High-Performance Computing Group

Prof. Dr. Christian Plessl

Computer Science Department and
Paderborn Center for Parallel Computing

Efficient FPGA-to-FPGA Collective Communication in ACCL

Bachelor/Master Thesis

At a glance

- Research on efficient collective communication algorithms
- C implementation of algorithms in ACCL firmware
- Validation of the implementation using the existing ACCL emulator and simulator
- Evaluation on real hardware with up to 48 FPGAs for different communication scenarios

MPI is the de-facto standard for inter-process communication on HPC systems. Next to the simple point-to-point communications, the MPI standard also describes calls for common communication patterns like broadcasts, scatter ,gather, and reductions. These so called collectives allow MPI implementations to provide optimized algorithms for each communication pattern depending on the network topology in the HPC system, message sizes, number of processes, and other parameters.

In the last years, Field Programmable Gate Arrays (FPGAs) gained more and more attraction as accelerator for HPC workloads. Like GPUs, they often come as accelerator cards and allow the design of custom hardware to increase the performance and/or power efficiency of HPC applications.
Like for CPUs, the FPGA workload needs to be distributed over multiple devices to better utilize the available resources on an HPC system. This also requires communication among the FPGA cards. In some systems like Noctua 2, the FPGA cards are directly connected to a network switch to support direct, low latency and high bandwidth communication between FPGAs.
However, to better make use of this novel infrastructure, a framework is required to provide MPI-like primitives and collectives on the FPGAs. One of these frameworks is the Accelerated Collective Communication Library (ACCL). All its communication routines are implemented as control firmware on the FPGA. Many of the implemented collectives use naïve algorithms which can quickly become a performance bottleneck when scaling to higher number FPGAs. Still, there may not be “the best” implementation of a collective for all scenarios.

Your task would be to research on optimized collective communication algorithms and provide C implementations of promising algorithms for the ACCL firmware. The implementations can be validated with an existing emulator and simulator for fast iterations. The final evaluation and benchmarks will be executed directly on the FPGAs on Noctua2.

Further reading:
https://accl.readthedocs.io/en/latest/
https://github.com/Xilinx/ACCL
https://ieeexplore.ieee.org/abstract/document/9651265

Contact:
Marius Meyer, Phone: 05251/60-1718 E-Mail: marius.meyer@uni-paderborn.de

For information on theses offered by the HPC group or PC2 please visit:
https://en.cs.uni-paderborn.de/hpc/teaching/open-theses