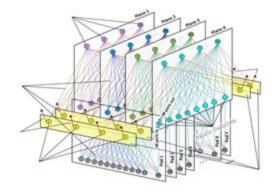
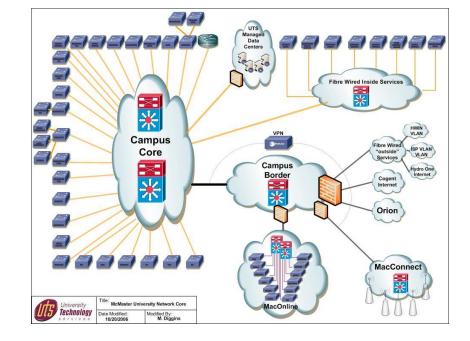
Large Systems:

Design + Implementation + Administration







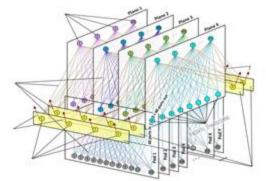
4 november 2024

Large Systems: Design + Implementation

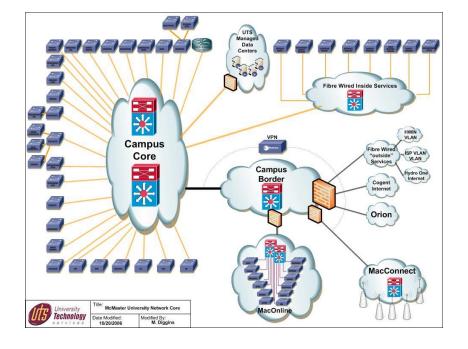
2024-2025

Week2-L3: Virtualization- Part 3

Shashikant Ilager shashikantilager.com





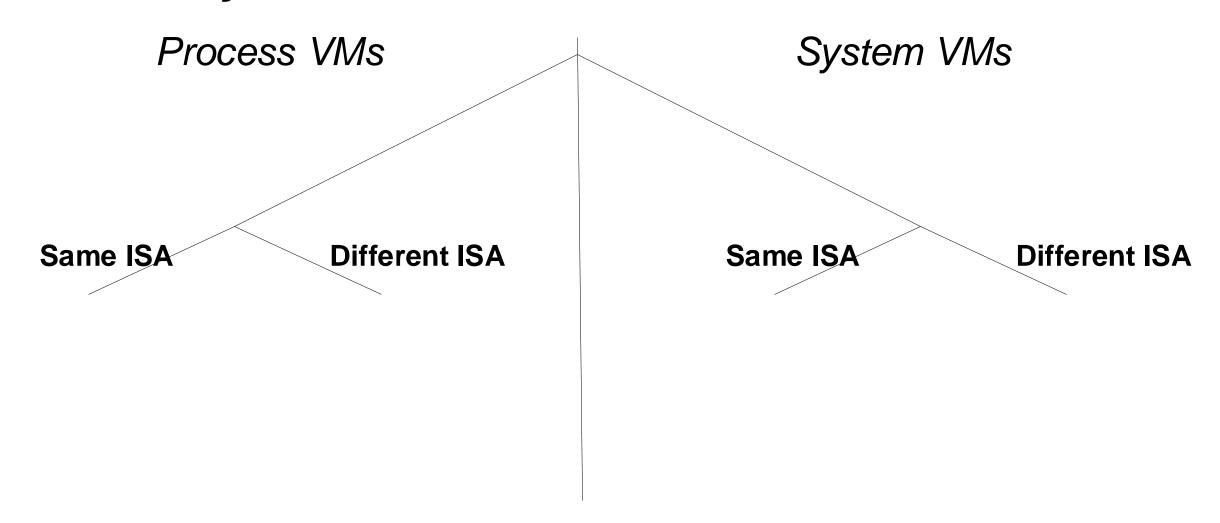


4 november 2024

Recap

• Difference Process VM and System VM in terms of the interface virtualized?

