

Bringing HPC to Engineering Innovation

Although there's a widespread belief that the effective application of high-performance computing will dramatically increase industrial innovation, progress in this area has been slow and limited because of a combination of technical and economic impediments. Here, such impediments are outlined, along with efforts to address them.

It's well recognized that US industry must focus on innovation. A review of the current Council on Competitiveness publication list (see www.compete.org/publications) clearly indicates that the application of simulation using high-performance computing (HPC) is critical to industrial innovation. Case studies demonstrate the importance of HPC across all industrial sectors. It's also well recognized that taking advantage of advances in nanotechnology is at the core of many of the innovations possible in product development and healthcare. However, the ability to translate those advances into new products and industries requires the transformation of existing modeling, analysis, and design methodologies into ones that explicitly account for the interactions of phenomena across the atomic, molecular, microscopic, and macroscopic scales. The computational needs of such simulations are dramatically higher than those of single-scale analyses, and the software infrastructure needed is also much more complex.

Some companies make extensive use of massively parallel simulation. What isn't as obvious is that in areas where computer-aided engineering (CAE) has been used for many years, the level of computation being used for the majority

of simulations is far from what's needed, and it's far below what current HPC systems can provide. Closer examination of the engineering problems being addressed indicates that, in most cases, the resolution of the models and discretizations applied isn't high enough for engineers to ensure the simulation results' reliability, and the simulations being applied are at a single scale, ignoring the innovations made possible by performing multiscale simulations. For example, in an April 2009 case study,¹ a 168-processor system was applied to support a major manufacturer's HPC needs. Although this case study does demonstrate impressive gains, 168 cores is less than 1/1,000th of the 294,912 processors used for a single simulation with tools² that we're applying to industrial problems. Additionally, these massively parallel machines can support concurrent execution of multiple simulations. This capability for high throughput when applied to design optimization and parameter studies can result in a dramatic reduction in time to completion.

You could argue that machines with hundreds of thousands of processing cores are well beyond what industry would obtain—however, industry will easily be able to justify next-generation massively parallel machines with more than 10,000 cores due to the continued dramatic decreases in machine costs and power requirements over that of current systems. In addition, through opportunities such as the US Department of Energy's (DOE's) Innovative and Novel Computational Impact on Theory and Experiment (INCITE) and the National Science Foundation's (NSF's) Extreme

1521-9615/13/\$31.00 © 2013 IEEE
COPUBLISHED BY THE IEEE CS AND THE AIP

MARK S. SHEPHARD, CAMERON SMITH, AND JOHN E. KOLB
Rensselaer Polytechnic Institute

Science and Engineering Discovery Environment (XSEDE), along with Europe's Partnership for Advanced Computing in Europe (PRACE), industry has access to leadership-class computing facilities (with more than 100,000 processors) for use on their largest problems.

Here, we overview the current state-of-the-art with respect to providing massively parallel computing technologies and indicate the impediments to its widespread application in CAE. After considering these impediments, we outline some activities that will improve industry's ability to apply massively parallel computing.

Current State-of-the-Art

Advances in hardware and algorithms have provided many orders of magnitude improvement in our ability to perform large-scale simulations. Computer-aided simulation software can operate on petascale computers—for example, unstructured mesh computational fluid dynamics (CFD) software that scales to more than 290,000 processing cores.² As plans to move to exascale computing proceed,³ the inability to effectively increase CPU clock rates requires that all truly large-scale computations be performed on massively parallel computers. These future massively parallel computers will be more heterogeneous and therefore more complex to program. On the positive side, progress on the development of next-generation massively parallel computers is leading to systems that are much more cost effective to purchase, to provide electric power to run, and to support system operations. This means an increased ability to cost-effectively employ the most computationally intense simulations in engineering design processes, assuming the required software tools and methods of applying the software are available.

The US national labs—particularly the DOE, with programs such as the Scientific Discovery through Advanced Computing (SciDAC) and Predictive Science Academic Alliance Program (PSAAP)—are actively developing new generations of software that can effectively operate on massively parallel computers. These developments include simulation tools for DOE applications and software that aids in the development of large-scale simulation tools. The following are four examples of different classes of tools that help support the development of parallel simulations:

- Trilinos (see <http://trilinos.sandia.gov>) consists of a large set of software components that build, to varying degrees, on a large software

infrastructure that can be used to construct large-scale, multiphysics simulations.

