Very Early Evaluation of the Cray X1 at Oak Ridge National Laboratory

Patrick H. Worley
Thomas H. Dunigan, Jr.

Oak Ridge National Laboratory

45th Cray User Group Conference
May 13, 2003
Hyatt on Capital Square
Columbus, OH

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

UT-BATTELLE

Acknowledgements

- Research sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

- Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the United States Department of Energy under Contract No. DE-AC05-00OR22725.

Phoenix

Cray X1 with 8 SMP nodes

- 4 Multi-Streaming Processors (MSP) per node
- 4 Single Streaming Processors (SSP) per MSP
- Two 32-stage 64-bit wide vector units running at 800 MHz and one 2-way superscalar unit running at 400 MHz per SSP
- 2 MB Ecache per MSP
- 16 GB of memory per node

for a total of 32 processors, 128 GB of memory , and 400 GF/s peak performance. System will be upgraded to 64 nodes (256 MSPs) by the end of September, 2003.

Talk and Paper Topics

- What performance you might expect
- Performance quirks and bottlenecks
- Performance optimization tips

Performance data repositories at

http://www.csm.ornl.gov/~dunigan/cray

and

http://www.csm.ornl.gov/evaluation

Current Evaluation Goals

- Verifying advertised functionality and performance
- Quantifying performance impact of
  - Scalar vs. Vector vs. Streams
  - Contention for memory within SMP node
  - SSP vs. MSP mode of running codes
  - Page size
  - MPI communication protocols
  - Alternatives to MPI: SHMEM and Co-Array Fortran
  - …

Caveats

- These are VERY EARLY results, resulting from approx. one month of benchmarking.
- As more codes are ported and optimized, and more performance data are analyzed, the understanding will improve.
- Performance characteristics are still changing, due to continued evolution of OS and compilers and libraries.
  - This is a good thing - performance continues to improve.
  - This is a problem - advice on how to optimize performance continues to change, and performance data has a very short lifespan.

Take nothing for granted. Check back (with us and others) for latest evaluation results periodically. Verify everything with your own codes. Share what you find out.

Outline

- Standard or External Benchmarks (unmodified)
  - Single MSP performance
  - Memory subsystem performance
  - Interprocessor communication performance
- Custom Kernels
  - Performance comparison of compiler and runtime options
  - Single MSP and SSP performance
  - SMP node memory performance
  - Interprocessor communication performance
- Application Codes
  - Performance comparison of compiler and runtime options
  - Scaling performance for fixed size problem

Other Platforms

- Earth Simulator: 640 8-way vector SMP nodes and a 640x640 single-stage crossbar interconnect. Each processor has 8 64-bit floating point vector units running at 500 MHz.
- HP/Compaq AlphaServer SC at Pittsburgh Supercomputing Center: 750 ES45 4-way SMP nodes (1GHz Alpha EV68) and a Quadrics QsNet interconnect with two network adapters per node.
- Compaq AlphaServer SC at ORNL: 64 ES40 4-way SMP nodes (667MHz Alpha EV67) and a Quadrics QsNet interconnect with one network adapter per node.
- IBM p690 cluster at ORNL: 27 32-way p690 SMP nodes (1.3 GHz POWER4) and an SP Switch2 with two to eight network adapters per node.

Other Platforms (cont.)

- IBM SP at the National Energy Research Supercomputer Center (NERSC): 184 Nighthawk II 16-way SMP nodes (375MHz POWER3-II) and an SP Switch2 with two network adapters per node.
- IBM SP at ORNL: 176 Winterhawk II 4-way SMP nodes (375MHz POWER3-II) and an SP Switch with one network adapter per node.
- NEC SX-6 at the Arctic Region Supercomputing Center: 8-way SX-6 SMP node. Each processor has 8 64-bit floating point vector units running at 500 MHz.
- SGI Origin 3000 at Los Alamos National Laboratory: 512-way SMP node. Each processor is a 500 MHz MIPS R14000.

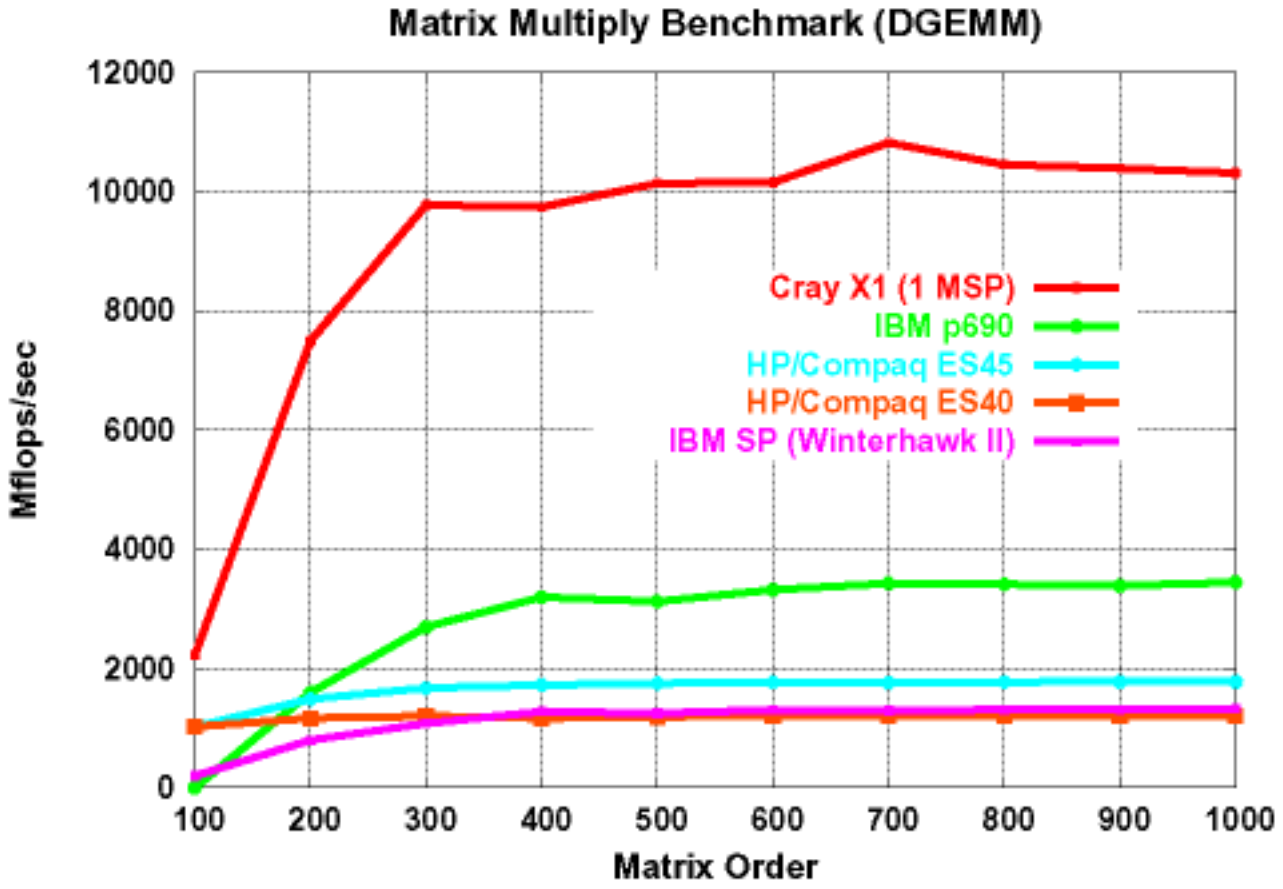
Standard Benchmarks

- Single MSP Performance
  - DGEMM matrix multiply benchmark
  - Euroben MOD2D dense eigenvalue benchmark
  - Euroben MOD2E sparse eigenvalue benchmark
- Memory Subsystem Performance
  - STREAMS triad benchmark
- Interprocessor Communication Performance
  - HALO benchmark
- Parallel Performance
  - NAS Parallel Benchmark code MG

DGEMM Benchmark

Comparing performance of vendor-supplied routines for matrix multiply. Cray X1 experiments used routines from the Cray scientific library libsci.

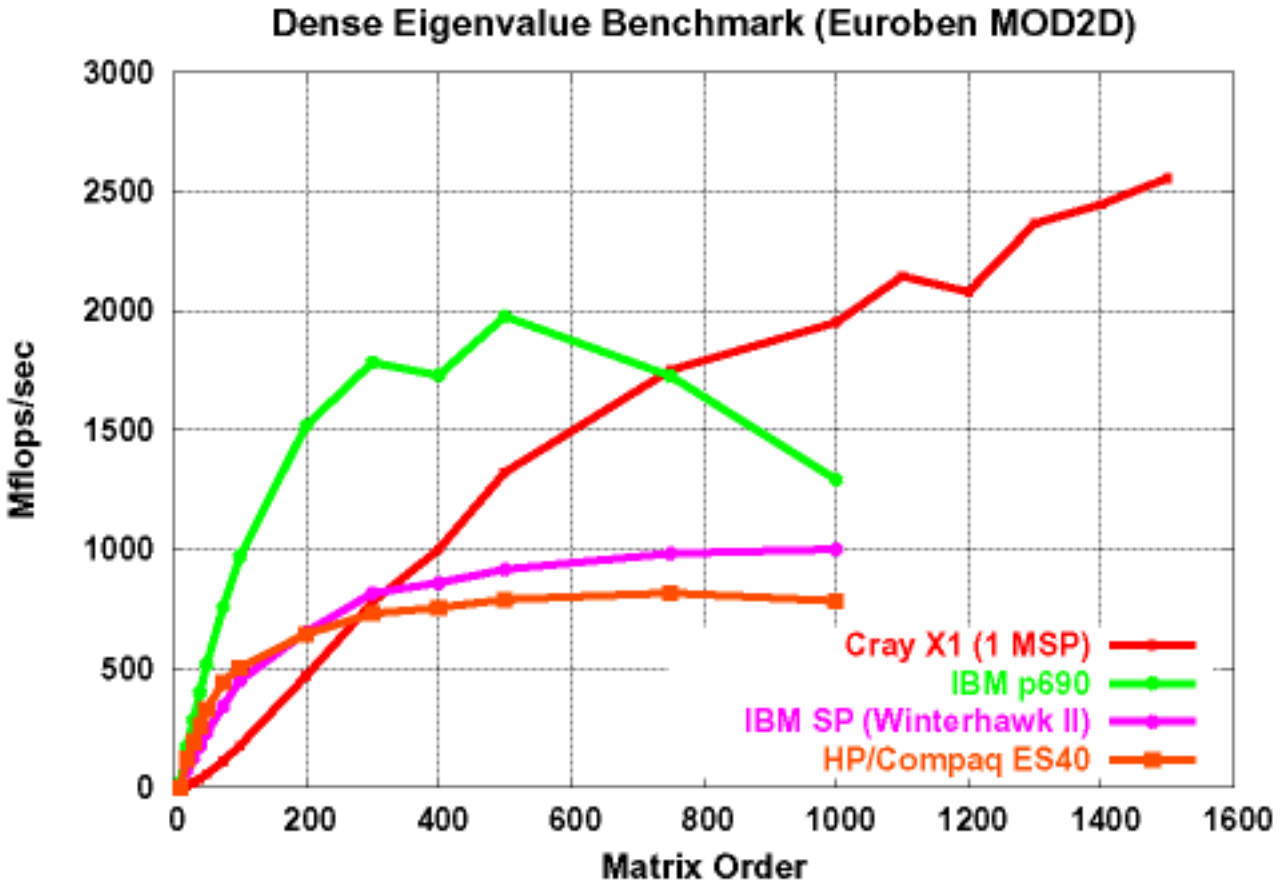
Good performance achieved, reaching 80% of peak relatively quickly.



