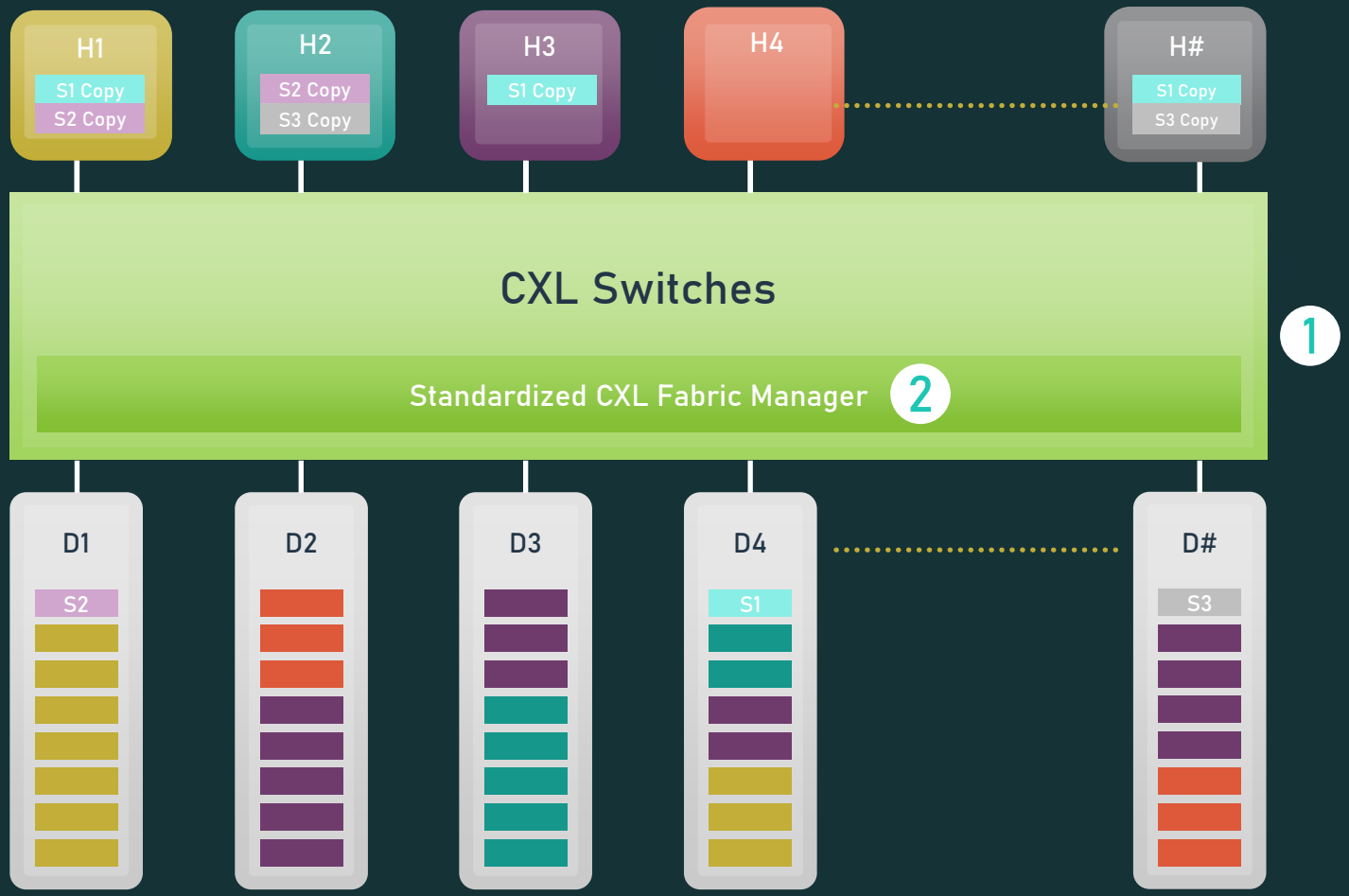
Beyond the Memory Wall: How CXL Memory Pooling and Sharing Are Transforming AI Inference

CXL Consortium

Meet our Panelists

- Moderator: Anil Godbole, CXL MWG Chair (Intel)
- Panelists:
  - Sandeep Dattaprasad (Astera Labs)
  - Sumit Puri (Liqid)
  - Luis Ancajas (Micron)
  - Ronen Hyatt (UnifabriX)

CXL Memory Pooling & Sharing



Benefits:

- Low-latency data sharing
  - No shuffling of data
- Total Cost of Ownership

Applications:

- Distributed databases
- RAG
- Graph database
- KVCache storage
- Algo Trading

CXL Integrators List

Device Type		Feature Set		Spec Revision		Function	
Type 1	7	CXL Core 1.1	41	CXL 1.1	30	Accelerator	3
Type 2	8	CXL Core 2.0	26	CXL 2.0	30	Host	11
Type 3	57					IP	14
						MEM Expander	35

Show 15 entries

Search:

Company Name	Product Name	Device ID	Device Type	Feature Set	Spec Revision	PHY Speed	Max Lane	Form Factor	Function	Compliance Event (CTE) Approved
Micron Technology, Inc.	CZ122	6404	Type 3	CXL Core 1.1, CXL Core 2.0, MEM 2.0	CXL 2.0	32GT/s	x8	EDSFF	MEM Expander	CTE 006
Micron Technology, Inc.	Micron Rev B	6400	Type 3	CXL Core 1.1	CXL 1.1	32GT/s	x8	EDSFF	MEM Expander	CTE 001
Micron Technology, Inc.	Micron Rev A	6400	Type 3	CXL Core 1.1	CXL 1.1	32GT/s	x8	EDSFF	MEM Expander	CTE 001
Montage Technology Co., Ltd	Memory Expander	C001	Type 3	CXL Core 2.0	CXL 2.0	32GT/s	x8	CEM	MEM Expander	CTE 006
Montage Technology Co., Ltd	Memory Expander	C002	Type 3	CXL Core 2.0	CXL 2.0	32GT/s	x8	CEM	MEM Expander	CTE 006
Montage Technology Co., Ltd	CXL Memory Expander Controller (MXC)	0xC001	Type 3	CXL Core 1.1	CXL 1.1	32GT/s	x8	CEM	MEM Expander	CTE 001
Phison Electronics Corporation	PS7201 PCIe 5.0 Retimer	PS7201	Type 3	CXL Core 1.1, CXL Core 2.0	CXL 2.0	32GT/s	x16	CEM	Retimer	CTE 007

OEMs offering CXL-capable servers:

- US / EMEA
  - Dell
  - HPE
  - Lenovo
  - Supermicro
- APAC
  - Advantech
  - Giga
  - Quanta
  - AIC

Scan the QR code to view the Integrators List

CXL Deployment Momentum

Azure delivers the first cloud VM with Intel Xeon 6 and CXL memory - now in Private Preview

Elyse_Ge_Hylander MICROSOFT
Nov 18, 2025

CXL on Azure

Meta fights soaring hardware costs by reusing old DDR4 server memory in new DDR5-only servers — custom CXL 2.0 chip marries legacy DDR4-2400 with cutting-edge DDR5-6400

News By Anton Shilov | Published June 30, 2026

CXL memory expanders give new life to DDR4 memory.

(Image credit: cyberchief/Reddit)

Alibaba Cloud Officially Announces New Positioning as a 'Full-Stack Artificial Intelligence Service Provider'; What Innovations Does the World's First CXL Database Server Bring?

cls.cn · Sep 29, 2025 04:10

US BABA -0.14%

① Alibaba Cloud launched the world's first dedicated PolarDB database server based on CXL2.0 Switch technology. In addition to the large model "7 continuous transmission" and the full AI infrastructure upgrade, it also emphasizes that data processing is the focus of AI layout; ② Cloud native databases have officially entered the "computing, memory, and storage" phase, and CXL technology provides the foundation for cloud databases to get rid of the limitations of stand-alone memory.

KV Cache Acceleration with CXL Memory Pooling

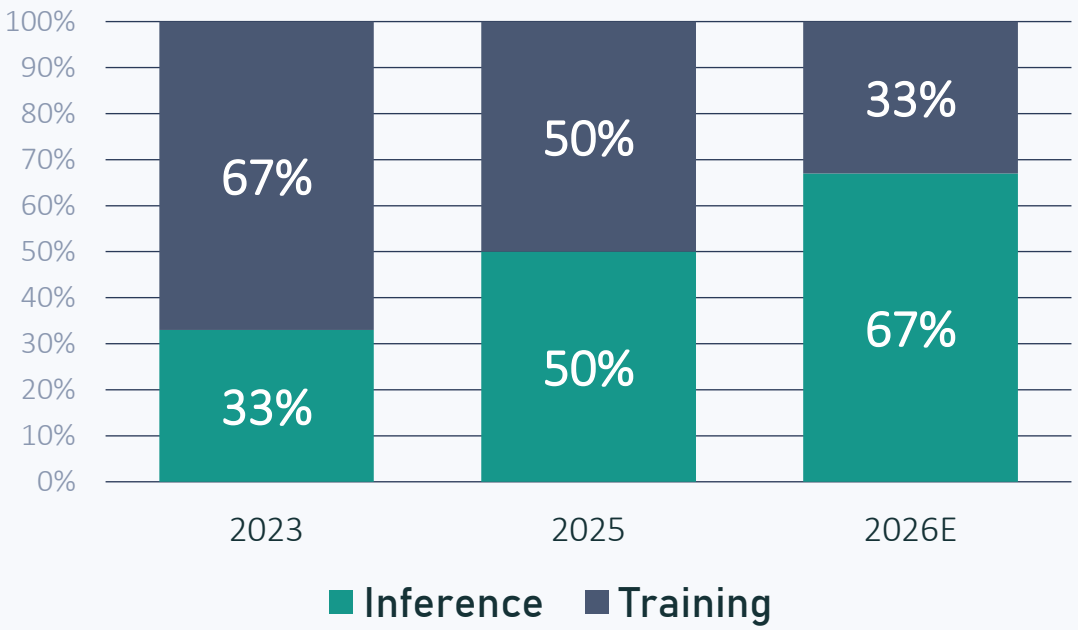
Sumit Puri (Liqid)

Industry Shift from Training to Inference

THE SHIFT TO INFERENCE

Using AI will outweigh building AI

How AI compute gets used — training new models vs. inference (answering users), %



Source: Deloitte 2026 TMT Predictions.

WHY IT MATTERS

The game completely changes

BUILDING AI (TRAINING)	USING AI (INFERENCE)
Happens in bursts	Runs 24/7, non-stop
One-time investment	Ongoing cost, tied to revenue
Some waste is tolerable	Every idle chip is lost money
Speed of buildout wins	Cost per answer wins
Bigger is better	Closer to the user is better

New Industry Standards:

Tokens per Dollar – Tokens per Watt – Tokens per Second

The Problem: High Cost. Low Performance. Poor Utilization.

40–60%

GPU CYCLES WASTED

GPUs sit unused between workload peaks — wasted capital at $2-4/hour/GPU in stranded capex

$450B

AI INFRA CAPEX

Global AI infrastructure spend continues to surge with AI demand

3–5x

MEMORY BOTTLENECK

AI inference models demand 3–5x more DRAM than servers natively support. Difficult to solve with more chips.

Billions in GPU and DRAM are stranded — physically locked to individual servers, unable to be shared or reallocated.

CXL memory pooling enables higher performance and lower cost

Memory Pooling Delivers Superior Tokenomics

Tokens Per Second ▪ Tokens Per Dollar ▪ Tokens Per Watt

Dynamically pool, scale, and allocate CXL memory and GPUs in real time to ensure optimal utilization and maximum performance

Breaking the Memory Wall

Pioneering GPU and Memory Pooling

Resource Expansion Chassis

Host Interface Card

Optical Cables

Fabric Switch

30x GPUs

10x Per Chassis

or

160TB DRAM

40TB Per Chassis

SUPERMICRO · cisco · FUJITSU · DELL Technologies · Hewlett Packard Enterprise

Servers

Liqid Matrix
Fabric Manage

Memory Pools
Up to 160TB

GPU Pools
Up to 30 GPUs

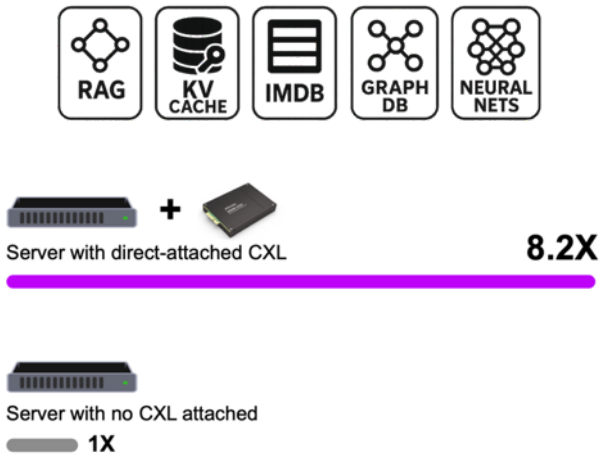
NVIDIA · AMD · intel · Micron · SAMSUNG

Key Workload Benefits of Memory Pooling

AI Acceleration

Highest Token/$ and Token/W

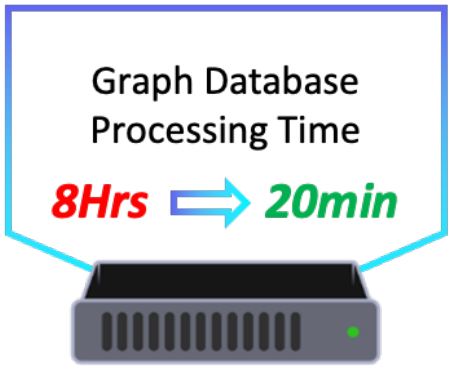
~10x Higher Performance



In Memory Database

Reduced Processing Time

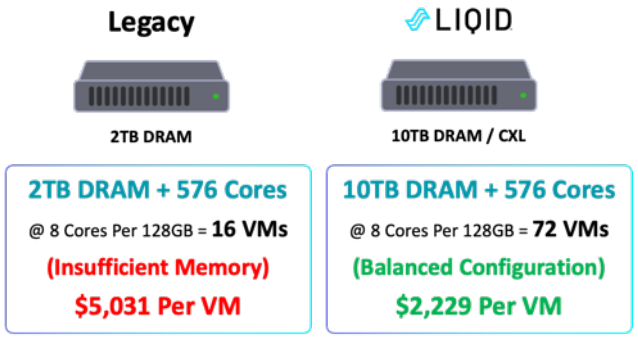
~20x Faster Queries



Virtualization

Increased VMs Per Server

~4x Higher VM Per Server



50% Reduced Cost Per VM

AI Inference Drives Memory Consumption

Limited Memory Supply

3x

Global Suppliers

Memory Supply Is Structurally Constrained

Memory is scarce, expensive, unable to scale with AI inference demand

Bigger Context Windows

250×

Context Growth

Context Windows Are Exploding Key-Value (KV) Cache Demand

KV Cache scales with context length, driving exponential memory demand

Memory Drives Cost

80%

HBM → KV cache

Inference Economics Are Breaking Down

Expensive HBM for KV cache is driving unsustainable inference costs

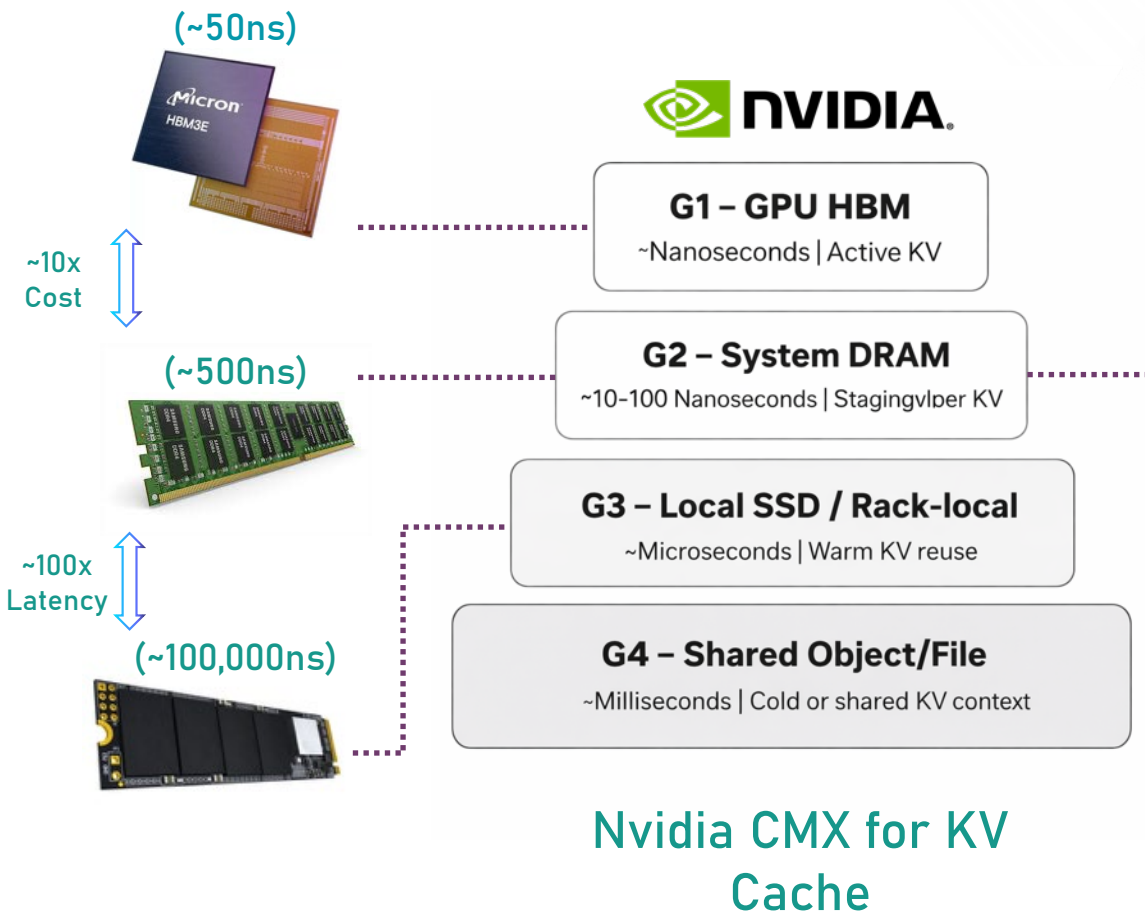
Larger Context Window Drives Higher Memory Demand for KV Cache

