Center for Understandable Performant Exascale Communication Systems

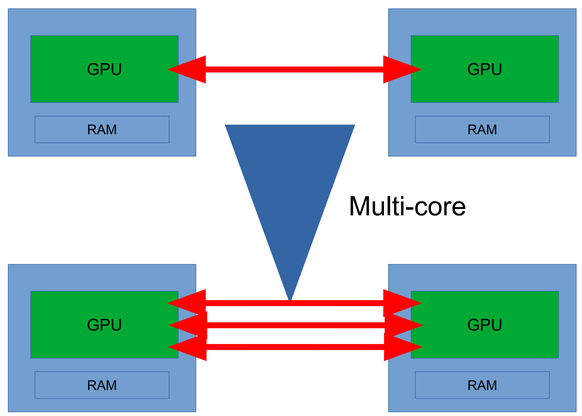
THE UNIVERSITY OF NEW MEXICO

Optimizing the GPU Allreduce using Multiple Processes per GPU

Michael Adams and Amanda Bienz (advisor) | Department of Computer Science

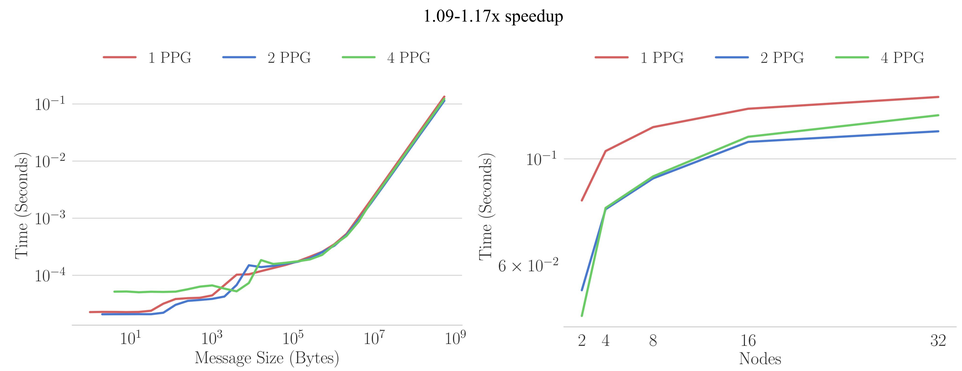
Background

- Deep learning and sparse linear algebra applications rely on the GPU-aware Allreduce
- Copy-to-CPU and GPUDirect RDMA communication paths rely on NIC configuration and device concurrency [3] to enhance performance
- We take advantage of growing underutilized core counts to utilize more cores during communication via GPU Interprocess Communication to yield further speedup
  - NCSA Delta uses Copy-to-CPU with an intermediate host buffer
  - LLNL Tuolumne uses GPUDirect RDMA with an intermediate device buffer



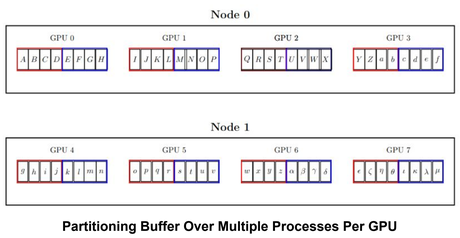
Results

The left figures show timings on Tuolumne using Cray MPICH with 1 to 4 processes per GPU, where each process reduces a portion of the buffer using an intermediate device buffer. Speedup is seen for large Allreduces. 2 processes per GPU shows greater speedup at scale, yielding **1.17x** speedup on 32 nodes where 4 processes per GPU yields **1.09x** speedup.

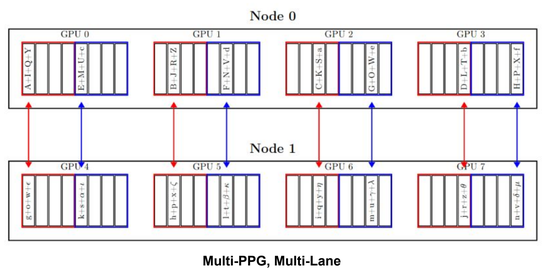


Methods

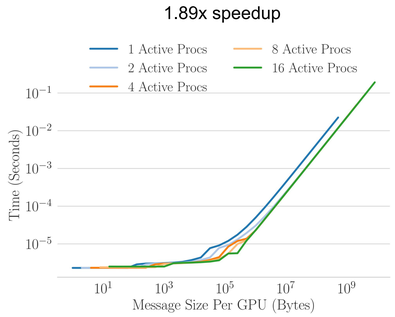
- Create new communicators:
  - Standard multi-PPG Allreduce
    - new_comm: processes with equal node rank
  - Lane multi-PPG Allreduce [1, 2]
    - group_comm: on-node processes with equal (node rank % number of GPUs)
    - lane_comm: off-node processes with equal node rank
- Get device memory handle with (cuda/hip)GetMemHandle
- Broadcast to other processes associated with target device with MPI_Bcast
- Get device pointer on other associated processes with (cuda/hip)OpenMemHandle
- Perform a standard or multi-lane Allreduce on disjoint partitions of input buffer



