Hindawi Publishing Corporation
International Journal of Antennas and Propagation
Volume 2015, Article ID 690510, 10 pages
http://dx.doi.org/10.1155/2015/690510

Research Article

An Optimized Parallel FDTD Topology for Challenging Electromagnetic Simulations on Supercomputers

Shugang Jiang, Yu Zhang, Zhongchao Lin, and Xunwang Zhao

School of Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071, China

Correspondence should be addressed to Shugang Jiang; zaishuiyifang131@126.com

Received 23 March 2015; Accepted 14 May 2015

Academic Editor: Giuseppe Mazzarella

Copyright © 2015 Shugang Jiang 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.

It may not be a challenge to run a Finite-Difference Time-Domain (FDTD) code for electromagnetic simulations on a supercomputer with more than 10 thousands of CPU cores; however, to make FDTD code work with the highest efficiency is a challenge. In this paper, the performance of parallel FDTD is optimized through MPI (message passing interface) virtual topology, based on which a communication model is established. The general rules of optimal topology are presented according to the model. The performance of the method is tested and analyzed on three high performance computing platforms with different architectures in China. Simulations including an airplane with a 700-wavelength wingspan, and a complex microstrip antenna array with nearly 2000 elements are performed very efficiently using a maximum of 10240 CPU cores.

1. Introduction

The principle of FDTD is that the calculation region is discretized by Yee grid to make the components of **E** and **H** be distributed at time and space alternately [1]. Then, there are four **H** (or **E**) components around each **E** (or **H**) component. This character makes the algorithm parallel in nature and using this the Maxwell equation can be transferred to a set of difference equations. The electromagnetic fields can be solved at time axis step by step. Then the electromagnetic fields distribution in each time step later can be obtained by the original values and boundary condition [2].

Researches on MPI based parallel FDTD for simulating complicated models have been published in the past decade. In 2001, Volakis et al. presented a parallel FDTD algorithm using the MPI library, where they raised an MPI Cartesian 2D topology [3]. Andersson developed parallel FDTD with 3D MPI topology in the same year [4]. In 2005, the authors studied the optimum virtual topology for MPI based parallel conformal FDTD algorithm on PC clusters [5–7]. In 2008, Yu et al. tested the parallel efficiency of the parallel FDTD [8] on BlueGene/L Supercomputer successfully and gave a parallel efficiency of 4000 cores under the case of balanced loads.

Although there are many publications on parallel FDTD, few of them involve parallel FDTD simulations utilizing more than 10000 cores. Most of the papers focused on load balancing when parallel efficiency was concerned, in addition to a more precise rule to achieve the best performance that needs to be given, especially in the case of simulations using tens of thousands of CPU cores on supercomputers.

With these concerned, in this paper, the influence of different virtual topology schemes on parallel performance of FDTD is studied through a theory model analysis. Then some tests are made on National Supercomputer Center in Tianjin (NSCC-TJ) and National Supercomputing Center in Shenzhen (NSCC-SZ) to verify the feasibility of theory. With the proposed theory model, some electrically large problems whose parallel scales up to 10240 cores are provided in this paper. And the parallel efficiency is nearly 80% when 10240 cores of SSC were utilized for an array with nearly 2000 elements. To the best of our knowledge, the proposed method achieves one of the best efficiencies ever reached using more than 10 thousands of CPU cores.

2. Computation Resources from Supercomputers

The program is tested on different clusters in three supercomputer centers, National Supercomputer Center in Tianjin (NSCC-TJ) [9], National Supercomputing Center in

Table 1: Parameters of computation resources.

Platform	CPU	Memory/node	Clock speed	Cores/node	Total cores used in this paper
SSC	AMD 8347HE 64-bit, four-core	64 G	1.9 GHz	16	8000
		32 G			4800
NSCC-TJ	Intel Xeon, 5670 six-core	24 G	2.93 GHz	12	120
NSCC-SZ	Intel Xeon, 5650 six-core	24 G	2.56 GHz	12	512

Shenzhen (NSCC-SZ) [10], and Shanghai Supercomputer Center (SSC) [11]. The parameters of computation resources in this paper are listed in Table 1.

3. Communication Model for Parallel FDTD

Communication is the main factor which affects the parallel performance of parallel codes. Therefore, reducing the amount of communication in FDTD by adjusting the virtual topology is selected as the optimization target.

Assume that the communication time in one time step is

$$T = \alpha C + \beta \cdot 2L, \tag{1}$$

where α is communication delay time, C is communication number, β is transmission speed, and L is the communication data amount of **E** or **H**. The calculated equation of each parameter is as follows:

$$C = 6P_xP_yP_z - 2\left(P_xP_y + P_yP_z + P_zP_x\right), \tag{2}$$

$$L = \left(P_x - 1\right)N_yN_z + \left(P_y - 1\right)N_zN_x + \left(P_z - 1\right)N_xN_y, \tag{3}$$

where P_x, P_y, and P_z are the topology values in three directions and N_x, N_y, and N_z are the grids number in x, y, and z directions.

From (1), it is known that when the total communication data amount is the same, the different topology scheme may bring the different communication number C, which comes to different total time T.

Take Dawning 5000A as example, the parameters α = 1.8 us~2.5 us and β = 1/(1.6563 Gb/s) [12]. Assume that the total grids are $1000 \times 1000 \times 1000$ and total cores are 1000, now the total communication delay time (9.72 ms) is about less an order of magnitude than the total communication time (121 ms). Under this cores scale, the communication delay time is the secondary factor.

