SSD Accelerated Parallel Out-of-Core Higher-Order Method of Moments and Its Large Applications

Zhongchao Lin [1], Sheng Zuo [1], Xunwang Zhao [1], Yu Zhang [1], and Weijun Wu [2]

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

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

Abstract ─ The performance of the parallel out-of-core Higher-order Method of Moments (HoMoM) is analysed in this paper. The I/O to hard disks significantly affects the performance of the out-of-core algorithm. In order to reduce the I/O time, solid state drives (SSD) with high read and write speeds are utilized. The size of the in-core buffer allocated to each process, IASIZE, is tuned to achieve the optimum performance of the out-of-core algorithm. Numerical results show that the out-of-core algorithm using SSD exhibits better performance than that using SAS hard drives. As a challenging application, a slot array with 2068 elements is analysed using this method.

Index Terms ─ HoMoM, out-of-core algorithm, SAS, slot arra, SSD.

I. INTRODUCTION

The method of moments (MoM) is a numerically accurate method for solving electromagnetic radiation and scattering problems [1-3]. However, for large complex objects, MoM needs a large amount of physical memory to deal with the dense impedance matrix. The matrix is usually too large to be stored in the main memory (RAM) of the system. One choice is using fast algorithms combined with iterative solvers, such as the multilevel fast multipole algorithm (MLFMA) [4]. However, slow convergence or even divergence often occurs when an object contains complex structures or diverse materials specially dealing with radiation problems. Other choice is using fast direct solver incorporate compression [5] to reduce the amount of data that has to be stored. However, the low rank property of the matrix is indispensable in this method, so it will expire most of time.

An alternative solution is developing an out-of-core computation method using low-cost hard disks instead of memory to store data and using a direct solver like LU factorization for avoiding the slow convergence issue or the demand for low rank property of the matrix. The out-of-core algorithm is designed according to the multi-layered memory hierarchies of the computation machines, which uses memory as in-core buffer and hard disks to store large matrices. The major task of developing an out-of-core algorithm is adding an efficient hard-disk I/O interface to an in-core algorithm, including both matrix filling and matrix equation solving procedures.

The state of the art high performance computing (HPC) technique can be employed to further improve the capability of the out-of-core algorithm. Due to the overhead of hard-disk I/O, the out-of-core algorithm typically requires more simulation time than its in-core counterpart. To reduce the I/O time, the high-speed SSD are utilized to accelerate the I/O [6, 7]. The radiation of an airborne antenna array is simulated to test the optimum IASIZE. Numerical results show that the performance of the out-of-core algorithm is affected by the value of IASIZE, the simulation time of the algorithm using SSD is reduced compared with that using SAS hard drives, and the developed parallel codes have excellent stability.

II. HIGHER-ORDER METHOD OF MOMENTS

A. Integral equations

The method is based on the solution of Surface Integral Equations (SIEs) [8] in the frequency domain for equivalent electric and magnetic currents over dielectric boundary surfaces and electric currents over perfect electric conductors (PECs). The integral equation employed is the Poggio-Miller-Chang-Harrington-Wu (PMCHW) formulation [9, 10], which is solved in frequency domain for equivalent electric and magnetic currents over dielectric boundary surfaces and electric currents over perfect electric conductors (PECs). The set of integral equations are solved by using MoM, specifically the Galerkin's method.

Submitted On: October 24, 2016
Accepted On: September 9, 2018

For the case when one of the two domains sharing a common boundary surface is a PEC, the magnetic currents are equal to zero at the boundary surface and therefore, the PMCHW formulation degenerates into the electric field integral equation (EFIE) [11].

B. Higher-order basis functions

Higher-order polynomials over bilinear quadrilateral patches are used as basis functions over relatively large subdomains [9],

$$\boldsymbol{F}_{ij}(p,s) = \frac{\boldsymbol{\alpha}_s}{\left|\boldsymbol{\alpha}_p \times \boldsymbol{\alpha}_s\right|} p^i s^j, \tag{1}$$

$$\boldsymbol{\alpha}_p = \frac{\partial \boldsymbol{r}(p,s)}{\partial p} \quad \boldsymbol{\alpha}_s = \frac{\partial \boldsymbol{r}(p,s)}{\partial s}, \tag{2}$$

where, p and s are local coordinates, i and j are orders of basis functions, and $\boldsymbol{\alpha}_p$ and $\boldsymbol{\alpha}_s$ are covariant unitary vectors. The basis orders can be adapted to deal with nonuniform patches; that is to say for each patch may have completely different order in two directions. A quadrilateral patch is illustrated in Fig. 1.

