OPENFABRICS ALLIANCE

2024 OFA Virtual Workshop

ACCELERATING MPI ALLREDUCE COMMUNICATION WITH EFFICIENT GPU-BASED COMPRESSION SCHEMES ON MODERN GPU CLUSTERS

Hari Subramoni and Qinghua Zhou

The Ohio State University

Email:

{subramoni.1,Zhou.2595}@osu.edu

PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Ring AllReduce with Collective-level Online Compression
  - Recursive-Doubling AllReduce with Collective-level Online Compression
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan

INTRODUCTION AND MOTIVATION

- **AllReduce** is a communication collective operation that is commonly used in HPC applications as well as distributed DL training.
- Existing AllReduce algorithms for transferring large GPU data still suffer from poor performance due to the limited interconnect bandwidth of networks
- Naive point-to-point compression for each data transmission may introduce redundant compression/decompression operations and hinder non-blocking send/receive operations
- How to co-design and optimize the **GPU-based compression** at the **collective-level** along with the communication patterns of advanced AllReduce algorithms?
- We propose two design approaches along with these directions.
  - **Ring AllReduce** with Collective-level Online Compression
  - **Recursive-Doubling AllReduce** with Collective-level Online Compression

PRESENTATION OVERVIEW

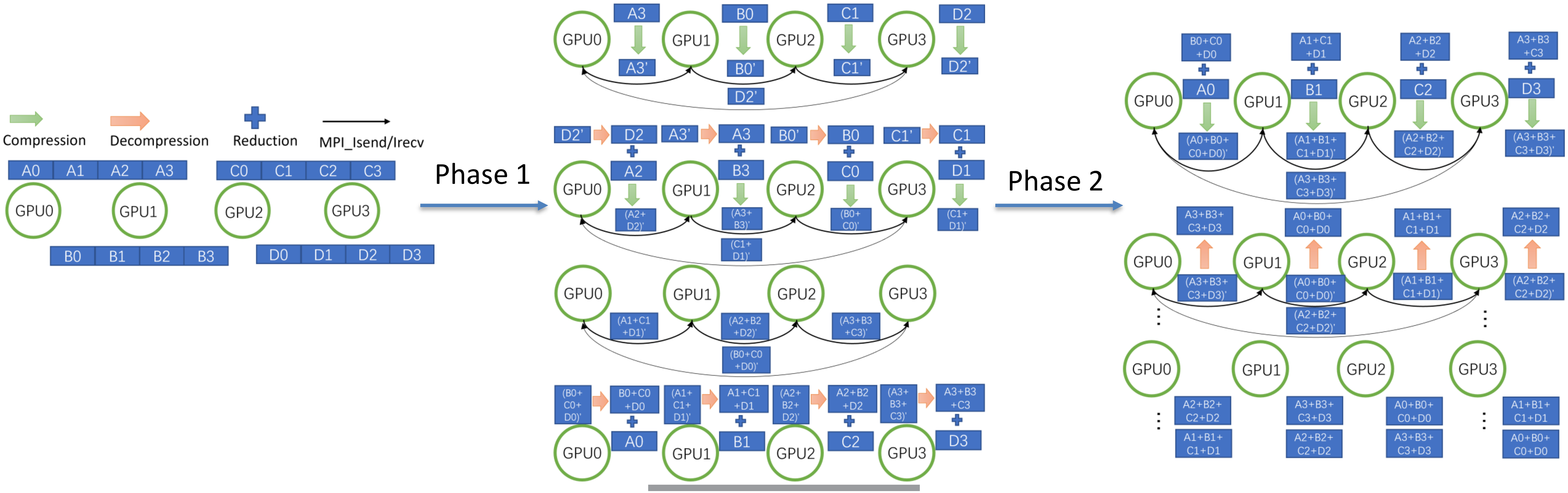
- Introduction & Motivation
- Design Approaches
  - Ring AllReduce with Collective-level Online Compression
  - Recursive-Doubling AllReduce with Collective-level Online Compression
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan

DESIGN APPROACHES

- **Ring and Recursive-Doubling MPI_AllReduce with Collective-level 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
    - Hinder the non-blocking send/receive operations in AllReduce
  - Propose a collective-level online compression for Ring based MPI_Allreduce
  - Propose a collective-level online compression for Recursive-Doubling MPI_Allreduce
  - Optimize the ZFP compression library to enable execution of compression/decompression/reduction kernels on multiple CUDA streams
  - Achieve overlap between the compression/decompression/reduction kernels and send/receive operations

