Using Different MPI Implementations on HPE Cray EX Supercomputers for Native and Containerized Applications Execution

Maciej Pawlik, Academic Computer Centre CYFRONET, m.pawlik@cyfronet.pl
Maciej Szpindler, Academic Computer Centre CYFRONET, m.szpindler@cyfronet.pl
Marcin Krotkiewski, University of Oslo, marcin.krotkiewski@usit.uio.no
Alfio Lazzaro, HPE, alfio.lazzaro@hpe.com

Cray User Group (CUG) 2025
May 4 – May 8, 2025

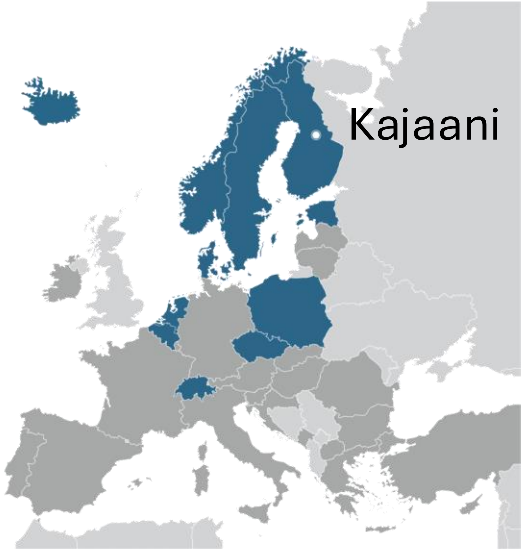
Outline

- The LUMI Supercomputer
  - Consortium
  - Hardware
- The Helios System
- MPI implementations
  - Motivations / Background / Goals
  - Software configuration description
  - Containers
- Performance results
  - Synthetic benchmarks
  - Real application benchmarks
- Conclusion

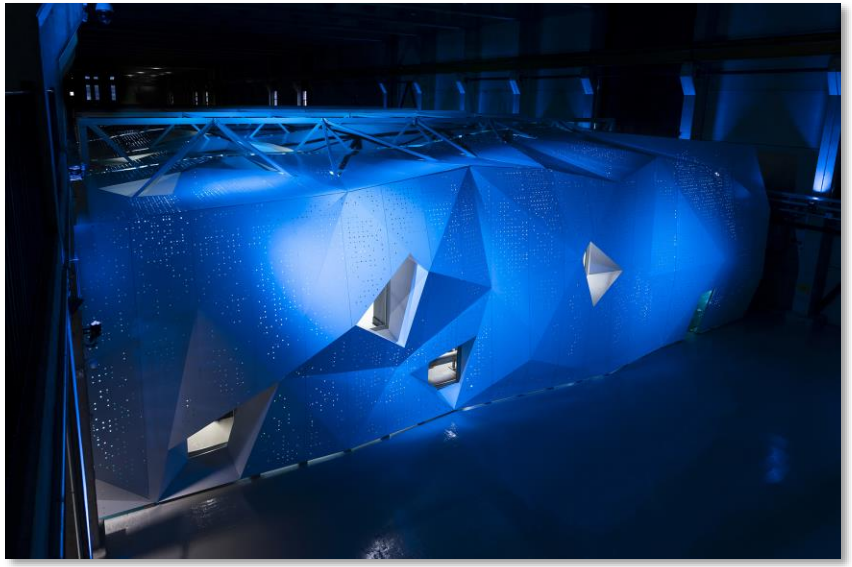
www.lumi-supercomputer.eu

The LUMI Supercomputer

- LUMI – Large Unified Modern Infrastructure
  - The word "lumi" means "snow" in Finnish
- Co-funded by the EuroHPC Joint Undertaking and the LUMI Consortium
  - Finland, Belgium, the Czech Republic, Denmark, Estonia, Iceland, the Netherlands, Norway, Poland, Sweden, and Switzerland
- Located in CSC's data center in Kajaani, Finland
  - The machine is sited in a former paper mill
  - 100% renewable energy
  - Managed by CSC



https://www.lumi-supercomputer.eu/lumi-consortium/



https://www.lumi-supercomputer.eu/media/

LUMI System

- HPE Cray EX system
  - ~550 PFlop/s peak performance
  - #8 in Top500 list (Nov. 2024)
    - Fastest Supercomputer in Europe May 2022 – May 2024

Image © CSC, Finland

- Two partitions:
  - **LUMI-C**
    - Only CPU nodes, 2048 nodes
    - 2 × AMD EPYC 7763 “Milan”
  - **LUMI-G**
    - GPU nodes, 2978 nodes
    - AMD EPYC 7A53 “Trento” + 4 AMD MI-250X GPUs
- HPE Cray Slingshot-11 (SS-11) network

Helios System

- HPE Cray EX4000 system
  - ~35 PFlop/s peak performance
  - #69 in Top500 list (Nov. 2024) - GPU part
    - Fastest Supercomputer in Poland 2023 – now (2025)
  - #7 in Green500 list (Nov. 2024) - GPU part

- Two partitions:
  - **Helios CPU**
    - Only CPU nodes, 432 nodes
    - 2 × AMD EPYC 9654 “Genoa”
  - **Helios GPU**
    - GPU nodes, 110 nodes
    - 4x NVIDIA Grace Hopper GH200

- HPE Cray Slingshot-11 (SS-11) network

Motivations

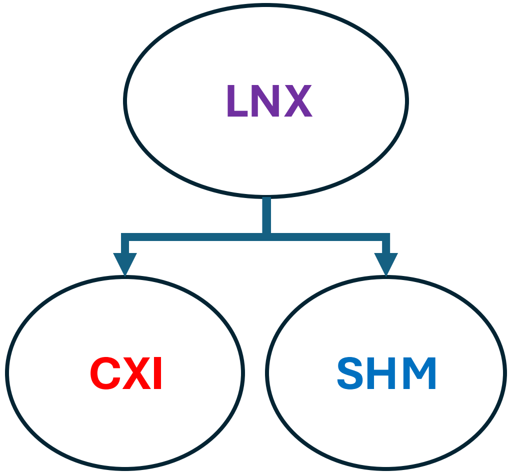
- The advantages of running with other MPI implementations on HPE Cray EX systems to the pre-installed Cray MPI are the possibility of
  - comparing performance
  - investigating bugs
  - checking new MPI functionalities (for example, as part of the new MPI standard)

➔ **We need to check that the performance is comparable to Cray MPI**

- OpenMPI is quite popular as the default MPI implementation in most common Linux distributions
  - EasyBuild FOSS stacks use OpenMPI and they are popular at all sites of interest
  - Providing OpenMPI as an alternative can benefit users who have applications or containers that are targeting OpenMPI and where some effort would be needed to change

Background

- OpenMPI version v5.0 has improvements for running on SS-11 interconnects
  - *Open MPI for HPE Cray EX Systems*, Pritchard, Howard P., *et al.*, CUG23 proceedings
  - https://cug.org/proceedings/cug2023_proceedings/includes/files/pap140s2-file1.pdf
- This is achieved by relying on a new OFI Libfabric provider, called LNX, which has been released as part of the OFI Libfabric version 2.0.0 (https://ofiwg.github.io/libfabric/v2.1.0/man/fi_lnx.7.html)
  - LNX provider allows applications to seamlessly use multiple providers, specifically HPE Cassini (CXI) provider for SS-11 network (inter-node traffic) and shared memory (SHM) provider (intra-node traffic)



Goals

- A goal was to install OpenMPI + OFI Libfabric and compare the performance against Cray MPI
- We are targeting “traditional” HPC applications, i.e. no AI applications
- Specifically, we consider GPU-to-GPU (G2G) MPI communications
  - Cray MPI supports G2G communications via the HPE proprietary GPU Transport Layer (GTL) library, which has to be included during the application's linking phase
- By having a host-installed version of OpenMPI, we can run a container with OpenMPI and bind the host library at runtime (Hybrid model)
  - This allows us a simple OpenMPI install into the container
  - Assumes ABI compatibility between host and container OpenMPI
  - OpenMPI is not ABI compatible with Cray MPI

