

Early Evaluation of the Cray X1

Patrick H. Worley
Thomas H. Dunigan Jr.
Mark R. Fahey
James B. White III

Oak Ridge National Laboratory

SC03

November 19, 2003

Phoenix Convention Center

Phoenix, Arizona

Acknowledgements

- Research sponsored by the Atmospheric and Climate Research Division and 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.
- These slides have been authored by a contractor of the U.S. Government under contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes
- Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the United States Department of Energy under Contract No. DE-AC05-00OR22725.

Evaluation of Early Systems

A project that attempts to evaluate *quickly* the promise of “early” (possibly immature) systems:

- Verifying advertised functionality and performance
- Quantifying performance impact of unique system characteristics
- Providing guidance to (early) users
 - What performance to expect
 - Performance quirks and bottlenecks
 - Performance optimization tips

Early Systems

ORNL is currently “blessed” with a number of early systems:

- Cray X1
 - 64 processors installed in March 2003; upgraded to final 256 processor configuration on 10/14/03.
- SGI Altix
 - Initial system installed in August 2003; upgraded to 1.5 GHz processors on 10/15/03.
- IBM Federation switch (linking 32-way p690 nodes)
 - Part of Early Ship Program; pre-GA hardware delivered in October 2003.

Evaluation Methodology

“Measure early, measure often, analyze just in time”

- Hierarchical evaluation
 - Microbenchmarks
 - Application-relevant kernels
 - Compact or full parallel application codes
- Open evaluation
 - Rapid posting of evaluation results
 - Systems available to external performance researchers
- Fair evaluation
 - Determining appropriate ways of using system, evaluating *both* traditional and alternative programming paradigms
 - Collecting data with *both* standard and custom benchmarks

Phoenix

Cray X1 with 64 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 256 processors (MSPs), 1024 GB of memory, and 3200 GF/s peak performance.



OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY


UT-BATTELLE

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.
- 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.
- 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.
- SGI Altix 3700 at ORNL: 2 128-way SMP nodes and NUMAflex fat-tree interconnect. Each processor is a 1.5 GHz Itanium 2 with a 6 MB L3 cache

Outline

Quick sampling of current results ...

- Standard kernel benchmarks (unmodified)
- Parallel application success stories
 - POP ocean code
 - GYRO fusion code

For custom kernel and microbenchmark measurements of subsystem performance, see the paper.

For more performance data, visit

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

Also visit Army HPC Research Center Booth for additional application success stories.

Caveats

- These are EARLY results (even on the Cray after 6 months), resulting from sporadic benchmarking on evolving system software and hardware configurations.
- Performance characteristics are still changing, due to continued evolution of OS and compilers and libraries.

Aside: What is a processor on the X1?

It depends on why you are asking the question ...

- As a user, I want fewer, more powerful, processors, in order to minimize parallel overheads. So I *want* an MSP to be “the processor”.
- As a user, a processor is what my code views as a processor. If it is written explicitly to exploit SSPs, then the SSP is the processor. If the compiler assigns work to SSPs without any intervention other than possibly simple compiler directives, then an MSP is the processor.
- The Cray X1 compiler can often partition inner loops and assign work to functional units in different SSPs (within the same MSP) and achieve good performance. This level of integration is typically seen only within a processor, so it is reasonable to call an MSP a processor in these cases.

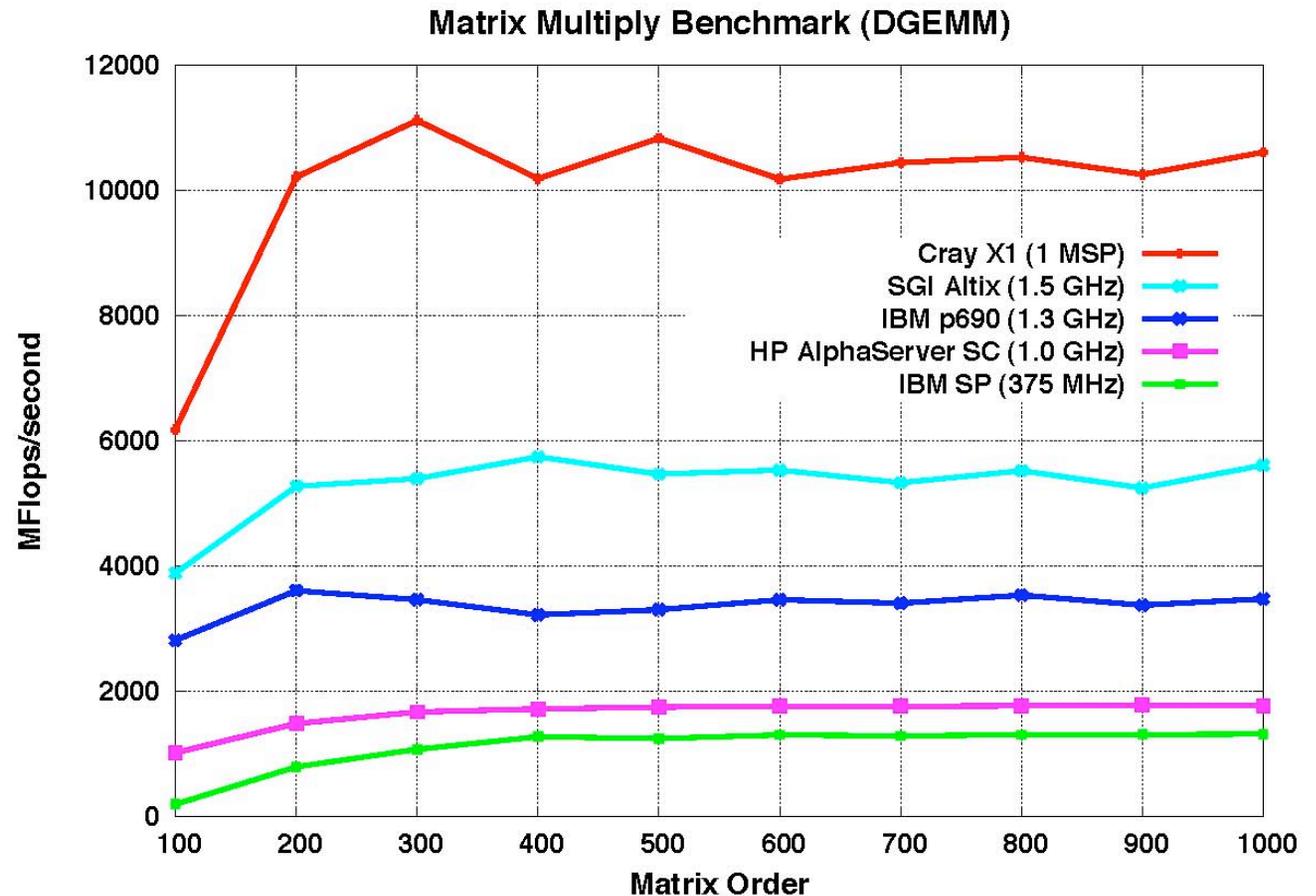
Kernel Benchmarks

- Single Processor Performance
 - DGEMM matrix multiply benchmark
 - Euroben MOD2D dense eigenvalue benchmark
 - Euroben MOD2E sparse eigenvalue benchmark
- Interprocessor Communication Performance
 - HALO benchmark

