MVAPICH/MVAPICH2 Update

Presentation at Open Fabrics Developers Conference (Nov. '07)

by

Dhabaleswar K. (DK) Panda
Department of Computer Science and Engg.
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

Presentation Overview

- Overview of MVAPICH/MVAPICH2 Project
- Features of MVAPICH 1.0 and MVAPICH2 1.0
- Sample Performance Numbers
  - Point-to-point (Mellanox and Qlogic)
  - Point-to-point with Intel-Connects Cable
  - Point-to-point with Obsidian IB-WAN
  - Multi-core-aware Optimized Collectives
  - UD-based Design
  - Hot-spot Avoidance Mechanism (HSAM)
- Upcoming Features and Issues
  - XRC support
  - Enhanced UD-based Design
  - Asynchronous Progress
- Conclusions

MVAPICH/MVAPICH2 Software Distribution

- High Performance and Scalable Implementations
  - MPI-1 (MVAPICH)
  - MPI-2 (MVAPICH2)
- Both are being available with OFED
- With OFED 1.3
  - MVAPICH 1.0-beta
  - MVAPICH2 1.0
- Directly downloaded and used by more than 580 organizations in 42 countries
- Empowering many production and TOP500 clusters. Examples include
  - 3rd ranked TOP500 system (14,336 cores) in Nov '07 list, delivering 126.9 TFlops (MVAPICH)
  - 8,192-core cluster at NCSA (MVAPICH2)
- More details at
  http://mvapich.cse.ohio-state.edu

New Features of MVAPICH 1.0

- Asynchronous Progress
  - Provides better overlap between computation and communication
- Flexible message coalescing
  - enable/disable coalescing
  - Allows varying degrees of coalescing
- UD-based support
  - Best performance and scalability with constant memory footprint for communication contexts
- Support for Automatic Path Migration (APM)
- Multi-core optimizations for Collectives
- Enhanced mpirun_rsh for scalable launching
  - Provides a two-level approach (nodes and cores within a node)
- Support for ConnectX
- Support for Qlogic/PSM

New Features of MVAPICH2 1.0

- Message coalescing support
- Hot-spot avoidance mechanism for alleviating network congestion in large clusters
- Application-initiated systems-level checkpoint
  - in addition to the automatic systems-level checkpoint from 0.9.8
- Automatic Path Migration (APM) support
- RDMA Read
- Blocking
- Multi-rail support for iWARP
- RDMA CM-based connection management (Gen2-IB and Gen2-iWARP)
- On-demand connection management for uDAPL (including Solaris)

Support for Multiple Interfaces/Adapters

- OpenFabrics/Gen2-IB
  - All IB adapters supporting Gen2
  - ConnectX
- Qlogic/PSM
- uDAPL
  - Linux-IB
  - Solaris-IB
  - Other adapters such as Neteffect 10GigE
- OpenFabrics/Gen2-iWARP
  - Chelsio
- VAPI
  - All IB adapters supporting VAPI
- TCP/IP
  - Any adapter supporting TCP/IP interface
- Shared Memory Channel (MVAPICH), for running applications in a node with multi-core processors

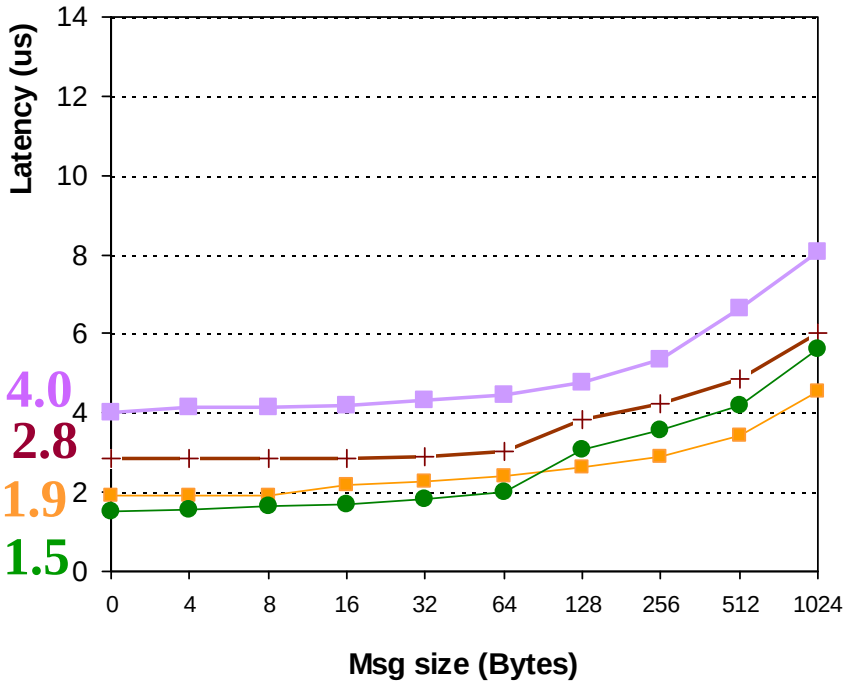
Presentation Overview

- Overview of MVAPICH/MVAPICH2 Project
- Features of MVAPICH 1.0 and MVAPICH2 1.0
- Sample Performance Numbers
  - Point-to-point (Mellanox and Qlogic)
  - Point-to-point with Intel-Connects Cable
  - Point-to-point with Obsidian IB-WAN
  - Multi-core-aware Optimized Collectives
  - UD-based Design
  - Hot-spot Avoidance Mechanism (HSAM)
- Upcoming Features and Issues
  - XRC support
  - Enhanced UD-based Design
  - Asynchronous Progress
  - Passive synchronization support
- Conclusions

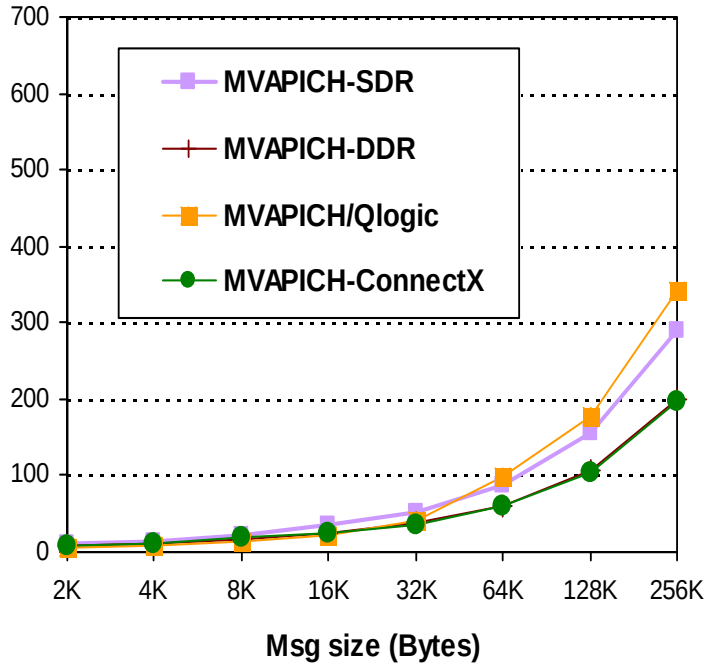
MPI-level Latency (One-way): IBA (Mellanox and QLogic)

