Application and Micro-benchmark Performance using MVAPICH2-X on SDSC Gordon Cluster

Mahidhar Tatineni (mahidhar@sdsc.edu)

MVAPICH User Group Meeting

August 27, 2014

NSF grants: OCI #0910847 Gordon: A Data Intensive Supercomputer

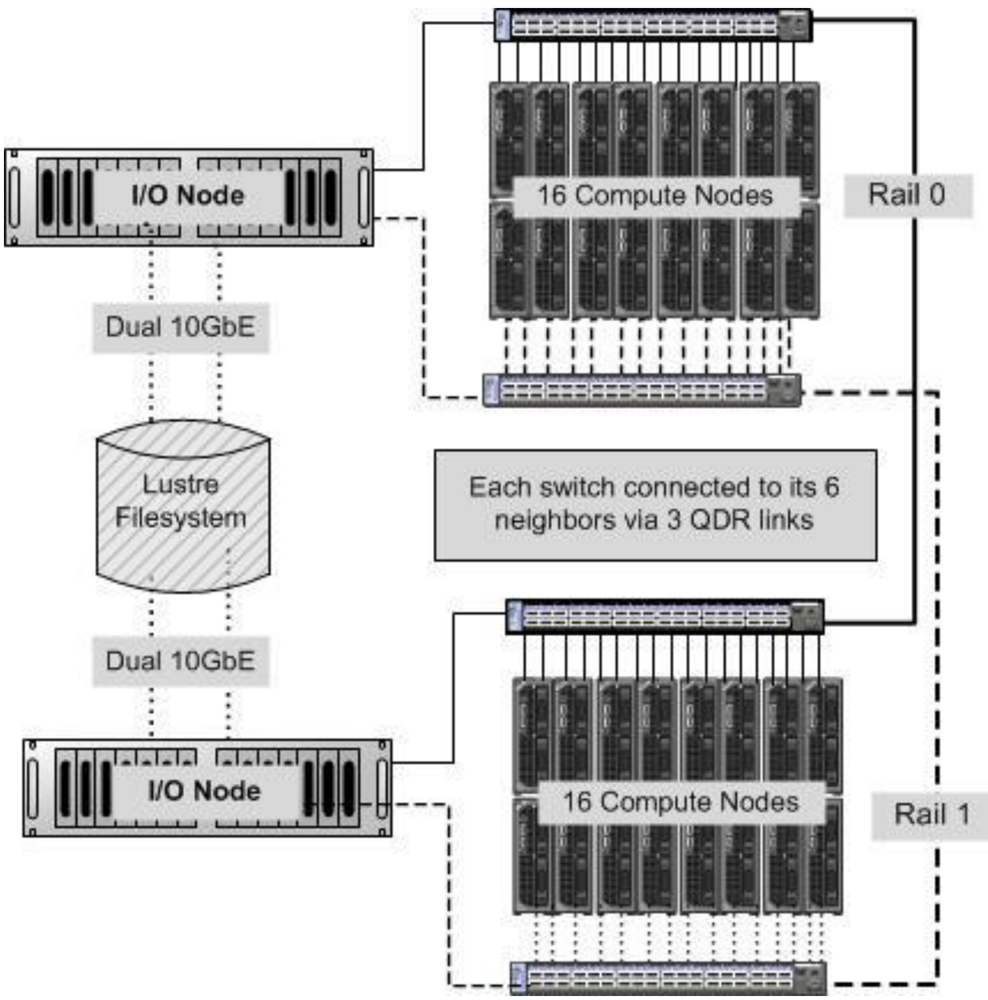
CCF-1213084 Unified Runtime for Supporting Hybrid Programming Models on Heterogeneous Architecture

Gordon – A Data Intensive Supercomputer

- Designed to accelerate access to massive amounts of data in areas of genomics, earth science, engineering, medicine, and others
- Appro integrated 1,024 node Sandy Bridge cluster
- 300 TB of high performance Intel flash
- Large memory supernodes via vSMP Foundation from ScaleMP
- 3D torus interconnect from Mellanox
- In production operation since February 2012
- Funded by the NSF and available through the NSF Extreme Science and Engineering Discovery Environment program (XSEDE)

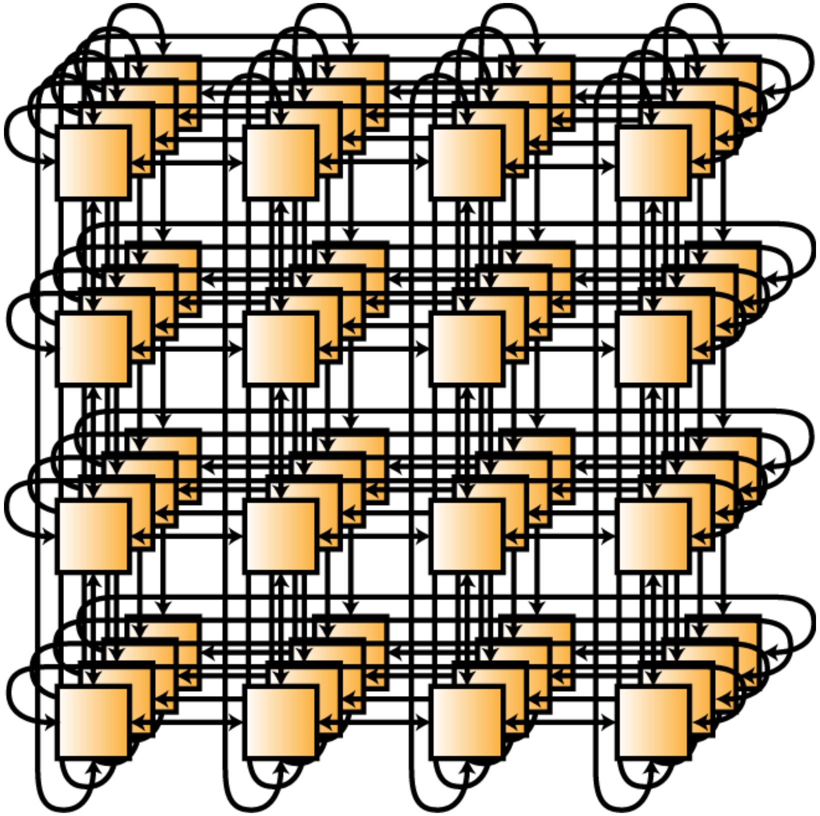
SDSC APPRO NSF intel Mellanox TECHNOLOGIES ScaleMP XSEDE Extreme Science and Engineering Discovery Environment