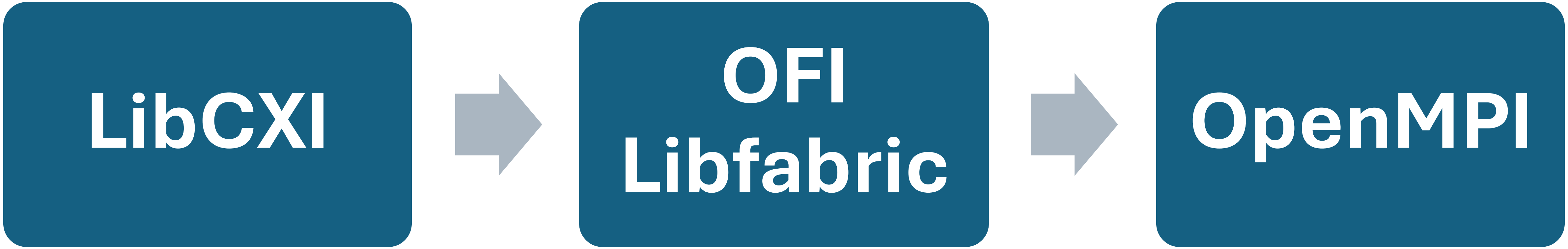
LUMI Software (relevant to our work)

- SUSE Linux Enterprise Server 15 SP5
- HPE Cray Program Environment (CPE): 24.03
  - Cray MPI 8.1.29
  - GNU 13.2
- OFI Libfabric 1.15.2.0 with CXI network provider (for the Cray MPI)
  - SS-11 software (LibCXI) 2.1.1
  - CXI NIC firmware 1.5.49
- ROCm 6.0.3
- Slurm batch system (23.02.7)
  - PMIx (v4) plugin is available but no PMIx library installed
- SingularityCE 4.1.3

Helios Software (relevant to our work)

- SUSE Linux Enterprise Server 15 SP5
- Cray MPI 8.1.28
- GNU 12.3
- OFI Libfabric 1.15.2.0 (for the Cray MPI)
  - NOTE: this version of the library is not optimized for NVIDIA GH nodes!
- **CUDA 12.8 (new versions to support GH)**
- GDRCopy 2.5
- Slurm batch system (23.02.7)
  - PMIx (v4) plugin is available but no PMIx library installed

Configuration – Building – Overview



LibCXI

OFI Libfabric

OpenMPI

Configuration – Building – Details

- **LibCXI a38a9c5** (https://github.com/HewlettPackard/shs-libcxi)
  - The system installed version is too old, we update it at the user-level
  - Dependencies:
    - Libconfig 1.7.3 (https://github.com/hyperrealm/libconfig)
    - Libuv 1.50.0 (https://github.com/libuv/libuv)
    - lm-sensors 3.6.0 (https://github.com/lm-sensors/lm-sensors)
    - libfuse 2.9.9 (https://github.com/libfuse/libfuse)
    - shs-cxi-driver d2ce7e6 (https://github.com/HewlettPackard/shs-cxi-driver)
    - shs-cassini-headers 59b6de6 (https://github.com/HewlettPackard/shs-cassini-headers)
- **OFI Libfabric 2.1.0** (https://github.com/ofiwg/libfabric)
  - Other dependency: json-c (https://github.com/json-c/json-c)
  - Built-in providers: lnx, dmabuf_peer_mem, hook_hmem, hook_debug, trace, perf, lpp, sm2, shm, cxi
- **OpenMPI 5.0.7** (https://github.com/open-mpi/ompi)
  - Apply a patch for G2G, described at https://github.com/open-mpi/ompi/issues/13048
  - Link with xpmem and ROCM/CUDA available on the systems

Configuration – Running (1)

- Since OpenMPI v5.0, PMIx integration is the only supported method to directly run applications based on OpenMPI via SLURM
  - PMIx library is not system installed, need sudo installation to make it work with SLURM
- We managed to run jobs within SLURM allocations by relying on the OpenMPI `mpirun` command and setting the following variables **(OpenMPI+LNX configuration)**:
  - `FI_LNX_PROV_LINKS="shm+cxi"`
  - `FI_SHM_USE_XPMEM=1`
  - `OMPI_MCA_opal_common_ofi_provider_include="shm+cxi:lnx"`
  - `PRTE_MCA_ras_base_launch_orted_on_hn=1`
    - suggested in the OpenMPI documentation page https://docs.open-mpi.org/en/main/tuning-apps/networking/ofi.html#what-are-the-libfabric-ofi-components-in-open-mpi
  - (NVIDIA) `FI_HMEM_CUDA_USE_GDRCOPY=1`

Configuration – Running (2)

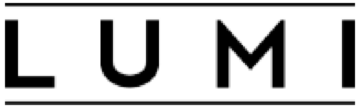
- Other configurations for comparison:
  - **OpenMPI+SHM**:
    - `OMPI_MCA_pml=ob1`
    - `OMPI_MCA_btl=self,sm`
  - **OpenMPI+CXI**:
    - `OMPI_MCA_pml=cm`
    - `OMPI_MCA_mtl=ofi`
- We consider: H2H (host-to-host) and G2G (GPU-to-GPU) communications
- All tests used the **default settings**, i.e. we don’t use any specific setting for improving performance for either OpenMPI and CrayMPI runs

Performance – Synthetic benchmarks

- OSU benchmark v7.5 (https://mvapich.cse.ohio-state.edu/benchmarks/)
- MPI Point-to-Point (P2P) test:
  - Bidirectional bandwidth: `osu_bibw`
  - We consider 1 node (intra-node) and 2 nodes (inter-node) communications
- MPI collectives tests: `alltoall, allgather`
  - We run using 4 (Helios) and 8 (LUMI) MPI processes per node
  - 4, 16, 64 nodes
- OpenMPI and CrayMPI tests are executed on the same set of nodes
- We validate all results (`--validation` flag)
- MPI processes are mapped by L3cache and bound to cores and have exclusive access to the closest GPU by setting the proper `{HIP|CUDA}_VISIBLE_DEVICES` variable
- CrayMPI results are used as a baseline measure (OpenMPI / CrayMPI)
  - Bandwidth ratio: >1 means OpenMPI is better
  - Latency ratio: <1 means OpenMPI is better

