

International Journal of High Performance Computing Applications

<http://hpc.sagepub.com/>

The International Exascale Software Project: a Call To Cooperative Action By the Global High-Performance Community

Jack Dongarra, Pete Beckman, Patrick Aerts, Frank Cappello, Thomas Lippert, Satoshi Matsuoka, Paul Messina, Terry Moore, Rick Stevens, Anne Trefethen and Mateo Valero

International Journal of High Performance Computing Applications 2009 23: 309 originally published online 12 October 2009
DOI: 10.1177/1094342009347714

The online version of this article can be found at:
<http://hpc.sagepub.com/content/23/4/309>

Published by:



<http://www.sagepublications.com>

Additional services and information for *International Journal of High Performance Computing Applications* can be found at:

Email Alerts: <http://hpc.sagepub.com/cgi/alerts>

Subscriptions: <http://hpc.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

Citations: <http://hpc.sagepub.com/content/23/4/309.refs.html>

THE INTERNATIONAL EXASCALE SOFTWARE PROJECT: A CALL TO COOPERATIVE ACTION BY THE GLOBAL HIGH-PERFORMANCE COMMUNITY

Jack Dongarra¹
Pete Beckman²
Patrick Aerts³
Frank Cappello⁴
Thomas Lippert⁵
Satoshi Matsuoka⁶
Paul Messina²
Terry Moore¹
Rick Stevens⁷
Anne Trefethen⁸
Mateo Valero⁹

Abstract

Over the last 20 years, the open-source community has provided more and more software on which the world's high-performance computing systems depend for performance and productivity. The community has invested millions of dollars and years of effort to build key components. Although the investments in these separate software elements have been tremendously valuable, a great deal of productivity has also been lost because of the lack of planning, coordination, and key integration of technologies necessary to make them work together smoothly and efficiently, both within individual petascale systems and between different systems. A repository gatekeeper and an email discussion list can coordinate open-source development within a single project, but there is no global mechanism working across the community to identify critical holes in the overall software environment, spot opportunities for beneficial integration, or specify requirements for more careful coordination. It seems clear that this completely uncoordinated development model will not provide the software needed to support

the unprecedented parallelism required for peta/exascale computation on millions of cores, or the flexibility required to exploit new hardware models and features, such as transactional memory, speculative execution, and GPUs. We believe the community must work together to prepare for the challenges of exascale computing, ultimately combing their efforts in a coordinated International Exascale Software Project.

Key words: exascale, software, international, applications, scientific computing

1 The Time is Now

When processor clock speeds flatlined in 2004, after more than 15 years of exponential increases, the computational science community lost the key to the automatic performance improvements its applications had traditionally enjoyed. Subsequent developments in processor and system design—hundreds of thousands of nodes, millions of cores, reduced bandwidth and memory available to cores, inclusion of special purpose elements—have made it clear that a broad divide has now opened up between the software infrastructure that we have, and the one we will certainly need to have to perform the kind of computationally intensive and data intensive work that tomorrow's scientists and engineers will require. Given the daunting conceptual and technical problems that such a change in design paradigms brings with it, we believe that this software gap will require an unprecedented level of cooperation and coordination within the worldwide open-source software community. In forming the *International Exascale Software Project* (IESP), we hope to plan for and catalyze the kind of community-wide effort that we believe is necessary to meet this historic challenge.

¹ DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, THE UNIVERSITY OF TENNESSEE, USA
(DONGARRA@EECS.UTK.EDU)

² ARGONNE LEADERSHIP COMPUTING FACILITY, ARGONNE NATIONAL LABORATORY, USA

³ NETHERLANDS ORGANISATION FOR SCIENTIFIC RESEARCH, THE NETHERLANDS

⁴ INRIA, LABORATOIRE EN RECHERCHE INFORMATIQUE, FRANCE

⁵ JÜLICH SUPERCOMPUTING CENTRE, GERMANY

⁶ TOKYO INSTITUTE OF TECHNOLOGY, JAPAN

⁷ COMPUTING ENVIRONMENT AND LIFE SCIENCES DIVISION, ARGONNE NATIONAL LABORATORY, USA

⁸ UNIVERSITY OF OXFORD, OXFORD, UK

⁹ TECHNICAL UNIVERSITY OF CATALONIA, SPAIN

Our belief in the need for broad-based, coordinated action by the global scientific software community to address the looming crisis reflects, in part, the fact that computational methods are now universally accepted as indispensable to future progress in science and engineering. The last time a disruption of comparable dimensions occurred—during the transition from vector to distributed-memory supercomputers more than two decades ago—only a relatively small part of the scientific community felt the consequences of the struggle to replace, wholesale, the programming models, numerical and communication libraries, and all of the other software components and tools on which application scientists were already building. Computational science was still relatively young, and computationally intensive methods were still largely the province of relatively small scientific elite in a relatively small number of physical sciences.

Today, aided by the success of the scientific software research and development community, researchers in nearly every field of science and engineering have been able to turn computational modeling/simulation and high-throughput data analysis to open new areas of inquiry (e.g., the very small, very large, very hazardous, very complex), to dramatically increase research productivity, and to amplify the social and economic impact of their work. Recent reports (NRC, 2008; Stevens et al., 2008) make a compelling case, in terms of both scope and importance, for the profound expansion of our research horizons that will occur if we can rise to the challenge of peta/exascale computing. However, in light of the radical changes in computing we are currently undergoing, it is clear that the software infrastructure necessary to make that ascent does not yet exist and that we are a long way from being in a position to create it.

At the same time, the increasing use of computationally intensive modeling and simulation across science and engineering generally, along with the Internet revolution, have made the economic, social, and political importance of compute and data research manifest. Such work is now being carried out at every level of the platform development change, from the desktop, to the campus cluster, to the national supercomputer. Our hope that we can act, individually and collectively, to solve the frightening array of critical problems that now confront our world depends on the power of science and engineering, enabled by advanced cyberinfrastructure, to help us formulate those solutions. Thus, bridging the looming software infrastructure gap is not only essential to future scientific progress, but also in the vital and strategic interest of governments and societies around the world.

We have formed the IESP to initiate and carry out the planning and organizational processes that are necessary to solve this fundamental problem. More specifically,

The guiding purpose of the IESP is to empower ultra-high resolution and data intensive science and engineering research through the year 2020 by developing a plan for (1) a common, high quality computational environment for peta/exascale systems and for (2) catalyzing, coordinating, and sustaining the effort of the international open source software community to create that environment as quickly as possible.

