IOMMUFD Discussion

Eric Auger
Yi Liu
Baolu Lu
Jason Gunthorpe
Kevin Tian
Session Goals

Overview of iommufd and its goals
Review progress so far
Open discussions from mailing list
Discussion merging threshold
Overview

- Expose IOMMU HW to userspace control
- Provide advanced IOMMU features
- Provide high performance “kernel bypass” to support VMs
- Support all kernel subsystems equally: VDPA, VFIO, uacce, etc
iommufd-native

- iommufd replaces VFIO group and container FDs
- VFIO becomes a simple device driver
  - Device is opened via cdev
  - No use of group or container
- New VFIO APIs to manage iommufd connection
- New IOMMU features flow through iommufd
Basic uAPI

IOMMUFD_CMD_DESTROY
Destroy an object

IOMMUFD_CMD_IOAS_ALLOC
Allocate an IO Address Space (IOAS)

IOMMUFD_CMD_IOAS_ALLOW_IOVAS
Ensure a range of IOVA is available

IOMMUFD_CMD_IOAS_COPY
Copy the mapping from a range of IOVA to another

IOMMUFD_CMD_IOAS_UNMAP
Remove IOVA

IOMMUFD_CMD_IOAS_MAP
Install user VA to IOVA

IOMMUFD_CMD_IOAS_IOVA_RANGES
Inquire valid ranges of IOVA

IOMMUFD_CMD_VFIO_IOAS
Work with the VFIO compat IOAS
# Iommufd Features

<table>
<thead>
<tr>
<th>Features</th>
<th>Ver</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>iommufd generic interface</td>
<td>RFC v2</td>
<td>Clean and in a reviewable state</td>
</tr>
<tr>
<td>VFIO integration with iommufd (compat. mode)</td>
<td>github</td>
<td>In preparation for formal patch series</td>
</tr>
<tr>
<td>IOMMU dirty tracking</td>
<td>github</td>
<td>Prototype ready but not fully scrutinized</td>
</tr>
<tr>
<td>iommufd nested translation</td>
<td>github</td>
<td>Prototype in progress</td>
</tr>
<tr>
<td>VFIO integration with iommufd (via cdev)</td>
<td>github, Qemu</td>
<td></td>
</tr>
<tr>
<td>I/O page fault</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>PASID support</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Share KVM page table with IOMMU</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>
## Preparatory Series

<table>
<thead>
<tr>
<th>Series</th>
<th>Ver</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplify domain/device compat. checking</td>
<td>v6</td>
<td>Respin with different error codes</td>
</tr>
<tr>
<td>Redo SVA with domain-centric model</td>
<td>v13</td>
<td>Close to be merged</td>
</tr>
<tr>
<td>Introduce a device struct for vfio_device</td>
<td>v2</td>
<td>Collecting reviewed-by</td>
</tr>
<tr>
<td>VFIO API for dirty tracking</td>
<td>v6</td>
<td>Merged</td>
</tr>
<tr>
<td>Complete restructuring the vfio mdev model</td>
<td>v7</td>
<td>Still see breakage on ccw</td>
</tr>
<tr>
<td>VFIO dma_buf exporter for VFIO</td>
<td>v2</td>
<td>Requires major infrastructure work</td>
</tr>
<tr>
<td>Isolate vfio container in preparation for iommufd</td>
<td>v1</td>
<td>Simple change with only a naming open.</td>
</tr>
<tr>
<td>Provide iommu_domain ops for power</td>
<td>v1</td>
<td>Not elegant but no owner to further improve if it’s not picked by maintainer</td>
</tr>
</tbody>
</table>
Differences

- TYPE1v2 only
- No dma_entry_limit (rely on memory cgroup)
- Pinned page accounting (user vs. mm)
- VFIO P2P is a followup
- TBD VFIO_DMA_MAP_FLAG_VADDR
- errno differences
- SPA PR is left out (until it is moved to use iommu_domain)
Threshold for merging

As-is VFIO integration and functional complete
dpdk, qemu functionally work, with known limitations
6.2 is a plausible target
Want to get parallel work into the community