Performance and Functional Improvements in MPT software for the XT

Mark Pagel

pags@cray.com

May 7, 2007

CRAY

THE SUPERCOMPUTER COMPANY

Outline

- Latest Cray XT MPI Performance Improvements
  - Portals Improvements
  - New MPI env variables
  - SHMEM performance improvements
- Latest Cray XT MPI Functional Improvements
- Future Cray XT MPI Performance Improvements
- Future Cray XT MPI Functional Improvements

Latest Cray XT MPI Performance Improvements

- Portals improvements (1.5.07, 1.4.28)
  - Send to self short-circuit optimizations
  - Symmetric portals syscall optimizations
  - Portals API extended（PtlMEMDPost）
- MPI use of PtlMEMDPost（1.5.07, 1.4.28）
- New MPI env variables
  - MPICH_RANK_REORDER_METHOD (1.5.08 and 1.4.30)
  - MPI_COLL_OPT_ON (1.5.11 and 1.4.32)
  - MPICH_FAST_MEMCPY (1.5.30 and 1.4.46)
  - MPICH_PTL_MATCH_OFF (1.5.39 and 1.4.50)

MPICH_RANK_REORDER_METHOD

- MPICH_RANK_REORDER_METHOD env variable to control rank placement (1.5.08 and 1.4.30)

  - yod default placement:

NODE	0	1	2	3
RANK	0&4	1&5	2&6	3&7

  - Setting env to “1” causes SMP style placement

NODE	0	1	2	3
RANK	0&1	2&3	4&5	6&7

MPICH_RANK_REORDER_METHOD (cont.)

- Setting env to “2” causes folded rank placement

NODE	0	1	2	3
RANK	0&7	1&6	2&5	3&4

- Setting env to “3” causes custom rank placement using “MPICH_RANK_ORDER” file. For example:

0-15 Places the ranks in SMP-style order

15-0 Places ranks 15&14 on the first node, 13&12 on next, etc.

0,4,1,5,2,6,3,7 Places ranks 0&4 on the first node, 1&5 on the next, 2&6 together, and 3&7 together.

