A Novel Parallel Computing Method for Computational Fluid Dynamics

Bai Yuguang[1,2,*], Wu Guoqiang[1,2] and Zhang Yuguang[2]

**[1] State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology
Dalian, Liaoning 116024, China**

**[2] School of Aeronautics and Astronautics, Dalian University of Technology
Dalian, Liaoning 116024, China**

Abstract

A novel parallel computing method for computational fluid dynamics research in engineering is presented. The message passing language PVM was employed to develop a convenient parallel environment with effective computing resources. It makes three dimensional numerical simulations of fluid-structure interaction problems feasible for most engineers. Wind-induced vibration problems of blunt structures with three dimensional turbulent flow simulations are implemented on the proposed clusters built with the present parallel computing method and get a direct benefit in engineering.

Keywords: *Parallel Computing Method; PVM Code; Computational Fluid Dynamics; Three Dimensional numerical Simulation*

1. Introduction

Fluid-structure interaction (FSI) is a challenging engineering problem which has been studied extensively through wind tunnel and real life tests. Over the past decade much computational work has been undertaken in the area of computational fluid dynamics (CFD) because advances in computer power and algorithms have made this approach increasingly feasible [1]. But for some blunt structures existing widely over the world, e.g. long-span bridges, two dimensional (2D) numerical simulations cannot be accurate enough, while three dimensional ones can give accurate predictions and direct visual evaluations [2].

For 3D CFD simulations, even the most advanced personal computer (PC) over the present world cannot undertake the burden of the computational demands. So parallel computing clusters are necessary. Until recently, these clusters are mainly from professional suppliers and rather expensive. For most engineers, they cannot afford these huge hardware budgets. For the past few years, PC cluster is a cost effective solution for people to have computing power of supercomputers. Besides the budget issue, another reason for the popularity of the PC cluster is that people can deploy the system as what they want. This is the benefits of open standard and open source software.

Some studies have given encouraging advices for parallel computing method. Khan *et al.* [3] presented the classification of load balancing conditions for parallel and distributed systems and discuss the use of his method. Nourah *et al.* [4] proposed the Multiprocessor scheduling when the parallel genetic algorithm was used, and he discussed task scheduling which is a most challenging problem in the parallel computing. Synchronous master to slave algorithm outperforms the sequential algorithm in case of complex and high number of generations problem was also proposed. An iterative decoding of generalized parallel concatenated block codes using cyclic permutations was proposed by Allouch *et al.* [5], which has significant benefit for numeric simulation in engineering. The effects of various parameters component codes such as inter-leaver size, block size, and the number of iterations were evaluated. The work can provide important samples for parallel applications. Olmedo *et al.* [6] proposed an approach of the point to point processing of digital images using parallel computing. It can provide an effective way for us to build smooth communication among master computers and slave computers.

A novel parallel computing method based on free operating system and software to connect common PCs together for 3D CFD numerical simulations in engineering is presented. And the advantages of it are introduced. After several clusters have been built successfully with this method, 3D wind-induced vibrations of long-span bridge structures are simulated on these clusters with detailed introductions of parallel computing including flow field, pressure contours.

2. Parallel Computing Method

2.1 Hardware and Software Demands

The demands of hardware for the present parallel computing method are fast Ethernet switch devices and common PCs, each of which has a PXE network interface card. PCs are connected to the switch and they can be transformed into a working group.

The demands of software are free GNU/Linux operating systems such as Centos. We developed the parallel code and run it in our PC cluster with Linux system. The message passing language PVM [7] (i.e. Parallel Virtual Machine) was employed to develop a parallel environment because it is convenient for people to grasp.

2.2 Installation of the Parallel Computing Clusters

One cluster consists of a server computer and many client computers. For 3D CFD simulations, many computers with same hardware were used, each of which has two cores in one Inter or AMD CPU. A roadmap for installation steps is given by Figure 1 and the last three steps are only for CFD. Actual clusters built with the present method are pictorial in Figure 2.

2.3 Advantages of the Proposed Parallel Method

The advantages of the present parallel computing method can be summarized as:

• The installation is simple. The present method uses PXE/etherboot, NFS, and NIS to provide services to client computers so that it is not necessary to install GNU/Linux on the client hard drives individually. Once the server is ready, the client computers can boot via PXE/etherboot (diskless).

• Save on hardware, budget and maintenance fees. Clusters built with the present method can achieve the same computing power as professional clusters with much lower budget than that of the professional ones. The hardware is easy to buy and the maintenance fee can be controlled to a lower standard than that of the professional clusters.

