MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Achieving High Performance on Microsoft Azure HPC Cloud using MVAPICH2

Microsoft Azure Booth Talk at SC '19

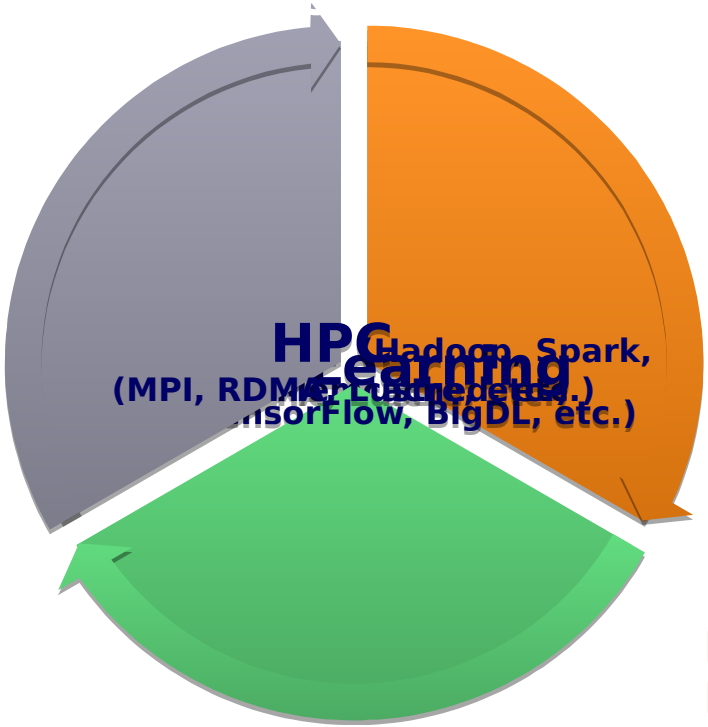
by

Dhabaleswar K. (DK) Panda

The Ohio State University

E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda



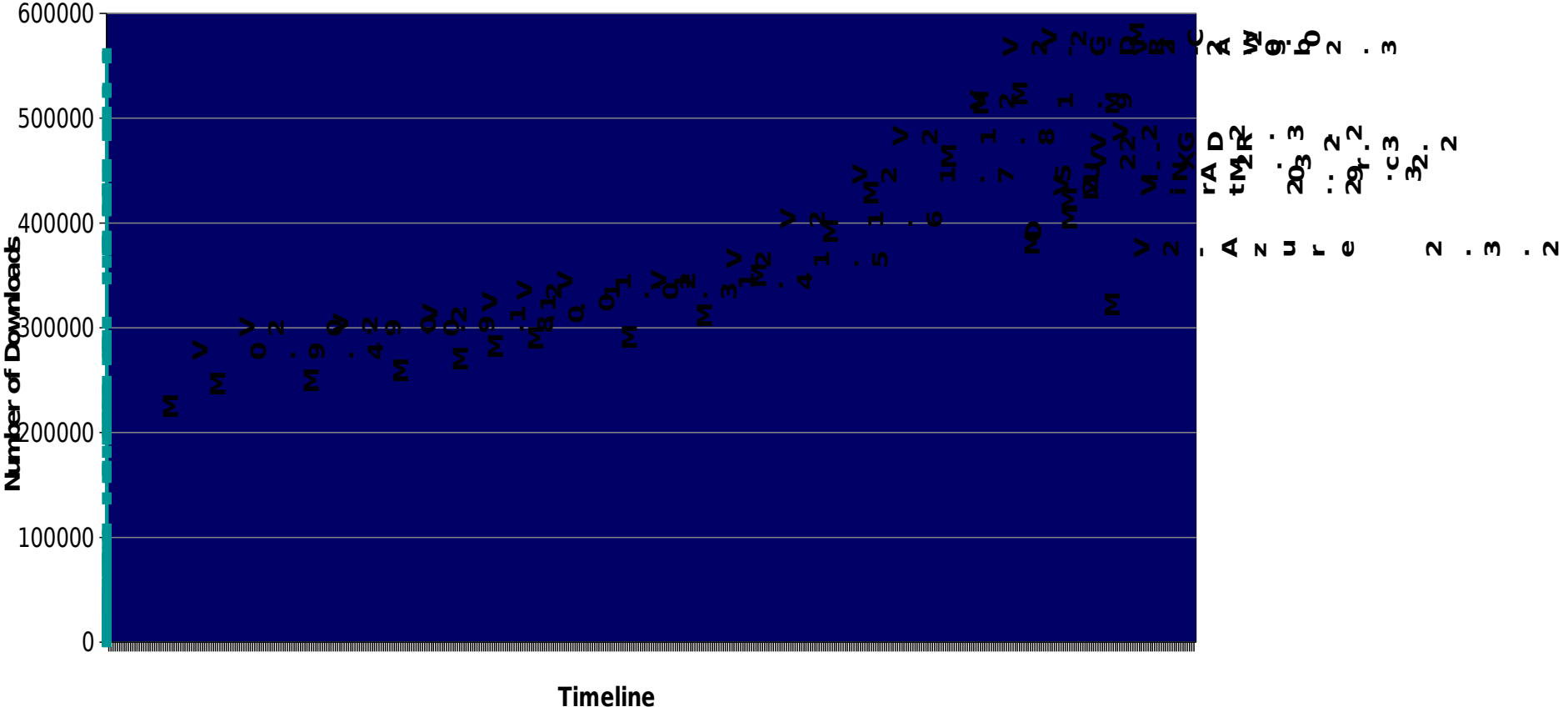
Convergence of HPC, Big Data, and Deep Learning!

