Memory Disaggregation with Coherent Interconnect Technologies

Lukas Wenzel, Felix Eberhardt, Andreas Grapentin, Prof. Dr. Andreas Polze
<firstname>.<lastname>@hpi.de
Operating Systems and Middleware Group, Hasso Plattner Institute

06.04.2023

Memory Disaggregation Concept

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

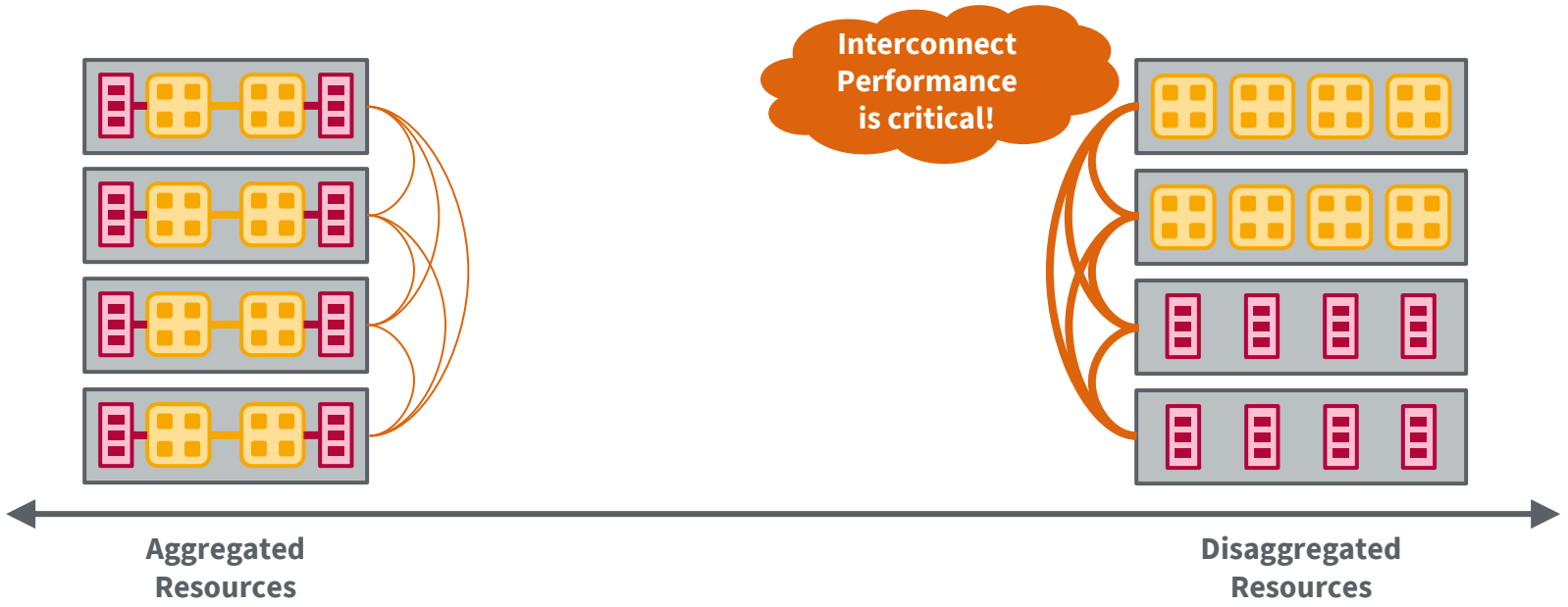
Disaggregation

The separation of an aggregate body into its component parts.

from the GNU version of the Collaborative International Dictionary of English

Instead of separate machines with a **fixed**, **tightly coupled** resource configuration

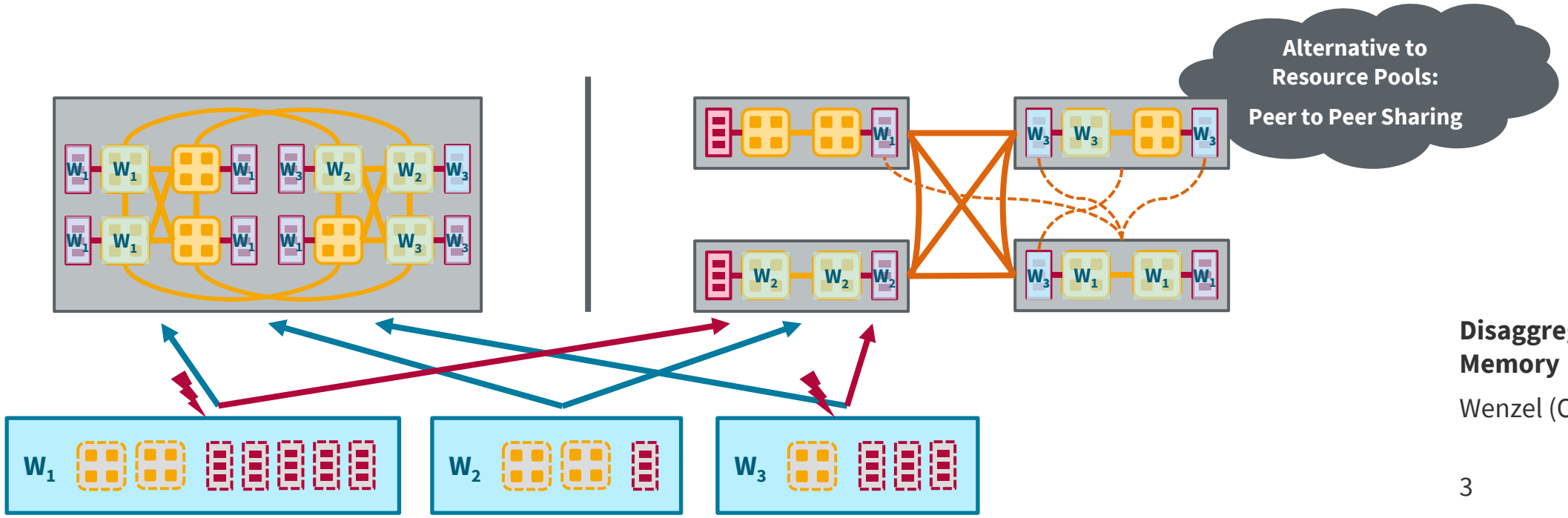
→ different resource pools with **dynamic**, **loosely coupled** associations



Memory Disaggregation
Benefits

Operational and provisioning flexibility:

- Cost efficiency
- Energy efficiency (?)



Disaggregated Memory

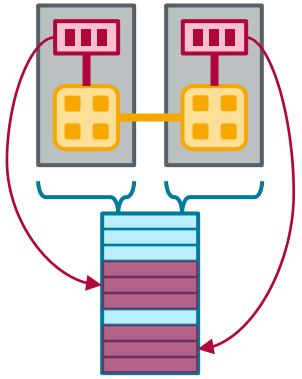
Wenzel (OSM HPI)

Memory Disaggregation
Related Concepts

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

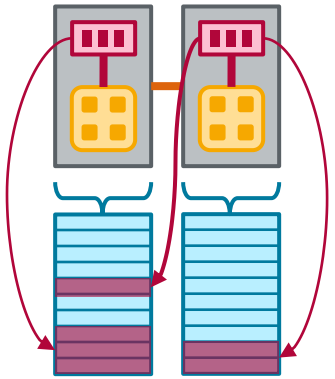
Non-Uniform Memory Access (NUMA) System

Nodes share a global physical address space,
where local and remote memory resources are accessible.



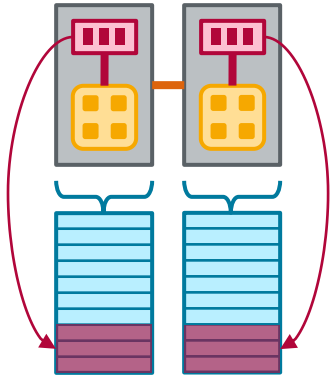
Disaggregated Memory System

Machines have private physical address spaces,
but may map remote memory resources (often exclusively, sometimes shared).



Distributed / Shared-Nothing System

Machines have private physical address spaces,
and can only access local memory resources.



Landscape
Early 1990s

Early proposals appeared in the wake of Multicomputer system developments

= Remote memory replaces local disks as **swap space** through custom page fault handlers or block device drivers

= Movement of complete pages, **often over LAN**

- D. Comer and J Griffioen **"A New Design for Distributed Systems: The Remote Memory Model."** USENIX Summer, **1990**.

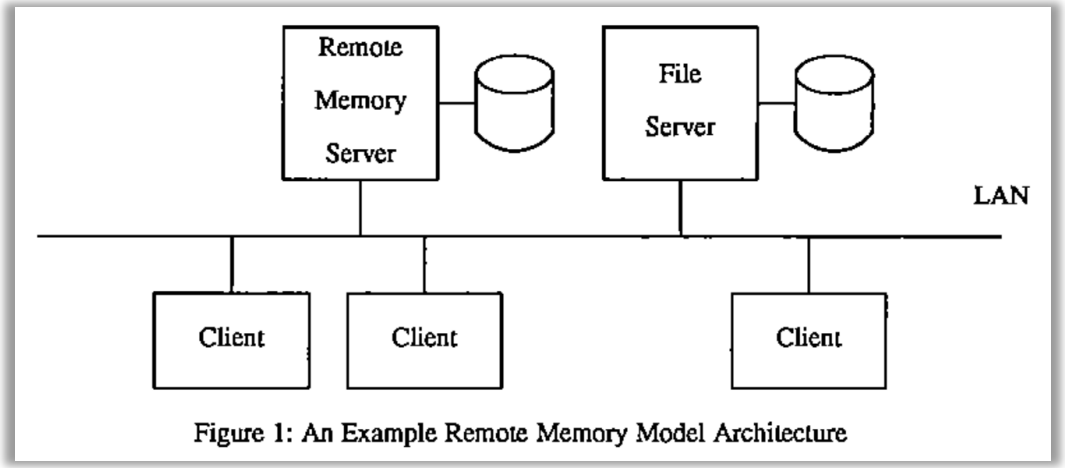


Figure 1: An Example Remote Memory Model Architecture

- E. W. Felten and J. Zahorjan "**Issues in the Implementation of a Remote Paging System."** Technical Report TR 91-03-09, Computer Science Department, University of Washington, **1991**.

