High Performance MPI Support in MVAPICH2 for InfiniBand Clusters

A Talk at NVIDIA Booth (SC '11)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

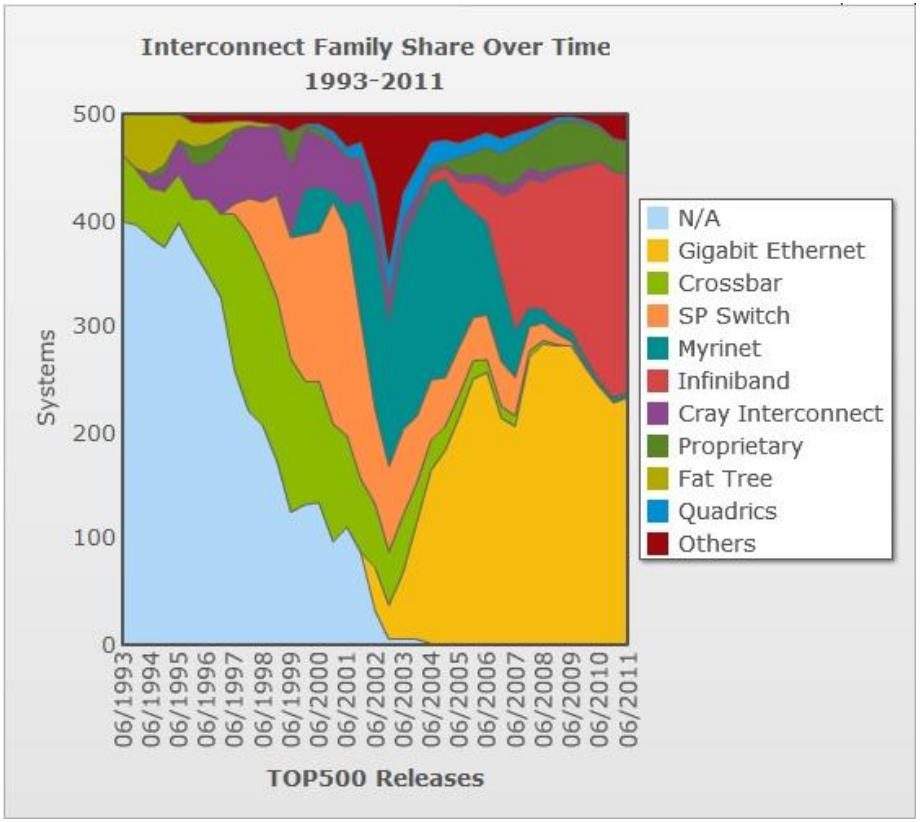
E-mail: panda@cse.ohio-state.edu

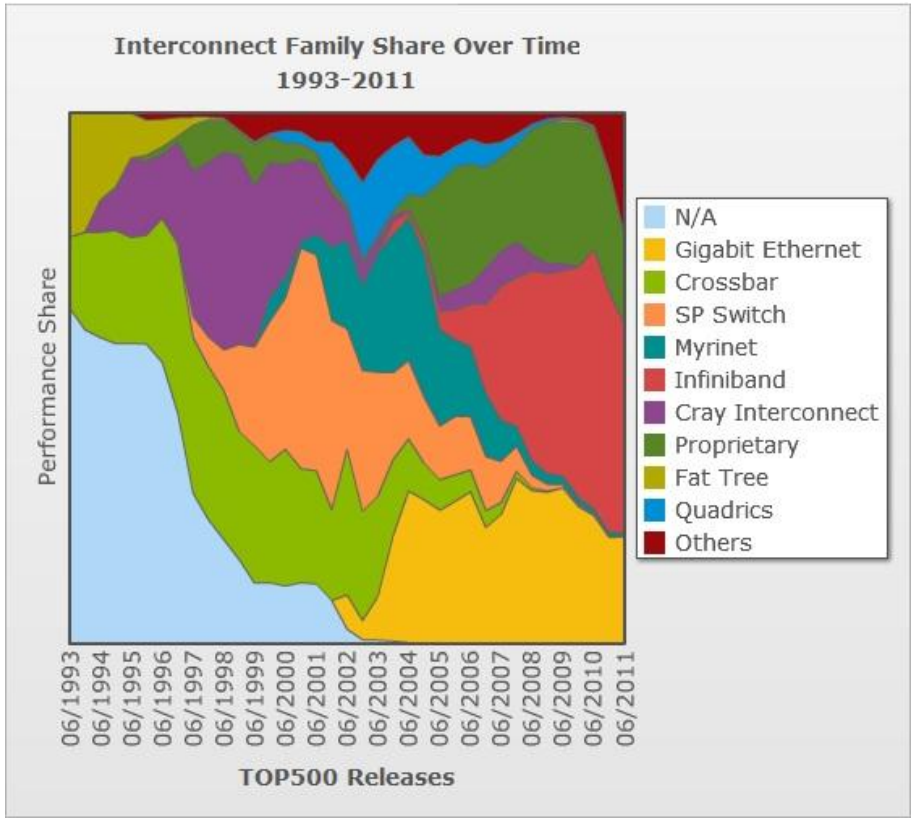
http://www.cse.ohio-state.edu/~panda

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

Nov. 1996: 0/500 (0%)	Jun. 2002: 80/500 (16%)	Nov. 2007: 406/500 (81.2%)
Jun. 1997: 1/500 (0.2%)	Nov. 2002: 93/500 (18.6%)	Jun. 2008: 400/500 (80.0%)
Nov. 1997: 1/500 (0.2%)	Jun. 2003: 149/500 (29.8%)	Nov. 2008: 410/500 (82.0%)
Jun. 1998: 1/500 (0.2%)	Nov. 2003: 208/500 (41.6%)	Jun. 2009: 410/500 (82.0%)
Nov. 1998: 2/500 (0.4%)	Jun. 2004: 291/500 (58.2%)	Nov. 2009: 417/500 (83.4%)
Jun. 1999: 6/500 (1.2%)	Nov. 2004: 294/500 (58.8%)	Jun. 2010: 424/500 (84.8%)
Nov. 1999: 7/500 (1.4%)	Jun. 2005: 304/500 (60.8%)	Nov. 2010: 415/500 (83%)
Jun. 2000: 11/500 (2.2%)	Nov. 2005: 360/500 (72.0%)	Jun. 2011: 411/500 (82.2%)
Nov. 2000: 28/500 (5.6%)	Jun. 2006: 364/500 (72.8%)	Nov. 2011: 410/500 (82.0%)
Jun. 2001: 33/500 (6.6%)	Nov. 2006: 361/500 (72.2%)	
Nov. 2001: 43/500 (8.6%)	Jun. 2007: 373/500 (74.6%)	

InfiniBand in the Top500





Percentage share of InfiniBand is steadily increasing

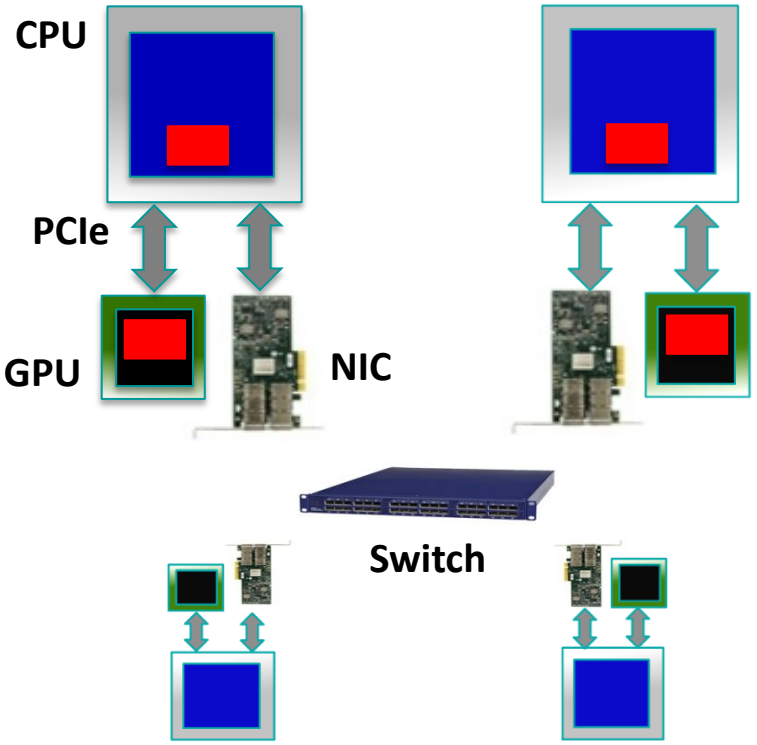
Large-scale InfiniBand Installations

- 209 IB Clusters (41.8%) in the November'11 Top500 list (http://www.top500.org)
- Installations in the Top 30 (13 systems):