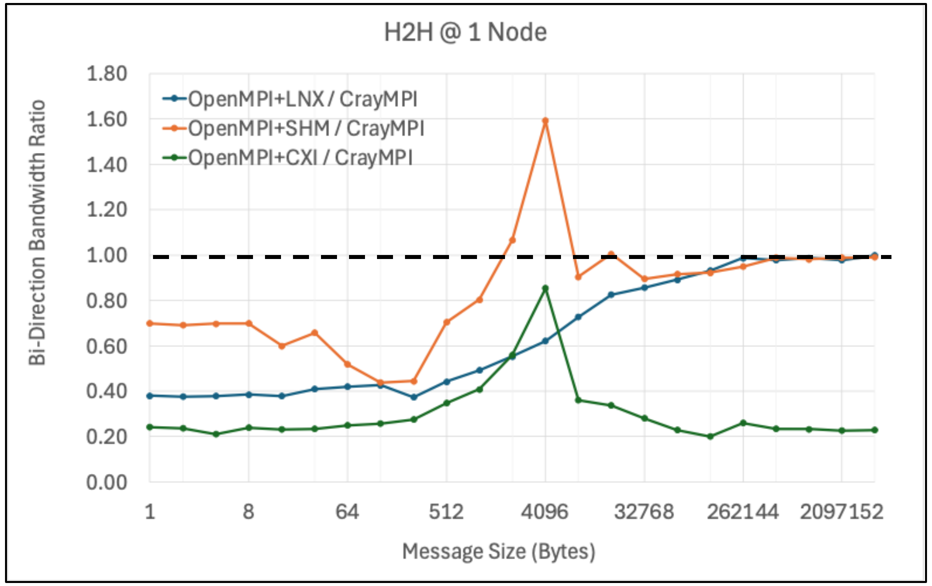
Reminder of the systems we are using

- Logo will appear on the top right corner of the slide with benchmark results

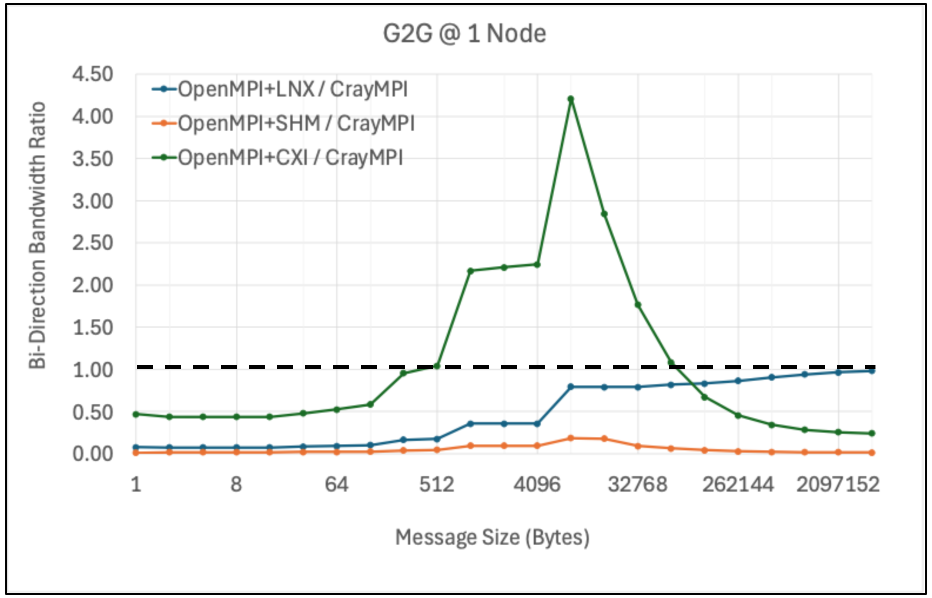


P2P Bidirectional Bandwidth – 1 node

- OpenMPI+LNX<OpenMPI+SHM: LNX overhead?
- OpenMPI+LNX poor performance for small messages
- OpenMPI+LNX==CrayMPI for large messages

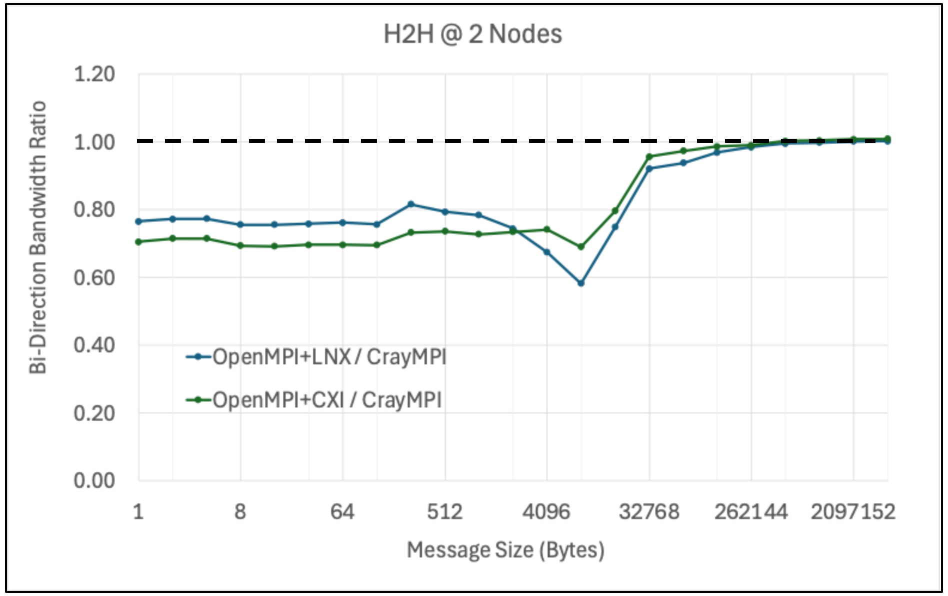


- OpenMPI+LNX>OpenMPI+SHM: SHM is not optimized for G2G
- CrayMPI uses a combination of H2H for small message sizes (default threshold is 1024 bytes), GPU-IPC, and other specific GPU optimizations for large messages
- OpenMPI+CXI outperforms CrayMPI when CrayMPI starts to use GPU-IPC (>1024 bytes)
- OpenMPI+LNX poor performance for small messages
- OpenMPI+LNX==CrayMPI for large messages

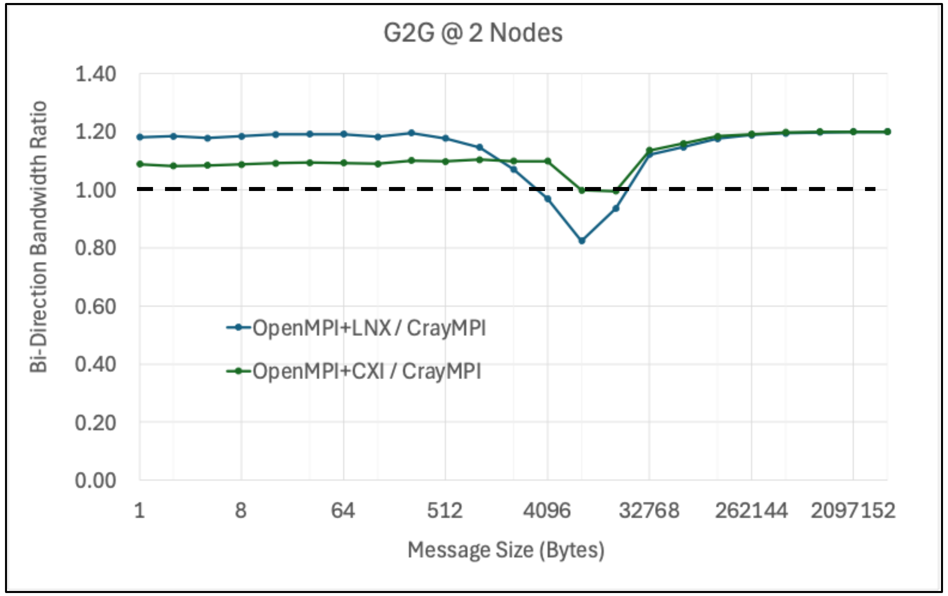


P2P Bidirectional Bandwidth – 2 nodes

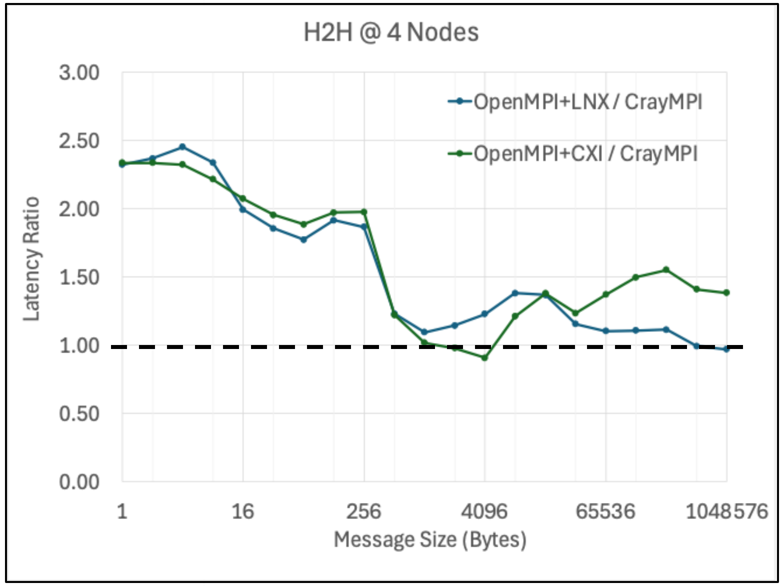
- OpenMPI+LNX poor performance for small messages
- OpenMPI+LNX==CrayMPI for large messages

