SAMSUNG

Heterogeneous Memory Opportunity

with Agentic AI and Memory Centric Computing

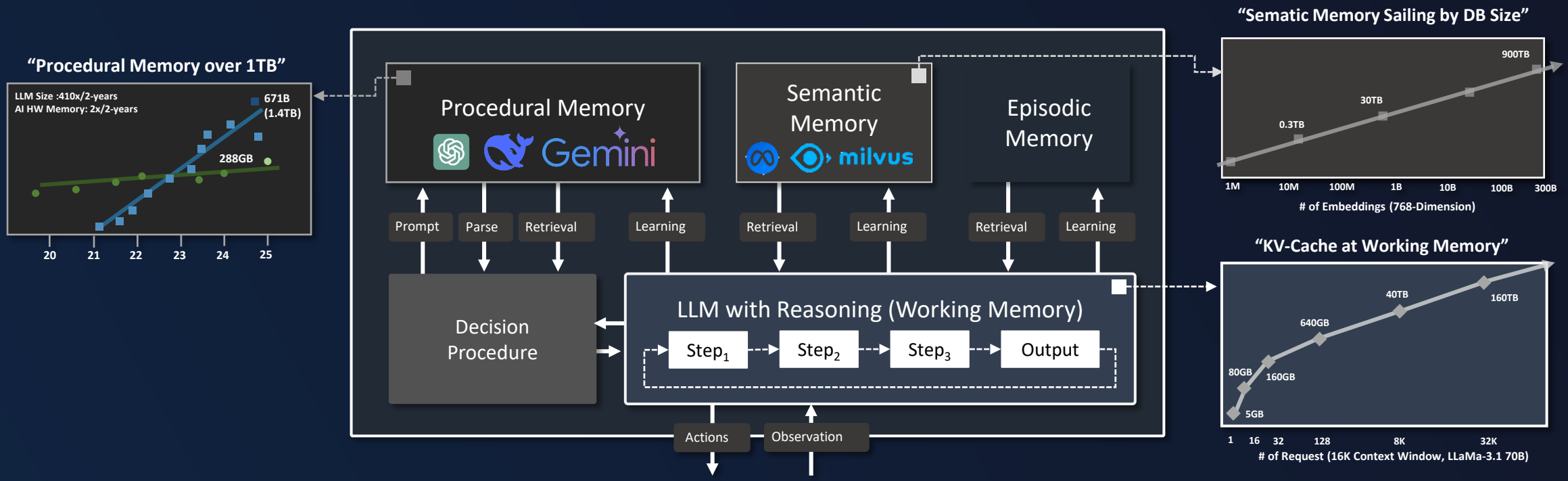
Jinin So, Senior Direct & System Architect
Head of Memory Architecture Group
Samsung Memory Division

Agentic AI Memory Requirement

Agentic AI based on LLM requires significant memory capacity & Bandwidth relying on external databases.

Blending working and long-term memory powers adaptive intelligence

- Working memory is a space where the agent temporarily stores and processes information for immediate use in current tasks or reasoning
- Procedural Memory (Implicit knowledge stored in LLM weights), Semantic Memory (Agent's knowledge about the world and itself)
- Episodic memory stores experiences from earlier decisions, such as training input-output pairs, history event flows, game trajectories



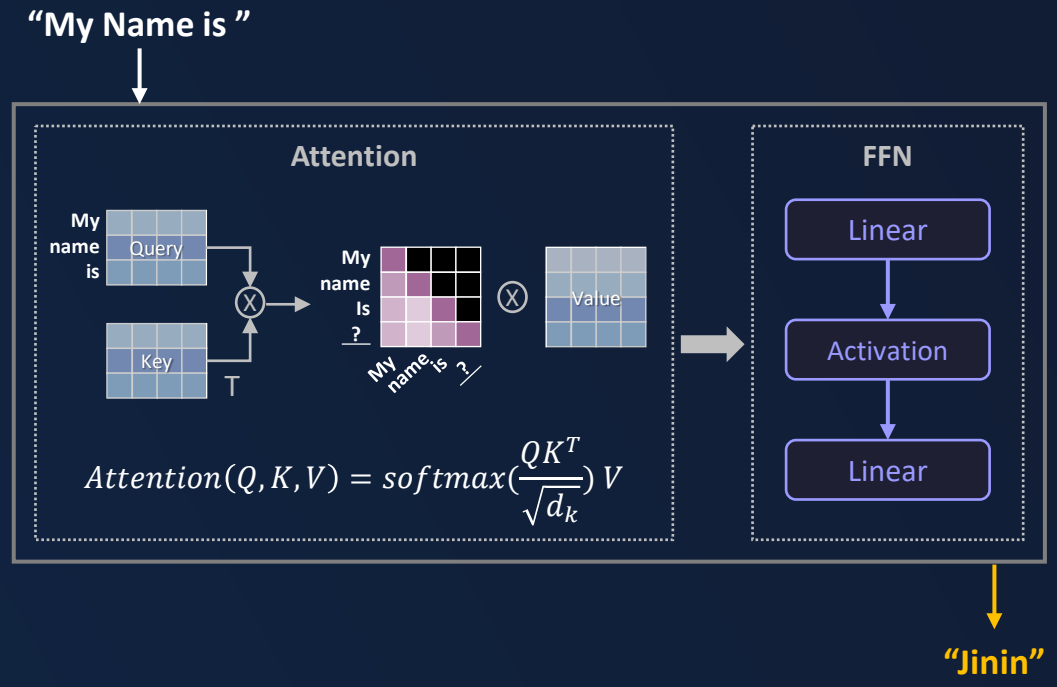
< Overall Architecture of LLM-based Agentic AI Model >

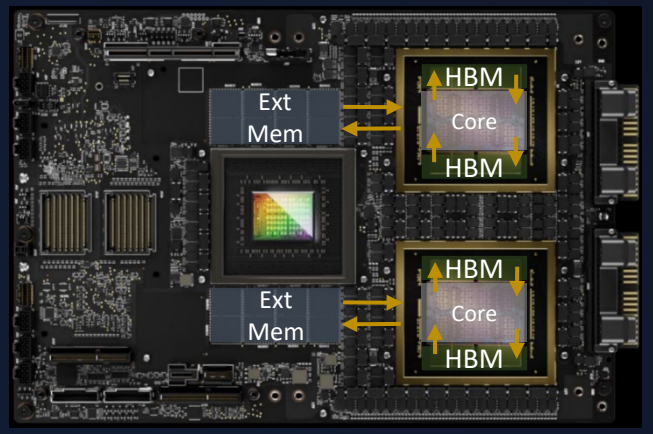
Procedural Memory Definition & Role

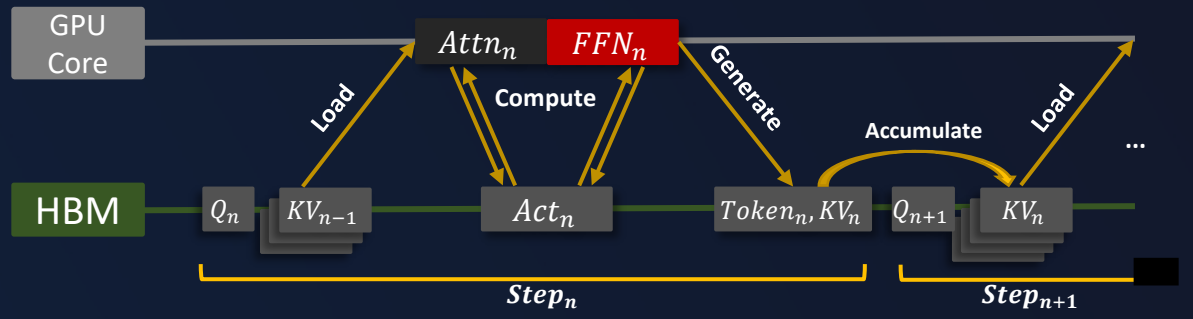
Memory that remember “How” to do something, rather than “What” is it