120,640 cores (Nebulae) in China (4th)	29,440 cores (Mole-8.5) in China (21st)
73,278 cores (Tsubame-2.0) in Japan (5th)	42,440 cores (Red Sky) at Sandia (24th)
111,104 cores (Pleiades) at NASA Ames (7th)	62,976 cores (Ranger) at TACC (25th)
138,368 cores (Tera-100) at France (9th)	20,480 cores (Bull Benchmarks) in France (27th)
122,400 cores (RoadRunner) at LANL (10th)	20,480 cores (Helios) in Japan (28th)
137,200 cores (Sunway Blue Light) in China (14th)	***More are getting installed !***
46,208 cores (Zin) at LLNL (15th)	
33,072 cores (Lomonosov) in Russia (18th)	

Data movement in GPU+IB clusters

- Many systems today want to use systems that have both GPUs and high-speed networks such as InfiniBand
- Steps in Data movement in InfiniBand clusters with GPUs
  - From GPU device memory to main memory at source process, using CUDA
  - From source to destination process, using MPI
  - From main memory to GPU device memory at destination process, using CUDA
- Earlier, GPU device and InfiniBand device required separate memory registration
- GPU-Direct (collaboration between NVIDIA and Mellanox) supported common registration between these devices
- However, GPU-GPU communication is still costly and programming is harder



Sample Code - Without MPI integration

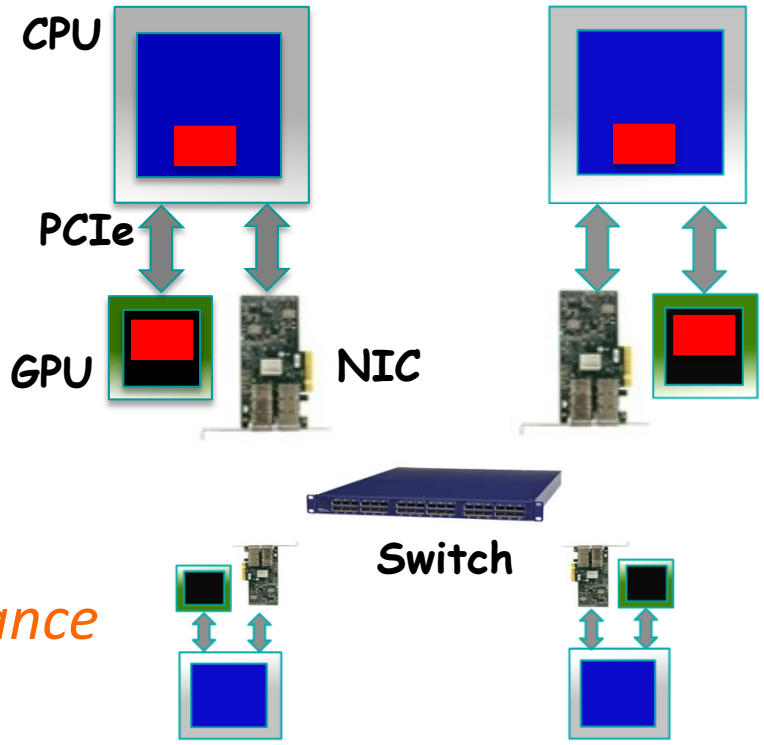
- Naïve implementation with standard MPI and CUDA

At Sender:

```
cudaMemcpy(sbuf, sdev, . . .);
MPI_Send(sbuf, size, . . .);
```

At Receiver:

```
MPI_Recv(rbuf, size, . . .);
cudaMemcpy(rdev, rbuf, . . .);
```



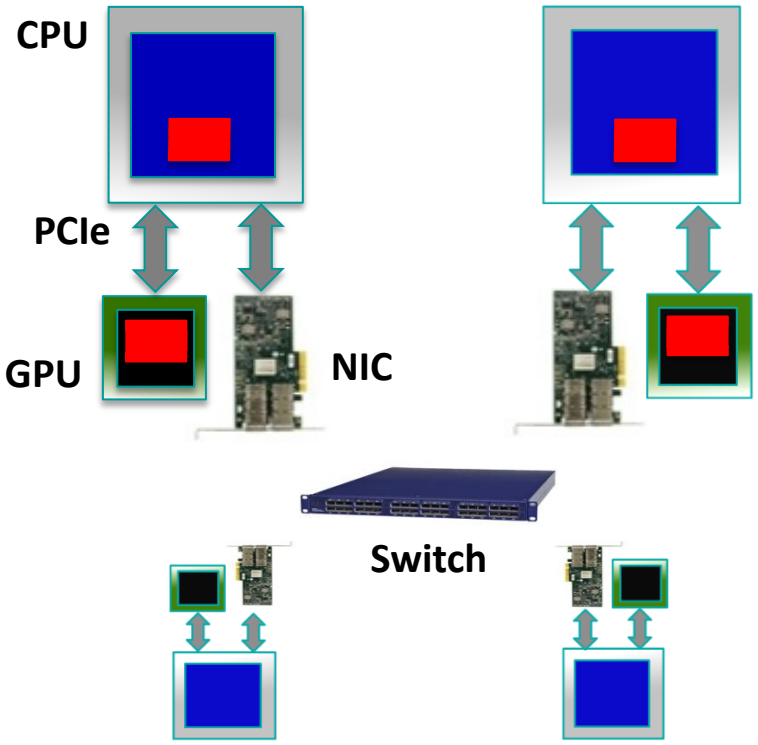
- *High Productivity and Poor Performance*

Sample Code – User Optimized Code

- Pipelining at user level with non-blocking MPI and CUDA interfaces
- Code at Sender side (and repeated at Receiver side)

At Sender:

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



- User-level copying may not match with internal MPI design
- *High Performance and Poor Productivity*

Can this be done within MPI Library?

- Support GPU to GPU communication through standard MPI interfaces
  - e.g. enable MPI_Send, MPI_Recv from/to GPU memory
- Provide high performance without exposing low level details to the programmer
  - Pipelined data transfer which *automatically* provides optimizations inside MPI library without user tuning
- **A new Design incorporated in MVAPICH2 to support this functionality**

MVAPICH/MVAPICH2 Software

- High Performance MPI Library for IB and HSE
  - MVAPICH (MPI-1) and MVAPICH2 (MPI-2.2)
  - Available since 2002
  - Used by more than 1,810 organizations in 65 countries
  - More than 85,000 downloads from OSU site directly
  - Empowering many TOP500 clusters
    - 5th ranked 73,278-core cluster (Tsubame 2) at Tokyo Tech
    - 7th ranked 111,104-core cluster (Pleiades) at NASA
    - 17th ranked 62,976-core cluster (Ranger) at TACC
  - Available with software stacks of many IB, HSE and server vendors including Open Fabrics Enterprise Distribution (OFED) and Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu

