Eliminating the Hypervisor Attack Surface for a More Secure Cloud

Jakub Szefer
Ruby B. Lee
Eric Keller
Jennifer Rexford

CCS 2011

PRINCETON
UNIVERSITY

Public Cloud Infrastructures

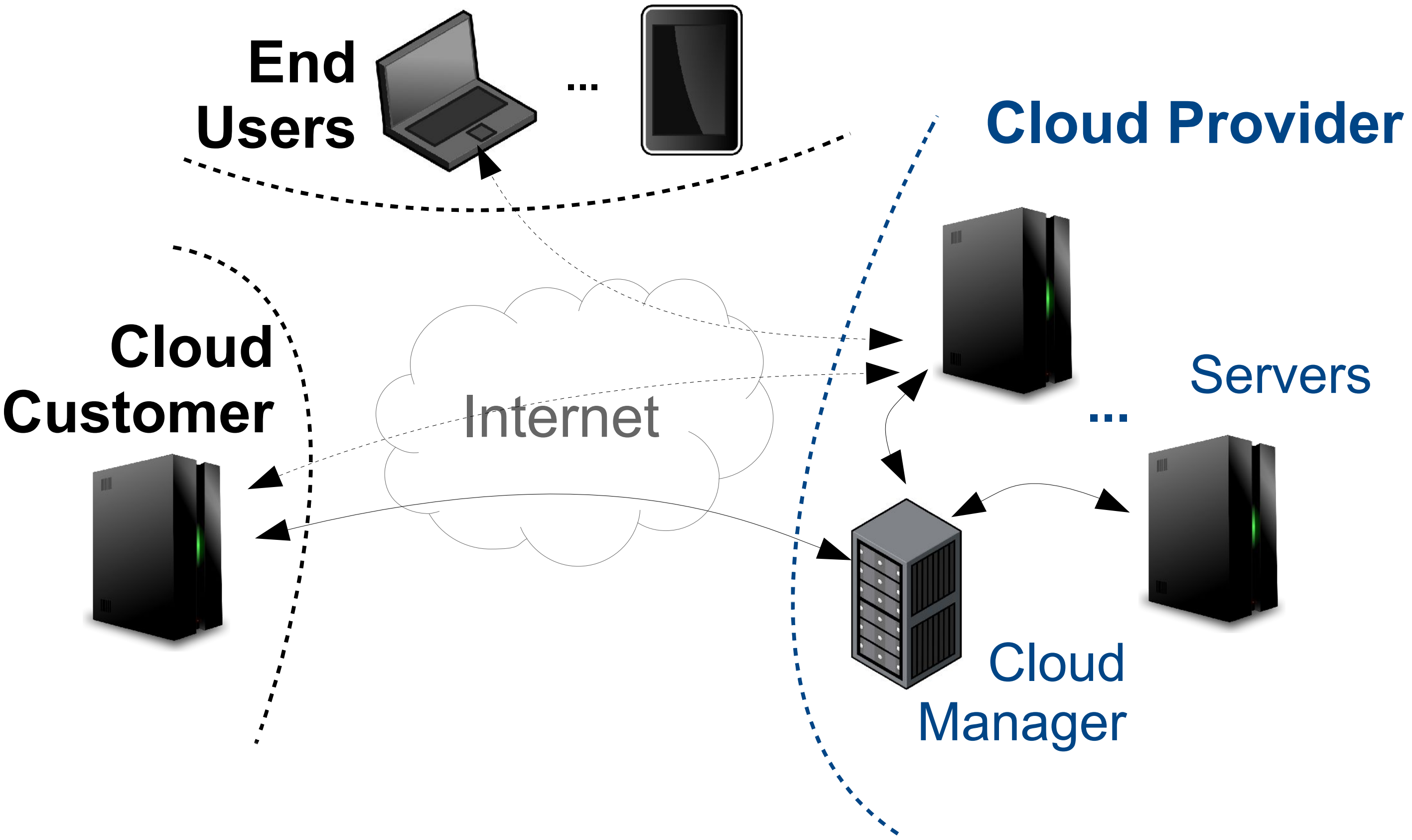
- Providers maintain and lease computing resources:

ElasticHosts
Flexible servers in the cloud
amazon web services™
flexiscale™
the rackspace cloud
GOGRID beta
A ServePath Company
Windows Azure™

- Benefits:
  - Public (anybody can use)
  - Economies of scale (lower cost)
  - Flexibility (pay per use)

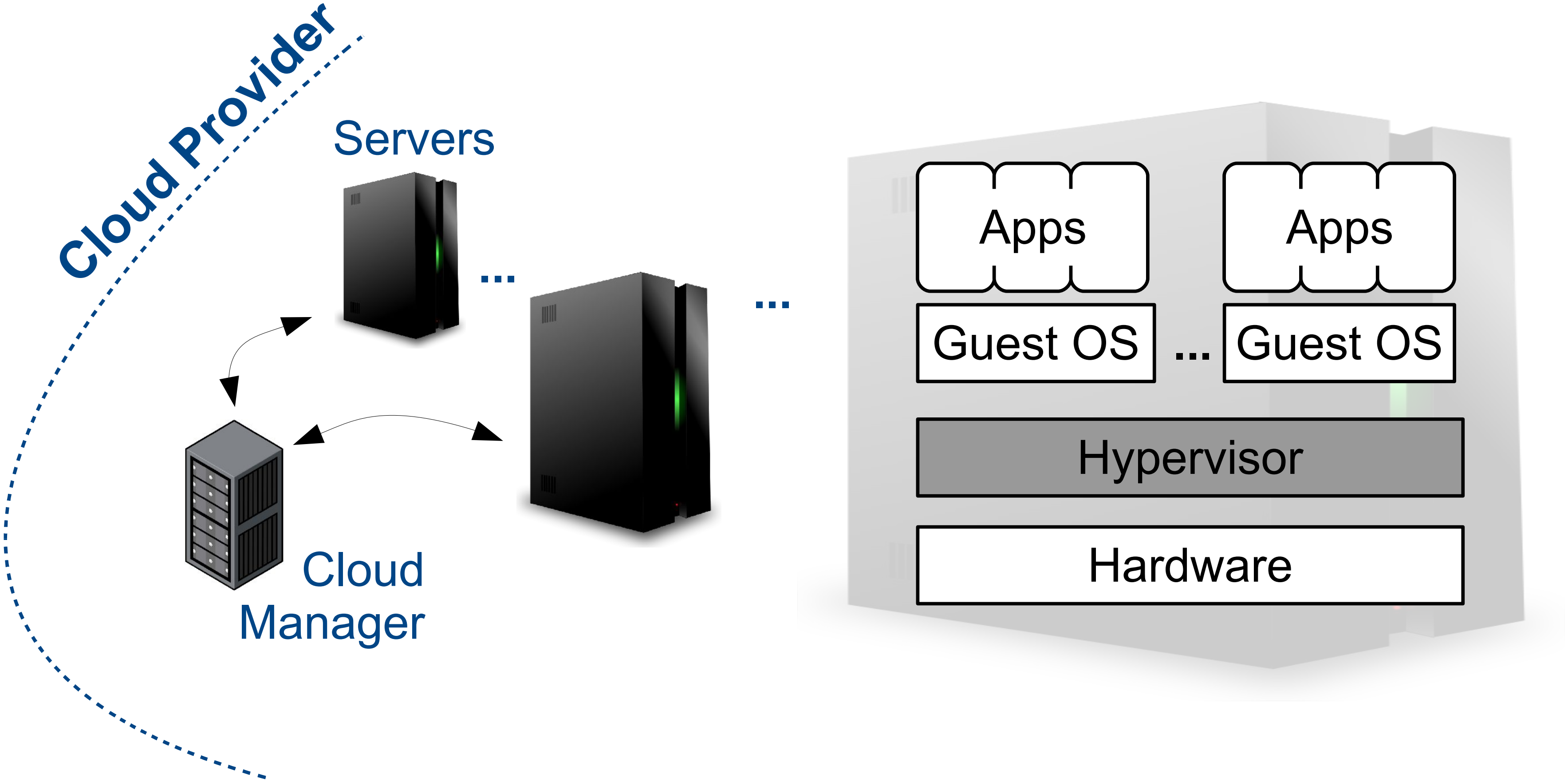
Public Cloud Infrastructures

- Infrastructure-as-a-Service (IaaS) cloud:



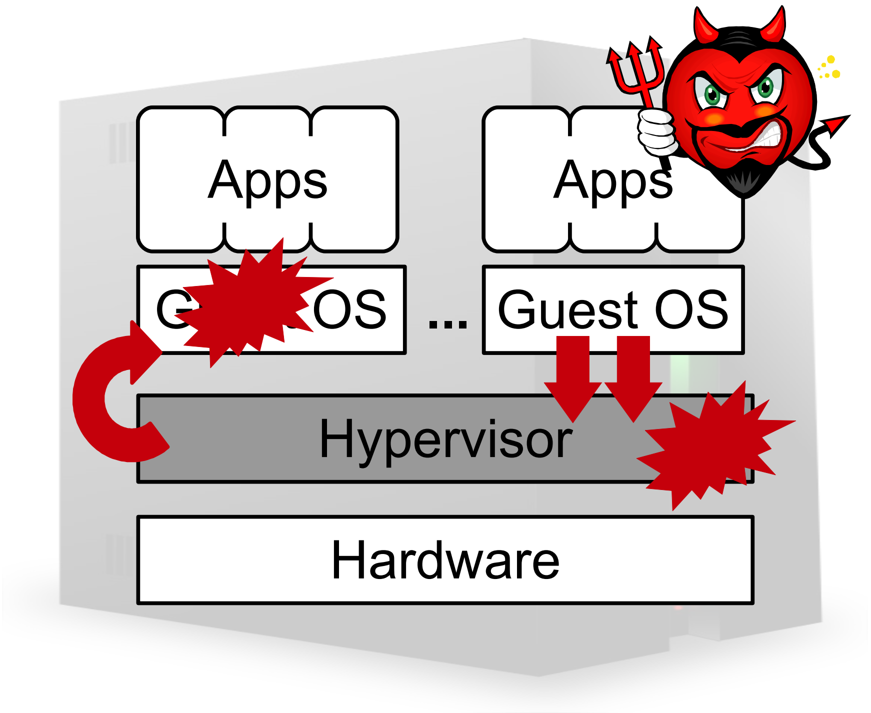
Virtualization and IaaS

- Virtualization allows many VMs to share single server:



Threat Model

- Protect against attacks on the hypervisor by the guest VMs



Assumptions:

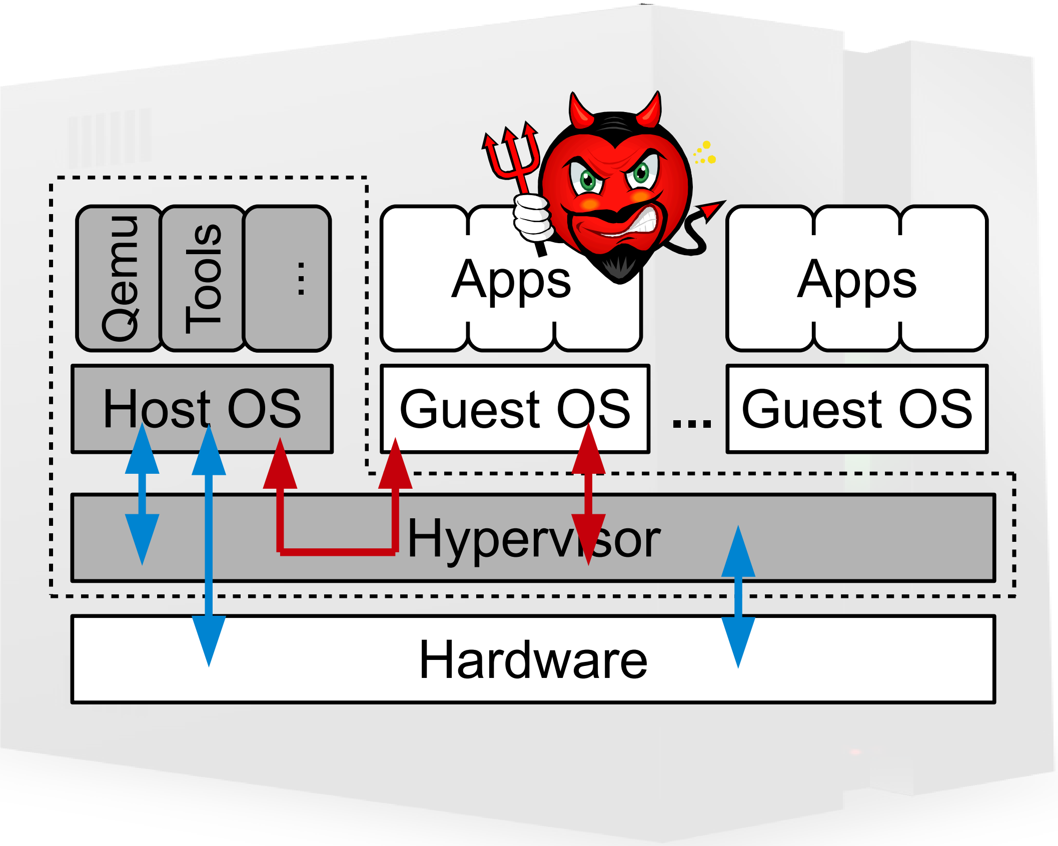
- Non-malicious infrastructure provider, secure facilities
- Guest VM and applications security is out-of-scope

Attack Surface: VM Exits

- Each VM to hypervisor interaction is a potential attack vector

- VMs interact with hypervisor through the VM Exits

- 56 reasons for VM Exits on modern Intel x86



- Interaction is very frequent, average 600 times per second

PRINCETON UNIVERSITY

Security Threat Scale

- Complex and large software base leads to many bugs

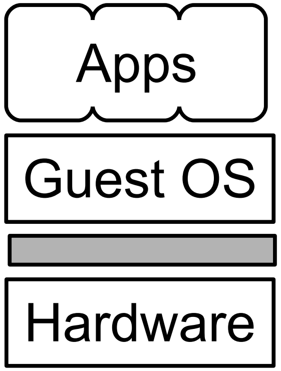
Software	SLOC
seL4	8,000
Hyper-V	100,000
Xen 4.0	194,000
VMWare ESX	200,000

- Reports of bugs: Xen 98 and VMware ESX 78 (NIST's National Vulnerability Database)

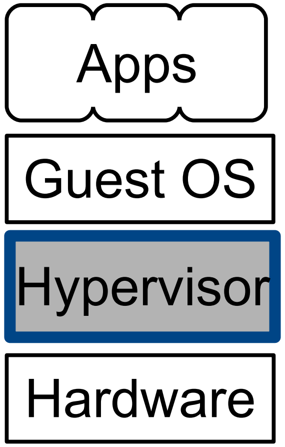
- E.g. Xen vulnerability CVE-2011-1780 (May 2011): *"Malicious guest user space process can trick the emulator into reading a different instruction than the one that caused the* ***VM exit*** *[to] potentially use this flaw to* ***crash the host****."*

Countering The Threat

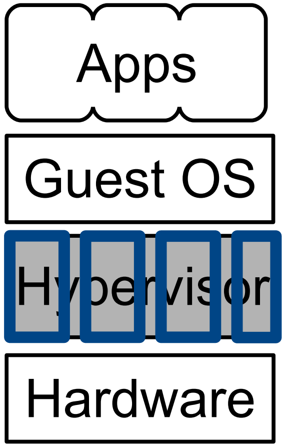
- Could minimize the hypervisor, e.g. SecVisor.



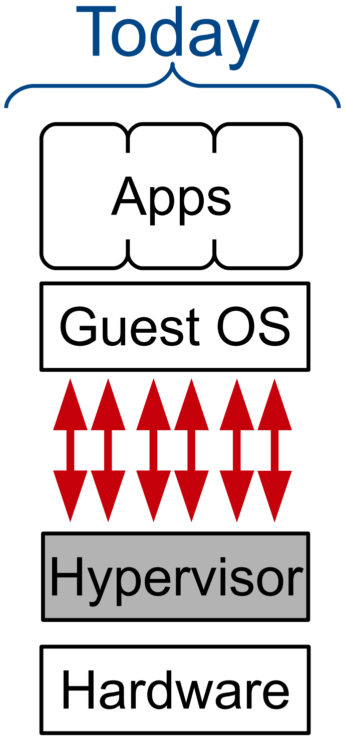
- Could harden the hypervisor, e.g. HyperSafe.