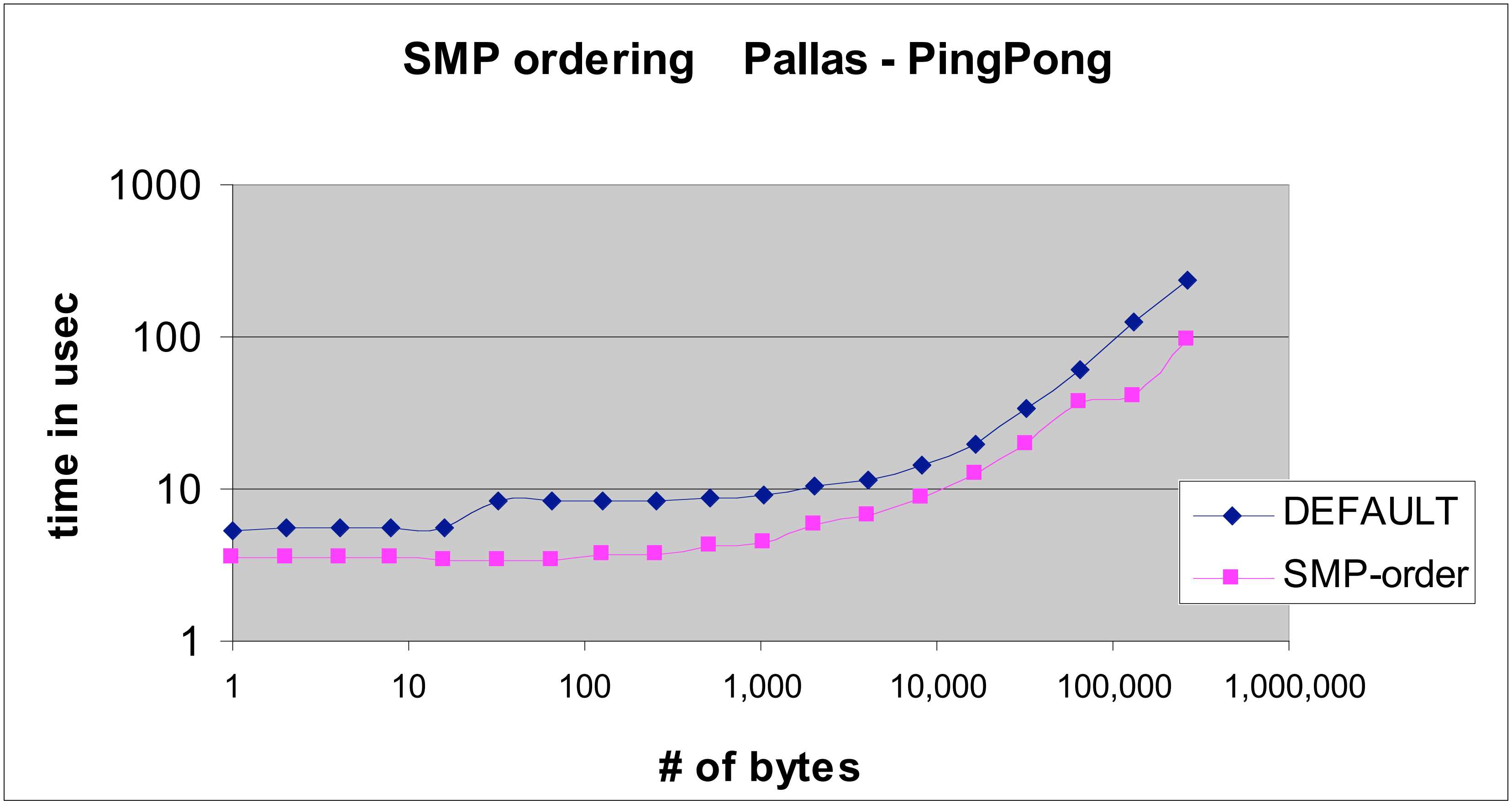
  - MPICH_RANK_FILE_BACKOFF

    Specifies the number of milliseconds for backoff.
  - MPICH_RANK_FILE_GROUPSIZE

