

## Research Article

# Study on High Performance of MPI-Based Parallel FDTD from WorkStation to Super Computer Platform

**Z. L. He, K. Huang, Y. Zhang, Y. Yan, and C. H. Liang**

*Science and Technology on Antenna and Microwave Laboratory, XiDian University, Xi'an 710071, China*

Correspondence should be addressed to Z. L. He, 305111889@qq.com

Received 15 June 2011; Revised 12 September 2011; Accepted 25 October 2011

Academic Editor: Jun Hu

Copyright © 2012 Z. L. He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Parallel FDTD method is applied to analyze the electromagnetic problems of the electrically large targets on super computer. It is well known that the more the number of processors the less computing time consumed. Nevertheless, with the same number of processors, computing efficiency is affected by the scheme of the MPI virtual topology. Then, the influence of different virtual topology schemes on parallel performance of parallel FDTD is studied in detail. The general rules are presented on how to obtain the highest efficiency of parallel FDTD algorithm by optimizing MPI virtual topology. To show the validity of the presented method, several numerical results are given in the later part. Various comparisons are made and some useful conclusions are summarized.

## 1. Introduction

The Finite Difference Time Domain (FDTD) method, introduced by Yee, in 1966 [1], is one of the most popular three-dimensional methods in computational electromagnetics. The FDTD method has been applied to many radar cross section (RCS) calculations with accurate results. However, as a powerful numerical technique, the FDTD method is restrained to computation resource when analyzing the scattering of electrically large targets. To overcome the problem, a Parallel FDTD algorithm using the Message Passing Interface (MPI) library was developed by Volakis et al. in 2001 [2]. It is easy to implement because the Yee scheme is explicit. The FDTD in Cartesian coordinates can be easily divided into many subspaces, and each computer in a parallel system deals with one or several subdomains. The FDTD algorithm is combined with the MPI to run on parallel system. The MPI functions are employed to exchange the tangential electric (magnetic) fields on the boundary of the subdomain among the adjacent neighbors [3–5]. Parallel computation of the E-H components with an MPI Cartesian 2D topology is adopted and has been explained step by step in [6]; as the authors of it addressed, it is the first paper on Parallel FDTD using MPI protocol. Zhang et al. used an MPI Cartesian 3D topology in his research [7]. As an

extension of these researches, we have developed a parallel FDTD code using an MPI Cartesian 3D topology. It has been successfully run on PC cluster [5, 6] and blade server [8], both of which belong to Science and Technology on Antenna and Microwave Laboratory in China.

Now, the code is developed to solve larger-scale electromagnetic problems. It is applied on super computer based on Linux system, which belongs to Shanghai Supercomputer Center of China (SSC). Numerical examples prove that the virtual topology will affect the computational efficiency of Parallel FDTD severely. In this paper, the influence of different virtual topology schemes on parallel performance of Parallel FDTD will be studied in detail. The general rules are presented on how to obtain the highest efficiency of Parallel FDTD algorithm by optimizing MPI virtual topology.

In Section 2, parallel FDTD algorithm is presented briefly. In Section 3, some numerical results are given, which show that this method is efficient and accurate. Current distribution over the surface of scatter and the near field distribution are plotted. Discussions about the influence of different virtual topology schemes and different number of processors on parallel performance of Parallel FDTD are presented in Section 4. Finally, some useful conclusions are summarized.