Taxonomy

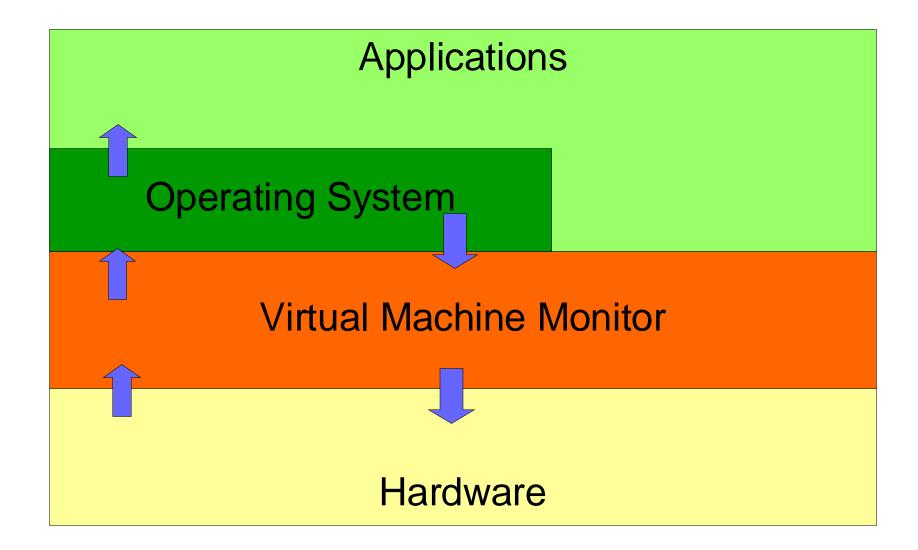


Java, Xen and Android emulation

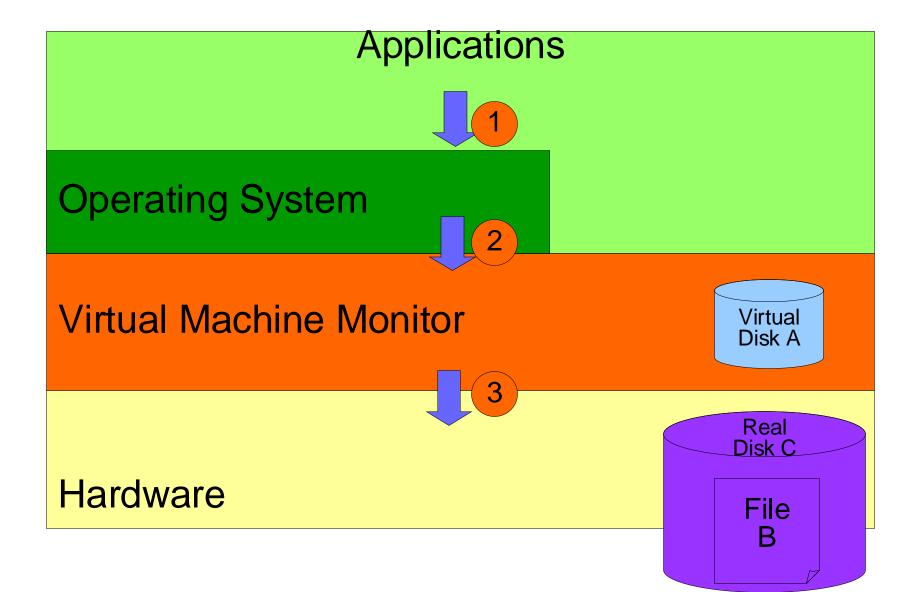
Recap

- What is Intel VT-x?
- What problem does it solve?