    Specifies the number of ranks in the group size.

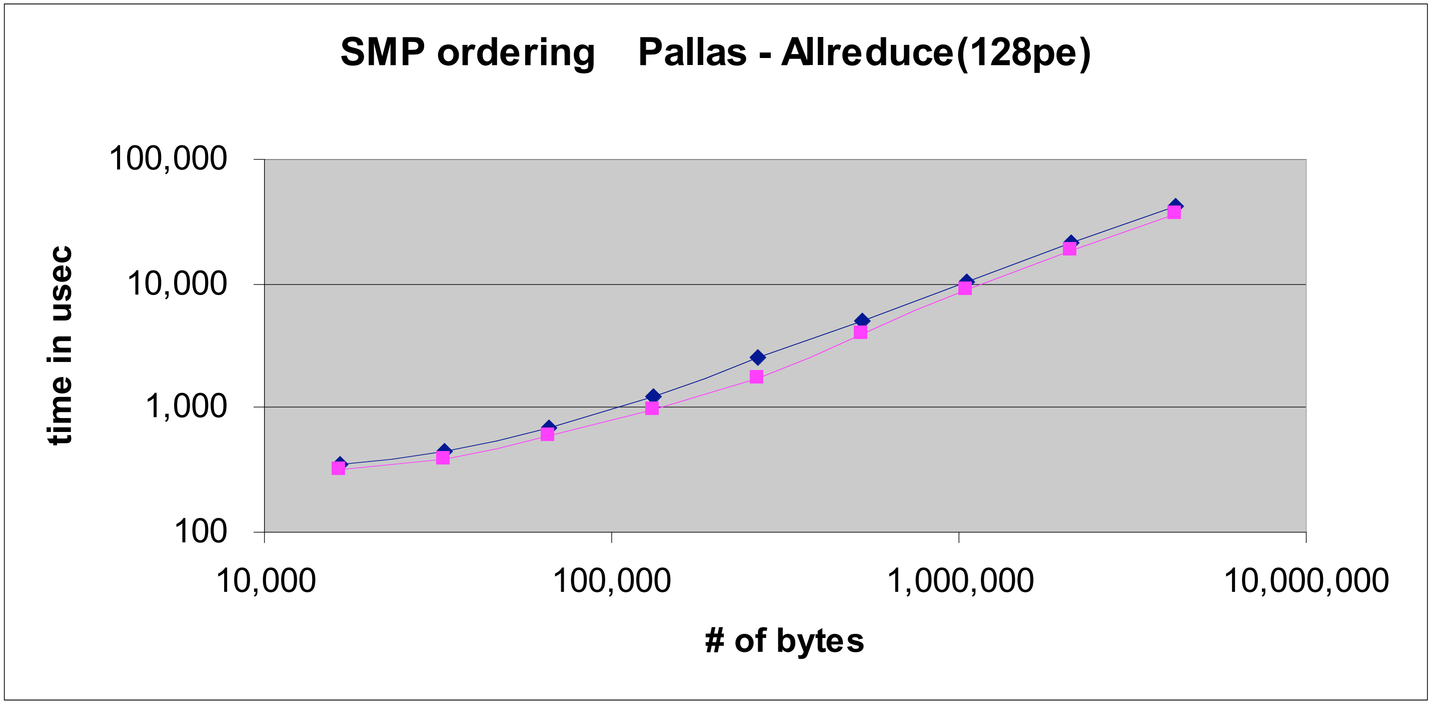
NOTE: Setting PMI_DEBUG will display rank information to stdout

