GA-A21159

OCEAN PREDICTABILITY STUDIES IN A PARALLEL COMPUTING ENVIRONMENT

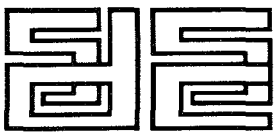
DOE Contract DE-FG03-91ER61217

FINAL REPORT

by

R. H. Leary

NOVEMBER 1992



SAN DIEGO SUPERCOMPUTER CENTER

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or others supporting the San Diego Supercomputer Center.

DISCLAIMER

GA-A21159

OCEAN PREDICTABILITY STUDIES IN A PARALLEL COMPUTING ENVIRONMENT

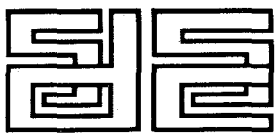
DOE Contract DE-FG03-91ER61217

FINAL REPORT

by

R. H. Leary

NOVEMBER 1992



SAN DIEGO SUPERCOMPUTER CENTER

Ocean Predictability Studies in a Parallel Computing Environment

DOE Contract DE-FG03-91ER61217
Final Report

by

R. H. Leary
San Diego Supercomputer Center

Project Staff:

R. H. Leary, PI (SDSC)
G. A. Hanyzewski (SDSC)
G. R. Montry (Consultant)

Abstract

The goal of the SDSC effort described here is to evaluate the performance potential of the Oberhuber isopycnal (OPYC) ocean global circulation model on the 64-node iPSC/860 parallel computer at SDSC and its near term successor, the Intel Paragon, relative to that of a single vector processor of a CRAY Y-MP and its near term successor, the CRAY C90. This effort is in support of a larger joint project with researchers at Scripps Institution of Oceanography to study the properties of long (10 to 100-year scale), computationally intensive integrations of ocean global circulation models to determine the character of natural variability in the simulations and their implications for ocean predictability.

Generally, performance of the OPYC model on the iPSC/860 has proved quite disappointing, with a simplified version of the entire model running at approximately 1.4 Mflops on a single i860 node in its original form and after extensive optimization, including coding in assembler, at 3.2 Mflops (i. e., 8% of the theoretical peak speed of 40 Mflops/node). We estimate overall performance to be limited to 75 Mflops or less on the 64-node machine, as compared to 180 Mflops for a single CRAY Y-MP processor and 500 Mflops for a single CRAY C90 processor. Similarly, we believe an implementation on an Intel Paragon, even with several hundred nodes, will not be competitive with a single processor C90 implementation. Reasons for this relatively low performance are related to architectural features of the i860 nodes and the specific computational demands of the algorithms used in the OPYC model.

Additionally, the Hamburg Large Scale Geostrophic (LSG) model has been implemented on high-performance workstations and evaluated for its computational potential as an alternative to OPYC for the long term integrations. Such integrations for this model appear to be feasible on this class of equipment, as does acceleration through implementation on a distributed network of high performance workstations.

1. Introduction

The basic behavior of ocean global circulation models (OGCMs) is not well studied. In the simplest case, their ability to reproduce the seasonal cycle in the upper ocean or the climatological features of the mean ocean circulation is not quantitatively well documented. Studies of the behavior of the models under extended integration are even fewer, even though such long integration may well yield remarkable insights into global climate change. One of the main impediments to a more complete diagnosis of OGCM behavior is the large amount of "traditional" computational resources they require. Recent advances in computer technology, particularly the advent of massively parallel computers with performance competitive with, and soon expected to exceed, that of traditional vector supercomputers, suggest this obstacle to ocean modeling may soon be overcome.

In support of a study by researchers at Scripps Institution of Oceanography (SIO) of ocean predictability in the 10 to 100-year time scale through long term OGCM integrations, the San Diego Supercomputer Center (SDSC) has undertaken the current project of evaluating the performance potential of the 64-node Intel iPSC/860 parallel supercomputer on a specific OGCM. The model chosen is the Oberhuber isopycnal model (OPYC) [1] developed by J. Oberhuber at the Max Planck Institute (MPI) for Meteorology in Hamburg.

