

# MoM-PO/SBR Algorithm Based on Collaborative Platform and Mixed Model

TANG Xiaobin<sup>1</sup>, FENG Yuan<sup>1,2\*</sup>, GONG Xiaoyan<sup>3</sup>

1. China Academy of Electronics and Information Technology, Beijing 100041, P. R. China;

2. Department of Radar Technology, Academy of Air Force Early Warning, Wuhan 430019, P. R. China;

3. Department of Command, Rocket Army Command Academy, Wuhan 430012, P. R. China

(Received 26 March 2017; revised 16 July 2017; accepted 6 June 2019)

**Abstract:** For electromagnetic scattering of 3-D complex electrically large conducting targets, a new hybrid algorithm, MoM-PO/SBR algorithm, is presented to realize the interaction of information between method of moment (MoM) and physical optics (PO)/shooting and bouncing ray (SBR). In the algorithm, the COC file that based on the Huygens equivalent principle is introduced, and the conversion interface between the equivalent surface and the target is established. And then, the multi-task flow model presented in this paper is adopted to conduct CPU/graphics processing unit (GPU) tests of the algorithm under three modes, i.e., MPI/OpenMP, MPI/compute unified device architecture (CUDA) and multi-task programming model (MTPM). Numerical results are presented and compared with reference solutions in order to illustrate the accuracy and the efficiency of the proposed algorithm.

**Key words:** graphics processing unit (GPU); multi-task programming model (MTPM); physical optics (PO); method of moment (MoM)

**CLC number:** TN393.027

**Document code:** A

**Article ID:** 1005-1120(2019)04-0589-10

## 0 Introduction

The development in computational electromagnetic has gradually improved the algorithms in the field of complex electrically large conducting targets. However, all the algorithms have their own advantages and limits. In addition, these algorithms are unable to solve the electrically large electromagnetic targets including mini-structures such as stitch closures, cavities and protrusions on the surface. Since such targets have sophisticated electromagnetic scattering mechanisms, researchers usually adopt the combination of high and low frequencies for the solution. For a target with different electrical size regions, the strong coupling effect exists between the electrically large and small regions<sup>[1]</sup>. Therefore, the coupling effect between the field-based method and the electromagnetic flow-based method must be resolved.

In order to solve this problem, Jakobus et al.<sup>[2]</sup>

applied the method of moment-physical optics (MoM-PO) method in the electromagnetic analysis of targets with dielectric coating structure. However, Ref. [2] did not solve the problem of coupling between MoM and PO regions. Yu et al.<sup>[3]</sup> expanded the work of Jakobus and applied, for the first time, the impedance boundary condition-based MoM-PO method in analyzing the coupling effect between MoM and PO regions. However, their research did not study the current of PO region that changes due to the impact of the zigzag reflections.

Jin et al.<sup>[4]</sup> presented the MoM-PO combined method and considered the coupling between the closed MoM and PO regions. However, they were unable to correctly calculate the interactive sophisticated structure existed in the electrically large PO region.

In 2008, Prof. Zhang et al.<sup>[5]</sup> developed the low-order and high-order parallel out-of-core solving technique. Their technique overcame the physical in-

\*Corresponding author, E-mail address: science\_xiaoxiao@126.com.

ternal storage limitations, greatly expanded the application scope of the MoM, and solved the million-magnitude pure MoM issues with 512 processes. In 2010, Du et al.<sup>[6]</sup> proposed a scheme of multi-row data transmission in the GPU video memory and computer's internal storage. The new scheme significantly reduced the data communication traffic and improved the program performance. However, the scheme requires the transmission of matrices to the GPUS video memory immediately after the LU decomposition of the matrices begins, which limits the computing scope.

On the aspect of platform construction, the GPU is extensively used in the field of high-performance computing due to its strong parallel computing capability. The parallel data processing capability of the GPU far exceeds the computing capability of traditional CPU cores. The new heterogeneous parallel architecture with the feature of "general host processor + accelerated coprocessor" has become a new research direction<sup>[7]</sup> for the development of high-performance computer systems. This type of architecture has been adopted by supercomputers like "Tianhe-1" "Tianhe-2" "Shenwei" and "Blue-ray". The GPU many-core processor brings the turning point for the development of high-performance computer systems, but also faces new challenges. While realizing the domestic production of cores, it is more urgent to realize the domestic production of the platform software and the application programs. The popular parallel programming model and the parallel operating systems of isomorphic CPU have been gradually replaced because it is extremely difficult for them to fully utilize the accelerated computing capability of the heterogeneous parallel system.

Wang et al.<sup>[8]</sup> developed the YPC/CUDA hybrid programming model. The MGP developed by Barak et al.<sup>[9]</sup> extended the OpenMP and established the program development environment for multiple CPU/GPU scenarios during the operation of OpenCL. However, in these hybrid programming models, the access of GPU to data still requires the users to manually copy the data, leading to redundant data copy.

