The Changing Face of Scientific and Engineering Computing

By Douglass Post

I RECENTLY PARTICIPATED IN A FEDERAL AGENCY WORKSHOP TO IDENTIFY WAYS TO IMPROVE THE PRODUCTIVITY OF SCIENCE-BASED HIGH-PERFORMANCE COMPUTING (HPC) SOFTWARE APPLICATIONS. THE AGENCY IS CONCERNED THAT HPC APPLICATIONS WILL NOT BE AVAILABLE to utilize next-generation high-performance computers—in the exascale range (approximately 10^{18} floating-point operations [FLOPs] per second). The computing power now becoming available (10^{15}–10^{18} FLOPs) will give society the unprecedented capability to use HPC to solve some of the hardest technical problems facing the world today. This computing power will let us simultaneously

- utilize highly accurate solution methods,
- include all of the scientific effects we know to be important,
- validate the correctness of the models for those effects and quantify their uncertainties,
- model full-scale systems, and
- achieve reasonable problem turnaround times.

Over the next few years, the lower end of this computing power will become available to the general scientific and engineering community, not just to a handful of major research centers.

The impact of HPC is already being felt. It's enabling major advances in scientific research and engineering and bringing about a paradigm shift in research and engineering methods. To name a few, science-based HPC applications are beginning to be able to

- predict the weather with greater accuracy than before (including the unusually complex path of major storm systems such as Hurricane Sandy);
- improve automotive safety through crash simulations;
- increase the fuel efficiency and reduce the noise of new commercial aircraft (for example, the Boeing 787 versus the 777), and
- analyze data from large telescopes and satellites to identify planets orbiting other stars.

These applications, together with high-performance computers, are enabling significant advances in scientific research and engineering design. For example, theoretical chemistry is now done with large-scale HPC computer applications such as the General Atomic and Molecular Electronic Structure System (GAMESS; www.msg.ameslab.gov/gamess), Gaussian (www.gaussian.com), and NWChem (www.nwchem-sw.org). The impact of computational chemistry was recently recognized by the 2013 Nobel Prize in Chemistry. The discovery of the Higgs boson required the use of HPC to analyze large experimental datasets to conclusively identify the small number of decays of a Higgs boson out of many, many decays of other collision products.

Clearly, HPC will continue to revolutionize scientific research. But while there are many challenges for HPC, there are initiatives that are emerging for handling these challenges.

HPC Challenges

HPC isn't only computers or software applications. Successful HPC requires an ecosystem of sponsors, subject matter expert users, software applications, validation experiments and data, high-speed networks, high-performance computers, and data storage facilities. Without every single one of these, the system is crippled. Today, the weakest link is software, partly because each technical area generally requires a different software application. The study of protein folding, aircraft performance, weather forecasting, and other complex phenomena require different software applications even if they can all take advantage of the same networks,

1521-9615/13/$31.00 © 2013 IEEE

computers, and data storage facilities. There are three additional major challenges:

- efficiently exploiting next-generation massively parallel complex computer architectures with special-purpose processors;
- developing, deploying, and supporting software applications that can provide accurate solutions for complex problems easily and quickly; and
- ensuring sponsor support over the HPC ecosystem's whole life cycle.

Overcoming the "power wall" is the first challenge. For approximately the last 30 years, individual processor clock speeds have doubled every 18 months—a consequence of "Moore's law." After 2005, power dissipation limits have clamped processor clock speeds to about 2 GHz. Unable to increase the clock speed, chip manufacturers have continued to improve computer performance with multiple processors on each chip (multicore), and special-purpose processors (such as general-purpose graphics processing units, or GPGPUs). This improves performance at the cost of increased computer and programming complexity. Applications now must be capable of running efficiently on massively parallel computers with a mix of heterogeneous processors.

The second challenge is to integrate the many different scientific effects that govern the system of interest's behavior. Practical algorithms for each important effect must be developed and integrated into a single application. Successful development generally requires large, multidisciplinary, non-collocated groups from different organizations and funding sources, working together as a tightly knit team. These are all tremendously demanding scientific, coordination, and management challenges. The efforts of small and large groups of highly skilled staff must be coordinated closely. This is causing a major paradigm change in the structure and sociology of application development teams. There's a profound shift from small- and medium-sized applications developed by small groups primarily for their own research, to large-scale applications used by scientists and engineers outside the development group. These external users lack intimate knowledge of the code's strengths and weaknesses. Representative examples of such large-scale codes include the astrophysics Flash code (www.flash.uchicago.edu), the already mentioned computational chemistry code GAMESS, and the weather-prediction code Weather Research and Forecasting Model (WRF; www.wrf-model.org). Such codes require high levels of software quality, effective software development practices and processes, and agile software project management. Verification and validation, robustness, usability, documentation, problem set up, and analysis of results and user support all assume much greater importance than for small group research codes. Although there are examples of success, there are many that didn't succeed due to resource limitations, inadequate attention to the aforementioned issues, and less technical capability and fewer features than the competition.

