Spatial Partitioning Strategy for Parallelization of MLFMA with Reduced Communication

Xunwang Zhao [1], Chang Zhai [1], Zhongchao Lin [1], Yu Zhang [1], and Qifeng Liu [2]

[1] Shaanxi Key Laboratory of Large Scale Electromagnetic Computing
School of Electronic Engineering, Xidian University, Xi'an, 710071, China
xwzhao@mail.xidian.edu.cn

[2] Science and Technology on Electromagnetic Compatibility Laboratory
China Ship Development and Design Center, Wuhan 430064, China

Abstract — The bottleneck of the spatial partitioning for parallelizing the multilevel fast multipole algorithm (MLFMA) lies in higher levels of the tree, at which boxes are usually fewer than parallel processors, yielding a serious load imbalance. To solve the bottleneck, the higher levels of the tree are truncated to generate plenty of subtrees, which are distributed among processors to facilitate balancing the work load. At the coarsest level, the communication volume during translation between far-away processors is drastically reduced by adopting the far-field approximation. Therefore, the communication mainly occurs between nearby processors, which is favorable for modern computing clusters. In comparison with the parallel strategies that hybridize the spatial partitioning with the *k*-space partitioning, the proposed approach is more straightforward and shows good scalability.

Index Terms — Multilevel fast multipole algorithm (MLFMA), parallelization, reduced communication, spatial partitioning, subtrees.

I. INTRODUCTION

The multilevel fast multipole algorithm (MLFMA) is widely applied in the electromagnetic scattering analysis of electrically large objects. During last decade, high performance computing techniques have been used in order to boost its performance by designing efficient parallel strategies. Due to the use of a tree-like structure in the spatial domain and plane-wave expansions in the spectral (*k*-space) domain, the parallelization of MLFMA is much more complicated compared with other numerical methods such as the method of moments (MoM) [1] and the finite-difference time-domain (FDTD) method [2, 3]. Generally, researchers use two basic strategies when parallelizing MLFMA: the spatial partitioning (SP) and the *k*-space partitioning. When going up from the finest level to the coarsest level of the tree, the number of spatial boxes gradually decreases from $O(N)$ to $O(1)$, but the number of plane waves or *k*-space samples, by contrary, increases from $O(1)$ to $O(N)$, where N is the number of unknowns. When a large number of parallel processors are used, it is very difficult to achieve good load balance through a simple use of one of the strategies.

Therefore, to achieve high scalability, a commonly used method is to combine the aforementioned strategies in a hybrid manner [4, 5] or in a more efficient hierarchical manner [6–8]. As an efficient alternative, the MLFMA with the fast Fourier transform (FFT) was parallelized to keep up with the modern computational resources with mixed (shared/distributed) memory architectures and achieved very high parallel efficiency using MPI combined with OpenMP [9]. It takes advantage of the high scalability behavior of the fast multipole method (FMM)-FFT for the distributed-memory computations implemented at the coarsest level, while the algorithmic efficiency of the MLFMA benefits the shared-memory computations at finer levels. Internode communications are only required at the coarsest level, where all-to-all communications are carried out to accomplish the transfer between the two basic strategies. It is worth noting that all-to-all communication is one of the most demanding and the least scalable MPI collective operation, and thus the operation needs to be implemented very carefully.

Although the combination of the two basic strategies improves the scalability of the parallel MLFMA, it increases the difficulty in algorithm design and results in complex coding. Recently, a parallel discontinuous Galerkin boundary element method (DG-BEM) has been developed [10], which employs a graph partitioning library METIS to partition the entire computational domain into subdomains with nearly equal number of unknowns. The number of subdomains is kept proportional to the number of processors with the help of METIS, and thus, subdomains as well as unknowns are approximately uniformly distributed among processors.

Submitted On: November 6, 2018
Accepted On: December 24, 2018

This can be referred to as a spatial partitioning parallelization strategy. However, independent octrees created for all subdomains may have different numbers of levels and multipoles because of various diameters of subdomains, possibly resulting in unbalanced loads among processors. Besides, it is complicated to deal with radiation coupling among subdomains due to overlap or intersection of these octrees.

