MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Designing MPI and PGAS Libraries for Exascale Systems: The MVAPICH2 Approach

Talk at OpenFabrics Workshop (April 2016)

by

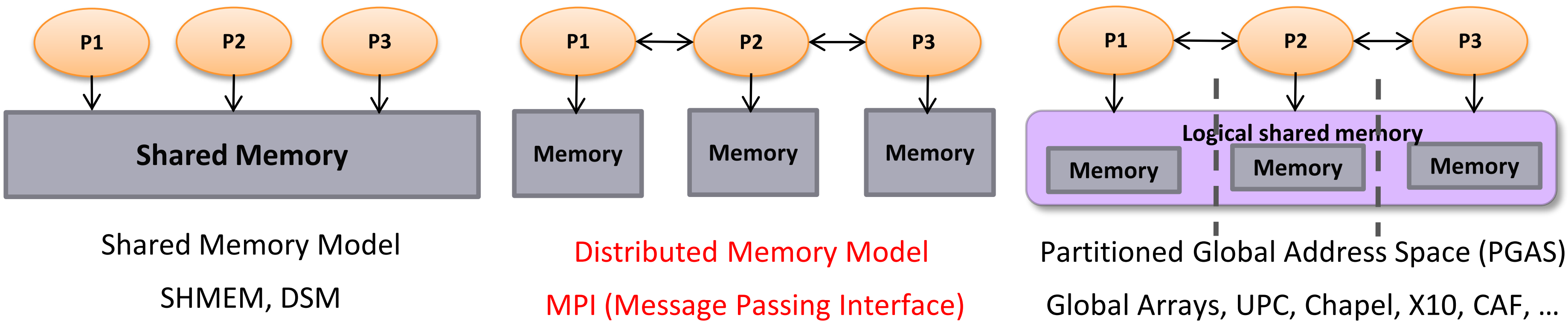
Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Parallel Programming Models Overview



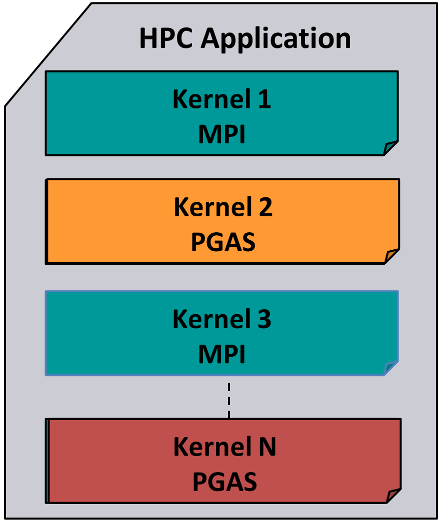
- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Partitioned Global Address Space (PGAS) Models

- Key features
  - Simple shared memory abstractions
  - Light weight one-sided communication
  - Easier to express irregular communication
- Different approaches to PGAS
  - Languages
    - Unified Parallel C (UPC)
    - Co-Array Fortran (CAF)
    - X10
    - Chapel
  - Libraries
    - OpenSHMEM
    - UPC++
    - Global Arrays

Hybrid (MPI+PGAS) Programming

- Application sub-kernels can be re-written in MPI/PGAS based on communication characteristics
- Benefits:
  - Best of Distributed Computing Model
  - Best of Shared Memory Computing Model
- Exascale Roadmap*:
  - "Hybrid Programming is a practical way to program exascale systems"



**** The International Exascale Software Roadmap, Dongarra, J., Beckman, P. et al., Volume 25, Number 1, 2011, International Journal of High Performance Computer Applications, ISSN 1094-3420***

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, 10-40Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Available since 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - **Used by more than 2,550 organizations in 79 countries**
  - **More than 360,000 (> 0.36 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '15 ranking)
    - 10^{th} ranked 519,640-core cluster (Stampede) at TACC
    - 13^{th} ranked 185,344-core cluster (Pleiades) at NASA
    - 25^{th} ranked 76,032-core cluster (Tsubame 2.5) at Tokyo Institute of Technology and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3^{rd} in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Stampede at TACC (10^{th} in Nov'15, 519,640 cores, 5.168 Plops)

MVAPICH2 Architecture

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 Technology
(InfiniBand, iWARP, RoCE, OmniPath)

Transport Protocols: RC, XRC, UD, DC

Modern Features: UMR, ODP*, SR-IOV, Multi Rail

Support for Modern Multi-/Many-core Architectures
(Intel-Xeon, OpenPower, Xeon-Phi (MIC, KNL*), NVIDIA GPGPU)

Transport Mechanisms: Shared Memory, CMA, IVSHMEM

Modern Features: MCDRAM*, NVLink*, CAPI*

* Upcoming

MVAPICH2 Software Family

Requirements	MVAPICH2 Library to use
MPI with IB, iWARP and RoCE	MVAPICH2
Advanced MPI, OSU INAM, PGAS and MPI+PGAS with IB and RoCE	MVAPICH2-X
MPI with IB & GPU	MVAPICH2-GDR
MPI with IB & MIC	MVAPICH2-MIC
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA

MVAPICH2 2.2rc1

- Released on 03/30/2016
- Major Features and Enhancements
  - Based on MPICH-3.1.4
  - Support for OpenPower architecture
    - Optimized inter-node and intra-node communication
  - Support for Intel Omni-Path architecture
    - Thanks to Intel for contributing the patch
    - Introduction of a new PSM2 channel for Omni-Path
  - Support for RoCEv2
  - Architecture detection for PSC Bridges system with Omni-Path
  - Enhanced startup performance and reduced memory footprint for storing InfiniBand end-point information with SLURM
    - Support for shared memory based PMI operations
    - Availability of an updated patch from the MVAPICH project website with this support for SLURM installations
  - Optimized pt-to-pt and collective tuning for Chameleon InfiniBand systems at TACC/UoC
  - Enable affinity by default for TrueScale(PSM) and Omni-Path(PSM2) channels
  - Enhanced tuning for shared-memory based MPI_Bcast
  - Enhanced debugging support and error messages
  - Update to hwloc version 1.11.2

