SERVER:
COMPOSABLE MEMORY
SYSTEMS (CMS)



XPU Scale Composable Smart Memory Fabric

Gaurav Agarwal *Senior Distinguished Engineer, Marvell*

Satananda Burla *Distinguished Engineer, Marvell*

OCP PLATINUM® Member

© 2026 Marvell. All rights reserved.

Forward-looking statements

Except for statements of historical fact, this presentation contains forward-looking statements (within the meaning of the federal securities laws) including statements related to future revenue, future earnings, and the success of our product releases that involve risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “projects,” “believes,” “seeks,” “estimates,” “can,” “may,” “will,” “would” and similar expressions identify such forward-looking statements. These statements are not guarantees of results and should not be considered as an indication of future activity or future performance. Actual events or results may differ materially from those described in this presentation due to a number of risks and uncertainties.

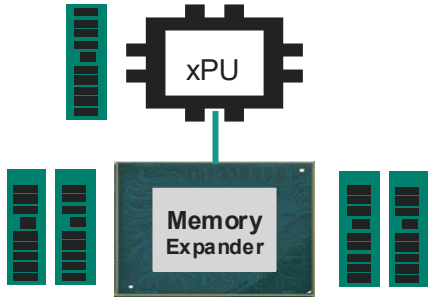
Forward-looking statements are only predictions and are subject to risks, uncertainties and assumptions that are difficult to predict, including those described in the “Risk Factors” section of our Annual Reports on Form 10-K, Quarterly Reports on Form 10-Q and other documents filed by us from time to time with the SEC. Forward-looking statements speak only as of the date they are made. You are cautioned not to put undue reliance on forward-looking statements, and no person assumes any obligation to update or revise any such forward-looking statements, whether as a result of new information, future events or otherwise.

© 2026 Marvell. All rights reserved.