To develop a simpler and more efficient algorithm, we remove higher levels of the tree and move the coarsest level down to a level where the number of boxes is larger than the number of processors, and then uniformly distribute those boxes and the consequent subtrees among processors, facilitating the load balance. It is worth emphasizing that moving down the coarsest level may cause the computational complexity to increase higher than $O(N\log N)$. Meanwhile, given the fact that communication latencies are higher between far-away processes than between nearby processes in modern parallel computers, we use the far-field approximation to drastically reduce both the computational complexity and the communication volume between far-away processes during translation at the coarsest level. In other words, most of the communication volume is kept localized in a neighborhood. Note that we map nearby and far-away message passing interface (MPI) processes to nearby and far-way processors, respectively.

The proposed method bears some similarity to the parallel MLFMA-FFT and DG-BEM, where plenty of subtrees or subdomains are generated and distributed among processes. However, it differs in the following manner: (a) its communication pattern better fits with non-uniform network latencies in high performance computing clusters; and (b) its computational complexity is able to reach as low as the conventional MLFMA when the coarsest level and far-field criterion are properly chosen.

This paper is organized as follows: in Section II, the improved SP strategy and its implementation are described. Next, in Section III, the parallel efficiency is investigated, and an application including a multiscale ship model is proposed, followed by the conclusion in Section IV.

II. PARALLELIZATION

A. Spatial partitioning based on a truncated tree

In a typical MLFMA, a tree with $O(\log N)$ levels is established by recursively grouping or subdividing the N unknowns, as illustrated in Fig. 1. The one-buffer-box criterion is utilized, and the coarsest level L_c is usually set at Level 2 to make the algorithm efficient. Obviously, there are not enough boxes at higher levels to be distributed to a large number of processes, yielding an unbalanced work load among processes at these levels. A straightforward method to solve this issue is to move L_c down to a level at which the boxes become more than the processes. At this new coarsest level, the boxes are now enough to be distributed to processes, and the interactions between increasing far boxes are taken into account by using FMM. As an example, shown in Fig. 1, L_c is set at Level 4 instead of Level 2; in this case, there are ten coarsest boxes distributed to four processes as well as their descendants. It is worth noting that the operation of moving down the coarsest level is equivalent to truncating the higher levels of a MLFMA tree, which generates many subtrees below the coarsest level. Given the load balance, it is easier to distribute these shallower subtrees than distribute a single deeper tree to processes.

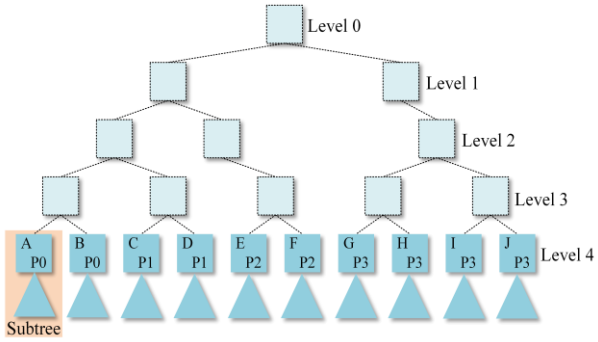

Fig. 1. Illustration of a tree in MLFMA. By moving the coarsest level down to Level 4, the boxes denoted by A–J with their descendants form plenty of subtrees, which are distributed to four processes P0–P3.

B. Reduced communication between far-away processes during translation

Nowadays, the communication between processes has become an important factor in determining the parallel performance of electromagnetic codes, especially in supercomputer environments. Even for a relatively moderate machine size, messages might travel a large number of hops on average [11]. The hop count refers to the number of intermediate devices through which data must pass between source and destination [12]. For modern mixed memory computing clusters, communications among processors belonging to the same computing node are significantly faster than those among processors located in different machines [6]. Therefore, a desirable task is to map the communicating processes using a nearby processors criterion.

Let us refer the coarsest level boxes, marked in dark blue in Fig. 1, as the observation boxes. With one-buffer-box criterion taken into account, if an observation box and its near-neighbor source boxes with their descendants are located in the same process, the communication during the aggregation and disaggregation phases can be completely avoided at the expense of some data replication [9]. However, due to the use of FMM at the

coarsest level, the communication during the translation phase becomes very expensive, especially when one process communicates with its far-away neighbors to deal with the far interaction boxes.

It is noticed that, when a source box is far enough from an observation box, only one k-space direction of the translator contributes mostly to the interaction of the two boxes, whereas the other directions can be negligible. That direction points directly from the source box to the observation box, as illustrated in Fig. 2. In order to use this far-field approximation, the distance R between the two boxes should satisfy [13]:

$$R > 3\gamma\sqrt{D_x^2 + D_y^2 + D_z^2}, \quad (1)$$

where $D_{x,y,z}$ is the side lengths of the box and $\gamma \geq 1$. Consequently, the number of k-space samples for a translator is reduced from $2L^2$ to 1, where L is the number of terms in the addition theorem for FMM and proportional to the box size. Thus, if two far interaction boxes are distributed to different processes, the communication volume is also reduced from $2L^2$ to 1.