DGEMM Benchmark

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

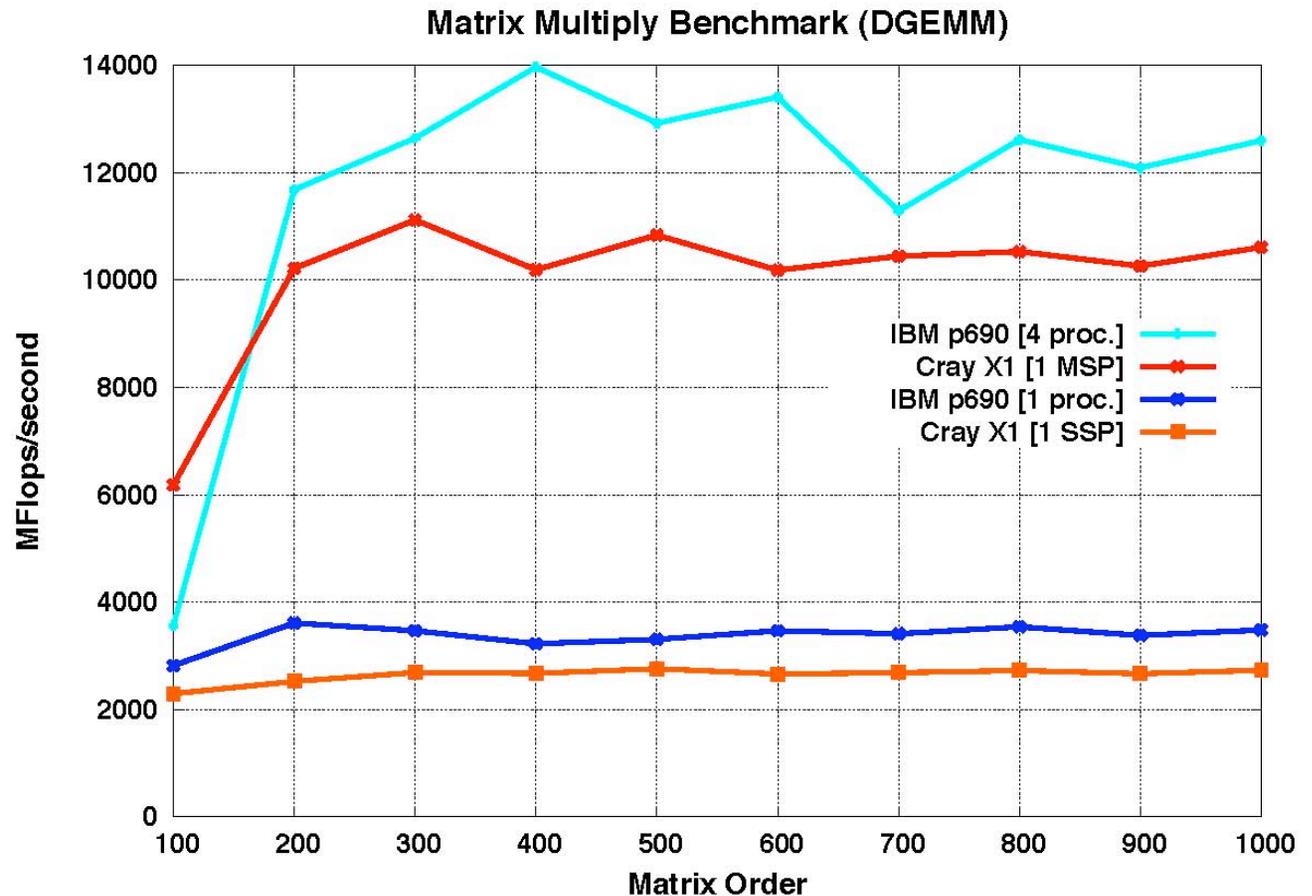
Good performance achieved, reaching 80% of peak relatively quickly.



DGEMM Benchmark - What's a Processor?

Comparing performance of X1 MSP, X1 SSP, p690 processor, and four p690 processors (in a single MCM using PESSL parallel library).

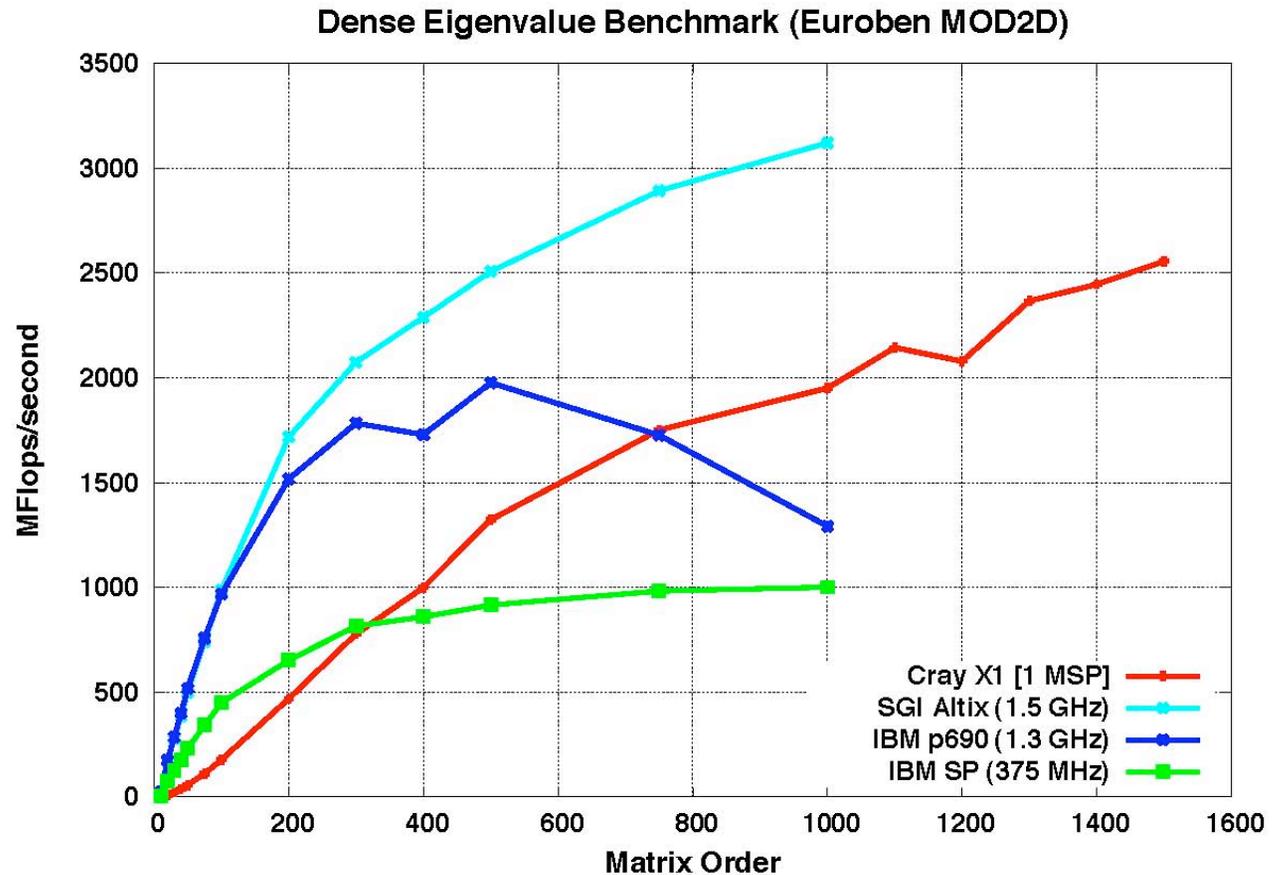
Max. percentage of peak -
X1 SSP: 86%
X1 MSP: 87%
p690 (1): 70%
p690 (4): 67%



