A discussion of advanced computing hardware and software trends and related Computational Dynamics implications

Dan Negrut*, **Radu Serban***, **Hammad Mazhar***, **Ang Li***, **Omkar Deshmukh***, **Andrew Seidl***

* University of Wisconsin-Madison
1513 University Ave., Madison, WI, 53706, USA
[negrut, serban, mazhar, ali28, odeshmukh, aaseidl]@wisc.edu

Abstract

We discuss hardware and programming techniques that have the potential to reduce execution times associated with the simulation of large and/or complex dynamical systems. These systems might comprise a large number of components interacting through friction in the presence of unilateral and bilateral constraints; model fluid-solid coupling phenomena; or handle the dynamics of flexible components. The questions of what fast computing means and how it can be achieved provide the backdrop for a discussion that touches on GPU, multi-core, and distributed (multi-node) computing. Each of these three alternatives is scrutinized from the point of view of the underlying hardware organization and of the accompanying software stack that mediates the user–hardware interaction. Two common sense observations will be emphasized [1]: (*i*) most often, chip $\leftrightarrow$ memory data movement, as dictated by the adopted algorithms and the software implementation choices, dictates how fast a simulation will run; and (*ii*) presently, the advanced computing landscape is fluid to the point where the answer to the question “what are the right architecture and software ecosystem for my application?” is problem specific and depends on subjective factors, such as the familiarity of the software designer with the underlying hardware.

The fluid nature of the advanced computing landscape stems from the observation that Moore’s Law is waning in relevance. As the scientific community transitions into the next decade, it becomes manifest that Moore’s law will fail to provide the backdrop for the steady efficiency gains in scientific computing that the community has come to expect. What is certain is that Intel has the technology to sustain a reduction in feature length to 14 nm in 2014, 10 nm in 2016, 7 nm in 2018, and 5 nm by 2020. However, the fundamental problem is that over the last five years the Dennard scaling, which translated reduced device feature lengths into increasing computational efficiency, reached its practical limits. Taking into account the interplay between feature length, voltage, and frequency, the Dennard scaling calls for a reduction of the voltage as the feature length goes down. Unfortunately, the sheer number of transistors per unit area and the current leaks at small feature length run the danger of extreme chip heating. In some sense, the chip has become the victim of its own success: the feature length has become so small that current leaks from increasingly many transistors per unit area lead to a chain reaction that can trigger thermal runaways. It comes as no surprise that new chip designs increasingly host what is called dark silicon – transistors that cannot be powered lest the chip is compromised [2].

This gloomy outlook is counteracted by two encouraging trends: there is strong momentum behind the physical integration, at the chip level, of multi-core and accelerator architectures, with early examples available in Intel Haswell, AMD Kaveri, and NVIDIA Jetson. Second, 3D memory modules, such as the one defined by the High-Bandwidth Memory (HBM) Standard, promise major increases in memory bandwidths combined with reductions in latency and power. If the typical memory bandwidth in a modern CPU is in the neighborhood of 50 GB/s, the next generation of GPUs will deliver close to 1 TB/s on a 25% power budget. These two emerging technologies; i.e., chip-level CPU-accelerator integration and new memory technologies, provide further incentive in Computational Dynamics to (1) leverage parallel computing; and (2) attack large scale multi-physics applications.

The last part of this communication focuses on Euler – a heterogeneous cluster supercomputer assembled in the Simulation-Based Engineering Lab at the University of Wisconsin-Madison. Euler, whose layout is schematically shown in Fig. 1, has been used for fluid-solid interaction problems that display fine grain parallelism and are well suited for GPU computing [3]; to simulate the dynamics of granular dynamics on multi-core CPUs and/or GPU hardware [4]; and to test the potential of sparse GPU-based linear solvers to accelerate the simulation of flexible body dynamics using implicit integration [5] in the context of the Absolute Nodal Coordinate Formulation [6]. Since July 2011, Euler has run jobs for over 230 colleagues