- The Portable, Extensible Toolkit for Scientific Computation (PETSc; www.mcs.anl.gov/petsc/petsc-as/features/index.html) is known primarily for its set of linear and nonlinear algebraic system equation solvers that have been integrated into many simulation codes.
- Zoltan (www.cs.sandia.gov/Zoltan/Zoltan_phil.html) is a parallel load balancing service that interacts with application data to determine how to distribute it for the most effective parallel execution.
- The Interoperable Technologies for Advanced Petascale Simulations (ITAPS; www.itaps.org) has interoperable interface components that provide an infrastructure to support unstructured mesh operations on massively parallel computers.

In addition, there are several parallel analysis procedures that execute specific simulations produced by DOE and Department of Defense (DOD) labs. These primarily open source software packages are beginning to receive increased attention by industry and, to some extent, independent software vendors (ISVs). Although the open source nature of such software is attractive, the majority of these software packages are developed and supported by small teams that are typically focused on the advancement of a specific science application. Thus, the packages typically include specialized features designed for use by domain experts, they're complex to integrate into a complete simulation infrastructure, they're not of industrial grade, and they lack adequate support systems for broad use by industry. There are a few packages that have been made more generally usable and are supported by more substantial developer teams. However, the ability for those teams to continue to provide support through government R&D budgets is a complex issue and not ensured in the long run.

As CAD and CAE technologies have matured, the use of advanced engineering simulations has become a cornerstone in the design and manufacture of products, ranging from aerospace vehicles to consumer products to medical devices. The key CAD/CAE tools being used in these processes include geometric design, analysis-model generation, and engineering analysis and visualization. An increasing number of these engineering analysis packages are capable of executing in parallel. In some cases, parallelism has focused on taking advantage of the higher-core-count shared memory workstations rather than addressing distributed memory methods as needed for internode

parallelism on massively parallel computers. Those packages that have addressed a more complete parallelism path typically must develop new versions of the code to gain a reasonable level of scalability. However, these new codes, at least initially, have a limited set of functionalities as compared to existing, more fully featured codes that have been under development for many years.

Some CAE-oriented ISVs have begun to develop new generations of software that employ data infrastructures and algorithms that let them operate and, for the computationally intensive portions, scale on massively parallel computers. In addition, some of this software is designed to interact through easy-to-use interfaces, which lets users combine software components from multiple sources to meet their simulation needs. The business models being used by these ISVs range from providing support and advancement of open source software to licensing proprietary software components.

Impediments to Industrial Use of HPC

Although there's progress being made on the adoption of large-scale parallel computing in specific cases, there are a set of impediments that must be overcome for its broad applications in the engineering of products and processes. These impediments include

- the inability to access suitable parallel computing hardware;
- a lack of scalable application software;
- the high cost of software licenses;
- insufficient internal structures and personnel; and
- an insufficiently defined business case.

The application of simulations on parallel computing clusters (typically 16–64 processors) and/or specialized parallel workstations (such as GPUs) is reasonably common in industry. In many cases, this came about because engineering departments were in a position to obtain the hardware within their unit and didn't have to go through a long process to gain acceptance by the corporate IT organization. As these systems became more numerous, several companies began having their IT organizations provide the needed hardware capabilities, which wasn't too hard because the computing hardware was pretty much general purpose. Obtaining larger systems was much less common due to the combination of high cost and the complexity of operations. With large-scale parallel computers, the cost had to consider

hardware, electric usage, and systems support/operations. Key aspects of the cost side of this issue are actively being addressed. The cost of the hardware continues to decrease and, in the case of massively parallel computers, future machines will be much more energy efficient. Because of the reliability demands of the largest parallel computers, the systems tend to be easier to support than previous parallel computers, where support refers only to just the basic operation (not to the providing of the applications software).

The options for companies to obtain access to the needed parallel computing hardware are increasing. In addition to the option to obtain their own in-house systems, companies have an increasing number of options to gain access to externally operated machines. One option that's quickly growing in popularity is commercial "cloud" computing systems that provide highly cost-effective computing. Although such systems will meet a number of the parallel computing needs of industry, they're currently not well suited to support a number of types of simulations with a high degree of parallelism. As indicated in the previous section, there are opportunities for industry to obtain access to the large national supercomputer centers. However, this access is typically limited to a complex "hero" type of application, requires winning a grant, and might have restrictions on the degree of proprietary work that can be done. Another option that's in more of an experimental phase is access to regional computing facilities that provide access to computing power as well as some degree of user support. In any of these cases, the widespread adoption of any of these modes within a company will require the involvement of the central IT organization to provide the needed systems, or to ensure that the access to external systems will meet the companies' needs with respect to providing reliable computing, security, and the ability to do company proprietary work. It's common for a company's IT organizations to be not only reluctant to move in these directions but also to lack the specific technical expertise required to most effectively do it. This is as serious a structural and personnel issue as any other in terms of keeping a company from moving to large-scale parallel computing.

