Performance Characterization and Tuning of Non-uniform All-to-all Data Exchanges

Kunting Qi, Fan Ke, Sidharth Kumar

University of Illinois Chicago, Department of Computer Science

Introduction

Non-uniform MPI_Alltoallv communication is critical in many high-performance computing (HPC) applications where data exchange patterns vary significantly between processes. However, existing MPI implementations rely on fixed heuristics that are not well-suited for dynamic, irregular workloads, often leading to suboptimal performance. Furthermore, there are customized non-uniform All-to-all functions having performance advantage compared to official MPI Alltoallv, under certain scenarios. Creating a framework which can accurately select the optimal algorithm will be extremely beneficial, but challenging due to its potential complexity.

Currently, there are research works regarding the data-driven approach for tuning MPI functions, but most of them they focus on MPI functions such as MPI_Scatter, MPI_Reduce, MPI_gather, uniform MPI_Alltoall. Non-uniform Alltoall is still an under-developed area. One challenge of developing a tuning framework for non-uniform Alltoall is that the blocksizes between processes are not fixed and it introduces significant additional complexity to the tuning workflow.

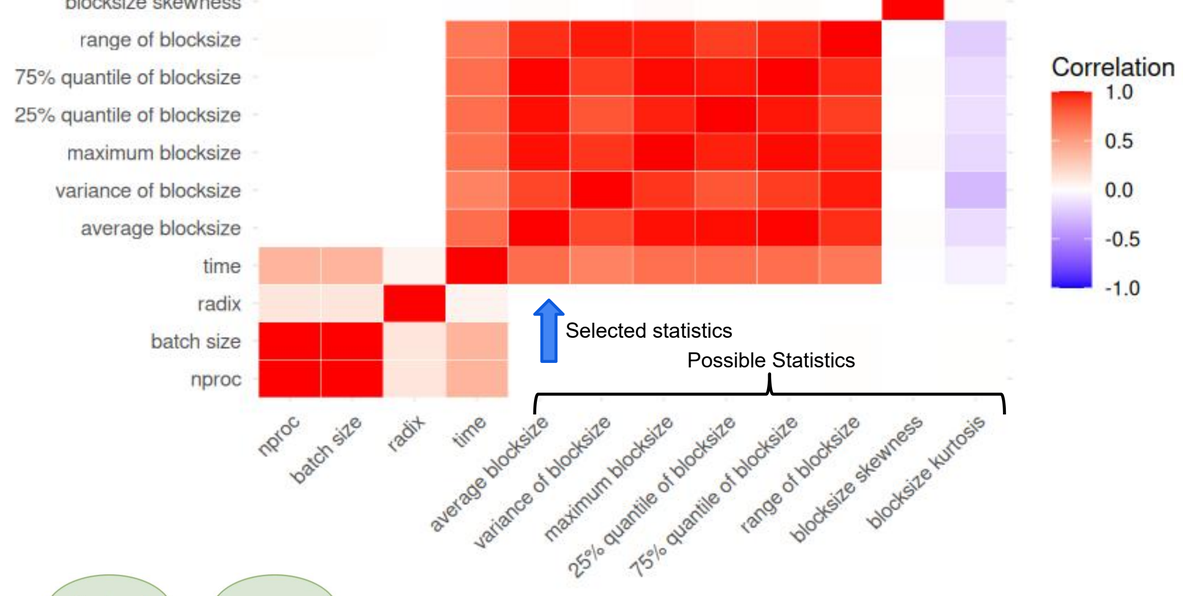
Goal

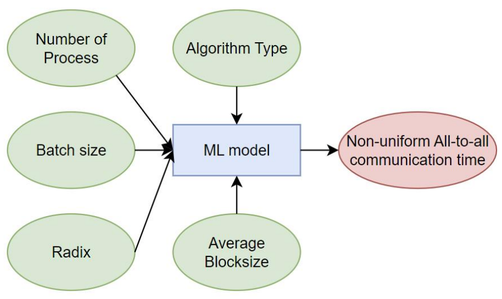
Creating an effective auto-tuning framework for non-uniform All-to-all

- Data-driven approach
- Handling the variable blocksize
- Considering additional customized MPI Alltoallv implementations
  - Two Phase Bruck
  - Tunable Non-uniform All-to-all algorithm (TuNA)

Sensitivity Analysis

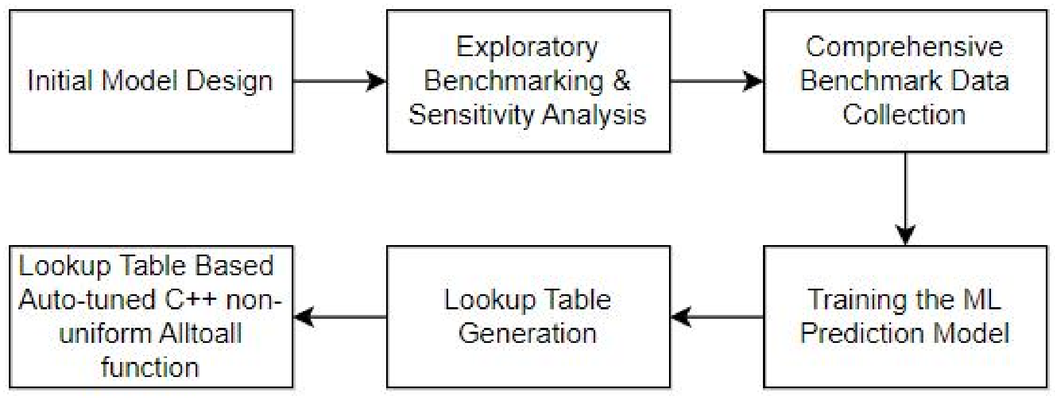
Block sizes are variable and assumed to be statistical random variables, and there can be many associated statistics. However, not all statistics are necessary, so we need to select the important statistics.



