CXL: Welcome to the Real World

Oct 31, 2023

PJ Waskiewicz, Kernel Engineer

NetDev 0x17

Vancouver, BC, Canada


Welcome,
to the real
world

Protocol Realities

- Not an extension of PCIe
- CXL is a collection of protocols
  - CXL.io: equivalent to PCIe
  - CXL.mem: Memory-coherent with the host
  - CXL.cache: Cache-coherent partner with host CPUs (!!!)
- Oversight by CXL Consortium
- Last interconnect standing!
  - Killed CCIX and ate Gen-Z

Timeline Realities

- CXL 1.1: Released June 2019
- CXL 2.0: Released November 2020
- CXL 3.0: Released August 2022
- CXL 3.1: **Releasing** November 2023

- Intel Sapphire Rapids: January 2023
- AMD Genoa: November 2022
- Intel Emerald Rapids: Likely December 2023

- AMD Turin: Likely EOY 2024
- Intel Granite Rapids: Likely 2024

Upstream Realities

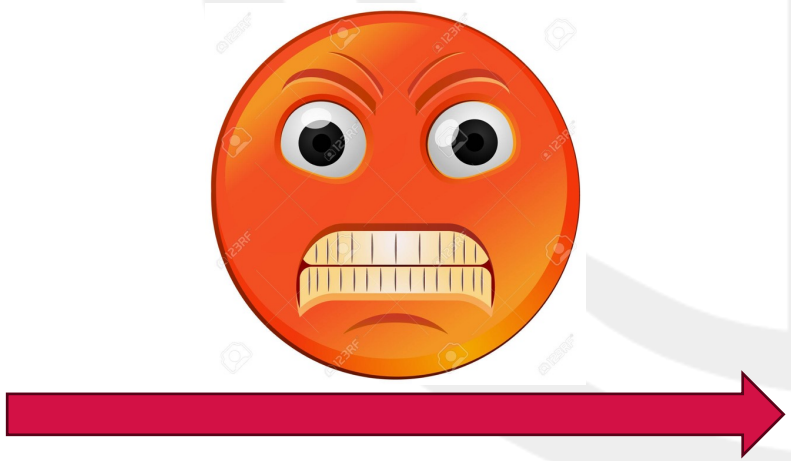
- Upstream support pretty robust
- https://elixir.bootlin.com/linux/latest/source/drivers/cxl
- Mostly support for Type-3 memory expanders
- Presents standardized memory device files
- Userspace tools with DAX support integrated:
  https://docs.pmem.io/ndctl-user-guide/cxl-man-pages/cxl-1

Upstream (unfortunate) Realities

- Upstream chasing latest spec versions
- Mismatches between what is real, and what is slideware
- https://git.kernel.org/pub/scm/linux/kernel/git/ppwaskie/net.git/commit/?id=0a19bfc8de93d5b5d12cf0a7bb74efc88b9ad077

Upstream (unfortunate) Realities

```
$ lscpu
Architecture:            x86_64
CPU op-mode(s):          32-bit, 64-bit
Byte Order:              Little Endian
Vendor ID:               GenuineIntel
CPU family:              6
Model:                   143
Model name:              Intel(R) Xeon(R) Platinum 8462Y+
```



EMR and Genoa support ACPI0017

```
$ ls -l /sys/bus/acpi/devices | grep ACPI
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:10 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:10
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:11 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:11
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:12 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:12
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:13 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:13
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:14 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:14
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:15 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:15
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:16 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:16
lrwxrwxrwx 1 root root 0 Oct 28 07:13 ACPI0016:17 ->
../../../devices/LNXSYSTM:00/LNXSYBUS:00/ACPI0016:17
```

Successful Realities

```
[    0.000000] ACPI: SRAT: Node 0 PXM 0 [mem 0x00000000-0x7fffffff]
[    0.000000] ACPI: SRAT: Node 0 PXM 0 [mem 0x100000000-0x407fffffff]
[    0.000000] ACPI: SRAT: Node 1 PXM 1 [mem 0x4080000000-0x807fffffff]
[    0.000000] ACPI: SRAT: Node 2 PXM 2 [mem 0x8080000000-0x80bfffffff]
```

```
Capabilities: [500 v1] Designated Vendor-Specific: Vendor=1e98 ID=0000 Rev=2 Len=60: CXL
    CXLCap:  Cache+ IO+ Mem+ Mem HW Init+ HDMCount 1 Viral-
    CXLCtl:  Cache+ IO+ Mem+ Cache SF Cov 0 Cache SF Gran 0 Cache Clean- Viral-
    CXLSta:  Viral-
```

Continued Realities

- Possible to map CXL.mem regions directly in drivers
  - Parse DVSEC (Designated Vendor-Specific Extended Capabilities) registers directly
  - Parse HDM (Host-managed Device Memory) decoders
  - `ioremap()` with desired memory access profile
- Knife fight with `CONFIG_DAX_HMEM`

Future Realities

- Standardization at CXL 2.0 should help the field
- Hotplug support is badly needed…
- Standard tools to manage CXL.mem devices will help vendors with software support
- RAS features coming in CXL 3.0…

Questions?