

# A Hierarchical Partitioning Strategy for an Efficient Parallelization of the Multilevel Fast Multipole Algorithm

Özgür Ergül, *Student Member, IEEE*, and Levent Gürel, *Fellow, IEEE*

**Abstract**—We present a novel hierarchical partitioning strategy for the efficient parallelization of the multilevel fast multipole algorithm (MLFMA) on distributed-memory architectures to solve large-scale problems in electromagnetics. Unlike previous parallelization techniques, the tree structure of MLFMA is distributed among processors by partitioning both clusters and samples of fields at each level. Due to the improved load-balancing, the hierarchical strategy offers a higher parallelization efficiency than previous approaches, especially when the number of processors is large. We demonstrate the improved efficiency on scattering problems discretized with millions of unknowns. In addition, we present the effectiveness of our algorithm by solving very large scattering problems involving a conducting sphere of radius 210 wavelengths and a complicated real-life target with a maximum dimension of 880 wavelengths. Both of the objects are discretized with more than 200 million unknowns.

**Index Terms**—Large-scale problems, multilevel fast multipole algorithm, parallelization, scattering problems, surface integral equations.

## I. INTRODUCTION

**S**URFACE integral equations are commonly used to formulate scattering and radiation problems involving three-dimensional conducting bodies with arbitrary shapes [1]. The application of boundary conditions for the electric field and the magnetic field on the surface of an object leads to the electric-field integral equation (EFIE) and the magnetic-field integral equation (MFIE), respectively. For closed surfaces, EFIE and MFIE can be combined to obtain the combined-field integral equation (CFIE), which is free of the internal-resonance problem [2]. Numerical solutions of integral equations require the discretization (e.g., triangulation) of surfaces. Then, unknown surface currents are expanded in a series of basis functions, and integral equations are tested by employing a

set of testing functions. Finally, solutions of resulting  $N \times N$  dense matrix equations provide the expansion coefficients, which can be used to compute the scattered or radiated electric and magnetic fields everywhere.

Surface integral equations provide accurate results when they are discretized appropriately by using small elements with respect to wavelength. Therefore, when a problem involves a large object with dimensions of several wavelengths, its accurate discretization leads to a large matrix equation with hundreds of thousands of unknowns. Such a large problem can be solved iteratively, where the required matrix-vector multiplications (MVMs) are performed efficiently by the multilevel fast multipole algorithm (MLFMA) [3]. For an  $N \times N$  dense matrix equation, MLFMA reduces the complexity of MVMs from  $\mathcal{O}(N^2)$  to  $\mathcal{O}(N \log N)$ , allowing for the solution of large problems with limited computational resources. On the other hand, accurate solutions of many real-life problems require discretizations with millions of elements, leading to matrix equations with millions of unknowns, which cannot easily be solved with sequential implementations of MLFMA running on a single processor. To solve such large problems, it is helpful to increase computational resources by assembling parallel computing platforms and, at the same time, by parallelizing MLFMA.

The parallelization of MLFMA is not trivial because of the complicated structure of this algorithm. Simple parallelization techniques usually fail to provide efficient solutions, due to communications among processors, poor load-balancing of the workload, and unavoidable duplications of computations over multiple processors. Advanced parallelization techniques have been developed to improve the parallelization of MLFMA by using novel partitioning strategies, load-balancing algorithms, and optimizations for communications [4]–[11]. This way, it has become possible to solve problems with tens of millions of unknowns on relatively inexpensive computing platforms with distributed-memory architectures [4]–[6], [9], [10].

Recently, we developed a hierarchical partitioning strategy that is well suited for the multilevel structure of MLFMA [12]. With the enhanced load-balancing offered by the hierarchical strategy, parallelization of MLFMA can be improved significantly. In this paper, we provide the details of our parallelization algorithm. We employ canonical problems involving sphere geometries of various sizes for the comparison of the hierarchical strategy with previous approaches. We show that the efficiency of the parallelization is improved drastically, especially when the number of processors is large. Improved efficiency provided

Manuscript received June 23, 2008; revised October 16, 2008. Current version published June 03, 2009. This work was supported in part by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 105E172 and 107E136, in part by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and in part by contracts from ASELSAN and SSM.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey and also with the Computational Electromagnetics Research Center (BiLCEM), Bilkent University, TR-06800 Bilkent, Ankara, Turkey (e-mail: ergul@ee.bilkent.edu.tr; lgurel@bilkent.edu.tr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAP.2009.2019913

by the hierarchical strategy is also demonstrated on scattering problems discretized with more than 100 million unknowns. Finally, we present the solutions of very large scattering problems involving a sphere of radius  $210\lambda$  and a stealth airborne target with a maximum dimension of  $880\lambda$ , which are discretized with 204,823,296 and 204,664,320 unknowns, respectively, and  $\lambda$  denotes the wavelength.

The rest of the paper is organized as follows. In Section II, we summarize an efficient implementation of MLFMA, focusing on the main stages of the algorithm. Section III presents the parallelization of MLFMA using the hierarchical partitioning strategy. We investigate the communications among processors in Section IV and compare our parallelization technique with the previous approaches in Section V. Finally, numerical results are presented in Section VI, followed by our concluding remarks in Section VII.

## II. MULTILEVEL FAST MULTIPOLE ALGORITHM

For perfectly-conducting objects, discretizations of surface integral equations lead to  $N \times N$  dense matrix equations in the form of

$$\sum_{n=1}^N Z_{mn} a_n = v_m, \quad m = 1, 2, \dots, N \quad (1)$$

where the matrix elements  $Z_{mn}$  for  $m, n = 1, 2, \dots, N$  can be interpreted as electromagnetic interactions of discretization elements, i.e., basis and testing functions. The matrix equation (1) can be solved iteratively via a Krylov subspace algorithm, where the required MVMs are performed efficiently by MLFMA [3]. In general, MLFMA splits MVMs as

$$\bar{\mathbf{Z}} \cdot \mathbf{x} = \bar{\mathbf{Z}}_{NF} \cdot \mathbf{x} + \bar{\mathbf{Z}}_{FF} \cdot \mathbf{x} \quad (2)$$

where near-field interactions denoted by  $\bar{\mathbf{Z}}_{NF}$  are calculated directly and stored in memory to perform the partial multiplications  $\bar{\mathbf{Z}}_{NF} \cdot \mathbf{x}$ , while multiplications involving far-field interactions, i.e.,  $\bar{\mathbf{Z}}_{FF} \cdot \mathbf{x}$ , are performed approximately and efficiently. In this section, we briefly describe an efficient implementation of MLFMA by summarizing the main stages of the algorithm.

### A. Discretization of the Object

Without losing generality, we consider a smooth object with an electrical dimension of  $kD$ , where  $k = 2\pi/\lambda$  is the wavenumber. Discretization (triangulation) of the object with  $\lambda/10$  mesh size leads to  $N$  unknowns, where  $N = \mathcal{O}(k^2 D^2)$ . As basis and testing functions, we use Rao-Wilton-Glisson (RWG) [13] functions defined on planar triangles.

### B. Clustering

To calculate electromagnetic interactions in a multilevel scheme, a tree structure is constructed by placing the object in a cubic box and recursively dividing the computational domain into subdomains, until the box size is about  $0.25\lambda$ . A multilevel tree structure with  $(L+2) = \mathcal{O}(\log(kD)) = \mathcal{O}(\log N)$  levels

is obtained by considering nonempty boxes (clusters)<sup>1</sup>. At level  $l$  from 1 to  $L$ , the number of clusters can be approximated as

$$N_l \approx 4^{(1-l)} N_1 \quad (3)$$

where  $N_1 = \mathcal{O}(N)$ . In other words, the number of clusters decreases approximately by a factor of four from a level to the next upper level.