There are good reasons to think such a plan is urgently needed. First and foremost, the magnitude of the technical challenges that the new architectures and systems bring with them, and the corresponding sweep of the changes that will be required to high-performance computing (HPC) software infrastructure, are formidable, to say the least. These problems, which are already appearing on the leadership class systems of the US National Science Foundation (NSF) and Department of Energy (DOE), are more than sufficient to require the wholesale redesign and replacement of the operating systems, programming models, libraries and tools on which the computational science and engineering communities have come to depend. Moreover, both HPC vendors and representative application communities will need to provide substantial input to the planning process. Second, the complex web of interdependencies and side effects that exist among HPC the software components of advanced computing infrastructure means that making sweeping changes to this infrastructure will require a high degree of coordination and collaboration. Failure to identify critical holes or potential conflicts in the software environment, to spot opportunities for beneficial integration, or to adequately specify component requirements will tend to restrict or disrupt everyone's progress. Since creating a software environment adapted for peta/exascale systems (e.g. NSF's Blue Waters) will require the collective effort of a broad community, this community must have good mechanisms for internal coordination. Finally, it seems clear that the scope of the effort must be truly international: in terms of its rationale, the HPC software infrastructure serves scientific communities that include global collaborations working on problems of global significance and leveraging resources in transnational configurations; in terms of feasibility, the dimensions of the task of recreating this infrastructure to meet the new realities of advanced scientific computing is simply too large for any one country, or small consortium of countries, to undertake on its own.

The IESP has been formed to help achieve this goal. Its initial effort is to stage a series of three international meetings, one each in the United States, Europe and Asia beginning in the spring 2009. Information about all of these meetings can be found at the project website, <http://www.exascale.org>. On-line collaboration technology will

be heavily used to lay the groundwork for and consolidate the results of each meeting and to ensure that the final plan develops in a way that elicits broad participation and input. The plan for these meetings incorporates the following objectives:

- *Provide a framework for organizing the software research community.* The IESP will articulate an organizational framework designed to enable the international software research community to work together to deliver more capable and productive HPC systems. The framework will include elements such as initial working groups, outlines of a system of governance, alternative models for shared software development with common code repositories, feasible schemes for selecting valuable software research and incentivizing its translation into usable, production-quality software for application developers, etc. This organization must also foster and help coordinate R&D efforts to address the emerging needs of users and application communities on new platforms, such as Kraken and Blue Waters.
- *Create a thorough assessment of needs, issues and strategies.* As part of its planning process, the IESP will assess the short-term, medium-term and long-term needs of applications for peta/exascale systems. Participation in the IESP from representative application communities and vendors will help ensure the adequacy of these assessments. The work of the organization that emerges from the IESP must be prepared to provide the NSF and other domestic and foreign research-oriented agencies with a series of well-crafted reports on the critical technical issues of peta/exascale software infrastructure and with alternative strategies, both technical and programmatic, for solving them.
- *Initiate coordinated software roadmap.* Working with the results of its application needs assessment, the IESP will initiate the development of a coordinated roadmap to guide open-source HPC software development with better coordination and fewer missing components. This roadmap will help to guide both cooperative development and joint research efforts.
- *Encourage and facilitate collaboration in education and training.* The magnitude of the changes in programming models and software infrastructure and tools brought about by the transition to peta/exascale architectures will produce tremendous challenges in the area of education and training. The IESP plan will therefore provide for cooperation in the production of education and training materials to be used in curricula, at workshops and on-line.
- *Engage and coordinate vendor community in crosscutting efforts.* To leverage resources and create a more capable software infrastructure for supporting exascale science, the IESP will engage and coordinate with ven-

dors across all of its other objectives. Vendor participation in and contributions to all of these objectives, comprehensive application needs assessment, well-ordered but adaptive software roadmap, organized framework for cooperation, coordinated R&D programs for new exascale software technologies, will be encouraged and facilitated.

By the time this article appears, a version of the IESP plan and roadmap will have been drafted and distributed publicly. It is clear to us, however, that if these documents are to ultimately provide a viable foundation for a global cooperative effort to create the new petascale/exascale software infrastructure, they must be developed and matured through a process of discussion and criticism in which the entire HPC software community participates. We believe that building a plan and roadmap around which a broad consensus can form is one of the keys to eliciting the kind of voluntary cooperation that this unique and indispensable enterprise will require. Our call to action is a call to join in this enterprise even during this formative phase.

In the following we describe these elements of the IESP in more detail. First, in Section 2, we discuss the three critical factors that, collectively, provide a compelling motivation for creating the IESP: the ground breaking science and engineering research that will be enabled by peta/exascale computing, the daunting technical challenges that will have to be surmounted in order to make computing at that scale feasible, and the commitment that the international open-source software community will have to make in order to succeed. The discussion of the IESP initial planning process, focusing around the organization of the workshops and collateral contributions and on-line activities are presented in Section 3. We discuss who will participate (including representatives from the vendor and science application communities), how the participants will be asked/enabled to contribute (before, during and after the workshop), and how the process of drawing up a plan to achieve a working consensus will be structured. The milestones and deliverables for the project are summarized in a table. In Section 4, we describe the software roadmap. In Section 5 we describe the broader impact of the project, including its potential impact on infrastructure development, education in computational science, the health of the HPC vendor community, and international collaboration in research and education. We conclude by presenting the credentials of the principal investigators that are relevant to their leadership role in the IESP.

2 Background

The creation of the IESP is motivated by the convergence of three separate factors: the compelling science case for

peta/exascale computing, the obsolescence of current HPC software infrastructure and the formidable obstacles to replacing it, and the near complete lack of planning and coordination in the global HPC open-source software community in confronting this situation. These factors are described in turn below.