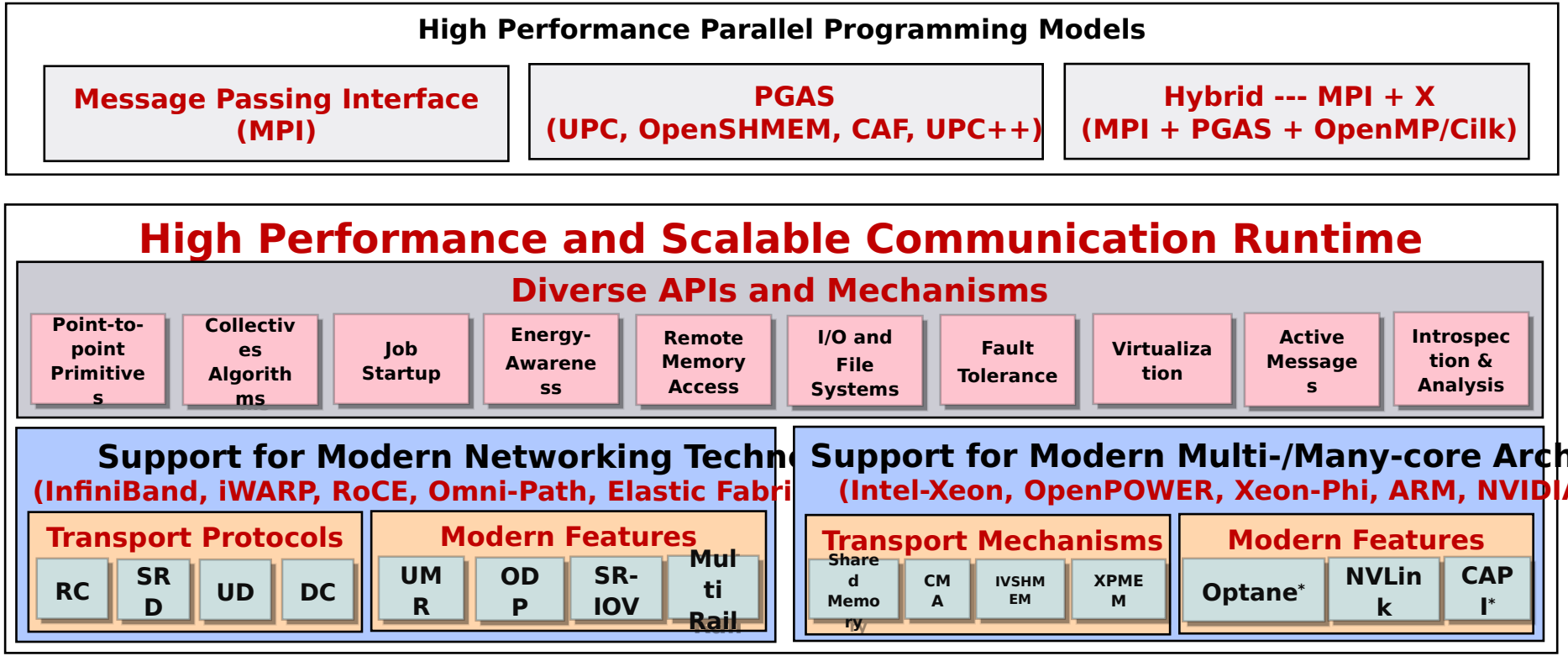
Increasing Need to Run these applications on the Cloud!!

- **Overview of the MVAPICH2 Project**
- Overview of Azure HPC Environments
- Designing MVAPICH2 for Azure
- Performance Results
- Future Plans

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002 (Supercomputing '02)
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Avai
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) s
  - **Used by more than 3,050 organizations in 89 countrie**
  - **More than 615,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '19 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 14th, 570,020 cores (Nurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**

Partner in the #5th TACC Frontera Syste


600000
500000
400000
300000
200000
100000
0
Number of Downloads
Timeline


High Performance Parallel Programming Models
Message Passing Interface (MPI)
PGAS (UPC, OpenSHMEM, CAF, UPC++)
Hybrid --- MPI + X (MPI + PGAS + OpenMP/Cilk)
High Performance and Scalable Communication Runtime
Diverse APIs and Mechanisms
Point-to-point Primitives
Collectives Algorithms
Job Startup
Energy-Awareness
Remote Memory Access
I/O and File Systems
Fault Tolerance
Virtualization
Active Messages
Introspection & Analysis
Support for Modern Networking Techn
(InfiniBand, iWARP, RoCE, Omni-Path, Elastic Fabri
Transport Protocols
RC
SRD
UD
DC
Modern Features
UMR
ODP
SR-IOV
Multi Rail
Support for Modern Multi-/Many-core Archi
(Intel-Xeon, OpenPOWER, Xeon-Phi, ARM, NVIDIA
Transport Mechanisms
Shared Memory
CMA
IVSHMEM
XPMEM
Modern Features
Optane*
NVLink
CAPI*

* Upcoming

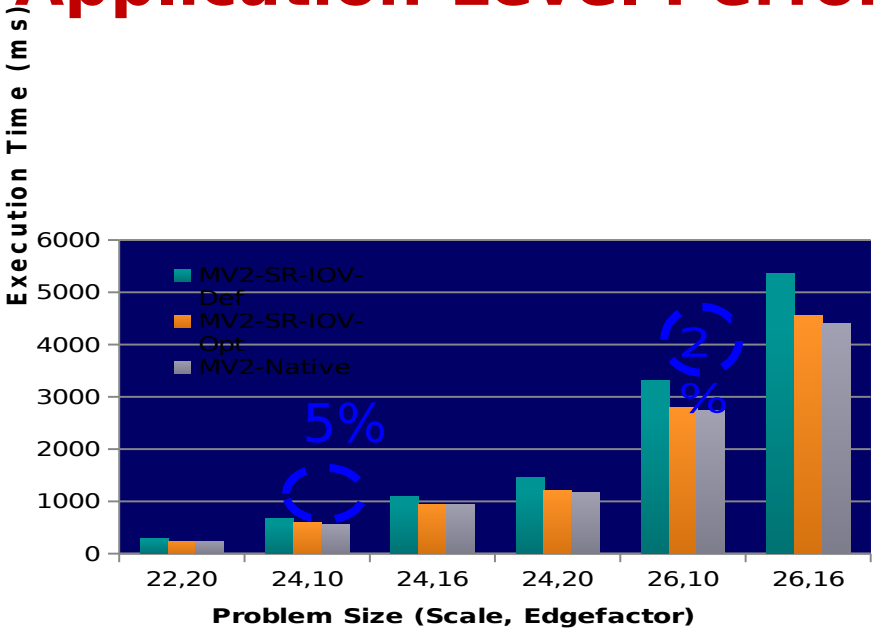
- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2 support for SR-IOV
- MVAPICH2-Virt 2.2 supports:
  - OpenStack, Docker, and singularity

J. Zhang, X. Lu, J. Jose, R. Shi and D. K. Panda, Can Inter-VM Shmem Benefit MPI Applications on SR-IOV based Virtualized InfiniBand Clusters? EuroPar'14

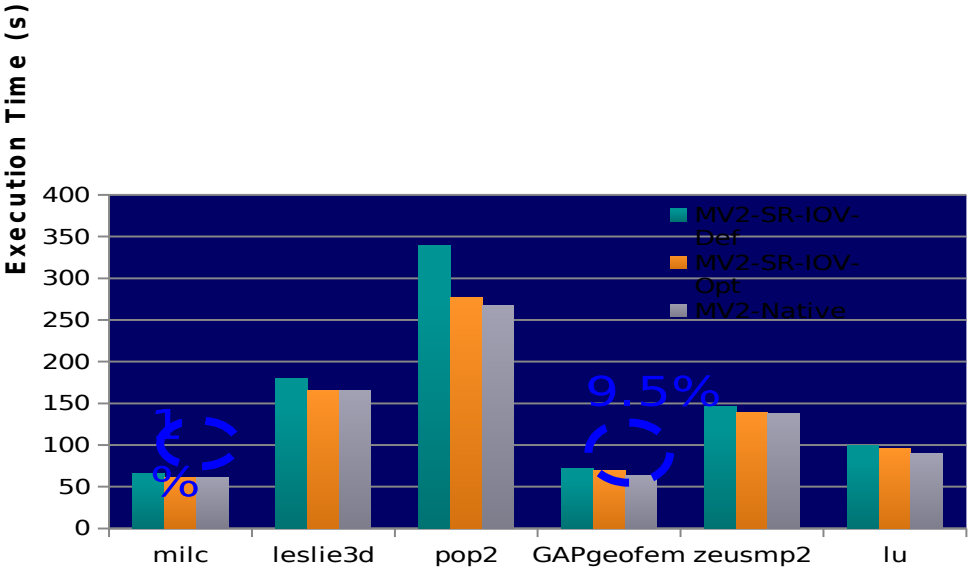
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC'14

