POSTER: Optimizing Collective Communications with Error-bounded Lossy Compression for GPU Clusters

Jiajun Huang
jhuan380@ucr.edu
University of California, Riverside
Riverside, United States of America

Sheng Di
sdi1@anl.gov
Argonne National Laboratory
Lemont, United States of America

Xiaodong Yu
xyu38@stevens.edu
Stevens Institute of Technology
Hoboken, United States of America

Yujia Zhai
yzhai015@ucr.edu
University of California, Riverside
Riverside, United States of America

Jinyang Liu
jliu447@ucr.edu
University of California, Riverside
Riverside, United States of America

Yafan Huang
yafan-huang@uiowa.edu
University of Iowa
Iowa City, United States of America

Ken Raffenetti
raffenet@anl.gov
Argonne National Laboratory
Lemont, United States of America

Hui Zhou
zhouh@anl.gov
Argonne National Laboratory
Lemont, United States of America

Kai Zhao
kzhao@cs.fsu.edu
Florida State University
Tallahassee, United States of America

Zizhong Chen
chen@cs.ucr.edu
University of California, Riverside
Riverside, United States of America

Franck Cappello
cappello@mcs.anl.gov
Argonne National Laboratory
Lemont, United States of America

Yanfei Guo
yguo@anl.gov
Argonne National Laboratory
Lemont, United States of America

Rajeev Thakur
thakur@anl.gov
Argonne National Laboratory
Lemont, United States of America

Abstract

GPU-aware collective communication has become a major bottleneck for modern computing platforms as GPU computing power rapidly rises. To address this issue, traditional approaches integrate lossy compression directly into GPU-aware collectives, which still suffer from serious issues such as underutilized GPU devices and uncontrolled data distortion. In this paper, we propose *GPU-LCC*, a general framework that designs and optimizes GPU-aware, compression-enabled collectives with well-controlled error propagation. To validate our framework, we evaluate the performance on up to 64 NVIDIA A100 GPUs with real-world applications and datasets. Experimental results demonstrate that our *GPU-LCC*-accelerated collective computation (Allreduce), can outperform NCCL as well as Cray MPI by up to 3.4× and 18.7×, respectively. Furthermore, our accuracy evaluation with an image-stacking application confirms the high reconstructed data quality of our accuracy-aware framework.

PPoPP '24, March 2–6, 2024, Edinburgh, United Kingdom

ACM ISBN 979-8-4007-0435-2/24/03
https://doi.org/10.1145/3627535.3638467

CCS Concepts: • **Computing methodologies** → **Distributed algorithms**; **Parallel algorithms**; • **General and reference** → **Performance**.

Keywords: GPU, Collective Communication, Compression

1 Introduction

For GPU-aware collective communication, numerous researchers are actively working on mitigating network congestion in large-message collectives. In fact, network saturation is often the major bottleneck because of limited network bandwidth. For example, even with advanced networks, such as HPE Slingshot 10, the network bandwidth is only about 100 Gbps. A straightforward solution is designing large-message algorithms that can minimize the transferred data volume instead of latency [1, 4, 5, 9, 11]. Another promising solution is shrinking the message size by error-bounded lossy compression techniques [2, 8, 10, 12], as it can significantly reduce the data volume and maintain the data quality.

Previous lossy-compression-integrated approaches can be divided into two categories. The first is *compression-enabled point-to-point communication* (namely CPRP2P) [13], which directly uses the 1D fixed-rate ZFP [8] to compress the data before it is sent and decompresses the received data after it is received. This method may cause significant overheads and unbounded errors in the collective communications as shown in [3]. The other category is to particularly optimize the *compression-enabled collectives*. Huang et al. designed an optimized general framework for compression-enabled

collectives that can realize high performance for all MPI collectives with controlled errors [3]. Nevertheless, this approach suffers from suboptimal performance on modern GPU clusters because of under-utilized GPU devices.

To address the aforementioned limitations, we design a generic framework for GPU-aware compression-accelerated collective communications that can realize both high performance and controlled error propagation.

