Progress In Electromagnetics Research, PIER 80, 393–408, 2008

A NOVEL FDTD APPROACH FEATURING TWO-LEVEL PARALLELIZATION ON PC CLUSTER

Y. Liu, Z. Liang, and Z. Q. Yang

College of Physical Electronics
University of Electronic Science and Technology of China
Chengdu 610054, China

Abstract—To improve the parallel efficiency in the case of the fine-grained FDTD computing on PC cluster, the concept of "two level parallelization on PC cluster" is presented, and a high performance MPI-OpenMP hybrid FDTD algorithm is developed. In the hybrid algorithm, MPI is used in conjunction with OpenMP multithreading to achieve two level parallelism of the data and tasks at the basis of the domain decomposition FDTD method. Besides, to enhance the flexibility of the parallel FDTD, the interpolation between subspaces is also discussed. The simulation example of a printed antenna for automobile is given. Computations are performed for different numbers of PCs and contrasted with two conventional parallel FDTD algorithms on PC cluster. The results show that with the decrease of the computational granularity on each computer, the novel algorithm is more efficient, and moreover, it can also lessen the influence of the sub-domains virtual topology on the parallel FDTD performance.

1. INTRODUCTION

The finite-difference time-domain (FDTD) method provides a simple and efficient means of solving Maxwell's equations for a wide variety of problems [1–5]. However, some problems, involving the electrically large and complex structures, require large amounts of memory and long simulation time [6–8], which could place the solution of the problems beyond the capabilities of any one single-processor computer. To overcome the bottlenecks of the computational power and storage requirement, the exploitation of the parallel FDTD implementation has been presented and developing rapidly in the last fewer years [9–11].

A cost-effective, high-performance computing platform for the parallel implementation of the FDTD algorithm is the PC cluster using the message-passing interface (MPI) library, which is a local area network system consisting of multiple interconnected personal computers (PCs), and is already widely employed for parallel computing [12, 13]. Due to the loosely-connected configuration of a PC cluster, the communication overheads among computers are quite high when the parallel FDTD runs on a cluster system. It brings forth a severe constrain on obtaining high parallel efficiency. From previous literatures [14, 15], the parallel FDTD can achieve high parallel efficiency only in the coarse-grained parallelism. With the quantity of computers increasing, this corresponds to decrease the computational granularity on each computer, and its parallel efficiency deteriorates sharply. In the circumstances, lots of computing capability of the PC cluster is wasted due to low parallel efficiency. Hence it is quite important to improve the efficiency of the fine-grained parallel FDTD on PC cluster.

This paper presents the concept of "two level parallelization on PC cluster", and develops a high performance hybrid parallel FDTD algorithm based on the domain decomposition strategy (DD-FDTD), which features both OpenMP shared memory programming and MPI message passing. In the hybrid algorithm, the task parallelization at the basis of the data parallelization of the conventional MPI-FDTD algorithm is achieved with the use of OpenMP multithreading as the second level parallelization. Besides, to enhance the flexibility of the parallel FDTD, the interpolation between sub-domains is also discussed. By using this hybrid method, a simulation of a modern hidden printed antenna for automobile is conducted, and various comparisons are made to prove the efficiency and accuracy of this method. The results show when the grain size of the domain decomposition is decreased, this novel hybrid parallelization scheme is an effective technique for improving the efficiency of the parallel FDTD on PC cluster.

2. NUMERICAL MODEL AND ANALYSIS

Consider a hidden printed antenna for automobile as shown in Figure 1. The antenna is very thin, and sticks tightly beneath the windshield of the rear roof. The vehicle was modeled using perfectly conducting plates and windshields. The geometry of the model is 4140 mm along the x axis, 1540 mm along the y axis and 1400 mm along the z axis. Figure 2 shows the top view of the discrete vehicle model used in FDTD computing. With the dielectric constant of the windshield $\varepsilon_r = 4$, $\varepsilon_r = 5.5$ and $\varepsilon_r = 7$, we employ the parallel FDTD to calculate the return loss (S_{11}) of the antenna respectively [16–18].

