Optimizing Collectives with Large Payloads on GPU-based Supercomputers

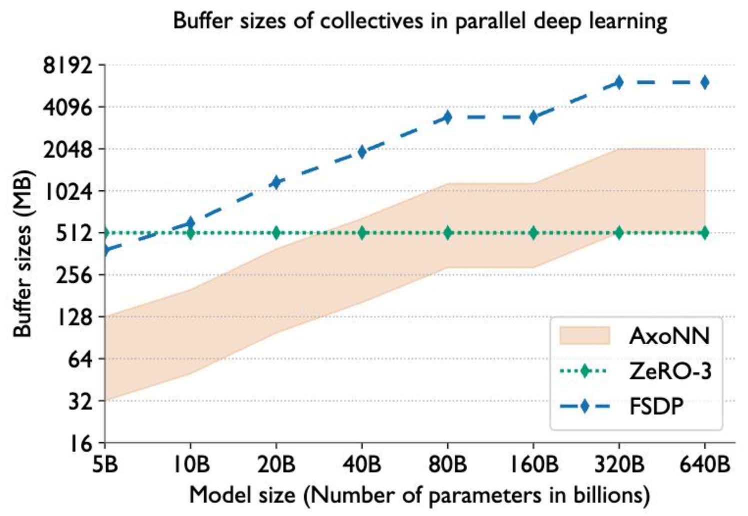
Siddharth Singh, Mahua Singh, Keshav Pradeep, Abhinav Bhatele
Department of Computer Science, University of Maryland

Abstract

We evaluate the current state of collective communication on GPU-based supercomputers for large language model (LLM) training at scale. Existing libraries such as RCCL and Cray-MPICH exhibit critical limitations on systems such as Frontier -- Cray-MPICH underutilizes network and compute resources, while RCCL suffers from severe scalability issues. To address these challenges, we introduce PCCL, a communication library with highly optimized implementations of all-gather and reduce-scatter operations tailored for distributed deep learning workloads. PCCL is designed to maximally utilize all available network and compute resources and to scale efficiently to thousands of GPUs. It achieves substantial performance improvements, delivering 6-33x speedups over RCCL and 28-70x over Cray-MPICH for all-gather on 2048 GCDs of Frontier. These gains translate directly to end-to-end performance: in large-scale GPT-3-style training, PCCL provides up to 60% and 40% speedups over RCCL for 7B and 13B parameter models, respectively.

Collective Communication in Parallel Deep Learning

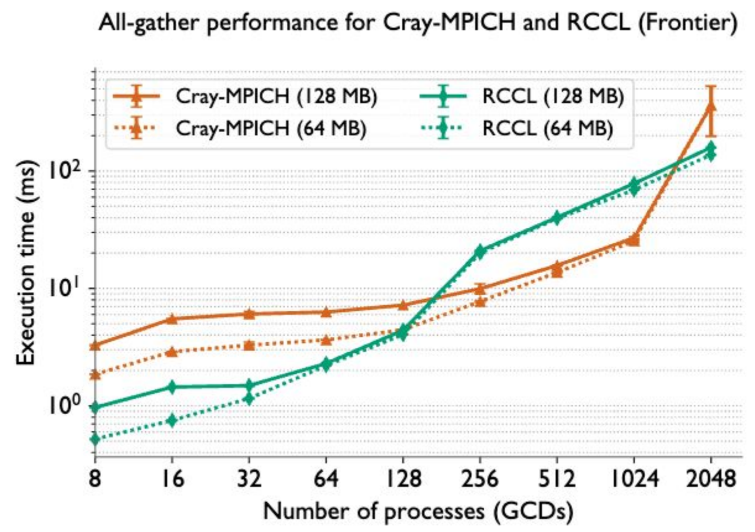
- *all-gather* & *reduce-scatter* operations:
  - used to synchronize & distribute model parameters
  - **primary bottleneck** in large-scale GPU-based LLM training
- Buffer sizes for *all-gather*/*reduce-scatter* collectives range from tens to hundreds of MB, even **exceeding 1 GB** for larger models [1]. (Fig on right)



