HV-Coll: Co-Designing Collective Communication with Vectorized Homomorphic Compression

Jiefeng Zhou
jiefeng@usf.edu
University of South Florida
Tampa, FL, USA

Sheng Di (advisor)
sdi1@anl.gov
Argonne National Laboratory
Lemont, IL, USA

Yanfei Guo (advisor)
yanfeig@nvidia.com
NVIDIA Corporation
Santa Clara, CA, USA

Lingqi Zhang (advisor)
lingqi.zhang@riken.jp
RIKEN Center for Computational Science
Kobe, Japan

Kai Zhao (advisor)
kai.zhao@fsu.edu
Florida State University
Tallahassee, FL, USA

Rajeev Thakur (advisor)
thakur@anl.gov
Argonne National Laboratory
Lemont, IL, USA

Franck Cappello (advisor)
cappello@mcs.anl.gov
Argonne National Laboratory
Lemont, IL, USA

Jiajun Huang (advisor)
jiajunhuang@usf.edu
University of South Florida
Tampa, FL, USA

Abstract

Inter-node collective communication constrains large-scale scientific simulations and distributed AI. Error-bounded lossy compression can reduce data volume, but prior CPU frameworks underuse memory bandwidth with scalar packing, lose compression ratio (CR) at SIMD-friendly block sizes, and repeat decompress–operate–compress (DOC) work during reductions. We present **HV-Coll**, which co-designs a block-aligned representation, SIMD encoding, outlier-aware fixed-length encoding, compressed-domain addition, and MPI collective schedules. Evaluated on 3,584 CPU cores, HV-Coll's homomorphic compression is up to 8.1× faster than hZCCL's scalar DOC workflow, and its outlier-aware encoding improves CR by up to 27.2%. HV-Coll collectives reach up to 47.2×, 23.4×, 43.8×, and 67.6× speedups over native MPI for Bcast, Scatter, Allreduce, and Alltoall, respectively.

Keywords

Collective Communication, Data Compression

1 Introduction

Inter-node communication limits the scale of scientific simulations and distributed AI [6]. Error-bounded compression reduces transferred bytes but adds processing cost. hZCCL already avoids repeated decompress–operate–compress (DOC) work in reductions through homomorphic compression [3, 4].

Existing CPU implementations leave compressed formats misaligned with collective dataflow. Scalar fixed-length packing hinders SIMD execution, and one outlier widens the entire block and lowers compression ratio (CR) when larger blocks are needed for vector efficiency. Prior homomorphic designs also concentrate on reductions; the communication-heavy MPI Alltoall operation lacks a CPU-centric homomorphic design [2, 13]. HV-Coll shares an encoded block across compression and collectives, with a SIMD path, an outlier path for CR recovery, and schedules that reuse payloads or use native MPI when compression is unprofitable.

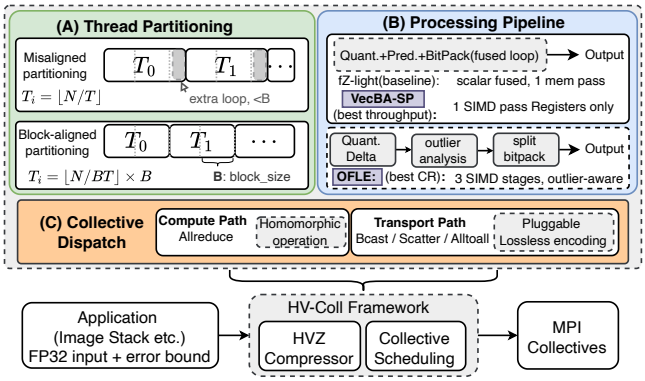


Figure 1: High-level HV-Coll overview. A shared block-aligned format connects SIMD encoding, homomorphic reduction, and MPI transport.

2 HV-Coll Co-design Overview

HV-Coll shares this format across its compressor, homomorphic operator, and scheduler. **Panel A** in Figure 1 aligns thread partitions to block boundaries, and block headers store metadata and payload offsets for parallel consumption. **Panel B** presents two compressor modes. VecBA-SP fuses SIMD quantization, delta generation, sign handling, and packing into a register-resident path. OFLE separates normal and outlier streams, preventing exceptional values from widening the bit width of an entire block and lowering compression ratio. Finally, **Panel C** shows how the format determines each collective's execution.

For Allreduce, quantized integer blocks are reduced exactly in the encoded domain, introducing no additional rounding error. Following C-Coll's normal-error model, the accumulated error for a single element reduced across P ranks lies approximately within $\pm 2\sqrt{P}\sigma$ with probability 95.44%, where σ is the standard deviation of the per-rank compression error [4]. Lazy compression prepares only the first ring segments, overlaps later compression with forwarding, and reuses the reduced payload in allgather. Bcast and Scatter send compressed threadblocks through a binomial tree. For

Alltoall, we use two buffers to overlap sends and next-segment compression. This design follows established MPI principles for payload reuse and compute–communication overlap [9, 11].