SMP Rank placement speedups (MPICH_RANK_REORDER_METHOD=1)



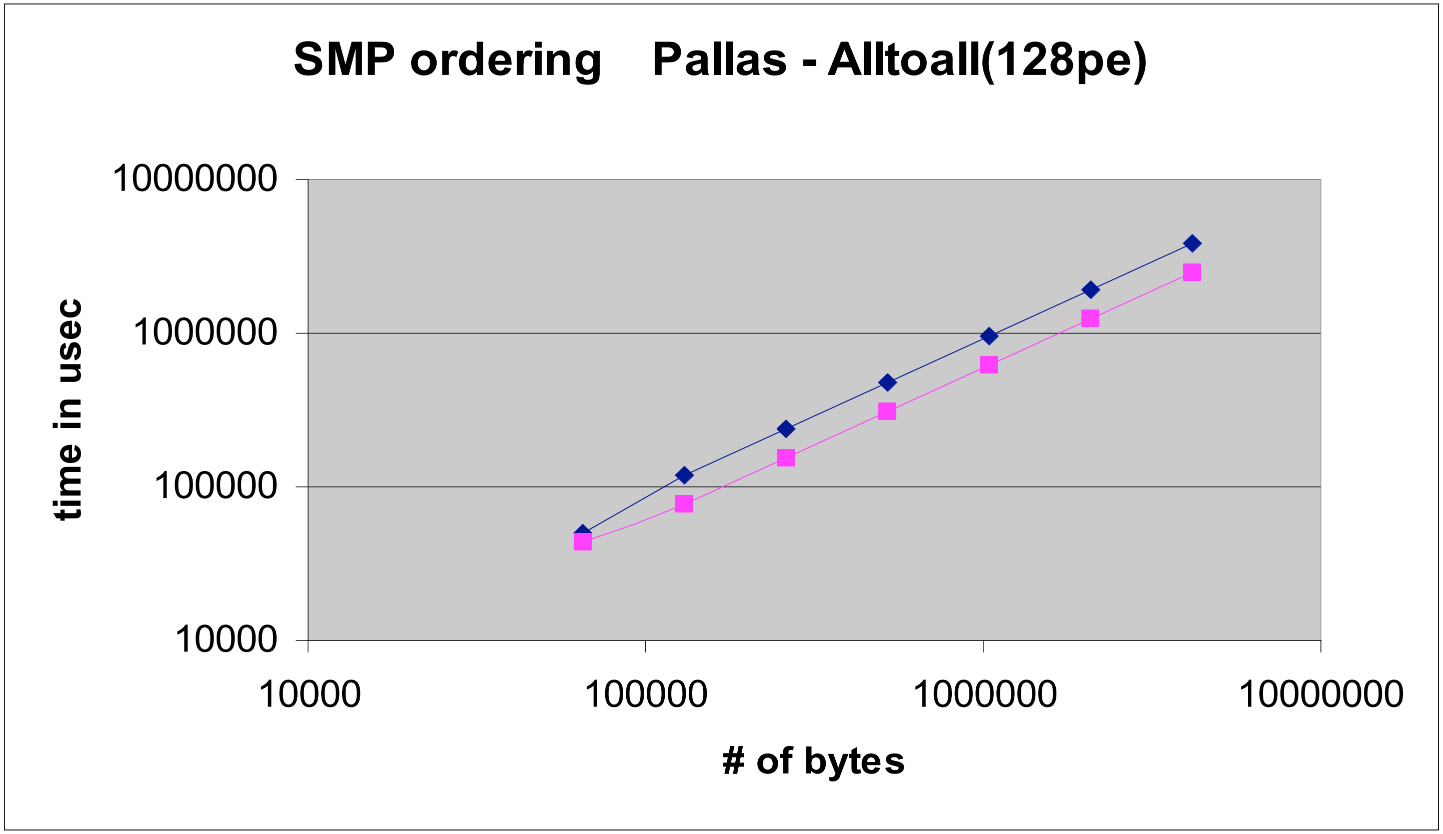
pt2pt faster by 35% at 8 byte to 60% at 256K bytes

SMP Rank placement speedups (MPICH_RANK_REORDER_METHOD=1)



