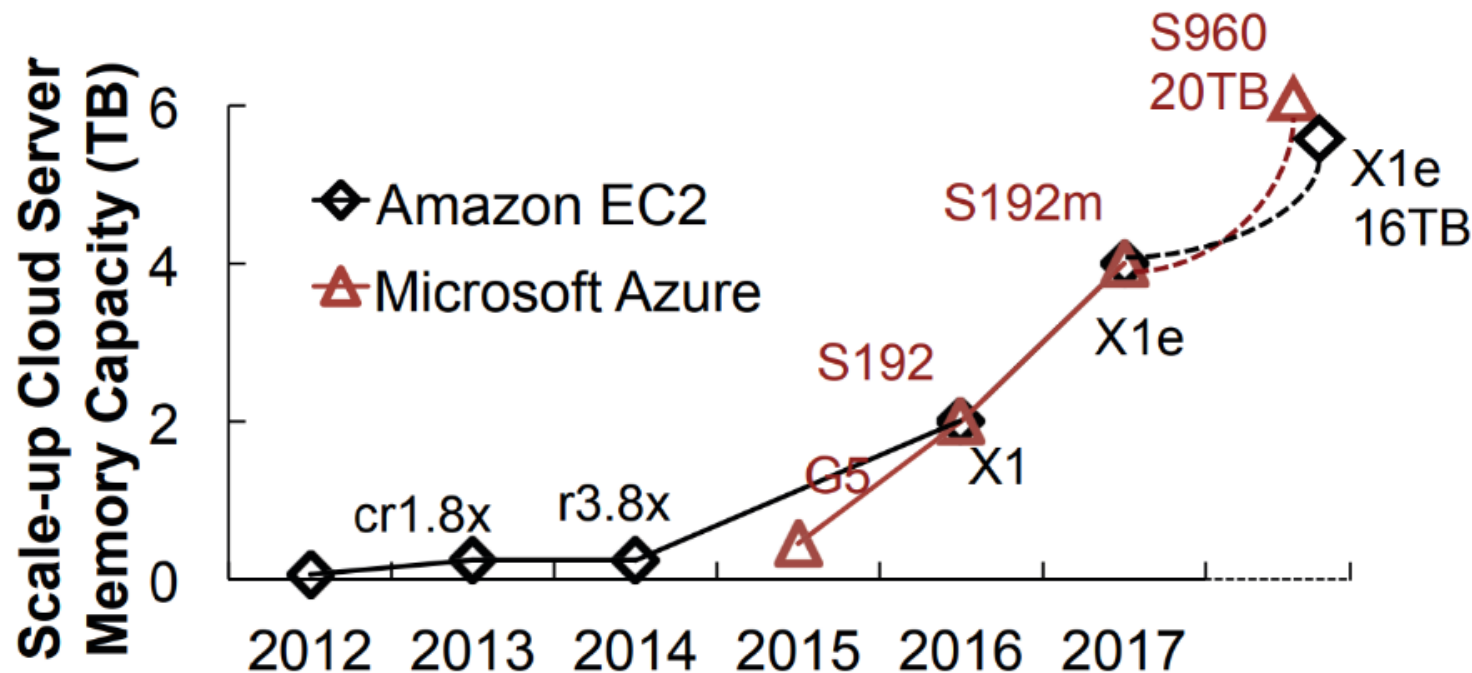


CXL Memory Expansion

Memory Tiering and Pooling
Tutorial for Lecture 02

Advanced AI Computing and Systems

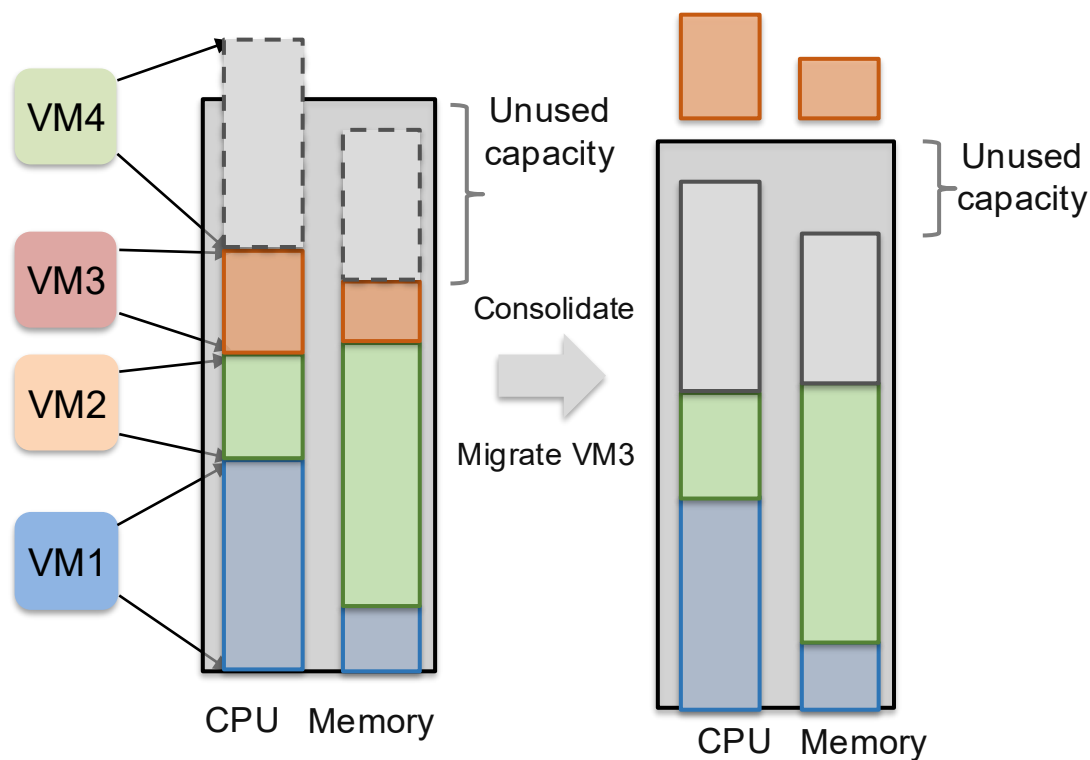
TB-Scale Memory Demand



Historical cloud scale-up configurations. Dates and labels belong to the original figure.

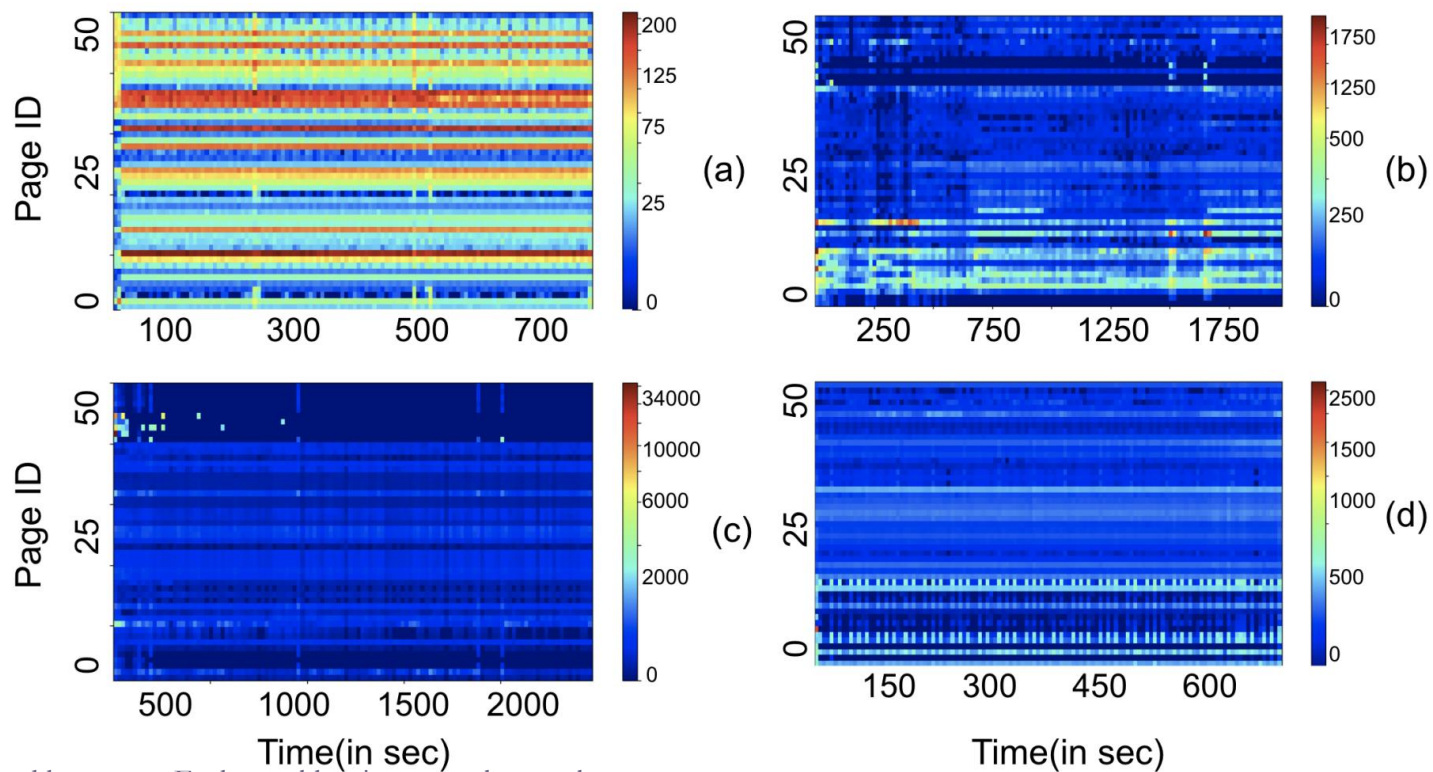
- Databases, large indexes and AI state can require TB-scale capacity.
- A larger resident working set can avoid expensive data movement.

Stranded Memory and Fixed Ratios



- Fixed CPU-to-memory ratios can strand capacity beside busy resources.
- VM consolidation helps, but some capacity can remain unavailable to other hosts.

Why Heterogeneous Memory?



Original workload heatmaps. Each panel has its own colour scale.

- Page activity is uneven and changes over time.
- DRAM can serve active data while capacity memory or SSD holds colder data.