LLM weights act as procedural memory, implicitly capturing how to perform countless tasks

- Based on procedural knowledge stored in its weights, the LLM can automatically determine and execute “how” to process any given input
- GPU cores leverage HBM's high bandwidth to rapidly receive and process the computational data needed for attention and FFN operations



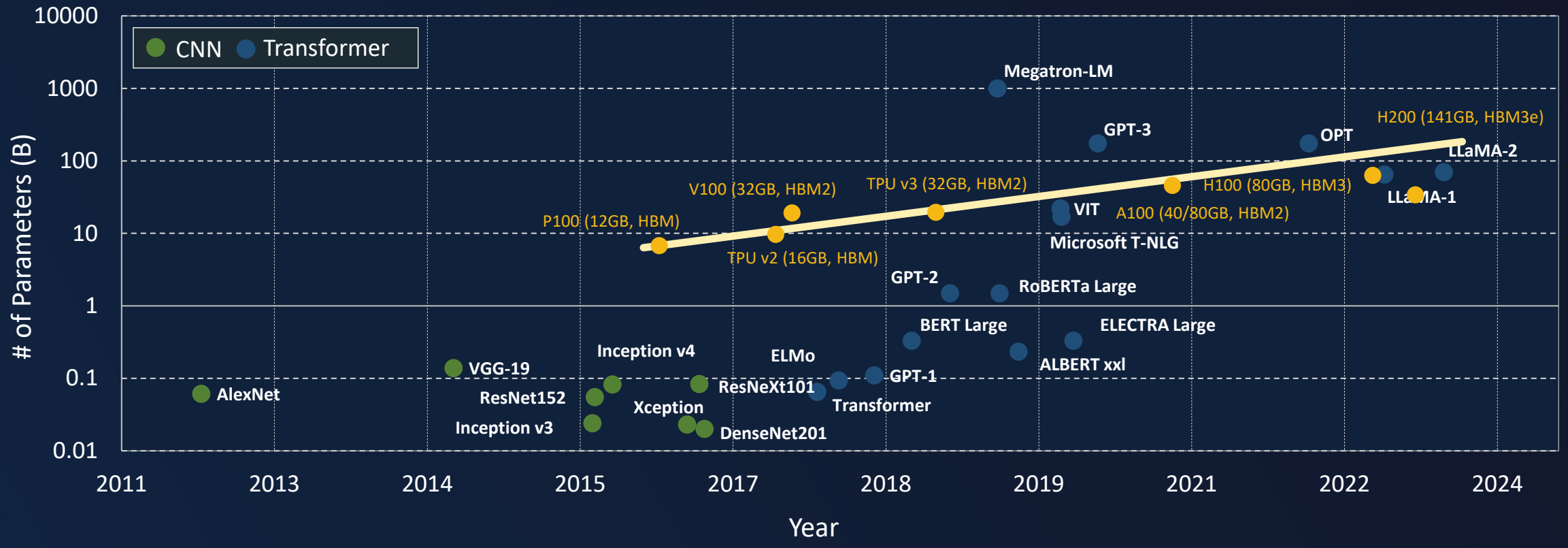




Procedural Memory – Capacity Requirement

LLMs require more memory bandwidth/capacity to satisfy service latency agreement (SLA)

Multiple GPUs leverage HBM's high memory bandwidth and capacity to serve LLM inference



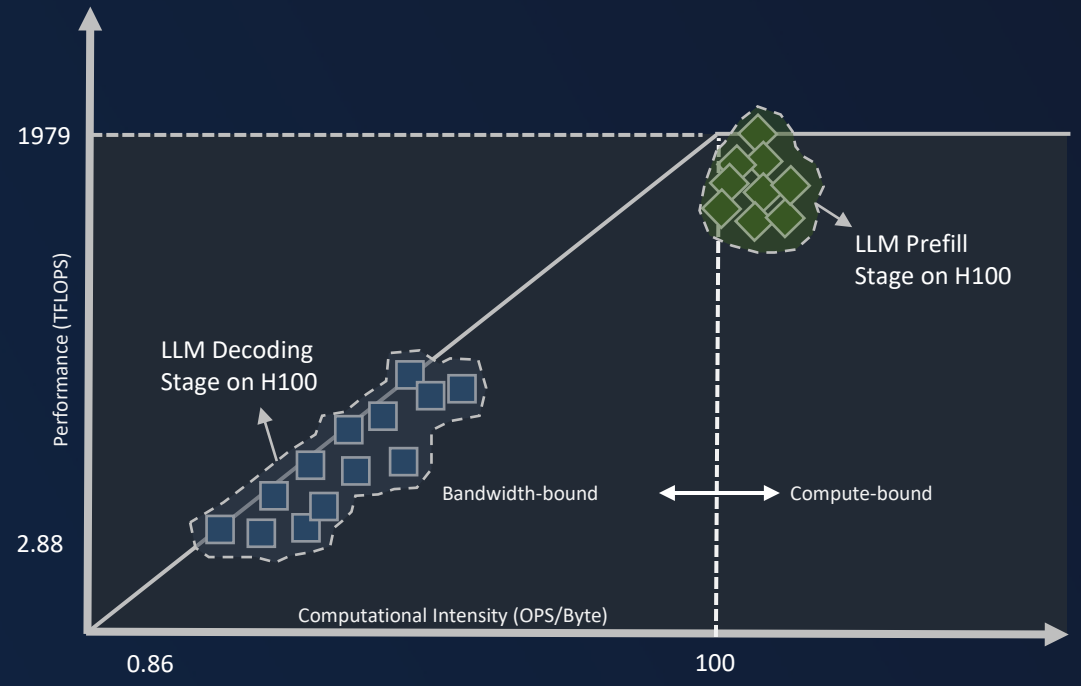
*Reference: AI and Memory Wall (https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8)