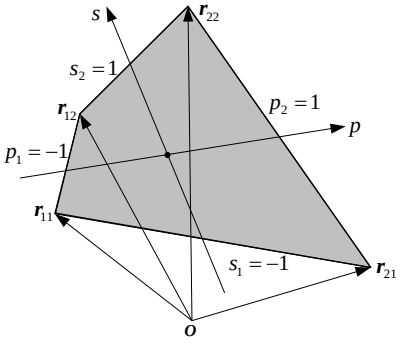


Fig. 1. Bilinear quadrilateral patch defined by four vertices with the position vectors of $\boldsymbol{r}_{11}$, $\boldsymbol{r}_{21}$, $\boldsymbol{r}_{12}$, $\boldsymbol{r}_{22}$.

The orders can be adjusted according to the electrical size of the geometric element. The orders become higher as the element size increases. The electrical size of the geometric element can be as large as two wavelengths. Typically, the number of unknowns for the HOBs is reduced by a factor of 5–10 compared with that for traditional piecewise basis functions, e.g., the RWG basis functions [12], and thus the use of HOBs drastically reduces the computational amount and memory requirement. Note that the polynomials can also be used as basis functions for wire structures. In this case, truncated cones are used for geometric modeling.

For bilinear surfaces, the surface current is decomposed into its p and s-components; p and s being the two parametric coordinates of the unit quadrangle, p, $s \in$ [-1, 1]. The approximation for the s-component of the electric current is (analogous expressions hold for the p-component of the electric current and for the magnetic current):

$$\boldsymbol{J}_s(p,s) = \sum_{i=0}^{N_p} \left[\begin{array}{c} c_{i1}\,\boldsymbol{E}_{i1}(p,s) + c_{i2}\,\boldsymbol{E}_{i2}(p,s) \\ + \sum_{j=2}^{N_s} a_{ij}\,\boldsymbol{P}_{ij}(p,s) \end{array} \right], \tag{3}$$

$$-1 \le p \le 1 \qquad -1 \le s \le 1, \tag{4}$$

where N_p and N_s are the degrees of the approximations along the coordinates, and a_{ij}, c_{i1} and c_{i2} are the unknown coefficients.

Expression (3) stands for the representation of the current in terms of edge basis functions $\boldsymbol{E}_i(p, s)$ and interior or patch basis functions $\boldsymbol{P}_{ij}(p, s)$ which can be compactly expressed as:

$$\boldsymbol{E}_{ik}(p,s) = \frac{\boldsymbol{\alpha}_s}{|\boldsymbol{\alpha}_p \times \boldsymbol{\alpha}_s|} \begin{cases} p^i N(s), & k=1 \\ p^i N(-s), & k=2 \end{cases}, \tag{5}$$

$$\boldsymbol{P}_{ij}(p,s) = \frac{\boldsymbol{\alpha}_s}{|\boldsymbol{\alpha}_p \times \boldsymbol{\alpha}_s|} p^i S_j(s), \tag{6}$$

$$N(s) = \frac{1-s}{2}, \; S_i(s) = \begin{cases} s^i - 1 & i \text{ is even} \\ s^i - s & i \text{ is odd} \end{cases}. \tag{7}$$

Edge basis functions $\boldsymbol{E}_{i2}$ and patch basis functions $\boldsymbol{P}_{ij}$ are zero along the first edge (s=-1); being $\boldsymbol{E}_{i1}$ and $\boldsymbol{P}_{ij}$ zero along the second edge (s=1). Thus, the continuity equation can easily be imposed on a common edge.

Figure 2 shows the different polynomials orders in use for the simulation of an microstrip antenna and the orders range from 1 to 4. The orders of the basis functions over the patches that close to the feed are higher than those over other patches, because the current distribution on these patches changes much faster than that over other patches, as shown in the inserted figure inside Fig. 2.

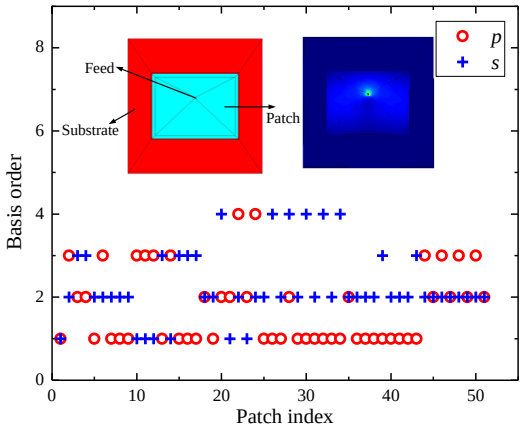


Fig. 2. Basis orders along p and s directions of each quadrilateral patch of the microstrip antenna. The basis orders are adaptive to the patch size along p and s local coordinates.

