Adding RAS Support for CXL Port Devices

2023 LINUX PLUMBERS CONFERENCE
CXL MICROCONFERENCE

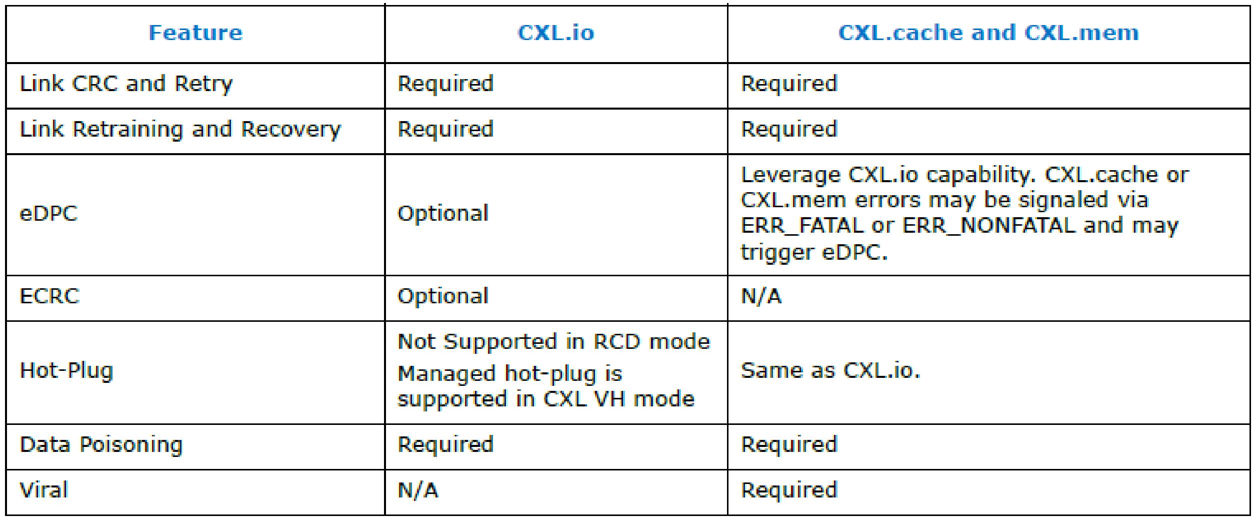
Meet the Presenters

- Robert Richter
  - Kernel and CXL software engineer @AMD
- Terry Bowman
  - Kernel and CXL software engineer @AMD

CXL: Reliability, Availability, and Serviceability (RAS)

- CXL can simultaneously carry three protocols: CXL.io, CXL.cache, and CXL.mem.
- The table below lists the RAS features supported by CXL and their applicability to CXL.io vs. CXL.cache and CXL.mem.

Table 12-1. CXL RAS Features



Feature	CXL.io	CXL.cache and CXL.mem
Link CRC and Retry	Required	Required
Link Retraining and Recovery	Required	Required
eDPC	Optional	Leverage CXL.io capability. CXL.cache or CXL.mem errors may be signaled via ERR_FATAL or ERR_NONFATAL and may trigger eDPC.
ECRC	Optional	N/A
Hot-Plug	Not Supported in RCD mode Managed hot-plug is supported in CXL VH mode	Same as CXL.io.
Data Poisoning	Required	Required
Viral	N/A	Required

[1] CXL 3.0 - Supported RAS Features, Table 12-1

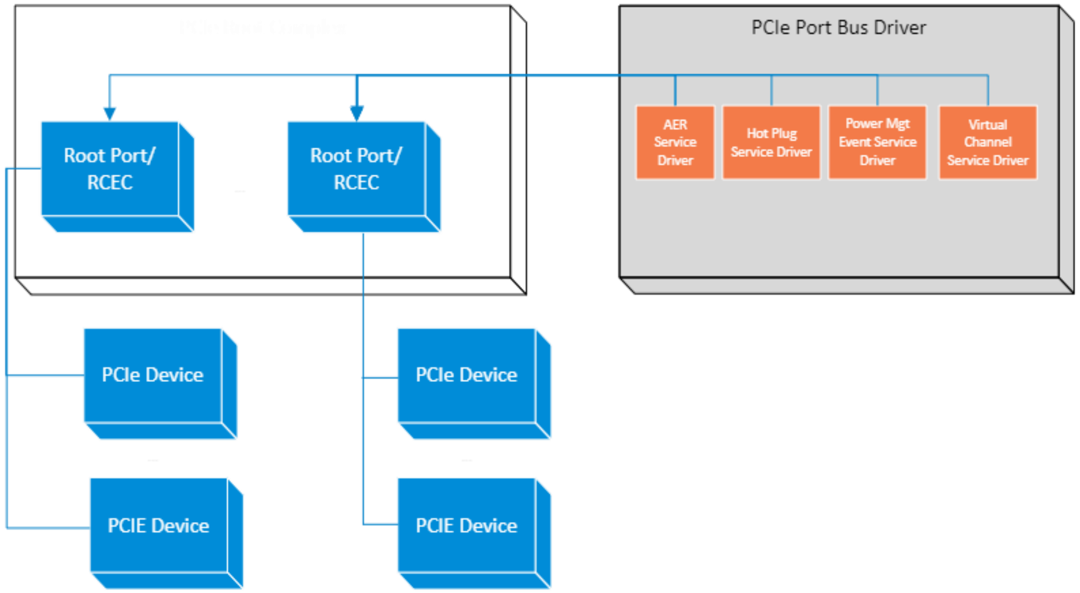
CXL: Reliability, Availability, and Serviceability (RAS)