MOD2D Benchmark

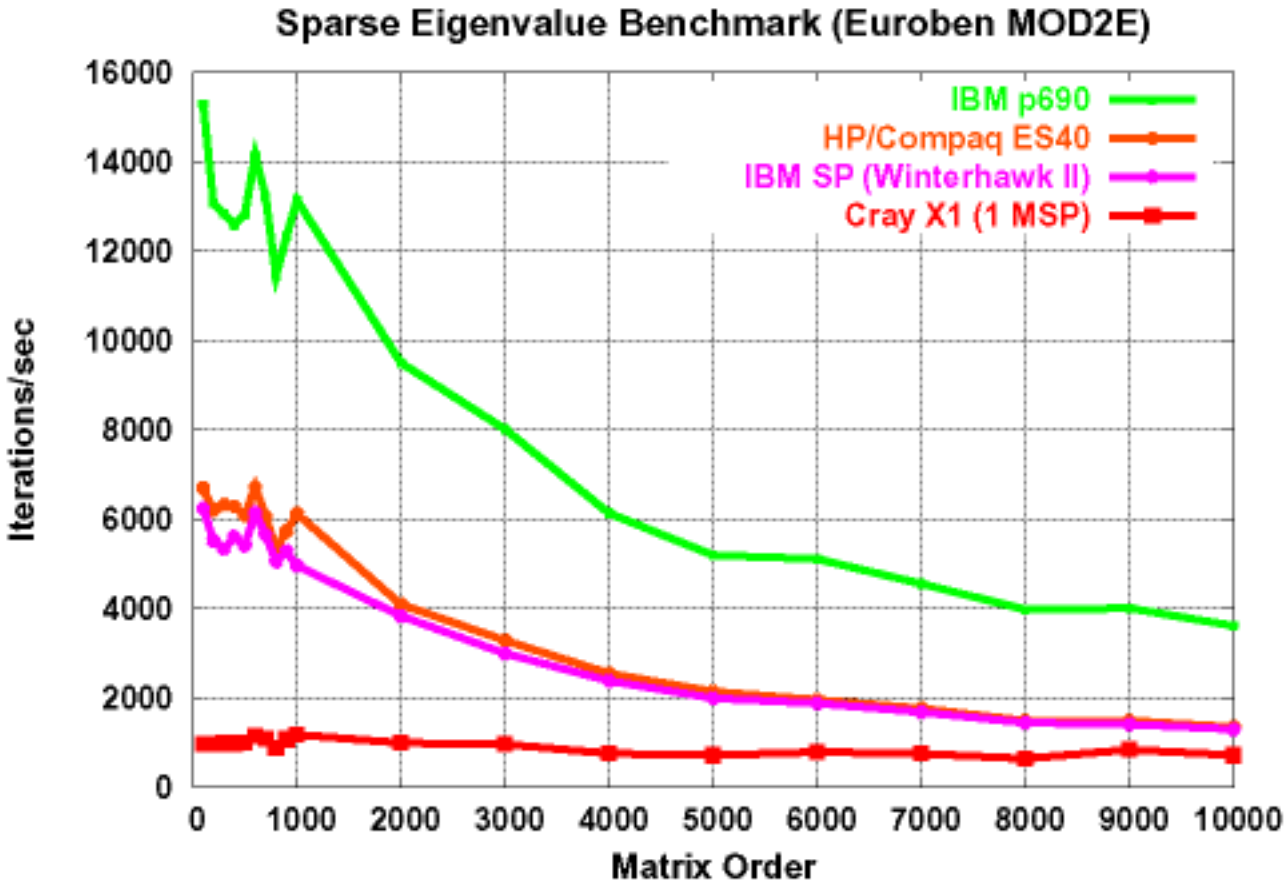
Comparing performance of vendor-supplied routines for dense eigenvalue analysis. Cray X1 experiments used routines from the Cray scientific library libsci.

Performance still growing with problem size. (Had to increase standard benchmark problem specifications.) Performance of nonvector systems has peaked.



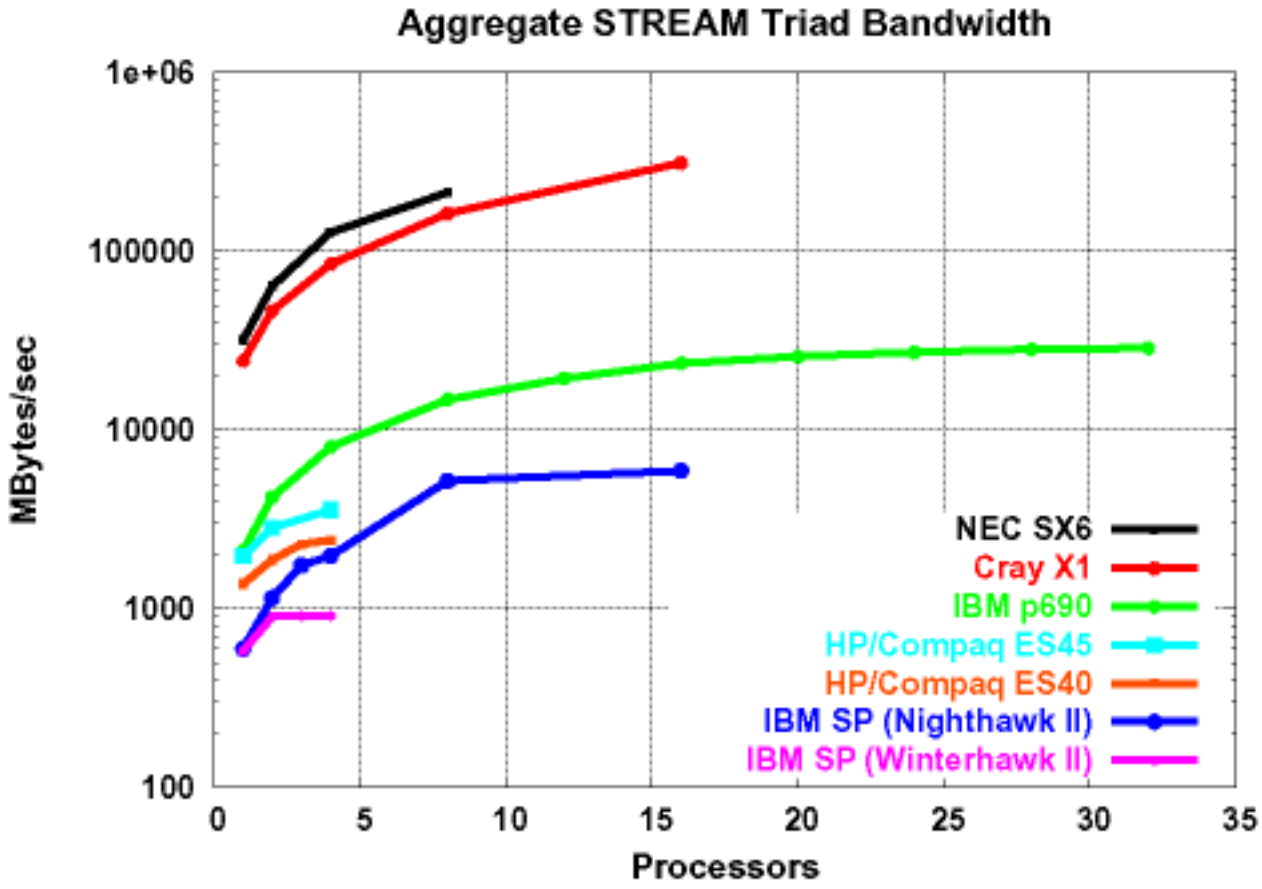
MOD2E Benchmark

Comparing performance of Fortran code for sparse eigenvalue analysis. Aggressive compiler options were used on the X1, but code was not restructured and compiler directives were not inserted. Performance is improving for larger problem sizes, so some streaming or vectorization is being exploited. Performance is poor compared to other systems.



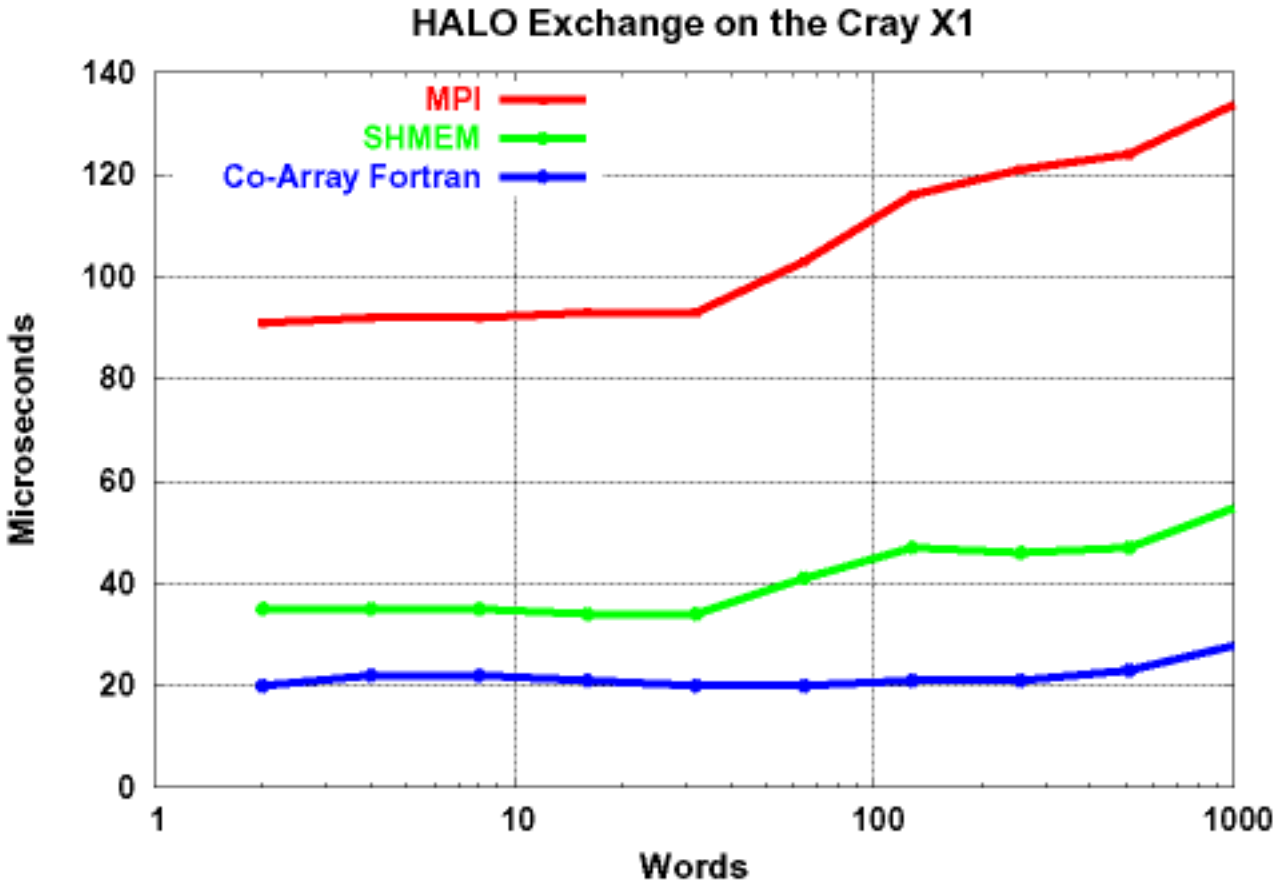
STREAMS Benchmark

Comparing aggregate bandwidth measured with the STREAMS triad benchmark. X1 performance much closer to that of the SX-6 than to the nonvector systems. For more than 4 processors, X1 data involves multiple SMP nodes. All other systems show performance within a single SMP node.



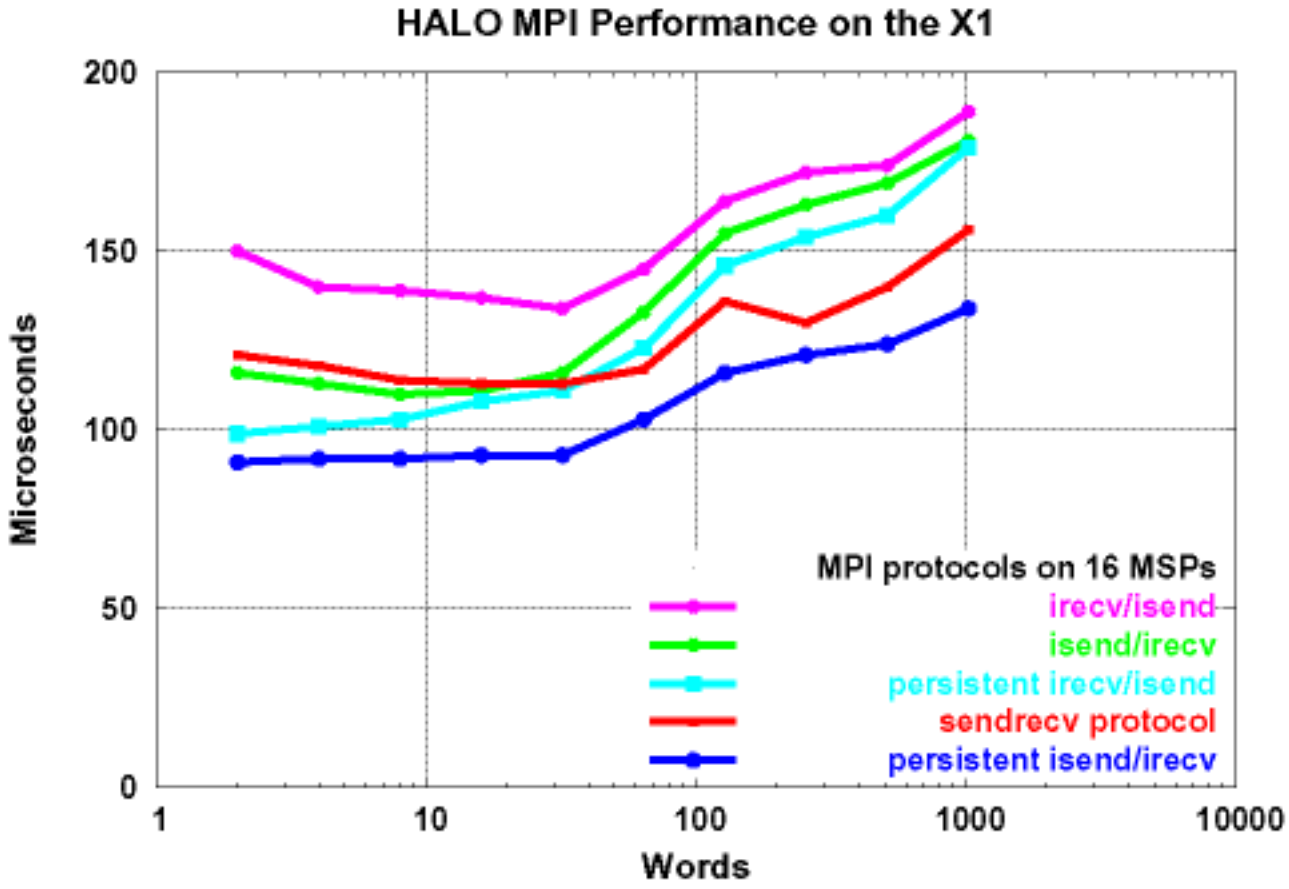
HALO Paradigm Comparison

Comparing performance of MPI, SHMEM, and Co-Array Fortran implementation of HALO on 16 MSPs. SHMEM and Co-Array Fortran are substantial performance enhancers for this benchmark.



