Xen is not just paravirtualization

Dongli Zhang

Oracle Asia Research and Development Centers (Beijing)

dongli.zhang@oracle.com

December 16, 2016

Plan

- Virtualization
- Xen Virtualization

Plan

- Virtualization
- Xen Virtualization

When discussing virtualizatin ...

1) CPU Virtualization?

2) Memory Virtualization?

3) Device Virtualization?

What is virtualization

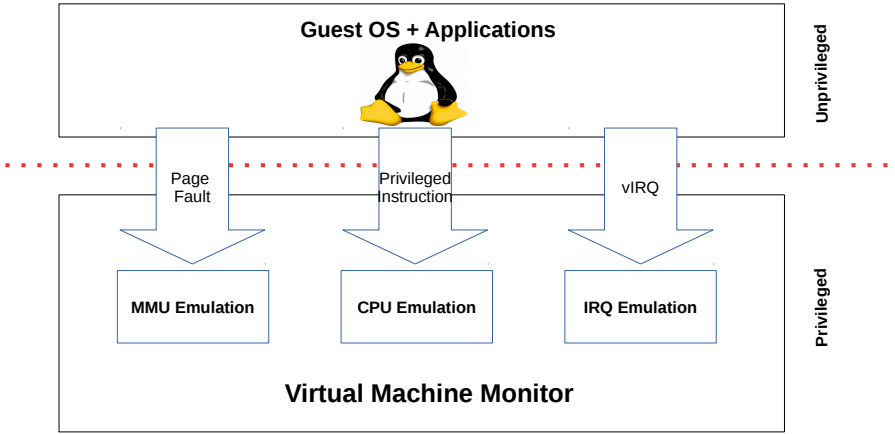
- A virtual machine is taken to be an efficient, isolated duplicate of the real machine (by Formal Requirements for Virtualizable Third Generation Architectures, Gerald J.Popek and Rebert P. Goldberg, 1974)

What is virtualization

- A virtual machine is taken to be an efficient, isolated duplicate of the real machine (by Formal Requirements for Virtualizable Third Generation Architectures, Gerald J.Popek and Rebert P. Goldberg, 1974)

Trap and Emulate

- Virtual Machine (Guest) at **Unprivileged Mode**
- Virtual Machine Monitor (Host or Hypervisor) at **Priviledged Mode**



x86 is NOT virtualizable

- Virtualizable Architecture: all **sensitive instructions** must also be **privileged instructions** (by Gerald J.Popek and Rebert P. Goldberg)
- **critical instructions** = *sensitive instructions* − *privileged instructions*

x86 is NOT virtualizable

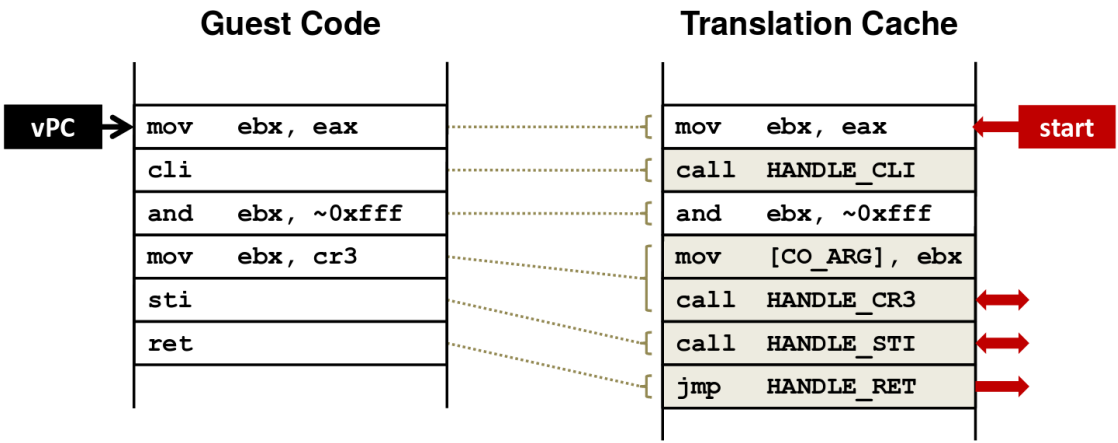
- Virtualizable Architecture: all **sensitive instructions** must also be **privileged instructions** (by Gerald J.Popek and Rebert P. Goldberg)
- **critical instructions** = *sensitive instructions* − *privileged instructions*
- 18 critical instructions on x86 (Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor. USENIX Security 2000):
  - *SGDT/SIDT/SLDT, SMSW, PUSHF/POPF*
  - *LAR/LSL, VERR/VERW, POP/PUSH*
  - *CALL, JMP, INT n, RET*
  - *STR, MOV*

x86 is NOT virtualizable

- Virtualizable Architecture: all **sensitive instructions** must also be **privileged instructions** (by Gerald J.Popek and Rebert P. Goldberg)
- **critical instructions** = *sensitive instructions* − *privileged instructions*
- 18 critical instructions on x86 (Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor. USENIX Security 2000):
  - *SGDT/SIDT/SLDT, SMSW, PUSHF/POPF*
  - *LAR/LSL, VERR/VERW, POP/PUSH*
  - *CALL, JMP, INT n, RET*
  - *STR, MOV*
- Solutions:
  - Binary Translation (QEMU, VMWare)
  - Paravirtualization (Xen)
  - Hardware-Assisted Virtualization (Xen, KVM, VMWare based on Intel-VT and AMD-V)

