MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

HiDL
High-Performance Deep Learning

HiBD
High-Performance Big Data

Benefits of On-the-Fly Compression on GPU-to-GPU Communication for HPC and Data Science Applications

Presentation at MUG '21

Qinghua Zhou

Network Based Computing Laboratory (NBCL)

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

zhou.2595@osu.edu

Follow us on

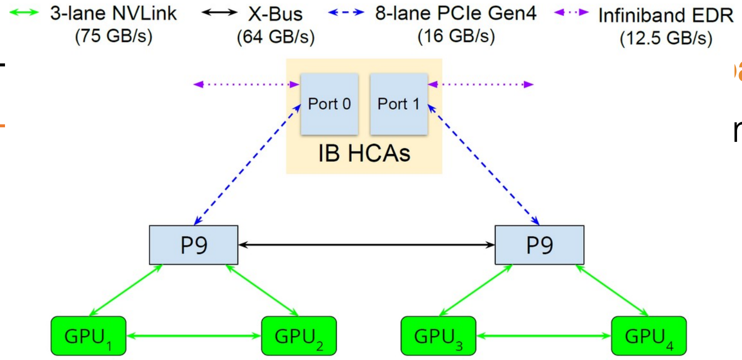
https://twitter.com/mvapich

Outline

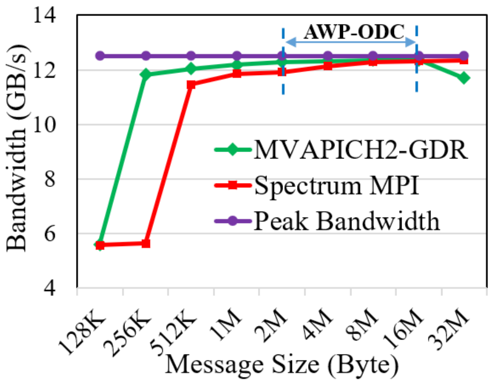
- Motivation
- Focus of the Work
- On-the-fly Compression Designs
  - Compression algorithms
  - Framework for GPU-based on-the-fly compression
- Performance Evaluation
- Conclusions and Future Work

Motivation

- For HPC and data science applications on modern GPU clusters
  - With larger problem sizes, applications exchange **orders of magnitude more data** on the network
  - Leads to significant **increase in communication times** for these applications on larger scale (AWP-ODC)
  - On modern HPC systems, there is **disparity** between intra-node and inter-node GPU communication bandwidths that prevents efficient scaling of applications on larger GPU
  - **andwidth**
  - nd lower th ited



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



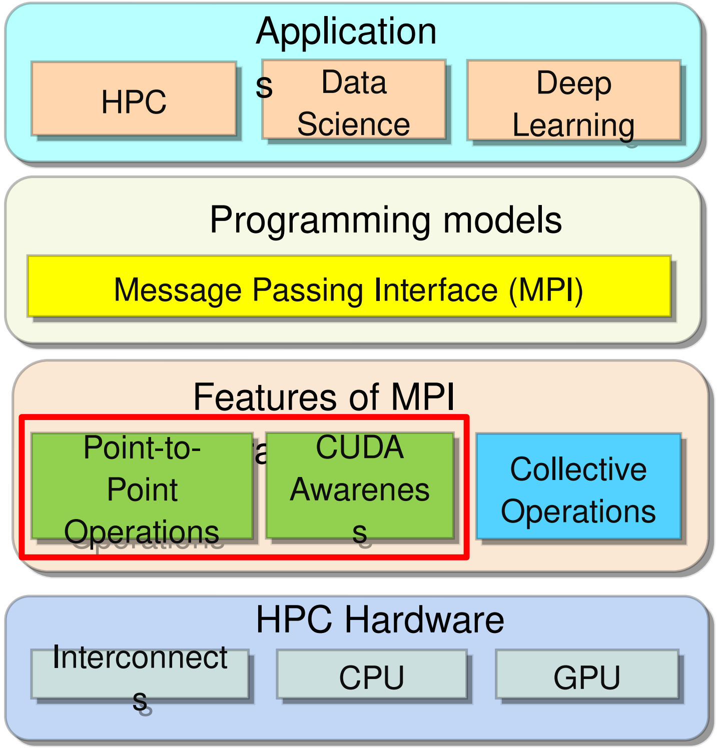
(b) Saturated bandwidth at large message size

[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.

