

The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing

Presentation at OSU Booth (SC '19)

by

Hari Subramoni

The Ohio State University

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

<http://www.cse.ohio-state.edu/~subramon>

Drivers of Modern HPC Cluster Architectures



Multi-core Processors

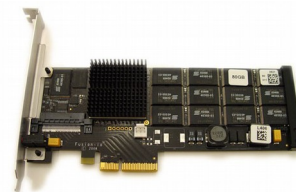
- Multi-core/many-core technologies
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), NVMe-SSD
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)
- Available on HPC Clouds, e.g., Amazon EC2, NSF Chameleon, Microsoft Azure, etc.



High Performance Interconnects - InfiniBand
<1usec latency, 200Gbps Bandwidth>



Accelerators / Coprocessors
high compute density, high performance/watt
>1 TFlop DP on a chip



SSD, NVMe-SSD, NVRAM



Summit



Sierra

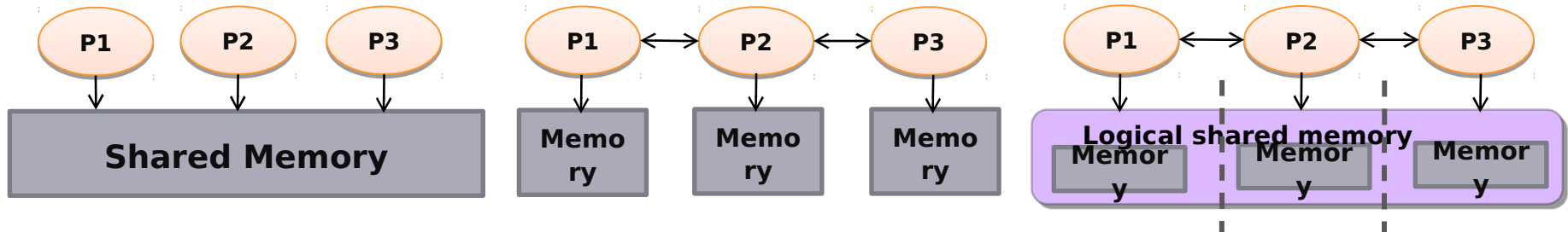


Sunway



K -

Parallel Programming Models Overview



Shared Memory Model
SHMEM, DSM

Distributed Memory Model Partitioned Global Address Space (PGAS)
MPI (Message Passing Interface) Global Arrays, UPC, Chapel, X10, CAF,

- 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

Designing Communication Libraries for Multi-Petaflop and Exaflop Systems: Challenges

Application Kernels/Applications

Middleware

Programming Models

MPI, PGAS (UPC, Global Arrays, OpenSHMEM),
CUDA, OpenMP, OpenACC, Cilk, Hadoop
(MapReduce), Spark (RDD, DAG), etc.

Communication Library or Runtime for Programming Models

Point-to-point
Communication

Collective
Communication

Energy-Awareness

Synchronization and
Locks

I/O and
File
Systems

Fault
Tolerance

Networking Technologies
(InfiniBand, 40/100GigE,
Aries, and OmniPath)

**Multi/Many-core
Architectures**

**Accelerators
(GPU and FPGA)**

**Co-Design
Opportunities
and
Challenges
across
Various
Layers**

**Performance
Scalability
Fault-Resilience**

MPI+X Programming model: Broad Challenges at Exascale

- Scalability for million to billion processors
 - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
 - Scalable job start-up
- Scalable Collective communication
 - Offload
 - Non-blocking
 - Topology-aware
- Balancing intra-node and inter-node communication for next generation nodes (128-1024 cores)
 - Multiple end-points per node
- Support for efficient multi-threading
- Integrated Support for GPGPUs and FPGAs
- Fault-tolerance/resiliency
- QoS support for communication and I/O
- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI+UPC++, MPI + OpenSHMEM, CAF, ...)
- Virtualization
- Energy-Awareness

Overview of the MVAPICH2 Project

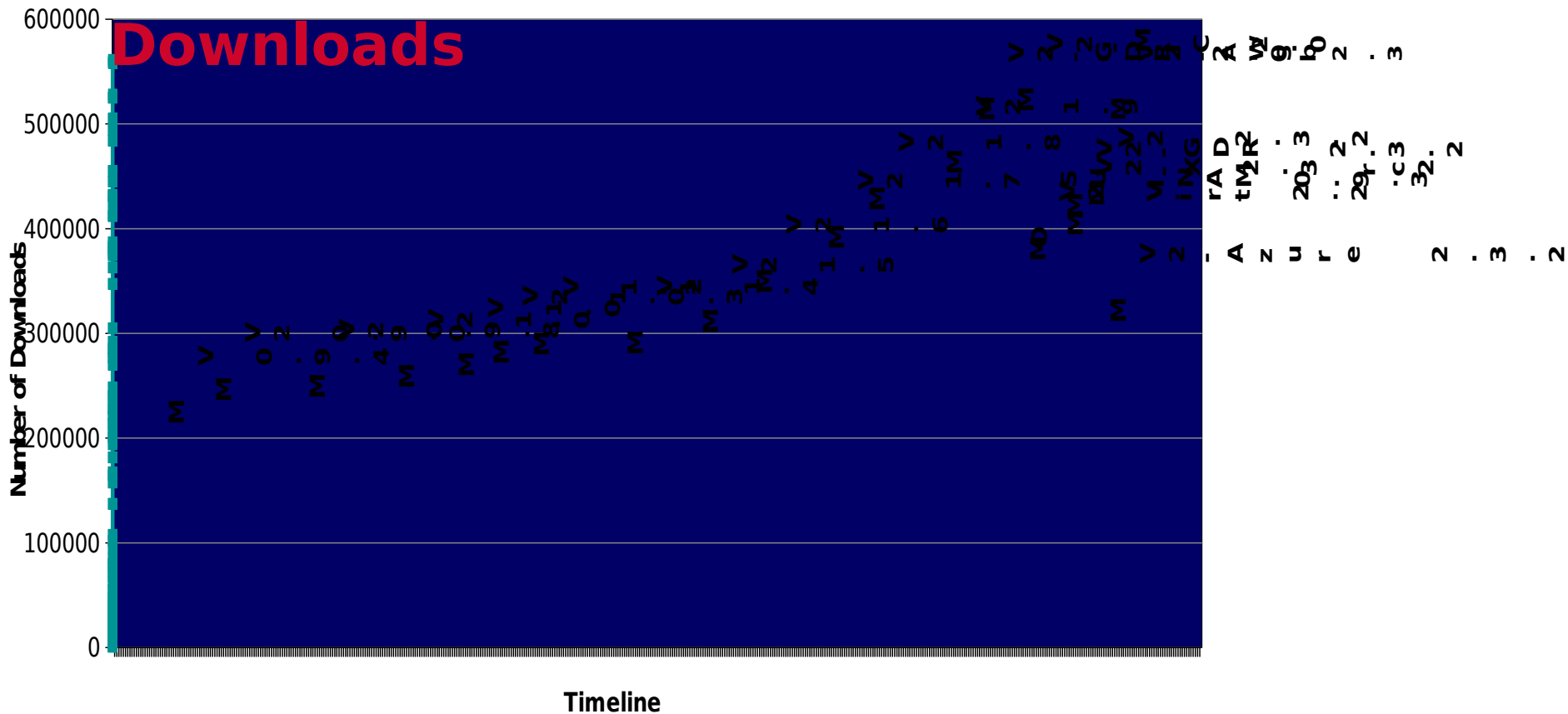
- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
 - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
 - MVAPICH2-X (MPI + PGAS), Available since 2011
 - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
 - Support for Virtualization (MVAPICH2-Virt), Available since 2015
 - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
 - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) s
 - **Used by more than 3,050 organizations in 89 countries**
 - **More than 614,000 (> 0.6 million) downloads from the**
 - Empowering many TOP500 clusters (Nov '18 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
 - 15th, 570,020 cores (Neurion) in South Korea and many others
 - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)



Partner in the TACC Frontera System

– <http://mvapich.cse.ohio-state.edu>

MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family

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 Technologies
(InfiniBand, iWARP, RoCE, Omni-Path, Elastic Fabric)

Transport Protocols

RC

SRD

UD

DC

Modern Features

UMR

ODP

SR-IOV

Multi-Rail

Support for Modern Multi-/Many-core Architectures
(Intel-Xeon, OpenPOWER, Xeon-Phi, ARM, NVIDIA)

Transport Mechanisms

Shared Memory

CMA

IVSHMEM

XPME

Modern Features

Optane*

NVLink

CAP I*

* Upcoming

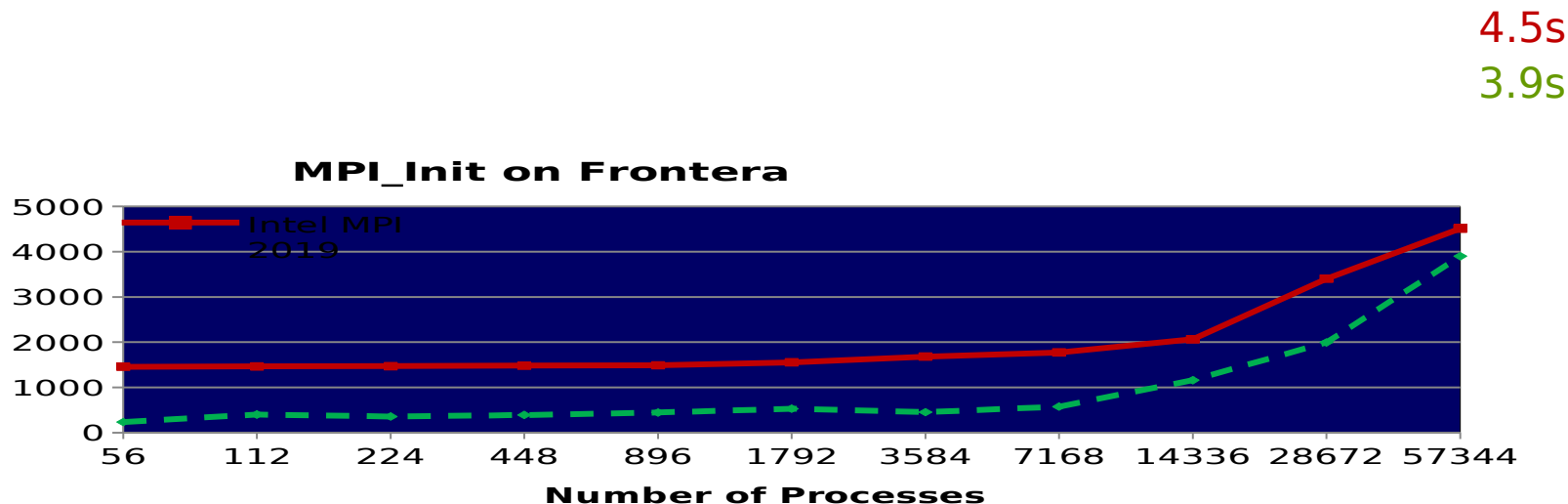
MVAPICH2 Software Family

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

MVAPICH2 Distributions

- MVAPICH2
 - Basic MPI support for IB, Omni-Path, iWARP and RoCE
- MVAPICH2-X
 - Advanced MPI features and support for INAM
 - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
 - Optimized for HPC Clouds with IB and SR-IOV virtualization
 - Support for OpenStack, Docker, and Singularity
- OSU Micro-Benchmarks (OMB)
 - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
 - InfiniBand Network Analysis and Monitoring Tool

