Virtualization - Introduction

- What is a Virtualization?
- Types of Virtualization
- Role of the Hypervisor
- Why Virtualization?

What is Virtualization?

"A virtual machine is taken to be an *efficient*, *isolated duplicate* of the real machine."

G.J. Popek, R.P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures", CACM, July 1974

Duplicate: VM should behave identically to the real machine.

Isolated: Several VMs can share real machine without interfering with each other.

Efficient: VM should execute at speed comparable to that of the real machine.

What is Virtualization?

Duplicate: VM should behave identically to the real machine.
- Software **cannot distinguish** between real and virtual hardware.
- Exceptions: Fewer resources, some timing differences.

Isolated: Several VMs can share real machine without interfering with each other.

Efficient: VM should execute at speed comparable to that of real machine.
- Requires that most instructions are **executed directly** by real hardware.

Simulation, Emulation, Virtualization

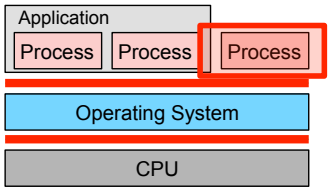
Simulation: Abstract model of a system is functionally simulated.

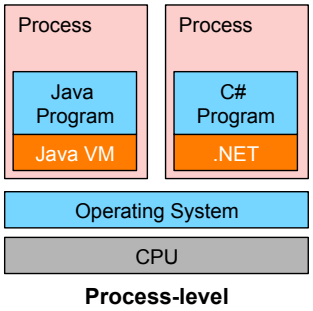
Emulation: Hardware or software (or both) emulates the behavior of the guest in a host so that emulated behavior is close to behavior of real system.

"Simulators as high-level emulators."

Virtualization: Virtualization involves simulating parts of a computer's hardware - enough for a guest operating system to run unmodified - but most operations still occur on the real hardware for efficiency reasons.

