MemCon 2024

Exploring CXL Use Cases and the Future of Disaggregated Heterogeneous Memory Architecture

Ping Zhou

Senior Researcher/Architect, ByteDance Inc.

March 2024

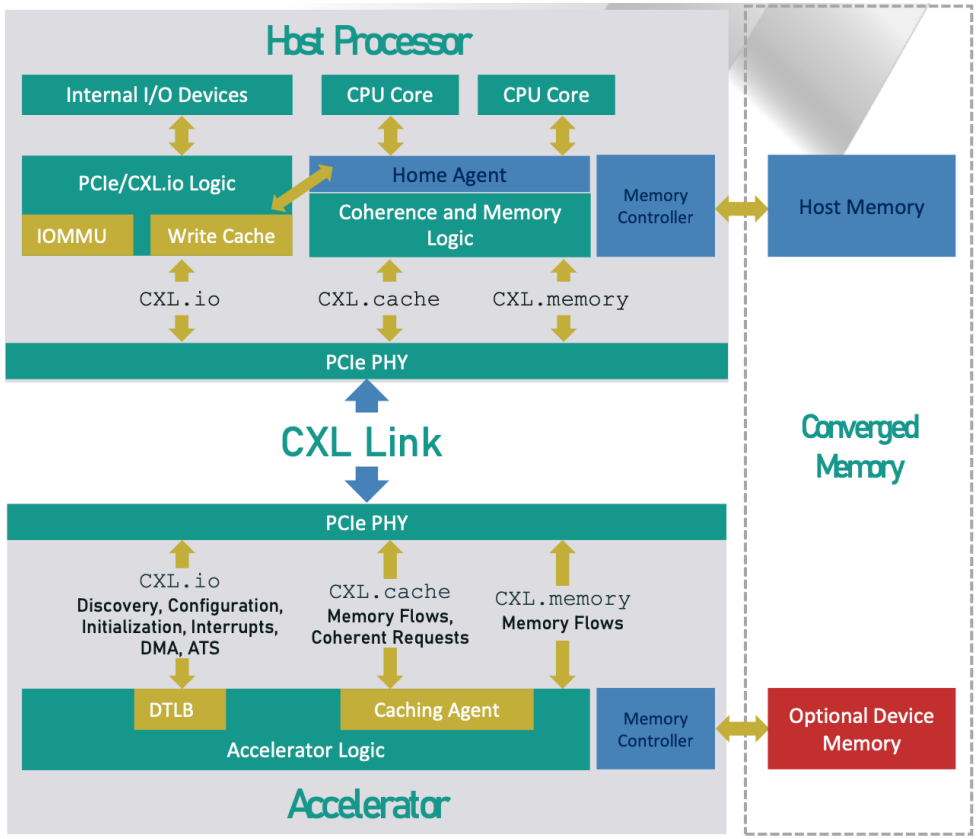
ByteDance 字节跳动

Agenda

- Background and Motivation
- Understanding the Performance Characteristics of CXL Memory
- Exploring Use Cases with CXL Memory
- Architecting Next-Gen Memory Systems for AI/ML