• Good expansibility. New computers can be connect to the server easily as client computers and can not influence other client ones.

So the presented parallel computing method has great potential to be applied in engineering simulations. Many engineers can afford the budget of this kind of clusters and use them simply to give better prediction of complicated engineering problems.

2.4 System Services for CFD Simulations

There are several absolutely necessary system services for CFD parallel computing. They are XINETD, RSH and RLOGIN. Many kinds of commercial software for CFD parallel computing need these services which are not started by default. They must be started before parallel computing is implemented on the clusters.

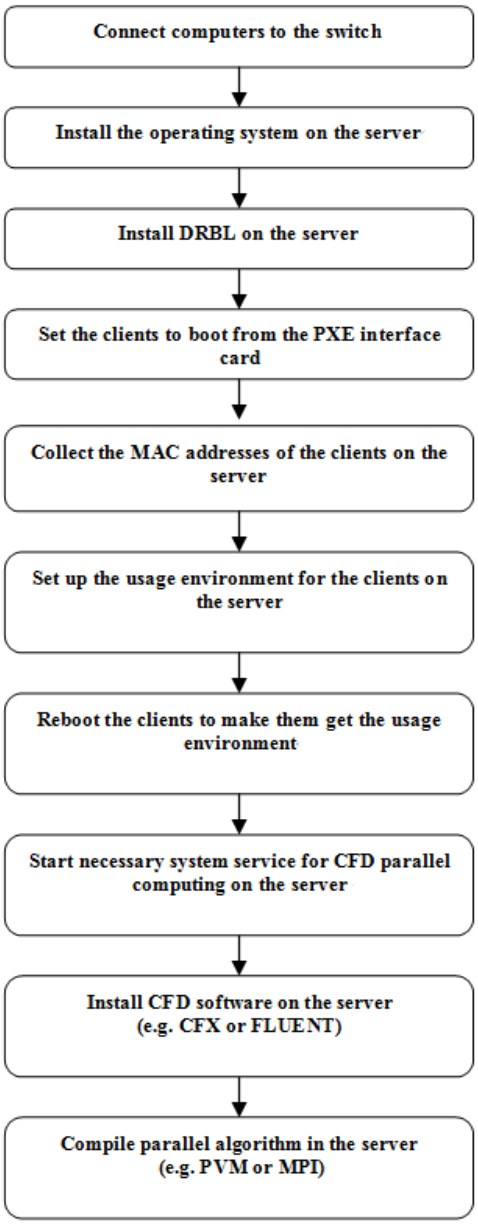

Fig. 1 Installation steps.

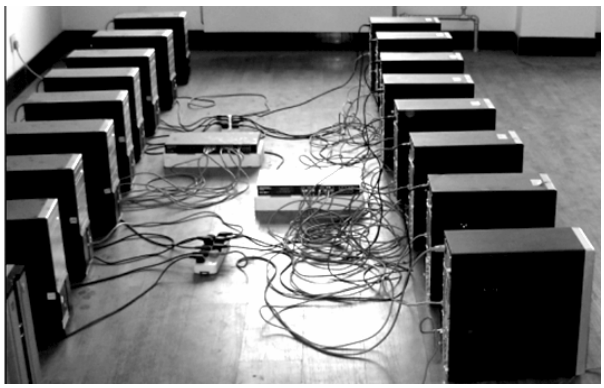

Fig. 2 Actual parallel clusters built with the proposed method

3. Application in the Simulation with CFD

3.1 Numerical Algorithm for FSI

The Navier-Stokes (NS) equations and the associated continuity equation are used with: a modification to account for turbulence; a moving mesh; and the Smagorinsky eddy viscosity model [2].

The governing equations for the FSI come from both the corresponding flow and structural analyses. All these equations can be treated as time and space dependent and need to be discretised before solution methods can be applied. The discretised incremental Navier–Stokes and structural equations can be expressed by

$$\mathbf{N(a, b)} = 0\,, \quad \mathbf{S(b)} = \mathbf{f(a)} \qquad (1)$$

where **a** and **b** are the field vectors consisting of the unknowns at the time step n+1 currently being solved for; $\mathbf{a} = \langle \mathbf{u}^{n+1}, \mathbf{p}^{n+1} \rangle$ contains the variables from the fluid domain; $\mathbf{b} = \langle h^{n+1}, \mathbf{h}^{n+1} \rangle$ contains the variables from the structural domain, where h is the vector of the decoupled generalized displacements. The field variables at the previous time step n are assumed to be known, and are not reflected in Eq. (1). The two equation sets are fully coupled.