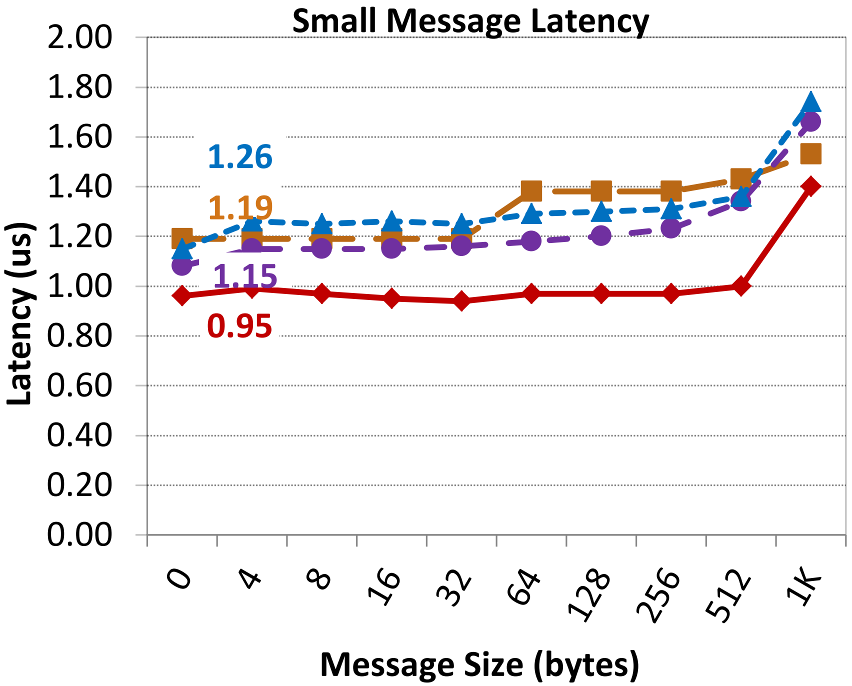
MVAPICH2-X 2.2rc1

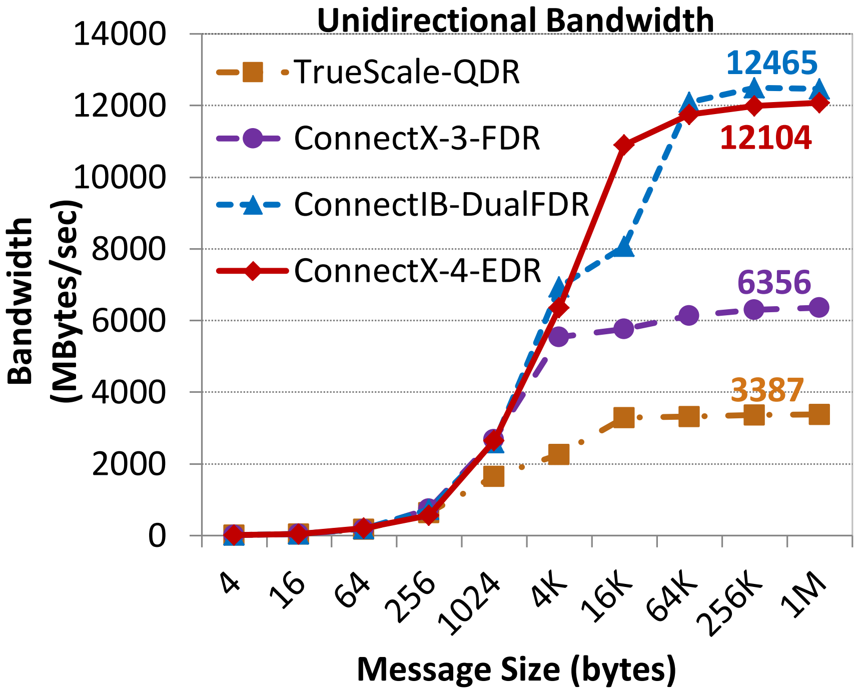
- Released on 03/30/2016
- Introducing UPC++ Support
  - Based on Berkeley UPC++ v0.1
  - Introduce UPC++ level support for new scatter collective operation (upcxx_scatter)
  - Optimized UPC collectives (improved performance for upcxx_reduce, upcxx_bcast, upcxx_gather, upcxx_allgather, upcxx_alltoall)
- MPI Features
  - Based on MVAPICH2 2.2rc1 (OFA-IB-CH3 interface)
  - Support for OpenPower, Intel Omni-Path, and RoCE v2
- UPC Features
  - Based on GASNET v1.26
  - Support for OpenPower, Intel Omni-Path, and RoCE v2
- OpenSHMEM Features
  - Support for OpenPower and RoCE v2
- CAF Features
  - Support for RoCE v2
- Hybrid Program Features
  - Introduce support for hybrid MPI+UPC++ applications
  - Support OpenPower architecture for hybrid MPI+UPC and MPI+OpenSHMEM applications
- Unified Runtime Features
  - Based on MVAPICH2 2.2rc1 (OFA-IB-CH3 interface). All the runtime features enabled by default in OFA-IB-CH3 and OFA-IB-RoCE interface of MVAPICH2 2.2rc1 are available in MVAPICH2-X 2.2rc1
  - Introduce support for UPC++ and MPI+UPC++ programming models
- Support for OSU InfiniBand Network Analysis and Management (OSU INAM) Tool v0.9
  - Capability to profile and report process to node communication matrix for MPI processes at user specified granularity in conjunction with OSU INAM
  - Capability to classify data flowing over a network link at job level and process level granularity in conjunction with OSU INAM

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided RMA)
  - Support for advanced IB mechanisms (UMR and ODP)
  - Extremely minimal memory footprint
- Collective communication
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

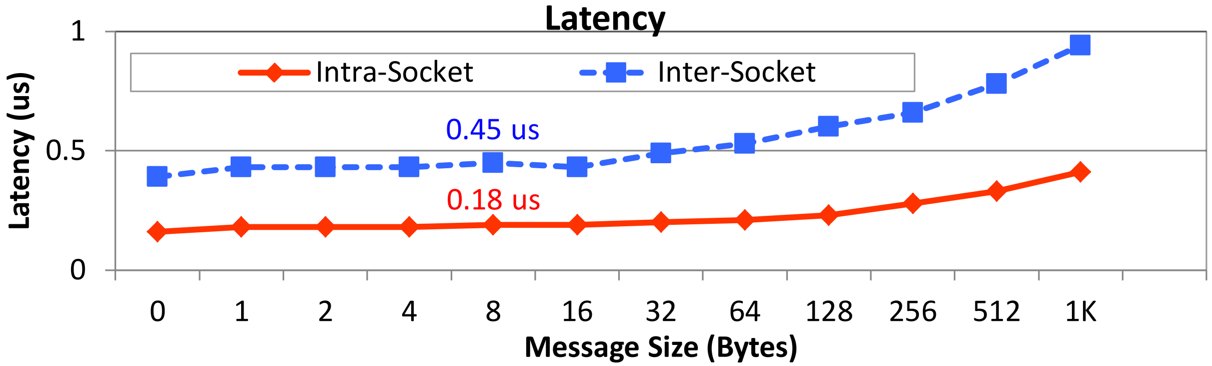
Latency & Bandwidth: MPI over IB with MVAPICH2





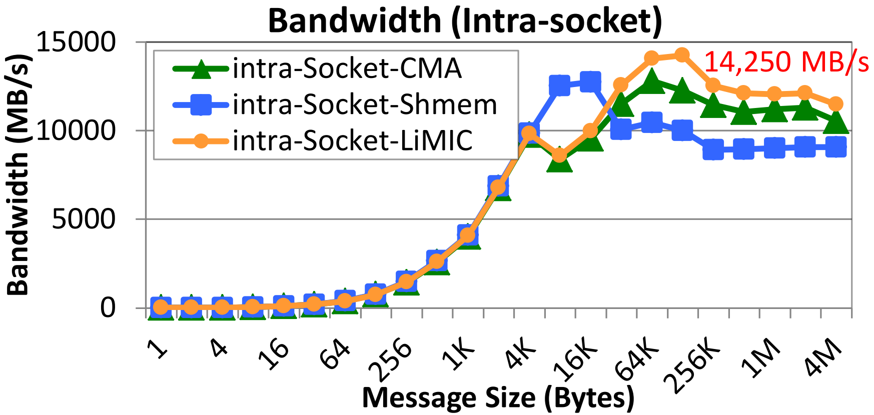
TrueScale-QDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectX-4-EDR - 2.8 GHz Deca-core (Haswell) Intel PCI Gen3 Back-to-back

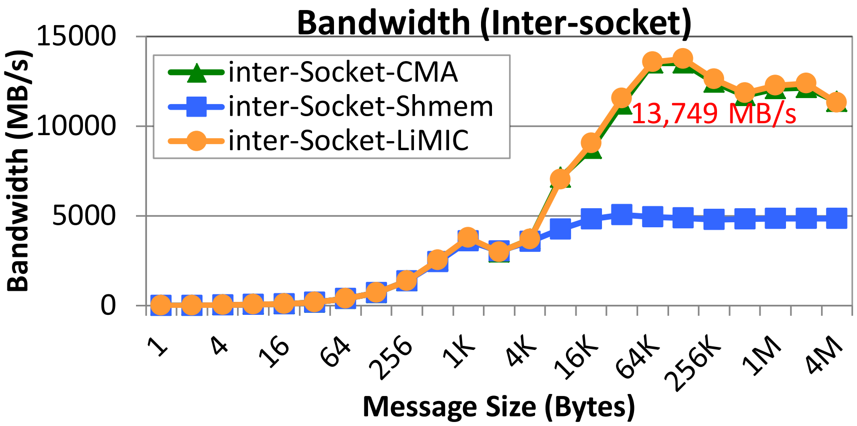
MVAPICH2 Two-Sided Intra-Node Performance
(Shared memory and Kernel-based Zero-copy Support (LiMIC and CMA))



Latest MVAPICH2 2.2rc1

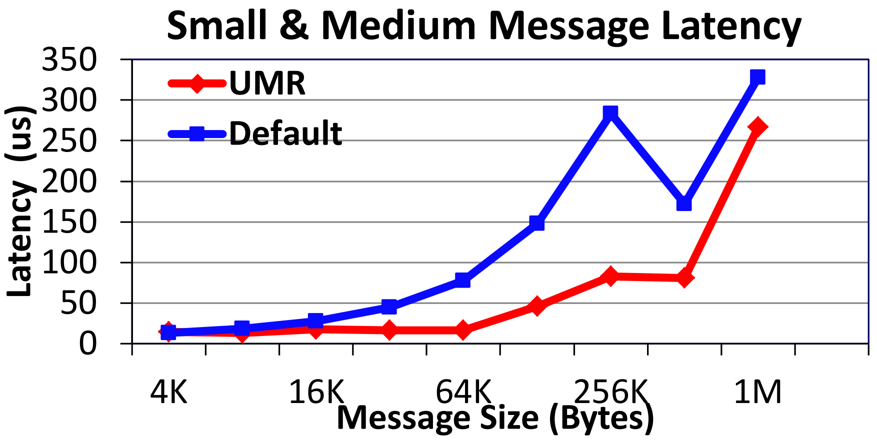
Intel Ivy-bridge

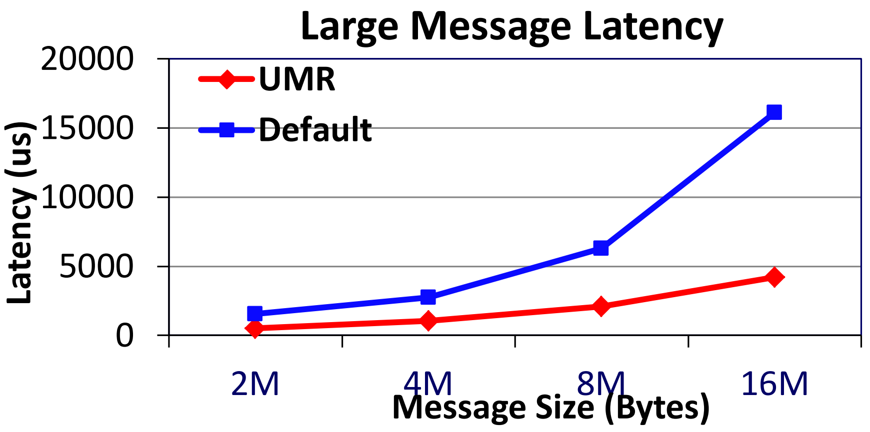




User-mode Memory Registration (UMR)