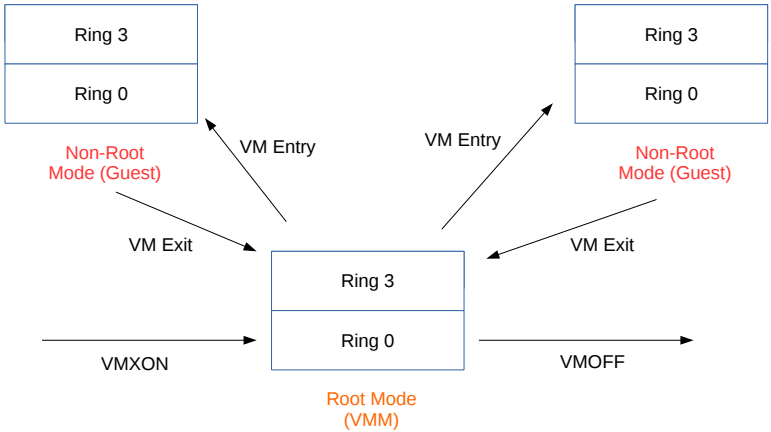
Solution 1/3: Binary Translation

- **philosophy: rewrite critical instructions**



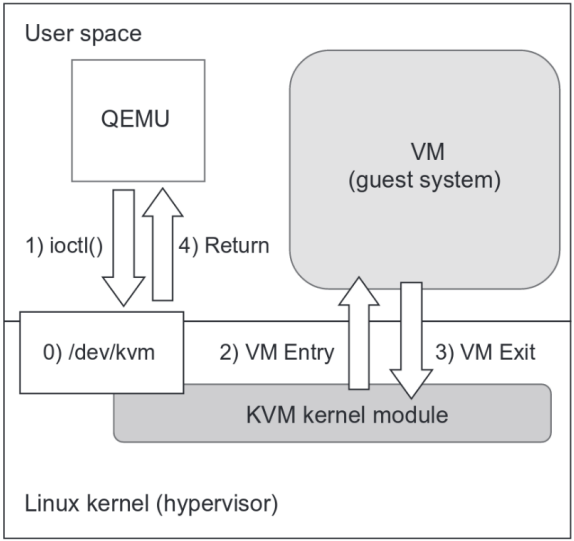
Solution 2/3: Hardware Virtualization (Intel VT)

- **philosophy: instroduce new privileged mode**



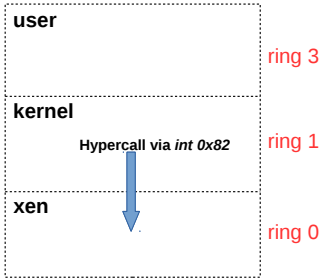
KVM (Kernel-based Virtual Machine)

- CPU hardware virtualization extensions (Intel VT or AMD-V)
- Loadable kernel module (kvm.ko, kvm-intel.ko/kvm-amd.ko)
- QEMU as userspace emulator

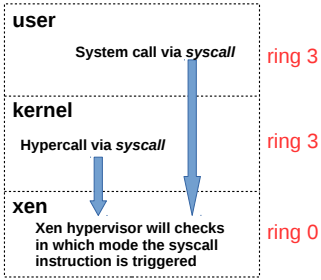


Solution 3/3: Paravirtualization

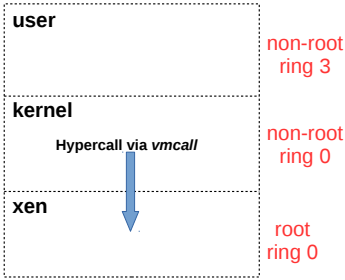
- **philosophy: replace critical instructions with hypercalls**
- A hypercall is a software trap from a domain to the hypervisor, just as a syscall is a software trap from an application to the kernel
  - x86_32: *int 0x82*
  - x86_64: *syscall instruction*
  - x86 Intel-VT *vmcall instruction*



x86 32-bit pvm



x86 64-bit pvm



x86 vt-x hvm/pvhvm

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)
- Paravirtualization (Xen)

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)
- Paravirtualization (Xen)
- Hardware-assisted Virtualization (KVM, Xen, VMware)

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)
- Paravirtualization (Xen)
- Hardware-assisted Virtualization (KVM, Xen, VMware)
- OS-level Virtualization (Linux Container)

