Designing Software Libraries and Middleware for Exascale Systems: Opportunities and Challenges

Talk at Brookhaven National Laboratory (Oct 2014)

by

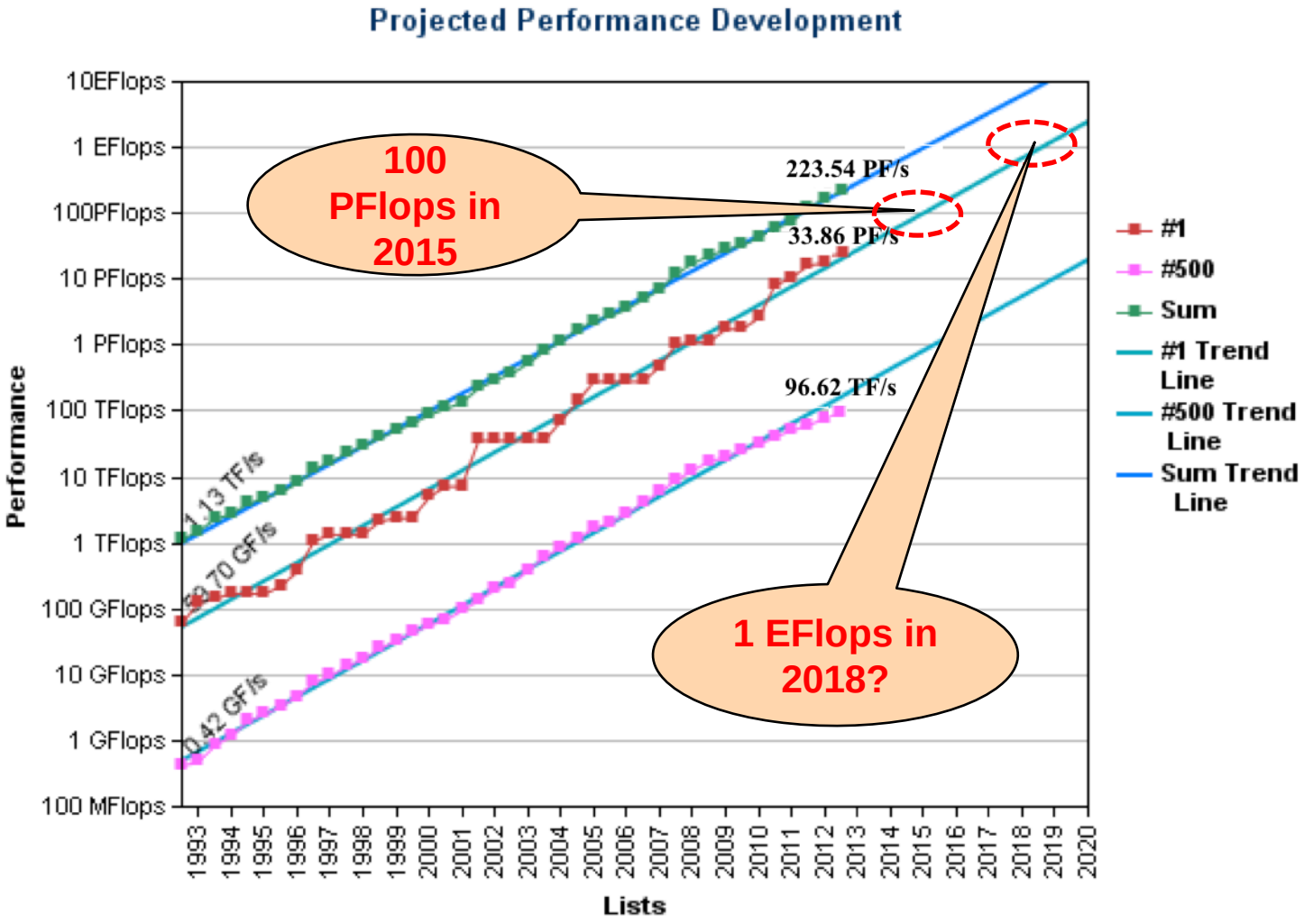
Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

High-End Computing (HEC): PetaFlop to ExaFlop

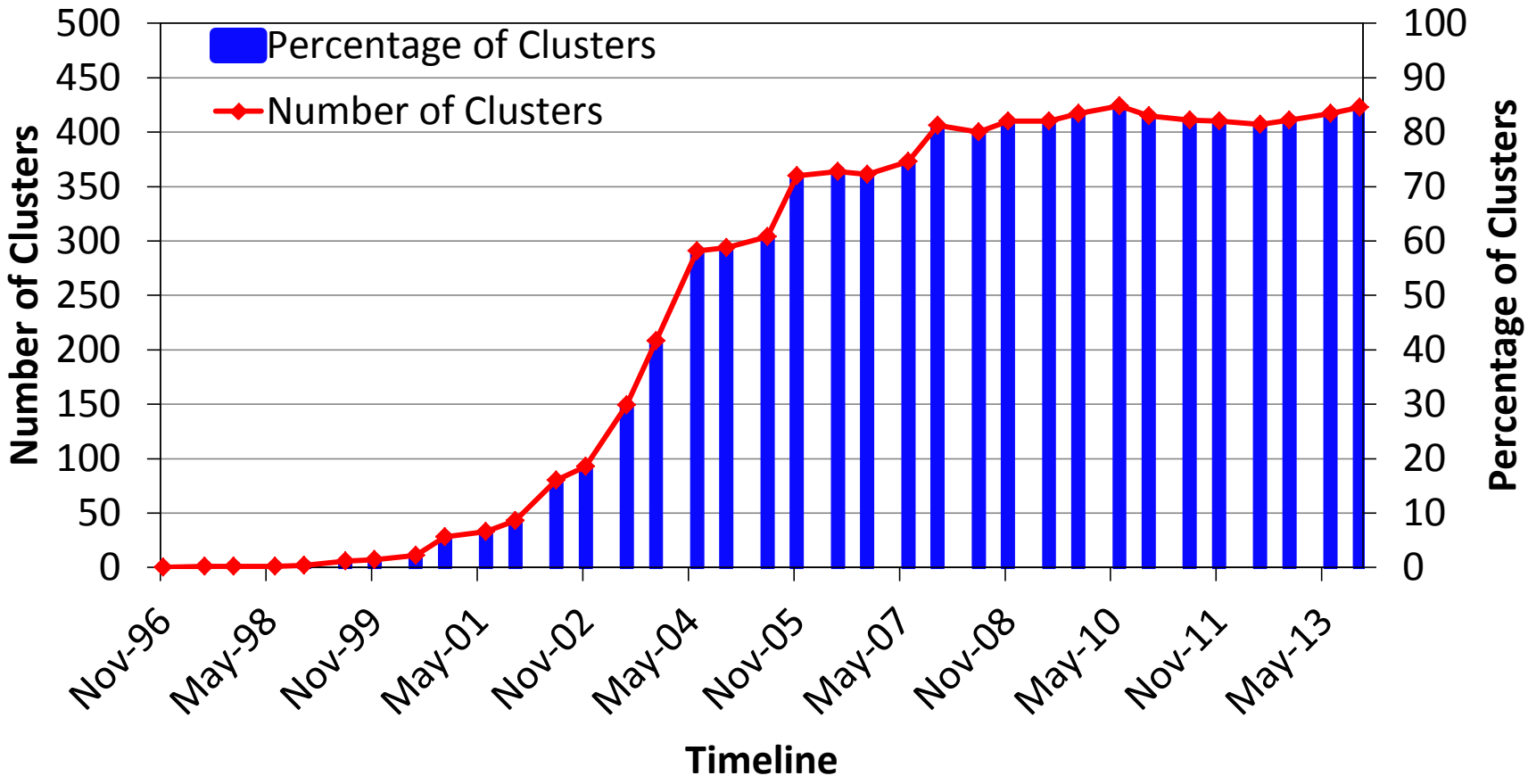


Expected to have an ExaFlop system in 2020-2024!

Two Major Categories of Applications

- Scientific Computing
  - Message Passing Interface (MPI), including MPI + OpenMP, is the Dominant Programming Model
  - Many discussions towards Partitioned Global Address Space (PGAS)
    - UPC, OpenSHMEM, CAF, etc.
  - Hybrid Programming: MPI + PGAS (OpenSHMEM, UPC)
- Big Data/Enterprise/Commercial Computing
  - Focuses on large data and data analysis
  - Hadoop (HDFS, HBase, MapReduce)
  - Spark is emerging for in-memory computing
  - Memcached is also used for Web 2.0
- Applications can run on a single-site or across sites over WAN

Trends for Commodity Computing Clusters in the Top 500 List (http://www.top500.org)



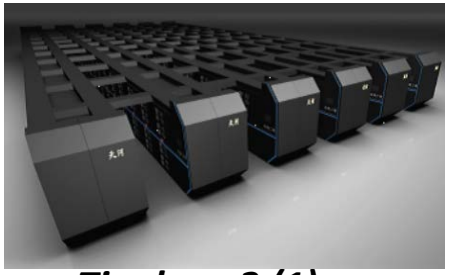
Drivers of Modern HPC Cluster Architectures

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

- Multi-core processors are ubiquitous
- InfiniBand very popular in HPC clusters
- Accelerators/Coprocessors becoming common in high-end systems
- Pushing the envelope for Exascale computing



Tianhe – 2 (1)

Titan (2)

Stampede (6)

Tianhe – 1A (10)

Large-scale InfiniBand Installations

- 223 IB Clusters (44.3%) in the June 2014 Top500 list (http://www.top500.org)
- Installations in the Top 50 (25 systems):

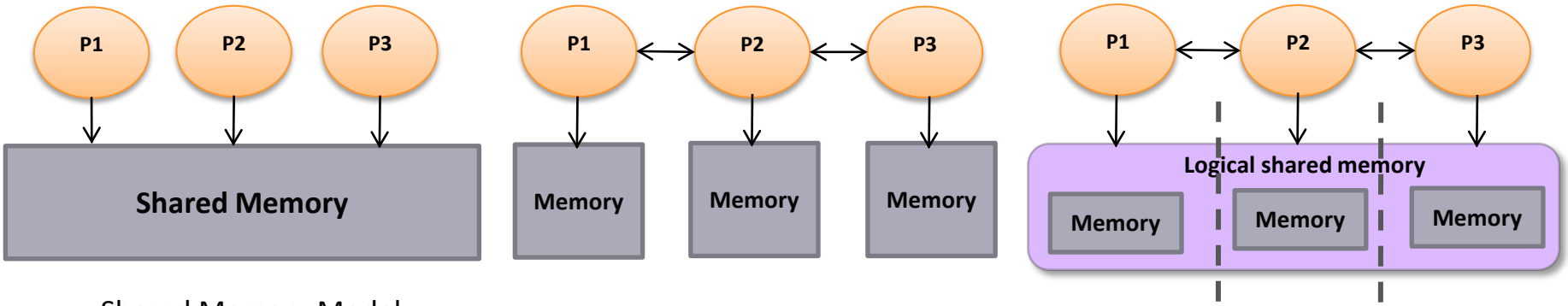
519,640 cores (Stampede) at TACC (7th)	120, 640 cores (Nebulae) at China/NSCS (28th)
62,640 cores (HPC2) in Italy (11th)	72,288 cores (Yellowstone) at NCAR (29th)
147, 456 cores (Super MUC) in Germany (12th)	70,560 cores (Helios) at Japan/IFERC (30th)
76,032 cores (Tsubame 2.5) at Japan/GSIC (13th)	138,368 cores (Tera-100) at France/CEA (35th)
194,616 cores (Cascade) at PNNL (15th)	222,072 cores (QUARTETTO) in Japan (37th)
110,400 cores (Pangea) at France/Total (16th)	53,504 cores (PRIMERGY) in Australia (38th)
96,192 cores (Pleiades) at NASA/Ames (21st)	77,520 cores (Conte) at Purdue University (39th)
73,584 cores (Spirit) at USA/Air Force (24th)	44,520 cores (Spruce A) at AWE in UK (40th)
77,184 cores (Curie thin nodes) at France/CEA (26h)	48,896 cores (MareNostrum) at Spain/BSC (41st)
65,320-cores, iDataPlex DX360M4 at Germany/Max-Planck (27th)	**and many more!**

Towards Exascale System (Today and Target)

Systems	2014 Tianhe-2	2020-2022	Difference Today & Exascale
System peak	55 PFlop/s	1 EFlop/s	~20x
Power	18 MW (3 Gflops/W)	~20 MW (50 Gflops/W)	O(1) ~15x
System memory	1.4 PB (1.024PB CPU + 0.384PB CoP)	32 – 64 PB	~50X
Node performance	3.43TF/s (0.4 CPU + 3 CoP)	1.2 or 15 TF	O(1)
Node concurrency	24 core CPU + 171 cores CoP	O(1k) or O(10k)	~5x - ~50x
Total node interconnect BW	6.36 GB/s	200 – 400 GB/s	~40x -~60x
System size (nodes)	16,000	O(100,000) or O(1M)	~6x - ~60x
Total concurrency	3.12M 12.48M threads (4 /core)	O(billion) for latency hiding	~100x
MTTI	Few/day	Many/day	O(?)

Courtesy: Prof. Jack Dongarra

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.

How does MPI Plan to Meet Exascale Challenges?

- Power required for data movement operations is one of the main challenges
- Non-blocking collectives
  - Overlap computation and communication
- Much improved One-sided interface
  - Reduce synchronization of sender/receiver
- Manage concurrency
  - Improved interoperability with PGAS (e.g. UPC, Global Arrays, OpenSHMEM)
- Resiliency
  - New interface for detecting failures

Major New Features in MPI-3

- Major features
  - Non-blocking Collectives
  - Improved One-Sided (RMA) Model
  - MPI Tools Interface
- Specification is available from: http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf

Partitioned Global Address Space (PGAS) Models

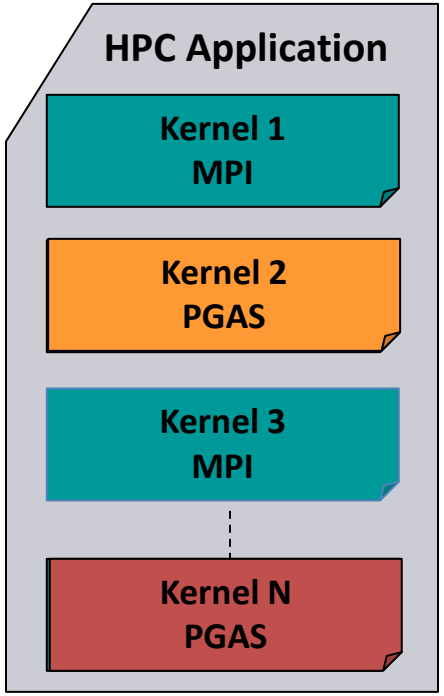
- Key features
  - Simple shared memory abstractions
  - Light weight one-sided communication
  - Easier to express irregular communication
- Different approaches to PGAS
  - Languages
    - Unified Parallel C (UPC)
    - Co-Array Fortran (CAF)
    - X10
  - Libraries
    - OpenSHMEM
    - Global Arrays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

MPI+PGAS for Exascale Architectures and Applications

- Hierarchical architectures with multiple address spaces
- (MPI + PGAS) Model
  - MPI across address spaces
  - PGAS within an address space
- MPI is good at moving data between address spaces
- Within an address space, MPI can interoperate with other shared memory programming models
- Can co-exist with OpenMP for offloading computation
- Applications can have kernels with different communication patterns
- Can benefit from different models
- Re-writing complete applications can be a huge effort
- Port critical kernels to the desired model instead

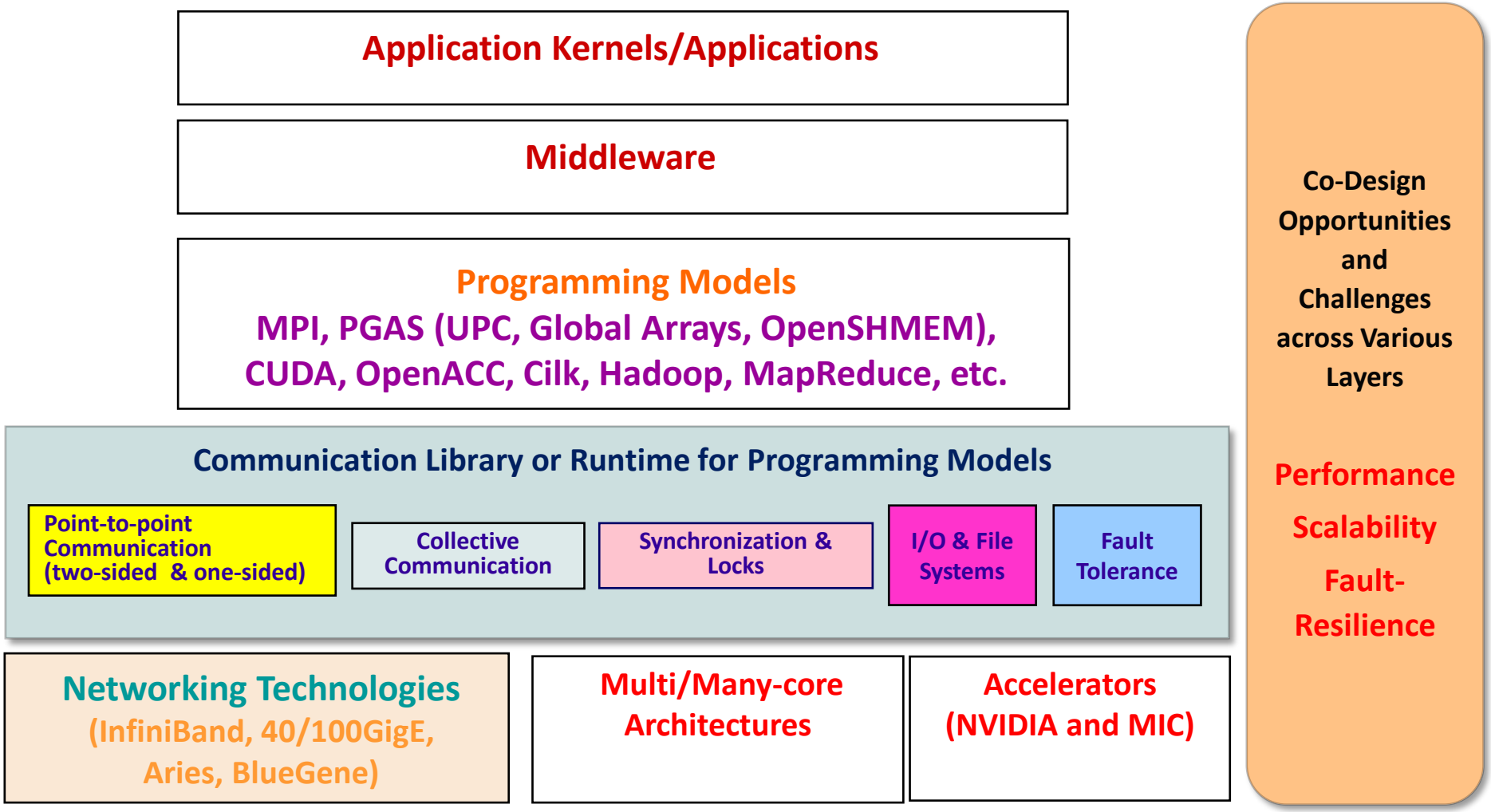
Hybrid (MPI+PGAS) Programming

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



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

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



Challenges in Designing (MPI+X) at Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Balancing intra-node and inter-node communication for next generation multi-core (128-1024 cores/node)
  - Multiple end-points per node
- Support for efficient multi-threading
- Support for GPGPUs and Accelerators
- Scalable Collective communication
  - Offload
  - Non-blocking
  - Topology-aware
  - Power-aware
- Fault-tolerance/resiliency
- QoS support for communication and I/O
- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI + OpenSHMEM, …)
- Virtualization

Additional Challenges for Designing Exascale Middleware

- **Extreme Low Memory Footprint**
  - Memory per core continues to decrease
- **D-L-A Framework**
  - **D**iscover
    - Overall network topology (fat-tree, 3D, …)
    - Network topology for processes for a given job
    - Node architecture
    - Health of network and node
  - **L**earn
    - Impact on performance and scalability
    - Potential for failure
  - **A**dapt
    - Internal protocols and algorithms
    - Process mapping
    - Fault-tolerance solutions
  - Low overhead techniques while delivering performance, scalability and fault-tolerance

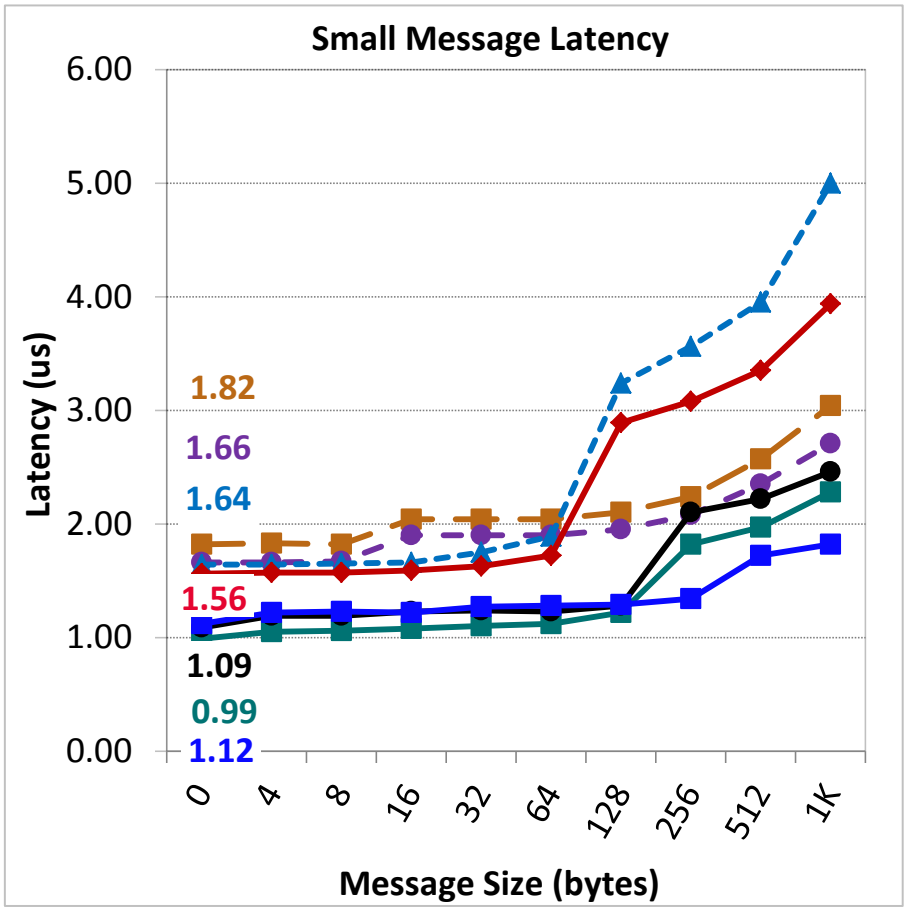
MVAPICH2/MVAPICH2-X Software

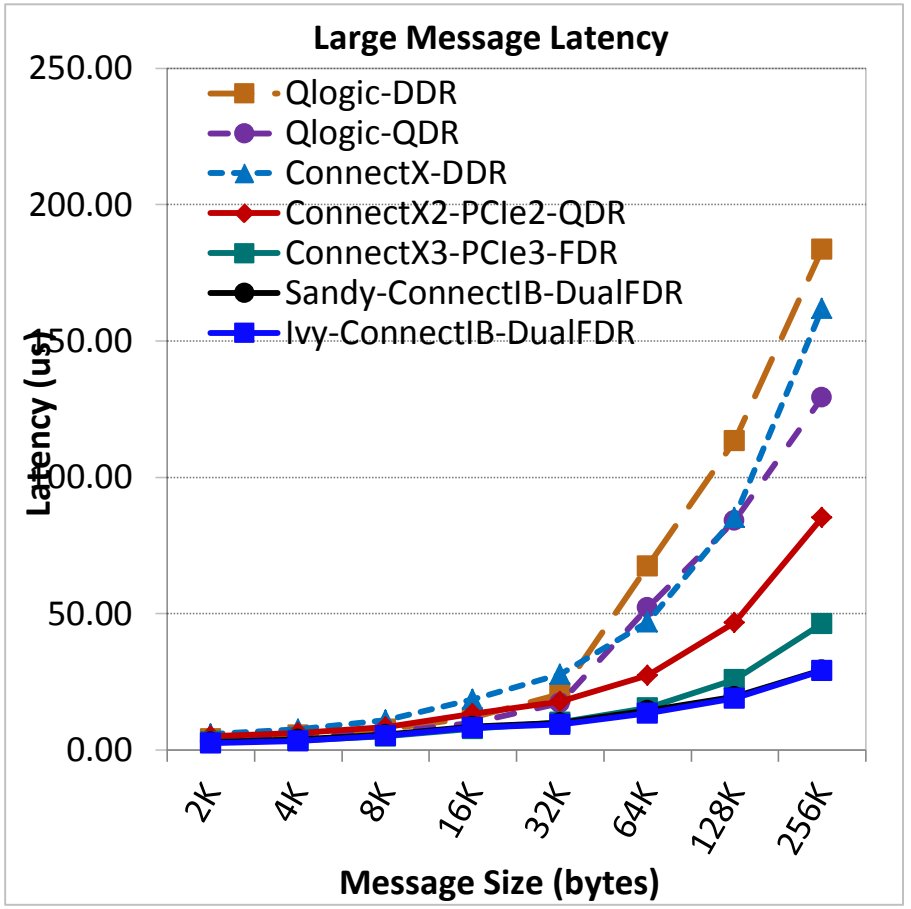
- High Performance open-source MPI Library for InfiniBand, 10Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Available since 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2012
  - Support for GPGPUs and MIC
  - **Used by more than 2,225 organizations in 73 countries**
  - **More than 224,000 downloads from OSU site directly**
  - Empowering many TOP500 clusters
    - 7th ranked 519,640-core cluster (Stampede) at TACC
    - 13th, 74,358-core (Tsubame 2.5) at Tokyo Institute of Technology
    - 23rd, 96,192-core (Pleiades) at NASA and many others
  - Available with software stacks of many IB, HSE, and server vendors including Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Partner in the U.S. NSF-TACC Stampede System