Background of CXL

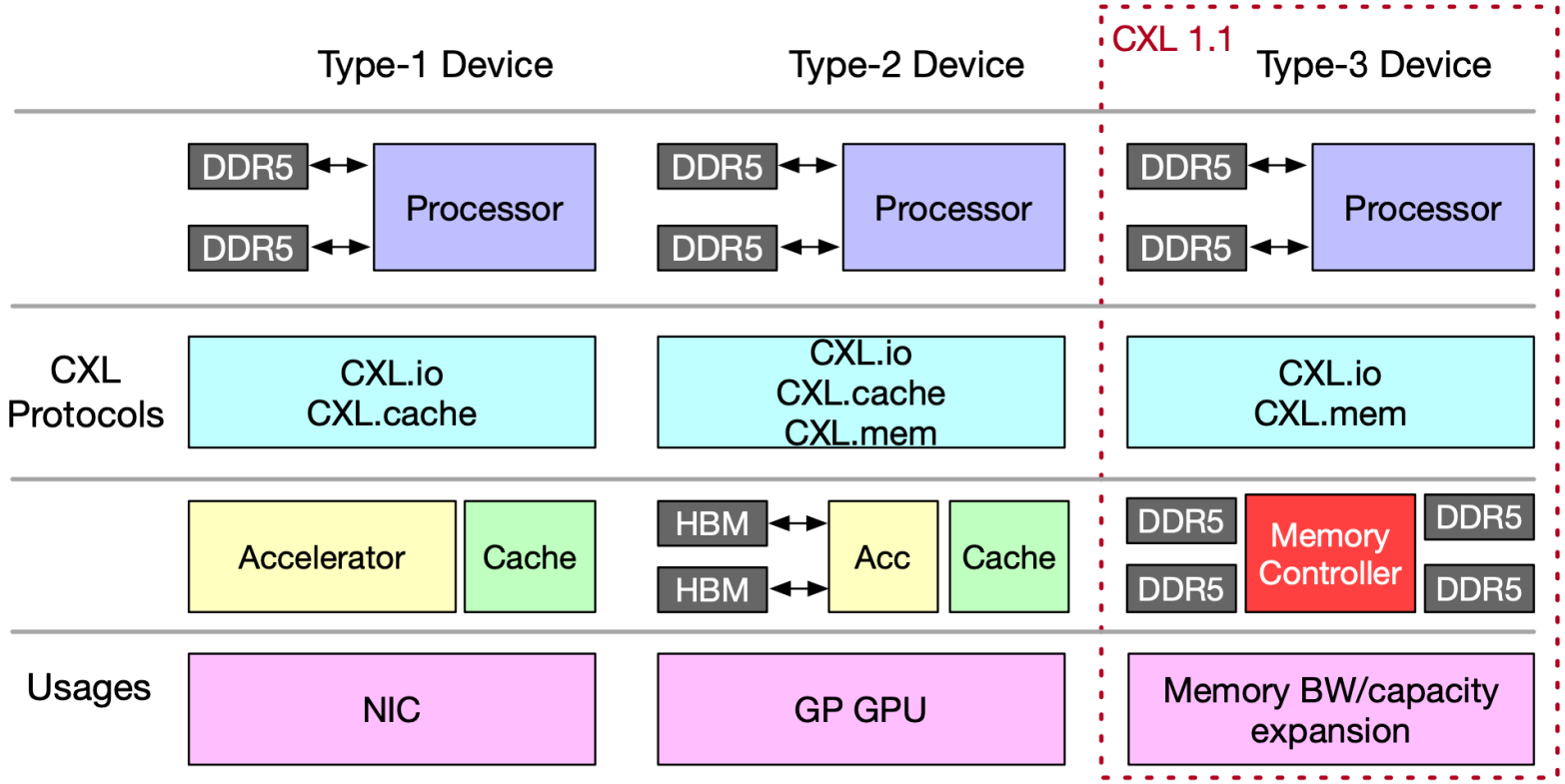
- High-speed, efficient interconnect between CPU, memory and accelerators
- Built on top of PCIe infrastructure
- Open industry standard
  - CXL 1.1: memory expansion
  - CXL 2.0: CXL switch, memory pooling
  - CXL 3.0/3.1: memory sharing



Compute Express Link™ (CXL™): An Open Industry Standard for Composable Computing (FMS 2023 tutorial)