Sample Code – MVAPICH2-GPU

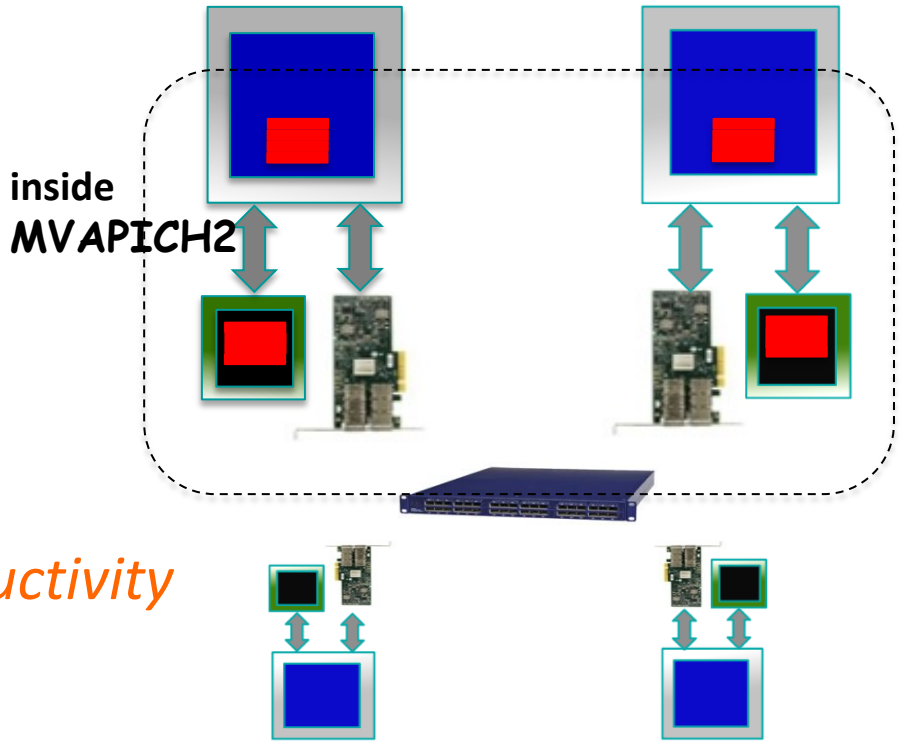
- MVAPICH2-GPU: standard MPI interfaces used

At Sender:

```
MPI_Send(s_device, size, …);
```

At Receiver:

```
MPI_Recv(r_device, size, …);
```

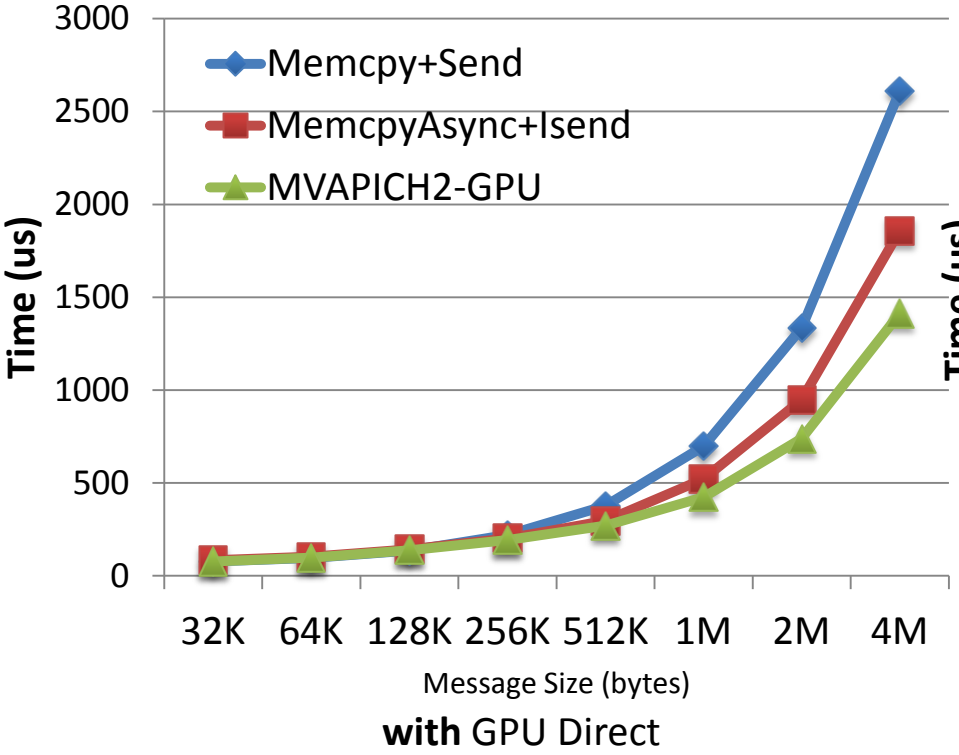


- *High Performance and High Productivity*

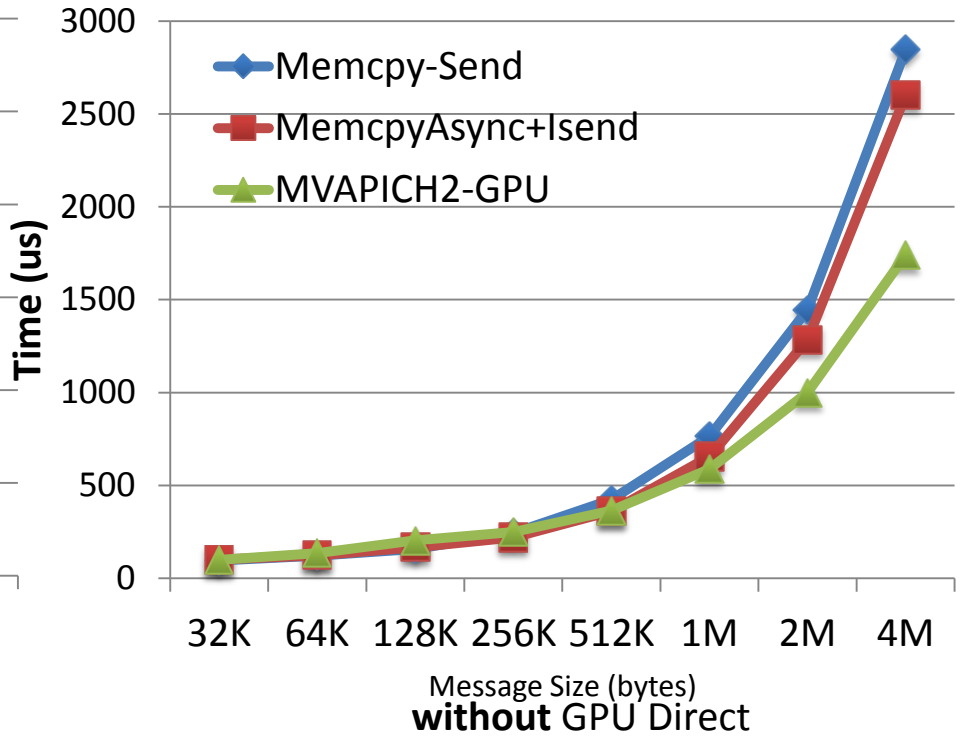
Design considerations

- Memory detection
  - CUDA 4.0 introduces *Unified Virtual Addressing (UVA)*
  - MPI library can differentiate between device memory and host memory without any hints from the user
- Overlap CUDA copy and RDMA transfer
  - Data movement from GPU and RDMA transfer are DMA operations
  - Allow for asynchronous progress