2.33 GHz Quad-core Intel with IB switch

Small message latency

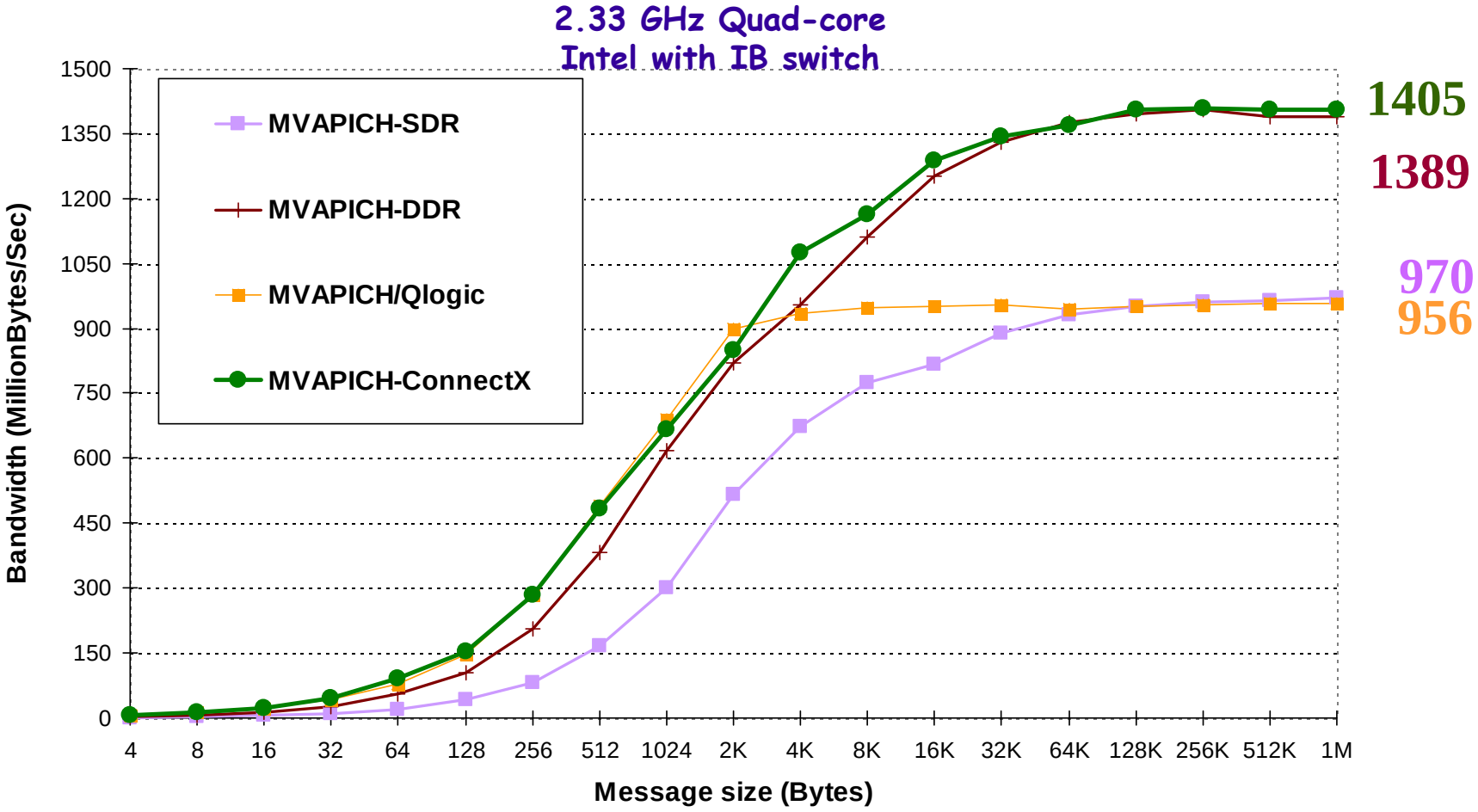


Large message latency

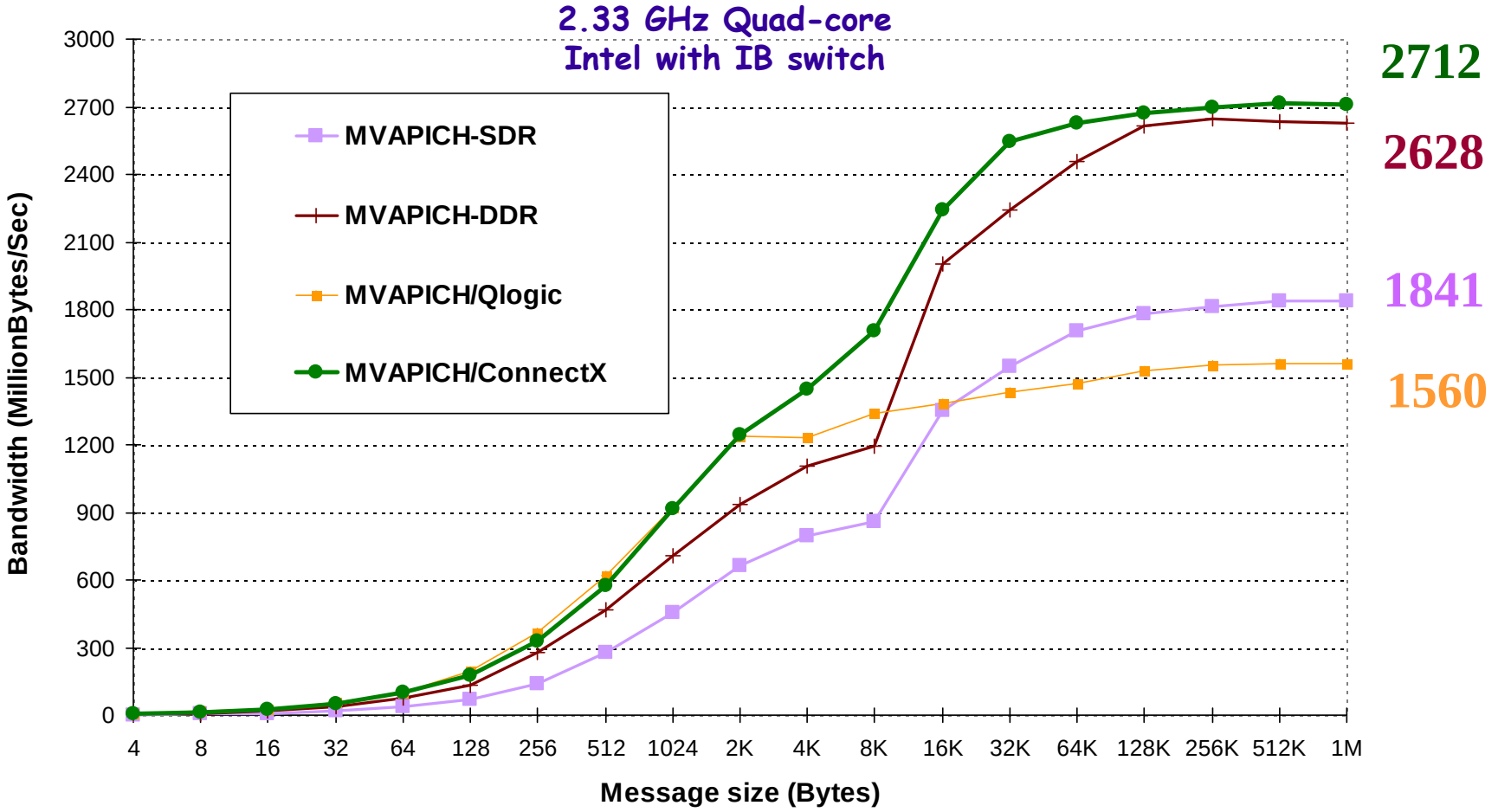


- From various papers - SC '03, Hot Interconnect '04, IEEE Micro (Jan-Feb) '05, one of the best papers from HotI '04
- Also from `Performance' link of MVAPICH page

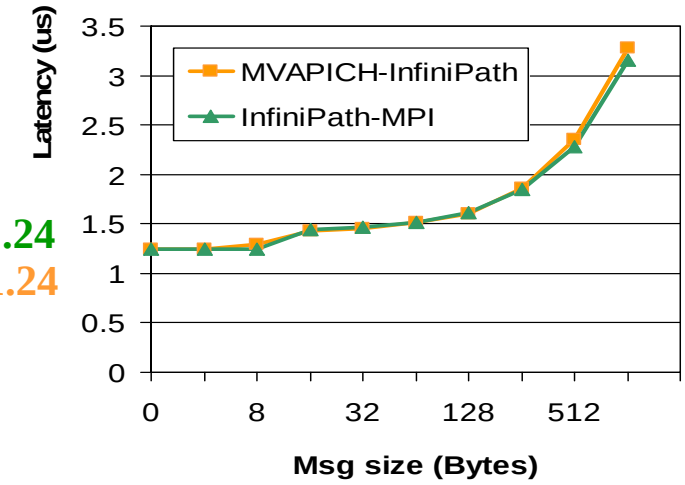
MPI-level Bandwidth (Uni-directional): IBA (Mellanox and QLogic)



