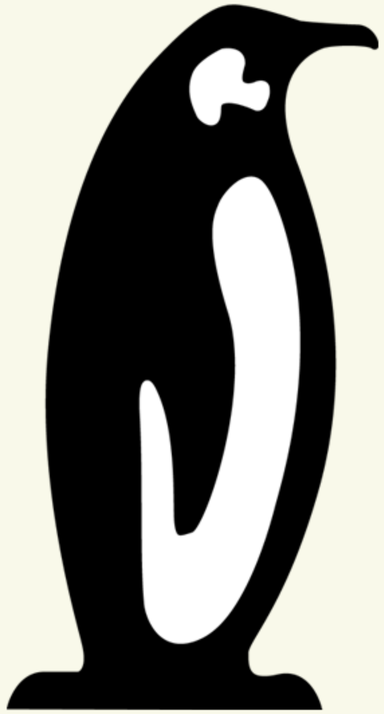


Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

IOMMUFD Discussion

Jason Gunthorpe

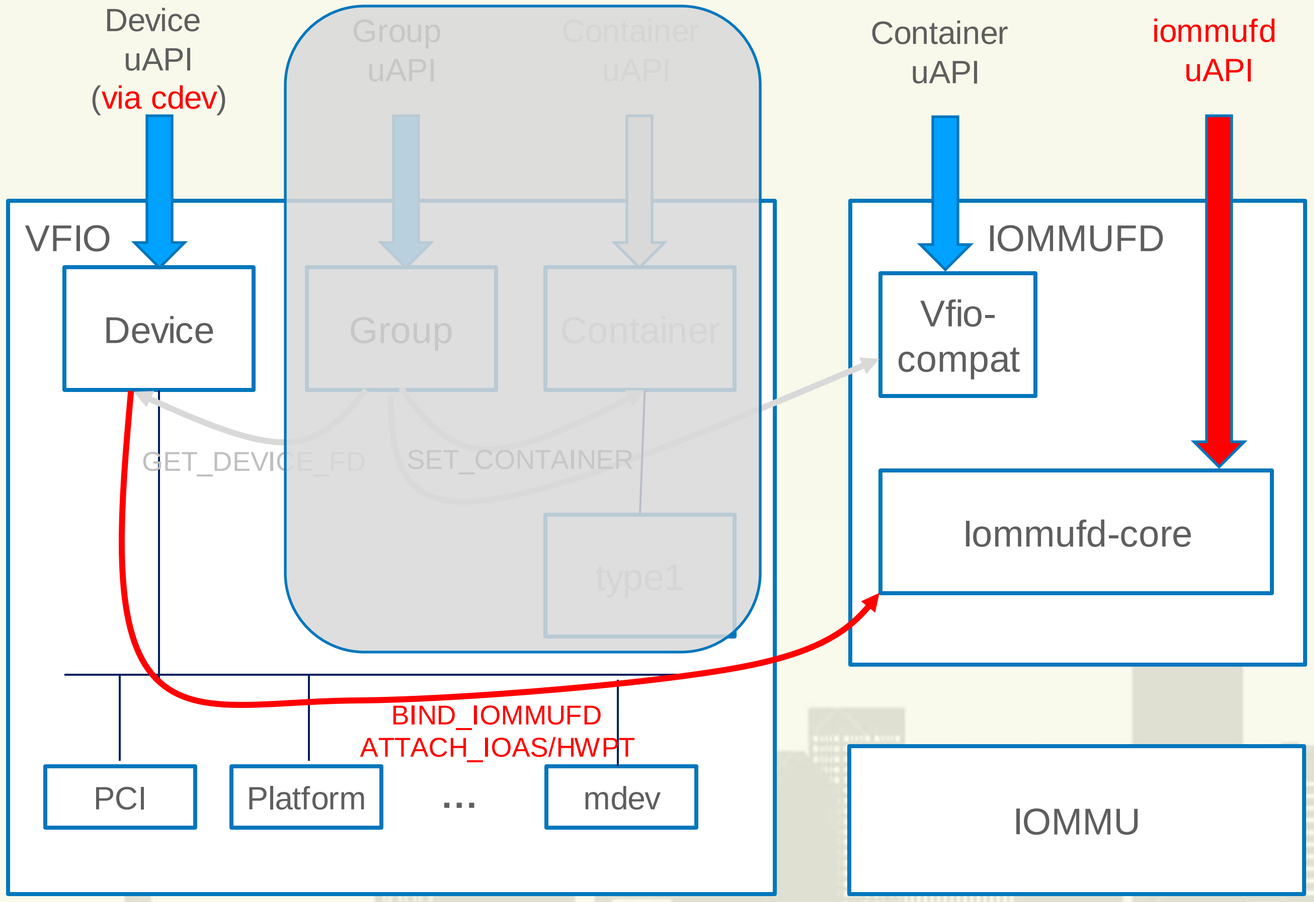
Session Goals

Review of iommufd
Patches merged
Next patches
Trouble spots

Overview

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

Architecture



Support IOMMU HW for vIOMMU acceleration

vIOMMU Concepts

vIOMMU Concept	IOMMUFD
Guest vPCI device	DEVICE
Guest Physical Map (identity)	IOAS & HWPT
Guest IO Page Table (shadowed)	IOAS & HWPT
Guest Physical Map (nesting base/S2)	IOMMU_HWPT_ALLOC_NEST_PARENT
Guest IO Page Table (nested/S1)	HWPT w/ pt_id=HWPT
Guest PASID & IO Page Table (nested)	HWPT w/ pt_id=HWPT
DMA Dirty Logging	IOMMU_HWPT_ALLOC_DIRTY_TRACKING
IOTLB/ATC Invalidation (trapped)	Invalidation cmd to HWPT
HW Commands (direct)	Create CMDqueue & mmap (HWCMD)
IO Event	IO Event FD
KVM Guest Physical PTEs	HWPT

IOMMUFD features

Feature	Version	
iommufd generic interface	v6.2	Merged
VFIO integration with iommufd	v6.2	
VFIO cdev support	v6.6	
IO Page Table dirty tracking	v6.7	
User IO Page Table	v6.7	
User IO Page Table Invalidation	v5	In Progress
Fault delivery to user space	v2	
PASID Support	RFC	
SIOV Support	RFC	TBD
VDPA Integration	RFC	
Share KVM page table with IOMMU	N/A	
Confidential Compute TDISP	N/A	
Non-mediated MSI Configuration	N/A	

VFIO Container Gaps

- PCI Peer to Peer
- VFIO_UPDATE_VADDR
- ERRNO audit
- POWER/SPAPR

Trouble Spots

- PASID PCIe Capability in VF
- Invalidation IOCTL Design
- Driver Modernization
- ARM GICv3/4 ITS Page
- Dirty Tracking Optimization

PASID VF Capability

PCI Spec defines the PASID Capability block as being PF ONLY

No way for the guest to discover PASID capability of a VF

