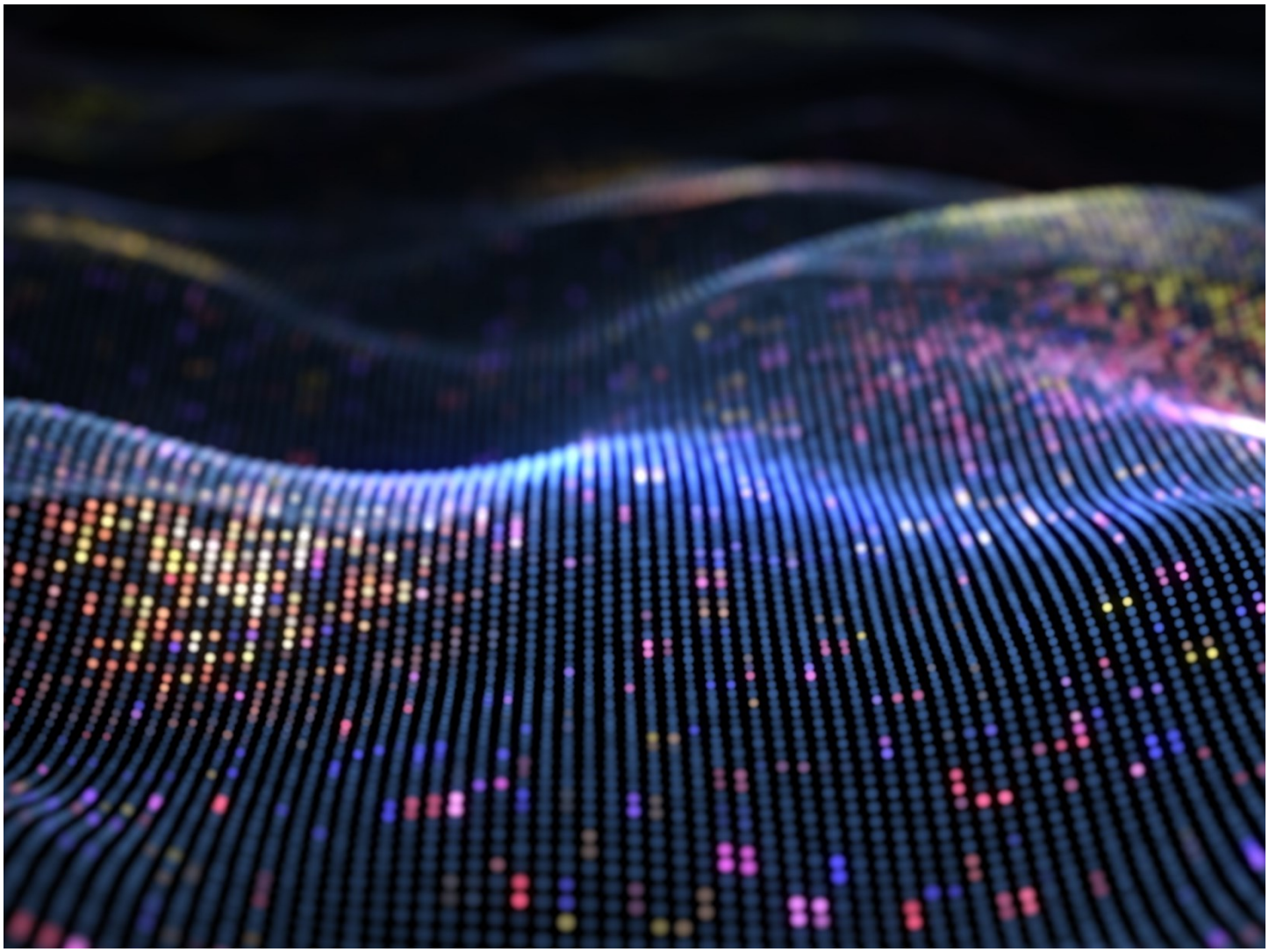
SAMSUNG | SGL

White Paper

Composable Memory for AI Inference at Scale

A Joint Technical White Paper



Samsung Memory Solutions Lab and
SGLang creators from RadixArk

September 2026

LEGAL DISCLAIMER

This document has been prepared by Samsung Electronics Co., Ltd. (“Samsung”). The contents of this document are the property of Samsung, protected by applicable laws and non-disclosure agreements between Samsung and you or your employer, as applicable. You are strictly prohibited from, including, but not limited to disclosing, copying, reproducing, distributing, transmitting, modifying, rebroadcasting, re-encoding, re-presenting, exploiting, or creating derivative works of this document or any parts of this document without the prior written permission from Samsung.