III. PARALLEL OUT-OF-CORE ALGORITHM

The MoM is based on the solution of Surface Integral Equations (SIEs) [8, 13], and the integral equations are discretized into $N \times N$ dense matrix equations in a general form of:

$$\mathbf{AX} = \mathbf{B}, \tag{8}$$

where **A** is the complex dense matrix, **X** is the unknown vector to be solved for, and **B** is the excitation vector.

The MoM is parallelized through partitioning the large dense matrix into blocks, which are uniformly distributed to Message Passing Interface (MPI) processes in a block-cyclic manner. As an example, assume that matrix A is divided into 6×6 matrix blocks, and distributed it to six processes in the 2×3 process grid. Figure 3 presents the block-cyclic distribution methodology based on ScaLAPACK math libraries. Two factors, process grid and block size, significantly affects the performance of the parallel algorithm. The reader is referred to [14] for more detailed discussion of the theory.

	0	1	2	0	1	2
0	11	12	13	14	15	16
1	21	22	23	24	25	26
0	31	32	33	34	35	36
1	41	42	43	44	45	46
0	51	52	53	54	55	56
1	61	62	63	64	65	66

(a)

0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)
0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)
0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)

(b)

Fig. 3. Block-cyclic distribution scheme: (a) a matrix consisting of 6×6 blocks; (b) rank and coordinates of each process owning the corresponding blocks in (a).

The main idea of designing an out-of-core filling algorithm is to modify the in-core filling algorithm structure and fill a portion of the matrix at a time instead of the whole matrix. The detail procedure is shown in Fig. 4. When performing an out-of-core LU factorization, each portion of the matrix is read into the RAM and the LU decomposition is started. On completion, the result of the LU factorization of this portion of the matrix is written back to the hard disk (HD). The code then proceeds with the next portion of the matrix until the entire matrix is LU factored. The memory is used as the in-core buffer for the out-of-core algorithm, and the buffer size, IASIZE, determines the number of portions. A larger value of IASIZE will lead to higher CPU usage, and this will result in faster computation. However, reading/writing a large file from/to the hard disk will need more time if the memory is not properly allocated. Therefore, the value of IASIZE is a critical factor for obtaining optimum performance. In the following sections, we investigate how the performance of a cluster is affected by the choice of the IASIZE parameter.

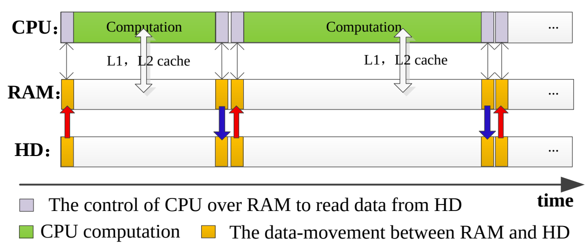


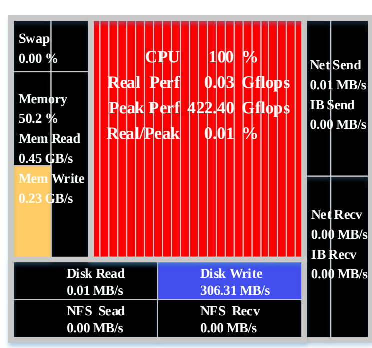
Fig. 4. The out-of-core algorithm.

The left-looking variant of LU factorization, which requires less hard-disk I/O amount than the right-looking variant, is used and the total amount of I/O required is [2, 13]:

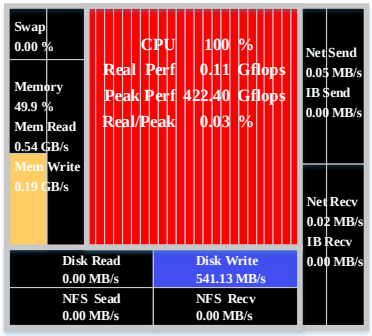
$$\frac{M^3}{2n_b}\left[1+O\left(\frac{n_b}{M}\right)\right]R + 2M^2\left[1+O\left(\frac{n_b}{M}\right)\right]W, \tag{9}$$

where M is the order of the dense matrix, n_b is the matrix block size, and R and W are the time required to read and write one matrix element, respectively.

The read and write speeds of SAS and SSD that be monitored by paramon software [15] are shown in Fig. 5 and Fig. 6. And the read and write speeds of SSD are higher than those of ordinary hard drives, e.g., SAS. Therefore, R and W in (9) can be reduced by using SSD.