MPI-Level Two-sided Communication



with GPU Direct

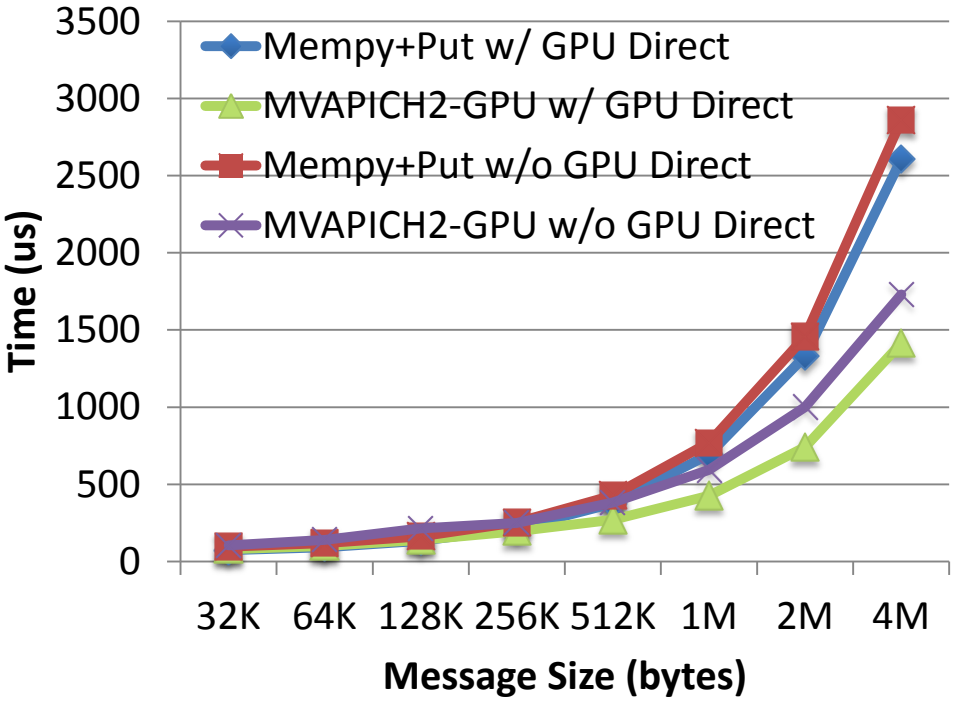


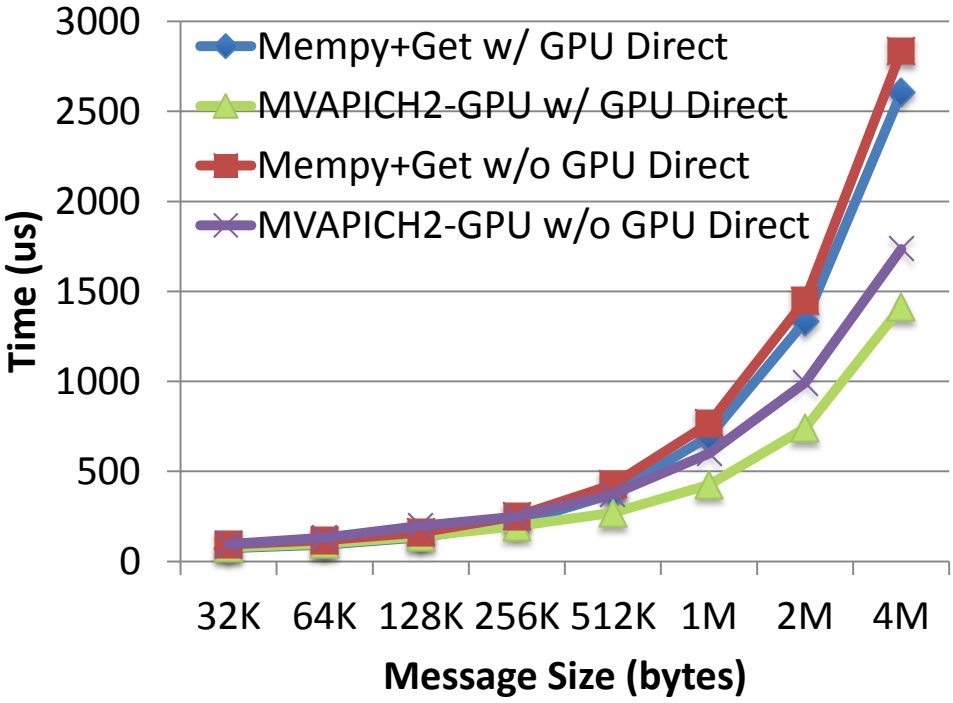
without GPU Direct

- 45% and 38% improvements compared to Memcpy+Send, with and without GPUDirect respectively, for 4MB messages
- 24% and 33% improvement compared with MemcpyAsync+Isend, with and without GPUDirect respectively, 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

MPI-Level One-sided Communication





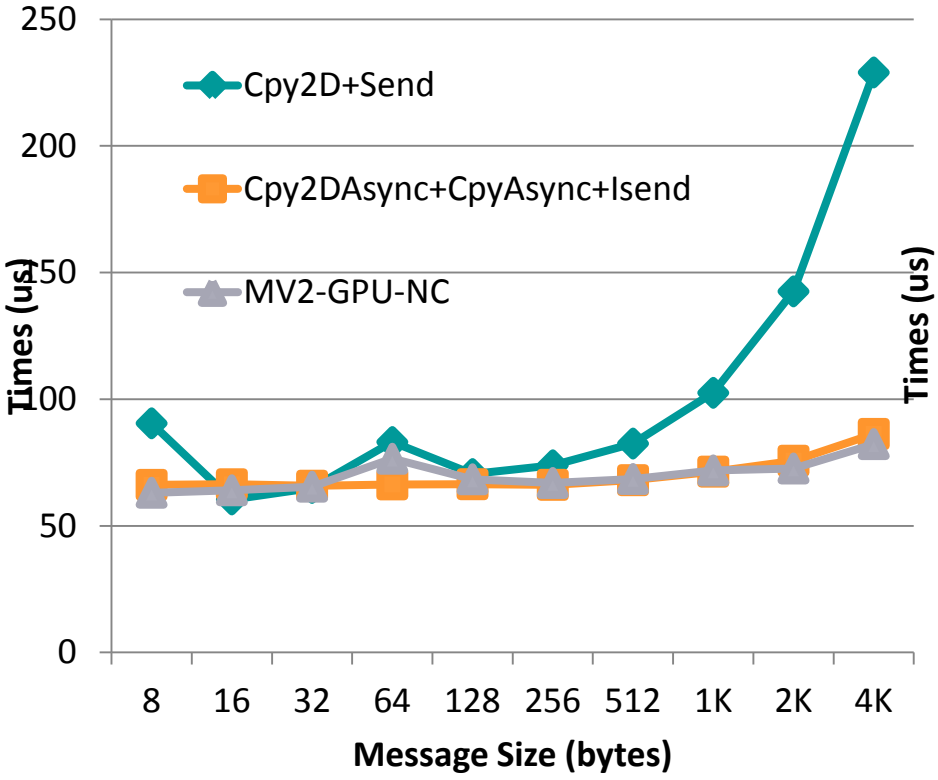
- 45% improvement compared to Memcpy+Put (with GPU Direct)
- 39% improvement compared with Memcpy+Put (without GPU Direct)
- Similar improvement for Get operation

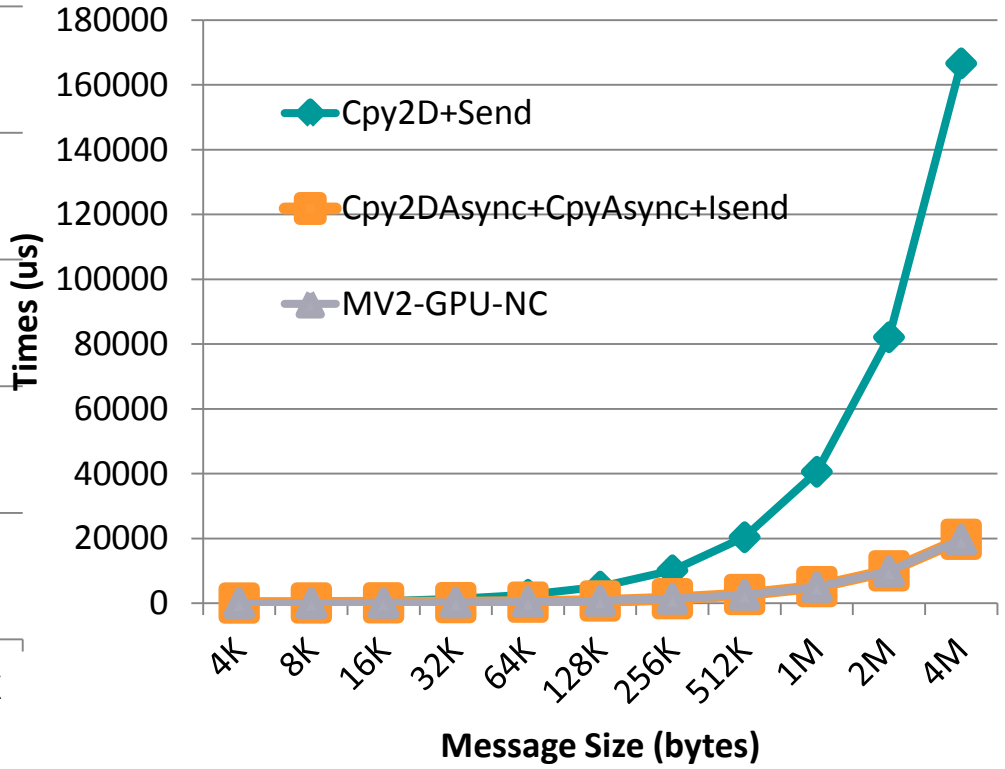
MVAPICH2-GPU-NonContiguous Datatypes: Design Goals

- Support GPU-GPU non-contiguous data communication through standard MPI interfaces
  - e.g. MPI_Send / MPI_Recv can operate on GPU memory address for non-contiguous datatype, like MPI_Type_vector
- Provide high performance without exposing low level details to the programmer
  - Offload Datatype pack and unpack to GPU
    - Pack: pack non-contiguous data into contiguous buffer inside GPU, then move out
    - Unpack: move contiguous data into GPU memory, then unpack to destination address
  - Pipelined data transfer which *automatically* provides optimizations inside MPI library without user tuning

H. Wang, S. Potluri, M. Luo, A. Singh, X. Ouyang, S. Sur and D. K. Panda, Optimized Non-contiguous MPI Datatype Communication for GPU Clusters: Design, Implementation and Evaluation with MVAPICH2, IEEE Cluster '11, Sept. 2011.