MPI-level Bandwidth (Bi-directional): IBA (Mellanox and QLogic)

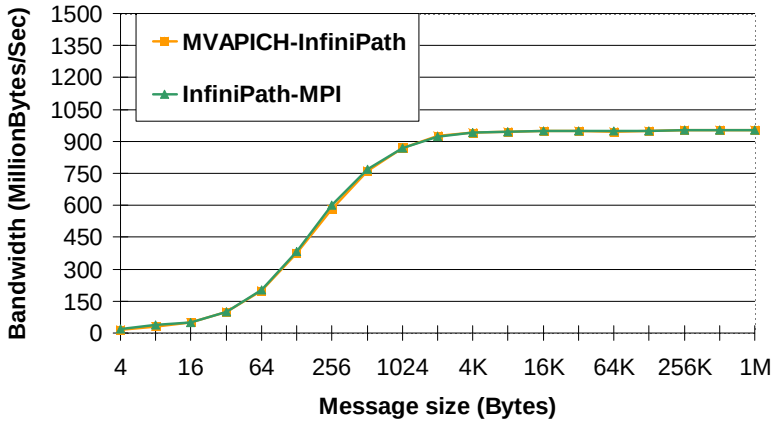


MVAPICH-PSM Performance: AMD Opteron with HT



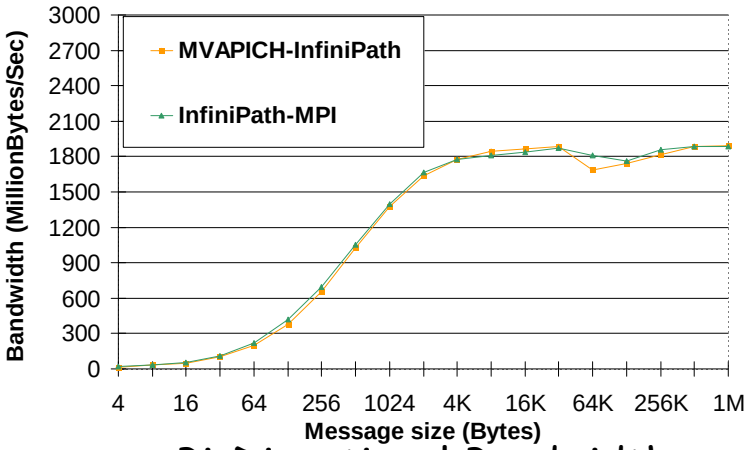
1.24
1.24

Small Message Latency



953
952

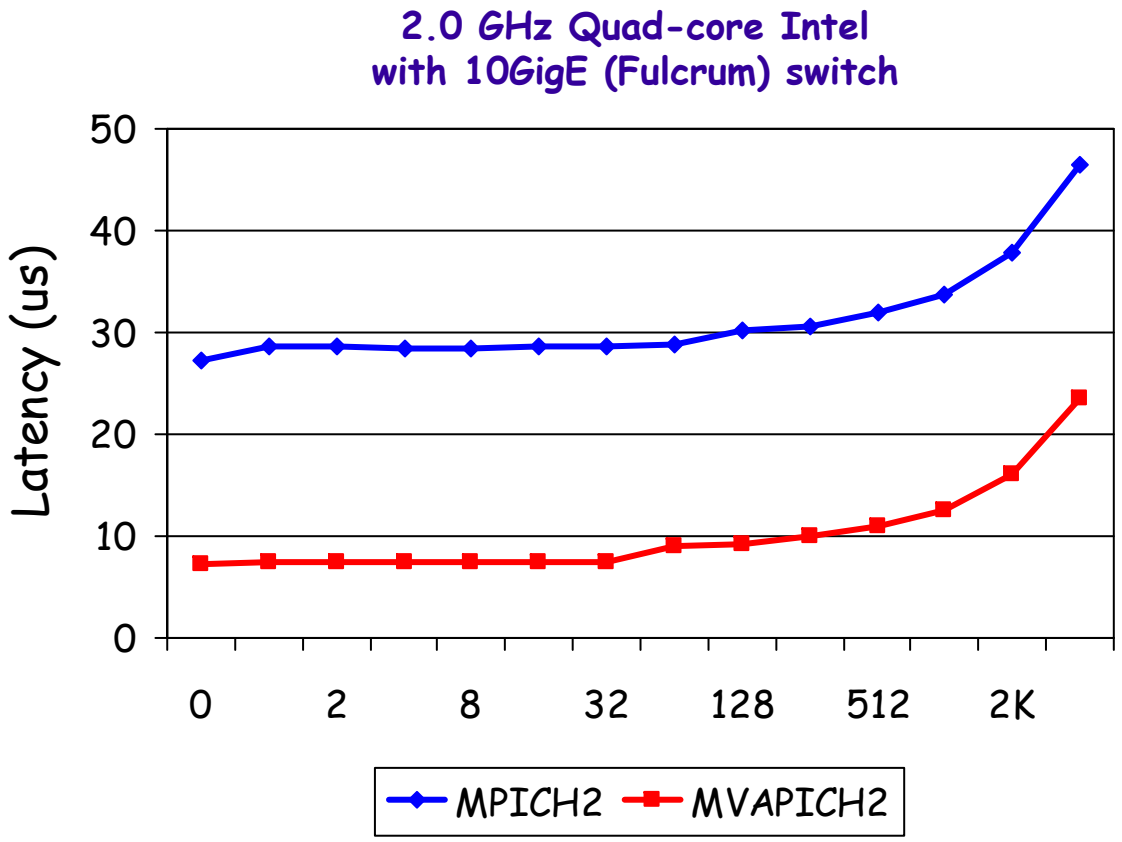
Uni-Directional Bandwidth



1888
1888

Bi-Directional Bandwidth

MPI-level Latency (One-way): iWARP with Chelsio

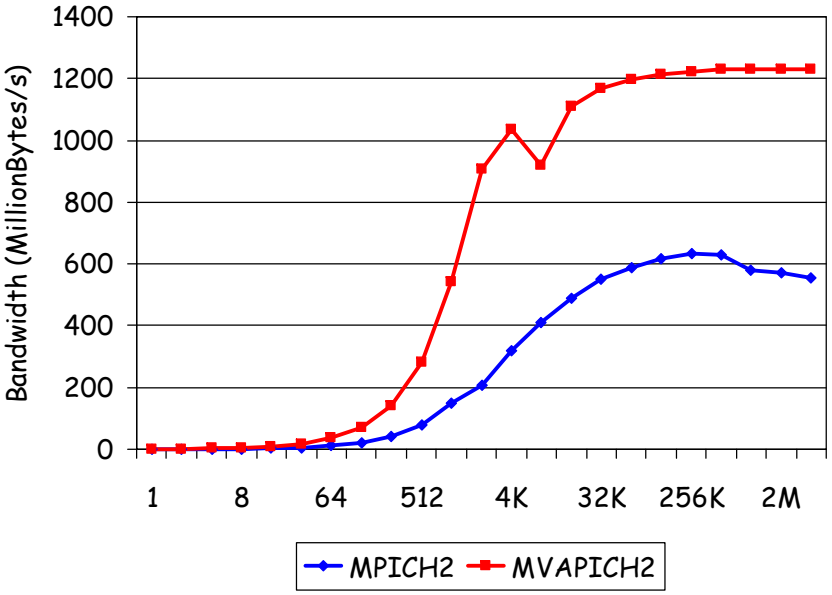


MVAPICH2 gives a latency of about 7.39us as compared to 28.5 for MPICH2

