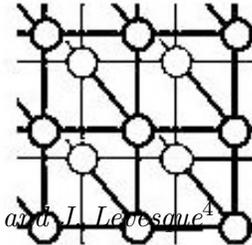


# Practical performance portability in the Parallel Ocean Program (POP)



*Concurrency and Computation: Practice and Experience*

P. W. Jones<sup>1,\*</sup>, P. H. Worley<sup>2</sup>, Y. Yoshida<sup>3</sup>, J. B. White III<sup>2</sup>, and J. Levesque<sup>4</sup>

<sup>1</sup> *Theoretical Division, Los Alamos National Laboratory, T-3, MS B216, Los Alamos, NM 87545-1663*

<sup>2</sup> *Computer Science and Mathematics Division, Oak Ridge National Laboratory P.O. Box 2008, Oak Ridge, TN 37831-6367*

<sup>3</sup> *Central Research Institute of Electric Power Industry, 1646 Abiko Abiko-shi Chiba, 270-1194, Japan*

<sup>4</sup> *Cray, Inc.*

## SUMMARY

The design of the Parallel Ocean Program (POP) is described with an emphasis on portability. Performance of POP is presented on a wide variety of computational architectures, including vector architectures and commodity clusters. Analysis of POP performance across machines is used to characterize performance and identify improvements while maintaining portability. A new design of the POP model, including a cache blocking and land point elimination scheme is described with some preliminary performance results.

KEY WORDS: ocean models; portability; performance; vector; scalar

## 1. Introduction

High-performance computing is an important tool for understanding ocean circulation and its role in the Earth's climate system. Accurate simulations of global ocean circulation require high spatial resolution to resolve energetic mesoscale eddies and to adequately represent topographic features [16, 8]. Simulations must also be integrated for long times in order to study century-scale climate change scenarios. Study of the thermohaline circulation, the deep ocean circulation driven by density differences due to heat and salt content, also requires multi-century integrations. This combination of fine spatial scales and long time scales requires very

---

\*Correspondence to: T-3, MS B216, Los Alamos National Laboratory, Los Alamos, NM 87545-1663



high-end computational resources and the ability to utilize new architectures as they become available.

The Parallel Ocean Program (POP) was developed at Los Alamos National Laboratory to take advantage of high-performance computer architectures. POP is used on a wide variety of computers for eddy-resolving simulations of the world oceans [16, 8] and for climate simulations as the ocean component of coupled climate models [1, 12]. In the next section, we will describe the POP model with particular emphasis on software design for performance portability. Later sections will describe the performance achieved on a variety of architectures and analysis of that performance. Descriptions and preliminary results of some recent performance-related improvements will then be described and conclusions presented.

## 2. POP Description

### 2.1. Model and Methods

POP is an ocean circulation model derived from earlier models of Bryan [2], Cox [4], Semtner [13] and Chervin [3] in which depth is used as the vertical coordinate. The model solves the three-dimensional primitive equations for fluid motions on the sphere under hydrostatic and Boussinesq approximations. Spatial derivatives are computed using finite-difference discretizations which are formulated to handle any generalized orthogonal grid on a sphere, including dipole [15] and tripole [10] grids which shift the North Pole singularity into land masses to avoid time step constraints due to grid convergence.

Time integration of the model is split into two parts. The three-dimensional vertically-varying (baroclinic) tendencies are integrated explicitly using a leapfrog scheme. The very fast vertically-uniform (barotropic) modes are integrated using an implicit free surface formulation in which a preconditioned conjugate gradient solver is used to solve for the two-dimensional surface pressure.

A wide variety of physical parameterizations and other features are available in the model and are described in detail in a reference manual distributed with the code. Because POP is a public code, many improvements to its physical parameterizations have resulted from external collaborations with other ocean modeling groups and such development is very much a community effort. Detailed descriptions of the numerical discretizations and methods are described in the reference manual and in previous publications [5, 6, 7].

### 2.2. Software

Although POP was originally developed for the Connection Machine, it was designed from the start for portability by isolating all routines involving communication into a small set (5) of modules which can be modified for specific architectures. Currently, versions of these routines exist for MPI [9] and SHMEM [14] communication libraries and also for serial execution. For the Cray X1, a Co-array Fortran [11] version was created. The appropriate directory is chosen at compile time and no pre-processor directives are used to support different machines. Support



for hybrid programming using threads and message passing has recently been added and will be described in a later section.