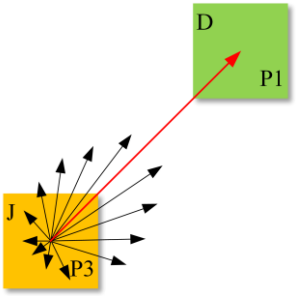

Fig. 2. Translation between two far interaction boxes J (source box) and D (observation box). The red arrow denotes the translator component along the direction from the source box to the observation box, and the black arrows denote other components.

According to the one-buffer-box criterion and Eq. (1), for an observation box at the coarsest level, its source boxes are classified into three types: near-region, resonant-region and far-field boxes, as shown in Fig. 3. The contribution from the near-region boxes is computed using MLFMA at lower levels, in the case of the resonant-region boxes, the contribution is computed using FMM, meanwhile for the last kind of boxes, the contribution is computed using the far-field approximation. When the number of boxes M is approximately $N^{0.5}$ at the coarsest level, the computational cost of the method can be as low as $0.5N\log(N)$ comparing with the conventional MLFMA, if N is very large [13]. However, as the value of γ increases at the coarsest level, more boxes are handled using FMM, resulting in increasing computational complexity and communication volume among processes. To achieve low complexity and high performance, a relatively small γ is preferred for the method. Given the above-mentioned factors, the coarsest level is commonly selected in such a way in which $M \approx N^{0.5}$, and thus the number of processes P is bounded by $O(N^{0.5})$. Assume that each process is attributed $O(N^{0.5})/P$ coarsest boxes and each box has $O(N^{0.5})$ k-space samples. At the coarsest level, the communication volume is $O(N)/P$ during full translation between two processes, whereas it is reduced to $O(N)/P^2$ by utilizing the far-field approximation. It is worth noting that the method will be more expensive than the conventional MLFMA if $M \approx N^{0.5}$, when N is small. In this case, the coarsest level is usually slightly moved down to a level at which $M < N^{0.5}$. In other words, the method might become inefficient if N is small or γ is large.

Fig. 3. Illustration of three types of source boxes for observation box D at the coarsest level (Level 4). Green boxes are near-region boxes, yellow ones are resonant-region boxes, and blue ones are far-field boxes.

C. Implementation detail

Distributing the coarsest boxes or subtrees equally among processes may fail to provide good load balance because the amount of work per box is not constant. This distribution scheme can be improved by considering the estimated amount of work per box, as was done in [9]. For convenience, more sophisticated distributions are not taken into account herein.

In a typical MLFMA with one-buffer-box criterion, the number of translators stored at each level can be reduced by exploiting the symmetries associated with translators [14]. However, as γ increases in Eq. (1), the number of translators required also increases, resulting in a larger memory footprint. Hence, translators are interpolated at the coarsest level by using Lagrange polynomial interpolation with six points and five times the required sampling rates [15], whereas at the lower levels, the translators are stored in memory with the symmetries taken into account.

For the sake of communication during translation, we build two interaction lists in each process at the coarsest level: one for resonate-region boxes and the other one for far-field boxes. The former contains box indices, and each process sends and receives full outgoing plane-wave expansions. Given a large amount of the data, we exchange them in blocks to reduce the number of communication calls. The latter contains box indices and the corresponding k-space directions, exchanging them in one block by using one communication call. Four directions are used for calculating the translator because we use four-point interpolation. To minimize latency, the MPI non-blocking communication is performed to overlap communication

and computation. Note that all communications occur during translation, but aggregation and disaggregation require no communication.

During the solution procedure, we can solve the equation by using a Krylov space solver. Alternatively, we can iteratively solve the equation associated with each subtree firstly and then consider the coupling among subtrees or coarsest boxes through outer iterations. This inner-outer iterative manner has been utilized in domain decomposition methods [10]. In addition, the use of a few plane waves to compute the coupling between two far-field coarsest boxes is similar to using the ray-tracing method to take account of the coupling [16]. It is noted that the proposed method has higher numerical accuracy than that in [16] due to the rigorous computation of the coupling between near-region and resonant-region coarsest boxes.

