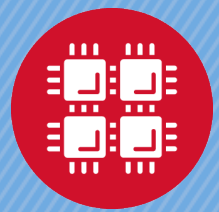
Optimizing Amber for Device-to-Device GPU Communication

Samuel Khuvis[1], Karen Tomko[1], Scott R. Brozell[1], Chen-Chun Chen[2], Hari Subramoni[2], Dhabaleswar K. Panda[2]

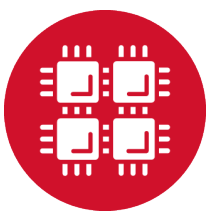
[1]Ohio Supercomputer Center, [2]Ohio State University

MUG 2023

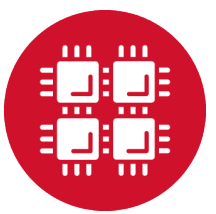


Outline

- Introduction
- Experimental Setup
- Performance Analysis
- Code modifications
- Results
- Summary

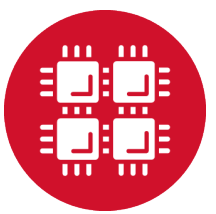


Introduction



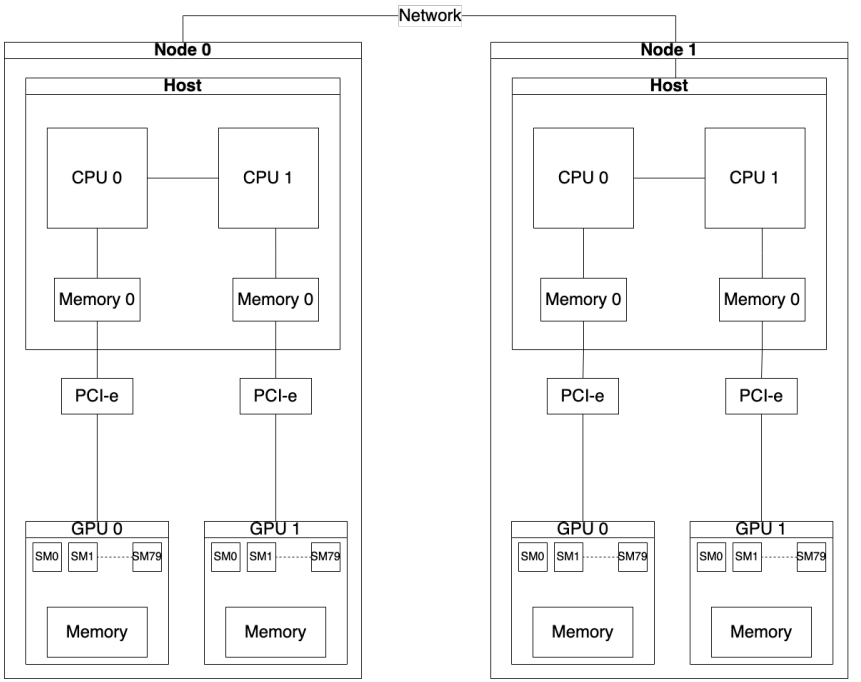
Overview of Amber

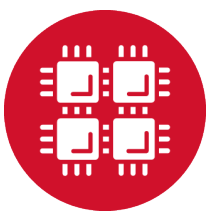
- Widely-used suite for MD simulations of proteins and nucleic acids.
- Supports particle-mesh Ewald (PME) explicit solvents and Generalized Born implicit solvents.
- Added GPU support in version 11.
- Multi-GPU implementations in Amber are done by passing data from GPU buffer to host buffer, performing MPI communication, and then passing back to GPU buffers.



GPU support in MPI

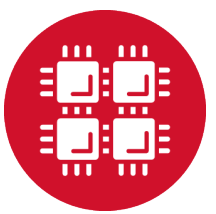
- Capability to specify device buffers added to MVAPICH2 in 2011 and HPC-X OpenMPI v1.7.0 in 2013.
- If appropriate hardware/software is available, communication is performed between GPUs without host transfers.
- If unavailable, MPI performs interprocess communication (IPC) via host-to-device and device-to-host communication.
- Additional data copying affects performance.



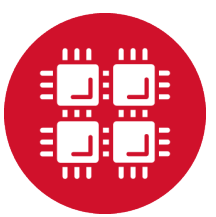


Availability of Device-to-Device Technologies

- Hardware has only started to become available at HPC centers in the last few years.
- Many scientific codes have been slow to take advantage.
- GROMACS only added support for GPU-to-GPU communication in its 2022 release.



