Accelerating Deep Learning Applications with GPU-Based On-the-Fly Compression

Presentation at GTC '24

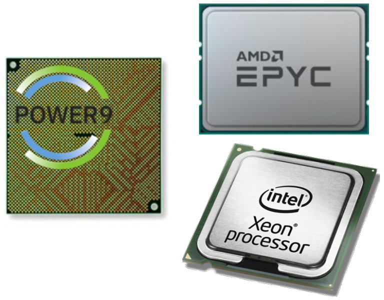
Hari Subramoni

Network Based Computing Laboratory (NBCL)

Dept. of Computer Science and Engineering, The Ohio State University

subramoni.1@osu.edu

Trends in Modern HPC Systems: Interconnects lag behind

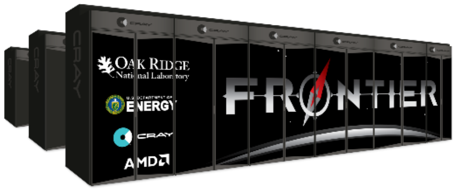


Multi/ Many-core
Processors

Accelerators (GPUs, FPGA)
High compute power
High peak memory bandwidth
(H100: 24576 Gb/s memory bandwidth,
7200 Gb/s NVLINK)

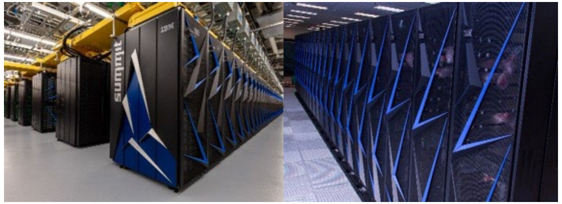
High Performance Interconnects
InfiniBand, Omni-Path, EFA
<1usec latency, 200Gbps+ Bandwidth

SSD, NVMe-SSD, NVRAM
Node local storage



#1 Frontier
AMD Instinct MI250X
(37632 GPUs)

#2 Fugaku
(158,976 nodes with A64FX
ARM CPU, a GPU-like processor)



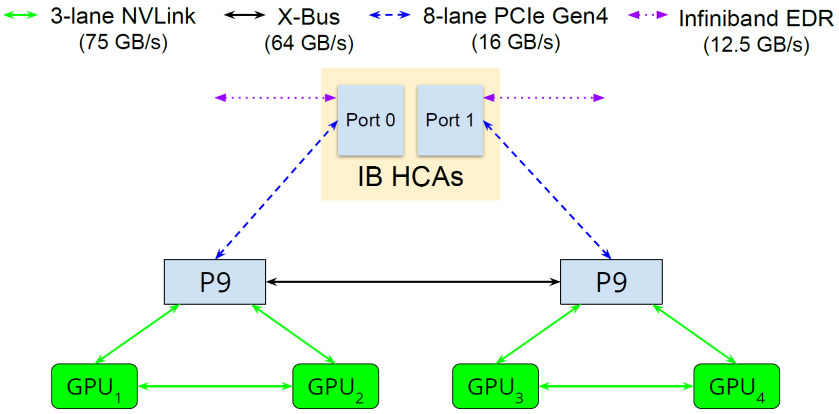
#5 Summit (27,648 GPUs)
#6 Sierra (17,280 GPUs)

#9 Selene
NVIDIA DGX A100 SuperPOD
(2,240 GPUs)

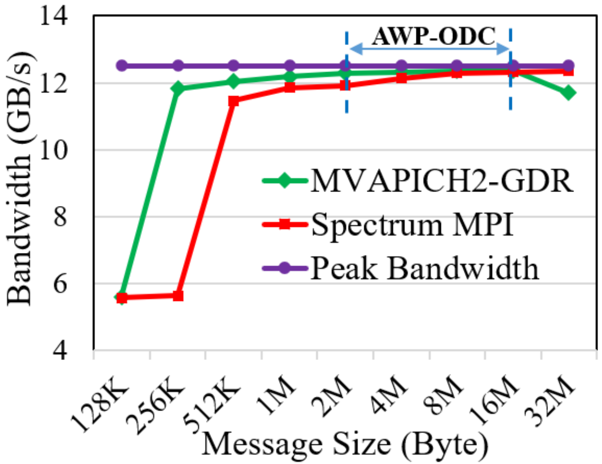
https://www.top500.org/

Motivation

- Disparity between intra-node and inter-node GPU communication prevents efficiently scaling applications to larger GPU systems
- Bandwidth of IB network is saturated for large message



(a) Disparity between intra-node and inter-node GPU communication on Sierra OpenPOWER supercomputer [1]



(b) Saturated bandwidth at large message size

[1] K. S. Khorassani, C.-H. Chu, H. Subramoni, and D. K. Panda, "Performance Evaluation of MPI Libraries on GPU-enabled OpenPOWER Architectures: Early Experiences", in International Workshop on Open-POWER for HPC (IWOPH 19) at the 2019 ISC High Performance Conference, 2018.

Research Challenges

- For HPC and Deep Learning applications on modern GPU clusters
  - What are the other techniques—besides improving the communication bandwidth—that can be used to reduce the communication time?
  - ✓ **Compression** can reduce the data size and lower the pressure on network with limited bandwidth
  - How can we design efficient **on-the-fly** message compression schemes to improve the performance of these applications?
  - ✓ We integrate **GPU-based compression** algorithms into MVAPICH2-GDR with optimization to achieve high performance on-the-fly message compression for
    - ✓ Point-to-point operations
    - ✓ Various collective operations (Alltoall, Allgather, Broadcast, Reduce Scatter)

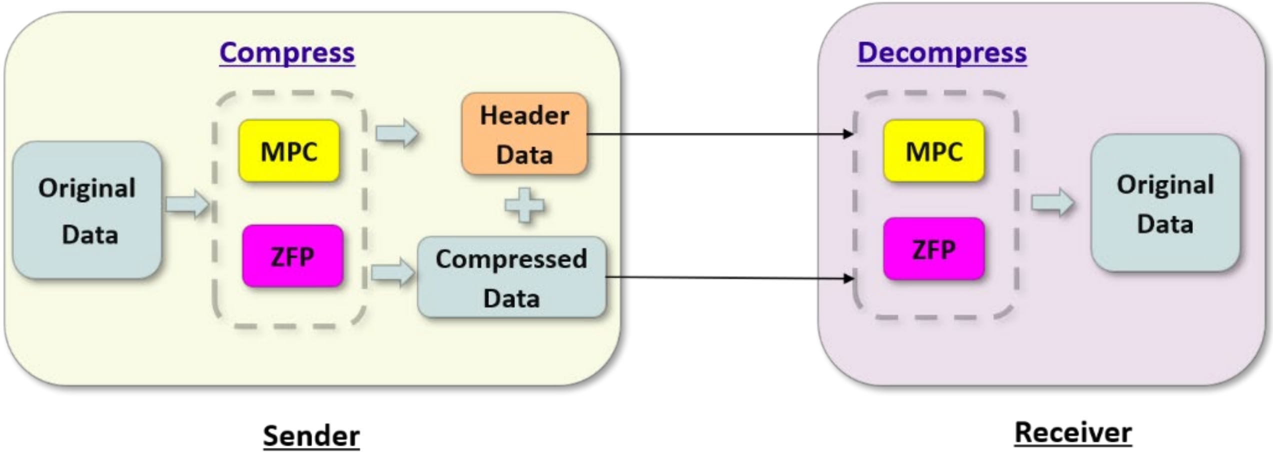
Overview of the MVAPICH2 Project

- **High Performance open-source MPI Library**
- **Support for multiple interconnects**
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), and 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 1.76 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**