2.1 Opening New Frontiers of Discovery on the Path to Exascale Computing

A recent report from the NRC (2008), begins by acknowledging that “Many federal funding requests for more advanced computer resources assume implicitly that greater computing power creates opportunities for advancement in science and engineering.” (NRC, 2008, p. 1) Such an assumption has certainly been convenient for cyberinfrastructure builders over the past two decades, a time when HPC relentlessly charted exponential increases in available processing power. It sometimes seemed as if the demand by scientists and engineers for computational resources would not be able to catch up or keep up with the exploding supply, although it consistently has done so. However, this era of overabundance may be coming to an end. Not only are science and engineering disciplines across the board becoming ever more aggressive in their computational strategies, the practical path to exponential growth in available computing power is getting much steeper. The implicit assumption that scientists and engineers will always want more computing power is about to be explicitly tested.

The history of computational science offers good reasons to think that this traditional view will easily pass the test. As more and more disciplines have learned how to achieve their research goals and see more deeply into nature by using computationally intensive methods, the appetite for high-end computing power has become increasingly insatiable. Scientists in disciplines that build on such methods inevitably want to resolve multiscale models at larger sizes and over longer durations; to raise the number of dimensions and degrees of freedom that a model/simulation captures and increase the resolution at which it captures them; and to sharpen the accuracy of their statistical projections and quantify the residual uncertainty they involve. Exploiting ever-higher levels of computing power is essential to satisfying these needs.

Two major studies—NRC (2008) and Stevens et al. (2008)—that were recently published on the feasibility and potential impact of peta/exascale computing on diverse science and engineering research agendas explore this critical question in great detail. Conducted independently of one another, they draw heavily on input from leaders in an impressive sample of domain fields: climate, atmospheric and Earth systems sciences; astrophysics; evolutionary and systems biology, genomics and

proteomics; chemical separations; energy sciences, including combustion, nuclear fusion, Solar and nuclear fission; socioeconomic modeling. Each one, with differing levels of detail and emphasis, surveys a range of leading research questions and challenges, and attempts to ascertain whether and how much multiple order of magnitude increases in available computing power would contribute to or accelerate progress. They converge on the conclusion that there is a extremely strong science case to be made, for most but not all of the fields covered, for pressing forward to continue to escalate the high-end computing power available to the nation’s research community.

Although the details of the analysis for each individual discipline must be omitted here, a reasonable sense of this conclusion can be conveyed by a few brief examples.

Climate and Atmospheric Science. The diverse and interdisciplinary research occurring in different areas of climate and atmospheric science are characterized not only by their complexity and the level of purely scientific interest they arouse, but also by the urgency of the social, economic and political issues that they affect. It is difficult to think of a realm of science with a larger number of critical questions of global import awaiting answers. However, to achieve the kind of systematic understanding of the Earth systems that we need—one that integrates observation and coupled models of physical and bio/geo/chemical systems across various space and time scales—and to apply the theories we have to extend the range, accuracy and usefulness of our weather, pollution, and climate predictions and forecasts, requires prodigious amounts of computing power. For example, increasing the horizontal mesh resolution by a factor of between 4 and 10 for more accurate weather predictions will require a hundred- to thousand-fold increase in computing capability (NRC, 2008, p. 3, 58). Figure 1 shows, in a more general and somewhat more perspicuous way, how increasing the complexity of model integration and the resolution requirements affect the requirements for computing power (Stevens et al., 2008, p. 16).

Astrophysics. In terms of the productive application of computationally intensive methods, astrophysics is a very mature discipline. Not only do astrophysicists want to model and simulate diverse phenomena involving a large number of dimensions at incredible scales (e.g. the formation of galaxies, quasars, super massive black holes, stars and planets; the mechanisms underlying supernovae explosions and gamma ray bursts), they also have a massive and ever-expanding wealth of observational data to digest and explore. In order to resolve astrophysical processes involving such an enormous range of distances and durations at the necessary level of precision (e.g. to compare from new instruments, such as

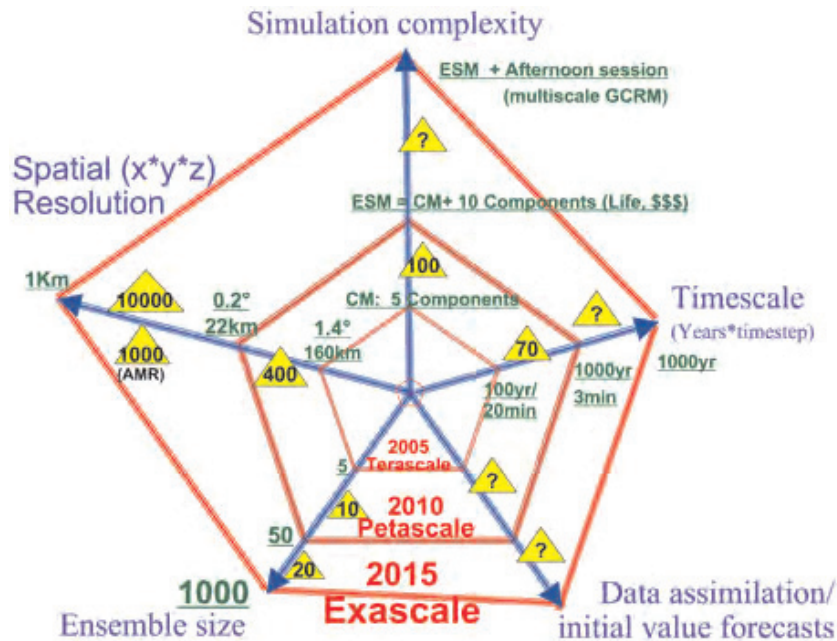


Fig. 1 Investment of exascale and petascale computational resources in several aspects of a simulation: spatial resolution, simulation complexity, ensemble size, etc. Each pentagon represents a balanced investment at a compute scale.

the Large Synoptic Survey Telescope), petascale systems of the order of tens of thousands of processors, at a minimum, will have to be utilized. Both reports conclude that astrophysics is well positioned and hungry to escalate its use of computational resources to do breakthrough science.