Overview of A Few Challenges being Addressed by MVAPICH2/MVAPICH2-X for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Support for GPGPUs
- Support for Intel MICs
- Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, …) with Unified Runtime
- Virtualization

One-way Latency: MPI over IB with MVAPICH2





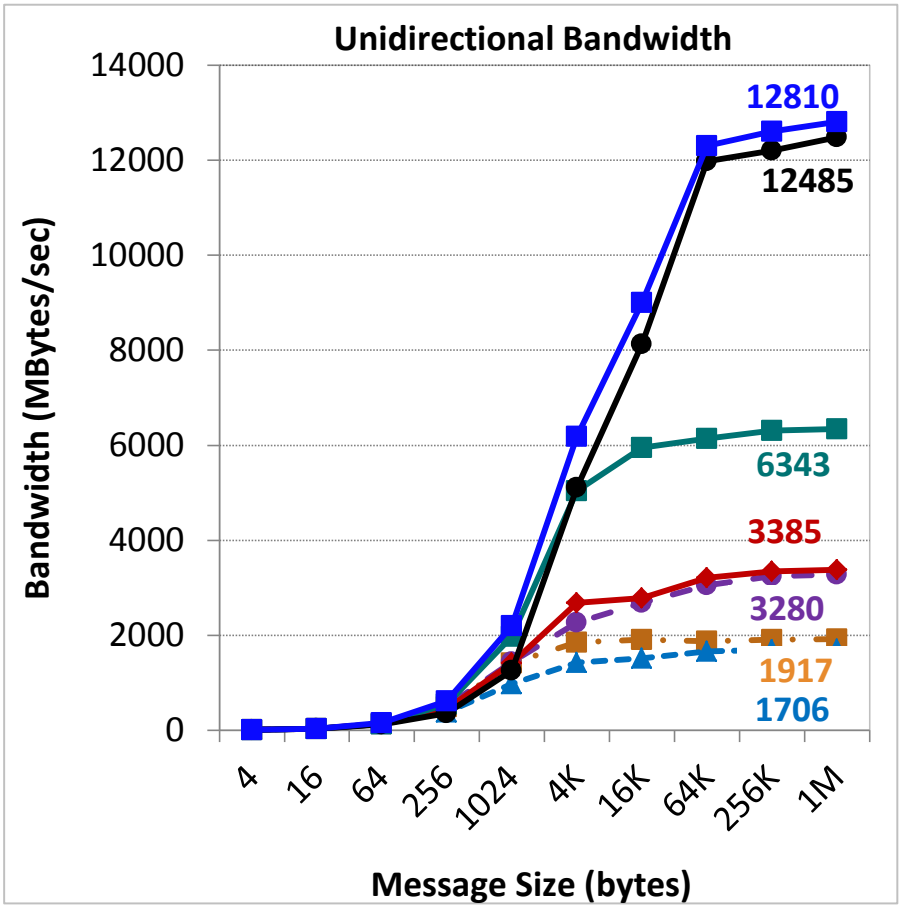
DDR, QDR - 2.4 GHz Quad-core (Westmere) Intel PCI Gen2 with IB switch

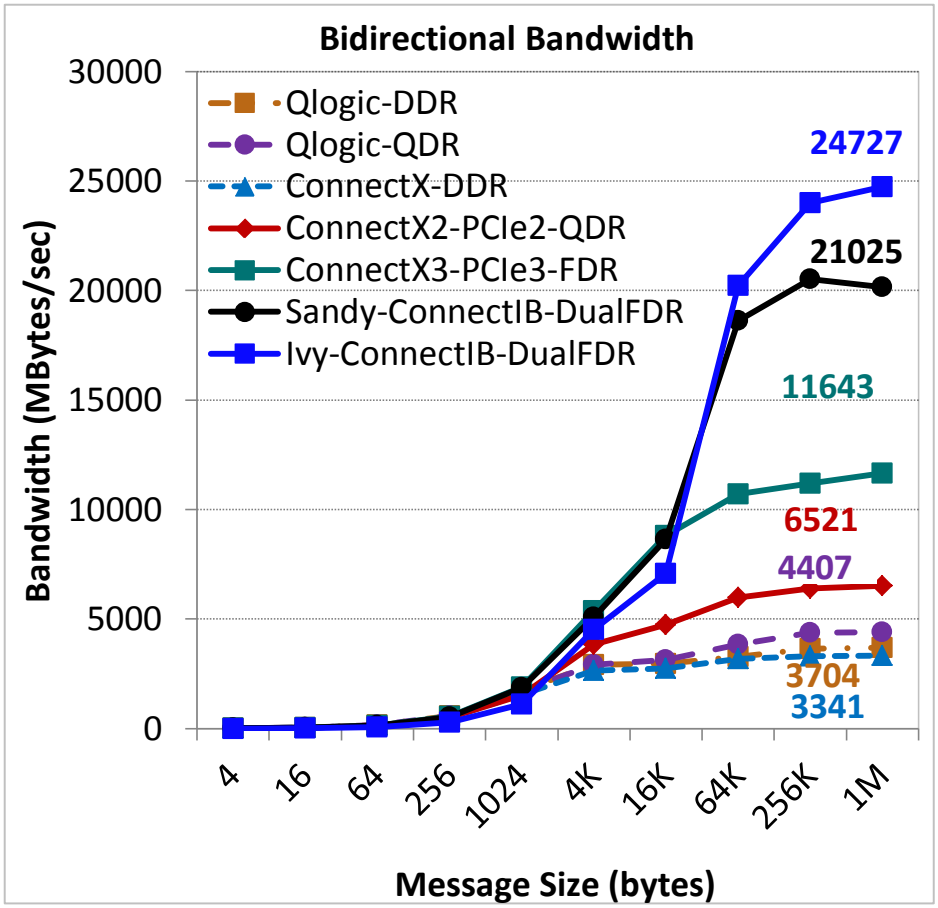
FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

Bandwidth: MPI over IB with MVAPICH2





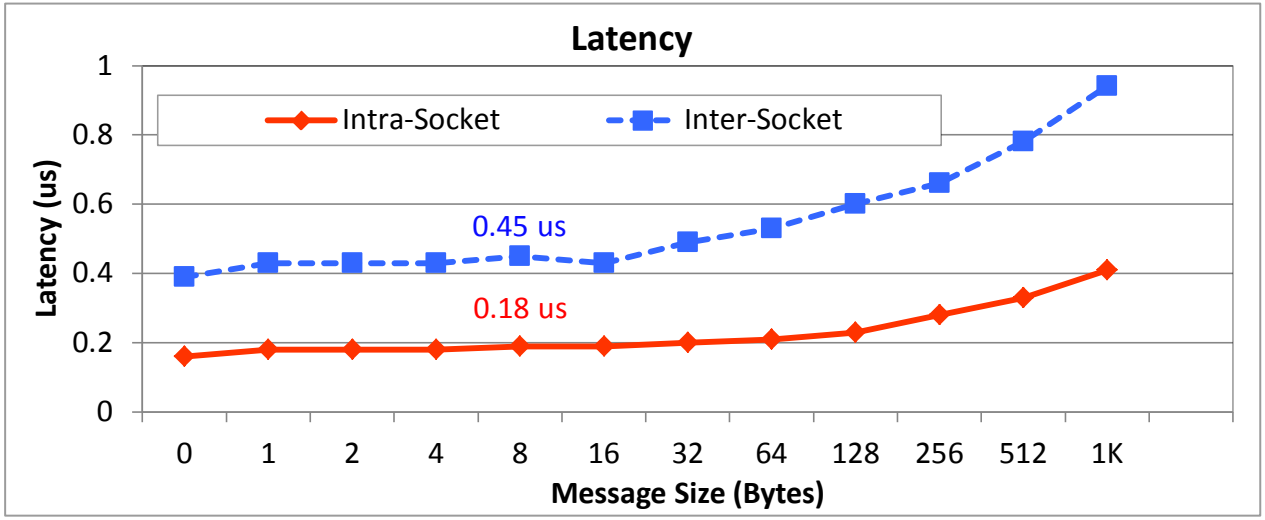
DDR, QDR - 2.4 GHz Quad-core (Westmere) Intel PCI Gen2 with IB switch

FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 2.6 GHz Octa-core (SandyBridge) Intel PCI Gen3 with IB switch

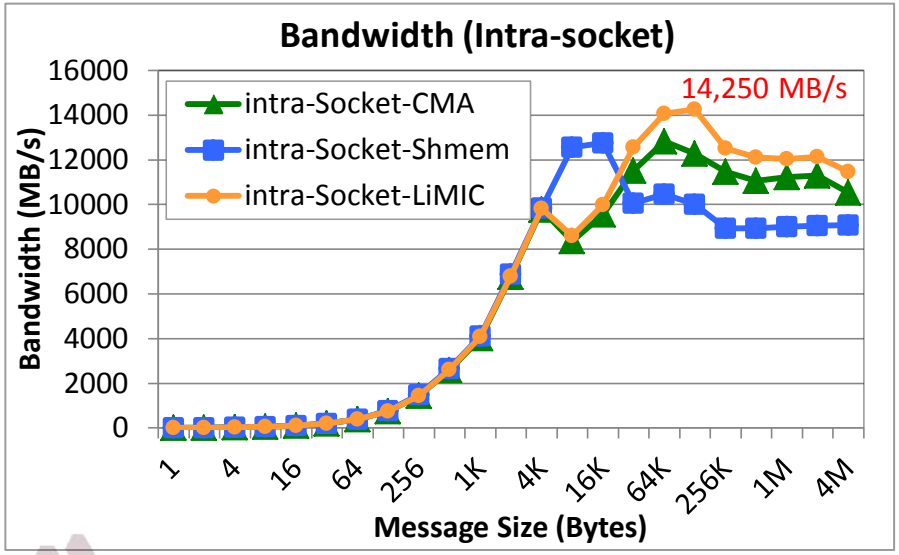
ConnectIB-Dual FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

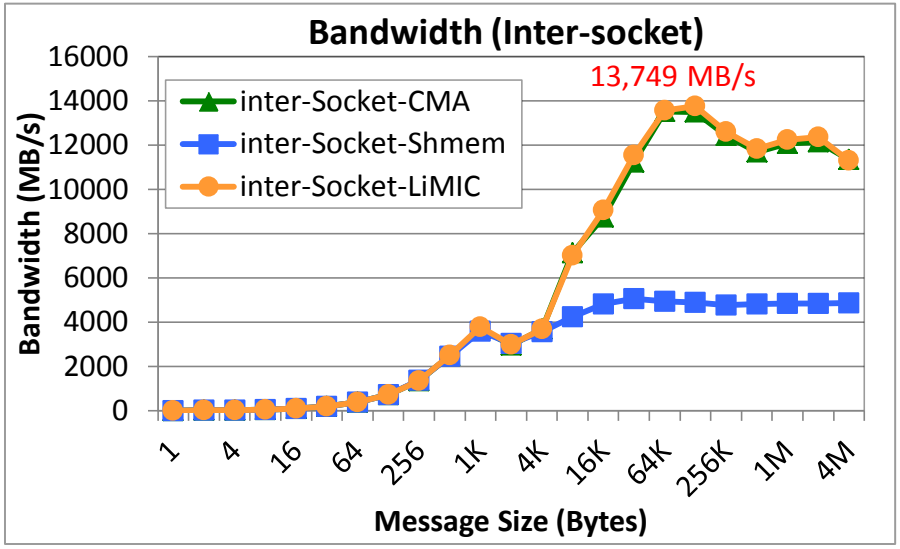
MVAPICH2 Two-Sided Intra-Node Performance
(Shared memory and Kernel-based Zero-copy Support (LiMIC and CMA))



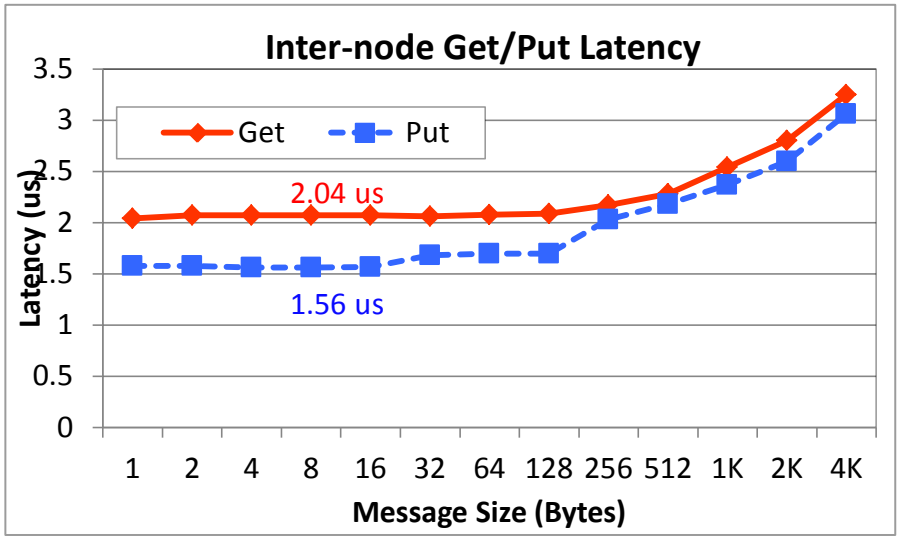
Latest MVAPICH2 2.1a

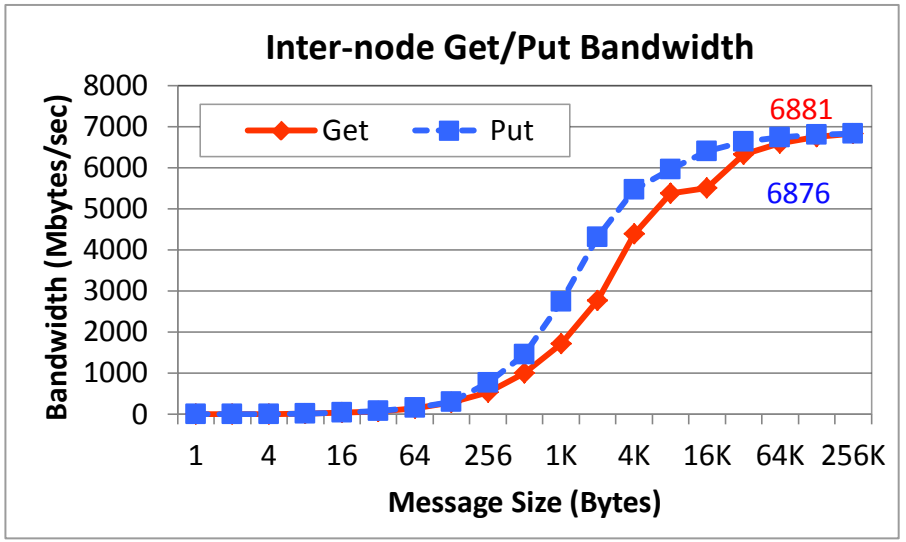
Intel Ivy-bridge

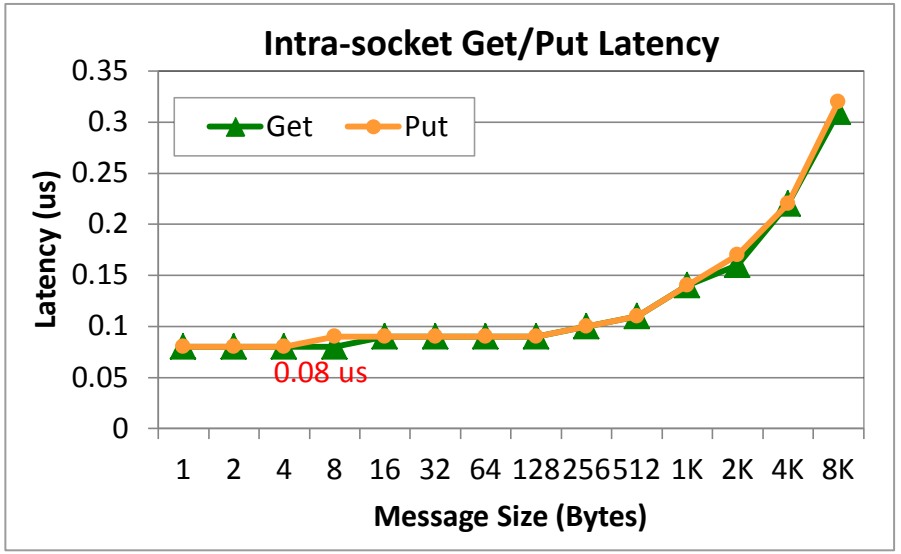


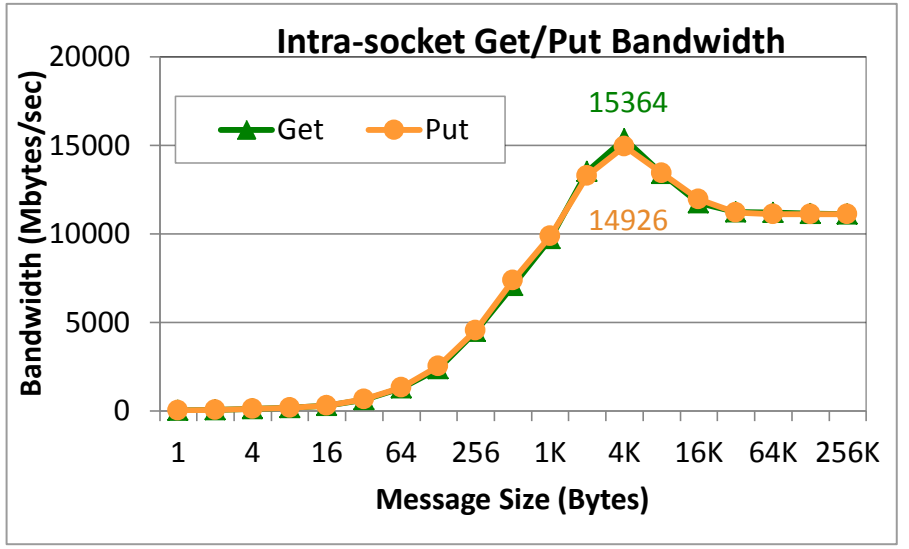


MPI-3 RMA Get/Put with Flush Performance



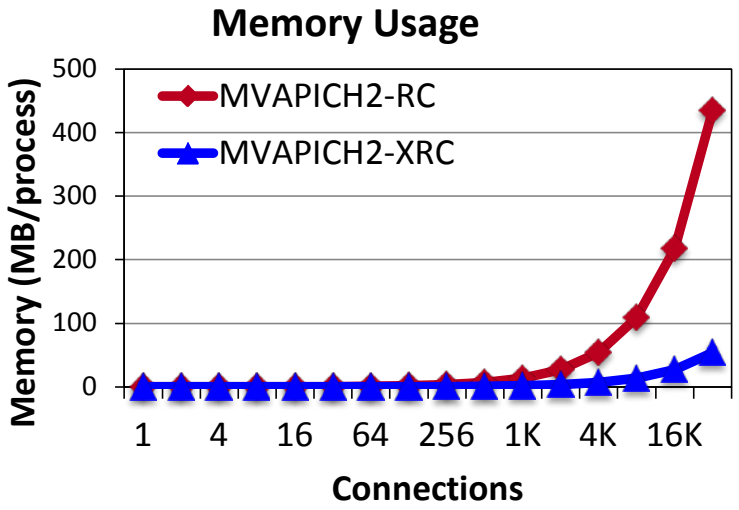


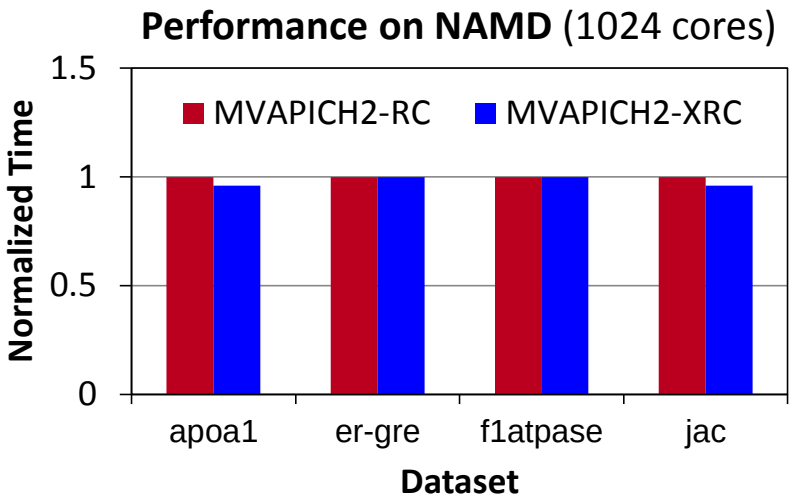




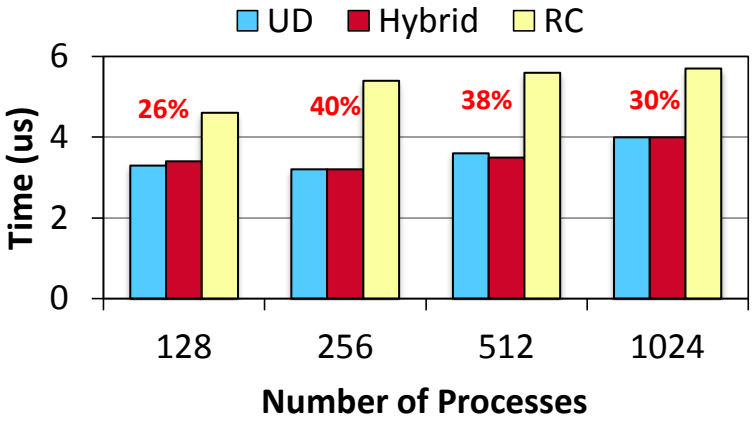
Latest MVAPICH2 2.0rc2, Intel Sandy-bridge with Connect-IB (single-port)

Minimizing memory footprint with XRC and Hybrid Mode





- Memory usage for 32K processes with 8-cores per node can be 54 MB/process (for connections)
- NAMD performance improves when there is frequent communication to many peers



- Both UD and RC/XRC have benefits
  - Hybrid for the best of both
- Available since MVAPICH2 1.7 as integrated interface
- Runtime Parameters: RC - default;
  - UD - MV2_USE_ONLY_UD=1
  - Hybrid - MV2_HYBRID_ENABLE_THRESHOLD=1

M. Koop, J. Sridhar and D. K. Panda, "Scalable MPI Design over InfiniBand using eXtended Reliable Connection," Cluster '08

Overview of A Few Challenges being Addressed by MVAPICH2/MVAPICH2-X for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Support for GPGPUs
- Support for Intel MICs
- Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, …) with Unified Runtime
- Virtualization

