Progress In Electromagnetics Research, Vol. 173, 37–52, 2022

Massively Parallel Multilevel Fast Multipole Algorithm for Extremely Large-Scale Electromagnetic Simulations: A Review

Wei-Jia He, Xiao-Wei Huang, Ming-Lin Yang*, and Xin-Qing Sheng

Abstract—Since the first working multilevel fast multipole algorithm (MLFMA) for electromagnetic simulations was proposed by Chew's group in 1995, this algorithm has been recognized as one of the most powerful tools for numerical solutions of extremely large electromagnetic problems with complex geometries. It has been parallelized with different strategies to explore the computing power of supercomputers, increasing the size of solvable problems from millions to tens of billions of unknowns, thereby addressing the crucial demand arising from practical applications in a sense. This paper provides a comprehensive review of state-of-the-art parallel approaches of the MLFMA, especially on a newly proposed ternary parallelization scheme and its acceleration on graphics processing unit (GPU) clusters. We discuss and numerically study the advantages of the ternary parallelization scheme and demonstrate its flexibility and efficiency.

1. INTRODUCTION

The surface integral equation (SIE) method is popular for the computing of electromagnetic field interactions with metallic or homogeneous dielectric materials [1, 2]. The final matrix equation system arising from SIE using the method of moments (MoM) is dense, thereby severely limiting the solvable problem size. To solve the MoM dense matrix equation system more efficiently, fast solvers have been proposed [3–6]. The multilevel fast multipole algorithm (MLFMA), which was first successfully introduced to the field of computational electromagnetics by Chew's group in 1995 [5], is a powerful tool for accelerating the iterative solution of the SIE. Since then, the MLFMA has gained broad interest and has attracted wide attention worldwide, especially for solving electromagnetic problems with large electrical sizes.

Although the computational complexity of the MLFMA is only $O(N\log N)$, and it enables solution of a large problem with millions of unknowns by using a single computer, real-life applications always have higher demand. Even though the hardware technology of modern computers has been highly strengthened in the past few decades, the computing power of a single computer is still far from satisfying practical engineering demand. Therefore, the parallelization of the MLFMA to benefit from the present supercomputer power is crucial. To this aim, many studies have contributed to MLFMA parallelization on shared, distributed, and mixed-memory computers by using different partitioning strategies and parallel programming models.

Efforts for the parallelization of the MLFMA were made soon after the method was proposed and has continued for over two decades [7–36]. The parallelization of the MLFMA on distributed-memory parallel computers is very challenging due to the complicated multilevel structure of the algorithm: the number of boxes increases, while the number of plane waves in each box decreases from the current level to the next lower level. The simple parallelization approach of the MLFMA partitions only boxes

Received 12 January 2022, Accepted 3 March 2022, Scheduled 9 March 2022

* Corresponding author: Ming-Lin Yang (yangminglin@bit.edu.cn).

The authors are with the Center for Electromagnetic Simulation, School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China.

among processes [7]. The idea of a simple parallelization approach is very straightforward, but parallel efficiency can be promised for a small number of processes only, as the number of boxes on the second highest level MLFMA tree is no more than eight. Compared with the simple parallelization approach, the development of a hybrid parallelization strategy provides a one-step forward larger parallelization scale [8–18]. This strategy partitions only boxes at lower levels among processes and only planewaves in a box among processes on higher levels. With the hybrid strategy, problems involving over 10 million unknowns were solved for the first time with 126 processes [8]. Later, with optimization of the hybrid strategy, a series of studies were proposed with increasing problem size solvability [9–18]. Gurel's group proposed a solution of problems with tens of millions of unknowns [12, 16]. At the same time, a sophisticated parallel MLFMA was proposed by Sheng's group to calculate electrically large objects involving 130 million unknowns [15].

Even though other optimizations have been performed on the hybrid parallel strategy of the MLFMA and improved its performance significantly [17], the parallel scale is still severely limited, and the parallel efficiency drops fast. The reason is that in the hybrid strategy, the MLFMA levels are partitioned among processes either only by boxes or only by planewaves in each box (along the θ-direction). For generally 3D objects, when the process number is large, there are some levels that fail to provide efficient workload partitioning just by boxes or by planewaves. In addition, to switch box partitioning and planewave partitioning patterns on the intermediate level, global all-to-all communications are required. The commonly used MPI library uses 32-bit pointers, which corresponds to a maximum 2 GB data limitation for all-to-all communication. Partitioning of messages not only increases latencies but also is difficult and inefficient. Therefore, global all-to-all communication has become another bottleneck, especially for large-scale problems. In [18], a hybrid MPI and OpenMP parallel MLFMA matching with the architectures of popular mixed-memory supercomputers was proposed, where the OpenMP multithread programming model was used to accelerate the pure MPI parallel MLFMA. By using OpenMP, the load balance and scalability of the hybrid parallel MLFMA was improved, and problems involving a billion unknowns were solved efficiently with over a thousand CPU cores (including processes and threads). Although the use of OpenMP in accelerating MPI parallel MLFMA allows the use of more CPU cores, the limitation of the hybrid approach caused by all-to-all communications on the transition level remains unchanged.

The development of the hierarchical strategy is a significant progress in the parallelization of the MLFMA. In the hierarchical parallelization approach of the MLFMA, both boxes and plane waves in a box at each level are partitioned among MPI processes. The hierarchical strategy has been applied successfully to 2D problems [19, 20] as well as general 3D objects with arbitrary shapes [21–24]. Since in the implementation of the hierarchical strategy, boxes and plane waves are bisected, the total number of processes used is strictly limited to 2^n. Using the hierarchical strategy, numerical solutions of electromagnetic problems involving a maximum of 374 million unknowns were obtained [23]. The conventional hierarchical strategy partitions only plane waves among processes in the same group along the θ-direction and may still suffer from load imbalance when a large number of processors are used. An alternative strategy, namely, the blockwise hierarchical partitioning approach, was developed, where partitioning of the plane waves in both θ and φ directions is performed [25]. This improved approach has been proven to have high parallel efficiency and is especially suitable for computing with a very large number of CPU cores at the price of a number of processes strictly limited to 4^n. Using the blockwise hierarchical partitioning approach, an electrically large electromagnetic scattering problem with as many as 3 billion unknowns was solved, with 4096 processes, which was the largest number of unknowns and the highest number of parallel processes ever reported at that time [27]. The limitation of the process number in the hierarchical strategy results in substantial inconvenience in practical implementations. An adaptive direction partition optimization of the hierarchical strategy was proposed for parallel wideband MLFMA [28, 29] but may partly lose the important advantage of accelerating interprocess communication in the hierarchical strategy.

To overcome the shortcomings of state-of-the-art parallelization strategies of the MLFMA, we presented a flexible and efficient ternary partitioning scheme in [30]. This scheme integrates the advantages of available partitioning strategies in an appropriate manner to parallelize MLFMA tree structures on given numbers of processors. By using hierarchical structure partitioning (HSP) levels to joint pure plane wave partitioning (PWP) levels and box partitioning (BP) levels and using a virtual

local transition (LT) level to switch patterns between BP and HSP, the scheme resolves the disadvantage of the hierarchical strategy, as well as the hybrid strategy. Numerical results proved that the scheme achieved a parallel efficiency as high as that of the hierarchical parallelization approach, while the limitation of the number of MPI processes was significantly weakened. Using this ternary strategy, together with the auxiliary-tree-based parallel mesh refinement technique and a hybrid octree storage strategy, electromagnetic scattering by extremely large objects with dimensions exceeding 10 thousand wavelengths and over 10 billion unknowns was solved, representing the largest number of unknowns to be solved and publicly reported via MLFMA to date. The ternary parallelization approach was further developed by incorporating a discontinuous Galerkin (DG) framework [31] and later extended to accelerate the solution of the DG-based self-dual integral equation method for objects with impedance boundary conditions (IBCs) [32].

