Flash Memory Summit

An OCP reference system for CXL-enabled compute disaggregation

Presenter: Siamak Tavallaei
CXL Advisor to the Board, CXL Consortium
Server Project Incubation Committee Rep., OCP Foundation

Outline

5 min	A quick reminder of CXL specification
10 min	System-level use case examples and challenges
5 min	CXL-related activities within OCP Server Project

CXL Technical Work Groups

CXL Consortium

https://www.computeexpresslink.org

CXL Board of Directors (BoD)

https://www.computeexpresslink.org/meettheboard
Marketing Work Group (**MWG**)
Technical Task Force (**TTF**)

Active CXL WGs run under Technical Task Force (TTF)

PWG (Protocol WG)

SSWG (System and Software WG)

PHY (Physical and Link Work Group)

MSWG (Memory System WG)

Compliance WG

OCP Server Project

Active OCP Server Project subprojects and workstreams

Drive **contributions** (Base Spec, Design Spec, Products, …)
https://www.opencompute.org/wiki/Server

CMS (Composable Memory System)
https://www.opencompute.org/projects/composable-memory-system

DC-MHS (Datacenter-ready Modular Hardware System)
https://www.opencompute.org/projects/dc-mhs

Extended Connectivity Workstream (for PCIe and CXL)
https://www.opencompute.org/wiki/Server/PCIe_Extended_Connectivity_Requirements_Workstream

ODSA (Open Domain-Specific Architecture)
https://www.opencompute.org/wiki/Server/ODSA

OAI (Open Accelerator Infrastructure)
https://www.opencompute.org/wiki/Server/OAI

HPC (High-performance Computing)
https://www.opencompute.org/wiki/HPC

OCP NIC
https://www.opencompute.org/wiki/Server/NIC

Siamak Tavallaei
CXL Advisor to the Board of Directors, CXL™ Consortium
Aug 8, 2023

Compute Express Link™ (CXL™)
A Coherent Interface for Ultra-High-Speed Transfers

CXL Resources

Webinars:

- Webinar: A look into the CXL device ecosystem and the evolution of CXL use cases
- Webinar: Introducing CXL 3.0: Enabling composable systems with expanded fabric capabilities
- Webinar: CXL 1.1 vs. CXL 2.0 – What's the difference?
- Webinar Archive: https://www.computeexpresslink.org/webinars

Blogs:

- Upcoming Webinar: A Look into the CXL™ Device Ecosystem and the Evolution of CXL Use Cases
- CXL 3.0 Webinar Q&A Recap
- CXL™ Consortium Member Spotlight: UnifabriX
- Blog Archive: https://www.computeexpresslink.org/blog

White Papers:

- CXL 3.0 specification
- An Overview of Reliability, Availability, and Serviceability (RAS) in Compute Express Link™ 2.0

CXL 3.0 Spec Feature Summary

Features	CXL 1.0 / 1.1	CXL 2.0	CXL 3.0
Release date	2019	2020	Aug 2022
Max link rate	32GT/s	32GT/s	64GT/s
68-byte Flit (up to 32 GT/s)	✓	✓	✓
Type 1, Type 2, and Type 3 Devices	✓	✓	✓
Memory Pooling w/ MLDs		✓	✓
Global Persistent Flush		✓	✓
CXL IDE		✓	✓
Switching (Single-level)		✓	✓
Switching (Multi-level)			✓
Multiple Type 1 & Type 2 Devices per root port			✓
Direct memory access for peer-to-peer			✓
256-byte Flit (up to 64 GT/s)			✓
256-byte Flit (Enhanced coherency)			✓
256-byte Flit (Memory sharing)			✓
256-byte Flit (Fabric capabilities)			✓

Not supported

✓ Supported

CXL™ Approach

Coherent Interface

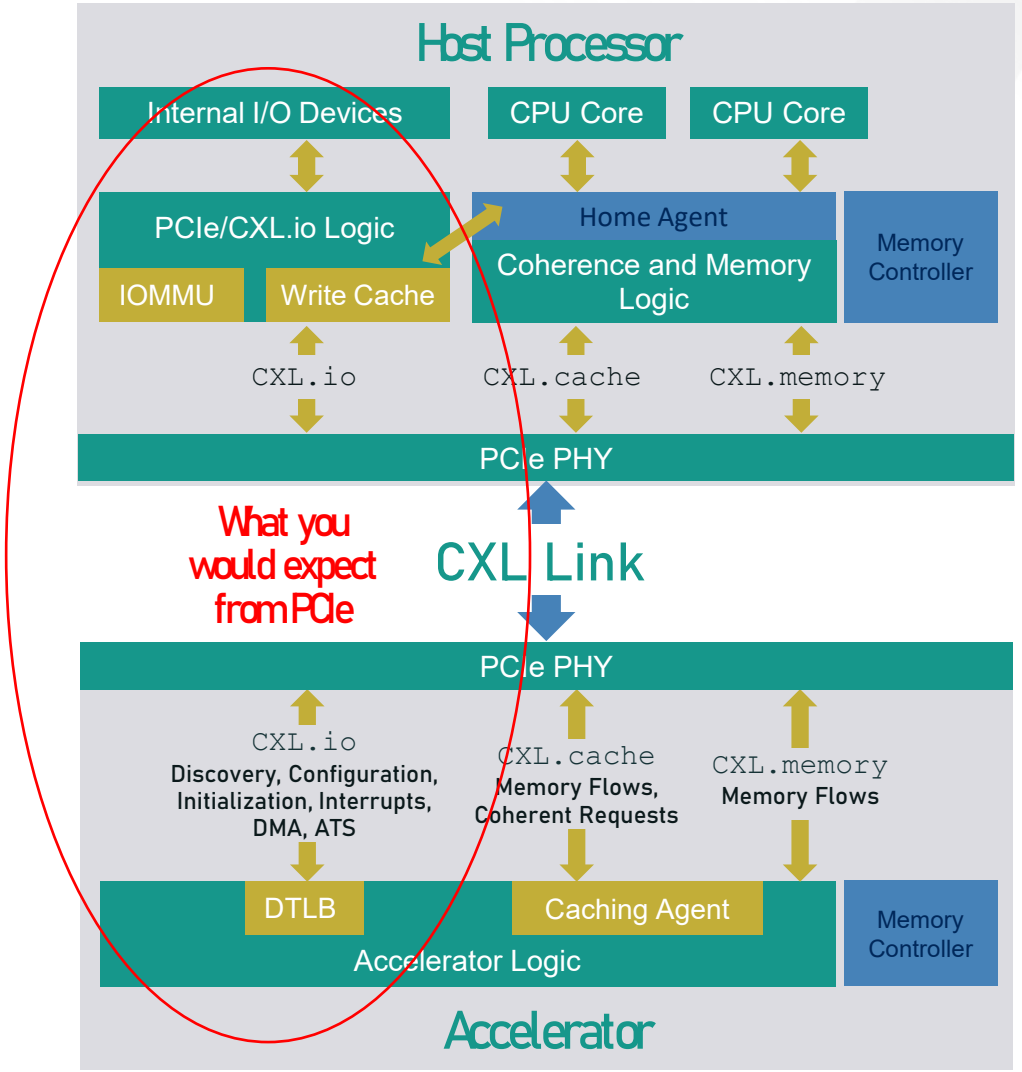
- Leverages PCIe with three multiplexed protocols
- Built on top of PCIe® infrastructure

Low Latency

- CXL.Cache/CXL.Memory targets near CPU cache coherent latency (<200ns load to use)

Asymmetric Complexity

- Eases the burden of cache coherence interface designs for devices



CXL™ Approach

Coherent Interface

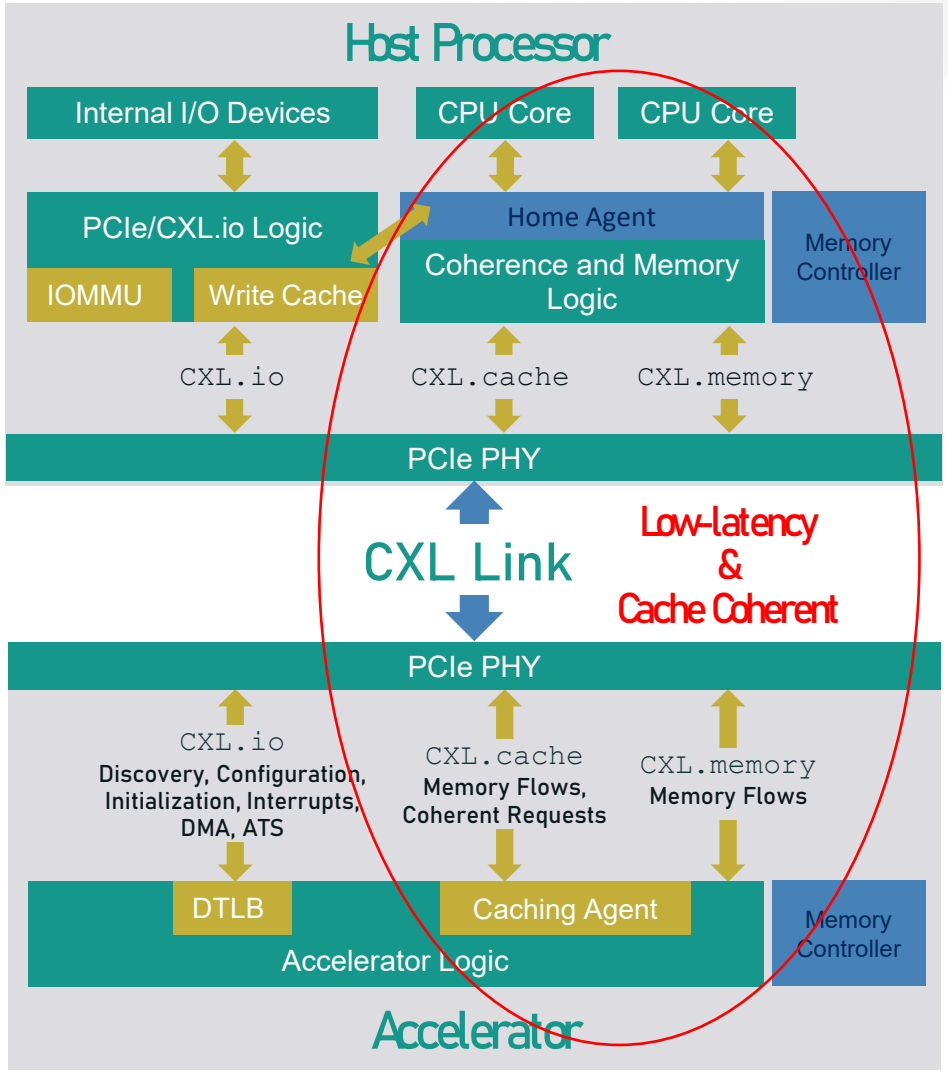
- Leverages PCIe with three multiplexed protocols
- Built on top of PCIe® infrastructure

Low Latency

- `CXL.Cache/CXL.Memory` **targets near CPU cache coherent latency (<200ns load to use)**

Asymmetric Complexity

- Eases the burden of cache coherence interface designs for devices



CXL™ Approach

Coherent Interface

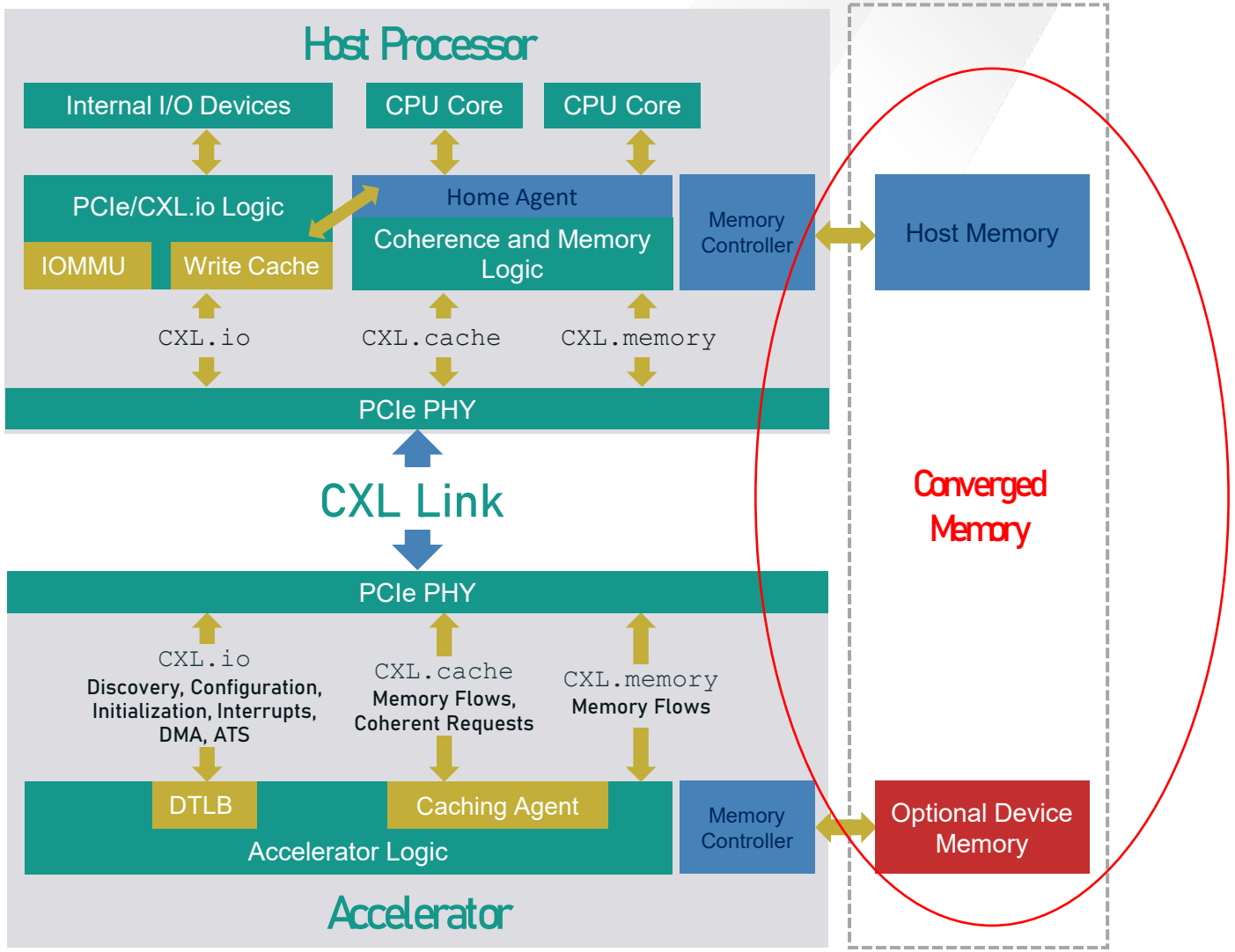
- Leverages PCIe with three multiplexed protocols
- Built on top of PCIe® infrastructure

Low Latency

- CXL.Cache/CXL.Memory targets near CPU cache coherent latency (<200ns load to use)

Asymmetric Complexity

- Eases the burden of cache coherence interface designs for devices



A converged memory environment
is the main key

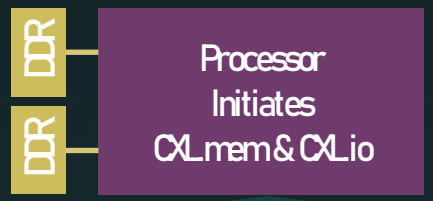
for ease of software programing!

and efficient data-movement!