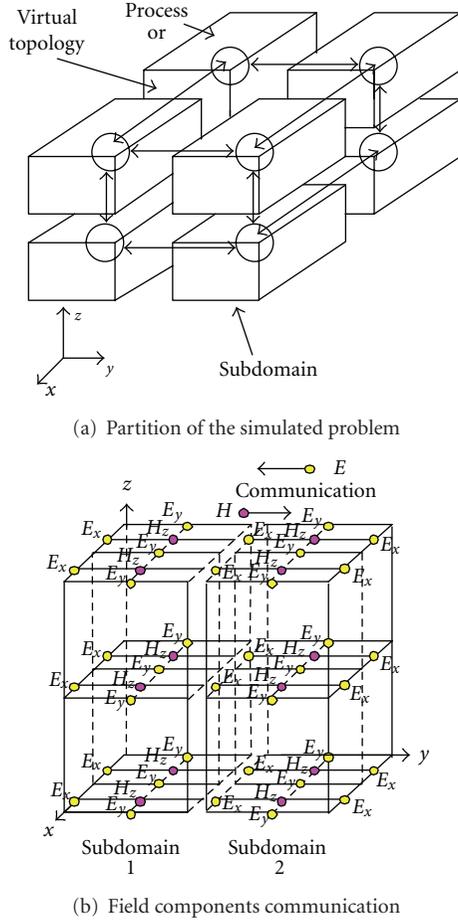
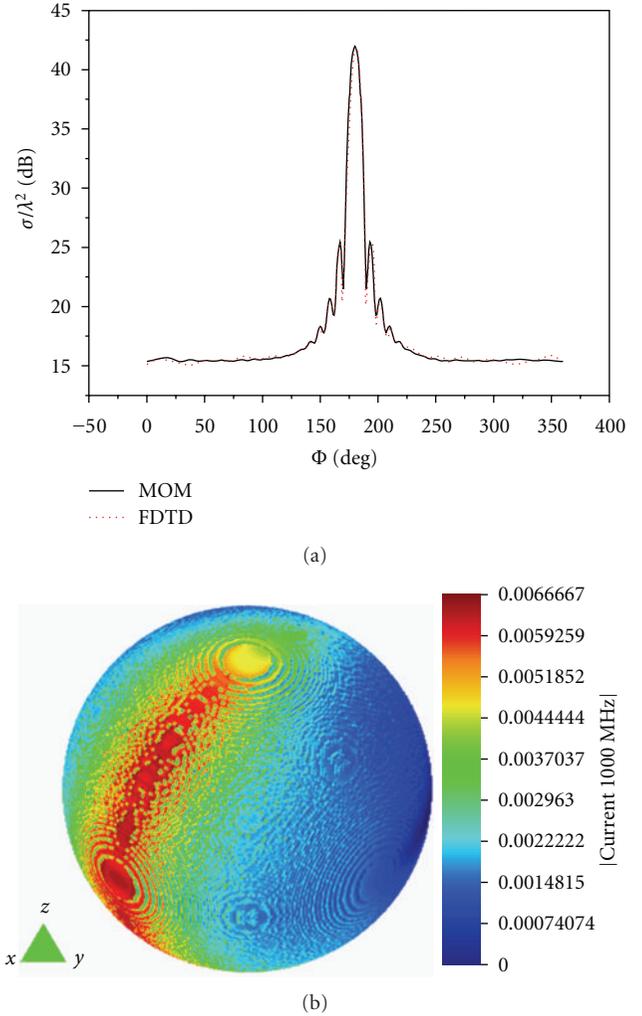


FIGURE 1: Division and communication of Parallel FDTD in 3D.

## 2. Parallel Algorithm

MPI was proposed as a standard by a broadly based committee of vendors, implementers, and users. Now, it becomes a definition for interfaces among a cluster of computers or the processors of a multiprocessor parallel computer. The key problem that MPI-based programming relates to is how to distribute the tasks of users to processors according to the capability of each processor and reduce the communications among processors to as little as possible. Reducing the communications is especially crucial as the speed of communication is far slower than that of computation.

FDTD is easy to implement because the Yee scheme is explicit. Besides, it has the principle advantage that since the grid is regular and orthogonal, electromagnetic field components are easily indexed by  $(i, j, k)$  in Cartesian coordinates. The parallelism of the FDTD algorithm is based on a spatial decomposition of the simulation problem geometry into contiguous nonoverlapping rectangular subdomains. The computational space can be easily divided into nearly equal parts along the three directions, and each processor in a parallel system deals with one or several subdomains, as shown in Figure 1(a). The virtual topology of the processors' distribution is chosen in a similar shape as

FIGURE 2: (a) Comparison of the RCS results in  $xoy$  plane obtained by FDTD and MOM; (b) current distribution over the surface of the sphere.

the problem partition. Each subdomain is mapped to its associated node where all the field components belonging to this subdomain are computed. To update field values lying on interfaces between sub-domains, it is necessary to exchange data between neighboring processors. An  $x$ - $z$  slice of the computational volume at one of the interfaces between nodes in the  $y$ -direction is shown in Figure 1(b). In this paper, we adopt 3D communication pattern, which is introduced as follows.

The FDTD algorithm is combined with MPI to run on parallel system. The MPI functions are employed to exchange the tangential electric (magnetic) fields on the boundary of the subdomain among the adjacent neighbors.

The parallel algorithm can be described as follows.

(1) Initialization.

(a) MPI Initialization.

(b) Reading the modeling parameters from the input files.