3 Experimental Evaluation

We compare HV-Coll with native MPI and the hZCCL baseline [3], whose scalar-fused compressor we denote fZ-light, on 3,584 cores of 32 Sapphire Rapids (SPR) nodes on the Stampede3 cluster [10]. Datasets come from SDRBench [12], Open-SciVis [5], and Qwen3.5 embeddings [7, 8], eight datasets in total.

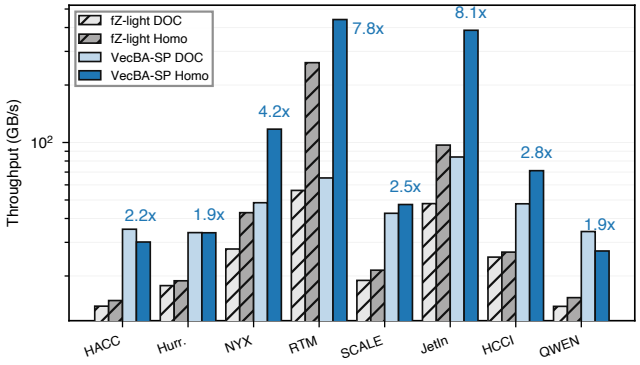


Figure 2: Homomorphic-compression throughput at B=128 and ε_{rel}=10^{-4}.

Table 1: OFLE CR gain over fZ-light's average CR at B=128.

ε_{rel}	HACC	Hurr.	NYX	RTM	SCALE	JetIn	HCCI	QWEN
10^{-2}	+2.7%	+8.6%	+0.2%	+1.7%	+2.1%	+0.0%	+0.0%	+0.2%
10^{-3}	+2.9%	+26.7%	+1.3%	+6.8%	+7.5%	+3.5%	+5.4%	+1.1%
10^{-4}	**+12.8%**	**+27.2%**	**+4.9%**	**+7.0%**	**+12.1%**	**+9.4%**	**+10.4%**	**+4.5%**

Figure 2 shows that the homomorphic workflow of VecBA-SP is 1.9–8.1× faster than the fZ-light DOC workflow. Table 1 shows that OFLE preserves or improves CR across all eight datasets, with gains up to 27.2%. Figure 3 evaluates Allreduce and Alltoall across message sizes. Because hZCCL lacks Alltoall, the hZCCL-compressor baseline runs its compressor on HV-Coll's baseline Alltoall path. At 112 threads, OFLE's geometric-mean speedups are 30.5× for Allreduce and 27.1× for Alltoall; its peaks are 43.8× and 67.6×, respectively. At one thread, both HV-Coll Alltoall variants outperform the hZCCL-compressor baseline at five of the six message sizes. Allreduce is mixed: hZCCL leads at 100–200 MiB, and an HV-Coll variant leads at every size from 300 MiB.

For Bcast and Scatter (Table 2), HV-Coll's best-per-size full-node speedups over native MPI have geometric means of 30.6× for Bcast and 8.6× for Scatter, with peaks of 47.2× and 23.4×, respectively. At 100 MiB, the 99.9× single-thread Bcast speedup corresponds to minimum latencies of 3.742 s for native MPI and 37.47 ms for HV-Coll, which is also 1.11× faster than hZCCL there.

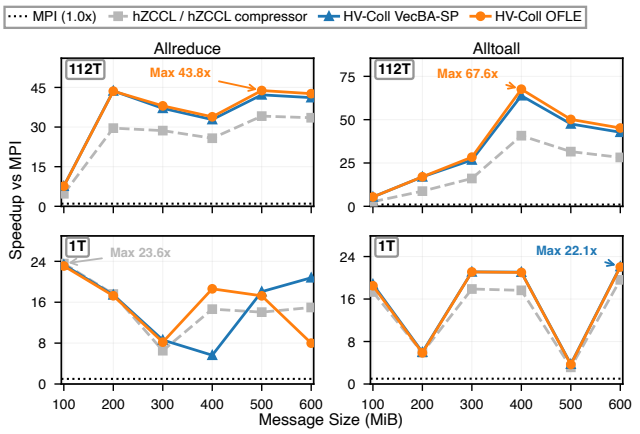


Figure 3: End-to-end speedup on RTM at 32 SPR nodes with B=128 and ε_{rel}=10^{-4}. Top: 112 threads; bottom: one thread; dotted line: native MPI (1×).