The original code was written in CM Fortran with extensive array syntax in a data-parallel programming model. Although it still retains much of the array syntax, some inefficient uses of array syntax have been replaced with more efficient loop constructs. POP has also evolved into a more traditional message-passing code using two-dimensional data decomposition of the horizontal domain with ghost cells to reduce communication. To maintain single-source portability with vector computers, the longer horizontal axes remain innermost. Early experiments with swapping indices to make the short vertical axis innermost for better cache performance did not show enough performance improvement to justify the additional code complexity and inhibited portability. Instead, a new domain decomposition scheme has recently been implemented to address cache issues and will be described in a later section.

The baroclinic portion of the code is the most computationally intensive, computing all of the three-dimensional tendency terms. To reduce memory use, most tendencies are computed on two-dimensional horizontal slices. The baroclinic computation has been designed so that only one ghost cell update is required and the calculation of baroclinic terms can proceed completely in parallel.

In contrast, the barotropic solver is a preconditioned conjugate gradient (PCG) algorithm that consists of a single application of a nine-point stencil operator followed by global reductions to perform the necessary inner products in the PCG method. Because it is only a two-dimensional mode, there are relatively few operations and the solver is dominated by many very small messages corresponding to inner product global sums. The solver is therefore very sensitive to message latency.

### 3. Performance

#### 3.1. Benchmark configurations

In order to assess the performance of POP across various machine architectures, two benchmark configurations were set up which accurately reflect two common production configurations.

The first configuration (called x1) is a relatively coarse resolution that is currently used in coupled climate models. The horizontal resolution is roughly one degree (320x384) and uses a displaced-pole grid with the pole of the grid shifted into Greenland and enhanced resolution in the equatorial regions. The vertical coordinate uses 40 vertical levels with a smaller grid spacing near the surface to better resolve the surface mixed layer. Because this configuration does not resolve eddies, it requires the use of computationally-intensive subgrid parameterizations. This configuration is set up to be identical to the actual production configuration of the Community Climate System Model [1] with the exception that the coupling to full atmosphere, ice and land models has been replaced by analytic surface forcing.

The second configuration (0.1) is nearly identical to current 0.1° global eddy-resolving production simulations. The horizontal grid is 3600x2400 and varies in spacing from 10km in equatorial regions to as low as 2.5km in the Arctic. The pole in this grid is displaced into the North American continent near Hudson Bay. The vertical grid again uses 40 levels with finer



spacing near the surface. Because the grid resolution is fine enough to resolve eddies, the most expensive parameterizations are not required; computational complexity comes entirely from the size of the simulation. As above, the benchmark is configured identically to the production simulations with the exception of using analytic forcing rather than the data-intensive daily surface forcing.

### 3.2. Results

Results for the x1 configuration were obtained on all the machines shown in Table 1. Cheetah, Eagle, and Phoenix are located in the Center for Computational Sciences at Oak Ridge National Laboratory. Lemieux is located at the Pittsburgh Supercomputer Center. Guyot is located in the Advanced Computing Laboratory at Los Alamos National Laboratory. Seaborg is located in the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory. The Earth Simulator is housed in the Earth Simulator Center in Yokohama, Japan.

The 0.1 configuration was too large to run on many of the machines, but results for the Earth Simulator and large IBM configurations were obtained. While most of the machines are very stable and mature systems, the Cray X1 system is very new and software continues to evolve rapidly. Software upgrades are expected to continue to improve POP performance in the near future.

The POP code used for all of these benchmarks is the 1.4.3 version. It was used without modification on all but the two vector machines. For vector machines, a few minor modifications were required. Two routines which perform a tridiagonal solve in the vertical had been optimized for cache-based machines by placing the loops over the short vertical index innermost with outer loops over the horizontal domain. For vector machines, some interchanging of loops was required to move one or both of the horizontal loops inside to enable vectorization over horizontal loops. These modifications affected less than 100 lines of code and we are investigating ways to incorporate these changes in a way that does not affect performance on cache-based microprocessors.

