Parallel Implementation of the INM Atmospheric General Circulation Model on Distributed Memory Multiprocessors

Val Gloukhov

Moscow State University,
Institute for Numerical Mathematics of the Russian Academy of Sciences,
gluhoff@inm.ras.ru,
WWW home page: http://www.inm.ras.ru/~gluhoff

Abstract. The paper is devoted to a distributed memory parallel implementation of a finite difference Eulerian global atmospheric model utilizing semi-implicit time stepping algorithm. The applied two-dimensional checkerboard partitioning of data in horizontal plane necessitates boundary exchanges between the neighboring processors all over the model code and multiple transpositions in the Helmholtz equation solver. Nevertheless, quite reasonable performance has been attained on a set of cluster multiprocessors.

1 Introduction

One of the most challenging prospects requiring advances of supercomputers power is acknowledged to be the Earth climate system modeling, in particular, those studies concerning the global warming issues and related climatic changes from increasing concentration of greenhouse gases in the atmosphere. The main predictive tools here are the general circulation models run for decadal and even centennial simulation periods.

INM Atmospheric General Circulation model (AGCM) was designed at the Institute for Numerical Mathematics of the Russian Academy of Sciences and originates from earlier works of G.I. Marchuk et al. [1]. The model participated in AMIP II intercomparison project as "DNM AGCM" [2] and some other experiments [3], [4]. Though most of simulations performed with the INM AGCM are based on the Monte Carlo method, allowing different trajectories to be integrated independently, the main interest of the present paper lays in the intrinsic parallelism of the model.

Mostly, the model provides substantial degree of parallelism, except the implicit part of the time-stepping scheme that solves Helmholtz equation on a sphere and spatial filters damping the fast harmonics of prognostic fields at the poles. In opposite to Barros and Kauranne [5], we parallelized a direct Helmholtz equation solver [7] involving the longitudinal fast Fourier transforms (FFTs) and latitudinal Gaussian elimination for tri-diagonal linear systems.

In the next section, we will outline the model structure. Then, in Sect. 3, we will proceed with details of parallelization technique. Performance of the obtained parallel version of the model is presented in Sect. 4. We have carried out benchmarking on MBC1000M computing system, located at the Joint Supercomputer Center, and SCI-cluster, maintained by the Research Computing Center of the Moscow State University.

2 Model structure

INM AGCM solves the system of partial differential equations of hydro-thermodynamics of the atmosphere under hydrostatic approximation on the rotating sphere [2]. The vertical coordinate is generally either pressure or a terrain-following sigma ($\sigma = p/\pi$, where π is the surface pressure) or a hybrid of the two. The equations are discretized on a staggered Arakawa "C" grid [6], meaning all prognostic variables are co-located, except zonal and meridional wind components that are staggered in the longitudinal and latitudinal directions, respectively. The grid has 2° resolution in latitude, 2.5° in longitude, and $K = 21$ irregularly spaced vertical levels. The size of the computational domain is thus $145 \times 73 \times 21$. All floating point data are kept in single precision (32-bit) representation.

Let u and v represent zonal and meridional wind components, correspondingly, T is the temperature, d is the horizontal divergence; $P = \Phi + RT_0 \log \pi$, Φ is the geopotential, R is the gas constant, $T_0 = 300K$. Then the model governing equations after discretization can be rewritten as

$$\begin{aligned}
\delta_t \overline{\mathbf{u}}^t + \frac{1}{a\cos\varphi}\delta_\lambda\left(\frac{1}{2}\delta_{tt}\mathbf{P}\right) &= \mathbf{A}_u, \\
\delta_t \overline{\mathbf{v}}^t + \frac{1}{a}\delta_\varphi\left(\frac{1}{2}\delta_{tt}\mathbf{P}\right) &= \mathbf{A}_v, \\
\delta_t \overline{\mathbf{T}}^t + \mathbf{B}\left(\frac{1}{2}\delta_{tt}\mathbf{d}\right) &= \mathbf{A}_T, \\
\delta_t \overline{\log\pi}^t + \nu^T\left(\frac{1}{2}\delta_{tt}\mathbf{d}\right) &= A_\pi,
\end{aligned} \tag{1}$$

where t stands for time, λ is longitude, φ is latitude, δ_t, δ_λ and δ_φ are the discrete analogues of the corresponding differential operators of the form

$$\delta_x \psi(x) = \left(\psi\left(x + \frac{\Delta x}{2}\right) - \psi\left(x - \frac{\Delta x}{2}\right)\right) \Big/ \Delta x,$$