- Introduced by Mellanox to support direct local and remote noncontiguous memory access
  - Avoid packing at sender and unpacking at receiver
- Available since MVAPICH2-X 2.2b



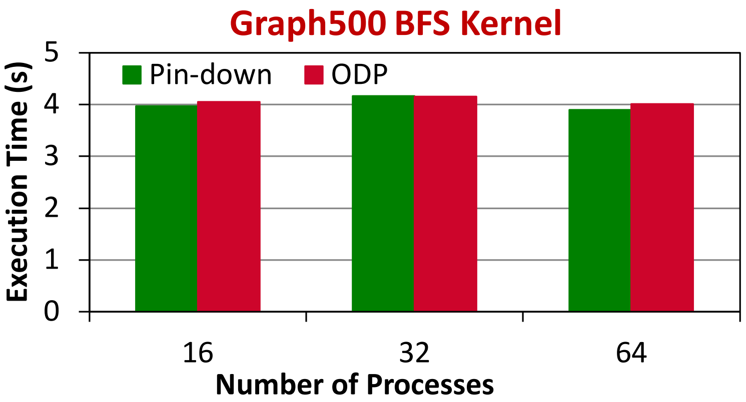


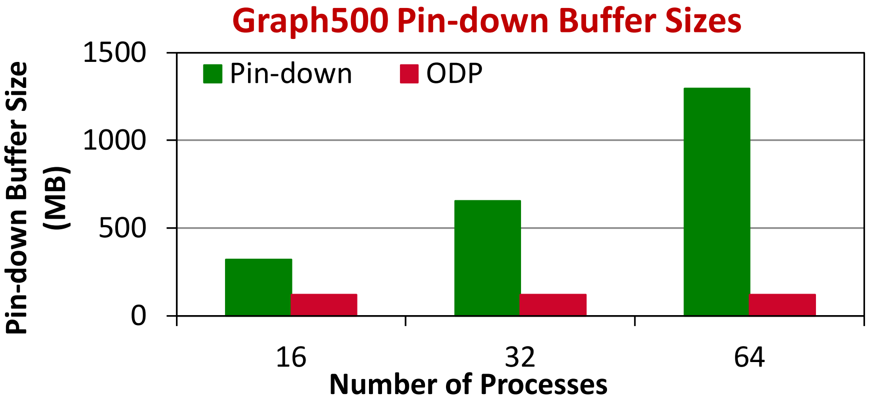
Connect-IB (54 Gbps): 2.8 GHz Dual Ten-core (IvyBridge) Intel PCI Gen3 with Mellanox IB FDR switch

M. Li, H. Subramoni, K. Hamidouche, X. Lu and D. K. Panda, High Performance MPI Datatype Support with User-mode Memory Registration: Challenges, Designs and Benefits, CLUSTER, 2015

On-Demand Paging (ODP)

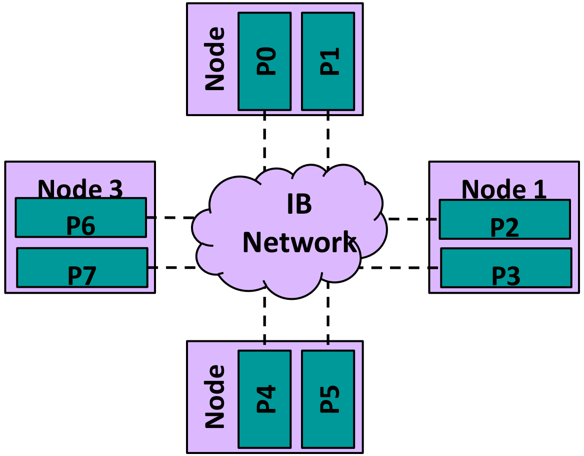
- Introduced by Mellanox to support direct remote memory access without pinning
- Memory regions paged in/out dynamically by the HCA/OS
- Size of registered buffers can be larger than physical memory
- Will be available in future MVAPICH2 release





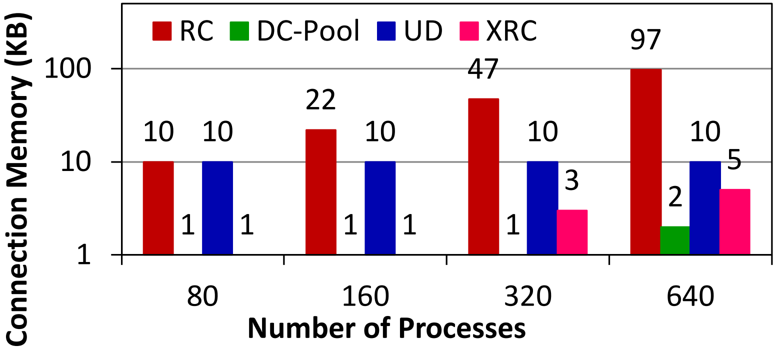
Connect-IB (54 Gbps): 2.6 GHz Dual Octa-core (SandyBridge) Intel PCI Gen3 with Mellanox IB FDR switch

Minimizing Memory Footprint by Direct Connect (DC) Transport

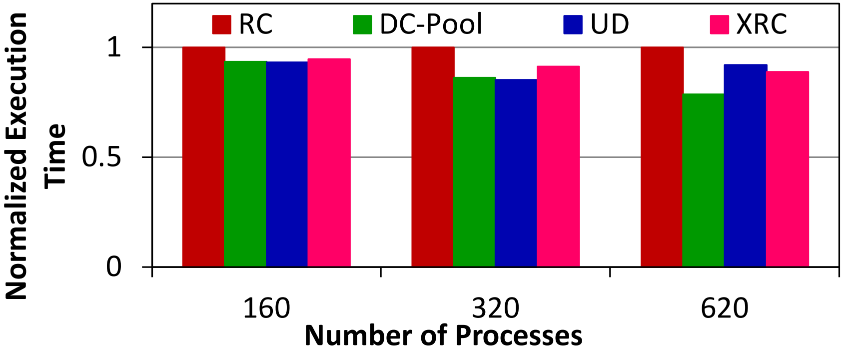


- Constant connection cost (*One QP for any peer*)
- Full Feature Set (RDMA, Atomics etc)
- Separate objects for send (DC Initiator) and receive (DC Target)
  - DC Target identified by "DCT Number"
  - Messages routed with (DCT Number, LID)
  - Requires same "DC Key" to enable communication
- Available since MVAPICH2-X 2.2a

Memory Footprint for Alltoall



NAMD - Apoa1: Large data set

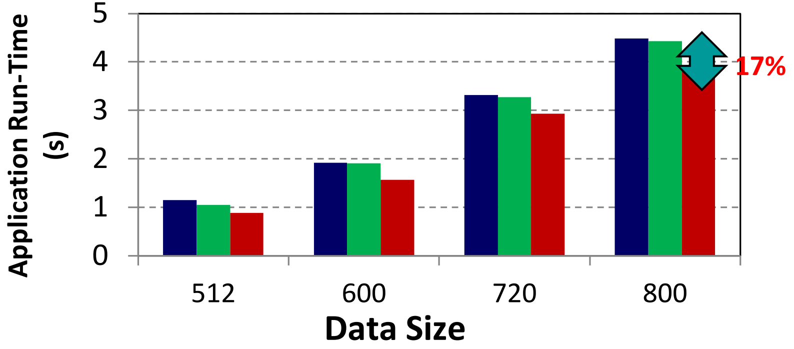


H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty and D. K. Panda, Designing MPI Library with Dynamic Connected Transport (DCT) of InfiniBand : Early Experiences. IEEE International Supercomputing Conference (ISC '14)

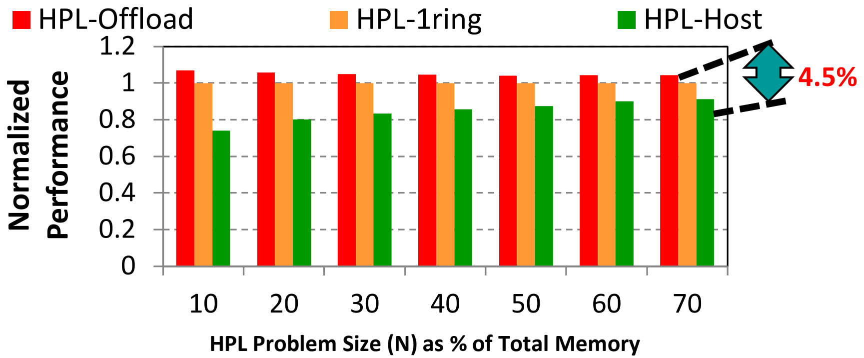
Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