J. Zhang, X .Lu, M. Arnold and D. K. Panda, MVAPICH2 Over OpenStack with SR-IOV: an Efficient Approach to build HPC Clouds, CCGrid'15

Application-Level Performance on Chameleon



Graph500



SPEC MPI2007

- 32 VMs, 6 Core/VM
- Compared to Native, 2-5% overhead for Graph500 with 128 Procs
- Compared to Native, 1-9.5% overhead for SPEC MPI2007 with 128 Procs

- Overview of the MVAPICH2 Project
- **Overview of Azure HPC Environments**
- Designing MVAPICH2 for Azure
- Performance Results
- Future Plans

- Has been using RDMA-enabled network and software stacks for the last several years
- Moved to native InfiniBand support with Mellanox OFED for new instances (HB, HC, and upcoming ones)
- Uses SR-IOV support for virtualization
- Uses one VM per node

- Overview of the MVAPICH2 Project
- Overview of Azure HPC Environments
- **Designing MVAPICH2 for Azure**
- Performance Results
- Future Plans

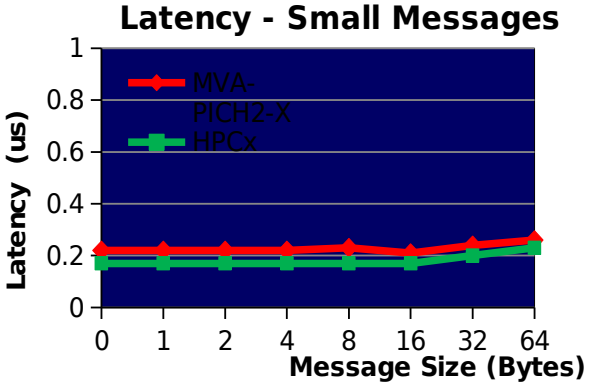
Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

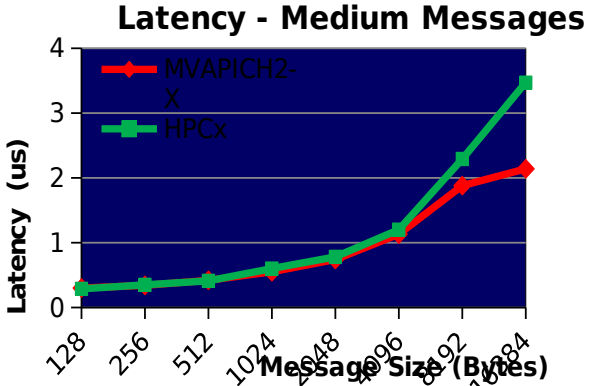
- **Released on 08/16/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-2.3.2**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for Azure HB & HC virtual machine**
  - **Flexibility for 'one-click' deploy**
  - **Tested with Azure HB & HC VM instances**

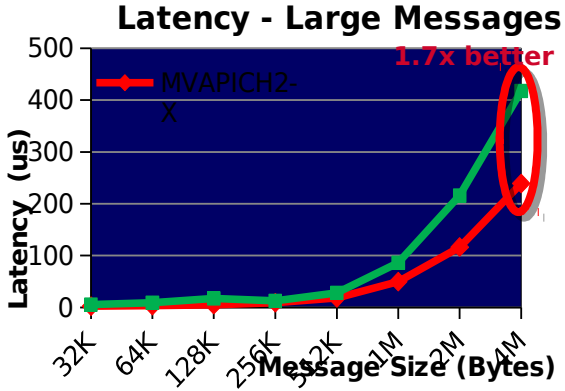
- Available for download from http://mvapich.cse.ohio-state.edu/downloads/
- Detailed User Guide: http://mvapich.cse.ohio-state.edu/userguide/mv2-azure/
- On-going Work
  - Optimizing MVAPICH2-X features on Azure

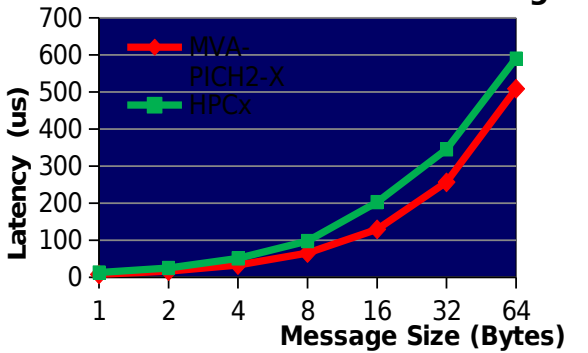
- Overview of the MVAPICH2 Project
- Overview of Azure HPC Environments
- Designing MVAPICH2 for Azure
- **Performance Results**
- Future Plans

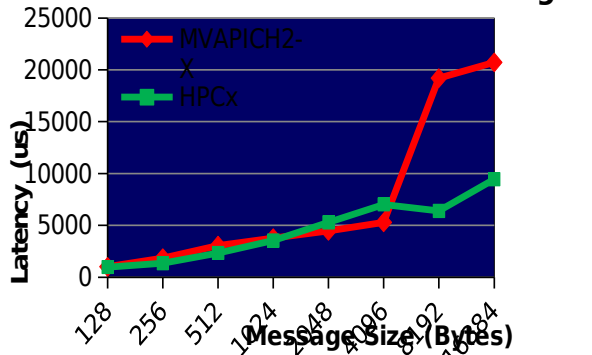
- VM type: Azure HC
- CPU: Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
- Cores: 2 socket, 44 cores
- MVAPICH2 Version: Latest MVAPICH2-X w/ XPMEM support
- HPCx Version: Built-in HPCx-v2.5.0-gcc-MLNX_OFED_LINUX-4.7-1.0.0.1-redhat7.6-x86_64
- OMB Version: OSU-MicroBenchmars-5.6.2

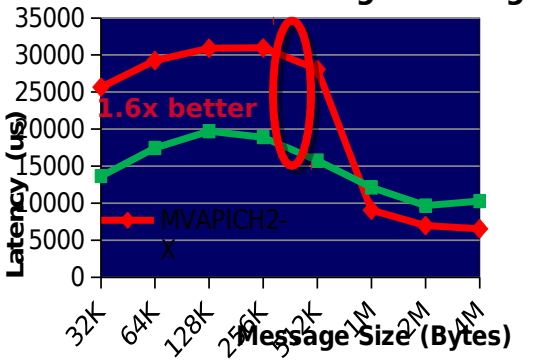

Latency - Small Messages
Latency (us)
Message Size (Bytes)
MVA-PICH2-X
HPCx
0
0.2
0.4
0.6
0.8
1
0
1
2
4
8
16
32
64

