MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

High-performance and Scalable MPI+X Library for Emerging HPC Clusters & Cloud Platforms

Talk at Intel HPC Developer Conference (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

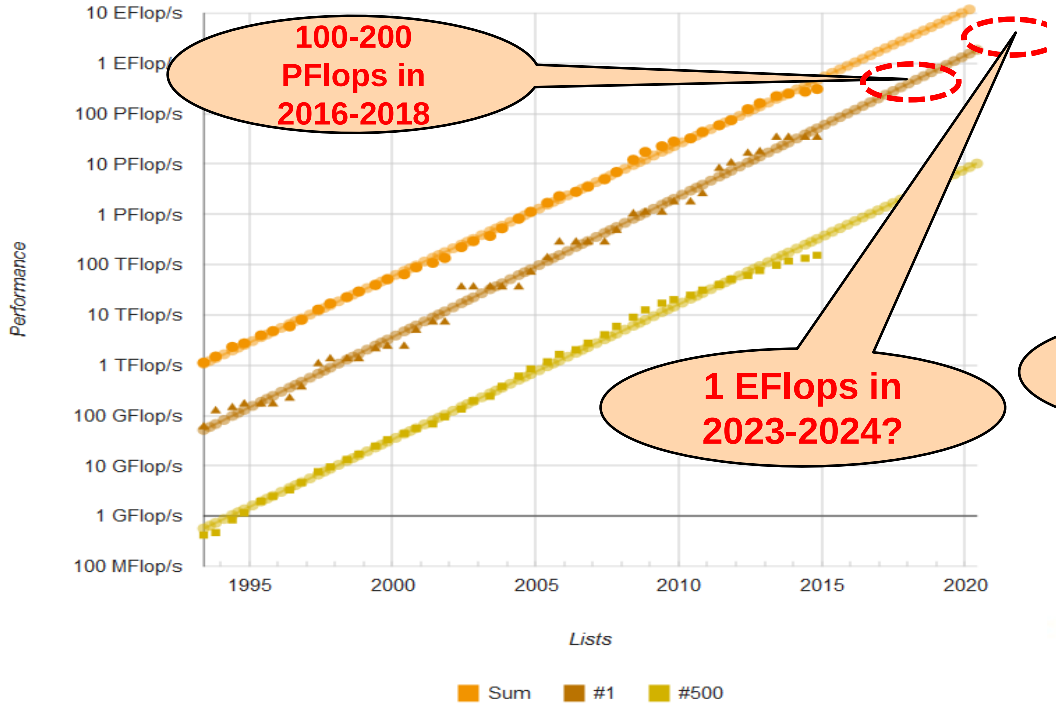
Hari Subramoni

The Ohio State University

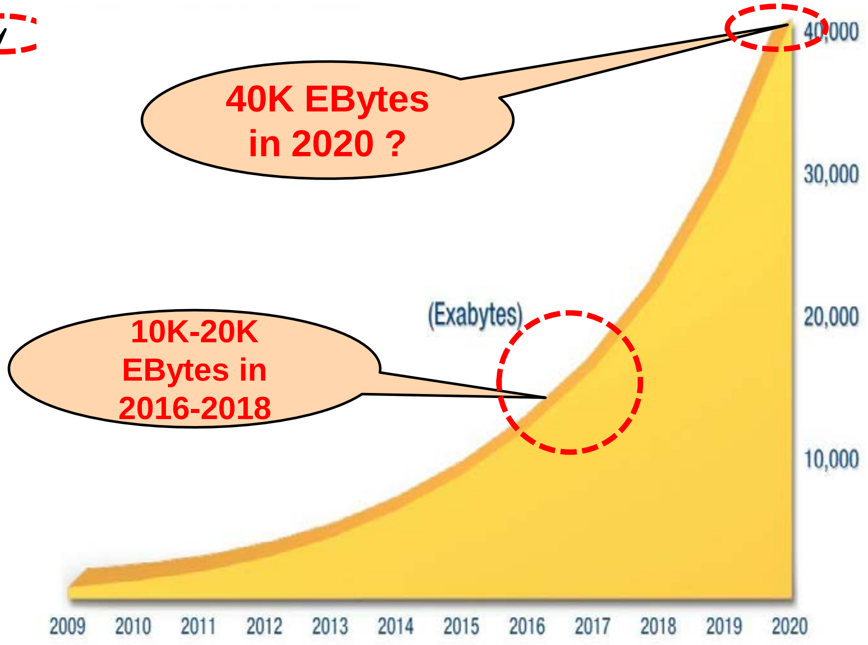
E-mail: subramon@cse.ohio-state.edu

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

High-End Computing (HEC): ExaFlop & ExaByte



ExaFlop & HPC



ExaByte & BigData

Drivers of Modern HPC Cluster Architectures

Multi-core Processors

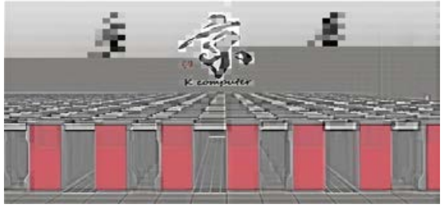
High Performance Interconnects – InfiniBand, Omni-Path
<1usec latency, 100Gbps Bandwidth>

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

SSD, NVMe-SSD, NVRAM

- Multi-core/many-core technologies
- High Performance Interconnects
- High Performance Storage and Compute devices
- MPI is used by vast majority of HPC applications

Sunway TaihuLight



K - Computer

Tianhe – 2

Titan

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

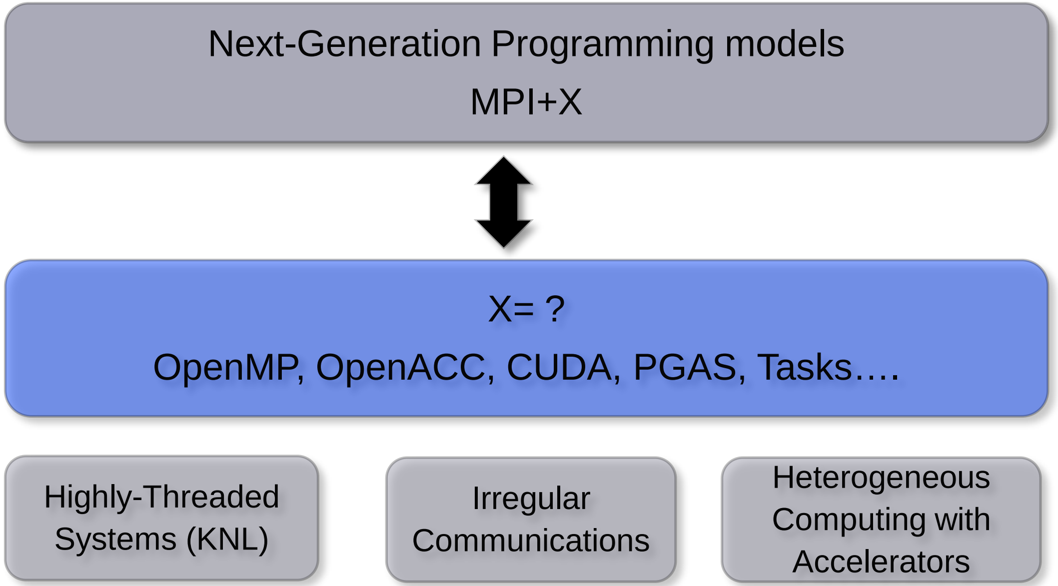
Accelerators (GPU and FPGA)

Co-Design Opportunities and Challenges across Various Layers

- Performance
- Scalability
- Fault-Resilience

Exascale Programming models

- The community believes exascale programming model will be MPI+X
- But what is X?
  - Can it be just OpenMP?
- Many different environments and systems are emerging
  - Different `X' will satisfy the respective needs



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)

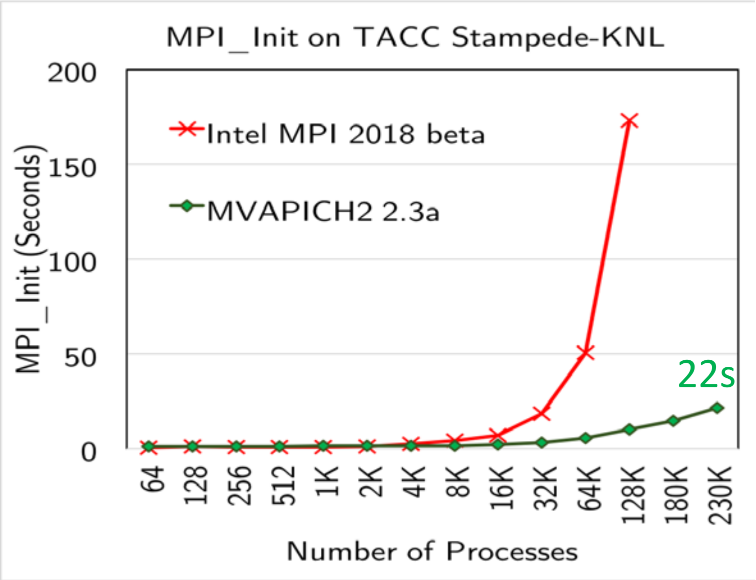
MVAPICH2 Software Family

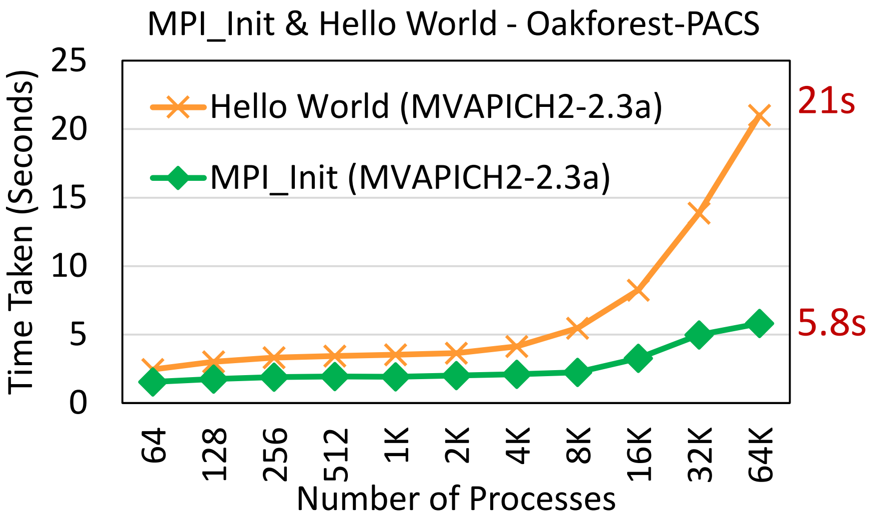
High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters 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