The tree structure in MLFMA can be constructed by using a top-down or a bottom-up strategy [10]. In the top-down strategy, the size of the largest cube enclosing the object is minimized, while the size of the smallest boxes at the lowest level depends on the size of the object and the number of levels. In the bottom-up strategy, however, the size of the smallest boxes is fixed to some value (such as  $0.25\lambda$ ), and the sizes of the boxes at higher levels are recursively doubled until the whole object is enclosed by the largest box. For a given problem, one of the two strategies can be preferable in terms of efficiency and accuracy.

### C. Sampling

For each cluster in the tree structure, radiated and incoming fields are defined and sampled on the unit sphere. We choose samples regularly spaced in the  $\phi$  direction and use the Gauss-Legendre quadrature in the  $\theta$  direction [14]. For level  $l = 1, 2, \dots, L$ , the number of samples is  $S_l^\theta = (T_l + 1)$  and  $S_l^\phi = 2(T_l + 1)$  along  $\theta$  and  $\phi$  directions, respectively, where  $T_l$  is the truncation number determined by the excess bandwidth formula [15], i.e.,

$$T_l \approx 1.73ka_l + 2.16(d_0)^{2/3}(ka_l)^{1/3}. \quad (4)$$

In (4),  $a_l$  is the box size at level  $l$ , and  $d_0$  is the desired digits of accuracy. The sampling rate depends on the cluster size as measured by the wavelength ( $ka_l = 2\pi a_l/\lambda$ ), and the total number of samples can be approximated as

$$S_l = S_l^\theta S_l^\phi \approx 2^{(l-1)} S_1^\theta \times 2^{(l-1)} S_1^\phi = 4^{(l-1)} S_1 \quad (5)$$

where  $S_1 = \mathcal{O}(1)$ .

### D. Far-Field Interactions

In MLFMA, far-field interactions are calculated in a cluster-by-cluster manner using the diagonalization and factorization of the homogenous-space Green's function [14]. In each MVM, three main stages, i.e., aggregation, translation, and disaggregation, are performed as described below.

1) *Aggregation*: In this stage, radiated fields of clusters are calculated from the bottom of the tree structure to the highest level ( $l = L$ ). At the lowest level, radiation patterns of basis functions, which are calculated during the setup of MLFMA, are multiplied with the coefficients provided by the iterative solver

<sup>1</sup>In this paper,  $L$  represents the number of effective levels, where MLFMA stages, i.e., aggregation, translation, and disaggregation, are performed. The actual number of levels is  $(L+2)$ , but the highest two levels are not used directly in MLFMA.

and combined to obtain the radiated fields of the smallest clusters. Then, the radiated fields of clusters at higher levels are obtained by shifting and combining the radiated fields of clusters at lower levels. During the aggregation stage, we use a local Lagrange interpolation between successive levels to match different sampling rates for fields.

2) *Translation*: In this stage, radiated fields computed during the aggregation stage are translated into incoming fields. For each cluster at any level, there are  $\mathcal{O}(1)$  clusters to translate the radiated field to. In addition, using the symmetry of cubic (identical) clusters, the number of different translation operators is  $\mathcal{O}(1)$ , independent of the level [4]. Translation operators are calculated during the setup of MLFMA in  $\mathcal{O}(N)$  processing time using local interpolation methods [16].

3) *Disaggregation*: This stage involves the calculation of total incoming fields at cluster centers from the top of the tree structure to the lowest level. At the highest level, the total incoming field for a cluster is obtained by the combination of incoming fields due to translations. At lower levels, however, the incoming field to the center of a cluster involves a contribution from the incoming field to the center of its parent cluster. We use transpose interpolation (anterpolation) between consecutive levels during the disaggregation stage to match different sampling rates of the levels [17]. Following the disaggregation operations at the lowest level, incoming fields are received by the testing functions. Similar to the radiation patterns of basis functions, receiving patterns of testing functions are also calculated during the setup of MLFMA.

Considering the three stages of MLFMA, the processing time and memory required for all operations at level  $l$  is proportional to the product of the number of clusters and the number of samples, i.e.,

$$N_l S_l \approx 4^{(1-l)} N_1 4^{(l-1)} S_1 = N_1 S_1 = \mathcal{O}(N). \quad (6)$$

We note that all levels of MLFMA have equal importance with  $\mathcal{O}(N)$  complexity in terms of processing time and memory.

### E. Near-Field Interactions

In MLFMA, there are also  $\mathcal{O}(N^2/N_1) = \mathcal{O}(N)$  near-field interactions, which are calculated directly in the setup stage of the program and stored in memory to be used multiple times during the iterations. These interactions are between the basis and testing functions that are located close to each other. We use singularity extraction techniques [18]–[21] and Gaussian quadratures [22] in order to calculate the near-field interactions accurately and efficiently.

## III. HIERARCHICAL PARALLELIZATION OF MLFMA

The main task in the parallelization of MLFMA on distributed-memory architectures is partitioning the multilevel tree structure among processors. Simple parallelization techniques, based on distributing clusters among processors, usually fail to provide efficient solutions. This is mainly due to dense communications between processors, duplication of computations, and unbalanced distribution of the workload among processors [7], [8]. Since such problems arise mostly at the

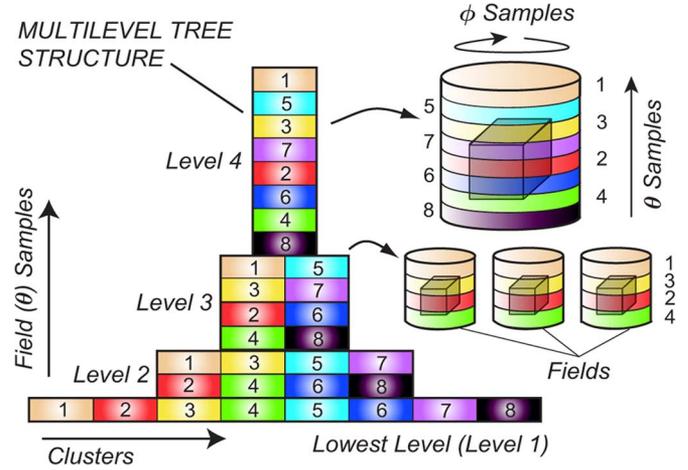


Fig. 1. Distribution of a four-level tree structure among eight processors using the hierarchical partitioning strategy.

higher levels of MLFMA, a hybrid parallelization technique, which applies different partitioning strategies for the lower and the higher levels, is developed to improve the efficiency [7]–[10]. In this technique, processor assignments are made on the basis of fields of clusters at the higher levels. In other words, each cluster at higher levels is shared by all processors, while each processor is assigned to the same portion of fields for all clusters. Even though the hybrid parallelization technique increases the parallelization efficiency significantly, compared to simple parallelization approaches, the improvement can be insufficient, especially when the number of processors is large.