With the exception of the Cray X1, all of the simulations used MPI for the message passing between processors. As mentioned earlier, a Co-array Fortran form of the communication routines used in barotropic solver was used on the Cray X1, requiring only the change of a directory in the makefile.

Figure 1 shows POP performance on the x1 problem in simulated model years per wall-clock day, a preferred metric in the climate community that emphasizes production throughput. POP performs much better on either vector machine than on any of the other commodity microprocessor-based machines, indicating that POP performance is improved by the increased memory bandwidth available on vector machines. Event counters on SGI processors show that the ratio of load/stores to floating point operations is a little over two, providing additional evidence that memory bandwidth is important for POP performance. On the SGI, performance on 32 processors averages 104 Mflops/processor, approximately ten percent of peak performance. Such performance is similar on all machines built with commodity microprocessors and is also typical of many other scientific simulation codes on such machines.



Figure 2 shows the efficiency relative to performance on 4 processors:  $[T_4/PT_p]$  where  $T_p$  is the execution time when using P processors. (The IBM SP results are not shown as their inclusion makes it difficult to distinguish between the different curves.) Note the superlinear speed-up (efficiency greater than 1) on the SGI and HP platforms for moderate processor counts, indicating the improved performance due to better cache locality, and decreased demands on memory performance, as the per process problem granularity decreases. The cache-less Earth Simulator, the small cache Cray X1 and IBM p690 do not show this behavior. This effect disappears on the SGI and the HP systems as the communication costs and other parallel overheads become dominant.

Figures 3 and 4 illustrate the scaling of the baroclinic and barotropic parts of the model. The IBM SP results are again not shown in order to more easily understand the figures. In these figures, seconds per simulated day is used as the metric to emphasize scalability and because a throughput metric has little meaning when measuring portions of the code. As mentioned previously, the baroclinic part of POP contains very few communications and scales well on all machines, as shown in Fig. 3 where the curves are all nearly linear with similar slopes. The relative positions of the curves is due to differences in single-processor performance. The Earth Simulator shows some signs of slowing down on the x1 configuration at high processor counts because the subgrid size on each processor is becoming a relatively small multiple of the vector length. In contrast, the Cray X1 is able to maintain good vector performance for smaller vector lengths than the Earth Simulator.

Contrary to the baroclinic part, the barotropic is dominated by communication, particularly by global reductions. As mentioned previously, the communication consists of very small messages and global reductions and performance of the barotropic solver is dominated by message latency. Figure 4 shows that scaling on many machines for the x1 configuration is very poor above 16 processors where the subgrid has dropped to a size of 80x96 and there is not enough computational work to mask message latency. The exception to this is the SGI which has a very low-latency network in order to support the single-system-image shared memory. Because the barotropic solver is generally a small fraction of the simulation time, the lack of scaling doesn't begin to affect the total simulation time until using 64 or 128 processors.

The poor scaling of the barotropic solver prevents large numbers of processors from being utilized for coarse or moderate resolution problems. In most cases, production simulations of this size do not utilize such large processor counts due to other constraints like resource scheduling or scaling of other component models of the coupled system so the poor scaling has not been an impediment to such simulations. A search for alternative solvers continues, but alternative solvers have either had difficulties with irregular boundaries or have had slower convergence rates. Recent successes with an explicit subcycling of the barotropic mode may lead to use of a subcycling scheme in a future version of POP. Note that even a subcycling scheme will require communication at each subcycling step, but should avoid global reductions common in many iterative solvers.

The high resolution (0.1) benchmark configuration is ideally suited for the Earth Simulator, where the large problem size requires high memory bandwidth and high single-processor performance. POP simulations at this resolution on the Earth Simulator perform an order of magnitude faster than the IBM machine on similar numbers of processors as seen in Figure 5. Even at this resolution however, the poor scaling of the barotropic solver affects performance



at high processor counts, as can be seen in the relative efficiency curves in Figure 6. (Unlike the earlier figure, the Earth Simulator and p690 efficiencies are calculated relative to different baselines, 16 and 128 processors, respectively.) A direct comparison of these results with the x1 configuration is difficult because the configuration for the two cases are quite different. The 0.1 case uses a factor of ten smaller timestep and 70 times as many grid points, but the simpler parameterizations in the 0.1 configuration require a factor of 2.5 fewer flops per grid point. For an equivalent processor setup, this would mean the 0.1 case would take approximately 300 times longer to integrate a year-long simulation. The IBM results are roughly consistent with that estimate. The Earth Simulator is far more efficient at the larger problem size. At the smaller x1 problem size, the vector lengths are too short for maximum vector efficiency on the Earth Simulator. The Cray X1 architecture, with each processor consisting of four units of two vector pipes each, can perform efficiently with shorter vector lengths and performance on the smaller problem size competes favorably with the Earth Simulator.

