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

Jiefeng Zhou[1], Sheng Di[2] (Advisor), Yanfei Guo[3] (Advisor), Lingqi Zhang[4] (Advisor), Kai Zhao[5] (Advisor), Rajeev Thakur[2] (Advisor), Franck Cappello[2] (Advisor), Jiajun Huang[1*] (Advisor)

[1]University of South Florida [2]Argonne National Laboratory [3]NVIDIA Corporation [4]RIKEN Center for Computational Science [5]Florida State University

Why compress collective communication?

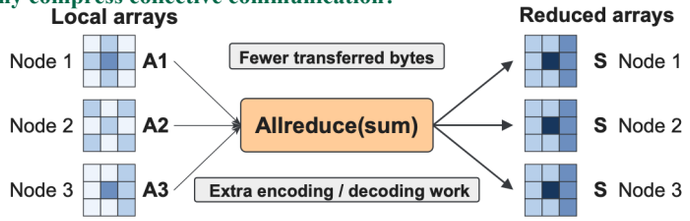


Figure 1: Allreduce sums corresponding elements and returns the same result to all nodes.

Collectives exchange or combine arrays across nodes in scientific simulations and distributed training. Large transfers can become a bottleneck. Error-bounded lossy compression reduces transferred bytes within user-defined error bound but adds encoding and decoding work.

What still limits compressed collectives?

- **Prior progress:** Compressed collectives reduce network traffic [1,2]. hZCCL performs addition in the encoded domain, avoiding repeated decompression and recompression during reductions [3].
- **CPU processing:** Scalar packing limits encoding throughput.
- **Block-size tradeoff:** Larger blocks suit SIMD processing, but one outlier can increase the bit width for the entire block and reduce compression ratio.
- **Core idea of HV-Coll:** HV-Coll utilizes a block-aligned representation for SIMD encoding, co-designing collective scheduling with homomorphic compression.

How HV-Coll Works

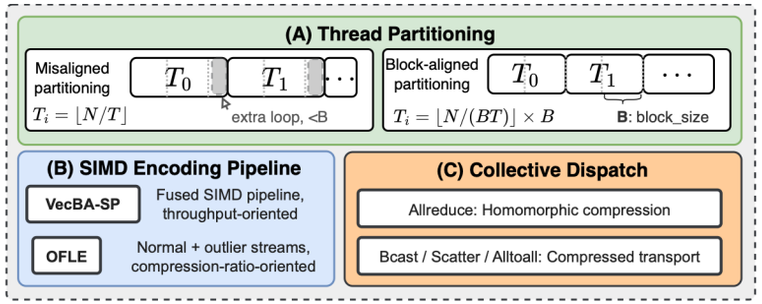


Figure 2: High-level overview of HV-Coll.

- **(A) Align thread partitions:** Assign whole blocks to threads to avoid splitting blocks at thread boundaries and support SIMD processing.
- **(B) Choose an encoding mode:** VecBA-SP fuses SIMD processing for high throughput. OFLE separates normal and outlier information to improve compression ratio.
- **(C) Match the collective path:** HV-Coll combines (A), (B) along with homomorphic compression into collective paths.

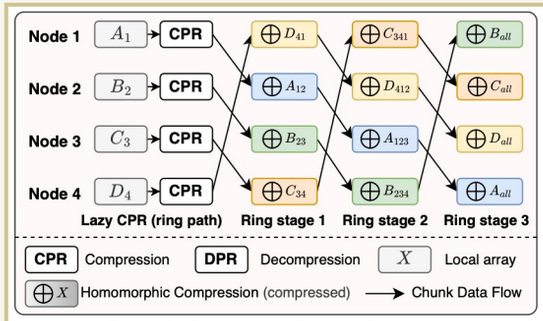


Figure 3: Co-designed Reduce-Scatter.

The ring path compresses only the segments needed to start, then compresses remaining segments as the reduction progresses. At each ring step, homomorphic compression adds received encoded blocks to local data without a full decompress–operate–compress cycle. The reduced payload is then reused in Allgather without recompression.

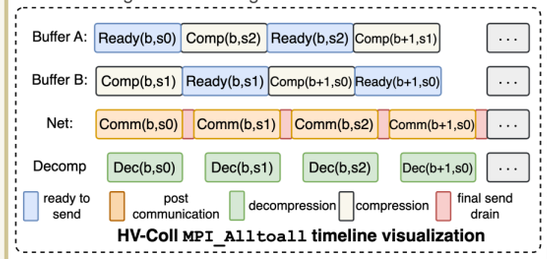


Figure 4: Alltoall pipeline in HV-Coll.

Alltoall alternates two send buffers: communication of the current segment overlaps compression of the next.

Receivers decompress completed segments. A global compression-ratio check selects native MPI if any rank's sampled ratio falls below the cutoff.

Experimental Evaluation

Hardware: Intel Xeon CPU MAX 9480 (SPR) node, interconnected with 100Gb/s Omni-Path network with fat-tree topology

Baseline: native MPI, prior SOTA (hZCCL)

Execution: pinned to fixed NUMA socket and use same OpenMP threads.