Subrack Level Architecture



16 compute nodes per subrack

3D Torus of Switches



- Linearly expandable
- Simple wiring pattern
- Short Cables- Fiber Optic cables generally not required
- Lower Cost :40% as many switches, 25% to 50% fewer cables
- Works well for localized communication
- Fault Tolerant within the mesh with 2QoS Alternate Routing
- Fault Tolerant with Dual-Rails for all routing algorithms

3rd dimension wrap-around not shown for clarity

Gordon System Specification

INTEL SANDY BRIDGE COMPUTE NODE	
Sockets	2
Cores	16
Clock speed	2.6
DIMM slots per socket	4
DRAM capacity	64 GB
INTEL FLASH I/O NODE	
NAND flash SSD drives	16
SSD capacity per drive/Capacity per node/total	300 GB / 4.8 TB / 300 TB
Flash bandwidth per drive (read/write) **Flash bandwidth per node (write/read)**	270 MB/s / 210 MB/s 4.3 /3.3 GB/s
SMP SUPER-NODE	
Compute nodes	32
I/O nodes	2
Addressable DRAM	2 TB
Addressable memory including flash	12TB
GORDON	
Compute Nodes	1,024
Total compute cores	16,384
Peak performance	341TF
Aggregate memory	64 TB
INFINIBAND INTERCONNECT	
Aggregate torus BW	9.2 TB/s
Type	Dual-Rail QDR InfiniBand
Link Bandwidth	8 GB/s (bidirectional)
Latency (min-max)	1.25 µs – 2.5 µs
DISK I/O SUBSYSTEM	
Total storage	/oasis/scratch (1.6 PB), /oasis/projects/nsf(1.5PB)
I/O bandwidth	100 GB/s
File system	Lustre

OSU Micro-Benchmark Results

Software Environment Details

- **MVAPICH2-X version 2.0 + Intel Compilers**
  - Includes unified support for MPI, UPC, OpenShmem, and OpenMP

- **UPC - berkeley_upc, version 2.18.2**

- **GASNET – version 1.22.4 (ibv conduit)**

- **OpenSHMEM – release: 1.0f**

MVAPICH2 – Dual Rail Performance

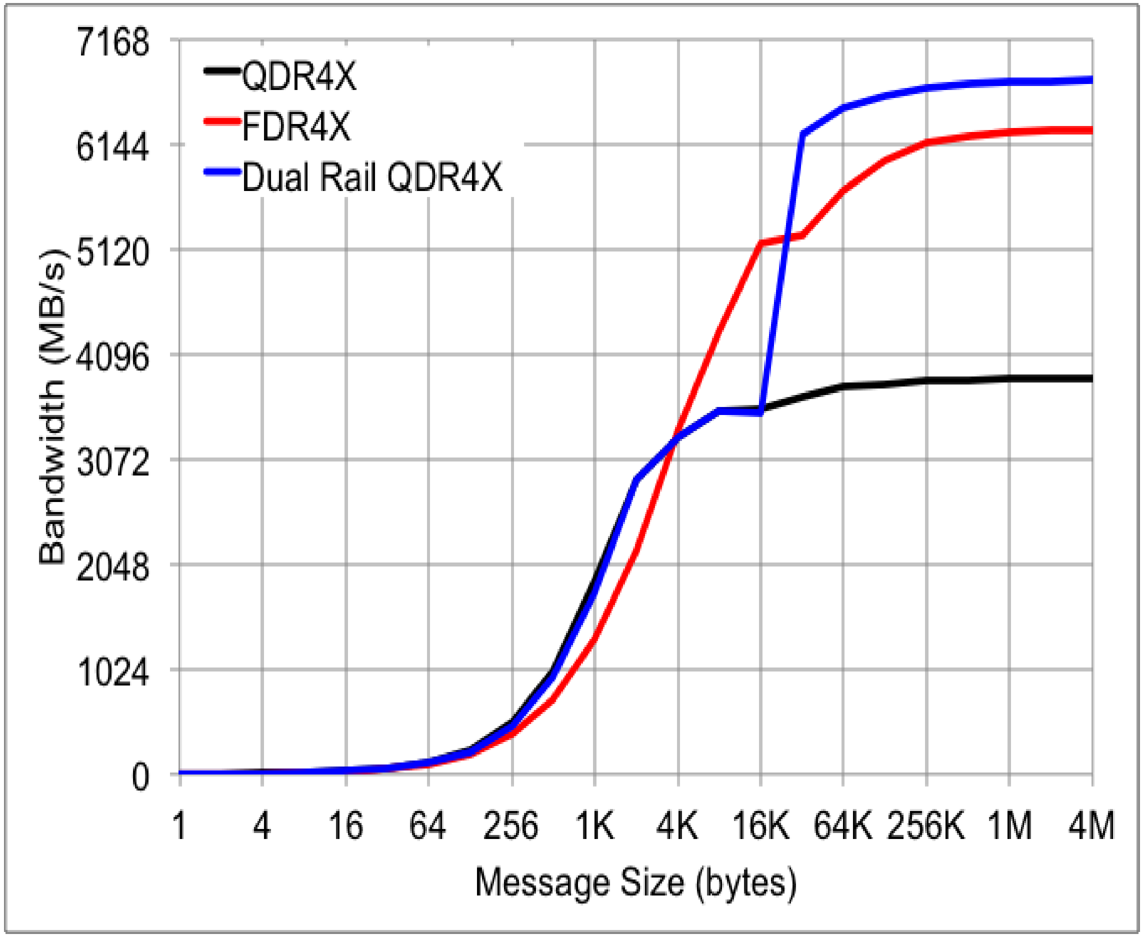
- **Results from XSEDE14 paper***

- **Performance on OSU latency and bandwidth micro-benchmarks. Single and dual rail QDR InfiniBand, FDR InfiniBand compared.**