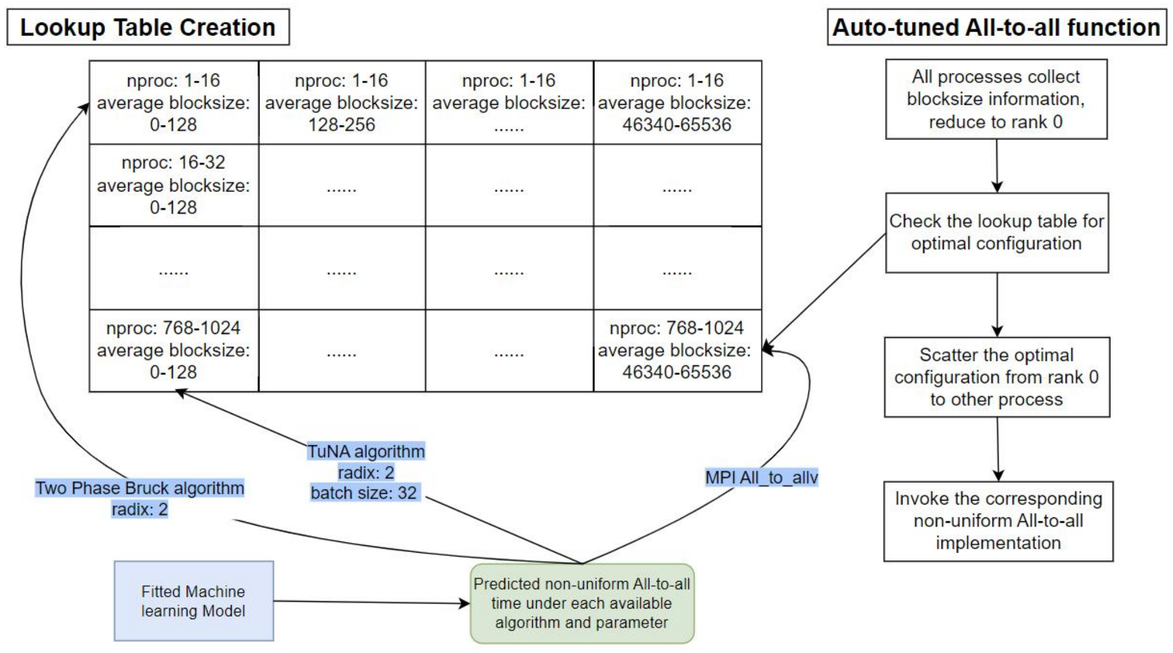


After sensitivity analysis, the performance prediction model is simplified without significantly decreasing accuracy.

Methodology



- Using machine learning model directly in C++ introduces significant overhead, so we created a lookup table from fitted ML model, and use the lookup table in the implemented auto-tuned C++ function.
- Based on the lookup table generated, we implement auto-tuned non-uniform All-to-all function which can dynamically select the optimal algorithm and parameter for non-uniform All-to-all communication.



Benchmarking

Nproc: 32, 64, 96, 128, ..., 1024, 2048 (10 levels in total)
Blocksizes are randomly generated from 1 to maximum blocksize
Maximum blocksize: 2, 4, 6, 8, ..., 3072, 4096, ..., 32768, 49152, 65536 (31 levels in total)

Benchmarking is conducted on Polaris Cluster at Argonne National Lab

Algorithms to be benchmarked

- **MPI All-to-Allv**
- **Two Phase Bruck**
  - Radix = {2, 4, 8, 16, 32, ..., p} and fine-grid levels around radix = 2, $\sqrt{p}$, and p
- **TuNA**
  - Batch size: 1, 2, 3, ..., number of nodes
  - Radix = {2, 4, 6, 8, ..., 32} and fine-grid levels around radix = 2, $\sqrt{32}$, and 32

Evaluation

Benchmark data are divided as follows:

- 70% for training set
- 10% for validation set and hyper-parameter tuning
- 20% for testing set

Fitted ML models are evaluated under the testing set.

Model	Absolute Percentage of Error (%)
Linear Regression	19.49
Regression Tree	1.05
Random Forest	1.14
Lasso Regression	21.93
Ridge Regression	20.35

Our final auto-tuned non-uniform All-to-all function is evaluated on an MPI parallel transitive closure computing application:

- Under nproc = 256
- On Polaris Cluster at Argonne National Lab

Implementation	Communication Time (Sec)	Total Time (Sec)
Official MPI Alltoallv	13.1367	96.1985
Ours	1.9292	85.4119

UNIVERSITY OF ILLINOIS CHICAGO

Argonne NATIONAL LABORATORY