Outline

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication
  - Scalable Start-up
  - Dynamic and Adaptive Communication Protocols and Tag Matching
  - Optimized Collectives using SHArP and Multi-Leaders
  - Optimized CMA-based Collectives
- Hybrid MPI+PGAS Models for Irregular Applications
- Heterogeneous Computing with Accelerators
- HPC and Cloud

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

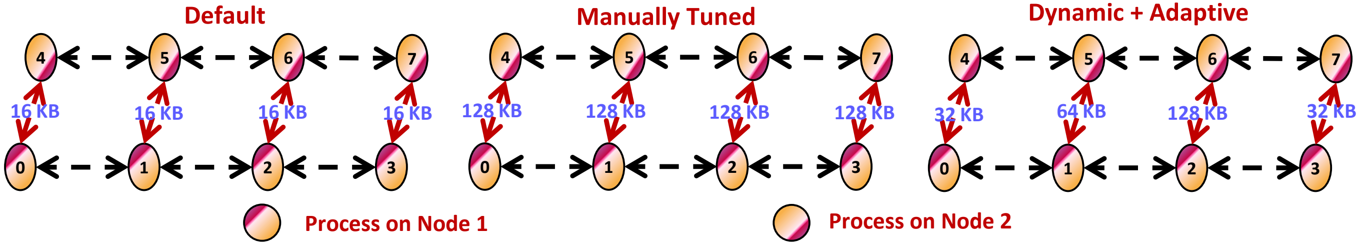
New designs available in latest MVAPICH2 libraries and as patch for SLURM-15.08.8 and SLURM-16.05.1

Dynamic and Adaptive MPI Point-to-point Communication Protocols

Desired Eager Threshold

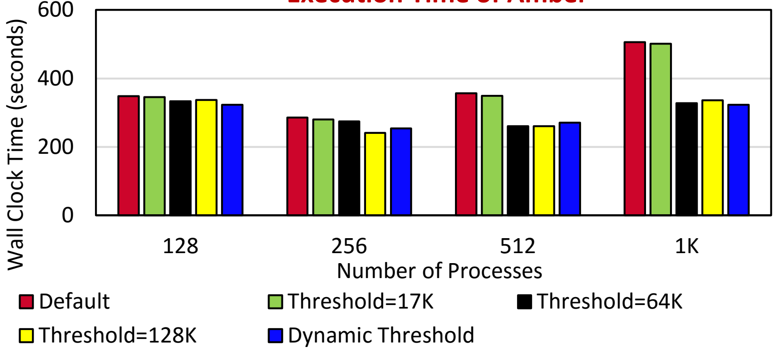
Process Pair	Eager Threshold (KB)
0 – 4	32
1 – 5	64
2 – 6	128
3 – 7	32

Eager Threshold for Example Communication Pattern with Different Designs

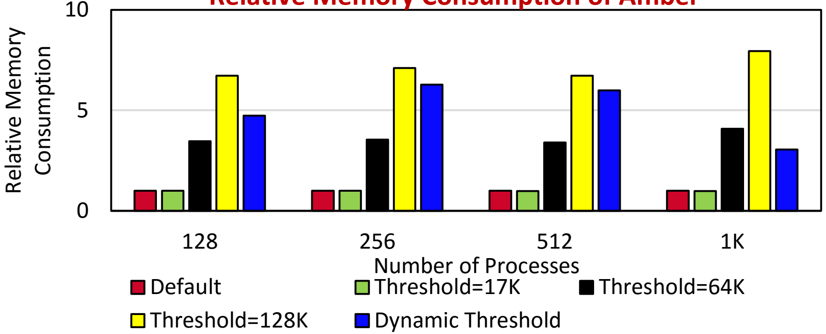


Default	Poor overlap; Low memory requirement	Low Performance; High Productivity
Manually Tuned	Good overlap; High memory requirement	High Performance; Low Productivity
Dynamic + Adaptive	Good overlap; Optimal memory requirement	High Performance; High Productivity

Execution Time of Amber



Relative Memory Consumption of Amber



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

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

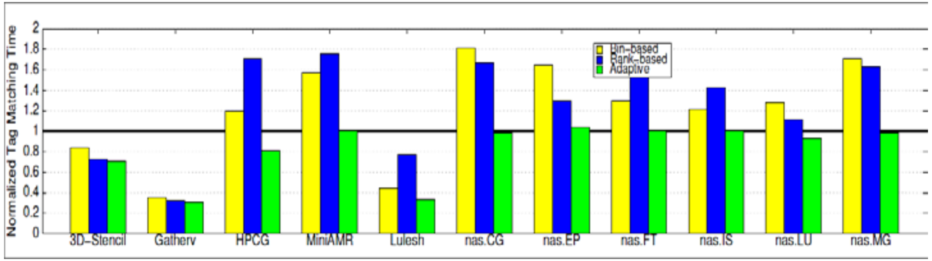
- Decisions are based on the number of request object that must be traversed before hitting on the required one

Results

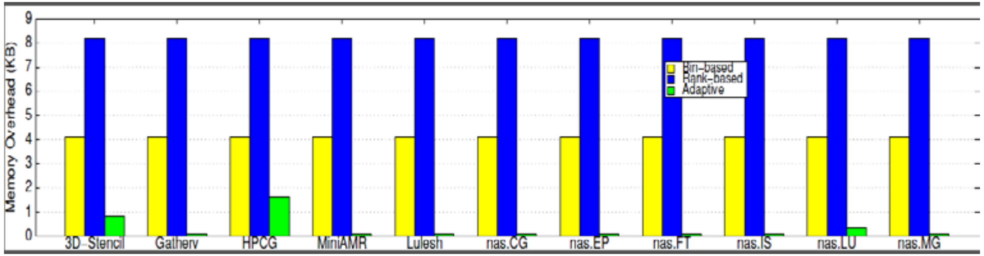
Better performance than other state-of-the art tag-matching schemes

Minimum memory consumption

Will be available in future MVAPICH2 releases



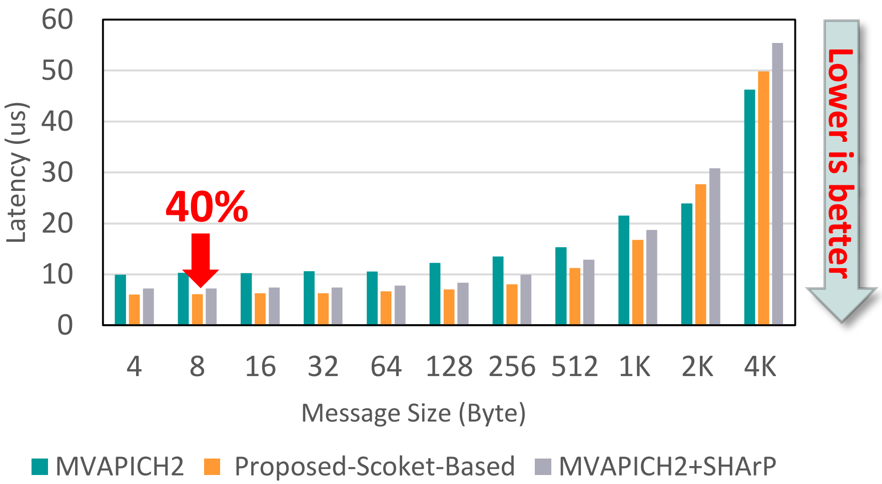
Normalized Total Tag Matching Time at 512 Processes
Normalized to Default (Lower is Better)



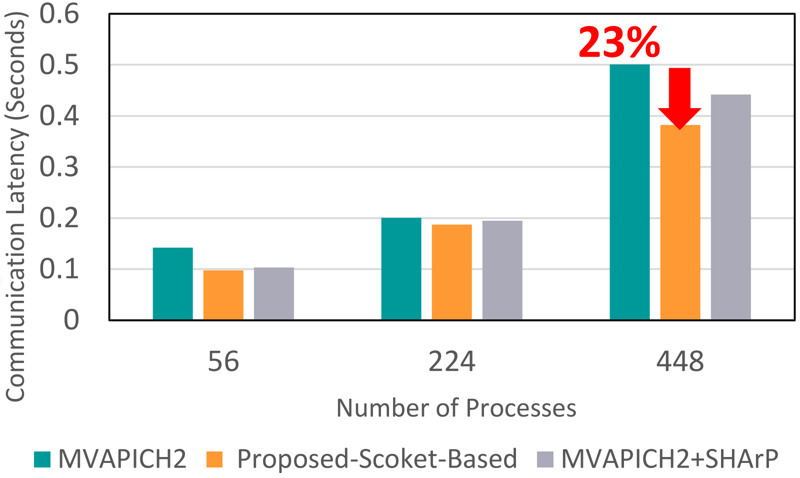
Normalized Memory Overhead per Process at 512 Processes
Compared to Default (Lower is Better)

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

Advanced Allreduce Collective Designs Using SHArP and Multi-Leaders



OSU Micro Benchmark (16 Nodes, 28 PPN)