In this section, we provide the details of the hierarchical parallelization of MLFMA for the efficient solution of large-scale problems. The hierarchical parallelization is based on the simultaneous partitioning of clusters and their fields at all levels. We adjust the partitioning in both directions (clusters and samples of fields) appropriately by considering the number of clusters and the number of samples at each level. As an example, Fig. 1 depicts a four-level tree structure ( $L = 4$ ), where levels are represented by two-dimensional rectangles. Horizontal and vertical dimensions of rectangles correspond to clusters and  $\theta$  samples of fields, respectively. The tree structure is partitioned among eight processors labeled 1 to 8. At the lowest level, clusters are distributed among eight processors, and each cluster is assigned to a single processor, without any partitioning of field samples. Then, at the next level ( $l = 2$ ), field samples are partitioned among two groups of processors, i.e., (1,3,5,7) and (2,4,6,8), while the number of cluster partitions is reduced to four. At this level, samples of each cluster are shared by two processors. As we proceed to the higher levels, the number of partitions for clusters and samples of fields are systematically decreased and increased, respectively.

In the following subsections, we present the hierarchical parallelization of MLFMA in detail by considering the main stages of the algorithm.

### A. Partitioning of the Tree Structure

We consider the parallelization of MLFMA on a cluster of  $p$  processors, where  $p = 2^i$  for some integer  $i$ . Using the hierar-

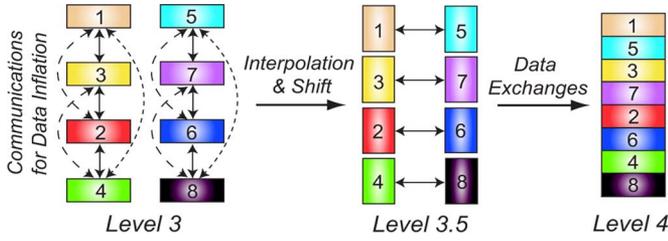


Fig. 2. Aggregation operations from level 3 to level 4 for the partitioned tree structure in Fig. 1.

chical partitioning strategy, the number of partitions for clusters at level  $l$  is chosen as

$$p_{l,c} = \max \left\{ \frac{p}{2^{(l-1)}}, 1 \right\} = \max \left\{ p 2^{(1-l)}, 1 \right\}. \quad (7)$$

We note that clusters are not partitioned for levels  $l > \log_2(p)$ , if such a level exists. The number of clusters assigned to each processor can be approximated as

$$N_l^p \approx \frac{N_l}{p_{l,c}} \approx \left\{ \begin{array}{l} \frac{2^{(1-l)} N_1}{p}, \quad l \leq \log_2(p) \\ 4^{(1-l)} N_1, \quad l > \log_2(p) \end{array} \right\}. \quad (8)$$

In addition, samples of the fields are divided into

$$p_{l,s} = \frac{p}{p_{l,c}} = \min \{ 2^{(l-1)}, p \} \quad (9)$$

partitions along the  $\theta$  direction for level  $l$ . Field samples are partitioned only along the  $\theta$  direction for an easy implementation of interpolation/interpolation operations [7]. The number of  $\theta$  samples assigned to each processor is

$$S_l^{\theta,p} \approx \frac{S_l^\theta}{p_{l,s}} \approx \left\{ \begin{array}{l} S_1^\theta, \quad l \leq \log_2(p) \\ \frac{2^{(l-1)} S_1^\theta}{p}, \quad l > \log_2(p) \end{array} \right\}. \quad (10)$$

Also considering the sampling in the  $\phi$  direction, the total number of samples per processor can be written as

$$S_l^p = S_l^{\theta,p} S_l^\phi \approx \left\{ \begin{array}{l} 2^{(l-1)} S_1, \quad l \leq \log_2(p) \\ \frac{4^{(l-1)} S_1}{p}, \quad l > \log_2(p) \end{array} \right\}. \quad (11)$$

Finally, the size of the local data at each processor is

$$N_l^p S_l^p \approx \frac{N_1 S_1}{p} \quad (12)$$

for  $l = 1, 2, \dots, L$ .

### B. Aggregation Stage

For the partitioned tree structure in Fig. 1, aggregation operations from level 3 to level 4 are depicted in Fig. 2 and can be listed as follows.

1) *One-to-One Communications for Data Inflation*: During aggregations from a level to the next higher level, interpolations are required to increase the sampling rate for radiated fields. Using local Lagrange interpolation, each target point in the fine grid has contributions from a set of neighboring points in the coarse grid. Therefore, when samples of fields are partitioned

among processors, interpolations in each processor need samples located in other processors. Consequently, before interpolations, one-to-one communications are required between pairs of processors to inflate the local data, in accordance with the interpolation requirements.

For the partitioned tree structure in Fig. 1, aggregation from level 3 to level 4 requires one-to-one communications within two separate groups of processors that are located in the same columns, i.e., (1,2,3,4) and (5,6,7,8), as depicted in Fig. 2. As an important advantage of the hierarchical partitioning strategy, distribution of the  $\theta$  samples into large numbers of partitions is avoided. Therefore, communications are required mostly between pairs of processors located “next to each other.” For example, processor 3 communicates mainly with processors 1 and 2, but not with processor 4. Processors 3 and 4 would need to communicate with each other if the number of the  $\theta$  samples required for the interpolation is larger than the number of the  $\theta$  samples per processor. However, using the hierarchical strategy, the number of partitions along the  $\theta$  direction, hence the number of the  $\theta$  samples per processor, can be adjusted such that those secondary communications between “distant” processors are avoided.

2) *Interpolation and Shifting*: When the required data is prepared by one-to-one communications for a cluster, its radiated field is interpolated and shifted to the center of its parent cluster. Temporary levels involving parent clusters and field samples after the interpolation and shifting operations are denoted as intermediate levels. As an example, for the partitioned tree structure in Fig. 1, level 3.5 is depicted in Fig. 2. Following the interpolation, the number of samples along the  $\theta$  direction assigned to each processor is doubled, i.e.,

$$S_{l+1/2}^{\theta,p} \approx 2S_l^{\theta,p} \approx \left\{ \begin{array}{l} 2S_1^\theta, \quad l \leq \log_2(p) \\ \frac{2^l S_1^\theta}{p}, \quad l > \log_2(p) \end{array} \right\}. \quad (13)$$

At the same time, the number of clusters in each processor can be written as

$$N_{l+1/2}^p \approx \frac{N_l^p}{4} \approx \left\{ \begin{array}{l} \frac{2^{-(l+1)} N_1}{4}, \quad l \leq \log_2(p) \\ \frac{4^{(-l)} N_1}{4}, \quad l > \log_2(p) \end{array} \right\}. \quad (14)$$

Intermediate levels are defined temporarily and used to arrange the data in each processor, before communications are performed to modify the partitioning according to the hierarchical strategy.

3) *Data Exchanges*: From an intermediate level  $(l + 1/2)$  to the next level  $(l + 1)$ , data is exchanged among processors, if  $l \leq \log_2(p)$ . As depicted in Fig. 2, processors are paired according to their positions in the partitioning scheme. Each processor performs the following communications.

- Send half of the field samples of each cluster at the intermediate level;
- Receive the complementary data, which involves field samples of some clusters, from the associated processor.