Framework of GPU-based on-the-fly compression

- Compression algorithms MPC and ZFP are integrated into MVAPICH2-GDR
- Rendezvous protocol is used to send the header data and compressed data

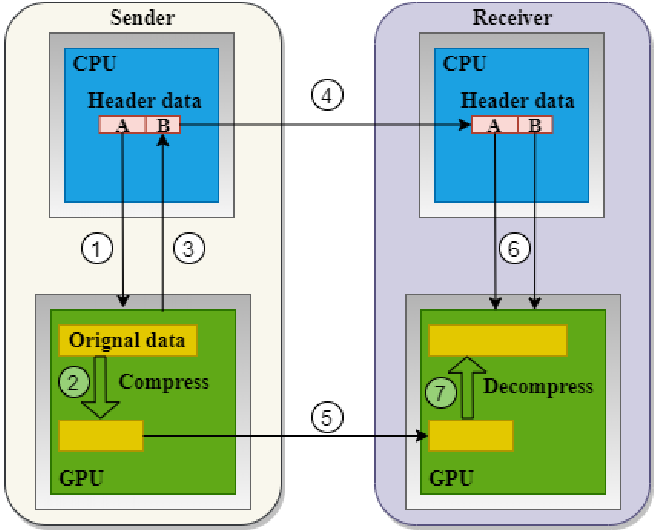


Framework of GPU-based on-the-fly compression [2]

[2] Q. Zhou, C. Chu, N. Senthil Kumar, P. Kousha, M. Ghazimirsaeed, H. Subramoni, D. Panda, “Designing High-Performance MPI Libraries with On-the-fly Compression for Modern GPU Clusters”, in 35th IEEE International Parallel & Distributed Processing Symposium, May 2021. [Best Paper Finalist]

Data Flow: Point-to-Point On-the-fly Compression

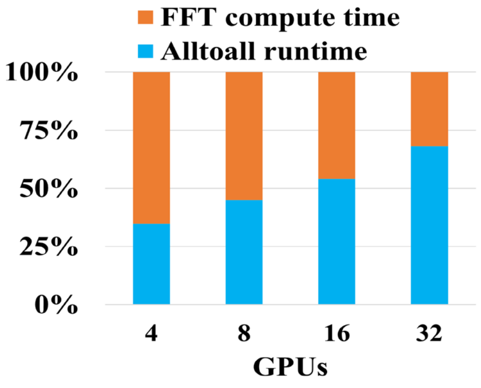
- Data flow
  1. Launch compression kernel with control parameters
  2. Run compression kernel on GPU
  3. Return compressed size
  4. Send header data with RTS packet
  5. Send compressed GPU data
  6. Launch decompression kernel with header data
  7. Run decompression kernel to restore the data.



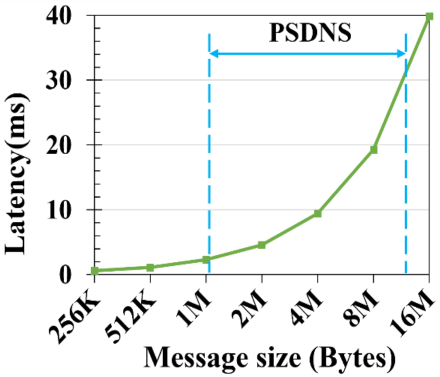
Data flow of GPU communication with Point-to-Point On-the-fly compression

Limitation of Point-to-Point compression for Alltoall

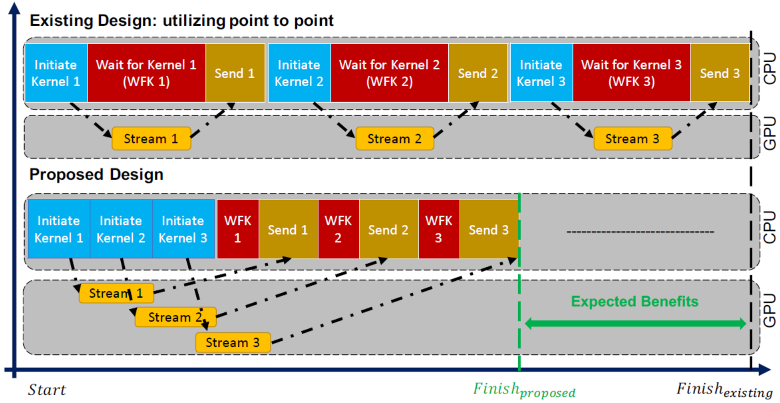
- AlltoAll is one of the most communication-intensive MPI operations that become the bottleneck of efficiently scaling these applications(e.g, PSDNS, DeepSpeed) to larger dense GPU systems
- Existing Point-to-Point based compression has limitation of overlapping compression/decompression kernels across send/receive operations.
- How to overcome the limitation of Point-to-Point based compression to accelerate applications?
  – Move the point-to-point compression to the collective-level
  – Revamp and optimize GPU-based compression for the collective-level online compression



(a) PSDNS Time Breakdown



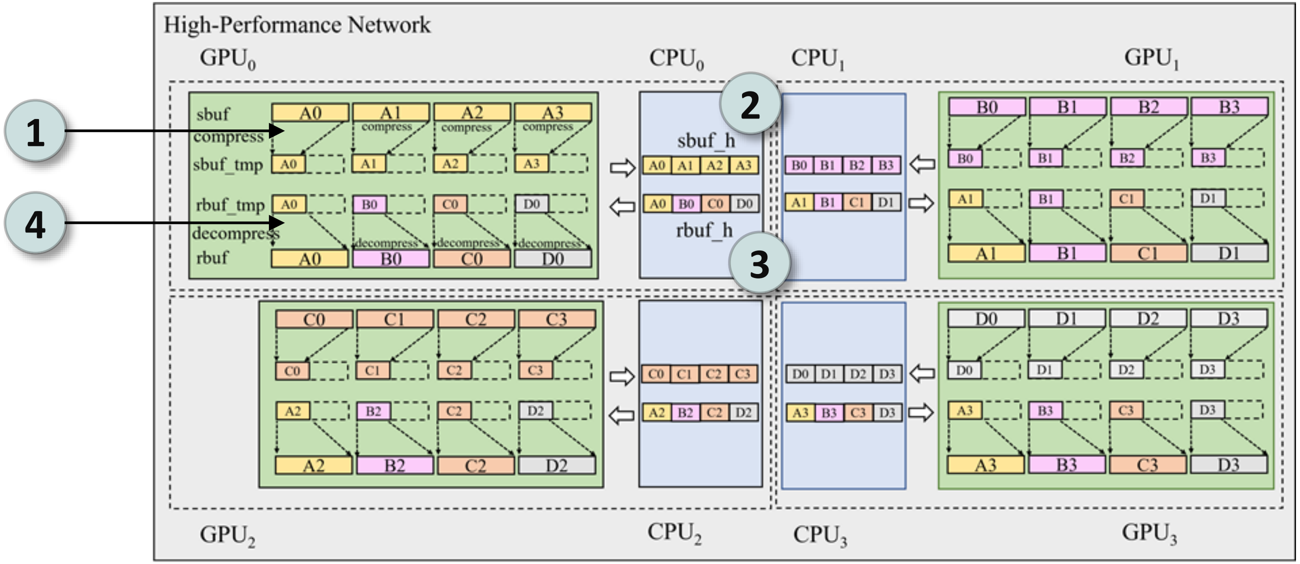
(b) AlltoAll Latency for 8 GPUs on 2 Longhorn(V100) nodes