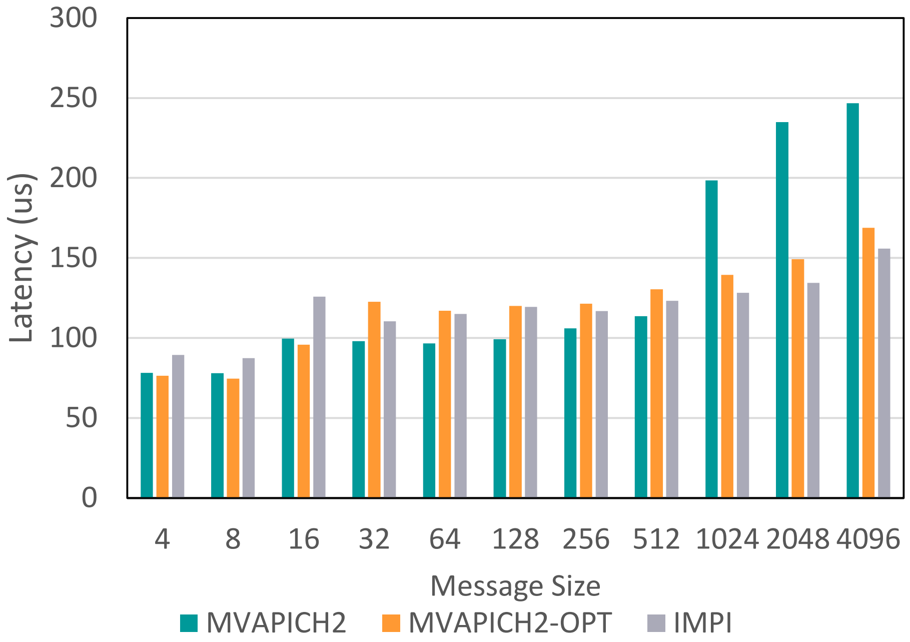


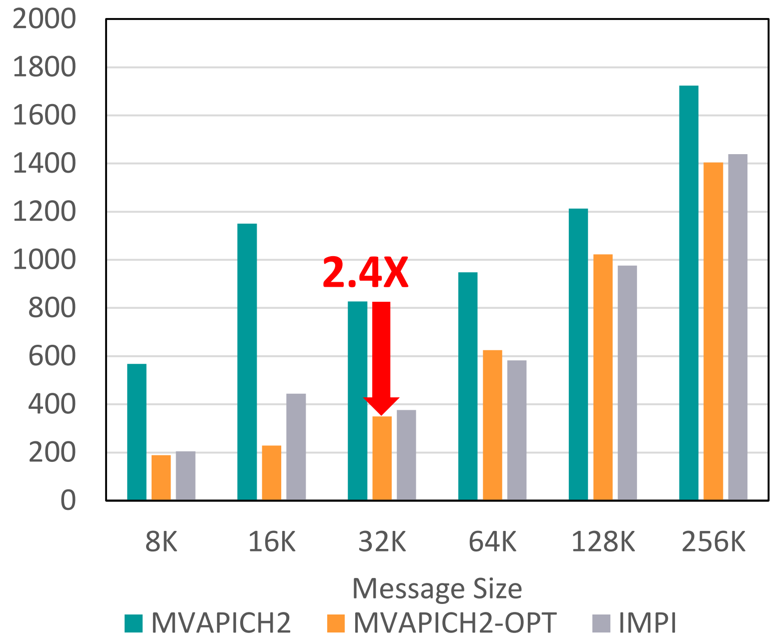
HPCG (28 PPN)

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

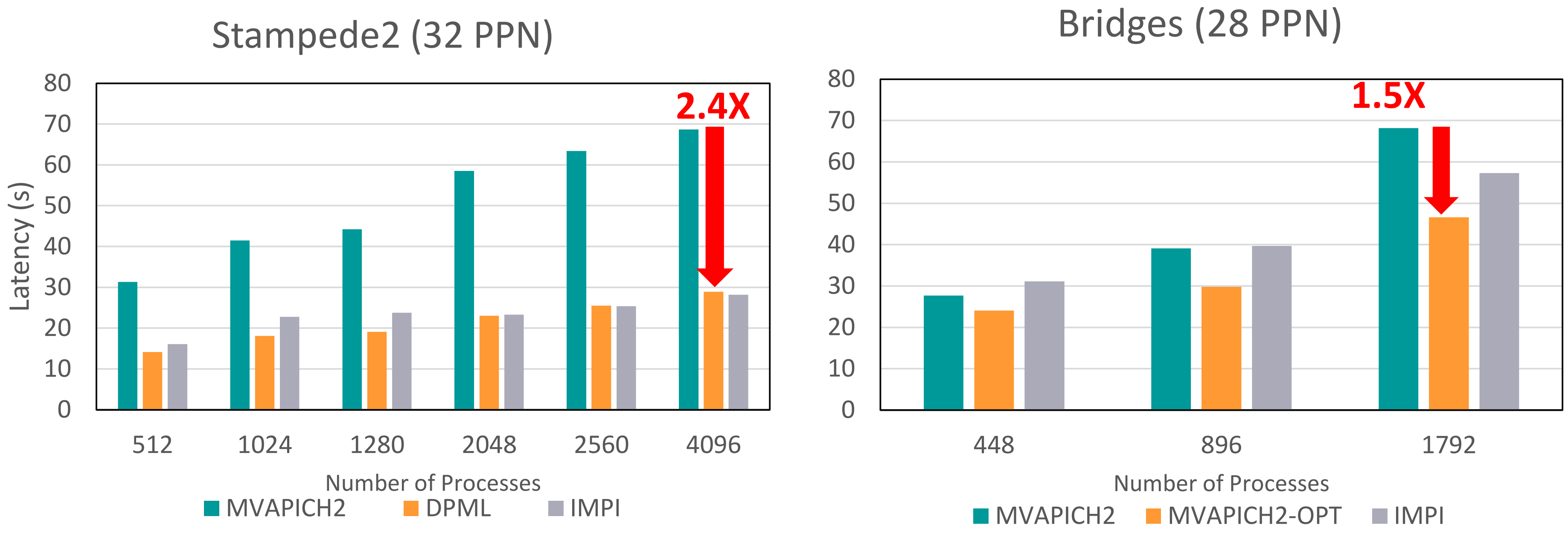




OSU Micro Benchmark 64 PPN

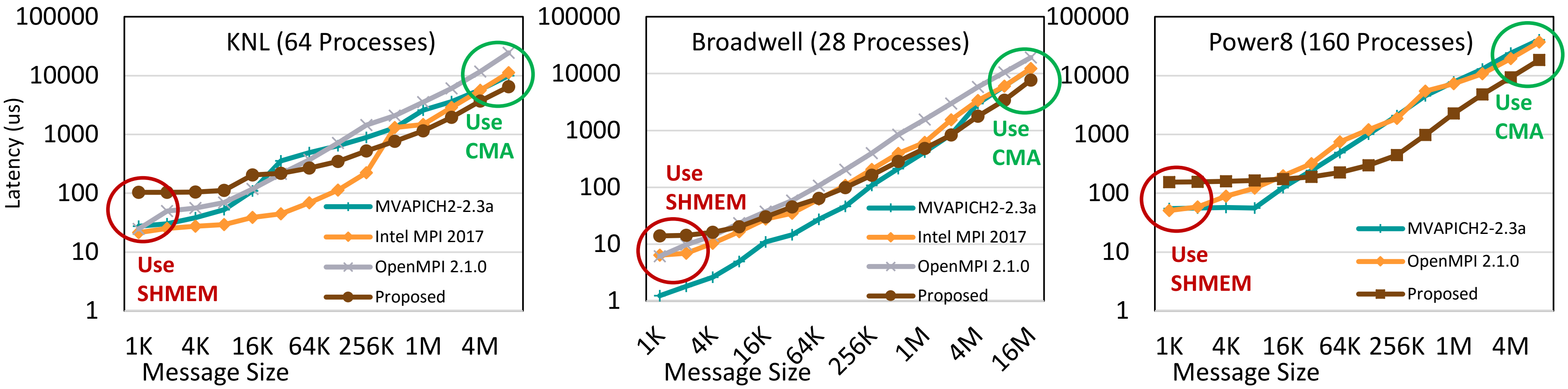
- MPI_Allreduce latency with 32K bytes reduced by 2.4X

Performance of MiniAMR Application On Stampede2 and Bridges



- For MiniAMR Application latency with 2,048 processes, MVAPICH2-OPT can reduce the latency by 2.6X on Stampede2
- On Bridges, with 1,792 processes, MVAPICH2-OPT can reduce the latency by 1.5X

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

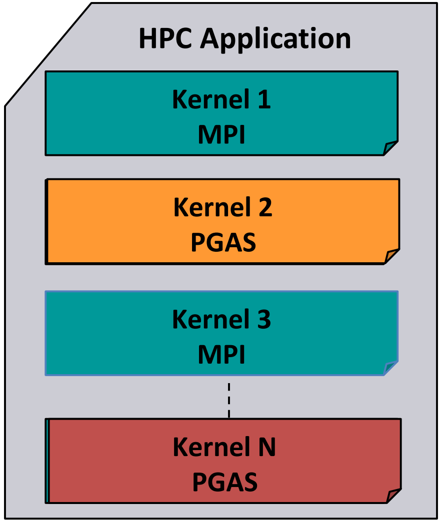
S. Chakraborty, H. Subramoni, and D. K. Panda, **Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems,** ***IEEE Cluster '17, BEST Paper Finalist***

