MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing

Presentation at Mellanox Theatre (SC '17)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Drivers of Modern HPC Cluster Architectures

Multi-core Processors

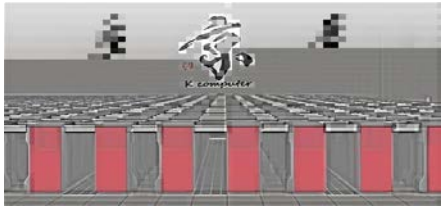
High Performance Interconnects - InfiniBand
<1usec latency, 100Gbps Bandwidth>

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

SSD, NVMe-SSD, NVRAM

- 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.

Sunway TaihuLight

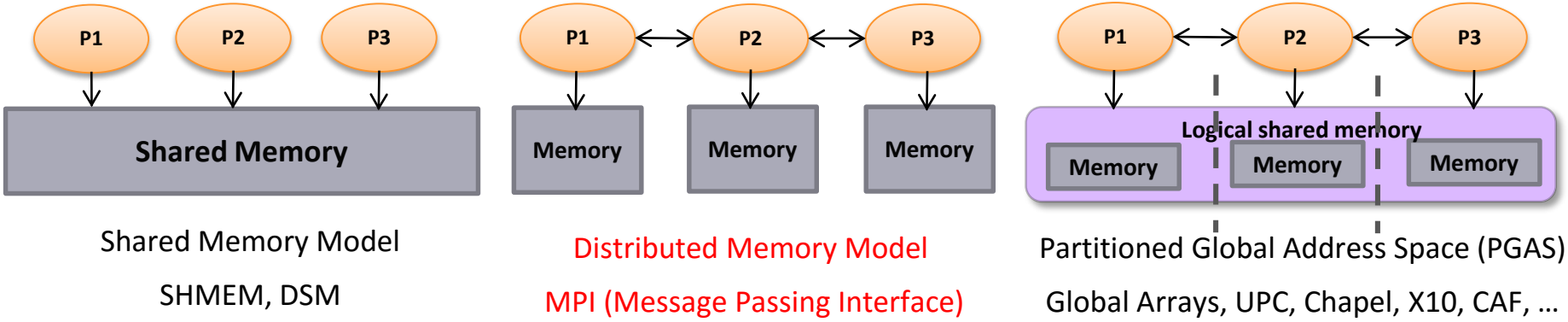


K - Computer

Tianhe – 2

Titan

Parallel Programming Models Overview



Shared Memory Model

SHMEM, DSM

Distributed Memory Model

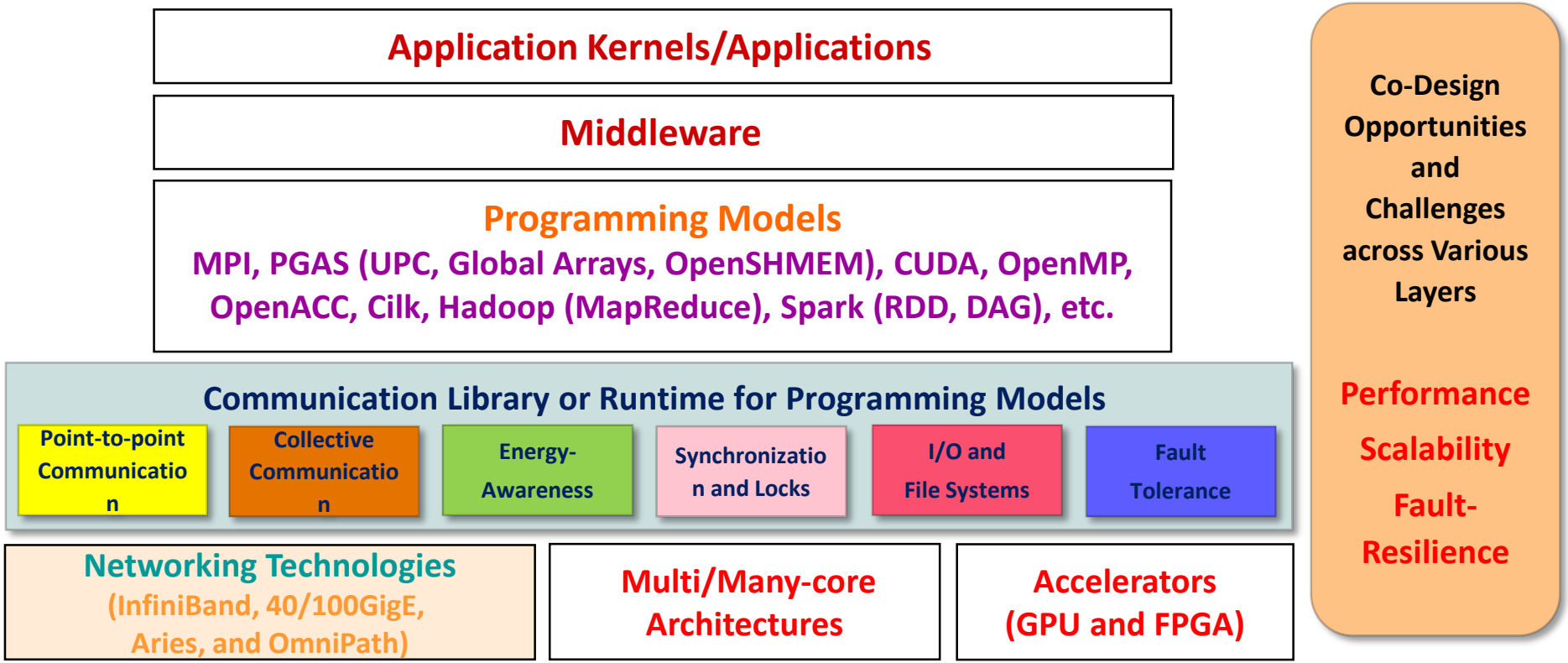
MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)

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 gradually receiving importance

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



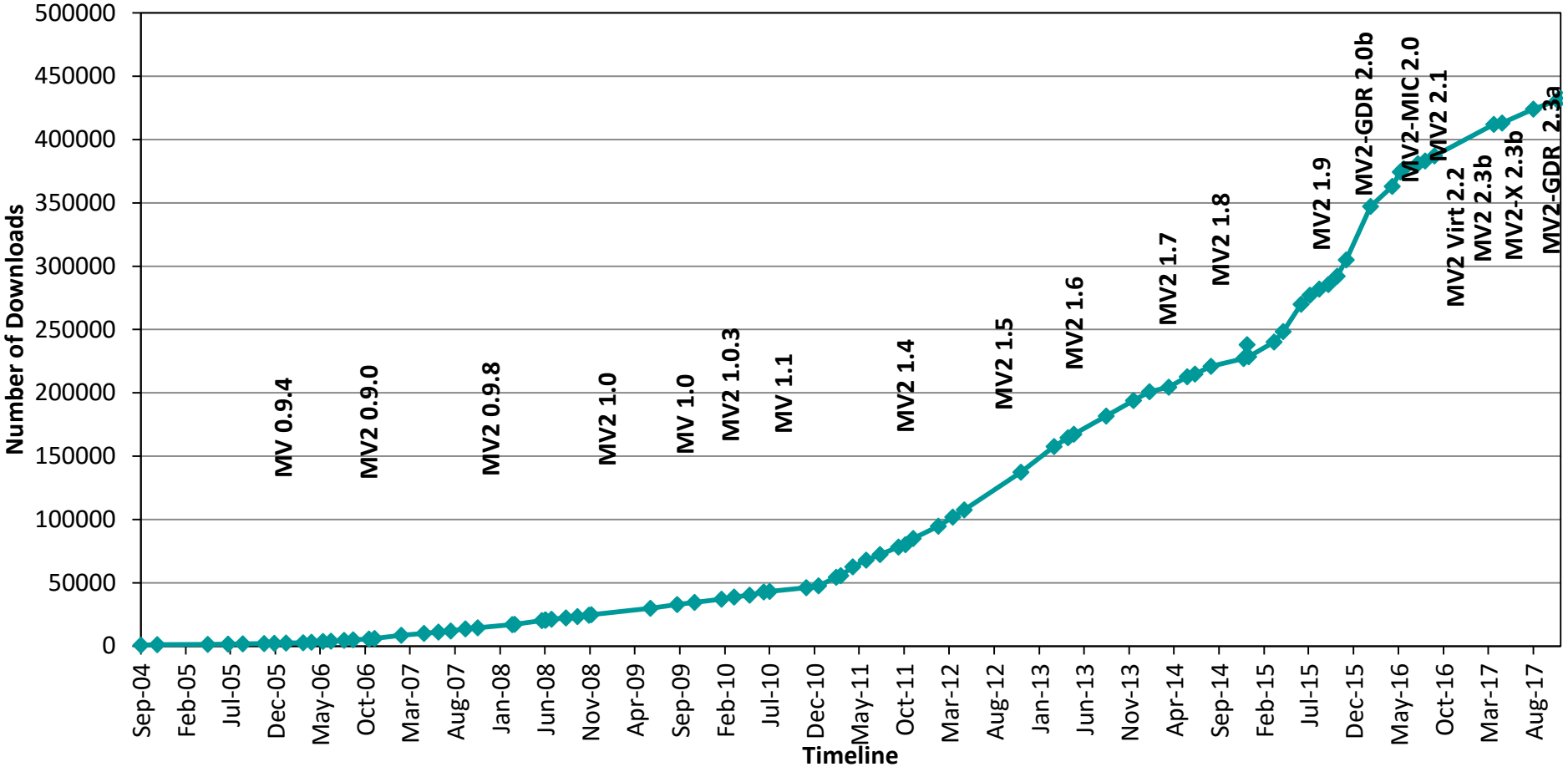
MPI+X Programming model: Broad Challenges at Exascale