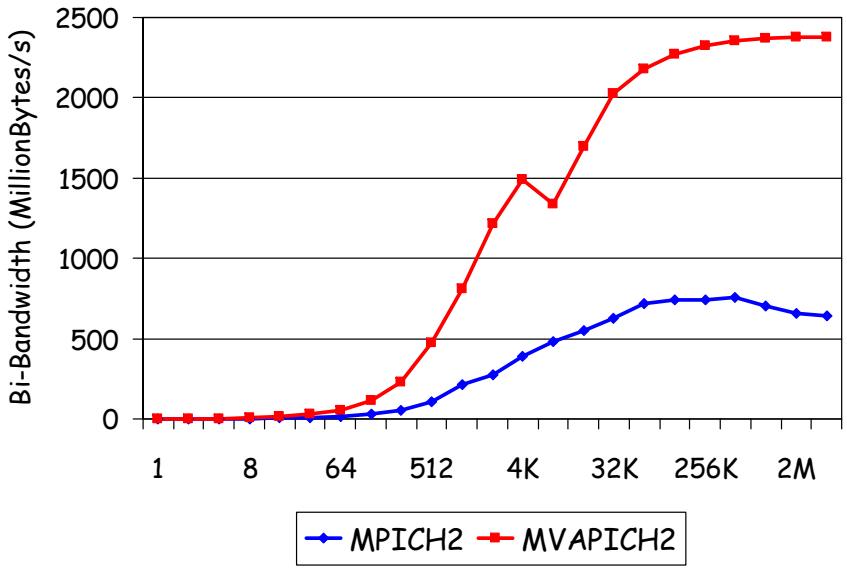
MPI-level Bandwidth: iWARP with Chelsio

2.0 GHz Quad-core Intel with 10GigE (Fulcrum) switch

Uni-directional



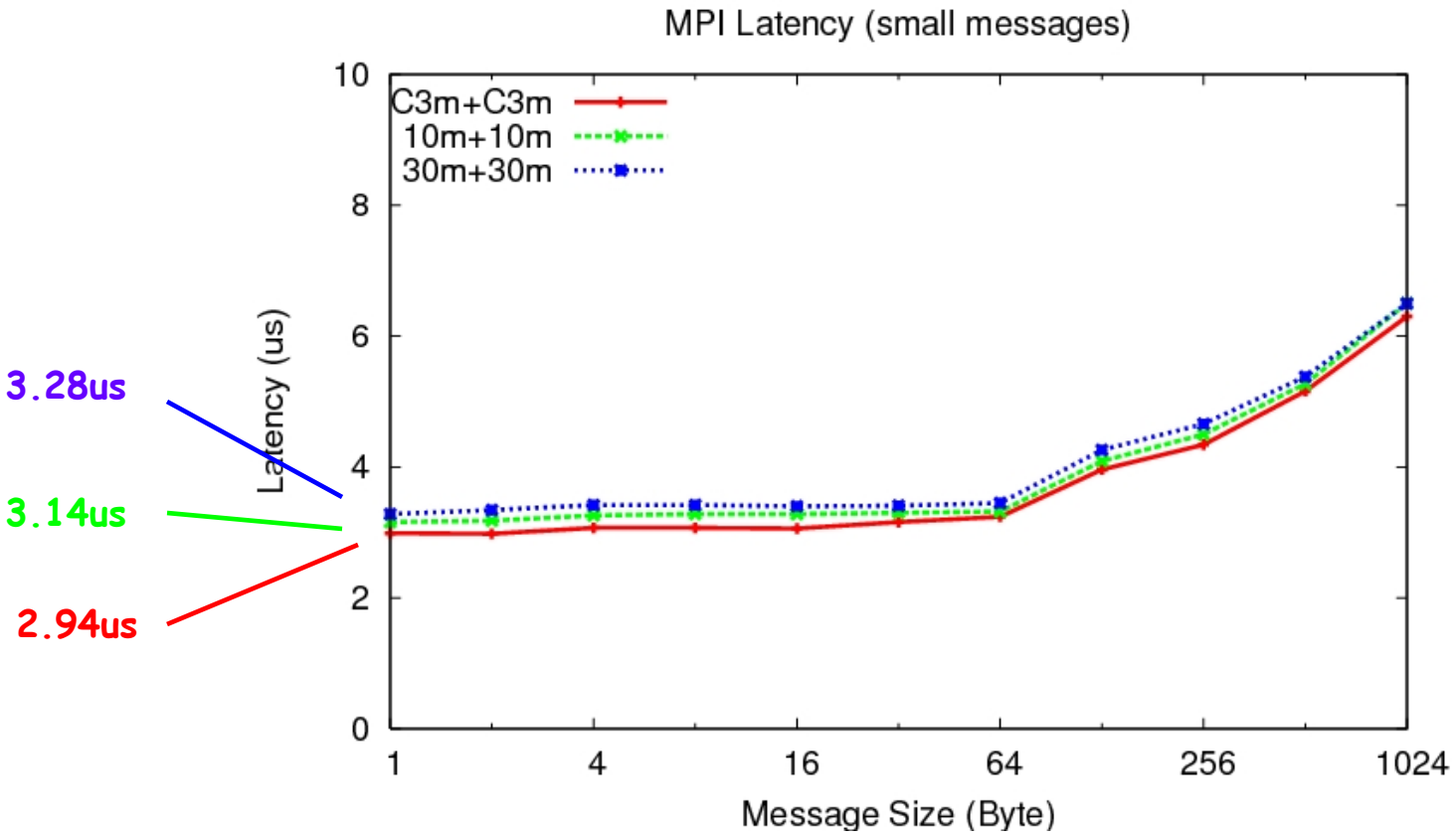
Bi-directional



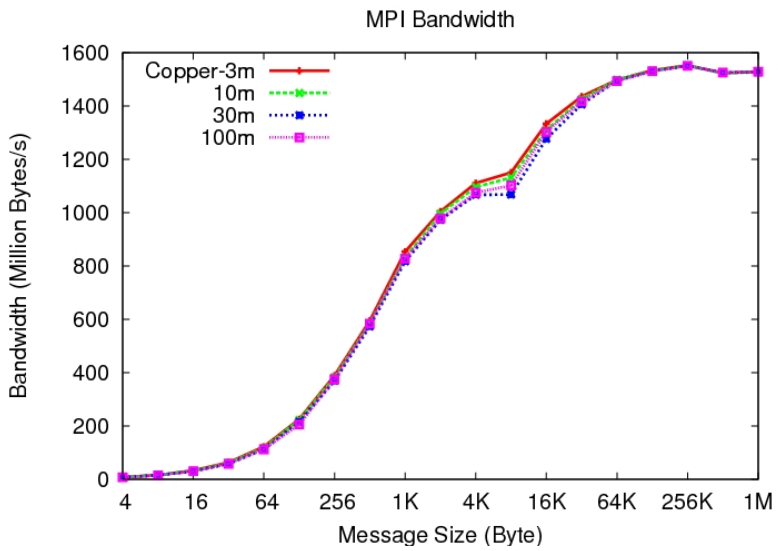
Peak bandwidth of about 1231 MillionBytes/s

Peak bidir-bandwidth of about 2380 MillionBytes/s

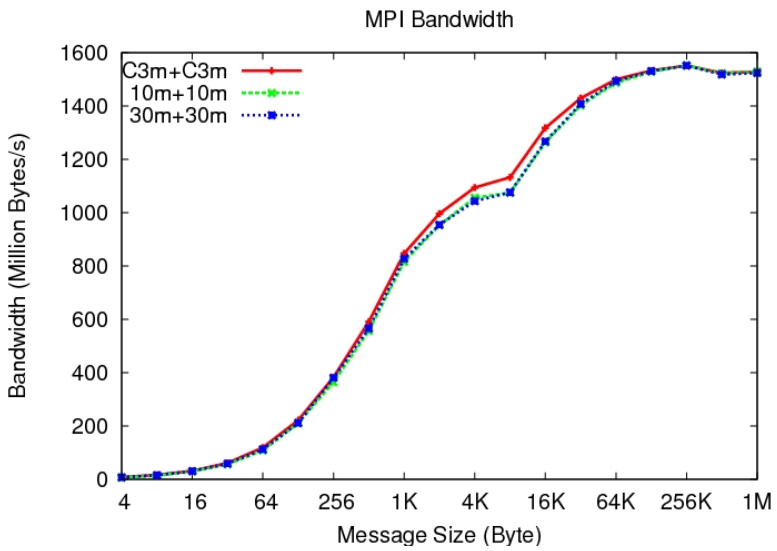
Performance with Intel-Connect IB cable (MPI Latency with switch)