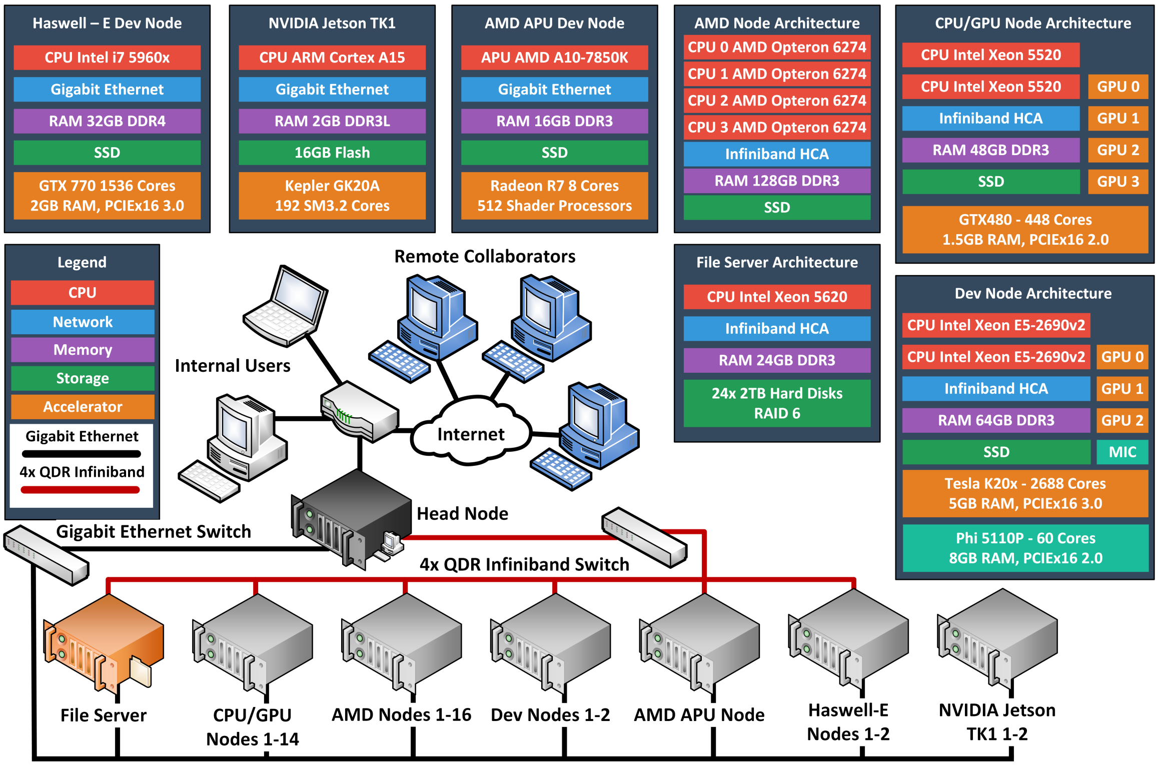

Figure 1: Current layout of Euler, a "shared-use" heterogeneous supercomputer cluster available to the broader community for research projects/collaborations in the area of Multibody Dynamics [7].

from 32 research groups. During this time, Euler executed more than 1.2 million jobs using over 6.5 million CPU hours and 660,000 GPU hours. Subject to certain regulations and constraints, this "shared-use" hardware asset is open to domestic and international researchers from other institutions interested in Multibody Dynamics research who need access to a hardware asset that includes a variety of experimental platforms for advanced computing.

References

[1] D. Negrut, R. Serban, H. Mazhar, and T. Heyn. Parallel computing in Multibody System Dynamics: Why, when and how. *Journal of Computational and Nonlinear Dynamics*, 9:041007–1, 2014.

[2] H. Esmaeilzadeh, E. Blem, R. St.Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, pages 365–376, June 2011.

[3] A. Pazouki and D. Negrut. A numerical study of the effect of particle properties on the radial distribution of suspensions in pipe flow. *Computers and Fluids (accepted)*, 2014.

[4] H. Mazhar, A. Heyn, Tasora, and D. Negrut. Using Nesterov's method to accelerate Multibody Dynamics with friction and contact. *ACM Transactions on Graphics (TOG)–under review*, 2014.

[5] R. Serban, D. Melanz, A. Li, I. Stanciulescu, P. Jayakumar, and D. Negrut. A GPU-based preconditioned Newton-Krylov solver for flexible multibody dynamics. *International Journal for Numerical Methods in Engineering*, submitted, 2014.

[6] M. Berzeri, M. Campanelli, and A. A. Shabana. Definition of the elastic forces in the finite-element absolute nodal coordinate formulation and the floating frame of reference formulation. *Multibody System Dynamics*, 5:21–54, 2001.

[7] SBEL. Euler: A CPU/GPU–Heterogeneous Cluster at the Simulation-Based Engineering Laboratory, University of Wisconsin-Madison. `http://sbel.wisc.edu/Hardware`, 2014.