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

When discussing virtualization...

1) CPU Virtualization?
2) Memory Virtualization?
3) Device Virtualization?
Virtualization

Xen Virtualization

When discussing virtualization ...
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 **Privileged Mode**

```
Guest OS + Applications

Page Fault

MMU Emulation

CPU Emulation

IRQ Emulation

vIRQ

Virtual Machine Monitor

Unprivileged

Privileged

Dongli Zhang (Oracle)  
Xen is not just paravirtualization  
December 16, 2016  4 / 30
```
Virtualizable Architecture: all **sensitive instructions** must also be **privileged instructions** (by Gerald J. Popek and Rebert P. Goldberg)

**critical instructions** = **sensitive instructions** − **privileged instructions**
- 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 Robert 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

<table>
<thead>
<tr>
<th>Guest Code</th>
<th>Translation Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mov ebx, eax</code></td>
<td><code>mov ebx, eax</code></td>
</tr>
<tr>
<td><code>cli</code></td>
<td><code>call HANDLE_CLI</code></td>
</tr>
<tr>
<td><code>and ebx, ~0xffff</code></td>
<td><code>and ebx, ~0xffff</code></td>
</tr>
<tr>
<td><code>mov ebx, cr3</code></td>
<td><code>mov [CO_ARG], ebx</code></td>
</tr>
<tr>
<td><code>sti</code></td>
<td><code>call HANDLE_CR3</code></td>
</tr>
<tr>
<td><code>ret</code></td>
<td><code>call HANDLE_STI</code></td>
</tr>
<tr>
<td></td>
<td><code>jmp HANDLE_RET</code></td>
</tr>
</tbody>
</table>

- vPC

- start
Solution 2/3: Hardware Virtualization (Intel VT)

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

KVM (Kernel-based Virtual Machine)
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

![Diagram showing the relationship between user, kernel, xen, and hypervisor modes.](image)

Dongli Zhang (Oracle) - Xen is not just paravirtualization - December 16, 2016
State of the Art Virtualization

- Binary Translation (QEMU, Bochs, VMWare)

Dongli Zhang (Oracle)
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 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

Domain 0
- xm
- xl
- xenstore
- QEMUs

PVM
- Legacy Device Drivers
- Backend PV Drivers
- Frontend PV Drivers

HVM
- Legacy Device Drivers

PVHVM
- Frontend PV Drivers

Xen Hypervisor
- Memory Management
- CPU Virtualization
- Timer Virtualization
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 #ifdef CONFIG_X86_32
106 ELFNOTE(Xen, Xen_ELFOOT_GUEST_OS, .asciz "Linux")
107 ELFNOTE(Xen, Xen_ELFOOT_GUEST_VERSION, .asciz "2.6")
108 ELFNOTE(Xen, Xen_ELFOOT_XEN_VERSION, .asciz "xen-3.6")
109 #ifdef CONFIG_X86_32
110 ELFNOTE(Xen, Xen_ELFOOT_VIRT_BASE, __ASM_PTR __PAGE_OFFSET)
111 #else
112 ELFNOTE(Xen, Xen_ELFOOT_VIRT_BASE, __ASM_PTR __START_KERNEL_map)
113 /* Map the p2m table to a 512GB-aligned user address. */
114 ELFNOTE(Xen, Xen_ELFOOT_INIT_P2M, .quad PGDER_SIZE)
115 #endif
116 ELFNOTE(Xen, Xen_ELFOOT_ENTRY, __ASM_PTR startup_xen)
117 ELFNOTE(Xen, Xen_ELFOOT_HYPERCALL_PAGE, __ASM_PTR hypercall_page)
118 ELFNOTE(Xen, Xen_ELFOOT_FEATURES, .asci "writable page tables\npage tables\nlong (PVH_FEATURES) 1\n(R \nXENFEAT_writable_table\n1 \nXENFEAT_DOM)\nXENFEAT_pae\nXENFEAT_loader\nXENFEAT_LI_MFN\n1\nquad \nPAGE_PRESENT\nquad \nPAGE_PRESENT\nXENFEAT_HV\nXENFEAT_MOD\nXENFEAT_HYPERVISOR\nXENFEAT_PADDR\n
119 #endif
120 ELFNOTE(Xen, Xen_ELFOOT_PAE_MODE, .asci "yes")
121 ELFNOTE(Xen, Xen_ELFOOT_LOADER, .asci "generic")
122 ELFNOTE(Xen, Xen_ELFOOT_LI_MFN_VALID,
123 .quad \PAGE_PRESENT, \quad \PAGE_PRESENT
124 ELFNOTE(Xen, Xen_ELFOOT_SUSPEND_CANCEL, .long 1)
125 ELFNOTE(Xen, Xen_ELFOOT_MOD_START_PFN, .long 1)
126 ELFNOTE(Xen, Xen_ELFOOT_HV_START_LOW, __ASM_PTR __HYPERVISOR_VIRT_START)
127 ELFNOTE(Xen, Xen_ELFOOT_PADDR_OFFSET, __ASM_PTR 0)
128 #endif
```

