Compute Cache Architecture for the Acceleration of Mission-Critical Data Analytics

H. Lam, S. Bhat, K. Rajasekaran,V. Srinivasan D. Ojika
Center for Space, High-Performance, and Resilient Computing (SHREC)
University of Florida (UF), http://nsf-shrec.org, hlam@ufl.edu

This study explores how to exploit a ***compute cache*** architecture to bring computation close to memory. Using a combination of experimental prototypes, benchmarking, and modeling & simulation, we perform architectural and application explorations of emerging/notional memory devices and compute cache architectures of the future to accelerate data analytics applications.

Analogous to a memory cache hierarchy in which different levels of memory caches (with different capacities and performance) are strategically placed in and near the CPU compute device (Fig. 1(a)), a ***complementary*** *hierarchy* of compute "caches" can be strategically placed to provide compute-in-memory (CiM) and compute-near-memory (CnM) capabilities (Fig. 1(b)). ***In-memory*** compute cache implements ***compute primitives*** (e.g., arithmetic ops, data-ordering ops) which are simple enough to be embedded in the logic layer of emerging memory (eMEM) devices. Analogous to the in-core memory caches, compute primitives provide low functionality but high performance. ***Near-memory*** compute caches can provide ***CnM capabilities*** to implement ***compute kernels***. Near-memory compute caches can be implemented on an FPGA along-side an eMEM device in the same package. CnM compute kernels make use of CiM primitives to accelerate functions (e.g., FFT, convolution, Bloom filter) or to preprocess data from the memory device (e.g., reorder, reduce) in order to support the acceleration of the host data analytics or machine-learning ***application***. This concept can be extended to near-storage compute caches.

Note that existing memory devices (HMC, HBM2) do not have or have limited CiM capabilities. Thus, for experimentation today, we implement these CiM primitives also on the FPGA of existing (FPGA + HMC/HBM2) platforms and collect benchmarking (performance) data for the modeling/simulation tasks. The benchmarking data collected from our experimental studies, along with research findings in the literatures (on emerging or notional memory devices) are used to develop models and perform simulation studies to perform *design-space exploration* of emerging/notional memory devices and compute cache architectures of the future.

In summary, the compute cache architecture provides a fundamental impact by introducing a novel approach to add a ***new dimension in parallel computing architecture***, increasing parallelism by distributing performance-critical computing near or in memory, and at different levels of granularity. A near-term impact is to demonstrate how the compute cache architecture (FPGA + eMEM) can ***amplify the acceleration capabilities of FPGAs*** (e.g., SPMV, graph analytics, DNN training) by providing low-latency, high-bandwidth memory access for FPGA acceleration.

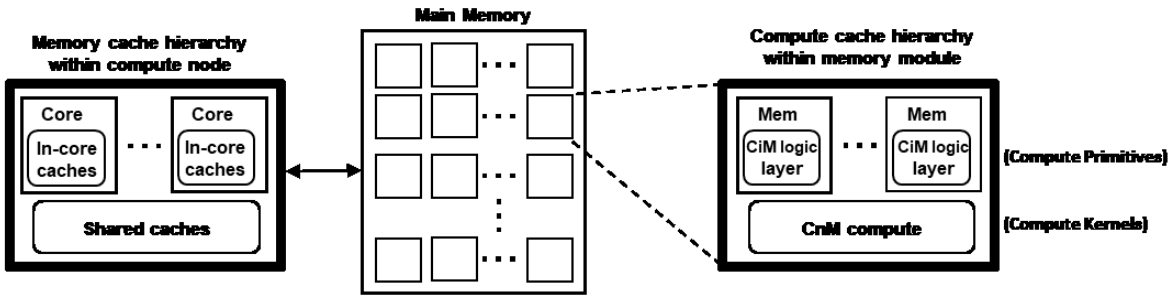

Figure 1(a) Memory cache hierarchy

Figure 1(b) Compute cache hierarchy

* This research is funded 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.