Recap: CXL 1.0/1.1 Representative Use Cases

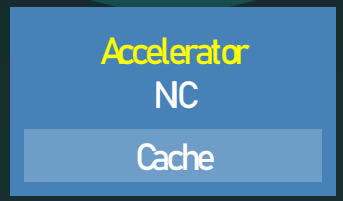
Caching Devices / Accelerators

TYPE 1



CXL

PROTOCOLS
- CXL.io
- CXL.cache

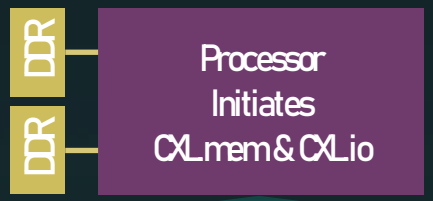


Initiates CXL Cache

USAGES
- PGAS NIC
- NIC atomics

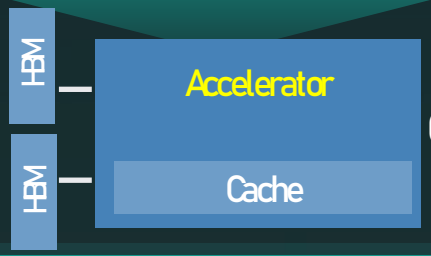
Accelerators with Memory

TYPE 2



CXL

PROTOCOLS
- CXL.io
- CXL.cache
- CXL.memory

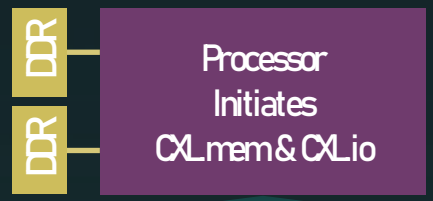


Initiates CXL Cache

USAGES
- HDM-D: Device-Coherent HDM region type
- GP GPU
- Dense computation

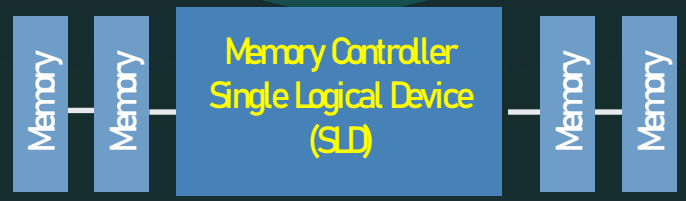
Memory Buffers

TYPE 3



CXL

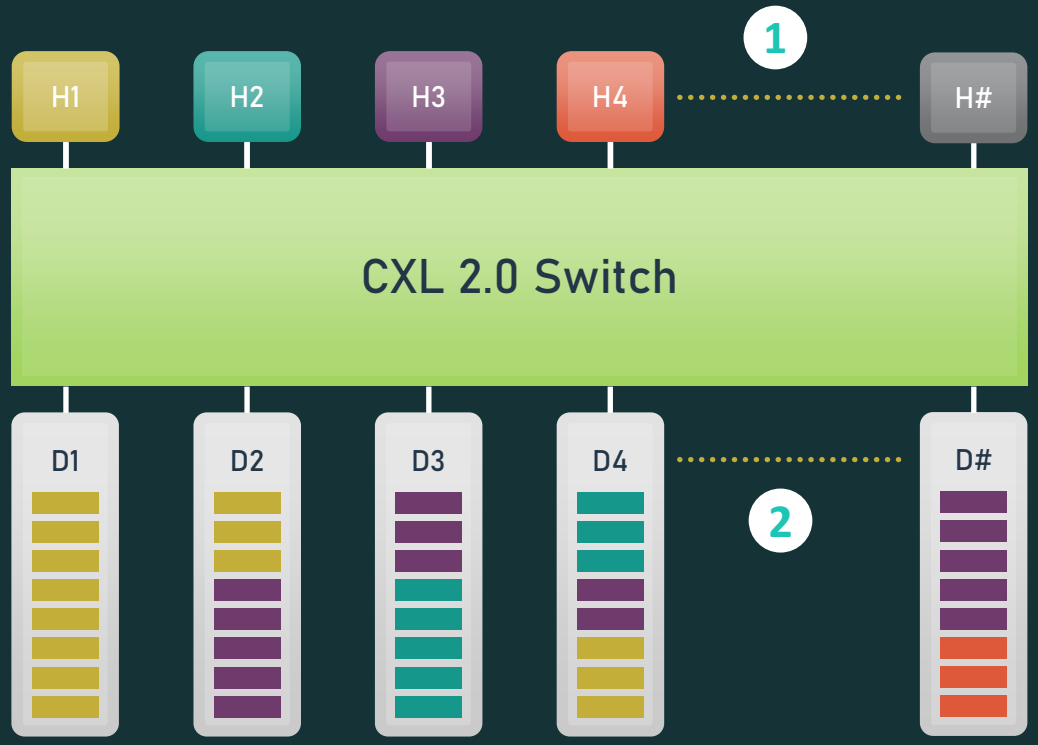
PROTOCOLS
- CXL.io
- CXL.memory



USAGES
- HDM-H: Host-only Coherent HDM region type
- Memory BW expansion
- Memory capacity expansion
- Storage-class memory

Recap: CXL 2.0 FEATURE SUMMARY

CXL 2.0 Feature Summary
Memory Pooling w/ MLDs
Global Persistent Flush (storage)
CXL IDE (security)
Switching (single-level)



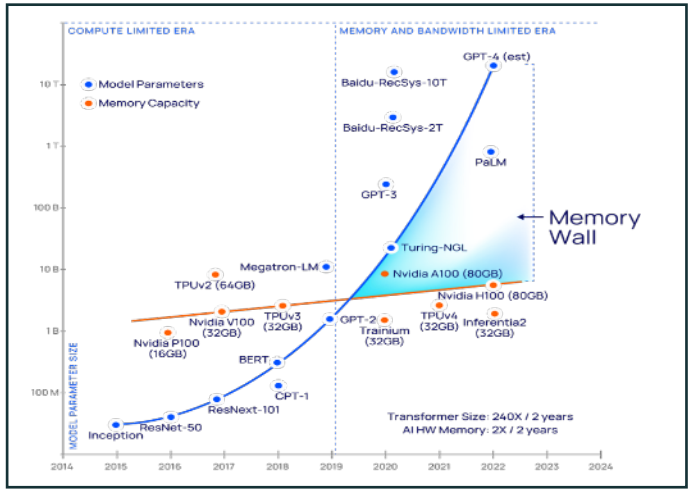
SLD MLD: MEMORY POOLING

1. Device memory can be allocated across multiple hosts
2. Multi Logical Devices (MLD) allow finer grain memory allocation
3. Persistence Flows
4. Pooling of accelerators Hot-plug flows

Industry trends

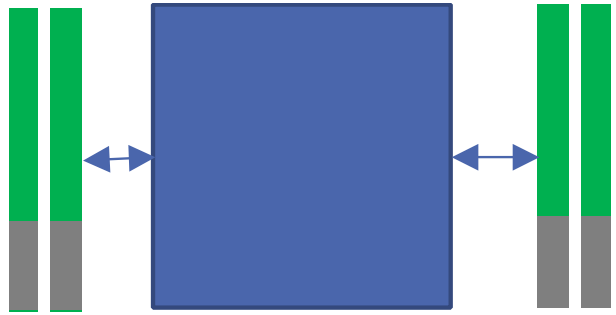
- Use cases keep driving the need for higher IO and memory bandwidth:
  e.g., high performance accelerators, system memory, SmartNIC etc.
- CPU capability requiring more memory capacity and bandwidth per core
- Efficient peer-to-peer resource-sharing & messaging across multiple domains
- Need to overcome memory bottlenecks due to CPU pin and thermal constraints

AI Memory Wall:



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

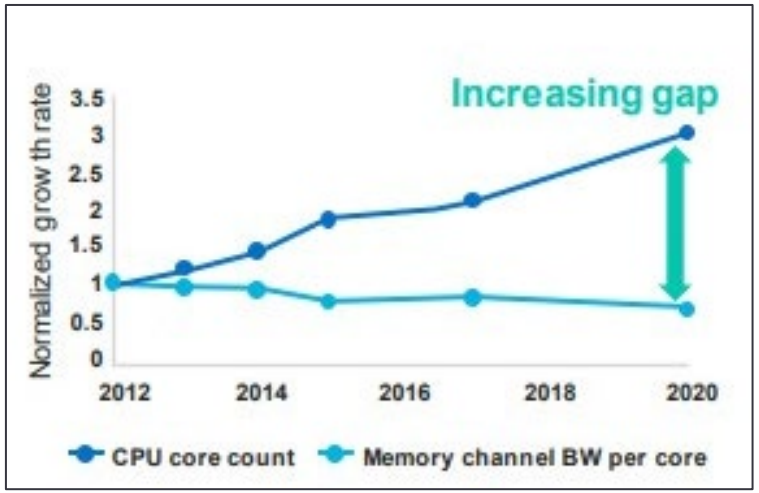
Unutilized Memory:



"Up to 30% of server memory can be stranded" – Microsoft Azure[1]

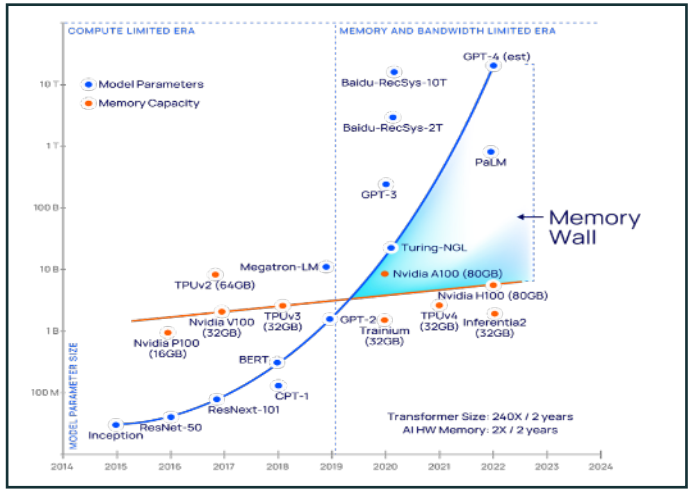
Source: https://ieeexplore.ieee.org/document/10034802

More Cores:



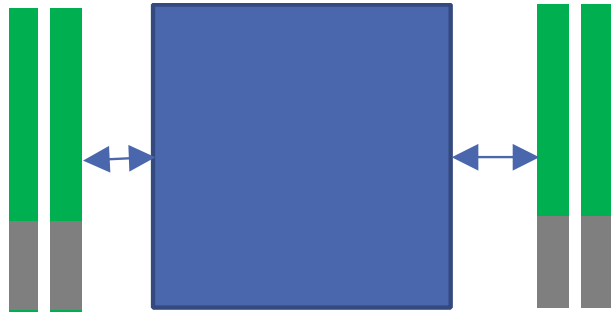
Source: Meta OCP Presentation Nov 2021

AI Memory Wall: Need More High-BW Mem



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

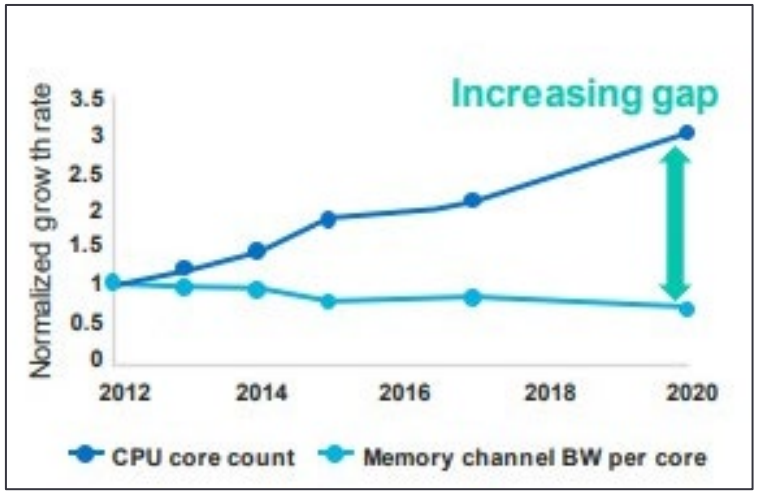
Unutilized Memory: Need Access



"Up to 30% of server memory can be stranded" – Microsoft Azure[1]

Source: https://ieeexplore.ieee.org/document/10034802

More Cores: Need more Memory BW



Source: Meta OCP Presentation Nov 2021

CXL 3.0 Specification

Expanded capabilities for increased scale and optimized resource utilization

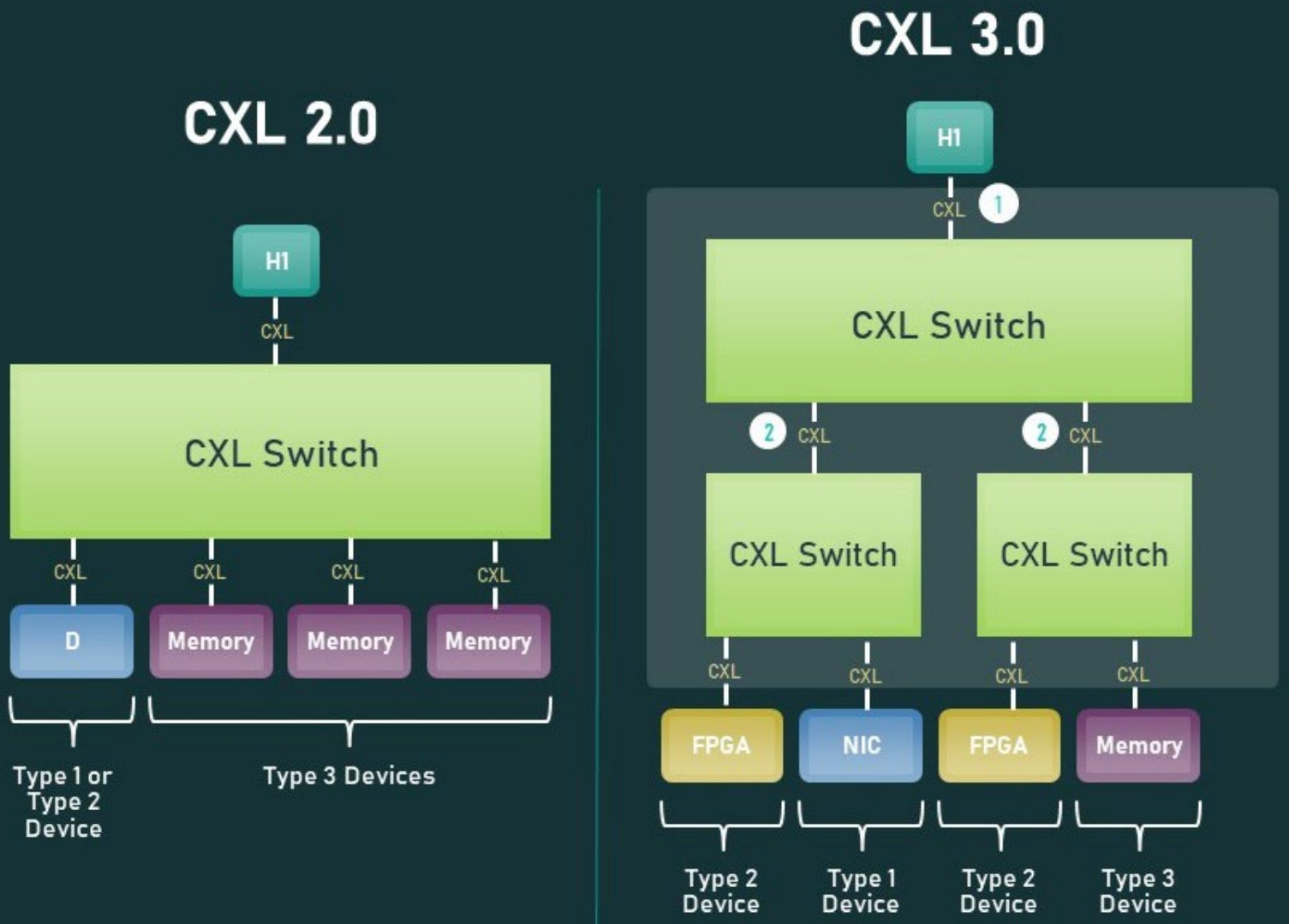
- Double the bandwidth PCIe® 6.0 (Bandwidth) (Zero added latency over CXL 2.0)
- Enhanced Switching (Connectivity)
  - Direct memory & Peer-to-Peer access by devices
  - Multiple Type 1 & Type 2 devices per root port
  - Multi-level switching
- Fabric capabilities (Composability)
  - Fabric-attached memory (GFAM)
  - Enhanced fabric management framework
  - Support for composable disaggregated infrastructure
- New Capabilities
  - Multi-headed devices
  - New symmetric memory capabilities
  - Enhanced memory pooling and sharing
  - Near-memory processing
  - Improved software capabilities
- Fully backward-compatible with:
  - CXL 2.0
  - CXL 1.1
  - CXL 1.0

CXL 3.0 is a huge step function with fabric capabilities while maintaining full backward-compatibility with prior generations

