SAMSUNG

CMM-H

CXL-Enabled Large Scale Memory Expansion with a Hybrid NAND/DRAM Solution

White Paper

Samsung Memory Solutions Lab (MSL) Authors:
Reza Soltaniyeh, Gongjin Sun, Caroline Kahn, Hingkwan Huen, Senthil Murugesapandian, Amir Beygi, Andrew Chang, Xuebin Yao, Ramdas Kachare

Table of Contents

SAMSUNG

1. Introduction

The exponential growth of data has significantly increased the demand for applications such as machine learning, Artificial General Intelligence (AGI), and data analytics. These memory-intensive applications have driven up the total cost of ownership (TCO) in data centers due to escalating memory and storage needs. Additionally, DRAM scalability limitations hinder effective vertical scaling, necessitating innovative solutions. To address these challenges, the industry is converging on a coherent interconnect protocol called Compute Express Link™ (CXL™) [1].

CXL, an open standard cache-coherent interconnect between processors and devices, enhances PCIe by incorporating load/store semantics and enabling unparalleled memory expansion. By decoupling the memory controller from the CPU and facilitating attachment of additional memory devices via switches, CXL overcomes traditional CPU-attached memory limitations and enables a trade-off between performance and cost in data centers.

CXL Memory Module Hybrid (CMM-H) leverages CXL's capabilities to address evolving data centers requirements, offering seamless integration with existing infrastructure while unlocking the full range of benefits described above. CMM-H introduces a distinctive memory architecture that combines slower NAND with high-speed DRAM serving as a cache. Equipped with a DRAM cache and cache management unit, CMM-H accelerates data access and reduces reliance on slower backend NAND storage, optimizing performance and cost efficiency.

This white paper explores CMM-H's design, performance, and real-world impact to provide insights into system-level memory performance and architecture. We evaluated CMM-H using micro-benchmarks and a diverse set of real-world applications spanning multiple domains. The goal of this publication is to inspire the development of software and solutions that maximize the benefits of CXL-based memory expansion through CMM-H.

Samsung Memory Solutions Lab (MSL)

[1] Compute express link (cxl). https://computeexpresslink.org/

2. CMM-H Overview

I. Samsung Memory

CMM-H is a type 3 (CXL.io and CXL.mem) CXL-based memory solution designed for large-scale memory expansion, achieved by integrating a hybrid architecture combining NAND and DRAM modules. To address latency and data granularity mismatch between host requests (64 bytes) and SSD block sizes (4 KB), CMM-H employs internal DRAM as a cache for NAND-resident data. Figure 1 (a) shows CMM-H's high-level architecture.

The key components of CMM-H include NVMe SSD and DRAM, orchestrated by an internal cache controller that manages the device cache. The cache controller implements a caching mechanism including baseline cache replacement and eviction policies. CMM-H communicates with the host via PCIe-CXL interface, with the CXL End Point (EP) on CMM-H receiving and managing the transactions belonged to CXL.mem and CXL.IO protocols. Acting as the intermediary between the host and device cache controller, the CXL EP transmits the incoming memory requests to the cache controller logic via internal interfaces.

The host-requested data can be served either from the device cache (DRAM) or the SSD (NAND). The host requests may follow a distinct data path, based on where the data is present. Figure 1 (b) illustrates various data paths for incoming read/write requests. For read requests, CMM-H device cache controller initially checks whether the requested data resides in the device cache. If the data is found in the cache, the data is promptly served to the host so no SSD access is required. However, if the requested data is not cached, the cache controller initiates retrieval from the backend SSD before responding to the request. If necessary, the device cache controller may evict certain data blocks from the device cache to make room for the new data and write those evicted blocks back to SSD.

Likewise, for write requests, the CMM-H device cache controller verifies availability of the data in the cache. If the data is not present in the device cache, and there is insufficient cache space for new data, the cache controller first evicts some of the existing pages to the SSD to make room for the incoming data. Subsequently, the new data is written into the device cache for future references.

Finally, for both read/write requests, the device cache controller responses are sent to the CXL EP to communicate with the host. The device cache is managed internally, and it is transparent to the host. In other words, the host only sees the advertised capacity (i.e. SSD capacity), and is not aware of the underlying caching. The main benefit of this approach is that the CMM-H can be used by applications seamlessly, and all the details about caching between tiers are abstracted away from the host.