$\delta_{tt}\psi(t) = \psi(t-\Delta t) - 2\psi(t) + \psi(t+\Delta t)$, $\delta_t \overline{\psi}^t(t) = (\psi(t+\Delta t) - \psi(t-\Delta t))/(2\Delta t)$, $\mathbf{A}_u$, $\mathbf{A}_v$, $\mathbf{A}_T$, and A_π represent the explicit dynamical tendencies which comprise both the model forcing and discretized spatial operator of the model governing

Table 1. INM AGCM basic routines and percentage of their CPU time measured on MBC1000M computing system and SCI-cluster in uniprocessor mode

Routine	Purpose	MBC	SCI
Physics:			
FASFL	Convection processes	1%	1%
RADFL	Radiative processes	33%	37%
DSP	Gravity-wave drag	8%	8%
PBL	Atmospheric boundary layer	4%	4%
Dynamics:			
ADDTEN	Add physical tendencies to the solution	1%	1%
VDIFF	Vertical diffusion	13%	16%
RHSHLM	Explicit dynamical tendencies generation	18%	10%
HHSOLV	Helmholtz equation solver	4%	4%
DYNADD	Add dynamical tendencies to the solution	5%	4%
VISAN4M	Horizontal diffusion	6%	10%
RENEW	Spatial and temporal filtration	6%	4%
Other routines		0.1%	0.1%

equations, a is the mean radius of the earth, $\mathbf{B}$ is a matrix, ν is a vector. We emphasize matrices of size $K \times K$ and vectors of length K with bold letters.

The model parameterizes long and short wave radiations, deep and shallow convection, vertical and horizontal diffusion, large scale condensation, planetary boundary layer, and gravity wave drag.

The fraction of time spent in various routines of INM AGCM is shown in Table 1. Physical components are computed hourly, except the radiative block (RADFL) involved every 3 model hours; dynamical block, for the grid resolution specified above, is calculated 9 times within a model hour.

3 Parallelization technique

We apply a 2-dimensional domain decomposition to the globe in both the longitude and latitude dimensions, or the so called checkerboard partitioning [8]. The resulting rectangular subdomains are assigned to the processors arranged into a $P_{lon} \times P_{lat}$ virtual grid. We will refer further this distribution of data as the basic one.

3.1 Physics

The basic distribution is highly suitable for INM AGCM's physical parameterizations that are column oriented and thus can be done concurrently by columns. Although communications are required to interpolate quantities between the staggered and non-staggered grids and some load imbalance occurs in radiative processes component (RADFL).

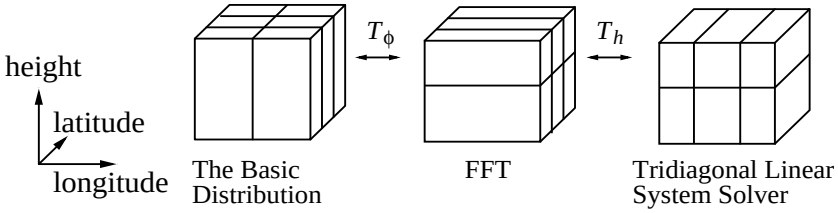

Fig. 1. Data distributions used in the Helmholtz solver ($P_{lon} = 2$, $P_{lat} = 3$)

3.2 Helmholtz equation solver

The four equations of system (1) are reduced to a Helmholtz-like equation solved by the routine `HHSOLV`

$$\frac{1}{2}\delta_{tt}\mathbf{d} - (\Delta t)^2\mathbf{G}\nabla^2\left(\frac{1}{2}\delta_{tt}\mathbf{d}\right) = \mathbf{RHS}, \tag{2}$$

where $\mathbf{G}$ is a matrix, ∇^2 is the discrete analogue of the horizontal Laplace operator in spherical coordinates, $\mathbf{RHS}$ is a right hand side. The periodical boundary conditions in longitude are imposed on (2) as well as on system (1). Diagonalization of matrix $\mathbf{G}$ allows to uncouple (2) and solve it independently on the vertical levels by a direct method involving Fourier transforms in longitudinal direction and Gaussian elimination in the meridional direction. Thus, `HHSOLV` has the following structure:

1. Transformation of the right hand side of (2) to the basis of eigenvectors of matrix $\mathbf{G}$.
2. Forward Fourier transform.
3. Tri-diagonal matrix solver.
4. Backward Fourier transform.
5. Transformation of the solution to the original basis.

Steps 1 and 5 compute matrix vector product column-by-column that makes them very suitable for the basic data distribution. Steps 2 and 4 require all longitudes while step 3, on the contrary, all latitudes.

To carry out step 2, we transpose the data in the height-longitude plane gathering longitudes but distributing levels over the processors (T_φ, Fig. 1). Upon completion of the FFTs each processor contains all wave-numbers but a part of levels and latitudes. To proceed with the tri-diagonal matrix solver we make use of the transposition again but in the longitude-latitude plane. Now it collects all latitudes but distribute longitudes (T_h, Fig. 1). Having obtained the solution of tri-diagonal systems, we rearrange the data back into the basic distribution performing the transpositions in the reverse order and calculating the inverse FFT.

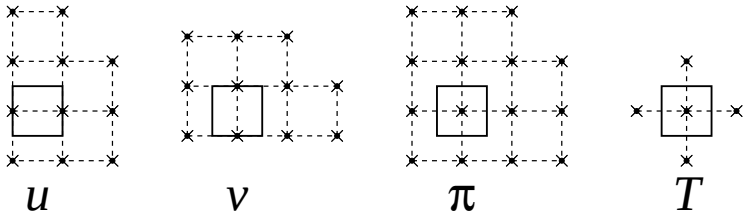

Fig. 2. Dependence stencils for the Helmholtz equation right hand side generation

3.3 Explicit dynamical tendencies generation

The routine `RHSHLM` generates the right hand side of equation (2) using the formula

$$\mathbf{RHS} = \Delta t\, \mathrm{div}\mathbf{A} - \mathbf{d}(t) + \mathbf{d}(t-\Delta t) + \Delta t \nabla^2 \left\{\mathbf{P}(t) - \mathbf{P}(t-\Delta t) - \Delta t \mathbf{A}_P\right\}, \quad (3)$$

where

$$\mathrm{div}\mathbf{A} = \frac{1}{a\cos\varphi}\left\{\delta_\lambda \mathbf{A}_u + \delta_\varphi(\mathbf{A}_v \cos\varphi)\right\}, \quad (4)$$

$$\mathbf{A}_P = \mathbf{\Gamma}\mathbf{A}_T + RT_0\mathbf{A}_\pi, \quad (5)$$

$\mathbf{\Gamma}$ is a matrix. Expressions of $\mathbf{A}_u$, $\mathbf{A}_v$, $\mathbf{A}_T$ and $\mathbf{A}_\pi$ are rather bulky and can be retrieved from [2]. They comprise values of known fields u, v, T, π, their derivatives and vertical integrals.

Basically, the routine `RHSHLM` has no global dependencies in the horizontal plane, with few exceptions occurred near the poles. In particular, to estimate vorticity ζ and surface pressure π at the North pole, u-velocity and surface pressure π are averaged along the most northern $p-1/2$ latitude circle

$$\zeta_{i,p,k} = \frac{1}{Na\Delta\varphi}\sum_{i=1}^{N} u_{i,p-1/2,k}, \quad \pi_{i,p} = \frac{1}{N}\sum_{i=1}^{N}\pi_{i+1/2,p-1/2}, \quad (6)$$

where N is the number of points along the longitude dimension. Since both u and π are distributed in longitude, we made use of `MPI_Allreduce` calls [9] to evaluate the sums in (6). The South pole is processed in a similar way.

