Introduction to CXL Multi-Headed Devices

EMPOWERING OPEN.

OCP GLOBAL SUMMIT

OCTOBER 18-20, 2022
SAN JOSE, CA

Introduction to CXL Multi-Headed Devices

SERVER

Vincent Haché, Director of Systems Architecture, Rambus

Jérôme Glisse, Engineer, Google

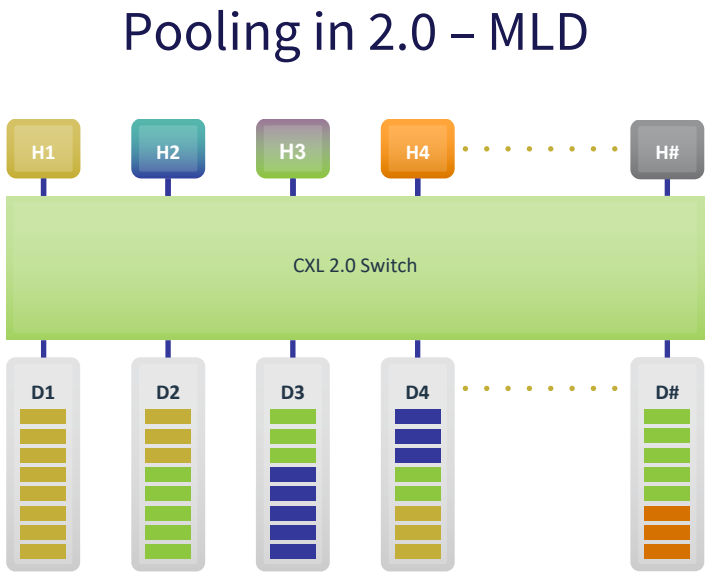
OPEN PLATINUM™

Introduction to CXL Multi-Headed Devices

- Introducing Multi-Headed Devices (MHDs)
- Multi-Headed MLD versus Multi-Headed SLD
- MHD Management
- Dynamic Capacity Devices (DCDs)
- How Software can use memory from Multi-Headed device
- Host Software requirement
- How to manage memory at infrastructure level

Pooling in CXL 2.0

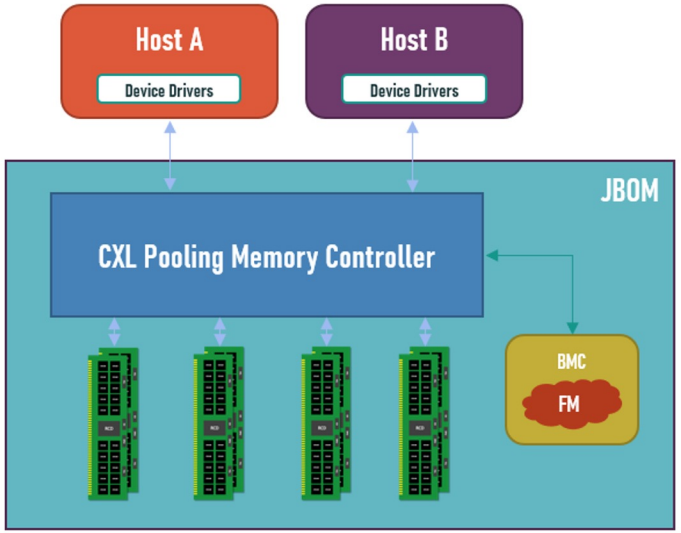
- Pooling in 2.0 – Multi-Logical Device (MLD)
- Up to 16 hosts
- Requires MLD-capable switch



Introducing Multi Headed Devices (MHDs)