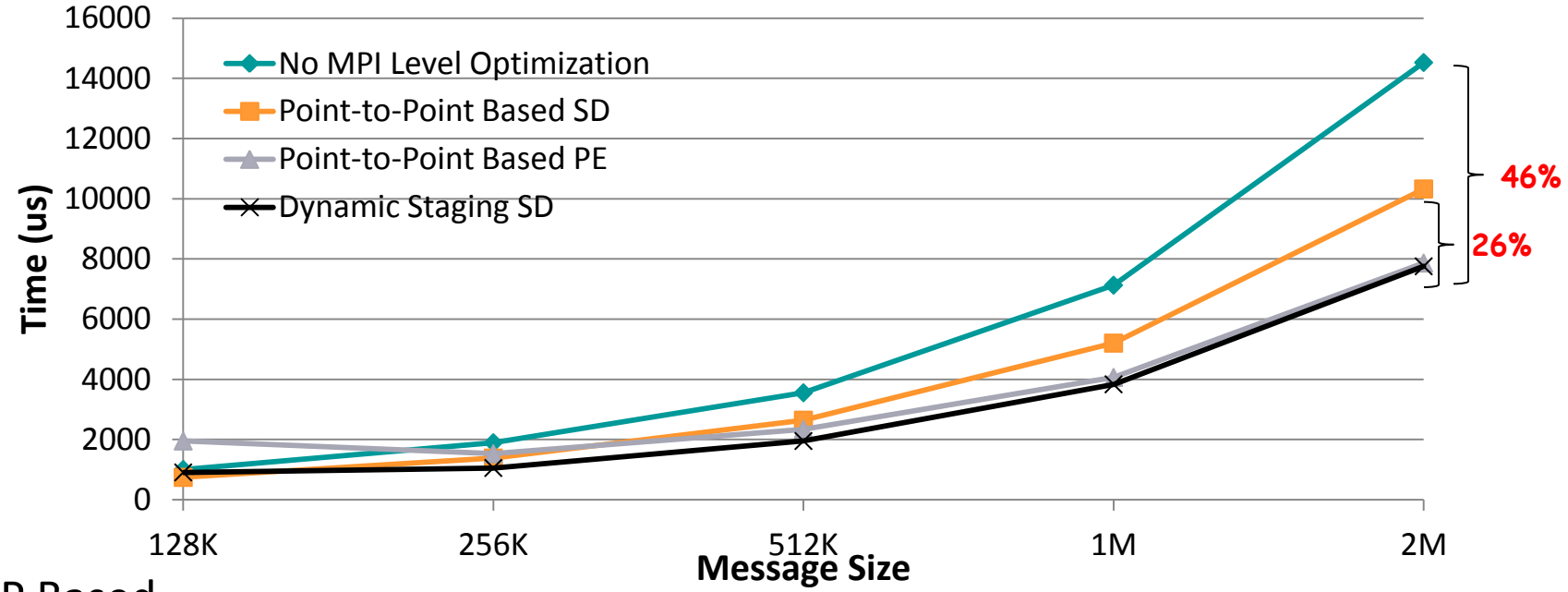
MPI Latency for Vector Data





- MV2-GPU-NonContiguous (MV2-GPU-NC)
  - 88% improvement compared with Cpy2D+Send at 4MB
  - Have similar latency with Cpy2DAsync+CpyAsync+Isend, but much easier for programming

Optimization with Collectives: Alltoall Latency (Large Message)



- P2P Based
  - Pipeline is enabled for each P2P channel (ISC'11); better than No MPI Level Optimization
- Dynamic Staging Scatter Dissemination (SD)
  - Not only overlap DMA and RDMA for each channel, but also for different channels
  - up to 46% improvement for Dynamic Staging SD over No MPI Level Optimization
  - up to 26% improvement for Dynamic Staging SD over P2P Based method SD

A. Singh, S. Potluri, H. Wang, K. Kandalla, S. Sur and D. K. Panda, MPI Alltoall Personalized Exchange on GPGPU Clusters: Design Alternatives and Benefits, Workshop on Parallel Programming on Accelerator Clusters (PPAC '11), held in conjunction with Cluster '11, Sept. 2011.

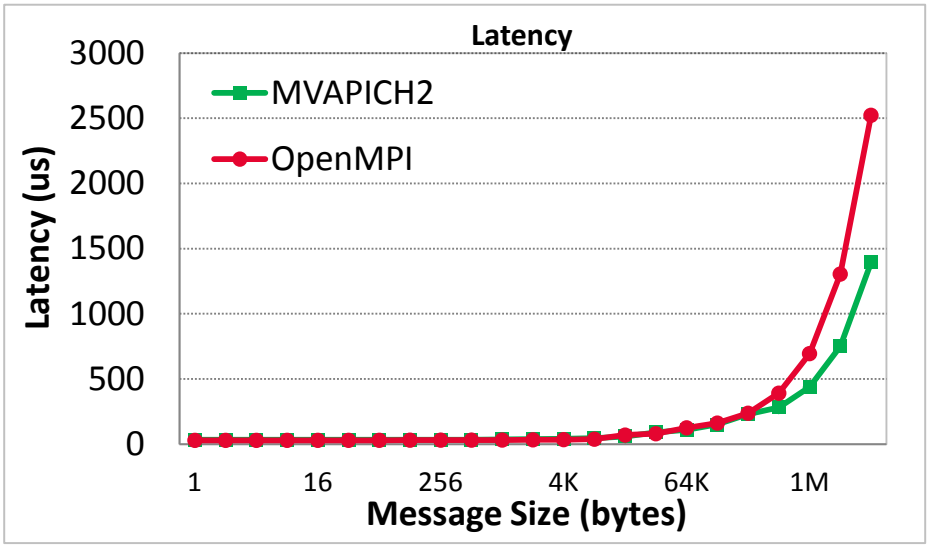
MVAPICH2 1.8a1p1 Release

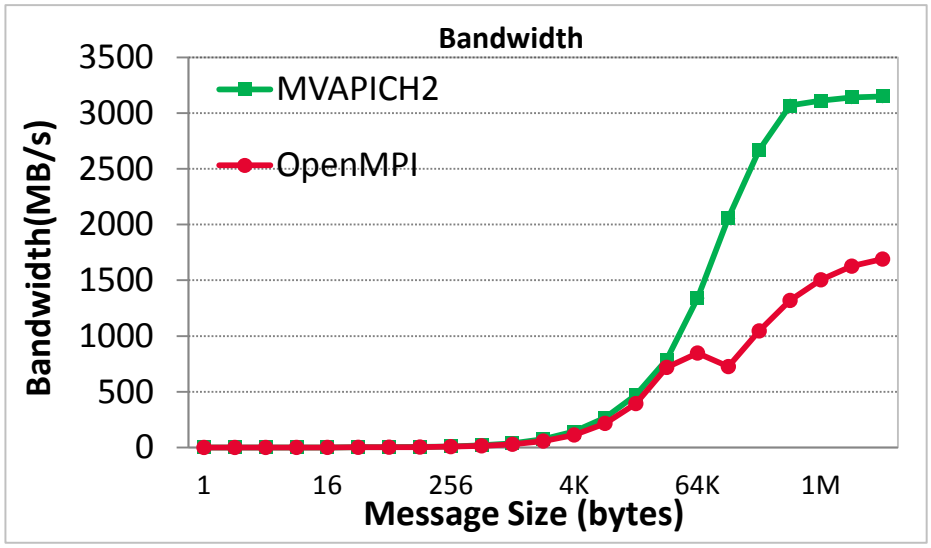
- Supports basic point-to-point communication (as outlined in ISC '11 paper)
- Supports the following transfers
  - Device-Device
  - Device-Host
  - Host-Device
- Supports GPU-Direct through CUDA support (from 4.0)
- Tuned and optimized for CUDA 4.1 RC1
- Provides flexibility for block-size (MV2_CUDA_BLOCK_SIZE) to be selected at runtime to tune performance of an MPI application based on its predominant message size
- Available from http://mvapich.cse.ohio-state.edu