- **Evaluate the impact of railing sharing, scheduling, and threshold parameters**

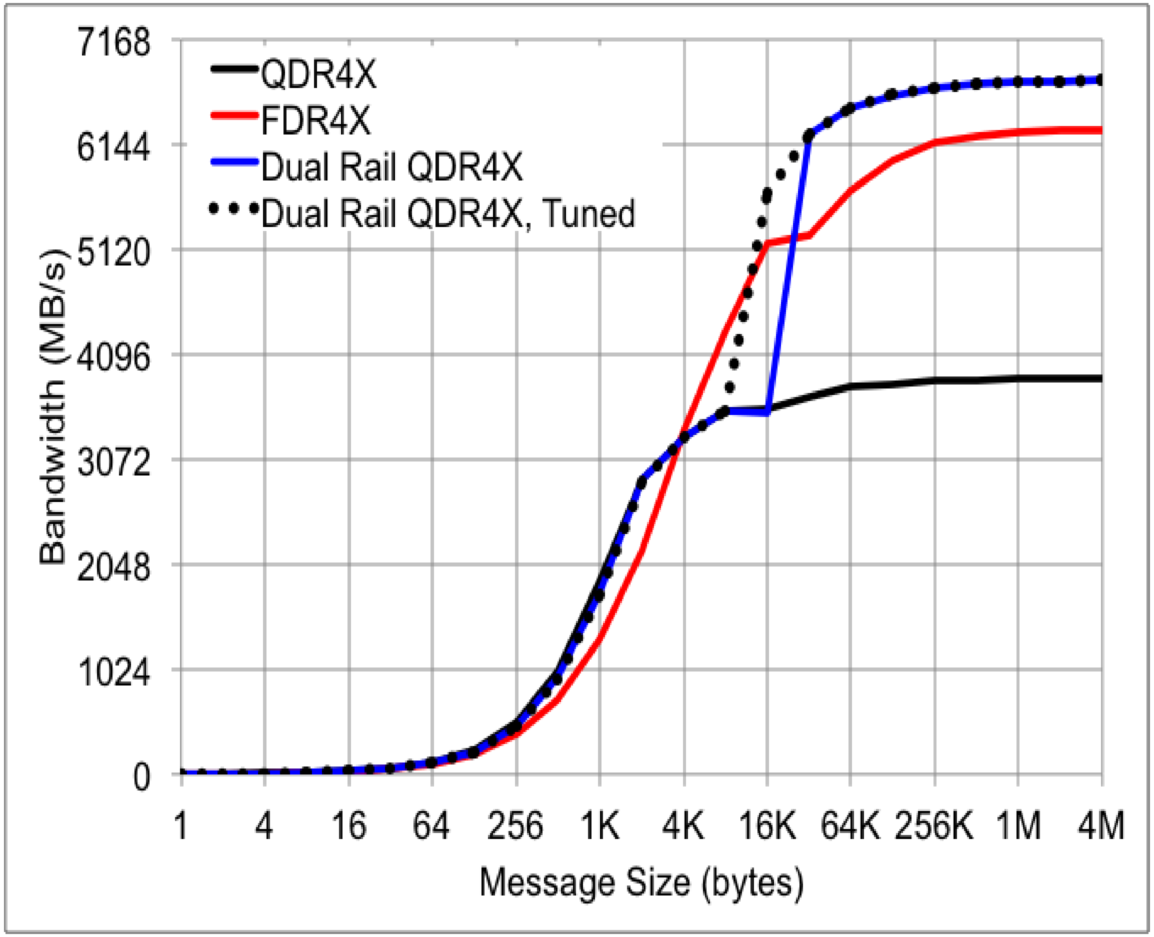
***Performance of Applications using Dual-Rail InfiniBand 3D Torus network on the Gordon Supercomputer**
Dong Ju Choi, Glenn K. Lockwood, Robert S Sinkovits, and Mahidhar Tatineni

OSU Bandwidth Test Results for Single Rail QDR, FDR, and Dual-Rail QDR Network Configurations