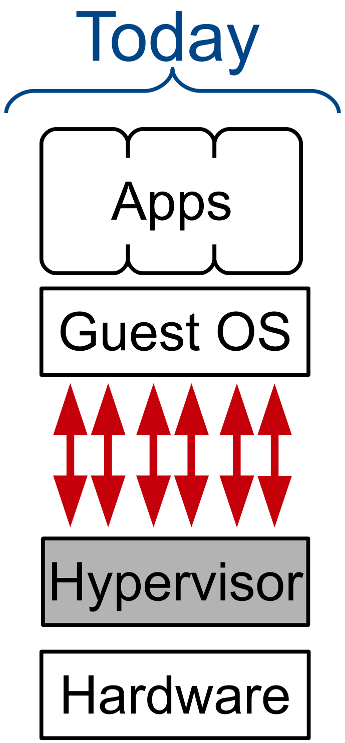
- Could partition functionality of the hypervisor, e.g. Xoar.



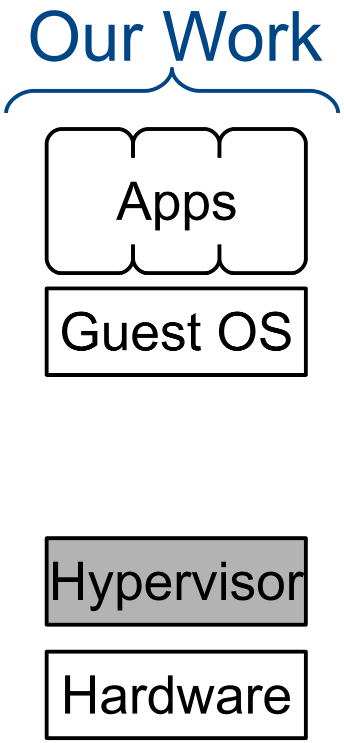
Eliminating the Hypervisor Attack Surface



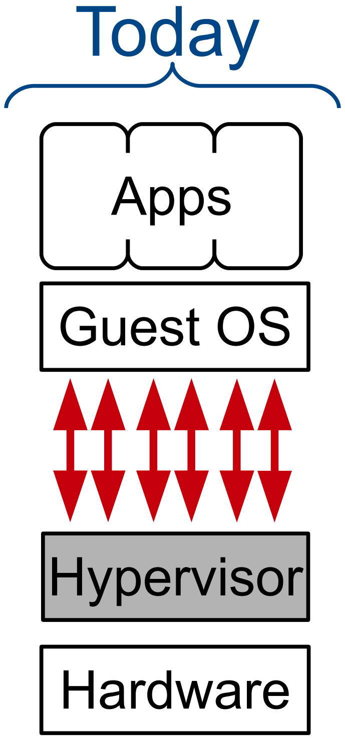
Eliminating the Hypervisor Attack Surface



We remove the need for VM exit to hypervisor for services.

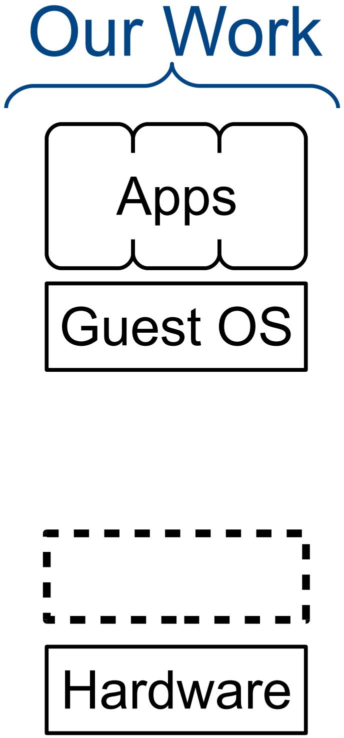


Eliminating the Hypervisor Attack Surface



We remove the need for VM exit to hypervisor for services.

And now can remove active hypervisor.

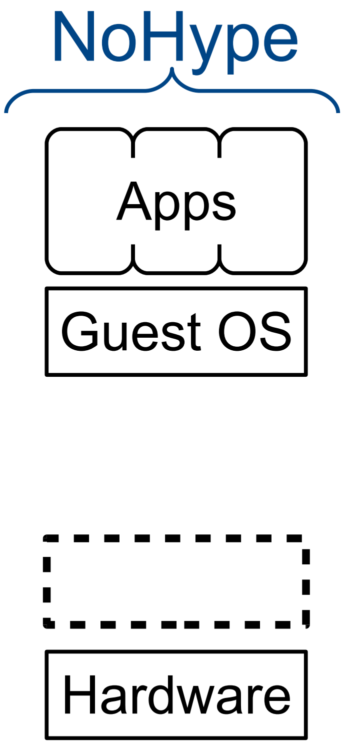


Introducing NoHype

NoHype supports:

- On-demand creation and termination of VMs
- Multi-tenancy
- Devices commonly used in VMs deployed in the cloud

NoHype can be realized today.



Virtualization without a Hypervisor ... a Contradiction?

The cloud environment offers unique opportunities:

Virtualization without a Hypervisor ... a Contradiction?

The cloud environment offers unique opportunities:

- Limited number of devices which need to be supported
  - Network, Disk

Removes need for active emulation of other devices

Virtualization without a Hypervisor ... a Contradiction?

The cloud environment offers unique opportunities:

- Limited number of devices which need to be supported
  - Network, Disk

Removes need for active emulation of other devices

- Pay-per-use where user selects needed resources upfront
  - CPU, Memory, Disk, Network

Can pre-assign resources based on the request

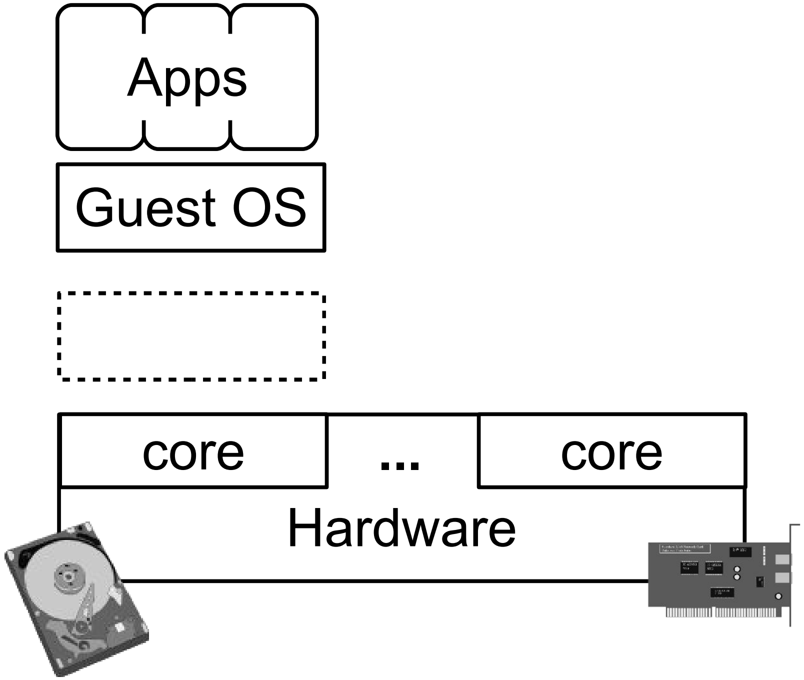
NoHype on Today's Hardware

- Pre-allocating memory and cores
- Using hardware virtualized I/O devices
- Short-circuiting the system discovery process
- Avoiding indirection

Pre-allocating Memory and Cores

Remove need for hypervisor involvement by:

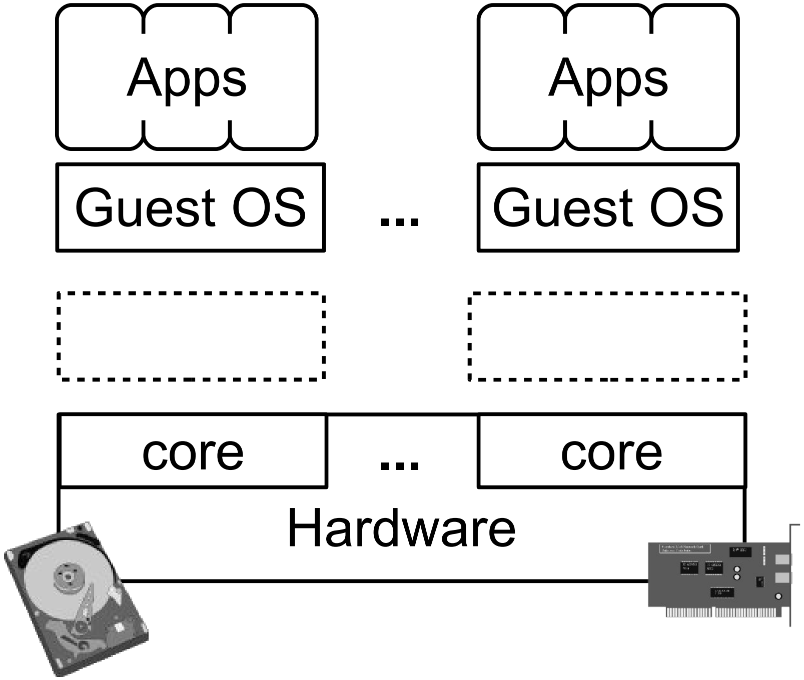
- Assigning cores based on customer's request
- Pre-allocating memory to match customer's request