- 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

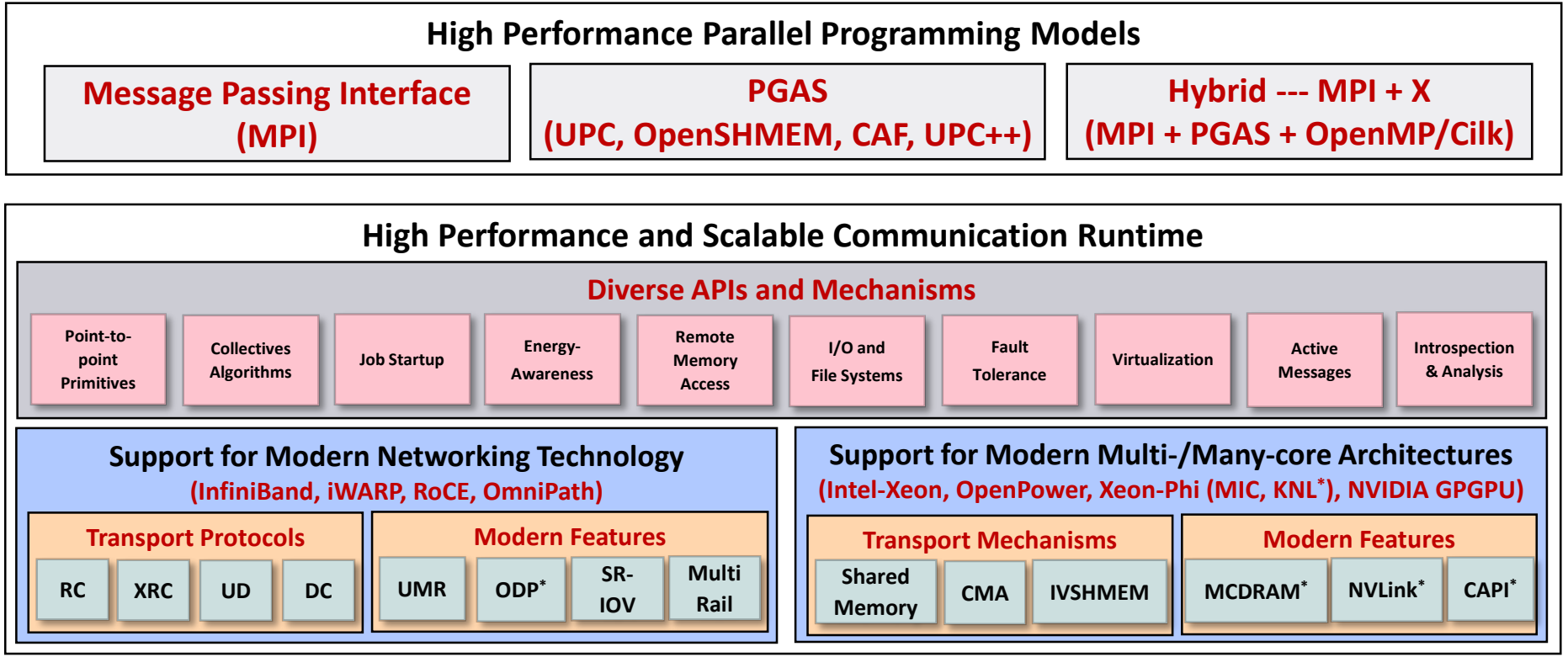
Overview of the MVAPICH2 Project

- 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-3.0), 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) since 2015
  - **Used by more than 2,825 organizations in 85 countries**
  - **More than 432,000 (> 0.4 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '17 ranking)
    - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
    - 15th, 241,108-core (Pleiades) at NASA
    - 20th, 462,462-core (Stampede) at TACC
    - 44th, 74,520-core (Tsubame 2.5) at Tokyo Institute of Technology
  - 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 (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Sunway TaihuLight (1st in Jun'17, 10M cores, 100 PFlops)

MVAPICH2 Release Timeline and Downloads



MVAPICH2 Architecture



* Upcoming

MVAPICH2 Software Family

High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud
MVAPICH2-EA	Energy aware and High-performance MPI
MVAPICH2-MIC	Optimized MPI for clusters with Intel KNC
Microbenchmarks	
OMB	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs
Tools	
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration
OEMT	Utility to measure the energy consumption of MPI applications

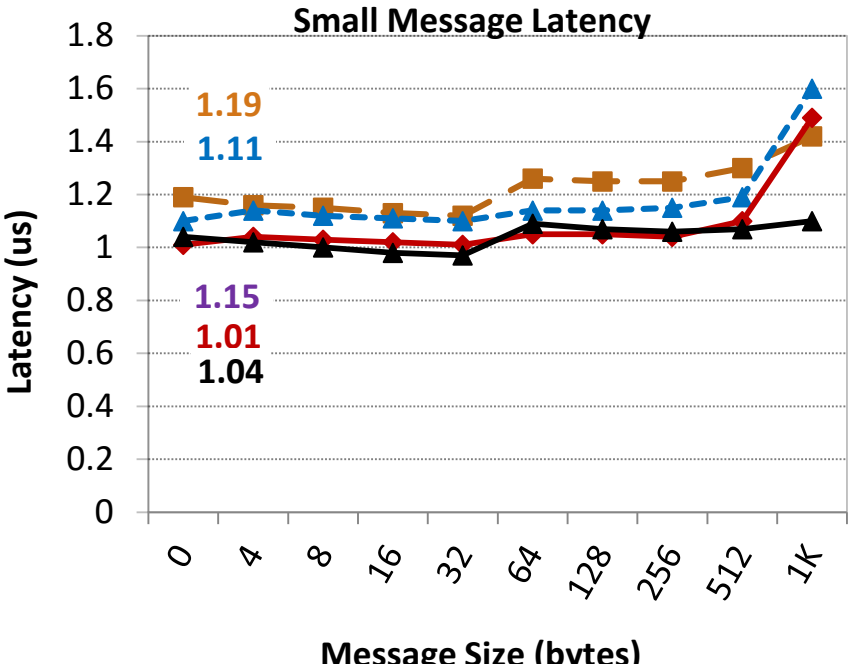
MVAPICH2 Distributions

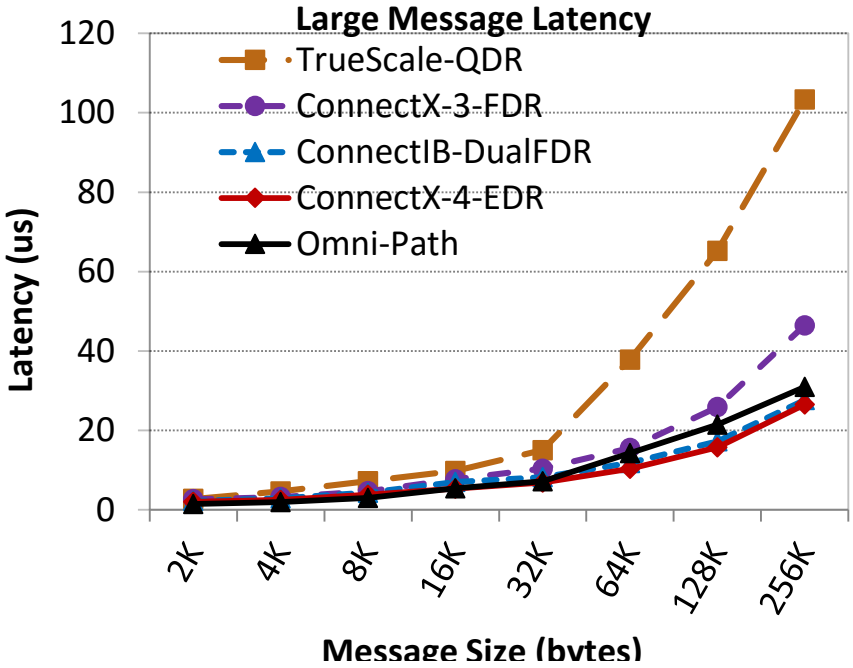
- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

MVAPICH2 2.3b

- Released on 08/10/2017
- Major Features and Enhancements
  - **Enhance performance of point-to-point operations for CH3-Gen2 (InfiniBand), CH3-PSM, and CH3-PSM2 (Omni-Path) channels**
  - **Improve performance for MPI-3 RMA operations**
  - **Introduce support for Cavium ARM (ThunderX) systems**
  - **Improve support for process to core mapping on many-core systems**
    - **New environment variable MV2_THREADS_BINDING_POLICY for multi-threaded MPI and MPI+OpenMP applications**
    - **Support `linear' and `compact' placement of threads**
    - **Warn user if oversubscription of core is detected**
  - **Improve launch time for large-scale jobs with mpirun_rsh**
  - **Add support for non-blocking Allreduce using Mellanox SHARP**
  - **Efficient support for different Intel Knight's Landing (KNL) models**
  - **Improve performance for Intra- and Inter-node communication for OpenPOWER architecture**
  - **Improve support for large processes per node and huge pages on SMP systems**
  - **Enhance collective tuning for Intel Knight's Landing and Intel Omni-Path based systems**
  - **Enhance collective tuning for Bebop@ANL, Bridges@PSC, and Stampede2@TACC systems**
  - **Enhance large message intra-node performance with CH3-IB-Gen2 channel on Intel Knight's Landing**
  - **Enhance support for MPI_T PVARs and CVARs**

One-way Latency: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

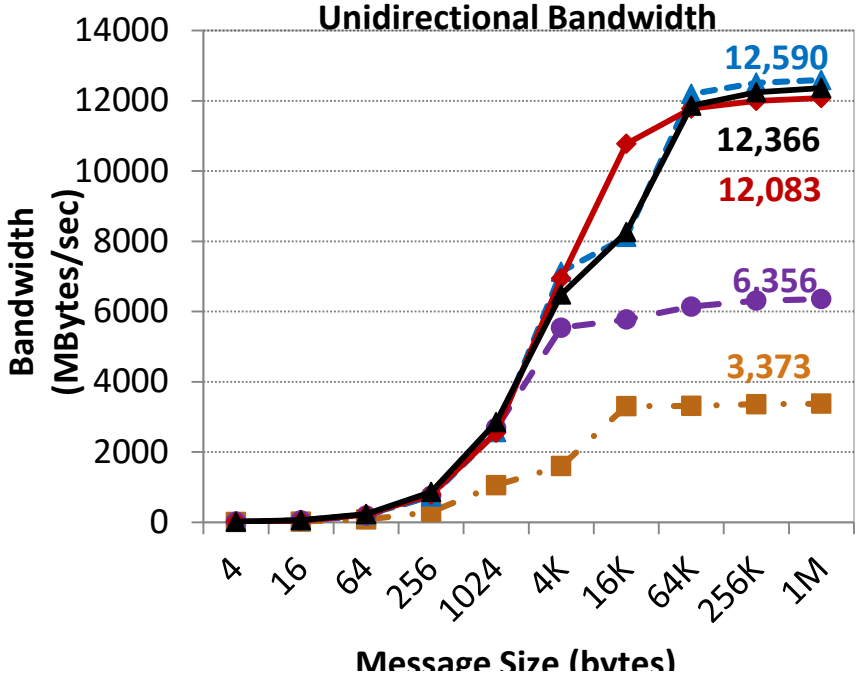
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

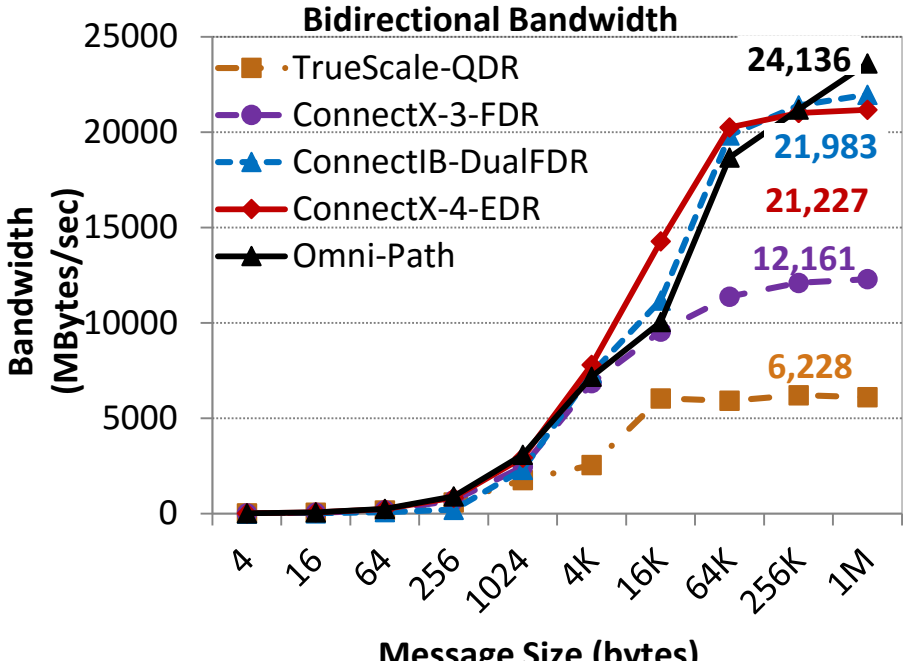
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

