# EFFICIENT PARALLELIZATION OF MULTILEVEL FAST MULTIPOLE ALGORITHM

**Özgür Ergül**[1] **and Levent Gürel**[1,2]

[1]*Department of Electrical and Electronics Engineering, Bilkent University, TR-06800, Ankara, Turkey*
[2]*Computational Electromagnetics Research Center (BiLCEM), Bilkent University, TR-06800, Ankara, Turkey*
*Email: ergul@ee.bilkent.edu.tr, lgurel@bilkent.edu.tr*

## ABSTRACT

We report our efforts for the solution of large electromagnetics problems accurately and efficiently with the parallel multilevel fast multipole algorithm. We carefully investigate different stages of the parallelization and identify the bottlenecks to develop new strategies. The required modifications are implemented in order to increase the efficiency of the solutions of scattering problems involving various geometries.

Key words: Multilevel fast multipole algorithm; parallelization.

## 1. INTRODUCTION

We consider the solution of large three-dimensional problems of computational electromagnetics with the multilevel fast multipole algorithm (MLFMA) [1]. Accurate solutions of quite large problems can be performed iteratively with the accelerated matrix-vector products by MLFMA since this algorithm reduces the complexity of the processing time and the memory requirement to $O(N \log N)$. However, most of the real-life applications involve large geometries that can be modelled with millions of unknowns. In addition, solutions of these complex problems may require large numbers of iteration steps for convergence to desired levels of accuracy. Finally, electromagnetics problems are usually required to be solved at multiple frequencies and for different excitations involving various illumination angles and polarizations. As a consequence, in addition to the advances in the solution algorithms, it becomes essential to take advantage of the advances in the computer hardware in order to solve these large and complex problems. For this purpose, we construct parallel clusters of relatively inexpensive processors connected via special fast networks.

Since MLFMA requires the evaluation of all electromagnetic interactions, it is not trivial to distribute a problem among the processors. Communications between the processors and duplications of the computations are known to reduce the efficiency of the parallelization. A good parallelization strategy should take into consideration every potential bottleneck of the algorithm. For this reason, we investigate the separate parts of the matrix-vector products to understand and solve the inefficiencies. Specifically, we use the shared-level concept proposed in [2] to distribute the work load among the processors in a better way. To investigate the efficiency of the techniques applied for parallelization, we carefully analyze the processes performed in each processor. This way, it becomes possible to detect the bottlenecks of the program and improve the parallelization by simple modifications.

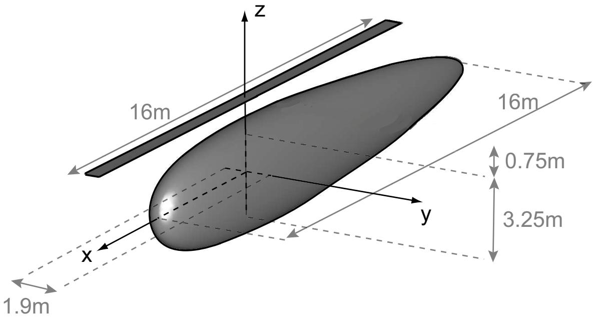


*Figure 1. A geometry including a long strip over an ellipsoid, which is used in the tests of the parallelization of MLFMA.*

In this paper, we consider the solutions of scattering problems involving three different geometries. The first geometry is a sphere of radius $6\lambda$ leading to a matrix equation with 132,003 unknowns. The second geometry is depicted in Fig. 1, where a long strip is placed above an ellipsoid with the dimensions given in the picture. This problem is solved at 500 MHz and has 117,366 unknowns. Finally, as an example to geometries with large dimension in one direction, we solve the strip in Fig. 1 alone at 1 GHz leading to a 62,961-unknown problem. All three problems have moderate numbers of unknowns so that they can be solved by a single processor for speedup measurements.