The contents of this document are provided for informational purposes only. No representation or warranty (whether express or implied) is made by Samsung or any of its officers, advisers, agents, or employees as to the accuracy, reasonableness or completeness of the information, statements, opinions, or matters contained in this document, and they are provided on an "AS-IS" basis. Samsung will not be responsible for any damages arising out of the use of, or otherwise relating to, this document. Nothing in this document grants you any license or rights in or to information, materials, or contents provided in this document, or any other intellectual property.

The contents of this document may also include forward-looking statements. These forward-looking statements include all matters that are not historical facts, as well as statements regarding Samsung’s intentions, beliefs and current expectations concerning, among other things, market prospects, growth, strategies, and the industry in which Samsung operates. By definition, forward-looking statements involve risks and uncertainties because they relate to events and depend on circumstances that may or may not occur in the future. Samsung hereby reminds you that forward-looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or the industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements contained in this document or in the accompanying oral statements. In addition, even if the information contained herein or the accompanying oral statements are shown to be accurate, those developments statements may not be indicative of future developments.

All contents in this document may be subject to change without notice. Without limiting the generality of the foregoing,

1. All design, features and specifications represented herein may change without notice;
2. Images shown here have been adjusted for demonstration purposes and may appear differently on the actual products;
3. All data on products herein, including their performances, are based on internal testing using standard Samsung benchmarks under laboratory conditions. Test results do not guarantee future performance under such test conditions, and the actual throughput or performance that any user will experience may vary depending upon many factors; and
4. All images on screen are simulated, except where otherwise noted.

BY CONTINUING TO ACCESS THIS DOCUMENT, YOU ARE DEEMED TO HAVE READ, UNDERSTOOD, AND AGREED WITH THE FOREGOING TERMS AND CONDITIONS.

Abstract

Extending GPU Memory Boosts AI Inference

Serving large numbers of concurrent users across modern AI inference workloads — multi-turn chatbots and long-context agentic reasoning — is fundamentally a memory capacity problem. The High Bandwidth Memory (HBM) on each GPU must be shared between static model weights and the growing key-value (KV) cache of every active session.

As user concurrency scales, the KV cache exhausts the HBM headroom that remains after model weights are loaded, causing latency to spike long before the GPU's compute capacity is reached. This paper describes how the Samsung Cognos AI Memory Agent and the SGLang HiCache hierarchical KV cache layer, working together, solve this bottleneck at the infrastructure level.

The two systems combine to expose an AI Memory Node™ containing tiered composable memory as a transparent extension of the GPU's local memory hierarchy. In a benchmark designed to reflect a realistic enterprise mix of long-context coding agents and multi-turn chatbots, the joint solution delivers up to a 3x improvement in inter-token latency, a 2.5x improvement in cache hit rate, and a 2.2x improvement in token throughput on identical GPU hardware, while reducing the cost per served user (normalized to steady state concurrent sessions) by approximately 40 percent. The remainder of this paper describes the architecture, deployment methodology, and measured results in full.

1. Problem: KV Cache Pressure at Scale

1.1 Memory Wall in LLM Inference

Every active LLM session occupies a fixed share of GPU HBM as its KV cache — the key and value tensor pairs that the transformer attention mechanism retains for every token in context. GPU HBM serves two roles at the same time: It holds the model weights, which are static and loaded once, and it holds the KV caches of all concurrent sessions, which are dynamic and grow with context length. When the total KV cache demand across concurrent sessions exhausts the HBM headroom that remains after model weights are loaded, the inference server faces a difficult choice. It can evict an active session's KV blocks and recompute them on the next turn, or it can cap the number of sessions it will accept.

The critical observation is that performance degrades as more users are served. Maintaining Service Level Objectives (SLOs) at scale becomes the dominant operational concern for multi-tenant workloads. When enterprises are constrained in the number of GPUs they can deploy, extending the memory hierarchy to slower tiers such as DRAM and CXL significantly reduces this degradation.

1.2 Why Coding Agents Hit This Wall Hardest

Coding agents are long-context workloads, and they represent one of the most memory-intensive LLM workload profiles in enterprise deployment. Three characteristics compound the KV cache pressure:

1. Long, persistent context per session. Developers load source files, repository documentation, and multi-turn conversation history into context. A typical coding session spans 2,000 to 8,000 or more tokens, far larger than a short-form chat interaction. Each active developer session occupies a correspondingly large KV footprint in HBM.

2. Shared organizational prefixes. All developers on a team share the same system prompt, coding style guidelines, and codebase documentation. This prefix-level content generates identical KV data across sessions, data that is recomputed from scratch on every session start when no caching tier exists to retain it.

3. Strict interactive latency requirements. A developer waiting on a code completion has a time-to-first-token (TTFT) tolerance measured in hundreds of milliseconds. Eviction-induced recomputation can multiply TTFT by three to ten times, well beyond the threshold at which the tool feels responsive. Unlike batch workloads, there is no slack to absorb latency spikes.