Pre-allocating Memory and Cores

Remove need for hypervisor involvement by:

- Assigning cores based on customer's request
- Pre-allocating memory to match customer's request

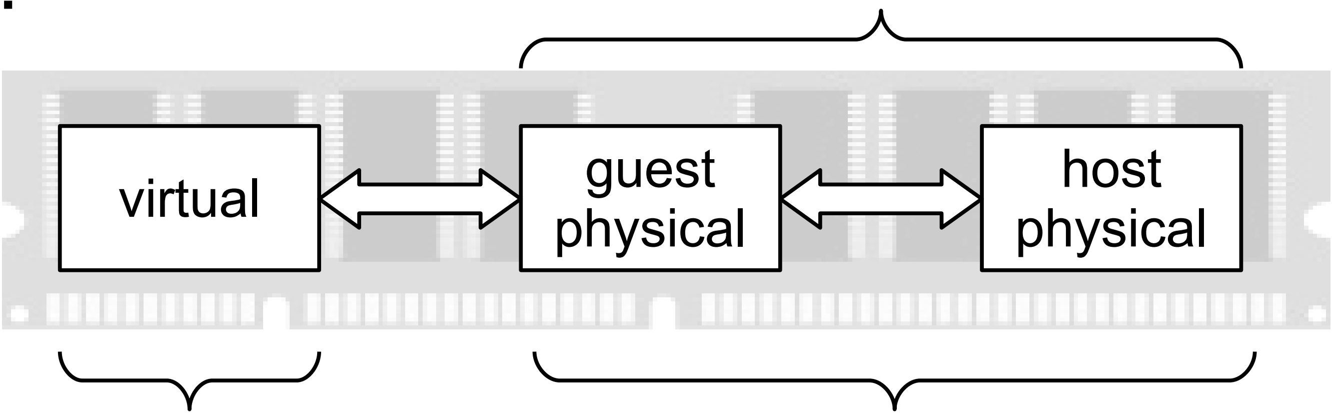


Pre-allocating Memory and Cores

Remove need for hypervisor involvement by:

- Enforcing using existing hardware mechanisms

E.g. Memory:



Intel's Extended Page Tables (EPT) hardware

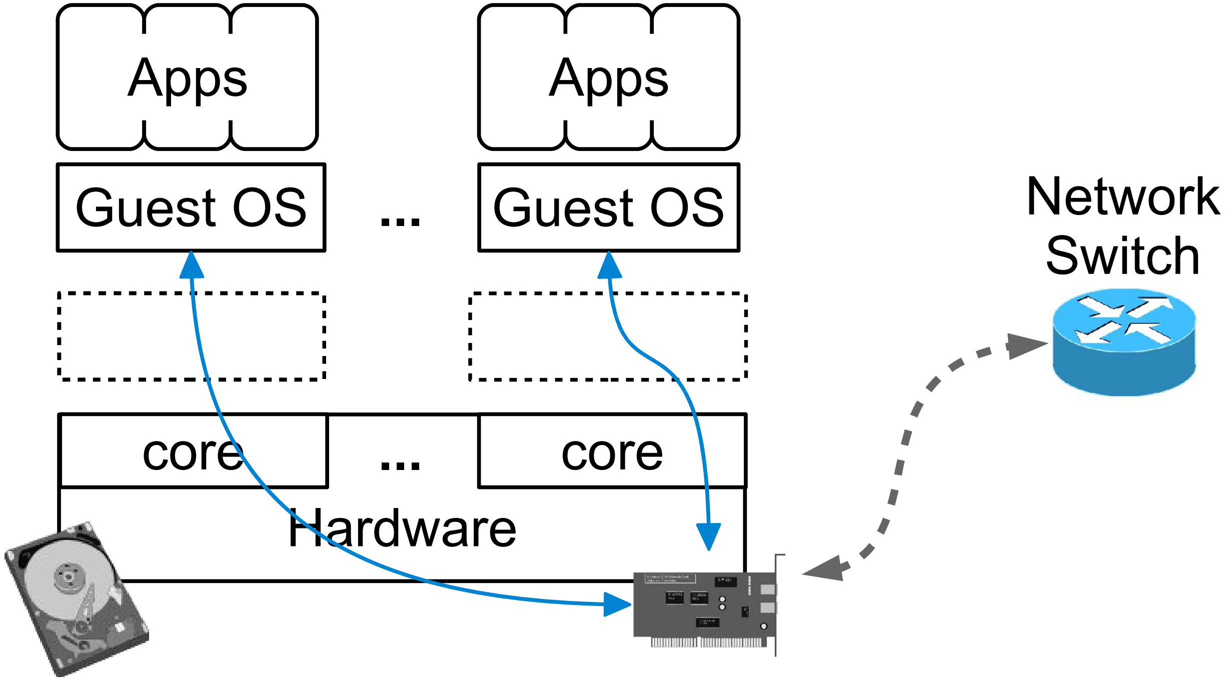
OS manages virtual memory without need for hypervisor

Lock-in pre-assigned memory before VM starts

Using Hardware Virtualized I/O Devices

Use of hardware virtualized I/O devices so:

- Each guest OS can receive dedicated devices
- No need to emulate the devices

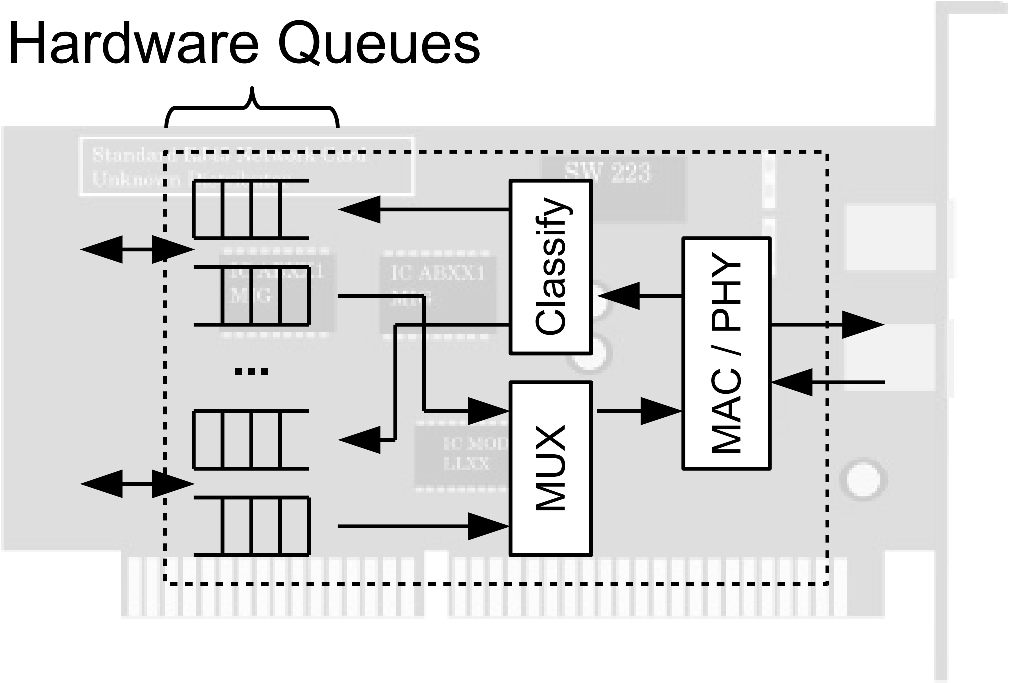


Using Hardware Virtualized I/O Devices

Use of hardware virtualized I/O devices so:

- There is no need for separate physical device for each OS

E.g. Network Card:

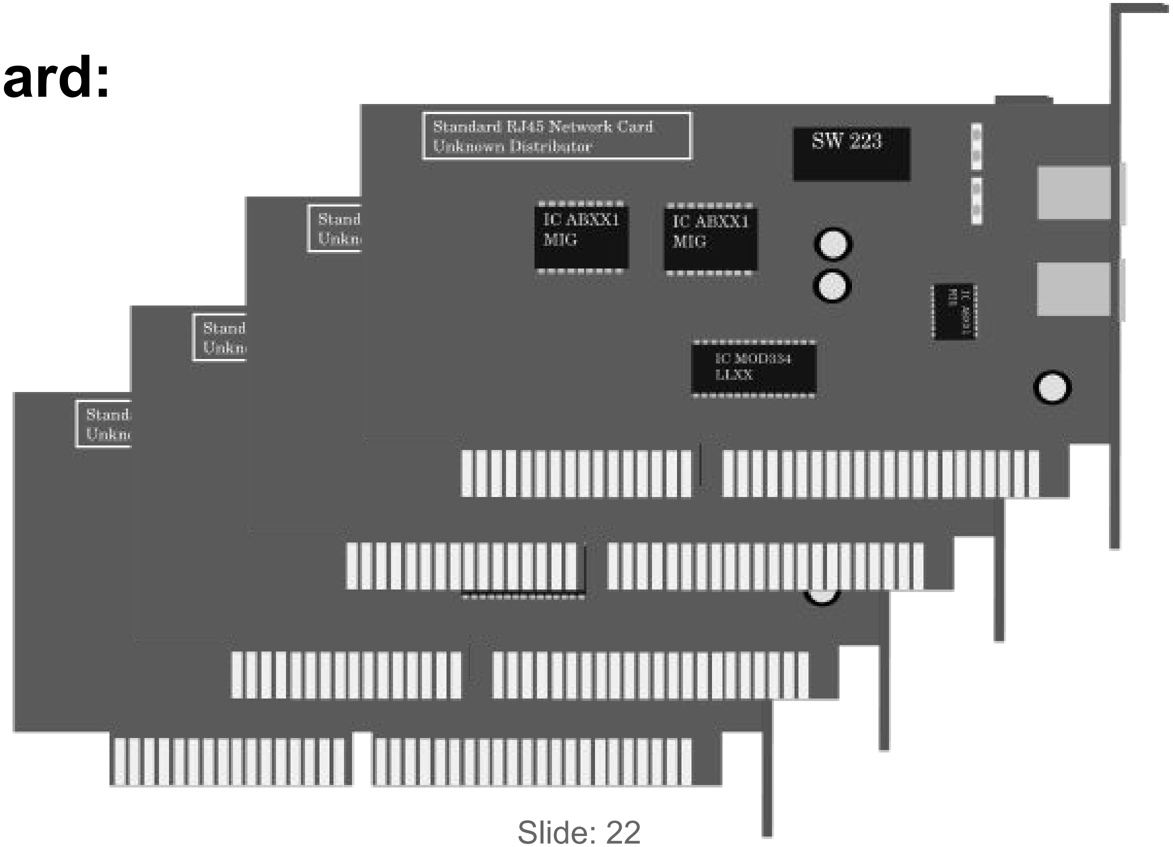


Using Hardware Virtualized I/O Devices

Use of hardware virtualized I/O devices so:

- There is no need for separate physical device for each OS
- But guest VMs still see separate devices

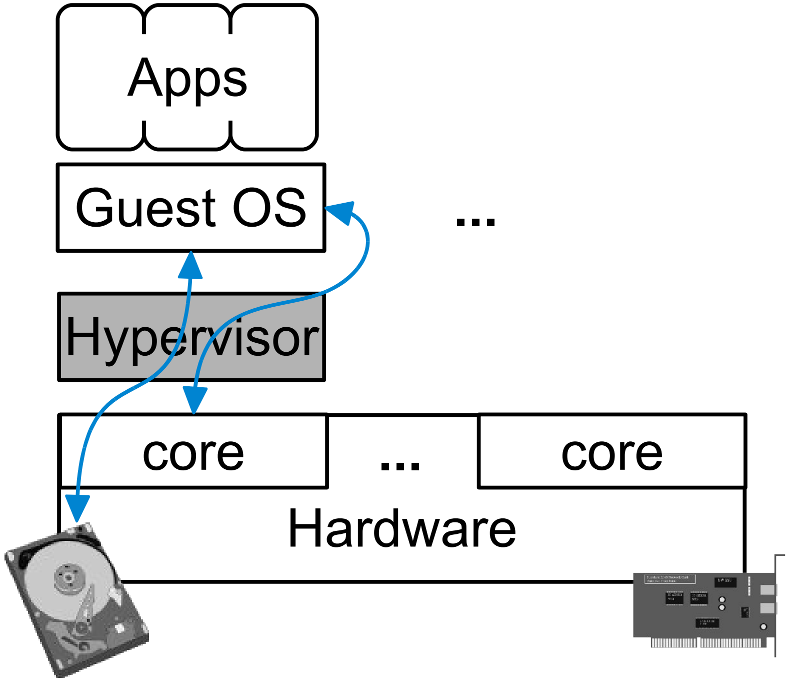
E.g. Network Card:



Short-Circuiting the System Discovery

System discovery, today:

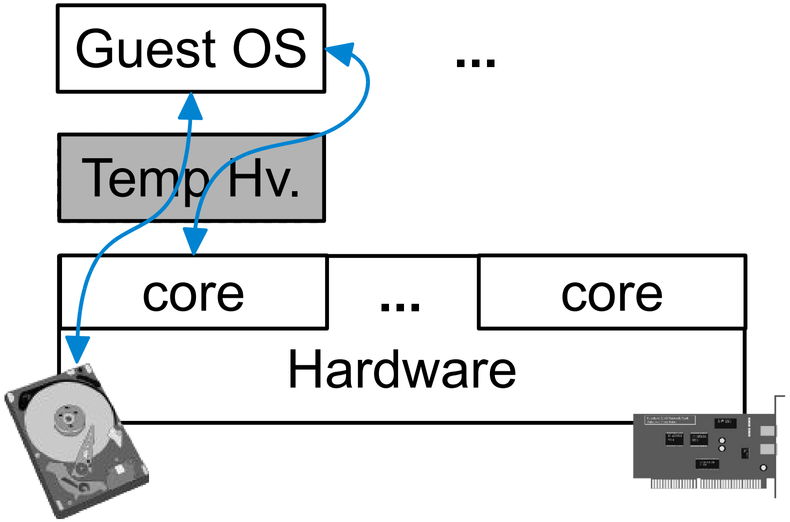
- Guest OS discovers functionality of underlying hardware
- Parts of discovery are not virtualizable today



Short-Circuiting the System Discovery

Short-circuit system discovery by:

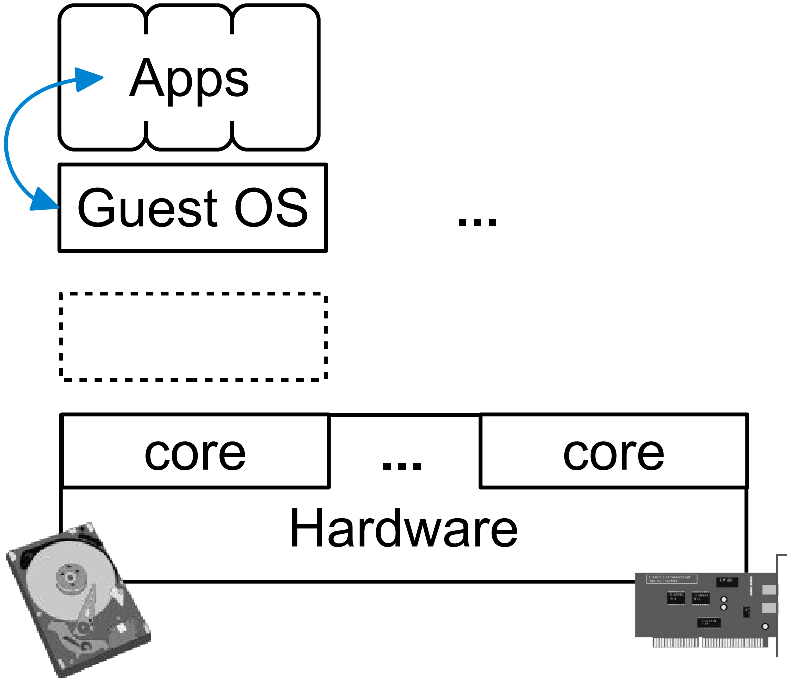
- Gathering **all** information at start of bootup
- Guest OS interacting with temporary hypervisor



Short-Circuiting the System Discovery

Short-circuit system discovery by:

- Gathering **all** information at start of bootup
- Guest OS interacting with temporary hypervisor
- Using stored information as VM runs