Another direction of developing massively parallel distributed-memory MLFMA is using the fast Fourier transform (FFT) [33–36]. It preserves the scaling propensity of the FMM while the translation stage is accelerated by using the FFT. Compare with the directly parallelization of MLFMA, this approach simplifies the parallelization significantly. The idea of FMM-FFT is combined with the MLFMA for the hybrid MPI/OpenMP parallel implementation, in which the FMM-FFT is used for the distributed computation, while the shared memory MLFMA can be used for accelerating computation on each process [35]. These term parallelization approaches are suitable for modern mixed-memory supercomputers and are reported to be able to solve problems with 620 million unknowns [35]. Readers can refer to [33–36] for technical details of the terms of the FFT-based parallel MLFMA.

There are other potential and important techniques to simplify the parallelization of FMM and MLFMA, one of which is the Gaussian beam concept. With the aid of Gaussian beam concept, the translation stage in FMM can be significantly improved and simplified by using the [37–39]. By using the Gaussian beam FMM translation operators, the number of levels can be reduced considerably, as well as the computational effort. Since the very coarse levels are not needed any more, the parallelization just on a box level is sufficient. This technique provides an effective way to make significant progress on the parallelization of MLFMA.

All the aforementioned parallel implementations of the MLFMA are based on a general CPU architecture. Although the capability of the CPU-based parallel MLFMA has been well strengthened, due to hardware limitations, it is very challenging to further improve its efficiency to satisfy practical engineering demand. Similar to the development trend of high-performance computing, many-core accelerators, graphics processing units (GPUs), Intel Xeon Phi coprocessors (MICs), etc., which are orders of magnitude faster than general CPUs, have received increasing attention. Among these many-core accelerators, the GPU has shown great potential, as almost 70% of supercomputers in the newest annual edition of the TOP500 (58^{th}) use GPU accelerators. Due to the special architecture of GPUs and the complicated structure of the MLFMA, only a few advancements in accelerating MLFMA with GPUs have been made: GPU implementation of a low-frequency MLFMA [40], multi-GPU implementation of the MLFMA using OpenMP on a single node [41], and using MPI on GPU clusters [42, 43], which is far less than that of the CPU-based parallel MLFMA. The parallelization of the MLFMA on a GPU-accelerated supercomputer platform not only requires highly efficient distributed memory parallel to the MLFMA but also careful treatments in coincidence with the MLFMA tree partitioning strategy at different tree levels. To our knowledge, the problem size solved using the GPU-accelerated MLFMA reported in the society of computational electromagnetics is no larger than millions of unknowns with no more than tens of GPUs. How to make the MLFMA run efficiently on distributed-memory clusters with over tens of thousands of GPUs is a challenging task. Recently, the ternary strategy was accelerated by using GPUs on heterogeneous platforms, which further enlarged the solvable problem size to 24000 wavelengths and 41.8 billion unknowns and the parallel scale beyond tens of thousands of MPI processes, showing the superior performance and great potential of the approach. This is a significant progress in the parallelization of the MLFMA. By using GPU accelerators, problems of the same size can be solved faster than using pure CPU. It can not only used to accelerate the simulation of electromagnetic radiation and scattering, but also can be applied as an accurate and fast solver to reduce the computational burden brought by repeatedly computed forward solutions in the inverse scattering problems [44, 45].

In this paper, we review the ternary strategy and provide a comprehensive discussion on its advantage over the major state-of-the-art parallel strategies of the MLFMA. Our recent progress on

the GPU acceleration of the ternary parallel MLFMA is also given in brief. We demonstrate the flexibility and efficiency provided by the ternary strategy on electrically large scattering problems.

2. TERNARY PARALLELIZATION

In the MLFMA, the matrix-vector multiplication in each iteration is split into two parts, namely, the near-field interaction and far-field interaction. The near-field interaction matrix is calculated the same as the conventional MoM, with the matrix stored as a block sparse matrix. The far-field interaction is addressed by aggregation, translation, and disaggregation stages in a group manner, level by level. The far-field interaction is performed with the aid of the multilevel MLFMA octree. This octree is constructed by placing the object in a cubic box first. Then, each dimension of the box is bisected, resulting in at most eight nonempty subboxes. Such a procedure is performed recursively until the size of the smallest box is no greater than a given value, resulting in the multilevel MLFMA octree. In this octree, each node corresponds to a box on the level, and there are the same number of plane waves in all the boxes at the same level. Since unknowns in the SIE are defined only on the outer surface of the object, the number of nonempty boxes increases approximately fourfold, while the number of plane waves in each box decreases at approximately the same speed from the current level to the next lower level. The MLFMA tree is essentially a weighted octree, with the number of plane waves in the associated box as the weight of a node. The key problem in designing an efficient distributed-memory parallelization approach of the MLFMA is how to partition the weighted MLFMA tree among MPI processes accordingly. Since the computational complexity of each MLFMA tree level is $O(N)$, we need to assign the workload of each level, which is determined by the number of boxes and plane waves in each box, among all MPI processes carefully.

Different partitioning strategies result in different parallelization schemes. Figs. 1–3 illustrate the partitioning strategy of the multilevel tree in the hybrid, hierarchical, and ternary parallelization approaches. Since the simple parallelization approach can be efficient for small problems only, we do not discuss it in detail here.

In the hybrid parallelization approach, on higher levels, each box partitions all its plane waves evenly on all processes along the θ-direction, and we note these levels as PWP levels; on lower levels, boxes are distributed among MPI processes, and we note them as BP levels, as illustrated in Fig. 1. The lowest PWP level can be determined by the condition that each process should keep at least 3θ-directions to satisfy that each MPI process communicates only with its neighboring processes when local Lagrange interpolation is adopted, thereby determining the levels that perform BP and PWP. Since on BP levels, all the plane waves of a box are on the same process, while on PWP levels, each process keeps only a part of the plane waves of a box, on the intermediate level between BP and PWP, a virtual transition level is designed, on which all-to-all MPI communications are performed to switch different

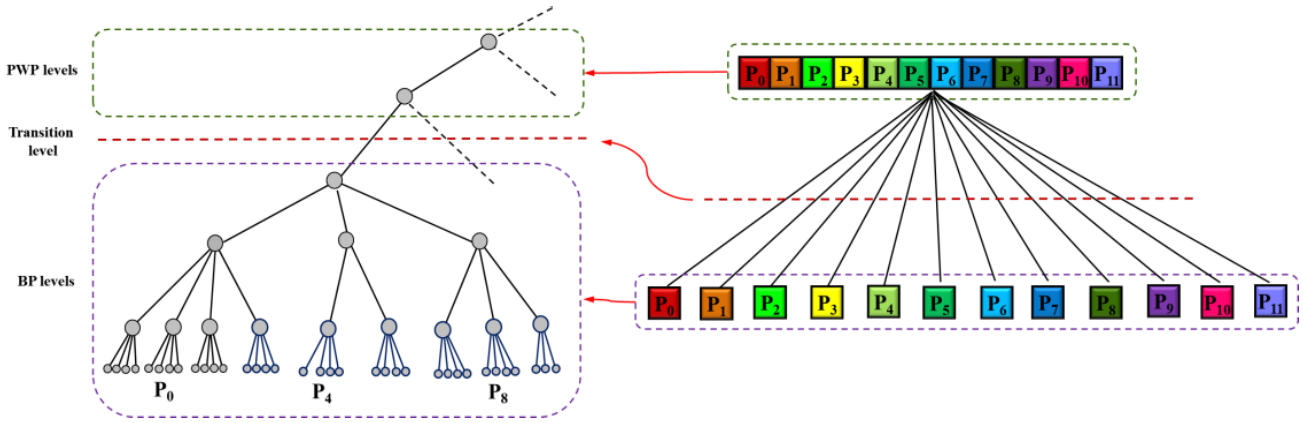

Figure 1. Illustration of tree partitioning patterns and process groups at different levels in the hybrid approach.