To accelerate the iterative convergence rate, a basis-function neighbor preconditioner is employed rather than the commonly used block diagonal preconditioner. For a given basis function, its neighbor basis functions within a certain distance are collected to create the preconditioner [17]. Because the basis-function neighbor preconditioner is built independently for each basis function, it can be efficiently implemented in parallel.

III. NUMERICAL EXAMPLES

In order to investigate the strong scalability of the proposed method, the scattering analysis of a conducting sphere is carried out. Then a ship model is simulated to demonstrate the efficiency of the method in computation of bistatic radar cross section (RCS). The models are formulated by the combined field integral equation (CFIE) with a combination factor 0.5 and discretized using the RWG basis functions [18]. The parallel generalized minimal residual (GMRES) method combined with a basis-function neighbor preconditioner is selected as the iterative solver. The computational platform has 16 computing nodes, each of which is configured with four 18-core 2.3 GHz CPUs and 192 GB memory. The nodes are connected by a 100 Gb/s network.

A. Scattering from a sphere model

The scattering analysis of a conducting sphere of diameter 266.6λ is computed to test the parallel efficiency of the algorithm, where λ is the free-space wavelength. The model is discretized into 58327428 unknowns. In this case, a ten-level MLFMA is used with an edge length for the finest box of 0.25λ.

In order to demonstrate the correctness of the proposed method, a comparison of the bistatic RCS with the analytical solution (Mie series) has been carried out. The simulation parameters for the proposed method are γ equal to 3 and L_c set to 4. The residual for iterations is set to 0.001. Figure 4 shows the comparison where a very good agreement is appreciated. However, if γ were to be reduced, the results would not be so accurate since the far-field approximation might be used in some resonate-region boxes. Readers are referred to [13] for an in-depth discussion about the accuracy of the far-field approximation.

The scattering analysis of the conducting sphere has been carried out by increasing the number of processes and calculating its parallel efficiency. The computational time employed in performing one MVP for the proposed spatial partitioning (SP) technique is given in Table 1, and the memory requirement is approximately 893.04 GB. According to the definition of speedup and parallel efficiency [8], the reference number of process should be set to 1. With consideration of the MVP time and memory requirement of the algorithm, it is reasonable to set a moderately larger number of processes as in [6]. In this example, it is set to 32. As seen from Table 1, the proposed strategy is able to achieve high parallel efficiency comparable to the hybrid and hierarchical strategies in [4, 6].

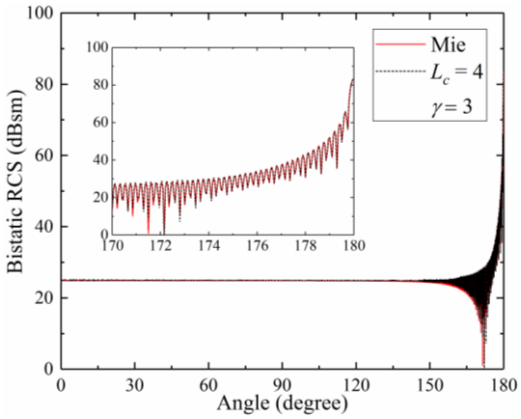

Fig. 4. Bistatic RCS comparison for a conducting sphere of diameter 266.6λ (VV polarization, for vertical transmitting and vertical receiving). A ten-level MLFMA is used, L_c is 4, and γ is chosen as 3.

Table 1: Strong scalability for one matrix-vector product in simulating the sphere when L_c is 4 and γ is 3

CPU Cores	MVP Time (s)	Speedup	Parallel Efficiency (%)
32	756.44	1.00	100.00
288	94.13	8.04	89.29
576	54.56	13.86	77.02
1152	32.89	23.00	63.89

It is noted that the maximum number of processes is limited by the number of boxes at the coarsest level in the proposed strategy. In this example, the maximum number of processes is 1152, which is slightly smaller than the number of boxes 1160. In order to improve the scalability of the proposed strategy, the coarsest level should be moved down to lower levels, where more

coarsest boxes can be obtained. This is equivalent to transfer from coarser-grained parallelism to finer-grained parallelism, facilitating load balance and scalability. However, it is possible that moving down the coarsest level might increase the computational complexity of the algorithm. To ensure high numerical accuracy and efficiency of the method, one has to set suitable parameters L_c and γ, as discussed in Section II. B.

