UNDERSTANDING INTER-GPU ALLREDUCE PERFORMANCE THROUGH HARDWARE TELEMETRY

Ram Sharan Chaulagain, Nathan Nichols, Servesh Muralidharan, Kalyan Kumaran
{rchaulagain,nnichols,smuralidharan,kumaran}@anl.gov
Argonne National Laboratory

Which hardware behaviors distinguish faster and slower AllReduce configurations?

MOTIVATION

- Inter-GPU AllReduce is core to distributed AI and HPC, so its performance bounds how well large runs scale.
- The same AllReduce runs many ways (algorithm, protocol, software parallelism) ; each reshaping traffic, sync, and data movement.
- It crosses tightly coupled components (GPU - PCIe - NIC - network), so a limitation can emerge anywhere along the path
- Every component exposes telemetry, but each counter set captures only its own local behavior. No single source sees the complete path, making it hard to localize where a slowdown occurs.
- Combined GPU–NIC telemetry is therefore needed to narrow the limit and direct optimization.

TELEMETRY SOURCES

GPU

Collected using *NVIDIA NSight Compute*

Active SMs and SM throughput.

Warp-stall behavior.

GPU/host memory traffic.

PCIe transfer volume and rate.

NIC

Collected using *libcxi* API

Packet and traffic volume.

Request--response latency.

Address-translation and cache activity

NIC command activity.

Blocking/congestion counters.

Per-rank traffic variation.

EXPERIMENTAL SETUP

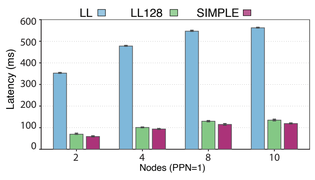
- Polaris: 4X NVIDIA A100 GPUs, 2X Cassini NICs, HPE Slingshot-11.
- One MPI rank, one GPU, and one NIC per node; inter-node communication only.
- 1~GiB double-precision AllReduce at 2, 4, 8, and 10 nodes.
- Cray MPICH~9.0.1,NCCL~2.26,CUDA~12.9,libfabric~2.2.0rc1, aws-ofi-nccl~1.9.1.
- Three case studies:
  - NCCL Simple, LL, and LL128 with Ring and channel count fixed
  - NCCL channel sweep from 1-32 with Ring and Simple fixed
  - GPU-aware MPI Rabenseifner versus NCCL Ring.

NCCL PROTOCOL SWEEP

NCCL Protocols

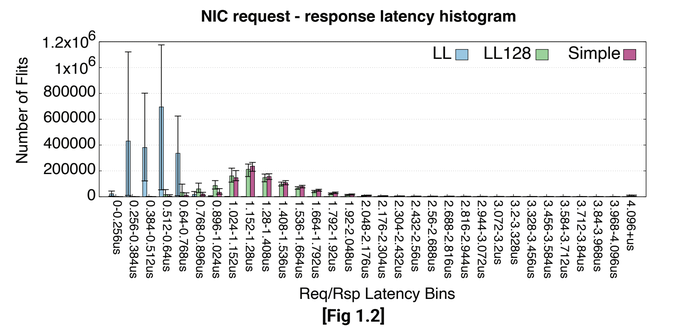
	Simple	Low Latency (LL)	LL128
Design Goal	High BW	Low Latency	Low latency + High BW
Sync. Mechanisms	Memory fence	Flag-based sync	Flag-based sync
Payload	Data chunks	4B data + 4B flag	120B data + 8B flag
Bandwidth Utilization	Near peak	25 ~ 50 % of peak	~ 95% of peak
Latency per hop	~6us	~1us	~2us
Buffer size	4MiB	256 KiB	4800KiB

For large message size, bandwidth matters more than per-event latency.



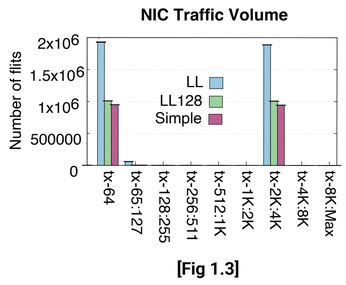
[Fig 1.1]

Simple is fastest for the 1 GiB workload. At 8–10 nodes, LL takes about 550–560 ms, versus 110–115 ms for Simple and 125–130 ms for LL128.