Performance with Intel-Connect IB Cable (MPI Bandwidth)



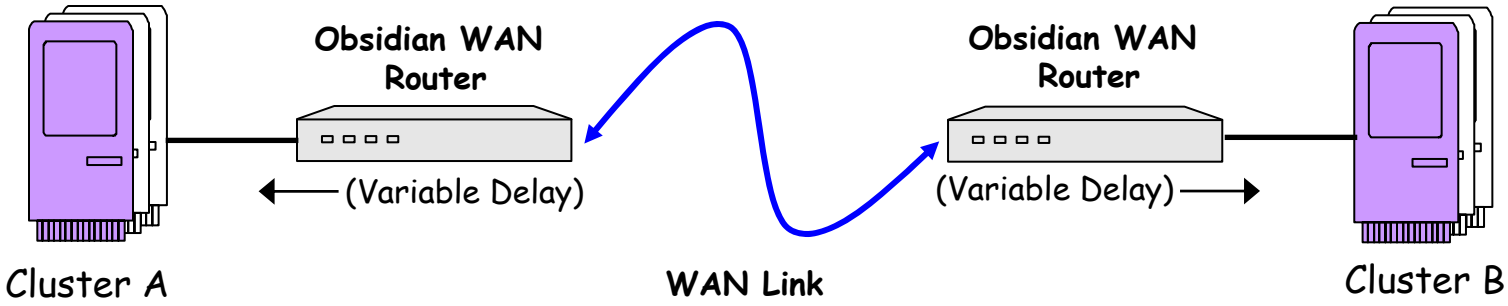
Back to back

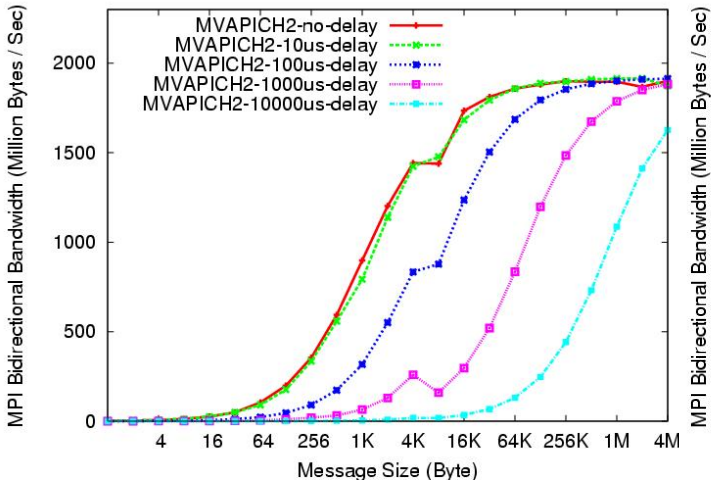


With switch

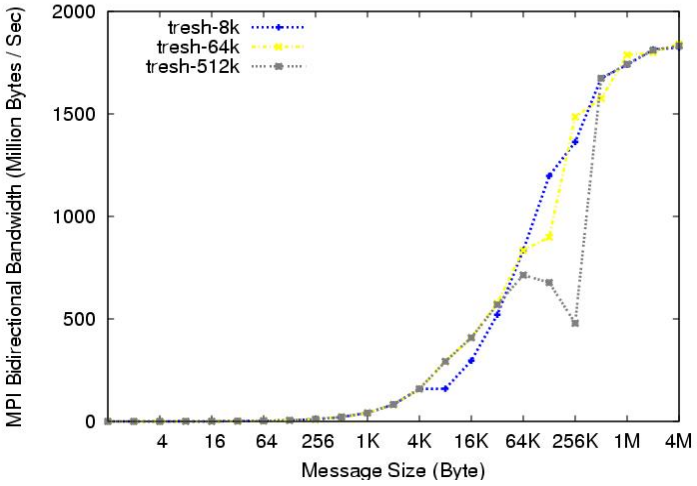
Intel Connect fiber optic cables closely match the performance of copper cables

IB WAN: Obsidian Routers





MPI Bidirectional Bandwidth



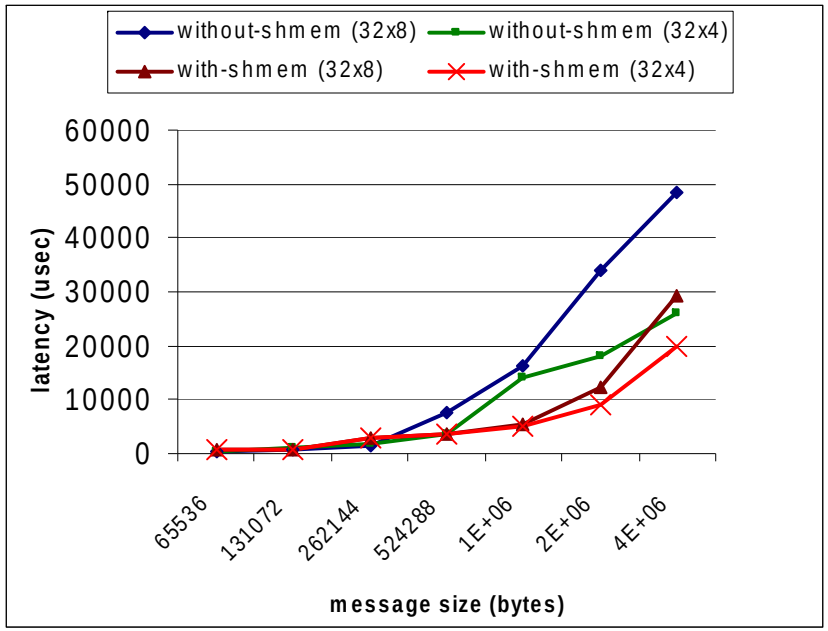
Varying protocol threshold (Delay 1 ms)

Delay (us)	Distance (km)
10	2
100	20
1000	200
10000	2000

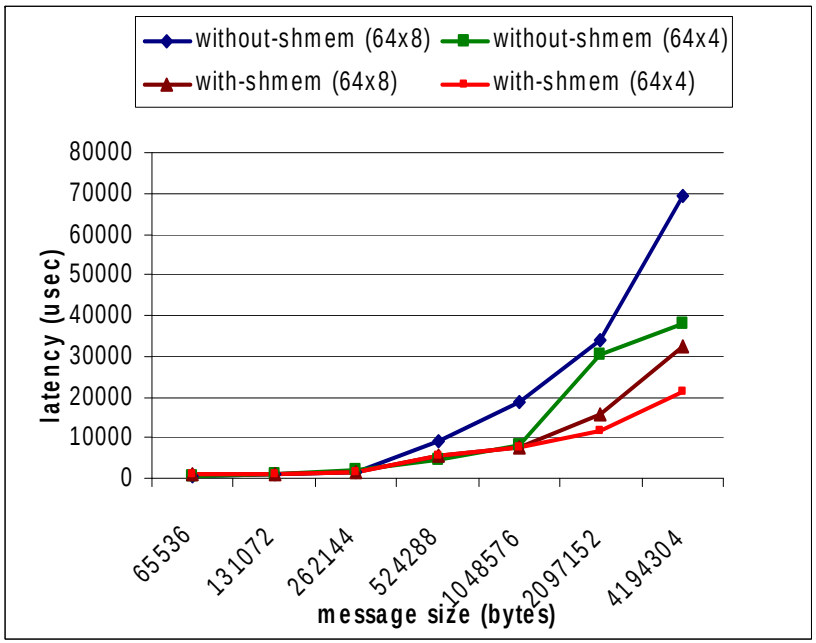
Up to 80% performance difference for 8k messages with and without protocol tuning