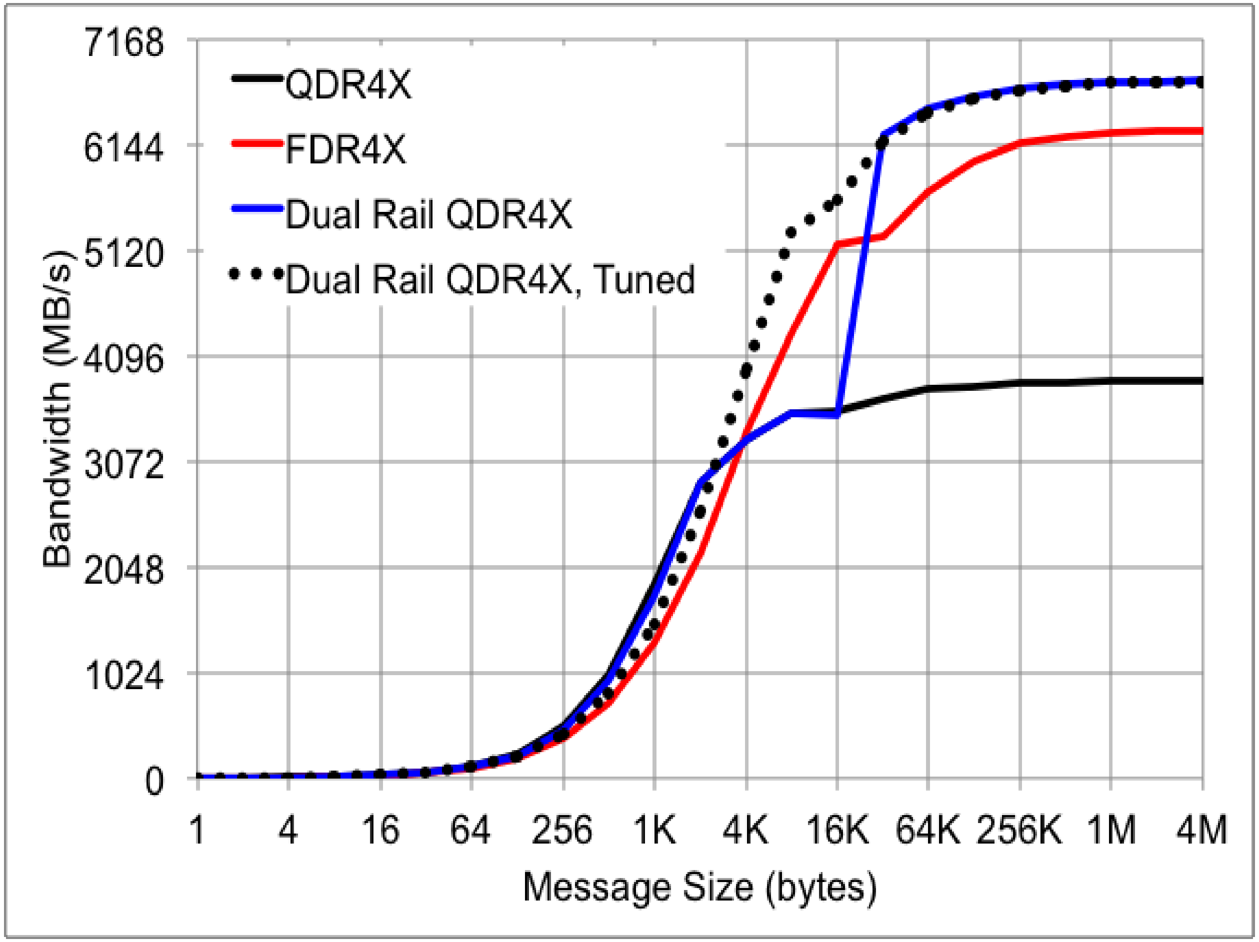
- Single rail FDR performance is much better than single rail QDR for message sizes larger than 4K bytes
- Dual rail QDR performance exceeds FDR performance at sizes greater than 32K
- FDR showing better performance between 4K and 32K byte sizes due to the rail-sharing threshold

OSU Bandwidth Test Performance with MV2_RAIL_SHARING_LARGE_MSG_THRESHOLD=8K



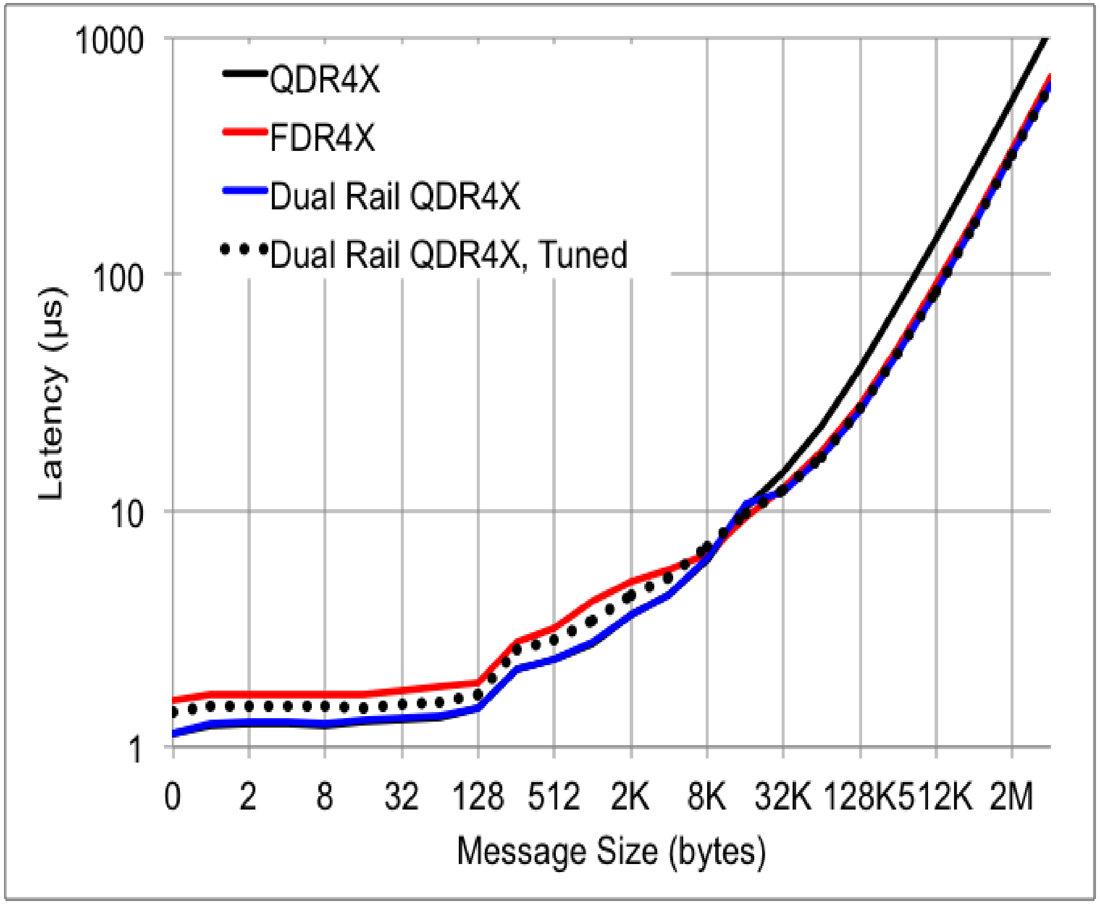
- Lowering the rail sharing threshold bridges the dual-rail QDR, FDR performance gap down to 8K bytes.

