
Contribution ID: **269** Type: **not specified**

# PCIe Portdrv - finding a path forwards?

*Wednesday 18 September 2024 10:00 (20 minutes)*

Key takeaway - interrupts are what makes this complex.

The PCIe port driver is an unusual beast:
- It binds to several Class Codes because they happen to have common features. (PCI Bridges of various types, Root Complex Event Collectors).
- It then gets ready to register a set of service drivers.
- Before registering those service drivers it has to figure out what interrupts are in use which requires per service driver code (so as not to use more interrupt vectors than necessary). An enable lots, check usage and shrink dance occurs.
- The available services are all baked in - the modularity is largely an illusion.

New features are being implemented in PCIe switches and Root Ports. These are enumerable via config space + BARs (VSEC / DVSEC / PCI 6.2 MCAP) Today three approaches exist to add support:
- If they need interrupts, they have to be a portdrv service (e.g. CXL Performance Monitoring Units)
- If they don't use interrupts, then a parallel search and registration infrastructure can be used (CXL ports / HDM decoders, Designware RP PMUs) - however this creates non obvious life time issues for switch ports which may be hot removed.
- Support only in the PCIe core - no interrupt possible (CMA for device attestation, interrupts would be nice!).

A number of discussions have taken place on the mailing list (most recently https://lore.kernel.org/linux-pci/20240605180409.GA520888@bhelgaas/ ) and in previous LPC corridor tracks.

The potential paths forward are:
1) Admit we are stuck with basic concept of portdrv. Work out how to make it extensible.
2) Push all the current service drivers (AER, DPC etc) into the PCI core and deal with interrupts (either dynamic MSI-X or quiescing to resize or just allocate N and assume enough!). Then support additional features via standard PCI drivers on top. (This runs into some snags due to devres)

The aim of this session is to first seek agreement on the requirements and then how they align with the possible options.

- Is MSI-X only for 'new' portdrv support features an option?
- Maintain existing /sys/bus/pci_express/devices/* that has no practical use?
- Bus master ok before driver load? (Block list needed?)
- Can we actually make the interrupt allocation dance work? (probably not!)

Early prototypes will hopefully identify additional open questions before LPC.

**Primary author:** CAMERON, Jonathan (Huawei Technologies R&D (UK))

**Presenter:** CAMERON, Jonathan (Huawei Technologies R&D (UK))

**Session Classification:** VFIO/IOMMU/PCI MC

**Track Classification:** VFIO/IOMMU/PCI MC