One can observe that **RHS** computation at a grid point not belonging to the polar boundaries requires values of known variables at some neighboring columns and, therefore, has to be preceded by a communication of subdomain boundaries. To investigate the dependences inherent in (3) we processed formulas (3)–(5) and the explicit tendencies expressions with a symbolic computing system, and obtained the dependence stencils depicted in Fig. 2. Boundaries of corresponding widths are interchanged beforehand. Whenever a longitudinal derivative is calculated, the very first and last processors in a processor row also interchange data to maintain the periodic boundary condition.

3.4 Filtering

The routine RENEW applies spatial and time filtering to the obtained solution of system (1). The spatial filter damping short zonal scales poleward 69° transforms the fields into Fourier space and back. We apply a transposition in the longitude-height plane to accomplish the transformation. Staying idle, the processors that don't contain any latitude poleward 69° give rise to load imbalance.

4 Performances

The benchmarking of the resulting parallel code of INM AGCM has been done both on MBC1000M computing system and the MSU RCC's SCI-cluster (Appendix A). For a given number of processors P we measured elapsed CPU times of 6 hours' integrations on all processor grids (P_{lon}, P_{lat}), such that $P_{lon}P_{lat} = P$ (Tables 2 and 3), and calculated speed-up as the ratio of the single processor time to the minimum time obtained on P processors (Tables 4 and 5). For instance, on 8 processors of MBC1000M we tried $(1,8)$, $(2,4)$, $(4,2)$, and $(8,1)$ grids and found $(2,4)$ configuration to be the best.

Fig. 4 represents the elapsed CPU time of 6 hours' modeling and speed-up obtains on MBC1000M supercomputer and the SCI-cluster as a function of number of processors. On both machines the parallel version of the model by far outperforms its sequential counterpart yielding the speed-up of about 15 on 32 processors.

To detect potential bottlenecks of model we have carried out profiling of its major components. From the chart plotted in Fig. 5, one can observe that the filtering (RENEW) is becoming more and more bulky, as the number of processors increasing, while the radiative component (RADFL) relative time is reducing.

The deviation of prognostic variables from their uniprpocessor values after 6 hours of modeling is shown in Fig. 3 to confirm correctness of the parallelization.

5 Concluding remarks

INM AGCM was ported to a set of cluster multiprocessors indicating the advantage of using parallel computing in climate studies. The obtained performances, being still far from the ideal, are reasonable and improvable.

The work on optimization of communication as well as computational algorithms are in progress. It comprises fitting of transposition algorithms to particular machines, overlapping of communication and computations, also intercomparison of semi-implicit and explicit time integration schemes could be undertaken in the future.

Acknowledgments

The author would like to thank Academicians V.P. Dymnikov and V.V. Voevodin for encouragement as well as Dr. E.M. Volodin provided sequential code of the

Table 2. Elapsed CPU times of INM AGCM 6 hours' integrations in seconds obtained on different $P_{lon} \times P_{lat}$ processor grids of MBC1000M computing system

$P_{lon} \backslash P_{lat}$	1	2	4	8	16
1	158.3	87.76	46.22	28.50	17.65
2	92.37	49.10	27.44	16.81	10.35
4	58.75	30.75	16.80	10.63	6.61
8	37.43	20.23	12.24	7.68	5.13
16	31.47	16.78	10.74	7.85	5.09

Table 3. Elapsed CPU times of INM AGCM 6 hours' integrations in seconds obtained on different $P_{lon} \times P_{lat}$ processor grids of SCI-cluster

$P_{lon} \backslash P_{lat}$	1	2	4	8	16
1	598.05	337.36	191.56	115.19	67.83
2	340.82	190.62	105.93	65.31	42.38
4	206.26	112.82	67.38	43.07	
8	124.01	70.37	43.75		
16	84.10	52.15			

Table 4. Speed-up and efficiency of INM AGCM on MBC1000M computing system

# procs	1	2	4	8	16	32	64	128	256
(P_{lon}, P_{lat})	(1,1)	(1,2)	(1,4)	(2,4)	(4,4)	(2,16)	(4,16)	(8,16)	(16,16)
Speed-up	1.00	1.80	3.42	5.77	9.42	15.29	32.96	30.89	31.13
Efficiency	100%	90%	86%	72%	59%	48%	37%	24%	12%