This isopycnal coordinate model, which is characterized by the use of surfaces of constant density in the vertical coordinate, is relatively unknown compared to the more usual z-coordinate models. However, the model has several interesting features, including a fully active embedded mixed layer coupled to the interior ocean. This allows the treatment of turbulent entrainment and convective overturning on the penetration of greenhouse gases into the deep ocean, without the computationally costly large number of closely spaced levels in the upper ocean that would be necessary in a z-coordinate model. OPYC also includes a realistic equation of state

for sea water and (optionally) a sea-ice model coupled to the mixed layer. The resolution is variable in both horizontal dimensions and in the number of layers, and the time stepping is implicit, allowing one-day time steps. Encouraging results have been obtained with CRAY Y-MP and CRAY-2 implementations of a regionalized version of the model at SIO using observed winds to reproduce many observed features of El Nino - Southern Oscillation events.

The initial goal of the SDSC effort was to implement the OPYC on the 64-node iPSC/860 parallel computer at SDSC in sufficient detail to evaluate its performance on that platform (and its near term successor, the Intel Paragon) relative to the fastest available current generation vector supercomputers (e.g., the CRAY Y-MP and its successor, the CRAY C90) for long term (10 to 100-year) global integrations. Such integrations may take as much as several thousand single processor CRAY Y-MP hours for grid sizes of interest. If the performance evaluation on the iPSC/860 proved to be sufficiently encouraging, a full optimized implementation of the entire model on the iPSC/860 and the Paragon would be undertaken.

The iPSC/860 has a theoretical peak speed of 40 Mflops (millions of 64-bit floating point operations per second) for each of its i860 chip nodes or 2.6 Gflops for the full 64-node configuration, while the CRAY Y-MP has a single processor peak speed of 333 Mflops. Similarly, the i860 nodes on the Intel Paragon, the follow-on to the iPSC/860, are capable of a peak speed of 50 Mflops/node, thus bringing overall performance into the tens of Gflops for systems with several hundred nodes (SDSC anticipates the installation of such a Paragon system in December, 1992). In comparison, the CRAY C90 , the follow-on to the Y-MP, has a peak computational rate of 1 Gflop per processor . Thus large potential speedups of the iPSC/860 relative to the CRAY Y-MP, and of the Paragon relative to the C90 are possible in principle if a) good performance on individual nodes is obtained and b) effective parallelization is achieved.

2. Discussion

The target machine for evaluation of the OPYC model in a parallel computing environment is the 64-node iPSC/860 parallel supercomputer at SDSC. This system consists of 64 computing nodes interconnected with a communication network having a hypercube topology. Each node of the of the iPSC/860 has an i860 processor with a clock frequency of 40 MHz, a local memory of 8 Mbytes, and an 8 Kbyte cache. Under ideal conditions, the i860 can perform a 64-bit floating point adddition every clock cycle and a 64-bit floating point multiplication every other clock cycle, for a peak performance rate of 60 Mflops. However, the effective peak rate is generally regarded to be 40 Mflops, corresponding to the more realistic mix of only one addition per multiplication that is generally encountered in numerical kernels (e.g., vector dot product, DAXPY). Thus the peak performance of the SDSC iPSC/860 is 2.6 Gflops or approximately eight times the peak speed of a single processor of a CRAY Y-MP (333 Mflops). For a more complete description of the iPSC/860 and the i860 processor with particular emphasis on architectural features that affect attainable performance on real applications, see [2] .

As of this writing, the iPSC/860 is being superseded by the Intel Paragon, which consists of a large number (the largest system currently on order has 2000 nodes, as opposed to a maximum of 128 for the iPSC/860) of updated i860 processsors interconnected in a mesh topology. The new i860 processors have an increased clock speed of 50 MHz for a peak computational rate of 50 Mflops per node. Also, the memory access bandwidth, which typically is a bottleneck for real applications, has also been increased by a factor of 2.5, so i860 performance increases of a factor of 1.5 to 2.0 are anticipated for the Paragon relative to the iPSC/860. In comparison, the peak single processor performance of the most recent CRAY vector supercomputer, the C90, is 1 Gflop, or the equivalent of 20 Paragon nodes.

The OPYC model provided to us by J. Oberhuber consisted of a large number of subroutines totalling approximately 20,000 lines of Fortran optimized to run on vector supercomputers such as the CRAY Y-MP or more recently, the CRAY C90. Initial efforts for evaluating OPYC performance in the iPSC/860 parallel environment concentrated on implementing, evaluating, and optimizing performance of the most computationally intensive subroutines on individual i860 nodes and comparing results to those on the CRAY Y-MP. The base case for overall evaluation relative to the CRAY was taken to be the T42 x-y-z grid of dimension 130 x 62 x 9. The CRAY version of the model is well vectorized with long vectors and runs quite fast - approximately 180 Mflops on a single processor of a Y-MP and 500 Mflops on a single processor of a CRAY C90.