In order to reduce the computing cost and increase the computing scope, the author successfully developed the "multi-algorithm collaborative solver" after spending 5 a in research and development of collaborative computing platform. This paper aims to conclude and summarize the preliminary work. The introduction of the multi-algorithm collaborative solver platform improves the collaborative adaptability of the algorithm and establishes the conversion interfaces between the equivalent surface and the computing targets (MoM and PO/SBR). At the same time, the multi-task flow model set out in this paper is adopted to conduct CPU/GPU tests of the algorithm under MPI/OpenMP, MPI/CUDA, and multi-task programming model (MTPM) modes. Then, the adaptability and the effectiveness of the proposed algorithm are validated by conducting tests using the collaborative computing platform.

## 1 Platform Structure

### 1.1 Distributed platform architecture

In order to conduct large-scale electromagnetic computing on the collaborative platform for the wide-area computing tasks, all electromagnetic computing tasks are submitted to the "multi-algorithm collaborative solver" module. The multi-algorithm breaks down the major computing tasks into several sub-tasks. Afterwards, according to a certain specification or standard, these sub-tasks are packaged, encoded and linked together in turn. Finally, the sub-tasks are presented to the resource collaborative platform of the collaborative computing platform in the form of sub-task resource requirement linked list. Once the sub-task resource requirement linked list is received from the multi-algorithm collaborative solver, the collaborative computing platform distributes the resources for the computing tasks based on real-time condition of the platform resources in order to rapidly return the computing results.

In this paper, the solver-platform interface is referred as solver platform interface (SPI). The SPI is designed to provide the standard of data transmission between the platform and the solver. Through the SPI, the solver divides a major computing task

into several sub-tasks. The resource requirement linked lists are then sent to the platform for demand analysis. The platform completes the resource planning for each computing task and makes dispatching decisions. After the platform computes the task, the results are returned to the solver. The process is shown in Fig.1.

The SPI represents the interface module that consists of two parts: the resource request interface of the solver and the computing feedback interface of the platform. The responsibilities of the SPI are: at the solver, converting the resource demand into the JSON string; at the platform, interpreting the JSON string describing the demand into standard interface definition, and submitting it to the platform for resource mapping. The SPI between both terminals is described by the multiple sub-task resource demands in the form of JSON string. These sub-tasks are linked together to form a complete task for the multi-algorithm collaborative solver. Once the computing is completed, the SPI returns the results to the solver.

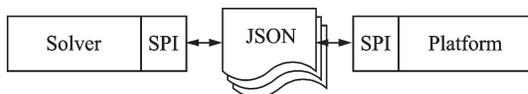


Fig.1 Data transmission relations between the solver and the platform

## 1.2 Task implementation patterns

Considering the architectural feature of the GPU<sup>[10]</sup> heterogeneous parallel system, this paper uses the idea of hybrid programming model to establish a multi-task programming model integrated with the GPU computing core and the CPU control. The architecture of the multi-task model is shown in Fig.2.

Since GPU lacks the ability of self-control, the computing platform is composed of the GPU arithmetic system and the logic control part of CPU.

Considering that the data processing of the algorithm requires few field components, the boundary data communication may be conducted synchronously with the calculation of field value in non-adjacent manner. The stream in CUDA is used to con-

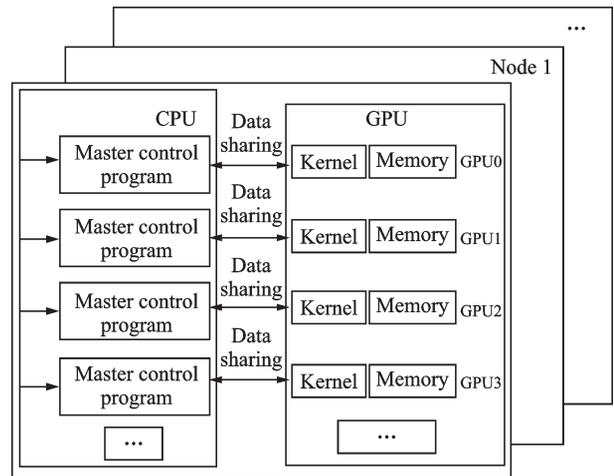


Fig.2 Architecture of the multi-task model

control the mechanism and manage the asynchronous memory copy function. Specifically, the serial execution segment's programmed instruction, which is completed by the collaboration of the CPU and the GPU, is compiled in accordance to the task flow, and the multi-task flow is then used to conduct the parallel execution for the program processing. Each of the task flow includes the master control panel program run by the CPU and the computing kernel run by the GPU. Each of the master control panel program corresponds to one data flow, and each GPU includes multiple blocks as GPU0, GPU1, GPU2, GPU3, etc. The master control panel program should be developed in accordance with the traditional message-passing programming model, while the computing kernel should be developed in accordance with the programming model of CUDA in order to achieve task-level and data-level parallelisms between the processors and the multi-GPU processing units, respectively. The entire algorithm program shows the parallel features at both task- and data-levels. Therefore, the advantage of the multi-level parallelism for hardware should be fully utilized. The message communication mode between the parallel tasks is used, and it should be consistent with the traditional message communication interface of the MPI for the convenience of user access. This will also guarantee the good expansibility and the inheritance of the currently developed parallel applications.