2 *GPU-LCC* Design and Optimization

In this section, we present our design and optimization strategies as shown in Figure 1. To be specific, we analyze the problems of prior solutions and do a comprehensive performance breakdown to identify potential bottlenecks. Additionally, we also characterize the performance of the lossy compressor and find that the direct application of ring-based algorithms for collective computation with GPU compression may not always yield optimal results. It is hence vital to explore other algorithms that may offer superior performance. After that, we propose the *GPU-LCC* framework to address and overcome the performance issues noted in the previous GPU-aware MPI collective framework that incorporates compression, such that a superior performance can be reached. Our contributions are 5-fold: (1) To circumvent the high cost of device-to-host data transfer inherent in traditional CPU-centric designs, we implement a GPU-centric design. (2) To improve collective performance in compression-enabled collectives, we adapt the lossy compression to suit the requirements of collective communications. (3) We explore new metrics regarding GPU compression-enabled collective performance, focusing on minimizing total compression cost and accuracy loss. (4) We propose two algorithm design frameworks for both collective computation and collective data movement to increase device utilization, decrease times of compression/decompression, and maximize the performance. (5) Furthermore, we improve the error-bounded lossy compressor (cuSZp[6]) and develop a multi-stream version to suit the context of the two collective performance optimization frameworks. In our performance optimization frameworks, we try to let as many operations as possible overlap with each other, including kernel launching, compression/decompression operation, and data movement.

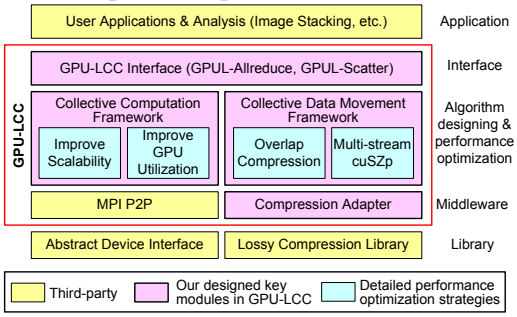

Figure 1. *GPU-LCC* design architecture.

3 Experimental Evaluation

We present and discuss the evaluation results as follows.

3.1 Experimental Setup

We perform the evaluation on a GPU supercomputer that involves 64 NVIDIA A100 80G GPUs with 4 GPUs per node, interconnected with a bandwidth of 100 Gbps. Two distinct RTM datasets [7], originating from the real-world 3D SEG/EAGE Overthrust model, are generated under two different simulation settings.

Evaluation with different message sizes. We evaluate the performance of our GPUL-Allreduce algorithm using various data sizes up to 600 MB on a configuration of 64 NVIDIA A100 GPUs across 16 nodes. As observed in Figure 2, our recursive doubling-based GPUL-Allreduce (ReDoub) consistently outperforms across all data sizes, achieving up to a speedup of 18.7× compared to Cray MPI and a 3.4× performance improvement over NCCL. Furthermore, with increasing data sizes, the speedup generally rises, demonstrating high scalability with respect to data size.

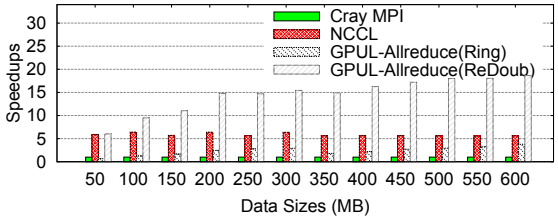

Figure 2. Performance evaluation of our GPUL-Allreduce with Cray MPI and NCCL in different data sizes.

4 Conclusion

This paper presents *GPU-LCC*, an innovative framework that optimizes GPU-aware collective communications, which can obtain 18.7× and 3.4× speedups over Cray MPI and NCCL on a testbed of 64 NVIDIA A100 GPUs.

Acknowledgment

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations – the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nation's exascale computing imperative. The material was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357, and supported by the National Science Foundation under Grant OAC-2003709, OAC-2104023, and OAC-2311875. We acknowledge the computing resources on Polaris (operated by Argonne Leadership Computing Facility).