Compare point-to-point compression operations versus proposed design

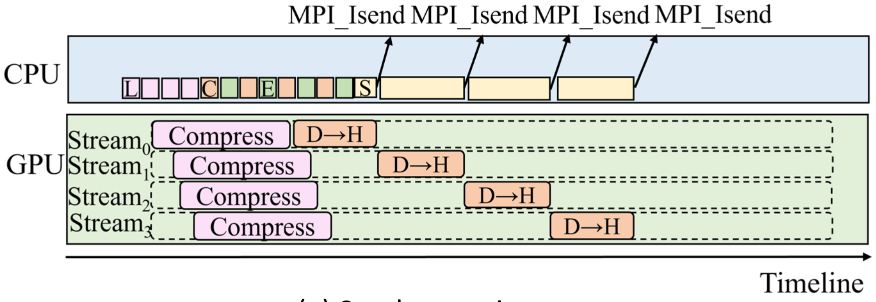
Host-Staging based Collective-level Compression

- Data Flow of Host-Staging based Collective-level 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. Received data is copied to the temporary device buffer asynchronously and decompressed to the target buffer

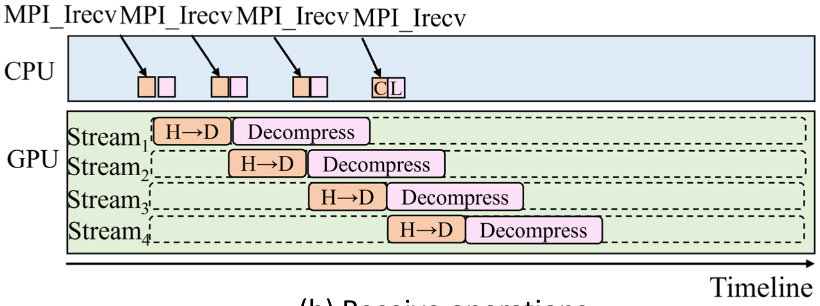


Optimization for Host-staging based Compression

- Enabling Multiple CUDA Streams in ZFP Library
  – Design new APIs zfp_compress_multi_stream and zfp_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. Use same stream for data movement (D->H, H->D) and the corresponding compression/decompression kernels
  – 3. Achieve overlap between the compression/decompression kernels across multiple send/receive operations



(a) Send operations

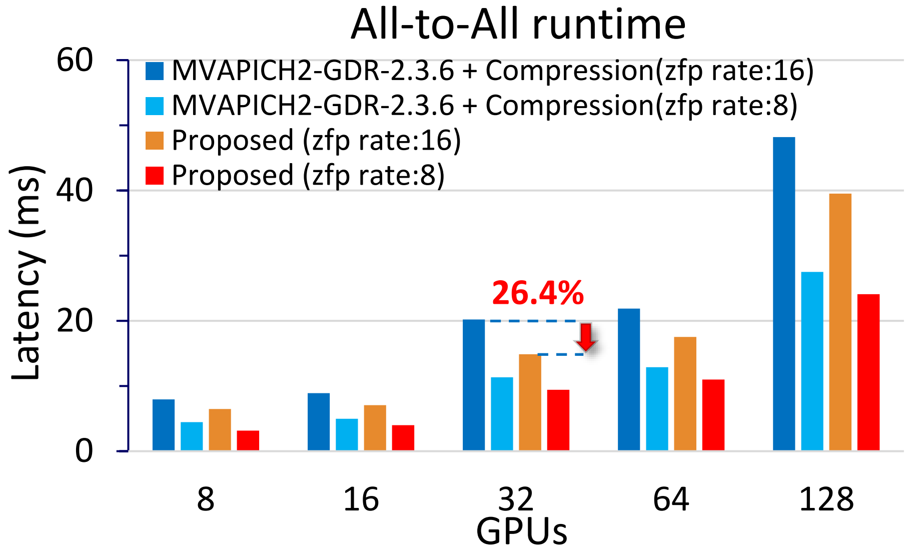


(b) Receive operations

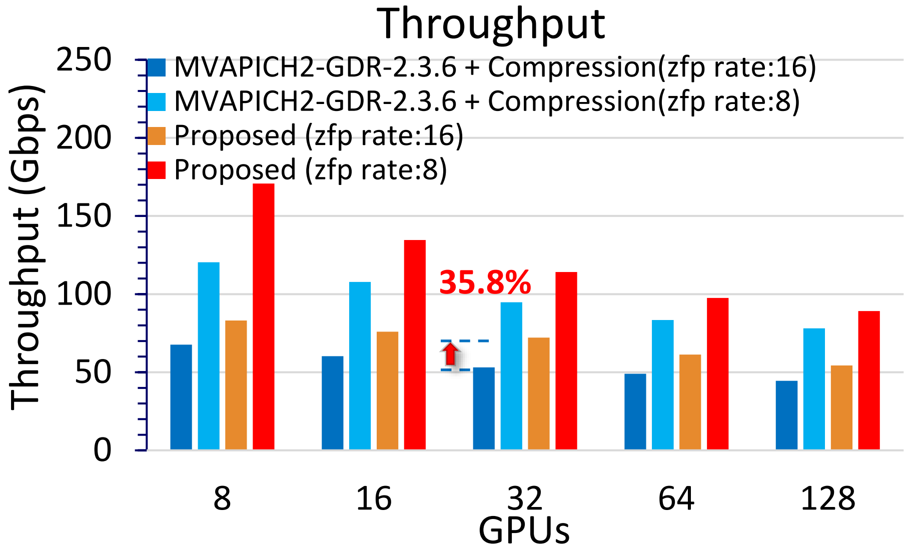
[3] Q. Zhou, P. Kousha, Q. Anthony, K. Khorassani, A. Shafi, H. Subramoni, and D. K. Panda, Accelerating MPI All-to-All Communication with Online Compression on Modern GPU Clusters. ISC High Performance 2022, May 2022.