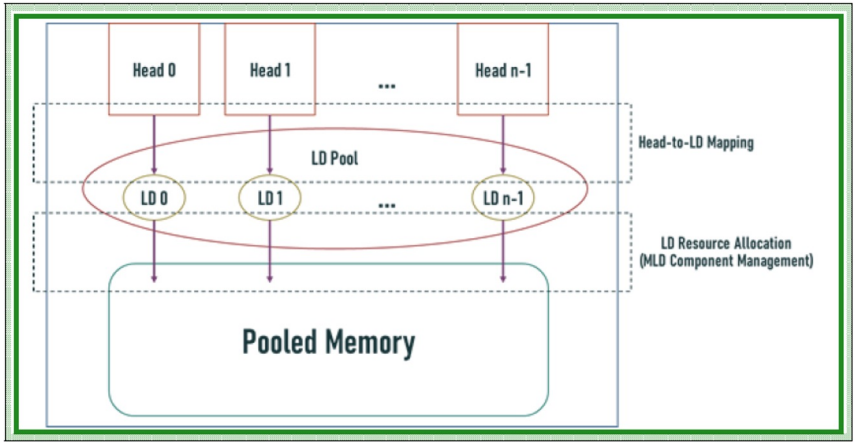
- New pooling option in 3.0
- Type 3 device with multiple CXL “heads”
- Two types of MHD:
  - Multi-Headed-Single Logical Device (MH-SLD)
  - Multi-Headed-Multiple Logical Device (MH-MLD)

New in 3.0 – MHD



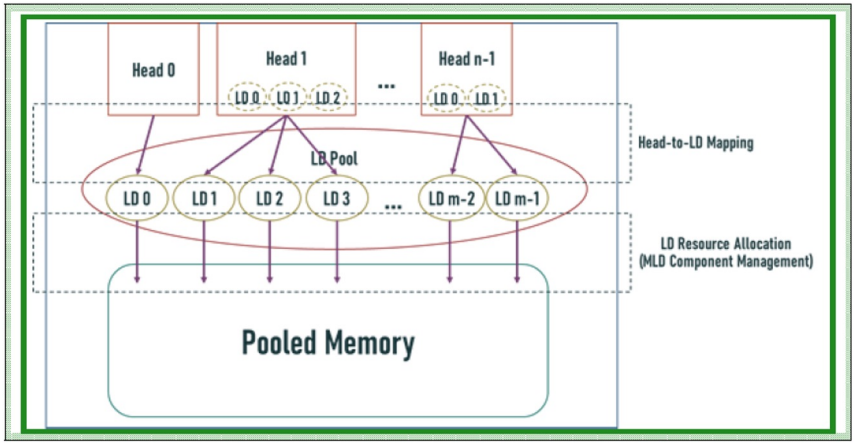
MH-SLD & MH-MLD

- MH-SLD presents an SLD to each head
  - 1:1 mapping of LDs to heads
  - Heads can direct attach to different hosts