DRAM is Optimal Solution for GPU Acceleration

Cost: $$$$$$
Performance: High
Capacity: Low

Cost: $$
Performance: High
Capacity (w\ Pooling): High

Cost: $
Performance: Low
Capacity: High

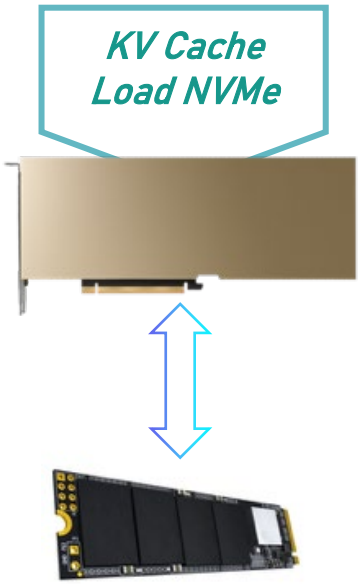


Nvidia CMX for KV Cache

DRAM pooling delivers optimal cost-performance with high capacity at lower cost

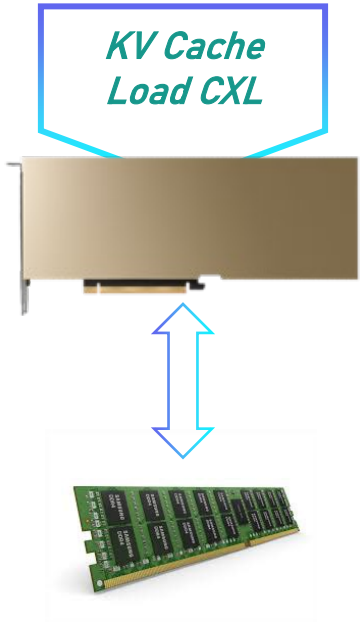
KV Cache Load Time | CXL vs. NVMe

Legacy Infrastructure

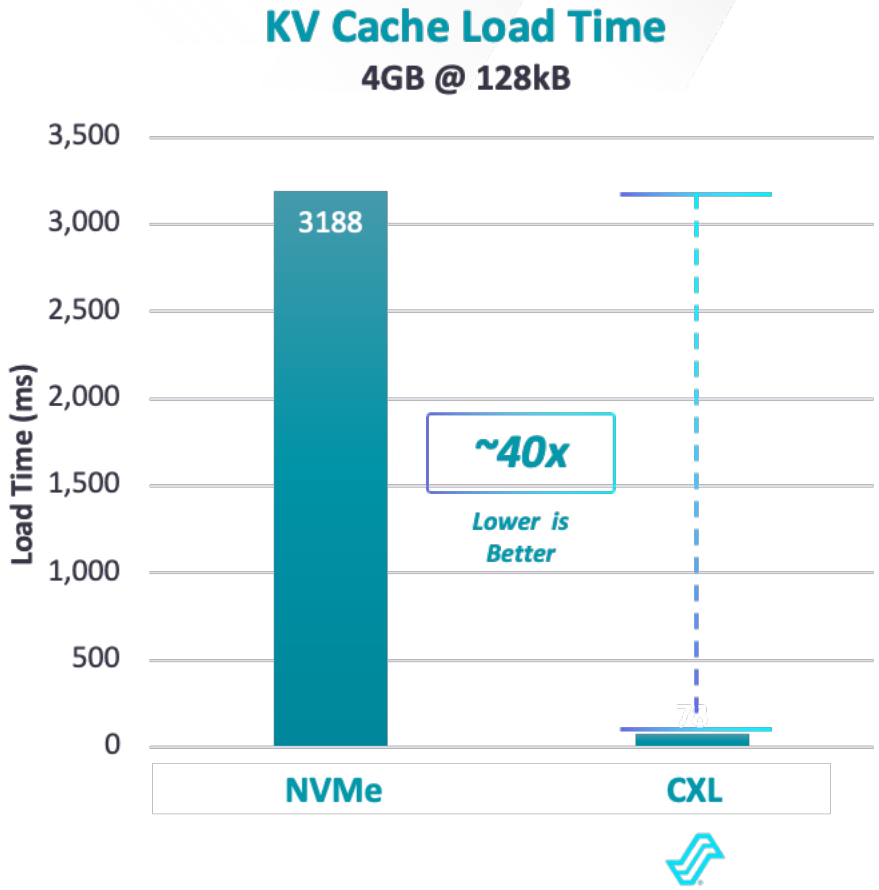


3,188ms Load Time
4GB @ 128kB

LIQID

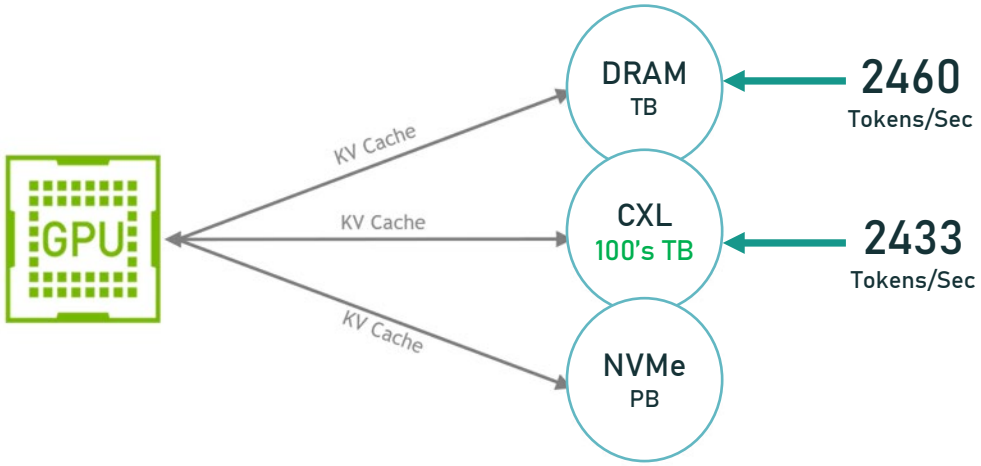


78ms Load Time
4GB @ 128kB



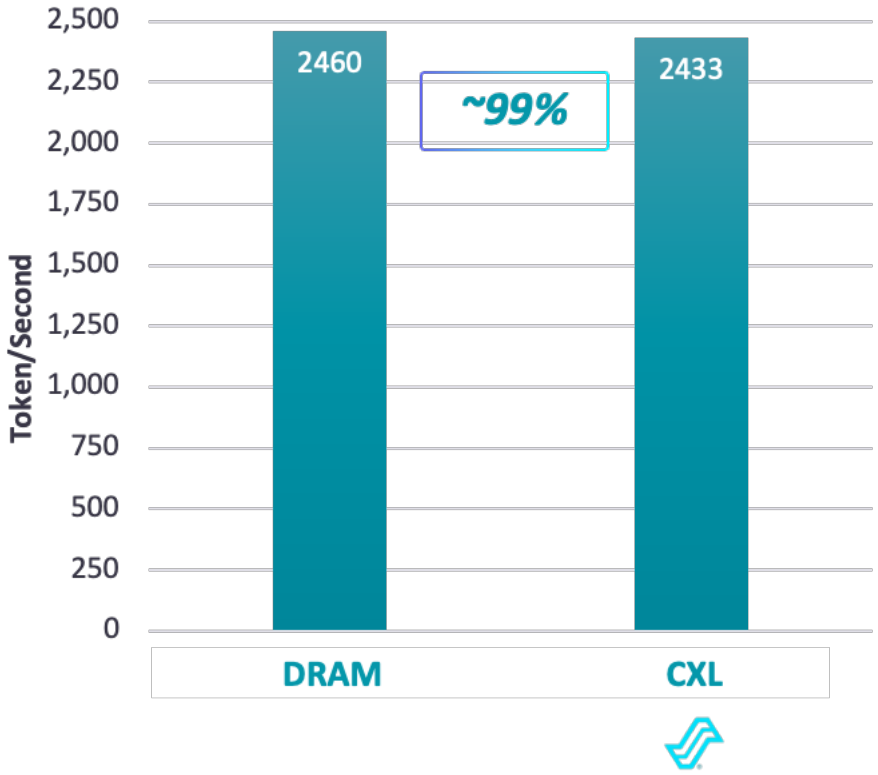
DRAM vs. CXL Performance | AIPerf Benchmark

Performance Delta on KVCache to DRAM vs. CXL



Offload KV Cache from DRAM to CXL Memory Pools Without Performance Loss at 100x Larger Scale

NVIDIA ai-dynamo/aiperf



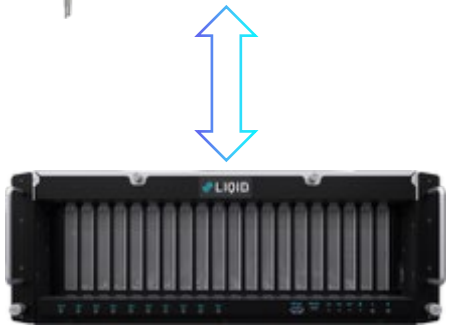
Cache vs. Recalculate | Performance vs. Hit Rate

Legacy Infrastructure

LIQID

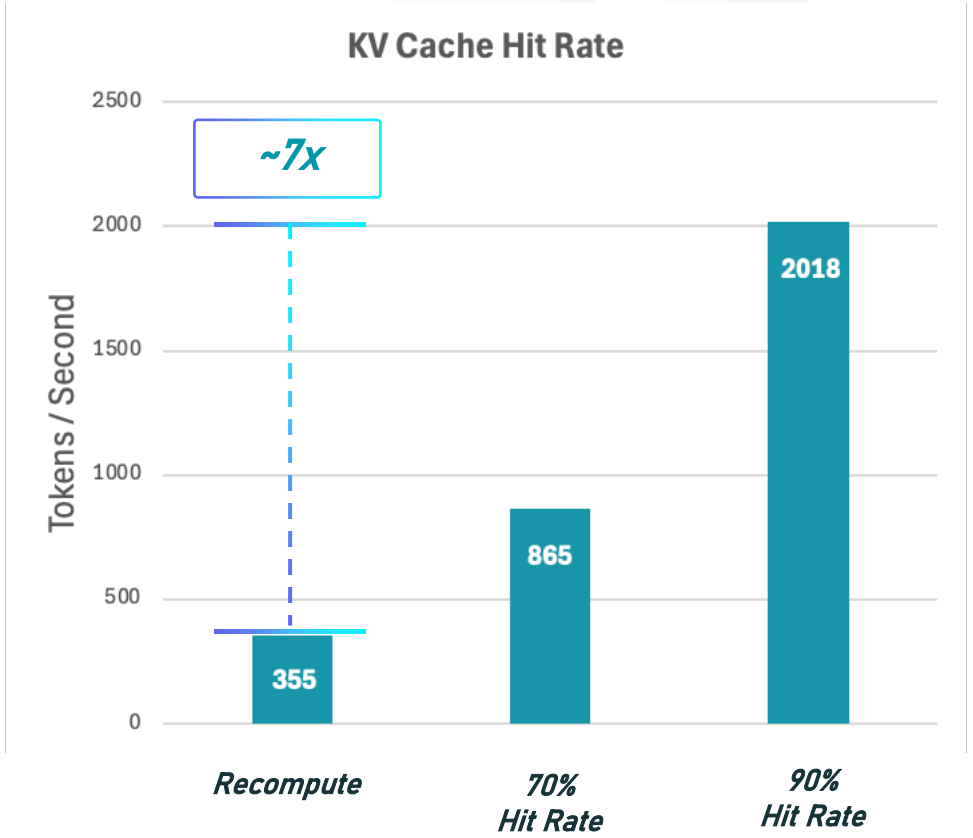
KV Cache Recompute

KV Cache 90% Hit Rate

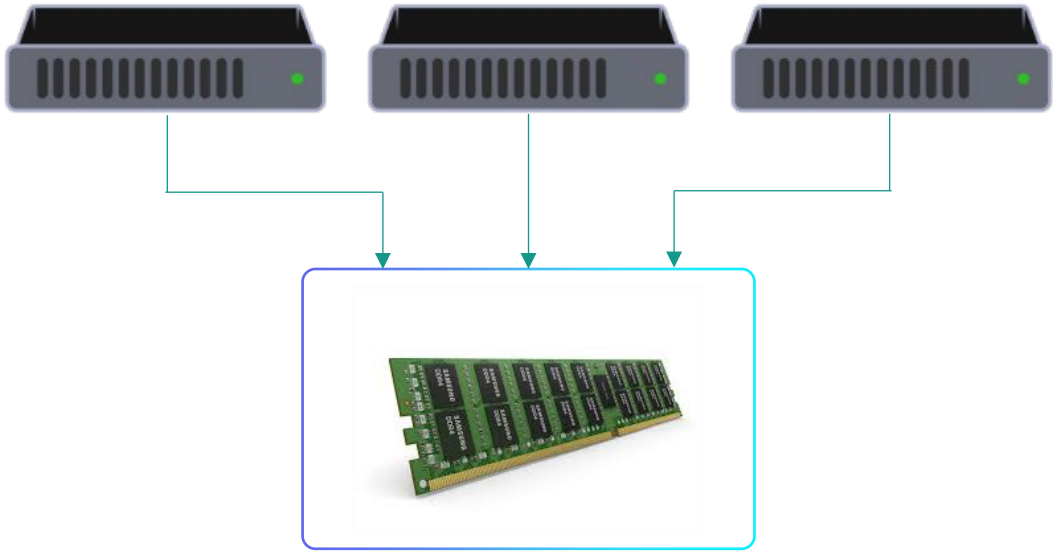


DRAM Memory Pooling Enables GPUs to Achieve Higher Hit Rate for KV Cache

90% KV Cache Hits Enable 7x Higher Performance



Multi-Server Memory Sharing



Host 1 (10.42.222.58)

```
Last login: Wed May 21 17:42:
liqid@gnr:~$ daxctl list
[
  {
    "chardev":"dax1.0",
    "size":5495410655232,
    "target_node":3,
    "align":2097152,
    "mode":"devdax"
  },
```

Host 2 (10.42.222.57)

```
liqid@gnr2:~$ daxctl list
[
  {
    "chardev":"dax1.0",
    "size":5495410655232,
    "target_node":3,
    "align":2097152,
    "mode":"devdax"
  },
  {
```