CXL as Memory Infrastructure

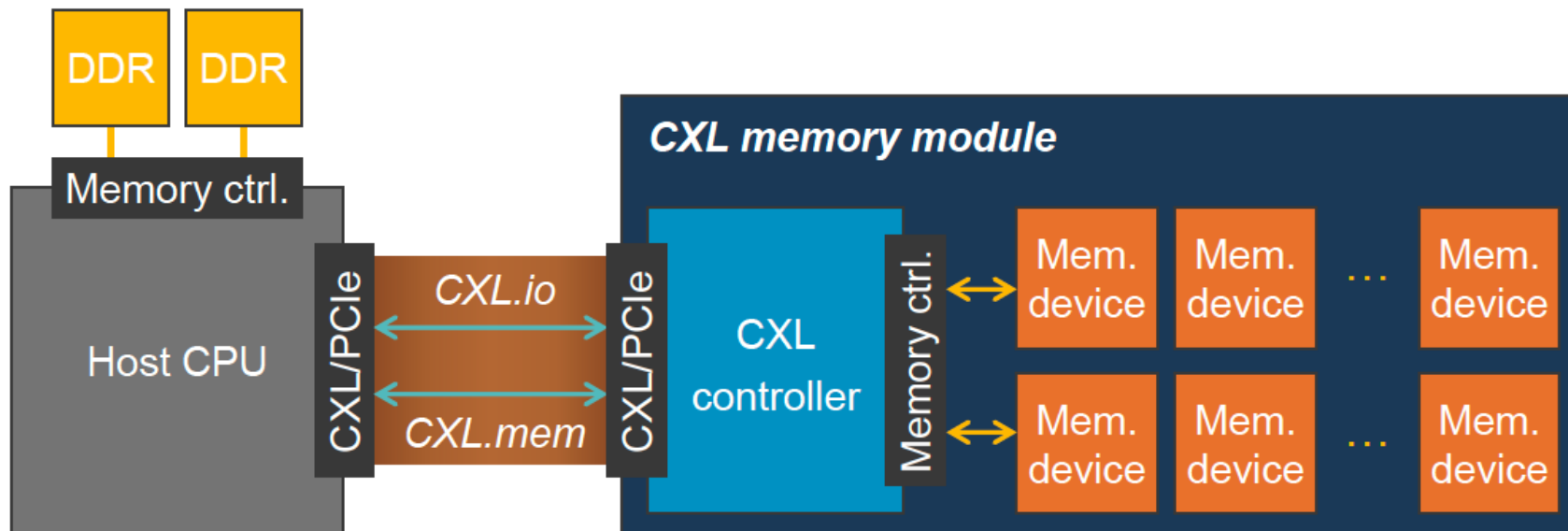
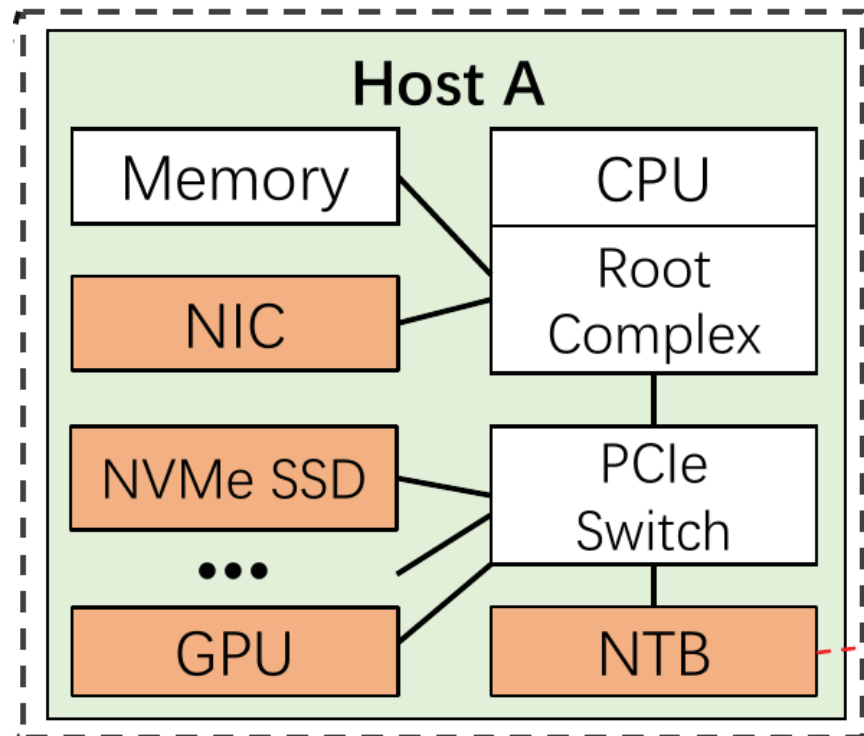


Figure 1: CXL memory module architecture.

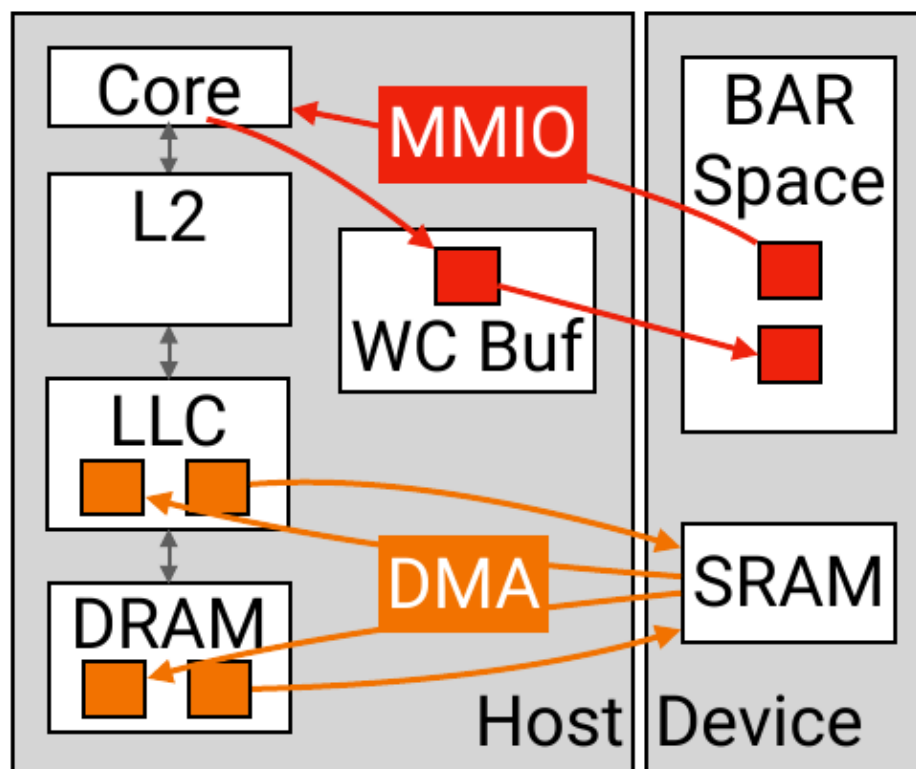
- CXL adds a host-accessible memory target beyond conventional DIMMs.
- Software decides how applications receive and use the added capacity.

PCIe in a Server



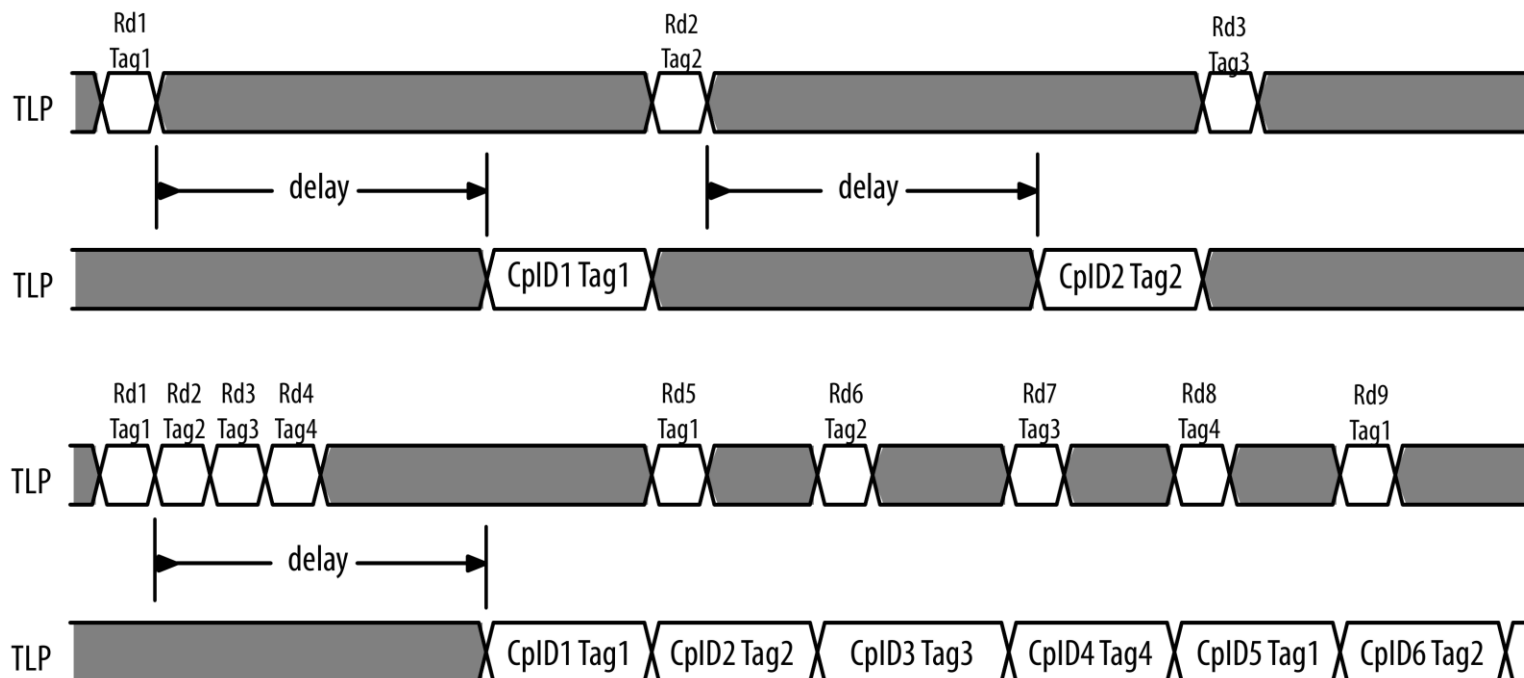
- The Root Complex connects the CPU and memory system to PCIe devices.
- Switches extend the tree to endpoints such as NICs, GPUs and storage.

MMIO and DMA



- MMIO: the CPU accesses a device's mapped register or memory window.
- DMA: the device accesses buffers in host memory.

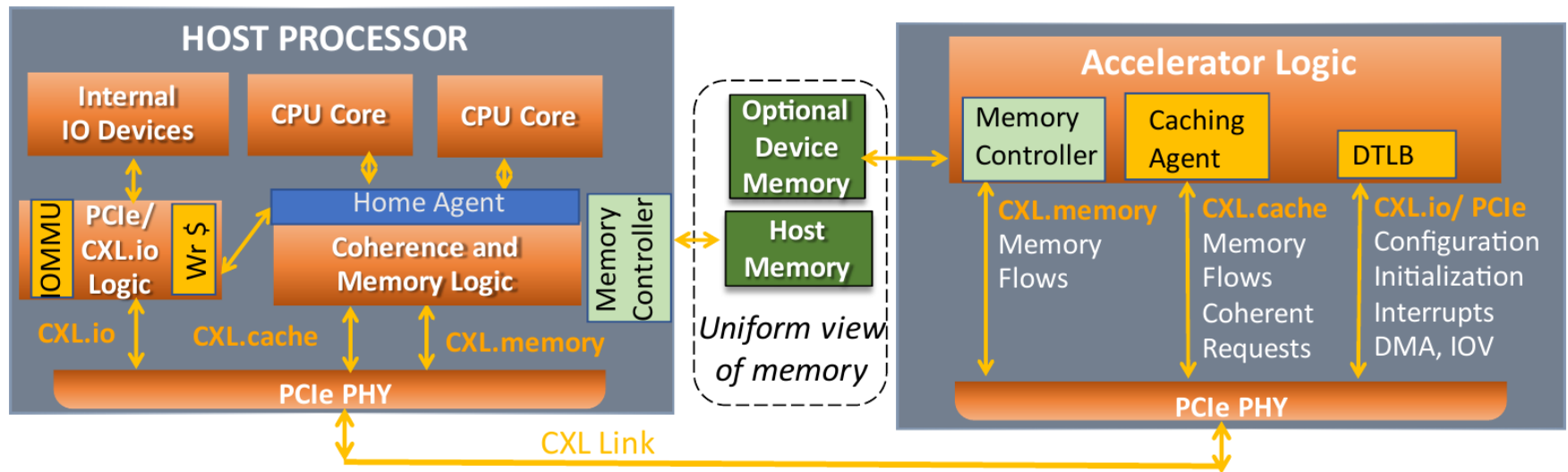
Reads, Writes, and Completions



Posted memory writes do not receive a successful transaction-layer Completion TLP.

