Development of a High Resolution Parallel Ocean Circulation Model on the Earth Simulator

Yukio Tanaka*, Motohiko Tsugawa and Yuichi Mimura

Frontier Research System for Global Change (FRSGC)
3173-25 Showa-machi, Kanazawa-ku, Yokohama, Kanagawa236-0001, Japan

Key words: Earth Global Simulation, Ocean

1. Introduction

The recent high performance scientific computers, which employ parallel architecture, are increased their performance rapidly. The Earth Simulator, which is operated from the April 2002, is a vector parallel computer. The 640 processor nodes are connected by a high speed single stage crossbar network and each node consists of eight vector processors connected with symmetric multiprocessor architecture. The peak computation performance is 40 Tflops and total memory size is 10TB. As the scientific computer performance is increased, the resolution of the ocean circulation calculation is increased. In the early 1990s, the typical resolution to calculate the world ocean circulation was about a few degrees (e.g., [1]). Recently, the 0.1-degree ocean calculation is reported [2].

In these circumstances, we have started to develop new ocean circulation models which have high computational performance of calculating high resolution ocean general circulation. As a first step of this development, a longitude-latitude coordinate ocean circulation model optimized for the Earth Simulator has been developed and its computational performance is measured. The development of the cubic grid ocean general circulation model has also started as the next generation ocean model, the time step width of which is expected to be larger than the longitude-latitude grid system [3].

2. Basic equations

The basic equations are the incompressible, Boussinesq form of the Navier-Stokes equations with hydrostatic approximation, i.e., the primitive equations [4]. These equations consist of four three-dimensional nonlinear partial differential equations with dependent variables of horizontal velocity, u and v, the potential temperature T, the salinity S. The independent variables are latitude λ, longitude ,ϕ and depth z. Arakawa-B grid method [5] is used for the horizontal discretization and Leap-frog is used for time integration.

These basic equations contain fast moving external gravity waves and slow moving internal gravity waves. In order to save computing time, the vertically integrated equations (barotoropic equations), which contains the fast moving external gravity waves, are separated from the vertical structure equations (baroclinic equations), which contains only the slow moving internal gravity waves. Since barotopic equations are two-dimensional equations, these can be solved with consuming small amount of computation time. The three-dimensional baroclinic equations are solved using relatively large time step. Consequently, overall computation time can be saved. This technique is known as the mode splitting method [6].

3. Parallelization and vectorization

The two-dimensional domain decomposition is employed for horizontal direction. No parallel decomposition is done for the vertical direction because the hydrostatic pressure calculation, the vertical implicit viscosity and diffusion calculation are sequential one in this direction.

* Corresponding author. E-mail: ytanaka@jamstec.go.jp

The parallelization is done using MPI library. The Earth Simulator consists of 640 nodes and each node consists of eight vector processors. For this computer architecture, using OpenMP in the node and MPI library between nodes would be suitable. However, there is no clear evidence that this hetero-structure parallelization is better thanthe parallelization done by using only MPI library. Therefore, only the MPI library is used for the parallelization.

It is important to use the vector processor efficiently to achieve high computational performance because the node of the Earth Simulator consists of the vector processors. Since the vector register length of the Earth Simulator is 256, employing loop length that is a little less than 256 is effective to obtain high vector processor usage. In addition, reducing memory access under consideration of the memory bandwidth is also effective to increase the vector computation efficiency. The peak computation speed of the double precision floating point calculations and the memory access bandwidth of the Earth Simulator are 8 Gflops and 32 GB/sec, respectively. Therefore, more than one double precession floating point memory access per two double precession floating point computations may degrade the vector computation performance.

It is possible to make longer loops by combining the longitude loops with the latitude ones. By using these long loops, the vector processor efficiency would be increased. In this method, the physical quantities are stored in one-dimensional arrays. The metrics of the spherical coordinates are also stored in one-dimensional arrays. Since the metrics are function of the only latitude coordinate, the one-dimensional arrays of the metrics store the same values in different addresses. These metrics are accessed from different addresses during calculations and it increases unnecessary memory access. This memory access increment may degrade the vector processor efficiency. Therefore, this loop combining method is not used for this program and the vectorization is done only for the longitude-direction.