### 1.3 Realization of parallel task flow

The proposed algorithm is implemented at the GPU colony of the high-performance computing center based on the CUDA software environment. The algorithm realizes the multi-task flow programming computing model with the integration of a new concept and the multiple parallel task flow. This model takes the MPI as the underlying communication architecture and realizes the parallelism among the respective GPU node using the Pthread method, as shown in Fig.3.

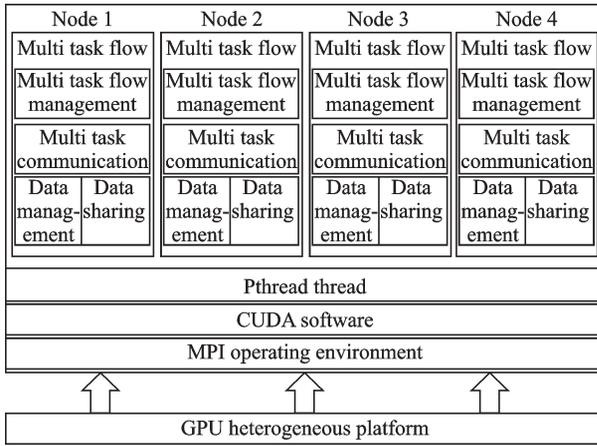


Fig.3 The proposed algorithm flow

The task management includes the start, distribution, synchronization and end of the parallel tasks. The multi-task flow in this paper adopts the integration of process, thread and flow as shown in Fig.4. The task management uses the management method with the integration of MPI\_process, Pthread and CUDA data flow. Specifically, the task management completes the creation, end and synchronization operations of process, thread and flow, respectively, through the interface constructed by the underlying software environment like MPI, Pthread and CUDA. Since there is no direct synchronous call for multi-thread synchronization, this paper achieves the multi-thread synchronization by changing the thread condition variable. The correlation between the thread and the CUDA flow is completed using the private variables of the thread. Each task flow in the nodes has an ID computed by using the process ID of the MPI process, of which the task flow belongs to and the sequence number of the task flow in the process. Suppose that there are  $N$  task flows (threads) in one MPI process, then the task flow ID number = MPI process ID  $\times N$  + process ID of the sub-task flow.

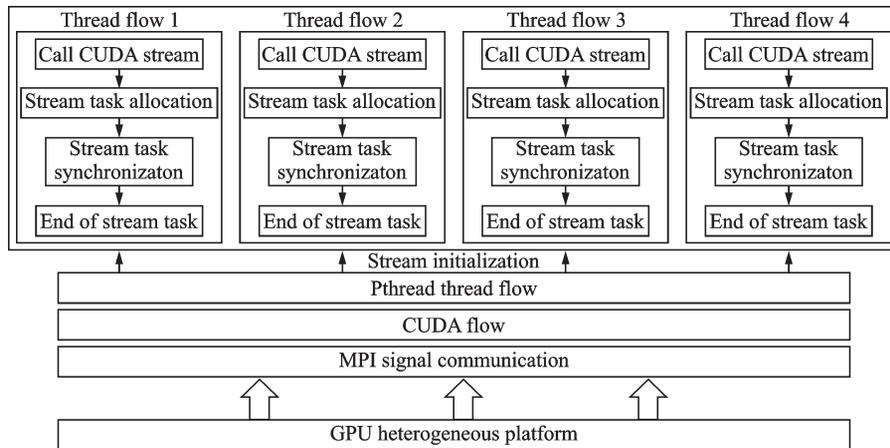


Fig.4 Multi-task flow

## 2 Improvement of MoM - PO/SBR Hybrid Method

This paper divides the target into smooth and non-smooth regions according to the smoothness of the target surface and adopts the PO and the MoM methods for solution, respectively.

For the non-smooth region, we have

$$\sum_{n=1}^N E^i = E^{i_{inc}} + \sum_{n=1}^N E^{s_{po}}, \sum_{k=1}^K E^{s_{po}} = E^{E_{po}} + E^{M_{po}} \quad (1)$$

where  $E^{E_{po}}$ ,  $E^{M_{po}}$  represent the electric-field strength of the current and the magnetic current generated at the non-smooth region, respectively. Then the matrix at the non-smooth region can be gained as

$$(\mathbf{Z}^{\text{MoM,MoM}} + \mathbf{Z}^{\text{MoM,PO}} \cdot \mathbf{P}^{\text{PO,MoM}}) \cdot \mathbf{I}^{\text{MoM}} = \langle \mathbf{E}^{\text{inc}}, f_m \rangle - \mathbf{Z}^{\text{MoM,PO}} \cdot \mathbf{P}^{\text{PO,inc}} \quad (2)$$

where  $\mathbf{Z}^{\text{MoM,MoM}}$  is the matrix of self-action at the non-smooth region, and

$$\langle \bar{\bar{\boldsymbol{\eta}}} \cdot f_n, f_m \rangle - \langle \mathbf{E}^E + \mathbf{E}^M((\bar{\bar{\boldsymbol{\eta}}} \cdot f_n) \cdot \mathbf{n}), f_m \rangle \quad (3)$$

where  $f_m$  is the function group and the same with the primary function;  $\mathbf{Z}^{\text{MoM,PO}}$  the matrix with the interaction of the non-smooth and the smooth regions;  $\mathbf{P}^{\text{PO,MoM}}$  the excitation of the non-smooth region to the smooth region; and  $\mathbf{Z}^{\text{MoM,PO}} \cdot \mathbf{P}^{\text{PO,inc}}$  the matrix with the coupling effect at the considered smooth region.

