PARALLELIZATION OF THE MULTILEVEL FAST MULTIPOLE ALGORITHM (MLFMA) ON HETEROGENEOUS CPU-GPU ARCHITECTURES

Allocation: Exploratory/62.5 Knh
PI: Levent Gürel[1]
Co-PI: Wen-mei Hwu[1]
Collaborators: Weng Cho Chew[1], Narayan Aluru[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

The aim of this allocation was to solve unprecedentedly large scattering problems requiring large amounts of memory and to utilize a multitude of GPUs (graphics processing units) available in Blue Waters. In the scope of this project, we solved a scattering problem with 1.5 billion unknowns on 512 XE nodes; this was beyond what we could achieve before using Blue Waters. Our efficient parallelization strategy [1,2] allowed us to spread problems among large numbers of nodes to access more memory, which is required by large problems. Additionally, we achieved the largest full-wave inverse-scattering solutions in near-real time on 256 XK nodes, and it was 4.34 times faster than the same number of XE nodes [3]. This was the first GPU implementation for nonlinear inverse solutions in the literature. We implemented an efficient multi-GPU MLFMA (multilevel fast multipole algorithm) to serve as the kernel of an inverse-scattering solver.

RESEARCH CHALLENGE

We have been working on solutions of extremely large problems that are derived mainly from electromagnetics, acoustics, and optics, to name a few. An electromagnetic scattering problem may become very large as either the frequency increases or the target size gets larger. Simulating the scattering of high-frequency radar waves from a large aircraft, for instance, may require the solution of millions or even billions of unknowns. Similarly, medical imaging with microwaves requires solutions of thousands of large problems. As the problem size (i.e., the number of unknowns) grows, memory requirement increases so we need more nodes, providing access to more memory. MLFMA operates on a multilevel tree structure and is mathematically involved; therefore, it is difficult to implement the parallel algorithm on multiple CPUs. Moreover, it is not easier to implement MLFMA on GPUs because the algorithm is memory-bound due to its O(N) computational complexity. The challenge is to develop efficient parallelization strategies to spread problems evenly among large number of nodes to obtain larger solutions, and to exploit the hierarchical memory architecture of GPUs to obtain large computational throughput.

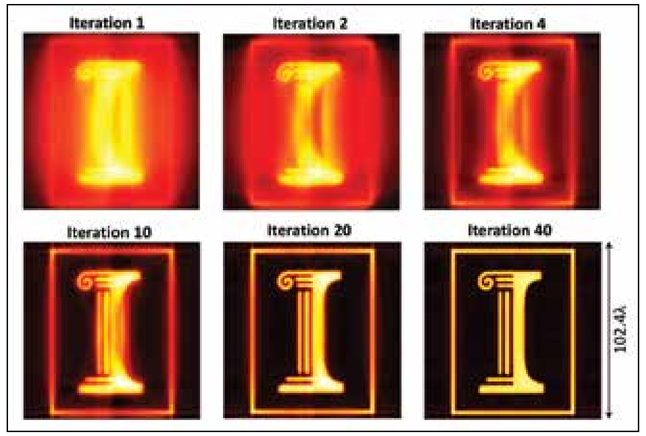

Figure 1: Convergence of the iterative inverse-scattering algorithm. The image details can be well seen after 40 iterations. It takes about two minutes to solve 38,400 forward-scattering problems with MLFMA on 256 XK nodes, and this is 4.34 times faster than employing the same number of XE nodes [5].

METHODS & CODES

The memory requirement of large problems is a challenge. Prior to Blue Waters, we reached a point where both the available total memory and the single-node memory became insufficient. Therefore, we experimented with out-of-core methods to use the disk storage as additional memory, even though we had to endure a huge penalty in the time required to solve problems [4]. The huge number of computing nodes on Blue Waters makes available a much larger total memory, thus enabling the solution of such enormous problems that were impossible for us to solve before. Additionally, improving parallelization and using more nodes led to much shorter solution times. The hierarchical parallelization strategy allows us to partition the MLFMA data structures among 512 XE nodes efficiently and evenly (i.e., in a load-balanced way). A scattering problem involving dense linear systems with 1.5 billion unknowns is solved within the 32 TB (terabytes) of memory provided by 512 XE nodes.

Unfortunately, MLFMA is a memory-bound algorithm. This is mainly due to the fast nature of MLFMA, where its O(N) complexity prevents data reuse on GPUs [5]. To exploit the hierarchical memory architecture of GPUs, we reformulate the MLFMA operations as matrix-matrix multiplications; this provides good data reuse and high computational throughput. To implement matrix-matrix multiplications efficiently, we use a hybrid shared-memory and register tiling algorithm with thread-coarsening methods.

For multi-GPU parallelization, the MLFMA tree structure is partitioned among GPUs and the operations among the branches are categorized as intra-GPU and inter-GPU. The inter-GPU data are first transferred to CPUs (central processing units), and then MPI (message passing interface) communications are performed to swap the data among computing nodes, and finally the data are transferred back into GPUs. The MPI communications are optimized to eliminate redundancy and to minimize the communication time. To reduce idle time of CPUs and GPUs, we overlap both MPI communications and CPU–GPU transfers with GPU computations. To do that, we propose a different order than the standard one such that the data-transfer time is completely overlapped with GPU computations. This provides excellent inter-node parallelization of MLFMA [6].