Composable memory solutions

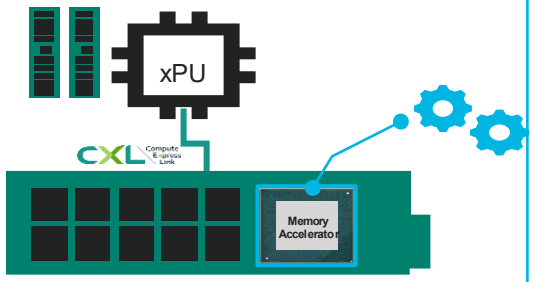
Direct Attach

Structera™ X



- Scale performance and capacity

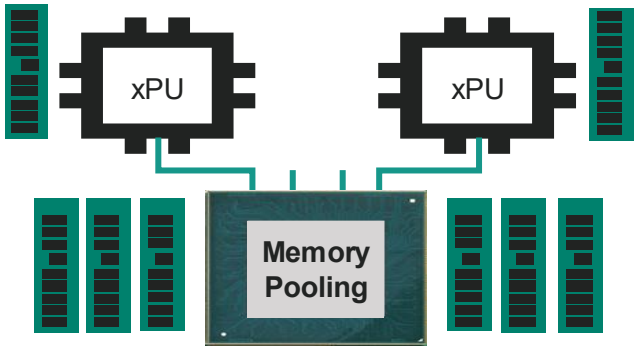
Structera A



- Memory accelerator

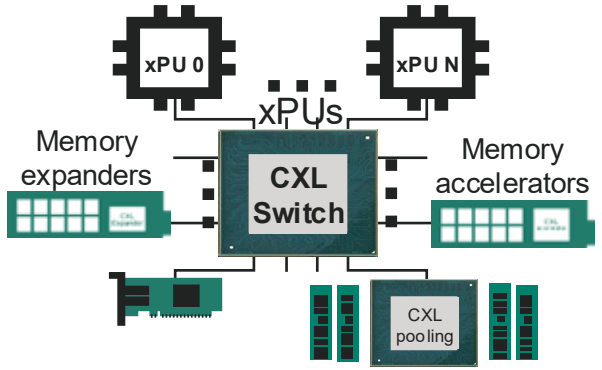
Disaggregated Server

Structera P



- Share memory across xPU

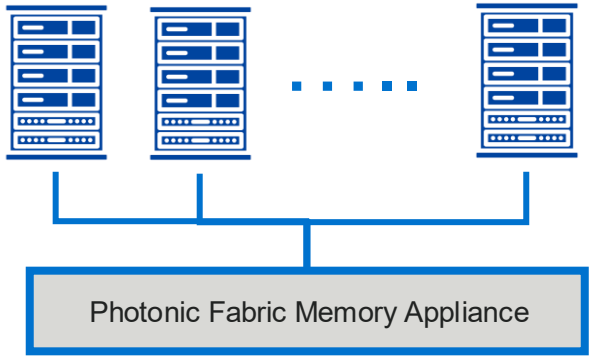
Structera S



- Rack scale composability

Disaggragted Rack

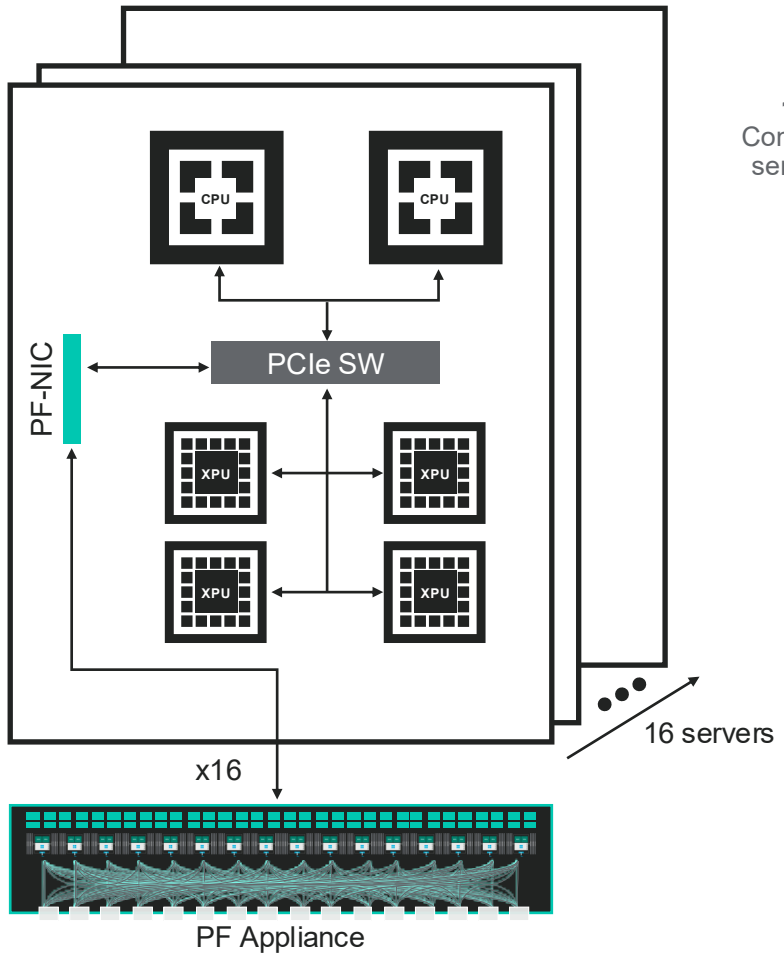
Photonic Fabric™ Memory Appliance

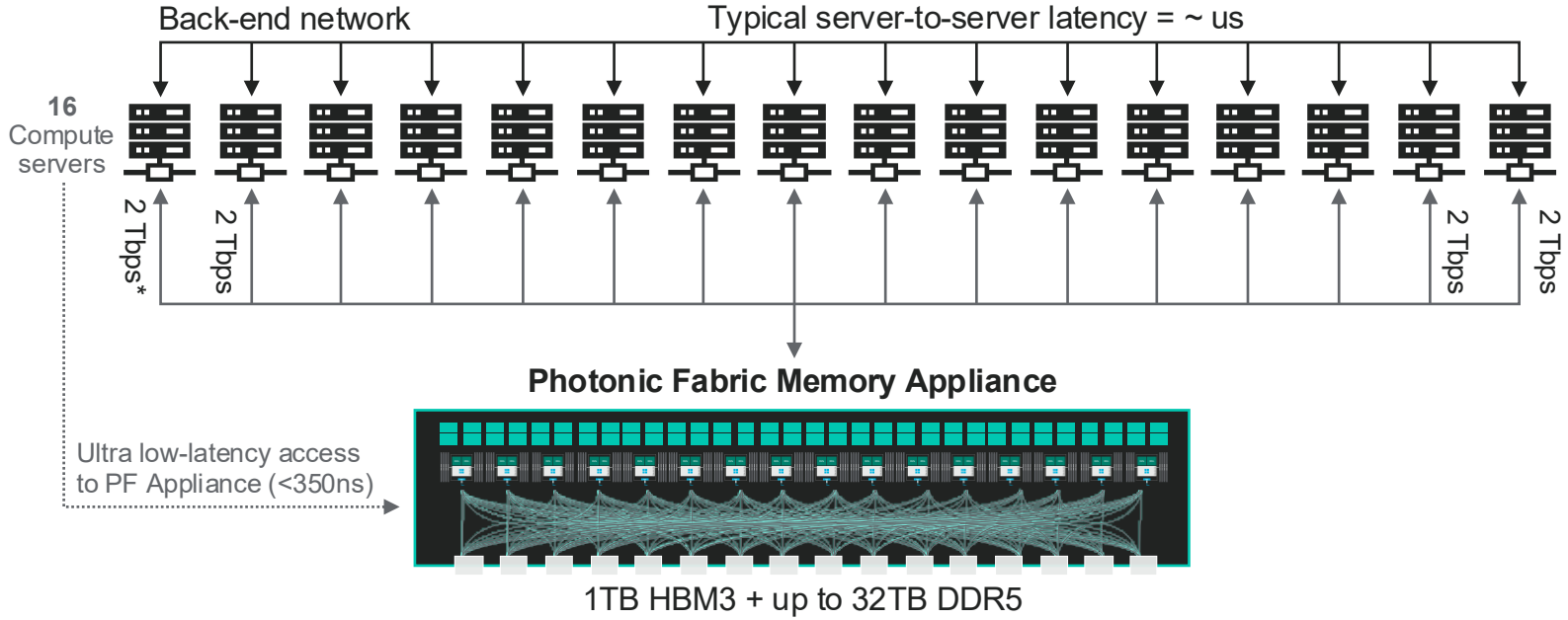


- Multi-Rack/Pod-Scale optically connected shared memory
- Unified memory for RAG/Agentic AI in CPU/XPU clusters

© 2026 Marvell. All rights reserved.

Photonic Fabric Memory Appliance





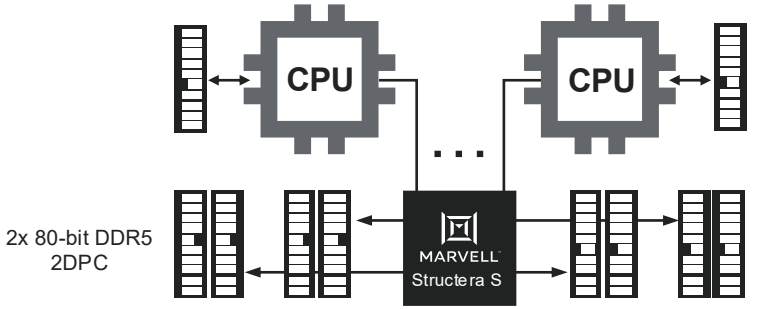
Rack mountable, pod-scale memory appliance, **supports** CXL 3.1 (incl. CXL DCD)	Connects 16 compute servers **or** 16 XPUs or **combination** with one PFMA
Memory capacity: 1TB HBM3e + 128x **standard** DDR5 DIMMs (16/32TB)	Unified shared address space with DCD

© 2026 Marvell. All rights reserved.

Structera S : CXL switch for rack scale pooling

Structera S
CXL switches





PCIe **5.0 16x16 | 32x8** CXL **2.0**	**2TB/s** Switching capacity	**Sub-100ns** Port-to-port latency	**Memory pooling and sharing**
1.6nm	**Intel and AMD** Server CPUs	**135W**	Up to **100TB** CXL memory pool

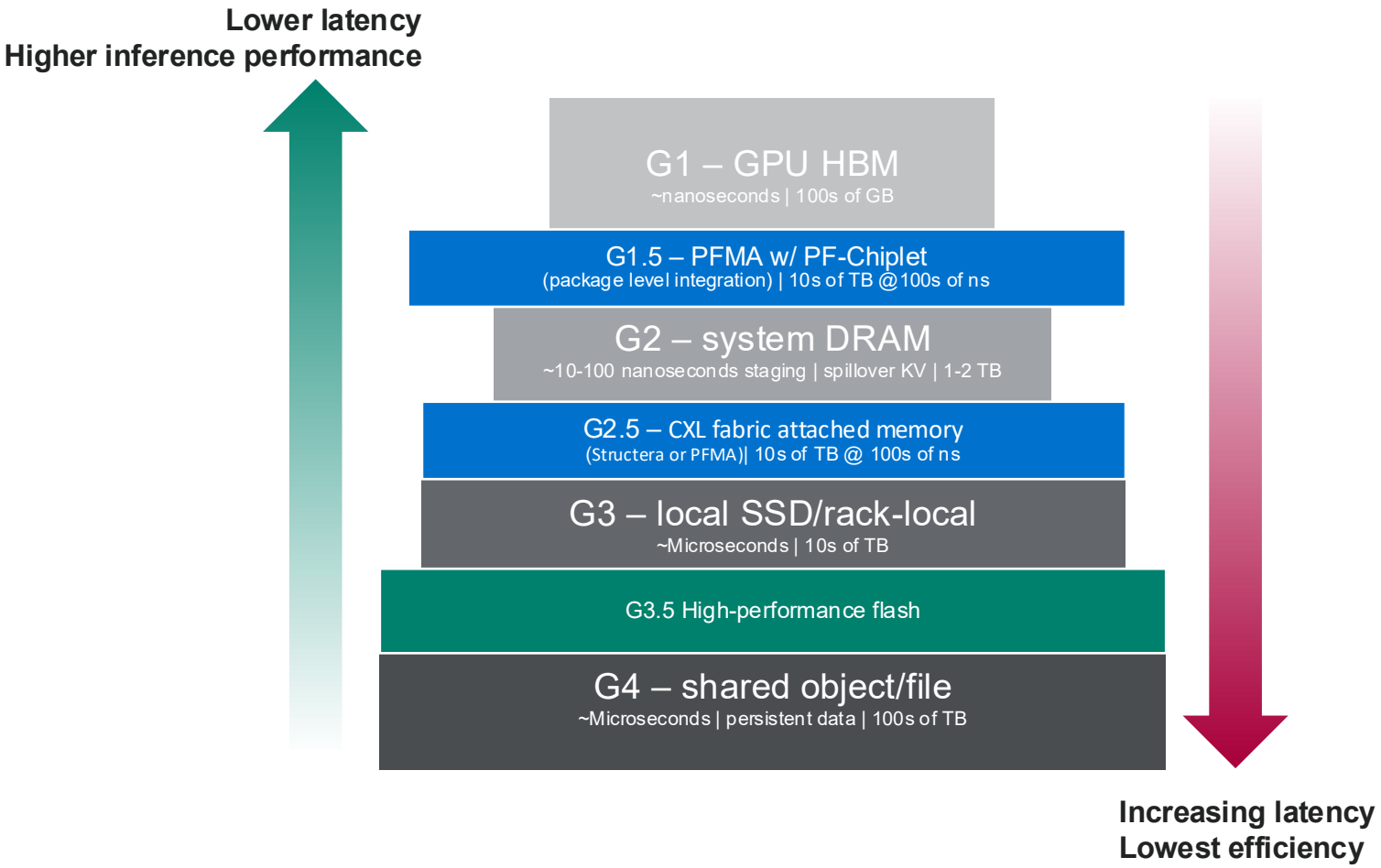
Memory pooling | **Peer-to-peer I/O** | **Ultra IO Transformer**

OCP GLOBAL SUMMIT | 2026

Scaling Innovation for the AI Era.