MPI_Bcast Latency

32 nodes

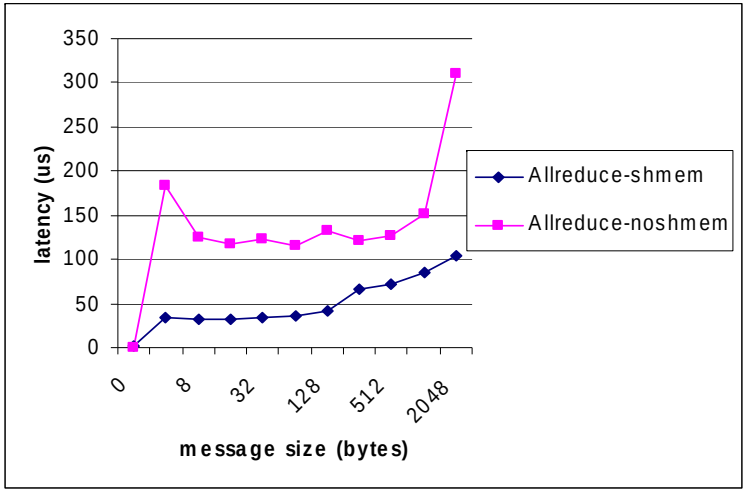


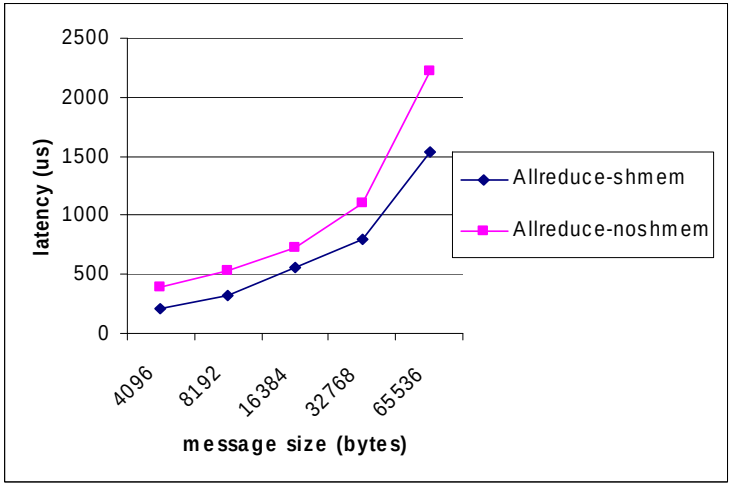
64 nodes



- Using shared memory improves the performance of MPI_Bcast on 512 cores by more than two times

Shared Memory-Based Collectives for Multi-core Platforms

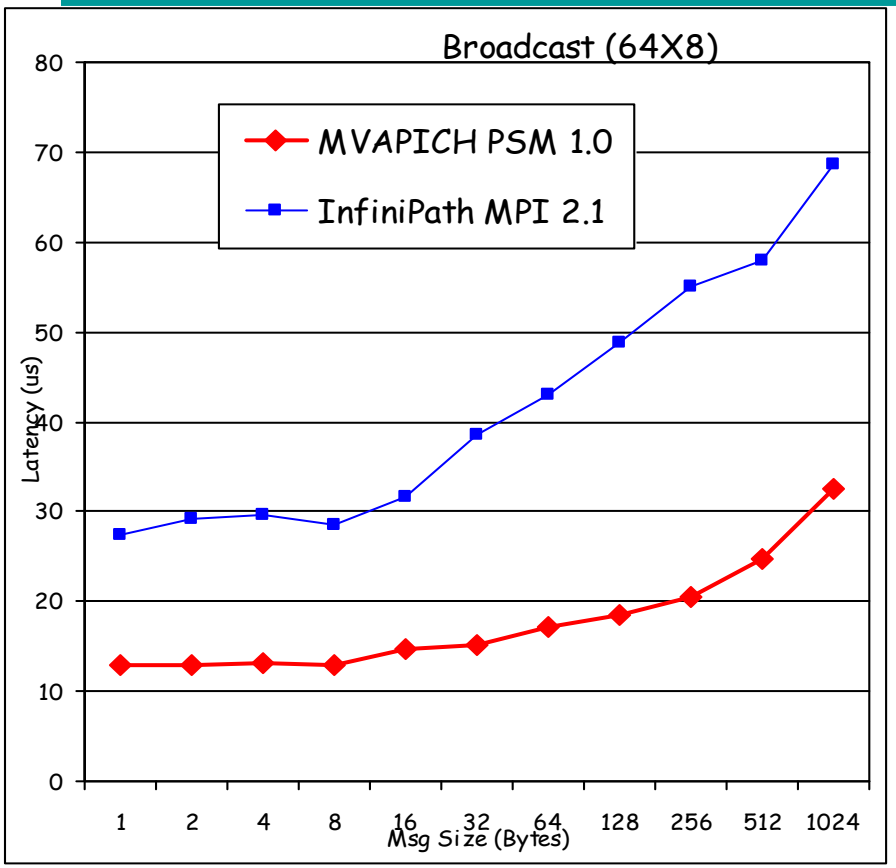


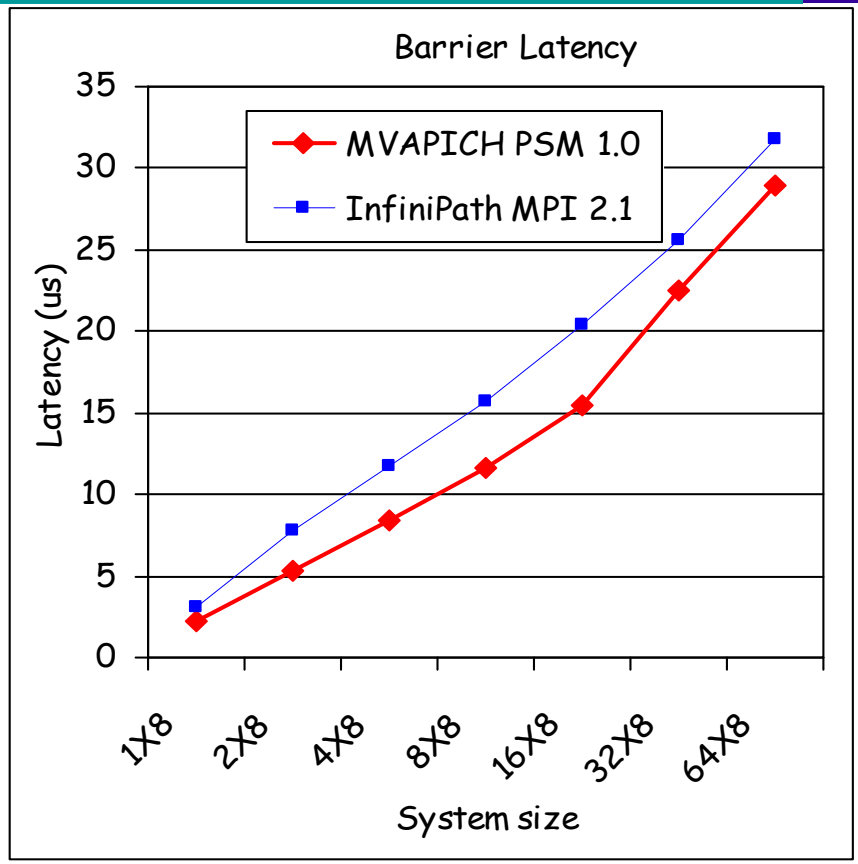


- 64 Intel Quad-core systems with dual sockets (512 cores)
- Improves performance by almost 3 times
- Similar performance improvement for MPI_Barrier

Efficient Shared Memory and RDMA based design for MPI_Allgather over InfiniBand, Amith R. Mamidala, Abhinav Vishnu and D. K. Panda, EuroPVM/MPI, September 2006