Increasingly, there's recognition that the lack of properly integrated and supported software is the pacing factor in preventing industry from applying massively parallel computing. Although there are analysis codes that effectively scale to more than 100,000 compute cores, the state of parallel CAE software is far from supporting most of

the industry's needs. Almost all codes that scale to large numbers of cores are research codes that are quite limited in functionality and not easily applied by industry. Current parallel versions of commercial-grade CAE software in most, but not all, cases scale adequately from 32–128 cores. More modern commercial codes are doing much better on scaling. For example, AcuSolve scales as well as some of the best scalable implicit research codes. Although it will continue to take time, and additional push from the user community, this situation will likely continue to improve. The expected sources of improvement will be the “hardening”/improvement of open source codes (for example, OpenFoam) and new versions of commercial CAE software.

In addition to having the analysis codes scale on massively parallel computers, it will become increasingly important that all the pre- and post-processing tools also run effectively in parallel on the same systems. In addition, it's critical that the “mesh-generation” aspects of preprocessing are much more automated than most currently available systems. One driver for parallel versions of these tools is the time that tools like automatic mesh generators can take when generating meshes on the order of 100 million elements. The other driver is the time and effort required to simply move the analysis model data and results between different computing systems, which is often higher than the execution of pre- and postprocessing operations. On the postprocessing side, parallel visualization packages (such as ParaView and Visit) are available and are being heavily used. Parallel model construction tools aren't as readily available; the vast majority are only serial versions. The meshing, geometry, and solution field components available from Simmetrix are an exception; they operate in parallel, interacting with a parallel-partitioned mesh and associated parallel geometry.⁴

Given access to the desired level of parallel computers and software that can effectively execute the simulations of interest at scale, the next impediment to applying massive simulation is the cost of software licenses. It's pretty clear that if the cost of software license for a 1,000-processor parallel simulation is a thousand times that of the single-processor simulation, that increase in cost will preclude most users from doing massively parallel simulations. Opinions as to what's an appropriate license pricing range from a lower limit where the price for x processors is the same as the cost for a single processor, to an upper limit, where the price for a run on x processors is x

times the single-processor license price. In reality, neither limit should be viewed as realistic: CAE software is large and complex, and scalable parallel algorithms are hard to program. Thus, there's a substantial cost involved with first developing scalable parallel versions of the software as well as the added cost associated with porting it to various parallel machines and performing the needed regression tests. At the other end, even if you accept the premise that you should pay for the level of calculation performed, there's little likelihood that companies can make the business case for applying massively parallel simulations with a linear cost increase.

Users should expect the license cost for any level of parallel version to be higher than a license for a serial version due to the added development and support costs. Although it does depend on the underlying design of the starting software, the easiest level of parallelization would typically be to use shared memory parallelism over multiple cores on a single workstation. The next level would likely be executing a computationally dominant, but limited, function (such as solving large linear systems) on an accelerator (such as a GPU). The third mode of parallelization is distributed memory message passing, which is typically far more expensive to implement than shared memory or even an accelerator. For more than a decade leading up to 2012, MPI-based distributed-memory parallelism was the best option to scalability on large parallel computers. As we go forward, most agree that the effective use of the machines we'll see for quite a few years will require at least two levels of a parallelism. They'll require message passing between compute nodes and, depending on the nodes' configuration, the use of accelerators (GPUs on a Cray node), or heavy threading over the cores on a node (for example, executing up to four threads per core on each of the 16 cores on a Blue Gene/Q).

There's a general perception in the user community that CAE vendors are continuing to use old pricing models that have substantial increases in cost as the number of compute cores increase. Although the change toward cost-effective parallel software licenses might be slow, it is changing. For example, CD-adapco provides specific analysis codes through a power session license that cost approximately twice as much as a single-processor, but can be run on as many cores as desired. On the mesh-generation side, the Simmetrix parallel mesh-generator license cost isn't a function of the number of cores used. Increasingly, CAE software providers are adopting license schemes that allow

companies to make cost-effective use of massively parallel computing.

