A PARALLEL IMPLEMENTATION STRATEGY FOR THE FDTD ALGORITHM*

Zheng-Hui Xue, Shi-Ming Yang, Ben-Qing Gao, Ze-Jie Zhang

Dept. of Electronics Engineering, Beijing Institute of Technology
No.5, South Street of Zhong Guan Cun, Hai Dian District, Beijing 100081, P. R. China
Tel: (86-10)68914841, Fax: (86-10)68412868, Email: zhxue@bit.edu.cn

ABSTRACT

This paper introduced a strategy for parallel implementation of the FDTD algorithm used in COW (Cluster of Workstation) parallel computing system and PVM parallel environment. Because of the one-dimension space division, this method is relatively high efficient. Some computing examples were given to prove the feasibility and correctness of this method. This method provides a solve-scheme for the EM computation of electrical-large-size complex object and can be used in radiation, scattering and EMC analysis.

INTRODUCTION

The requirement for high performance computing in engineering is endless. Many fields such as complex object modeling, engineering design and automation, electronics, aeronautics and medicine put forward challenges to computing. For example, when the RCS of an 15λ object was calculated with the FDTD method it need at least $200\times200\times200=8{,}000{,}000$ grids, all the variables can get to $10{,}000{,}000$ level. The calculating is impossible for a PC even through the performance of PC is much higher than before. For the requirements of factual problems, building parallel computing machine and developing parallel computing algorithm become necessary[3-4].

Now the research for parallel computing machine and parallel computing algorithm have already progressed much. The parallel computing machine can be classified to 6 kinds: SIMD (Single Instruction Multiple Data); PVP (Parallel Vector Processor); SMP (Symmetric Multiprocessor); DSM (Distributed Shared Memory); MPP (Massively Parallel Processor) and COW (Cluster of Workstation). Besides SIMD has special usage , the other 5 kinds belong to MIMD (Multiple-Instruction Multiple-Data)[4] and used normally.

The FDTD method, first proposed by Yee, permits one to study the electromagnetic fields with objects of arbitrary shape and material composition. It has been used for a wide variety of applications including scattering, absorption, bioelectromagnetics, antennas and EMC/EMI analysis.

The main advantages of this method lie in three aspects ： Firstly, this method is a time domain method that means the data in the whole frequency band can be obtained by only once calculation in time-domain. Secondly, this method can easily model complex objects. Thirdly , the necessary memory is relatively smaller than other low frequency numerical technique such as the Moment Method[1-2].

In the normal FDTD method, the EM field values of every last time step and other variables are stored in the memory of a computer. The larger the object's electrical size that is analyzed is, the more memory the computer needs. For example , a computer with 32 MHz memory is only capable for the computation of objects that correspond to the space size about $80\times80\times80$ grids . So, we often use a powerful workstation or even gigantic computer to analyze the EM problems of a large electro-size object. But if we have only a PC or a workstation that have no enough memory and also must handle with a electrically large-size object, even if we use the virtual memory technique, the computation still can not be executed, we would meet a serious problem.

In order to solve this problem, a high efficiency way is implement FDTD algorithm in a parallel computer. There are some researchers have already developed some strategy for FDTD parallel computing[7][10], and now this research are becoming bloom. This paper introduced a strategy for parallel implementation of the FDTD algorithm using COW (Cluster of Workstation) parallel computing system that are built with LINUX operating system and PVM parallel software. Some computing examples were given to prove the feasibility, correctness and high efficiency of this strategy.

PARALLEL COMPUTING SYSTEM

System Hardware

COW system have many advantage[4]: ①The algorithm programming is easy than other parallel machine. The programmers needs not to study special parallel programming language (such as parallel C, C++ and parallel FORTRAN) and can obtain parallel programs by inserting some functions and sentences provided by some parallel

* The work is supported by National Natural Science Foundation of China (No.69931030).

software into normal serial programs, then can be executed in COW; ②The structure of system is flexible. The user can use all kinds of PC or workstation to build a parallel computing machine and can get a wide range of usages. ③The ratio of performances to price is high. A COW can get to Gflops level in computing ability but cost much little than those of the gigantic machine or MPP having the same ability. ④A COW can use separate computing source. When the PC is not busy it can be used as a member of a COW to fulfill a part of parallel computing. ⑤The scalability is perfect. The user can increase the number of PC or workstation in a COW system to obtain high accelerating ratio.
COW system can be constructed by two parts: workstation and interlinkage network. Interlinkage network may be a normal Ether-net, FDDI or ATM. Workstation can be a normal PC or SMP. From the review angle of user or programmer, COW just likes a single computer but not many computers. Fig.1 shows the structure of our COW, the workstation is normal PC and the interlinkage network is the 10M/s Ethernet. When the system is simplest, whole system can be built by two PC with Ethernet card and HUB.