Bandwidth: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

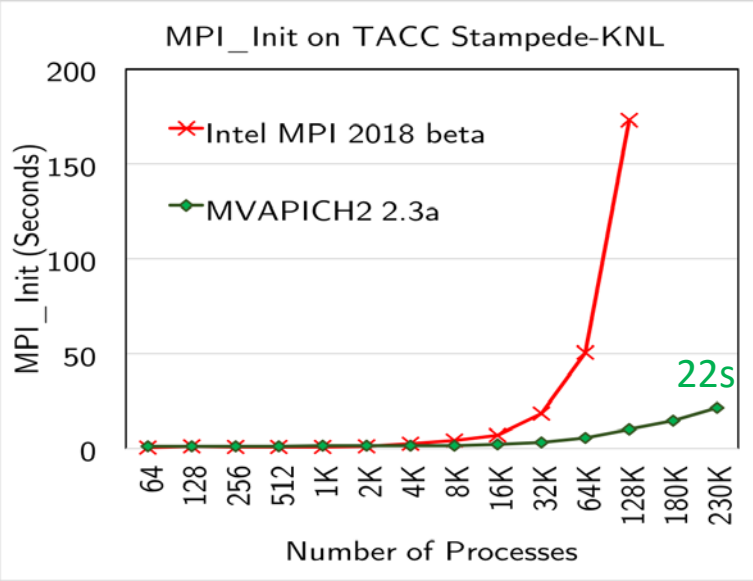
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

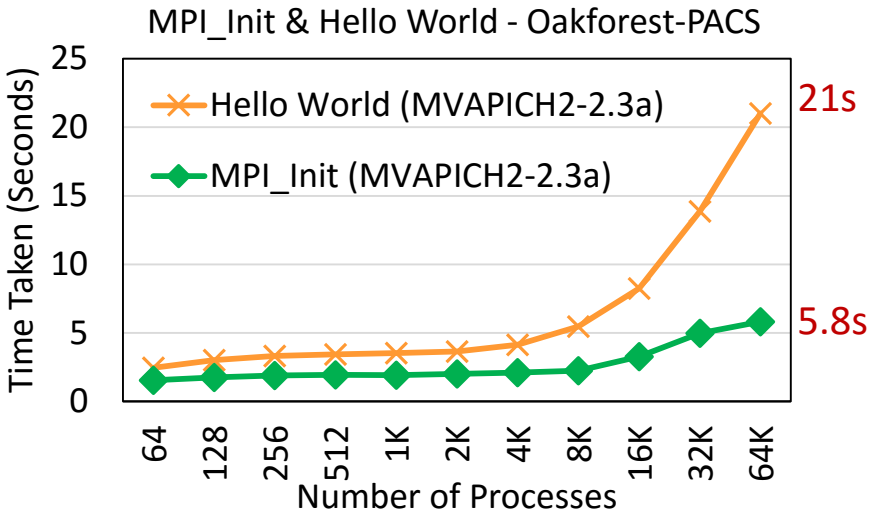
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 IB switch

Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

Startup Performance on KNL + Omni-Path

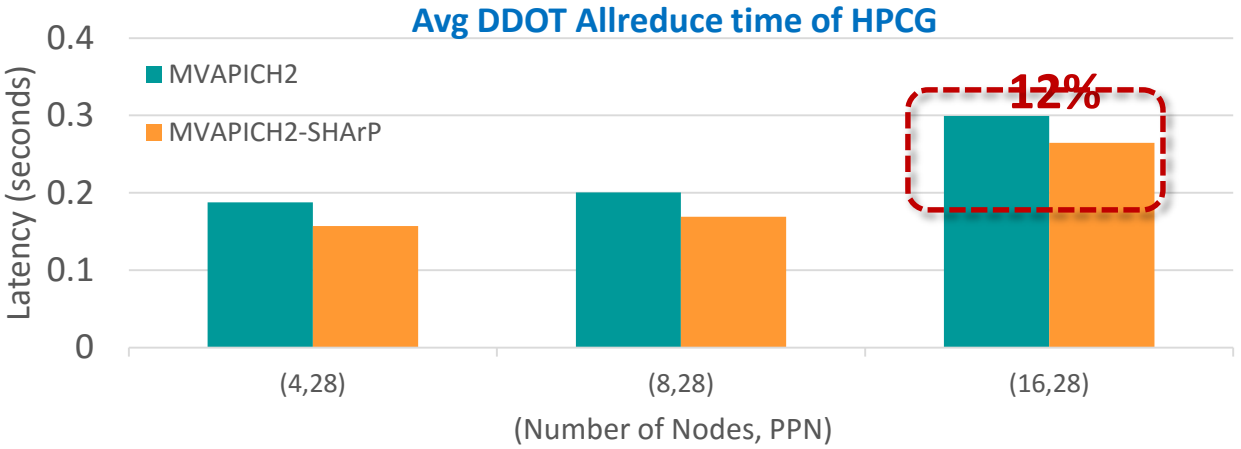


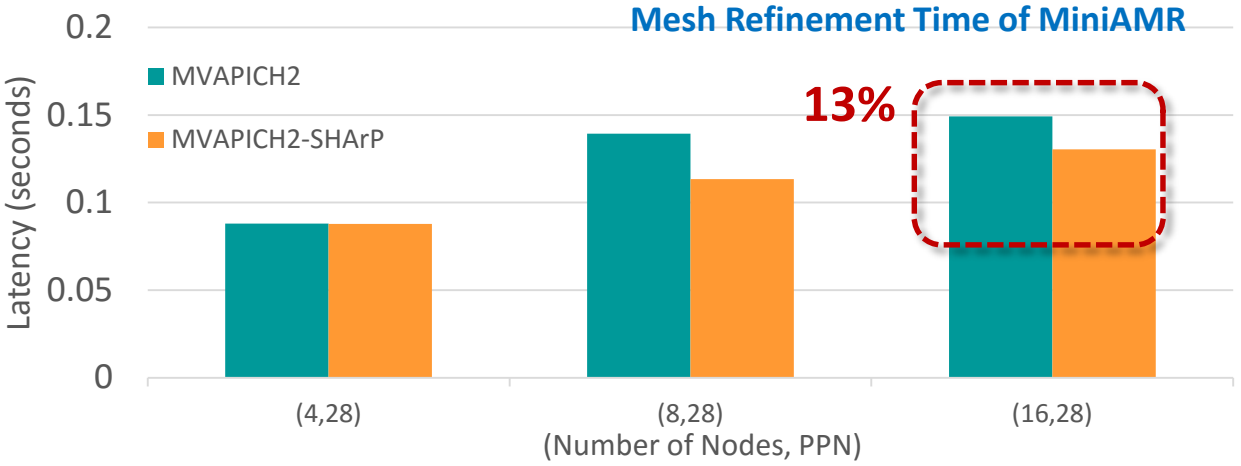


- MPI_Init takes 22 seconds on 229,376 processes on 3,584 KNL nodes (Stampede2 – Full scale)
- 8.8 times faster than Intel MPI at 128K processes (Courtesy: TACC)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per node

New designs available in latest MVAPICH2 libraries and as patch for SLURM-15.08.8 and SLURM-16.05.1

Advanced Allreduce Collective Designs Using SHArP

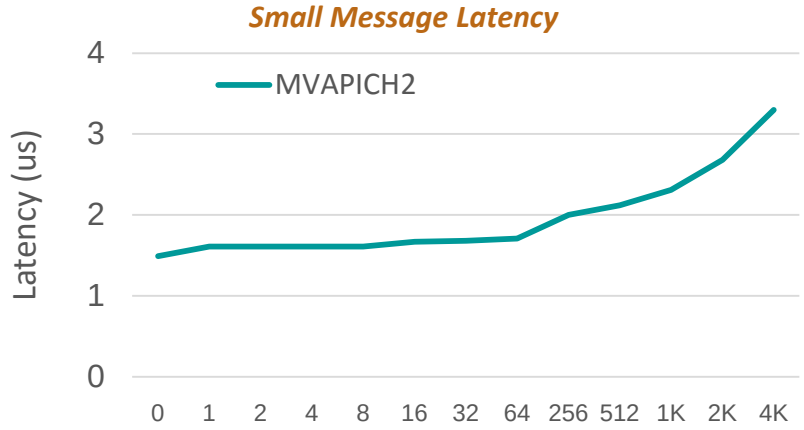


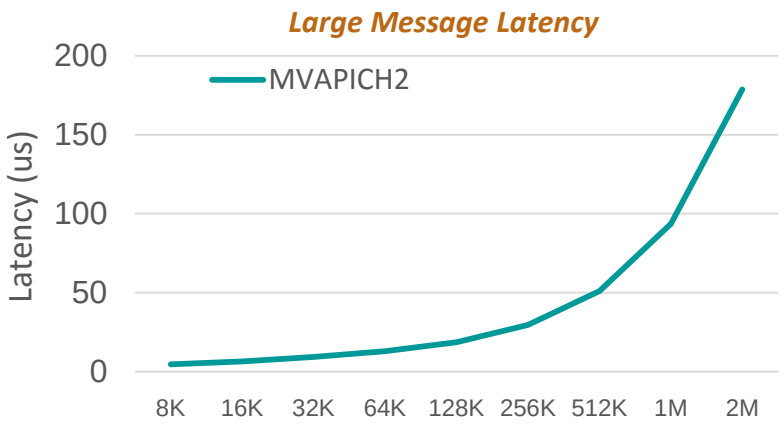


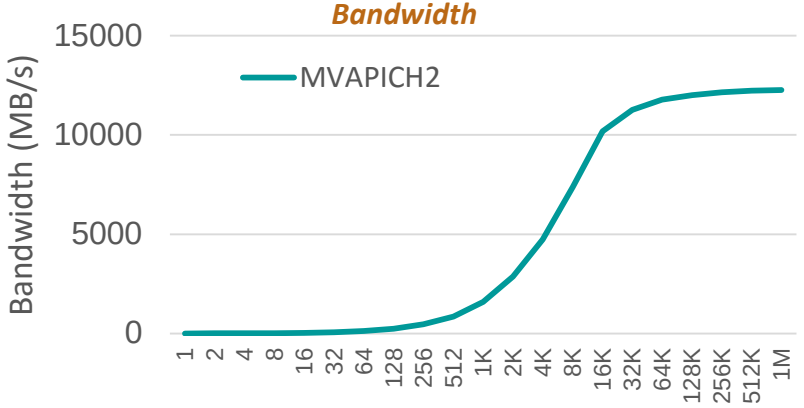
SHArP Support is available since MVAPICH2 2.3a

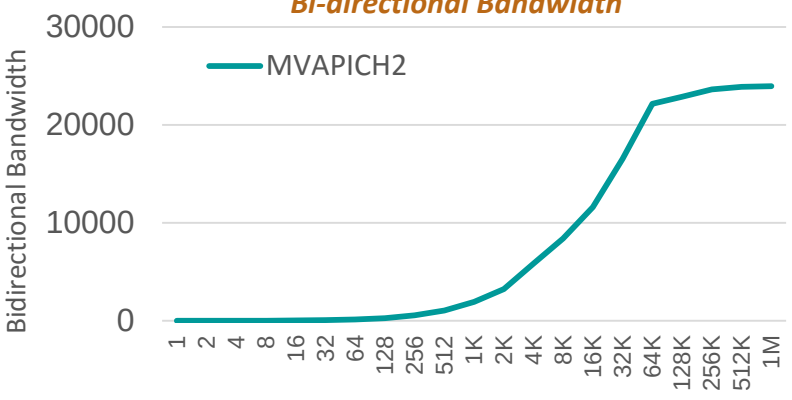
M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, and D. K. Panda, Scalable Reduction Collectives with Data Partitioning-based Multi-Leader Design, SuperComputing '17.