Table 5. Speed-up and efficiency of INM AGCM on SCI-cluster

# procs	1	2	4	8	16	32	36
(P_{lon}, P_{lat})	(1,1)	(1,2)	(2,2)	(2,4)	(2,8)	(2,16)	(2,18)
Speed-up	1.00	1.77	3.14	5.65	9.16	14.11	14.49
Efficiency	100%	89%	78%	71%	57%	44%	40%

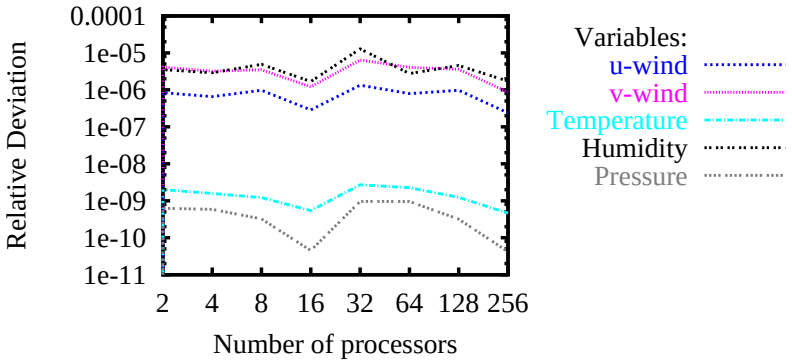

Fig. 3. Relative deviation of prognostic variables from their uniprocessor values estimated upon 6 hours' INM AGCM integrations on different number of processors of MBC1000M computing system

model. Also I want to express my special gratitude to Prof. V.N. Lykossov for his final remarks on this paper.

The work was supported by the Russian Foundation for Basic Research under grant No. 99-07-90401.

References

1. Marchuk, G.I., Dymnikov, V.P., Zalesny, V.B., Lykossov, V.N., Galin, V.Ya.: Mathematical Modelng of the Atmosphere and Ocean. Gidrometeoizdat, Leningrad (1984) (in Russian)
2. Alexeev, V.A., Volodin, E.M., Galin, V.Ya., Dymnikov, V.P., Lykossov, V.N.: Simulation of the present-day climate using the INM RAS atmospheric model. The description of the model A5421 (1997 version) and the results of experiments on AMIP II program. Institute of Numerical mathematics RAS, Moscow (1998) (reg. VINITI 03.07.98 No. 2086-B98)
3. Volodin, E.M., Galin V.Ya.: Sensitivity of summer Indian monsoon to El Nino of 1979-1998 from the data of the atmospheric general circulation model of the Institute of Numerical Mathematics, the Russian Academia of Sciences. Meteorologia i gidrologiya **10** (2000) 10–17 (in Russian)
4. Volodin, E.M., Galin, V. Ya.: Interpretation of winter warming on Northern Hemisphere continents in 1977-1994. Journal of Climate, Vol. 12, No. 10 (1999) 2947–2955