Startup Performance on TACC Frontera



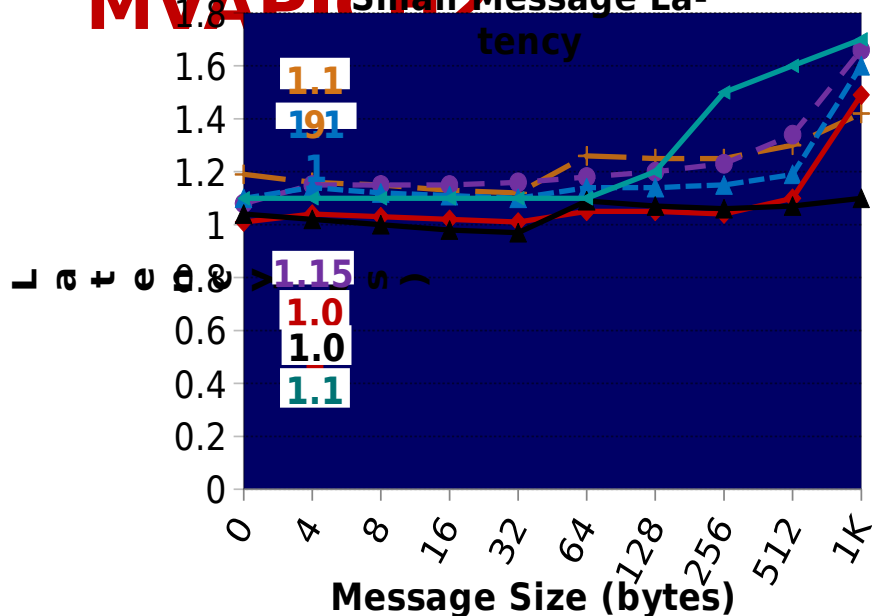
- MPI_Init takes 3.9 seconds on 57,344 processes on 1,024 nodes
- MPI_Init takes 195 seconds on 229376 processes on 4096 nodes while MVAPICH2 takes 31 seconds
- All numbers reported with 56 processes per node

New designs available in MVAPICH2-2.3.2

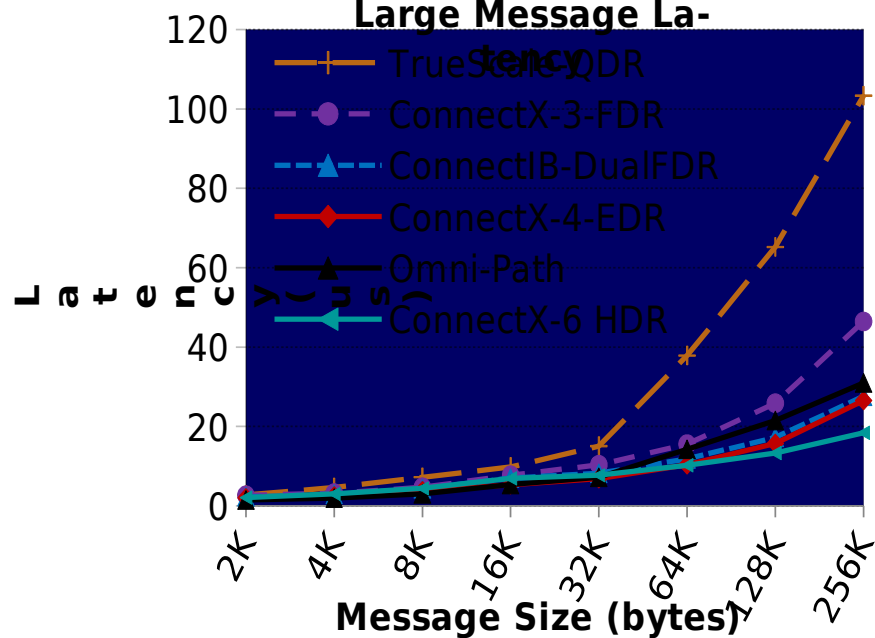
One-way Latency: MPI over IB with

MVAPICH2

Small Message Latency

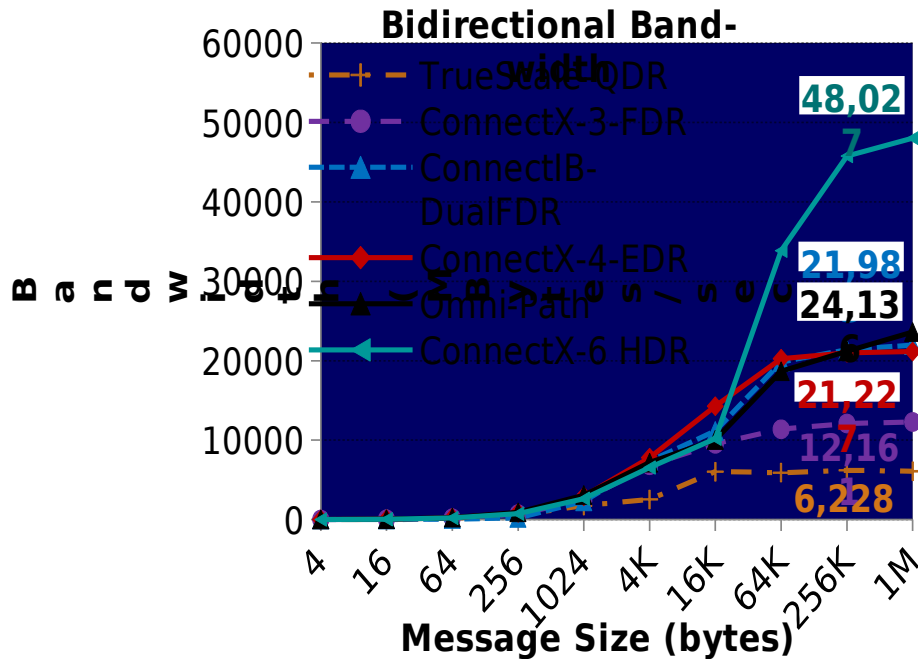
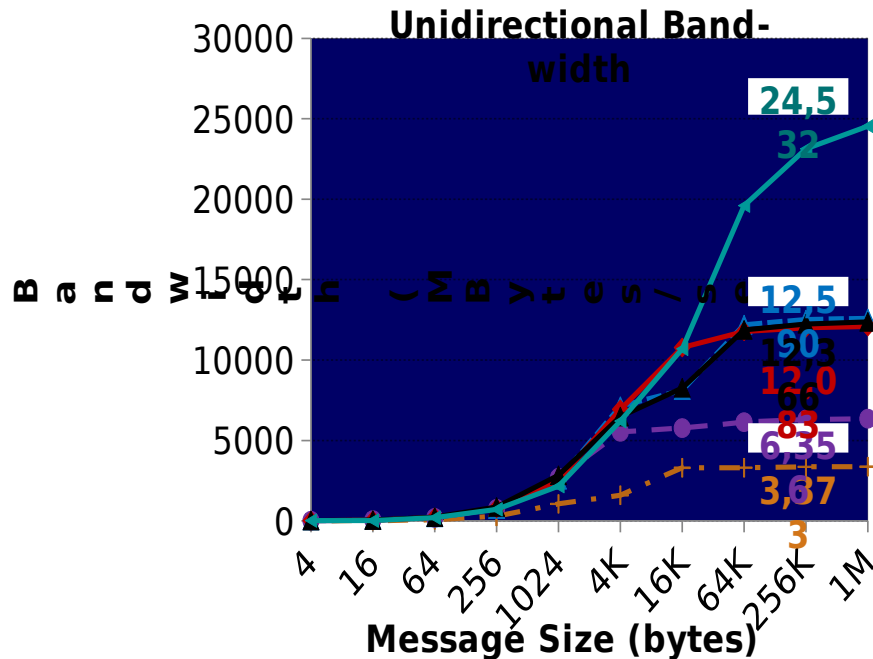


Large Message Latency



- TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
- ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
- ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
- ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch
- Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch
- ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

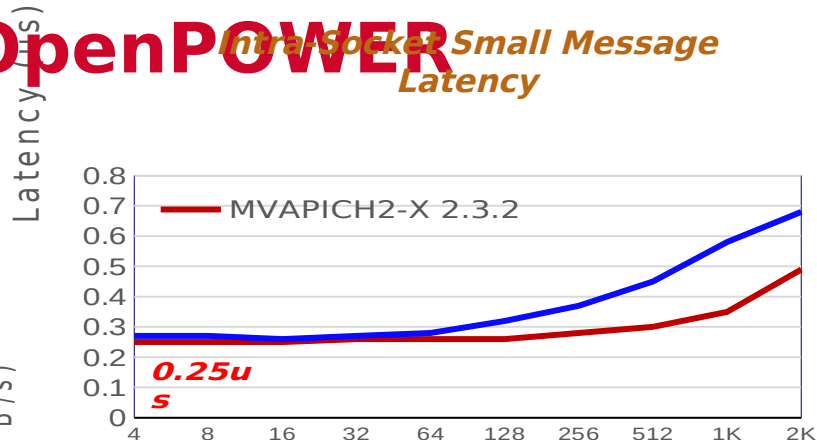
Bandwidth: MPI over IB with MVAPICH2



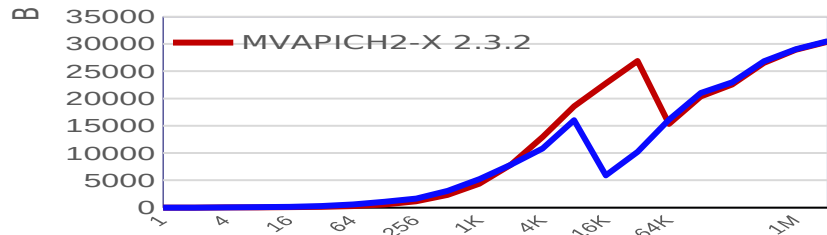
- TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
- ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
- ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
- ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch
- Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch
- ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Intra-node Point-to-Point Performance on OpenPOWER

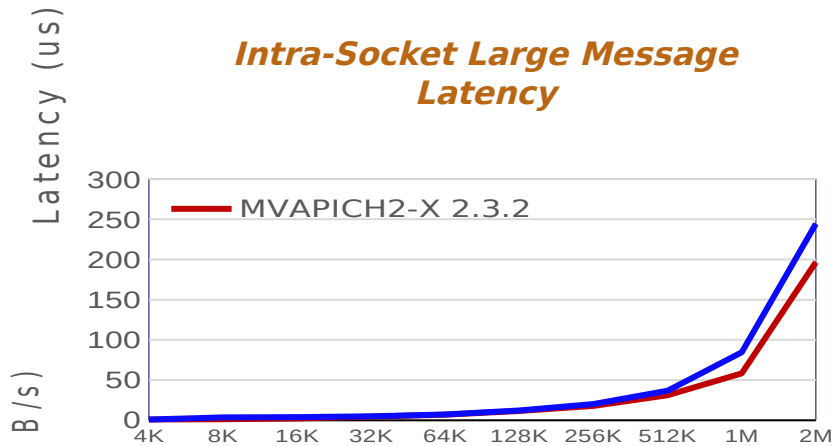
Intra-Socket Small Message Latency



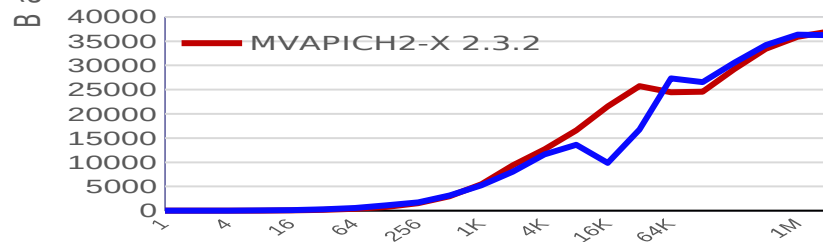
Intra-Socket Bandwidth



Intra-Socket Large Message Latency



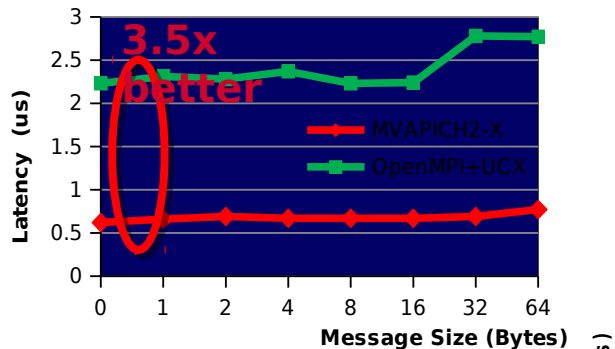
Intra-Socket Bi-directional Bandwidth



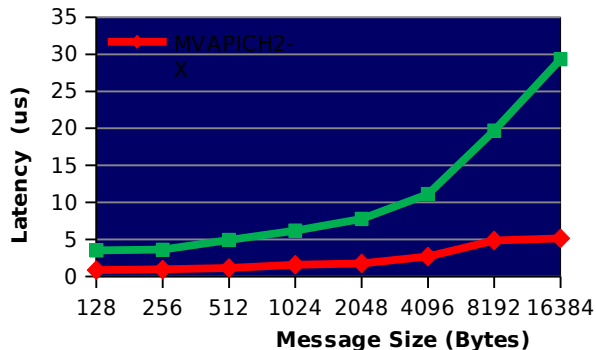
Platform: Two nodes of OpenPOWER (Power9-ppc64le) CPU using Mellanox EDR (MT4121) HCA