Increase

Connectivity

CXL 3.0: Multiple Level Switching, Multiple Type-1/2/3

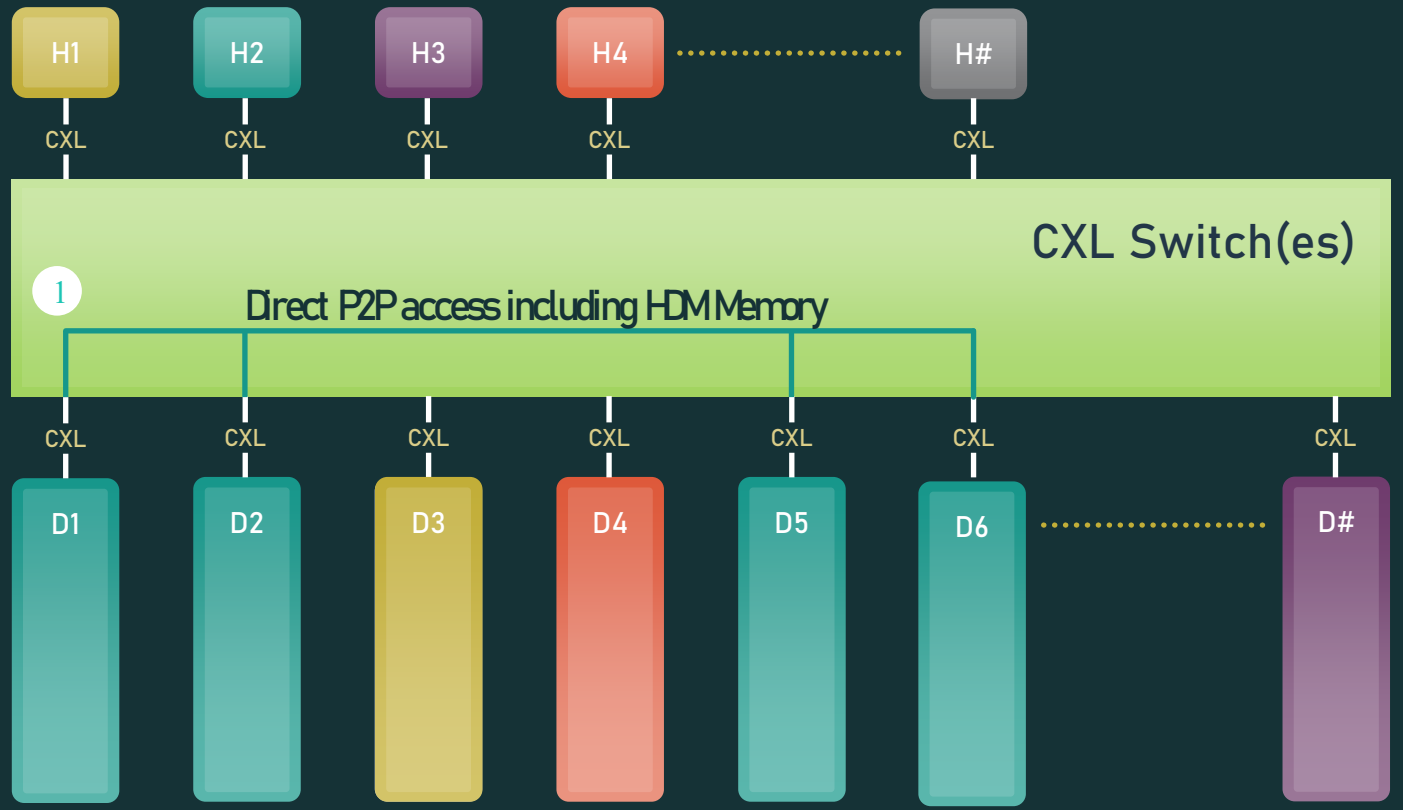


1. Each Host may connect to more than one device type via each Root Port
   - up to 16 CXL.cache devices

2. Multiple switch levels
   - Cascade
   - Supports fanout of all device types

Better Connectivity
&
Remove Bottlenecks

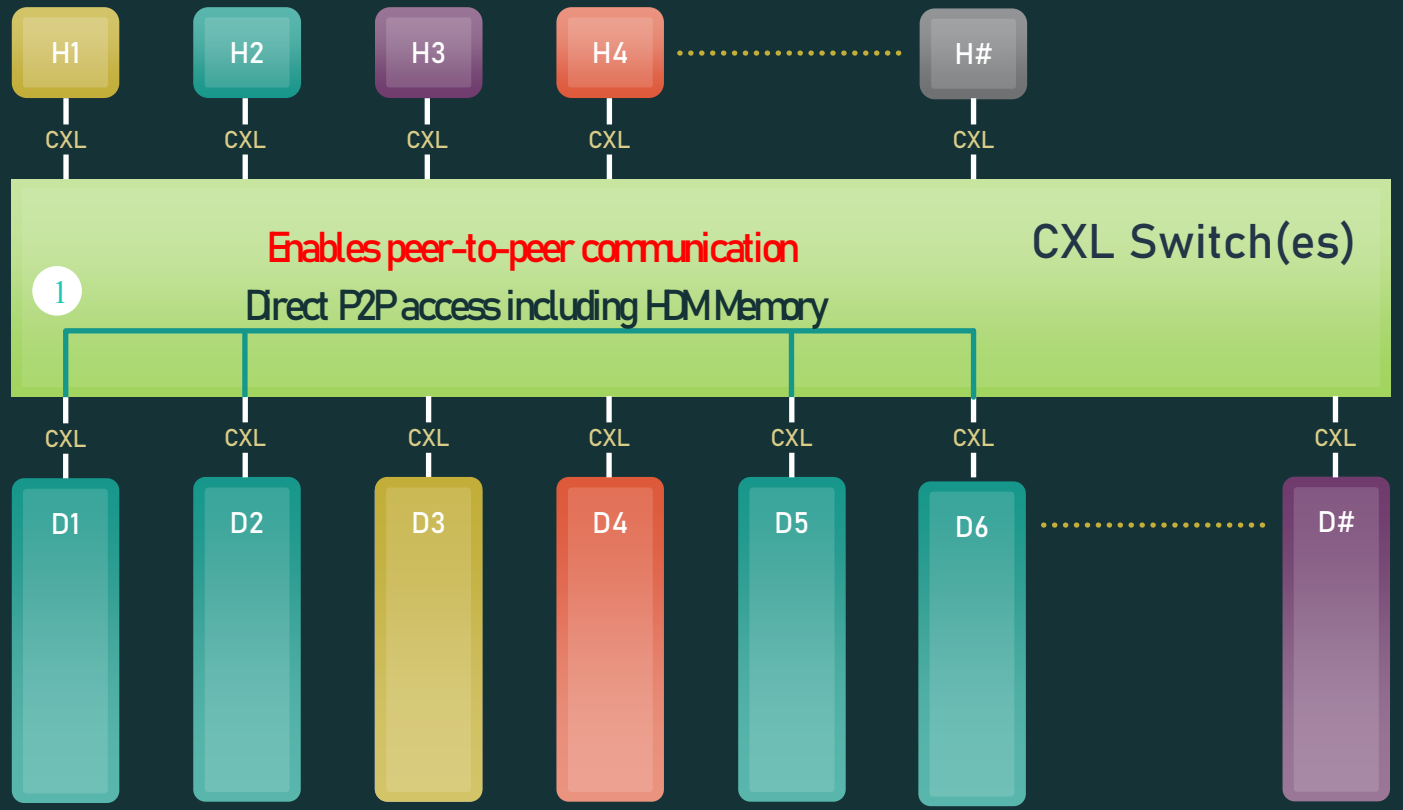
CXL 3.0 Protocol Enhancements (UIO and BI) for Device to Memory Connectivity



CXL 3.0 enables non-tree topologies and peer-to-peer communication (P2P) within a virtual hierarchy of devices

- A Virtual Hierarchy is the association of devices that maintain a coherency domain

CXL 3.0 Protocol Enhancements (UIO and BI) for Device to Memory Connectivity



CXL 3.0 enables non-tree topologies and peer-to-peer communication (P2P) within a virtual hierarchy of devices

- A Virtual Hierarchy is the association of devices that maintain a coherency domain

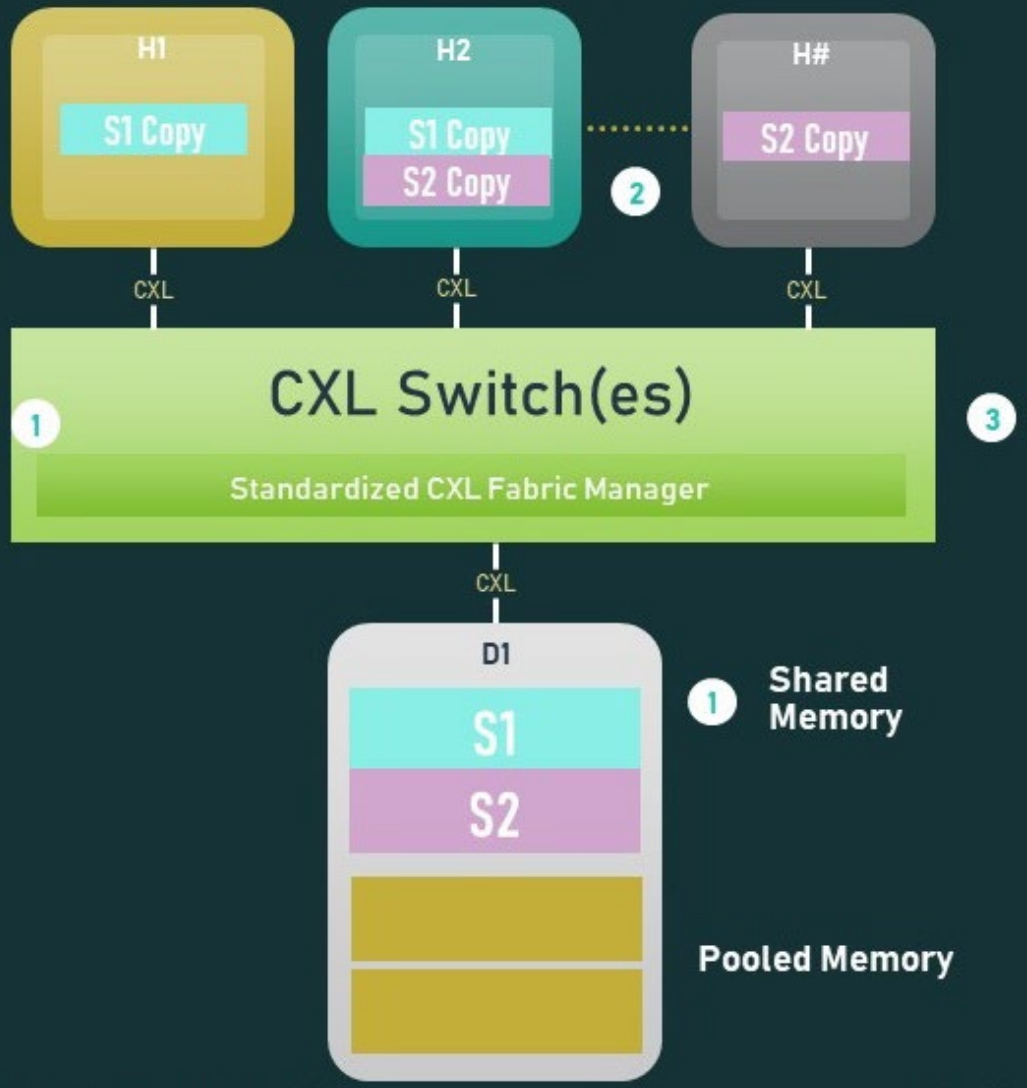
- P2P to HDM-DB memory is I/O Coherent via a new Unordered I/O (UIO) Flow in CXL.io
- Type 2 & 3 devices with attached HDM-DB memory monitor memory accesses. If coherency conflict, they generate a new Back-Invalidation flow (BI) (CXL.mem) to the Host(s) to ensure coherency

HDM-DB: Device-Coherent using Back-Invalidation HDM region type for device-attached Memory (DAMem). Can be used by Type 2 or Type 3 devices

Provide

New Capabilities

CXL 3.0: Memory Sharing

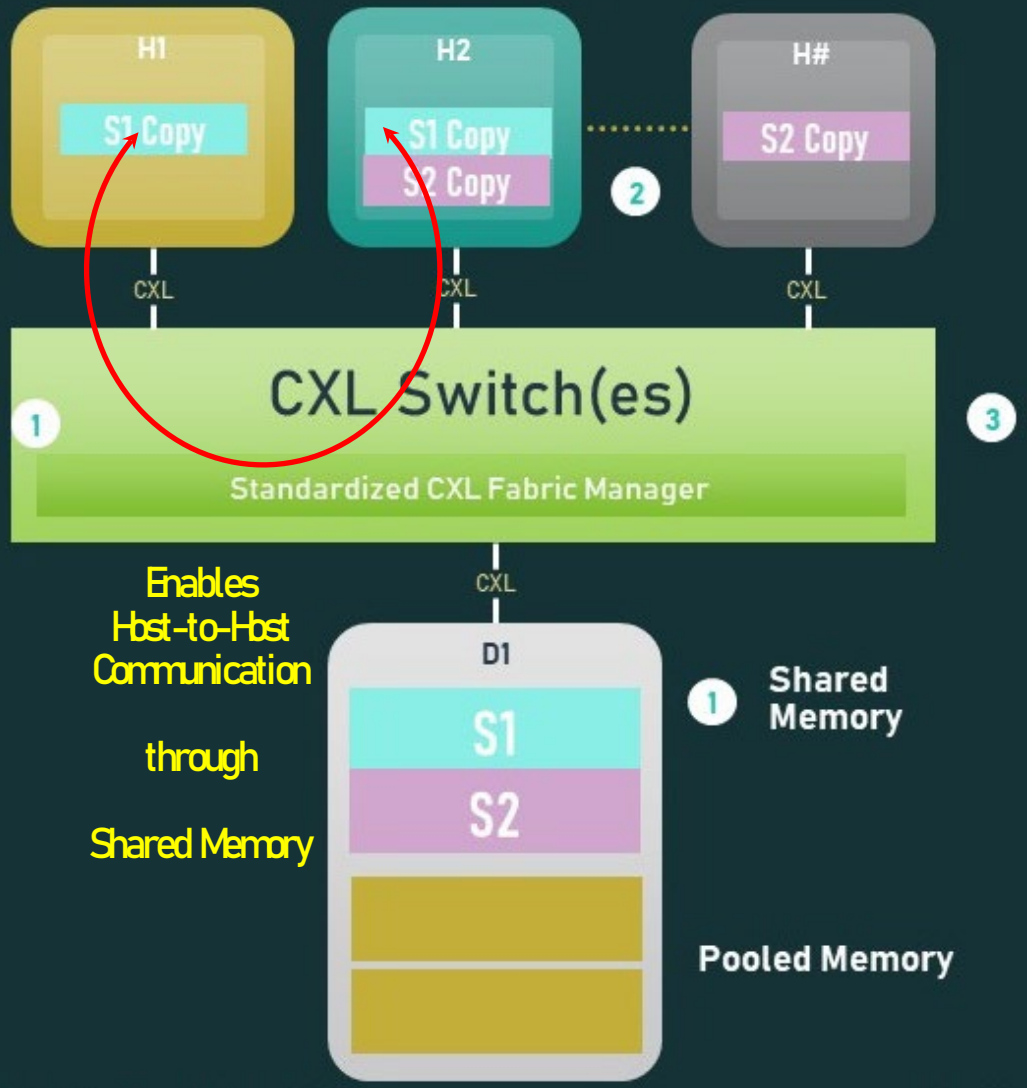


1. Device memory can be shared by all hosts to increase data-flow efficiency and improve memory utilization

2. Host can have a coherent copy of the shared region or portions of shared region in host cache

3. CXL 3.0 defined mechanisms to enforce hardware cache-coherency

CXL 3.0: Memory Sharing



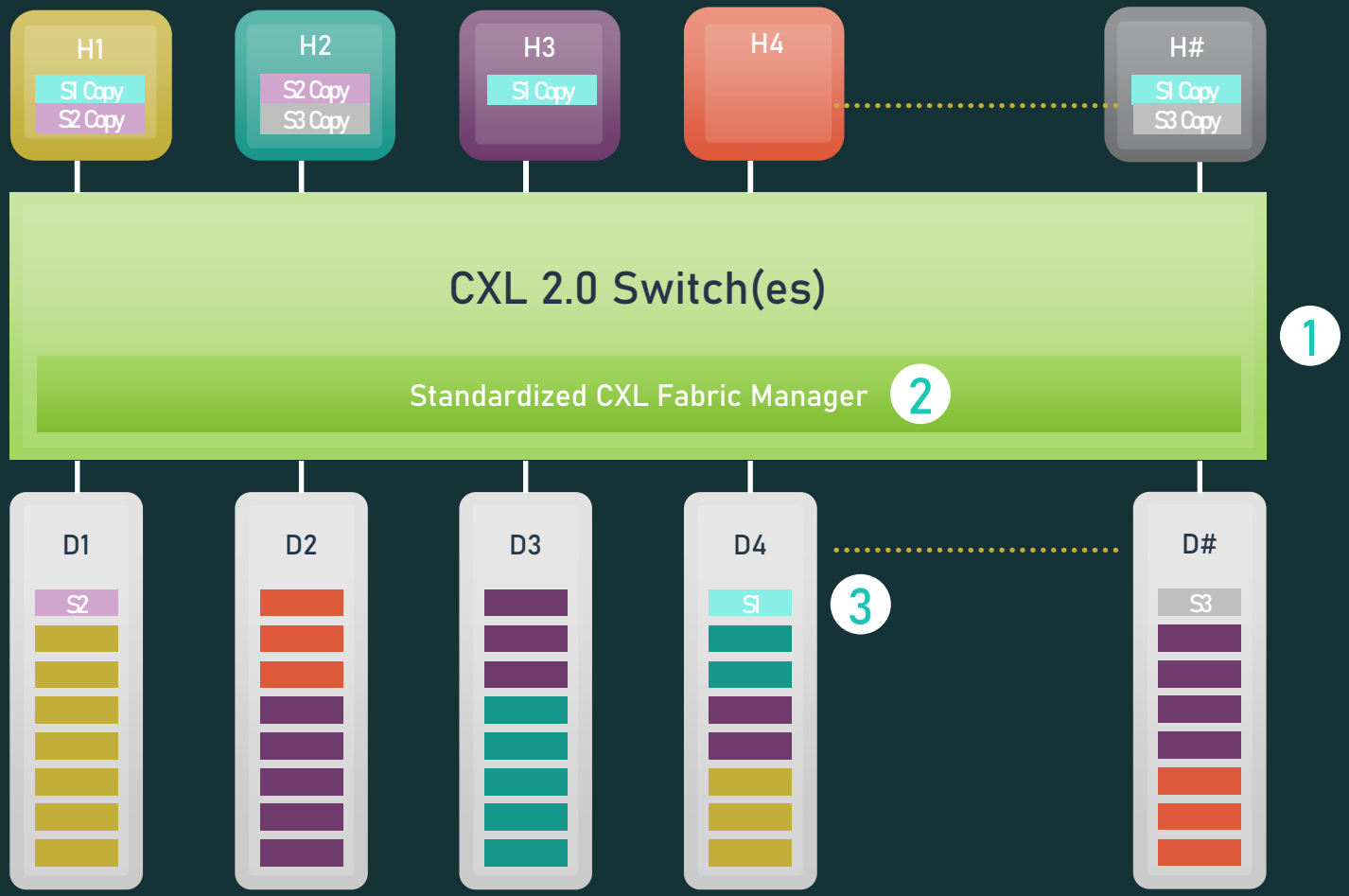
1. Device memory can be shared by all hosts to increase data-flow efficiency and improve memory utilization