Energy Research. Research breakthroughs in high-efficiency, low-emission and sustainable energy generation are clearly mission critical for our society and are therefore being pursued along a number of alternative tracks. The Town Hall study reviewed four such fields—combustion, fusion, Solar and fission—in terms of their need for peta/exascale computing. Since both combustion and fusion simulation share many properties in common with astrophysics (e.g. the desire to realistically model of turbulence-driven heat, particles, and momentum losses for highly dynamic systems), one expects to find that their computational needs follow a similar profile. Indeed, the report makes clear that the science is required to, and the scientists are eager to, move up to larger-scale systems as soon as it becomes feasible to do so. Simulation-based Solar energy research focuses on the experimentalized nanosystems for materials used in photovoltaic or Solar chemical fuel generation. Analysis shows that to linear

scaling algorithms and exascale computing will be required to simulate "... a whole nanostructure device—from photon absorption and exciton generation, to exciton dissociation and carrier collection in a nanosize solar cell" (Stevens et al., 2008, p. 35). Finally, the ultimate goal of the simulation work of the nuclear energy research community is to develop a predictive model for the entire nuclear fuel cycle, from mining through the final disposition of waste material, taking into account interacting factors such as market forces, socio-political effects and technology risk. Exascale platform requirements of only a few different components of this visionary model can presently be quantified. For example, high-resolution nuclear fuel performance simulation of around 1 billion elements with a 1 μm scale size and with full physics (e.g. thermodynamics, neutronics, fluid flow, etc.) will require at least 1 exaflop performance to achieve viable throughput (Stevens et al., 2008, p. 42).

Biology. Although a much smaller fraction of life sciences is currently computationally mature, the introduction of large-scale genomic analysis is revolutionizing every area of biology, spurring a corresponding increase in the use of computationally intensive modeling, simulation, and analysis tools. In areas where computational

methods are already leveraging the wealth of new data, such as phylogenetic analysis, researchers already need increased access to high-end resources for more statistical testing, larger simulations, and advanced visualization tools (Stevens et al., 2008, p. 79). The computational requirements of this work increases superexponentially with number of terminals in the phylogenetic trees being analyzed. Other areas of evolutionary biology that were surveyed, such as the origin and evolution of phenotypes and the evolutionary dynamics of the phenotype/environment interface, are also moving inexorably toward computationally intensive methods that will require peta/exascale resources. Focusing primarily on systems-level biology for microbial life, and especially on issues of protein structure and function, the Town Hall report found that the use of computational simulation is beginning to generate ideas that provide productive new directions of biological experimentation. However, as in the previous study, the complexity of the problems involved (e.g. the multiplicity of dimensions and scales) and the overwhelming amount of data now available are pushing the field relentlessly toward larger scale computing resources. For example, it currently takes 10^6 CPU-days, on current terascale platforms, to generate the 100 phylogenetic trees per protein family that will be necessary (but not sufficient) to make accurate protein function predictions for members of the family, and the current estimate of the number of protein families is around 60,000 (Stevens et al., 2008, p. 50).

While the studies we have drawn on describe the high end computing requirements, current and projected, of only a small (but strategically significant) sample of disciplines, the underlying pattern of factors (e.g., complexity, multiplicity of scales, need to increase problem size, etc.) driving these fields toward peta/exascale computing holds much more broadly for science and engineering generally. There is every reason to believe that many fields will be moving their research toward peta/exascale resources as quickly as they become available. The motivation for our proposal for an international project plan a new software infrastructure for high-end computing revolves around a different question: Can we really make peta/exascale computing available to the wide spectrum of domain sciences that will want to use it any time soon? As we outline in the next section, significant technical barriers will have to be overcome to do so.

2.2 The Steepness of Ascent from Petascale to Exascale

There is every reason to believe that desire of the science and engineering community for more processing power and more computational resources will only continue to grow and spread, but there is also good reason to believe

that the HPC community is far from ready to satisfy it. Even though exascale systems are not projected to arrive until the latter half of the next decade, at the earliest, the technical challenges that have emerged as we cross the petascale boundary—systems designed with hundreds of thousands of nodes and millions of cores, with processors having reduced bandwidth and memory available to individual cores and including of special components, such as GPUs and accelerators—have effectively rendered the programming models and software infrastructure that we have been building for nearly 20 years obsolete. Since this looming obsolescence has its primary source in the radical changes now occurring in processor architecture, research being carried out at every level of the platform development chain, from the desktop and departmental cluster on up, is likely to be hobbled. Consequently, the scientific computing community, in general, falls under this shadow. Both the NRC and the Town Hall report acknowledge the urgency of the situation. A main conclusion of NRC report puts the point bluntly:

“The emergence of new hardware architectures precludes the option of just waiting for faster machines and then porting existing codes to them. The algorithms and software in those codes must be reworked. There do not yet exist productive and easy-to-use programming methodologies or low-level blocks of code [i.e. libraries, system software, and tools] that can take full advantage of multicore processors. Multicore parallelism is unfamiliar to many commercial software developers, and it also requires different sorts of parallel algorithm development.” (NRC, 2008, p. 124)

More detailed in its discussion and explicit in its recommendations, the DOE report conveys a similar sense of urgency about the emerging situation:

“Exascale computer architectures will require radical changes to the software used to operate them and the applications that run on them. The shift from faster processors to multicore processors is as disruptive to software as the shift from vector to distributed memory supercomputers 15 years ago. That change required complete restructuring of scientific application codes, which took years of effort. The shift to multicore exascale systems will require applications to exploit million-way parallelism and significant reductions in the bandwidth and amount of memory available to millions of CPUs. This ‘scalability challenge’ affects all aspects of the use of HPC. It is critical that work begin today if the software ecosystem is to be ready for the arrival of exascale systems in the coming decade.” (Stevens et al., 2008, p. 103)

These reports do not even take into account the full dimensions of the hardware and architecture innovations that we are likely to see as we approach exascale; these changes have only recently been fully detailed (Kogge et al., 2008). However, there is broad consensus on several factors that will necessitate the redesign and replacement of many of the algorithms and most of the software infrastructure that HPC has built on for more than a decade, including the following.

Extreme parallelism. There is little doubt that the modeling and simulation communities are poorly prepared to do their work on the peta/exascale systems of the future, where applications will only achieve the desired performance if they can exploit million- or billion-way parallelism. Since it largely preceded the multicore era, when performance improvements were largely faster clock speeds and increased instruction-level parallelism, the movement from terascale to petascale was relatively smooth, at least for the handful of applications that had been designed with this movement in mind. A thousand-fold increase in performance required only a ten-fold increase in thread-level parallelism. In fact, only a small number of applications today can scale to even 20,000 threads, and most science teams have no idea how to scale their applications to a million threads. A natural strategy is to have the scientific software community encapsulate disruptive multi-core issues, such as a three-order-of-magnitude increase in parallelism, inside the most popular numerical and communication libraries. With such peta/exascale-ready algorithms and components underneath them, applications would be able to scale to significantly larger numbers of threads with much less code restructuring. However, that software infrastructure does not exist today.