Experimental Setup



Benchmarks

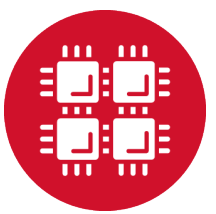
Benchmarks from the Amber20 Benchmark Suite:

- Particle-mesh Ewald (PME)
  - Cellulose production
  - FactorIX production
  - JAC production
  - STMV production
- Generalized Born (GB)
  - Myoglobin
  - Nucleosome

For each PME case, we run with the statistical ensembles:

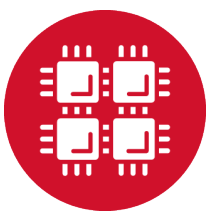
- **NVE** holds total number of particles constant; total energy (E) and volume (V) are conserved,
- **NPT** holds total number of particles constant; pressure (P) and temperature (T) are conserved.

There is a modest computational cost to pressure and temperature control in NPT in comparison to the more straightforward NVE.

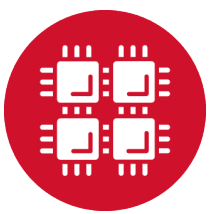


System Description

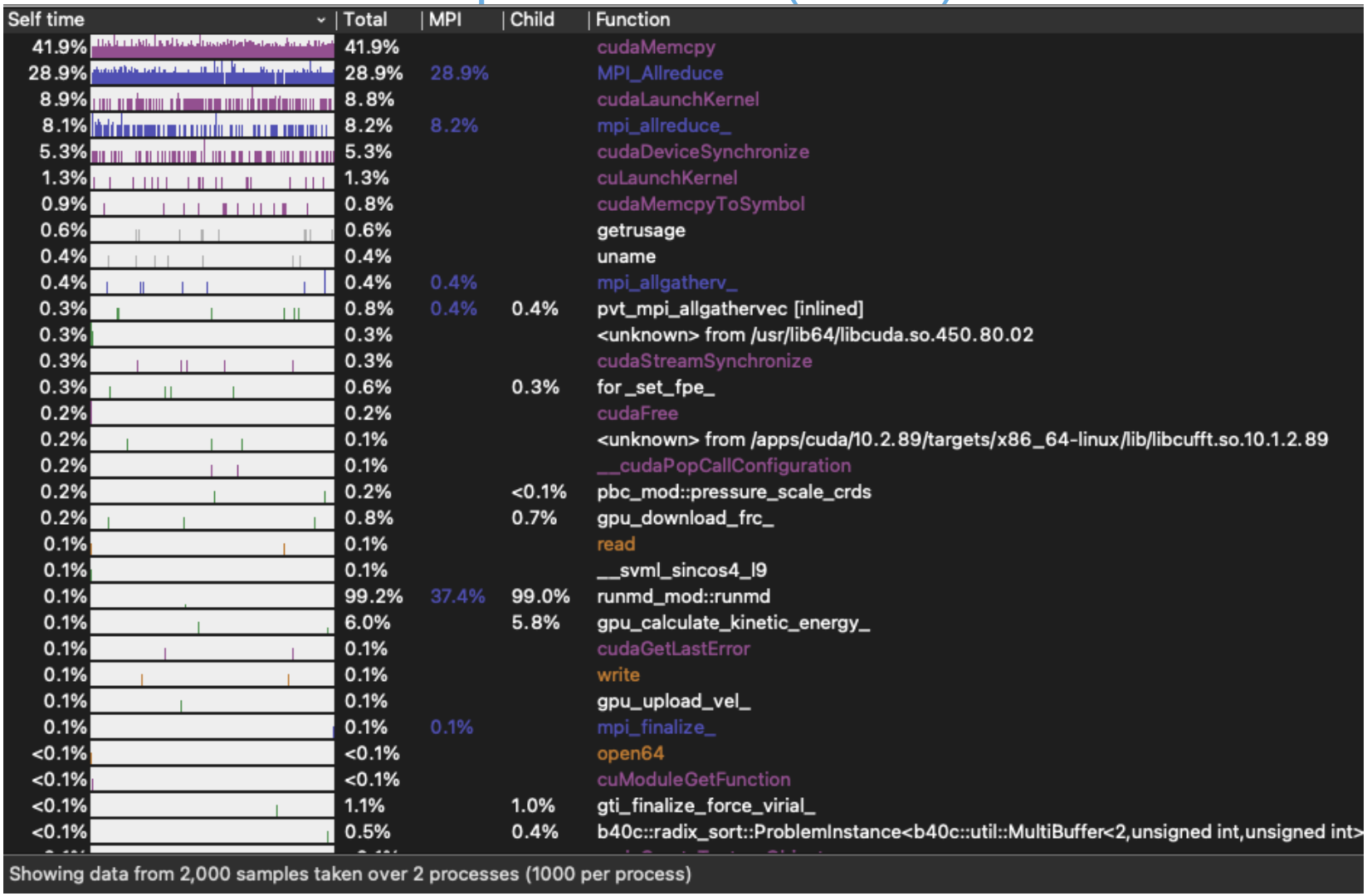
- Pitzer system at OSC
- 48-core cascade lake nodes with 2 NVIDIA V100 GPUs
- Ran with 2 processes per node and 2 GPUs per node on 2, 4, and 8 nodes.
- Mellanox EDR (100Gbps) Infiniband
- MPI implementations:
  - MVAPICH2 2.3.6
  - MVAPICH2-GDR 2.3.7 (cuda-aware)
  - HPC-X OpenMPI 3.1.6 (cuda-aware)