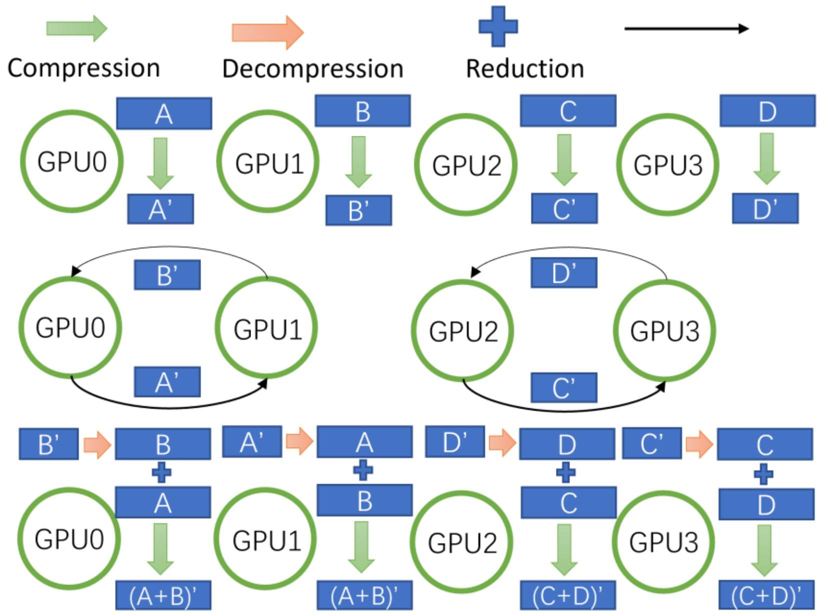
RING ALLREDUCE WITH COLLECTIVE-LEVEL ONLINE COMPRESSION

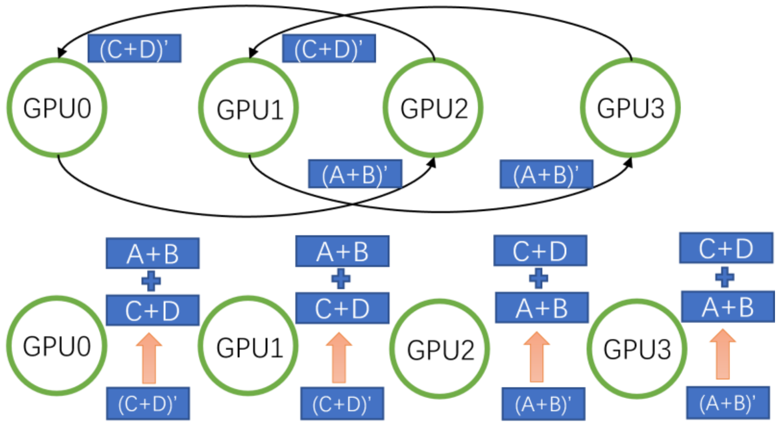
- **Data Flow of Ring AllReduce with Collective-level Online Compression**
  - Data on each GPU is split to multiple chunks
  - Phase1: Aggregate the values on each GPU in Ring manner
  - Phase2: Transfer the aggregated values to all GPUs in Ring manner
  - Compression/decompression run on each data chunk and aggregated value



RECURSIVE-DOUBLING ALLREDUCE WITH COLLECTIVE-LEVEL ONLINE COMPRESSION

- **Data Flow of Recursive-Doubling AllReduce with Collective-level Online Compression**
  - Specific pairs of processes exchange messages with each other in a pairwise manner
  - Whole data on each GPU is compressed and decompressed
  - Fewer data exchanges are needed across the processes, thus fewer compression operations
  - Achieve overlap between the compression/decompression/reduction kernels and send/receive operations





PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Ring AllReduce with Collective-level Online Compression
  - Recursive-Doubling AllReduce with Collective-level Online Compression
- 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**
- **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 3,375 organizations in 91 countries**
- **More than 1.75 Million downloads from the OSU site directly**
- Empowering many TOP500 clusters (June '23 ranking)
  - 11th , 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 29th, 448, 448 cores (Frontera) at TACC
  - 46th, 288,288 cores (Lassen) at LLNL
  - 61st, 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 29th ranked TACC Frontera system
- **Empowering Top500 systems for more than 18 years**

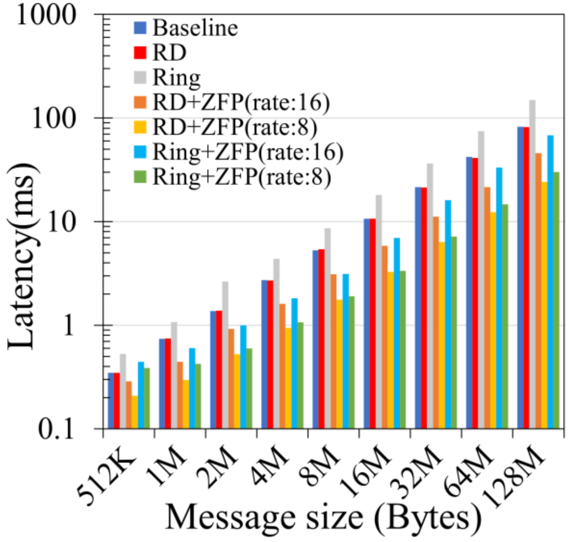
EXPERIMENTAL SETUP

- **Platform**
  - Pitzer @OSC (V100 GPU)
  - MRI @OSU (A100 GPU)
  - Frontera @TACC (RTX5000 GPU)
  - Lassen @LLNL (V100 GPU)
- **Baselines**
  - MVAPICH2-GDR 2.3.7
- **Benchmarks**
  - Benchmark-level evaluations:
    - `osu_allreduce` in OSU Micro-Benchmarks (OMB) suite
  - Application-level evaluations:
    - Distributed Data Parallel (DDP) training of DNN models with PyTorch