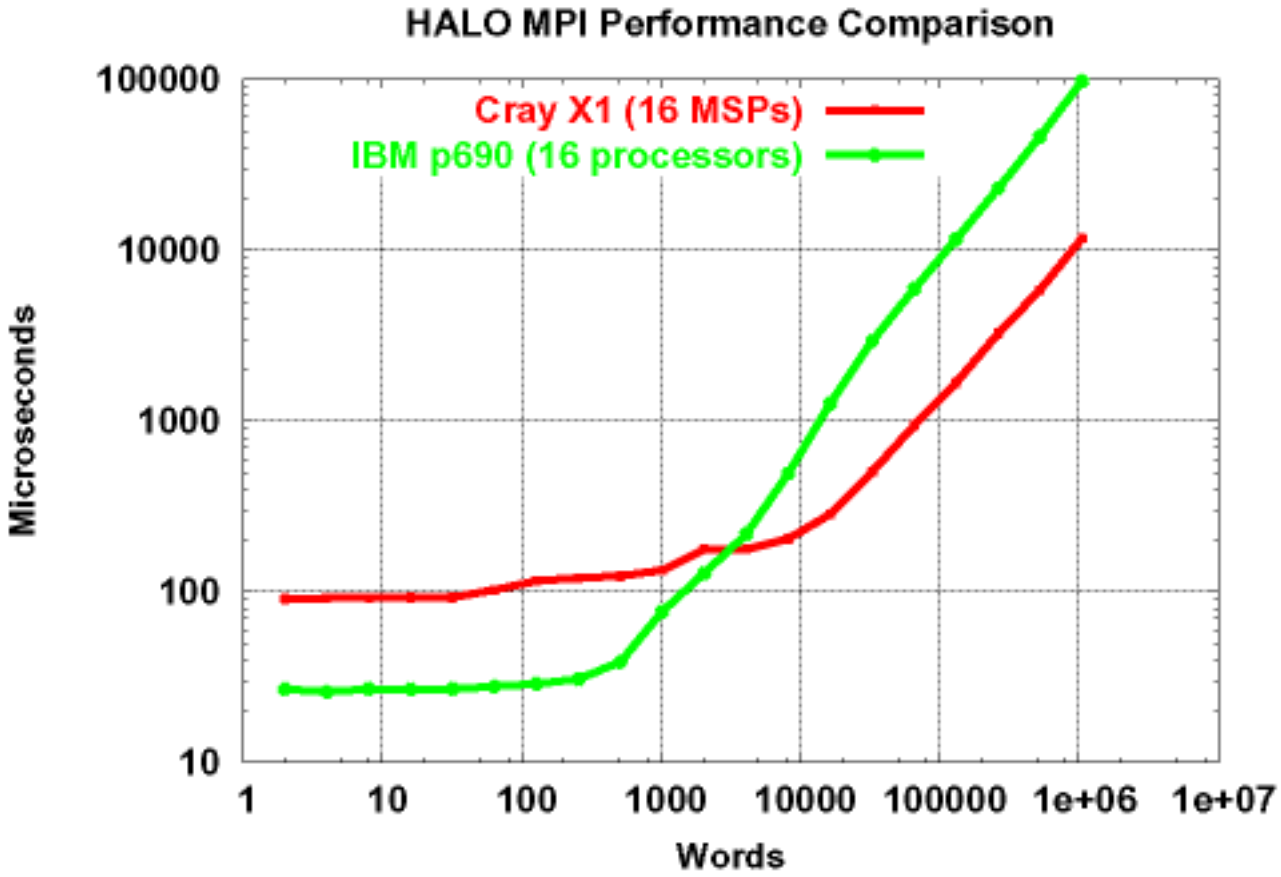
HALO MPI Protocol Comparison

Comparing performance of different MPI implementations of HALO on 16 MSPs. Persistent isend/irecv is always best. For codes that can not use persistent commands, MPI_SENDRECV is also a reasonable choice.



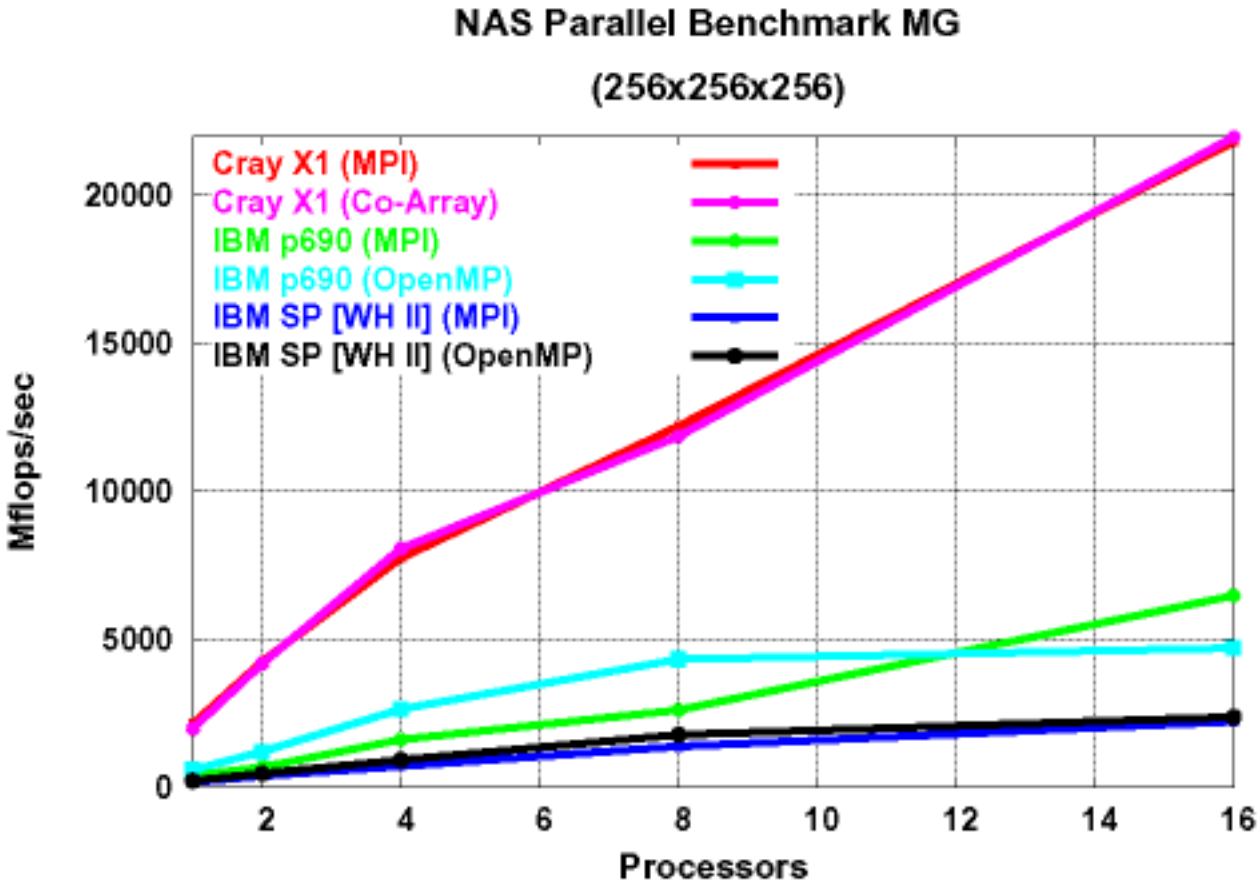
HALO Benchmark

Comparing HALO performance using MPI on 16 MSPs of the Cray X1 and 16 processors of the IBM p690 (within a 32 processor p690 SMP node). Achievable bandwidth is much higher on the X1. For small halos, the p690 MPI HALO performance is between the X1 SHMEM and Co-Array Fortran HALO performance.



MG Benchmark

Comparing performance of MG multigrid code. Aggressive compiler options were used on the X1, but code was not restructured and compiler directives were not inserted. Performance and scaling on the X1 are good compared to the IBM systems. MPI and Co-Array performance are nearly identical.



Custom Kernels

- PSTSWM
  - Impact of code, compiler, and runtime optimizations
  - MSP vs. SSP performance
  - Performance impact of memory contention
  - Benchmarking
- COMMTEST
  - Impact of distance and bandwidth contention on SWAP performance
  - MPI vs. SHMEM performance

PSTSWM Description

- The Parallel Spectral Transform Shallow Water Model represents an important computational kernel in spectral global atmospheric models. As 99% of the floating-point operations are multiply or add, it runs well on systems optimized for these operations. PSTSWM exhibits little reuse of operands as it sweeps through the field arrays; thus it exercises the memory subsystem as the problem size is scaled and can be used to evaluate the impact of memory contention in SMP nodes. PSTWM is also a parallel algorithm testbed, and all array sizes and loop bounds are determined at runtime. This makes it difficult for the X1 compiler to identify which loops to vectorize or stream.
- These experiments examine serial performance, both using one processor and running the serial benchmark on multiple processors simultaneously. Performance is measured for a range of horizontal problems resolutions for 18 vertical levels.

PSTSWM Experiment Particulars

Compiler Options

Default:

(nothing specified)

Aggressive:

-Oaggress,scalar3,vector3,stream3

Runtime Options

16MB pages (experimental default)

-p 16M:16M

64KB pages

-p 64K:64K

Horizontal Resolutions

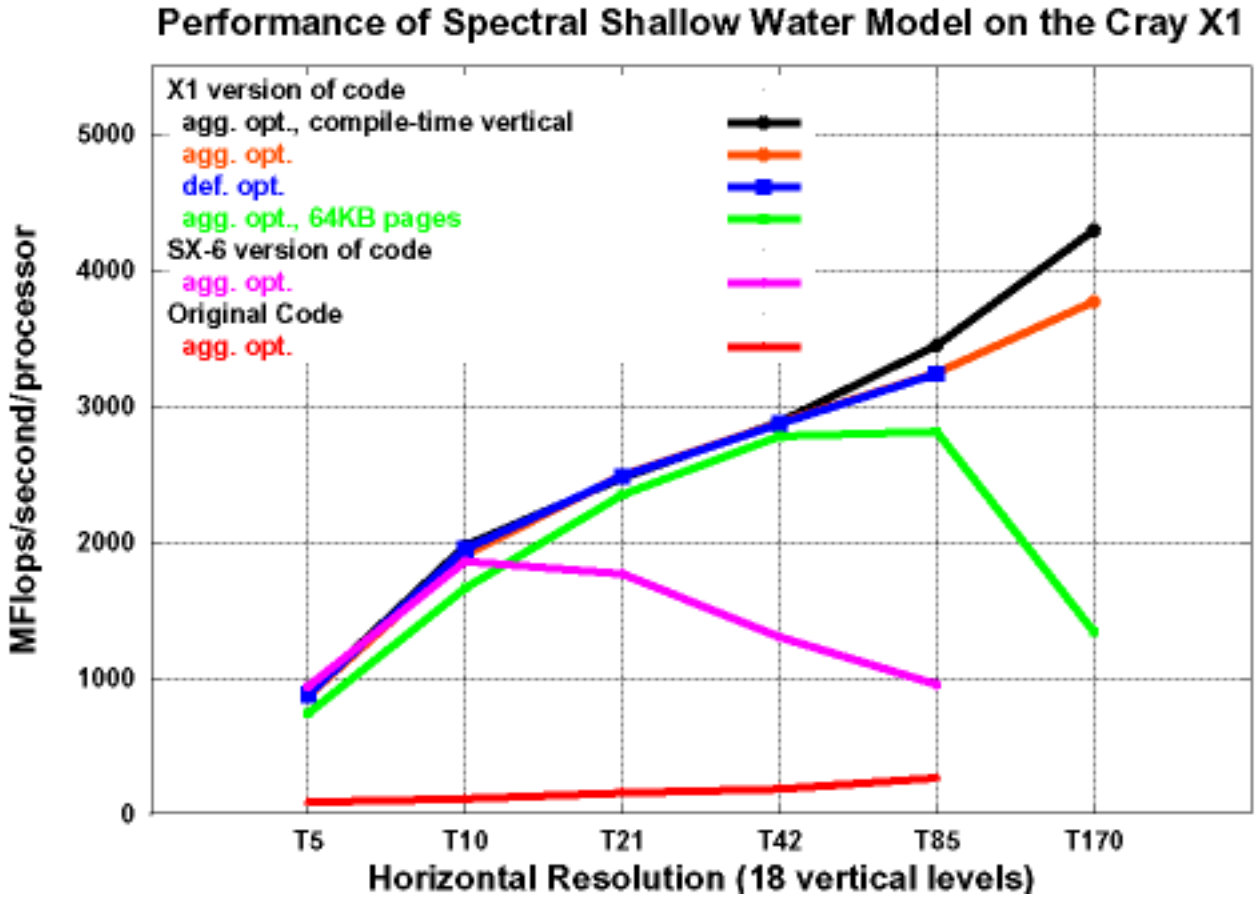
T5:	8x16
T10:	16x32
T21:	32x64
T42:	64x128
T85:	128x256
T170:	256x512

