Operating Systems – Virtualization

Péter Györke

http://www.mit.bme.hu/~gyorke/

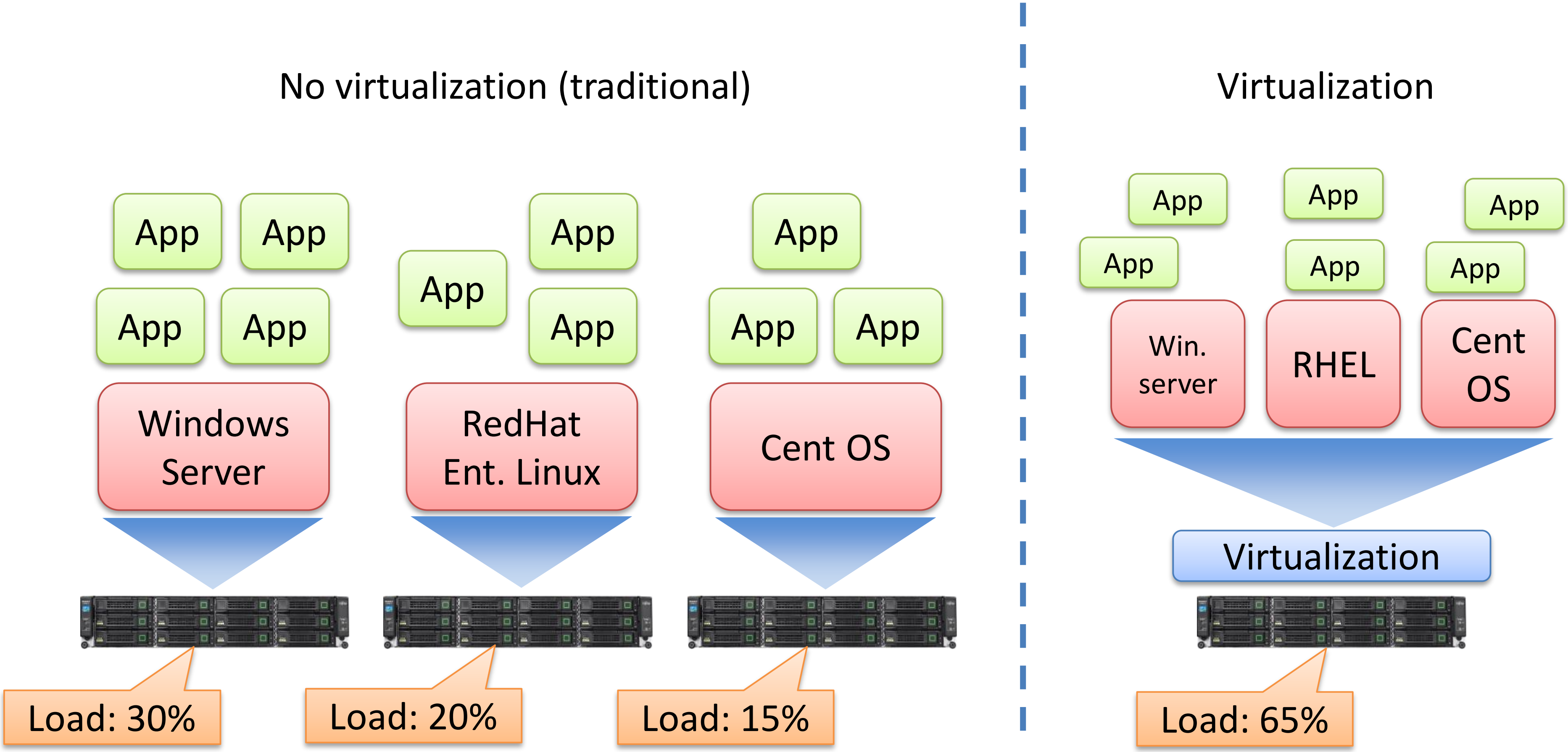
gyorke@mit.bme.hu

Budapest University of Technology and Economics (BME)
Department of Measurement and Information Systems (MIT)

The slides of the latest lecture will be on the course page. (https://www.mit.bme.hu/eng/oktatas/targyak/vimiab00)

Why use virtualization?

- Separate users/tasks/OS-s
  - Depends of the type of the virtualization
- Better utilization of the HW
- Better compatibility?

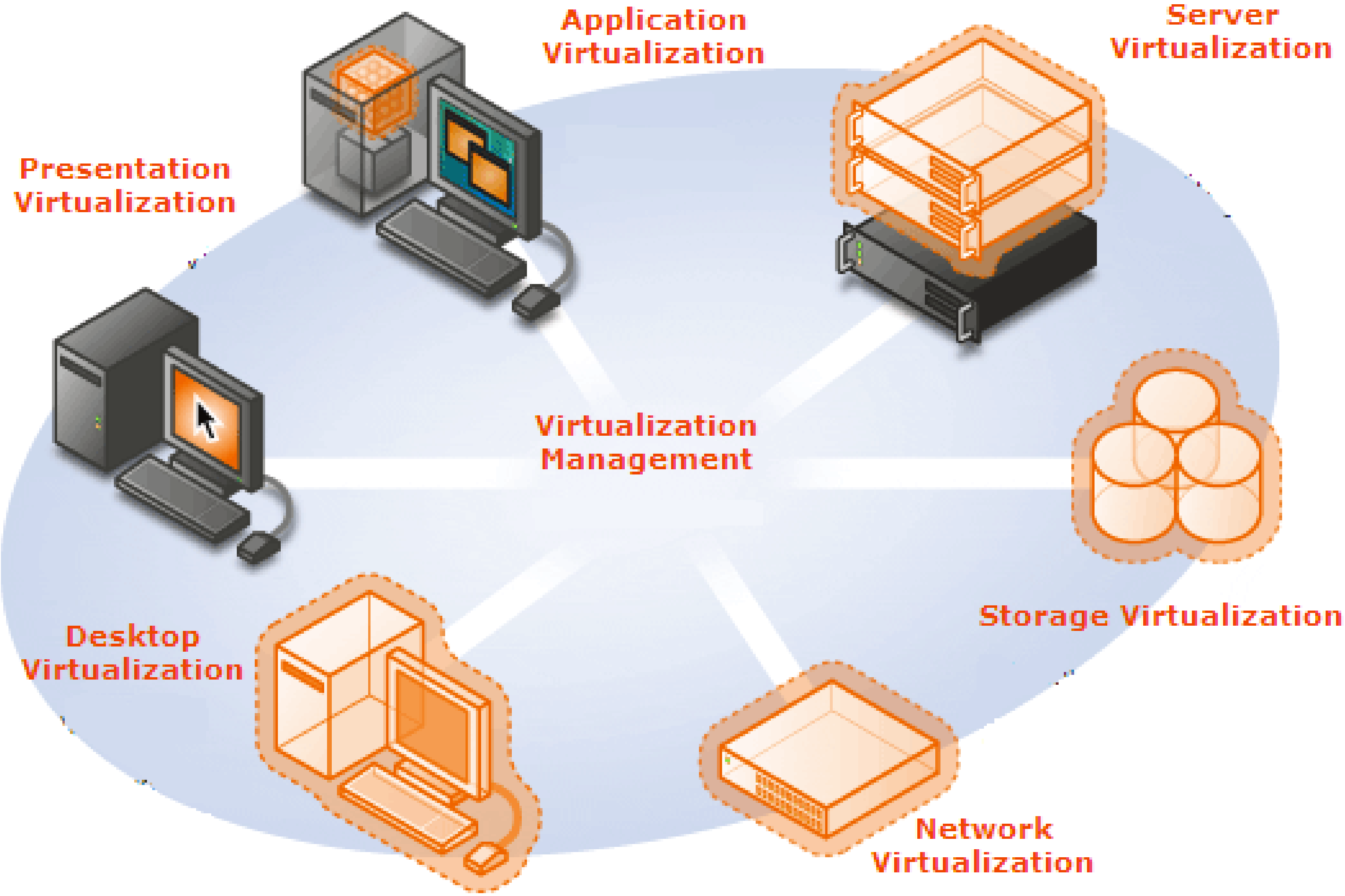


Types of virtualization

- Abstract virtual machine
  - virtual resources for the applications
  - Task separation
- Classic virtual machines
  - The HW components are shared between multiple OS-s, managed by the VMM (Virtual Machine Manager)
- Other (newer) concepts
  - OS level virtualization
    - Many users on the same OS, but they don't have to know about each other
    - Separate file systems, sytem libraries
    - Same kernel
    - E.g. Linux Containers
  - Application virtualization
    - Separate registry and file system for an application
    - More portable applications
  - Presentation virtualization
    - Remote monitor and input devices
    - Remote Desktop (RDP), VNC

