

CXL Memory Use Cases: Insights into Expansion and Pooling

PRESENTER

Minseong Kim



TABLE OF CONTENTS

Why do we need CXL Memory?

- Memory Capacity Requirement
- Memory Capacity Gap in AI applications & GPUs

CXL Usage Model Tree

- Single Server - Bandwidth Expansion and Capacity Expansion
- Multiple Servers - Pooling & Sharing

CXL Memory Usage Model Survey

- Bandwidth Expansion / Capacity Expansion / Tiering
- Memory Pooling & Sharing

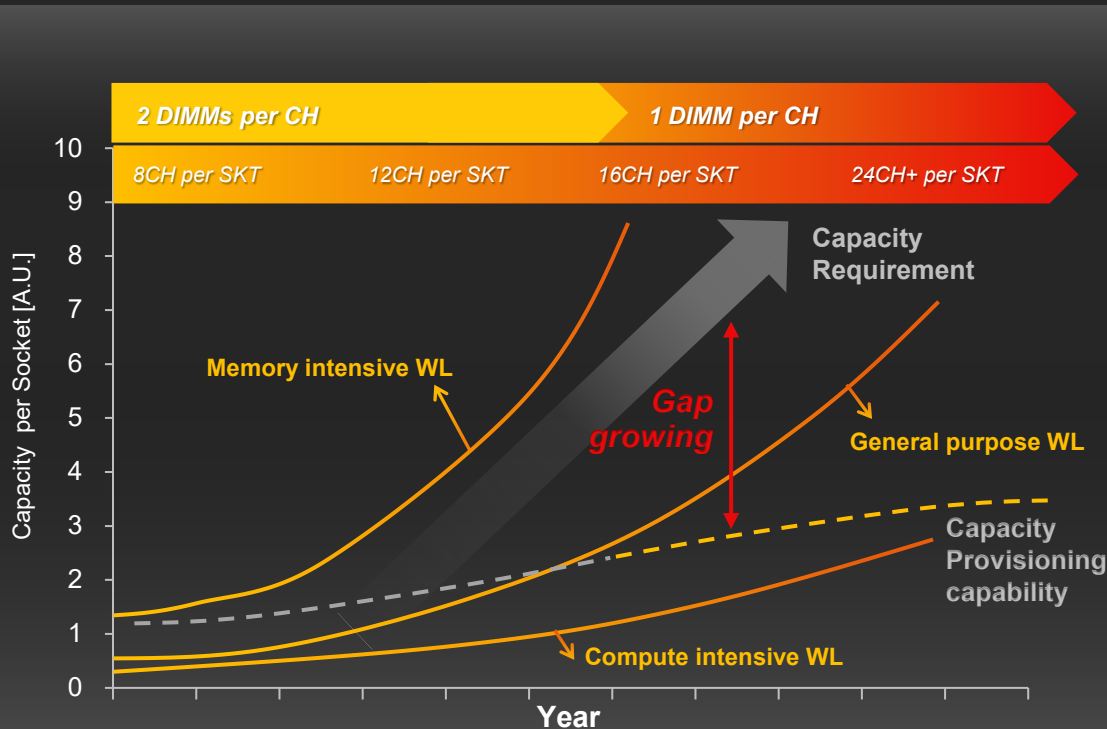
Experimental Results

- Bandwidth Expansion Case – Single Server with LLM Inference
- Capacity Expansion Case – Single Server with Redis IMDB
- Memory Pooling Case – Multiple Servers with Redis IMDB

