VT-d and SR-IOV Update

Feb-24-2009

Allen Kay

Eddie Dong

Xen Summit Oracle 2009

Agenda

- VT-d Update
  - Current Status
  - VT-d2 Queued Invalidation and Interrupt Remapping
  - Xm tools for PCI passthrough support
  - Assignable Device Filtering Description
  - Work In Progress/Planned: multi-PCI segment, ATS, NUMA
- SR-IOV Update
  - Current Status
  - Use flow

VT-d: Current Status

- VT-d1 was checked into Xen 3.2 release
  - HVM PCI passthrough
  - DMA remapping
- VT-d2 major features were checked into Xen 3.3 release
  - Queued Invalidation
  - Interrupt Remapping

VT-d2 Queue Invalidation

- VT-d1 does IOTLB and Context entry flush by setting a VT-d command register and polling on the status register
- Queued Invalidation allows invalidation commands to be queued in 1 or more pages
  - Default uses 1 page defined by a #define
- New capabilities include flushing of cached interrupt remapping entries and device translation cache
- Auto enable if HW feature is detected
  - Initialize invalidation function table with register based invalidation functions
  - If Queued Invalidation HW is detected in VT-d capability register, replace function table with Queued Invalidation functions

VT-d2 Interrupt Remapping Description

- It is a security and isolation feature
- It does not enable direct interrupt injection to a guest
- Without this feature, malicious guest can attack a host by:
  - Generating interrupts by setting up passthrough device to do DMA write transactions to the APIC 0xFEExxxxx region
- With interrupt remapping source ID checking enabled, interrupt generation is checked against PCI BDF in interrupt remapping table entry

VT-d2 Interrupt Remapping Implementation

- Low level IO_APIC_WRITE/IO_APIC_READ macros are modified to VT-d IOAPIC read/write functions
- If interrupt remapping HW is found, all IOAPIC RTE entries are converted to interrupt remap format
- Interrupt remap entries are allocated for each IOAPIC RTE and appropriate fields are initialized
- Similar modification were done for remapping MSI interrupts

PCI Passthrough Management Tools

- xm pci-list-assignable-devices
  - List all assignable devices
- xm pci-attach
  - Hot add a passthrough device to a domain
- xm pci-dettach
  - Hot remove a passthrough device from a domain
- xm pci-list
  - List pass-through pci device for a domain

Assignable Device Filtering Description

- Definition of terms
  - co-assigned-devices: devices that requires to be assigned to the same domain.
- There are two types of co-assigned devices
  - PCI devices behind the same PCI/PCI-x bridge - as defined in VT-d spec.
  - multi-funtion PCIe devices with no FLR capability. This is because we use SecondaryBusReset in place of FLR.

Assignable Device Filtering (cont)

- Algorithm for "xm pci-list-assignable-device"
- Find all the devices owned by pciback
- Prune the assignable device list as follows
  - For multi-function PCIe devices with no FLR capability, if any of its co-assigned-devices not owned by pciback then all the devices are not assignable.
  - If any device behind the same PCI/PCI-X bridge is not owned by pciback then all devices behind this bridge is not assignable.
  - For devices with non-page-aligned MMIO BAR, it and all its co-assigned-devices are not assignable.
  - For devices has already been assigned to a guest, it and all its co-assigned-devices are not assignable.
- Print out assignable devices. Co-assigned-devices are displayed on the same line.

Work In Progress: Multi-PCI Segment

- ACPI table reports device scope with PCI segment number in addtion to bus, device, function numbers
- Current VT-d code in Xen hypervisor does not comprehend PCI segment number
- Need to add PCI segment to pci_dev structure
- Incorporate PCI segment number when locating a VT-d engine serving a particular device
- Add segment to pci_add_device hypercall
- Control panel commands that uses BDF information, add segment as an optional field
  - xm pci-list/attach/detach/list-assignable-devices
  - PCI field in /etc/xen/hvm.conf
  - pciback.hide field in pciback driver

Work In Progress: ATS

- Address Translation Service (ATS) is defined for keeping VT-d IOTLB and device translation cache in sync
- VT-d2 queued invalidation has defined a command for invalidating device translation cache
- Changes needed:
  - Detect ATS capable root ports using ACPI ATS reporting table
  - Detect ATS capable devices
  - Turn on ATS on the device if it is under ATS capable root port
- Original patch required PCI mmcfg support in xen for parsing ATS capability in PCIe Extended Config Space
- Looking into leveraging dom0 for this