PSTSWM Code Versions

- Original (unvectorized) code
- Port to SX-6
  - changing loops and local arrays for select routines
- Port to X1
  - changing loops and local arrays for (same) select routines
- Port to X1 with compile-time specification of number of vertical levels

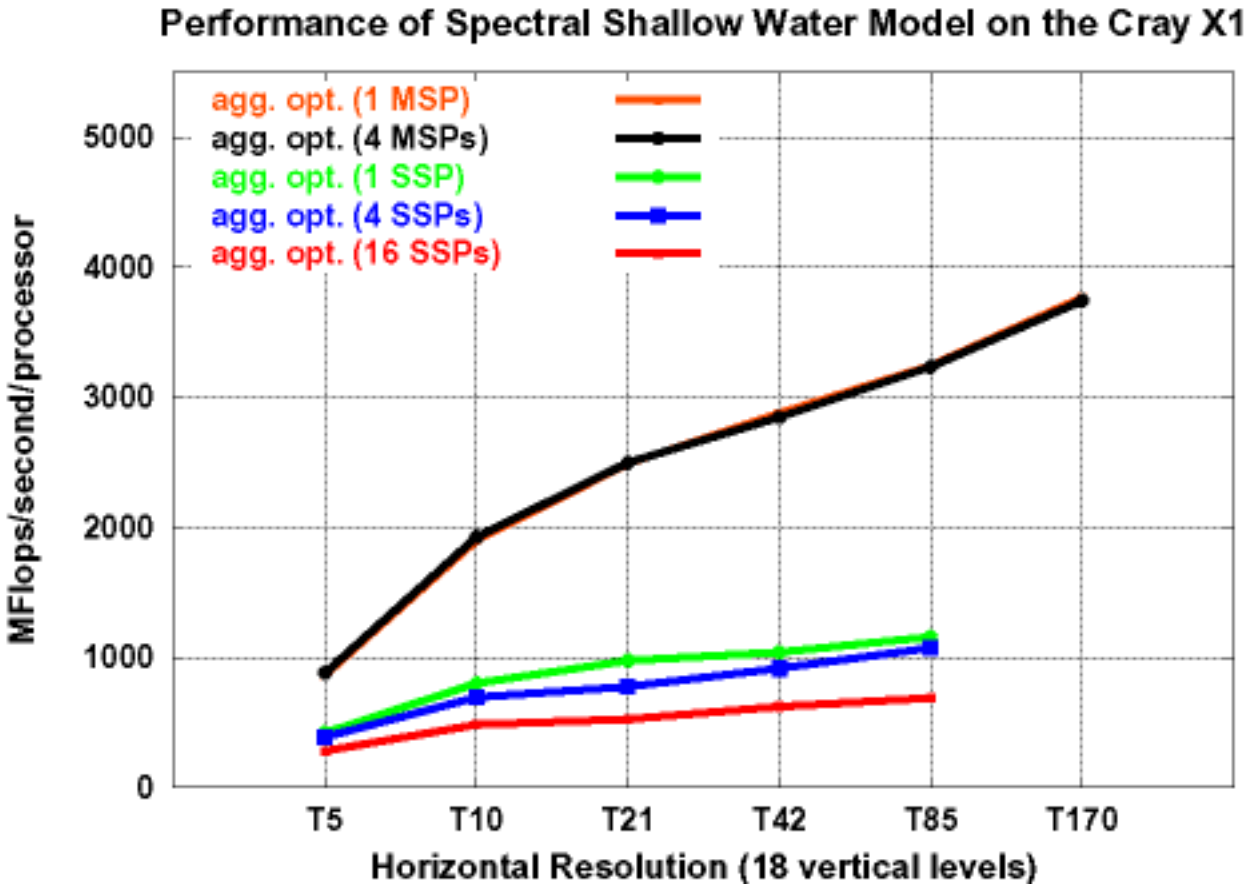
PSTSWM Compile and Runtime Comparisons

Comparing performance of different code versions, compiler optimizations, and (runtime) page size. Code modifications are crucial for this code, and the SX-6 modifications are not all appropriate. 16MB pages improve performance for large problem sizes, as does fixing vertical dimension at compile-time. Default compiler optimization was as good as more aggressive settings.



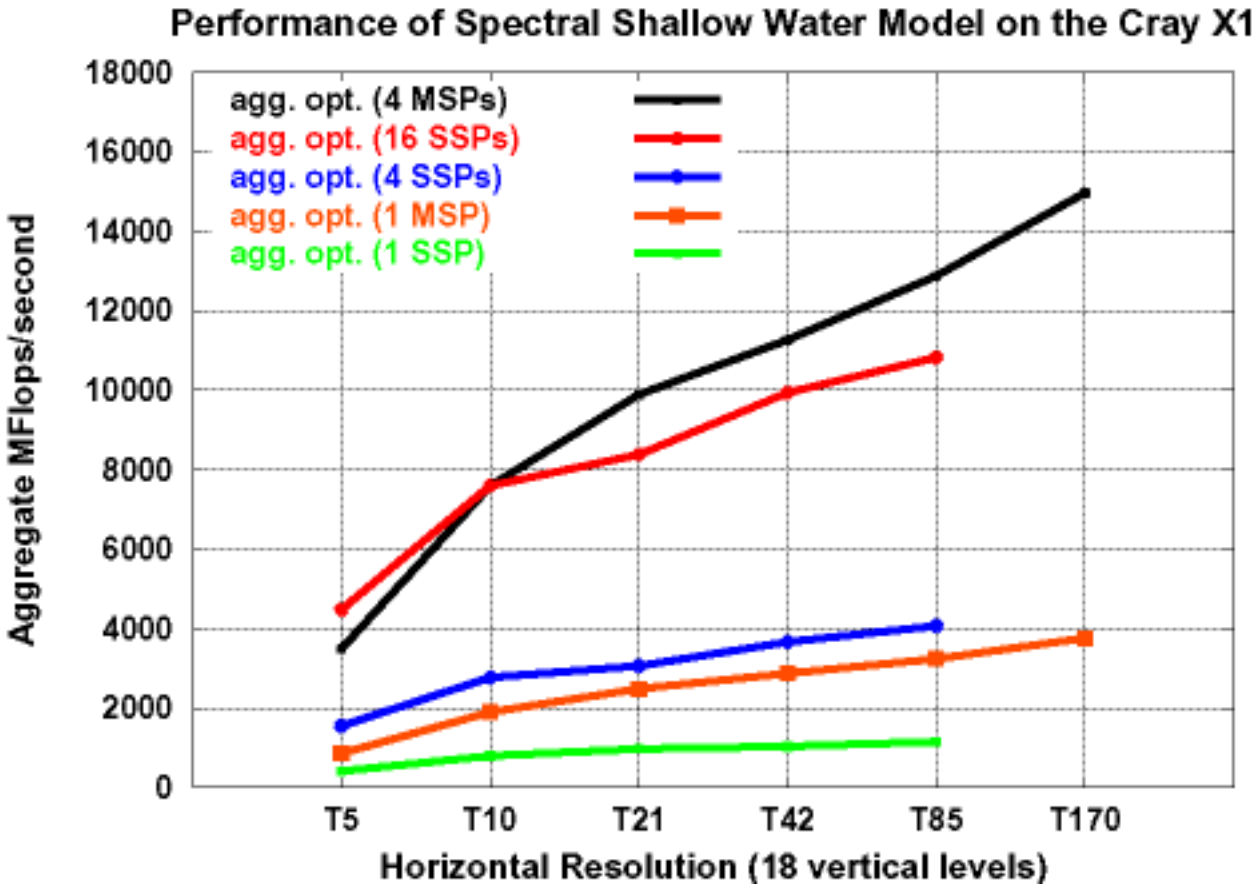
PSTSWM MSP vs. SSP Performance

Per processor performance when solving one instance of problem or same problem simultaneously on multiple processors, both SSP and MSP. One MSP and 4 MSP performance is identical. One, 4, and sixteen SSP performance shows interference, and performance degradation.



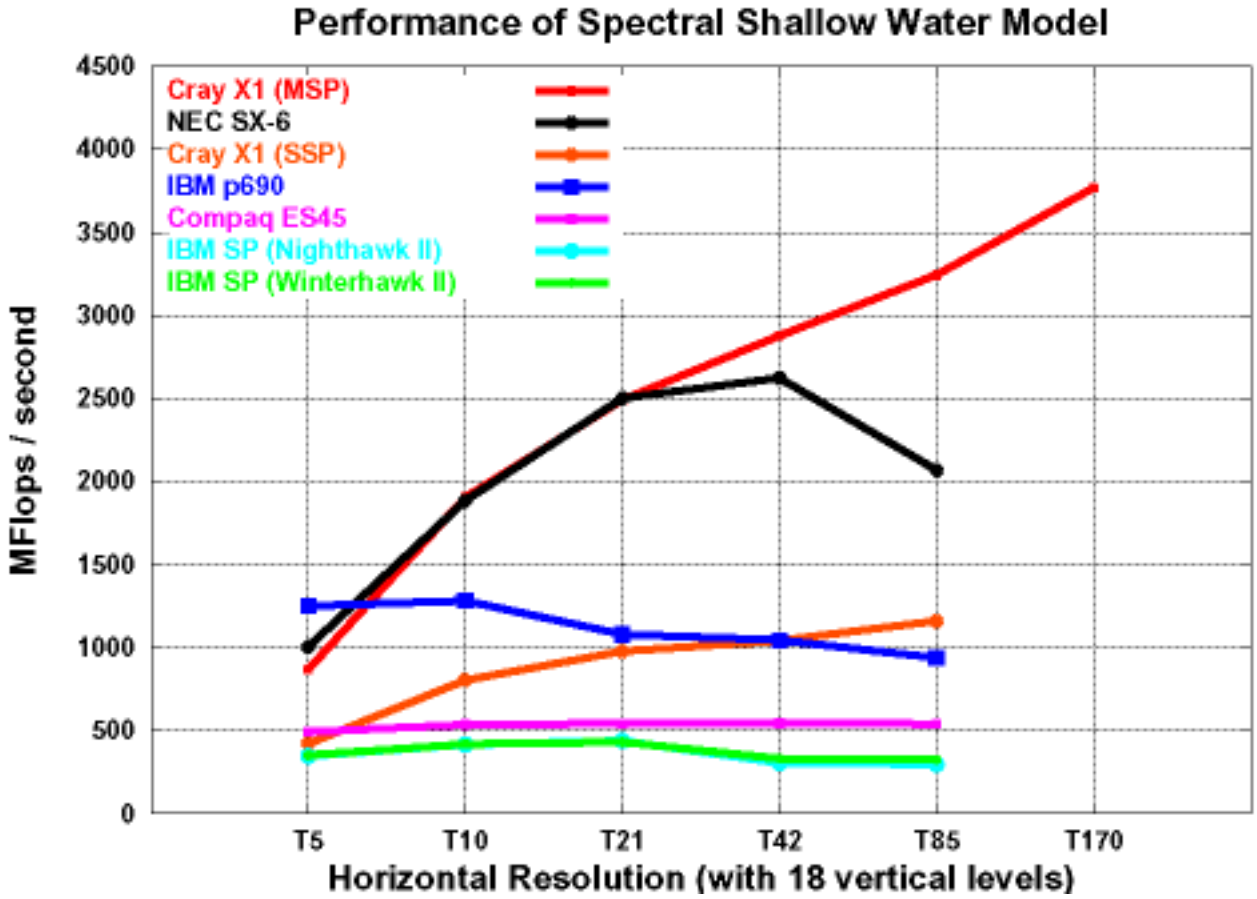
PSTSWM MSP vs. SSP Performance II

Aggregate performance when solving one instance of problem or same problem simultaneously on multiple processors. Four SSP performance is better than one MSP, but 4 MSP performance is better than 16 SSP performance.