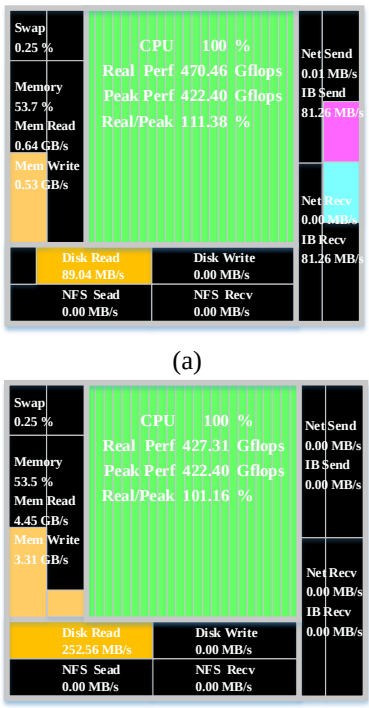


(a)



(b)

Fig. 5. Write speeds of hard disk: (a) SAS and (b) SSD.



(a)

(b)

Fig. 6. Read speeds of hard disk: (a) SAS and (b) SSD.

IV. NUMERICAL RESULTS

The computational platform used in this paper is a HPC cluster with 64 computing nodes. Each computing node has two 12-core Intel Xeon E5-2692v2 2.2 GHz EM64T processors (12×256 KB L2 Cache and 30 MB L3 Cache), 64 GB RAM. Among these 64 computing nodes, 32 computing nodes include SSD and each computing node has two 400 GB Intel MLC SSD, and the other 32 computing nodes include SAS hard disks and each computing node has two 600 GB 10K rpm SAS hard disks. The nodes are connected with Infiniband switches.

A. Correctness of the out-of-core algorithm

To validate the accuracy and efficiency of the HoMoM, the monostatic analysis of the NASA almond is considered. The parametric equations that define the geometry of the NASA almond are well known and available in the literature [16]. The Non-Uniform Rationale B-Spline (NURBS) model is shown in Fig. 7. The comparison between the computed result and the measurement for 9.92 GHz is shown in Fig. 8. The results agree with each other very well.

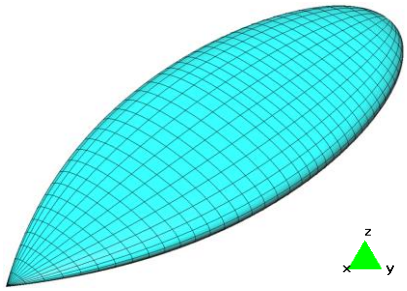


Fig. 7. NASA almond.

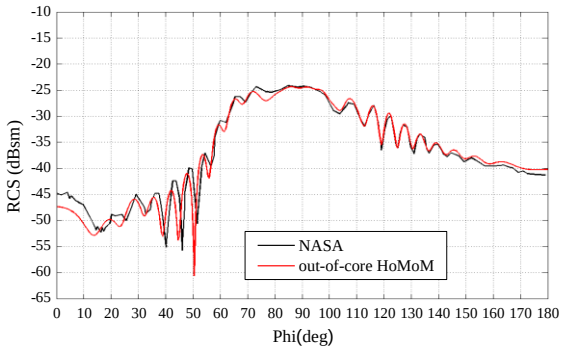


Fig. 8. VV polarized monostatic RCS of NASA almond at 9.92 GHz (0° starts from the *x*-axis in the *xoy*-plane).

B. IASIZE optimization

To find optimum IASIZE of the out-of-core algorithm using SSD and SAS, we calculate the radiation pattern of an airborne antenna array operating at 440 MHz as shown in Fig. 9. The antenna is a microstrip patch antenna array with 333 units and the material parameters of the substrate are ε_r = 4.2 and μ_r = 1.0. The dimensions of the airplane and the array are 54.0 m×53.8 m×10.5 m and 10.0 m×2.5 m×0.018 m, respectively. The number of unknowns (NUN) of the airborne array is 308,371. In all the simulation, the number of computing nodes used is 8, namely 192 CPU cores. The result is given in Fig. 10.

The time using different IASIZE is listed in Table 1. The measured wall time as a function of IASIZE is plotted in Fig. 11 and Fig. 12. From comparison, we can

see that the total time and matrix solving time of the out-of-core algorithm using SSD is evidently reduced compared with that using SAS, and the optimum IASIZE of the out-of-core algorithm using SSD and SAS are 3.0 GB and 2.3 GB, respectively.