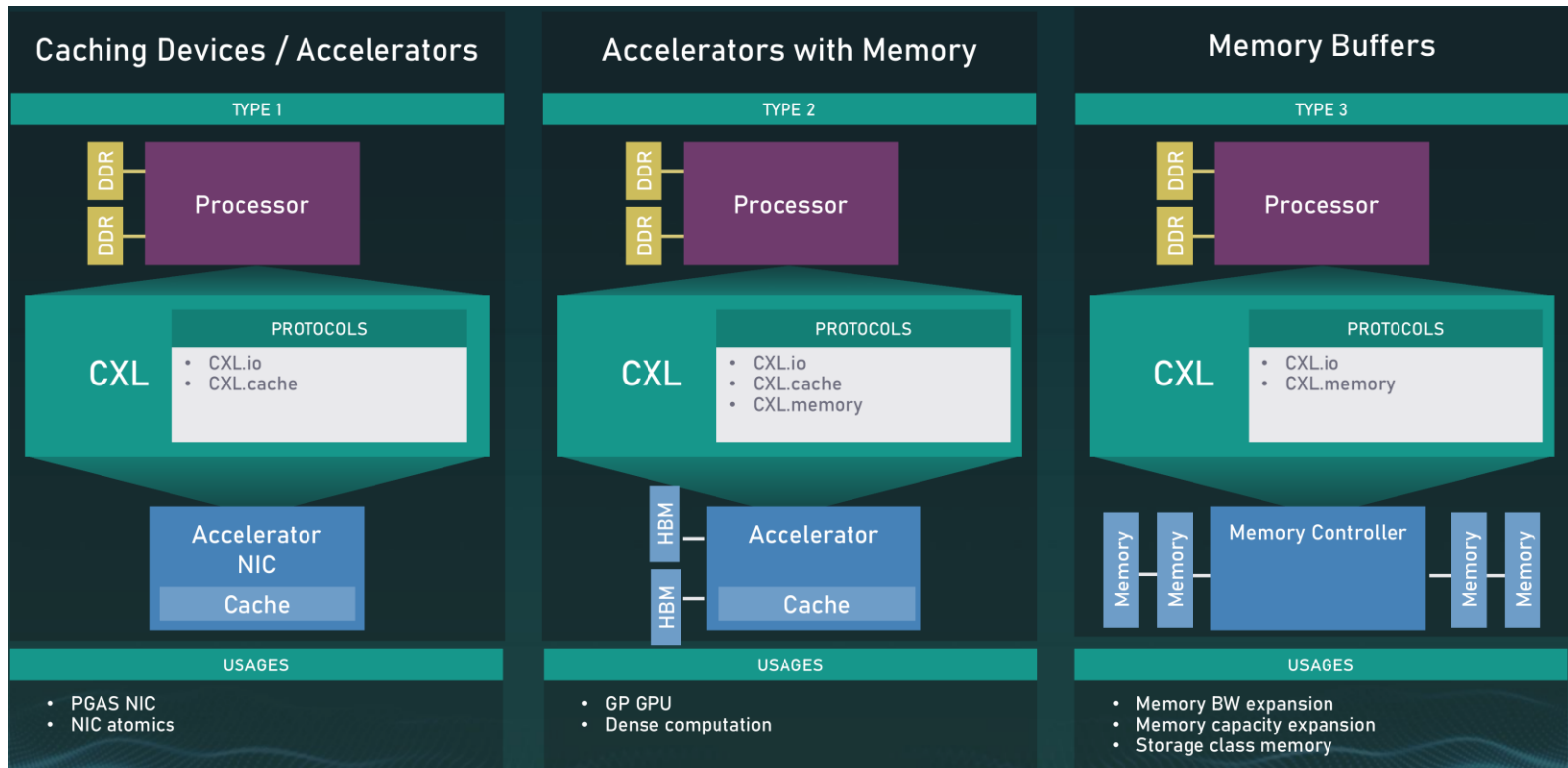
- A read sends MRd and receives data in one or more CplD packets.
- Multiple outstanding requests can cover the round-trip delay.

CXL on PCIe Infrastructure



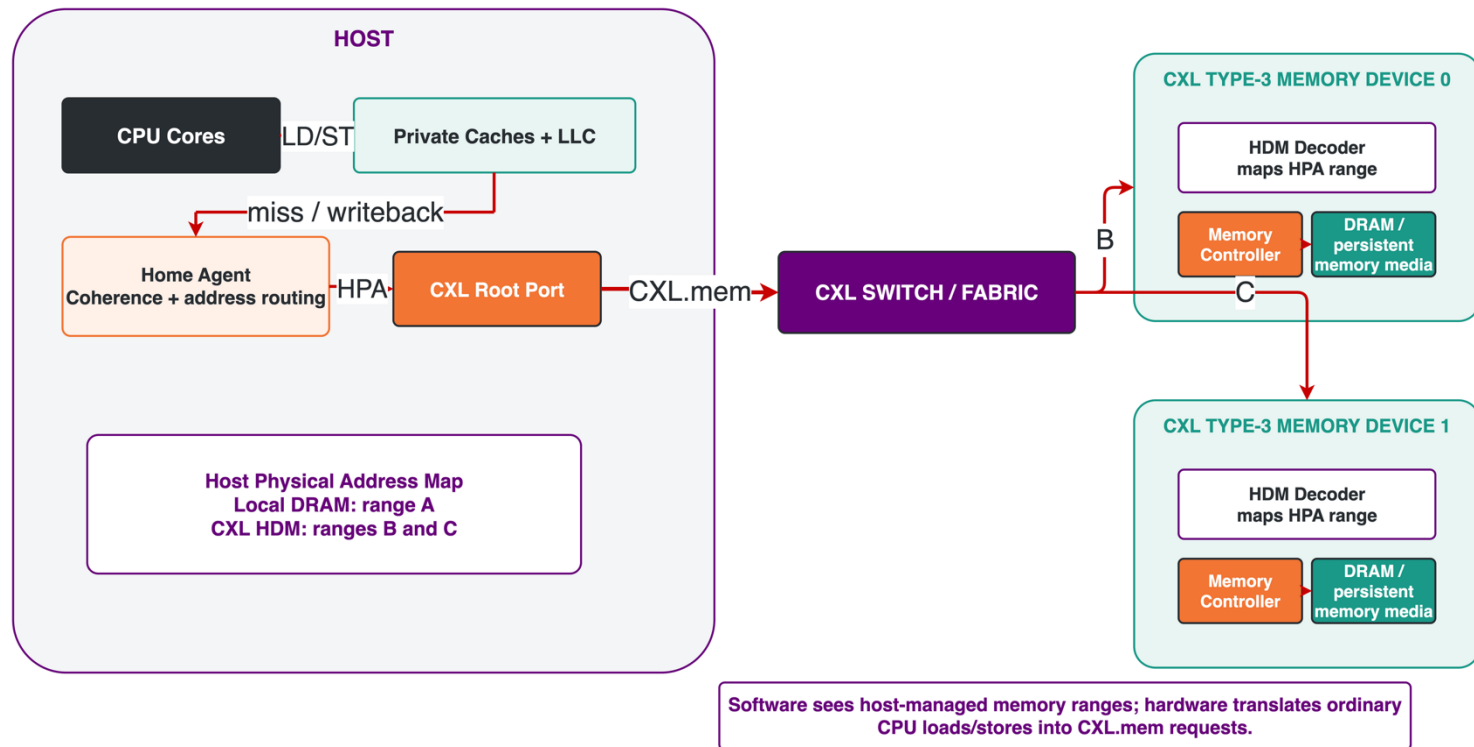
- CXL reuses the PCIe physical infrastructure and device-discovery foundation.
- CXL.io, CXL.cache and CXL.mem serve different protocol roles.

CXL Device Types



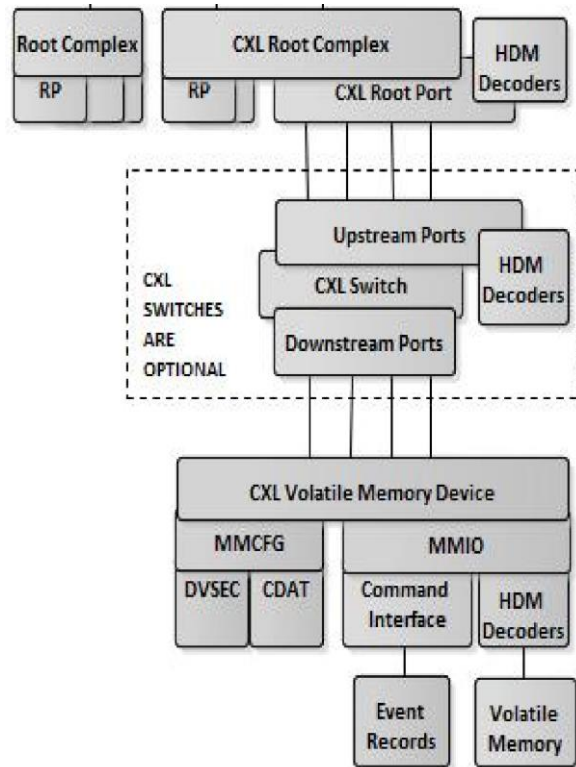
- Type 1 and Type 2 devices can cache host memory.
- Type 3 supplies host-accessible memory through CXL.mem.

Following a CPU Load



- A miss targeting a mapped CXL range follows the host's memory path.
- The endpoint serves the request and the returned line can enter CPU caches.