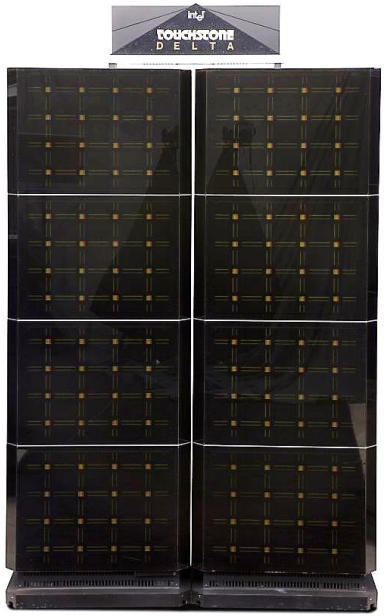
Landscape
Early 1990s

Early proposals appeared in the wake of Multicomputer system developments

= Remote memory replaces local disks as **swap space** through custom page fault handlers or block device drivers

= Movement of complete pages, **often over LAN**

- L. Iftode, et al. "**Memory Servers for Multicomputers.**" In Digest of Papers. Compcon Spring (pp. 538-547). IEEE, **1993**.



From Computer History Museum

https://www.computerhistory.org/revolution/supercomputers/10/74/288

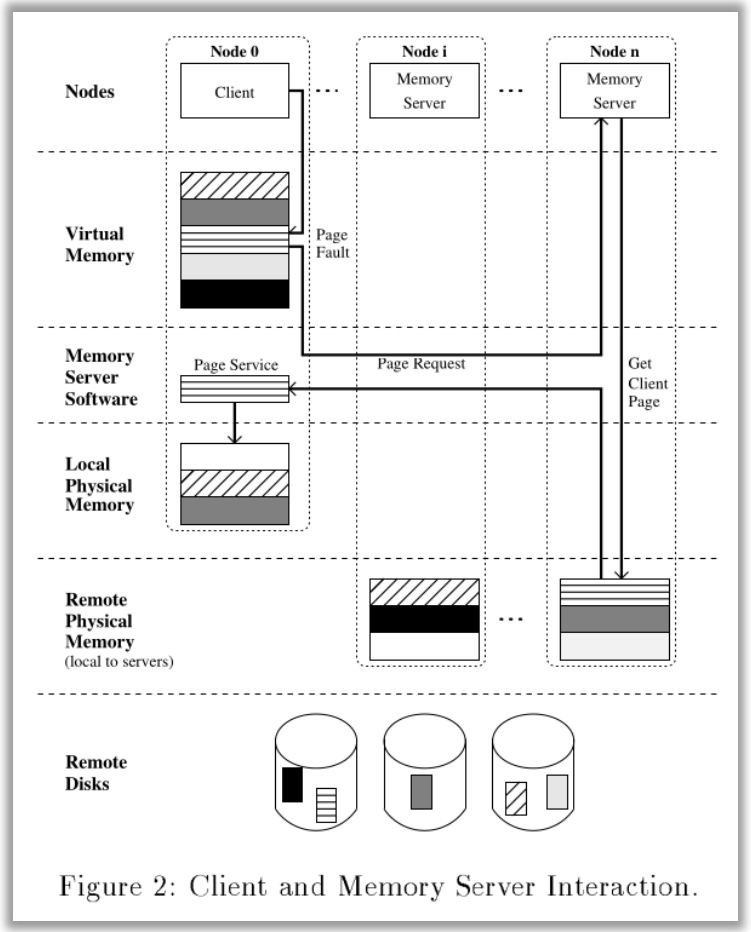


Figure 2: Client and Memory Server Interaction.

Landscape
Early 1990s

Early proposals appeared in the wake of Multicomputer system developments

= Remote memory replaces local disks as **swap space** through custom page fault handlers or block device drivers

= Movement of complete pages, **often over LAN**

- M. J. Feeley, et al. **"Implementing Global Memory Management in a Workstation Cluster."** In Proceedings of the 15th ACM symposium on operating systems principles (pp. 201-212), **1995**.

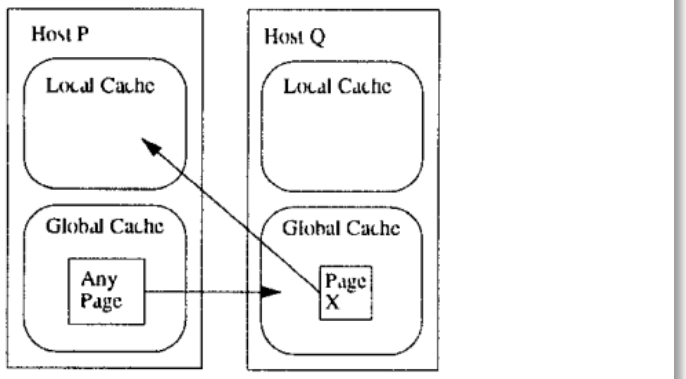


Figure 1: Global replacement with hit in the global cache.

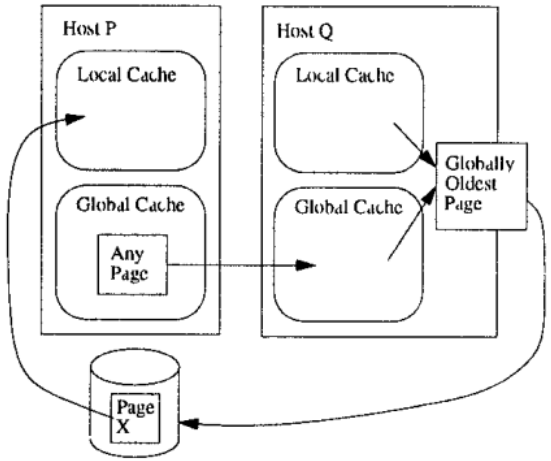


Figure 2: Global replacement showing miss in the global cache. The faulted page is read from disk, and the oldest page in the network is either discarded (if clean) or written back to disk.

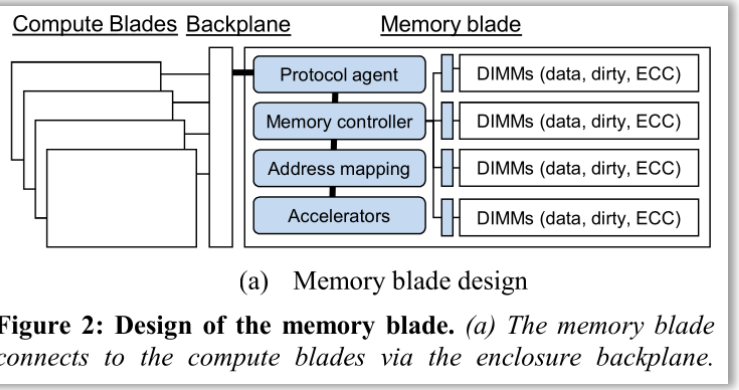
Landscape
Late 2000s to early 2010s

Advances in interconnect technology brought new approaches

= **Backplane** connections via PCIe or coherent socket-to-socket interfaces

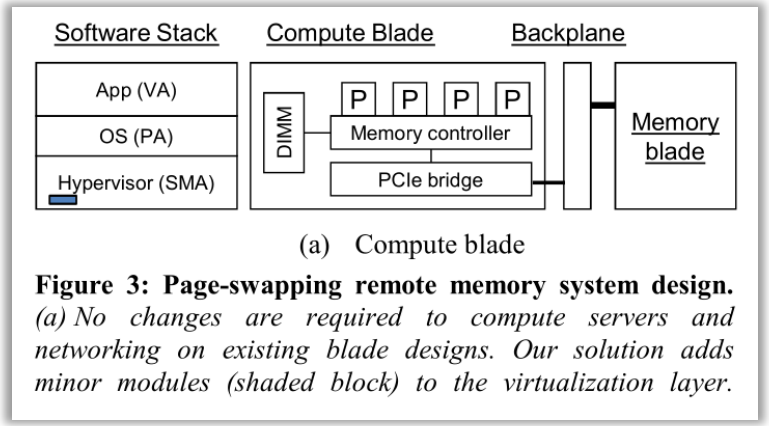
= First steps in **fine grained** remote memory **access**, i.e., cache-line granularity

- K. Lim, et al. **"Disaggregated Memory for Expansion and Sharing in Blade Servers."** ACM SIGARCH computer architecture news, 37(3), 267-278, **2009**.



(a) Memory blade design

Figure 2: Design of the memory blade. *(a) The memory blade connects to the compute blades via the enclosure backplane.*



(a) Compute blade

Figure 3: Page-swapping remote memory system design. *(a) No changes are required to compute servers and networking on existing blade designs. Our solution adds minor modules (shaded block) to the virtualization layer.*

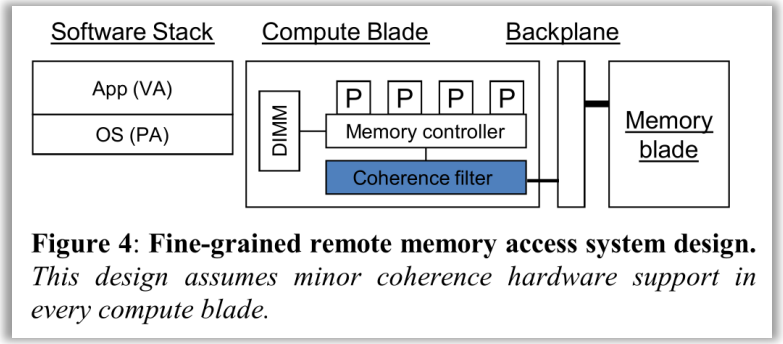


Figure 4: **Fine-grained remote memory access system design.** *This design assumes minor coherence hardware support in every compute blade.*

Landscape
Late 2000s to early 2010s

Advances in interconnect technology brought new approaches

= **Backplane** connections via PCIe or coherent socket-to-socket interfaces

= First steps in **fine grained** remote memory **access**, i.e., cache-line granularity

- H. Montaner, et al. **"A Practical Way to Extend Shared Memory Support Beyond a Motherboard at Low Cost."** In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (pp. 155-166), **2010**.