Point-to-point: Latency & Bandwidth (Inter-socket) on ARM

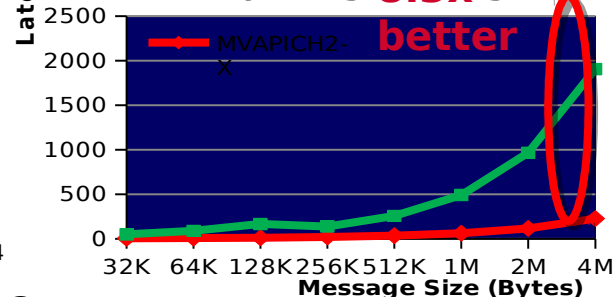
Latency - Small Messages



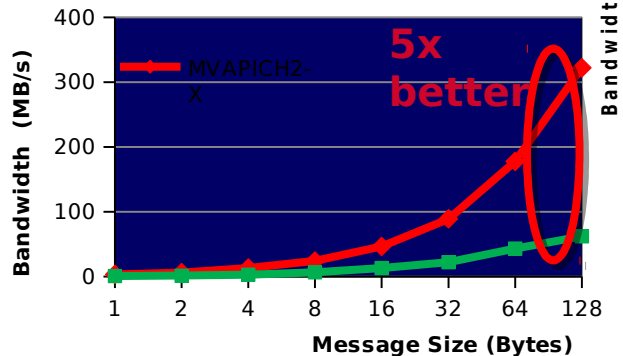
Latency - Medium Messages



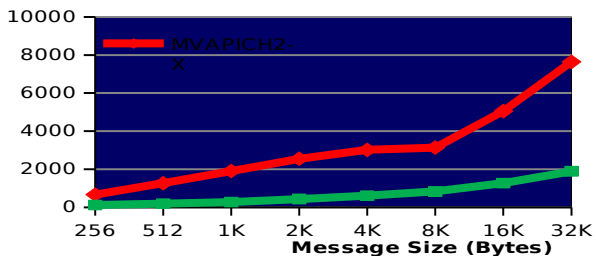
Latency - Large Messages



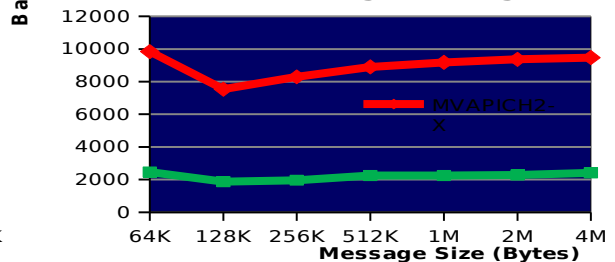
Bandwidth - Small Messages



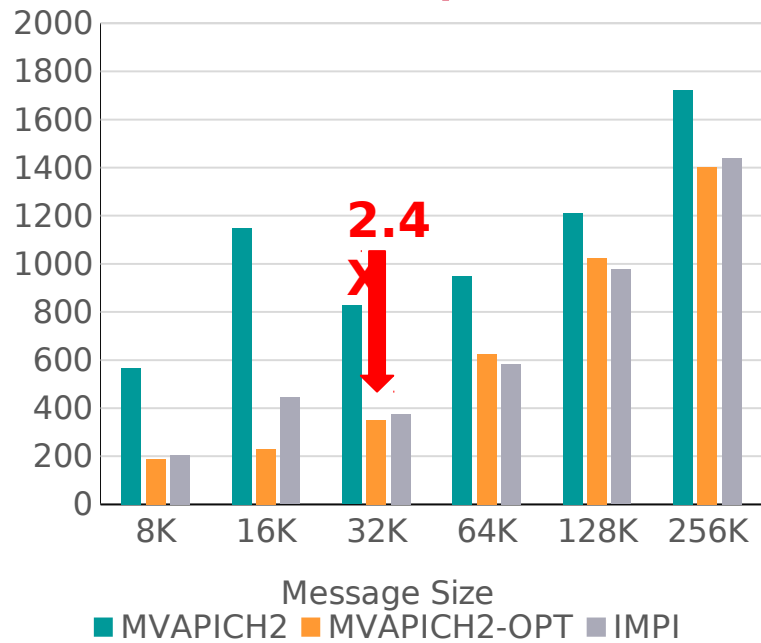
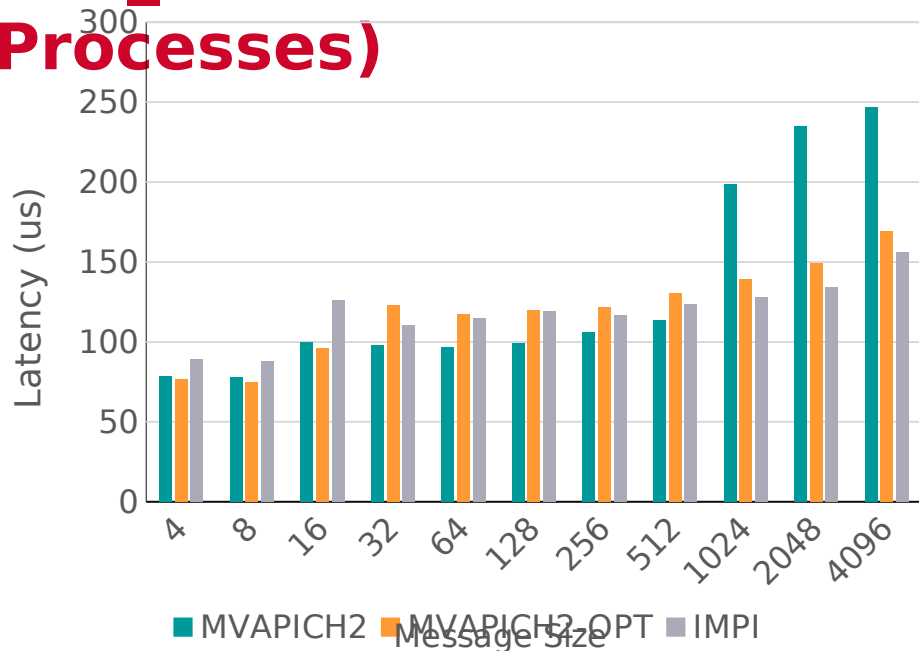
Bandwidth - Medium Messages



Bandwidth - Large Messages



MPI_Allreduce on KNL + Omni-Path (10,240 Processes)



OSU Micro Benchmark 64 PPN

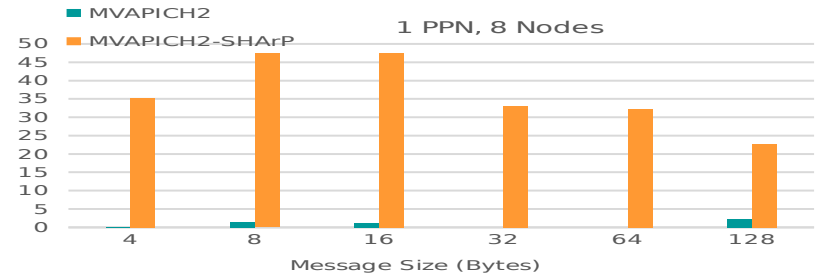
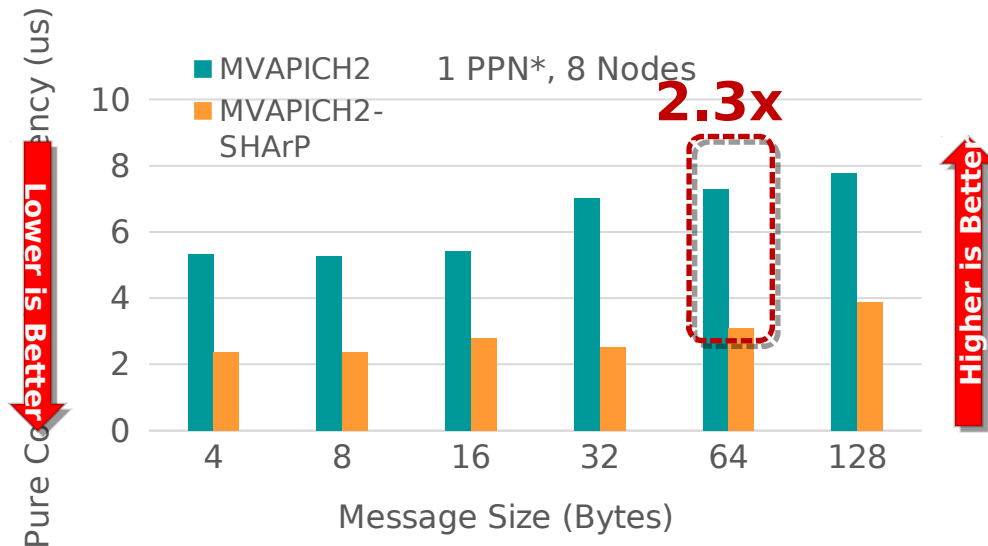
- For MPI_Allreduce latency with 32K bytes, MVAPICH2-OPT can reduce the latency by 2.4x

M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, and D. K. Panda, Scalable Reduction Collectives with Data Partitioning-based Multi-Leader Design, Super Computing '17

Available since MVAPICH2-X 2.3

Evaluation of SHArP based Non Blocking Allreduce

MPI_allreduce Benchmark

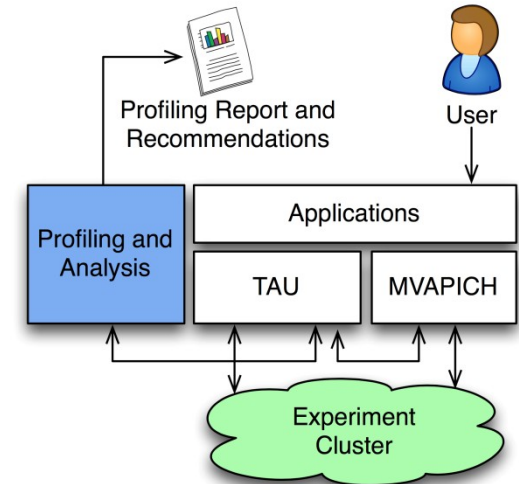


- Complete offload of Allreduce collective operation to Switch helps to have much higher overlap of communication and computation