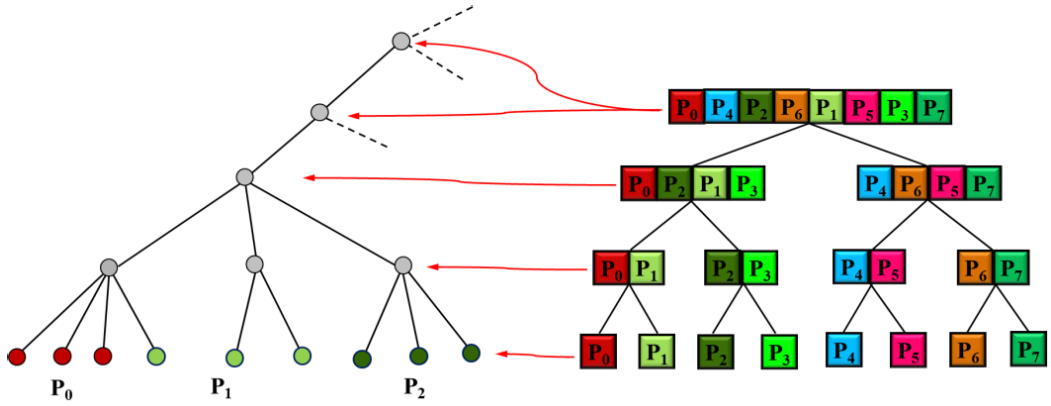

Figure 2. Illustration of tree partitioning patterns and process groups at different levels in the hierarchical approach.

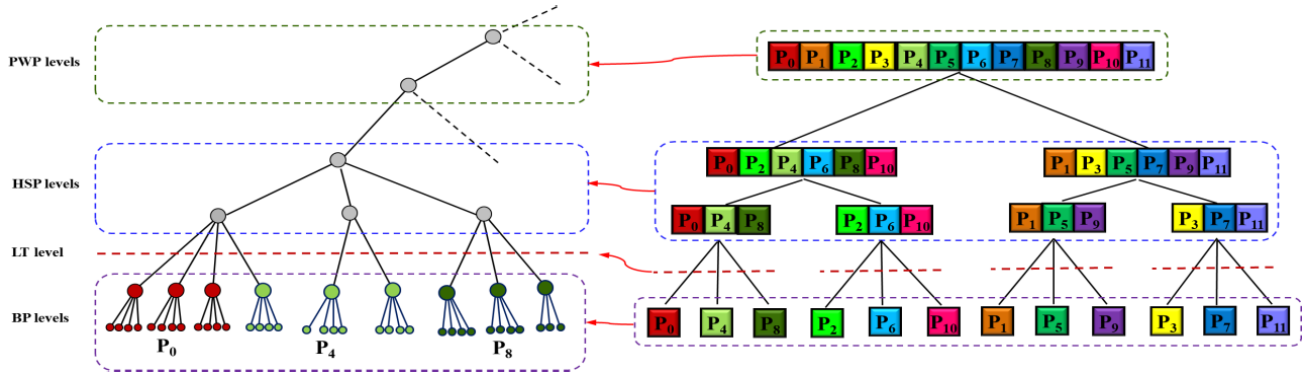

Figure 3. Illustration of tree partitioning patterns and process groups at different levels in the ternary approach.

partition patterns. During the all-to-all data transformation, the communication volume sent/received by a process can be approximately estimated as

$$\frac{N_b \times N_{pw}}{N_p} \tag{1}$$

where N_b denotes the number of boxes, N_{pw} the number of plane waves in a box, and N_p the number of MPI processes used in this calculation. Since on the middle levels of the MLFMA tree, $N_b \times N_{pw}$ is approximately a constant, the communication volume estimated as in Eq. (1) cannot be decreased by choosing a different transition level. If only 32-bit pointers are supported by the MPI standard, which is true for most of the commonly used MPI libraries, the maximum communication volume must be no greater than 2 GB. Although we can use more MPI processes to reduce the value given in Eq. (1), this leads to more severe load unbalance.

The hierarchical parallelization approach introduces a new partitioning pattern in which both the boxes and the plane waves at each level are partitioned among processes at the same time. To this aim, the MPI processes are also divided into groups level by level, forming a process clustering tree. Since the conventional hierarchical approach partitions planewaves among processes in the same group along the θ-direction only, a process binary tree is constructed, as the number of plane waves varies approximately twofold in the θ-direction between two MLFMA tree levels. The hierarchical approach partitions boxes among processes at the leaf level. Then, for the upper levels, two neighboring lower-level process groups are aggregated as a larger parent process group, and plane waves of an upper-level box are partitioned among processes in the same parent group. Such a procedure continues for higher levels until all MPI processes are organized in one group, as shown in Fig. 2. For the working level and all higher levels,

planewaves of the same box are partitioned among all MPI processes, which is the same as the PWP levels in the hybrid approach. The blockwise hierarchical partitioning approach is a modification of the conventional hierarchical approach, in which four processes are organized in a group, and both the θ- and φ-directions of plane waves in a box are bisected, resulting in four blocks of plane waves. The process numbers of the hierarchical and blockwise hierarchical approaches are critically limited to 2^n or 4^n, respectively. This brings great inconvenience in practical implementations. When only thousands of CPU cores can be used in a calculation, the MPI process number can be only 1024, 2048, 4096, or 8192 for the hierarchical partitioning approach, while for the blockwise hierarchical partitioning approach, the choice is only 1024 or 4096.

To overcome the shortcomings of the hybrid and hierarchical approaches, we proposed a ternary parallelization approach of the MLFMA, as illustrated in Fig. 3. It combines the advantages of available partitioning strategies in an appropriate manner. Similar to the hybrid approach, the higher levels of the MLFMA tree in the ternary approach are also termed the PWP levels. The lowest PWP level is also determined to be the same as in the hybrid approach to make each MPI process communicate only with its neighboring processes during interpolation/anterpolation stages. Then, processes in a parent group are bisected into odd- and even-numbered two subgroups according to their local ranks in the group, as well as the associated parent box group. Then, all plane waves of a child box are partitioned evenly along the θ direction among processes in the corresponding process subgroup, as depicted in Fig. 4. These levels are named HSP levels. Following the lowest HSP level are the BP levels. At the BP levels, all child boxes of a box group on the lowest HSP level, as well as all their descendants, are partitioned among the processes in the corresponding leaf group according to their weights. When the mesh is generally uniform, the weight of the box can be estimated by adding up those of all its descendants. The edges can also be partitioned among processes accordingly, as well as the near-field matrix. When the mesh is highly nonuniform, the near-field related calculation should be adjusted by partitioning edges among MPI processes in a different way from the far interaction to achieve a better workload balance [31]. At the lowest level, point-to-point communication is conducted to switch near and far-field partition patterns in each iterative solution step. The flowchart of the ternary PMLFMA is given in Fig. 5.

Note that the number of HSP levels in the ternary parallelization approach can be freely chosen as long as it is no greater than the depth of the process binary tree and the workload is well balanced.