Recap: App Scheduling



Recap: Example- Reading from disk



Native, Hosted, Paravirtualized VMs

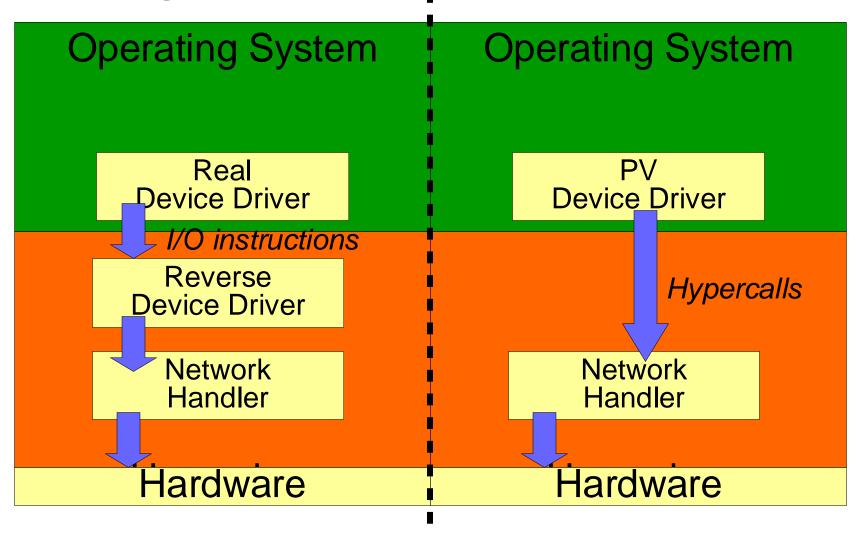
		Apps						
Apps		Guest OS		Guest OS		Apps		Modify the Guest OS!
Guest OS		Hypervisor		Guest OS		Oucst Oo:		
Hypervisor		Host OS		Hyperviso	r			
Hardware		Hardware		Hardware)			

Paravirtualization

- System VMs can be faster when Guest OS can be modified for virtualization
- Showcased in Xen Project, also used in KVM
- Modified
 - Linux
 - Windows XP
- Near native performances



Paravirtualizing I/O

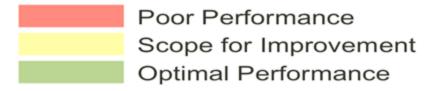


Guest OS knows it is not running on real hardware

Xen Evolution

- Problems:
 - only open-source OSes can be modified
 - Xen implementation tricks not on x86-64
- New approach: Start from Full virtualization with Hardware Support (e.g. VT-x)
- Apply Paravirtualization in areas where speed can be gained:
 - Disk and network I/O
 - Interrupts and timers
 - Emulated motherboard, legacy boot
 - Privileged instructions, page tables

Xen Mode: HVM



P = Paravirtualized

VS = Software Virtualized (QEMU)

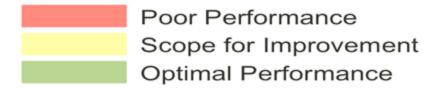
VH = Hardware Virtualized

Disk and Network Timers Book Book Oard, Istructions ith

Shortcut	Mode	With					
HVM / Fully Virtualized	HVM		VS	VS	VS	VH	

Source: Lars Kurth, http://wiki.xen.org/wiki/Virtualization_Spectrum

Xen Mode: PV



P = Paravirtualized

VS = Software Virtualized (QEMU)

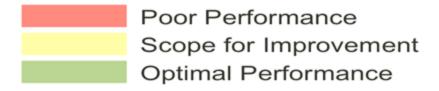
VH = Hardware Virtualized

Disk and Network Timers Priviled Priviled Paged Instructions,

Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH

PV P P P

Xen Mode: HVM + PV Drivers



P = Paravirtualized

VS = Software Virtualized (QEMU)

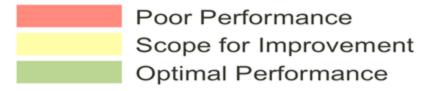
VH = Hardware Virtualized

	4	En.	P _a	
Oist a	Network	Emulated Legal	Privileged In Alberto and	
*//0	No.	\$ 4 2. PO 3.	Moth age	PST.
	Work	mers	Ook of the	bleschon.
			.4	à

Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	Р	VS	VS	VH

PV	PV	Р	Р	Р	Р

Xen Mode: PVHVM Drivers



P = Paravirtualized

VS = Software Virtualized (QEMU)