2. Host can have a coherent copy of the shared region or portions of shared region in host cache

3. CXL 3.0 defined mechanisms to enforce hardware cache-coherency

CXL 3.0: Memory Pooling & Sharing

A Fabric for Disaggregated and Compostable Architectures



1. Expanded use case showing memory sharing and pooling

2. CXL Fabric Manager is available to setup, deploy, and modify the environment

3. Shared Coherent Memory across hosts using hardware coherency (directory + Back-Invalidate Flows) (BI).

   Allows one to build large clusters to solve large problems through shared memory constructs.

   Defines a Global Fabric-Attached Memory (G-FAM) which can provide access to up to 4095 entities

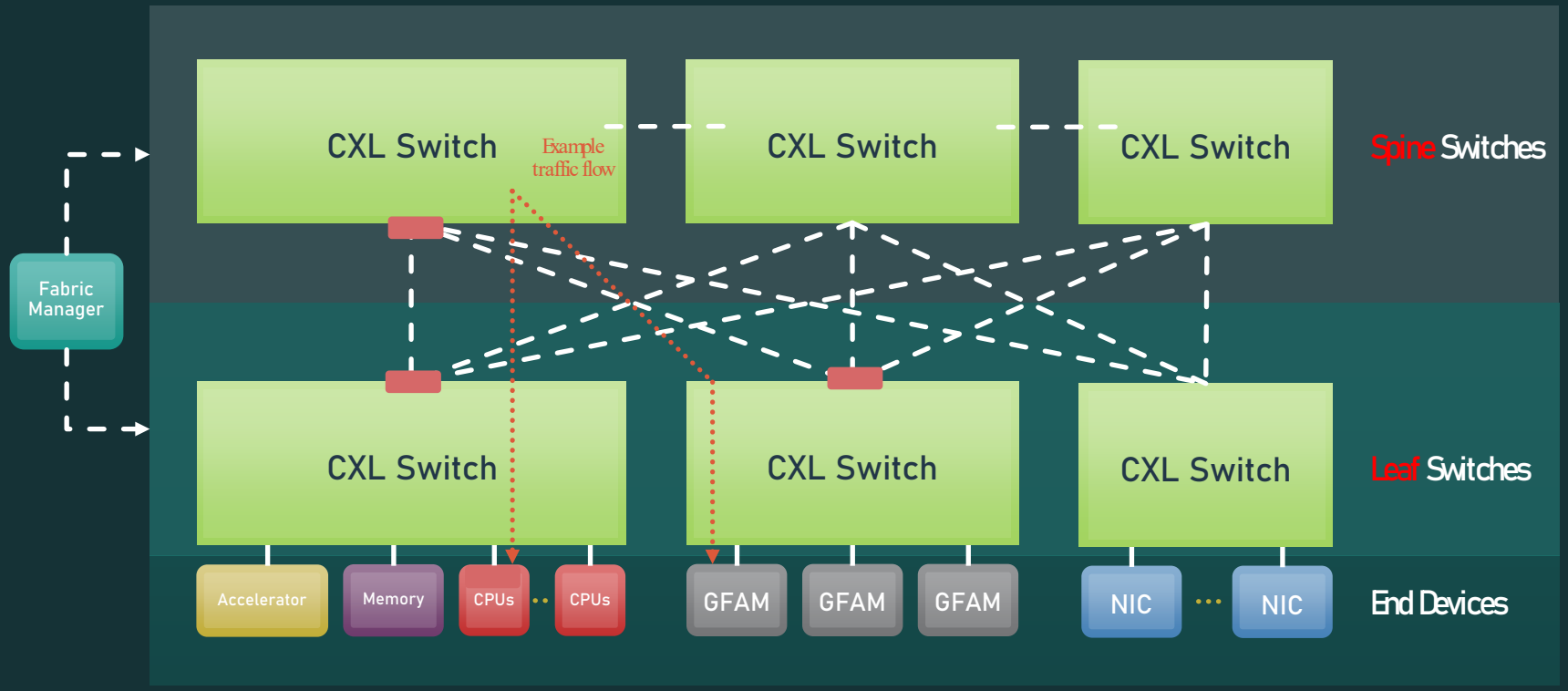
The CXL 3.0 Spec

allows for a large

interconnected Fabric

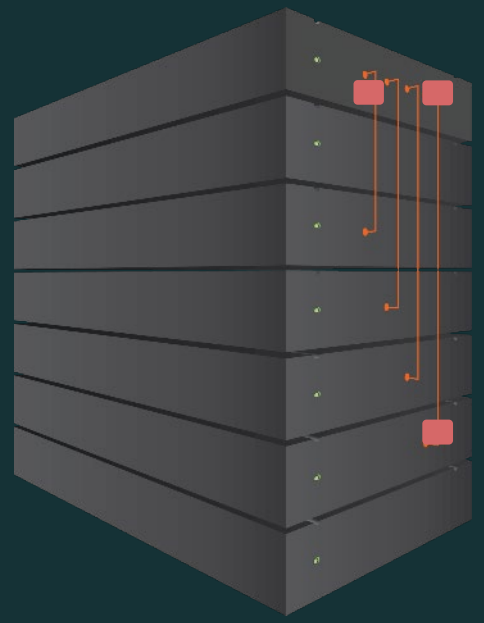
CXL 3.0 Fabrics

Composable Systems with Spine/Leaf Architecture



CXL 3.0 Fabric Architecture

- Interconnected Spine Switch System
- Leaf Switch NIC Enclosure
- Leaf Switch CPU Enclosure
- Leaf Switch Accelerator Enclosure
- Leaf Switch Memory Enclosure



OCP

is the place where we

Realize

Technologies

into

Integrated **Systems**

Community-driven hyperscale innovation for all.

OPEN Compute Project®

Composable Architecture

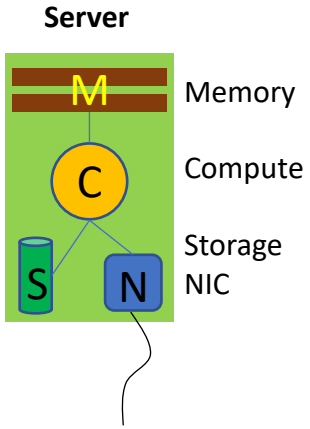
new paradigm for

disaggregated computing!

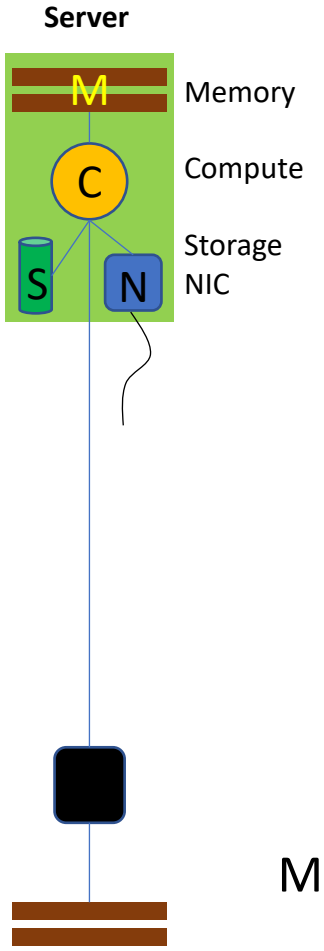
HPC, AI/ML, In-memory Databases

Community-driven hyperscale innovation for all.

OPEN Compute Project®

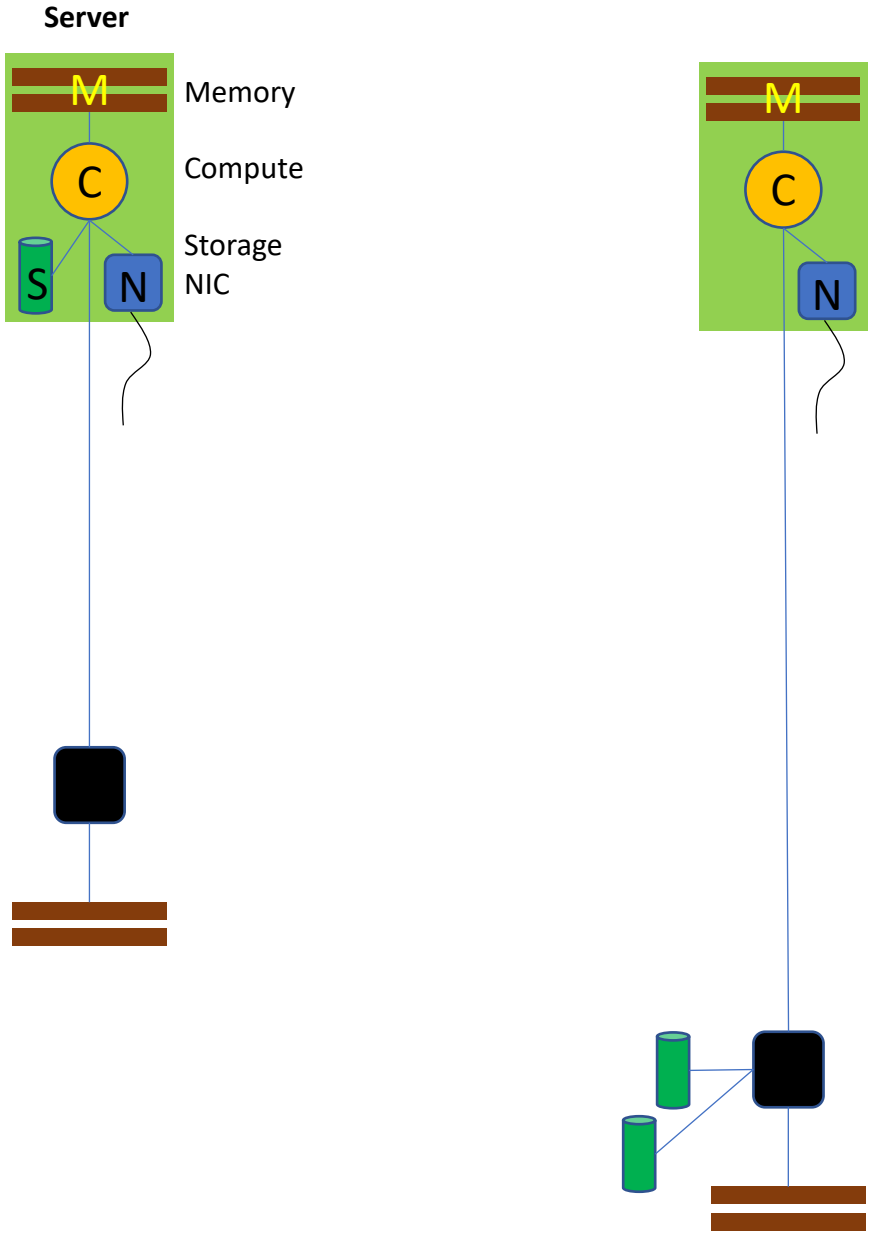


A traditional Server



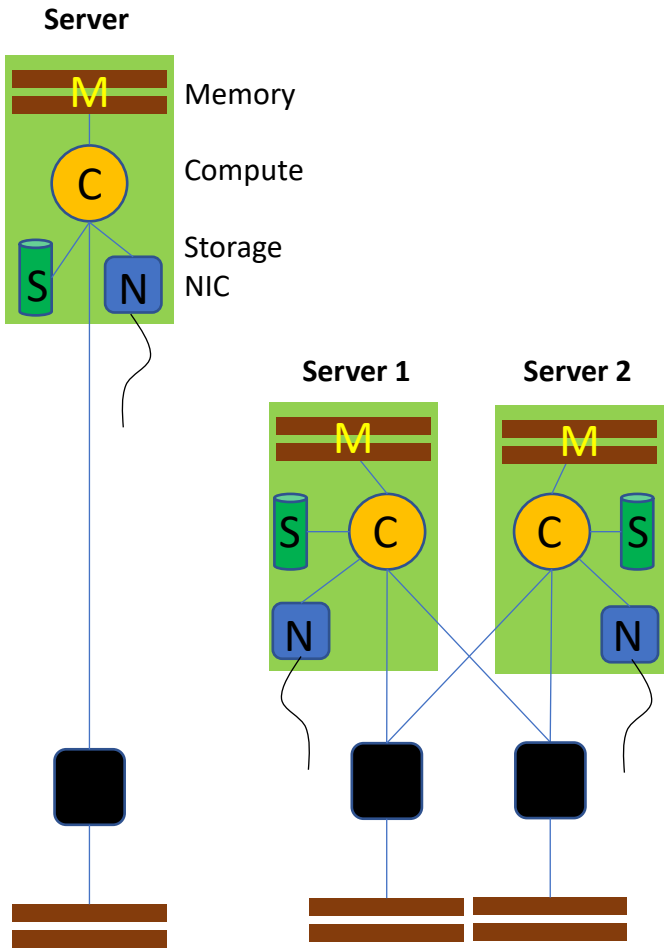
CXL enables

Memory Bandwidth and
Capacity Expansion



CXL enables

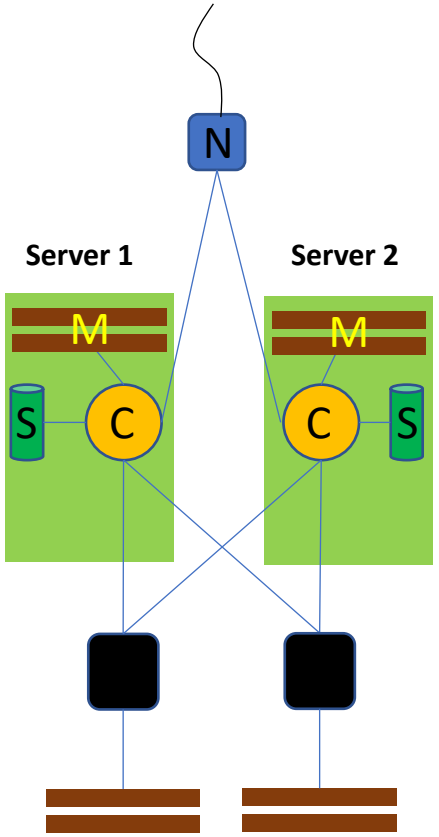
supporting different tiers of memory or storage technologies