The internal structures within most companies aren't ideally suited to take full advantage of HPC. The IT organizations that provide computing and networking aren't well connected to the engineering people that could really use HPC and there's often no incentive to improve these connections. Many companies do have technical people on the engineering side who would be capable of pushing to increase HPC. However, putting the effective use of HPC for product or process design in place takes time and effort, and thus proves to be a disincentive to middle management, because they're typically evaluated based on the ability to continually move design processes forward in the short term. Both of these internal structure issues can be addressed only by the active involvement of the higher-level management in changing the rewards systems and/or their internal structures. Companies such as Proctor & Gamble, Goodyear, Whirlpool, Caterpillar, and others have seen excellent return on their investments by strongly embracing these technologies. As an example of the level of support, Proctor & Gamble holds an internal modeling and simulation conference on a yearly basis that's attended by around 300 employees. The involvement of higher-level management in making the business case for the application of HPC is critical. Although there are well-known cases where companies have clearly embraced these technologies company wide, and cases where company representatives state that HPC was critical to keeping their company competitive, the details—the dollar side—of making the business case is difficult to document, and companies that have done these analyses aren't likely to share the data. Thus, companies that are considering embracing the use of HPC often have to both take a fairly long view of the process and have some degree of faith that the investment will pay off. This isn't something that's easy for companies to do.

Reducing the Impediments to HPC

Recently, there's been an increased understanding that specific efforts are required to remove impediments, if industry is to realize the full benefits of HPC in its design process. Here, we briefly discuss two specific efforts at reducing impediments. The first is a project supported by New York State to provide industry with the computing power and technical support to help it apply computing power to its problems of interest. The second activity is an R&D project that's

building on DOE- and NSF-supported research on interoperable simulation methods to use those methods in the construction of industrial simulation workflows.

New York State High-Performance Computing Consortium

The High-Performance Computing Consortium (HPC2; <http://hpc2.org>), supported by the Empire State Development Division of Science, Technology, and Innovation (NYSTAR), is a three-year, \$3-million effort to address impediments. HPC2 supports computational scientists working directly with New York State industry to apply massively parallel simulations on supercomputer systems that are also supported by New York State. Computational scientists are based at Rensselaer Polytechnic Institute, the University of Buffalo, and the State University of New York (SUNY) Stony Brook/Brookhaven National Lab. Critical to the success of computational scientists is the base institution's faculty, who have extensive knowledge of HPC in a broad range of application areas, existing industrial and software vendor collaborations, and on-site HPC hardware systems programmers.

Industrial partners work with Rensselaer through HPC2 at the level needed to address their computing requirements. At one extreme are industrial partners that have the necessary technical personnel, business case, and software to utilize available HPC hardware. One such partner is GNS Health Care. For these interactions, a NYSTAR-supported allocation on the 1,848-processor IBM Blade Opterons cluster and 32,768-processor IBM Blue Gene L at the Computational Center for Nanotechnology Innovations is sufficient.

The more common cases are industrial partners that identify a computing need but face most of the impediments we've previously mentioned. For these interactions, the appropriate combination of computational scientists, researchers, and software vendors are brought to bear on the problem. In many cases, the initial problem statement is broad, in which case the first task is to define a specific relevant problem that includes the relevant physical phenomena and geometric complexity. Execution of this first problem demonstrates the computational and analytical performance of the chosen HPC software technologies as well as the ability of those technologies to integrate into simulation workflows accessible to the industrial partners. The level of workflow integration can be a major limitation to technology transfer when there's little to no personnel with HPC experience; the engineer must be able to efficiently use