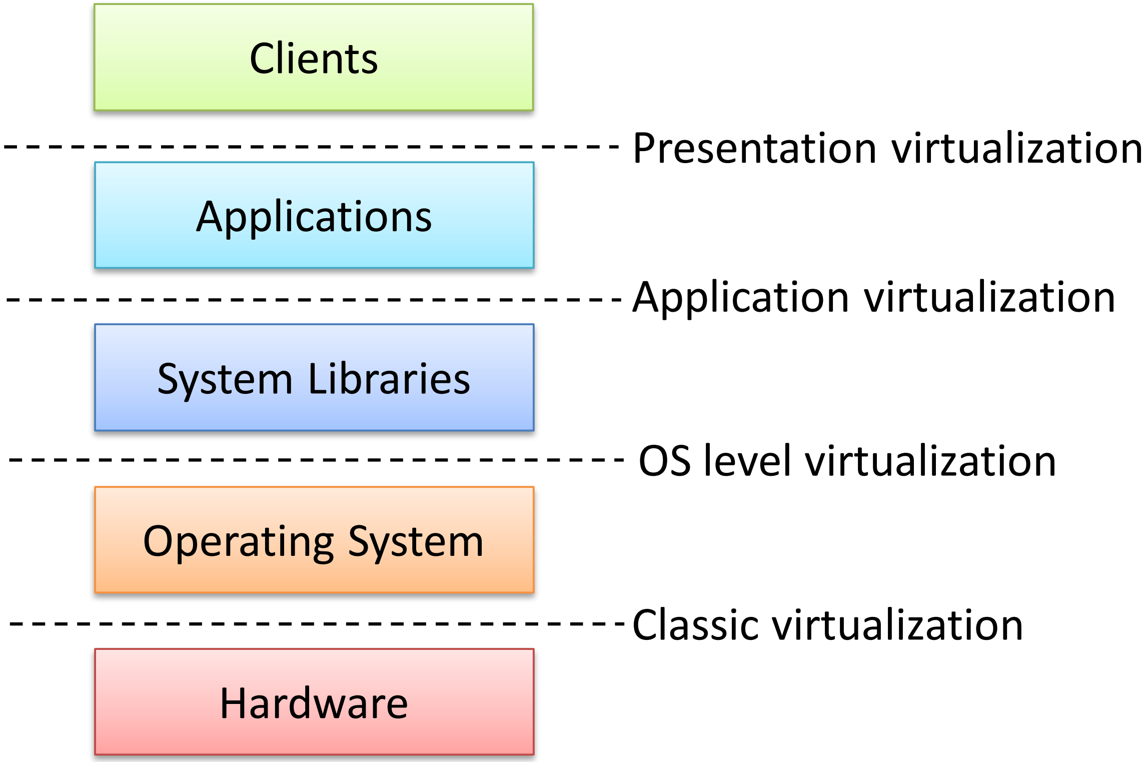
The types of virtualization in other words

- Different vendors use different terminology…



Types (levels) of virtualization

- Where we draw the line of separation?



