Research Statement

Mert Hidayetoglu (him/his)

My research interest is high-performance communication in high-end GPU systems in the realm of clusters, data centers, and supercomputers. Specifically, I envision automatic optimization of nonuniform data movement across hierarchical interconnect architectures for large-scale applications that can impact the future course of science and industry.

My dissertation work was on optimizing communication in large-scale applications with sparse data. I proposed hierarchical communications to take advantage of multi-GPU node architectures. I optimized data movement to be tailored to a specific sparsity pattern *and* the underlying system. As supercomputers and data center server systems increasingly adopted multi-GPU architectures, due to their superior compute throughput and power efficiency, I proposed techniques for several high-impact applications to make effective use of these complex designs. My postdoc research is aimed at identifying abstractions that allow developers to reason about the systems, adapt to different systems, and get performance out of them without making every program a one-off exercise. In the rest of this statement, I will talk about how I got here, my postdoc research, and future directions.

How did I get here?

My background is in electrical engineering, where I focused on solving large-scale computational electromagnetics problems. These problems have a large memory footprint, but we had a small computer cluster (16 nodes). I employed an SSD in each node and developed parallel out-of-core techniques to solve integral equations with billions of unknowns (in 2012) for computing radar-cross-sections of aircraft. During my PhD, my specific focus was to solve large-scale inverse problems in imaging. I solved 3D image reconstruction problems with fast algorithms running on the top supercomputers in the world (e.g., Blue Waters, Summit).

The common theme of the applications that I worked on is two-fold. First, they involve large enough data transfers to saturate the data-movement rate (bandwidth) of the system. Second, they involve the repetition of an iterative pattern. My research takes advantage of the hierarchical interconnect architecture by analyzing and memoizing the optimal sparse data movement patterns over high-bandwidth (on-chip) memory so that we can reduce the amount of data moved through "slow" memory, and reuses the optimal pattern in each iteration.

I was fortunate enough to spend summer 2018 at Argonne National Laboratory. There I applied hierarchical communications across GPUs specifically for solving a large-scale X-ray imaging problem [SC19, SC20]. The measurement data from a 3D mouse brain were collected using a particle accelerator (the Advanced Photon Source). This problem required multiplying a large (11 TB) sparse matrix and its transpose in each iteration, and it takes a supercomputer (with at least 768 GPUs) to even hold the data for the problem in memory. We used 4,096 nodes of the Summit supercomputer (24,576 GPUs) and applied hierarchical communications to solve the problem in under four minutes; with standard techniques the problem would have taken hours or even days to solve. The peak performance we achieved of 64 PFLOPS (32% of the theoretical limit) is unprecedented in sparse applications and was primarily due to efficient hierarchical communication.

Postdoc Research

In my PhD work I had great difficulty implementing and optimizing communications tailored to specific applications and systems, and I thought that "this should be easier." My postdoc research is about making developing hierarchical, optimized data movement across GPUs easier with a 1) composable, 2) portable, and 3) performant communication library.

Data movement on supercomputers can be very complex, and the lower-level optimizations vary drastically across systems. The differences between vendors are so significant that using communication software not carefully matched to the hardware can severely affect performance, resulting in wasted time and energy. However, optimizing communication algorithms for specific systems is very tedious with currently available programming models. Thus, my research explores intuitive programming models for communication and automatic optimization techniques that work across GPU systems of diverse architectures and scale.

I (with my collaborators) have already reached a few important milestones.

CommBench: I developed a configurable benchmarking tool to gain performance insights on current systems [CommBench]. We extensively benchmark different communication interconnects and implementation libraries across six different supercomputers (Summit, Delta, Perlmutter, DGX-A100, Frontier, and Aurora), and find optimization opportunities that existing mainstream communication library implementations (MPI, NCCL) do not exploit. For example, we take advantage of multiple network interface cards (NICs) on each node by "striping" the communication data to maximize the communication bandwidth across nodes. CommBench has already been used for acceptance tests for advanced (exascale) system procurement, specifically for testing the implementation libraries and their tuning for communications at each level of the network hierarchy of a specific system.