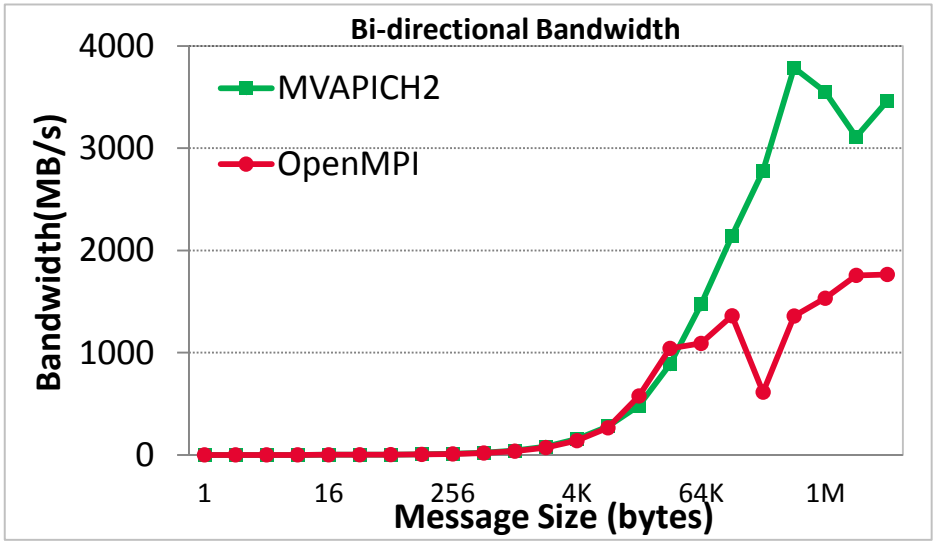
OSU MPI Micro-Benchmarks (OMB) 3.5 Release

- OSU MPI Micro-Benchmarks provides a comprehensive suite of benchmarks to compare performance of different MPI stacks and networks
- Enhancements done for three benchmarks
  - Latency
  - Bandwidth
  - Bi-directional Bandwidth
- Flexibility for using buffers in NVIDIA GPU device (D) and host memory (H)
- Flexibility for selecting data movement between D->D, D->H and H->D
- Available from http://mvapich.cse.ohio-state.edu/benchmarks
- Available in an integrated manner with MVAPICH2 stack

MVAPICH2 vs. OpenMPI (Device-Device, Inter-node)



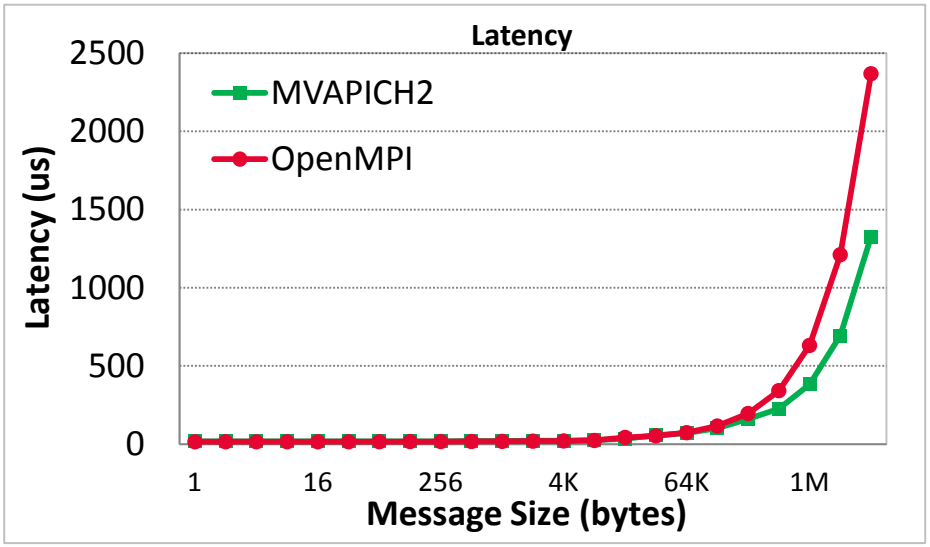


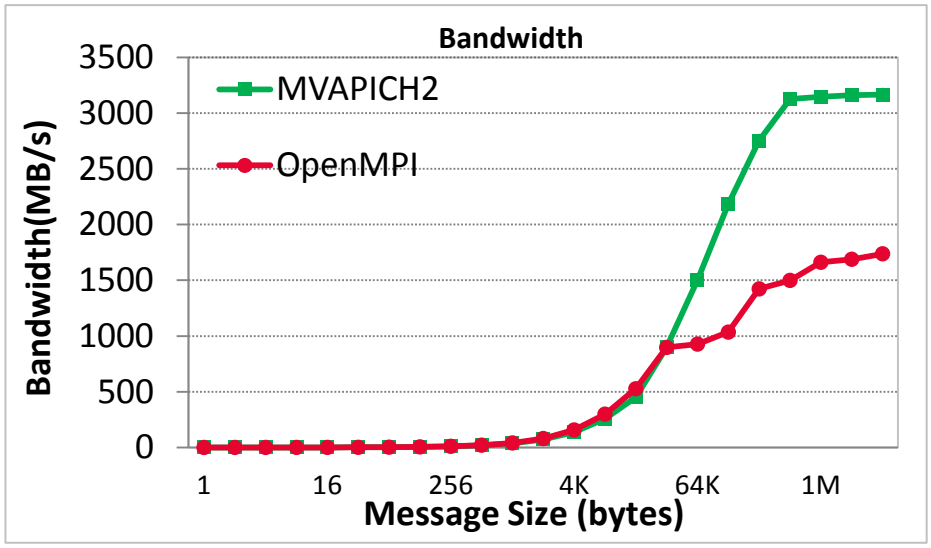


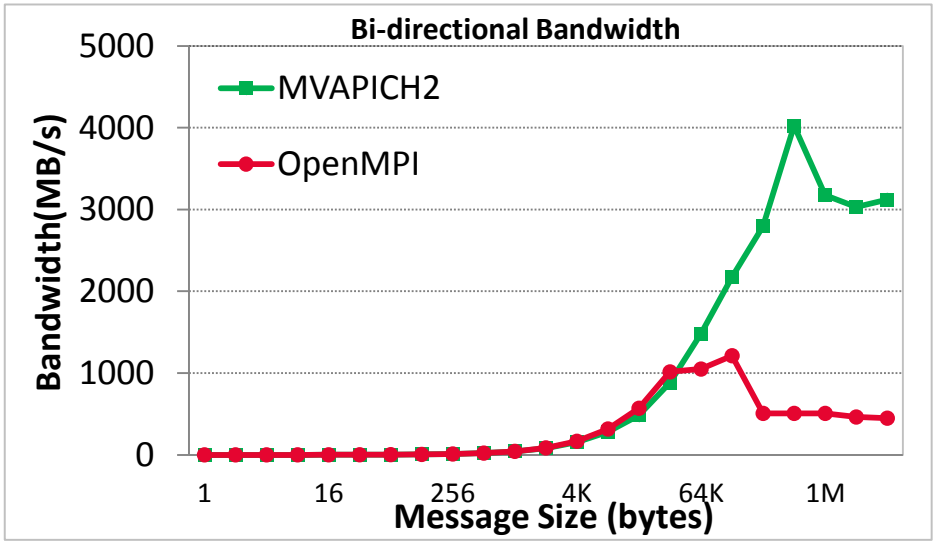
MVAPICH2 1.8 a1p1 and OpenMPI (Trunk nightly snapshot on Nov 1, 2011)

Westmere with ConnectX-2 QDR HCA, NVIDIA Tesla C2050 GPU and CUDA Toolkit 4.1RC1

MVAPICH2 vs. OpenMPI (Device-Host, Inter-node)



