Z6110X0035: Introduction to Cloud Computing

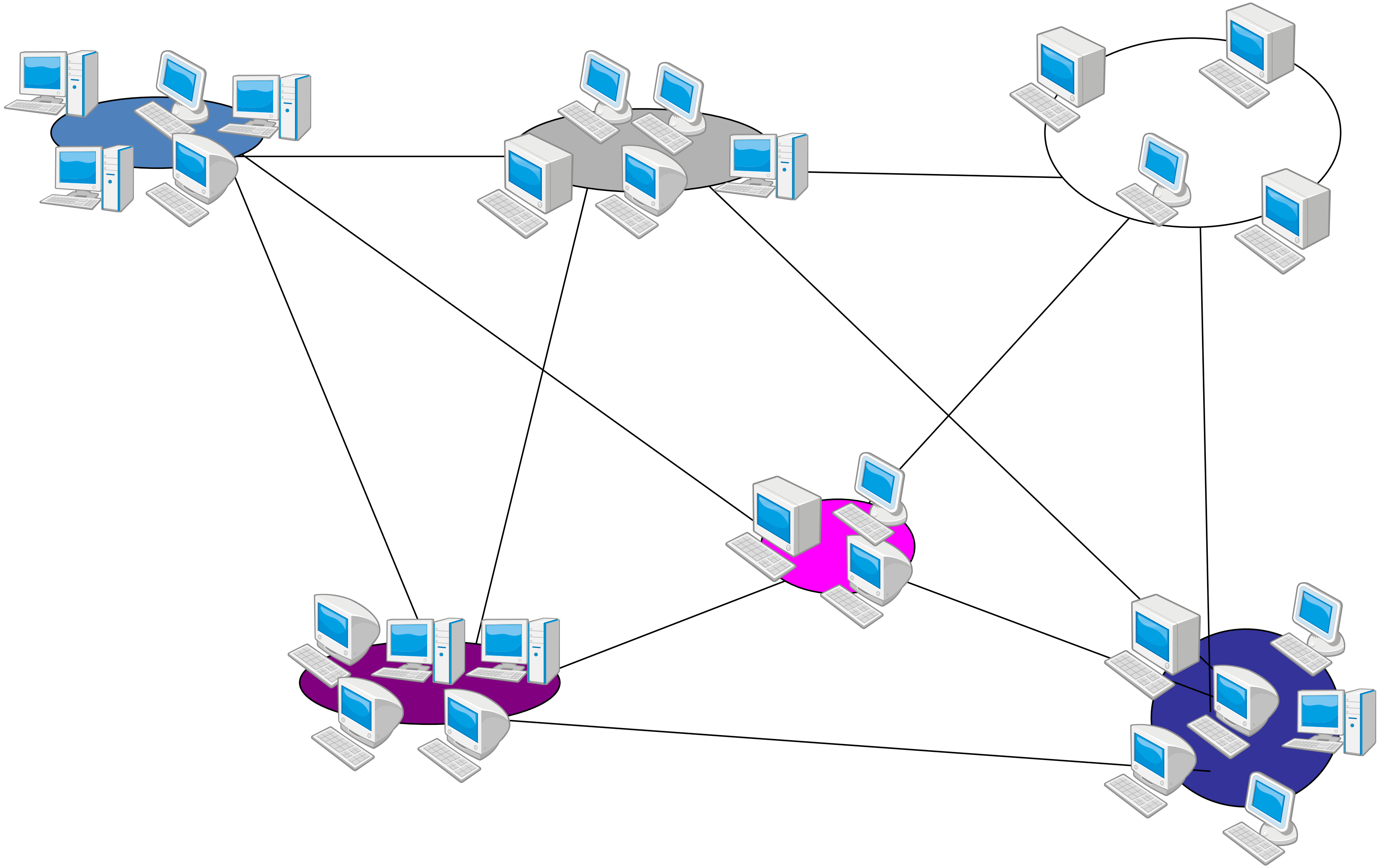
– **Virtualization**

Lecturer: Prof. Zichen Xu

Outline

The needs of virtualization
The concepts
Types of virtualization
Issues in virtualization
Implementation cases
Conclusion

In the computer-age...



A Lot of Servers/Machines...

Web server
Mail server
Database server
File server
Proxy server
Application server
…and many others

A Lot of Servers/Machines...

The data-centre is FULL
- Full of under utilized servers
- Complicate in management

Power consumption
- Greater wattage per unit area than ever
- Electricity overloaded
- Cooling at capacity

Environmental problem
- Green IT

Problem (continued)

Adding or upgrading hardware or OS is difficult

Testing and refitting active service
Complicated changeover tactics
…

Load balancing is impossible

Services tied to own systems
Some underused, some overused

Modest Example — Good's Goodlab cluster

Approx 20 difference services
Approx 20 server systems

- Approx. 80 processors
- > 1 terabytes of RAM
- ~ 20 terabytes of disk storage
- Multiple operating systems

Solution — Virtualization

Decouple [*OS, service*] pair from hardware
Multiplex lightly-used services on common host hardware
Migrate services from host to host as needed
Introduce new [*OS, service*] pairs as needed

- Commissioning new services
- Testing upgrades of existing services
- Experimental usage
- …

Virtual Machine

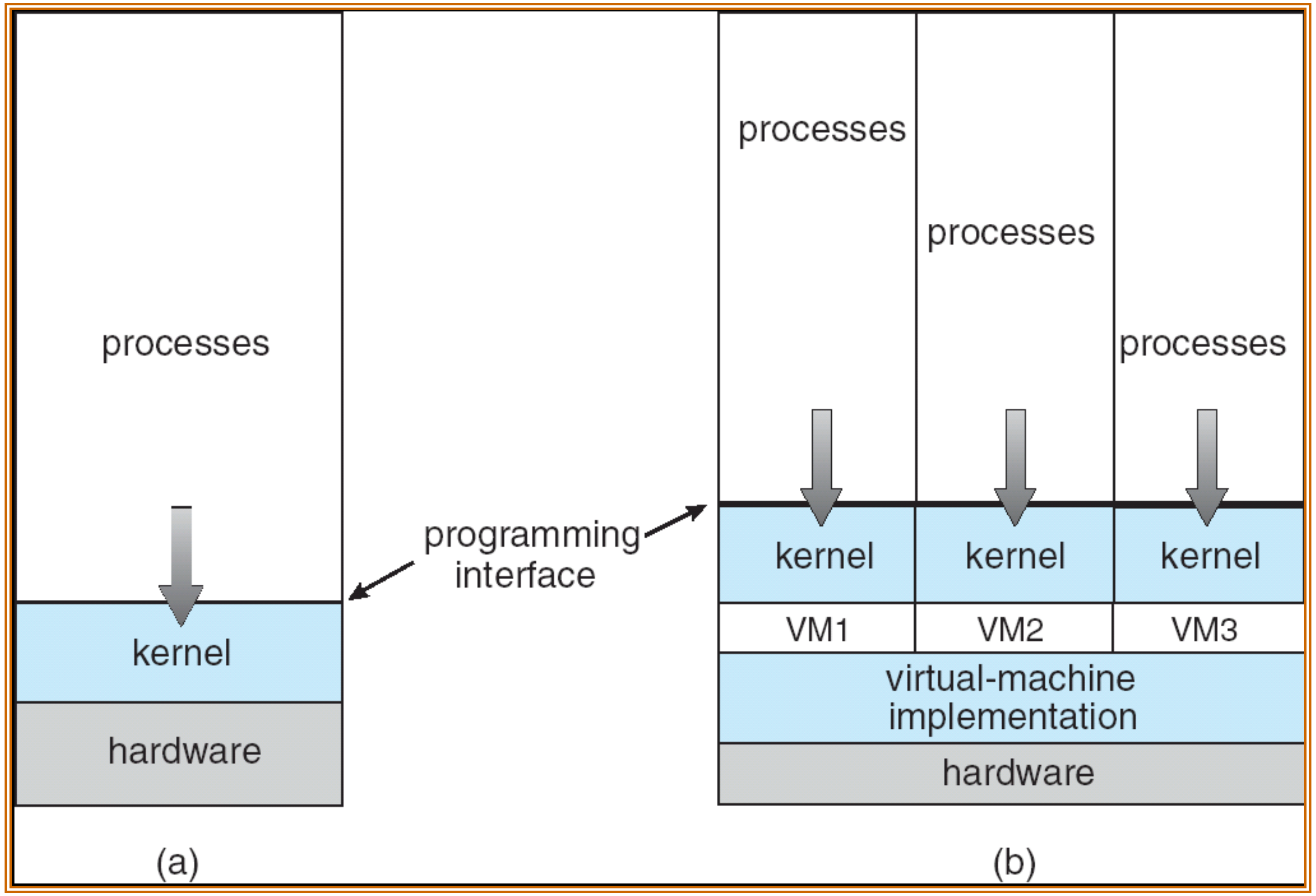
A virtual machine provides interface *identical* to underlying bare hardware

- I.e., all devices, interrupts, memory, page tables, etc.

Virtual Machine Operating System creates illusion of multiple processors

- Each capable of executing independently
- No sharing, except via network protocols
- Clusters and SMP can be simulated

Virtual Machines



(a) Nonvirtual machine (b) virtual machine

History – CP67 / CMS

IBM Cambridge Scientific Center

Ran on IBM 360/67

- Alternative to TSS/360, which never sold very well

Replicated hardware in each "process"

- Virtual 360/67 processor
- Virtual disk(s), virtual console, printer, card reader, etc.