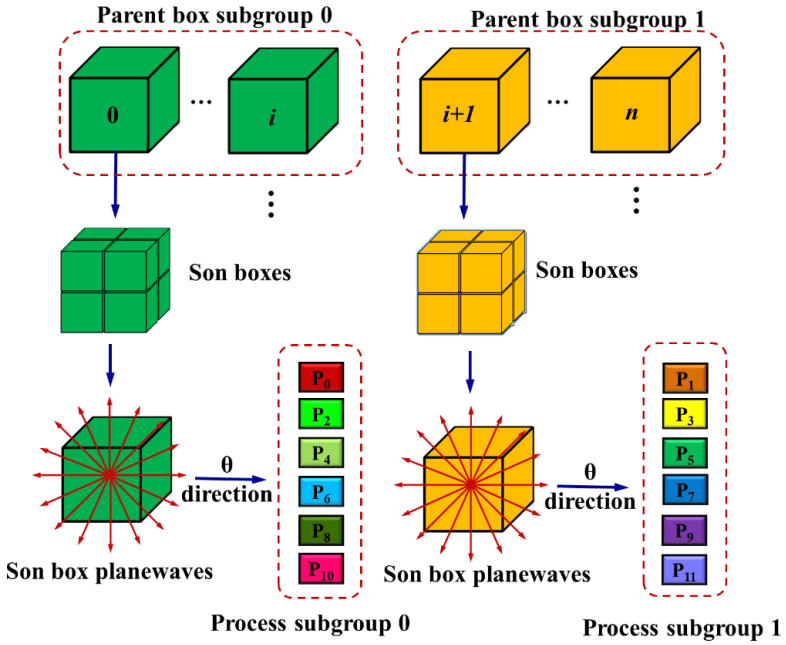

Figure 4. Hierarchical structure partitioning at the highest HSP level over 12 MPI processes.

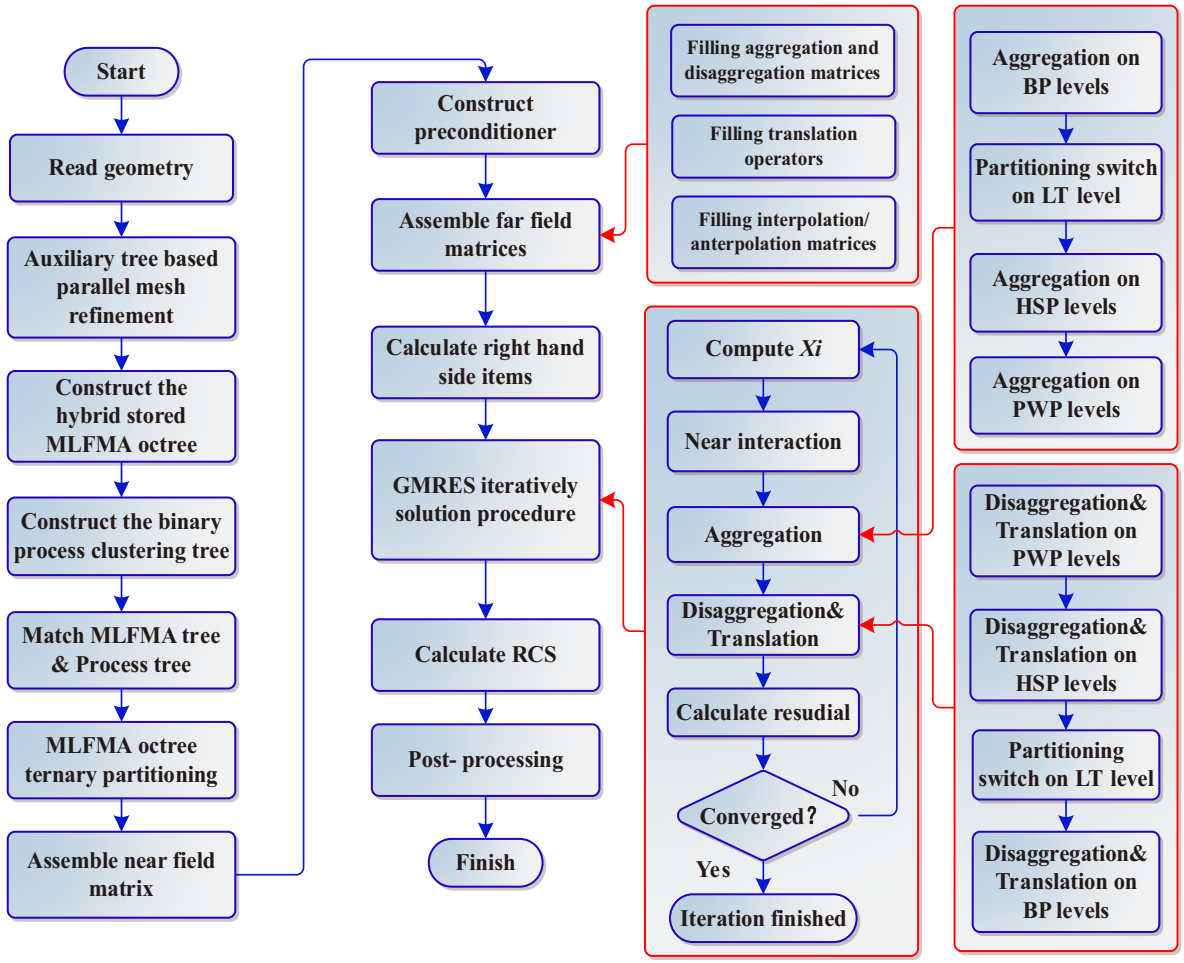

Figure 5. Flowchart of the ternary MLFMA.

The HPS levels are initially designed to address certain middle levels of the MLFMA tree where neither partitioning only the boxes nor plane waves is efficient. However, since for a general 3D object, at middle levels, the number of boxes increases approximately fourfold, while the number of plane waves in each box decreases twofold in the θ-direction from the current level to the next lower level, the use of only two or three HSP levels is sufficient to transmit PWP and BP partitioning patterns and make the workload among processes at all the MLFMA tree levels well balanced. Therefore, a parallel efficiency as high as that of the hierarchical parallelization approach can be realized, while the number of MPI processes is weakly limited.

3. ADVANTAGE OF THE TERNARY STRATEGY

The most important advantage of the ternary strategy is that it can maintain the advantages of the hybrid and hierarchical strategies at the same time. Our previous study proved that the ternary strategy can achieve parallel efficiency that is as high as the hierarchical parallelization approach, while the total number of MPI processes can be more freely chosen [30]. In the following, we provide a short discussion of comparisons of the ternary strategy with the popular hybrid and hierarchical approaches of the MLFMA.

Compared with the hybrid strategy, the introduction of HSP in the ternary approach significantly improves the load balancing, thereby bringing higher parallel efficiency, especially when the process number is large. Similar to the advantage of the hierarchical strategy over the hybrid strategy, the use of HPS decreases communications between processes, as well as the overall data transferred. It also

avoids the global all-to-all communication bottleneck in the hybrid approach. Although similar to the hybrid approach, on the intermediate level between the highest BP and the lowest HSP level, a virtual transition level is also required to switch patterns between box and plane wave partitions. Compared with the global all-to-all communication among all MPI processes in the hybrid approach, the transition is performed only among processes inside the same leaf node of the process binary tree. Therefore, only local all-to-all communication is required, and the communication cost is significantly reduced compared with the global all-to-all case in the hybrid strategy.

A major difference between the hierarchical approach and the ternary approach is that the former is done in a bottom-up manner, while the latter is done in a top-down manner. There are two main reasons for this. First, compared with the bottom-up approach, by using the top-down approach, we can preserve the parent-child relationships as much as possible to reduce interprocess communications in the far interaction. Second, in MLFMA, the important symmetry property of the plane waves can be used to reduce the amount of storage that is required. If the plane waves are partitioned into different MPI processes, the data required using the symmetry may not be local. In that case, extra interprocess communications are introduced, which may cause deficiency. Therefore, we want to keep as many BP levels as possible to reduce storage requirements.