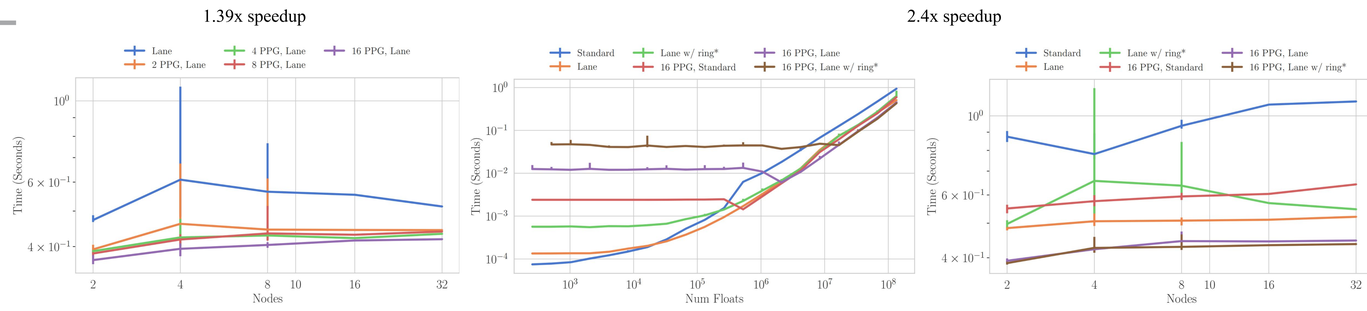
Partitioning Buffer Over Multiple Processes Per GPU



Multi-PPG, Multi-Lane



Tuolumne GPUDirect ping-pong benchmark shows that utilizing an extra CPU core per GPU yields speedup.



In the figures above, the left shows the performance of the multi-lane algorithm with increasing numbers of processes per GPU on Delta using an intermediate host buffer. Large Allreduces on 8 nodes yield a **1.39x** speedup over the naive multi-lane algorithm. The right shows the performance of all algorithms, comparing 1 versus 16 processes per GPU. The multi-lane algorithm with 16 processes per GPU achieves up to **2.4x** speedup for large data sizes.

Conclusions

Our multi-CPU-accelerated GPU-aware lane Allreduces yield speedup of up to 2.45x for large MPI Allreduces across the NVIDIA A100 GPUs of NCSA's Delta supercomputer. Extensions to Allreduces using GPUDirect RDMA communication yield speedup of up to 1.17x on LLNL's Tuolumne supercomputer.

We propose to extensively evaluate the performance characteristics of our multi-process approaches and also extend them to collectives such as the neighborhood Alltoall used in sparse linear systems.

Acknowledgements

This work was performed with partial support from the National Science Foundation under Grant No. CCF-2338077, the U.S. Department of Energy's National Nuclear Security Administration (NNSA) under the Predictive Science Academic Alliance Program (PSAAP-III), Award DE-NA0003966. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and the U.S. Department of Energy's National Nuclear Security Administration.
This research used the Tuolumne supercomputer at Lawrence Livermore National Laboratory. This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.
Generative AI was used to improve the formatting of data within plots.

References

1. J. L. Traff and S. Hunold, Decomposing mpi collectives for exploiting multi-lane communication, 2020 IEEE International Conference on Cluster Computing (CLUSTER), (2020).
2. De Rango, A., Utrera, G., Gil, M., Martorell, X., Giordano, A., D'Ambrosio, D., Mendicino, G.: Partitioned reduction for heterogeneous environments. In: 2024 32nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). pp. 285–289 (2024).
3. Hu, Zhiyi, Siyuan Shen, Tommaso Bonato, Sylvain Jeaugey, Cedell Alexander, Eric Spada, James Dinan, Jeff Hammond, Torsten Hoefler., Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms. arXiv:2507.04786, 2025.