Eq. (1) can be solved by the block-iterative method [2]. By using any available solvers for each of the two parts of (1) in conjunction with the block-Gauss–Seidel iterative algorithm, the NS equations are solved first for **a** and then for **b**. The iteration scheme which results can be written as

$$\mathbf{N}(\mathbf{a}^{(i+1)}, \mathbf{b}^{(i)}) = 0 \qquad (2)$$

$$\mathbf{S}(\mathbf{b}^{(i+1)}) = \mathbf{f}(\mathbf{a}^{(i+1)}) \qquad (3)$$

where i is the iteration counter, and it converges linearly.

For generality, (2) and (3) are both treated as nonlinear. Therefore linearization methods like the Newton–Raphson method or the Picard (fixed point) method must be used [2]. For the latter the linearization iteration is

$$\mathbf{a}^{(i+1)}_{(j+1)} = \mathbf{F}(\mathbf{a}^{(i+1)}_{(j)}, \mathbf{b}^{(i)}), \quad \mathbf{b}^{(i+1)}_{(j+1)} = \mathbf{G}(\mathbf{a}^{(i+1)}, \mathbf{b}^{(i+1)}_{(j)}) \qquad (4)$$

where j is the linearization iteration counter. The two layers of iteration i and j in (4) address both the field coupling and the non-linearity. With global convergence checked at every time step, the solution obtained should be identical to that given by the direct coupled solution to (1). Note too that the two iterations can be mixed and when merged, the equivalent iteration is

$$\mathbf{a}^{(k+1)} = \mathbf{F}(\mathbf{a}^{(k)}, \mathbf{b}^{(k)}), \quad \mathbf{b}^{(k+1)} = \mathbf{G}(\mathbf{a}^{(k+1)}, \mathbf{b}^{(k)}) \qquad (5)$$

where k is the merged iteration counter. Solvers in the form of (5) can be found in existing CFD and structural analysis software.

Figure 3 plots the flowchart of this process.

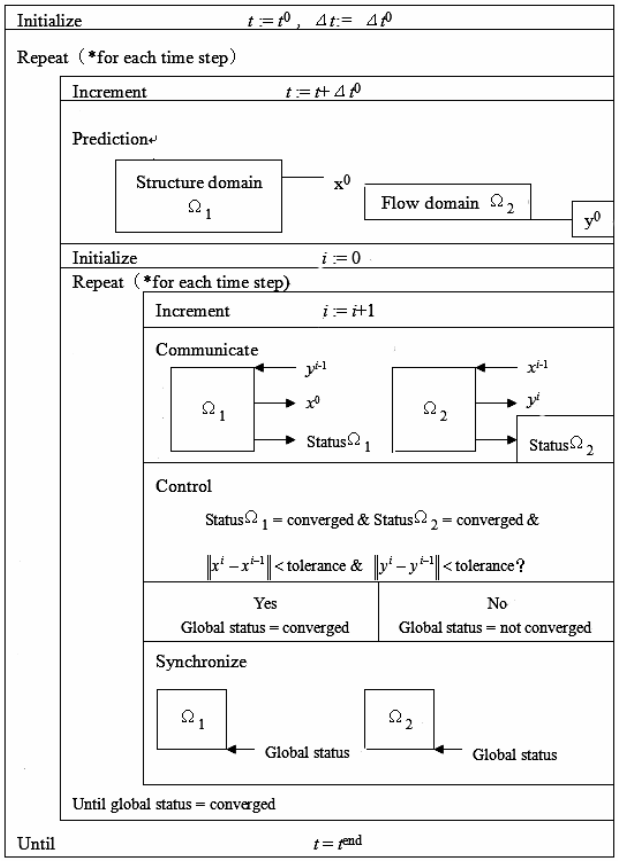

Fig. 3 Numerical analysis flowchart of the FSI problem

3.2 3D CFD Modelling

Blunt body structures like long-span bridges with different aerodynamic stability were simulated with 3D CFD turbulent modelling here. The geometrical features of them are shown in Figure 4. Figure 5 shows a 3D CFD modelling example using structure G1. The amounts of the mesh number for the five bridges are summarized in Table 1. It can be found that 3D CFD has much larger amounts of meshes than 2D CFD. The burden of 3D CFD computing must be undertaken at least by a cluster with 64 CPUs and corresponding memory. The present parallel computing for 3D CFD simulations used a cluster with 64 CPUs (i.e. including 128 CPU cores) and 128GB memory which was built with 64 PCs.