Available since MVAPICH2 2.3a

Performance Engineering Applications using MVAPICH2 and TAU

- Enhance existing support for MPI_T in MVAPICH2 to expose a richer set of performance and control variables
- Get and display MPI Performance Variables (PVARs) made available by the runtime in TAU
- Control the runtime's behavior via MPI Control Variables (CVARs)
- Introduced support for new MPI_T based CVARs to MVAPICH2
 - MPIR_CVAR_MAX_INLINE_MSG_SZ,
 - MPIR_CVAR_VBUF_POOL_SIZE,
 - MPIR_CVAR_VBUF_SECONDARY_POOL_SIZE
- TAU enhanced with support for setting MPI_T CVARs in a



VBUF usage without CVAR based tuning as displayed by ParaProf vs **VBUF usage with CVAR based tuning as displayed by ParaProf**

Name	Min Value	Max Value	Mean Value	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory used for VBUFs)	3,313,056	3,313,056	3,313,056	0	1	3,313,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	320	320	320	0	1	320
mv2_vbuf_available (Number of VBUFs available)	255	255	255	0	1	255
mv2_vbuf_freed (Number of VBUFs freed)	25,545	25,545	25,545	0	1	25,545
mv2_vbuf_inuse (Number of VBUFs inuse)	65	65	65	0	1	65
mv2_vbuf_max_use (Maximum number of VBUFs used)	65	65	65	0	1	65
num_calloc_calls (Number of MPI_T_calloc calls)	89	89	89	0	1	89

Name	Min Value	Max Value	Mean Value	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	1,815,056	1,815,056	1,815,056	0	1	1,815,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	160	160	160	0	1	160
mv2_vbuf_available (Number of VBUFs available)	94	94	94	0	1	94
mv2_vbuf_freed (Number of VBUFs freed)	5,479	5,479	5,479	0	1	5,479
mv2_vbuf_inuse (Number of VBUFs inuse)	66	66	66	0	1	66

Dynamic and Adaptive Tag Matching

Tag matching is a significant overhead for receivers

Existing Solutions are

- Static and do not adapt dynamically to communication pattern
- Do not consider memory overhead

A new tag matching design

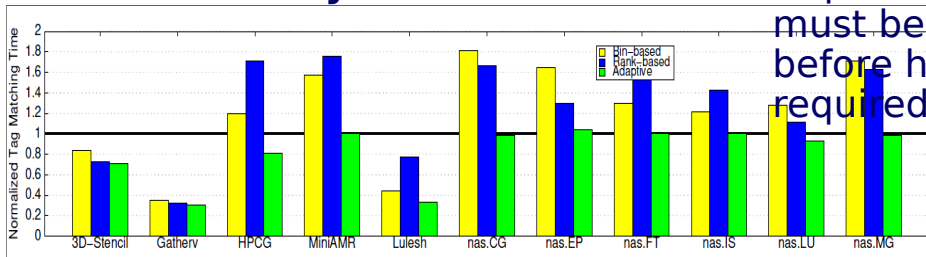
- Dynamically adapt to communication patterns

- Use different strategies for different ranks

- Decisions are based on the number of request object that

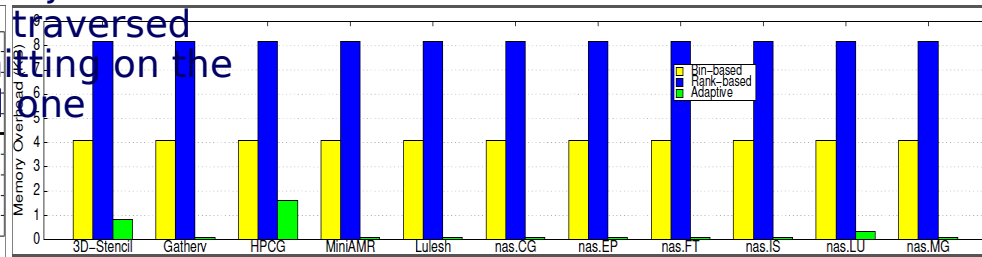
Better performance than other state-of-the-art tag-matching schemes
Minimum memory consumption

Will be available in future MVAPICH2 releases



Normalized Total Tag Matching Time at 512 Processes

Normalized to Default (Lower is Better)



Normalized Memory Overhead per Process at 512 Processes

Compared to Default (Lower is Better)

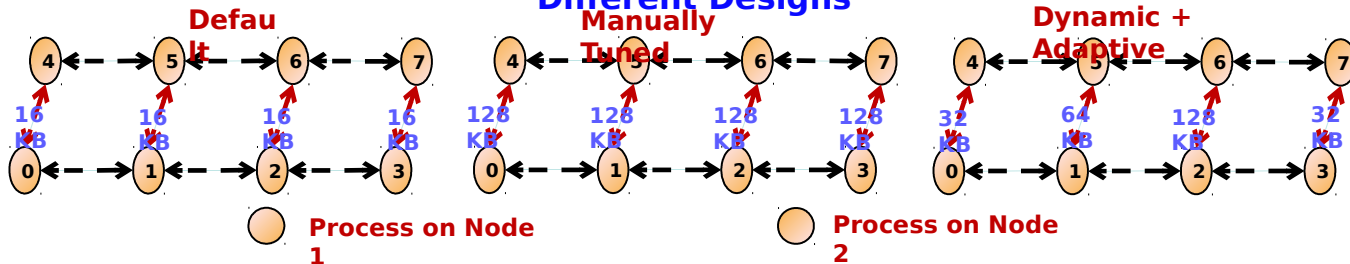
Adaptive and Dynamic Design for MPI Tag Matching; M. Bayatpour, H. Subramoni, S. Chakraborty, and D. K. Panda;

Dynamic and Adaptive MPI Point-to-point Communication Protocols

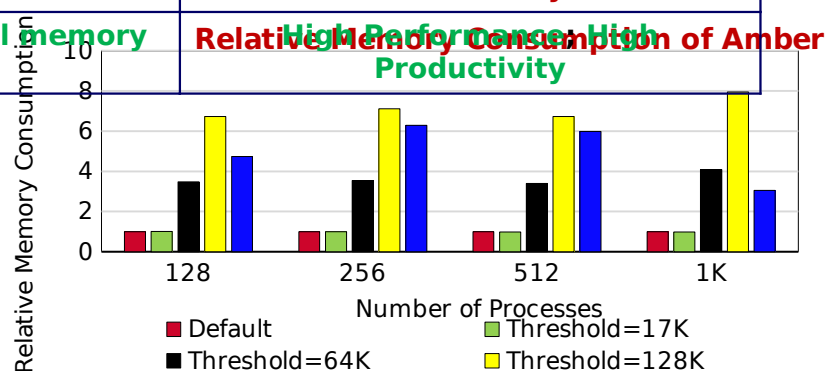
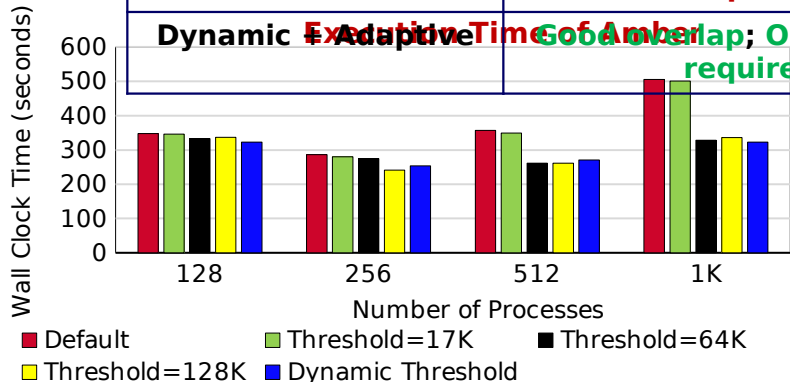
Desired Eager Threshold

Process Pair	Eager Threshold (KB)
0 - 4	32
1 - 5	64
2 - 6	128
3 - 7	32

Eager Threshold for Example Communication Pattern with Different Designs



Default	Poor overlap; Low memory requirement	Low Performance; High Productivity
Manually Tuned	Good overlap; High memory requirement	High Performance; Low Productivity
Dynamic + Adaptive	Good overlap; Optimal memory requirement	High Performance; High Productivity



MVAPICH2 Distributions

- MVAPICH2
 - Basic MPI support for IB, Omni-Path, iWARP and RoCE
- MVAPICH2-X
 - Advanced MPI features and support for INAM
 - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
 - Optimized for HPC Clouds with IB and SR-IOV virtualization
 - Support for OpenStack, Docker, and Singularity
- OSU Micro-Benchmarks (OMB)
 - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
 - InfiniBand Network Analysis and Monitoring Tool

MVAPICH2-X for Hybrid MPI + PGAS

High Performance Parallel Programming Models

MPI

Message Passing Interface

PGAS

(UPC, OpenSHMEM, CAF, UPC++)

Hybrid --- MPI + X

(MPI + PGAS + OpenMP/Cilk)

High Performance and Scalable Unified Communication Runtime

Diverse APIs and Mechanisms

Optimized
Point-to-point
Primitives

Remote
Memory
Access

Active
Messages

Collectives
Algorithms
(Blocking and
Non-Blocking)

Scalable Job
Startup

Fault
Tolerance

Introspection
& Analysis
with OSU
INAM

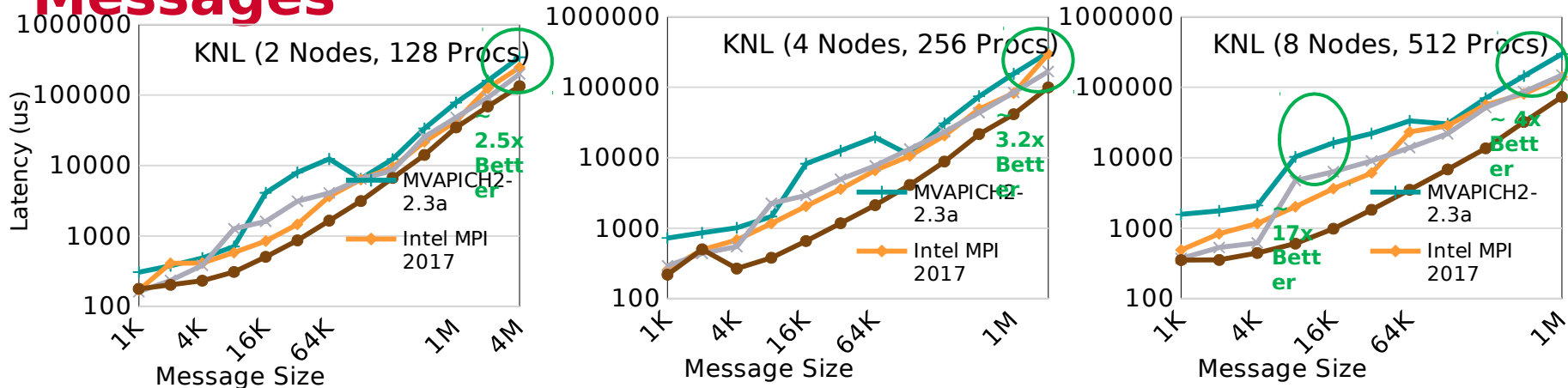
Support for Efficient Intra-node Communication
(POSIX SHMEM, CMA, LiMIC, XPMEM...)

Support for Modern Networking Technologies
(InfiniBand, iWARP, RoCE, Omni-Path...)

Support for Modern Multi-/Many-core Architectures
(Intel-Xeon, GPGPUs, OpenPOWER, ARM...)