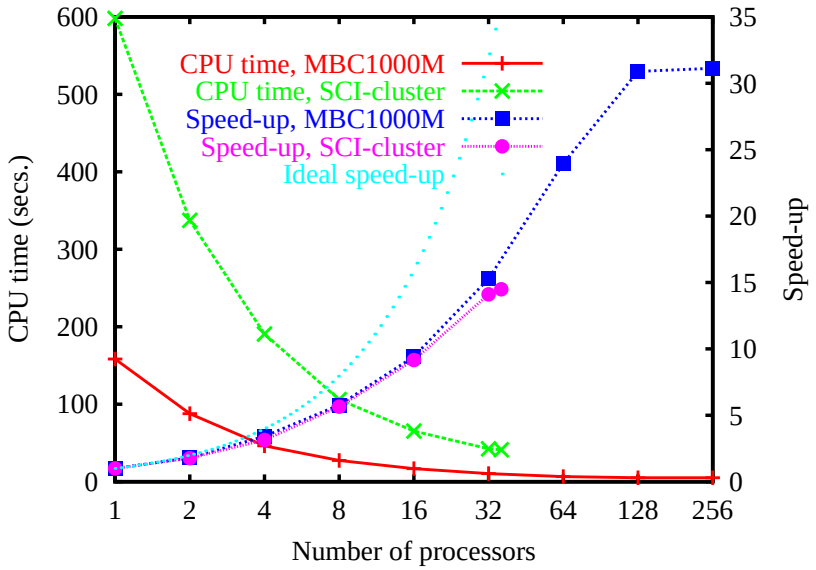

Fig. 4. Elapsed CPU time of INM AGCM 6 hours' integrations and the performance obtained on MBC1000M computing system and SCI-cluster as functions of number of processors

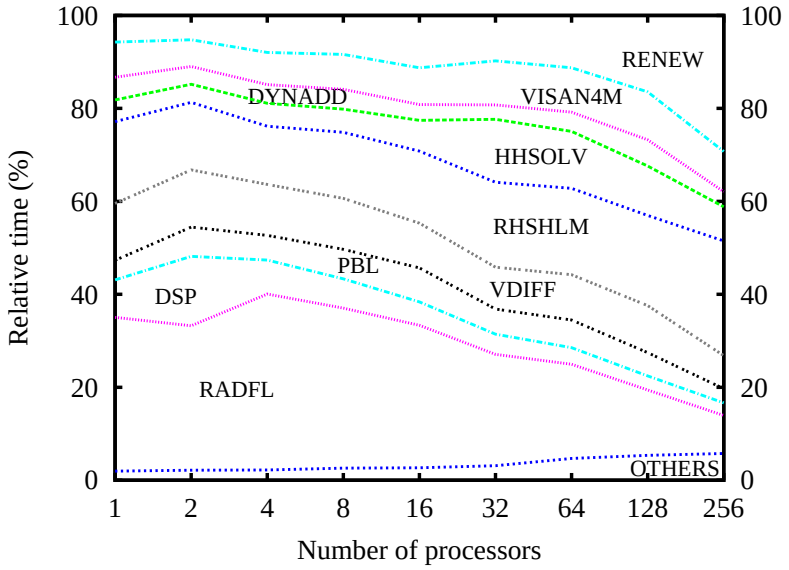

Fig. 5. Relative time spent in each part of INM AGCM on MBC1000M computing system

5. Barros, S.R.M., Kauranne, T.: Spectral and multigrid spherical Helmholtz equation solvers on distributed memory parallel computers. Fourth workshop on use of parallel processors in meteorology, Workshop proceedings, ECMRWF (1990) 1–27
6. Arakawa, A. Lamb, V.R.: Computational design of the basic dynamical processes of the UCLA general circulation model, Methods Comput. Phys. **17** (1977) 173–265
7. ECMWF Forecast Model Documentation Manual, ECMWF Research Department, edited by J-F Louis, Internal Report No. 27, Vol. 1, ECMWF (1979)
8. Kumar, V., Grama, A., Gupta, A., Karypis, G.: Introduction to Parallel Computing: Design and Analysis of Algorithms. Addison-Wesley/Benjamin-Cummings Publishing Co., Redwood City, CA (1994)
9. Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J.: MPI: The Complete Reference, The MIT Press, Cambridge, MA (1998)

A Appendix

A.1 Technical characteristics of MBC1000M computing system

MBC1000M is a cluster of biprocessor SMPs connected via 2 Gbit/s Myrinet network. Currently, it has 768 Alpha 21264A CPUs working at 667 MHz frequency. Each node has 1 Gb of main memory. The peak performance of MBC1000M installed at the Joint Supercomputer Center, Moscow is declared to be 1 Teraflop. The system is run under Linux OS. COMPAQ Fortran 90 compiler and a MPI library are available for program development.

A.2 Technical characteristics of the MSU RCC Linux-cluster

The MSU RCC Linux-cluster located at the Research Computing Center of the Moscow State University comprises 18 processing nodes equipped with doubled Pentium III processors working at 500 or 550 MHz frequency, 512 Kb second level cache, 1 Gb of main memory, Fast Ethernet card, and two SCI cards. The total number of CPUs is 36. The system is run under Red Hat Linux 6.1.