PARALLELIZATION OF AN UNSTRUCTURED MESH FLUID APPLICATION WITH BLACK-BOX SOLVERS

CHEN JUN*

* High Performance Computing Center
Institute of Applied Physics and Computational Mathematics,
Beijing, CHINA
e-mail: chenjun@iapcm.ac.cn

Key words: Parallel computation, Unstructured mesh, Parallel CFD application.

Summary. This paper presents a novel parallel method for an serial unstructured mesh fluid application with black-box solvers. These solvers are the main computing kernels here provided to be a library file and kept as black boxes, in which only small amount of information is revealed. We take the idea of separating the data structure and the computational methods in the solvers from each other, and develop a prototype to transform the serial application into parallel computation. The prototype includes parallel data structure converter, data restructure functionality for unstructured grids, and communication routines. Running the parallel application with tetrahedral unstructured mesh examples on a parallel computer, the results show that this method can easily transform such serial program with such black-box solvers into a parallel version. Some data analysis methods are also included in the prototype to help analyzing its parallel performance.

1 INTRODUCTION

Parallelization of unstructured mesh applications is not a new area in the field of Computational Fluid Dynamics (CFD). There is vast literature on parallelization techniques and parallel algorithms related to unstructured mesh applications. Many techniques are integrated into some frameworks or numerical solvers to support parallel application development. Usually, developing parallel applications on some frameworks need modify some source codes of serial programs to match the data structure and other requirements by these frameworks, in order to make use of their parallel functions. Meanwhile, most of these parallel numerical solvers are focus on some special points, such as parallel multigrid solver for large scale algebraic equation and so on. Parallelization of serial application with the help of such parallel solvers also need modify codes to be fit for parallel computation, in which some steps use the suitable parallel solvers for efficiency and high degree of resolution. Both of these methods need code modifications.

There is a special class of the serial applications need be parallelized. These serial applications are composed of a little source codes for data structure definition, describing main control flow, and time updating strategy, input and output codes and related files. Most of its main computing kernels are provided as a library file and their interfaces. Explicit methods are used in these kernels, called as “black-box” solvers in this paper. They are

*This work is supported by the National Natural Science Foundation of China under contact number 61672003.

usually legacy programs in practical computation. In this paper there is a serial program for the unstructured mesh CFD application of this class.

Since the codes of these “black-box” solvers can not be modified, the parallelization method with the help of frameworks is not always effective if the data structures are not matched well between the framework and the application. Also the parallelization method with parallel numerical solvers is also unfit. There are many researches about “black-box” problems, such as some specific numerical solvers[1-3], the parallel coupling of software[4] and parallel file system[5] and so on.

We provide a novel parallel strategy for this kind of a CFD application on unstructured mesh. We approach the idea of separating data structure and computing method from each other, design a new parallel data structure, then combine the new data structure and old computing method to realize the parallel computation.

2 PARALLELIZATION OF BLACK-BOX SOLVERS FOR UNSTRUCTURED GRID

The parallel strategies are composed of the following steps. Firstly, we split the unstructured mesh to each process using METIS tool[6-8], a software package for partitioning unstructured graphs and meshes, after input the mesh information and the number of processes. Then a parallel data structure for the unstructured mesh is designed according to the split results, including the mesh and topology information. After deleting the extra information and reconstructing the data structure, we supplements the new data structure by the ghost information. Doing these steps, the previous data structure is replaced by the new structure, which is provided to the main computing kernels transparently.

The procedure of transformation between old data structure and new parallel data structure is described as follows. The typical data structures in the previous serial program are insist of two indirect arrays. One gives the starting index and the length in the second array of a grid cell, node or face. Another array stores the index of real variables for grid cell, node or face. After splitting by METIS tool, the mesh distributions to process are known. Then the values in these two indirect arrays and the related physical variables which belong to the current process will be left and reconstructed to store in the contiguous positions, while the others are deleted from these arrays. At last, the ghost cells, nodes, or faces hosting in the neighbors are attached to these arrays for MPI communications. After the new parallel data structure is build, the variables for all the mesh topology and physical information are changed to be represented by the new structure.

