

Hierarchical Parallelization of the Multilevel Fast Multipole Algorithm (MLFMA)

Hierarchical parallelization of the multilevel fast multipole algorithm, a method suitable for problems with hundreds of millions of unknowns, is discussed in this paper.

By LEVENT GÜREL, *Fellow IEEE*, AND ÖZGÜR ERGÜL, *Senior Member IEEE*

ABSTRACT | Due to its $\mathcal{O}(N \log N)$ complexity, the multilevel fast multipole algorithm (MLFMA) is one of the most prized algorithms of computational electromagnetics and certain other disciplines. Various implementations of this algorithm have been used for rigorous solutions of large-scale scattering, radiation, and miscellaneous other electromagnetics problems involving 3-D objects with arbitrary geometries. Parallelization of MLFMA is crucial for solving real-life problems discretized with hundreds of millions of unknowns. This paper presents the hierarchical partitioning strategy, which provides a very efficient parallelization of MLFMA on distributed-memory architectures. We discuss the advantages of the hierarchical strategy over previous approaches and demonstrate the improved efficiency on scattering problems discretized with millions of unknowns.

KEYWORDS | Computational electromagnetics; multilevel fast multipole algorithm (MLFMA); parallelization; surface integral equations

Manuscript received July 19, 2011; revised December 3, 2011, April 28, 2012, and August 9, 2012; accepted August 27, 2012. Date of publication November 15, 2012; date of current version January 16, 2013. This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 110E268 and 111E203, by the Centre for Numerical Algorithms and Intelligent Software (EPSRC-EP/G036136/1), and by contracts from ASELSAN Inc. and Undersecretariat for Defense Industries (SSM).

L. Gürel is with the Department of Electrical and Electronics Engineering and the Computational Electromagnetics Research Center (BilCEM), Bilkent University, Bilkent, Ankara TR-06800, Turkey (e-mail: lgurel@bilkent.edu.tr).

Ö. Ergül is with the Department of Mathematics and Statistics, University of Strathclyde, Glasgow G11XH, U.K. (e-mail: ozgur.ergul@strath.ac.uk).

Digital Object Identifier: 10.1109/JPROC.2012.2222331

I. INTRODUCTION

The multilevel fast multipole algorithm (MLFMA) is a powerful tool for iterative solutions of large-scale electromagnetics problems [1]–[3]. This algorithm provides fast and accurate multiplications with dense matrices derived from the discretization of integral-equation formulations. Thanks to its low computational complexity, it is possible to solve electromagnetics problems several orders of magnitude faster by using MLFMA. Without exaggeration, this means accelerating the solutions by thousands or even millions of times, compared to the Gaussian elimination. However, due to the already-too-complicated structure of the algorithm, it is very difficult to parallelize MLFMA for the purpose of solving even larger problems on parallel computers.

Recently, there have been many efforts to improve the parallelization of MLFMA, especially by developing advanced partitioning and distribution schemes [4]–[22]. In [5], an efficient partitioning of computational boxes is discussed considering communications between processors. Using a simple strategy, where each box is assigned to a single processor, the success of such an optimization is limited. For more efficient solutions, traditional partitioning strategies that are based on distributing boxes among processors must be replaced with novel strategies, such as the hybrid partitioning strategy [4]. In [6], it is shown that the hybrid strategy can provide efficient parallelization of MLFMA for canonical and complicated objects on distributed-memory architectures involving as many as 32 processors. In [7] and [10], we further improve the hybrid strategy via load balancing and optimization of communications, enabling the solution of large-scale problems discretized with tens of millions of unknowns. The hybrid strategy is also used in [9], where

the truncation numbers for the far-field interactions are carefully reduced for solving larger problems. Nevertheless, even with the optimized versions of the hybrid strategy, the parallelization efficiency drops rapidly as the number of processors increases to more than 32. Hence, better strategies, such as the asynchronous strategy [11] that can efficiently handle those problems involving multiple dielectric objects, are required for more complicated and larger problems. Such alternative strategies are also developed for shared-memory architectures [18].

In addition to increasingly large numbers of processors, recently developed architectures usually involve multicore processors and highly nonuniform communication rates between cores, leading to new challenges for the efficient parallelization of MLFMA. Along this direction, using the fast Fourier transform (FFT) can provide important advantages when solving extremely large electromagnetics problems on supercomputers [15], [16]. One of the major advances in the parallelization of MLFMA is the development of the hierarchical strategy, which is based on the simultaneous partitioning of boxes and field samples among processors at all levels of tree structures [8]. This strategy is implemented for 2-D [12], [13] and 3-D problems [14], and its favorable properties are demonstrated on extremely large problems involving metallic [19], dielectric [17], [22], and composite [21] objects discretized with hundreds of millions of unknowns. In addition to solutions of challenging problems on moderately large computers, an adaptive application of the hierarchical strategy is used on supercomputers for the solution of multiscale problems [20].

The hierarchical strategy is a successful technique for the efficient parallelization of MLFMA without increasing the complexity of the solver (full parallelization). Low computational complexity enables the solution of large problems with limited computational resources, whereas high parallelization efficiency translates into an ability to efficiently use the available memory to solve even larger problems on moderately large computers with distributed-memory architectures. Within the limits of this particular realm, i.e., the combination of MLFMA and its full parallelization, we have been able to solve some of the largest integral-equation problems in the literature. Solutions of larger equations with other solver-parallelization combinations are reported, but not necessarily faster or more accurate. In general, it is harder to obtain high parallelization efficiency for low-complexity solvers. For example, it is much easier to parallelize the fast multipole method (FMM) [3], [23], [24] with $\mathcal{O}(N^{3/2})$ or $\mathcal{O}(N^{4/3})$ complexity, compared to MLFMA with $\mathcal{O}(N \log N)$ complexity. However, the same problem solved with FMM (e.g., with 95% parallelization efficiency) requires longer central processing unit (CPU) time and more memory than with MLFMA (with 70% parallelization efficiency, for instance) on a fixed number of processors. It is also possible to reduce the solution time and improve the parallelization

efficiency of a fast solver by deliberately decreasing the accuracy of the solution. Therefore, performances of fast solvers should be compared for the same accuracy of their results [25]–[27]. MLFMA is an error-controllable solver, and this feature should be preserved after parallelization.