Note that the optimum value of the IASIZE will vary slightly for the same platform when using different basis functions, or when running different projects with the same basis function. Also, the optimum value of the IASIZE may not be the same for different computational platforms. It is always advisable for the user to choose the proper IASIZE that 80%~90% of available memory size to avoid using virtual memory of the computer.

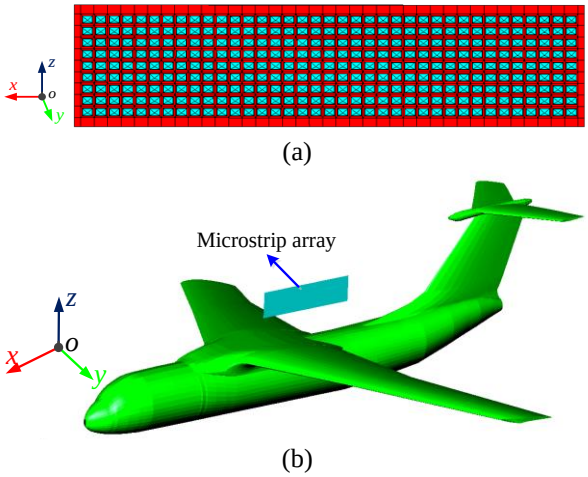


Fig. 9. Airborne array model: (a) the rectangular patch antenna array with 37×9 elements, and (b) the antenna array above an airplane.

Table 1: Wall times when using different IASIZE

IASIZE (GB)	Total Time(s)		LU Time(s)	
	SSD	SAS	SSD	SAS
2.0	31538	35360	25788	29528
2.1	30811	34194	25841	29086
2.2	30920	34047	25938	28960
2.3	31289	33524	26319	28488
2.4	31342	33846	26177	28806
2.5	30254	33961	26068	29705
2.6	30210	35717	25986	30803
2.7	30121	34507	25868	30228
2.8	30272	37172	26000	32715
2.9	30048	37123	25763	32460
3.0	30008	38894	25734	33062
3.1	30414	39344	26100	34177

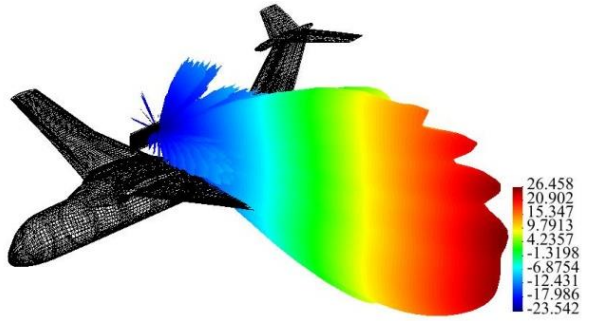


Fig. 10. 3D radiation pattern of the airborne antenna array.

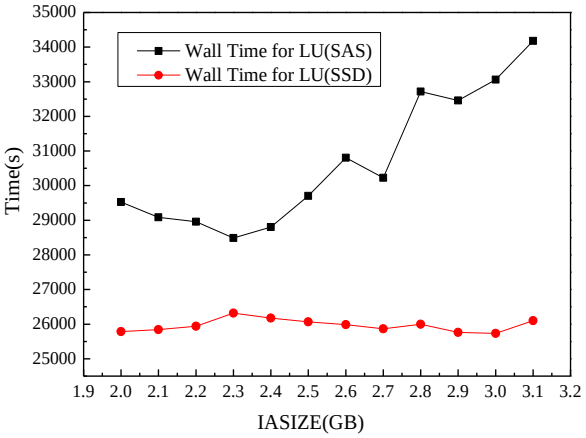


Fig. 11. Wall time for matrix solving when using different IASIZE.

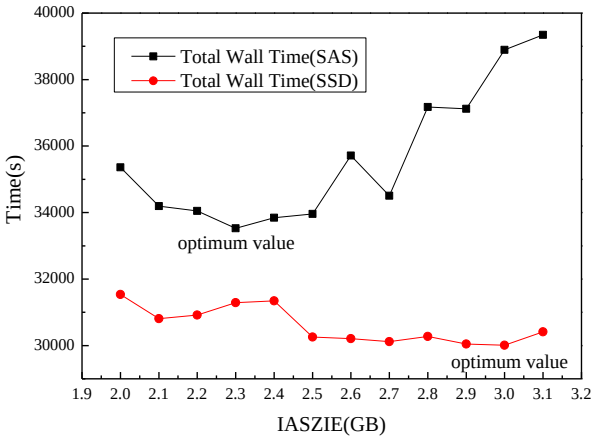


Fig. 12. Total wall time when using different IASIZE.