For the smooth region, we have

$$\mathbf{J}^{\text{PO}} = \delta \mathbf{n} \times \mathbf{H}_{\text{Total}} = \delta \mathbf{n} \times (\mathbf{H}^{\text{ipo}} + \mathbf{H}^{\text{rpo}}) \quad (4)$$

In the meantime, according to the principle of equivalence, for the lighting region

$$\mathbf{J}^{\text{PO}} = \delta \mathbf{n} \times (\mathbf{n} \times \mathbf{E}_{\text{Total}}) = \delta \mathbf{n} \times (\mathbf{E}^{\text{ipo}} + \mathbf{E}^{\text{rpo}}) / \mathbf{Z}_0 \quad (5)$$

The electromagnetic collaborative computing platform considers the equivalent surface as the interface. The equivalent surface is subdivided with a triangular mesh and the equivalent surface electromagnetic flow is spread based on the Rao-Wilton-Glisson (RWG) basis function. Considering that the mesh forms and the basic functions used on the equivalent surface and MoM are different, the adaptability of the MoM must be revised to establish a conversion interface between the equivalent surface and the MoM computing target.

Equivalent surface current is

$$\begin{aligned} \mathbf{J}_s^{\text{inc}} &= \sum_{i=1}^N j_{si}^{\text{inc}} \Lambda_{si}(r) \\ \mathbf{M}_s^{\text{inc}} &= \sum_{i=1}^N m_{si}^{\text{inc}} \Lambda_{si}(r) \end{aligned} \quad (6)$$

where  $\Lambda_{si}(r)$  is the RWG function,  $J_s^{\text{inc}}$  and  $M_s^{\text{inc}}$  are the incident electromagnetic current values, while  $j_{si}^{\text{inc}}$  and  $m_{si}^{\text{inc}}$  are equivalent to the incident electromagnetic current values. The equivalent electromagnetic current values of the scattering surface are

$$\begin{aligned} \mathbf{J}_s^{\text{sca}} &= \sum_{i=1}^N j_{si}^{\text{sca}} \Lambda_{si}(r) \\ \mathbf{M}_s^{\text{sca}} &= \sum_{i=1}^N m_{si}^{\text{sca}} \Lambda_{si}(r) \end{aligned} \quad (7)$$

That is, the RWG coefficient of the electro-

magnetic flow on the equivalent surface is converted into the targeted near field (NF) using the RWG2NF function module. The NF will be considered as the excitation source of the MoM. The NF on the equivalent surface is converted into the RWG coefficient of the equivalent surface using the NF2RWG function module, i.e., the collaborative computing interface.

The collaborative master control program can realize the collaborative computing by calling the MoM collaborative program. The equivalent surface and the mesh used by the MoM are shown in Fig.5. The MoM collaborative program mainly includes RWG2NF, MoM, and NF2RWG modules. The RWG2NF module transforms the RWG coefficient stored in the COC file into the targeted NF. The MoM module is responsible for the simulation of the targeted electromagnetic specificity and the computing of the NF on the equivalent surface. The NF2RWG module converts the NF computed by the MoM into the RWG coefficient in the COC file.

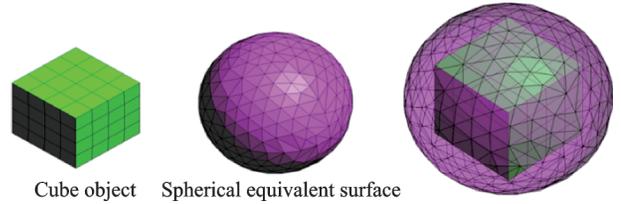


Fig.5 Schematic diagram of the equivalent processing

Finally, the total expression of the electromagnetic field of the equivalent surface is obtained as

$$\begin{aligned} \begin{bmatrix} \mathbf{J}_s^{\text{sca}} \\ \frac{1}{\eta} \mathbf{M}_s^{\text{sca}} \end{bmatrix} &= \begin{bmatrix} -\hat{\mathbf{n}}(r') \times \mathbf{K}_{EM}^S \\ -\frac{1}{\eta} \hat{\mathbf{n}}(r') \times \mathbf{L}_{EJ}^S \end{bmatrix} \cdot [\mathbf{L}_{pp}]^{-1} \cdot \\ &= \mathbf{S} \cdot \begin{bmatrix} \mathbf{J}_s^{\text{inc}} \\ \frac{1}{\eta} \mathbf{M}_s^{\text{inc}} \end{bmatrix} \quad (8) \end{aligned}$$

The transformation of electromagnetic flow is realized by the equivalent transformation. The equivalent surface conversion can be achieved by solving the upper model, that is, the computational collaboration interfaces, as shown in Fig.6. The MoM collaborative program can realize the interaction of the electromagnetic field information with PO/SBR by using the COC file.