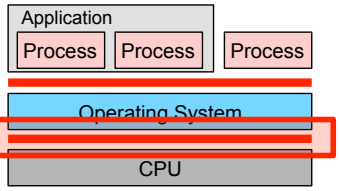
Types of Virtualization

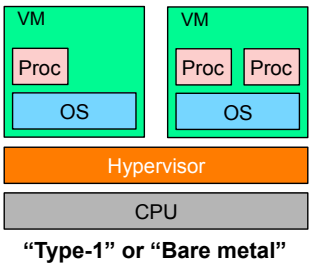




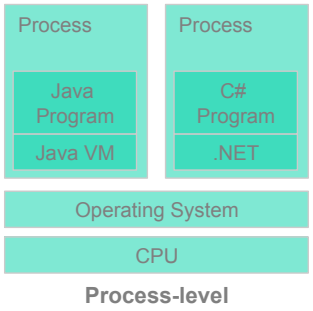
Process-level

Types of Virtualization

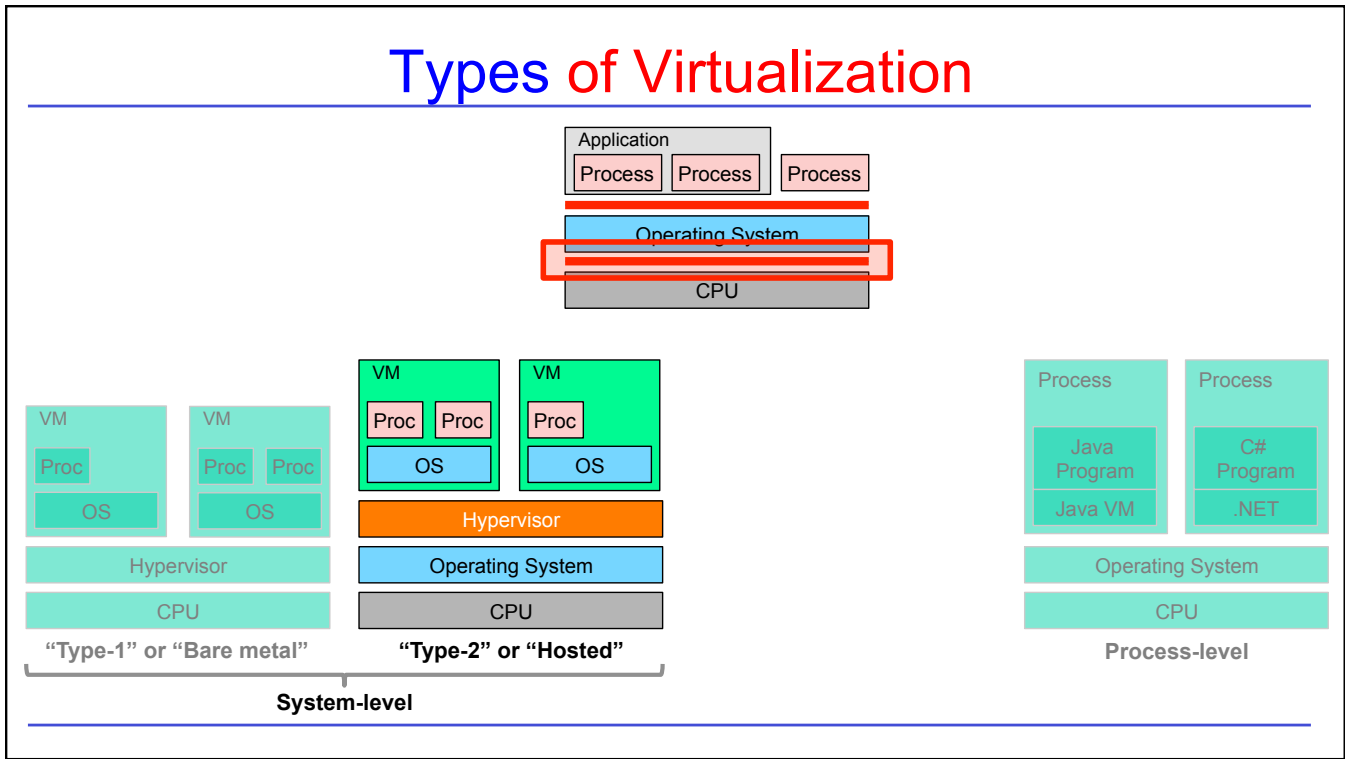


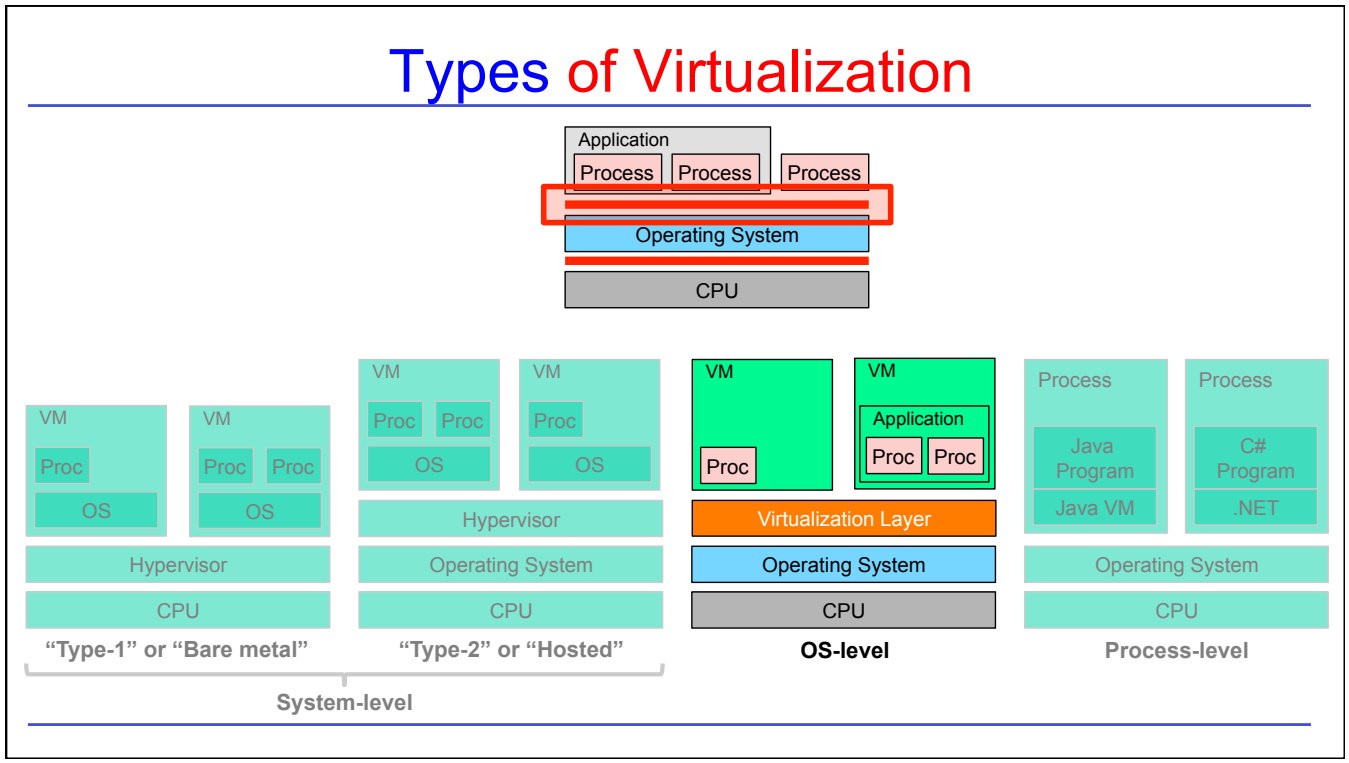


"Type-1" or "Bare metal"



Process-level


Types of Virtualization
Application
Process
Process
Process
Operating System
CPU
VM
Proc
OS
VM
Proc
Proc
OS
VM
Proc
Proc
OS
VM
Proc
OS
Hypervisor
Operating System
CPU
Process
Java Program
Java VM
Process
C# Program
.NET
Operating System
CPU
"Type-1" or "Bare metal"
"Type-2" or "Hosted"
System-level
Process-level


Types of Virtualization
Application
Process
Process
Process
Operating System
CPU
VM
Proc
OS
VM
Proc
Proc
OS
VM
Proc
Proc
OS
VM
Proc
OS
VM
Proc
Application
Proc
Proc
Hypervisor
Operating System
CPU
Virtualization Layer
Operating System
CPU
Process
Java Program
Java VM