VH = Hardware Virtualized

Disk and Network Timers Boorboard, Schools & Timers Boorboard, Selections,

Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	Р	VS	VS	VH
PVHVM	HVM	PVHVM Drivers	Р	Р	VS	VH

PV	PV	Р	Р	Р	Р

Xen Mode: PVH

Poor Performance
Scope for Improvement
Optimal Performance

P = Paravirtualized

VS = Software Virtualized (QEMU)

VH = Hardware Virtualized

Disk and Network Timers Book Book Colors, the the state of the state o

Shortcut	Mode	With					
HVM / Fully Virtualized	HVM		VS	VS	VS	VH	Windows
HVM + PV drivers	HVM	PV Drivers	Р	VS	VS	VH	lows
PVHVM	HVM	PVHVM Drivers	Р	Р	VS	VH	Linux,
PVH	PV	pvh=1	Р	Р	Р	VH	x, BSDs,
PV	PV		Р	Р	Р	Р	, i



Poor Performance
Scope for Improvement
Optimal Performance

P = Paravirtualized

VS = Software Virtualized (QEMU)

VH = Hardware Virtualized

Disk and Network Timers Boorboard

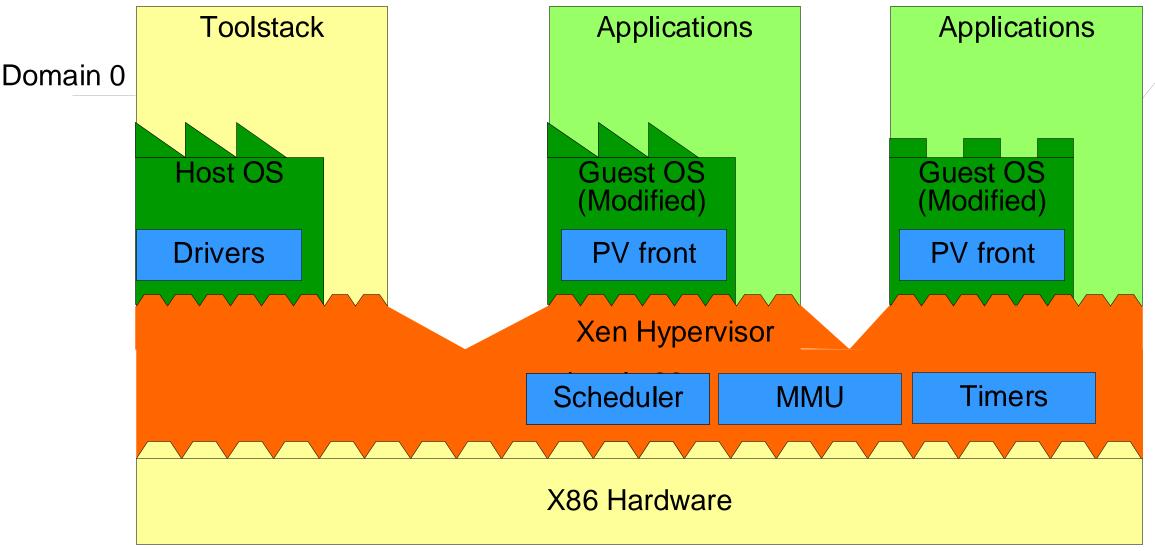
Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	Р	VS	VS	VH
KVM	HVM		Р	VS	V: P	VH
PVHVM	HVM	PVHVM Drivers	Р	Р	VS	VH
PVH	PV	pvh=1	Р	Р	Р	VH
PV	PV		Р	Р	Р	Р