Background of CXL

- CXL Type-3 Device: Memory expansion in CXL 1.1

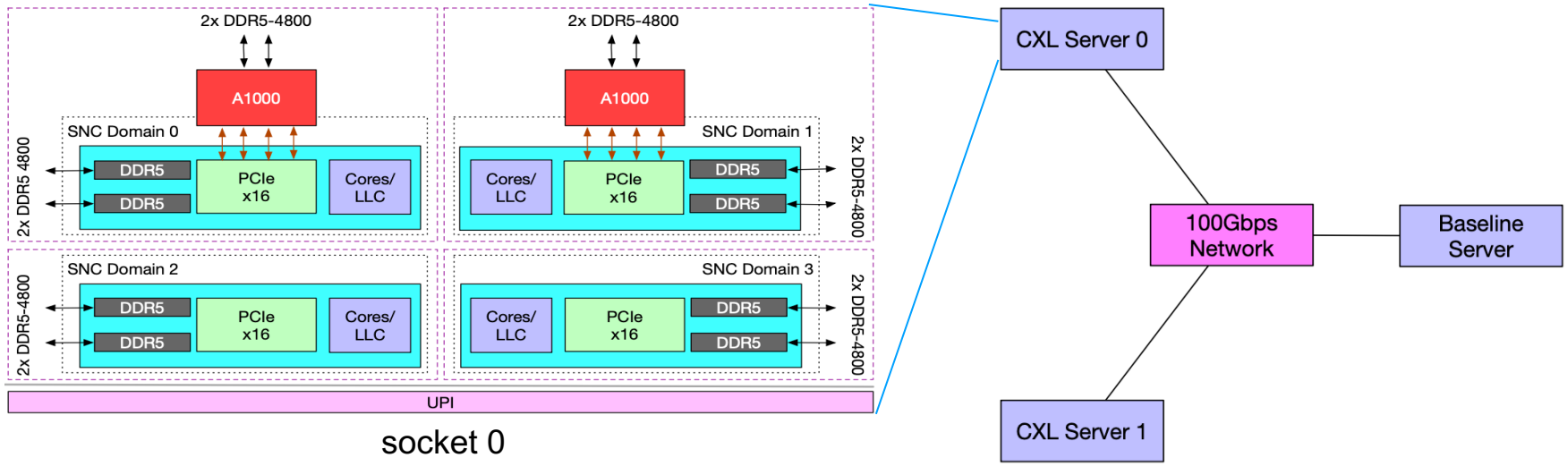


Motivation

- Evaluate CXL memory expansion in real-world environment
- Understand performance characteristics of ASIC-based CXL hardware
- Explore potential use cases and understand how they’ll benefit from CXL memory

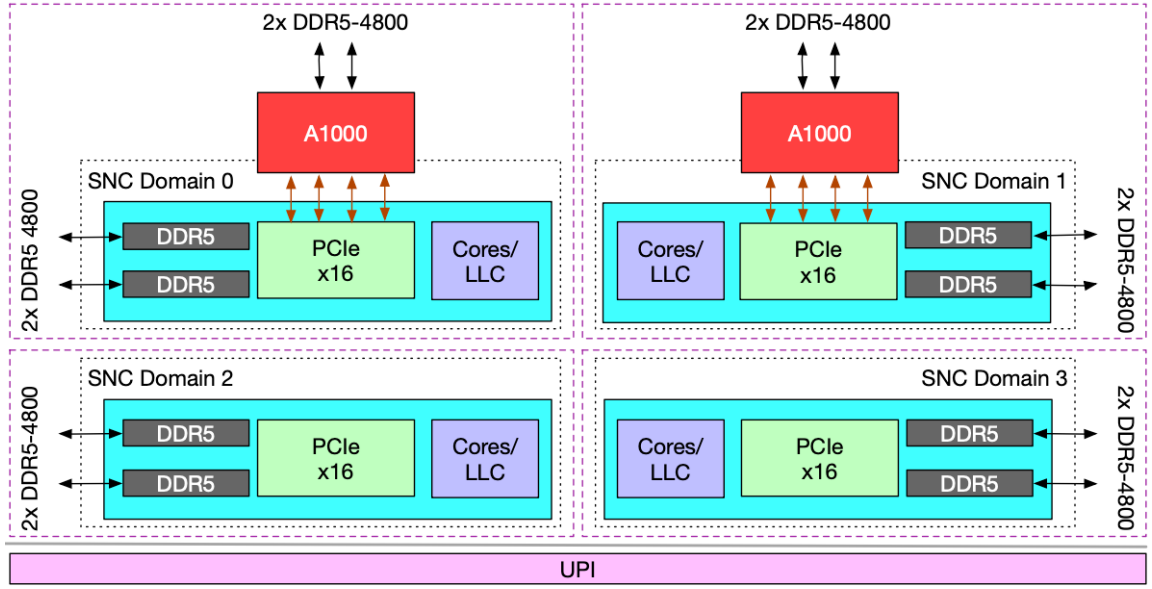
Experimental Setup (1)

- Experimental setup
  - Server: Intel Xeon 4th Generation, dual socket, 1TB DDR5-4800 memory
  - CXL Memory Expanders: Astera Labs A1000, 512GB DDR5-4800 in total



Experimental Setup (2)

- Sub-NUMA Clustering (SNC) enabled for microbenchmark tests
  - Decomposes a single NUMA node into multiple sub-nodes
  - Allowing fair comparison between local memory vs. CXL memory in microbenchmark tests
- Memory Latency Checker
- Various read/write ratio, 64B access, 16 threads



Performance Characterization (1)