MPI + CUDA - Naive

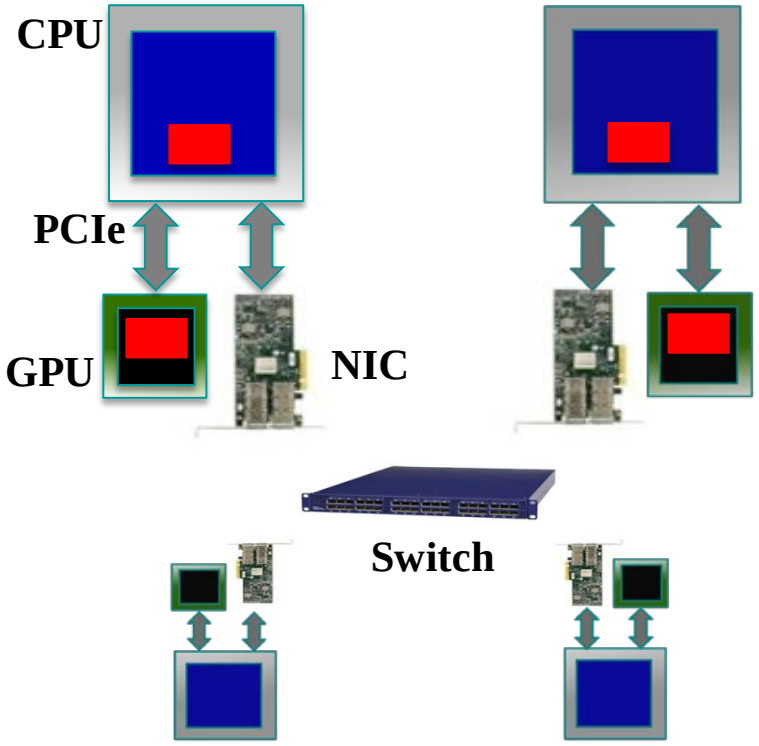
- Data movement in applications with standard MPI and CUDA interfaces

At Sender:

```
cudaMemcpy(s_hostbuf, s_devbuf, . . .);
MPI_Send(s_hostbuf, size, . . .);
```

At Receiver:

```
MPI_Recv(r_hostbuf, size, . . .);
cudaMemcpy(r_devbuf, r_hostbuf, . . .);
```



High Productivity and Low Performance

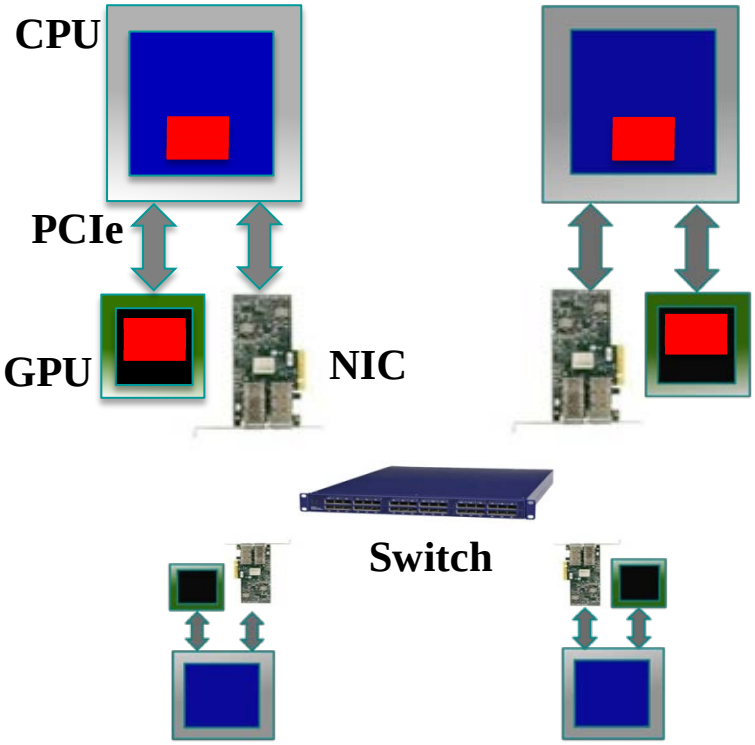
MPI + CUDA - Advanced

- Pipelining at user level with non-blocking MPI and CUDA interfaces

At Sender:

```
for (j = 0; j < pipeline_len; j++)
    cudaMemcpyAsync(s_hostbuf + j * blk, s_devbuf + j * blksz,
      …);
for (j = 0; j < pipeline_len; j++) {
       while (result != cudaSucess) {
            result = cudaStreamQuery(…);
             if(j > 0) MPI_Test(…);
        }
        MPI_Isend(s_hostbuf + j * block_sz, blksz . . .);
 }
MPI_Waitall();
```

<<Similar at receiver>>



Low Productivity and High Performance

GPU-Aware MPI Library: MVAPICH2-GPU

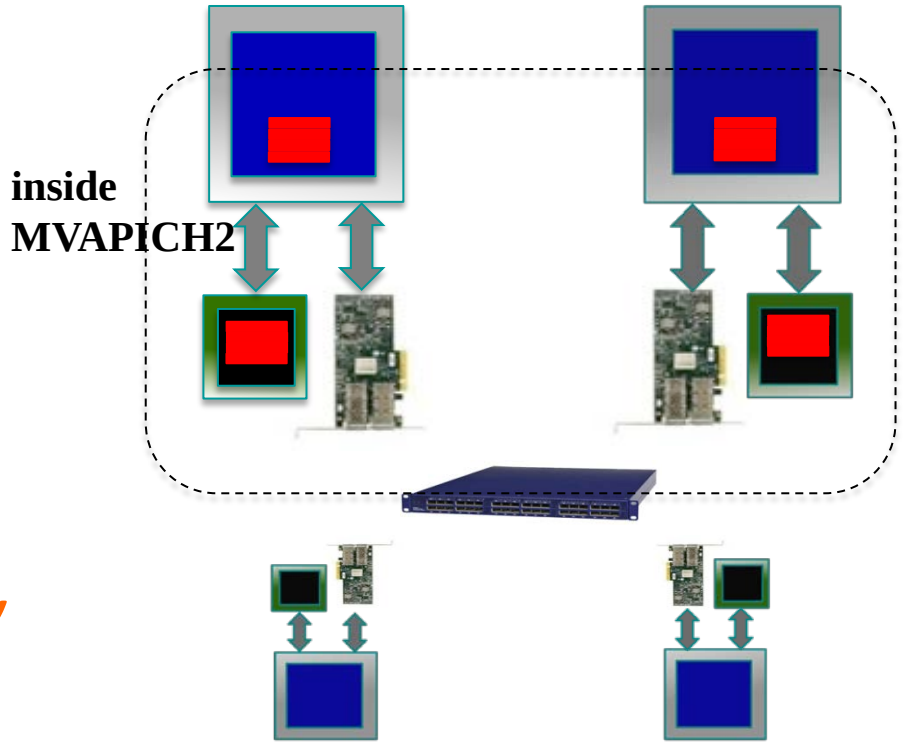
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

```
MPI_Send(s_devbuf, size, …);
```

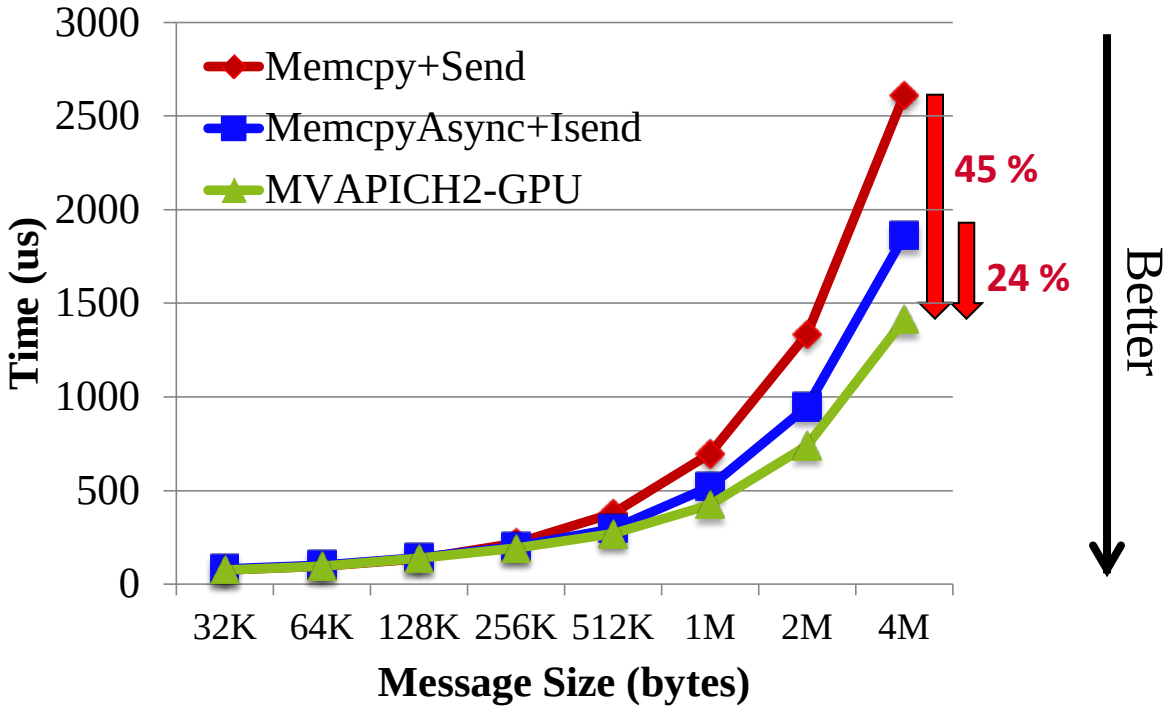
At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```



High Performance and High Productivity

MPI Micro-benchmark Performance



- 45% improvement compared with a naïve user-level implementation (Memcpy+Send), for 4MB messages
- 24% improvement compared with an advanced user-level implementation (MemcpyAsync+Isend), for 4MB messages

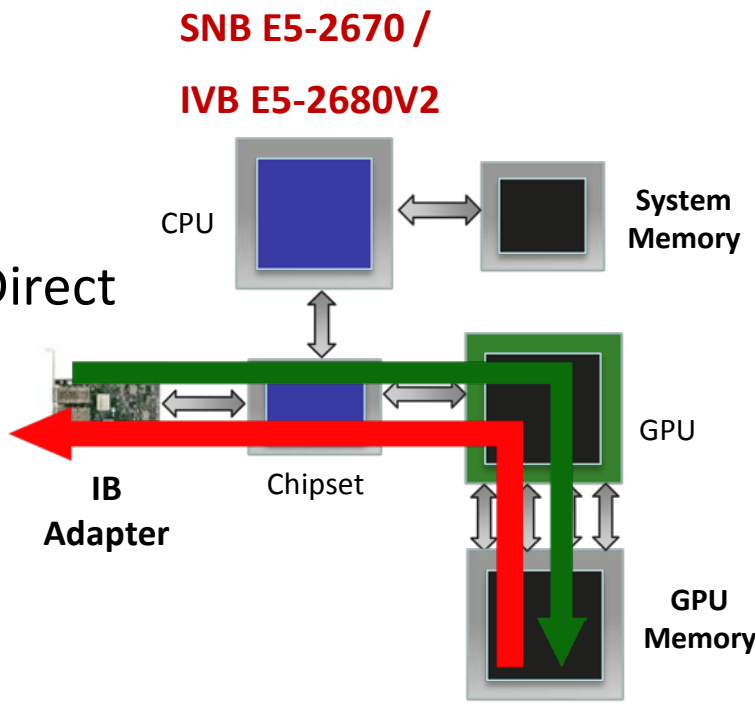
H. Wang, S. Potluri, M. Luo, A. Singh, S. Sur and D. K. Panda, MVAPICH2-GPU: Optimized GPU to GPU Communication for InfiniBand Clusters, ISC '11

CUDA-Aware MPI: MVAPICH2 1.8, 1.9, 2.0 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

GPU-Direct RDMA (GDR) with CUDA

- OFED with support for GPUDirect RDMA is developed by NVIDIA and Mellanox
- OSU has a design of MVAPICH2 using GPUDirect RDMA
  - Hybrid design using GPU-Direct RDMA
    - GPUDirect RDMA and Host-based pipelining
    - Alleviates P2P bandwidth bottlenecks on SandyBridge and IvyBridge
  - Support for communication using multi-rail
  - Support for Mellanox Connect-IB and ConnectX VPI adapters
  - Support for RoCE with Mellanox ConnectX VPI adapters



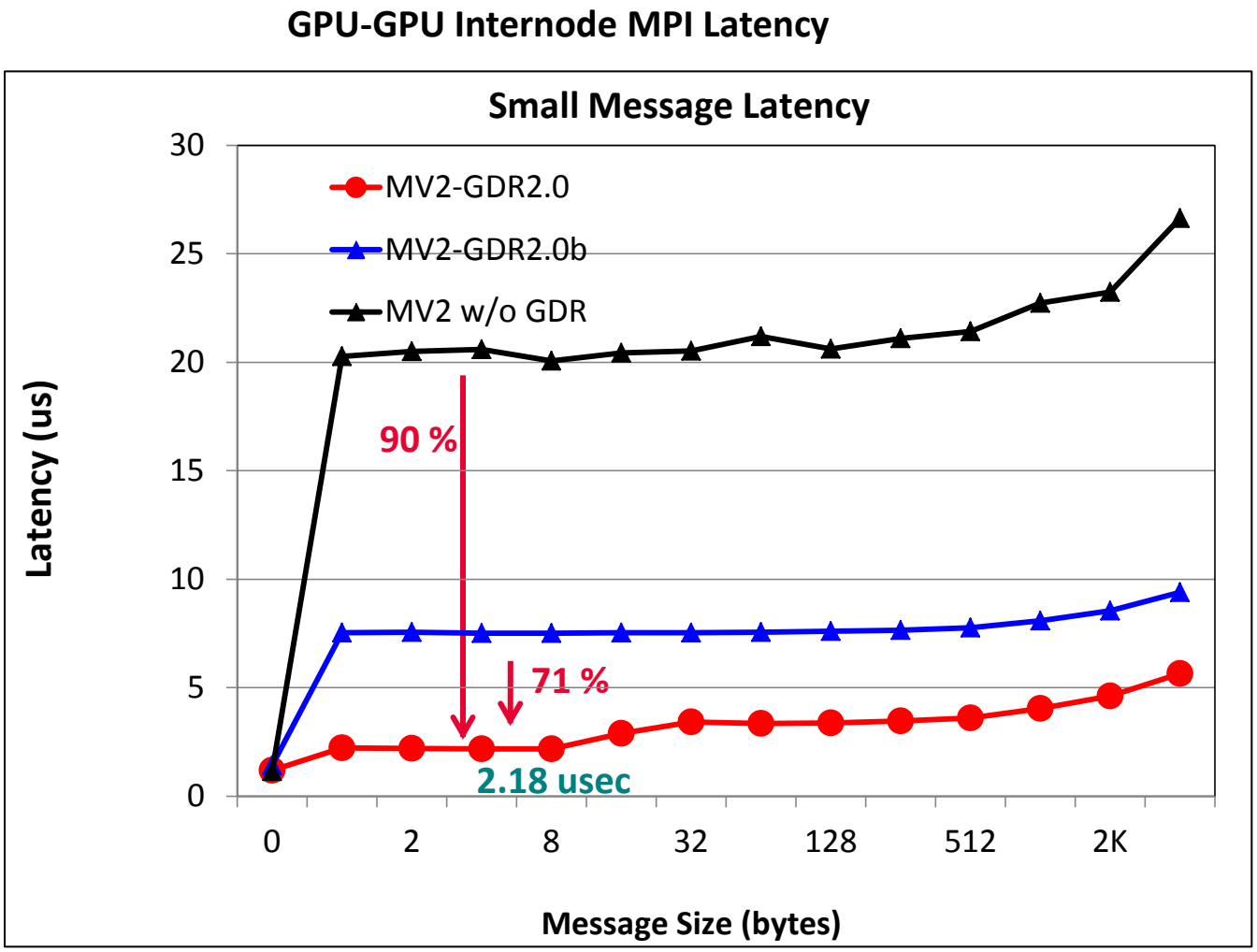
SNB E5-2670

P2P write: 5.2 GB/s
P2P read: < 1.0 GB/s

IVB E5-2680V2

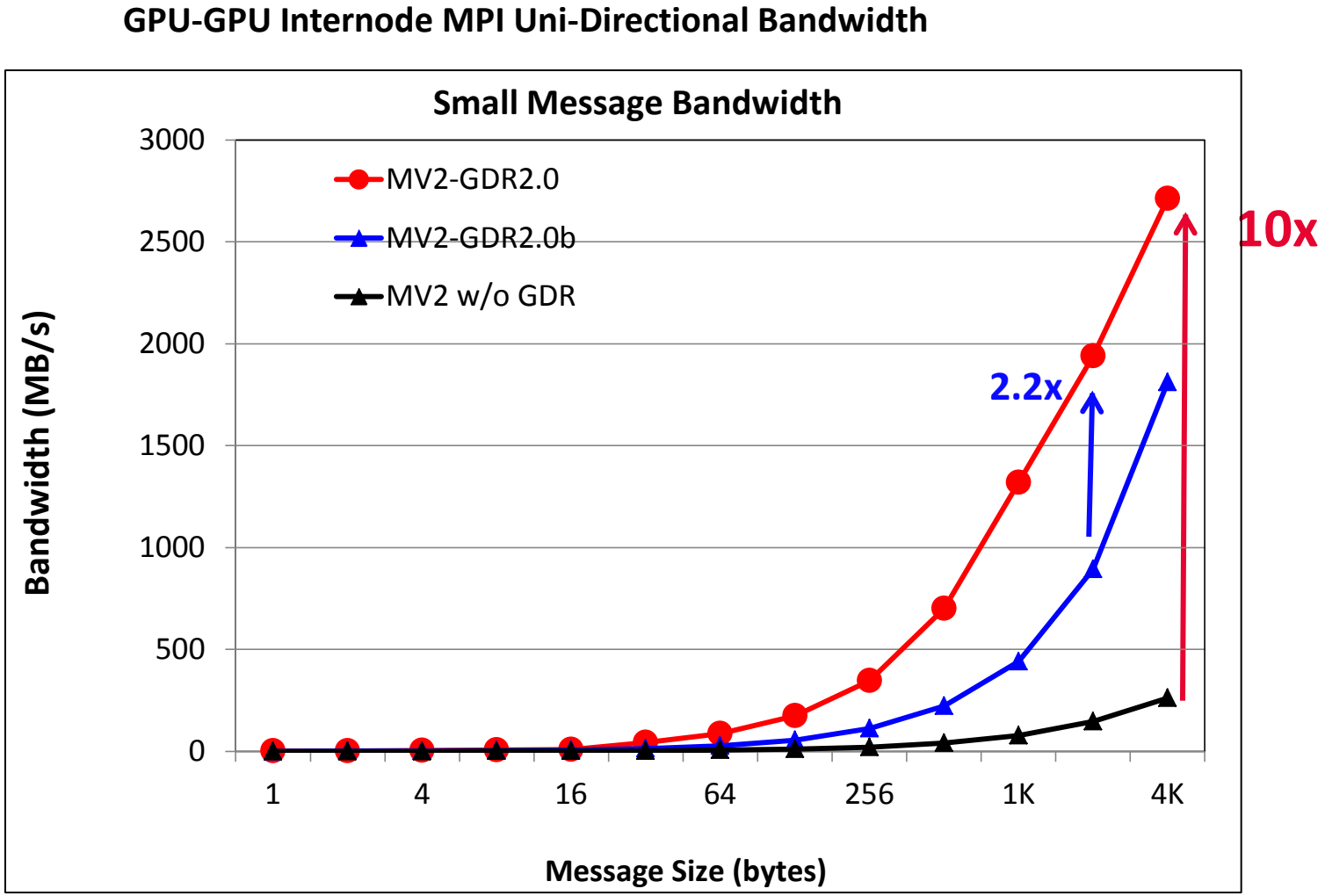
P2P write: 6.4 GB/s
P2P read: 3.5 GB/s

Performance of MVAPICH2 with GPU-Direct-RDMA: Latency



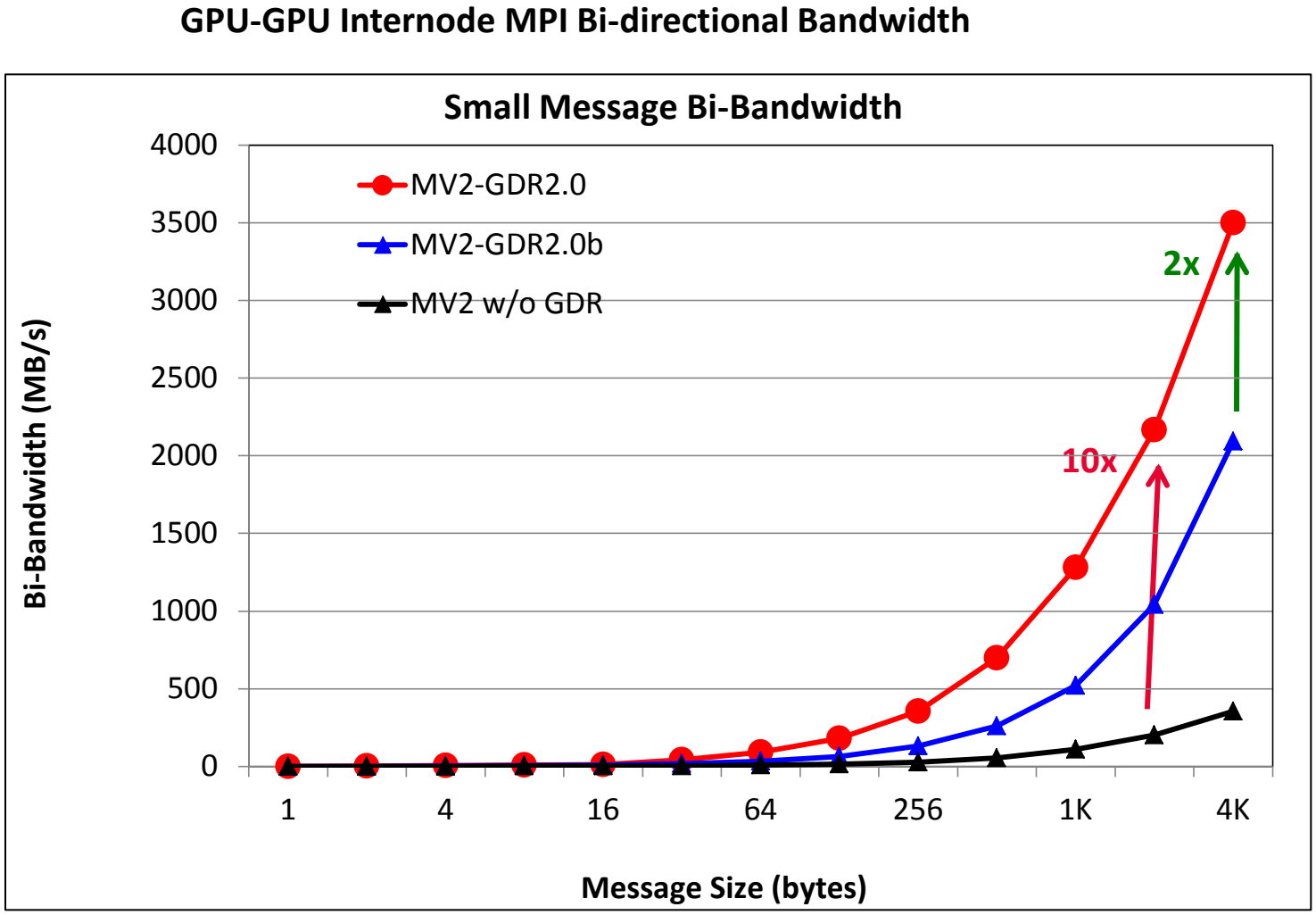
MVAPICH2-GDR-2.0
Intel Ivy Bridge (E5-2680 v2) node with 20 cores
NVIDIA Tesla K40c GPU, Mellanox Connect-IB Dual-FDR HCA
CUDA 6.5, Mellanox OFED 2.1 with GPU-Direct-RDMA

Performance of MVAPICH2 with GPU-Direct-RDMA: Bandwidth



MVAPICH2-GDR-2.0
Intel Ivy Bridge (E5-2680 v2) node with 20 cores
NVIDIA Tesla K40c GPU, Mellanox Connect-IB Dual-FDR HCA
CUDA 6.5, Mellanox OFED 2.1 with GPU-Direct-RDMA

