Compute, Memory, and Orchestration: Specialized Architecture in Edge Computing

Neena Imam
Oak Ridge National Laboratory

Ferrol Aderholdt
Middle Tennessee State University

Nageswara S. V. Rao
Oak Ridge National Laboratory

Abstract

Task-specific core assignment is a well-known technique of dedicating cores for specific tasks. This technique is effective in virtualization and High Performance computing (HPC) work flows. A common set of edge computing tasks includes lightweight computation, data and sensor signal streaming, and machine learning computation. For these types of applications, we propose a generic architecture by focusing on three main components: Compute, Memory orchestration, and Networking (CMN) as shown in Figure 1.

Edge computing is an increasingly popular computing paradigm focused on the reduction of application latency and power consumption by placing computing devices close to a consumer and leveraging remote cloud or HPC systems. The devices enabling this paradigm are generally resource constrained such as ARM-based and Raspberry PI platforms. Many of the ARM-based CPUs enabling these devices are equipped with multiple computing cores with systems ranging from 4 cores to 32 cores [2]. With increased number of cores, the efficiency of these devices has also increased, and indeed many of them provide good performance while maintaining lower power consumption.

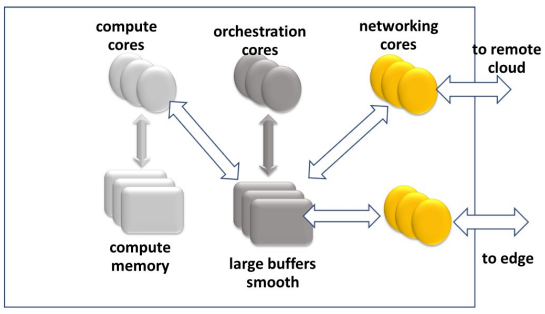

Figure 1: Generic CMN architecture.

In commodity computing systems, the increasing core counts have motivated researchers to specialize cores to specific tasks to obtain higher levels of performance. Core specialization has also been considered at the hardware-level by various vendors including IBM with the cell processors and ARM with the bigLITTLE architecture. Core specialization and related software allow for more efficient computing overall. As such, we explored specialized architectures for edge and fog computing. At the edge, many common work flows, such as scientific and machine learning applications and data analytics, rely on an effective utilization of (i) computation, (ii) memory and process orchestration, and (iii) networking. The overall work flow performance in terms of the execution time and energy consumption depends both on the collection of cores with different capabilities and their allocation to computing and communications tasks. Thus, we levereaged our CMN architecture to study the underlying performance trade-offs. We studied combinations of efficient memory and networking cores that target communications tasks and more performant cores that target computations.

To assess the feasibility and performance of the CMN architecture, we developed a prototype CPU implementing these specializations in the gem5 simulator [1]. We evaluated our prototype using a combination of synthetic benchmarks (i.e., matrix multiplication and 2D 5 pt stencil computation) and the application, You Only Look Once (YOLO) version 3. We found the CMN processor was able to increase performance as much as 63% and decrease power consumption by as much as 7%.

References

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

[2] Cavium/Marvel ThunderX2. Marvell thunderx2 product brief, 2019.