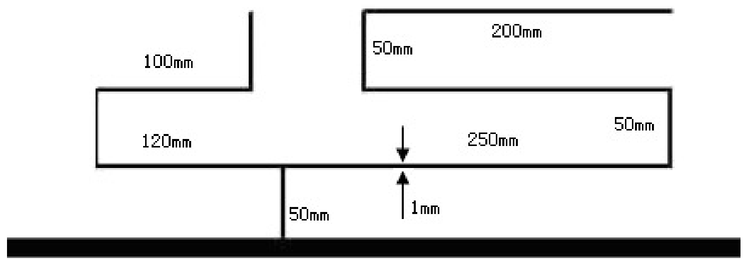

Figure 1. The geometry of the printed antenna for automobile.

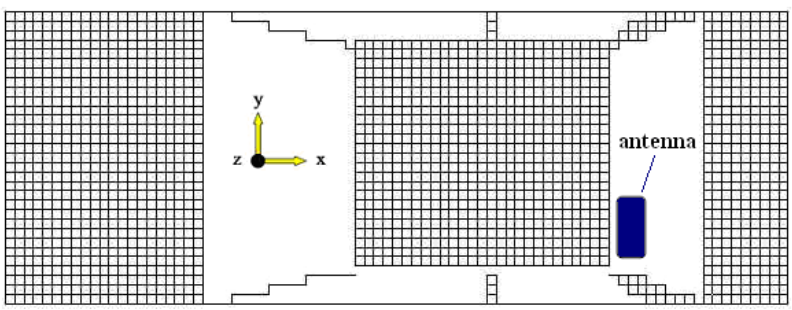

Figure 2. The top view of the discrete vehicle model.

To calculate the electric/magnetic field at (i, j, k) by using FDTD algorithm, we need its previous time value and that of the nearest neighbouring magnetic/electric fields. If we divide the whole FDTD computation space into many subspaces, we are only required to exchange the tangential electric/magnetic field components at the subspace boundaries. The FDTD method is, therefore, very appropriate to parallel computation, and each subspace is able to be treated by a computer in PC cluster. Figure 3 shows how the field components are exchanged between adjacent subspaces in the case of the two-dimensional domain decomposition.

However, this is only the most simple situation that the whole computation space is divided uniformly in Cartesian coordinates. It is more general that the computation space is divided according to the features of the original problem, and each subspace can have its own conformal lattices to suit for the special shapes. The FDTD computation is carried out independently in local meshes and local time step [19]. In order to exchange fields data between adjacent subspaces,

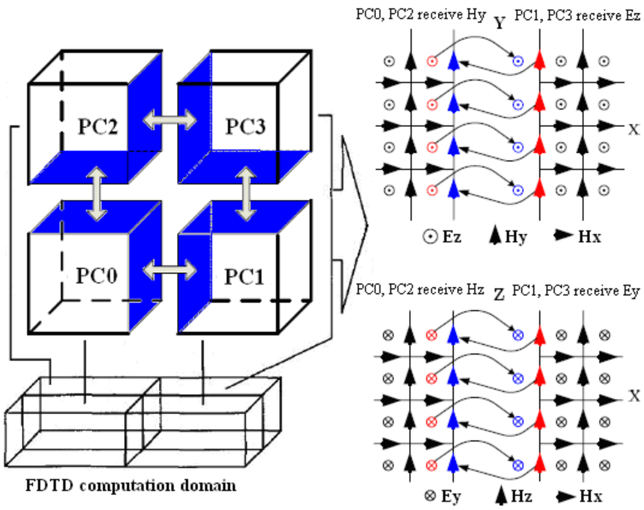

Figure 3. Fields passing scheme for the two-dimensional domain decomposition FDTD.

the divided subspaces have an overlapping boundary region, and the interpolation scheme with overlapped meshes is applied [20].

It can be seen from Figure 2 that the automobile antenna is mounted beneath the tilted rear windshield of the vehicle, and is not parallel with the Cartesian coordinate plane. As the discretization grids of the conventional FDTD are usually rectangular in shape, the original Yee scheme does not produce accurate results for simulating the antenna on the inclined surface owing to the staircasing approximation. In view of the factor that the printed antenna is the chief radiation source of the vehicle body, the staircasing introduces significant errors. To avoid compromising accuracy and simplify the antenna modeling, the subspaces related to the rear windshield are inclined to fit the printed antenna plane to an FDTD lattice.