Performance of MVAPICH2 with GPU-Direct-RDMA: Bi-Bandwidth

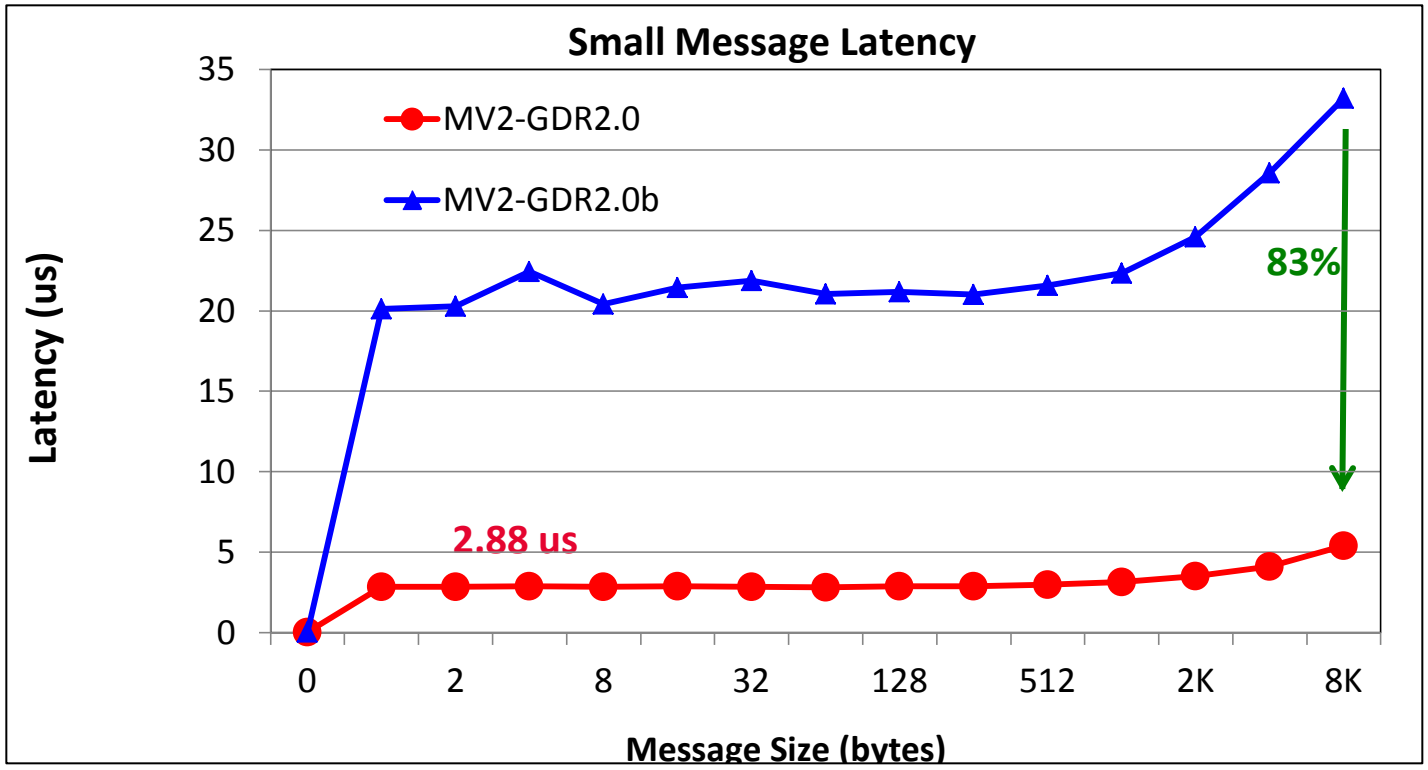


MVAPICH2-GDR-2.0
Intel Ivy Bridge (E5-2680 v2) node with 20 cores
NVIDIA Tesla K40c GPU, Mellanox Connect-IB Dual-FDR HCA
CUDA 6.5, Mellanox OFED 2.1 with GPU-Direct-RDMA

Performance of MVAPICH2 with GPU-Direct-RDMA: MPI-3 RMA

GPU-GPU Internode MPI Put latency (RMA put operation Device to Device)

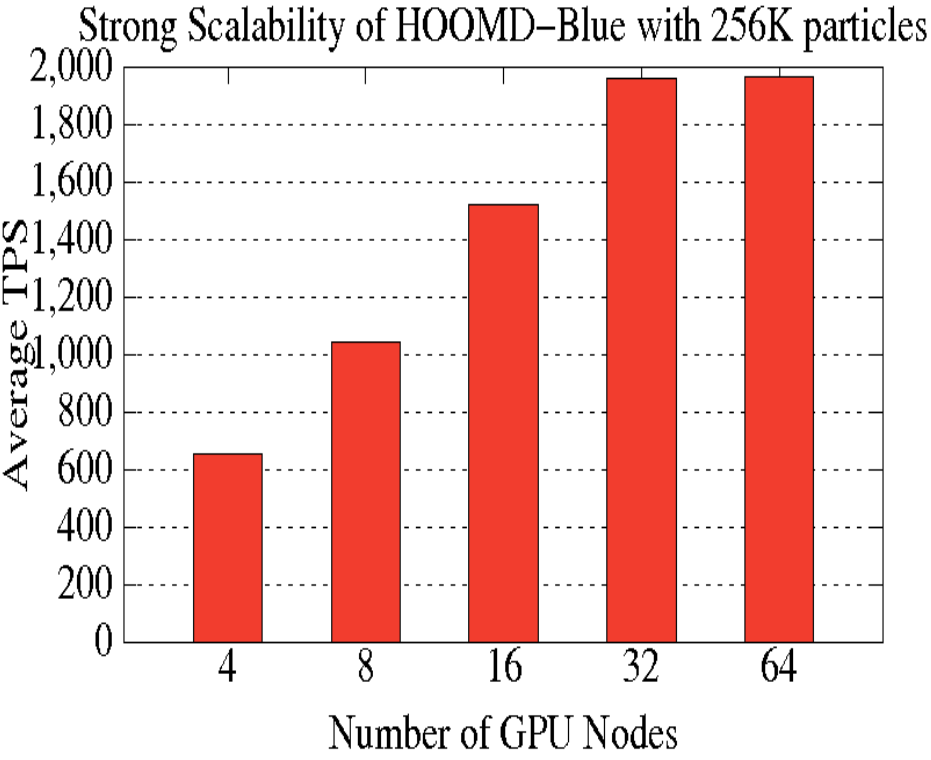
MPI-3 RMA provides flexible synchronization and completion primitives



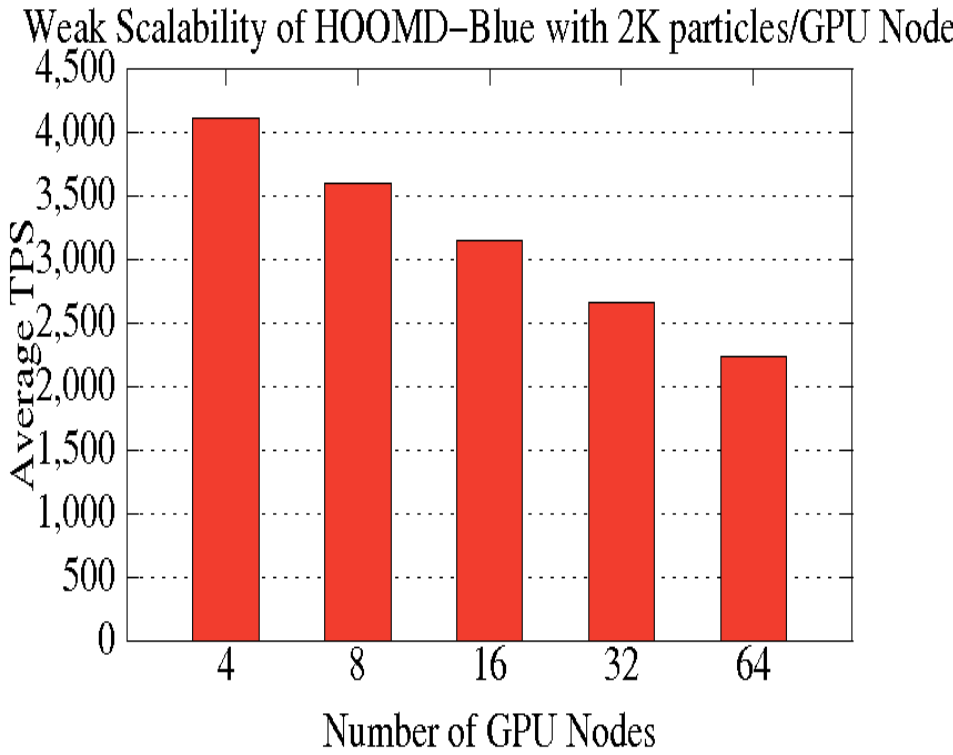
MVAPICH2-GDR-2.0
Intel Ivy Bridge (E5-2680 v2) node with 20 cores
NVIDIA Tesla K40c GPU, Mellanox Connect-IB Dual-FDR HCA
CUDA 6.5, Mellanox OFED 2.1 with GPU-Direct-RDMA

Application-Level Evaluation (HOOMD-blue)

HOOMD-blue Strong Scaling



HOOMD-blue Weak Scaling



- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- MV2-GDR 2.0 (released on 08/23/14) : try it out !!
  - 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

Using MVAPICH2-GPUDirect Version

- MVAPICH2-2.0 with GDR support can be downloaded from https://mvapich.cse.ohio-state.edu/download/mvapich2gdr/
- System software requirements
  - Mellanox OFED 2.1 or later
  - NVIDIA Driver 331.20 or later
  - NVIDIA CUDA Toolkit 6.0 or later
  - Plugin for GPUDirect RDMA
  - http://www.mellanox.com/page/products_dyn?product_family=116
  - Strongly Recommended : use the new GDRCOPY module from NVIDIA
    - https://github.com/drossetti/gdrcopy
- Has optimized designs for point-to-point communication using GDR
- Contact MVAPICH help list with any questions related to the package

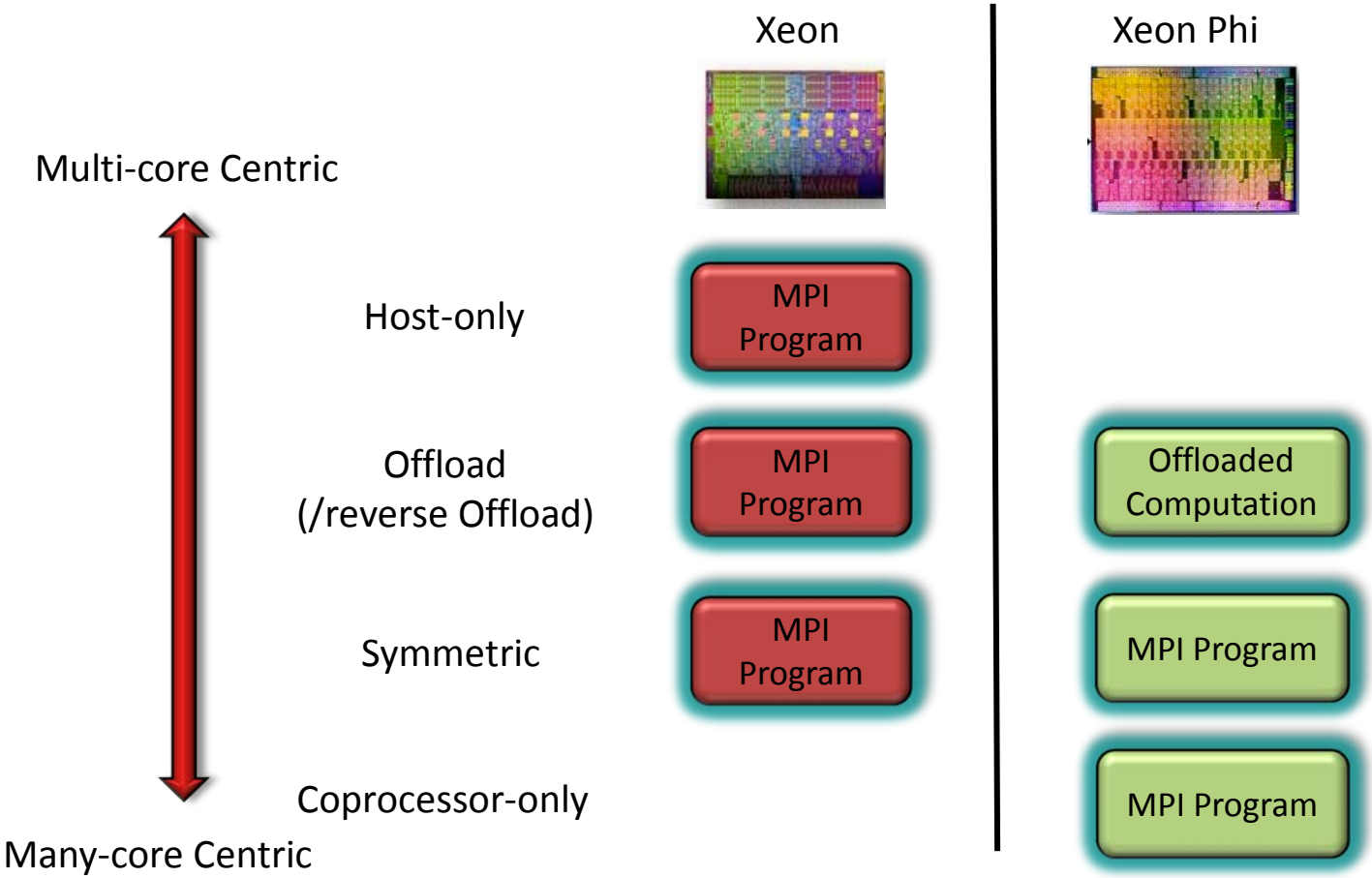
  mvapich-help@cse.ohio-state.edu

Overview of A Few Challenges being Addressed by MVAPICH2/MVAPICH2-X for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Support for GPGPUs
- Support for Intel MICs
- Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, …) with Unified Runtime
- Virtualization

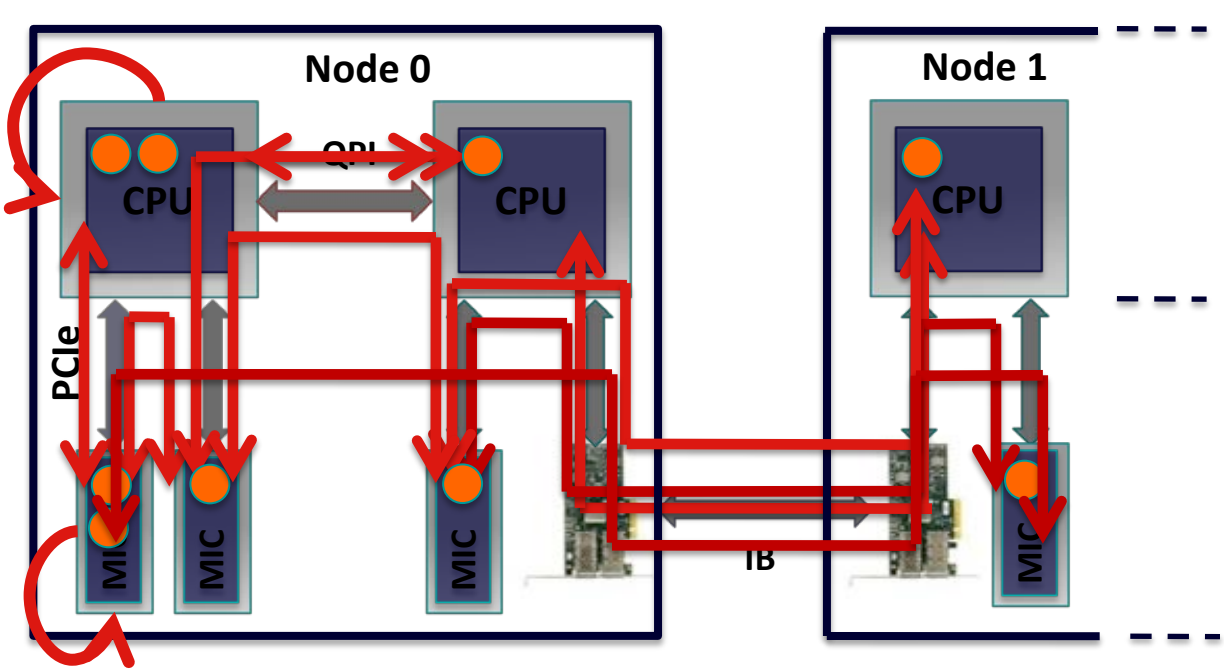
MPI Applications on MIC Clusters

- Flexibility in launching MPI jobs on clusters with Xeon Phi



Data Movement on Intel Xeon Phi Clusters

- Connected as PCIe devices – Flexibility but Complexity



MPI Process

1. Intra-Socket
2. Inter-Socket
3. Inter-Node
4. Intra-MIC
5. Intra-Socket MIC-MIC
6. Inter-Socket MIC-MIC
7. Inter-Node MIC-MIC
8. Intra-Socket MIC-Host
9. Inter-Socket MIC-Host
10. Inter-Node MIC-Host

11. Inter-Node MIC-MIC with IB adapter on remote socket

and more . . .

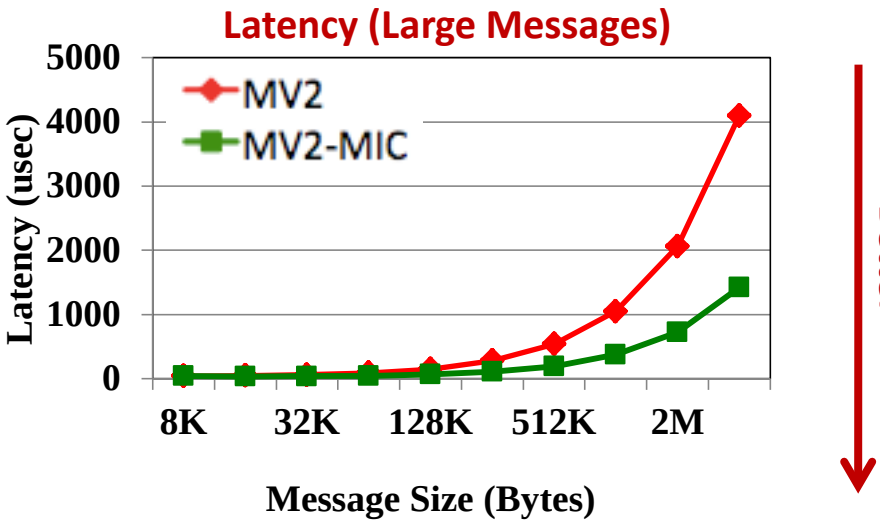
- Critical for runtimes to optimize data movement, hiding the complexity

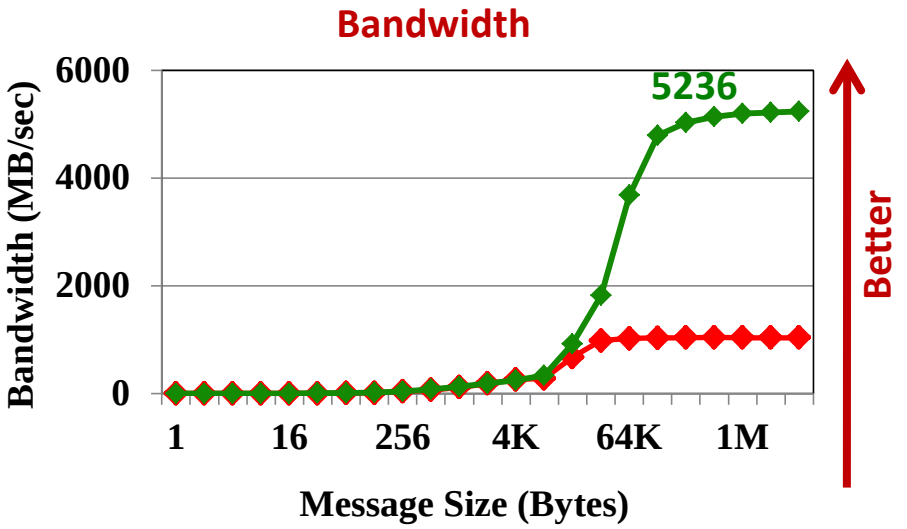
MVAPICH2-MIC Design for Clusters with IB and MIC

- Offload Mode
- Intranode Communication
  - Coprocessor-only Mode
  - Symmetric Mode
- Internode Communication
  - Coprocessors-only
  - Symmetric Mode
- Multi-MIC Node Configurations