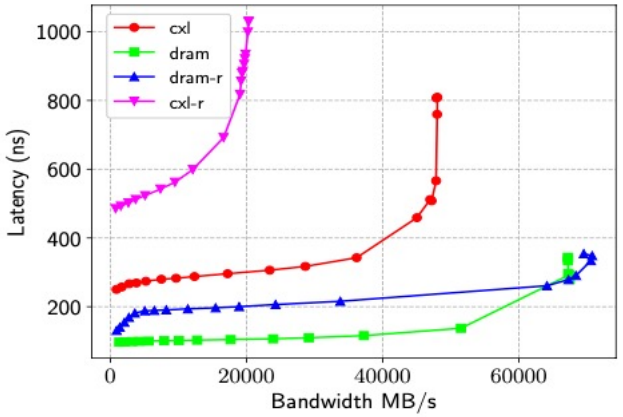


Figure: Read-only

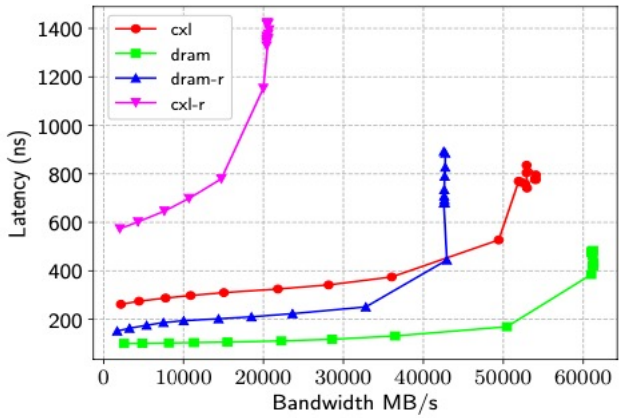


Figure: Read:Write = 2:1

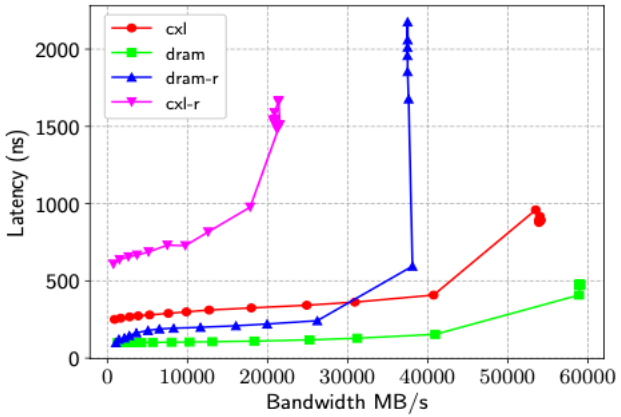


Figure: Read:Write = 1:1

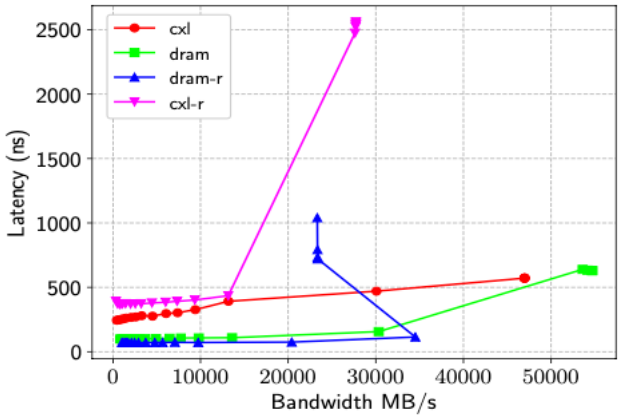


Figure: Write-only

- CXL memory on socket 0
- Local DRAM/CXL:
  Data on socket 0, access from socket 0
- Remote DRAM/CXL:
  Data on socket 0, access from socket 1
- 2-ch CXL vs 2-ch DRAM (using SNC)

Performance Characterization (2)

Latency (read-only, no load)

	Local	Remote
DRAM	97ns	130ns
CXL	251ns	485ns

Maximum bandwidth observed (2-channels)

	Local (R)	Local (W)	Remote (R)	Remote (W)
DRAM	70.8GB/s	54GB/s	70GB/s	35GB/s
CXL	48GB/s	47GB/s	20GB/s	27GB/s

Key Observations

- Latency grows exponentially when bandwidth is highly utilized (“loaded latency”)
  - Mainly due to “queuing delay” on memory controller (“A Case for CXL-Centric Server Processors”)
  - Increase in total bandwidth could result in lower access latency
- Significant performance drop in remote CXL access
  - Bandwidth halved compared to local access
  - Possibly due to hardware limitation from the CPU used in our test

