HotStorage 2025

Revisiting Memory Hierarchies with CMM-H: Using Device-side Caching to Integrate DRAM and SSD for a Hybrid CXL Memory

Mohammadreza Soltaniyeh, **Gongjin Sun**, Xuebin Yao, Amir Beygi, Ramdas Kachare, Dongwan Zhao, Hingkwan Huen, Andrew Chang, Senthil Murugesapandian, Caroline Kahn

Memory Solutions Lab
Samsung Semiconductor Inc.

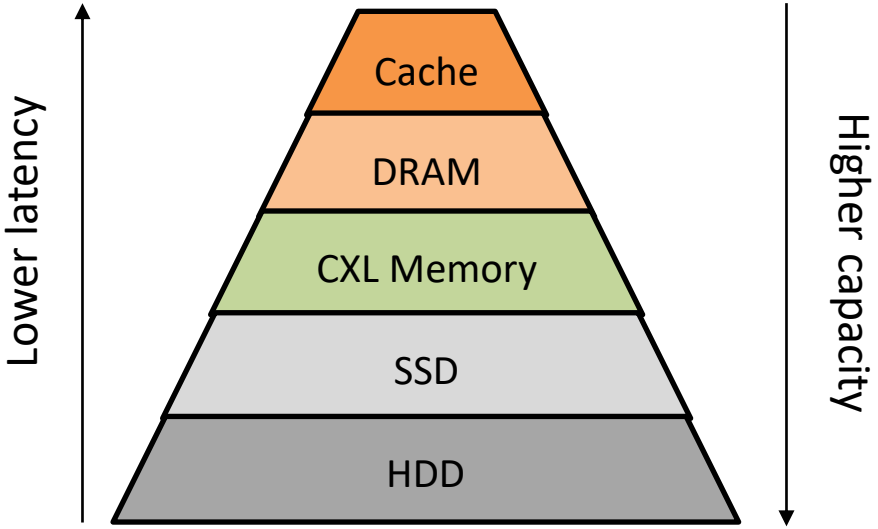
HotStorage 2025, July 10-11, Boston, MA

The Memory Capacity & Cost Challenge

- Modern workloads (AI, AGI, In-Memory DBs) demand massive memory.

- DRAM is fast, but faces significant limitations:
  - High cost per gigabyte.
  - Scalability challenges (power, density, thermal).
  - Increasing Total Cost of Ownership (TCO) for data centers.

Emerging Memory Hierarchy With CXL



CXL: Bridging the Memory-Storage Gap

CXL: A New Paradigm for Memory Expansion

- Compute Express Link (CXL) decouples the memory controller from the CPU.
- Extends memory semantics (load/store) to external devices.
- **Current CXL Devices:**
  - DRAM-only memory expanders.
  - Successfully increase capacity and bandwidth.
  - **But** still rely exclusively on expensive DRAM w/ dedicated CXL controller
    - inheriting its cost and capacity limitation issues.
- **Question:** How can we leverage CXL to achieve *both* massive capacity and cost-effectiveness?

Our Solution: The CMM-H Hybrid Module

- **Core Concept:** A single CXL device integrating:
  - A small, fast DDR DRAM cache.
  - A large, cost-effective NAND flash backend.

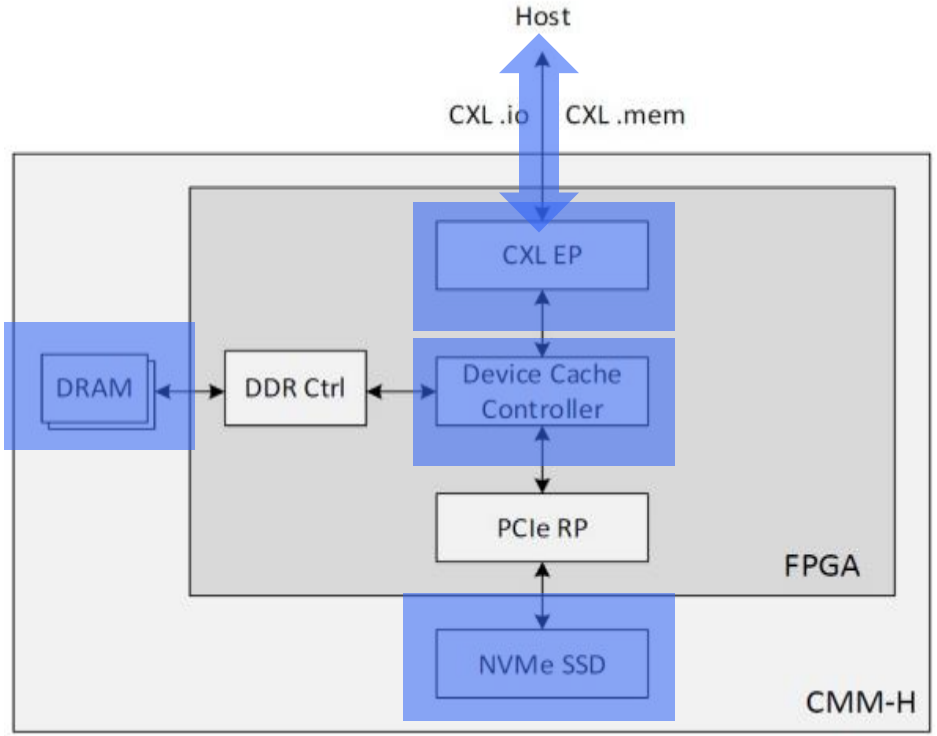
- **Goal:** Expose terabyte-scale capacity while device-side caching delivers high performance.