1.3 Cost of Scaling with GPUs Alone

The default response to KV cache pressure is to add more GPU pods, acquiring additional HBM headroom together with compute that may largely sit idle. The economics are unfavorable. Total cost of ownership for a GPU pod — hardware, power, cooling, and rack space — scales linearly with node count regardless of compute utilization. For an enterprise growing from 50 to 100 concurrent developers, a purely GPU-scaled solution doubles infrastructure cost for what is fundamentally a memory capacity problem. A purpose-fit memory expansion at a fraction of GPU cost per gigabyte is the more efficient solution.

Local storage is not a sufficient alternative either. A natural response is to offload KV cache to fast local storage on the GPU node using GPUDirect Storage (GDS), keeping data close to the GPU over the PCIe bus. This defers the problem rather than solving it. Local storage is bounded by the drive capacity and PCIe slots physically available within the GPU chassis, so it runs into the same capacity ceiling as HBM — just at a higher threshold. More importantly, its performance degrades sharply under concurrency. As shown in Figure 1, a local NVMe-backed L3 tier (GDS-style) tracks the memory-over-fabric approach at low load but diverges rapidly beyond 64 concurrent users: At 128 concurrent users, P50 time-to-first-token for the local-NVMe tier reaches 50.3 seconds, roughly 62% higher than the 31.1 seconds delivered by CMM-D memory over fabric. Because local storage cannot be scaled independently of the GPU node, it offers neither the elastic capacity nor the sustained performance that production SLAs demand at scale. A disaggregated memory node — capacity that scales independently of the GPU pod and holds latency flat under load — is the more efficient path.

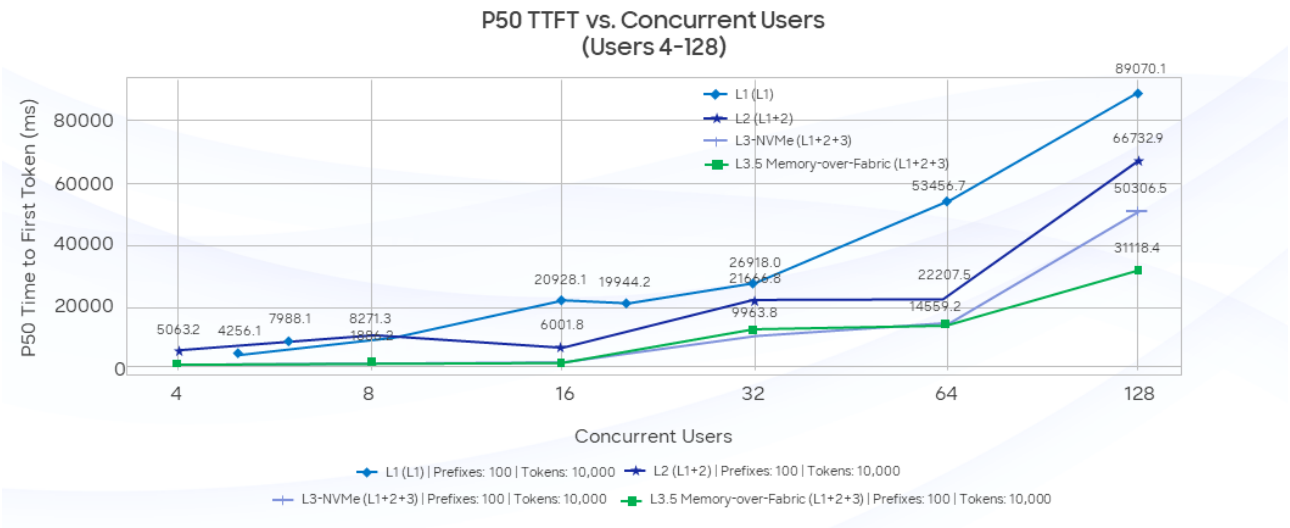


Figure 1. P50 TTFT under increasing concurrency — local NVMe (GDS) vs. CMM-D memory over fabric. Local storage degrades sharply beyond 64 users; memory over fabric holds latency substantially lower.

2. Solution: AI Memory Node™ Enabled by Samsung Cognos and SGLang HiCache

2.1 Cognos AI Memory Agent

Samsung Cognos is an AI memory orchestration platform for composable memory systems. Its goal is to meet application service-level agreements (SLAs) by managing three things together: memory allocation, memory tiering, and device health. Cognos pools heterogeneous memory — spanning direct-attached (HBM, DRAM, LPDDR, CMM-D, SSD), fabric attached (memory over RDMA/CXL), and emerging memory classes — and allocates it to workloads according to their SLA requirements for capacity, bandwidth, and latency. It tiers data/KV cache across these memory classes so that the most performance-sensitive data/KV resides in the fastest-appropriate tier, and it continuously monitors device health to sustain the reliability and predictability that SLA-driven environments demand. These capabilities are especially valuable for memory-intensive AI inference, where meeting latency and throughput SLAs depends on placing the right data in the right memory tier.