Datasets: refer to Table 1.

Dataset	Domain	Fields	Size
HACC	Cosmology (N-body)	6	6.27 GB
Hurricane	Weather Simu.	20	1.25 GB
NYX	Cosmology	6	3.10 GB
RTM Simu.	Seismic Imaging	151	95.3 GB
SCALE-LETKF	Climate Simu.	5	6.30 GB
JetIn	Turbulent Flow Simu.	1	6.23 GB
HCCI	Combustion Simu.	1	0.67 GB
QWEN-3.5-2B	AI Embeddings	53	9.09 GB

Table 1: Evaluation datasets used in experiments.

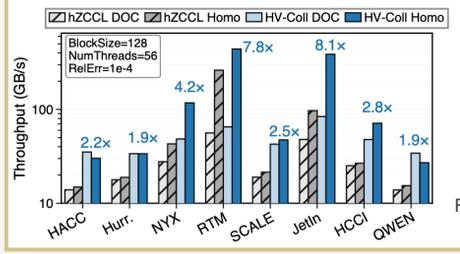


Figure 5: Homomorphic compression throughput comparison.

Homomorphic compression operates directly on compressed data, bypassing the DOC workflow.

By combining this path with VecBA-SP, HV-Coll achieves 1.9–8.1× speedup over hZCCL's conventional DOC path.

ε_{rel}	HACC	Hurr.	NYX	RTM	SCALE	JetIn	HCCI	QWEN
1E-2	+2.7%	+8.6%	+0.2%	+1.7%	+2.1%	+0.0%	+0.0%	+0.2%
1E-3	+2.9%	+26.7%	+1.3%	+6.8%	+7.5%	+3.5%	+5.4%	+1.1%
1E-4	**+12.8%**	**+27.2%**	**+4.9%**	**+7.0%**	**+12.1%**	**+9.4%**	**+10.4%**	**+4.5%**

Table 2: HV-Coll OFLE compression-ratio gains over hZCCL.

OFLE preserves SIMD-friendly block sizes while recovering more compression ratio on outlier-heavy datasets. Gains become more pronounced under tighter error bounds, reaching up to 27.2% over hZCCL.

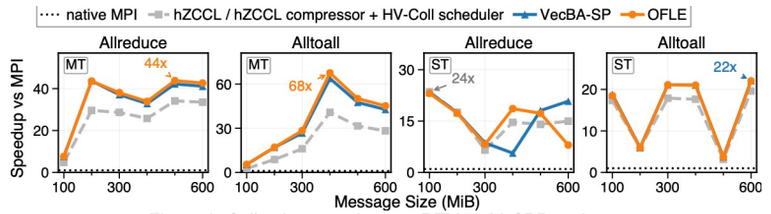


Figure 6: Collective speedups on RTM at 32 SPR nodes, B = 128 and εrel = 10^{-4}. MT: 112 threads; ST: one thread.

Allreduce generally outperforms hZCCL in multi-threaded runs, while single-thread results are mixed. For Alltoall, we use hZCCL's scalar compressor with the same HV-Coll transport path as a controlled baseline. Our ablation study shows that most gains come from the vectorized compressor and homomorphic compression, while scheduling adds a smaller benefit.

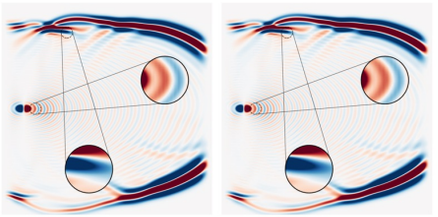


Figure 7: Representative image output from native MPI (left) and HV-Coll (right) in image stacking application.

At a local absolute compression bound of 10^{-5}, HV-Coll preserves the visual structure of the lossless reference with minimal visible distortion.

Summary

HV-Coll utilizes SIMD intrinsics to address the scalar packing issues, with VecBA-SP and OFLE modes to choose between throughput and compression ratio. Experimental evaluation shows that HV-Coll significantly outperforms prior SOTA in both compressor throughput and lossy collectives.

Acknowledgement

This work is supported by a startup fund from the Bellini College of Artificial Intelligence, Cybersecurity and Computing at the University of South Florida; U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (DE-AC02-06CH11357); National Science Foundation (OAC-2311875, OAC-2514036, OAC-2513768). This work used Stampede3 at the Texas Advanced Computing Center (TACC) through allocation CIS251134 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Huang, Jiajun, et al. "An optimized error-controlled mpi collective framework integrated with lossy compression." 2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2024.
[2] Huang, Jiajun, et al. "ZCCL: Significantly improving collective communication with error-bounded lossy compression." arXiv preprint arXiv:2502.18554 (2025).
[3] Huang, Jiajun, et al. "hZCCL: Accelerating collective communication with co-designed homomorphic compression." SC24: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2024.

Jiefeng@usf.edu