Focus of the Work

- Designing **on-the-fly** message compression schemes in an MPI library:
  - The **first of its kind** GPU-based compression design is implemented using MVAPICH2-GDR
- Optimizing the existing GPU based compression algorithms
- Accelerating **point-to-point** communication performance of transferring large GPU-to-GPU data
- Demonstrating performance benefits for two categories of applications:
  - AWP-ODC (HPC) [2]
  - Dask (Data science) [3]

https://dask.org



[2] Y. Cui, K. B. Olsen, T. H. Jordan, K. Lee, J. Zhou, P. Small, D. Roten, G. Ely, D. K. Panda, A. Chourasia, J. Levesque, S. M. Day, and P. Maechling, “Scalable earthquake simulation on petascale supercomputers,” in SC ’10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, 2010, pp. 1–20.
[3] M. Rocklin, “Dask: Parallel computation with blocked algorithms and task scheduling,” in Proceedings of the 14th Python in Science Conference, K. Huff and J. Bergstra, Eds., 2015, pp. 130 – 136.

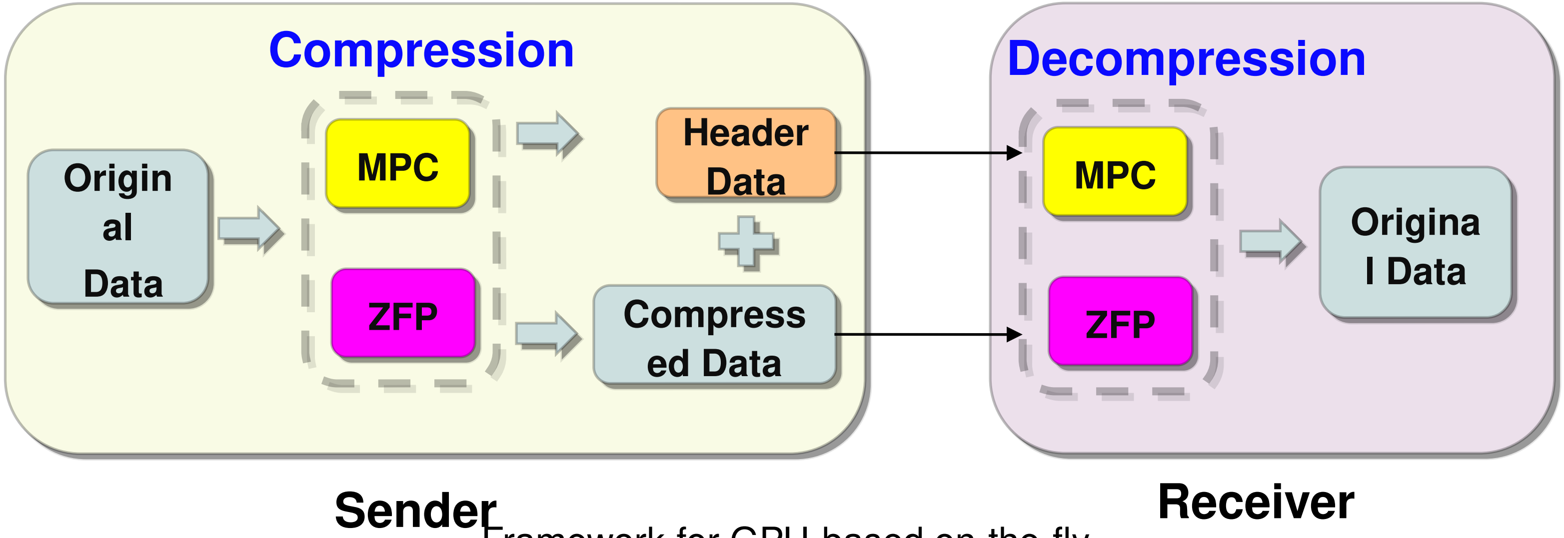
Compression Algorithms for Floating-point Data