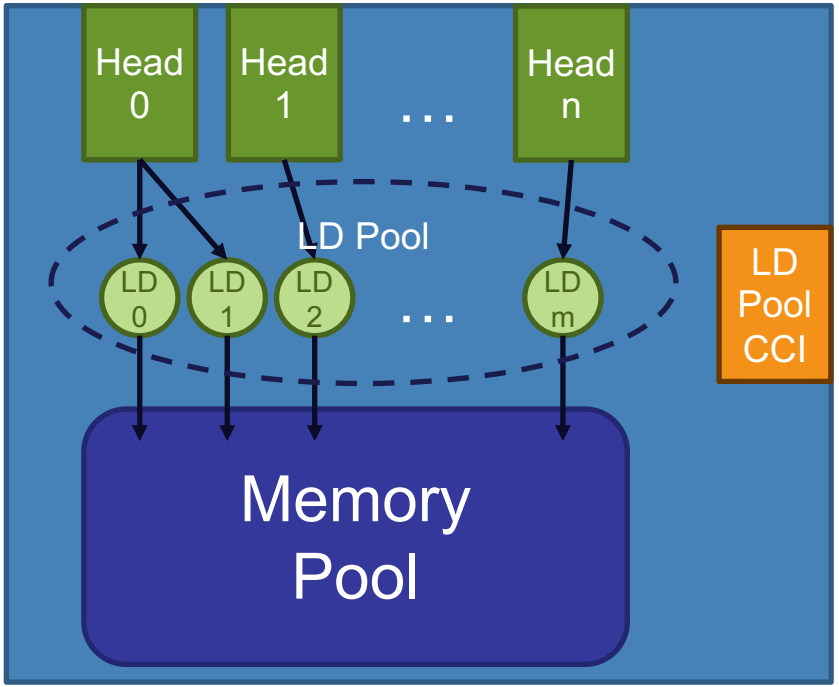
MH-SLD & MH-MLD

- MH-MLD presents an MLD to each head
  - Up to 16 LDs mapped to a single head
  - Requires a switch for MLD functionality



MHD Management

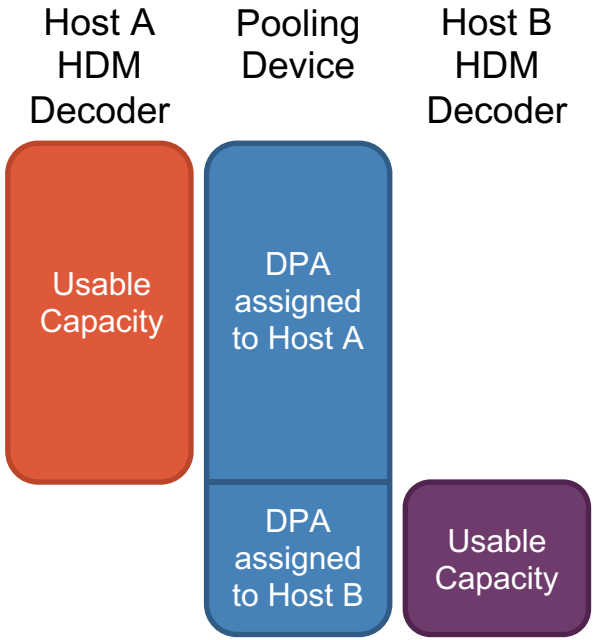
- MHD is a device with multiple LDs
- LD Pool managed from “LD Pool CCI”
- Reuse MLD management framework from 2.0



Dynamic Capacity Devices (DCDs)

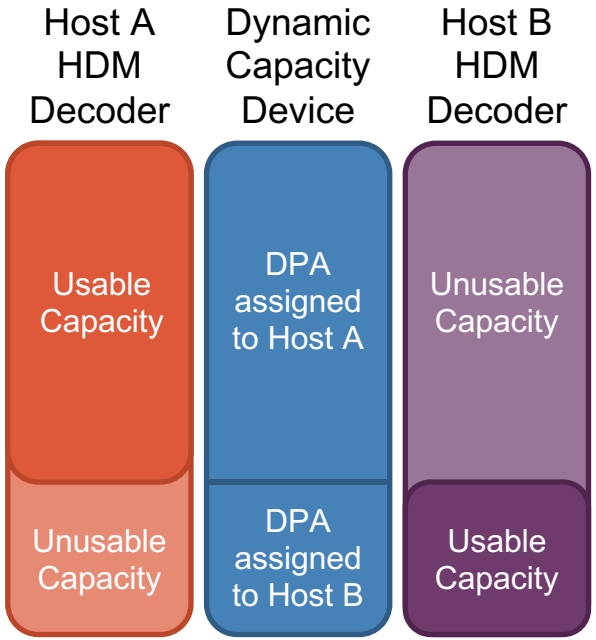
Problem Statement

- Before DCD, changing memory allocation was very disruptive
- Required HDM decoder reprogramming
  - Traffic quiesced
  - System reset likely