HiCCL: I used CommBench as a tool to develop a hierarchical communication library (HiCCL) for optimizing any user-defined communication pattern across systems with various architectures and different vendors, currently including Nvidia, AMD, and Intel. Two fundamental data-dependency schemes (one-to-many and many-to-one) across GPUs are used as primitives to express all standard collective functions in a few lines of code. HiCCL then synthesizes optimized communication code automatically for an abstract machine represented with a few (five) parameters. HiCCL uses a level of abstraction that is general enough to work across contemporary node architectures and sufficiently descriptive to take advantage of a specific architecture. I tested HiCCL across the eight most common collective communication patterns on five supercomputers (40 cases) and found that HiCCL utilized approximately 95% of the peak communication throughput. In contrast, the corresponding MPI implementations utilize 10% of all systems, and NCCL functions utilize 80%, where available (12 cases).

Future Directions

I would like to form a group focused on high performance data movement in large-scale applications on both current and future architectures. I will work with students and collaborators with varied backgrounds and expertise to explore a few potential research directions:

1. Foundational machine learning (ML) models are large, static, and repetitive. My research is a perfect candidate to reduce the cost of such demanding applications, such as the gradient exchange communication while training large language models. Furthermore, the training of these models is data-hungry, and once the main operations are optimized, storage and I/O may become the bottleneck. I will extend the hierarchical communication pipeline to distributed disk I/O on supercomputers for streamlining the data flow into the training of foundational models.

2. I am interested in sustainable computing from a systems point of view. Optimizing data movement already saves a lot of energy since it is costlier than computations. Nevertheless, it still takes a supercomputer to solve large problems because of the memory requirements. An under-explored alternative is to use storage systems that have slower bandwidth yet allow solving large problems with smaller compute resources. The challenge is efficient (sparse) memory access to the disk, which I investigated in different contexts. My future research will investigate storage systems for fitting larger problems in smaller systems with a minimal penalty.

3. Dynamic workloads with small messages are the opposite extreme of my focus so far on large and repetitive workloads. This direction is especially challenging because they are typically latency-critical, and we cannot make optimization decisions in advance. Moreover, the sparse data dependencies must be analyzed on-the-fly to enable packing and routing of data in an optimized way to take full advantage of the communication interconnect of the system.

4. In my research, the machine abstractions are parameterized for fitting on systems with different shapes and sizes. Nevertheless, choosing good parameters for a specific system and workload still requires expertise. I will investigate the auto-tuning mechanism, where the proposed tuner queries the system and performs measurements for choosing the optimal parameters for a specific system automatically. We would like to converge to the desired parameters within a few trials, and therefore I will investigate nonlinear optimization schemes and control theory, informed by intuition, for fast convergence.

5. We have already demonstrated sparse hierarchical communication on X-ray image reconstruction at an unprecedented scale. The new generation of light sources, such as LCLS-II at SLAC or APS (Gen-5) at Argonne will be brighter, which means (i) larger amounts of data will be produced and (ii) new imaging techniques will be unveiled. We will use the proposed frameworks to enable novel X-ray imaging techniques on new exascale systems. I will use my collaborations to demonstrate the work in practice.

Long Term

Machines are increasing in complexity, and communication in future data centers will increasingly dominate time and energy costs. Thus there will be challenging and important research problems to address in making communications performant, portable, and programmable for many years to come. Enabling application developers to reduce the costs of data movement without introducing excessive complexity into the applications will be critical for continued progress in computing.

[SC19] MemXCT: Memory-centric X-ray image reconstruction with massive parallelization.

[HPEC20] At-scale sparse deep neural network inference with efficient GPU implementation.

[SC20] Petascale XCT: 3D Image reconstruction with hierarchical communications on multi-GPU nodes. **(best paper)**

[CommBench] https://github.com/merthidayetoglu/CommBench.