- Multi-Server Memory Sharing
- Simultaneous Memory Access
- SW Level Cache Coherency

cxl-micron-reskit/
famfs

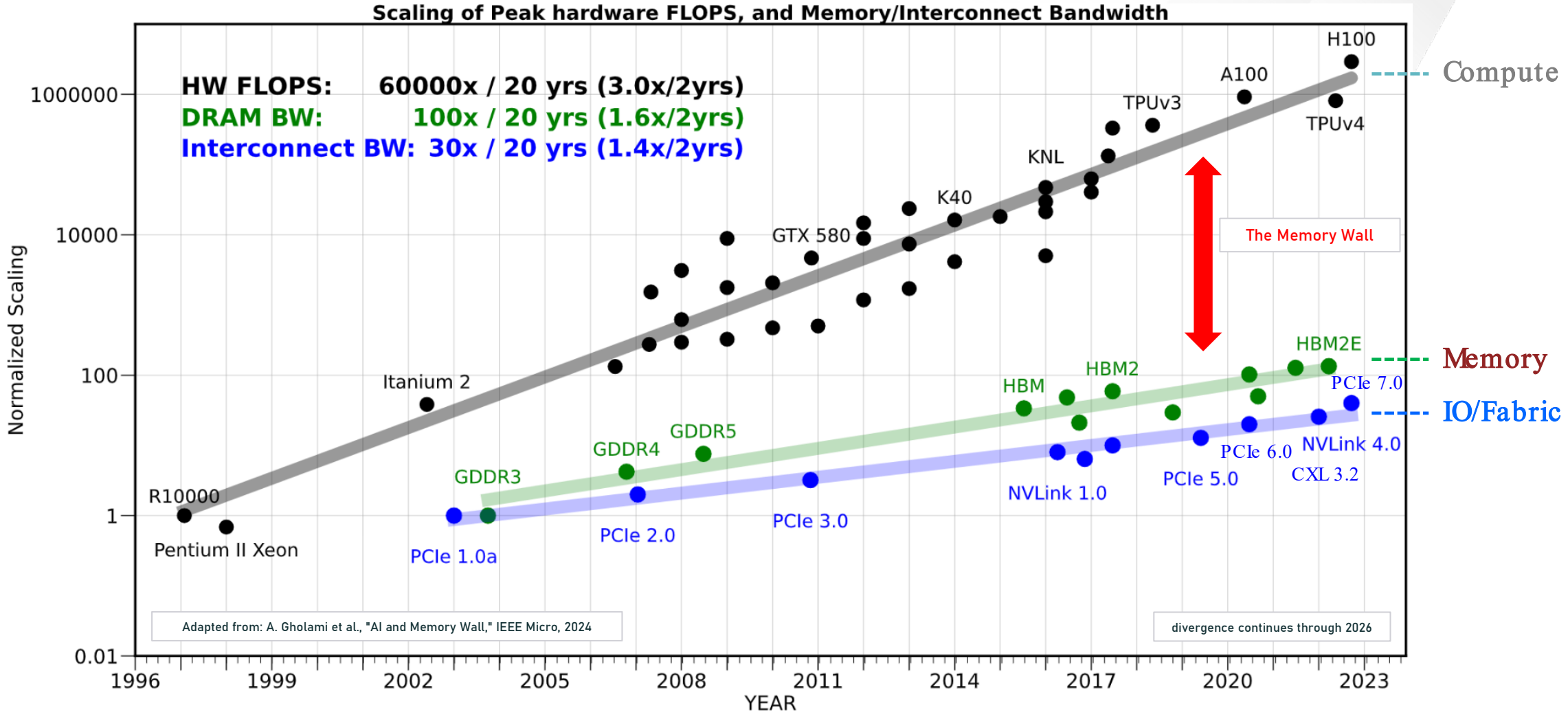
Released to Linux Kernel

Transforming AI Inference with CXL Memory Pooling and Sharing

Ronen Hyatt (UnifabriX)

The AI Memory Wall is a major roadblock to scaling compute

The gap is structural, not cyclical — In twenty years, compute 60,000×; memory bandwidth 100×; the fabric 30×



Workload inflection points: Driving Demand Straight into the Memory Wall

Structural shift to inference and agentic AI

	2025		2026	
Training	→	Inference	→	Agentic AI
Compute	→	Memory	→	Stateful Memory

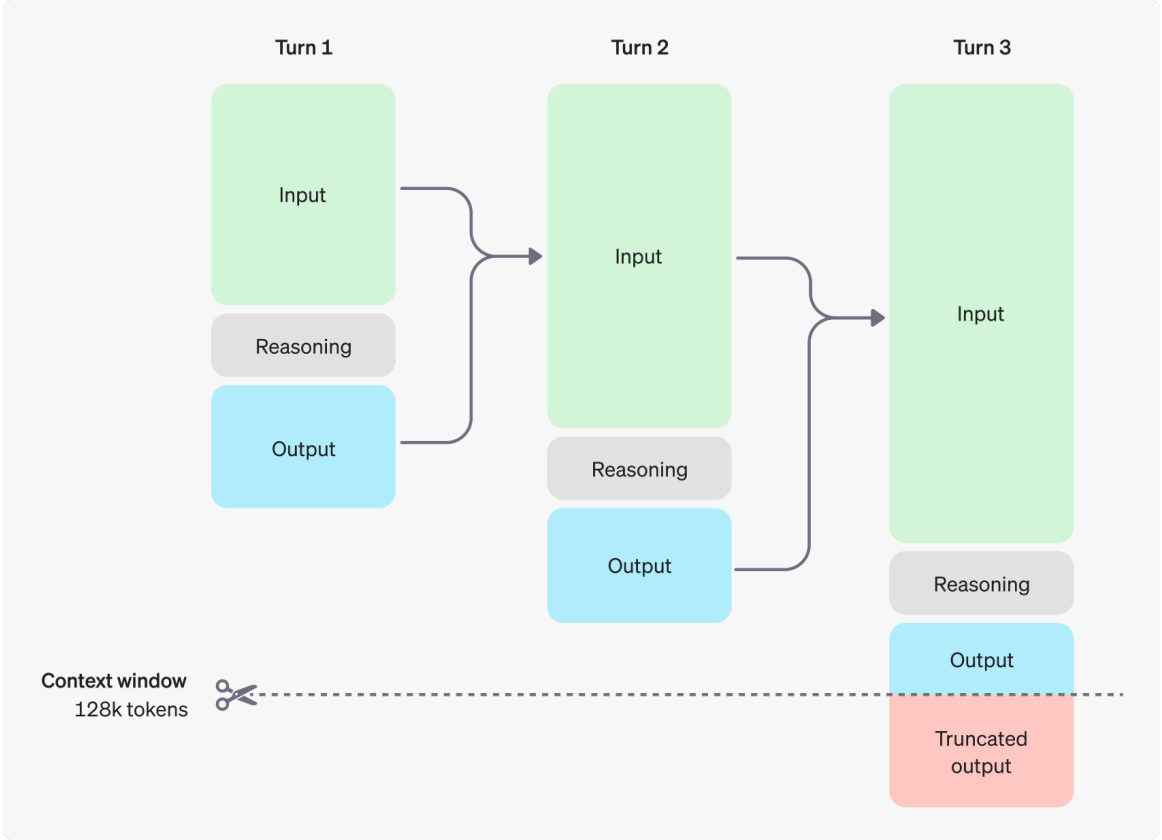
Agents carry state across steps and sessions (plans, context, tool results), and that state must stay resident in memory.

The workload: AI inference is a memory problem

Smarter LLM reasoning models and agentic AI drive memory pressure and costs up — GPU HBM is too small and too expensive to scale

- KV-cache dominates memory footprint at long contexts and with batching
  - Example: Llama-3-70B, 128k context KV-cache ~40 GB (batch=1)
  - KV grows linearly with sequence length and batch size, driving memory pressure up fast
- KV-caches are typically stored in GPU HBMs, the most expensive and scarce memory resource of an AI infrastructure
- Bigger memory footprints force multi-GPU sharding
  - Adding communication/synchronization overheads and lowering efficiency
  - Results in using more GPUs per request than pure compute would require
  - Contributing to higher costs of AI inference

Reasoning models require larger context windows, i.e., larger memory



Llama-3-70B (80 layers, GQA, FP16); ~0.3MB of KV per token, ×128k ≈ 41GB per request at batch=1

The invoice: Memory has become a scarce, expensive resource

Utilization, not capacity, is the lever that's left. Scarcity is policy, not cycle: wafers moved to HBM, 2026 prices set records.

Server memory prices could double by 2026 as AI demand strains supply

NETWORKWORLD

News

Nov 20, 2025 • 5 mins

Reuters

The AI frenzy is driving a memory chip supply crisis

By **Hyunjoo Jin**, **Fanny Potkin**, **Wen-Yee Lee**, **Anton Bridge** and **Max A. Cherney**

December 3, 2025 5:15 AM GMT+2 · Updated December 3, 2025

Reuters

Exclusive: Samsung hikes memory chip prices by up to 60% as shortage worsens, sources say

By **Hyunjoo Jin** and **Fanny Potkin**

November 17, 2025 4:28 AM GMT+2 · Updated November 17, 2025

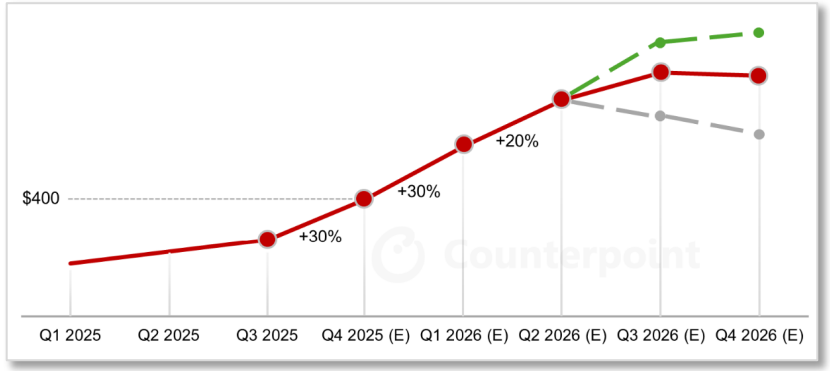
Advanced Memory Prices Likely to Double as DRAM Crunch Spreads on NVIDIA Pivot, Structural Factors

0 November 19, 2025 Counterpoint

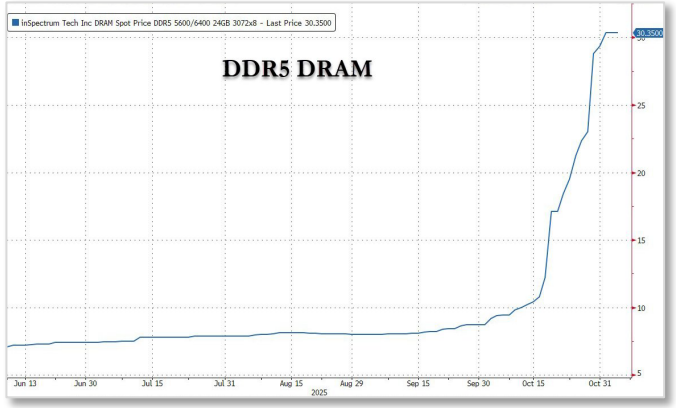
Sky-high DRAM prices are set to stick around beyond 2028 as Samsung and SK Hynix opt to 'minimize the risk' memory of oversupply' yahoo!finance

DRAM Prices Surge 172% YoY with No Signs of Slowing Down

by AleksandarK | Nov 6th, 2025 09:52 | Discuss (84 Comments)



Source: Counterpoint Research 'Memory Solutions for GenAI' Bi-Weekly, Issue 21



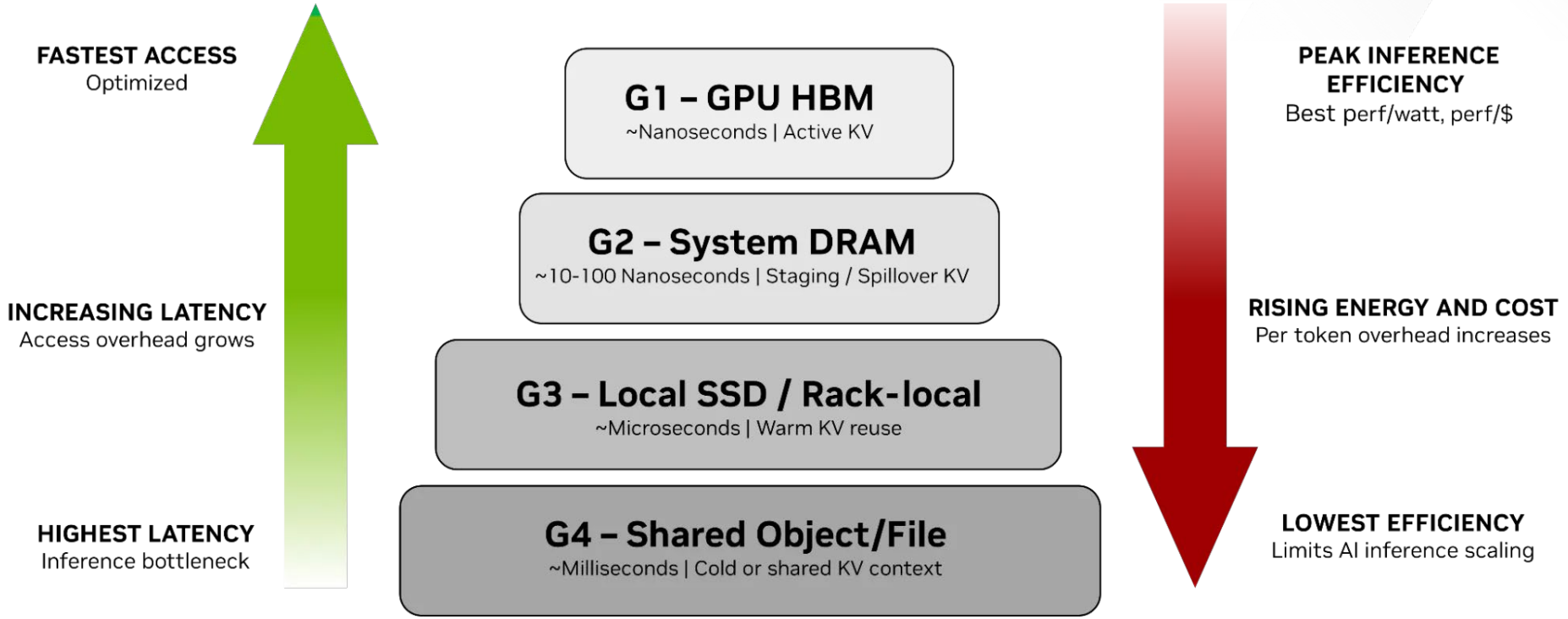
Reuters

Micron to exit consumer memory business amid global supply shortage

By **Reuters**

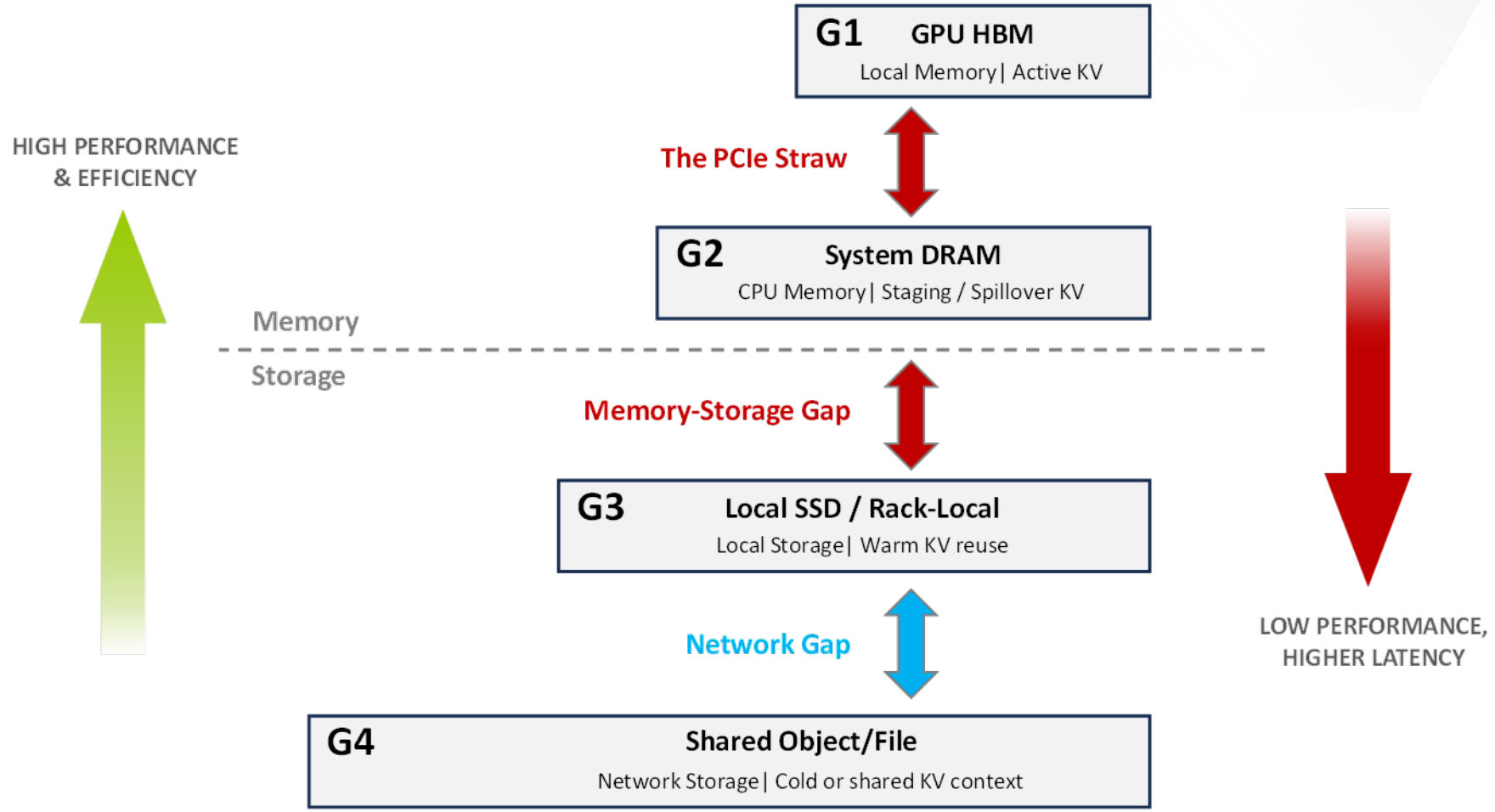
December 3, 2025 7:37 PM GMT+2 · Updated December 3, 2025