Building on this foundation, Cognos is extending its platform to bring composable memory directly into the AI inference path with AI Memory Node™, providing unified scale-up and disaggregated scale-out shared memory pools to GPU fabric. Each AI Memory Node™ can consist of various composable memory devices (e.g., DRAM, CXL, and SSD).

Cognos bridges the inference server and the AI Memory Node™. A Cognos client runs on the GPU node and interfaces with SGLang HiCache (described in the next section). It communicates with a Cognos server running on the AI Memory Node™, which orchestrates the composable memory pool and exposes it to the inference server as an additional cache tier.

Because the client and server handle KV movement between tiers, the inference server never manages the individual tiers itself. That abstraction raises effective GPU utilization, enables KV reuse across inference instances, and reduces redundant KV transfers. Just as importantly, decoupling memory capacity from GPU compute allows the two to be scaled independently.

2.2 SGLang HiCache

Solving the KV cache capacity problem in hardware requires an inference engine capable of managing the KV cache natively across multiple tiers of memory — treating GPU HBM, host CPU DRAM, and external composable memory as a single, hierarchically managed pool. SGLang HiCache provides this capability.

Designed by the LMSYS team and inspired by the three-level cache hierarchy of modern CPUs, HiCache organizes KV storage into GPU HBM as L1, host CPU DRAM as L2, and external composable memory (AI Memory Node™) as L3.5. A unified metadata structure, the HiRadixTree, tracks the precise location of every KV block across all three tiers. When a session's KV blocks have migrated to a lower tier, the tree locates them immediately and triggers an asynchronous prefetch before the GPU requests them. The developer experiences a warm session resume rather than a cold recompute.

Three implementation decisions make HiCache viable within production TTFT SLOs:

1. **Compute-transfer overlap.** During prefill, while the GPU computes KV data for layer N, HiCache concurrently transfers the precomputed KV data for layer N+1 from DRAM to GPU. Transfer latency is hidden behind computation rather than added to it.
2. **Optimized I/O kernels.** Specialized kernels for L2-to-L1 (DRAM-to-HBM) transfers achieve up to three times the throughput of standard host-to-device memory copy primitives, accelerating KV prefetch under high concurrency.
3. **Zero-copy L3/L3.5 transfer.** Write-backs to external memory pass memory addresses and sizes directly to the L3/L3.5 backend, eliminating intermediate data copies and maximizing bandwidth utilization on the network path to the AI Memory Node™.

   The L3.5 tier is backed by a clean abstract interface, HiCache, that any external memory backend can implement. Samsung Cognos implements this interface, registering the composable memory pool as a

native HiCache L3.5 tier. From SGLang's perspective, the Cognos-backed pool is a recognized memory and storage provider. No modifications to the SGLang codebase are required.

2.3 HiCache-Cognos Co-design System Architecture

The baseline deployment establishes the GPU node-only performance and the KV saturation point. The proposed solution introduces two additions: SGLang HiCache, which extends the inference engine to manage memory across multiple tiers, and Samsung Cognos, which manages the extended memory beyond the GPU pod through the AI Memory Node™. The AI Memory Node™ forms the L3.5 tier and is a standard CPU server containing DRAM and Samsung CMM-D.

The full system architecture is shown in Figure 2. The dark blue components represent the additions to the baseline deployment.

