The Journal of Supercomputing (2024) 80:8702–8718
https://doi.org/10.1007/s11227-023-05759-2

Massive parallelization of multilevel fast multipole algorithm for 3-D electromagnetic scattering problems on SW26010 many-core cluster

Xin-Duo Liu[1] · Wei-Jia He[1] · Ming-Lin Yang[1] · Xin-Qing Sheng[1]

Accepted: 21 October 2023 / Published online: 28 November 2023

Abstract
This paper presents a massively parallel approach of the multilevel fast multipole algorithm (PMLFMA) on homegrown many-core SW26010 cluster of China, noted as (SW-PMLFMA), for 3-D electromagnetic scattering problems. In this approach, the multilevel fast multipole algorithm (MLFMA) octree is first partitioned among management processing elements (MPEs) of SW26010 processors following the ternary partitioning scheme using the message passing interface (MPI). Then, the computationally intensive parts of the PMLFMA on each MPI process, matrix filling, aggregation and disaggregation are accelerated by using all the 64 computing processing elements (CPEs) in the same core group of the MPE via the Athread parallel programming model. Different parallelization strategies are designed for many-core accelerators to ensure a high computational throughput. In coincidence with the special characteristic of local Lagrange interpolation, the compressed sparse row (CSR) and the compressed sparse column (CSC) sparse matrix storage format is used for storing interpolation and anterpolation matrices, respectively, together with a specially designed cache mechanism of hybrid dynamic and static buffers using the scratchpad memory (SPM) to improve data access efficiency. Numerical results are included to demonstrate the efficiency and versatility of the proposed method. The proposed parallel scheme is shown to have excellent speedup.

Keywords Multilevel fast multipole algorithm · Distributed memory parallelization · Many-core acceleration · Electromagnetic scattering · SW26010 processor

✉ Wei-Jia He
heweijia@bit.edu.cn

[1] Center for Electromagnetic Simulation, Beijing Institute of Technology, Beijing 100081, China

1 Introduction

The multilevel fast multipole algorithm (MLFMA) is a powerful and essential tool in computational electromagnetics (CEM), particularly for large-scale objects in stealth and anti-stealth design, as documented in various studies [1–3]. To extend the solvable problem scale of MLFMA, researchers have proposed several approaches for the distributed memory parallelization of MLFMA over the last few decades. These methods involve the utilization of message passing interface (MPI), hybrid MPI-OpenMP parallel implementations, the fast Fourier transform (FFT)-based method, etc. [4–19]. With these efficient parallel implementations of MLFMA, problems with over ten billion unknowns can now be addressed [17].

To improve the computational capability of pure CPU systems, which is limited by the power and heat dissipation, heterogeneous multicore systems with many-core accelerator become popularized, the CPU and graphics processing unit (GPU), the CPU and Intel Xeon Phi Coprocessors (MIC), the homegrown many-core processor Sunway SW26010, etc. As the first supercomputer system with peak performance greater than 100P in the world, the computing performance and power efficiency of Sunway Taihulight are outstanding [20, 21]. Compared with MIC and GPU, the SW26010 processor of Sunway Taihulight also has several unique hardware features, such as the utilization of scratchpad memory (SPM) for each computing processing element (CPE), direct memory access (DMA) to transfer data between the main memory and the SPM and the design of register communication channels. Due to the challenges in many-core parallel programming on SW26010 processor brought by its special architecture, comparison with progresses on GPU accelerated parallel implementation of MLFMA has been made [22–27], and fewer successful many-core implementation of the MLFMA in CEM community has been proposed [28].

In this article, we extend our prior work in [28] to address electrically large-scale scattering problems through the development of a massively parallel, distributed memory, many-core accelerated parallel MLFMA algorithm on SW26010 processors, denoted as SW-PMLFMA. The proposed SW-PMLFMA algorithm partitions the MLFMA octree tree among processes using a ternary partitioning scheme [17]. Each MPI process's computationally intensive components are then accelerated using the CPEs in the same core groups (CG). We employ a hierarchical parallelization strategy that simultaneously partitions both boxes and planewaves to ensure even workload distribution among all CPEs on higher MLFMA levels. We use the compressed sparse row (CSR) and compressed sparse column (CSC) sparse matrix storage formats for interpolation and anterpolation matrices, which are tailored to local Lagrange interpolation. To enhance data access efficiency in interpolation and anterpolation operations, we designed a hybrid dynamic and static buffer cache scheme. Further optimizations, such as the double buffering scheme and the structure of array (SoA) interface, are employed to improve the algorithm's performance. We demonstrate the efficacy of the proposed SW-PMLFMA algorithm through various simulation results, which shows that it achieves high efficiency without sacrificing accuracy.

The structure of this article is as follows: Sect. 2 provides an overview of the MLFMA and its ternary parallelization approach, while Sect. 3 describes the architecture of the SW26010 many-core processor and the Athread programming model. Section 4 details the implementation of the SW-PMLFMA algorithm. In Sect. 5, we present the results of our numerical simulations, and Sect. 6 concludes the paper.

2 An outline of MLFMA for 3-D electromagnetic scattering and its ternary parallelization approach

The surface integral equation method and its multilevel fast multipole algorithm (MLFMA) acceleration are briefly reviewed in this section. For a general three-dimensional (3-D) object irradiated by an incident field$\left(\mathbf{E}_i,\ \mathbf{H}_i\right)$, the combined field integral equation (CFIE) can be obtained as [3]:

$$\begin{aligned} &\alpha\hat{\mathbf{n}}\times\eta_0\mathcal{L}(\mathbf{J})\times\hat{\mathbf{n}}+(1-\alpha)\hat{\mathbf{n}}\times\eta_0\left[\mathcal{K}(\mathbf{J})+0.5\hat{\mathbf{n}}\times\mathcal{I}\right]\\ &\quad=-\alpha\hat{\mathbf{n}}\times\mathbf{E}^{inc}\times\hat{\mathbf{n}}-(1-\alpha)\eta_0\hat{\mathbf{n}}\times\mathbf{H}^{inc}\end{aligned}\tag{1}$$