Application-Level Evaluations (DeepSpeed Benchmark)

- Improvement compared to MVAPICH2-GDR with Point-to-Point compression
  - Reduces All-to-All runtime by up to **26.4%** with ZFP(rate: 16) on 32 GPUs
  - Improves the throughput by up to **35.8%** with ZFP(rate: 16) on 32 GPUs



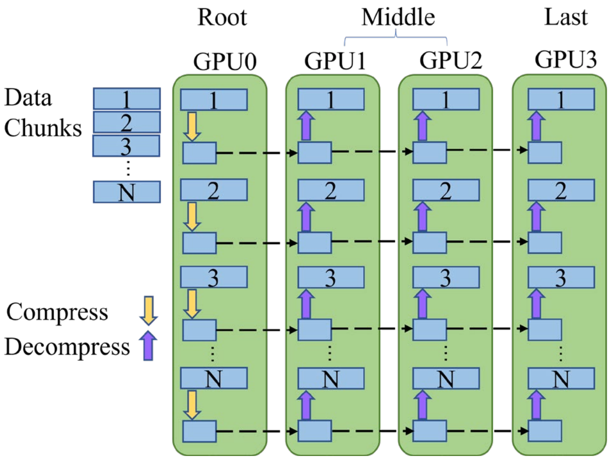
Lassen system (NVIDIA V100)



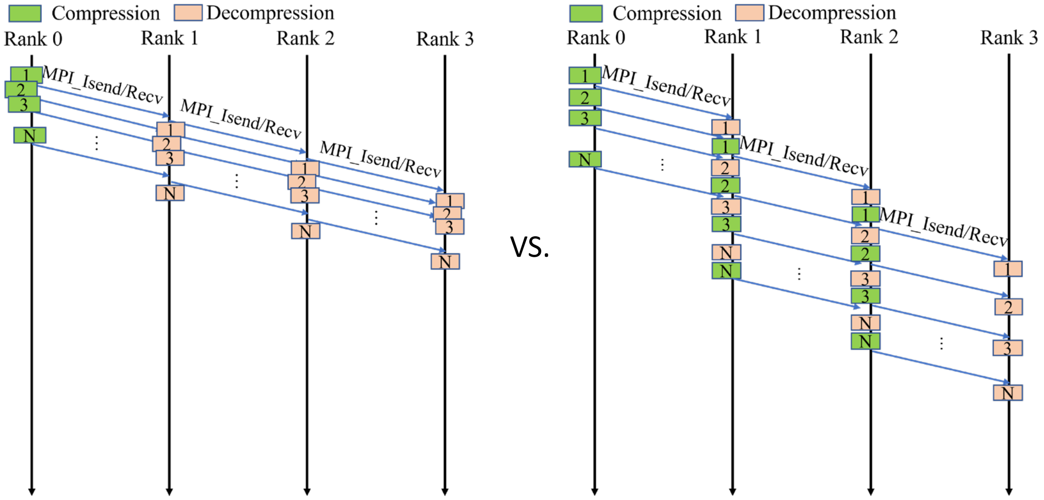
Lassen system (NVIDIA V100)

Broadcast with Collective-level Online Compression

- Chunked-Chain based Broadcast with Collective-level Online Compression
  - Launch ZFP compression/decompression kernel on non-default CUDA stream to achieve overlap
  - Middle ranks send the received compressed data to the right rank and only run decompression
  - Launch an MPI_Bcast operation to transfer the compressed message sizes of all the chunks



(a) Data Flow of Broadcast with Collective-level Compression

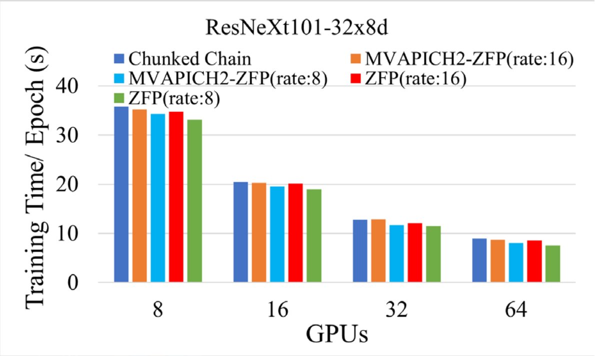


(b) Collective-level Compression vs. Point-to-Point Compression

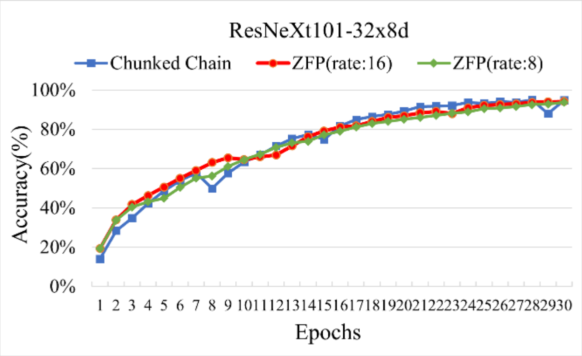
Q. Zhou, Q. Anthony, A. Shafi, H. Subramoni, and D. K. Panda, Accelerating Broadcast Communication with GPU Compression for Deep Learning Workloads, 29th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC ’22), Dec 2022.

Application-Level Evaluations (PyTorch DDP training)

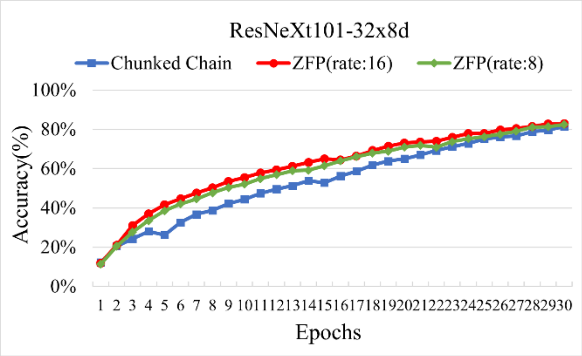
- PyTorch (v1.12) DDP training with ZeroRedundancyOptimizer on CIFAR10 dataset
- Improvements compared to original Chunked-Chain and Point-to-Point compression
  - Reduces training time by up to **15.0%** with ZFP(rate: 8) on 64 GPUs vs. Chunked-Chain
  - Reduces training time by up to **6.4%** with ZFP(rate: 8) on 64 GPUs vs. Point-to-Point compression
  - Training accuracy converges to similar value as original Chunked-Chain Broadcast



Training Time per epoch
(BS=128, LR=0.001) on Longhorn (NVIDIA V100)



Training Accuracy
(8 GPUs) on Bridges-2 (NVIDIA V100)