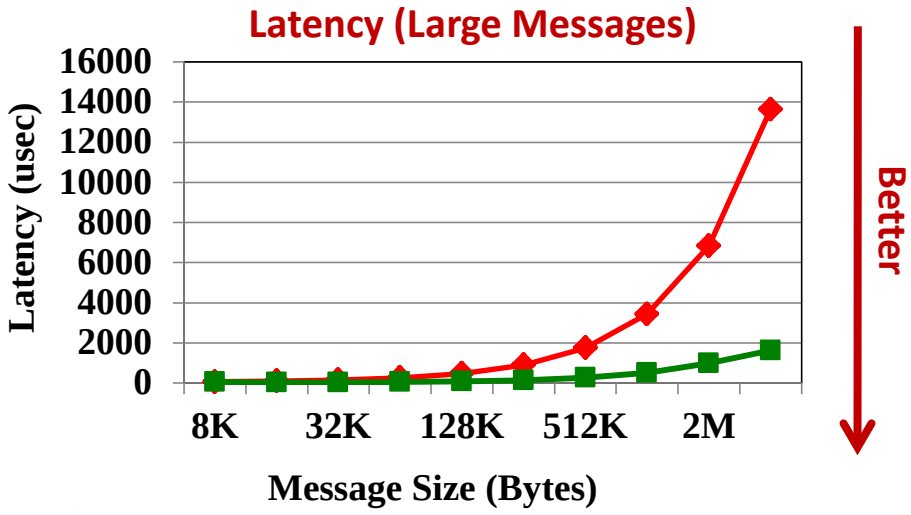
MIC-Remote-MIC P2P Communication

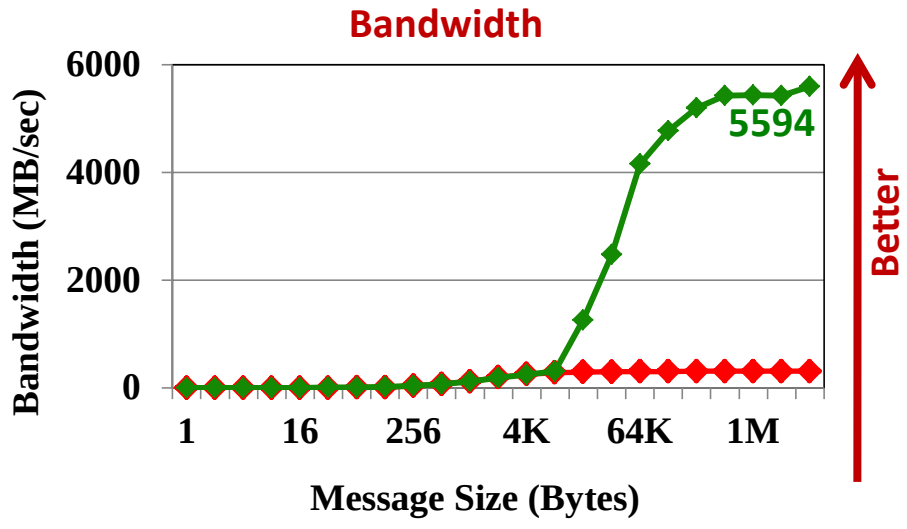
Intra-socket P2P





Inter-socket P2P

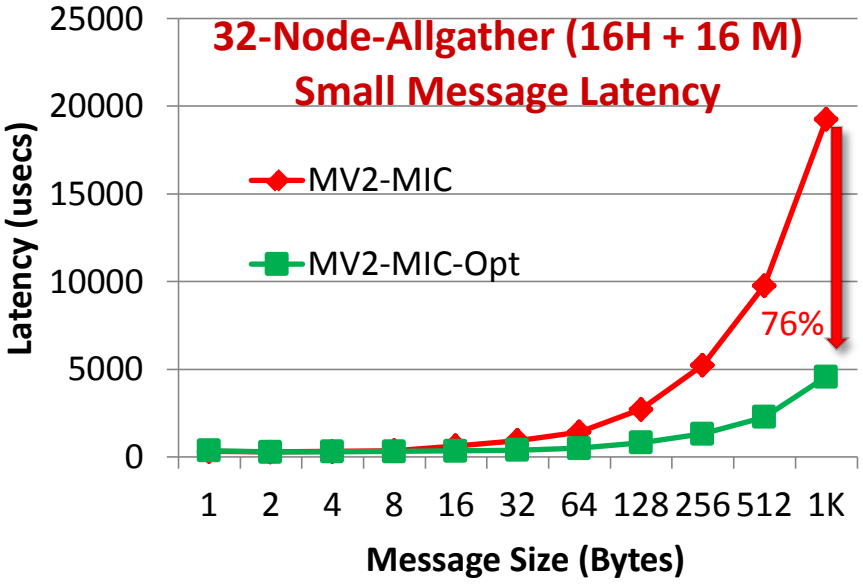


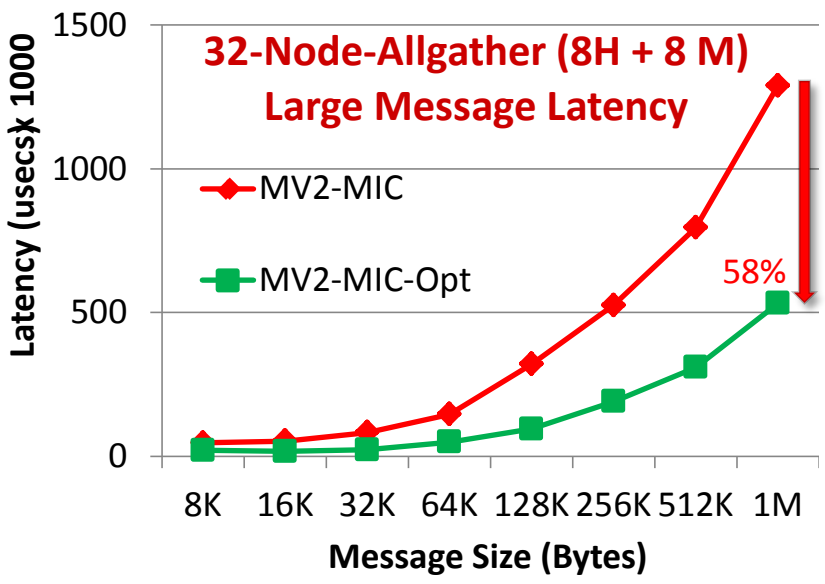


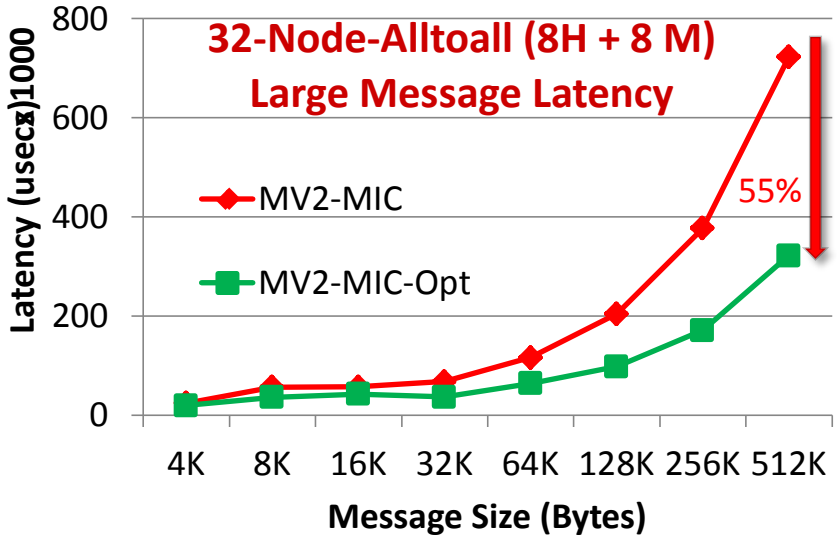
Latest Status on MVAPICH2-MIC

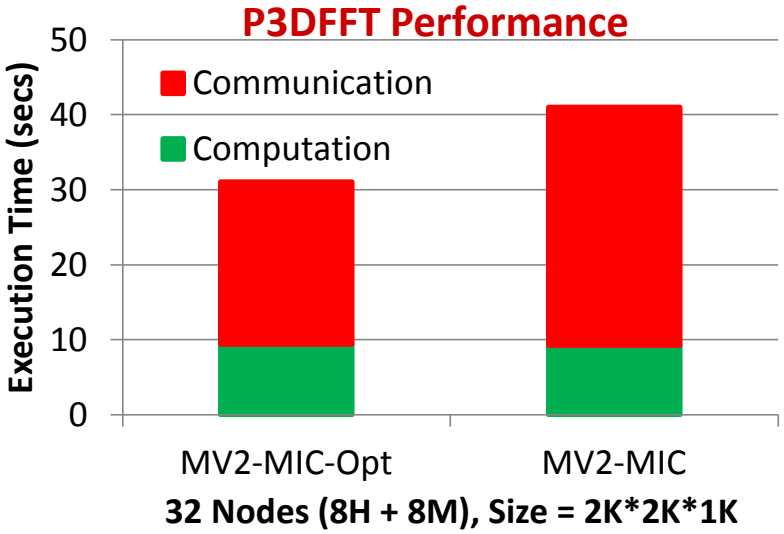
- Running on three major systems
  - Stampede : module swap mvapich2 mvapich2-mic/20130911
  - Blueridge(Virginia Tech) : module swap mvapich2 mvapich2-mic/1.9
  - Beacon (UTK) : module unload intel-mpi; module load mvapich2-mic/1.9
- A new version based on MVAPICH2 2.0 is being worked out
- Will be available in a few weeks

Optimized MPI Collectives for MIC Clusters (Allgather & Alltoall)







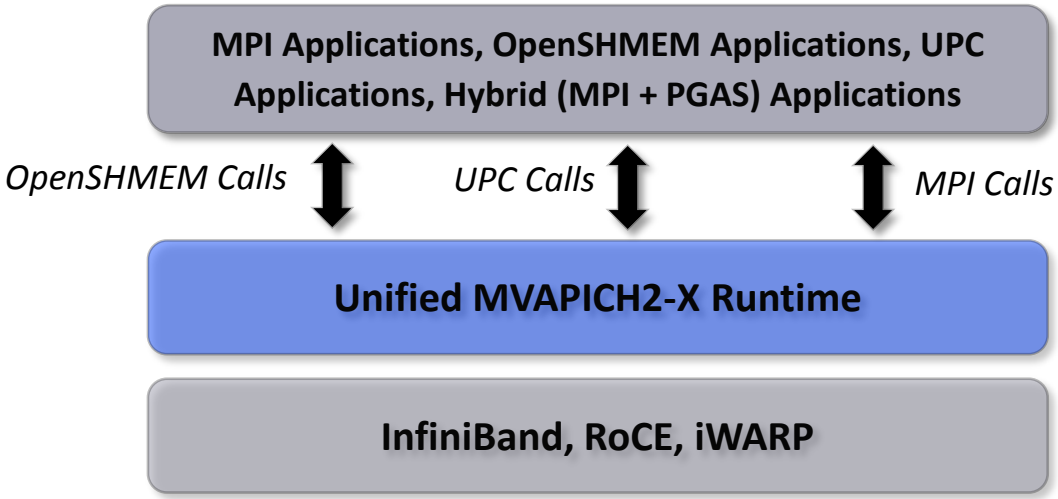


A. Venkatesh, S. Potluri, R. Rajachandrasekar, M. Luo, K. Hamidouche and D. K. Panda - High Performance Alltoall and Allgather designs for InfiniBand MIC Clusters; IPDPS'14, May 2014

Overview of A Few Challenges being Addressed by MVAPICH2/MVAPICH2-X for Exascale

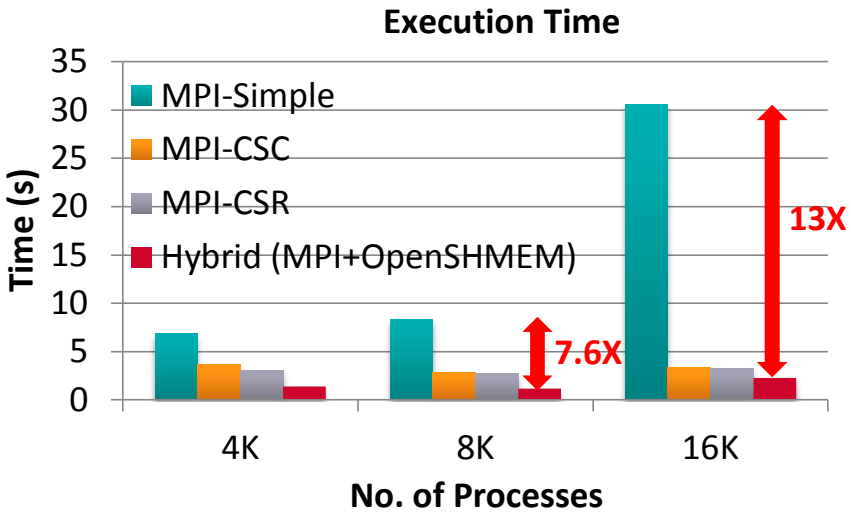
- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Support for GPGPUs
- Support for Intel MICs
- Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, …) with Unified Runtime
- Virtualization

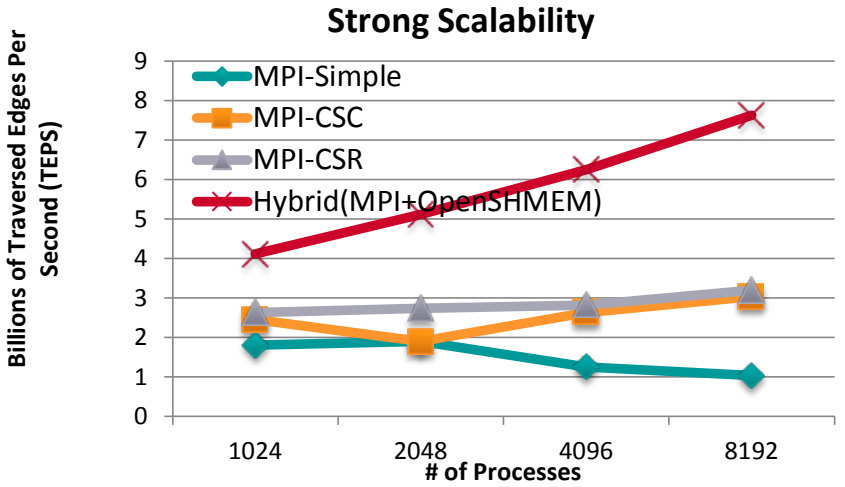
MVAPICH2-X for Hybrid MPI + PGAS Applications

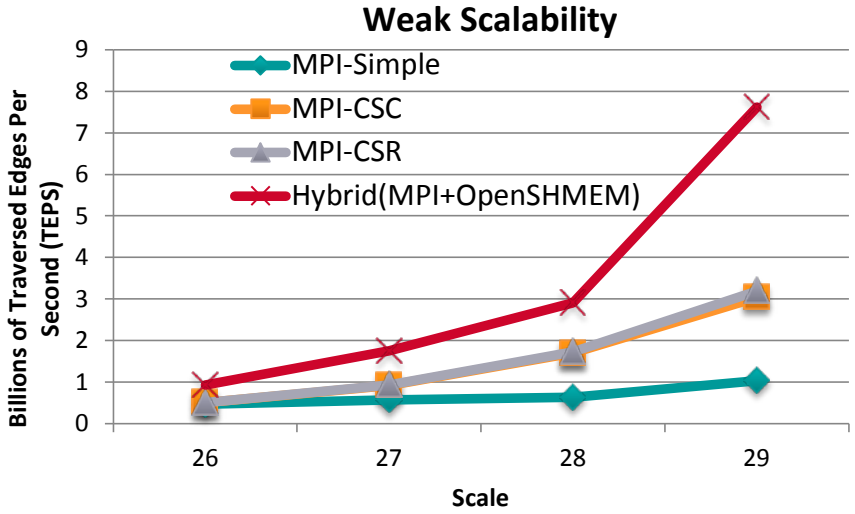


- Unified communication runtime for MPI, UPC, OpenSHMEM available with MVAPICH2-X 1.9 onwards!
  - http://mvapich.cse.ohio-state.edu
- Feature Highlights
  - Supports MPI(+OpenMP), OpenSHMEM, UPC, MPI(+OpenMP) + OpenSHMEM, MPI(+OpenMP) + UPC
  - MPI-3 compliant, OpenSHMEM v1.0 standard compliant, UPC v1.2 standard compliant
  - Scalable Inter-node and Intra-node communication – point-to-point and collectives

Hybrid MPI+OpenSHMEM Graph500 Design





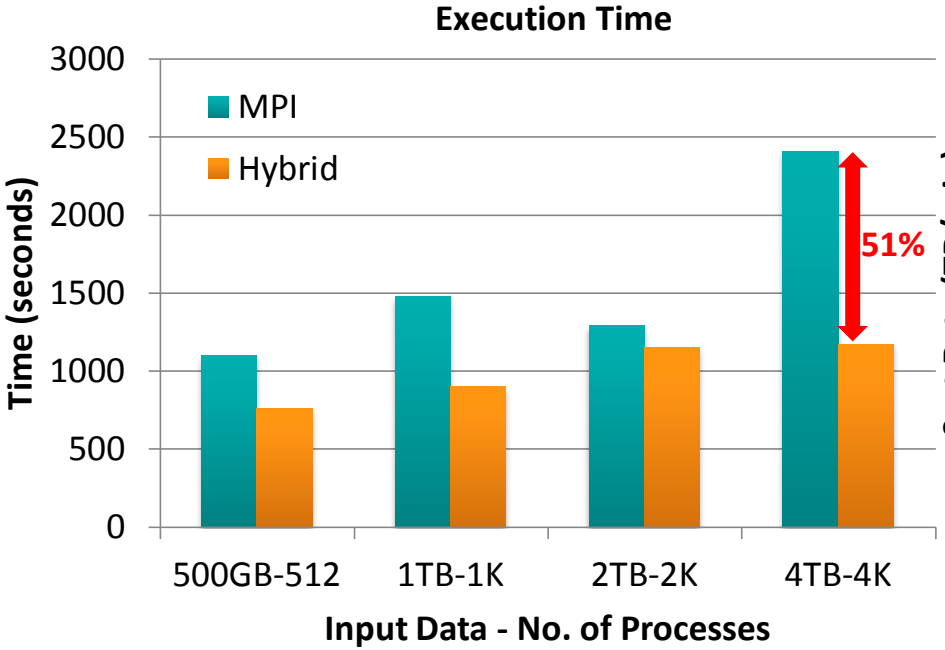


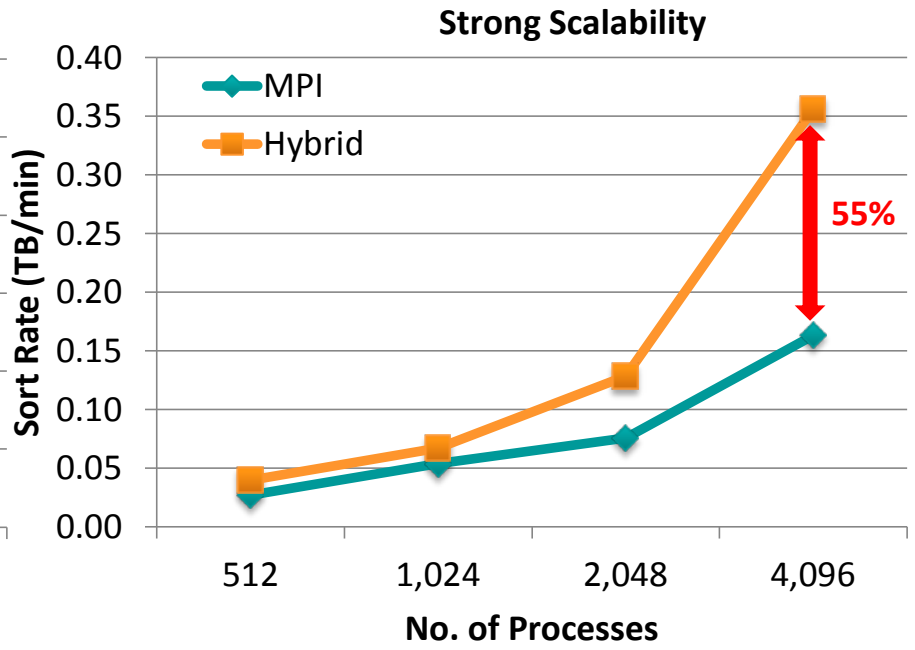
- Performance of Hybrid (MPI+OpenSHMEM) Graph500 Design
  - 8,192 processes
    - **2.4X** improvement over MPI-CSR
    - **7.6X** improvement over MPI-Simple
  - 16,384 processes
    - **1.5X** improvement over MPI-CSR
    - **13X** improvement over MPI-Simple

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

Hybrid MPI+OpenSHMEM Sort Application





- Performance of Hybrid (MPI+OpenSHMEM) Sort Application
  - Execution Time
    - 4TB Input size at 4,096 cores: MPI – 2408 seconds, Hybrid: 1172 seconds
    - **51%** improvement over MPI-based design
  - Strong Scalability (configuration: constant input size of 500GB)
    - **At 4,096 cores:** MPI – 0.16 TB/min, Hybrid – 0.36 TB/min
    - 55% improvement over MPI based design

J. Jose, S. Potluri, H. Subramoni, X. Lu, K. Hamidouche, K. Schulz, H. Sundar and D. K. Panda, Designing Scalable Out-of-core Sorting with Hybrid MPI+PGAS Programming Models, PGAS'14, Oct 2014

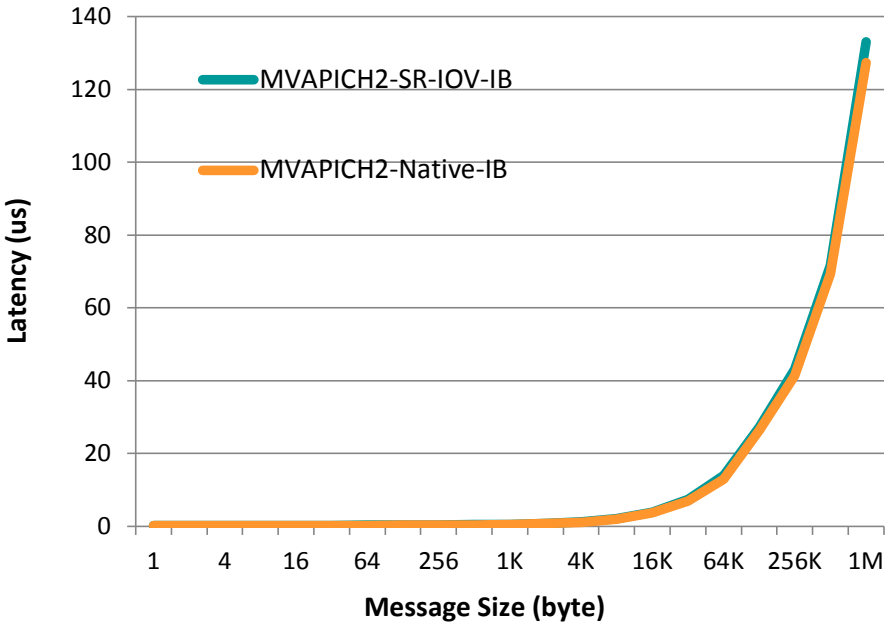
Overview of A Few Challenges being Addressed by MVAPICH2/MVAPICH2-X for Exascale

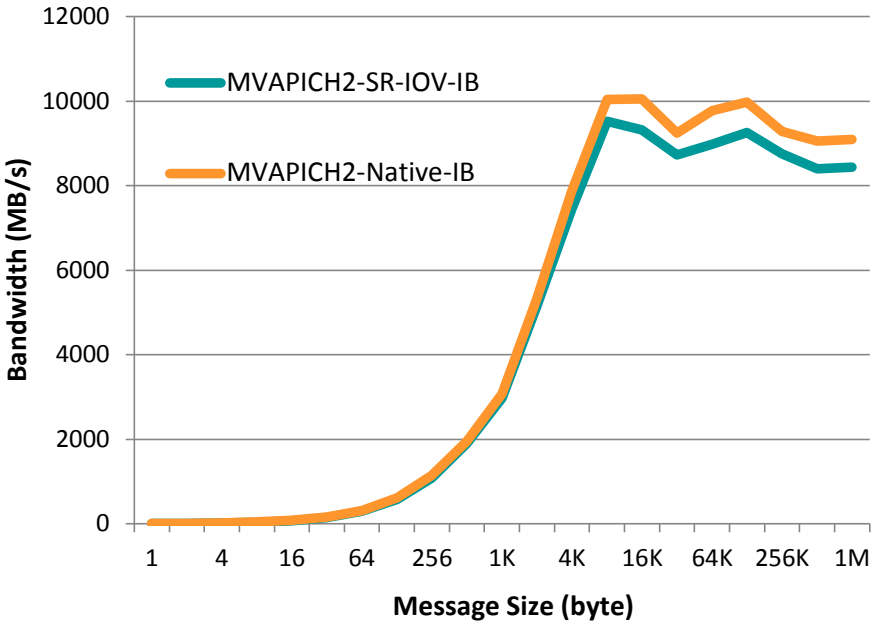
- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Extremely minimum memory footprint
- Support for GPGPUs
- Support for Intel MICs
- Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, …) with Unified Runtime
- Virtualization

Can HPC and Virtualization be Combined?

- Virtualization has many benefits
  - Job migration
  - Compaction
- Not very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters
- Initial designs of MVAPICH2 with SR-IOV support

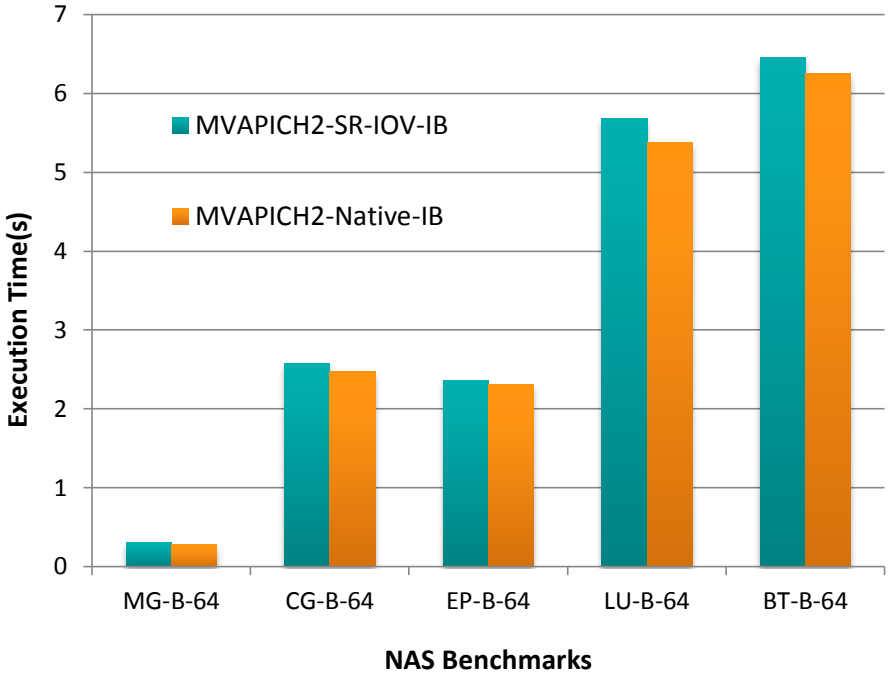
Intra-node Inter-VM Point-to-Point Latency and Bandwidth

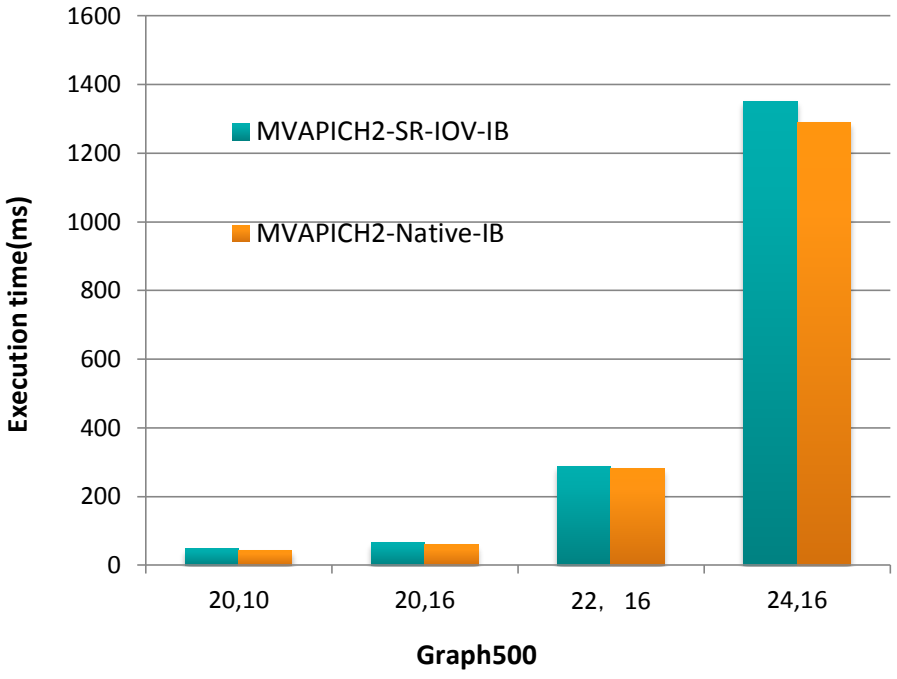




- 1 VM per Core
- MVAPICH2-SR-IOV-IB brings only 3-7% (latency) and 3-8%(BW) overheads, compared to MAPICH2 over Native InfiniBand Verbs (MVAPICH2-Native-IB)

