

•
•
•
•
•
•
•
•
•
•
•
•

OSU MPI (MVAPICH and MVAPICH2): Latest Status, Performance Numbers and Future Plans

Presentation at OpenFabrics Developers Summit
(Nov '06)

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 and Experience with OFED 1.1
- Selected Features of the latest MVAPICH/MVAPICH2 releases
 - SRQ, On-Demand and Scalability
 - Checkpoint/Restart
 - RDMA CM and iWARP
 - Multi-core-aware
 - Optimized collectives
- Upcoming Features and Issues
 - Overlap of Computation and Communication
 - Automatic Path Migration (APM)
 - Multi-Network Support with uDAPL
 - Congestion Avoidance Multi-Pathing
 - Messaging Rate
- Overview of Xen-IB Project
- Conclusions

MVAPICH/MVAPICH2 Software Distribution

- High Performance and Scalable Implementations
 - MPI-1 (MVAPICH)
 - MPI-2 (MVAPICH2)
- Has enabled a large number of **production IB clusters** all over the world to take advantage of IB
 - Sandia Thunderbird
 - LLNL Peloton
- Have been directly downloaded and used by more than **430 organizations worldwide**
- More details at <http://nowlab.cse.ohio-state.edu/projects/mpi-iba/>

Support for Multiple Interfaces/Adapters

- Gen2-IB
 - All IB adapters supporting Gen2
- uDAPL
 - Linux-IB
 - Solaris-IB
 - Neteffect 10GigE
 - support introduced in MVAPICH2 0.9.8
- Gen2-iWARP
 - Introduced in MVAPICH2 0.9.8
 - Tested with Chelsio (10GigE) and Ammasso (GigE)
- VAPI
 - All IB adapters supporting VAPI
- TCP/IP
 - Any adapter supporting TCP/IP interface
- Support for QLogic/PathScale at the PSM-level will be available soon



Excellent Performance with OFED 1.1

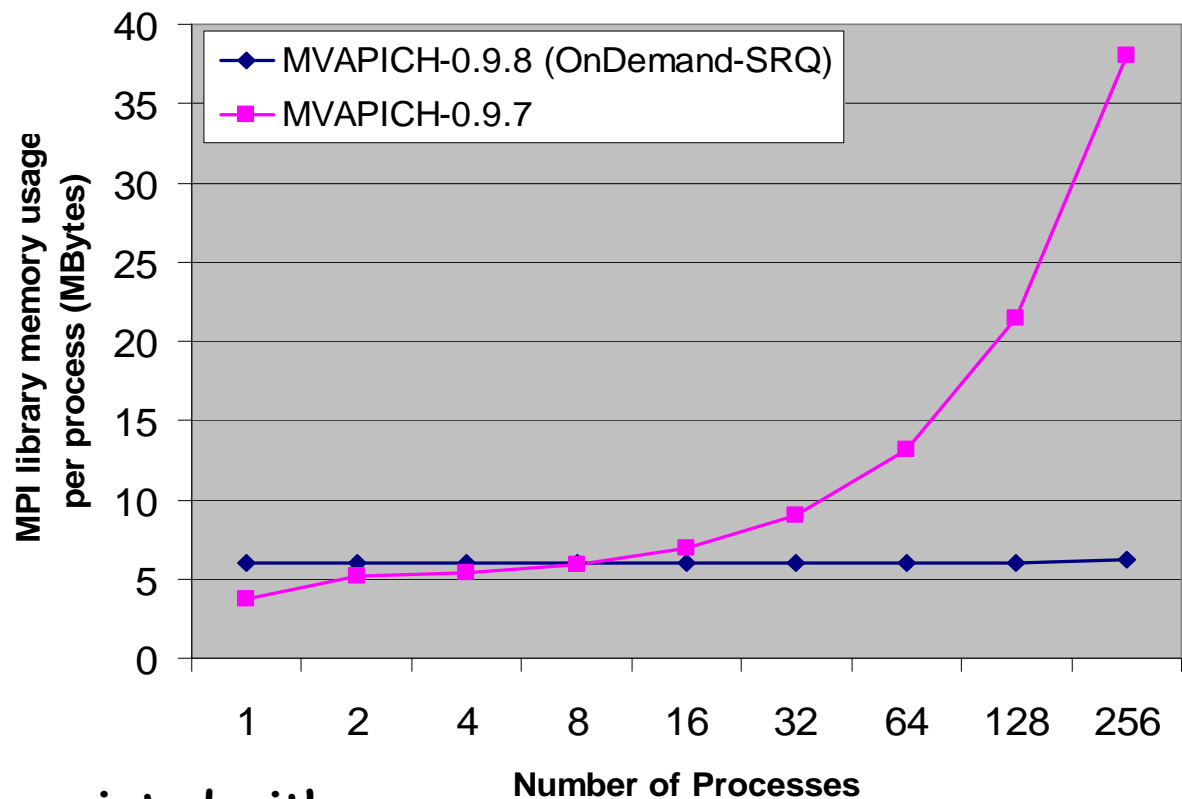


- EM64T Platform with single-rail DDR
 - Latency (4 bytes): 2.81 microsec
 - Bandwidth: 1561 MB/sec
 - Bi-directional Bandwidth: 2935 MB/sec
- EM64T Platform with dual-rail DDR
 - Latency (4 bytes): 2.81 microsec
 - Bandwidth: 3127 MB/sec
 - Bi-directional Bandwidth: 5917 MB/sec
- Performance on other platforms are available from MVAPICH web page

Presentation Overview

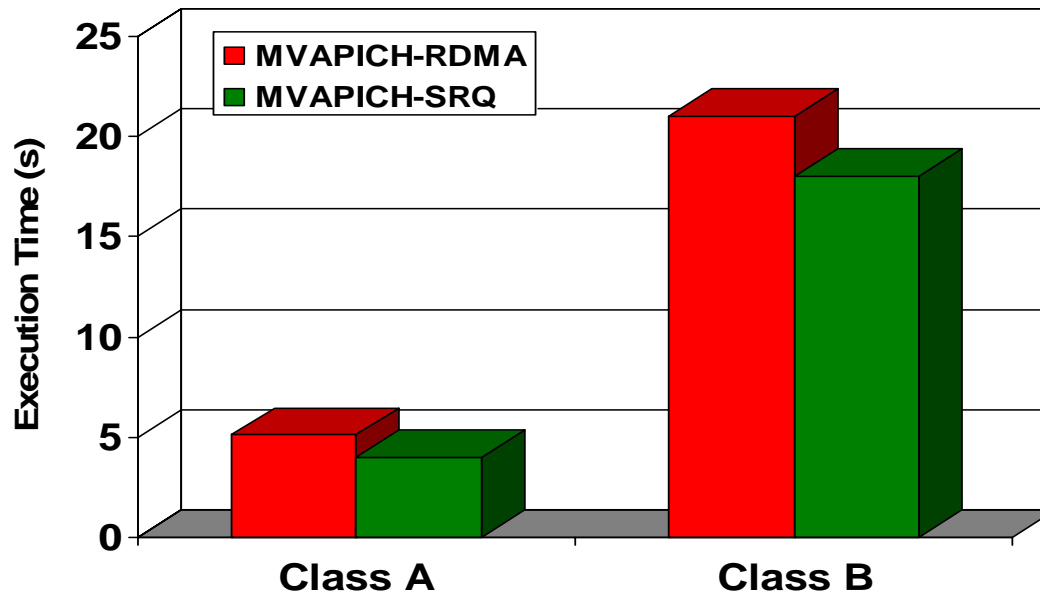
- Overview and Experience with OFED 1.1
- Selected Features of the latest MVAPICH/MVAPICH2 releases
 - SRQ, On-Demand and Scalability
 - Checkpoint/Restart
 - RDMA CM and iWARP
 - Multi-core-aware
 - Optimized collectives
- Upcoming Features and Issues
 - Overlap of Computation and Communication
 - Automatic Path Migration (APM)
 - Multi-Network Support with uDAPL
 - Congestion Avoidance Multi-Pathing
 - Messaging Rate
- Overview of Xen-IB Project
- Conclusions