CMM-H's baseline caching mechanism may cater to many applications/use cases. CMM-H also offers flexibility allowing the host application to control/influence the device caching management directly when such requirements are needed. API provided for CMM-H enables direct control over the device cache, allowing the host to provide hints to the cache controller regarding prefetch and eviction of specific pages. Host applications can also query the status of the pages and their temperature (i.e., how frequently they are accessed). This API enables application engineers to leverage their insights from the application's memory access patterns and to use those insights to improve overall performance by optimizing CMM-H internal caching usage.

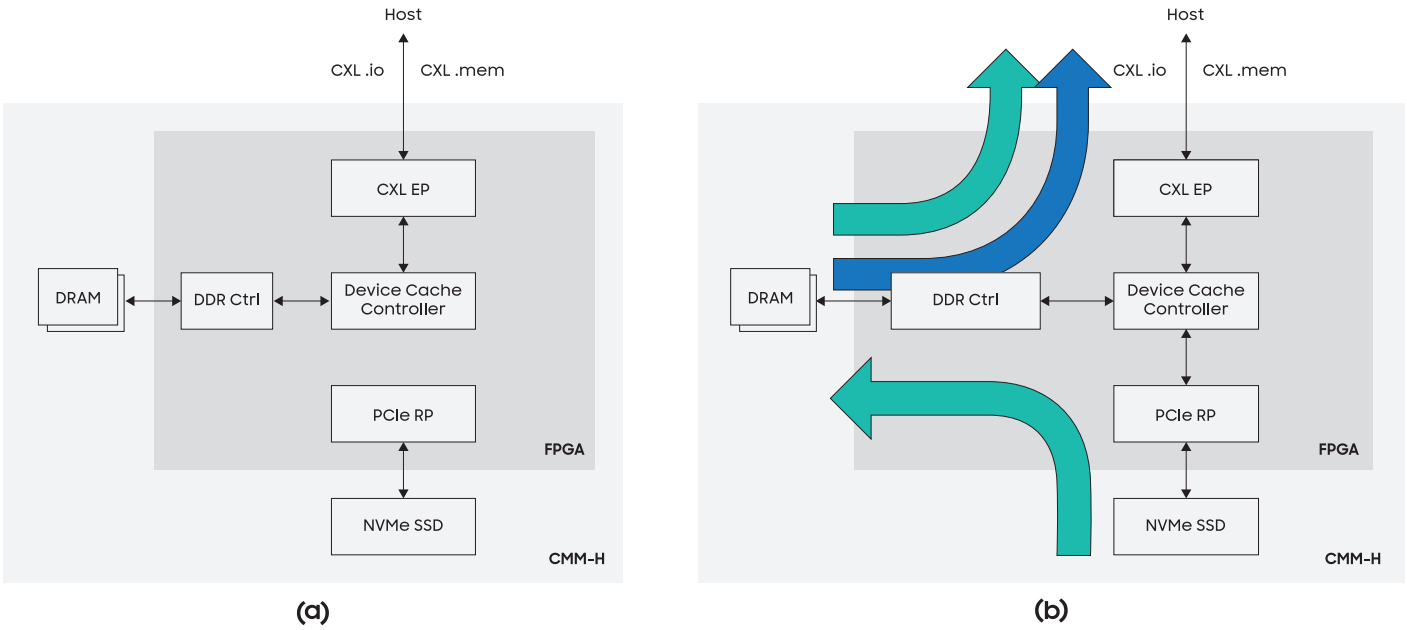

Figure 1. **(a)** The architecture of CMM-H, highlighting the CXL End Point (EP), device cache controller, and PCIe root port connecting to the NVMe SSD. The two memory modules include DRAM (used as a cache) and NVMe SSD. **(b)** Data paths for handling incoming host requests: the light blue path represents a device cache hit scenario, while the dark blue path illustrates a cache miss where the data is retrieved from the NVMe SSD first.

In summary, CMM-H's versatile cache mechanisms cater to diverse use cases. It offers a host-agnostic caching approach and host-managed options, enabling a wide range of use cases. This flexibility empowers users to leverage their proprietary software and application insights, enhancing overall efficiency of the device cache.

II. Device Configuration

CMM-H can advertise up to the total capacity of SSD. In this evaluation, a device with 1TB of advertised memory was used. Table 1 outlines the full CMM-H device configurations including details of device memory, host interface, and form factor.

