Parallelization of the French Meteorological Mesoscale Model MésoNH

Patrick Jabouille[1], Ronan Guivarch[4], Philippe Kloos[12], Didier Gazen[3], Nicolas Gicquel[1], Luc Giraud[2], Nicole Asencio[1], Veronique Ducrocq[1], Juan Escobar[3], Jean-Luc Redelsperger[1], Joël Stein[1], and Jean-Pierre Pinty[3]

[1] Centre National de Recherches Météorologiques (CNRM)
42 av G Coriolis, 31057 Toulouse Cedex 01, France
[2] Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique,
42 av G Coriolis, 31057 Toulouse Cedex 01, France
[3] Laboratoire d'Aérologie, Observatoire Midi-Pyrénées,
14 av E Belin, 31400 Toulouse, France
[4] ENSEEIHT-LIMA-IRIT, 2 rue C Camichel, 31071 Toulouse Cedex 7, France

Abstract. Numerical simulation of the atmospheric motions requires the most powerful machines and a high performance computer technology. The French meteorological model MésoNH is designed as a research tool for small and mesoscale atmospheric processes. This paper describes softwars and techniques used for implementing this numerical model on parallel processor computers and presents first results and performances.. . .

1 Introduction

The MésoNH atmospheric simulation system is a joint effort of Centre National de Recherches météorologiques (CNRM Météo-France) and Laboratoire d'aérologie (LA).

It comprises several elements : a numerical model with a comprehensive physical package, an ensemble of facilities to prepare initial states, post-processing and graphical tools to visualize the results.

MésoNH is a gridpoint limited area model. To be able to use a horizontal resolution lower than 5 km, the model is nonhydrostatic, i.e. that it uses a complete prognostic equation for the vertical speed. This assumption implies to solve an elliptic equation to determine the pressure.

The model solves the equations governing the atmospheric state evolution on a computational grid. This grid is non orthogonal to maintain the vertical coordinate and take into account the topography (more details about the Mésonh equations and discretization can be found in [1] or by visiting our web site : http://www.aero.obs-mip.fr/mesonh/).

In addition to the dynamic part of the flow, the model takes account of many physical parameterizations (turbulence, radiations, surface processes, microphysics ...). It allows for the transport and diffusion of passive scalars, to be coupled with a chemical module.

P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 1417–1422, 1999.

The model is able to perform several simultaneous simulations on a nested grid. This technique (called grid nesting) allows to focus on specific regions described by a higher spatial resolution, preserve a correct representation of large scale flow with a moderate size memory occupation.

The code is entirely written in FORTRAN 90 and in a first stage it has been developed to be run on a mono processor computer.

In order to improve the spatial resolution and the representation of physical processes, a parallel implementation of the MésoNH model is become very crucial. To achieve this goal, the CERFACS laboratory expert on efficient algorithms for solving large-scale scientific problem has joined CNRM and LA to develop tools and techniques used for the model parallelization.

2 Interface Routines

The work of parallelization began by a precise specification of requirements related to communications between processors to run the model on a parallel machine.

- Decomposition of a model on n processors
- Decomposition of m horizontally nested models on n processors
- Parallelization is achieved to be as much as possible transparently to users not aware of parallelization technics
- Full compatibility of model running on 1 processor with the same code
- Portability on any computers owning the MPI library
- Routines allowing I/O flow along a transparent way for users

To meet these goals, an interface library named ComLib has been developed. It contains all routines necessary to parallelize the Méso-NH model and is based on the standard library MPI (Message Passing Interface).

Current development is focusing on the parallelization of multiple nested models which requires additional interface routines to perform exchanges data between parent and child models, both decomposed over processors. This important part of ComLib are not described in the paper.

The Méso-NH ComLib package has been designed to enable an easy implementation of the Méso-NH code on parallel distributed memory computers. This package is implemented in Fortran 90 on top of the MPI (Message Passing Interface) library. Its main purpose is to provide the Méso-NH developers, who are not necessarily expert in parallel computing, with all the required communication capabilities while hiding to him the low level message passing paradigm details.