© 2026 Marvell. All rights reserved.

Memory tiers for KV$ offloading (updated)



© 2026 Marvell. All rights reserved.

XPU scale smart memory fabric

Compute cluster

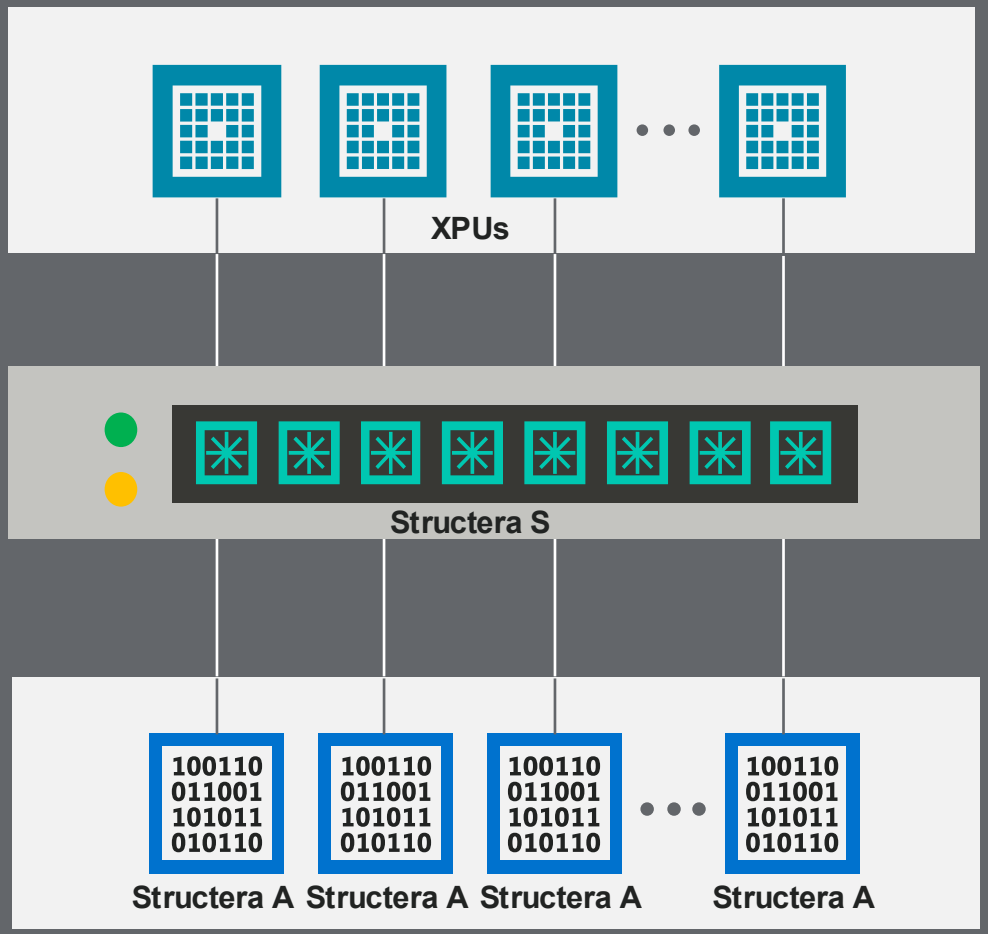
- Access AI data objects in disaggregated memory via API

High-performance data-path

- Peer-to-peer PCIe
- UIOT: PCIe/CXL hybrid switching

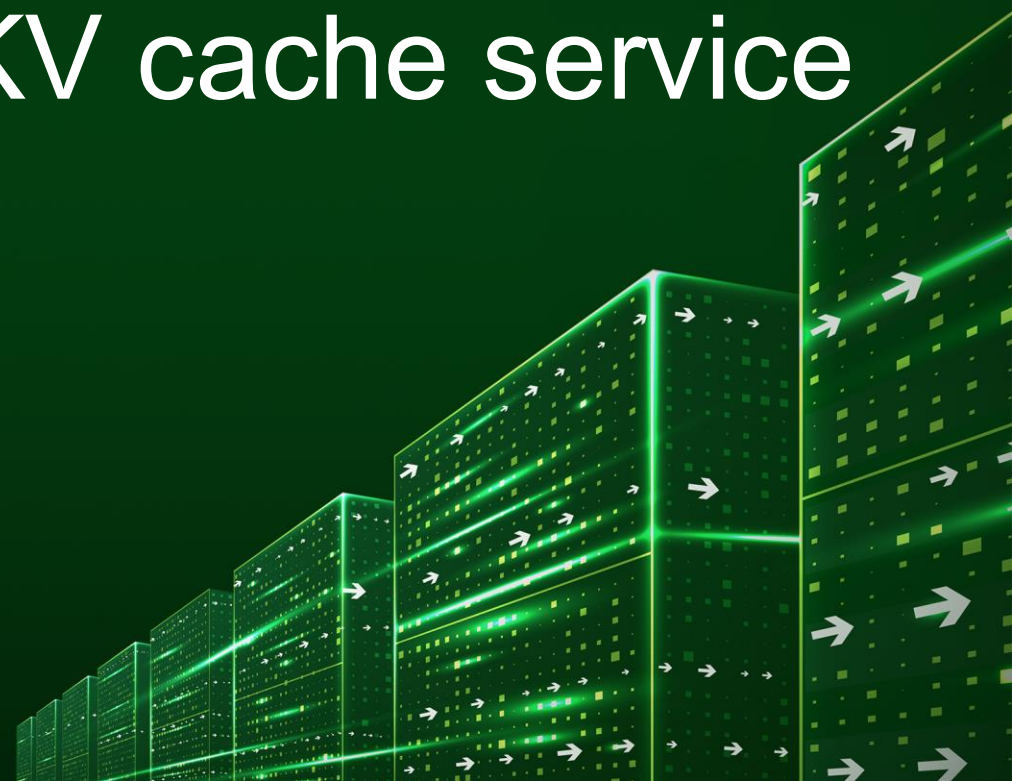
Disaggregated memory w/ compute

- Native AI objects aware
- Provides data as a service



© 2026 Marvell. All rights reserved.