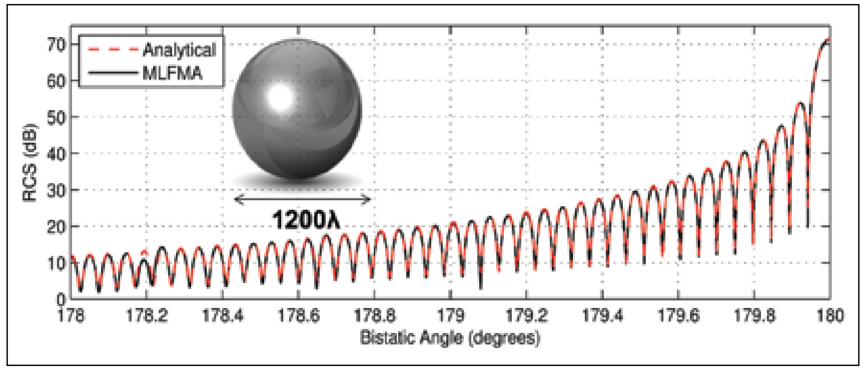

Figure 2: Radar cross-section of a conducting sphere of 1,200 wavelengths in diameter requires the solution of a 1,500,000,000 x 1,500,000,000 dense matrix equation. The computed bistatic scattering values are compared to semi-analytical Mie-series solution to demonstrate the high accuracy of the solution.

All of our codes are developed in-house with Fortran, C++, and CUDA. The PETSc framework is used in the Fortran code for iterative solutions.

RESULTS & IMPACT

We have increased the number of unknowns in the problems we can solve by achieving the solution of an electromagnetic scattering problem with 1.5 billion unknowns. The ensuing 1,500,000,000 x 1,500,000,000 dense matrix equation is solved with MLFMA and parallelized on 512 XE nodes of Blue Waters.

We obtained an efficient multi-GPU MLFMA implementation for volumetric problems for the first time within the scope of this project. A single GPU speedup is 55.14 and 3.97 times with respect to the sequential and 16-core execution baselines. The CPU executions are obtained on XE nodes whereas the GPU executions are obtained on XK nodes. This implies 3.97 times speedup on XK nodes over XE nodes. The speedup of 16 XK nodes is 846.41 and 15.34 times over sequential and a single XK node. This implies 96% parallelization efficiency among XK nodes.

Large-scale inverse-scattering solutions on GPUs are obtained on 256 XK nodes, where each node is employed with an MLFMA solver. This parallelization scheme decreases an inverse solution from 11.5 hours (sequential execution) to 7 seconds on 128 XK nodes. This provides images in near-real time, allowing our fast and massively-parallel algorithms to be employed for real-life applications. We plan to employ supercomputers (e.g., Blue Waters) for the benefit of complex imaging problems.

WHY BLUE WATERS

Our team is located at the University of Illinois at Urbana-Champaign and, therefore, it is very easy contact the Blue Waters staff. Occasionally, our Ph.D. students visit the NCSA building several times a day to solve their issues on compilers, libraries, and other Blue Waters utilities with the help of the staff. The vast amount of DRAM (dynamic random-access memory) and the number of GPUs available on Blue Waters enable scaling our science to much larger problems. Blue Waters (with its large number of CPU and GPU nodes) is ideally suited for our research. We can also make short-term (e.g., one-hour) reservations for a large number of nodes (e.g., 1,024 nodes) for large benchmark runs [7].

PUBLICATIONS AND DATA SETS

Hidayetoğlu, M., C. Pearson, W. C. Chew, L. Gürel, and W.-M. Hwu, Large Inverse-Scattering Solutions with DBIM on GPU-Enabled Supercomputers. *International Applied Computational Electromagnetics Society Symposium*, ACES, Florence, Italy, March 26–30, 2017.

Hidayetoğlu, M., et al., Large-Scale Inverse Scattering Solutions with Parallel Born-Type Fast Solvers. *Progress on Electromagnetics Research Symposium* (PIERS, Shanghai, China, August 8–11, 2016), pp. 916–920.

Hidayetoğlu M., et al., Incorporating Multiple Scattering in Imaging with Iterative Born Methods. *USNC-URSI National Radio Science Meeting*, Boulder, Colo., January 4–7, 2017.

Hidayetoğlu M., et al., Fast DBIM Solutions on Supercomputers with Frequency-Hopping for Imaging of Large and High-Contrast Objects. *Progress on Electromagnetics Research Symposium*, St. Petersburg, Russia, May22–26, 2017.

Hidayetoğlu M., W.-M. Hwu, and W. C. Chew, Performance Considerations on Various Iteration Schemes for the Distorted-Born Iterative Method. *2017 IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, San Diego, Calif., July 9–14, 2017.