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)
- Paravirtualization (Xen)
- Hardware-assisted Virtualization (KVM, Xen, VMware)
- OS-level Virtualization (Linux Container)
- Programming Language Virtualization (Java, .NET CLR)

State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)
- Paravirtualization (Xen)
- Hardware-assisted Virtualization (KVM, Xen, VMware)
- OS-level Virtualization (Linux Container)
- Programming Language Virtualization (Java, .NET CLR)
- Library Virtualization (Wine, Cygwin)

What is Xen

Wikipedia

Xen Project is a hypervisor using a **microkernel** design, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

What is Xen

Wikipedia

Xen Project is a hypervisor using a **microkernel** design, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

SOSP 2003: Xen and the Art of Virtualization

This paper presents Xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality.

What is Xen

Wikipedia

Xen Project is a hypervisor using a **microkernel** design, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

SOSP 2003: Xen and the Art of Virtualization

This paper presents Xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality.

Basic Idea of Paravirtualization

Actively inform the hypervisor with the action guest is going to taken via hypercall

Xen Framework 1/2

xen hypervisor (microkernel): dictator

- scheduling, memory management, interrupt and device control
- per-domain and per-vcpu info management

Xen Framework 1/2

xen hypervisor (microkernel): dictator

- scheduling, memory management, interrupt and device control
- per-domain and per-vcpu info management

dom0 (host): privileged admin

- xm/xend/xl (libxc)
- pygrub/hvmloader
- xenstored
- qemu and paravirtual driver backend
- native device driver

Xen Framework 1/2

xen hypervisor (microkernel): dictator

- scheduling, memory management, interrupt and device control
- per-domain and per-vcpu info management

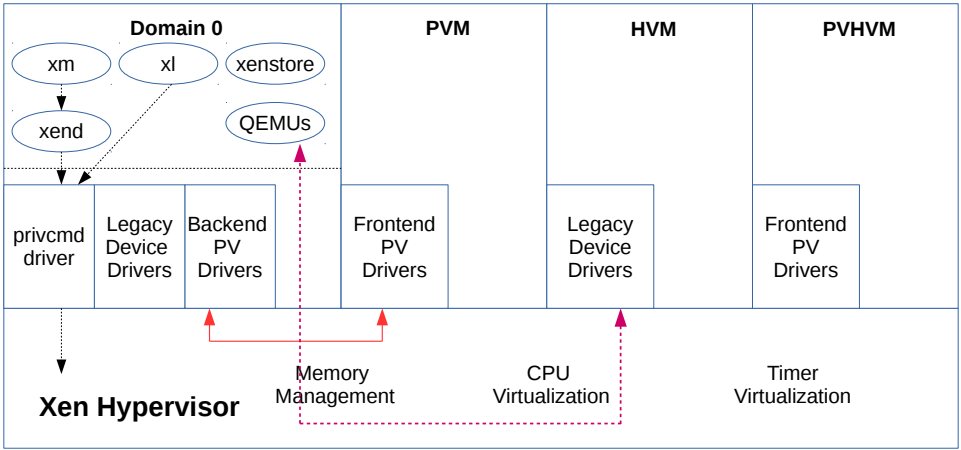
dom0 (host): privileged admin

- xm/xend/xl (libxc)
- pygrub/hvmloader
- xenstored
- qemu and paravirtual driver backend
- native device driver