Inter-node Point-to-Point Performance on OpenPower



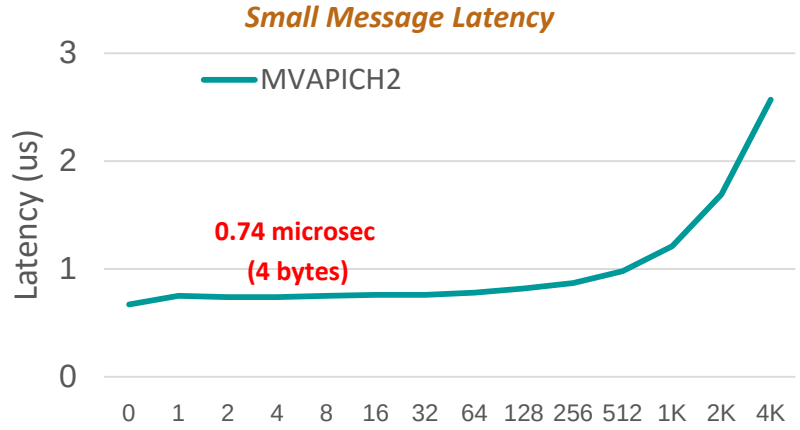


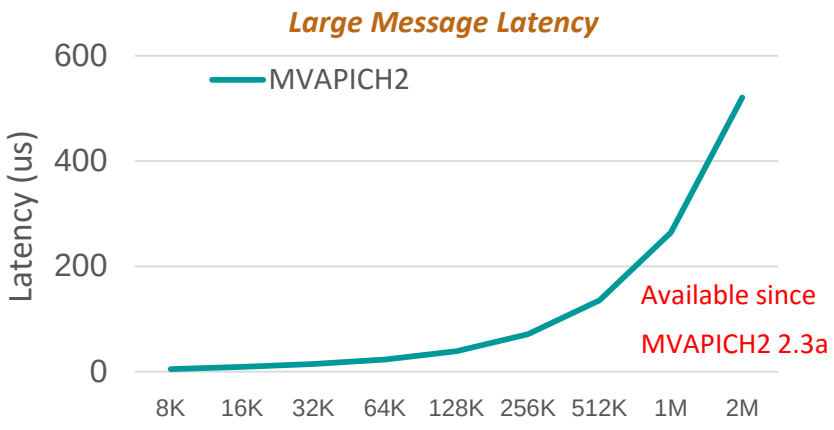


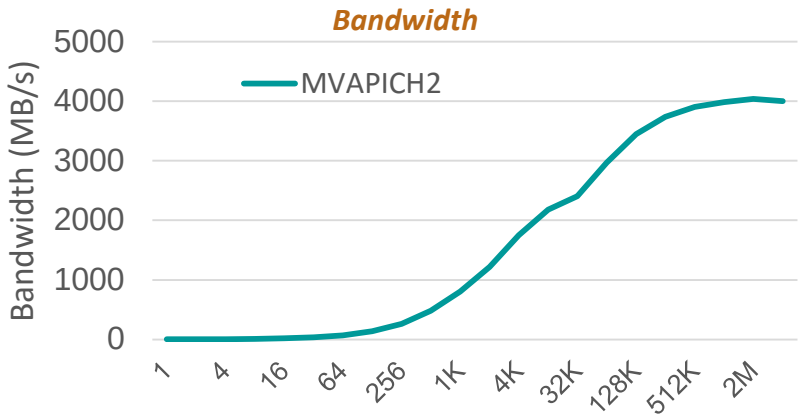


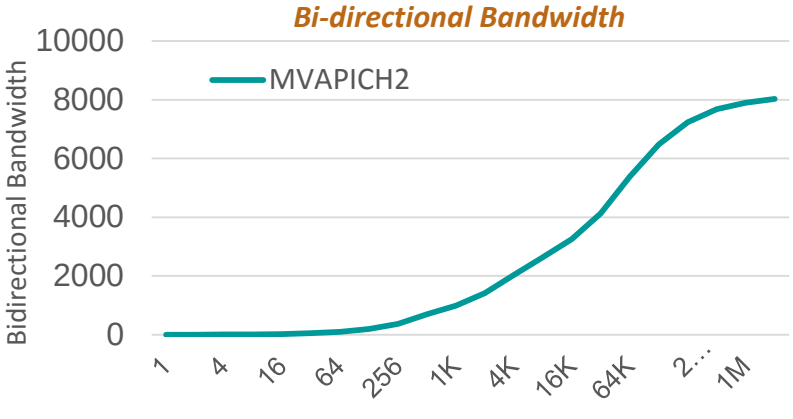
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA.

Intra-node Point-to-point Performance on ARMv8





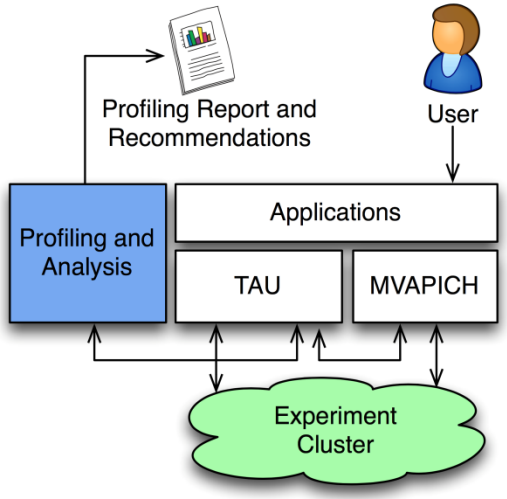




Platform: ARMv8 (aarch64) MIPS processor with 96 cores dual-socket CPU. Each socket contains 48 cores.

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 non-interactive mode for uninstrumented applications



VBUF usage without CVAR based tuning as displayed by ParaProf

Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory in bytes 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 MPIT_calloc calls)	89	89	89	0	1	89

VBUF usage with CVAR based tuning as displayed by ParaProf

Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamp...	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

Challenge

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

Solution

A new tag matching design

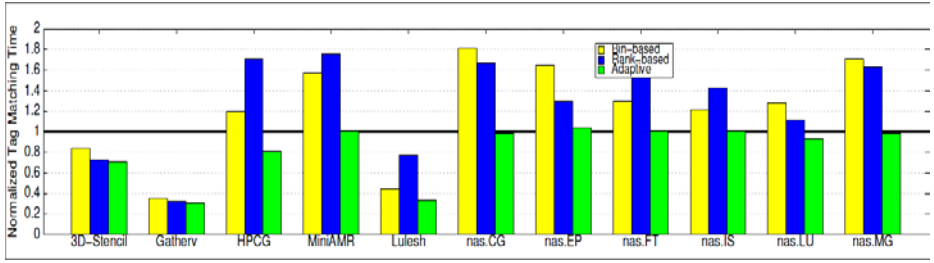
- Dynamically adapt to communication patterns
- Use different strategies for different ranks
- Decisions are based on the number of request object that must be traversed before hitting on the required one

Results

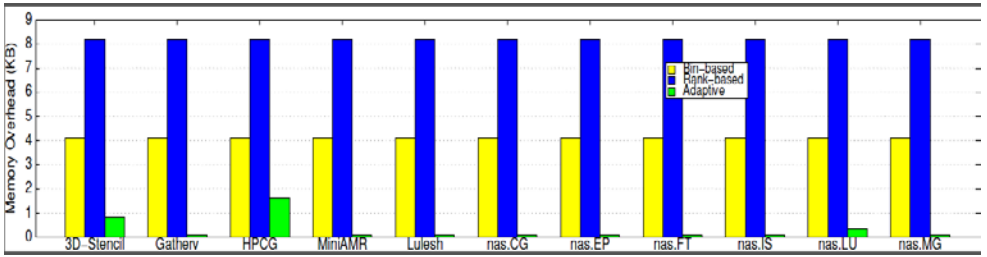
Better performance than other state-of-the art tag-matching schemes

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)

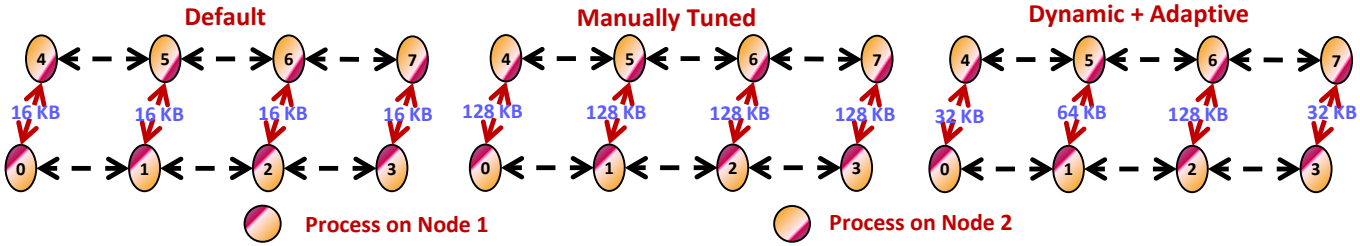
Adaptive and Dynamic Design for MPI Tag Matching; M. Bayatpour, H. Subramoni, S. Chakraborty, and D. K. Panda; IEEE Cluster 2016. [Best Paper Nominee]

Dynamic and Adaptive MPI Point-to-point Communication Protocols

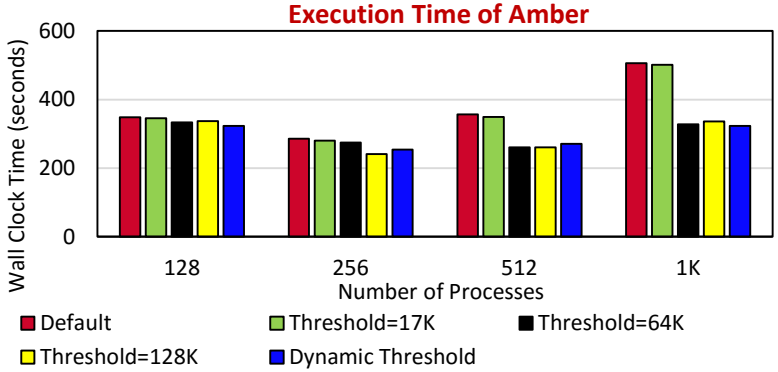
Desired Eager Threshold

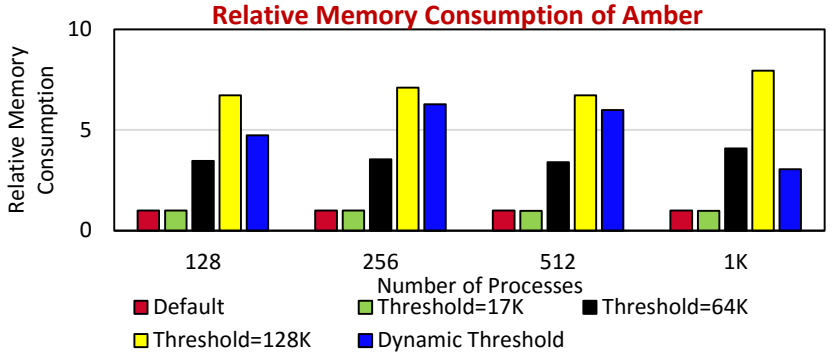
Process Pair	Eager Threshold (KB)
0 – 4	32
1 – 5	64
2 – 6	128
3 – 7	32