Observed Cray-MPICH & RCCL Issues on Frontier

We observed the following issues:

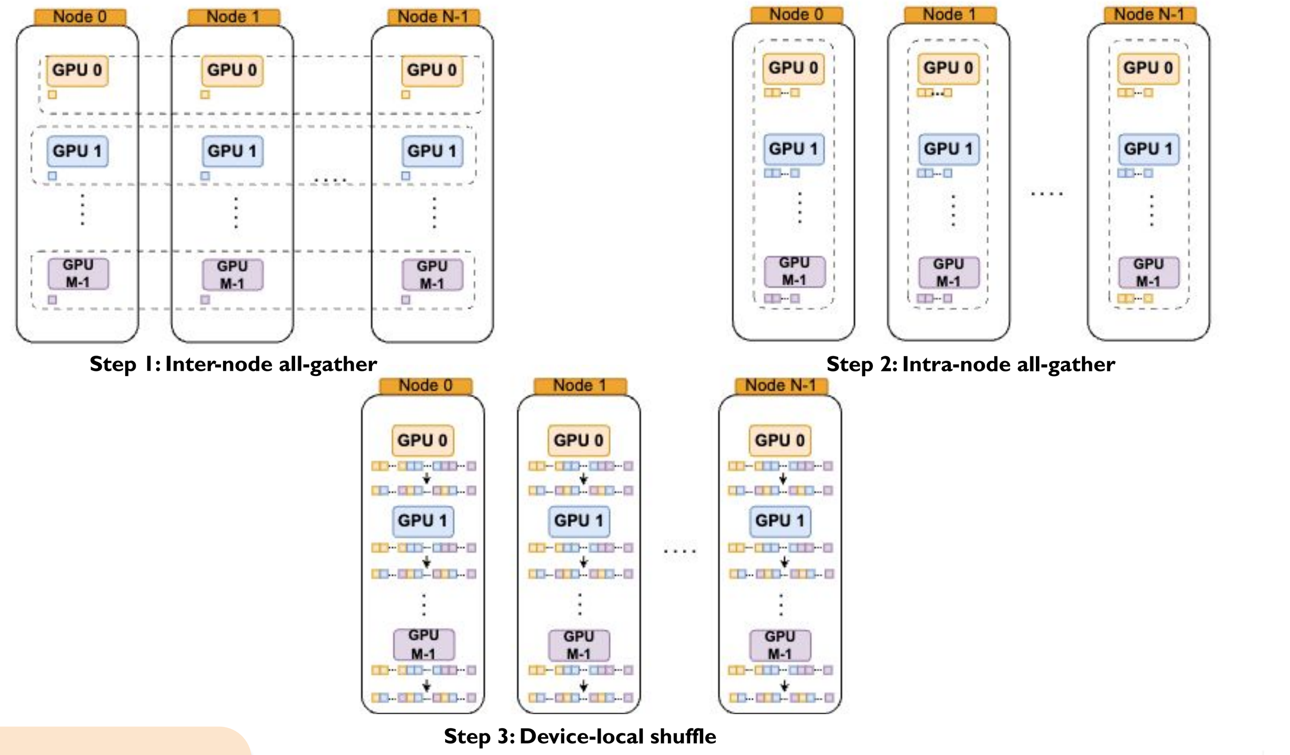
- Cray-MPICH exhibits **NIC imbalance**, resulting in underutilization of bandwidth
- Cray-MPICH uses **CPU-based reduction** operations, introducing significant overhead for data-movement
- Both Cray-MPICH and RCCL use ring *all-gather*/*reduce-scatter*, **limiting scaling for latency-bound workloads** at large GPU counts. (Fig on right)



Note: ideal scaling is flat line

Optimizing All-gathers and Reduce-scatters

- Addresses NIC underutilization by:
  - force each GCD send/recv traffic to/from assigned NIC
- Addresses CPU-based reductions by:
  - perform reductions with GPU-kernels
- Addresses scaling for latency-bound scenarios by:
  - **two-level hierarchical design** (Fig on right)
  - Prior works show hierarchical algorithms to reduce latency & improve scalability [2, 3].