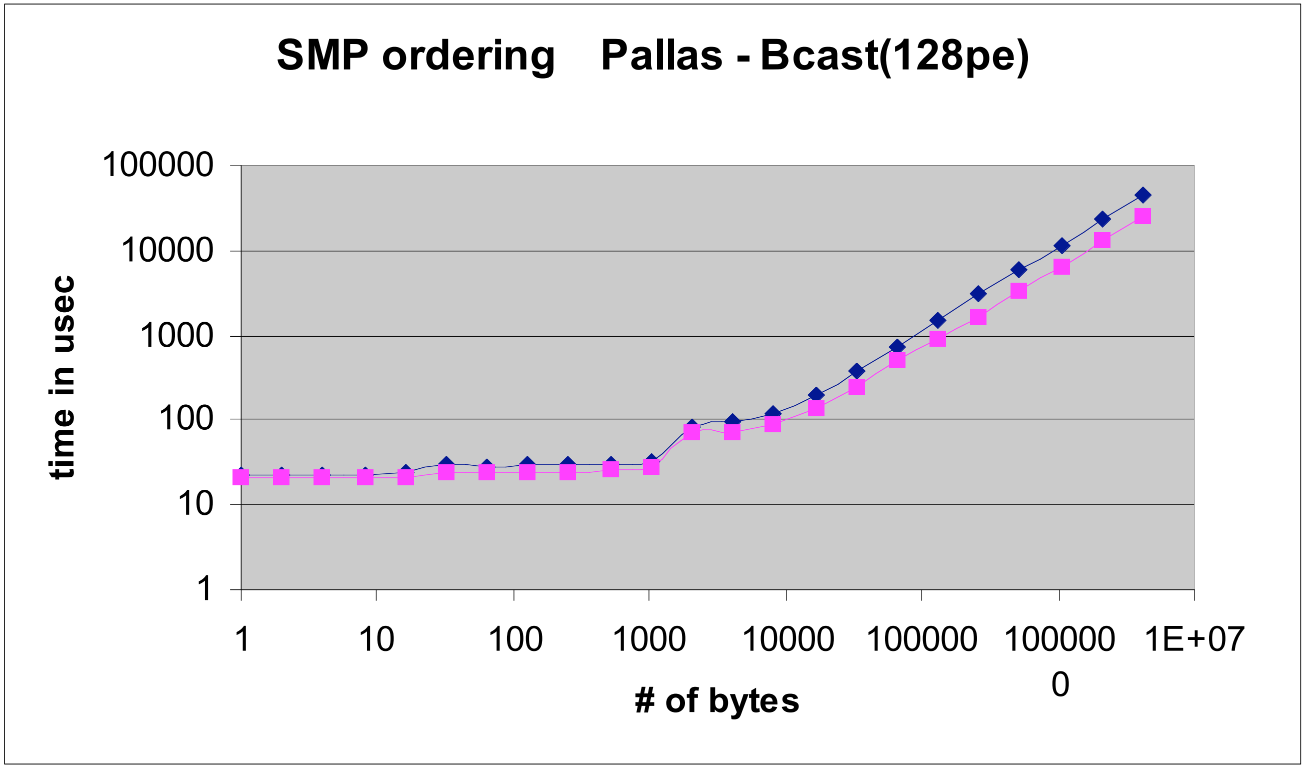
Allreduce faster by 7% to 32% above 16K bytes

SMP Rank placement speedups (MPICH_RANK_REORDER_METHOD=1)



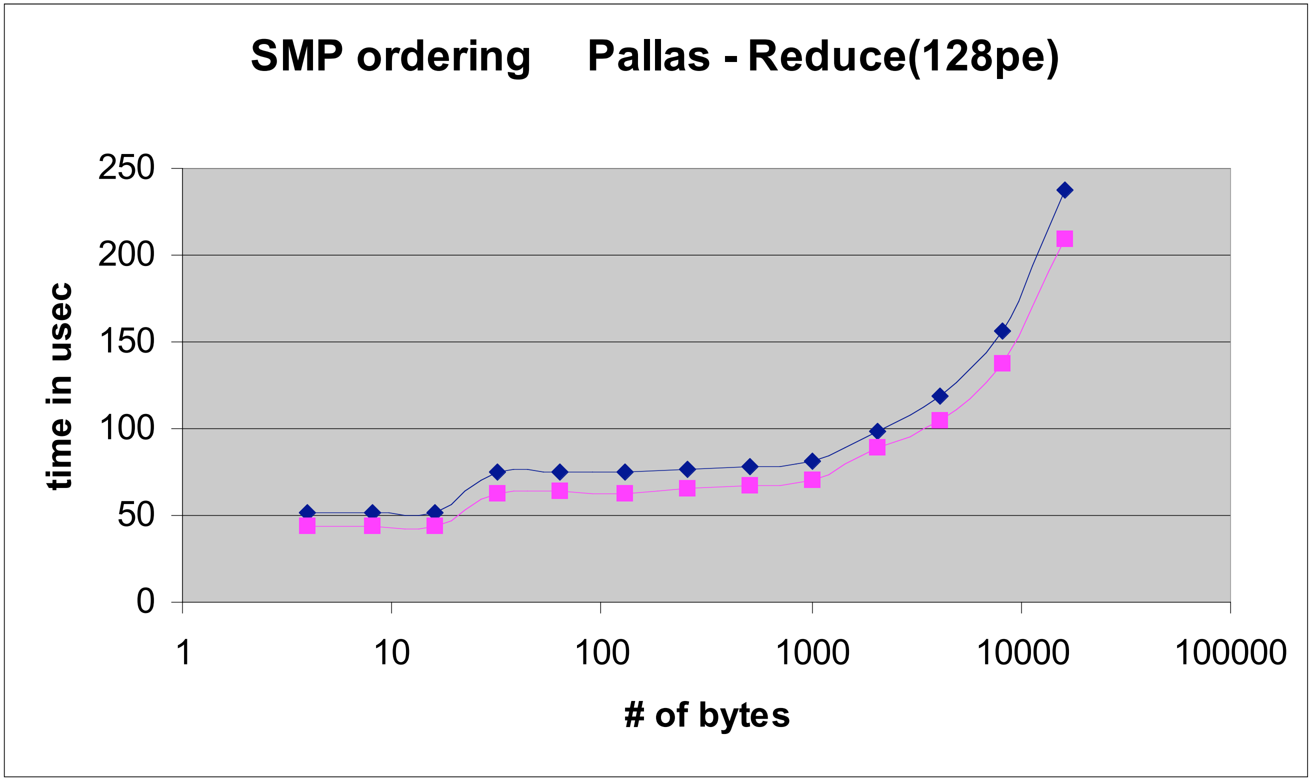
Alltoall faster by 15% to 36% above 65K message size