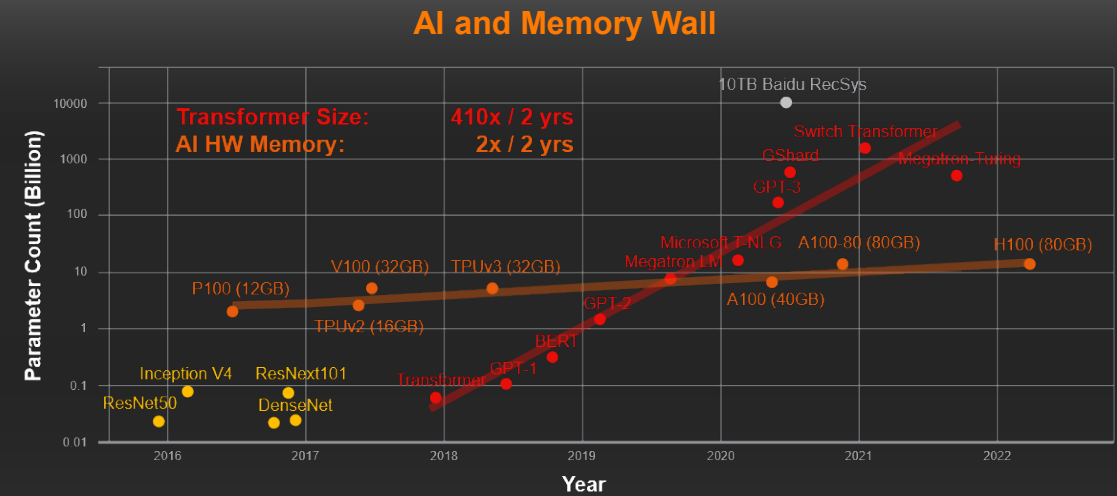
Why do we need CXL Memory?

Memory wall exists relative to CPU Core Count, and memory wall also exists in GPU-based AI Memory System
Need to overcome the difficulty of high-capacity scaling with pooling/switching-based scale-up/scale-out

Memory Capacity Requirement (CPU)



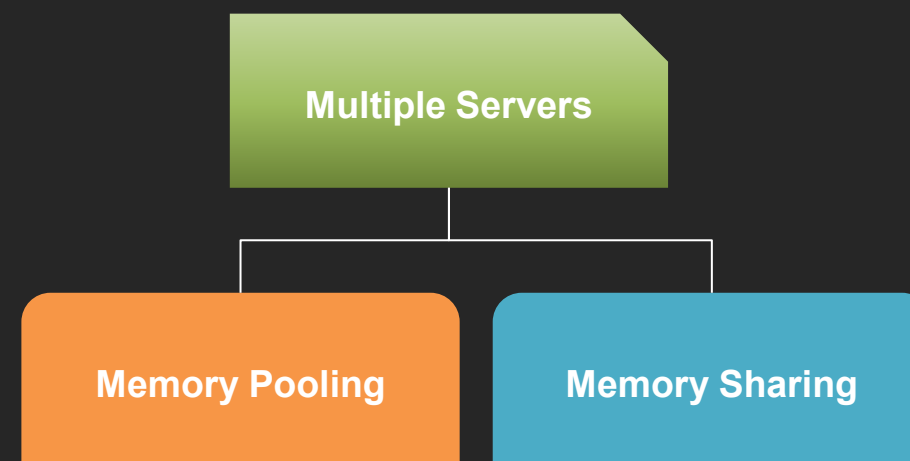
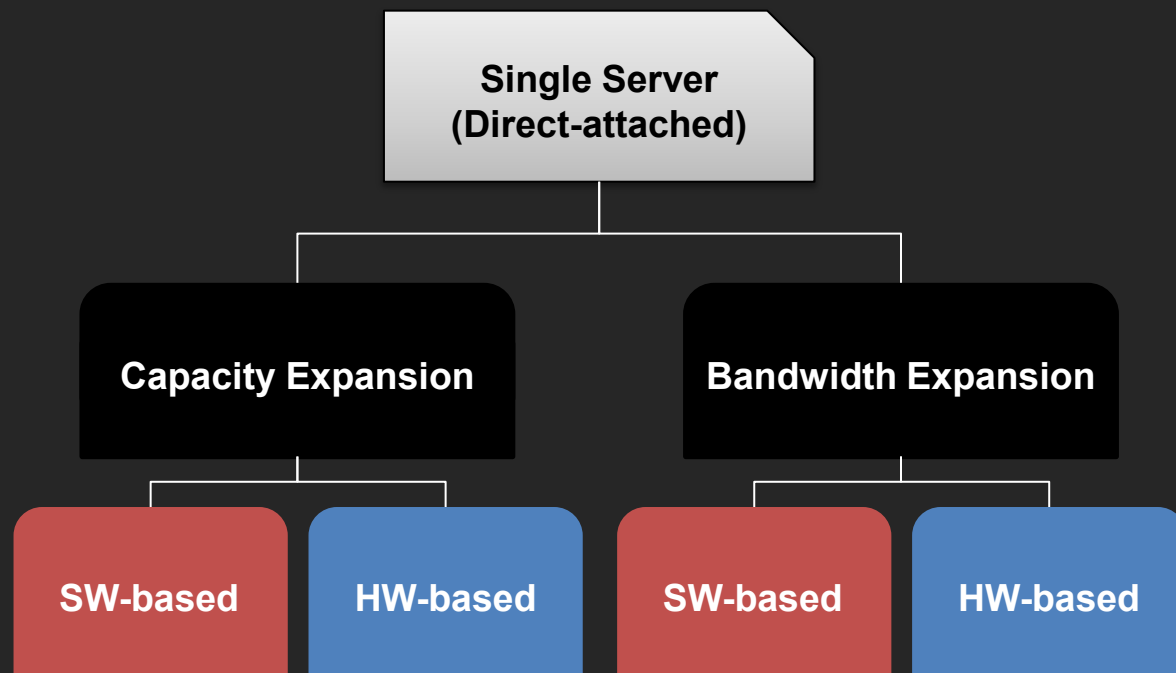
Memory Capacity Gap in AI applications & GPUs