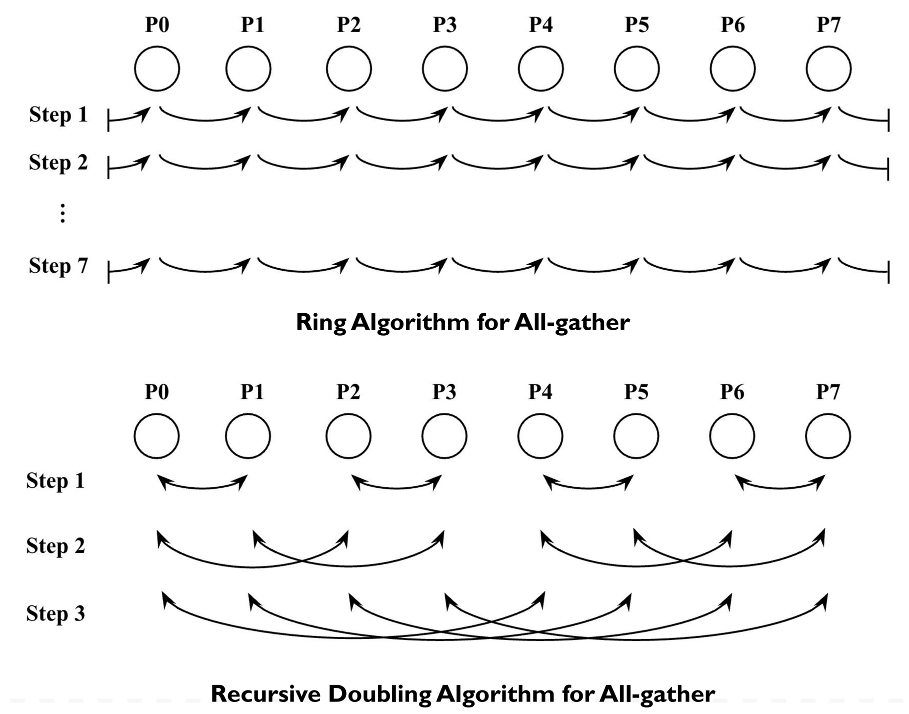


Choice of Communication Libraries for Each Level of Hierarchy

- **Intra-node: RCCL is chosen** since it is highly optimized for GPU-to-GPU intra-node communication, leveraging shared memory, PCIe, and Infinity fabric.
- **Inter-node: Cray-MPICH is chosen** primarily for reliability, as RCCL has been reported to be unstable and prone to crashing at scale [4].

Choice of Algorithms for Each Level of Hierarchy

- **Intra-node: Ring is chosen** because the small number of GCDs per node allows it to effectively saturate the available bandwidth.
- **Inter-node: Recursive-doubling/halving is chosen** to achieve logarithmic latency terms, enabling significantly better performance at large GPU counts.