domU (guest): non-privileged user

- paravirtual driver frontend

Xen Framework 2/2



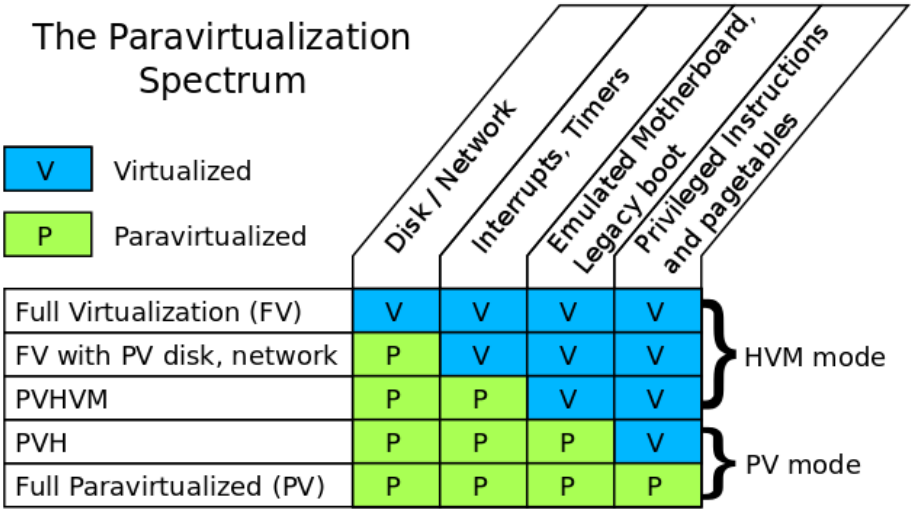
Convert Linux to Paravirtual Dom0/DomU

- ELF notes (Linux) or __xen_guest section (MiniOS) in kernel image
- Enable xen features in .config when building kernel

```
105     ELFNOTE(Xen, XEN_ELFNOTE_GUEST_OS,       .asciz "linux")
106     ELFNOTE(Xen, XEN_ELFNOTE_GUEST_VERSION,  .asciz "2.6")
107     ELFNOTE(Xen, XEN_ELFNOTE_XEN_VERSION,    .asciz "xen-3.0")
108 #ifdef CONFIG_X86_32
109     ELFNOTE(Xen, XEN_ELFNOTE_VIRT_BASE,      _ASM_PTR __PAGE_OFFSET)
110 #else
111     ELFNOTE(Xen, XEN_ELFNOTE_VIRT_BASE,      _ASM_PTR __START_KERNEL_map)
112     /* Map the p2m table to a 512GB-aligned user address. */
113     ELFNOTE(Xen, XEN_ELFNOTE_INIT_P2M,       .quad PGDIR_SIZE)
114 #endif
115     ELFNOTE(Xen, XEN_ELFNOTE_ENTRY,          _ASM_PTR startup_xen)
116     ELFNOTE(Xen, XEN_ELFNOTE_HYPERCALL_PAGE, _ASM_PTR hypercall_page)
117     ELFNOTE(Xen, XEN_ELFNOTE_FEATURES,       .ascii "!writable_page_tables|pae_
118     ELFNOTE(Xen, XEN_ELFNOTE_SUPPORTED_FEATURES, .long (PVH_FEATURES) |
119                         (1 << XENFEAT_writable_page_tables) |
120                         (1 << XENFEAT_dom0))
121     ELFNOTE(Xen, XEN_ELFNOTE_PAE_MODE,       .asciz "yes")
122     ELFNOTE(Xen, XEN_ELFNOTE_LOADER,         .asciz "generic")
123     ELFNOTE(Xen, XEN_ELFNOTE_L1_MFN_VALID,
124         .quad _PAGE_PRESENT; .quad _PAGE_PRESENT)
125     ELFNOTE(Xen, XEN_ELFNOTE_SUSPEND_CANCEL, .long 1)
126     ELFNOTE(Xen, XEN_ELFNOTE_MOD_START_PFN,  .long 1)
127     ELFNOTE(Xen, XEN_ELFNOTE_HV_START_LOW,   _ASM_PTR __HYPERVISOR_VIRT_START)
128     ELFNOTE(Xen, XEN_ELFNOTE_PADDR_OFFSET,   _ASM_PTR 0)
129
130 #endif /*CONFIG_XEN */
"arch/x86/xen/xen-head.S" 130 lines --100%--
```

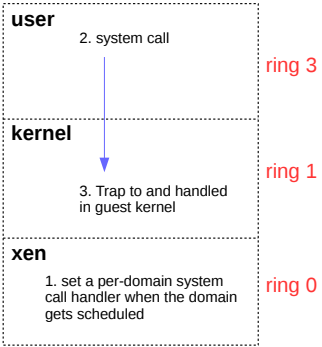
```
CONFIG_XEN=y
CONFIG_XEN_DOM0=y
CONFIG_XEN_PVHVM=y
CONFIG_XEN_512GB=y
CONFIG_XEN_SAVE_RESTORE=y
CONFIG_XEN_BLKDEV_FRONTEND=y
CONFIG_XEN_BLKDEV_BACKEND=m
CONFIG_XEN_NETDEV_FRONTEND=y
CONFIG_XEN_NETDEV_BACKEND=m
CONFIG_INPUT_XEN_KBDDEV_FRONTEND=m
CONFIG_XEN_FBDEV_FRONTEND=m
CONFIG_XEN_BALLOON=y
CONFIG_XEN_BALLOON_MEMORY_HOTPLUG=y
CONFIG_XEN_BALLOON_MEMORY_HOTPLUG_LIMIT=512
CONFIG_XEN_DEV_EVTCHN=m
CONFIG_XEN_BACKEND=y
CONFIG_XEN_XENBUS_FRONTEND=y
CONFIG_XEN_GNTDEV=m
CONFIG_XEN_GRANT_DEV_ALLOC=m
CONFIG_XEN_TMEM=m
CONFIG_XEN_PCIDEV_BACKEND=m
CONFIG_XEN_PRIVCMD=m
```