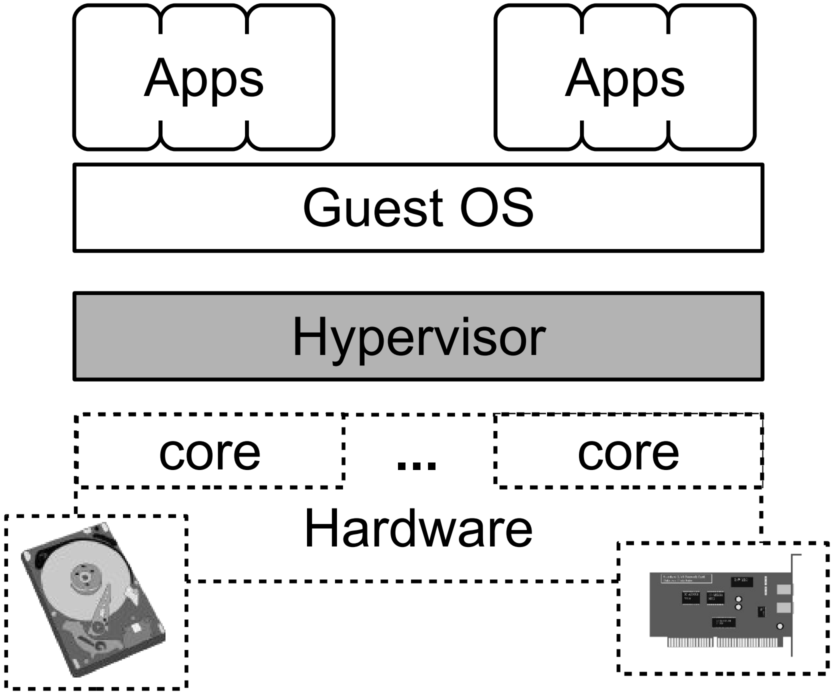


PRINCETON UNIVERSITY

Avoiding Indirection

Indirection, today:

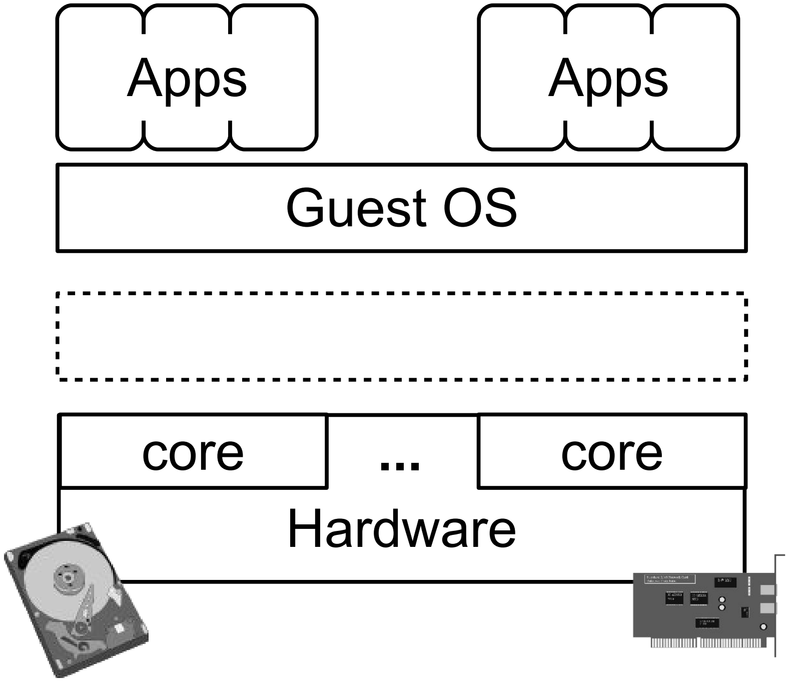
- Hypervisor presents abstract view of underlying hardware
- VMs can be scheduled on different cores
- Interrupts and timers require hypervisor involvement



Avoiding Indirection

NoHype avoids indirection by allowing guest VM to:

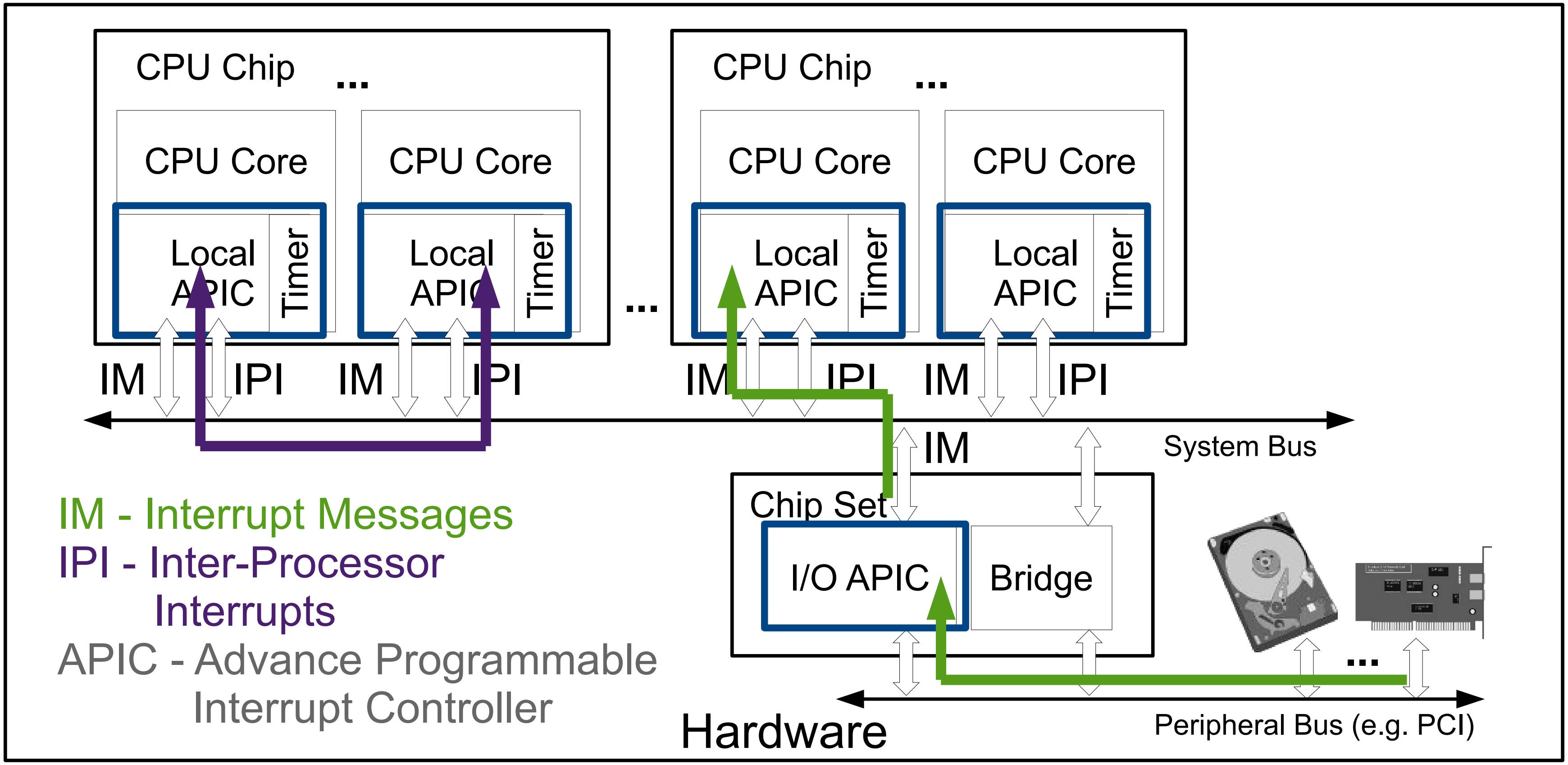
- Have more direct access to hardware
- Handle interrupts and timers



Avoiding Indirection

NoHype voids indirection by allowing guest VM to:

- Have more direct access to hardware
- Handle interrupts and timers



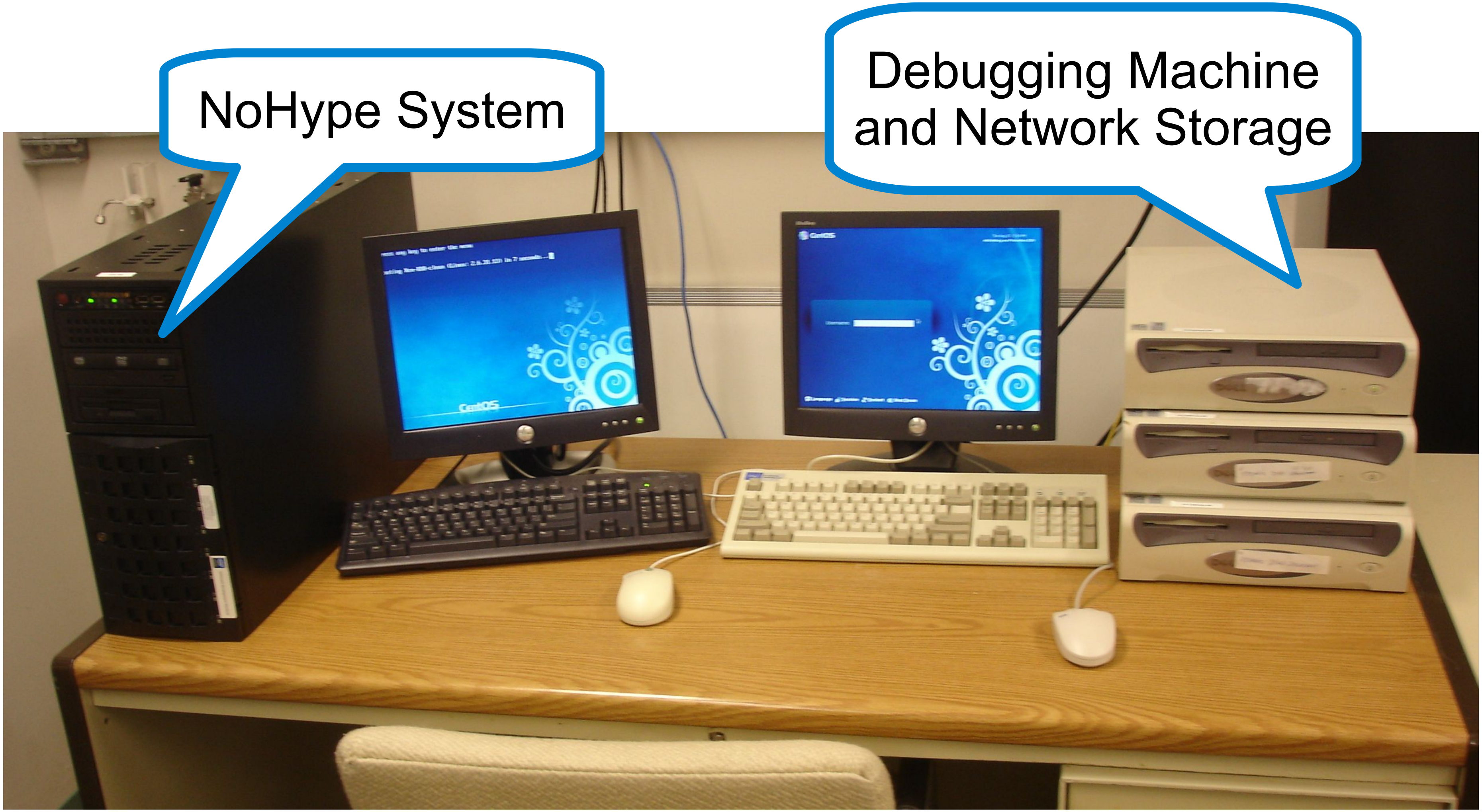
IM - Interrupt Messages

IPI - Inter-Processor Interrupts

APIC - Advance Programmable Interrupt Controller

Security and Performance Evaluation

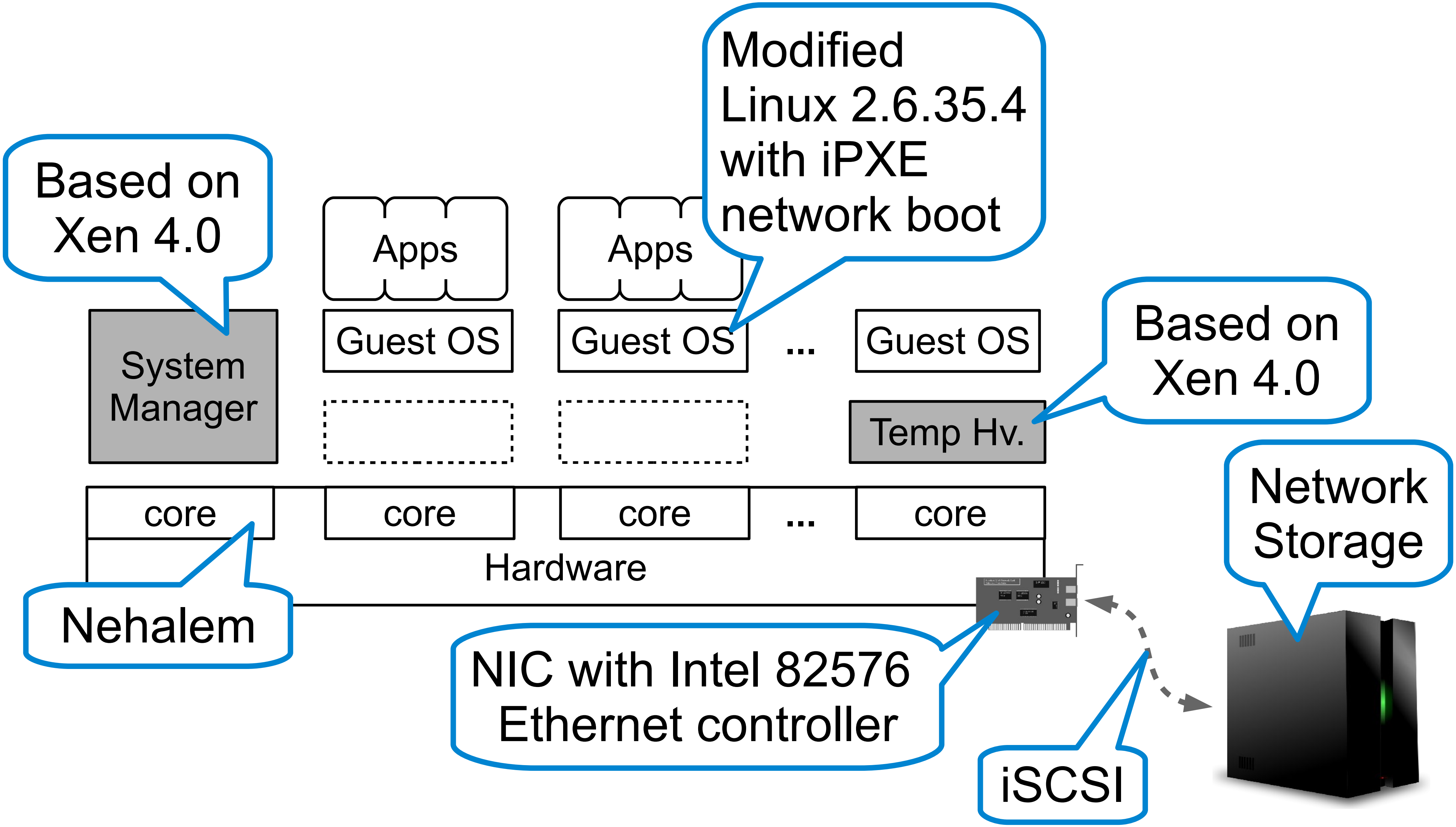
NoHype Implementation



NoHype System

Debugging Machine and Network Storage

NoHype Implementation



Based on Xen 4.0

System Manager

Apps

Guest OS

Apps

Guest OS

Modified Linux 2.6.35.4 with iPXE network boot

...

Guest OS

Based on Xen 4.0

Temp Hv.

core

core

core

...

core

Hardware

Network Storage

Nehalem

NIC with Intel 82576 Ethernet controller

iSCSI

Security Analysis: C.I.A.