The communication amount of single process is

$$\begin{aligned} \text{ave} &= \frac{L}{P_xP_yP_z} \\ &= \frac{\left(P_x - 1\right)N_yN_z + \left(P_y - 1\right)N_zN_x + \left(P_z - 1\right)N_xN_y}{P_xP_yP_z}. \end{aligned} \tag{4}$$

Divided by a constant $N_x \cdot N_y \cdot N_z$, (4) will be

$$\begin{aligned} \text{ave}' &= \frac{\left(P_x - 1\right)N_yN_z + \left(P_y - 1\right)N_zN_x + \left(P_z - 1\right)N_xN_y}{P_xP_yP_z \cdot N_xN_yN_z} \\ &= \frac{1}{P_xP_yP_z} \cdot \left(\frac{P_x - 1}{N_x} + \frac{P_y - 1}{N_y} + \frac{P_z - 1}{N_z}\right). \end{aligned} \tag{5}$$

From (5), it is known when and only $(P_x - 1)/N_x = (P_y - 1)/N_y = (P_z - 1)/N_z$, namely, the topology conformal as calculation region, the communication amount of single process is the least. Generally, the equation above cannot be satisfied absolutely. So the topology distribution is required that it is divided along the direction which is conformal as calculation region as possible to make (5) the least.

Generally speaking, the communication time is less between processes in one node than that between processes which belong to different nodes [12, 13]; that is, the one byte data communication time factor β is different between processes in one node and across nodes. So, when the factors C and L are the same between two different topologies, the communication amount across nodes is needed to be considered.

For certain grids, the total memory requirement (called M) keeps the same for different topologies. The memory distribution of each process (called m) is as follows:

$$m = \frac{M}{P_xP_yP_z}, \tag{6}$$

Equation (6) indicates that the memory distribution of each process is unrelated to the virtual topologies.

From the analysis above, it is known that the communication surface area varies from different virtual topology scheme for certain grids. The communication time will be changed associated with the virtual topology scheme while the memory distribution of each process remains the same. Thus, the communication amount is the main factor which affects the parallel performance.

4. Discussions on Parallel Performance

4.1. Simulation Model. Based on the theory above, a four-element microstrip antenna array is used as the model for benchmark. The parallel FDTD code is run to analyze the

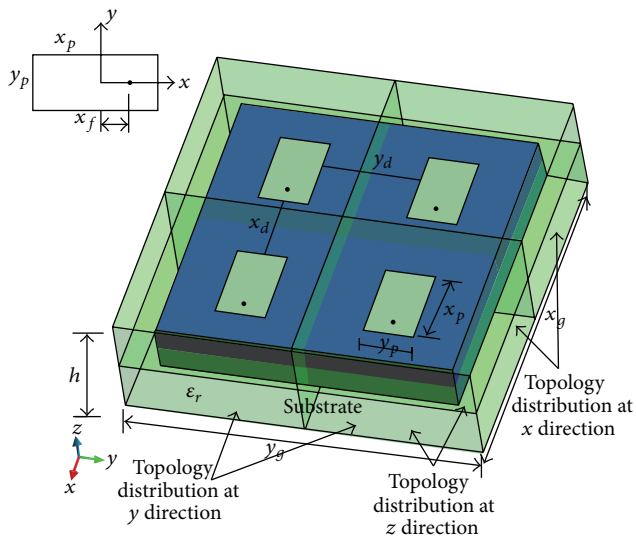


Figure 1: 2 × 2 microstrip array.

virtual topology schemes on two supercomputer center platforms, National Supercomputer Center in Tianjin (NSCC-TJ) and National Supercomputing Center in Shenzhen (NSCC-SZ), as listed in Table 1.

The array model is shown in Figure 1. The parameters of this array are as follows: Central frequency is 4.97 GHz, x_p = 14 mm, y_p = 9.6 mm, x_d = 15 mm, y_d = 15 mm, ε_r = 4.34, h = 0.8 mm, x_g = 60 mm, y_g = 60 mm, and x_f = 3.6 mm. The size of grid is $dx = dy = dz = 0.4$ mm.

Actually, the amount of total grids is just 200 × 200 × 50. However, to test the influence of different virtual topology schemes on parallel performance of parallel FDTD, the computational space needs to be extended. So, in this test, the amount of total grids is set as 1200 × 1200 × 300.

The radiation patterns of the microstrip array are shown in Figure 2, compared with the results obtained from HFSS. The figure shows that there is a well agreement between them.

4.2. Discussion of Parallel Performance. Here, we select several groups of virtual topology schemes to be tested. The following are the test results on the two supercomputer center platforms.

4.2.1. NSCC-TJ. Table 2 is the comparison of total computation time (in seconds) with different CPU cores and different virtual topology schemes. The maximum number of CPU cores used for test is 120.

In Table 2, 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, 2 × 1 × 1 means that there is no topology in y and z directions, respectively; thus the virtual topology is actually in one dimension. Similarly, 8 × 8 × 1 means that there is no topology in z direction; thus the virtual topology is actually in two dimensions. In our work, one process uses one CPU core.

Table 2: Comparisons of virtual topology, amount of communication, and computation time on NSCC-TJ.

CPU cores	Virtual topology ($x \times y \times z$)	Amount of communication	Computation time (in seconds)
12	3 × 2 × 2	2520000	6466.68
16	4 × 2 × 2	2880000	6140.56
32	4 × 4 × 2	3600000	2858.97
64	8 × 8 × 1	5040000	1376.85
	8 × 4 × 2	5040000	1379.92
	16 × 2 × 2	7200000	1824.37
96	8 × 6 × 2	5760000	946.52
	8 × 4 × 3	6480000	1500.54
	12 × 4 × 2	6480000	1551.95
	16 × 3 × 2	7560000	1679.42
120	6 × 10 × 2	6480000	702.16
	10 × 6 × 2	6480000	808.65
	5 × 12 × 2	6840000	713.67
	12 × 5 × 2	6840000	863.65
	5 × 6 × 4	7560000	724.90
	6 × 5 × 4	7560000	1076.41
	15 × 4 × 2	7560000	1096.52

The speedup and parallel efficiency of the code are shown in Figure 3. From Figure 3, it can be seen that the parallel efficiency reaches up to 80% on NSCC-TJ.

From Table 2, it is obvious that increasing the number of CPU cores 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 processes. Next the parallel performance of the parallel FDTD will be discussed.

