BRING AN INTEL® SCALABLE IOV CAPABLE DEVICE INTO LINUX*

Xin Zeng <xin.zeng@intel.com>
Yi Liu <yi.l.liu@intel.com>
1st, Nov. 2019
Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

© Intel Corporation
Agenda

- Recap Intel® Scalable IOV technology
- The software stack in Linux*
- Develop the device driver in Linux*
- Intel® Scalable IOV & Virtual Shared Virtual Address (vSVA)
- Summary & Opens
Recap Intel® Scalable IOV technology

**Hardware**
- Spec is public by Intel in 2018
- PASID granular DMA isolation
- Finer assignable hardware resource

**Software**
- Composes assignable Virtual Device (VDEV)
- Mediates access to hardware

**Combined together**
- Hardware enforced DMA isolation while keeping scalability and flexibility
What’s new in hardware?

**Device**
- Assignable device interface (ADI)
  - Minimal assignable hardware unit
  - MMIO only
  - Isolated resource
-Interrupt message storage (IMS) for ADI
- Intel® Scalable IOV capability in DVSEC
- PASID capability is required

**Intel® VT-d (IOMMU)**
- Scalable mode support
SW architecture in Linux*

- **Guest**
  - Virtual device (VDEV) driver

- **QEMU**
  - Be agnostic to Intel® Scalable IOV VDEV pass-through

- **Host**
  - Mediated Device Framework
  - Device specific Virtual Device Composition Module (VDCM)
  - IOMMU driver
Design the VDCM

- **Determine the VDEV types**
  - The services VDEV provides

- **Organize the VDEV resources**
  - The virtual config space
  - The virtual bar regions
    - Fast path: e.g. work submission interface
    - Slow path: e.g. config, control
  - The interrupts
    - ADI Interrupt from IMS
    - Virtual interrupt
Design the VDCM (Cont’d)

• Design the VDEV-VDCM communication channel
  • Software based mechanism
    – Memory backed virtual MMIO
  • Hardware based mechanism
    – Mailbox MMIO

• Compose and manage VDEVs
VDEV lifecycle management with VDCM

- **Create**
  - `<Vdev type/uuid>`

- **Assign**

- **Run-time access**

- **Reset**

- **Release**

- **Remove**

Create:
1. Invoke `mdev_set_iommu_device`
2. Request IMS IRQ
3. Allocate ADIs, setup VDEV resources, and compose VDEV
VDEV lifecycle management with VDCM

- **Create**
  - `<Vdev type/uuid>`
- **Assign**
- **Run-time access**
- **Reset**
- **Release**
- **Remove**

 ![Diagram showing the lifecycle management process with open, ioctl, and mmap operations](image-url)
VDEV lifecycle management with VDCM

- **Create**
  - `<Vdev type/uuid>`
- **Assign**
- **Run-time access**
- **Reset**
- **Release**
- **Remove**

VDCM emulates the virtual interrupt and inject it to the VM.
VDEV lifecycle management with VDCM

- **Create**
  - `<Vdev type/uuid>`
- **Assign**
- **Run-time access**
- **Reset**
- **Release**
- **Remove**

Reset:
1. Ensure ADI quiesed
2. Reset ADI and emulation part resource as well.
3. Keep the PASID for ADI unchanged.
VDEV lifecycle management with VDCM

- Create
  - `<Vdev type/uuid>`
- Assign
- Run-time access
- Reset
- Release
- Remove

release:
1. Invalidate pasid in ADI
2. Reset VDEV to ensure no further activities in hardware
3. Clean up VDEV resource allocated after opening.
VDEV lifecycle management with VDCM

- **Create**
  - <Vdev type/uuid>
- **Assign**
- **Run-time access**
- **Reset**
- **Release**
- **Remove**

![Diagram showing VDEV lifecycle management with VDCM]

- **Remove**
  - 1. Remove iommu device from VDEV
  - 2. Release IMS IRQ
  - 3. Release ADIs and the associated resources
  - 4. Any state need to be cleaned up
What's missing? - Discover Intel® scalable IOV capability!