OSU Bandwidth Test Performance with MV2_RAIL_SHARING_LARGE_MSG_THRESHOLD=8K And MV2_RAIL_SHARING_POLICY = ROUND_ROBIN



- Adding explicit round-robin tasks to communicate over different rails

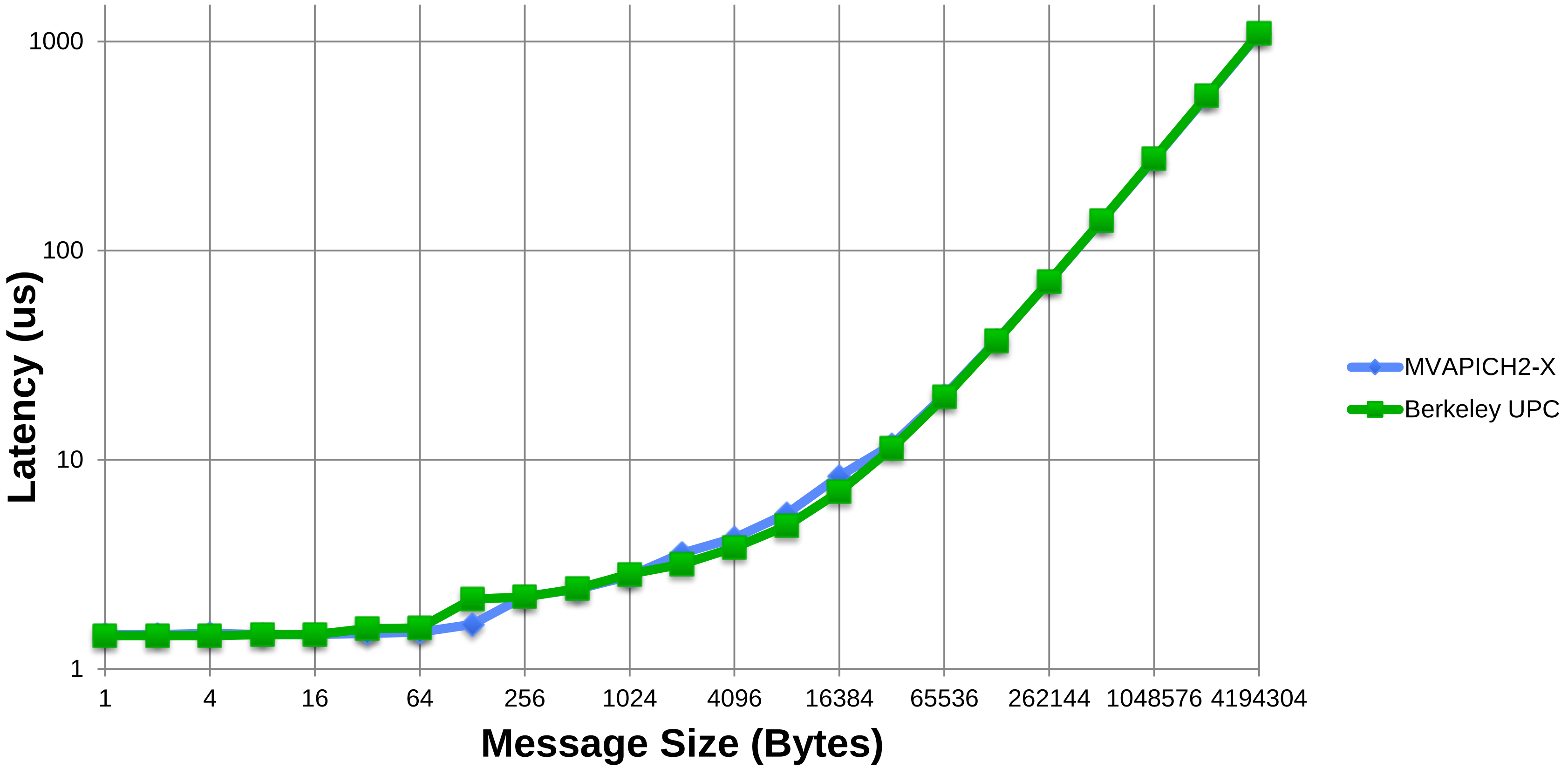
OSU Latency Benchmark Results for QDR, Dual-Rail QDR with Round Robin Option, FDR