Memory Usage with OnDemand-SRQ



SRQ is associated with Flow Control

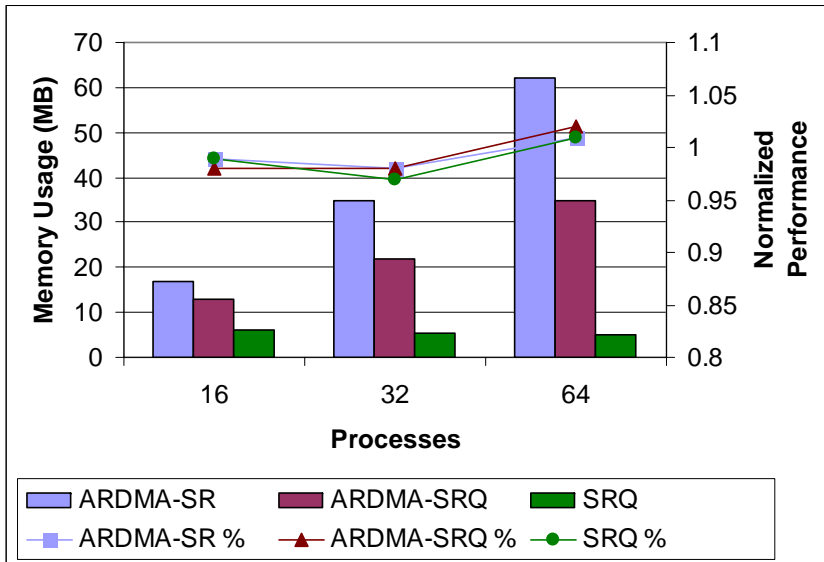
NAS LU Performance



- NAS LU Class A and B are run on 32 processes on Cluster B
- MVAPICH-SRQ outperforms MVAPICH-RDMA by 22% for Class B
- LU has mainly short messages
 - Larger available window at receiver for MVAPICH-SRQ leads to benefits
 - Reduced polling time for MVAPICH-SRQ compared to MVAPICH-RDMA

8

NAMD (apoa1)



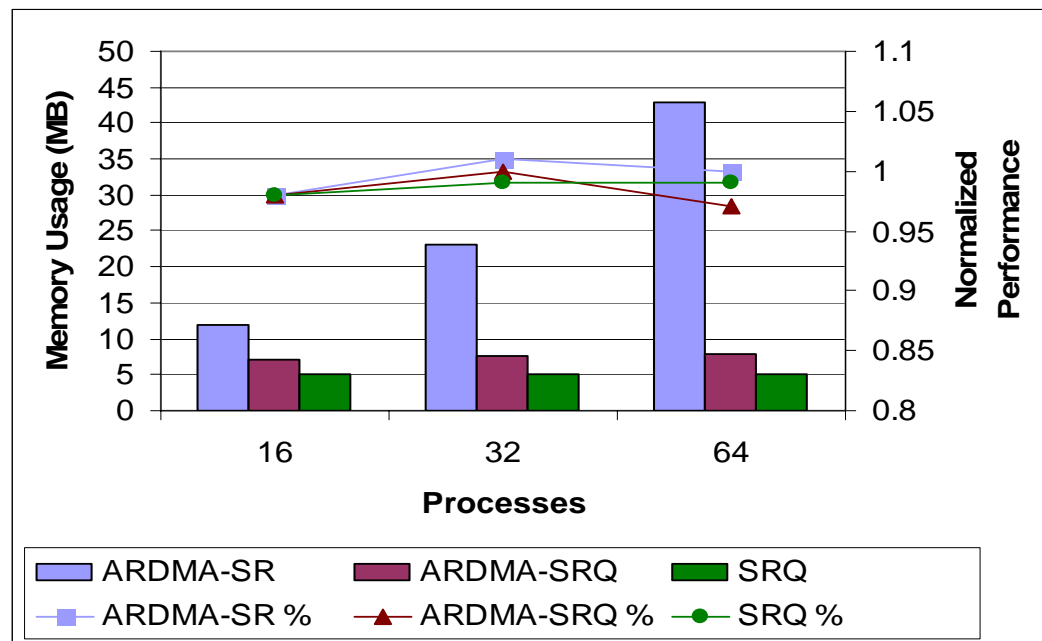
Avg. RDMA channels	53.15
Avg. Low watermarks	0.03
Unexpected Msgs (%)	48.2
Total Messages	3.7e6
MPI Time (%)	23.54

NAMD on 64 nodes

- Technical Paper presented at SC '06
- 50% messages < 128 Bytes, other 50% between 128 Bytes and 32 KB
- There are 53 RDMA connections setup for 64 process experiment
- SRQ Channel takes 5-6MB of memory
- Memory required by SRQ channel decreases by 1MB going from 16 to 64

9

High-Performance Linpack (HPL)



- 50% messages < 128 Bytes, mainly control messages
- There are only around 6 RDMA connections setup for 64 processes
- SRQ Channel takes 5-6MB of memory
- Messages are not sent in bursts, so there are no “low-watermark” events

Fault Tolerance

- Component failures are the norm in large-scale clusters
- Imposes need on reliability and fault tolerance
- Working along the following three angles
 - End-to-end Reliability with memory-to-memory CRC
 - Available since *MVAPICH 0.9.7*
 - Application transparent Process Fault Tolerance with Efficient Checkpoint and Restart
 - Available in *MVAPICH2 0.9.8*
 - Reliable Networking with Automatic Path Migration (APM) utilizing Redundant Communication Paths
 - Will be available soon
 - uDAPL-based Network Fault-Tolerance
 - Will be available soon

Checkpoint/Restart Support in MVAPICH2 0.9.8

- Process-level Fault Tolerance
 - User-transparent, system-level checkpointing
 - Based on BLCR from LBNL to take coordinated checkpoints of entire program, including front end and individual processes
 - Designed novel schemes to
 - Coordinate all MPI processes to drain all in flight messages in IB connections
 - Store communication state and buffers, etc. while taking checkpoint
 - Restarting from the checkpoint
 - Tested with NFS, PVFS2, Ext3 (local disk)

A Running Example (Cont.)

Terminal A:
Start running LU

```
[gaoq@c5-gen2 test]$ mpirun -n 4 -cr_file /tmp/save ./lu.A.4

NAS Parallel Benchmarks 3.2 -- LU Benchmark

Size: 64x 64x 64
Iterations: 250
Number of processes: 4

Time step 1
Time step 20
```