Procedural Memory – Bandwidth Requirement

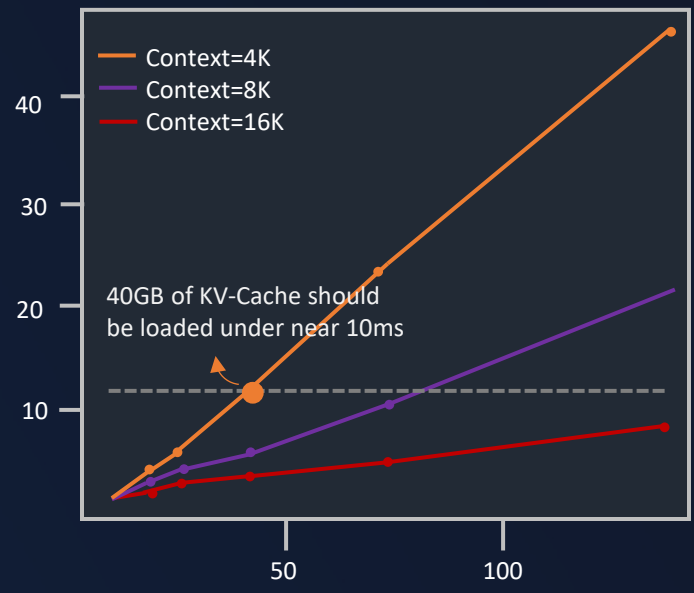
Too meet the LLM SLA, High Bandwidth Memory (HBM) is essential

More HBM bandwidth required to move weight fast from memory to GPUs

- Due to LLM's memory-intensive demands, delivering services within SLA requires memory that offers both high bandwidth and large capacity
- HBM is undergoing continuous scaling in both capacity and throughput, ensuring that it can efficiently store increasingly large model weights
- Advancement in memory technology tackles today's performance issues while supporting robust, real-time operations in LLM inference



< Roofline of LLM inference on H100 GPU >

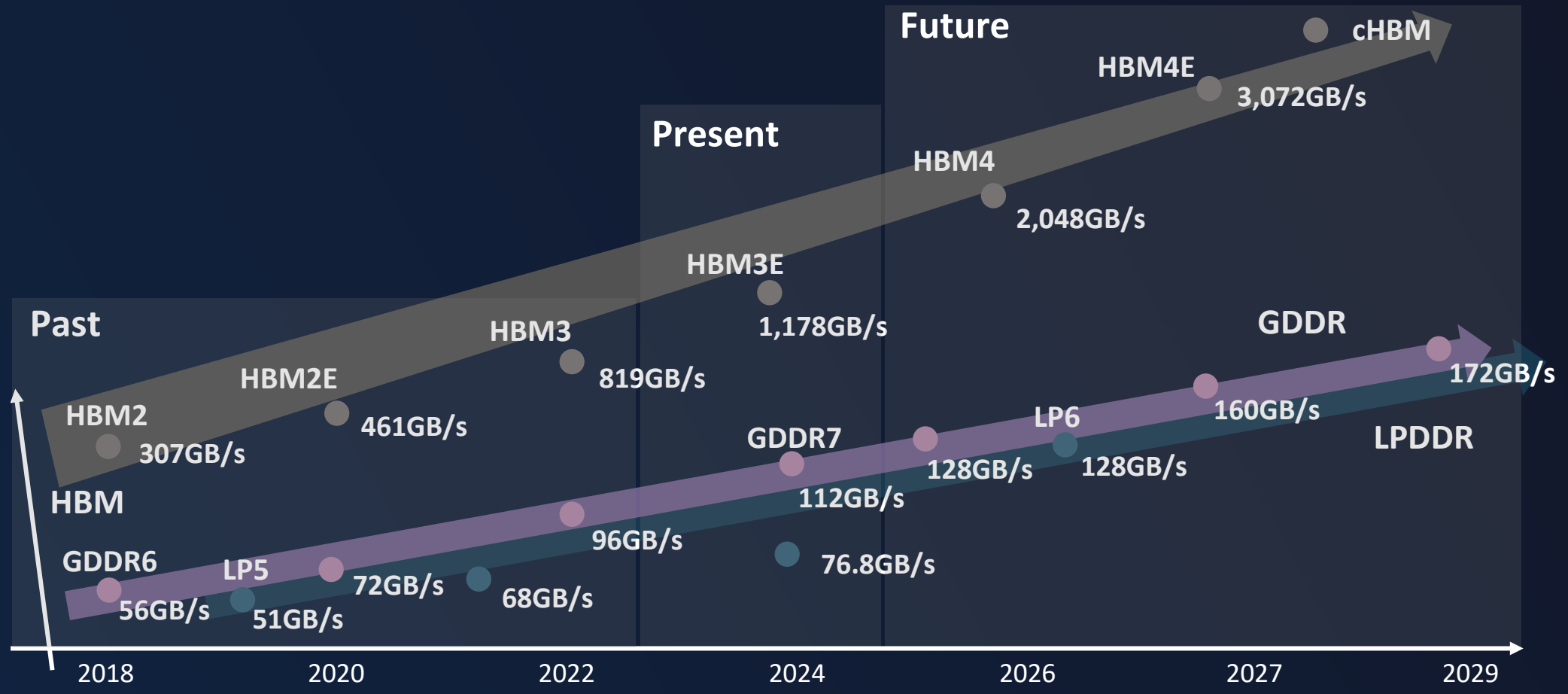


< LLaMA2-70B Query Latency >

HBM: DRAM Solution for Procedural Memory

Continuous Samsung's HBM innovation to support explosive memory bandwidth increase

More HBM bandwidth required to move weight fast from memory to GPUs

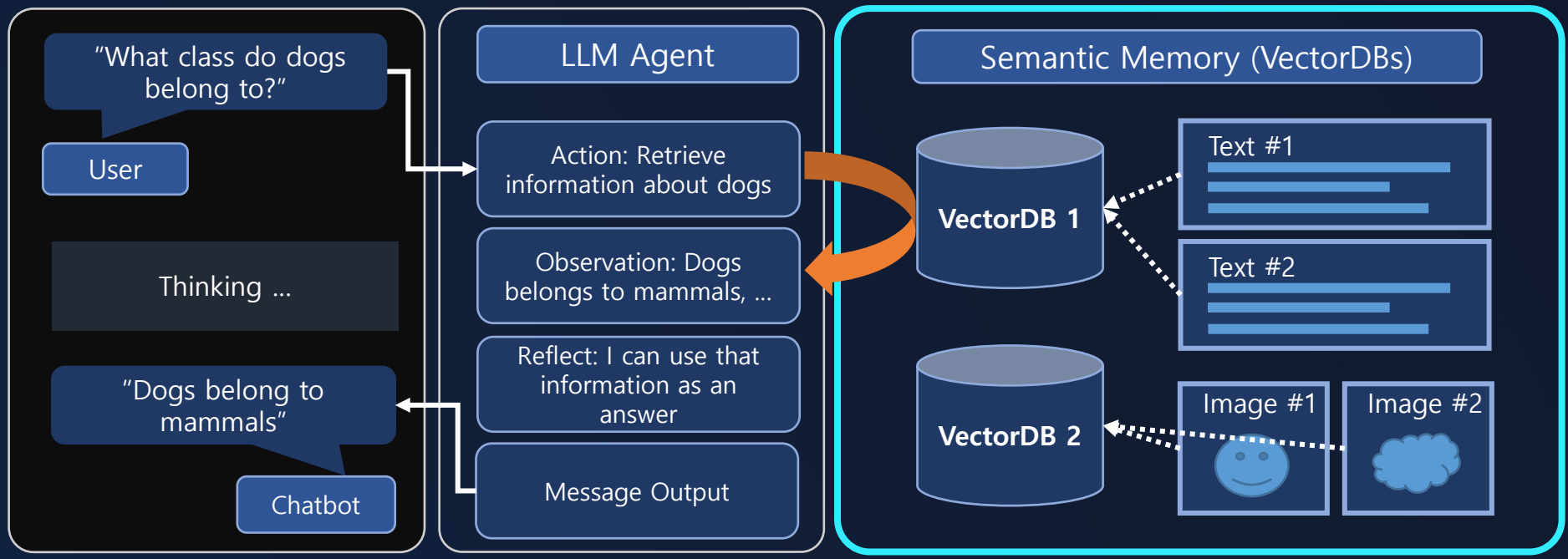


Semantic Memory Definition & Role

Memory is general knowledge about the world, and usually implemented as vector database