Performance Analysis



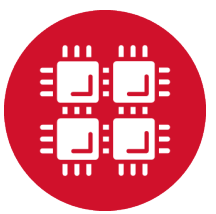
MAP Profile of JAC production (NPT)



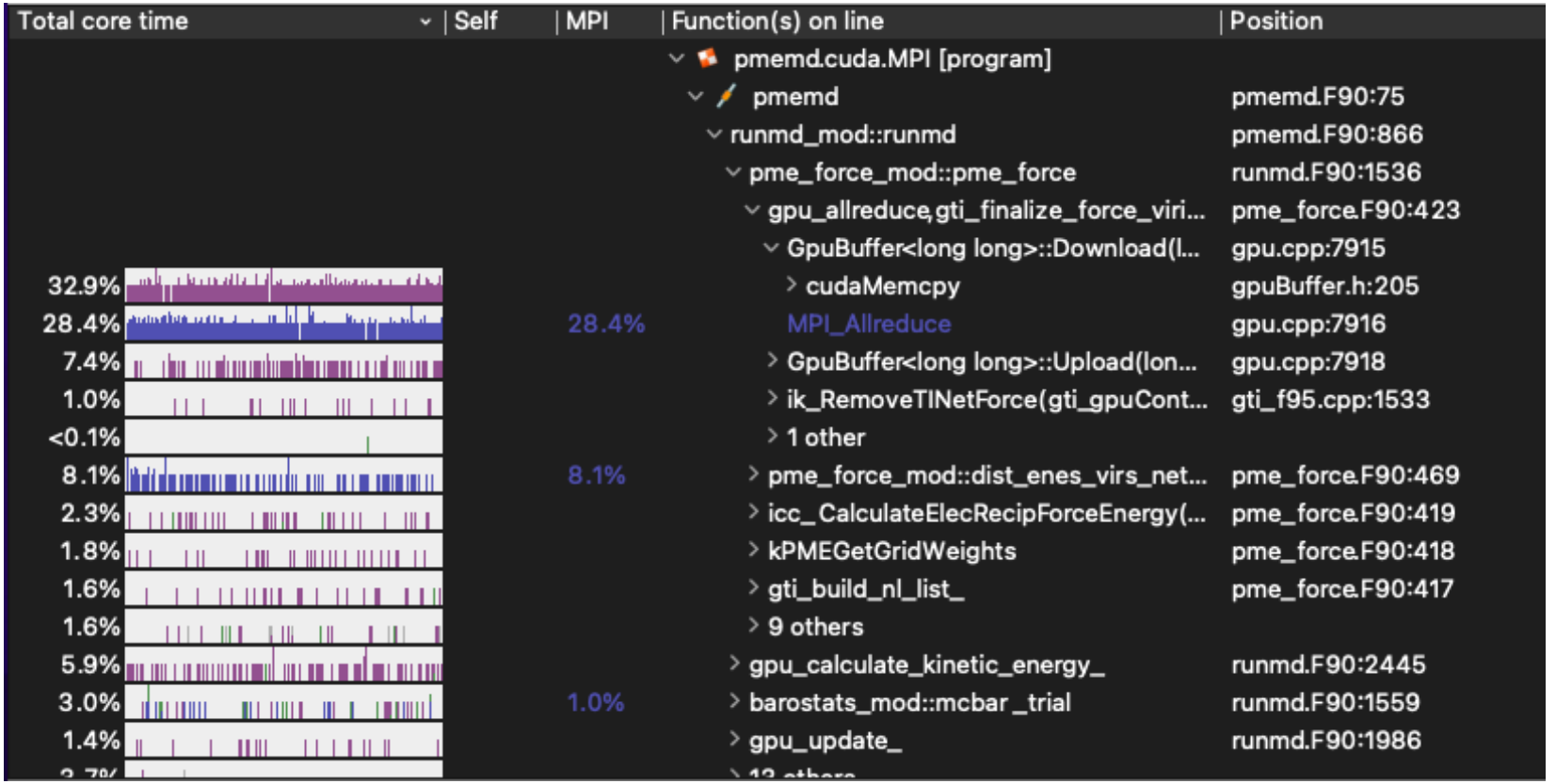
Self time	Total	MPI	Child	Function
41.9%	41.9%			cudaMemcpy
28.9%	28.9%	28.9%		MPI_Allreduce
8.9%	8.8%			cudaLaunchKernel
8.1%	8.2%	8.2%		mpi_allreduce_
5.3%	5.3%			cudaDeviceSynchronize
1.3%	1.3%			cuLaunchKernel
0.9%	0.8%			cudaMemcpyToSymbol
0.6%	0.6%			getrusage
0.4%	0.4%			uname
0.4%	0.4%	0.4%		mpi_allgatherv_
0.3%	0.8%	0.4%	0.4%	pvt_mpi_allgathervec [inlined]
0.3%	0.3%			<unknown> from /usr/lib64/libcuda.so.450.80.02
0.3%	0.3%			cudaStreamSynchronize
0.3%	0.6%		0.3%	for_set_fpe_
0.2%	0.2%			cudaFree
0.2%	0.1%			<unknown> from /apps/cuda/10.2.89/targets/x86_64-linux/lib/libcufft.so.10.1.2.89
0.2%	0.1%			__cudaPopCallConfiguration
0.2%	0.2%		<0.1%	pbc_mod::pressure_scale_crds
0.2%	0.8%		0.7%	gpu_download_frc_
0.1%	0.1%			read
0.1%	0.1%			__svml_sincos4_l9
0.1%	99.2%	37.4%	99.0%	runmd_mod::runmd