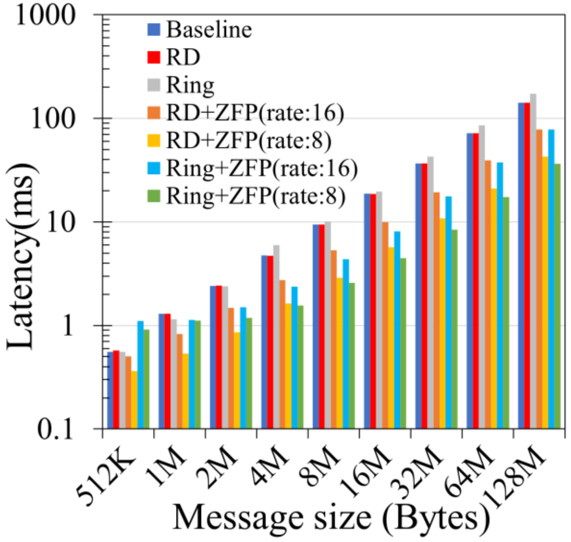
BENCHMARK-LEVEL EVALUATIONS

Ring and Recursive-Doubling AllReduce with Collective-level Online Compression

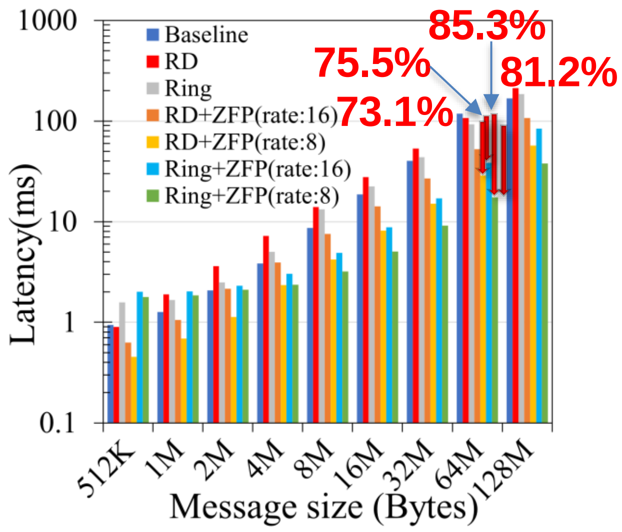
- **MPI_Allreduce Communication Latency with OSU Micro-Benchmark on Pitzer (V100 GPUs)**
  - Ring AllReduce with compression (Ring+ZFP)
    - Reduces the latency by **81.2%** (64MB, 16 GPUs, rate:8) vs. original Ring, **85.3%** (64MB, 16 GPUs) vs. Baseline.
  - Recursive-Doubling with compression (RD+ZFP)
    - Reduces the latency by **73.1%** (64MB, 16 GPUs, rate:8) vs. original RD, **75.5%** (64MB, 16 GPUs) vs. Baseline.



Pitzer: 4 GPUs
(2 nodes, 2 ppn)



Pitzer: 8 GPUs
(4 nodes, 2 ppn)



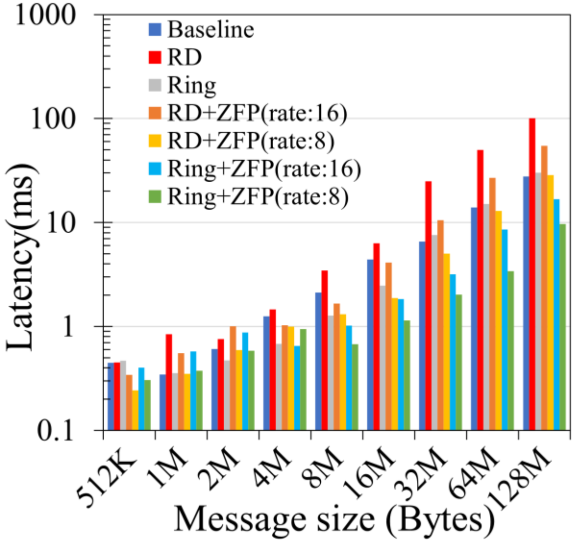
Pitzer: 16 GPUs
(8 nodes, 2 ppn)

Q. Zhou, B. Ramesh, A. Shafi, M. Abduljabbar, H. Subramoni, and D.K. Panda, "Accelerating MPI AllReduce Communication with Efficient GPU-Based Compression Schemes on Modern GPU Clusters", ISC '24 (Accepted to be presented).