Unlike previous parallelization techniques, with the hierarchical partitioning strategy, the tree structure of MLFMA is distributed among processors by partitioning both boxes and samples of fields at each level. Due to improved load-balancing and reduced communications, this strategy offers a higher parallelization efficiency than previous approaches, especially when the number of processors is large. In this paper, we review the hierarchical strategy and discuss its advantages over previous approaches. We demonstrate the improved efficiency provided by the hierarchical strategy on scattering problems discretized with millions of unknowns.

II. HIERARCHICAL PARALLELIZATION

For the efficient parallelization of MLFMA, it is crucial to understand the multilevel tree structure, which needs to be distributed among processors. Consider a 3-D object discretized with $\mathcal{O}(N)$ unknowns. A tree of $L = \mathcal{O}(\log N)$ levels can be constructed by placing the object in a computational domain and dividing it into subdomains (boxes). Each box at the lowest level ($l = 1$) involves $\mathcal{O}(1)$ unknowns; hence the number of the lowest level boxes is $\mathcal{O}(N)$. The number of nonempty boxes decays exponentially (usually by a factor of four between two consecutive levels) from the lowest level to the top of the tree structure, and there are $\mathcal{O}(1)$ boxes at the highest level. During a matrix–vector multiplication, only the near-field interactions that are between neighboring boxes at the lowest level are calculated directly. All other interactions are calculated in a group-by-group manner using the factorization and diagonalization of the homogeneous-medium Green’s function [24]. Specifically, each matrix–vector multiplication requires a cycle of aggregation, translation, and disaggregation stages that are performed on the tree structure in a multilevel scheme.

In the aggregation stage, radiated fields for boxes are calculated from the lowest level to the highest level. Using the coefficients provided by the iterative algorithm, radiated fields at the lowest level are obtained by the superposition of radiated fields of the discretization elements, i.e., basis functions. Then, radiated fields at the higher levels are obtained by combining radiated fields at the lower levels. In the diagonalized form, radiated fields are expressed in terms of plane waves, but the addition theorem that is used for the factorization of the interactions is based on multipoles. The number of plane-wave directions, i.e., the number of samples on the unit sphere, can be determined rigorously via excess bandwidth formulas [3], [28] and depends on the box size. In general, $\mathcal{O}(1)$ samples are required at the lowest level and the number of

samples grows exponentially (usually by a factor of four between two consecutive levels) from the lowest level to the highest level of the tree structure.

The aggregation stage is followed by the translation stage, where radiated fields are converted into incoming fields. Using plane waves, translations are simply diagonal with one-to-one mapping, i.e., the incoming field in a direction is the translation of the radiated field in the same direction. For each box, incoming fields from $\mathcal{O}(1)$ different boxes are combined. After translations, the disaggregation stage is performed by calculating the total incoming fields from the top to the bottom of the tree structure. The total incoming field for a box is the combination of incoming fields due to translations and the incoming field from the parent box, if it exists. At the lowest level, incoming fields are received by the discretization elements, i.e., testing functions, to complete the matrix–vector multiplication.

The time and memory complexity of MLFMA is $\mathcal{O}(N \log N)$, and interestingly, each level of the tree structure makes an equal contribution to this overall cost. Generally, the cost of a level is the number of boxes times the number of samples per box. At the lowest level, there are $\mathcal{O}(N)$ boxes and $\mathcal{O}(1)$ samples per box, leading to $\mathcal{O}(N)$ cost. At the highest level, there are $\mathcal{O}(1)$ boxes and $\mathcal{O}(N)$ samples per box, again leading to $\mathcal{O}(N)$ cost. At the intermediate levels, the numbers of boxes and samples balance each other and the cost is $\mathcal{O}(N)$ per level. This interesting property is also the reason why MLFMA is difficult to parallelize. Since all levels of MLFMA have an equal cost, an efficient parallelization of MLFMA should give equal importance to all levels. In fact, the hierarchical strategy is based on this principle; it uses the best partitioning at each level.

A. Partitioning of the Tree Structure

Fig. 1 depicts the partitioning of a multilevel tree structure among eight processors using different strategies. The tree structure involves four levels, each represented by a 3-D rectangular prism partitioned into eight colors (processors). As also depicted in the figure, the horizontal direction stands for boxes in MLFMA, whereas the other two directions are used for field samples on the unit sphere in the θ - and ϕ -directions. Hence, the prism representing the lowest level is long in the horizontal direction (to account for many boxes), but short in the other directions (due to few field samples). Moving higher, the prism dimensions change accordingly, i.e., by shrinking in the horizontal direction (the number of boxes decreases) and expanding in the other two directions (the number of samples increases).

Fig. 1(a) shows a “simple parallelization” of the four-level tree structure among eight processors. In this parallelization strategy, boxes are distributed among processors at all levels. Hence, the partitioning is only in the horizontal direction. This is quite straightforward at the lowest

level involving $\mathcal{O}(N)$ boxes, considering that N is much greater than the number of processors. Unfortunately, problems arise at the higher levels because the number of boxes decreases, making it difficult to distribute small numbers of boxes equally among processors. In fact, quite extreme cases, e.g., distributing 50 boxes among 128 processors, are encountered in real-life simulations. Unequal distribution of boxes at the higher levels is not the only disadvantage of the simple parallelization strategy. Specifically, since the levels are connected via aggregation and disaggregation operations, load balancing at a higher level significantly affects load balancing at the lower levels. Hence, duplications, communications, or both are required in accordance with the relationships between sub-boxes. Consequently, the simple parallelization is efficient only for cases with several (lower) levels and with a small number of processors.