- Lossless
  - Fpzip: CPU, supports double (64 bit) & single (32 bit) precision FP, low throughput
  - FPC: CPU, supports double & single precision FP, low throughput
  - ISOBAR: CPU, supports double & single precision FP, low throughput
  - GFC: GPU, supports double precision FP, high throughput
  - **MPC** [4]: **GPU**, supports double & single precision FP, **high throughput**
- Lossy
  - **ZFP** [5]: **GPU**, supports double & single precision FP, **high throughput**
  - SZ: GPU, supports double & single precision FP, high throughput

[4] A. Yang, H. Mukka, F. Hesaaraki, and M. Burtscher, "MPC: A Massively Parallel Compression Algorithm for Scientific Data," in IEEE Cluster Conference, September 2015.
[5] P. Lindstrom, "Fixed-rate compressed floating-point arrays," IEEE Transactions on Visualization and Computer Graphics, vol. 20, 08 2014.

Framework for GPU-based on-the-fly compression

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



Framework for GPU-based on-the-fly compression[6]

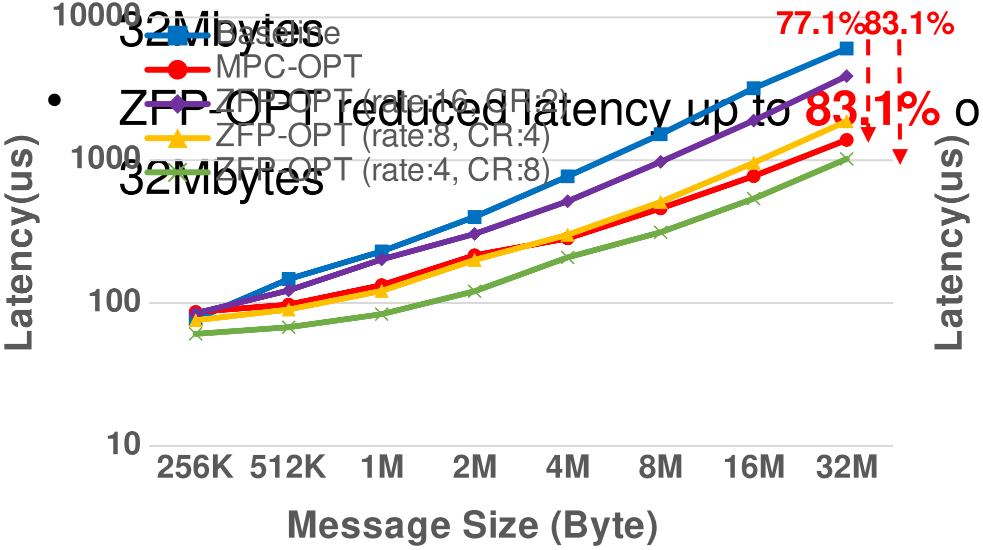
[6] Q. Zhou, C. Chu, N. Senthil Kumar, P. Kousha, M. Ghazimirsaeed, H. Subramoni, and D.K. Panda, "Designing High-Performance MPI Libraries with On-the-fly Compression for Modern GPU Clusters", 35th IEEE International Parallel & Distributed Processing Symposium (IPDPS21), May 2021. **[Best Paper Finalist]**