Clients

Presentation virtualization

Applications

Application virtualization

System Libraries

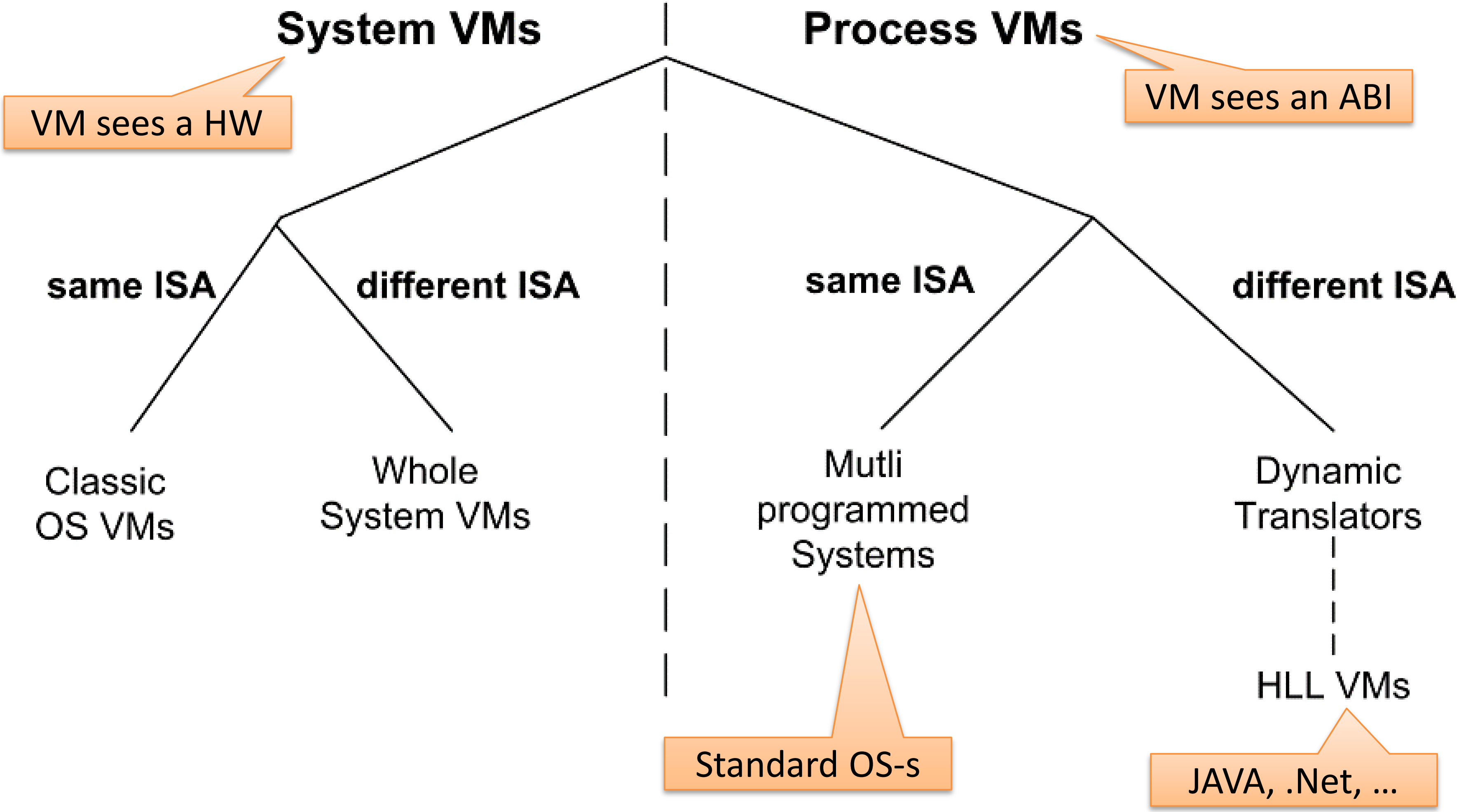
OS level virtualization

Operating System

Classic virtualization

Hardware

Virtual machine taxonomy*



System VMs

VM sees a HW

same ISA

Classic OS VMs

different ISA

Whole System VMs

Process VMs

VM sees an ABI

same ISA

Mutli programmed Systems

Standard OS-s

different ISA

Dynamic Translators

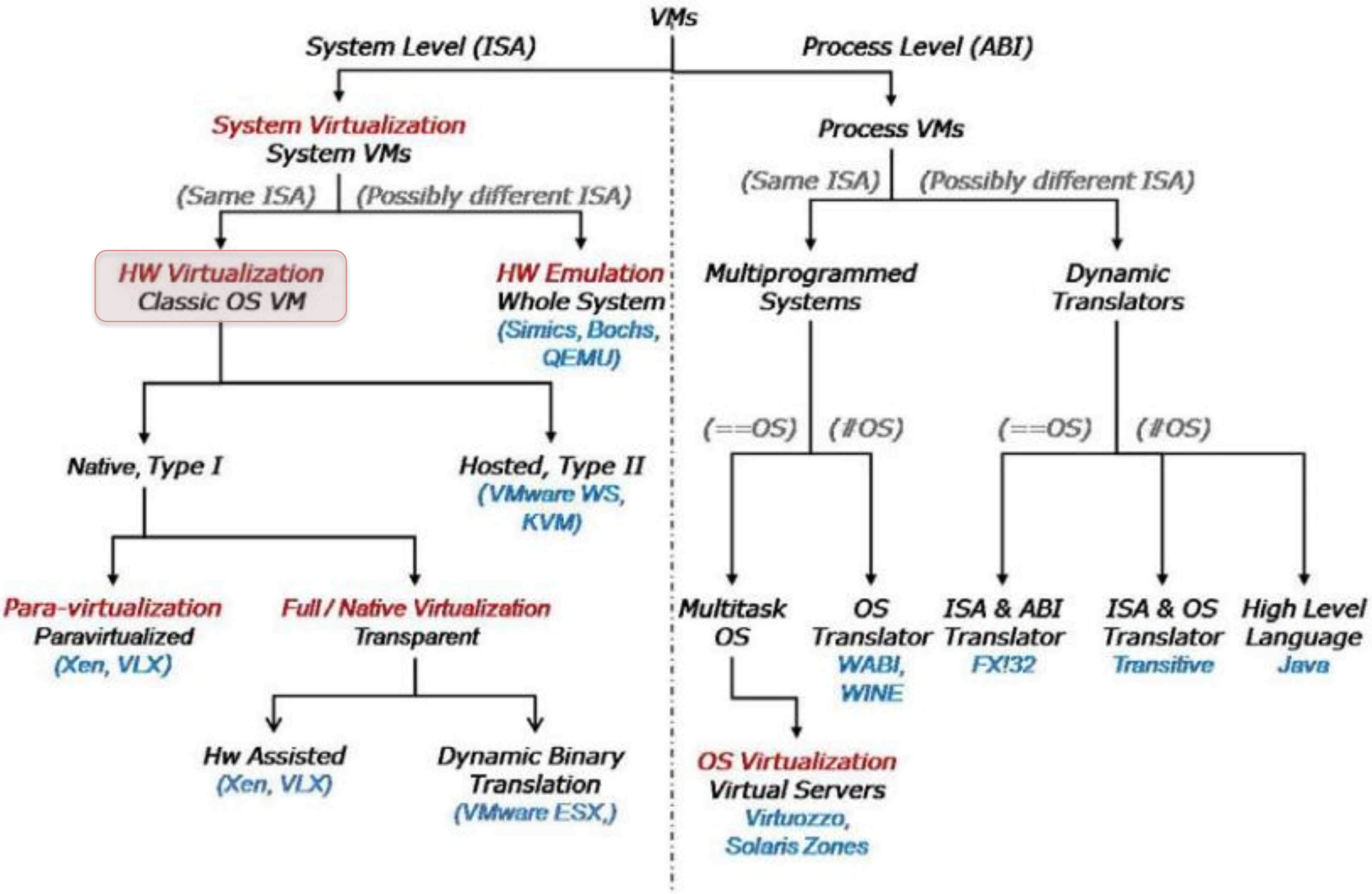
HLL VMs

JAVA, .Net, …

Source: J. Smith and Ravi Nair, “The architecture of virtual machines,” *IEEE Computer*, vol. 38, 2005, pp. 32-38.

* taxonomy ~ structure for presenting relationships between concepts

Virtual machine taxonomy detailed



VMs

System Level (ISA)

Process Level (ABI)

System Virtualization
System VMs

(Same ISA)

(Possibly different ISA)

HW Virtualization
Classic OS VM

HW Emulation
Whole System
(Simics, Bochs, QEMU)

Native, Type I

Hosted, Type II
(VMware WS, KVM)

Para-virtualization
Paravirtualized
(Xen, VLX)

Full / Native Virtualization
Transparent

Hw Assisted
(Xen, VLX)

Dynamic Binary Translation
(VMware ESX,)

Process VMs

(Same ISA)

(Possibly different ISA)

Multiprogrammed Systems

Dynamic Translators

(==OS)

(≠OS)

(==OS)

(≠OS)

Multitask OS

OS Translator
WABI, WINE

ISA & ABI Translator
FX!32

ISA & OS Translator
Transitive

High Level Language
Java

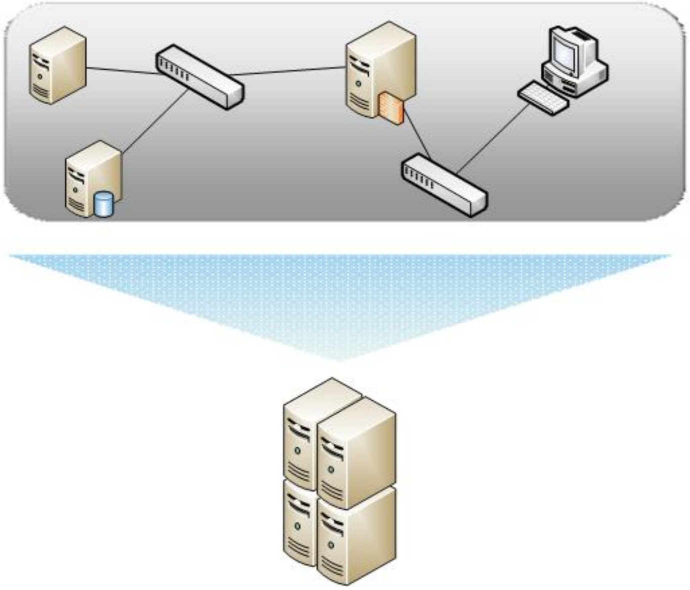
OS Virtualization
Virtual Servers
Virtuozzo, Solaris Zones

Suggested terminology

- **Platform** virtualization: virtualizing a full computer, running multiple OS on one hardware
  - Also known as: server, computer, hardware virtualization…
- Definitions:
  - **Host machine**: physical computer
  - **Guest machine**: virtual computer
  - **Virtual Machine Monitor** (VMM): program managing the virtual machines

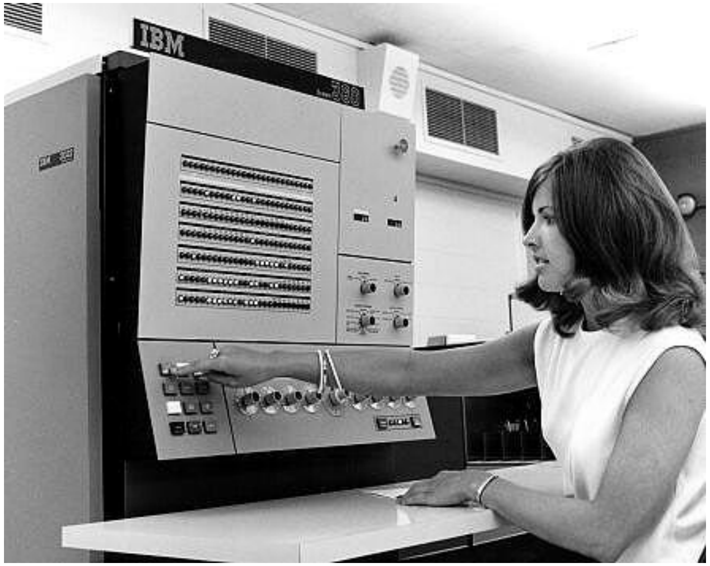
Why is platform virtualization good?