Table 1. CMM-H configuration

Parameter	Description
Advertised Capacity	1TB
Host Interface	PCIe Gen5x8, CXL Type 3 (.IO and .Mem)
Device Memory	NVMe SSD Gen4x4, 48GB DDR4
Device FPGA	Intel Altera Agilex 7, Quad-core Arm Cortex-A53
Form Factor	Add-in Card (AIC), Single Slot, Full Height, 3/4 Length

III. Experimental Evaluation

We used different types of micro-benchmarks and real-world applications to evaluate overall performance of CMM-H.

A. Platform Configuration

The evaluation was conducted on an Intel production server with CXL support (Table 2). The dual-socket system was equipped with DDR5 memory operating at 5600 MT/s and one CMM-H device per socket, appearing as a CPU-less NUMA node.

All experiments were conducted on Linux Ubuntu 24.04 running kernel 6.11, using an off-the-shelf CXL driver and Linux's memory tiering feature (numactl utility) for NUMA node management. This provides basic policies such as membind (only allocate from a specific numa node) and weighted interleaving. In addition to numa node, CMM-H can also be integrated in the system as a Direct Access (DAX) memory, which has not been explored in this paper.

Table 2. Platform configuration

Parameter	Description
CPU	Intel(R) Xeon(R) 6710E, 2.4GHz
Cache	L1d: 4MiB, L2: 128MiB, L3: 192MiB
Main Memory	8 DDR5 (5600 MT/s), peak theoretical DDR bandwidth: 358GB/s
Device FPGA	Ubuntu 24.04, Kernel 6.11

B. Bandwidth and Latency Analysis of CMM-H

Table 3 summarizes CMM-H's key performance metrics, including peak memory bandwidth and latency. Due to its heterogeneous design, CMM-H exhibits latency and bandwidth variability influenced by working set size and access patterns. Figure 2 shows memory bandwidth trends across working set sizes, measured using 32 CPU cores. Bandwidth peaks within the 4 GB to 26 GB range, aligned with the capacity of device cache, where elevated cache hit rates significantly enhance performance. Beyond this range, bandwidth gradually stabilizes at approximately 5 GB/s. Nevertheless, even with larger working sets, the device cache effectively mitigates the slower backend SSD, maintaining high cache hit rates between 85% to 95%.

Table 3. Overview of CMM-H latency and bandwidth metrics

Bandwidth	Latency
Peak: 27GB/s	Average: 553 ns Median: 505 ns 99.9%: 1547 ns

Table 4. Memory bandwidth for different DRAM: CMM-H memory interleaving ratios

DRAM Ratio	CMM-H Ratio	All-Read (GB/s)	3R:1W (GB/s)	2R:1W (GB/s)	1R:1W (GB/s)
1	0	161	151	149	145
2	1	77	50	41	48
3	1	105	69	60	62
6	1	156	120	111	114
10	1	168	157	156	152

