Many-core Algorithms for High-order Finite Element Methods: When Time To Solution Matters

T. Warburton[a]

[a] ***Department of Computational and Applied Mathematics, Rice University***

The ultimate success of many modeling applications depends on time to solution. I will illustrate the critical nature of time to solution by describing a joint project between my group at Rice University and Dr David Fuentes at the MD Anderson Cancer Center. The project goal is to evaluate the role and viability of using finite element modeling as part of the treatment planning process for MR Guided Laser Induced Thermal Therapy. The success of this project will depend in great part on the ability to model individual treatments with calculations that take mere seconds.

Modern many-core processing units, including graphics processing units (GPU), presage a new era in on-chip massively parallel computing. The advent of processors with $\mathcal{O}(1000)$ floating point units (FPU) raises new issues challenging conventional measures of "optimality" of numerical methods. The ramp up in FPU counts for each new generation of GPU over the past four years has been accompanied by a slower increase in the the memory capacity of the GPU. For example, a few hundred US dollars currently buys a parallel computer that is capable of performing $\mathcal{O}(4 \cdot 10^{12})$ floating point operations per second, but only of reading $\mathcal{O}(5 \cdot 10^{10})$ values from memory per second. From the point of view of numerical analysis, this means that the traditional approach of comparing optimality of alternative numerical methods based on their floating point operation count per degree of freedom has become mostly irrelevant. Claims of optimality derived from this measure therefore need to be reevaluated and the formulation of numerical methods in general need to be revisited given the changing computational landscape.

In 2009 we demonstrated that the nodal discontinuous Galerkin time-domain method for computational electrogmagnetics can achieve a high percentage of peak performance of NVIDIA GPUs. In subsequent articles we demonstrated that it is possible to improve time-stepping efficiency of these methods using local time-stepping methods on GPUs. We also demonstrated scalability when using multiple GPUs in a workstation and on over 400 GPUs in a larger scale cluster . Finally we also created a new variant of the discontinuous Galerkin methods that enables the use of curvilinear elements to accurately model non-planar domain boundaries within the restricted memory budget of current GPUs . The most challenging aspect of this effort is not in fact the implementation but rather the analysis of the new method .

The presentation will touch on several important and inter-linked issues that impacted the development of high-order finite-element methods including spectral element and discontinuous Galerkin based solvers for a moving many-core architecture target. We will discuss on-chip scalability, multi-GPU scalability, inter-generational GPU scaling, specialization for different element types and how we modified the solver memory requirements. Finally we will discuss programming tools that we are currently developing to enhance the productivity of programmers engaged in this type of implementation task.