```
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_EVICTN=m
CONFIG_XEN_BACKEND=y
CONFIG_XEN_BUS_FRONTEND=y
CONFIG_XEN_GNTDEV=m
CONFIG_XEN_GRANT_DEV_ALLOC=m
CONFIG_XEN_TMEM=m
CONFIG_XEN_PCIEDEV_BACKEND=m
CONFIG_XEN_PRIVCMD=m
```
The Paravirtualization Spectrum

<table>
<thead>
<tr>
<th></th>
<th>Disk / Network</th>
<th>Interrupts, Timers</th>
<th>Emulated Motherboard</th>
<th>Privileged Instructions and Pagetables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>V</strong> Virtualized</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td><strong>P</strong> Paravirtualized</td>
<td>P</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td><strong>Full Virtualization (FV)</strong></td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td><strong>FV with PV disk, network</strong></td>
<td>P</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td><strong>PVHVM</strong></td>
<td>P</td>
<td>P</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td><strong>PVH</strong></td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>V</td>
</tr>
<tr>
<td><strong>Full Paravirtualized (PV)</strong></td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>
Xen CPU Virtualization

- vcpu \approx \text{task_struct}
- domain \approx \text{container or process group}
- xen schedules vcpu

\[ \text{xen schedules vcpu} \]

\[ \begin{align*}
\text{user} & \quad \text{ring 3} \\
\text{kernel} & \quad \text{ring 1} \\
\text{xen} & \quad \text{ring 0}
\end{align*} \]

1. set a per-domain system call handler when the domain gets scheduled
2. system call
3. Trap to and handled in guest kernel

x86 32-bit pvm

\[ \begin{align*}
\text{user} & \quad \text{ring 3} \\
\text{kernel} & \quad \text{ring 3} \\
\text{xen} & \quad \text{ring 0}
\end{align*} \]

1. system call
2. Route to guest kernel system call handler
3. Handled in guest kernel directly

x86 64-bit pvm

\[ \begin{align*}
\text{user} & \quad \text{ring 3} \\
\text{kernel} & \quad \text{ring 3} \\
\text{xen} & \quad \text{ring 0}
\end{align*} \]

1. system call
2. Trap to and handled in guest kernel directly
3. Handled in guest kernel directly

x86 vt-x hvm/pvhvm

Dongli Zhang (Oracle)
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

**Domain 0**
- **xen_evtchn_do_upcall** will traverse and handle each pending event
- set eip to **xen_hypervisor_callback** during scheduling if vcpu has pending event

**PVM**
- **xen_evtchn_do_upcall** will traverse and handle each pending event
- set eip to **xen_hypervisor_callback** during scheduling if vcpu has pending event

**HVM**
- Guest will handle interrupt as native machine
- Intel-vt based interrupt injection and one vector for each irq

**PVHVM**
- **xen_evtchn_do_upcall** will traverse and handle each pending event
- IRQ handler for vector 0xf3 is called
- Intel-vt based interrupt injection and vector 0xf3 for each event

**Xen Hypervisor**

Dongli Zhang (Oracle)
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)
**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 aware by guest

---

**Direct Paging (MMU Paravirtualization)**

- Guest OS
- Xen Hypervisor
- PFN
- MFN

**Shadow Page Table**

- Guest OS
- Xen Hypervisor
- PFN
- MFN

P2m Table is mapped to guest by hypervisor
HVM emulated legacy device (QEMU)
- HVM emulated legacy device (QEMU)
- Paravirtual (PV) drivers
Xen Device Virtualization

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

<table>
<thead>
<tr>
<th></th>
<th>PCI driver</th>
<th>PV driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>device abstraction</td>
<td>pci_device, pci_driver</td>
<td></td>
</tr>
<tr>