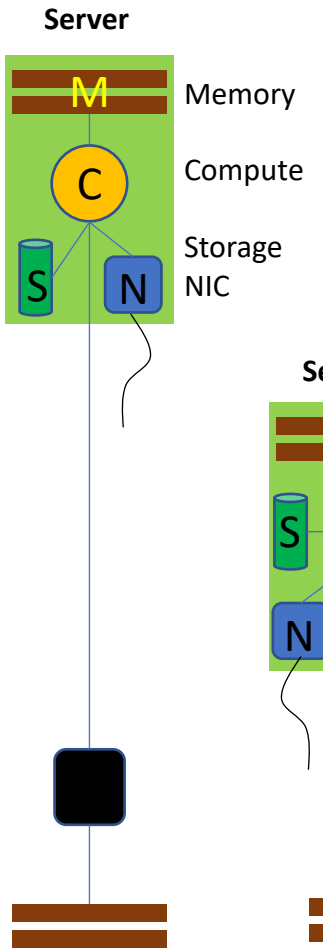


CXL enables

Interconnected Servers

Pooled/Shared Resources

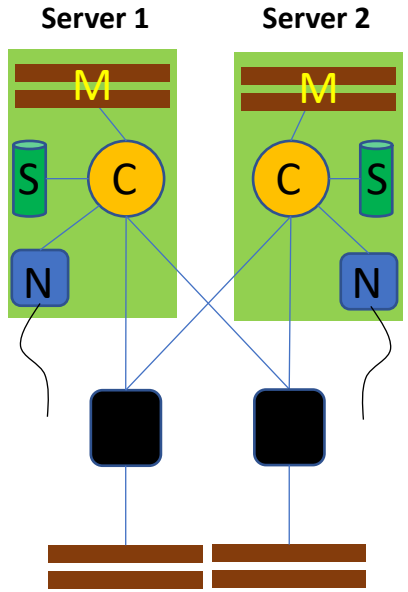


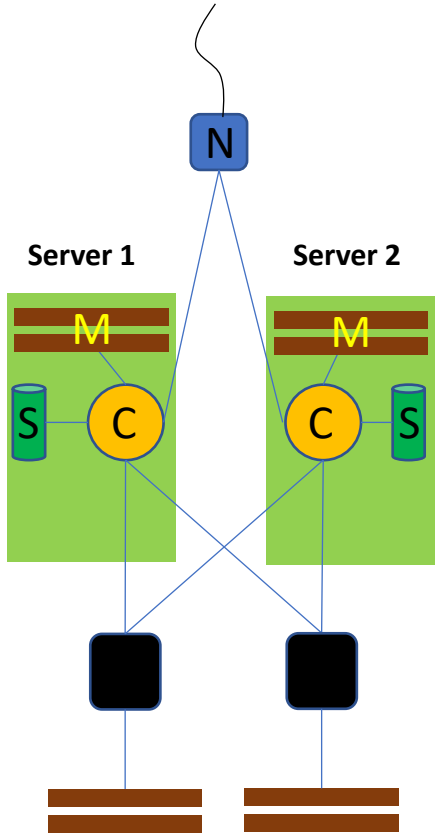


CXL enables

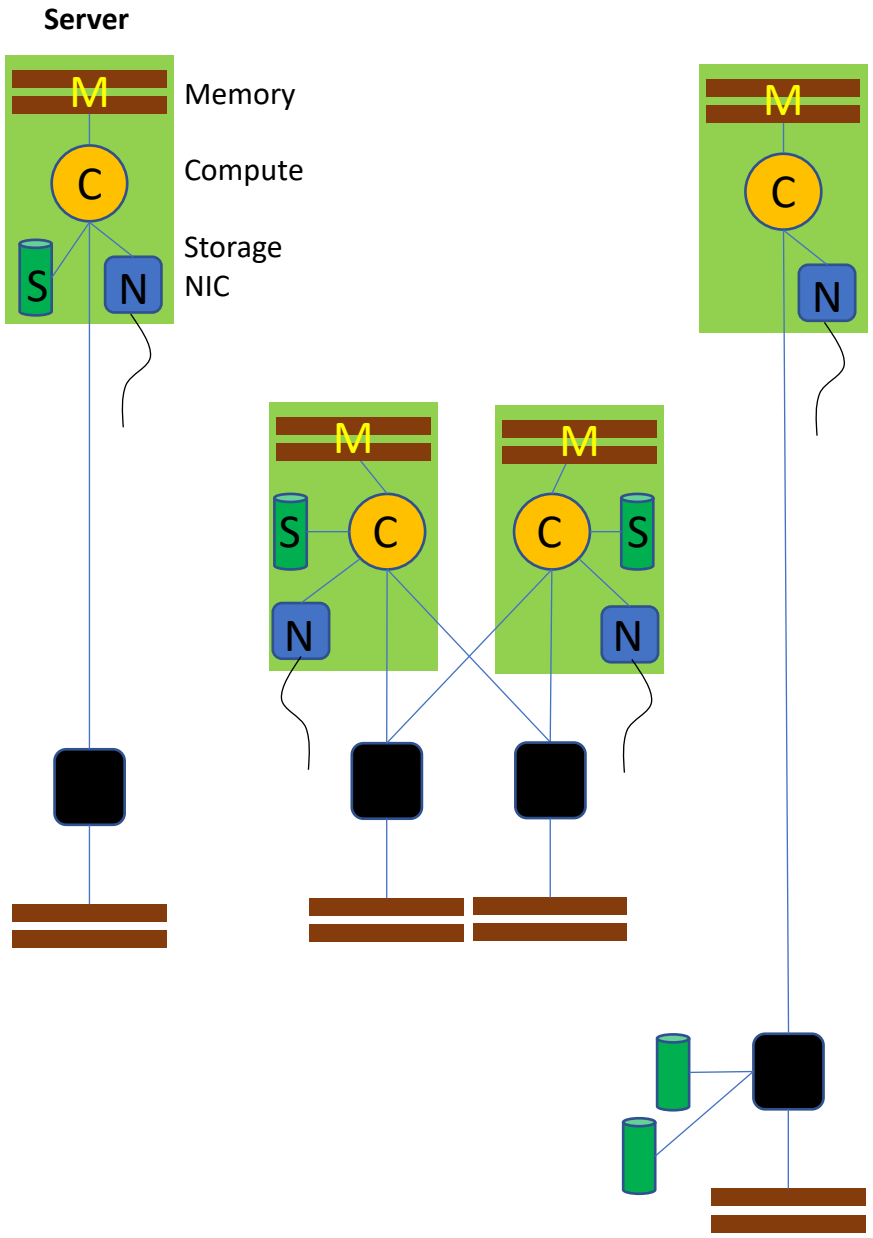
Interconnected Servers

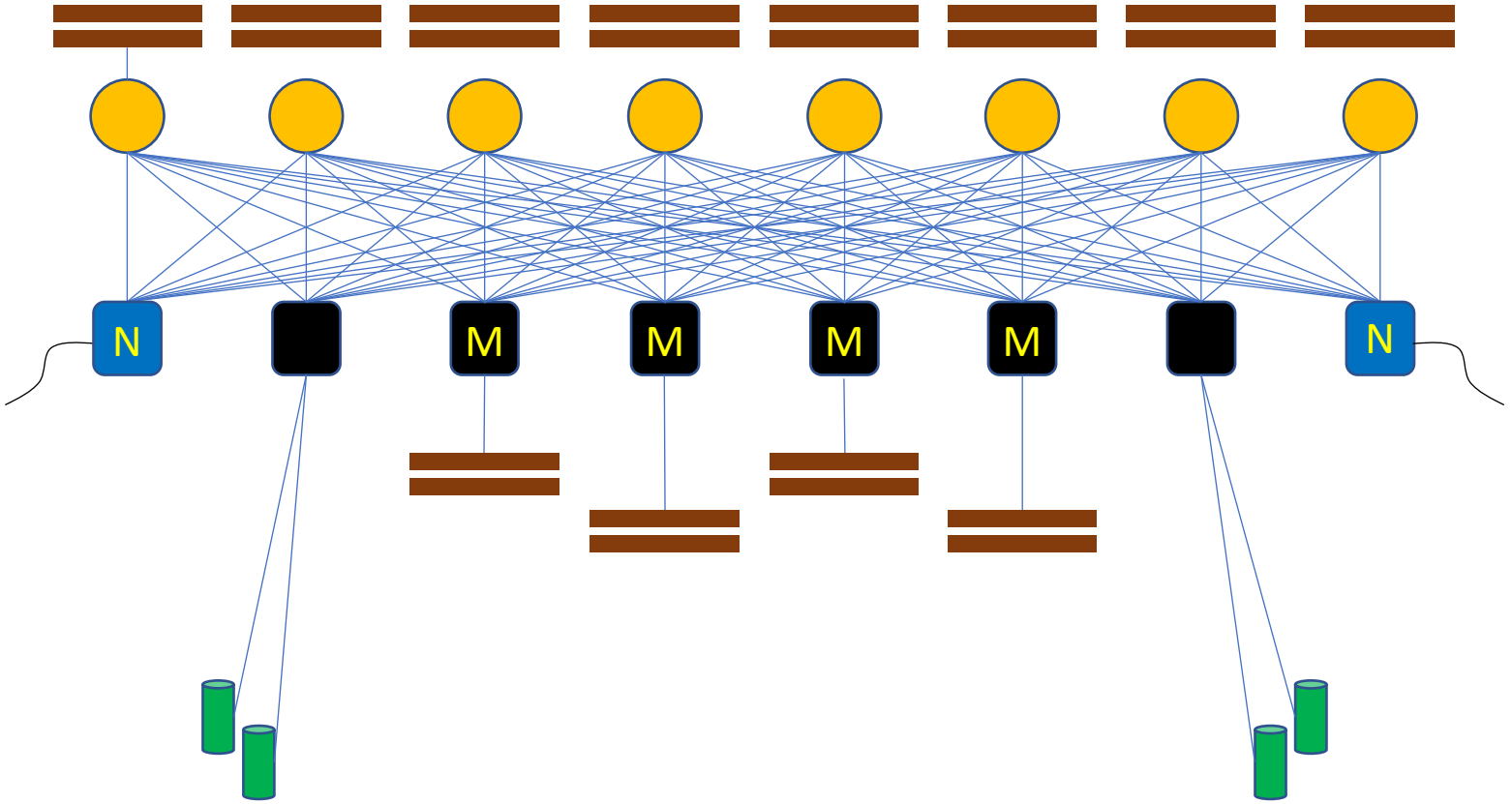
Pooled/Shared Resources



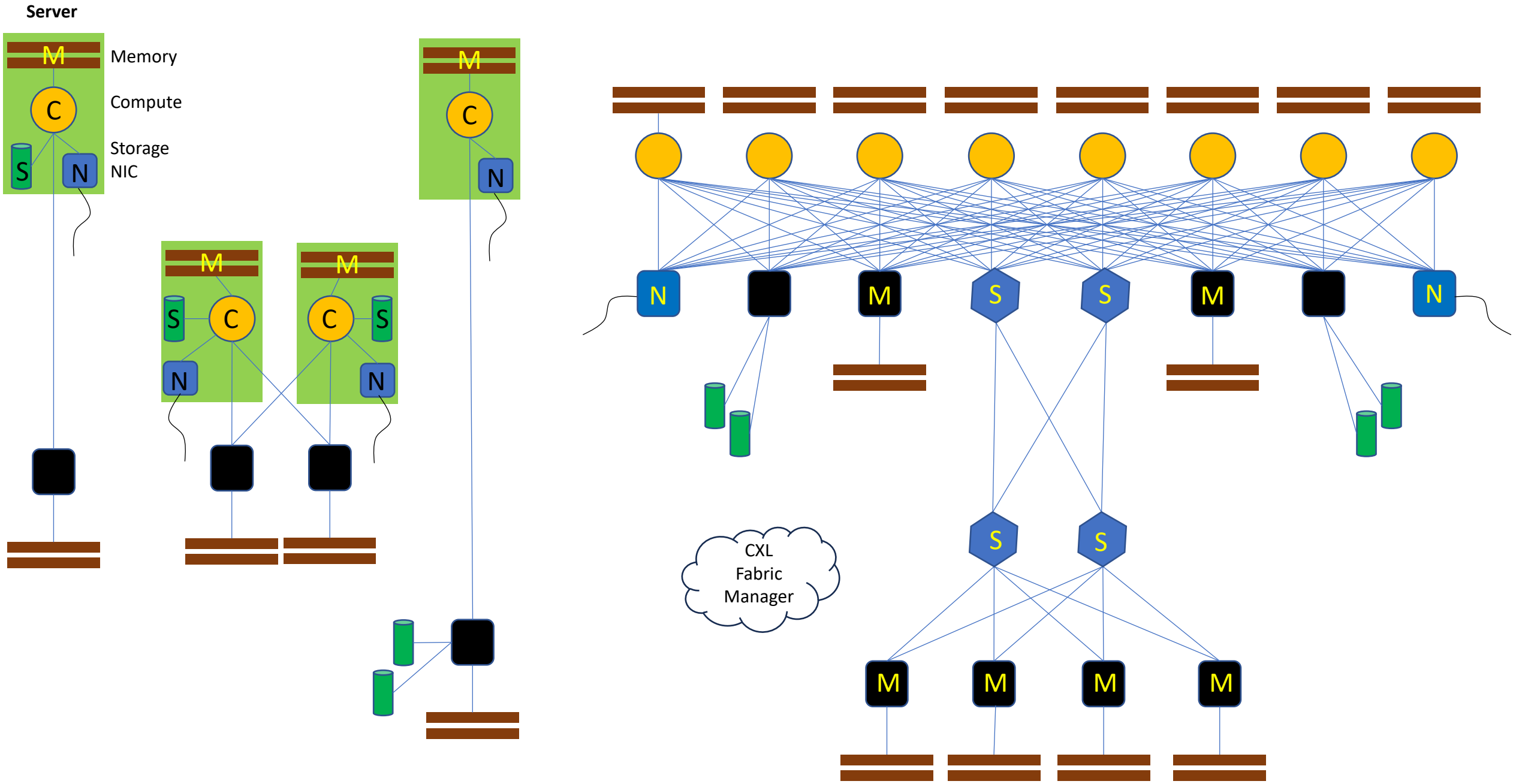


Silicon components are now available from several CXL technology suppliers to evaluate these capabilities via PoCs





We can do even more!


Server
M
Memory
C
Compute
S
N
Storage
NIC
CXL
Fabric
Manager

Challenges

Complex

Interconnect?

Disaggregation Challenges

Switches provide value while adding system considerations

- Space, Power, Latency, Cost, Complexity, Firmware

Copper Interconnect provides connectivity with added complexity

- Space, bend radius, EMI/EMC, Cost, Signal Integrity, BER

High Availability

- Fault Zones, Power Zones, Cooling Zones
- New fault modes requiring redundancy

Authenticity, Reliability, Integrity, Security, Privacy, and Management

Successful Disaggregation Approach

First do no harm

- The OS running on a Server: The Platform and the CXL Fabric Manager provide the same experience as a static server system
- Remedy every new fault mode
- Ride on PCIe, UEFI, and traditional RAS and Security
- Reduce the problem to that which has been solved before!

Put things where they belong

- Partition the system efficiently: ease of use, serviceability, maintenance

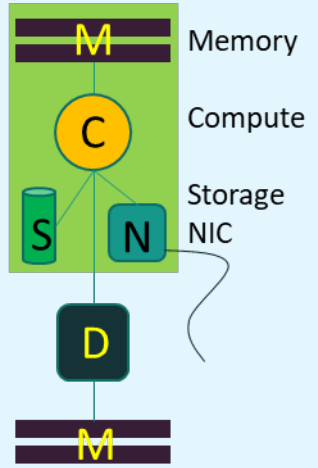
While pushing the envelop, if it hurts, don’t do it!

- Retreat from the extremes and avoid too many variables for the first generation
- Fail Fast, learn, and grow the solution through PoCs

Successful Disaggregation Approach

First do no harm

- The OS running on a Server: The Platform and the CXL Fabric Manager provide the same experience as a static server system
- Remedy every new fault mode
- Ride on PCIe, UEFI, and traditional RAS and Security
- Reduce the problem to that which has been solved before!



Put things where they belong

- Partition the system efficiently: ease of use, serviceability, maintenance

While pushing the envelop, if it hurts, don't do it!

- Retreat from the extremes and avoid too many variables for the first generation
- Fail Fast, learn, and grow the solution through PoCs

Successful Disaggregation Approach

First do no harm

- The OS running on a Server: The Platform and the CXL Fabric Manager provide the same experience as a static server system
- Remedy every new fault mode
- Ride on PCIe, UEFI, and traditional RAS and Security
- Reduce the problem to that which has been solved before!

Put things where they belong

- Partition the system efficiently: ease of use, serviceability, maintenance

While pushing the envelop, if it hurts, don’t do it!

- Retreat from the extremes and avoid too many variables for the first generation
- Fail Fast, learn, and grow the solution through PoCs

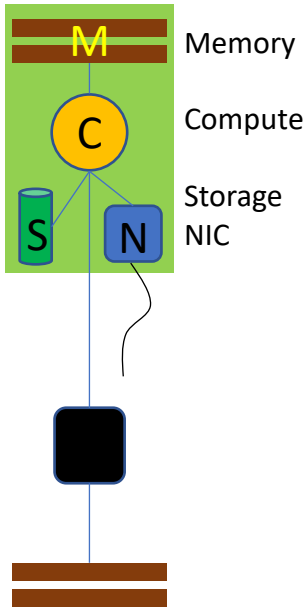
Based on the Modular Building Block Architecture (**MBA**),

Build a reference hardware system

to allow the software architecture

to be ready for

the full set of features!