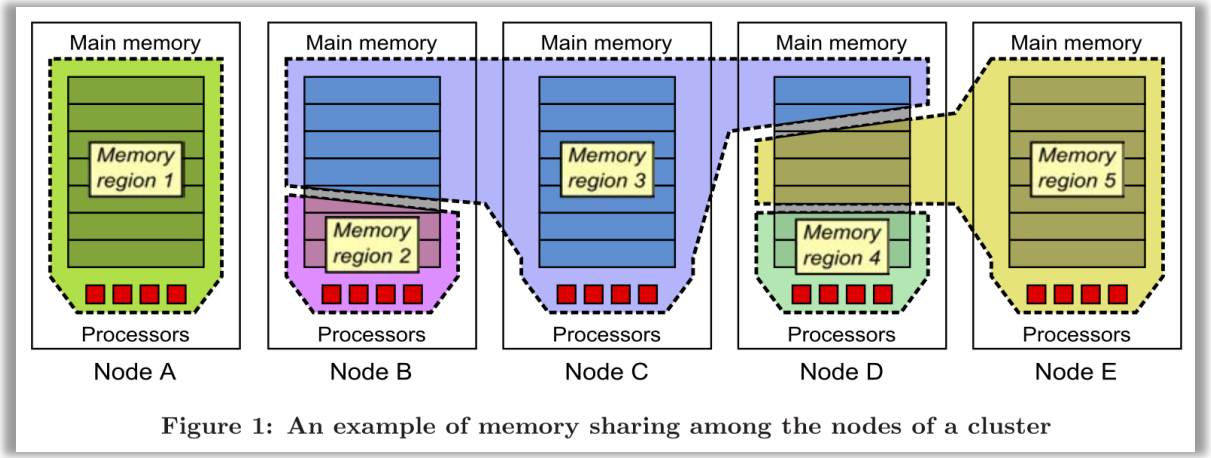


Figure 1: An example of memory sharing among the nodes of a cluster

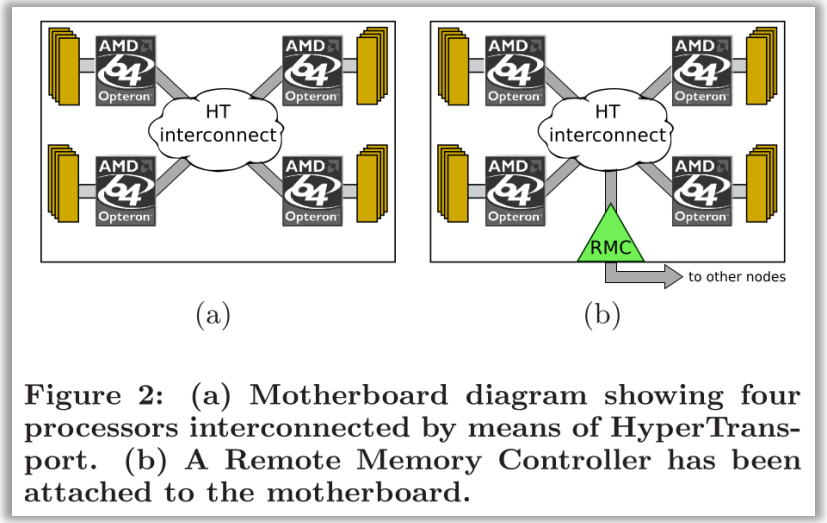


Figure 2: (a) Motherboard diagram showing four processors interconnected by means of HyperTransport. (b) A Remote Memory Controller has been attached to the motherboard.

Landscape
Late 2000s to early 2010s

Advances in interconnect technology brought new approaches

= **Backplane** connections via PCIe or coherent socket-to-socket interfaces

= First steps in **fine grained** remote memory **access**, i.e., cache-line granularity

- R. Hou, et al. **"Cost Effective Data Center Servers."** In 19th International Symposium on High Performance Computer Architecture (HPCA) (pp. 179-187), IEEE, **2013**.

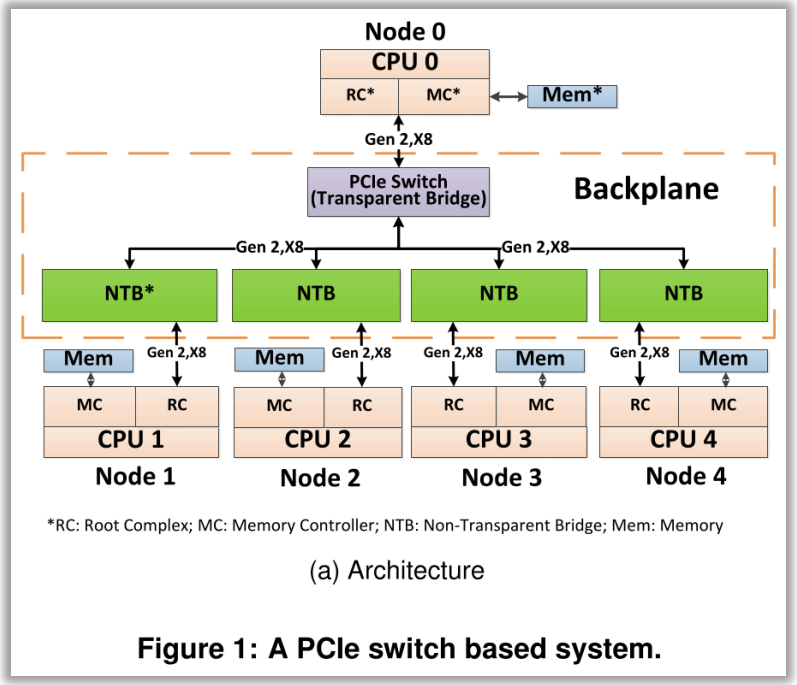
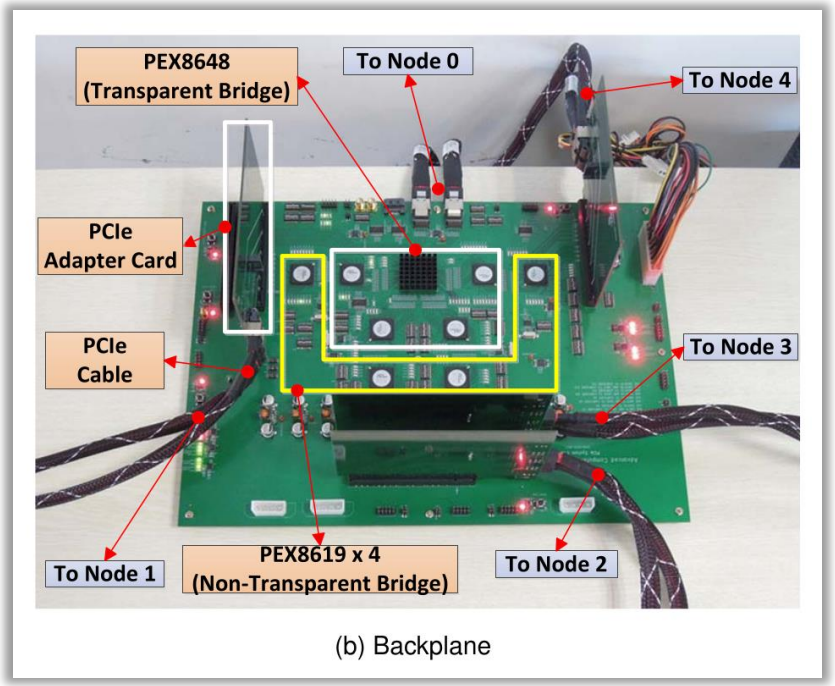


Figure 1: A PCIe switch based system.

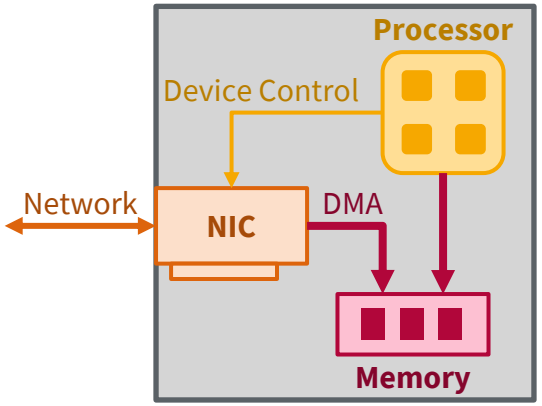


HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Remote Direct Memory Access (RDMA)

= Protocol and NIC technology allowing **network memory access without processor activity**

- Originally layered on top of InfiniBand, later on Ethernet (RoCE) and even SCTP or TCP transports (iWARP)
- NIC implements RDMA protocol without processor support
- NIC has autonomous DMA access to the memory subsystem
  - ➢ Operating System must allocate buffers and setup mappings and access permissions first



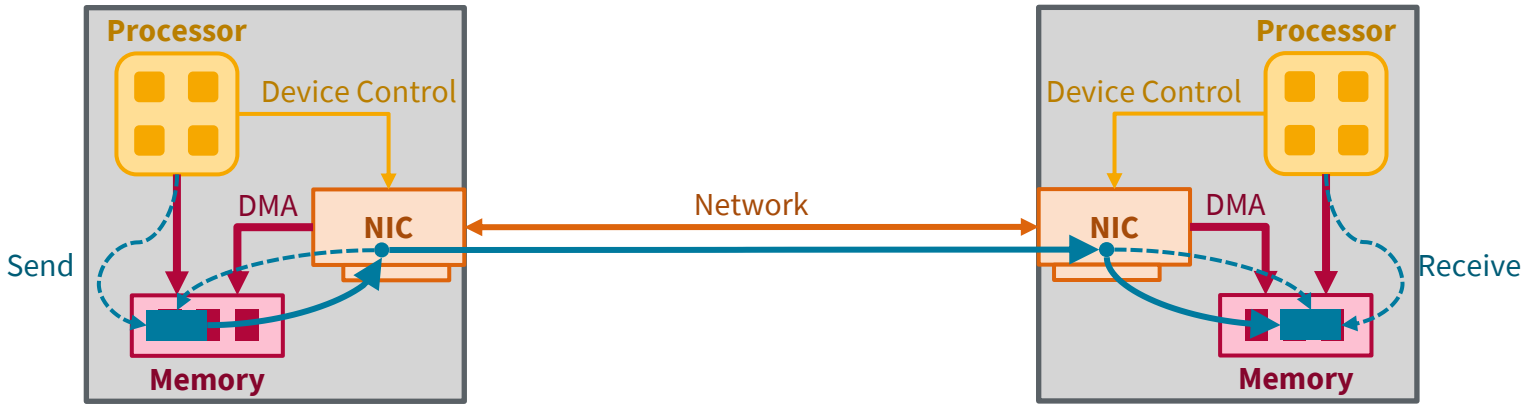
HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Remote Direct Memory Access (RDMA)