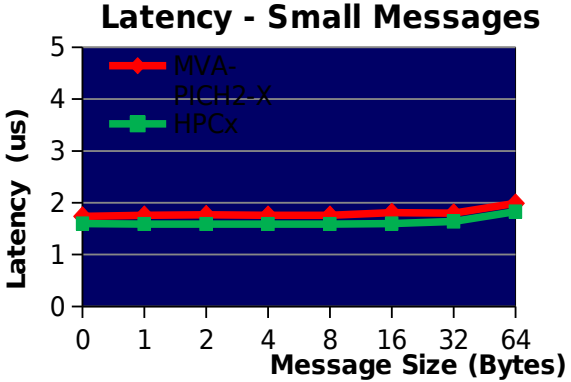

Latency - Medium Messages
Latency (us)
Message Size (Bytes)
MVAPICH2-X
HPCx
0
1
2
3
4
128
256
512
1024
2048
4096
8192
16384

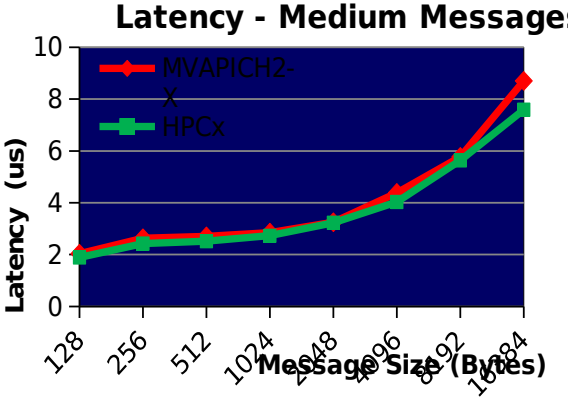

Latency - Large Messages
1.7x better
Latency (us)
Message Size (Bytes)
MVAPICH2-X
0
100
200
300
400
500
32K
64K
128K
256K
512K
1M
2M
4M

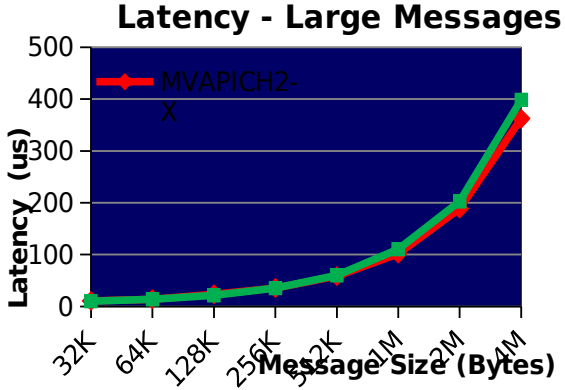

Bandwidth - Small Messages
Latency (us)
Message Size (Bytes)
MVA-PICH2-X
HPCx
0
100
200
300
400
500
600
700
1
2
4
8
16
32
64

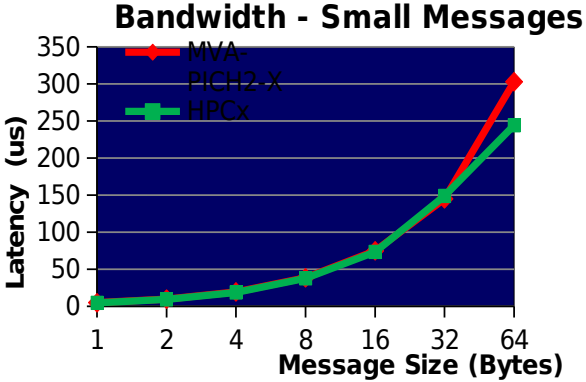

Bandwidth - Medium Messages
Latency (us)
Message Size (Bytes)
MVAPICH2-X
HPCx
0
5000
10000
15000
20000
25000
128
256
512
1024
2048
4096
8192
16384

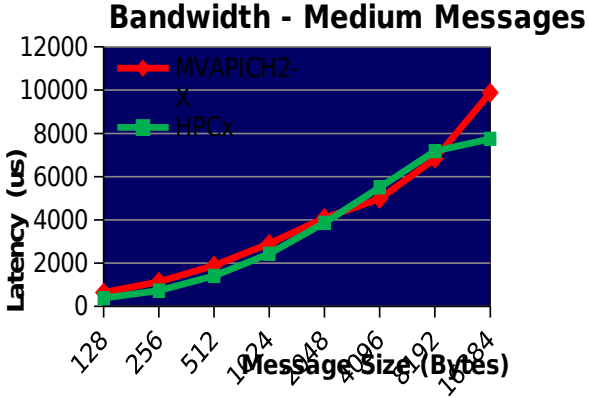

Bandwidth - Large Messages
1.6x better
Latency (us)
Message Size (Bytes)
MVAPICH2-X
0
5000
15000
25000
30000
35000
32K
64K
128K
256K
512K
1M
2M
4M

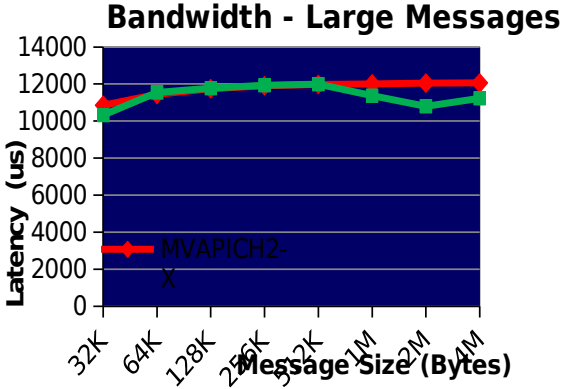

Latency - Small Messages
Latency (us)
MVA-
PICH2-X
HPCx
0
1
2
3
4
5
0
1
2
4
8
16
32
64
Message Size (Bytes)

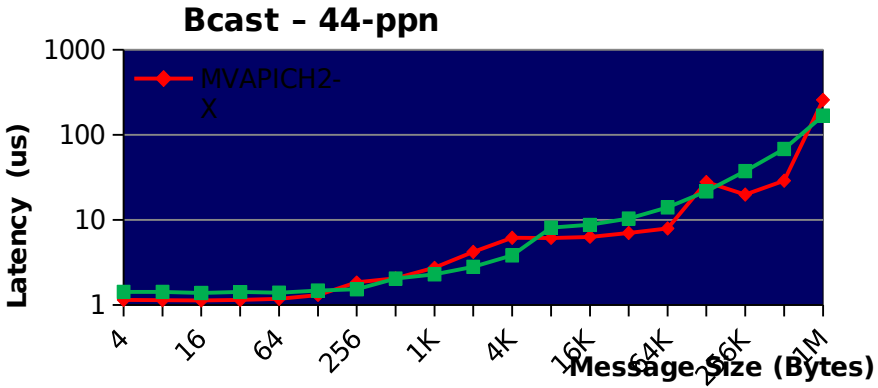

Latency - Medium Messages
Latency (us)
MVAPICH2-
X
HPCx
0
2
4
6
8
10
128
256
512
1024
2048
4096
8192
16384
Message Size (Bytes)