The Memory Hierarchy: KV Cache Point-of-View



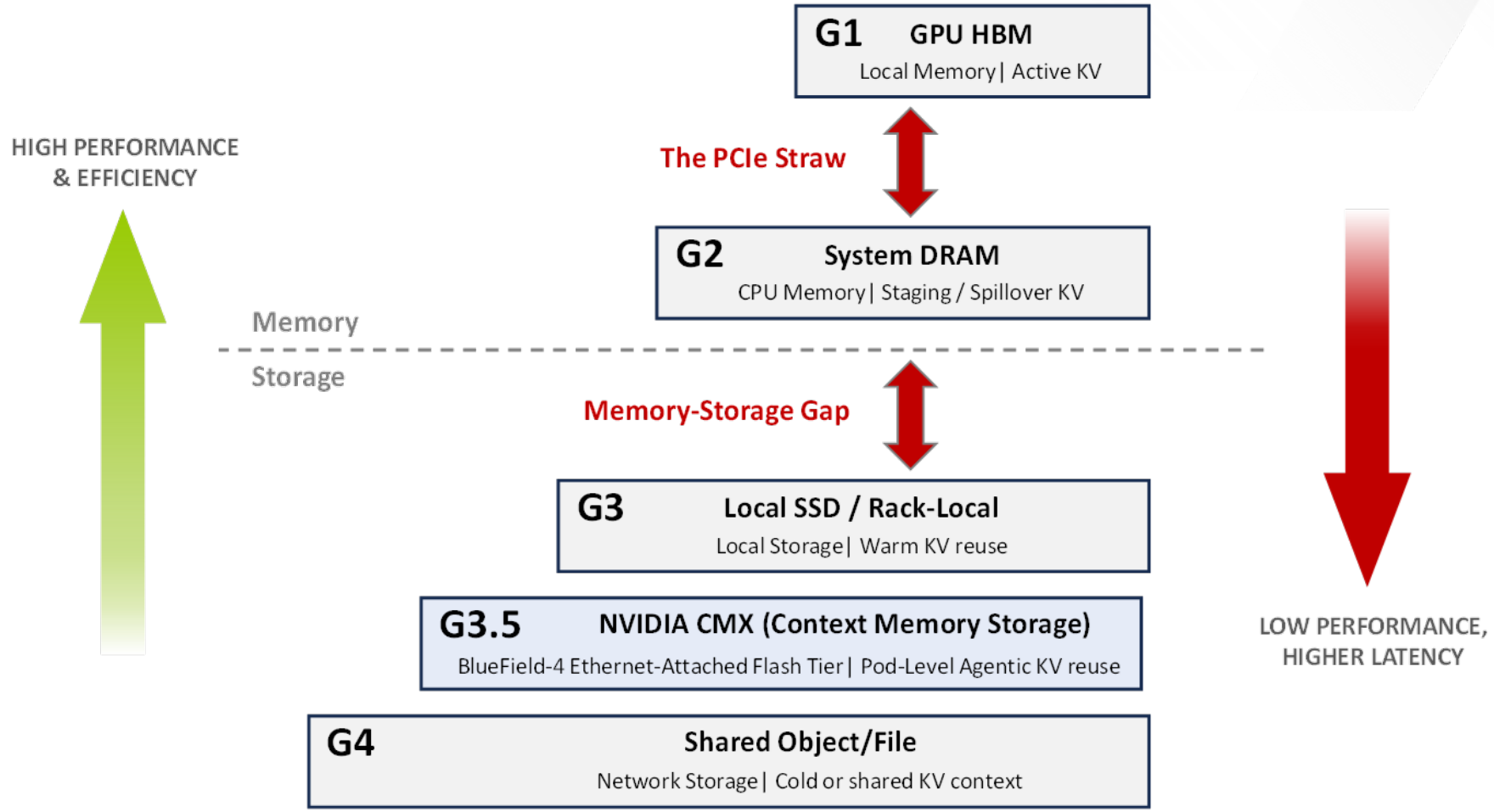
The memory hierarchy has two (three) empty tiers

NVIDIA's own layering model defines the gaps



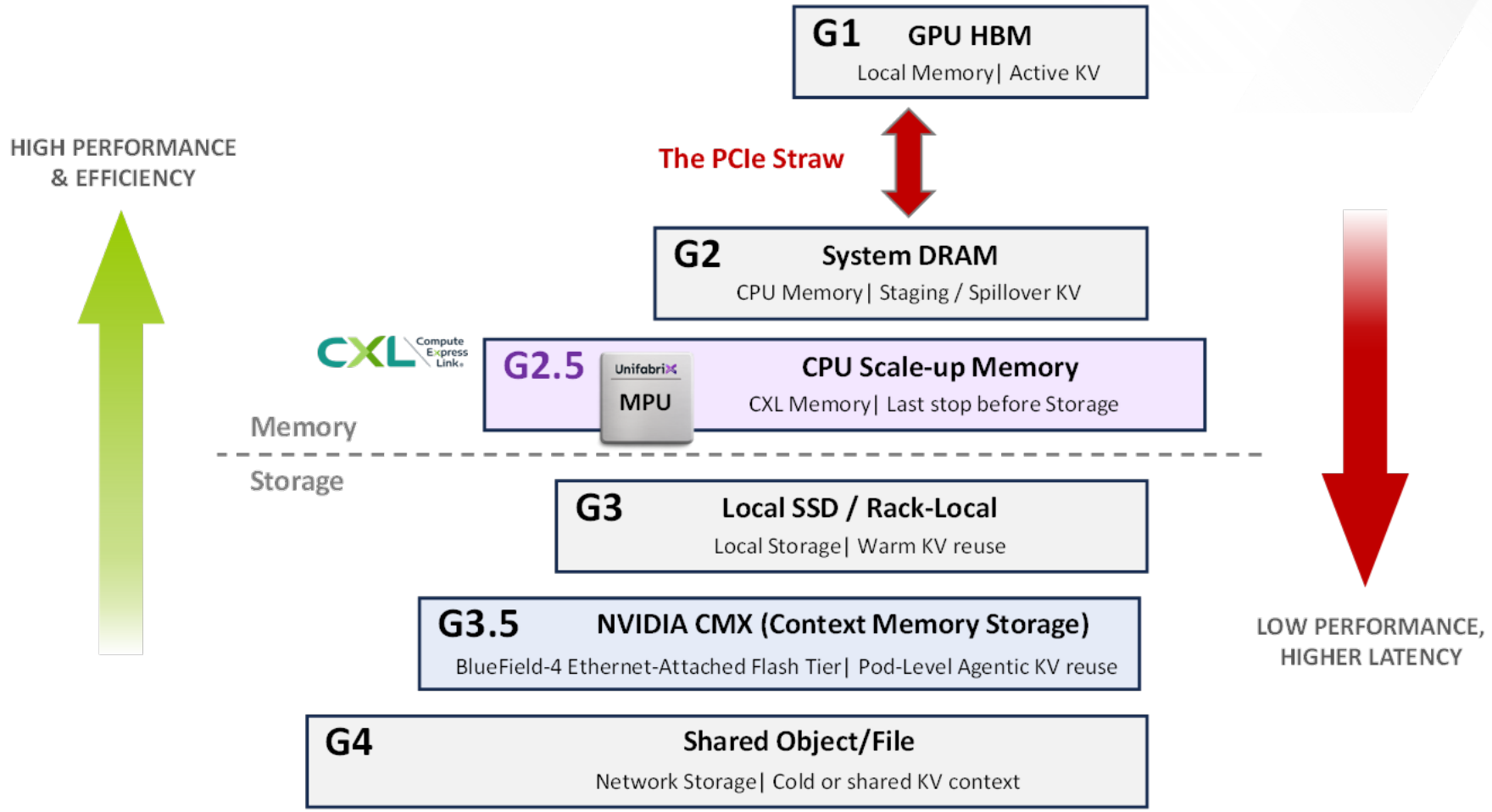
The memory hierarchy has two (~~three~~) empty tiers

NVIDIA's own layering model defines the gaps: its CMX platform fills G3.5 with Ethernet-attached flash below the storage line. The memory-side missing tiers above it are not addressed.



Putting CXL to Work: Introducing UnifabriX MPU @G2.5 — Shared CXL Memory Pool

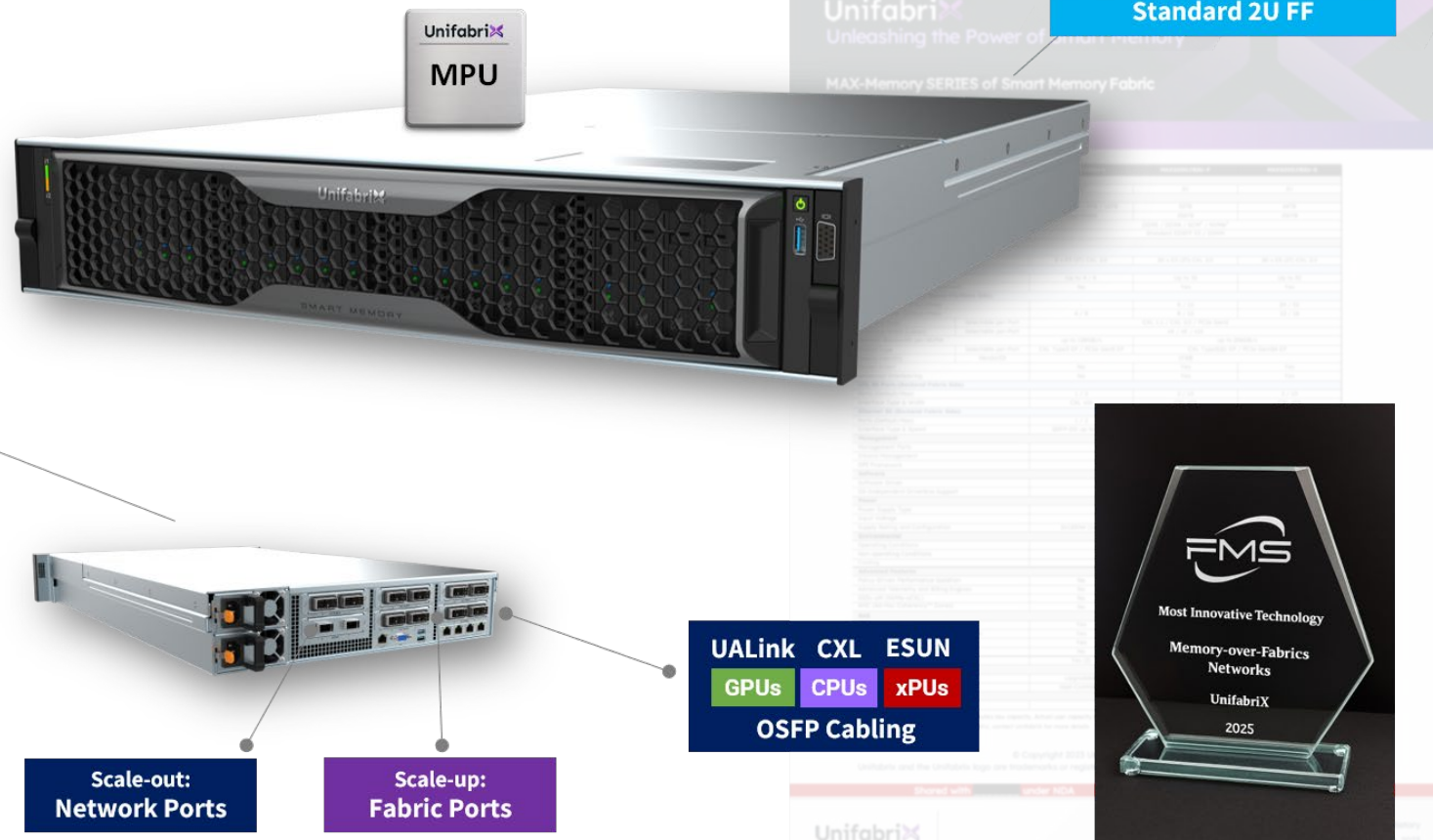
MPU (Memory Processing Unit) @G2.5 provides a shared KV store way above G4



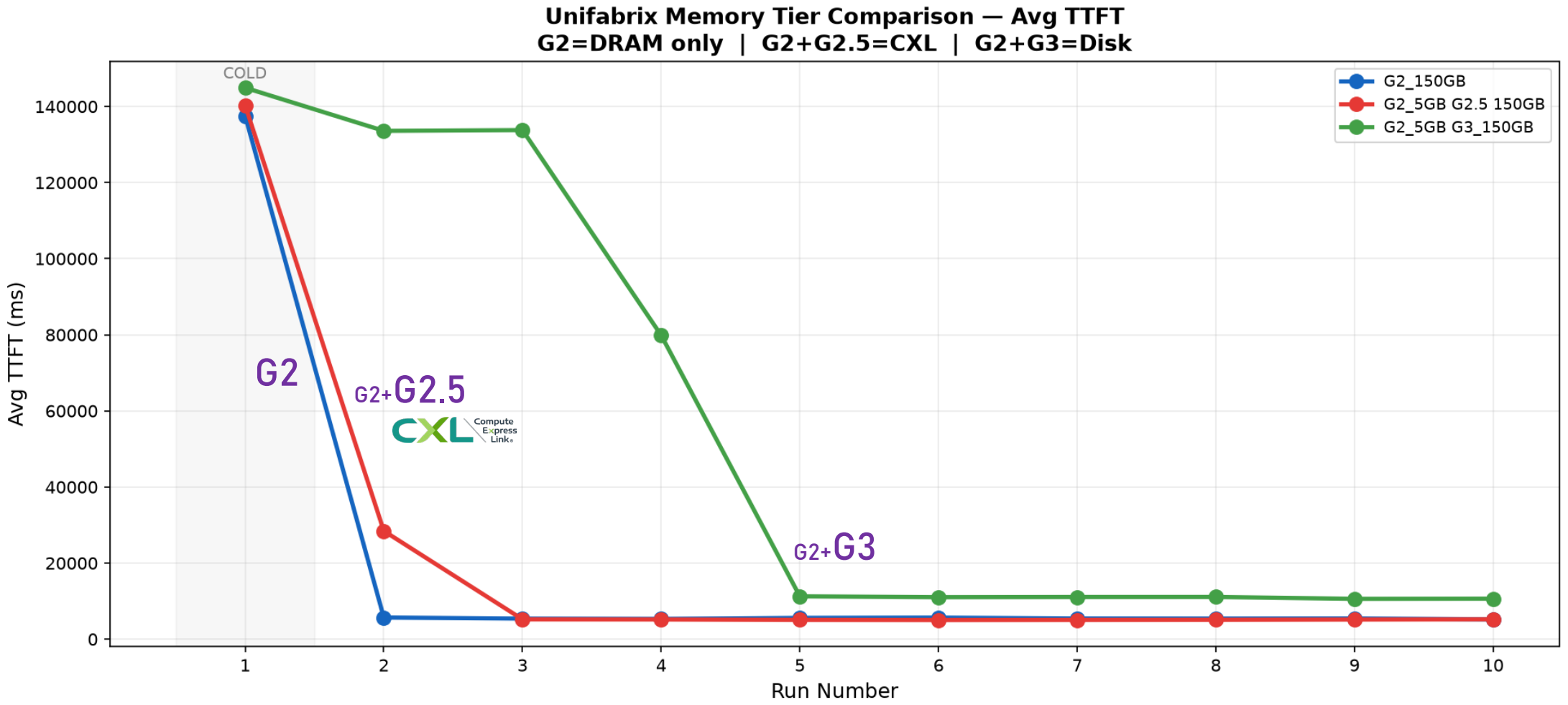
UnifabriX Memory Switch Platform

Powered by UnifabriX MPU (Memory Processing Unit), FMS'25 Award for Most Innovative Technology