MVAPICH/PSM Collective Performance

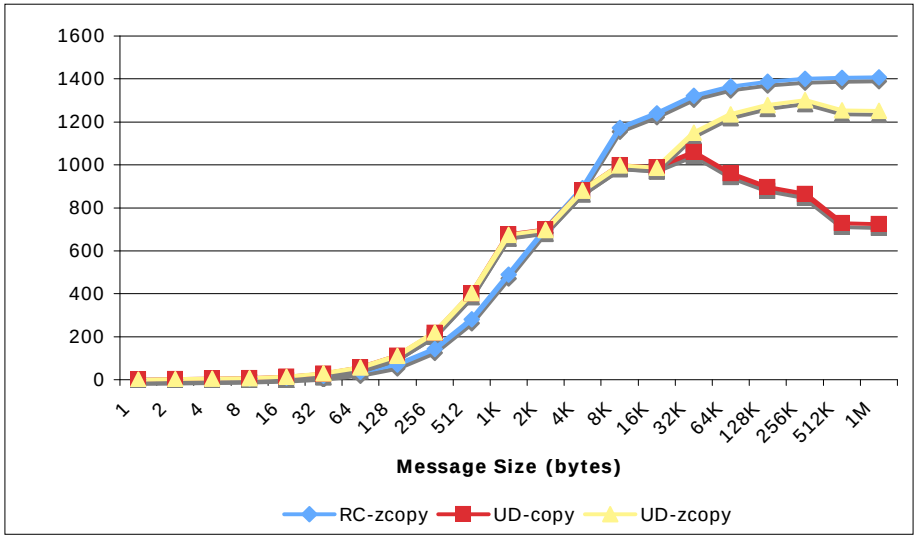




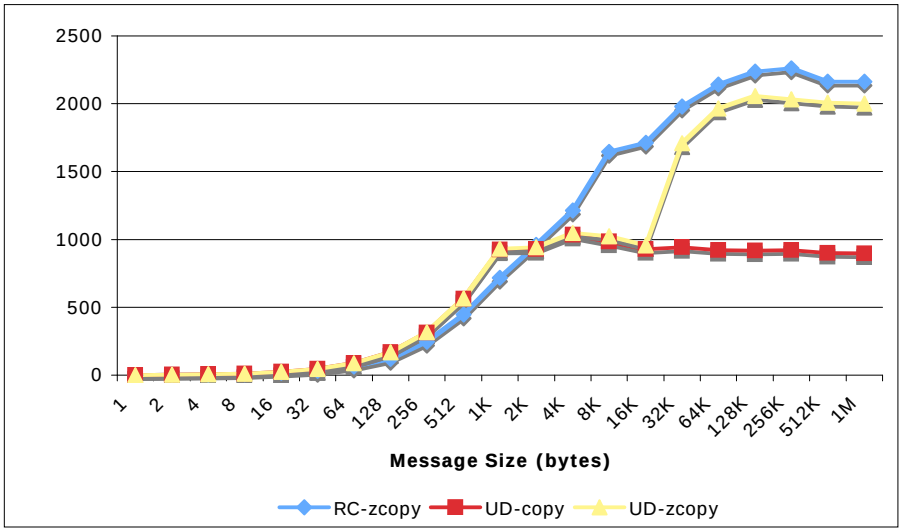
- 64 Intel Quad-core systems with dual sockets; PCIe InfiniPath Adapters
- Significant performance improvement for MPI_Bcast and MPI_Barrier

Zero-Copy over Unreliable Datagram (UD)

Uni-Directional Bandwidth



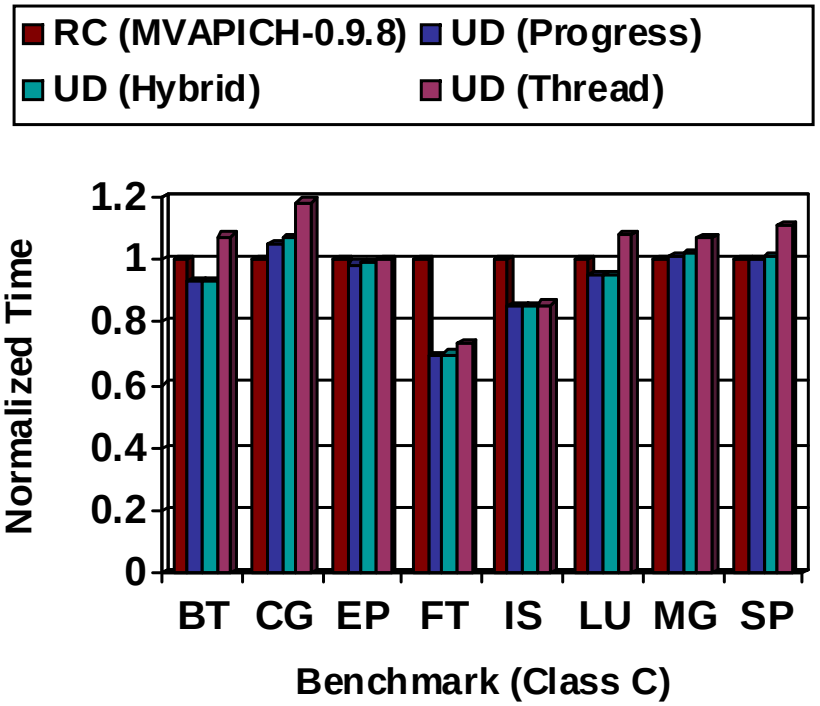
Bi-Directional Bandwidth



- **Using a novel technique, zero-copy transfers can be made over UD.**
- **Performance very close to that of RC**
- **Supported in MVAPICH 1.0-beta**

M. Koop, S. Sur and D. K. Panda, Zero-Copy Protocol for MPI using InfiniBand Unreliable Datagram, Cluster 2007

NAS Parallel Benchmarks with UD



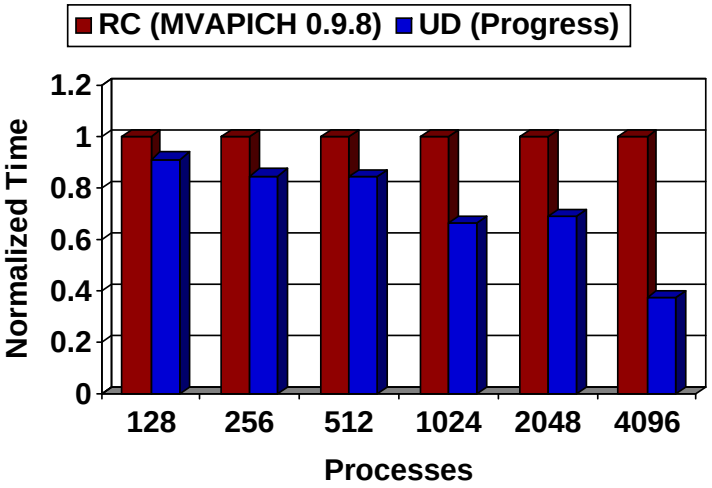
Normalized Time - 256 processes

- CFD Kernels with varied communication patterns
- UD Progress shows better performance than the thread or hybrid models
- FT and IS both use large MPI_Alltoall collective calls, in which each process communicates directly with every other process
  - ICM cache misses for RC
  - Large improvement for UD, even with 256 processes

M. Koop, S. Sur and D. K. Panda, High Performance MPI Design using Unreliable Datagram for Ultra-Scale InfiniBand Clusters, ICS 2007

