HARD- AND SOFTWARE-BASED ACCELERATION TECHNIQUES FOR FIELD COMPUTATION

Martin Schauer[#], Peter Thoma[*]
[#] CST of America, San Mateo, CA, United States
[*] CST AG, Darmstadt, Germany

Abstract

Due to a high demand in more realistic graphics rendering for computer games and professional applications, commercial, off-the-shelf graphics processing units (GPU) increased their functionality over time. Recently special application programming interfaces (API) allow programming these devices for general purpose computing. This paper will discuss the advantages of this hardware platform for time domain simulations using the Finite-Integration-Technique (FIT). Examples will demonstrate typical accelerations over conventional central processing units (CPU).

Next to this hardware based accelerations for simulations also software based accelerations are discussed. A distributed computing scheme can be used to accelerate multiple independent simulation runs. For memory intense simulations the established Message Passing Interface (MPI) protocol enables distribution of one simulation to a compute cluster with distributed memory access. Finally, the FIT framework also allows special algorithmic improvements for the treatment of curved shapes using the perfect boundary approximation (PBA), which speeds up simulations.

INTRODUCTION

Simulation performance is a frequently discussed topic since users of simulation software want to achieve faster time-to-market in order to gain a competitive advantage. As a prerequisite for the following studies, "performance" needs to be defined first and includes the full design process from the idea to the realization.

1. Pre-processing
   - CAD Modeling / Workflow integration
   - Parameter definition
2. Solver
   - Advanced numerical algorithms
   - High performance computing
3. Post-processing
   - Derive secondary quantities
   - Optimize parameter

It is important to note that on the solver side not only the speed, but also the accuracy of this algorithm needs to be taken into account.

Performance = speed * accuracy

Most of the performance studies in this paper are discussed within the FIT framework [1]. We would also like to emphasize that the right solver choice can speed up the simulation significantly and should therefore be preferred, before hard- and software based acceleration techniques are chosen.

The paper is subdivided into two parts: hardware and software based acceleration techniques. Figure 1 illustrated how these techniques play together in [2].

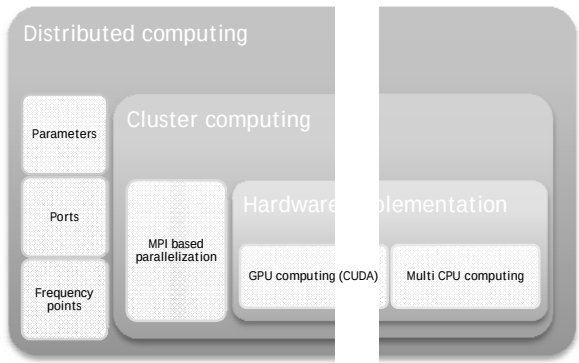

Figure 1: Overview of acceleration techniques and their hierarchy

HARDWARE ACCELERATION TECHNIQUES

Multi-CPU/multi-core is a classical hardware acceleration technique. In the recent years GPU computing emerged as a competing technique. In terms of performance we need to distinguish between algorithms with a high count of operations per memory access and algorithms with a low count.

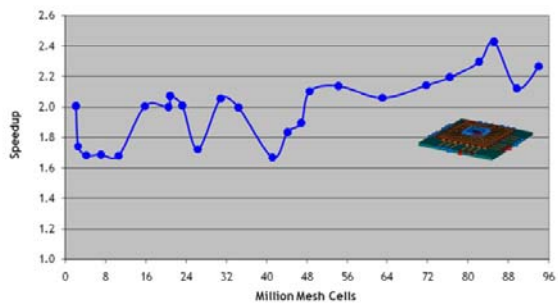

Figure 2: Simulation speed-up of the latest Intel generation CPU [3] vs. the previous generation

Figure 2 demonstrates the performance of a time domain FIT simulation using the latest hardware. This algorithm falls into the "memory-bandwidth" limited category and therefore benefits directly from recent CPU developments [3]. GPU computing even further accelerates this example [4]. Figure 3 highlights the speed-up when one, two or four GPUs are used over a reference system with the latest generation of CPUs.

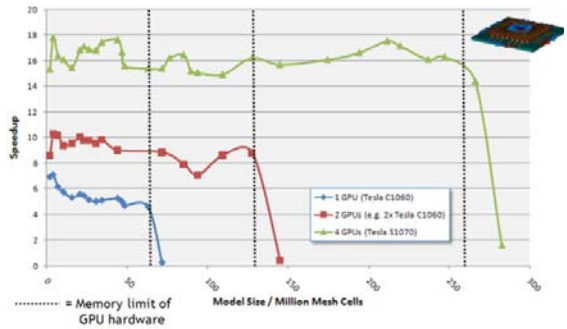

Figure 3: Speed-up of GPU computing. Reference is the new CPU generation [3].

Again, the speed-up is possible due to the higher memory bandwidth on the GPU hardware platform, which allows a greater throughput of the memory bandwidth limited algorithm.

Frequency domain techniques on the other hand require a matrix inversion within the solution process. This tends to require much more floating point operations per memory access. Therefore they benefit from multiple cores/CPUs even on systems with lower memory bandwidth.

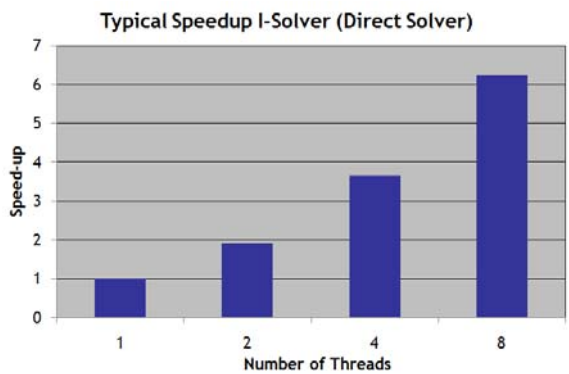

Figure 4: Typical speed-up of a direct methods-of-moments (MoM) solver vs. number of threads used on 8 core machine

Figure 4 shows the speed-up of multi core usage for a direct methods-of-moments (MoM) solver. It can be observed that an excellent speed-up of more than 6 on a 8 core machine can be achieved.