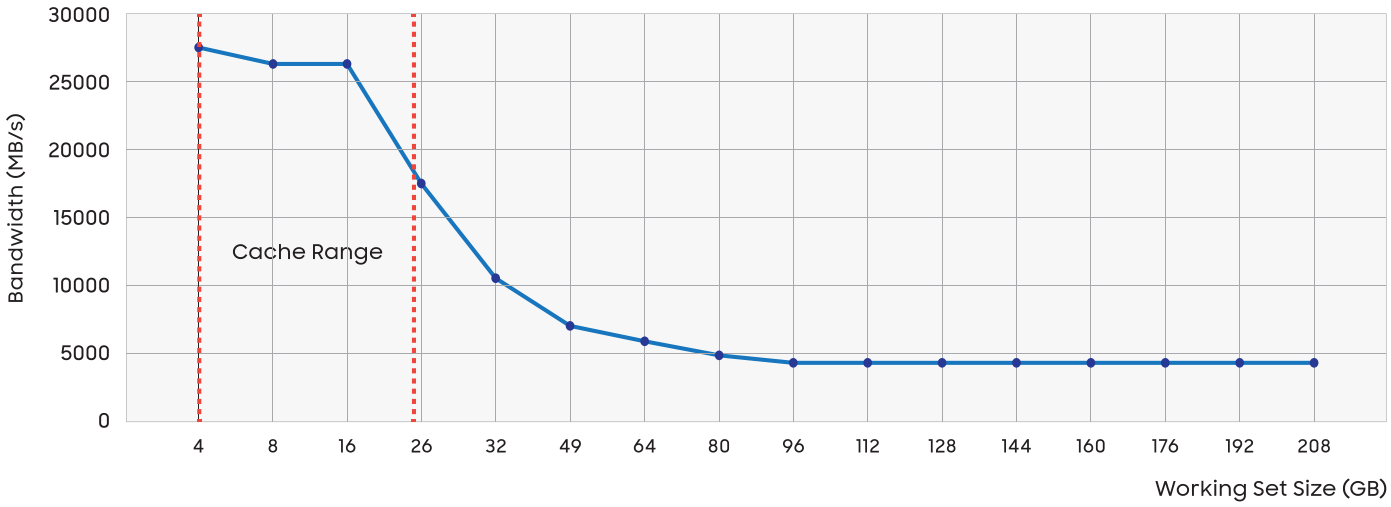

Figure 2. CMM-H bandwidth across different working set sizes.

C. System-Wide Memory Performance with CMM-H

Table 4 displays memory bandwidth for a local socket comprised of both DRAM and CMM-H memory. Utilizing the Linux kernel's weighted interleaving feature, we measured memory bandwidth across various DRAM-to-CMM-H ratios. One of the significant findings is that CMM-H enhances system's peak memory bandwidth by 4% (for 10 to 1 DDR to CMM-H ratio). Additionally, CMM-H delivers substantial memory capacity through its heterogeneous NAND+DRAM configuration. Even with a more aggressive CMM-H adoption (e.g., 3:1 DRAM-to-CMM-H ratio), the bandwidth decreases by only 30%, compared to DRAM-only cases. To assess the performance impact of CMM-H, we measured system-wide memory bandwidth and latency, using a Memory Latency Checker (MLC) [3] benchmark. Specifically, we incrementally increased the delay between memory requests, simulating varying levels of memory contentions and loads. This approach allows us to evaluate how bandwidth and latency respond as the system transitions from idle to saturated states, providing a comprehensive view of performance under different workload intensities.

[3] Intel. Memory latency checker v3.11a.
https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html, 2024.

Figure 3 presents a system-wide latency versus bandwidth plot, comparing two configurations with and without CMM-H. As illustrated, under heavy loads, the system with CMM-H demonstrates lower latency at equivalent memory bandwidth levels, compared to a DRAM-only system. Another notable observation is that memory latency initially decreases as the load increases before rising again, exhibiting a U-shaped trend.

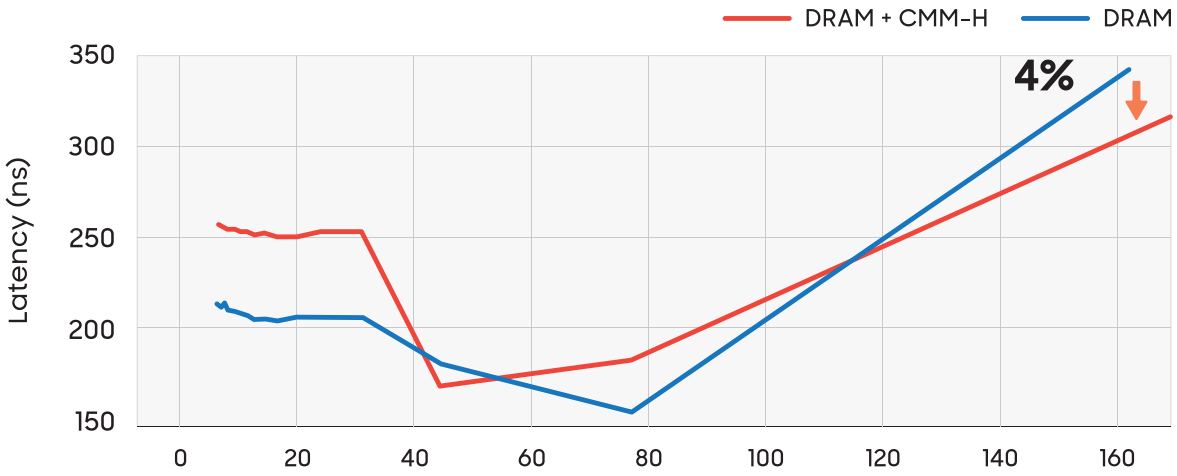

Figure 3. Latency vs. bandwidth comparison: DRAM-only Vs. DRAM+CMM-H integration.