Here, the cases of 96 and 120 cores are taken as the examples. From (3)

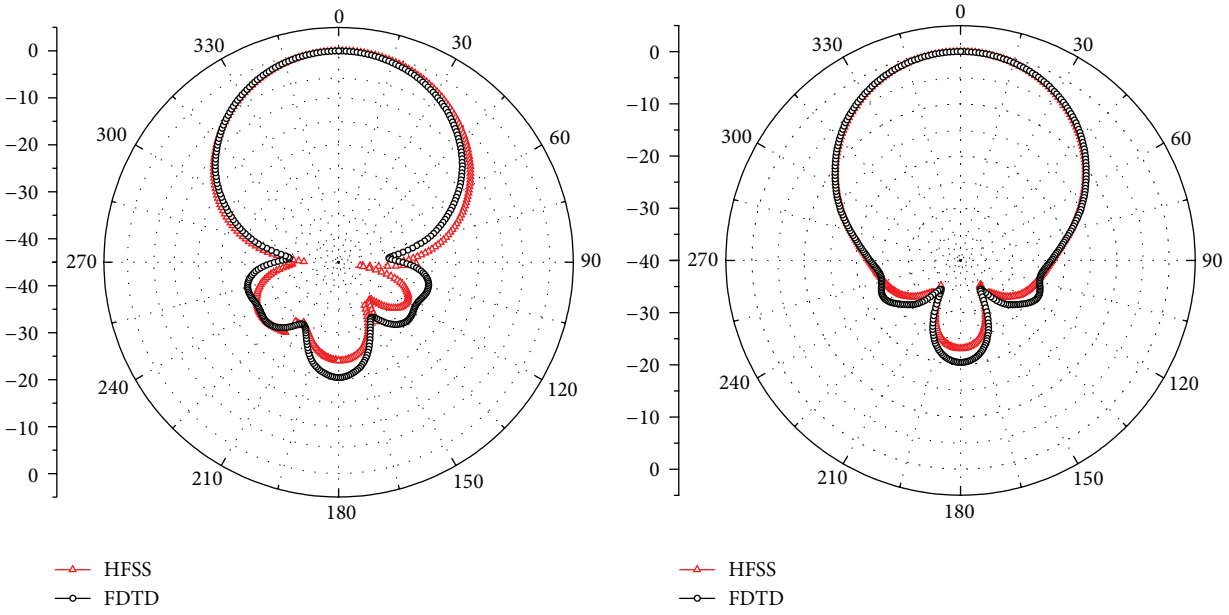
$$L = (P_x - 1) N_y N_z + (P_y - 1) N_z N_x + (P_z - 1) N_x N_y. \quad (7)$$

The following is known.

(a) 96 Cores. Consider

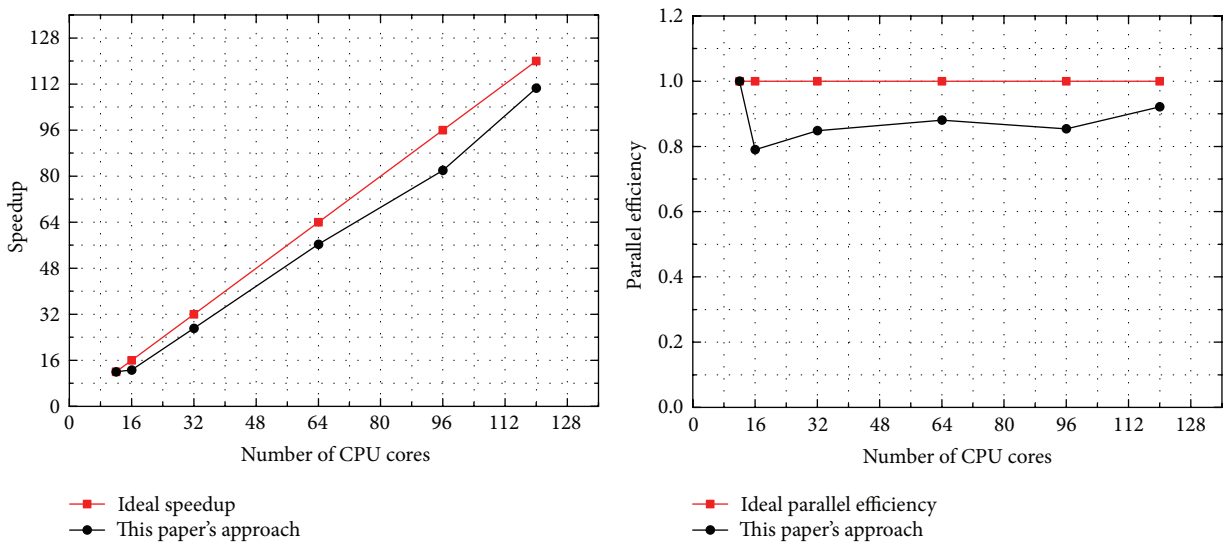
8 × 6 × 2: (8 − 1) × (1200 × 300) + (6 − 1) × (1200 × 300) + (2 − 1) × (1200 × 1200) = 5760000 (946.52 s) (0.5 GB/process)

8 × 4 × 3: (8 − 1) × (1200 × 300) + (4 − 1) × (1200 × 300) + (3 − 1) × (1200 × 1200) = 6480000 (1500.54 s) (0.5 GB/process).



(a) *xoz* plane (b) *yoz* plane

Figure 2: The radiation patterns of the 2 × 2 microstrip antenna array.



(a) Speedup (b) Parallel efficiency

Figure 3: The speedup and parallel efficiency of the code from 12 CPU cores to 120 CPU cores on NSCC-TJ.