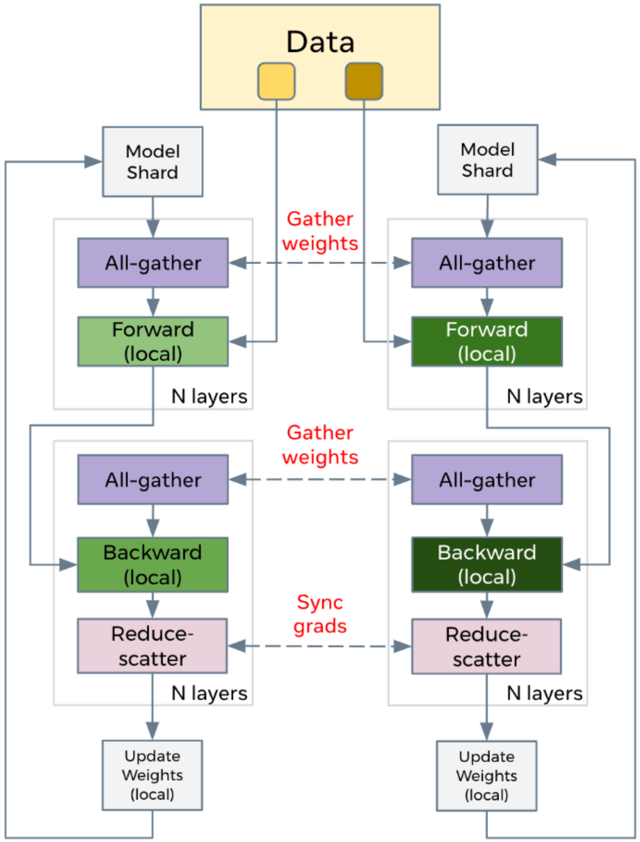


Training Accuracy
(16 GPUs) on Bridges-2 (NVIDIA V100)

Fully Sharded Data Parallel (FSDP)

- For Deep Learning training on modern GPU clusters
  - Model size has been increasing greatly (BERT, GPT, ...)
  - **Fully Sharded Data Parallel (FSDP)*** scheme has been introduced in PyTorch (v1.11) to shard the parameters, gradients, and optimizer states of the DL models amongst multiple GPUs.
  - Relies on the **Allgather** and **Reduce-Scatter** communication primitives to gather weights and sync up gradients.
  - Brings extra communication cost in training of large DNN models.

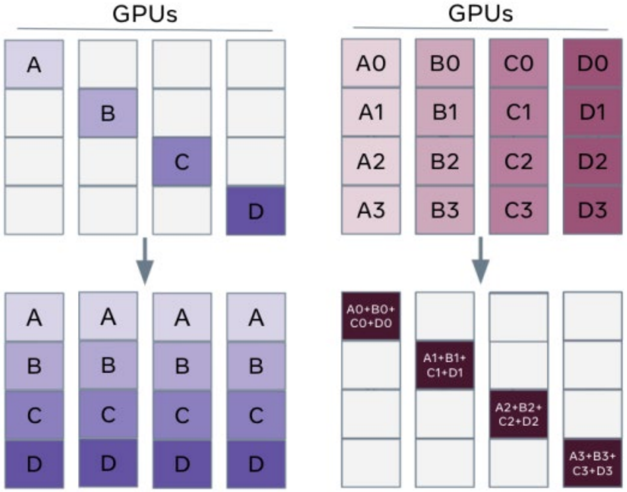
Fully Sharded Data Parallel Training



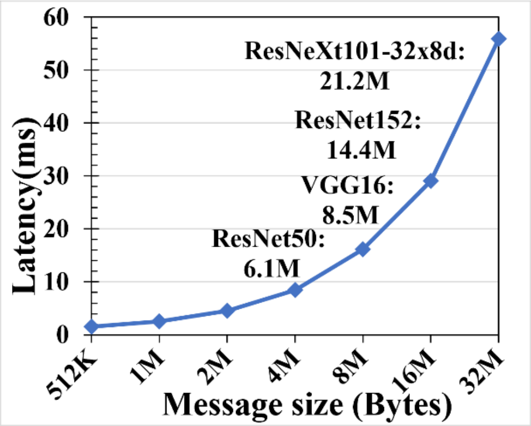
PyTorch, "Fully Sharded Data Parallel (FSDP)," https://pytorch.org/blog/introducing-pytorch-fully-sharded-data-parallel-api

Bottleneck of Allgather and Reduce-Scatter in FSDP

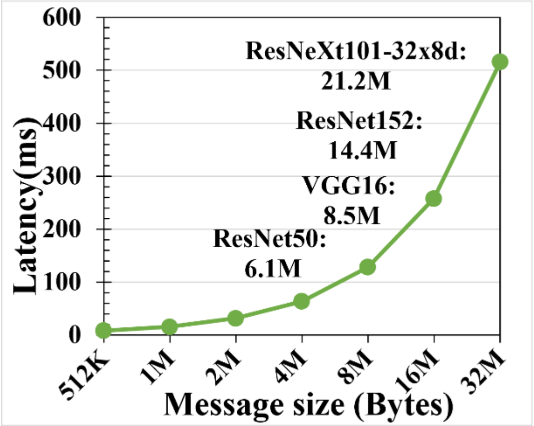
- Existing Allgather and Reduce-Scatter algorithms for transferring large GPU data suffer from poor performance due to the **limited interconnect bandwidth** between the GPUs.
- Allgather and Reduce-Scatter communication primitives add **large overheads** to the training of large models



(a) Allgather and Reduce-Scatter operations[2]