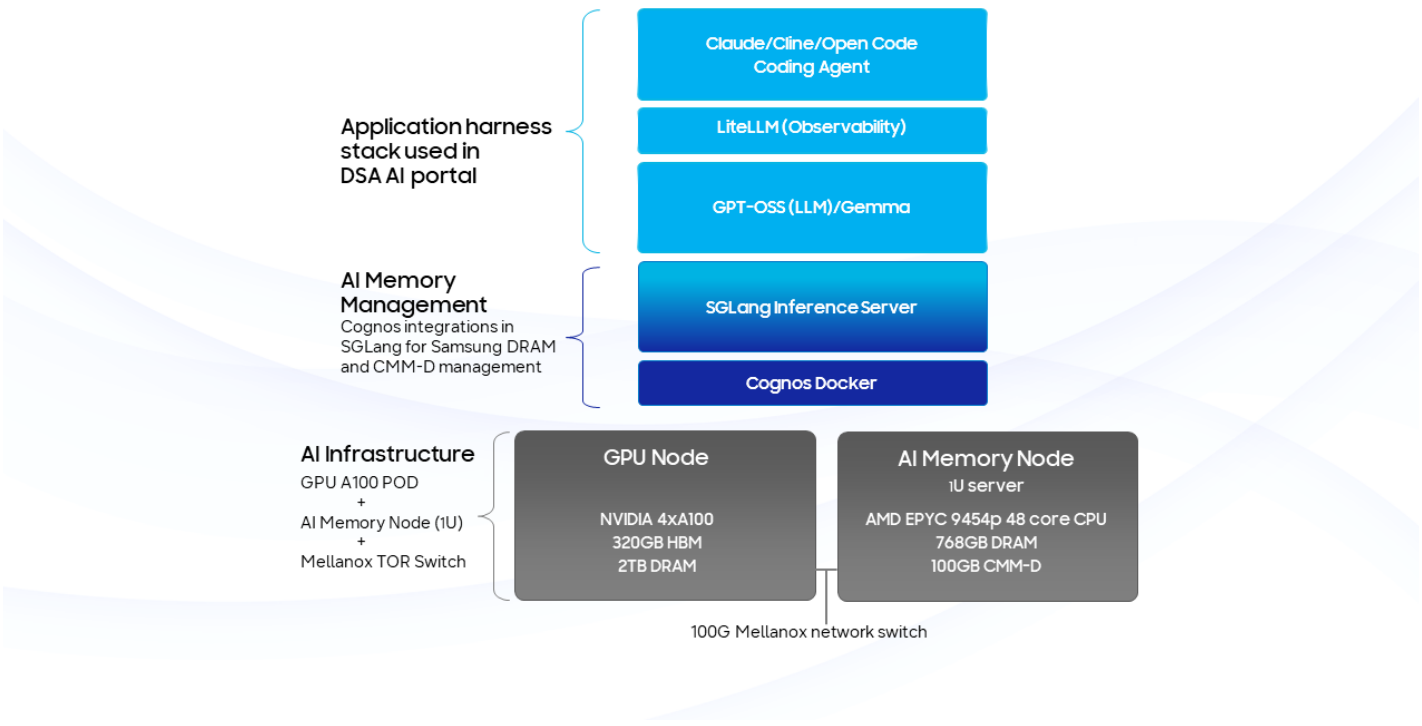


Figure 2. System Architecture — GPU node-only baseline extended with SGLang HiCache and the Samsung Cognos AI Memory Node™.

3. Evaluation Methodology

3.1 Benchmark Specification

The results in this report are obtained from a long-context benchmark running concurrently with a multi-turn benchmark in the background. The request rate of the long-context benchmark is held constant at 50 requests per minute. For each data point the multi-turn request rate is varied, and the corresponding long-context user percentage is defined as:

long-context-user (%) = long-context rate / (long-context rate + multi-turn rate) x 100

For the runs reported here, multi-turn request rates of 25, 30, 40, and 80 requests per minute were used, producing a sweep across long-context user percentages that represent realistic enterprise workload mixes.

3.2 Setup

Parameter	Value
Model	Open-weight 120 B-parameter language model
Inference engine	SGLang v0.5.10.post1
GPU configuration	4 GPUs with 80 GB HBM each
L1 (HBM) model weights per rank	55 GB
L1 (HBM) KV cache per rank	15 GB
L2 (host DRAM) HiCache size per rank	25 GB (total 100 GB)
L3.5 capacity	1.25 TB (DRAM + CXL CMM-D; full capacity not used)
AI Memory Node™ CPU	48-core x86 server CPU
Network	100 GbE

4. Results

The figures below report a workload mix of 55 percent long-context coding users and 45 percent multi-turn chatbot users, simulating a realistic enterprise environment. All metrics compare the joint solution (HiCache plus Samsung Cognos AI Memory Node™) against a GPU node-only baseline on the same GPU hardware.

4.1 Median Time to First Token

The joint solution reduces median TTFT by 61 percent (a 2.6x improvement) at the target workload mix, keeping interactive latency within the threshold at which coding agents feel responsive.