B. Scattering from a multiscale ship model

The second example consists of the scattering analysis of a conducting ship model. The model is 167 m long, 19 m wide and 34.7 m high, as shown in Fig. 5.

The bistatic RCS is computed at 1 GHz to verify the accuracy of the proposed SP strategy. The number of unknowns is 21772044 in this case. Figure 6 illustrates the results for this analysis where a comparison with the parallel MLFMA has been carried out [8]. Both results present a good agreement. The proposed method converges to 0.01 with 51 iterations, and it takes 920.12 s and needs 306.38 GB memory in total when $P = 1152$. The time for computing one MVP is 16.00 s and the parallel efficiency relative to 32 cores is 60.28% (the MVP time is 347.21 s when $P = 32$).

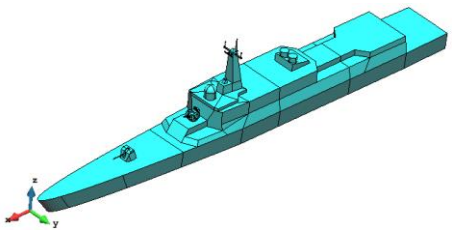

Fig. 5. Conducting ship model with dimensions of 167 m × 19 m × 34.7 m.

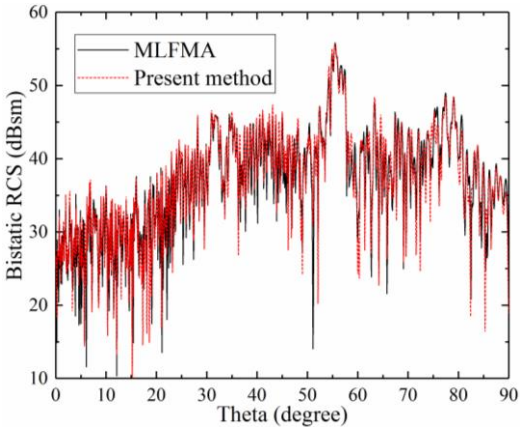

Fig. 6. Bistatic RCS comparison for the ship (VV polarization) at 1 GHz. A plane wave of frequency 1 GHz is incident at $\theta_{\mathrm{inc}} = 55.5°$ and $\phi_{\mathrm{inc}} = 0°$, and the observation directions are set as $0° \le \theta_{\mathrm{scat}} \le 90°$ and $\phi_{\mathrm{scat}} = 0°$. An eleven-level MLFMA is used. L_c is 7, and γ is chosen as 4.

We then consider the scattering analysis of the ship at a higher frequency 2.3 GHz. The electrical length of the ship is 1280.3λ and discretization of its model generates 115444341 unknowns, which is approximately five times the unknowns at 1 GHz. The total solution time is 4121.94 s and the memory requirement is 1495.09 GB when 1152 cores are used. The time for carrying one MVP is 72.81 s, approximately five times the MVP time at 1 GHz. In addition, the memory requirement is also about five times the memory at 1 GHz. This implies that the complexity of the present method is approximate $N\log(N)$. The bistatic RCS curve is plotted in Fig. 7.

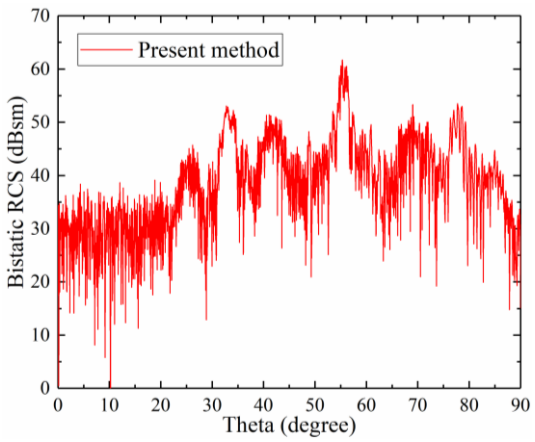

Fig. 7. Bistatic RCS comparison for the ship (VV polarization) at 2.3 GHz. The incident angle and the observation directions are the same as those in Fig. 6. A twelve-level MLFMA is used. L_c is 7, and γ is chosen as 4.

IV. CONCLUSION

The scalability of the spatial partitioning strategy for parallelizing MLFMA is improved from $O(1)$ to $O(N^{0.5})$ processes by properly setting the coarsest level. Because of the important role of communication in determining performance of parallel algorithms, the far-field approximation is employed to reduce the communication volume between far-away processes during translation at the coarsest level, and hence, most of the communication volume occurs between nearby processes that are mapped to nearby processors. The proposed algorithm can be referred to as a network topology aware algorithm. In addition, the proposed strategy can be combined with the *k*-space partitioning strategy to achieve better scalability.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0202102, in part by the NSFC (61301069), in part by the China Postdoctoral Science

Foundation funded project under Grant 2017M613068, and in part by the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) under Grant No. U1501501.