CMM-H Architecture

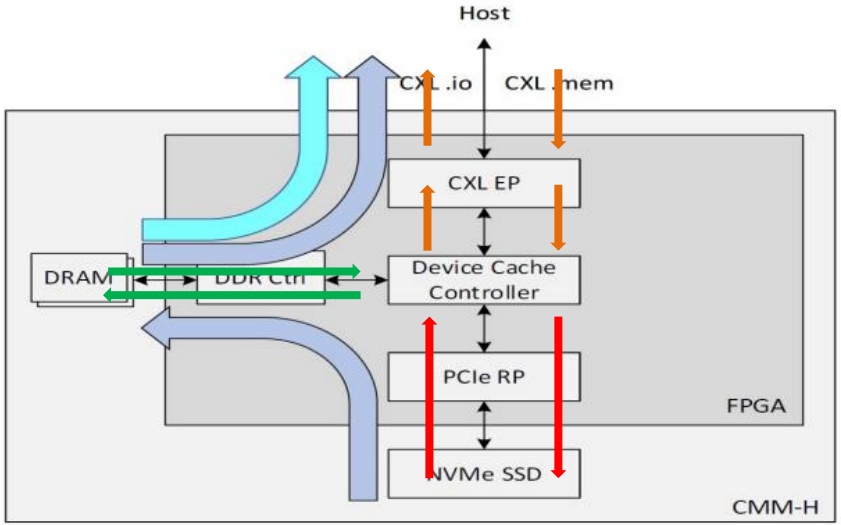
- **Host Interface:** PCIe Gen5x8, CXL 1.1 Type 3.

- **Device Memory:**
  - 48 GB DDR4 DRAM (Cache)
  - 1 TB off-the-shelf NVMe SSD (Backend Storage, 2/4 TB and 2 SSDs are supported as well)

- **Controller:**
  - FPGA-based CXL End Point (EP).
  - **Internal Cache Controller** manages data movement.
  - Policy: 8-way LRU replacement, Write allocate/Write-back.



Operational Workflow: Hits and Misses



1.Host issues a memory request (Read/Write).

2.Cache Hit (Light Blue Path): Data is in DRAM cache. Fast response.

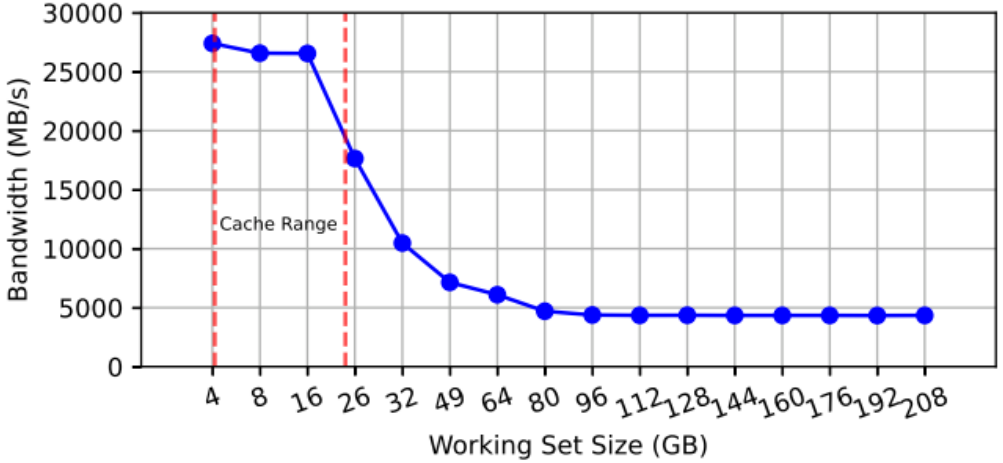
3.Cache Miss (Dark Blue Path): Data is fetched from the SSD backend and placed into the DRAM cache.