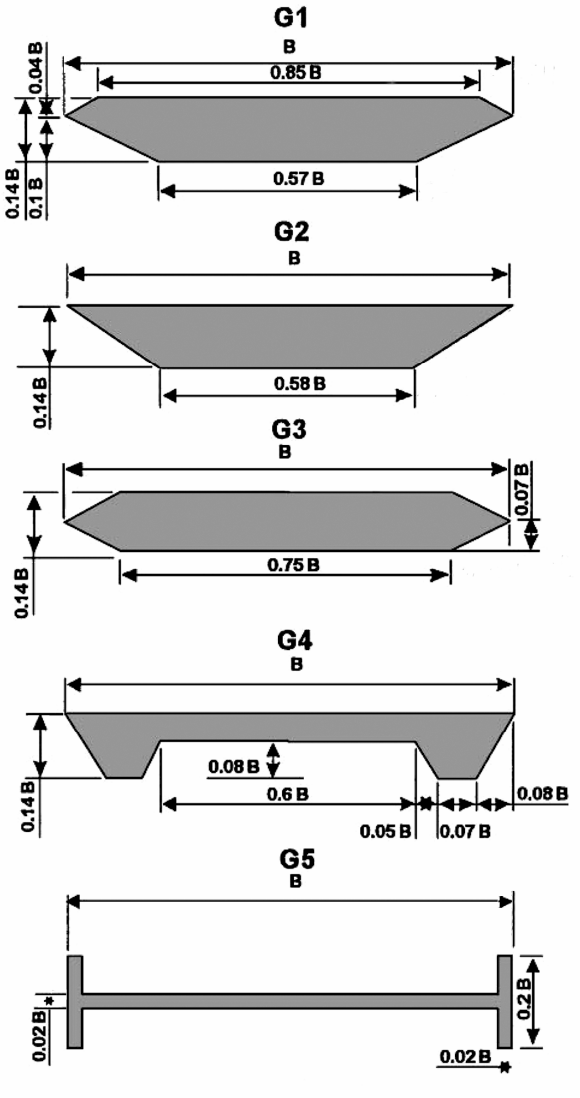

Fig. 4 The five long-span bridge deck sections G1 to G5 used.

Table 1: Mesh numbers of structures with 2D or 3D CFD simulations

	G1	G2	G3	G4	G5
2D CFD	59k	57k	67k	65k	73k
3D CFD	3183k	3093k	3696k	3602k	4090k

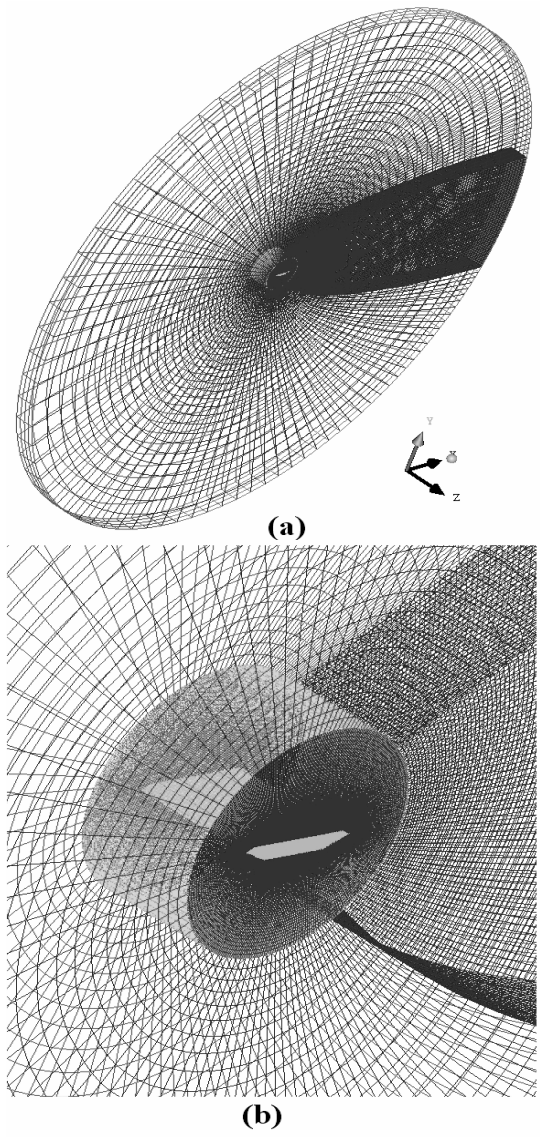

Fig. 5 3D meshes used for bridge G1: (a) overview; (b) enlargement of the inner region

3.3 Parallel Computing Process of 3D CFD

During the 3D simulation, the fluid domain is divided into 128 blocks. The block parallel code can be proposed by Allouch *et al.* [5]. The computation for each of the blocks is delegated to one of 128 processors, with one processor used as the master and the rest as the slaves.