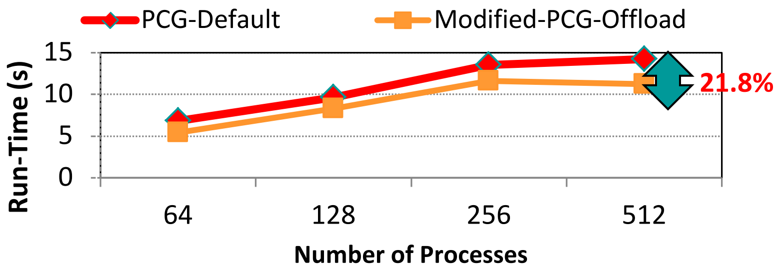
Co-Design with MPI-3 Non-Blocking Collectives and Collective Offload Co-Direct Hardware (Available since MVAPICH2-X 2.2a)



Modified P3DFFT with Offload-Alltoall does up to 17% better than default version (128 Processes)



Modified HPL with Offload-Bcast does up to 4.5% better than default version (512 Processes)



Modified Pre-Conjugate Gradient Solver with Offload-Allreduce does up to 21.8% better than default version

K. Kandalla, et. al.. High-Performance and Scalable Non-Blocking All-to-All with Collective Offload on InfiniBand Clusters: A Study with Parallel 3D FFT,

K. Kandalla, et. al, Designing Non-blocking Broadcast with Collective Offload on InfiniBand Clusters: A Case Study with HPL, HotI 2011

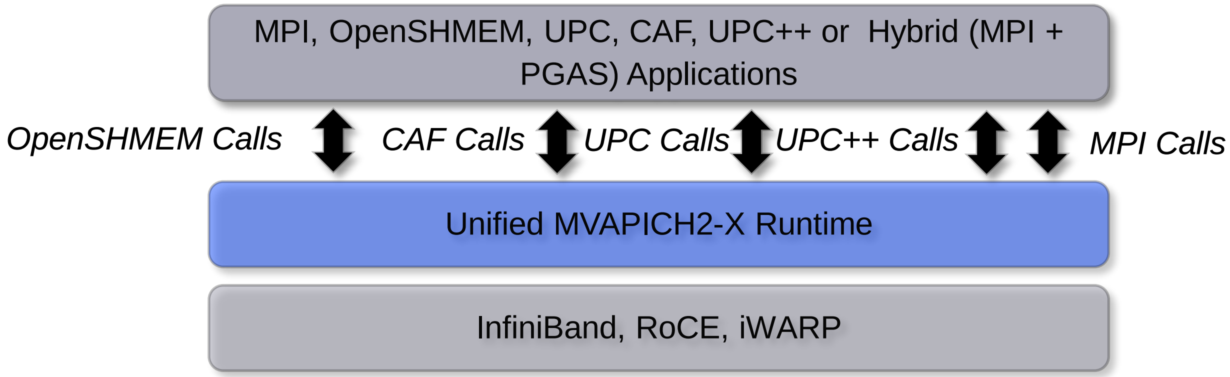
K. Kandalla, et. al., Designing Non-blocking Allreduce with Collective Offload on InfiniBand Clusters: A Case Study with Conjugate Gradient Solvers, IPDPS '12

Can Network-Offload based Non-Blocking Neighborhood MPI Collectives Improve Communication Overheads of Irregular Graph Algorithms? K. Kandalla, A. Buluc, H. Subramoni, K. Tomko, J. Vienne, L. Oliker, and D. K. Panda, IWPAPS' 12

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

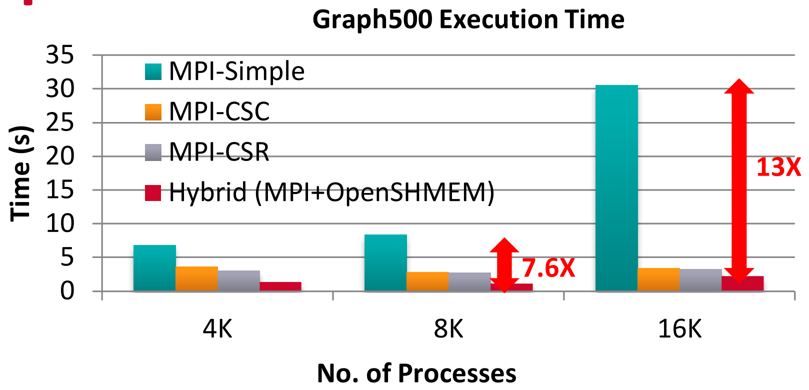
- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

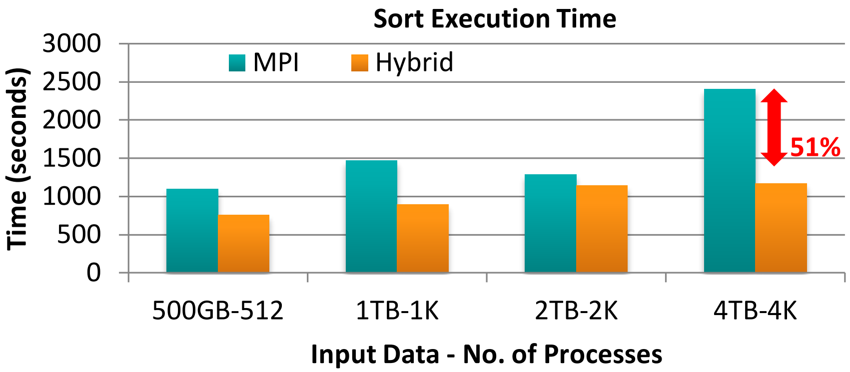
MVAPICH2-X for Advanced MPI and Hybrid MPI + PGAS Applications



- Unified communication runtime for MPI, UPC, OpenSHMEM, CAF, UPC++ available with MVAPICH2-X 1.9 onwards! (since 2012)
- UPC++ support is available in MVAPICH2-X 2.2RC1
- Feature Highlights
  - Supports MPI(+OpenMP), OpenSHMEM, UPC, CAF, UPC++, MPI(+OpenMP) + OpenSHMEM, MPI(+OpenMP) + UPC
  - MPI-3 compliant, OpenSHMEM v1.0 standard compliant, UPC v1.2 standard compliant (with initial support for UPC 1.3), CAF 2008 standard (OpenUH), UPC++
  - Scalable Inter-node and intra-node communication – point-to-point and collectives

Application Level Performance with Graph500 and Sort





- Performance of Hybrid (MPI+ OpenSHMEM) Graph500 Design
  - 8,192 processes
    - **2.4X** improvement over MPI-CSR
    - **7.6X** improvement over MPI-Simple
  - 16,384 processes
    - **1.5X** improvement over MPI-CSR
    - **13X** improvement over MPI-Simple

- Performance of Hybrid (MPI+OpenSHMEM) Sort Application
  - 4,096 processes, 4 TB Input Size
    - MPI – 2408 sec; 0.16 TB/min
    - Hybrid – 1172 sec; 0.36 TB/min
    - **51%** improvement over MPI-design

J. Jose, K. Kandalla, S. Potluri, J. Zhang and D. K. Panda, Optimizing Collective Communication in OpenSHMEM, Int'l Conference on Partitioned Global Address Space Programming Models (PGAS '13), October 2013.

J. Jose, S. Potluri, K. Tomko and D. K. Panda, Designing Scalable Graph500 Benchmark with Hybrid MPI+OpenSHMEM Programming Models, International Supercomputing Conference (ISC'13), June 2013

J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

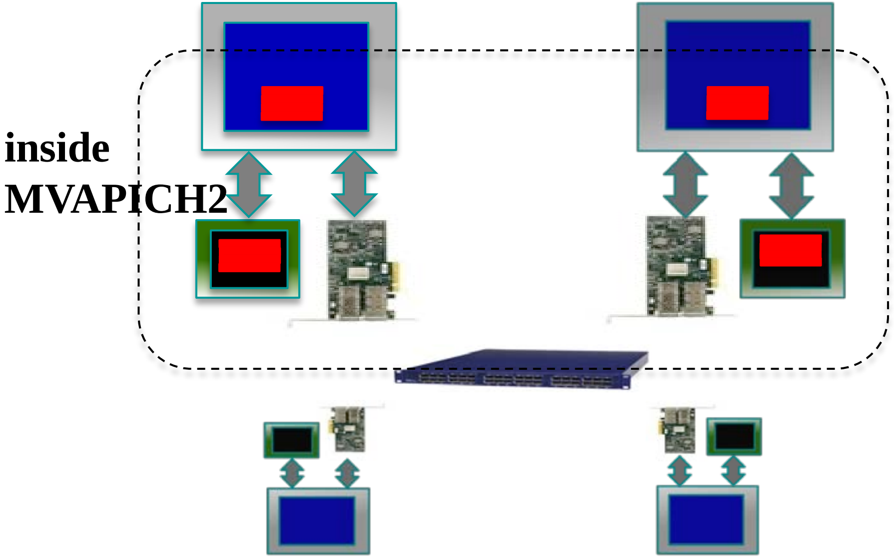
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