0.1%	6.0%		5.8%	gpu_calculate_kinetic_energy_
0.1%	0.1%			cudaGetLastError
0.1%	0.1%			write
0.1%	0.1%			gpu_upload_vel_
0.1%	0.1%	0.1%		mpi_finalize_
<0.1%	<0.1%			open64
<0.1%	<0.1%			cuModuleGetFunction
<0.1%	1.1%		1.0%	gti_finalize_force_virial_
<0.1%	0.5%		0.4%	b40c::radix_sort::ProblemInstance<b40c::util::MultiBuffer<2,unsigned int,unsigned int>

Showing data from 2,000 samples taken over 2 processes (1000 per process)

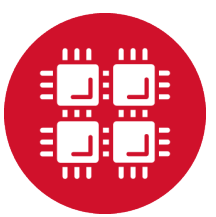
- Significant time spent in MPI communication (`MPI_Allreduce`) and device-to-host communication (`cudaMemcpy`).



Callstack of JAC production (NPT)



- Device-to-host communication and MPI communication called from `gpu_allreduce`.

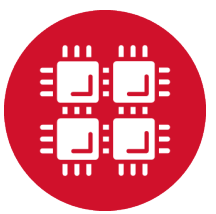


Code modifications

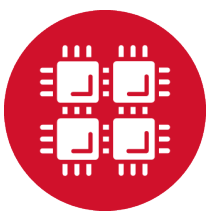
```
cudaMemcpy Device-to-Host
MPI_Allreduce Host-to-Host
cudaMemcpy Host-to-Device
```

```
cudaDeviceSynchronize
MPI_Allreduce Device-to-Device
cudaMemcpy Device-to-Host
```