Offloaded Data Services: KV cache service

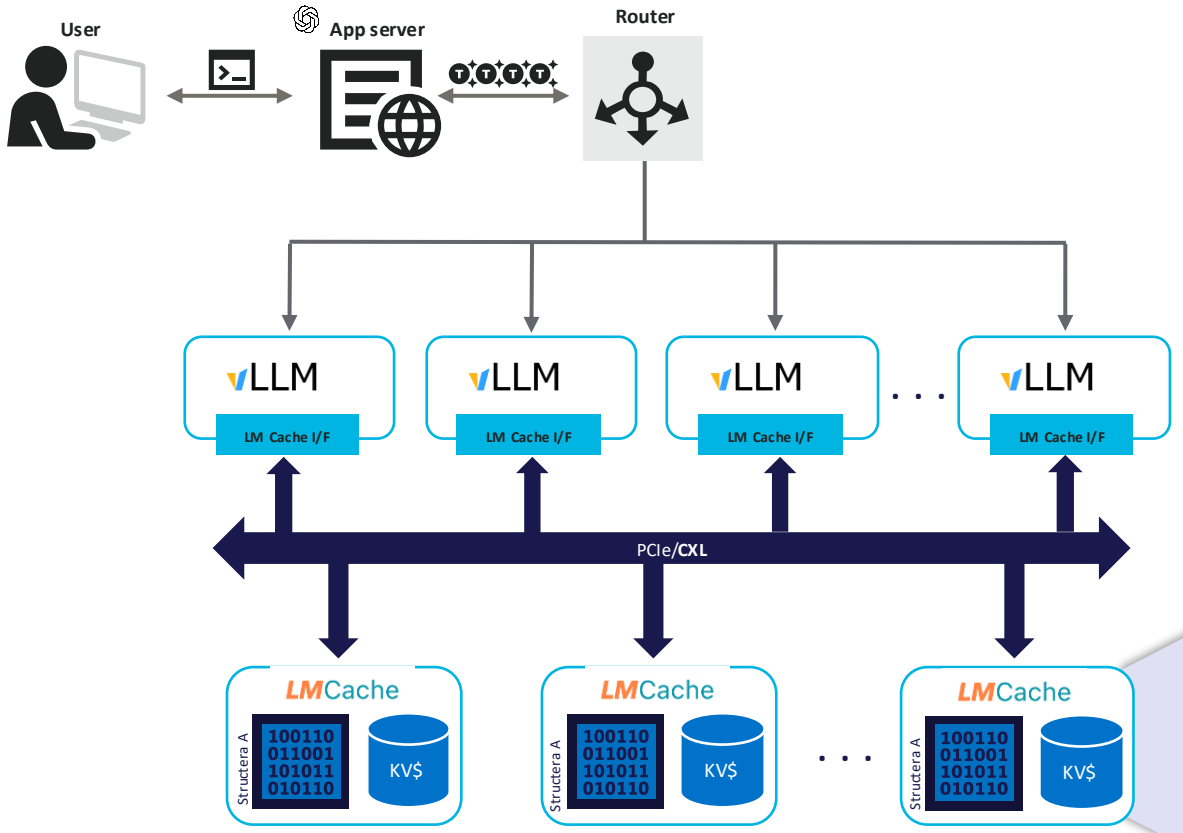


OCP GLOBAL SUMMIT | 2026

© 2026 Marvell. All rights reserved.

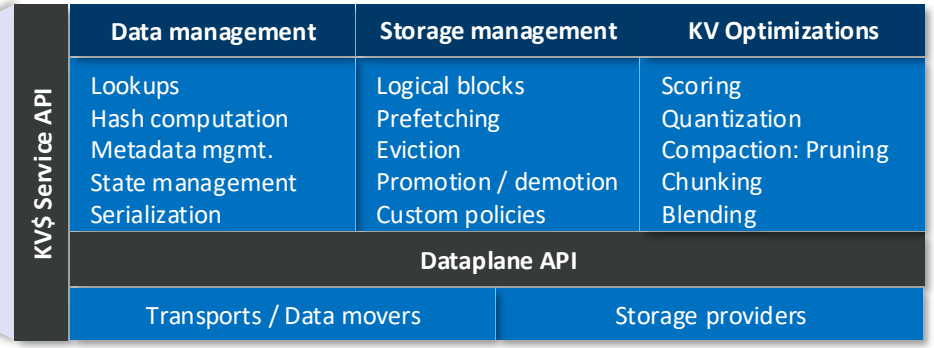
KV cache service offload to smart memory fabric

Agentic AI Inferencing System



Highlights

- Complete KV cache service offload to the fabric
- Memory efficient shared model
- Store once read multiple
- Cache tier scales independent of compute tier
- Cache availability is independent of compute
- No synchronization with compute overhead
- Simple load-based routing – avoids complexity
- Equidistant to compute – predictable performance
- KV$ optimizations naturally fits Near Memory Compute

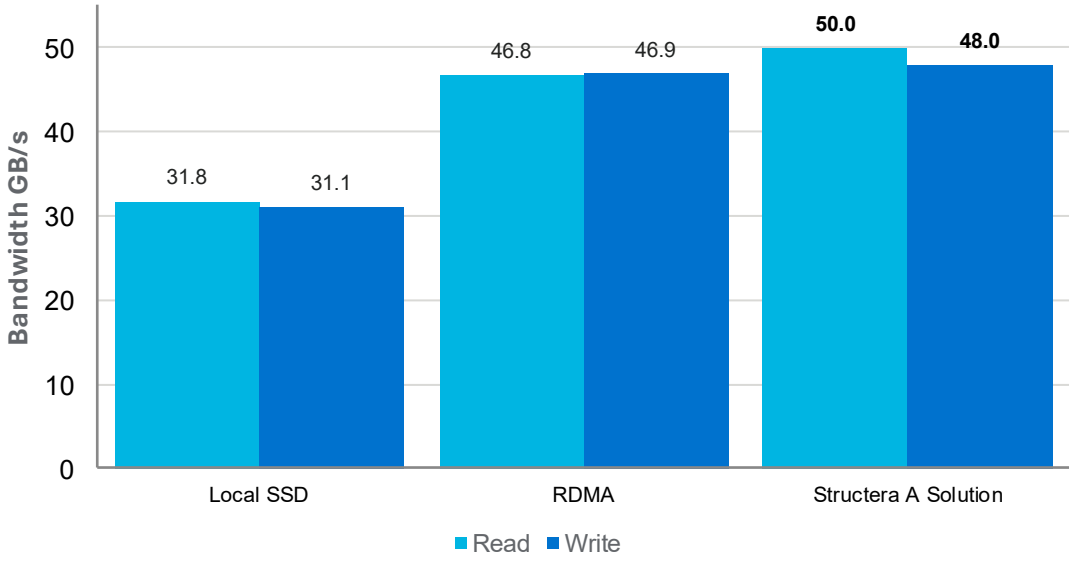


© 2026 Marvell. All rights reserved.

KV$ service offload performance (early results)

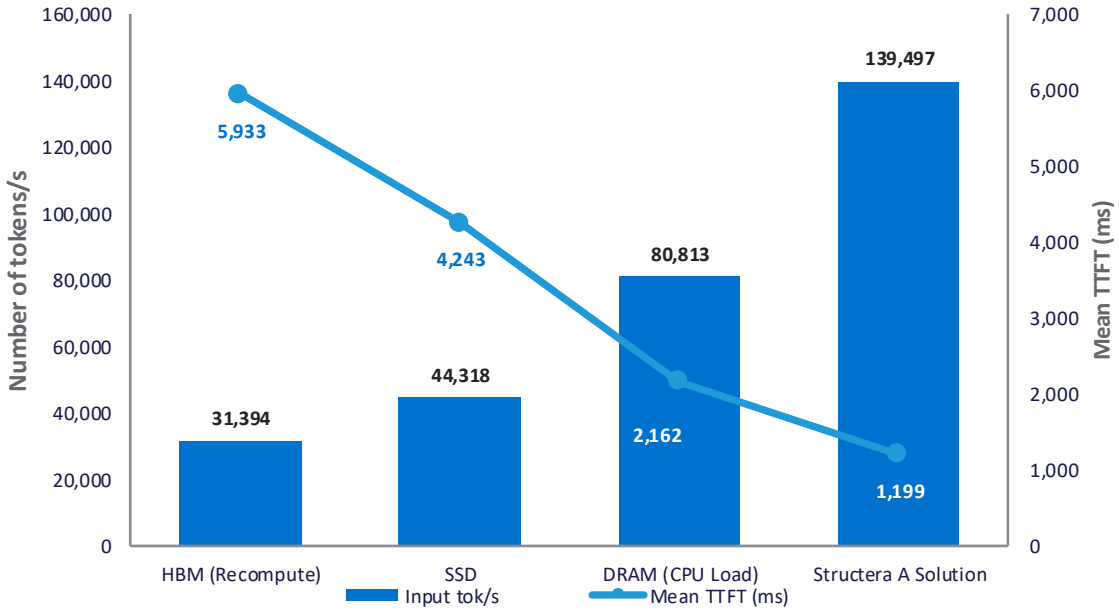
Raw transport bandwidth (GB/s)

read & write · higher is better



End-to-end inference throughput

input tok/s · higher is better · TTFT · lower is better



TTFT: Time To First Token (inference latency)