MOD2D Benchmark

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

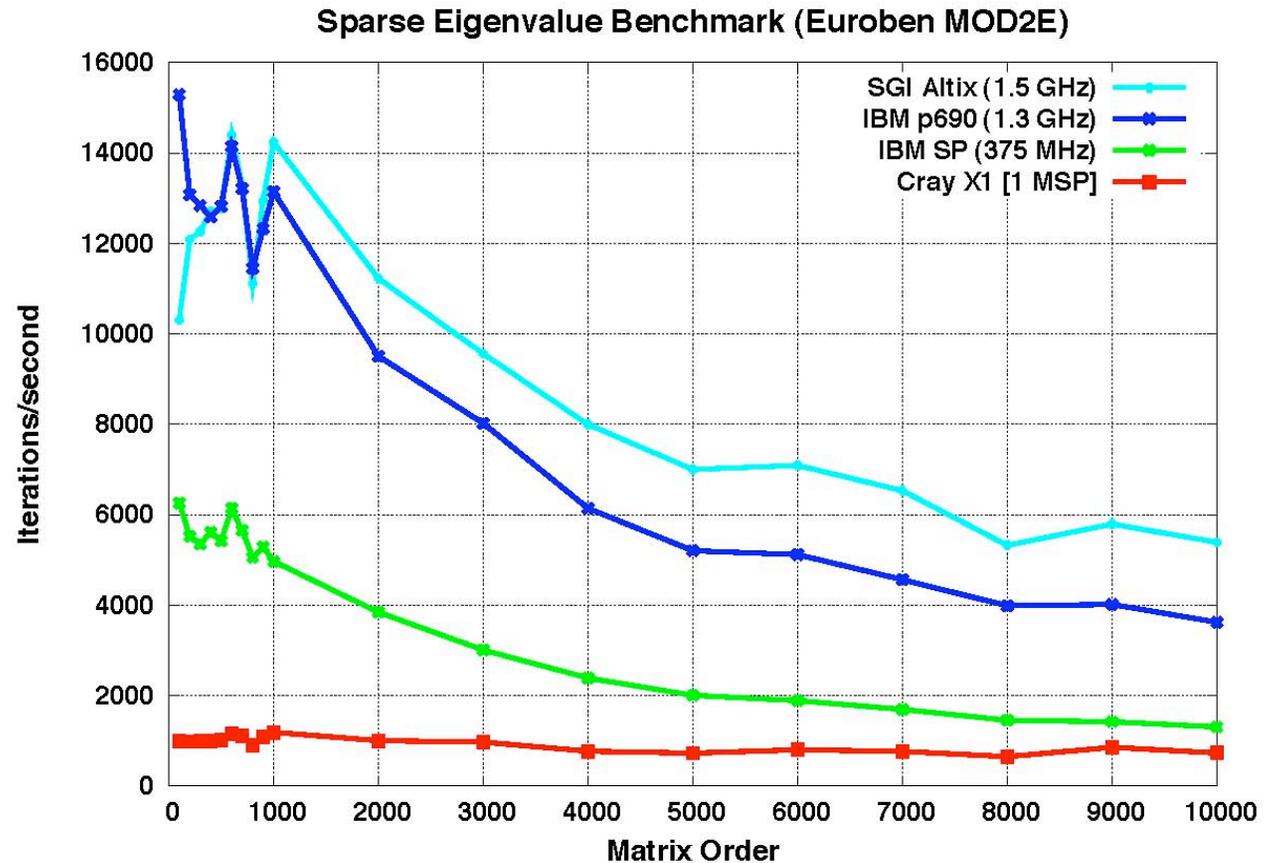
Performance still growing with problem size for Cray and SGI. Performance of IBM systems has peaked.



MOD2E Benchmark

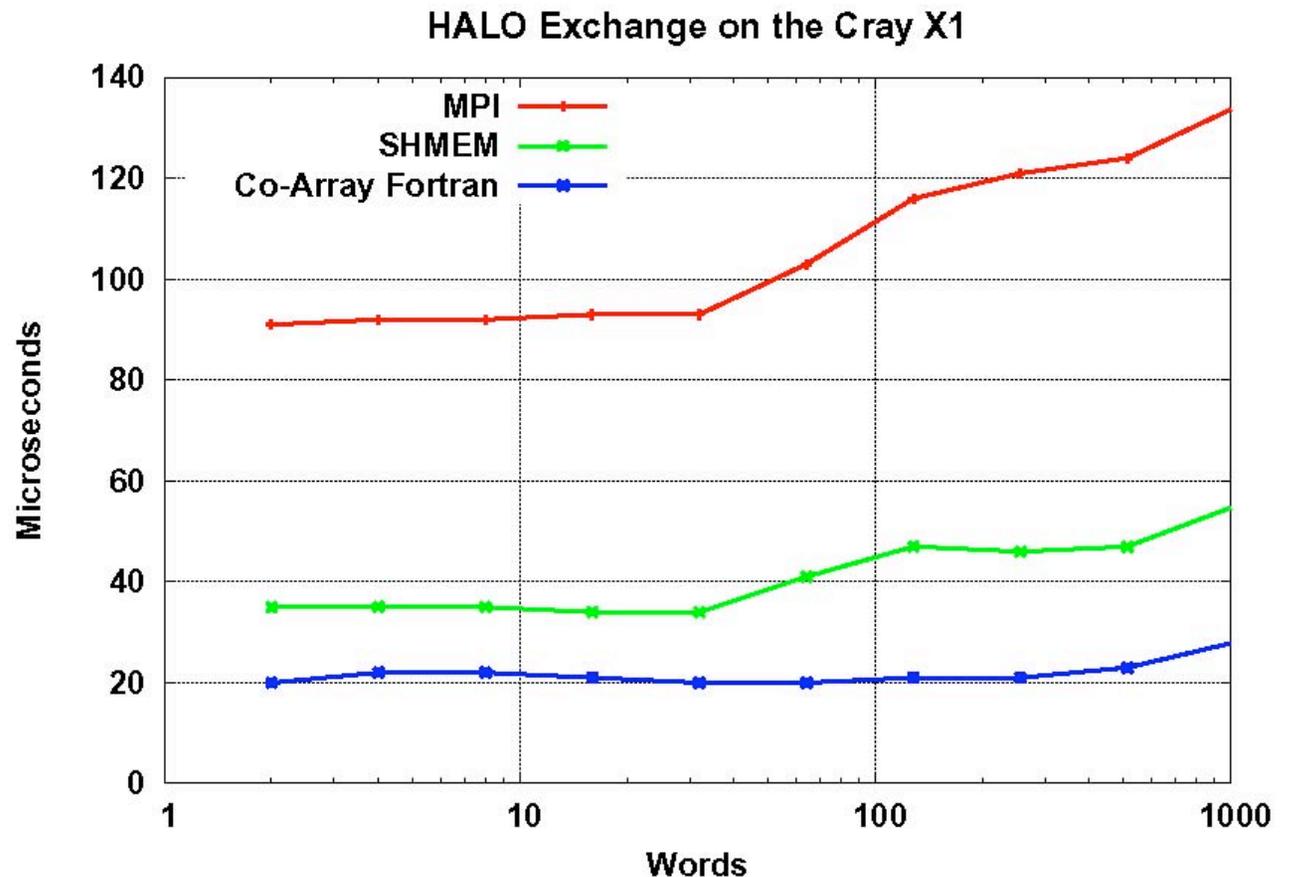
Comparing performance of Fortran code for sparse eigenvalue analysis. Aggressive compiler options were used on the X1, but code was not restructured and compiler directives were not inserted. Performance is improving for larger problem sizes, so some streaming or vectorization is being exploited. Performance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