A recent study on the parallelization of MLFMA has led to the development of the “hybrid parallelization” strategy [4], [6]. This strategy is illustrated on the four-level tree structure in Fig. 1(b). Comparing with Fig. 1(a), it can be seen that the lowest two levels are distributed exactly as in the simple strategy. These levels are considered to be parallelized efficiently using the simple strategy. At the higher levels, however, samples instead of boxes are distributed among processors. Since there are many samples at the higher levels, partitioning samples instead of boxes may lead to good load balancing. In fact, as the number of processors increases, it has been shown that the hybrid strategy significantly improves parallelization efficiency compared to the simple strategy. One needs to decide the level at which to change the partitioning (from boxes to samples), but this can be done heuristically, based on the experimental data. In addition to better load balancing at the higher levels, partitioning samples eliminates the need for communications during translations. Although this is an important advantage, one should keep in mind that communications are now introduced in the aggregation and disaggregation stages. At many levels, suppressing these new communications can be a challenging task, if not impossible [10]. The need to reduce communications during the aggregation and disaggregation stages is also the reason for applying the partitioning only along the θ -direction, without any partitioning in the ϕ -direction.

By using different partitioning schemes for the lower and higher levels, the hybrid parallelization strategy can provide more efficient parallelization of MLFMA, compared to the simple strategy. Unfortunately, the hybrid strategy also fails to provide efficient solutions, especially when the number of processors is larger than 16. This is because partitioning only the boxes or samples may not be efficient for some levels in the middle of the tree structures. Specifically, for these levels, partitioning the boxes leads to unequal work distribution, while partitioning the samples leads to excessive communications during the aggregation and disaggregation stages. This dilemma can

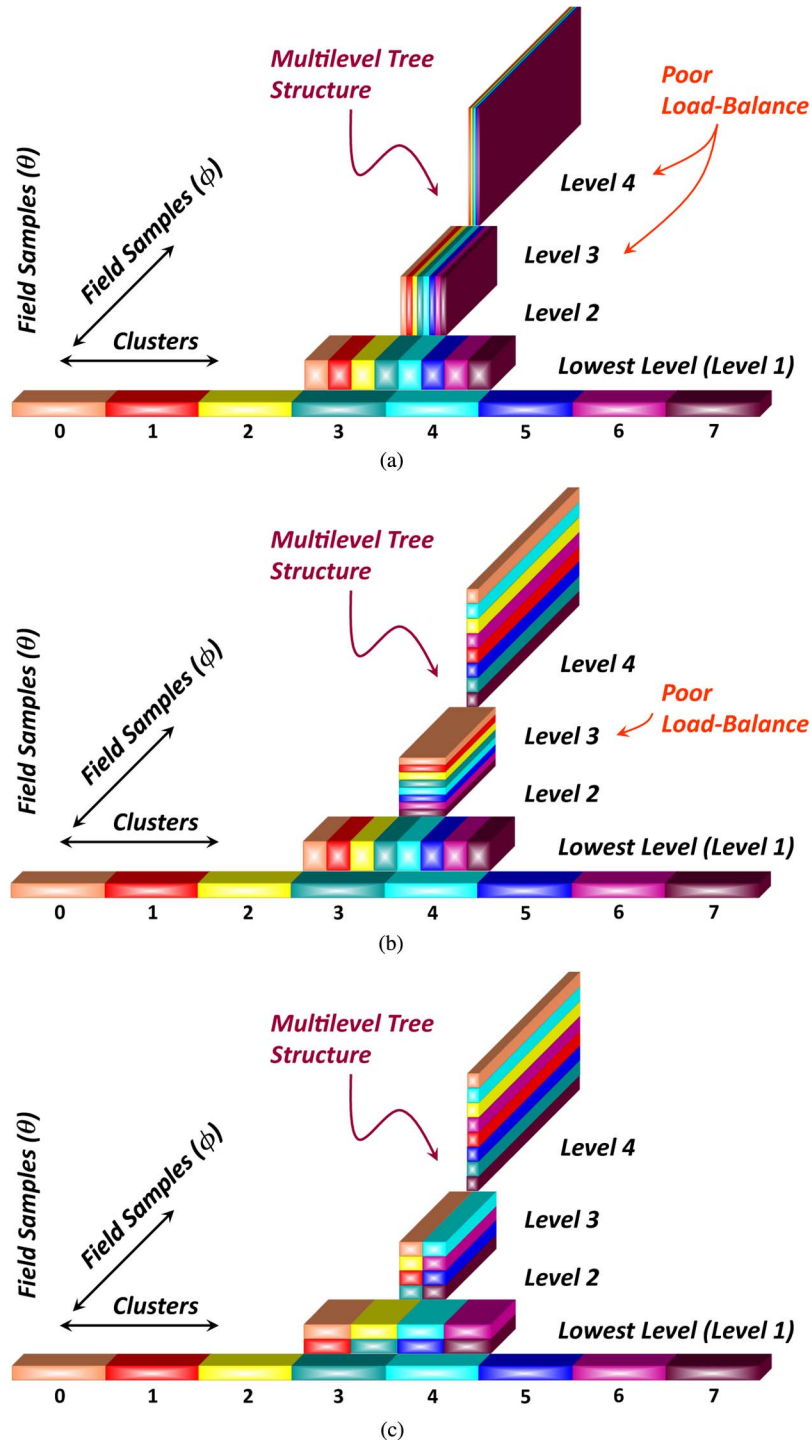


Fig. 1. Partitioning of a four-level tree structure among eight processors using (a) simple, (b) hybrid, and (c) hierarchical strategies.

be solved with the “hierarchical parallelization” strategy proposed and developed in [8] and [14].

