Weak Scaling Analysis for the Parallel MLFMA: The Quest for Ever Larger Simulations

Bart Michiels[1], Jan Fostier[1], Ignace Bogaert[1], Daniël De Zutter[1]
[1]Department of Information Technology (INTEC), Ghent University, Belgium

***Abstract*—In the past decade, several data distribution strategies for the parallel, distributed-memory Multilevel Fast Multipole Algorithm (MLFMA) have been introduced. The common goal is to distribute the computations in the MLFMA uniformly over the different parallel processes, while minimizing dependencies and data communication between them. However, as clusters with thousands of CPU cores are becoming increasingly available, the asymptotic behavior of algorithms for a very large number of processes P and problem size N (weak scaling analysis) starts to play a dominant role. Parallel MLFMA implementations based on a hierarchical distribution of boxes and radiation pattern sampling points that exhibit excellent weak scaling behavior appear to be attractive candidates to tackle even larger problems on future hardware clusters.**

***Index Terms*—parallel MLFMA, weak scaling analysis**

I. Introduction

The Multilevel Fast Multipole Algorithm (MLFMA) is a fast matrix-vector multiplication scheme, used during the iterative solution of a Method of Moments (MoM) discretization of large boundary integral equation problems. Its computational complexity is $\mathcal{O}(N \log N)$ [1], with N the number of unknowns. State-of-the-art sequential MLFMA implementations can handle over a million of unknowns on a single workstation. However, several real-life, electrically large geometries require discretization into hundreds of millions, if not billions of unknowns. In order to handle those effectively, parallel, distributed-memory implementations are developed in order to take advantage of the aggregated memory and compute capacity of clusters.

In [2], the weak scaling behavior of a number of existing partitioning schemes was examined. Weak scaling implies that larger problems can be solved on proportionally larger computational clusters without loss of efficiency. In a weak scaling setup, the number of processes P scales linearly with N, i.e., $P = \mathcal{O}(N)$ and hence, the number of unknowns per process is kept fixed. Given a sequential complexity for the MLFMA of $\mathcal{O}(N \log N)$, this implies that the complexity per node should not exceed $\mathcal{O}(1)$ per level, or hence, $\mathcal{O}(\log N)$ in total.

None of the existing schemes for the parallel MLFMA (spatial partitioning (SP), hybrid partitioning (HyP) [3] and hierarchical partitioning (S-HiP) [4], [5]) were found to exhibit good weak scaling behavior. In all cases, the worst-case computational complexity per node and per level exceeds $\mathcal{O}(1)$ (see entries 1-3 of Table I). As a consequence, when considering larger problems on larger clusters, the load imbalance between processes will grow, leaving many processes idle while only a few of them are performing useful calculations.

TABLE I
Overview of the different partitioning strategies and the worst-case computational complexity for a node per level.

partitioning strategy	worst-case complexity per node and per level
spatial (SP)	$\mathcal{O}(N)$
hybrid (HyP)	$\mathcal{O}(\sqrt{N})$
hierarchical - strip (S-HiP)	$\mathcal{O}(\sqrt{N})$
hierarchical - block (B-HiP)	$\mathcal{O}(1)$

The major difficulty in the parallelization of the MLFMA is the fact that the data structures and the relative contribution from different types of computations differ from level to level in the MLFMA tree. At the top levels, there are only few boxes, each box containing a large number of radiation pattern sampling points, whereas at the lower levels, there are many boxes, each box containing much smaller radiation patterns. Existing data distribution schemes (SP, HyP and S-HiP) fail to distribute the load evenly on all levels (for sufficiently high N and P). For the SP, the load imbalance becomes already apparent for a very modest number of parallel processes (e.g. at 16 processes). For the more advanced S-HiP scheme, the load imbalance only manifests itself at a very high number of parallel processes and very large problem sizes.

We proposed a modification of the hierarchical scheme, in which the sampling points on the sphere are uniformly sampled in elevation θ and azimuth ϕ [6], and distributed in both θ and ϕ [2]. This scheme, denoted by B-HiP (see last entry of Table I), was found to exhibit excellent weak scaling behavior (i.e., worst-case complexity of $\mathcal{O}(1)$ per node and per level [7]), at the cost of being significantly more complex to implement.

Nowadays, computational clusters with thousands of CPU are available, enabling the solution of problems with hundreds of millions or even billions of unknowns. For such extremely large-scale problems and high number of parallel processes, the load imbalance in the SP, HyP and even the S-HiP scheme are becoming apparent. In the next section, we analyze such large-scale problem and indicate where the bottlenecks emerge. At the time of conference, the accurate solution of a problem involving billions of unknowns using thousands of CPU cores will be presented.

II. Numerical Example

We examine the behavior of all three previously existing partitioning strategies (SP, HyP and S-HiP) for a large, but nowadays realistic problem size N and a high, but again realistic number of parallel processes P. For each of the schemes, the load imbalance that causes bad performance is pinpointed. We emphasize that these bottlenecks will become even more pronounced when N and P are increased even further, as can be expected with future hardware and problems.

Consider a perfectly electrically conducting (PEC) cube geometry with an edge size of 1638λ. Using a $\lambda/10$ discretization, this problem contains slightly over 1.2 billion of unknowns. Table II shows, for the different levels of the MLFMA tree, the sampling rate of the radiation patterns and the number of boxes at the level of tree. Assume that we wish to solve this problem using 4096 parallel processes. This results in roughly 300 000 unknowns per process.