C. Performance comparison of SSD, SAS and RAM

In this section, we use the optimum IASIZE obtained in previous section to ensure best performance. We present one set of electromagnetic scattering results for demonstrating the performance of the out-of-core algorithm using SSD and SAS. The results of the in-core

algorithm are also given for comparison. The airplane model with dimensions of 11.6 m×7.0 m×2.93 m is shown in Fig. 13. The range of simulation frequency is from 600 MHz to 2.3GHz and the corresponding numbers of unknowns are given in Table 2, where the time for matrix filling and matrix equation solving is listed. The results in Fig. 14 illustrate that the out-of-core algorithm does not result in loss of numerical accuracy of the MoM.

According to Table 2, the performance comparison of the three solving approaches is evaluated and shown in Fig. 15. For the process of matrix filling, it can be seen that the in-core solver is the fastest, followed by the out-of-core solver using SSD, and out-of-core solver using SAS is the slowest. With the number of unknowns increasing, the difference among three approaches is more and more obvious. From the process of matrix equation solving, it can be seen that the performance of out-of-core algorithm using SSD is obviously better than using SAS and very close to the in-core algorithm. Because the matrix equation solving time is much longer than the matrix filling time, it can be concluded that the out-of-core algorithm using SSD achieves nearly the same performance as the in-core algorithm.

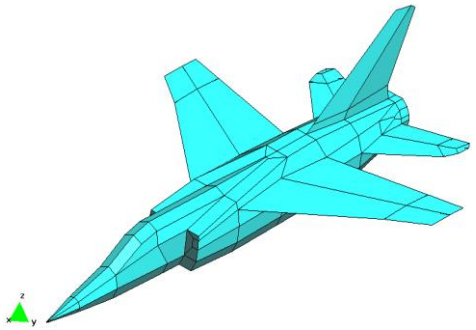
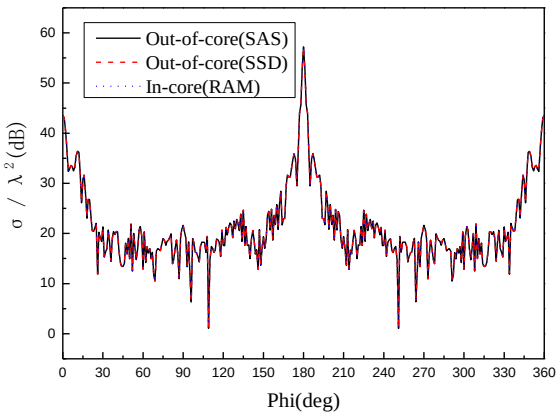
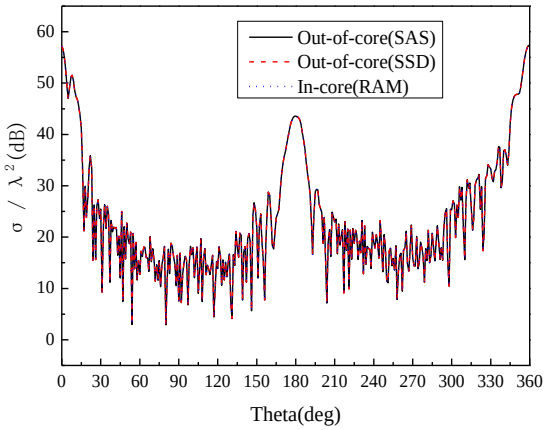


Fig. 13. An airplane model.



(a)

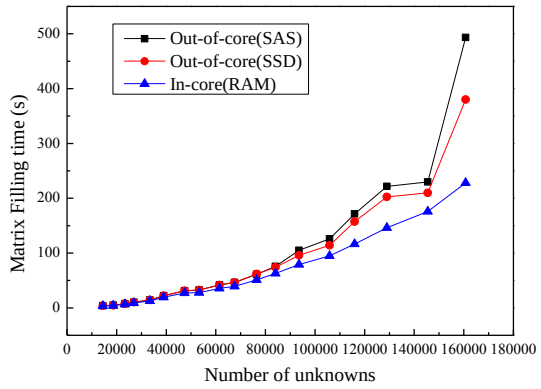


(b)

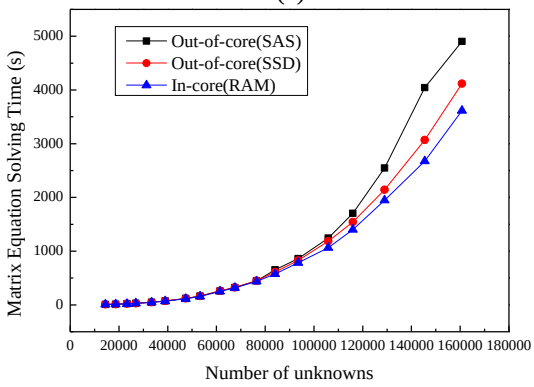
Fig. 14. Bistatic RCS of the airplane at 2 GHz: (a) *xoy* plane and (b) *xoz* plane. The plane wave is incident along –*x* axis.