Outline

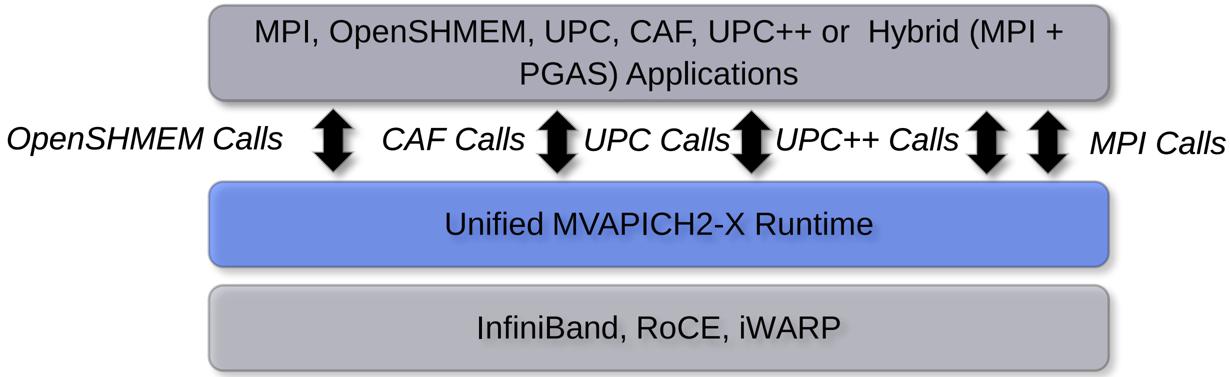
- Scalability for million to billion processors
- Hybrid MPI+PGAS Models for Irregular Applications
- Heterogeneous Computing with Accelerators
- HPC and Cloud

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
- Cons
  - Two different runtimes
  - Need great care while programming
  - Prone to deadlock if not careful

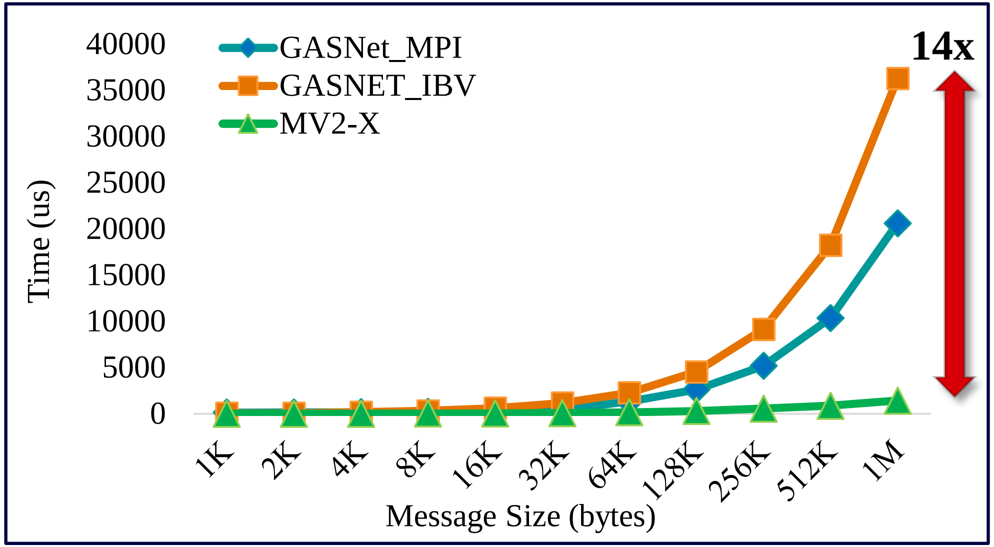


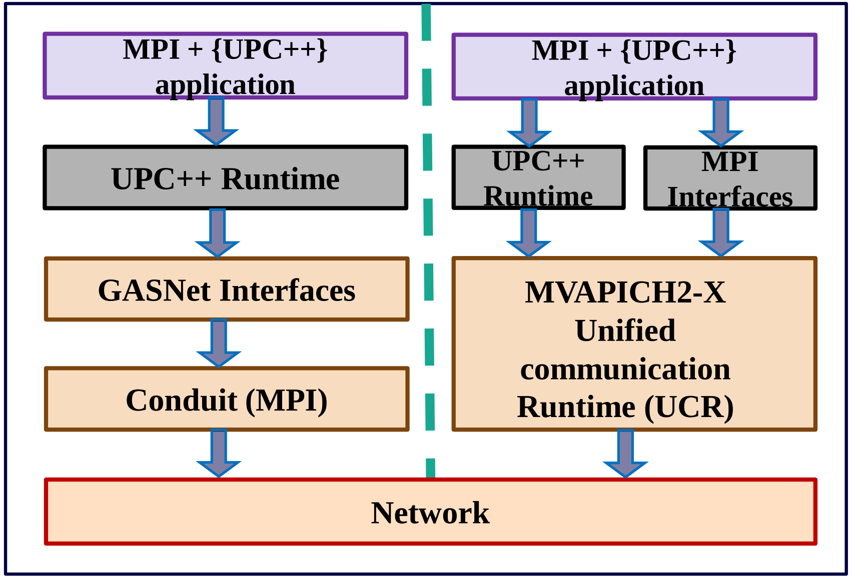
MVAPICH2-X for Hybrid MPI + PGAS Applications



- Unified communication runtime for MPI, UPC, OpenSHMEM, CAF, UPC++ available with MVAPICH2-X 1.9 onwards! (since 2012)
  - http://mvapich.cse.ohio-state.edu
- 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

UPC++ Collectives Performance



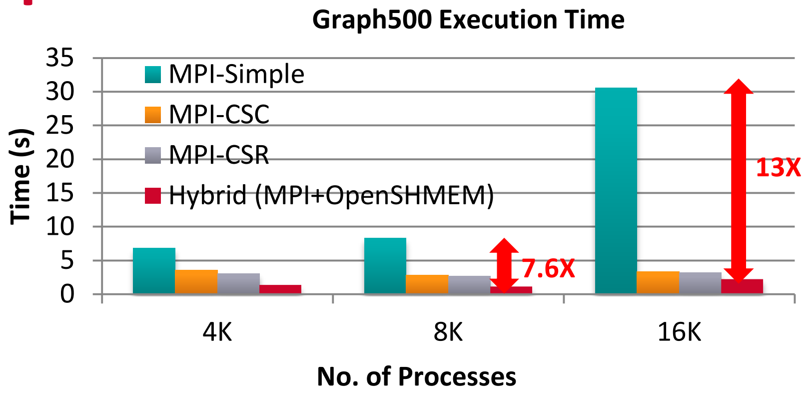


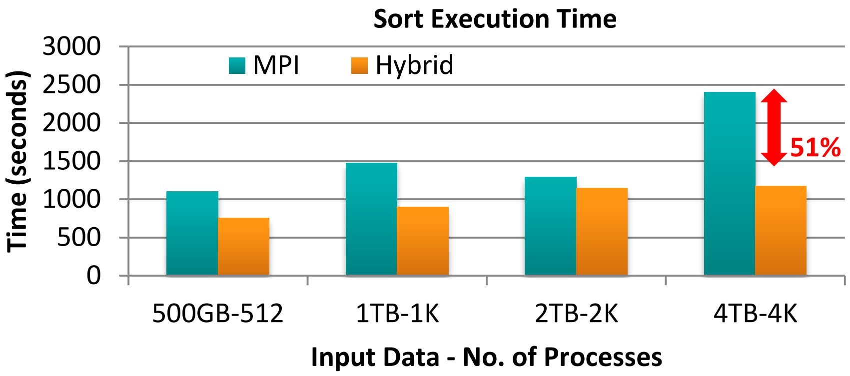
Inter-node Broadcast (64 nodes 1:ppn)

- Full and native support for hybrid MPI + UPC++ applications
- Better performance compared to IBV and MPI conduits
- OSU Micro-benchmarks (OMB) support for UPC++
- Available since MVAPICH2-X 2.2RC1

J. M. Hashmi, K. Hamidouche, and D. K. Panda, Enabling Performance Efficient Runtime Support for hybrid MPI+UPC++ Programming Models, IEEE International Conference on High Performance Computing and Communications (HPCC 2016)

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

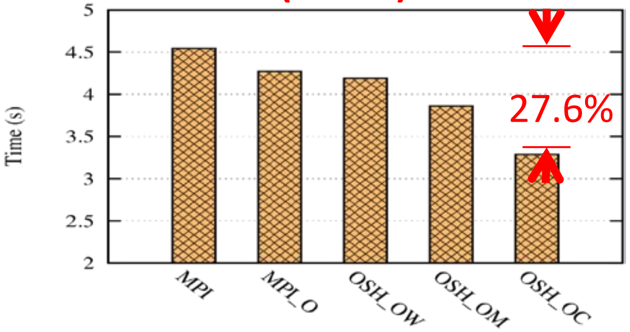
J. Jose, S. Potluri, H. Subramoni, X. Lu, K. Hamidouche, K. Schulz, H. Sundar and D. Panda Designing Scalable Out-of-core Sorting with Hybrid MPI+PGAS Programming Models, PGAS'14

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

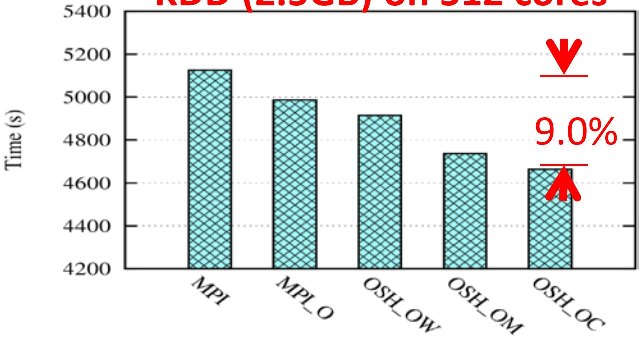
Accelerating MaTEx k-NN with Hybrid MPI and OpenSHMEM

- **MaTEx:** MPI-based Machine learning algorithm library
- **k-NN:** a popular supervised algorithm for classification
- **Hybrid designs:**
  - Overlapped Data Flow; One-sided Data Transfer; Circular-buffer Structure