(b) Allgather latency with 16 V100 GPUs

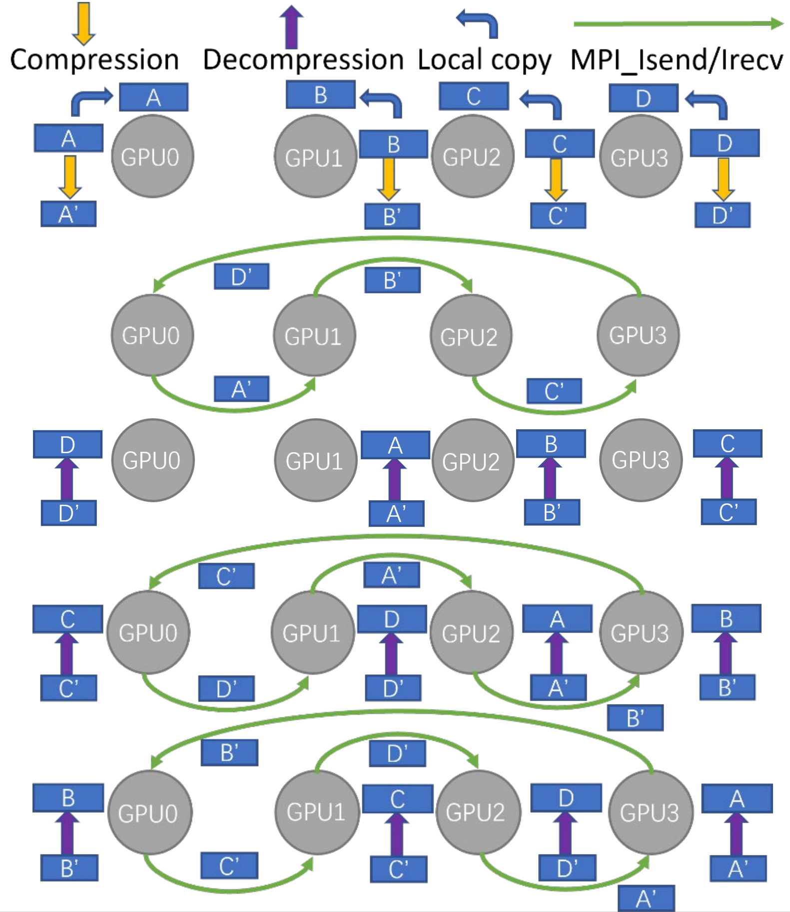


(c) Reduce-Scatter latency with 16 V100 GPUs

https://engineering.fb.com/wp-content/uploads/2021/07/FSDP-graph-2a.png?w=1024

Ring-based Allgather Communication with Collective-level Online Compression

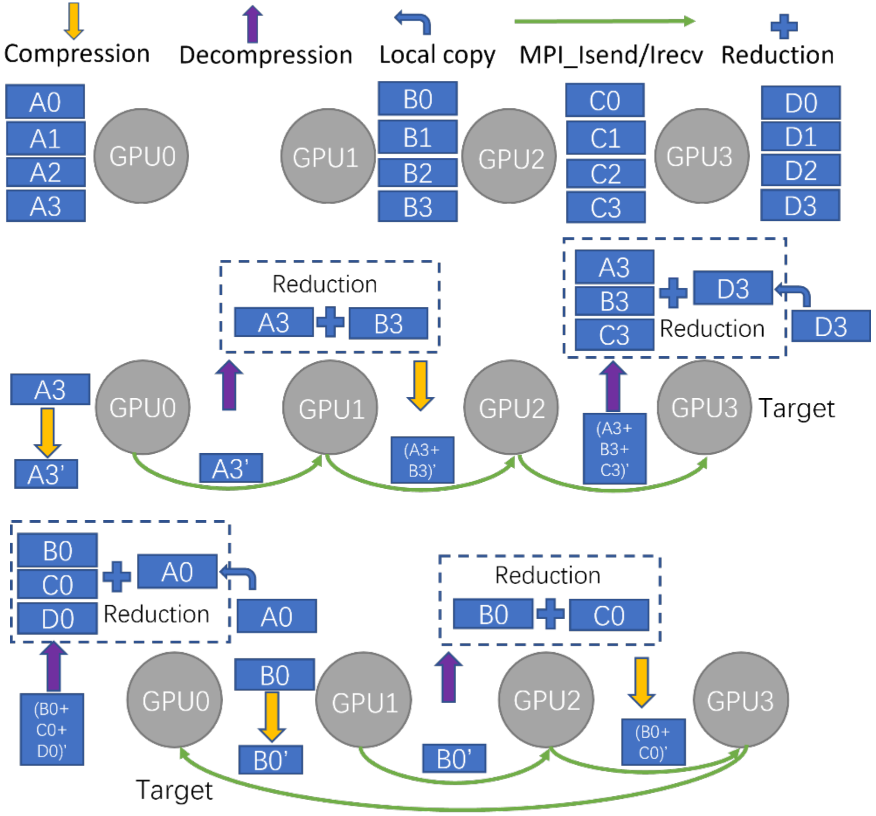
- Each GPU copies the its own data from send buffer to the receiver buffer directly
- Compression operation is only executed once
- MPI_Irecv is posted immediately after launching compression on non-default stream
- MPI_Isend is posted to send out the compressed data
- Decompression kernel is launched on a non-default CUDA stream to restore the data



Q. Zhou, Q. Anthony, L. Xu, A. Shafi, M. Abduljabbar, H. Subramoni, and D. K. Panda, Accelerating Distributed Deep Learning Training with Compression Assisted Allgather and Reduce-Scatter Communication, 37th IEEE International Parallel Distributed Processing Symposium (IPDPS '23), May 2023.

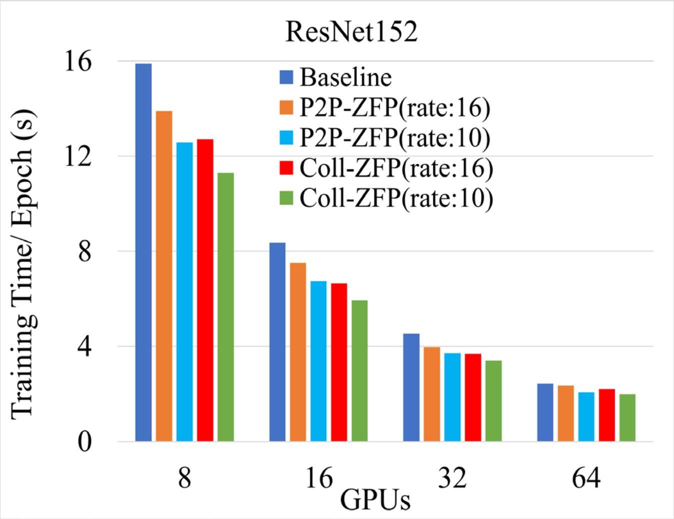
Ring-based Reduce-Scatter Communication with Collective-level Online Compression

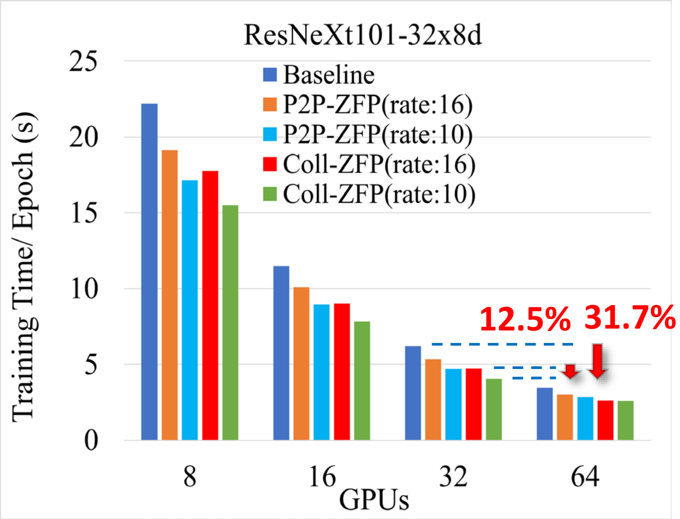
- Data elements on each GPU are scattered to all the corresponding GPUs
- Compression kernel is launched for data element or reduction result
- Launch reduction kernel on GPU to get the aggregated result
- MPI_Isend/Irecv transfer compressed data element or reduction result
- Decompression kernel is launched to restore data element or reduction result



Application-Level Evaluation (FSDP Training)

- PyTorch FSDP training performance
  - Use enhanced MPI backend with proposed compression designs for Allgather and Reduce-Scatter
  - Reduces training time by up to **31.7%** (32 GPUs, ZFP rate: 10) vs. Baseline
  - Reduces training time by up to **12.5%** (32 GPUs, ZFP rate: 10) vs. Point-to-Point compression ("P2P")