Fig. 1(c) depicts the hierarchical strategy on the four-level tree structure. In this strategy, each level is partitioned to optimize communications and the load balancing of computations. Specifically, both boxes and their samples

(along the θ -direction) are partitioned among processors, and the partitioning is determined by load-balancing algorithms. Usually, the lowest level is partitioned only along boxes, without partitioning samples. Then, the partitioning is changed accordingly at the higher levels, depending on the optimizations. Typically, as depicted in Fig. 1(c),

the number of partitions along boxes/samples is decreased/increased by a factor of two from one level to the next higher level. Changing the partitioning between levels bears an additional cost, but this is negligible in comparison to the improved load-balancing and reduced communications. Advantages of the hierarchical strategy are detailed in Section II-B.

B. Advantages of the Hierarchical Strategy

The major advantage of the hierarchical strategy is the improved load balancing due to partitioning both boxes and their samples. Computations are distributed almost equally among processors at all levels. Although less obvious, the hierarchical strategy also decreases communications between processors. Changing the partitioning between levels leads to a new type of communication, i.e., data exchanges, but in fact, the overall data transfer is significantly reduced. Theoretical bounds for communications in the hierarchical strategy are given in [14]; in this paper, we will present experimental comparisons.

Table 1 lists all types of communications required for the solution of a scattering problem involving a conducting sphere of radius 20λ , where λ is the wavelength in the host medium. The problem is discretized with 1 462 854 unknowns and the solution via a seven-level MLFMA is parallelized into 64 processes using the hybrid and hierarchical strategies. In the hybrid strategy, boxes are partitioned in the lowest four levels ($l = 1, 2, 3, 4$) and samples

are partitioned in the highest three levels ($l = 5, 6, 7$). In the hierarchical strategy, the numbers of partitions of boxes and samples are 64×1 , 64×1 , 32×2 , 16×4 , 8×8 , 4×16 , and 2×32 for $l = 1, 2, \dots, 7$, respectively. Table 1 lists both the number of communication events and the amount of communications (in bytes) between the processors in various categories, i.e., interpolations, data exchanges and partitioning switches, and translations at different levels. Not all types of communications are required for both strategies; for example, data exchanges are required only for the hierarchical strategy to change partitioning between levels, whereas the hybrid strategy requires a partitioning switch at an intermediate level. Further, the hybrid strategy does not require communications for the interpolations at the lower levels or for the translations at the higher levels, as expected.

In addition to the communications in different categories, Table 1 lists the number of communication events and amount of communications in the aggregation, translation, and disaggregation stages, as well as the overall values for a matrix-vector multiplication. Comparing the values for the hybrid and hierarchical strategies, the following conclusions can be drawn.

- The number of communication events is reduced by 54% (from 25 335 to 11 611) using the hierarchical strategy instead of the hybrid strategy.
- The total amount of communications is reduced by 31% (from 6 112 844 to 4 241 784 B) using the hierarchical strategy instead of the hybrid strategy.
- The average package size is increased from 241 to 365 B. A larger package size (without increasing the overall volume) means more effective communication.

As shown in this example, the hierarchical strategy not only improves load balancing but also reduces the amount of communications.

Finally, in addition to reduced communications, the hierarchical strategy allows for processor rearrangements to communicate faster. To demonstrate this, we consider again the four-level tree structure parallelized among eight processors. Fig. 2 shows the lowest three levels and a hypothetical distribution of the processes in two processor packages, each involving four cores. Considering this arrangement of the cores, it can be shown that all communications during the aggregation and disaggregation stages are performed between pairs of cores that are located in the same processor package. Note that communications between the cores are faster if the cores are located in the same package. Using the hierarchical strategy, it is relatively easy to rearrange processes such that most of the communications are between the cores that are physically close to each other. This is an important advantage, particularly for the recently developed architectures involving multicore/multiprocessor nodes and highly nonuniform communication rates between processors.

Table 1 Communications During a Matrix-Vector Multiplication for the Solution of a Scattering Problem Involving a Sphere Discretized With 1 462 854 Unknowns on 64 Processors

	Hybrid		Hierarchical	
	Number of Events	Amount (Bytes)	Number of Events	Amount (Bytes)
Interp. at $l = 7$	1360	646 272	136	107 712
Interp. at $l = 6$	6960	1 030 080	292	108 040
Interp. at $l = 5$	0	0	590	148 680
Interp. at $l = 4$	0	0	1170	182 520
Data Exc. at $l = 5$	0	0	2	55 552
Data Exc. at $l = 4$	0	0	2	73 920
Data Exc. at $l = 3$	0	0	2	106 560
Data Exc. at $l = 2$	0	0	2	122 640
Data Exc. at $l = 1$	0	0	2	213 304
Partitioning Switch	1160	171 680	0	0
Aggregation	9480	1 848 032	2198	1 118 928
Disaggregation	9480	1 848 032	2198	1 118 928
Trans. at $l = 7$	0	0	56	111 104
Trans. at $l = 6$	0	0	148	139 120
Trans. at $l = 5$	0	0	235	156 288
Trans. at $l = 4$	425	749 700	427	179 340
Trans. at $l = 3$	910	615 160	1005	313 560
Trans. at $l = 2$	1812	587 088	1861	602 964
Trans. at $l = 1$	3228	464 832	3483	501 552
Translation	6375	2 416 780	7215	2 003 928
Overall MVM	25 335	6 112 844	11 611	4 241 784
Average Package	241 Bytes		365 Bytes	
Interp.: Interpolation, Data Exc.: Data Exchange, Trans.: Translation, MVM: Matrix-Vector Multiplication.				