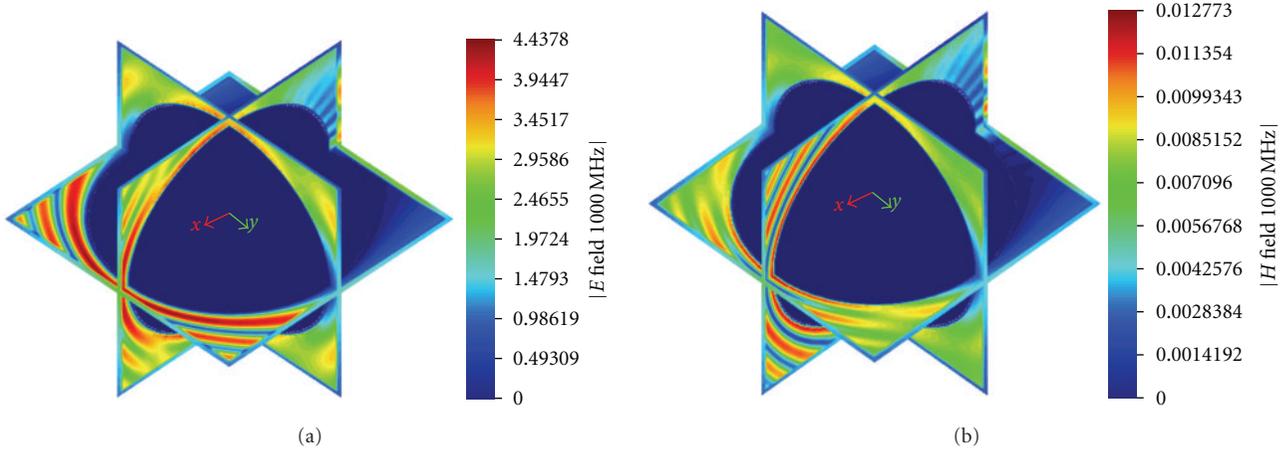


FIGURE 3: Smooth contour fill of near field distribution in frequency domain on three planes:  $x = 0.0$ ,  $y = 0.0$ , and  $z = 0.0$ : (a) E field distribution and (b) H field distribution.

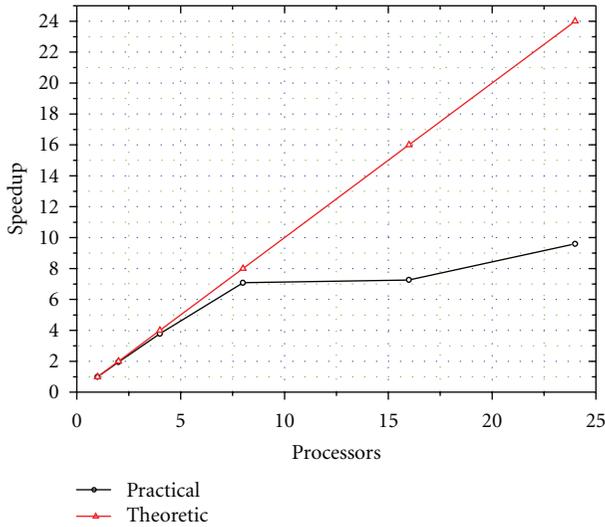


FIGURE 4: Parallel efficiency.

- (c) Creation of the three-dimensional Cartesian topology.
  - (d) Creation of the derived data types for communication.
  - (e) Start time measurements.
  - (f) Allocation of memory.
  - (g) Setting all field components to zero.
- (2) At each time step.
- (a) Exciting source only on processors that include the source plane.
  - (b) Calculation of new magnetic field components on each processor.
  - (c) Communication of the magnetic field components between processors.

- (d) Calculation of new electric field components on each processor.
  - (e) Communication of the electric field components between processors.
  - (f) Calculation of transmission only on processors that include the transmission plane.
  - (g) Collection of field variables only on processors that include detection points.
- (3) Reducing the transmission to a fixed processor and writing it on file.
  - (4) Saving results in files.
  - (5) Deallocation of memory.
  - (6) Stop time measurement.
  - (7) MPI Finalization.
  - (8) End.