PV, HVM or PVHVM

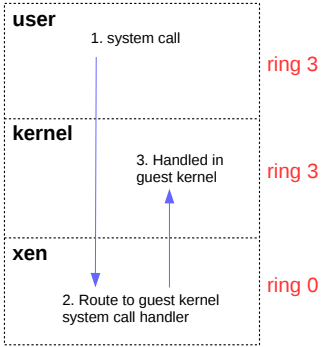


Xen CPU Virtualization

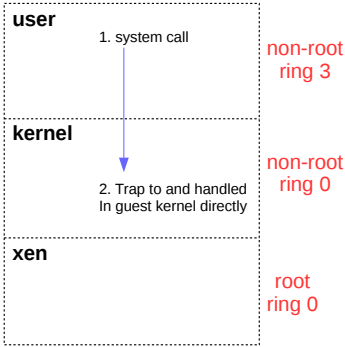
- vcpu ≈ task_struct
- domain ≈ container or process group
- xen schedules vcpu



x86 32-bit pvm



x86 64-bit pvm



x86 vt-x hvm/pvhvm

Xen Interrupt Virtualization: Event Channel 1/2

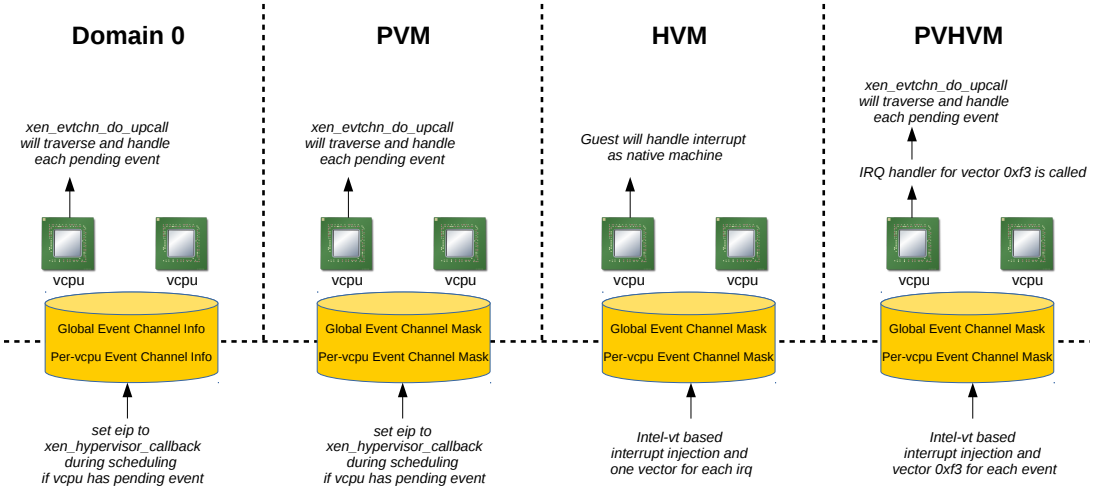
Event Channel Types

- Interdomain Event
- Virtual IRQ Event
- Physical IRQ Event
- IPI Event

Registration

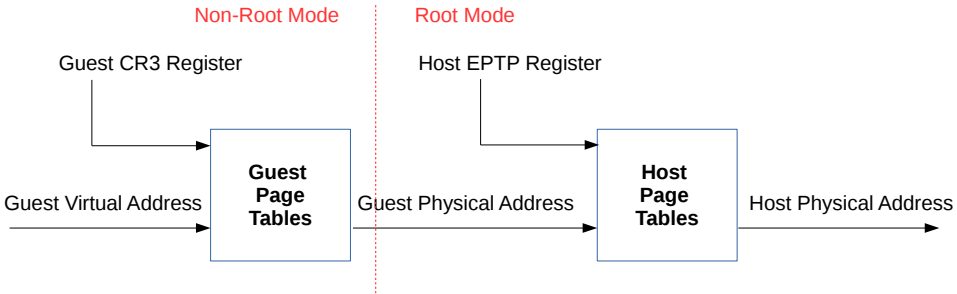
- PVM registers event channel handler to Xen via register_callback(CALLBACKTYPE_event, xen_hypervisor_callback)
- PVHVM sets HYPERVISOR_CALLBACK_VECTOR via HYPERVISOR_hvm_op(HVMOP_set_param, &a)

Xen Interrupt Virtualization: Event Channel 2/2



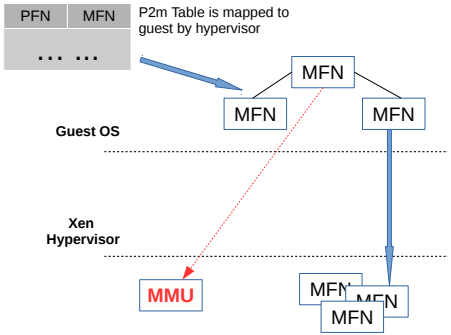
Xen Memory Virtualization 1/2

- Address Types
  - GVA (Guest Virtual Address)
  - GPA (Guest Physical Address) or GFN (Guest page Frame Number)
  - HPA (Host Physical Address) or MFN (Machine page Frame Number)
- **Hardware-assisted Memory Virtualization** (Method 1/3): Second-Level Page Table
  - : Intel: Extended Page Table (EPT)
  - : AMD: Nested Page Table (NPT)