System Software

The system software has two parts: operating system and parallel message passing software. The operating system is "Linux" system based on UNIX, parallel message passing software adopt Parallel-Virtual-Machine(PVM) software. About the details of Linux and UNIX, it can be refered to references [8],[9],[3] and [6].

FDTD PARALLEL ALGORITHM IN COW

The details on FDTD algorithm have already been discussed in many references such as [1] and [2]. The preconditions of the parallel FDTD algorithm is the dividing of all computation task and then every node of COW can compute one part of all computation task. Firstly, the basic principle of FDTD algorithm makes the division of computational space possible. According to the principle of FDTD algorithm, the electromagnetic field value at certain position can be decided by the value of last time step at this position and electromagnetic field value of this time step at nearby position. The electromagnetic field value has no direct relation to the values at position far from this point. So, the whole computational space can be divided into some sections that can be commutated in some nodes of parallel computing system. The exchange of field values between nodes can be executed only at interface between sections. According to the basic cognition, the relay computing between parallel nodes can be executed to simulate the serial computing in a single PC or workstation. This is the key point of our parallel FDTD algorithm.

Division of Computational Space and Exchange of Field Value

In our parallel FDTD algorithm, the Master-Slave programming-style has been used. The Master-Slave program has three step normally: Firstly, slave programs are created by master program and the information of task and the parameter of computing are delivered from master program to every slave programe; Secondly, the parallel computing are executed in every node of COW, within every time step the synchronization and communication between every node are kept; Finally, the computing results are transferred to master program from every node and the computing are terminated. Taking two nodes COW as an example, the computing task are divided into two parts along Y axis in one dimension. The segmentation is shown in Fig.2, and this fig also shows the data needed to tranfer between nodes. Taking the iterative computing of H_x and E_z components of one-dimension as an example, the difference formula for $j = m - 1/2$ and $j = m$ point are shown below[1].

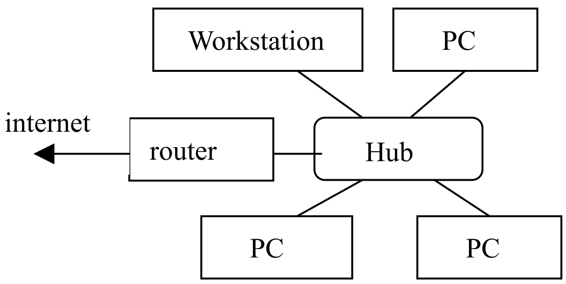

Fig.1. Structure of Parallel System

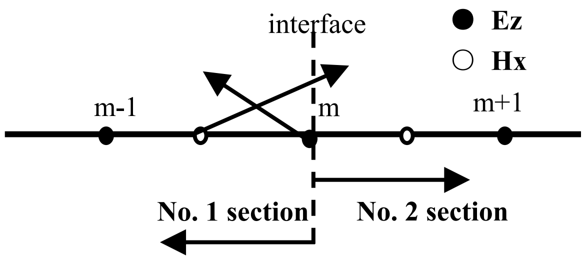

Fig.2 the one-dimension segmentation and the data needed to tranfer between nodes

$$H_x^{n+1/2}(m-1/2) = C'_x(m-1/2)\{-D'_x(m-1/2)H_x^{n-1/2}(m-1/2) - [E_z^n(m) - E_z^n(m-1)]/\Delta y\} \tag{1}$$

$$E_z^{n+1}(m) = C_z(m)\{-D_z(m)E_z^n(m) - [H_x^{n+1/2}(m+1/2) - H_x^{n+1/2}(m-1/2)]/\Delta y\} \tag{2}$$

Form Fig.2 and (1)(2), we know that we must deliver the value of $E_z^n(m)$ from node No.2 (section 2) to No.1 (section 1) before the value of $H_x^{n+1/2}(m-1/2)$ can be calculated and we also must deliver the value of $H_x^{n+1/2}(m-1/2)$ from node No.1 (section 1) to No.2 (section 2) before the value of $E_z^{n+1}(m)$ can be calculated.