Experimental Setup

- **Host System:**
  - CPU: Intel Xeon 6710E (Dual-socket)
  - Local Memory: DDR5 @ 5600 MT/s
  - OS: Ubuntu 24.04 (Kernel 6.11)
  - Management: numactl for memory tiering/interleaving

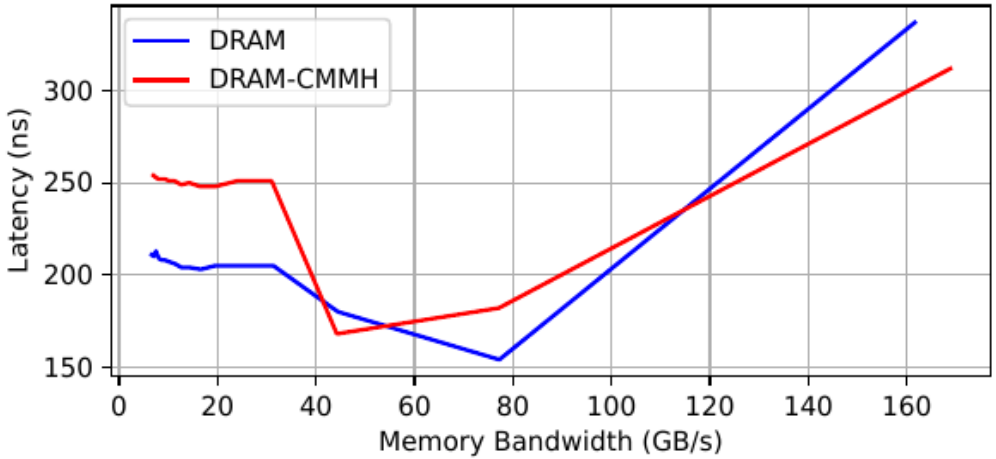
- **CMM-H Device:**
  - Altera Agilex 7 FPGA, Advertised Capacity: 1 TB (up to 4 TB)
  - Cache: 48 GB DDR4
  - One CMM-H device per CPU socket, appearing as a CPU-less NUMA node.

Microbenchmark: Bandwidth vs. Working Set Size



- **Peak Bandwidth (~27 GB/s):** Achieved when the working set fits within the 48 GB DRAM cache.
- **Graceful Degradation:** Bandwidth decreases as the working set exceeds cache size.
- **Stable Miss-driven Performance (~5 GB/s):** The bandwidth stabilizes once the working set is much larger than the cache.

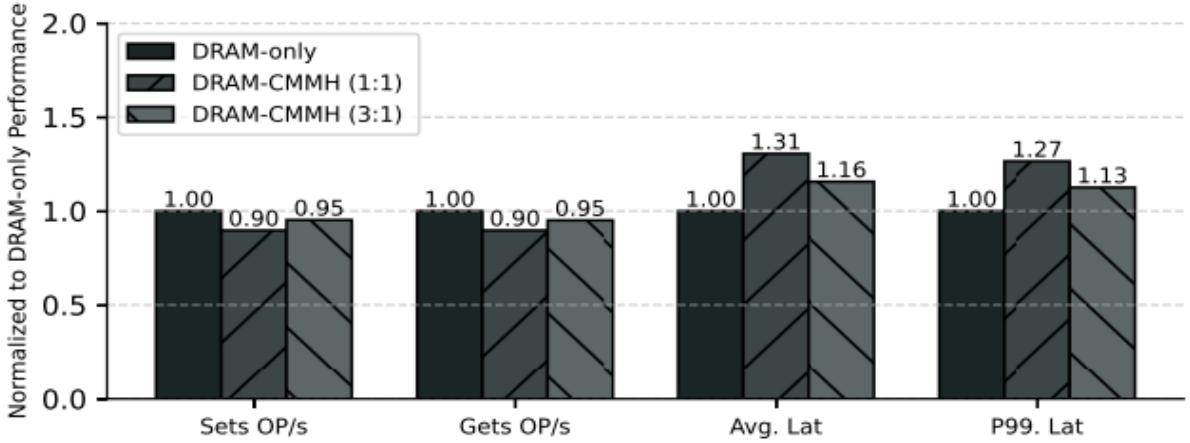
System-Level: Latency vs. Bandwidth



- **Comparison:** DRAM-Only vs. DRAM + CMM-H (10:1 interleave)

- **Key Finding:**
  - Combined system achieves **4% higher peak bandwidth**.
  - At high bandwidth levels, the combined system exhibits **lower average latency** due to better request scheduling.