SMP Rank placement speedups (MPICH_RANK_REORDER_METHOD=1)



Bcast faster by 12% at 8 bytes to 45% at 1M bytes

SMP Rank placement speedups (MPICH_RANK_REORDER_METHOD=1)

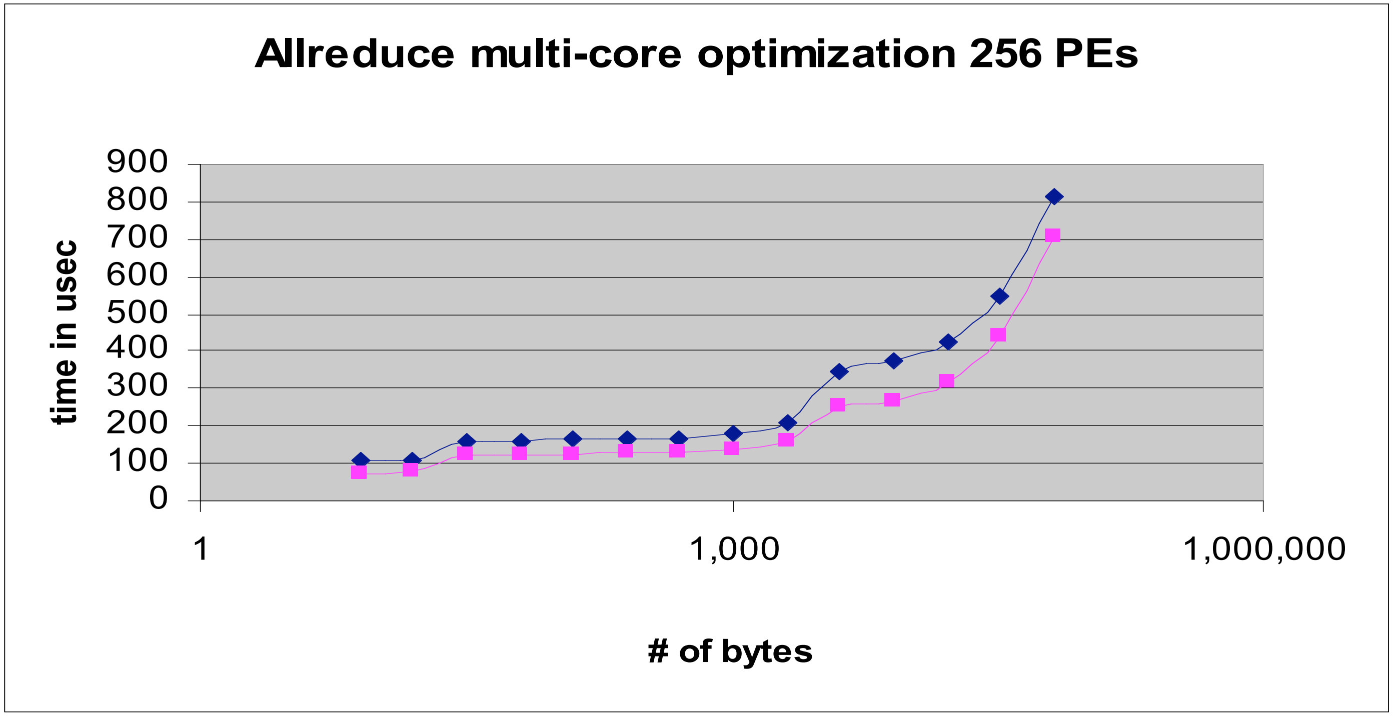


Reduce faster by 16% at 8 bytes to 12% at 16K bytes

MPI_COLL_OPT_ON

- MPI_COLL_OPT_ON multi-node collective optimizations (1.5.11 and 1.4.32)
  - MPI_Allreduce 30% faster for 16K bytes or less (Pallas 256 PEs)
  - MPI_Barrier - 25% faster (Pallas 256 PEs)

Multi-core optimization speedup (MPI_COLL_OPT_ON)



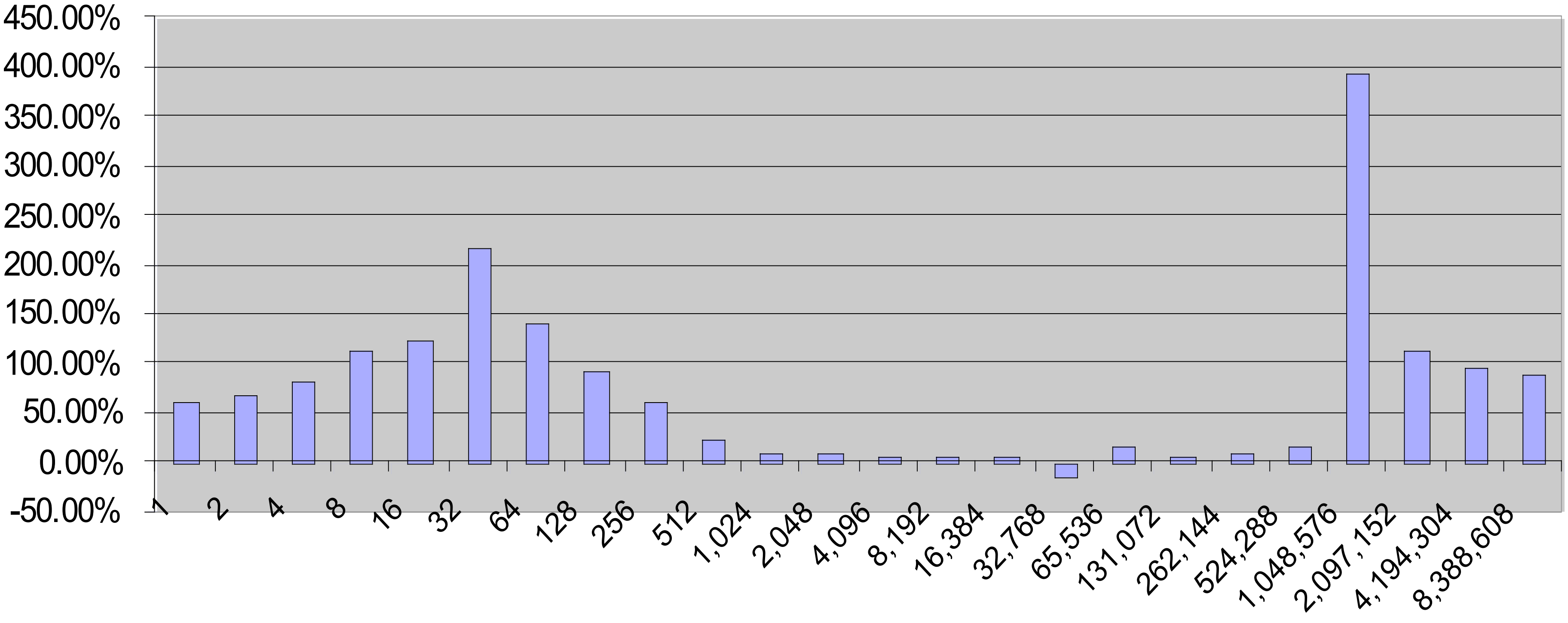
Allreduce faster by 4% at 1M bytes to 42% at 8 bytes

MPICH_FAST_MEMCPY

- New improved memcpy used within MPI for local copies for pt2pt and collectives.