SOFTWARE ACCELERATION TECHNIQUES

The message passing interface protocol (MPI) [5] is a standard library used for parallel processing across multiple compute nodes via a standard interconnect protocol, e.g. Internet Protocol (IP). It is used in [2] to implement a Domain Decomposition scheme. Unlike the distributed computing scheme, see figure 1, MPI introduces some overhead in the simulation. In its time domain implementation the nodes need to exchange information about field components across the domain boundaries in each time step. Therefore the cluster computing becomes most efficient if the overhead - the time to synchronize the nodes - is small compared to the time to update the fields in the volume of one domain.

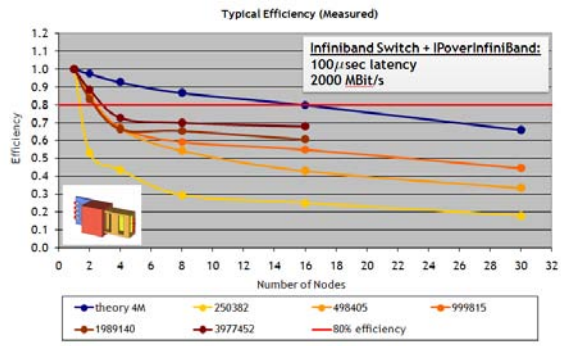

Figure 5: Efficiency of MPI based cluster computing for different discretizations vs. number of used cluster nodes

Figure 5 visualizes the efficiency of cluster computing, when multiple compute nodes are used. In addition it includes multiple discretizations in terms of mesh cell count for the same example. In theory the efficiency drops as the overhead gets larger with multiple nodes, highlighted in blue. If the model is large enough, one can get close to this limit, making it an effective scheme for simulations with distributed memory.

For independent simulations, multiple computers in the network can be used to simulate different

- Port excitations (time-domain solver)
- Frequency samples (frequency domain solver)
- Parameter combinations during parameter sweep or optimization

in parallel. Figure 6 illustrates the process of submitting a simulation from the frontend machine to the main controller. Since all the setup is done on the frontend it detects N independent simulations and asks the main controller for N individual jobs. The main controller checks the status of the connected solvers and distributes the simulations. After the solver is finished it reports back to the main controller. The frontend at the end automatically merges the results of the individual simulations, in order to make this process seamless for the user. The overall simulation performance therefore is an almost perfect linear speed-up with the number of

computing nodes for independent simulations. The fact that there is no user interaction required makes this scheme very efficient and scalable.

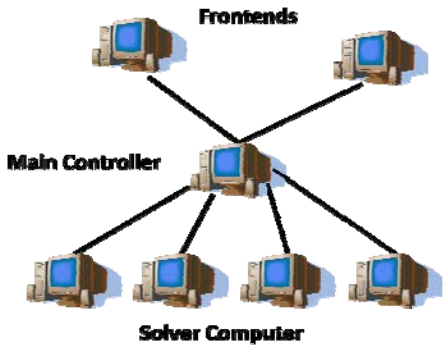

Figure 6: Computers involved and nomenclature of distributed computing

For the software/algorithmic acceleration, [2] offers an extension to FIT, the perfect boundary approximation (PBA) technique [6]. This conformal discretization method circumvents the geometry approximation error in classical finite-difference (FD) type of simulations. It allows using a much coarser mesh in order to achieve the same accuracy level and therefore speeds up the simulation, while the advantages of a structured meshing algorithm are maintained*. In addition the convergence during mesh refinement behaves more consistent, an important feature for any a posteriori error estimation.

SUMMARY

This paper introduced several hardware and software acceleration techniques, in particular for electromagnetic field computations. Figure 1 summarizes these and also illustrates how they interact with each other for maximum performance, e.g. cluster of GPU accelerated nodes.

As mentioned in the definition of performance the whole process from the idea to the final simulation design needs to be taken into account, including simulation setup and post processing. Another important factor for the performance is the right solver choice, e.g. time domain or frequency domain and mesh choice, e.g. hexahedral or tetrahedral.

Finally, the discretization method needs to be accurate (e.g. Perfect Boundary Approximation) otherwise the acceleration technique will only make a slow method less slow, but never fast.

* In [2] the user is not limited to structured hexahedral meshes, but can also use an unstructured mesh in time domain FIT simulations. This multilevel subgridding scheme also includes the PBA extension, giving the user full meshing flexibility and highest accuracy.

REFERENCES

[1] Weiland, T.: A discretization method for the solution of Maxwell's equations for six-component fields: Electronics and Communication, (AEÜ), Vol. 31, pp. 116-120, 1977.
[2] CST STUDIO SUITE™ 2009, www.cst.com
[3] Intel® Xeon® processor 5500 series, http://www.intel.com/technology/architecture-silicon/next-gen/
[4] NVIDIA® Tesla™ GPU http://www.nvidia.com/object/tesla_computing_solutions.html
[5] Message Passing Interface (MPI), http://www.mcs.anl.gov/research/projects/mpi/
[6] Krietenstein, B.; Schuhmann, R.; Thoma, P.; Weiland, T.: The Perfect Boundary Approximation technique facing the challenge of high precision field computation: Proc. of the XIX International Linear Accelerator Conference (LINAC'98), Chicago, USA, pp. 860-862, 1998.