Memory

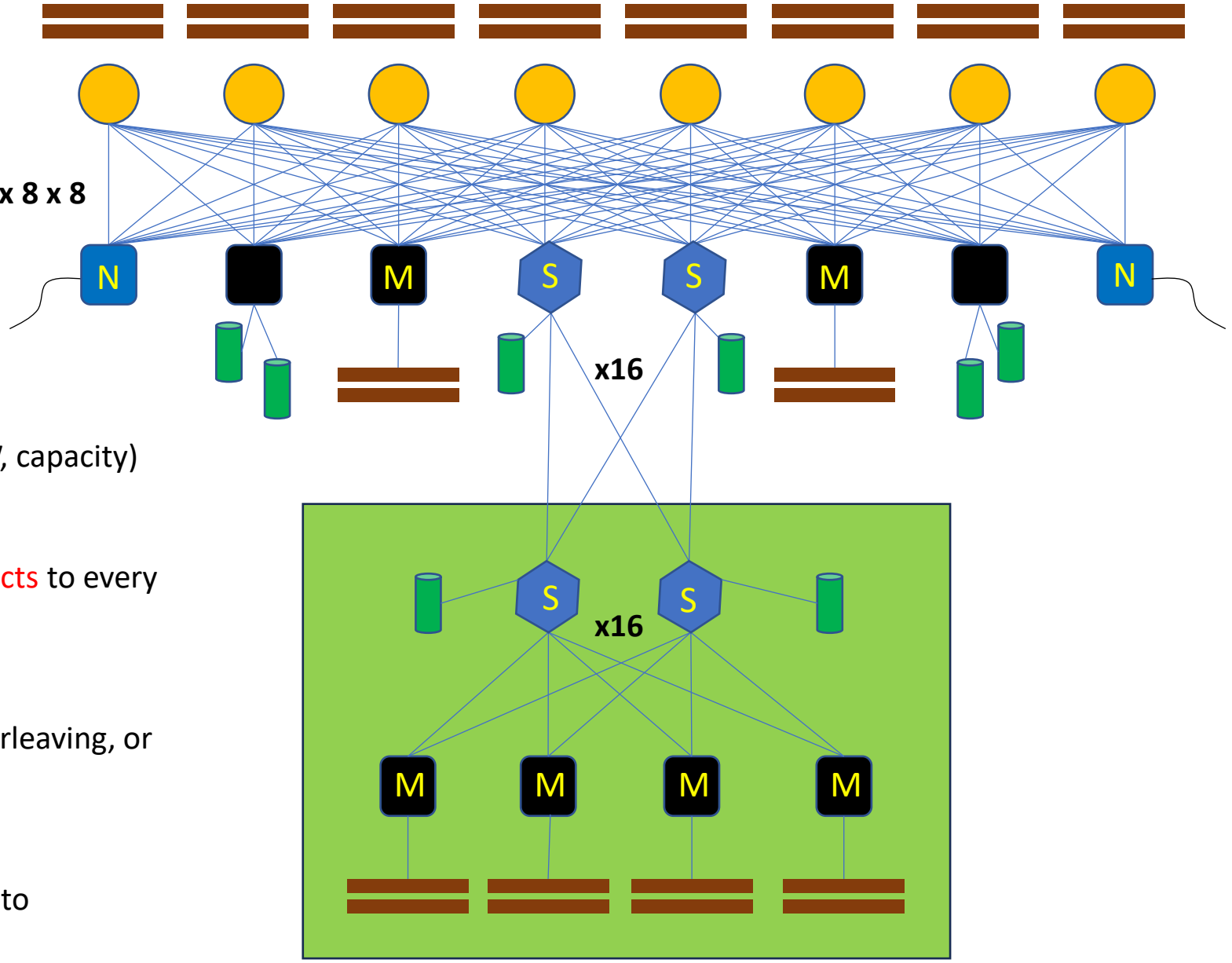
Compute

Storage
NIC

CXL
Fabric
Manager

Manages every Switch
and multi-ported device

Requires Connectivity!



Well-balanced (core-to-memory/IO ratio, BW, capacity)

Not all-to-all, but close!

Every CPU or GPU in the top row connects to every device in the bottom row

Redundant Paths for essential elements

Multiple Paths for High-availability, Interleaving, or Capacity/BW expansion

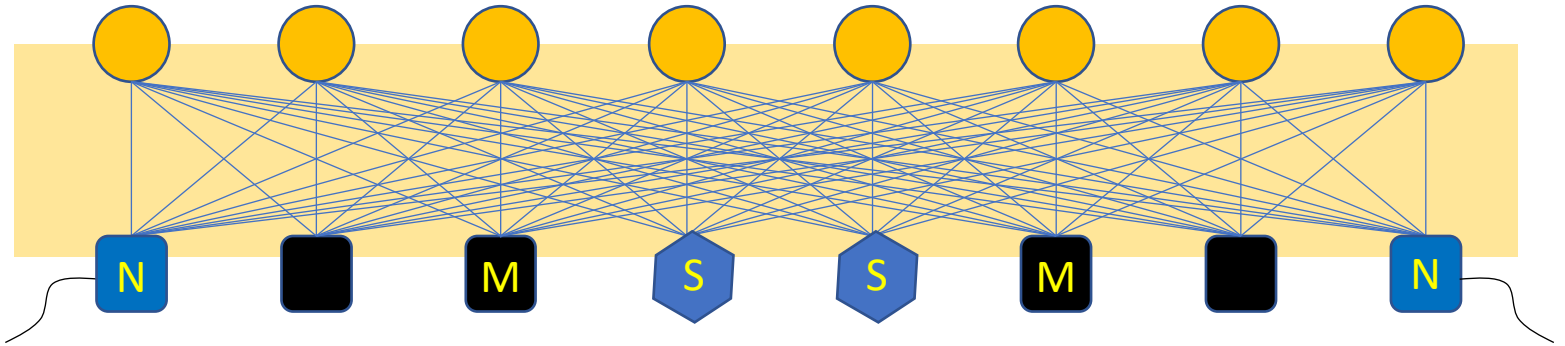
Local Disaggregation within a Chassis

With the option to Extend connectivity to Expansion Chassis

Interconnect
Mechanical Robustness

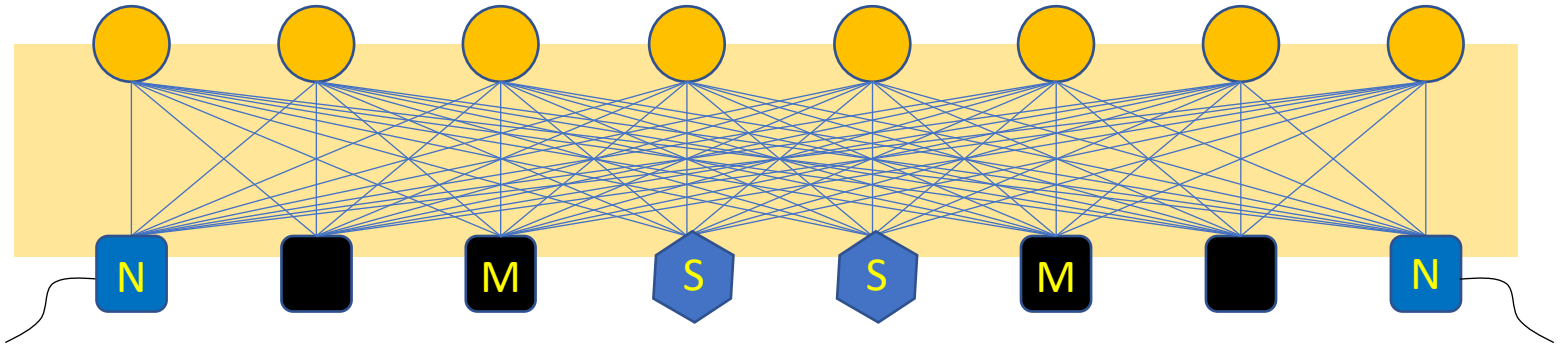
Copper Backbone
(in one assembly!)

Connectors
Cables



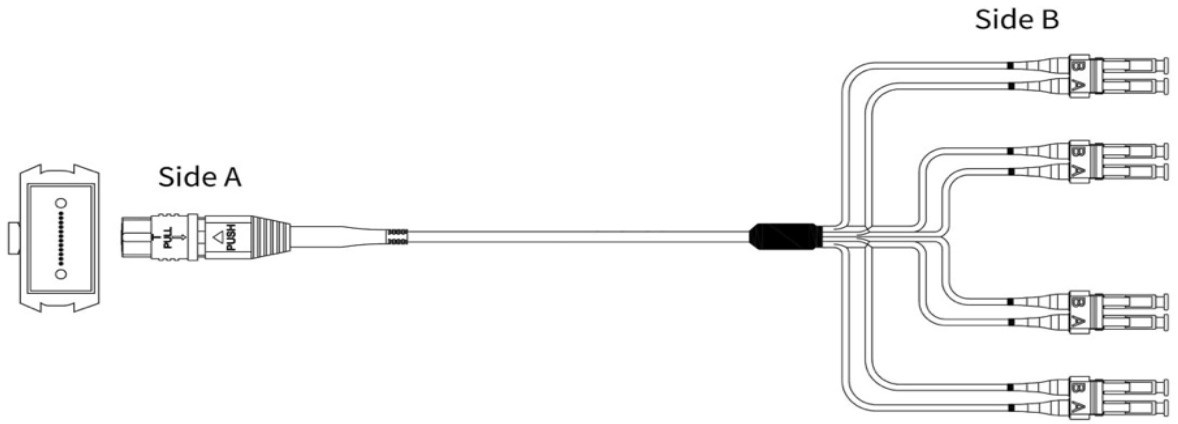
Interconnect
Mechanical Robustness

Copper Backbone
(in one assembly!)



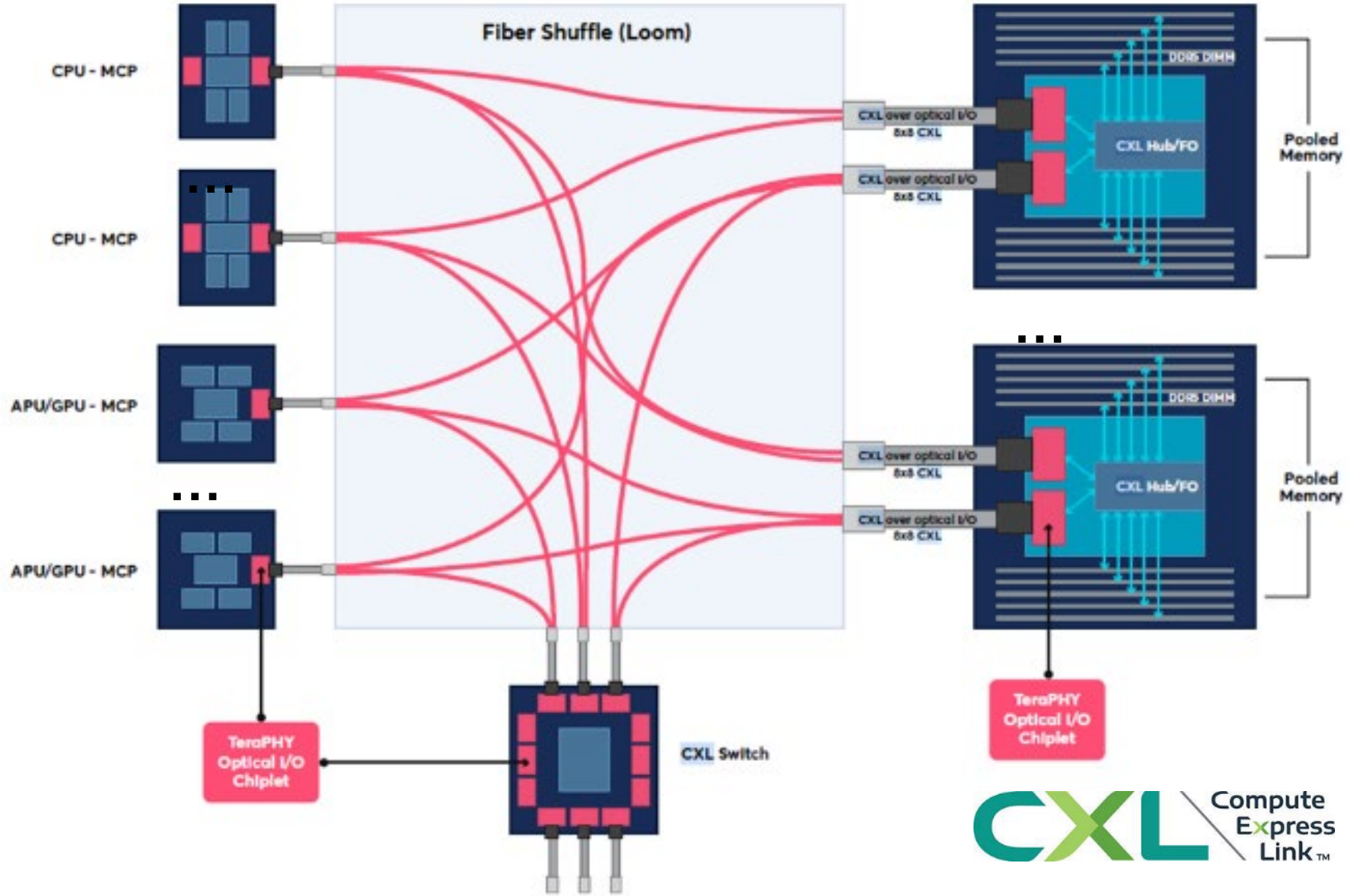
Connectors
Cables

Optical Example



Source: https://edgeoptic.com/wp-content/uploads/2021/02/12A-MPO-12-4LC.jpg

Optical I/O Enables Composability & Disaggregation

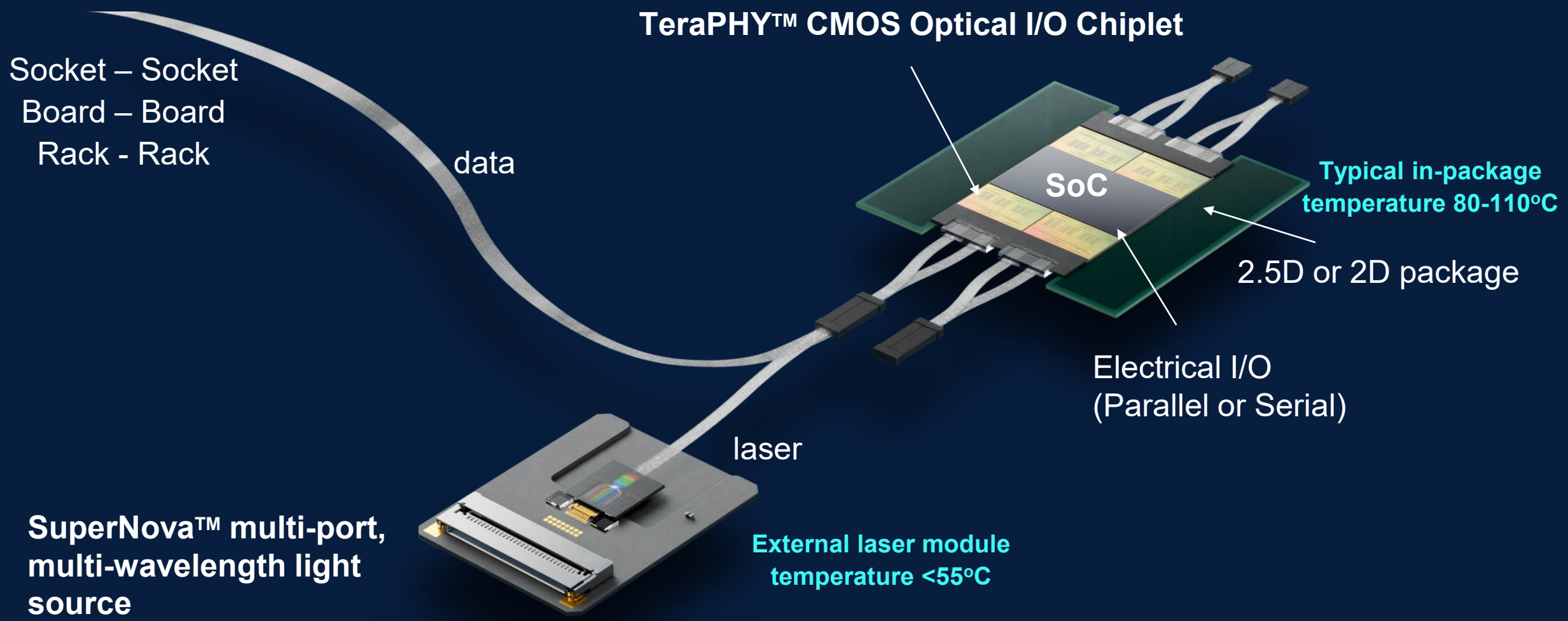


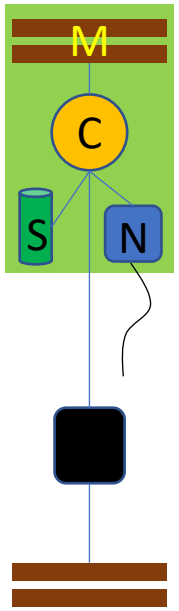
UCIe™ Universal Chiplet Interconnect Express