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY



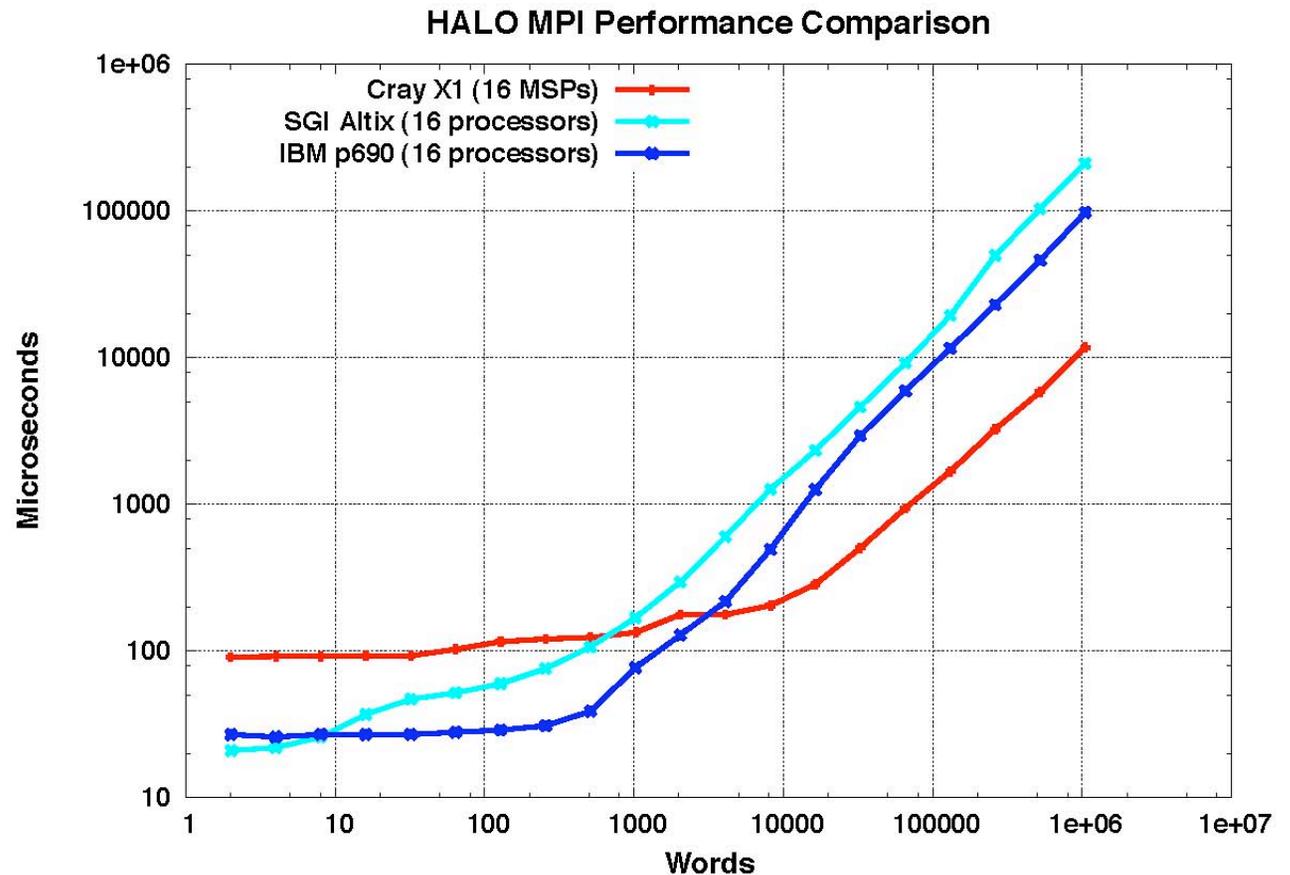
HALO Paradigm Comparison

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



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 SMP) and the SGI Altix (within a 128 processor SMP). 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.



Summary of Kernel Benchmarks

- The Cray X1 is a vector architecture. Codes that do not vectorize or which have very short vector lengths will not perform well.
- Interprocessor communication latency is low on the X1 if use Co-Array Fortran or SHMEM, but not if use MPI (currently).
- Interprocessor bandwidth is excellent on the X1, with MPI, SHMEM, or Co-Array Fortran.

Application Benchmark I: Parallel Ocean Program (POP)

- Developed at Los Alamos National Laboratory. Used for high resolution studies and as the ocean component in the Community 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 and Trey White of Oak Ridge National Laboratory.
- Optimization on the X1 ongoing.

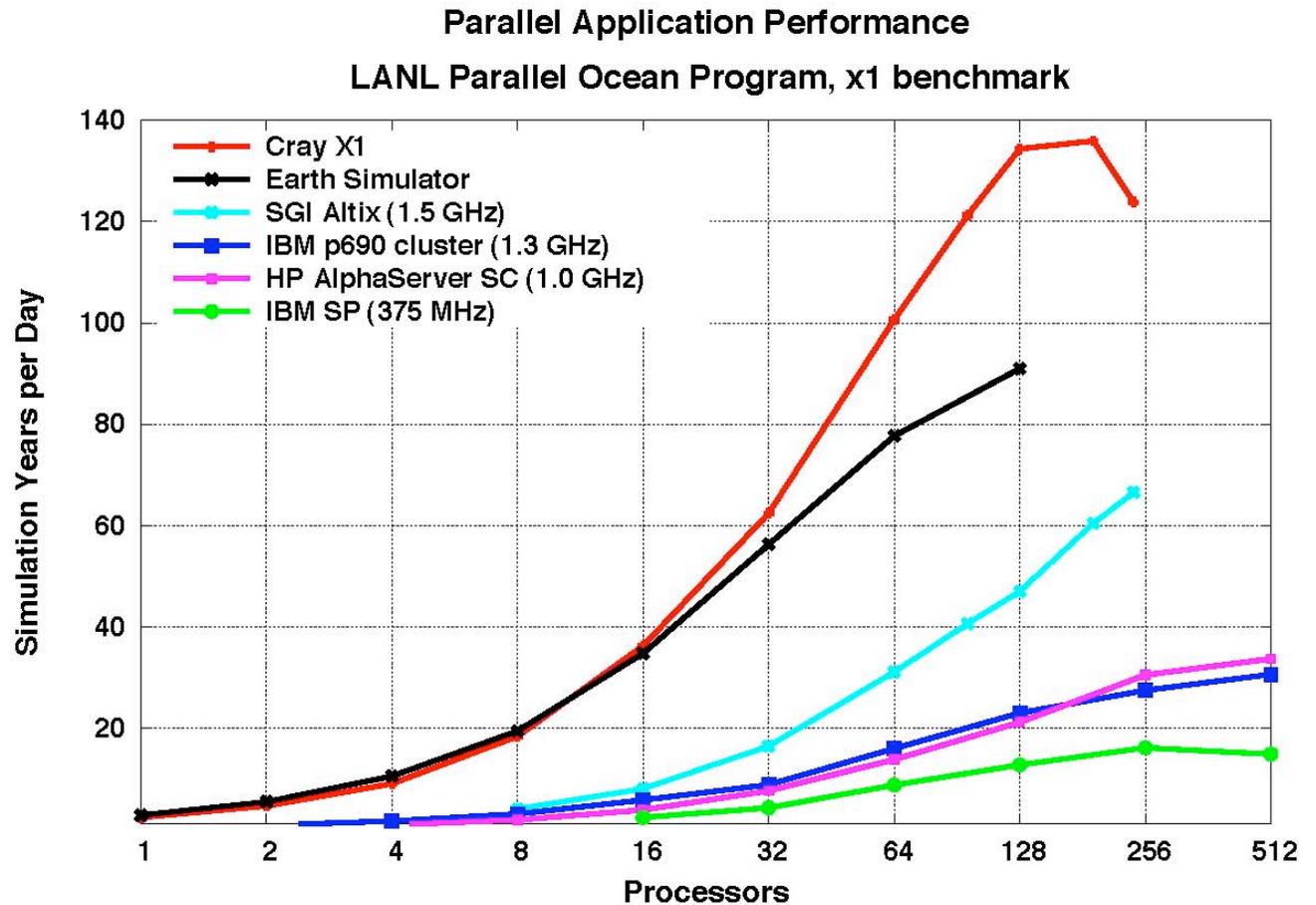
POP Experiment Particulars

- 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 (small, but an important size for climate modeling)
- 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 Simulation Rate

