Unified I/O Page Table Management for Passthrough Devices

Kevin Tian, Baolu Lu
 Agenda

• A brief background
• Development plan
• Manage security context for user-initiated DMAs
• Miscellaneous opens if time allows
Proposal for a Unified Framework

* Thank Jason Gunthorpe for initiating this idea!
**Key Concept**

1. **Bind**
   - VFIO

2. **DEVICE_GET_INFO**
   - Struct device *(RID)*
   - PASID (optional)
   - User provided cookie

3. **ALLOC_IOASID**
   - iommu
   - Map/unmap DMA
   - Invalidate iotlb
   - Fault handling

4. **Attach**

5. **iommufd**
   - I/O Address Spaces
     - address space id (ioasid)
     - type
     - page table format
     - parent ioasid (optional)
     - list of attached devices
   - Bound device
   - PASID (optional)

**User-managed**

**Kernel-managed**

**Shared (mm, ept, etc.)**

**IOMMU**
Need community collaboration. Many tasks can be done in parallel!
Manage User-initiated DMAs

• Establish a secure context (iommu unmanaged domain) which restricts user-initiated DMAs to
  • process memory
  • sibling devices in the same group

• Guarantee exclusive DMA ownership on the group, i.e. devices in the group must be
  • Bound to the owner driver (e.g. vfio), or
  • Bound to a driver known DMA-safe (e.g. pci-stub), or
  • Driverless

• iommufd can copy what vfio does today, with one exception
  • Need manage multiple security contexts due to decoupled bind/attach

• Current IOMMU API has problem on the transition between unmanaged domains
  • Default domain is automatically re-attached after detaching from previous context

Need cooperation from IOMMU core!
Manage User-initiated DMAs (Cont.)

iommufd (no group knowledge)

Open devicefd (block access)

Binding iommufd (unbind access)

Attach ioas1

DMA blocked

Detach ioas1

default domain

Attach ioas2

DMA blocked

Detach ioas2

Device-centric IOMMU API

IOMMU Core (user-dma awareness)

Default domain

Default domain

default domain

user ioas1

DMA blocked

DMA blocked

DMA blocked

DMA blocked

user ioas2

user ioas2

Device centric IOMMU API

Default domain

Default domain

Default domain

Default domain

Default domain

Open devicefd (block access)

Bind iommufd (unbind access)

Device centric IOMMU API

IOMMU Core (user-dma awareness)

iommufd_device_init_user_dma():
- If the first device in the group
  * Validate and start monitoring group
  DMA ownership
  * Mark the group for user-dma
  * Block DMA for the entire group
  - Else
    * refcount_inc(user_dma_cnt)

iommufd_attach_device():
- If the first device in the group
  * Attach the group to ioas
  - Else
    * Attach the group to default domain
    * Refcount_inc(attach_cnt)

iommufd_detach_device():
- If the last device in the group
  * Detach the group from ioas
  * Block DMA instead of re-attaching to default domain
  - Else
    * Detach the group to default domain
    * Stop monitoring group DMA ownership
    * Refcount_dec(user_dma_cnt)

iommufd_device_exit_user_dma():
- If the last device in the group
  * Clear user-dma flag
  * Re-attach the group to default domain
  * Stop monitoring group DMA ownership
  - Else
    * Refcount_dec(user_dma_cnt)
If time allows

- ioasid naming conflict
  - fd-local software handle
  - /drivers/iommu/ioasid.c, representing hw asid (pasid/ssid)

- Module name and devnode
  - iommufd and /dev/iommu
  - uiommufd and /dev/uiommu

- Consolidate vfio type1v2

- /dev/vfio/devices/ hierarchy
  - A plain layout mixing all types together
    - /dev/vfio/devices/0000:00:14.2 (pdev-pci)
    - /dev/vfio/devices/PNP0103:00 (pdev-platform)
    - /dev/vfio/devices/83b8f4f2-509f-382f-3c1e-e6bfe0fa1001 (mdev)
  - Subdirectories based on device types
    - pci, platform, ccw, etc.
    - Pdev vs. mdev

- Do we need to build iommufd as a separated module?
Backup
Current Situation

VFIO

IOMMU Core

vDPA

Not a scalable architecture moving forward!
Manage User-initiated DMAs

Current VFIO

Groupfd: set container
  - Open devicefd
  - Close devicefd

Groupfd: unset container

default domain 🛡️ user ioas 🛡️ default domain

User-initiated DMAs

- A single security context
- Monitor DMA ownership