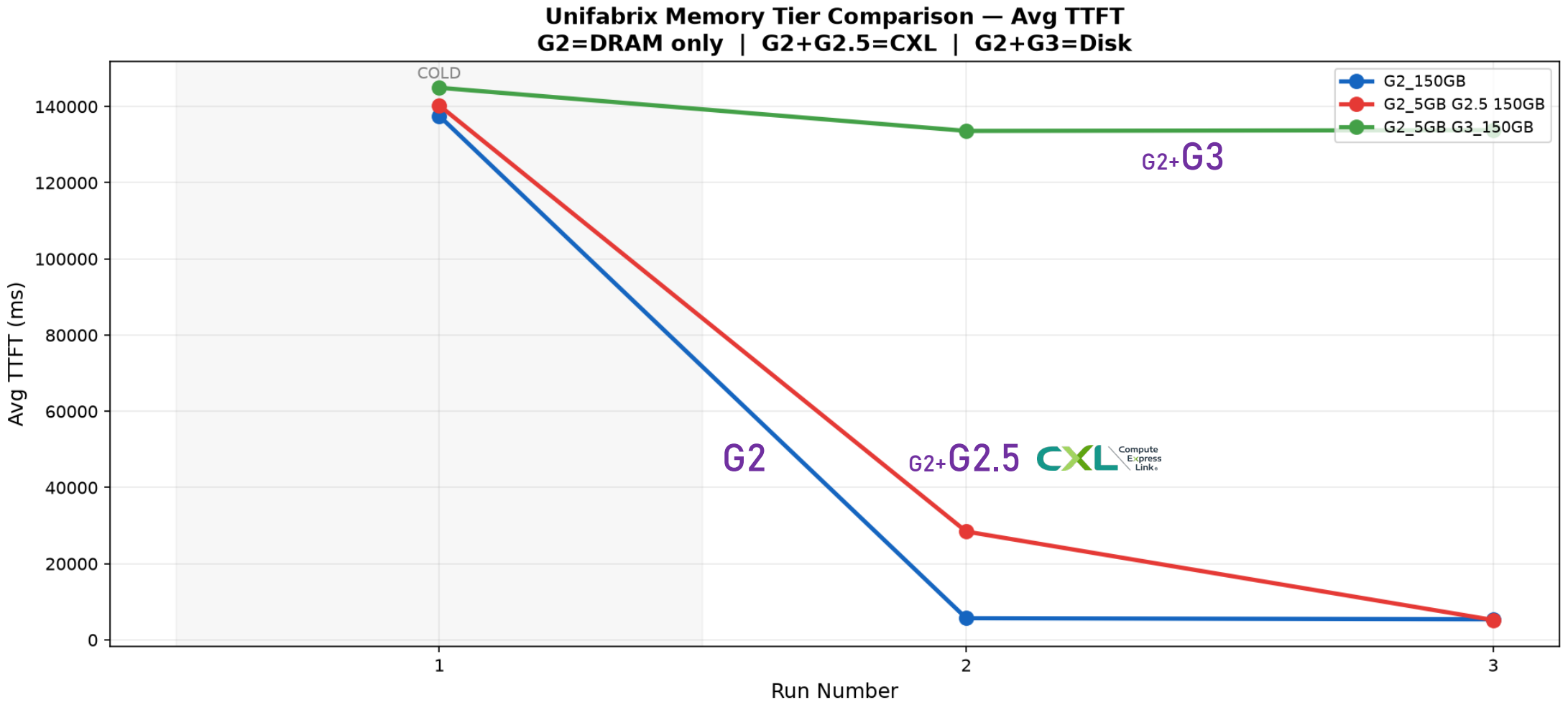
- Fabric Independent
- Memory Pooling
- Adaptive Memory Sharing
- Transactional Switch
- Large Address Space
- Server-Grade RAS
- Link Redundancy
- Memory PFA (Predictive Failure Analysis)
- Mirroring and Striping
- Resource Interleaving
- Processing Near Memory
- Security
- Memory Encryption
- FlexMemory



UnifabriX G2.5 Memory Switch Platform: Benchmarking llama3.1-70B

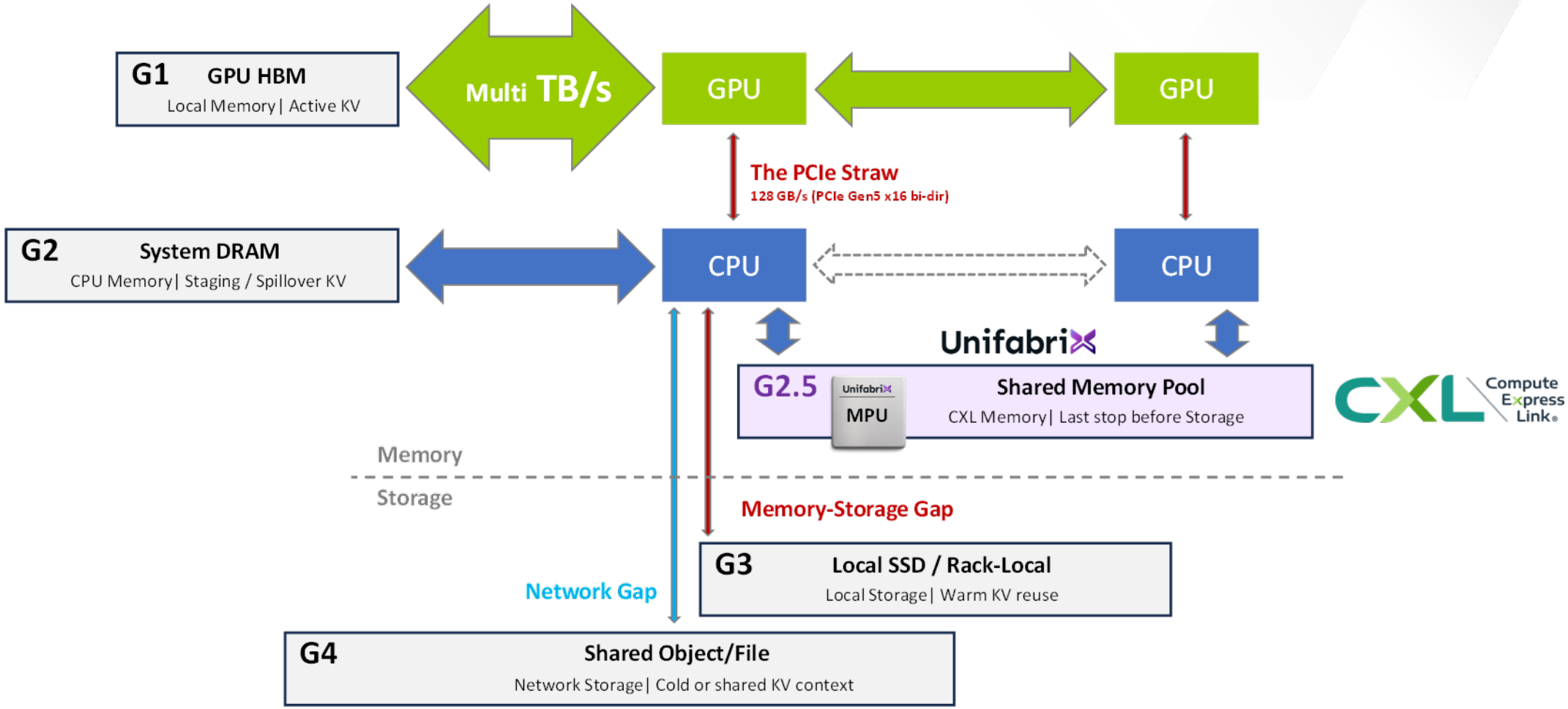


UnifabriX G2.5 Memory Switch Platform: Benchmarking llama3.1-70B



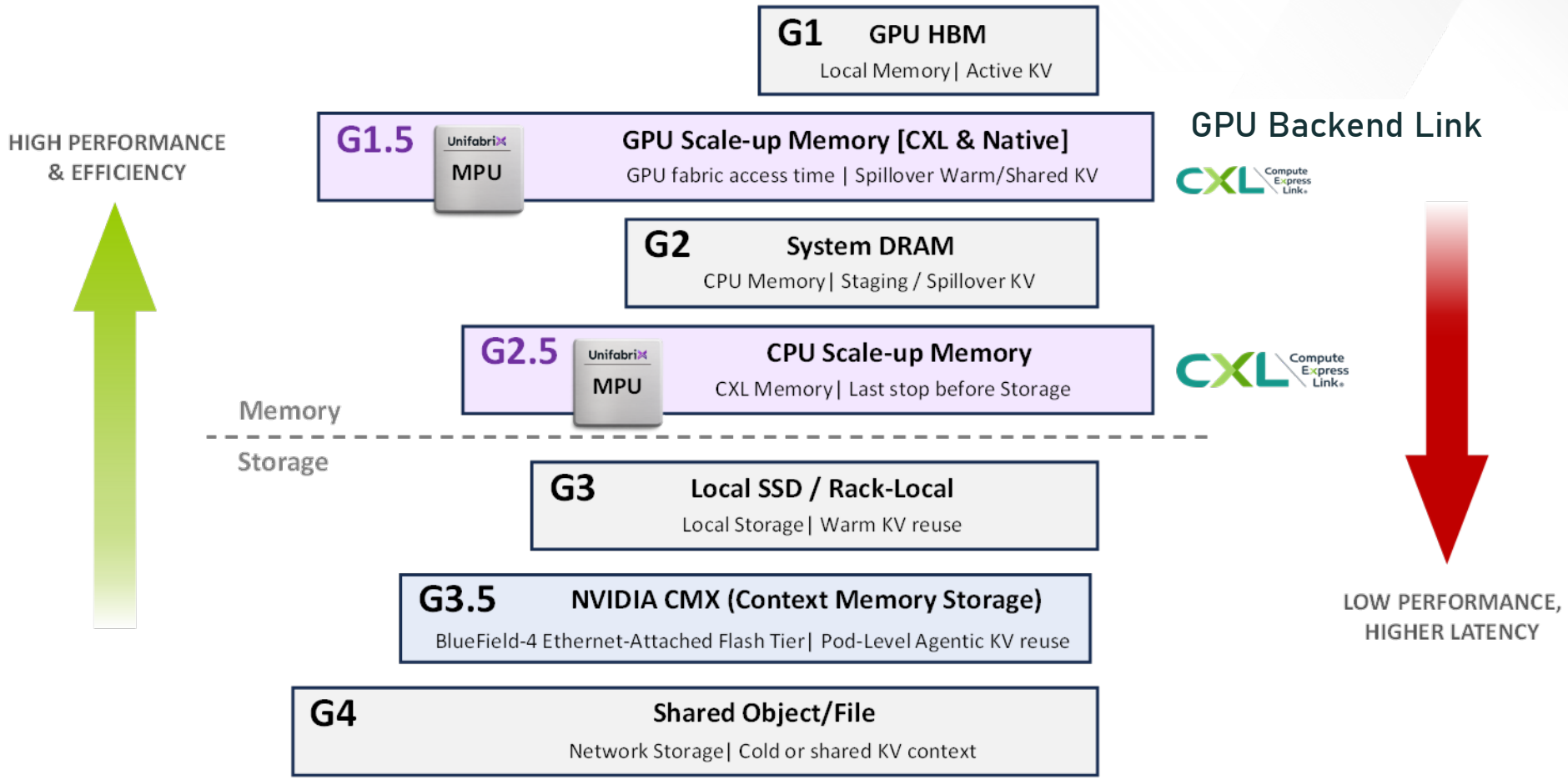
Mapping the Bottlenecks

G1-to-G2 is currently a bottleneck over PCIe, with next PCIe generations providing significant relief. Proprietary G1-to-G2 (e.g., NVLink-C2C).



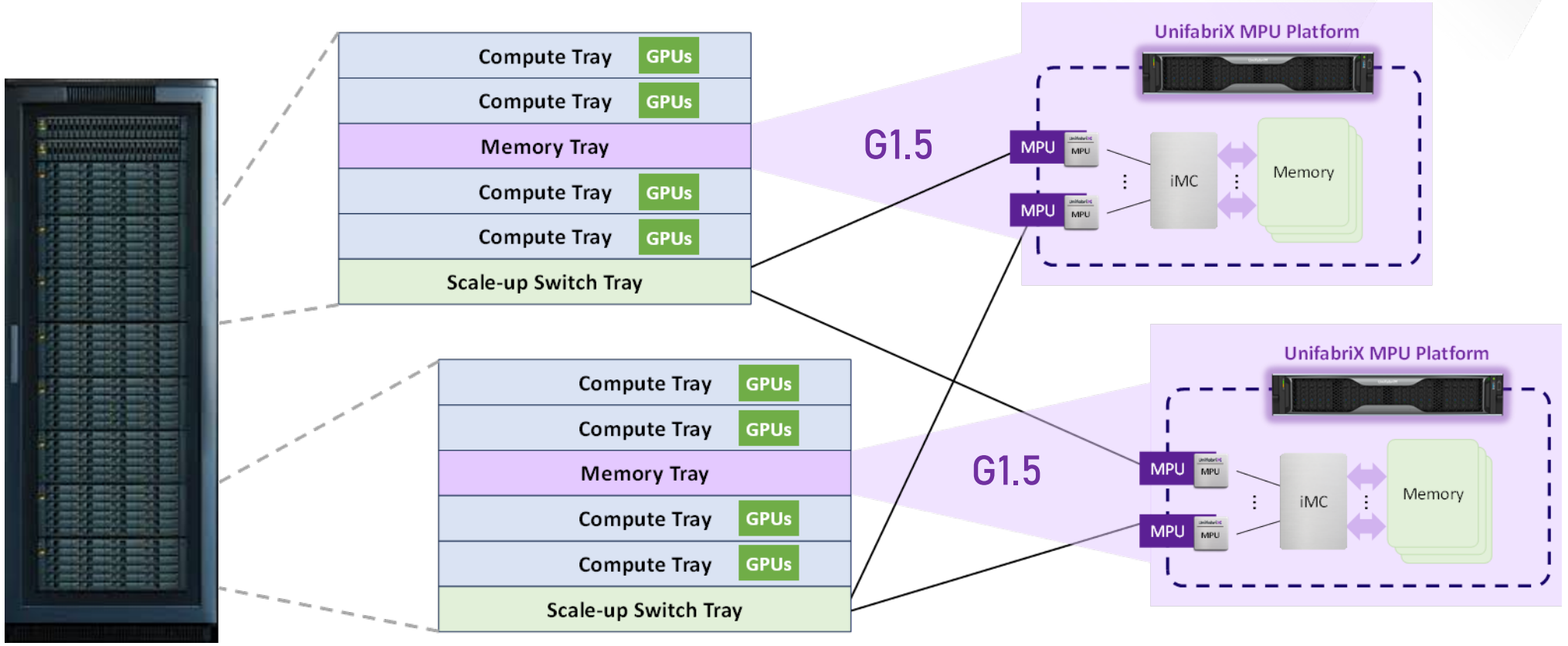
Finalizing the Model: Introducing UnifabriX MPU @G1.5/G2.5 — CXL is all around