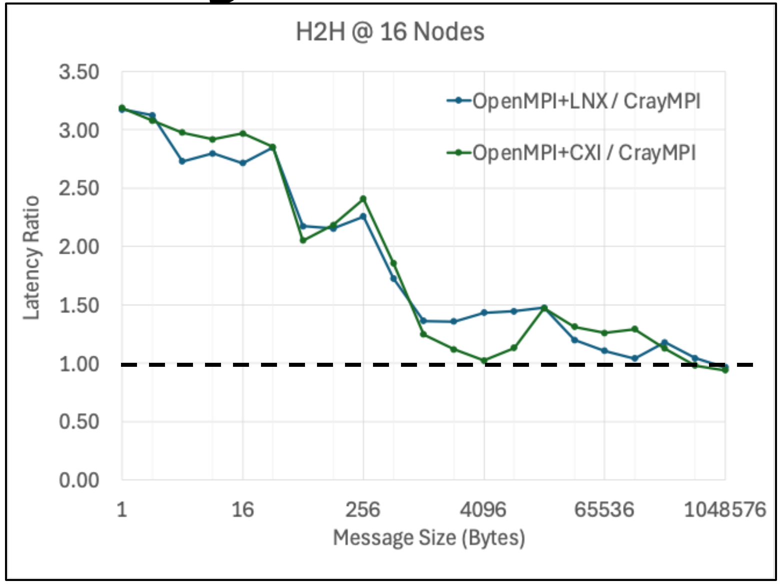


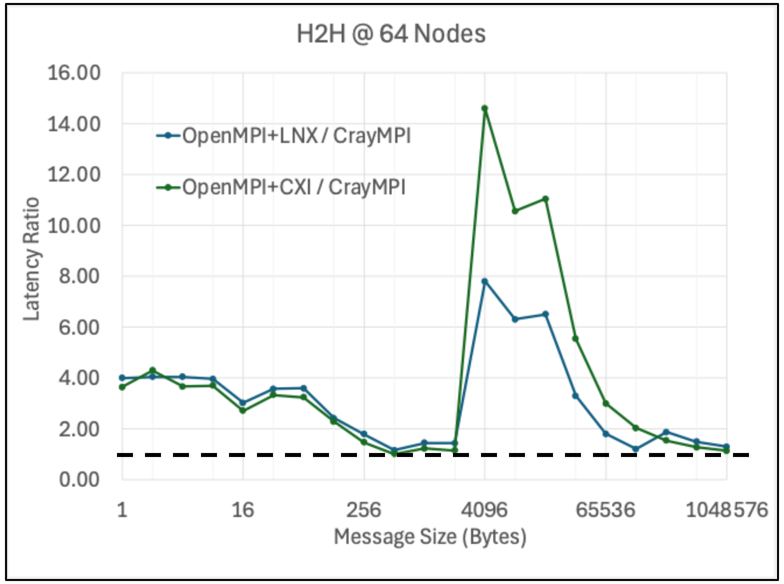
- OpenMPI+LNX outperforms CrayMPI

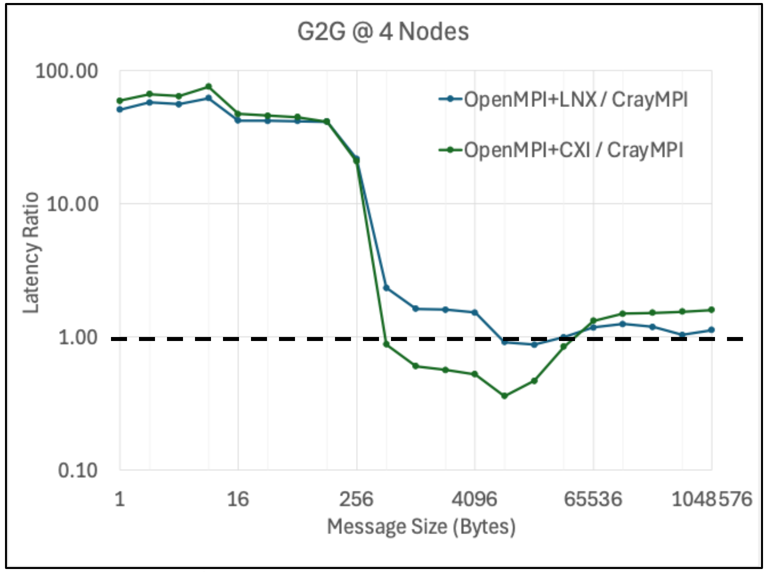


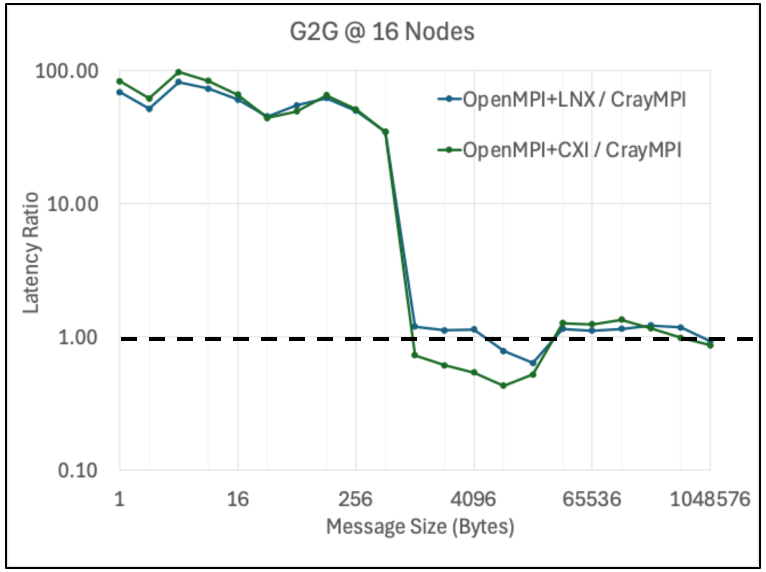
Alltoall Latency

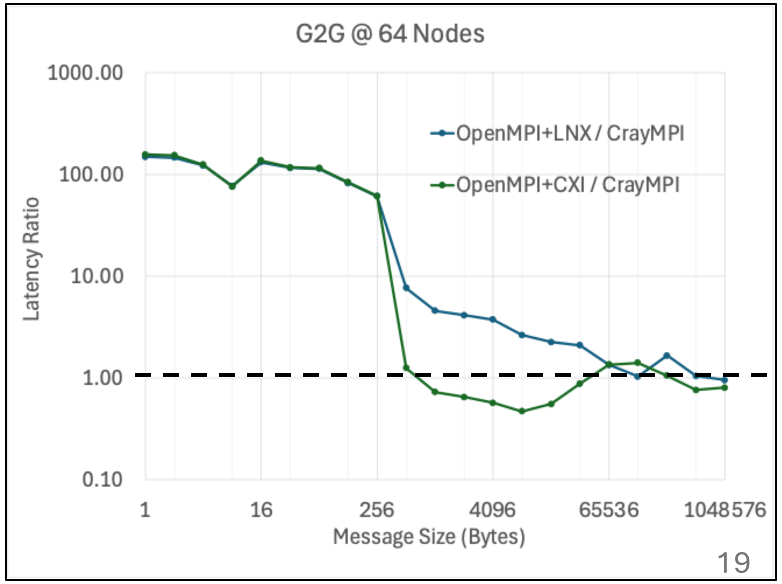




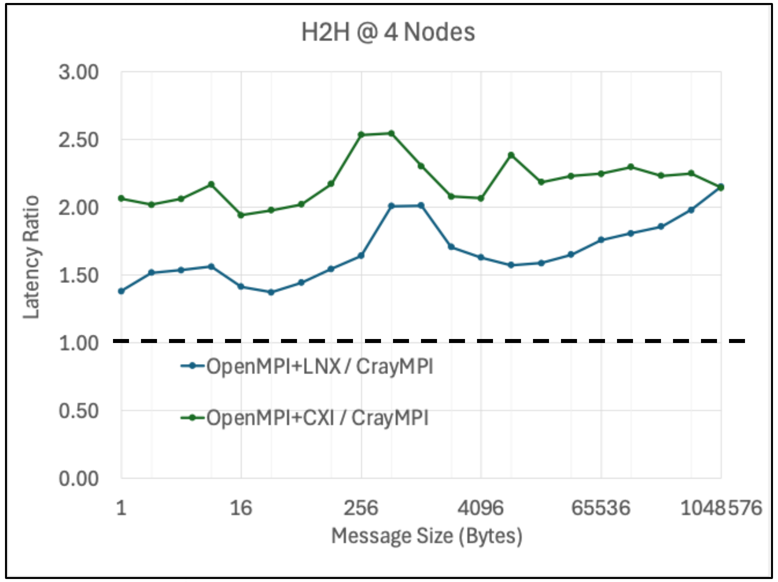


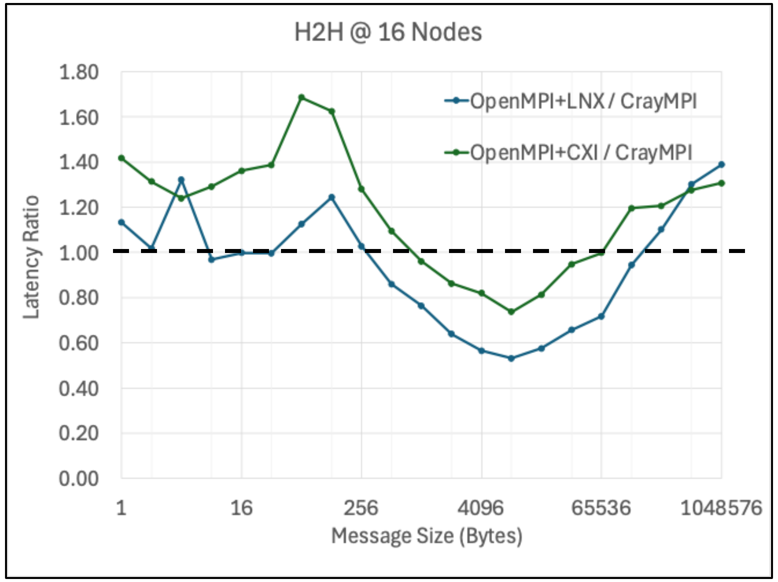