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



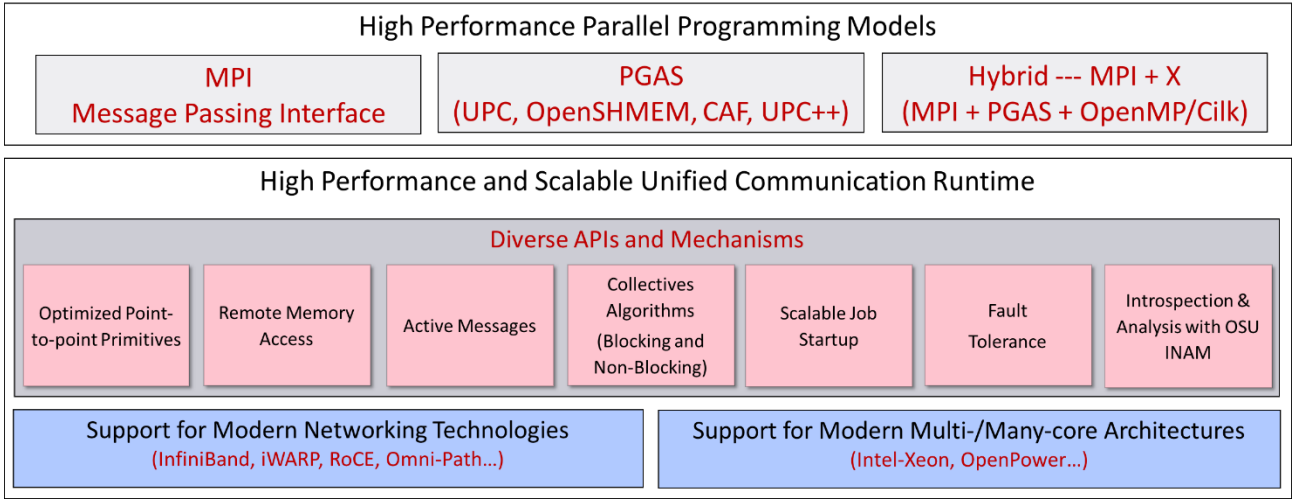


H. Subramoni, S. Chakraborty, D. K. Panda, Designing Dynamic & Adaptive MPI Point-to-Point Communication Protocols for Efficient Overlap of Computation & Communication, ISC'17 - **Best Paper**

MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

MVAPICH2-X for Hybrid MPI + PGAS Applications

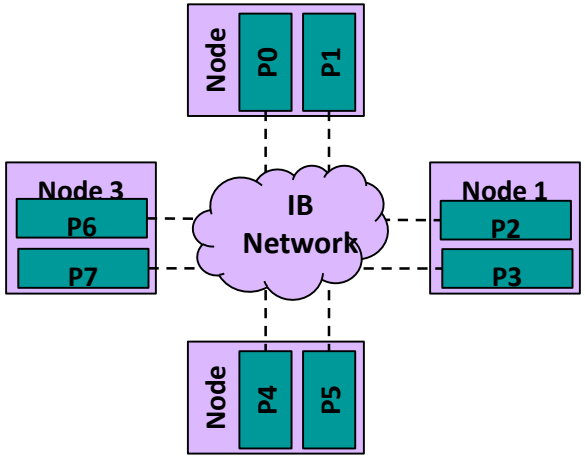


- Current Model – Separate Runtimes for OpenSHMEM/UPC/UPC++/CAF and MPI
  - Possible deadlock if both runtimes are not progressed
  - Consumes more network resource
- Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF
  - Available with since 2012 (starting with MVAPICH2-X 1.9)
  - http://mvapich.cse.ohio-state.edu

MVAPICH2-X 2.3b

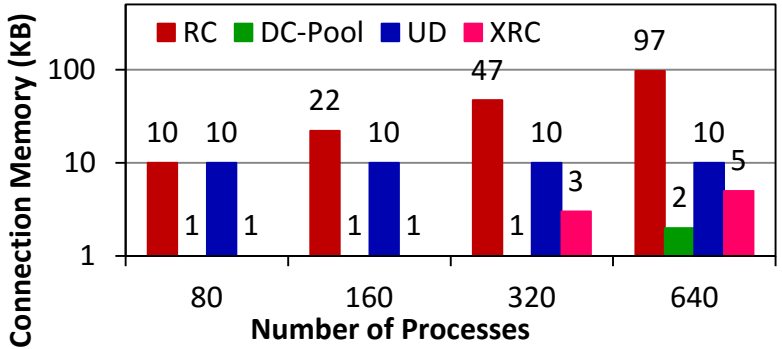
- Released on 10/30/2017
- Major Features and Enhancements
  - MPI Features
    - Based on MVAPICH2 2.3b
    - OFA-IB-CH3, PSM-CH3, and PSM2-CH3 interfaces
    - Support for ARM architecture
    - Optimized support for OpenPOWER architecture
    - Collective tuning for ARM architecture
    - Collective tuning for Intel Skylake architecture
  - MPI (Advanced) Features
    - Support Data Partitioning-based Multi-Leader Design (DPML) for MPI collectives
    - OFA-IB-CH3, PSM-CH3, and PSM2-CH3 interfaces
    - Support Contention Aware Kernel-Assisted MPI collectives
      - OFA-IB-CH3, PSM-CH3, and PSM2-CH3 interfaces
    - Support for OSU InfiniBand Network Analysis and Management (OSU INAM) Tool v0.9.2
  - OpenSHMEM Features
    - Based on OpenSHMEM reference implementation 1.3
    - Support Non-Blocking remote memory access routines
  - Unified Runtime Features
    - Based on MVAPICH2 2.3b (OFA-IB-CH3 interface). All the runtime features enabled by default in OFA-IB-CH3 and OFA-IB-RoCE interface of MVAPICH2 2.3b are available in MVAPICH2-X 2.3b interface of MVAPICH2 2.2 GA are available in MVAPICH2-X 2.2 GA

Minimizing Memory Footprint by Direct Connect (DC) Transport

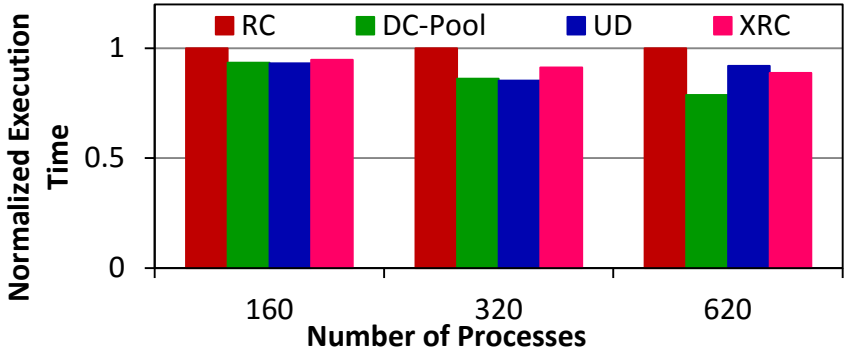


- Constant connection cost (*One QP for any peer*)
- Full Feature Set (RDMA, Atomics etc)
- Separate objects for send (DC Initiator) and receive (DC Target)
  - DC Target identified by "DCT Number"
  - Messages routed with (DCT Number, LID)
  - Requires same "DC Key" to enable communication
- Available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

Memory Footprint for Alltoall



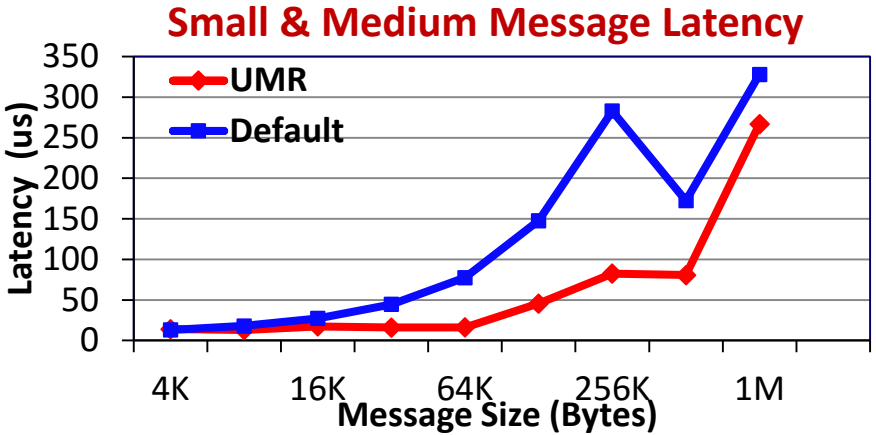
NAMD - Apoa1: Large data set

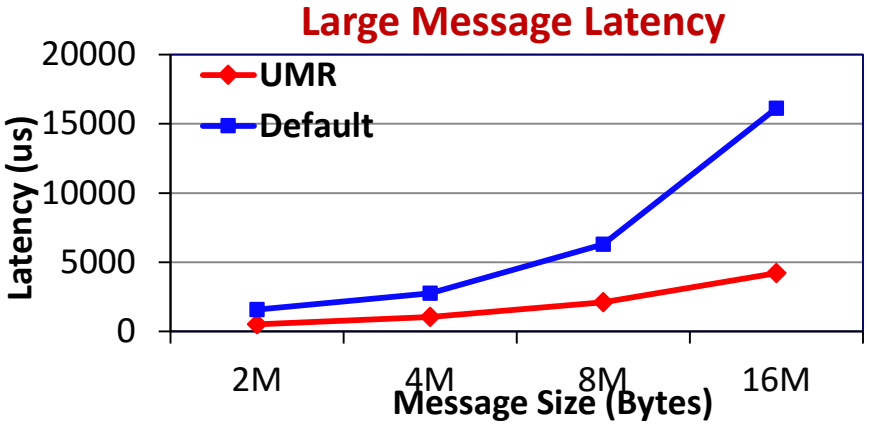


H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty and D. K. Panda, Designing MPI Library with Dynamic Connected Transport (DCT) of InfiniBand : Early Experiences. IEEE International Supercomputing Conference (ISC '14)

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 in MVAPICH2-X 2.2b



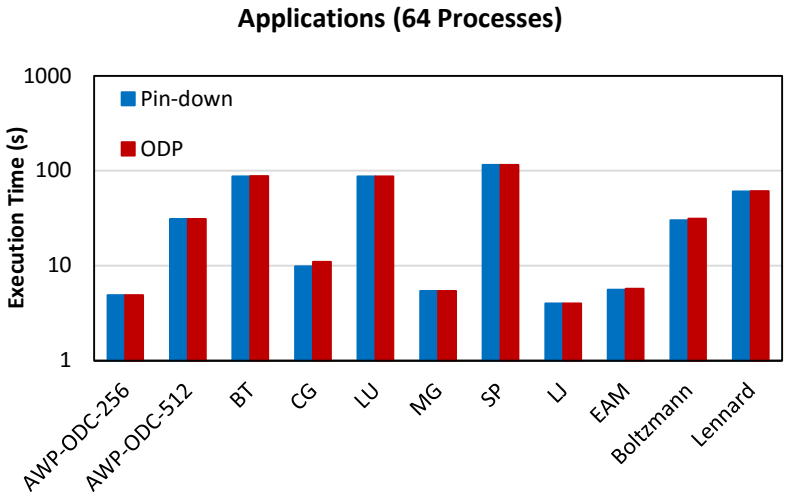


Connect-IB (54 Gbps): 2.8 GHz Dual Ten-core (IvyBridge) Intel PCI Gen3 with Mellanox IB FDR switch

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

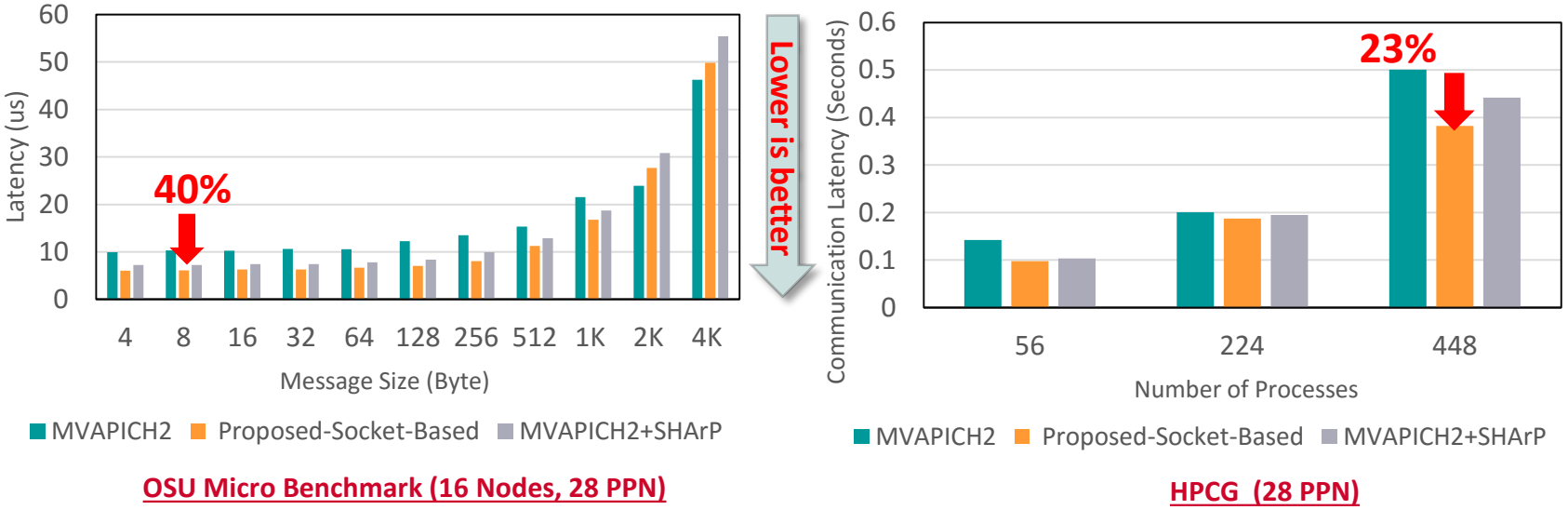
On-Demand Paging (ODP)

- 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
  - Physical memory optimization



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.