Shared layers at G1.5 and G2.5 in the memory scale-up domain



G1.5: Hyperscaler's blueprint architecture of MPU deployment in AI datacenters

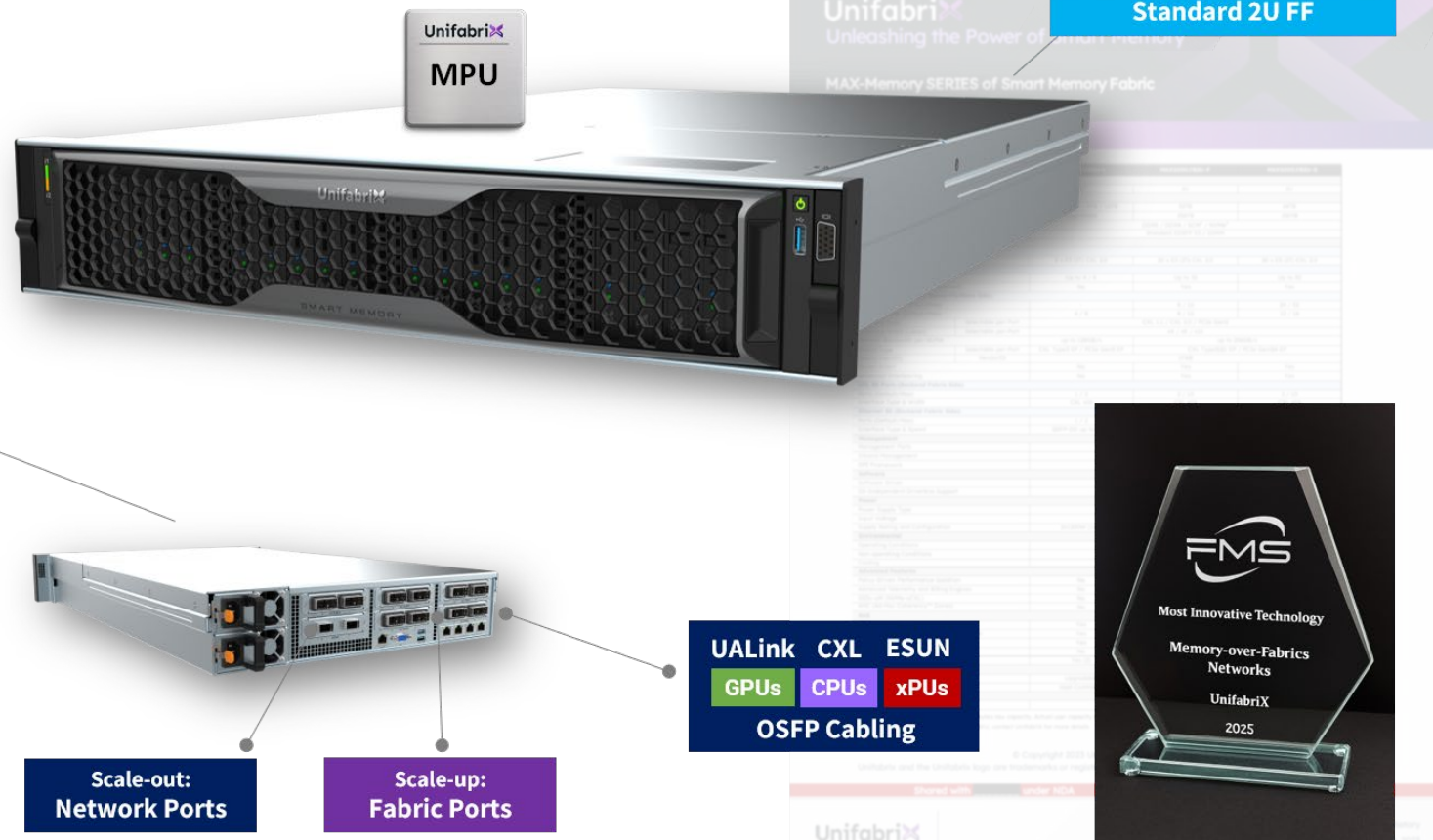
Memory as infrastructure. Reachable by any GPU over the scale-up fabric, within and across racks.



UnifabriX Memory Switch Platform

Powered by UnifabriX MPU (Memory Processing Unit), FMS'25 Award for Most Innovative Technology

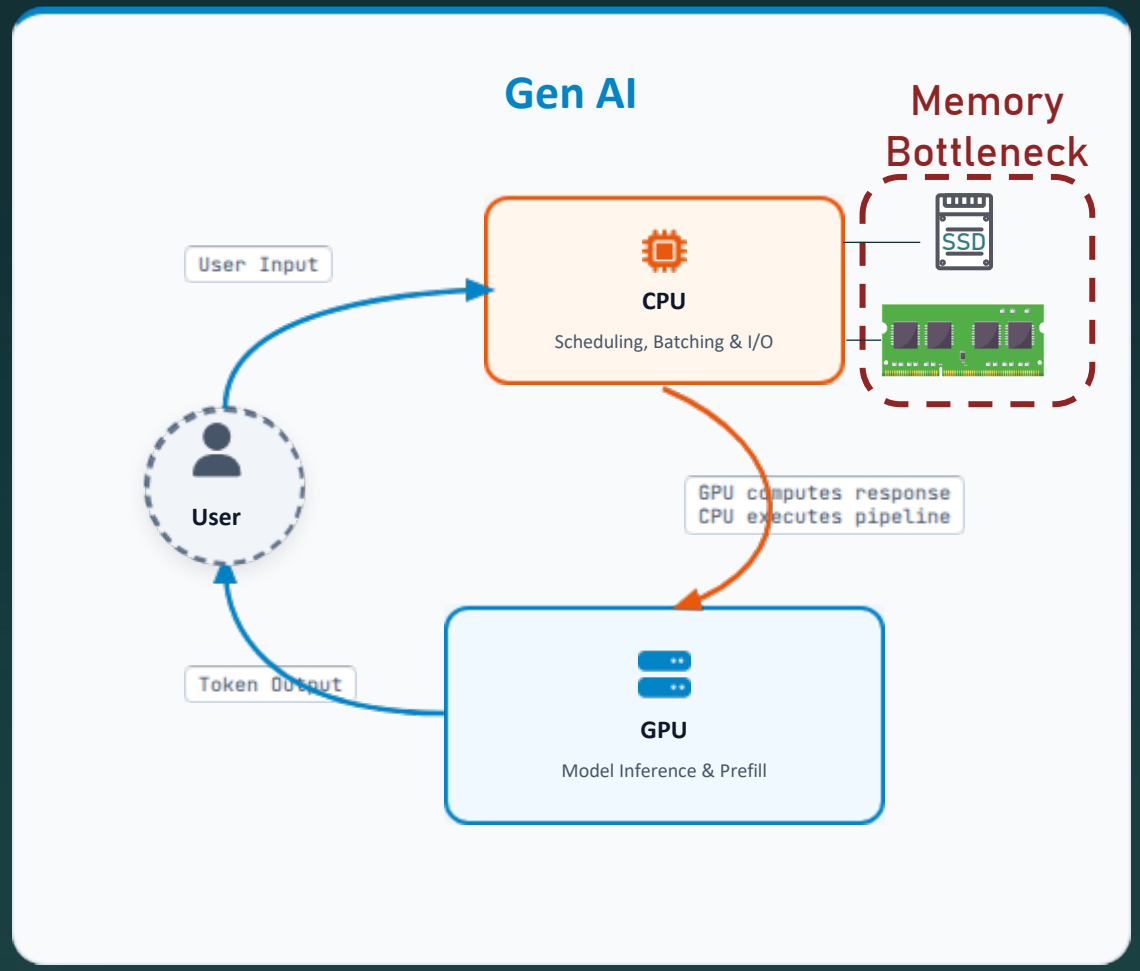
- Fabric Independent
- Memory Pooling
- Adaptive Memory Sharing
- Transactional Switch
- Large Address Space
- Server-Grade RAS
- Link Redundancy
- Memory PFA (Predictive Failure Analysis)
- Mirroring and Striping
- Resource Interleaving
- Processing Near Memory
- Security
- Memory Encryption
- FlexMemory



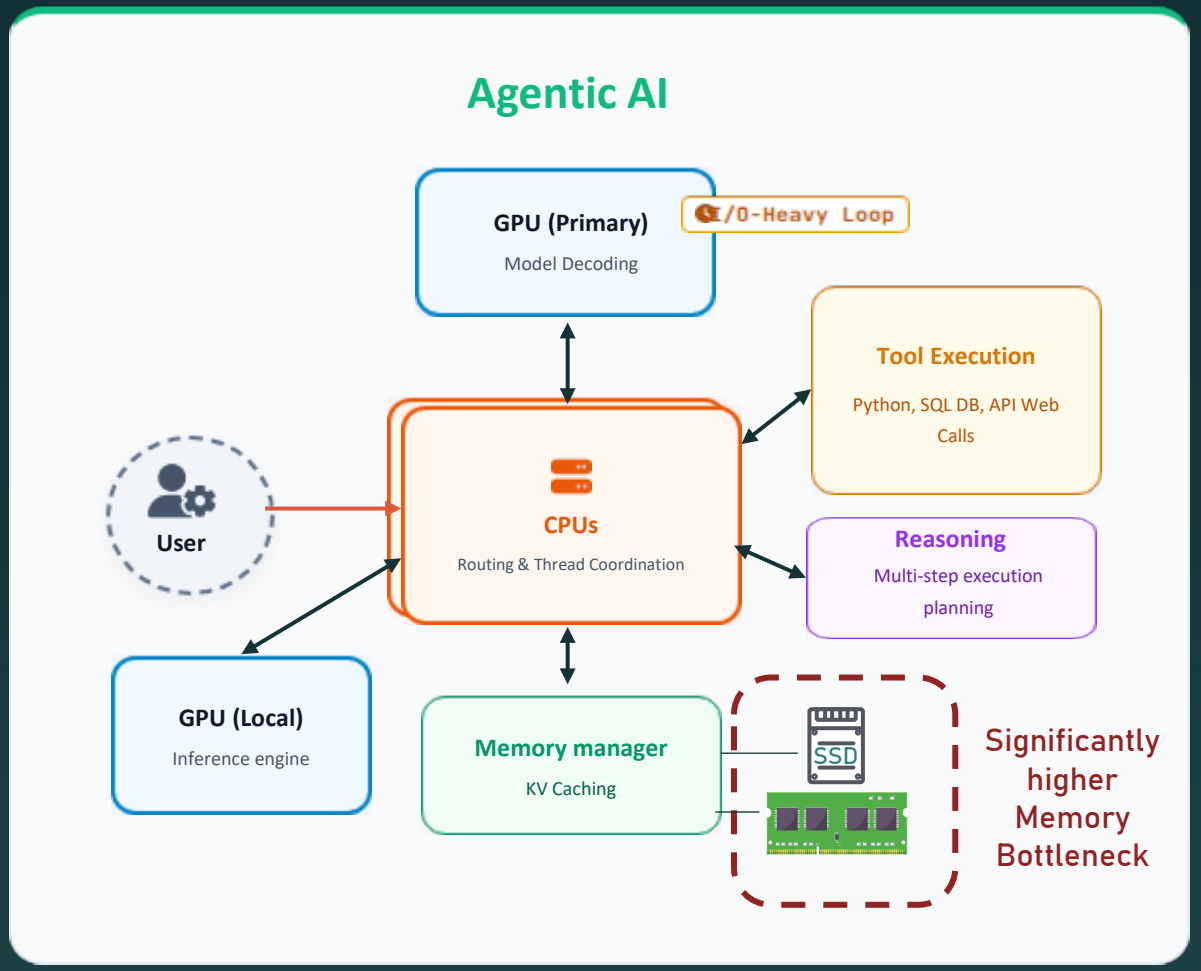
CXL Pooling & Sharing for AI Inference

Sandeep Dattaprasad (Astera Labs)

Inference Bottlenecks in Modern AI Systems



GPU → performs Computation
CPU → orchestration
3-4 GPU : 1 CPU



GPU → performs Computation/token Gen
CPU → orchestration, execution, planning
1 GPU : 1 CPU

Flexible KV Cache Solutions for LLMs and Agentic AI with Leo

AsteraLabs
Leo
CXL Smart Memory Controller
P-Series

LLM Application Examples

Chatbots
reuse early chat content as context for later chat input

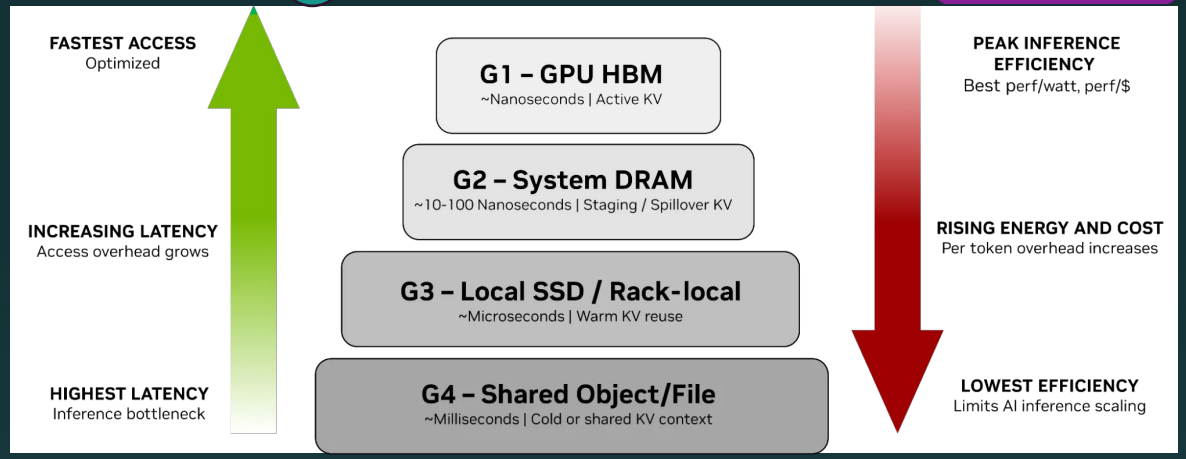
Bug Fixing
Frequently using entire code repository as context

Agents
Assistant and rec sys query large document sets

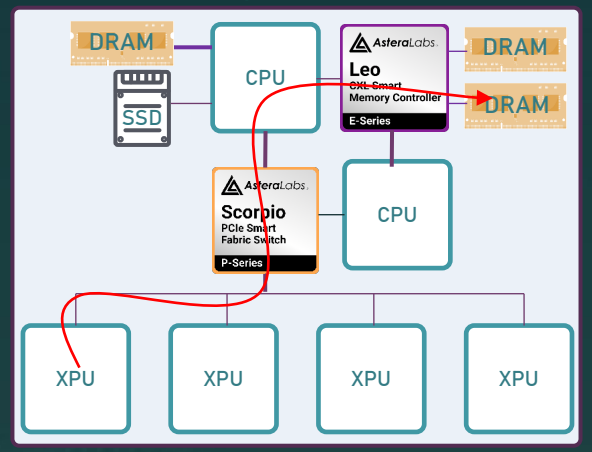
1 Increasing KV Cache Capacity

- Larger models
- Longer context
- Multi-turn conversations
- Reuse across agents
- Concurrent users/sessions
- Long term memory

