Highly Efficient Alltoall and Alltoallv Communication Algorithms for GPU Systems

Chen-Chun Chen, Kawthar Shafie Khorassani, Quentin G. Anthony, Aamir Shafi, Hari Subramoni and Dhabaleswar K. Panda

Presentation at the 10th Annual MVAPICH User Group (MUG) Meeting (MUG ‘22)

by

Chen-Chun Chen
The Ohio State University

E-mail: chen.10252@osu.edu
Presentation Outline

• Introduction & Motivation

• Design Approaches
  – Optimized designs for Alltoall
  – Optimized designs for Alltoallv

• Performance Evaluation
  – Benchmark-level evaluation
  – Application-level evaluation

• Conclusion & Future Work
Introduction & Motivation

• Alltoall(v) are two of the most communication-intensive MPI operations in HPC and Deep Learning applications that become the bottleneck of efficiently scaling these applications to larger dense GPU systems.

• Existing Alltoall(v) design does not consider dense GPU systems, most of the existing implementations simply use send-recv based algorithm for Alltoall(v) communication.

• It requires new designs for the modern dense GPU systems
Presentation Outline

• Introduction & Motivation

• Design Approaches
  – Optimized designs for Alltoall
  – Optimized designs for Alltoallv

• Performance Evaluation
  – Benchmark-level evaluation
  – Application-level evaluation

• Conclusion & Future Work
Design Approaches

• GPU-aware IPC-advanced algorithm and hybrid designs
  – IPC enables the efficient transfer of messages between GPUs within the same node
  – The existing Alltoall designs usually use simple send-recv pairs to transfer data, no matter in inter or intra-node communication.
  – The proposed IPC-advanced designs provided overlap potential of intra-node and inter-node communication through utilizing zero-copy load-store IPC mechanisms.
  – The proposed hybrid designs took advantage of different techniques and implementations according to message sizes.

• Extension to Alltoallv
Since CUDA 4.1, the Inter-Process Communication (IPC) interface has enabled the efficient transfer of messages between GPUs within the same node.

The existing Alltoall designs usually use simple send-recv pairs to transfer data, no matter in inter or intra-node communication.

Evaluation showed that IPC-enabled design benefits the intra-node latency on DGX-A100 system.
Hybrid Designs

• We found that there is an overhead for launching IPC.
• Different implementations, kernel-based IPC and memcpy-based IPC, introduce different overheads.
• The existing optimized Alltoall (using CPU staging technique) algorithm is good at small messages.
• Kernel-based IPC implementation is good at medium messages.
• Memcpy-based IPC implementation is good at large messages.
• We proposed hybrid designs to take advantage of the lowest overhead over all message sizes.
Extension to Alltoallv

- Alltoallv requires the offsets (sdispls and rdispls) of sendbuffer/recvbuffer where the process should send/place data.
- The information is only related to the current rank, but IPC data transferring needs remote information.
- Exchange the destination offsets (sdispls and rdispls) in advance before performing IPC-advanced data transferring.
Presentation Outline

• Introduction & Motivation

• Design Approaches
  • Optimized designs for Alltoall
  • Optimized designs for Alltoallv

• Performance Evaluation
  – Benchmark-level evaluation
  – Application-level evaluation

• Conclusion & Future Work
Performance Evaluation

Platform
• ThetaGPU @ALCF
• Lassen @LLNL

Baselines
• MVAPICH2-GDR 2.3.6
• OpenMPI 4.1.1 + UCX 1.11.1
• NCCL 2.11.4
• Spectrum-MPI 10.3.1

Benchmark-level evaluations:
• *osu_alltoall* in OSU Micro-Benchmarks (OMB) suite 5.8
• *alltoall* in NVIDIA NCCL Tests 2.11.0

Application-level evaluations:
• **DeepSpeed**, a popular distributed DL framework built on top of the PyTorch DL framework
• **heFFTe**, a highly efficient Fast Fourier Transform (FFT) library which supports GPU kernels
• **PSDNS**, a kernel-based Fourier pseudo-spectral numerical simulation application
• Compare with state-of-the-art MPI libraries
• The proposed designs provide speedups for the Alltoall latency of 16B by up to 13x, and of 1MB by up to 1.2x on 16 ThetaGPU nodes using 128 GPUs

The proposed designs provide **60x** throughput against OpenMPI on 8 ThetaGPU nodes (64 GPUs)

The proposed designs support common datatypes that NCCL does not support.

The proposed designs provide 16x throughput on 16 ThetaGPU nodes using Alltoall communication.

The proposed designs provide 28x throughput on 16 ThetaGPU nodes using Alltoallv communication.
The proposed designs provide **3.5x** speedup on 64 Lassen nodes (256 GPUs)
Presentation Outline

• Introduction & Motivation
• Design Approaches
  • Optimized designs for Alltoall
  • Optimized designs for Alltoallv
• Performance Evaluation
  – Benchmark-level evaluation
  – Application-level evaluation
• Conclusion & Future Work
Conclusion & Future Work

• We proposed a new design, GPU-aware IPC-advanced hybrid design, and improved the performance of GPU-based Alltoall and Alltoallv MPI collective calls on dense GPU systems.

• Considering the different properties of implementations, we have developed a hybrid strategy to use the best communication mechanism to reduce the overhead.

• The evaluations have shown that the proposed designs outperform the baseline by 13x and 1.2x for small and large messages on 16 ThetaGPU nodes.

• The proposed designs is available in MVAPICH2-GDR 2.3.7 release.

• In the future, we want to extend our work to Gather and Scatter communication.
Thank You!

chen.10252@osu.edu

Network-Based Computing Laboratory

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

Follow us on
https://twitter.com/mvapich

MVAPICH
MPI, PGAS and Hybrid MPI-PGAS Library

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

HiBD
High-Performance Big Data

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

HiDL
High-Performance Deep Learning

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/