Experimental Environment

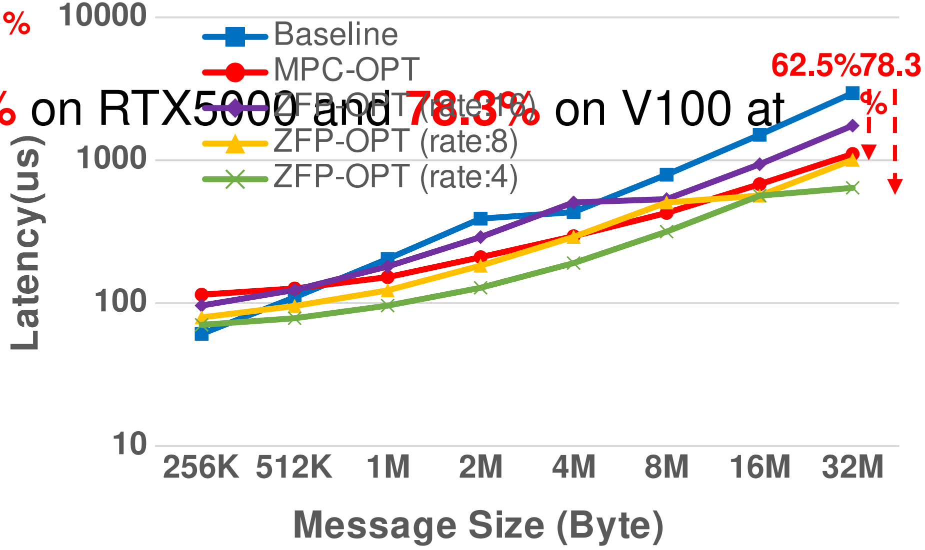
Cluster Specs	Frontera Longhorn	Frontera Liquid	Lassen
CPU Processor	Dual-socket IBM POWER9 AC922 2.3GHz, 20 Cores/socket	Dual-socket Intel Xeon E5-2620 2.10GHz, 8 Cores/socket	Dual-sock IBM POWER9 AC922 3.14GHz, 44 Cores/Socket
System Memory	256 GB	384 GB	256 GB
GPU Processor	4 NVIDIA Tesla V100	4 NVIDIA Quadro RTX 5000	4 NVIDIA Tesla V100
GPU Memory	4 x 16 GB	4 x 16 GB	4 x 16 GB
Interconnects between CPU and GPU	NVLink-2 (one-way 75 GB/s)	PCIe Gen3 x16 and x64 switches (one-way 16 GB/s)	NVLink-2 (one-way 75 GB/s)
Interconnects between GPUs	NVLink-2 (one-way 75 GB/s)	PCIe Gen3 x16 and x64 switches (one-way 16 GB/s)	NVLink-2 (one-way 75 GB/s)
Interconnects between nodes	Mellanox InfiniBand EDR (one-way 12.5 GB/s)	Mellanox InfiniBand FDR (one-way 7 GB/s)	Dual-rail Mellanox InfiniBand EDR (one-way 25 GB/s)
Operating System	RHEL 7.6 (4.14.0-115.10.1.1)	CentOS 7.6.1810 (3.10.0-957.27.2.el7)	RHEL 7.3 (4.14.0-115.10.1.1)
NVIDIA Driver Version	440.33.01	430.40	418.87.00
CUDA Toolkit Version	10.1.168	10.1.243	10.1.243

Inter-node GPU-GPU latency

- OSU Micro Benchmark for MPC-OPT and ZFP-OPT on RTX5000 and V100 GPU nodes
  - MPC-OPT and ZFP-OPT are optimized and integrated into MVAPICH2-GDR library
- MPC-OPT reduced latency up to **77.1%** on RTX5000 and **62.5%** on V100 at 32Mbytes
- ZFP-OPT reduced latency up to **83.1%** on RTX5000 and **78.3%** on V100 at 32Mbytes