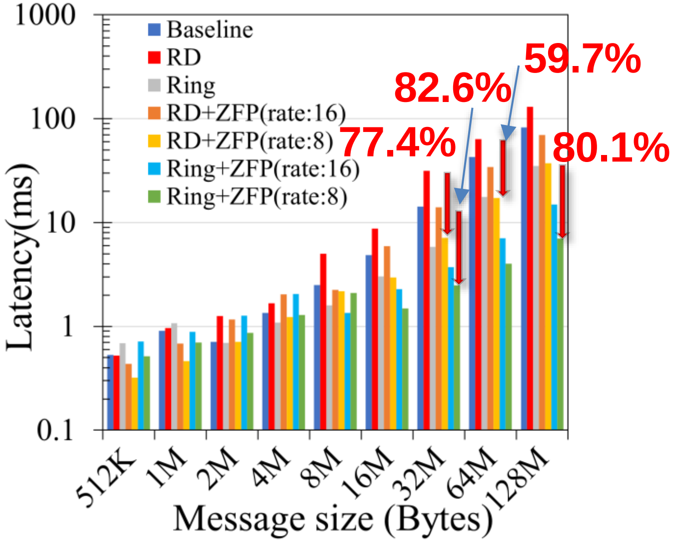
BENCHMARK-LEVEL EVALUATIONS

Ring and Recursive-Doubling AllReduce with Collective-level Online Compression

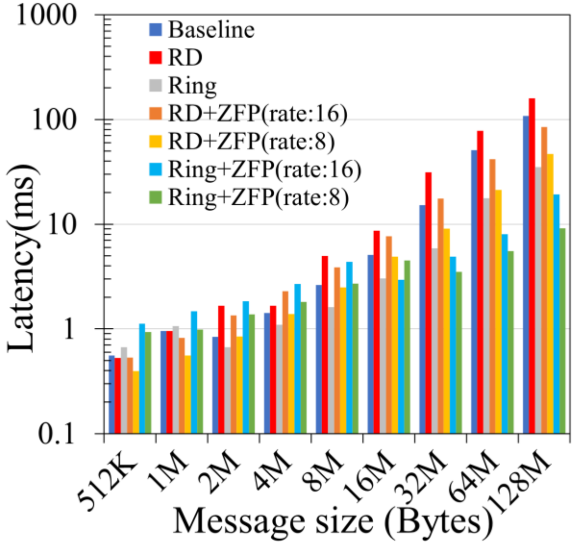
- **MPI_Allreduce Communication Latency on MRI system (A100 GPUs)**
  - Ring AllReduce with compression (Ring+ZFP)
    - Reduces the latency by **80.1%** (128MB, 8 GPUs, rate:8) vs. original Ring, **82.6%** (32MB, 8 GPUs) vs. Baseline.
  - Recursive-Doubling with compression (RD+ZFP)
    - Reduces the latency by **77.4%** (32MB, 8 GPUs, rate:8) vs. original RD, **59.7%** (64MB, 8 GPUs) vs. Baseline.



MRI: 4 GPUs
(2 nodes, 2 ppn)



MRI: 8 GPUs
(4 nodes, 2 ppn)

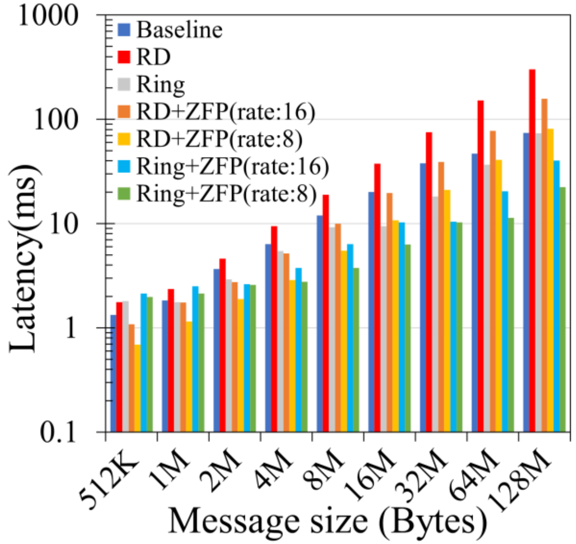


MRI: 16 GPUs
(8 nodes, 2 ppn)