Semantic Memory is important for accurate answers of LLM Agents

- Semantic memory is general knowledge about the world, and usually implemented as vector database in Agentic AI
- Each knowledge item (text, image, …) can be embedded to vectors, and the number of vectors can be billion scale
- Agent can generate appropriate answers based on the retrieved relevant information (=Retrieval Augmented Generation, RAG)



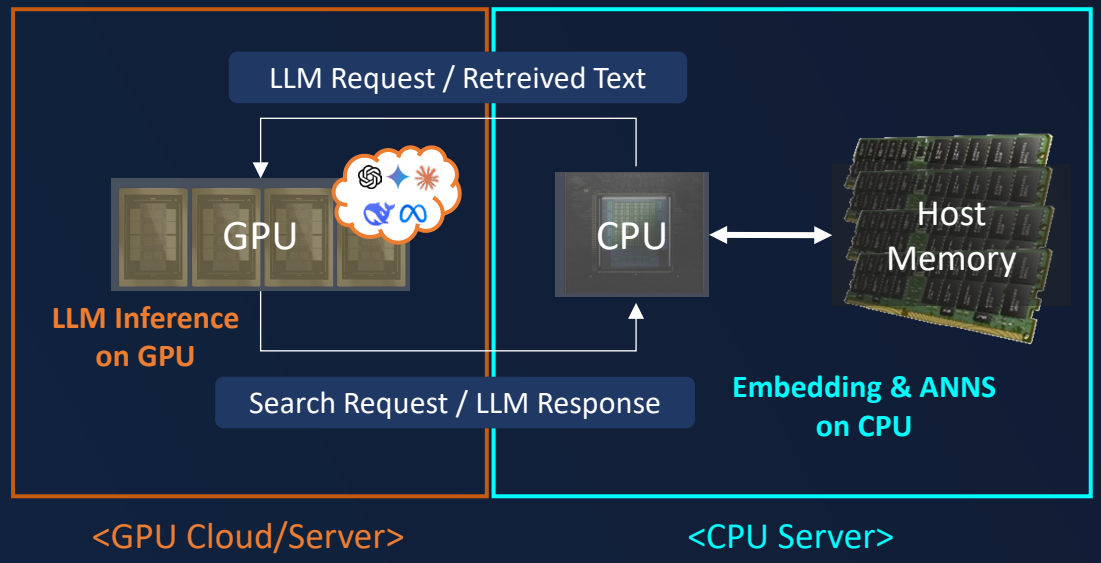
< Agentic RAG Workflow >

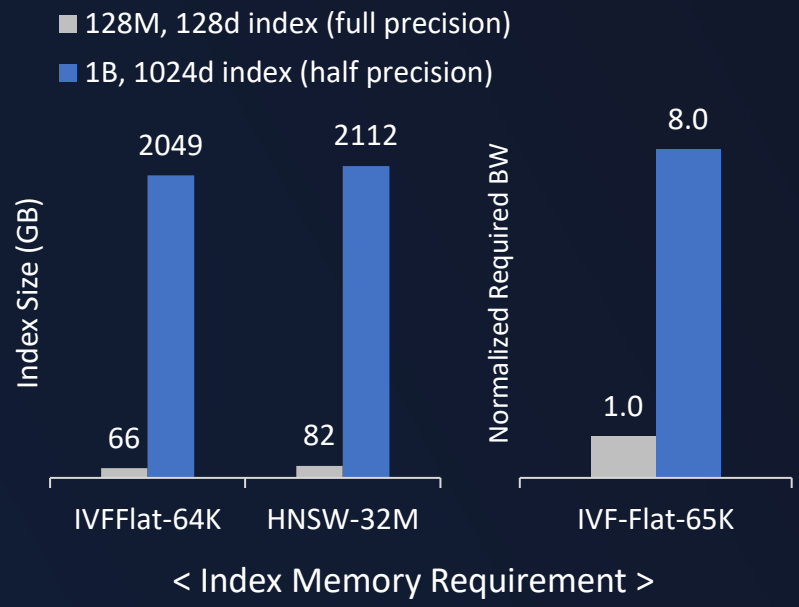
Semantic Memory Requirement

Semantic Memory usually consumed by CPU, high capacity and middle BW memory required.

Vector Search for RAG requires both large host memory capacity & high bandwidth

- While GPU is busy with inferencing LLMs, CPU is suitable accelerator for vector search (Nearest Neighbor Search)
- Recent commercial embedding model requires more than 1k dimensional vectors, resulting in higher capacity (~TBs)
- Required memory bandwidth is also growing, as the required bandwidth is proportional to the capacity





MRDIMM: DRAM Solution for Semantic Memory

Multiplexed Combined Rank DIMM beyond 12.8Gb/s, Larger capacity enabled with 2U F/F and 32Gb-based TSV