Work Planned: VT-d NUMA Support

- ACPI table reports Remapping Hardware Static Affinity Structure (RHSA)
- Associates VT-d remapping hardware to proximity domains
- Need to allocate remapping table structures base on the this proximity domain instead of the CPU (which can migrate)
- This means context table and page table for a particular VT-d HW should be allocated with associated proximity domain

SR-IOV Status

- Linux kernel changes are being pushed to upstream kernel
  - Latest patch is version 10 (architecturally stable)
  - Patches for Xen 2.6.18 submitted to xen mailing list in Sep-10-08
  - Will rebase after upstream kernel's acceptance
- Patch presents SR-IOV virtual functions as regular PCI devices
- Passthrough of SR-IOV VF's is the same as PCI devices
- No SR-IOV specific changes required in Xen and QEMU
- To enable SR-IOV in upstream Xen
  - 2.6.18 dom0 patch (published xen-devel in Sep-10-08)
  - 82576 PF and VF drivers (published to netdev & LKML on Feb-19-09)

SR-IOV Use Flow

- Boot dom0 with PF driver
- "echo 7 > /sys/class/net/ethx/num_vfs"
  - PF will allocates 7 VFs on the NIC
  - PCI subsystem instantiates a PCI BDF for each VF
- Bind passthrough VF's BDF to pciback
  - "echo -n 0000:05:02.0 > /sys/bus/pci/driver/pciback/new_slot"
  - "echo -n 0000:05:02.0 > /sys/bus/pci/driver/pciback/bind"
- VF's BDF can then be used to passthrough to the guest in hvm.conf same as regular VT-d PCI passthrough devices
  - pci = ['05:02.0']
- During guest boot, VF driver generates a random MAC address which is unique on the system
- Can also manually set MAC address by
  - Echo "00:01:02:aa:bb:cc" > /sys/class/net/ethx/VFn/cfg/macaddr

Challenges

- HVM guest memory swapping implications
  - Current VT-d code cannot handle HVM memory swapping
  - No PCI devices supporting IO page fault restart yet
- Live migration
  - Teaming up with virtual NIC
  - Hot add/remove passthrough device in guest

Questions?

Backup

VT-d PCI Passthrough Overiew

- PCI Passthrough
  - Develop software to allow PCI device assignment to HVM guest domain using PCI Bus/Dev/Func address
- DMA Remapping with VT-d
  - Develop software to enumeration VT-d engine via ACPI static table, initialize VT-d HW for Dom0, ability to change VT-d HW structures to assign PCI devices to different guest domains