= Protocol and NIC technology allowing **network memory access without processor activity**

Operations (Verbs):

- Message Semantics, two-sided (Send and Receive)
  - Receiver prepares receive buffer and requests Receive
  - Sender fills send buffer and requests Send
  - Sender NIC reads send buffer, transmits to Receiver NIC, which writes receive buffer



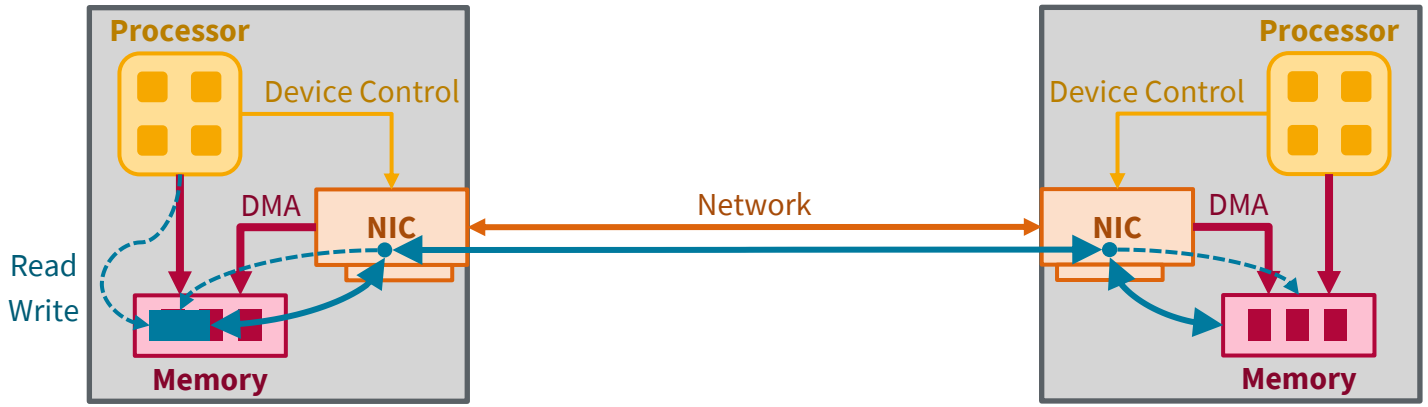
HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Remote Direct Memory Access (RDMA)

= Protocol and NIC technology allowing **network memory access without processor activity**

Operations (Verbs):

- Memory Semantics, one-sided (Read or Write)
  - Reader/Writer sends RDMA request to remote NIC
  - Remote NIC performs memory access on Readers/Writers behalf



Landscape
RDMA

RDMA sparked wide range of disaggregated memory proposals

= **Serving** memory is handled by RDMA NIC with **low overhead**

= **Using** remote memory requires **explicit initiation** of RDMA transactions

- Variety of approaches place RDMA initiative on different components
  - Libraries
  - Runtime environment
  - Operating system
  - Hypervisor

➢ Comprehensive survey for further details and references:

M. Ewais and P. Chow **"Disaggregated Memory in the Datacenter: A Survey."** *IEEE Access* 11: 20688-20712, **2023**.

Landscape
Scalable Coherent Interconnects

Classic Coherent Interconnects (like HyperTransport, QPI, UPI, ...) designed for multi-socket machines

- Limited reach beyond board or backplane
- Limited interoperability between vendors and device types

In 2016, several consortia formed to standardize coherent interconnects

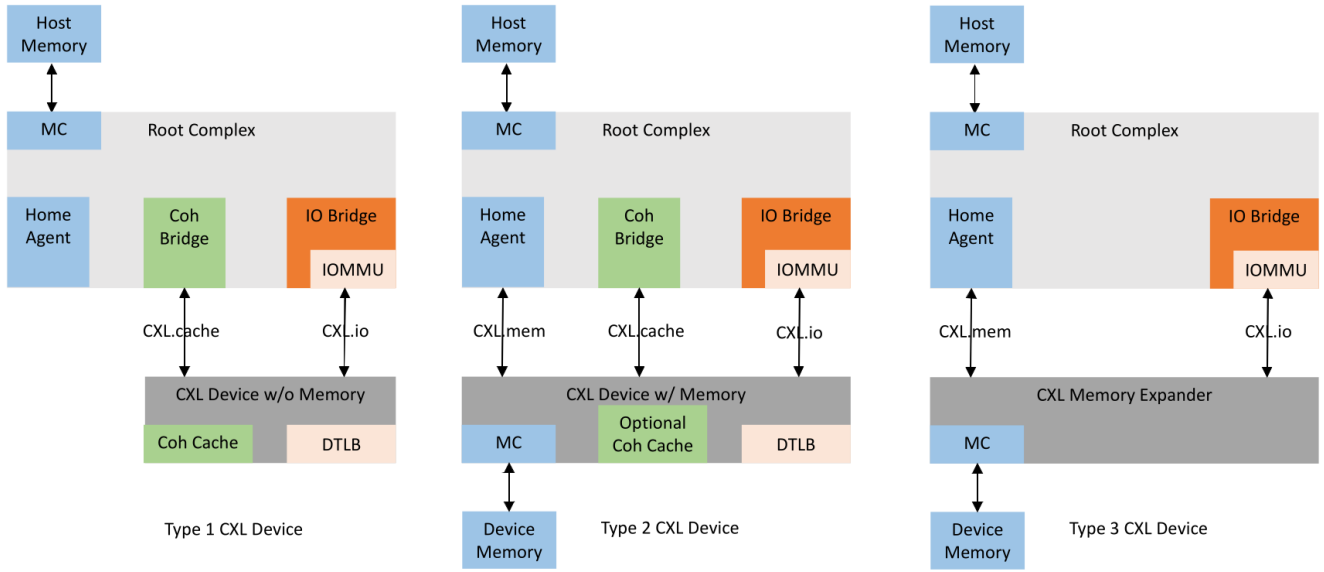
- **Gen-Z**: switchable, envisioned flexibility to span multiple racks
- **CCIX**: point-to-point, announced but not validated support in AMD Epyc 7002 (x86) and Huawei TaiShan 200 (ARM)
- **OpenCAPI**: point-to-point, evolved from IBM proprietary CAPI interface, available on POWER8 and POWER9

In 2019, **CXL** consortium was formed,
previous efforts started to converge (GenZ transferred in 2021, OpenCAPI in 2022)

Compute Express Link

- Layered on top of **PCIe**
- Processor (host) implements different **subprotocols** with separate semantics:
  - **CXL.io**: discovery, register access, interrupts (~regular PCIe)
  - **CXL.cache**: coherent host memory access (device → host, full or io coherency)
  - **CXL.mem**: access to device memory resources (host → device)

Figure 2-1. CXL Device Types



From: Compute Express Link Specification, Revision 3.0, Version 1.0

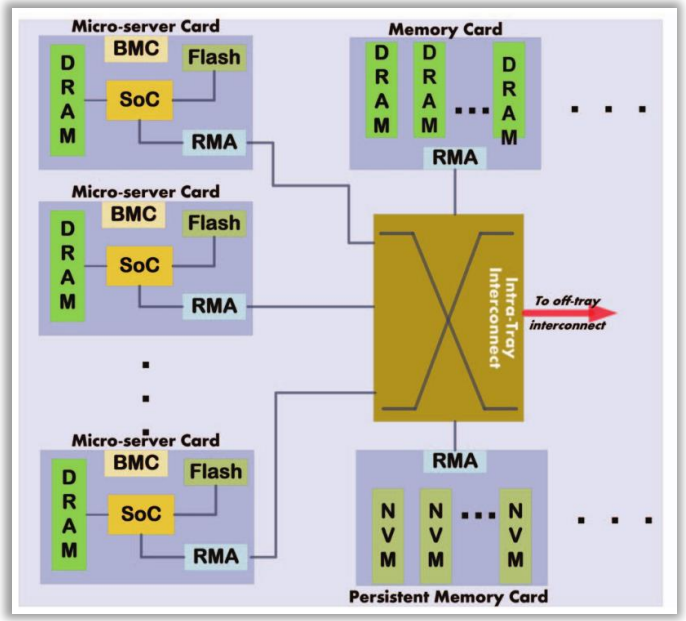
Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- K. Katrinis, et al. **"Rack-Scale Disaggregated Cloud Data Centers: The dReDBox Project Vision."** Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, **2016**.

Position Paper



- I. Calciu, et al. **"Rethinking Software Runtimes for Disaggregated Memory."** Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, **2021**.

Simulated System

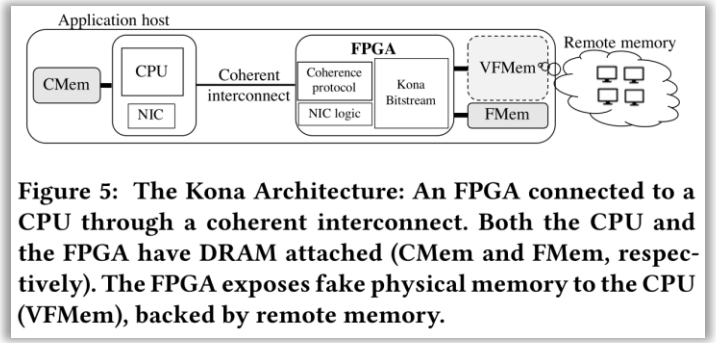


Figure 5: The Kona Architecture: An FPGA connected to a CPU through a coherent interconnect. Both the CPU and the FPGA have DRAM attached (CMem and FMem, respectively). The FPGA exposes fake physical memory to the CPU (VFMem), backed by remote memory.

Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- V. R. Kommareddy, et al. **"DeACT: Architecture-Aware Virtual Memory Support for Fabric Attached Memory Systems."** IEEE International Symposium on High-Performance Computer Architecture (HPCA), **2021**.

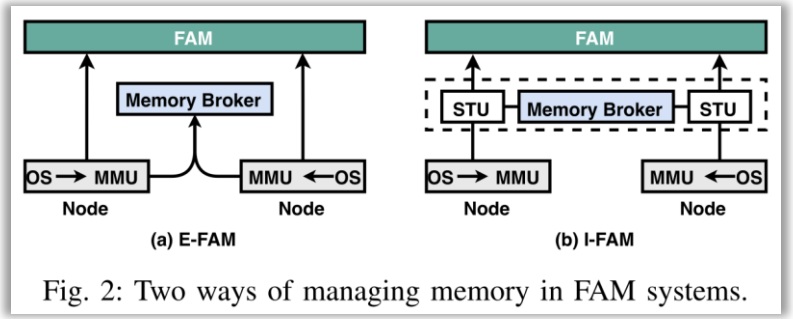


Fig. 2: Two ways of managing memory in FAM systems.

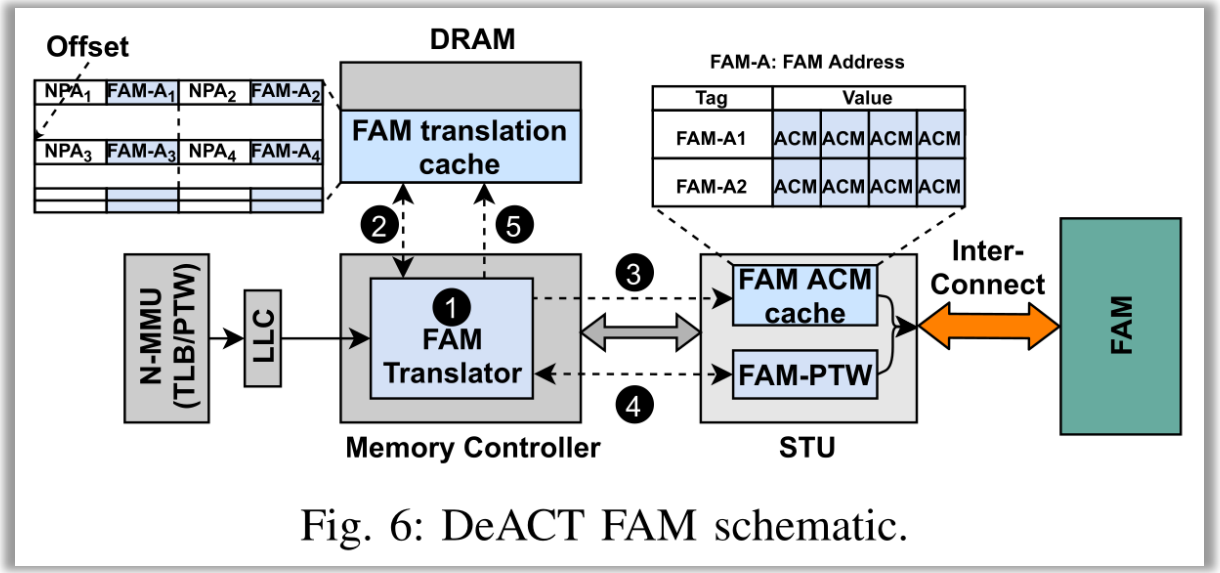


Fig. 6: DeACT FAM schematic.

Disaggregated Memory

Wenzel (OSM HPI)

Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- C. Pinto, et al. **"ThymesisFlow: A Software-Defined, HW/SW co-Designed Interconnect Stack for Rack-Scale Memory Disaggregation."** 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), **2020**.

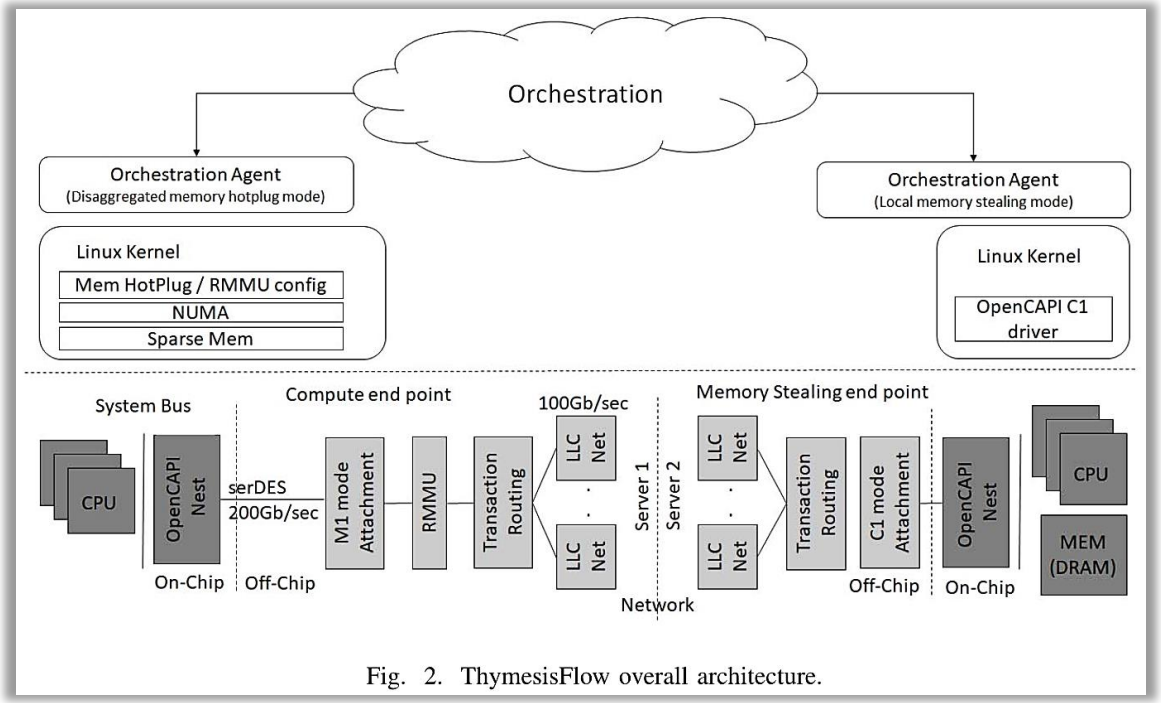


Fig. 2. ThymesisFlow overall architecture.

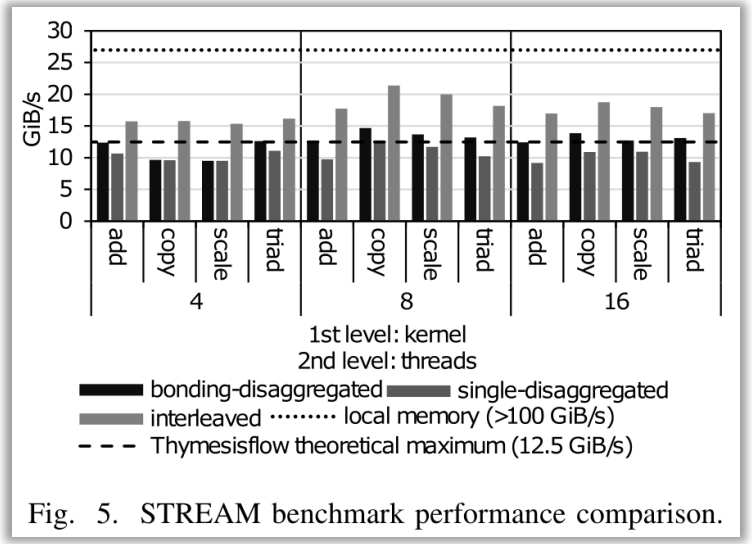


Fig. 5. STREAM benchmark performance comparison.

*"RTT latency of this prototype is roughly **950ns**, which includes four crossings of the FPGA stack and six serDES crossings."*

Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- D. Gouk, et al. **"Direct Access, High-Performance Memory Disaggregation with DirectCXL."** USENIX ATC, **2022**.

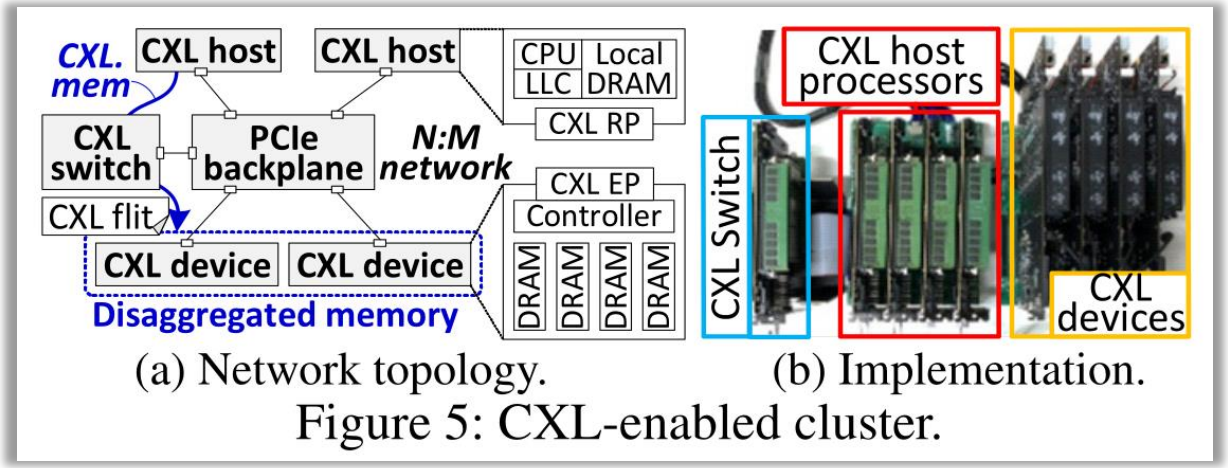


Figure 5: CXL-enabled cluster.