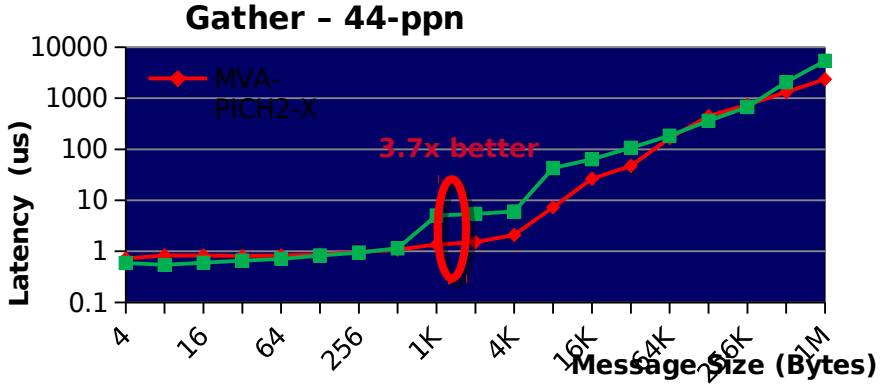

Latency - Large Messages
Latency (us)
MVAPICH2-
X
0
100
200
300
400
500
32K
64K
128K
256K
512K
1M
2M
4M
Message Size (Bytes)

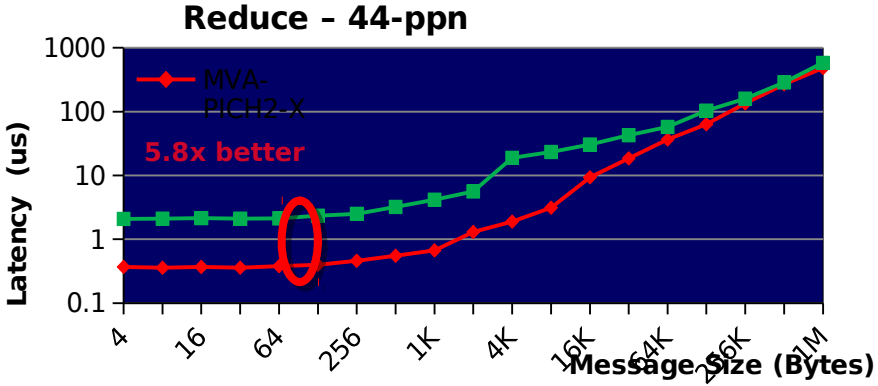

Bandwidth - Small Messages
Latency (us)
MVA-
PICH2-X
HPCx
0
50
100
150
200
250
300
350
1
2
4
8
16
32
64
Message Size (Bytes)

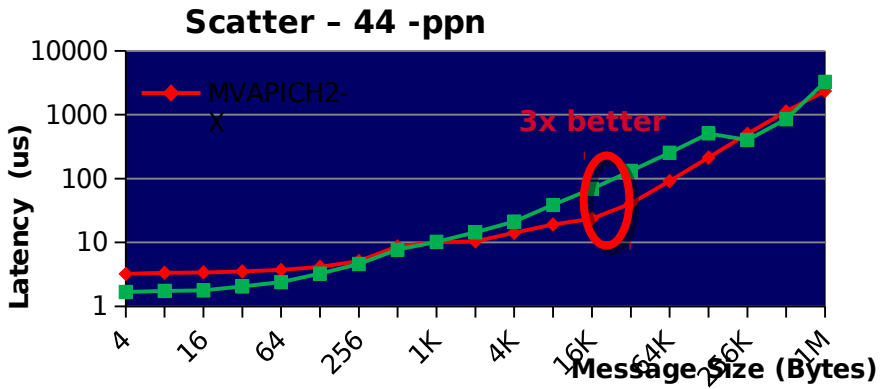

Bandwidth - Medium Messages
Latency (us)
MVAPICH2-
X
HPCx
0
2000
4000
6000
8000
10000
12000
128
256
512
1024
2048
4096
8192
16384
Message Size (Bytes)

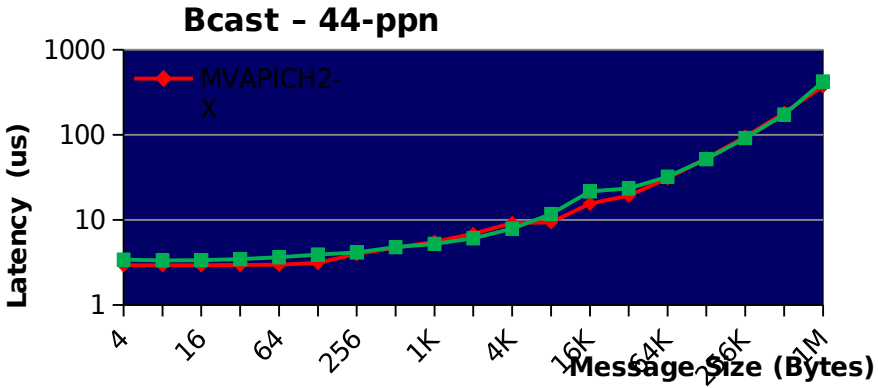

Bandwidth - Large Messages
Latency (us)
MVAPICH2-
X
0
2000
4000
6000
8000
10000
12000
14000
32K
64K
128K
256K
512K
1M
2M
4M
Message Size (Bytes)

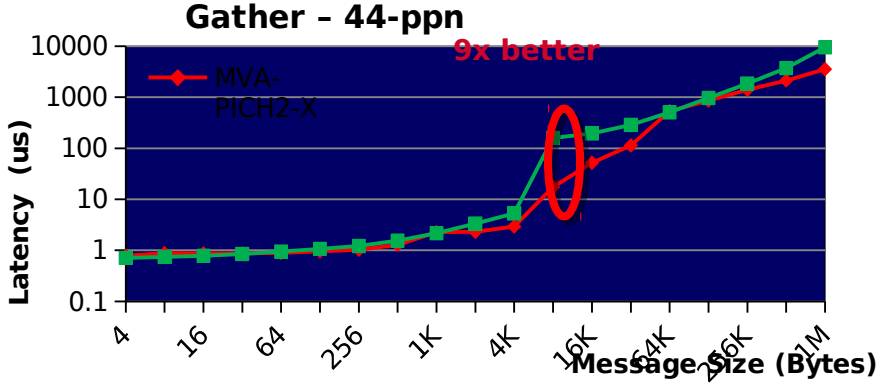

Bcast – 44-ppn
MVAPICH2-X
Latency (us)
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