Tightening memory/bandwidth bottleneck. At the same time that fundamental physical limits on power and clock speed are forcing next-generation architectures to use multicore with ever more cores (“The core is the new transistor”), attempts to scale memory bandwidth continue to lag further and further behind. As the number of cores escalates into the millions, the amount of available memory per core shrinks, bisection bandwidth becomes increasingly expensive, and applications will hit the “memory wall”. The struggle to dramatically reduce communication costs will become much more intense, especially for applications that are more data intensive (Murphy, 2007; Moore, 2008). As in the case of parallelism, new algorithms and new software libraries will have to be developed that address the critical issue of communication costs, keeping them to a minimum wherever possible.

Necessary fault tolerance. Even making generous assumptions on the reliability of a single processor, it is clear

that as the processor count in high-end clusters grows into the hundreds of thousands and millions, the mean time to failure will drop from hundreds of days to a few hours, or less. Although today’s architectures are robust enough to incur process failures without suffering complete system failure, at this scale and failure rate, the only technique available to application developers for providing fault tolerance within the current parallel programming model – checkpoint/restart – has performance and conceptual limitations that make it completely inadequate for future peta/exascale systems, where routine component failure will be the norm. A new fault-tolerance paradigm will have to be developed and implemented in system software and libraries in order for future applications to use future peta/exascale systems with any reasonable degree of productiveness.

This short list of examples of the set of momentous technical challenges that software infrastructure builders must address to make peta/exascale systems truly available to the rapidly growing community of scientists and engineers who will need to use them is far from exhaustive. If tomorrow’s platforms are to deliver on their promise, new programming models, mechanisms for data I/O and tools for managing power consumption will all have to be incorporated into tomorrow’s computing environment. The goal of the IESP is to formulate a plan that takes those requirements into account and helps to organize and coordinate the sustained effort that will be required to bring it about.

2.3 The Leadership Role of the Open-source HPC Community

It is easy to see why the open-source community is ground zero for the software revolution that must occur on the path from petascale to exascale computing. Over the last 20 years, this community has provided much of the software infrastructure on which the World’s HPC systems, ranging from supercomputers to campus clusters, have depended for their performance and productivity. It has invested millions of dollars and years of effort to build most of the key components, including math libraries (e.g. LAPACK and PETSc), MPI, low-level performance counter interfaces (e.g. PAPI) for Linux operating systems, GNU tools, new languages (e.g. CoArray FORTRAN, UPC, and Fortress), and many others.

The vendor community has, no doubt, widely encouraged, leveraged, and in some cases contributed to this effort. However, current economic conditions and their short-term interests are not sufficiently aligned with the goals of high-end computing to enable them to lead the way across this divide. Despite its importance in driving the development of new technologies, the HPC market is so small that computing vendors have very little interest

in tackling the spectrum of challenges that software infrastructure development now presents. For that reason, new hardware innovations tend to be uncoordinated with the software changes that applications require to make use of them; the software that vendors deliver is only what is strictly specified as necessary to pass acceptance tests; and the open-source components that vendors deliver to make systems more generally usable represent only a “snapshot” of the component’s code tree, often stale by the time of delivery. Without advanced planning and coordination with the vendor, important value-added components from the open-source community (e.g. HPC Toolkit, optimized libraries, PAPI) can be *years* late.

Thus, as we confront a situation in which nearly all of the current the software infrastructure for scientific computing will have to be rethought and re-implemented from the ground up, the computational science community will have to depend on the global network of open-source software researchers and developers to do the vast majority of the work. The IESP has been formed in the belief that the scale, complexity and urgency of this historic challenge require a much higher level of coordination and cooperation within that community.

For example, although the different efforts that ultimately lead to the current HPC software stack were tremendously valuable, a great deal of productivity was lost because of the lack of planning, coordination, and the kind of integration necessary to make technologies work together smoothly and efficiently. Moreover, while open-source development within a single project can be coordinated by a repository gatekeeper and an email discussion list, there is no global mechanism working across the community to identify critical holes in the overall software environment, spot opportunities for beneficial integration, or specify requirements that require more careful coordination. It seems clear that this completely uncoordinated development model will not provide, in a timely way, the software that is *already* needed to support the unprecedented parallelism required for petascale computation, or the flexibility required to exploit new hardware features, such as GPUs. We are only at the beginning of this disruptive period of change.

The reasons for focusing the IESP plan on an *international* effort are also straightforward. In the first place, experience shows that the creation of a new, high-quality software stack for scientific computing, one which can meet both the diverse requirements of future applications and the rigors of peta/exascale hardware architectures, will demand investment on a scale that no single country can provide in the time required. To avoid significant disruptions in critical research agendas, we need to leverage the collective resources of the global community. Moreover, even leaving the magnitude of the investment required aside, the software infrastructure that must be

created is intended to serve a very broad spectrum of science and engineering communities, all of which are international in scope and need to be able to leverage resources at a variety of scales. To serve such transnational research collaborations, the IESP needs to proceed in a totally open manner and solicit input on requirements from computational science communities worldwide. Work on software infrastructure for peta/exascale science in other countries, especially in Europe and Asia, are already underway. We therefore have an ideal opportunity to make this work part of a larger vision for HPC.

3 IESP Phase 1: Workshop and Planning Process

The goal of the initial phase of the IESP is to provide essential support to advanced computational science over the next decade by developing a plan for:

1. a common, high-quality computational environment for peta/exascale systems and
2. catalyzing, coordinating and sustaining the effort of the international open-source software community to create that environment as quickly as possible.

The process for developing this plan centers around three international meetings: the first was held in Santa Fe, NM, on April 7 and 8; the second, in Paris, France, on June 28 and 29; and the third in Tsukuba, Japan, on October 18–20, 2009. All of the meetings were supported by funding from the DOE’s Office of Science and National Science Foundation’s Office of Cyberinfrastructure, as well as by international agencies and partners in Europe and Asia. In the following we discuss the current IESP leadership, workshop organization and development process, the projected features of plan/roadmap produced, and the strategy for post-plan follow up.