In the FDTD computing, we choose $\Delta x = \Delta y = \Delta z = 50\,\text{mm}$, and a particular type of PML absorbing boundary condition, referred to as *the uniaxial perfectly matched layer* (UPML) introduced by Gedney [21], is used to truncate the computation domain. The whole discretized space along x, y, z direction is $129 \times 77 \times 86$, and is divided into six subspaces along XY plane. To get the field data at the subspace boundary, the inclined subspaces related to the rear windshield utilize the interpolation scheme with overlapped meshes, and the other subspaces exchange directly the tangential field data

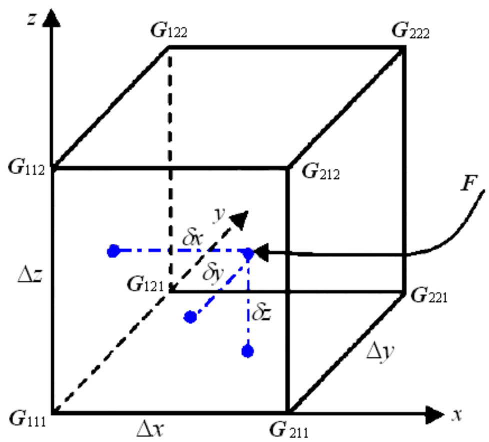

Figure 4. The tri-linear interpolation scheme.

at the boundaries. In this example, exclusive of being tilted along coordinate plane, the meshes of the inclined subspaces corresponding to other subspaces are coincident. Thus, we can ensure the stability and accuracy of the FDTD iteration by using tri-linear interpolation method in the overlapped region of two adjacent subdomains. As shown in Figure 4, F is a unknown electric or magnetic field value to be interpolated at a subspace boundary, with coordinates (δx, δy, δz), and G is a known electric or magnetic field value in another subspace, with the grid increments (Δx, Δy, Δz). From which we obtain the interpolation formula as shown in Equation (1).

$$\begin{aligned} F = &\frac{(\Delta x-\delta x)(\Delta y-\delta y)(\Delta z-\delta z)}{\Delta x\Delta y\Delta z}G_{111} + \frac{\delta x(\Delta y-\delta y)(\Delta z-\delta z)}{\Delta x\Delta y\Delta z}G_{211} \\ &+\frac{\delta x\delta y(\Delta z-\delta z)}{\Delta x\Delta y\Delta z}G_{221} + \frac{\delta y(\Delta x-\delta x)(\Delta z-\delta z)}{\Delta x\Delta y\Delta z}G_{121} \\ &+\frac{\delta z(\Delta x-\delta x)(\Delta y-\delta y)}{\Delta x\Delta y\Delta z}G_{112} + \frac{\delta x(\Delta y-\delta y)\delta z}{\Delta x\Delta y\Delta z}G_{212} \\ &+\frac{\delta x\delta y\delta z}{\Delta x\Delta y\Delta z}G_{222} + \frac{(\Delta x-\delta x)\delta y\delta z}{\Delta x\Delta y\Delta z}G_{122} \end{aligned} \tag{1}$$

When the meshes corresponding to different subspaces are not coincident, the tri-linear interpolation may lead to the instability of the FDTD iteration. To reduce the error resulted from the interpolation, one technique is to amend the interpolation formula based on the generalized absorbing boundary, which can correct the interpolation errors on the second electric fields layer [22], and the other is to lessen the grid increment of the overlapping domain. The latter will result in the increase of computation and storage, however, which can be effectively solved by parallel computation.

3. IMPLEMENTATION OF THE HYBRID PARALLEL FDTD

The parallel DD-FDTD algorithm possesses the typical features of data parallelization, and the data of the problems are portioned among the different processors. Therefore, it can be implemented easily by using the Message Passing Interface (MPI) library. As one of the most popular standard for parallel programming, MPI provides some optimized communication functions which are very important to the design of high efficient parallel FDTD [23]. For example, nonblocking send/receive functions MPI_Isend()/MPI_Irecv(), which are used in parallel algorithm to overlap computations with communications, and reduce the interaction overheads among subspaces. However, unlike the commodity cluster, the PC cluster based on the traditional Ethernet network has not the self-governed communication controllers. When the data are transferred from one PC to another PC, the NIC (Network Interface Card) must interrupt the CPU time by the time-sharing technique, and can not give directly support to the optimized communication modes of the MPI functions from the underlying hardware. Thus, for PC cluster, the performance differences of the various MPI communication modes are inapparent, and the optimized modes plays a relatively small role in improving the parallel efficiency as compared with the commodity cluster. It is difficult to achieve higher performance only by using the optimized communication functions. This conclusion was also confirmed by literature [24] in which the numerical tests were conducted on PC cluster.