#### 4. New decomposition scheme

In the previous section, POP is shown to perform well on vector machines. Performance on cache-based microprocessors is typical of many scientific codes, but is limited by memory bandwidth. In an attempt to achieve better performance on cache-based microprocessors, a recent new release of the POP code (2.0, not the version 1.4.3 used to obtain the results in the previous section) implements a new decomposition scheme. In version 2.0, the horizontal domain is still decomposed into Cartesian blocks. However, the block size can be adjusted based on machine architecture. For example, block size can be small to fit into cache or can be large on vector machines. Once the domain has been decomposed into blocks, blocks which consist only of land are dropped from the simulation, reducing the overall work load. The remaining blocks are then distributed across nodes using a static load-balancing scheme. The use of a Cartesian decomposition for the blocks rather than a more complicated partitioning scheme was to preserve the current structure of the code (many routines required only minimal changes) and minimize the burden on a large user and developer base. The block decomposition scheme described above provides a means of performing load balancing and land point elimination while retaining some back compatibility.

The new block decomposition scheme also provides a mechanism for a hybrid programming model using threads and message passing. If many blocks are assigned to each node, threading (eg OpenMP) can be used to assign blocks to threads within a node and message-passing is used between nodes. The threaded loop over blocks on a node is at a very high level in the code, ensuring a large amount of work within threaded loops to amortize overhead.

Finally, a different decomposition of blocks and different number of processors can be used for the barotropic solver. In particular, if the barotropic solver begins to slow down at high processor counts, a fewer number of processors can be assigned. While this does not completely eliminate the scaling problems, it can reduce the effects for those cases where the solver takes longer at high processor counts.

This new decomposition strategy combines cache-blocking, land point elimination, load balancing and hybrid parallelism to help improve performance. Initial results indicate that the



new decomposition can improve performance by 30% or more in cases like the 0.1 case above. The improvement is due both to improved use of cache and to land point elimination. At coarse resolution like the x1 case, there are few blocks which are completely land, so improvements were only seen at low processor counts where cache-blocking improved performance. At higher processor counts where the block size was already small, there was no improvement and in some cases a slight (but measurable) performance penalty. The new scheme has not been fully optimized and more work will be done to characterize and improve the new model.

## 5. Conclusion

POP is being used effectively on a wide variety of machine architectures using the same source code on both vector machines and machines built using commodity microprocessors. Performance on both classes of machines can be improved and code changes to implement these improvements are in progress. Further testing of POP 2.0 will be used to tune and optimize the new decomposition scheme and hybrid threaded/message-passing programming model. Performance modeling using POP will be used to guide design and development of future releases.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the entire team of POP developers, including R. Malone, R. Smith, M. Maltrud, M. Hecht, J. Dukowicz and J. Davis at Los Alamos National Laboratory (LANL). POP has also benefited from contributions by collaborators at the National Center for Atmospheric Research, including F. Bryan, G. Danabasoglu, P. Gent, W. Large and N. Norton. POP development is supported by the U.S. Department of Energy's (DOE) Climate Change Prediction Program and Scientific Discovery through Advanced Computing (SciDAC) program. LANL is operated by the University of California for DOE under contract W-7405-ENG-36. The work of J.B. White and P.H. Worley was sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the United States Department of Energy under Contract No. DE-AC05-00OR22725. The timing on the Earth Simulator was done with the support from the Ministry of Education, Culture, Sports, Science and Technology of the Japanese Government through the Project for Sustainable Coexistence of Human, Nature and the Earth. The timings on the Seaborg system at the National Energy Research Scientific Computing Center (NERSC) were collected by Dr. T. Mohan of NERSC. The computer time on PSC's Lemeiux was supported via NSF NPACI NRAC award 'A Framework for Performance Modeling and Prediction'

This paper have been authored by a contractor of the U.S. Government under contract Nos. W-7405-ENG-36 and 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.