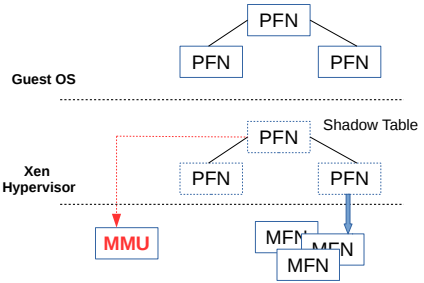


Xen Memory Virtualization 2/2

- **Direct Paging** (Method 2/3): guest manage the (GVA, HPA) page table directly
- **Shadow Paging** (Method 3/3): xen hypervisor maintains a shadow (GVA, HPA) page table which is not awared by guest



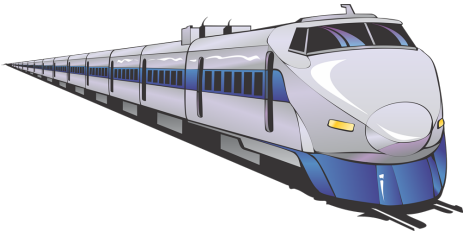
Direct Paging (MMU Paravirtualization)



Shadow Page Table

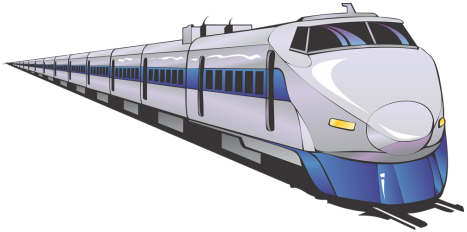
Xen Device Virtualization

- HVM emulated legacy device (QEMU)



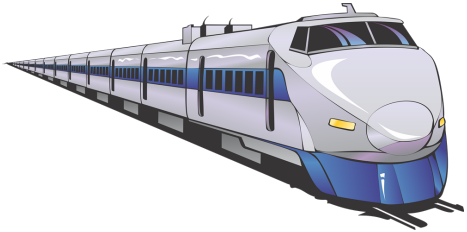
Xen Device Virtualization

- HVM emulated legacy device (QEMU)
- Paravirtual (PV) drivers



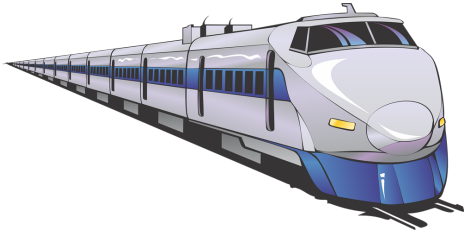
Xen Device Virtualization

- HVM emulated legacy device (QEMU)
- Paravirtual (PV) drivers
- Device Passthrough (vt-d)



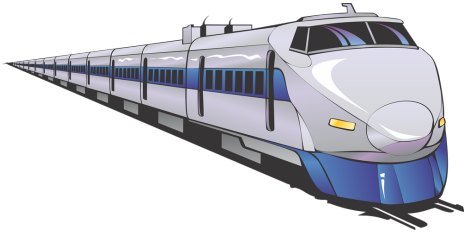
Xen Device Virtualization

- HVM emulated legacy device (QEMU)
- Paravirtual (PV) drivers
- Device Passthrough (vt-d)
- Virtual Function (vt-d)



Xen Device Virtualization

- HVM emulated legacy device (QEMU)
- **Paravirtual (PV) drivers**
- Device Passthrough (vt-d)
- Virtual Function (vt-d)



PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	
device discovery	PCI Tree	
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	

PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	

PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	

PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	

PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	
interrupt	IOAPIC, MSI, MSI-X	

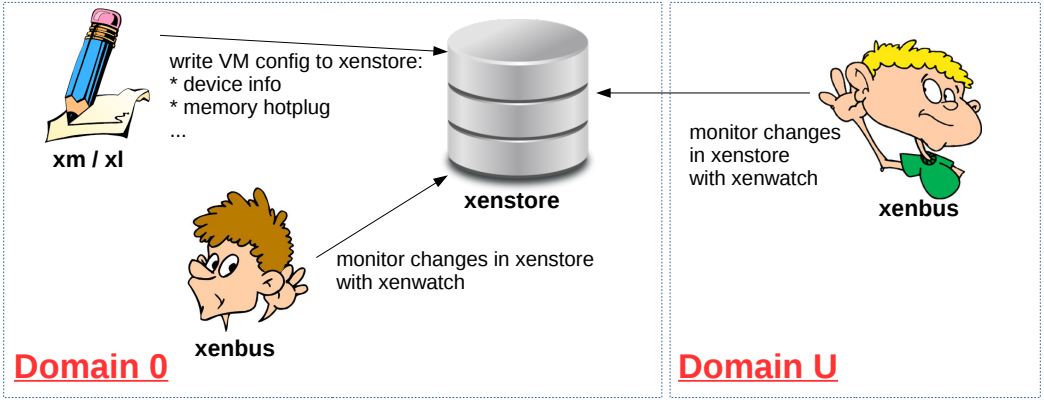
PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	Grant Table
interrupt	IOAPIC, MSI, MSI-X	