The CPUs, such as Pentium4 with hyper-threading technology, dual-core Conroe or AMD64 X2, are already widely employed on modern local area network, and give directly support to multithreading or multiple tasks from the underlying hardware. Therefore, the PC cluster is essentially a hierarchical parallel platform, and possesses the capability of two level parallelization: 1) multithreading parallelization mode based on shared memory within PC; 2) data parallelization mode based on message passing between PCs. By using two level parallelization modes simultaneously, referred to as *hybrid parallelism*, we can improve availably the efficiency of the parallel computation, and enable the FDTD applications to scale to a large number of PCs.

In order to exploit two-level parallelism, OpenMP is used in conjunction with MPI, which is a parallel programming standard for shared-memory mode, and consists of a set of compiler directives and a library of support functions [25]. One of the most popular approach on the hybrid OpenMP-MPI programming is to parallelize the loop sections of the MPI application using multithreading with the aid of OpenMP. Resorting this approach, the hybrid parallel FDTD simulation has been made on an SGI Origin 2000 parallel computer in the literature [26]. The framework of the hybrid algorithm is as follows:

Algorithm: Hybrid-FDTD-1

Do the initialization work, apply initial conditions;

for $t = 1$ ***to*** *tmax* **do**

for $i, j, k = 1$ ***to*** *imax, jmax, kmax* **do**

{

Using MPI message passing, exchange ***H***-field components with neighbors, and interpolate ***H***-field components at sub-domains boundaries;

Using OpenMP multithreading, #pragma omp parallel for

update ***E***-field components using ***H***-field components;

Using MPI message passing, exchange ***E***-field components with neighbors, and interpolate ***E***-field components at sub-domains boundaries;

Using OpenMP multithreading, #pragma omp parallel for

update ***H***-field components using updated ***E***-field components;

Using OpenMP multithreading, #pragma omp parallel for

update fields at boundaries, apply boundary conditions;

}

The principle of the algorithm is to insert OpenMP compiler directives (**#pragma omp parallel for**) into the nested loops which update EM fields in the traditional MPI-FDTD program. The loop iterations are split among threads, and go multithreaded. Thus, the computing time of the FDTD subspace is reduced. This approach is very simple, and requires hardly changing the structure of the existing MPI-FDTD parallel program. However, for the local area network with the high communication overheads, only achieving the second level parallelism involving loop iteration inside subspaces, the communication performance between subspaces cannot be improved. For the communication procedure of the parallel program goes still single-threaded, it can not be achieved availably to overlap computations with communications. When the grain size of the domain decomposition is decreased to the critical point, to shorten the computing time of the FDTD subspace will lead to the lower computation to communication ratio, which further causes the deterioration of the parallel salability and efficiency.

The FDTD subspace consists of the interior grids and the boundary grids exchanged. The communication takes place between subspaces, only involving the boundary grids. Therefore, the whole task of the FDTD computation can be divided into two relatively independent subtasks: 1) the iteration and interpolation of the exchanged grids at the subspace boundary and communication; 2) the iteration of the interior grids. Two subtasks can be parallelized by going OpenMP multithreaded. The different threads within PC can access the same memory pool, hence the communications of two subtasks take place without the costly overheads [27].

The framework of the modified algorithm is as follows:

Algorithm: Hybrid-FDTD-2

Do the initialization work, apply initial conditions;

for $t = 1$ ***to*** $tmax$ **do**

{

Using OpenMP multithreading, #pragma omp parallel sections

{

#pragma omp parallel section

{Computing and interpolating the exchanged fields at sub-domains boundaries;

Using MPI message passing, exchange fields with neighbors;**}**

#pragma omp parallel section

{Update ***E***-field components using ***H***-field components inside sub-domains;

Update ***H***-field components using ***E***-field components inside sub-domains;**}**

}

Update fields at boundaries, apply boundary conditions;

}

Unlike the pure data parallelization of hybrid-FDTD-1, hybrid-FDTD-2 introduces task parallelization at the basis of the data parallelization of the traditional MPI-FDTD application. With the use of OpenMP compiler directives (**#pragma omp parallel sections**), the computation and communication of the FDTD subspace are divide into two independent subtasks, which go multithreaded. Thus, the overlap of the computation and communication can be achieved by way of two-level parallelization of the data and tasks.