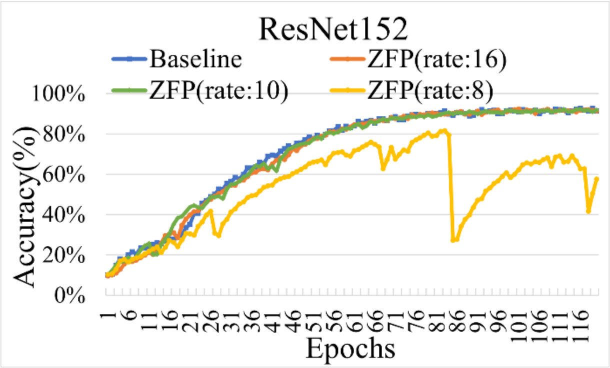


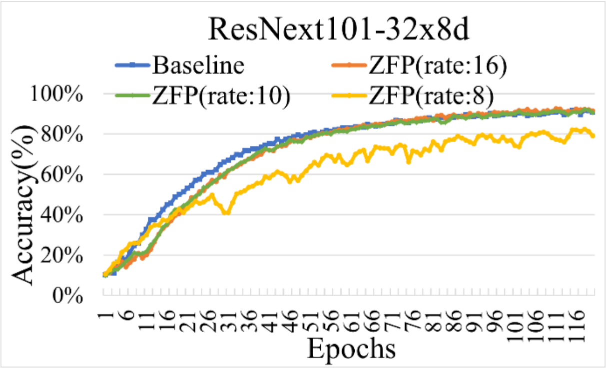


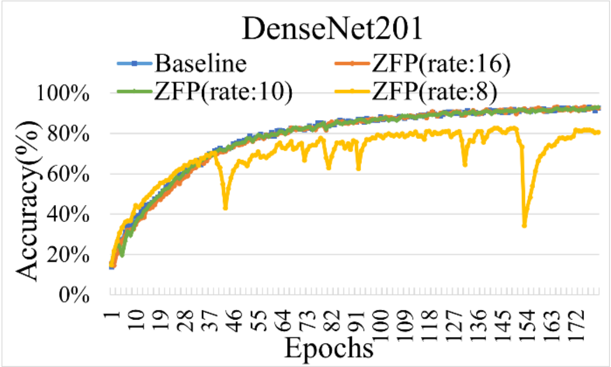
Cluster: Longhorn(NVIDIA V100), Dataset: CIFAR10, Batch Size=128, Learning Rate=0.001

Application-Level Evaluation (FSDP Training)

- PyTorch FSDP training accuracy
  - Use enhanced MPI backend with proposed compression designs for Allgather and Reduce-Scatter
  - Proposed design with ZFP compression (rate:16, rate:10) achieves similar convergent training accuracy vs. Baseline
  - Big accuracy drop and large variance with lower compression rate: 8 due to larger compression errors added to weights and gradients



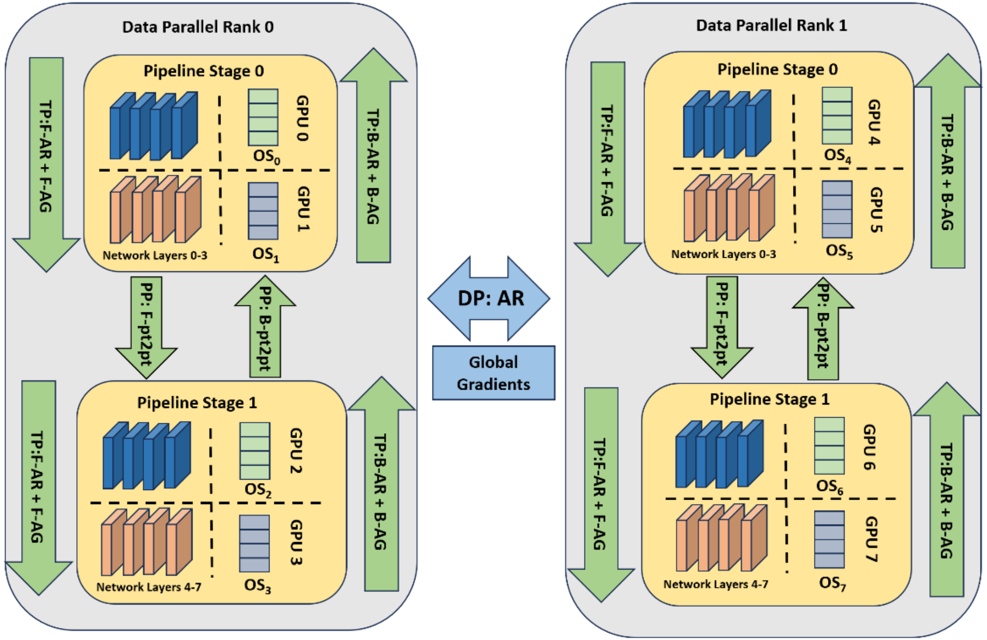




Cluster: Pitzer(NVIDIA V100), Dataset: CIFAR10, Batch Size=128, Learning Rate=0.001

LLM training with Hybrid Compression design

- For LLM training on modern GPU clusters
  - Model size exceeds memory capacity
  - **3D Parallelism** adopted with Megatron+DeepSpeed to efficiently perform training across thousands of GPUs.
    - **Data Parallelism** (Allreduce)
    - **Pipeline Parallelism** (Point-to-point)
    - **Tensor Parallelism** (Allgather + Allreduce)
    - **ZeRO** (Reduce-Scatter + Allgather)
  - Heavy communication saturating interconnect bandwidth

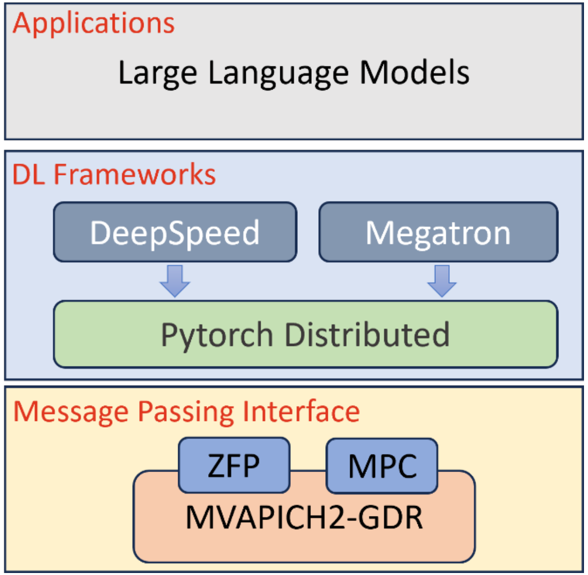


Hybrid Compression Solution

- Naïve ZFP or MPC solution poses different pros and cons
  - Lossy ZFP provides speedups but degradation in accuracy
  - Lossless MPC maintains baseline accuracy but degradation in throughput
- DP Gradients are sparse, MP activations are dense
  - Possible Hybrid solution for according parallelism degree

Experiment setup

CPU	IBM Power9 44 Cores/Node
Memory	256GB
GPU	NVIDIA Tesla V100 (32GB)
Interconnect	InfiniBand EDR 100GB/s