CMS: *Cambridge Monitor System*

- A single user, interactive operating system

Commercialized as VM370 in mid-1970s

Virtualization

Virtualization -- the abstraction of computer resources.

Virtualization hides the physical characteristics of computing resources from their users, be they applications, or end users.

This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple virtual resources; it can also include making multiple physical resources (such as storage devices or servers) appear as a single virtual resource.

Why now?

1960—1999

IBM, CP-40, CP/CMS, S/360-370, VM370, Virtual PC, VMware

2000—2005

IBM z/VM, Xen

2006

Intel VT-x

AMD's AMD-V

2008—

History (continued)

"Hypervisor" systems – mid 1970s⇒mid 1990s

- Large mainframes (IBM, HP, etc.)
- Internet hosting services
- Virtual dedicated services
- …

Modern Virtualization Systems

- VMware
  - *Workstation* and *Player*
  - Multiple versions of *VMware Server*
  - Virtual appliances
- Xen
  - Public domain hypervisor
  - Adaptive support in operating systems
  - Emerging support in processor chips
    - Intel, AMD
- Macintosh *Parallels*

Virtualization being embraced by major OS vendors

Red Hat Enterprise Linux
Suse Enterprise Linux
Microsoft *Longhorn* server (est. 2007-2008)
...

(*Red Hat*) Marketing "Promises"

Freedom from upgrades

If new OS version causes problems with a service, keep old OS version for that service

Security

Reduces potential number of users logging into a service

Reduces undesirable sharing

Narrows scope of attacks

Development and Testing

Viable platform for developers in quasi-real environment

Reduces number of test machines

Automated scripts for intensive testing, crash records, etc.

…

(*Red Hat*) Marketing "Promises" (continued)

- Live Migration – move services from one host to another while still running
  - No interruption in service visible to clients
  - Preparation for taking down hardware for maint.
  - Preparation for heavy batch run, etc.
- Failure Isolation
  - Crash of one service does not affect other services
  - Particularly on SMP system
  - Hot backups of services can be maintained

(SUSE) Marketing "Promises"

Increased server hardware utilization

- Consolidate disparate services on hardware
- Lower capital, maintenance, and energy costs

Rebalancing loads to meet peak demands

- Adjust for time-of-day differences

Application portability across platforms

…

Hardware evolution

- Faster CPU clock than ever
  - Though almost hit its top
- More CPU cores in a single chip
  - 4-core CPUs already in the market
  - 6- or 8-core CPUs will be there soon
- Multi-core architectures make parallel processing more realizable
- Virtualization support on chip from CPU manufacturers (e.g., Intel, AMD)

Software maturity

More than one credible player in the market

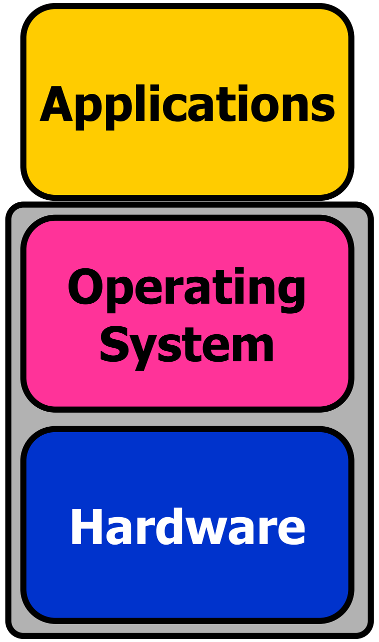
Available and stable open-sourced software

- OS, DB, Web server, Java, PHP, gcc, etc.

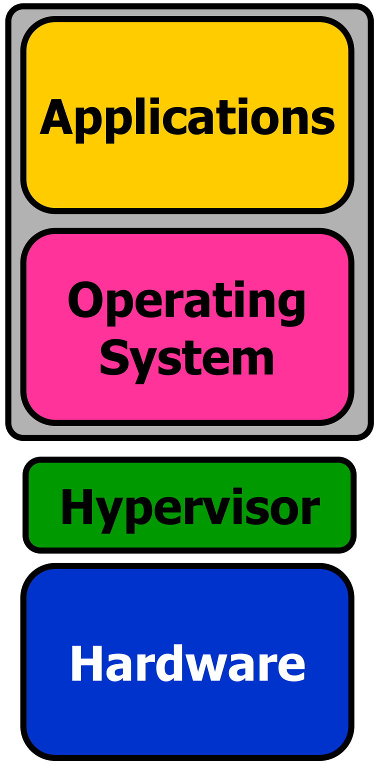
Established and mature software standards

- Web service, XML, SOAP, COM, etc.