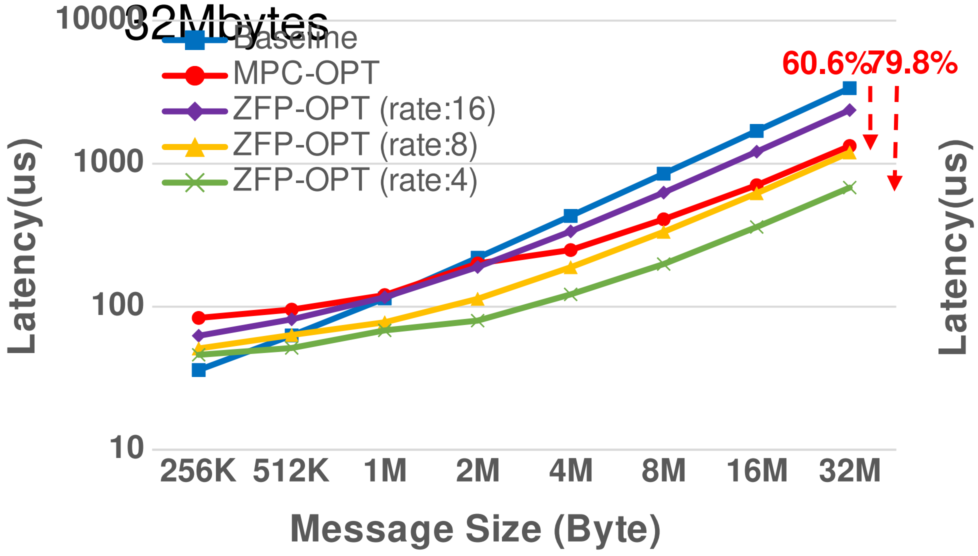
Frontra-RTX5000, Inter-node D-D latency



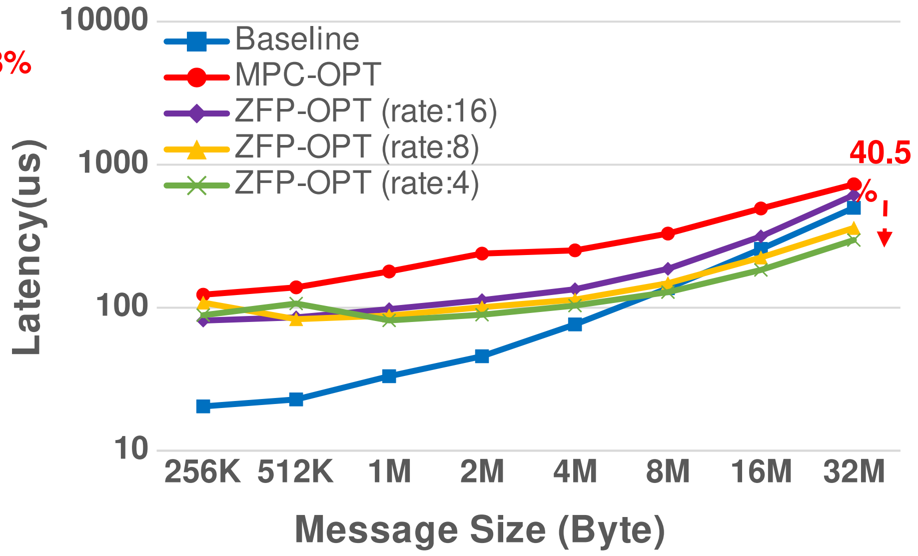
Longhorn-V100, Inter-node D-D latency

Intra-node GPU-GPU latency

- MPC-OPT reduced latency up to **60.6%** on RTX5000 at 32Mbytes
  - High-speed NVLink on Longhorn is faster than the MPC compression/decompression
- ZFP-OPT reduced latency up to **79.8%** on RTX5000 and **40.5%** on V100 at 32Mbytes