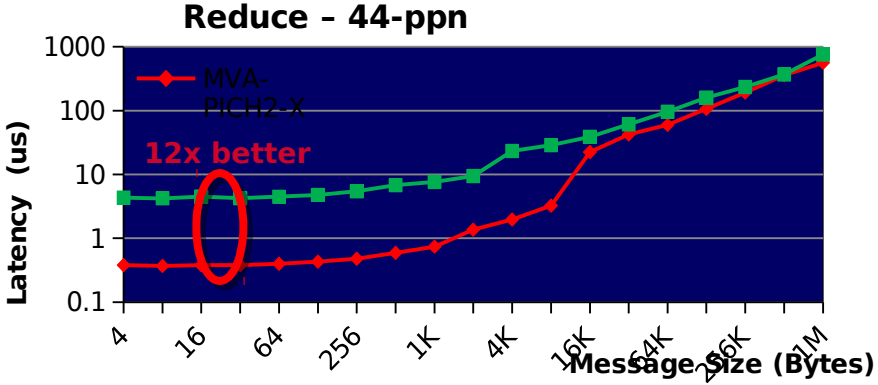

Gather – 44-ppn
MVA-PICH2-X
3.7x better
Latency (us)
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

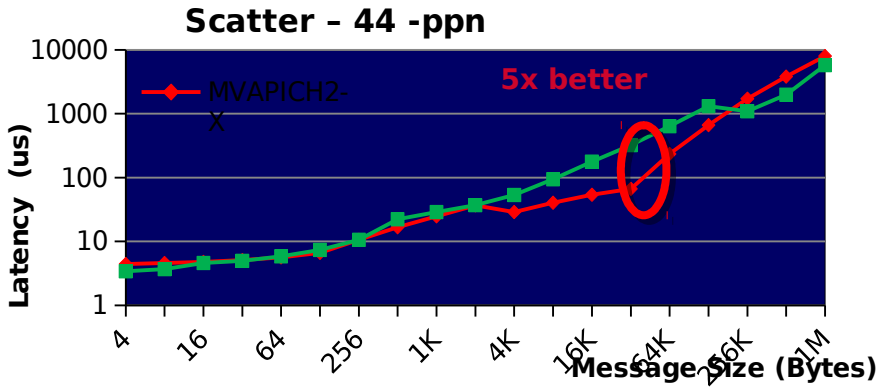

Reduce – 44-ppn
MVA-PICH2-X
5.8x better
Latency (us)
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

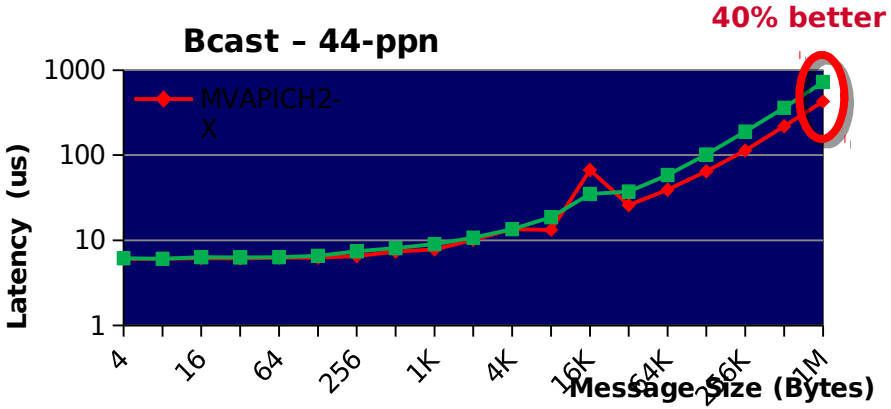

Scatter – 44 -ppn
MVAPICH2-X
3x better
Latency (us)
10000
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

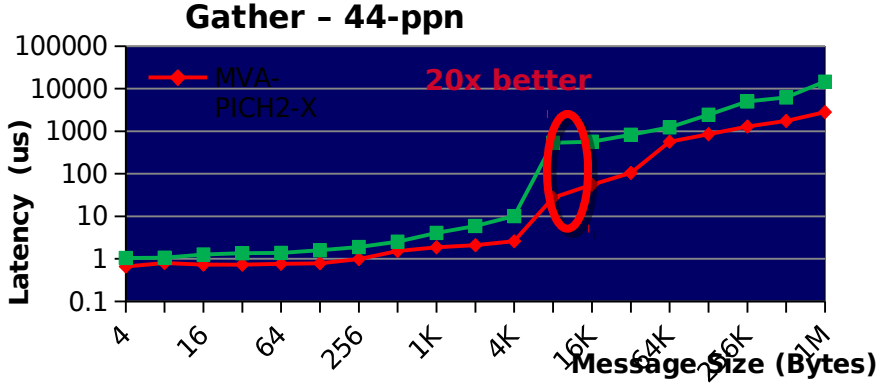

Bcast – 44-ppn
MVAPICH2-X
Latency (us)
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

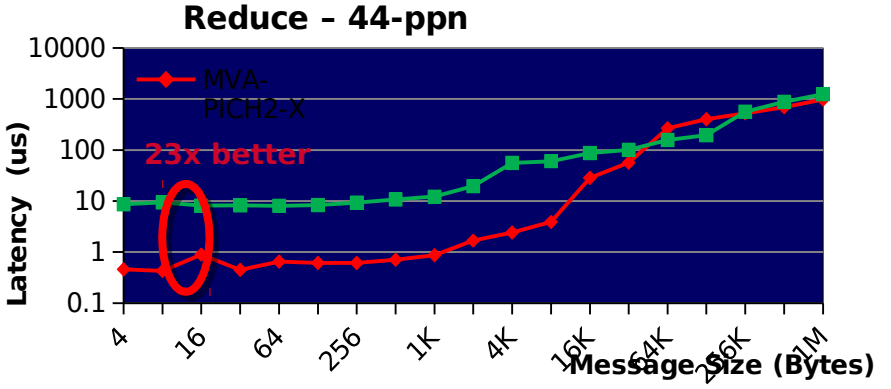

Gather – 44-ppn
9x better
MVA-PICH2-X
Latency (us)
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

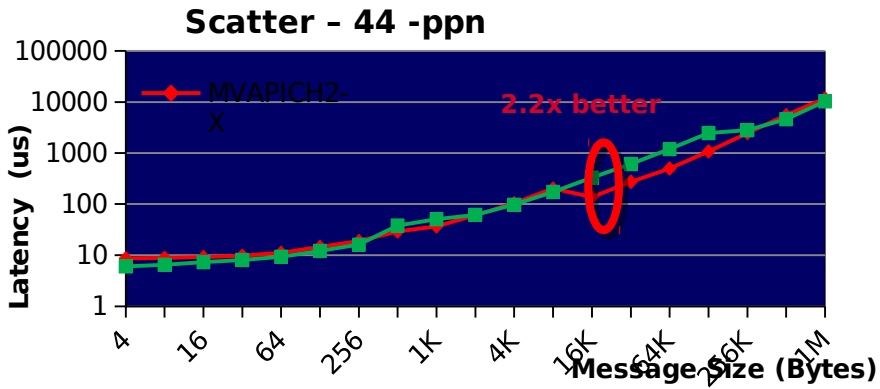

Reduce – 44-ppn
MVA-PICH2-X
12x better
Latency (us)
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