5x

lower TTFT → 5.9 to 1.2 s

4x

higher throughput → 31k tok/s to 139k tok/s

17x

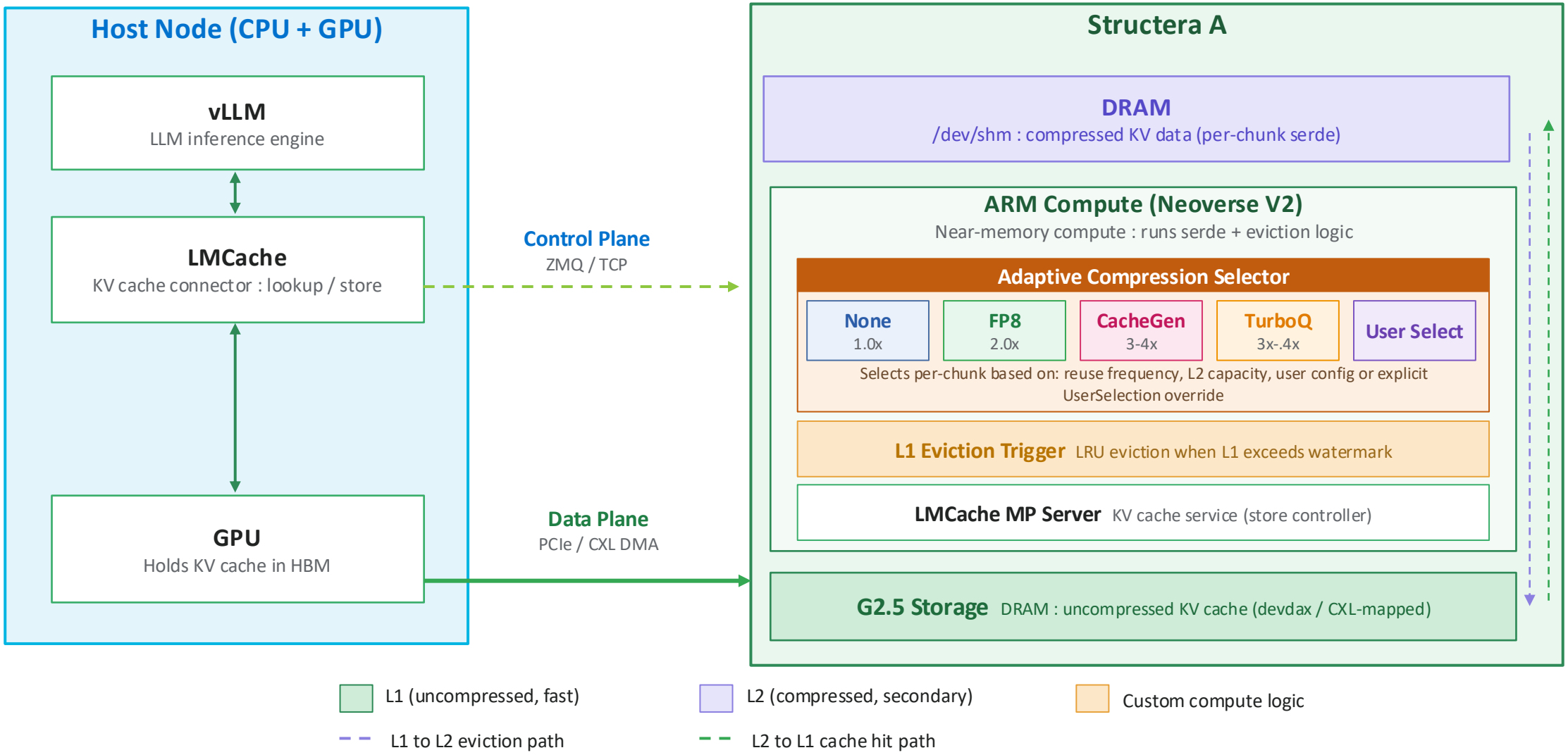
lower read latency vs Local SSD
52us → 3us

lmbench long-doc-qa · Peak users 16 · Llama-3.1-8B · H200

Source: Marvell

© 2026 Marvell. All rights reserved.

Algorithmic optimizations: KV$ compression



© 2026 Marvell. All rights reserved.

KV$ compression offload performance

Compression Method	Mean TTFT ↓	Input tok/s ↑	L2 Size Occupied ↓	Compression Ratio ↑	KV Cache Hit Rate ↑	ARM Avg Cores ↓	Accuracy Loss ↓
None (raw BF16)	4864 ms	16,832	8000 MB	**1.0x**	35.1%	0.5	0
FP8 (float8_e4m3fn)	**3374 ms**	**22,860**	3328 MB	**2.0x**	33.4%	2	**1.52**
TurboQuant k8v4	2914 ms	**22,845**	1884 MB	**2.61x**	46.5%	**7.4**	3.9
CacheGen*	2834 ms	25,594	278 MB	5.6x	34.8%	0.6	**13.4**

Benchmark Environment:
Model Used: meta-llama/Llama-3.1-8B-Instruct
Dtype: torch.bfloat16
Workload: lmbench long-doc-qa
4 documents x 64K-token context x queried 10 rounds (tile-shuffled)
4 x in-flight concurrently
Hardware: 1xH200(27% mem_util), 1xStructeraA (32GB + 32GB)
Software: vLLM + LMCache, LMCache MP Server

G2.5 Storage: DDR size 6GB
G3 Storage: Secondary storage for storing compressed data 12GB

Note:
↓: Lower the better ↑: Higher the better
CacheGen*: Implementation incomplete
Accuracy Loss: Cosine Similarity Angle (0* equal, 90* all not equal)
ARM Avg Cores: Used ARM Neoverse V2 out of 16
TTFT: Time To First Token (Inference Latency)
tok/s : Input Tokens processed per second

Source: Marvell

© 2026 Marvell. All rights reserved.

Data Transfer Across Fabric



OCP GLOBAL SUMMIT | 2026

© 2026 Marvell. All rights reserved.

Structera ODM for GPUDirect transfers

Key Insight: GPU-facing traffic routes over PCIe P2P DMA (no CXL needed on GPU side).

PCIe P2P DMA

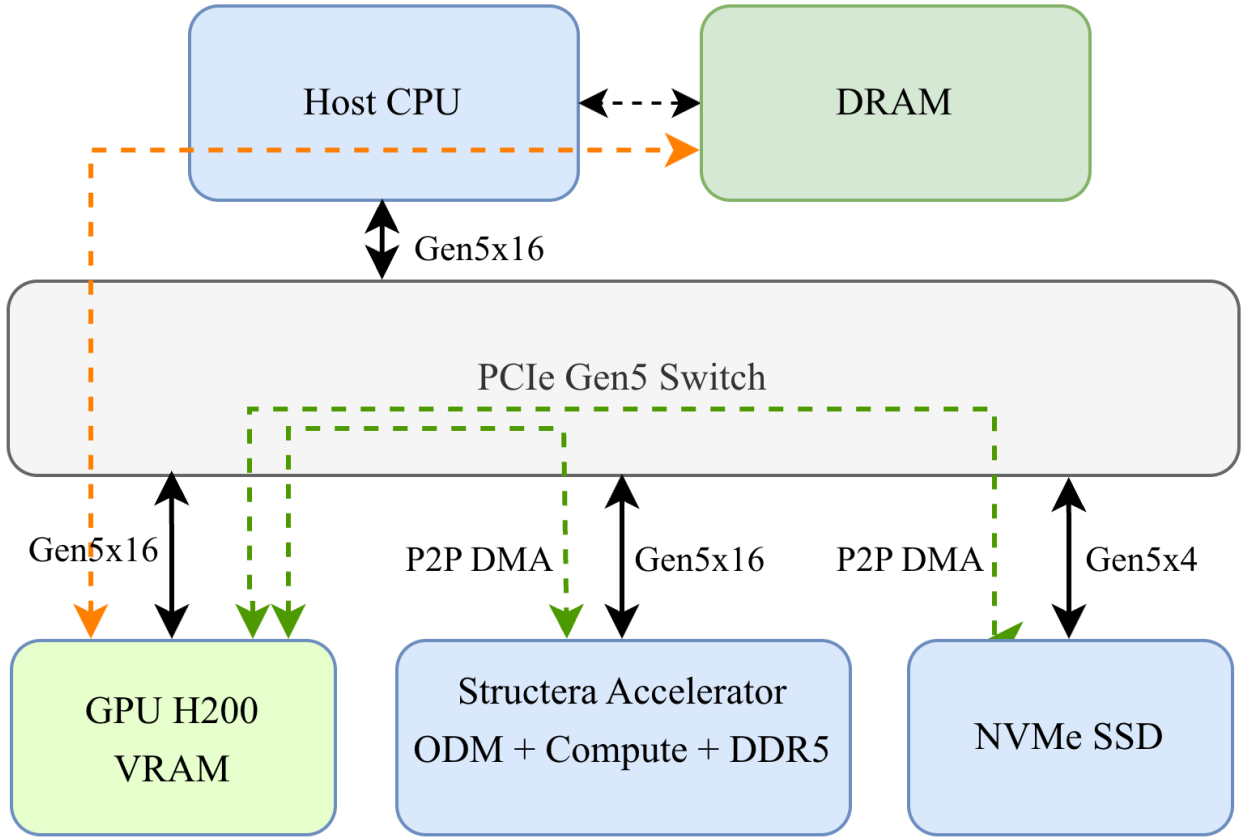
Expander DMA controller initiates transfers directly against GPU HBM — GPU is passive