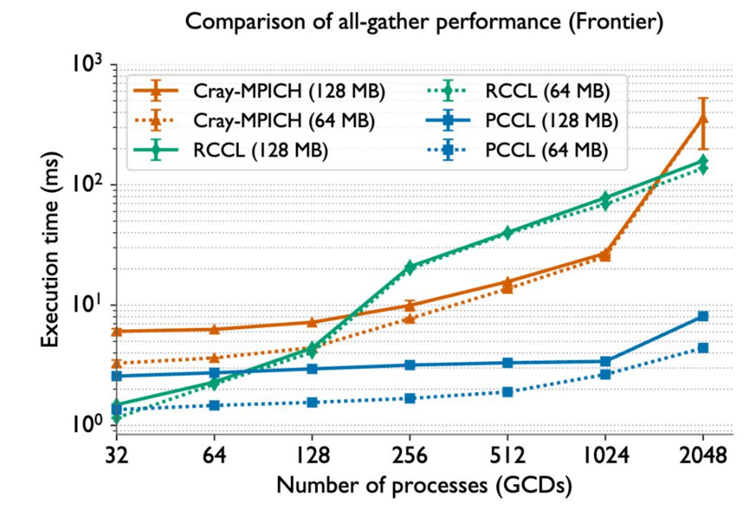


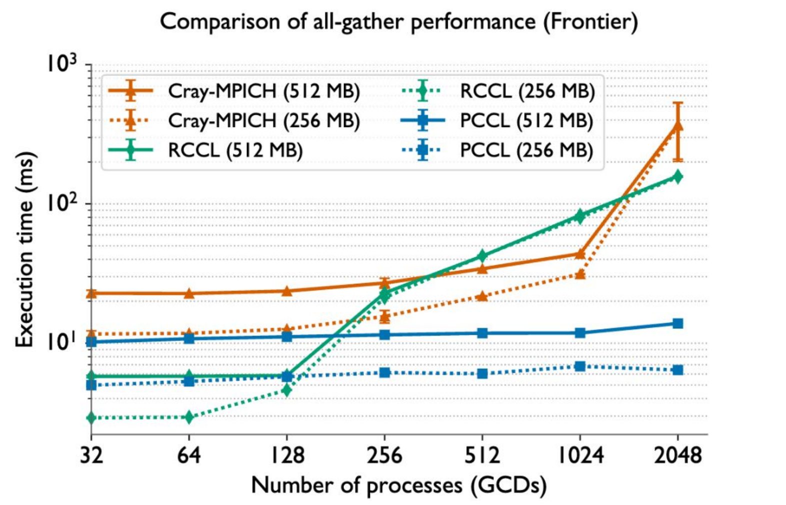
Experimental Setup

- Experiments conducted on Frontier
- Benchmarking *all-gather* and *reduce-scatter* operations for PCCL and RCCL
  - message sizes ranging from 16MB to 1GB
  - job sizes ranging from 32 to 2048 GCDs
- Benchmarking end-to-end training of GPT-3 style models with DeepSpeed ZeRO-3:
  - 7B and 13B parameter counts
  - Strong-scaling between 128 and 2048 GCDs

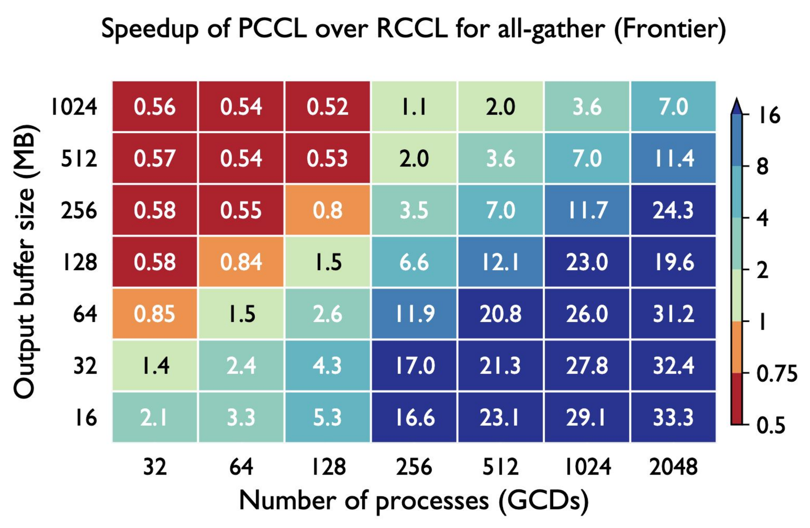
	OLCF Frontier
GPU model	4x AMD MI250X (8 GCDs)
Device memory	64 GB per GPU
CPU model	64-core AMD EPYC 7713 Trento CPU
Interconnect	4x HPE Slingshot 200 Gbps NICs

