Parallel Atmospheric General Circulation Model Code Analysis and Optimization

V.P. Parkhomenko

Computing Centre of the Russia Academy of Sciences

Vavilov Str. 40 Moscow 119967 Russian Federation

E-mail: parhom@ccas.ru

Computing Centre (CC) atmospheric general circulation model (AGCM) uses uniform 72 on longitude and 46 on a latitude horizontal grid for single processor computer. Program was modified for high performance cluster. An analysis is presented of the primary factors influencing the performance of a parallel implementation of AGCM on distributed-memory, cluster computer system. Several modifications to the original parallel AGCM code aimed at improving its numerical efficiency, load-balance code performance are discussed. The impact of these optimization strategies on the performance on MVS 1000M parallel computer is presented and analyzed. There are two major components of the AGCM model code: Dynamics, which computes the evolution of the fluid flow governed by the primitive equations by means of finite-differences, and Physics, which computes the effect of processes not resolved by the model's grid (such as convection on cloud scales) on processes that are resolved by the grid. The AGCM Dynamics itself consists of two main components: a spectral filtering part and the finite difference calculations. The filtering operation is needed at each time step in regions close to the poles to ensure the effective grid size there satisfies the stability requirement for explicit time-difference schemes when a fixed time step is used throughout the entire spherical finite-difference grid.

It is found that implementation of a load-balanced Fourier algorithm results in a reduction in overall execution time of approximately 40% compared to the original algorithm. Preliminary results of the application of a load-balancing scheme for the Physics part of the AGCM code suggest additional reductions in execution time of 15% can be achieved.

A two-dimensional grid partition in the horizontal plane is used in the parallel implementation of the AGCM model. Each subdomain in such a grid is a rectangular region which contains all grid points in the vertical direction. Timing measurements on the main components of the original parallel AGCM code, using the 4 x 5 x 9 degrees resolution are shown on Table.

Table. Execution times of major components in the AGCM code

Number of processors	Dynamics, %	Physics, %	Fourier in Dynamics, %
1	63	33	10
8	67	30	18
16	70	27	20

Comparing the two modules in the main body, we can see the Dynamics part is dominant in cost especially on large numbers of nodes. Furthermore, our timing analysis on the Dynamics part indicates that the spectral filtering is a very costly component.

To solve the load-balance problem in filtering, we need to redistribute the data rows to be filtered along the latitudinal direction. In the AGCM code, the spectral filtering is performed at each time step before the finite-difference procedures are called. If it could be assumed that exactly half of the data rows in one hemisphere are to be filtered, the implementation of data redistribution for load balancing would be a relatively simple task.

The Physics component of the AGCM code consists of a large amount of local computations with no interprocessor communication required with the two-dimensional partition of the grid. The amount of computation required at each grid point is determined by several factors, including whether it is day or night, the cloud distribution, and the amount of cumulus convection determined by the conditional stability of the atmosphere. Adding to the difficulty of physics load-balancing is the unpredictability of the cloud distribution and the distribution of cumulus convection, which implies an estimation of computation load in each processor is required before any efficient load-balancing scheme can proceed.

The computation load for each processor needs to be computed or estimated by some means. The approach that we decided to adopt requires only pair wise interprocessor communications for data movement and a small amount of bookkeeping. The scheme begins with an evaluation of the local load in each processor whether it is day or night. The data load is sorted , and a pair-wise data exchange between processors is initiated for balancing the load. We would expect even better scaling be achieved for the parallel filtering as well as for the overall AGCM code for higher horizontal and vertical resolution versions.

References

Wehner, M.F., A.A. Mirin, P.O. Eltgroth, WP. Dannevik, C.R. Mechoso, J. Farrara, J.A. Spahr, Performance of a Distributed Memory Finite-Difference Atmospheric General Circulation Model, Parallel Computing 21, 1655-1675, 1995.

Lou, J. Z., J. D. Farrara, Performance Analysis and Optimization on a Parallel Atmospheric General Circulation Model Code, ISSN 1063-7133/97, 1997.