*Proc. 'EuCAP 2006', Nice, France*
*6–10 November 2006 (ESA SP-626, October 2006)*

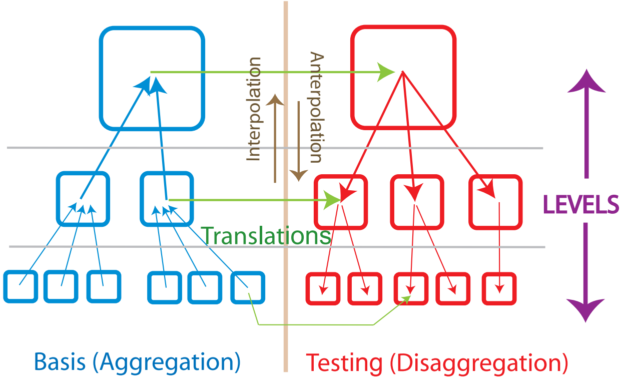


*Figure 2. Multilevel tree structure of MLFMA.*

## 2. STRUCTURE OF MLFMA

Using the physical setting of moment methods, a matrix-vector multiplication can be regarded as a set of interactions between the basis and testing functions. In order to perform these interactions in a group-by-group manner, the whole geometry is placed into a large cube and it is recursively divided into smaller ones until the smallest cubes contain only a few basis or testing functions. During the partitioning, if any of the cubes become empty, recursion is stopped there. As depicted in Fig. 2, a tree structure is formed. This provides a hierarchical representation of the computational domain. Nonempty cubes in the MLFMA tree are referred to as clusters. The fundamental idea in MLFMA is to replace element-to-element interactions with cluster-to-cluster interactions in a multilevel scheme. This computational scheme relies on the factorization of the Green's function, which is valid only for basis and testing functions that are far from each other. In the lowest level, interactions between the near-field clusters are computed directly and stored in a sparse matrix. Interactions among the far-field clusters are computed approximately using MLFMA in the following way [1]. First, the radiated fields of each cluster are aggregated at the centers of the clusters. Then, for each pair of far-field clusters, whose parents are near to each other, cluster-to-cluster interaction is computed via a translation. Finally, after the translations, the matrix-vector multiplication is completed by disaggregating the incoming fields to the centers of the testing clusters and onto the testing functions.

An MLFMA implementation can be divided into two parts. In the first part, the required data for the matrix-vector multiplications are computed and stored in the memory. The data include the tree structure and clustering of MLFMA, near-field interactions, Fourier transforms for the basis and testing functions, and translation matrices for cluster-cluster interactions. In the second part, the problem is solved by an iterative algorithm employing MLFMA for matrix-vector multiplications. In each matrix-vector multiplication, aggregation, translation, and disaggregation steps are performed on the tree structure. In addition, preconditioners are applied to accelerate the iterative convergence so that a solution involving the preconditioner matrix is required in each iteration.

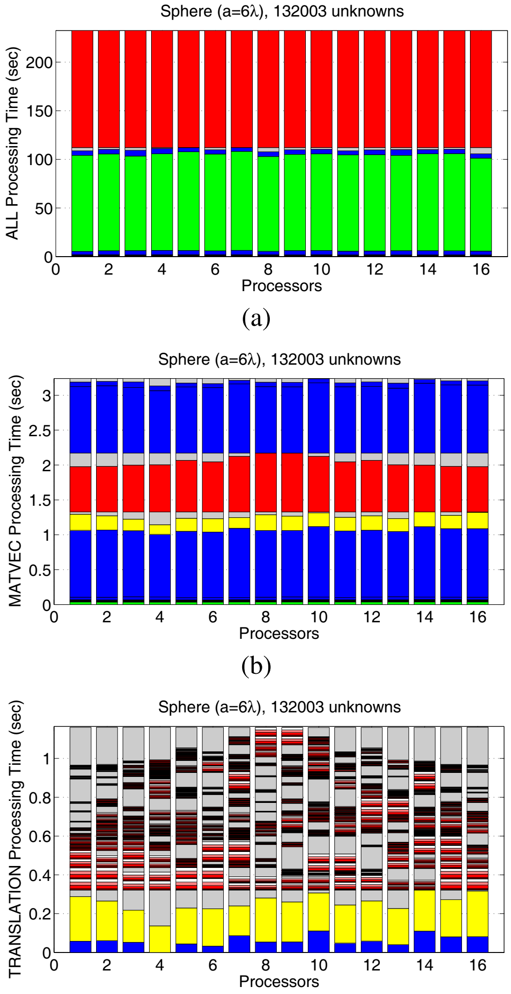


*Figure 3. Processing time for the solution of the sphere problem on a cluster of 16 processors. (a) Total time including the setup and iterative solution, (b) a matrix-vector multiplication, and (c) translations in a matrix-vector multiplication.*

## 3. SIMPLE PARALLELIZATION OF MLFMA

In the parallelization of MLFMA, near-field interactions are distributed among the processors in a row-wise manner. Even for the problems involving symmetric geometries, such as a sphere, it is essential to apply load-balancing for the near-field interactions instead of equally distributing the rows of the system matrix [3]. Fig. 3 presents the solution of a scattering problem involving a sphere of radius $6\lambda$ on a cluster of 16