- Current Model - Separate Runtimes for OpenSHMEM/UPC/UPC++/CAF and MPI
 - Possible deadlock if both runtimes are not progressed
 - Consumes more network resource
- Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF

Optimized CMA-based Collectives for Large Messages



Performance of MPI_Gather on KNL nodes (64PPN)

- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPower)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

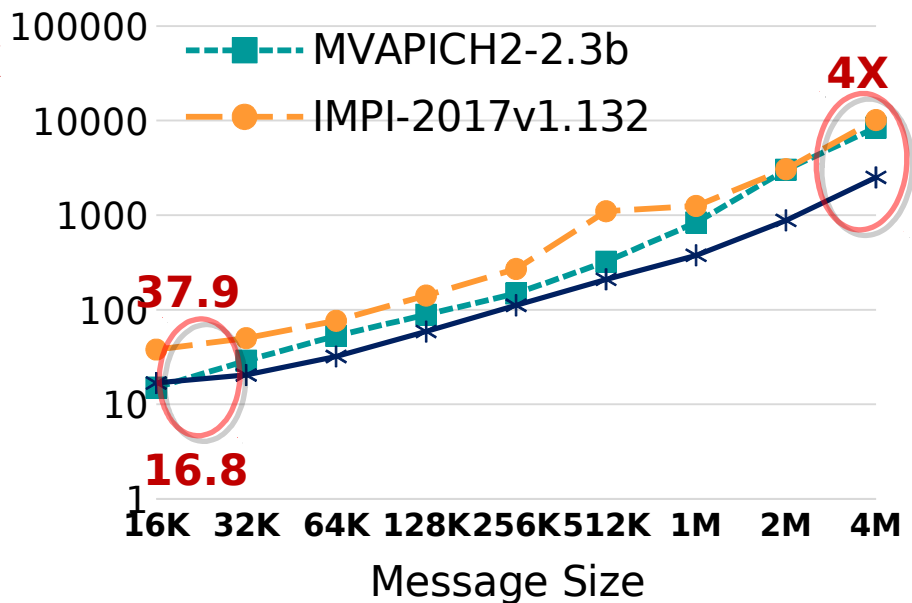
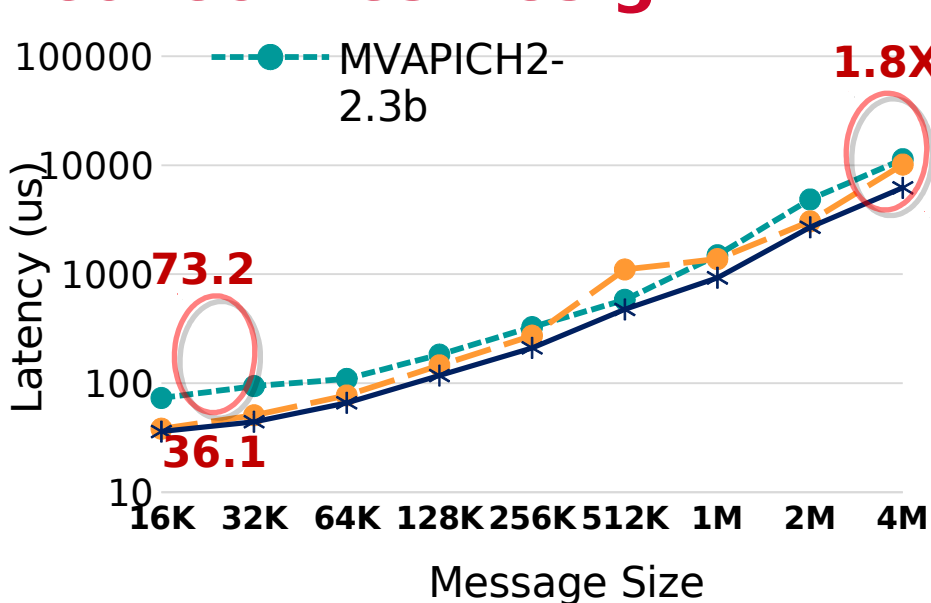
S. Chakraborty, N. Subramoni, and D. K. Panda, Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems, *IEEE Cluster '17, BEST Paper Award*

Available since MVAPICH2-X 2

Shared Address Space (XPMEM)-based

Collectives Design

OSU_Allreduce (Broadwell 256 procs) OSU_Reduce (Broadwell 256 procs)

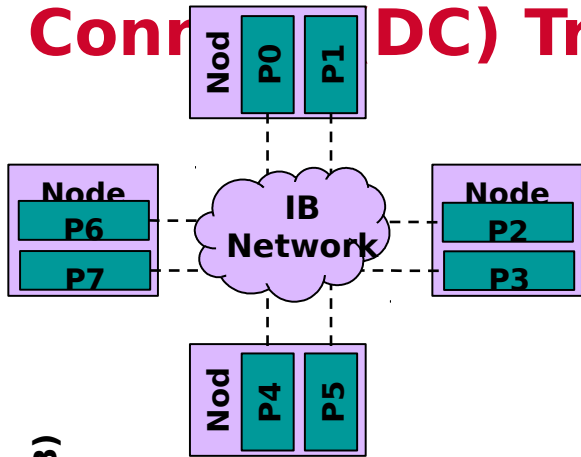


- “Shared Address Space”-based true zero-copy Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation

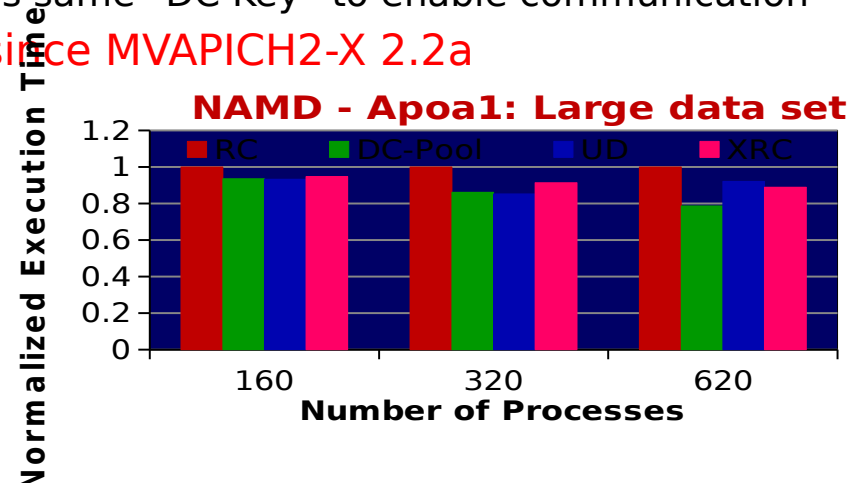
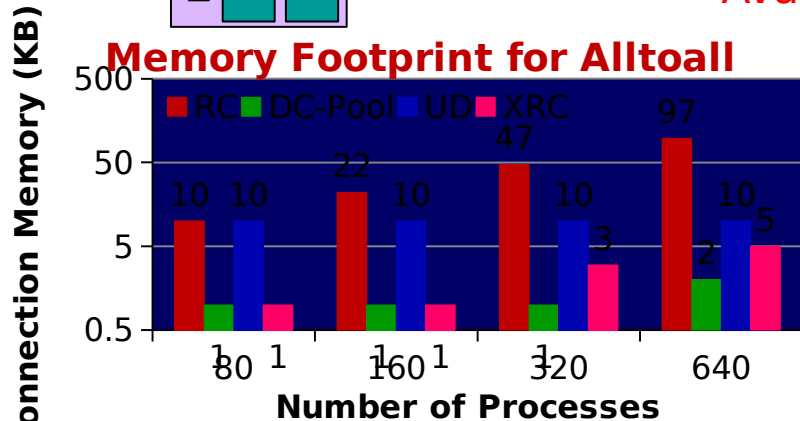
Available in MVAPICH2-X 2.3

Up to 3.3x improvement for 4MB Reduce and up to 1.8x improvement for 4M

Minimizing Memory Footprint by Direct Connected (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

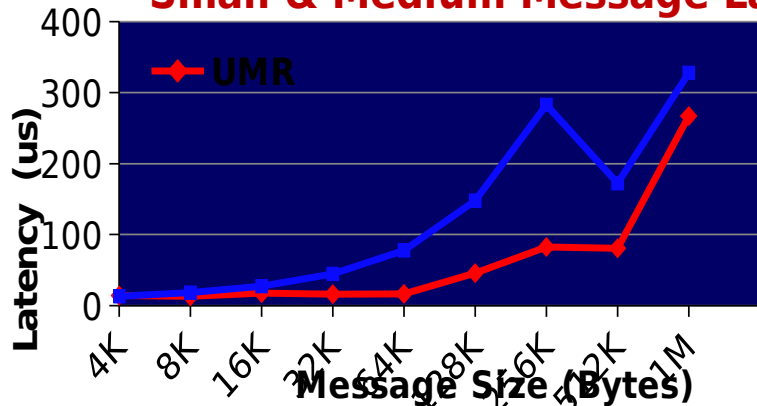


H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty and D. K. Panda, Designing MPI Library with Dynamic Connected Transport (DCT) of InfiniBand : Early Experiences. IEEE International Supercomputing Conference (ISC

User-mode Memory Registration (UMR)

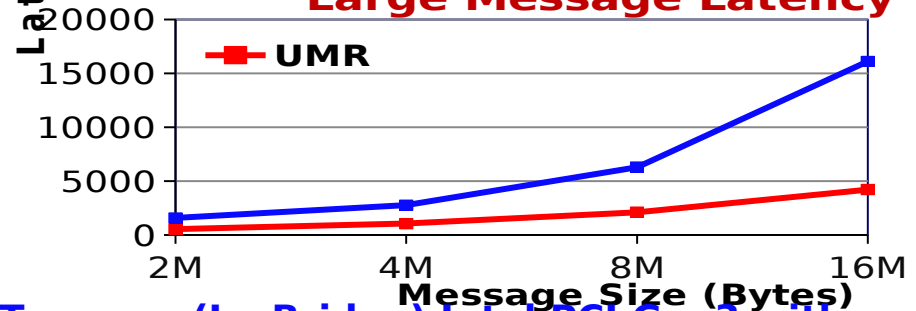
- Introduced by Mellanox to support direct local and remote noncontiguous memory access
- Avoid packing at sender and unpacking at receiver
- Available since MVAPICH2-X 2.2b

Small & Medium Message Latency



Latency (us)

Large Message Latency



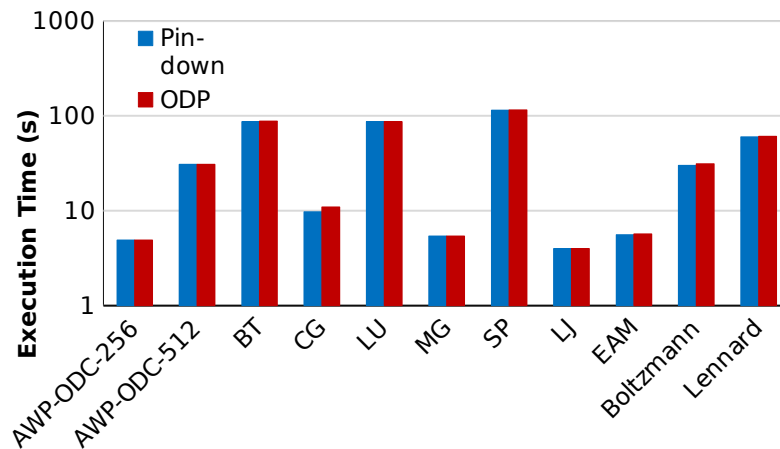
Connect-IB (54 Gbps): 2.8 GHz Dual Ten-core (IvyBridge) Intel PCI Gen3 with Mellanox IB FDR switch

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)

- Applications no longer need to pin down underlying physical pages
- Memory Region (MR) are **NEVER** pinned by the OS
 - Paged in by the HCA when needed
 - Paged out by the OS when reclaimed
- ODP can be divided into two classes
 - **Explicit ODP**
 - Applications still register memory buffers for communication, but this operation is used to define access control for IO rather than pin-down the pages
 - **Implicit ODP**
 - Applications are provided with a special memory key that represents their complete address space, does not need to register any virtual address range
- Advantages
 - Simplifies programming
 - Unlimited MR sizes