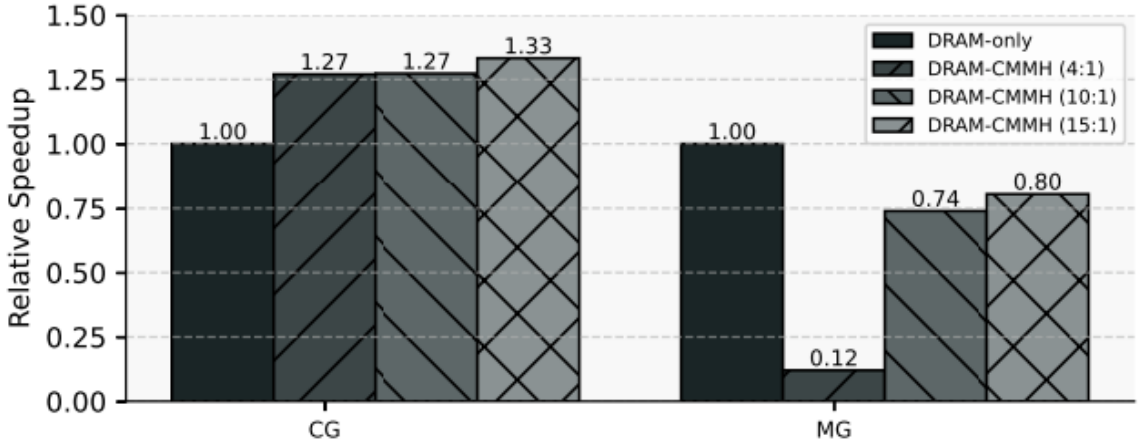
Real-World App 1: In-Memory Database (Redis)



- **Key Results (Normalized to DRAM-Only):**
  - **3:1 DRAM-to-CMM-H Ratio:** Achieves **95%** of the baseline throughput.
  - **1:1 DRAM-to-CMM-H Ratio:** Achieves **90%** of the baseline throughput.

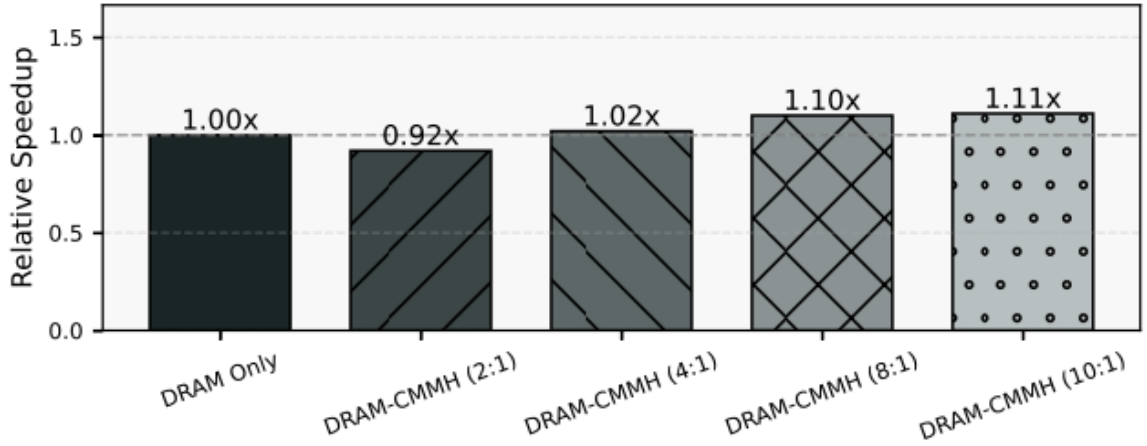
- **Latency:** p99 latency increases by about 30% due to cache misses.

Real-World App 2: HPC (NAS Parallel Benchmarks)



- **Two Kernels, Two Stories:**
  - **Conjugate Gradient (CG):** Performance **improves** by up to 33% with CMM-H due to increased memory bandwidth.
  - **Multi-Grid (MG):** Performance degrades due to latency sensitivity and sub-optimal NUMA placement.

Real-World App 3: Graph Analytics (Graph500)



- **Workload:** Breadth-First Search (BFS) on a 100 GB graph.
- **Key Results:**
  - **2:1 DRAM-to-CMM-H Ratio:** A minor 8% performance decrease.
  - **10:1 DRAM-to-CMM-H Ratio:** A **10% performance improvement** over the DRAM-only system.

Summary of Contributions

1. **We present CMM-H,** a novel hybrid CXL memory architecture integrating DRAM and SSD.
2. **We demonstrate its effectiveness:** The device-side cache successfully hides SSD latency.
3. **We show CMM-H can augment system bandwidth,** boosting performance for certain applications.
4. **We provide insights into tiered memory performance,** highlighting the importance of access patterns and NUMA-awareness.

Conclusion & Future Work

- **Conclusion:**
  - CMM-H represents a viable and cost-effective path toward terabyte-scale, high-performance memory expansion.
  - It paves the way for more flexible and composable memory hierarchies.

- **Future Directions:**
  - Evaluating multi-CMM-H device configurations.
  - Exploring advanced, application-aware caching policies.
  - Leveraging the host API for direct application control (cache prefetching and evicting).

Thank You

Questions?

Learn more about CMM-H

Gongjin Sun, gongjin.s@samsung.com