Process
C# Program
.NET
Operating System
CPU
"Type-1" or "Bare metal"
"Type-2" or "Hosted"
OS-level
System-level
Process-level

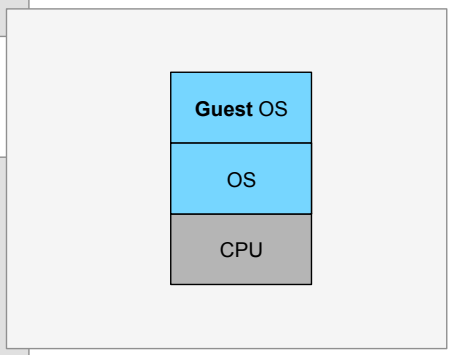
Role of the Hypervisor

Hypervisor: Program that runs on hardware or OS to implement the virtual machine(s).

Also called "Virtual Machine Monitor".

Controls resources:

- Partitions hardware
- Schedules guests
- Mediates access to shared resources
- Switches between "world" as seen by hypervisor and guests.



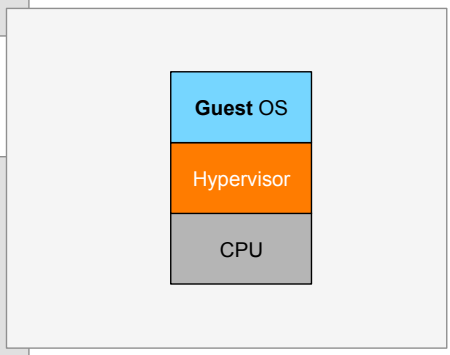
Role of the Hypervisor

Hypervisor: Program that runs on hardware or OS to implement the virtual machine(s).

Hypervisor "control resources".

Outlook on implications:

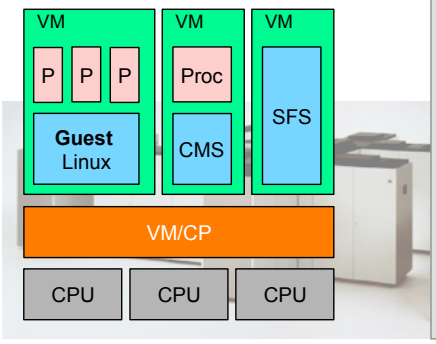
- Hypervisor must run in privileged mode.
- Guests must run in non-privileged mode.
- Privileged instructions of guests must be intercepted and run by hypervisor.