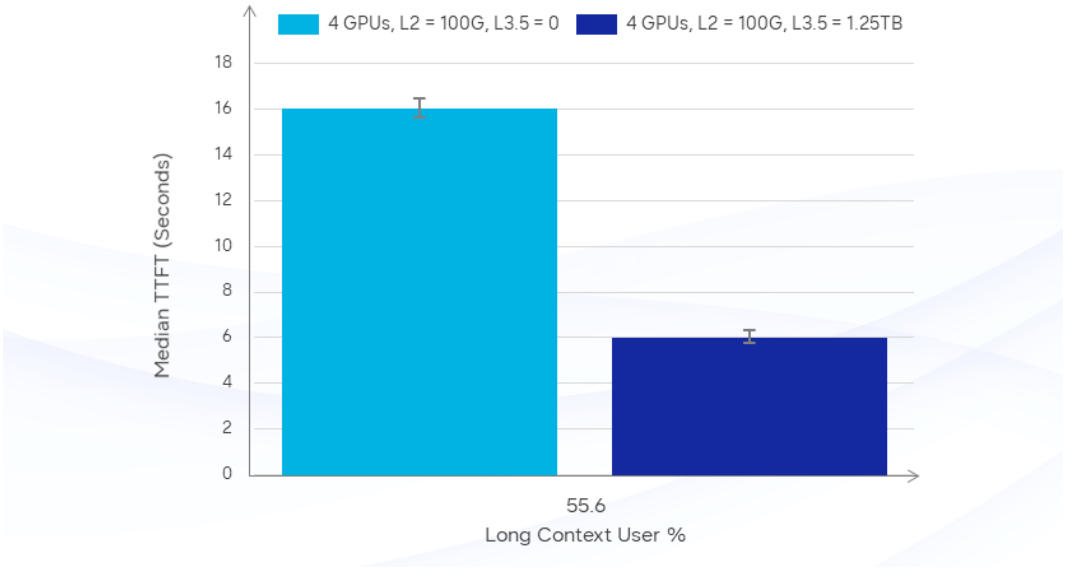


Figure 3. Median TTFT vs. Long-Context User Percentage — 2.6x improvement.

4.2 Median Inter-Token Latency

The joint solution reduces median inter-token latency (ITL) by 69 percent (a 3.1x improvement). This is the most pronounced gain in the benchmark and directly reflects HiCache's ability to keep KV data warm across the L1, L2, and L3.5 tiers under sustained concurrency.

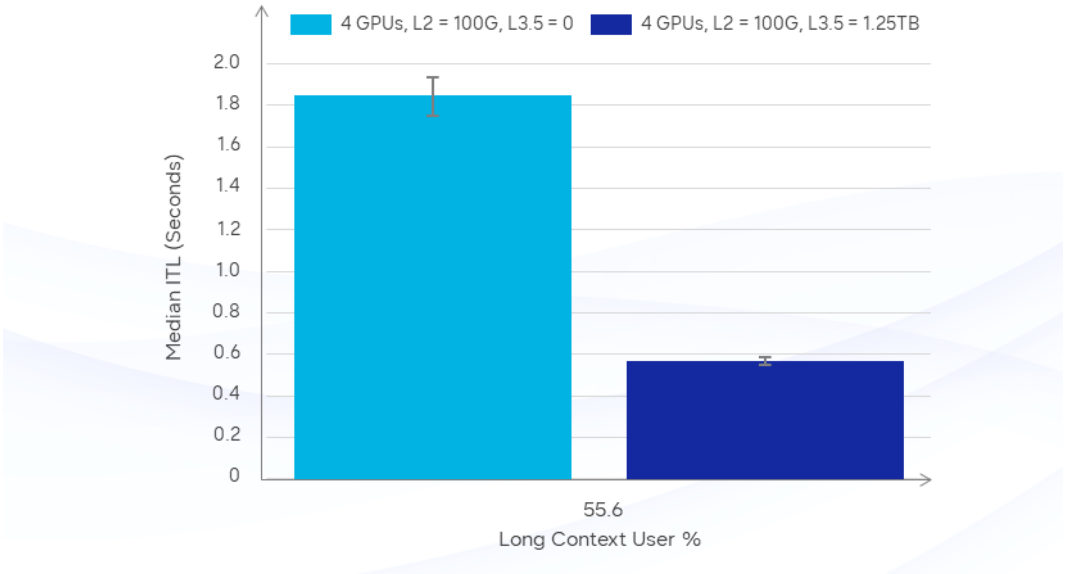


Figure 4. Median Inter-Token Latency vs. Long-Context User Percentage — 3.1x improvement.

4.3 Median End-to-End Latency

End-to-end latency, which combines TTFT and per-token generation time, is reduced by 63 percent (a 2.7x improvement).