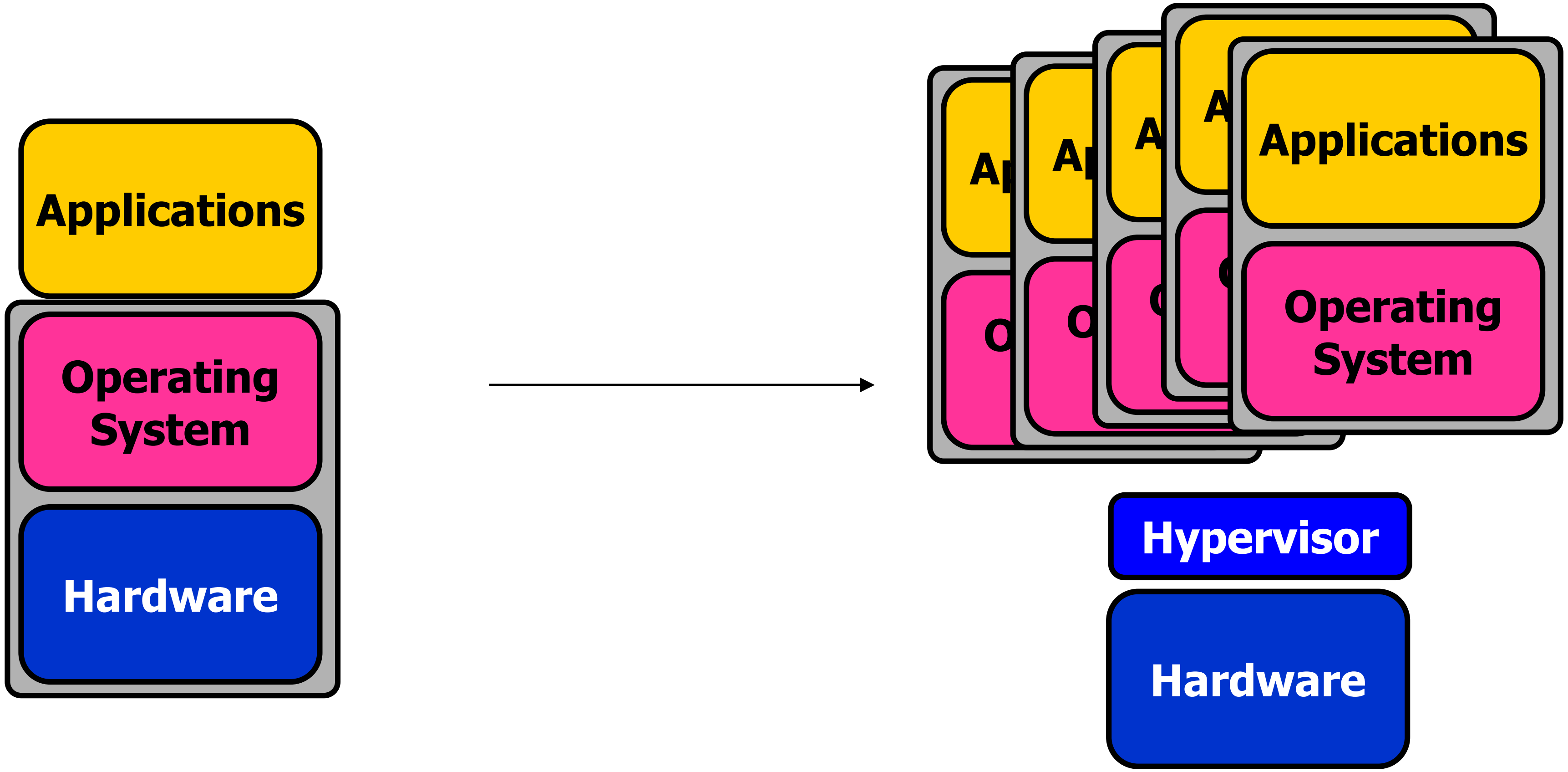
The Use of Computers



Virtualization

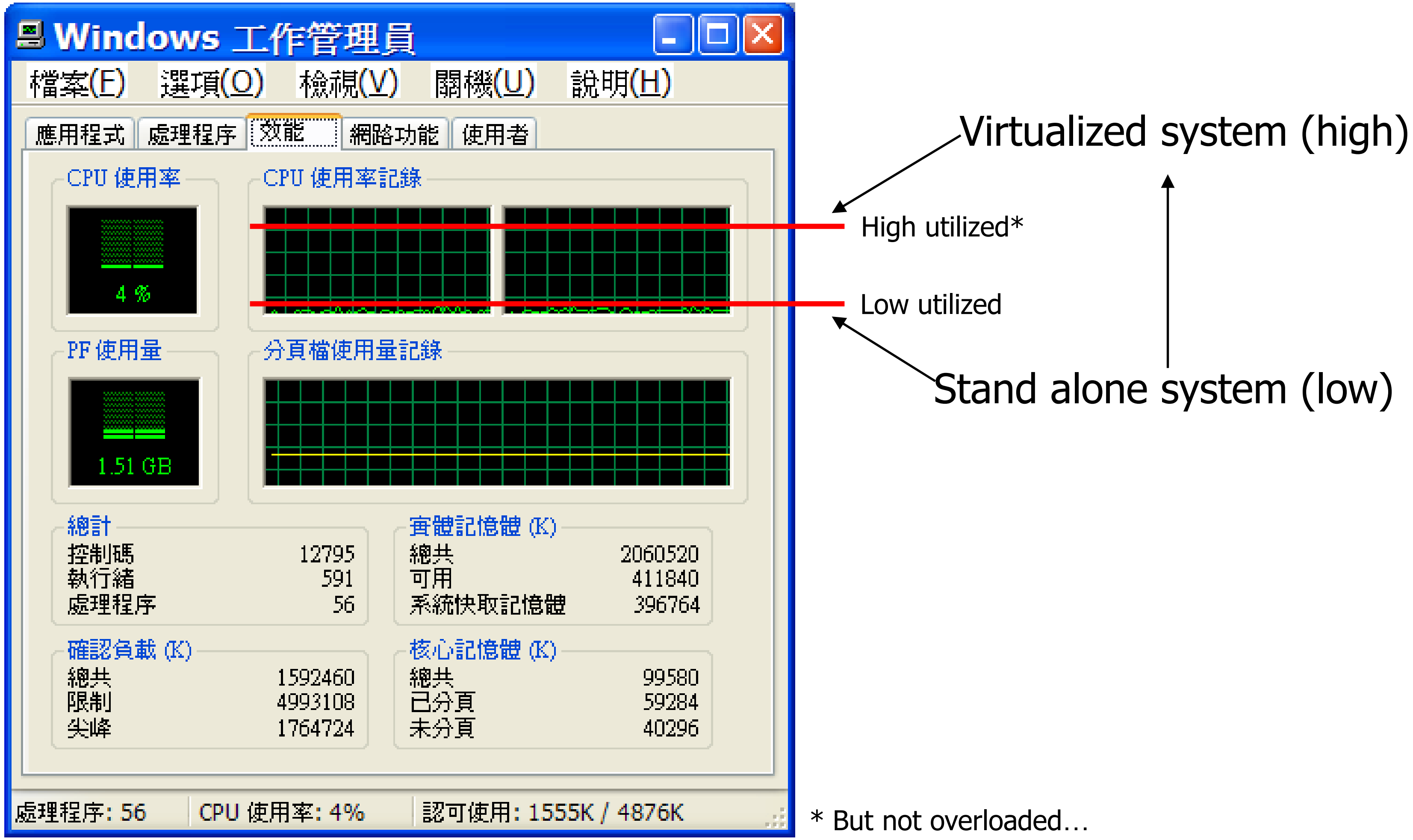


Virtualization -- a Server for Multiple Applications/OS



Hypervisor is a software program that manages multiple operating systems (or multiple instances of the same operating system) on a single computer system.
The hypervisor manages the system's processor, memory, and other resources to allocate what each operating system requires.
Hypervisors are designed for a particular processor architecture and may also be called **virtualization managers**.

Capacity Utilization



* But not overloaded…

Types of Virtualization

- Virtual memory
- Desktop virtualization
- Platform virtualization
  - Full virtualization
  - Paravirtualization
  - Hardware-assisted virtualization
  - Partial virtualization
  - OS-level virtualization
  - Hosted environment (e.g. User-mode Linux)
- Storage virtualization
- Network virtualization
- Application virtualizationPortable application
  - Cross-platform virtualization
  - Emulation or simulation
  - Hosted Virtual Desktop

Category in Wiki

In this talk, we mainly focus on Platform virtualization which is mostly related to clou computing

- Full virtualization
- Binary transaltion
- Hardware-assisted virtualization
- Paravirtualization
- OS-level virtualization
- Hosted environment (e.g. User-mode Linux)

- Hardware level
- Operating system level
- Application level

Full Virtualization

A certain kind of virtual machine environment: one that provides a complete simulation of the underlying hardware.
The result is a system in which all software (including all OS's) capable of execution on the raw hardware can be run in the virtual machine.
Comprehensively simulate all computing elements as instruction set, main memory, interrupts, exceptions, and device access.
Full virtualization is only possible given the right combination of hardware and software elements.

Full virtualization has proven highly successful

- Sharing a computer system among multiple users
- Isolating users from each other (and from the control program) and
- Emulating new hardware to achieve improved reliability, security and productivity.

Full Virtualization

It needs a single machine that could be multiplexed among many users. Each such virtual machine had the complete capabilities of the underlying machine, and (for its user) the virtual machine was indistinguishable from a private system.

Examples

- First demonstrated with IBM's CP-40 research system in 1967
- Re-implemented CP/CMS in IBM's VM family from 1972 to the present.
- Each CP/CMS user was provided a simulated, stand-alone computer.

Full Virtualization

Virtualization requirements (by Popek and Goldberg) :

Equivalence: a program running under the VMM should exhibit a behavior essentially identical to that demonstrated when running on an equivalent machine directly;
Resource control (safety): the VMM must be in complete control of the virtualized resources;
Efficiency: a statistically dominant fraction of machine instructions must be executed without VMM intervention.

VMM: Virtual Machine Monitor

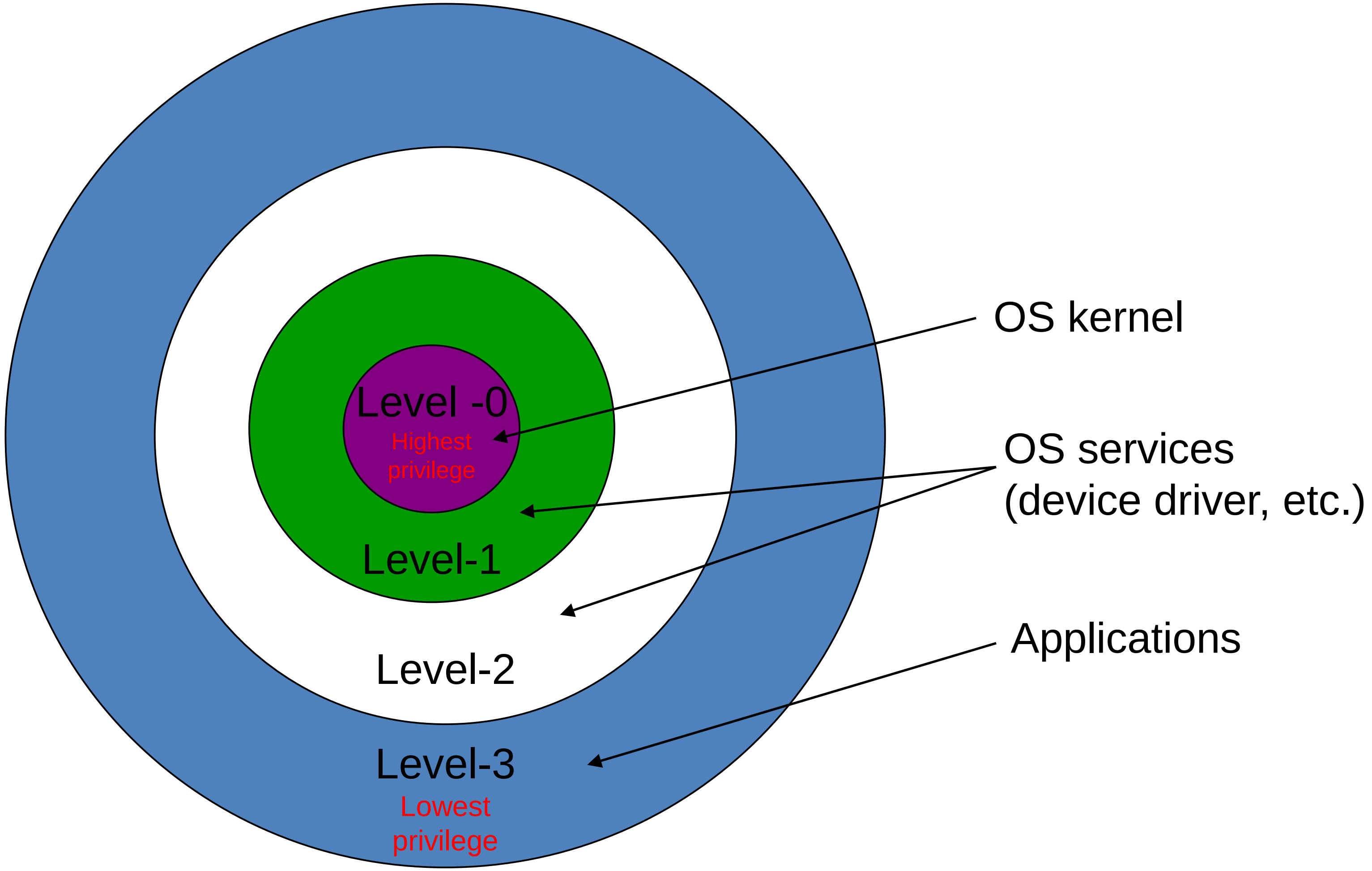
Full Virtualization -- challenge

Security issues -- Interception
Simulation of privileged operations -- I/O instructions
The effects of every operation performed within a given virtual machine must be kept within that virtual machine – virtual operations cannot be allowed to alter the state of any other virtual machine, the control program, or the hardware.
Some machine instructions can be executed directly by the hardware,