### 3. Hardware Platform

- (i) *Think Station*. Its machine-type model is 4155-D43 with a total of 24 Intel(R) Xeon(R) X5650 CPU cores (2.67 GHz per CPU) and a total of RAM approximately equal to 64 GB.
- (ii) *Shanghai Supercomputer Center (SSC)*. The 37 nodes from Magic-cube Machine with a total of 512 AMD CPU cores (1.9 GHz per CPU and 4 cores on each CPU) 16 CPU cores on each node and 4 GB RAM per core, and a total of RAM approximately equal to 2.3TB. Infiniband is used for the network interconnection.

### 4. Numerical Results

For the absorbing UPML medium, we use a thickness of 5 cells in the following examples.

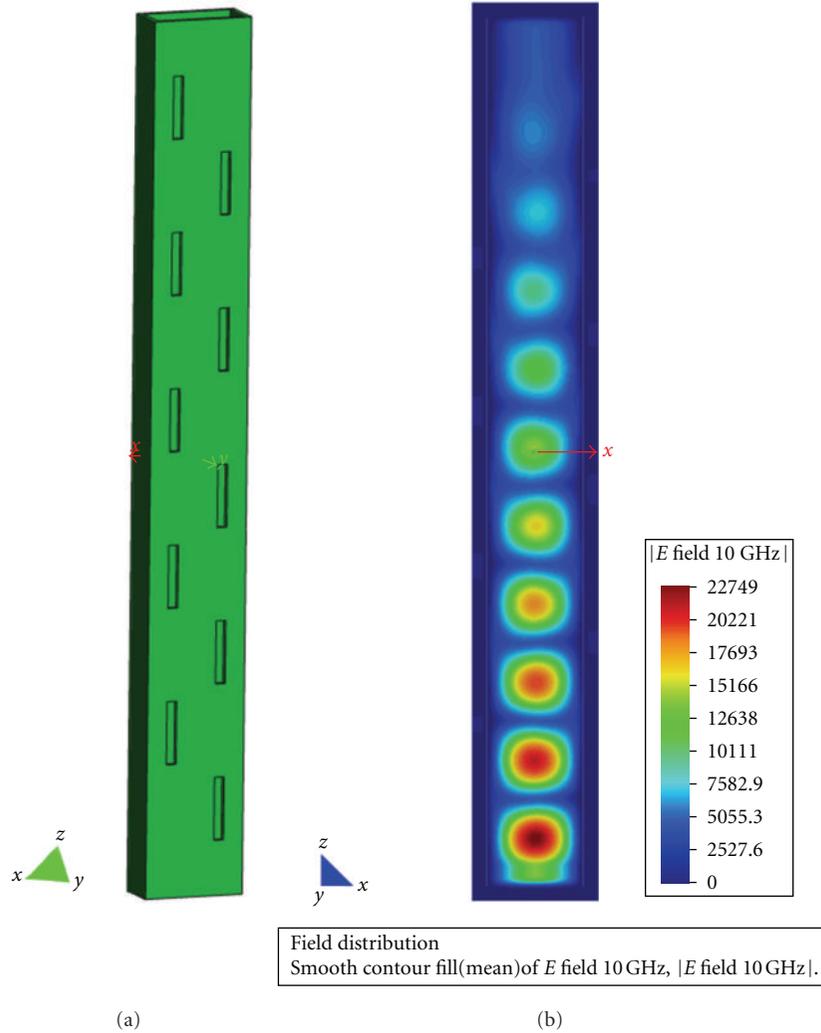


FIGURE 5: (a) Model of the waveguide with ten slots; (b) smooth contour fill of E field distribution of the  $xoz$  plane.

**4.1. An Example for Validation.** For validation, the bistatic RCS is calculated for a PEC sphere with a 1 m ( $10\lambda/3$ ) radius, and the incident plane wave is arriving from the  $\geq x$  axis and the polarization is along  $-z$  axis. Working frequency is 1.0 GHz. The increment  $dx = dy = dz = 0.02$  m ( $\lambda/15$ ) is used here, and the amount of FDTD grids is  $144 \times 144 \times 144$ . The bistatic RCS of the sphere is shown in Figure 2(a). The result is compared with the one obtained by the Moment of Method (MOM), which shows a good agreement between them. Current distribution over the surface of the sphere is given in Figure 2(b). In Figures 3(a) and 3(b), smooth contour fills of the amplitude of E field and H field distributions in frequency domain are plotted, respectively. This problem is calculated on *Think Station*. Total computation time (in seconds) with different number of processors and different virtual topology schemes in *1000 time-steps* are compared in Table 1.