4x

Capacity

512GB with 2U F/F, 32Gb TSV

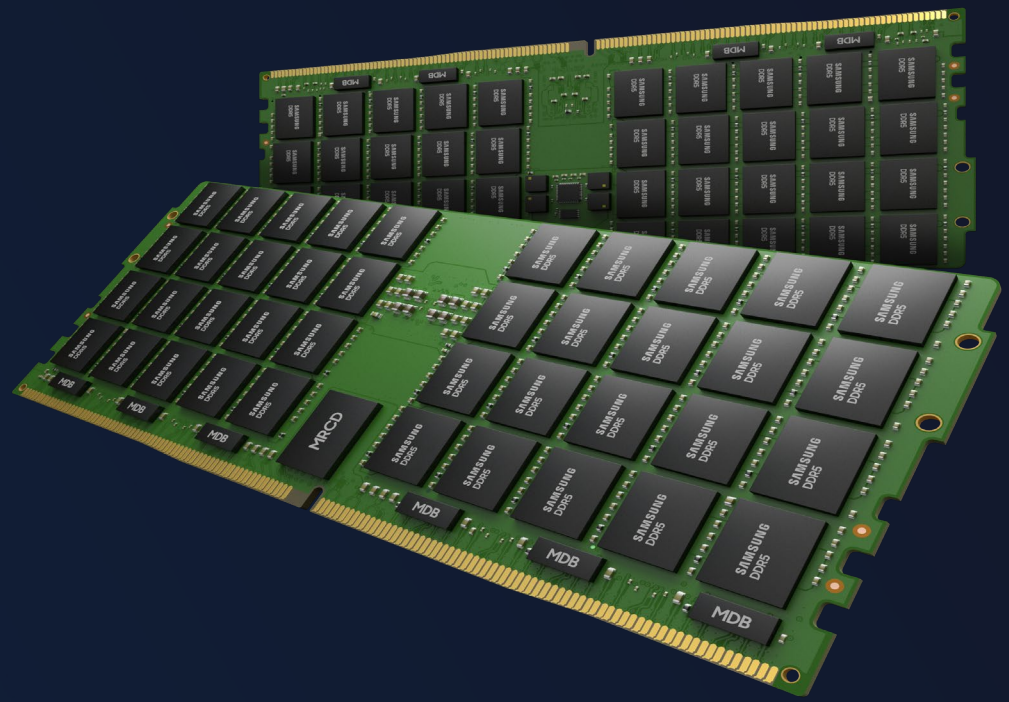
Compared to 32Gb mono die

2x

Performance

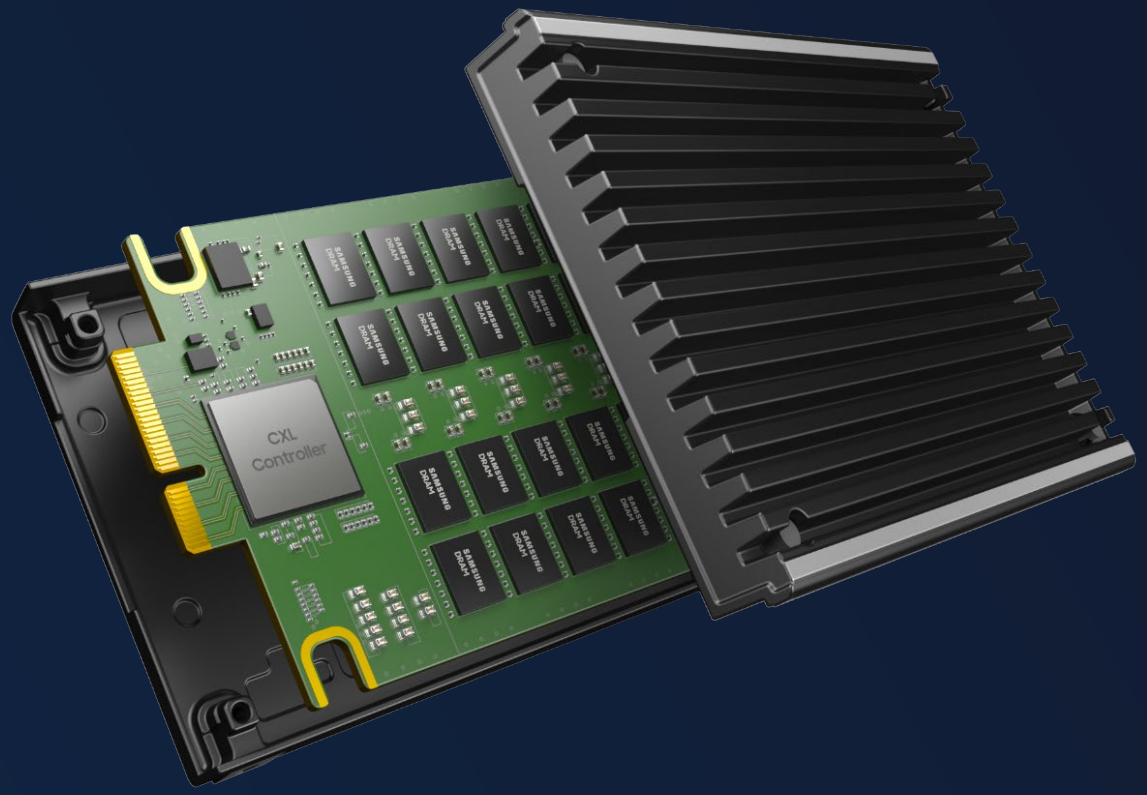
Up to 12.8Gbps

Compared to 6.4Gbps RDIMM



CMM-D: DRAM Solution for Semantic Memory

Leveraging CXL technology to provide scalable, high-capacity memory for data center and AI workloads



the industry[1st] CXL memory

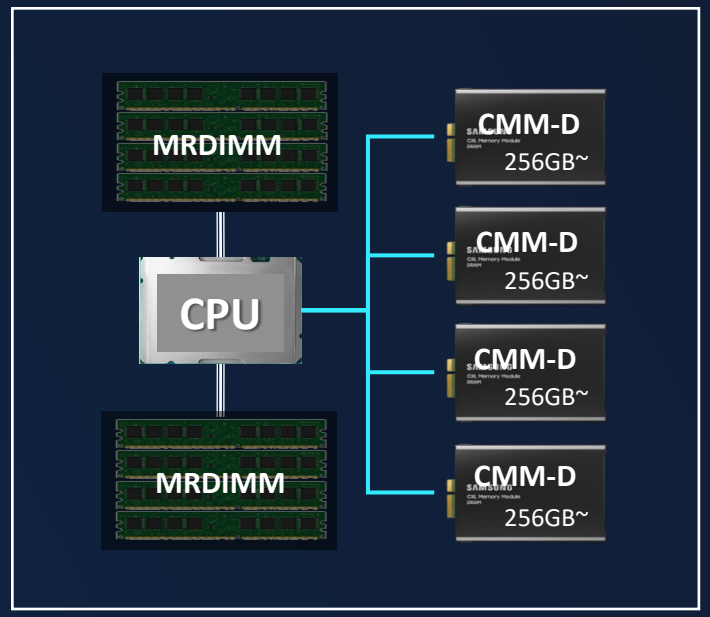
CMM-D D	CMM-D 2.0	CMM-D 3.0 0
Capacity	128/256GB	Up to 1TB
Bandwidth	36GB/s	72GB/s
Specifications	▪ CXL 2.0 ▪ PCIe Gen 5.0 w/ Single DDR5 Ch. ▪ Form Factor: EDSFF (E3.S, 2T) *Max. 80 DRAMs applicable for E3.S	▪ CXL 3.x *Enhanced for Pooling & Sharing ▪ PCIe Gen 6.0 w/ Dual DDR5 Ch.
Product Status	▪ C/S, Now	▪ F/S, 2026

