Data-Centric Design

Samira Khan, University of Virginia/Google, samirakhan@virginia.edu

In this current data-centric era, data generated by social media, video sharing applications, swarms of sensors, autonomous cars, etc. is growing exponentially. Unfortunately, as the technology scaling slows down, the semiconductor industry has been facing a major challenge in providing better performance while processing such large datasets. As a result, we need to innovate how we design our systems to sustain the demand for computing over exponentially growing datasets. However, the fundamental model of computing has not changed over many decades. In our current Von Neumann model, *data* sits in the slower persistent storage and it is moved back and forth to faster memory for computation by the processor. My research focuses on enabling a paradigm shift in how data is stored and processed in future systems. The goal of ShiftLab (motivated to introduce a paradigm shift [1]) is to fundamentally rethink our current processor-centric computing model and introduce a data-centric computing model.

Current Trend of Moving Computation to Data

Decades of technology scaling made our processor faster without paying much attention to data storage and communication, resulting in an imbalanced system. Our current processor-centric systems are heavily bottlenecked by data movement in every aspect–from memory to storage to network devices. It takes 1000X more time and energy to bring two floating point numbers from memory than to perform a addition on them. Memory latency and bandwidth is a major bottleneck for data-intensive applications. Google workloads spend 60% of the total energy for data movement [2]. On the storage side, the storage I/O constitutes 50-90% of total query execution time [3]. The internal bandwidth of SSD grew by almost 100X in the last 10 years, but the I/O bandwidth only grew by 5X [4, 3]. Similarly, our network devices now operate at 100-400 Gbps and CPUs cannot keep up with this extensive data rate anymore [5].

To this end, the current trend shifted quite drastically towards moving computation close to data. The recent emergence of 3D-stacked memory has unlocked the opportunity of revisiting processing-in-memory (PIM) architectures, a technology that directly places processing units near memory to reduce the overhead of data movement between host CPU and memory. SmartSSDs place logic close to NAND flash chips using simple cores or specialized hardware accelerators [4, 3]. SmartNICs process data at line-rate in switch/NIC while data is moving through them [5]. The emergence of FPGA-enabled SmartSSDs, SmartNICs and smart memory devices have diverged from the tradition processor-centric design. Instead of moving data to computation, all these computation models move computation to data. In addition to having these different tiers of computations on the server, the edge devices will provide another layer of computational tier on-device. Figure 1 demonstrates these tiers of computations in our future data-centric system.

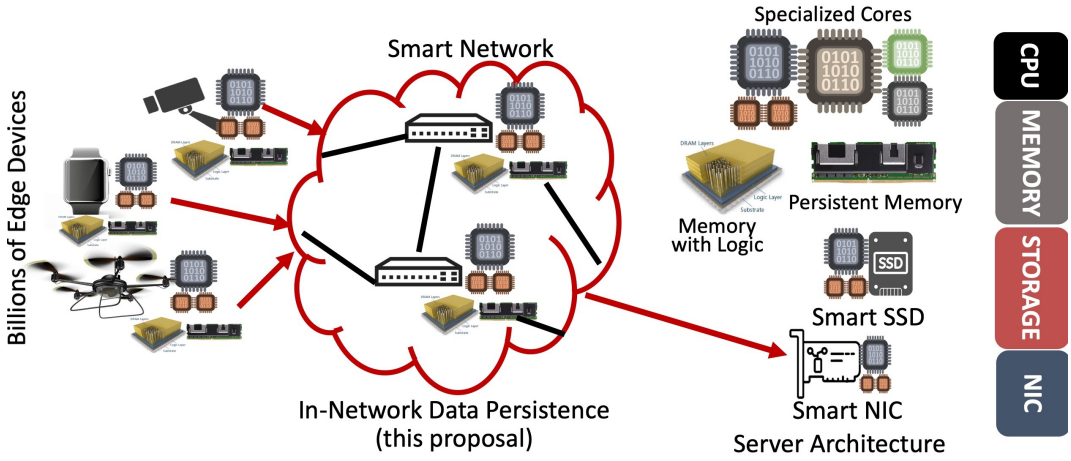

Figure 1: Difference between a single-server vs. data-center scale organization.

To this end, the major challenges of such a heterogeneous system are the following: *(i)* how to place data and computation in the appropriate level, *(ii)* how to orchestrate different stages of computations to minimize the overhead of scheduling and transferring the intermediate results, *(iii)* how to provide faster data-path avoiding going back and forth to the host cores, *(iv)* how to provide transparent support for programmers, *(v)* how to provide consistency between these computational tiers, etc. We envision a framework that can choose the best computational tier based on the data locality and and orchestrate the stages appropriately in a transparent manner.

References

[1] Thomas S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, 1962.
[2] Amirali Boroumand et al. Google workloads for consumer devices: Mitigating data movement bottlenecks. ASPLOS, 2018.
[3] Jaeyoung Do et al. Query processing on smart SSDs: Opportunities and challenges. SIGMOD, 2013.
[4] Sangyeun Cho, Chanik Park, Hyunok Oh, Sungchan Kim, Youngmin Yi, and Gregory R. Ganger. Active disk meets flash: A case for intelligent SSDs. ICS, 2013.
[5] Dan R. K. Ports and Jacob Nelson. When should the network be the computer? HotOS, 2019.