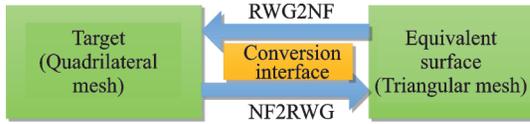


Fig.6 Synergy improvement of the MoM method

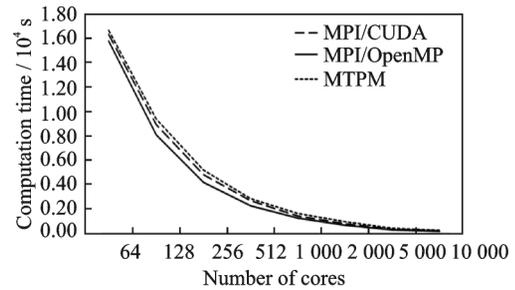
Based on the unified framework of equivalent Huygens algorithm, the collaborative calling interface between the MoM and the PO/SBR algorithms is established. In the traditional high-low frequency mixed method, the computation of the coupling matrix between the two areas is required to revise the PO/SBR current in the MoM method area. Therefore, the largest bottleneck of the mixed algorithm of MOM-PO/SBR lies in the multiplication of these two large matrices that require substantial amount of time. Provided that there are  $N$  and  $M$  unknown values in the MoM and the PO/SBR areas, respectively, the scales of these two matrices would be  $N \times M$  and  $M \times N$ , respectively. The computing task of the multiplication would be  $N^2 M$ . Therefore, reducing the quantity of the unknown values will accelerate the algorithm by square. Through replacing the original object with the equivalent surface of the MoM area, the effect on the PO/SBR can be coupled into the equivalent surface (ES) and the system equation can be solved iteratively on ES. This can reduce both the solving scale of the coupling matrix and the demand of the algorithm on internal storage.

While obtaining the current with higher precision on the surface of the smooth region using ray tracing, it can only obtain the multiple reflection field values of ray tubes. The field of the ray tubes cannot be effectively transformed into the current of the smooth region in the receiving processes of the ray tubes. Hence, this paper uses the ray-density normalization (RDN) idea to obtain the field solution of the scattered rays, and then obtain the equivalent current on the face unit by interpolation.

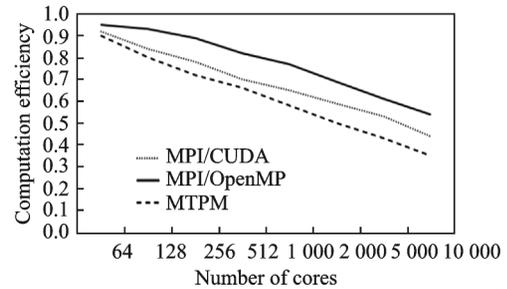
### 3 Parallel Implementation

The hybrid programming model of MPI/OpenMP, MPI/CUDA and MTPM are used for the acceleration test comparison between the CPU and the GPU.

First, different numbers of CPU cores are used to calculate the radar-cross section (RCS) of the model. The tests on the MPI/OpenMP and the MPI/CUDA platforms have no task flow control. Therefore, only one task flow is initiated in the test of MTPM model for control with 64, 128, 256, 512, 100, 2 000, 5 000 and 10 000 as the numbers of cores (Fig.7).



(a) Computing time comparison under single task flow



(b) Computing efficiency comparison under single task flow

Fig.7 Parallel performance comparison under single task flow

Secondly, only one task flow is initiated to test the computing speed and the computing efficiency under different numbers of cores of the GPU (Fig.8).

Lastly, two blocks on the GPU node and two cores of the CPU are randomly selected to test the computing speed and the computing efficiency under the three different platform models and the different numbers of task flows.

On the colony, the scalability of the algorithmic routine is tested first and results are shown in Fig.7. It can be seen from the figure that the decline tendencies of the computing time curve under the three platform models are basically consistent. The computing time at the same testing point is  $t_{\text{MTPM}} > t_{\text{MPI/CUDA}} > t_{\text{MPI/OpenMP}}$ , and the computing effect of the MTPM model under the single task flow mode is bad. The parallel efficiency is significantly reduced