REFERENCES

[1] X. Zhao, Y. Chen, H. Zhang, Y. Zhang, and T. K. Sarkar, "A new decomposition solver for complex electromagnetic problems [EM Programmer's Notebook]," *IEEE Antennas and Propag. Mag.*, vol. 59, no. 3, pp. 131-140, June 2017.

[2] W. Yu, X. Yang, Y. Liu, L.-C. Ma, T. Su, N.-T. Huang, R. Mittra, R. Maaskane, Y. Lu, Q. Che, R. Lu, and Z. Su, "A new direction in computational electromagnetics: solving large problems using the parallel FDTD on the BlueGene/L supercomputer providing Teraflop-level performance," *IEEE Antennas and Propag. Mag.*, vol. 50, no. 2, pp. 26-44, Apr. 2008.

[3] S. Jiang, Y. Zhang, Z. Lin, and X. Zhao, "An optimized parallel FDTD topology for challenging electromagnetic simulations on supercomputers," *International Journal of Antennas and Propagation*, vol. 2015, Article ID 690510, 10 pages, 2015.

[4] 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.

[5] X.-M. Pan, W.-C. Pi, M.-L. Yang, Z. Peng, and X.-Q. Sheng, "Solving problems with over one billion unknowns by the MLFMA," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2571-2574, May 2012.

[6] 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, June 2009.

[7] B. Michiels, J. Fostier, I. Bogaert, and D. D. Zutter, "Weak scalability analysis of the distributed-memory parallel MLFMA," *IEEE Trans. Antennas Propag.*, vol. 61, no. 11, pp. 5567-5574, Nov. 2013.

[8] X. Zhao, S.-W. Ting, and Y. Zhang. "Parallelization of half-space MLFMA using adaptive direction partitioning strategy," *IEEE Antennas and Wireless Propag. Lett.*, vol. 13, pp. 1203-1206, 2014.

[9] J. M. Taboada, M. G. Araujo, F. O. Basteiro, J. L. Rodriguez, and L. Landesa, "MLFMA-FFT parallel algorithm for the solution of extremely large problems in electromagnetics," *Proceedings of the IEEE*, vol. 101, no. 2, pp. 350-363, Feb. 2013.

[10] B. MacKie-Mason, A. Greenwood, and Z. Peng, "Adaptive and parallel surface integral equation solvers for very large-scale electromagnetic modeling and simulation (invited paper)," *Progress In Electromagnetics Research*, vol. 154, pp. 143-162, 2015.

[11] T. Agarwal, A. Sharma, A. Laxmikant, and L. V. Kale, "Topology-aware task mapping for reducing communication contention on large parallel machines," *in Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium*, Rhodes Island, Greece, 25-29 April 2006.

[12] Hop (networking), accessed on June 25, 2017. [Online]. https://en.wikipedia.org/wiki/Hop_(networking).

[13] W. C. Chew, T. J. Cui, and J. M. Song, "A FAFFA-MLFMA algorithm for electromagnetic scattering," *IEEE Trans. Antennas Propag.*, vol. 50, no. 11, pp. 1641-1649, Nov. 2002.

[14] S. Velamparambil, W. C. Chew, and J. Song. "10 million unknowns: is it that big?," *IEEE Antennas and Propag. Mag.*, vol. 45, no. 2, pp. 43-58, Apr. 2003.

[15] J. Song and W. C. Chew, "Interpolation of translation matrix in MLFMA," *Microw. Opt. Techn. Lett.*, vol. 30, no. 2, pp. 109-114, July 2001.

[16] C. Delgado and M. F. Cátedra, "Combination of ray-tracing and the method of moments for electromagnetic radiation analysis using reduced meshes," *Journal of Computational Physics*, vol. 361, pp. 412-423, Jan. 2018.

[17] Y. Zhang, Y. J. Xie, and C. Liang, "A highly effective preconditioner for MoM analysis of large slot arrays," *IEEE Trans. Antennas Propag.*, vol. 52, no. 5, pp. 1379-1381, May 2004.

[18] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, pp. 409-418, May 1982.