In addition, matching with the ternary partitioning strategy, we are able to design a hybrid storage strategy of the MLFMA octree, which is the key for making the MLFMA applicable for solving problems with tens of billions of unknowns [30]. In the hybrid storage strategy, MLFMA tree levels are divided into distributed levels and full levels. On these distributed levels, only a part of the corresponding level MLFMA is maintained by a process. Those nonlocal tree nodes required to be traversed are compressed and stored by level, forming the proxy tree. The remaining tree levels are fully stored levels, at which each process keeps all tree nodes of these levels. According to the MLFMA algorithm, for a given tree node, all the second near and near interaction boxes of its descendants must be descendants of its near boxes. In the ternary partition strategy, the second BP level to the lowest level can be chosen as distributed levels, and the nonlocal proxy tree can be conveniently constructed. At the highest BP level

Table 1. Memory cost for storing the MLFMA tree of an aircraft model at 32 GHz using different strategies.

MLFMA level	Complete storage (MB)	Hybrid storage (MB)	
		Local tree	Proxy tree
0	0.022	0.022	0
1	0.044	0.044	0
2	0.176	0.176	0
3	0.527	0.527	0
4	2.988	2.988	0
5	10.063	10.063	0
6	38.145	38.145	0
7	150.557	150.557	0
8	627.693	627.693	0
9	2522.834	2522.834	0
10	10238.445	10238.445	0
11	40885.686	42.589	0
12	163111.267	169.908	0
13	648741.204	675.772	0
14	2558446.963	2665.049	0
15	9941657.827	10355.894	187.89
Total	13,366,434.441	27,687.89	

and those higher levels, each process keeps a complete set of all tree nodes on these levels. It is very easy to find all near boxes of a box partitioned to the process on the highest BP level and then fill the proxy tree. Since the number of boxes per level decreases approximately fourfold from a level to the next higher level, the vast majority of tree storage is taken by these several lowest levels. Significant memory reduction can be achieved by distributing some of these levels, and the distribution of three levels can reduce the total tree structure storage to less than 2%. We list in Table 1 that the memory usage for storing the MLFMA octree of an aircraft model has a length of 57.5 m at 32 GHz using 960 MPI processes. Using the hybrid storage strategy, the total memory for storing the MLFMA octree is reduced from about 13.4 TB to only 27.6 GB. Although the hybrid storage strategy is also applicable to the hierarchical approach, it brings significant inconveniences in constructing and traversing the MLFMA tree.

In summary, the ternary strategy hybrids three suitable schemes for partitioning workloads of different MLFMA tree levels, thereby gaining flexibility, high parallel efficiency, and memory reduction at the same time. Compared to the hybrid strategy, the ternary strategy improves the load balancing and the interprocess communications and notably removes the expensive global all-to-all communication, thereby achieving a higher parallel efficiency; compare to the hierarchical approach, it breaks through the critical limitation of total process number, keeps as many BP levels as possible to reduce the memory requirement by using the symmetry of plane waves, and brings great convenience in applying the hybrid storage strategy as well.

4. GPU ACCELERATION OF THE TERNARY MLFMA

GPUs are specially designed for dealing with massive data parallelism. Compared with the CPU, however, their ability to execute logical instructions is very weak. Therefore, GPUs are usually used to accelerate the computationally insensitive parts of a code. As shown in Fig. 5, MLFMA code can be generally divided into two parts: the setup stage and the iterative solution stage. In the setup stage, all MLFMA-related matrices are filled and stored: the near-field matrix, the aggregation and disaggregation matrices, the translation operators, etc. In the iterative solution stage, in each iteration of the GMRES solver, matrix-vector multiplications are split into near-field interactions and far interaction parts. The former involves essentially block sparse matrix-vector multiplication, while the latter involves aggregation, translation, and disaggregation performed level by level.

One of the keys in designing an efficient GPU-accelerated parallel MLFMA is how to partition the computationally insensitive parts of the workload that have already been distributed appropriately to a process. "Appropriate" here means that we not only partition the workload into enough smaller amounts and distribute them among thousands of CPU cores but also fully utilize the hierarchical memory of the GPU and reduce data communications between the CPU host memory and GPU global memory as much as possible. Based on the above principles, we proposed a GPU-accelerated ternary parallel approach of the MLFMA, which is able to solve extremely large electromagnetic scattering problems with tens of billions of unknowns on GPU heterogeneous platforms. This approach uses the CPU and GPU asynchronous computing patterns, in which CPU cores and GPU can handle different parts of the same task or different tasks at the same time. The compute unified device architecture (CUDA) programming model and the OpenMP multithreaded model are used for GPU and CPU cores, respectively. Here, we briefly describe the strategy used in accelerating ternary MLFMA with a GPU.

A sketch map of the GPU acceleration of the setup stage of the ternary MLFMA is given in Fig. 6. Aiming at solving problems with over tens of billions of unknowns, there are enough boxes at the lowest level for us to use the idea of "one thread per near box pair" and "several thread warps per block" to partition the workload of the near-field matrix assembly procedure among cores of a GPU accelerator. In practice, each near box pair associated with the box partitioned on the given MPI process is assigned to a GPU thread, and then a set of threads with the total number equal to several times the GPU warp size are organized as a block. Similar strategies are used for the aggregation and disaggregation matrix filling except that the observer is a box instead. For the translation operators, the idea of "one thread per plane wave" and "one block per operator" is used.

For the iteration solution procedure, since the MLFMA tree is termed the BP, PWP, and HSP levels, different parallelization strategies coinciding with the ternary partition are used. Generally, the

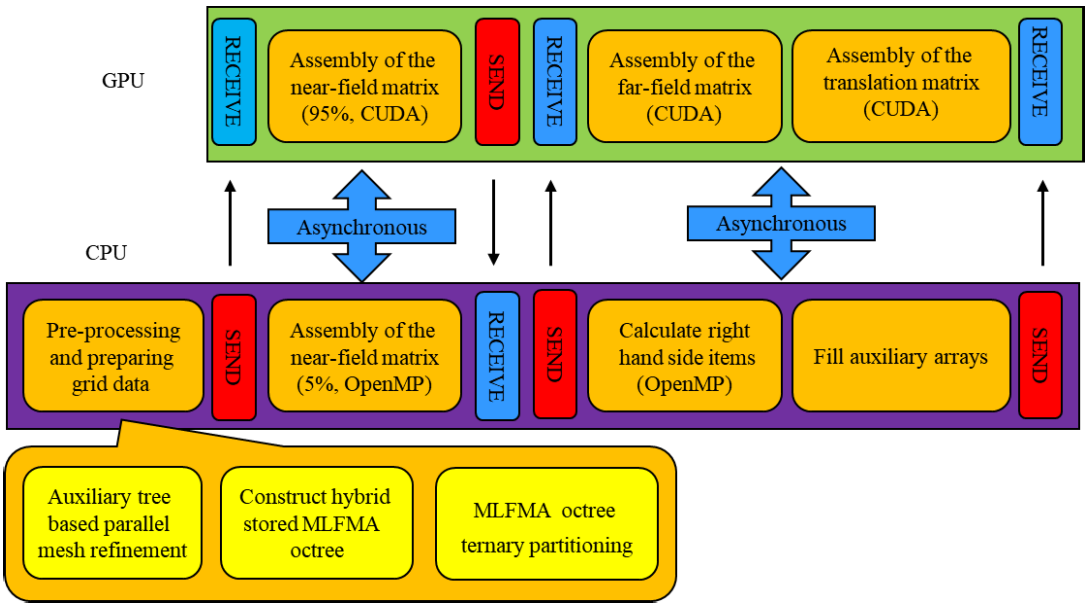

Figure 6. GPU acceleration of the setup stage of the ternary MLFMA.