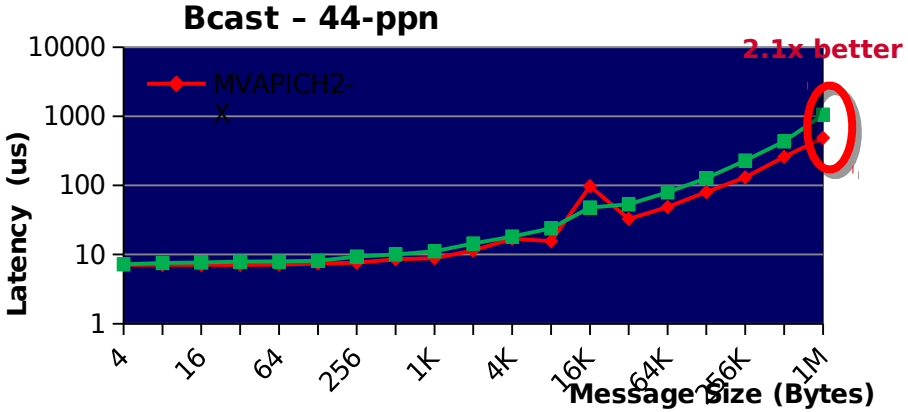

Scatter – 44 -ppn
5x better
MVAPICH2-X
Latency (us)
10000
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

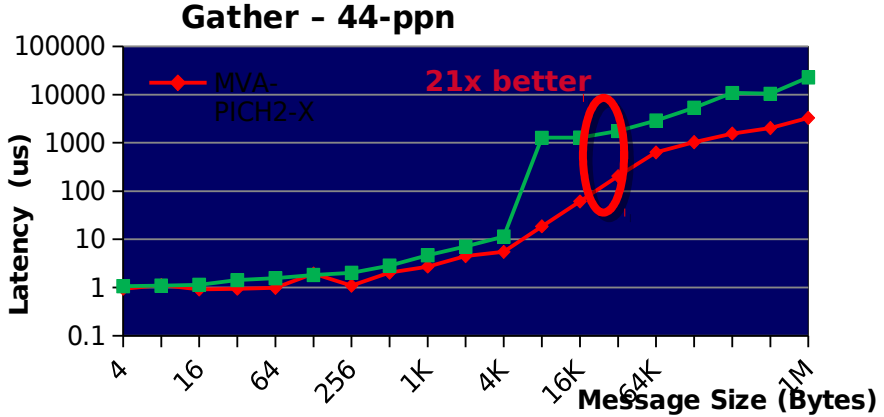

Bcast – 44-ppn
40% better
MVAPICH2-X
Latency (us)
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

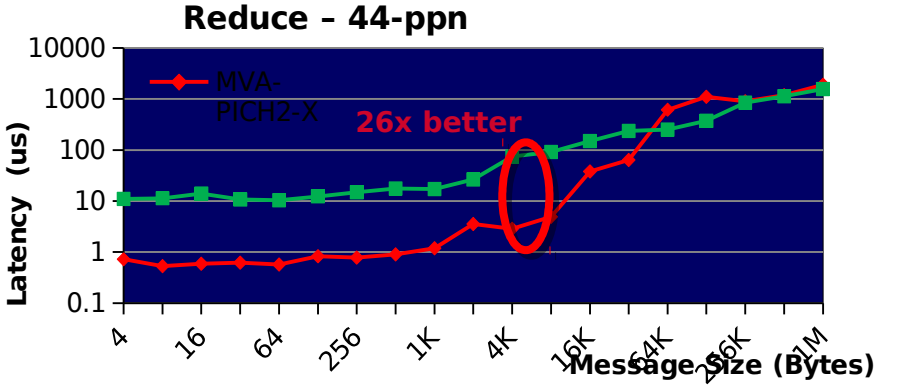

Gather – 44-ppn
MVA-PICH2-X
20x better
Latency (us)
100000
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

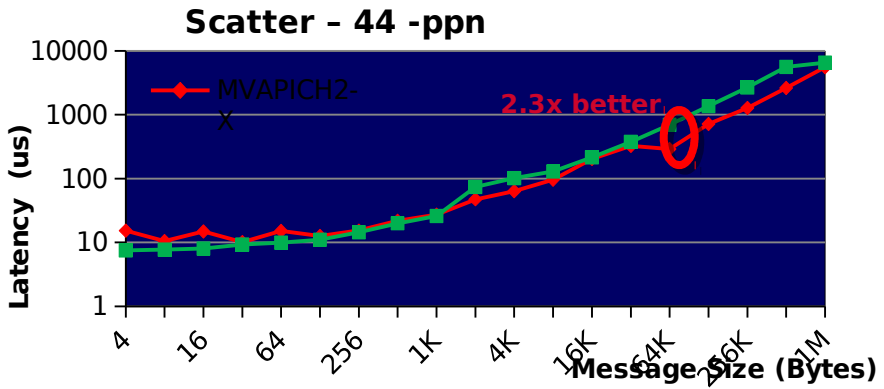

Reduce – 44-ppn
MVA-PICH2-X
23x better
Latency (us)
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