- We improve confidentiality and integrity protection:
  - e.g. no device emulation that could be exploited to access or modify other VM's data or code

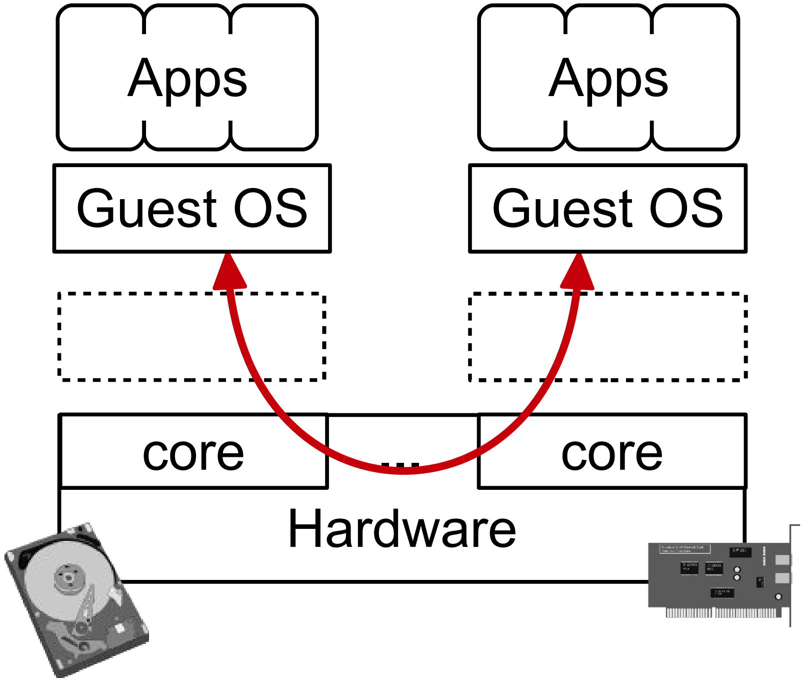
- We improve availability:
  - e.g. no VM exits, significantly harder to trigger a bug and crash the system

- We reduce side channels:
  - e.g. dedicated CPU cores, no sharing of L1 caches

Sample Security Evaluation

Bringing guest OS closer to hardware opens a new attack:

- Malicious interprocessor interrupts (IPIs) between guests

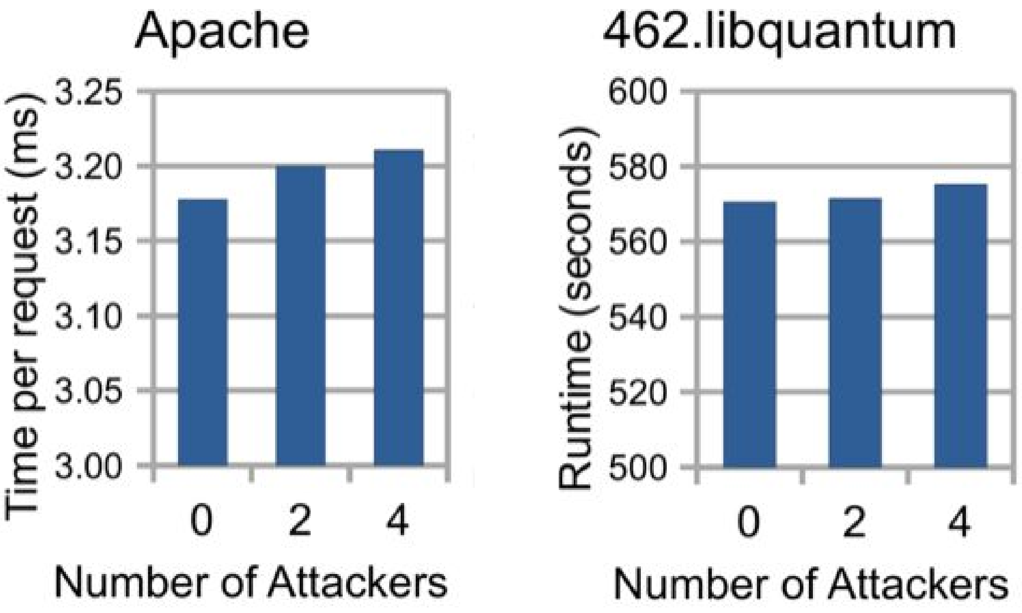


Sample Security Evaluation

VM to VM attack using inter-processor interrupts (IPIs):

- Software defense available

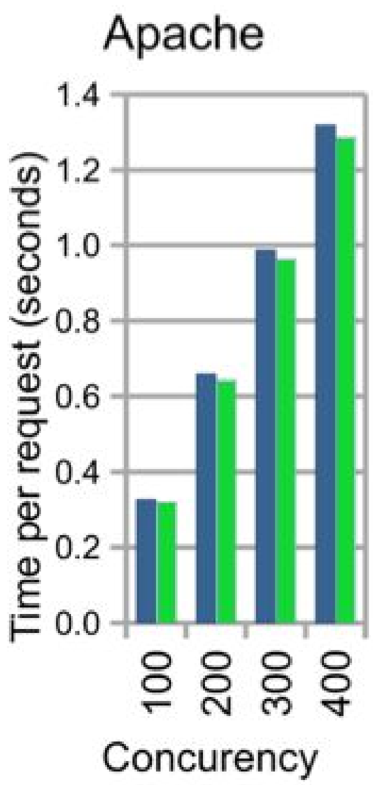
- Limited impact on guest VM performance

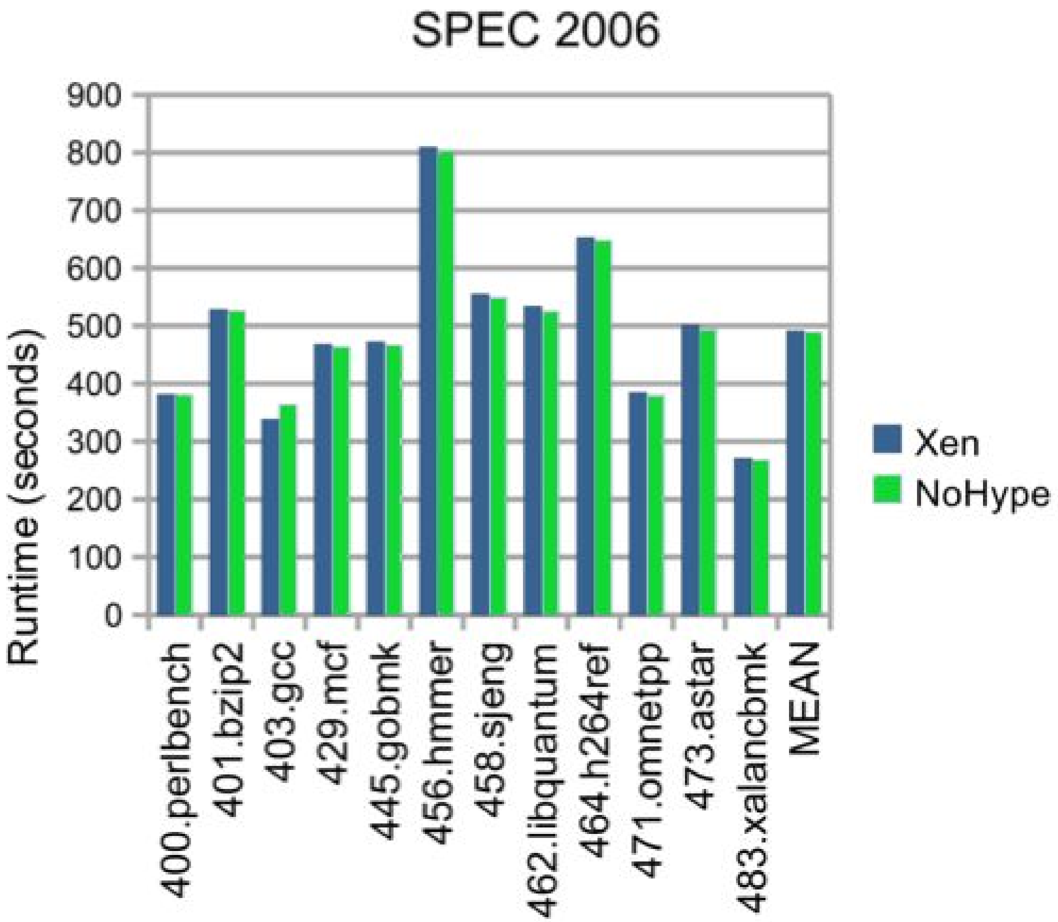


More evaluation and security analysis is in the paper.

Sample Performance Evaluation

NoHype shows about 1% performance improvement:





Summary

- Rethinking of virtualization for cloud computing:
  - some things don't need to be done at all,
  - some functionality can be done in hardware, and
  - certain things can be done entirely during boot.
- Improved security by eliminating hypervisor attack surface through the VM Exits.
- Better security and performance.

Ongoing Work and Opportunities

Ongoing work:

- VM migration
- Nested virtualization
- Software switch for networking
- Hardware modification for protecting VMs

Research Opportunities:

- ***How can we refactor system functionality for better security and performance by embracing unique opportunities offered by cloud computing paradigms?***

Thank You.