## REFERENCES

1. Blackmon M, et al. The Community Climate System Model. *Bull. Am. Meteorol. Soc.* 2001; **82**:2357–76.



2. Bryan K. A numerical method for the study of the circulation of the world ocean. *J. Comput. Phys.* 1969; **4**:347.
3. Chervin RM, Semtner Jr AJ. An ocean modeling system for supercomputer architectures of the 1990s. In *Proc. of the NATO Advanced Research Workshop on Climate-Ocean Interaction*, Schlesinger M (ed.). Kluwer: Dordrecht, 1988.
4. Cox MD. A primitive equation, 3-dimensional model of the ocean. GFDL Ocean Group Technical Rept. No. 1, GFDL/NOAA: Princeton, 1984.
5. Dukowicz JK, Smith RD. Implicit free-surface method for the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.* 1994; **99**:7991–8014.
6. Dukowicz JK, Smith RD, Malone RC. A reformulation and implementation of the Bryan-Cox-Semtner ocean model on the Connection Machine. *J. Atmos. Ocean. Tech.* 1993; **10**:195–208.
7. Jones PW. The Los Alamos Parallel Ocean Program (POP) and coupled model on MPP and clustered SMP computers. In *Making its Mark: The Use of Parallel Processors in Meteorology*, Hoffman GR (ed.). World Scientific, 1997.
8. Maltrud ME, Smith RD, Semtner AJ, Malone RC. Global eddy resolving ocean simulations driven by 1985-1994 atmospheric winds. *J. Geophys. Res.* 1998; **103**:30825–30853.
9. MPI Standard Document.  
<http://www-unix.mcs.anl.gov/mpi/> [28 March 2003].
10. Murray RJ. Explicit generation of orthogonal grids for ocean models. *J. Comp. Phys.* 1996; **126**:251.
11. Numrich RW, Reid JK Co-array Fortran for parallel programming. Rutherford Appleton Laboratory Technical Report RAL-TR-1998-060 1998.
12. Randall DA, Ringler TD, Heikes RP, Jones PW, Baumgardner JR. Climate modeling with spherical geodesic grids. *Computing in Science and Eng.* 2002; **4**:32–41.
13. Semtner Jr AJ. Finite-difference formulation of a world ocean model. In *Advanced Physical Oceanographic Numerical Modeling*, O'Brien JJ (ed.). Reidel: Dordrecht, 1986.
14. SHMEM Chapter of Cray Manual.  
<http://www.cray.com/craydoc/manuals/004-2518-002/html-004-2518-002/z826920364dep.html> [28 March 2003].
15. Smith RD, Kortas S. Curvilinear coordinates for global ocean models. Los Alamos Unclassified Report LA-UR-95-1146, 1995.
16. Smith RD, Maltrud ME, Bryan FO, Hecht MW. Numerical simulation of the North Atlantic ocean at 1/10. *J. Phys. Oceanogr.* 2000; **30**:1532–61.
17. Smith RD, Dukowicz JK, Malone RC. Parallel ocean general circulation modeling. *Physica D* 1992; **60**:38–61.



Table I. Properties of machines used in benchmarks.

Machine Name	Machine Description	Proc	Proc Speed (MHz)	Cache (Mb)
Cheetah	IBM p690 cluster	Power4	1300	1.5 (L2)
Lemieux	HP AlphaServer SC	EV68	1000	8 (L2)
Guyot	SGI Origin3000	R14000	500	8 (L2)
Phoenix	Cray X1	Cray	800/400	2
ES	Earth Simulator	ES	1000/500	N/A
Eagle	IBM SP	Power3-II	375	8 (L2)
Seaborg	IBM SP	Power3-II	375	8 (L2)

Machine Name	Memory/Proc (Mb)	SMP Size	Switch/Network
Cheetah	1000	32	SP Switch2 (Corsair)
Lemieux	1000	4	Quadrics QsNet
Guyot	128	512	CrayLink
Phoenix	4000	4	Cray
ES	2000	8	ES
Eagle	500	4	SP Switch
Seaborg	1000	16	SP Switch2 (Colony)