KVM Architecture

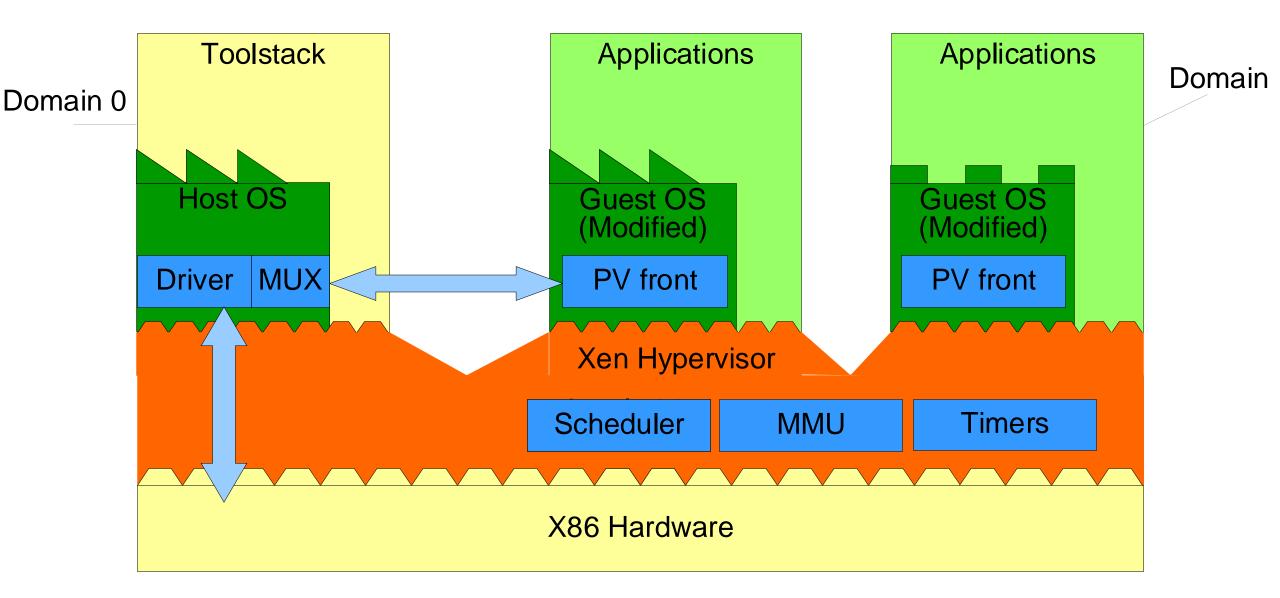
- The kernel is both OS and hypervisor
- The guest kernel uses QEMU to emulate hardware
- Guest + QEMU talk to kvm.ko kernel module