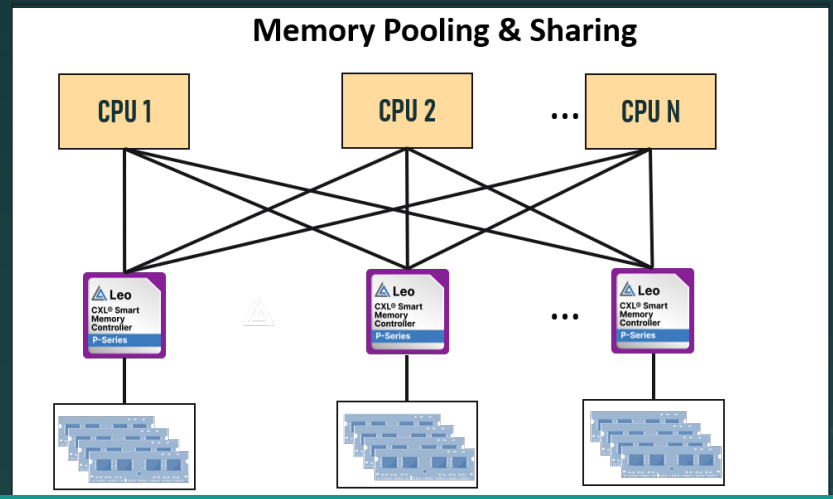
2 Multiple Memory Tiers



3 More G2 memory w/ pooled/shared CXL



4 Higher Memory Utilization w/ pooling/sharing



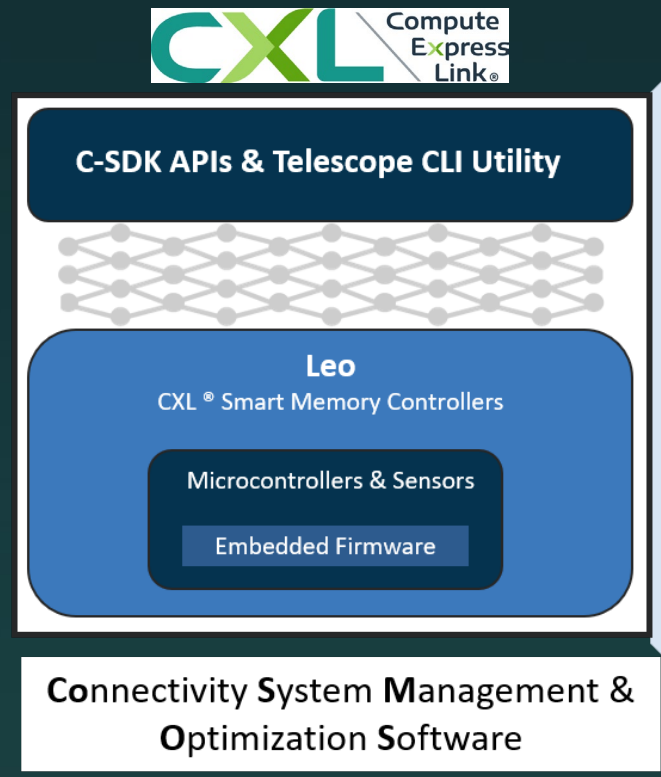
Leo + Scorpio enable multiple memory tiers and flexible system/rack architectures

CXL Compute Express Link

Leo Memory Pooling/Sharing Feature Enablement

Feature	Description	Leo 1 Support
Link resilience	Link stability and resilience w/ multiple host connection	• Verified
Memory management	Pool allocation/deallocation	• Static allocation/deallocation verified • Dynamic allocation/deallocation in progress
Reset handling	Stability across host resets (cold/warm)	• Verified
Link Resilience	Graceful handling of link up/down events on host(s)	• Verified • Additional testing in progress
Telemetry	Logging, Monitoring & host notification	• Verified
Inband management	Primary and secondary mailbox from individual hosts via PCIe	• Verified
Out-Of-Band management	CXL spec defined and COSMOS SDK features via I2C/I3C	• Verified
RAS	Spec defined RAS features	• CXL 2.0 error handling verified • Additional testing in progress
Performance	BW and latency with different benchmarks/user applications/workloads	• Baseline performance testing complete • Additional benchmarking in progress
Security	Memory isolation to protect user data across multi-host topology	• In progress
Fabric management	Mgmt between nodes and efficient utilization of memory	• FW/COSMOS framework verified • Pending host SW/BMC support availability

Leo CXL Smart Memory Controller COSMOS



Connectivity **S**ystem **M**anagement & **O**ptimization **S**oftware

Link Management

- PCIe® features - LTSSM, lane margining, AER, etc.
- Signal integrity features – FOM, eye capture
- DDR 2D margining, JEDEC DDR training/init seq

Reliability, Availability, Serviceability (RAS)

- CXL 3.2, JEDEC CMC02 RAS features
- CXL mailbox, CCI, viral/poison
- Advanced ECC detection & correction (Chipkill)
- Memory pattern generation and testing
- PPR, Row Hammer mitigation, scrubbing, etc
- DIMM component error detection & reporting

Fleet Management

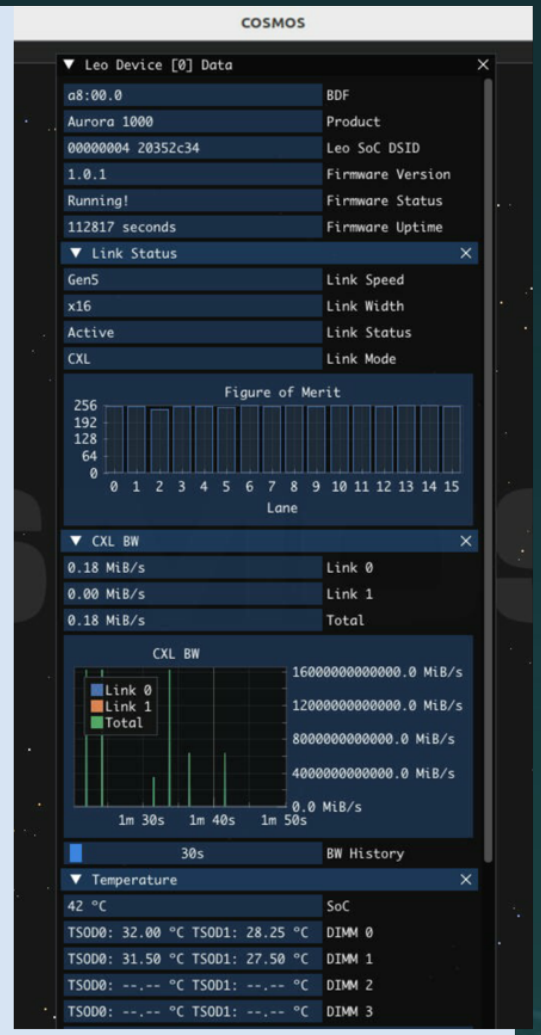
- Firmware Update via Inband and Out-Of-Band
- Performance/Temperature monitoring
- Memory component failure detection & recovery
- Hotness tracking

Security Management

- Key provisioning, secure boot/update/dbg/recovery
- Attestation & cert chain management

Memory Management

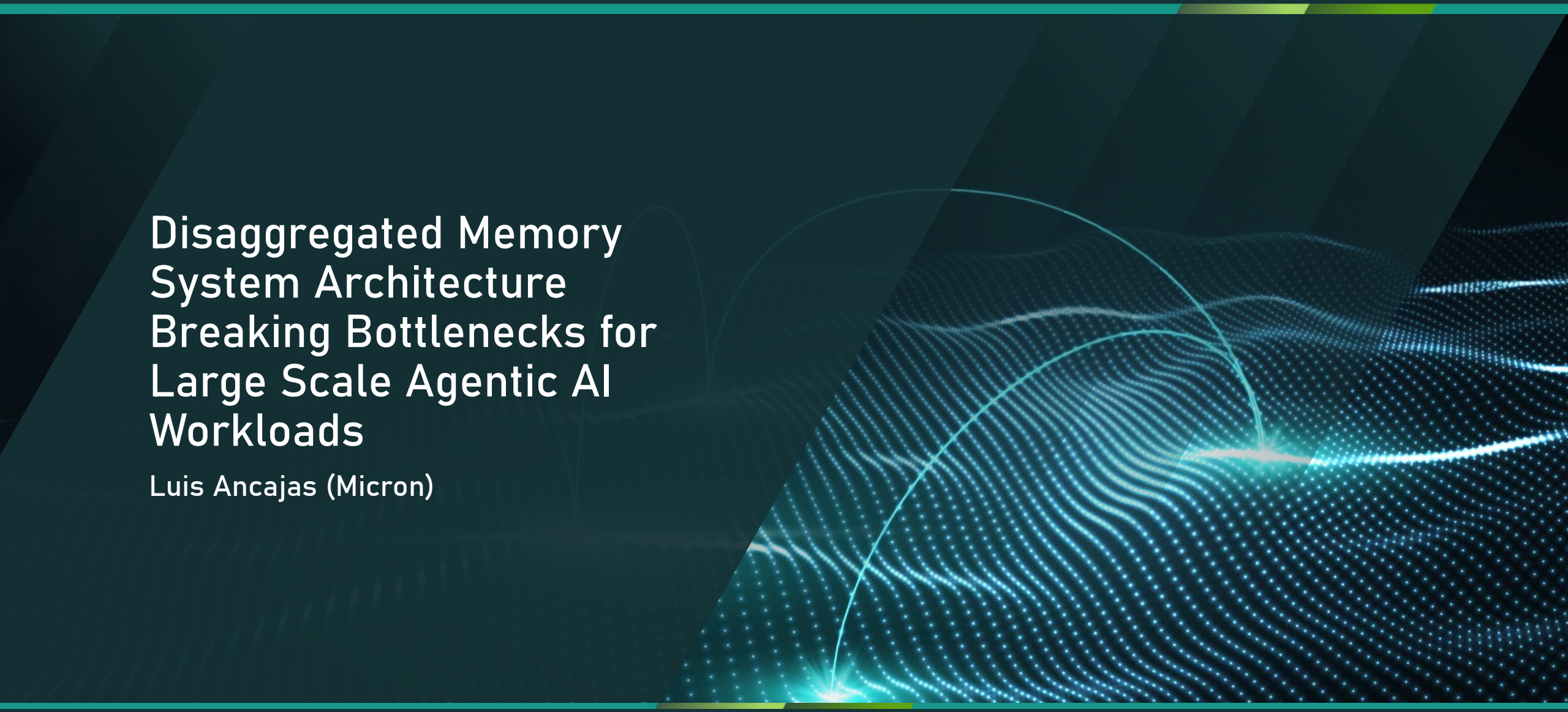
- Fabric management: Memory allocation and deallocation
- Performance monitoring with hot page tracker for faster page promotion and latency histogram
- Expanded error reporting
- Telemetry: DDR counters, rd/wr counters, Logging, Monitoring & host notification



Simplifying customers' interface for management of link, RAS and fleet capabilities

Disaggregated Memory System Architecture Breaking Bottlenecks for Large Scale Agentic AI Workloads

Luis Ancajas (Micron)



Disaggregated Memory Vision

Industry-leading JBOM capacities & performance for rack-scale shared memory

Disaggregated Memory: Memory contained within a separate chassis (JBOM) & decoupled from servers via CXL switching, enabling rack-scale shared memory

- Similar to: Disaggregated storage systems (NAS, SAN, JBODs)

Top 3 Target Workloads & Performance Improvements Using Less Power

- AI training, inference, prediction (KV cache with Dynamo): ~8x faster*
- Agentic large databases (SQL, NoSQL, GraphDB, in-memory): ~34x faster*
- Big data index: ~12x faster*

#1 Metric: Performance per Watt per $ (Hours → Minutes Using Less Power)

- Queries / second / Watt / $
- Tokens / second / Watt / $

How Disaggregated Memory Achieves This

- Faster disk-to-data round trip times
- Minimizes disk thrash & eliminates data sharding
- Increases TB per rack or server of accessible memory
- Integrates seamlessly with applications via Micron famfs file system

*Server configuration: 2-socket X86 sockets, 2TB Micron® DDR5 RDIMMs, 7x2TB + 1x20TB PCIe Gen5 SSDs; Liqid memory chassis: CXL 2.0 switch, 10x 1TB CXL 2.0 Smart Modular AIC modules (10TB); OS: Red Hat Linux 9.1 with *famfs* kernel updates

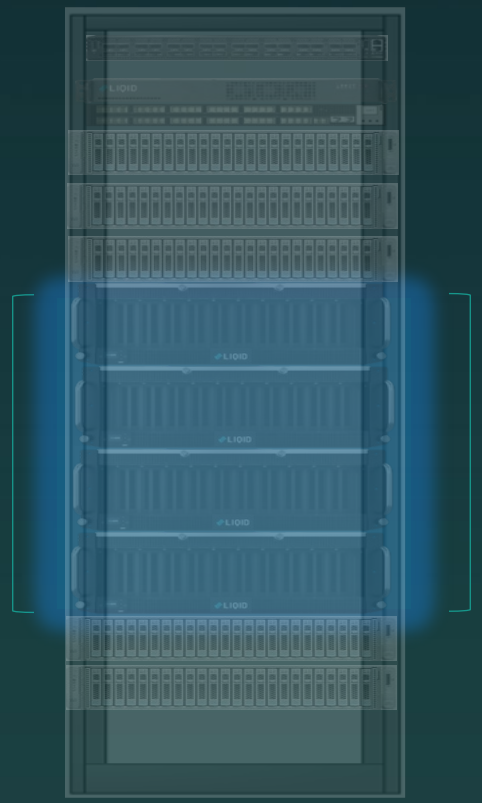
Abaco 3.0 Reference Architecture (DOE/PNNL program)

World's highest capacity shared memory system architecture for AI & HPC workloads

Goal: ***Independently scale memory*** like the industry already has for compute, storage, and networking. Micron is the preferred memory partner for this configuration.

Memory Footprint	Rack Capacity (4x 4U JBOMs)	JBOM Capacity (4U)	JBOM RDIMM Configuration	System Availability
Small	40TB	10TB	160x 64GB	CQ3-26
Medium	80TB	20TB	160x 128GB	CQ3-26
Large	160TB	40TB	160x 256GB	CQ4-26
Extra Large	320TB	80TB	160x 512GB	CQ4-27

All configurations based on Micron® DDR5 RDIMMs, Primemas® PCIe add-in cards (10x AICs per system, 16x RDIMMs per AIC), and Liqid® EX-5410C Memory Expansion Chassis (4U) with 4 PCIe ports x16, dual-socket CPUs with 16 memory channels, and 6000W chassis power limit (no thermal or cooling concerns since Micron reference architecture only needs ~3000W)



Abaco 3.0 Rack Specs

- **1x Fabric manager server host**
- **1x CXL switch**
- **5x AI GPU compute servers**
  - PCIe Gen5/6
  - Dual-socket CPU
  - CXL host adapter card
  - 4x GPUs
- **4x JBOMs**
  - Liqid® EX-5410C Memory Expansion Chassis (4U)
  - 10x Primemas® PMA Memory Expansion CXL Add-In Cards
  - 160x Micron® DDR5 RDIMMs
- **Target Applications***
  - KV cache for LLM inference
  - Large databases
  - Graph analytics
  - Vector RAG
  - Graph RAG

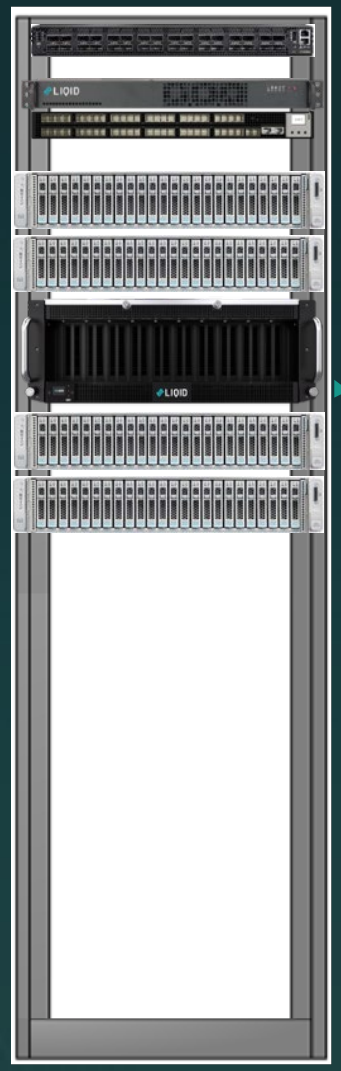
*Leveraging open-source Famfs (fabric-attached memory file system), contributed by Micron

Introducing Abaco-based benchmark experiment

Real world benchmark results based on production system and software and Industry leading LLMs

Hardware	
Host platform	AIC ODM server
CPU	Intel Xeon 6 processor
GPU	Nvidia A100 (2 per server)
Native HBM	40GB HBM2 / GPU
Native DRAM	Micron 128GB DDR5-5600MT/s RDIMMs Capacity: 2TB (16 RDIMMs)
Native Storage	Micron (7x) 7450 SSD/ 9450 SSD Capacity: 30TB
CXL memory chassis	Liqid Memory Chassis 10x Smart Modular Add-in cards 8x 128GB D5 RDIMMs / Add-in cards Total Bandwidth: 250GB/s (read bandwidth) Memory latency: 434 ns Capacity: 10TB

Software	
Applications	RocksDB, Pometry GraphDB, Milvus, Dynamo
OS	Red Hat Enterprise Linux
Kernel	6.14.11 – with weighted software interleaving-enabled Includes famfs kernel updates for seamless memory sharing across multiple hosts