With data exchanges, the number of clusters in each processor is doubled with respect to the number of clusters at the intermediate level, while the number of samples along the  $\theta$  direction is halved. Then, we have

$$N_{l+1}^p \approx \left\{ \begin{array}{l} \frac{2^{(-l)} N_1}{4}, \quad (l + 1) \leq \log_2(p) \\ 4^{(-l)} N_1, \quad (l + 1) > \log_2(p) \end{array} \right\} \quad (15)$$

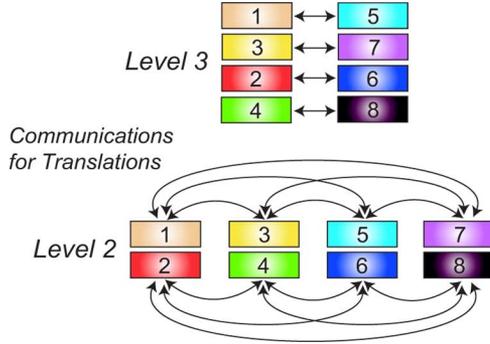


Fig. 3. One-to-one communications during the translation stage at levels 2 and 3 of the partitioned tree structure in Fig. 1.

and

$$S_{l+1}^{\theta,p} \approx \begin{cases} S_1^\theta, & (l+1) \leq \log_2(p) \\ \frac{2^l S_1^\theta}{p}, & (l+1) > \log_2(p) \end{cases} \quad (16)$$

which agree with the expressions in (8) and (10), respectively.

### C. Translation Stage

Using the hierarchical partitioning strategy, one-to-one communications are also required during the translation stage, since clusters are partitioned, and some translations are needed among clusters located in different processors. These communications are achieved by pairing the processors and transferring the radiated fields of clusters between the pairs. As depicted in Fig. 3, communications are required only among the processors that are located in the same row of the partitioning. For example, communications at level 2 are performed within two separate groups of processors, i.e., (1,3,5,7) and (2,4,6,8). In general, for “inter-processor” translations at level  $l$ , each processor is paired one by one with the other  $(p_{l,c} - 1)$  processors. Once a pairing is established, radiated fields of clusters are transferred, and translations are performed by the receiver processor.

In addition to inter-processor translations, there are also “intra-processor” translations that are related to clusters located in the same processor. These translations can be performed independently in each processor, without any communication.

### D. Disaggregation Stage

The parallelization of the disaggregation stage is very similar to the parallelization of the aggregation stage. In general, operations in the aggregation stage are performed in a reverse manner.

1) *Data Exchanges*: When incoming fields are calculated at cluster centers at level  $l$ , partitioning is modified via data exchanges among the processors. This way, the partitioning at level  $(l - 1/2)$  is generated as required for ant interpolation and shifting operations.

2) *Ant interpolation and Shifting*: Incoming fields at the centers of clusters are ant interpolated and shifted to the centers of their subclusters at level  $(l - 1)$ .

3) *One-to-One Communications for Data Deflation*: Since the ant interpolation is the transpose of the interpolation, some of the samples obtained from an ant interpolation operation should be sent to other processors. This is because interpolations during

the aggregation stage are performed using the inflated data prepared by one-to-one communications among processors. As the reverse of this process, ant interpolations produce inflated data, which must be deflated via one-to-one communications. Following an ant interpolation operation, some of the resulting data are used locally, while the rest are sent to other processors. Similar to the communications during interpolations, data are transferred mostly among neighboring processors in the same column of the partitioning scheme.

## IV. COMMUNICATIONS IN THE HIERARCHICAL PARALLELIZATION OF MLFMA

Using the hierarchical partitioning strategy, computations on the tree structure are distributed among processors with improved load-balancing, compared to previous strategies based on partitioning with respect to only clusters or only samples of fields. However, there are still unavoidable communications among processors, which may reduce the efficiency of the parallelization. In this section, we investigate these communications in detail.

### A. Communications in the Aggregation/Disaggregation Stages

During an interpolation operation in a processor at level  $l = 2, 3, \dots, (L - 1)$ , the amount of data required from other processors for each cluster is proportional to the number of samples in the  $\phi$  direction. Considering also the number of clusters per processor, the communication time for interpolations at level  $l$  can be written as

$$\begin{aligned} t_{\text{int},l} &\propto 2^{(l-1)} S_1^\phi \begin{cases} \frac{2^{(1-l)} N_1}{p}, & l \leq \log_2(p) \\ 4^{(1-l)} N_1, & l > \log_2(p) \end{cases} \\ &= \begin{cases} \frac{N_1 S_1^\phi}{p}, & l \leq \log_2(p) \\ \frac{N_1 S_1^\phi}{2^{(l-1)}}, & l > \log_2(p) \end{cases} \\ &\leq \frac{N_1 S_1^\phi}{p}, \quad (l = 2, 3, \dots, L - 1). \end{aligned} \quad (17)$$

We note that the communication time  $t_{\text{int},l}$  tends to decrease with the increasing number of processors  $p$ .

To switch the partitioning scheme from level to level, each processor exchanges half of its data produced during the aggregation stage. The processing time for these communications can be expressed as

$$\begin{aligned} t_{\text{exc},l} &\propto \begin{cases} \frac{N_1 S_1}{p}, & l \leq \log_2(p) \\ 0, & l > \log_2(p) \end{cases} \\ &\leq \frac{N_1 S_1}{p}, \quad (l = 1, 2, \dots, L - 1) \end{aligned} \quad (18)$$

where the upper bound is again inversely proportional to the number of processors  $p$ . The processing time required for communications during the disaggregation stage is the same as the time required for communications during the aggregation stage.

### B. Communications in the Translation Stage

During the translation stage, each processor is paired one by one with  $(p_{l,c} - 1) = \max\{p2^{(1-l)} - 1, 0\}$  processors to perform inter-processor translations. For each pair, the number of cluster-cluster interactions required to be performed is proportional to the number of clusters per processor. In addition, the

size of the data transferred in each interaction is proportional to the number of local samples per cluster, i.e.,  $S_l^p$ . Therefore, the communication time for translations can be written as

$$t_{\text{trans},l} \propto \max \left\{ p2^{(1-l)} - 1, 0 \right\} N_l^p S_l^p \leq \frac{N_1 S_1}{2^{(l-1)}}, \quad (l = 1, 2, \dots, L). \quad (19)$$

The communication time for translations can be significant, especially at the lower levels of MLFMA.

### V. COMPARISONS WITH PREVIOUS PARALLELIZATION TECHNIQUES

In this paper, we compare the hierarchical parallelization technique with two previous approaches, namely, the simple and the hybrid parallelization techniques. As mentioned in Section III, the simple parallelization of MLFMA is based on the distribution of clusters among processors at all levels. A major disadvantage of this technique is the difficulty in distributing a small numbers of clusters at the higher levels of the tree structure [8]. When the number of processors is large, those clusters must be duplicated over multiple processors. Otherwise, a large amount of data is required to be communicated during the aggregation and disaggregation stages. In addition, when using the simple parallelization technique, translations involve dense communications among processors [7], [8].

The hybrid parallelization technique was successfully developed to improve the parallelization of MLFMA [7]. In this technique, the lower (distributed) levels of MLFMA are partitioned as in the simple technique, i.e., clusters are distributed among processors. In the higher (shared) levels, however, samples of fields are distributed, instead of clusters. Unlike the hierarchical parallelization, samples in a shared level are distributed among all processors, without any partitioning of clusters. Distributing samples provides improved load-balancing and communication-free translations for the higher levels of the tree structure. On the other hand, problems arise for some levels at the middle of the tree, where it is not efficient to distribute either fields or clusters among processors [10].

The hierarchical parallelization technique provides two important advantages, compared to simple and hybrid techniques.