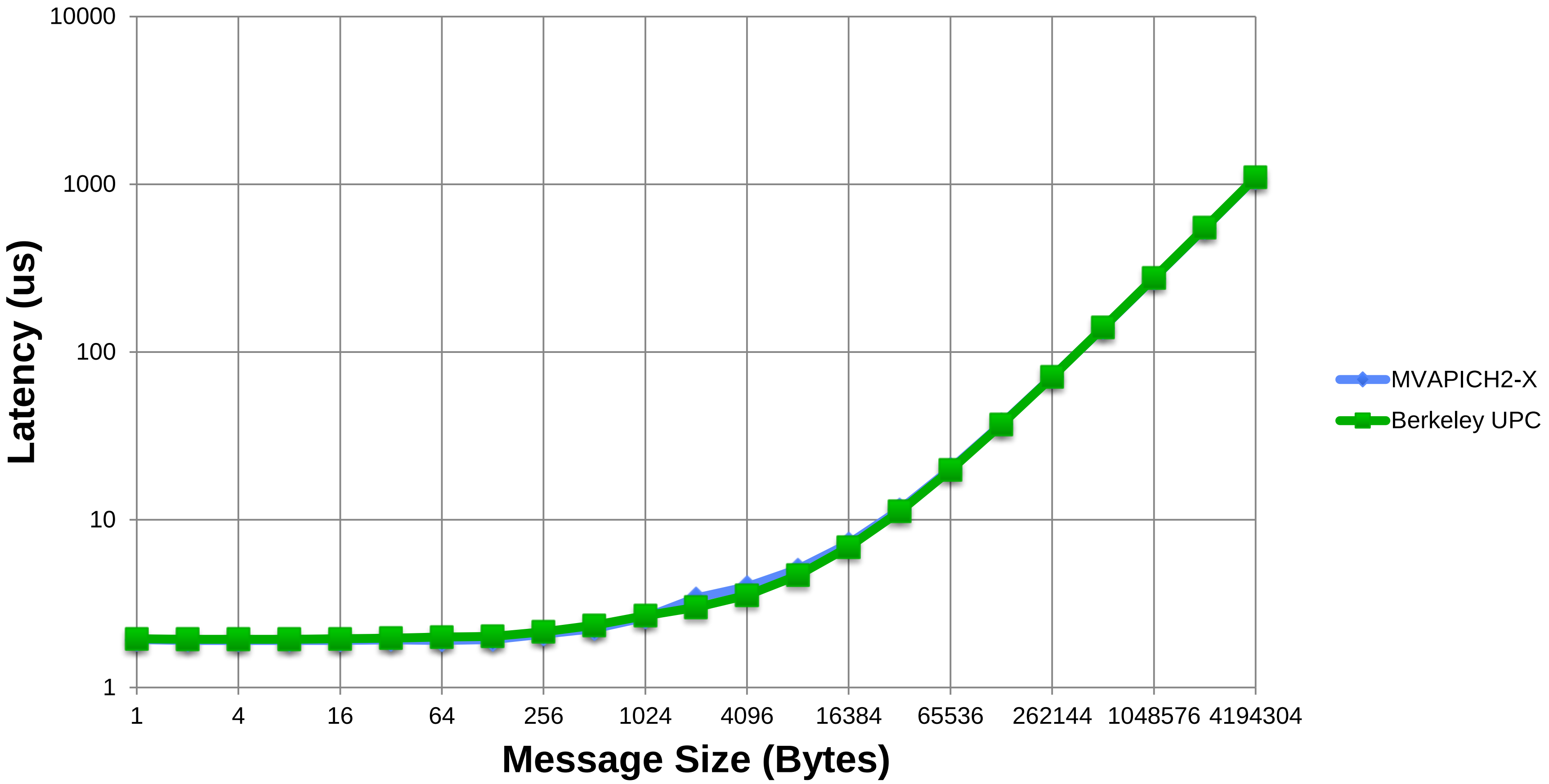
- Distributing messages across HCAs using the round-robin option increases the latency at small message sizes.

- Again, the latency results are better than the FDR case.

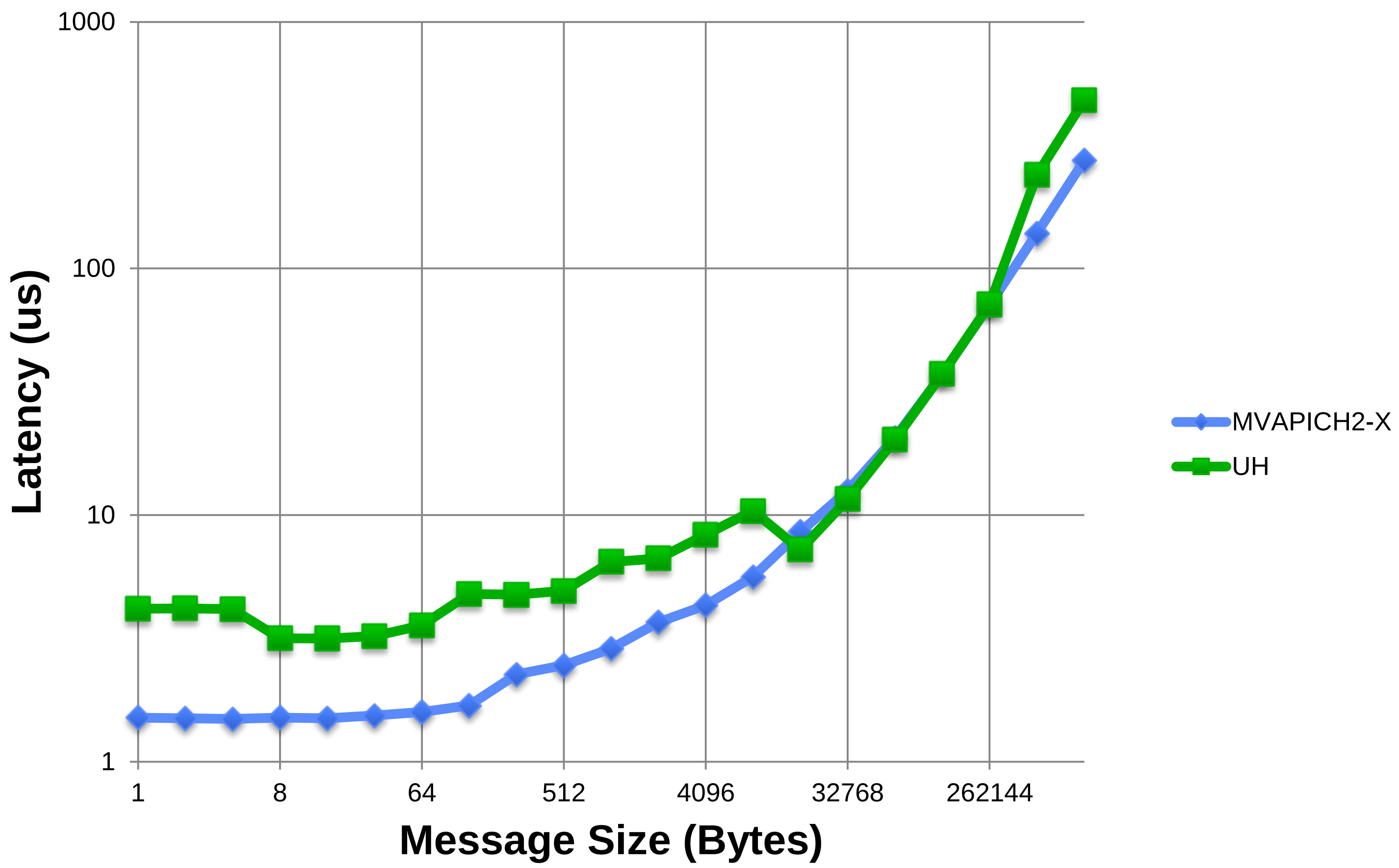
UPC memput – MVAPICH2-X, Berkeley UPC
Two nodes, 1 task/node