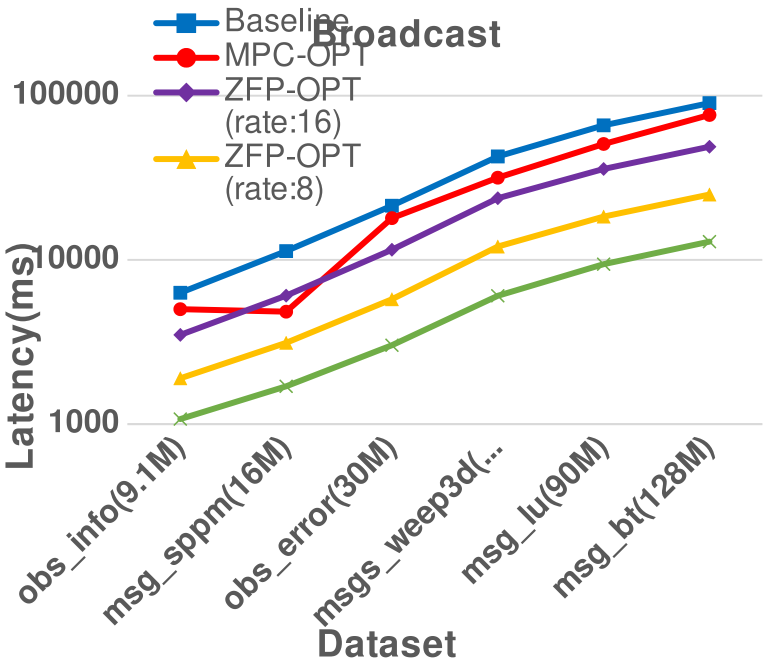
Frontra-RTX5000, Intra-node D-D latency

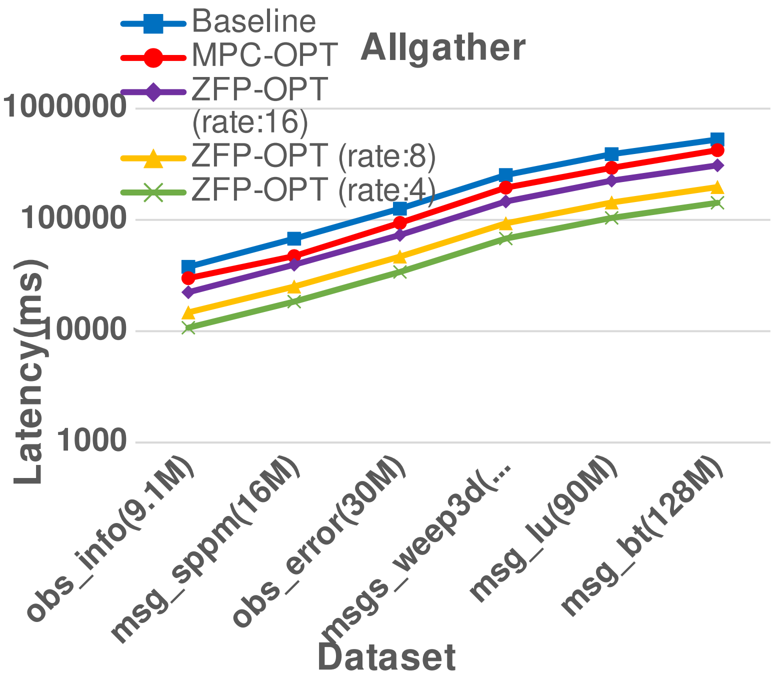


Longhorn-V100, Intra-node D-D latency

Collective Operations

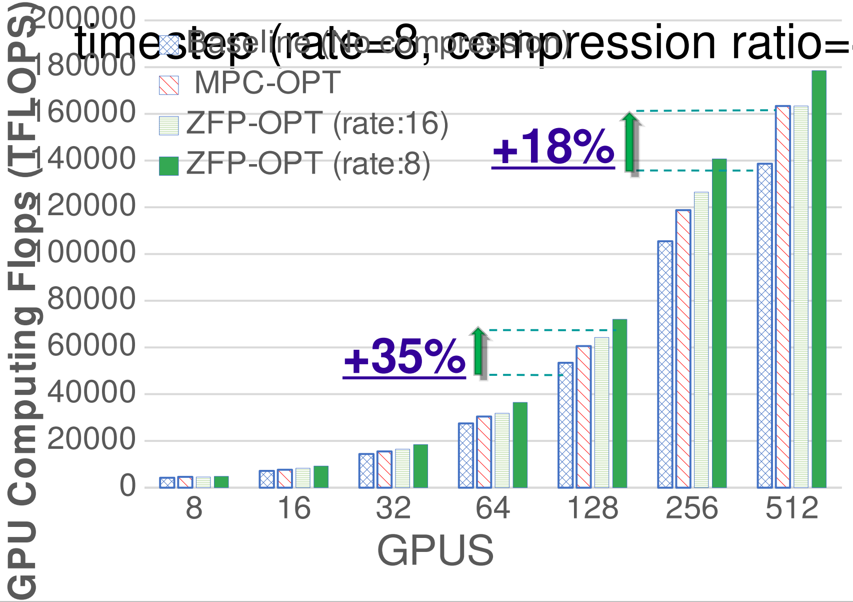
- MPI_Bcast MPC-OPT: **57%** benefits on msg_sppm, ZFP-OPT(rate:4): **85%** benefits
- MPI_Allgather MPC-OPT: **30%** benefits on msg_sppm, ZFP-OPT(rate:4): **73%** benefits