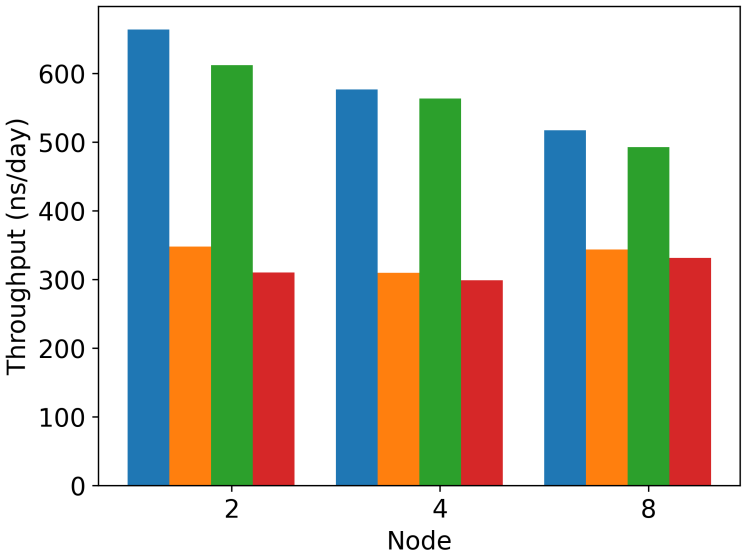
Only 5 lines of code changed.



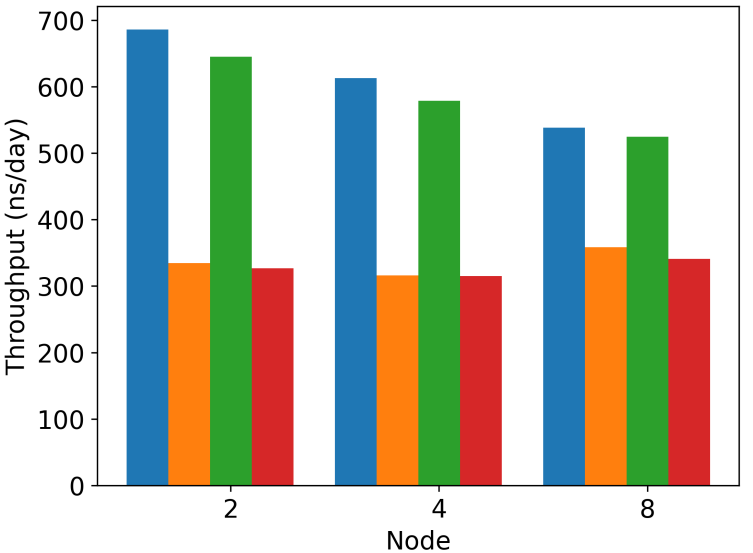
Results



JAC Benchmark



(a) NPT



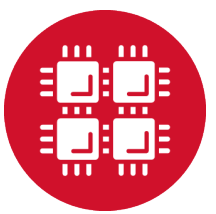
(b) NVE

MV2-GDR (modified code) MV2 (modified code)
HPC-X (modified code) MV2 (original code)

Average Throughputs for All Benchmarks

MPI Implementation	Throughput (ns/day)	Speedup
MVAPICH2 (original code)	192.5	1.00
MVAPICH2 (modified code)	201.4	1.05
HPC-X OpenMPI (modified code)	234.7	1.22
MVAPICH2-GDR (modified code)	262.9	1.36

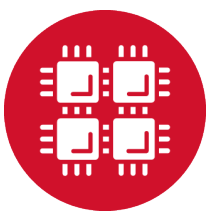
- 36% improvement with GDR over original code.



Average Throughputs for PME Benchmarks

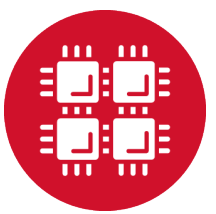
MPI Implementation	Throughput (ns/day)	Speedup
MVAPICH2 (original code)	118.2	1.00
MVAPICH2 (modified code)	120.3	1.02
HPC-X OpenMPI (modified code)	197.2	1.67
MVAPICH2-GDR (modified code)	217.8	1.84

- 84% improvement with GDR over original code.



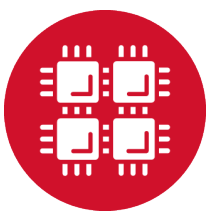
Results

- For the myoglobin benchmark, performance of GDR is degraded compared to reference and modified code with MV2.
- Some benchmarks (Cellulose and STMV) show worse performance with modified code compared to original code.
- From the benchmarks tested, only GB nuclesome shows good scaling.