Application fragmentation makes it difficult to know sweet spot capacity, but TB to 10TB+ of high capacity memory is needed within TCO budget








Reference: AI and Memory Wall, A. Gholami et al, IEEE Micro

CXL Memory Usage Model Tree



✓ We explore primary approaches to utilizing the CXL Memory Module.

CXL Memory Usage Model Survey

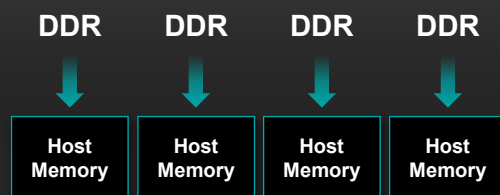
Category	Capacity Expansion	Bandwidth Expansion	Tiered Memory	Pooling/Sharing	Notes
	In-memory Database Redis + YCSB	LLM Inference (Llama.cpp + Llama3)	1) In-memory DB, 2) LLM Inference (Redis + memtier_bench, Llama.cpp + Llama3)	In-memory DB In-memory Analytics (CloudSuite) Ray Shuffle	
	DLRM inference/training ¹⁾	DLRM inference/training ¹⁾		In-memory DB ²⁾ (TPC-DS on SAP HANA)	IMDG Databases & Caches AI/ML Workloads Financial Services
 ³⁾ 	In-memory DB (MSSQL + TPC-H)	HPC (CloverLeaf)	Apache Spark Based Machine Learning (SVM) (Big Data Workload)		
⁴⁾ 	1) PostgreSQL + TPC-H 2) RocksDB + db_bench 3) RAG Pipeline				
			Weaviate Vector DB on gist dataset (ANNS) ⁵⁾		
	Azure Service (PoC) Various Benchmark			Azure Service (PoC) Various Benchmark	Pond (ASPLOS 23) Octopus (2025)

✓ *We focus performance analysis of IMDB and LLM Inference across various scenarios.*

Bandwidth Expansion Case – Single Server

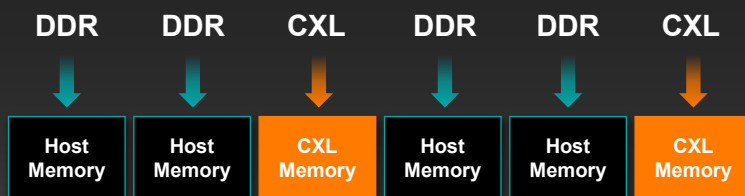
We ask a question to AI model, CMM+DDR shows 30% better performance of system.
Next CMM has 2x better BW compared to AI model result of current product.