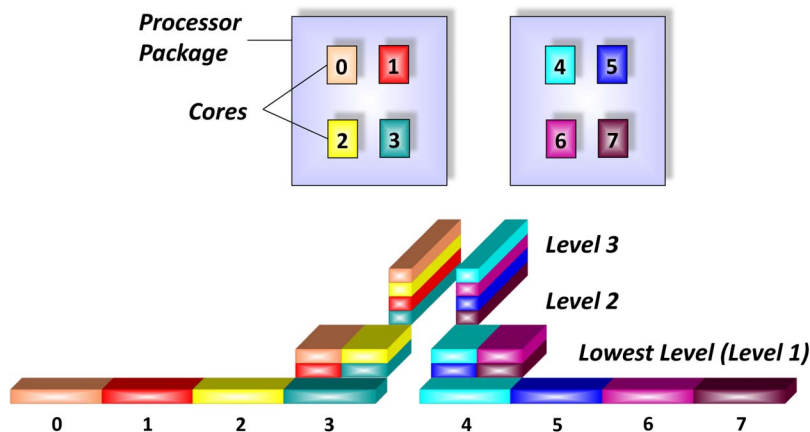


Fig. 2. Hierarchical partitioning of the tree structure in Fig. 1 among eight processors. Using the hierarchical strategy, communications during the aggregation and disaggregation stages can be performed faster since they are between the cores that are physically close to each other.

In summary, compared to other parallelization strategies, the hierarchical strategy improves the load balancing, reduces the amount of communications, and allows for faster communications via processor arrangements. Further comparisons of the hierarchical strategy with the simple and hybrid strategies, especially in terms of the parallelization efficiency, can be found in [8] and [14].

III. MEMORY CONSIDERATIONS

With the hierarchical strategy, the available memory is used very efficiently thanks to the improved load balancing. On the other hand, parallel implementations of MLFMA usually have complicated memory footprints. Dynamic memory allocations and deallocations are essential to recycle the used memory as much as possible. As predicted by Amdahl's law, sequential data structures, particularly those that are allocated in the initial stages of implementations, often become bottlenecks as the problem size grows and more processors are used. Three simple rules are used repetitively to avoid stagnations.

- Rule 1: Allocate memory for a data structure just before it is required.
- Rule 2: Deallocate memory used for a data structure as soon as it becomes useless so that it can be used later in the program.
- Rule 3: Rearrange the program such that Rules 1 and 2 can be further applied.

Rule 3 is particularly useful to reduce memory peaks before iterations and matrix-vector multiplications, which are parallelized very efficiently with the hierarchical strategy.

Code rearrangements for memory recycling depend on the implementation and, to the best of our knowledge, no common procedure exists. We again present an experimental demonstration of how these rearrangements and the resulting memory recycling can be effective to reduce

peak memory. Fig. 3 depicts memory recycling based on code rearrangements on a very large scattering problem involving a sphere of radius 260λ discretized with 307 531 008 unknowns. The solution of the problem is parallelized into 128 processes using the hierarchical strategy. Fig. 3(a) and (b) presents the memory required for each process as a function of time steps before and after memory recycling is applied. In both cases, the memory required for the master process is quite different than those of other processes due to some initial sequential operations for data input and management. Most importantly, there are memory peaks in all processes before the matrix-vector multiplications are performed. These peaks are the major bottlenecks before memory recycling strategy is applied. Using code rearrangements and memory recycling, the peak memory of the implementation is significantly reduced (from 12.6 to 10.4 GB), as depicted in Fig. 3(b). This means that larger problems can now be solved by using the same memory.

IV. NUMERICAL RESULTS

To demonstrate the efficient parallelization of MLFMA with the hierarchical partitioning strategy, we consider the solution of scattering problems involving a conducting sphere of radius 20λ . The sphere is discretized with 1 462 854 unknowns, and the scattering problem is solved on various parallel computers, listed in Table 2. All computers involve computing nodes that are connected via Infiniband networks and Intel Xeon processors with different clock rates. MVAPICH is used as the message passing interface (MPI) version. In addition, Intel Math Kernel Library (MKL) and Portable-Extensible Toolkit for Scientific Computation (PETSc) are used for mathematical functions and iterative solutions. Table 3 presents the setup, iterative solution, and total computation times for the solution of the scattering problem on these computers

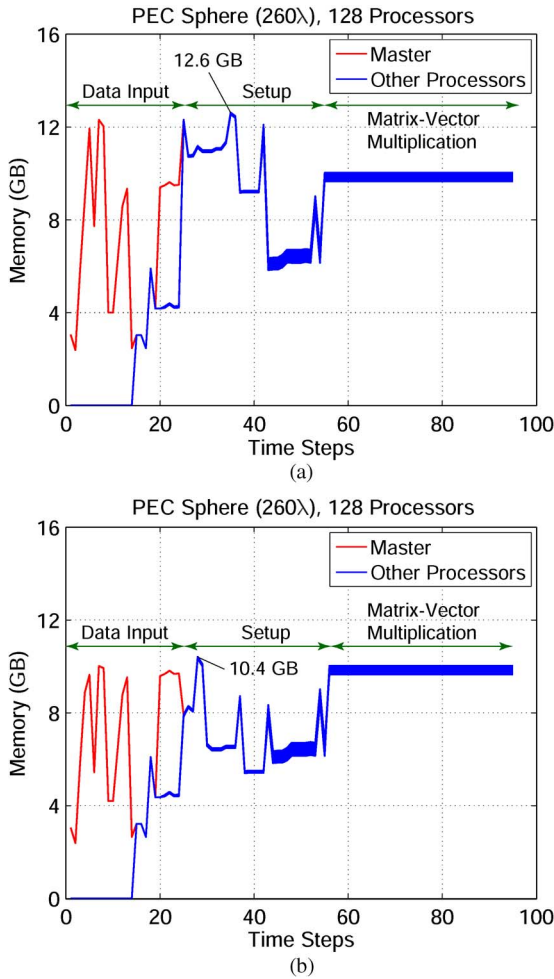


Fig. 3. Memory used for the solution of a scattering problem involving a sphere of radius 260λ discretized with 307 531 008 unknowns. The solution of the problem is parallelized into 128 processes using the hierarchical strategy. Memory for each process is plotted as a function of time steps (a) before and (b) after code rearrangements and memory recycling.