Although individual subroutines of the of the T42 grid OPYC model can be tested in isolation on single i860 nodes, the entire model is too large to run on one node. With the assistance of Oberhuber, who spent a week working with us in the early part of the project, a simplified "box" model with a flat bottom topography on a 22 x 22 x 3 grid, capable of being run on a single node, was constructed to aid in evaluating overall node performance. Also, single node self-contained versions of those components of the model associated with the solution of the 3-D wave equations and other computationally intensive subroutines on the full T42 grid were constructed.

Generally, performance on individual nodes proved quite disappointing. The box model ran at approximately 1.4 Mflops on a single node in its original form. After an extensive optimization effort, including the use of assembler coding where Fortran performance was notably poor, this was increased to 3.2 Mflops. While somewhat faster speeds may be obtained on the full grid, we believe the increment will be small. Similarly, the computationally most intensive subroutines, which consist of direct (Gaussian elimination) solvers for block tridiagonal systems arising from the 3-D wave equations and tridiagonal and 2 by 2 block tridiagonal systems arising from advection and diffusion equations, all ran at less than 3.5 Mflops when dimensioned for the full

grid.

Optimization efforts for the 3-D solver resulted in an overall speed increase to approximately 3.5 Mflops for the full grid. This block tridiagonal solver is the most computationally intensive module, directly accounting for approximately 30% of the overall model running time. Together with associated routines for setting up the equations and performing auxiliary control, diagnostic, and prognostic computations, the overall 3-D wave equation computation accounts for approximately 60% of the model running time. Note that the block size for these equations is equal to NZ, the number of Z grid points, and hence the full system bandwidth is 3*NZ. For the T42 grid, this bandwidth is thus 27.

Optimization efforts for the tridiagonal (bandwidth = 3) and 2 by 2 block tridiagonal (bandwidth = 6) systems association with advection and diffusion were less successful and were limited to about 1.5 Mflops at best in Fortran and 2.8 Mflops in assembler. This is due to the fact that the solution of such systems by Gaussian elimination contains an unavoidably high percentage of divisions (about 22% of the total operation count for the tridiagonal case), and division on the i860 is implemented in software and is extremely slow. The best assembler code for performing a single division uses a hardware low precision reciprocal operation followed by three Newton iterations and requires 38 clock periods for execution. Thus a loop executing only a sequence of divisions runs at about 1 Mflop for the best assembler coded implementation. This same loop in Fortran runs at 0.16 Mflop primarlily due to the overhead induced by a subroutine call to a library division routine generated by the current Intel PGI Fortran compiler, which does not use inlining.

In addition to the performance difficulties caused by the large number of divisions, the solution of banded linear systems with small bandwidths poses a second problem for the i860 architecture. For such systems, data localization is poor and there is little opportunity for holding data in floating point registers and cache for reuse to avoid

memory references. Thus there is a relatively high ratio of memory accesses to floating point computations. For the i860, low memory access bandwidth is a major performance bottleneck. At best only one 64-bit data item can be directly accessed from memory every two clock cycles, a rate which is adequate to support only a small fraction of the peak floating point computational rate. (Computational rates approaching peak performance can typically only be achieved in special circumstances with assembler coding that carefully manages memory accesses and the on-chip data cache.) In Fortran, the situation is much worse, since memory accesses must proceed indirectly through the cache, which effectively reduces the peak access rate by half.

Given the observed computational rates for individual nodes on the various components of the OPYC model, we believe the overall performance of an iPSC/860 parallel implementation, before overhead for parallelization and internode communication, will be limited to about 3 Mflops per node. The technical reasons for this relatively low performance are related to the internal architecture of the i860 chip, including the aforementioned limited memory access bandwidth and very poor floating point divide performance. The consequent performance penalties are particularly severe for Gaussian elimination applied to the types of narrowly banded linear systems which dominate the OPYC model.