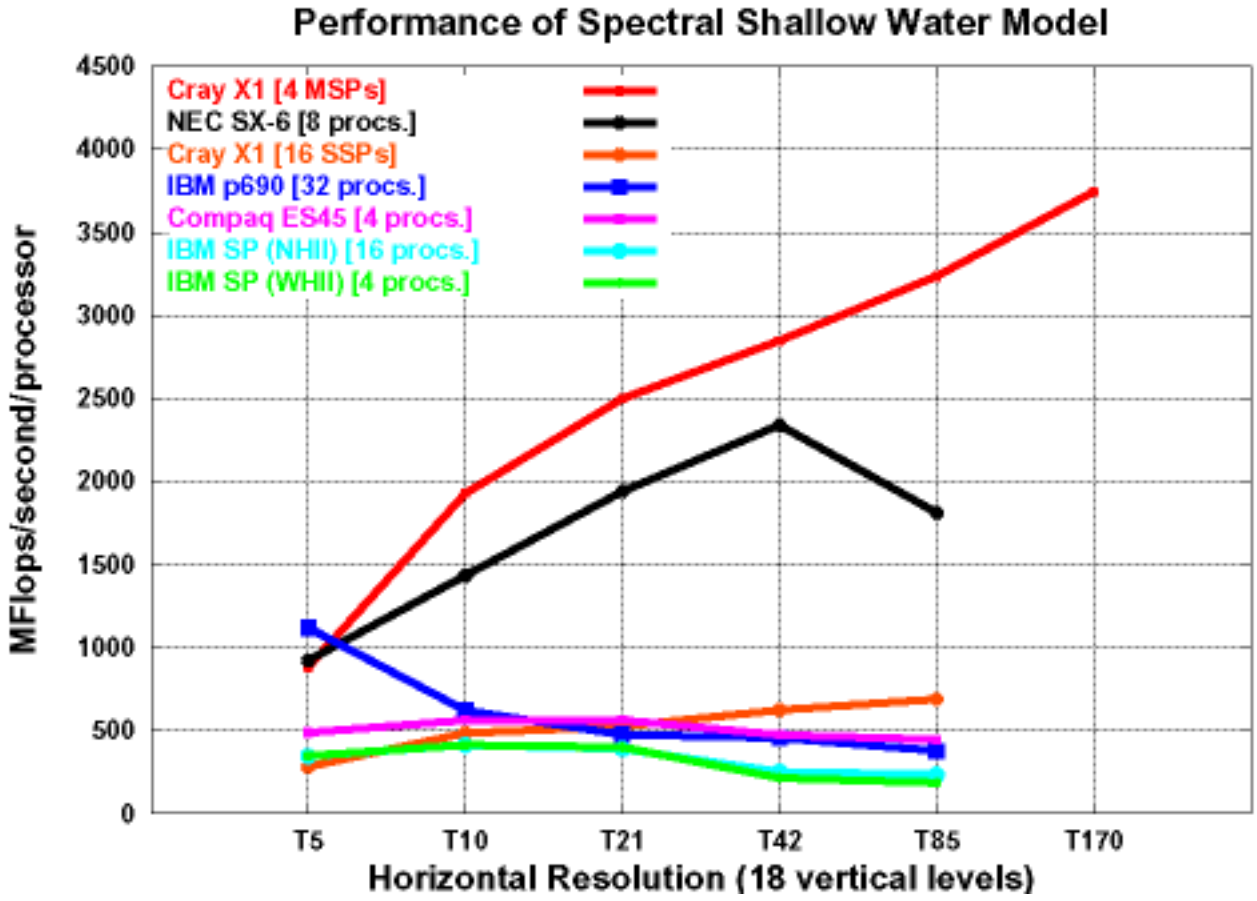
PSTSWM Processor Benchmark

Comparing single processor performance with PSTSWM, using runtime vertical specification on all systems. X1 MSP version performance scaling well with problem size, and even performance of SSP version exceeds p690 processor performance for the larger problem sizes.



PSTSWM SMP Node Benchmark

Comparing per processor performance when solving same problem simultaneously on all processors in SMP node. X1 MSP data are only data indicating no performance degradation when compared to single processor experiment. Appears to indicate additional advantage to X1 over other systems for scale-up type experiments (for this code).



COMMTEST Description

COMMTEST is a suite of codes that measure the performance of MPI interprocessor communication. In particular, COMMTEST evaluates the impact of communication protocol, packet size, and total message length in a number of “common usage” scenarios. It also includes simplified implementations of the SWAP and SENDRECV operators using SHMEM.

COMMTEST Experiment Particulars

0-1

MSP 0 swaps data with MSP 1 (within the same SMP node)

0-4

MSP 0 swaps data with MSP 4 (between two neighboring nodes)

0-8

MSP 0 swaps data with MSP 8 (between two more distant nodes)

i-(i+1)

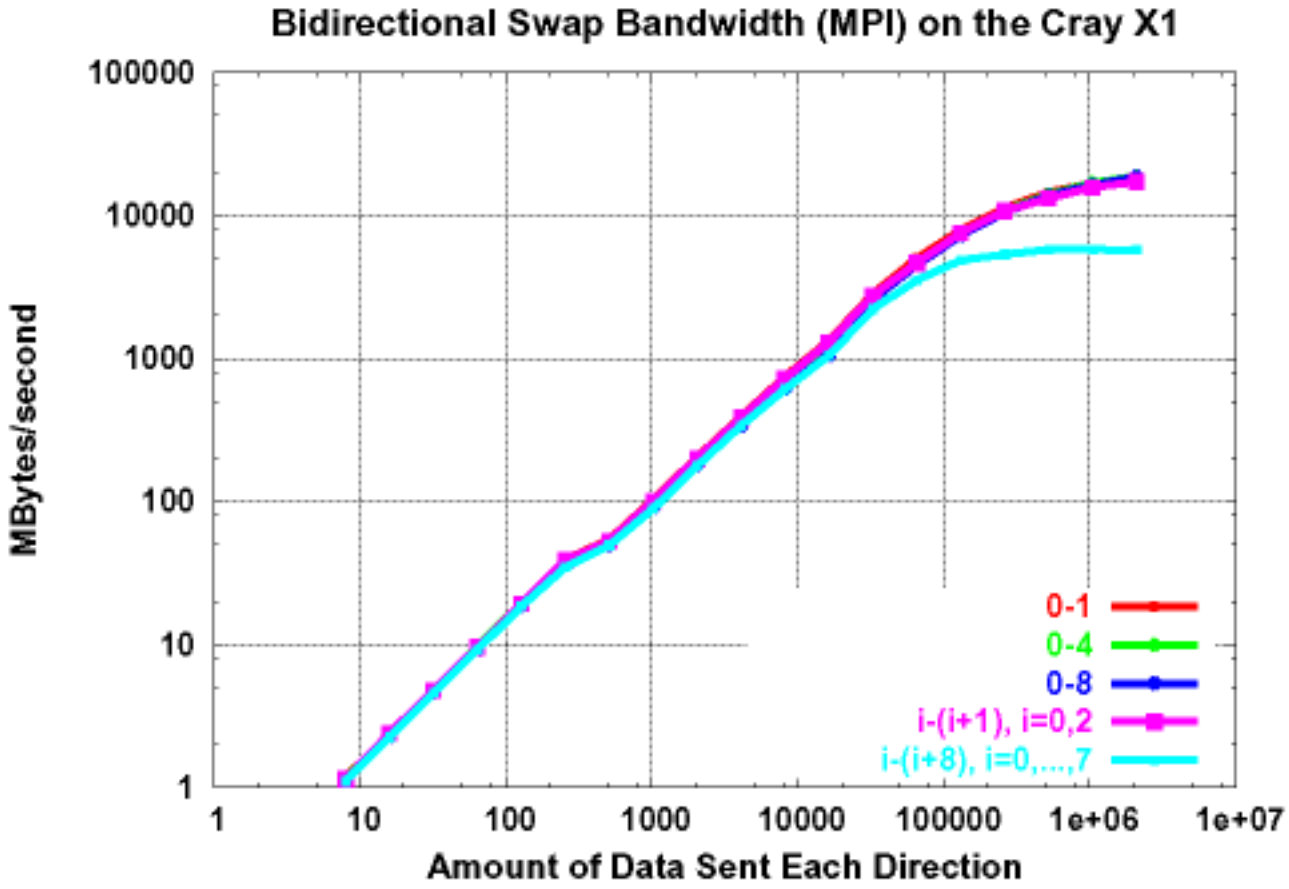
MSP 0 swaps with MSP 1 and MSP 2 swaps with MSP 3 simultaneously (within the same SMP node)

i-(i+8)

MSP i swaps with MSP (i+8) for i=0,…,7 simultaneously, i.e. 8 pairs of MSPs across 4 SMP nodes swap data simultaneously.

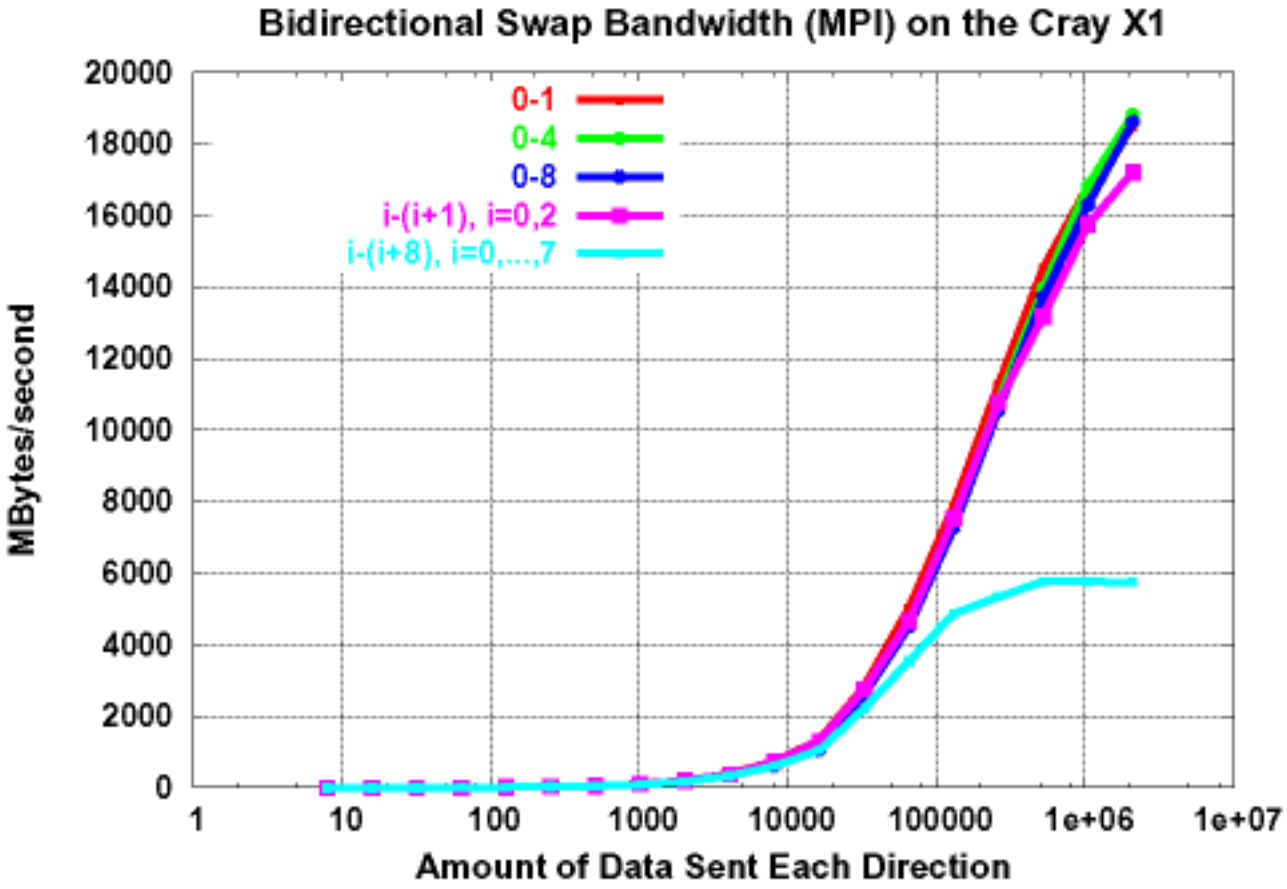
COMMTEST SWAP Benchmark

Comparing performance of SWAP for different communication patterns. All performance is identical except for the experiment in which 8 pairs of processors swap simultaneously. In this case, contention for internode bandwidth limits the single pair bandwidth.