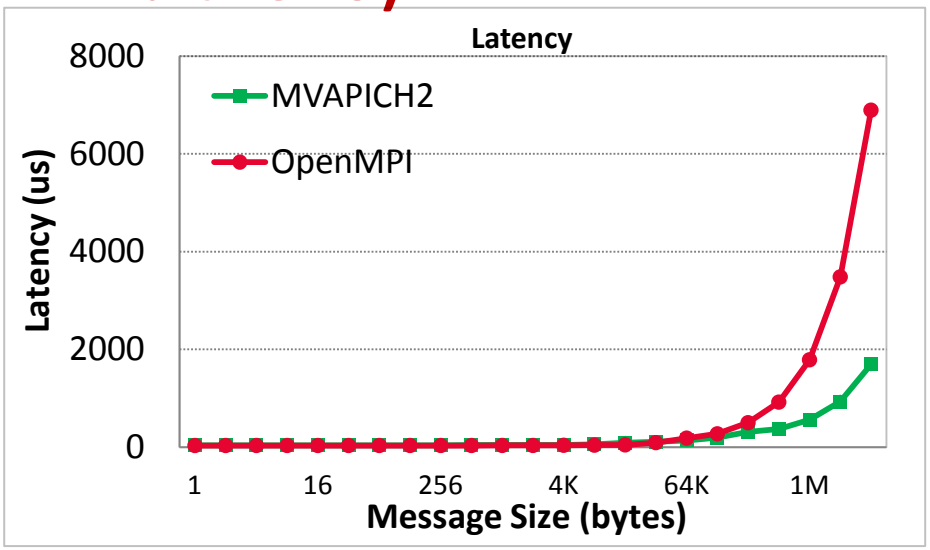


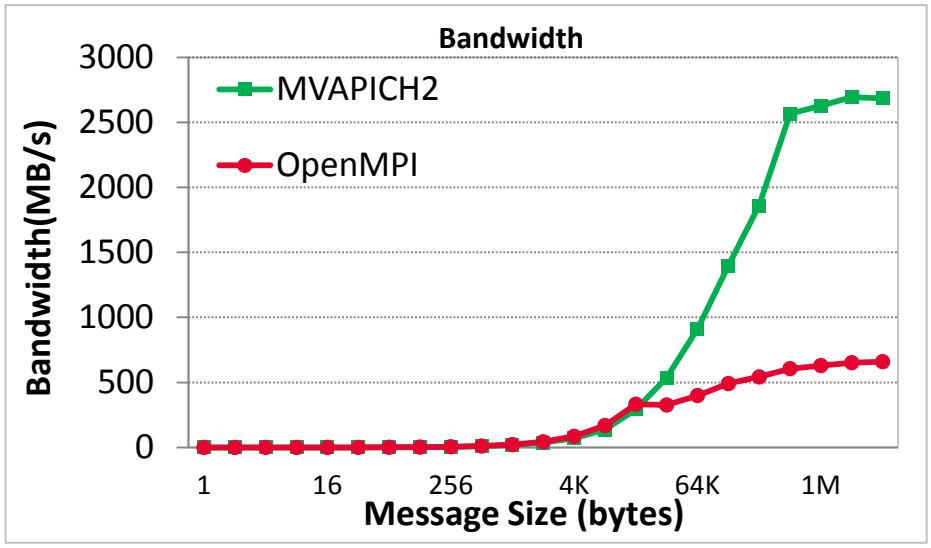


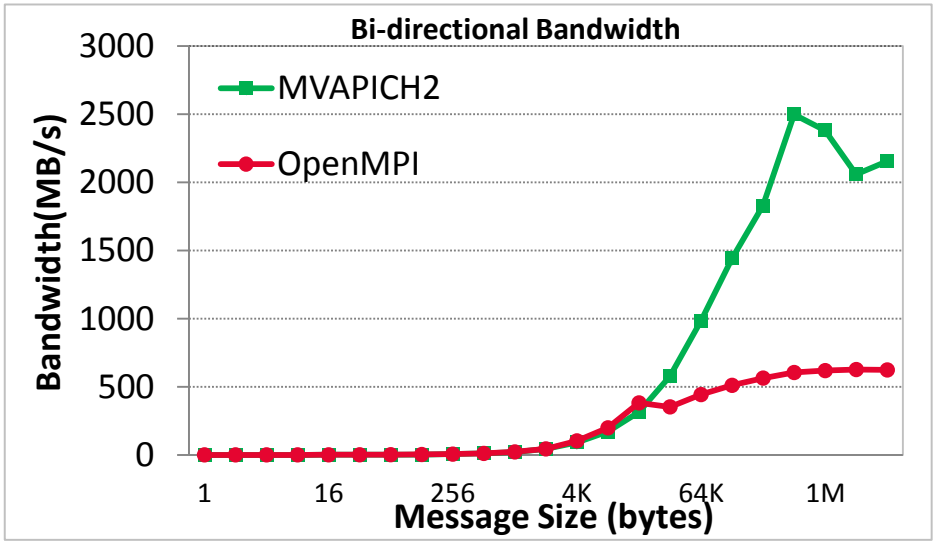
Host-Device Performance is Similar

MVAPICH2 1.8 a1p1 and OpenMPI (Trunk nightly snapshot on Nov 1, 2011)
Westmere with ConnectX-2 QDR HCA, NVIDIA Tesla C2050 GPU and CUDA Toolkit 4.1RC1

MVAPICH2 vs. OpenMPI (Device-Device, Intra-node, Multi-GPU)