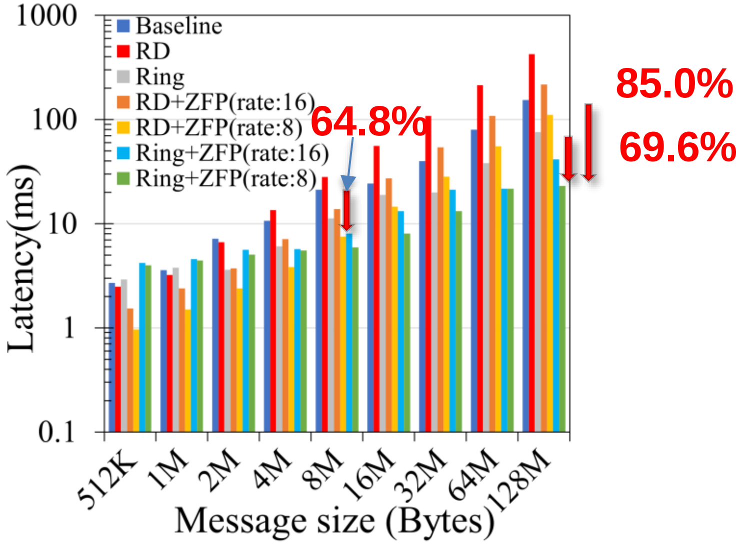
BENCHMARK-LEVEL EVALUATIONS

Ring and Recursive-Doubling AllReduce with Collective-level Online Compression

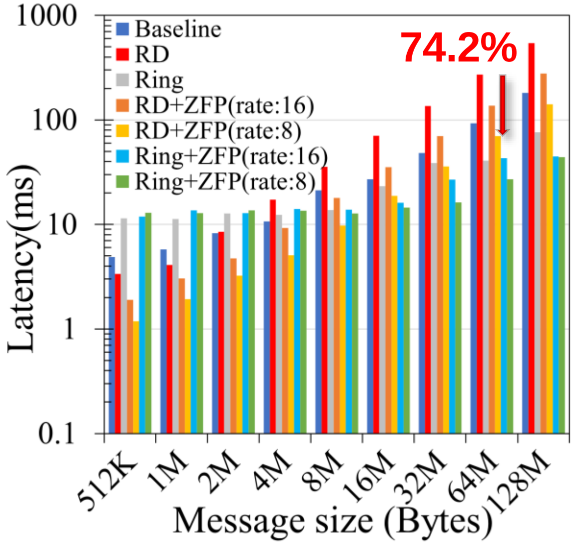
- **MPI_Allreduce Communication Latency on Frontera Liquid system (RTX5000 GPUs)**
  - Ring AllReduce with compression (Ring+ZFP)
    - Reduces the latency by **69.6%** (128MB, 32 GPUs, rate:8) vs. original Ring, **85.0%** (128MB, 32 GPUs) vs. Baseline.
  - Recursive-Doubling with compression (RD+ZFP)
    - Reduces the latency by **74.2%** (64MB, 64 GPUs, rate:8) vs. original RD, **64.8%** (8MB, 32 GPUs) vs. Baseline.



Frontera: 16 GPUs
(4 nodes, 4 ppn)



Frontera: 32 GPUs
(8 nodes, 4 ppn)

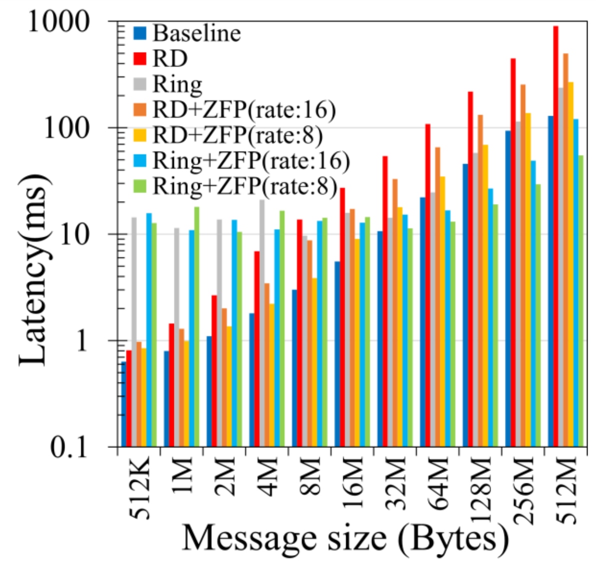


Frontera: 64 GPUs
(16 nodes, 4 ppn)