1

Terminal B:
Get its PID

```
xfs 2990 1 0 Feb04 ? 00:00:00 xfs -droppriv -daemon
daemon 3009 1 0 Feb04 ? 00:00:00 /usr/sbin/atd
root 3033 1 0 Feb04 ? 00:00:00 cups-config-daemon
root 3075 1 0 Feb04 tty1 00:00:00 /sbin/mingetty tty1
root 3076 1 0 Feb04 tty2 00:00:00 /sbin/mingetty tty2
root 3077 1 0 Feb04 tty3 00:00:00 /sbin/mingetty tty3
root 3078 1 0 Feb04 tty4 00:00:00 /sbin/mingetty tty4
root 3079 1 0 Feb04 tty5 00:00:00 /sbin/mingetty tty5
root 3080 1 0 Feb04 tty6 00:00:00 /sbin/mingetty tty6
root 10204 9 0 Feb04 ? 00:00:00 [pdflush]
root 10387 9 0 Feb04 ? 00:00:00 [pdflush]
root 11341 1 0 04:02 ? 00:00:00 cupsd
root 14453 2733 0 10:44 ? 00:00:00 sshd: gaoq [priv]
gaoq 14455 14453 0 10:44 ? 00:00:00 sshd: gaoq@pts/0
gaoq 14456 14455 0 10:44 pts/0 00:00:00 -bash
root 14595 2733 0 12:17 ? 00:00:00 sshd: gaoq [priv]
gaoq 14597 14595 0 12:17 ? 00:00:00 sshd: gaoq@pts/1
gaoq 14598 14597 0 12:17 pts/1 00:00:00 -bash
gaoq 14846 1 0 12:21 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
root 14870 2733 0 12:22 ? 00:00:00 sshd: gaoq [priv]
gaoq 14872 14870 0 12:22 ? 00:00:00 sshd: gaoq@pts/2
gaoq 14873 14872 0 12:22 pts/2 00:00:00 -bash
root 14923 2733 0 12:26 ? 00:00:00 sshd: gaoq [priv]
gaoq 14925 14923 0 12:26 ? 00:00:00 sshd: gaoq@pts/3
gaoq 14926 14925 0 12:26 pts/3 00:00:00 -bash
root 14952 2733 0 12:27 ? 00:00:00 sshd: gaoq [priv]
gaoq 14954 14952 0 12:27 ? 00:00:00 sshd: gaoq@pts/4
gaoq 14955 14954 0 12:27 pts/4 00:00:00 -bash
gaoq 15374 1 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15377 14926 0 12:55 pts/3 00:00:00 mpirun -n 4 -cr_file /tmp/save ./lu.A.4
gaoq 15379 15377 0 12:55 pts/3 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15380 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15381 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15382 15381 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15383 15380 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15389 14955 0 12:56 pts/4 00:00:00 ps -ef
[gaoq@c5-gen2 test]$
```

2

A Running Example (Cont.)

Terminal A:
LU is running

```
[gaoq@c5-gen2 test]$ mpirun -n 4 -cr_file /tmp/save ./lu.A.4

NAS Parallel Benchmarks 3.2 -- LU Benchmark

Size: 64x 64x 64
Iterations: 250
Number of processes: 4

Time step 1
Time step 20
Time step 40
Time step 60
Time step 80
Time step 100
Time step 120
Time step 140
```

3

Terminal B:
Now, Take checkpoint

```
root 3033 1 0 Feb04 ? 00:00:00 cups-config-daemon
root 3075 1 0 Feb04 tty1 00:00:00 /sbin/mingetty tty1
root 3076 1 0 Feb04 tty2 00:00:00 /sbin/mingetty tty2
root 3077 1 0 Feb04 tty3 00:00:00 /sbin/mingetty tty3
root 3078 1 0 Feb04 tty4 00:00:00 /sbin/mingetty tty4
root 3079 1 0 Feb04 tty5 00:00:00 /sbin/mingetty tty5
root 3080 1 0 Feb04 tty6 00:00:00 /sbin/mingetty tty6
root 10204 9 0 Feb04 ? 00:00:00 [pdflush]
root 10387 9 0 Feb04 ? 00:00:00 [pdflush]
root 11341 1 0 04:02 ? 00:00:00 cupsd
root 14453 2733 0 10:44 ? 00:00:00 sshd: gaoq [priv]
gaoq 14455 14453 0 10:44 ? 00:00:00 sshd: gaoq@pts/0
gaoq 14456 14455 0 10:44 pts/0 00:00:00 -bash
root 14595 2733 0 12:17 ? 00:00:00 sshd: gaoq [priv]
gaoq 14597 14595 0 12:17 ? 00:00:00 sshd: gaoq@pts/1
gaoq 14598 14597 0 12:17 pts/1 00:00:00 -bash
gaoq 14846 1 0 12:21 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
root 14870 2733 0 12:22 ? 00:00:00 sshd: gaoq [priv]
gaoq 14872 14870 0 12:22 ? 00:00:00 sshd: gaoq@pts/2
gaoq 14873 14872 0 12:22 pts/2 00:00:00 -bash
root 14923 2733 0 12:26 ? 00:00:00 sshd: gaoq [priv]
gaoq 14925 14923 0 12:26 ? 00:00:00 sshd: gaoq@pts/3
gaoq 14926 14925 0 12:26 pts/3 00:00:00 -bash
root 14952 2733 0 12:27 ? 00:00:00 sshd: gaoq [priv]
gaoq 14954 14952 0 12:27 ? 00:00:00 sshd: gaoq@pts/4
gaoq 14955 14954 0 12:27 pts/4 00:00:00 -bash
gaoq 15374 1 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15377 14926 0 12:55 pts/3 00:00:00 mpirun -n 4 -cr_file /tmp/save ./lu.A.4
gaoq 15379 15377 0 12:55 pts/3 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15380 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15381 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15382 15381 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15383 15380 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15389 14955 0 12:56 pts/4 00:00:00 ps -ef
[gaoq@c5-gen2 test]$ checkpoint 15377
Checkpoint Done
[gaoq@c5-gen2 test]$
```

4

A Running Example (Cont.)

Terminal A:
LU is not affected.
Stop it using CTRL-C

```
[gaoq@c5-gen2 test]$ mpirun -n 4 -cr_file /tmp/save ./lu.A.4

NAS Parallel Benchmarks 3.2 -- LU Benchmark

Size: 64x 64x 64
Iterations: 250
Number of processes: 4

Time step 1
Time step 20
Time step 40
Time step 60
Time step 80
Time step 100
Time step 120
Time step 140
Time step 160
Time step 180
Time step 200
CTRL+C Caught... exiting
[gaoq@c5-gen2 test]$
```