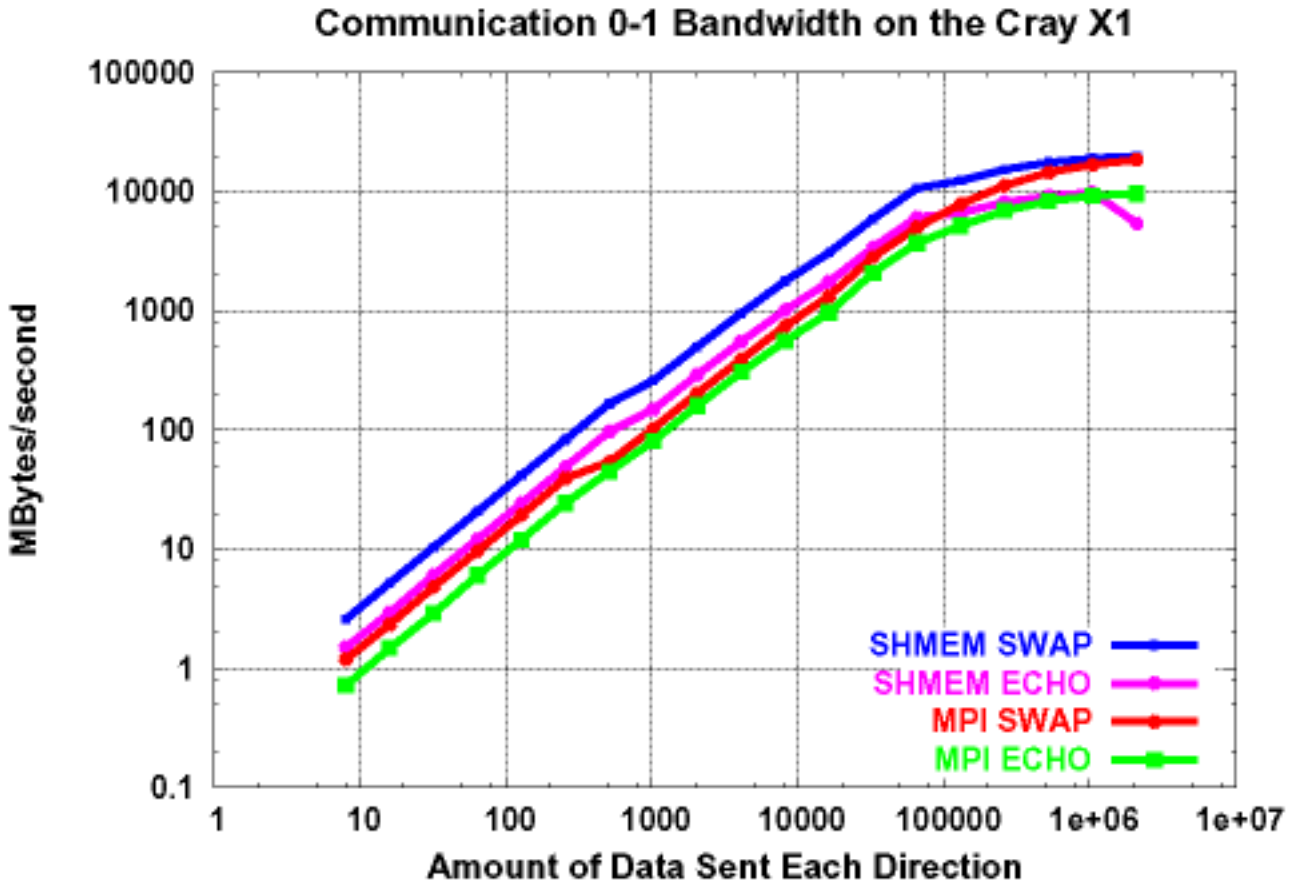
COMMTEST SWAP Benchmark II

Comparing performance of SWAP for different communication pattern, plotted on a log-linear scale. The single pair bandwidth has not reached its peak yet, but the two pair experiment bandwidth is beginning to reach its maximum.



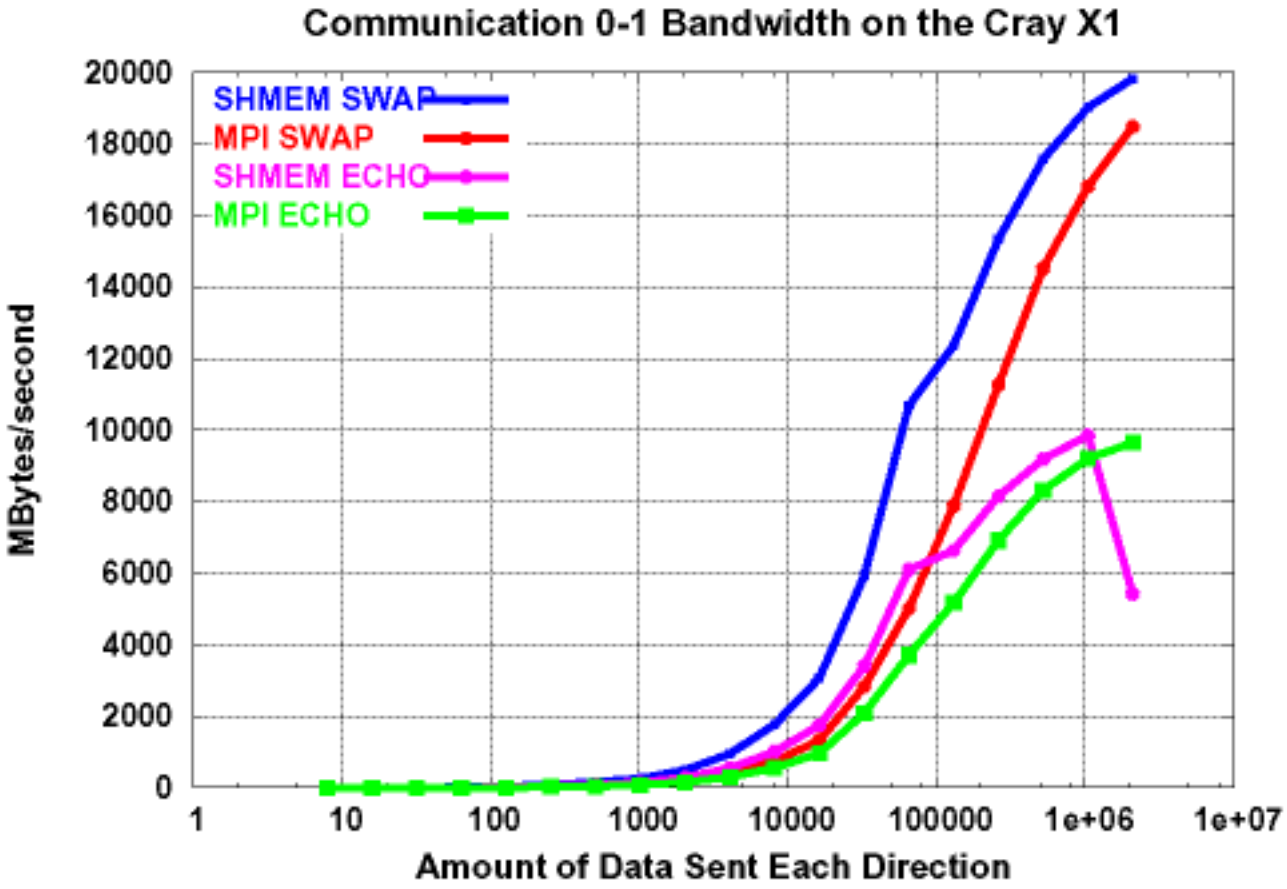
PSTSWM MPI vs. SHMEM 0-1 Comparison

Comparing MPI and SHMEM performance for 0-1 experiment, looking at both SWAP (bidirectional bandwidth) and ECHO (unidirectional bandwidth). SHMEM performance is better for all but the largest messages.



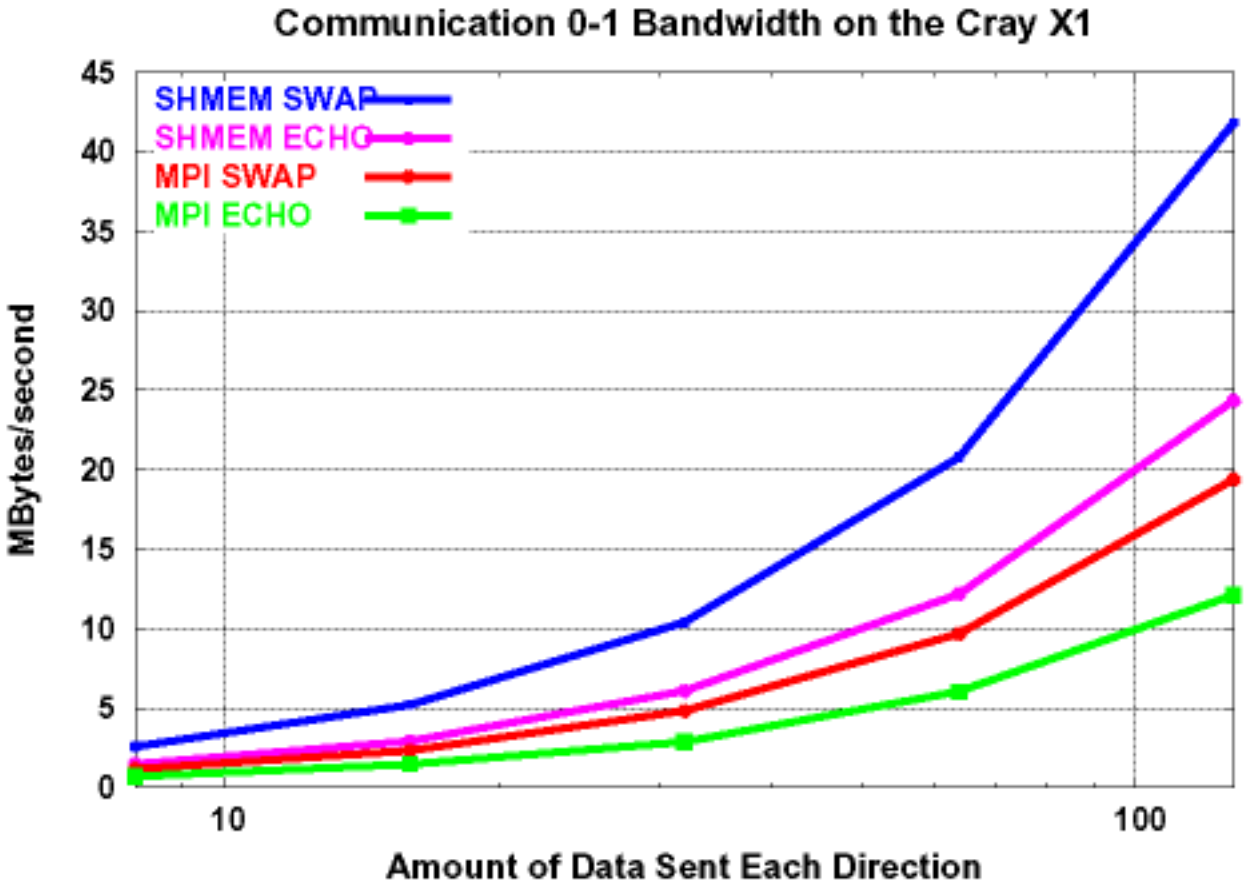
PSTSWM MPI vs. SHMEM 0-1 Comparison II

Comparing MPI and SHMEM performance for 0-1 experiment, using a log-linear scale. MPI performance is very near to that of SHMEM for large messages (when using SHMEM to implement two-sided messaging).



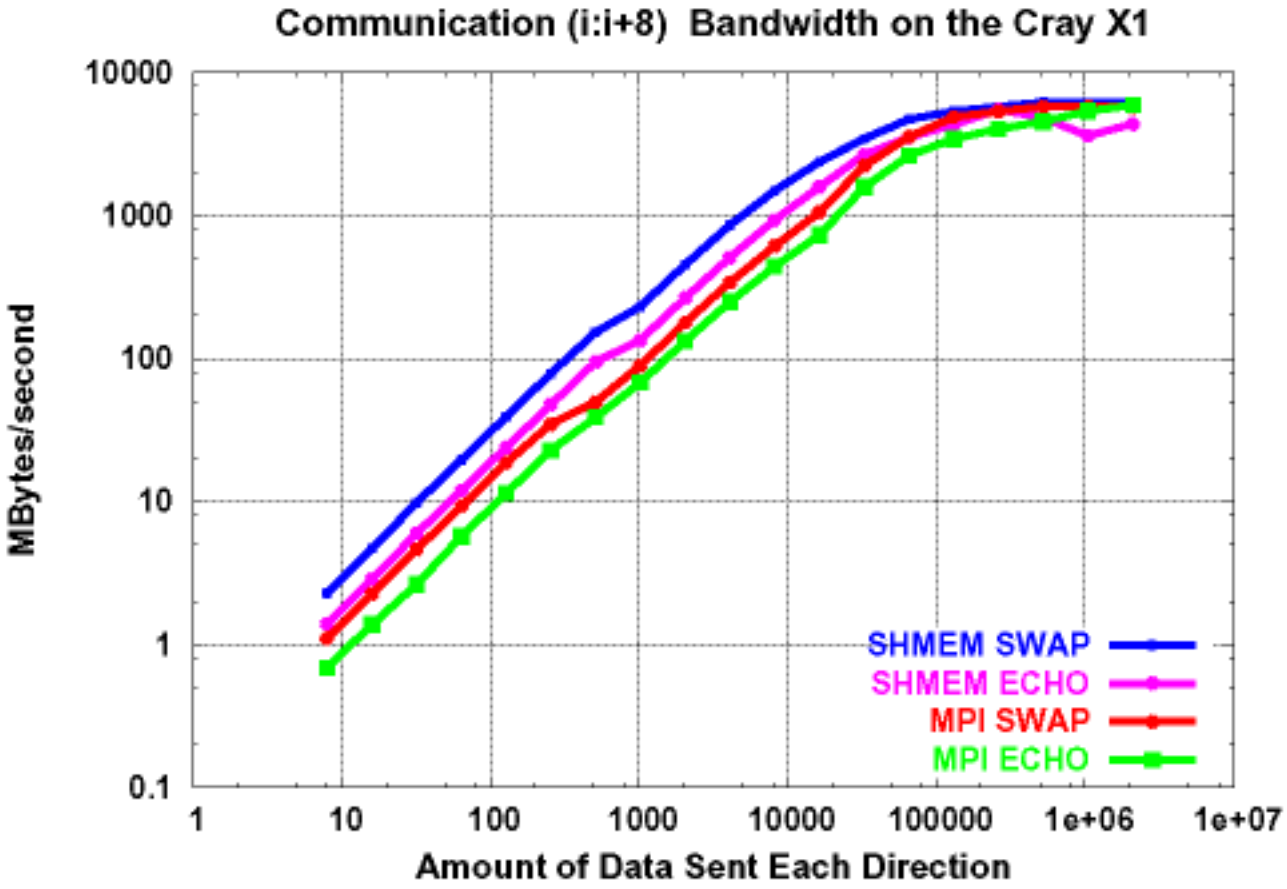
PSTSWM MPI vs. SHMEM 0-1 Comparison III

Comparing MPI and SHMEM performance for 0-1 experiment, using a log-linear scale and looking at small message sizes. The ECHO bandwidth is half of the SWAP bandwidth, so full bidirectional bandwidth is being achieved.