HDM Decoding: HPA to DPA



Noninterleaved example

Host base: 0x40000000

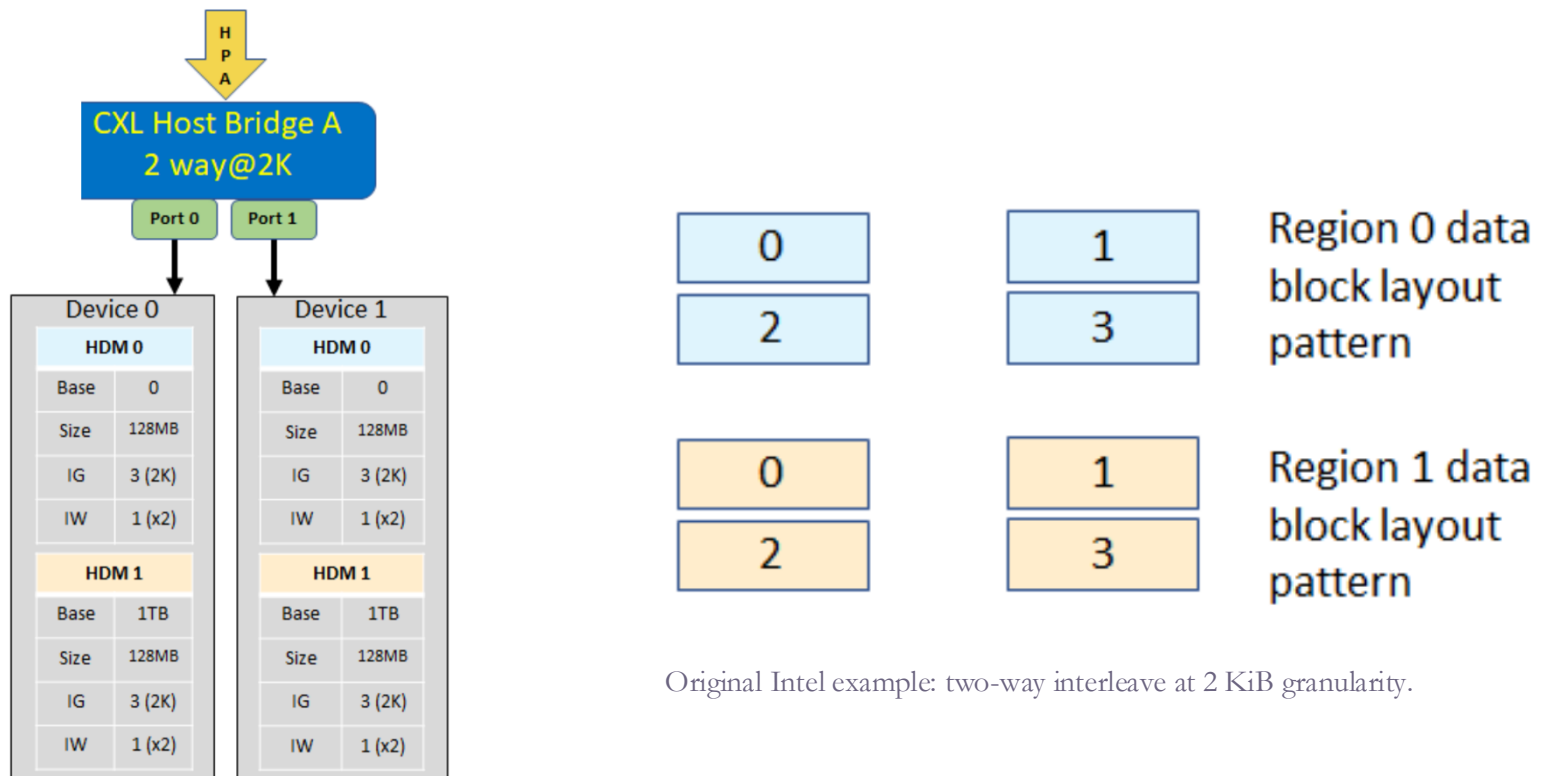
HPA: 0x40001240

Device base: 0

$$\begin{aligned} \text{DPA} &= \text{HPA} - \text{host base} \\ &= 0x1240 \end{aligned}$$

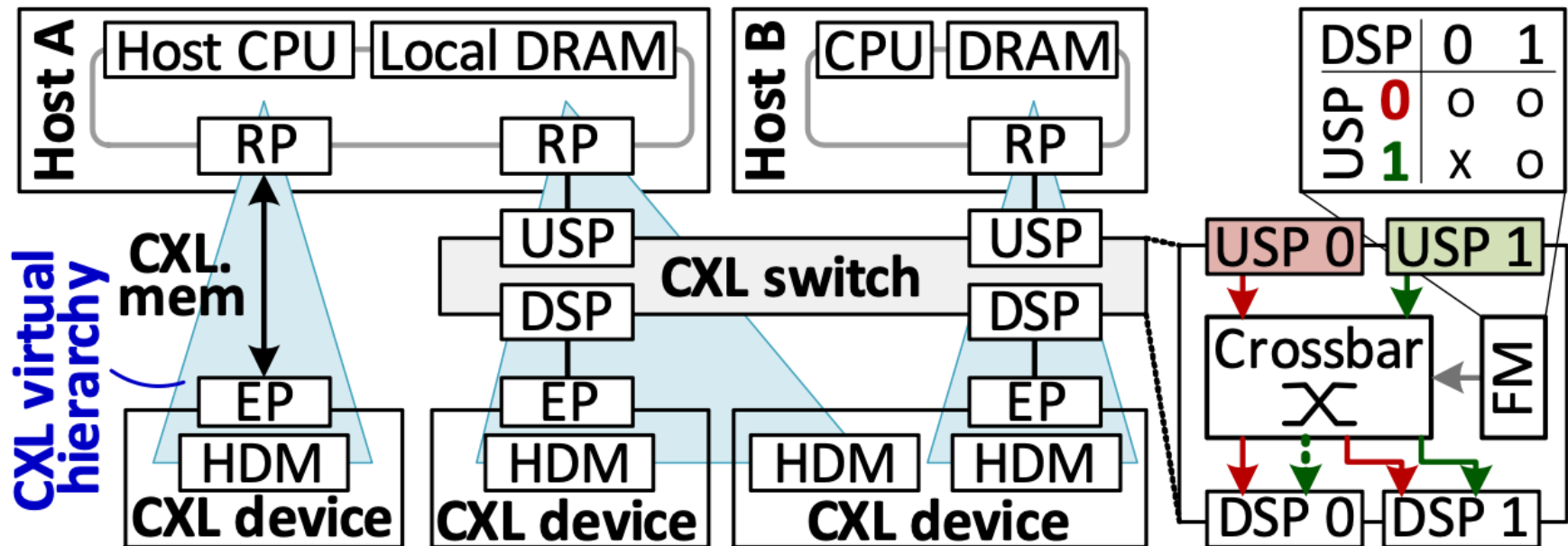
- Root and switch decoders select paths while preserving HPA.
- The endpoint translates the address into its device-local range.

Interleaving Across Devices



- Ways chooses the number of participating targets. Granularity sets the stripe size.
- Parallel requests can use multiple paths, while one small load targets one stripe.

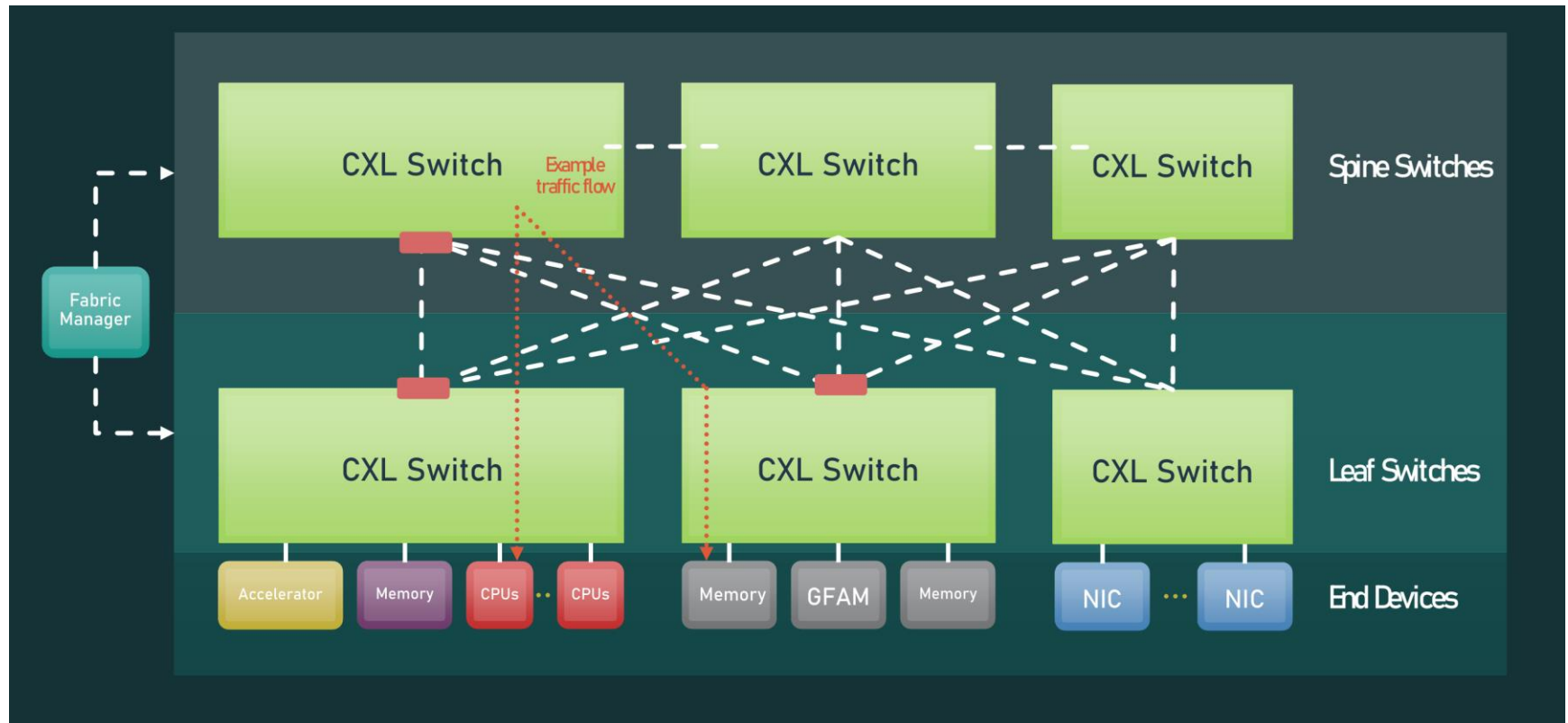
Pooling and Fabric Management



RP: root port. USP/DSP: upstream/downstream ports. EP: endpoint. FM: Fabric Manager.

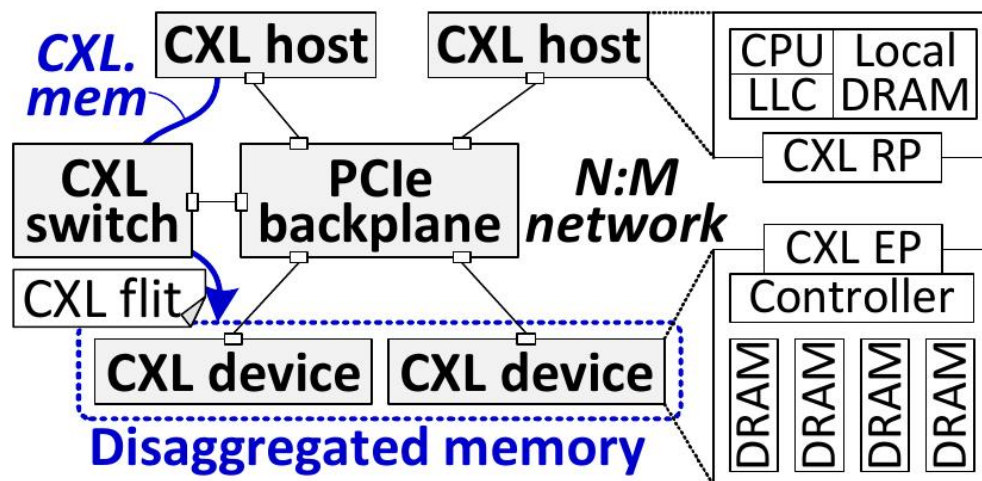
- The Fabric Manager binds resources into a host virtual hierarchy.
- Ordinary memory requests follow the configured data path.