- Protocol errors detected by CXL components are escalated and reported using standard PCIe error reporting mechanisms over CXL.io as AER uncorrectable internal errors (UIEs) and/or correctable internal errors (CIEs).[1]
- There are no error pins that connect CXL devices to the Host. Errors are communicated between the Host and the CXL components via messages over CXL.io.[1]
- Note that RAS protocol errors are handled separately and in different flow than device errors reported through mailbox event records.

[1] CXL 3.0 - 12.2 - CXL Error Handling, pg. 701.

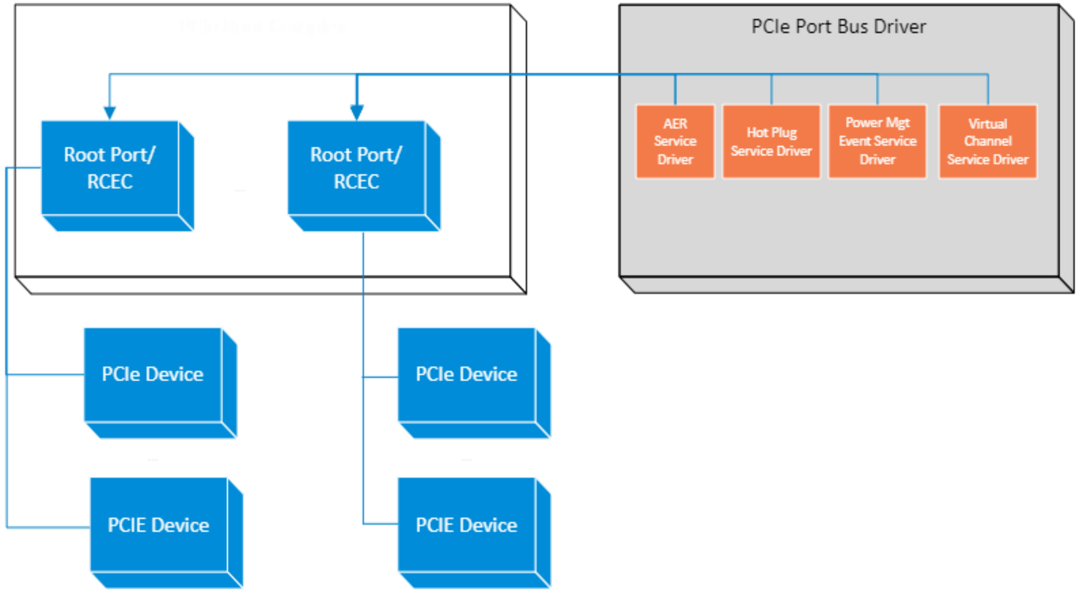
PCIe Port Bus Driver

- Linux (driver model) currently binds the PCIe port bus driver to PCIe port bridge class devices of type: PCIE_EXP_TYPE_ROOT_PORT, PCIE_EXP_TYPE_RC_EC, PCIE_EXP_TYPE_UPSTREAM, PCIE_EXP_TYPE_DOWNSTREAM
- PCIe port bus driver's probe routine queries the device's capabilities for supported services to manage.
- PCIe port bus driver uses service drivers to manage device capabilities AER, PME, HP, and VC.
- Note that the AER service driver usage is gated by:
  1.) Native AER platform,
  2.) Kernel compilation with CONFIG_PCIEAER.