The flux conservation between blocks is managed by the master processor. Solutions for both the fluid flow and the structural analyses according to (5) are synchronized at each coefficient iteration loop. The time for calculating structural response is much shorter than that for the fluid analysis and to reduce the time for communicating the structural response, the structural analysis is carried out on both the master and the slaves. During the iteration, the master and the slave nodes perform the mesh deformation for their own blocks using the calculated structural responses. The structural and fluid analyses here have been closely coupled.

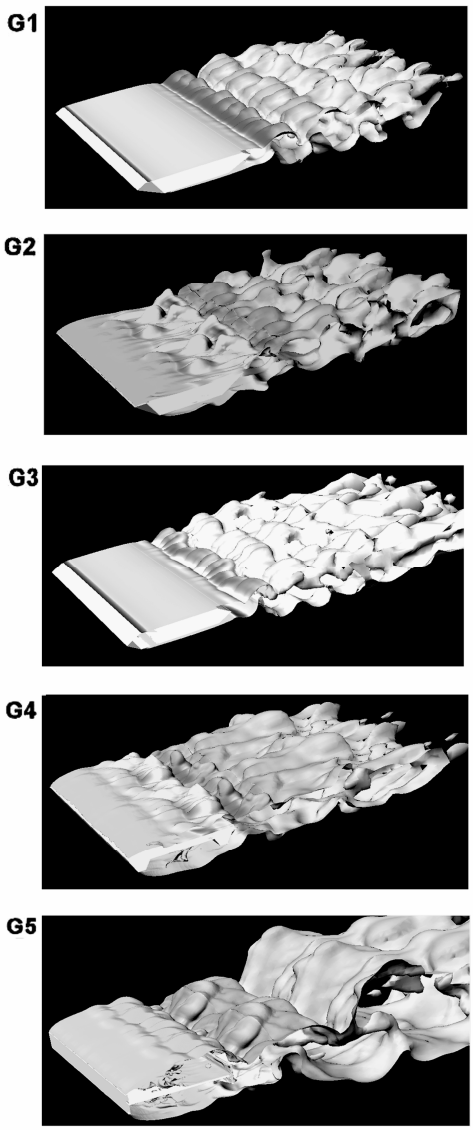

Fig. 6 Pictures of the 3D wake flow for the five bridge sections G1 to G5.

4. Results and Discussion

4.1 Simulation Results

The features of the flow field at different parts of the long-span deck cross section can be shown visually with 3D CFD simulations through the present parallel computing, e.g. the 3D wake flows for the five deck cross sections are shown pictorially in Figure 6. Hence it can be seen that structure G1 has the best aerodynamic stability, while structure G5 has the worst. Such visualization of wake through 3D CFD simulations is of direct benefit for aerodynamic analysis of structures and engineers can make their design and manufacture much more convenient with low budget.

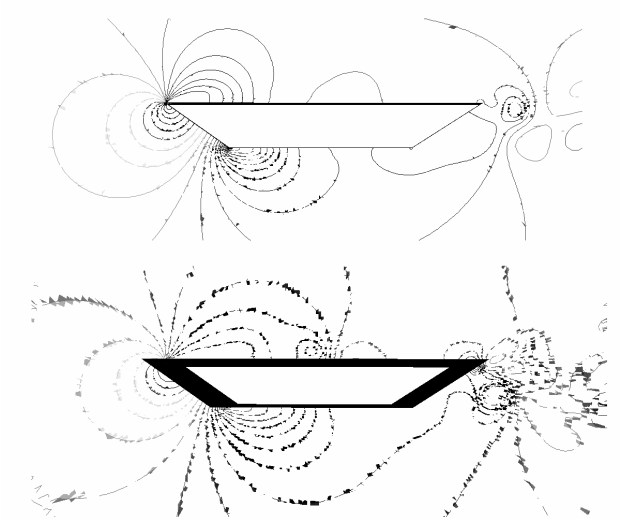

Fig. 7 Pressure contours of structure G2

Figure 7 plotted the pressure contours of structure G2 (i.e. the upper part from two-dimensional simulation and the lower part from three-dimensional simulations). It can be found that pressure distribution from three-dimensional simulations are quite different from those from two-dimensional simulations. From the contour of the lower half of Figure 7, it can be found that around structure G2, the flow reattachment occurs at the middle of the upper deck, and a large vortex region can be found at the rear part, where the structure has a slant angle.