The implemented coarse grain parallelism exploits a parameterizable 2D decomposition in the x-y direction of the 3D physical domain (the vertical dimension is not decomposed). The automatic partitioning of the 3D domains generates by ComLib produces as many non-overlapping vertical rectangular beams

as processors requested by the user. Each beam is then assigned to one processor of the parallel target computer. Some overlapping data-structure referred to as "halo" in the sequel are required; this enables to reuse all the computational routines existing in the sequential version of the Méso-NH code. Because most of the operators are discretized using 5 point stencil in the horizontal plane the "halo" have a width of only one cell. However this width may be easily increased.

The ComLib library is based on data structure organized in an object-oriented style, with Fortran 90 user defined types. There are two top-level types

- the configuration of all the processors i.e. the way the domain is splitted into subdomains corresponding to a processor
- the communications to be performed by the current processor (send and receive operations) for each kind of communication (halo updates, transpositions, grid-nesting...).

These types are recursive, (which means that they contain pointers to variables of the same type), in order to treat the model nesting.

The definition of the characteristics of the 2D decomposition and the communication informations is entirely managed the ComLib package and its related representation is stored in private data structure of the package.

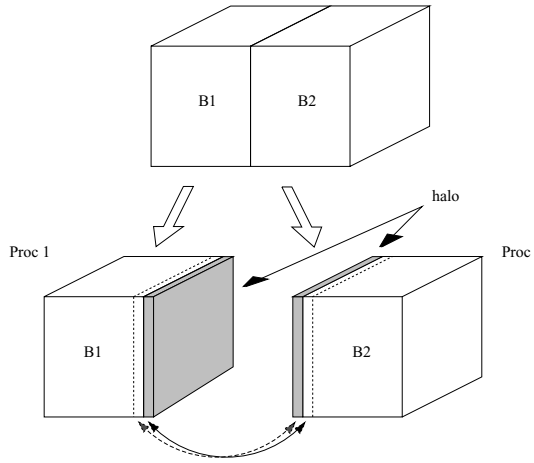

Fig. 1. Example of decomposition

In Figure 1 we depict an example of decomposition suitable for a parallel execution of Méso-NH on a two processor computer. Beam B_1 and B_2 are respectively allocated to processor 1 and 2; the variables associated with internal nodes in B_i can be updated simultaneously provided that data in the halo (shadowed area in Figure 1) have been updated correctly prior to the computation.

It is up-to the user to define the list of fields that should be kept updated in the halo region through specific subroutines to manipulate list of fields , as well as when this list of field should be updated by calling an other subroutine.

The distribution depicted on figure 2a does not work to solve the elliptic pressure system because a fast Poisson solver based on FFT computations is used. Two other types of distribution are used (figure 2b and c) respectively called x-slice and y-slice. Communication routines have been implemented that move a field between these different decompositions, then it is possible to perform the FFT for each horizontal direction. The parallel implementation of the pressure solver is presented in an other paper [2].

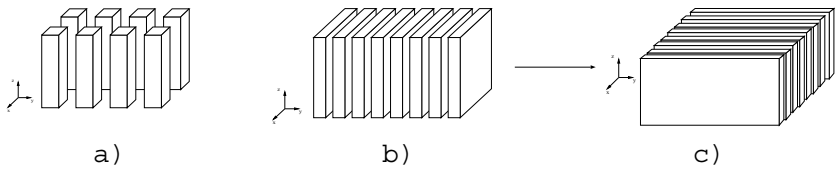

Fig. 2. the different distributions used in the model

The IO routines have been modified in order to carry out at the same time the reading and the distribution of the input fields and the gathering and the writing of the output fields, in a transparent way for the user.

3 MésoNH Parallelization

The parallelization of the code has been achieved in modifying some routines in the temporal loop to take into account the data domain decomposition. Three types of routines can be distinguished: routines to communicate fields between processors, routines allowing the treatment of physical boundary conditions and routines doing global operations.

- The main processor communications in a time step are located at the end of temporal loop to refresh the halo of all prognostic variables (wind, temperature, humidity ...). Because of the only one width of the halo area, other supplementary communications for a few fields are required to obtain a correct computation in the inner subdomain (communication of surface fluxes for example). These minor communications should be removed by increasing the width of the halo.
- Modifications in some Méso-NH routines have been brought to distinguish processor located on physical border. Special ComLib functions are used to do this.