- Building test systems
  - Experimenting with other OS-s
  - Using a SW which is only runnable of a specific OS
- HW consolidation
- Legacy systems
  - Keeping them alive
- On-demand architectures
- High availability, disaster recovery
- Portable applications
- …



History of platform virtualization

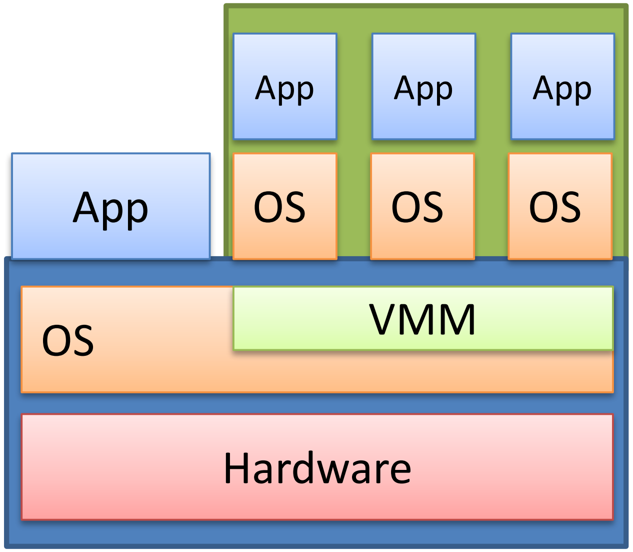
- ~1960 - IBM CP-40 system
  - in the mainframe products
- x86 virtualization
  - Seemed impossible
    - The instruction set wasn't prepared for virtualization
    - Only SW methods are possible → can be extremely slow
  - 1997: Stanford, Disco projects
  - 1998: VMware solution
  - 2000- Other solutions
- Now:
  - HW support
  - has its own business
  - becomes widely used
  - On the enterprise level, this is the common practice



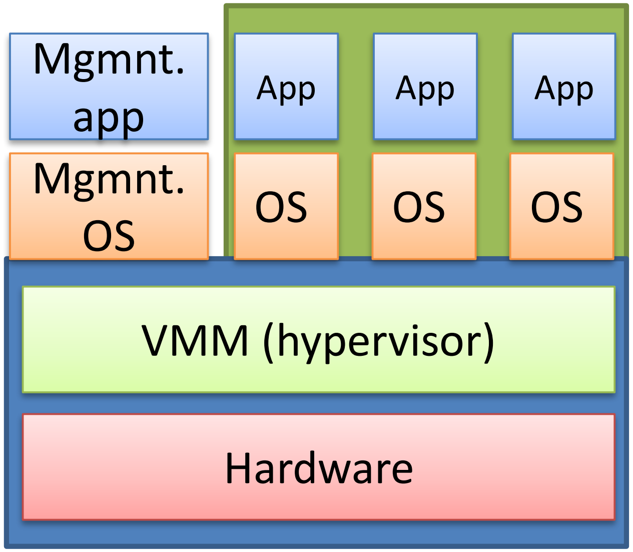
Platform virtualization

- Two main approaches

Hosted (VMM has kernel level parts)



Bare-metal (whole VMM runs at kernel level)



Mainly desktop products: VMware Workstation, Server, Player, Oracle VirtualBox, MS VirtualPC, KVM, UML

🟩 GUEST

🟦 HOST

Mainly server products: VMware ESX Server, Xen Enterprise, MS Hyper-V

Requirements and challenges

- Requirements for a virtualization solution:
  - **Equivalence**: programs in a VM should perform indistinguishable from running on the hardware
  - **Resource control**: the VMM should handle all the physical resources
  - **Efficiency**: most of the VM’s instructions should run directly on the hardware
- Challenges
  - The system have to be protected from the guest(s)
    - Not every operation is allowed
      - E.g.: HLT (Halt) instruction
    - Solution: the instructions must be monitored by the VMM
      - Privileged instructions should be handled differently – no direct execution

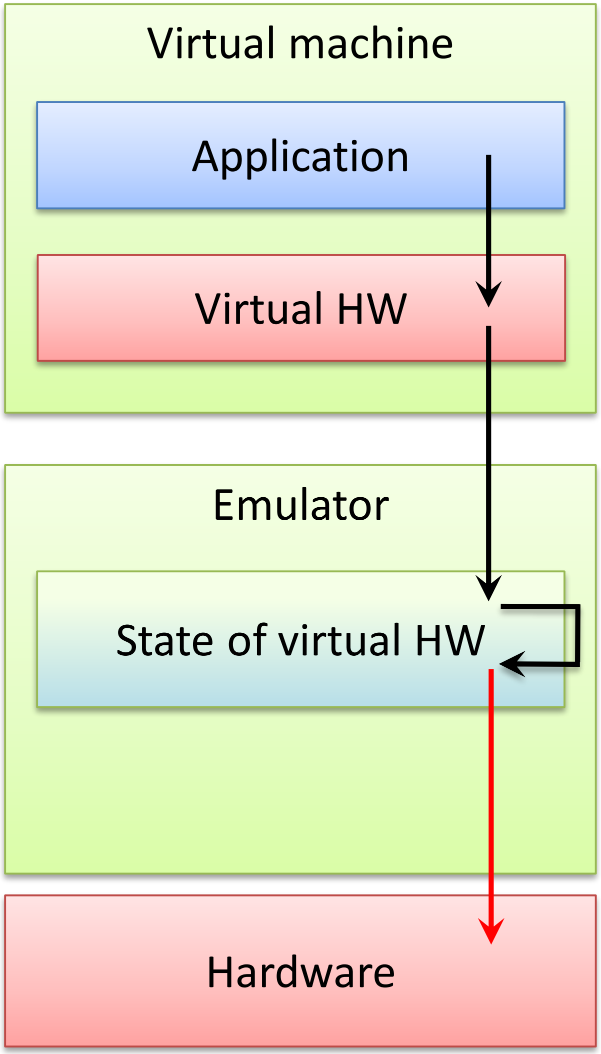
Gerald J. Popek, Robert P. Goldberg: Formal Requirements for Virtualizable Third Generation Architectures. Commun. ACM 17(7): 412-421 (1974)

Theory behind platform virtualization

- CPU virtualization
  - How to translate the instructions?
    - Every instruction is translated – emulation
    - Some instructions are translated, some executed directly
    - HW support?
    - Instruction privileges
- Memory virtualization
  - We have only 1 MMU
  - Context change between virtual machines has a high overhead
  - How to handle page tables and the TLB?
- I/O virtualization
  - How to manage a HW device? (e.g. a network adapter)
    - Use generic drivers
    - Use special virtual device drivers
    - Use special HW devices, which supports virtualization

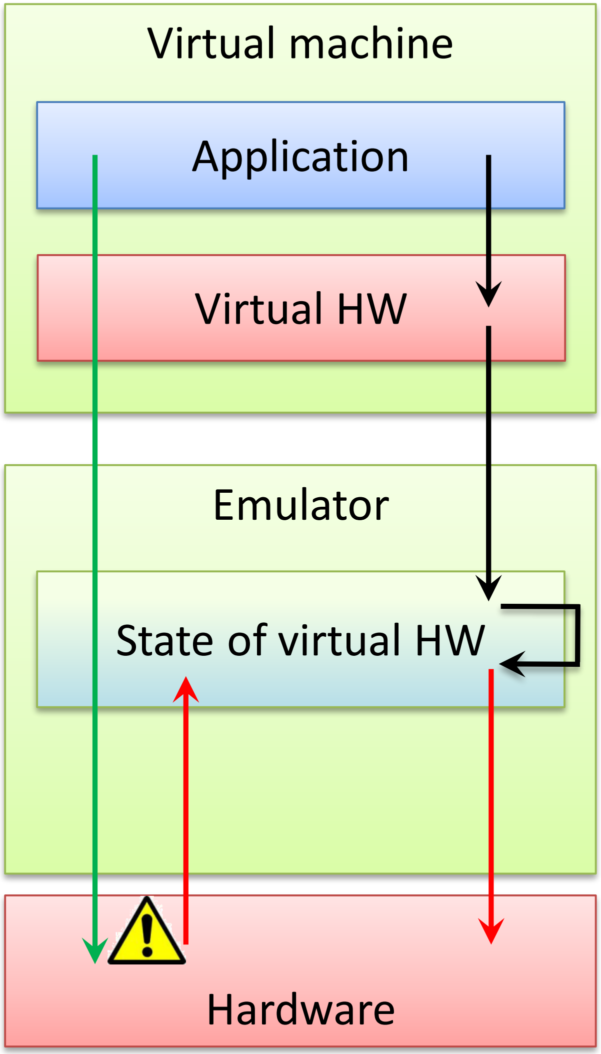
CPU virtualization – Full emulation

- The emulator
  - Stores the full state of the HW
  - Every instruction is inspected and translated, then executed
- Pro
  - Different CPU-s can be emulated
- Con
  - slow