References

[1] George Almási, Philip Heidelberger, Charles J. Archer, Xavier Martorell, C. Chris Erway, José E. Moreira, B. Steinmacher-Burow, and Yili Zheng. 2005. Optimization of MPI Collective Communication on BlueGene/L Systems. In *Proceedings of the 19th Annual International Conference on Supercomputing* (Cambridge, Massachusetts) *(ICS '05)*. Association for Computing Machinery, New York, NY, USA, 253–262. https://doi.org/10.1145/1088149.1088183

[2] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 730–739.

[3] Jiajun Huang, Sheng Di, Xiaodong Yu, Yujia Zhai, Zhaorui Zhang, Jinyang Liu, Xiaoyi Lu, Ken Raffenetti, Hui Zhou, Kai Zhao, Zizhong Chen, Franck Cappello, Yanfei Guo, and Rajeev Thakur. 2023. An Optimized Error-controlled MPI Collective Framework Integrated with Lossy Compression. arXiv:2304.03890 [cs.DC]

[4] Jiajun Huang, Kaiming Ouyang, Yujia Zhai, Jinyang Liu, Min Si, Ken Raffenetti, Hui Zhou, Atsushi Hori, Zizhong Chen, Yanfei Guo, and Rajeev Thakur. 2023. Accelerating MPI Collectives with Process-in-Process-Based Multi-Object Techniques. In *Proceedings of the 32nd International Symposium on High-Performance Parallel and Distributed Computing* (Orlando, FL, USA) *(HPDC '23)*. Association for Computing Machinery, New York, NY, USA, 333–334. https://doi.org/10.1145/3588195.3595955

[5] Jiajun Huang, Kaiming Ouyang, Yujia Zhai, Jinyang Liu, Min Si, Ken Raffenetti, Hui Zhou, Atsushi Hori, Zizhong Chen, Yanfei Guo, and Rajeev Thakur. 2023. PiP-MColl: Process-in-Process-based Multi-object MPI Collectives. In *2023 IEEE International Conference on Cluster Computing (CLUSTER)*. 354–364. https://doi.org/10.1109/CLUSTER52292.2023.00037

[6] Yafan Huang, Sheng Di, Xiaodong Yu, Guanpeng Li, and Franck Cappello. 2023. cuSZp: An Ultra-fast GPU Error-bounded Lossy Compression Framework with Optimized End-to-End Performance. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–13.

[7] Suha Kayum et al. 2020. GeoDRIVE – A high performance computing flexible platform for seismic applications. *First Break* 38, 2 (2020), 97–100.

[8] Peter Lindstrom. 2014. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics* 20 (2014), 2674–2683.

[9] Pitch Patarasuk and Xin Yuan. 2009. Bandwidth optimal all-reduce algorithms for clusters of workstations. *J. Parallel and Distrib. Comput.* 69, 2 (2009), 117–124.

[10] Dingwen Tao, Sheng Di, and Franck Cappello. 2017. Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization. https://doi.org/10.1109/IPDPS.2017.115

[11] Rajeev Thakur, Rolf Rabenseifner, and William Gropp. 2005. Optimization of collective communication operations in MPICH. *The International Journal of High Performance Computing Applications* 19, 1 (2005), 49–66.

[12] Kai Zhao, Sheng Di, Xin Liang, Sihuan Li, Dingwen Tao, Zizhong Chen, and Franck Cappello. 2020. Significantly improving lossy compression for HPC datasets with second-order prediction and parameter optimization. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*. 89–100.

[13] Q. Zhou, C. Chu, N. S. Kumar, P. Kousha, S. M. Ghazimirsaeed, H. Subramoni, and D. K. Panda. 2021. Designing High-Performance MPI Libraries with On-the-fly Compression for Modern GPU Clusters. In *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 444–453. https://doi.org/10.1109/IPDPS49936.2021.00053