Exploring Potential Use Cases

- Capacity expansion

- Bandwidth expansion

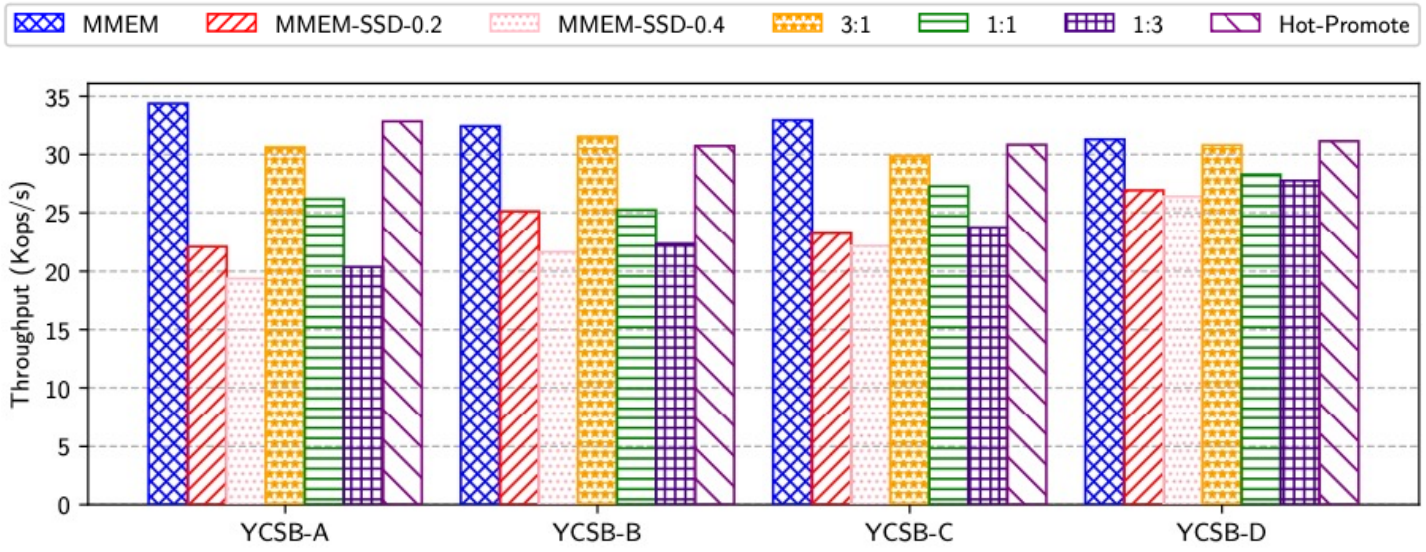
Use Case: In Memory Database (1)

- KeyDB
  - Open source derivative of Redis, with spill-to-SSD feature

Configuration	Description
`MMEM`	Entire working set in main memory.
`MMEM-SSD-0.2`	20% of the working set is spilled to SSD.
`MMEM-SSD-0.4`	40% of the working set is spilled to SSD.
`3:1`	Entire working set in memory (75% MMEM + 25% CXL, 3:1 interleaved).
`1:1`	Entire working set in memory (50% MMEM + 50% CXL, 1:1 interleaved).
`1:3`	Entire working set in memory (25% MMEM + 75% CXL, 1:3 interleaved).
`Hot-Promote`	Entire working set in memory (50% MMEM + 50% CXL), with hot page promotion kernel patches discussed in §2.

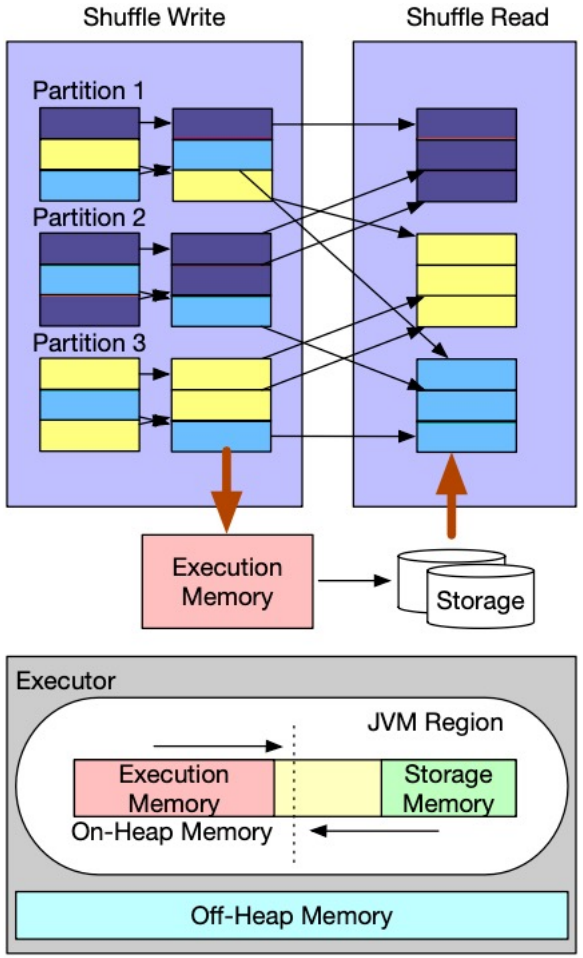
Use Case: In Memory Database (2)