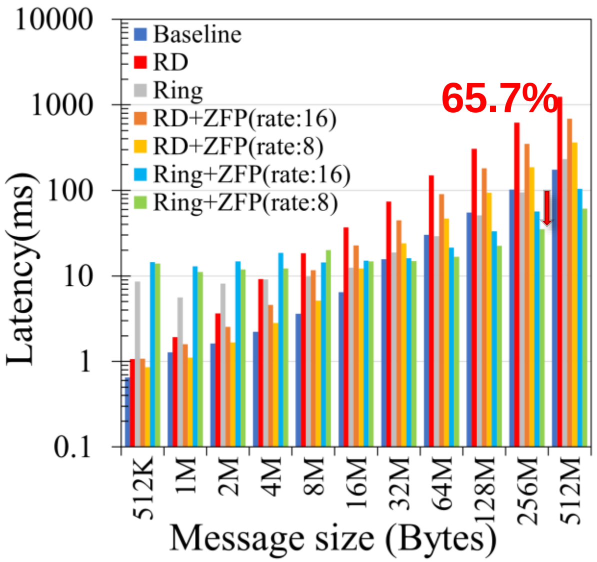
BENCHMARK-LEVEL EVALUATIONS

Ring and Recursive-Doubling AllReduce with Collective-level Online Compression

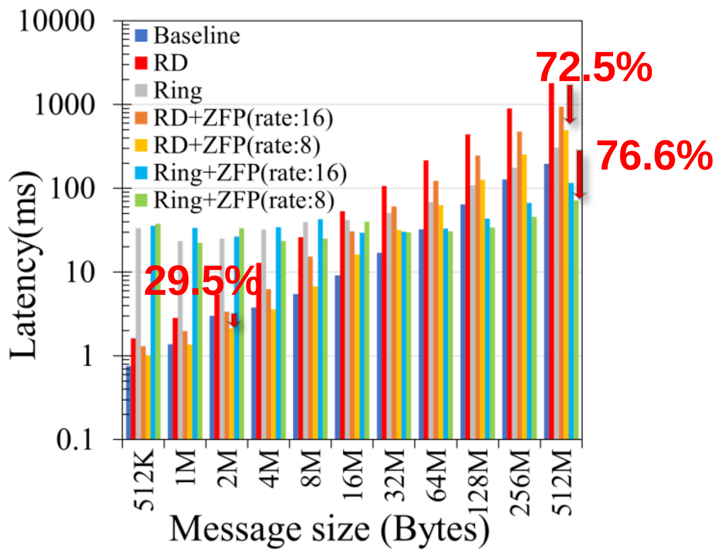
- **MPI_Allreduce Communication Latency on Lassen system (V100 GPUs)**
  - Ring AllReduce with compression (Ring+ZFP)
    - Reduces the latency by **76.6%** (512MB, 256 GPUs, rate:8) vs. original Ring, **65.7%** (256MB, 128 GPUs) vs. Baseline.
  - Recursive-Doubling with compression (RD+ZFP)
    - Reduces the latency by **72.5%** (512MB, 256 GPUs, rate:8) vs. original RD, **29.5%** (2MB, 256 GPUs) vs. Baseline.



Lassen: 64 GPUs
(16 nodes, 4 ppn)



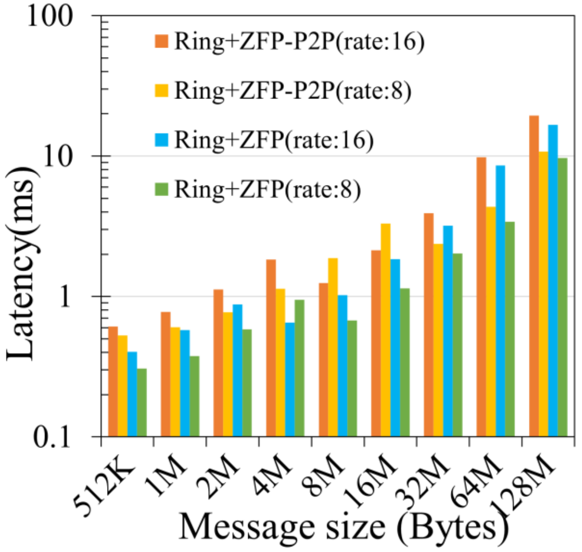
Lassen: 128 GPUs
(32 nodes, 4 ppn)



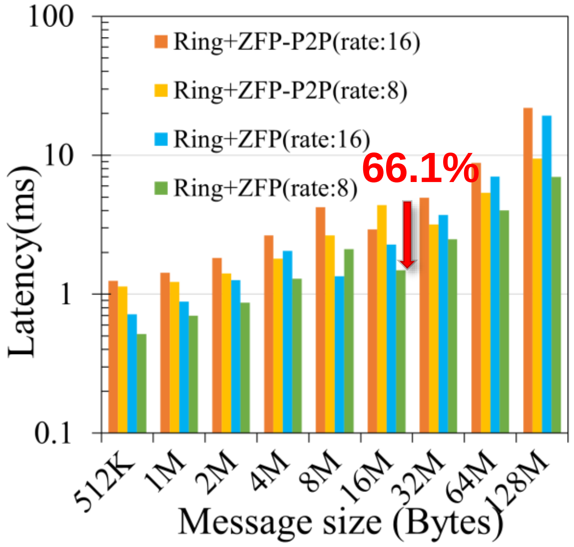
Lassen: 256 GPUs
(64 nodes, 4 ppn)

BENCHMARK-LEVEL EVALUATIONS

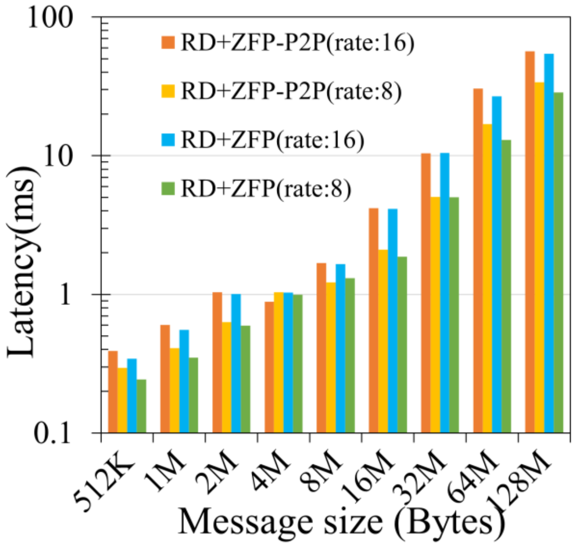
Compare Collective-level Compression with Point-to-Point Compression