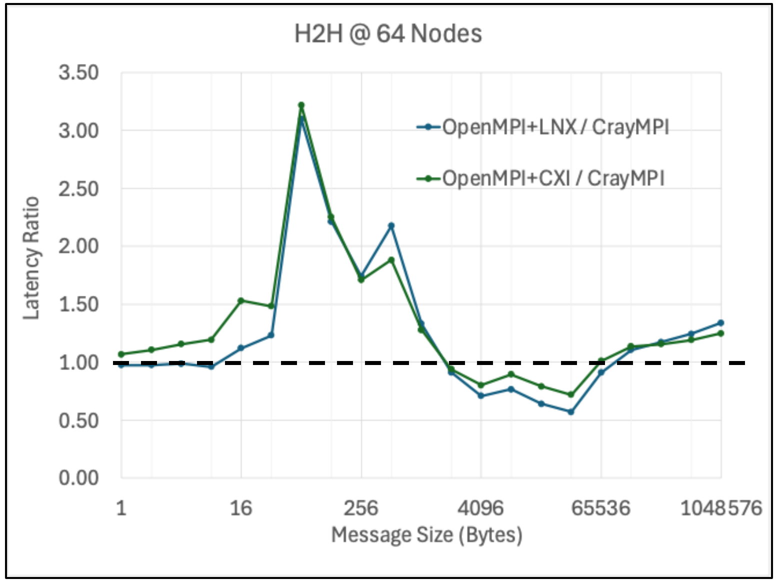


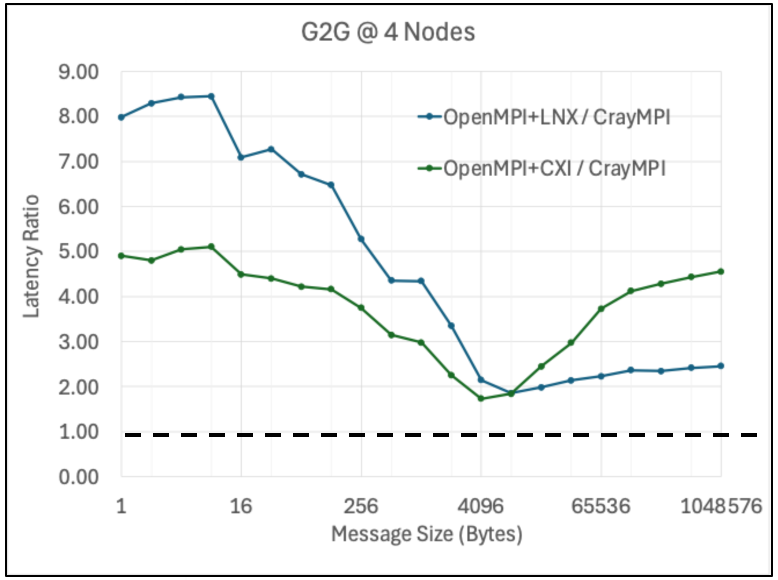


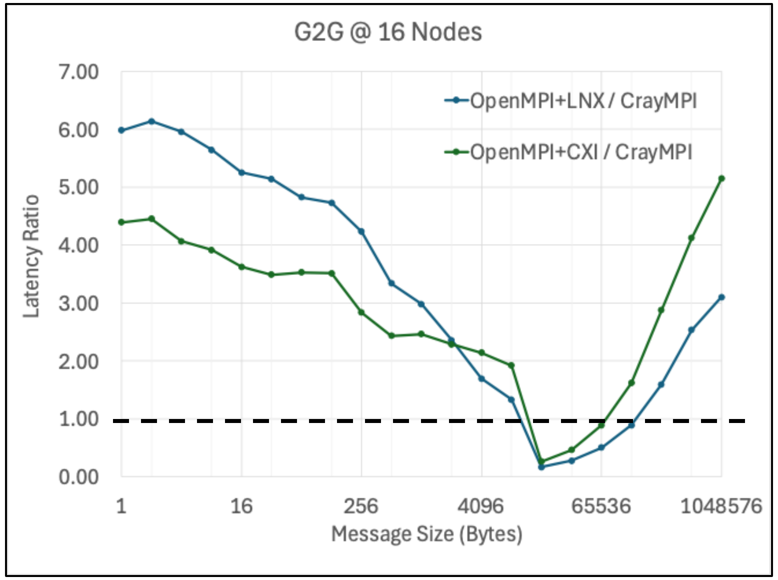
Allgather Latency

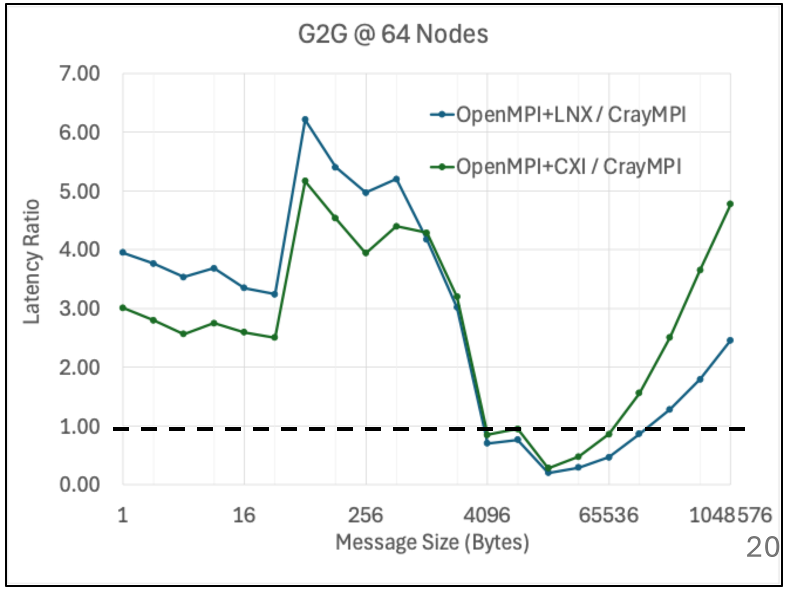












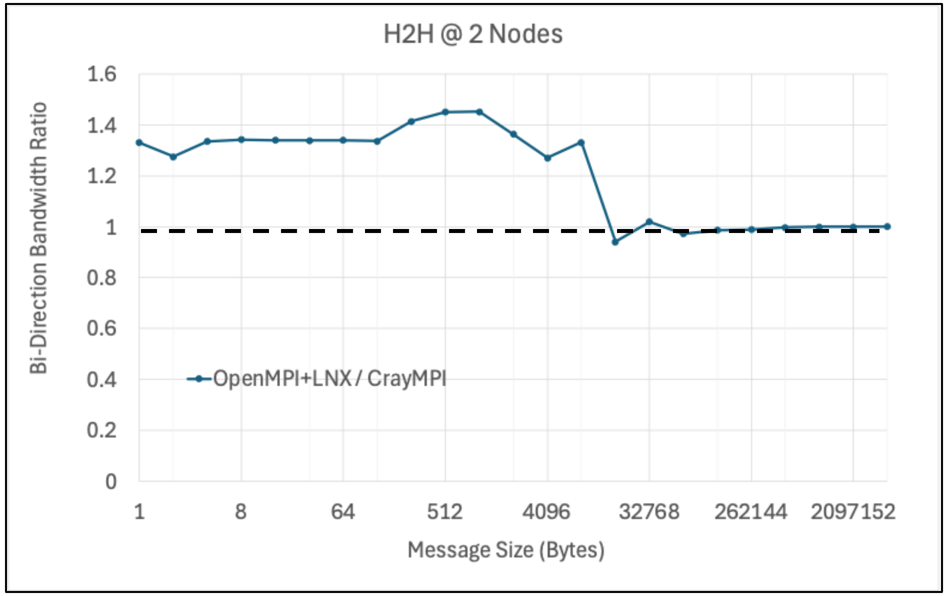
P2P Bidirectional Bandwidth

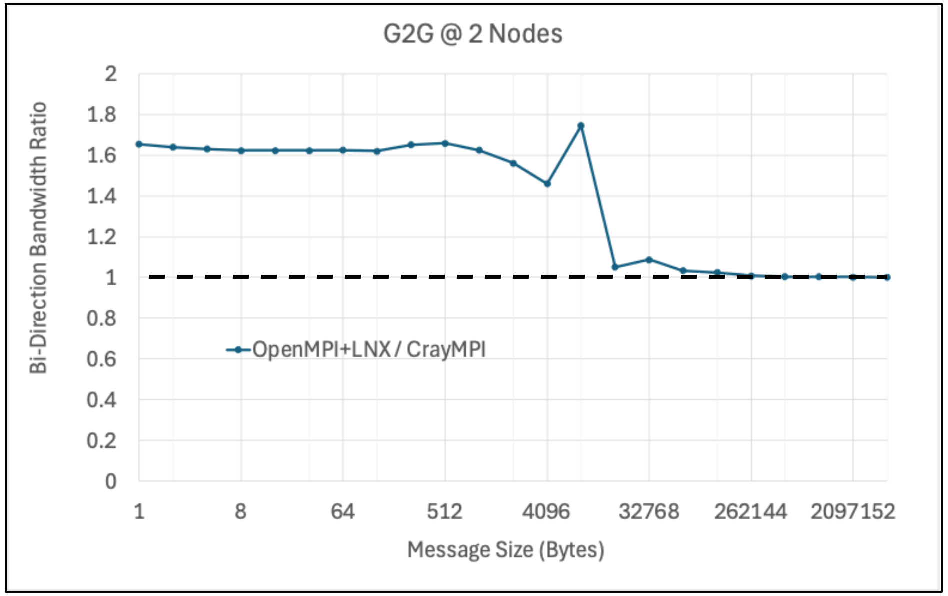
- OpenMPI+LNX>=CrayMPI

BUT