5

Terminal B:
Then, restart from
the checkpoint

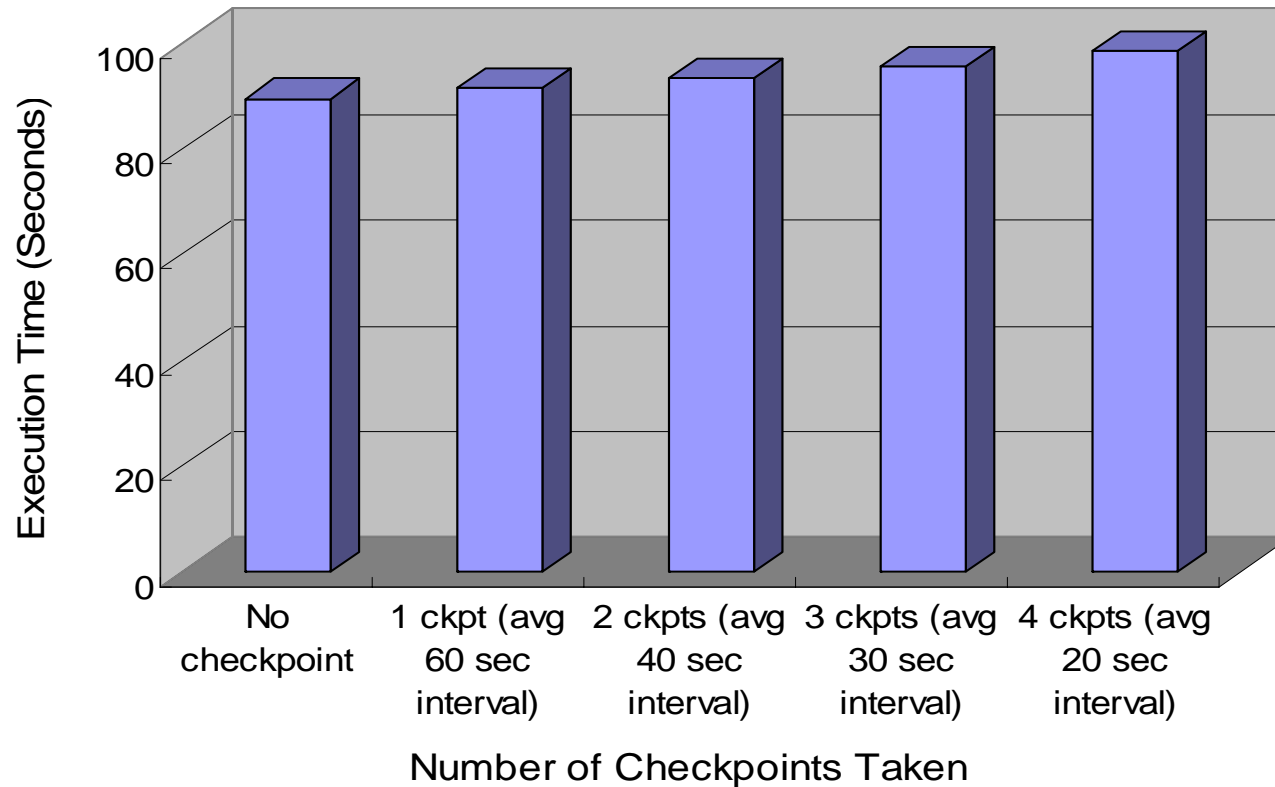
```
root 3078 1 0 Feb04 tty4 00:00:00 /sbin/mingetty tty4
root 3079 1 0 Feb04 tty5 00:00:00 /sbin/mingetty tty5
root 3080 1 0 Feb04 tty6 00:00:00 /sbin/mingetty tty6
root 10204 9 0 Feb04 ? 00:00:00 [pdflush]
root 10387 9 0 Feb04 ? 00:00:00 [pdflush]
root 11341 1 0 04:02 ? 00:00:00 cupsd
root 14453 2733 0 10:44 ? 00:00:00 sshd: gaoq [priv]
gaoq 14455 14453 0 10:44 ? 00:00:00 sshd: gaoq@pts/0
gaoq 14456 14455 0 10:44 pts/0 00:00:00 -bash
root 14595 2733 0 12:17 ? 00:00:00 sshd: gaoq [priv]
gaoq 14597 14595 0 12:17 ? 00:00:00 sshd: gaoq@pts/1
gaoq 14598 14597 0 12:17 pts/1 00:00:00 -bash
root 14846 1 0 12:21 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
root 14870 2733 0 12:22 ? 00:00:00 sshd: gaoq [priv]
gaoq 14872 14870 0 12:22 ? 00:00:00 sshd: gaoq@pts/2
gaoq 14873 14872 0 12:22 pts/2 00:00:00 -bash
root 14923 2733 0 12:26 ? 00:00:00 sshd: gaoq [priv]
gaoq 14925 14923 0 12:26 ? 00:00:00 sshd: gaoq@pts/3
gaoq 14926 14925 0 12:26 pts/3 00:00:00 -bash
root 14952 2733 0 12:27 ? 00:00:00 sshd: gaoq [priv]
gaoq 14954 14952 0 12:27 ? 00:00:00 sshd: gaoq@pts/4
gaoq 14955 14954 0 12:27 pts/4 00:00:00 -bash
gaoq 15374 1 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15377 14926 0 12:55 pts/3 00:00:00 mpirun -n 4 -cr_file /tmp/save ./lu.A.4
gaoq 15379 15377 0 12:55 pts/3 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15380 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15381 15374 0 12:55 ? 00:00:00 python2.3 /home/3/gaoq/tasks/MVAPICH2-CR/install-cr-
gaoq 15382 15381 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15383 15380 97 12:55 ? 00:00:42 ./lu.A.4
gaoq 15389 14955 0 12:56 pts/4 00:00:00 ps -ef
[gaoq@c5-gen2 test]$ checkpoint 15377
Checkpoint Done
[gaoq@c5-gen2 test]$ restart checkpoint_file
Time step 160
Time step 180
Time step 200
```

6

15

Checkpoint/Restart Performance with PVFS2

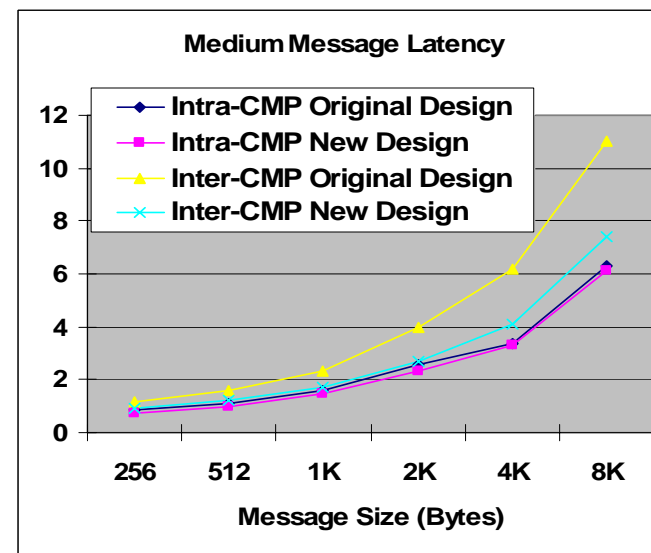
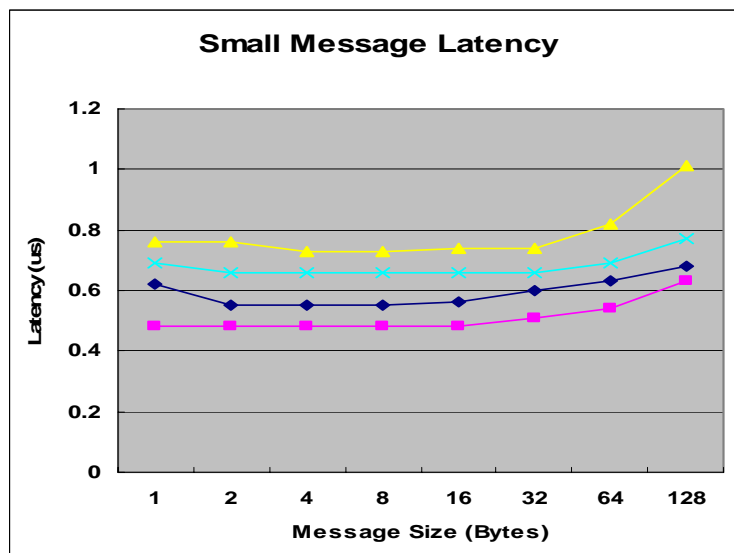
NAS, LU Class C, 32x1 (Storage: 8 PVFS2 servers on IPoIB)



RDMA CM and iWARP Support

- Available in MVAPICH2 0.9.8
- RDMA CM is supported for both
 - IB
 - iWARP
- Plan to carry out performance evaluation of RDMA CM support
- iWARP support is tested with
 - Chelsio
 - Ammasso

Multi-core-Aware MVAPICH2 Design

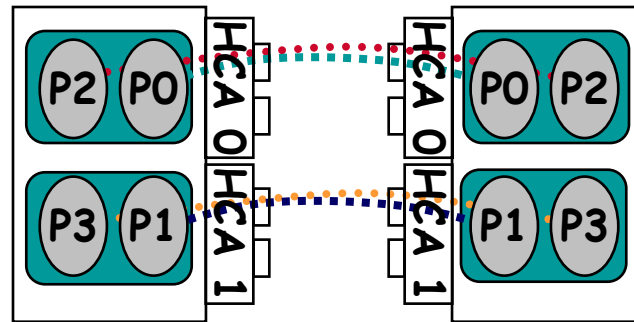


- Dual dual-core AMD Opteron processors, 2.0GHz, 1MB L2 cache
- The new design improves inter-CMP latency for all the messages
Similar benefits for bandwidth also
- Available in MVAPICH2 0.9.8