In Table 1, virtual topology schemes are described as  $(x \times y \times z)$  for all three communication patterns. If the value is 1 in some direction, it implies that there is no topology in this direction. For example,  $1 \times 4 \times 1$  means there is no

TABLE 1: Comparisons of computation time.

Cores	Virtual topology ( $x \times y \times z$ )	Computation time (in seconds)
1	$1 \times 1 \times 1$	646.25
2	$2 \times 1 \times 1$	330.75
4	$4 \times 1 \times 1$	172.25
4	$2 \times 2 \times 1$	170.88
8	$8 \times 1 \times 1$	98.38
8	$4 \times 2 \times 1$	92.75
8	$2 \times 2 \times 2$	91.25
16	$8 \times 2 \times 1$	95.12
16	$4 \times 4 \times 1$	91.38
16	$4 \times 2 \times 2$	89.00
24	$8 \times 3 \times 1$	69.38
24	$4 \times 3 \times 2$	67.38

topology in  $x$  and  $z$  directions, respectively, thus the virtual topology is actually in one dimension. Similarly,  $1 \times 2 \times 2$  means there is no topology in  $x$  direction, thus the virtual topology is actually in two dimensions.

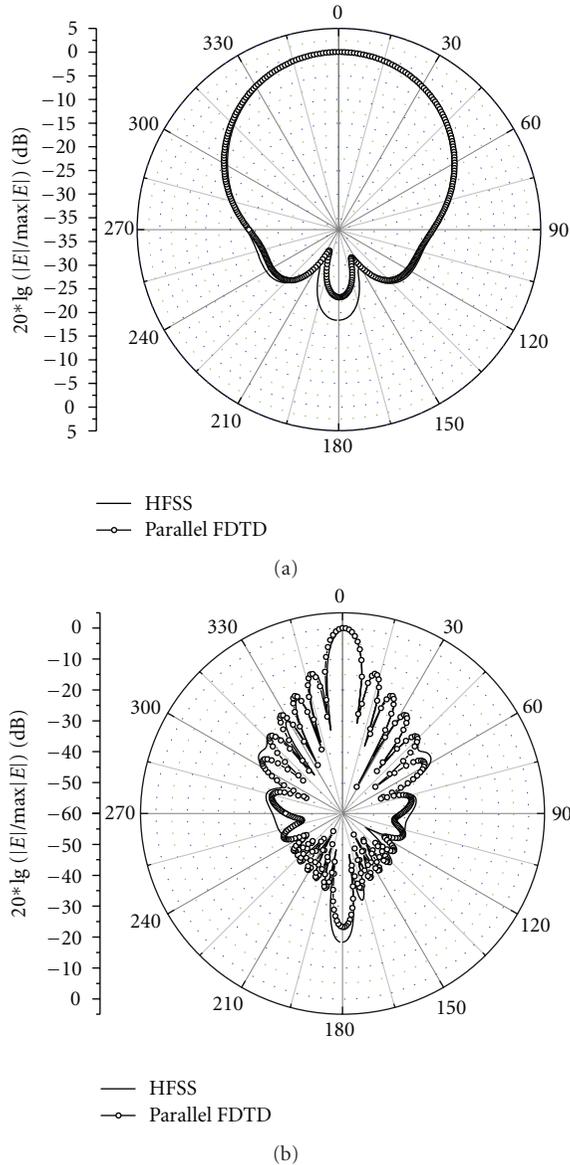


FIGURE 6: Results of radiation: (a) radiation pattern of the  $xoy$  plane and (b) radiation pattern of the  $yoz$  plane.

From Table 1, it is obvious that increasing the number of processors can bring us the reduction of the computation time rapidly. But different virtual topology schemes will cost different computation time even if the code is run with the same number of processors.

We discuss the parallel performance of the Parallel FDTD using the different dimensional virtual topology with the same number of processors. In this case, the computing time is calculated with the shortest time as the reference. Take the case of eight processors as the example. The reference calculating time is 91.25 seconds using the  $2 \times 2 \times 2$  virtual topology scheme:

$$\begin{aligned} \text{Comparison: } & 8 \times 1 \times 1(98.38 - 91.25) = 6.13 \text{ (sec)}, \\ & 4 \times 2 \times 1(92.75 - 91.25) = 1.50 \text{ (sec)}. \end{aligned} \quad (1)$$

From above, it is obvious that for the same number of processors, the more the dimension of the virtual topology, the less the computation time required.