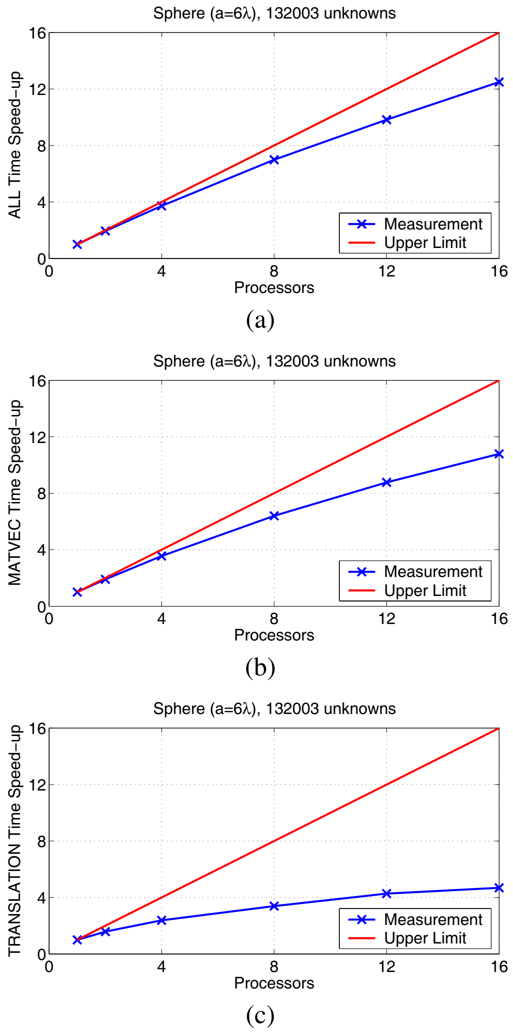


*Figure 4. Speedup in the solution of the sphere problem for (a) total time including the setup and iterative solution, (b) a matrix-vector multiplication, and (c) translations in a matrix-vector multiplication.*

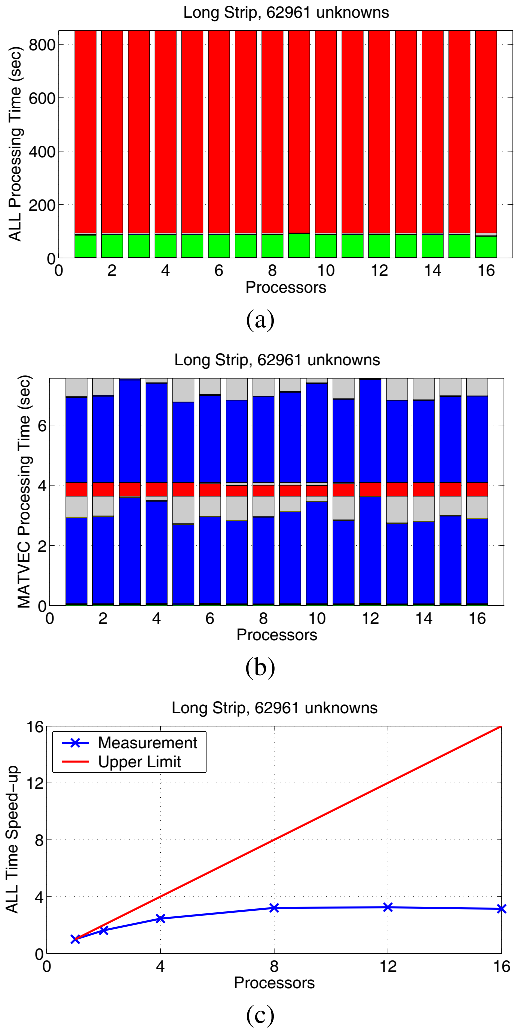


*Figure 5. (a) Total processing time, (b) a matrix-vector multiplication, and (c) overall speedup for the solution of the strip problem on a cluster of 16 processors.*

processors[1]. The entire program flow is depicted in Fig. 3(a), where red represents the iterative solution completed in 18 iterations, green represents the calculation of near-field interactions before the iterations, and blue parts represent the setups for Fourier transforms and translations. Due to a good load-balancing of the near-field interactions, the processors wait for a short time represented by gray colors in Fig. 3(a) before the iterative solution.