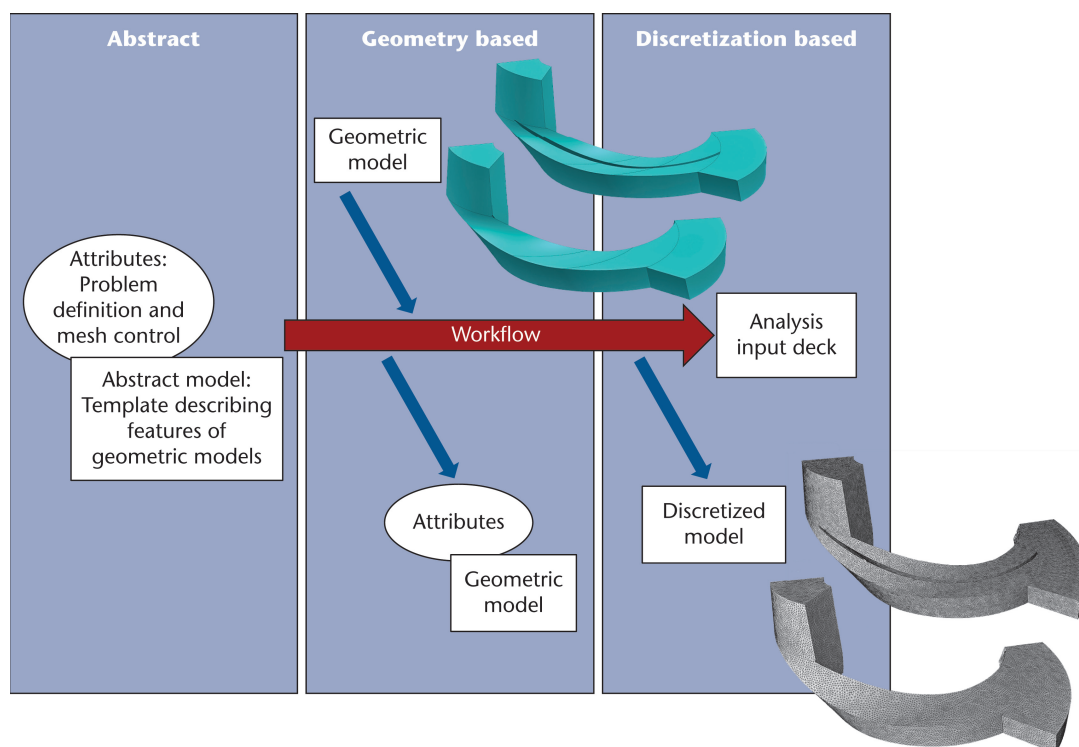


Figure 1. Workflow from abstraction to solver input deck. This is based on Simmetrix's *SimModSuite* documentation.

the tools without being unnecessarily burdened by the underlying technology.

From a successful first problem demonstration, the business case can be stated and supported. For the technical contact at the industrial partner, the business case often provides the necessary management-level support to continue work with HPC2 computational scientists to generalize the workflows for general application. Typically, industrial partners understand there's potential for HPC to improve their competitive advantage, but face one or two of the impediments. Working with HPC2 helps them bridge the technical gaps that exist, thus removing the impediments. The following are brief descriptions of the efforts of HPC2 on the development of workflows for two different equipment manufacturers.

Hydraulic engineers at a New York State pump manufacturer execute pump design using CFD simulations over a range of operating conditions with the goal of developing an optimum pump configuration and geometry. Two factors are critical to providing this company with a competitive advantage. The first is that the simulations provide reliable predictions that don't have to be constantly validated via testing and the second is that the simulations can be executed within a few days, and not the weeks historically required. Because

the vast majority of the time required for the simulations is in the problems setup, Rensselaer computational scientists are working with Simmetrix software engineers to define a workflow to automate the problem setup for the ensemble of simulations to be executed. Simulation setup entails associating mesh-generation controls and problem-definition attributes with the geometric model, mesh generation, and creation of the CFD analysis software inputs.

In the workflow depicted in Figure 1, based on the Simmetrix *AbstractModel* component, a template is defined that describes the common features of a set of geometric models, the mesh-generation controls, and the problem-definition attributes associated with the template. The template is then associated with a specific instance of the geometric model, and the mesh and CFD analysis inputs are generated. Using the workflow, hydraulic engineers create the set of inputs required for a CFD analysis by inputting an instance of a geometric model that fits the defined template, thus avoiding many tedious and error-prone steps. This automation increases the time the hydraulic engineer can spend on design and analysis, which results in better products. Future efforts will aim at leveraging the Computational Center for Nanotechnology Innovations (CCNI) systems for the simultaneous

