Hardware Acceleration of an FMM-FFT Solver using Consumer-Grade GPUs

Malcolm J. Miranda and Tayfun Özdemir
Virtual EM Inc.
Ann Arbor, Michigan, USA
tayfun@virtualem.com

Robert J. Burkholder
Dept. of Electrical and Computer Engineering
The Ohio State University
Columbus, Ohio, USA
burkholder.1@osu.edu

***Abstract*—A surface method of moment (MoM)-based computational electromagnetics solver accelerated via the fast multipole method combined with the fast Fourier transform (FMM-FFT) is studied using different grade GPUs and multi-core CPUs as the hardware acceleration platforms. Economics related to hardware choice is studied and consumer grade GPUs are found to be very competitive against the high-end GPUs when employed in conjunction with multi-core CPUs. Though the code is equally capable of predicting radiation and scattering, this work concerns with predicting Radar Cross Section (RCS) of electrically large objects.**

I. Introduction

Accurate electromagnetic analysis of large systems is a critical need for many applications. For numerical solutions, the large electrical size and complexity of realistic systems demand an extremely high number of unknowns resulting in prohibitive memory and run time requirements. Fast integral equation solvers such as the fast multipole method (FMM) have greatly reduced the bottleneck and affordable acceleration hardware has become commonplace in the form of multi-core CPUs and GPUs. The current wave of CEM software development is concerned with adapting algorithms to these systems [1]. In this paper, a fast integral equation solver based on the FMM-FFT method [2] is applied to Radar Cross Section (RCS) computations accelerated using multi-core CPUs and consumer-grade GPUs. While high-end professional GPUs can achieve high performance for many engineering problems, consumer-grade GPUs can achieve similar run times at a fraction of the cost. The algorithm is briefly described in Section II and validation results are presented in Section III. Lastly, Section IV presents the hardware and Section V shows the acceleration results.

II. Computational Electromagnetics Model: FMM-FFT

The Computational Electromagnetics (CEM) model is based on the combined source integral equation (CSIE) discretized with Rao-Wilton-Glisson (RWG) triangular rooftop basis functions over a facetized CAD representation of the object's surface. The CSIEs are solved iteratively using the BiCGStab or GMRES algorithm, accelerated using the FMM-FFT approach to compute the matrix-vector products. The FMM-FFT is based on a single-level FMM grouping strategy arranged on a regular grid such that the group-to-group translations are Toeplitz. This reduces the computational cost of the FMM-FFT to O(Nlog$N$) and the storage requirement to O(N) without resorting to a multi-level grouping strategy. This single-level Toeplitz grouping strategy is much easier to parallelize and results in much better scaling than a multi-level FMM approach [2].

III. Validation

Validation against measured data was done for both antenna coupling and RCS simulation, but in this work, only the RCS acceleration results are being presented. Electrically large metallic objects of various shapes and sizes have been simulated over nine octave frequency range, and in all cases, the average difference between the predicted and the measured data was less than 1 dB.

IV. Hardware

A system based on a dual socket motherboard was constructed with two Intel Xeon E5-2687W (8 cores/16 threads) and 128GB RAM. The motherboard can support four "professional grade" GPU cards such as the NVIDIA Tesla or AMD FirePro. Instead, four AMD Radeon R9 280X 6GB cards were installed in the system (see Figure 1). These are "consumer grade" cards designed for video games but use the same computational units as the AMD FirePro cards. Currently, high-end professional cards cost about \$3300 or more each, which is more than the cost of each Xeon E5-2687W, which is \$1,900. Four FirePro W9100 cards would cost \$13,200, but four R9 280X cost about \$1,000 to \$1,320. The only practical advantage of the AMD FirePro cards is they are stable at temperatures above 90°C while the R9-series is not, but R9 280X cards can be easily kept below 75°C by installing the correct fans into the system's chassis. The AMD FirePro W8100, for \$1100, is an attractive alternative to the W9100 when cooling cannot be provided. The performance of the W9100 and W8100 cards for the FMM-FFT code is approximately the same as the R9 280X, since the bandwidth of the PCIe-bus on the motherboard is a limiting factor. Another limiting factor is the CPU, so using a powerful CPU such as the Intel Xeon greatly reduces runtimes. Using two Xeon CPUs allow each CPU to manage two GPU cards (for a total of four GPUs).

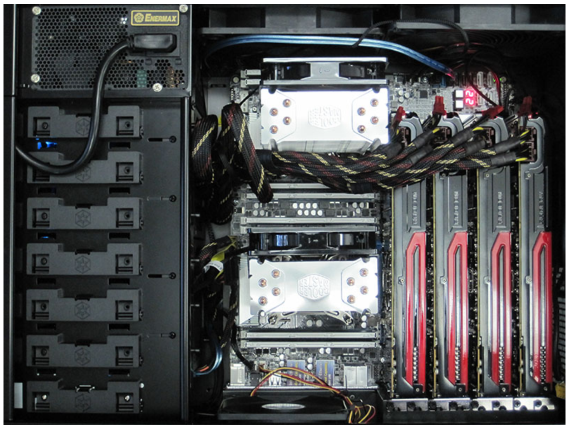

Figure 1: Dual Xeon system with four AMD R9 280X GPUs.