4. The barotropic equation solver

In the barotropic equation solver, the ratio of the data communication time to the total elapse time is much larger than that of the baroclinic equation solver [7]. When the data size becomes small, communication time of parallel computers is determined by the latency time which does not depend on the size of the data and is not decreased when the size of the data is decreased. In the case of the Earth Simulator, the communication time is determined by the latency time when the data size becomes less than about 512KB [8]. For the 0.1°x0.1° horizontal resolution problem using 3840 processors, the communication data sizes of the barotropic solver for one time-step are about 6 KB for the latitude direction and 200 B for the longitude direction. For these data sizes, the communication time is determined by the latency time and does not decrease as increasing the processor numbers.

In order to reduce the communication time, the data communication frequency should be reduced. The communication frequency can be reduced by sending several boundary values of each domain to the neighbor's overlap region and the time evolution of the physical values for several time steps are computed in the overlap region without the data communication. According to reference [9], Woodward was one of the first to apply this technique [10]. Table 1 shows the elapse time of the barotropic solver which uses this technique. When the overlap grid number are increased, the total elapse time, which is the sum of the computation and communication time, is successfully reduced.

Table 1. The computation performance of the barotropic solver for several overlapped grid numbers.

overlap grid number	1	2	3	4
elapse time (msec)	8.983	5.697	4.585	4.215
communication time (msec)	7.221	3.921	2.736	2.226
computation time (msec)	1.762	1.776	1.849	1.989
speed (Gflops)	0.496	0.934	1.347	1.677

5. Computation performance

The computational performance of 0.1°x0.1° horizontal resolution and 42 levels problem is measured by using 960, 1920, and 2840 processors. The table 2 shows the performance of the top five subroutines using four overlap grids described previous section. The subroutine 'momentum' and 'tracer' solve the momentum and tracer equations, respectively. The data communications for these equations are done in the subroutine 'exchange data'. The 'advection' subroutine calculates the advection velocities used in the both the momentum and tracer equations. The subroutine 'equation of state' calculates the seawater density from the temperature, salinity, and pressure. The elapse time of 'momentum', 'tracer', 'advection', and 'equation of state' are decreased linearly as the number of the processors is increased. The computation speed of these equations are more than 50 % of the peak computational speed of the Earth Simulator's vector processor unit, whose peak performance is 8 Gflops. The speed of the 'equation of state' reaches around 85 % because this subroutine accesses less memory than the others. The subroutine 'free surface' solves the barotropic equations by using the technique described in the previous section, the timing of which includes the communication time. Although the timing of this subroutine decreases as the number of processors is increased, the decreasing rate is slower than the other subroutines.

Table 2. Computational performance of 0.1°x0.1° horizontal resolution and 42 levels problem.

Processors	960	(16x12)	1920	(16x240)	3840	(16x240)
subroutine name	Gflops	msec	Gflops	msec	Gflops	msec
free surface	2.770	8.095	2.214	6.308	1.677	4.215
momentum	4.222	19.972	4.185	10.387	4.083	5.330
tracer	4.671	14.560	4.598	7.625	4.540	3.867
advection	4.173	3.203	4.000	1.749	3.905	0.898
equation of state	6.855	2.420	6.851	1.243	6.790	0.628
exchange data	0.000	2.756	0.000	2.711	0.000	1.566
TOTAL (Tflops, msec)	3.964	55.814	7.421	32.668	13.884	17.820

Figure 1 shows the ocean sea surface temperature after 1000 year simulation. The initial conditions are constant temperature and salinity. The observed wind stress distribution is given on the sea surface. The resolution of this run is about 0.5x0.5 in horizontal and 42 levels.