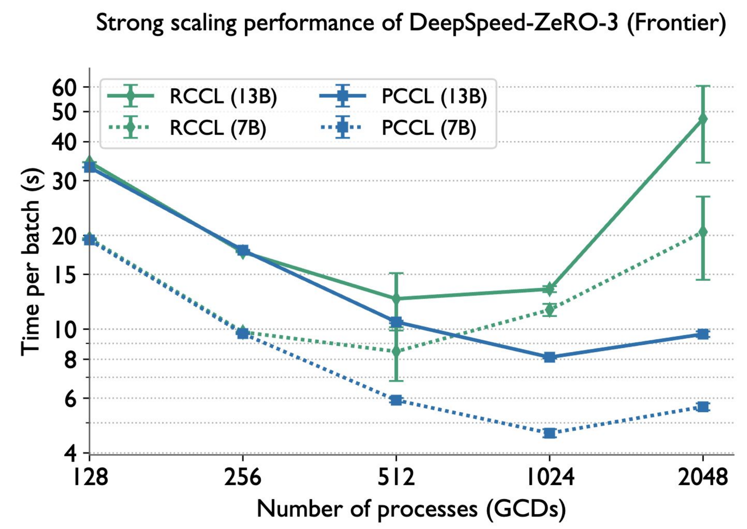
Results





- PCCL *all-gather* maintains **near-flat scaling** with GCD count across message sizes
- Similar trends observed for PCCL *reduce-scatter*





- **Left:** PCCL achieves up to **33.3x speedup** over RCCL in **latency-bound** scenarios, but up to 0.52x slowdowns in bandwidth-bound scenarios
- **Right:** For end-to-end DeepSpeed ZeRO-3 training, at 2048 GCDs, PCCL **reduces batch time** by 72% (7B model) and **79%** (13B model) relative to RCCL.

Conclusion

- This work identifies critical shortcomings in existing collective communication libraries including Cray-MPICH & RCCL, which make them unsuitable for scalable deep learning model training
- We develop PCCL to address these bottlenecks with highly optimized implementations for all-gather and reduce-scatter, achieving 6-33x speedups over RCCL for all-gather on Frontier, translating to significant end-to-end training gains of up to 79% for a 13B model.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1650114. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award m2404 for 2023.

References

[1] Siddharth Singh, Prajwal Singhania, Aditya Ranjan, John Kirchenbauer, Jonas Geiping, Yuxin Wen, Neel Jain, Abhimanyu Hans, Manli Shu, Aditya Tomar, Tom Goldstein, and Abhinav Bhatele. 2024. Democratizing AI: Open-source Scalable LLM Training on GPU-based Supercomputers. In Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC '24).
[2] Minsik Cho, Ulrich Finkler, David Kung, and Hillery Hunter. 2019. BlueConnect: Decomposing All-Reduce for Deep Learning on Heterogeneous Network Hierarchy. In Proceedings of Machine Learning and Systems, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 241–251. https://proceedings.mlsys.org/paper_files/paper/2019/file/0c8abcf158ed12d0dd94480681186fda-Paper.pdf
[3] Truong Thao Nguyen, Mohamed Wahib, and Ryousei Takano. 2018. Hierarchical Distributed-Memory Multi-Leader MPI-Allreduce for Deep Learning Workloads. In 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW). IEEE Computer Society, Los Alamitos, CA, USA, 216–222. doi:10.1109/CANDARW.2018.00048
[4] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. 2025. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. arXiv:2502.05171 [cs.LG] https://arxiv.org/abs/2502.05171