CXL Compute Express Link™

AyarLabs

Ayar Labs Optical I/O Solution





Memory

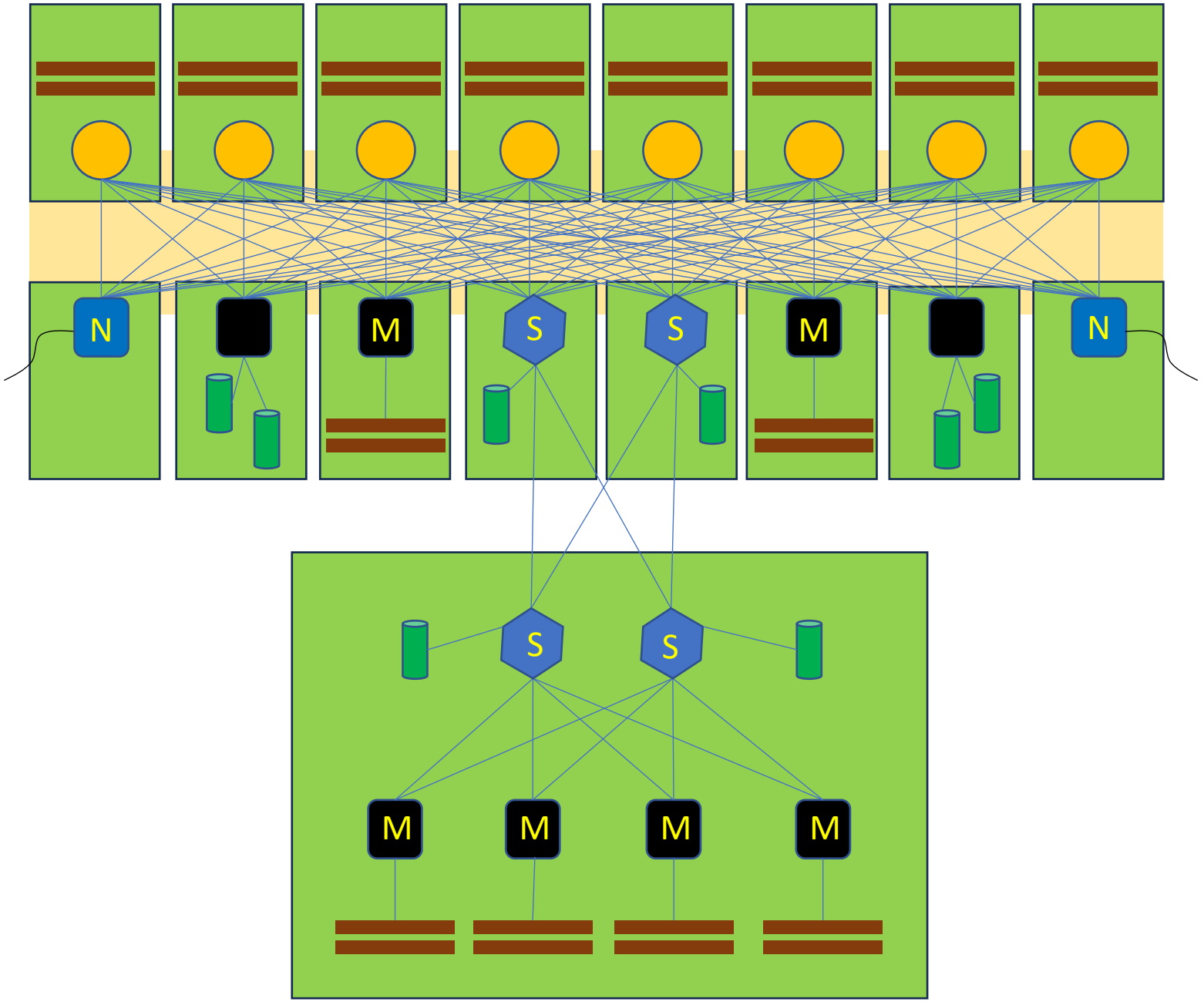
Compute

Storage
NIC

CXL
Fabric
Manager

Manages every Switch
and multi-ported device

Required Connectivity!

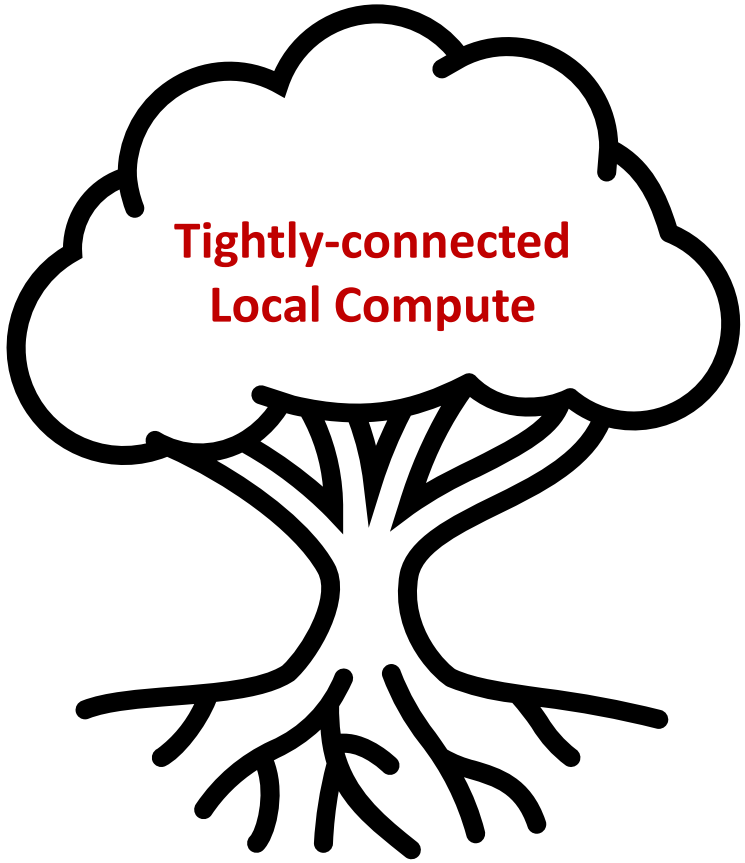


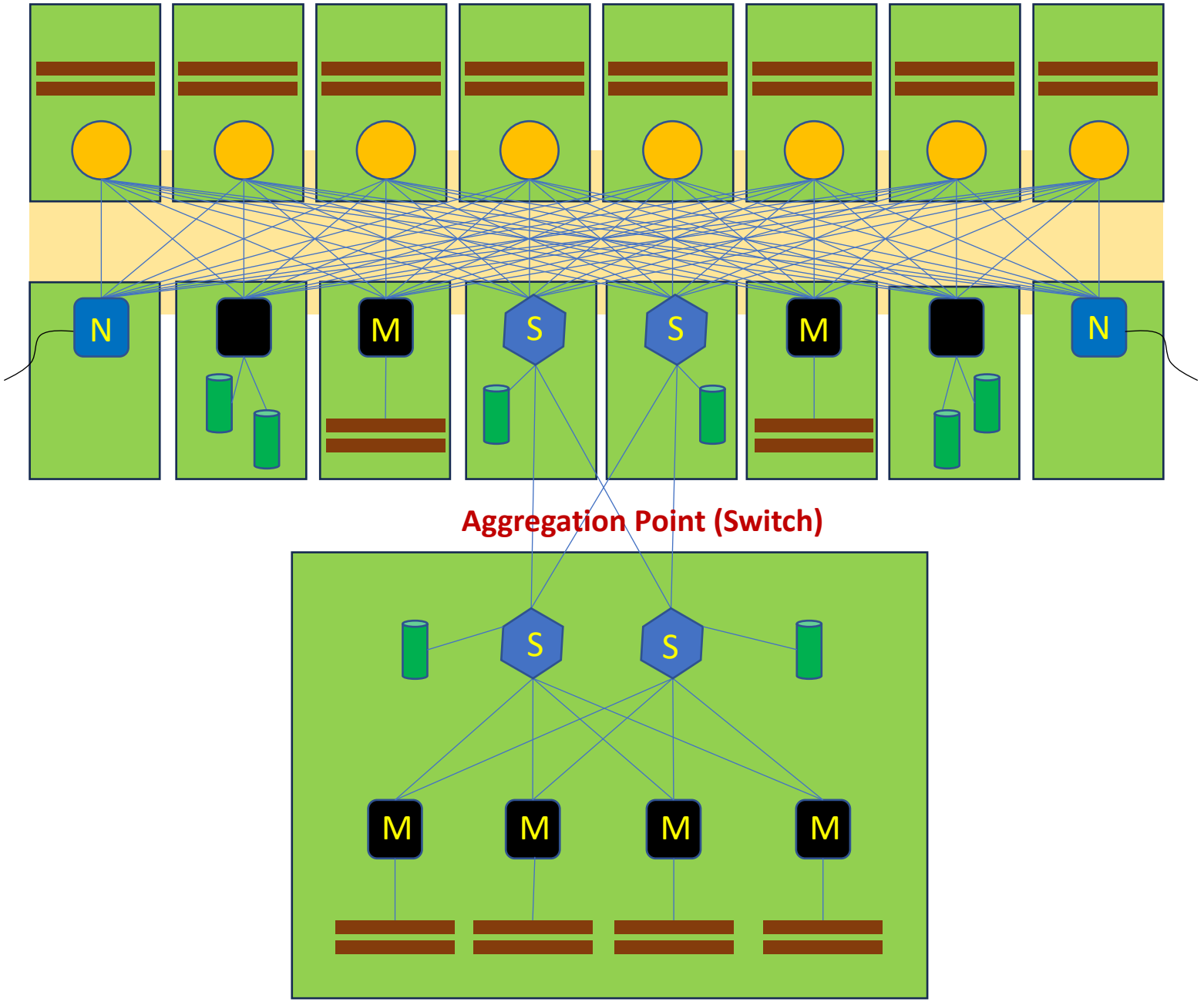
Mechanical Structure

Power

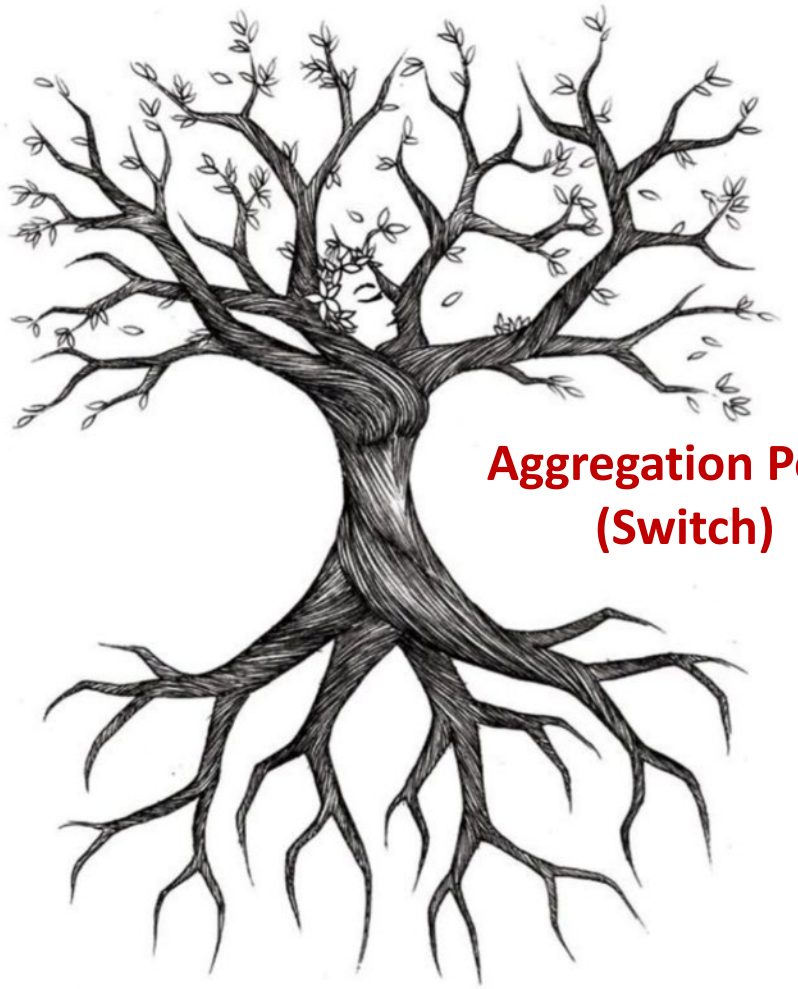
&

Cooling


Tightly-connected
Local Compute


N
M
S
S
M
N
Aggregation Point (Switch)
S
S
M
M
M
M

Tightly-connected,
Local Clusters



Aggregation Point
(Switch)

Short, local links
(High Connectivity)

Short, local links
(High Connectivity)

Fat Pipes
(High Bandwidth)

Stretched
(Disaggregated)

Celestial AI Photonic Fabric™

Optical Interconnect Technology for Memory and Compute

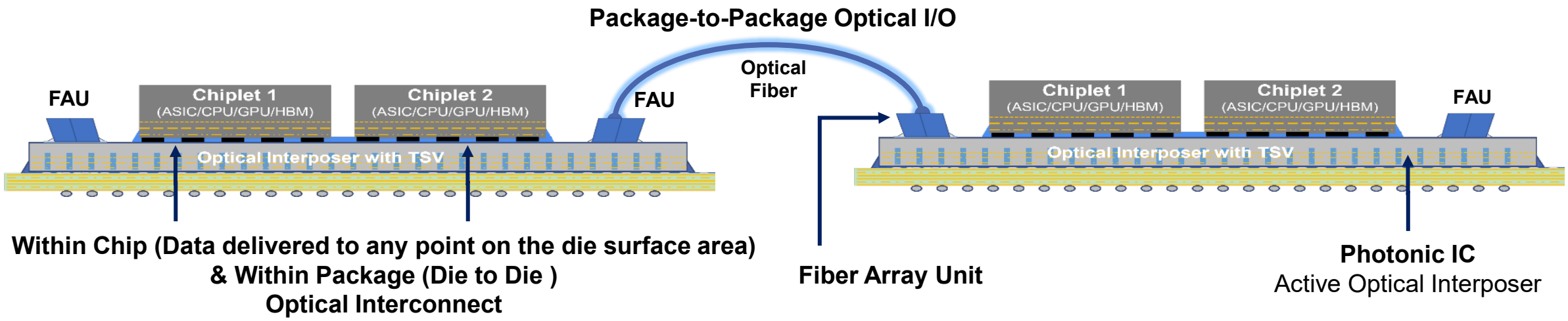
- Scalable from mm to meters
- On-chip, compute-to-compute and compute-to-memory