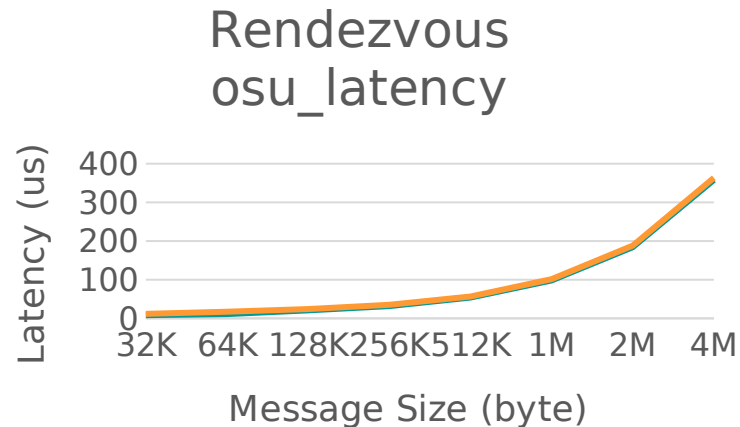
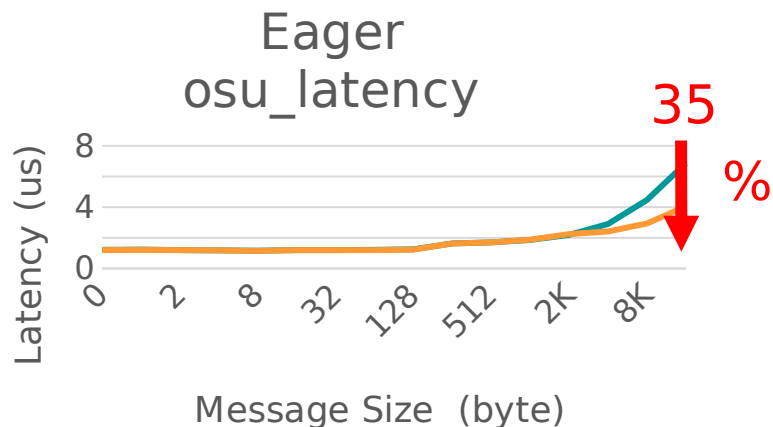
Applications (64 Processes)



M. Li, K. Hamidouche, X. Lu, H. Subramoni, J. Zhang, and D. K. Panda, "Designing MPI Library with On-Demand Paging (ODP) of InfiniBand: Challenges and Benefits", SC 2016.

Available since MVAPICH2-X 2.3

Impact of Zero Copy MPI Message Passing using HW Tag Matching (Point-to-point)

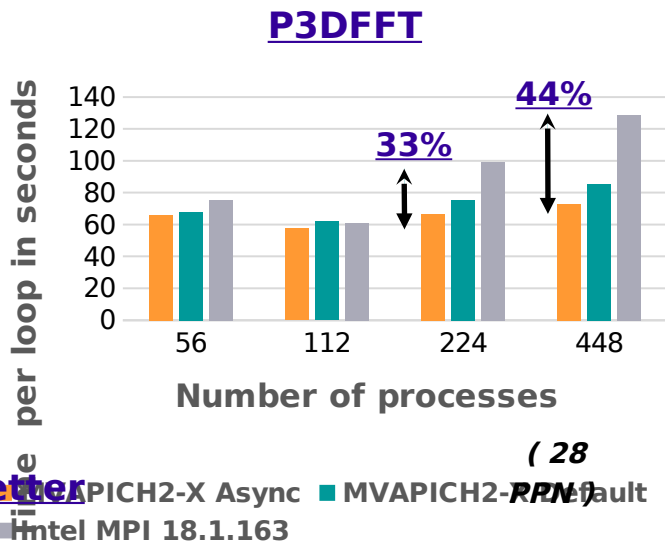


— MVAPICH2
— MVAPICH2+HW-TM

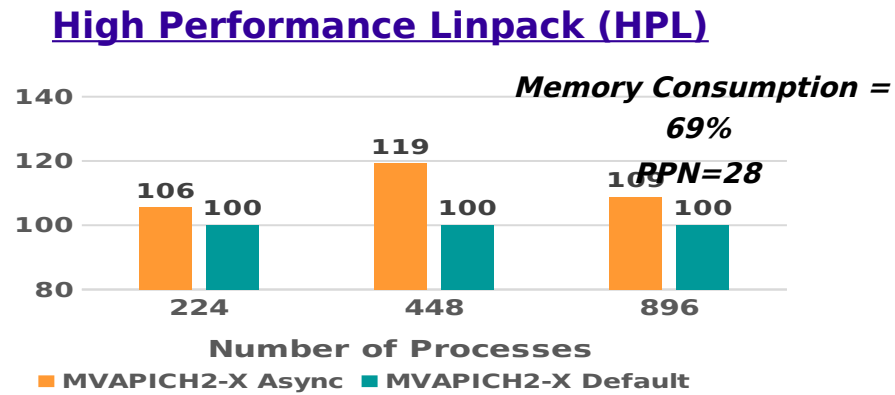
— MVAPICH2
— MVAPICH2+HW-TM

Removal of intermediate buffering/copies can lead up to 35% performance improvement in latency of medium messages on TA0

Benefits of the New Asynchronous Progress Design: Broadwell + InfiniBands



Normalized Performance in GFLOPS



Lower is better

Higher is better

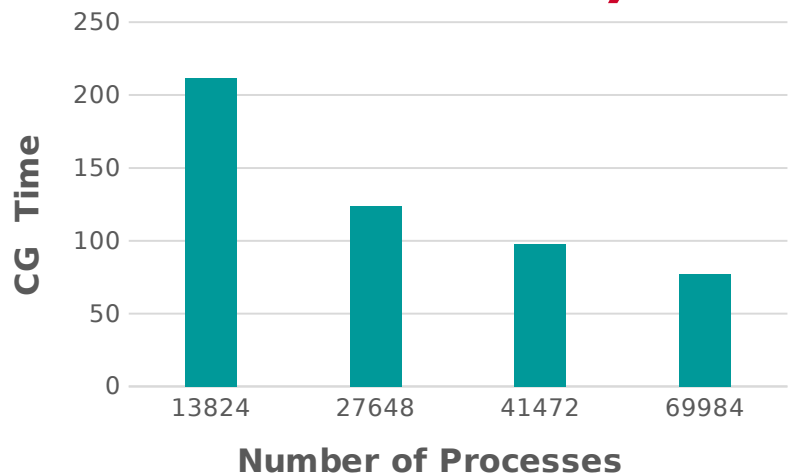
Up to **44%** performance improvement in P3DFFT application with **448 processes**

Up to **19% and 9%** performance improvement in HPL application with **448 and 896 processes**

A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and D.K. Panda, Efficient Asynchronous Progress for MPI without Dedicated Resources, EuroMPI 2018

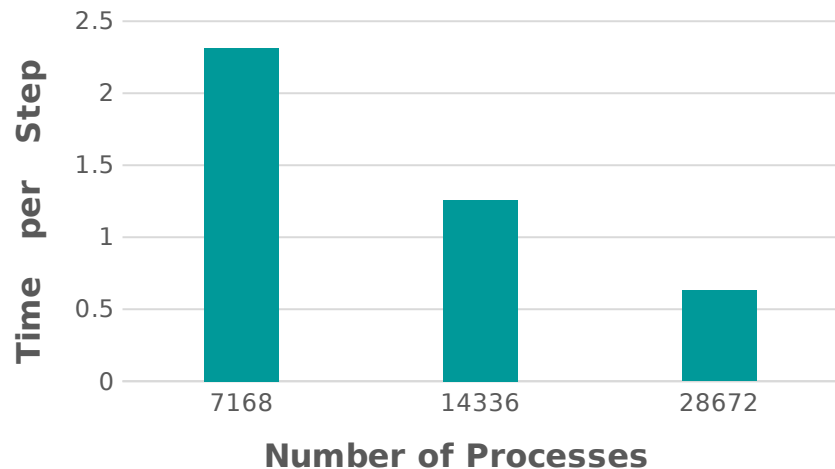
Available in MVAPICH2-X 2.3rc

Evaluation of Applications on Frontera (Cascade Lake + HDR100)



PPN=54

MIMD Lattice Computation (MILC)



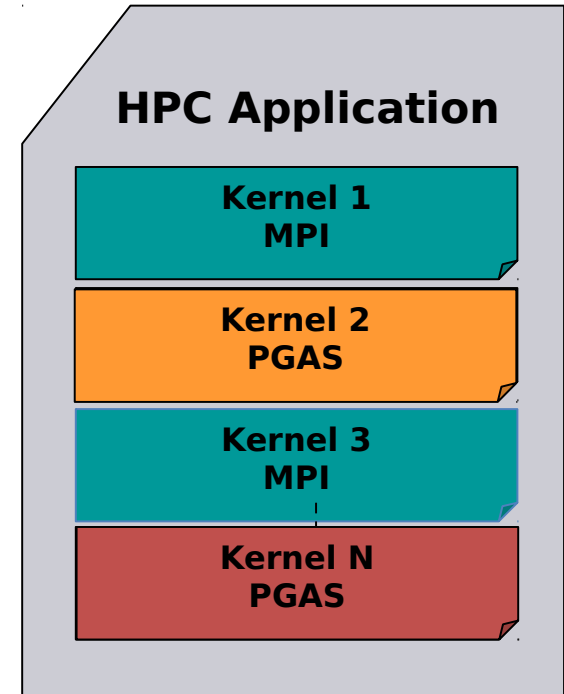
PPN=56

WRF2

Performance of MILC and WRF2 applications scales well with increase in system size

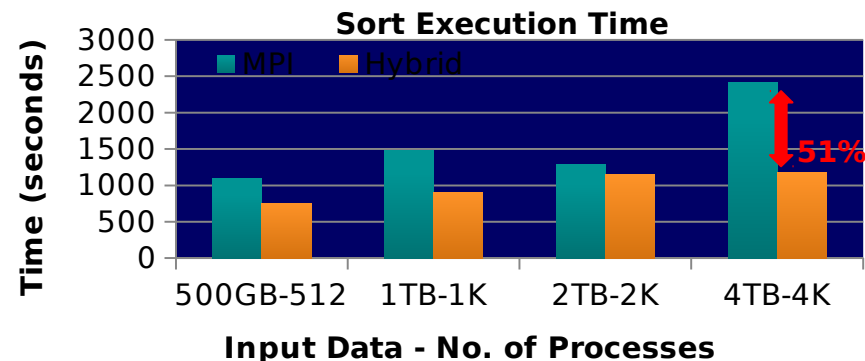
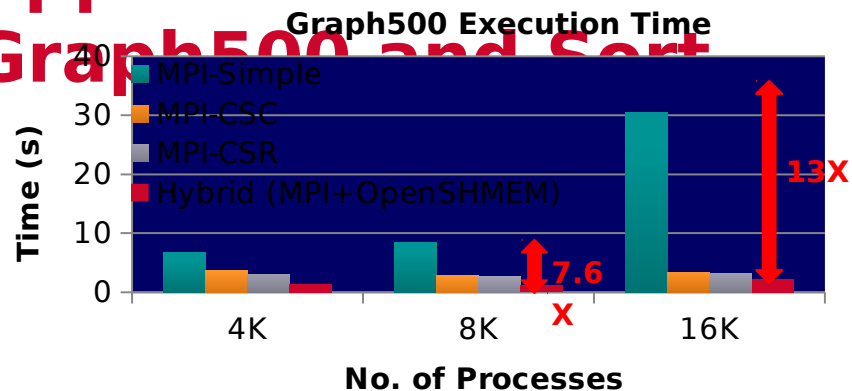
Hybrid (MPI+PGAS) Programming

- 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



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

MVAPICH2 Distributions

- MVAPICH2
 - Basic MPI support for IB, Omni-Path, iWARP and RoCE
- MVAPICH2-X
 - Advanced MPI features and support for INAM
 - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
 - Optimized for HPC Clouds with IB and SR-IOV virtualization
 - Support for OpenStack, Docker, and Singularity
- OSU Micro-Benchmarks (OMB)
 - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
 - InfiniBand Network Analysis and Monitoring Tool

Can HPC and Virtualization be Combined?