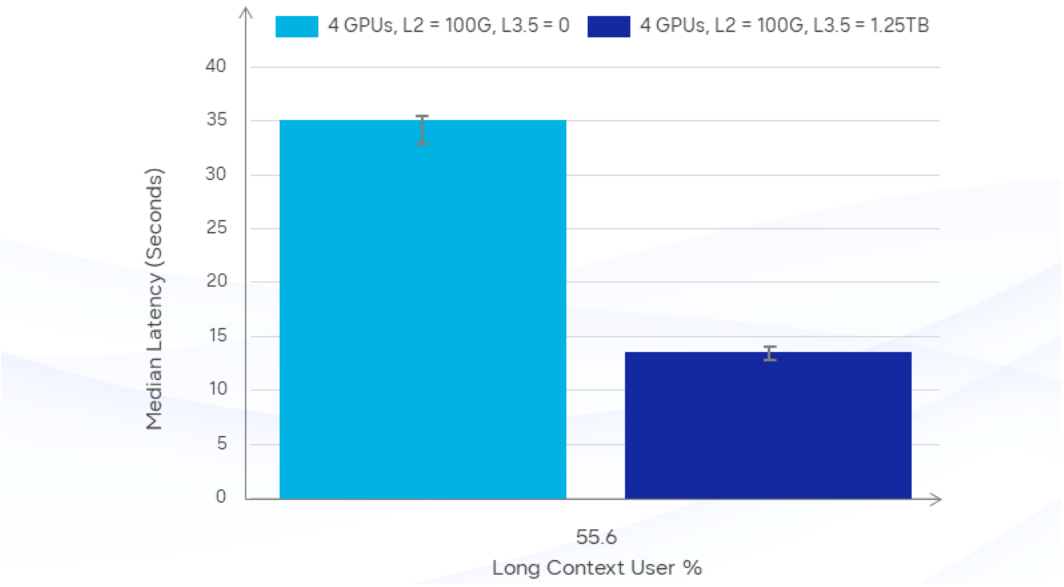


Figure 5. Median End-to-End Latency vs. Long-Context User Percentage — 2.7x improvement.

4.4 Cache Hit Rate

The hierarchical cache achieves a 2.5x higher hit rate than the GPU node-only baseline. This is the underlying mechanism driving the latency and throughput improvements: Prefix-level content that would otherwise be recomputed on every session is served from L2 or L3.5.

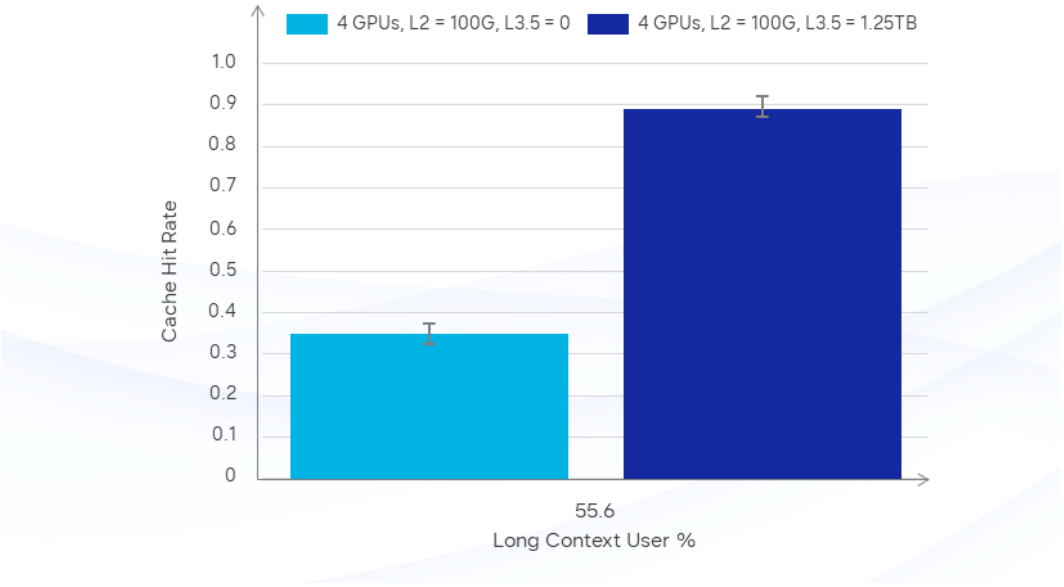


Figure 6. Cache Hit Rate vs. Long-Context User Percentage — 2.5x improvement.

4.5 Input and Output Token Throughput

Both input and output token throughput are 2.2x higher than the baseline, indicating that the GPU is no longer stalled waiting for KV data and is operating closer to its compute ceiling.

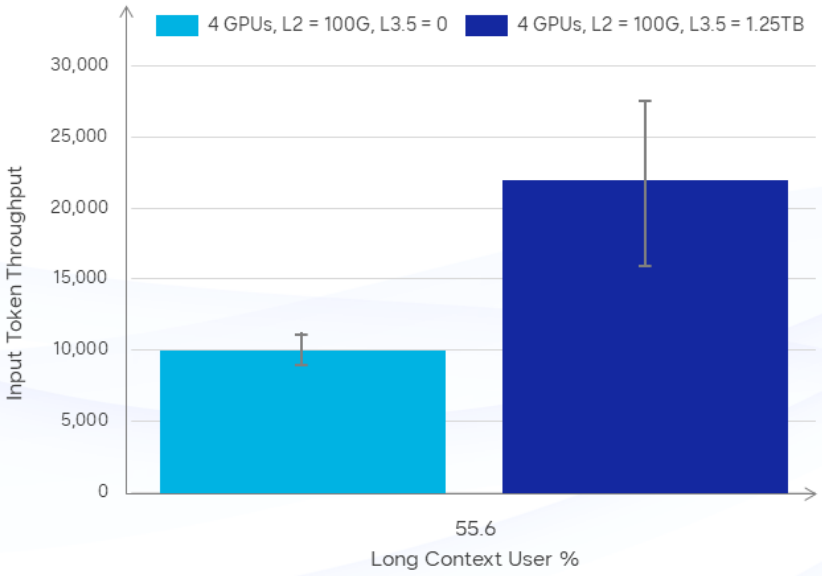


Figure 7. Input Token Throughput vs. Long-Context User Percentage — 2.2x improvement.