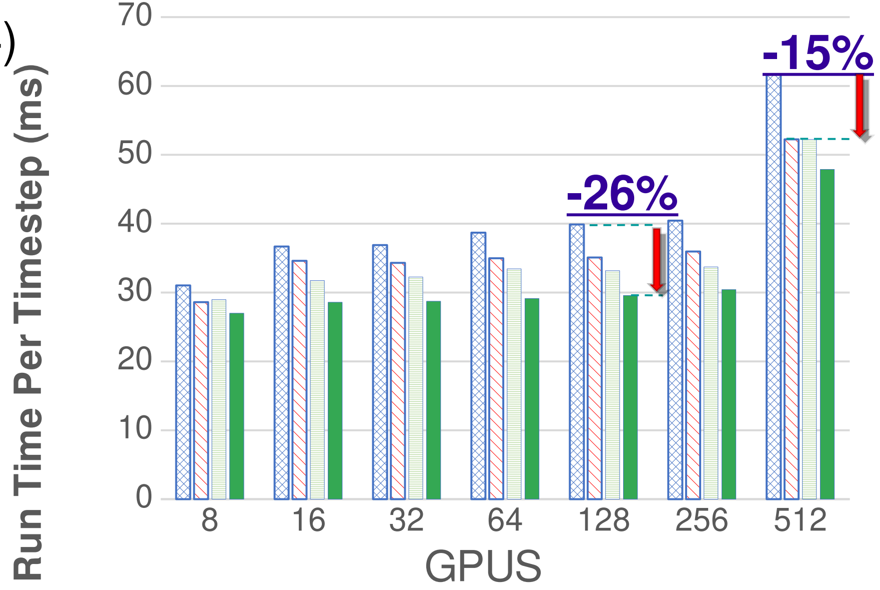




Application Results (AWP-ODC)

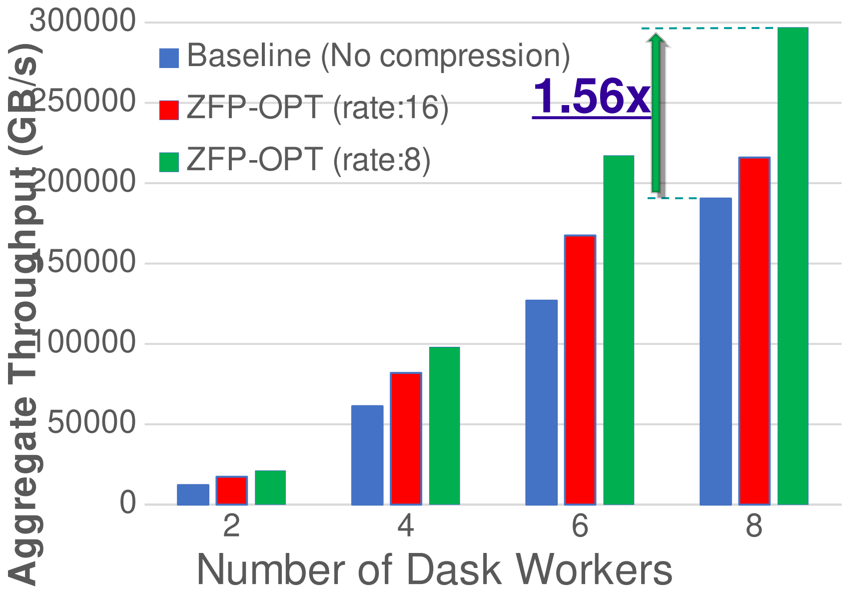
- Weak-Scaling of HPC application **AWP-ODC** on Lassen cluster (V100 nodes)
- MPC-OPT achieves up to **+18%** GPU computing flops, **-15%** runtime per timestep
- ZFP-OPT achieves up to **+35%** GPU computing flops, **-26%** runtime per timestep (rate=8, compression ratio=4)