If this problem were to be solved using the spatial partitioning (SP) technique, then the boxes on all levels would have to be partitioned evenly among all parallel processes. Clearly, this is only possible for the lowest levels. For levels 6 and higher, the number of boxes is even lower than the number of parallel processes. At these levels, the load balancing will be particularly problematic as there are fewer boxes than processes, leaving many processes idling. At level 9, only 56 out of 4096 processes would be active. However, the amount of work at this level is roughly the same as on all other levels. This explains why the SP achieves only very limited efficiency. The same reasoning can be applied to the hybrid (HyP) scheme at the transition level (i.e., the highest level that is still using spatial partitioning, typically the middle level).

If the hierarchical partitioning technique with a one-dimensional partitioning of radiation patterns (S-HiP) were to be used (e.g. only in the θ or ϕ direction), then at level 9, the 2871 sampling directions along θ or the 5744 sampling directions along ϕ would have to be distributed uniformly among the 4096 processes. Again, it is impossible to obtain a uniform partitioning, however, the amount of load imbalance is clearly less severe than in the SP or HyP case. Nevertheless, the number of sampling points in either θ or ϕ direction at a top level only grows as $\mathcal{O}(\sqrt{N})$ whereas the number of parallel processes grows as $P = \mathcal{O}(N)$. This means that the load imbalance will effectively become more severe for higher N and P.

This analysis also explains why smaller simulations (tens or hundreds of millions of unknowns) using a (for current standards) moderate number of parallel processes (e.g. 128) can effectively be performed with high efficiency using the HyP or S-HiP approach. Whereas the breakdown for the SP technique becomes apparent already at 16 parallel processes, the breakdown for the S-HiP scheme appears only when the number of processes exceeds the number of sampling points in the θ or ϕ direction. This occurs only when considering the very largest problems with over thousands of parallel processes.

TABLE II
Sampling rate and number of boxes per level of the MLFMA tree for a cube geometry with a size of $1638\ \lambda$.

MLFMA level	box size	radiation pattern sampling rate ($\theta \times \phi$)	number of boxes
0	0.5λ	17×36	1 6098 266
1	1λ	26×54	4 024 568
2	2λ	40×82	1 003 688
3	4λ	66×134	249 698
4	8λ	115×232	62 426
5	16λ	209×420	15 608
6	32λ	392×786	3 752
7	64λ	751×1504	866
8	128λ	1462×2924	218
9	256λ	2873×5748	56
10	512λ	5683×11368	8
11	1024λ	11286×22574	1

III. Conclusion

We reviewed several schemes (SP, HyP, S-HiP, B-HiP) for the distributed-memory parallel MLFMA, and reviewed their weak scaling behaviour by asymptotic analysis and through a numerical example for finite N and P. Only the B-HiP scheme exhibits good weak scaling behaviour. The B-HiP scheme attains a per node complexity of $\mathcal{O}(\log N)$ when using $P = \mathcal{O}(N)$ parallel processes. As future clusters will likely incorporate more and more parallelism, the B-HiP scheme appears to be the most attractive approach to handling even larger problems.

Acknowledgment

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by Ghent University, the Hercules Foundation and the Flemish Government department EWI. The work of B. Michiels was supported by a doctoral grant from the Special Research Fund (BOF) at Ghent University. The work of I. Bogaert was supported by a post-doctoral grant from Research Foundation-Flanders (FWO-Vlaanderen).

References

[1] W.C. Chew, J. Lin, E. Michielssen and J. Song, *Fast and Efficient algorithms in Computational Electromagnetics*. Boston: Artech House, 2001.

[2] B. Michiels, J. Fostier, I. Bogaert, and D. De Zutter, "Weak Scalability Analysis of the Distributed-Memory MLFMA," accepted for publication in *IEEE Transactions on Antennas and Propagation*, 2013.

[3] S. Velamparambil and W.C. Chew, "Analysis and Performance of a Distributed Memory Multilevel Fast Multipole Algorithm", IEEE Transactions on Antennas and Propagation, vol. 53, no. 8, pp. 2719–2727, 2005.

[4] Ö. Ergül and L. Gürel, "A Hierarchical Partitioning Strategy for an Efficient Parallelization of the MultiLevel Fast Multipole Algorithm", IEEE Tra nsactions on Antennas and Propagation, vol. 57, no. 6, pp. 1740–1750, June 2009.

[5] Ö. Ergül, and L. Gürel, "Hierarchical Parallelisation Strategy for Multilevel Fast Multipole Algorithm in Computational Electromagnetics," *Electronics Letters*, vol. 44, no. 1, pp. 3–4, Jan. 2008.

[6] J. Sarvas, "Performing Interpolation and Anterpolation entirely by Fast Fourier Transform in the 3-D Multilevel Fast Multipole Algorithm", SIAM Journal on Numerical Analysis, 41(6):2180–2196, 2003.

[7] J. Fostier, and F. Olyslager, "Provably Scalable Parallel Multilevel Fast Multipole Algorithm," *Electronics Letters*, vol. 44, no. 19, pp. 1111–1112, Sept. 2008.