Multilevel CXL Fabrics

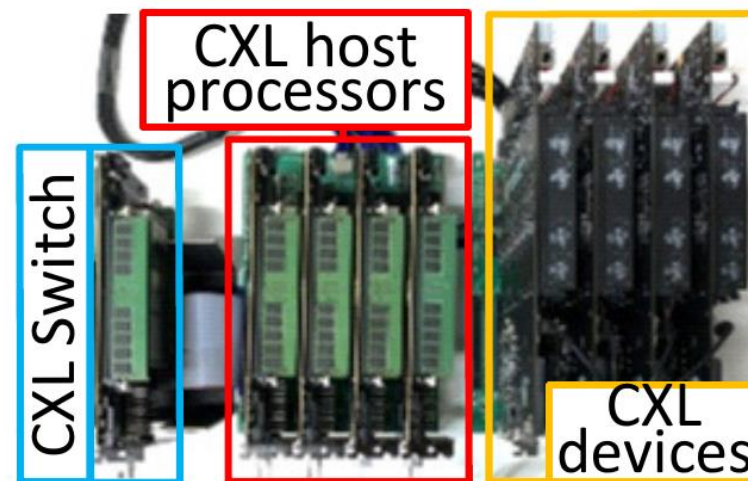


- CXL 3.0 adds multilevel switching and fabric-attached resources.
- Placement selects hop count and the shared links that carry traffic.

DirectCXL: Hardware Organization



(a) Network topology.

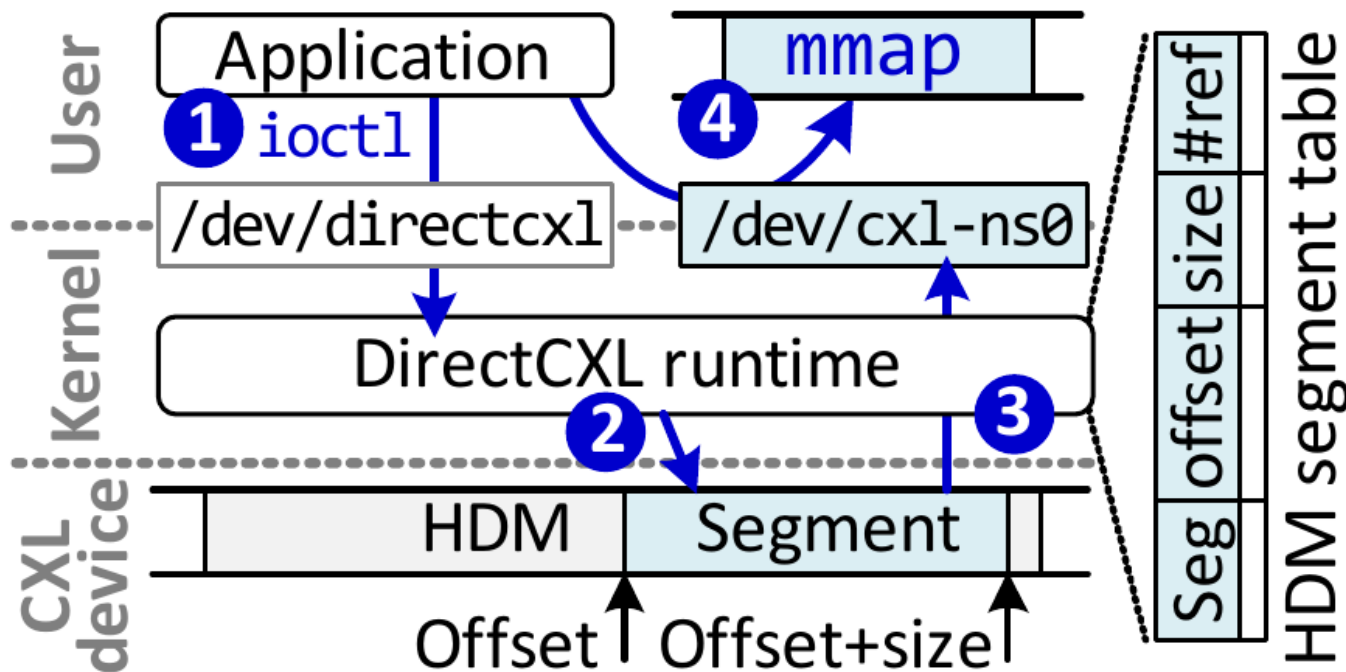


(b) Implementation.

Four FPGA hosts and four memory devices. Host CPU: 100 MHz. CXL/PCIe IP: 250 MHz.

- The prototype integrates hosts, memory blades and a custom CXL switch.
- Its system topology and the 1:1 latency experiment are different configurations.

DirectCXL: From Allocation to Load/Store



- The driver allocates an HDM segment and exposes a namespace.
- After `mmap`, ordinary memory instructions access the mapped range.

Link Budget and Useful Bandwidth

Decimal GB/s. Interface budgets after encoding and before transaction or service overhead.

Interface budget	Calculation	Result
PCIe 5.0 ×16	$32 \text{ GT/s} \times 16 \times 128/130 \div 8$	$\approx 63.0 \text{ GB/s per direction}$

DDR5-4800, 64 data bits	$4.8 \text{ GT/s} \times 64 \div 8$	38.4 GB/s shared by reads/writes
--------------------------------	-------------------------------------	----------------------------------

Transport budget, memory payload and application bandwidth are different quantities.

- CXL memory payload throughput is lower than the transport budget.
- Compare one link with one channel, then account for the full system configuration.

Real CXL Devices Differ

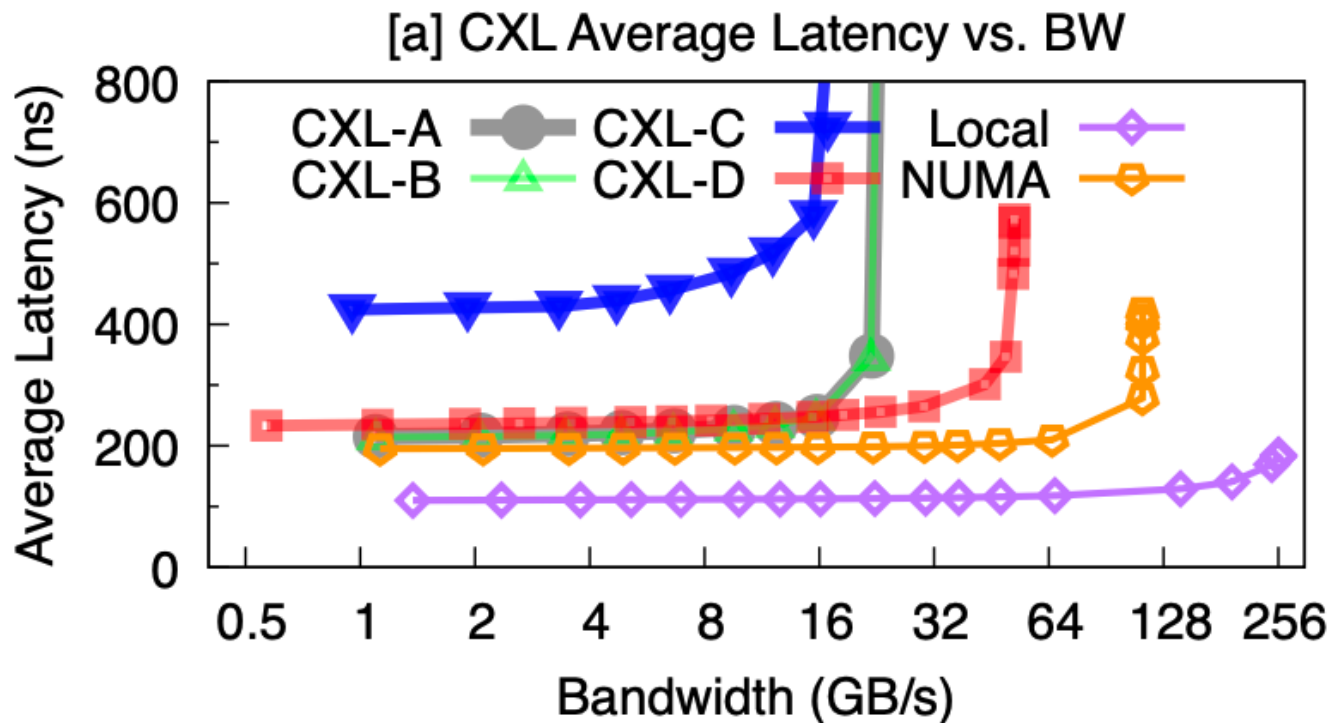
Melody Table 1. Same-socket access. MLC latency_matrix and bandwidth_matrix.

Device	Implementation	Link	Idle latency	Matrix BW
CXL-A	ASIC, DDR4	×8	214 ns	24 GB/s
CXL-B	ASIC, DDR5	×8	271 ns	22 GB/s
CXL-C	FPGA, DDR4	×8	394 ns	18 GB/s
CXL-D	ASIC, DDR5	×16	239 ns	52 GB/s

A/B/C: SPR2S or EMR2S. D: separate EMR2S-prime. Matrix BW does not specify a traffic mix.

- These are real Type 3 devices operating in CXL 1.1 mode.
- The host and device implementation matter. CXL-D uses a separate machine.

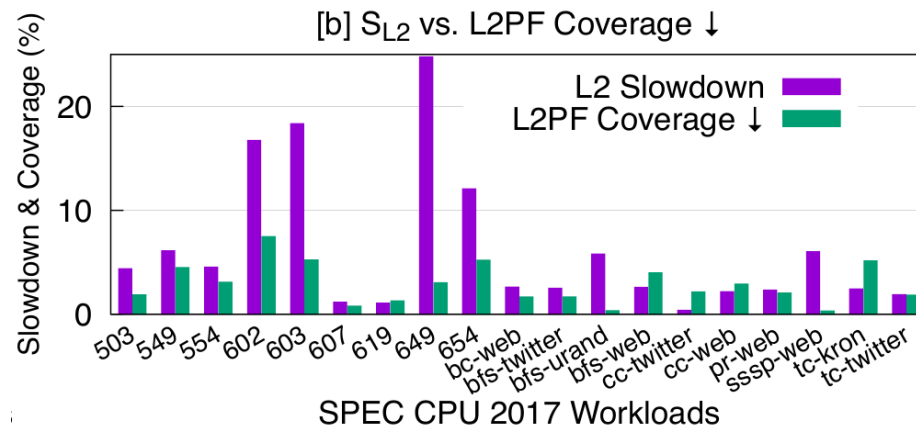
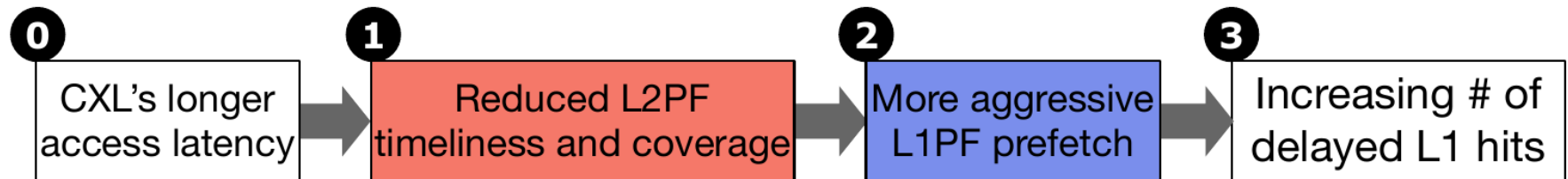
Loaded Latency and Saturation



MLC delay sweep: 0–20K cycles. X-axis logarithmic. Different device curves include different host setups.