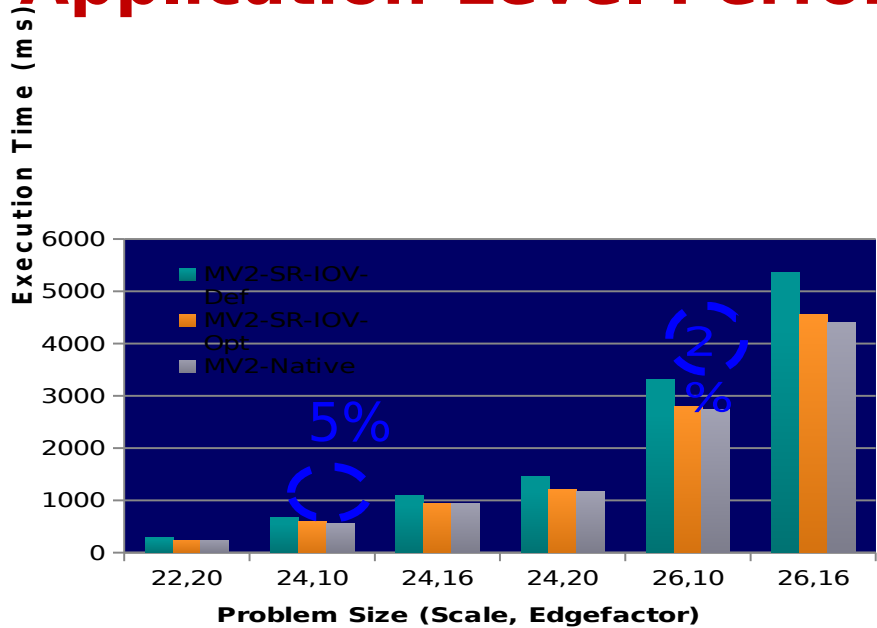
- Virtualization has many benefits
 - Fault-tolerance
 - Job migration
 - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root - IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2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

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

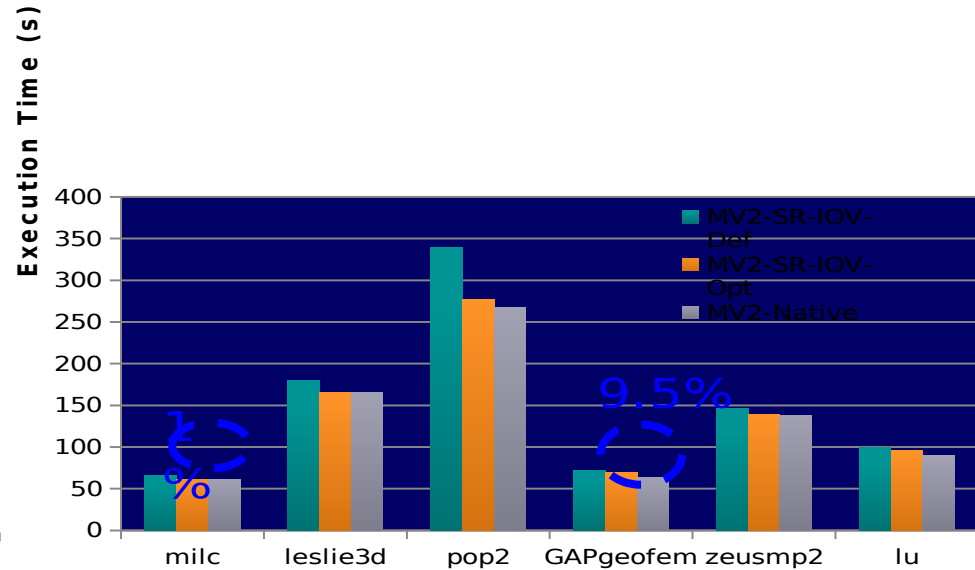
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Library over SR-IOV enabled InfiniBand Clusters, HiPC'14

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

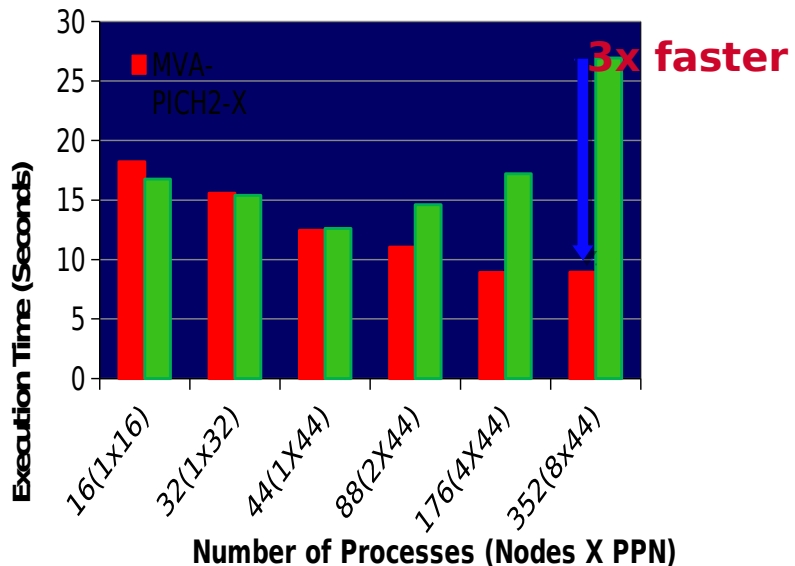


SPEC
MPI2007

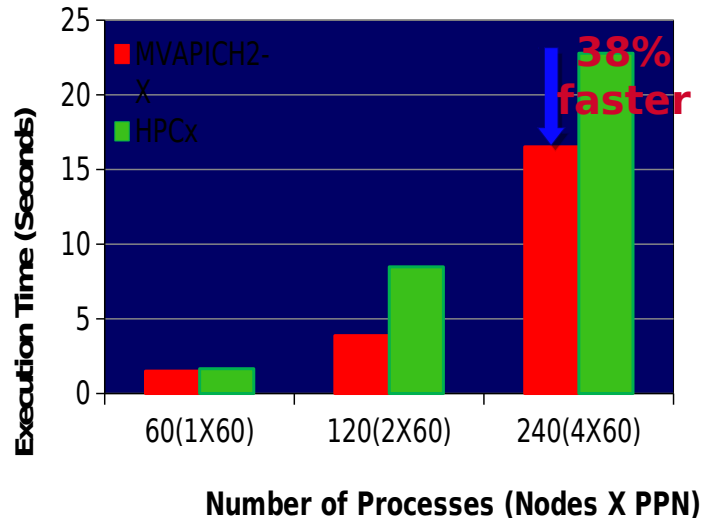
- 32 VMs, 6 Core/VM
- Compared to Native, 2-5% overhead for Graph500 with 128 Procs
- Compared to Native, 1-9.5% overhead for SPEC MPI2007 with 128 Procs

Performance of Radix on Microsoft Azure

Total Execution Time on HC (Lower is better)



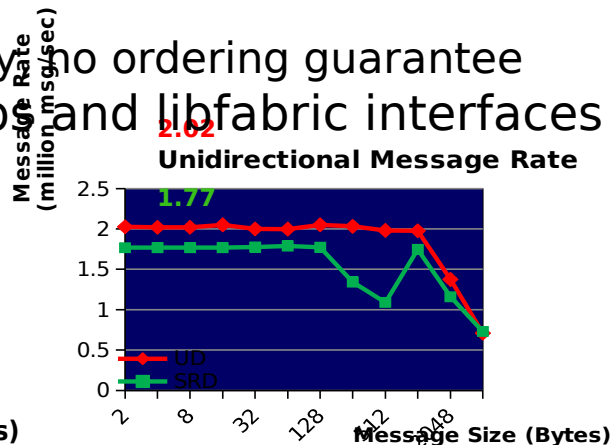
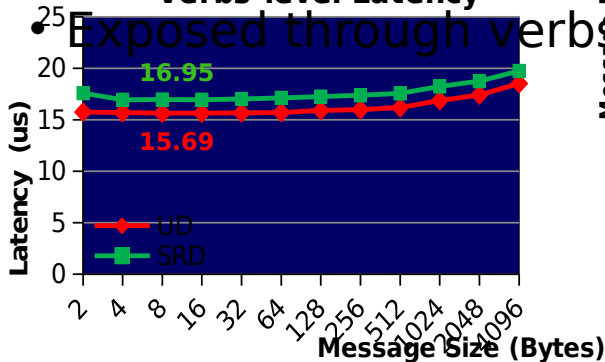
Total Execution Time on HB (Lower is better)



Amazon Elastic Fabric Adapter (EFA)

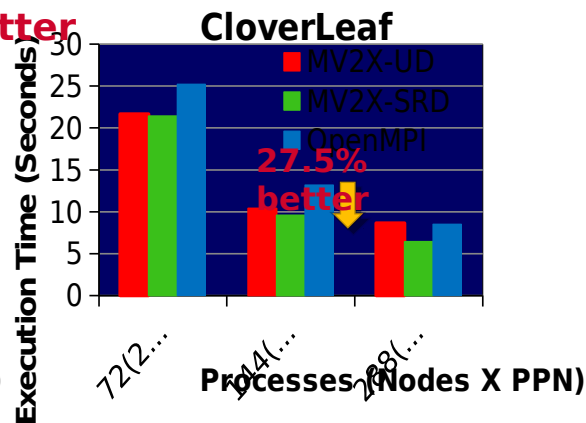
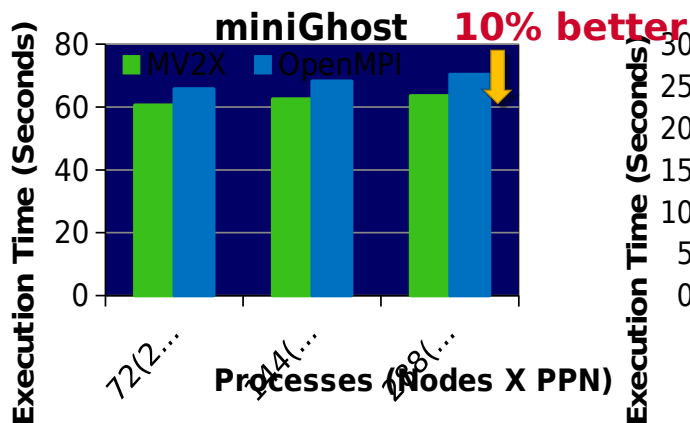
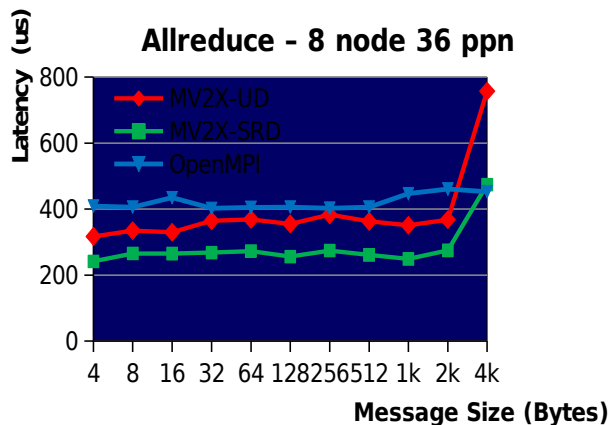
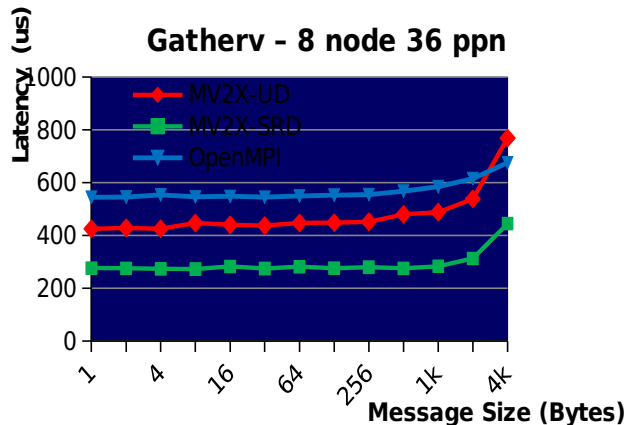
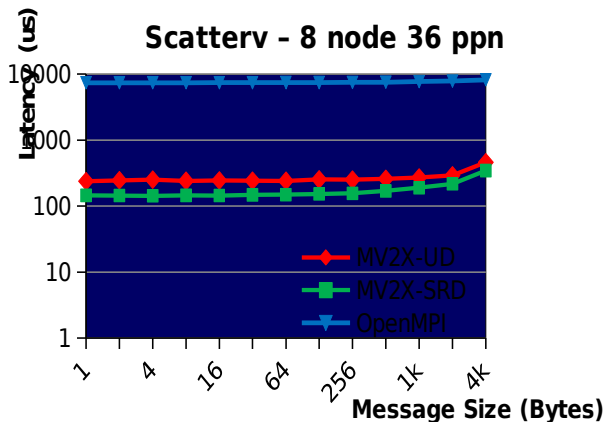
- Enhanced version of Elastic Network Adapter (ENA)
 - Allows OS bypass, up to 100 Gbps bandwidth
- New QP type: **Scalable Reliable Datagram (SRD)**