Table 2: Simulation time for the airplane

NUN	Matrix Filling Time(s)			Matrix Equation Solving Time(s)		
	SSD	SAS	RAM	SSD	SAS	RAM
14482	4.07	4.06	3.66	7.80	7.88	7.45
18689	4.76	4.79	4.16	13.27	13.90	12.75
23293	7.53	7.55	6.66	20.88	21.01	20.39
26943	10.38	10.33	9.11	29.50	29.39	28.47
33415	14.32	14.39	12.60	50.85	49.59	47.41
38964	22.14	22.08	19.35	73.26	72.63	69.28
47411	31.10	31.13	27.35	121.7	120	114.7
53307	32.63	32.55	27.93	165.1	163.4	156.6
61515	41.82	42.09	35.57	262.0	258.1	251.8
67552	46.35	46.53	39.35	325.2	321.9	320.4
76459	62.05	61.28	51.17	450.7	445.6	435.8
84059	74.00	75.69	62.88	610.8	651.3	575.7
93509	95.82	105.2	79.08	828.5	861	783.8
105905	114.1	126.1	94.89	1187	1243	1064
115934	157.2	171.6	116.4	1542	1703	1400
129012	202.6	221.6	146.3	2144	2548	1949
145483	209.7	229.7	175.8	3071	4042	2677
160770	380.1	493.4	228.2	4119	4903	3615



(a)



(b)

Fig. 15. Performance comparison of the out-of-core algorithm using SSD and SAS and the in-core algorithm: (a) matrix filling time, and (b) matrix equation solving time.

D. Radiation calculation of a waveguide slot antenna

In order to validate the stability of the parallel out-of-core algorithm and further confirm the acceleration performance of SSD, the radiation problem of a waveguide slot antenna operating at 35GHz as shown in Fig. 16 is calculated. The antenna model is composed of 24 waveguide components with a total 2068 slots, for an overall dimension of 66λ by 17λ. The total number of unknowns for this antenna is 660,779, which requires approximately 6.3 TB of storage to analyze the problem using double precision arithmetic. The general computer is unable to provide enough memory, so we use the parallel out-of-core MoM with higher-order basis functions.

The computational parameters are listed in Table 3. We can see that the required memory of this simulation is about 6.3 TB and the provided memory of the CPUs is 2.0 TB, so the parallel out-of-core algorithm has broken the limitation of the memory of the CPUs. Compared with the out-of-core algorithm using SAS hard drives, the acceleration percentage of the out-of-core algorithm using SSD is about 18.66%. The overall speedup does not seem that good, but the hard disk only affects the time of reading and writing files in the out-of-core algorithm, so it is in line with expectations. Actually, acceleration of 18.66% can also reduce a lot of computing time for very large problem. The SSD is used to accelerate out-of-core algorithm in here is mainly want to reduce the performance loss that caused by reading/writing hard disk compared with the in-core algorithm. The result is given in Fig. 17. This simulation demonstrates that the code is stable and can be used for even more unknowns as long as sufficient hard disk is available.

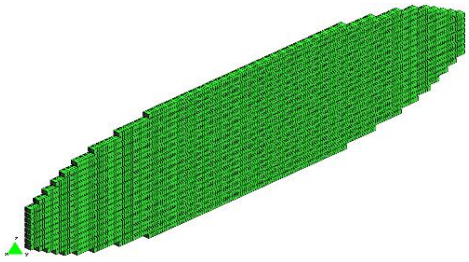
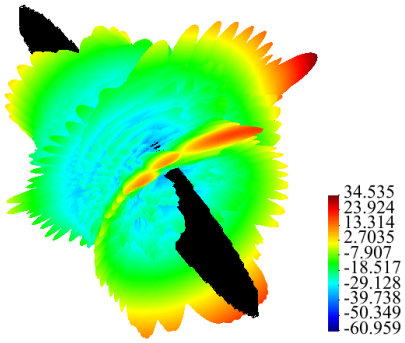
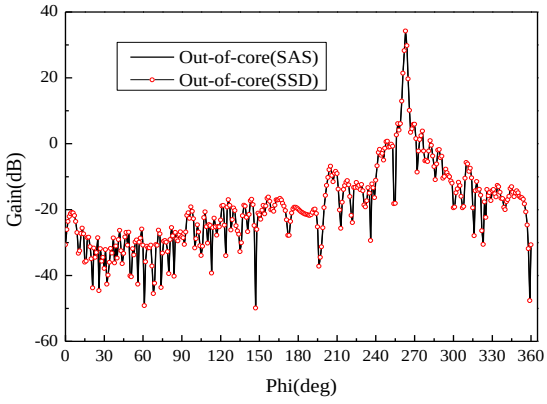


Fig. 16. Waveguide slot antenna model.