MPI_Send(s_devbuf, size, …);

At Receiver:

MPI_Recv(r_devbuf, size, …);

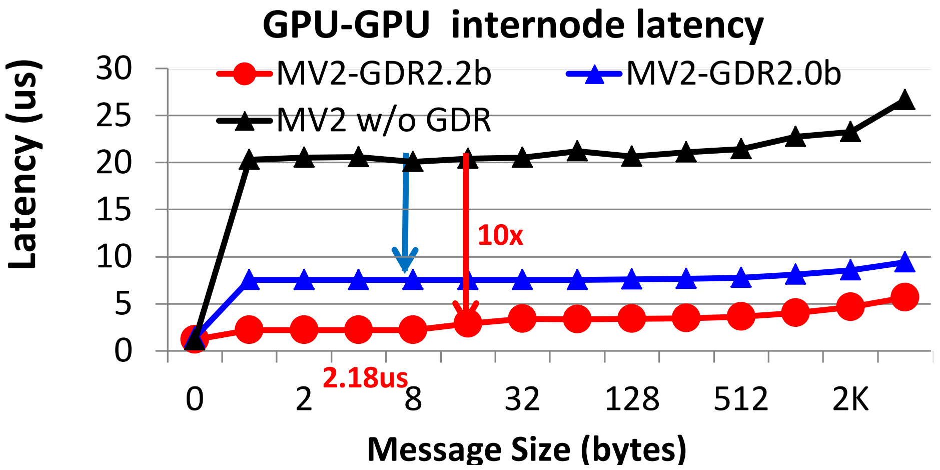


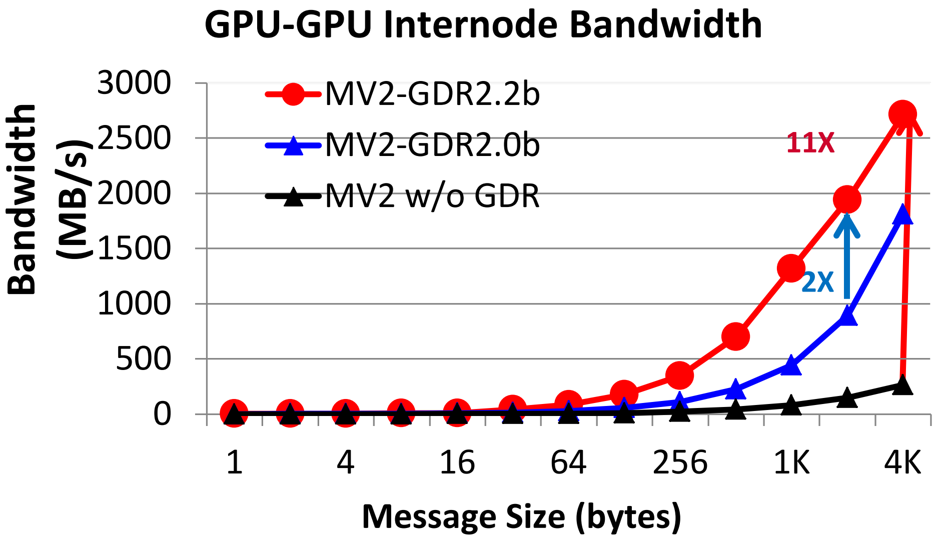
High Performance and High Productivity

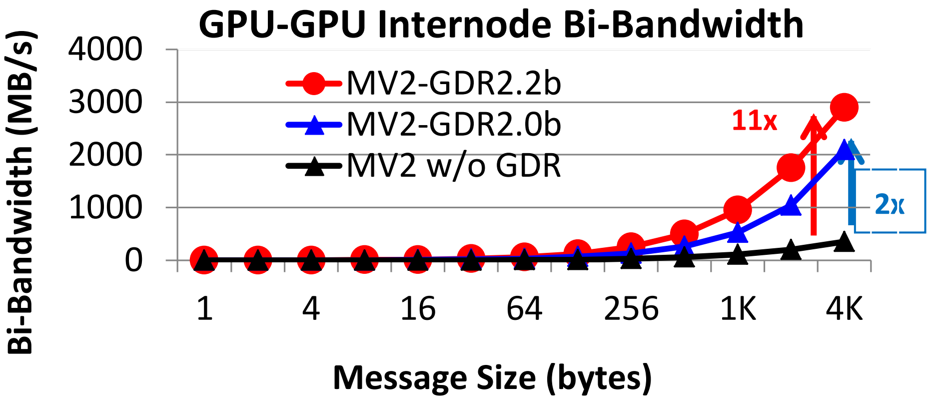
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.2 Releases

- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage of CUDA IPC (available since CUDA 4.1) in intra-node communication for multiple GPU adapters/node
- Optimized and tuned collectives for GPU device buffers
- MPI datatype support for point-to-point and collective communication from GPU device buffers

Performance of MVAPICH2-GPU with GPU-Direct RDMA (GDR)

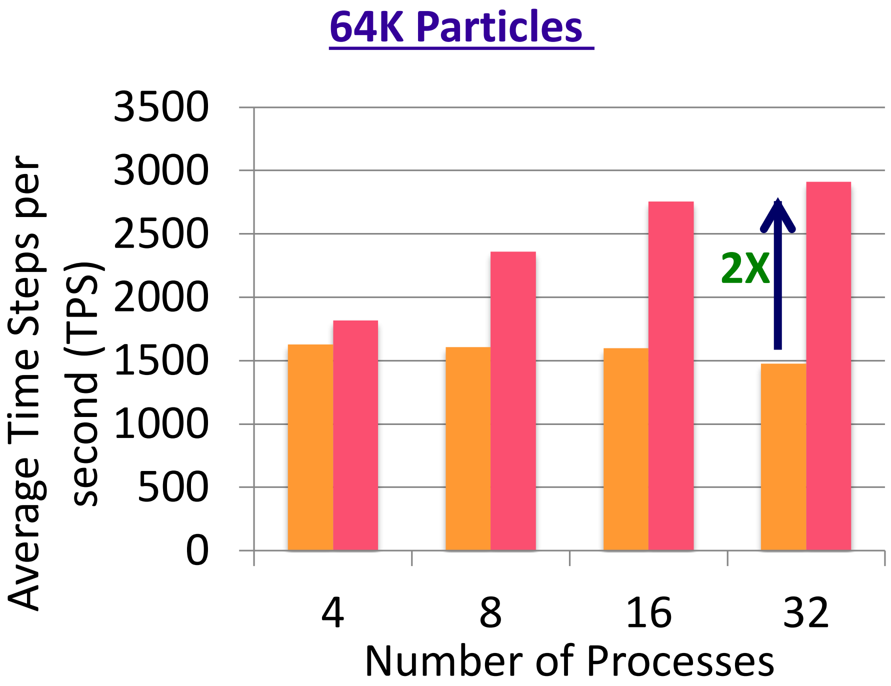


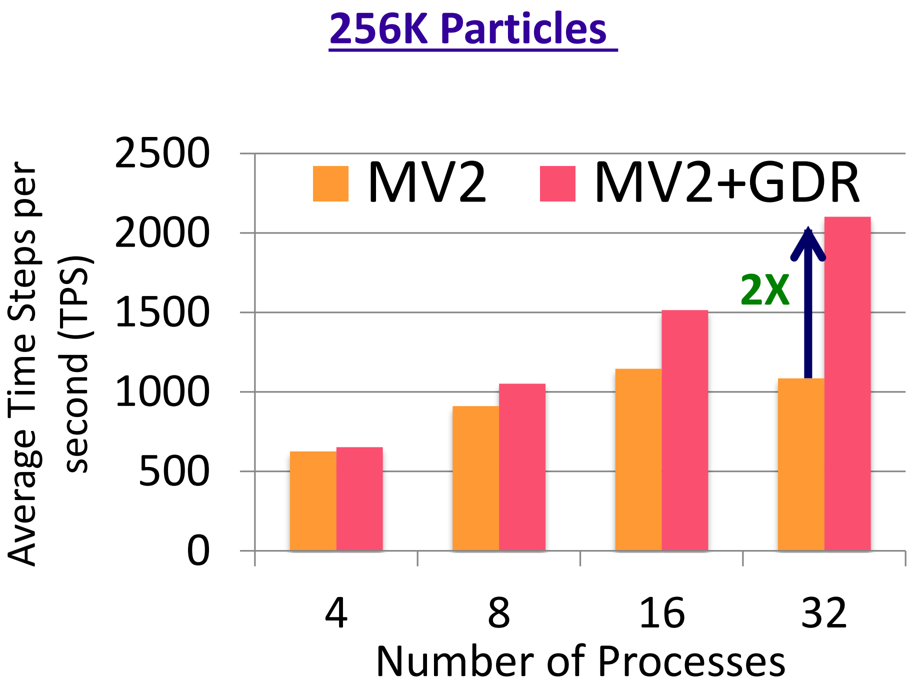




MVAPICH2-GDR-2.2b
Intel Ivy Bridge (E5-2680 v2) node - 20 cores
NVIDIA Tesla K40c GPU
Mellanox Connect-IB Dual-FDR HCA
CUDA 7
Mellanox OFED 2.4 with GPU-Direct-RDMA