Advanced Allreduce Collective Designs



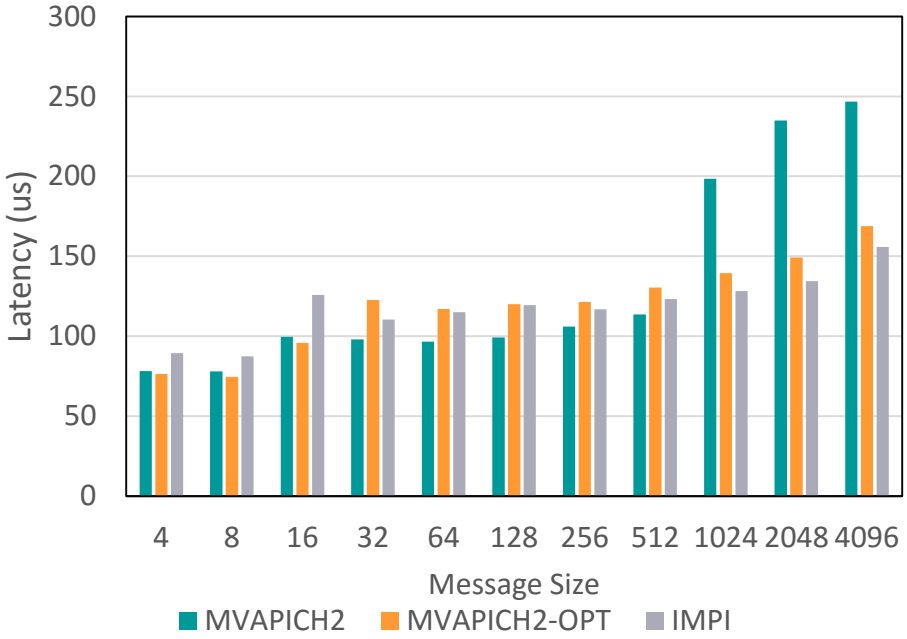
OSU Micro Benchmark (16 Nodes, 28 PPN)

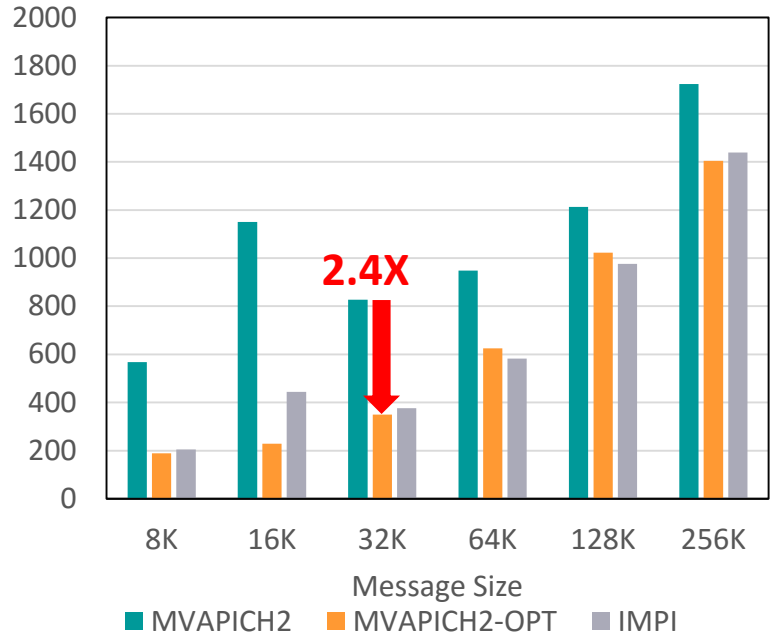
HPCG (28 PPN)

- Socket-based design can reduce the communication latency by 23% and 40% on Xeon + IB nodes
- **Support is available in MVAPICH2-X 2.3b**

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

Performance of MPI_Allreduce On Stampede2 (10,240 Processes)

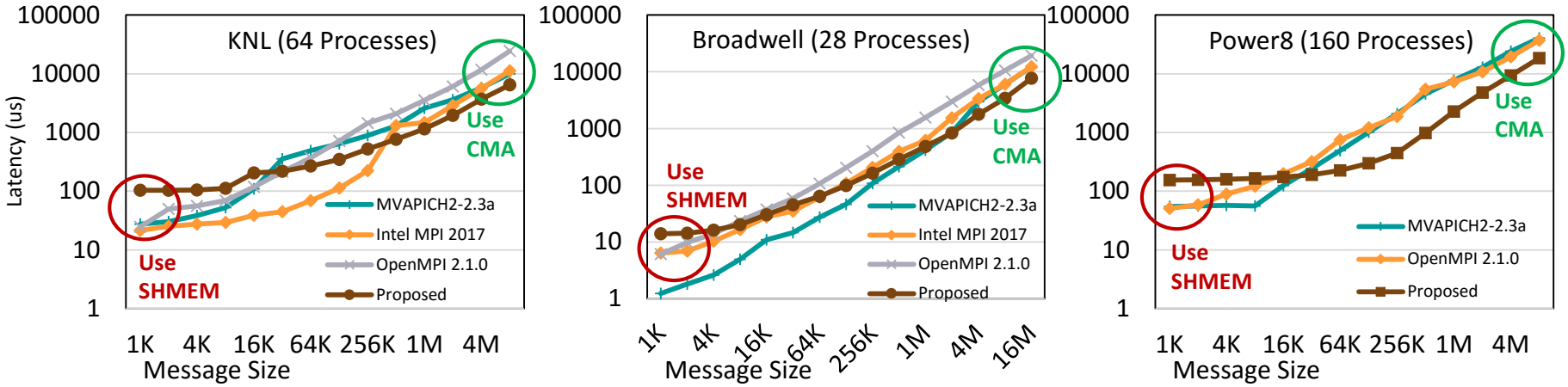




OSU Micro Benchmark 64 PPN

- MPI_Allreduce latency with 32K bytes reduced by 2.4X

Enhanced MPI_Bcast with Optimized CMA-based Design



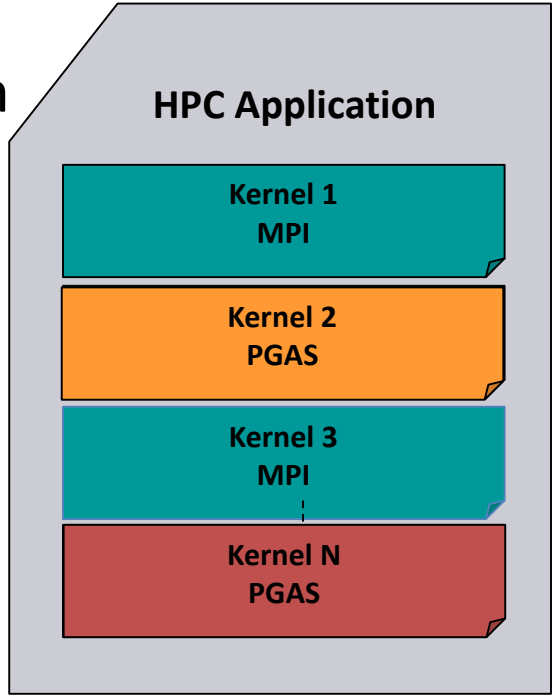
- Up to 2x - 4x improvement over existing implementation for 1MB messages
- Up to 1.5x – 2x faster than Intel MPI and Open MPI for 1MB messages
- Improvements obtained for large messages only
  - p-1 copies with CMA, p copies with Shared memory
  - Fallback to SHMEM for small messages

Support is available in MVAPICH2-X 2.3b

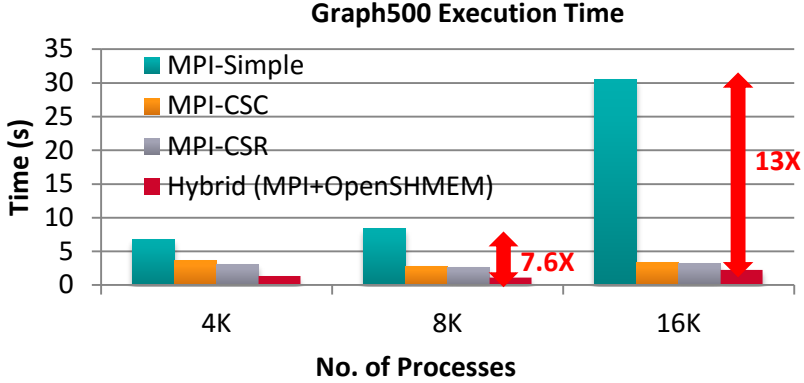
S. Chakraborty, H. Subramoni, and D. K. Panda, Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems, *IEEE Cluster '17, BEST Paper Finalist*

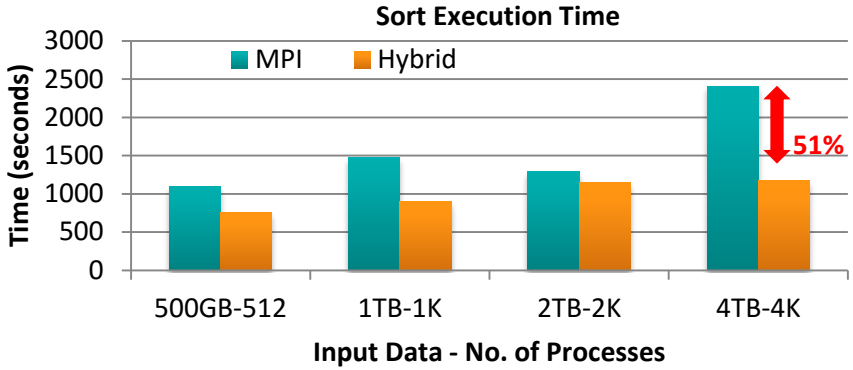
Hybrid (MPI+PGAS) Programming

- 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, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

Can HPC and Virtualization be Combined?