- **Software**
  - Detects Intel® scalable IOV capability
  - Registers parent operations to mdev core driver

- **Hardware**
  - Presents Intel® capability in DVSEC

```
<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Next Capability Offset</th>
<th>Cap Version</th>
<th>PCI Express Extended Capability ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00h</td>
<td>1</td>
<td>0x23</td>
</tr>
<tr>
<td></td>
<td>04h</td>
<td>0</td>
<td>DVSEC Vendor ID = 8086</td>
</tr>
<tr>
<td></td>
<td>08h</td>
<td>None</td>
<td>DVSEC ID for Scalable IOV = 5</td>
</tr>
<tr>
<td></td>
<td>0Ch</td>
<td>Supported Page Sizes (RO)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10h</td>
<td>System Page Size (RW)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14h</td>
<td>Capabilities (RO)</td>
<td></td>
</tr>
</tbody>
</table>
```

**Device** Intel® Scalable IOV Cap

**Device Driver W/ VDCM**

1. detect
2. Register callback, enable IOMMU AUX domain
3. Create entry

**Mdev core driver**

**Sysfs**
Intel® Scalable IOV & vSVA

- Recap vSVA
- VDCM to support vSVA
Recap vSVA

- Shared Virtual Addressing (SVA) is a hardware feature that allows address space sharing between CPU and I/O device for memory access.

- SR-IOV: the generic software (VFIO/IOMMU/QEMU) changes is in community
- Scalable IOV reuses the generic software arch with SR-IOV

SVA in KVM: https://www.youtube.com/watch?v=Kq_nfGK5MwQ
vSVA in Scalable IOV and SR-IOV

- Different DMA transaction types in PCI Express* hardware
  - SVA transaction targets to a MMU managed address space
  - Non-SVA transaction targets to an IOMMU managed address space

memory requests differences in PCIe*

<table>
<thead>
<tr>
<th></th>
<th>SR-IOV</th>
<th>Scalable IOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVA transaction</td>
<td>w/ PASID</td>
<td>w/ PASID</td>
</tr>
<tr>
<td>Non-SVA transaction</td>
<td>w/o PASID</td>
<td>w/ PASID</td>
</tr>
</tbody>
</table>

VDCM should deal with the PASID differences
VDCM to support vSVA

- **PASID Management**
  - **Track the PASIDs**
    - Non-SVA PASID: AUX domain default PASID
VDCM to support vSVA

- **PASID Management**
  - **Track the PASIDs**
    - Non-SVA PASID: AUX domain default PASID
    - SVA PASID: set from guest software
VDCM to support vSVA

- **PASID Management**
  - **Track the PASIDs**
    - Non-SVA PASID: AUX domain default PASID
    - SVA PASID: set from guest software
  - **Switch the PASID for the ADI per guest operation**
    - Enable vSVA or disable vSVA
  - **Do PASID drain/reset**
    - SVA PASID free
VDCM to support vSVA (Cont.)

- **Compose VDEV with SVA related Capabilities**
  - PCI Express* PASID/ATS/PRS capability structures in VDEV configuration space
  - Native software should have enabled the above capabilities

- **Handle IOMMU page fault**
  - Register fault handler to iommu driver
  - Notify user space client
Summary

- Intel® Scalable IOV enforces DMA isolation at PASID granularity
- Intel® Scalable IOV brings more scalability and flexibility
- To develop the VDCM for the PASID granular DMA isolation capable technology like Intel® Scalable IOV
  - Determine your virtual device types
  - Organize your virtual device resource into slow path and fast path
  - Take care the hardware interrupt (IMS in Intel® Scalable IOV) and PASID programming
  - Leverage existing mediated device framework for VDEV management
  - Emulate SVA capabilities stuff and handle IOMMU page fault to support vSVA
QUESTIONS?