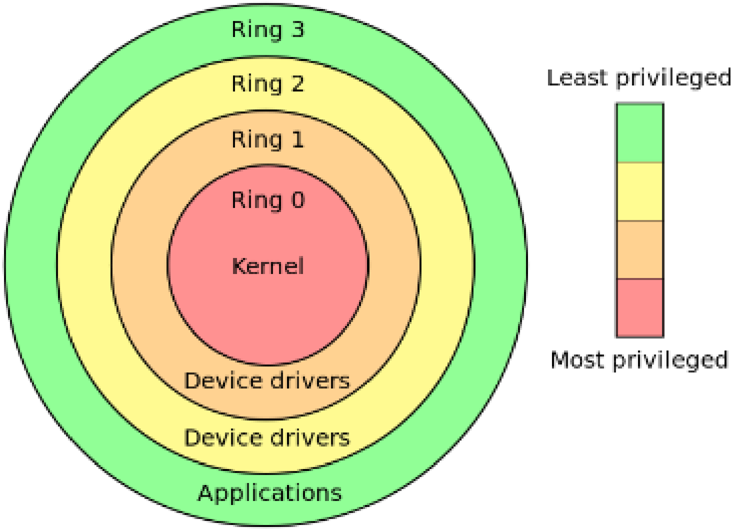
CPU virtualization – Trap and emulate

- Trap
  - HW exception handling, which resumes execution after the handler (VMM)
- HW support is required
  - Protection modes (x86 rings)
  - VM runs in a lower modes
  - Privileged instructions should case a trap when called from a non-privileged mode



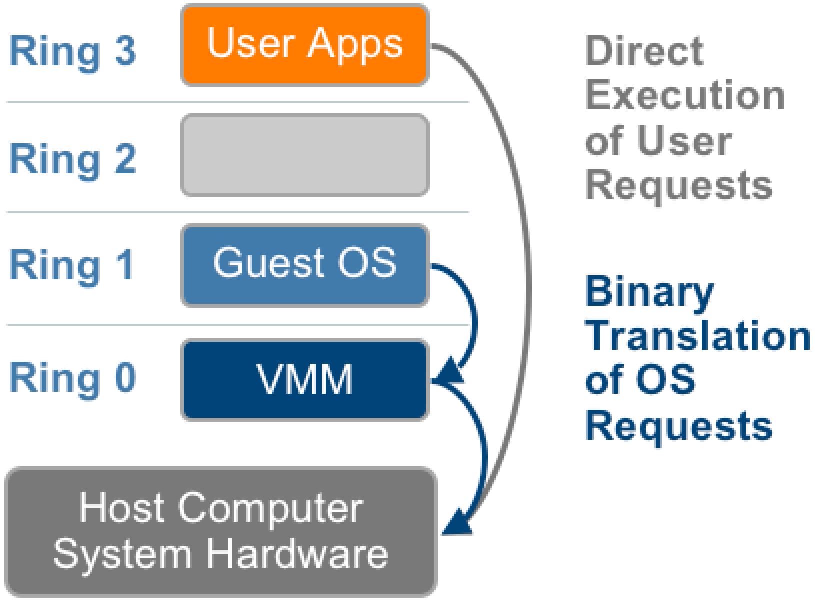
Issues with x86 virtualization

- Some architectures can be easily virtualized
  - **x86 cannot**
- From ~250 instructions 17 violate the classical requirements, e.g.
  - POPF instruction: modifies EFLAGS register
    - But if not executed in ring 0, doesn't throw an exception
- Privileged state can be detected
  - OS can detect whether it's running in a VM → violating the equvivalence requirement
- **Conclusion**: the trap & emulate method cannot be used on the original x86



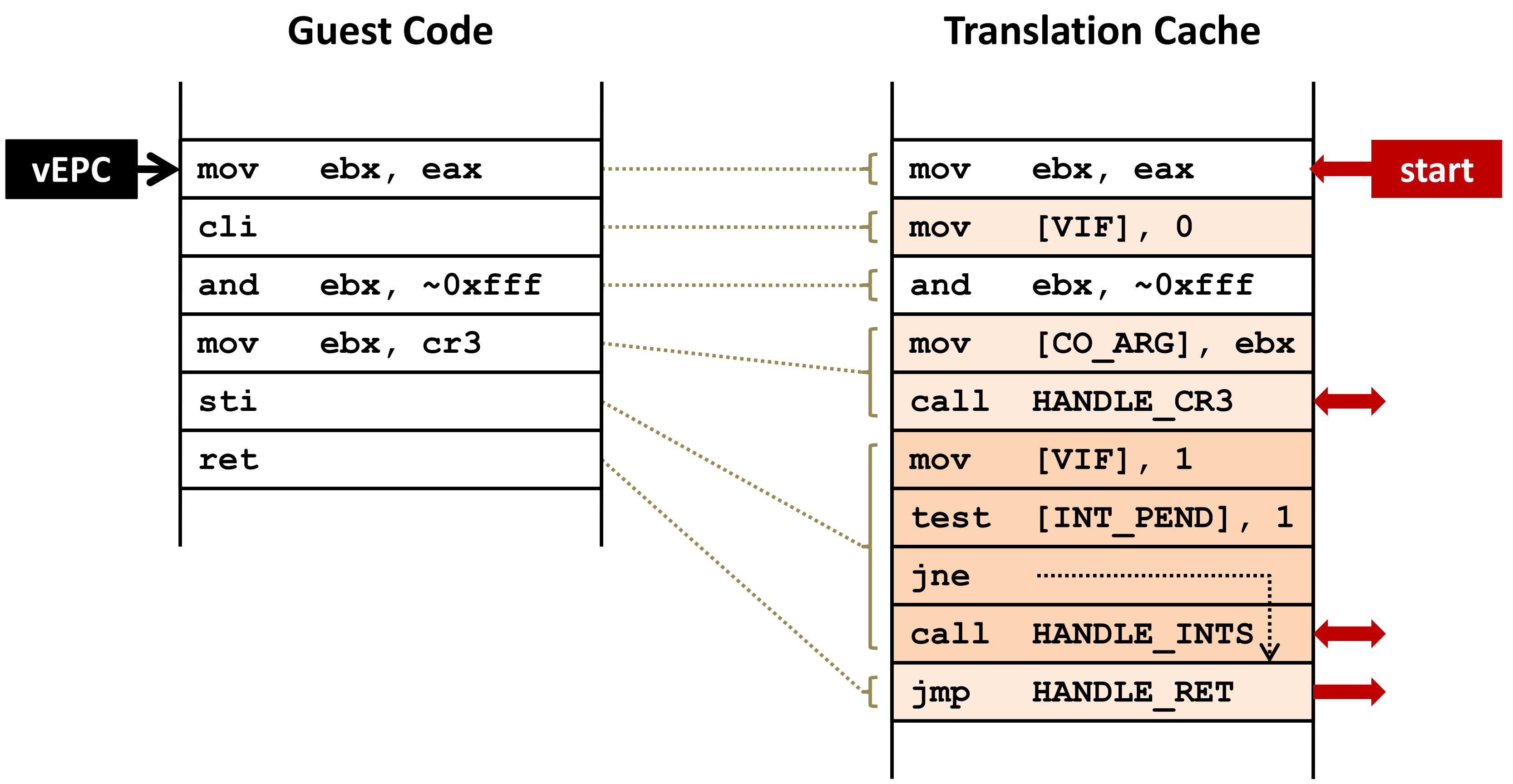
Solutions for x86 – binary translation

- Most of the instructions run directly
- The instructions are inspected in blocks
- Privileged instructions translated runtime
- Doesn’t need source code
- Caches translated code
- Guest OS not aware of virtualization


Ring 3
User Apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer System Hardware
Direct Execution of User Requests
Binary Translation of OS Requests

Source: VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assisted Virtualization http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