3.1 Workshop Leadership and Participation

Given the dimensions of the task before it, the IESP needs leadership with a proven ability to catalyze and manage such a community-wide undertaking and who are capable of eliciting enthusiastic participation from a wide range of well-networked participants. The IESP’s initial executive committee includes the following individuals:

Jack Dongarra, University of Tennessee/Oak Ridge National Laboratory, USA
Pete Beckman, Argonne National Laboratory, USA
Patrick Aerts, Netherlands Organisation for Scientific Research, the Netherlands

Franck Cappello, INRIA, France
Thomas Lippert, Jülich Supercomputing Centre,
Germany
Satoshi Matsuoka, Tokyo Institute of Technology,
Japan
Paul Messina, Argonne National Laboratory, US
Anne Trefethen, University of Oxford, UK
Mateo Valero, Technical University of Catalonia,
Spain

The meeting participants reflect the community that the IESP plan will need to draw upon to achieve its goals: representatives of the HPC software research community (both academic and government laboratories), vendors, application user communities, and relevant government agencies. We expect increased participation of the workshops by application community representatives as the series of workshops progresses.

It should be noted that although the initial list of IESP participants comes from the United States, Europe and Asia, we fully expect countries from outside the initial group to join the effort. Representatives from Russia have already expressed their intention to participate, and we anticipate wider participation from Asian countries (e.g., China, India) and South America will be forthcoming.

3.2 Anticipated Components and Achievable Outcomes of the IESP Plan

The overarching goal of the IESP plan is to dramatically improve the productivity and impact of computational science generally by creating a software environment that enables a wide range of applications to make *routine* use of the advanced systems now emerging and provides a path for the transition of the exascale systems that are likely to arrive by or before 2020. The point of the initial phase of the project is to develop this plan and try to forge a working consensus on it among participating national groups. Yet it is still necessary to have some minimal conception of its content in advance in order to structure and guide the discussions and the deliberative process. Based on the dimensions of the current peta/exascale problem space, as described above, and our background understanding of how the software community for scientific computing has traditionally worked (or failed to work), we anticipate that the plan that emerges from the IESP process will contain at least the following components:

- *Organizational framework for cooperative software infrastructure development.* The IESP will articulate an organizational framework designed to enable international software research community to work together to deliver more capable and productive HPC systems.

These efforts will aim to converge on a common software environment for the developers of research applications at various levels of the platform development chain. The framework will include elements such as initial working groups, outlines of a system of governance, alternative community models for shared software development with common code repositories, feasible schemes for incubating valuable software research and incentivizing its translation into usable, production-quality software for application developers, etc. The open-source community offers some outstanding examples of success on this front, such as Apache.org, which has nearly 800 “committers” contributing to a common code base. These communities also offer a valuable source of ideas for funding models that the IESP could incorporate. This organization must also foster and help coordinate R&D efforts to address the emerging needs of users and application communities on new platforms, such as Kraken and Blue Waters.

- *Thorough assessment of needs.* As part of its planning process, the IESP will thoroughly assess the short-term, medium-term and long-term software infrastructure needs of applications for peta/exascale systems. Any such assessment must include not only a comprehensive knowledge of the current state of the art (e.g. numerical libraries, operating systems, programming models), but also well-informed projections of science and engineering application requirements and future hardware platform developments. Thus, participation from representative application communities and vendors in the work of the IESP will be needed to help ensure the adequacy of these assessments.
- *Coordinated software infrastructure roadmap.* The IESP plan will initiate the development of a coordinated roadmap to guide open-source HPC development with better coordination and fewer missing components. The model for this effort is the International Technology Roadmap for Semiconductors (ITRS) assembled and regularly revised by a group of semiconductor industry groups in order to offer a well-informed projection of the future direction of semiconductor technology. We believe that such a software infrastructure roadmap would be extremely valuable, if not essential, in helping to guide both cooperative development and joint research efforts among globally distributed teams and partnerships. By coordinating with hardware platform projections supplied by vendors, such a roadmap would increase scientific productivity by helping to get the software community working on the infrastructure that new platforms require before they become available. The process of building and updating this roadmap should also tend to reveal gaps and missing components on future architectures so that they can be addressed in a timely way.

- *Joint programs in education and training.* The magnitude of the changes in programming models, software infrastructure and tools brought about by the transition to peta/exascale architectures will produce tremendous challenges in the area of education and training. A higher degree of commonality in the scientific software stack (e.g. common middleware libraries and system software), which the IESP aims to produce, can help to mitigate this problem; but it will certainly not solve it. The IESP plan should therefore provide for cooperation in the production of appropriate curriculum materials and education and training materials to be used at workshops and made available on-line.
- *Strategy for engaging and coordinating vendor community in crosscutting efforts.* To leverage resources and create a more capable software to support peta/exascale science, the IESP will engage and coordinate with vendors across all of its other objectives. Vendor participation in and contributions to all of these objectives—comprehensive application needs assessment, well-ordered but adaptive software roadmap, organized framework for cooperation, coordinated R&D programs for new exascale software technologies—will be encouraged and facilitated.

3.3 IESP Process: Community Input, Workshop Organization and Result Synthesis

The major nodes of the IESP phase 1 planning process consists of a series of three international meetings the project will organize, one each in the United States, Europe and Asia, beginning in the spring 2009. However, the productivity of those workshops relative to the plan that emerges depends not only on what occurs during them, but also on what occurs before and after them, i.e. on pre-workshop input from the community and post-workshop synthesis of issues and agreements. Consequently our strategy for the IESP provides for inter-meeting collaboration to facilitate community input.

The hub of the IESP's on-line collaboration activity will be the IESP wiki site, www.exascale.org. We are using exascale.org to help organize and structure each meeting, publish commentary and planning documents, and support interactive participation in the development of the organizational model, HPC software roadmap, and other elements of the IESP plan. Our aim is to make it easy for the global IESP community to contribute during all phases of the project.

Short white papers that express participant ideas or points of view on the key issues targeted by successive workshops are a key part of the preparation for each meeting. In getting ready for the first meeting, individuals and groups contributed 15 white papers that were both posted on exascale.org and distributed at the workshop. Not

only did these position papers inform the discussion that occurred at the meeting, more than half of them have been revised since the meeting and in preparation for the next round.

