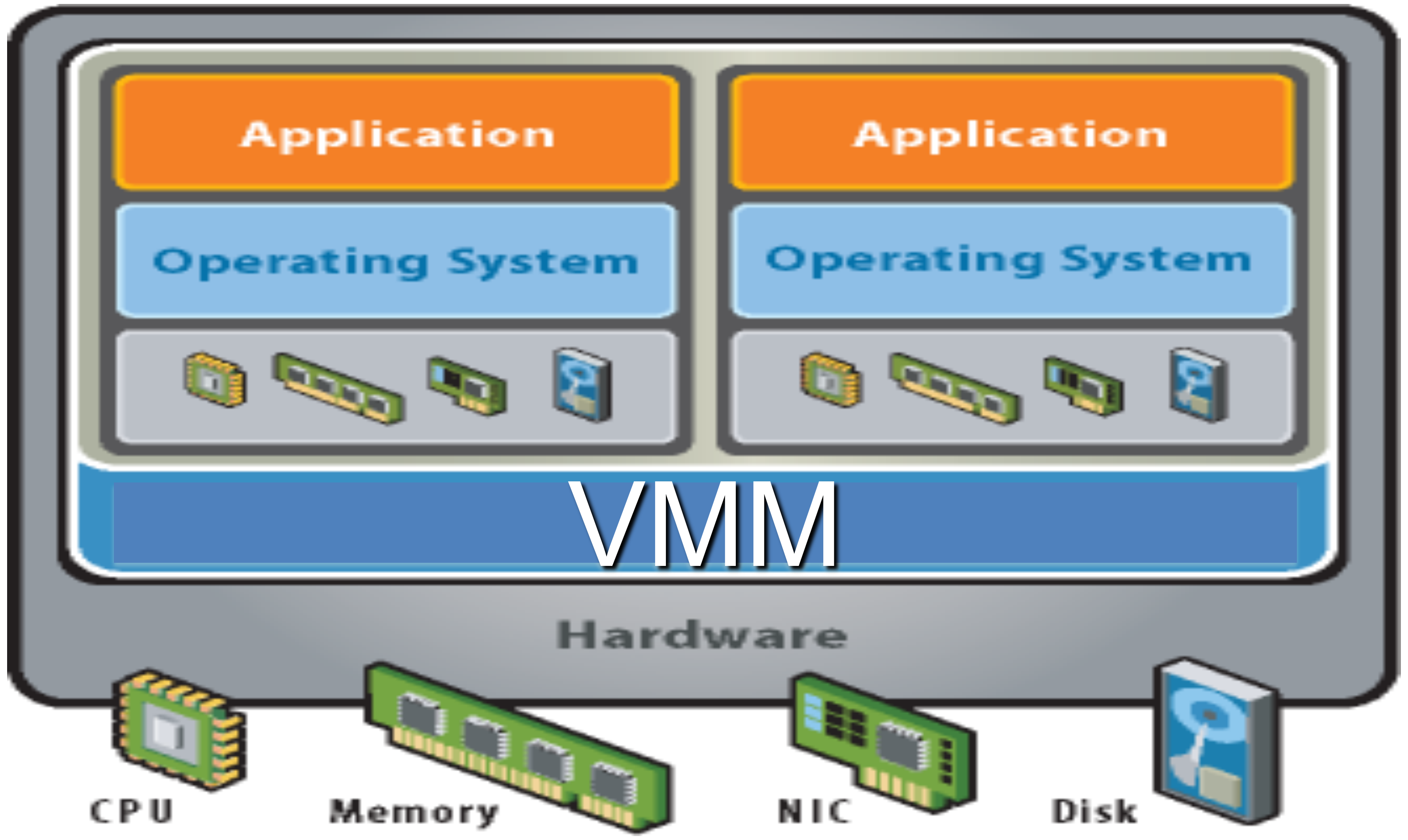
