Modeling and Benchmarking Irregular MPI Communication Performance

Carson Woods

September 29, 2022

Background

- Many scientific applications rely on irregular MPI communications for exchanging data between processes.
- That irregularity stems from the data to be exchanged constantly changing throughout the runtime of the application (unknown at compile time).
- The exact behavior (and performance) varies across applications and MPI implementations, so this makes it challenging to fully understand and analyze the performance of these applications.

Modeling Applications

- We aim to model communication performance across a variety of applications using a single benchmark.
- Throughout their runtime, applications modify parameters which determine communication behavior and performance.
- By extracting those parameters and using them to tune our benchmark, we can model (and analyze) the communication performance of these applications in a consistent manner.

The Benchmark

- Handles communication data exchange via the L7 communication library in the CLAMR mini-app [1].
  - Allows our benchmark to support MPI + (OpenCL, OpenMP, CUDA)
- Tunes the communication behavior via the following parameters:
  - **n-owned** – the amount of data belonging to each process
  - **n-remote** – the amount of data being sent to communication partners
  - **block size** – the size of the messages being sent
  - **stride** – the number of bytes between blocks
  - number of communication partners per process

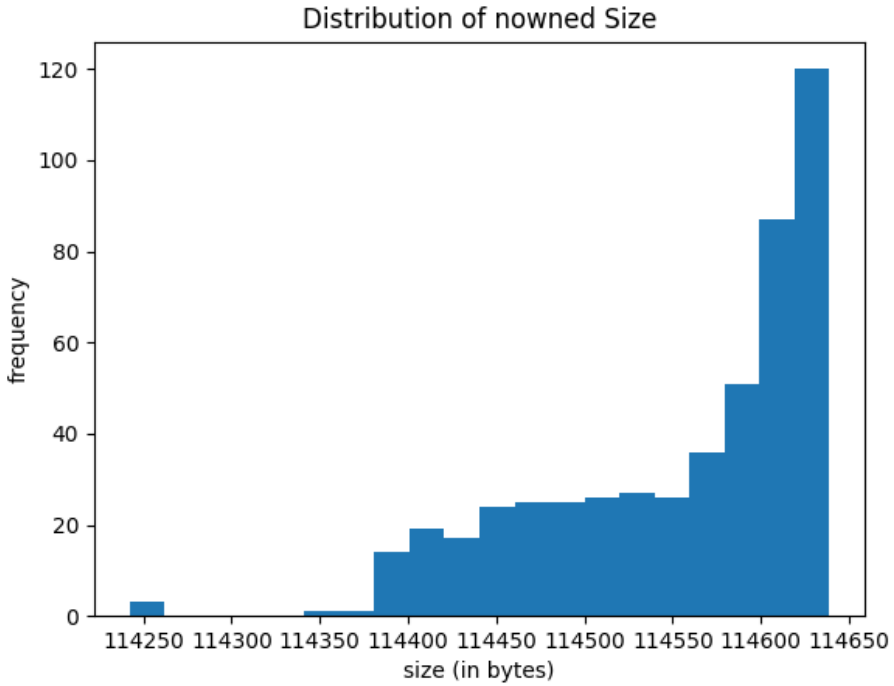
[1] D. Nicholaeff, N. Davis, D. Trujillo, & R. W. Robey (2012). Cell-Based Adaptive Mesh Refinement Implemented with General Purpose Graphics Processing Units.

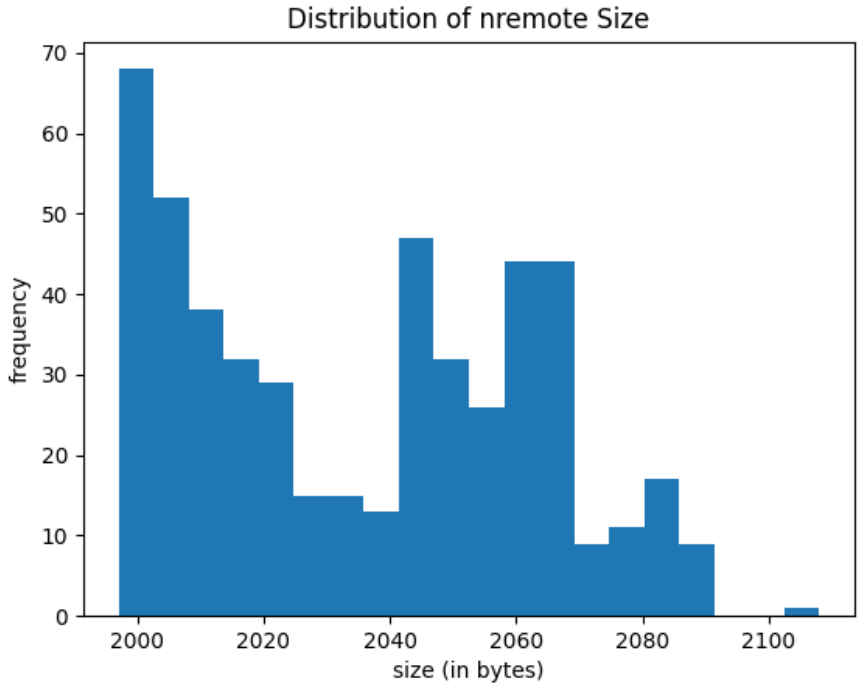
Current Progress

- We have built a robust, but lightweight benchmark which can provide performance metrics based on parameters from 3rd party applications.
- We have begun extracting these parameters from CLAMR and put them into our benchmark, and we intend to collect these parameters from other mini-apps soon.
- We are also taking the time to understand the distribution of these parameters across the various collected applications.
  - It is useful to better understand the distribution types, means, and standard deviations for each parameter across each application to better tune the benchmark.

Parameters Distribution

n-owned – amount of data ”owned” by a process
n-remote – amount of data sent to neighbors





Distribution of nremote and nowned parameters from CLAMR utilizing using 288 processes across 8 nodes. The CLAMR job used a grid size of 5734 and ran for 500 timesteps.

Next Steps

- Intend to sample parameters across a range of applications and perform the same statistical analysis we're already performing on parameters from CLAMR.
- Use the benchmark to profile communication performance (data packing, unpacking, bandwidth, latency, etc.) across MPI implementations and higher-level communication libraries.
- Use the collected data and performance information to identify where irregular communication performance can be improved.

Conclusions

- Understanding irregular MPI communication performance across a range of applications is a complex task.
- By modeling this behavior in a consistent and relatively lightweight benchmark, we can extract comparable communication behavior data across a range of applications and MPI implementations.

Questions?