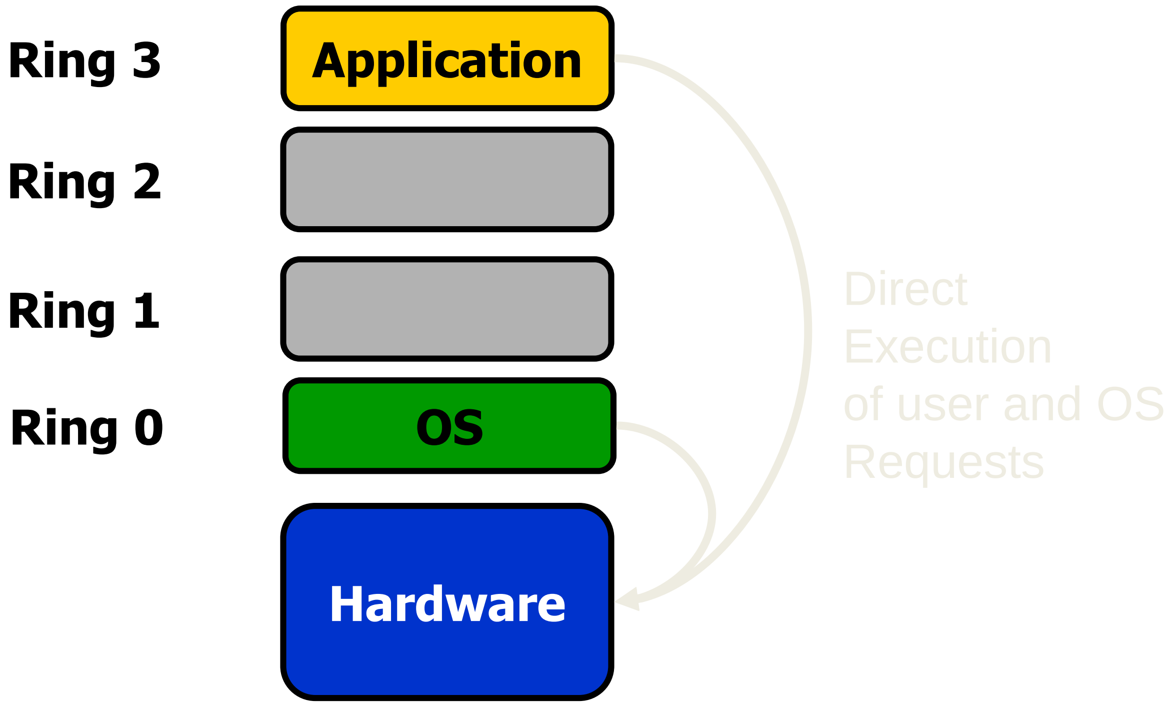
E.g., memory locations and arithmetic registers.

But other instructions that would "pierce the virtual machine" cannot be allowed to execute directly; they must instead be trapped and simulated. Such instructions either access or affect state information that is outside the virtual machine.
Some hardware is not easy to be used for full virtualization, e.g., x86

Restrict on Intel IA32 Protection Rings



The challenges of x86 hardware virtualization



The Problems and the Solutions

Originally designed for "personal use" (PC)

Security problems caused by Interception and privileged operations becomes critical

Solutions to Full virtualization of x86 CPU

- Full description of operations of all x86 hardware (but they evolve)
- Binary translation (almost established)
- OS-assisted (or paravirtualization)
- Hardware-assisted (future direction)

Definitions

Host Operating System:

The operating system actually running on the hardware

Together with *virtualization layer*, it simulates environment for …

Guest Operating System:

The operating system running in the simulated environment

I.e., the one we are trying to isolate

OS assisted (Paravirtualization)

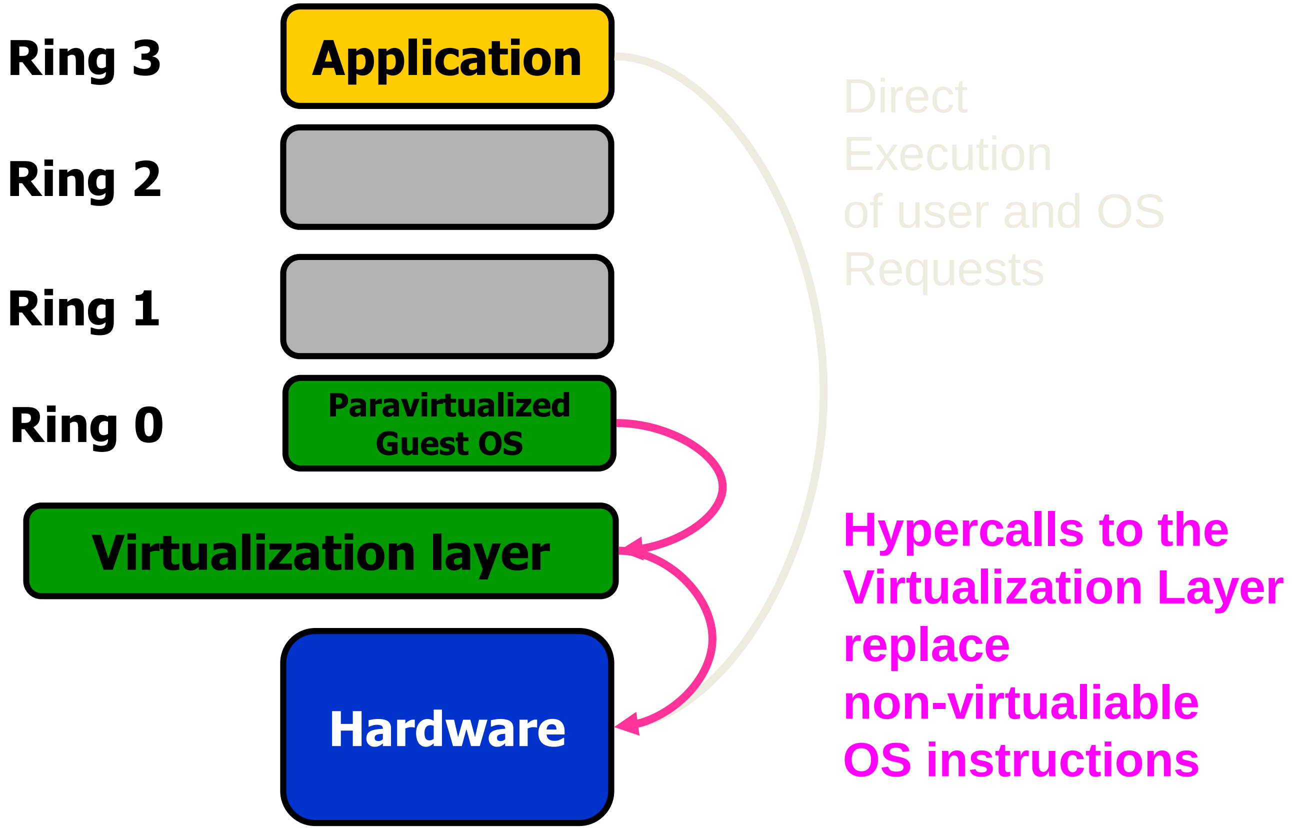
Paravirtualization – via an modified OS kernel as guest OS

- It is very difficult to build the more sophisticated binary translation support necessary for full virtualization.
- Paravirtualization involves modifying the OS kernel to replace non-virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor.
- The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling and time keeping.
- Paravirtualization is different from full virtualization, where the unmodified OS does not know it is virtualized and sensitive OS calls are trapped using binary translation.
- Paravirtualization cannot support unmodified OS

Example:

- Xen -- modified Linux kernel and a version of Windows XP

OS assisted (Paravirtualization)