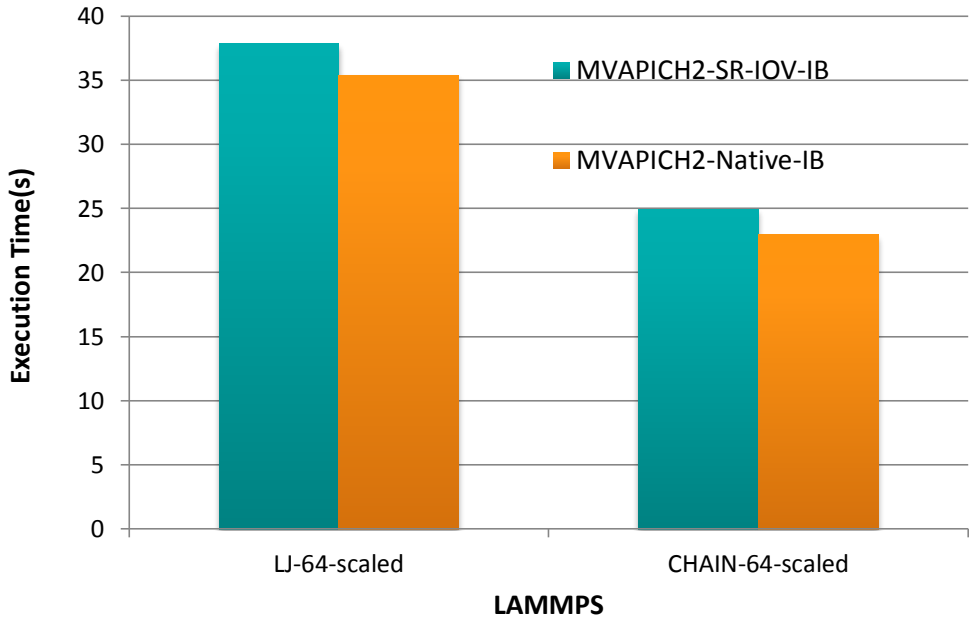
Performance Evaluations with NAS and Graph500





- 8 VMs across 4 nodes, 1 VM per Socket, 64 cores totally
- MVAPICH2-SR-IOV-IB brings 3-7% and 3-9% overheads for NAS Benchmarks and Graph500, respectively, compared to MVAPICH2-Native-IB

Performance Evaluation with LAMMPS



- 8 VMs across 4 nodes, 1 VM per Socket, 64 cores totally
- MVAPICH2-SR-IOV-IB brings 7% and 9% overheads for LJ and CHAIN in LAMMPS, respectively, compared to MVAPICH2-Native-IB

J. Zhang, X. Lu, J. Jose, R. Shi and Dhabaleswar K. (DK) Panda, Can Inter-VM Shmem Benefit MPI Applications on SR-IOV based Virtualized InfiniBand Clusters?, EuroPar 2014, August 2014

J. Zhang, X. Lu, J. Jose, R. Shi, M. Li and Dhabaleswar K. (DK) Panda, High Performance MPI Library over SR-IOV Enabled InfiniBand Clusters, HiPC '14, Dec. 14

MVAPICH2/MVPICH2-X – Plans for Exascale

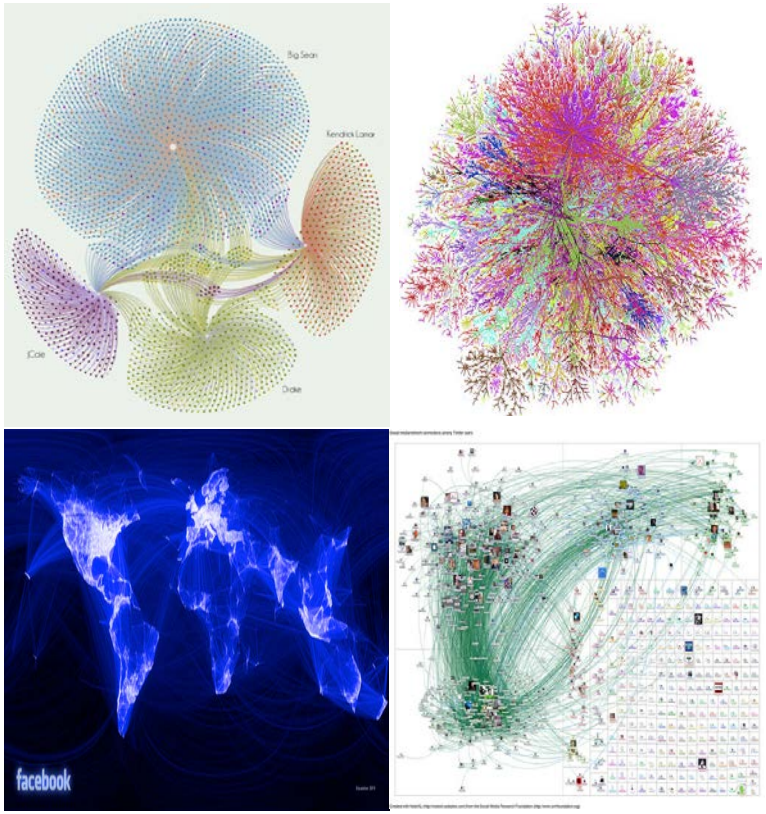
- Performance and Memory scalability toward 900K-1M cores
  - Dynamically Connected Transport (DCT) service with Connect-IB
- Enhanced Optimization for GPGPU and Coprocessor Support
  - Extending the GPGPU support (GPU-Direct RDMA) with CUDA 6.5 and Beyond
  - Support for Intel MIC (Knight Landing)
- Taking advantage of Collective Offload framework
  - Including support for non-blocking collectives (MPI 3.0)
- RMA support (as in MPI 3.0)
- Extended topology-aware collectives
- Power-aware collectives
- Support for MPI Tools Interface (as in MPI 3.0)
- Checkpoint-Restart and migration support with in-memory checkpointing
- Hybrid MPI+PGAS programming support with GPGPUs and Accelerators
- High Performance Virtualization Support

Two Major Categories of Applications

- Scientific Computing
  - Message Passing Interface (MPI), including MPI + OpenMP, is the Dominant Programming Model
  - Many discussions towards Partitioned Global Address Space (PGAS)
    - UPC, OpenSHMEM, CAF, etc.
  - Hybrid Programming: MPI + PGAS (OpenSHMEM, UPC)
- Big Data/Enterprise/Commercial Computing
  - Focuses on large data and data analysis
  - Hadoop (HDFS, HBase, MapReduce)
  - Spark is emerging for in-memory computing
  - Memcached is also used for Web 2.0
- Applications can run on a single-site or across sites over WAN

Introduction to Big Data Applications and Analytics

- Big Data has become the one of the most important elements of business analytics
- Provides groundbreaking opportunities for enterprise information management and decision making
- The amount of data is exploding; companies are capturing and digitizing more information than ever
- The rate of information growth appears to be exceeding Moore's Law



- Commonly accepted **3V**'s of Big Data
  - **Volume, Velocity, Variety**

  Michael Stonebraker: Big Data Means at Least Three Different Things, http://www.nist.gov/itl/ssd/is/upload/NIST-stonebraker.pdf
- **5V**'s of Big Data – **3V** + **Value, Veracity**

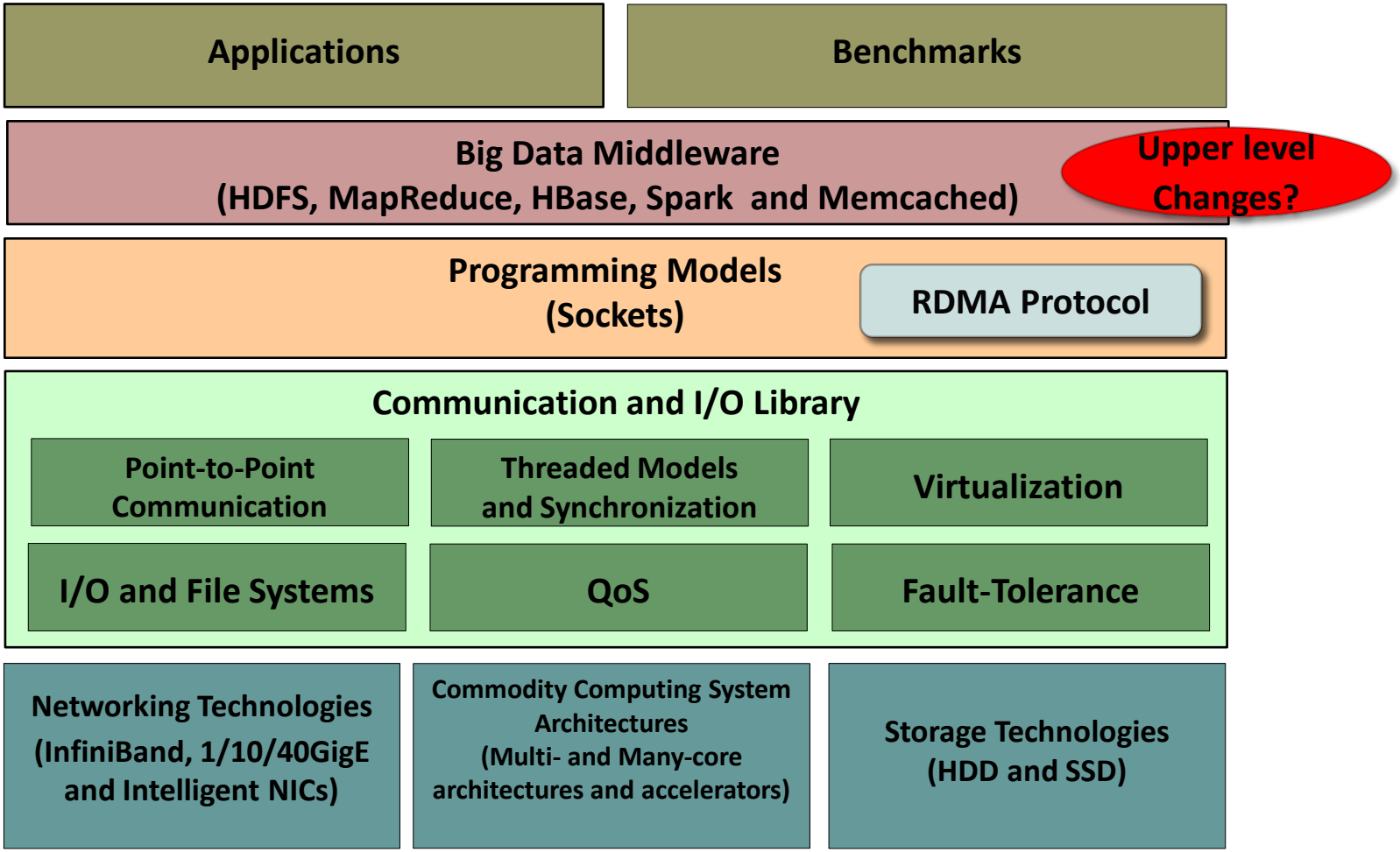
Can High-Performance Interconnects Benefit Big Data Middleware?

- Most of the current Big Data middleware use Ethernet infrastructure with Sockets
- Concerns for performance and scalability
- Usage of high-performance networks is beginning to draw interest from many companies
- What are the challenges?
- Where do the bottlenecks lie?
- Can these bottlenecks be alleviated with new designs (similar to the designs adopted for MPI)?
- Can HPC Clusters with high-performance networks be used for Big Data middleware?
- Initial focus: Hadoop, HBase, Spark and Memcached

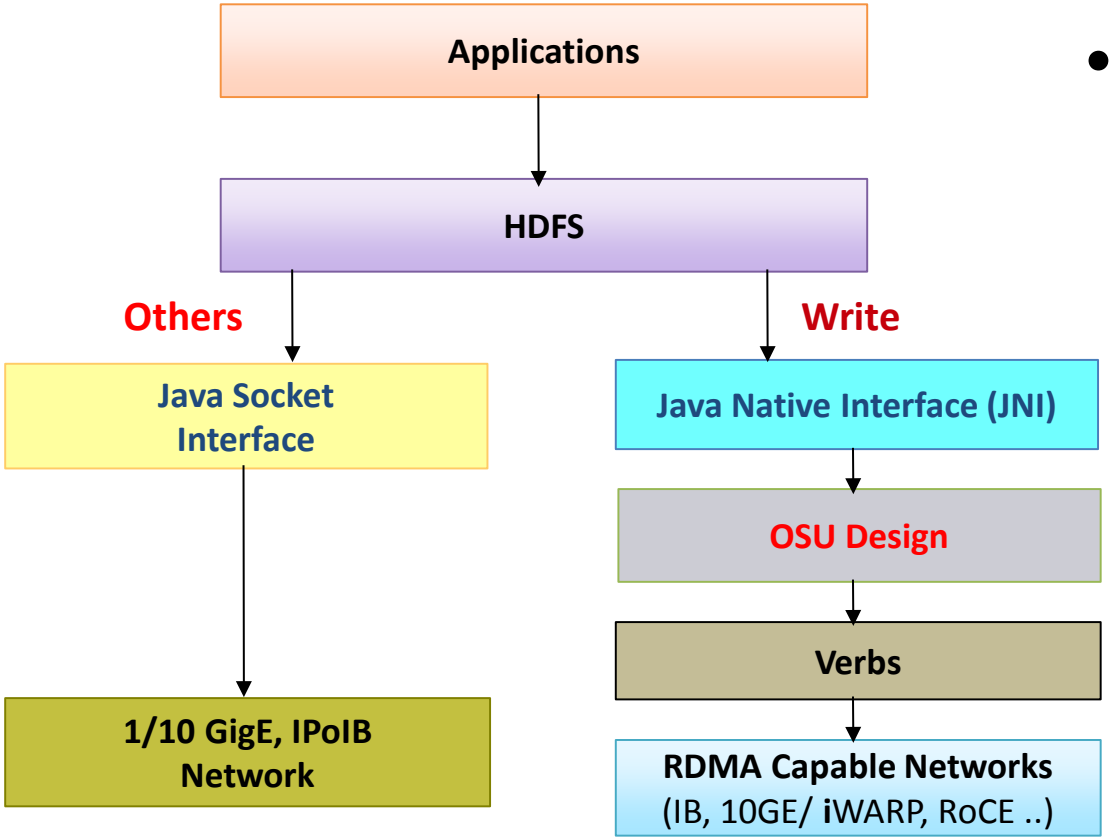
Overview of Presentation

- Big Data Processing
  - RDMA-based designs for Apache Hadoop
    - Case studies with HDFS, RPC and MapReduce
    - RDMA-based MapReduce on HPC Clusters with Lustre
  - RDMA-based design for Apache Spark
  - HiBD Project and Releases

Designing Communication and I/O Libraries for Big Data Systems: Solved a Few Initial Challenges



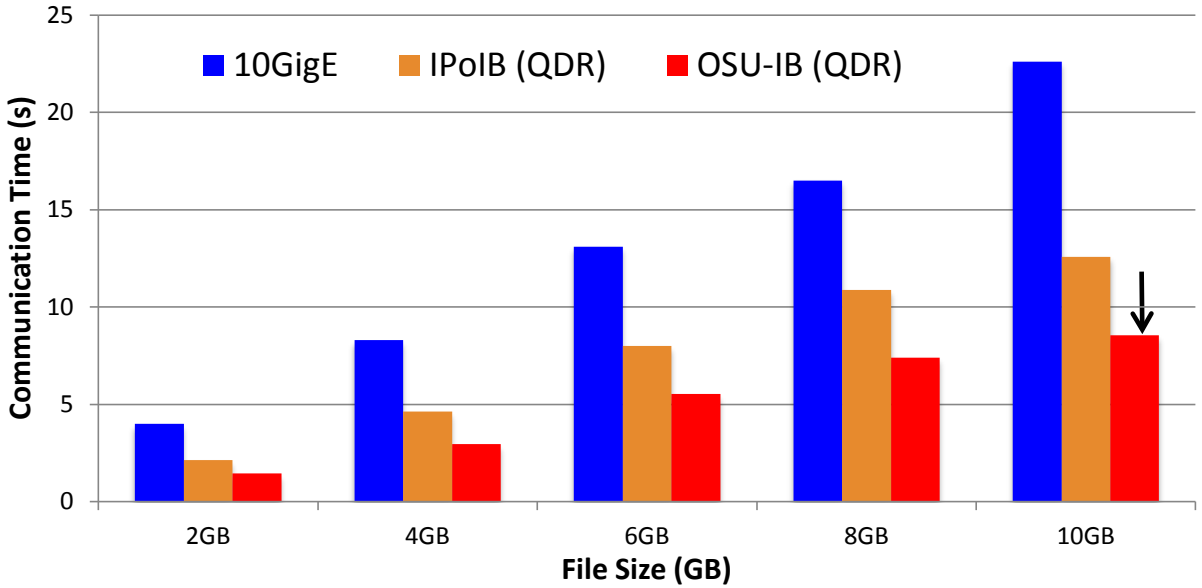
Design Overview of HDFS with RDMA



- Design Features
  - RDMA-based HDFS write
  - RDMA-based HDFS replication
  - Parallel replication support
  - On-demand connection setup
  - InfiniBand/RoCE support

- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Java based HDFS with communication library written in native code

Communication Times in HDFS

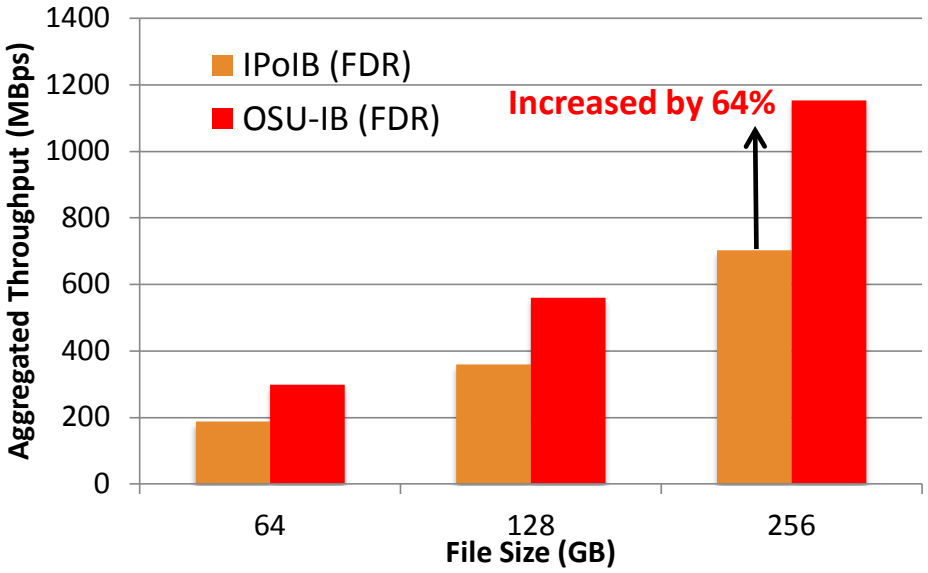


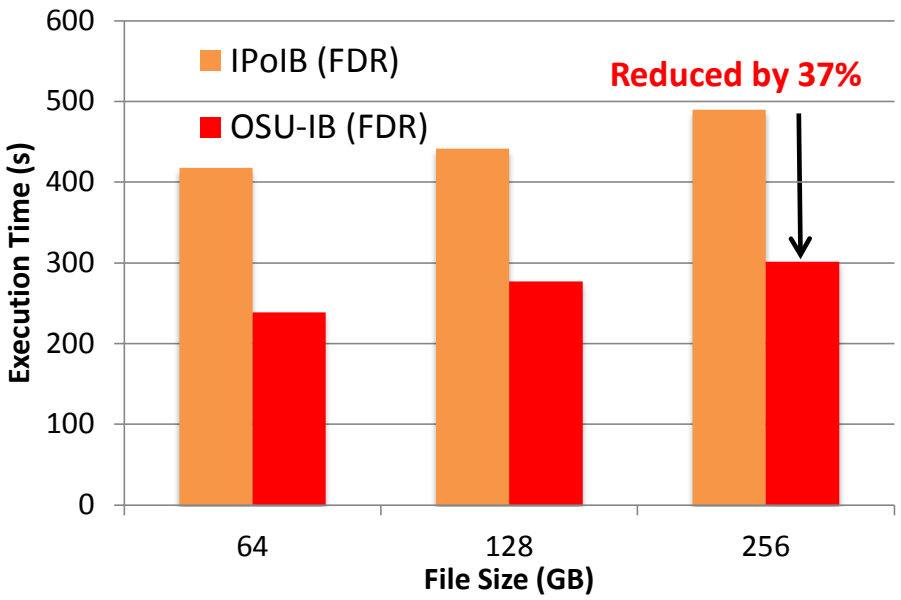
- Cluster with HDD DataNodes
  - **30%** improvement in communication time over IPoIB (QDR)
  - **56%** improvement in communication time over 10GigE
- Similar improvements are obtained for SSD DataNodes

N. S. Islam, M. W. Rahman, J. Jose, R. Rajachandrasekar, H. Wang, H. Subramoni, C. Murthy and D. K. Panda , High Performance RDMA-Based Design of HDFS over InfiniBand , Supercomputing (SC), Nov 2012

N. Islam, X. Lu, W. Rahman, and D. K. Panda, SOR-HDFS: A SEDA-based Approach to Maximize Overlapping in RDMA-Enhanced HDFS, HPDC '14, June 2014

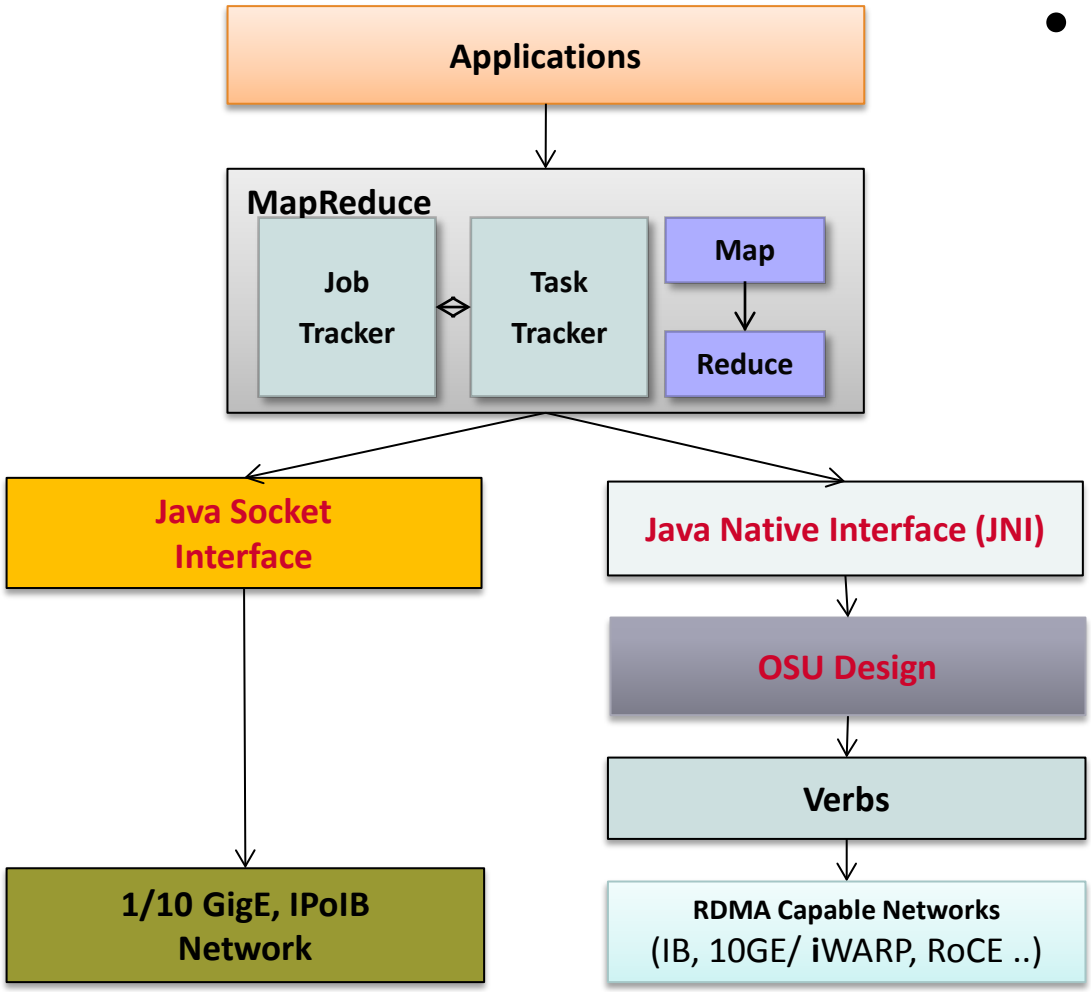
Evaluations using Enhanced DFSIO of Intel HiBench on TACC-Stampede





- Cluster with 64 DataNodes (1K cores), single HDD per node
  - **64%** improvement in throughput over IPoIB (FDR) for 256GB file size
  - **37%** improvement in latency over IPoIB (FDR) for 256GB file size