MRDIMM & CMM-D Capacity & BW Benefit

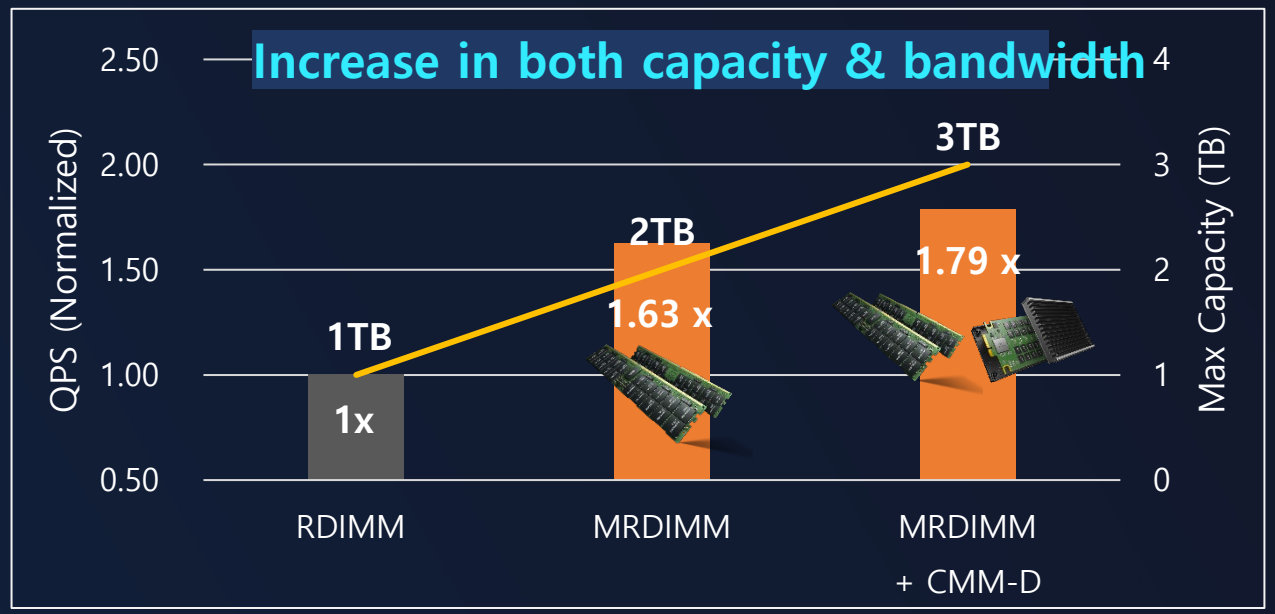
The combination of MRDIMM & CMM-D can bring both BW and Capacity for RAG application

Approximate Nearest Neighbor Search Performance

- HNSW index supports fast and accurate search, but requires large memory capacity because of additional graph information
- Compared to RDIMM, MRDIMM has a 60% query-per-second (QPS) improvement due to the additional BW provided
- When CMM-D is added, there is an additional 16% QPS improvement along with additional 1TB DRAM capacity



<Future System with MRDIMM & CMM-D>



< Index(HNSW) performance benefit of MRDIMM & CMM-D* >

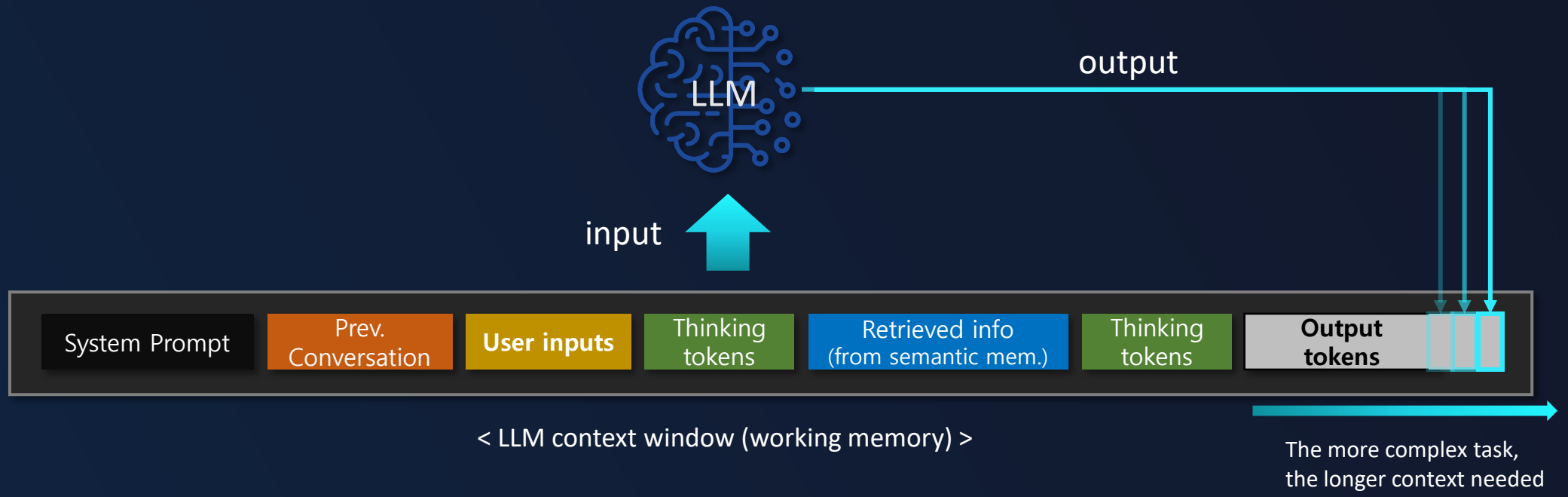
*HNSW @Recall>0.95 Configuration. 1 socket GNR-SP, 8ch RDIMM-4800/MRDIMM-12800, 4ch CMM-Ds
<Samsung NAND AE, SMRC>

Working Memory Definition & Role

Working memory enables AI agents to perform complex tasks through continuous reasoning

Working memory is where user prompts, past decisions, retrieved information are stored for continuous and comprehensive thinking

- AI agent's working memory is implemented as key/value cache, which consists of vectors generated by LLMs
- The sufficient context window is required for multi-step reasoning and decision-making of complex task