Meetings have a similar structure, although obviously the pattern may be changed to reflect new needs, if and when they emerge. An outline of the basic IESP meeting structure is as follows:

- Welcome and introduction.
- Plenary presentations focusing on the main IESP issues under examination at the meeting.
- Division of participants into breakout groups, which are charged to discuss, analyze and make recommendations relative to a defined set of issues.
- Participants reconvene from breakouts for reports from the breakouts and initial attempts to summarize and synthesize the recommendations for the IESP plan with respect to those issues.
- Discussion of next steps for the IESP planning process.

At the Santa Fe meeting, there were two breakout groups (see www.exascale.org). One surveyed on the current technical landscape that the IESP plan needs to address, examining the possibility of drawing up a technical roadmap for critical software infrastructure to meet the top challenges and significant opportunities of peta/exascale computing. The other explored potential organizational issues and opportunities that the IESP should consider in developing a model for international collaboration to create a common peta/exascale software stack. The summary results of those breakouts are available on exascale.org and have begun to be integrated into an initial planning document which, along with a new round of white papers on a different set of issues, will be ready for the next meeting.

The IESP workshops are being structured so as to provide progressively more definition for the components of the IESP plan, with each successive meeting building on the results of the previous meeting. The results of the first meeting are being collated, summarized and integrated into an initial draft version of the IESP plan, a straw man to stimulate comment and provide skeleton around which consensus can begin to form. That draft will be made available on the exascale.org site and, using the tools that the wiki offers, participants will be able to modify, comment, and amend the plan before the next meeting.

3.4 Possible Implementation Plan

While the IESP, working through its executive committee and workshop participants, is developing its plan and building consensus, it must frame a complementary plan

for implementation of rapid and long-term positive impact. Without trying to construct such a plan in detail, we believe that IESP should be guided by at least three general considerations. First it should aim for *immediate* payoffs; second, it should focus on developing early successes; and third, any new capabilities to be added to make scalable application development easier and more effective should be taken up after the first 2 years. One possible approach to realizing these objectives is as follows:

- *First 2 years.* To have maximum short-term impact, resources could be seeded to two groups: (1) to computer and computational scientists who can bring their prototype tools and libraries up to a level of maturity that facilitates the work of vendor partners and HPC facility staff in further development, integration and hardening; and (2) to application scientists working on critical applications, who would explore and adopt the new scalable programming models and tools necessary to help applications to achieve peta/exascale performance.
- *Future years.* Through the process beginning in the first two years, the IESP (or some future incarnation or successor) develops a far-sighted plan for the inclusion of software capabilities needed over the lifetime of the project. This plan may include activities to foster research and development aimed at delivering important future capabilities. The goal of this activity will be to reach beyond the nearest milestones to provide new capabilities that can make leading edge systems more broadly usable in the future.

3.5 IESP Timeline

2009	Schedule
April 7–8	Workshop 1, Santa Fe, NM, USA
June 21	First straw-man plan
June 28–29	Workshop 2, France
Aug. 15	Initial reports in summer
Aug. 15	Broad engagement by the community
Oct. 18–20	Workshop 3, Tsukuba, Japan
Nov.	Draft report for first year presented at Birds of a Feather session at SC09
Dec. 10	Plan presented for community comment
2010	Schedule
January	Begin to implement the IESP plan
Spring	First follow on workshop

4 Roadmap Plan

Working with the results of its application needs assessment, the IESP activity is initiating the development of a coordinated roadmap to guide open source HPC development with better coordination and fewer missing components. The model for this effort is the ITRS assembled and regularly revised by a group of semiconductor industry groups in order to offer a well-informed projection of the future direction of semiconductor technology. Nothing of this kind currently exists for the world of HPC software. Instead, today's investments in this area remain short-term in scope, with limited strategic planning and a paucity of cooperation across disciplines and international agencies.

We believe that such a HPC software infrastructure roadmap would be extremely valuable, if not essential, in helping to guide both cooperative development and joint research efforts among globally distributed teams and partnerships. By coordinating with hardware platform projections supplied by vendors, such a roadmap would increase scientific productivity by helping to get the software community working on the infrastructure that new platforms require before they become available. The process of building and updating this roadmap would also tend to reveal gaps and missing components on future architectures so that they can be addressed in a timely way.

Informed by the needs of science, engineering and humanities application requirements, this roadmap must address, at a minimum, the anticipated path of computing system hardware, networking, software, data management, and visualization. The roadmap must identify and prioritize the difficult technical problems and establish a timeline and milestones for successfully addressing them. It must identify the roles of government, academia, and industry. The roadmap must be assessed and updated at least every 5 years; ideally it should be treated as a living document that is updated more frequently, based on objective measures of performance and evolving need.

In general, the new computational science software roadmap would re-orient current support structures to address primary community goals, evolve new structures and components holistically, guide and coordinate future R&D investments, minimize technological disruptions, and create a sustained infrastructure and communication system allowing researchers and skilled individuals in many disciplines to work together. In addition, it would address the current acute shortage of educated and skilled people in the discipline.

The computational science software investment priorities should include, but not be limited to, the following areas:

- Software, including operating systems, libraries, compilers, software development, debugging and perform-

- ance analysis tools, software engineering, reliability, and serviceability.
- Numerical and non-numerical algorithms and software tools for solving complex, large-scale problems.
 - Infrastructure for computational science, including software sustainability centers, data repositories and analysis tools.
 - Data analysis, management, and discovery tools for heterogeneous, multimodal data, including business intelligence, scientific and information visualization, mining and processing capabilities.
 - Certain community applications in physical and life sciences, engineering, social sciences and humanities, Earth and atmospheric sciences, energy and environment.

Successful roadmapping generally involves planning, identifying needs, establishing process requirements and/or recommendations, and conducting periodic assessments of the roadmap itself. The roadmap should:

- specify ways to re-invigorate the computational science software community throughout the international community;
- include the status of computational science software activities across industry, government, and academia;
- be created and maintained via an open process that involves broad input from industry, academia and government;
- identify quantitative and measurable milestones and timelines;
- be evaluated and revised as needed at prescribed intervals;
- specify opportunities for cross-fertilization of various agency activities, successes and challenges;
- The roadmap should shape agency strategies for computational science;
- use strategic plans to recognize and address roadmap priorities and funding requirements.

