redhat.

Status of Dual Stage SMMUv3 integration

Eric Auger

Plumber Sept 2019

VFIO/IOMMU/PCI MC

Some Background

- Goal: Virtual SMMUv3 able to work with VFIO assigned devices

- SMMUv3: no caching mode
  - arm-smmu-v3 tlbi-on-map option RFC (July/Aug 2017)

- SMMUv3 HW nested paging:
  - 1 stage “owned” by the guest, 1 stage owned by the hyp
  - First RFC sent on Aug 2018
  - Shared dependencies with Intel's SVA series:
    - Fault Reporting infra (now upstream)
    - cache_invalidate

Last State

- Aug 19:
  - [PATCH v9 00/14] SMMUv3 Nested Stage Setup (IOMMU part)
  - [PATCH v9 00/11] SMMUv3 Nested Stage Setup (VFIO part)

- Tested on Cavium ThunderXv2 HW, also tested by Linaro (Zhangfei Gao)

- QEMU integration code:
  - [RFC v5 00/29] vSMMUv3/pSMMUv3 2 stage VFIO integration

[PATCH v9 00/14] SMMUv3 Nested Stage Setup (IOMMU part)

- attach/detach_pasid_table != Intel's sva_bind/unbind_gpasid

- Most complex part related to MSI SMMU binding
  - bind/unbind_guest_msi
  - IOMMU_DMA_NESTED_MSI_COOKIE

- SMMUv3 internal changes
  - New State machine Transitions (S2 <-> S1 + S2)
  - Implement above iommu uapis

[PATCH v9 00/11] SMMUv3 Nested Stage Setup (VFIO part)

- VFIO API directly mapping onto IOMMU uapi

- Bulk of the series now related to the fault report feature
  - Relies on specific region and IRQ
  - mmappable ring and trapped header

- Removed complex version handling
  - Upgrade to recov fault region handled through struct vfio_region_info_cap_fault versioning

Open Discussion

- Any Conceptual Blocker?
- Questions/Open points about the iommu uapi?
  - where is the last version of the uapi?
- Fault Handler Unregistration still can fail
- MSI binding headache