where α denotes the combined coefficient, $\hat{\mathbf{n}}$ denotes the unit normal vector, η_0 is the impedance in free space. $\mathcal{L}$ and $\mathcal{K}$ are integral-differential operators defined by:

$$\mathcal{L}(\mathbf{J})\ :=-jk_0\int_S'\left(\mathcal{I}+\frac{1}{k_0^2}\nabla\nabla\cdot\right)\mathbf{J}(r)'G(\mathbf{r},\mathbf{r}')dS'\tag{2}$$

$$\mathcal{K}(\mathbf{J})\ :=P.V.\int_S'\nabla G(\mathbf{r},\mathbf{r}')\times\mathbf{J}(\mathbf{r}')dS'\tag{3}$$

where $G(\mathbf{r},\mathbf{r}')=(e^{-jk_0R}/4\pi R)$ is the Green's function in free space with $R=|r-r'|$, $k_0=2\pi/\lambda_0$. $\mathcal{I}$ denotes the identity operator, $\mathbf{J}$ represents the equivalent electric currents, and $P.V.$ stands for the Cauchy principal value integration.

Following the procedure of method of moments (MoM), by using the Rao-Wilton-Glisson (RWG) functions as the basis functions, and applying the Galerkin's method, a system of linear equations can be obtained:

$$\left[Z_{mn}\right]\left\{J_n\right\}=\left\{f_m\right\}\tag{4}$$

with

$$\begin{aligned} Z_{mn}=\alpha&\int\limits_S\mathbf{g}_m\cdot\left[\hat{\mathbf{n}}\times\eta_0\mathcal{L}(\mathbf{g}_n)\times\hat{\mathbf{n}}\right]\mathrm{d}S\\ &+(1-\alpha)\int\limits_S\mathbf{g}_m\cdot\hat{\mathbf{n}}\times\eta_0\left[\mathcal{K}(\mathbf{J})+0.5\hat{\mathbf{n}}\times\mathcal{I}\right]\mathrm{d}S\end{aligned}\tag{5}$$

$$f_m = \int_S \mathbf{g}_m \cdot \left[-\alpha \hat{\mathbf{n}} \times \mathbf{E}^{inc} \times \hat{\mathbf{n}} - (1-\alpha)\eta_0 \hat{\mathbf{n}} \times \mathbf{H}^{inc} \right] \mathrm{d}S \tag{6}$$

where **g** represents the RWG basis function. The full matrix equation system (4) can be solved iteratively by using Krylov subspace methods such as the generalized minimal residual method (GMRES), with MLFMA to accelerate the matrix–vector multiplication with a complexity of $O(N \log N)$.

MLFMA can be employed to accelerate matrix–vector multiplications. The interaction can be separated into near-field and far-field interactions. The near field is calculated using conventional MoM, while the far field is computed in a group-wise manner by traversing the MLFMA octree level by level. The octree is constructed by recursively bisecting the dimensions of a cubic box until the size of the box on the lowest level is below a given threshold. The calculation of far-field interaction involves three operators: aggregation, translation and disaggregation. In contrast to FMM (fast multipole method) for Laplace equations, where the number of plane waves in a box remains constant, MLFMA for electromagnetic scattering problems solving Helmholtz equation has a twofold increase in the number of planewaves for a parent box at approximately the same rate in the θ- and φ-directions as its child boxes. In the aggregation stage, plane waves from child boxes at the lower level are interpolated and centrally shifted to obtain higher-level plane waves of their parent boxes. In the disaggregation stage, incoming plane waves at the centers of boxes are anterpolated and shifted to the centers of their child boxes at the lower level. During disaggregation, translation operations of plane waves at the same level are required. The translation operators used here are diagonal in nature [2].

The key in designing a scalable parallelization approach of MLFMA is how to partitioning the weighted MLFMA tree among processes properly. In distributed memory parallelization of MLFMA, workload of each MLFMA level can be partitioned among MPI processes by boxes, by planewave directions, or both two simultaneously. The use of different partitioning strategies results in different parallelization approaches, the simple, the hybrid [4], the hierarchical [8] and the ternary [17]. The advantages of available partitioning strategies are integrated by the ternary parallelization approach, addressing the disadvantages presented by other methods. A parallel efficiency comparable to the hierarchical parallelization approach is achieved, and the restriction on the number of MPI processes utilized is substantially reduced.

In the ternary parallelization approach, several higher levels of the MLFMA tree are termed as plane wave partitioning (PWP) levels, on which each box partitions all its plane waves evenly on MPI processes along only the θ-direction. Lower levels are termed as box partitioning (BP) levels, and boxes of the same level are partitioned among MPI processes. The rest levels are hierarchical structure partitioning (HSP) levels, which switch the partitioning patterns of PWP and HSP by partitioning boxes into groups, and then partitioning planewaves in each box on a group of MPI processes along the θ-direction. Readers can refer to [17] for technical details of the ternary parallelization approach of MLFMA.

3 SW26010 many-core architecture and the Athread programming model

In this scholarly article, a comprehensive exposition of the SW26010 many-core architecture and the Athread programming model is presented by the authors, followed by a detailed exploration of the SW-PMLFMA.

The SW26010 processor is characterized by its heterogeneity and is distinguished from prevailing pure CPU, CPU-MIC, and CPU-GPU architectures. It is composed of four CGs, within each of which one management processing element (MPE) and a CPE cluster of 64 CPEs are housed. As shown in Fig. 1, in a single SW26010 processor, four MPEs and 256 CPEs are encompassed.

The four CGs are interconnected by an on-chip network and typically used as four independent nodes. Additionally, each CPE has two execution pipelines, and its computational power is approximately half that of the MPE. Moreover, two types of memory spaces are featured in the SW26010 processor: an 8GB main memory assigned to each MPE, and a distinct local 64 KB SPM dedicated to each CPE.

The main memory can be accessed directly by all CPEs through global load/store with a bandwidth of 1.45GB/s. Data can also be transferred into the SPM of a CPE with a bandwidth of approximately 22.6GB/s via DMA channels. Additionally, the DMA operation is asynchronous, enabling users to implement hidden communication operations easily. Utilizing the SPM fully and reducing data communications are critical in SW26010 many-core acceleration computation.