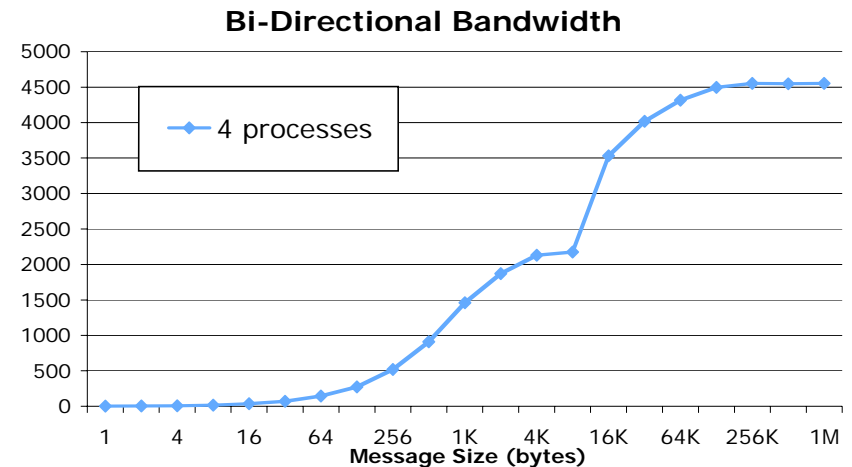
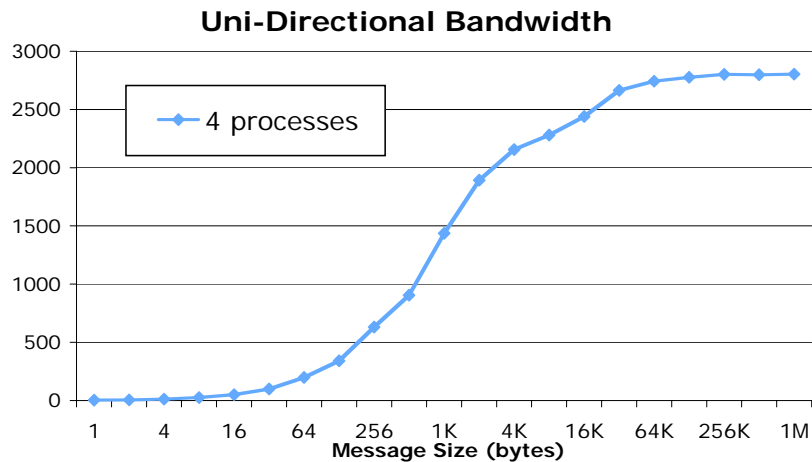
L. Chai, A. Hartono and D. K. Panda, Designing High Performance and Scalable MPI Intra-node Communication Support for Clusters, presented at Cluster '06

MPI over InfiniBand Performance

(Dual-core Intel Woodcrest Systems with PCI-Express and Dual-Rail DDR InfiniBand)

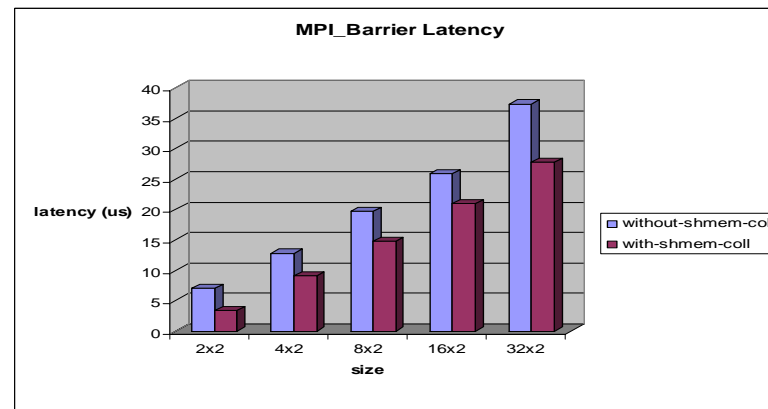
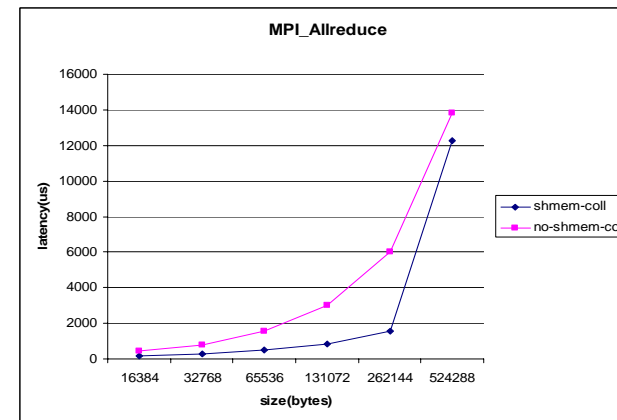
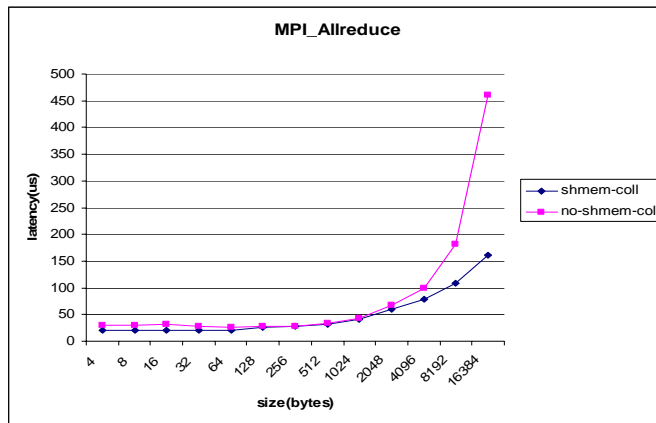


4-processes on each node concurrently communicating over Dual-rail InfiniBand DDR (Mellanox)



M. J. Koop, W. Huang, A. Vishnu and D. K. Panda, Memory Scalability Evaluation of Next Generation Intel Bensley Platform with InfiniBand, Hot Interconnect Symposium (Aug. 2006)

Optimizing Collectives based on Shared Memory (Allreduce, Reduce and Barrier)