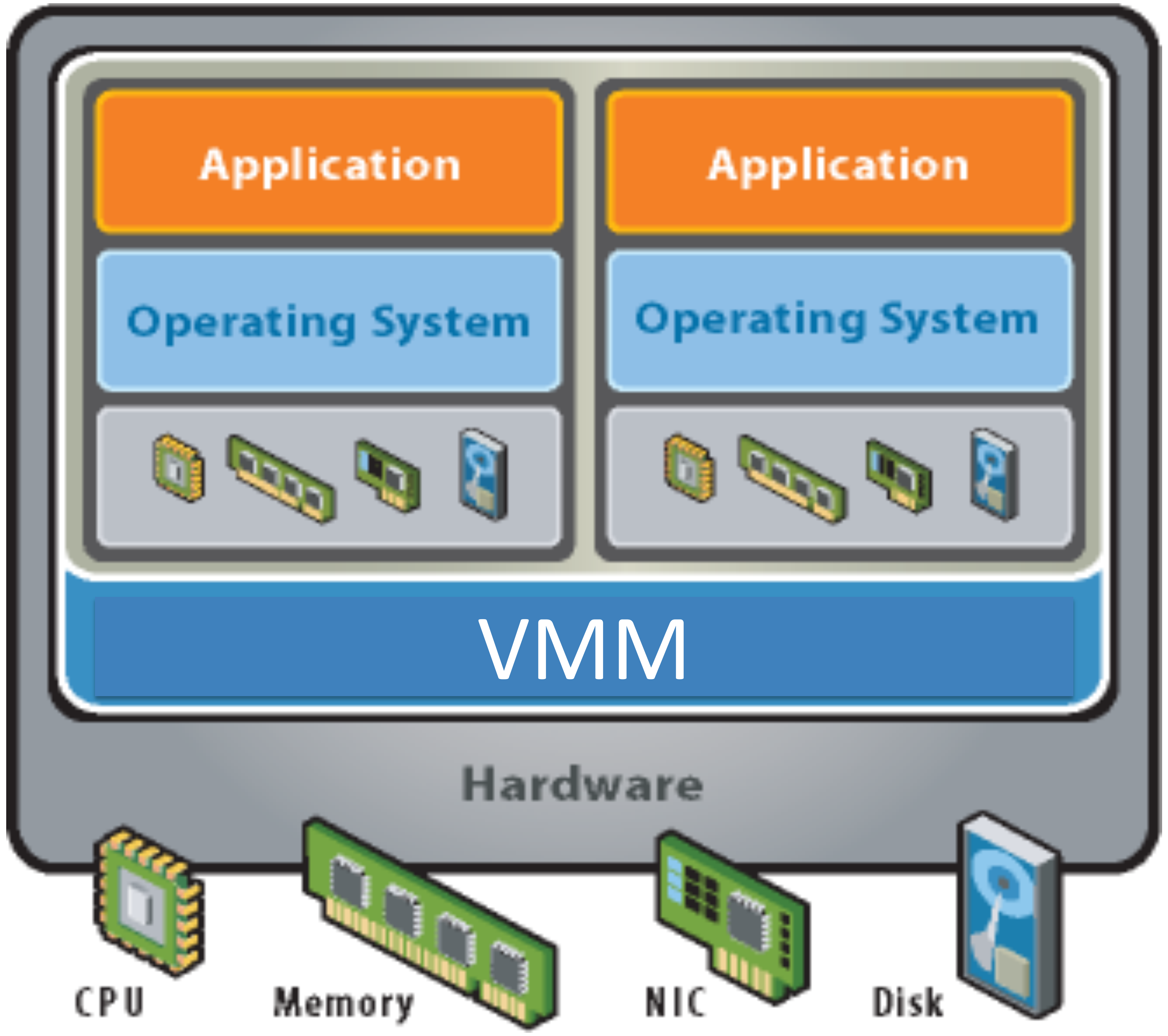
Virtualization