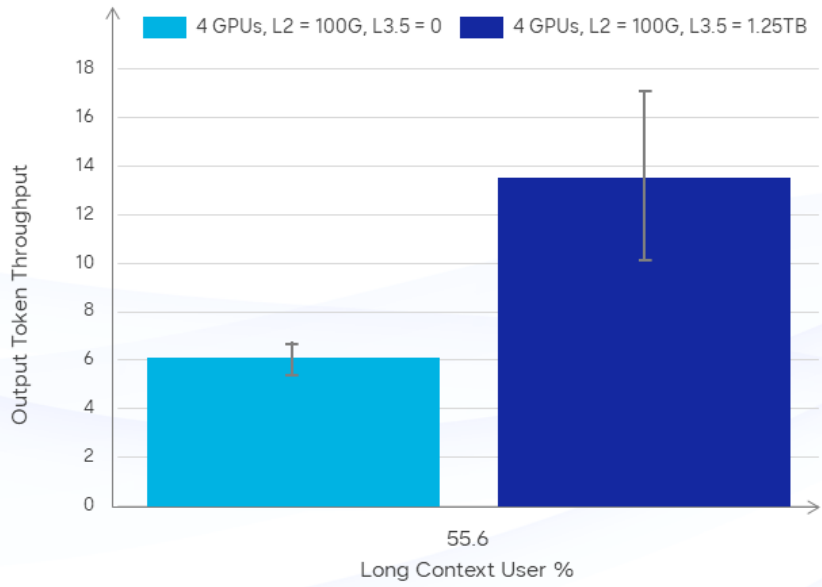


Figure 8. Output Token Throughput vs. Long-Context User Percentage — 2.2x improvement.

5. Conclusion

GPU HBM is a finite resource that constrains the number of concurrent user sessions well below the GPU's compute ceiling. The KV cache, not model weights and not FLOPS, is what fills it. Samsung Cognos and SGLang HiCache together address this constraint at the infrastructure level by extending the memory hierarchy to an AI Memory Node™ built on a standard CPU server. HiCache provides the software-defined multi-tier KV management layer. Samsung Cognos manages the low-latency composable memory hardware that serves as that layer's external tier.

Across the benchmark sweep, the joint solution delivers a 2x to 3.1x improvement in TTFT, ITL, end-to-end latency, cache hit rate, and token throughput. The HiCache integration interface makes the solution reproducible for any organization deploying SGLang. The composable memory backend is a pluggable component, not a one-off integration.

Future work involves expanding beyond the current 120B-class, coding-intensive workload to include models of various sizes and workloads with less KV reuse (such as ad-hoc Q&A, multilingual customer support, streaming summarization).

This investigation will help determine where benefits diminish. The evaluation is based on a stable, non-oversubscribed 100 GbE network, so systems with higher traffic or longer RTTs might experience smaller or more inconsistent improvements, particularly in tail latency. Addressing these scenarios will be part of future solutioning.

References

1. https://semiconductor.samsung.com/about-us/locations/us-rnd-labs/memory-labs/data-fabric-solutions/
2. https://semiconductor.samsung.com/news-events/tech-blog/redefining-possibilities-in-memory-technology-samsung-cxl-memory-appliance-with-orchestration-console/
3. https://semiconductor.samsung.com/news-events/tech-blog/significant-infrastructure-cost-savings-for-ai-fraud-detection-using-high-density-memory/
4. https://download.semiconductor.samsung.com/resources/white-paper/04_01_Latest_Samsung_Cognos_Mgmt_Console_Whitepaper_10139392999591.pdf
5. https://download.semiconductor.samsung.com/resources/white-paper/Usage_Aware_Memory_Auto_Tiering.pdf
6. https://semiconductor.samsung.com/cxl-memory/cmm-d/
7. HiCache - SGLang
8. HiCache System Design and Optimization - SGLang Documentation

About Samsung Semiconductor

Samsung Semiconductor is a global technology leader in advanced memory, logic, and foundry solutions designed for next-generation computing. Our semiconductors power the future of AI, intelligent edge devices, and embedded platforms—delivering high-performance and energy-efficient solutions. Through close collaboration with customers, we help optimize system architectures, accelerate time to market, and enable innovation across various applications, including servers, PCs, mobile and automotive. For more information, please visit semiconductor.samsung.com.

Samsung Electronics Co., Ltd.
1-1, Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do 18448, Korea www.samsung.com 1995-2026