No easy way to insert a Capability block into an arbitrary VF config space

Qemu has a quirk list and inserts a Capability Block when safe

Invalidation

- Exposure of IOMMU Cachable data to Guest Memory
- vIOMMU emulates the real HW and talks in terms of HW cachable objects
- Invalidation commands must be 1:1 with the actual HW

- Eg: On ARM an invalidation of an ASID does not convey enough information to generate ATC invalidations. VM must generate ATC invalidations. ATS must follow Guest Setting.

Invalidation IOCTL forwards the native command list

Driver Modernization

1. Global Static ‘never fail’ BLOCKED and IDENTITY domains
2. Map before Attach for PAGING domains
3. attach_dev() failure does not change HW
4. Hitless IDENTITY->DMA/PAGING->IDENTITY for active IOVA
5. PAGING->BLOCKED->PAGING is !IDENTITY, even under failure
6. PAGING domains attached to a PASID
7. Set IDENTITY/BLOCKED/PAGING on RID while PASID in use
8. SVA domains attached to a PASID
9. SVA domains use new core infrastructure
10. Hitless change between domains when possible

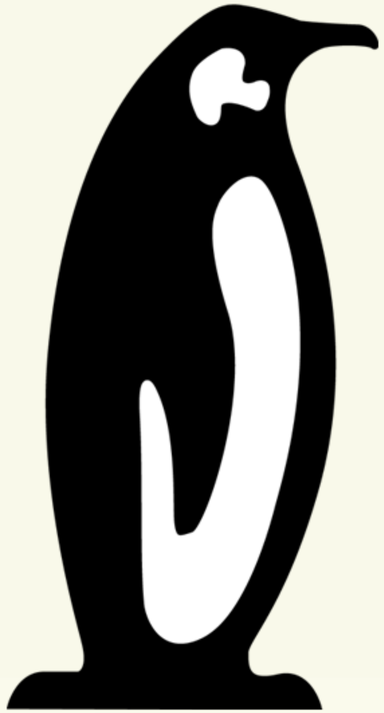
ARM ITS Page

- Real ITS page is hardwired in kernel to map at MSI_IOVA_BASE=0x8000000
- VM's GIC driver gets a fake ITS page at the usual physical address. qemu sets up an ACPI table for RESV_DIRECT
- VM sees two reserved regions @MSI_IOVA_BASE: RESV_DIRECT (from ACPI) and RESV_SW_MSI (from smmu-v3)
- GIC driver avoids calling iommu_dma_prepare_msi()

Nested VFIO doesn't work, MSI trapping is mandatory, nested IMS doesn't work, TDISP doesn't work

Dirty Tracking

- Improve io page table performance
- Dynamically change IOPTE page size
- Threaded qemu dirty data fetch
- PRI based dirty logging



Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023