- A foreground latency thread runs alongside 31 traffic generators.
- Latency rises as demand approaches a path's sustainable service rate.

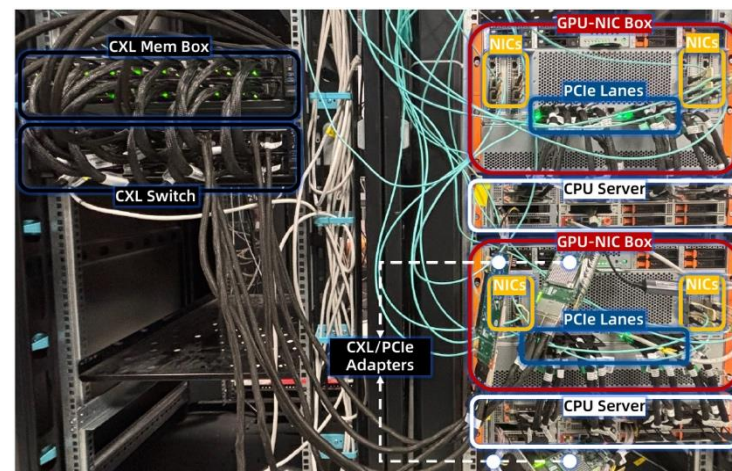
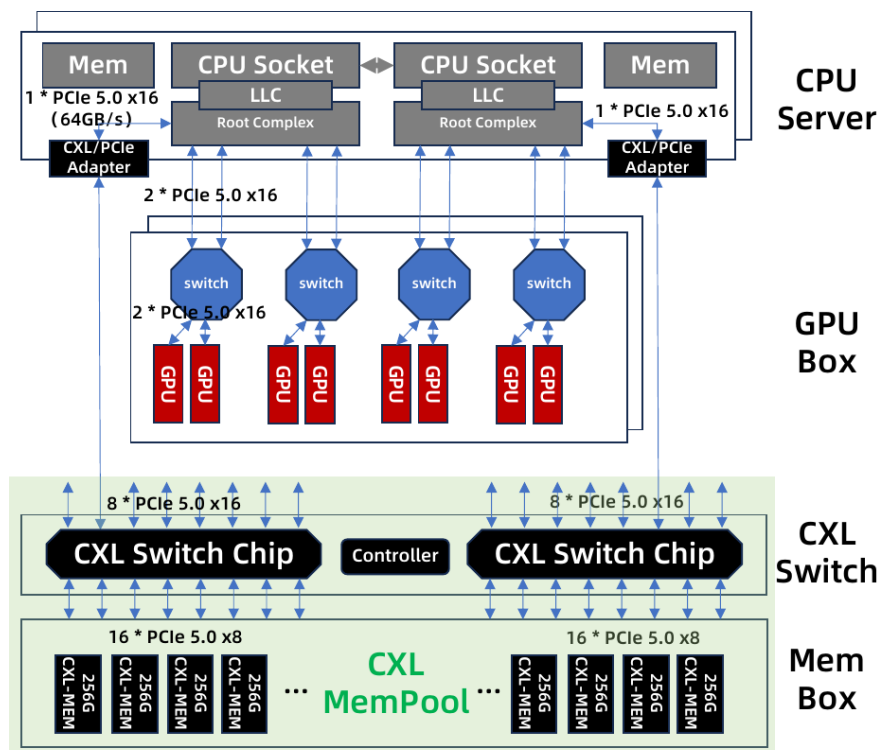
CXL Latency and CPU Prefetching



Melody published Figures 12(b) and 13. Findings are specific to the evaluated Intel platforms.

- Longer fills can reduce prefetch timeliness and coverage on the evaluated CPUs.
- The paper links available counter measurements to its interpretation of cache stalls.

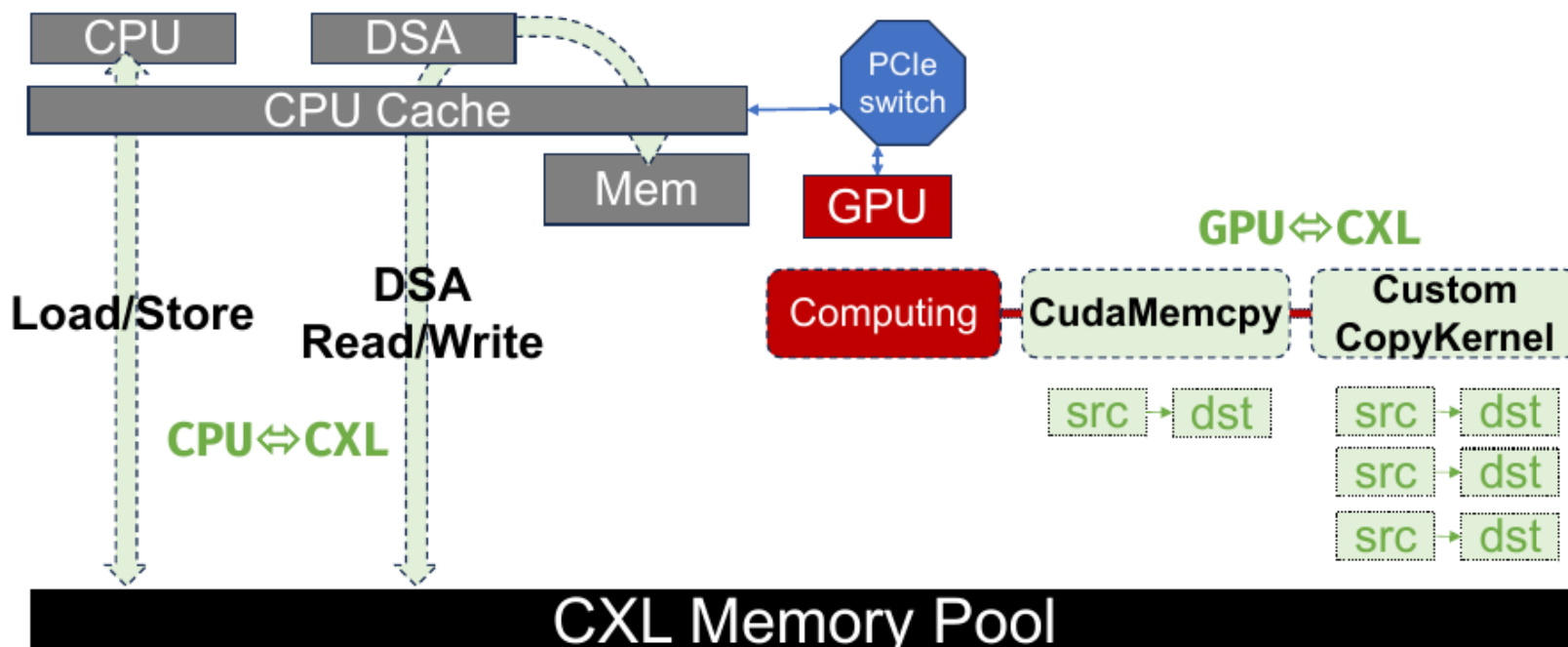
Beluga: A Built CXL Memory Pool



Physical deployment and evaluated configuration; 8 TB installed, 2 TB used for fair comparisons.

- Two GPU servers connect to an 8 TB CXL 2.0 memory pool.
- The evaluation uses 16 H20 GPUs and a matched 2 TB cache budget.

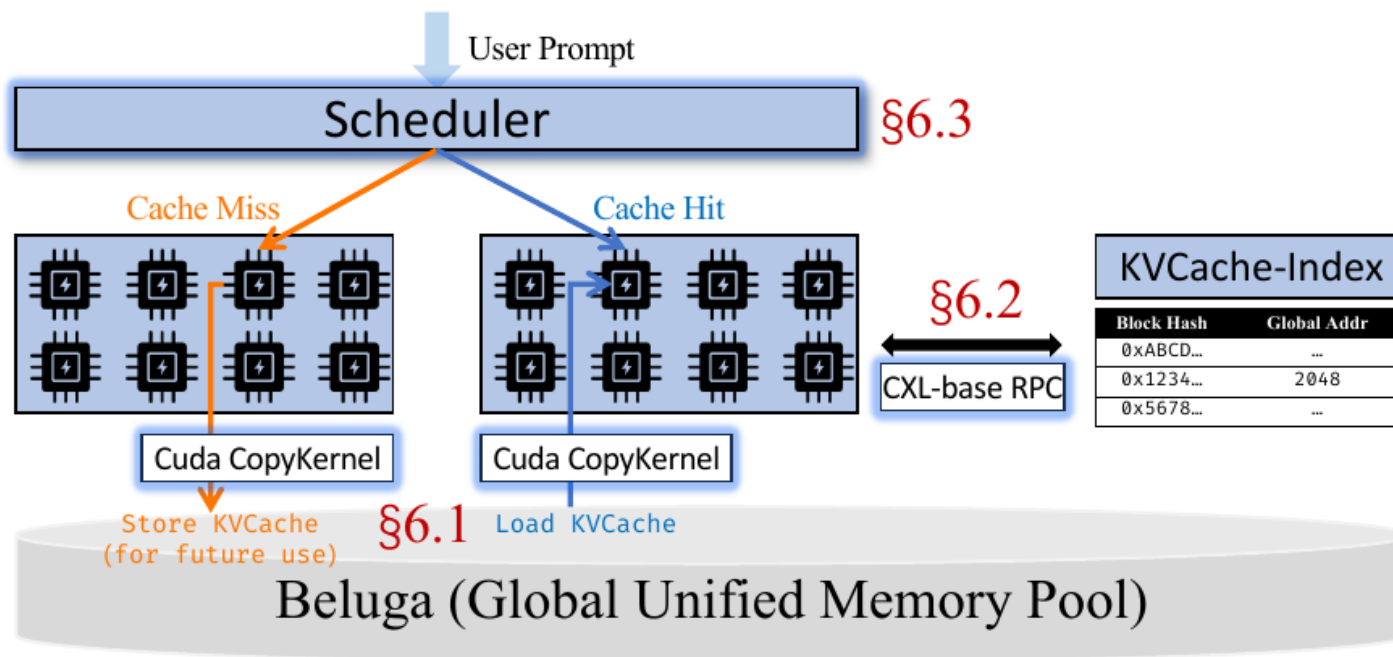
Beluga: Access Paths and Correctness



Beluga CXL 2.0 access paths and software-managed sharing; no hardware coherence across host caches.