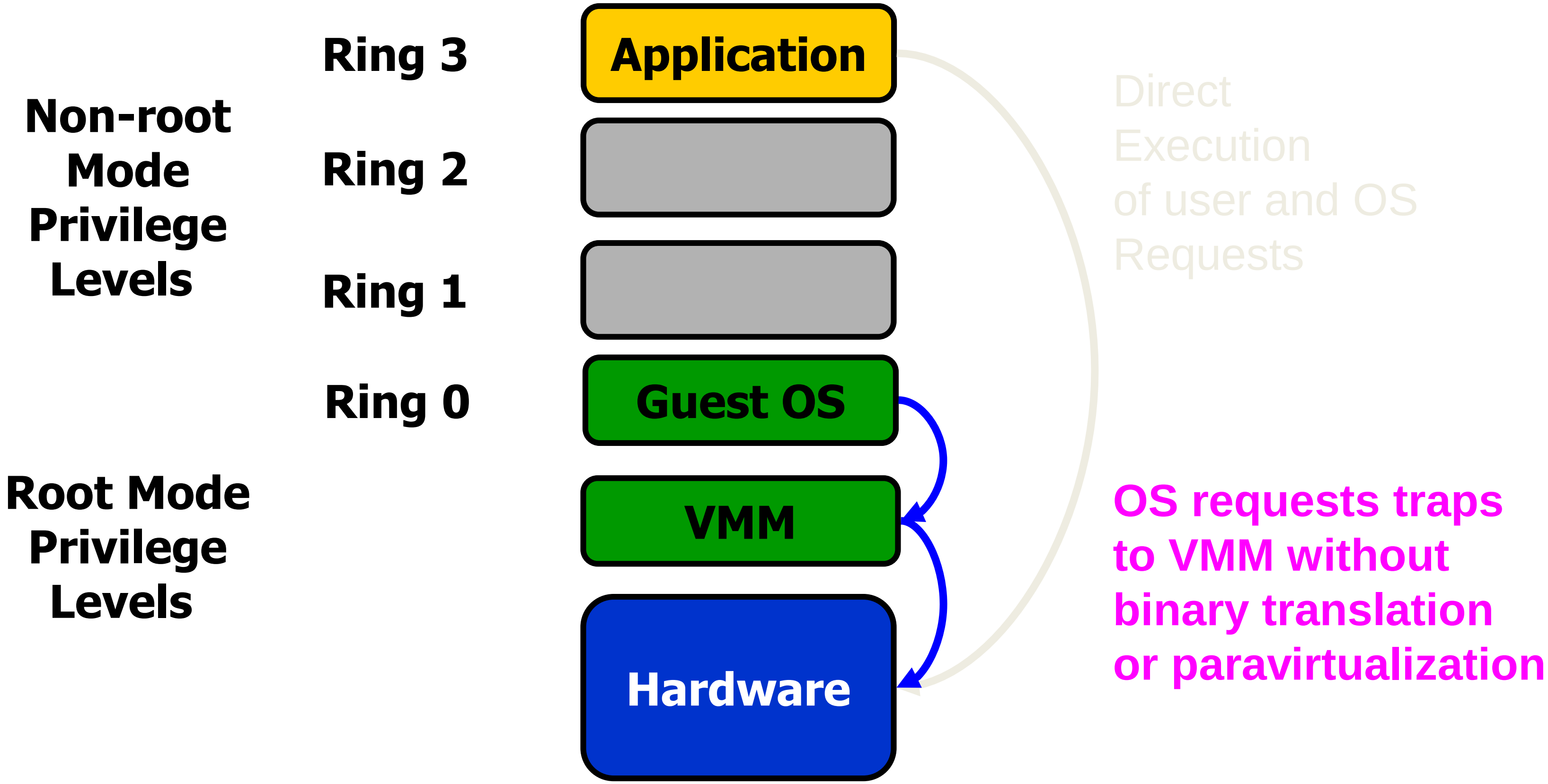


VMM: Virtual Machine Monitor

Hardware Assisted Virtualization

- Also known as accelerated virtualization, hardware virtual machine (Xen), native virtualization (Virtual iron).
- Hardware switch supported by CPU, e.g.
  - Intel Virtualization Technology (VT-x)
  - AMD's AMD-V
  - target privileged instructions with a new CPU execution mode feature that al lows the VMM to run in a new root mode below ring 0.
- Privileged and sensitive calls are set to automatically trap to the hypervisor, removing the need for either binary translation or paravirtualization.
- The guest state is stored in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Blocks (AMD-V).
- High hypervisor to guest transition overhead and a rigid programming model

Hardware Assisted Virtualization



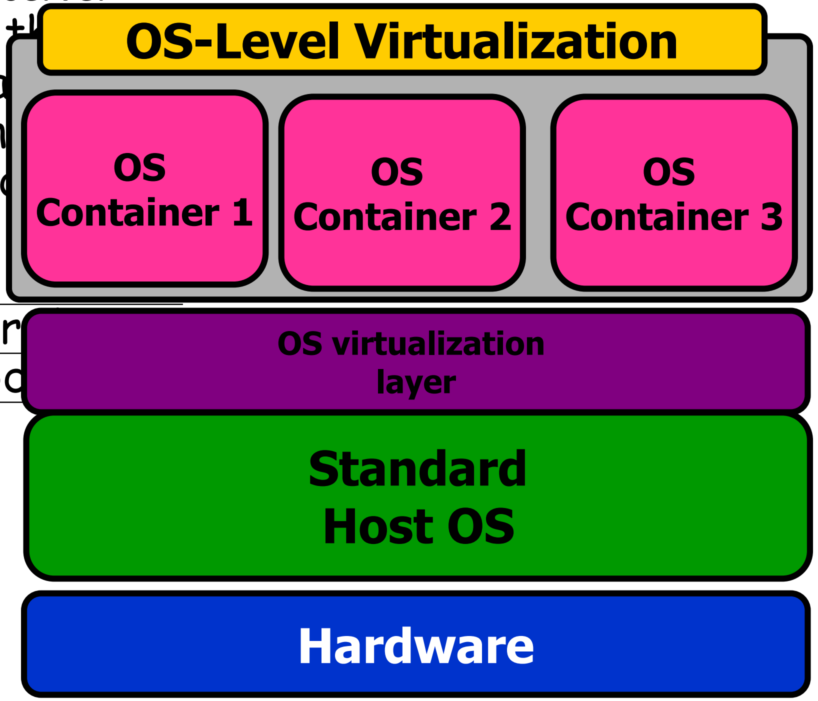
VMM: Virtual Machine Monitor

OS-Level Virtualization

OS-level virtualization

- kernel of an OS allows for multiple isolated user-space instances, instead of just one.
- Each OS instance looks and feels like a real server

OS virtualization virtualizes servers on th
system (kernel) layer. This creates isola
containers on a single physical server an
instance to utilize hardware, software, d
and management efforts with maximum
<u>OS-level virtualization implementations</u>
<u>capable of live migration can be used for</u>
<u>load balancing of containers between nod</u>
<u>cluster.</u>



Confusion…

OS-Level Virtualization. A type of server virtualization technology which works at the OS layer. The physical server and single instance of the operating system is virtualized into multiple isolated partitions, where each partition replicates a real server. The OS kernel will run a single operating system and provide that operating system functionality to each of the partitions.

Operating system virtualization refers to the use of software to allow system hardware to run multiple instances of different operating systems concurrently, allowing you to run different applications requiring different operating systems on one computer system. The operating systems do not interfere with each other or the various applications.

Example – Page tables

- Suppose *guest OS* has its own page tables Then *virtualization layer* must
  - Copy those tables to its own
  - Trap every reference or update to tables and simulate it
- During page fault
  - *Virtualization layer* must decide whether fault belongs to *guest OS* or self
  - If *guest OS*, must simulate a page fault
- Likewise, *virtualization layer* must trap and simulate *every* privileged instruction in machine!

Virtual Machines

- Some hardware architectures or features are impossible to *virtualize*
  - Certain registers or state not exposed
  - Unusual devices and device control
  - Clocks, time, and real-time behavior
- Solution – drivers or tools in guest OS
  - *VMware Tools*
  - *Xen* configuration options in Linux build

Snapshots & Migration

Snapshot: freeze a copy of virtual machine

Identify all pages in disk files, VM memory

Use copy-on-write for any subsequent modifications

To revert, throw away the copy-on-write pages

Migration: move a VM to another host

Take snapshot (fast)

Copy all pages of snapshot (not so fast)

Copy modified pages (fast)

Freeze virtual machine and copy VM memory

- Very fast, fractions of a second

Cloning

Simple clone:

Freeze virtual machine
Copy all files implementing it
Use copy-on-write to speed up

Linked clone:

Take snapshot
Original and each clone is a copy-on-write version of snapshot

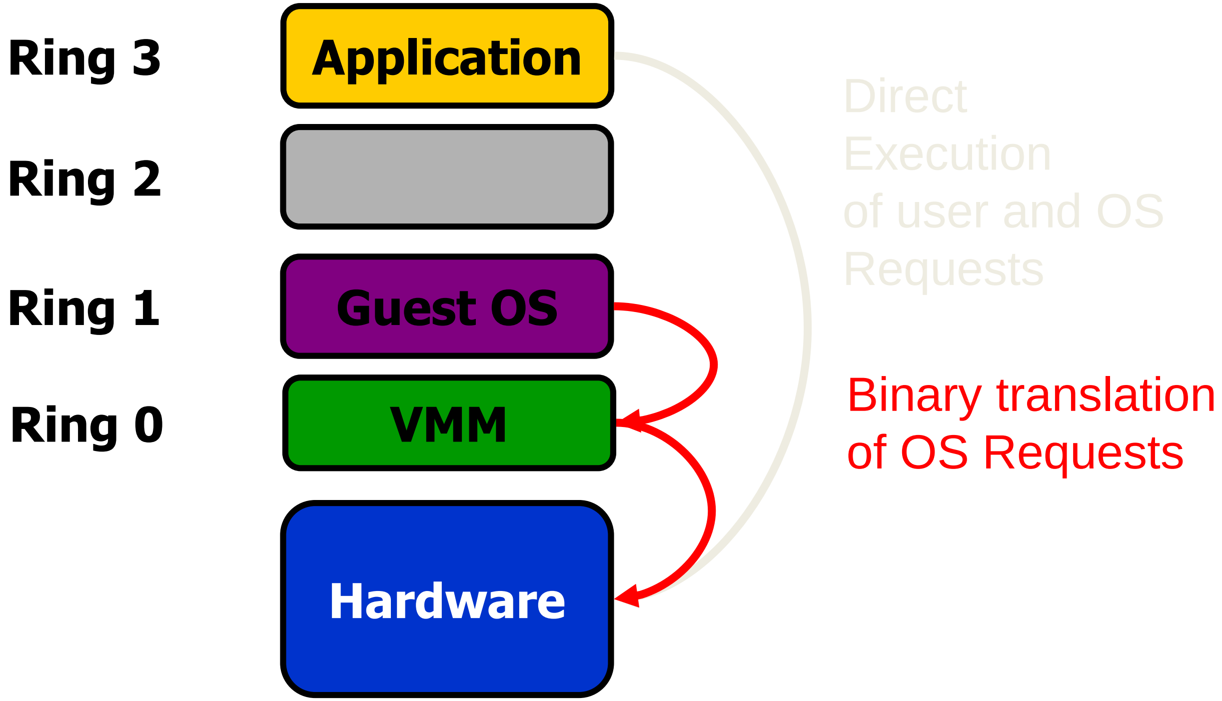
Binary translation

Kernel code of non-virtualizable instructions are translated to replace with new sequences of instructions that have the intended effect on the virtual hardware. Each virtual machine monitor provides each Virtual Machine with all the services of the physical system, including a virtual BIOS, virtual devices and virtualized memory management.
This combination of binary translation and direct execution provides Full Virtualization as the guest OS is fully abstracted (completely decoupled) from the underlying hardware by the virtualization layer. The guest OS is not aware it is being virtualized and requires no modification.