<Native DDR Only>



Utilizing Host DRAM 2-channel only

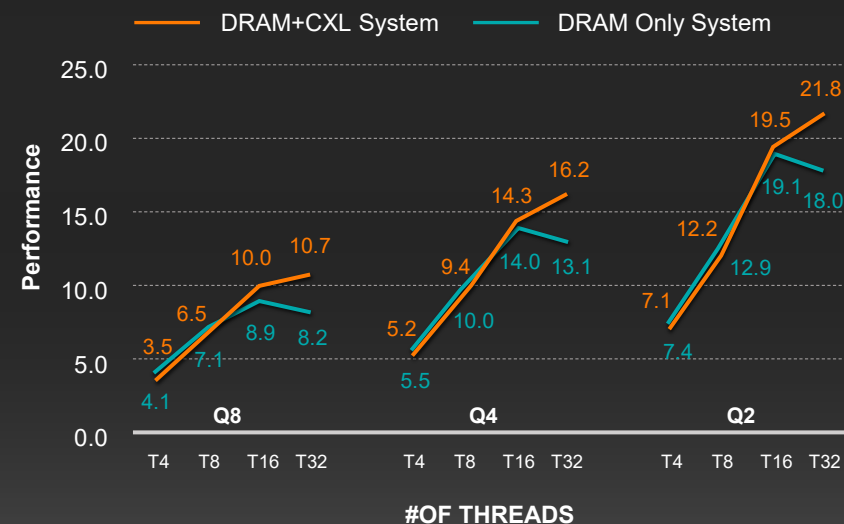
<Native DDR+CXL>



Utilizing Host DRAM 2-channel + CXL 1-channel

Enhanced System Performance

Throughput Comparison of AI Inference Workload

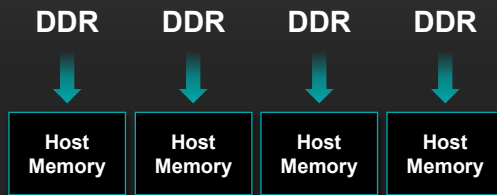


Capacity Expansion Case – Single Server

We use the YCSB benchmark for the Redis in-memory database.

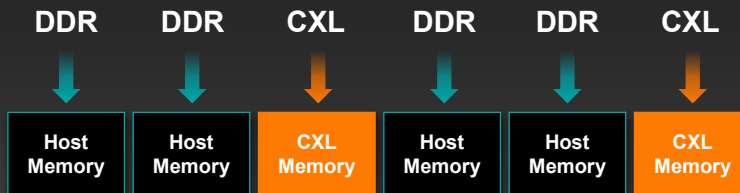
CMM allows us to load a database up to the additional CMM capacity on a single server without throughput drop.

<Native DDR Only>



Utilizing Host DRAM 2-channel only

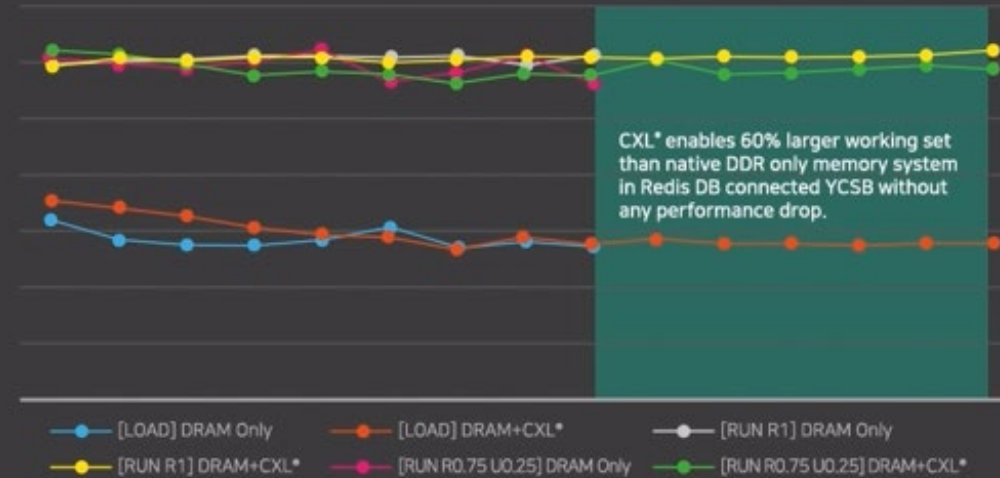
<Native DDR+CXL>



Utilizing Host DRAM 2-channel + CXL 1-channel

No Significant Performance Drop + Load More Database

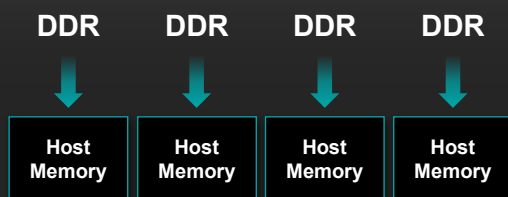
Capacity Expansion Case (Redis + YCSB)