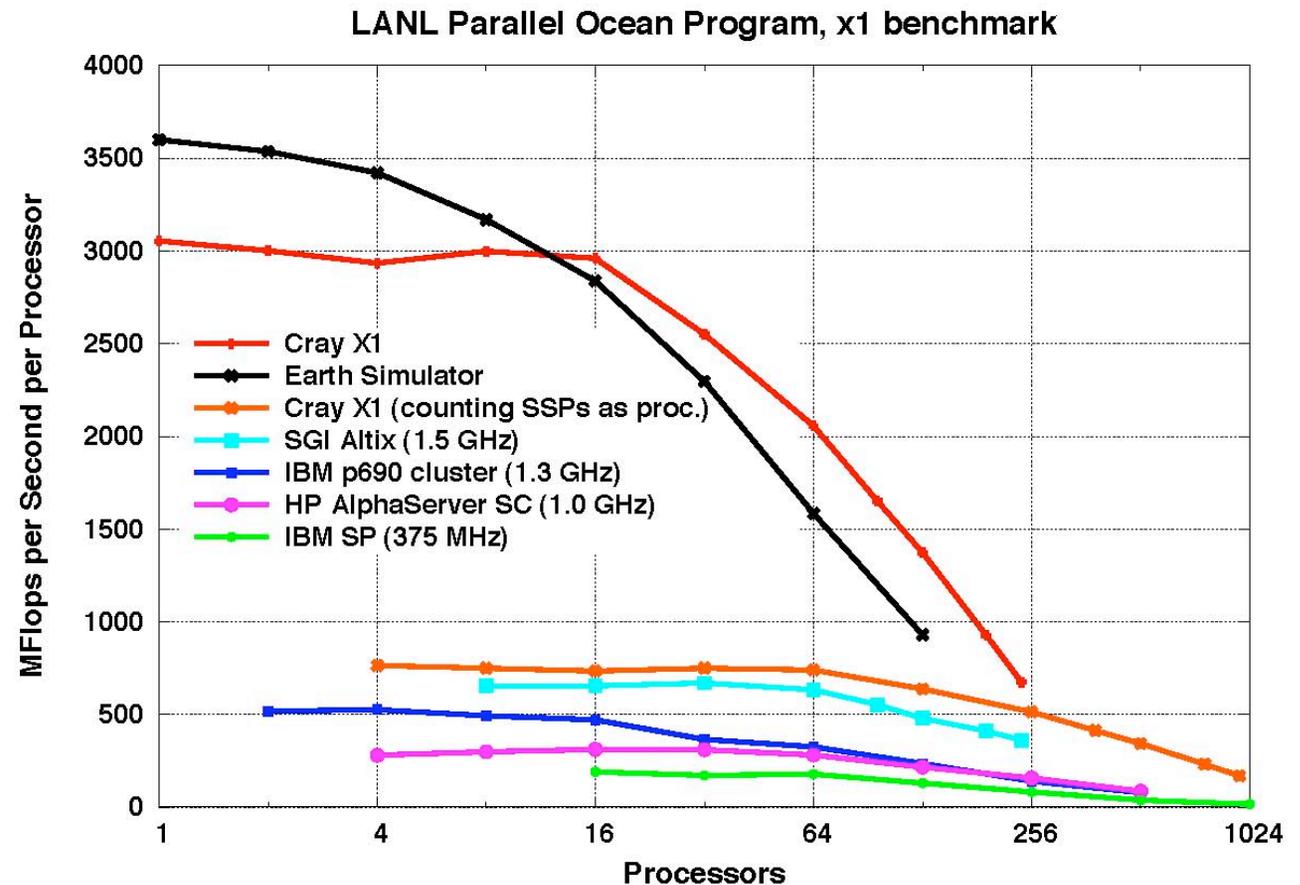
Comparing performance and scaling across platforms.

- Earth Simulator results courtesy of Dr. Y. Yoshida of the Central Research Institute of Electric Power Industry (CRIEPI).
- IBM SP results courtesy of Dr. T. Mohan of Lawrence Berkeley National Laboratory (LBNL)



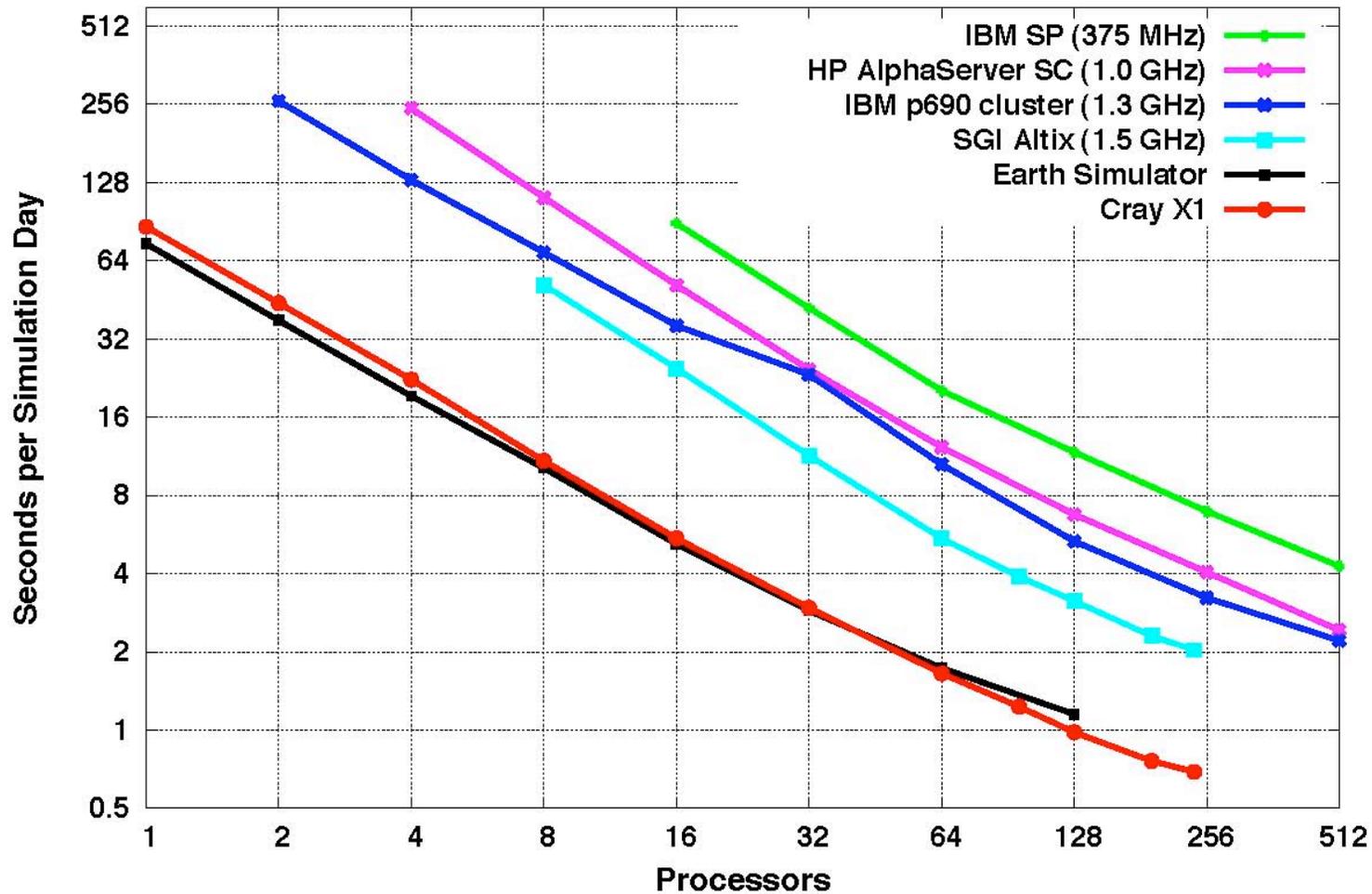
POP Computational Rate

- X1 is not as fast as the Earth Simulator for small processor counts. X1 maintains performance better as granularity (and vector length) decrease.
- POP views the MSP as the processor. However, replotting data with SSPs as the processor still shows a significant performance advantage compared to the nonvector processors.