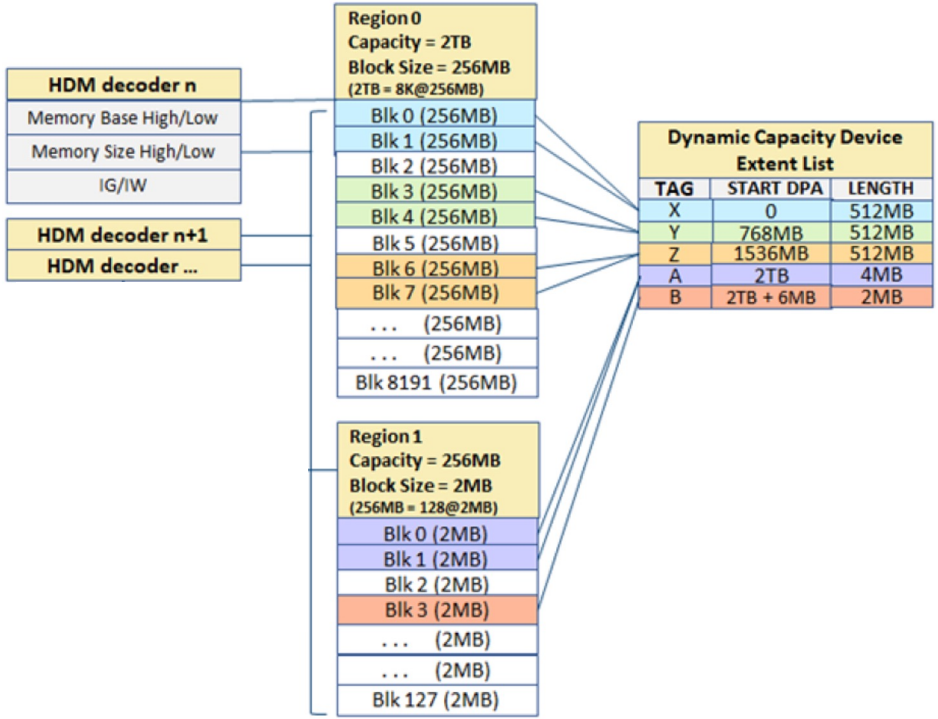
Dynamic Capacity Devices

- Allows memory capacity changes without HDM decoder reprogramming
- DCD presents maximum capacity to each host
  - HDM decoders programmed for full DPA range
  - DCD command set used to discover *actual* memory allocation



Dynamic Capacity Devices

- Capacity defined per host in a block-based “Extent List” – [DPA start, length, tag]
- Up to 8 DC regions with different properties (coherency, sharing, block size)
- Fabric Manager configures DCD allocations



Software how to use ?

Same as today CXL connected memory no changes !

Usage model will depend on memory characteristics:

- Bandwidth
- Latency

Various approach:

- Extra memory capacity
- Tiered memory
- Persistent storage

New usage model:

- Shared memory across hosts

Host Software Requirements

- **Linux kernel support memory hotplug & hotremove today**
- Need Dynamic Capacity Driver in Linux kernel
- Policy should be implemented in userspace
  - When to request memory (hotplug)
  - When to release memory (hotremove)
- OS improvement to make hotplug & hotremove faster (keep region map, …)
- OS improvement to avoid memory pinning (which block hotremove)
  - Linux kernel already have some of that (zone movable comes with tradeoff)

Manage Memory at Infrastructure Level

- Memory becomes a resource that can be re-allocated dynamically to host
- Infrastructure software (host fleet management software) can allocate resources:
  - Hotplug memory to host that are running out
  - Hotremove memory from host that not using all the memory
- Enable memory capacity load balancing across host:
  - Improve TCO by avoiding memory capacity over-provisioning
- Industry to work on **Open Source low level software**
- Memory management policy part of the fleet management software

Call to Action

- Consider joining the CXL consortium if you want to develop Hardware
- Join the Linux Kernel CXL mailing list for software discussion
- First Multi-Headed Hardware demonstration
- Dynamic Capacity Hardware in the coming year
- CXL specification freely available

Join us on this journey !

Thank you!

EMPOWERING OPEN.

OCP GLOBAL SUMMIT

OCTOBER 18-20, 2022
SAN JOSE, CA