Deliver data directly at any point on silicon die

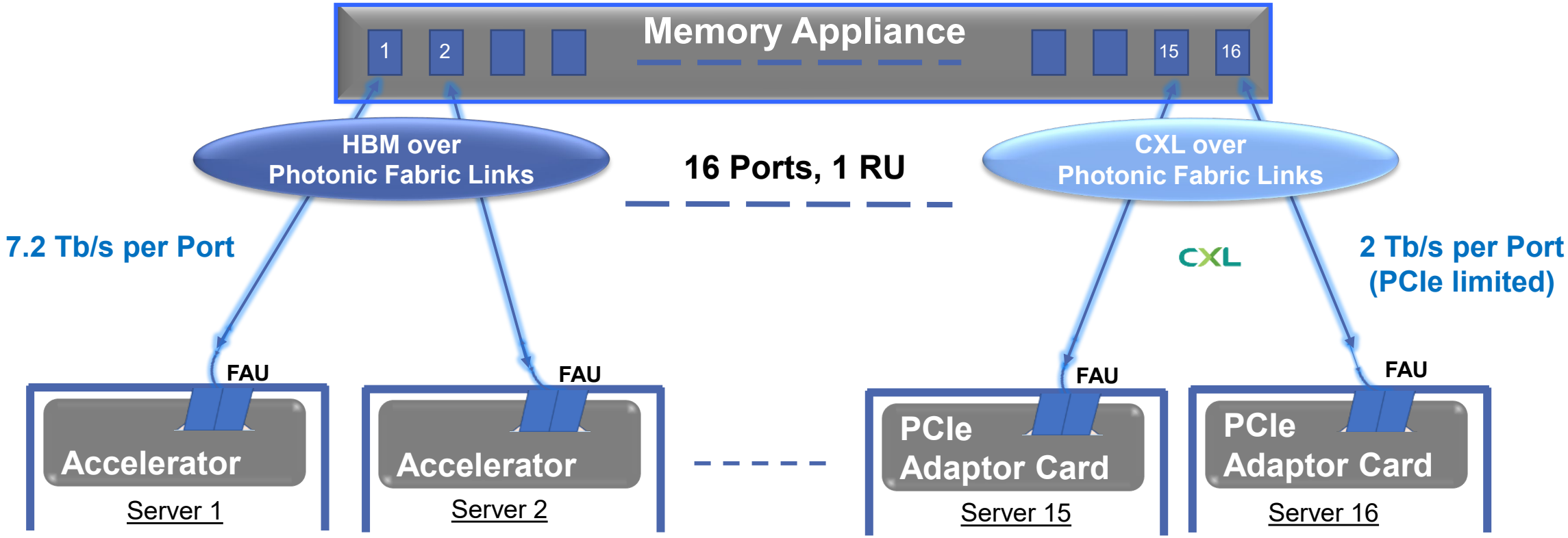
- Massive bandwidth density (1.5 Tbps/mm^2)
- Not limited by die beachfront
- Package bandwidth >700 Tbps

Supports Industry Standards

- CXL, PCIe, UCIe, JEDEC (HBM) and compatible with proprietary protocols
- Manufacturable in proven high-volume technologies



Celestial AI Pooled Memory Appliance



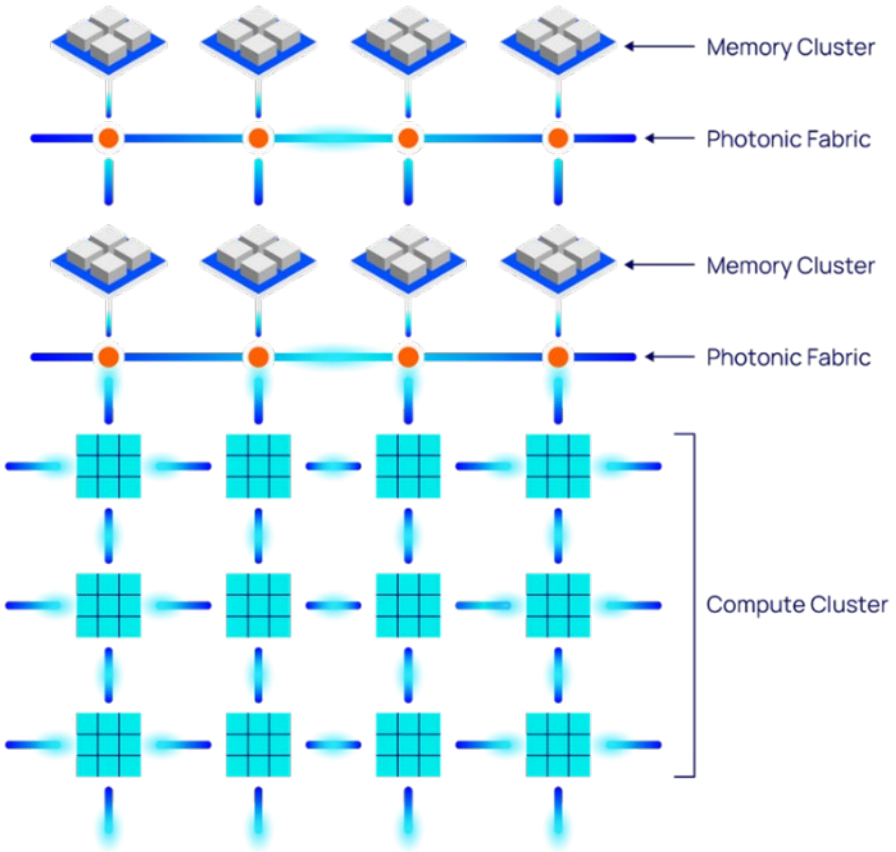
HBM Pooling over Photonic Fabric™
delivering FULL HBM bandwidth

DDR Memory Pooling with CXL3.0 over
Photonic Fabric™ with FULL PCIe bandwidth

Enables memory disaggregation with long-reach optical I/O at a fraction of PCIe latency

No CXL re-timers & switches needed

Photonic Fabric Enables Memory Disaggregation



Independently scale memory from compute

Allocate memory and compute as needed based on workload requirements

Multiple processors can share a common pool of memory

(efficient data-movement)

Enablers (Software and Firmware Ingredients)

CXL Fabric Manager

- Secure composability, allocation, on-lining/off-lining

Pre-boot Environment

- Discovery, enumeration, setup, …

CXL Bus/Class Driver

- Configuration, Resource Allocation

CXL Memory Device Driver

- Interactions with Bus/Class Driver, Fabric Manager, VMM, …
- RAS, Security, Fault-isolation, On-lining, Off-lining, …
- Error Isolation, Telemetry, Performance Monitoring

OS-specific Software

- VMM, Hypervisor
- VM Allocation, Orchestration, Fault-isolation & Recovery

A common software architecture

built on a reference hardware system

will drive hardware interoperability forward!

Activities within OCP Server Project
in support of CXL-enabled Systems

CMS (Composable Memory System)

Software Infrastructure for managing tiered, **composable**, disaggregated systems

https://www.opencompute.org/projects/composable-memory-system

DC-MHS (Datacenter-ready **Modular Hardware System**)

M-SIF (modular shared infrastructure)

Partition the system efficiently: ease of use, serviceability, maintenance

https://www.opencompute.org/projects/dc-mhs

https://www.opencompute.org/wiki/Server/DC-MHS

Extended Connectivity Workstream (for PCIe and CXL)

https://www.opencompute.org/wiki/Server/PCIe_Extended_Connectivity_Requirements_Workstream

Interconnect for the Disaggregated Computing

Local Disaggregation within a **Chassis**

With the option to **Extend** connectivity to **Expansion Chassis**

Considerations for **Copper** and **Photonic** Interconnect

OPEN Compute Project®

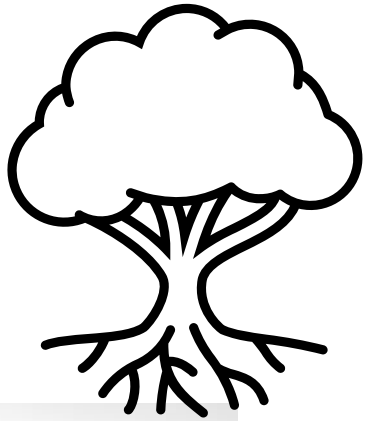
[Public]

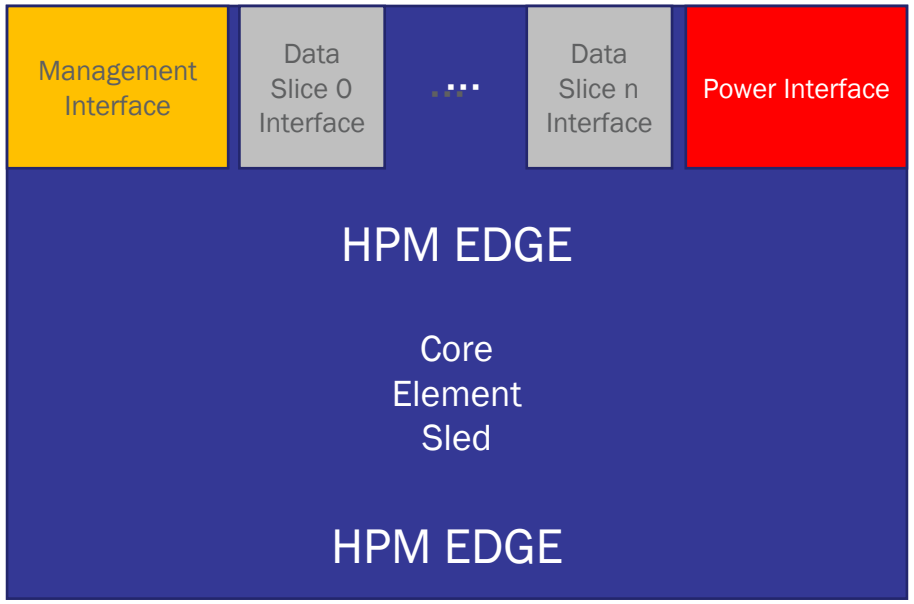
DC-MHS (Datacenter-ready Modular Hardware System)

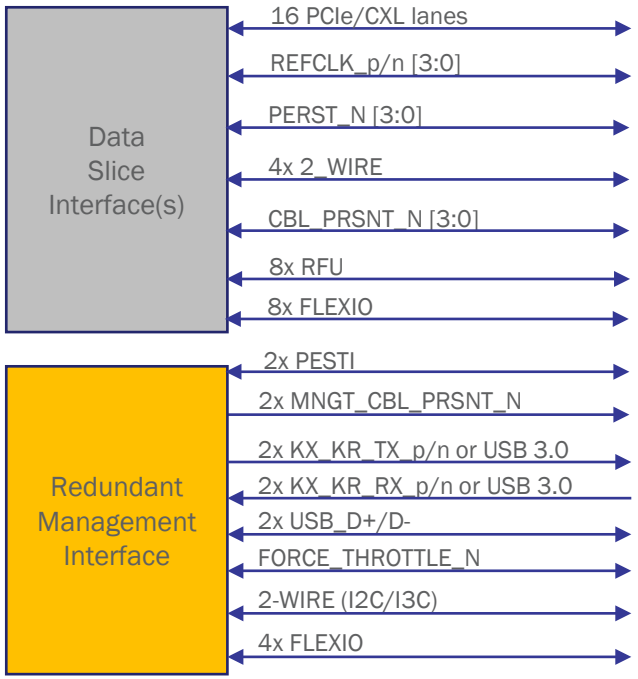
M-SIF (modular shared infrastructure)

Local Disaggregation:

Tightly-connected, Local Compute





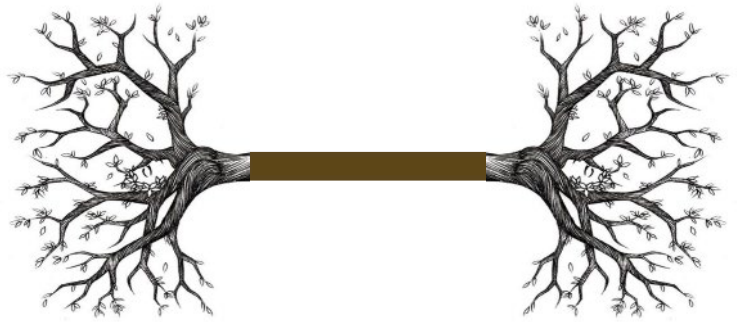


PCIe Lane Mapping			
15:12	11:8	7:4	3:0
x16			
x8		x8	
x4	x4	x4	x4
x8		x4	x4
x4	x4	x8	

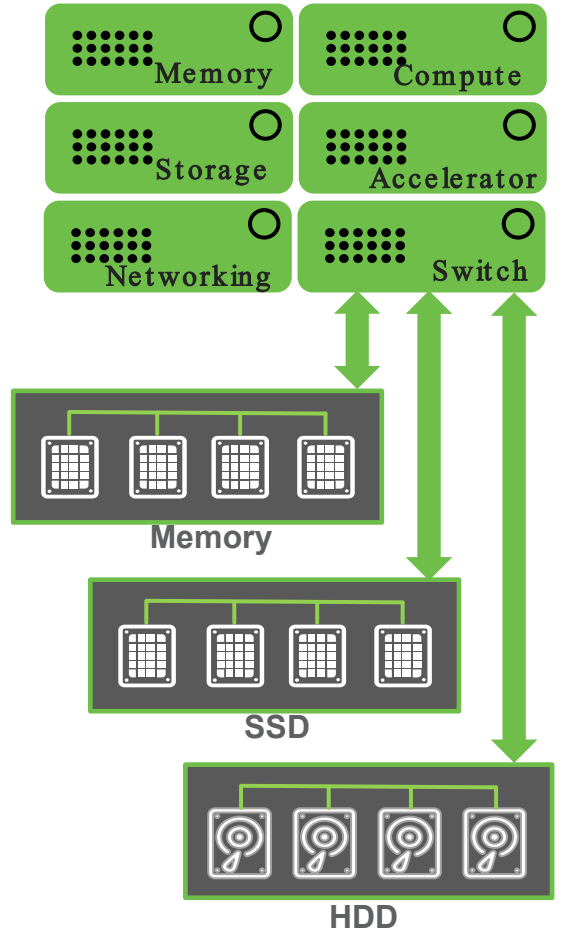
Data Slice Bifurcation & Directionality is negotiated over the management interfaces

Extended Connectivity Workstream

OCP Requirement Workstream Charter



- Research and document the future Compute, Storage, Accelerator and Memory connectivity scenarios for the NVMe & CXL enabled disaggregated datacenter
- Identify and document Commonalities and Differences between Switches, Compute, Accelerator, Networking, Storage and Memory appliance connectivity requirements
- Explore the Cost, Bandwidth, Latency, Density, Distance expectations with Electrical and Optical solutions to meet the PCIe NVMe & CXL requirements
- Produce detailed extended intra-rack & inter-rack PCIe (NVMe & CXL) connectivity High-Level scenarios and requirements



Summary: Successful Disaggregation Approach(at OCP)

First do no harm (software compatibility, security, and management: **CMS**)

- The OS running on a Server: The Platform and the CXL Fabric Manager provide the same experience as a static server system
- Remedy every new fault mode
- Ride on PCIe, UEFI, and traditional RAS and Security
- Reduce the problem to that which has been solved before!

Put things where they belong (**modular hardware** system: DC-MHS/M-SIF)

- Partition the system efficiently: ease of use, serviceability, maintenance

While pushing the envelop, if it hurts, don't do it! (robust Extendeded **Connectivity**)

- Retreat from the extremes; avoid too many variables for the first generation
- Fail Fast, learn, and grow the solution through PoCs

Call to Action!

Join **CXL Consortium**, drive new use cases, propose solutions, and help draft specifications

Join **OCP Server Project** and actively participate in the **subprojects**

Drive initiatives, make **contributions** (Base Spec, Design Spec, Products, …)
https://www.opencompute.org/wiki/Server

CMS (Composable Memory System)

https://www.opencompute.org/projects/composable-memory-system

DC-MHS (Datacenter-ready Modular Hardware System)

https://www.opencompute.org/projects/dc-mhs

Extended Connectivity Workstream (for PCIe and CXL)

https://www.opencompute.org/wiki/Server/PCIe_Extended_Connectivity_Requirements_Workstream

ODSA (Open Domain-Specific Architecture)

https://www.opencompute.org/wiki/Server/ODSA

OAI (Open Accelerator Infrastructure)

https://www.opencompute.org/wiki/Server/OAI

HPC (High-performance Computing)

https://www.opencompute.org/wiki/HPC

OCP NIC

https://www.opencompute.org/wiki/Server/NIC

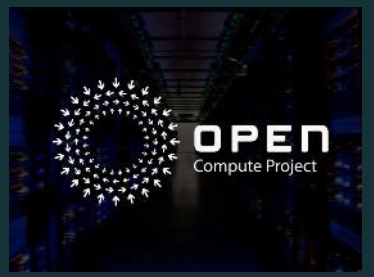
OPEN Compute Project®

CXL

Efficient Interconnect

Technology

Compute Express Link™



OCP

Hardware/Software Co-design

System Integration

Presenter BIO

Siamak Tavallaei joined the CXL effort as a founding member of the CXL Consortium and co-chair of the Technical Task Force (TTF) in 2019 to develop the CXL 2.0 specification. He is currently the CXL Advisor to the Board at CXL Consortium. He has served on the CXL Board of Directors and as the CXL President. In 2016, he joined Open Compute Project (OCP) as a co-lead of Server Project. He is currently the Incubation Committee Representative for the Server Project where he drives open-sourced modular design concepts for integrated hardware/software solutions. His recent focus has been the optimization of large-scale, mega-datacenters for general-purpose and tightly-connected accelerated machines built on co-designed hardware, software, security, and management. His experiences as Chief Systems Architect at Google Cloud, Senior Principal Architect at Microsoft Azure's Hardware Architecture team, a Distinguished Technologist at HP, a Principal Member Technical Staff at Compaq, and his contributions to industry collaborations such as CXL, OCP, EISA, PCI, InfiniBand, and PCIe give Siamak a broad understanding of requirements for the Enterprise, Hyperscale, and Edge datacenters for industry-wide initiatives.