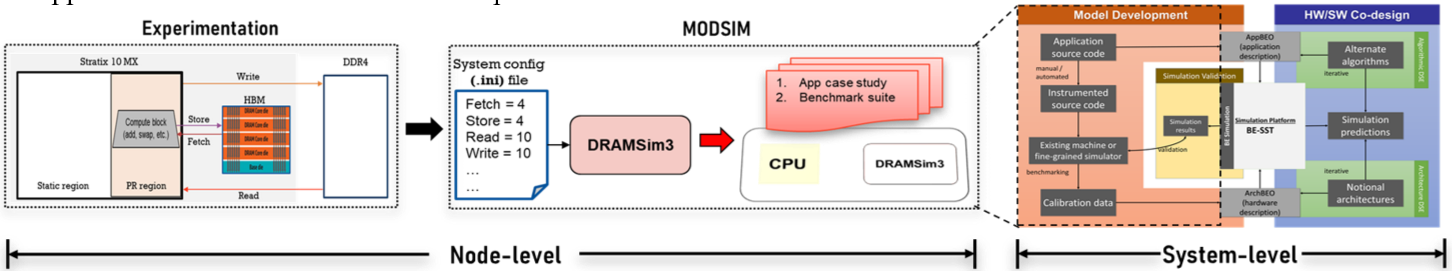
System-level MODSIM of CiM Architectures for Memory-Intensive Applications

Aravind Neelakantan and Herman Lam
Center for Space, High-Performance, and Resilient Computing (SHREC), University of Florida (UF)

In the past decades, memory devices have been playing catch-up to the improving performance of processors. Although memory performance can be improved by the introduction of various configurations of a *memory cache* hierarchy, memory remains the performance bottleneck at a system level for big-data analytics and machine learning applications. An emerging solution for this problem is the use of a complementary *compute cache* architecture, using Compute-in-Memory (CiM) technologies, to bring computation close to memory. CiM implements compute primitives (e.g., arithmetic ops, data-ordering ops) which are simple enough to be embedded in the logic layers of emerging memory devices. Analogous to in-core memory caches, the CiM primitives provide low functionality but high performance by reducing data transfers. In this abstract, we describe a novel methodology to perform design space exploration (DSE) through *system-level* performance modeling and simulation (MODSIM) of CiM architectures for big-data analytics and machine learning applications.

The foundation of this work is based on Behavioral Emulation (BE) [1], a coarse-grained, *system-level* MODSIM methodology; and BE-SST [2,3], a simulation platform built on SST [4]. BE and BE-SST were developed as a part of the DOE PSAAP II Center for Compressible Multiphase Turbulence (CCMT) [5]. The workflow of BE, illustrated on the right side of Fig. 1, consists of two major phases: (1) Model Development (design, validation and calibration) and (2) Hardware/Software Co-design for performance prediction. In the *Model Development phase*, the application of interest is instrumented and run on an existing system to collect training and test data. The training data is used to develop architectural models, using methods including symbolic regression and various interpolation techniques. The test data is used for validation and calibration. Along with the architectural models, the input to the BE-SST simulation platform is an application model, modeling the application of interest. In the *HW/SW Co-design phase* of the workflow, application design-space exploration (DSE) can be performed; e.g., alternate algorithms of an application can be studied on a given architecture. Alternatively, architectural DSE can be performed for a given application. In the work described in this abstract, we extend the Model Development phase to enable the modeling and study of *emerging and notional* (e.g., CiM) architectures, using an approach which is a combination of experimentation and MODSIM.



```
Fetch = 4
Store = 4
Read = 10
Write = 10
…
…
```

Figure 1 Methodology for MODSIM of CiM architecture

Experimentation. Experimentation is performed using a Stratix 10 MX FPGA (with 16GB HBM2 on the same package) to emulate the compute primitives to be found in a notational HBM (anticipated with embedded logic functions). Simple compute primitives such as data-ordering ops, arithmetic and logical ops are implemented in the PR (partial reconfiguration) region of the FPGA. These primitives are used in an application of interest to access data in the HBM. We measure the read/write cycles between the compute primitives and the HBM, which will be used in the MODSIM phase. Note that these "read/write" cycles will become "fetch/store" cycles in a notational CiM memory (as shown in the Fig. 1). Compared to other approaches with standalone emulation of CiM architectures on ASIC/FPGA, the emulation data collected in our platform will be used in the MODSIM phase.