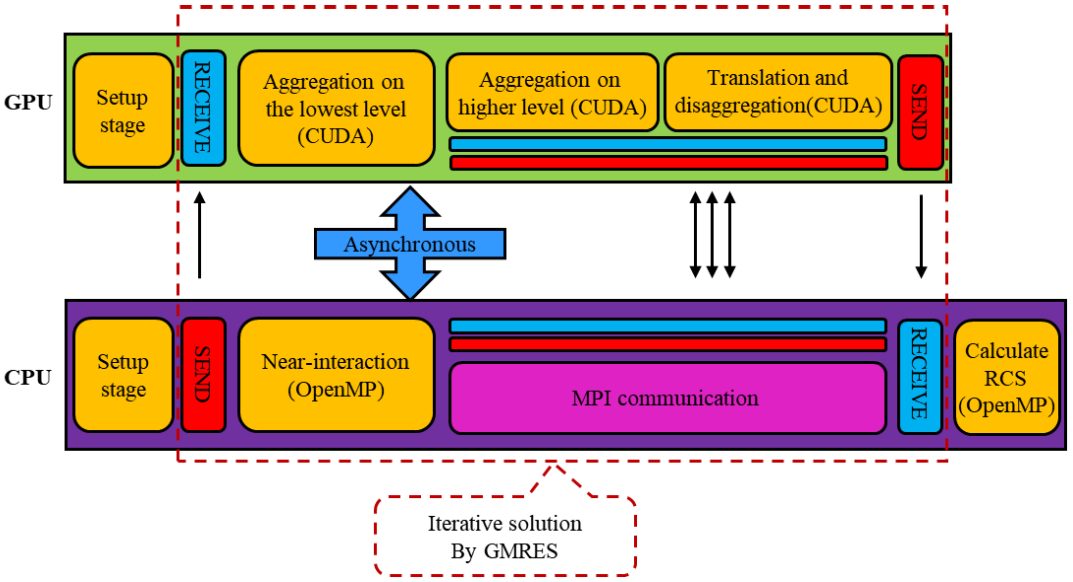

Figure 7. GPU acceleration of the iterative solution stage of the ternary MLFMA.

idea of "one thread per box" and "several warps of threads per block" is used for BP levels, the idea of "one block per box" and "one thread per planewave" is used for HSP levels, and the idea of "several blocks per box" and "one thread per planewave" is used for the PWP levels. These ideas are designed to adapt the varying number of boxes and plane waves between different MLFMA tree levels. The GPU acceleration of the iterative solution stage of the ternary MLFMA is illustrated in Fig. 7. Again, the asynchronous computing pattern between the CPU and GPU is used, with a carefully designed CPU and GPU workload ratio to make the computation time cost for these two parts overlap to reduce the overall running time.

5. NUMERICAL RESULTS AND DISCUSSIONS

In this section, a series of numerical experiments are conducted to evaluate the accuracy, efficiency and performance of the presented algorithm. All the computations are performed on a heterogeneous parallel computer platform, namely, *Xiandao-1*, at the Computer Network Information Centre (CNIC), Chinese Academy of Sciences (CAS). Each node has 128 gigabytes (GB) of host memory, a 32-core x86 CPU and 4 noncommercial SIMT accelerators, which have similar performance to NVidia P100 GPU with PCIe. The nodes are interconnected by a 200 Gbps Infiniband HDR network.

We first show the parallelization efficiency of the ternary parallel approach. Fig. 8 presents the parallelization efficiency for the total duration (including the setup and iterations) of solving the scattering by a sphere of diameter 144 m at 0.3 GHz discretized with 20,104,617 unknowns. The solution is parallelized with 2, 4, 8, 16, 32, 64, 80, 100 and 128 MPI processes. The parallel efficiency when 2 processes are used is defined as 100%. The parallel efficiency remains as high as over 80% on with 128 MPI processes, corresponding to an over 102-fold speedup. The parallel efficiency given in [30] on *Era-II* is approximately 72% when 64 MPI processes are used compared with a single-process case. The parallel efficiency here is much larger than that in [30]. This is because the network used for *Xiandao-1* is 200 Gbps Infiniband HDR, which is significantly faster than that for *Era-II* (100 Gbps Infiniband) when the number of computing nodes used is small. In addition, defining the parallel efficiency of 2 processes as 100% also brings a higher value.

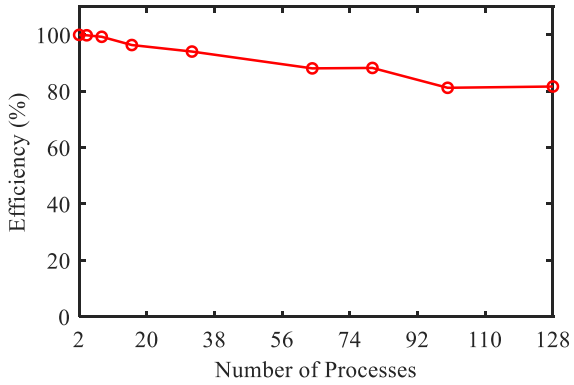

Figure 8. Parallel efficiency for the solution of a sphere of diameter 144 m at 0.3 GHz involving 20,104,617 unknowns.

We then demonstrate accuracy of the method. The scattering by an extremely large sphere with a diameter of 3040 m is considered. There are in a total of 10,068,755,376 edges, 14 MLFMA tree levels. The GMRES solver converges to 0.001 in 158 iterations. The entire computation takes 3 hours and 48 minutes. In total, 42.8 TByte RAM and 36.6 TByte GPU memory were used. The calculated results, together with Mie series, are presented in Fig. 9. According to this figure, the calculated results of the parallel MLFMA well agree with the Mie series.

To evaluate the performance and flexibility of the proposed ternary parallelization approach for MLFMA for handling arbitrary geometries, a complicated aircraft model is calculated. We first set the frequency of the incident planewave as 4 GHz to show the speedup of the GPU-accelerated PMLFMA (GPU-PMLFMA) compared with 8 OpenMP thread-accelerated PMLFMA (CPU-PMLFMA). At this frequency, the aircraft has a maximum dimension of 767λ, with 61,370,448 unknowns. The problem is solved using 16 MPI processes and 8 threads for each process. When GPU accelerators are used, a total of 16 GPU accelerators are used, with each MPI process managing one card. After 42 iterations, a relative residual error of 0.005 is realized. As listed in Table 2, compared with the eight-threaded CPU-PMLFMA, the speedup of GPU-PMLFMA is 6.45, which can be quite important for practical applications.

Then, we increase the incident planewave frequency to 60 GHz. Fig. 10 presents the VV-polarized bistatic radar cross section (RCS) of the aircraft model at 60 GHz calculated using the GPU acceleration version of the PMLFMA. At this frequency, the aircraft has a maximum dimension of 11503λ, with

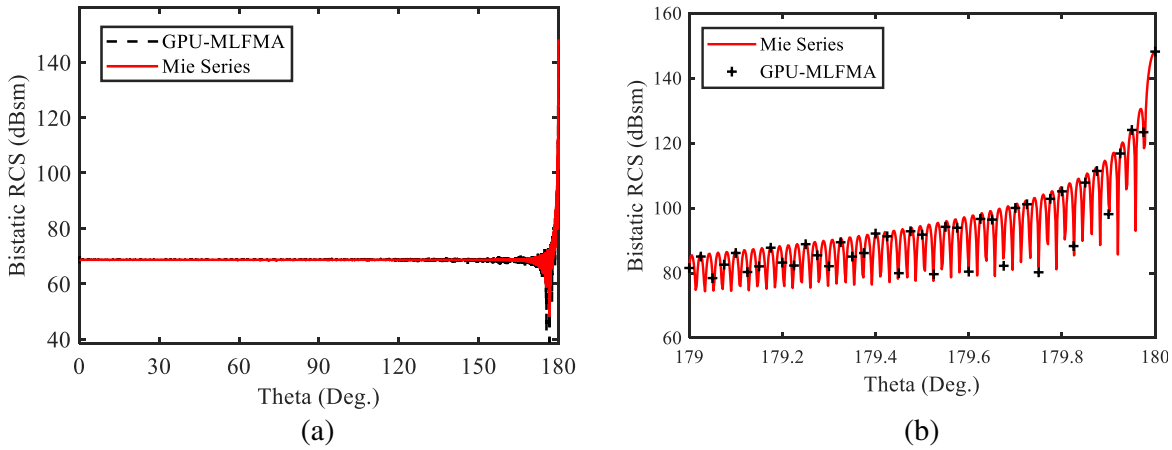

Figure 9. VV-polarized bistatic RCS of a sphere of diameter 3040 m that is discretized with 10,068,755,376 unknowns, where 0 and 180 correspond to the back-scattering and forward-scattering directions, respectively. (a) From 0° to (b) 180°, a detailed view of the RCS along the forward scattering direction.