We attribute this behavior to the memory controller's ability to manage operations more efficiently at intermediate loads, optimizing request spacing to minimize queuing delays and enhance scheduling. Furthermore, the integration of CMM-H boosts peak memory bandwidth by 4%, compared to a DRAM-only configuration, underscoring its performance advantages.

D. Real-world Applications

We evaluated CMM-H's impact through real-world applications, focusing on its end-to-end performance. Our goal is to evaluate the effectiveness of CMM-H achieving two goals: (1) expanding system memory without significantly compromising performance and (2) enhancing memory bandwidth available to CPU through CXL. We selected applications from diverse domains, including in-memory databases, high-performance computing (HPC), and graph analytics.

I. Redis In-memory Database

To benchmark Redis, we used Memtier [4], a widely recognized standard evaluating performance of Redis. We measured performance using operations per second for GET and SET commands, alongside p99 latency, as our key metrics.

Figures 4 illustrates the speedup across various DRAM and CMM-H interleave configurations, with all speedups normalized to the DRAM-only experiment. The result shows only a 10% drop in throughput with a 1:1 DRAM to CMM-H ratio. At a 3:1 DRAM-to-CMM-H ratio, performance reaches 95% of the DRAM-only baseline. Additionally, SET and GET operations exhibit comparable performance. For latency, CMM-H is 30% slower than DRAM due to the higher latencies of the accesses that miss the device cache and reach the backend SSD.

[4] Redis Lab. Memtier benchmark. https://github.com/RedisLabs/memtier_benchmark, 2024.

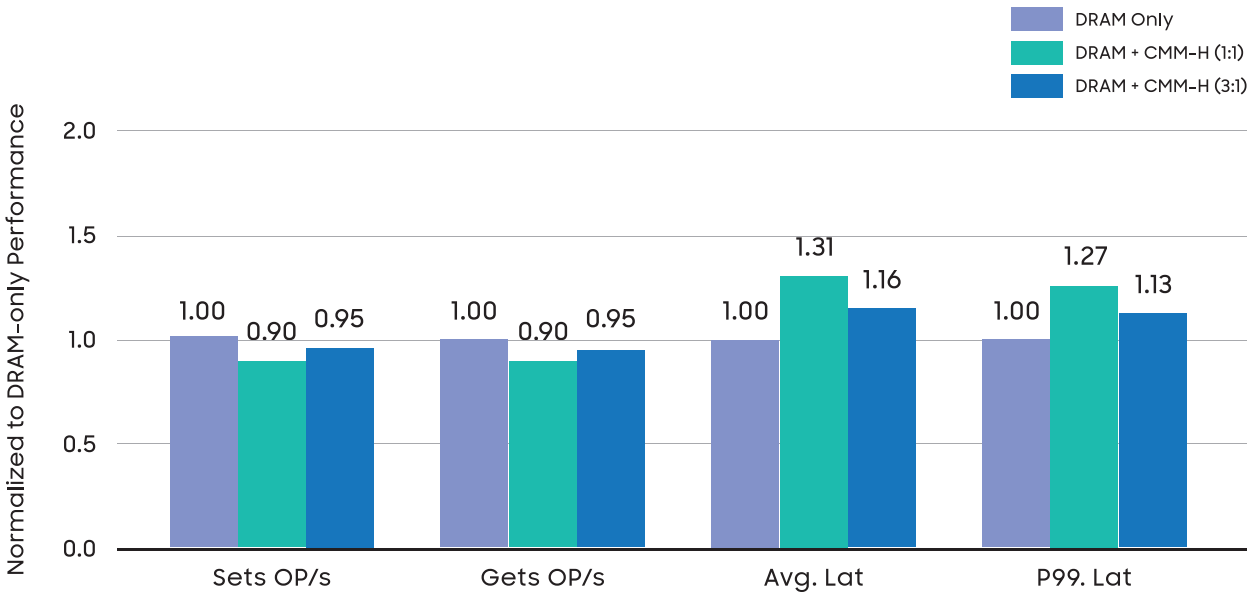

Figure 4. Benchmarking Redis performance using the Memtier benchmark across various memory configurations with DRAM and CMM-H.