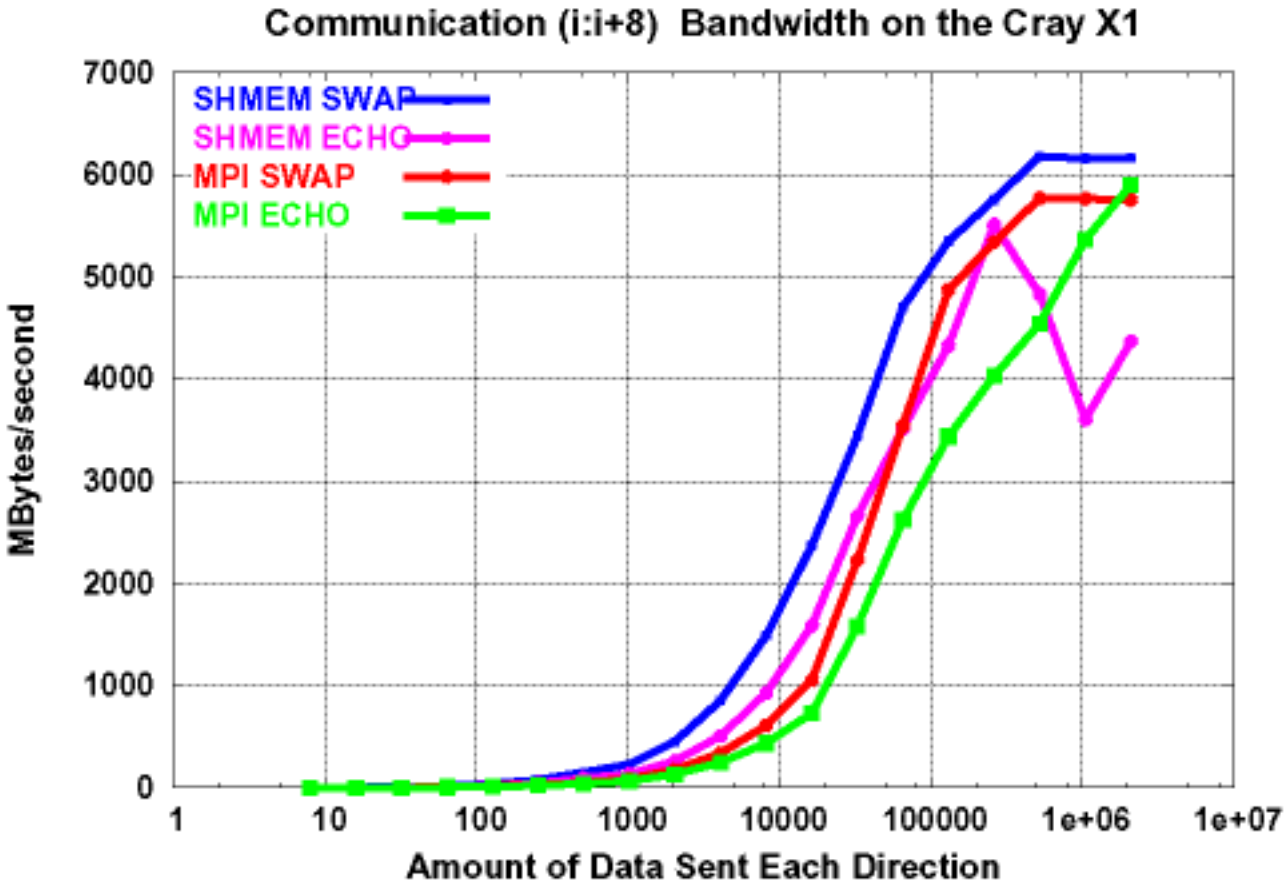
PSTSWM MPI vs. SHMEM i-(i+8) Comparison

Comparing MPI and SHMEM performance for i-(i+8) experiment, looking at both SWAP (bidirectional bandwidth) and ECHO (unidirectional bandwidth). Again, SHMEM performance is better for all but the largest messages.



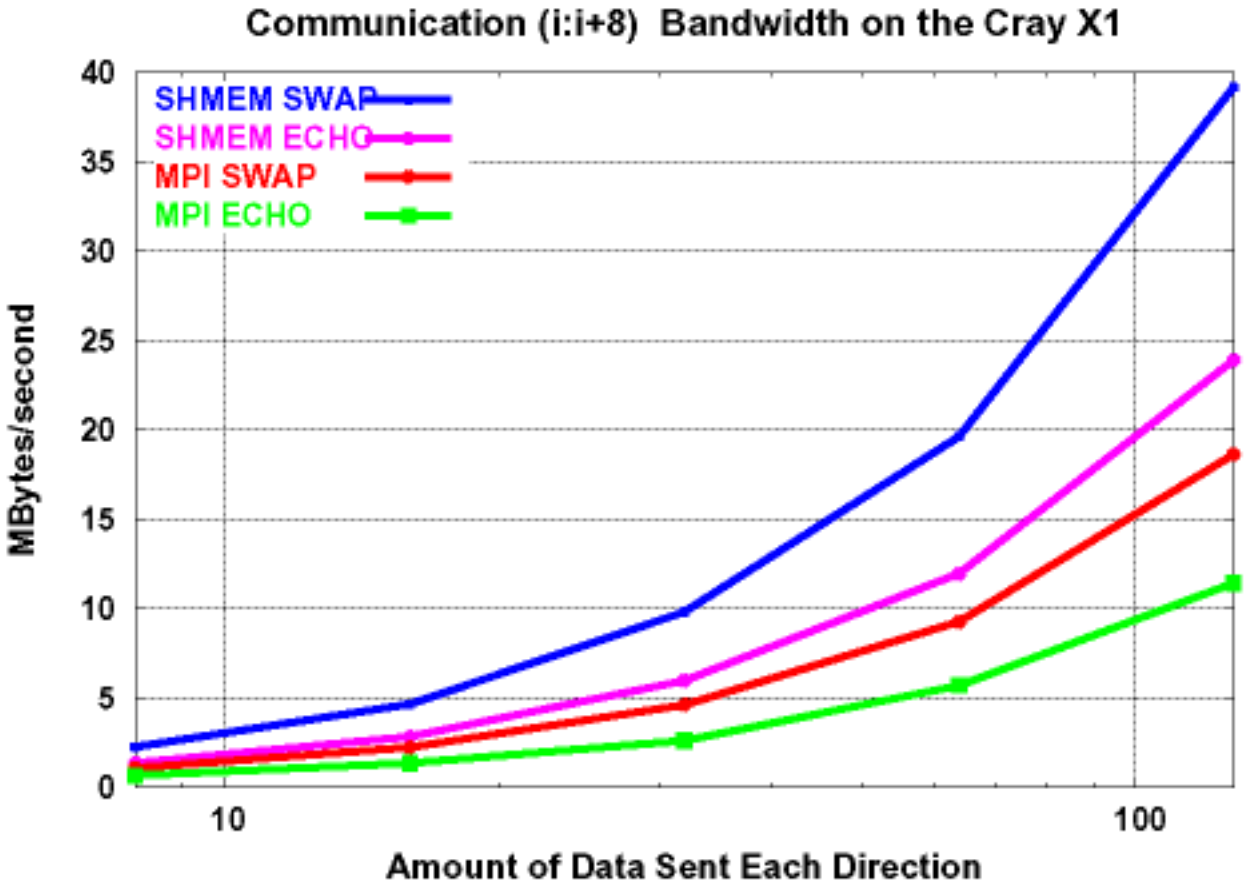
PSTSWM MPI vs. SHMEM i-(i+8) Comparison II

Comparing MPI and SHMEM performance for i-(i+8) experiment, using a log-linear scale. MPI performance is very near to that of SHMEM for large messages (when using SHMEM to implement two-sided messaging). For the largest message sizes, SWAP bandwidth saturates the network and MPI ECHO bandwidth exceeds MPI SWAP bandwidth.



PSTSWM MPI vs. SHMEM i-(i+8) Comparison III

Comparing MPI and SHMEM performance for i-(i+8) experiment, using a log-linear scale and looking at small message sizes. The ECHO bandwidth is more than half of the SWAP bandwidth, and something less than full bidirectional bandwidth is achieved.



Application Code

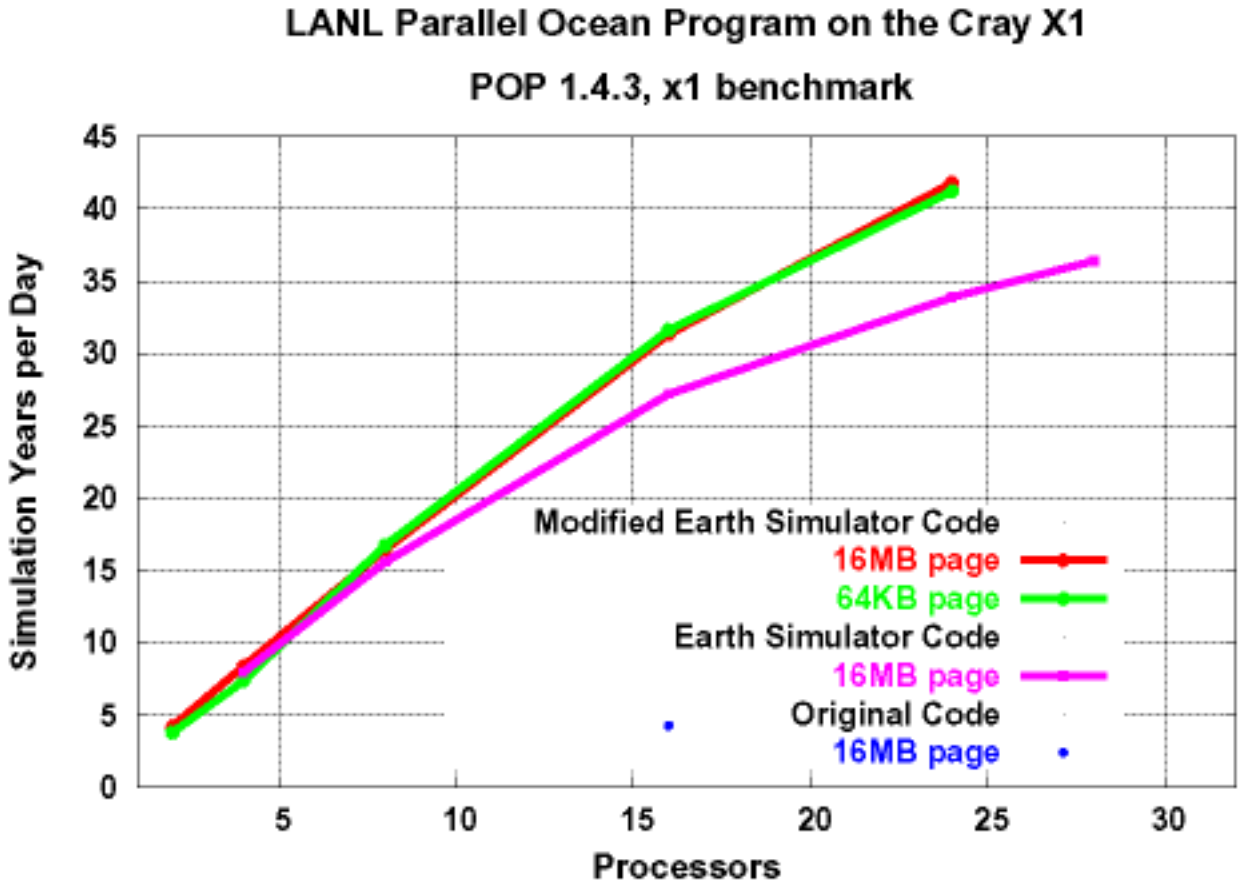
- Parallel Ocean Program (POP)
  - Developed at Los Alamos National Laboratory. Used for high resolution studies and as the ocean component in the Coupled Climate System Model (CCSM)
  - Ported to the Earth Simulator by Dr. Yoshikatsu Yoshida of the Central Research Institute of Electric Power Industry (CRIEPI).
  - Initial port to the Cray X1 by John Levesque of Cray, using Co-Array Fortran for conjugate gradient solver.
  - X1 and Earth Simulator ports merged and modified by Pat Worley of Oak Ridge National Laboratory.
  - Optimization on the X1 ongoing.

