Toward High-Fidelity Heterogeneous Memory System Modeling in gem5

Maryam Babaie, Ayaz Akram, and Jason Lowe-Power
Computer Science Department, University of California Davis

HPC systems will employ various memory technologies to meet applications demands. The ability to model heterogeneous systems (from both compute and memory perspectives) makes gem5 a highly suitable tool to evaluate future HPC systems. In this presentation, we'll discuss new contributions to gem5 that are extending its modeling support for heterogeneous memory systems. Specifically, we show the new DRAM cache model, the improved HBM model in gem5, and the refactoring of its memory controller models. Systems like Intel's Knights Landing (heavily employed in HPC centers) are configured to use high bandwidth memory as a cache to main memory. Anticipated disaggregation of memory resources also necessitates using local DRAMs as a cache to a large pool of remote memory. Similarly, HBM devices are expected to be a vital contributor to exascale systems. Therefore, we believe this work enables gem5 to help design these future HPC systems.

We'll describe a cycle-level unified DRAM cache controller, *UDCC,* for gem5. The protocol is inspired by the actual hardware providing DRAM cache, such as Intel's Cascade Lake, in which a DRAM cache is backed by an NVRAM. Our controller model implements the timing and micro-architectural details enforced by the DRAM and NVRAM memory interfaces. Like Intel's Cascade Lake design, we implement a DRAM cache model that is direct-mapped, insert-on-miss, and write-back. We aim to provide the ability to model cache and its backing store using various DRAM technologies, like Intel's Sapphire Rapids where HBM2 and DDR5 are used in the Cache-Mode.

The existing HBM model in gem5 follows HBM1 specifications and has several limitations: 1) lack of asymmetric read/write timings, 2) supports only single DRAM interface per controller (inability to model HBM2 pseudo channels), 3) lack of separate row and column command bus bandwidth checks. We implemented an HBM-specific memory controller in gem5 capable of controlling two DRAM interfaces and supporting the use of a shared or partitioned (among pseudo channels) queues and provided the ability for separate read/write DRAM timings. Moreover, we modified the memory controller model of gem5 to improve its modularity and readability, making the extension process convenient. We have restructured the memory to facilitate the maintenance and insertion of additional features for memory systems. These changes are going to be included in the latest gem5's release.

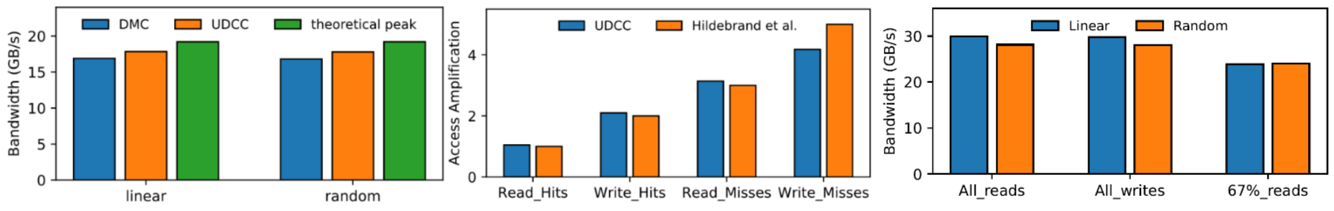

Figure 1. Comparison of observed bandwidth by LLC in DMC vs. UDCC (left), the comparison of access amplification of UDCC protocol vs Intel's Cascade Lake (middle), and observed bandwidth by LLC from two pseudo channels of HBM2 using HBM controller (right).

Evaluation: (A) We verified our DRAM cache model against the analysis of Intel's Cascade Lake studied by Hildebrand et. al.[1] First, we compared the effective bandwidth (observed by the last level cache (LLC)) for gem5's default memory controller (DMC) and UDCC. We used a synthetic read traffic pattern such that all requests will be cache hits in UDCC. Figure 1 (left) compares these bandwidth numbers to the theoretical peak of DDR4 (act as DRAM cache) and shows the similarity in observed bandwidth (controllers' scheduling policy explains the slight differences). Moreover, we calculate the access amplification values for UDCC by dividing the effective bandwidth by the sum of the average bandwidth of DRAM and NVRAM devices for a particular run. The comparison of the two access amplification values is shown in Figure 1 (middle). Our results match the actual hardware in all cases with only one exception, write misses. In actual hardware, on a write miss in DRAM cache, the block is first allocated by reading it from NVRAM and then writing it into DRAM. The actual data is then written into the DRAM. We merge these two writes; thus, our model leads to one less access than the actual hardware for a write miss. (B) We performed traffic generator-based studies using different linear/random read/write traffic combinations to validate the HBM2 pseudo channel model. Figure 1 (right) presents the bandwidth numbers for a single physical HBM2 channel (made of two pseudo-channels), controlled by an HBM controller (peak theoretical bandwidth is 32GB/s). These numbers are comparable to other memory simulation tools like DRAMSys. We will discuss the implementation details and extend evaluation of these models in our presentation at the workshop.

[1]M. Hildebrand, J. T. Angeles, J. Lowe-Power, and V. Akella, "A case against hardware managed dram caches for nvram based systems," in 2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2021, pp. 194–204.