For SW26010, MPI and Athread parallel programming models are used for different aims. MPI is a widely-used, standardized, and portable message passing interface that facilitates the management of parallel computations across diverse computational grids. On the other hand, Athread is a directive-based, user-driven and performance-portable parallel programming model that leverages the parallelism of many-core processors. It enables the use of normal fork and join parallelism with up to 64 CPEs in the same CG, with one thread per CPE. In addition, it offers a set of DMA operation interfaces. The processing on the CPEs can be executed asynchronously to the MPE, and a hybrid MPI and Athread parallel programming model

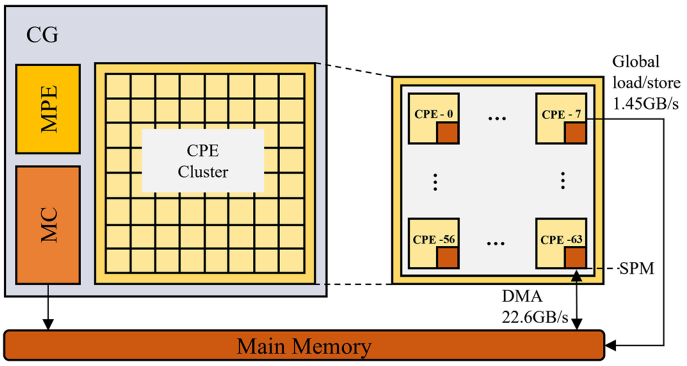

Fig. 1 SW26010 many-core architecture

using the MPE and CPE asynchronous computing pattern is depicted in Fig. 2. The serial phase of the program on each MPI process is first executed on the MPE, followed by the parallel phase where CPEs in the same CG undertake a computation-intensive task. Upon completion of the parallel work by every CG, the MPE host process resumes runtime and executes instructions in the serial phase.

4 Many-core acceleration implementation of the ternary PMLFMA

The MLFMA ternary parallelization approach involves a two-part computation process, namely the setup and the iteratively solution parts. During the setup stage, the near-field matrix, as well as the aggregation/disaggregation matrix at the lowest level, interpolation/anterpolation and translation matrix at each level, is explicitly distributed and stored. In the iteratively solution procedure, each matrix–vector multiplication (MVM) computation is further divided into near- and far-field interactions. The near-field interaction involves a sparse matrix–vector multiplication (SpMV), while the far-field interaction comprises three stages, namely the aggregation, the translation and the disaggregation. To accelerate the computation on SW26010 processors, we propose a many-core accelerated massively parallel MLFMA, which we abbreviate as SW-PMLFMA. This approach is implemented using a hybrid MPI and Athread programming model. The execution order of the different stages of the SW-PMLFMA is illustrated in Fig. 3. In the subsequent sections, we provide a detailed description of the primary stages of the proposed approach.

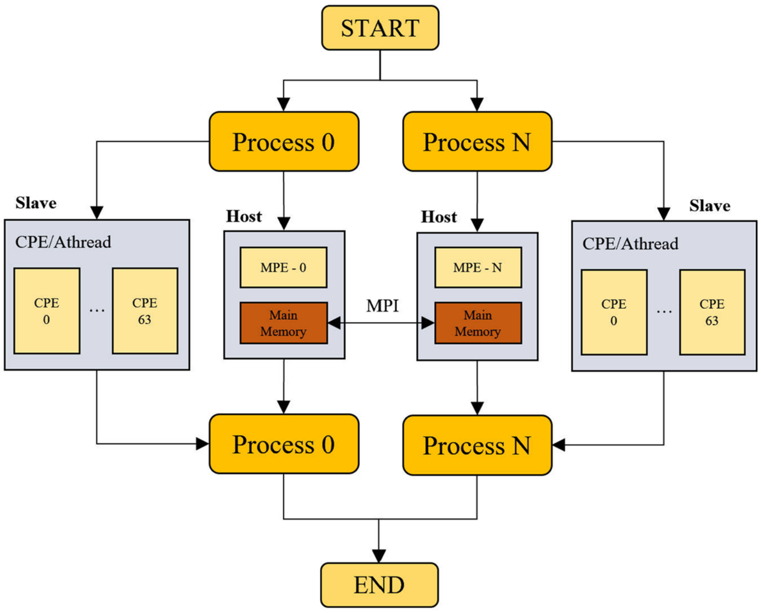

Fig. 2 MPI and Athread parallel programming model

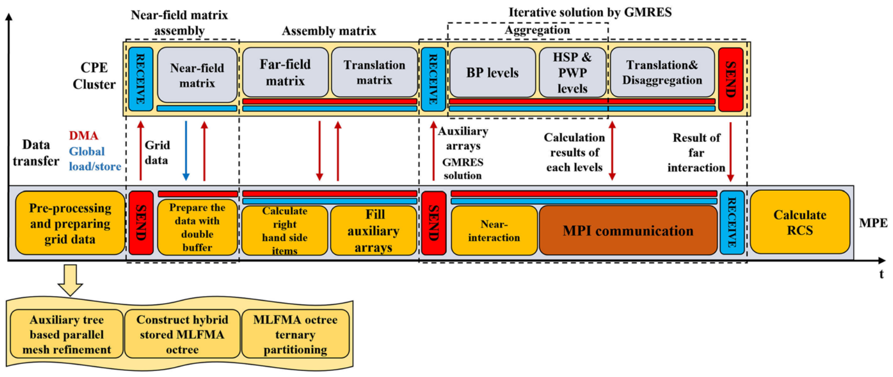
Near-field matrix assembly
Assembly matrix
Iterative solution by GMRES
Aggregation
CPE Cluster
RECEIVE
Near-field matrix
Far-field matrix
Translation matrix
RECEIVE
BP levels
HSP & PWP levels
Translation& Disaggregation
SEND
Data transfer
DMA
Global load/store
Grid data
Auxiliary arrays GMRES solution
Calculation results of each levels
Result of far interaction
Pre-processing and preparing grid data
SEND
Prepare the data with double buffer
Calculate right hand side items
Fill auxiliary arrays
SEND
Near-interaction
MPI communication
RECEIVE
Calculate RCS
MPE
t
Auxiliary tree based parallel mesh refinement
Construct hybrid stored MLFMA octree
MLFMA octree ternary partitioning