Binary translation – example



Guest Code:

```
mov   ebx, eax
cli
and   ebx, ~0xfff
mov   ebx, cr3
sti
ret
```

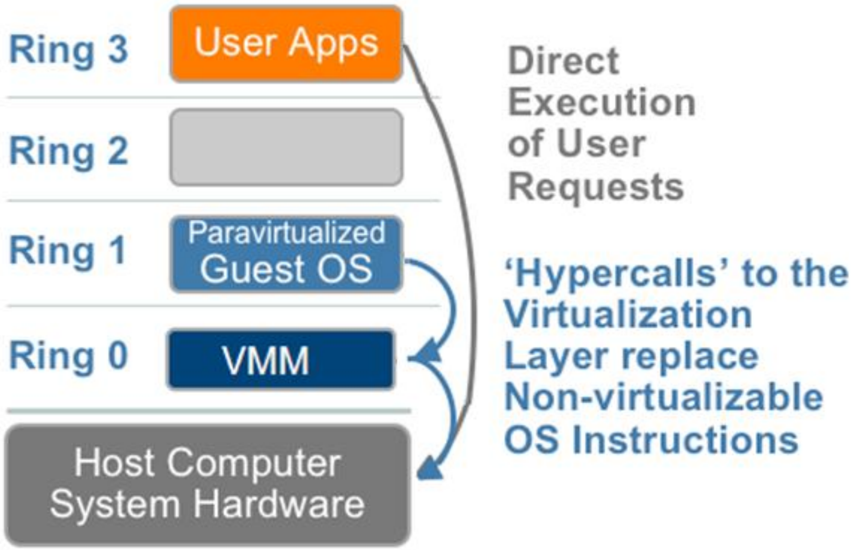
Translation Cache:

```
mov   ebx, eax
mov   [VIF], 0
and   ebx, ~0xfff
mov   [CO_ARG], ebx
call  HANDLE_CR3
mov   [VIF], 1
test  [INT_PEND], 1
jne
call  HANDLE_INTS
jmp   HANDLE_RET
```

Source: Carl Waldspurger, Introduction to Virtual Machines

Solutions for x86 – paravirtualization

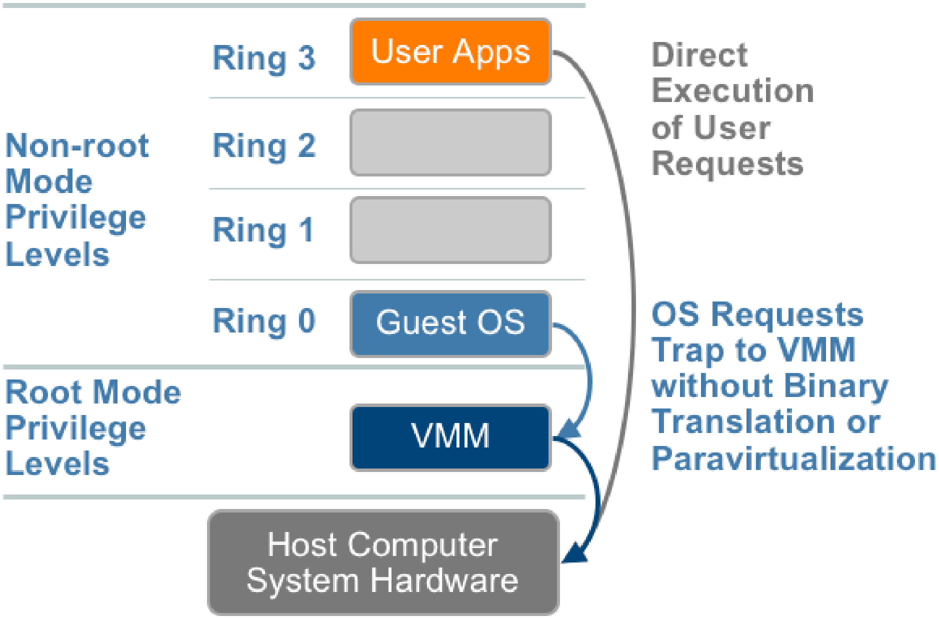
- Modifying the source of the guest OS
- Replacing “problematic” instructions
- Hypercall: calling the VMM directly


Ring 3
User Apps
Ring 2
Ring 1
Paravirtualized Guest OS
Ring 0
VMM
Host Computer System Hardware
Direct Execution of User Requests
‘Hypercalls’ to the Virtualization Layer replace Non-virtualizable OS Instructions

Source: VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assisted Virtualization http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

Solutions after x86 – hardware-assisted virtualization

- ~2005: Intel Virtualization Technology (VT-x) and AMD AMD-V
- HW support: root mode, VMCS
  - Instructions: VMCALL, VMLAUNCH
- Trap & emulate now works
- Backward compatibility with the x86 ring system


Ring 3
User Apps
Direct Execution of User Requests
Non-root Mode Privilege Levels
Ring 2
Ring 1
Ring 0
Guest OS
OS Requests Trap to VMM without Binary Translation or Paravirtualization
Root Mode Privilege Levels
VMM
Host Computer System Hardware

Source: VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assisted Virtualization http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

Comparison between CPU virtualization methods

- Which one is the best?
  - The answer changing constantly
    - Depends on the environment, workload
  - Most products mix several techniques

- Examples
  - 2006. VMware: [BT is better than HW assisted virtualization]
  - 2008. [VMware: Paravirtualization + BT is better than pure BT]
  - 2009. [Comparing Hardware Virtualization Performance Utilizing VMmark v1.1]

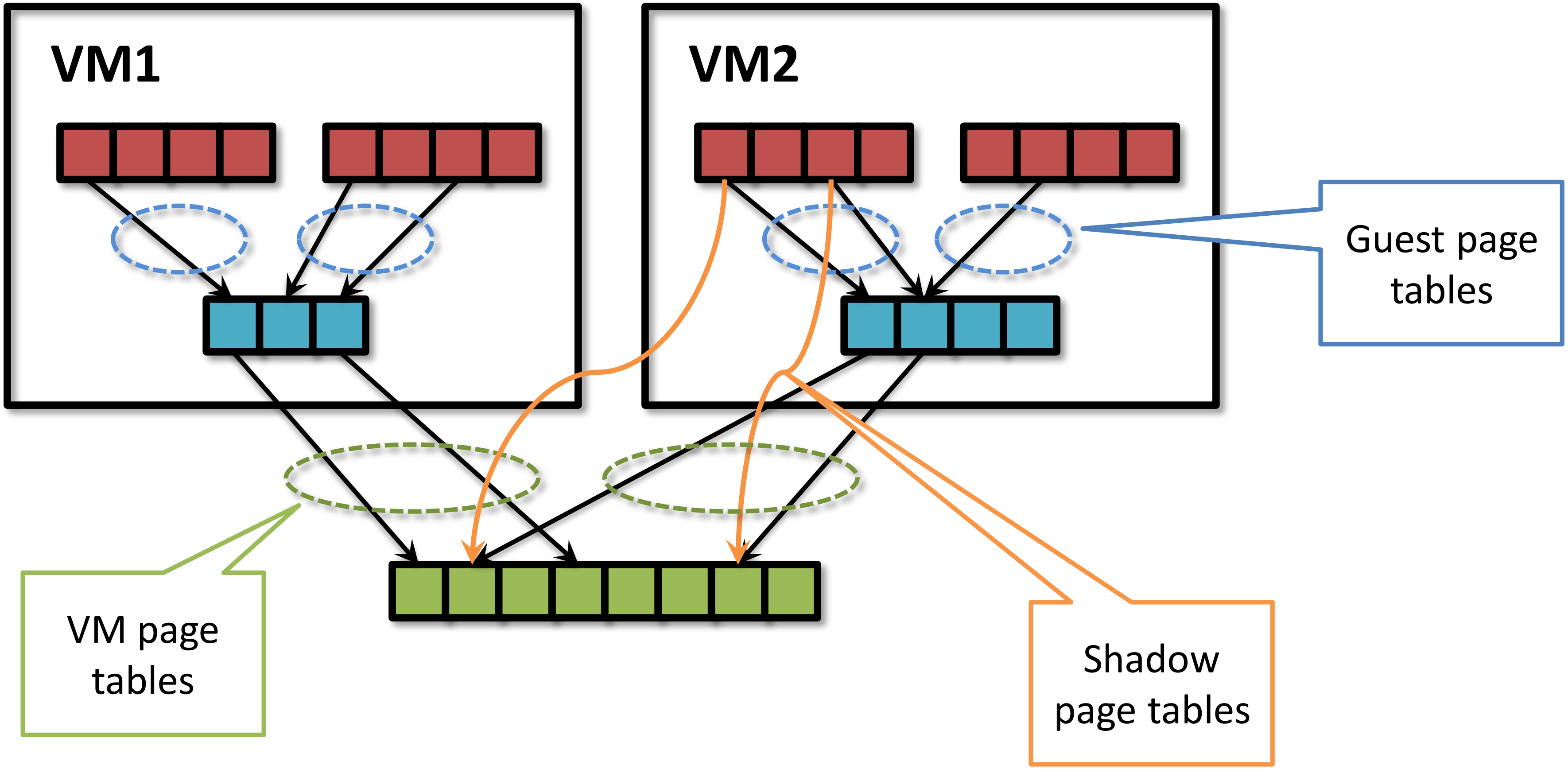
Memory virtualization (software)