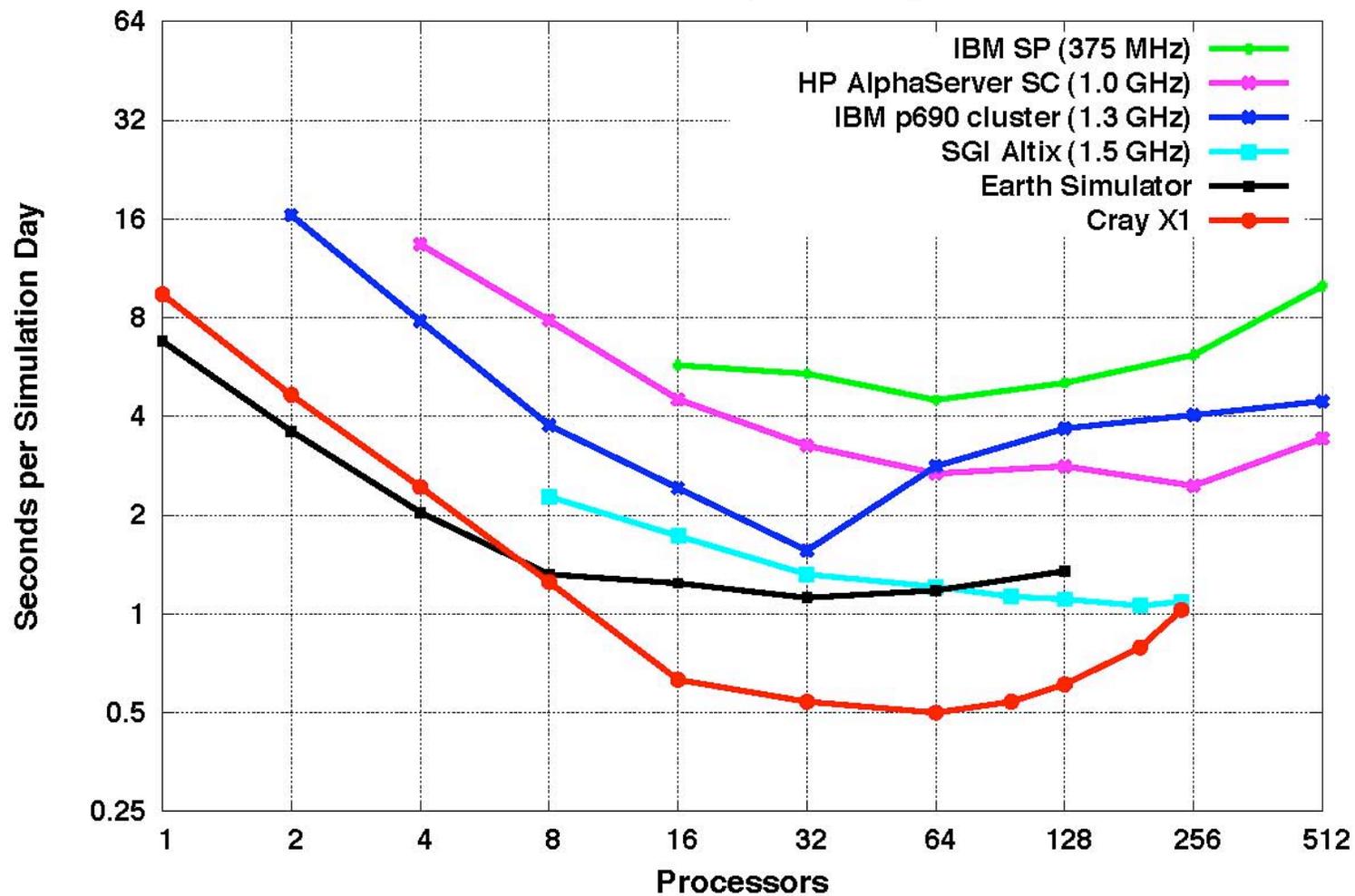
POP Performance Diagnosis: Baroclinic

POP Baroclinic Timings



POP Performance Diagnosis: Barotropic

POP Barotropic Timings



POP Performance Diagnosis

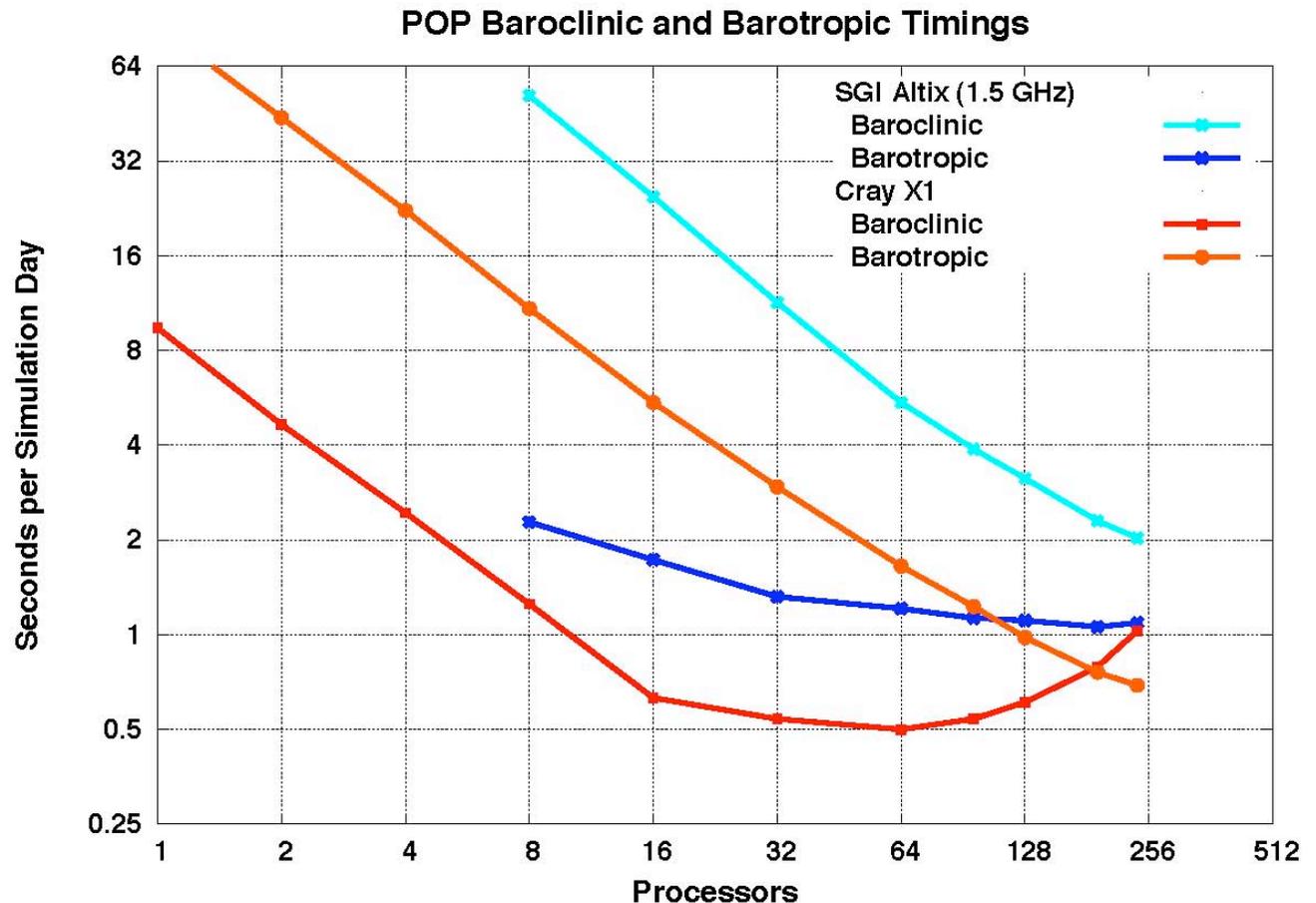
Cray X1

Communication-bound for more than 192 processors, with communication costs increasing. Communication algorithms known to have scaling problems, and alternatives being developed.

SGI Altix

Not yet communication bound. Using MPI point-to-point and collectives for barotropic. Initial experiments with SHMEM do not show significant improvement.

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY



What's next for POP?

- Additional Cray X1 optimizations
 - Scalable (tree-based) allreduce
 - Cray-specific vectorization
- Increased resolution
 - 0.1 degree resolution
- More recent versions of the model
 - CCSM version of POP 1.4.3
 - POP 2.0
 - HYPOP

Application Benchmark II: GYRO

- GYRO is an Eulerian gyrokinetic-Maxwell solver developed by R.E. Waltz and J. Candy at General Atomics. It is used in the DOE SciDAC Fusion Energy project studying plasma microturbulence.
- GYRO comes with ports to a number of different platforms. The port and optimization on the Cray X1 is primarily due to Mark Fahey of ORNL. In the Cray X1 port, GYRO is coded as if the MSP is the processor.
- Optimization on the X1 ongoing.