(a)



(b)

Fig. 17. Radiation patterns of the waveguide slot antenna: (a) 3D radiation pattern, and (b) xoy plane radiation pattern.

Table 3: Computational parameters for the antenna

NUN	Storage (TB)	Computational Resources	Total Wall Time(s)
660,779	6.354	32 nodes with SSD	387,319
		32 nodes with SAS	476,156

V. CONCLUSION

The performance of the parallel out-of-core MoM is improved through SSD, and the sensitivity of the speedup on IASIZE is much less with the SSD than with the SAS disks. The proposed algorithm allows matrices to be written to the hard disk and is no longer limited to the size of the memory. The algorithm does not suffer from slow-convergence issue, and thus it is suitable for accurately modeling large electromagnetic problems including complex structures and diverse materials.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0202102, in part by the China Postdoctoral Science Foundation funded project under Grant 2017M613068, in part by the National Key Research and Development Program of China under Grant 2016YFE0121600, in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2014AA01A302, in part by the Key Research and Development Program of Shandong Province under Grant 2015GGX101028, and in part by the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) under Grant No.U1501501.

REFERENCES

[1] R. F. Harrington, *Field Computation by Moment Methods*. in IEEE Series on Electromagnetic Waves. New York: IEEE, 1993.

[2] Y. Zhang, Z. Lin, X. Zhao, and T. K. Sarkar, "Performance of a massively parallel higher-order method of moments code using thousands of CPUs and its applications," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 12, pp. 6317-6324, 2014.

[3] Z. Lin, Y. Zhang, S. Jiang, X. Zhao, and J. Mo, "Simulation of airborne antenna array layout problems using parallel higher-order MoM," *International Journal of Antennas and Propagation*, vol. 2014, Article ID 985367, 11 pages, 2014.

[4] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Transactions on Antenna and Propagation*, vol. 53, no. 8, pp. 2719-2727, 2005.

[5] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns, [J], IEEE Transactions on Antennas & Propagation, 56(8):2306-2313, 2008.

[6] J. No, S.-S. Park, and C.-S. Lim, "ReHypar: A recursive hybrid chunk partitioning method using NAND-flash memory SSD," *The Scientific World Journal*, vol. 2014, Article ID 658161, 9 pages, 2014.

[7] J. Liu, Y. Lan, J. Liang, Q. Cheng, C.-C. Hung, C. Yin, and J. Sun, "An efficient schema for cloud systems based on SSD cache technology," *Mathematical Problems in Engineering*, vol. 2013, Article ID 109781, 9 pages, 2013.

[8] P. Ylä-Oijala, M. Taskinen, and S. Järvenpää, "Analysis of surface integral equations in electromagnetic scattering and radiation problems," *Engineering Analysis with Boundary Elements*, vol. 32, no. 3, pp.196-209, 2008.

[9] Y. Zhang and T. K. Sarkar, *Parallel Solution of Integral Equation Based EM Problems in the Frequency Domain*. Hoboken, NJ: Wiley, 2009.

[10] R. F. Harrington, "Boundary integral formulations for homogenous material bodies," *Journal of Electromagnetic Waves and Applications*, vol. 3, no. 1, pp. 1-15, 1989.

[11] John L. Volakis and Kubilay Sertel, *Integral Equation Methods for Electromagnetics*. Raleigh, NC: SciTech Pub., 2012.

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

[13] Y. Kuo, S. W. Tian, Z. Z. Ying, and T. M. Song, "Electromagnetic analysis for inhomogeneous interconnect and packaging structures based on volume-surface integral equations," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, no. 8, pp. 1364-1371, 2013.

[14] J. J. Dongarra, S. Hammarling, and D. W. Walker, "Key concepts for parallel out-of-core LU factorization," *Parallel Computing*, vol. 23, pp. 49-70, 1997.

[15] http://www.paratera.com/

[16] A. Woo, H. Wang, M. Schuh, and M. Sanders, "EM programmer's notebook-benchmark radar targets for the validation of computational electromagnetics programs," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 1, pp. 84-89, Feb. 1993.