– Global operations using the FORTRAN 90 functions SUM(A) or MIN(A), where A is an array, have been replaced by equivalent parallel ComLib function involving processor communications.

It can be noticed that the modifications in the fortran code are very limited and correspond to the call of a few ComLib routines. So, it will be easy for a Mésonh user to add his own source modification without problem concerning the parallelization.

4 First Results and Performance Analysis

To validate the parallel implementation of MésoNH, a numerical simulation has been performed two times using one and four processors. The meteorological situation corresponds to a large convective area located over France. The grid is composed by $100 \times 100 \times 30$ points which represent an horizontal domain of 2000 by 2000 km. The time step of the numerical integration is 30s. The simulation has been made on Fujitsu VPP-700 of Météo-France.

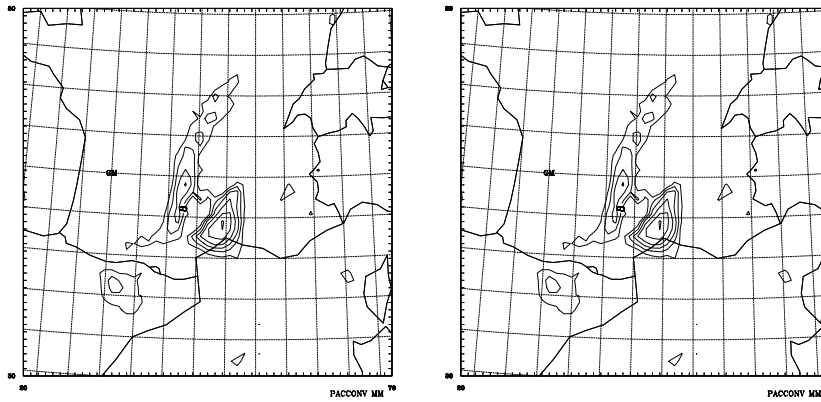

Fig. 3. cumulative convective rainfall during 3 hours (contour interval is 4 mm) for one and four processors simulation

The simulation lasts three hours and the complete physical package (turbulence, radiation, convective scheme, microphysic...) is used. The integrated convective rainfall is compared for the two simulations in figure3. No difference can be observed in these two pictures and a precise computation shows that the maximum numerical discrepancy between the two simulations is about 1%. This discrepancy is due to the compiler optimization in the pressure solver part. Simulations made over a few time steps without use of pressure solver do not reveal significant discrepancy.

For the three hours simulation, the speed-up on 4 processors is equal to 2. This moderate result can be partially explained by the decrease of vectorization efficiency. By using 4 processors, each processor performs its computations on a $50 \times 50 \times 30$ size domain. This size is rather short for a vectorial calculator such VPP700. A comparison between a sequential simulation on a 100×100 domain and a 4 processors parallel simulation on a 200×200 domain has been made. In this configuration, the amount of data per processor is the same. The efficiency of parallelization becomes 0.65. The optimization of the parallel version is still in progress.

A complete study of the parallel performance including different platforms and using more processors will be shown during the presentation.

5 Future Evolution

Some developments are required to finish the parallelization of the whole Méso-NH atmospheric simulation system : the implementation of processor communications for the grid nesting technique, the implementation of a distributed method for preparation of initial files, the use of a station network...

The final goal is to provide a research tool able to produced very fine simulations (using a huge number of grid point and including advanced parameterizations) to help the understanding of complex mesoscale phenomena.

References

[1] Lafore, J.P., Stein, J., Asencion, N., Bougeault, P., Ducrocq, V., Duron, J., Fischer, C., Hereil, P., Mascart, P., Pinty, J.P., Redelsperger, J.L, Richard, E., and Vila-Guerau de Arellano, J.: The Méso-NH atmospheric simulation system. Part I: adiabatic formulation and control simulations. Annales Geophysicae,**16** (1988) 90-109.

[2] Giraud, L., Guivarch, R., and Stein, J.: Parallel distributed fast 3D Poisson solver for Meso-scale atmospheric simulation. in this proceedings