CS522 – Principles of Computer Systems

Edouard Bugnion

Virtual machines– undergraduate lecture



Fundamental Attributes:

- Equivalence – essential identical environment
- Performance – nearly the same speed
- Safety – the VMM is in complete control

The Popek / Goldberg Principles [1974]

Popek and Golberg: Proving it formally [1974]

EPFL

For any conventional computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.

$$\{\textit{control-sensitive}\} \cup \{\textit{behavior-sensitive}\} \subseteq \{\textit{privileged}\}$$

Proof by construction (of the VMM)

The proof

- All you need is a minimalistic architecture with support for:
  - Kernel mode and user mode
  - Virtual memory (e.g. via segmentation)
- Don’t screw up the instruction set architecture
- Don’t screw up virtual memory
- Sketch the implementation of the VMM that
  - Runs the application in user mode
  - Runs the kernel in user mode

Q.E.D.

Details are in the paper (how hard can it be)

Great contributions from 1974

Formal Requirements for Virtualizable Third Generation Architectures

Gerald J. Popek
University of California, Los Angeles
and
Robert P. Goldberg
Honeywell Information Systems and
Harvard University

Virtual machine systems have been implemented on a limited number of third generation computer systems, e.g. CP-67 on the IBM 360/67. From previous empirical studies, it is known that certain third generation computer systems, e.g. the DEC PDP-10, cannot support a virtual machine system. In this paper, model of a third-generation-like computer system is developed. Formal techniques are used to derive precise sufficient conditions to test whether such an architecture can support virtual machines.

CACM '74

ABBA '74

Great corrections since 1974

Disco Demolition night, Chicago, IL, 7/12/1979

ABBA '74

Known corrections following 1974

EPFL

Formal Requirements for Virtualizable Third Generation Architectures

Gerald J. Popek
University of California, Los Angeles
and
Robert P. Goldberg
Honeywell Information Systems and
Harvard University

Virtual machine systems have been implemented on a limited number of third generation computer systems, e.g. CP-67 on the IBM 360/67. From previous empirical studies, it is known that certain third generation computer systems, e.g. the DEC PDP-10, cannot support a virtual machine system. In this paper, model of a third-generation-like computer system is developed. Formal techniques are used to derive precise sufficient conditions to test whether such an architecture can support virtual machines.

CACM '74

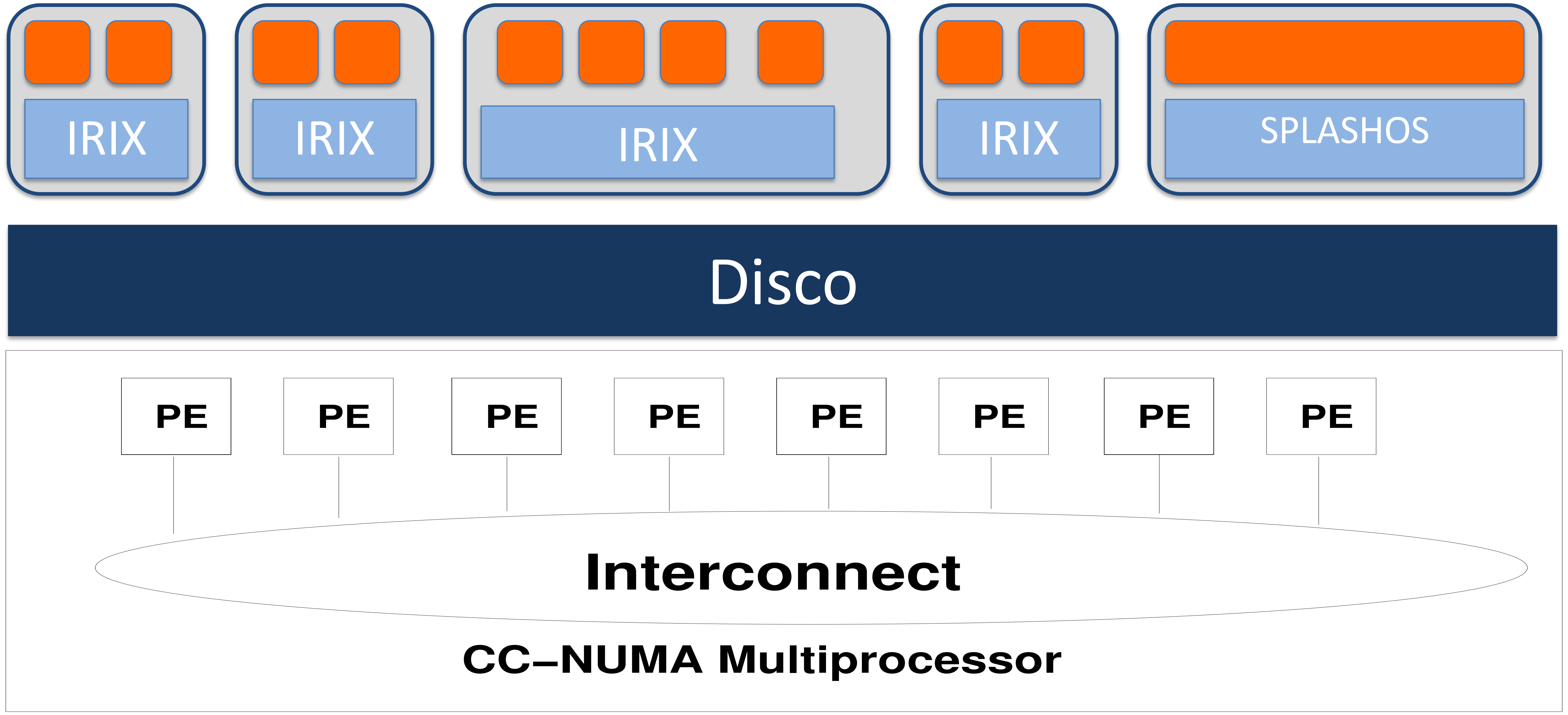
Disco: Running Commodity Operating Systems on Scalable Multiprocessors