When a simple parallelization technique is applied to MLFMA, the clusters are distributed and assigned to the processors. In this case, a level (level of distribution) is chosen to share the clusters equally among the processors. If a cluster belongs to a processor, then all of its sub-clusters in the lower levels are also assigned to the same processor. This way, the aggregation and disaggregation steps from the bottom of the tree up to the level of distribution can be performed without any communication. On the other hand, for the upper levels above the level of distribution, there can be two schemes. In the first scheme, the clusters in the levels are still distributed among the processors equally as much as possible. Then, a cluster and its parent cluster may belong to different processors. As a consequence, aggregation and disaggregation steps involve communications for high levels of the tree structure. Rather than this scheme, we prefer the second one, where the clusters in the levels above the level of distribution are allowed to be duplicated in different processors, if required. This way, all the aggregation and disaggregation steps can be performed without any communication. When the radiated fields are translated into incoming fields, the contributions from different processors are combined automatically for the duplicated clusters.

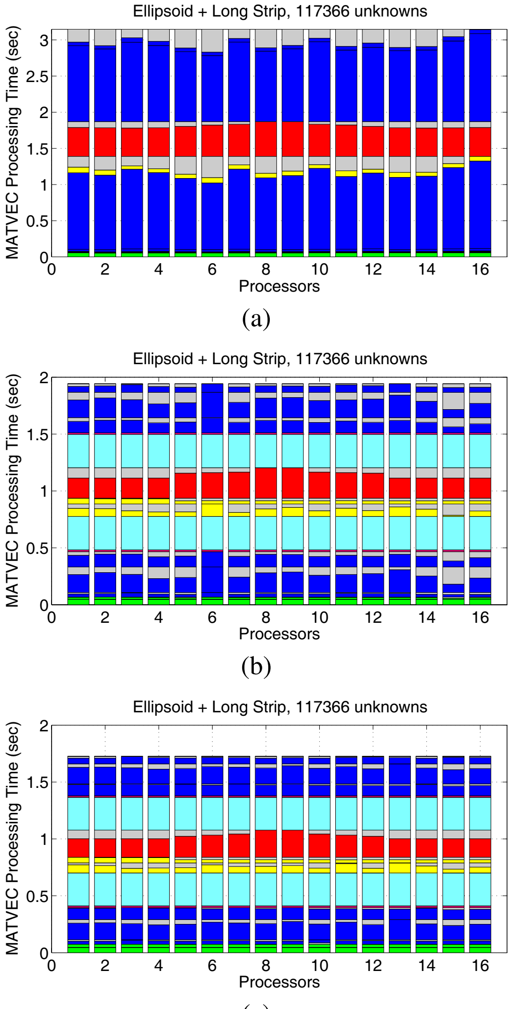
Fig. 3(b) presents a single matrix-vector multipli-

[1] All the solutions presented in this paper are obtained on an 8-way SMP server with dual-core AMD Opteron processors.

cation, where blue represents the aggregation and disaggregation steps, gray represents waiting periods, and the translations are represented by yellow and red parts. We also note that the near-field interactions are shown by green color, which is negligible compared to the other parts, in spite of its long setup depicted in Fig. 3(a). In the simple parallelization of MLFMA, the translations are important since they easily become the bottleneck of the program. As presented in Fig. 3(b), there are two stages of translations. In the first stage represented by yellow, the translations related to the clusters that are assigned to the same processor are completed without any communication. Following the translations inside each processor, the second stage is performed, which is represented by red blocks in Fig. 3(b) and requires dense communications among the processors. In general, each processor has radiation data to be sent to all other processors. We carefully handle the data traffic by using a fixed communication map between the processors. For $p$ processors, all of the communications are completed in about $p$ steps, each of which includes simultaneous exchanges of data between pairs of coupled processors. When the processors are coupled, data set of a cluster is transferred from the sender to the receiver if it is required. After a transfer, the receiver processor completes all the translations related to the transferred data. This way, it becomes possible to avoid transferring the same data more than once.

Fig. 3(c) depicts a translation stage for the sphere problem, where the red block in Fig. 3(b) is expanded in detail. In the figure, the gray parts indicate the waiting periods between the synchronization and coupling of the processors for data transfer. When the processors are coupled, the data transfer and translation operations are performed simultaneously for all levels, from top to bottom, represented by colors in red tones. Translations for low and high levels are represented by light and dark colors, respectively. Then, the lowest and highest levels are shown by white and black, respectively. Although the inefficiency of the translations in parallel MLFMA is usually attributed to the communications in high levels [1], this is not the case for the sphere problem as shown in Fig. 3(c). In other words, the dark colors in Fig. 3(c) do not dominate the light colors and this might be due to relatively fast network of our parallel system.