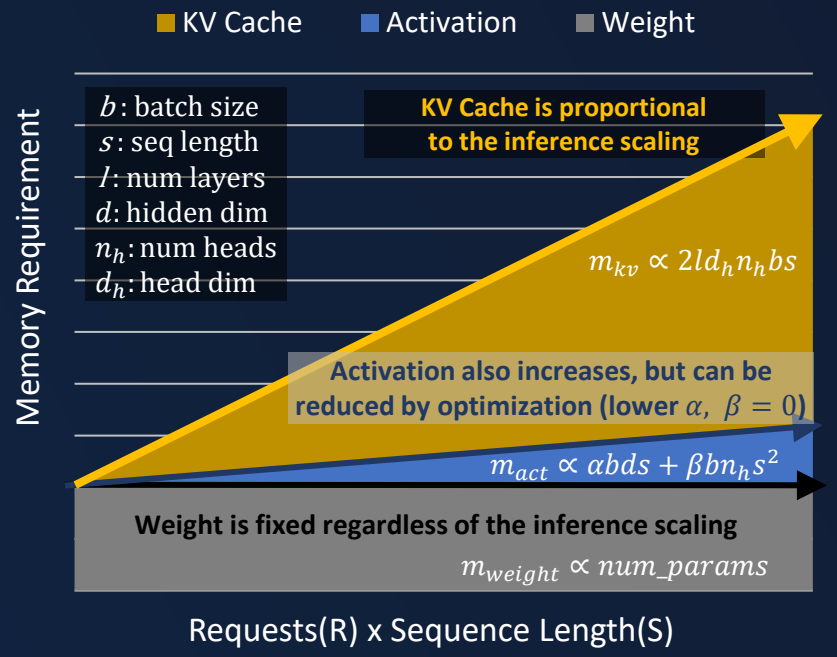


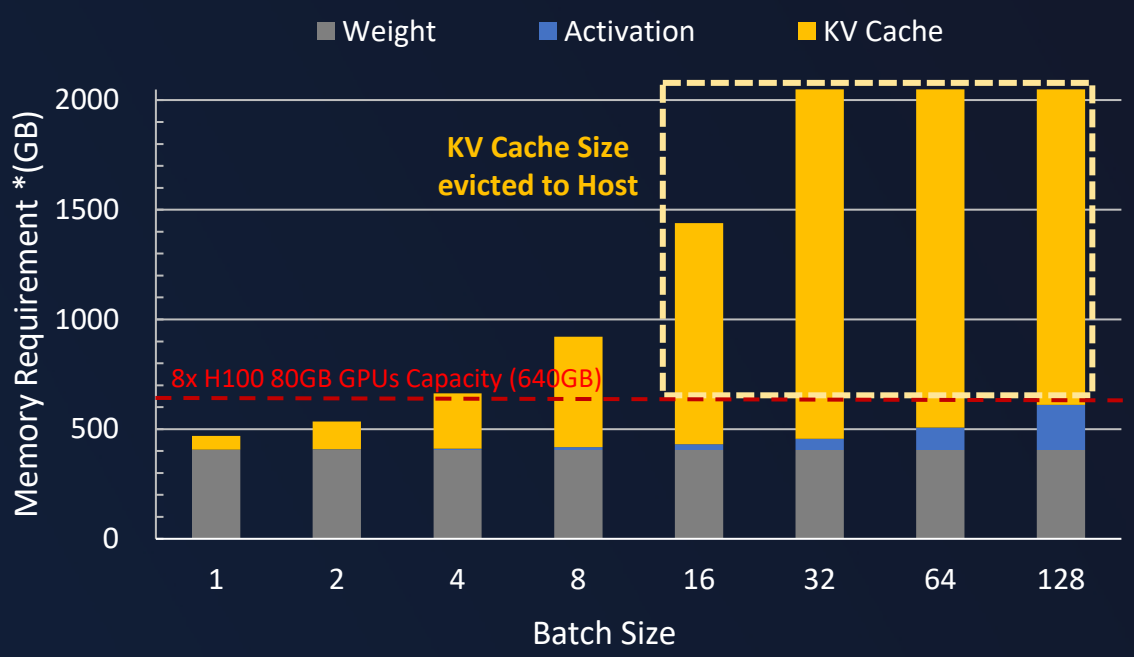
Working Memory – KV cache

KV cache is proportional to batch-size and context length and needed to be stored to 2nd tier memory

The size of KV-cache with large batch can surpass the memory capacity of GPU devices

- KV cache : 2*[num_layers:fixed]*[head_dim:fixed]*[num_heads:fixed]*[batch_size:variable]*[context_length:variable]
- Exceeding amount of KV cache should be evicted to host memory (rather than discarding) for efficient inference





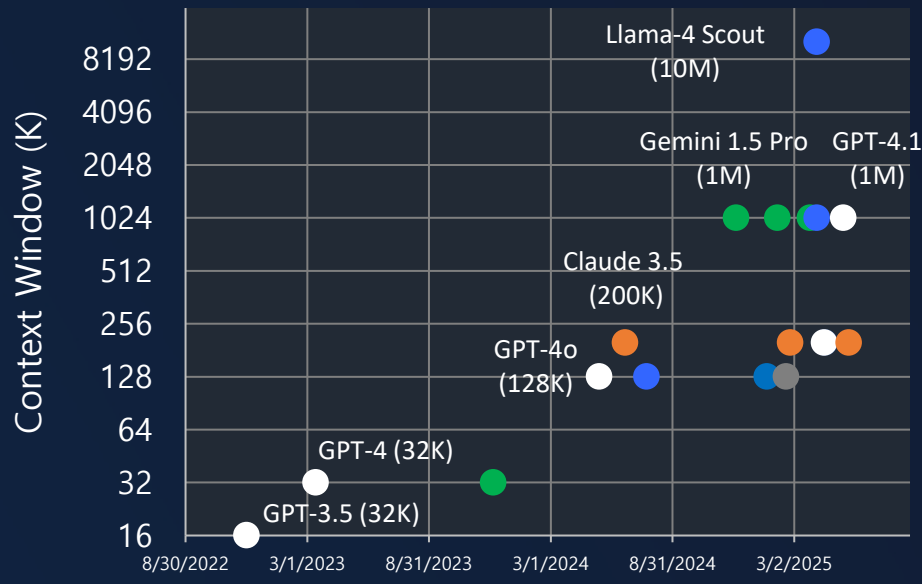
*Workload: LLaMA 3.1-405B, 128K context length per request.

Working Memory Requirement

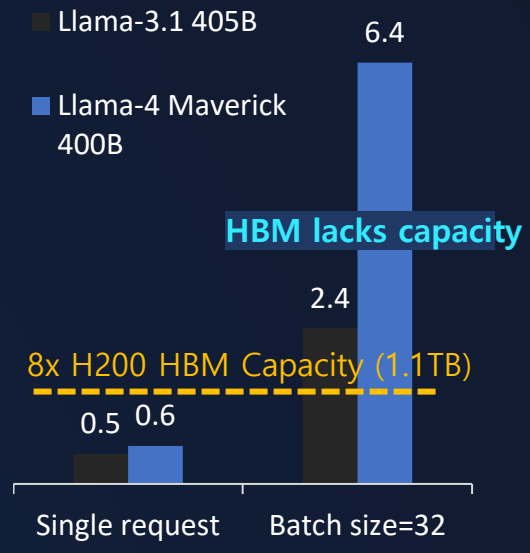
To support working memory, a solution satisfying both high capacity & bandwidth is required

As the size of context window increases, the LLM requires both very large capacity and enormous bandwidth to meet the SLA

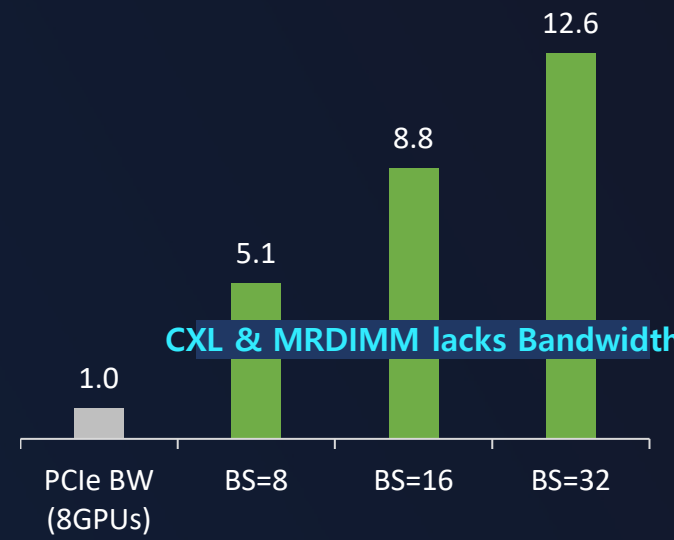
- HBM provides high bandwidth, but has shortage in capacity, and cost-per-capacity is expensive
- CMM-D & MRDIMM provide high capacity, but has shortage in interconnect bandwidth to the computing location (GPU)
- A solution is needed to address both capacity and the bandwidth limitation