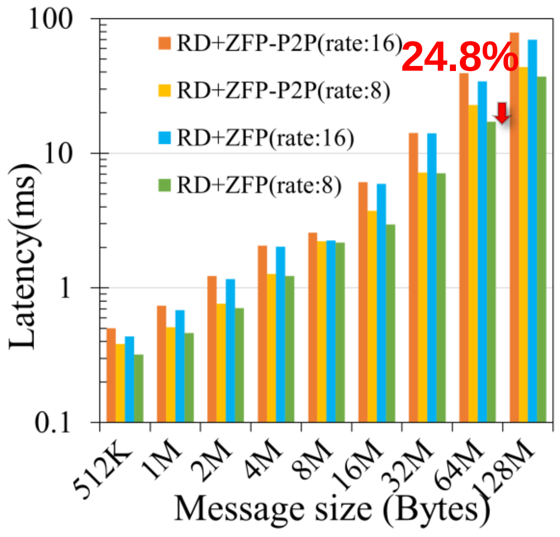
Ring (Pitzer: 4 GPUs)



Ring (Pitzer: 8 GPUs)



Recursive-Doubling (Pitzer: 4 GPUs)



Recursive-Doubling (Pitzer: 8 GPUs)

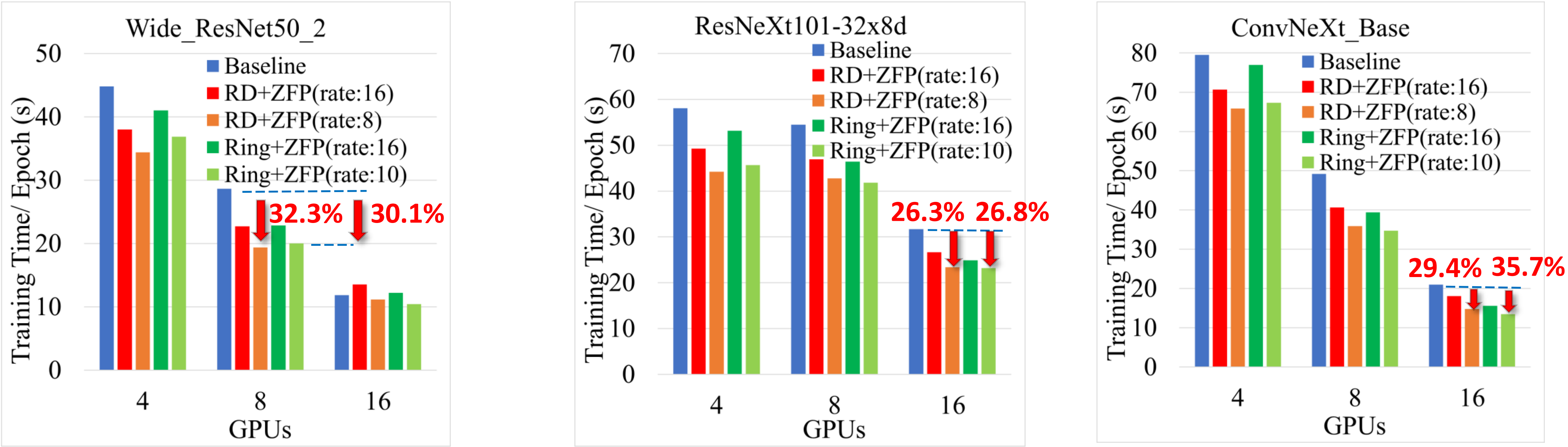
- **Compare with Ring and RD AllReduce algorithms with point-to-point compression in MVAPICH2-GDR-2.3.7**
  - Ring AllReduce with compression reduces the latency by **66.1%** (16MB, 8 GPUs, rate:8) vs. P2P compression (Ring+ZFP-P2P)
  - RD AllReduce with compression reduces the latency by **24.8%** (64MB, 8 GPUs, rate:8) vs. P2P compression (RD+ZFP-P2P)

PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Ring AllReduce with Collective-level Online Compression
  - Recursive-Doubling AllReduce with Collective-level Online Compression
- **Performance Evaluation**
  - Benchmark-level evaluation
  - Application-level evaluation
- **Conclusion and Future Plan**