Furthermore, while the tridiagonal and 2 x 2 block tridiagonal systems occur in sufficiently many parallel instantiations to allow effective parallelization over 64 (or even 128) processors for the T42 grid, the block tridiagonal systems that arise from the wave equations do not. Here the most straightforward strategy is a one dimensional parallelization over the y-grid, with the parallel linear systems that arise representing equations in the x-z planes. The number of such parallel block tridiagonal systems is equal to half the number of y grid points, or 31 in the T42 case (an iterative scheme is used in which parallel systems corresponding to the odd and even y-grid points are alternately solved). Thus on a 64 node system, approximately half the processors

would be idle at any given time when solving the wave equations. Unfortunately, this is the dominant portion of the entire model calculation. Note that this lack of scalability of the y-coordinate parallelization is a generic problem for all parallel processors and is not specific to the iPSC/860.

When parallelization and internode communications overhead is also considered, we expect that the overall performance of the 64-node iPSC/860 will be less than 75 Mflops on the OPYC model, and that a larger system with more computational nodes will not help appreciably. Thus we have concluded that further efforts to implement the entire model in its current form in a fully parallel optimized version on the iPSC/860 using the y-coordinate parallelization are not warranted.

Increased node performance will, of course, increase total system performance. The next generation Intel machine, the Paragon, is expected to have a factor of as much as 2 improvement in typical node performance. A parallel implementation on the Paragon thus will still be considerably slower than a single CRAY C90 processor for the T42 grid. The Paragon also has an increased internode communications performance of a factor of about 80 relative to the iPSC/860, and thus a much lower communications overhead and improved applicability to finer grained problems. Potentially with a major algorithmic reformulation of the model to allow for a finer grained parallelization decomposition over x-y regions, the Paragon could become competitive with a single C90 processor, and perhaps outperform it on higher resolution grids.

Thus for the immediate future, we believe the OPYC model, in its current algorithmic form, will be far better suited to state-of-the-art vector supercomputers than massively parallel distributed memory computers. A major algorithmic reformulation of the model to allow a two-dimensional parallelization of the 3-D solver for better scaling to larger parallel machines appears to be necessary before a parallel implementation of OPYC appears warranted. As suggested by L. Margolin [3], such a reformulation could also

gain considerable effciency by introducing a baroclinic/barotropic decomposition such as that used in the Semtner/Chervin [4] OGCM (and several other OGCMs) and is currently under study at Los Alamos National Laboratory. With such a decomposition, the implicit part of the code can be restricted to solving a 2-D Poisson equation, a problem for which very efficient two-dimensional parallelization techniques are well known.

Based on the projected performance of OPYC in its current form on the iPSC/860, we have also investigated an alternative model. The Hamburg Large Scale Geostrophic Ocean General Circulation Model (LSG) [5] has been obtained from the Max Planck Institute for Meteorology and with the help of M. Lautenschlager from MPI, the original CONVEX version has been ported to the CRAY Y-MP. The model physics are considerably simpler than those of the OPYC model, and consequently running times for a given simulation period are much faster. Performance on a single processor of the CRAY Y-MP is excellent . In particular, an initial large matrix factorization for a 72 x 76 x 11 grid , which need only be performed once, takes about 20 seconds and runs at 60 Mflops. For the remainder of the computation, which runs at 150 Mflops, the basic time step of 30 days takes approximately 0.75 seconds, allowing the completion of a 100-year integration in less than 15 CPU minutes and a 1000-year integration in less than 3 CPU hours. The code is very memory intensive (approximately 48 Mbytes for this grid on the CRAY), as the matrix factors of a large system of linear equations are saved and used at each time step.

It appears to be feasible to implement the code on current medium and high performance workstations and obtain reasonable performance (e.g. 10-year integrations within several wall-clock hours). We feel such an implementation will be of considerable interest within the climate modelling community, as it would provide wide access to an OGCM that would run on almost universally available computational equipment. Consequently we have made such a port to Silicon Graphics (model 4D/440 VGX), SUN (model 370-TAAC) and DEC (model 5000/200) workstations. In

all cases a single time step simulating 30 days required less than one minute of CPU time, and results were obtained that were essentially identical to the CRAY and CONVEX versions. With the current 72 x 76 x 11 grid, the code requires about 45 MBytes of total (physical + virtual) memory and runs efficiently with as little as 32 Mbytes of physical memory.