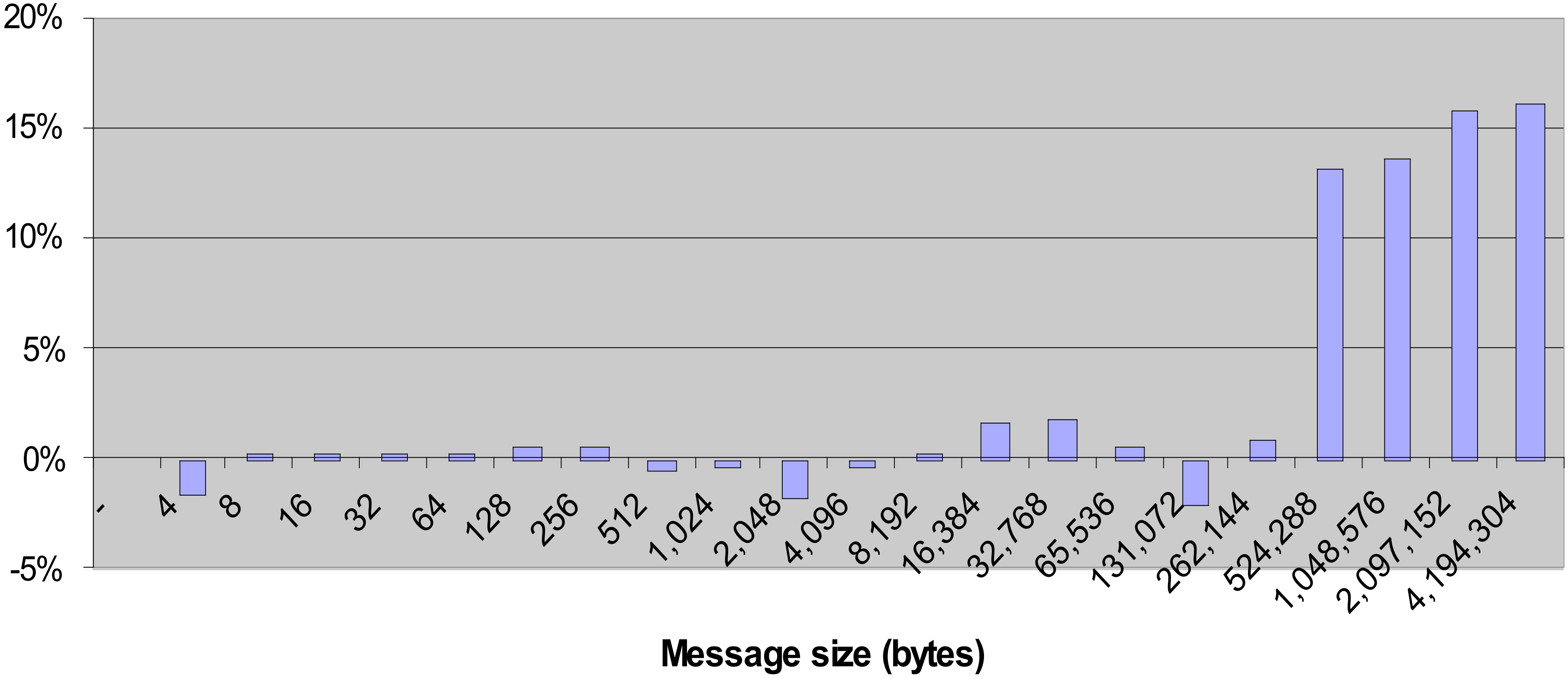
- Many collectives 8-20% faster above 256K bytes

MPICH_FAST_MEMCPY - Raw Memcpy Comparison
Percent Improvement using Optimized Memcpy over Default Memcpy



Transfer Size (bytes)