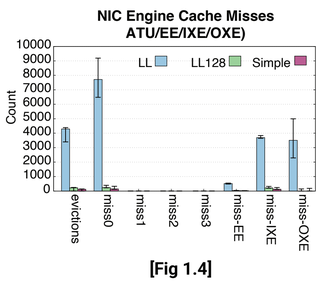


[Fig 1.2]

LL lowers per-event NIC latency, but not end-to-end latency. Most LL request–response events fall in the 0.256–0.640 us range, compared with about 1.0–1.5 us for Simple and LL128.

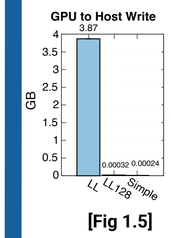


[Fig 1.3]

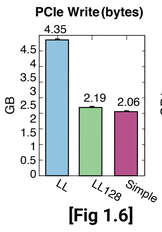


[Fig 1.4]

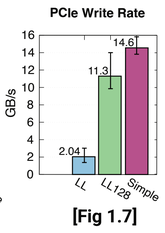
- LL performs more communication work: its 4B data + 4B flag format doubles wire traffic and generates ~20× more NIC commands. This higher activity increases data-path pressure, with more ATU evictions, translation misses, and NIC engine-cache misses, indicating greater NIC-side processing overhead.



[Fig 1.5]



[Fig 1.6]



[Fig 1.7]

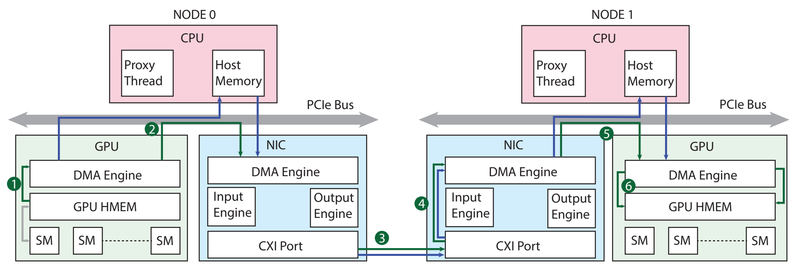
LL generates 3.87 GB GPU-to-host and 4.35 GB PCIe writes at 2.04 GB/s, versus Simple's 2.06 GB and 14.6 GB/s.

LL optimizes individual low-latency transfers, but for a large 1 GiB AllReduce it amplifies traffic, increases NIC command pressure, and introduces additional low-rate host/PCIe movement. Simple and LL128 avoid these costs.

LIMITATIONS AND FUTURE WORK

- Limited to 1 GiB AllReduce on Polaris/A100.
- Only selected GPU and NIC counters are analyzed.
- Telemetry alone shows resource behavior, not software-level causality.
- Function-level profiling and logs provide the missing software context.
- Evaluate multi-rank-per-node configurations to capture intra-node contention and scaling effects.

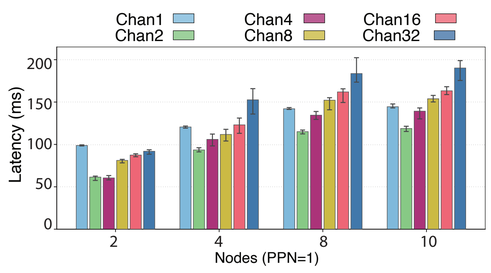
INTER-GPU COMMUNICATION



1. GPU DMA reads data from HMEM
2. Data moves over PCIe to the NIC
3. NIC injects data into the network
4. Remote NIC receives the data
5. Data moves over PCIe to the GPU
6. GPU DMA writes data to HMEM

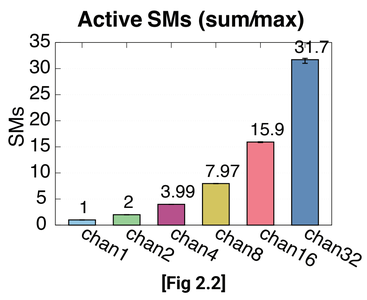
NCCL CHANNEL SWEEP

More channels do not continuously improve AllReduce performance.

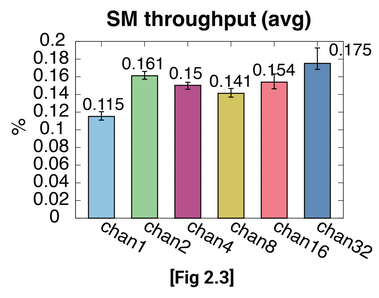


[Fig 2.1]

U-shaped latency: lowest around 2 channels.

