SAND2021-8810D

Integrating PGAS and MPI-Based Graph Analysis

CSRI Summer Proceedings 2021

Presented by: Trevor McCrary

SAND2021-XXXX

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

This work describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

boratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LL
Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE
SAND2021-XXXX

Integrating PGAS and MPI-Based Graph Analysis

Intern: Trevor McCrary, Mississippi State University, **Virtual at:** Starkville, Mississippi
Mentor: Karen Devine, 1465 Department of Scalable Algorithms

Abstract This project demonstrates that Chapel programs can interface with MPI-based libraries written in C++ **without storing multiple copies of shared data**. Chapel is a language for productive parallel computing using global namespaces (PGAS). We identified two approaches to interface Chapel code with the MPI-based Grafiki and Trilinos libraries. The first uses a single Chapel executable to call a C function that interacts with the C++ libraries. The second uses the mmap function to allow separate executables to read and write to the same block of memory on a node.

Problem Domain

Graph Analysis

Technical Approach

PGAS + MPI

Mission Application

Strategic Partnerships

SAND2021-XXXX

Integrating PGAS and MPI-Based Graph Analysis preliminary results

Intern: Trevor McCrary, Mississippi State University, **Virtual at:** Starkville, Mississippi
Mentor: Karen Devine, 1465 Department of Scalable Algorithms

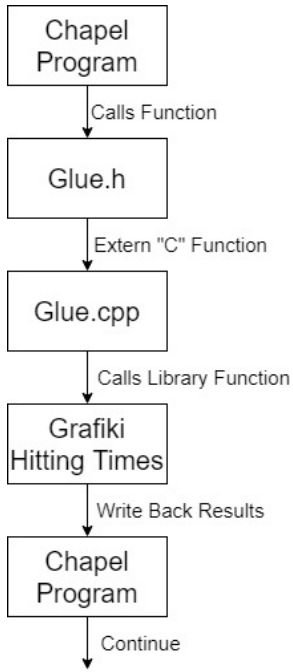

Figure: Method 1: a single Chapel executable calls a C function that interacts with the C++ libraries

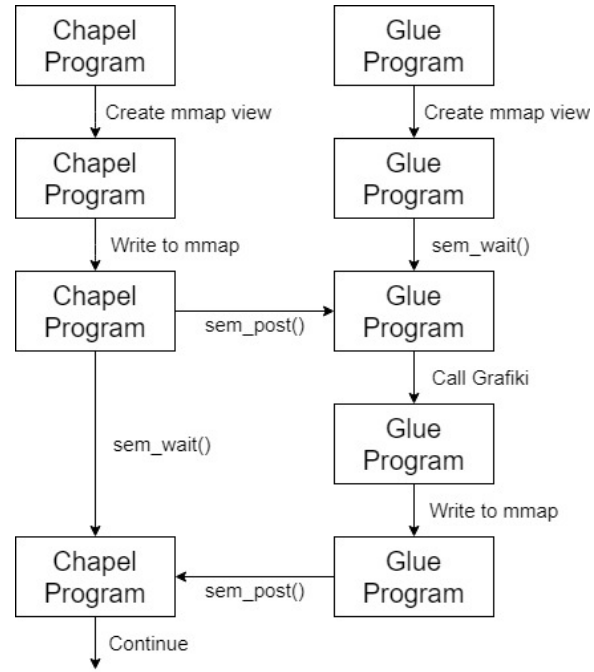

Figure: Method 2: two executables use the mmap function to read and write to the same block of memory.

SAND2021-XXXX