Parallel efficiency is shown in Figure 4, in which the computing time consumed by different number of processors is referred as the shortest time for each case. Parallel efficiency is decreased with the increasing of processors. That is because the amounts of the transferred data are increased with processors, and then the time consumed on communication is increased.

In addition, the topology along the direction where the amount of the FDTD grids is larger can save the computation time for the same dimensional virtual topology. Different division subdomains with the same dimensional virtual topology lead to different amounts of the transferred data. Expression for the total number of the grids lay on interface between processors is

$$(nx - 1) \times b \times c + (ny - 1) \times a \times c + (nz - 1) \times a \times b, \quad (2)$$

where  $a, b, c$  represent the total number of grids in  $x, y, z$  direction, respectively, and  $nx, ny, nz$  are the values of virtual topology in three directions.  $nx, ny,$  and  $nz$  are integers and should satisfy the condition:

$$nx \times ny \times nz = \text{Processors}, \quad nx \geq 1, ny \geq 1, nz \geq 1. \quad (3)$$

So, the topology scheme should be created along the directions where the amount of the FDTD grids is larger to decrease the amounts of the transferred data, then to save the communication time.

Till now, the general rules on how to obtain the highest efficiency of Parallel FDTD algorithm by optimizing MPI virtual topology can be drawn as follows.

- (1) If possible, the optimum virtual topology scheme should be created in three dimensions, and then the better is in two dimensions, which can bring us higher efficiency than in one dimension.
- (2) As to the same dimensional virtual topology, the topology scheme should be created along the directions where the amount of the FDTD grids is larger.

**4.2. Radiation of the Waveguide with Ten Slots.** A waveguide with ten slots is analyzed by parallel FDTD. The dimension of the waveguide and the slot structure in this example are chosen as follows: the thickness of the waveguide wall is 1.27 mm, the length of the slot is 15.785 mm, the width of the slot is 2.54 mm, and all of the offsets of the slots are 6.35 mm. Its FDTD model is shown in Figure 5(a). Its working frequency is 10 GHz. The increment  $dx = dy = dz = 0.635$  mm is used here, and the total amount of FDTD grids is  $70 \times 50 \times 384$ . We introduce a sinusoid ally modulated Gaussian pulse excitation by modifying the

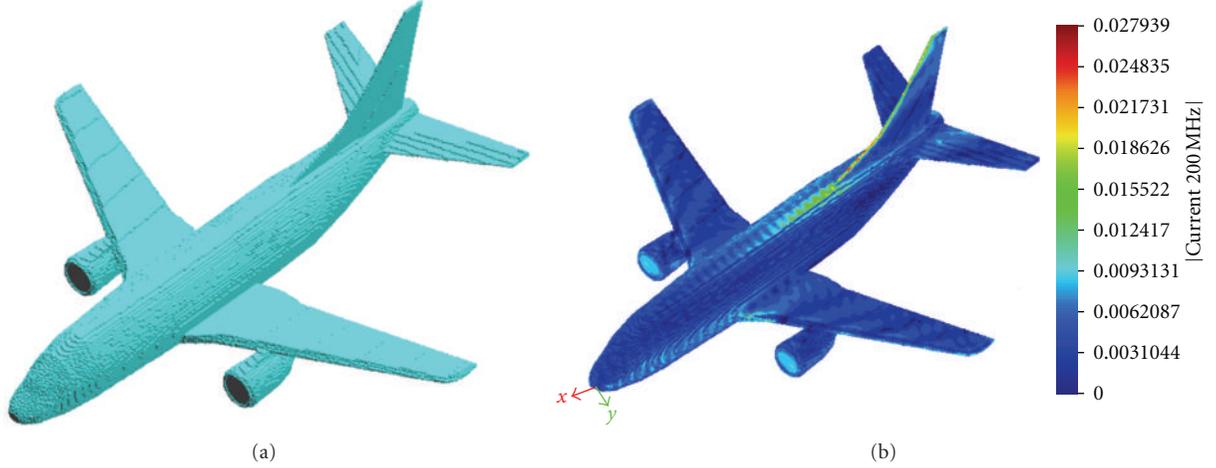


FIGURE 7: (a) Cartesian Mesh of the airplane and (b) smooth contour fill of the amplitude of surface inductive current on the airplane surface.