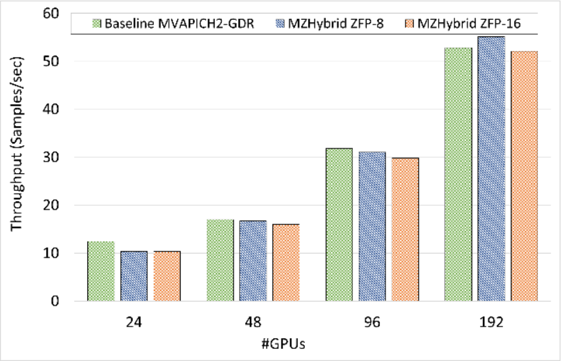


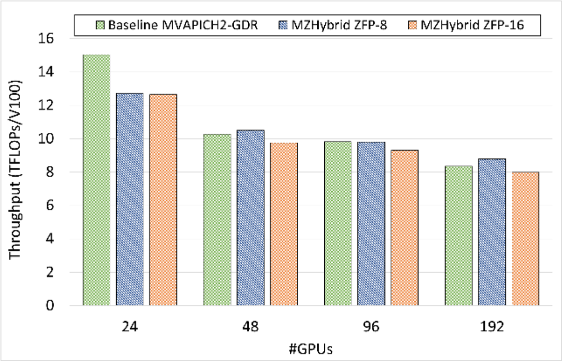
Lassen cluster setup

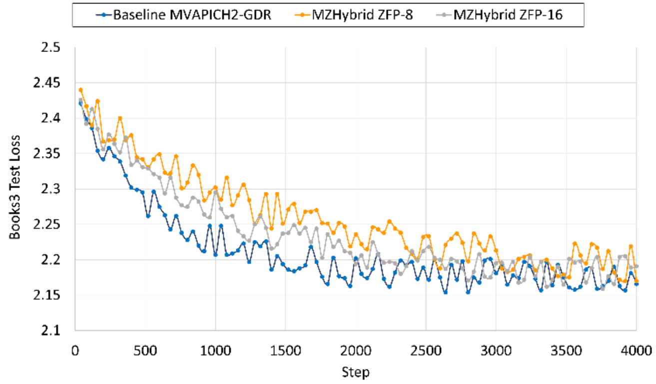
Model	**GPT-NeoX-20B**
Dataset	Books3
PP Degree	6
MP Degree	4
Grad Accumulation Step	1
Micro batch size per GPU	4

Hybrid Compression Solution (MZHybrid)

- lossy ZFP compression for Data Parallel gradient Allreduce + lossless MPC compression for Model Parallel (TP + PP) communication
- Good performance speedup (4.4% increase for samples/sec & 5.3% increase for TFLOPS), loss curves greatly improved



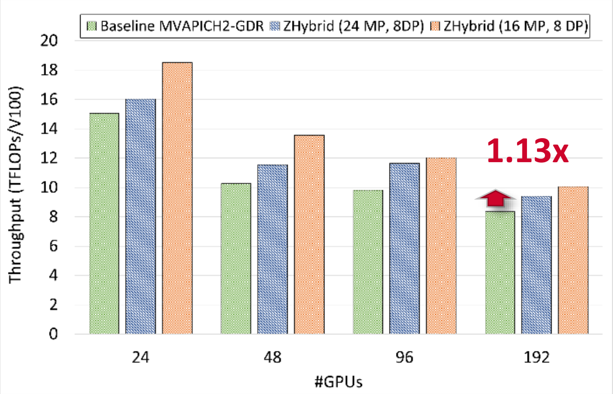


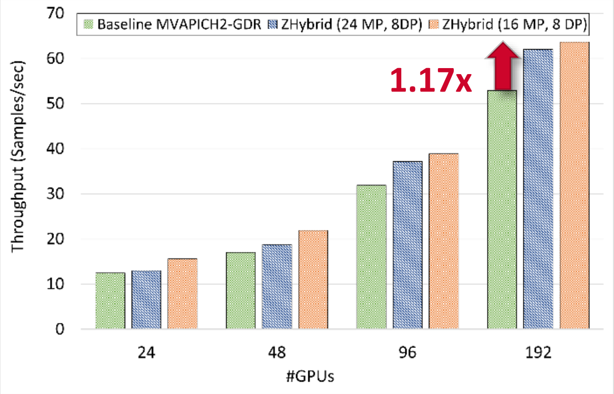


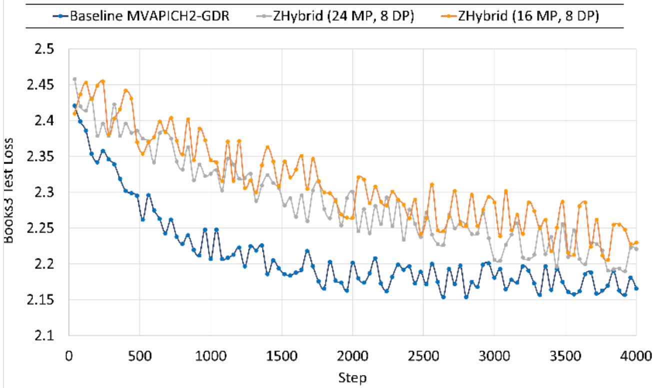
Cluster: Lassen (NVIDIA V100)

Hybrid Compression Solution (ZHybrid)

- Low-rate ZFP compression for Data Parallel gradient Allreduce + high-rate ZFP compression for Model Parallel (TP + PP) communication
- Even better performance speedup (17.3% increase for samples/sec & 12.7% increase for TFLOPS), loss curves still acceptable







Cluster: Lassen (NVIDIA V100)

Conclusion

- Integrated lossless(MPC) and lossy(ZFP) compression algorithms into MVAPICH2-GDR
- Implemented various compression designs for various communication operations
  - Proposed **Host-staging** based collective-level compression for **All-to-All** operation
  - Proposed **Chunked-Chain** based compression design for optimizing **Broadcast** communication
  - Proposed **Ring-based** compression design for optimizing **Allgather** and **Reduce-Scatter**
- Accelerating AI workloads in Deep Learning training
  - Reduced Alltoall communication time in DeepSpeed benchmark by up to **26.4%**
  - Reduced the PyTorch DDP training time by up to **15.0%**
  - Reduced the PyTorch FSDP training time by up to **31.7%**
  - Accelerated the training of LLMs like GPT-NeoX-20B by up to **17.3%**
- Future work
  - Study and incorporate more GPU-based compression algorithms (e.g., NVIDIA nvCOMP, etc.)
  - Extend our designs to other common collectives

Thank You!

subramoni.1@osu.edu

Follow us on

https://twitter.com/mvapich

Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

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/

Outline

- Motivation and Research Challenges
- Framework and data flow of GPU-based Point-to-Point On-the-fly compression
- Host-staging based collective-level compression for AlltoAll communication
- DDP training with Chunked-Chain based collective-level compression for Bcast communication
- FSDP training with Ring-based collective-level compression for Allgather and Reduce-Scatter
- LLM training with hybrid compression schemes
- Performance result: DeepSpeed Benchmark, DDP training, FSDP training, LLM training
- Conclusion & Future work