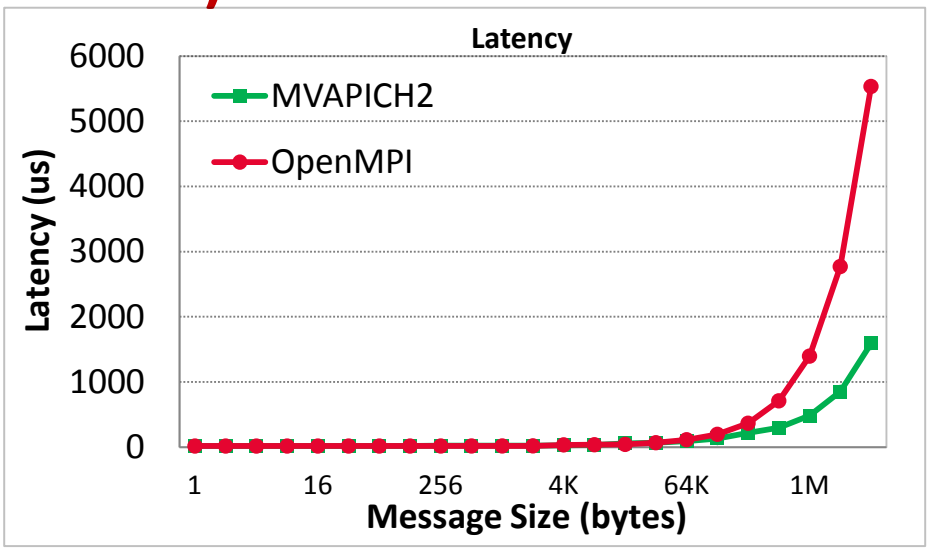


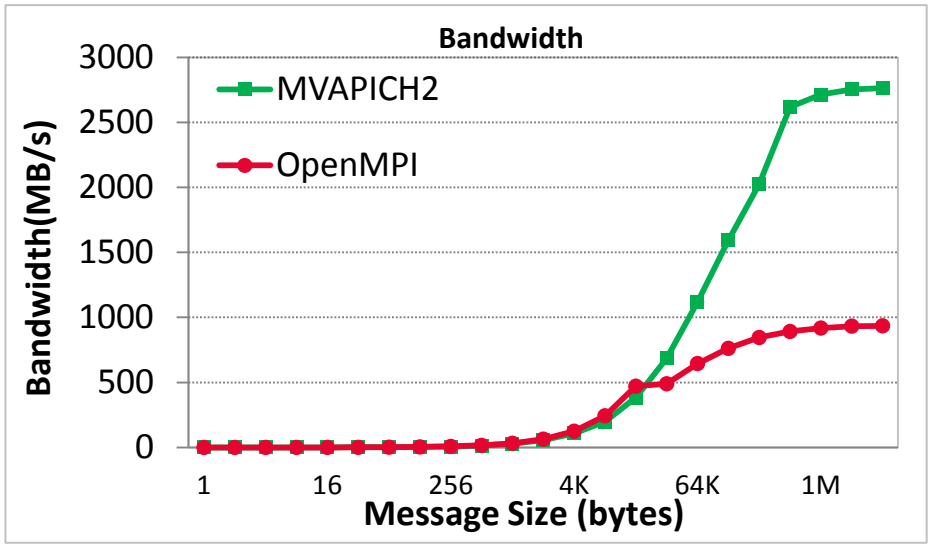


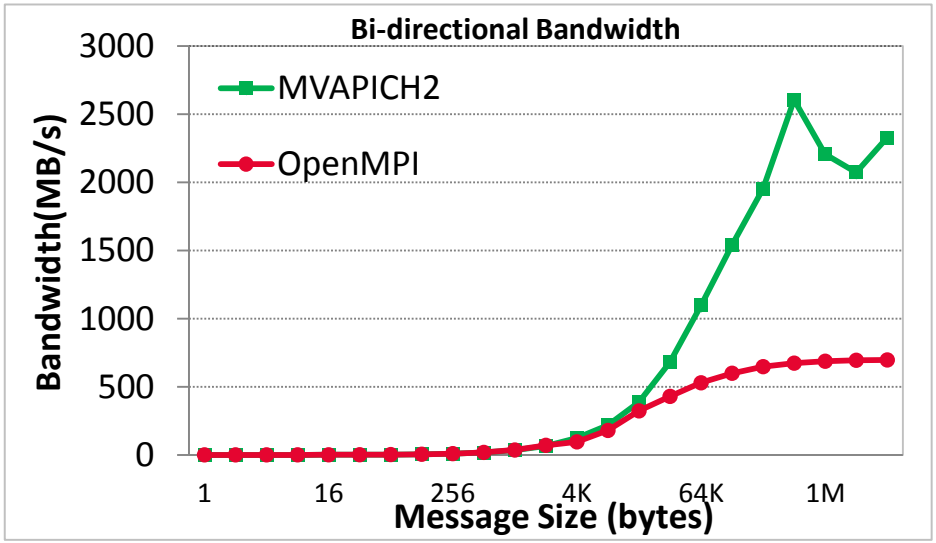
MVAPICH2 1.8 a1p1 and OpenMPI (Trunk nightly snapshot on Nov 1, 2011)

Magny Cours with 2 ConnectX-2 QDR HCAs, 2 NVIDIA Tesla C2075 GPUs and CUDA Toolkit 4.1RC1

MVAPICH2 vs. OpenMPI (Device-Host, Intra-node, Multi-GPU)





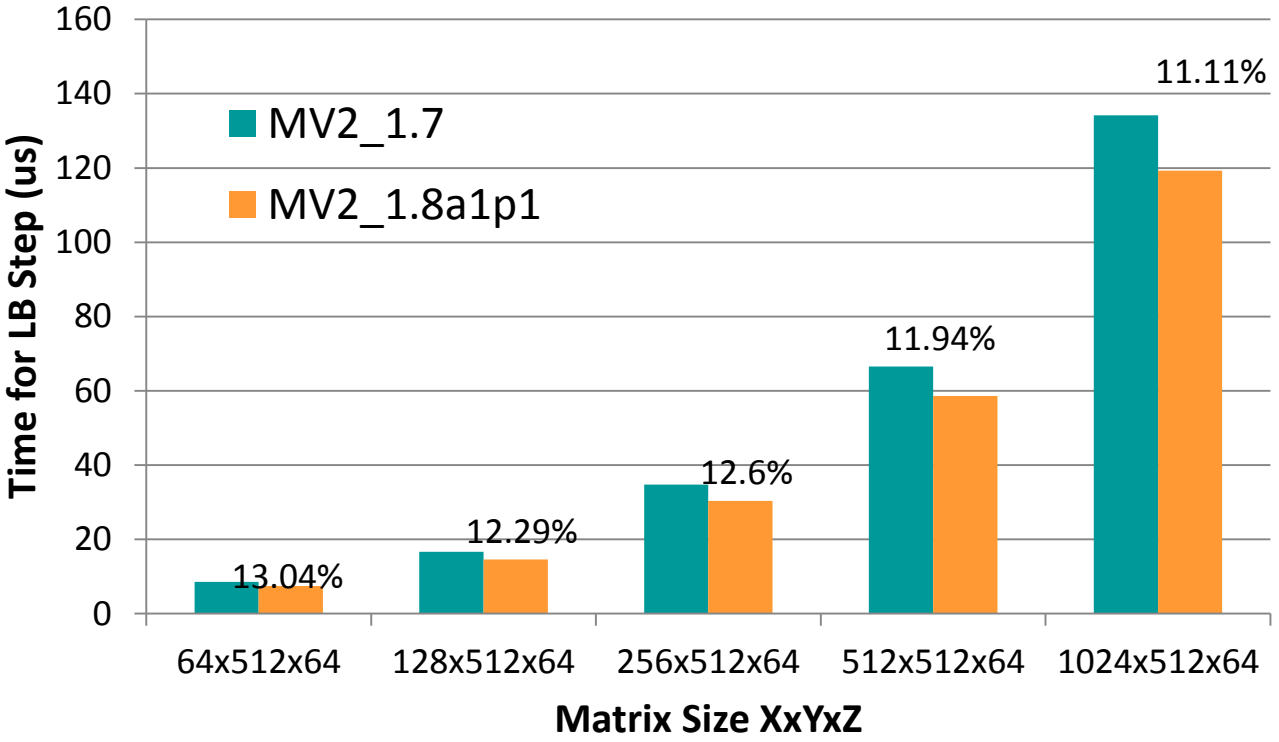


Host-Device Performance is Similar

MVAPICH2 1.8 a1p1 and OpenMPI (Trunk nightly snapshot on Nov 1, 2011)

Magny Cours with 2 ConnectX-2 QDR HCAs, 2 NVIDIA Tesla C2075 GPUs and CUDA Toolkit 4.1RC1

Applications-Level Evaluation (Lattice Boltzmann Method (LBM))



- LBM-CUDA (Courtesy Carlos Rosale, TACC) is a parallel distributed CUDA implementation of a Lattice Boltzmann Method for multiphase flows with large density ratios
- NVIDIA Tesla C2050, Mellanox QDR InfiniBand HCA MT26428, Intel Westmere Processor with 12 GB main memory; CUDA 4.1 RC1, MVAPICH2 1.7 and MVAPICH2 1.8a1p1
- Run one process on each node for one GPU (4-node cluster)

Future Plans

- Extend the current design/implementation to support
  - Point-to-Point communication with Datatype Support
  - Direct one-sided communication
  - Optimized designs for all collectives
- Additional tuning based on
  - Platform characteristics
  - InfiniBand adapters (FDR)
  - Upcoming CUDA versions
- Carry out applications-based evaluations and scalability studies and enhance/adjust the designs
- Extend the current designs to MPI-3 RMA and PGAS Models (UPC, OpenShmem, etc.)
- Extend additional benchmarks in OSU MPI Micro-Benchmark (OMB) suite with GPU support

Conclusion

- Clusters with NVIDIA GPUs and InfiniBand are becoming very common
- However, programming these clusters with MPI and CUDA while achieving highest performance is a big challenge
- MVAPICH2-GPU design provides a novel approach to extract highest performance with least burden to the MPI programmers
- Significant potential for programmers to write high-performance and scalable MPI applications for NVIDIA GPU+IB clusters

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students

- V. Dhanraj (M.S.)
- J. Huang (Ph.D.)
- N. Islam (Ph.D.)
- J. Jose (Ph.D.)
- K. Kandalla (Ph.D.)
- M. Luo (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekhar (Ph.D.)
- M. Rahman (Ph.D.)
- A. Singh (Ph.D.)
- H. Subramoni (Ph.D.)

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.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- P. Lai (Ph. D.)
- J. Liu (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- G. Santhanaraman (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)

Current Post-Docs

- J. Vienne
- H. Wang

Current Programmers

- M. Arnold
- D. Bureddy
- J. Perkins

Past Post-Docs

- X. Besseron
- H.-W. Jin
- E. Mancini
- S. Marcarelli

Past Research Scientist

- S. Sur

Web Pointers

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

http://nowlab.cse.ohio-state.edu

MVAPICH Web Page

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

panda@cse.ohio-state.edu