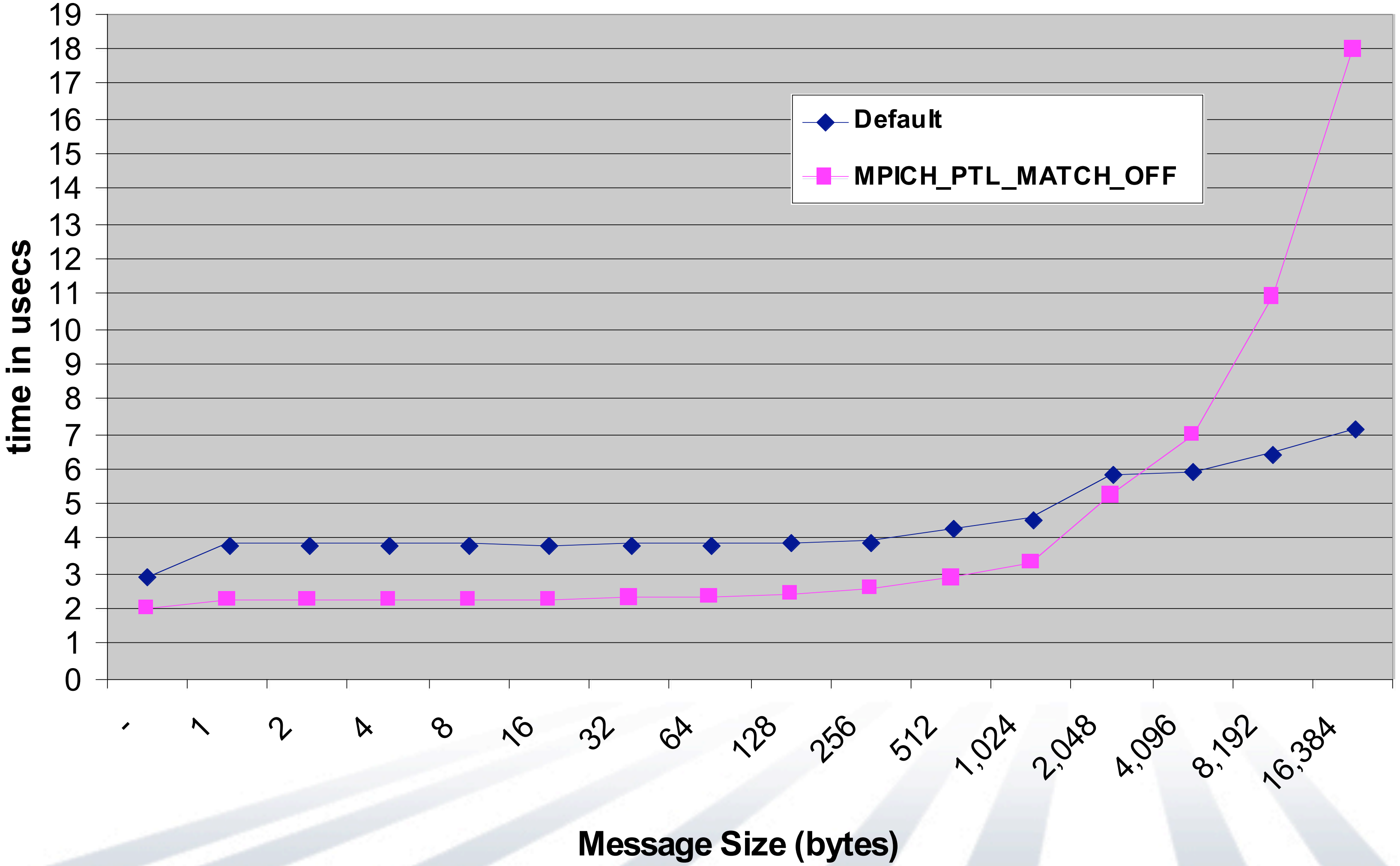
MPICH_FAST_MEMCPY - Allreduce 128pe
Percent Improvement using Optimized Memcpy over Default Memcpy



MPICH_PTL_MATCH_OFF

- Code was originally developed to allow apps to run which had exceeded portals resources by doing receive matching in MPI instead of portals
- Discovered that it helped ping-pong latency as much as 40% on node
- Option available in 1.5.39 and 1.4.50
- Many collectives 5-20% faster below 4K bytes
- Can be worse above 4K message size
- Looking at ways for portals and MPI to work together to get best performance across all sizes

MPI Latency Comparison (on-node) using MPICH_PTL_MATCH_OFF



XT Cray SHMEM Improvements

- XT3 Cray SHMEM perf improvements (1.5.9, 1.4.30)
  - SHMEM reduction improvements (40% - 4X faster)
  - SHMEM broadcast improvements (50% - 5X faster)

Latest Cray XT MPT Functional Improvements

- MPICH_PTL_SEND_CREDITS env variable for apps(like NAMD) that run out of unexpected event entries
- MPI and SHMEM “-default64” option (1.5)
- Improved intro_mpi man page
- Support for Pathscale compilers (1.5.38)
- Support for GNU 4.1.1 (includes GNU fortran90) (1.5.39)

Future Cray XT MPT Perf Improvements

- Eval of Sandia portals collective library for MPI
- MPI-IO optimizations
  - Enable MPI-IO collective I/O optimizations
  - Add IOBUF support to MPI-IO
  - Investigate other MPI-IO optimizations
- Enable more optimizations by default
- Use fast memcpy for other cases in MPI, etc.
  - derived data types
  - MPI-IO
  - portals short-circuit
  - user memcpy
  - other kernel memcpy

Future Cray XT MPT Functional Improvements

- MPI and SHMEM async build and release
- Gather and dump various MPI stats
- Improve MPI error messages with more recommendations
- Provide fully functional and optimized MPI and SHMEM for Compute Node Linux (CNL)
  - CNL launcher (aprun) uses SMP placement as default
  - Multi-device MPI-portals(across nodes) and SMP(on node)

More Info

- Man pages
  - intro_mpi
  - intro_shmem
  - yod
  - aprun (CNL)
- Cray XT Programming Environment User's Guide
- MPI Standard documentation

  (http://www.mpi-forum.org/docs/docs.html)
- MPICH2 implementation information

  (http://www-unix.mcs.anl.gov/mpi/mpich2)