With respect to software, our goal is to provide algorithms and software that application developers can reuse in the form of high-quality, high-performance, sustained software components, libraries and modules that lead to a better capability to develop high-performance applications, and develop a community environment that allows sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

The software roadmap should include the following:

Software language issues. A variety of languages are used for application development. There is a need to consider how best to support this mixed language environment to allow better code re-use.

Ease of use. Higher-level abstractions should allow application developers an easier development environment. The provision of efficient, portable “plug-and-play” libraries would also simplify the application developers’ tasks.

Support for development of software libraries and frameworks. More effective code reuse is essential. This could be achieved by supporting software library development and frameworks for reuse.

Validation of software and models. There are concerns from many application developers that there are no well-defined methods and techniques for validating scientific software and the underlying models. In some application areas observational data can play a role in validation, but for many this is not the case.

Software engineering. It is often the case that application teams developing scientific software are not as skilled in software engineering as would be desired. Guidance on best practices for software engineering development would be a step to assist the community.

Lack of standards. Identify and develop where needed.

Self adaptive libraries and code generation. In order to be able to move from one platform to another it would be beneficial to have underlying libraries that “do the right thing” for any given platform. This is becoming increasingly important with the plethora of new architectures that need to be considered.

Sustainability. There is general concern regarding the sustainability of application codes, software libraries and skills. There is a need to develop models for sustainable software that might include long-term funding, industrial translation and open community support.

5 Anticipated Benefits for Stakeholder Communities

Although the IESP in its first phase is primarily about planning, organization and consensus building, its focus is the software infrastructure that leading edge applications will require to run at high performance on new peta/exascale architectures. Accordingly we may distinguish between the anticipated benefits of the *planning and organizational process*, and the broader impact that the *success of the plan* could be expected to have. We believe that even in the case of the former, that benefits will be significant. As we have already argued, the dimensions of the problems we face in order to move the research community to peta/exascale platforms are

real, immense and still largely uncharted. Accelerating their community-wide discussion and initiating collaborative activities to address them would be an extremely important step. We strongly hope that our call to engage the international software research community in the IESP planning process will, in itself, help software researchers converge on critical problems for important user communities (e.g. prospective users of Blue Waters), focus collaborative efforts on solving those problems, and lead to better coordination of future software research and development.

The success of the plan that the IESP is designed to produce would have a very widespread and positive impact indeed. The organization that grows out of it would provide a complete software environment for scientific applications and the computational scientists who use them, offering them a model and software core from which to build their peta/exascale capable applications. In terms of impact that enhances infrastructure for research and for education, the myriad of components the IESP community would be concerned with are as important to the future of research and education as any facility, hardware system or instrument. Experience shows that positive effects of good software infrastructure are deep and pervasive. In assessing these beneficial effects, the following considerations are also worthy of notice:

- *Benefits for computational science education.* As noted above, the components of the IESP infrastructure will provide a common environment for the developers of research applications at various levels of the platform development chain. The IESP's common software stack will ensure that the same versions of middleware libraries are available at every level and that system software behaves similarly enough to make job flows, procedures, and scripts portable. This means that the interfaces on the systems on which students learn will move with them as their research changes and expands, moving them up the ladder of increasingly capable systems. The open-source nature of these components will also encourage participation, support and innovation from across the entire international community.
- *Benefits for national leadership facilities.* The success of the IESP planning and organizational efforts will mean that high-end computing sites in different countries will need to perform far less integration and testing of all the application-required software as they now do. The IESP consortium will supply a single point of contact for all software not developed by the vendor, rather than having to interface with multiple development groups. This will make the process of reporting and resolving problems more straightforward. It will also enable researchers (and their students) at smaller institutions to develop their applications on

local resources but position them to use large national systems, such as NSF's petascale systems, as they arrive.

- *Benefits for application developers.* With the right open-source organizational model, developers of science and engineering applications will benefit from having a single point of contact for both registering their requirements and for migrating their software innovations to leadership class systems; this pathway to leadership class architectures may also mean that some financial support will be available for the hardening and scalability testing that go beyond their normally funded research and development activities.
- *Benefits for vendors.* Vendors are an important but often unacknowledged part of research and education infrastructure. The IESP effort will also give vendors a partner in assembling the complete software environment needed in order for applications and scientists to have success on their machines. The IESP and its partners may eventually provide testing and integration services that they currently must share.

References

- Anderson, E. et al. (1999). LAPACK User's Guide, 3rd edn. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Berman, F. et al. (2001). The GrADS Project: Software Support for High Level Grid Application Development. *Int. J. High Perform. Appl. Supercomput.* **15**(4): 327–344.
- Blackford, S. et al. (1997). ScaLAPACK User's Guide. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Kogge, P. M. et al. (2008). ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems. Washington, DC: DARPA Information Processing Techniques Office, 28 September 2008.
- Moore, S. K. (2008). Multicore is Bad News for Supercomputers. *IEEE Spectrum Online*.
- Murphy, R. C. (2007). On the Effects of Memory Latency and Bandwidth on Supercomputer Application Performance. In IEEE 10th International Symposium on Workload Characterization (IISWC 2007), pp. 35–43.
- NRC (2008). The Potential Impact of High-End Capability Computing on Four Illustrative Fields of Science and Engineering. Washington, DC: National Research Council Committee on the Potential Impact of High-End Computing on Illustrative Fields of Science and Engineering.
- Petit, A. et al. (2001). Numerical Libraries and the Grid: The GrADS Experiments with ScaLAPACK. *Int. J. High Perform. Appl. Supercomput.* **15**(4): 359–374.
- Snir, M. et al. (1998). MPI: The Complete Reference, Volume 1, The MPI Core, 2nd edition. Boston, MA: MIT Press.
- Stevens, R., Zacharia, T. and Simon, H. (2008). Modeling and Simulation at the Exascale for Energy and the Environment report, Department of Energy Office of Advance

- Scientific Computing Research. Washington DC, <http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/Town-Hall.pdf>.
- Vadhiyar, S., Dongarra, J. and YarKhan, A. (2003). GrAD-Solve-RPC for High Performance Computing on the Grid. In 9th International Euro-Par Conference, Klagenfurt, Austria, 2003.
- Whaley, R., Petit, A. and Dongarra, J. (2001). Automated Empirical Optimization of Software and the ATLAS Project. *Parallel Comput.* **27**(1–2): 3–25.