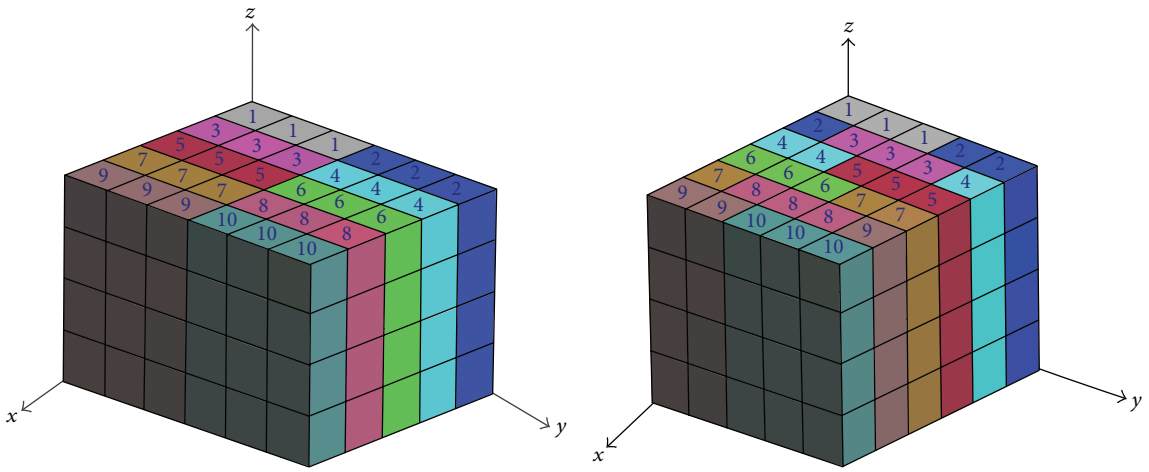
(b) 120 Cores. Consider

$5 \times 6 \times 4$: $(5-1) \times (1200 \times 300) + (6-1) \times (1200 \times 300) + (4-1) \times (1200 \times 1200) = 7560000$ (724.90 s) (0.4 GB/process)

$6 \times 5 \times 4$: $(6-1) \times (1200 \times 300) + (5-1) \times (1200 \times 300) + (4-1) \times (1200 \times 1200) = 7560000$ (1076.41 s) (0.4 GB/process).

From above, it is known that, for the virtual topologies with the same dimensions, the total grids at interfaces less (the amount of communication data less) can save calculation time effectively. Meanwhile, it is obvious that the memory distribution of each process is the same for different topologies with the same number of CPU cores.

But, from above, it also can be seen that, for the same amount of grids, the calculation time has certain differences.



(a) $5 \times 6 \times 4$ (b) $6 \times 5 \times 4$

FIGURE 4: The diagram of communication mode across nodes between the virtual topologies $5 \times 6 \times 4$ and $6 \times 5 \times 4$.

For example, the topologies $6 \times 5 \times 4$ and $5 \times 6 \times 4$ with the same grids number and the calculation time are 1076.41 seconds and 724.90 seconds, respectively. Generally, we believe that the consumption time between processes in one node is less and the more time consumed between ones across nodes [12, 13]. So, here we speculate that the different amount of the communication grids across nodes causes the difference between the two cases.

For $5 \times 6 \times 4$, it has $4 \times (1200 \times 300) + 1 \times (1200 \times 300) = 1800000$ FDTD grids needed to communicate across nodes, and its calculation time is 724.90 seconds, while, for $6 \times 5 \times 4$, it has $5 \times (1200 \times 300) + (1 + 2/6) \times (1200 \times 300) = 2280000$ FDTD grids needed to communicate across nodes, and its calculation time is 1076.41 seconds. The calculation way of communication data above is shown in Figure 4.

In Figure 4, every two adjacent figures with different colors have data communication. Figures 1 to 10 present ten nodes, and the adjacent figures with the same colors present that they are in the same node. For (b), the adjacent columns will transfer $((1 + 2/6) \times (1200 \times 300))$ data in *xoz* plane, and the adjacent rows will transfer $(5 \times (1200 \times 300))$ data in *yoz* plane.

4.2.2. NSCC-SZ. Table 3 is the comparison of total computation time (in seconds) with different CPU cores and different virtual topology schemes. The maximum number of CPU cores used for test is 480.

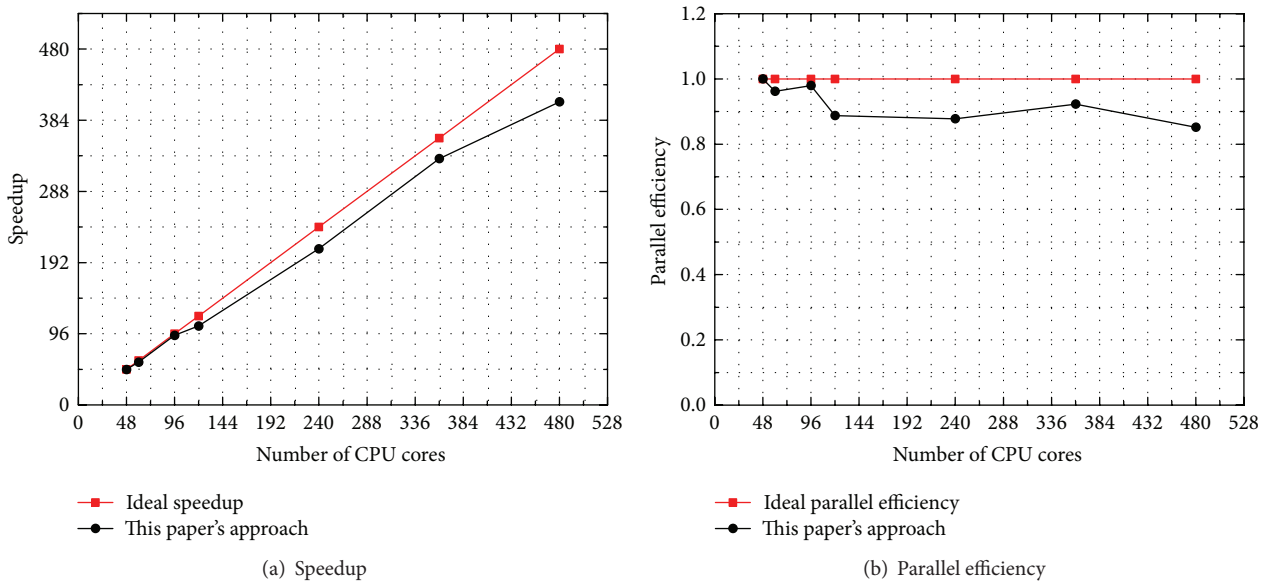
In Table 3, virtual topology schemes are described as $(x \times y \times z)$ for all three communication patterns and the meaning of each figure in each topology is the same with the description in Section 4.2.1 above.