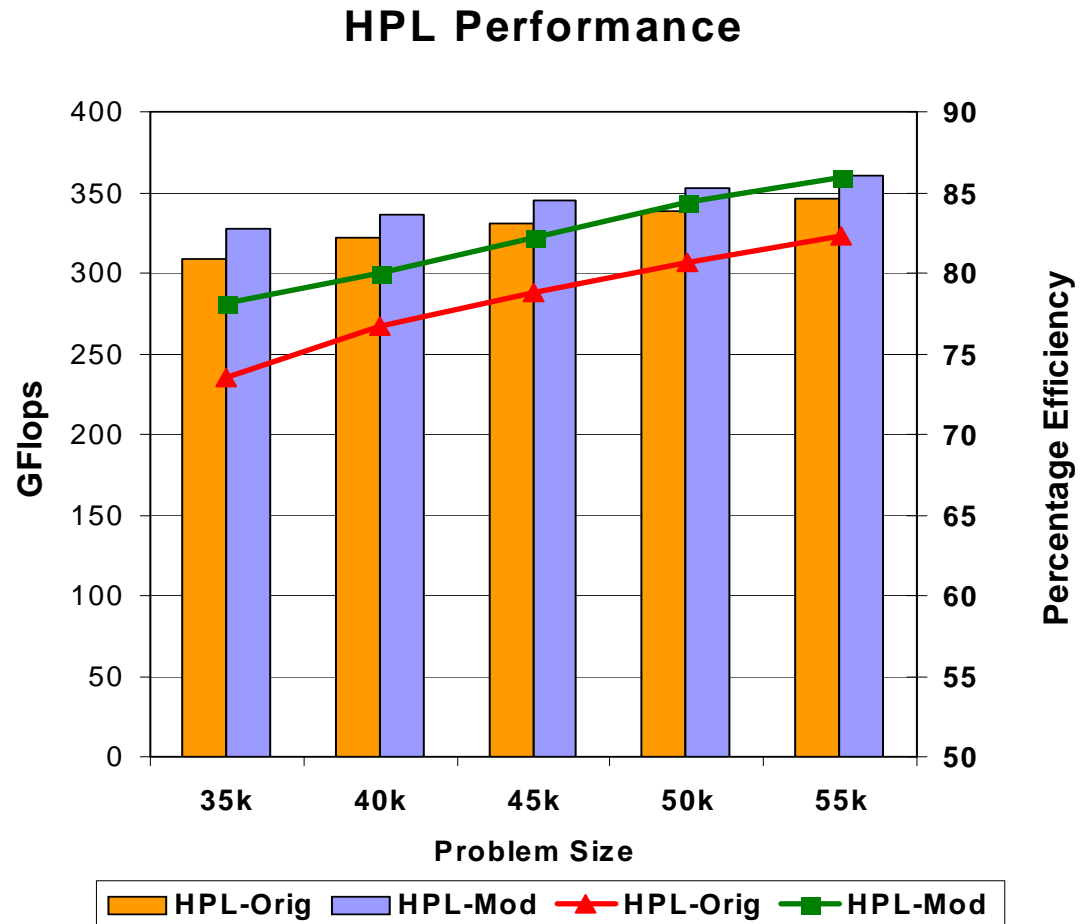
- Introduced in MVAPICH2 0.9.8

Presentation Overview

- Overview and Experience with OFED 1.1
- Selected Features of the latest MVAPICH/MVAPICH2 releases
 - SRQ, On-Demand and Scalability
 - Checkpoint/Restart
 - RDMA CM and iWARP
 - Multi-core-aware
 - Optimized collectives
- Upcoming Features and Issues
 - Overlap of Computation and Communication
 - Automatic Path Migration (APM)
 - Multi-Network Support with uDAPL
 - Congestion Avoidance with Multi-Pathing
 - Messaging Rate
- Overview of Xen-IB Project
- Conclusions

Enhancing Overlap Capabilities in HPL

- MVAPICH has RDMA Read support
- RDMA Read with Interrupt can provide
 - Asynchronous progress
 - Overlap of computation and communication
- Have enhanced HPL to add overlapping at the sender side
- Results on 32 dual dual-core nodes with IB DDR
- MPI overlap increase the overall application efficiency by 5-6%
- Improvement rate consistent with increasing problem size



Network-Level Fault Tolerance with APM

- Designed a solution using InfiniBand Automatic Path Migration (APM) Hardware mechanism
 - Utilizes Redundant Communication Paths
 - Multiple Ports
 - LMC
- APM support available in Gen2 trunk only (not with OFED)

Screenshots: APM with OSU Bandwidth test

Step #1: Bandwidth Test Running

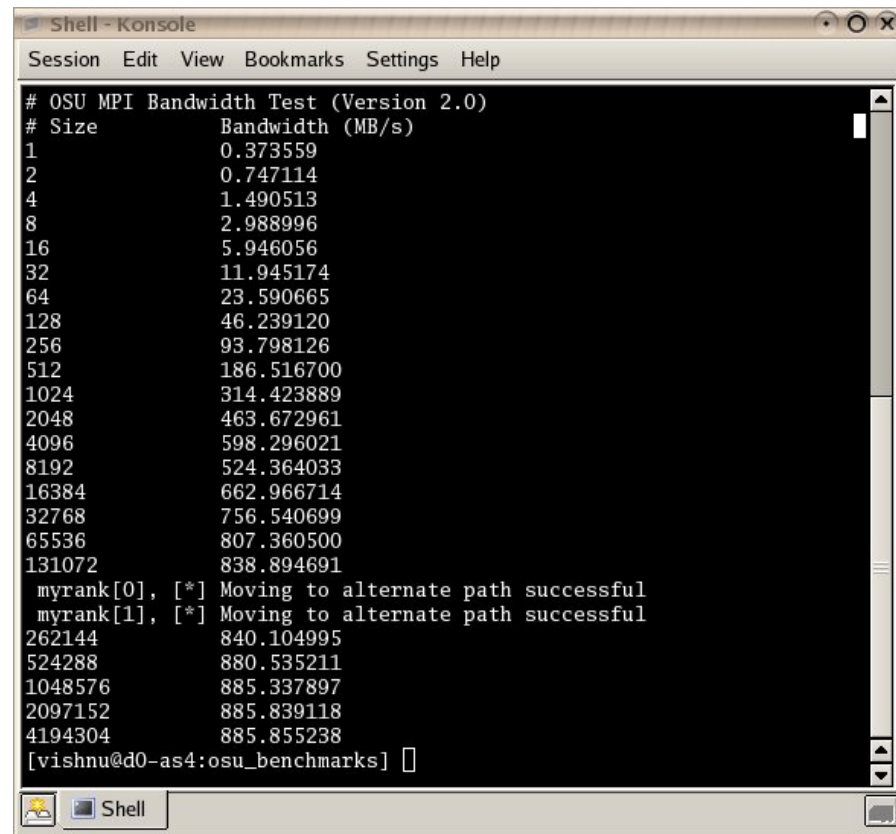
```
Shell - Konsole
Session Edit View Bookmarks Settings Help
[vishnu@d0-as4:osu_benchmarks] ../bin/mpicc osu_bw.c -o bw
[vishnu@d0-as4:osu_benchmarks] ../bin/mpirun_rsh -np 2 d0 d2 ./bw
# OSU MPI Bandwidth Test (Version 2.0)
# Size      Bandwidth (MB/s)
1           0.373559
2           0.747114
4           1.490513
8           2.988996
16          5.946056
32          11.945174
64          23.590665
128         46.239120
256         93.798126
512         186.516700
1024        314.423889
2048        463.672961
4096        598.296021
8192        524.364033
16384       662.966714
32768       756.540699
65536       807.360500
[]
```