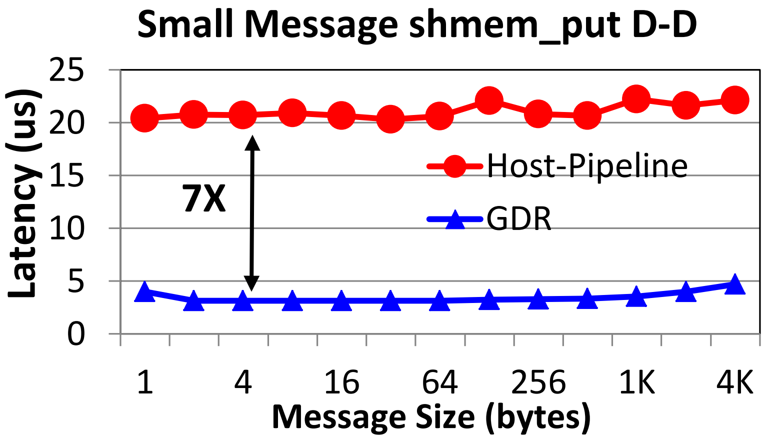
Application-Level Evaluation (HOOMD-blue)

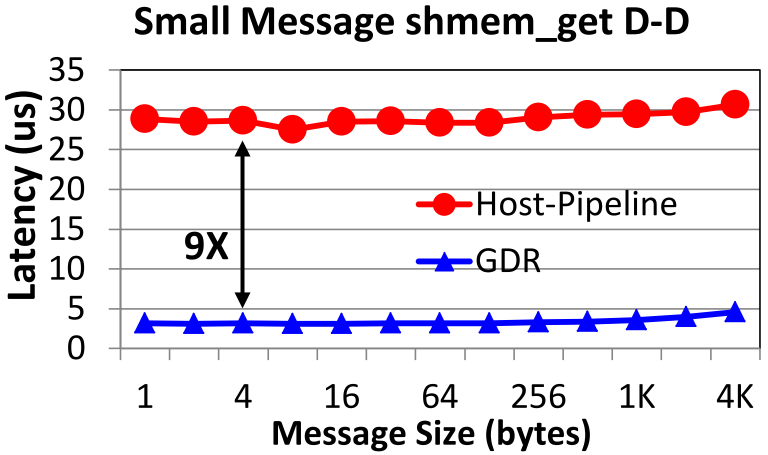


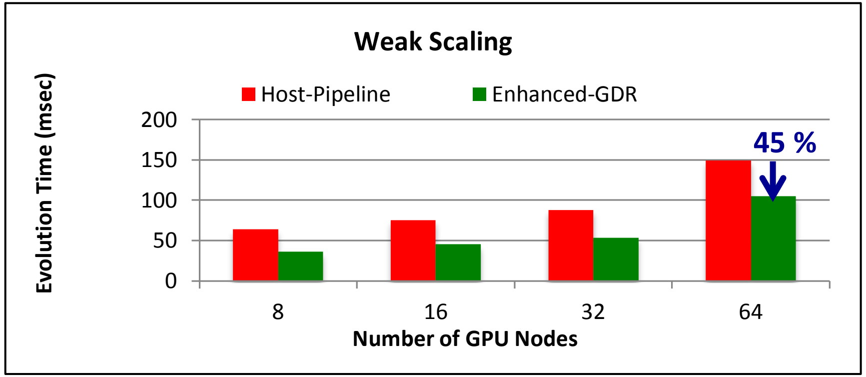


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

Exploiting GDR for OpenSHMEM







GPULBM: 64x64x64

- Introduced CUDA-aware OpenSHMEM
- GDR for small/medium message sizes
- Host-staging for large message to avoid PCIe bottlenecks
- Hybrid design brings best of both
- 3.13 us Put latency for 4B (7X improvement) and 4.7 us latency for 4KB
- 9X improvement for Get latency of 4B

K. Hamidouche, A. Venkatesh, A. Awan, H. Subramoni, C. Ching and D. K. Panda, Exploiting GPUDirect RDMA in Designing High Performance OpenSHMEM for GPU Clusters. IEEE Cluster 2015. Also accepted for a special issue of Journal PARCO

Will be available in future MVAPICH2-X Release

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

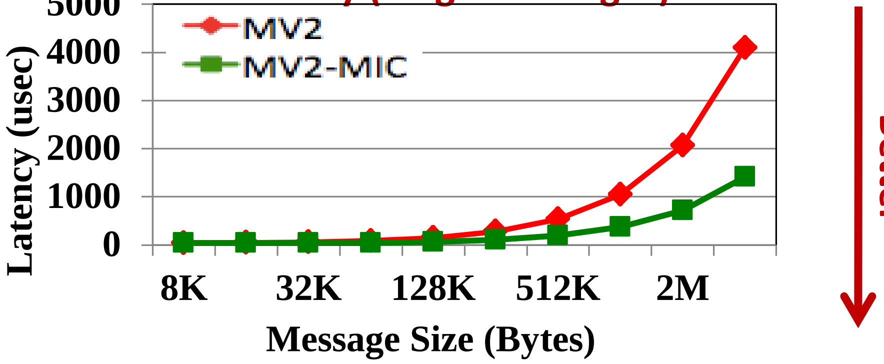
MVAPICH2-MIC 2.0 Design for Clusters with IB and MIC

- Offload Mode
- Intranode Communication
  - Coprocessor-only and Symmetric Mode
- Internode Communication
  - Coprocessors-only and Symmetric Mode
- Multi-MIC Node Configurations
- Running on three major systems
  - Stampede, Blueridge (Virginia Tech) and Beacon (UTK)

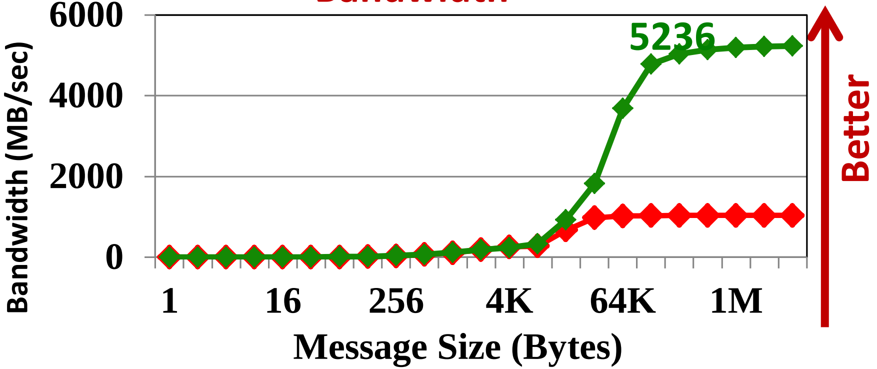
MIC-Remote-MIC P2P Communication with Proxy-based Communication

Intra-socket P2P

Latency (Large Messages)