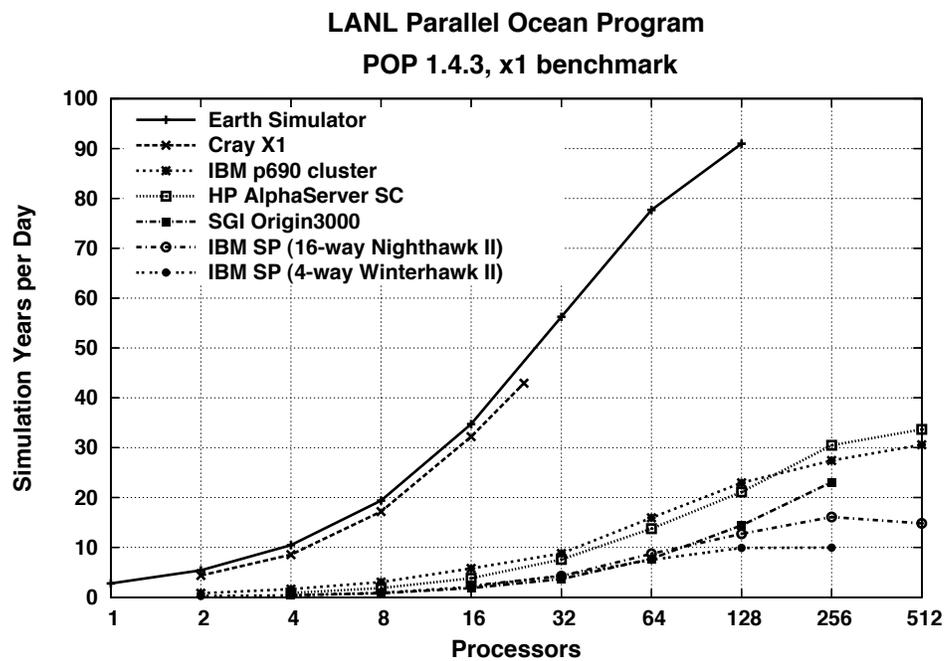


Figure 1. Performance in model years per CPU day as a function of processor count for the x1 configuration

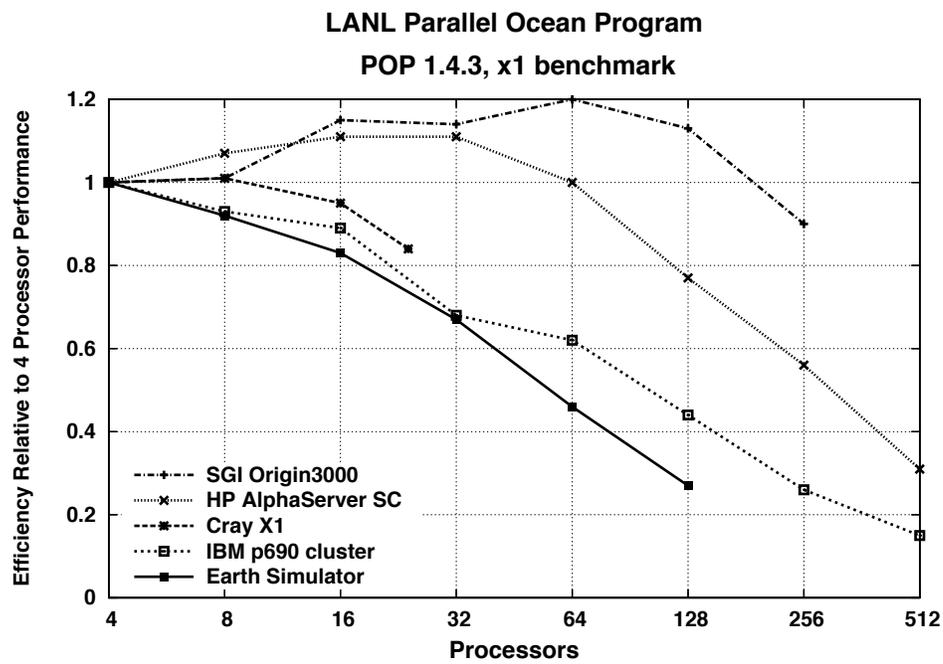


Figure 2. Parallel efficiency relative to four processors for the x1 configuration