The hypervisor translates all operating system instructions on the fly and caches the results for future use, while user level instructions run unmodified at native speed.
Examples

- VMware
- Microsoft Virtual Server

Binary translation



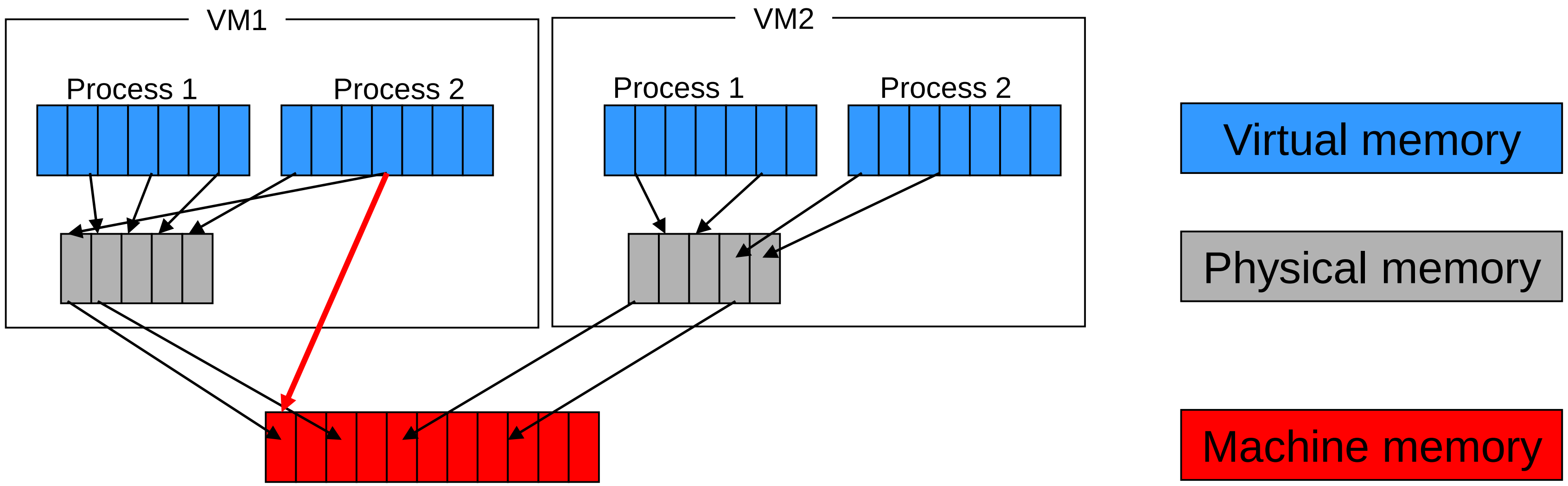
VMM: Virtual Machine Monitor

Application virtualization

- Application runs on
  - Different OS, platform, etc.
  - Same OS, different version/framework
  - Encapsulation of OS/platform
  - Improve portability, manageability and compatibility of applications
- A fully virtualized application is not installed in the traditional sense, although it is still executed as if it is (runtime virtualization)
- Full application virtualization requires a virtualization layer.

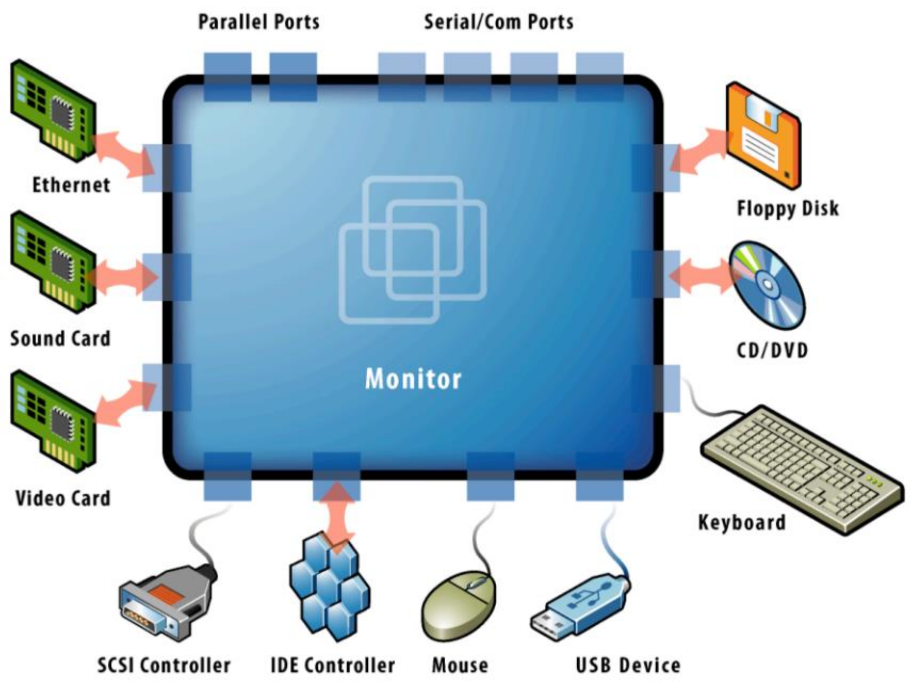
Memory Virtualization

Not only virtual memory
Hardware support
e.g., x86 MMU and TLB
To run multiple virtual machines on a single system, another level of memory virtualization is required.
The VMM is responsible for mapping guest physical memory to the actual machine memory, and it uses shadow page tables to accelerate the mappings.



Device and I/O Virtualization

VMM supports all device/IO drivers
Physically/virtually existed



Source: VMware white paper, “Understanding Full Virtualization, Paravirtualization, and Hardware Assist”

Techniques for X86 virtualization

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	yes	yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

Source: VMware white paper, “Understanding Full Virtualization, Paravirtualization, and Hardware Assist”

Virtualization

Binary translation is the most established technology for full virtualization

Hardware assist is the future of virtualization, but it still has a long way to go

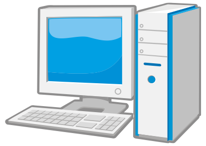
Paravirtualization delivers performance benefits with maintenance costs