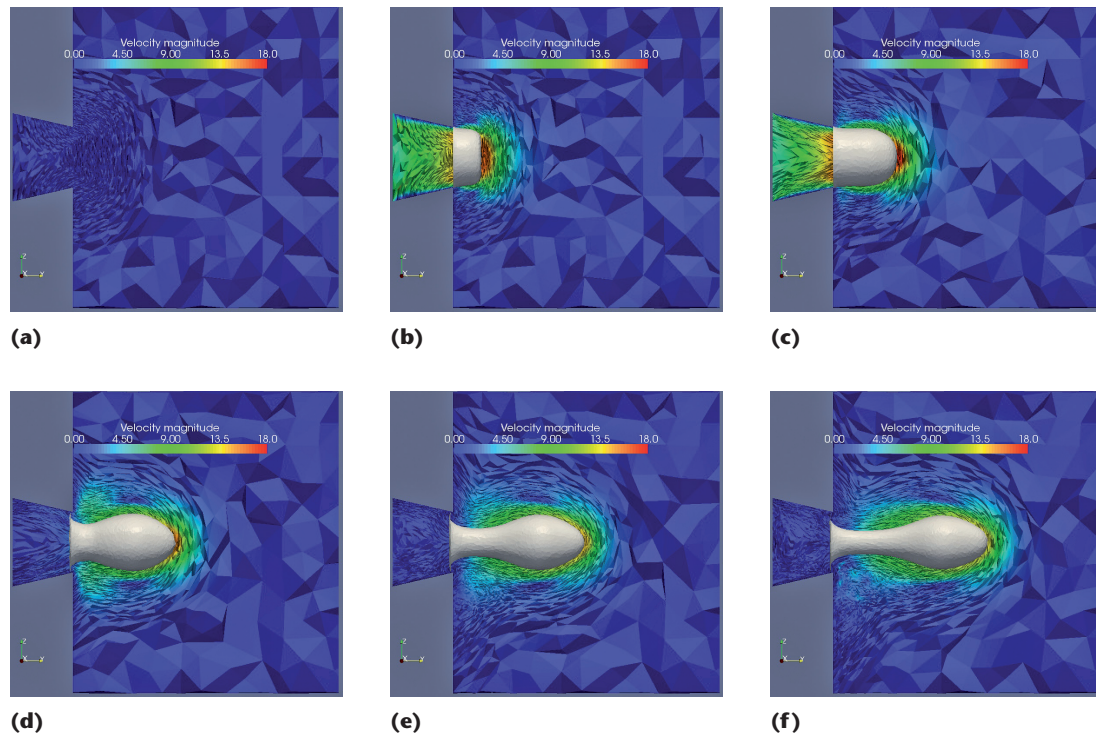


Figure 2. An axial slice of the 3D mesh at six consecutive adaptation cycles in the multiphase Parallel Hierarchic Adaptive Stabilized Transient Analysis (PHASTA) simulation. The grey surface is the phasic interface.

execution of the ensemble of simulations and subsequent automated aggregation of results. The combination of these efforts will result in a drastic reduction in the total time required for setting up and executing the design study.

Micromechanical device engineers at a New York State equipment manufacturer study the design of a multiphase flow system that's driven by a structural boundary condition using a commercial CFD software suite run on in-house multicore workstations. Engineers typically run reduced-fidelity 2D simulations because of the long execution time of 3D simulations, limited computing resources, and finite design periods. In these simulations, the fidelity is further reduced as the structure driving the boundary condition isn't influenced by the fluid flow.

Rensselaer computational scientists defined and demonstrated an end-to-end workflow for guiding device design that uses the Parallel Hierarchic Adaptive Stabilized Transient Analysis (PHASTA) CFD suite of tools for automated 3D parallel adaptive simulations² that account for fluid-structure interactions. Modifications to the PHASTA parallel fluid dynamics simulation software were required to interface to the structural mechanics simulation software provided by the device engineers. To support this interface, mechanisms were implemented to interpolate

fields between the different time and spatial discretizations. Modifications were also made to the PHASTA adaptation component to refine the mesh across the phasic interface, accurately resolve the flow near the change in material properties, coarsen the mesh away from the interface, reduce computational costs, and blend the mesh sizes between the fine and coarse zones.

Figure 2 depicts the mesh over six adaptation cycles. Simulations run with this workflow were efficiently run on up to 512 processors of the CCNI Opteron system. Efforts are underway for increased scalability of this workflow by defining a file-free coupling of the PHASTA analysis component with the adaptation component implementing mesh modification procedures that reduce the frequency of mesh adaptations, and replacing the serial structural mechanics simulation software with a parallel one. This scalable workflow will enable device engineers to leverage the computing resources of the CCNI to run high-fidelity system simulations in hours instead of days and thus reduce the time required to develop a new device.

Approach to Eliminating Technical Impediments

The execution of industrial simulations typically requires a workflow that couples a number of different simulation components. For example, a single

physics simulation requires linking CAD systems with mesh generators and a mesh-based solver. For multiple reasons—ranging from a company's best practices and validation processes to the interactions of multiple companies doing different steps in the process—these tools are typically not provided within a single software system, and thus the effective coupling of multiple software tools is critical. The need to couple tools becomes more acute in multiphysics problems where different analysis tools are coupled and in multiscale problems where different tools based on different models using different physical parameters must be coupled across scales. The ability to support these workflows is made even more complex when specific computationally intensive simulation steps must be executed on a massively parallel computer. One approach to the coupling of simulation components that has a strong potential for the effective construction of simulation workflows is the use of interoperable components that interact through functional interfaces.

One well-established example of the utility of a functional interface is a unified geometry interface that allows access to geometry information within a number of geometric modeling kernels and CAD systems. Such an interface has been developed to support the full range of mesh-based simulation geometry needs.⁵ It supports the commercial and open source geometric modeling kernels and many CAD systems.