II. NAS Parallel Benchmark

We utilized NASA Advanced Supercomputing (NAS) Parallel Benchmarks (NPB) [2] as a proxy for memory-intensive high-performance computing (HPC) applications. For our benchmarking, we selected two kernels: Conjugate Gradient (CG), characterized by irregular memory access patterns and significant communication, and memory-intensive Multi-Grid (MG) that operates on a sequence of meshes. Each kernel has a predefined problem size. In our experiments, we used Class E for both, employing the OpenMP implementation with a max of 64 threads to parallelize execution. Figure 5 (a) illustrates performance across various DRAM-to-CMM-H interleave configurations, with speedup results normalized to a DRAM-only system. The performance outcomes differ significantly between the two kernels. For the CG kernel, incorporating DRAM-CMM-H yields a higher speedup than the DRAM-only system across all three configurations, with a higher DRAM-to-CMM-H ratio slightly enhancing performance (up to 43% speedup). Conversely, the MG kernel experiences substantial performance degradation with DRAM-CMM-H interleaving, although a higher DRAM-to-CMM-H ratio markedly reduces this decline. One possible explanation for this performance drop is suboptimal data placement across NUMA nodes, leading to significant latency penalties that outweigh bandwidth advantages of CXL-based memory expansion.

III. Graph500

Graph500 [5] is a data-intensive workload benchmark. It begins by constructing an undirected graph, followed by a breadth-first search (BFS) on that graph. For performance evaluation, we focus solely on the BFS step as the graph construction phase is performed only once. Since all experiments in this study are conducted on a single server, we configured the problem size to approximate the benchmark's defined mini problem set, which requires roughly 100 GB of memory in total.