- Improved load-balancing: Partitioning both clusters and samples of fields leads to an improved load-balancing of the workload among processors at each level. Consequently, duplication of computations, which may occur in the simple parallelization, and waits for the synchronization of processors are minimized.
- Reduced communications: Although the hierarchical partitioning increases the types of communications, compared to simple and hybrid approaches, the amount of data transferred is decreased. In addition, due to the improved load-balancing, the average package size is enlarged, and the number of communication events is reduced. As a result, the communication time is significantly shortened.

Finally, another important advantage of the hierarchical parallelization algorithm appears when MLFMA is employed on a cluster with multiprocessor nodes. Most of the mainboards built recently have multiple processors connected via

high-speed buses. Then, parallel computers are constructed by clustering a number of multiprocessor computing units (nodes), instead of processors. Resulting parallel computers are highly nonuniform, since communications among processors in the same node are significantly faster than those among processors located in different nodes. Using multicore processors further complicates the situation, since communications within nodes also have diverse rates, depending on whether the inter-core communications are taking place in the same processor or between two processors in the same node. The hierarchical parallelization technique is suitable for such parallel platforms. As an example, let the tree structure in Fig. 1 be partitioned among two nodes, each having four processors, i.e., processors 1–4 and processors 5–8 are located in two different nodes. Then, all communications during the aggregation and disaggregation stages from level 1 to level 3 are performed “inside” nodes. Inter-node communications are required only for translations and data exchanges during the aggregation/disaggregation stages between level 3 and level 4. In general, the hierarchical partitioning strategy facilitates the processor arrangements in nonuniform platforms to minimize inter-node communications. However, in this paper, we do not use this advantage directly; hence, the improved efficiency obtained with the hierarchical parallelization is general and valid for all types of distributed-memory architectures.

### VI. RESULTS

The results of this paper can be categorized into three parts. First, we demonstrate the improved efficiency provided by the hierarchical parallelization strategy, compared to the previous parallelization approaches, on scattering problems involving spheres of various sizes discretized with millions of unknowns. Second, parallelization efficiency is demonstrated on large-scale scattering problems involving a sphere (a canonical object) and an airborne target Flamme [23] (a complicated object). Finally, we present the solution of very large scattering problems involving a sphere of radius  $210\lambda$  and the Flamme with a maximum dimension of  $880\lambda$ , which are discretized with 204,823,296 and 204,664,320 unknowns, respectively. These are the solutions of the largest problems of their classes ever reported in the literature, to the best of our knowledge.

#### A. Formulation and Solution Parameters

In this paper, scattering problems involve closed conductors, which can be formulated with CFIE. Matrix equations provided by CFIE are usually better conditioned than those obtained with EFIE and MFIE [24], [25]. Using CFIE, iterative convergence is achieved rapidly, and it can be further accelerated by employing simple preconditioners, such as a block-diagonal preconditioner (BDP). In all solutions, problems are discretized with about  $\lambda/10$  mesh size, and near-field interactions are calculated with maximum 1% error. For small problems involving 1.5–13.5 million unknowns, tree structures are constructed by using a bottom-up strategy, and far-field interactions are calculated with three digits of accuracy. For large problems (involving more than 100 million unknowns), we use a top-down

strategy to construct the tree structures, while the far-field interactions are calculated with two digits of accuracy. During the aggregation stage, interpolations from a level to the next higher level are performed using  $6 \times 6$  samples in the coarse grid for each sample in the fine grid. Finally, iterative solutions are performed using the biconjugate-gradient-stabilized (BiCGStab) algorithm [26] accelerated with BDP, and the residual error for the iterative convergence is set to  $10^{-6}$  and  $10^{-3}$  for small and large problems, respectively.

### B. Parallel Computing Platforms

Scattering problems are solved on three different parallel clusters, each involving 16 computing nodes.

- Tigerton Cluster: Each node has 32 gigabytes (GB) of memory and two quad-core Intel Xeon Tigerton processors with 2.93 GHz clock rate;
- Harpertown Cluster: Each node has 32 GB of memory and two quad-core Intel Xeon Harpertown processors with 3.00 GHz clock rate;
- Dunnington Cluster: Each node has 48 GB of memory and four six-core Intel Xeon Dunnington processors with 2.40 GHz clock rate.

In all three clusters, memory in a node is available to all cores in the node. The nodes are connected via Infiniband networks, while the processors in a node are connected through high-speed mainboard buses. In the context of parallelization, we use the terms “processor” and “core” synonymously. For a solution on  $p$  processors, we use the maximum number of nodes available, i.e., the number of processes per node is minimized. In other words, if a code is parallelized into  $2^i$  processes, and if  $2^i \leq 16$ , then we use  $2^i$  nodes, each running only one process. When  $2^i > 16$ , however, the solution is parallelized over 16 nodes, and  $2^i/16$  processors are employed per node.

### C. Parallel Efficiency Results and Comparisons

The solutions presented in this subsection are performed on the Tigerton cluster. Fig. 4 presents the parallelization efficiency obtained for the solution of a scattering problem involving a sphere of radius  $20\lambda$  discretized with 1,462,854 unknowns. Fig. 4(a) depicts the efficiency for the total time (including the setup and iterations), when the solution is parallelized onto 2, 4, 8, 16, 32, 64, and 128 processors. The parallelization efficiency is defined as

$$\varepsilon_p = \frac{T_1}{pT_p} \quad (20)$$

where  $T_p$  is the processing time of the solution with  $p$  processors. Fig. 4(a) shows that the hierarchical scheme improves the parallelization efficiency significantly, compared to simple and hybrid approaches, especially when the number of processors is large. The hybrid parallelization performs better than the simple parallelization; however, its efficiency drops rapidly for  $p > 32$ , and it becomes inefficient, compared to the hierarchical parallelization. Using 128 processors, the hierarchical parallelization technique provides 58% efficiency, corresponding to a 74-fold speedup with respect to the single-processor solution.

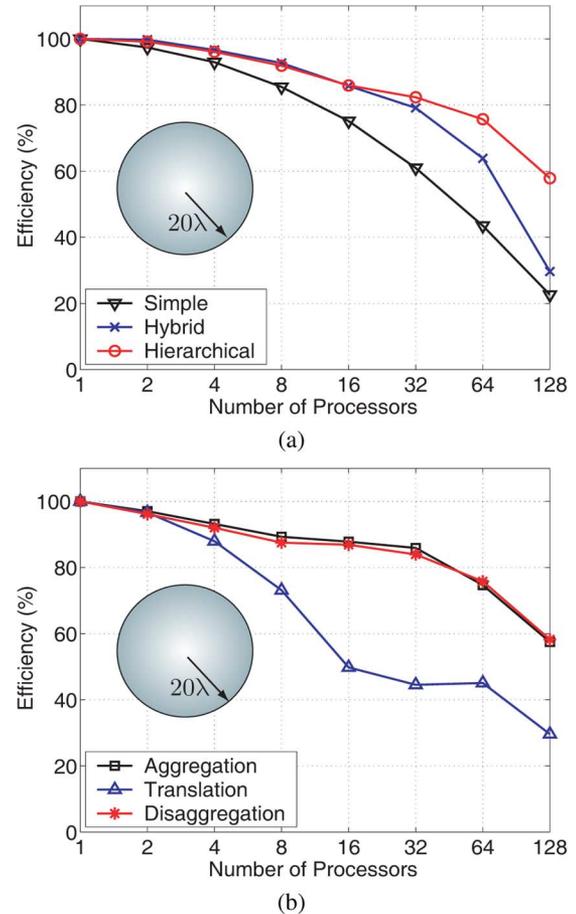


Fig. 4. Parallelization efficiency with respect to the number of processors for the solution of a scattering problem involving a sphere of radius  $20\lambda$  discretized with 1,462,854 unknowns. (a) Overall efficiency including setup and iterations, when the solution is parallelized by using simple, hybrid, and hierarchical techniques. (b) Efficiencies for MLFMA stages, i.e., aggregation, translation, and disaggregation, using the hierarchical technique.