- CrayMPI uses a version of OFI libfabric, which is not optimized for NVIDIA GH nodes
- Some open issues for OFI Libfabric/OpenMPI on GH200 nodes:
  - https://github.com/ofiwg/libfabric/issues/10865
  - https://github.com/open-mpi/ompi/issues/13156

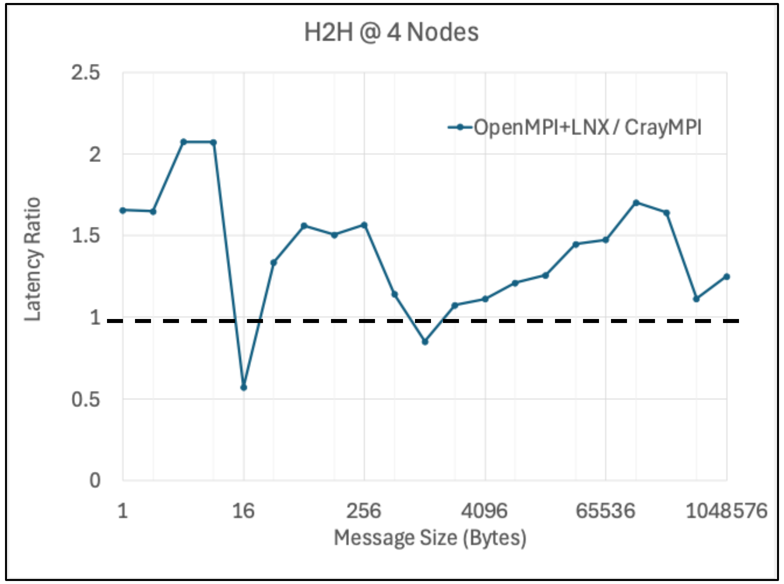
➔ Preliminary results, more tests are needed with new software stacks!

