Contribution ID: **242** Type: **not specified**

Exposing PCIe topology to Guest OS for peer-to-peer

Monday, 12 September 2022 12:30 (30 minutes)

Doing peer-to-peer (aka p2p) is becoming more common these days. Whether it is done between GPUs and RDMA NICs, or between AI accelerators and NVME devices, doing p2p can decrease the CPU load, increase the b/w and improve the latency of data movement.

When implementing p2p code, whether it is using the p2pdma infrastructure or dma-buf framework, the kernel code eventually needs to calculate the PCI distance between the peers in order to validate whether they can perform p2p.

This is done today by calling pci_p2pdma_distance_many() which validates that either all the peer devices are behind the same PCI root port or the host bridges connected to each of the devices are listed in the 'pci_p2pdma_whitelist'.

The problem is that this function is not able to calculate the distance when working inside a Virtual Machine because the PCIe topology is not exposed to the Guest OS. Also the host bridges are not exposed, so even if they are whitelisted, the Guest OS wouldn't know that.

I would like to brainstorm on how to solve this problem because today the only way to do p2p inside a VM is to run a modified kernel which bypass this function.

I agree to abide by the anti-harassment policy

Yes

Primary author: GABBAY, Oded (Intel)

Presenter: GABBAY, Oded (Intel)

Session Classification: VFIO/IOMMU/PCI MC

Track Classification: LPC Microconference: VFIO/IOMMU/PCI MC