Full Host Bypass

Data never touches host DRAM or traverses host IOMMU — pure switch-domain transfer

Framework Compatible

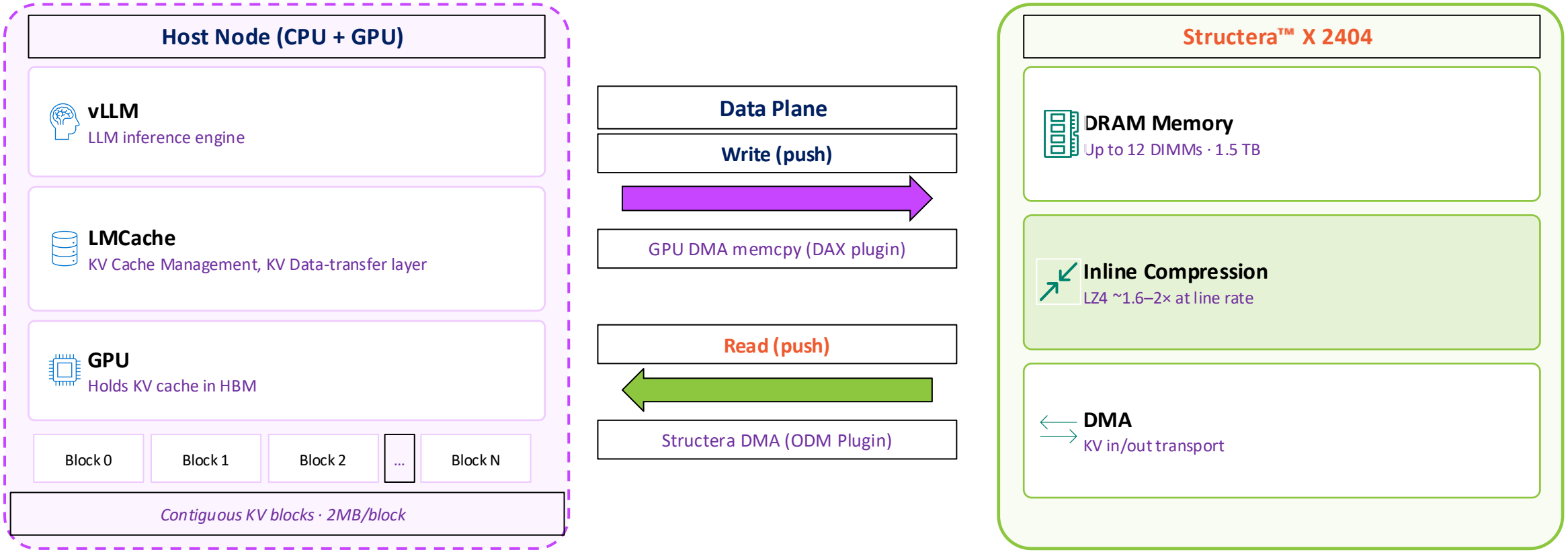
Integrates via a custom NIXL backend plugin — frameworks using NIXL require no architectural changes



© 2026 Marvell. All rights reserved.

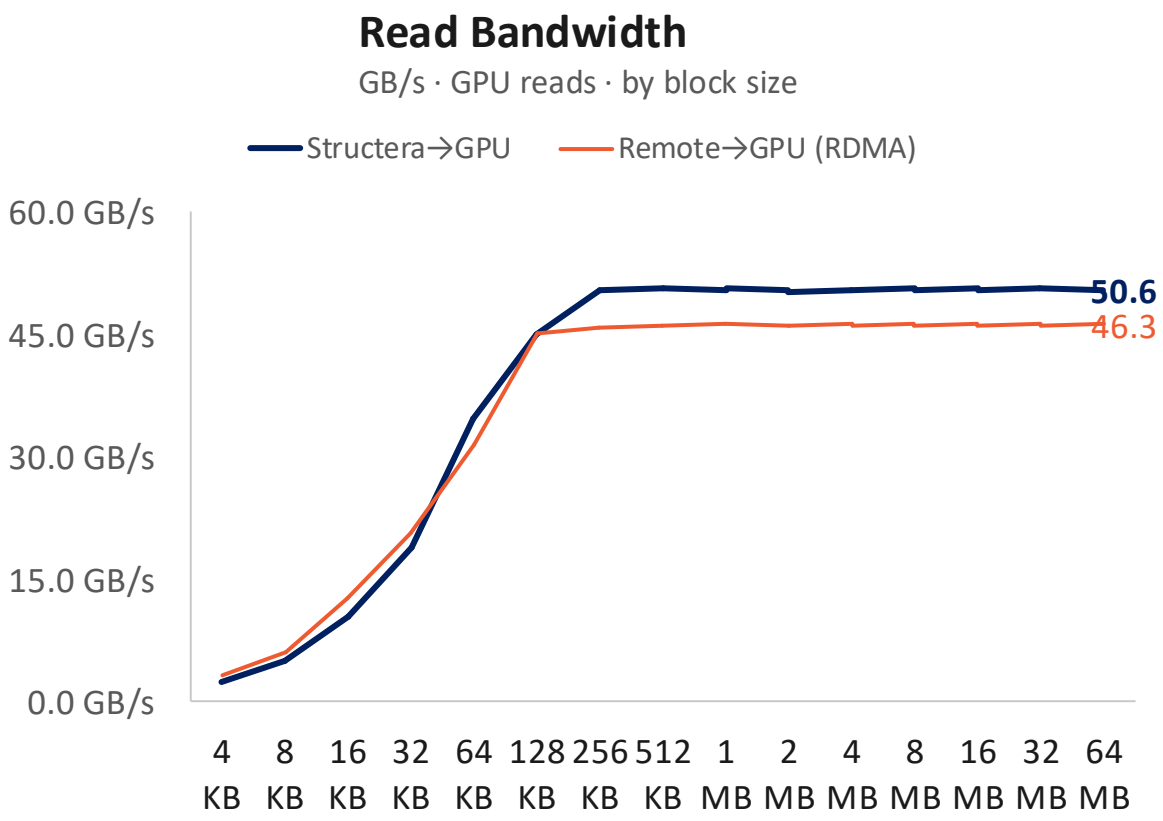
KV cache transfer: Bidirectional data path