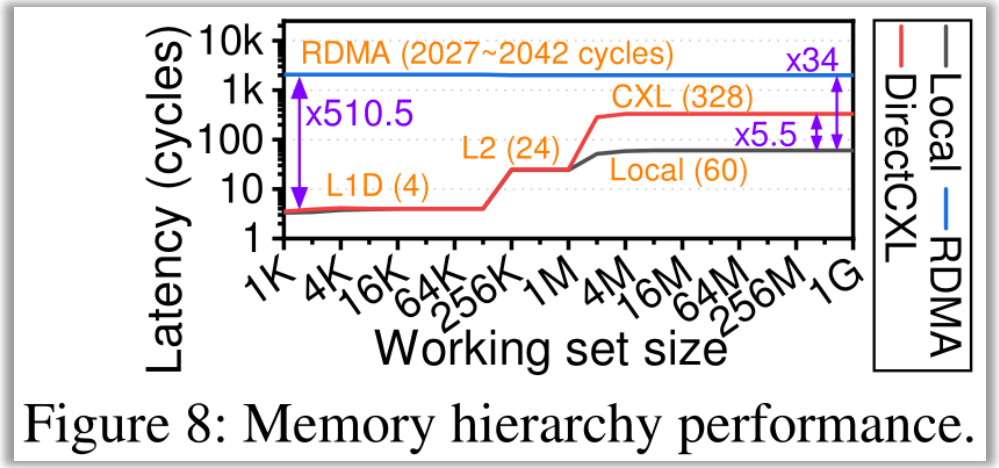


Figure 8: Memory hierarchy performance.

Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- C. Wang, et al. **"CXL over Ethernet: A Novel FPGA-based Memory Disaggregation Design in Data Centers."** arXiv preprint arXiv:2302.08055, **2023**.

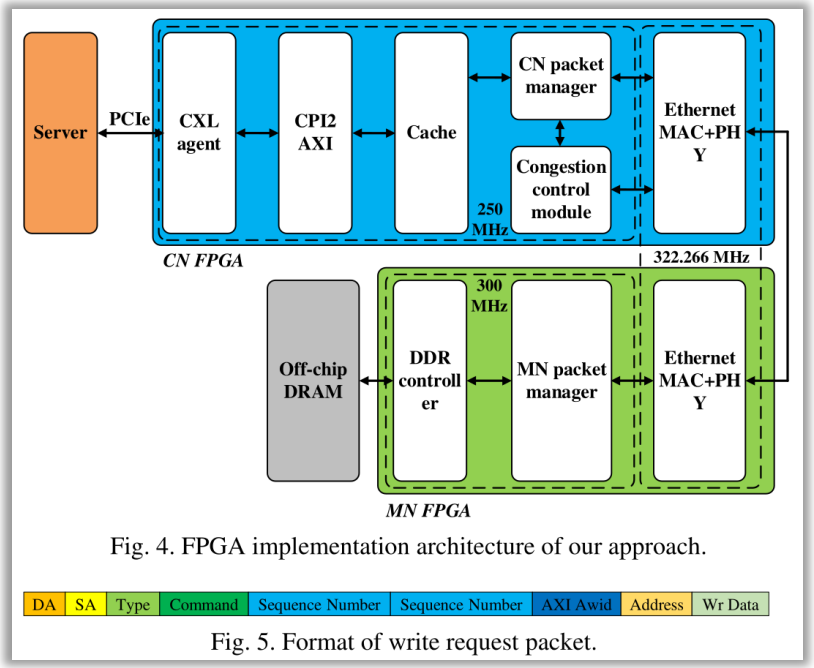
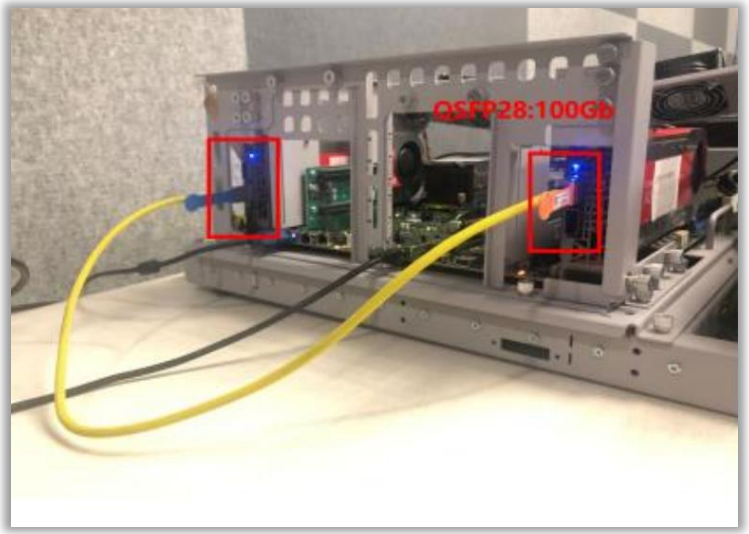


Fig. 4. FPGA implementation architecture of our approach.

Fig. 5. Format of write request packet.



Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- C. Wang, et al. **"CXL over Ethernet: A Novel FPGA-based Memory Disaggregation Design in Data Centers."** arXiv preprint arXiv:2302.08055, **2023**.

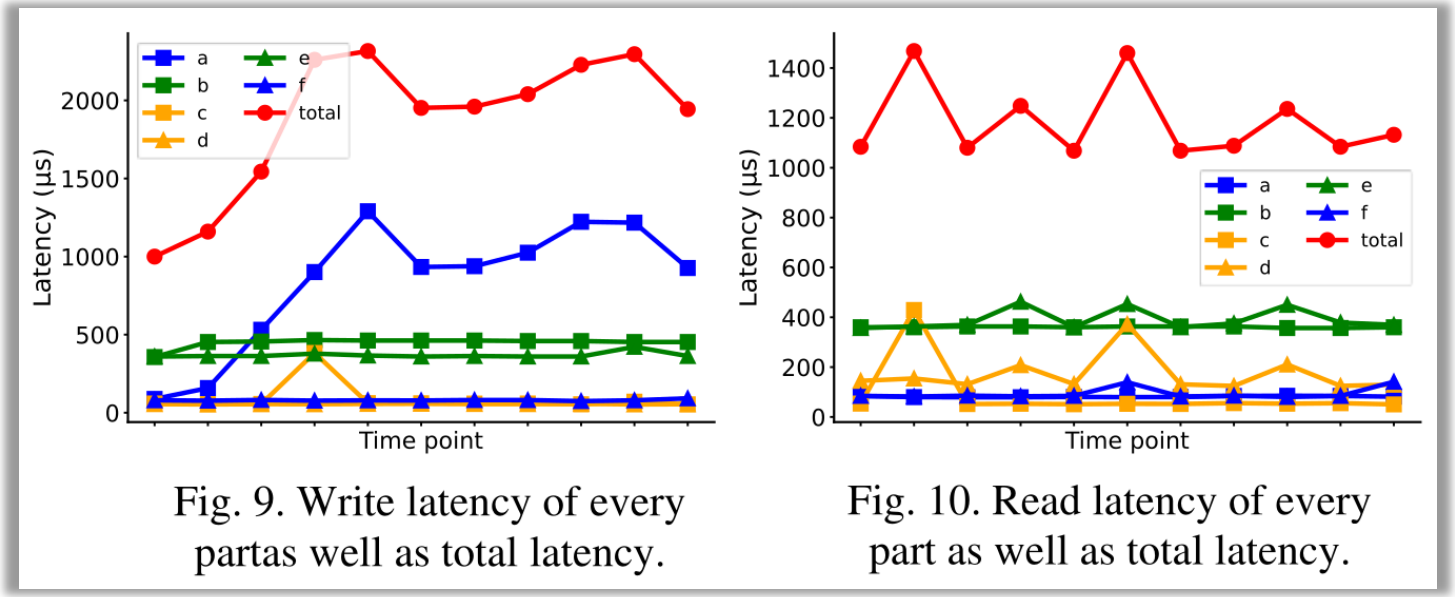
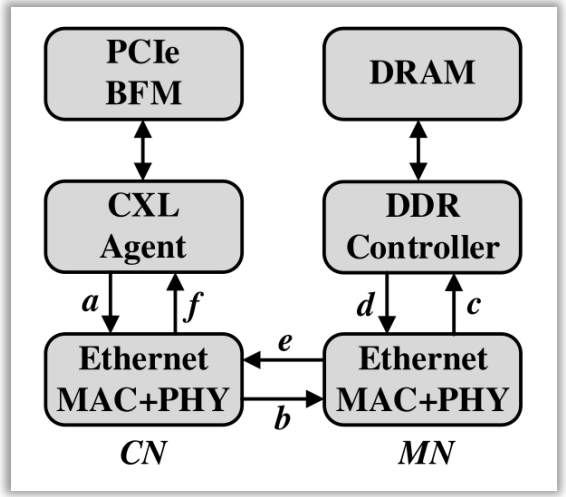


Fig. 9. Write latency of every partas well as total latency.

Fig. 10. Read latency of every part as well as total latency.



Disaggregated Memory

Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- Y. Sun, et al. **"Demystifying CXL Memory with Genuine CXL-Ready Systems and Devices."** arXiv preprint arXiv:2303.15375, **2023**.