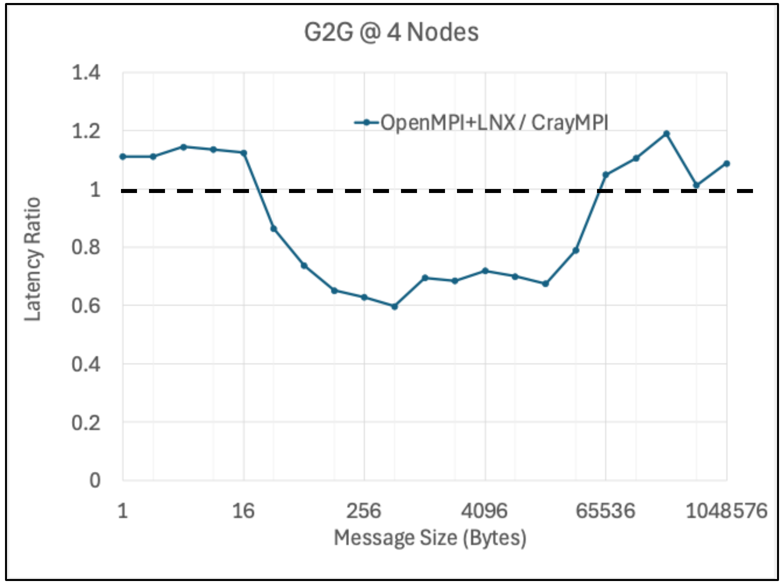


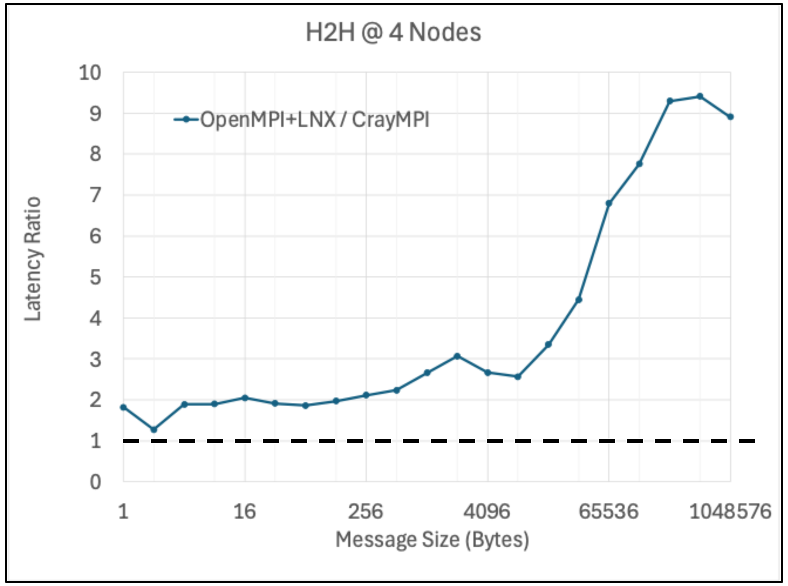


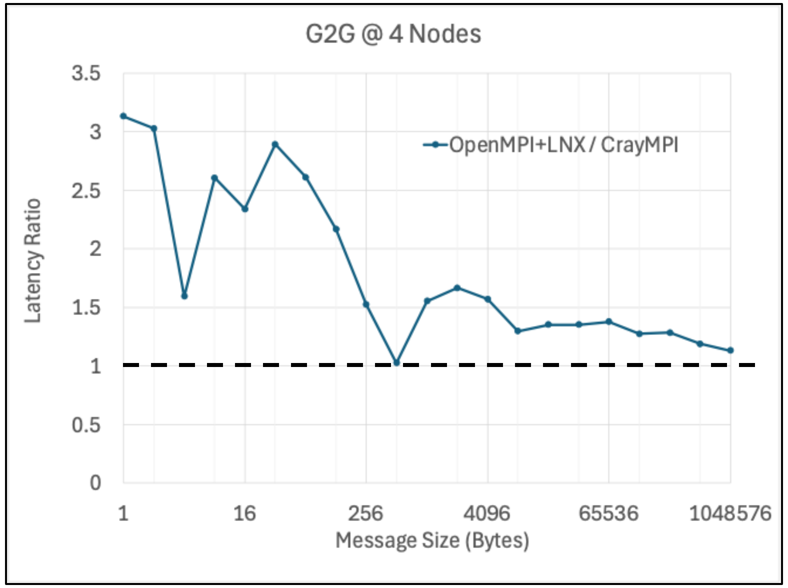
Alltoall and Allgather Latency

Alltoall





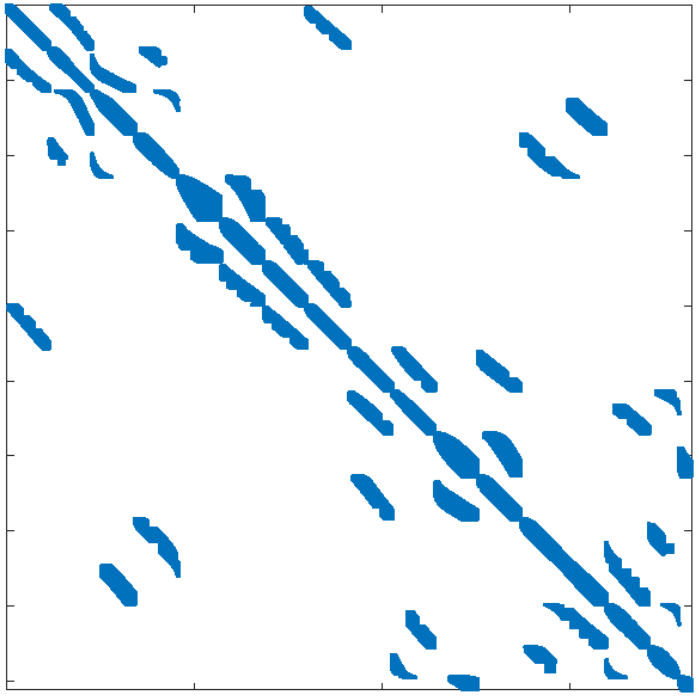




Allgather

Performance - Applications

- Dirac equation solver
  - Discretization: unstructured sparse matrix
  - Partitioning: Metis
- Main compute part
  - BiCGSTAB with ILU(0) preconditioner
  - Parallel Sparse Matrix – Vector multiplication (SpMV)
  - Data and computations fully on GPUs
- Communication:
  - Point-to-point (`MPI_Isend / MPI_Irecv`) in SpMV
  - Collectives (`MPI_Allreduce`) in BiCGSTAB
- Communication implementation:
  - Impl. 1: `MPI_Type_indexed`: MPI library does packing / unpacking
  - Impl. 2: Explicit pack / unpack GPU kernels



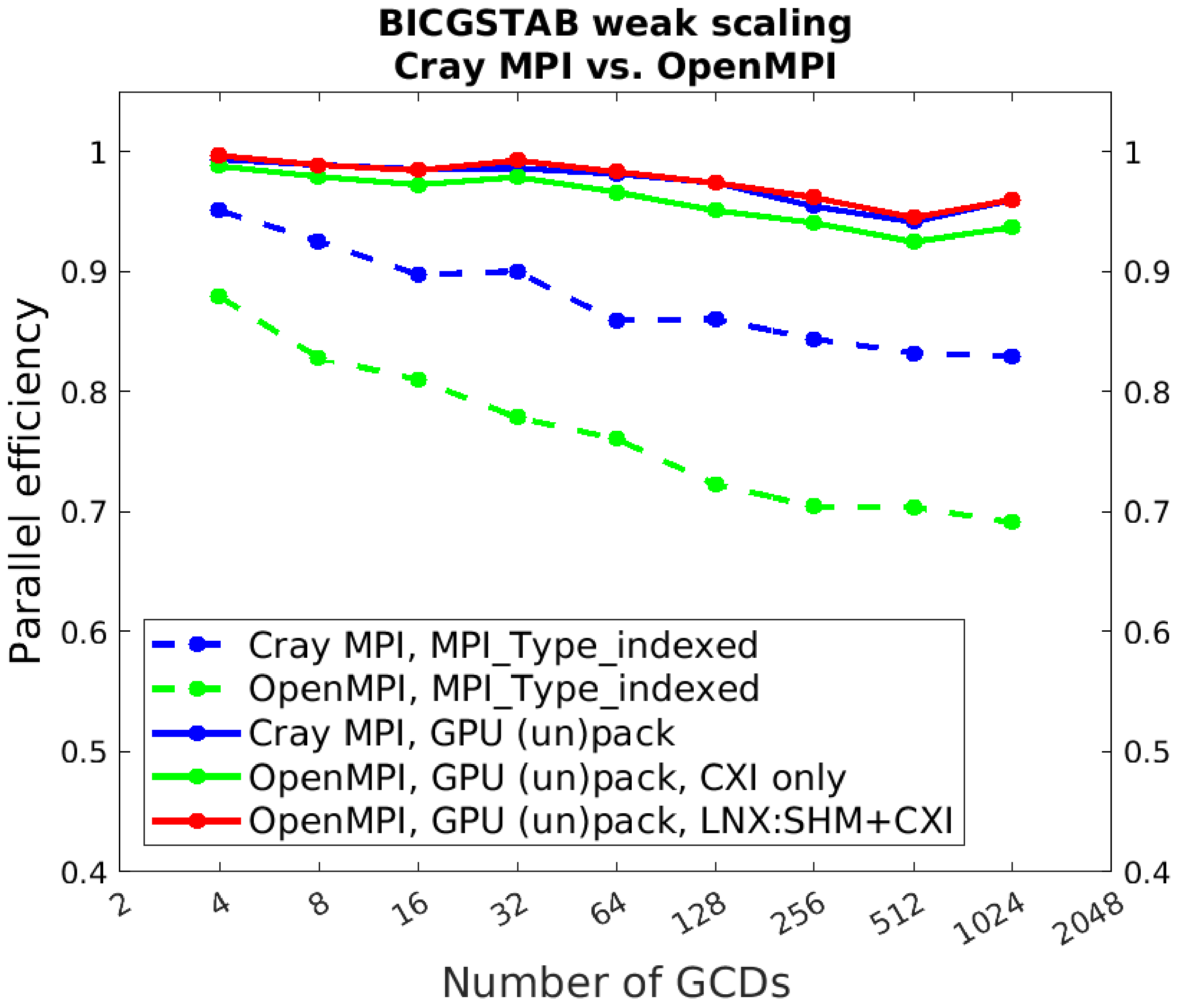
Performance - Applications

- 8 MPI ranks/node
  - 1 rank per GCD

- Indexed types are slow
  - Cray MPI better

- `MPI_Isend` with explicit (un)pack works well
  - OpenMPI with LNX has similar performance to Cray MPI

- LNX does improve the performance over CXI-only



Containers

- Build a container with OpenMPI 5.0
  - Simple case: use package distribution, e.g. Ubuntu 25.04

- Run the container and bind the host OpenMPI at runtime to replace the container OpenMPI (*hybrid mode*, standard technique in HPC container community)
  - Replace the shared libraries
  - Requires proper bindings of paths and shared libraries