EDOUARD BUGNION, SCOTT DEVINE, KINSHUK GOVIL, and MENDEL ROSENBLUM
Stanford University

In this article we examine the problem of extending modern operating systems to run efficiently on large-scale shared-memory multiprocessors without a large implementation effort. Our approach brings back an idea popular in the 1970s: virtual machine monitors. We use virtual machines to run multiple commodity operating systems on a scalable multiprocessor. This solution addresses many of the challenges facing the system software for these machines. We demonstrate our approach with a prototype called Disco that runs multiple copies of Silicon Graphics' IRIX operating system on a multiprocessor. Our experience shows that the overheads of the monitor are small and that the approach provides scalability as well as the ability to deal with the nonuniform memory access time of these systems. To reduce the memory overheads associated with running multiple operating systems, virtual machines transparently share major data structures such as the program code and the file system buffer cache. We use the distributed-system support of modern operating systems to export a partial single system image to the users. The overall solution achieves most of the benefits of operating systems customized for scalable multiprocessors, yet it can be achieved with a significantly smaller implementation effort.

SOSP '97, TOCS '97

Stanford FLASH + Disco



Disco – highly-scalable Virtual Machine Monitor

Uses virtual machines to create a virtual cluster

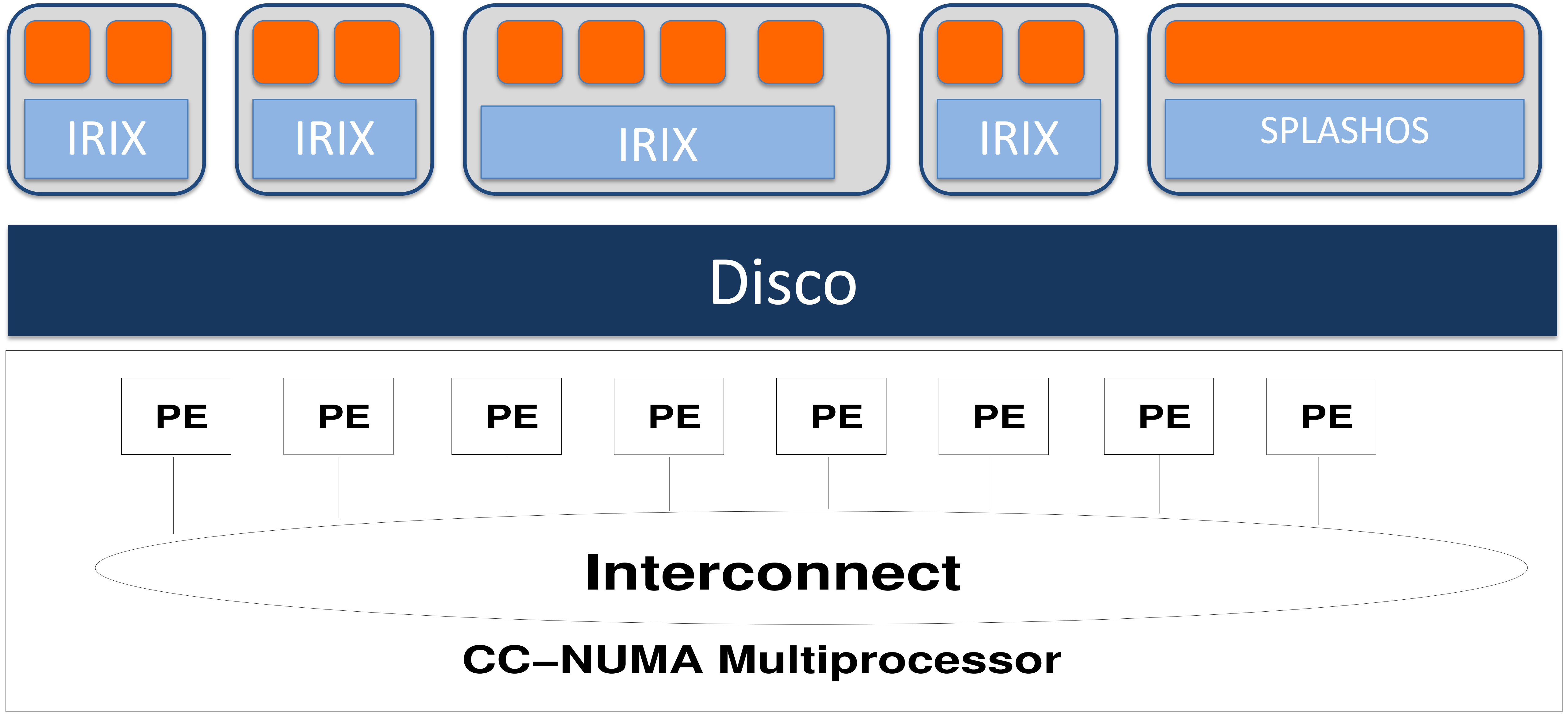
- Runs commodity OS in virtual machines

Challenge – how to approximate a single system image

- At least from an efficiency perspective …
- … with **mostly** unmodified operating systems

Stanford FLASH + Disco

EPFL



Disco – highly-scalable Virtual Machine Monitor

Uses virtual machines to create a virtual cluster

Totally academic treatment

- At least from an efficiency perspective …
- … with **<u>mostly</u>** unmodified operating systems