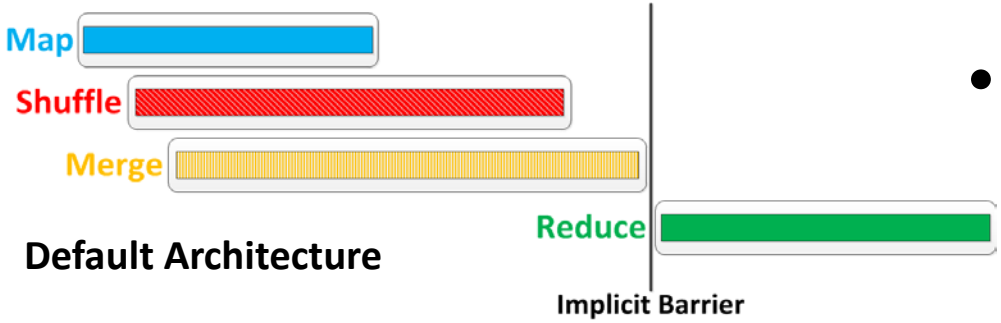
Design Overview of MapReduce with RDMA



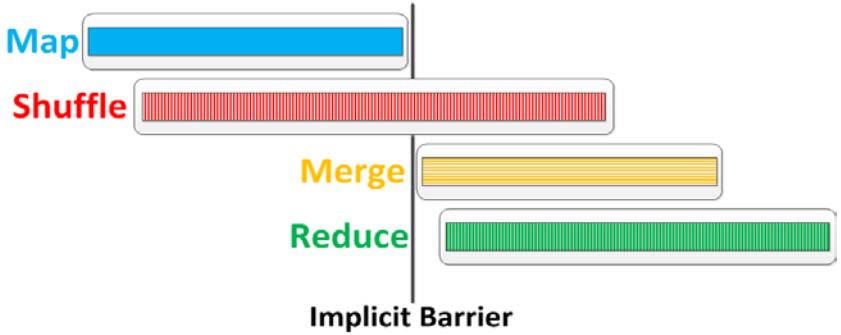
- Design Features
  - RDMA-based shuffle
  - Prefetching and caching map output
  - Efficient Shuffle Algorithms
  - In-memory merge
  - On-demand Shuffle Adjustment
  - Advanced overlapping
    - map, shuffle, and merge
    - shuffle, merge, and reduce
  - On-demand connection setup
  - InfiniBand/RoCE support

- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Java based MapReduce with communication library written in native code

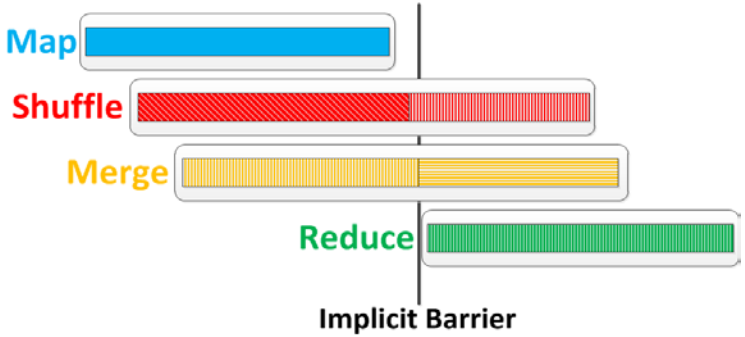
Advanced Overlapping among different phases



Default Architecture



Enhanced Overlapping

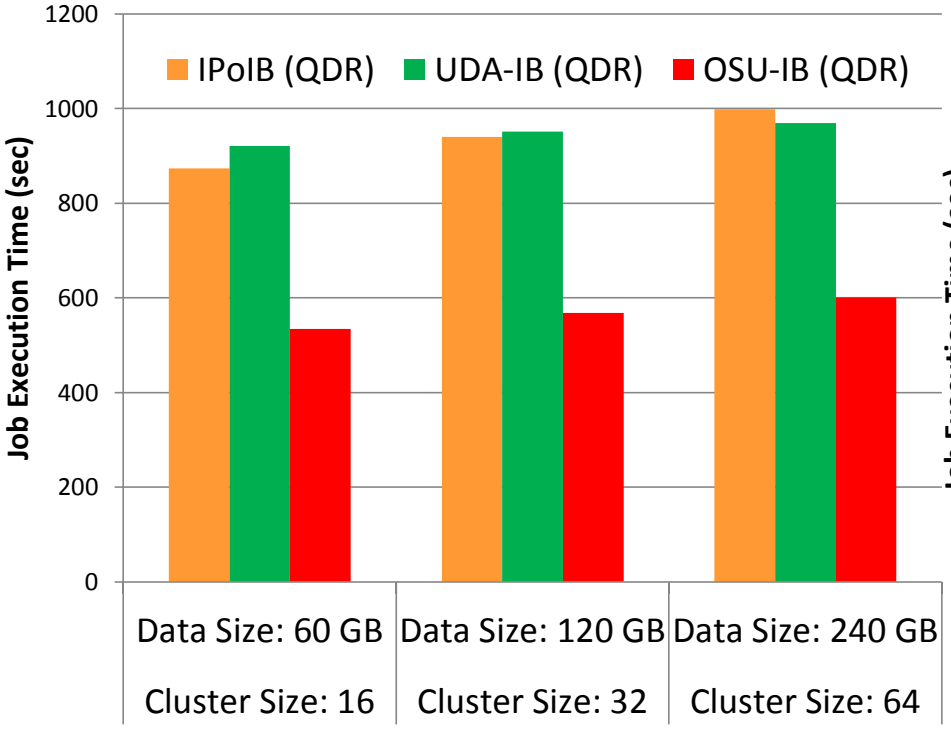


Advanced Overlapping

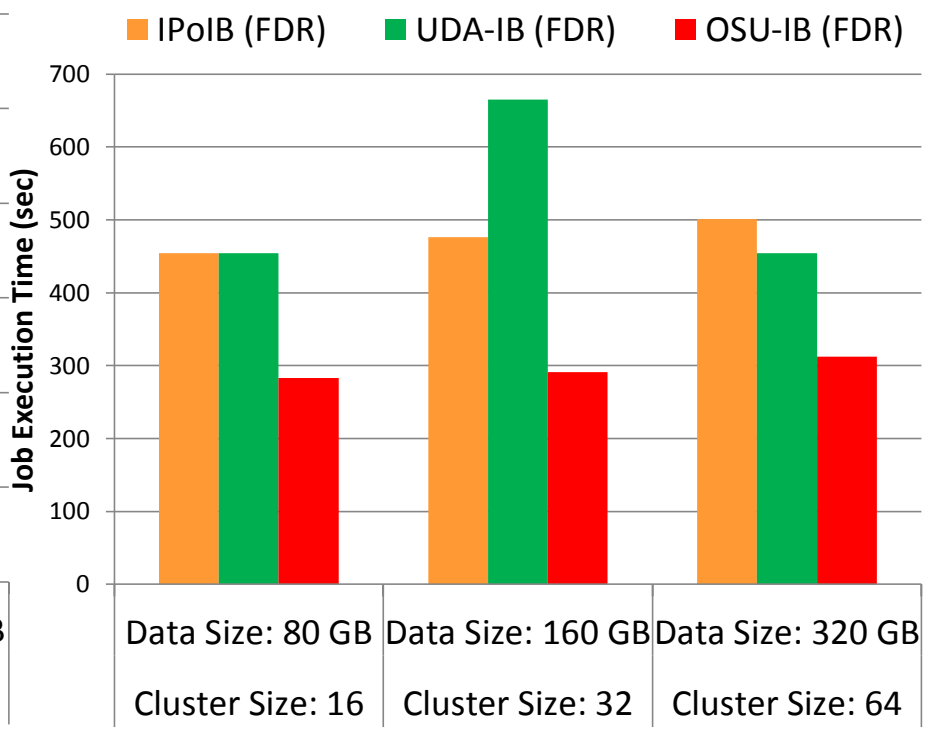
- A hybrid approach to achieve maximum possible overlapping in MapReduce across all phases compared to other approaches
  - Efficient Shuffle Algorithms
  - Dynamic and Efficient Switching
  - On-demand Shuffle Adjustment

M. W. Rahman, X. Lu, N. S. Islam, and D. K. Panda, HOMR: A Hybrid Approach to Exploit Maximum Overlapping in MapReduce over High Performance Interconnects, ICS, June 2014.

Performance Evaluation of Sort and TeraSort



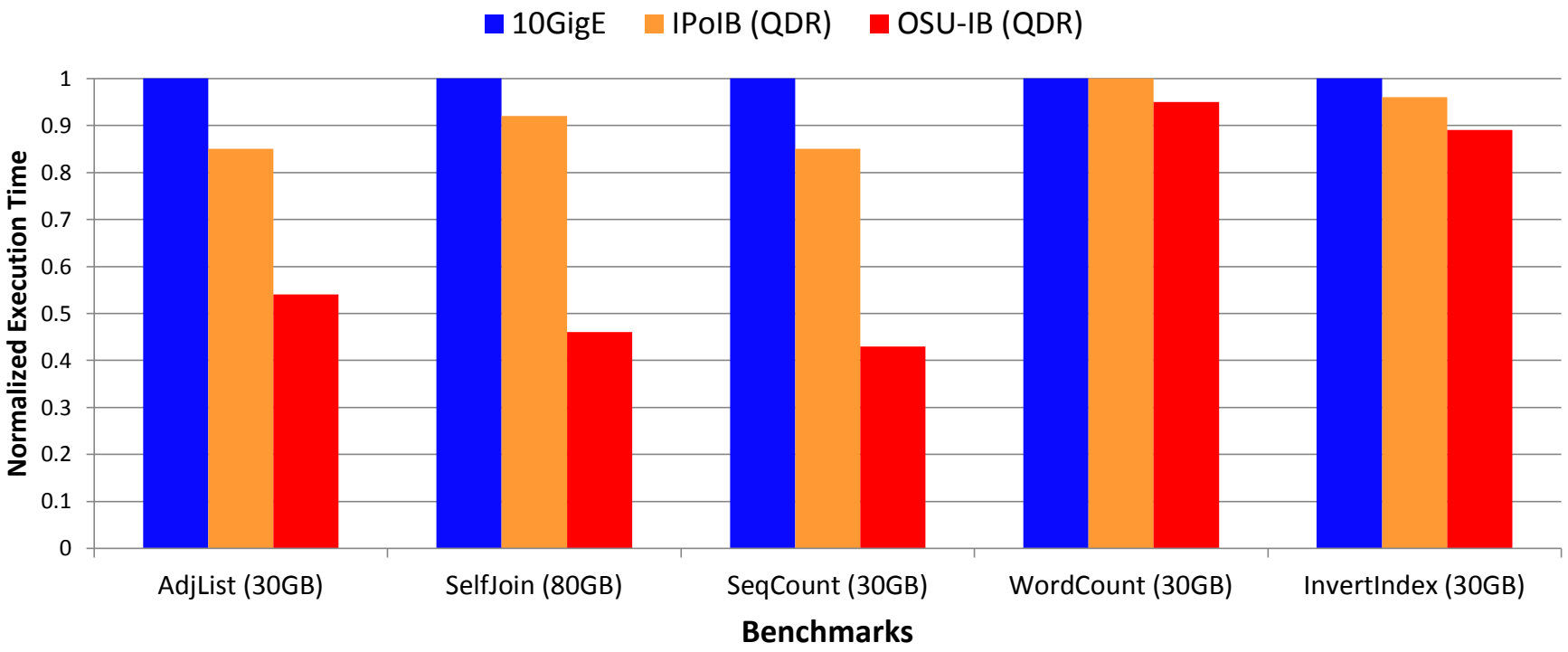
Sort in OSU Cluster



TeraSort in TACC Stampede

- For 240GB Sort in 64 nodes (512 cores)
  - 40% improvement over IPoIB (QDR) with HDD used for HDFS
- For 320GB TeraSort in 64 nodes (1K cores)
  - 38% improvement over IPoIB (FDR) with HDD used for HDFS

Evaluations using PUMA Workload

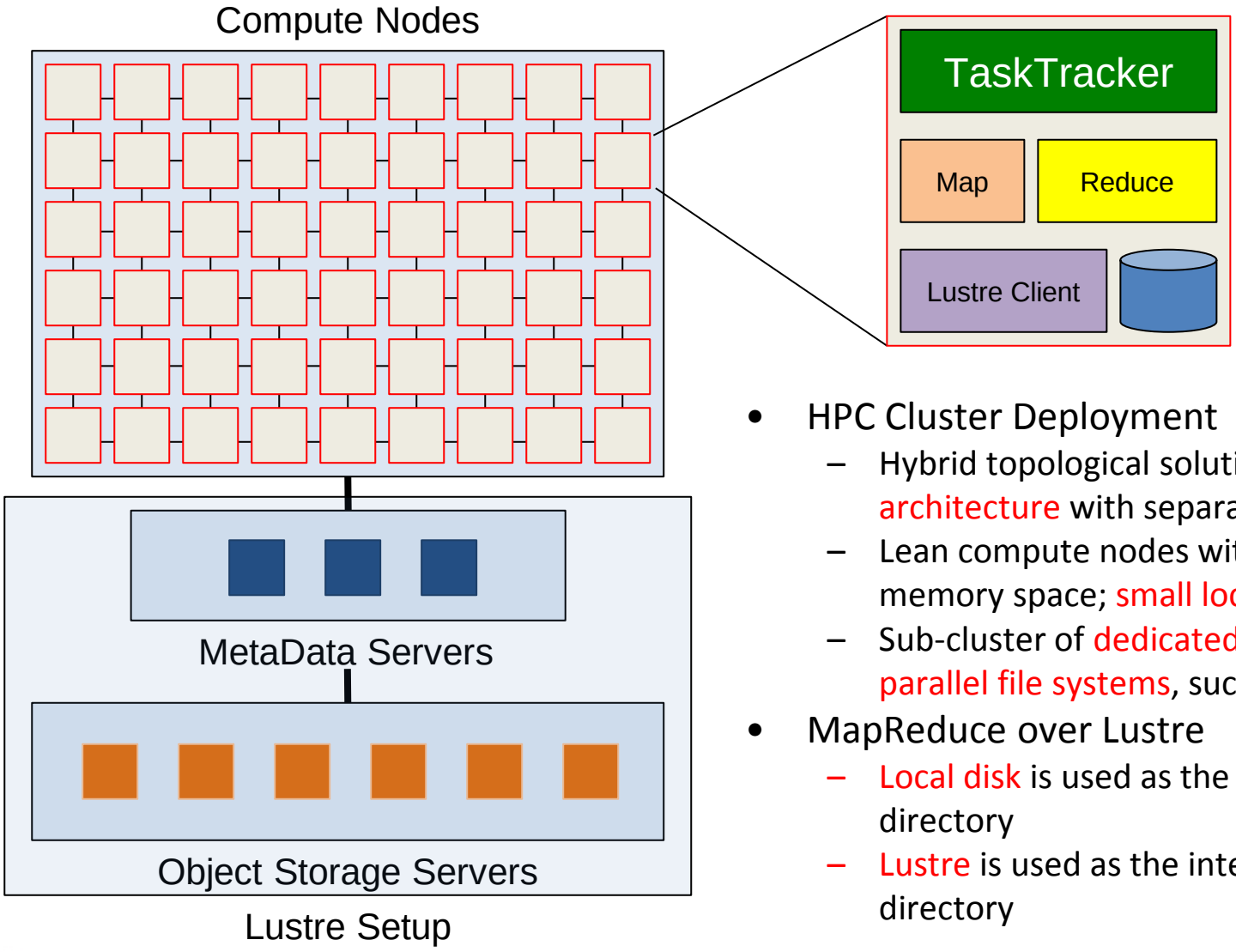


- 50% improvement in Self Join over IPoIB (QDR) for 80 GB data size
- 49% improvement in Sequence Count over IPoIB (QDR) for 30 GB data size

Overview of Presentation

- Big Data Processing
  - RDMA-based designs for Apache Hadoop
    - Case studies with HDFS, RPC and MapReduce
    - RDMA-based MapReduce on HPC Clusters with Lustre
  - RDMA-based design for Apache Spark
  - HiBD Project and Releases

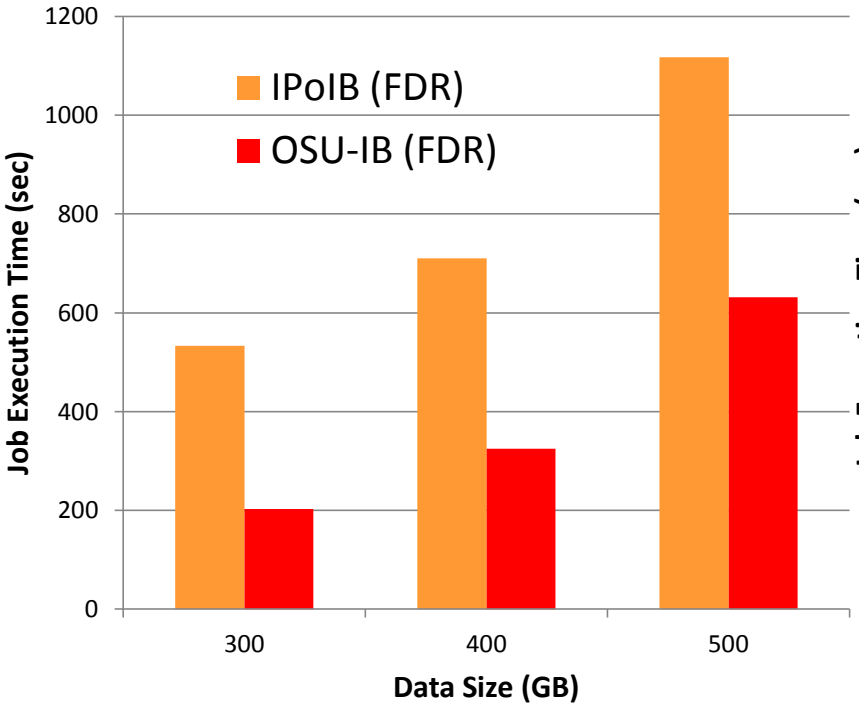
Optimized Apache Hadoop over Parallel File Systems

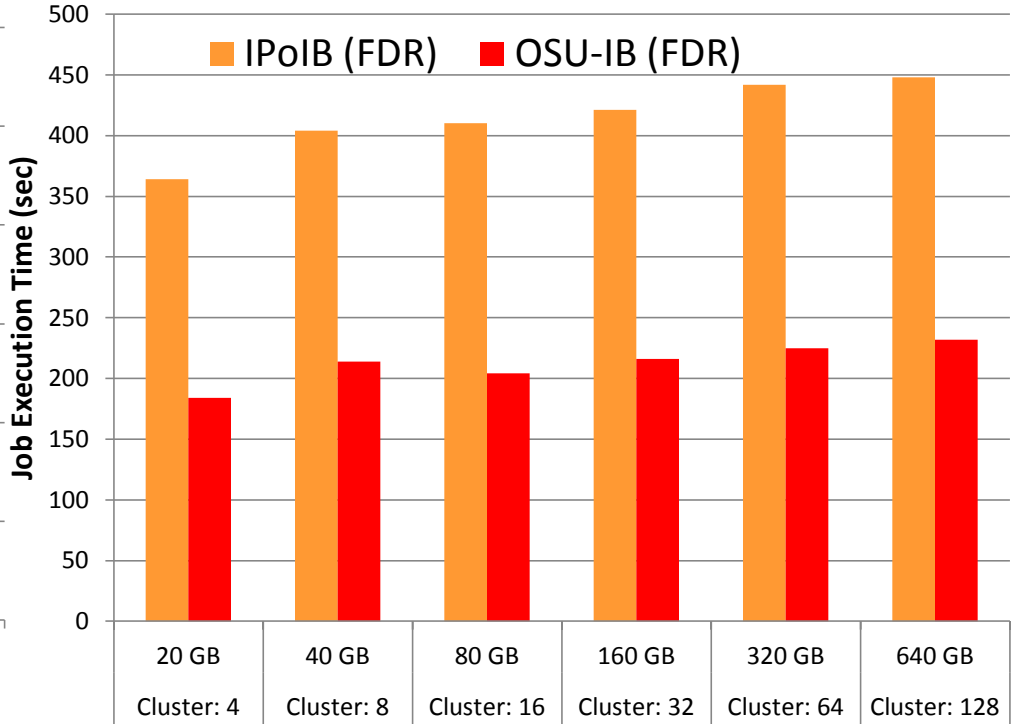


- HPC Cluster Deployment
  - Hybrid topological solution of Beowulf architecture with separate I/O nodes
  - Lean compute nodes with light OS; more memory space; small local storage
  - Sub-cluster of dedicated I/O nodes with parallel file systems, such as Lustre
- MapReduce over Lustre
  - Local disk is used as the intermediate data directory
  - Lustre is used as the intermediate data directory

Case Study - Performance Improvement of RDMA-MapReduce over Lustre on TACC-Stampede

- **Local disk is used as the intermediate data directory**



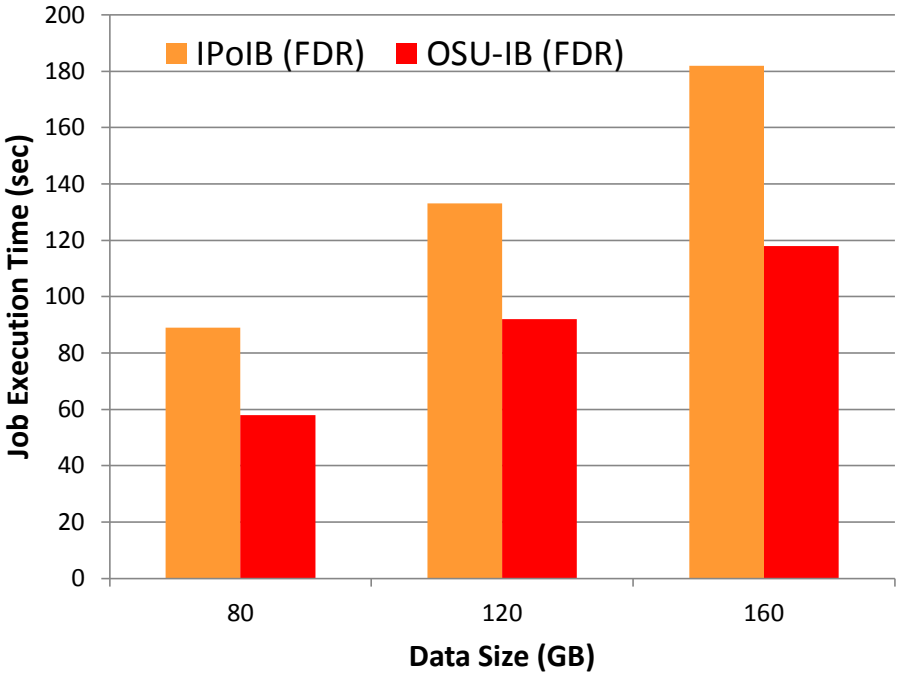


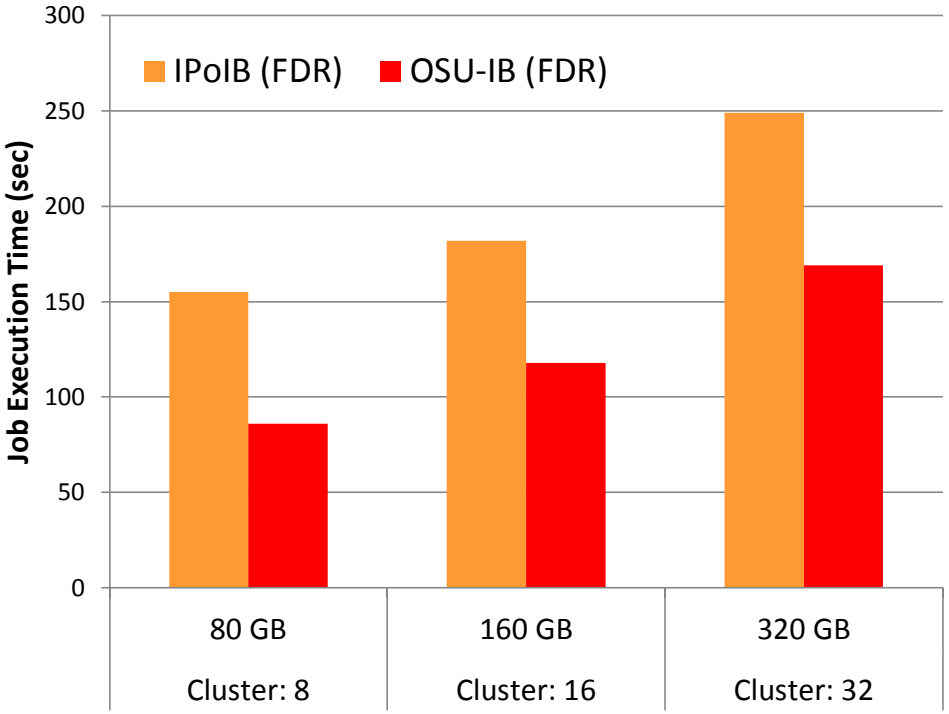
- For 500GB Sort in 64 nodes
  - 44% improvement over IPoIB (FDR)
- For 640GB Sort in 128 nodes
  - 48% improvement over IPoIB (FDR)

M. W. Rahman, X. Lu, N. S. Islam, R. Rajachandrasekar, and D. K. Panda, MapReduce over Lustre: Can RDMA-based Approach Benefit?, Euro-Par, August 2014.