The speedup and parallel efficiency of the code are shown in Figure 5. From Figure 5, it can be seen that the parallel efficiency reaches up to 80% on NSCC-SZ.

From Table 3, it can be seen that, for the virtual topologies with the same dimensions, the total grids at interfaces less (the amount of communication data less) can save calculation

TABLE 3: Comparisons of virtual topology, amount of communication, and computation time on NSCC-SZ.

CPU cores	Virtual topology $(x \times y \times z)$	Amount of communication	Computation time (in seconds)
48	$4 \times 6 \times 2$	4320000	1271.02
	$6 \times 4 \times 2$	4320000	1256.12
60	$5 \times 6 \times 2$	4680000	1055.02
	$3 \times 10 \times 2$	5400000	1044.47
	$3 \times 5 \times 4$	6480000	1214.09
	$10 \times 2 \times 3$	6480000	1117.17
96	$6 \times 8 \times 2$	5760000	596.99
	$8 \times 4 \times 3$	6480000	637.09
	$6 \times 4 \times 4$	7200000	604.82
	$4 \times 6 \times 4$	7200000	641.13
120	$6 \times 10 \times 2$	6480000	492.96
	$10 \times 6 \times 2$	6480000	752.48
	$8 \times 5 \times 3$	6840000	510.25
	$5 \times 8 \times 3$	6840000	566.13
240	$10 \times 12 \times 2$	8640000	286.12
	$8 \times 10 \times 3$	8640000	296.48
	$10 \times 8 \times 3$	8640000	328.07
	$12 \times 10 \times 2$	8640000	417.3
360	$12 \times 10 \times 3$	10080000	197.46
	$10 \times 12 \times 3$	10080000	278.73
	$8 \times 15 \times 3$	10440000	181.46
	$12 \times 15 \times 2$	10440000	297.34
480	$10 \times 12 \times 4$	11520000	147.48
	$12 \times 10 \times 4$	11520000	156.53
	$15 \times 8 \times 4$	11880000	157.57
	$15 \times 16 \times 2$	11880000	162.54
	$12 \times 8 \times 5$	12240000	161.85
	$8 \times 12 \times 5$	12240000	165.3



(a) Speedup

(b) Parallel efficiency

Figure 5: The speedup and parallel efficiency of the code from 48 CPU cores to 480 CPU cores on NSCC-SZ.

time effectively. This conclusion coincided with the case on NSCC-TJ.

The amount of communication grids of each virtual topology is calculated by (3), while, for certain topologies with the same number of grids, it is found that the calculation time is less with the more crossing-node communication, analyzed by the way of NSCC-TJ. It is contrary to the speculation theory above. Therefore, we speculate that whether it is caused by the reason that the communication amount of nodes with the main communication is great in the topology. Next, the cases of $10 \times 2 \times 3$ and $3 \times 5 \times 4$ in 60 cores are taken as the examples to analyze the speculation (the amount of communication is 6480000).

For $10 \times 2 \times 3$, the crossing-node communication is $4 \times (1200 \times 300) + 1 \times (1200 \times 300) = 1800000$, while, for $3 \times 5 \times 4$, the one is $2 \times (1200 \times 300) + 16/12 \times (1200 \times 300) = 1200000$. But the consumption time for $10 \times 2 \times 3$ is 1117.17 seconds and the one for $3 \times 5 \times 4$ is 1214.09 seconds. This does not agree with the speculation of crossing-node theory on NSCC-TJ. To further explore the reason, the heaviest communication and the lightest communication of the related processes from these two virtual topology schemes are listed in Table 4. The heaviest communication load in the virtual topology scheme $10 \times 2 \times 3$ is $(1200 \times 300)/30 + (1200 \times 300) \times 2/6 + (1200 \times 1200) \times 2/20 = 276000$, while, in the virtual topology scheme $3 \times 5 \times 4$, the one is $(1200 \times 300) \times 2/12 + (1200 \times 300) \times 2/20 + (1200 \times 1200) \times 2/15 = 288000$. The heaviest communication loads are assigned to the processes located at the center of the process grid. Similarly, the lightest communication loads that are set to the processes at the corner of the process grid can be calculated. The difference between the heaviest communication load and the lightest communication load in the virtual topology scheme $3 \times 5 \times 4$ is larger than the one in $10 \times 2 \times 3$, which results in different computation time. This indicates that when the total amount of communication is the same, a virtual topology scheme with a more balanced communication load may bring a better performance, although with a more crossing-node communication.