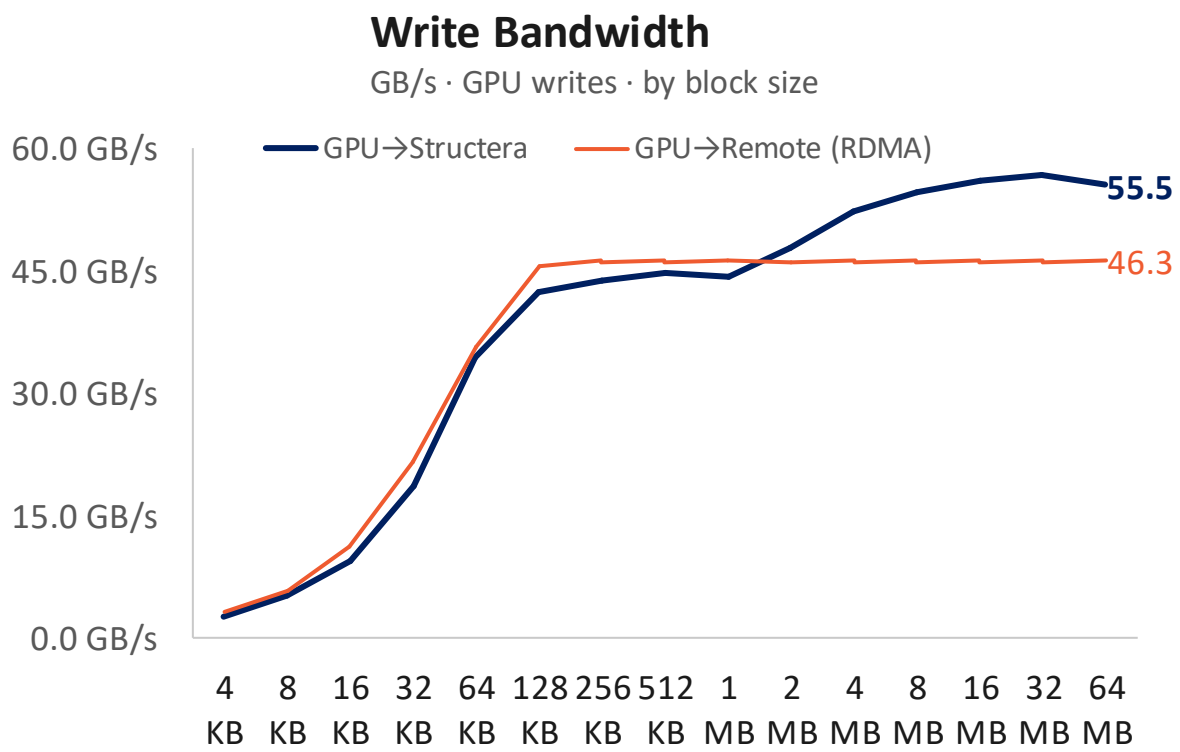
Write: GPU DMA DAX plugin · Read: Structera DMA ODM plugin



© 2026 Marvell. All rights reserved.

Nixlbench bandwidth: RDMA vs Structera

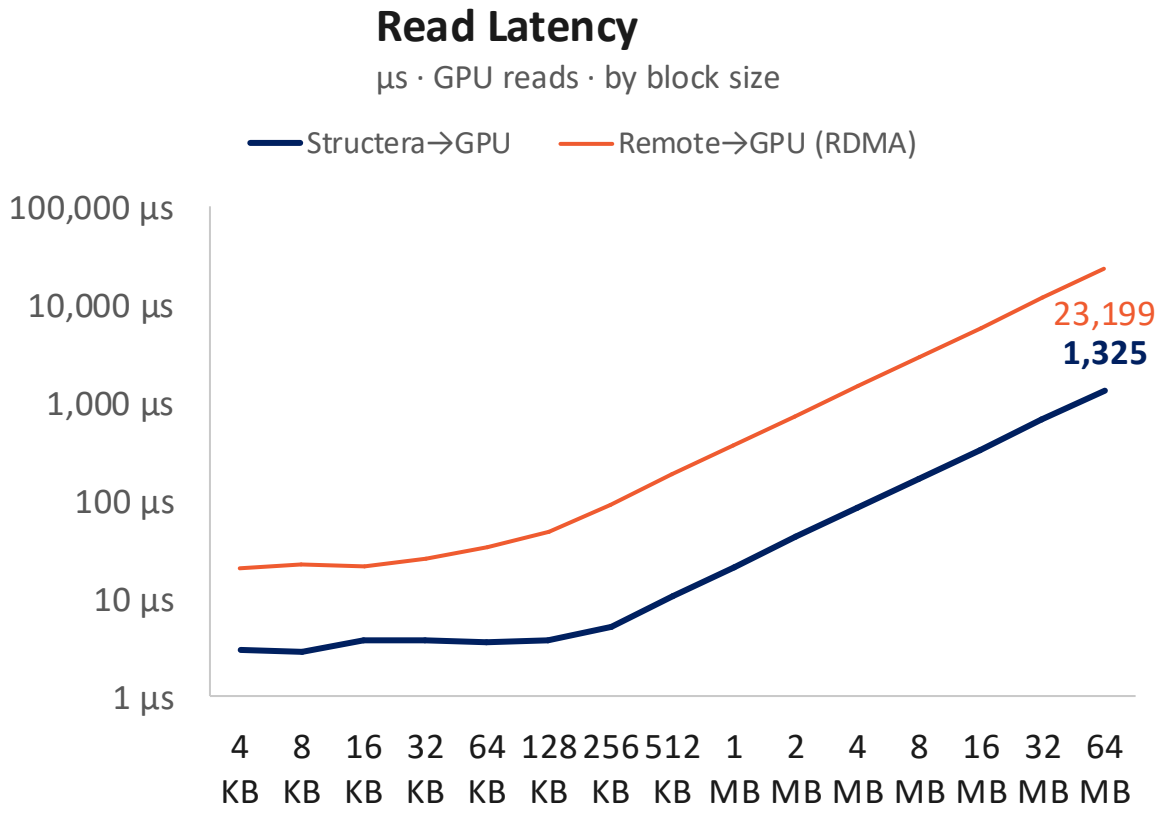


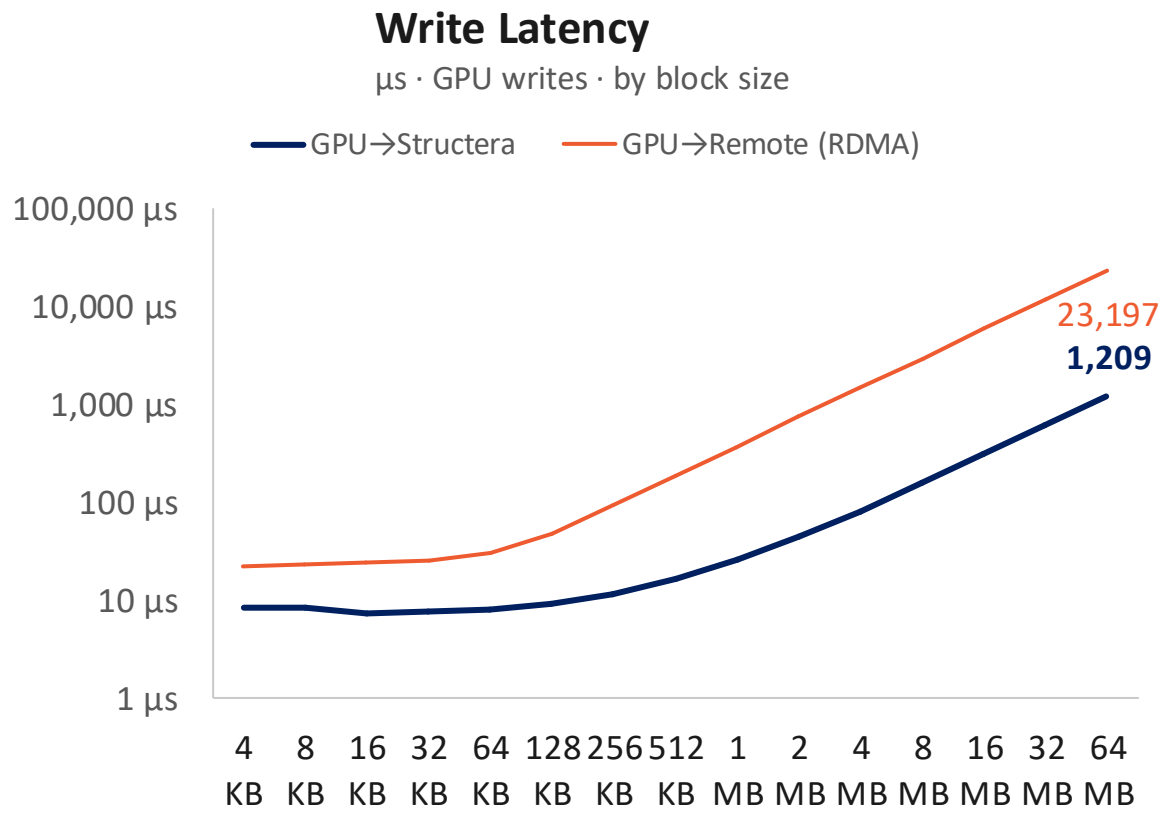


Source: Marvell

© 2026 Marvell. All rights reserved.