4. RESULTS AND DISCUSSION

4.1. Environment of Parallel FDTD Computing

The PC cluster, on which the proposed parallel FDTD will run, consists of 10 personal computers. Each computer in the cluster has a CPU of Pentium4 2.96G with hyper-threading technology and 1024MB DDR2 memory, interconnected by a 1000 Mbps high speed Ethernet network, and runs an operation system of Windows XP. The software support is composed of Microsoft Visual Studio .net 2005 and MPICH.NT 1.2.5. The former includes compilers and debuggers which support OpenMP, the latter provides MPI communication library.

4.2. Numerical Results

Figure 5 shows the radiation patterns of the automobile antenna ($\varepsilon_r = 5.5$) obtained by using three methods: 1) hybrid-FDTD-2; 2) serial FDTD; 3) measurement. There is a good agreement between calculations and measurements, which verifies the feasibility and correctness of the hybrid-FDTD-2. Besides, the equal results between hybrid-FDTD-2 and serial FDTD have been achieved, which

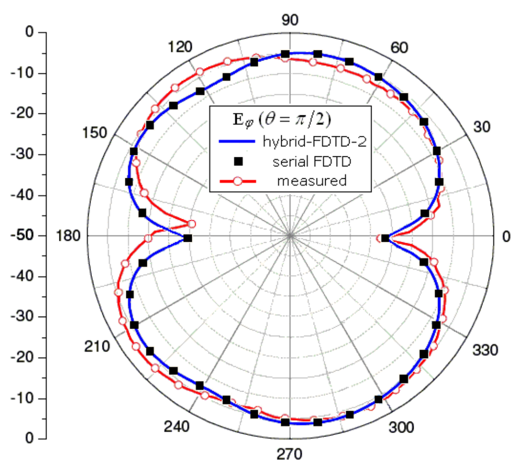

Figure 5. Calculated and measured radiation patterns at FM 100 MHz.

verifies the stability and accuracy of the interpolation scheme adopted in this example.

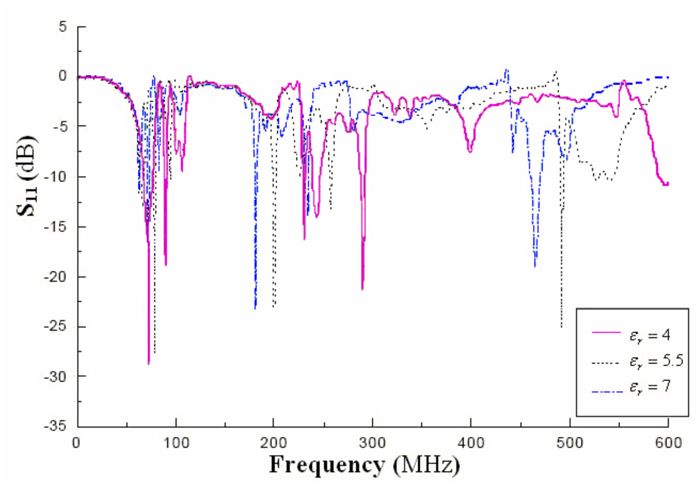

Figure 6. Comparison of the return loss (S_{11}) for the automobile antenna with different ε_r.

Figure 6 shows the simulated return loss S_{11} in three cases ($\varepsilon_r = 4$, $\varepsilon_r = 5.5$ and $\varepsilon_r = 7$). It is found that the different dielectric constants of the windshield will cause relatively obvious change to the return loss S_{11}. The operating frequency of the antenna will decrease as the dielectric constant increases, nevertheless, the difference is not distinct for the lower FM frequency band, and the -10 dB bandwidth can cover the majority of FM frequency range from 88 MHz to 108 MHz in three cases. For DAB (Digital Audio Broadcasting) frequency range from 174 MHz to 240 MHz, the -10 dB bandwidth cannot cover availably the operating frequency band in three cases, however, with the increase of dielectric constant, the electrical performance of the antenna, to certain extent, will improve. Therefore, according to the numerical results, the automobile antenna can effectively operate at FM frequency band, but require further optimizing at DAB frequency band.