- CPU loads, DSA transfers and GPU copy kernels reach the same pool.
- Software must make writes visible and prevent stale reads across hosts.

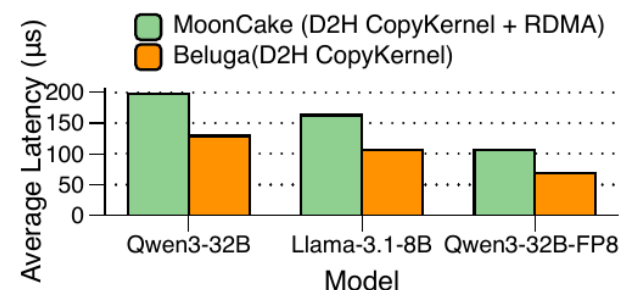
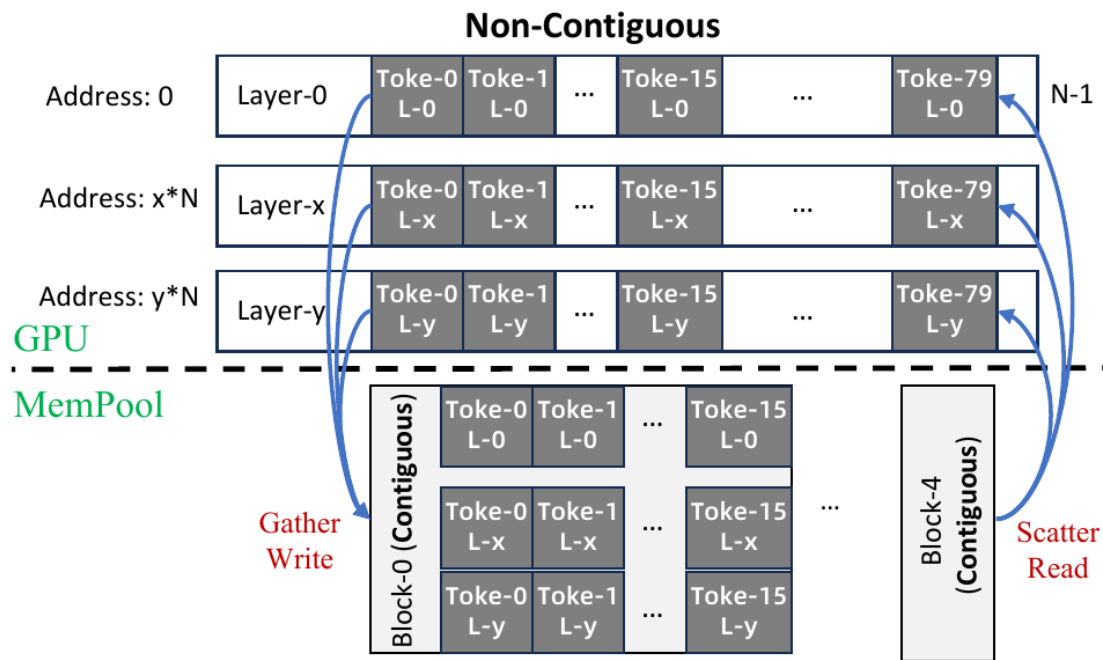
Beluga: KVCache Placement and Reuse



Beluga KVCache workflow, vLLM integration and CXL-based metadata RPC.

- A miss computes and stores KVCache for later reuse.
- A hit resolves a block address and loads KVCache from the pool.

Beluga: Gather and Scatter Transfers



Dense KVCache transfer microbenchmark, Figure 14(a); the percentage describes writes in the tested configurations.

- Gather writes pack scattered GPU regions into contiguous pool blocks.
- The dense-transfer benchmark reports 36.2% lower write latency than MoonCake.

Real Hardware and NUMA Emulation

Two setups in Beyond Hotness (OSDI 2025). Different hosts and local-memory baselines.

Target tier	Platform / path	Latency	Bandwidth
Real ASIC expander	Sapphire Rapids, PCIe 5.0 $\times 8$	271 ns	26 GB/s

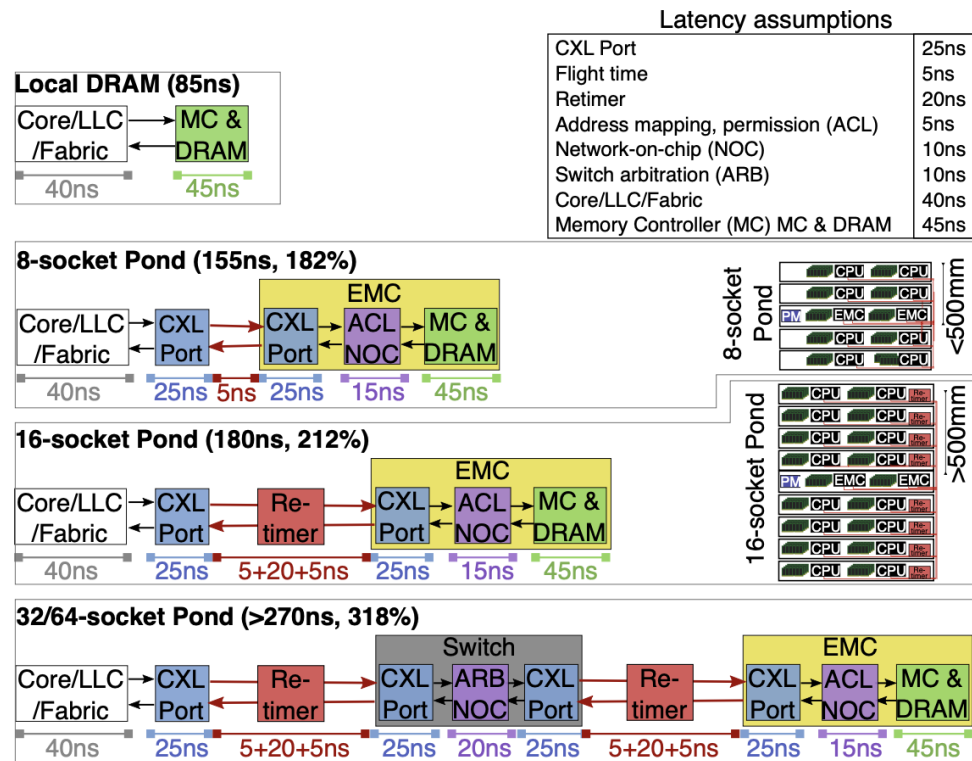
Emulated CXL tier	Skylake remote DRAM	190 ns	17 GB/s
-------------------	---------------------	--------	---------

Local DRAM on real-CXL host: 114 ns, 218 GB/s

Local DRAM on emulation host: 90 ns, 49 GB/s

- NUMA emulation controls an existing socket path rather than implementing CXL hardware.
- Keep the host baseline and measurement method when comparing reported results.

Pond's Topology Latency Budget



- Ports, flight time, retimers and switches contribute different costs.
- These are Pond's analytical budgets, not a measurement of every device.

Worked HPA/DPA Examples

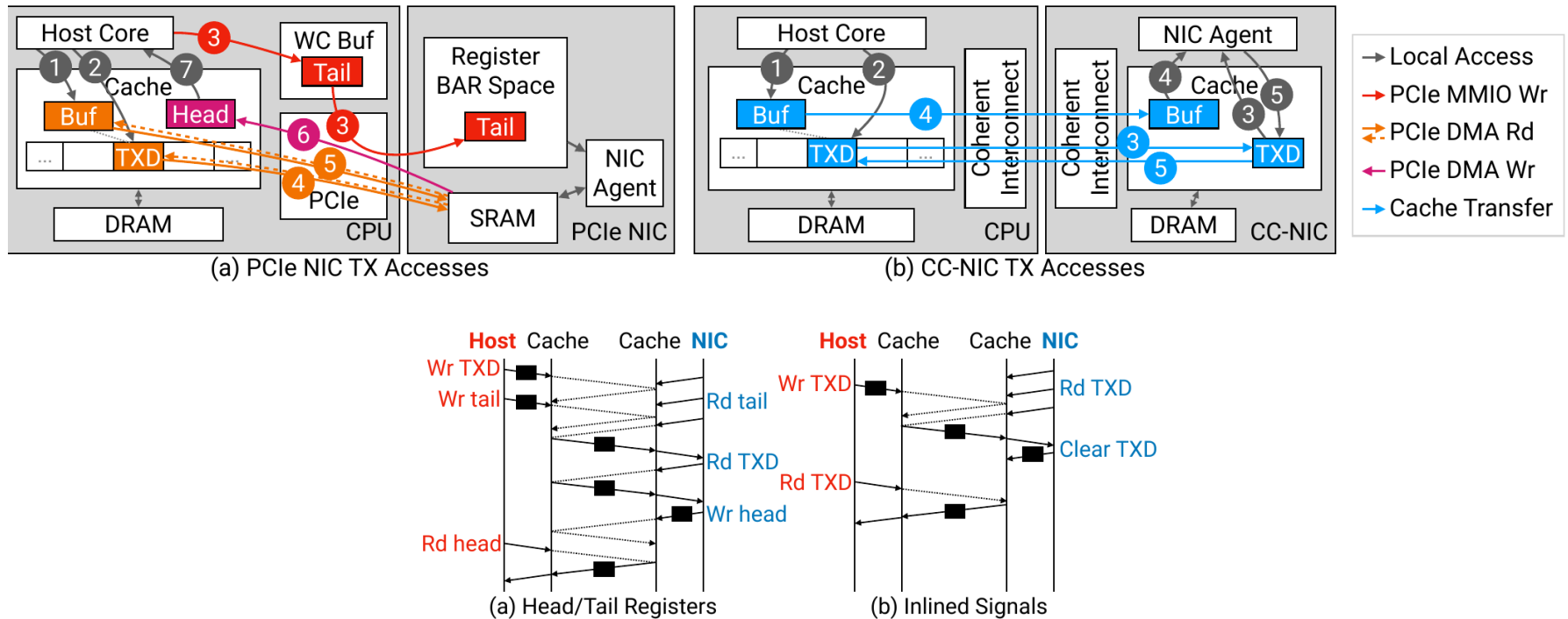
Noninterleaved: $0x40001240 - 0x40000000 = \text{DPA } 0x1240$

Illustrative two-way, 256 B stripes. Host base $0x40000000$. Device bases are zero.

Host physical address	Target	Device physical address
0x40000040	Device 0	0x00000040
0x40000140	Device 1	0x00000040
0x40000240	Device 0	0x00000140
0x40000340	Device 1	0x00000140

- Noninterleaved mapping subtracts the host window base in this example.
- Interleaving selects a target, then packs each target's stripes into DPA.