- Naive interleaving results in significant performance drop
- CXL with Hot-Promote performs nearly as well as local DRAM



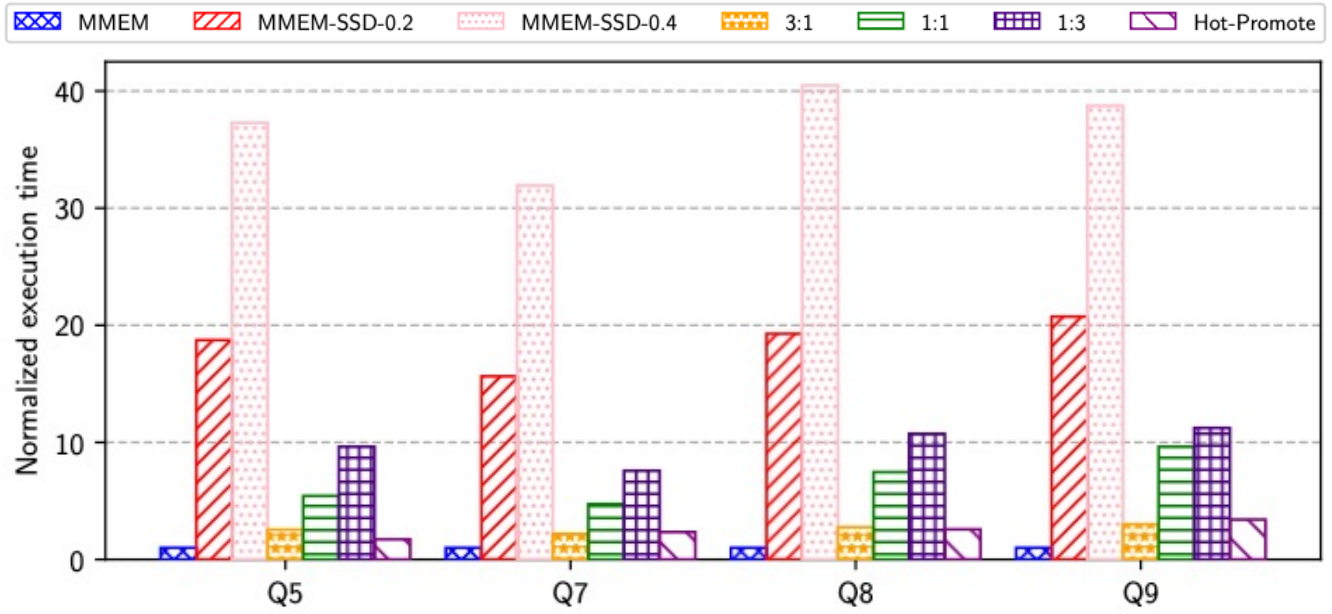
Use Case: Data Analytics (1)

- Spark SQL
- Shuffling could be a dominant factor of performance
- Memory usage could grow beyond capacity threshold – data spilled to disk



Use Case: Data Analytics (2)

- Spill-to-SSD performs much worse than DRAM or CXL.
- With Hot-Promote, MMEM+CXL (1:1) outperforms 3:1 naïve interleaving (~34% slower than 100% MMEM).