MODSIM. A combination of gem5 [6] and modified DRAMSim3 [7] (with the capability to simulate compute primitives in memory) will be used to simulate CPU+CiM architecture at a node-level. The resulting data from these fine-grained simulators is equivalent to training and test data collected from benchmarking on an existing system, as described above for our existing BE-SST platform. Note that our MODSIM approach is not just a low-level architectural (gate-level and transistor-level) simulation; it uses fine-grained MODSIM to provide data for a coarse-grained, system-level MODSIM.

At the workshop, we will present initial results of performance prediction of memory-intensive applications such as machine learning and big-data analytics on notional CiM systems. These initial results will demonstrate the performance difference (in terms of speedup) of the baseline version of these apps and compare that to the refactored/optimized version for the CiM architecture. Thus, this approach will become a framework to study the performance of other applications on notational architectures; and study how to refactor their algorithms to best use these new architectures. The framework will also support the analysis of different CiM architecture configurations; thus, providing the ability to perform DSE of both applications and architectures. Although we currently target only performance (execution time, clock cycles), this can also be extended to study other parameters (e.g., energy/power, reliability) as a part of future work. The compute cache concept can also be extended to include Compute-near-Memory; and include other memory devices such as storage-class memories.

ACKNOWLEDGEMENT

This research is funded in part by the NSF SHREC Center and the National Science Foundation (NSF) through its IUCRC Program under Grant No. CNS-1738420; and by NSF CISE Research Infrastructure (CRI) Program Grant No. 1405790

REFERENCE

[1] Kumar N., Pascoe C., Hajas C., Lam H., Stitt G., George A. (2016) **Behavioral Emulation for Scalable Design-Space Exploration of Algorithms and Architectures**. In: Taufer M., Mohr B., Kunkel J. (eds) *High Performance Computing. ISC High Performance 2016*. Lecture Notes in Computer Science, vol 9945. Springer, Cham

[2] Aravind Neelakantan, Sai Chenna, Trokon Johnson, Herman Lam, Greg Stitt. **BE-SST: Coarse-Grained Simulation Method & Tools for Full-System Modeling and Simulation**. Presented at *Workshop for Modeling and Simulation (MODSIM 2019)*. (Extended Abstract)

[3] Ajay Ramaswamy, Nalini Kumar, Aravind Neelakantan, Herman Lam, Greg Stitt. **Scalable Behavioral Emulation of Extreme-Scale Systems Using Structural Simulation Toolkit**. Published at *47th International Conference on Parallel Processing 2018*, Eugene, OR.

[4] Arun F Rodrigues, K Scott Hemmert, Brian W Barrett, Chad Kersey, Ron Oldfield, Marlo Weston, Rolf Risen, Jeanine Cook, Paul Rosenfeld, E CooperBalls, et al. 2011. **The structural simulation toolkit**. *SIGMETRICS Performance Evaluation Review 38*, 4 (2011), 37-42.

[5] CCMT. **PSAAP-II Center for Compressible Multiphase Turbulence**. https://www.eng.ufl.edu/ccmt/.

[6] Shang Li, Rommel Sánchez Verdejo, Petar Radojković, and Bruce Jacob. 2019. **Rethinking Cycle Accurate DRAM Simulation**. *In Proceedings of the International Symposium on Memory Systems (MEMSYS '19)*, September 30-October 3, 2019, Washington, DC, USA. ACM, New York, NY, USA

[7] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R Hower, Tushar Krishna, Somayeh Sardashti, et al. 2011. **The gem5 simulator**. *ACM SIGARCH Computer Architecture News 39*, 2 (2011), 1–7.