- Double address translation has a high overherad

🟥 Guest virtual memory

🟦 Guest „physical” memory

🟩 Machine physical memory

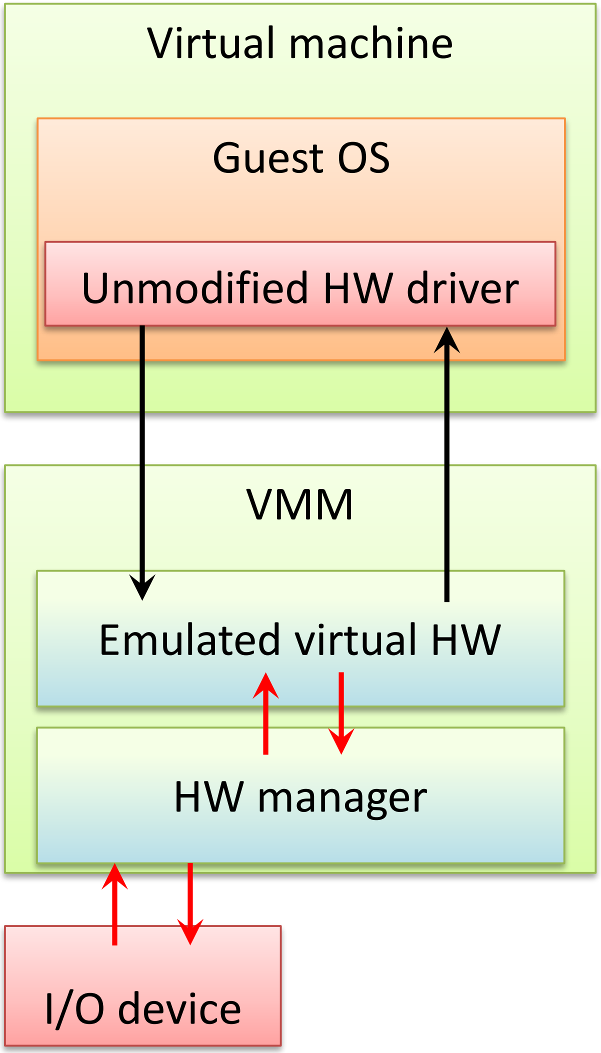


Memory virtualization – paravirtualization and hardware support

- Paravirtualization
  - Also uses shadow page tables
  - Modifying the guest OS source code
  - When the OS modifies it's page tables, it should notify the VMM also
- HW support for virtualization
  - HW support in the recent CPUs
    - AMD Rapid Virtualization Indexing , Intel Extended Page Tables
  - Nested page table
    - Storing guest physical -> machines physical translation
    - Traversed by HW address translation
  - Tagging TLB entries
  - Great performance increase:
    - 2008. 04., KVM: MMU paravirtualization is dead
    - 2009., VMware: Performance Evaluation of AMD RVI Hardware Assist, 42% improvement in some cases

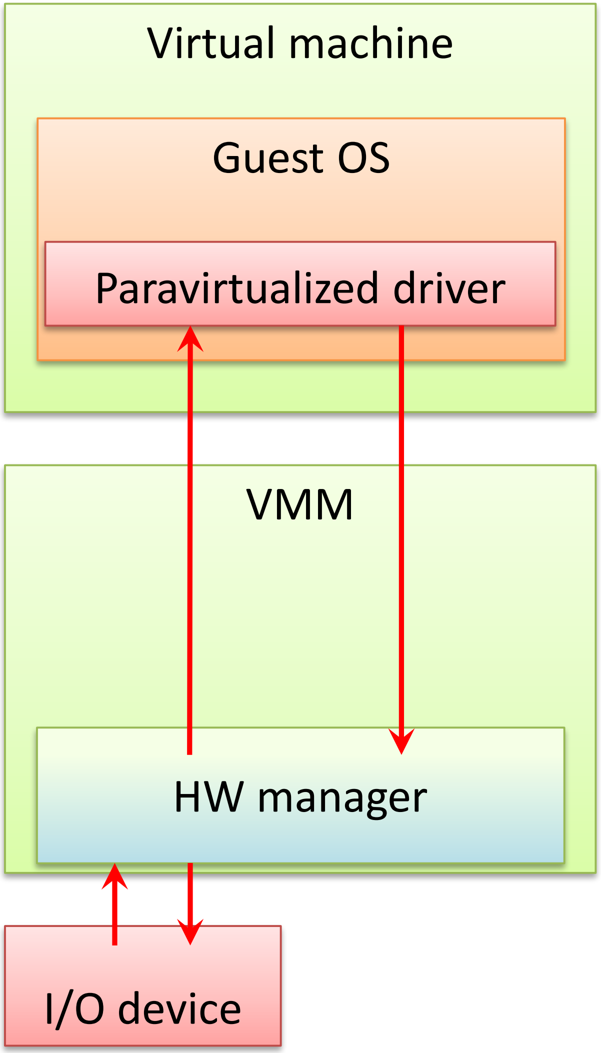
I/O virtualization – software

- Emulating the whole real communication
  - No special drivers (compatibility)
  - Can be really slow
- E.g.: many VMM-s emulate a TRIO VGA card (or other legacy type), because every OS has drivers for it



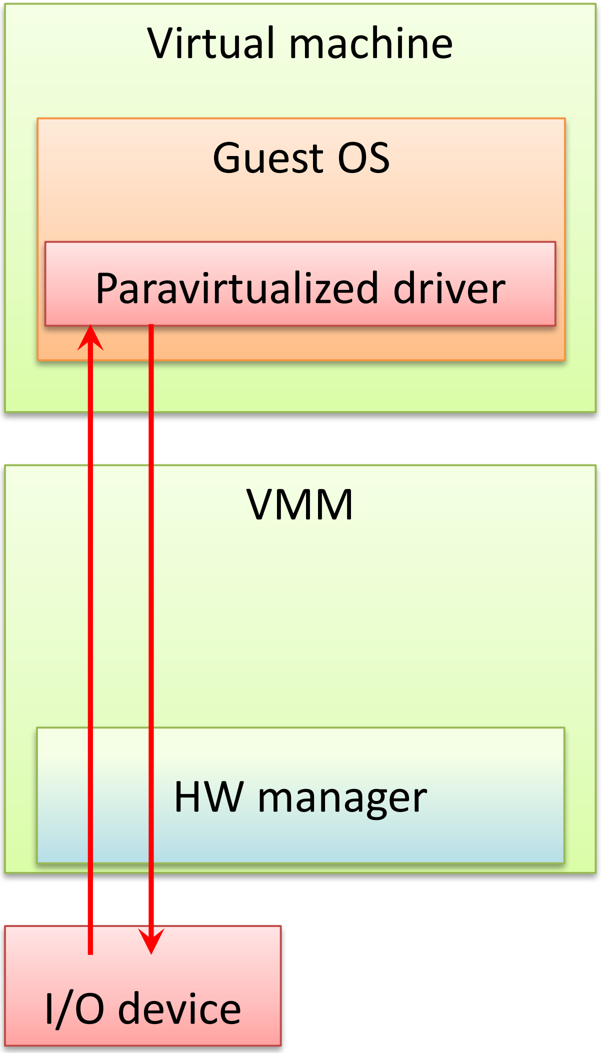
I/O virtualization – paravirtualization

- Operation
  - A special (virtual)driver is installed in the guest OS
  - Simplified calls
  - Communication through shared memory
  - Efficient operation
- Special package installed in the VM:
  - VMware Tools, Virtual PC Additions
  - **Always install these!**