POP Experiment Particulars

- Two primary computational phases
  - Baroclinic: 3D with limited nearest-neighbor communication; scales well.
  - Barotropic: dominated by solution of 2D implicit system using conjugate gradient solves; scales poorly
- One benchmark problem size
  - One degree horizontal grid (“by one” or “x1”) of size 320x384x40
- Domain decomposition determined by grid size and 2D virtual processor grid. Results for a given processor count are the best observed over all applicable processor grids.

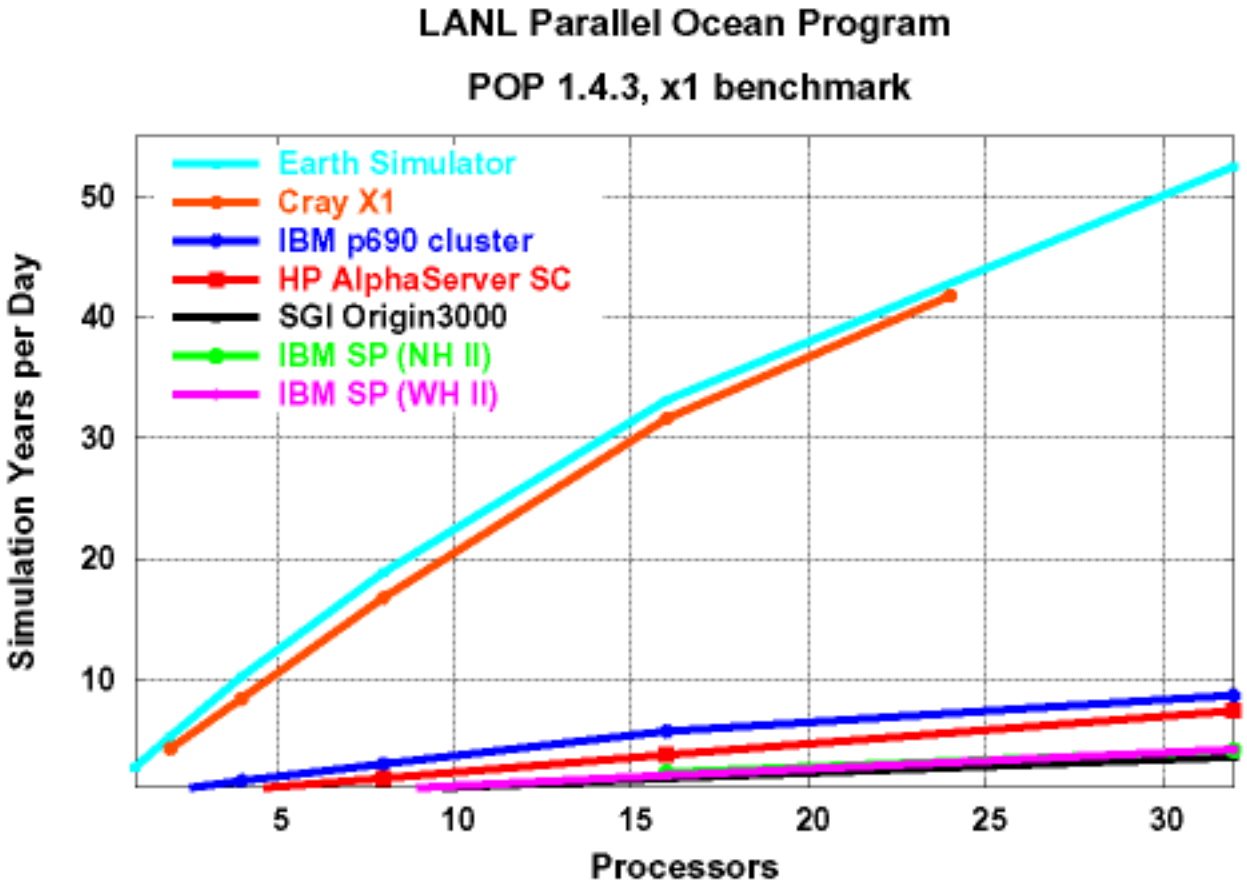
POP Version and Page Size Comparison

Comparing performance of different versions of POP and when running with different page sizes. Unlike PSTSWM, the page size is not a performance issue for POP, for this problem size. The Earth Simulator version performs reasonably well. Most of the performance improvement in the current version is due to Co-Array implementation of the linear solver.



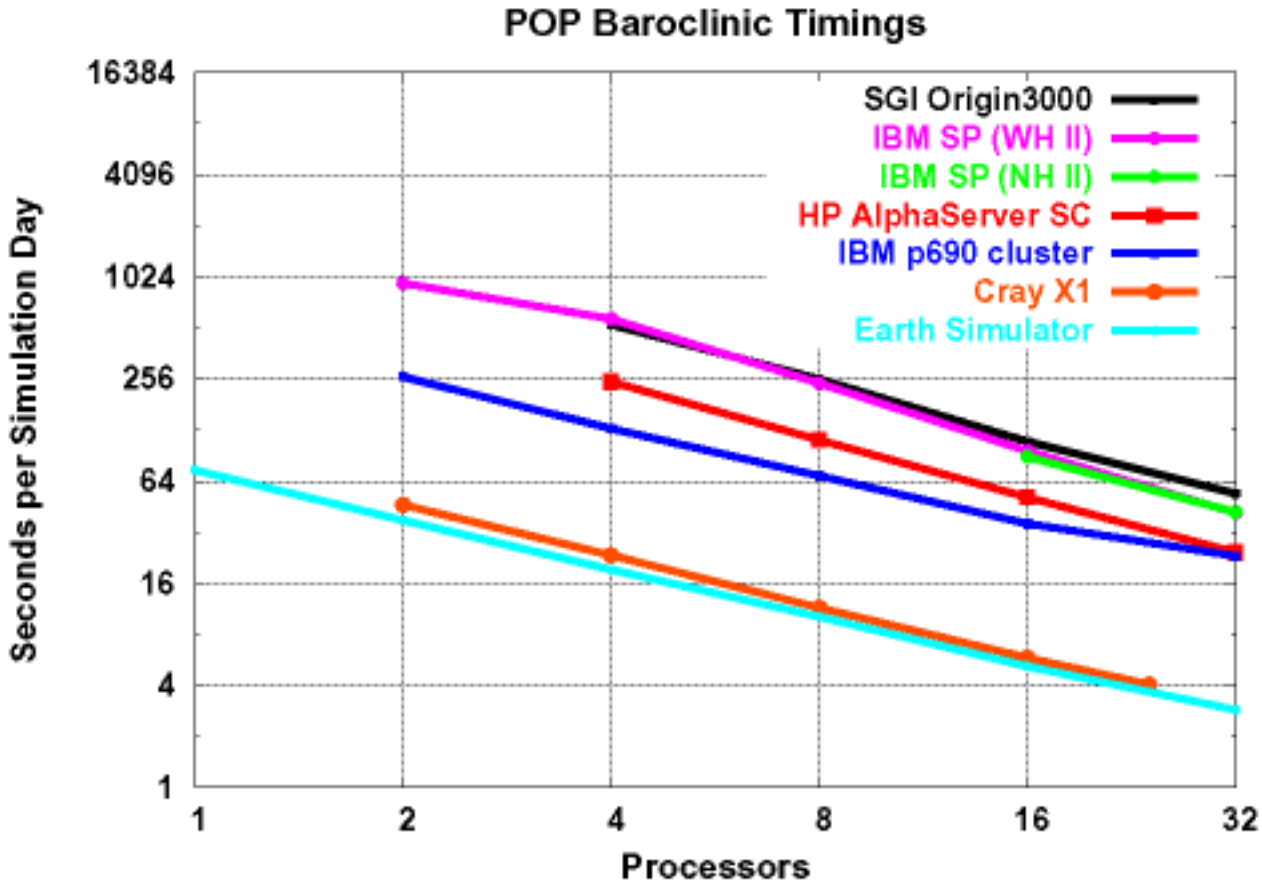
POP Benchmark

Comparing performance of POP. The current X1 performance is almost the same as that of the Earth Simulator, even though significant portions of the code are unchanged from the Earth Simulator port. Performance is much better than on the other platforms.



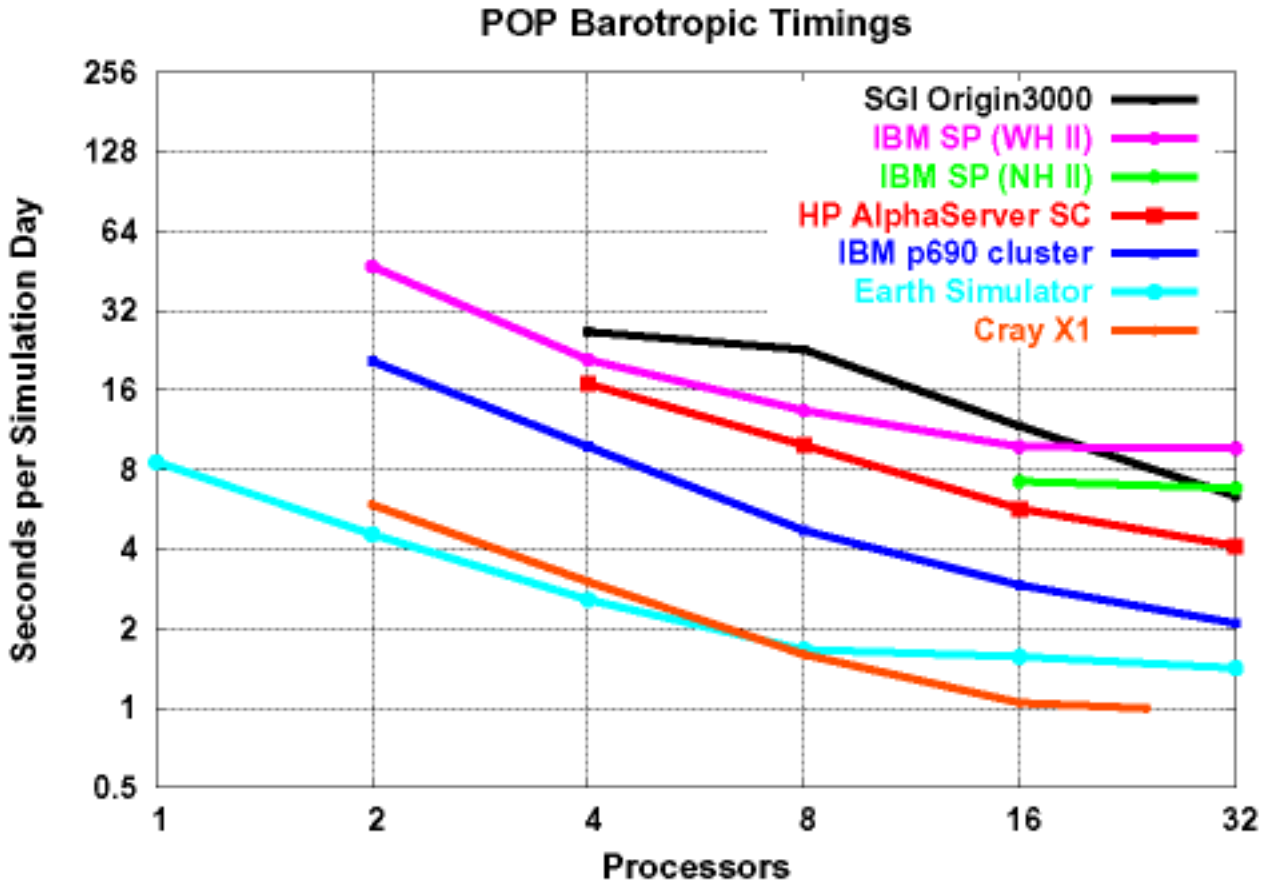
POP Baroclinic Process Scaling

Comparing timings for POP Baroclinic process. Scaling is excellent on all systems but the p690. X1 and Earth Simulator is similar, and converging.



POP Barotropic Process Scaling

Comparing timings for POP Barotropic process. Scaling is a problem on all systems, but especially on the vector systems. The Co-Array implementation of the lienar solver gives the advantage to the X1 for larger processor counts.



Conclusions?

- System Works.
- We need more experience with application codes.
- We need experience on a larger system, with more memory.
- SHMEM and Co-Array Fortran performance can be superior to MPI. However, we hope that MPI small message performance can be improved.
- Page size can impact performance.
- Both SSP and MSP modes of execution work fine. MSP mode should be preferable for fixed size problem scaling, but which is better is application and problem size specific.

Questions ? Comments ?

For further information on these and other evaluation studies, visit

http://www.csm.ornl.gov/evaluation .