Course of Time and Space Parallel Computing

Taking three nodes COW and one dimension computing (y direction) as an example, making m_1 and m_2 denote the grid number of boundary between No.1 and No.2 sections and between No.2 and No.3 sections, The process of computing are shown in Fig.3. In this figure the symbol $\Longrightarrow$ means normal FDTD discrelization process and the symbol $\longrightarrow$ means the delivering process between sections.

Supposed the computing starts in No.1 section. Firstly, No.2 section must wait for the results of interface that computed in No.1 section. When the data needed for transferring arrive at No.2 section, the No.2 section starts the computing and delivers the result data needed by No.1 section to No.1 section. Then the same processes continue in No.2 and No.3 section. After the computing in No.3 finish, this time step will stop. In the next time step, the above process repeat till all time step finish. So the three slave program are parallel in computing space and computing time. When the section are divided more than three and FDTD computing is executed in two or three dimension, the process is similar with described above.

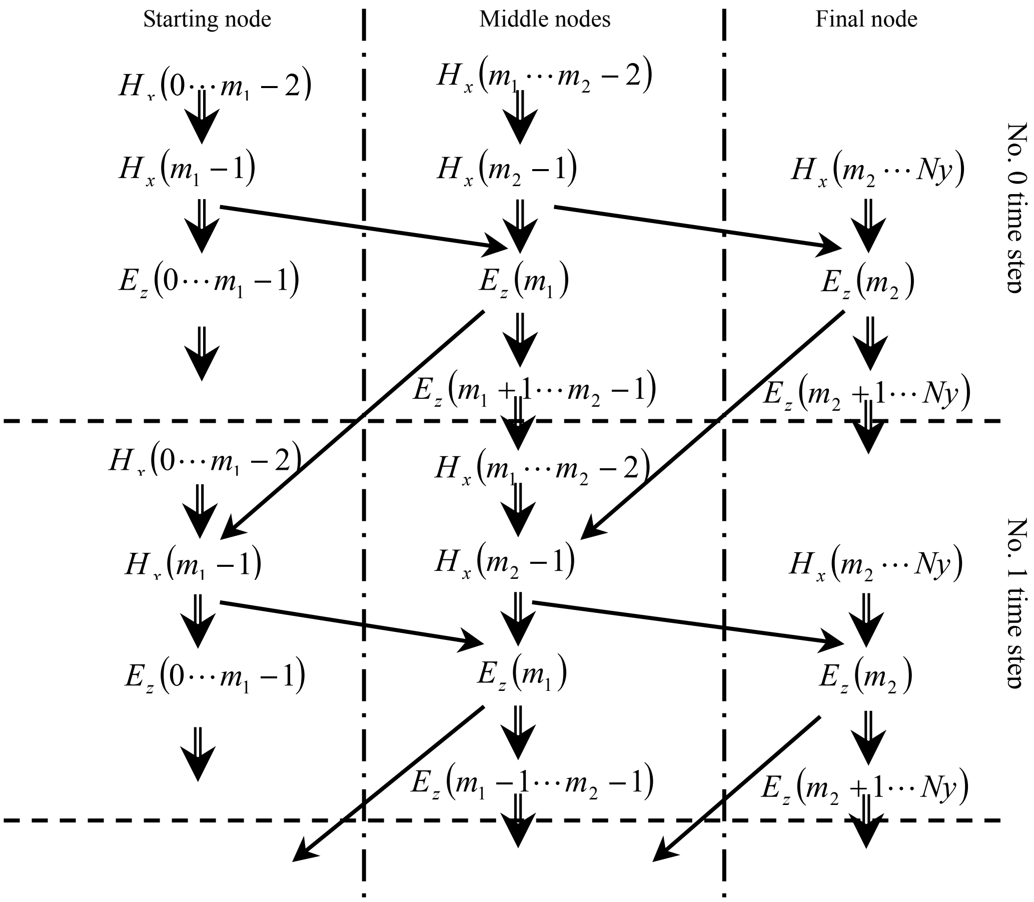

Fig.3 the process of parallel computing

Other Problems Related to Parallel Computing

Some periphery problems related to FDTD parallel computing must be settled down for the successful executing of FDTD parallel program, they are: division of computing model, parallel disposal for absorb boundary condition and connect boundary condition between scattering field and whole field, division of far field integral surface. All these can be dealt with according to the principle described above.