The neighborhood relationship between grid cells, nodes and faces are then computed as well as those among processes. For examples, all the reduced grid cells after reconstruction step are classified into two groups. One group is identified for local cells, whose computation can complete with the local information and need no communication from neighbor processes. The second group is identified for ghost cells, which will receive information from some neighbor processes. In the first group, there is a special kind of cells, noted as boundary cells, which are near to the artificial boundary partitioned for parallel computation, and will be sent to some neighbor processes to fill the corresponding ghost cells in these processes. All the nodes and faces are also classified into many groups as above.

We developed a parallel prototype showed in Fig.1, which implements a parallel black-box integration, data restructure functionality for previous grids, and communication routines. It allows a very flexible and easy code integration of black-box solver.

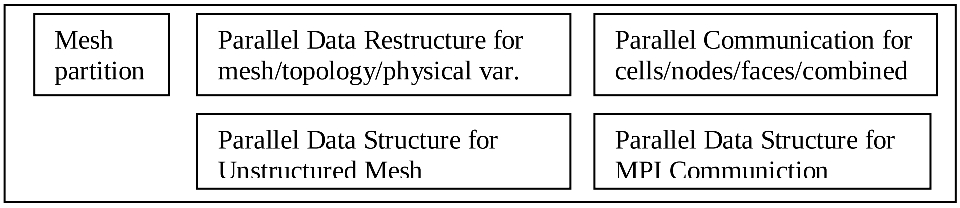

Figure 1: The structure of the parallel prototype to turn the serial unstructured mesh application with black-box solvers into parallel version.

3 NUMERICAL EXPERIMENTS

The parallel computer we used have nodes connected with Intel Ommi-Path network, and each node is equipped with 64GB RAM and two 12-core 2.5GHz Intel E5-2680 processors.. METIS 5.1.0 is used here to partition the unstructured mesh. Three models are used for test, labeled as mod_5k, mod_12w, and mod_100w respectively, where each model has different size of grid cells, from above 5 thousands, twelve thousands to a million.

First we test the partition quality of METIS when used for this application with explicit methods are used for tetrahedral grid. After measuring the mean value of local cells, nodes and faces partitioned on each process respectively, we find that their mean values are all very near to the idea value, which linear decreasing with the number of processes. Figure 2 provides the normalized standard variance of local and all(local and ghost) cells/nodes/faces in each process when running the parallel program on 2 to 64 processes. There is difference between each process, and the variance is becoming larger with the increasing number of processes used. It implies that some load imbalances are exist among processes.

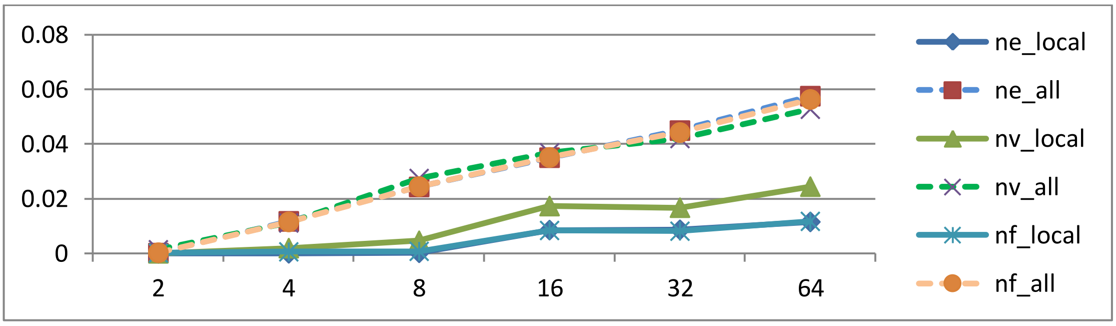

Figure 2: The normalized standard variance of local and all cells/nodes/faces in each process after mesh partition.