KDD-tranc (30MB) on 256 cores



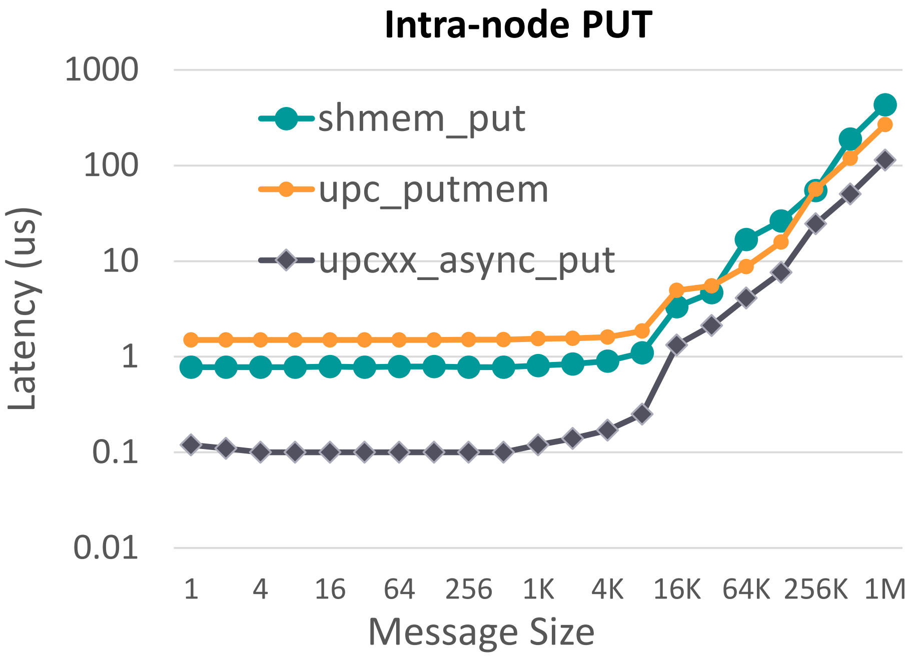
KDD (2.5GB) on 512 cores

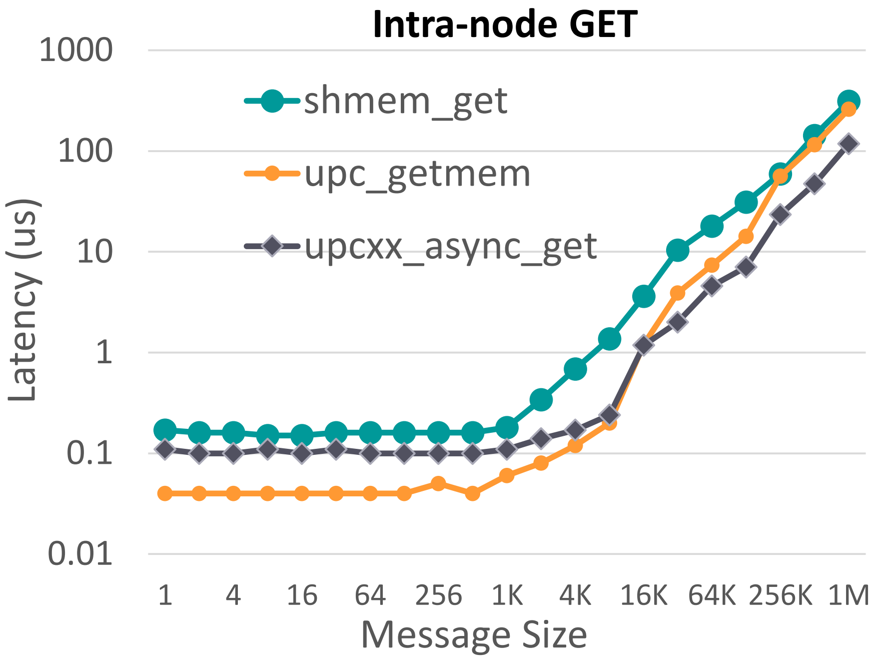


- Benchmark: KDD Cup 2010 (8,407,752 records, 2 classes, k=5)
- For truncated KDD workload on 256 cores, reduce 27.6% execution time
- For full KDD workload on 512 cores, reduce 9.0% execution time

J. Lin, K. Hamidouche, J. Zhang, X. Lu, A. Vishnu, D. Panda. Accelerating k-NN Algorithm with Hybrid MPI and OpenSHMEM, OpenSHMEM 2015

Performance of PGAS Models on KNL using MVAPICH2-X

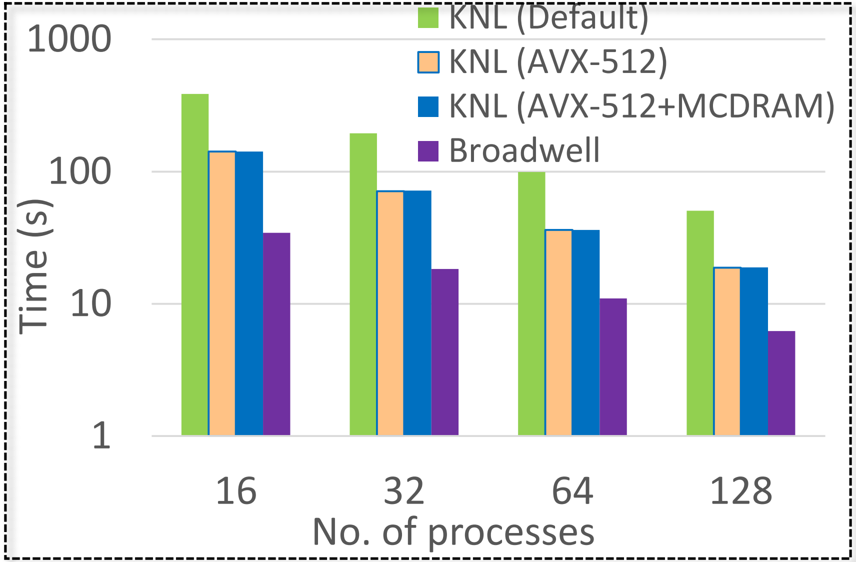




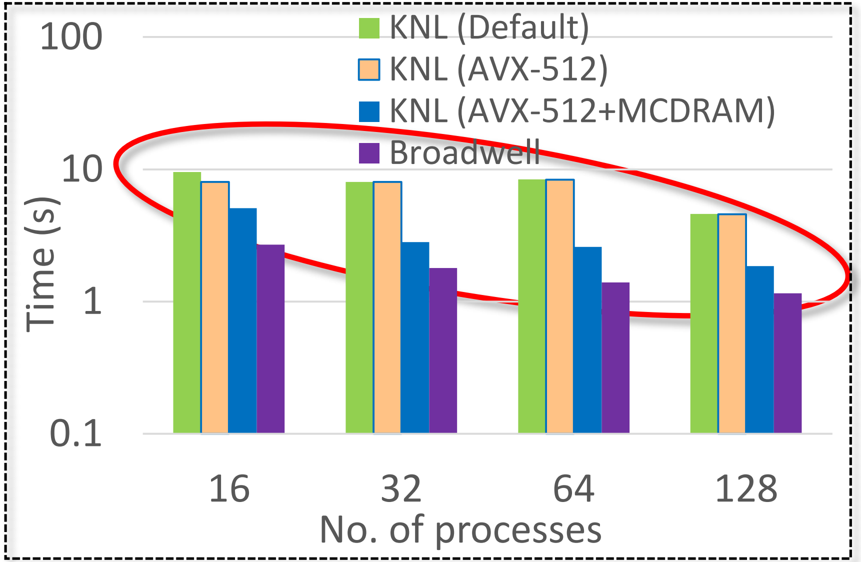
- Intra-node performance of one-sided Put/Get operations of PGAS libraries/languages using MVAPICH2-X communication conduit
- Near-native communication performance is observed on KNL

Optimized OpenSHMEM with AVX and MCDRAM: Application Kernels Evaluation

Heat-2D Kernel using Jacobi method



Heat Image Kernel



- On heat diffusion based kernels AVX-512 vectorization showed better performance
- MCDRAM showed significant benefits on Heat-Image kernel for all process counts. Combined with AVX-512 vectorization, it showed up to 4X improved performance

Outline

- Scalability for million to billion processors
- Hybrid MPI+PGAS Models for Irregular Applications
- Heterogeneous Computing with Accelerators
- HPC and Cloud

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

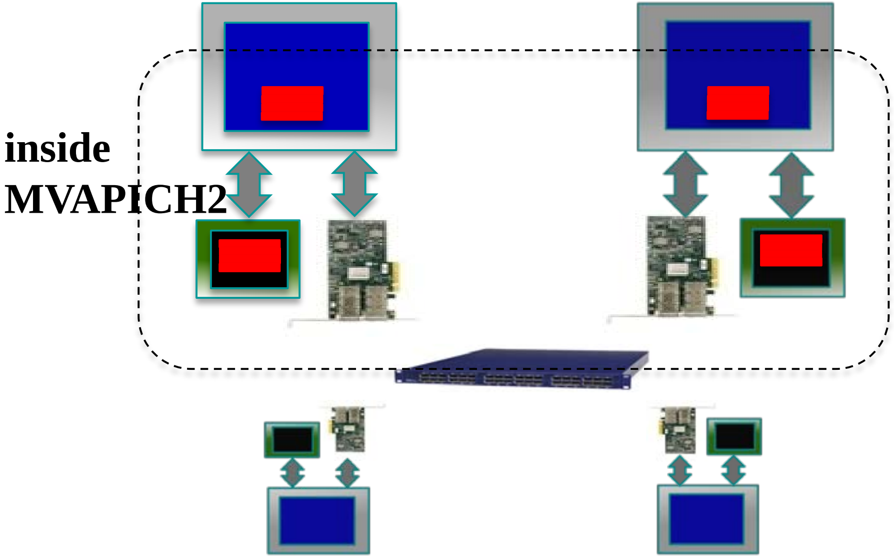
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