- Xen
- VMWare

Issues in Virtualization for Cloud-Computing

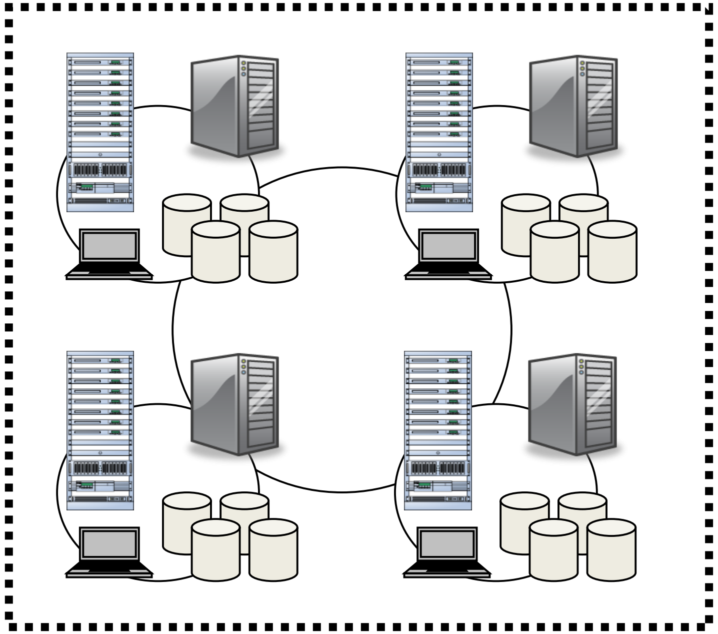
Aspects and expectation from

End-user
Operator/Manager



Virtualization

=



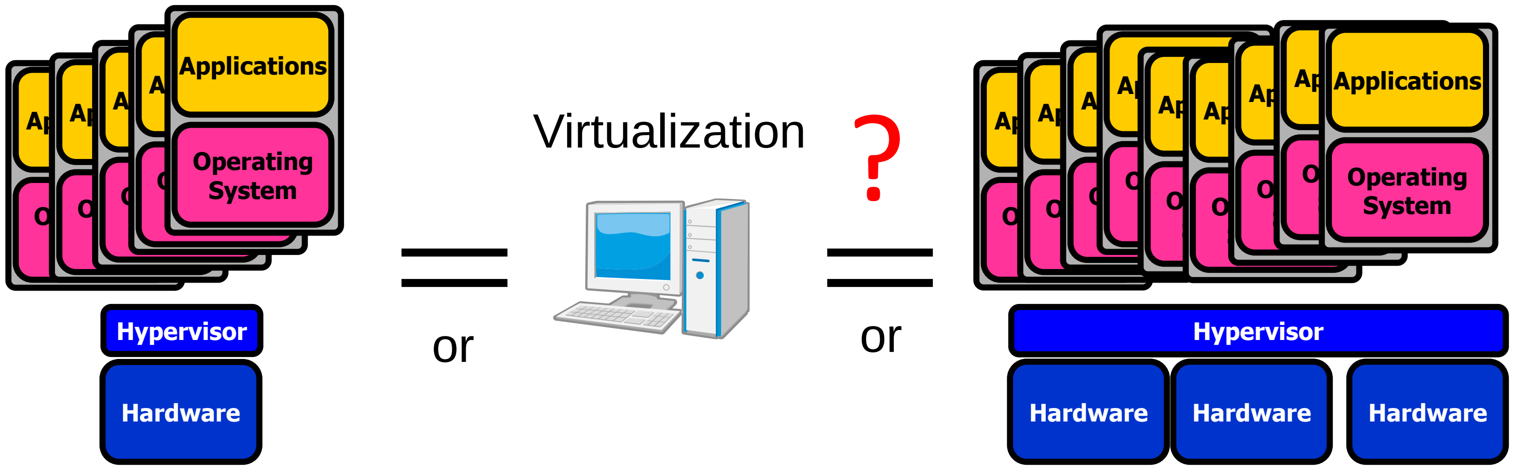
Issues in Virtualization for Cloud-Computing

Virtualization implemented on

- a single machine (with multi-core CPUs)
- a cluster of machines (with multi-core CPUs)

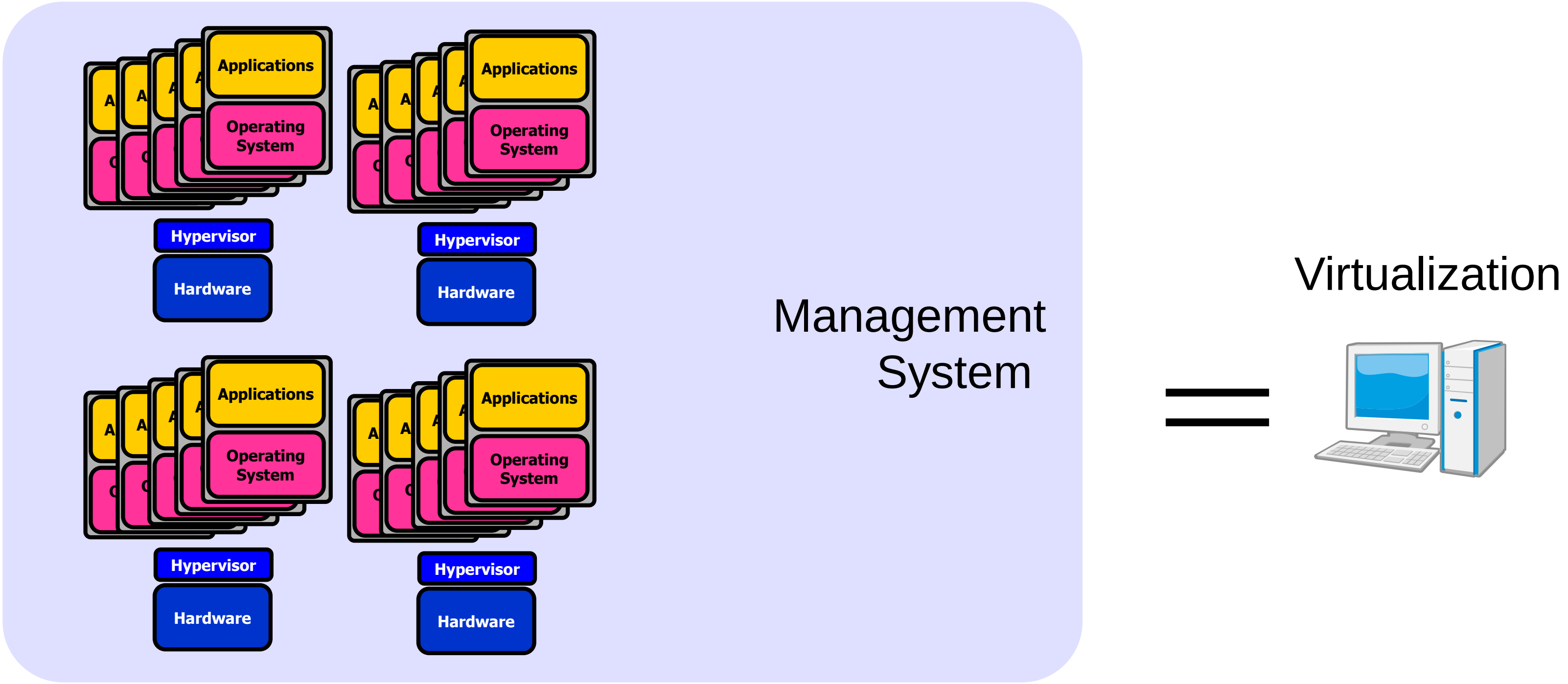
The state-of-the-art

- Running a Xen or a cluster of Xens



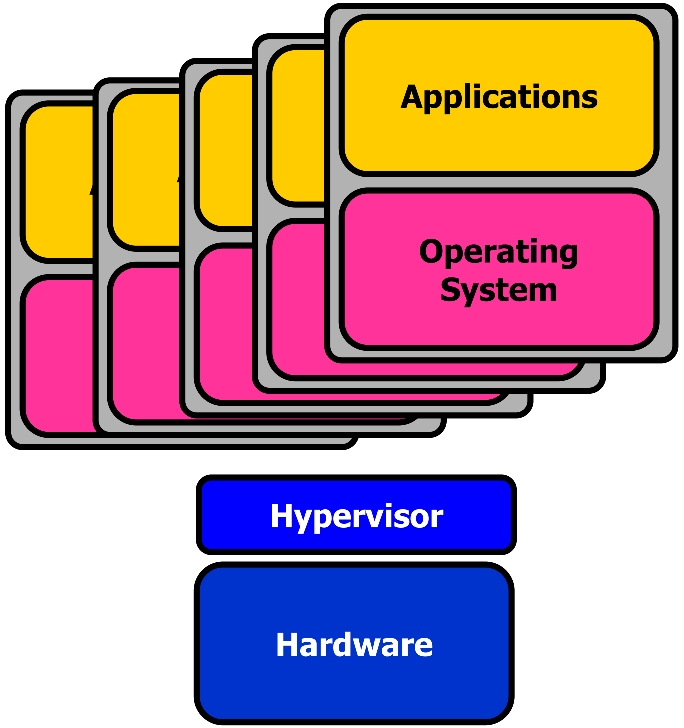
Issues in Virtualization for Cloud-Computing

Abiquo/abicloud may provide partial solutions



Running multiple OS and applications

Virtualization: One physical hardware can run multiple OS and applications through a hypervisor. A hypervisor is the virtualization manager on a physical hardware.



Popular hypervisors

Xen
KVM
QEMU
virtualBox
VMWare
Xen is the selected hypervisor of the project.