Fig. 4 presents the speedup of the parallel MLFMA for the solution of the sphere problem. The speedup for the entire program is over 12 for 16 processor as depicted in Fig. 4(a). The deviation from the ideal case is mostly due to the solution part, which involves the matrix-vector multiplications with the speedup given in Fig. 4(b). Finally, the speedup for the matrix-vector multiplications is about 11, mostly due to the low speedup of the translation stage depicted in Fig. 4(c). We are investigating how to increase the speedup of the translations by designing better maps for data traffic in order to reduce the waits shown as gray blocks in Fig. 3(c).



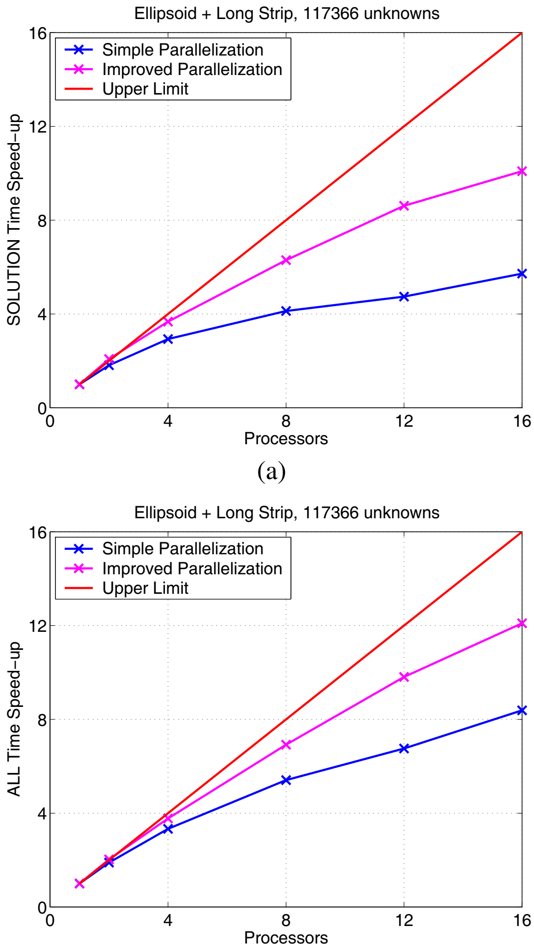
(a)

(b)

(c)

*Figure 6. Matrix-vector multiplication for the scattering problem of the geometry in Fig. 1 by (a) simple parallelization, (b) improved parallelization, and (c) improved parallelization including load-balancing.*

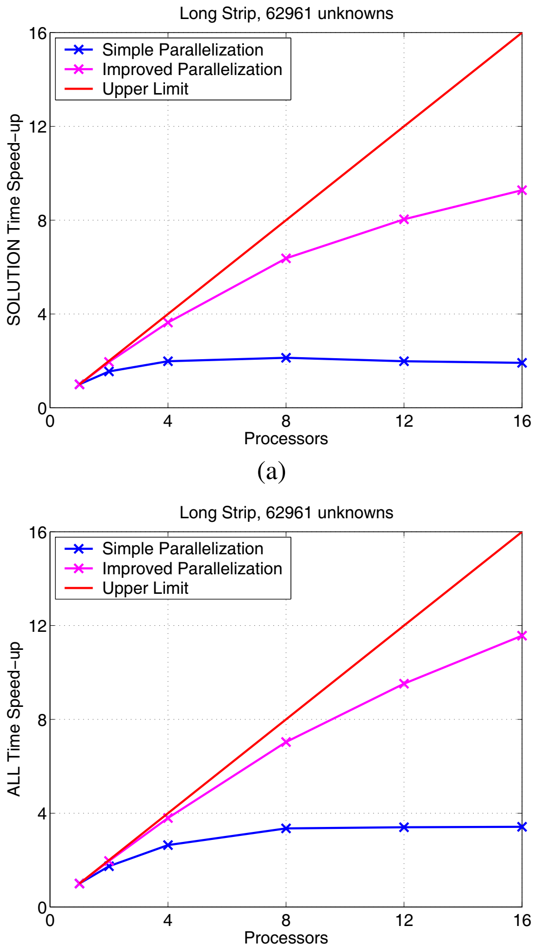
Although a simple parallelization of MLFMA is relatively successful for the solution of smooth problems (e.g., the sphere), it usually fails to provide an acceptable speedup for the solution of other problems, such as the long strip in Fig. 1. Fig. 5(a) depicts the solution of the problem, where the iterative part dominates the setup part although the number of iterations is only 50. In addition, when Fig. 5(b) is compared to Fig. 4(b), we observe that the time required for the matrix-vector multiplications is larger for the strip problem, even though it has fewer unknowns compared to the sphere. Finally, the inefficiency in the solution of the strip problem must be due to the aggregation and disaggregation steps since the blue parts are dominant in Fig. 4(b). Fig. 5(c) also supports the fact that a simple parallelization of MLFMA fails to provide an efficient solution for the strip problem since the overall speedup is below 4 for 16 processors.