From Disco to VMware

<u>Disco</u>: designed for big UNIX servers

Stick with MIPS servers

Port to Alpha servers and support Windows, VMS, ULTRIX

Prove on desktops
Roadmap servers

From Disco to VMware

Disco: designed for big UNIX servers

Stick with MIPS servers

Port to Alpha servers

ows,

Prove on desktops
Roadmap servers

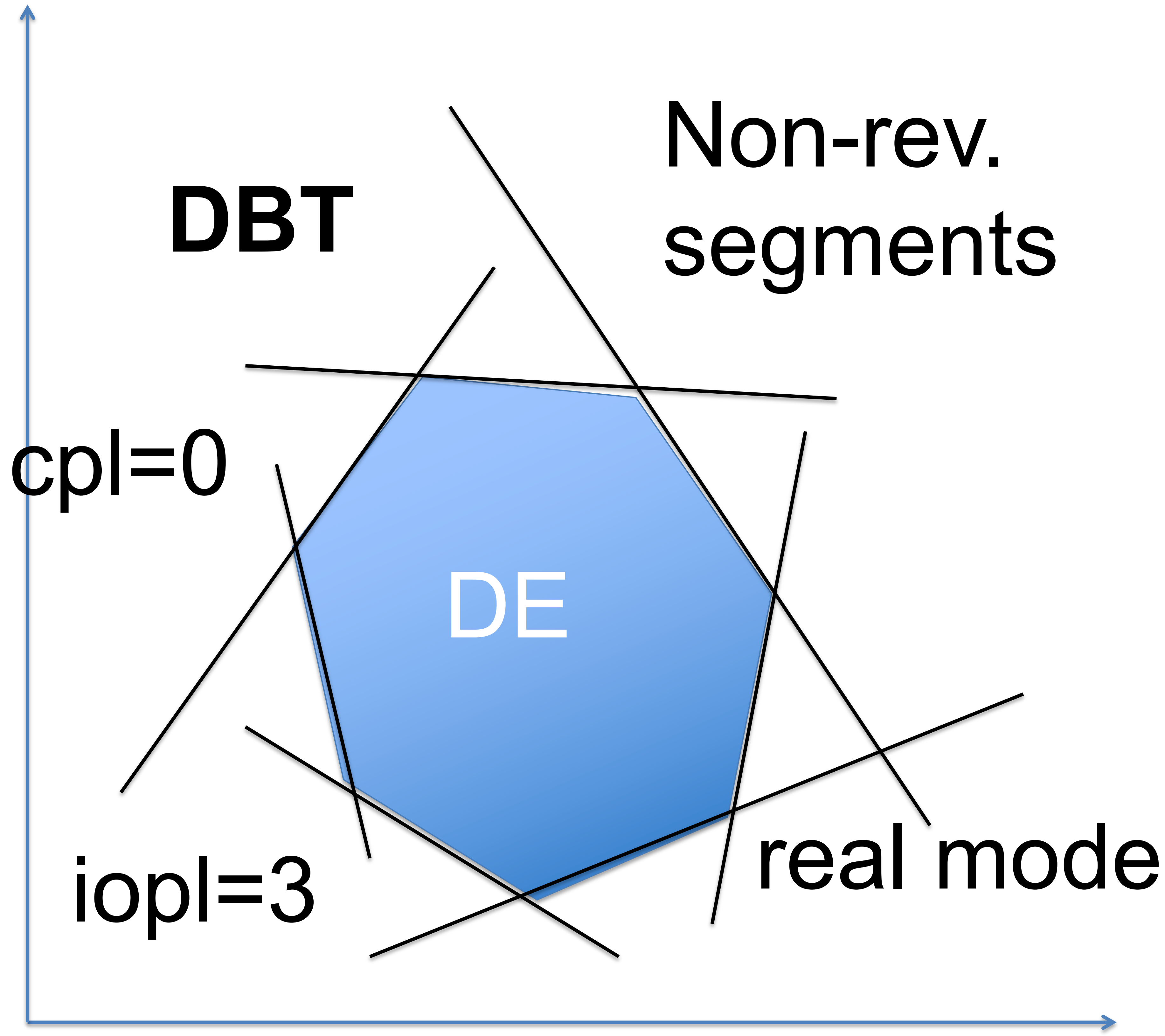
The 3rd theorem from Popek and Goldberg [1974]

A hybrid virtual machine monitor may be constructed for any conventional third-generation machine in which the set of user-sensitive instructions are a subset of the set of privileged instructions.

“If you only screw up the architecture for kernel mode,
you can still use direct execution while handling user mode”

VMware Workstation [1999] [TOCS 2012]

EPFL



- Equivalence – essential identical environment (minor holes)
- Performance – mostly
- Safety – yes

- Encapsulation – checkpoint → live migration
- Hardware-independence – I/O emulation

New attributes would prove essential for the cloud

Virtualization today – hardware support

EPFL

Ubiquitously deployed, with broad hardware support:

- CPU – VT-x, AMD-v and ARM-v
- MMU – Extended Page Tables (aka Nested Page Tables)
- Interrupts – vAPIC, ELI, …
- I/O Bus – IOMMU
- I/O Device – SR-IOV
- Network Virtualization / overlays: VXLAN, Geneve, …

And yet:

- Segments were removed in x86-64 …. Breaks application sandboxing
- Intel introduced SGX, but unclear if it works with VT-x

Virtualization today – hardware support

Ubiquitously deployed, with broad hardware support:

- CPU – VT-x, AMD-v and ARM-v
- MMU – Extended Page Tables (aka Nested Page Tables)
- Interrupts – vAPIC, ELI, …
- I/O Bus – IOMMU
- I/O Device – SR-IOV
- Network Virtualization / overlays: VXLAN, Geneve, …

Caution: history might repeat itself !