Fig. 3 Specific implementation process of SW-PMLFMA

4.1 Setup stage

At the very beginning of the program, preparation tasks include the auxiliary tree-based parallel mesh refinement, the constructing and partitioning of the MLFMA octree, which are executed by using only MPE. After the preparation tasks are completed, matrices filling is done with the aid of CPEs. Since the total number of CPEs in each CG is only 64, the idea of 'one CPE per observer' is used. The observer stands for a near box pair for near-field matrix filling, a box for the aggregation/disaggregation matrix filling, a box pair for the translation matrix filling, etc. Taking near-field matrix filling as an example, there are far more boxes than CPEs at all the lowest MLFMA level. Hence, when partitioning workload among CPEs during near-field matrix filling, computation associated with a box is assigned to a CPE. During execution, the required mesh data for RWG basis functions in a near box pair are fetched by each CPE, followed by the calculation of corresponding nonzero entries in the near-field matrix as dictated by the MoM procedure.

As studied in [28], the mesh data need to be preprocessed based on the structure of array to make the data in the order of calculation, which significantly reduces the total times of accessing global memory and in turn improves overall computation efficiency. The double buffering scheme proposed in [28] is also adopted to make data transition between main memory and the SPM overlapped CPE's computation. Similar optimizations are also required for these arrays used for filling far-field matrices, the aggregation/disaggregation matrix, the interpolation/anterpolation matrix, the translation matrix, etc. In the study, the asynchronous features of SW26010's MPE and CPE were leveraged. For instance, data requisite for double buffering was prepared by the MPE during the CPE's near-field matrix population. Concurrently, while the CPE populated the far-field matrix and transfer matrix, the right-hand side was computed by the MPE, and other auxiliary arrays were filled.

Since the SPM of each CPE in SW26010 is very small, frequent data transferring is inevitable. The storage formats of different matrices should be carefully designed to improve data access efficiency in the iterative solution procedure through DMA. When local Lagrange interpolation is adopted to match different sampling rate of planewaves between successive levels, each target point in the fine grid has contributions from at most 16 neighboring points in the coarse grid, as shown in Fig. 4. The interpolation operation is in fact a SpMV products when local Lagrange interpolation is adopted, where the number of nonzero entries for each row of an interpolation matrix is no more than 16. The anterpolation matrix is the transpose of the interpolation matrix; therefore, the number of nonzero entries for each column of an anterpolation matrix is no more than 16. In coincidence with the special characteristic of local Lagrange interpolation, the CSR and the CSC sparse matrix storage format is used for storing interpolation and anterpolation matrices, respectively. To reduce the amount of storage that is required, the symmetry of the plane waves is used on the BP levels, which can significantly save the memory for storing lowest level aggregation/disaggregation matrix, the translation matrix and the interpolation/anterpolation matrix can be achieved.

4.2 Iterative solution stage

After finishing the setup stage, the final matrix equation system is solved iteratively, with the aid of MLFMA to speed up MVM in each iteration. These coefficients are multiplied by the near-field matrix to evaluate the near interaction. The near interaction evaluation can be done with the aid of CPEs using the idea of 'one CPE per box'. The acceleration of near interaction is simple and straightforward. Since the SPM is too small, matrix entries associated with a near box pair and the corresponding coefficients are transferred to SPM via DMA. After finishing computation, the resulting data are transferred back to the main memory, and then the next computation begins. The double buffering scheme is also used to make computation and data transformation overlapped. The far-field interaction calculation is far more complicated. At the very beginning, the coefficients provided by the iterative solver are multiplied by the aggregation matrix to obtain the radiation patterns. Since the number of edges and planewaves in a box is limited, the idea of 'one CPE per box' is used for aggregation on the lowest level. Each lowest level box is assigned to a CPE, and contributions from each edge source in the box to different direction planewaves is calculated in sequence. The contribution to a same plane wave from all edges in a

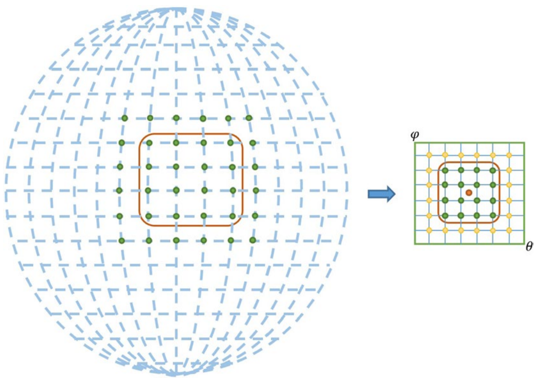

Fig. 4 Local interpolation of a higher-level plane wave by lower-level waves

box should be summarized to obtain the result. Then, the radiated plane waves of the child boxes at the lower level are interpolated and central shifted to obtain the radiation plane waves of boxes on the higher level. This process persists until the second level of the MLFMA tree is achieved. Different parallelization strategies are adopted for CPEs at different MLFMA tree levels. On these levels that the total number of boxes partitioned to the process is far greater than 64, the idea of 'one CPE per box' is used in the simple partitioning strategy. However, on the remaining high levels, for example the second highest level, the number of nonempty boxes is comparable or even smaller than the number of CPEs. If we still use the simple partitioning strategy, serious load unbalance occurs. To overcome this problem, the hierarchical partitioning strategy is selected, the idea of 'one CPE group per box' and 'one CPE per planewave group' is used. Firstly, 64 CPEs are divided into groups with the same number of CPEs, and then partitioning plane waves in a box is partitioned along the θ-direction on CPEs in the same CPE group. Being aware of that the number of plane waves in each box decreases twofold in the θ-direction from the current level to the next lower level, the number of CPE group should increase at the same speed, as shown in Fig. 5.