Case Study - Performance Improvement of RDMA-MapReduce over Lustre on TACC-Stampede

- **Lustre is used as the intermediate data directory**





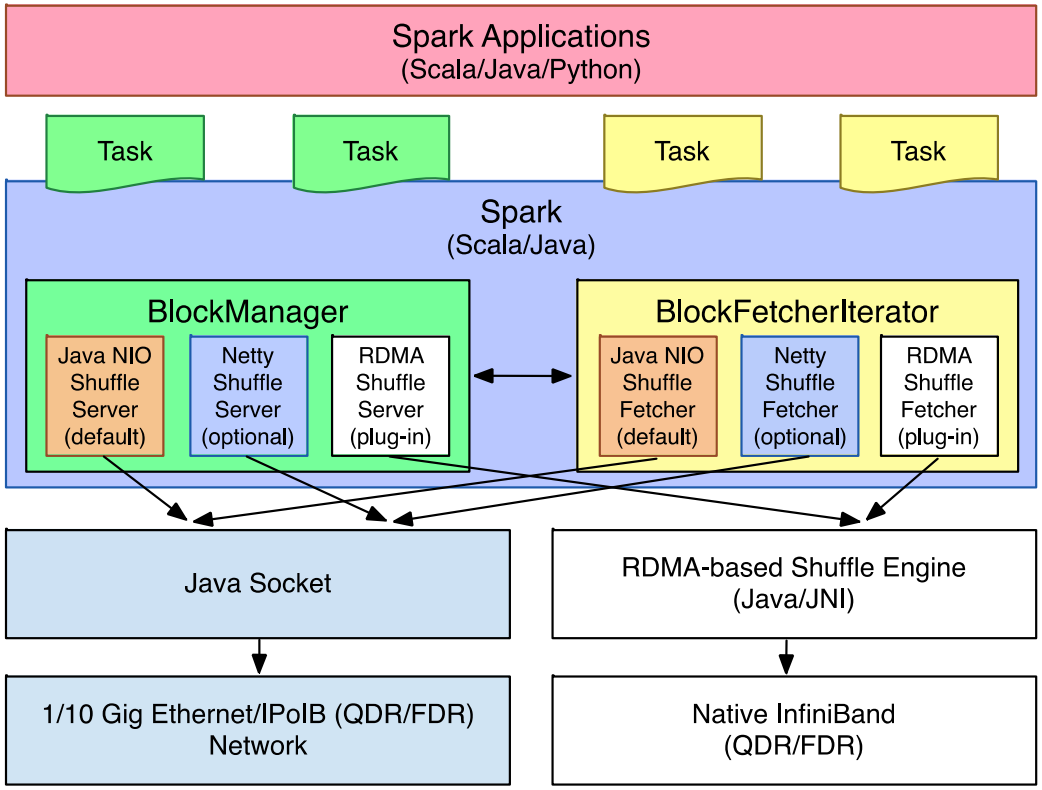
- For 160GB Sort in 16 nodes
  - 35% improvement over IPoIB (FDR)
- For 320GB Sort in 32 nodes
  - 33% improvement over IPoIB (FDR)

- **Can more optimizations be achieved by leveraging more features of Lustre?**

Overview of Presentation

- Big Data Processing
  - RDMA-based designs for Apache Hadoop
    - Case studies with HDFS, RPC and MapReduce
    - RDMA-based MapReduce on HPC Clusters with Lustre
  - RDMA-based design for Apache Spark
  - HiBD Project and Releases

Design Overview of Spark with RDMA

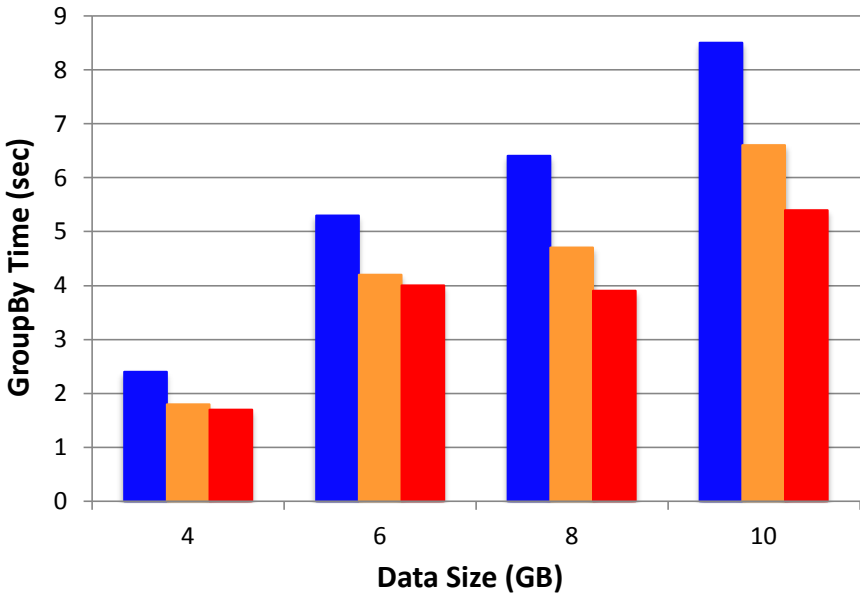


- Design Features
  - RDMA based shuffle
  - SEDA-based plugins
  - Dynamic connection management and sharing
  - Non-blocking and out-of-order data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

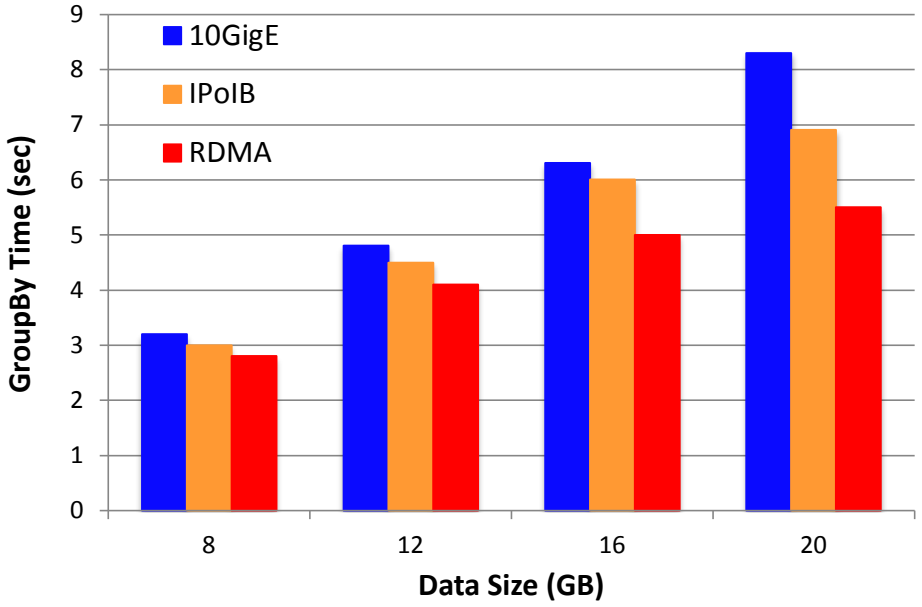
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

Preliminary Results of Spark-RDMA Design - GroupBy



Cluster with 4 HDD Nodes, GroupBy with 32 cores



Cluster with 8 HDD Nodes, GroupBy with 64 cores

- Cluster with 4 HDD Nodes, single disk per node, 32 concurrent tasks
  - **18%** improvement over IPoIB (QDR) for 10GB data size
- Cluster with 8 HDD Nodes, single disk per node, 64 concurrent tasks
  - **20%** improvement over IPoIB (QDR) for 20GB data size

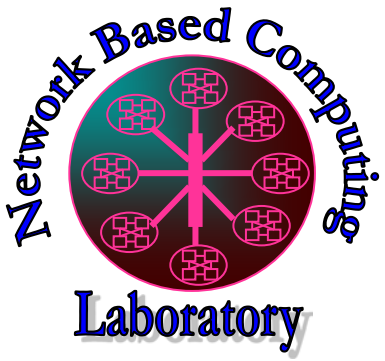
Overview of Presentation

- Big Data Processing
  - RDMA-based designs for Apache Hadoop
    - Case studies with HDFS, RPC and MapReduce
    - RDMA-based MapReduce on HPC Clusters with Lustre
  - RDMA-based design for Apache Spark
  - HiBD Project and Releases

The High-Performance Big Data (HiBD) Project

- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- RDMA for Memcached (RDMA-Memcached)
- OSU HiBD-Benchmarks (OHB)
- **http://hibd.cse.ohio-state.edu**
- Users Base: 76 organizations from 13 countries
- RDMA for Apache HBase and Spark

High-Performance Big Data



THE OHIO STATE UNIVERSITY

RDMA for Apache Hadoop 1.x/2.x Distributions

- High-Performance Design of Hadoop over RDMA-enabled Interconnects
  - High performance design with native InfiniBand and RoCE support at the verbs-level for HDFS, MapReduce, and RPC components
  - Easily configurable for native InfiniBand, RoCE and the traditional sockets-based support (Ethernet and InfiniBand with IPoIB)
- Current release: 0.9.9/0.9.1
  - Based on Apache Hadoop 1.2.1/2.4.1
  - Compliant with Apache Hadoop 1.2.1/2.4.1 APIs and applications
  - Tested with
    - Mellanox InfiniBand adapters (DDR, QDR and FDR)
    - RoCE support with Mellanox adapters
    - Various multi-core platforms
    - Different file systems with disks and SSDs
  - http://hibd.cse.ohio-state.edu

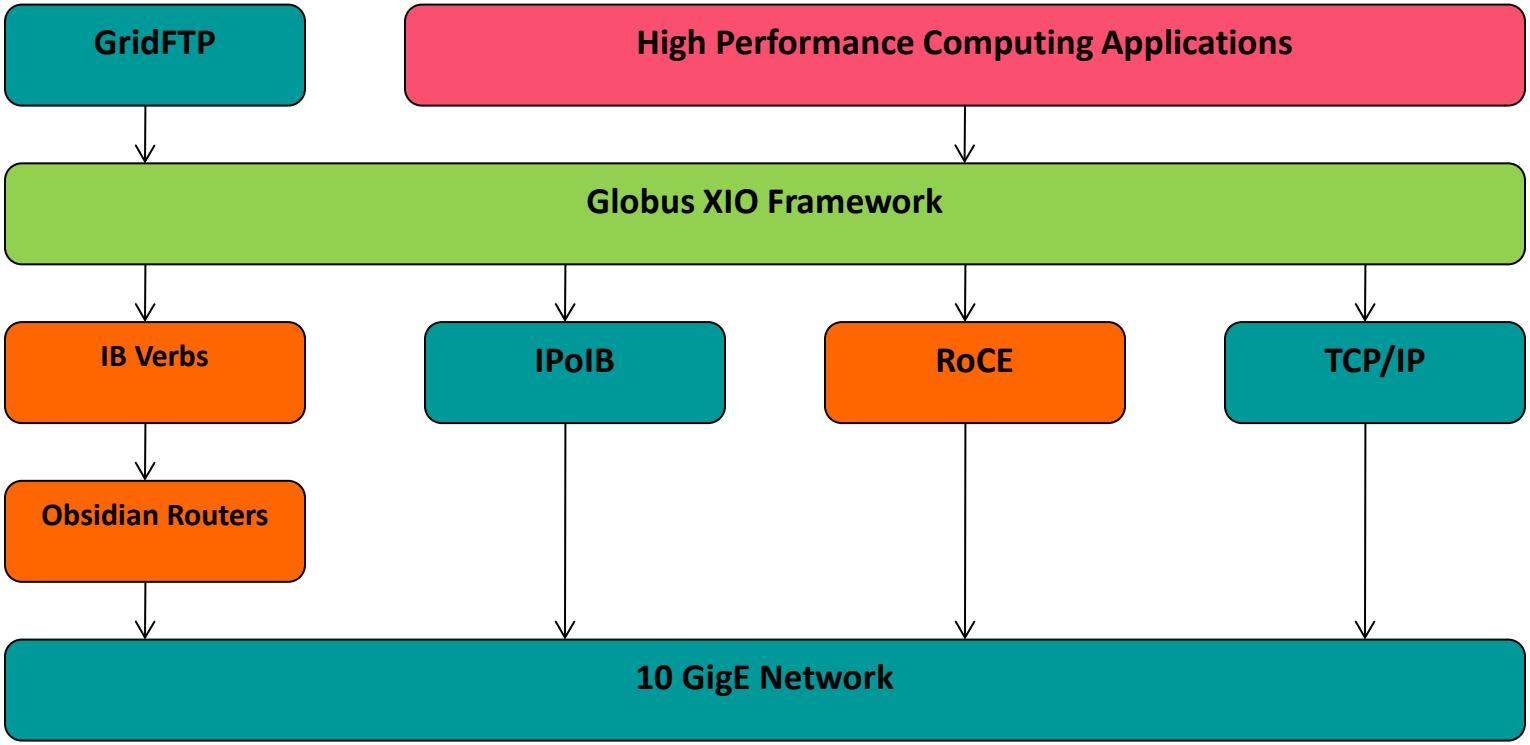
Future Plans of OSU High Performance Big Data (HiBD) Project

- Upcoming Releases of RDMA-enhanced Packages will support
  - Hadoop 2.x MapReduce & RPC
  - Spark
  - HBase
- Upcoming Releases of OSU HiBD Micro-Benchmarks (OHB) will support
  - HDFS
  - MapReduce
  - RPC
- Advanced designs with upper-level changes and optimizations
  - E.g. MEM-HDFS

Two Major Categories of Applications

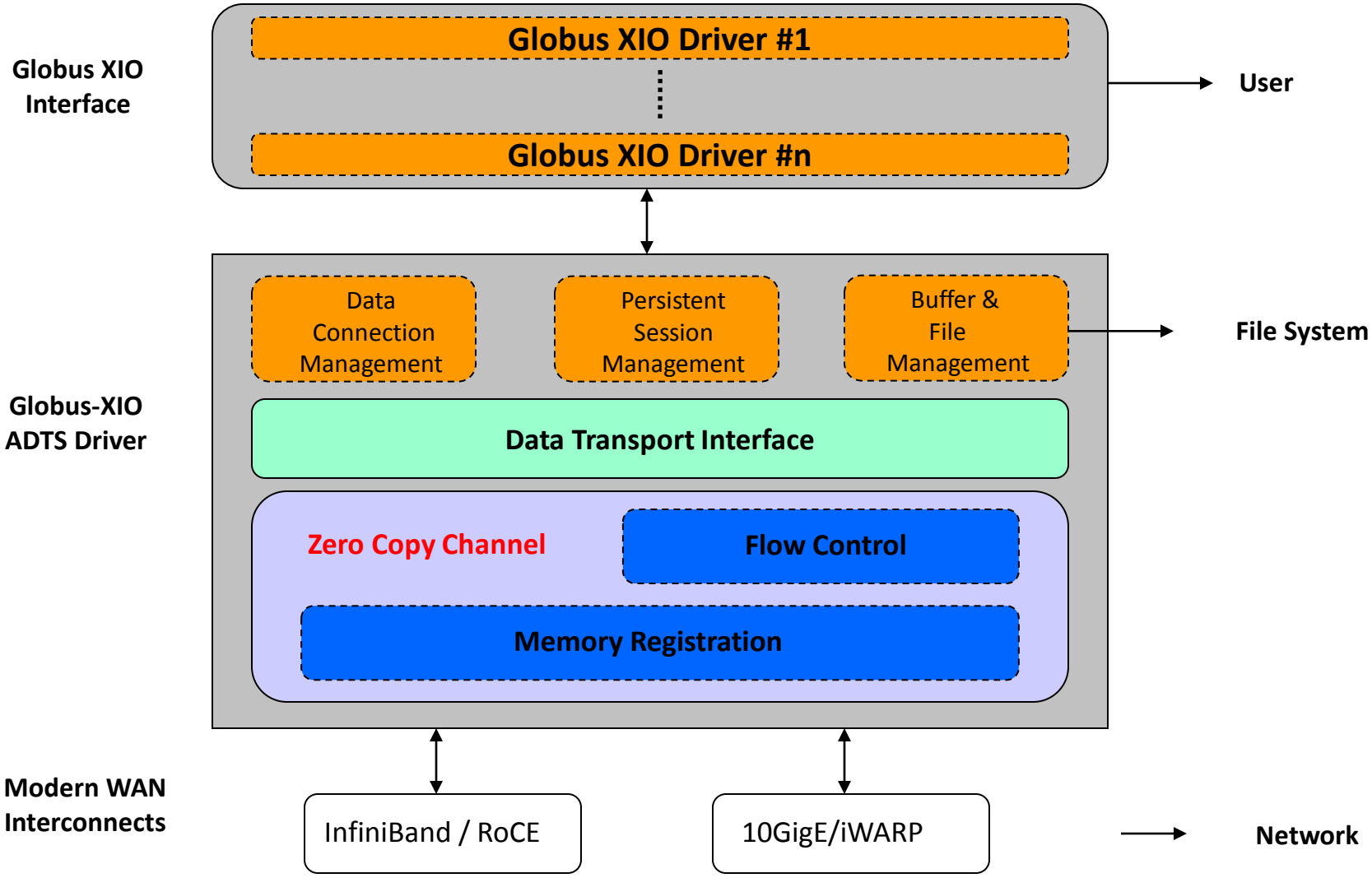
- Scientific Computing
  - Message Passing Interface (MPI), including MPI + OpenMP, is the Dominant Programming Model
  - Many discussions towards Partitioned Global Address Space (PGAS)
    - UPC, OpenSHMEM, CAF, etc.
  - Hybrid Programming: MPI + PGAS (OpenSHMEM, UPC)
- Big Data/Enterprise/Commercial Computing
  - Focuses on large data and data analysis
  - Hadoop (HDFS, HBase, MapReduce)
  - Spark is emerging for in-memory computing
  - Memcached is also used for Web 2.0
- Applications can run on a single-site or across sites over WAN

Communication Options in Grid



- Multiple options exist to perform data transfer on Grid
- Globus-XIO framework currently does not support IB natively
- We create the Globus-XIO ADTS driver and add native IB support to GridFTP

Globus-XIO Framework with ADTS Driver

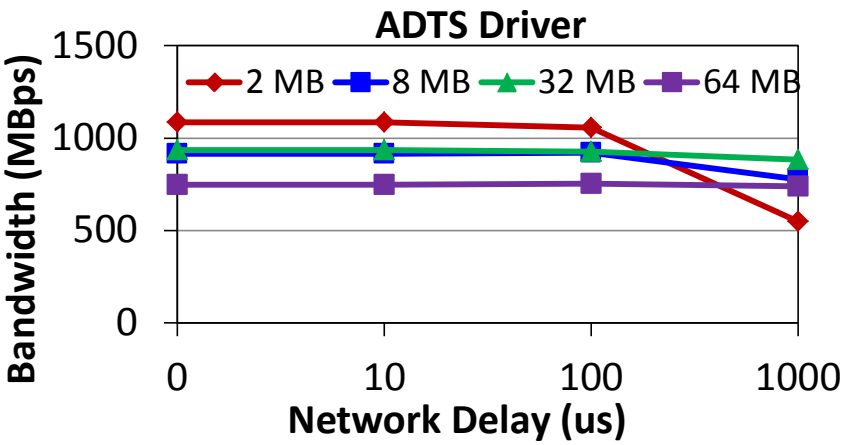


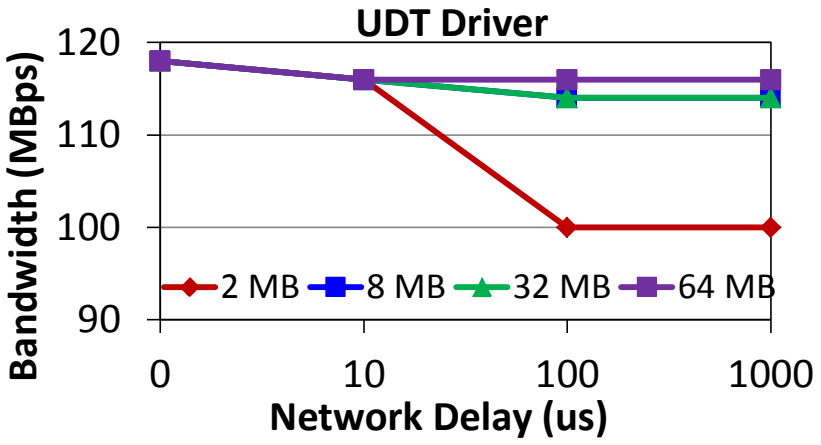
H. Subramoni, P. Lai, R. Kettimuthu and D. K. Panda, High Performance Data Transfer in Grid Environment Using GridFTP over InfiniBand, Int'l Symposium on Cluster Computing and the Grid (CCGrid), May 2010

P. Lai, H. Subramoni, S. Narravula, A. Mamidala and D. K. Panda, Designing Efficient FTP Mechanisms for High Performance Data-Transfer over InfiniBand, Intl Conference on Parallel Processing (ICPP '09), Sept. 2009.

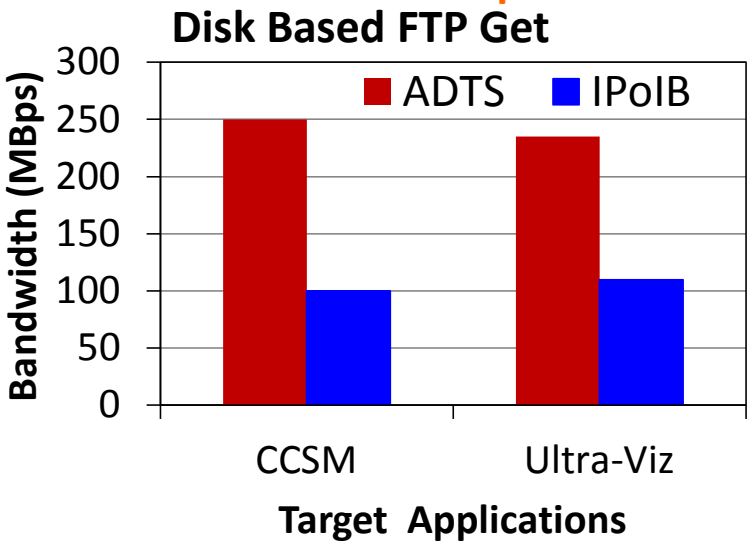
Performance Comparison of ADTS & UDT Drivers

In memory data transfer performance of ADTS & UDT drivers for different buffer sizes





ADTS based implementation is able to saturate the link bandwidth



- Community Climate System Model (CCSM)
  - Part of Earth System Grid Project
  - Transfers 160 TB in chunks of 256 MB
  - Network latency - 30 ms
- Ultra-Scale Visualization (Ultra-Viz)
  - Transfers files of size 2.6 GB
  - Network latency - 80 ms
- **The ADTS driver out performs the UDT driver (IPoIB) by more than 100%**

Concluding Remarks

- InfiniBand with RDMA feature is gaining momentum in HPC systems with best performance and greater usage
- As the HPC community moves to Exascale, new solutions are needed in the MPI and Hybrid MPI+PGAS stacks for supporting GPUs and Accelerators
- Demonstrated how such solutions can be designed with MVAPICH2/MVAPICH2-X and their performance benefits
- New solutions are also needed to re-design software libraries for Big Data environments to take advantage of RDMA
- Such designs will allow application scientists and engineers to take advantage of upcoming exascale systems

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students

- A. Bhat (M.S.)
- S. Chakraborthy (Ph.D.)
- N. Islam (Ph.D.)
- M. Li (Ph.D.)
- R. Rajachandrasekhar (Ph.D.)
- M. Rahman (Ph.D.)
- D. Shankar (Ph.D.)
- R. Shir (Ph.D.)
- A. Venkatesh (Ph.D.)
- J. Zhang (Ph.D.)

Current Senior Research Associate

- X. Lu
- H. Subramoni

Current Post-Docs

- K. Hamidouche
- J. Lin

Current Programmers

- M. Arnold
- J. Perkins

Past Students

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

Past Post-Docs

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

Past Research Scientist

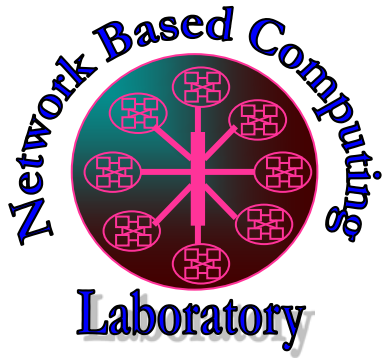
- S. Sur

Past Programmers

- D. Bureddy

Thank You!

panda@cse.ohio-state.edu



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

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

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