I/O virtualization – hardware support

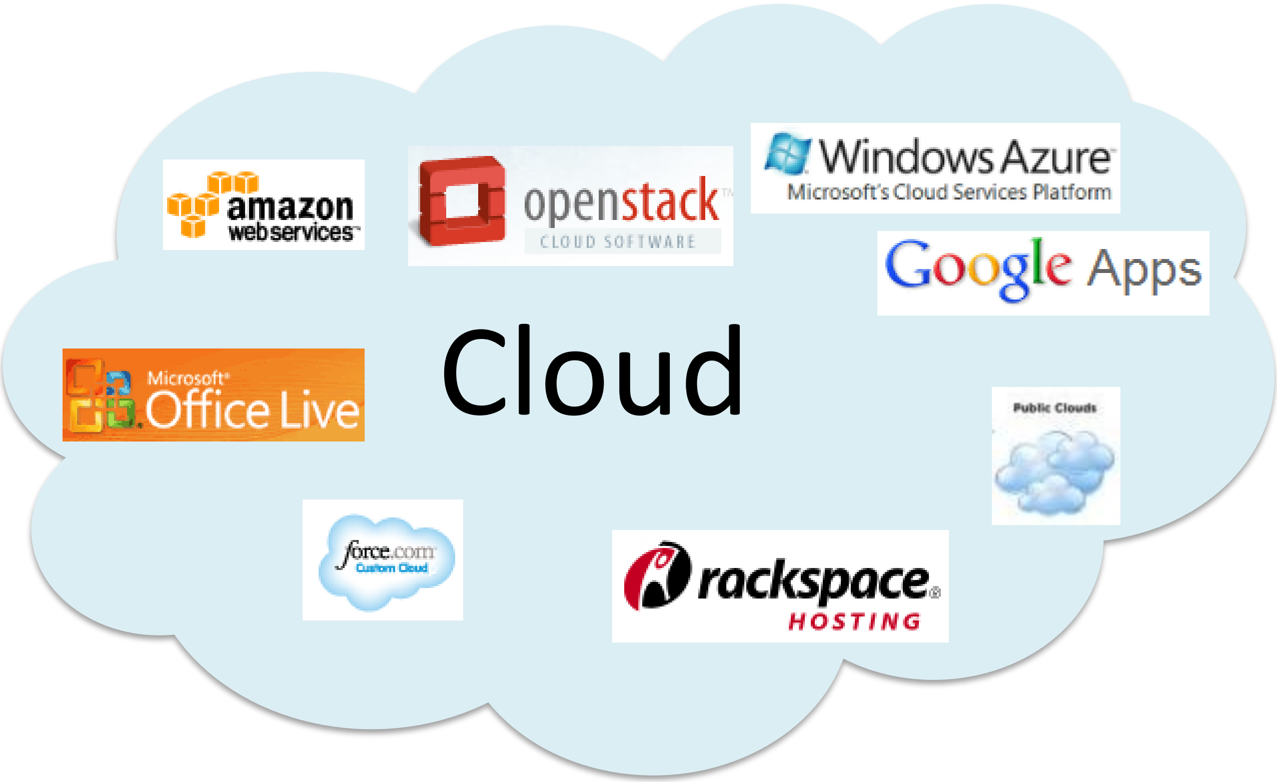
- Allowing direct access to an I/O device are not safe without a supervisor
  - Shared address range
  - More guest machines can cause conflicts
- Solutions
  - Using a HW level (fast) supervisor
    - Intel VT-d, AMD IOMMU
  - Using special I/O devices which are aware of the virtualized usage
    - PCI standard extensions: I/O Virtualization (IOV)
- Some I/O devices
  - can be shared between VMs
  - can be directly assigned to one VM
  - E.g.: GPU
  - Problems: VM context change



Products and companies

vmware	ESXi, vSphere…
Xen	open source hypervisor
CiTRiX	XenServer, XenApp
Microsoft	Virtual PC, Hyper-V, System Center
Sun microsystems ORACLE	Solaris Containers, Oracle VM, VirtualBox
KVM	Kernel based Virtual Machine (KVM)
IBM	mainframe, powerVM
	…

Cloud computing



Cloud

Types of cloud computing

IaaS

- Infrastructure-as-a-Service
- Getting a VM
- Amazon EC2, RackSpace…

PaaS

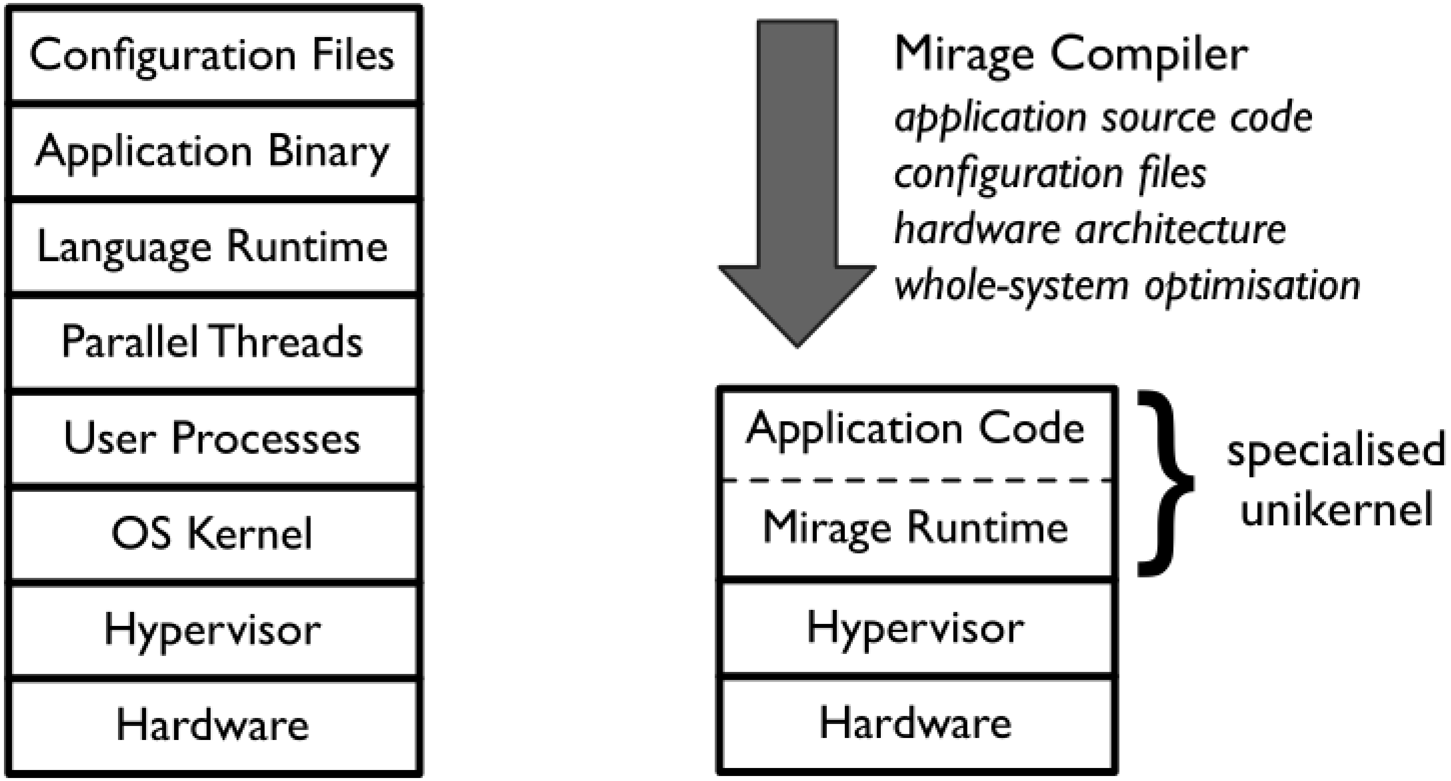
- Platform-as-a-Service
- Getting a runtime environment
  - Java container, .NET, database…
- MS Azure, Google AppEngine…

SaaS

- Software-as-a-Service
- Getting a service
- Google Docs, Microsoft Office 365, SalesForce CRM…

Current developements, future trends?

- Mirage OS
  - – Idea: compile a lightweight OS for a specific task
    - E.g.: for running a webserver



Summary

- Virtualization benefits
  - Better utilization, portability, sandboxing, …
- Virtualization types (levels)
  - Platform (classic), OS, Application, Presentation
- Virtualizing
  - CPU
  - Memory
  - I/O devices
- Paravirtualization
  - Modify the existing OS or drivers to achie greater efficiency than pure VMM SW solutions
- Cloud computing
  - IaaS, PaaS, SaaS