Memory Pooling Case – Multiple Servers

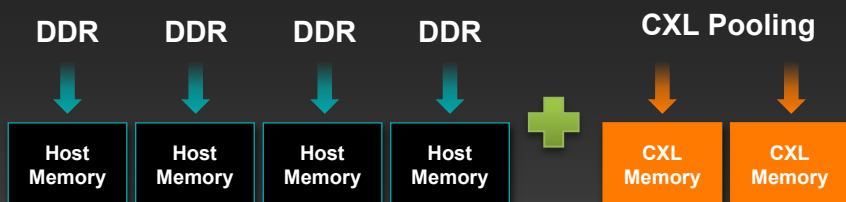
We evaluated the performance of the Redis using the YCSB benchmark with CXL Memory Pooling. We found no significant performance difference. This confirms that IMDB is a suitable use case for CXL memory pooling.

<Native DDR Only>



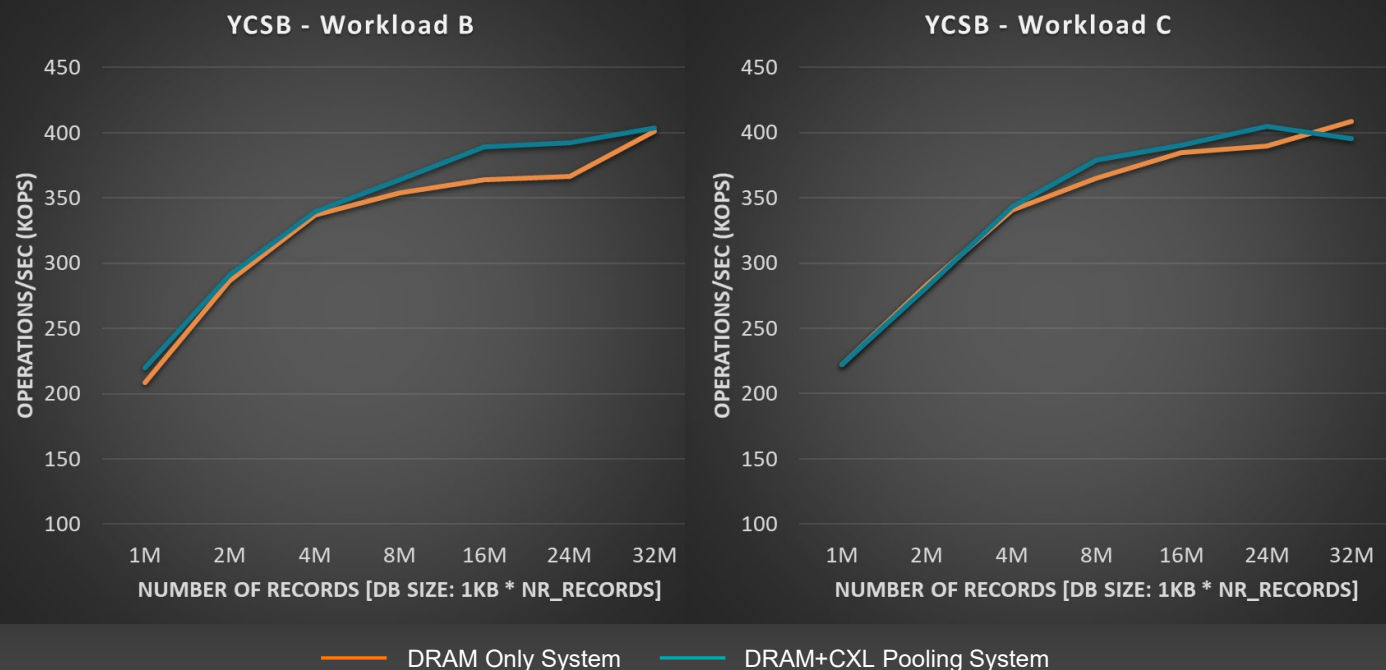
Utilizing Host DRAM 2-channel only

<Native DDR+ CXL Pooling>



Utilizing Host DRAM 2-channel + CXL Pooling

No Significant Performance Drop in CXL Pooling System



Questions?