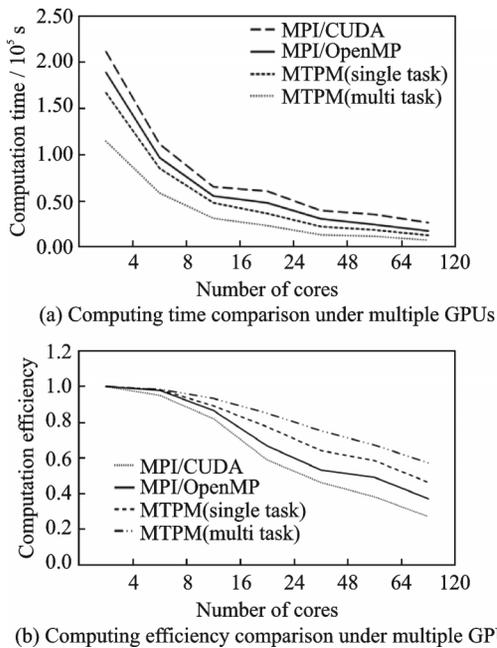


Fig.8 Multi-GPU parallel performance comparison under single/multiple task flows

with the increase of cores with over 90% for 64 cores. When the number of cores is increased to 10 000, the parallel efficiency of the MTPM platform is the lowest (only 35%), and the model of MPI/OpenMP is the highest (54%).

The high time consumption and low efficiency of the MTPM model are mainly due to the initiation of the task flow. The page locks the memory for message buffer, and the locking and unlocking of the message buffer cause additional time and efficiency loss. With the increase of communication data, although the MTPM model has reduced the data transmission process, there is no significant change in the additional expense ratio. Hence, the delay in the MTPM model is still larger than the delays in the MPI/OpenMP and the MPI/CUDA models.

Since the testing results of the MTPM model under single task flow control are relatively unsatisfactory, the single task control testing is not conducted after the interface is called collaboratively through the mixed algorithm.

The multi task flow is developed under the MTMP model. It can be seen from Fig.8 that there is no significant difference in computing time and efficiency among MPI/OpenMP, MPI/CUDA, MTMP (single task) if the core (block)/task flow

is small. With the increase of cores (blocks)/task flow, the computing advantage of multi-task flow stands out, the computing time is reduced by 21.6% and the computing efficiency is improved by 13% at 120 cores (blocks)/flow. Comparing the tests of the single task and the multi-task flows under the MTMP model, the computing time of the multi-task flow is significantly reduced and the efficiency is notably improved. However, with the increase of cores (blocks) and the increase of task flow, both the reduction range of the increased computing time and the efficiency are reduced.

The CPU is used as the host to collaborate with the GPU as the coprocessor, and it is responsible for the transaction handling and the serial computing that require strong logicity, while the GPU focuses on the implementation of highly threaded parallel processing tasks. In the meantime, the task flow is also added for control to test the computing time and the efficiency of the multiple task flows and the multiple cores under the three models.

At the middle and the later stages of the research, the MTPM model is set up in the collaborative computing platform to test the collaborative interface calling of the mixed algorithm. According to the above-described conclusions that appropriate numbers of CPU cores and GPU blocks should be selected to ensure computing efficiency, the multi-task flow comparison test is conducted with 512 CPU cores and 64 GPU blocks.

Fig.9 shows the cases of calling and not calling the platform interface. "Not calling the platform interface" indicates the results of not using the method proposed in this paper; while "calling the platform interface" indicates the results after using the proposed method. The computing time in both cases decreased significantly, but the performance is much better when the interface is called. With regard to the computing efficiency, under the condition of calling the platform, the decreasing trend of efficiency rate is smoother and the efficiency is higher than that with not calling the platform interface. The results validate that the multi-task flow model can satisfactorily adapt to the computing demand of the collaborative platform interface.

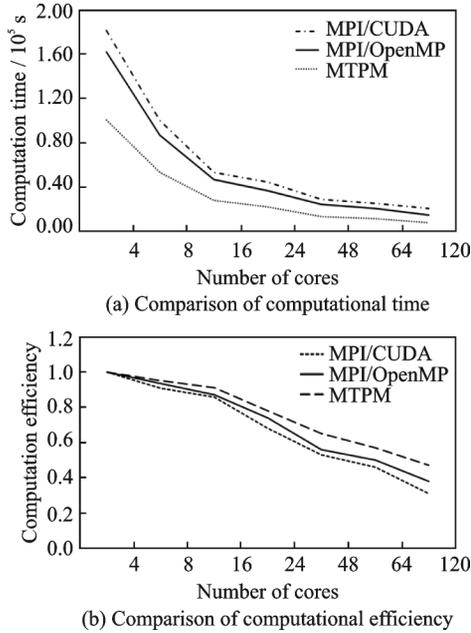


Fig.9 Parallel performance comparison in multi-task streams

From Figs.7, 8 and 9, the parallel efficiency always shows a decline tendency with the increase of cores (blocks) for both the colony of GPU and the colony of CPU. The reason is that the computing time is decided by the volume of computational domain, while the communication time is decided by the surface area of the computational domain. Therefore, with the increase of cores (blocks) i.e., the increase of the divided computational domain, the reduction in the computing time will be greater than that in the communication time. With the increase of computing resource, the proportion of computing time in computing plus communication time will be gradually reduced and the effect of the parallel acceleration will be reduced, which will reduce the parallel efficiency.