MPI_Send(s_devbuf, size, …);

At Receiver:

MPI_Recv(r_devbuf, size, …);

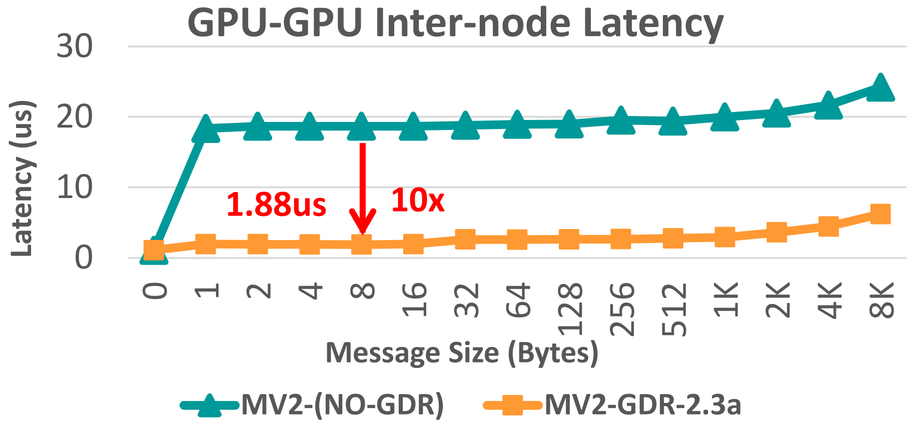


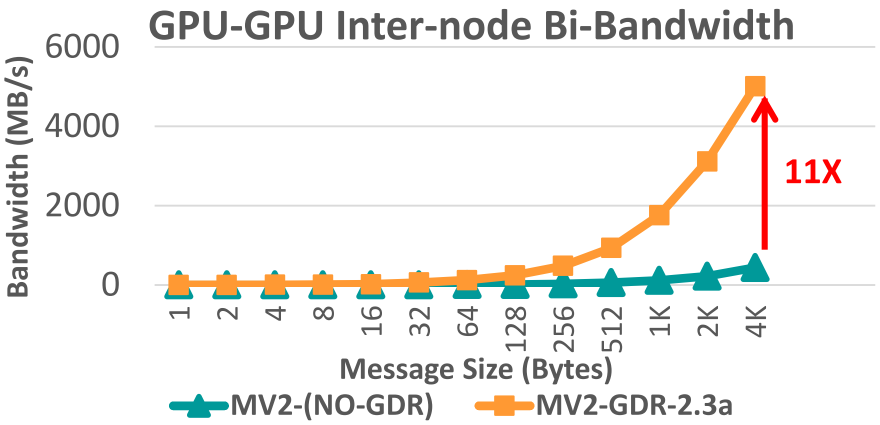
High Performance and High Productivity

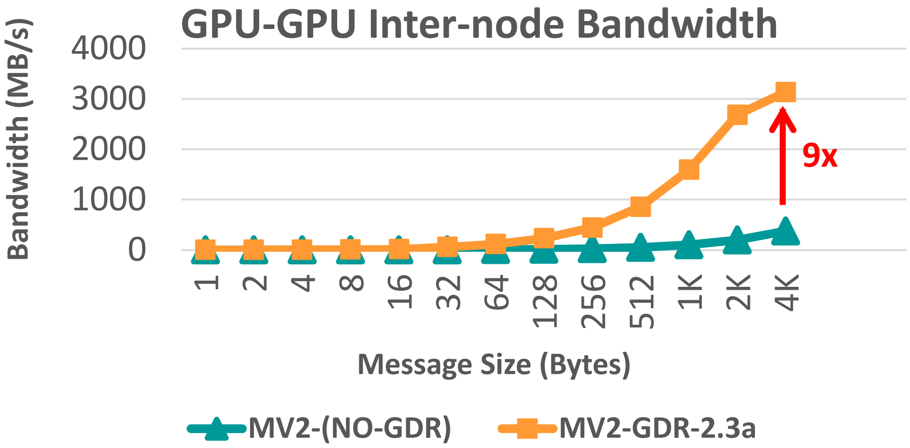
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

- 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
- Unified memory

Optimized MVAPICH2-GDR Design

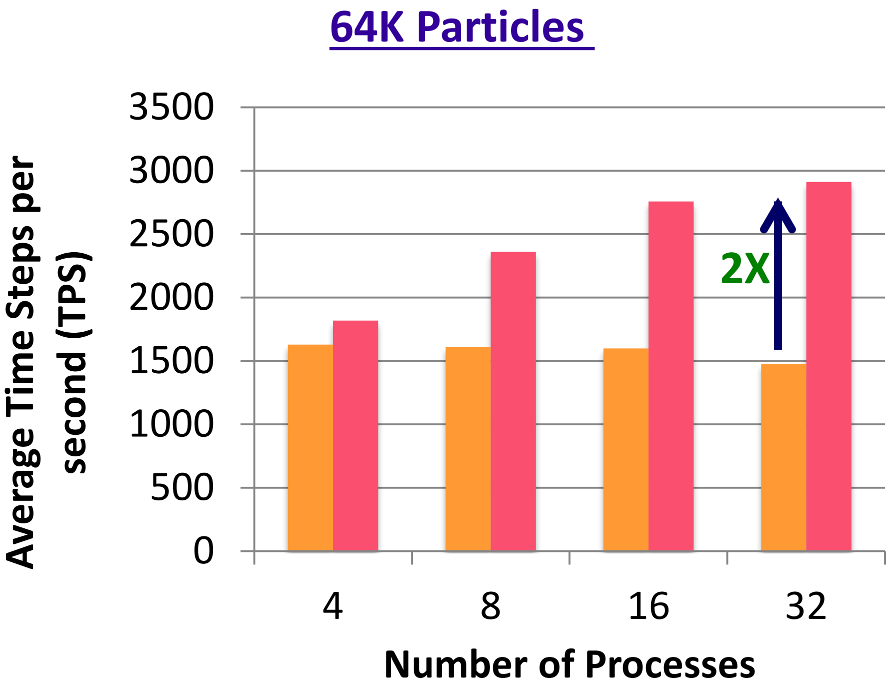


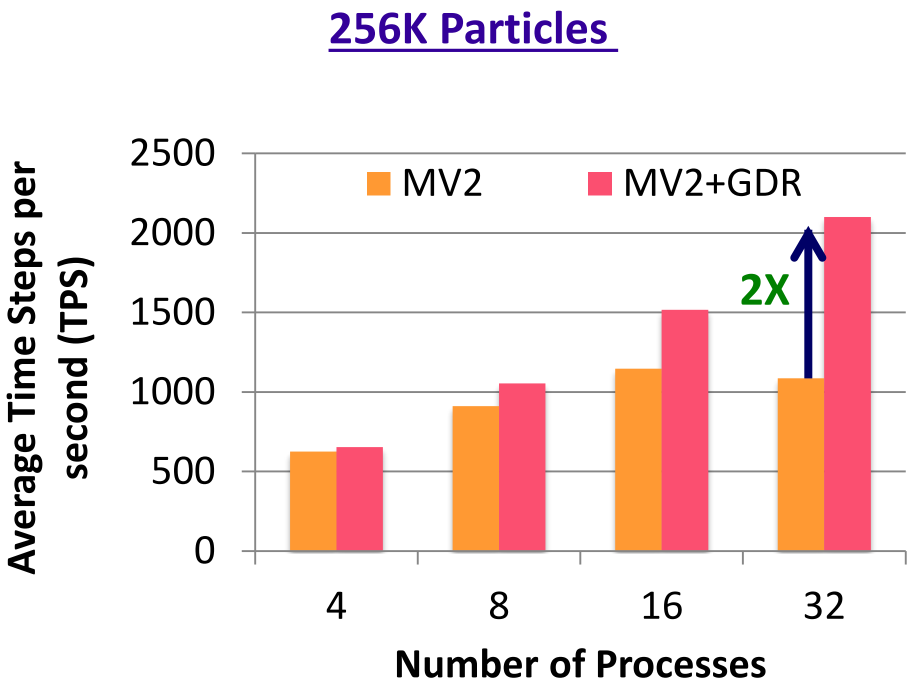




MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

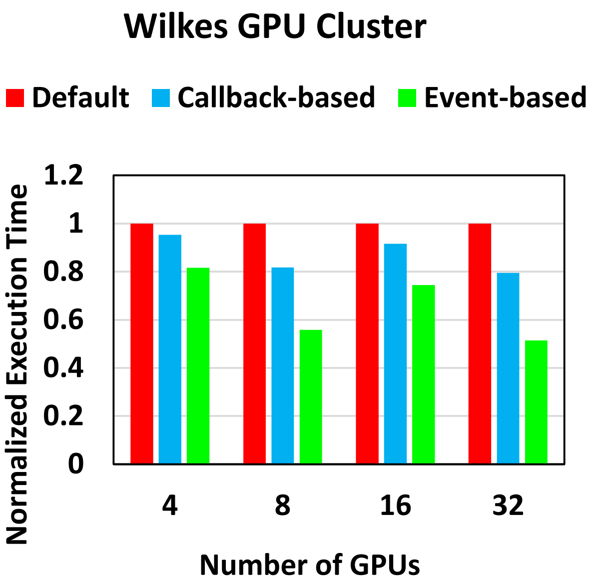
Application-Level Evaluation (HOOMD-blue)

