Contribution ID: **99** Type: **not specified**

Unified I/O page table management for passthrough devices, in-kernel API discussion between IOMMU core and /dev/iommu

Thursday, September 23, 2021 9:45 AM (45 minutes)

When a device is passed through to user space, DMAs from this device are untrusted by the kernel. I/O page tables must be enabled in the IOMMU so each assigned device can only access the I/O virtual address space that is created by respective device passthrough frameworks (VFIO, vDPA, etc.).

Until now I/O page tables are considered as a device attribute, thus managed through VFIO/vDPA specific uAPIs. However this model doesn't scale toward advanced I/O virtualization usages, e.g. subdevice passthrough which requires more than one I/O page table per device, SVA virtualization which needs to support user-provisioned I/O page table (nested on a kernel page table), and I/O page faults which are necessary for improved memory utilization, etc. Better avoid reinventing the new wheel in every framework.

Having an unified uAPI is the answer here. The proposal is generalizing things about I/O page table management via a new interface (/dev/iommu), while allowing passthrough frameworks to connect their devices with selected I/O page tables via a simple protocol. This approach allows VFIO/vDPA to focus on aspects about device management, leaving DMA isolation enforced through the generic interface. This talk is aimed to get consensus on the overall design choices and execution plan cross multiple subsystems.

As we have reached a consensus on the /dev/iommu proposal (https://lore.kernel.org/linux-iommu/MWHPR11MB1886422D4839B372C6A
it's time to have some discussions on the in-kernel APIs between the IOMMU core and the /dev/iommu implementation. This discussion can provide some guidance for the developers who are going to implement /dev/iommu.

I agree to abide by the anti-harassment policy

I agree

Primary authors: TIAN, Kevin (Intel); LU, Baolu

Presenters: TIAN, Kevin (Intel); LU, Baolu

Session Classification: VFIO/IOMMU/PCI MC

Track Classification: VFIO/IOMMU/PCI MC