

Figure of Merit for Computational Electromagnetics Solvers

Tayfun Özdemir⁽¹⁾, and Robert J. Burkholder⁽²⁾

(1) Virtual EM, Inc., Ann Arbor, Michigan, USA, www.virtualem.com

(2) The Ohio State University, Columbus, Ohio, USA, www.osu.edu

Abstract—A figure of merit (FoM) is proposed for fair comparison of various computational electromagnetics solvers. This FoM could be formulated in various fashions depending on what is being evaluated and this paper concerns itself primarily with the FE-BI/FMM-FFT type solvers accelerated on multi-core CPUs and GPUs on single-motherboard environments (or PCs). Examples are given for a particular solver and alternative FoMs are also proposed. The data and the conclusions in this paper may or may not be extended to other types of solvers or hardware and this work presented here should be viewed as the start of a much-needed discussion on the topic.

I. INTRODUCTION

The practitioners in computational electromagnetics (CEM) tend to focus on speed and accuracy, each of which is often presented alone without mentioning the other. Worse, there is almost never a mention of cost of the hardware the solver is executed on. This is done not out of malice but due to lack of appreciation for what a typical industrial practitioner considers when purchasing a software product from a vendor. From an industrial perspective, the cost of the computer system used to run the software is just as important as the speed and accuracy of the tool. One can also throw into the pile the life-time ownership cost of the hardware system as well as the cost of learning to use the tool as some may require advanced knowledge of CEM and some may not. The author of this paper has no illusions regarding the complexity of the subject and hence opted to focus on a narrower angle to be able to present numbers for the FoM of a particular solver. The motivation is to start a discussion on establishing a level playing field for comparing various computational tools and, in the process, arrive at a formula that could be readily used to make judgments.

II. EXISTING FIGURES OF MERIT

Computational community often cites GFlops/\$ or GFlops/watts or operations per joule as the efficiency of a computational hardware depending on the case. A six-core Intel Core i7 980 XE processor offers 0.2 GFlops/\$ [1] or 1 GFlops/watt [2]. Such numbers are readily available since they represent core computational power and hence, by definition, are not very useful to a practitioner such as an antenna design engineer who is often concerned with the wall-clock run time and or the cost of the PC system or the hourly rate of a cloud service. Therefore, a different definition of computational efficiency or a figure of merit must be formulated to help the practitioner when deciding which computational tool to use

and what system to purchase (in case of a PC system) or subscribe to (in case of a cloud service).

III. PROPOSED FIGURE OF MERIT

A. Formula

Formula in (1) below is proposed for evaluating different CEM solvers and establishes a level playing field. It also ensures that the most accurate and the fastest solver requiring the cheapest hardware is ranked the highest as it should. *Error* must be computed by comparing the simulated data to a measured standard antenna or a target and could be the input impedance or the gain pattern of antenna or the radar cross-section (RCS) of a target. When evaluating different solvers, all solvers must compute the same parameter of the same object. *Speed* is the wall-clock run time for completing the simulation while the *Affordability* is the inverse of the life-time ownership cost of the hardware used to carry out the computations.

FoM = (Accuracy) x (Speed) x (Affordability)

$$= \frac{1}{Error} \times \frac{1}{RunTime} \times \frac{1}{Cost} \quad [\text{dB} \times \text{Second} \times \$]^{-1} \quad (1)$$

B. Preventing Cheating

The formula in (1) was constructed primarily to prevent cheating since it establishes penalties for typical tactics. When comparing two solvers, one must fix two of the parameters and compare the third. For example, the typical practice is to fix the accuracy and compare the speed while ignoring the cost of the hardware. One may use a faster hardware (which typically costs more) to achieve faster runs for the same accuracy (and hence report “faster run times”) while the above formula could well establish that the two solvers are, in fact, equal in computational efficiency due to the cost of the hardware. Alternatively, one may opt for asymptotic solvers producing data in blazing speed but at the expense of accuracy so the resulting FoM may not be so great do to the *Error* penalty.