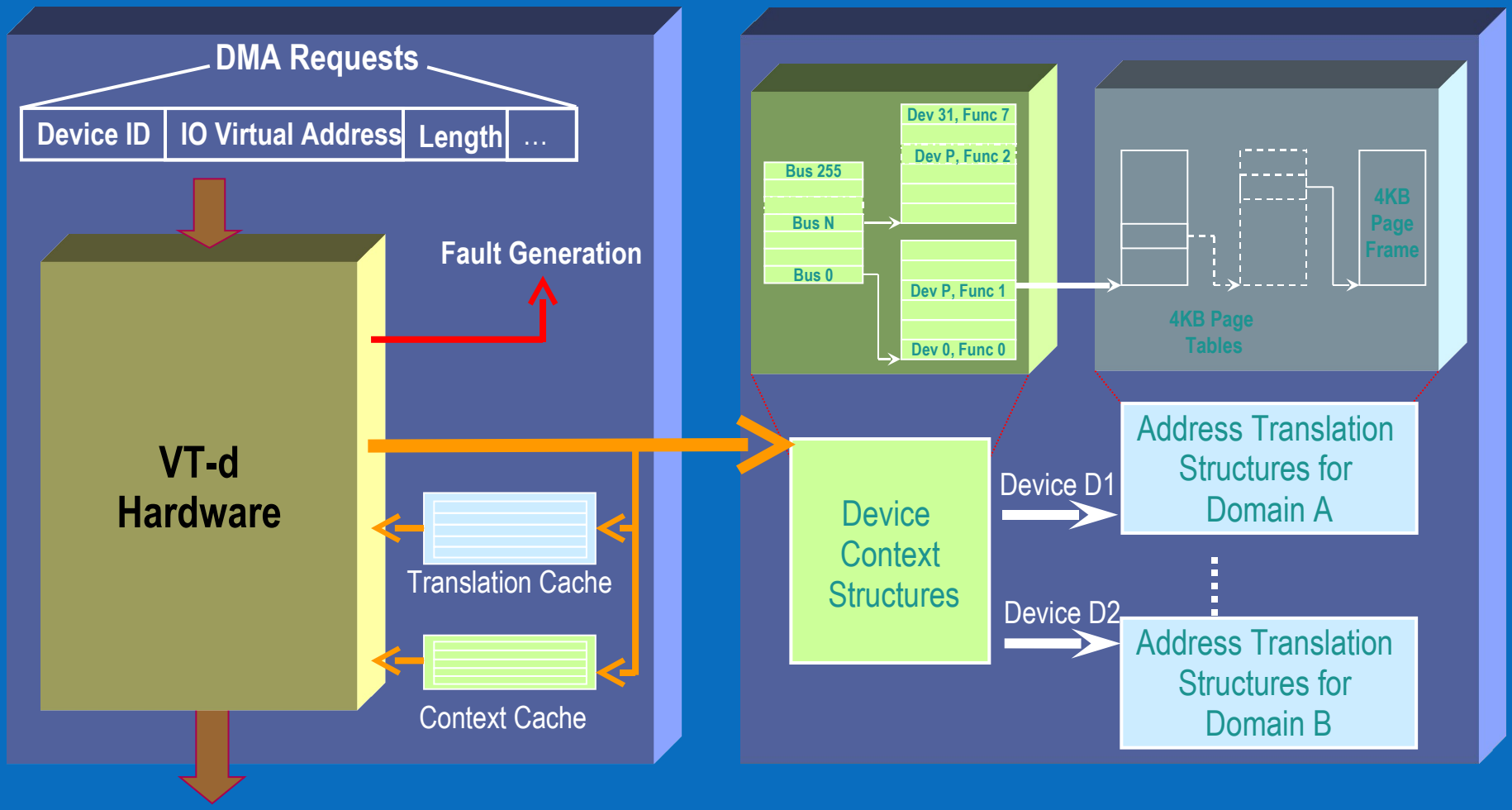
PCI Passthrough: Interrupt Handling

- VMM gets physical interrupt from assigned device
- Interrupt is injected to vIOAPIC device model
- vIOAPIC propagates interrupt to vLAPIC
- VMM checks vLAPIC for pending interrupts before it returns to the guest
- If there is pending interrupt in vLAPIC, interrupt is injected to the guest via VMCS

PCI Passthrough Details

- Hide PCI device from Dom0 device driver with OS boot load option
- Assign PCI devices to guest with xen guest config file
  - /etc/xen/hvm.conf
- Attach the assigned device to QEMU vPCI bus
- Assign corresponding VT-d context entry to guest domain
- Intercept PCI config access in Qemu – command register accesses are passed on to HW
- Emulate IO port access in Xen
- Install P2M entry for MMIO access of the assigned device
- Xen intercepts physical device interrupts and re-injects to the target guest domain with vIRQ

VT-d : Hardware Overview



Memory Access with Host Physical Address

Memory-resident IO Partitioning & Translation Structures

VT-d HW Programming Details

- Enumerate ACPI table for VT-d hardware
- After construct_dom0() in setup.c:
  - Build VT-d page tables base on domain's physical pages
  - Allocate a page for VT-d context-entry table and initialize context entries with valid PCI devices to point to the same dom0 VT-d page table structure
  - For each VT-d engine, allocate a page for Root-entry table and initialize corresponding context entries as devices are initialized in dom0
  - Create identity mapping for VT-d reserved memories (RMRR) in VT-d page table
  - Flush VT-d context and TLB caches
  - Enable VT-d translation
- VT-d HW changes needed to assign PCI devices to guest domain
  - Build VT-d IO page table from guest domain page list
  - Change corresponding context entry to guest domain ID and point to guest's IO page table structure

