Combined Model OpenMP & MPI for Parallel Computations in Gas Dynamics Program

A.S. Zhdanov*, A.N. Bykov

RFNC-VNIIEF, ITMF, Sarov, Russia

Key words: Standards MPI and OpenMP, Efficiency of parallel computations, Gas dynamics program.

Numerical simulations of 3D time-dependent physical processes require large computer powers to develop special technique.

The needed level of code speed of operation in solution of those problems can not be achieved without parallelism.

Cluster type computer systems with nodes containing several processors have received wide acceptance recently. Processors located on one node share common memory, those from different nodes interact via the communication network.

Studying features and models of parallel computations in the cluster systems is of a great importance for development of efficient application programs for numerical simulations.

Two basic models of multiprocessor parallel computations are recognized in the modern cluster systems:

distributed computations communicating through message passing,

parallel computations on the common field of external memory.

The model representing both the basic types is called combined model.

The principal criterion in the selection of the parallel computation model is its efficiency. In the progress of work the combined model was studied theoretically and its efficiency was estimated, the dependence of the efficiency of the combined model on those of the "pure" models was found.

There are two standards (de facto) specifying the function set to support parallel computations.

The MPI standard (along with the relevant library) specifies the scheme and basic procedures of process interaction through message communication.

The MPI was originally designed for data communication among processes using distributed memory, its current version also provides for its use in process run on shared memory.

The OpenMP standard has been developed to support the communication on shared memory (primarily in cluster systems). Unlike MPI, this is a set of directives built-in into the program body.

The OpenMP parallelization of program sections uses system Visual KAP.

In ordinary conditions it was impossible to obtain the parallel version of actual algorithms in this system. Appropriate investigation resulted in a parameter set specifying the program representation and transformation as well as a number of syntactic limitations. If these conditions are met, the process run is normal and yields a good result.

The gas dynamics equation system is transformed into a set of finite difference equations using implicit approximations in combination with the method of splitting into directions. The resultant tridiagonal-matrix equation system is solved with the sweep method.

In this program the solution domain decomposition is performed on the principle of geometry parallelism.

The program implements two principal communication schemes: the communication of faces and the parallel-pipeline method. The face communication is used when computing three-

* Corresponding author. E-mail: azhdanov@vniief.ru

point equation coefficients, new velocities, densities, volumes. The parallel-pipeline method is employed when solving three-point equations with the sweep method.

The data is not replicated on processors, and the amount of required memory for each processor decreases in proportion to their number. The communication proceeds in the non-blocking manner, and computations requiring no data from other processors are performed during the message communication.

The computation distribution proceeds outside the procedures and is supported by setting their parameters. The algorithm of the procedures is a three-dimensional loop; by changing the loop iteration range, a needed subdomain, which the computations will be performed in, can be separated from the domain of solution.

The algorithm for OpenMP operation distribution inside a MPI process is composed of the following components:

1. Creation of threads.
2. Variable scope area declaration.
3. Support of proper operation of the MPI interface procedures.
4. Distribution of the computational loop iteration range among the threads.

The principal iteration algorithm characteristics responsible for the computational load distribution mechanism are as follows:

1) computations in one iteration depend on results of the previous iteration;
2) shared memory regions are used in different iterations;
3) the computation amount changes in transition from one iteration to another.

For a given problem class the iterations in two directions are independent and can be executed in parallel. Each step uses its own memory space. The computation amount is about the same for different iterations.

The OpenMP standard offers a number of significant advantages against the MPI during development of a code. The OpenMP computation distribution makes no significant changes in the algorithms. The thread creation in one of the modules does not affect the remaining program executed in the sequential mode or in the parallel mode using MPI. This allows execution in the step-by-step manner with individual validation of each module. The processes interacting with using the MPI have no direct access to the data of each other. Therefore the algorithm of many modules must be significantly modified simultaneously in the progress of work.

The accuracy and efficiency of the developed program has been verified on a 3D problem of triaxial ellipsoid expansion to vacuum. The problem is as follows in outline. In a triaxial ellipsoid G with semiaxes a_X , a_Y , a_Z filled with gas having adiabatic exponent $\gamma = 1.4$, initial density and specific internal energy distribution is given:

$$\rho(x,y,z,0) = (1-\nu)^{\frac{1}{\gamma-1}},\ \varepsilon(x,y,z,0) = \frac{1-\nu}{\gamma-1},\ \text{where}\ \ \nu(x,y,z) = \left(\frac{x}{a_x}\right)^2 + \left(\frac{y}{a_y}\right)^2 + \left(\frac{z}{a_z}\right)^2,$$

that is density and specific internal energy are constant across the layers resulting from uniform partition of the ellipsoid along the radius from its center to the external boundary. On the ellipsoid boundary constant pressure is maintained:

$P(x,y,z,t)=0, \quad t \geq 0$.

Application of the combined model in that program gave no significant efficiency gain. The major efficiency drop therewith occurs in one node. It has been suggested that the efficiency reduction is caused not by the higher communication load, but is due to the fact that the system bus does not cope with the data communication between the processors and main memory.

To assess the scale of the effect of this feature on the computation efficiency, several identical sequential tasks have been executed on one node.

The results of the numerical experiments indicate that the effect of the memory sharing on the computation efficiency is quite significant. That is the efficiency drop is determined not by overheads of MPI or OpenMP, but the computer node architecture. Of course, this effect is much dependent on the program and problem features.

The following has been done in the progress of work.

- ✓ Efficiency of the combined parallel computation model has been studied theoretically. Principal parameters have been estimated, functions and estimating expressions have been derived.
- ✓ The scheme and basis for construction of parallel execution programs have been investigated. A practical technique for producing the programs using an appropriate translator has been determined.
- ✓ A program for solving gas dynamics equations has been developed that implements the combined parallel computation model.
- ✓ Experimental characteristics of the computation efficiencies have been obtained. They confirm the theoretical estimations made.
- ✓ Some essential features of the computational environment have been found using the developed program.