GYRO Experiment Particulars

Two benchmark problems, both time dependent:

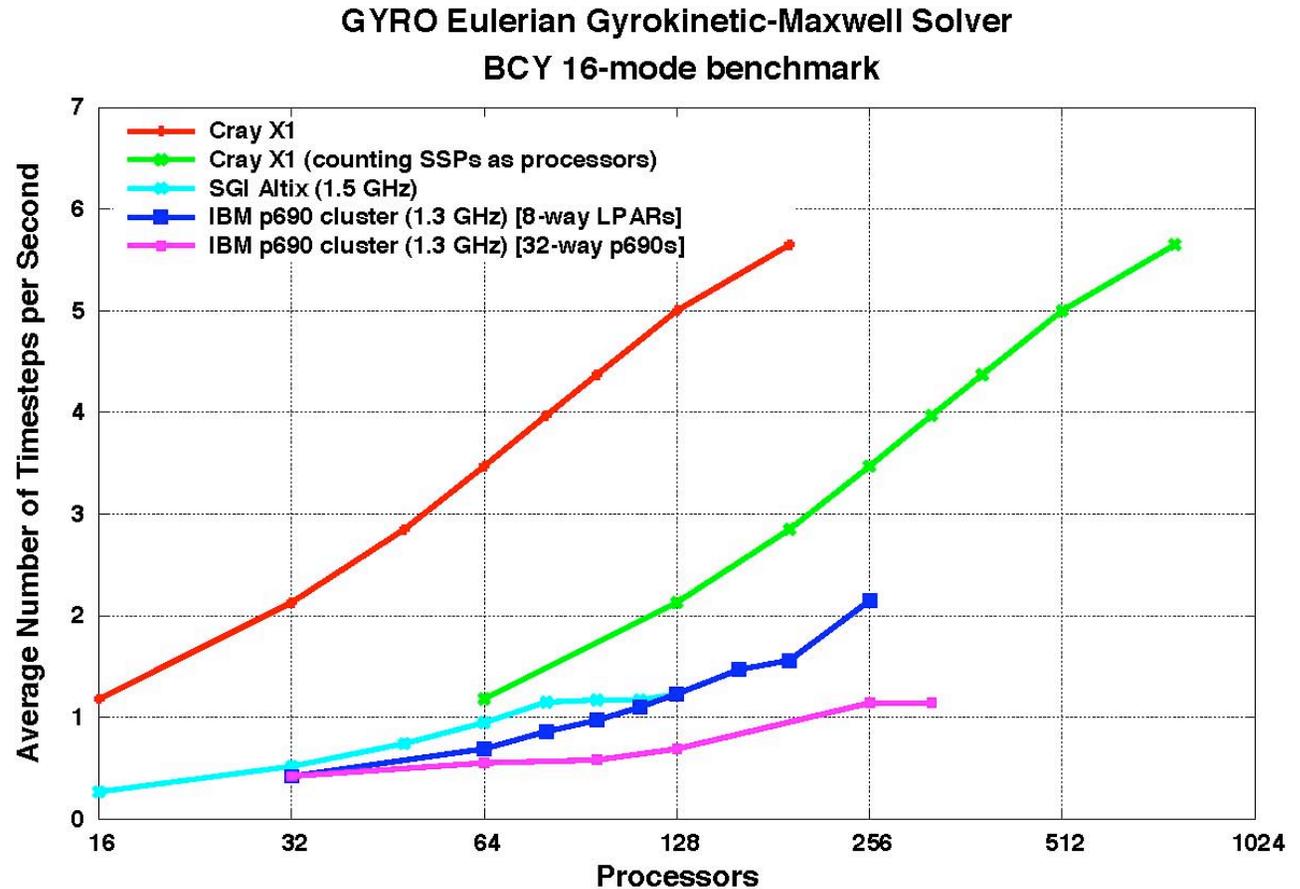
1. BCY.n16.b.25
 - 16-mode electromagnetic case. It is run on multiples of 16 processors. Duration is 8 simulation seconds, representing 1000 timesteps.
2. GTC.n64.500
 - 64-mode adiabatic electron case. It is run on multiples of 64 processors. Duration is 3 simulation seconds, representing 100 timesteps.

Current production runs use 32 modes, so benchmark #1 is somewhat small, while benchmark #2 is very large. (J. Candy is in the process of reformulating these benchmarks to also cover production-size problems.)

GYRO Simulation Rate

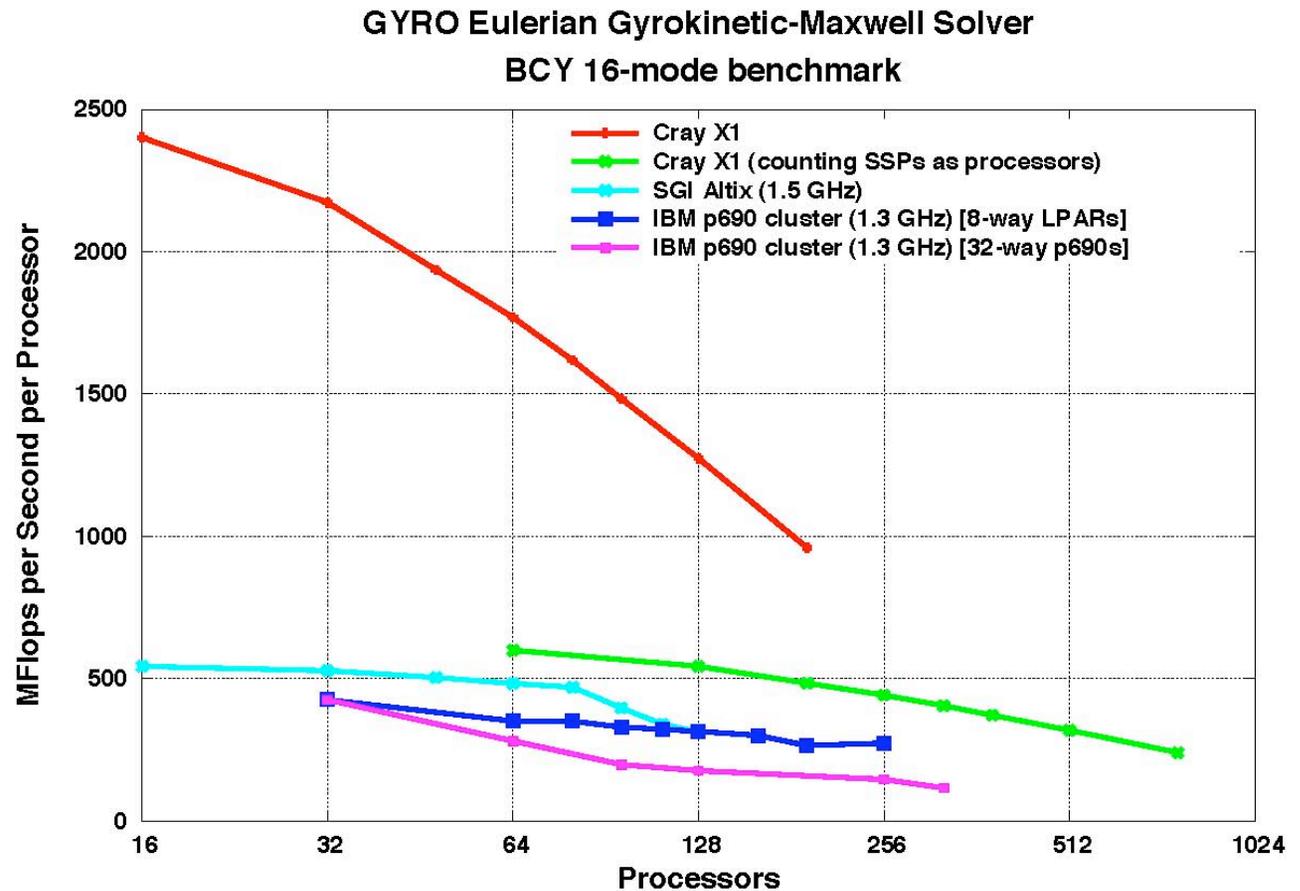
Comparing performance and scaling across platforms.

- X1 performance is significantly better than that on other platforms, even for this modest size problem, and advantage grows with processor count. Even replotting data with SSPs as processors indicates that the X1 is the faster platform.



GYRO Computational Rate

- IBM performance is limited by communication overhead.
- X1 performance is limited by nonscaling part of parallel algorithm. At 192 processors, 25% of the time is spent in a phase that does not scale beyond 16-way parallelism.