- Network aware multi-path routing - low tail latency
- Guaranteed Delivery, no ordering guarantee



Feature	UD	SRD
Send/Recv	✓	✓
Send w/ Immediate	✗	✗
RDMA Read/Write/Atomic	✗	✗
Scatter Gather Lists	✓	✓
Shared Receive Queue	✗	✗
Reliable Delivery	✗	✓
Ordering	✗	✗
Inline Sends	✗	✗
Global Routing Header	✓	✗
MTU Size	4KB	8KB

MPI-level Performance with SRD



Instance type: c5n.18xlarge
 CPU: Intel Xeon Platinum 8124M @ 3.00GHz
 MVAPICH2 version: MVAPICH2-X 2.3rc2 + SRD support
 OpenMPI version: Open MPI v3.1.3 with libfabric 1.7

S. Chakraborty, S. Xu, H. Subramoni, and D. K. Panda, *Designing Scalable and High-performance MPI Libraries on Amazon Elastic Fabric Adapter, to be presented at the 26th Symposium on High Performance Interconnects, (HOTI '19)*

MVAPICH2 Distributions

- MVAPICH2
 - Basic MPI support for IB, Omni-Path, iWARP and RoCE
- MVAPICH2-X
 - Advanced MPI features and support for INAM
 - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
 - Optimized for HPC Clouds with IB and SR-IOV virtualization
 - Support for OpenStack, Docker, and Singularity
- OSU Micro-Benchmarks (OMB)
 - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
 - InfiniBand Network Analysis and Monitoring Tool

OSU Microbenchmarks

- Available since 2004
- Suite of microbenchmarks to study communication performance of various programming models
- Benchmarks available for the following programming models
 - Message Passing Interface (MPI)
 - Partitioned Global Address Space (PGAS)
 - Unified Parallel C (UPC)
 - Unified Parallel C++ (UPC++)
 - OpenSHMEM
- Benchmarks available for multiple accelerator based architectures
 - Compute Unified Device Architecture (CUDA)
 - OpenACC Application Program Interface
- Part of various national resource procurement suites like NERSC-8 / Trinity Benchmarks
- Please visit the following link for more information

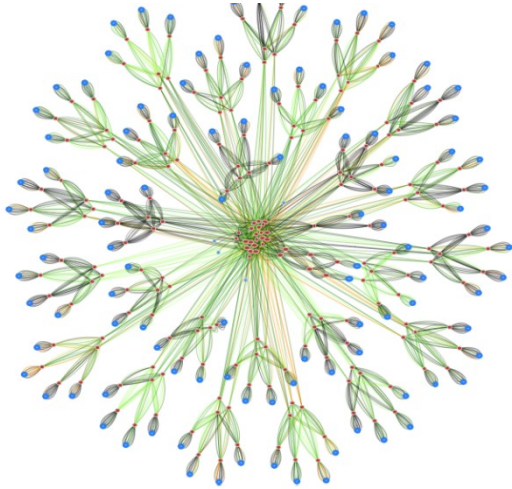
MVAPICH2 Distributions

- MVAPICH2
 - Basic MPI support for IB, Omni-Path, iWARP and RoCE
- MVAPICH2-X
 - Advanced MPI features and support for INAM
 - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
 - Optimized for HPC Clouds with IB and SR-IOV virtualization
 - Support for OpenStack, Docker, and Singularity
- OSU Micro-Benchmarks (OMB)
 - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
 - InfiniBand Network Analysis and Monitoring Tool

Overview of OSU INAM

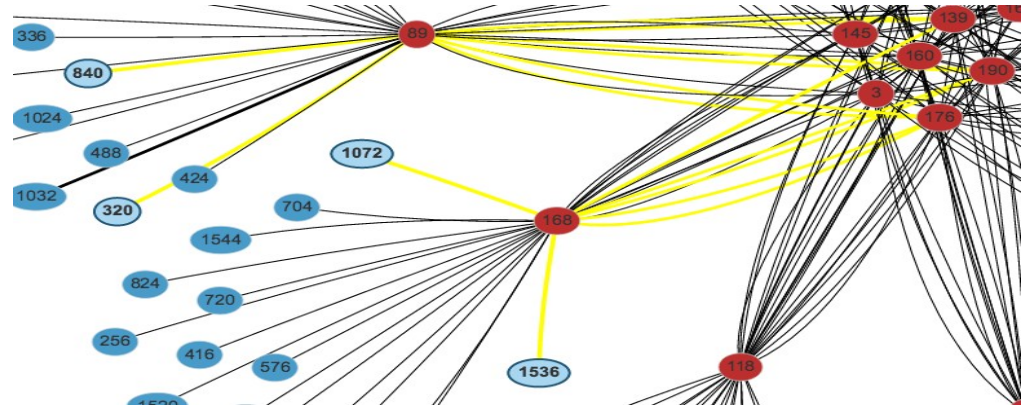
- A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime
 - <http://mvapich.cse.ohio-state.edu/tools/osu-inam/>
- Monitors IB clusters in real time by querying various subnet management entities and gathering input from the 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 0.9.4 released on 11/10/2018**
 - Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs
 - Ability to gather InfiniBand performance counters at sub-second granularity for very large (>2000 nodes) clusters
 - Redesign database layout to reduce database size
 - Enhanced fault tolerance for database operations
 - Thanks to Trey Dockendorf @ OSC for the feedback
 - OpenMP-based multi-threaded designs to handle database purge, read, and insert operations simultaneously

OSU INAM Features



Comet@SDSC --- Clustered View

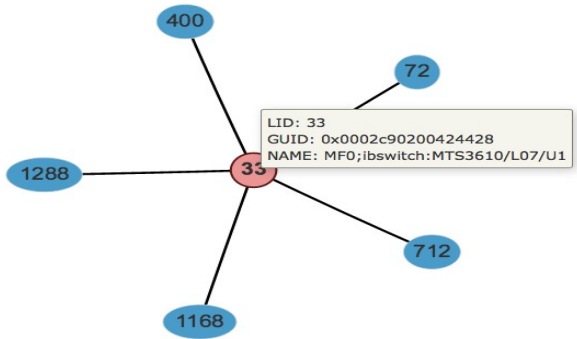
(1,879 nodes, 212 switches, 4,377



Finding Routes Between Nodes

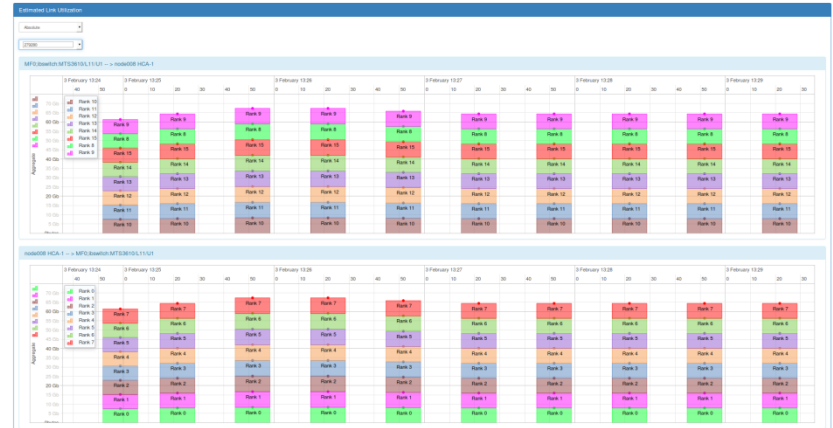
- Show network topology of large clusters
- Visualize traffic pattern on different links
- Quickly identify congested links/links in error state
- See the history unfold - play back historical state of the network

OSU INAM Features (Cont.)



Visualizing a Job (5 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 - details per process or per node
 - CPU utilization for each rank/node



Estimated Process Level Link

- Estimated Utilization view
 - Classify data flowing over a network link at different granularity in conjunction with MVAPICH2-X 2.2rc1
 - Job level and
 - Process level

Applications-Level Tuning: Compilation of Best Practices

MPI runtime has many parameters

- Tuning a set of parameters can help you to extract higher performance
- Compiled a list of such contributions through the MVAPICH Website
 - http://mvapich.cse.ohio-state.edu/best_practices/
- Initial list of applications
 - Amber
 - HoomDBLue
 - HPCG
 - Lulesh
 - MILC
 - Neuron
 - SMG2000
 - Cloverleaf
 - SPEC (LAMMPS, POP2, TERA_TF, WRF2)
- Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu.

Commercial Support for MVAPICH2 Libraries

- Supported through X-ScaleSolutions (<http://x-scalesolutions.com>)
- Benefits:
 - Help and guidance with installation of the library
 - Platform-specific optimizations and tuning
 - Timely support for operational issues encountered with the library
 - Web portal interface to submit issues and tracking their progress
 - Advanced debugging techniques
 - Application-specific optimizations and tuning
 - Obtaining guidelines on best practices
 - Periodic information on major fixes and updates
 - Information on major releases
 - Help with upgrading to the latest release
 - Flexible Service Level Agreements
- Support provided to Lawrence Livermore National Laboratory (LLNL) during last two years

MVAPICH2 - Plans for Exascale

- Performance and Memory scalability toward 1M-10M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF ...)
 - MPI + Task*
- Enhanced Optimization for GPUs and FPGAs*
- Taking advantage of advanced features of Mellanox InfiniBand
 - Tag Matching*
 - Adapter Memory*
- Enhanced communication schemes for upcoming architectures
 - 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 FT support
- Support for * features will be available in future MVAPICH2 Releases

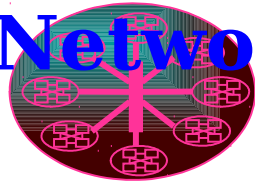
Join us for Multiple Events at SC '19

- Presentations at OSU and X-Scale Booth (#2094)
 - Members of the MVAPICH, HiBD and HiDL members
 - External speakers
- Presentations at SC main program (Tutorials, Workshops, BoFs, Posters, and Doctoral Showcase)
- Presentation at many other booths (Mellanox, Intel, Microsoft, and AWS) and satellite events
- Complete details available at
<http://mvapich.cse.ohio-state.edu/conference/752/talks/>

Thank You!

panda@cse.ohio-state.edu

Network Based Computing



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



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



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



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