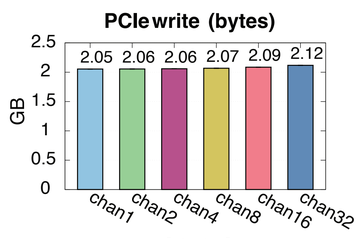


[Fig 2.2]

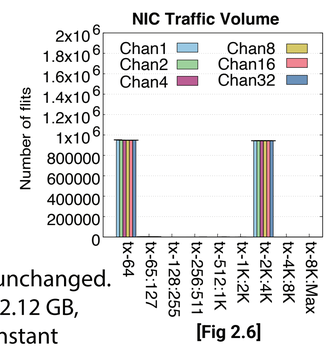


[Fig 2.3]

More channels activate more GPU resources, but do not increase useful throughput. Active SMs rise from about 1 to 31.7, while SM throughput remains nearly flat at 0.12–0.18.

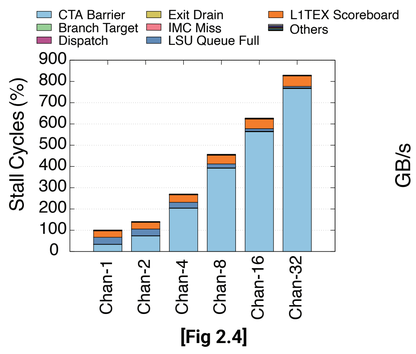


[Fig 2.5]

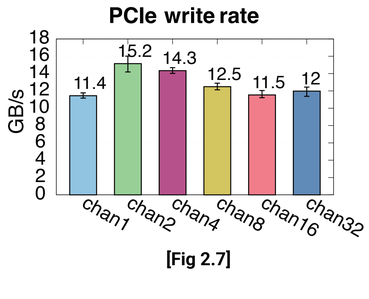


[Fig 2.6]

Communication volume remains nearly unchanged. PCIe write volume stays around 2.05–2.12 GB, and NIC packet volume is nearly constant across the sweep.



[Fig 2.4]

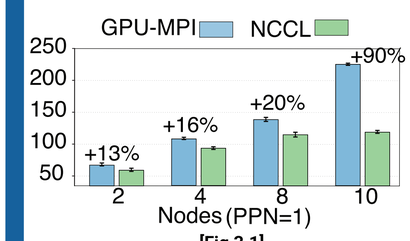


[Fig 2.7]

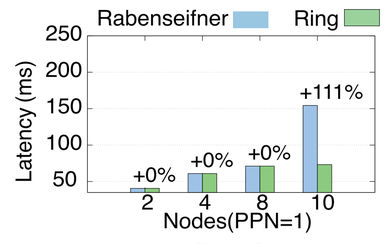
More channels increase parallelism, but beyond the optimal point, higher barrier stalls and lower PCIe transfer rates limit AllReduce performance.

MPI vs NCCL

Same work, slower progress at 8 nodes; extra work and imbalance at 10 nodes.



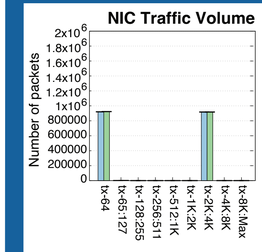
[Fig 3.1]



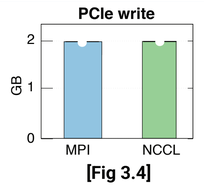
[Fig 3.2]

At 2–8 nodes, Rabenseifner and Ring have similar modeled cost, but MPI is 13–20% slower than NCCL.
At 10 nodes, Rabenseifner requires extra pre/post-processing for the two excess ranks, and MPI becomes about 90% slower than NCCL.

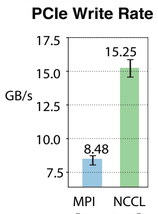
8 Nodes



[Fig 3.3]



[Fig 3.4]

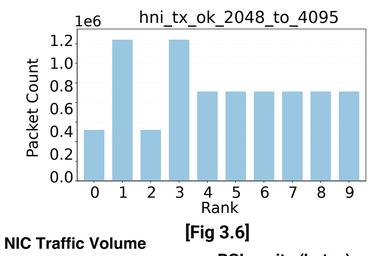


[Fig 3.5]

MPI moves this data more slowly. PCIe write rate is 8.48 GB/s for MPI versus 15.25 GB/s for NCCL.

