CSE523
Integration of Modern Programming Languages with Partitioned Global Address Space Parallelism

Contact

Please send resume/CV to Dr. Barbara Chapman (barbara dot chapman @ stonybrook.edu, and Tony Curtis (anthony dot curtis @ stonybrook.edu).

Prerequisites

- Understanding of parallel programming, preferably distributed like MPI
- Enrolled MS/PhD student
- Proficiency in C/C++
- Some knowledge of Rust/Go!/D (optional)

Motivation

Much High Performance Computing (HPC) programming uses C/C++ for applications like scientific simulations (Fortran is also widely used). C, and to a lesser extent, C++, are well-known for the ability to shoot oneself in the foot with issues such as memory access errors. US government agencies, industry, and researchers at national labs (e.g. Los Alamos, one of our sponsors) are interested in moving to newer, safer languages for a number of reasons, including: national security, verification, and scientific integrity, but they also want to take advantage of other distributed parallelism features that are not currently addressed in these languages, and to interface with legacy code.

Overview
The Partitioned Global Address Space (PGAS) programming family is a set of HPC languages and libraries that employs a relaxed parallelism model utilizing Remote Memory Access (RMA or RDMA). Traditionally most libraries are targeted at C/C++ (and also Fortran) and are then used to implement PGAS programming languages or models. Notable libraries include ARMCI and GASNet. Programming languages include UPC, UPC++ and the co-array feature of Fortran.

We have been working with a PGAS library called OpenSHMEM for a number of years. OpenSHMEM provides a communication and memory management API for C/C++. OpenSHMEM focuses on sending data between processes on different compute nodes directly into remote memory without interrupting CPUs, instead utilizing capabilities of high performance interconnects (e.g. Infiniband). Our implementation is based on a new communications substrate library called UCX, which is also used in the widely deployed Open-MPI & MPICH libraries and in various distributed machine learning frameworks. We interact frequently with other developers of both Open-MPI and UCX, and our group is well-known in these, and other, HPC communities.

There is interest in providing OpenSHMEM interaction with newer “safe” programming languages.

**Project**

This project will investigate how best to integrate OpenSHMEM and Rust, and possibly D as a 2nd priority. Some work has already been done with Rust, Go!, and others, so the goal is to understand what has already been done, and how to provide an extensible framework for other languages. The outcomes are

- A written report on the work done
- Presentation to our research group
- Demo implementations of yoking OpenSHMEM and Rust/D/…
- And ideas for future work on interaction frameworks.

**For More Information**

- Our group
  - [https://you.stonybrook.edu/exascallab/](https://you.stonybrook.edu/exascallab/)
- [http://www.openshmem.org/](http://www.openshmem.org/)
- [https://www.openucx.org/](https://www.openucx.org/)
- [https://www.rust-lang.org/](https://www.rust-lang.org/)
- [https://golang.org/](https://golang.org/)
- Fortran CoArrays
  - https://dl.acm.org/doi/pdf/10.1145/2754942.2754944?casa_token=Id-YpX2obJoAAAAA:XTCGBth8BcsYVUQi8vh32ljbbFdmkGQQrgi4J1CC1Apql_Ot8OZQR3U17FdHzK4BinrcaPalPJr