DESIGNING HIGH-PERFORMANCE ALLTOALL SOLUTIONS
ON DENSE GPU SYSTEMS

Hari Subramoni, Dhabaleswar Panda, Qinghua Zhou, Chen-Chun Chen, Kawthar Shafie Khorassani and Aamir Shafi

Network Based Computing Laboratory
The Ohio State University
Email: panda@cse.ohio-state.edu
PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Host-Staging based Online Compression
  - GPU-aware IPC-advanced algorithm and hybrid designs
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan
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) algorithms for transferring GPU data still suffer from poor performance due to the limitation of commodity networks and data transfer patterns. How can we optimize the AlltoAll(v) algorithms by leveraging the emerging GPU communication technologies or revamping the data transfer pattern to accelerate these applications?

We propose two design approaches along with these directions.

- Host-Staging based Online Compression
- GPU-aware IPC-advanced algorithm and hybrid designs
PRESENTATION OVERVIEW

- Introduction & Motivation
- **Design Approaches**
  - Host-Staging based Online Compression
  - GPU-aware IPC-advanced algorithm and hybrid designs
- **Performance Evaluation**
  - Benchmark-level evaluation
  - Application-level evaluation
- **Conclusion and Future Plan**
HOST-STAGING BASED ONLINE COMPRESSION

- **Compression** can reduce the data size and lower the pressure on network with limited bandwidth.
- Existing Point-to-Point based compression has limitation of overlapping compression/decompression kernels across send/receive operations.
- Propose a collective-level online compression for Host-Staging based MPI_Alltoall.
- Optimize the ZFP compression library to enable execution of compression/decompression kernels on multiple CUDA streams.

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-receive pairs to transfer data, no matter in inter or intra-node communication.
- Our IPC-advanced designs provide overlap potential of intra-node and inter-node communication through utilizing zero-copy load store IPC mechanisms.
- Our hybrid designs take advantage of different techniques and implementations according to message sizes.
Data Flow of Host-Staging based Online Compression

1. GPU data is compressed to the temporary device buffer and copied to the host buffer asynchronously.
2. MPI_Isend sends out the data in the host buffer to other CPUs.
3. MPI_Irecv receives the data to the host buffer from other CPUs.
4. The received data is copied to the temporary device buffer asynchronously and decompressed to the target buffer.
OPTIMIZATION FOR HOST-STAGING BASED ONLINE COMPRESSION

- **Enabling Multiple CUDA Streams in ZFP Library**
  - Design new APIs `zfp_compress(decompress)_multi_stream`
  - Propose new execution policy `zfp_exec_cuda_multi_stream`

- **Co-design the GPU-based compression at the collective level**
  - 1. Launch compression/decompression kernels on multiple CUDA streams
  - 2. Execute the data copy (D→H, H→D) on the same stream as the corresponding compression/decompression kernels
  - 3. Achieve overlap between the compression/decompression kernels across multiple send/receive operations
GPU-AWARE IPC-ADVANCED ALGORITHM AND HYBRID DESIGNS

- **Key idea: overlap the intra and inter-node communication**
  - With the utilization of IPC in the intra-node environment, the IPC buffer pointers can be exchanged and utilized by other GPUs to create this intra-node transfer.
  - The integrated designs support single/multi Alltoall(v) communication patterns and cover common MPI datatypes.

Existing All-to-all Designs:

<table>
<thead>
<tr>
<th>GPU 0</th>
<th>Intra-node data transfer</th>
<th>GPU 0</th>
<th>GPU 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU 0</td>
<td>Intra-node data transfer</td>
<td>GPU 0</td>
<td>GPU 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intra-node data transfer</th>
<th>Inter-node data transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU 0</td>
<td>NODE 0 (GPU 0) NODE 1 (GPU 0)</td>
</tr>
<tr>
<td>GPU 0</td>
<td>NODE 0 (GPU 0) NODE 1 (GPU 1)</td>
</tr>
</tbody>
</table>

Proposed IPC-advanced Designs:

<table>
<thead>
<tr>
<th>GPU 0</th>
<th>Intra-node data transfer (via IPC)</th>
<th>Data placement</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU 0</td>
<td>Intra-node data transfer</td>
<td>Data placement</td>
</tr>
<tr>
<td>GPU 0</td>
<td>Intra-node data transfer</td>
<td>Data placement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intra-node data transfer</th>
<th>Inter-node data transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE 0 (GPU 0) NODE 1 (GPU 0)</td>
<td>NODE 0 (GPU 0) NODE 1 (GPU 1)</td>
</tr>
</tbody>
</table>

**Expected Benefit from overlap of intra and inter-node transfer**

- Intra-node communication via NVLink
- Inter-node communication via IB

© OpenFabrics Alliance
PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Host-Staging based Online Compression
  - GPU-aware IPC-advanced algorithm and hybrid designs
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan
OVERVIEW OF THE MVAPICH2 PROJECT

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), AWS EFA, Rockport Networks, and Slingshot
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs (NVIDIA and AMD)
- Started in 2001, first open-source version demonstrated at SC ‘02
- Supports the latest MPI-3.1 standard
- [http://mvapich.cse.ohio-state.edu](http://mvapich.cse.ohio-state.edu)

Additional optimized versions for different systems/environments:

- MVAPICH2-X (Advanced MPI + PGAS), since 2011
- MVAPICH2-GDR with support for NVIDIA (since 2014) and AMD (since 2020) GPUs
- MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
- MVAPICH2-Virt with virtualization support, since 2015
- MVAPICH2-EA with support for Energy-Awareness, since 2015
- MVAPICH2-Azure for Azure HPC IB instances, since 2019
- MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019

Tools:

- OSU MPI Micro-Benchmarks (OMB), since 2003
- OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015

- Used by more than 3,200 organizations in 89 countries
- More than 1.56 Million downloads from the OSU site directly

- Empowering many TOP500 clusters (Nov ‘21 ranking)
  - 4th, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 13th, 448,448 cores (Frontera) at TACC
  - 26th, 288,288 cores (Lassen) at LLNL
  - 38th, 570,020 cores (Nurion) in South Korea and many others

- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 13th ranked TACC Frontera system
- Empowering Top500 systems for more than 16 years
EXPERIMENTAL SETUP

- **Platform**
  - Frontera @TACC
  - Longhorn @TACC
  - 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

- **Benchmarks**
  - 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
BENCHMARK-LEVEL EVALUATIONS
Host-Staging based Online Compression

- **MPI_AlltoAll Communication Latency with OSU Micro-Benchmark**
  - The proposed design reduces the latency by up to **87.1%** for 16MB on 2nodes and 4nodes with ZFP-OPT (rate: 4)

---

**Q. Zhou, P. Kousha, Q. Anthony, K. Khorassani, A. Shafi, H. Subramoni, and D.K. Panda, “Accelerating MPI All-to-All Communication with Online Compression on Modern GPU Clusters”, ISC ’22 (Accepted to be presented).**
BENCHMARK-LEVEL EVALUATIONS

Host-Staging based Online Compression

- Compare with existing Alltoall algorithms with point-to-point compression in MVAPICH2-GDR-2.3.6
  - The proposed design reduces the Alltoall latency of 16MB by up to 11.2%, 17.8%, and 26.6% respectively compared to the Scatter Destination, Pairwise Exchange, and CPU Staging (w/o compression)

Compare with State-of-the-art MPI libraries
- The proposed designs provide the Alltoall latency of 16B by up to 1.4x, and of 64MB by up to 1.2x on 1 ThetaGPU nodes using 8 GPUs
- The proposed designs provide 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 similar trend is also noticed in Alltoallv benchmark

PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Host-Staging based Online Compression
  - GPU-aware IPC-advanced algorithm and hybrid designs
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan
APPLICATION-LEVEL EVALUATIONS
Host-Staging based Online Compression

- Improvement compared to MVAPICH2-GDR-2.3.6 with Point-to-Point compression
  - PSDNS: Reduce All-to-All runtime by up to 29.2% with ZFP(rate: 24) on 64 GPUs
  - DeepSpeed benchmark: Reduce All-to-All runtime by up to 35.8% with ZFP(rate: 16) on 32 GPUs

APPLICATION-LEVEL EVALUATIONS

GPU-aware IPC-advanced Algorithm and Hybrid Designs

- **DeepSpeed**
  - The proposed designs provide 7x throughput for small tensors (1K x 16) against OpenMPI on 8 ThetaGPU nodes (64 GPUs)
  - It also designs provide 60x throughput for large tensors (16M x 16) against OpenMPI on 8 ThetaGPU nodes (64 GPUs)

**heFFTe**

- The proposed designs support common datatype 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.

---

*C.-C. Chen, K. Shafie Khorassani, Q. Anthony, A. Shafi, H. Subramoni and D. Panda, “Highly Efficient Alltoall and Alltoallv Communication Algorithms for GPU Systems”, HCW ’22 (Accepted to be presented)*
### PSDNS

- The proposed designs provide **1.3x** speedup on 1 Lassen nodes (4 GPUs) at large size case ($512^3$)
- It provides **3.5x** speedup on 64 Lassen nodes (256 GPUs) at large size case ($1024^3$)

---

**PSDNS latency on 1 node (4 GPUs)**

- **Proposed**
- **MVAPICH2-GDR 2.3.6**

---

**PSDNS latency on 4 node (16 GPUs)**

- **Proposed**
- **MVAPICH2-GDR 2.3.6**

---

**PSDNS latency on 16 node (64 GPUs)**

- **Proposed**
- **MVAPICH2-GDR 2.3.6**

---

**PSDNS latency on 64 node (256 GPUs)**

- **Proposed**
- **MVAPICH2-GDR 2.3.6**

---


© OpenFabrics Alliance
PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Host-Staging based Online Compression
  - GPU-aware IPC-advanced algorithm and hybrid designs
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan
CONCLUSION AND FUTURE PLAN

- We improved the performance of GPU-based Alltoall and Alltoallv MPI collective calls on dense GPU systems and proposed two new designs: **Host-Staging based Online Compression** and **GPU-aware IPC-advanced hybrid design**.

- The **Host-Staging based Online Compression** reduces Alltoall runtime by up to **29.2%** with ZFP(rate:24) in PSDNS application on 64 GPUs and **35.8%** with ZFP(rate:16) in DeepSpeed benchmark on 32 GPUs.

- In HPC application heFFTe, the **GPU-aware IPC-advanced hybrid design** reaches approximately **27x** better performance on ThetaGPU.

- As future work, we plan to extend our designs to other common collectives, such as broadcast, gather(v), scatter(v) and allgather(v).
THANK YOU!

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

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

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