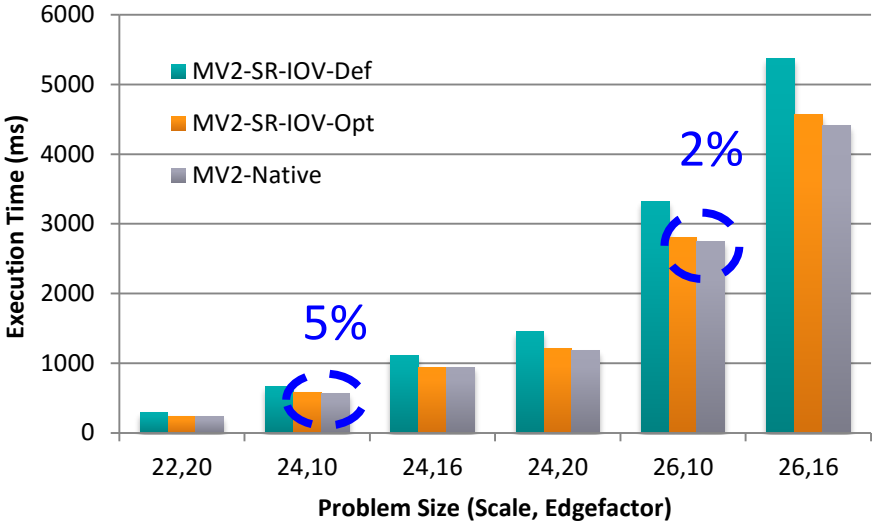
- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2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

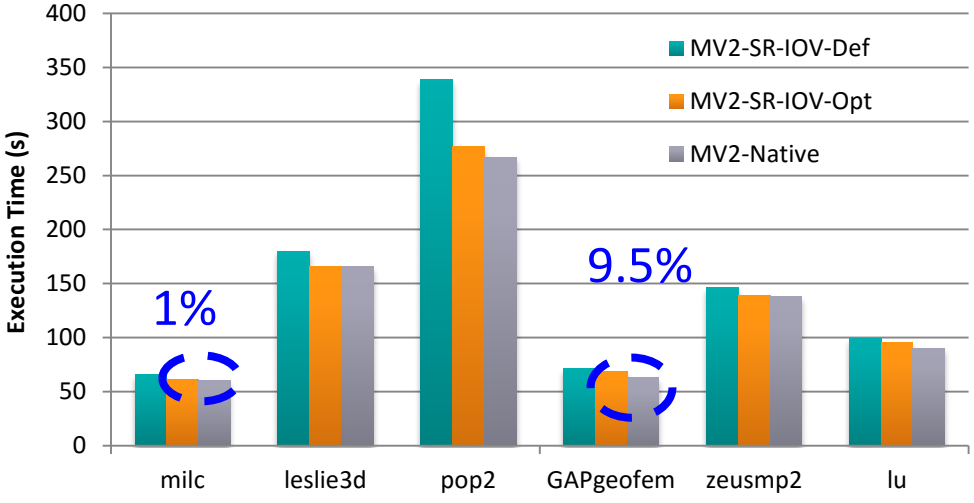
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC'14

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



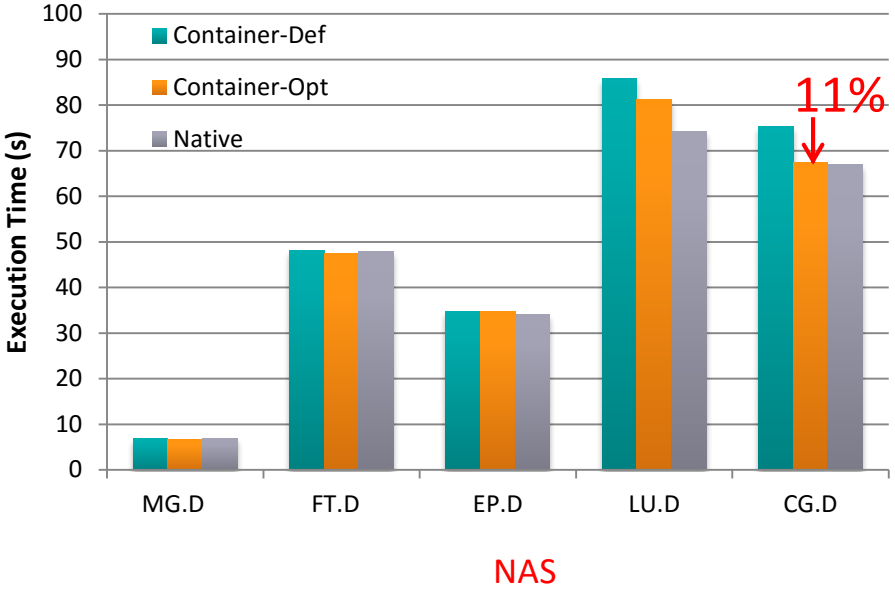
Graph500



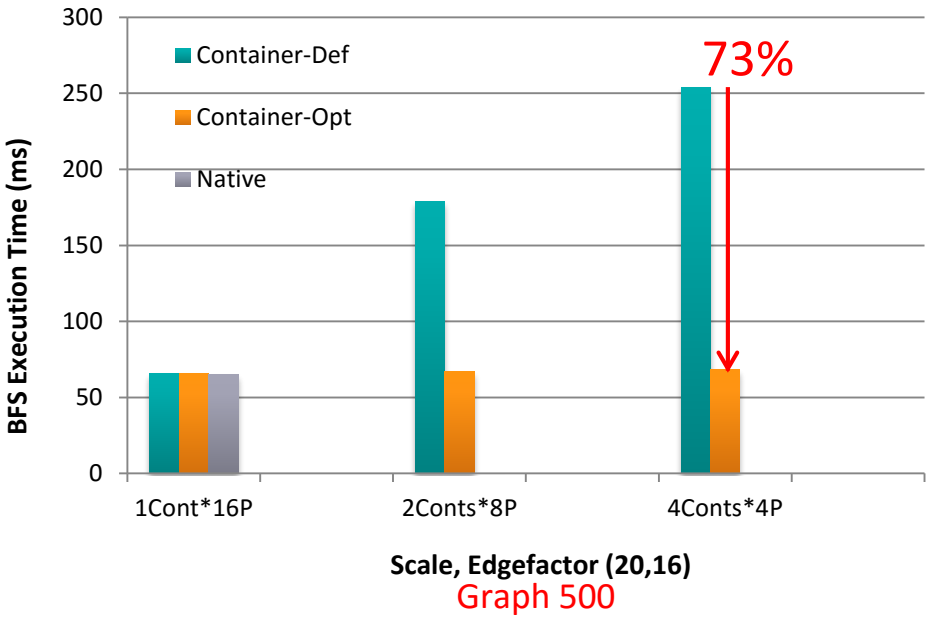
SPEC MPI2007

- 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

Application-Level Performance on Docker with MVAPICH2



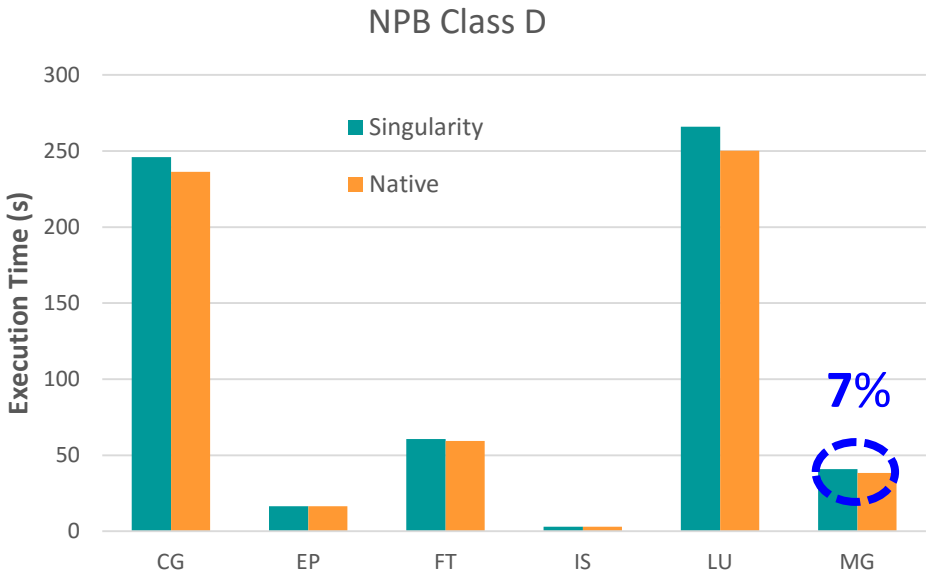
NAS

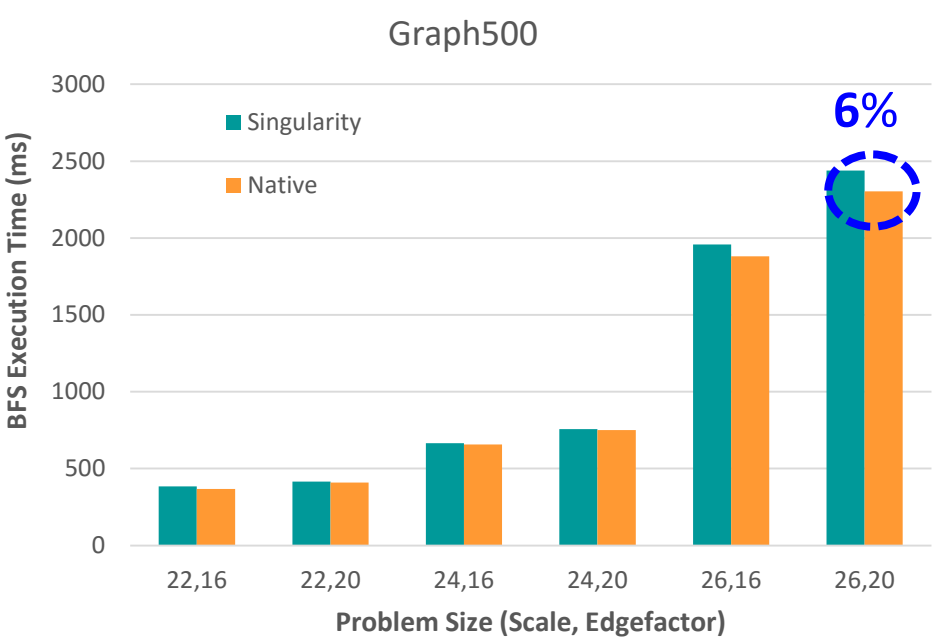


Graph 500

- 64 Containers across 16 nodes, pining 4 Cores per Container
- Compared to Container-Def, up to 11% and 73% of execution time reduction for NAS and Graph 500
- Compared to Native, less than 9 % and 5% overhead for NAS and Graph 500