However, very long simulations, or simulations with a higher grid resolution, will strain both the computational and storage capabilities of current workstations. While new generation workstations such as the DEC Alpha, which is expected to be an order of magnitude faster than a DEC 5000/200, will provide some relief, another complementary approach is to harness the parallel computational power and memory storage of multiple workstations in a loosely coupled network. Public domain software which provides such a framework for the development and execution on (possibly) heterogeneous computers of large coarse grained parallel applications exists in the form of PVM [6]. PVM has been successfully implemented at SDSC and applications have been run using both homogeneous networks of identical workstations as well as heterogeneous networks involving scalar workstations, large vector computers (CRAY Y-MP processors), and parallel computers (iPSC/860). No special hardware is necessary for using PVM and it can be easlily implemented at most sites with multiple workstations connected over existing networks.

We have profiled the LSG code extensively and investigated its parallelization potential with respect to a network of loosely coupled workstations. All of the computations appear to be readily adaptable to such a framework using coarse grain parallel executables, with the exception of the initial matrix factorization and subsequent triangular system backsolves involved in the computation of the barotropic wave solution. The matrix factorization is only done once, but the backsolve is done once per time step and accounts for approximately 5% of the total computational effort. Hence over many time steps, approximately 95% of the code should be parallelizable within the PVM framework.

3. Conclusions

Based on the work reported here, we conclude:

a) The maximum performance of the 64-node iPSC/860 for the OPYC model, using the simplest y-coordinate parallelization strategy, is likely to be less than 75 Mflops on the T42 grid, which compares unfavorably with the observed speeds of 180 Mflops and 500 Mflops for single processor CRAY Y-MP and C90 implementations, respectively. The primary reasons for the relatively poor performance of the iPSC/860 are a) the low memory access bandwidth within the i860 nodes, and b) the poor i860 floating point divide performance. The most computationally intensive routines in the OPYC model are all direct banded linear equation solvers applied to narrowly banded systems. These algorithms are particularly demanding of memory bandwidth and use a relatively large number of floating point divisions. Thus they are especially sensitive to the above mentioned performance limitations of the i860.

b) The performance of the i860 nodes on the Paragon is expected to be at most a factor of 2 faster than those on the iPSC/860. With the y-coordinate parallelization strategy limiting the number of useful computational nodes, even a large Paragon is expected to significantly underperform a single processor of a CRAY C90 on the T42 grid. This expected performance is insufficient to warrant a full implementation of the OPYC model in its current algorithmic form on the iPSC/860 or the Paragon. However, with a finer grid, or a parallelization strategy based on decomposing the x-y plane, a Paragon with several hundred nodes may become competitive.

c) Although we have not investigated whether its simplified physical formulation is adequate for the predictability studies of interest, the LSG model offers a computational feasible alternative to OPYC. Integrations of 100 years require less than an hour of single processor CRAY Y-MP CPU time for modestly sized grids, while

a similar computation can be performed on current generation DEC, SUN and SGI scientific workstations in less than a day. Also, the LSG model is amenable to computation on distributed networks of workstations using software such as PVM. This will allow the treatment of higher resolution grids and longer integrations, as well as offer wider accessibility to OGCM computing to researchers without direct access to vector or parallel supercomputers.

4. Acknowledgment

This work was supported by the DOE Office of Energy Research under contract DE-FG03-91ER61217. Computational facilities were provided by the San Diego Supercomputer Center.

5. References

1. J. M. Oberhuber, "Simulation of the Atlantic Circulation with a Coupled Sea - Ice - Mixed Layer - Isopycnal General Circulation Model," Max Planck Institute for Meteorology, Hamburg, Report No. 59, November, 1990.

2. R. Berrendorf and J. Helin, "Evaluating the Basic Performance of the Intel iPSC/860 Parallel Computer," *Concurrency: Practice and Experience*, **4** (May, 1992), pp.223-240.

3. L. Margolin, Los Alamos National Laboratory, pers. comm.

4. A. Semtner and R. Chervin, "A Simulation of the Global Ocean Circulation with Resolved Eddies, *J. Geophys. Res.*, **93** (C12, 1988), pp.15,502-15,522.

5. E. Maier-Reimer and U. Mikolajewicz, "The Hamburg Large Scale Geostrophic General Circulation Model (Cycle 1), Max Planck Institute for Meteorology, Hamburg,

Deutches KlimaRechenZentrum Report No. 2, October, 1991.

6. A. Beguelin, J. Dongarra, A. Geist, R. Manchek, and V. Sunderam, *A Users' Guide to PVM Parallel Virtual Machine*, Univeristy of Tennessee Technical Report CS-91-136, July 1991.