

Exascale Computing: Challenges and Opportunities for Applied Mathematicians and Engineers

Ravi Samtaney*

Division of Mathematical and Computer Sciences, King Abdullah University of Science & Technology, Thuwal, Kingdom of Saudi Arabia

Computational science has emerged as the third leg of the scientific enterprise alongside theory and experimentation. Since the advent of the digital computer, followed by parallel computing we have made tremendous strides in many scientific areas. However, hardware advances alone were not responsible; significant advances in algorithms also accompanied advances in hardware. As Moore's law flattens out, we expect the next advances in hardware will be in the spirit of "many cores on a chip", i.e., massive parallelism. We are soon to be in the throes of another revolution in parallel computing in which $O(10^8)$ cores or more will become the norm at a few select high-performance computing centers at first and then these systems will likely proliferate to other supercomputing facilities around the world. Welcome to the exascale!

One motivation for seeking to reach exascale (10^{18}) computing, beyond conveying the sentiment in Mallory's comment "because it's there" about the desire to climb Everest, is to overcome or mitigate the tyranny of scales so prevalent in multiphysics applications. Furthermore, one of the central tenets of exascale computing is to transform modeling and simulation in science and engineering to a genuine science-based predictive discipline. The US Department of Energy is one government entity that has paid significant attention to this upcoming revolution. It has organized several workshops over the past few years [1]. These workshops have identified several areas of science where exascale computing will push the frontiers in a transformative fashion. These science areas include climate science, materials, renewable energy, biology, socioeconomic modeling etc. Research opportunities within energy and environment science areas for example are discussed in reference [2].

Today, it is not unusual to encounter decades old numerical kernels in many multiphysics simulation codes, and many of these kernels were written first on serial computers, then ported to vector architectures and finally parallelized in an ad-hoc fashion for parallel architectures. Even when the number of processors increased to $O(10^3)$, these numerical kernels survived. This modus operandi cannot be extrapolated to the exascale. There are a number of reasons for the previous claim. Going to the exascale will require radical changes in hardware, and a thorough understanding of hardware architecture in programming applications to effectively use tens of millions of cores. The new mantra is that while physical memory is cheap, accessing memory is extremely expensive. Codes will have to be rewritten to make most effective use of cache and other architectural design features and heterogeneity (for example a mix of CPUs and GPUs). Particular attention will have to be paid to Amdahl's law.

Another important consideration is fault-tolerance so that in the event of hardware failure of a single core, the code can recover gracefully. Yet another important consideration will be "green algorithms", i.e., algorithms that pay attention to not only data movement and placement but also reducing power consumption. A new theme that has emerged is that of co-design wherein both software and hardware designs are performed in a coupled fashion to implement a particular function or

application. An example of co-design is in the area of climate modeling is discussed in reference [3].

In all associated areas of exascale computing, applied mathematicians and computational scientists can find several research opportunities. Let us consider a simple task. Suppose one would like to develop a Newton-Krylov type implicit algorithm to solve a system of partial differential equations exhibiting multiple time and spatial scales [4]. The core of a Newton-Krylov consists of a nonlinear function evaluation, which is also used to compute the Jacobian-vector product during the Krylov iterative solution step. Usually some form of preconditioning also becomes necessary. Now imagine implementing this algorithm and computing on tens of millions of cores. The nonlinear function evaluation has to pay special attention to the deep memory hierarchy, and the linear solver (which dominates the computing cost as the problem size gets larger) needs to be written in a fault-tolerant, communication minimizing and synchronization-hiding fashion [5]. Apart from coming up with clever heuristics and ad-hoc techniques to keep computational scientists busy, applied mathematics will be engaged to prove theorem about optimality, or lack thereof, and asymptotic efficiencies of proposed algorithms. Clearly is enough work here to keep a whole generation of applied mathematicians and computational scientist busy in the worthwhile endeavor of exascale computing.

References

1. The Opportunities and Challenges of Exascale Computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee. U. S. Department of Energy Report.
2. Simon H, Zacharia T, Stevens R (2007) Modeling and Simulation at the Exascale for the Energy and Environment. U. S. Department of Energy Report.
3. Wehner MF, Olier L, Shalf J, Donofrio D, Leroy A, et al. (2011) Hardware/Software codesign of global cloud system resolving models. J Adv Model Earth Syst 3:M10003.
4. Knoll DA, Keyes DE (2004) Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J Comput Phys 193:357-397.
5. Report on the Workshop on Extreme-Scale Solvers: Transition to Future Architectures, U. S. Department of Energy Report.

*Corresponding author: Ravi Samtaney, Associate Professor, Division of Mathematical and Computer Sciences, King Abdullah University of Science & Technology, Thuwal, Kingdom of Saudi Arabia, E-mail: ravi.samtaney@kaust.edu.sa

Received November 21, 2012; Accepted November 24, 2012; Published November 27, 2012

Citation: Samtaney R (2012) Exascale Computing: Challenges and Opportunities for Applied Mathematicians and Engineers. J Appl Computat Math 1:e126. doi:10.4172/2168-9679.1000e126

Copyright: © 2012 Samtaney R. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.