4.3. Parallel Performance Analysis

We compare the performance of three programming methods for the parallel FDTD on PC cluster: 1) traditional MPI-FDTD; 2) hybrid-FDTD-1; 3) hybrid-FDTD-2.

To simplify the implementation of the algorithm, when the quantity of the FDTD subspaces is even number, the two-dimensional domain decomposition ($x \times 2 \times 1$) is adopted, and when the quantity of the subspaces is odd number, the one-dimensional domain decomposition ($x \times 1 \times 1$) is adopted. The data transference between subspaces utilizes the standard nonblocking mode based on MPI library.

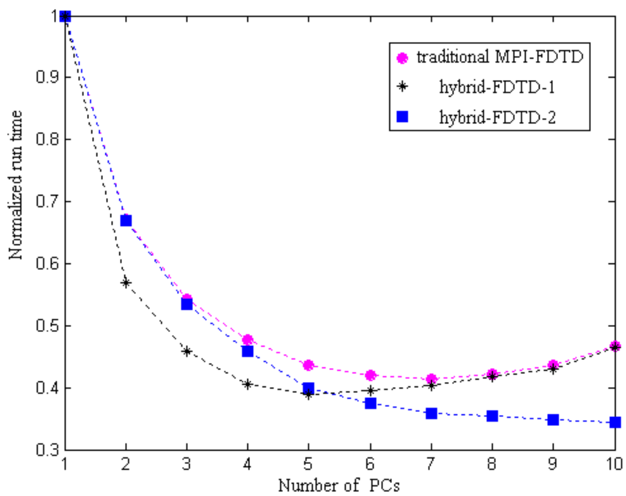

Figure 7. The normalized run time versus the number of PCs for three methods.

Figure 7 shows the benchmarking results of the normalized run time versus the number of PCs in three methods. For the fixed size problem in this paper, the traditional MPI-FDTD employing seven PCs reaches the parallel speedup limitation. Thus, the computation time, on the contrary, will increase as the computers employed are more in quantity than seven PCs. This happens because the PC cluster spends less time computing but essentially more time communicating between subspaces. The hybrid-FDTD-1 employing five PCs reaches the parallel speedup limitation. This is the same conclusion as discussed in Section 3, and happens because the second level parallelism only involving loop iteration can further reduce the computational time of each subspace as compared to the traditional MPI-FDTD on the equal number of PCs, whereas it cannot lower the communication overheads between subspaces, which causes the parallel program to reach more rapidly the speedup limitation. Unlike two algorithms as mentioned above, the speedup curve of the hybrid-FDTD-2 maintains still a relatively stable drop when the number of PCs is ten. This is because the overlap of the computation and communication lessens the influence of the network overheads, which brings about the result that the parallel performance depends on, to a great extent, the implementation of the overlapping operation. For the modern CPUs give directly support to thread-level parallelism from the hardware, the overlapping operation can be achieved availably by way of two level parallelization of the data and tasks. Therefore, the hybrid-FDTD-2 is well suited for the fine grained parallel FDTD problem on PC cluster.

The different domain decomposition mode will lead to the different volume of data transferred between subspaces, which brings about the different communication latency overheads. Therefore, the performance of the traditional MPI-FDTD has a close link with the virtual topology or Cartesian 3D topology of the domain decomposition, as analyzed in literature [28]. With eight PC nodes and different virtual topology schemes, we compare the parallel performance between the traditional MPI-FDTD and the hybrid-FDTD-2 in 1000 time steps. The virtual topology of subspaces are described as $x \times y \times z$ for all the cases: 1) $8 \times 1 \times 1$, 2) $1 \times 8 \times 1$, 3) $2 \times 2 \times 2$, 4) $4 \times 2 \times 1$, 5) $4 \times 1 \times 2$, 6) $2 \times 4 \times 1$. Figure 8 shows the results of our benchmarking.