Why Virtualization?

Historically, used for easier sharing of expensive mainframes, e.g.

- IBM CP-40 research project in early 60's
- Became CP-67 in 1967 on System/360
- VM/CMS in 1972



VMs can run

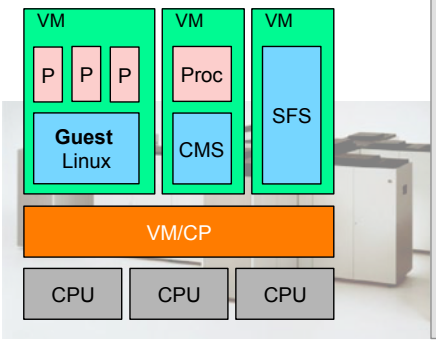
- single-user guest OS (e.g. CMS)
- mainstream guest OS (e.g. MVS, AIX, Linux)
- "second level" guest VMs
- specialized OS subsystems (e.g. spooling, shared file server, etc.)

Great support for legacy systems.

Why Virtualization?

Historically, used for easier sharing of expensive mainframes, e.g.

- IBM CP-40 research project in early 60's
- Became CP-67 in 1967 on System/360
- VM/CMS in 1972



Other than for mainframes, largely disappeared in 80's.

- Time sharing supported by other operating systems.
- Hardware too cheap to worry about consolidation.

Why Virtualization?

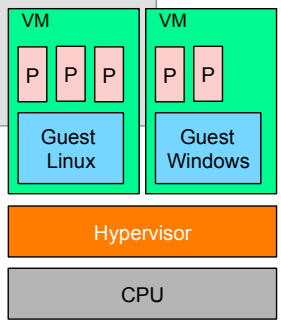
Improved Isolation helped virtualization become popular again in the early 2000's.

Isolation:
- improved Quality of Service
- improved security
- different concurrent OSs

Encapsulation:
- replication
- migration/consolidation
- checkpointing
- debugging

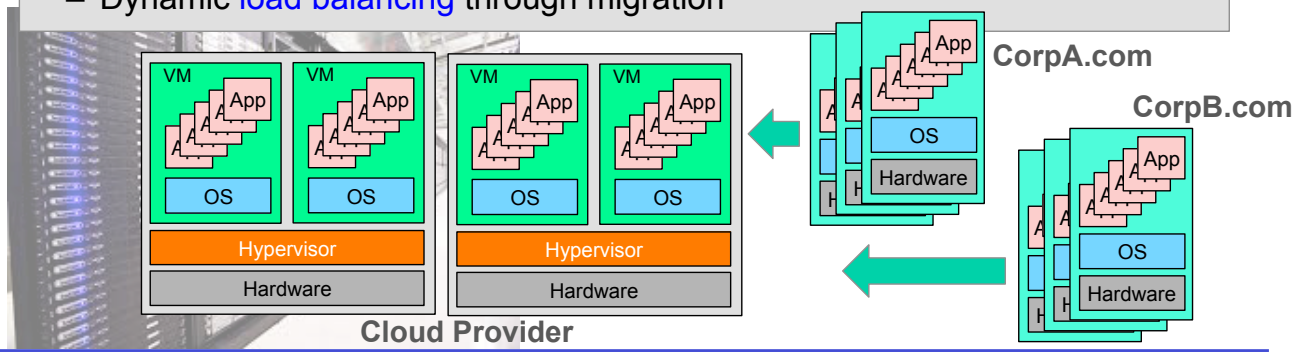
"This would be mostly unnecessary ... if OSes were doing their job!"

Guernot Heiser



Why Virtualization?

Cloud Computing is today's main driver.
- Increased utilization by sharing hardware (resource overallocation)
- Reduced maintenance through increased scale (pets vs. cattle)
- On-demand provisioning
- Dynamic load balancing through migration



Virtualization - Introduction

- What is a Virtualization?
- Types of Virtualization
- Role of the Hypervisor
- Why Virtualization?