The interpolation operation between plane waves on two successive levels is in fact a SpMV products, in which the input and output vectors are plane waves of a child box and its parent, respectively. As discussed in the last section, when local Lagrange interpolation is adopted, nonzero entries of each row of the interpolation matrix which is stored in CSR format, is no greater than 16. Hence, in each interpolation operation of a child box, the rows of the interpolation matrix are partitioned into several subblocks. In each time, all nonzero entries of a subblock is transferred into SPM via DMA. On lower MLFMA levels, the box size is limited, plane waves in each child box can be cached in the SPM completely. However, on higher levels, the memory requirement for caching the whole plane waves of a box exceeds SPM size. If the input vector is stored in main memory, a notable computational deficiency is introduced due to the costly memory access latency from irregular access patterns to the input vector.

To solve the problem, a specially designed cache mechanism of hybrid dynamic and static buffers using the SPM is proposed in Fig. 6. In the proposed buffer scheme, a large fixed size static buffer and a small dynamic buffer are used. In each computation,

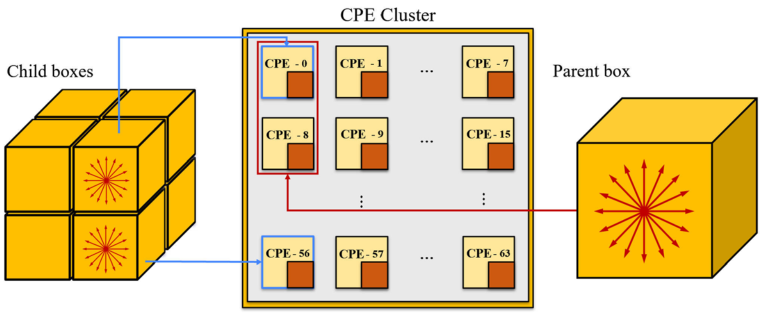

Fig. 5 Illustration of hierarchical partitioning scheme for aggregation on high levels

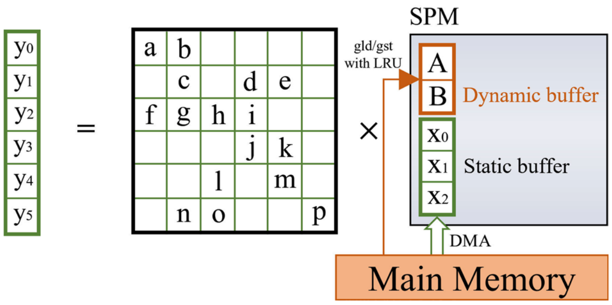

Fig. 6 Dynamic and static combination buffer

the static buffer first caches a fixed length of continuous input vector entries in it, where the start row number equals to the column number of the first nonzero submatrix entry. During calculation, if row number of a required input vector entry is judged to be not cached in the static buffer, it will be searched in the dynamic buffer. If it still not founded in the dynamic buffer, a small length of row continuous input vector entries, with the not founded one as the first, will be fetched from the main memory into the dynamic buffer. Such procedure continues until the SpMV is finished. The result vector is stored continuously and finally transferred back to main memory and stored in the corresponding position.

The disaggregation is conducted oppositely the traversal direction for aggregation. The parallelization strategy used for disaggregation is similar to the aggregation stage. The anterpolation operation in disaggregation is also a SpMV, with the input and output vectors are plane waves of a parent box and its child, respectively. For the same two boxes, the anterpolation matrix is the transpose of the interpolation. When stored in CSC format, the nonzero entries of each column are stored continuously, with maximum number of nonzero entries no greater than 16. Different from the interpolation computation, the input vector entries are continuously cached into SPM, while the output vector entries are stored using the hybrid static and dynamic buffer scheme, where the dynamic buffer is used for writing data back to the main memory. The use of hybrid dynamic and static buffers scheme improves data access efficiency significantly, which in turn improves the computation efficiency. During disaggregation, translation operations of plane waves belonging to the second near boxes on the same level are done at the same time. The parallelization strategy used for disaggregation is also the same as that for the aggregation and disaggregation stages. At the lowest level, the incoming plane waves are multiplied by the disaggregation matrix and received by the testing functions. The final calculation results are then summed with the near interaction results to complete the matrix–vector multiplication.

5 Numerical results and analysis

This section presents a series of numerical experiments to assess the accuracy, efficiency and performance of SW-PMLFMA on the Sunway Taihulight supercomputer. The computational frequency of each MPE and CPE of an SW26010 CPU was 1.45GHz, with MLFMA being executed solely on MPEs, while SW-PMLFMA was executed with 64 CPEs to enhance the computation. The proposed approach was firstly validated by comparing the Radar Cross Sections (RCSs) of a conducting sphere with Mie series, where a 30 m diameter sphere was illuminated by a 0.6 GHz plane wave and discretized into 786,597 triangular patches with 2,312,652 unknowns. In total, 28 CGs were employed for the computation. The calculated results, together with Mie series and MLFMA, are presented in Fig. 7. According to this figure, the calculated results of the SW-PMLFMA well agree with the Mie series, with a root mean square (RMS) error of 0.2186 dB. The RMS error of RCS is defined as follows:

$$\mathrm{RMS} = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\left|\sigma_{\mathrm{ref}} - \sigma_{\mathrm{cal}}\right|^2} \tag{7}$$

The computational statistics for the sphere with diameter 30 m were presented in Table 1, in which ‘MPE’ and ‘CPE’ represent the MLFMA and the SW-PMLFMA, respectively. V_s and V_f denote aggregation and disaggregation matrix assembly on the lowest MLFMA level, and Z_{near} denotes near-field matrix assembly. As the table shows, the speedup of near-field system matrix assembly part was over 14 times, which was significant because it was highly computational dense. However, the speedup achieved in the aggregation/disaggregation/translation process was not as

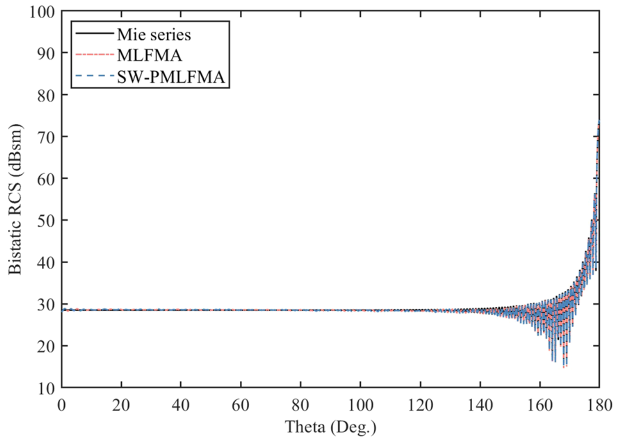

Fig. 7 VV-polarized bistatic RCS of a sphere of diameter 30 m, where 0 and 180 correspond to the back-scattering and forward-scattering directions, respectively