Another example that's relevant to a number of industrial simulation needs is the ITAPS interoperable interfaces for unstructured mesh-based continuum simulations being developed as part of the DOE SciDAC program center. The ITAPS interfaces support a full range of low-level geometry, mesh, and field operations all building on a parallel, distributed mesh. These procedures are being used to develop parallel, adaptive simulation procedures² capable of providing reliable simulation results on problems defined over general 3D domains. Such capabilities are exactly the type needed by industry. The DOE Trilinos project is building sets of simulation components using similar interface technologies. Trilinos includes full analysis components (such as finite-element analysis components) and a number of specific operational components, from linear equation solvers to dynamic load-balancing procedures. It's becoming increasingly important to support multiscale simulations, where consideration is given to the methods associated with linking information when different forms of models are used at each scale.⁶ Research consideration

of component-based approaches to multiscale simulation, including multifidelity models and knowledge-based methods, is underway.⁶

The efforts to develop interoperable tools to support automated adaptive simulations are having a direct effect on the ability of HPC2 computational scientists to develop industrially relevant workflows. The experience gained in these efforts is also providing useful feedback that's helping the research efforts. A number of the HPC2 activities developing a simulation-based design capability for a company included some form of flow simulation. In each of these cases, the company was unable to meet its design goals because the computational effort required couldn't be delivered in a time-effective manner using workstations or small local clusters. Thus, in each case it was necessary to employ a flow solver that could scale to at least several hundred processing cores. Another aspect common to each of these simulation-based design capabilities was the need to couple multiple software components together to address the company's simulation needs. The components that were used in each example included problem definitions, mesh generation, load balancing, analysis, and postprocessing. In several cases error estimation and mesh adaptation components were also included, and in some cases additional physics analyses and coupling tools were needed.

The most effective problem definitions for CAE simulations of manufactured objects are geometry-based forms. Over the years, many of the CAE vendors have moved from "mesh-based" problem definitions to more geometry-based definitions. These problem definitions are typically executed using a GUI that can accept some degree of CAD information. Many of these interfaces are oriented toward a specific vendor's set of CAE analysis tools, while others are oriented to interact through APIs supporting general interactions with geometry⁵ and simulation attribute information. In those HPC2 projects where the workflow developed employs components from several sources, we found that using a generalized interface that can interact with geometry, attributes, and analysis procedures was quite effective. Given the problem definition, most analysis procedures require that the domain be decomposed into a controlled mesh of simple shapes. The greatest flexibility is provided when the parallel analysis procedures can accept an unstructured mesh. Such procedures based on finite-element or finite-volume analysis methods are available for many classes of problems. The use of such methods allows both the application of fully automatic mesh generation

and adaptive mesh control. Parallel mesh generation and adaptation components have been used in the mesh-based HPC2 simulation workflows to provide reliable automation of the simulation process.^{2,4} Critical to the successful application of parallel adaptive simulations is the application of a dynamic load-balancing component, such as Zoltan, as the simulation progresses.

Several important insights have been gained as part of the process of developing the HPC2 simulation workflows. It's quite possible to construct highly successful parallel simulation workflows for industrially relevant problems using interoperable components. The development of these workflows does take a reasonable amount of effort, although as the tools have improved, and we've gained more experience, we're able to construct new workflows quite quickly (in a number of weeks).


These efforts have also taught us valuable lessons about developing interoperable interfaces. The first is that there are some common high-level interfaces that can be defined for coupling many of the simulation components. These interfaces are primarily focused on the methods to load the input into the data structures of analysis components and the extraction of simulation data for those components. In those cases, when a geometry-based problem definition and a solution field interface is used, this approach lets us quickly integrate multiple meshing, analysis, and visualization tools, and it also lets us quickly replace any of those components.

Within the HPC2 projects, multiple commercial and open source analysis components have been used, allowing companies to compare the effectiveness of those components. The typical initial implementation of these interface methods tends to pass information between major components using files. Although it offers the most straightforward implementation, file I/O (serial or parallel) is still a major bottleneck when executing large-scale parallel simulations. When the method used to execute the coupling of components is through APIs, it's conceptually straightforward to bypass the use of files and transfer information directly between component data structures. Technically, the effective implementation, given already-defined components, is reasonably complex. There are current research efforts underway that are addressing the technical complexities in the construction of completely adaptive simulation processes, which start with existing file-based components and produce a file-free adaptive simulation loop.

There are cases where more incremental operations are needed between components. In those

cases, the availability of finer-grain interfaces is needed. The implementation in these cases does require a more detailed interaction of the components, which will typically require some degree of modification/extension when preexisting components are to be integrated.

As the technical, economic, and organizational barriers delaying the widespread use of large-scale parallel simulations are addressed, issues associated with training the needed workforce will come to the forefront. In terms of training, there's often an emphasis on training those capable of developing the needed simulation technologies. Although there's a real need for these personnel, the more serious need is for engineers who can effectively integrate these technologies into simulation workflows.