4.2 Parallel Computing Efficiency

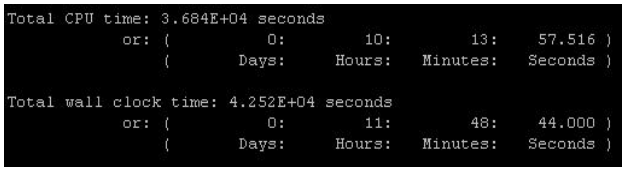

Fig. 8 The efficiency of the present parallel computing method.

The efficiency of the present parallel computing method can be calculated from Figure 8, in which CPU time means actual computing time and wall time means total time containing CPU time and communication time. It is 86.6% and it is sufficiently good for 3D CFD simulations.

4. Conclusions

A novel parallel computing method with free operating system and software is presented. It can be simply applied to common computers and has significant benefits in terms of saving on hardware, budget and maintenance fees. The parallel computing efficiency is also satisfactory. 3D CFD simulations with the present parallel method are practiced with five long-span bridge structures. The visualizations of the flows around these blunt structures can provide direct help to engineers for aerodynamic analysis of complicated structures. Also, clusters built with the present parallel computing method can be applied to computational analysis in many other parts of engineering without huge budget.

Acknowledgments

The work was financially supported by a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (Funding Code 2002030) and China Postdoctoral Science Foundation funded project (Funding Code 2012M520618)

References

[1] J. D. Anderson, Computational fluid dynamics, New York: Mc Graw Hill, 2005.
[2] Y. G. Bai, D. K. Sun, and J. H. Lin, “Three dimensional numerical simulations of long-span bridge aerodynamics using block-iterative coupling and DES,” Computers & Fluids, Vol. 39, 2010, pp. 1549-1561.
[3] Z. Khan, S. Singh, J. Alam, and S. Saxena, "Classification of load balancing conditions for parallel and distributed systems", International Journal of Computer Science Issues, Vol 8, No. 55(1), 2011, pp. 411-419.
[4] A. A. Nourah, and A. Abdullatif, "Multiprocessor scheduling using parallel genetic algorithm", International Journal of Computer Science Issues, Vol. 9, No. 44(3), 2012, pp. 260-264.
[5] H. Allouch, I. Chana, and M. Belkasmi, "Iterative decoding of generalized parallel concatenated block codes using cyclic permutation", International Journal of Computer Science Issues, Vol. 9, No. 55(1), 2012, pp. 191-198.
[6] E. Olmedo, J. de la Calleja, A. Benitez, and M. Medina Auxilio1, "Point to point processing of digital images using parallel computing", International Journal of Computer Science Issues, Vol. 9, No. 33(3), 2012, pp.1-10.
[7] Z. Roberto, P. Mario, and I. Angelo, "Enabling PVM to exploit the SCTP protocol", Journal of Parallel and Distributed Computing, Vol. 66, No. 11, 2006, pp. 1472-1479.

First Author: Yuguang Bai is a Lecturer in Dalian University of Technology, P.R.China. He received a Ph.D. degree in 2011 from Faculty of Vehicle Engineering and Mechanics in Dalian University of Technology (DUT), Dalian, P.R.China, also received his BE degrees in 2004 from Faculty of Vehicle Engineering and Mechanics in DUT. His research interests are in area of multi-physics modeling and simulation, high performance computing, and computational fluid dynamics.

Second Author: Guoqiang Wu is a Lecturer in Dalian University of Technology, P.R.China. He received a Ph.D. degree in 2009 from Astronautics School in Harbin Institute of Technology (HIT), Harbin, P.R.China, also received his BE and ME degrees in 2001 and 2004 from Solid Mechanics in HIT. His research interests are in area of multidisciplinary modeling and simulation, high performance computing, micro-satellite communication, and channel decoding.

Third Author: Yuguang Zhang is a master student in Dalian University of Technology, P.R.China. He received his BE degrees in 2010 from Department of Engineering Mechanics in Liaoning Technical University. His research interests are in area of multi-physics modeling and simulation, high performance computing, and computational fluid dynamics.