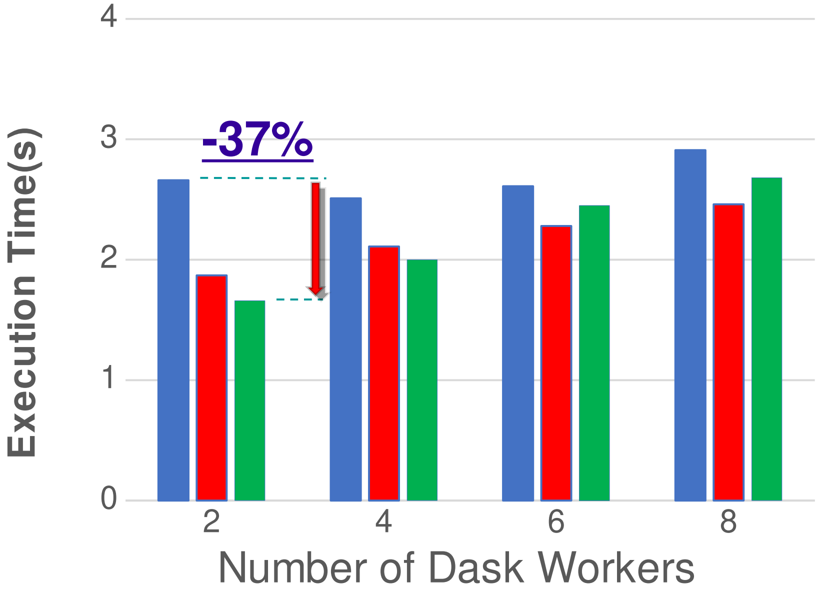




Application Results (Dask)

- Data science framework **Dask** on RI2 cluster (V100 nodes)
- Dask benchmark creates cuPy array and distributes its chunks across Dask workers
- ZFP-OPT achieves up to 1.56x throughput, -37% runtime (rate=8, compression ratio=4)





(cuPy Dims: 10Kx10K, Chunk size: 1K)

Conclusions and Future Work

- Presented on-the-fly compression techniques for optimizing GPU-to-GPU communication in an MPI library
- Enhanced GPU based pt2pt communication in MVAPICH2-GDR with optimized MPC and ZFP
- Benchmark-level Benefits (GPU-GPU Latency)

Compression	Inter-node	Intra-node	MPI_Bcast	MPI_Allgather
MPC-OPT	77.1%	60.6%	57%	30%
ZFP-OPT	83.1%	79.8%	85%	73%

- Application-level Benefits
  - AWP-ODC: up to 18% (MPC-OPT on 512 GPUs) and 35% (ZFP-OPT on 128 GPUs) improvement of GPU computing flops
  - Dask: up to 1.56x speedup of throughput and 37% reduced runtime
- Future work
  - Study and incorporate more GPU-based compression algorithms (e.g. cuSZ, NVIDIA nvCOMP, etc.)

Thank You!

Zhou.2595@osu.edu

Network Based Computing Laboratory

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/