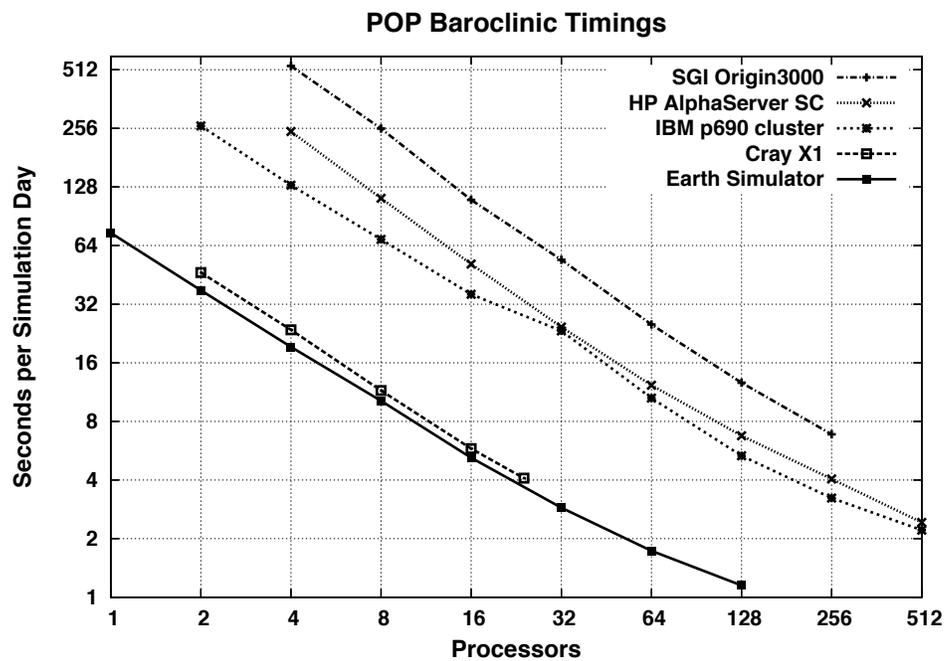


Figure 3. Performance in seconds per model day as a function of processor count for the baroclinic section in the x1 configuration

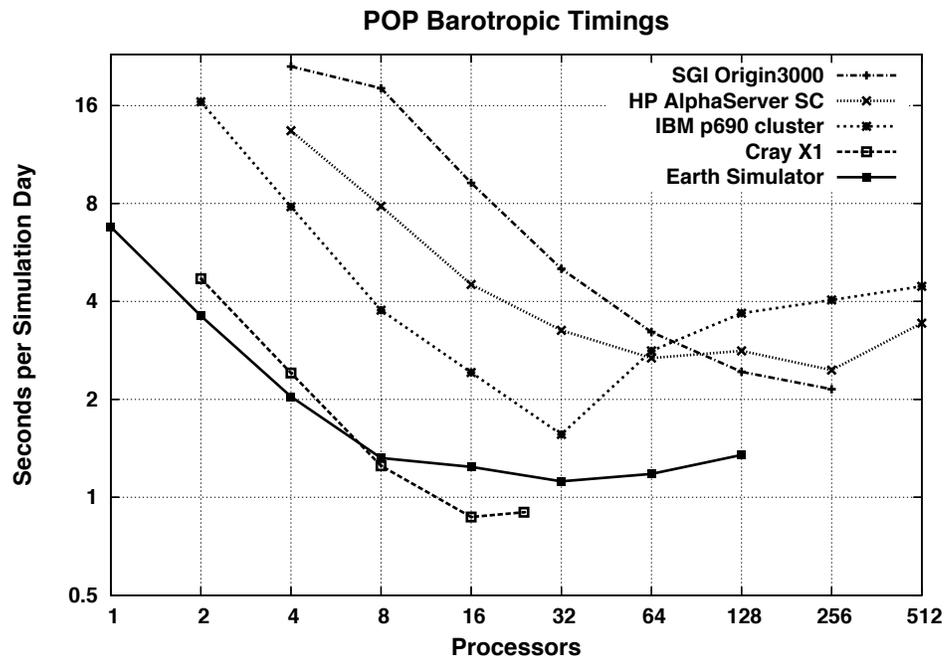


Figure 4. Performance in seconds per model day as a function of processor count for the barotropic solver in the x1 configuration