Although the parallel efficiency exhibits a decline tendency, the speed of decline is slowed down to some extent, thereby improving the computational efficiency. After the collaboration of the GPU and the CPU, the acceleration ratio reaches the highest value across all the tests in this paper, and the computing time reduction also reached its maximum. However, the parallel efficiency keeps decreasing at a relatively high speed with the increase of task flows. Therefore, during the program operation while using the GPU heterogeneous parallel sys-

tem, the weighted relation between the computational complexity and the communication time should be fully considered for the quantity of the parallel task flows in order to ensure that each GPU is distributed with sufficient computational complexity. It will avoid the idleness and uneven distribution of computing resources, and will guarantee the rational utilization of GPU computing resources.

## 4 Test Comparison

**Example 1** In order to validate the accuracy of the proposed algorithm, a coated metallic ball with a radius of 0.5 m is selected as the research object. The surface of the selected ball has two layers of 15 mm - thick coating medium, as shown in Fig.10. The electromagnetic parameters of the two layers of coating materials from inside to outside are set as follows. The dielectric coefficient and the magnetic permeability of coating layer 1 are  $3-2j$  and  $2-j$ , respectively, while the coating layer 2 has the dielectric coefficient of  $2-j$  and the magnetic permeability of  $3-2j$ . The vertical polarization wave frequency is  $f=10$  GHz, the incident angle is  $\theta=0^\circ$ ,  $\varphi=0^\circ$ , and the scattering angle is  $\theta=0^\circ-180^\circ$ ,  $\varphi=0^\circ$ . The normal direction of the triangle face at the smooth region and the primary function parallel to the incident wave are included in the non-smooth region, while the rest of the primary functions is included in the smooth region. Thus, the unknown numbers at the smooth and the non-smooth regions are 5 147 526 and 2 239 018, respectively, and the angle separation is  $0.1^\circ$ . In the range of  $\theta \in [0, 180^\circ]$ , by using fast blanking process (the surface current of the surface element in the shadowed re-

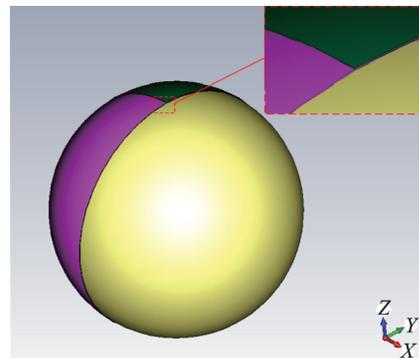


Fig.10 Illustration of the coated ball model

gion of the incident for the PO region is 0, while there is no impact on the MoM area), the primary function for the PO area can be reduced to 1 895 246, and then the double station RCS can be calculated. The computing results and the comparison diagram are shown in Fig.11. It can be seen from the figure that the two results of the MoM-PO/SBR and the Mie solution match well, indicating the accuracy of the algorithm proposed in this paper.

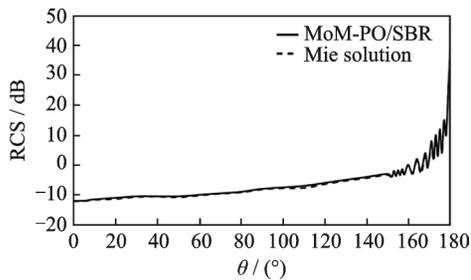


Fig.11 Double stations of the coated ball RCS (VV polarity)

**Example 2** The Yun-Ba aircraft platform loaded with antenna is selected for this second test. In order to conveniently compare the results with the literature data<sup>[11]</sup>, and to reduce the time cost of the parallel test, assume that it is the isotropic case, that is  $Z_u=Z_v$ , and  $n=0$ . The proposed algorithm in this paper is degraded to the situation of perfect electric conductor (PEC). The precision of  $I_p=h/6$  of the triangle face unit is selected for analysis, the incident frequency is  $f=2$  GHz, the antenna is a micro-strip antenna, which is the computational area of MoM. And the body of the aircraft is the computational domain of the PO/SBR. Due to the large geometric size that would consume a large amount of computer memory, Example 2 uses the multi-core CPU under colony environment for calculation. Fig.12 compares the computing result with the ana-

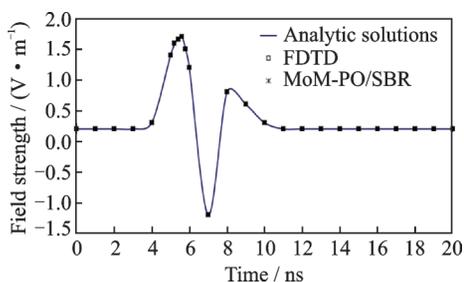


Fig.12 Comparison between the computing results and the analytic solution of the proposed algorithm

lytic solution of the proposed algorithm. The comparison diagram in Fig.12 shows that the two results match well, indicating the accuracy of the algorithm proposed in this paper.

## 5 Conclusions

The MoM-PO/SBR algorithm is developed to solve the complex electromagnetic radiation/scattering problem.