using different numbers of nodes and processors per node. The setup time includes (and is dominated by) the computation of the near-field interactions. The iterative solution time includes 27 iterations (54 matrix-vector multiplications) to reduce the residual error to below 10^{-6} . Finally, the total time includes the setup and iterative solution

Table 2 Parallel Computers With Distributed-Memory Architectures Used for Numerical Tests

Computer	Nodes	Processors Per Node	Memory Per Node
C-Clovertown	128	2 × E5345 (2.33 GHz)	16 GB
B-Harpertown	16	2 × X5472 (3.00 GHz)	32 GB
U-Nehalem	64	2 × X5550 (2.67 GHz)	24 GB
S-Nehalem	16	2 × X5560 (2.80 GHz)	32 GB
N-Nehalem	16	2 × X5560 (2.80 GHz)	48 GB
S-Westmere	16	2 × X5670 (2.93 GHz)	48 GB
S-Dunnington	16	4 × E7450 (2.40 GHz)	48 GB

Table 3 Solutions of a Scattering Problem Involving a Sphere Discretized with 1 462 854 Unknowns on Different Numbers of Processors, Using the Computers in Table 2

Computer	Number of Nodes × Number of Processes per Node					
	1 × 1	4 × 1	16 × 1	16 × 2	16 × 4	16 × 8
Setup Time (seconds)						
C-Clovertown	12560	3203	817	411	210	109
B-Harpertown	7767	1851	467	242	120	63
U-Nehalem	8259	2095	534	269	137	73
S-Nehalem	7950	2058	557	279	141	70
N-Nehalem	8467	2111	529	263	133	65
S-Westmere	7294	1844	472	237	120	62
S-Dunnington	9114	2348	591	297	154	80
Iterative Solution Time (seconds)						
C-Clovertown	6907	1892	563	319	201	150
B-Harpertown	5510	1564	456	259	168	142
U-Nehalem	5189	1456	432	255	167	135
S-Nehalem	4910	1498	455	232	140	122
N-Nehalem	5542	1572	459	243	147	93
S-Westmere	4388	1311	385	208	130	95
S-Dunnington	6634	2604	911	477	363	350
Total Time (seconds)						
C-Clovertown	19474	5103	1388	738	420	270
B-Harpertown	12679	3419	926	505	292	209
U-Nehalem	13462	3565	980	539	316	219
S-Nehalem	12862	3566	1023	522	291	203
N-Nehalem	14011	3686	991	508	282	161
S-Westmere	11684	3157	867	447	253	162
S-Dunnington	15751	4956	1507	778	525	440

times, as well as data input and management. It can be seen that the total time of the fastest solution is 161 s on 128 processors (of N-Nehalem).

As discussed in [14], the parallelization efficiency and speedup values can be misleading when comparing parallel implementations. This is mainly because the parallelization efficiency and speedup do not give complete information on the actual efficiency, i.e., the computation time. Specifically, a very slow implementation can be “embarrassingly” parallelizable while a faster implementation can be parallelized less efficiently. Accuracy of solutions (which is often relaxed or omitted) is another parameter that must accompany the time measurements [25]. Hence, we emphasize that the results presented in Table 2 are obtained efficiently (with many efforts to minimize the processing time) and accurately (with maximum 1% error in the scattered fields). For these accurate and efficient solutions, the parallelization efficiency is

$$\epsilon_p = \frac{t_1}{128 \times t_{128}} = 68\% \quad (1)$$

which corresponds to an 87-fold speedup, on the 128 processors of the N-Nehalem computer.

Finally, Fig. 4 presents the solution of a large scattering problem involving a conducting sphere of radius 210λ discretized with 204 823 296 unknowns. The problem is solved with maximum 1% target error on the N-Nehalem

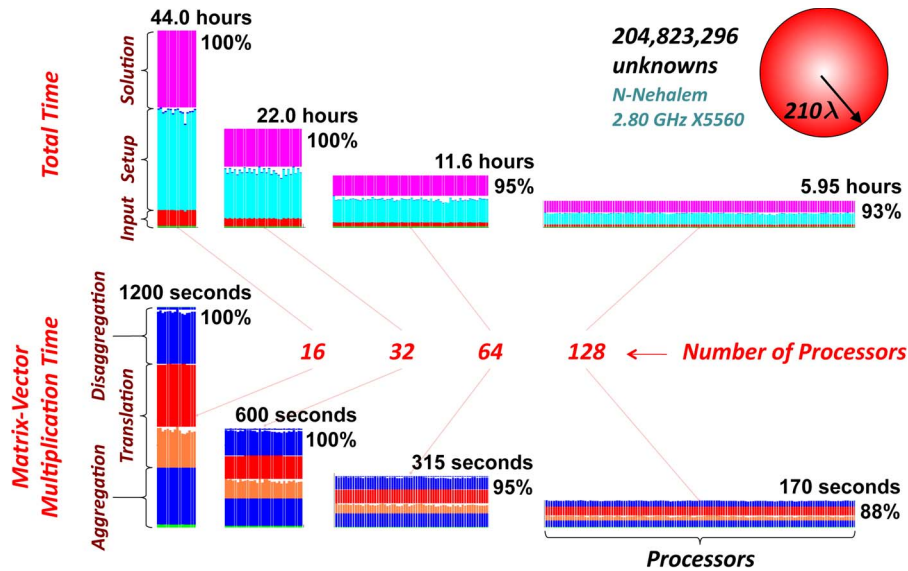


Fig. 4. Solutions of a scattering problem involving a conducting sphere of radius 210λ discretized with 204 823 296 unknowns. The total time and a matrix-vector multiplication time are plotted for all processors, when the solution is parallelized into 16, 32, 64, and 128 processes.