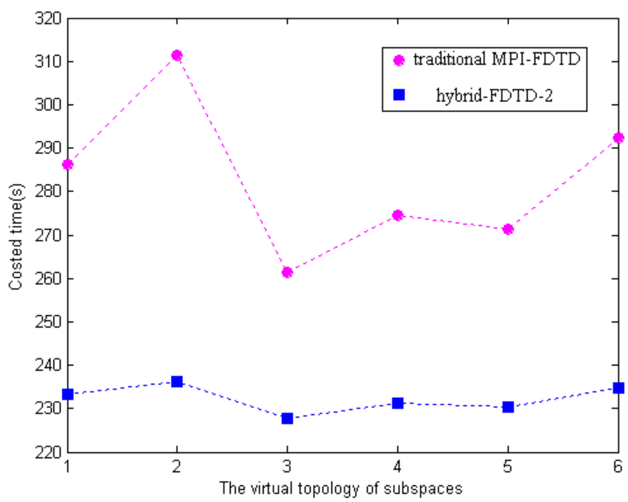

Figure 8. The costed time versus the subspaces virtual topology for two methods.

It can be seen from Figure 8 that the performance curve of the traditional MPI-FDTD has obvious oscillation, which indicates the parallel performance will change with the adopted virtual topology scheme. Thus, in order to improve the algorithm performance, the computation space is divided not only according to the features of the original problem, but also according to the optimum virtual topology. Nevertheless, the curve of the hybrid-FDTD-2 is relatively smooth as compared to the former, which indicates the overlap of the computation and communication brings about the case that the virtual topology of subspaces plays a small role in improving the parallel performance. Therefore, the computation space can be divided largely according to the problem features, which is a great convenience for the the implementation of the parallel FDTD algorithm.

5. CONCLUSIONS

In this paper, we study two-level parallel implementation of the FDTD algorithm on PC cluster, and develop a high performance MPI-OpenMP hybrid algorithm based on two-level parallelization of the data and tasks. The computed example of a printed antenna for automobile is given, and various comparisons are made to prove the efficiency and accuracy of this algorithm. The results show that with the decrease of the computational granularity on each computer, the novel algorithm is more efficient as compared to two traditional parallel FDTD algorithms on PC cluster, and moreover, it can also lessen the influence of the subspaces virtual topology on the parallel FDTD performance. As the multi-core CPUs which support directly multithreading from hardware are widely employed on modern local area network, this MPI-OpenMP hybrid approach is increasingly valuable to improve the efficiency of the parallel FDTD.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Fund (60571020) of P. R. China. The authors would like to thank Dr. X. Gao and Dr. Y. K. Sun for their useful discussions and suggestions during the development of the parallel FDTD system.

REFERENCES

1. Ding, W., Y. Zhang, P. Y. Zhu, et al., "Study on electromagnetic problems involving combinations of arbitrarily oriented thin-wire antennas and inhomogeneous dielectric objects with a hybrid MoM-FDTD method," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 11, 1519–1533, 2006.

2. Tang, L. and T. S. Ibrahim, "On the radio-frequency power requirements of human MRI," *PIERS Online*, Vol. 3, No. 6, 886–889, 2007.
3. Yang, D. K., X. Y. Hu, S. Y. Zhang, et al., "Numerical simulation for the effective conductivity of composite medium in high frequency," *PIERS Online*, Vol. 3, No. 4, 457–461, 2007.
4. Xu, J., P. Chen, Y. Shi, et al., "Numerical study on two-dimensional magnetic photonic crystals made of magnetized ferrites," *PIERS Online*, Vol. 3, No. 3, 305–307, 2007.
5. Wang, Q. Y., J. Wang, and S. L. Zhang, "Numerical simulation analysis of an optical virtual probe based on surface plasmon polaritonic band-gap structures," *PIERS Online*, Vol. 3, No. 1, 7–12, 2007.
6. Liu, Y., Z. Q. Yang, Z. Liang, et al., "A memory-efficient strategy for the FDTD implementation applied to the photonic crystals problems," *PIERS Online*, Vol. 3, No. 4, 374–378, 2007.
7. Gorodetsky, D. A. and P. A. Wilsey, "Reduction of FDTD simulation time with modal methods," *PIERS Online*, Vol. 2, No. 5, 510–513, 2006.
8. Huang, X. L., L. Xia, and H. Y. Chen, "Waveguide analysis using multiresolution time domain method," *PIERS Online*, Vol. 2, No. 6, 559–561, 2006.
9. Guiffaut, C. and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," *IEEE Antennas and Propagation Magazine*, Vol. 43, No. 2, 94–103, 2001.
10. Zhang, Y., J. Song, and C. H. Liang, "MPI based parallelized locally conformal FDTD for modeling slot antennas and new periodic structures in microstrip," *Journal of Electromagnetic Waves and Applications*, Vol. 18, No. 10, 1321–1335, 2004.
11. Yu, W., Y. Liu, T. Su, et al., "A robust parallel conformal finite-difference time-domain processing package using the MPI library," *IEEE Antennas and Propagation Magazine*, Vol. 47, No. 3, 39–59, 2005.
12. Chen, K., X. Chen, and K. Huang, "A novel microstrip dipole antenna with wideband and end-fire properties," *Journal of Electromagnetic Waves and Applications*, Vol. 21, No. 12, 1679–1688, 2007.
13. Chen, X., D. Liang, and K. Huang, "Microwave imaging 3-D buried objects using parallel genetic algorithm combined with FDTD technique," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 13, 1761–1774, 2006.