Fig. 4(b) presents the parallelization efficiency for the three stages of MVMs, i.e., aggregation, translation, and disaggregation, using the hierarchical strategy. We observe that the translation stage is a major bottleneck in the hierarchical parallelization of MLFMA. For a solution on 128 processors, the parallelization efficiency of translations drops below 30%. This is because the communication time for translations, given in (19), does not scale with the number of processors  $p$ , unlike the communication time for the aggregation and disaggregation stages. In addition, many communications required for inter-processor translations occur among processors located in different nodes. Then, the rate of communications during the translation stage is mostly limited by the speed of the Infiniband network. Nevertheless, even the parallelization of translations is improved with the hierarchical parallelization technique, and the overall efficiency provided by the hierarchical algorithm is consistently higher than those obtained with simple and hybrid approaches.

Fig. 5 presents the parallelization efficiency for solutions of scattering problems involving spheres of radii  $40\lambda$  and  $60\lambda$  discretized with 5,851,416 and 13,278,096 unknowns, respectively, where the efficiency is defined with respect to solutions

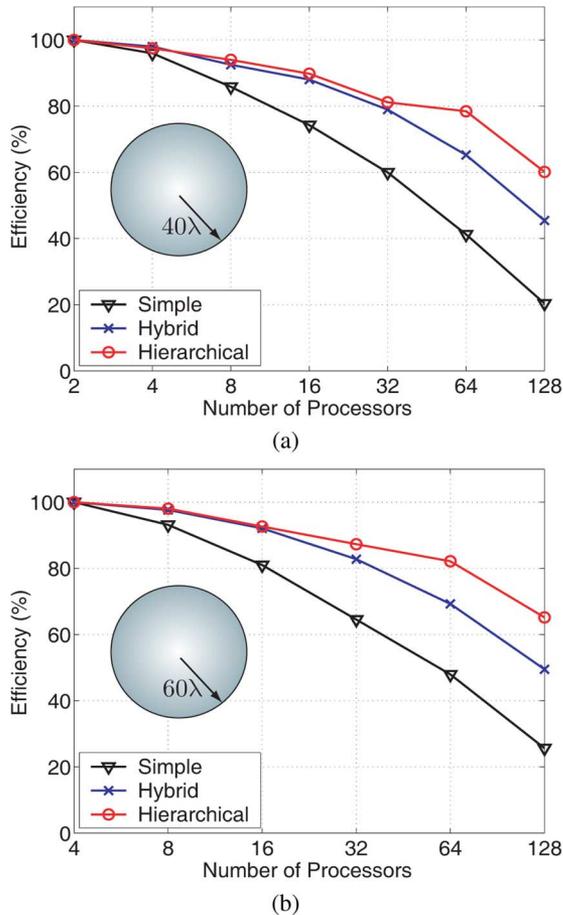


Fig. 5. Parallelization efficiency for the solution of scattering problems involving (a) a sphere of radius  $40\lambda$  discretized with 5,851,416 unknowns and (b) a sphere of radius  $60\lambda$  discretized with 13,278,096 unknowns. Parallel efficiency is defined with respect to two and four processors, respectively.

with two and four processors. Similar to the previous results, the parallelization efficiency is increased significantly by using the hierarchical parallelization technique.

Even though Figs. 4 and 5 compare the relative performances of different parallelization techniques, we emphasize that they do not provide the complete information for the efficiency of solutions. In general, one should also consider the following factors.

- **Clock Rate of the Processors:** Although using faster processors leads to faster solutions, the parallelization efficiency can be degraded as the computation time is reduced. This is because the communication time becomes more significant when the processing time for computations is small.
- **Efficient Implementation of the Algorithm:** We carefully implement MLFMA by minimizing the processing time, which may also have an adverse effect on the parallelization efficiency. For example, as opposed to common implementations of MLFMA, we calculate and store radiation and receiving patterns of basis and testing functions during the setup of the program, and we use them efficiently during iterations. Calculating the patterns on the fly in each MVM without storing them is also a common practice for low-memory implementations. That

TABLE I  
SOLUTIONS OF SPHERE PROBLEMS

Sphere Radius	$20\lambda$	$40\lambda$	$60\lambda$
Unknowns	1,462,854	5,851,416	13,278,096
Number of Iterations	27	30	43
SETUP TIME (seconds)			
Simple	184	1065	2630
Hybrid	89	356	812
Hierarchical	88	348	797
SOLUTION TIME (seconds)			
Simple	512	2467	5400
Hybrid	442	1162	3122
Hierarchical	184	804	2257

TABLE II  
TOTAL PROCESSING TIME AND PARALLELIZATION EFFICIENCY FOR THE SOLUTION OF SCATTERING PROBLEMS DISCRETIZED WITH MORE THAN 100 MILLION UNKNOWNNS

SPHERE (Radius: $360\lambda$ , Number of Unknowns: 135,164,928)			
	16 Processors	32 Processors	64 Processors
Total Time (minutes)	975	511	292
Efficiency	100%	96%	84%
FLAMME (Length: $720\lambda$ , Number of Unknowns: 134,741,760)			
	16 Processors	32 Processors	64 Processors
Total Time (minutes)	1186	646	345
Efficiency	100%	92%	86%

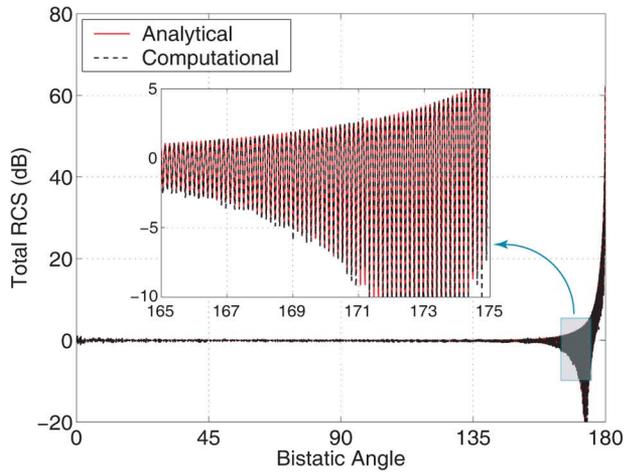
would increase the processing time, but the parallelization efficiency would also increase, since those calculations can be parallelized very efficiently.

- **Accuracy Parameters:** The accuracy of solutions also affects the parallelization efficiency. For example, most of the communications during the aggregation and disaggregation stages could be avoided by reducing the number of interpolation points. This would increase the parallelization efficiency, but the accuracy of the solutions would deteriorate.