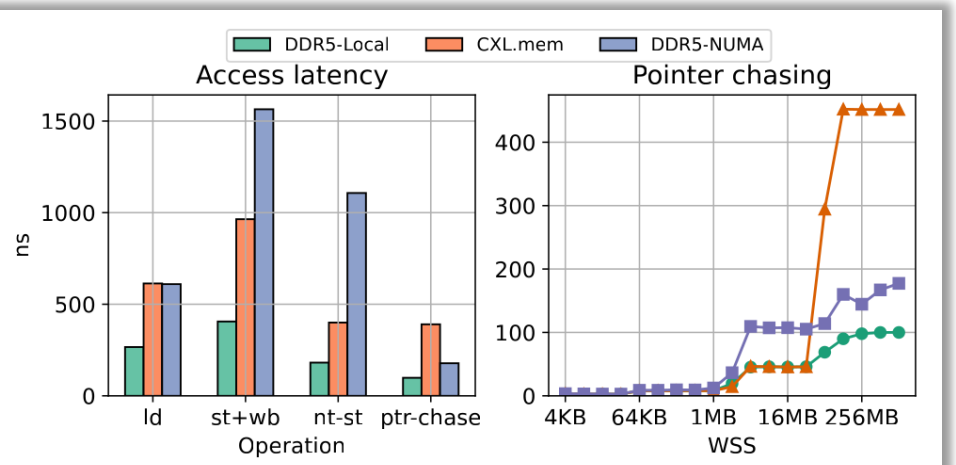
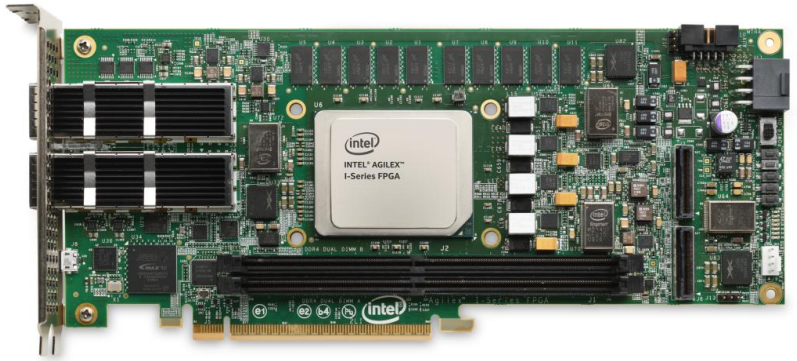


Figure 2: Access latency. Average latency for single AVX512 load (ld), store and write back (st+wb), non-temporal store (nt-st), and sequential pointer chasing in 1GB space(ptr-chase); Seqneutial pointer chasing latency vs. working set size (right). Prefetching at all levels are disabled in both cases

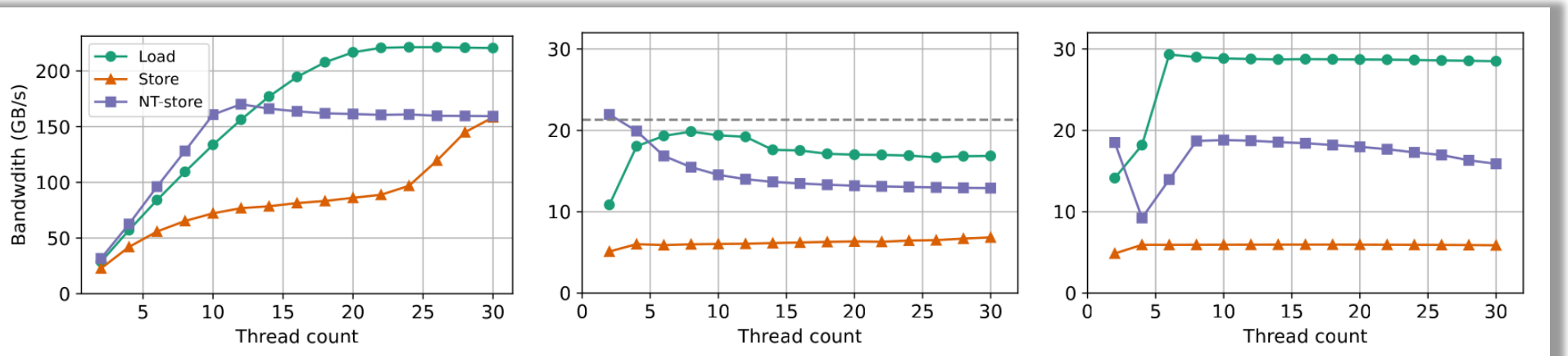


Landscape
Scalable Coherent Interconnects

Coherent interconnects enable new design space for disaggregated memory clients

= Application **load-store accesses** can be **forwarded transparently** to remote memory server

- Y. Sun, et al. **"Demystifying CXL Memory with Genuine CXL-Ready Systems and Devices."** arXiv preprint arXiv:2303.15375, **2023**.



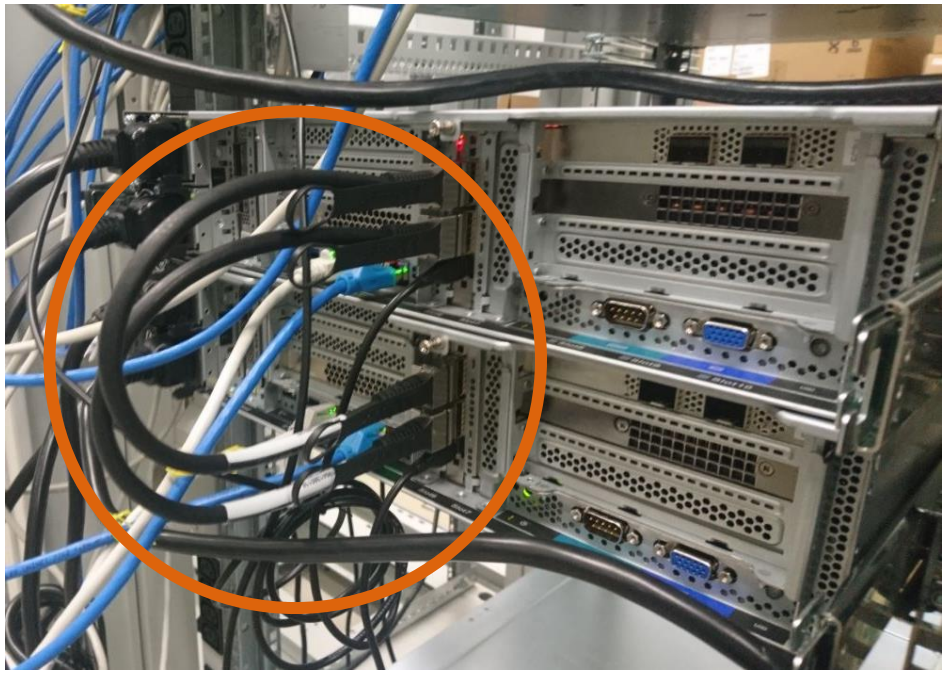
(a) DDR5-L8 **(b)** CXL memory **(c)** DDR5-R1

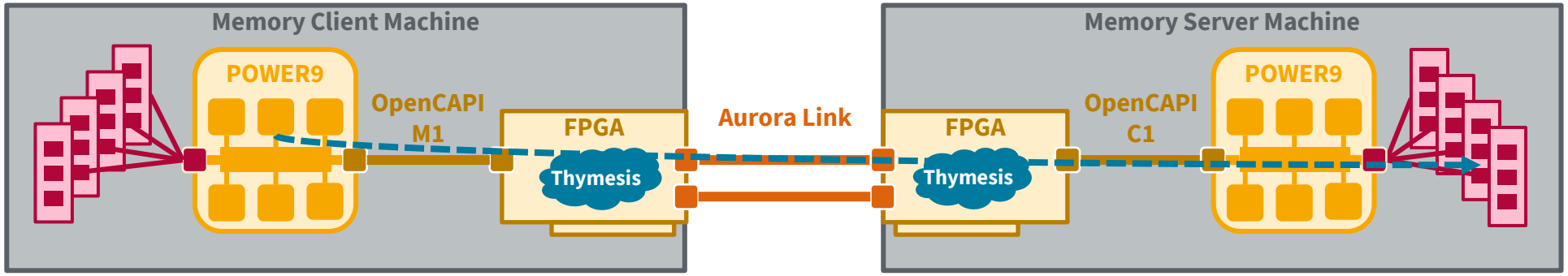
Figure 3: Sequential access bandwidth. An experiment showing the maximum possible bandwidth on local-socket DDR5 with 8-channels (a), and CXL memory (b), and remote-socket DDR5 with 1 channel (c). The grey dash line in (b) shows the theoretical max speed of DDR4-2666MT/s

Your comments or experiences?

ThymesisFlow @ HPI
Setup

- Hardware
  - 2x IBM **IC922** or IBM AC922 or Inspur FP5290G2
  - 2x AlphaData 9V3 FPGAs
  - 2x OpenCAPI cables (SlimSAS) 25 Gb/s x8
  - 2x 100 Gb/s network cabling



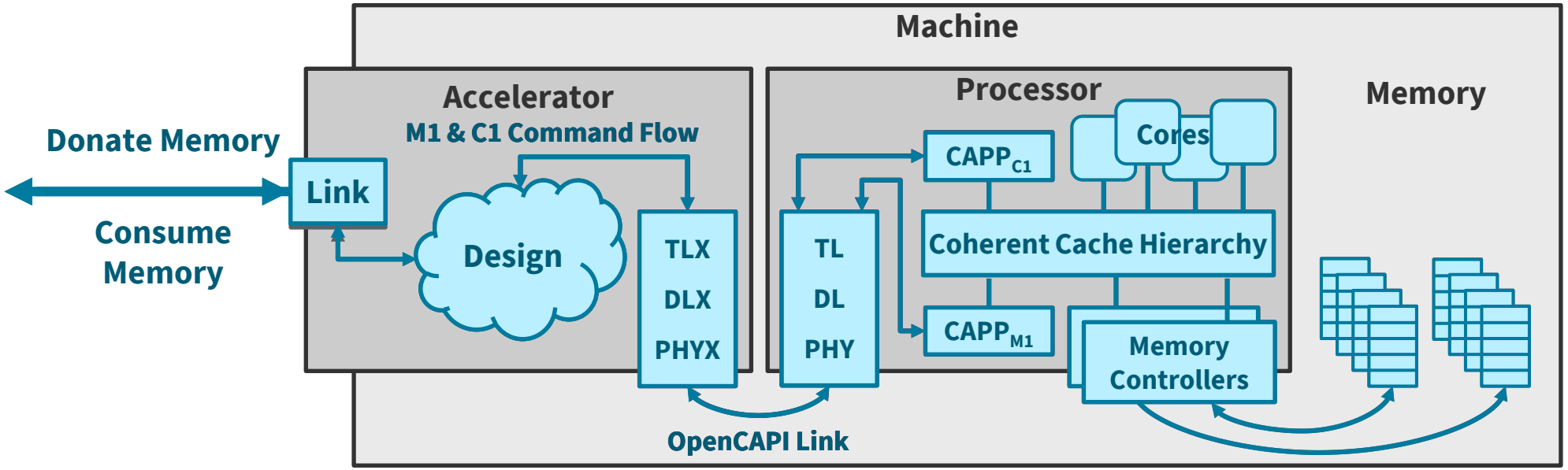


Disaggregated Memory

Wenzel (OSM HPI)