Resources

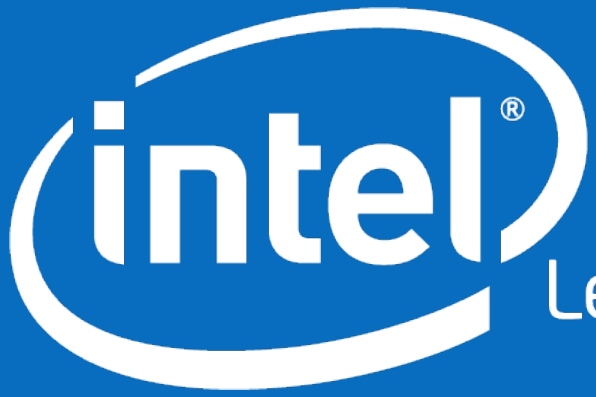
- VT-d specification:
  - http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf
- Xen VT-d wiki:
  - http://wiki.xensource.com/xenwiki/VTdHowTo
- SR-IOV Specification:
  - http://www.pcisig.com/members/downloads/specifications/iov/sr-iov1.0_11Sep07.pdf
- ATS 1.1 Specification:
  - http://www.pcisig.com/members/downloads/specifications/iov/ats_r1.1_22Apr08.pdf

Resource: Contacts

- VT-d Contacts
  - Allen.m.kay@intel.com
  - Weidong.han@intel.com
- SR-IOV Contacts
  - Eddie.dong@intel.com
  - yu.zhao@intel.com

Legal Information

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.
- Intel may make changes to specifications, product descriptions, and plans at any time, without notice.
- All dates provided are subject to change without notice.
- Intel is a trademark of Intel Corporation in the U.S. and other countries.
- *Other names and brands may be claimed as the property of others.
-



Leap ahead™

VT-d and SR-IOV Update

Feb-24-2009

Allen Kay

Eddie Dong

Xen Summit Oracle 2009

Agenda

- VT-d Update
  - Current Status
  - VT-d2 Queued Invalidation and Interrupt Remapping
  - Xm tools for PCI passthrough support
  - Assignable Device Filtering Description
  - Work In Progress/Planned: multi-PCI segment, ATS, NUMA
- SR-IOV Update
  - Current Status
  - Use flow

VT-d: Current Status

- VT-d1 was checked into Xen 3.2 release
  - HVM PCI passthrough
  - DMA remapping
- VT-d2 major features were checked into Xen 3.3 release
  - Queued Invalidation
  - Interrupt Remapping

VT-d2 Queue Invalidation

- VT-d1 does IOTLB and Context entry flush by setting a VT-d command register and polling on the status register
- Queued Invalidation allows invalidation commands to be queued in 1 or more pages
  - Default uses 1 page defined by a #define
- New capabilities include flushing of cached interrupt remapping entries and device translation cache
- Auto enable if HW feature is detected
  - Initialize invalidation function table with register based invalidation functions
  - If Queued Invalidation HW is detected in VT-d capability register, replace function table with Queued Invalidation functions

VT-d2 Interrupt Remapping Description

- It is a security and isolation feature
- It does not enable direct interrupt injection to a guest
- Without this feature, malicious guest can attack a host by:
  - Generating interrupts by setting up passthrough device to do DMA write transactions to the APIC 0xFEExxxxx region
- With interrupt remapping source ID checking enabled, interrupt generation is checked against PCI BDF in interrupt remapping table entry

VT-d2 Interrupt Remapping Implementation

- Low level IO_APIC_WRITE/IO_APIC_READ macros are modified to VT-d IOAPIC read/write functions
- If interrupt remapping HW is found, all IOAPIC RTE entries are converted to interrupt remap format
- Interrupt remap entries are allocated for each IOAPIC RTE and appropriate fields are initialized
- Similar modification were done for remapping MSI interrupts

PCI Passthrough Management Tools

- xm pci-list-assignable-devices
  - List all assignable devices
- xm pci-attach
  - Hot add a passthrough device to a domain
- xm pci-dettach
  - Hot remove a passthrough device from a domain
- xm pci-list
  - List pass-through pci device for a domain

Assignable Device Filtering Description

- Definition of terms
  - co-assigned-devices: devices that requires to be assigned to the same domain.
- There are two types of co-assigned devices
  - PCI devices behind the same PCI/PCI-x bridge - as defined in VT-d spec.
  - multi-funtion PCIe devices with no FLR capability. This is because we use SecondaryBusReset in place of FLR.