14. Zheng, K., D. Ge, and N. Ge, "Parallel FDTD computing for 3D EM scattering problem and its speedup factor analysis," *Chinese Journal of Radio Science*, Vol. 19, No. 6, 767–771, 2004.
15. Feng, J. and X. Cheng, "A study of parallel FDTD for simulating complex antennas on a cluster system," *Asia-Pacific Microwave Conference Proceedings*, 1607116, Suzhou, China, Dec. 4–7, 2005.
16. Niikura, K., R. Kokubo, K. Southisombath, et al., "On analysis of planar antennas using FDTD method," *PIERS Online*, Vol. 3, No. 7, 1019–1023, 2007.
17. Chai, W. W., X. J. Zhang, and J. B. Liu, "A novel wideband antenna design using U-slot," *PIERS Online*, Vol. 3, No. 7, 1067–1070, 2007.
18. Losito, O., "Design of conformal tapered leaky wave antenna," *PIERS Online*, Vol. 3, No. 8, 1316–1320, 2007.
19. Yee, K. S., J. S. Chen, and A. H. Chang, "Conformal finite-difference time-domain (FDTD) with overlapping grids," *IEEE Transactions on Antennas and Propagation*, Vol. 40, No. 9, 1068–1075, 1992.
20. Sha, W., X. L. Wu, and M. S. Chen, "A diagonal split-cell model for the high-order symplectic FDTD scheme," *PIERS Online*, Vol. 2, No. 6, 715–719, 2006.
21. Gedney, S. D., "An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 12, 1630–1639, 1996.
22. Xu, F. and W. Hong, "Domain decomposition FDTD algorithm for the analysis of a new type of E-plane sectorial horn with aperture field distribution optimization," *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 2, 426–434, 2004.
23. Sarkar, A., N. Benabbou, and R. Ghanem, "Domain decomposition of stochastic PDES and its parallel implementation," *20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment*, 17, St. John's, NF, Canada, May 14–17, 2006.
24. Liang, D., J. Feng, and X. Chen, "A study of efficient parallel FDTD methods on cluster systems," *Journal of Sichuan University (Natural Science Edition)*, Vol. 43, No. 3, 549–554, 2006.
25. Drosinos, N. and N. Koziris, "Performance comparison of pure MPI vs hybrid MPI-OpenMP parallelization models on SMP clusters," *18th International Parallel and Distributed Processing Symposium*, 193–202, Santa Fe, NM, USA, Apr. 26–30, 2004.

26. Su, M. F., I. El-Kady, D. A. Bader, et al., "A novel FDTD application featuring OpenMP-MPI hybrid parallelization," *Proceedings of International Conference on Parallel Processing*, 373–379, Montreal, Que, Canada, Aug. 15–18, 2004.
27. Akhter, S., J. Roberts, J. Reinders, et al., *Multi-Core Programing Increasing Performance through Software Multi-threading*, Intel Press Business Unit, Hillsboro, 2004.
28. Zhang, Y., W. Ding, and C. H. Liang, "Study on the optimum virtual topology for MPI based parallel conformal FDTD algorithm on PC clusters," *Journal of Electromagnetic Waves and Applications*, Vol. 19, No. 13, 1817–1831, 2005.