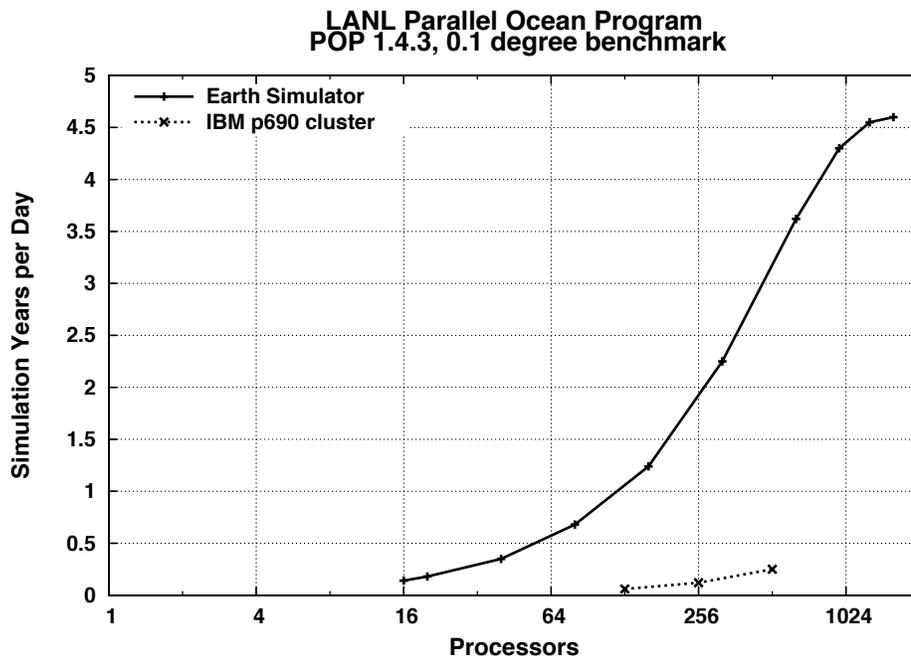


Figure 5. Performance in model years per CPU day as a function of processor count for the 0.1 configuration

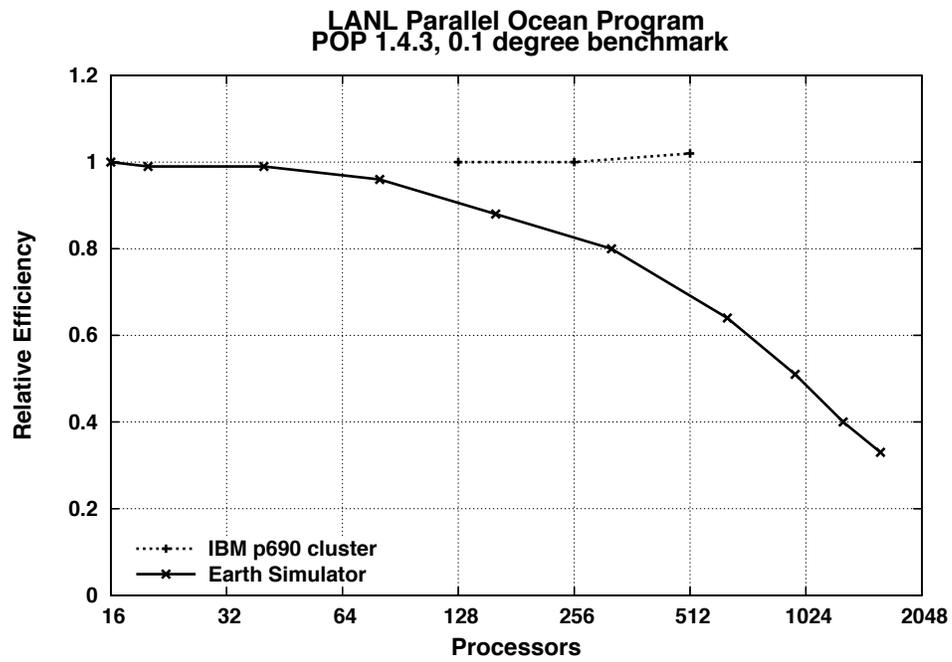


Figure 6. Parallel efficiency for the Earth Simulator relative to 16 processors and for the p690 relative to 128 processors in the 0.1 configuration