computer using 16, 32, 64, and 128 processors. Fig. 4 depicts both the total time and a matrix-vector multiplication time. In addition to the time measurements, the workload is shown as a function of time for each solution and all processors. We observe that the total time is reduced from 44 to 5.95 h (corresponding to 93% parallelization efficiency), while the matrix-vector multiplication time is reduced from 1200 to only 170 s (corresponding to 88% parallelization efficiency). Fig. 5 depicts the normalized bistatic radar cross section (RCS, in decibels) of the sphere from 0° to 180° . RCS values around the backscattering (0°) and forward-scattering (180°) directions are

focused in separate plots. Computation results obtained with MLFMA agree very well with the analytical Mie-series solution. The relative error in the computational values with respect to the Mie-series solution is found to be 1.20%, 0.90%, and 0.71% in the 0° – 30° , 0° – 90° , and 0° – 180° intervals, respectively. These errors are in agreement with the target 1% error of the solutions.

V. CONCLUDING REMARKS

This paper presents the hierarchical parallelization of MLFMA for rigorous solutions of large-scale electromagnetics

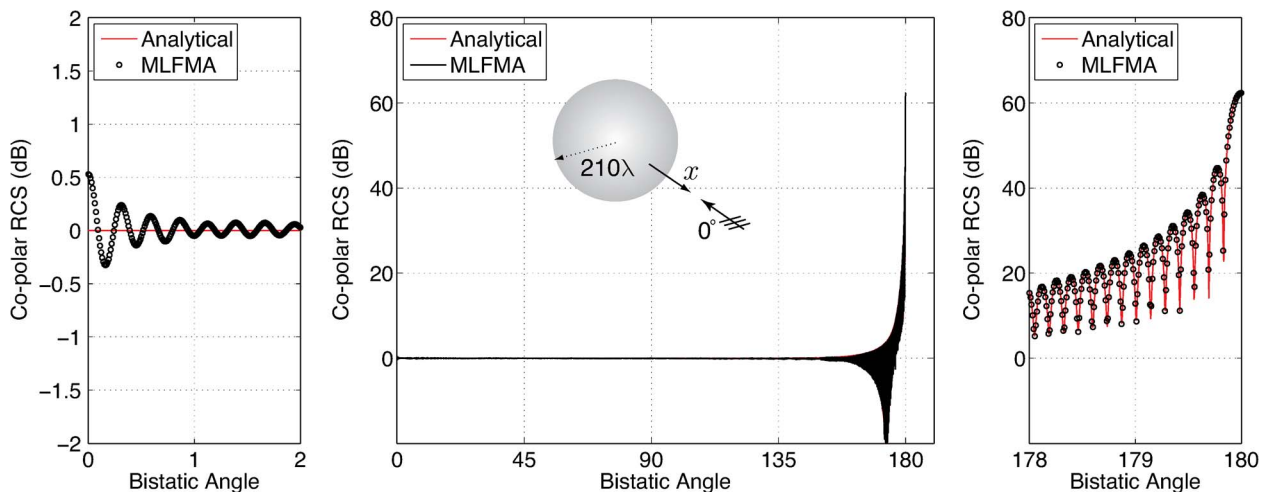


Fig. 5. Bistatic RCS (in decibels) of a conducting sphere of radius 210λ . Computational values obtained with MLFMA agree well with the analytical Mie-series solution.

problems. As discussed in detail, the hierarchical strategy has three important advantages over the previous approaches:

- improved load balancing with nearly equal distribution of the workload among processors;
- reduced amount of communications and more effective communications with larger data packages;

- faster communications due to localized (intraprocessor and/or intranode) data transfers.

Efficient parallelization of MLFMA using the hierarchical strategy translates into an ability to use more memory and to solve larger and more realistic problems with the available computing resources, as also demonstrated in [19] and [29]. ■