Xen Architecture

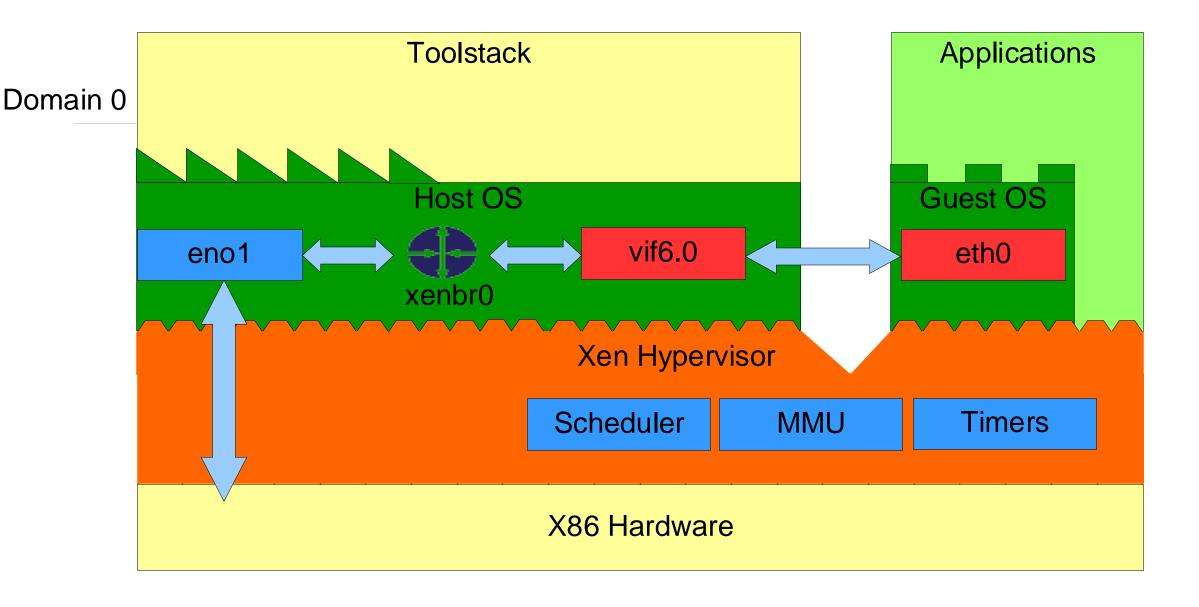
DomU = Unprivileged



Xen Architecture



Xen Virtual Devices

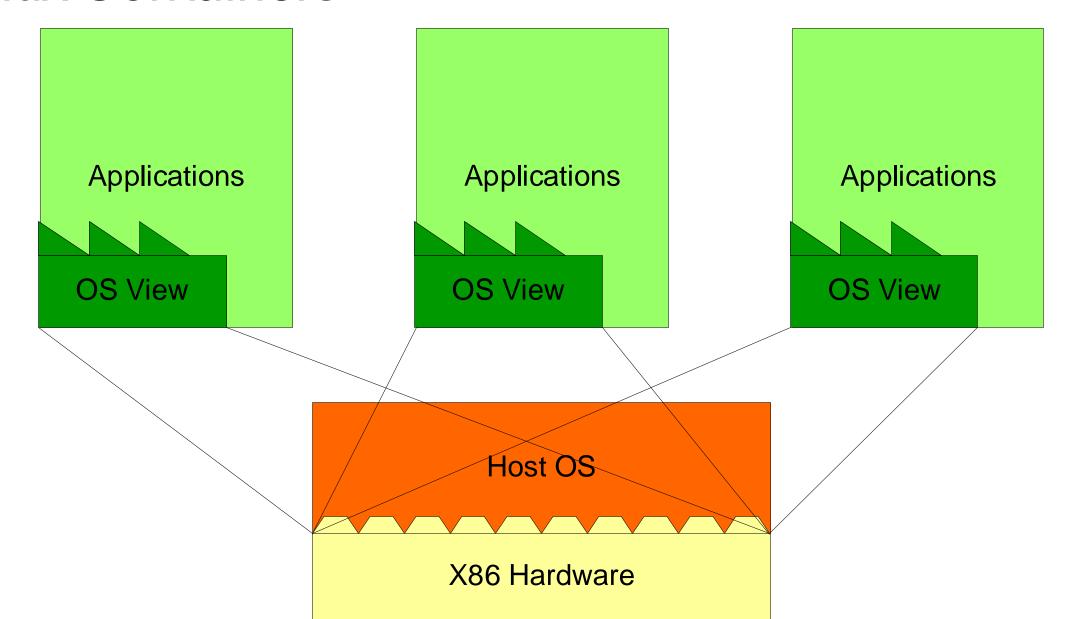


Operating-System Level Virtualization

- In between System VM and Process VM
- Not System VM:
 - Cannot choose OS
- Not Process VM:
 - Multiple processes, not isolated
- As if multiple instances of the same OS are running on the same machine
 - Example: Linux Containers
 - cf. Docker