Step #2: Fault on Link, APM Triggered

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
[vishnu@d0-as4:osu_benchmarks] ../bin/mpicc osu_bw.c -o bw
[vishnu@d0-as4:osu_benchmarks] ../bin/mpirun_rsh -np 2 d0 d2 ./bw
# OSU MPI Bandwidth Test (Version 2.0)
# Size      Bandwidth (MB/s)
1           0.373559
2           0.747114
4           1.490513
8           2.988996
16          5.946056
32          11.945174
64          23.590665
128         46.239120
256         93.798126
512         186.516700
1024        314.423889
2048        463.672961
4096        598.296021
8192        524.364033
16384       662.966714
32768       756.540699
65536       807.360500
131072      838.894691
myrank[0], [*] Moving to alternate path successful
myrank[1], [*] Moving to alternate path successful
262144      840.104995
[]
```


Screenshots: APM with OSU Bandwidth test

Step #3:
Bandwidth Test
Resumes and
Finishes

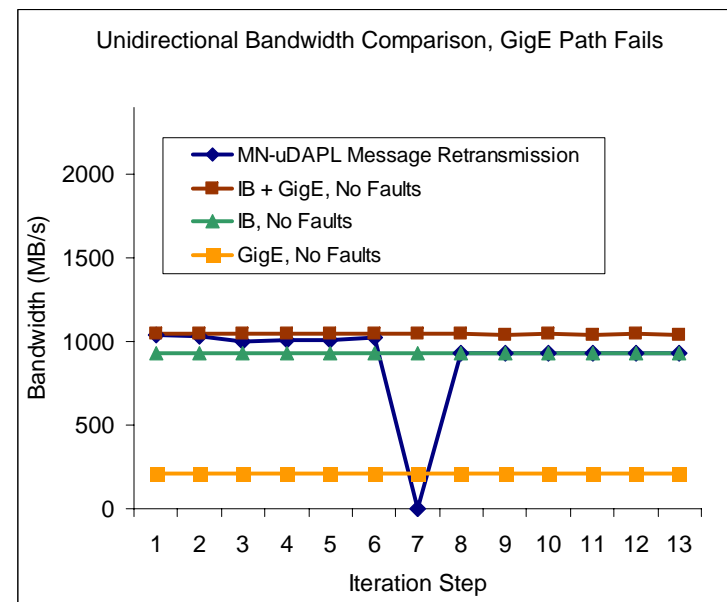
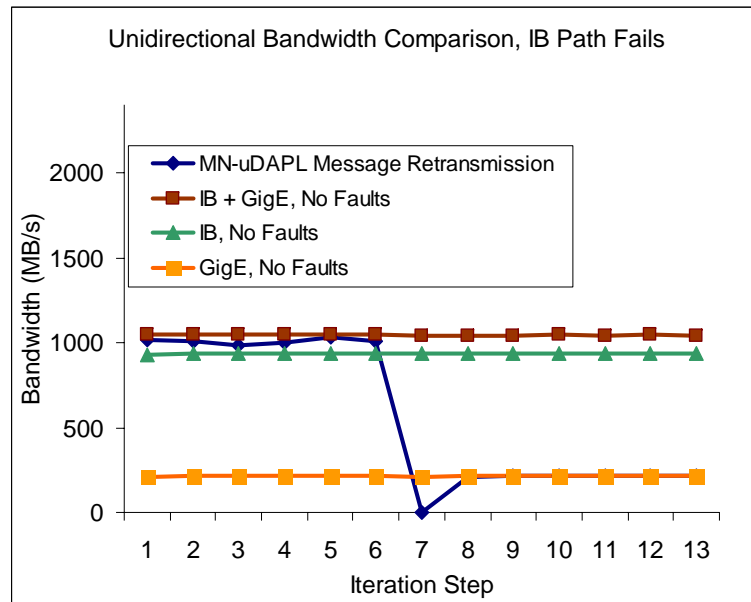