<td>device discovery</td>
<td>PCI Tree</td>
<td></td>
</tr>
<tr>
<td>device configuration</td>
<td>PCI Config Space (IO/MMIO)</td>
<td></td>
</tr>
<tr>
<td>data flow</td>
<td>DMA Ring Buffer</td>
<td></td>
</tr>
<tr>
<td>shared memory</td>
<td>N/A or IOMMU</td>
<td></td>
</tr>
<tr>
<td>interrupt</td>
<td>IOAPIC, MSI, MSI-X</td>
<td></td>
</tr>
</tbody>
</table>
## PV driver vs. PCI driver

<table>
<thead>
<tr>
<th></th>
<th>PCI driver</th>
<th>PV driver</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>device abstraction</strong></td>
<td>pci_device, pci_driver</td>
<td>xenbus_device, xenbus_driver</td>
</tr>
<tr>
<td><strong>device discovery</strong></td>
<td>PCI Tree</td>
<td></td>
</tr>
<tr>
<td><strong>device configuration</strong></td>
<td>PCI Config Space (IO/MMIO)</td>
<td></td>
</tr>
<tr>
<td><strong>data flow</strong></td>
<td>DMA Ring Buffer</td>
<td></td>
</tr>
<tr>
<td><strong>shared memory</strong></td>
<td>N/A or IOMMU</td>
<td></td>
</tr>
<tr>
<td><strong>interrupt</strong></td>
<td>IOAPIC, MSI, MSI-X</td>
<td></td>
</tr>
</tbody>
</table>
### PV driver vs. PCI driver

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

Dongli Zhang (Oracle)  
Xen is not just paravirtualization  
December 16, 2016
## PV driver vs. PCI driver

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

Dongli Zhang (Oracle)

Xen is not just paravirtualization

December 16, 2016
## PV driver vs. PCI driver

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

Dongli Zhang (Oracle)  
Xen is not just paravirtualization  
December 16, 2016
### PV driver vs. PCI driver

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

Dongli Zhang (Oracle)  

Xen is not just paravirtualization  

December 16, 2016
Xenstore/Xenbus

- **Domain 0**
  - `xm / xl`
  - `xenbus`
  - Monitor changes in xenstore with xenwatch

- **Domain U**
  - `xenbus`
  - Monitor changes in xenstore with xenwatch

- **Xen Hypervisor**

- Write VM config to xenstore:
  - * device info
  - * memory hotplug
  - ...

- Monitor changes in xenstore with xenwatch
1. Pick up a free grant table reference 19
2. I want to share pfn 1024 as grant table reference 19 to Domain 0. Domain 0 can map or copy from this page
3. Share ref 19 to domain 0 via xenstore or other ways
4. Can I map (copy) ref 19 to my memory space?
5. You are allowed to access ref 19. I will map or copy the data to your memory space
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

- **xm create vm.cfg**
- **xl create (libxc)**
- **xend (libxc)**
- **xen hypervisor**
- **xenstore**

**DomU Guest**

- **xm create** via socket
- **xend**
- **xen hypervisor**
- **xenstore**

**Dom0**

- **udev on Dom0**
- **hotplug script**

- **Ask xen hypervisor to create a VM, initiate vcpu, p2m, etc.**
- **Extract kernel and ramdisk from vdisk via pygrub for PVM**
- **Write VM device info to xenstore**
- **Boot PVM into protected mode**
- **Boot HVM/PVHVM into real mode via hvmloader**

**Watching at xenstore**

- **Initiate device driver at frontend**
- **Initiate device driver at backend**

**Synchronize with each other via xenstore and finish!**
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
Selected Xen Projects

- COLO - Coarse Grain Lock Stepping
- LivePatch

Dongli Zhang (Oracle)

Xen is not just paravirtualization

December 16, 2016 28 / 30
Selected Xen Projects

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

Dongli Zhang (Oracle)
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

Dongli Zhang (Oracle)
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
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