NUMERICAL RESULTS

According to the structure of hardware and software described above, we built a test COW system with two PC and 10M Ethernet. The two PC are PⅡ266/64M and PⅡ200/64M, which have Red-Flag Linux2.4 and PVM3.4.2. For proving the feasibility, correctness and high efficiency of this strategy, some computing problems have been executed by our parallel FDTD computing strategy and successful serial FDTD program respectively. We need not concern the normal setting of FDTD algorithm but assure the computing object is absolute same in parallel computing and reference serial computing. Taking a RCS calculation as an example, the whole computing space includes $76 \times 76 \times 76$ grids and is divided into two parts equally along Y-axis. The recorded results in far field are shown in Fig.4 and Fig.5 and the results obtained from normal FDTD are also shown in the two figure. From Fig.4 and Fig.5, we can find that this strategy of parallel FDTD Algorithm is successful and the precision is absolutely same as normal serial FDTD, so we can say this strategy is feasible. According to the computing, the accelerating ratio is 1.6 and the parallel efficiency is about 0.8. This means the parallel FDTD strategy is efficient. Now this strategy has already been generalized to radiation and EMC computing.

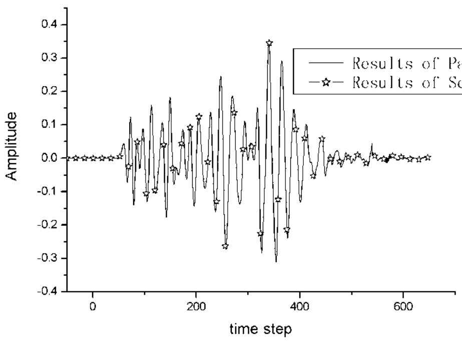

Fig.4 E_φ component in far field

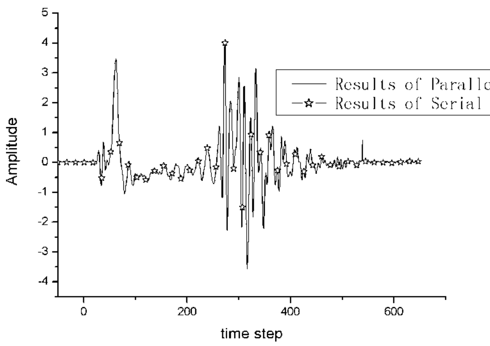

Fig.5 E_θ component in far field

REFERENCES

[1] Gao Benqing, "FDTD Method", Publishing House of China National Defence Industry，1995.
[2] Wang Changqing, Zhu Xili, "FDTD method in Electromagnetic", Peking University press, 1994.
[3] Sun Jiatan, Zhang Linbo, "Network Prallel Computing and Distributed Programming Envirument". China Science Press, 1996.
[4] Chen Guoliang, "Parallel Computing: Architecture, Algorithm and Programming", Higher Education Press,1999.
[5] Zhang Baolin, Gu Tongxiang, "Digital Parallel Computing Theory and Method", Publishing House of China National Defence Industry, 1999.
[6] A.Geist, A.Beguelin, J.Dongarra, W.C.Jiang, R.Manchek and V.Sunderam, "PVM: Parallel Virtual Machine". The MIT Press, 1994
[7] Amir Fijany, Michael A.Jensen, Yahya Rahmat-Samii, Jacob Barhen, "A massively Parallel Computation Strategy for FDTD: Time and Space Parallelism Applied to Electromagnetic Problems",IEEE Trans. on AP, pp 1441-1449, 1995
[8] Cheng Lifeng. "Red Hat Linux 6.0 Network and Communication", China People's Posts & Telecommunication Publishing House, 2000.
[9] Chen Jiu, Long Hao, "Red Hat Linux 7 Self-Culture Guide", Publishing House of National Defence Industry, 2001.
[10] Ma Jifu, "Generalized Computation of 3-Dimensional EM Field of Resonant Cavities and Its Applications", Thesis for Ph. D of China Institute of Electronics, 1998
[11] Xue Zhenghui, Gao Benqing, "Space Divided FDTD Method", Proceeding of 4th Microwave EMC of China, 1998.