Exposing a virtual IOMMU interface to KVM guests

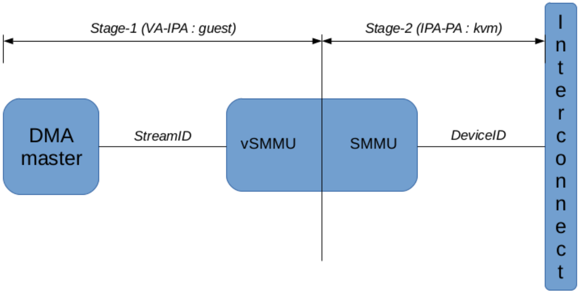
Linux Plumbers IOMMU Microconference

Will Deacon

<will.deacon@arm.com>

October, 2014

The Problem



- A KVM host can use an IOMMU for device passthrough to a guest
- But we also want to provide IOMMU services to the guest for DMA and userspace I/O
- Without the need for para-virtualisation

Modern IOMMU designs support this mode of operation in the hardware.

ARM SMMU Architecture

The ARM SMMUv2 architecture supports two stages of translation (similar to the CPU), which can be *nested*:

Stage-1 Translates a virtual address (VA or guest address) to an intermediate physical address (IPA or host VA in QEMU). The page table walker expects IPAs.

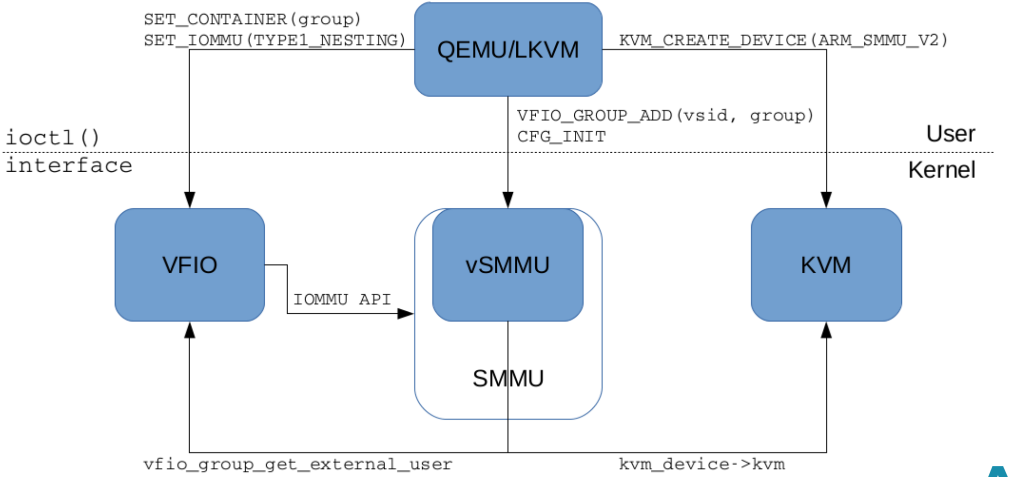
Stage-2 Translates an IPA into a physical address (host PA). This corresponds directly to the mapping created by `KVM_SET_USER_MEMORY_REGION`.

There are also some differences in permission model, address range, table format etc.

We want the guest to program stage-1 directly and KVM/VFIO to program stage-2.

Plumbing

KVM has its own device assignment ioctls, but they're deprecated in favour of VFIO.



Improvements and Questions

- Do other IOMMUs support this feature?
- Can we make the kvm-vfio code reusable?
- What about non-PCI devices?
- Dealing with I/O page faults (and also at stage-2?!)
- Error handling
- Sharing CPU page tables at both stages
- RID/SID/DID mapping
- Dealing with complex I/O topologies (multiple SMMUs, PCI bridges, etc)
- Actually getting a guest to use the IOMMU...
- Differences between VFIO IOMMU type, VFIO IOMMU extension, IOMMU domain attribute, IOMMU capability

Thank You

`git://git.kernel.org/pub/scm/linux/kernel/git/will/linux.git iommu/pci`