Engineers are needed who can serve as a bridge between the application engineers and the computational scientists providing the new simulation technologies. These individuals must understand the available simulation technologies and have the technical skills to develop simulation workflows from a set of simulation requirements defined by application engineers. Currently, there aren't many individuals with the expertise and temperament to effectively execute this job. As part of a recently started NSF project, small numbers of undergraduates will work with computational scientists and applications engineers to integrate simulation technologies into established workflows. The experience gained from this project will help identify the subjects that must be included in a student's education to produce individuals with the skills needed to construct new simulation workflows. These skills are critical for advancing simulation in the "missing middle" companies that hire many of these students. 

References

1. Council on Competitiveness, "Whirlpool's Home Appliance Rocket Science: Design to Delivery with High Performance Computing," case study, Apr. 2009; www.compete.org/publications/detail/682/whirlpools-home-appliance-rocket-science-design-to-delivery-with-high-performance-computing.
2. M. Zhou et al., "Unstructured Mesh Partition Improvement for Implicit Finite Element at Extreme Scale," *J. Supercomputing*, vol. 59, no. 3, 2012, pp. 1218–1228.
3. DOE Advanced Scientific Computing Research (ASCR), *Int'l Exascale Software Project Roadmap*, draft, Jan. 2010; www.exascale.org/mediawiki/images/a/a1/lesp-roadmap-draft-0.93-complete.pdf.

4. S. Tendulkar et al., "Parallel Mesh Generation and Adaptation for CAD Geometries," *Proc. NAFEMS World Congress*, Nat'l Agency for Finite Element Methods and Standards (NAFEMS), 2011; www.scorec.rpi.edu/REPORTS/2011-2.pdf.
5. M.W. Beall, J. Walsh, and M.S. Shephard, "A Comparison of Techniques for Geometry Access Related to Mesh Generation," *Eng. with Computers*, vol. 20, no. 3, 2004, pp. 210–221.
6. F. Delalonondre, C. Smith, and M.S. Shephard, "Collaborative Software Infrastructure for Adaptive Multiple Model Simulation," *Computational Methods of Applied Mechanical Eng.*, vol. 199, nos. 21–22, 2010, pp. 1352–1370.

Mark S. Shephard is the Samuel A. and Elisabeth C. Johnson, Jr. Professor of Engineering at Rensselaer Polytechnic Institute, where he is also director of the Scientific Computation Research Center. His research interests include high-performance computing, unstructured mesh generation and adaptation, and multiscale simulation. Professor Shephard has a PhD in civil engineering from Cornell University. He's a fellow in (and a past president of) the US Association

for Computational Mechanics, a fellow of the International Association for Computational Mechanics, and the editor of *Engineering with Computers*. Contact him at shephard@rpi.edu.

Cameron Smith is a computational scientist at the Computational Center for Nanotechnology Innovations (CCNI) and the Scientific Computation Research Center (SCOREC) within Rensselaer Polytechnic Institute. His research interests include simulation automation, parallel computation, and multiscale simulation. Smith has an MS in computer science from Rensselaer Polytechnic. Contact him at smithc11@rpi.edu.

John E. Kolb is vice president of Information Services and Technology and CIO at Rensselaer Polytechnic Institute. His interests include engineering, science, management, and IT. Kolb has a master's degree in electrical engineering from Rensselaer. He's a member of the American Society for Engineering Education and IEEE, and a recipient of the Boeing Outstanding Engineering Educator Award. Contact him at kolbj@rpi.edu.



Experimenting with your hiring process?

Finding the best computing job or hire shouldn't be left to chance. IEEE Computer Society Jobs is your ideal recruitment resource, targeting over 85,000 expert researchers and qualified top-level managers in software engineering, robotics, programming, artificial intelligence, networking and communications, consulting, modeling, data structures, and other computer science-related fields worldwide. Whether you're looking to hire or be hired, IEEE Computer Society Jobs provides real results by matching hundreds of relevant jobs with this hard-to-reach audience each month, in **Computer magazine** and/or online-only!

<http://www.computer.org/jobs>

The IEEE Computer Society is a partner in the AIP Career Network, a collection of online job sites for scientists, engineers, and computing professionals. Other partners include *Physics Today*, the American Association of Physicists in Medicine (AAPM), American Association of Physics Teachers (AAPT), American Physical Society (APS), AVS Science and Technology, and the Society of Physics Students (SPS) and Sigma Pi Sigma.

IEEE  computer society | **JOBS**