Table 1 Speedup of bistatic RCS calculation of a conductor sphere of diameter 30 m at 0.6 GHZ

	Time (s)			Speed up		
	MPE	1CPE	64CPE	MPE	1CPE	64CPE
V_s and V_f	2.02	11.93	1.03	1	0.17	1.96
Z_{near}	637.96	1267.55	44.51	1	0.50	14.33
Aggregation	34.38	151.86	5.20	1	0.23	6.61
Translation	77.07	696.15	25.13	1	0.11	3.07
Disaggregation	30.32	213.69	5.32	1	0.14	5.70
Total	781.75	2341.18	81.19	1	0.33	9.63

pronounced, owing to the fact that the speedup of the far interaction was restricted not only by the data communications between the MPE and CPE but also by the MPI communications between MPEs. Nevertheless, the overall speedup achieved, which was almost 10 times, was a highly commendable result.

In order to demonstrate the efficiency of the parallel approach, scattering by a sphere was solved with varying numbers of CGs ranging from 4 to 40, using 1 MPI process and 64 CPEs per CG. The parallel efficiency is plotted against the number of CGs in Fig. 8. The parallel efficiency was defined as 100% when 4 CGs were employed. It is worth noting that the parallel efficiency may occasionally exceed 100% due to the limited number of levels in the MLFMA and the varying level sizes. As illustrated in the aforementioned figure, the parallel efficiency drops to 84.2% when 60 CGs were employed. The decrement in computing efficacy can be attributed to the escalation in the number of CGs, resulting in a reduction of the computing time and a concomitant rise in the time taken for MPI communication.

To exemplify the efficiency and capability of the proposed method, a plane model as displayed in Fig. 9 was employed. The aircraft has a length of 38 ms and the

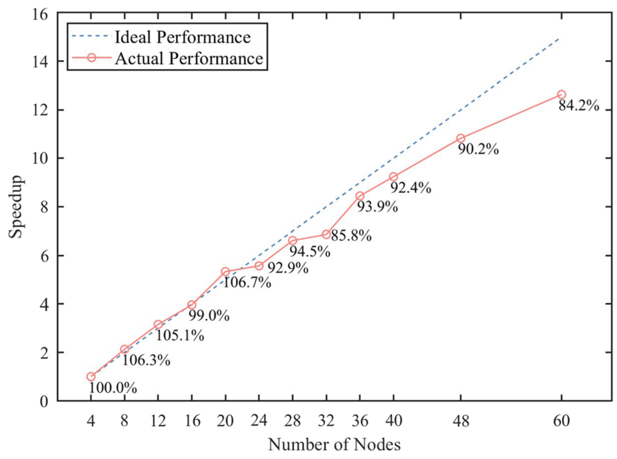

Fig. 8 Parallel efficiency for the solution of a sphere of diameter 30 m involving 2,312,652 unknowns

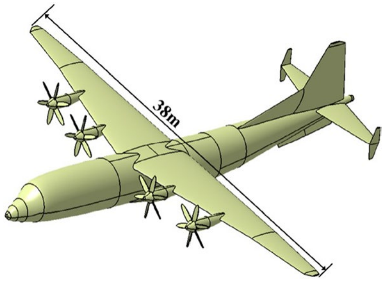

Fig. 9 Geometric illustration of an aircraft model

incident plane wave frequency was set at 1.34 GHz. The calculation utilized a final mesh comprising of 1,873,675 triangular patches, resulting in a total of 5,473,044 unknowns. In the process, 80 MPEs and 5,120 CPEs were utilized, and simulation details are presented in Table 2. The near-field system matrix assembly part demonstrated a speedup of approximately 10 times, whereas the overall speedup achieved was nearly 9 times, indicating an exceptional performance of the proposed method.

In order to compare the method presented in this manuscript with the approach mentioned by the authors in [27], numerical experiment is conducted on both the GPU and the SW26010 platforms using the same example. For fairness, NVIDIA Tesla K80 GPU released at almost the same time as SW26010 was chosen. The chosen platform was equipped with an NVIDIA Tesla K80 GPU and two Intel E5-2640V4s, with a total power consumption of 480w. The power consumption of a single SW26010 chip has not been publicly disclosed. As reported in the Green 500 list, the total power consumption of the Sunway Taihulight supercomputer is approximately 15.3MW, with a Power Usage Effectiveness (PUE) of around 1.22, leading to a net computing power consumption of about 12.5MW. This supercomputer comprises 40,960 computing nodes, implying an average power consumption of 306w per node. A simulation was conducted on a metal sphere model using one node from the SW26010 platform and the GPU platform previously mentioned. Upon discretization, this model consisted of 197,192 triangles and 295,788 edges. The total computation time on the SW26010 platform was 41.4 s, whereas it was 45.49 s on the GPU platform. Detailed information can be found in Table 3. Taking into account the different power consumptions of the two platforms, the PMLFMA running on SW26010 was approximately 1.72 times faster than its GPU version,

Table 2 Speedup of bistatic RCS calculation of an aircraft at 1.34 GHZ

	Time (s)			Speed up		
	MPE	1CPE	64CPE	MPE	1CPE	64CPE
V_s and V_f	2.76	15.39	1.71	1	0.18	1.61
Z_{near}	1065.02	2632.2	111.31	1	0.40	9.57
Aggregation	17.7	52.2	2.38	1	0.34	7.44
Translation	26.9	222.38	7.85	1	0.12	3.43
Disaggregation	17.00	95.7	4.67	1	0.18	3.64
Total	1129.38	3017.87	127.92	1	0.37	8.83

Table 3 Comparison of calculation time for electromagnetic scattering of metal sphere in Tesla K80 and SW26010

	Time (s)	
	Tesla K80	SW26010
V_s and V_f	0.23	0.49
Z_{near}	6.82	14.87
Aggregation/Translation/Disaggregation	38.44	26.04
Total	45.49	41.4

demonstrating the advantages and effectiveness of the proposed method and optimization measures.