Table 2. Calculation time of an aircraft model at 4 GHz using different PMLFMA approaches.

Algorithm	CPU-PMLFMA	GPU-PMLFMA	Speedup
V_s and V_f filling	7.16 s	0.70 s	10.23
T filling	89.11 s	13.81 s	6.45
Z_{near} filling	1911.50 s	197.61 s	9.67
Iteration	283.19 s	143.15 s	1.98
Total	2290.96 s	355.27 s	6.45

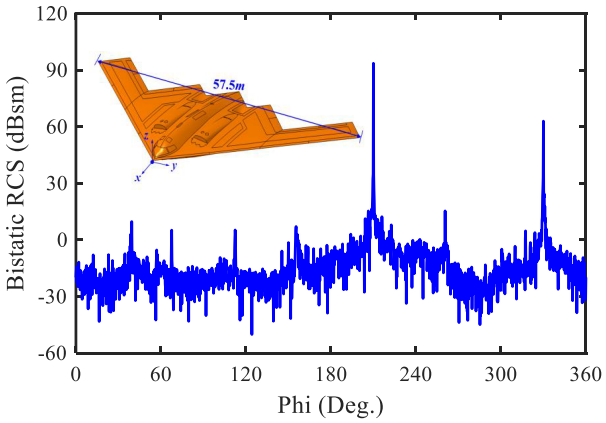

Figure 10. VV-polarized bistatic RCS of the aircraft model at 60 GHz. The target is illuminated by a plane wave that is propagating in the xy-plane at 30 degrees from the x-axis.

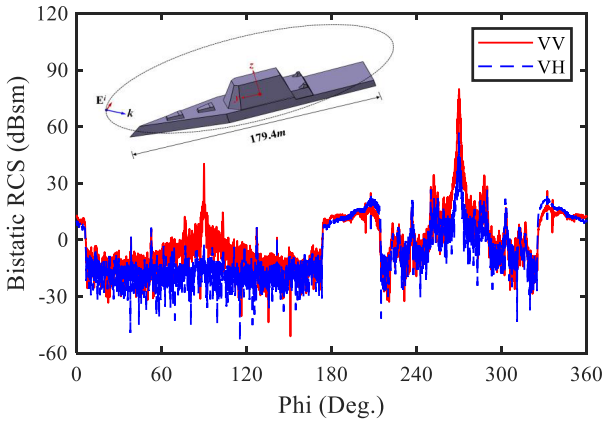

Figure 11. VV and VH polarized bistatic RCSs of the ship model at 20 GHz.

10,166,501,376 unknowns. The problem is solved using 2560 MPI processes and 2560 GPU accelerators, with each MPI process managing one card. After 116 iterations, a relative residual error of 0.005 is realized. The duration of the computation is 3 hours and 55 minutes. Detailed information is provided in Table 3. In total, 31.9 TByte RAM on the CPU and 36.7 TByte GPU host memory were used.

The last example is a ship model at 20 GHz, which is illustrated in Fig. 11. The maximum dimension of the ship is 11970λ at this frequency. Discretization of the problem leads to a total of 10,459,971,504 unknowns. As illustrated in Fig. 11, the nose of the ship is directed towards the z-axis and it is illuminated by a plane wave that is propagating in the yz-plane at 70° from the $+z$-axis. The electric field is polarized in the θ-direction. The VV- and VH-polar bistatic RCSs are plotted as functions of the bistatic angle in the observation plane that is defined by $\theta = 70°$ and $\varphi = 0 \sim 360°$ in Fig. 11. After 102 iterations, a relative residual error of 0.005 was realized. The duration of the computation is 3 hours and 45 minutes. Additional simulation details are also presented in Table 3.

Table 3. Simulation details.

Target	Aircraft model	Ship model
MLFMA levels	16	16
HSP levels	$5 \sim 10$	$5 \sim 10$
Residual error	0.005	0.005
Iteration number	116	102
V_s and V_f filling time	0.22 s	0.11 s
T filling time	260.32 s	294.99 s
Z_{near} filling time	158.03 s	225.89 s
Iteration time	12947 s	12074 s
Total time	14105 s	13501 s
Peak memory on the CPU	31.9 TB	29.2 TB
Peak memory on all GPUs	36.7 TB	36.6 TB

6. CONCLUSIONS

In this paper, we have reviewed the ternary parallelization approach of the MLFMA and its GPU acceleration for extremely large-scale electromagnetic simulations. As discussed in the paper, the ternary parallelization approach hybrids available partitioning strategies in an appropriate manner. Therefore, it maintains the well-scaling property and can achieve as high parallel efficiency as the hierarchical strategy, while the total process can be more freely chosen, which is important in practical application. Efficient parallelization of the MLFMA using the ternary strategy makes it applicable for solving large and realistic problems involving tens of billions of unknowns.

ACKNOWLEDGMENT

This work is supported by the NSFC under Grant No. 61971034 and the National Key Research and Development Program of China under Grant 2017YFB0202500.

REFERENCES

1. Mautz, J. R. and R. F. Harrington, "H-field, E-field, and combined field solutions for conducting bodies of revolution," *Aeu.*, Vol. 32, No. 4, 157–164, Apr. 1978.
2. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, No. 3, 409–418, May 1982.
3. Sarkar, T., E. Arvas, and S. Rao, "Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies," *IEEE Trans. Antennas Propag.*, Vol. 34, No. 5, 635–640, May 1986.