```
Shell - Konsole
Session Edit View Bookmarks Settings Help
# OSU MPI Bandwidth Test (Version 2.0)
# Size      Bandwidth (MB/s)
1           0.373559
2           0.747114
4           1.490513
8           2.988996
16          5.946056
32          11.945174
64          23.590665
128         46.239120
256         93.798126
512         186.516700
1024        314.423889
2048        463.672961
4096        598.296021
8192        524.364033
16384       662.966714
32768       756.540699
65536       807.360500
131072      838.894691
myrank[0], [*] Moving to alternate path successful
myrank[1], [*] Moving to alternate path successful
262144      840.104995
524288      880.535211
1048576     885.337897
2097152     885.839118
4194304     885.855238
[vishnu@d0-as4:osu_benchmarks] █
```

Multi-Network Support using uDAPL

- Network-independent interfaces like uDAPL are being available
- Can we design a unified MPI framework, with low overhead, flexibility, and adaptivity to support following
 - Network Heterogeneity
 - Network Failover
 - Asynchronous recovery of previously failed paths

Network Failover

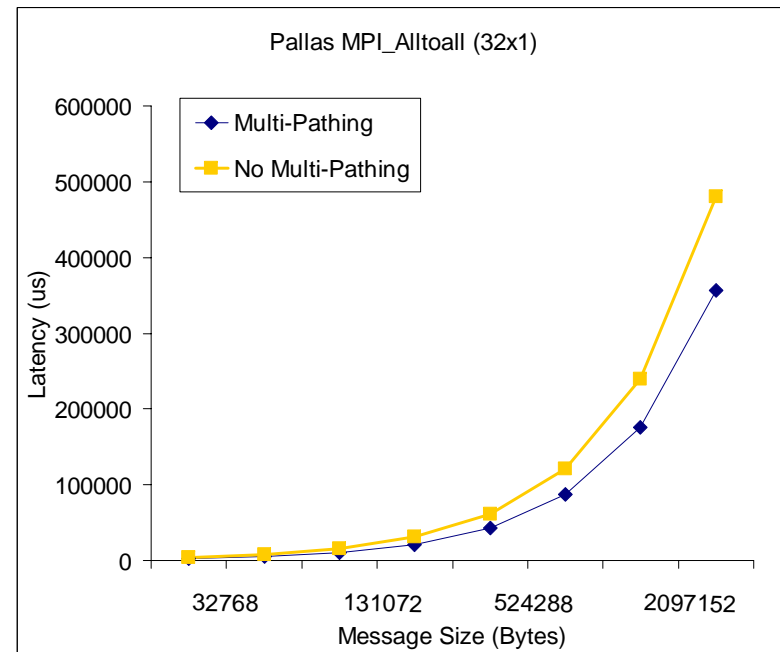
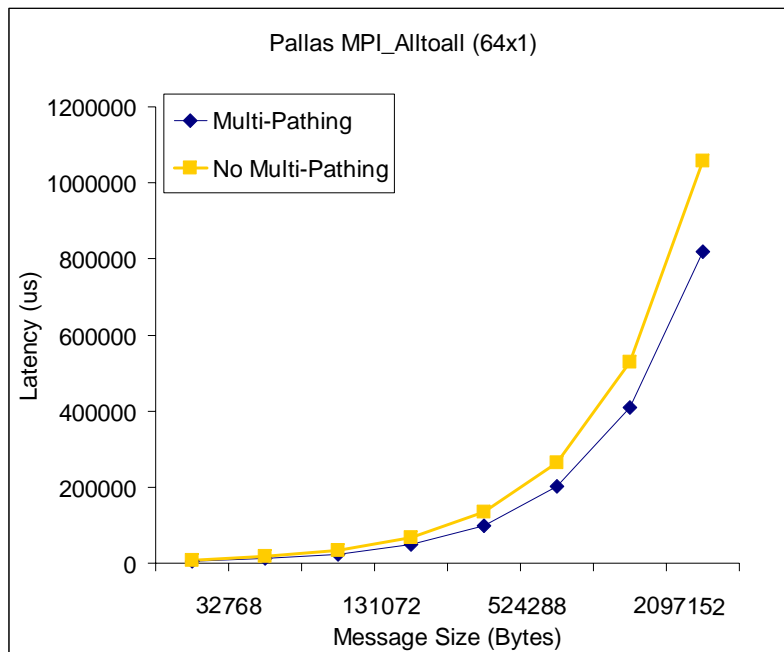


- Presented at SC '06 Technical Paper
- The peak bandwidth achieved after failover is same as achievable in no-faults case

Congestion Avoidance with Multi-Pathing

- Large scale clusters may not be complete fat tree
 - Congestion due to absence of CBB
- Location of MPI tasks in a job impact the overall performance
 - Static selection of paths may not work well for different MPI task allocations
- The situation becomes more complicated
 - Different communication patterns in same application
 - Different collective communication algorithms
 - Interaction due to other jobs in the cluster
- Can we design an adaptive scheme to take care of above scenarios?

Performance Evaluation with Multi-Pathing



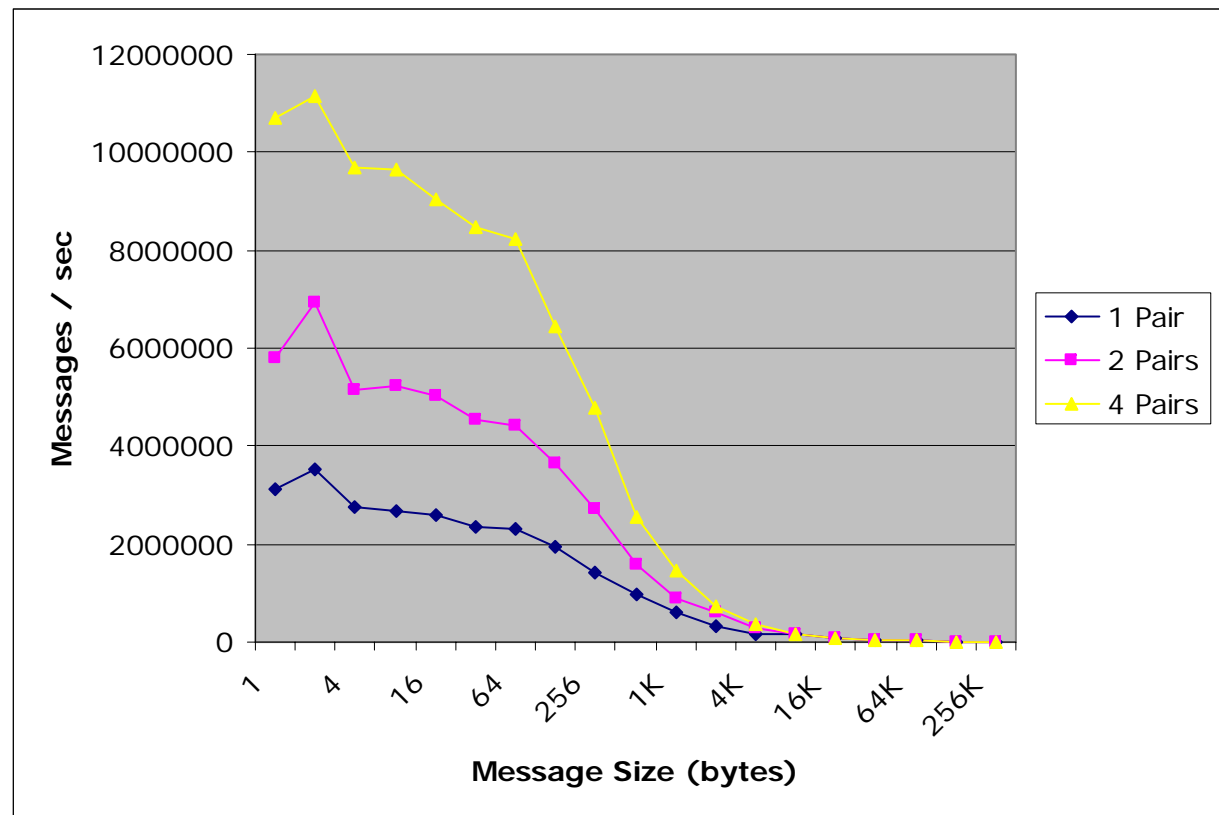
- Multi-pathing with LMC improves the performance of MPI_Alltoall
 - 27% for 32x1 evaluation testbed
 - 23% for 64x1 evaluation testbed
- For clusters with multi-thousand scale, more benefits are expected

QoS, Routing and Advanced Features

- As multi-thousand nodes with IB are deployed, many open challenges exist for
 - Usage of SL for traffic differentiation
 - Pt-to-pt and collective
 - Identifying optimal paths in the fabric
 - Support adaptive routing
 - Carrying out topology-aware collective operations
 - UD-based communication
 - Exploiting reliable multicast support (when available)
 - Complementing on-demand connection with teardown
 - Releasing unused communication memory resources
- Carrying out research on these angles and solutions will be available soon

Messaging Rate

- Design based on MVAPICH 0.9.8
- Preliminary performance results
 - Dual dual-core woodcrest
 - Single DDR card
- Around 11M Messages/sec



Presentation Overview

- Overview and Experience with OFED 1.1
- Selected Features of the latest MVAPICH/MVAPICH2 releases
 - SRQ, On-Demand and Scalability
 - Checkpoint/Restart
 - RDMA CM and iWARP
 - Multi-core-aware
 - Optimized collectives
- Upcoming Features and Issues
 - Overlap of Computation and Communication
 - Automatic Path Migration (APM)
 - Multi-Network Support with uDAPL
 - Congestion Avoidance with Multi-Pathing
 - Messaging Rate
- Overview of Xen-IB Project
- Conclusions

Xen-IB: Virtualizing InfiniBand in Xen

Design Overview:

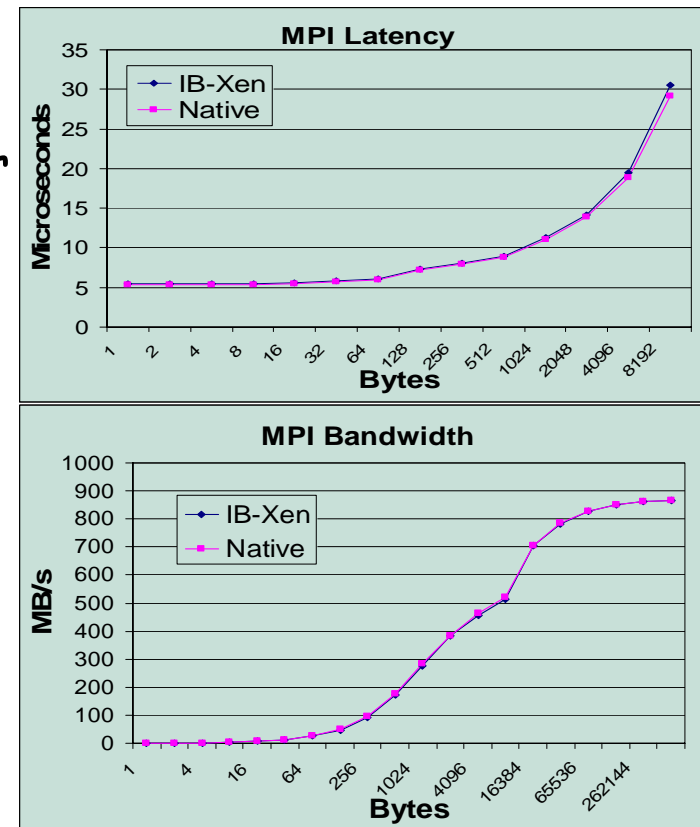
- Follows Xen split driver model
- Same IB-Gen2 Verbs Interface for applications in guest domains (domU)

Implementation:

- Prototype based on Gen2 stack
- Close to native performance

Three Publications:

- USENIX '06
- ICS '06
- NSDI '07 (Efficient support for Migration), under review



Conclusions

- MVAPICH and MVAPICH2 are being widely used in stable production IB clusters delivering best performance
- The user base stands at more than 430 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
- Also enabling clusters with iWARP support
- Access to larger cluster (as suggested by Matt) will be helpful

Acknowledgments

- Current Students
 - Lei Chai (PhD)
 - Qi Gao (PhD)
 - Wei Huang (PhD)
 - Matthew Koop (PhD)
 - Amith Mamidala (PhD)
 - Sundeep Narravula (PhD)
 - Ranjit Noronha (PhD)
 - G. Santhanaraman (PhD)
 - Sayantan Sur (PhD)
 - K. Vaidyanathan (PhD)
 - Abhinav Vishnu (PhD)
- Current Programmers
 - Shaun Rowland
 - Jonathan Perkins
- Past Post-Doc
 - Hyun-Wook Jin
- Past Students
 - Pavan Balaji (PhD)
 - Sitha Bhagvat (MS)
 - D. Buntinas (PhD)
 - B. Chandrasekharan (MS)
 - Weihang Jiang (MS)
 - Sushmita Kini (MS)
 - S. Krishnamoorthy (MS)
 - Jiuxing Liu (PhD)
 - Jiesheng Wu (PhD)
 - Weikuan Yu (PhD)

Web Pointers



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

MVAPICH Web Page

<http://nowlab.cse.ohio-state.edu/projects/mpi-iba/>

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