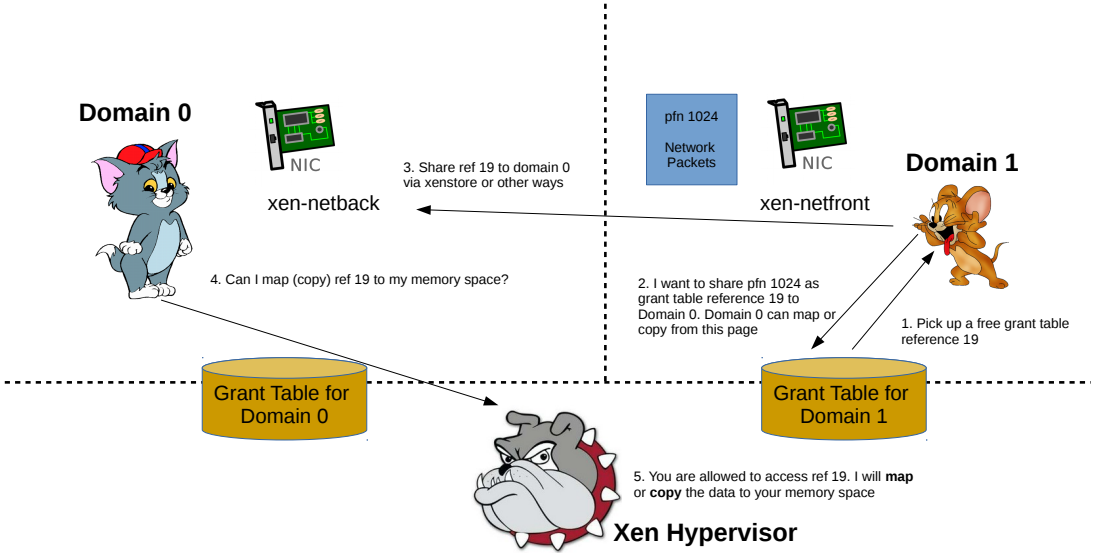
PV driver vs. PCI driver

	PCI driver	PV driver
device abstraction	pci_device, pci_driver	xenbus_device, xenbus_driver
device discovery	PCI Tree	Xenstore
device configuration	PCI Config Space (IO/MMIO)	Xenstore
data flow	DMA Ring Buffer	Memory Ring Buffer
shared memory	N/A or IOMMU	Grant Table
interrupt	IOAPIC, MSI, MSI-X	Event Channel

Xenstore/Xenbus

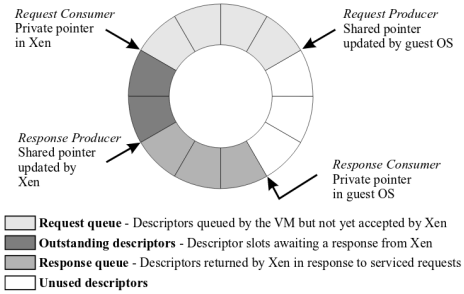


Grant Table

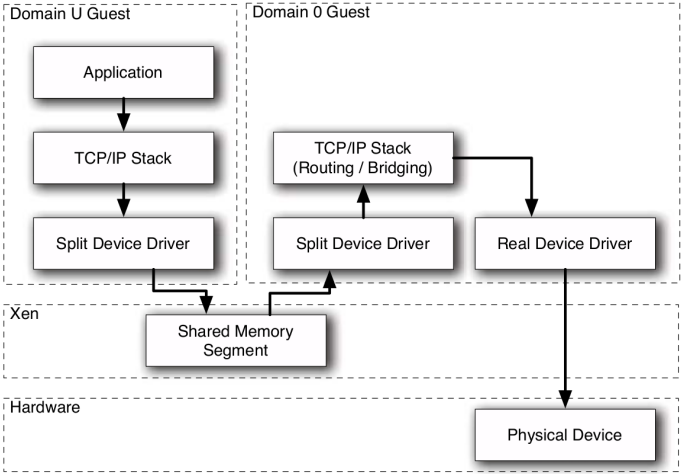

Domain 0
NIC
xen-netback
3. Share ref 19 to domain 0 via xenstore or other ways
pfn 1024
Network Packets
NIC
xen-netfront
Domain 1
4. Can I map (copy) ref 19 to my memory space?
2. I want to share pfn 1024 as grant table reference 19 to Domain 0. Domain 0 can map or copy from this page
1. Pick up a free grant table reference 19
Grant Table for Domain 0
Grant Table for Domain 1
5. You are allowed to access ref 19. I will **map** or **copy** the data to your memory space
Xen Hypervisor