4. Coifman, R., V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, Vol. 35, No. 3, 7–12, Jun. 1993.
5. Song, J. M. and W. C. Chew, "Multilevel fast multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microw. Opt. Tech. Lett.*, Vol. 10, 14–19, Sep. 1995.
6. Song, J. M., 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, 1488–1493, Oct. 1997.
7. Wu, F., 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, 110–118, Aug. 2005.
8. Velamparainbil, S. V., J. E. Schutt-Aine, J. G. Nickel, J. M. Song, and W. C. Chew, "Solving large scale electromagnetic problems using a linux cluster and parallel MLFMA," *IEEE International Symposium on Antennas and Propagation Digest*, Vol. 1, 636–639, Jul. 1999.
9. Donepudi, K. C., J. M. Jin, S. Velamparambil, J. M. Song, and W. C. Chew, "A higher order parallelized multilevel fast multipole algorithm for 3-D scattering," *IEEE Trans. Antennas Propag.*, Vol. 49, No. 7, 1069–1078, Jul. 2001.
10. Velamparambil, S., W. C. Chew, and J. M. Song, "10 million unknowns: Is it that big?," *IEEE Antennas Propag. Mag.*, Vol. 45, No. 2, 43–58, Apr. 2003.
11. Velamparambil, S. and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, Vol. 53, No. 8, 2719–2727, Aug. 2005.
12. Gurel, L. and O. Ergul, "Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns," *Electron. Lett.*, Vol. 43, No. 9, 499–500, Apr. 2007.
13. Waltz, C., K. Sertel, M. A. Carr, B. C. Usner, and J. L. Volakis, "Massively parallel fast multipole method solutions of large electromagnetic scattering problems," *IEEE Trans. Antennas Propag.*, Vol. 55, No. 6, 1810–1816, Jun. 2007.
14. Hu, J., Z. P. Nie, L. Lei, J. Hu, X. D. Gong, and H. P. Zhao, "Fast 3D EM scattering and radiation solvers based on MLFMA," *J. Syst. Eng. Electron.*, Vol. 19, No. 2, 252–258, Apr. 2008.
15. Pan, X. M. and X. Q. Sheng, "A sophisticated parallel MLFMA for scattering by extremely large targets," *IEEE Antennas Propag. Mag.*, Vol. 50, No. 3, 129–138, Jun. 2008.
16. Ergul, O. and L. Gurel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, Vol. 56, No. 8, 2335–2345, Aug. 2008.
17. Fostier, J. and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, Vol. 56, No. 8, 2346–2355, Aug. 2008.
18. Pan, X. M., 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, 2571–2574, May 2012.
19. Fostier, J. and F. Olyslager, "Provably scalable parallel multilevel fast multipole algorithm," *Electron. Lett.*, Vol. 44, No. 19, 1111–1113, Sep. 2008.
20. Fostier, J. and F. Olyslager, "Full-wave electromagnetic scattering at extremely large 2-D objects," *Electron. Lett.*, Vol. 45, No. 5, 245–246, Feb. 2009.
21. Ergul, O. and L. Gurel, "Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics," *Electron. Lett.*, Vol. 44, No. 6, 3–4, 2008.
22. Ergul, O. and L. Gurel, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 6, 1740–1750, Jun. 2009.
23. Ergul, O. and L. Gurel, "Rigorous solutions of electromagnetics problems involving hundreds of millions of unknowns," *IEEE Antennas Propag. Mag.*, Vol. 53, No. 1, 18–27, Feb. 2011.
24. Ergul, O. and L. Gurel, "Hierarchical parallelization of the multilevel fast multipole algorithm (MLFMA)," *Proc. IEEE*, Vol. 101, No. 2, 332–341, 2013.

25. Michiels, B., 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, 5567–5574, Nov. 2013.
26. Michiels, B., I. Bogaert, J. Fostier, and D. De Zutter, "A well-scaling parallel algorithm for the computation of the translation operator in the MLFMA," *IEEE Trans. Antennas Propag.*, Vol. 62, No. 5, 2679–2687, 2014.
27. Michiels, B., J. Fostier, I. Bogaert, and D. D. Zutter, "Full-wave simulations of electromagnetic scattering problems with billions of unknowns," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 6, 796–798, Feb. 2015.
28. Melapudi, V., 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, 2565–2577, 2011.
29. Hughey, S., H. M. Aktulga, M. Vikram, M. Lu, B. Shanker, and E. Michielssen, "Parallel wideband MLFMA for analysis of electrically large, non-uniform, multiscale structures," *IEEE Trans. Antennas Propag.*, Vol. 67, No. 2, 1094–1107, 2018.
30. Yang, M. L., B. Y. Wu, H. W. Gao, and X. Q. Sheng, "A ternary parallelization approach of MLFMA for solving electromagnetic scattering problems with over 10 billion unknowns," *IEEE Trans. Antennas Propag.*, Vol. 67, No. 11, 6965–6978, 2019.
31. Liu, R. Q., X. W. Huang, Y. L. Du, M. L. Yang, and X. Q. Sheng, "Massively parallel discontinuous galerkin surface integral equation method for solving large-scale electromagnetic scattering problems," *IEEE Trans. Antennas Propag.*, Vol. 69, No. 9, 6122–6127, 2021.
32. Huang, X. W., M. L. Yang, and X. Q. Sheng, "A simplified discontinuous Galerkin self-dual integral equation formulation for electromagnetic scattering from extremely large IBC objects," *IEEE Trans. Antennas Propag.*, 2021, doi: 10.1109/TAP.2021.3137485.
33. Taboada, J. M., L. Landesa, F. Obelleiro, J. L. Rodriguez, J. M. Bertolo, M. G. Araujo, J. C. Mouriño, and A. Gomez, "High scalability FMM-FFT electromagnetic solver for supercomputer systems," *IEEE Antennas Propag. Mag.*, Vol. 51, No. 6, 20–28, Dec. 2009.
34. Araújo, M. G., J. Taboada, F. Obelleiro, J. M. Bértolo, L. Landesa, J. Rivero, and J. L. Rodríguez, "Supercomputer aware approach for the solution of challenging electromagnetic problems," *Progress In Electromagnetics Research*, Vol. 101, 241–256, 2010.
35. Taboada, J., M. G. Araújo, J. M. Bértolo, L. Landesa, F. Obelleiro, and J. L. Rodríguez, "MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetic (Invited Paper)," *Progress In Electromagnetics Research*, Vol. 105, 15–30, 2010.
36. Taboada, J. M., M. G. Araújo, F. Obelleiro, J. L. Rodríguez, and L. Landesa, "MLFMA-FFT parallel algorithm for the solution of extremely large problems in electromagnetics," *Proc. IEEE*, Vol. 101, No. 2, 350–363, Feb. 2013.
37. Hansen, T. B., "Translation operator based on Gaussian beams for the fast multipole method in three dimensions," *Wave Motion*, Vol. 50, No. 5, 940–954, Jul. 2013.
38. Hansen, T. B. and O. Borries, "Gaussian translation operator in a multilevel scheme," *Radio Sci.*, Vol. 50, No. 8, 754–763, Aug. 2015.
39. Eibert, T. F. and T. B. Hansen, "Propagating plane-wave fast multipole translation operators revisited — Standard, windowed, Gaussian beam," *IEEE Trans. Antennas Propag.*, Vol. 69, No. 9, Sep. 2021.
40. Cwikla, M., J. Aronsson, and V. Okhmatovski, "Low-frequency MLFMA on graphics processors," *IEEE Antennas Wireless Propag. Lett.*, Vol. 9, 8–11, 2010.
41. Guan, J., Y. Su, and J. M. Jin, "An openMP-CUDA implementation of multilevel fast multipole algorithm for electromagnetic simulation on multi-GPU computing systems," *IEEE Trans. Antennas Propag.*, Vol. 61, No. 7, 3607–3616, 2013.
42. Tran, N. and O. Kilic, "Parallel implementations of multilevel fast multipole algorithm on graphical processing unit cluster for large-scale electromagnetics objects," *Appl. Comput. Electromag. Soc. J.*, Vol. 1, No. 4, 145–148, 2016.

43. Phan, T., N. Tran, and O. Kilic, "Multi-level fast multipole algorithm for 3-D homogeneous dielectric objects using MPI-CUDA on GPU cluster," *Appl. Comput. Electromag. Soc. J.*, Vol. 33, No. 3, 335–338, 2018.
44. Hesford, A. J. and W. C. Chew, "Fast inverse scattering solutions using the distorted Born iterative method and the multilevel fast multipole algorithm," *Journal of the Acoustical Society of America*, Vol. 128, No. 2, 679–690, 2010.
45. Roohani Ghehsareh, H., S. Kamal Etesami, and M. Hajisadeghi Esfahani, "Numerical investigation of electromagnetic scattering problems based on the compactly supported radial basis functions," *Zeitschrift für Naturforschung A*, Vol. 71, No. 8, 677–690, 2016.