KV Cache Benchmark Specifics	
LLM models	LLaMA3-8B -15GB Model Weights -128KB KV per token (fp16) GPT-OSS-20B -38GB Model Weights -48KB KV per token (fp16)
BM Tool	Nvidia AIPerf
Memory Tier Software	Nvidia Dynamo
Native Storage	Micron (7x) 7450 SSD/9450 SSD Capacity: 30TB
Inference Workload Description	Workload -Total Requests: 10,000 -concurrency = 64 -Duration: 10.2 min -Gen Length (OSL): 128 tokens

Disaggregated Memory

Famfs Linux kernel for memory sharing & pooling

Hardware and software abstraction layers

DB app server
famfs master
Famfs-enabled Linux

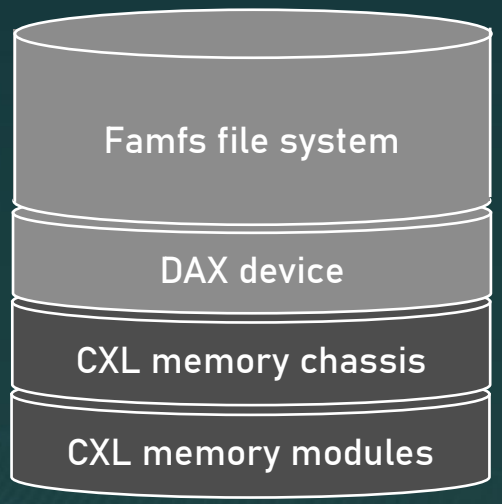
DB app server
famfs client
Famfs-enabled Linux

• • •

DB app server
famfs client
Famfs-enabled Linux

Pooled and shared memory as a file system

Famfs abstraction layers



5 Client: Mount DAX as a file system (famfs.mount)

4 Master: Create, mount DAX as a file system (famfs.mkfs, famfs.mnt, famfs.cp)

3 Host OS enumerates CXL memory as DAX device

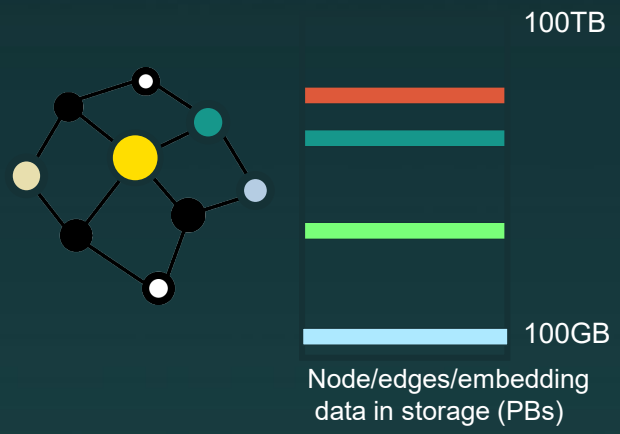
2 Memory chassis aggregates up to 10 modules (up to 40TB) into four host PCIe x16 lanes

1 CXL memory modules (256GB ea, 36GB/s)

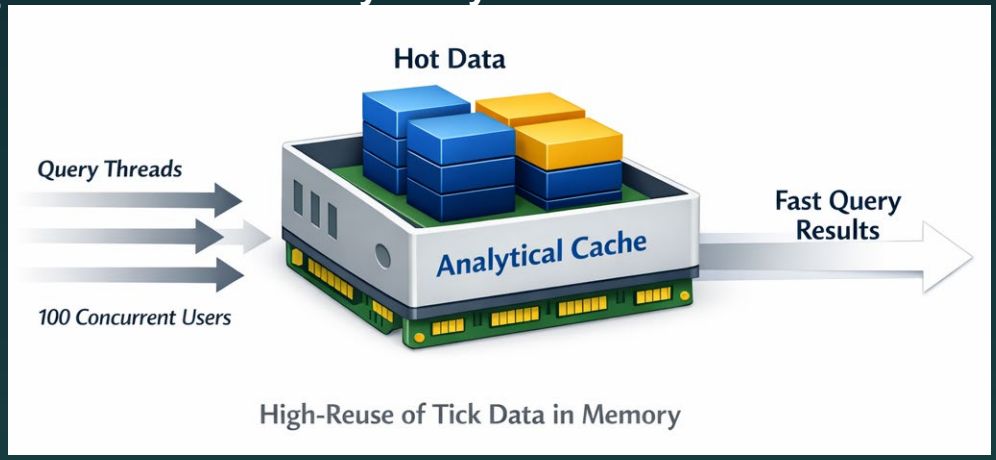
AI Workloads Studies using shared disaggregated memory

Leveraging CXL JBOM and famfs to enable large perf improvements

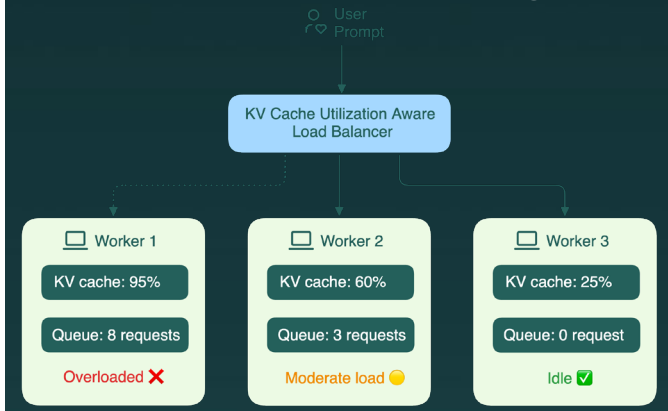
Graph analytics and neural networks



In-memory Analytical Cache



KV Cache LLM Inferencing



Proven: 32X Perf/$

Proven: 8X Perf/$

Estimated: 5.5X Perf/$

1. Server configuration: 2-socket x86 CPU, 16x128GB Micron® DDR5 RDIMMs (2TB), 7x2TB + 1x20TB PCIe Gen5 SSDs; CXL memory chassis: CXL 2.0 switch, 10x1TB CXL 2.0 AIC modules (10TB); OS: Red Hat Linux 9.1 with *famfs* kernel update

Case study 1: Baseline Time-Series Analytics Benchmark

Why Time-Series Analytics matters

- Represents real-world time-series analytics on financial tick data used for trading strategy development
- Evaluates end-to-end system performance:
  (database + compute + memory + storage)
- Designed to reflect behavior of large production datasets, even using smaller dataset

Key challenge

- Algorithms forces frequent storage access: data is not preloaded into memory and cache effects are periodically removed
- Performance is strongly impacted by memory residency and paging behavior
- Mixed workload profiles include both I/O-heavy and compute-heavy queries

Benchmark specifications	Description
Dataset model	Synthetic time-series tick database modeled on equities market data (e.g. trades and quotes), enabling controlled and repeatable scaling
Operational Mode	Baseline "storage-constrained" mode using a multi-terabyte dataset (3TB) designed to simulate larger datasets primarily residing on storage tiers
Query/Analytics patterns	Includes representative time-series analytics such as • Price extrema (year/quarter/month high bid) Volume-weighted pricing (VWAB/VWAP-style) • Market snapshot queries (random access) Statistical aggregation over time windows • Multi-user interval analytics and write operations
Concurrency model	Simulates multi-user workloads (1-100 concurrent request threads), including defined scaling tests for concurrency stress
Key Metrics	End-to-end query response time (submit -> full result)

Tick Analytics Performance Increases with Disaggregated Memory

Huge Breakthrough In Benchmark Results!

Server with 2TB of DRAM with 3TB Tick DB dataset in NVMe

Run time 46ms[1,2]

CXL JBOM
(10TB via 10 CXL AIC modules)

Up to 5 ports of PCIe x16 lanes

Server 2TB of DRAM with 3TB Tick DB in famfs

Run time 4ms[1,2]

1. Run time results (mean after 1000 runs) of the read and compute heavy benchmark.
2. Server configuration: 2-socket x86 CPU, 16x128GB Micron® DDR5 RDIMMs (2TB), 7x2TB + 1x20TB PCIe Gen5 SSDs; CXL memory chassis: CXL 2.0 switch, 10x1GB CXL 2.0 AIC (10TB); OS: Red Hat Linux 9.1 with *famfs* kernel updates

11.5X Perf

Read & Compute-Heavy Workload

3X or more in 13 of 15 tick analytics benchmarks

4.2X Perf/$

performance / cost

Reduction of TCO cost by 77%

5TB→100TB+

scale up of Tick DB sizes

Order of magnitude more DRAM per server enabled by CXL

Tick Analytics Improved Cost Efficiency (Scaling Up and Out)

Performance also scales linearly over servers

Server with 2TB of DRAM with 3TB Tick DB dataset in NVMe

→ 46 ms[1,2]

CXL JBOM
(5.5TB via 22 CXL modules)

Up to 5 ports of PCIe x16 lanes

→

Four servers with 3TB Tick DB dataset (1 shared copy) in famfs

Run times[1,2]
→ 4 ms
→ 4 ms
→ 4 ms[3]
→ 4 ms[3]

11.5X * 4 Perf ↑

Read & Compute-Heavy Workload

12X or more in 13 of 15 tick analytics benchmarks

8.0X Perf/$

Performance / cost

Reduction of TCO by 88% for the same performance

5TB→100TB+

larger index caching

Order of magnitude more DRAM per server enabled by CXL

1. Run time results (mean after 1000 runs) of the read and compute heavy benchmark.
2. Server configuration: 2-socket x86 CPU, 16x128GB Micron® DDR5 RDIMMs (2TB), 7x2TB + 1x20TB PCIe Gen5 SSDs; CXL memory chassis: CXL 2.0 switch, 10x1GB CXL 2.0 AIC (10TB); OS: Red Hat Linux 9.1 with *famfs* kernel updates
3. Two servers simultaneously running has been tested. Four servers configuration to be tested, but expect also linearity in performance.

Case study 2: LLM Inferencing with KV Cache

Why KV Cache in LLM Inferencing workload matters

- Agentic AI workloads will dominate the next wave of AI monetization
  - Transforms models from tools into systems that deliver complete, outcome driven tasks at scale
  - Agentic workflows will amplify pressure on memory and storage hierarchy through persistent long-lived context, iterative loops and reasoning, and multi-agent concurrency

Key challenge

- Agentic AI turns memory hierarchy– not compute– into the primary scaling bottleneck
- Memory demands grow linearly with context length and accumulates across time, and multiplies with mult-agent concurrencies, all to exceed memory capacities

Benchmark details	Description
LLM Model	Llama-3.1-70B (bf16), 4KB-16KB context size
Inference Workload	Total requests= 10K, Concurrency= 64, Duration: 10.2m, Gen Length (OSL): 128 tokens
Benchmark software	Nvidia AI Perf benchmark tool using Dynamo vLLM engine and KV Block Manager, and Micron famfs/Rapid sharing
Key Metrics	Tokens/$ comparing LLM benchmark results with and without Disaggregated Memory hierarchy

Disaggregated DRAM vs DRAM as KV Cache

Disaggregated memory achieves same Tokens/sec and adding scalability

Inference Server
(DRAM)

Run time
23.1 tokens/sec

Up to 4 ports of PCIe x16 lanes

Inference Server
(CXL JBOM)

Run time
23.0 tokens/sec

99% Tokens/sec

Performance parity

Achieve same tokens per second as local DRAM as KV Cache

90% TTFT

Latency parity

Across prompt sizes from 4K to 16K context size

No Memory Wall

Expandable, shareable KV Cache

To accommodate new AI workloads with growing context, more parameters, and agentic AI behaviors

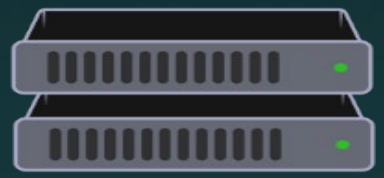
1. Run time results derived from two GNR servers with four H200 GPUs (two on each server). Memory sizes/server scaled to match Inf workload type of RAG/chat and LLM model Llama-3.1 70B (bf16) data size requirements and 4 users

LLM Inference Perf Analysis

Comparing Prefill Misses (Recompute) and Prefill Hits (KV Reuse)

100% KV recompute

Inference Server
(DRAM)



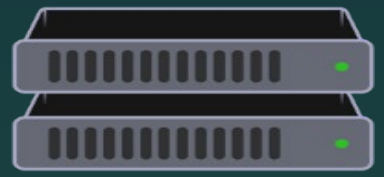
Performance
24 Tokens/sec

TTFT
343s

89% KV reuse

Up to 4 ports of PCIe x16 lanes

Inference Server
(DRAM+ DM DRAM)



Performance
160 Tokens/sec

TTFT
59s

1. Run time results derived from two GNR servers with four H200 GPUs. Memory sizes/server scaled to match Inf workload type of RAG/chat and LLM model Llama-3.1 70B (bf16) data size requirements and 4 users

6.7X

Performance Leap in Tokens/sec

At 89% KV Cache block reuse rate

2.7X

Less time for 1st token (TTFT)

At 89% KV Cache block reuse rate

Shared Mem

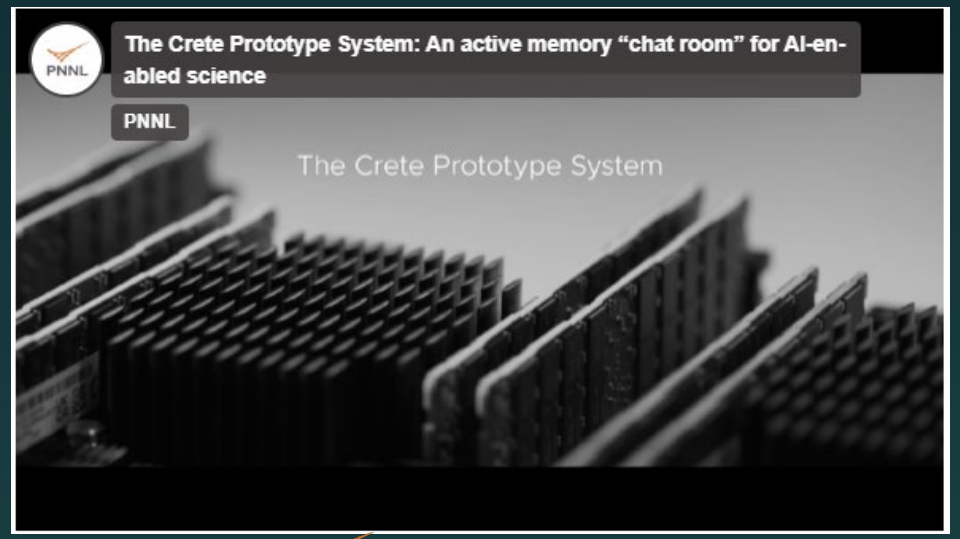
GPUs sharing DM for KV cache blocks

Each server sharing other server's KV Cache

From vision to reality

Disaggregated CXL memory breakthrough from Micron and DOE/PNNL Paving The Way to Production

Crete 2.0 prototype system designed and built by Micron for PNNL, delivered in June 2025



PNNL

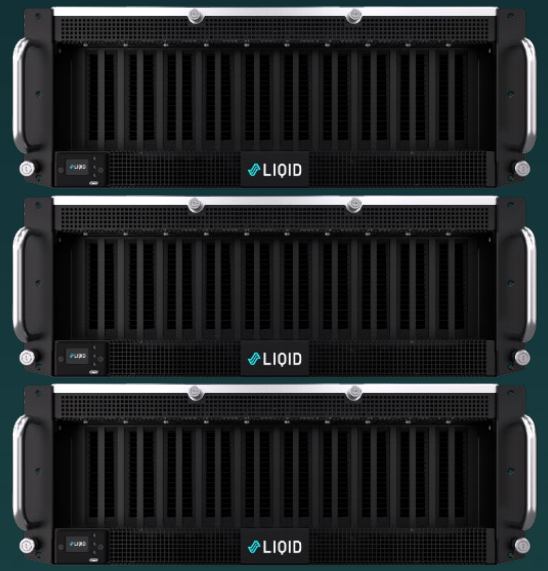
US Government Labs Partner

XCONN TECH

Foundational CXL Switch silicon

From 15TB to 100TB+ in a rack

Abaco 3.0 – A commercially available production system in Q4'26



LIQID

Foundational CXL chassis and fabric manager software

Primemas
Hublet® Pioneer

Foundational CXL PCIe Add-in-card

Questions?

Thank You

Visit the CXL Consortium Kiosk (Booth No. 725) to learn more about CXL solutions available today!