(a)

(b)

*Figure 7. Comparisons of the speedups obtained by simple and improved parallelizations of MLFMA in the solution of a scattering problem involving the geometry in Fig. 1.*



(a)

(b)

*Figure 8. Comparisons of the speedups obtained by simple and improved parallelizations of MLFMA in the solution of a scattering problem involving the strip geometry.*

## 4. IMPROVED PARALLELIZATION OF MLFMA

The inefficiency of the simple parallelization presented in the previous section is due to the unbalanced tree structure of the complex geometries. When the clusters in these MLFMA trees are investigated, it can be observed that there are a few large clusters in the upper levels so that they cannot be shared among the processors with a good load balance. As a consequence, the aggregation and disaggregation steps cannot be performed efficiently. Therefore, in order to improve the load-balancing, we have developed a parallel implementation using the shared-level concept proposed in [2]. The improved parallelization involves the application of different distribution schemes for the lower and upper levels of the tree structure. For the low levels of MLFMA, we still apply the simple strategy and distribute the clusters among the processors. Then, at an appropriate level (transition level), we switch to another strategy via an all-to-all communication, where the radiated and incoming fields are distributed among the processors. Above the transition level, each cluster is shared by all processors and each processor has the same angular portion of the radiated and incoming fields for all clusters.

Distributing the field instead of the clusters in the high levels of MLFMA has two consequences:

1. The translations can be performed without any communication while the aggregation and disaggregation processes require one-to-one communications.
2. In the high levels of MLFMA, there are a few large clusters with highly-sampled radiated and incoming fields. Therefore, load-balancing is improved since the distribution of the samples is easier than the distribution of clusters among the processors.

Fig. 6 presents the matrix-vector multiplications for the scattering problem involving the geometry in Fig. 1.

The result of the simple parallelization is depicted in Fig. 6(a), where the matrix-vector multiplication is completed in about 3 seconds. Employing the improved parallelization, the multiplication can be performed in 2 seconds as depicted Fig. 6(b), where the aggregation and disaggregation steps are divided in blue and cyan parts representing the distributed and shared levels, respectively. Finally, Fig. 6(c) shows that simple load-balancing algorithms can be employed to further improve the parallelization. We note that the processor loads in distributed and shared levels can be balanced independently and the transition level involving the all-to-all communications require negligible time.

Fig. 7 compares the speedups of the simple and improved parallelizations for the problem in Fig. 1. In the case of 16 processors, the speedup for the iterative solution increases from 6 to 10 as shown in Fig. 7(a). This leads to increase in the speedup for the overall time from 8 to 12 as shown Fig. 7(b). Finally, as depicted in Fig. 8, the improvement is more significant for the strip problem, where the speedups for the solution and overall processing times are increased from 2 to 9 and 4 to 12, respectively, in the case of 16 processors.

## 5. CONCLUSION

We present our efforts to obtain an efficient parallelization of MLFMA in order to solve large and complex problems of computational electromagnetics. Parallel solutions of various problems are investigated in detail to identify and eliminate the bottlenecks of the implementations.

## ACKNOWLEDGMENTS

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 105E172, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM.

## REFERENCES

[1] C.-C. Lu and W. C. Chew, “Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects,” *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[2] S. Velamparambil ve W. C. Chew, “Analysis and performance of a distributed memory multilevel fast multipole algorithm,” *IEEE Trans. Antennas Propag.*, vol. 53, pp. 2719–2727, Aug. 2005.

[3] L. Gürel and Ö. Ergül, “Solution of large CEM problems with parallel multi-level fast multipole algorithm (MLFMA),” *2005 IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting*, Washington, DC, USA, July 2005.