Nixlbench latency: RDMA vs Structera





https://www.hotstorage.org/2026/accepted.html#

22. CXL Memory for LLM Inference Staging via an On-Device DMA Controller

Veerasenareddy Burru, Satananda Burla, Pradeep Kumar Nalla, Alok Prasad (Marvell Technology Inc.)

Source: Marvell

© 2026 Marvell. All rights reserved.

Call to Action

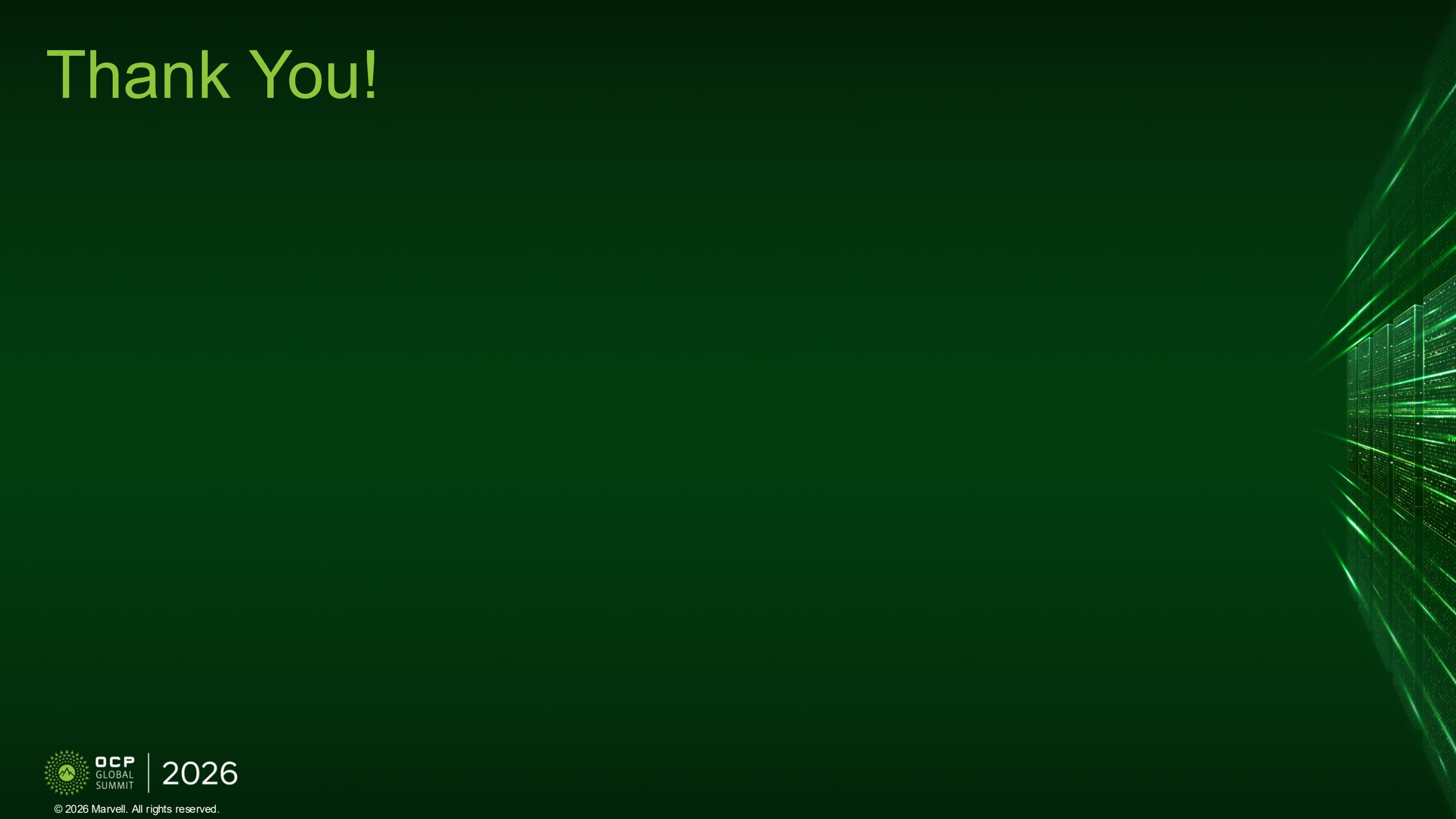
- CMS and CMS/CxP working with FTI DCC, Startups, and SNIA
  - Exploring open-formats based stable API for computational devices that aligns to AI frameworks and middleware
  - Explore a unified data-service API providing a consistent interface for XPU and AI framework for AI-native data services
- Academia
  - Advance Nead Data Processing architectures, open prototypes, and workload-driven benchmarks
- Hyperscalers
  - Co-design and validate memory-fabric solutions against real-world AI serving workloads
- Resources

  https://www.opencompute.org/projects/composable-memory-system

  https://www.opencompute.org/projects/data-centric-computing

© 2026 Marvell. All rights reserved.

Thank You!



OCP GLOBAL SUMMIT | 2026

© 2026 Marvell. All rights reserved.

Early benchmarking: workload

64,000 tokens

`--ldqa-document-length`

Each synthetic document is ~64K tokens long

20 rounds

`--ldqa-query-per-document`

Each document is queried 20 times (repeated Q&A over the same long context)

24 GB

`--kv-cache-volume`

Total KV data generated — about 3 documents of 64K tokens each (at 8,192 tokens/GB)

4

`--ldqa-num-inflight-requests`

4 concurrent users hitting the engine simultaneously

tile

`--ldqa-shuffle-policy`

Interleaves requests across documents in a tile pattern (not pure round-robin, not fully random)

8,192

`--tokens-per-gb-kvcache`

Conversion factor used to size the working set

Benchmark: **`lmcache bench`**

Workload: A **shared long-document Q&A** scenario (a small set of large documents queried repeatedly by multiple concurrent users)

Source: Marvell

© 2026 Marvell. All rights reserved.

Early benchmarking: configuration

TEST SYSTEM CONFIGURATION

HOST

GPU: 1xNVIDIA H200 NVL

CXL mapping: /dev/dax0.0, 32 GiB devdax window

CUDA: CUDA 13 driver

Boot param: iommu=pt

ARM

NMC: 16x aarch64 Neoverse-V2

CXL mapping: /dev/mem at physical base 0x800000000 (same CXL window)

Software: torch CPU + NO_GPU_EXT, LMCache Service on ARM Cores

SHARED FABRIC

CXL window: 32 GiB shared between host and ARM

Inference engine: vLLM 0.22 + LMCache (Custom CXL connector)

Connector: LMCacheMPConnector (Custom CXL transfer context)

vLLM INFERENCE ENGINE CONFIGURATION

CORE

Model: Meta Llama-3.1-8B-Instruct BF16

Inference engine: vLLM (--enforce-eager)

CONTEXT & ATTENTION

Max context: --max-model-len 80000 tokens

Attention: FlashAttention (Grouped Query Attention, 32:8)

Chunk size: 512 tokens

KV & MEMORY

GPU KV budget: --gpu-memory-utilization 0.26 — small, so KV overflows GPU and forces offload

Prefix caching: --no-enable-prefix-caching (LMCache manages caching)

KV connector role: kv_both (store + retrieve)

External memory: 32 GB (offload)

Source: Marvell

© 2026 Marvell. All rights reserved.