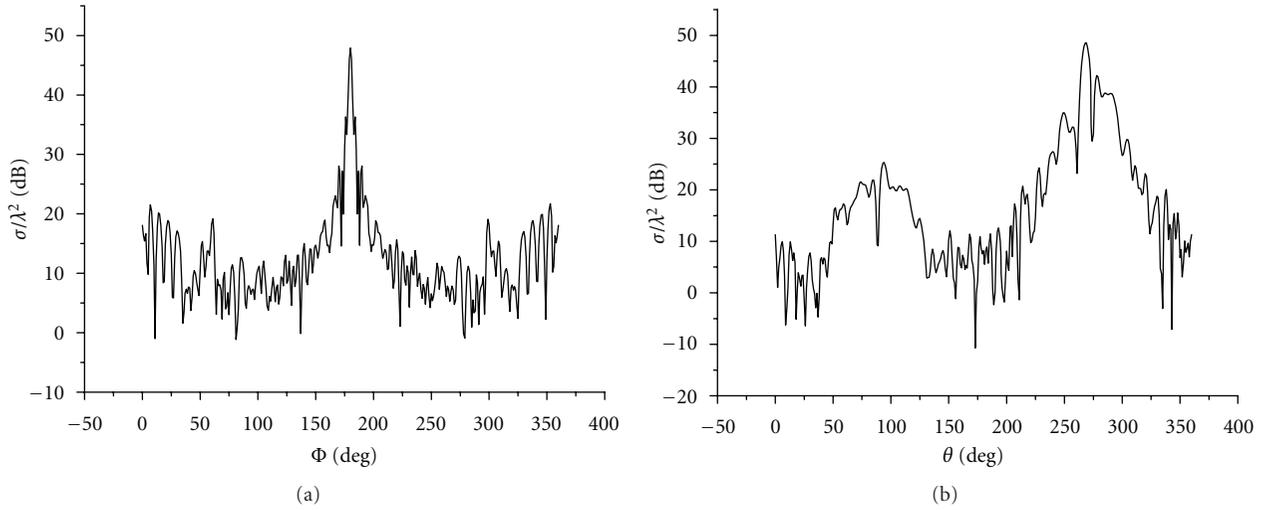


FIGURE 8: RCS of the airplane. (a) E-plane ( $xoy$ ). (b) H-plane ( $xoz$ ).

updating equation for the component in the excitation plane as follows:

$$E_y^{n+1}(i, j, k) = g(n\Delta t) + E_y^n(i, j, k) + \frac{\Delta t}{\epsilon_y(i, j, k)} \cdot \left[ \frac{H_x^{n+1/2}(i, j, k+1) - H_x^{n+1/2}(i, j, k)}{\Delta z} - \frac{H_z^{n+1/2}(i+1, j, k) - H_z^{n+1/2}(i, j, k)}{\Delta x} \right], \quad (4)$$

where  $g(n\Delta t) = \sin(\pi x/a) \sin(2\pi f_0 t) e^{-((t-t_0)/\sigma)^2}$ . By properly setting  $f_0$ ,  $t_0$ , and  $\sigma$ , we can get a useful frequency bandwidth. Finally, smooth contour fill of electric field distribution in frequency domain on plane  $y = 0.0$  is shown in Figure 5(b). Results of the radiation patterns obtained by parallel FDTD agree excellent with the ones obtained by HFSS shown in Figures 6(a) and 6(b). Total computation time (in seconds)

TABLE 2: Comparisons of computation time.

Cores	Virtual topology ( $x \times y \times z$ )	Total grids lay on interface between processors	Computation time (in seconds)
8	$1 \times 1 \times 8$	24500	121.09
8	$1 \times 2 \times 4$	37380	124.56
8	$2 \times 1 \times 4$	29700	116.09
8	$2 \times 2 \times 2$	49580	118.00

with eight processors and different virtual topology schemes in 1000 time-steps is compared in Table 2.

From Table 2, the shortest computing time consumed by virtual topology in two dimensions is shorter than in one dimension. However, three-dimensional virtual topology is not better than two-dimensional virtual topology. As shown