UPC memget - MVAPICH2-X, Berkeley UPC
Two nodes, 1 task/node



OpenSHMEM Put – MVAPICH2-X, OpenSHMEM V1.0f
Two tasks, 1 task/node



OpenSHMEM - OSU Atomics Benchmarks

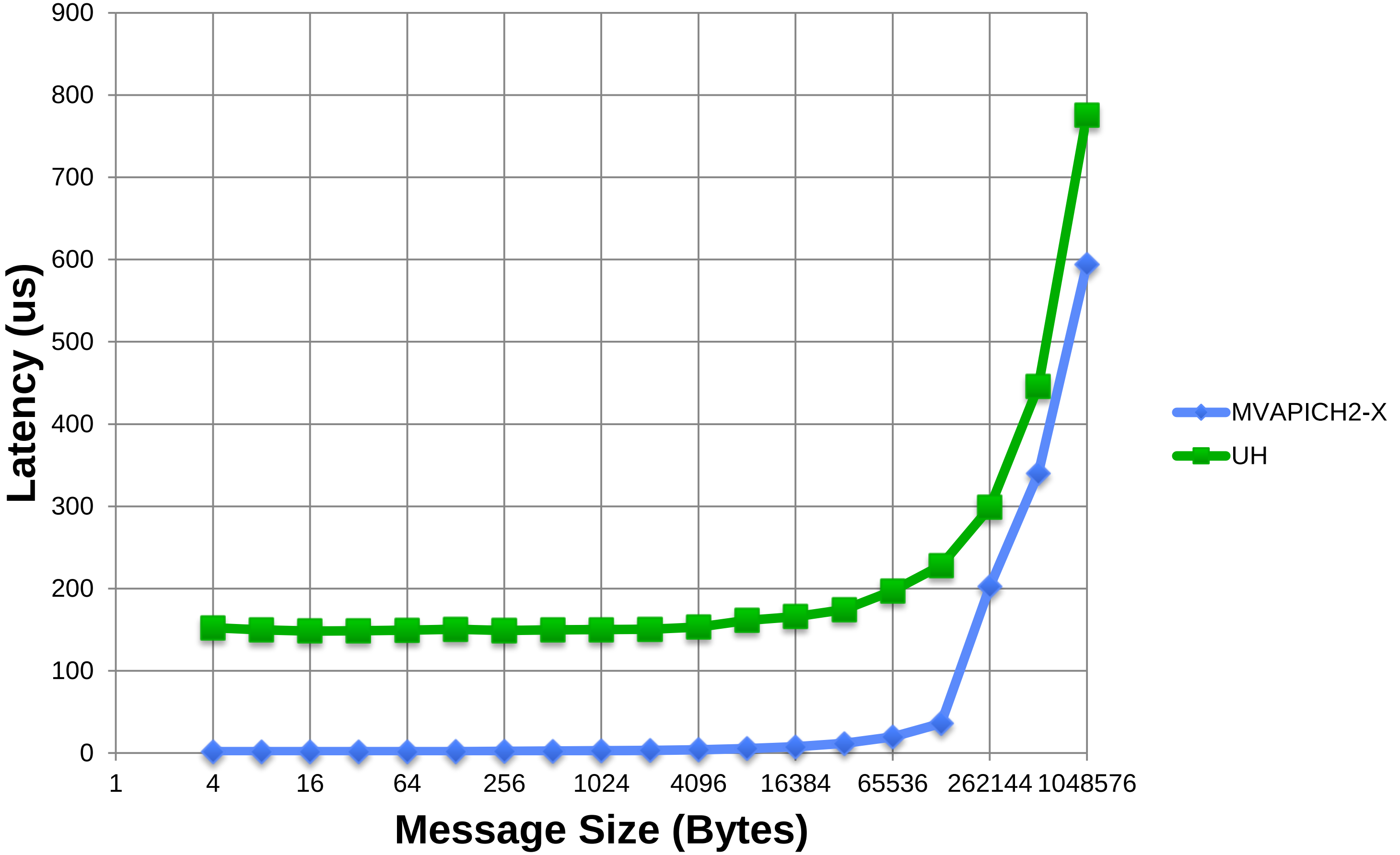
	MV2X-Ops/s	MV2X-Latency	UH-Ops/s	UH-Latency
shmem_int_fadd	0.31	3.19	0.18	5.50
shmem_int_finc	0.40	2.53	0.20	5.04
shmem_int_add	0.42	2.36	0.22	4.60
shmem_int_inc	0.41	2.44	0.01	69.22
shmem_int_cswap	0.38	2.66	0.21	4.83
shmem_int_swap	0.40	2.49	0.22	4.53
shmem_longlong_fadd	0.38	2.61	0.22	4.58
shmem_longlong_finc	0.42	2.41	0.01	71.51
shmem_longlong_add	0.42	2.38	0.23	4.42
shmem_longlong_inc	0.42	2.38	0.22	4.62
shmem_longlong_cswap	0.42	2.39	0.21	4.75
shmem_longlong_swap	0.40	2.50	0.03	33.10