The third challenge is due to the expense and long life cycle of software development and the entire HPC ecosystem. In the past (see Figure 1), research sponsors seldom explicitly funded the development of small-scale applications. They funded the scientific research but mostly left the researcher to scrounge the resources needed to develop the computational tool. Development of the application tool was done to facilitate the developer's research, not to build a tool for the use of others. For larger-scale applications with a large user community, the time and effort required to develop, deploy, and sustain these applications is too large to be covered as part of a few research grants. The life cycle of such codes is typically 20–30 years or longer, and involves budgets of $5 million or more per year—often much more. Even explicit code development projects, such as the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) projects have been funded for five years at most. This funding paradigm must change to provide support for the entire software application life cycle, or only a few applications that can exploit next generation computers will exist, and the full benefit of these computers will not be realized.

Welcome Aboard!

CiSE welcomes the following new editorial board member.

Jeffrey Carver is an associate professor of computer science at the University of Alabama. His expertise is in empirical software engineering, and his research focuses on the study and application of appropriate software engineering practices to the development of computational science and engineering software. Carver has a PhD in computer science from the University of Maryland, College Park. Contact him at carver@cs.ua.edu.

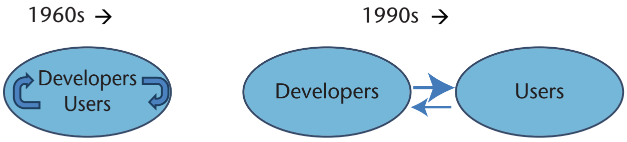

Figure 1. The software development life cycle for scientific and engineering codes is changing. Research codes are generally developed and used by the same team. General-use codes are developed by a group much smaller than the user base.

Understanding Requirements and Building Consensus

The HPC ecosystem, including the applications, performs similar functions as an experimental research, test, or design facility. Sponsors already understand what's required to design, build, operate, and sustain physical facilities. HPC ecosystems, including the software applications, have analogous requirements. As noted, the development, deployment, and sustainment of large-scale scientific software applications cost at least $5 to $10 million per year for the useful life of the application (often 30 or more years). If the application is successful in attracting a very large user community (and thus, it's really successful), the support cost can be even larger. The remaining HPC ecosystem also needs support. If adequate financial support doesn't exist, society doesn't get the maximum potential benefit. The support must also be continuous, especially software application support. Complex scientific and engineering software is a living intellectual construct. If a software application isn't continuously supported, it dies as the people working on the code scatter to other endeavors when the support withers. Those people generally don't come back if support is restored.

Engaging sponsor support requires a convincing business model. It's important to demonstrate that HPC ecosystems, including the applications, can enable research or design to be done more quickly, more efficiently, and more effectively than with conventional methods. In other words, it's important to see a tangible return on investment. In some cases, computing can enable research and testing that can't be done with physical systems (such as weather and climate forecasting; or testing many candidate design options for very large, complex systems). Scientists and engineers must aggressively "market" their ideas and vision for the use of HPC to improve research and engineering outcomes. In the past, they've concentrated on the quality of the science and let that speak for itself. But, in the future, they'll need to focus on communicating the scientific and engineering impact and the ability to reduce costs, schedule, risks, and performance shortfalls in their research and in the engineering design of new products.

Fortunately, US funding agencies and industry are becoming aware of the challenges and are moving to address them. A limited number of large-scale computing projects have been launched recently by several agencies. In 2010, the US Department of Energy recently launched the $20 million/year Consortium for Advanced Simulation of LWRs (CASL; www.casl.gov), led by the Oak Ridge National Laboratory. In 2008, the US Department of Defense (DoD) launched a similar scale project called Computational Research and Engineering Acquisition Tools and Environments (CREATE) to develop, deploy, and support physics-based computational engineering tools for the design and analysis of DoD weapons systems (www.hpcmo.hpc.mil/cms2/index.php/aboutcreate). Industry is also beginning to utilize multi-physics design tools, and there are many independent software vendor products that are beginning to provide multi-physics simulations of smaller-scale devices and systems. The workshop I mentioned at the beginning is evidence that progress is being made, and that there's a path forward for high-end scientific and engineering computing to have a bright and productive future. However, success is contingent on sponsors willing to provide the required funding for the HPC ecosystem and the time needed to develop the application software.

Douglass Post is the chief scientist of the US Department of Defense (DoD) High-Performance Computing Modernization Program (HPCMP) and an IPA from the Carnegie Mellon University Software Engineering Institute, where he's a member of the senior technical staff. He initiated and leads the DoD HPCMP Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program, a Tri-Service (Army, Navy, and Air Force) DoD program to develop and deploy physics-based HPC engineering tools for the design of ships, air vehicles, and RF antennas. Post has a PhD in physics from Stanford University. He's a Fellow of the American Nuclear Society, the American Physical Society, and IEEE. He recently received the 2011 Gold Medal from the American Society of Naval Engineers, their highest annual award. Contact him at douglass.post@hpc.mil.