- Performance is the same as the host OpenMPI installation

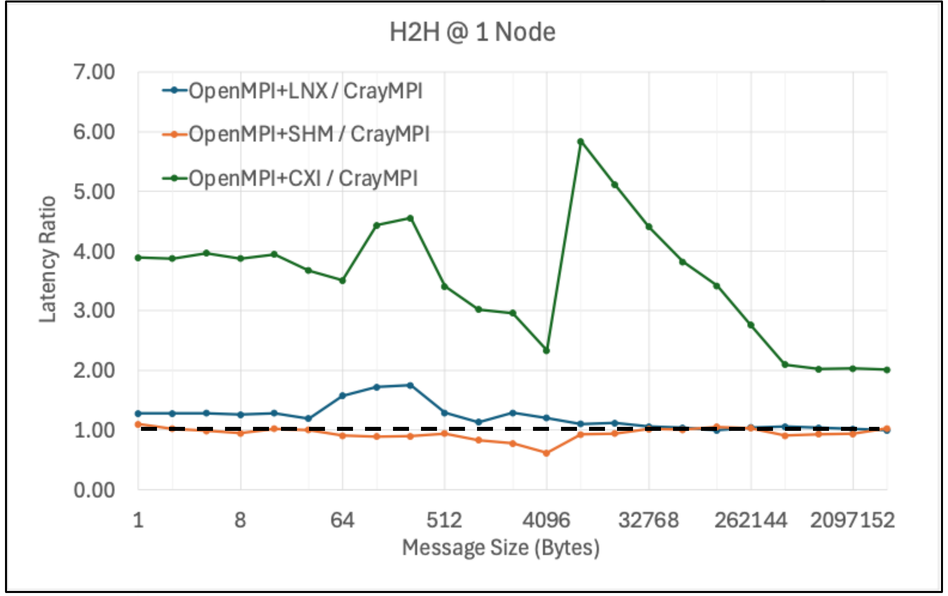
Conclusion and Future Work

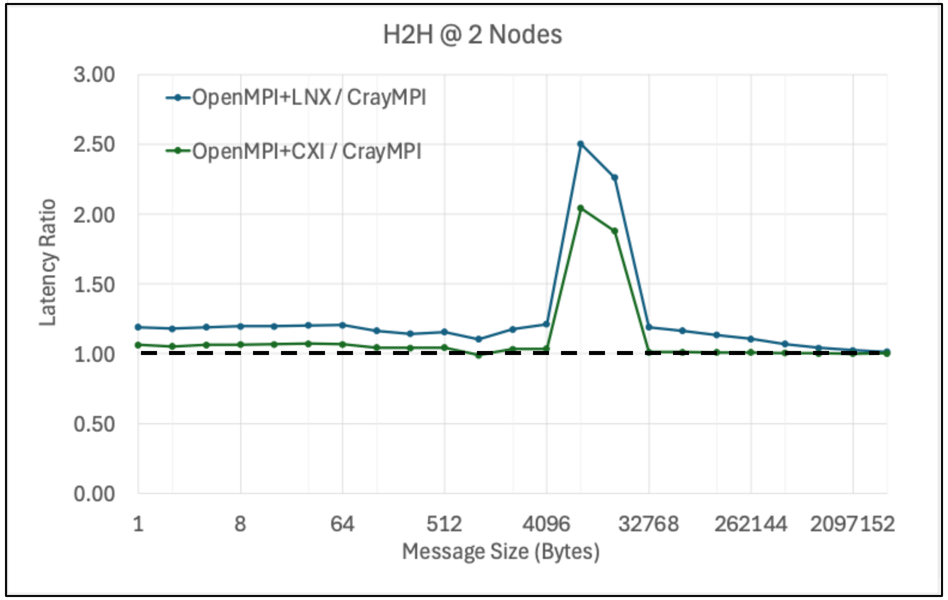
- OpenMPI with OFI Libfabric + LNX provider can improve the performance over the CXI-only provider
  - Main performance degradation for small message exchanges
  - Comparable to CrayMPICH performance for large message sizes
- Bonus of the OpenMPI host-installation: we can run containers with OpenMPI

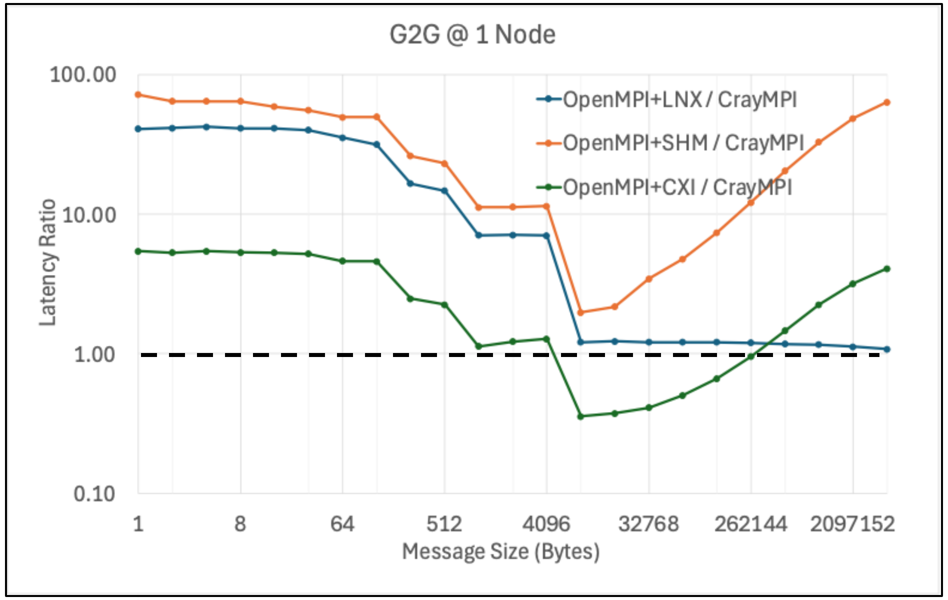
- **Future work:**
  - Rerun NVIDIA results with new software stacks
  - Experimenting with environment variables for tuning performance
  - Tests other MPI implementations (MPICH and MVAPICH)
  - Test with other compilers (CCE / AMD / NVIDIA)

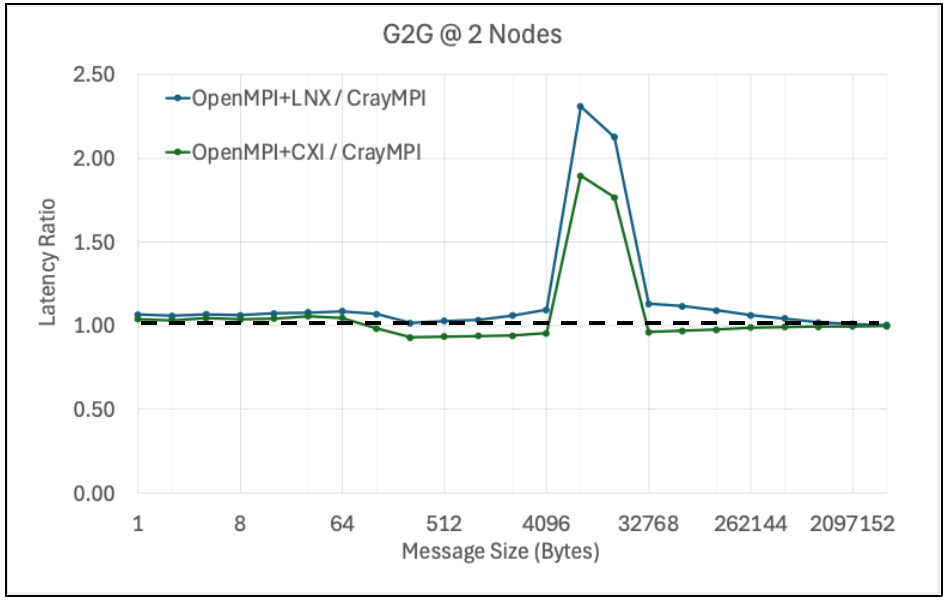
Backup Slides

P2P Latency


H2H @ 1 Node
OpenMPI+LNX / CrayMPI
OpenMPI+SHM / CrayMPI
OpenMPI+CXI / CrayMPI
Latency Ratio
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
1
8
64
512
4096
32768
262144
2097152
Message Size (Bytes)


H2H @ 2 Nodes
OpenMPI+LNX / CrayMPI
OpenMPI+CXI / CrayMPI
Latency Ratio
3.00
2.50
2.00
1.50
1.00
0.50
0.00
1
8
64
512
4096
32768
262144
2097152
Message Size (Bytes)


G2G @ 1 Node
OpenMPI+LNX / CrayMPI
OpenMPI+SHM / CrayMPI
OpenMPI+CXI / CrayMPI
Latency Ratio
100.00
10.00
1.00
0.10
1
8
64
512
4096
32768
262144
2097152
Message Size (Bytes)


G2G @ 2 Nodes
OpenMPI+LNX / CrayMPI
OpenMPI+CXI / CrayMPI
Latency Ratio
2.50
2.00
1.50
1.00
0.50
0.00
1
8
64
512
4096
32768
262144
2097152
Message Size (Bytes)