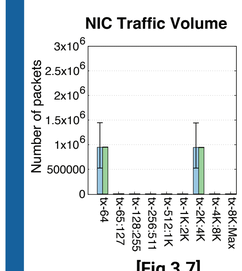
At 8 nodes, MPI and NCCL move about the same amount of data.
NIC traffic is balanced and PCIe write volume (2GB) is identical.

10 Nodes

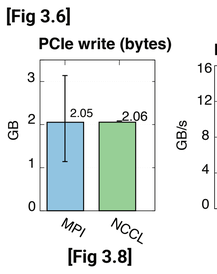


[Fig 3.6]

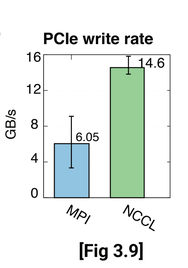
- MPI trafic becomes rank dependent at 10 nodes. Some ranks move substantially more NIC and PCIe data than others, while NCCL remains more uniform.



[Fig 3.7]

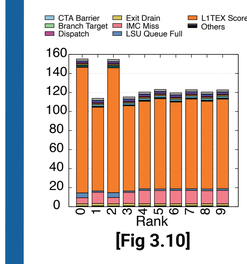


[Fig 3.8]

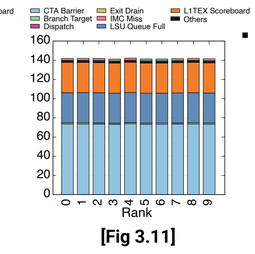


[Fig 3.9]

- MPI also transfers data more slowly: PCIe write rate drops to 6.05 GB/s, compared with 14.6 GB/s for NCCL.



[Fig 3.10]



[Fig 3.11]

- The GPU stall breakdown shows substantial waiting in both implementations. Therefore, the large 10-node gap is better explained by MPI's additional algorithmic work, rank imbalance, and lower transfer rate, rather than by a single GPU stall category.

CONCLUSIONS

- Joint GPU–NIC telemetry separates different AllReduce bottlenecks: extra data movement for NCCL LL, synchronization overhead at high channel counts, and slow transfer progress with MPI rank imbalance.
- No single counter explains performance. GPU and NIC measurements must be interpreted together to distinguish more work from slower progress or imbalance.
- Hardware telemetry shows where the difference appears, but not always why it occurs. Some causes are only clear when combined with software logs, algorithm behavior, and performance models.

REFERENCES

[1] Z. Hu, S. Shen, T. Bonato, S. Jeaugey, C. Alexander, E. Spada, J. Dinan, J. R. Hammond, and T. Hoefler, "Demystifying NCCL:
An in-depth analysis of GPU communication protocols and algorithms," in Proc. IEEE Symp. High-Performance Interconnects (HOTI), 2025, pp. 48–59, doi: 10.1109/HOTI66940.2025.00024.

[2] D. De Sensi, S. Di Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the Slingshot interconnect," in Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis, Nov. 2020, pp. 1–14, doi: 10.1109/SC41405.2020.00039.

[3] B. Videau. (2024, Oct.) Polaris: Overview of hardware and software. ALCF Hands-on HPC Workshop presentation slides. Argonne Leadership Computing Facility. Accessed: Jul. 29, 2026. [Online]. Available: https://www.alcf.anl.gov/sites/default/files/2024-11/Polaris-Overview-ALCF-HandsOnWorkshop-2024.pdf

[4] NVIDIA Corporation. (2026) NVIDIA Collective Communications Library (NCCL) user guide. NCCL 2.30.7 documentation. Accessed: Jul. 29, 2026. [Online]. Available: https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/

[5] R. Rabenseifner, "Optimization of collective reduction operations," in Computational Science—ICCS 2004, ser. Lecture
Notes in Computer Science, M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, Eds., vol. 3036. Berlin, Germany:
Springer, 2004, pp. 1–9, doi: 10.1007/978-3-540-24685-5 1.

[6] R. Thakur, R. Rabenseifner, and W. Gropp, "Optimization of collective communication operations in MPICH," Int. J. High Perform. Comput. Appl., vol. 19, no. 1, pp. 49–66, 2005, doi: 10.1177/1094342005051521.

[7] NVIDIA Corporation. (2026) NVIDIA Nsight Compute profiling guide. Nsight Compute 13.3 documentation. Accessed: Jul. 29, 2026. [Online]. Available: https://docs.nvidia.com/nsight-compute/ProfilingGuide/

U.S. DEPARTMENT of ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357.

Argonne NATIONAL LABORATORY