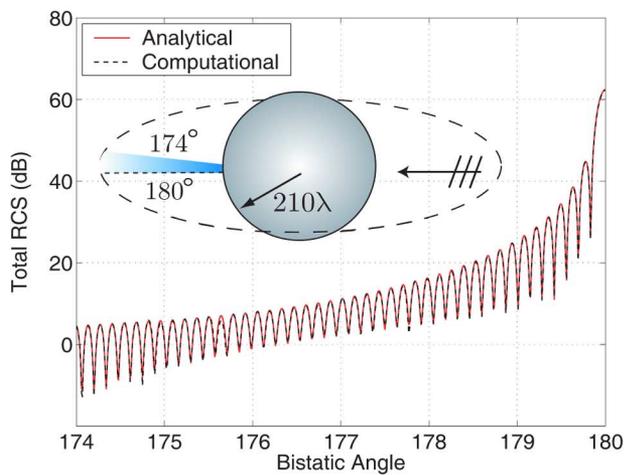
We note that parallel-efficiency results presented in Figs. 4 and 5 are obtained under strict circumstances, using an *efficient* and *accurate* implementation of MLFMA on a cluster of processors with a relatively *high clock rate*. To quantify the efficiency of the solutions, Table I lists processing times, when the three problems are solved on 128 processors. Using the hierarchical parallelization technique, the largest problem with 13,278,096 unknowns is solved in less than one hour.

#### D. Parallel Efficiency for Large-Scale Problems

Table II presents the solution of scattering problems discretized with more than 100 million unknowns. A sphere of radius  $180\lambda$  is discretized with 135,164,928 unknowns and solved by a 10-level MLFMA. We also consider a stealth airborne target, namely, the Flamme [23], having a maximum dimension of 6 meters ( $720\lambda$  at 36 GHz) and discretized with 134,741,760 unknowns. This problem is solved by an 11-level



(a)



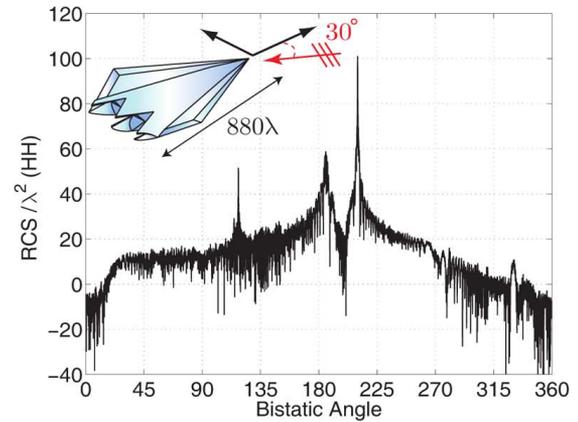
(b)

Fig. 6. Bistatic RCS (in dB) of a sphere of radius  $210\lambda$  discretized with 204,823,296 unknowns (a) from  $0^\circ$  to  $180^\circ$  and (b) from  $174^\circ$  to  $180^\circ$ , where  $0^\circ$  and  $180^\circ$  correspond to back-scattering and forward-scattering directions, respectively.

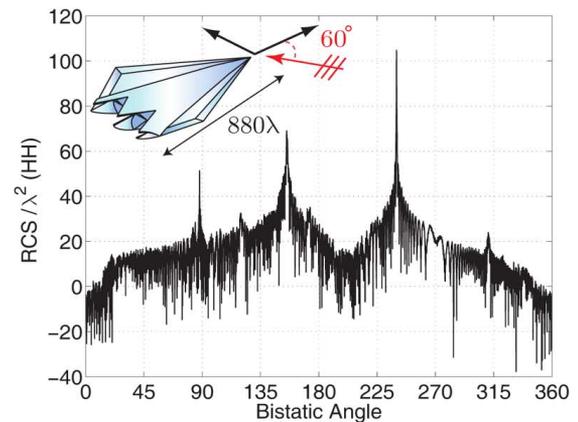
MLFMA. Both of the objects are illuminated by a plane wave, and solutions are performed on 16, 32, and 64 processors of the Harpertown cluster. The number of iterations is 23 and 44 for the sphere and the Flamme, respectively. Table II lists the total processing times including the setup and iterative solution parts, and the parallelization efficiency obtained for 32 and 64 processors with respect to 16 processors. Using 64 processors, parallelization efficiency is more than 80% for both problems. Due to this relatively high efficiency provided by the hierarchical partitioning strategy, we are able to perform each solution in five to six hours.

### E. Solutions of Very Large Problems

Finally, we present the solutions of very large scattering problems discretized with more than 200 million unknowns. A sphere of radius  $210\lambda$  is discretized with 204,823,296 unknowns and solved on 64 processors of the Dunnington cluster. Fig. 6(a) presents the normalized bistatic radar cross section values ( $\text{RCS}/\pi a^2$ , where  $a$  is the radius of the sphere in meters)



(a)



(b)

Fig. 7. Normalized co-polar bistatic RCS ( $\text{RCS}/\lambda^2$  in dB) of the stealth airborne target Flamme at 44 GHz. Maximum dimension of the Flamme is 6 meters, corresponding to  $880\lambda$ . The target is illuminated by plane waves propagating in the  $x$ - $y$  plane at (a)  $30^\circ$  and (b)  $60^\circ$  angles from the  $x$  axis, with the electric field polarized in  $\phi$  direction (horizontal polarization).

in decibels (dB) from  $0^\circ$  to  $180^\circ$  such that  $0^\circ$  corresponds to the back-scattering direction. Fig. 6(b) presents the same results from  $174^\circ$  to  $180^\circ$ . We observe that computational values are in agreement with the analytical values obtained by a Mie-series solution. The solution of the problem using the hierarchical parallelization strategy requires 645 minutes. Following the setup, which takes about 280 minutes, the iterative solution involving 25 iterations is performed in 360 minutes.

Fig. 7 presents the solution of a scattering problem involving the complicated target Flamme at 44 GHz. The maximum dimension of the Flamme is  $880\lambda$  at this frequency. Discretization of the problem with  $\lambda/10$  mesh size leads to 204,664,320 unknowns. As depicted in the insets of Fig. 7, the nose of the Flamme is directed towards the  $x$  axis, and it is illuminated by two plane waves (individually) propagating in the  $x$ - $y$  plane at  $30^\circ$  and  $60^\circ$  angles from the  $x$  axis. The electric field is polarized in the  $\phi$  direction (horizontal polarization). After the setup, which takes 265 minutes, the problem is solved twice for the two excitations in about 1300 minutes. The number of iterations for  $30^\circ$  and  $60^\circ$  illuminations are 38 and 42, respectively. Fig. 7 presents the normalized co-polar bistatic RCS ( $\text{RCS}/\lambda^2$  in dB) on the  $x$ - $y$  plane as a function of the bistatic angle  $\phi$ . For the  $30^\circ$

illumination in Fig. 7(a),  $30^\circ$  and  $210^\circ$  correspond to back-scattering and forward-scattering directions, respectively. We observe that the back-scattered RCS of the Flamme is extremely low; it is 90 dB less than the forward-scattered RCS. This is also observed for the  $60^\circ$  illumination in Fig. 7(b), where the back-scattered RCS at  $60^\circ$  is significantly lower than the forward-scattered RCS at  $240^\circ$ , due to the stealth property of the Flamme.

## VII. CONCLUSION

We present the details of a hierarchical partitioning strategy for the efficient parallelization of MLFMA on relatively inexpensive computing platforms with distributed-memory architectures. Our algorithm is based on partitioning both clusters and field samples among processors at all levels of the multilevel tree structure. This way, load-balancing is improved significantly, compared to previous parallelization approaches based on partitioning with respect to only clusters or only samples of fields. We demonstrate the improved efficiency provided by the hierarchical technique on large scattering problems discretized with millions of unknowns. We also present the solution of very large scattering problems discretized with more than 200 million unknowns. For accurate investigations of complicated targets, such as the Flamme, solutions obtained with parallel MLFMA are extremely important. This is because approximate methods, such as physical optics (PO), may not provide accurate results for those problems, even when objects are large.