ThymesisFlow @ HPI
OpenCAPI 3.0 Protocol

- OpenCAPI 3.0 allows **transparent** remote memory access:
- Accelerator Cards support two orthogonal modes
  - **C1** - Accelerator acts as another **C**ore and accesses host memory using *virtual addresses* (like CXL.memory)
  - **M1** - Accelerator acts as another **M**emory controller and handles memory accesses by the host using *physical addresses* (like CXL.cache)



ThymesisFlow @ HPI
Architecture

- Accelerator and Host both post and complete commands:
  - C1-role initiates read or write commands to be handled by host system
  - ThymesisFlow forwards commands from M1 to C1 card via Aurora point-to-point link
  - M1-role handles read or write commands from host system
- Repository: https://github.com/OpenCAPI/ThymesisFlow
- Deep Dive (C. Pinto): https://www.youtube.com/watch?v=4N5G_Am6U-M

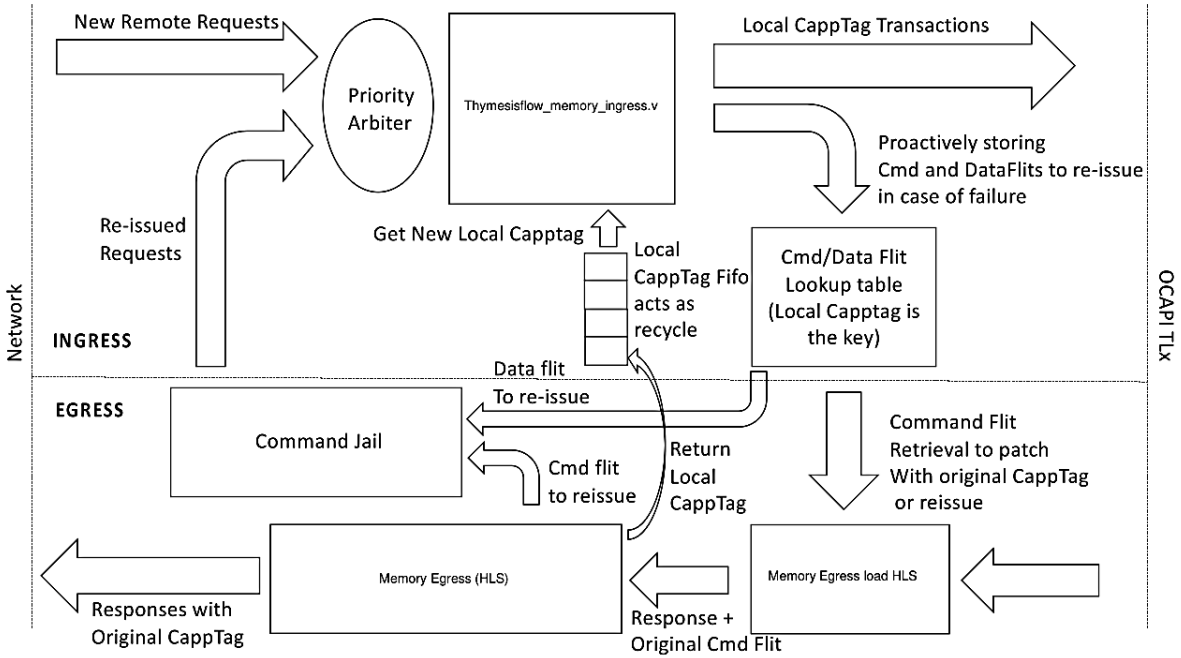


Figure 7: Memory Side ThymesisFlow Design

From:

Thymesis Flow Design and Deployment, Version 1.1, IBM Research Lab Dublin, 2020

ThymesisFlow @ HPI
Architecture

- **Memory Server**: C1 mode accelerator uses *virtual address* space of attached process
  - `thymesisf-agent` application allocates memory to serve and attaches to accelerator
- **Memory Client**: M1 mode accelerator appears at a fixed position in *physical address* space
  - Raw access via `mmap()`-able memory device at `/dev/mishmem-s<n>`
  - Registered as a CPU-less NUMA node → application can allocate remote memory through `libnuma`

Static Offsets

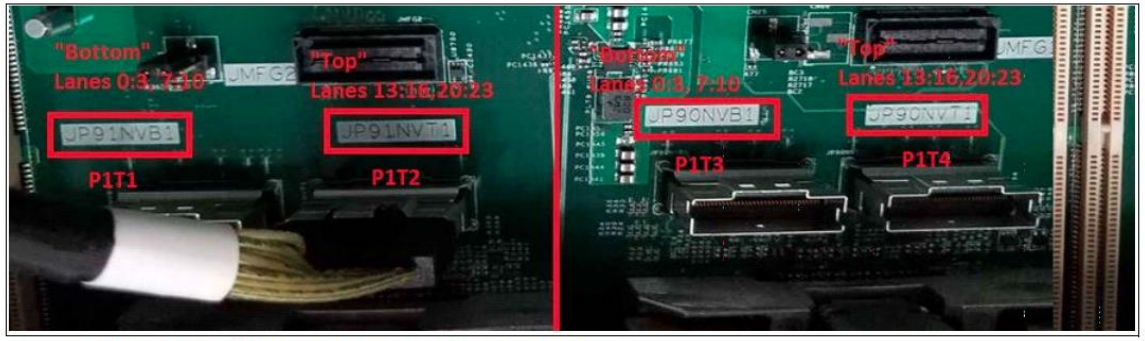
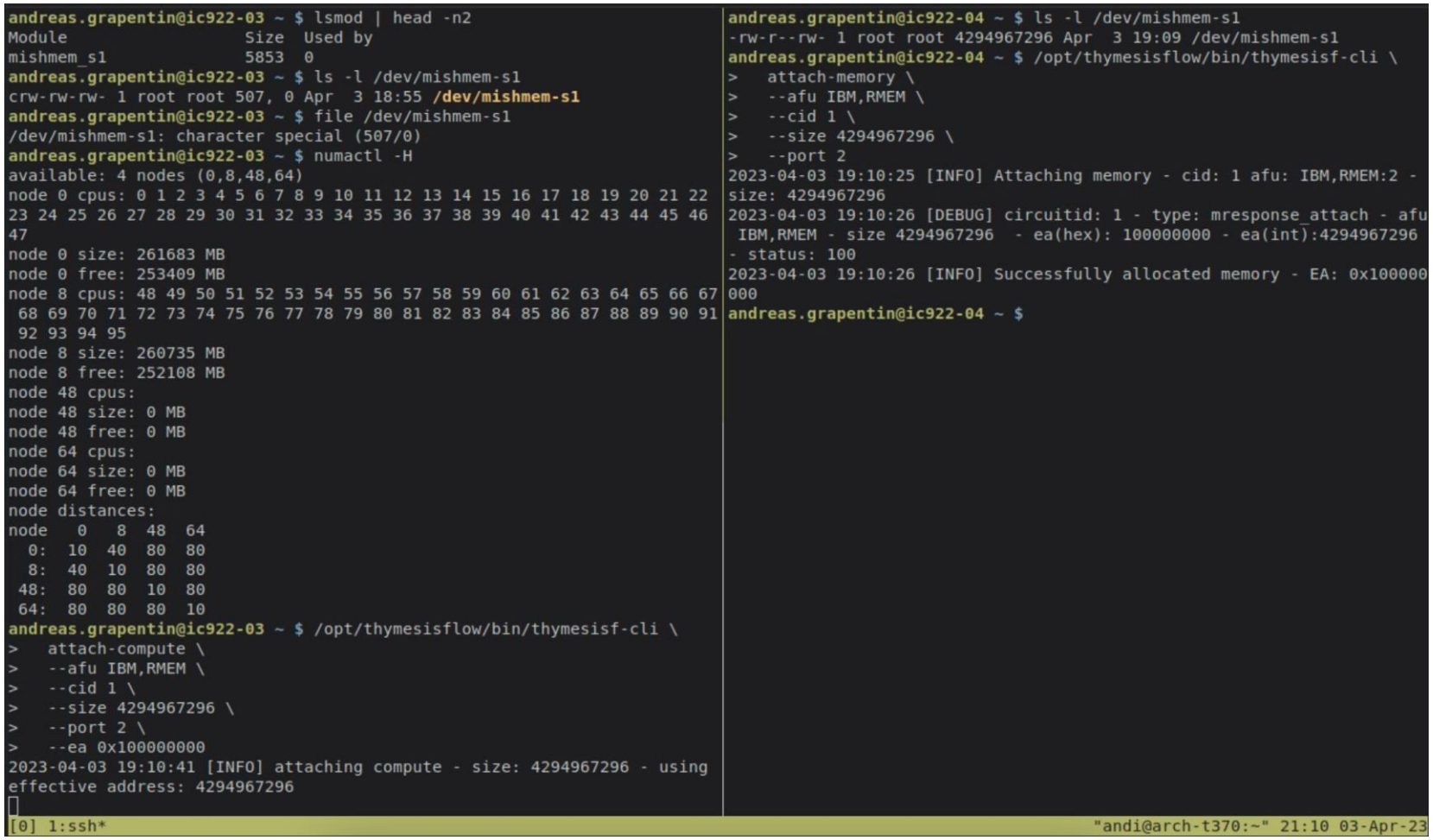


Figure 1-9 OpenCAPI 3.0 ports in the Power IC922 server

Processor 0: 0x2000000000000

Processor 1: 0x2200000000000

ThymesisFlow @ HPI
Demo



[STREAM] J. D. McCalpin. **"STREAM: Sustainable Memory Bandwidth in High Performance Computers."** Technical Report, University of Virginia, Charlottesville, **1991**.

[lmbench] L. McVoy, and C. Staelin. **"lmbench: Portable Tools for Performance Analysis."** USENIX annual technical conference, **1996**.

ThymesisFlow @ HPI
Collaboration

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Interested to experiment with or collaborate on extensions to this ThymesisFlow setup?

Message us!

{lukas.wenzel,felix.eberhardt,andreas.grapentin}@hpi.de