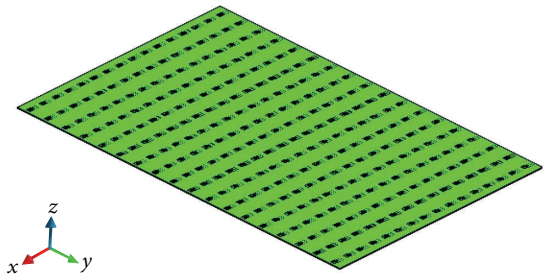


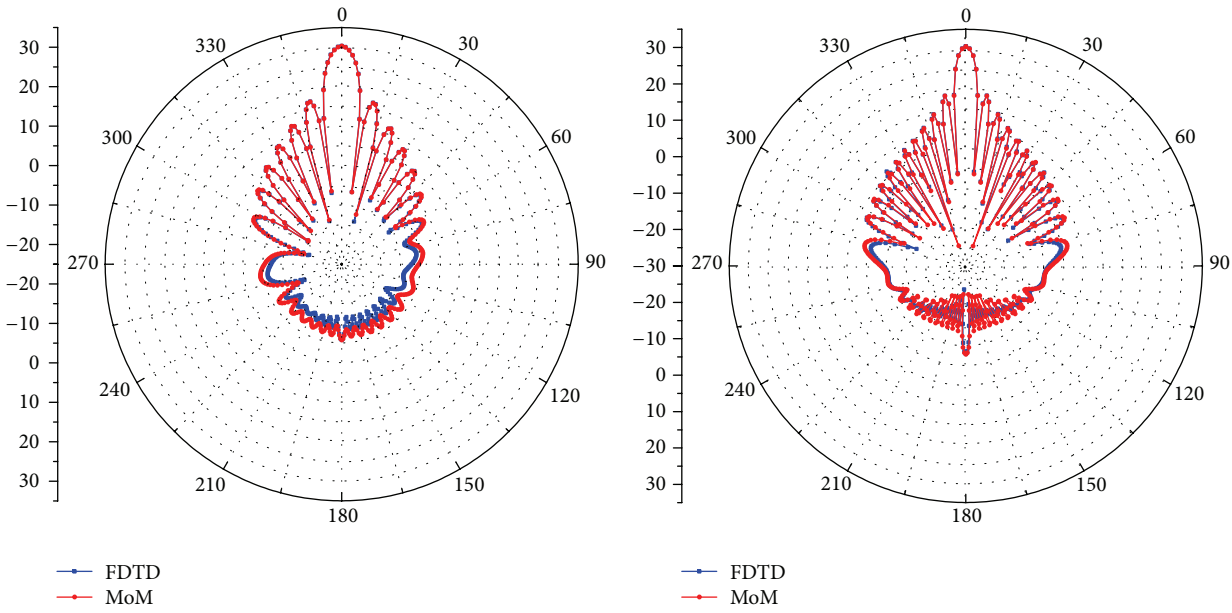
Figure 6: The model of the microstrip antenna array.

Table 4: Comparisons of communication load.

Topology	The heaviest	The lightest	Difference
$10 \times 2 \times 3$	276000	144000	132000
$3 \times 5 \times 4$	288000	144000	144000

4.3. The General Rules of Optimal Virtual Topology. Generally, when the amount of FDTD cells is the same, the MPI virtual topology by the way where the amount of transferred data is less can save the computation time for the same dimensional virtual topology. The best performance of a parallel FDTD code can be obtained by optimizing the MPI virtual topology scheme. The general rules for a better performance are as follows.

(a) Select MPI virtual topology scheme to make the total communication L (equation (3)) the smallest.



(a) *xoz* plane (b) *yoz* plane

Figure 7: The radiation patterns of the array.

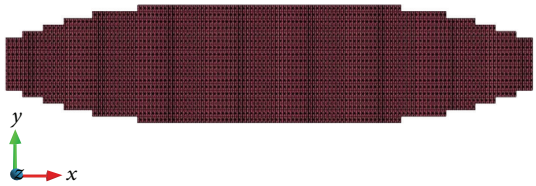


Figure 8: The model of the microstrip array.

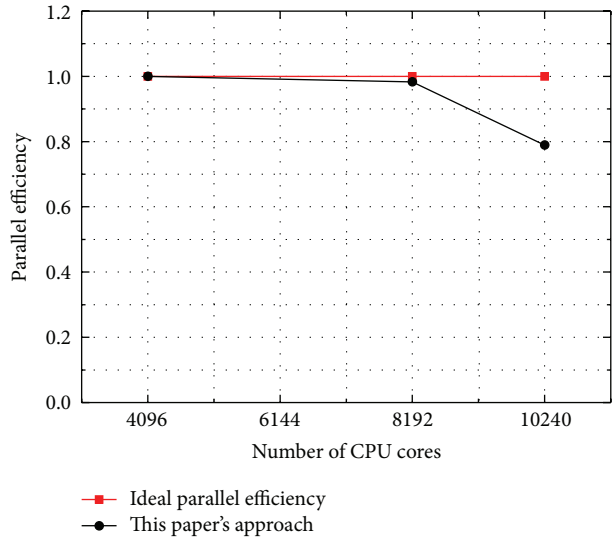


Figure 9: The parallel efficiency of 4096~10240 cores.

(b) When the total communication L is the same, set the topology as with less crossing-node communication.

(c) When the amount of crossing-node communication of different topologies is approximately the same, select the topology with a more balanced communication load.

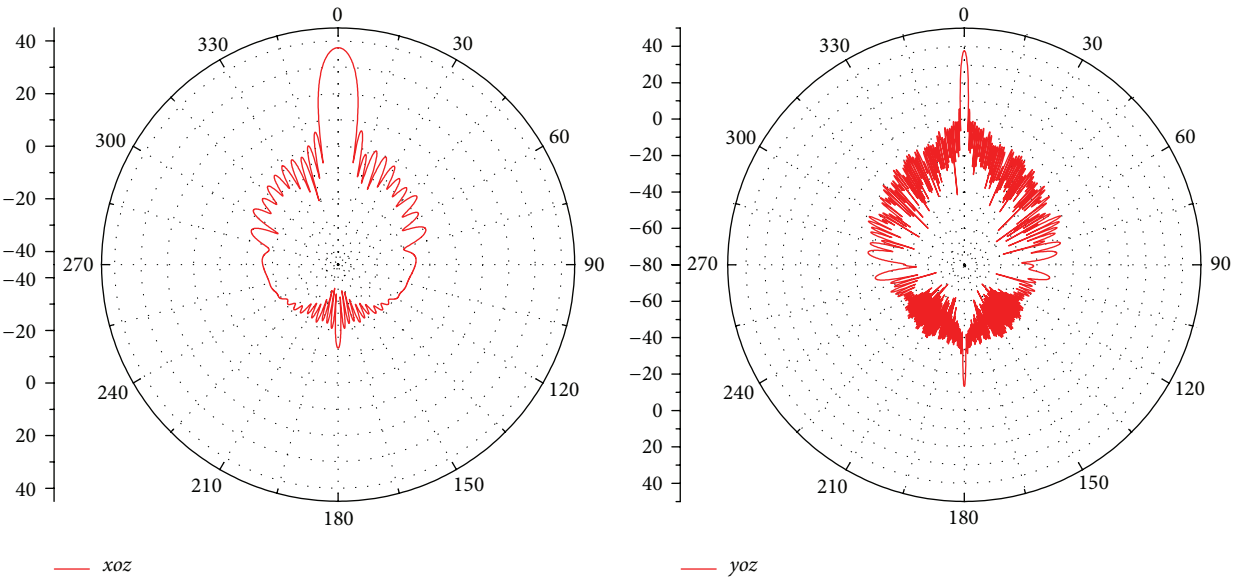
5. Applications Using 10 Thousands of CPU Cores