Then, we test the practical parallel performance of all the five black-box solvers in this application. Figure 3(1) gives the curve of ratio between each solver and the total with the process number increasing. Figure 3(2) provides the normalized standard variance of each solver among processes, which shows there are load imbalance in all solvers, especially serve in the “con” solver which consumes the most part of computation time.

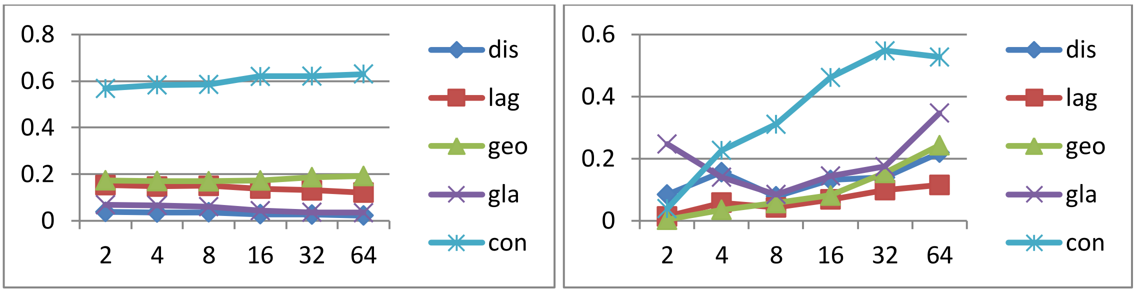

Figure 3: (1) Ratio between each solver and the total , (2) Unitary standard variance of each solver between processes.

4 CONCLUSIONS

- Using the above method, we can reuse some existing solver codes which benefit from the experiences or new developed solver codes with different computing method, and parallelize them transparently. With this approach, solver codes satisfying some limitations can easily be exchanged permitting the integration of existing codes in a plug-and-play manner. These black-box solvers offer the same interface to maximize the flexibility in exchanging codes.
- Now we have provided MPI communication for the parallel computation. These communication operations can also be converted for thread, GPU or FPGA computations. Providing various communication functions are our next work.

REFERENCES

[1] L. Grosz, C.Roll, W.Schonauer, *A black-box solver for the solution of general nonlinear functional equations by mixed FEM*, Finite element methods, fifty years of the courant element, pp.225-234, M.Dekker published, 1994.

[2] T.Adolph and W. Schonauer, Automatic domain decomposition for a black-box PDE solver, *17th Intl. Conf. on Domain Decomposition Methods*, St. Wolfgang, Austria, July 3-7, 2006.

[3] F. H. Pereira and S. I. Nabeta, *A parallel wavelet-based algebraic multigrid black-box solver and preconditioner*, J. of Applied Math., vol. 2012, Article ID 894074, Hindawi Publishing Corporation.

[4] B. Uekermann, H. J. Bungartz, B. Gatzhammer and M. Mehl., A parallel, black-box coupling algorithm for fluid-structure interaction. *5th Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, 2013.

[5] M.K.Aguilera, J.C.Mogul, J.L.Wiener, P.Reynolds and A. Muthitacharoen, *Performance debugging for distributed systems of black boxes*, In proc. Of the 19th ACM Symp. on Operating Systems Principles, pp.74-89, Bolton Landing, NY, Oct.2003.

[6] G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal on Scientific Computing*, 20(1): 359-392, 1999.

[7] G. Karypis, METIS, A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing ordering of sparse matrices version 5.1.0. March 30, 2013. http://www.cs.umn.edu/~karypis.

[8] D. LaSaille and G. Kaypis, Parallel hill-climbing refinement algorithm for graph partitioning, *Int. Conf. on Parallel Processing(ICPP)* 2016.