Application-Level Performance on Singularity with MVAPICH2





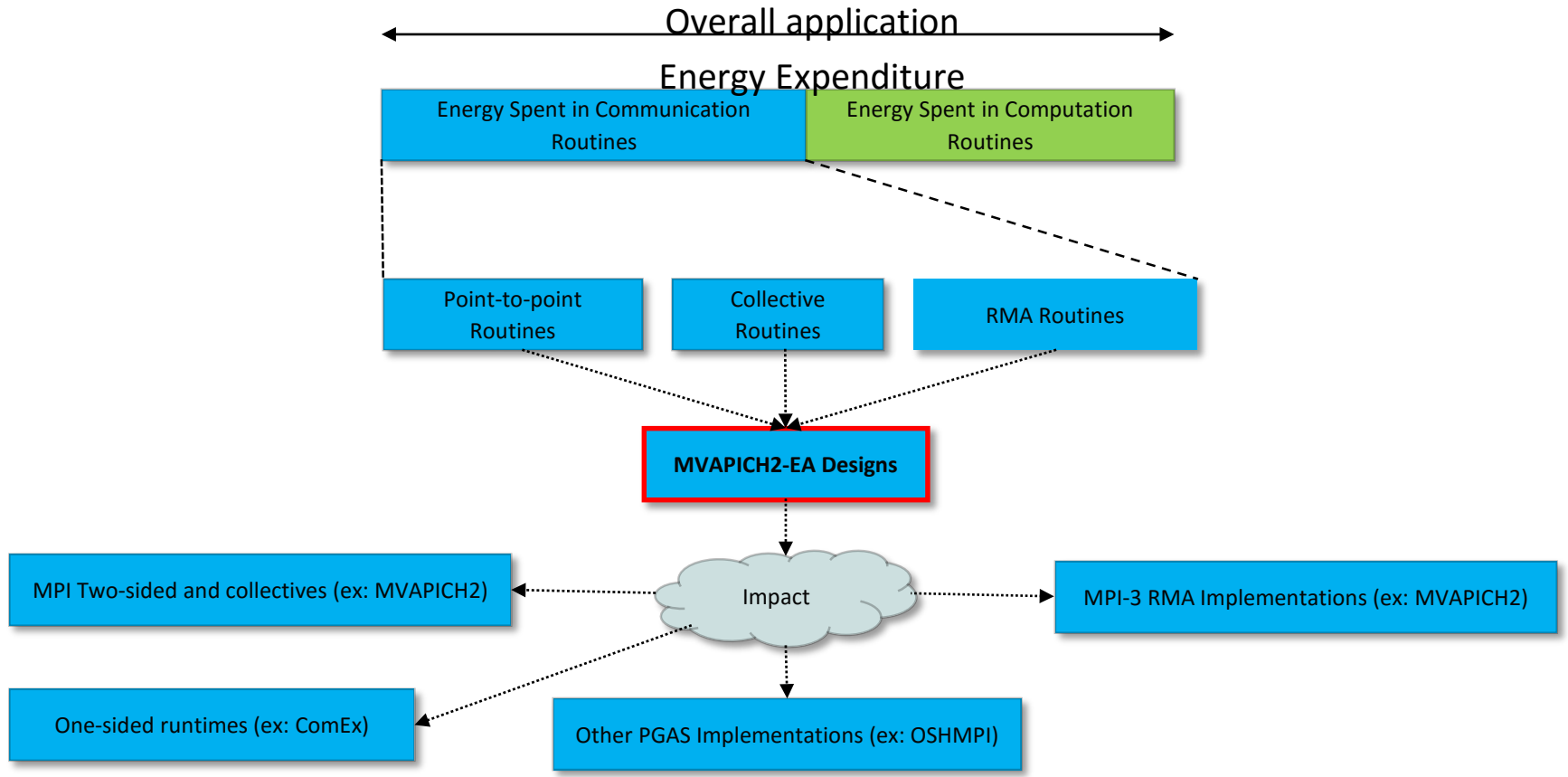
- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

J. Zhang, X .Lu and D. K. Panda, Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?, UCC '17

MVAPICH2 Distributions

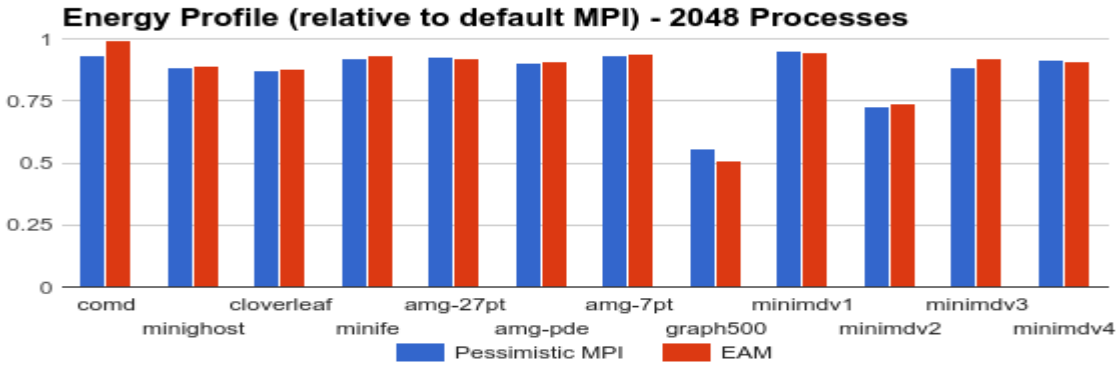
- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

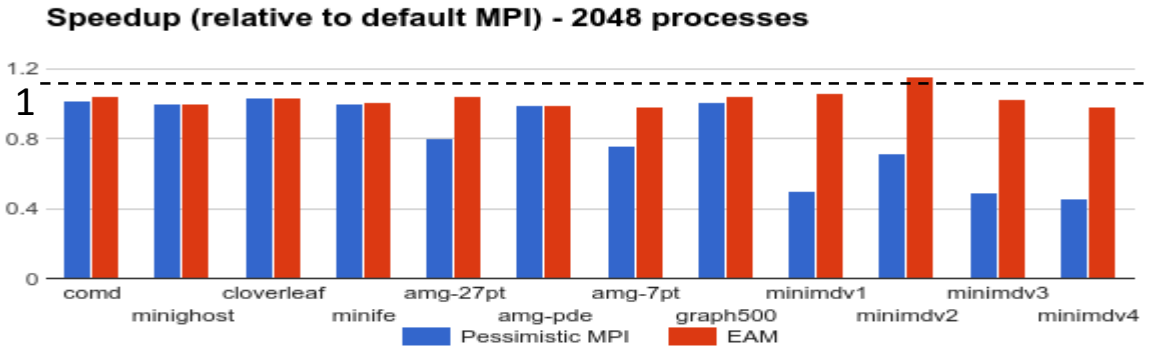
Designing Energy-Aware (EA) MPI Runtime



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
- Released (together with OEMT) since Aug'15





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]*

MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

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
  - http://mvapich.cse.ohio-state.edu/benchmarks/

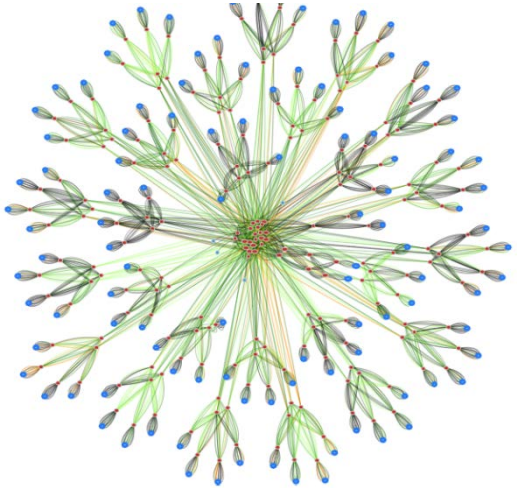
MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
  - Advanced MPI features and support for INAM
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)

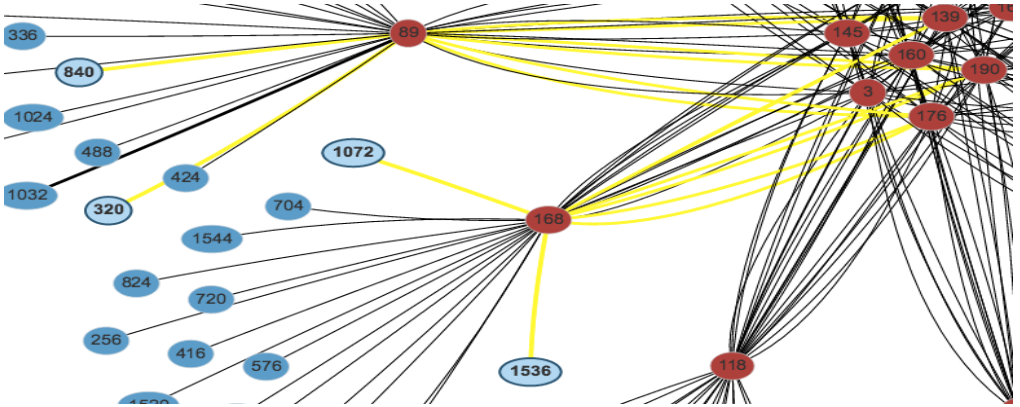
Overview of OSU INAM

- 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
- OSU INAM v0.9.2 released on 10/31/2017
- Significant enhancements to user interface to enable scaling to clusters with thousands of nodes
- Improve database insert times by using 'bulk inserts‘
- Capability to look up list of nodes communicating through a network link
- Capability to classify data flowing over a network link at job level and process level granularity in conjunction with MVAPICH2-X 2.3b
- “Best practices “ guidelines for deploying OSU INAM on different clusters
- 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 Features



Comet@SDSC --- Clustered View

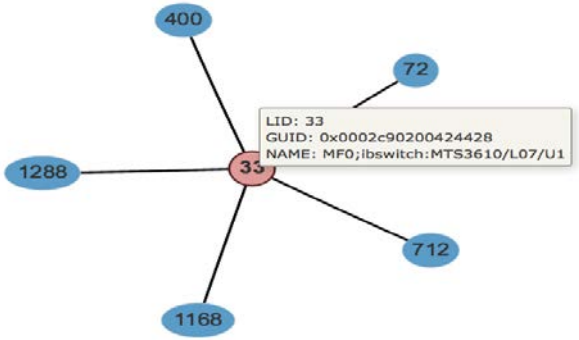


Finding Routes Between Nodes

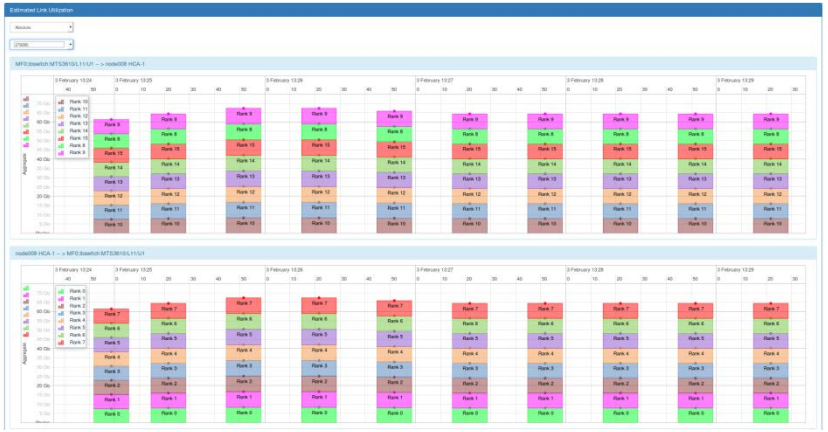
(1,879 nodes, 212 switches, 4,377 network links)

- 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)



Estimated Process Level Link Utilization

- 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
  - Bytes sent/received for MPI operations (pt-to-pt, collective, RMA)
  - Network metrics (e.g. XmitDiscard, RcvError) per rank/node

- Estimated Link 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
- Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu.
- We will link these results with credits to you.

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
  - Multi-host Adapters*
  - Hardware-based Tag Matching*
- Enhanced communication schemes for upcoming architectures
  - Knights Landing with MCDRAM*
  - CAPI*
- Extended topology-aware collectives
- Extended Energy-aware designs and Virtualization Support
- Extended Support for MPI Tools Interface (as in MPI 3.1)
- Extended Checkpoint-Restart and migration support with SCR
- Support for * features will be available in future MVAPICH2 Releases

Two More Presentations

- Wednesday (11/15/16) at 3:00pm

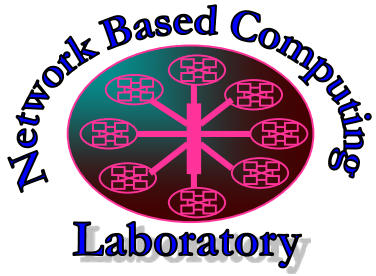
  High Performance Big Data (HiBD): Accelerating Hadoop, Spark and Memcached on Modern Clusters

- Thursday (11/16/16) at 10:30am

  MVAPICH2-GDR for HPC and Deep Learning

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/

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