REFERENCES

- [1] 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.
- [2] X.-Q. Sheng, J.-M. Jin, J. Song, W. C. Chew, and C.-C. Lu, "Solution of combined-field integral equation using multilevel fast multipole algorithm for scattering by homogeneous bodies," *IEEE Trans. Antennas Propag.*, vol. 46, no. 11, pp. 1718–1726, Nov. 1998.
- [3] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.
- [4] S. Velamparambil, 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] F. Wu, Y. Zhang, Z. Z. Oo, and E. Li, "Parallel multilevel fast multipole method for solving large-scale problems," *IEEE Antennas Propag. Mag.*, vol. 47, no. 4, pp. 110–118, Aug. 2005.
- [6] S. Velamparambil 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.
- [7] L. Gürel and Ö. Ergül, "Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns," *Electron. Lett.*, vol. 43, no. 9, pp. 499–500, Apr. 2007.
- [8] 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.
- [9] X.-M. Pan and X.-Q. Sheng, "A sophisticated parallel MLFMA for scattering by extremely large targets," *IEEE Antennas Propag. Mag.*, vol. 50, no. 3, pp. 129–138, Jun. 2008.
- [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] J. Fostier and F. Olyslager, "Provably scalable parallel multilevel fast multipole algorithm," *Electron. Lett.*, vol. 44, no. 19, pp. 1111–1113, Sep. 2008.
- [13] J. Fostier and F. Olyslager, "Full-wave electromagnetic scattering at extremely large 2-D objects," *Electron. Lett.*, vol. 45, no. 5, pp. 245–246, Feb. 2009.
- [14] Ö. Ergül and L. Gürel, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1740–1750, Jun. 2009.
- [15] M. G. Araujo, J. M. Taboada, F. Obelleiro, J. M. Bertolo, L. Landesa, J. Rivero, and J. L. Rodriguez, "Supercomputer aware approach for the solution of challenging electromagnetic problems," *Progr. Electromagn. Res.*, vol. 101, pp. 241–256, 2010.
- [16] J. M. Taboada, M. G. Araujo, J. M. Bertolo, L. Landesa, F. Obelleiro, and J. L. Rodriguez, "MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetics," *Progr. Electromagn. Res.*, vol. 105, pp. 15–30, 2010.
- [17] J. Fostier and F. Olyslager, "An open-source implementation for full-wave 2D scattering by million-wavelength-size objects," *IEEE Antennas Propag. Mag.*, vol. 52, no. 5, pp. 23–34, Oct. 2010.
- [18] X.-M. Pan, W.-C. Pi, and X.-Q. Sheng, "On OpenMP parallelization of the multilevel fast multipole algorithm," *Progr. Electromagn. Res.*, vol. 112, pp. 199–213, 2011.
- [19] Ö. Ergül and L. Gürel, "Rigorous solutions of electromagnetics problems involving hundreds of millions of unknowns," *IEEE Antennas Propag. Mag.*, vol. 53, no. 1, pp. 18–27, Feb. 2011.
- [20] V. Melapudi, B. Shanker, S. Seal, and S. Aluru, "A scalable parallel wideband MLFMA for efficient electromagnetic simulations on large scale clusters," *IEEE Trans. Antennas Propag.*, vol. 59, no. 7, pp. 2565–2577, Jul. 2011.
- [21] J. Fostier, B. Michiels, I. Bogaert, and D. De Zutter, "A fast 2-D parallel multilevel fast multipole algorithm solver for oblique plane wave incidence," *Radio Sci.*, vol. 46, RS6006, Nov. 2011.
- [22] Ö. Ergül, "Solutions of large-scale electromagnetics problems involving dielectric objects with the parallel multilevel fast multipole algorithm," *J. Opt. Soc. Amer. A.*, vol. 28, no. 11, pp. 2261–2268, Nov. 2011.
- [23] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comput. Phys.*, vol. 86, no. 2, pp. 414–439, Feb. 1990.
- [24] 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.
- [25] Ö. Ergül and L. Gürel, "Accuracy: The frequently overlooked parameter in the solution of extremely large problems," in *Proc. Eur. Conf. Antennas Propag.*, 2011, pp. 3928–3931.
- [26] Ö. Ergül and L. Gürel, "Benchmark solutions of large problems for evaluating accuracy and efficiency of electromagnetics solvers," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 2011, pp. 179–182.
- [27] BiLCEM Benchmarking Tool, Mar. 17, 2012. [Online]. Available: <http://www.cem.bilkent.edu.tr/benchmark>
- [28] 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.
- [29] Ö. Ergül and L. Gürel, "Accurate solutions of extremely large integral-equation problems in computational electromagnetics," *Proc. IEEE*, 2012, DOI: 10.1109/JPROC.2012.2204429.

ABOUT THE AUTHORS

Levent Gürel (Fellow, IEEE) received the B.Sc. degree from the Middle East Technical University (METU), Ankara, Turkey, in 1986 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 1988 and 1991, respectively.

He is the Director of the Computational Electromagnetics Research Center (BiLCEM), Bilkent University, Ankara, Turkey. 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, where he is currently a Professor. He was a Visiting Associate Professor at the Center for Computational Electromagnetics (CEM) of the 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. 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 structures, 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. Bioelectromagnetics, remote sensing, 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.

Prof. Gürel is a Fellow of the Applied Computational Electromagnetics Society (ACES) and the Electromagnetics Academy (EMA). Among his most notable recognitions are two prestigious awards: the 2002 Turkish Academy of Sciences (TUBA) Award and the 2003 Scientific and Technical Research Council of Turkey (TUBITAK) Award. He is named an IEEE Distinguished Lecturer for 2011-2013. He is currently serving as an Associate Editor for *Radio Science*, the IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS, *Journal of Electromagnetic Waves and Applications*, and *Progress in Electromagnetics Research*. 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 a member of the ACES Board of Directors and served as the Guest Editor for a special issue of the *ACES Journal*. He was invited to address the 2011 ACES Conference as a Plenary Speaker. 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 Co-Chairman of the 2003 IEEE International Symposium on Electromagnetic Compatibility. He was the organizer and General Chairman of the 2007, 2009, and 2011 Computational Electromagnetics International Workshops held in Izmir, Turkey.

Özgür Ergül (Senior Member, IEEE) received the B.Sc., M.S., and Ph.D. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001, 2003, and 2009, respectively.

He is currently a Lecturer in the Department of Mathematics and Statistics, University of Strathclyde, Glasgow, U.K. He is also a Lecturer of the Centre for Numerical Algorithms and Intelligent Software (NAIS). He served as a Teaching and Research Assistant in the Department of Electrical and Electronics Engineering, Bilkent University, from 2001 to 2009. He was also affiliated with the Computational Electromagnetics Group, Bilkent University, from 2000 to 2005 and with the Computational Electromagnetics Research Center (BiLCEM) from 2005 to 2009. He is the coauthor of 150 journal and conference papers. His research interests include fast and accurate algorithms for the solution of electromagnetics problems involving large and complicated structures, integral equations, parallel programming, iterative methods, and high-performance computing.

Dr. Ergül is a recipient of the 2007 IEEE Antennas and Propagation Society Graduate Fellowship, the 2007 Leopold B. Felsen Award for Excellence in Electrodynamics, the 2010 Serhat Özyar Young Scientist of the Year Award, and the 2011 International Union of Radio Science (URSI) Young Scientists Award.