SMG2000

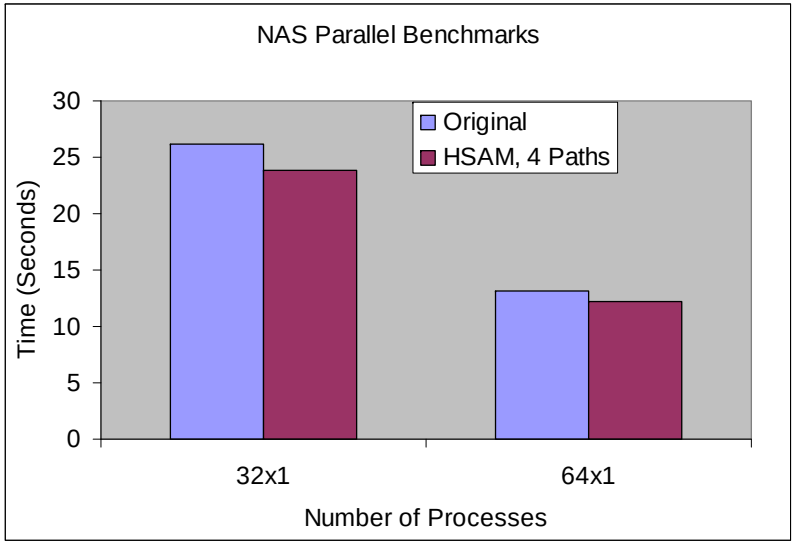
	RC (MVAPICH 0.9.8)				UD Design			
	Conn.	Buffers	Struct.	Total	Conn	Buffers	Struct	Total
512	22.9	65.0	0.3	88.2	0	37.0	0.2	37.2
1024	29.5	65.0	0.6	95.1	0	37.0	0.4	37.4
2048	42.4	65.0	1.2	107.4	0	37.0	0.9	37.9
4096	66.7	65.0	2.4	**134.1**	0	37.0	1.7	**38.7**



- Performance is enhanced considerably with UD
- Large number of communicating peers per process (992 at maximum)
  - UD reduces HCA cache thrashing
  - Very communication intensive
- 27 packet drops at 4K processes with 1.4 billion MPI messages
- Large difference in memory consumption, even only 1/4 of connections made

Hot-Spot Avoidance with MVAPICH

- Deterministic nature of InfiniBand routing leads to hot-spots in the network even with Fat-Tree
- Responsibility of path utilization is up to the MPI Library
- We Design HSAM (Hot-Spot Avoidance MVAPICH) to alleviate this problem



- For different FT Class benchmarks, performance improvement varies from 6-9 %

A. Vishnu, M. Koop, A. Moody, A. Mamidala, S. Narravula and D. K. Panda , " Hot-Spot Avoidance With Multi-Pathing Over InfiniBand: An MPI Perspective, " (CCGrid), Rio de Janeiro - Brazil, May 2007

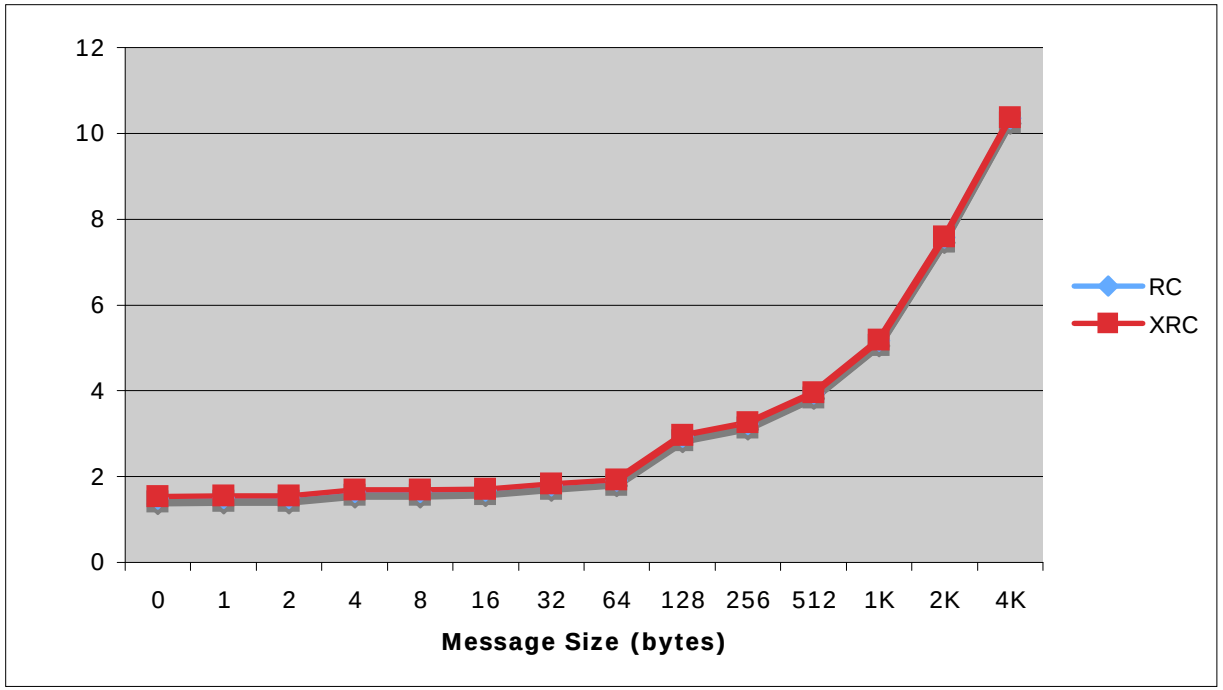
Presentation Overview

- Overview of MVAPICH/MVAPICH2 Project
- Features of MVAPICH 1.0 and MVAPICH2 1.0
- Sample Performance Numbers
  - Point-to-point (Mellanox and Qlogic)
  - Point-to-point with Intel-Connects Cable
  - Point-to-point with Obsidian IB-WAN
  - Multi-core-aware Optimized Collectives
  - UD-based Design
  - Hot-spot Avoidance Mechanism (HSAM)
- Upcoming Features and Issues
  - XRC support
  - Enhanced UD-based Design
  - Asynchronous Progress
  - Passive synchronization support
- Conclusions

XRC Support with ConnectX

- XRC (eXtended Reliable Connection) is being proposed for large-scale clusters
- We have designed and implemented an initial prototype of MVAPICH with XRC support
- In-depth results will be presented during tomorrow's XRC session

MVAPICH Latency: RC and XRC



- Results for latency are nearly identical between the use of RC and XRC transports
- 1.49usec for RC, 1.54usec for XRC

Enhanced UD-based Design

- Current UD-based design in MVAPICH 1.0 delivers good performance
- Some overheads on large-scale clusters for some applications
- Working on a new hybrid UD-RC design
- Delivers => better performance than RC or UD design
- Will be available in future releases

Asynchronous Progress

- Have added asynchronous progress (both at sender and receiver) in MVAPICH 1.0
- Allows to interrupt sender/receiver during long computation to handle communication
- Potential for overlap of computation and communication
- Carrying out performance evaluation with different applications to study the impact

Passive Synchronization for One-Sided Operations with Atomic Operations

- One-sided operations in MPI-2 semantics have two synchronization schemes
  - Active
  - Passive
- Have taken InfiniBand atomic operations into account to implement high performance and scalable passive synchronization
- Will be available in future releases

Conclusions

- MVAPICH and MVAPICH2 are being widely used in stable production IB clusters delivering best performance and scalability
- Also enabling clusters with iWARP support
- The user base stands at more than 580 organizations
- New features for scalability, high performance and fault tolerance support are aimed to deploy large-scale clusters (20K-50K) nodes in the near future

Acknowledgements

Our research is supported by the following organizations

- Current Funding support by

- Current Equipment support by

Acknowledgements

- Current Students
  - L. Chai (Ph.D.)
  - W. Huang (Ph.D.)
  - M. Koop (Ph.D.)
  - R. Kumar (M.S.)
  - A. Mamidala (Ph.D.)
  - S. Narravula (Ph.D.)
  - R. Noronha (Ph.D.)
  - G. Santhanaraman (Ph.D.)
  - K. Vaidyanathan (Ph.D.)
  - A. Vishnu (Ph.D.)
- Current Programmers
  - S. Rowland
  - J. Perkins
- Past Students
  - P. Balaji (Ph.D.)
  - D. Buntinas (Ph.D.)
  - S. Bhagvat (M.S.)
  - B. Chandrasekharan (M.S.)
  - W. Jiang (M.S.)
  - S. Kini (M.S.)
  - S. Krishnamoorthy (M.S.)
  - J. Liu (Ph.D.)
  - S. Sur (Ph.D.)
  - J. Wu (Ph.D.)
  - W. Yu (Ph.D.)

Web Pointers

MVAPICH Web Page
http://mvapich.cse.ohio-state.edu/

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