APPLICATION-LEVEL EVALUATIONS

DDP training of DNN models using PyTorch

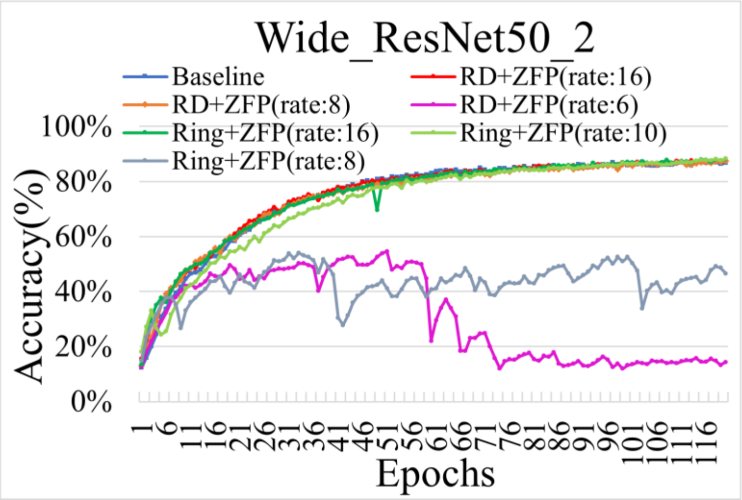


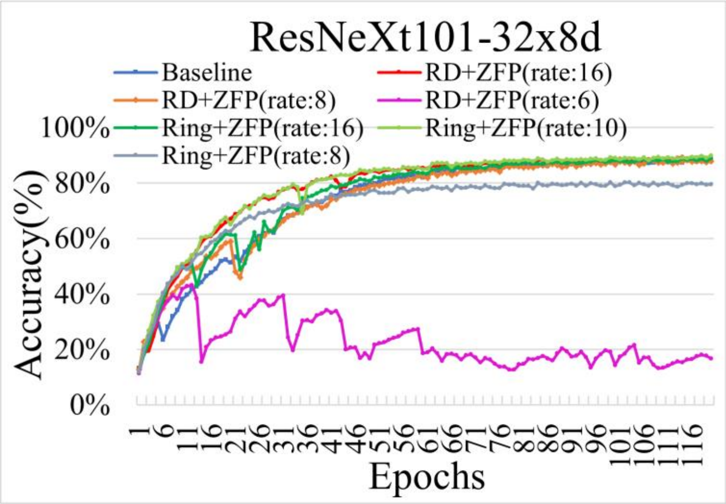
DDP training performance on Pitzer (V100 GPUs), Dataset: CIFAR10, Batch Size=128, Learning Rate=0.001

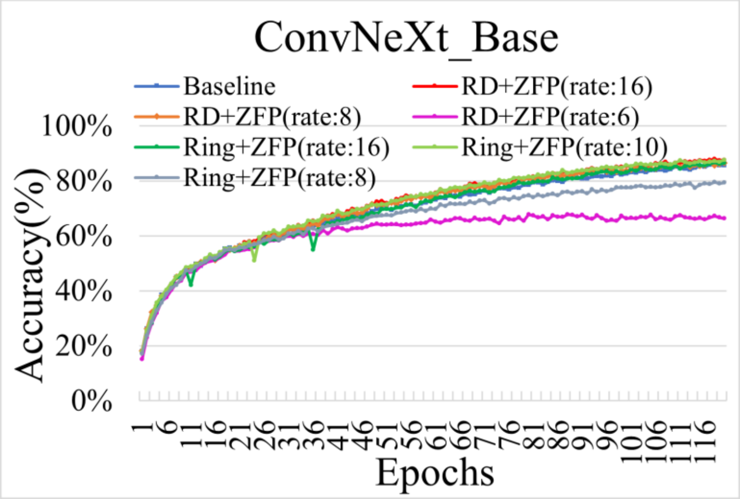
- **Improvement for DDP training of DNN models using PyTorch**
  - Use MPI backend with proposed Ring and Recursive-Doubling AllReduce with Online compression design
  - **Wide_ResNet50_2**: Reduces the training time by **30.1%** (Ring+ZFP, 8 GPUs, rate: 10), **32.3%** (RD+ZFP, 8 GPUs, rate: 8)
  - **ResNeXt101-32x8d**: Reduce the training time by **26.8%** (Ring+ZFP, 16 GPUs, rate: 10), **26.3%** (RD+ZFP, 16 GPUs, rate: 8)
  - **ConvNeXt_Base:** Reduce the training time by **35.7%** (Ring+ZFP, 16 GPUs, rate: 10), **29.4%** (RD+ZFP, 16 GPUs, rate: 8)

APPLICATION-LEVEL EVALUATIONS

DDP training of DNN models using PyTorch







DDP training accuracy on Pitzer (V100 GPUs), Dataset: CIFAR10, Batch Size=128, Learning Rate=0.001

- **DDP training accuracy with Ring and Recursive-Doubling Allreduce with Online Compression**
  - Achieves similar convergent training accuracy with proposed Ring+ZFP(rate: 16, 10) and RD+ZFP (rate:16, 8) vs. Baseline
  - Big accuracy drop with lower compression (Ring+ZFP(rate: 8), RD+ZFP(rate:6) due to larger compression errors added to gradients

PRESENTATION OVERVIEW

- Introduction & Motivation
- Design Approaches
  - Ring AllReduce with Collective-level Online Compression
  - Recursive-Doubling AllReduce with Collective-level Online Compression
- Performance Evaluation
  - Benchmark-level evaluation
  - Application-level evaluation
- Conclusion and Future Plan

CONCLUSION AND FUTURE PLAN

- We proposed **Collective-level online GPU-based Compression** design for **Ring** and **Recursive-Doubling** MPI_Allreduce communication in an MPI library on Modern GPU clusters
- At the benchmark level, the Ring and Recursive-Doubling AllReduce with online compression reduces communication latency by up to **85.3%** and **75.5%** respectively compared to the baseline, and by up to **66.1%** and **24.8%** respectively compared to point-to-point compression.
- In PyTorch DDP training, the Ring and Recursive-Doubling AllReduce with collective-level online compression reduce the training time by up to **35.7%** and **32.3%** respectively
- As future work, we plan to we intend to design compression schemes for other parallel strategies to accelerate the distributed training of larger DL models.

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/