Use Case: IaaS Computing

- CPU-to-Memory ratio of instances
- Rapidly growing # of cores
  - Hard to keep up the ratio
- Wasting/giving away vCPUs due to inadequate memory capacity
  - Lose the potential revenue

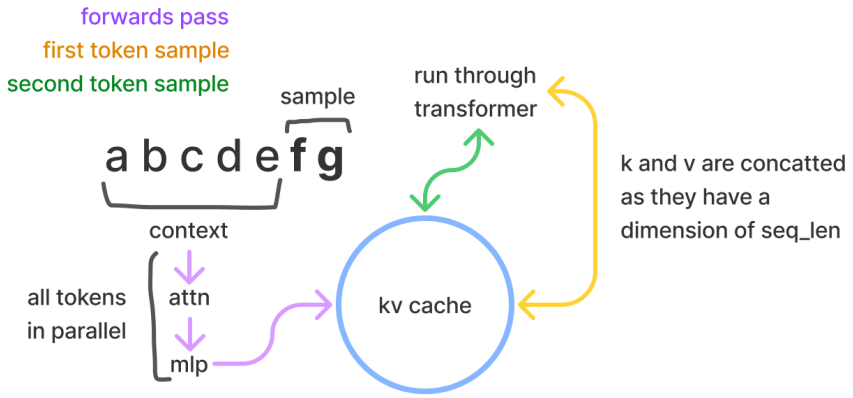
Instance Size	vCPU	Memory (GiB)
m7i-flex.large	2	8
m7i-flex.xlarge	4	16
m7i-flex.2xlarge	8	32
m7i-flex.4xlarge	16	64
m7i-flex.8xlarge	32	128

Instance Size	vCPU	Memory (GiB)
m7a.medium	1	4
m7a.large	2	8
m7a.xlarge	4	16
m7a.2xlarge	8	32
m7a.4xlarge	16	64
m7a.8xlarge	32	128
m7a.12xlarge	48	192
m7a.16xlarge	64	256
m7a.24xlarge	96	384
m7a.32xlarge	128	512
m7a.48xlarge	192	768
m7a.metal-48xl	192	768

https://aws.amazon.com/ec2/instance-types/m7i/
https://aws.amazon.com/ec2/instance-types/m7a/

Use Case: KV Cache in LLM Inference (1)

- KV Cache: commonly used for reducing LLM inference costs
- Requires large capacity and bandwidth
  - Size grows with sequence length and batch size
  - Each request has its own KV cache