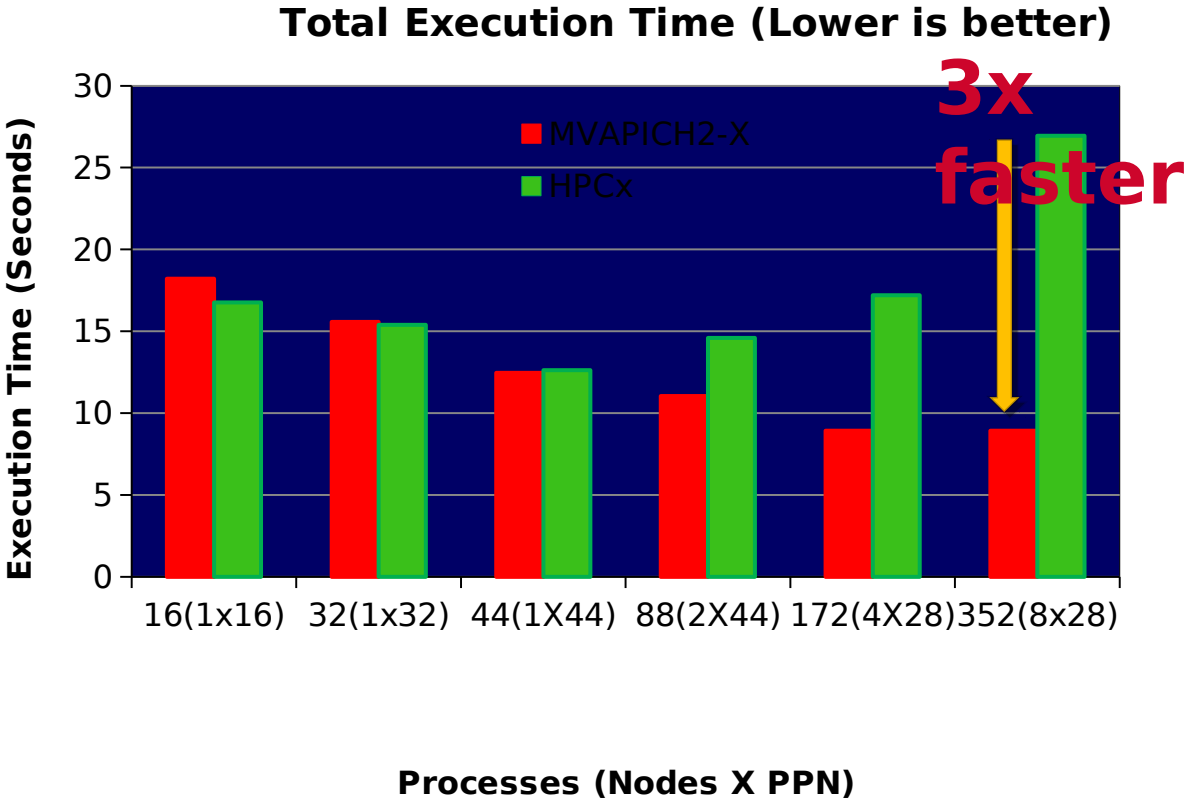

Scatter – 44 -ppn
MVAPICH2-X
2.2x better
Latency (us)
100000
10000
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

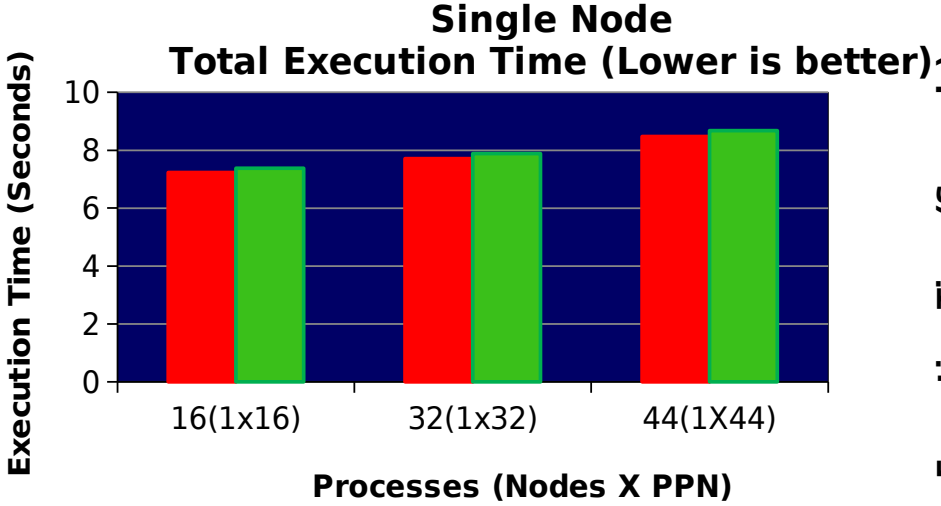

Bcast – 44-ppn
2.1x better
MVAPICH2-X
Latency (us)
10000
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)

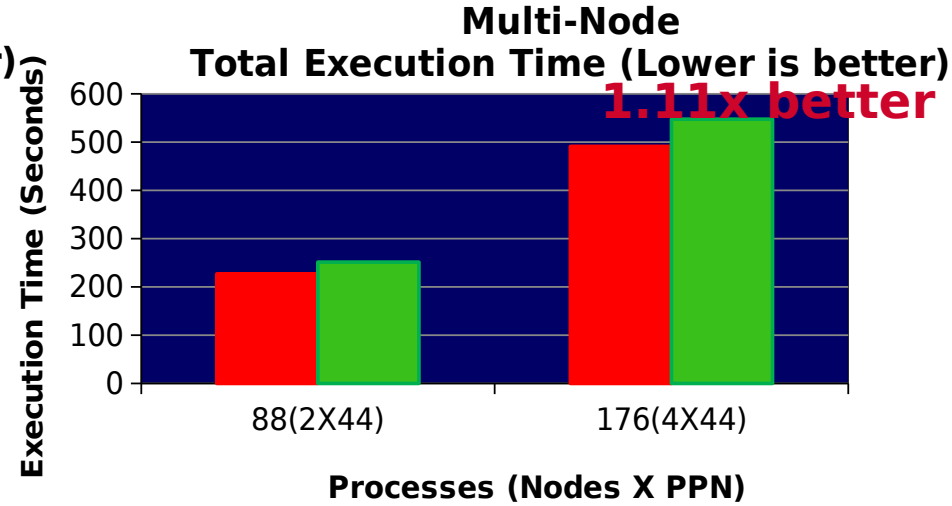

Gather – 44-ppn
21x better
MVA-PICH2-X
Latency (us)
100000
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
1M
Message Size (Bytes)


Reduce – 44-ppn
26x better
MVA-PICH2-X
Latency (us)
10000
1000
100
10
1
0.1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)


Scatter – 44 -ppn
2.3x better
MVAPICH2-X
Latency (us)
10000
1000
100
10
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)


Total Execution Time (Lower is better)
3x faster
MVAPICH2-X
HPCx
Execution Time (Seconds)
30
25
20
15
10
5
0
16(1x16)
32(1x32)
44(1X44)
88(2X44)
172(4X28)
352(8x28)
Processes (Nodes X PPN)





Part of input parameter:

&MESH IJK=5,5,5, XB=-1.0,0.0,-1.0,0.0,0.0,1.0, MULT_ID='mesh array'

- Overview of the MVAPICH2 Project
- Overview of Azure HPC Environments
- Designing MVAPICH2 for Azure
- Performance Results
- **Future Plans**

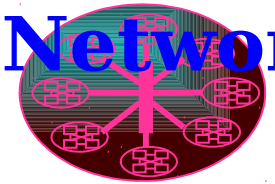
- Additional optimizations and tuning of MVAPICH2-X
- A new version will be released soon
- Making it available it in an integrated manner in the Azure portal
- Making it available through Azure Market Place
- Commercial Support available for End-Users, ISVs, and Organizations
  - Through X-ScaleSolutions (http://x-scalesolutions.com)

Thank You!

panda@cse.ohio-state.edu

Follow us on

https://twitter.com/mvapich



Network Based Computing Laboratory

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

HiBD
High-Performance Big Data

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

HiDL
High-Performance Deep Learning

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/