I/O Ring Buffer

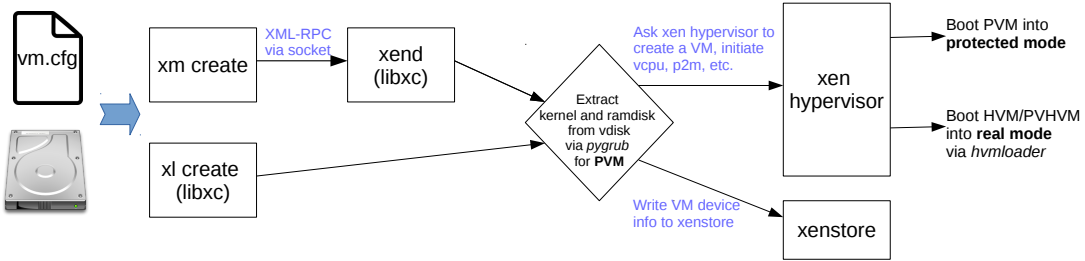
- Usually put grant ref (not data) in ring
- Grant ref of ring pages are shared via xenstore
- Usually one ring buffer for each device queue
- One or more pages for each ring
- Producer and Consumer (barrier)

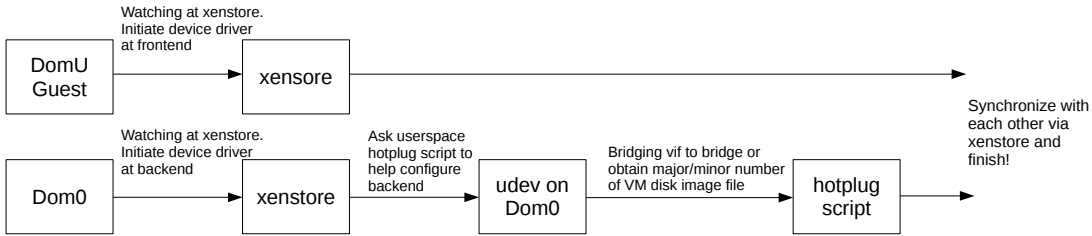


Xen Paravirtual Networking Framework



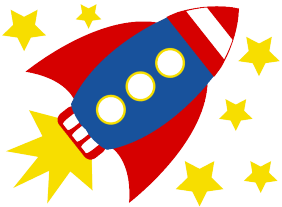
VM Creation Workflow





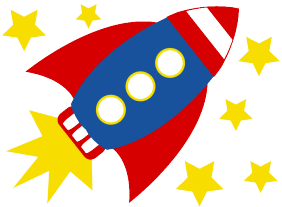
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping



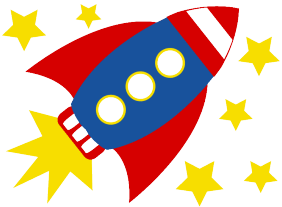
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch



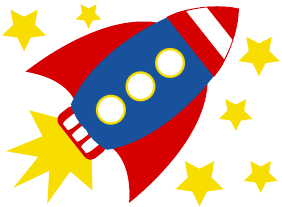
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch
- Stealthy monitoring with Xen altp2m



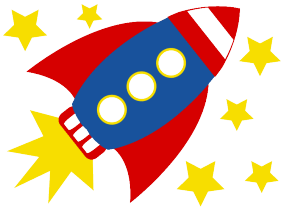
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch
- Stealthy monitoring with Xen altp2m
- Real-Time-Deferrable-Server(RTDS) CPU Scheduler



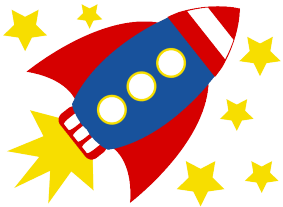
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch
- Stealthy monitoring with Xen altp2m
- Real-Time-Deferrable-Server(RTDS) CPU Scheduler
- Windows PV Receive Side Scaling



Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch
- Stealthy monitoring with Xen altp2m
- Real-Time-Deferrable-Server(RTDS) CPU Scheduler
- Windows PV Receive Side Scaling
- More at Xen Summit and xen-devel



Reference

Publications

- Xen and the art of virtualization. Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. SOSP 2003
- The Definitive Guide to the Xen Hypervisor. David Chisnall. 2007
- Intel 64 and IA-32 Architectures Software Developer Manuals
- Various system & security research paper and presentation

Miscellaneous

- Xen Project Developer Summit
- https://blog.xenproject.org
- https://github.com/finallyjustice/JOS-vmx

Take-Home Message

- What is virtualization

Take-Home Message

- What is virtualization
- Paravirtualization and Hardware-assisted Virtualization

Take-Home Message

- What is virtualization
- Paravirtualization and Hardware-assisted Virtualization
- Xen vs. KVM

Take-Home Message

- What is virtualization
- Paravirtualization and Hardware-assisted Virtualization
- Xen vs. KVM
- Grant Table, Event Channel, Paravirtual Drivers