OpenSHMEM Barrier – MVAPICH2-X, OpenSHMEM V1.0f

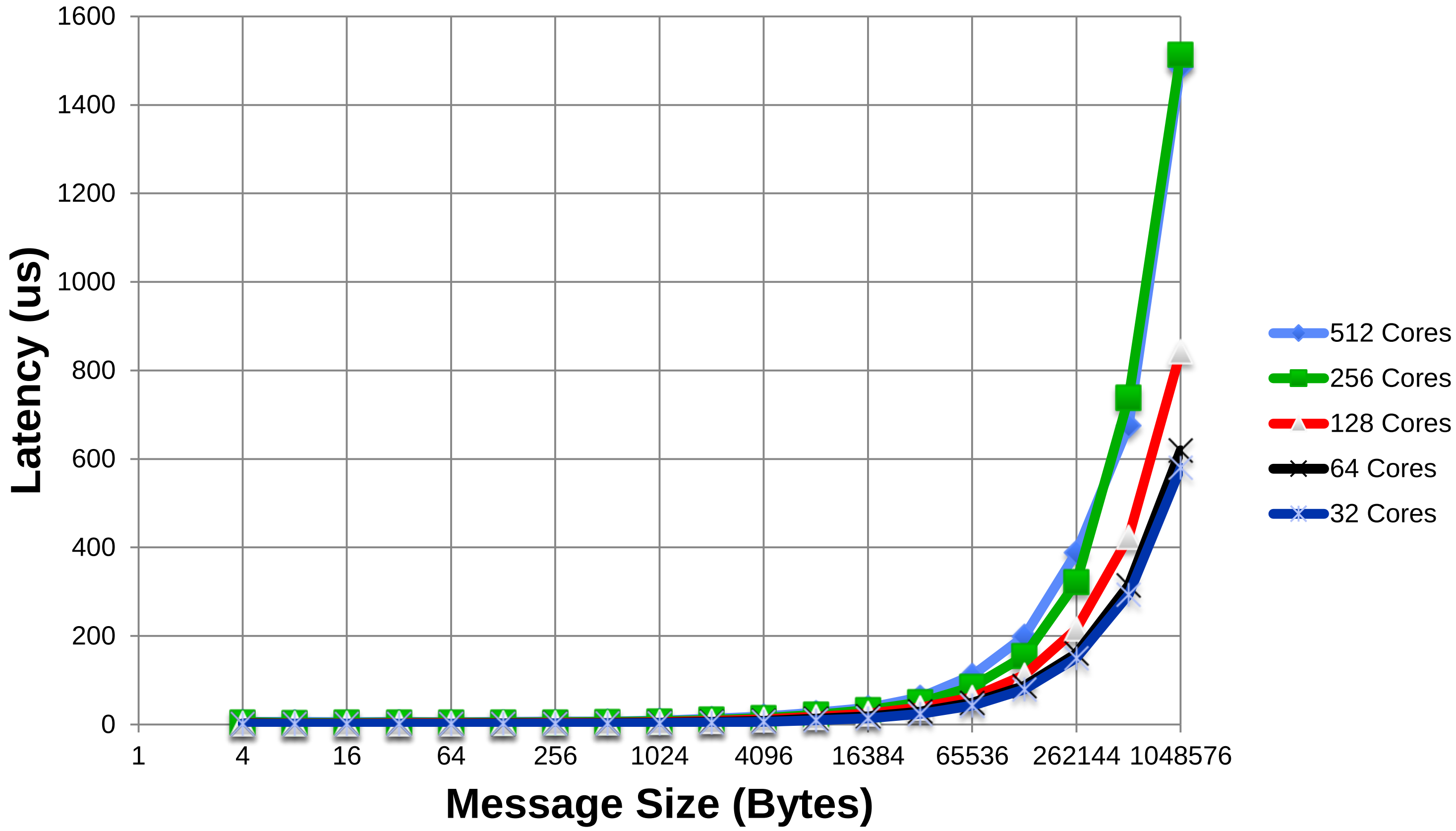
Tasks (Two nodes)	MVAPICH2-X (Latency in us)	Release Openshmem (UH) (Latency in us)*
2	1.87	15.85
4	2.44	77.97
8	2.71	191.90
16	3.29	430.99
32	7.01	1009.97

***Release OpenSHMEM version run with gasnet_ibv conduit. No optimizations attempted and further work may be needed.**

OpenSHMEM Broadcast
2 Nodes, 8 tasks/node (16 tasks total)



OpenSHMEM Broadcast – OSU Benchmark; MVAPICH2-X



Applications:

#1 NAS Parallel Benchmarks:

(a) 2.4 UPC version - GWU/HPCL

http://threads.hpcl.gwu.edu/sites/npb-upc

(b) 3.2 OpenSHMEM version – University of Houston

https://github.com/jeffhammond/openshmem-npbs

#2 CP2K (Hybrid MPI + OpenMP) [Ongoing]

#3 PSDNS (Hybrid MPI + SHMEM) [Ongoing]

NPB CLASS D CG – UPC Version, MVAPICH2-X

Cores (Nodes)	Time (in secs)
32 (2)	385.54
64 (4)	229.46
128 (8)	100.08
256 (16)	64.39

CG: Conjugate Gradient, irregular memory access and communication

NPB, CLASS C IS – UPC Version, MVAPICH2-X

Cores (Nodes)	Time (in secs)
16 (1)	1.52
32 (2)	1.11
64 (4)	0.90
128 (8)	0.74
256 (16)	0.46

IS: Integer Sort, random memory access

NPB CLASS D MG, UPC Version, MVAPICH2-X

Cores (Nodes)	Time (in secs)
32 (2)	46.89
64 (4)	34.77
128 (8)	18.14
256 (16)	8.85
512 (32)	6.07

MG: Multi-Grid on a sequence of meshes, long- and short-distance communication, memory intensive

NPB CLASS D SP, OpenSHMEM Version, MVAPICH2-X

Cores (Nodes)	Time (in secs)
16	2535.42
64	699.16
256	144.68

SP: Scalar Penta-diagonal solver

NPB CLASS D MG, OpenSHMEM Version, MVAPICH2-X

Cores (Nodes)	Time (in seconds)
16 (1)	169.95
32 (2)	93.97
64 (4)	35.90
128 (8)	19.48
256 (16)	9.81

MG: Multi-Grid on a sequence of meshes, long- and short-distance communication, memory intensive

Ongoing/Future Work

- **Further investigate performance results, look into OpenSHMEM v1.0f aspect.**
- **Testing at larger scales (Gordon, Stampede, *Comet*)**
- **Hybrid code performance**
  - PSDNS – MPI + OpenSHMEM version (Dmitry Pekurovsky)
  - CP2K – MPI + OpenMP

- **Big Thanks to Dr. Panda's team for their excellent work and support for MVAPICH2, MVAPICH2-X!**