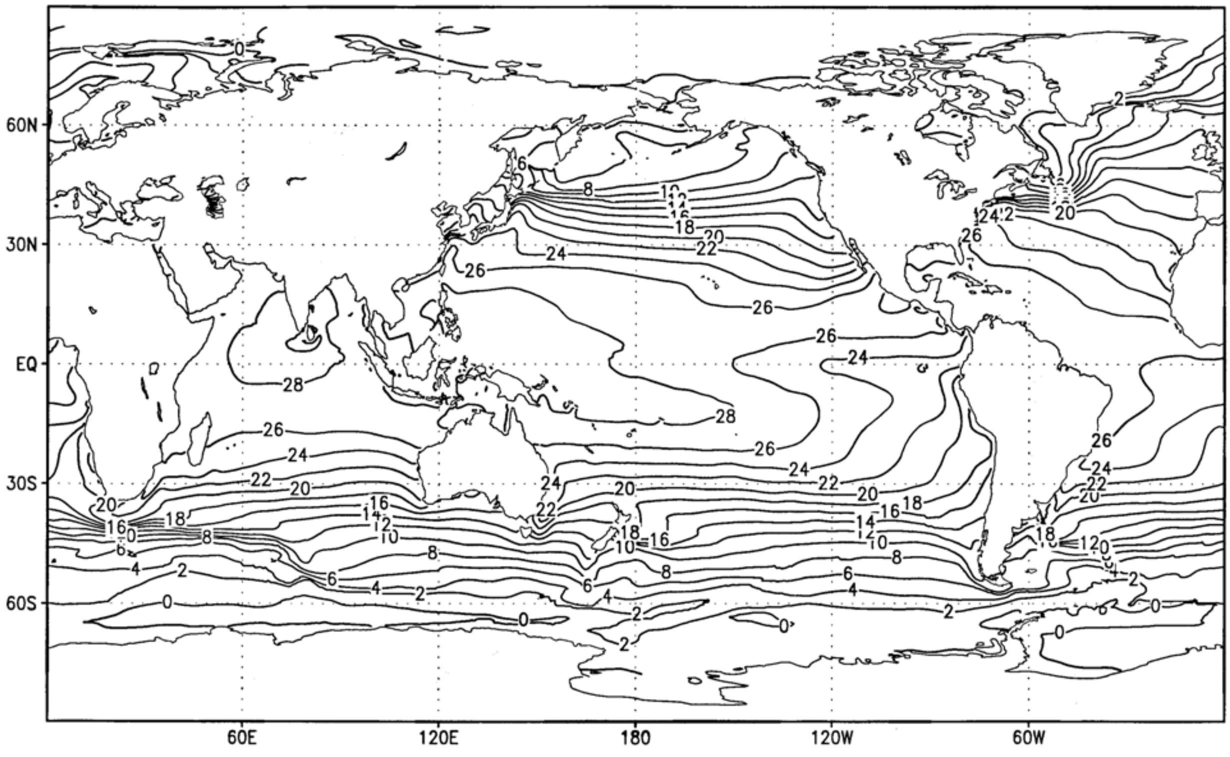

Figure 1. Simulated ocean surface temperature.

References

1. England, M. H., Representing the global-scale water masses in ocean general circulation models. J. Phys. Oceanogr., 23 (1993) 1523-1552.
2. Smith, R. D. , Maltrud, M. E. , Bryan, F. O., and Hecht, M. W., Numerical Simulation of the North Atlantic Ocean at 1/10°, J. Phys. Oceanogr., 30 (2000) 1532-1561.
3. Tsugawa, M., Tanaka, Y., Yoon, S. Y., 2002 Workshop on the Solution of Partial Differential Equations on the Sphere, Development of a Global Ocean Model on Quasi-Homogeneous Cubic Grid., (Aug. 2002).
4. Bryan, K., A numerical method for the study of the circulation of the World Ocean. J. Comp. Phys. 3 (1969) 347-376.
5. Arakawa, A., Computational design for long-term numerical integration of the equations of fluid motions, J. Comp. Physics. 1 (1966) 119-143.
6. Simons, T.J., Verification of numerical models of Lake Ontario. Part I, circulation in spring and early summer, J. Phys. Oceanogr., 4, (1974) 507-523.
7. Tanaka, Y., Yoon, S. Y., Tsugawa, M., Design and Performance Analysis of an Ocean General Circulation Model optimized for the Earth Simulator, Parallel CFD 2002, (2002).
8. Yoshida, K., Shigu, S., Research and development of the Earth Simulator, Proceedings of 9th ECMWF workshop, World Scientific, (2000) 1-13.
9. Oberpriller, W. D., Sawdey, A., O'Keefe, M. T., Gao, S., Parallelizing the Princeton Ocean Model Using TOPAZ, (1998) 331-352.
10. Colella, P., Woodward, P. R., The piecewise parabolic mehod (PPM) for gas-dynamical sumulation, J. Comp. Physics. 54 (1984), 174-201.