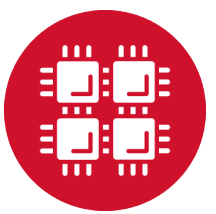
Summary

- Most expensive functions in benchmark runs were `MPI_Allreduce` and `cudaMemcpy` from `gpu_allreduce`.
- Modified `gpu_allreduce` to communicate between GPU buffers, reducing host $\leftrightarrow$ device communication.
- Increases throughput by 36% over all benchmarks and 84% for PME subset.
- Other molecular dynamics (MD) techniques would benefit from scalable multi-GPU capability, such as long time-scale MD, free energy calculations, enhanced sampling, conformational sampling, and drug discovery.



Paper

For more details, read our paper from PEARC:
Samuel Khuvis, Karen Tomko, Scott R. Brozell, Chen-Chun Chen, Hari Subramoni, and Dhabaleswar K. Panda. 2023. Optimizing Amber for Device-to-Device GPU Communication. In Practice and Experience in Advanced Research Computing (PEARC '23), July 23–27, 2023, Portland, OR, USA. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3569951.3597553



OH·TECH

Ohio Technology Consortium

A Division of the Ohio Department of Higher Education

info@osc.edu

osc.edu

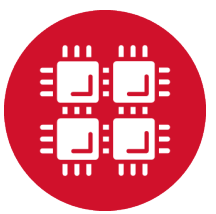
twitter.com/osc

oh-tech.org/blog

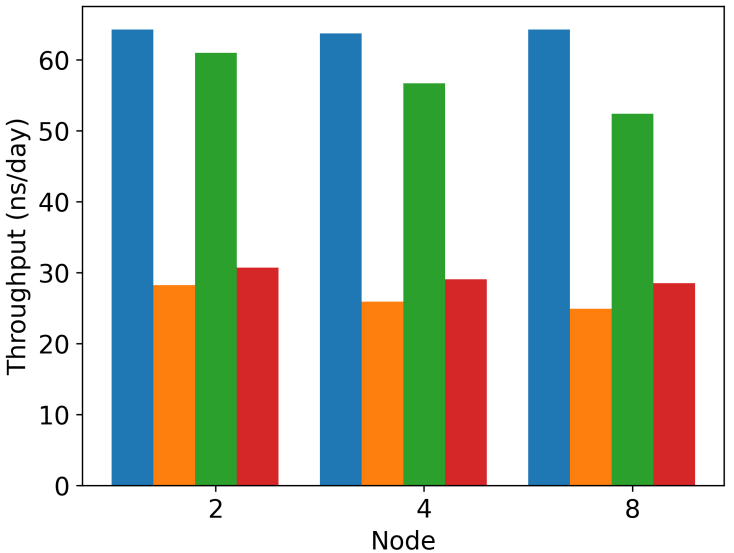
facebook.com/ohiosupercomputercenter

linkedin.com/company/ohio-supercomputer-center

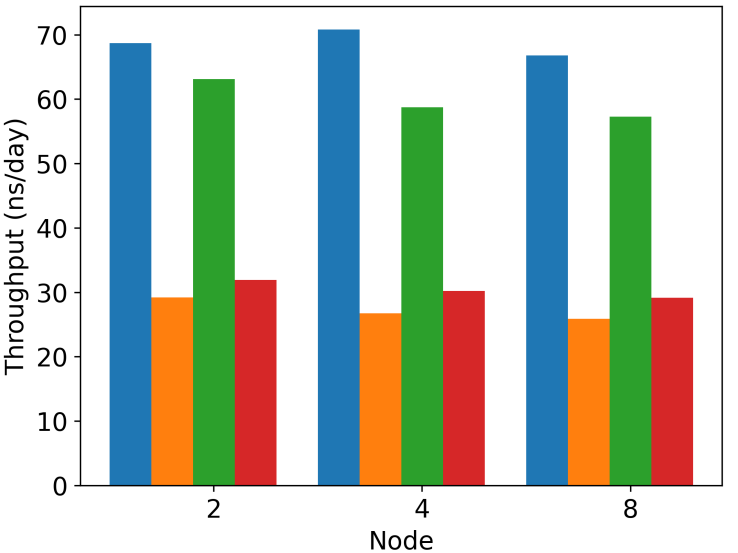
Appendix



Cellulose Benchmark



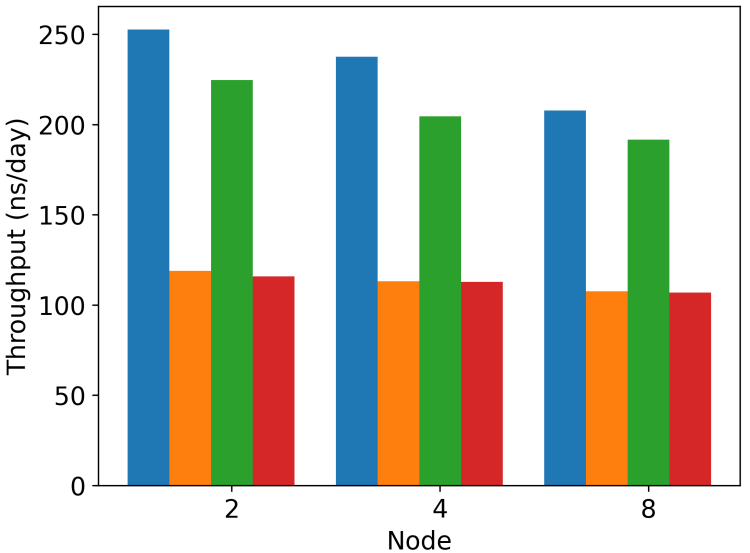
(a) NPT



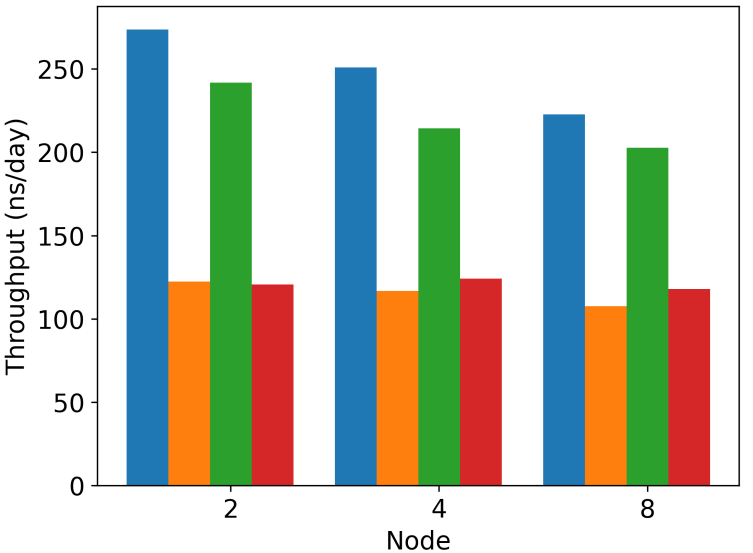
(b) NVE

MV2-GDR (modified code) MV2 (modified code)
HPC-X (modified code) MV2 (original code)