## ACKNOWLEDGMENT

The authors would like to thank J. Wilcox of Intel Corporation for the expert technical support and the invaluable assistance he provided to facilitate the experiments on parallel computers. Computer time was provided in part by a generous allocation from Intel Corporation.

## REFERENCES

- [1] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics*, R. Mittra, Ed. Oxford, U.K.: Pergamon Press, 1973, ch. 4.
- [2] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined field solutions for conducting bodies of revolution," *AEÜ*, vol. 32, no. 4, pp. 157–164, Apr. 1978.
- [3] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.
- [4] S. Velampambal, W. C. Chew, and J. Song, "10 million unknowns: Is it that big?," *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.
- [5] M. L. Hastriter, "A study of MLFMA for large-scale scattering problems," Ph.D. dissertation, Univ. Illinois at Urbana-Champaign, 2003.
- [6] G. Sylvand, "Performance of a parallel implementation of the FMM for electromagnetics applications," *Int. J. Numer. Meth. Fluids*, vol. 43, pp. 865–879, 2003.
- [7] S. Velampambal and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.
- [8] Ö. Ergül and L. Gürel, "Efficient parallelization of multilevel fast multipole algorithm," presented at the Proc. Eur. Conf. on Antennas and Propag. (EuCAP), 2006.
- [9] L. Gürel and Ö. Ergül, "Fast and accurate solutions of integral-equation formulations discretized with tens of millions of unknowns," *Electron. Lett.*, vol. 43, no. 9, pp. 499–500, Apr. 2007.

- [10] Ö. Ergül and L. Gürel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2335–2345, Aug. 2008.
- [11] J. Fostier and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2346–2355, Aug. 2008.
- [12] Ö. Ergül and L. Gürel, "Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics," *Electron. Lett.*, vol. 44, no. 1, pp. 3–5, Jan. 2008.
- [13] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.
- [14] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.
- [15] S. Koc, J. M. Song, and W. C. Chew, "Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem," *SIAM J. Numer. Anal.*, vol. 36, no. 3, pp. 906–921, 1999.
- [16] Ö. Ergül and L. Gürel, "Optimal interpolation of translation operator in multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 54, no. 12, pp. 3822–3826, Dec. 2006.
- [17] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comp. Phys. Comm.*, vol. 65, pp. 24–38, Apr. 1991.
- [18] R. D. Graglia, "On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle," *IEEE Trans. Antennas Propag.*, vol. 41, no. 10, pp. 1448–1455, Oct. 1993.
- [19] R. E. Hodges and Y. Rahmat-Samii, "The evaluation of MFIE integrals with the use of vector triangle basis functions," *Microw. Opt. Technol. Lett.*, vol. 14, no. 1, pp. 9–14, Jan. 1997.
- [20] P. Y.-Oijala and M. Taskinen, "Calculation of CFIE impedance matrix elements with RWG and  $\hat{n} \times$  RWG functions," *IEEE Trans. Antennas Propag.*, vol. 51, no. 8, pp. 1837–1846, Aug. 2003.
- [21] L. Gürel and Ö. Ergül, "Singularity of the magnetic-field integral equation and its extraction," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 229–232, 2005.
- [22] D. A. Dunavant, "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *Int. J. Numer. Meth. Eng.*, vol. 21, pp. 1129–1148, 1985.
- [23] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: measurement and calculation of the bistatic radar cross section (BRCS) of a stealth target," *Radio Sci.*, vol. 38, no. 3, Jun. 2003.
- [24] D. R. Wilton and J. E. Wheeler, III, "Comparison of convergence rates of the conjugate gradient method applied to various integral equation formulations," *Progr. in Electromagn. Res.*, pp. 131–158, 1991.
- [25] L. Gürel and Ö. Ergül, "Extending the applicability of the combined-field integral equation to geometries containing open surfaces," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, pp. 515–516, 2006.
- [26] H. A. v. d. Vorst, "Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, pp. 631–644, Mar. 1992.



**Özgür Ergül** (S'98) received the B.Sc. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001 and 2003, respectively, where he is currently working toward the Ph.D. degree.

Since 2001, he has served as a Teaching and Research Assistant in the Department of Electrical and Electronics Engineering, Bilkent University. He has been affiliated with the Computational Electromagnetics Group at Bilkent University from 2000 to 2005 and with the Computational Electromagnetics

Research Center (BilCEM) since 2005. His research interests include fast and accurate algorithms for the solution of electromagnetics problems involving large and complicated structures, integral equations, parallel programming, and iterative techniques.

Mr. Ergül is a recipient of the 2007 IEEE Antennas and Propagation Society Graduate Fellowship and the 2007 Leopold B. Felsen Award for Excellence in Electrodynamics. He is the Secretary of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. His academic endeavors are supported by the Scientific and Technical Research Council of Turkey (TUBITAK) through a Ph.D. scholarship.



**Levent Gürel** (S'87–M'92–SM'97–F'09) received the B.Sc. degree from the Middle East Technical University (METU), Ankara, Turkey, in 1986 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign (UIUC) in 1988 and 1991, respectively, all in electrical engineering.

He joined the Thomas J. Watson Research Center of the International Business Machines Corporation, Yorktown Heights, NY, in 1991, where he worked as a Research Staff Member on the electromagnetic compatibility (EMC) problems related to electronic packaging, on the use of microwave processes in the manufacturing and testing of electronic circuits, and on the development of fast solvers for interconnect modeling. Since 1994, he has been a faculty member in the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, where he is currently a Professor. He was a Visiting Associate Professor at the Center for Computational Electromagnetics (CCEM), UIUC, for one semester in 1997. He returned to the UIUC as a Visiting Professor in 2003–2005, and as an Adjunct Professor after 2005. He founded the Computational Electromagnetics Research Center (BilCEM) at Bilkent University in 2005, where he is serving as the Director. His research interests include the development of fast algorithms for computational electromagnetics (CEM) and the application thereof to scattering and radiation problems involving large and complicated scatterers, antennas and radars, frequency-selective surfaces, high-speed electronic circuits,

optical and imaging systems, nanostructures, and metamaterials. He is also interested in the theoretical and computational aspects of electromagnetic compatibility and interference analyses. Ground penetrating radars and other subsurface scattering applications are also among his research interests. Since 2006, his research group has been breaking several world records by solving extremely large integral-equation problems, most recently the largest involving as many as 205 million unknowns.

Prof. Gürel's many accomplishments include two prestigious awards from the Turkish Academy of Sciences (TUBA) in 2002 and the Scientific and Technical Research Council of Turkey (TUBITAK) in 2003, which are the most notable. He served as the Chairman of the AP/MTT/ED/EMC Chapter of the IEEE Turkey Section in 2000–2003. He founded the IEEE EMC Chapter in Turkey in 2000. He served as the Cochairman of the 2003 IEEE International Symposium on Electromagnetic Compatibility. He is the organizer and General Chair of the CEM'07 ve CEM'09 Computational Electromagnetics International Workshops held in 2007 and 2009. He is a member of the USNC of the International Union of Radio Science (URSI) and the Chairman of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. He served as a member of the General Assembly of the European Microwave Association (EuMA) during 2006–2008. He is currently serving as an Associate Editor for *Radio Science*, the *IEEE Antennas and Wireless Propagation Letters*, *Journal of Electromagnetic Waves and Applications (JEMWA)*, and *Progress in Electromagnetics Research (PIER)*.