Assignable Device Filtering (cont)

- Algorithm for "xm pci-list-assignable-device"
- Find all the devices owned by pciback
- Prune the assignable device list as follows
  - For multi-function PCIe devices with no FLR capability, if any of its co-assigned-devices not owned by pciback then all the devices are not assignable.
  - If any device behind the same PCI/PCI-X bridge is not owned by pciback then all devices behind this bridge is not assignable.
  - For devices with non-page-aligned MMIO BAR, it and all its co-assigned-devices are not assignable.
  - For devices has already been assigned to a guest, it and all its co-assigned-devices are not assignable.
- Print out assignable devices. Co-assigned-devices are displayed on the same line.

Work In Progress: Multi-PCI Segment

- ACPI table reports device scope with PCI segment number in addtion to bus, device, function numbers
- Current VT-d code in Xen hypervisor does not comprehend PCI segment number
- Need to add PCI segment to pci_dev structure
- Incorporate PCI segment number when locating a VT-d engine serving a particular device
- Add segment to pci_add_device hypercall
- Control panel commands that uses BDF information, add segment as an optional field
  - xm pci-list/attach/detach/list-assignable-devices
  - PCI field in /etc/xen/hvm.conf
  - pciback.hide field in pciback driver

Work In Progress: ATS

- Address Translation Service (ATS) is defined for keeping VT-d IOTLB and device translation cache in sync
- VT-d2 queued invalidation has defined a command for invalidating device translation cache
- Changes needed:
  - Detect ATS capable root ports using ACPI ATS reporting table
  - Detect ATS capable devices
  - Turn on ATS on the device if it is under ATS capable root port
- Original patch required PCI mmcfg support in xen for parsing ATS capability in PCIe Extended Config Space
- Looking into leveraging dom0 for this

Work Planned: VT-d NUMA Support

- ACPI table reports Remapping Hardware Static Affinity Structure (RHSA)
- Associates VT-d remapping hardware to proximity domains
- Need to allocate remapping table structures base on the this proximity domain instead of the CPU (which can migrate)
- This means context table and page table for a particular VT-d HW should be allocated with associated proximity domain

SR-IOV Status

- Linux kernel changes are being pushed to upstream kernel
  - Latest patch is version 10 (architecturally stable)
  - Patches for Xen 2.6.18 submitted to xen mailing list in Sep-10-08
  - Will rebase after upstream kernel's acceptance
- Patch presents SR-IOV virtual functions as regular PCI devices
- Passthrough of SR-IOV VF's is the same as PCI devices
- No SR-IOV specific changes required in Xen and QEMU
- To enable SR-IOV in upstream Xen
  - 2.6.18 dom0 patch (published xen-devel in Sep-10-08)
  - 82576 PF and VF drivers (published to netdev & LKML on Feb-19-09)

SR-IOV Use Flow

- Boot dom0 with PF driver
- "echo 7 > /sys/class/net/ethx/num_vfs"
  - PF will allocates 7 VFs on the NIC
  - PCI subsystem instantiates a PCI BDF for each VF
- Bind passthrough VF's BDF to pciback
  - "echo -n 0000:05:02.0 > /sys/bus/pci/driver/pciback/new_slot"
  - "echo -n 0000:05:02.0 > /sys/bus/pci/driver/pciback/bind"
- VF's BDF can then be used to passthrough to the guest in hvm.conf same as regular VT-d PCI passthrough devices
  - pci = ['05:02.0']
- During guest boot, VF driver generates a random MAC address which is unique on the system
- Can also manually set MAC address by
  - Echo "00:01:02:aa:bb:cc" > /sys/class/net/ethx/VFn/cfg/macaddr

Challenges

- HVM guest memory swapping implications
  - Current VT-d code cannot handle HVM memory swapping
  - No PCI devices supporting IO page fault restart yet
- Live migration
  - Teaming up with virtual NIC
  - Hot add/remove passthrough device in guest

Questions?