Based on the optimal virtual topology scheme above, the parallel FDTD code is applied to analyze some complicated EM problems, and they are run on Shanghai Supercomputer Center (SSC) platform.

5.1. Radiation of Microstrip Antenna Array

5.1.1. Validation of the FDTD Code. To validate the FDTD code, a microstrip antenna array with hundreds of the same antenna units is analyzed and the results are compared with the ones provided by MoM. The model is shown in Figure 6. The amount of total grids is $786 \times 1224 \times 54$. It is calculated on PC with the virtual topology scheme $4 \times 3 \times 2$.

The radiation patterns of the array are shown in Figure 7, compared with the ones provided by MoM. From Figure 7, one can see that there are very well agreements between them.



(a) *xoz* plane (b) *yoz* plane

FIGURE 10: The radiation patterns of the ellipse microstrip antenna array.

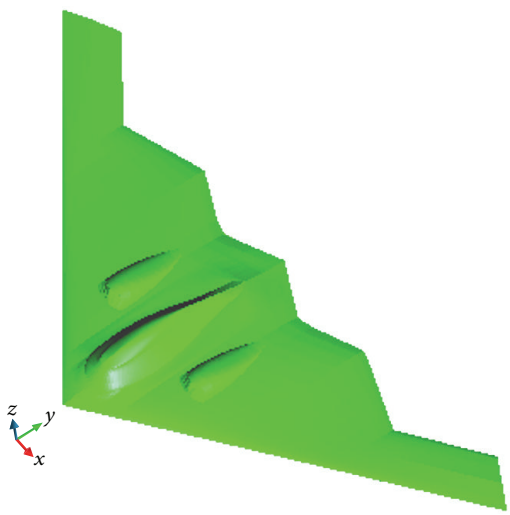


FIGURE 11: The model of the airplane.

5.1.2. Radiation of an Ellipse Antenna Array. An ellipse microstrip antenna array with nearly 2000 elements is shown in Figure 8. In this array, the amount of total grids is 6016 × 1160 × 54 (about 0.4 billion). It is tested on Shanghai Supercomputer Center (SSC), using 10240 cores, and the parallel efficiency from 4096 to 10240 cores is tested.

The parallel efficiency of this code is shown in Figure 9. And the radiation patterns of this array are shown in Figure 10.

From Figure 9, it is known that the parallel efficiency reaches to nearly 80% at 10240 cores with 4096 cores as benchmark, which achieves one of the best efficiencies ever reached using more than 10 thousands of CPU cores.

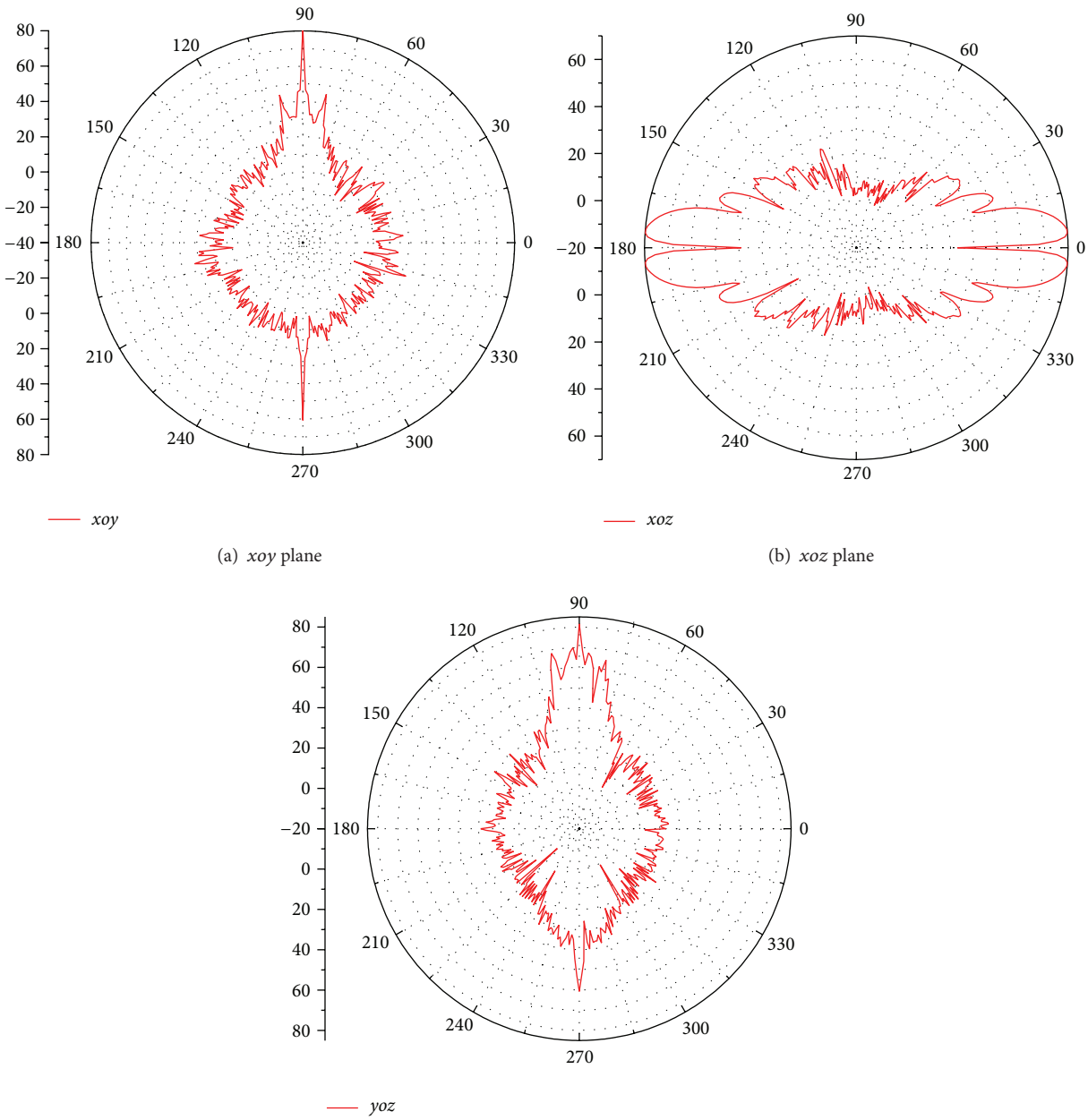
5.2. RCS of an Electrically Large Airplane