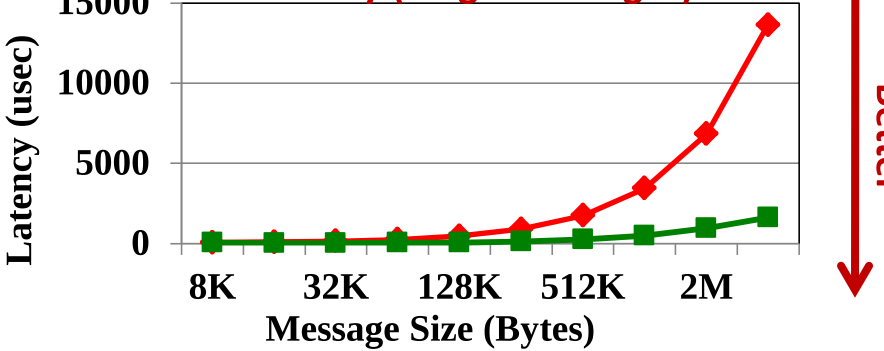


Bandwidth

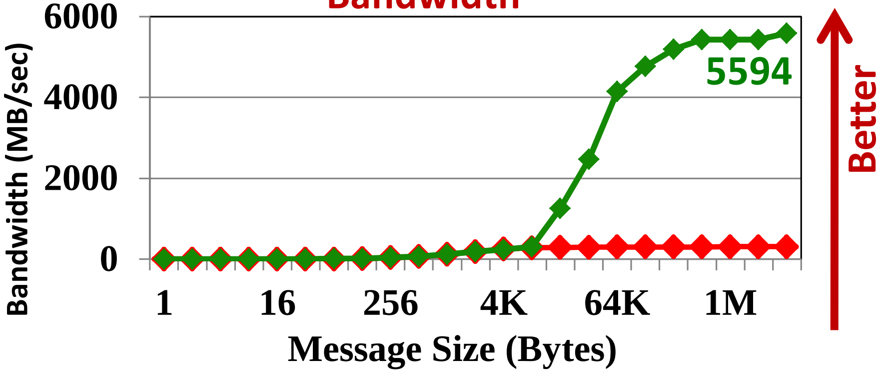


Inter-socket P2P

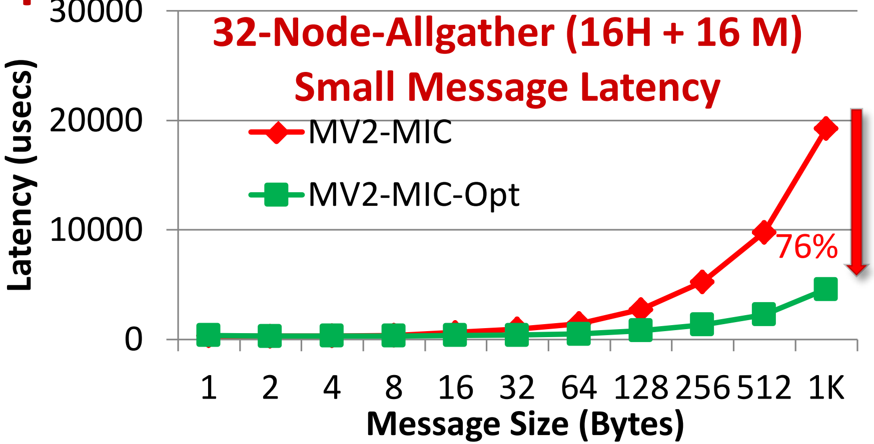
Latency (Large Messages)

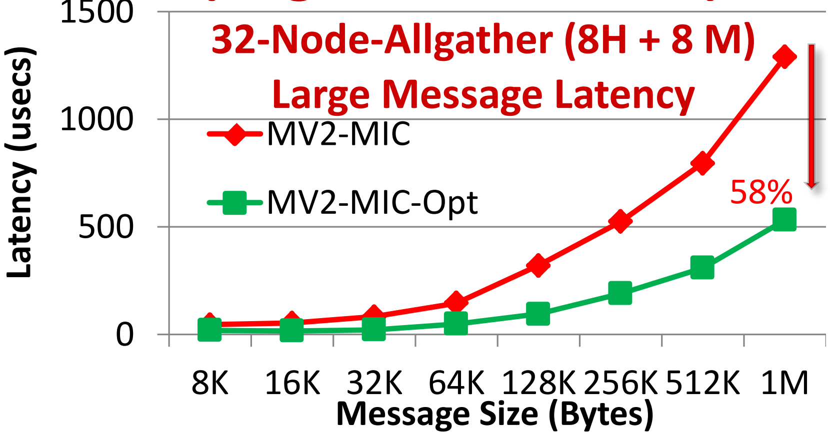


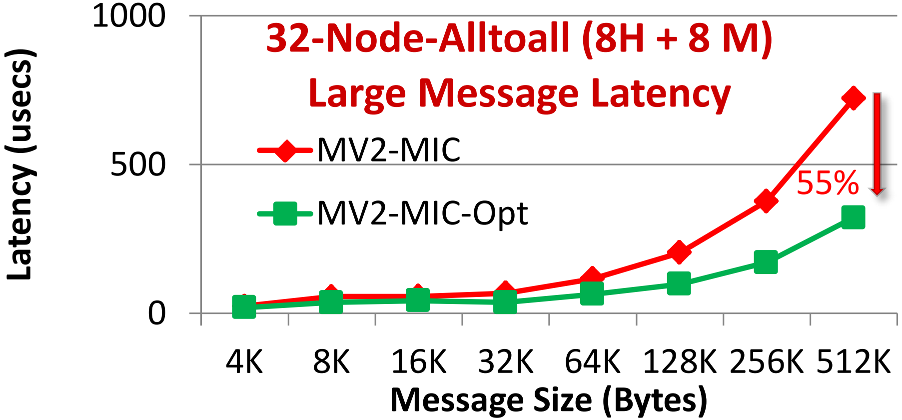
Bandwidth

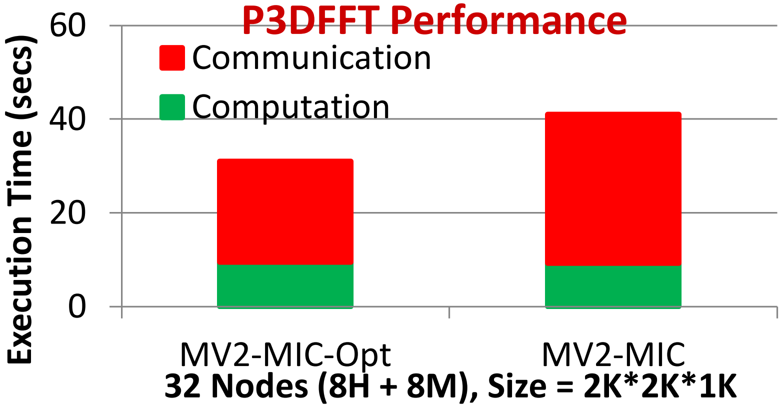


Optimized MPI Collectives for MIC Clusters (Allgather & Alltoall)









A. Venkatesh, S. Potluri, R. Rajachandrasekar, M. Luo, K. Hamidouche and D. K. Panda - High Performance Alltoall and Allgather designs for InfiniBand MIC Clusters; IPDPS'14, May 2014

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

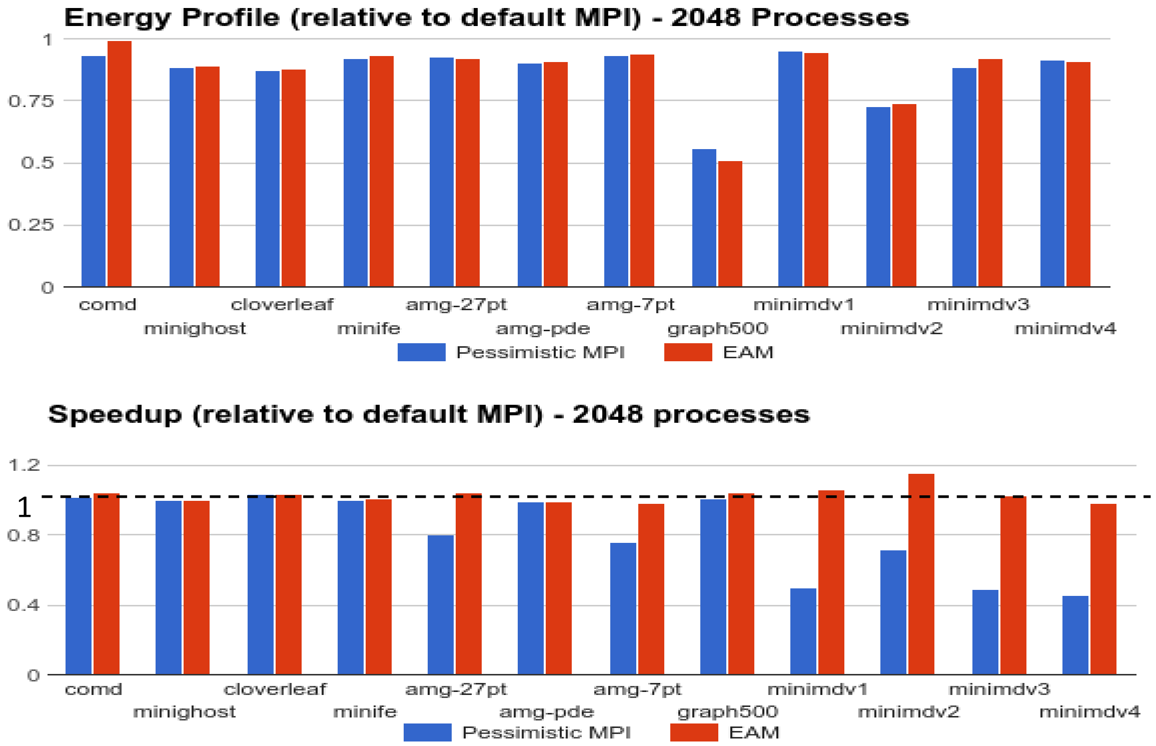
Energy-Aware MVAPICH2 & OSU Energy Management Tool (OEMT)

- MVAPICH2-EA 2.1 (Energy-Aware)
  - A white-box approach
  - New Energy-Efficient communication protocols for pt-pt and collective operations
  - Intelligently apply the appropriate Energy saving techniques
  - Application oblivious energy saving

- OEMT
  - A library utility to measure energy consumption for MPI applications
  - Works with all MPI runtimes
  - PRELOAD option for precompiled applications
  - Does not require ROOT permission:
    - A safe kernel module to read only a subset of MSRs

MVAPICH2-EA: Application Oblivious Energy-Aware-MPI (EAM)

- An energy efficient runtime that provides energy savings without application knowledge
- Uses automatically and transparently the best energy lever
- Provides guarantees on maximum degradation with 5-41% savings at <= 5% degradation
- Pessimistic MPI applies energy reduction lever to each MPI call