(1) Cooperative computing is implemented based on MoM-PO/SBR collaborative program. The MoM-PO/SBR cooperative program includes three functional modules: the RWG2NF, the high-order MoM and the NF2RWG. The RWG2NF module converts the RWG coefficients from the input COC file into the target surface NF. The high-order MoM module takes the NF generated by the RWG2NF as the excitation source to simulate the electromagnetic characteristics of the target and calculate the NF on the equivalent surface. Lastly, the NF2RWG module converts the NF calculated by the high-order MoM into the RWG coefficients from the output COC file. The MoM collaboration program realizes the collaborative computing with the PO/SBR through the COC file.

(2) The idea of hybrid programming model, a multi-task flow programming model integrating the GPU computing core and the CPU control are developed according to the CUDA programming model to realize task-level and data-level parallelisms between the processors and the multi-GPU processing units, respectively.

(3) The computing time and the efficiency of the proposed algorithm are tested using different CPU core/GPU blocks under different task flow controls. The CPU core and the GPU block numbers reach 10 000 and 120 cores, respectively. Thus, more suitable task flow and hardware computing resource allocations are obtained.

## Reference

- [1] HE Xiaoxiang, LIU Chen, YANG Yang. Fast SSED-MoM/FEM analysis for electromagnetic scattering of large-scale periodic dielectric structures[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2014,31(3): 255-259.

- [2] JAKOBUS U, LANDSTORFER F M. Improved PO-MM hybrid formulation for scattering from three-dimensional perfectly conducting bodies of arbitrary shape[J]. *IEEE Transactions on Antennas and Propagation*, 1995,43: 163-165.
- [3] YU Dingfeng, HE Siyuan, CHEN Xi, et al. Simulation of electromagnetic scattering for 3-D Impedance surface using MoM-PO method[J]. *IEEE Transactions on Antennas and Propagation*, 2012, 60: 145-146.
- [4] JIN Jianming, LING Feng, CAROLAN S T, et al. A hybrid SBR/MoM technique[J]. *Trans Antennas Propag*, 1998,46(9): 1349-1357.
- [5] ZHANG Y, SARKAR T K. Parallel solution of integral equation based EM problems in the frequency domain[M]. Hoboken NJ: Wiley, 2009: 12-24.
- [6] DU Peng, TOMOV S. Providing GPU capability to LU and QR within the ScaLAPACK framework, *Lawn272*, UT-CS-12-699 [EB/OL]. [2012-09] (2017-03). <http://www.netlib.org/lapack/lawns/downloads/>.
- [7] HU Jinhui, QIN Zhichao, SHI Lei, et al. Research on spatial information cloud service platform and application[J]. *Journal of China Academy of Electronics and Information Technology*, 2016,1: 54-57.
- [8] WANG L, HUANG M, NARAYANA V K, et al. Scaling scientific applications on clusters of hybrid multi-core/GPU nodes[C]//*Proceedings of the 8th ACM International Conference on Computing Frontiers*. Ischia, Italy: ACM, 2011: 1-6.
- [9] BARAK A, BEN-NUN T, LEVY E, et al. A package for OpenCL based heterogeneous computing on clusters with many GPU devices[C]//*Proceedings of the Workshop on Parallel Programming and Applications on Accelerator Clusters*. Heraklion, Greece: IEEE Cluster, 2010: 1-7.
- [10] KAN Caixia, LI Yuling, LIU Jinsheng, et al. FDTD simulations on plasmonic properties of end-to-end and side-by-side assembled Au nanorods [J]. *Transactions of Nanjing University of Aeronautics and Astronautics*, 2014,31(3): 248-249.
- [11] FENG Yuan, DAI Xiaoxia, TANG Xiaobin, et al. Parallel FDTD arithmetic simulation based on distributed heterogeneous cluster system [J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2016,42(9): 1874-1883. (in Chinese)

**Authors** Prof. TANG Xiaobin received the B.S degrees in electronics engineering from Xi'an Electronic and Science University, Xi'an, China. She works as a chief scientist in the China Electronic Technology Corporation (CETC). Her research is focused on high performance computing, computational electromagnetics and relevant fields.

Dr. FENG Yuan received the B.S. and Ph. D. degrees in electronics engineering from PLA University of Science and Technology, Nanjing, China, in 2002 and 2006, respectively. From 2011 to 2017, he was a postdoctoral research fellow in the Academy of Electronics and Information Technology, Beijing, China. His research is focused on high performance computing, computational electromagnetics and relevant fields.

Dr. GONG Xiaoyan received the B.S. and Ph. D. degrees in electronics engineering from Department of Command, Rocket Army Command Academy, Wuhan, China, in 2003 and 2007, respectively. From 2011 to 2017, she worked at the Rocket Army Command Academy, Wuhan, China. Her research is focused on campaign command, military equipment and relevant fields.

**Author contributions** Prof. TANG Xiaobin contributed data for the analysis of this study and instructed Dr. FENG Yuan to write the manuscript. Dr. FENG Yuan compiled the algorithm and conducted the analysis. Dr. GONG Xiaoyan contributed to the discussion and background of the study.

**Competing interests** The authors declare no competing interests.

(Production Editor: Zhang Huangqun)