Linux Containers

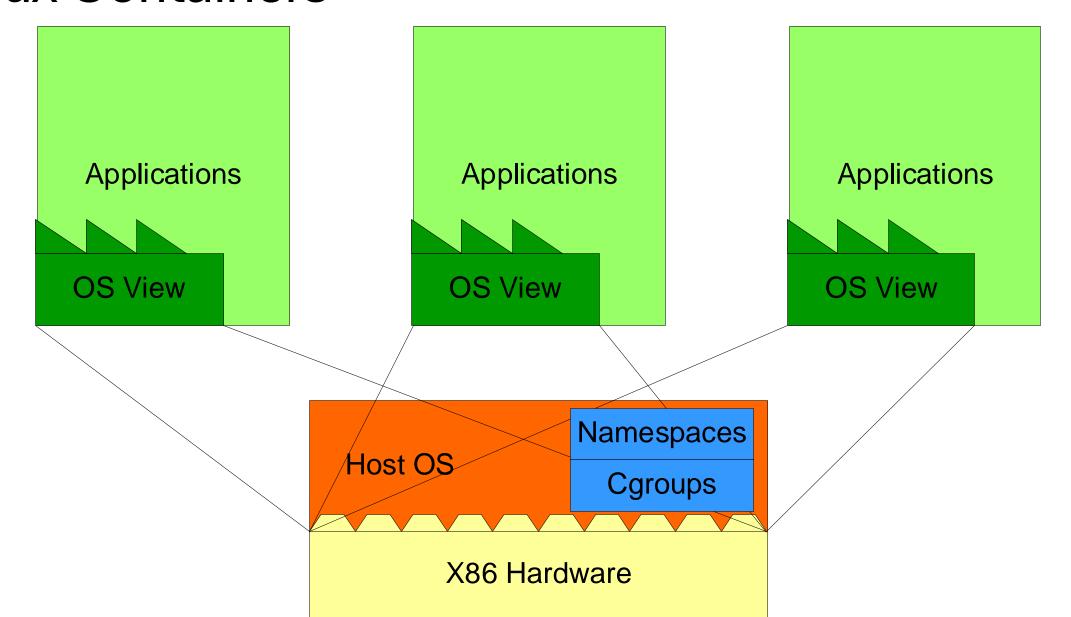


Containers

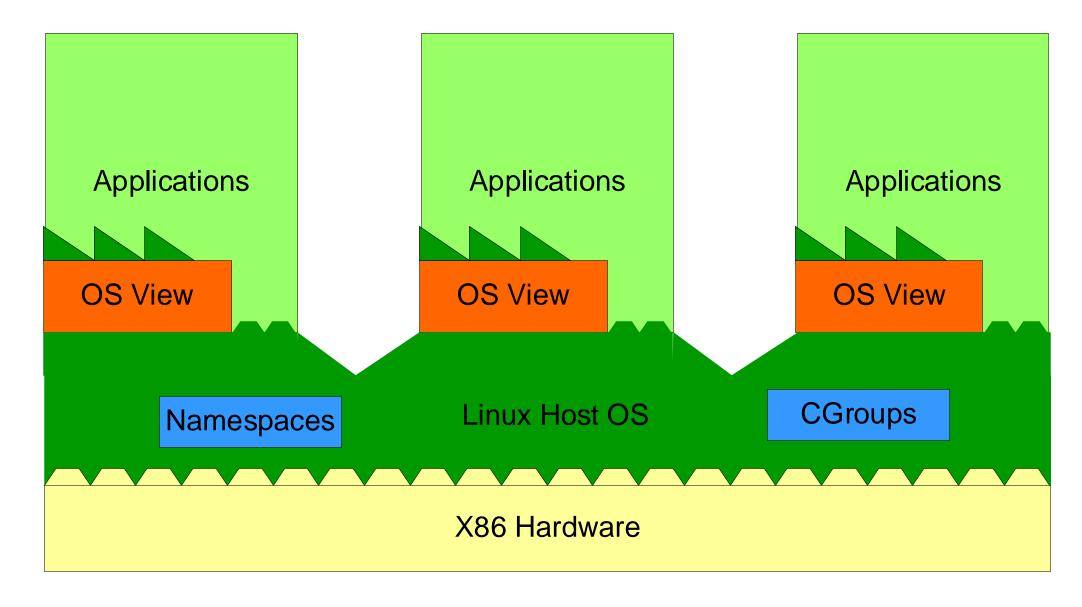
- Only one kernel is installed, and the hardware is not virtualized.
- Rather, the operating system is virtualized, providing processes within a container with the impression that they are the only processes on the system.
- One or more containers can be created, and each can have its own applications, network addresses, user accounts, and so on.

Source: Silberschatz et al. 9th US Edition.

Linux Containers



Linux Containers



Containers Continued...