V. Acceleration Results

A. Multi-Core CPU Accleration

Multi-core CPU acceleration is very important for achieving low run times utilizing GPUs, since not all sections of the FMM-FFT code can be accelerated with a GPU. For example, the matrix-vector product computations involved in the BiCGStab or GMRES iterations must be performed on the CPU, since a single GPU does not have enough memory to hold all the data needed to perform the execution. The iterative solver is the most time consuming algorithm in a FMM-FFT code that handles large problems with multiple excitations. Multi-core CPU acceleration is relatively straightforward to implement for an iterative solver. Figure 2 shows the multi-core CPU speed up achieved by the Iterative Solver for 1.1 million unknowns, and Figure 3 shows actual run times for different problem sizes using the same target. Simulations were carried out on the dual Intel Xeon system described above, where each physical processor supports 16 threads.

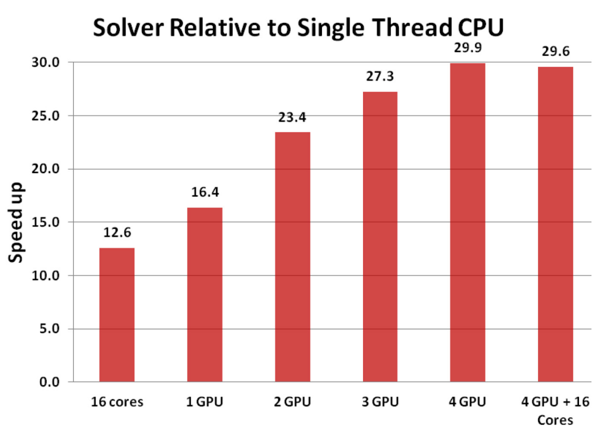

Figure 2: Solver speed up for 1.1 million unknowns.

B. GPU Acceleration

Achieving high acceleration levels with GPUs for a large problem with an iterative solver is very challenging since not all of the calculations can be performed by the GPU. Figure 2 shows the speed ups for one to four GPUs, with the CPU performing calculations that the GPUs cannot accelerate, and the last data point in the figure is simultaneous execution with all GPUs and CPU cores. The problem size of 1.1 million was chosen since it is large, but it still small enough to use one GPU. It is clear from the figure that using two GPUs gives the best incremental improvement, and this is true for both the dual Xeon system used to produce the data as well as a single CPU desktop. The advantage of using three or four GPUs is that it allows for larger problems to be accommodated. Simultaneous execution with all GPUs and CPU cores can marginally reduce performance but it helps increase the size of the largest problem that can be handled as shown in Figure 3 (where the number of unknowns increases with increasing frequency for constant sampling rate). Running the FMM-FFT code on the dual Xeon system with four R9 280X cards provides the best acceleration for three million unknowns or less. Beyond three million unknowns, the use of GPUs becomes impractical due to the data buffers becoming too large to maintain acceleration. On a good note, three million unknowns is a good size problem for many commercial applications such as modeling of a resonant antenna.

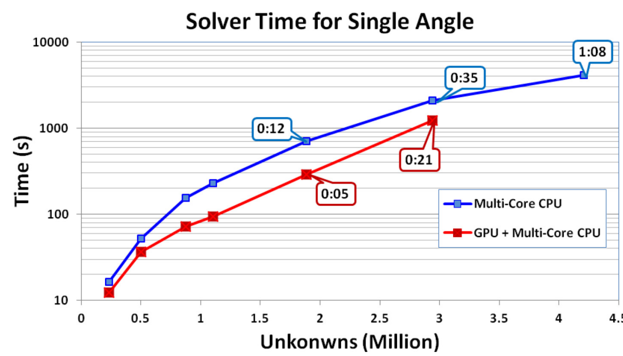

Figure 3: Single incidence angle mono-static scattering solver time for an electrically large cone-shape target.

VI. Conclusion

Consumer grade GPUs can economically accelerate the FMM-FFT method and are practical alternatives to expensive professional GPUs. Two GPUs provide the best incremental performance improvements, and adding more GPUs allows larger problems to be solved. Consumer grade GPUs can also boost the performance of a basic desktop to that of a workstation by avoiding the cost of a high-end CPU. In short, inexpensive GPUs are viable options for solving large problems when the budget is limited.

Acknowledgement

The work has been funded by the U.S. Missile Defense Agency.

References

[1] J. Guan and J.-M. Jin, "An OpenMP-CUDA Implementation of Multilevel Fast Multipole Algorithm for Electromagnetic Simulation on Multi-GPU Computing Systems," IEEE Trans. Antennas Propagation, 61(7): 3607-3616, July 2013.

[2] C. Waltz, K. Sertel, M. A. Carr, B. C. Usner and J. L. Volakis, "Massively Parallel Fast Multipole Method Solutions of Large Electromagnetic Scattering Problems," IEEE Trans. Antennas Propagation, 55(6): 1810-1816, June 2007.