A Case for Application-Oblivious Energy-Efficient MPI Runtime A. Venkatesh, A. Vishnu, K. Hamidouche, N. Tallent, D. K. Panda, D. Kerbyson, and A. Hoise, Supercomputing '15, Nov 2015 *[Best Student Paper Finalist]*

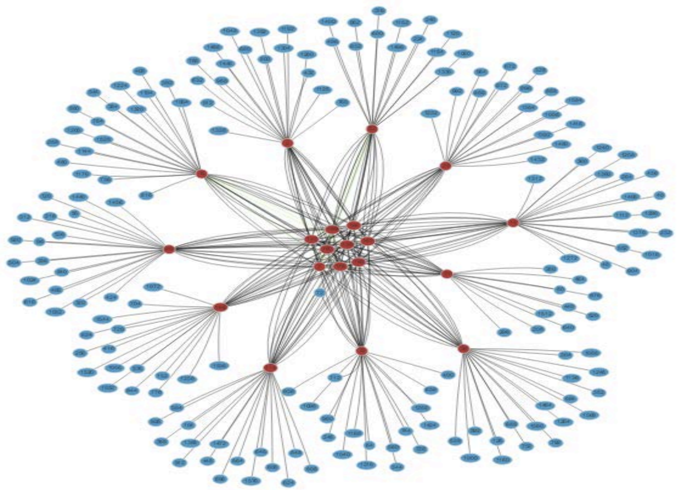
Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Collective communication
  - Offload and Non-blocking
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, …)
- Integrated Support for GPGPUs
- Integrated Support for MICs
- Energy-Awareness
- InfiniBand Network Analysis and Monitoring (INAM)

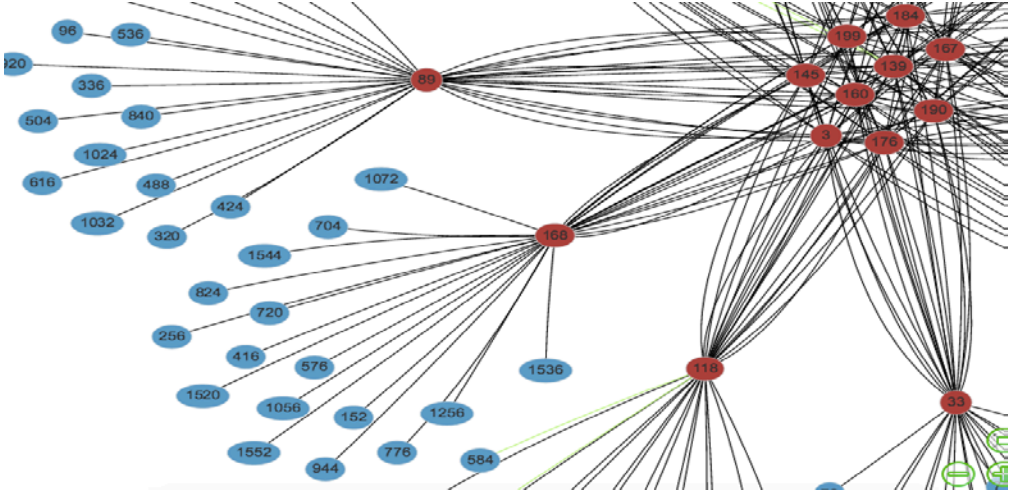
Overview of OSU INAM

- A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
  - http://mvapich.cse.ohio-state.edu/userguide/osu-inam/
- Monitors IB clusters in real time by querying various subnet management entities and gathering input from the MPI runtimes
- Capability to analyze and profile node-level, job-level and process-level activities for MPI communication (Point-to-Point, Collectives and RMA)
- Ability to filter data based on type of counters using "drop down" list
- Remotely monitor various metrics of MPI processes at user specified granularity
- "Job Page" to display jobs in ascending/descending order of various performance metrics in conjunction with MVAPICH2-X
- Visualize the data transfer happening in a "live" or "historical" fashion for entire network, job or set of nodes

OSU INAM – Network Level View



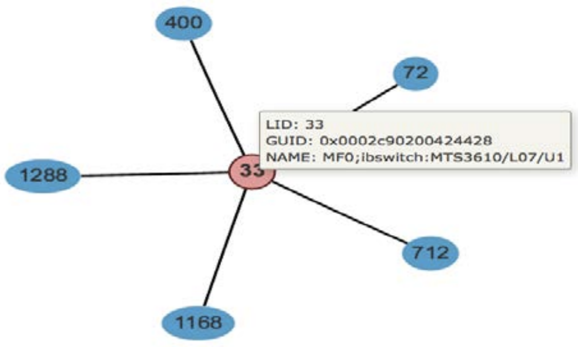
Full Network (152 nodes)



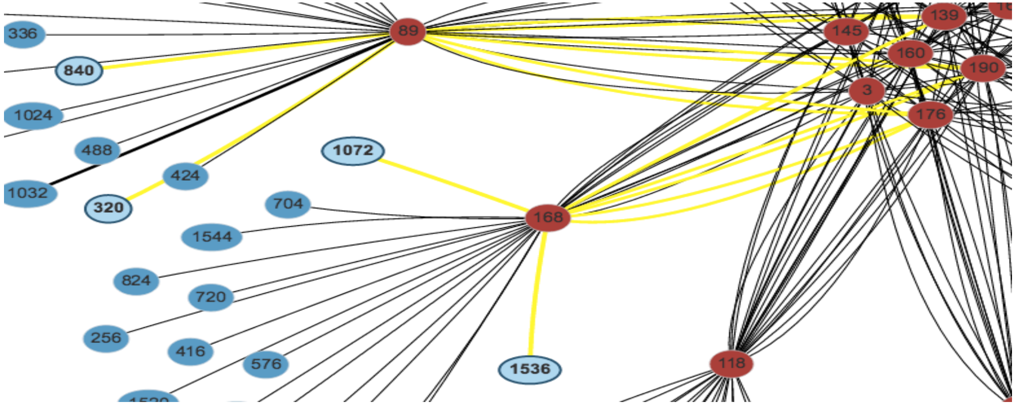
Zoomed-in View of the Network

- Show network topology of large clusters
- Visualize traffic pattern on different links
- Quickly identify congested links/links in error state
- See the history unfold – play back historical state of the network

OSU INAM – Job and Node Level Views



Visualizing a Job (5 Nodes)



Finding Routes Between Nodes

- Job level view
  - Show different network metrics (load, error, etc.) for any live job
  - Play back historical data for completed jobs to identify bottlenecks
- Node level view provides details per process or per node
  - CPU utilization for each rank/node
  - Bytes sent/received for MPI operations (pt-to-pt, collective, RMA)
  - Network metrics (e.g. XmitDiscard, RcvError) per rank/node

MVAPICH2 – Plans for Exascale

- Performance and Memory scalability toward 1M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF …)
    - MPI + Task*
- Enhanced Optimization for GPU Support and Accelerators
- Taking advantage of advanced features of Mellanox InfiniBand
    - On-Demand Paging (ODP)*
    - Switch-IB2 SHArP*
    - GID-based support*
- Enhanced communication schemes for upcoming architectures
    - Knights Landing with MCDRAM*
    - NVLINK*
    - CAPI*
- Extended topology-aware collectives
- Extended Energy-aware designs and Virtualization Support
- Extended Support for MPI Tools Interface (as in MPI 3.0)
- Extended Checkpoint-Restart and migration support with SCR
- Support for * features will be available in future MVAPICH2 Releases

Funding Acknowledgments

Funding Support by

Office of Science
U.S. DEPARTMENT OF ENERGY

NSF

Mellanox TECHNOLOGIES

CISCO SYSTEMS

QLOGIC

CRAY THE SUPERCOMPUTER COMPANY

intel

LINUX NETWORX

NVIDIA

NetApp

Sun microsystems

Equipment Support by

Mellanox TECHNOLOGIES

intel

AMD

Sun microsystems

APPRO

advanced clustering technologies, inc.

Microway Technology you can count on

NVIDIA

QLOGIC

Personnel Acknowledgments

Current Students

- A. Augustine (M.S.)
- A. Awan (Ph.D.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- N. Islam (Ph.D.)
- M. Li (Ph.D.)
- K. Kulkarni (M.S.)
- M. Rahman (Ph.D.)
- D. Shankar (Ph.D.)
- A. Venkatesh (Ph.D.)
- J. Zhang (Ph.D.)

Current Research Scientists

- H. Subramoni
- X. Lu

Current Senior Research Associate

- K. Hamidouche

Current Post-Doc

- J. Lin
- D. Banerjee

Current Programmer

- J. Perkins

Current Research Specialist

- M. Arnold

Past Students

- P. Balaji (Ph.D.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)

Past Post-Docs

- H. Wang
- X. Besseron
- H.-W. Jin
- M. Luo
- E. Mancini
- S. Marcarelli
- J. Vienne

Past Research Scientist

- S. Sur

Past Programmers

- D. Bureddy

Thank You!

panda@cse.ohio-state.edu

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

The MVAPICH2 Project
http://mvapich.cse.ohio-state.edu/