And yet:

- Segments were removed in x86-64 …. Breaks application sandboxing
- Intel introduced SGX, but unclear if it works with VT-x

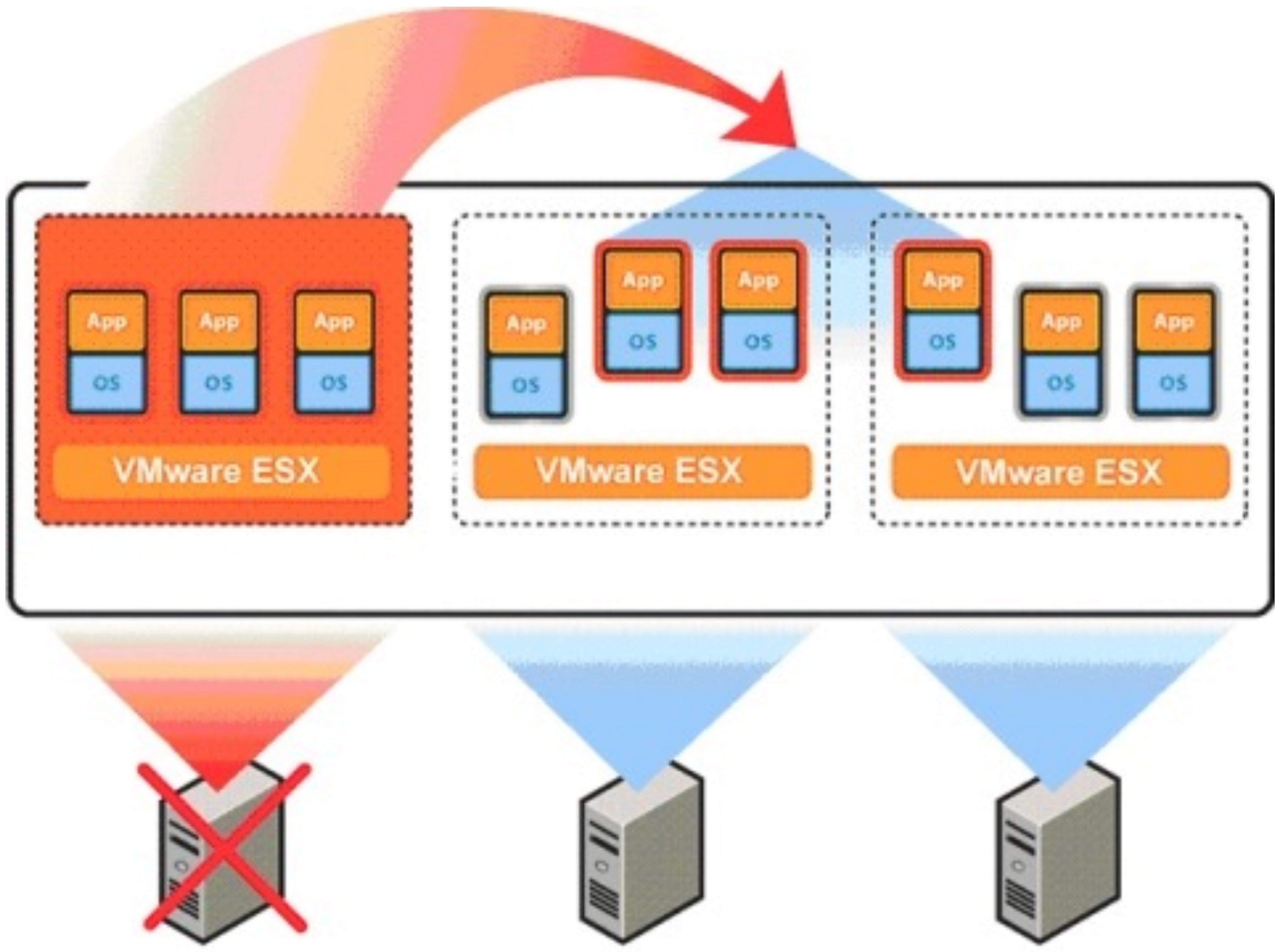
Quiz

Why does the cloud use VMs ?

Virtualization today – a foundation for:

- Scalable multi-tenancy (a.k.a. “cloud”)
- Server consolidation
- Storage consolidation and virtualization
- Appliance distribution
- High-availability
- Live migration / Distributed resource scheduling
- Network virtualization
- Hyper-converged deployments (server + storage + network functions)
- …

Virtual Machine High Availability



Decouple VM from underlying hardware

Ensure that it restarts on another node in the case of a failure

In-class design project

Design a Hyper-converged virtual infrastructure