FactorIX Benchmark



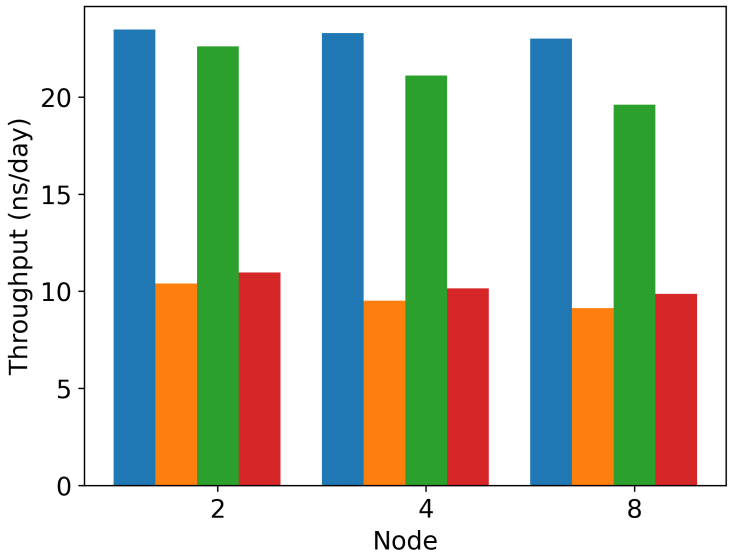
(a) NPT



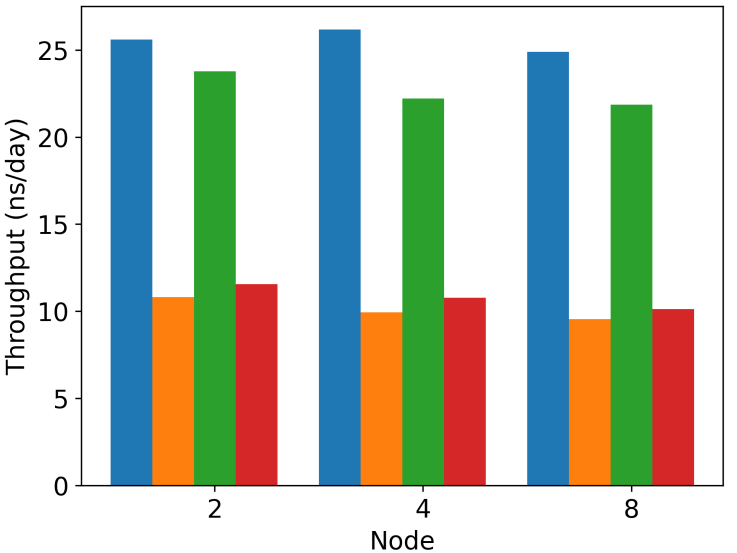
(b) NVE

MV2-GDR (modified code) MV2 (modified code)
HPC-X (modified code) MV2 (original code)

STMV Benchmark



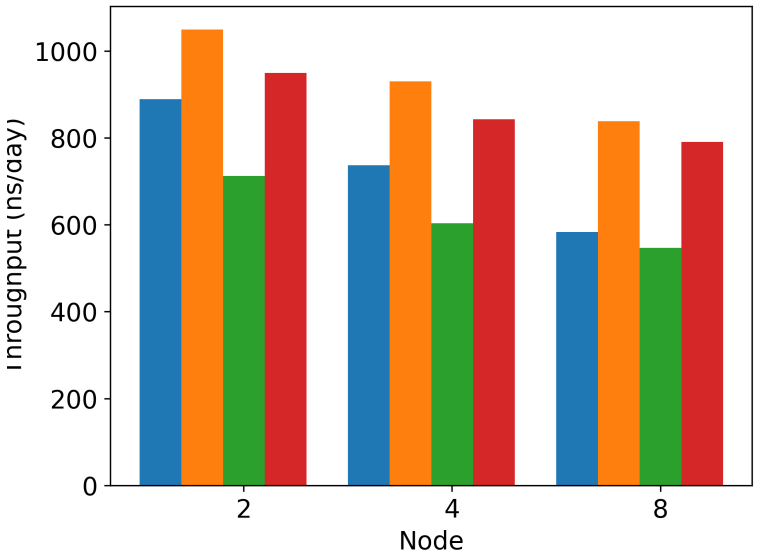
(a) NPT



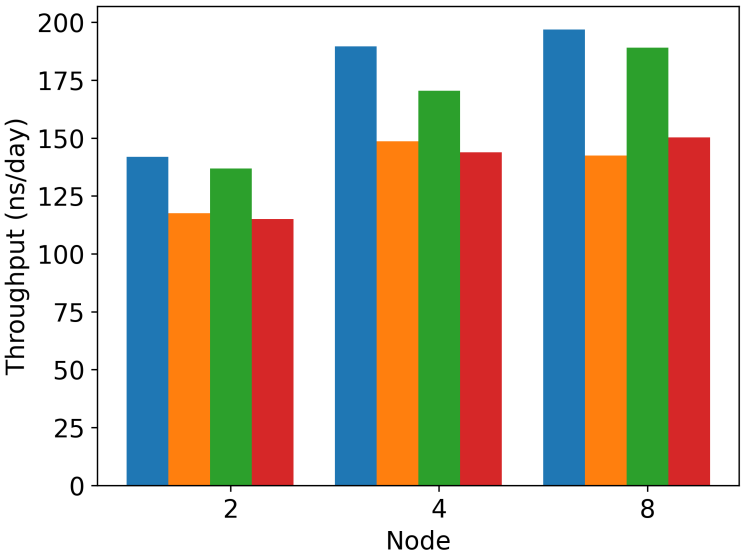
(b) NVE

MV2-GDR (modified code) MV2 (modified code)
HPC-X (modified code) MV2 (original code)

GB Benchmarks



(a) Myoglobin



(b) Nucleosome

MV2-GDR (modified code) MV2 (modified code)
HPC-X (modified code) MV2 (original code)

Message sizes for 2 MPI ranks

Benchmark	Message size (KB)
JAC	553
Cellulose	9577
FactorIX	2131
STMV	25011
myoglobin	20
nucleosome	197