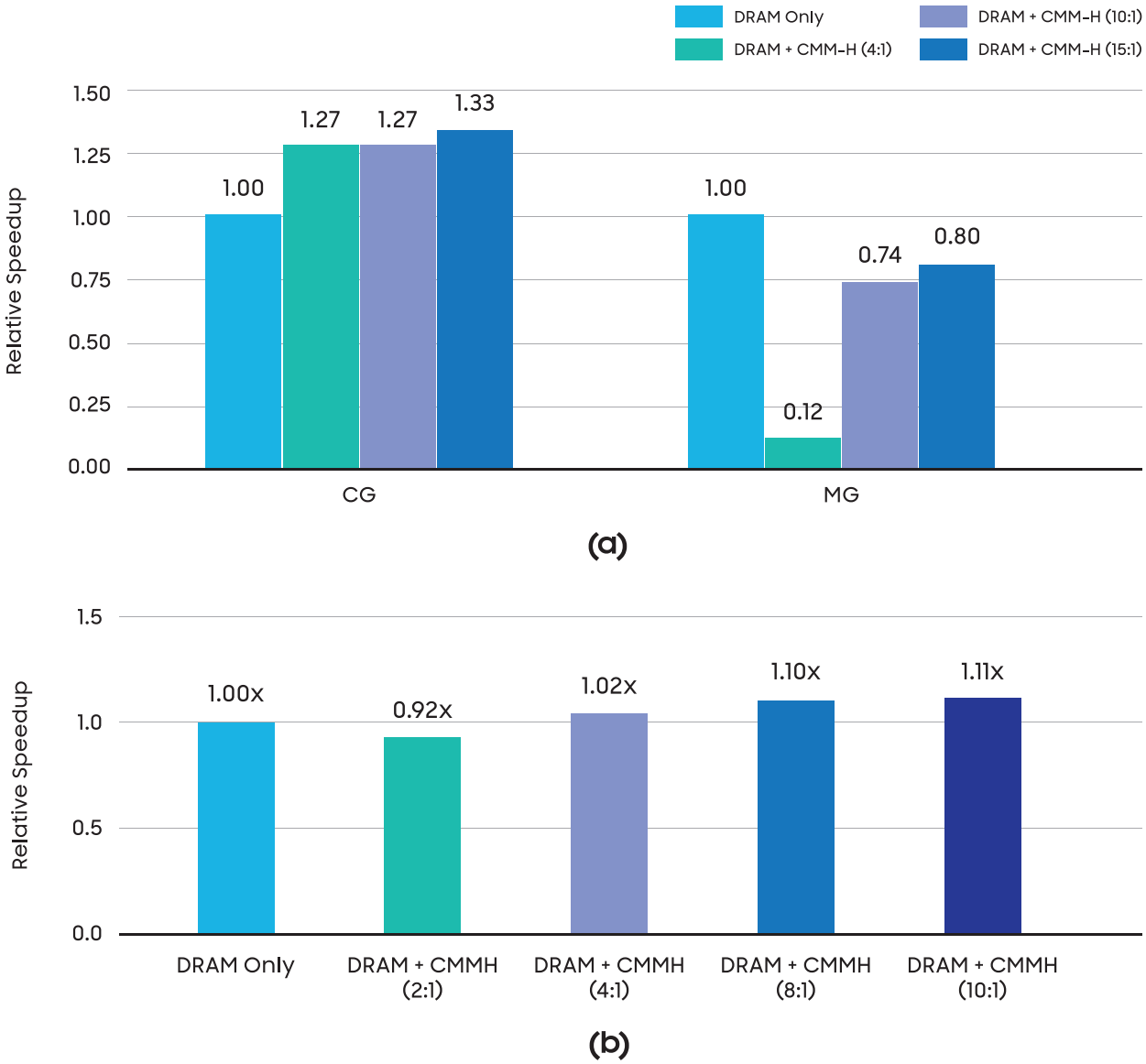

Figure 5. Normalized speedup of two memory-intensive NAS benchmarks (CG and MG) and Graph500-BFS, normalized to DRAM-only baseline.

Figure 5(b) compares the speed up of Graph500 in a system with and without CMM-H. The results show an 8% performance decrease when employing a 2:1 DRAM-to-CMM-H ratio. In contrast, a higher DRAM-to-CMM-H ratio delivers approximately a 10% performance improvement over a DRAM-only system.

[2] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga. The NAS parallel benchmarks —summary and preliminary results. In Proceedings of the 1991 ACM/IEEE Conference on Supercomputing, Supercomputing '91, page 158–165, New York, NY, USA, 1991. Association for Computing Machinery.

[5] Koji Ueno and Toyotaro Suzumura. Highly scalable graph search for the graph500 benchmark. In Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing, HPDC '12, page 149–160, New York, NY, USA, 2012. Association for Computing Machinery.

IV. Conclusion

CMM-H represents a breakthrough in cost-effective, large-scale memory expansion, leveraging advanced CXL interconnect capabilities to address the growing demands of modern data-intensive applications. By combining high-speed DRAM with cost-effective NAND and offering flexible cache management options, CMM-H delivers substantial memory capacity improvements without compromising performance. Experimental evaluations confirm that CMM-H enhances system bandwidth, optimizes latency under heavy workloads, and minimizes performance trade-offs, even when operating with mixed DRAM-to-CMM-H configurations. This seamless integration into existing infrastructures, combined with transparent host interaction and optional direct cache control, positions CMM-H as a versatile solution for next-generation system memory architecture constantly seeking to balance memory scalability, performance, and TCO reduction.

References

[1] Compute express link (cxl). https://computeexpresslink.org/.

[2] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga. The nas parallel benchmarks—summary and preliminary results. In Proceedings of the 1991 ACM/IEEE Conference on Supercomputing, Supercomputing '91, page 158–165, New York, NY, USA, 1991. Association for Computing Machinery.

[3] Intel. Memory latency checker v3.11a.
https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html, 2024.

[4] Redis Lab. Memtier benchmark. https://github.com/RedisLabs/memtier_benchmark, 2024.

[5] Koji Ueno and Toyotaro Suzumura. Highly scalable graph search for the graph500 benchmark. In Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing, HPDC '12, page 149–160, New York, NY, USA, 2012. Association for Computing Machinery.

For more information

For more information about the Samsung Semiconductor products, visit semiconductor.samsung.com.

About Samsung Electronics Co., Ltd.

Samsung Electronics Co. Ltd inspires the world and shapes the future with transformative ideas and technologies. The company is redefining the worlds of TVs, smartphones, wearable devices, tablets, digital appliances, network systems, memory, system LSI and LED solution. For the latest news, please visit the Samsung Newsroom at news.samsung.com.

Samsung Electronics Co., Ltd.
129 Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 16677, Korea **www.samsung.com** 1995-2025

SAMSUNG