C. Alternative Formulas

The formula in (1) makes sense to a prospective customer in possession of a particular hardware evaluating various CEM software products before making a decision to purchase one. Though it is not realistic to expect (at least not in the near future) that the vendors publish efficiency numbers for the identical hardware, the formula makes sense if one assumes that the faster hardware cost more. If, one day, such assumptions are no longer true, a more sophisticated formula

taking into account non-trivial price-performance relations can be offered as shown in (2), where the efficiency is no longer proportional to the run time or the cost of the hardware but rather is a function of them.

$$FoM = \frac{1}{Error} \times f_1\left(\frac{1}{RunTime}\right) \times f_2\left(\frac{1}{Cost}\right) \quad (2)$$

Though the formula in (2) has the potential to allow measurement of different solvers across different algorithms and hardware, it is presented here only to acknowledge the non-trivial nature of the subject. The author believes that the use of the formula in (1) is easier to study and has a much quicker return in terms of aiding the practitioner in the short run.

For those who are turned off by the unusual units shown in (1), perhaps still another variation could be the formula in (3) which measures the solvers against a reference solver on a reference hardware with a fixed accuracy and speed producing a relative FoM:

$$FoM_r = FoM / FoM_{ref} \quad (3)$$

where the FoM_{ref} and FoM are the FoMs of the reference solver on a reference hardware and the solver in question on an arbitrary hardware, respectively, as computed in (1) and FoM_r is the (unitless) relative FoM.

IV. EXAMPLE

A. Hardware

To put some numbers down, the VirAntenn™ software [3], which has an FMM-FFT solver was benchmarked on the PC system shown in Figure 1.

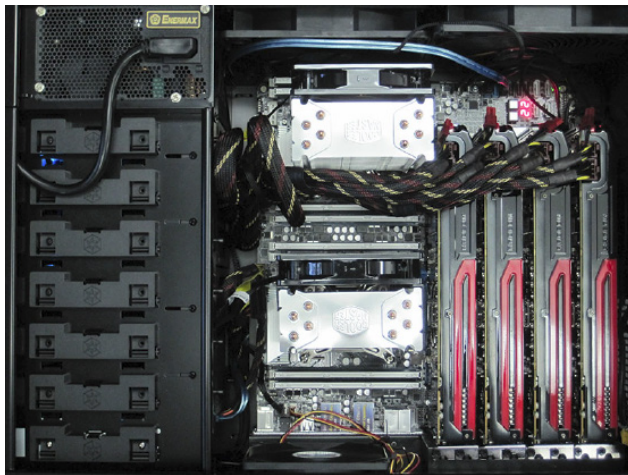


Figure 1: Dual Xeon system with four AMD R9 280X GPUs.

The system is based on a dual socket motherboard constructed with two Intel Xeon E5-2687W (8 cores/16

threads) and 128GB RAM. The system also contained four consumer grade AMD Radeon R9 280X 6GB cards Using two Xeon CPUs allows each CPU to manage two GPU cards (for a total of four GPUs). The FMM-FFT solver of the software has been optimized on the multi-core CPUs as well as the GPUs for solving large problems.

B. FoM Numbers

Figure 2 shows the FoM of the VirAntenn™ software for computing mono-static RCS of a metallic object using the formula in (1). The error is computed in dB and is the difference between the simulated and the measured data. Run time is the wall-clock time and the cost of the system (purchase price of the components and the labor cost for building it) is \$9,265.31. FoMs for two cases are given (simulation of the same target at 2GHz and 4GHz).

| Metallic Target | ERROR <i>(Average Difference from Measurement over 380 deg profile and averaged over four polarizations: VV, HV, HH, VH)</i> | RUN TIME <i>(All four polarizations, 360 Deg. Profile with 0.08 Deg. increment)</i> | COST | FoM <i>(dB × Second × \$)⁻¹</i> |
|-----------------|---|--|------------|---|
| 2GHz | 3.2dB | 13 hrs | \$9,265.31 | 7.2×10^{-10} |
| 4GHz | 4.9dB | 41 hrs | \$9,265.31 | 1.5×10^{-10} |

Figure 2: VirAntenn™ software’s performance in terms of FoM as defined in (1).

REFERENCES

- [1] www.wikipedia.com.
- [2] <http://www.pcgameshardware.com>.
- [3] <http://www.virantenn.com>.