Open Source
Technology
Center

Software and Solutions Group

16

Backup

Open Source
Technology
Center

Software and Solutions Group

VT-d PCI Passthrough Overiew

- PCI Passthrough
  - Develop software to allow PCI device assignment to HVM guest domain using PCI Bus/Dev/Func address
- DMA Remapping with VT-d
  - Develop software to enumeration VT-d engine via ACPI static table, initialize VT-d HW for Dom0, ability to change VT-d HW structures to assign PCI devices to different guest domains

Open Source Technology Center

PCI Passthrough: Interrupt Handling

- VMM gets physical interrupt from assigned device
- Interrupt is injected to vIOAPIC device model
- vIOAPIC propagates interrupt to vLAPIC
- VMM checks vLAPIC for pending interrupts before it returns to the guest
- If there is pending interrupt in vLAPIC, interrupt is injected to the guest via VMCS

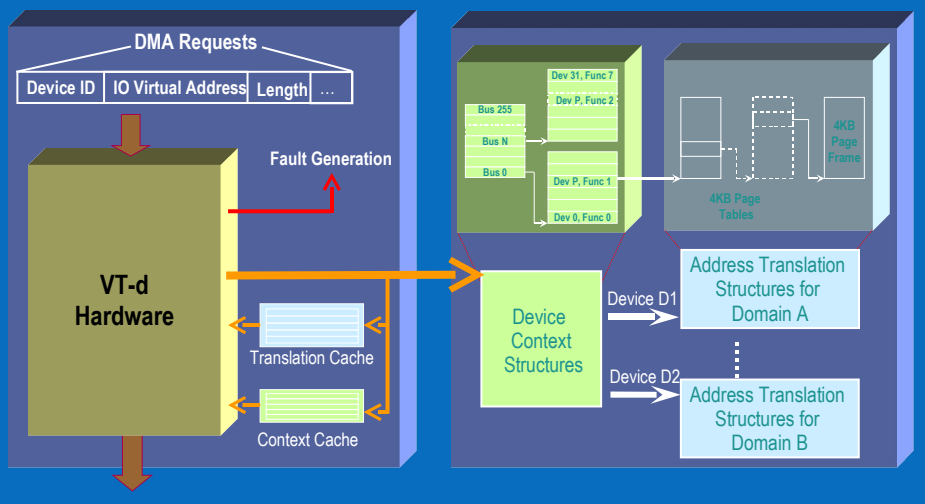
Open Source Technology Center

Software and Solutions Group

PCI Passthrough Details

- Hide PCI device from Dom0 device driver with OS boot load option
- Assign PCI devices to guest with xen guest config file
  - /etc/xen/hvm.conf
- Attach the assigned device to QEMU vPCI bus
- Assign corresponding VT-d context entry to guest domain
- Intercept PCI config access in Qemu – command register accesses are passed on to HW
- Emulate IO port access in Xen
- Install P2M entry for MMIO access of the assigned device
- Xen intercepts physical device interrupts and re-injects to the target guest domain with vIRQ

VT-d : Hardware Overview



Memory Access with Host Physical Address

Memory-resident IO Partitioning & Translation Structures

VT-d HW Programming Details

- Enumerate ACPI table for VT-d hardware
- After construct_dom0() in setup.c:
  - Build VT-d page tables base on domain's physical pages
  - Allocate a page for VT-d context-entry table and initialize context entries with valid PCI devices to point to the same dom0 VT-d page table structure
  - For each VT-d engine, allocate a page for Root-entry table and initialize corresponding context entries as devices are initialized in dom0
  - Create identity mapping for VT-d reserved memories (RMRR) in VT-d page table
  - Flush VT-d context and TLB caches
  - Enable VT-d translation
- VT-d HW changes needed to assign PCI devices to guest domain
  - Build VT-d IO page table from guest domain page list
  - Change corresponding context entry to guest domain ID and point to guest's IO page table structure

Resources

- VT-d specification:
  - http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf
- Xen VT-d wiki:
  - http://wiki.xensource.com/xenwiki/VTdHowTo
- SR-IOV Specification:
  - http://www.pcisig.com/members/downloads/specifications/iov/sr-iov1.0_11Sep07.pdf
- ATS 1.1 Specification:
  - http://www.pcisig.com/members/downloads/specifications/iov/ats_r1.1_22Apr08.pdf

Resource: Contacts

- VT-d Contacts
  - Allen.m.kay@intel.com
  - Weidong.han@intel.com
- SR-IOV Contacts
  - Eddie.dong@intel.com
  - yu.zhao@intel.com

Legal Information

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.
- Intel may make changes to specifications, product descriptions, and plans at any time, without notice.
- All dates provided are subject to change without notice.
- Intel is a trademark of Intel Corporation in the U.S. and other countries.
- *Other names and brands may be claimed as the property of others.
-


intel
Leap ahead
intel
Software
Open Source
Technology
Center
Software and Solutions Group
intel
26