< LLM Context Length >



< Required Capacity (TB) >



< PCIe 6.0 vs LLM Required BW*(TB/s)>

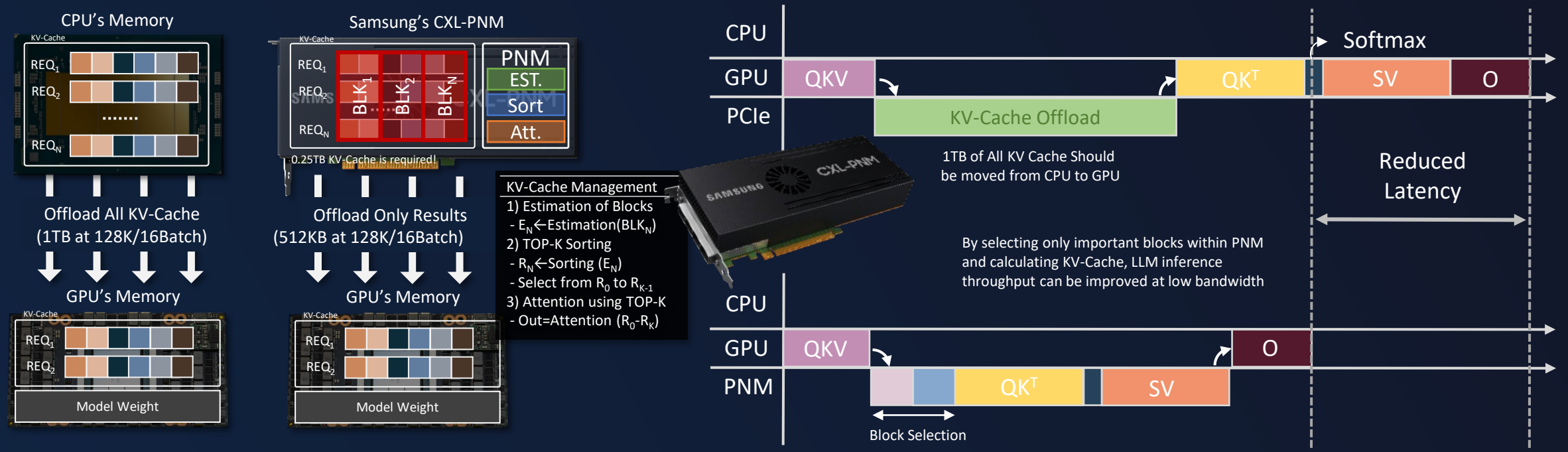
*SLA: 100ms TPOT, Llama 4 Maverick, Max context length (1M), Last token computation, partial KV offloading

PNM/PIM: DRAM Solution for Working Memory

By offloading KV Cache related operations, the inference performance can be improved 7.3x times better

As KV-Cache grows, data movement becomes a bottleneck, occupying most execution time

- When offloading a 128K context of LLaMA3.1-405B model, 1TB of data needs to be moved, but CXL2.0 moves data at 64GB/s bandwidth(x8)
- PNM performs token block (page) based selection directly w/ CXL memory module, reducing amount of computation and data movement
- By taking only important KV-Cache, amount of operation/bandwidth are reduced, PNM can performance improvement at lower bandwidth



< Computation of Attention Layers in CPU+GPU (Upper) and CPU+GPU+PNM (Lower) >

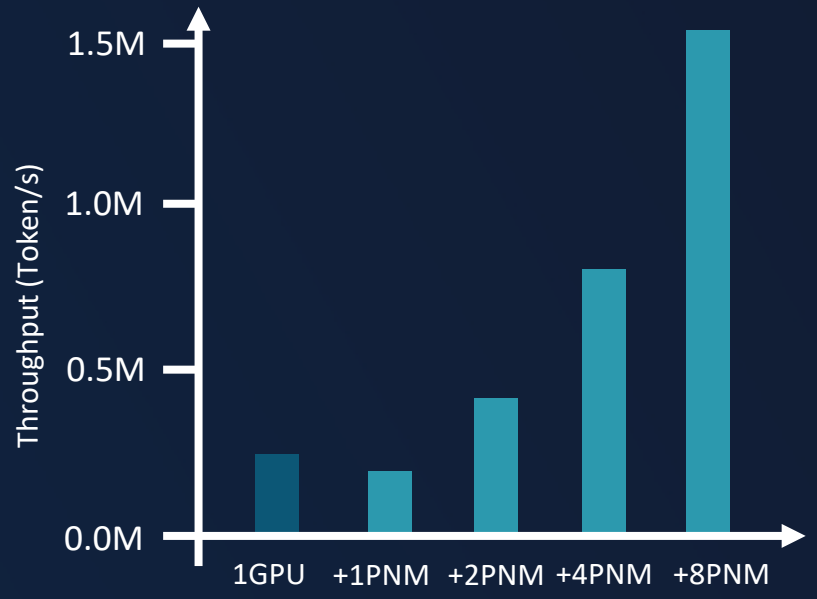
*Workload: LLaMA 3.1-405B, 128K context length per request

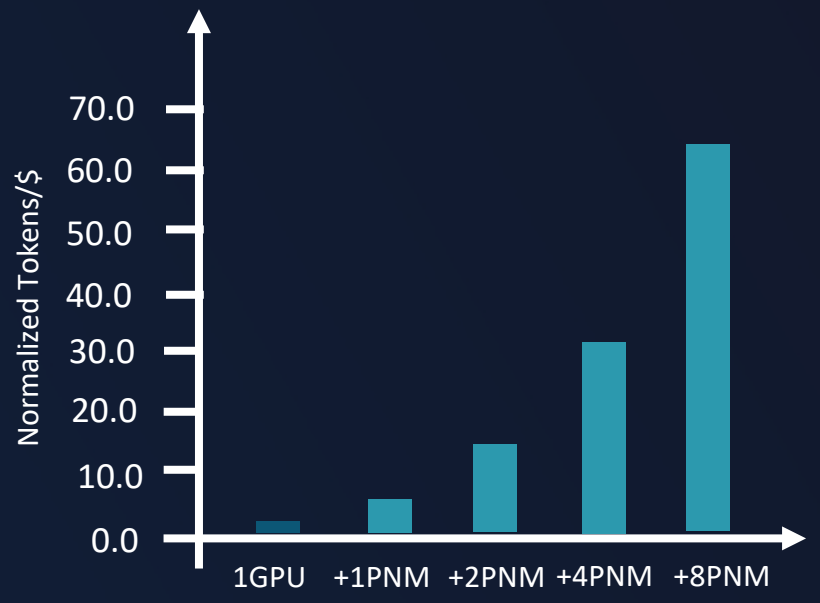
Use Case of PNM Solutions for Agentic AI

By offloading KV Cache related operations, LLM inference can be cover more batch size and more throughput

Efficient KV-Cache management running PNM for scalable LLM inference to long contexts

- CXL-PNM performs token block based selection directly within CXL memory module, eliminating GPU's KV-Cache offload cost
- By storing KV-Cache in CXL memory and offloading selection to PNM, GPU memory pressure and support larger batch sizes for FC layers
- With the integration of PIM technology, future KV Cache operations can be processed with greater speed and efficiency





< End-to-End throughput/Efficiency comparison at LLaMa3.1-70B with Context length at 128K tokens >

Summary

- Agentic AI represents a significant shift in AI, necessitating advanced, layered memory systems.
- Key challenges include efficient data movement, especially migrating working memory KV caches during operations.
- Combining CXL memory with PNM & PIM tech and optimizing KV cache management minimizes data movement effectively.
- Samsung collaborates with OCP's Data-Centric Computing FTI to develop these innovations.