VMware – Modern Virtual Machine System

Founded 1998, Mendel Rosenblum *et al.*

Research at Stanford University

VMware Workstation

Separates *Host OS* from *virtualization layer*

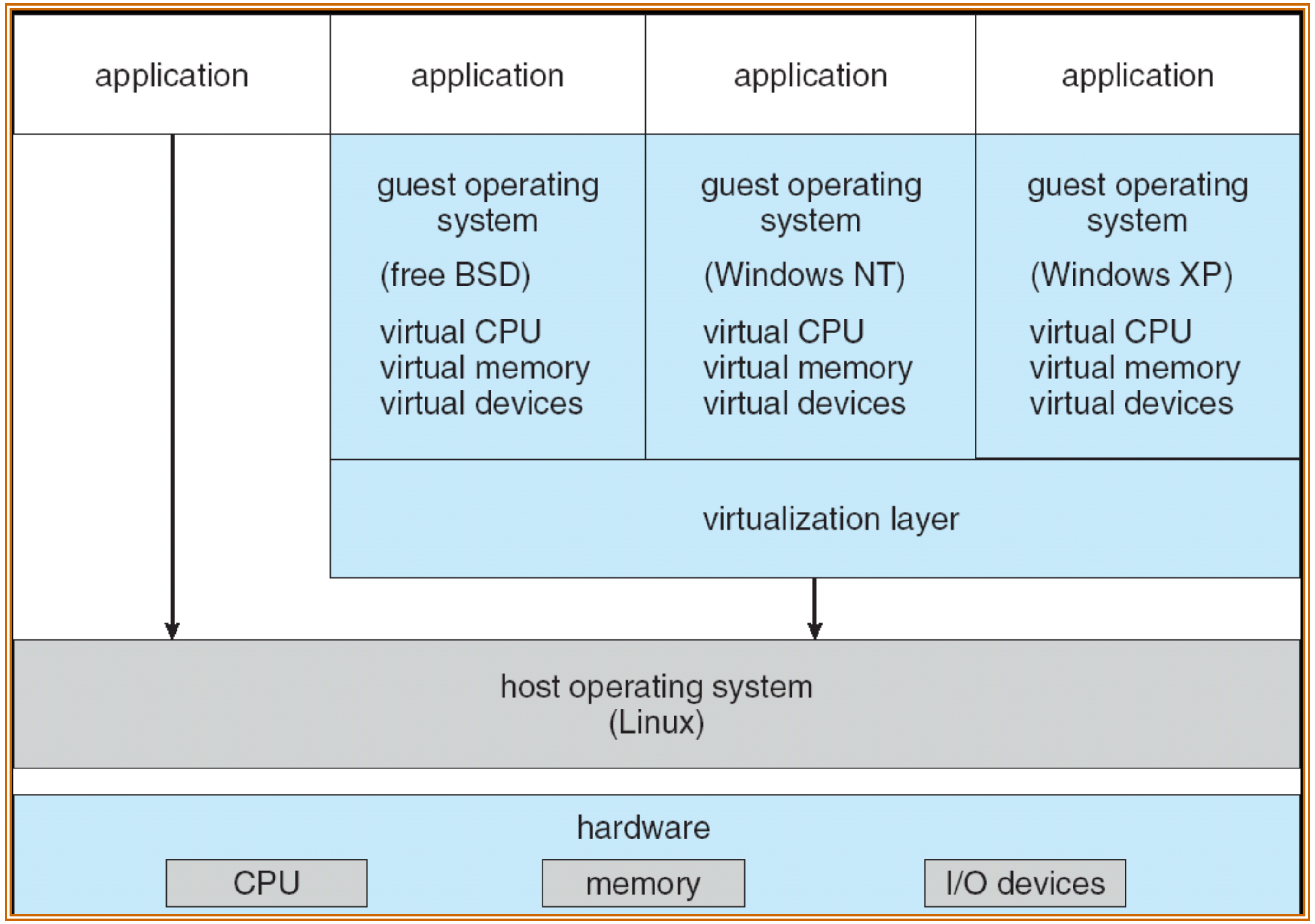
Host OS may be Windows, Linux, etc.

Wide variety of Guest operating systems

< $200

VMware Player is a free, stripped-down version of *VMware Workstation*

VMware Architecture



VMware Server

- Free version released in 2006
  - http://www.vmware.com/products/server/
  - Runs on any x86 server hardware and OS
  - Windows Server and Linux Host OS’s
- Partition a physical server into multiple virtual server machines
  - Target market – IT centers providing multiple services
  - Allows separate virtual servers to be separately configured for separate IT applications
    - *Provisioning*
  - Portability, replication, etc.

VMware Server ESX

- Total decoupling between hardware and applications
- High-end, high-performance IT applications
  - Oracle, SQL Server, Microsoft Exchange server, SAP, Siebel, Lotus Notes, BEA WebLogic, Apache
- Dynamically move *running* application to different hardware
  - Maintenance, hardware replacement
  - Provisioning new versions, etc.

Xen — Public Domain Virtualization Project

Cambridge University

- http://www.cl.cam.ac.uk/research/srg/netos/xen/

Philosophy – Adapt *Guest OS* to virtualization layer

- See configuration options of Suse Linux kernel

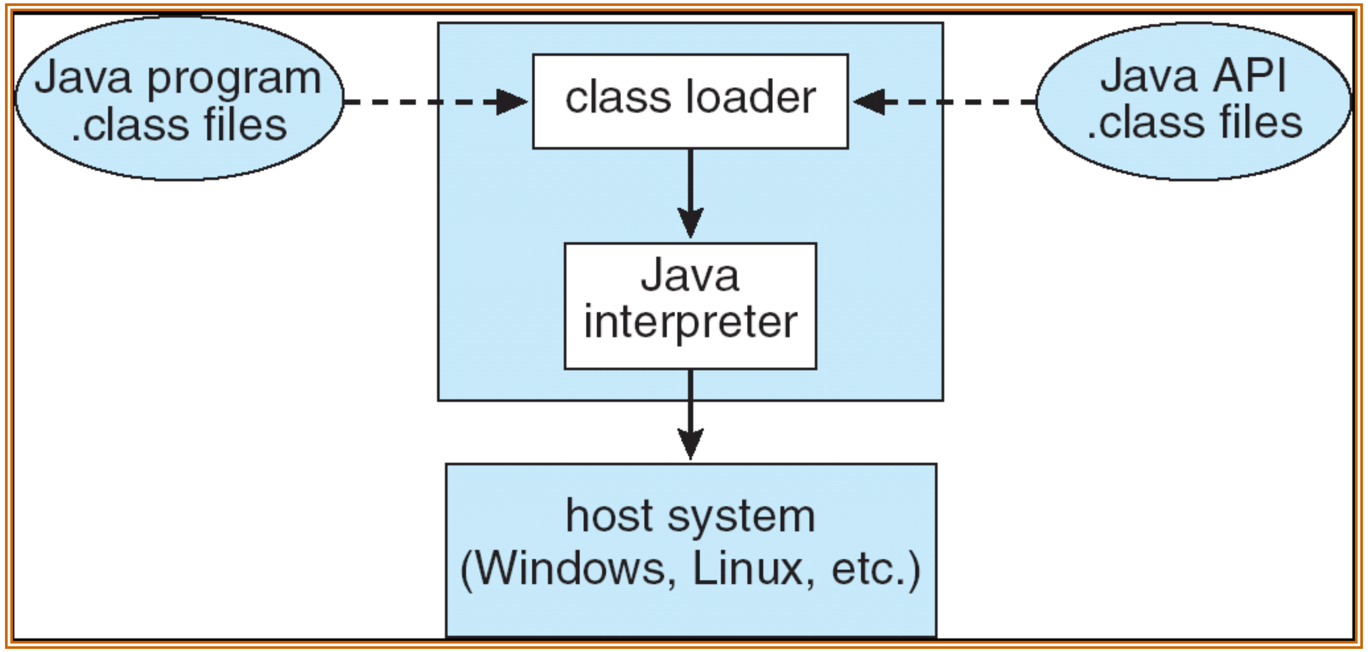
Must virtual machine be replica of host machine?

No, *virtualization layer* can simulate *any* architecture

Typically used for debugging specialized systems
Real-time systems, niche products, etc.

Guest architecture does not even have to be real hardware!

The Java Virtual Machine



Own idealized architecture

Stylized machine language

Byte codes

Readily available interpreter

Steps to use Xen

Connect to a Xen host (i.e., a physical hardware + Xen + Dom0 OS) via ssh.
Use xen-tools to create (xen-create-image), list (xen-list-images) and delete (xen-delete-image) images of virtual machines.
Use the xm tool to manage (create, list and shutdown) DomU guests.

Issues related to clouds with Xen

Xen-tools and xm are great for a single machine, but …
Today's private or public clouds often include hundreds or thousands of machines.
How to manage the cloud effectively and efficiently becomes a central issue in cloud computing.

Objectives of managing clouds

Easy-to-use client interface
Effective and efficient management of cloud infrastructure
Scalable deployment
Robust performance
Other nice characteristics associated with information systems management

Issues in Virtualization for Cloud-Computing

- Software deployment
  - Open-source
  - Commercial products
  - Re-installation or not
- Compatibility
  - Legacy software/database
- Copyright patent problem
  - Full virtualization
    - Hardware ISA?
  - Paravirtualization
    - Modifiable OS?
- Hardware assisted virtualization
  - Problem model
  - Re-write

Conclusion and Take-home Message

- We need to build sufficient computing resources that are isolated, always-available, shared, and reconfigurable
- Virtualization is a concept and method to suit the adaptivity
- We have discussed many classic virtualization techniques and tools
- Next talk: Services

Reference

VMWare ®
IBM ®
Miscrosoft®
Intel ®
AMD ®
http://www.xen.org/
http://en.wikipedia.org/
http://www.parallels.com/
http://www.webopedia.com/

References

Practical Virtualization Solutions: Virtualization from the Trenches by K. Hess and A. Newman, Prentice-Hall Inc., 2009, ISBN 13: 978-0-13-714297-2.

P. Li. *Selecting and using virtualization solutions – our experiences with VMware and Virtualbox*, CCSC 2009, vol.25, no.3, Jan 2010, pp.11-17.

References

http://www.vmware.com/pdf/virtualization.pdf

NoHype: Virtualized Cloud Infrastructure without the Virtualization. E. Keller, J. Szefer, J. Rexford, R. Lee. ISCA 2010.

Secure Virtual Machine Execution under an Untrusted Management OS. C. Li, A. Raghunathan, N.K. Jha. IEEE CLOUD, 2010.

An Introduction to Virtualization and Cloud Technologies to Support Grid Computing. I.M. Lorente. EGEE08.

Acknowledgement

Some slides are based on papers in references and

- Operating System Concepts, 7th ed., by Silbershatz, Galvin, & Gagne
- Modern Operating Systems, 2nd ed., by Tanenbaum
- http://web.cs.wpi.edu/~cs502
- https://www.cs.purdue.edu/homes/bb/cloud/
- U of Buffalo cse487
- U of Florida eel6686