5.2.1. Simulation Model. The airplane is shown in Figure 11. The parameters are as follows: Incident wave frequency is 4.0 GHz. The direction of incident wave is $+y$. The polarization is $\mathbf{E}_z$. The size of the airplane is 52.4256 m × 21.0312 m × 2.944213 m. Electric size is about $700\lambda \times 280\lambda \times 39\lambda$. The size of grid is $dx = dy = dz = 0.0075$ m. The amount of total grids is 7020 × 2840 × 420 (about 8.4 billion). It is tested on Shanghai Supercomputer Center (SSC), using 10240 cores.

5.2.2. Result. The computation time of this code is 6036.86 seconds. The RCS of the airplane is shown in Figure 12.

6. Conclusion

A guideline is presented for using parallel FDTD on supercomputers with more than 10 thousands of CPU cores, based on a theoretical communication model given in this paper. The benchmarks obtained on two supercomputers validated the optimal virtual topology rules. Radiation from a large microstrip antenna array and scattering from an electrically large airplane are simulated successfully, which indicate the capability of the method presented in this paper for those types of real-life EM problems.



(a) *xoy* plane

(b) *xoz* plane

(c) *yoz* plane

Figure 12: The RCS of the airplane.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National High Technology Research and Development Program of China (863 Program) (2012AA01A308), the NSFC (61301069, 61072019), the Project with Contract no. 2013KJXX-67, and the Program for New Century Excellent Talents in University of China (NCET-13-0949). The computational resources utilized in this research are provided by the National Supercomputer Center in Tianjin (NSCC-TJ), National Supercomputing Center in Shenzhen (NSCC-SZ), and Shanghai Supercomputer Center (SSC).

References

[1] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Norwood, Mass, USA, 2000.

[2] D. Ge and Y. Yan, *Finite-Difference Time-Domain Method for Electromagnetic Waves*, Version 3, Xidian University Press, Xi'an, China, 2011, (Chinese).

[3] 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.

[4] U. Andersson, *Time-domain methods for the Maxwell equations [Ph.D. thesis]*, Royal Institute of Technology, Stockholm, Sweden, 2001.

[5] 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.

[6] Y. Zhang, J. Song, and C. Liang, "Study on the parallel modified locally conformal FDTD algorithm on cluster of PCs for PBG structures," *Acta Electronica Sinica*, vol. 31, no. 12A, pp. 2142–2144, 2003.

[7] Z. Yu, D. Wei, and C. Liang, "Analysis of parallel performance of MPI based parallel FDTD on PC clusters," in *Proceedings of the Asia-Pacific Conference Proceedings: Microwave Conference Proceedings (APMC '05)*, vol. 4, December 2005.

[8] W. Yu, X. Yang, Y. Liu et al., "A new direction in computational electromagnetics: solving large problems using the parallel FDTD on the BlueGene/L supercomputer providing teraflop-level performance," *IEEE Antennas and Propagation Magazine*, vol. 50, no. 2, pp. 26–44, 2008.

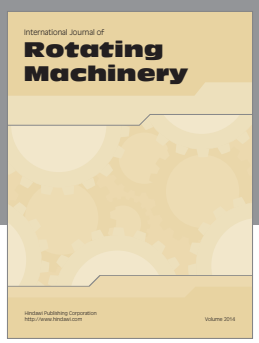
[9] http://www.nscc-tj.gov.cn/.

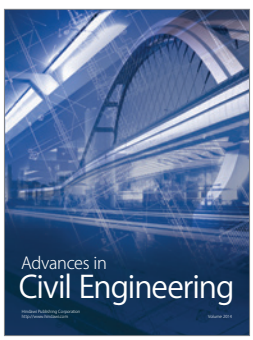
[10] http://www.nsccsz.gov.cn/.

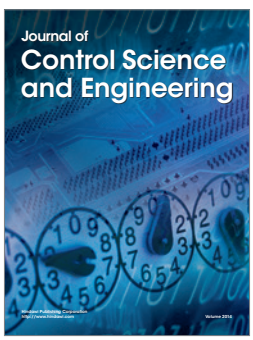
[11] http://www.ssc.net.cn/.

[12] W. Chen and J. Zhai, *Preliminary Analysis on Communication Performance of Dawning 5000A*, 863 High Performance Computer Testing Center of Tsinghua University, 2008.

[13] Intel Corporation, *Intel MPI Library for Linux OS Reference Manual*, Intel Corporation, 2011, https://software.intel.com/sites/products/documentation/hpc/mpi/linux/reference_manual.pdf.







Hindawi
Submit your manuscripts at
http://www.hindawi.com