Contribution ID: **426** Type: **not specified**

”VF keep alive”

Thursday, 19 September 2024 10:00 (45 minutes)

At Microsoft, we are working on a project called openHCL, which is a Linux-based paravisor featuring a user-mode virtualization stack.
For more details, you can check out this micro conference: LPC Event.

The paravisor is upgraded using a servicing operation where the old paravisor is shutdown and the new paravisor is booted into. Our goal is to minimize the
servicing time as much as possible. As part of this project, we manage several PCIe devices (e.g., NVMe) using VFIO (via vfio_pci_core.c). We have identified
that tearing down and reinitializing the devices takes a significant portion of this servicing time. To avoid this extra latency, we are considering making the DMA
buffers persistent across reboots and avoiding any actions (hardware access) that could alter these buffers. Since we are using noiommu option, the saving and
restoring IO pages is not a concern, also hypervisor allows to keep the physical pages intact which allowed us to keep the DMA buffers persistent across boots.
This solves the first part of the problem.

The another part of the solution is to keep the NVMe device alive across reboots with its hardware configurations intact. We have observed that accessing PCI
device registers in vfio_pci_core.c can trigger DMA actions, which may alter the DMA buffers. For example, the pci_clear_master function clears the
“Bus Master” bit, which resets the controller and invalidates all DMA buffers.

To prevent hardware access through VFIO following a reboot, we are considering implementing a flag to avoid all these hardware access. This flag can be passed
through a new vfio ioctl or sysfs, but we are also open to alternative methods that could be more appropriate for integrating this solution into the upstream.

Primary author: SINGH, SAURABH (Microsoft)

Session Classification: Birds of a Feather (BoF)

Track Classification: Birds of a Feather (BoF)