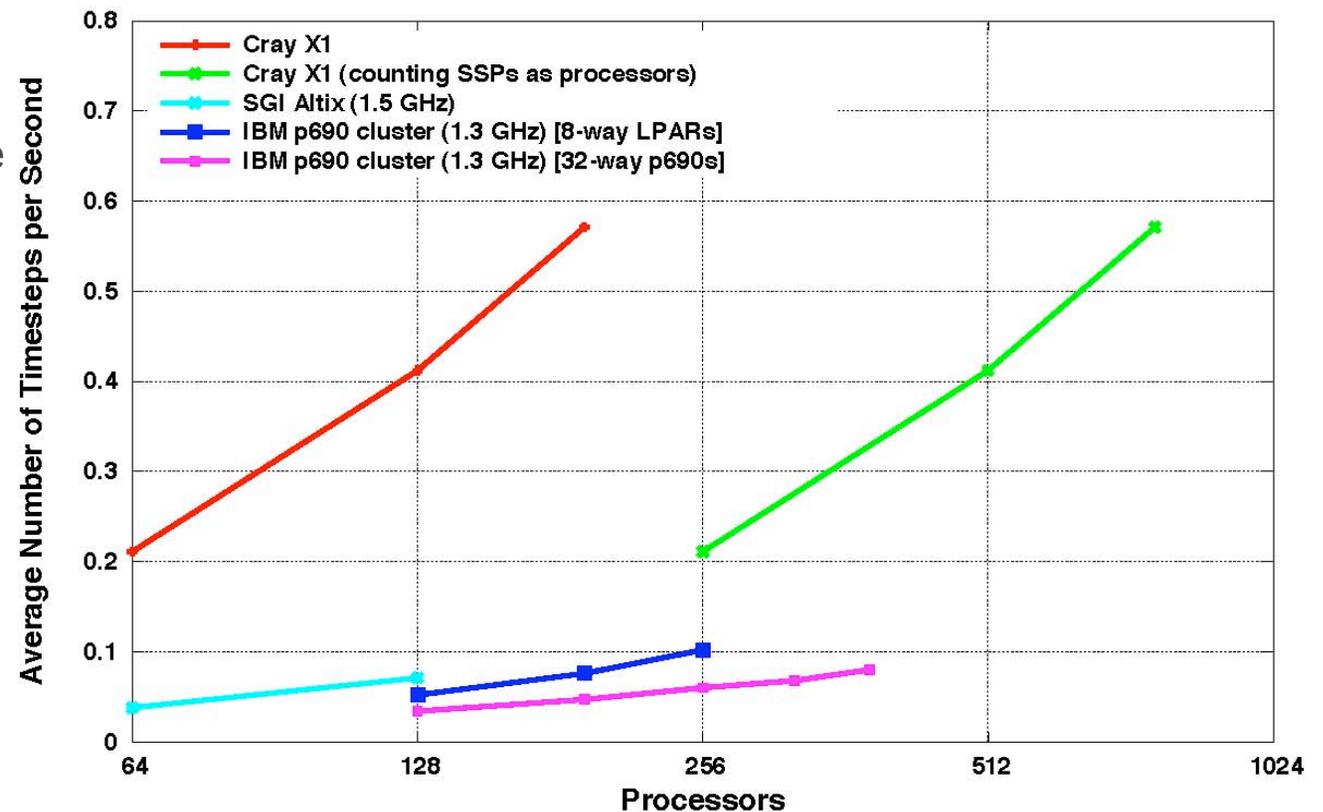


GYRO Simulation Rate

Comparing performance and scaling across platforms.

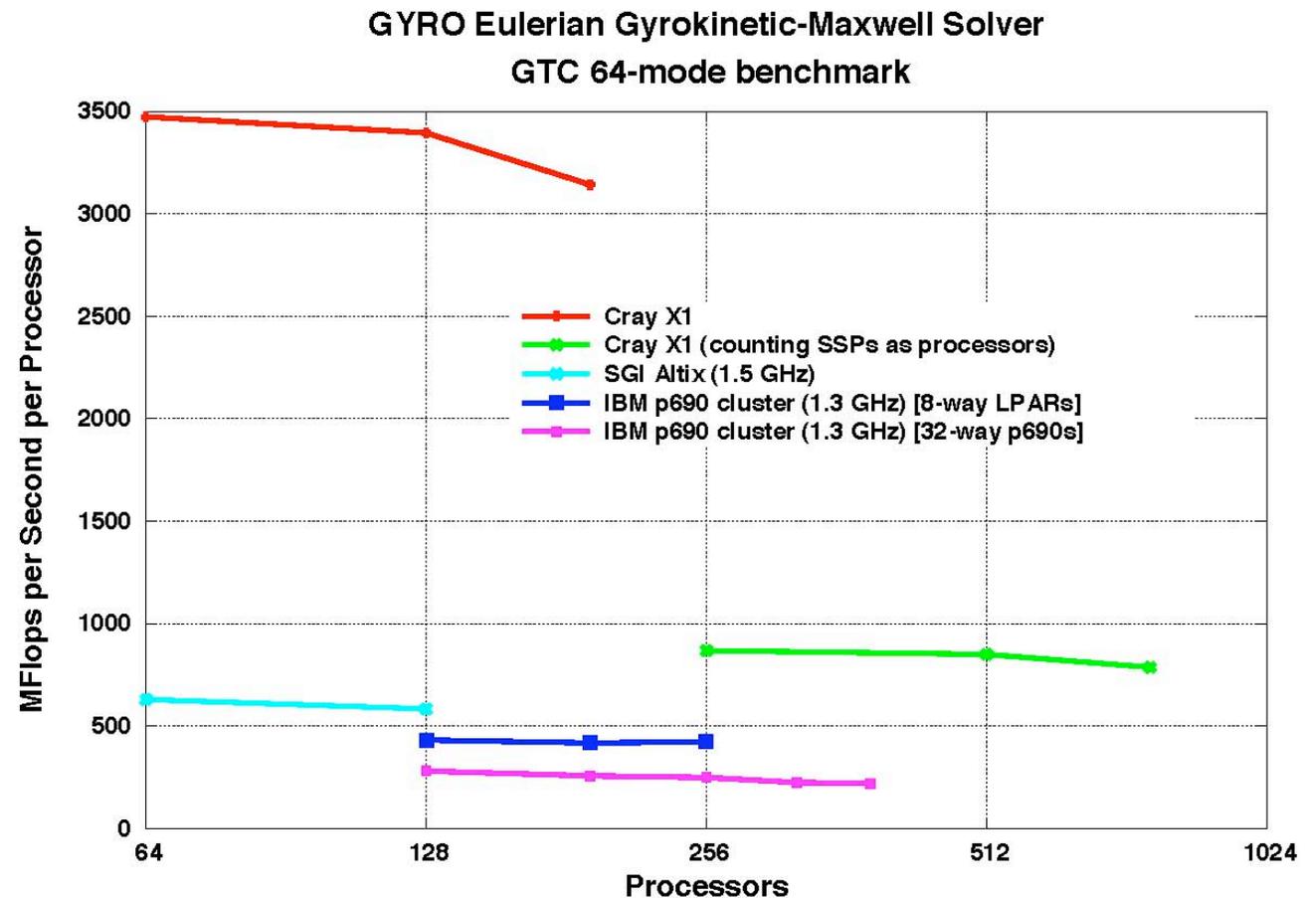
- X1 performance advantage is even more pronounced (factor of 8) for the larger problem size.

GYRO Eulerian Gyrokinetic-Maxwell Solver
GTC 64-mode benchmark



GYRO Computational Rate

- Nonscaling phase can use at most 64 processors, but is only 3% of execution time on X1 for 192 processors.
- All platforms show reasonable scaling, but IBM performance is still limited by bandwidth.



Application Summary

X1 may provide significant performance improvement relative to other platforms when

- Code vectorizes (or calls library routines that do), and
- Solving a sufficiently large problem.

The performance advantage increases when

- Performance is latency sensitive, and Co-Array Fortran or SHMEM can be used to implement the latency-sensitive algorithms, or
- Performance is bandwidth sensitive.

Conclusions?

- The Cray X1 works. Performance is continuing to improve, especially at scale, with updates to the OS and other system software.
- SHMEM and Co-Array Fortran performance can be superior to MPI. However, we hope that MPI small message and collective operator performance can be improved.
- 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.
- The X1 is a vector system, and there is no avoiding using vector-friendly code in order to achieve good performance.
- We need more experience with more application codes.

Questions ? Comments ?

For further information on these and other evaluation studies, visit

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