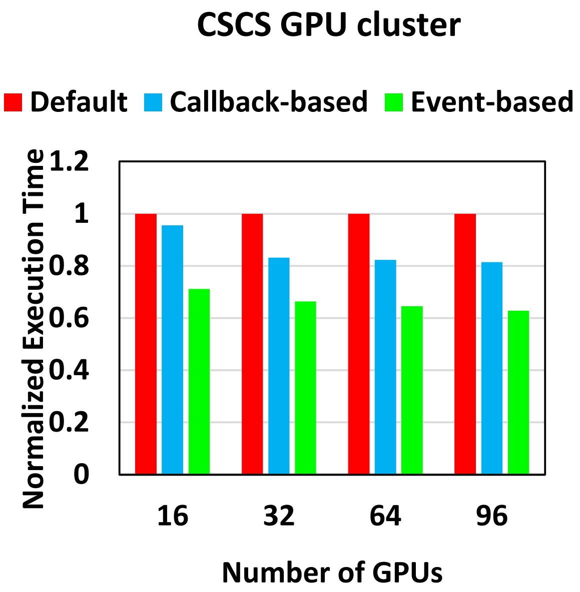


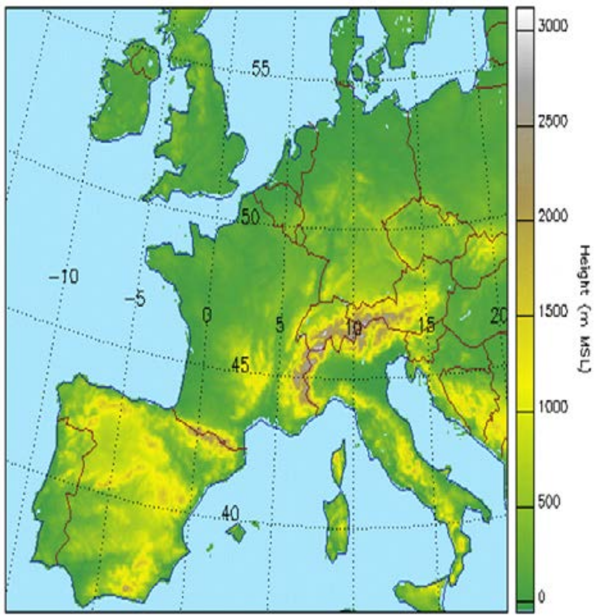


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland







- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

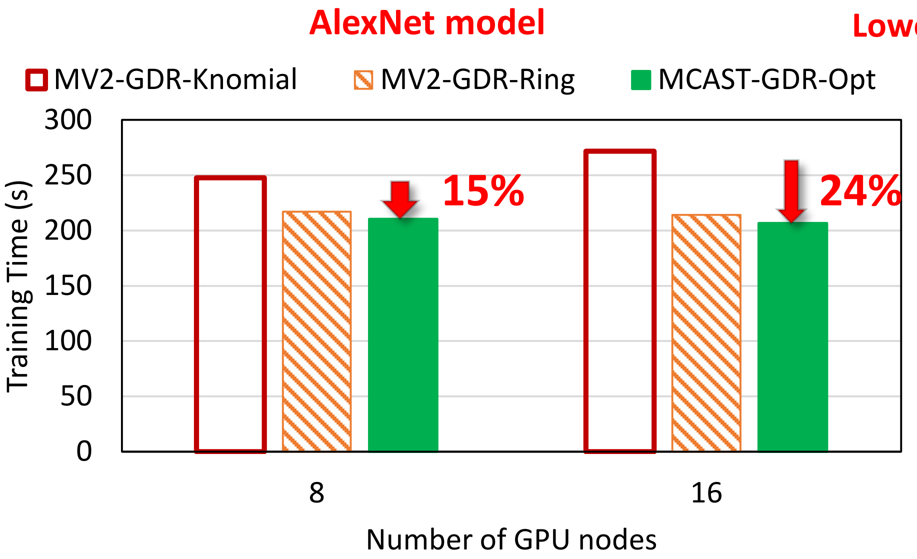
On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application

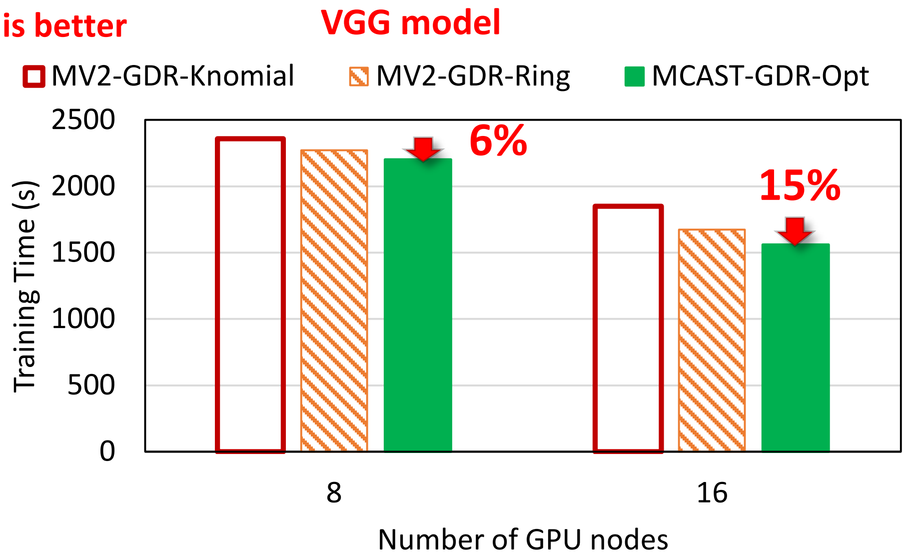
C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

Application Evaluation: Deep Learning Frameworks

- @ RI2 cluster, 16 GPUs, 1 GPU/node
  - Microsoft Cognitive Toolkit (CNTK) *[https://github.com/Microsoft/CNTK]*

Lower is better

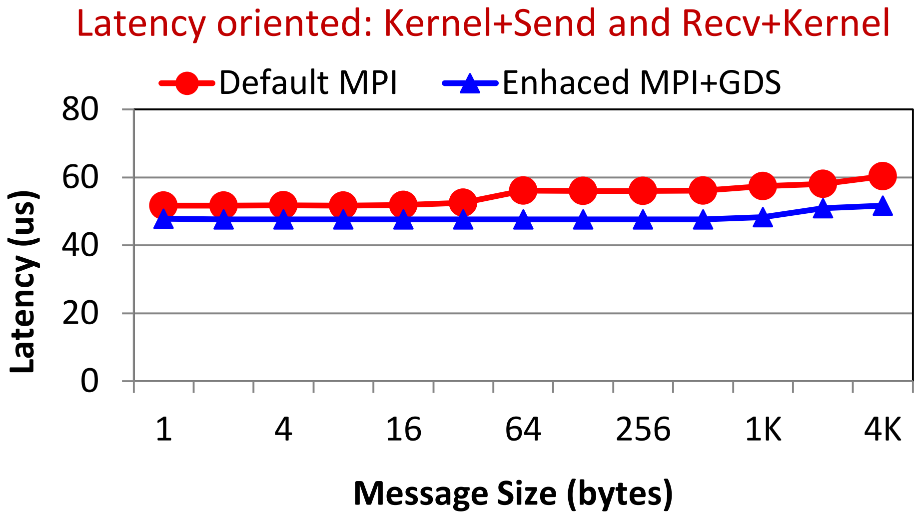


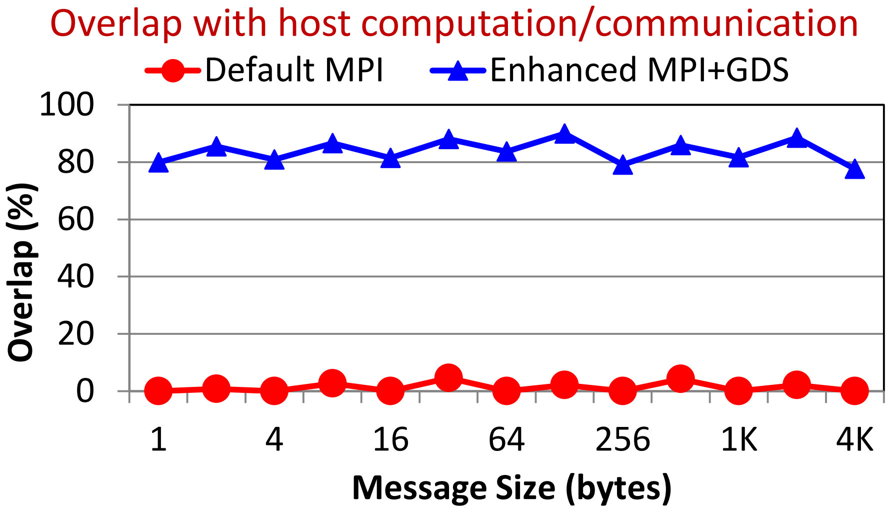


- Reduces up to 24% and 15% of latency for AlexNet and VGG models
- Higher improvement can be observed for larger system sizes

C.-H. Chu, X. Lu, A. A. Awan, H. Subramoni, J. Hashmi, B. Elton, and D. K. Panda, Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning, ICPP'17.

MVAPICH2-GDS: Preliminary Results





- Latency Oriented: Able to hide the kernel launch overhead
  - 8-15% improvement compared to default behavior
- Overlap: Asynchronously to offload queue the Communication and computation tasks
  - 89% overlap with host computation at 128-Byte message size

Intel Sandy Bridge, NVIDIA Tesla K40c and Mellanox FDR HCA
CUDA 8.0, OFED 3.4, Each kernel is ~50us

Will be available in a public release soon

Outline

- Scalability for million to billion processors
- Hybrid MPI+PGAS Models for Irregular Applications
- Heterogeneous Computing with Accelerators
- HPC and Cloud

Can HPC and Virtualization be Combined?