Table 2: HV-Coll Bcast and Scatter speedups (minimum rank-maximum latency) at 32 SPR nodes with B=128 and ε_{rel}=10^{-4}. MT: 112 threads; ST: one thread.

Size (MiB)	Bcast vs MPI MT	Bcast vs MPI ST	Bcast vs hZCCL MT	Bcast vs hZCCL ST	Scatter vs MPI MT	Scatter vs MPI ST	Scatter vs hZCCL MT	Scatter vs hZCCL ST
100	**47.2**	**99.9**	1.06	1.11	3.1	**1.9**	**1.83**	**1.20**
200	19.9	7.0	1.04	1.11	5.7	1.8	1.82	1.18
300	22.5	12.5	**1.28**	1.15	9.4	1.8	1.74	1.20
400	27.3	19.5	1.08	**1.17**	9.4	1.6	1.68	1.20
500	37.3	5.9	1.05	1.15	10.8	1.6	1.58	1.20
600	38.2	17.2	1.05	1.14	**23.4**	1.7	1.62	1.19

4 Conclusion

HV-Coll co-designs SIMD-native compression, outlier-aware CR recovery, compressed-domain reduction, and collective scheduling around a shared block-aligned representation. This co-design removes DOC work, improves CR at the production block size, and accelerates all four evaluated collectives with error-bounded compression, by up to 67.6× over native MPI.

Acknowledgments

This work was supported by a startup fund from the Bellini College of Artificial Intelligence, Cybersecurity and Computing at the University of South Florida. This work was also supported by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (DE-AC02-06CH11357); National Science Foundation (OAC-2311875, OAC-2514036, OAC-2513768). The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources. The authors used Claude [1] for format and grammar checks.

References

[1] Anthropic. 2026. Claude. https://claude.ai/. Large language model.

[2] Sébastien Cayrols, Jiali Li, George Bosilca, Stanimire Tomov, Alan Ayala, and Jack Dongarra. 2022. Lossy All-to-All Exchange for Accelerating Parallel 3-D FFTs on Hybrid Architectures with GPUs. In *Proceedings of the 2022 IEEE International Conference on Cluster Computing (CLUSTER)*. 152–160. doi:10.1109/CLUSTER51413.2022.00029

[3] Jiajun Huang, Sheng Di, Xiaodong Yu, Yujia Zhai, Jinyang Liu, Zizhe Jian, Xin Liang, Kai Zhao, Xiaoyi Lu, Zizhong Chen, Franck Cappello, Yanfei Guo, and Rajeev Thakur. 2024. hZCCL: Accelerating Collective Communication with Co-Designed Homomorphic Compression. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–15. doi:10.1109/SC41406.2024.00110

[4] 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. 2024. An Optimized Error-controlled MPI Collective Framework Integrated with Lossy Compression. In *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 752–764. doi:10.1109/IPDPS57955.2024.00072

[5] Pavol Klacansky. 2017. Open Scientific Visualization Datasets. http://klacansky.com/open-scivis-datasets/ Dataset collection website; accessed 2026-04-07.

[6] Minghao Li, Ran Ben Basat, Shay Vargaftik, ChonLam Lao, Kevin Xu, Michael Mitzenmacher, and Minlan Yu. 2024. THC: Accelerating Distributed Deep Learning Using Tensor Homomorphic Compression. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, 1191–1211. https://www.usenix.org/conference/nsdi24/presentation/li-minghao

[7] Qwen Team. 2026. Qwen3.5: Accelerating Productivity with Native Multimodal Agents. https://qwen.ai/blog?id=qwen3.5

[8] Qwen Team. 2026. Qwen/Qwen3.5-2B. https://huggingface.co/Qwen/Qwen3.5-2B Hugging Face model card; accessed 2026-04-07.

[9] Rolf Rabenseifner. 2004. Optimization of Collective Reduction Operations. In *Computational Science – ICCS 2004*, Marian Bubak, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–9. https://doi.org/10.1007/978-3-540-24685-5_1

[10] Texas Advanced Computing Center. 2026. Stampede3 - TACC HPC Documentation. https://docs.tacc.utexas.edu/hpc/stampede3/ Accessed 2026-04-09.

[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. doi:10.1177/1094342005051521

[12] Kai Zhao, Sheng Di, Xin Lian, Sihuan Li, Dingwen Tao, Julie Bessac, Zizhong Chen, and Franck Cappello. 2020. SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors. In *2020 IEEE International Conference on Big Data (Big Data)*. 2716–2724. doi:10.1109/BigData50022.2020.9378449

[13] Qinghua Zhou, C. Chu, N. S. Kumar, Pouya Kousha, Seyedeh Mahdieh Ghazimirsaeed, Hari Subramoni, and Dhabaleswar 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. doi:10.1109/IPDPS49936.2021.00053