AER Error Handling

- Once a device detects an unmasked/enabled protocol error, the device sends an AER notification message to its root port.
- The root port saves the device's BDF into the AER root source ID register.
- The root port notifies the OS through AER interrupt.
- The AER port service driver handles the interrupt.
- The AER port service driver reads the AER root source ID register to determine the BDF of the errored device.
- The AER port service driver logs the device's AER error information and handles fatal/uncorrectable errors.



Upstream Status for CXL RCH RAS (CXL 1.1)

- CXL and AER port driver require changes to handle restricted CXL host (RCH) downstream port protocol errors.
- This is addressed in currently pending upstream patch expected to be accepted into kernel v6.7.[1]
- Is required because the RCH downstream port is implemented as a root complex register block (RCRB).
- Due to the RCRB implementation, the RCH downstream port does not have a BDF.

[1] - https://lore.kernel.org/all/20231018171713.1883517-1-rrichter@amd.com/

Upstream Status for CXL RCH RAS (CXL 1.1) - Continued

- RCH AER flow:
  - The RCH downstream port sends an AER UIE/CIE message to the RCEC in the case of RAS/protocol errors.[1]
  - The RCEC's AER root source ID reflects the RCEC BDF.
  - The RCEC notifies the OS through AER interrupt.
  - The AER port service driver recognizes source ID BDF from an RCEC device with AER internal errors.
  - The AER port service driver then calls the RCEC's child RCiEPs (endpoints) error handlers for handling the downstream port's errors.

[1] - CXL 3.0 - 12.2.1 - Protocol and Link Error Reporting, pg. 701.

Supporting CXL VH Port Device RAS (CXL 2.0)

CXL 1.1 changes mentioned earlier are RCH downstream port specific and do not support CXL 2.0 port devices. As a result, CXL 2.0 root ports and switches do not currently include RAS support. Below are the constraints and requirements involved with adding CXL 2.0 port device RAS support.

- Constraints
  - PCIe port devices are bound to the PCIe port bus driver. This should be maintained to continue using other services drivers (HP, PME, VC).
  - UIE/CIE are device driver specific.[1]
- Requirements
  - Scalability
    - AER is PCIe error handling standard.
    - Support PCIe devices/drivers using AER UIE/CIE.
  - Handling
    - Driver specific handlers. i.e., driver callbacks.
    - Flexibility for device specific handling.
  - Enablement
    - Capable of enabling/disabling on per device granularity.

[1] PCI6.0 - 6.2.10 Internal Errors, p.693.

Solution 1: CXL RAS PCIe Port SERVICE

- Add PCIe port bus CXL service driver
  - Place next to the AER, hotplug, and power management event service drivers.
  - Use CXL DVSEC register locator as identifier in service driver probe routine.
  - Requires CXL RAS register discovery and mapping
    - Involves CXL capability parsing in MMIO space (not PCI config space).
    - Functionality already exists in drivers/cxl/core/regs.c.
    - Choose between duplicating code with existing CXL driver or export from CXL driver?
- Observations from development
  - Required copying component block and RAS register discovery and mapping from CXL driver.
  - Duplicated from CXL driver.

Solution 2: Update AER Service Driver for Port Device UIE/CIE

- Extend current AER port service driver to support UIE/CIE callbacks.
- AER port service driver currently assigns each port device an instance of ‘struct aer_rpc’.
- Extend 'struct aer_rpc' to include UIE/CIE callback.
- Extend AER port service driver to include exported routine initializing the 'struct aer_rpc’ UIE/CIE callbacks.
- This design adds generic CIE/UIE callback functionality for PCIe port devices.
    - Does not add support for non-PCIe port devices.
- Update the CXL driver to register AER port service driver UIE/CIE callbacks during switch discovery or initialization.
- In the case of AER UIE/CIE, the AER port driver will call the callbacks if they are registered.

Solution 3: Update AER Service Driver for PCIe UIE/CIE

- Variation of Solution 2.
- **Allows any PCIe device to handle AER UIE/CIE errors.**
- UIE/CIE callback function pointers are added to ‘struct pci_driver::err_handler’.
- Add exported callback registration function to AER service driver.
- For CXL port device case:
  - CXL port devices are bound to PCIe bus port driver as done today.
  - UIE/CIE handlers in ‘struct pci_driver’ are registered to CXL driver functions.
- In the case of AER UIE/CIE, the AER port driver will call the callbacks if they are registered.
- Difference between solution 3 and solution 2:
  - Solution3 includes callback handler in ‘struct pci_driver::err_handler’ and supports all PCIe devices.
  - Solution2 includes callback handler in 'struct rpc_aer' that is limited to only PCIe port devices.

Discussion, Feedback, and Q&A