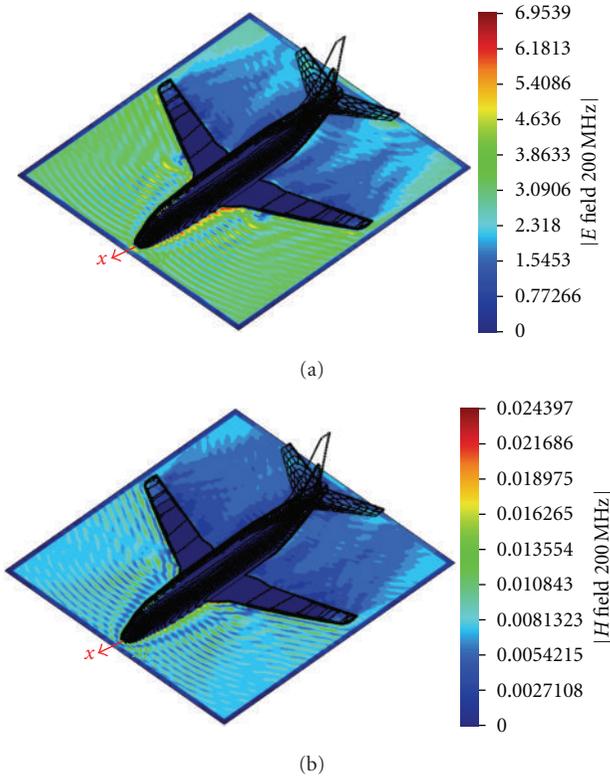


FIGURE 9: Smooth contour fill of near-field distribution in frequency domain on  $z = 0.05$  plane, (a) E field distribution, and (b) H field distribution.

TABLE 3: Comparisons of computation time.

Cores	Virtual topology ( $x \times y \times z$ )	Computation time (in seconds)	Total grids lay on interface between processors
512	$8 \times 8 \times 8$	896.00	2479680
512	$8 \times 16 \times 4$	3968.00	3000640

in Table 2, the transferred data for the  $2 \times 2 \times 2$  virtual topology scheme is much more than the one for the  $2 \times 1 \times 4$  case. Thus, as to the same dimensional virtual topology, the topology scheme should be created along the directions where the amount of the FDTD grids is larger.

**4.3. Analysis of the Scattering of an Airplane.** Then we analyze the scattering of a perfectly conducting airplane whose FDTD model is shown in Figure 7(a). Its working frequency is 200 MHz. The increment  $dx = dy = dz = 0.08$  m is used here. The direction of the incident wave is  $-x$  and the polarization is along  $+y$ . Inductive current distribution over the airplane surface is given in Figure 7(b). Figure 8 presents the bistatic RCS of the conducting airplane obtained by using Parallel FDTD. The frequency domain near-field distribution on the plane  $z = 0.05$  is given in Figure 9.

This example is calculated on super computer with 512 cores, which belong to the SSC. Time consumed by two virtual topology schemes with the same number of cores

are listed in Table 3. Total amount of the FDTD grids is  $440 \times 416 \times 200$ . It is obvious that the results accord with the rules presented before. With the same dimensional virtual topology, the parallel performance is mainly affected by the amounts of the transferred data especially in large-scale problems.

## 5. Conclusions

In this paper, parallel FDTD method is applied to analyze the scattering of the electrically large targets. The code we developed is successfully run on super computer in Shanghai Supercomputer Center of China (SSC). The influence of different virtual topology schemes on the parallel performance of Parallel FDTD is studied in depth and in detail. The results show that the computation time efficiency can be improved by properly choosing MPI virtual topology schemes. Following the two conclusions above, we can obtain the highest computational efficiency.

## Acknowledgments

This work is partly supported by the Fundamental Research Funds for the Central Universities of China (JY10000902002 and K50510020017) and the National Science Foundation of China (61072019). This work is also supported by Shanghai Supercomputer Center of China (SSC).

## References

- [1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. AP-14, pp. 302–307, 1966.
- [2] J. L. Volakis, D. B. Davidson, C. Guiffaut, and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 94–103, 2001.
- [3] V. Varadarajan and R. Mittra, "Finite-difference time-domain (FDTD) analysis using distributed computing," *IEEE Microwave and Guided Wave Letters*, vol. 4, no. 5, pp. 144–145, 1994.
- [4] A. Alighanbari and C. D. Sarris, "Parallel time-domain full-wave analysis and system-level modeling of ultrawideband indoor communication systems," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 1, pp. 231–240, 2009.
- [5] S. Prakash, E. Deelman, and R. Bagrodia, "Asynchronous parallel simulation of parallel programs," *IEEE Transactions on Software Engineering*, vol. 26, no. 5, pp. 385–400, 2000.
- [6] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 2nd edition, 2000.
- [7] Y. Zhang, 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, pp. 1321–1335, 2004.
- [8] U. Andersson, *Time-domain methods for the Maxwell equations*, Doctoral Dissertation, Royal Institute of Technology, Sweden, 2001.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

