisor
  - Any VM Exit causes a Kill VM
  - User can load kernel modules, any applications

Timeline


Set up
hvmloader
Kernel
(device disc.)
Customer code
Guest
VM
space
VMX
Root
time

Next Steps

- Assess needs for future processors

- Assess OS modifications
  - to eliminate need for golden image
    (e.g., push configuration instead of discovery)

Conclusions

- Trend towards hosted and shared infrastructures
- Significant security issue threatens adoption
- NoHype solves this by removing the hypervisor
- Performance improvement is a side benefit

Questions?

Contact info:

ekeller@princeton.edu

http://www.princeton.edu/~ekeller

szefer@princeton.edu

http://www.princeton.edu/~szefer