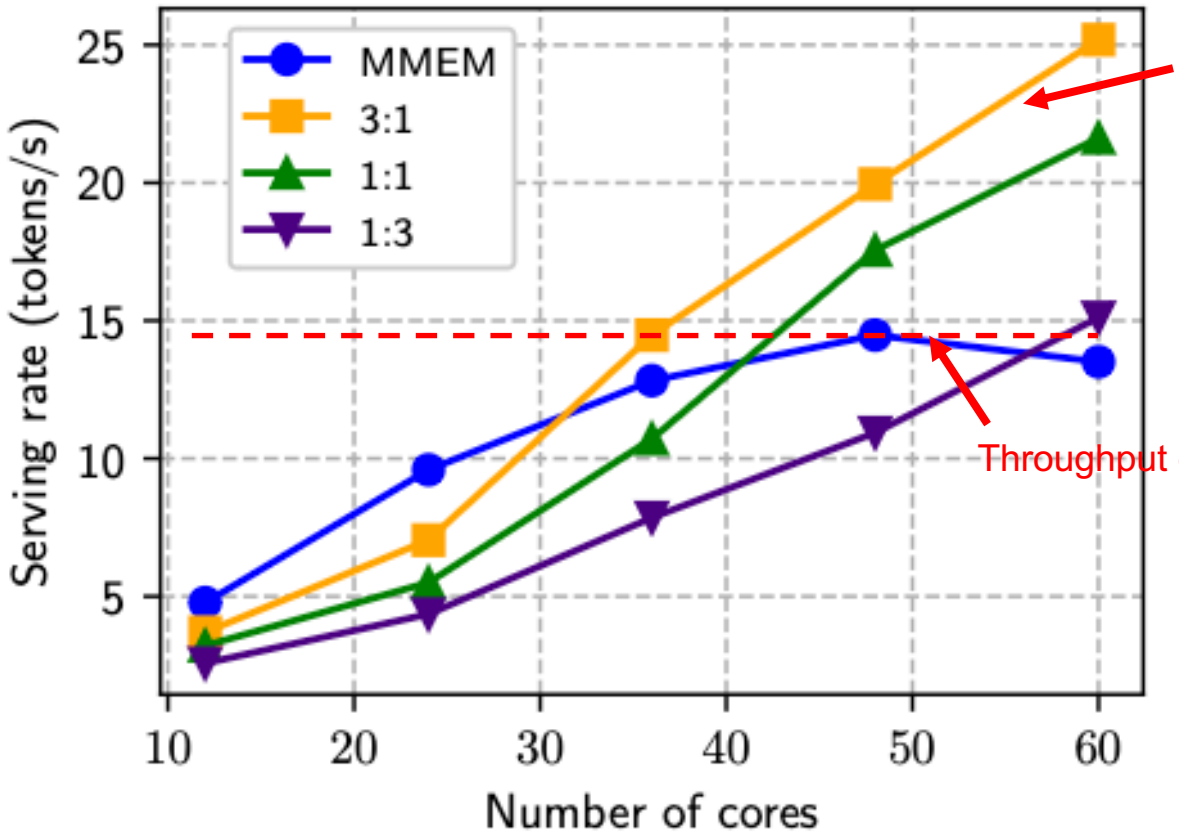


https://kipp.ly/transformer-inference-arithmetic/

"For a 500B+ model with multihead attention, the attention KV cache grows large: for batch size 512 and context length 2048, the KV cache totals 3TB, which is 3 times the size of the model's parameters."

-- [R. Pope et al., Efficiently Scaling Transformer Inference, MlSys 2023]

Use Case: KV Cache in LLM Inference (2)



Throughput further scales with extra bandwidth from CXL

Throughput capped due to bandwidth limitation

Insights (1)

- Diversified application characteristics
  - latency sensitive
  - capacity sensitive
  - bandwidth sensitive
- Non-linear correlation between latency and bandwidth
  - ”Idle latency” could be misleading, you often get latencies under load
  - Increase total BW → reduce BW utilization → lower latency
  - The extra bandwidth from CXL could help latency sensitive application too!

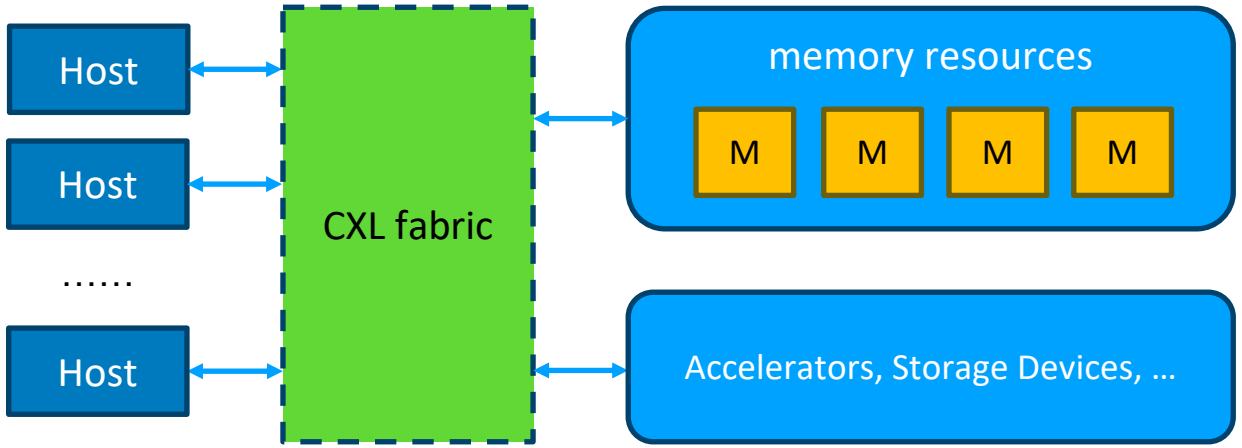
Insights (2)

- CXL decouples memory devices from CPU
  - CPU accesses CXL memories in a unified way
  - DRAM, Persistent Memory, SSD, …
- Could be a key booster of memory technology innovations
- Great opportunity for new memory technologies!

Insights (3)

Towards a Disaggregated Heterogeneous Memory Architecture

- Unified memory address space
  - Not just in general computing
  - Nvidia GH200
  - Apple M2 Ultra
  - Google TPU
- Increased heterogeneity
  - Local DRAM
  - Local CXL memory expansion
  - Remote CXL memory
  - Different # of hops
  - Different memory media



Challenges

- Scaling beyond a single rack
- Fault tolerance
- Software stack
- Costs

Conclusion

- Promising technology for scaling memory beyond individual machines
- Interesting observations from performance charaterizations
- Potential use cases of capacity & bandwidth expansion
- Evolving towards heterogeneous disaggregated memory architecture
- Many new opportunies and challenges ahead

More details in our upcoming EuroSys 2024 paper “Exploring Performance and Cost Optimization with ASIC-Based CXL Memory”