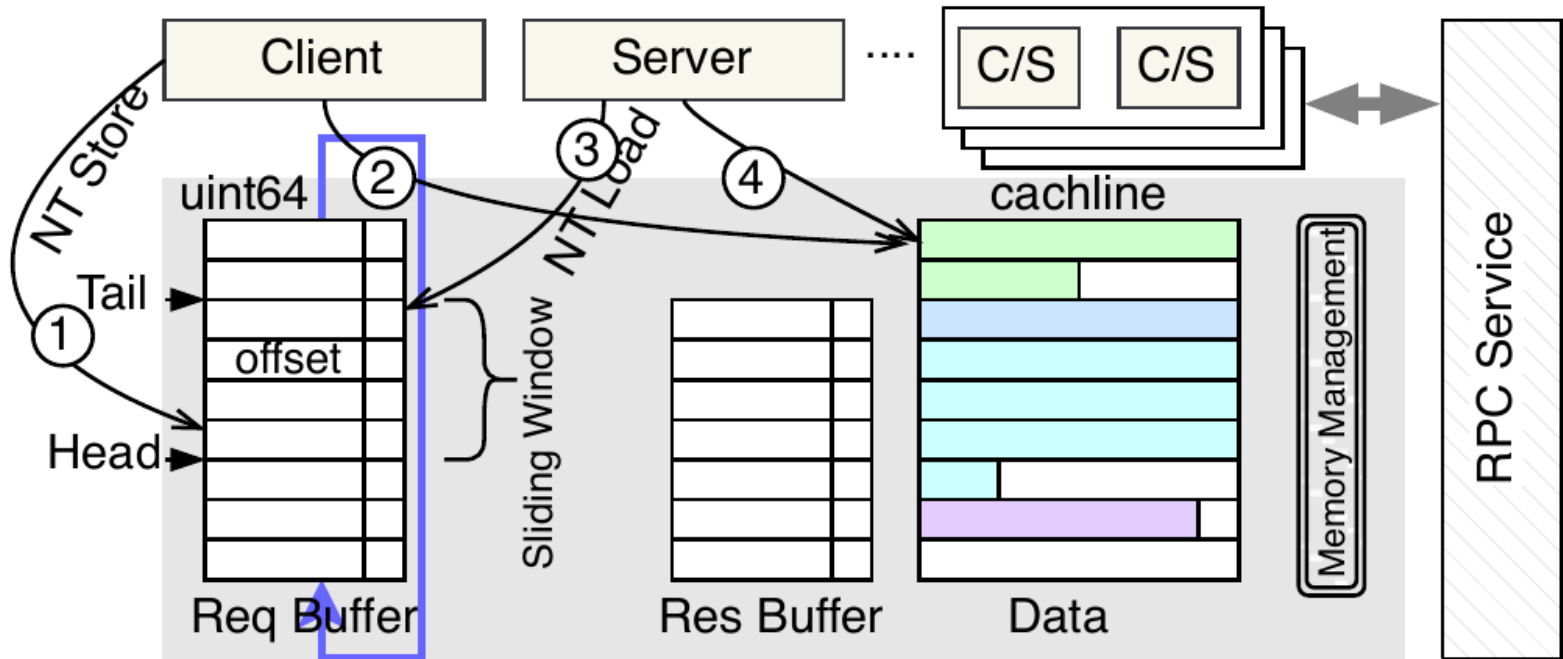
CC-NIC: Coherent Descriptors



Coherent NIC-interface design evaluated using Intel UPI; the figures compare transmit and signaling paths.

- CC-NIC carries readiness inside descriptors instead of separate head/tail registers.
- Cache transfers combine notification with descriptor delivery.

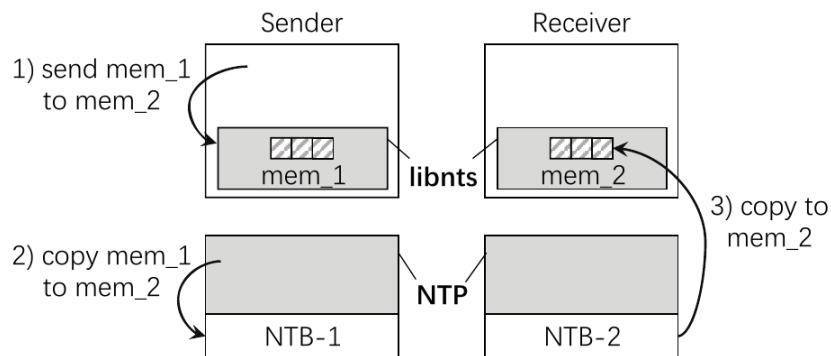
HydraRPC: Shared-Memory Rings



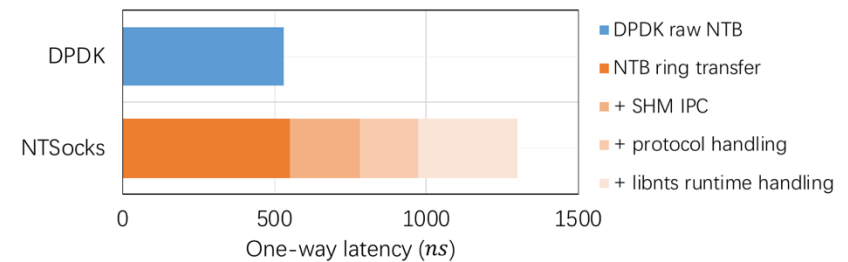
HydraRPC FPGA CXL prototype with CXL 1.1+ driver; software controls cache visibility and ordering.

- Each RPC connection uses request and response queues in shared CXL memory.
- A 64-bit entry carries a data offset and an arrival flag.

NTSocks: The NTB Fast Path



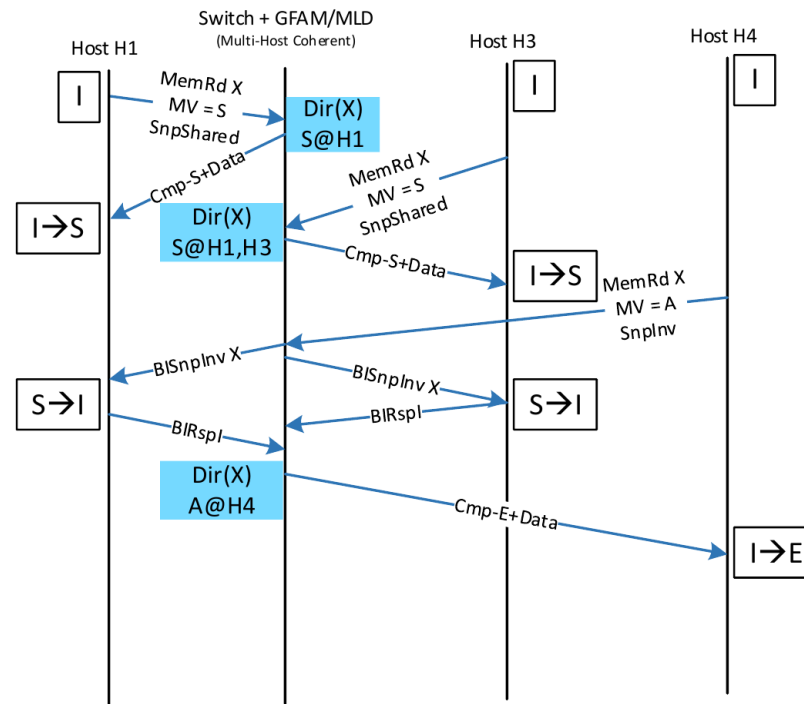
(c) The execution flow of *direct-nts*.



NTSocks evaluated two back-to-back servers using PCIe NTB. It is not a CXL prototype.

- Remote writes publish data to a peer buffer. The receiver reads locally.
- Socket semantics add buffering, queues and software processing.

CXL 3.0 Back-Invalidation

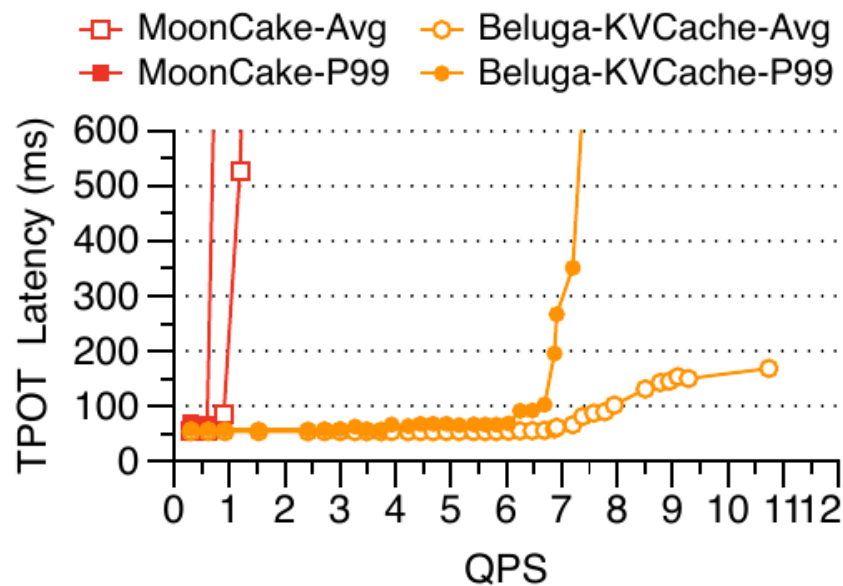
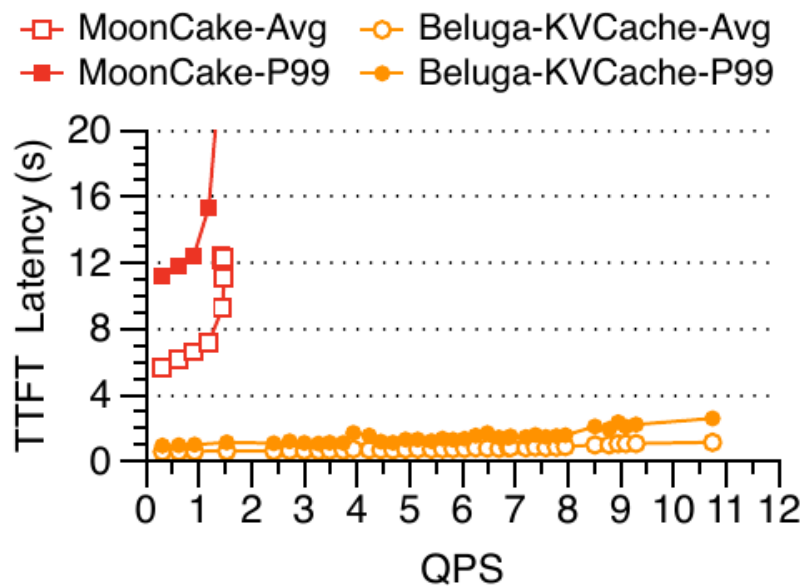


B: Shared Memory coherency across Multiple Domains with BI

CXL 3.0 mechanism with suitable host, device and fabric support.

- A sharing device tracks which hosts hold a cache line.
- Exclusive ownership waits for conflicting cached copies to be invalidated.

Beluga: End-to-End Behavior



(a) TTFT with different Request Rates **(b) TPOT with different Request Rates**

Populated-cache rerun on two servers with 16 H20 GPUs; both systems receive a 2 TB pool budget.

- Beluga sustains low TTFT and TPOT over a wider request-rate range.
- The experiment replays a populated cache, making KVCache reads the main bottleneck.