To compare the method presented in this article with the method presented in [28], we conducted numerical calculations using the same test cases. SW-MLFMA represents the MLFMA algorithm proposed in [28], while the SW-PMLFMA represents the method proposed in this article. The computations were performed on the SW26010 platform, employing the same computing resources, including 1 computing node consisting of 1 MPE and 64 CPE cores. A metal sphere model was computed after discretization, comprising 313,534 triangles and 470,301 edges. The detailed information of numerical calculation results is shown in Table 4. By utilizing the CSC/CSR sparse matrix storage format and implementing a hybrid dynamic and static buffer cache scheme, we effectively handled the challenges associated with large-scale matrix vector multiplication problems.

To demonstrate the computational proficiency of the SW-PMLFMA approach, the frequency of the incident plane wave was raised to 15GHz. The aircraft model surface was discretized into 213,243,788 triangular patches, resulting in a total of 633,132,588 unknowns. In the computational procedure, a combination of 1,600 MPEs and 102,400 CPEs is utilized, leading to a peak memory usage of 2.50TB. After 63 iterations, a relative residual error of 0.005 was attained within a computation time of 103 min.

The calculated results of SW-PMLFMA and PMLFMA are presented in Fig. 10. According to this figure, the calculated results of the SW-PMLFMA well agree with the PMLFMA, with a RMS error of 0.0258 dB, while additional information regarding the simulations is tabulated in Table 5.

Table 4 Comparison of electromagnetic scattering time between SW-MLFMA and SW-PMLFMA for metal sphere

	Time (s)		Speed up	
	SW-MLFMA	SW-PMLFMA	SW-MLFMA	SW-PMLFMA
V_s and V_f	13.7	2.68	1	5.11
Z_{near}	119	68.72	1	1.73
Aggregation/Translation/Disaggregation	273.35	206.31	1	1.32
Total	406.05	277.71	1	1.46

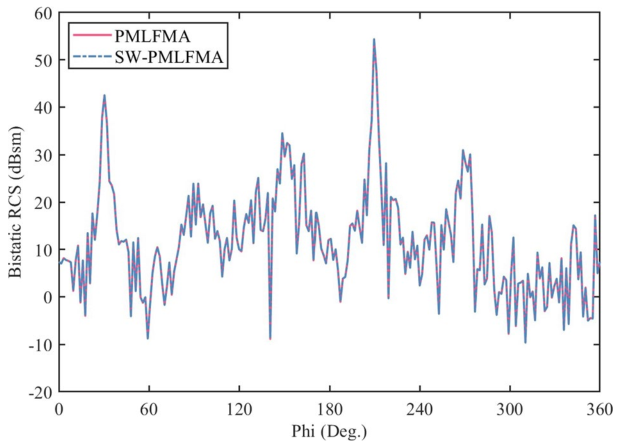

Fig. 10 VV-polarized bistatic RCS of the aircraft model at 15 GHz. The target is illuminated by a plane wave that is propagating in the xy-plane at 30 degrees from the x-axis

Table 5 Simulation details for an aircraft model

Target	Aircraft model
Maximums dimension	1901.31λ
Unknowns	633,132,588
MLFMA levels	8
Number of processes	1600
Number of MPEs	1600
Number of CPEs	102,400
Matrix filling time	25.96
Residual error	0.005
Iteration number	63
Iteration time	1 h,5min
Total time	1 h,43min
Peak memory on CPU	2.50TB

6 Conclusions

This study introduces a hybrid approach to parallelizing the MLFMA on SW26010 processors to solve extremely large 3-D scattering problems. The approach utilizes a combination of MPI and Athread parallelization techniques, employing a flexible ternary partitioning scheme for MPI processes and a simple and hierarchical strategy for parallelization on the CPEs at various MLFMA tree levels to achieve a balanced workload. The interpolation and anterpolation matrices are stored in the CSR and CSC sparse matrix formats, respectively. To enhance data access efficiency, a hybrid dynamic and static buffer scheme is proposed. Numerical results demonstrate the effectiveness of the algorithm, with a total speedup of the SW-PMLFMA achieved between 8.8 and 9.6 times compared to the MLFMA on the MPEs. Furthermore, the study includes the results of a RCS analysis of an electrically large aircraft model, comprising over 600 million unknowns.

Acknowledgements This work was supported by the NSFC (Grant Nos. 62231003 and 61971034).

Author contributions X-DL, W-JH and M-LY performed conceptualization; X-DL and M-LY provided methodology; X-DL and W-JH provided software; W-JH, M-LY and X-QS performed validation; W-JH and X-QS carried out formal analysis; W-JH and X-QS performed investigation; X-DL and X-QS provided resources; X-DL and M-LY performed data curation; X-DL and W-JH performed writing-original draft preparation; X-DL and W-JH performed writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Data availability Authors have the required data and supporting materials.

Declarations

Conflict of interest All authors disclosed no relevant relationships.

Ethical approval Not applicable.