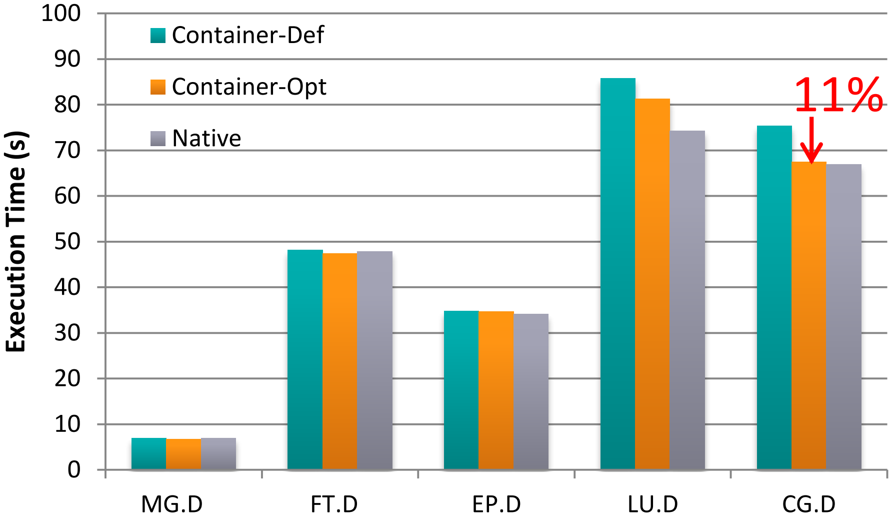
- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2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

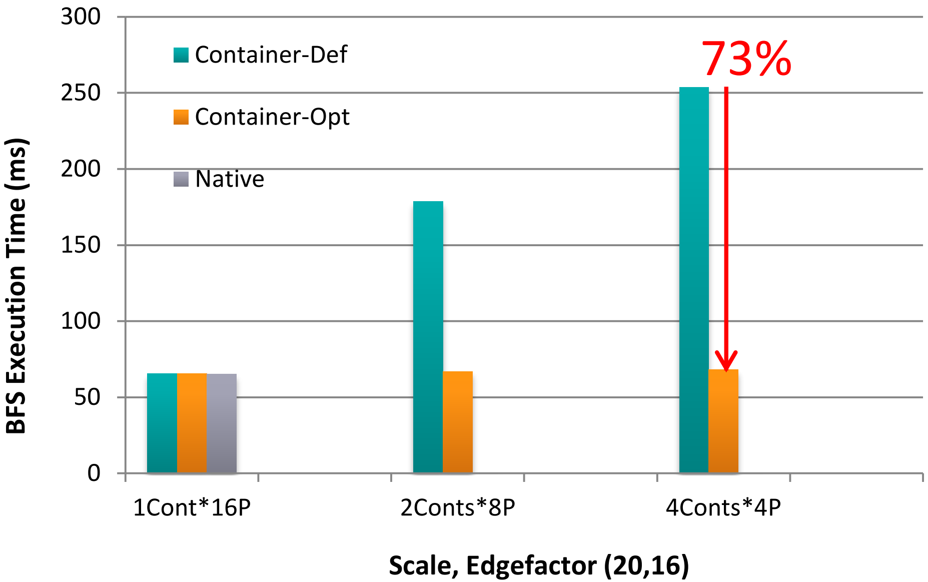
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC'14

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 Docker with MVAPICH2



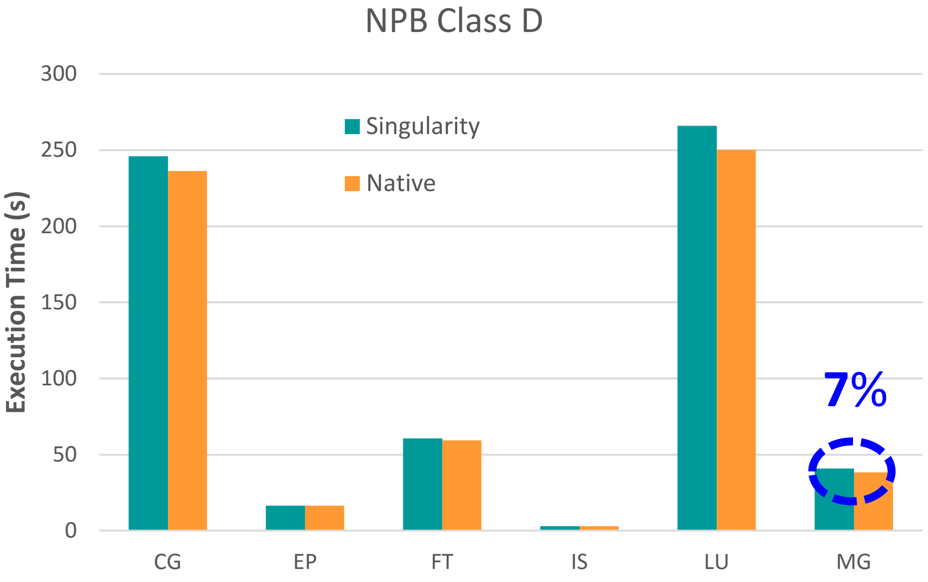
NAS

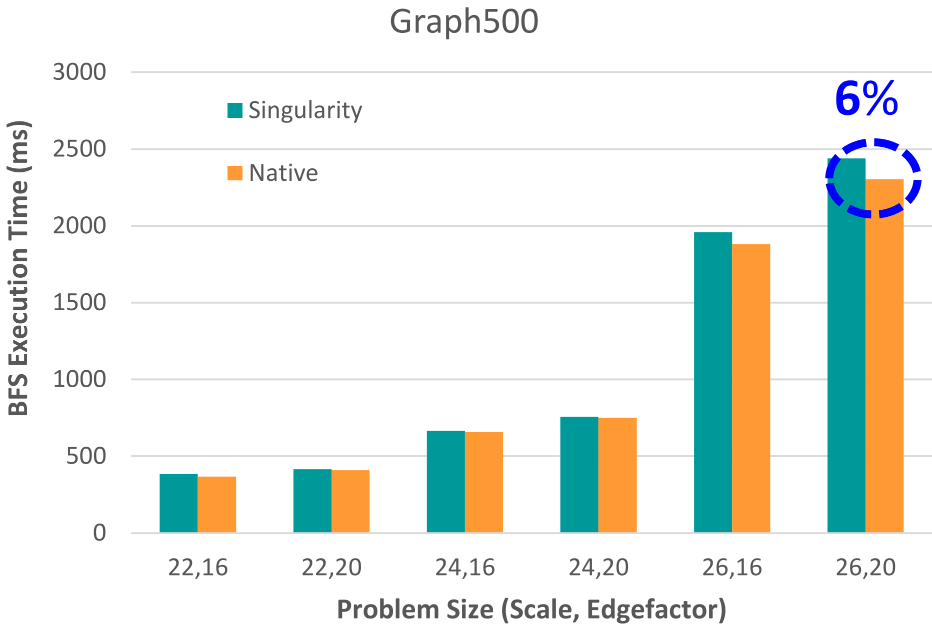


Graph 500

- 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

Looking into the Future ….

- Architectures for Exascale systems are evolving
- Exascale systems will be constrained by
  - Power
  - Memory per core
  - Data movement cost
  - Faults
- Programming Models, Runtimes and Middleware need to be designed for
  - Scalability
  - Performance
  - Fault-resilience
  - Energy-awareness
  - Programmability
  - Productivity
- High Performance and Scalable MPI+X libraries are needed
- Highlighted some of the approaches taken by the MVAPICH2 project
- Need continuous innovation to have the right MPI+X libraries for Exascale systems

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students

- A. Awan (Ph.D.)
- M. Bayatpour (Ph.D.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- S. Guganani (Ph.D.)
- J. Hashmi (Ph.D.)
- N. Islam (Ph.D.)
- M. Li (Ph.D.)
- M. Rahman (Ph.D.)
- D. Shankar (Ph.D.)
- A. Venkatesh (Ph.D.)
- J. Zhang (Ph.D.)

Current Research Scientists

- X. Lu
- H. Subramoni

Current Post-doc

- A. Ruhela

Current Research Specialist

- J. Smith
- M. Arnold

Past Students

- A. Augustine (M.S.)
- 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.)
- K. Kulkarni (M.S.)
- 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 Research Scientist

- K. Hamidouche
- S. Sur

Past Programmers

- D. Bureddy
- J. Perkins

Past Post-Docs

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

Thank You!

panda@cse.ohio-state.edu

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

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

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

Please join us for other events at SC'17

- Workshops
  - ESPM2 2017: Third International Workshop on Extreme Scale Programming Models and Middleware
- Tutorials
  - InfiniBand, Omni-Path, and High-Speed Ethernet for Dummies
  - InfiniBand, Omni-Path, and High-Speed Ethernet: Advanced Features, Challenges in Designing, HEC Systems and Usage
- BoFs
  - MPICH BoF: MVAPICH2 Project: Latest Status and Future Plans
- Technical Talks
  - EReinit: Scalable and Efficient Fault-Tolerance for Bulk-Synchronous MPI Applications
  - An In-Depth Performance Characterization of CPU- and GPU-Based DNN Training on Modern Architectures
  - Scalable Reduction Collectives with Data Partitioning-Based Multi-Leader Design
- Technical Talks
  - Designing and Building Efficient HPC Cloud with Modern Networking Technologies on Heterogeneous HPC Clusters
  - Co-designing MPI Runtimes and Deep Learning Frameworks for Scalable Distributed Training on GPU Clusters
  - High-Performance and Scalable Broadcast Schemes for Deep Learning on GPU Clusters
- Booth Talks
  - Scalability and Performance of MVAPICH2 on OakForest-PACS
  - The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing
  - Performance of PGAS Models on KNL: A Comprehensive Study with MVAPICH2-X
  - Exploiting Latest Networking and Accelerator Technologies for MPI, Streaming, and Deep Learning: An MVAPICH2-Based Approach
  - MVAPICH2-GDR Library: Pushing the Frontier of HPC and Deep Learning
  - MVAPICH2-GDR for HPC and Deep Learning

Please refer to http://mvapich.cse.ohio-state.edu/talks/ for more details