References

1. Coifman R, Rokhlin V, Wandzura S (1993) The fast multipole method for the wave equation: a pedestrian prescription. IEEE Antennas Propag Mag 35(3):7–12. https://doi.org/10.1109/74.250128
2. Rokhlin V (1993) Diagonal forms of translation operators for the Helmholtz equation in three dimensions. Appl Comput Harmon Anal 1(1):82–93. https://doi.org/10.1006/acha.1993.1006
3. Song J, Lu C-C, Chew WC (1997) Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. IEEE Trans Antennas Propag 45(10):1488–1493. https://doi.org/10.1109/8.633855
4. Velaparambil S, Chew WC, Song J (2003) 10 million unknowns: is it that big? IEEE Antennas Propag Mag 45(2):43–58
5. Gürel L, Ergül Ö (2007) Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns. Electron Lett 43(9):499–500
6. Waltz C, Sertel K, Carr MA, Usner BC, Volakis JL (2007) Massively parallel fast multipole method solutions of large electromagnetic scattering problems. IEEE Trans Antennas Propag 55(6):1810–1816. https://doi.org/10.1109/TAP.2007.898511
7. Pan X-M, Sheng X-Q (2008) A sophisticated parallel mlfma for scattering by extremely large targets [em programmer's notebook]. IEEE Antennas Propag Mag 50(3):129–138. https://doi.org/10.1109/MAP.2008.4563583
8. Ergul O, Gurel L (2009) A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm. IEEE Trans Antennas Propag 57(6):1740–1750. https://doi.org/10.1109/TAP.2009.2019913
9. Fostier J, Olyslager F (2008) An asynchronous parallel mlfma for scattering at multiple dielectric objects. IEEE Trans Antennas Propag 56(8):2346–2355. https://doi.org/10.1109/TAP.2008.926787
10. Taboada J, Araujo MG, Bertolo JM, Landesa L, Obelleiro F, Rodriguez JL (2010) Mlfma-fft parallel algorithm for the solution of large-scale problems in electromagnetics. Progr Electromagn Res 105(8):15–30. https://doi.org/10.2528/PIER10041603
11. Melapudi V, Shanker B, Seal S, Aluru S (2011) A scalable parallel wideband mlfma for efficient electromagnetic simulations on large scale clusters. IEEE Trans Antennas Propag 59(7):2565–2577. https://doi.org/10.1109/TAP.2011.2152311
12. Pan X-M, Pi W-C, Yang M-L, Peng Z, Sheng X-Q (2012) Solving problems with over one billion unknowns by the mlfma. IEEE Trans Antennas Propag 60(5):2571–2574. https://doi.org/10.1109/TAP.2012.2189746

13. Taboada JM, Araujo MG, Basteiro FO, Rodriguez JL, Landesa L (2013) Mlfma-fft parallel algorithm for the solution of extremely large problems in electromagnetics. Proc IEEE 101(2):350–363. https://doi.org/10.1109/JPROC.2012.2194269
14. Michiels B, Fostier J, Bogaert I, De Zutter D (2015) Full-wave simulations of electromagnetic scattering problems with billions of unknowns. IEEE Trans Antennas Propag 63(2):796–799. https://doi.org/10.1109/TAP.2014.2380438
15. Hughey S, Aktulga H, Vikram M, Lu M, Shanker B, Michielssen E (2019) Parallel wideband mlfma for analysis of electrically large, nonuniform, multiscale structures. IEEE Trans Antennas Propag 67(2):1094–1107. https://doi.org/10.1109/TAP.2018.2882621
16. MacKie-Mason B, Shao Y, Greenwood A, Peng Z (2018) Supercomputing-enabled first-principles analysis of radio wave propagation in urban environments. IEEE Trans Antennas Propag 66(12):6606–6617. https://doi.org/10.1109/TAP.2018.2874674
17. Yang M-L, Wu B-Y, Gao H-W, Sheng X-Q (2019) A ternary parallelization approach of mlfma for solving electromagnetic scattering problems with over 10 billion unknowns. IEEE Trans Antennas Propag 67(11):6965–6978. https://doi.org/10.1109/TAP.2019.2927660
18. Liu R-Q, Huang X-W, Du Y-L, Yang M-L, Sheng X-Q (2021) Massively parallel discontinuous galerkin surface integral equation method for solving large-scale electromagnetic scattering problems. IEEE Trans Antennas Propag 69(9):6122–6127. https://doi.org/10.1109/TAP.2021.3078558
19. Kong W-B, Zhou H-X, Zheng K-L, Hong W (2015) Analysis of multiscale problems using the mlfma with the assistance of the fft-based method. IEEE Trans Antennas Propag 63:4184–4188. https://doi.org/10.1109/TAP.2015.2444442
20. Fu H, Liao J, Yang J, Wang L, Song Z, Huang X, Yang C, Xue W, Liu F, Qiao F et al (2016) The sunway taihulight supercomputer: system and applications. Sci China Inf Sci. https://doi.org/10.1007/s11432-016-5588-7
21. Dongarra J (2016) Sunway TaihuLight supercomputer makes its appearance. Natl Sci Rev 3(3):265–266. https://doi.org/10.1093/nsr/nww044
22. Cwikla M, Aronsson J, Okhmatovski V (2010) Low-frequency mlfma on graphics processors. IEEE Antennas Wirel Propag Lett 9:8–11. https://doi.org/10.1109/LAWP.2010.2040571
23. Guan J, Yan S, Jin J-M (2013) An openmp-cuda implementation of multilevel fast multipole algorithm for electromagnetic simulation on multi-gpu computing systems. IEEE Trans Antennas Propag 61(7):3607–3616. https://doi.org/10.1109/TAP.2013.2258882
24. Phan T, Tran N, Kilic O (2021) Multi-level fast multipole algorithm for 3-d homogeneous dielectric objects using mpi-cuda on gpu cluster. Appl Comput Electromagn Soc J (ACES) 33(03):335–338
25. Tran N, Kilic O (2021) Parallel implementations of multilevel fast multipole algorithm on graphical processing unit cluster for large-scale electromagnetics objects. Appl Comput Electromagn Soc J (ACES) 33(02):180–183
26. Mu X, Zhou H-X, Chen K, Hong W (2014) Higher order method of moments with a parallel out-of-core lu solver on gpu/cpu platform. IEEE Trans Antennas Propag 62(11):5634–5646. https://doi.org/10.1109/TAP.2014.2350536
27. He W-J, Yang Z, Huang X-W, Wang W, Yang M-L, Sheng X-Q (2022) Solving electromagnetic scattering problems with tens of billions of unknowns using gpu accelerated massively parallel mlfma. IEEE Trans Antennas Propag 70(7):5672–5682. https://doi.org/10.1109/TAP.2022.3161520
28. He W-J, Yang M-L, Wang W, Sheng X-Q (2021) Efficient parallelization of multilevel fast multipole algorithm for electromagnetic simulation on many-core sw26010 processor. J Supercomput 77:1502–1516. https://doi.org/10.1007/s11227-020-03308-9