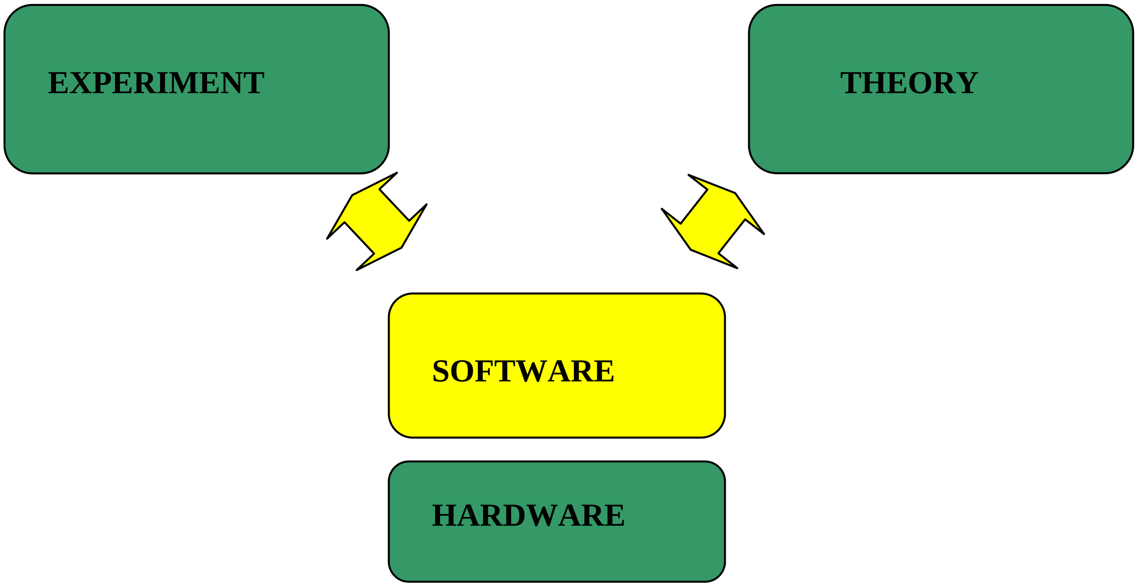
NSF Exploratory Workshop on Scientific Software Innovation Institutes:

Atomistic Modeling and Simulation



Final Report

September 2010

About the cover

Modern experiments have increased in accuracy and are beginning to explore very complex systems. Simple theoretical models are no longer able to provide a useful framework to interpret the results of these studies. To describe those experiments, theory in turn has become more sophisticated; to attain the required accuracy with theoretical calculations modern computers are indispensable. Software is now becoming the central switch where experiment, theory, and computer hardware connect to advance understanding, increase knowledge, and stimulate innovation. The “back of the envelope calculation” has been replaced by an iPhone connected to a supercomputer system like the NSF TeraGrid Cray XT5 *Kraken*.

NSF Exploratory Workshop on Scientific Software Innovation Institutes:

Atomistic Modeling and Simulation

Final Report

September 2010

Table of contents

Intended audience

- NSF program officers who need to decide on the form and requirements for Scientific Software Innovation Institutes (S2I2).
- Scientists and engineers who are users of computational modeling and simulation software in the general scope of atomistic modeling and simulation.
- Scientists and engineers, computer scientists and mathematicians who develop methods and implement them in software for the atomic modeling and simulation community.

Workshop Participants

The Workshop participants were chosen to represent a cross section of active investigators in the various scientific areas that were the target of the Workshop (including several with significant administrative experience), and also to represent areas of computer science that are relevant to software innovation. In addition to considering active workers known to the Workshop Chairs, assistance in identifying suitable participants was obtained from the leadership of the American Physical Society Division of Computational Physics, from a National Laboratory director, and from a number of senior investigators in scientific areas relevant to the Workshop. An effort was made to obtain geographic diversity and to secure adequate representation from under-represented groups. Because the call for the Workshop did not provide much lead time, several prospective participants had conflicting prior commitments and could not attend. Provision was made for their remote participation (see next paragraph).

Preparation for the Workshop and for this report was facilitated by establishing (as a Google Group) an on-line forum in which participants (and invitees who were unable to attend in person) could make observations about the Workshop and contribute to this report.

Workshop Chairs

Erik Deumens and Frank Harris

Workshop Participants

Paul	Ayers	McMaster University	ayers@chemistry.mcmaster.ca
Nelson	Beebe	University of Utah	beebe@math.utah.edu
Estela	Blaisten-Barojas	NSF CHE	eblaisten@nsf.gov
Daniel	Crawford	Virginia Tech	crawdad@vt.edu
Erik	Deumens	University of Florida	deumens@qtp.ufl.edu
Thom	Dunning	University Illinois UIUC	tdunning@ncsa.illinois.edu
Johannes	Feist	Harvard University	jfeist@cfa.harvard.edu
Evelyn	Goldfield	NSF CHE	egoldfie@nsf.gov
Frank	Harris	University of Florida	harris@qtp.ufl.edu
Robert	Harrison	Oak Ridge Natl Lab	rjharrison@gmail.com
Torsten	Hoefler	University Illinois UIUC	htor@illinois.edu
Srinivasan	Iyengar	Indiana State University	iyengar@indiana.edu
Jing	Kong	Q-Chem, Inc.	jkong@q-chem.com
Karol	Kowalski	PNNL	karol.kowalski@pnl.gov
Artem	Masunov	University of Central Florida	amasunov@gmail.com
Boyana	Norris	Argonne Natl Lab	norris@mcs.anl.gov
J. Vincent	Ortiz	Auburn University	jvo0001@auburn.edu
Manish	Parashar	NSF OCI	mparasha@nsf.gov
Bill	Poirier	Texas Tech University	bill.poirier@ttu.edu
Beverly	Sanders	University of Florida	sanders@cise.ufl.edu
H. Bernhard	Schlegel	Wayne State University	hbs@chem.wayne.edu
George	Shields	Bucknell University	George.Shields@bucknell.edu
Andrew	Taube	D.E. Shaw Research, Inc.	Andrew.Taube@DEShawResearch.com
Gary W.	Trucks	Gaussian, Inc.	trucks@gaussian.com
Ross	Walker	University California San Diego	ross@rosswalker.co.uk
Angela K.	Wilson	University of North Texas	akwilson@unt.edu

Workshop Electronic Forum Contributors

Garnet	Chan	Cornell University	gc238@cornell.edu
Hirata	So	University Illinois UIUC	hirata@illinois.edu
Valeev	Edward	Virginia Tech	evaleev@vt.edu

NSF Liaisons

Evelyn Goldfield and Estela Blaisten-Barojas, CHE, Manish Parashar, OCI

Executive Summary

The purpose of this Workshop was to assess the state of software development in the molecular and material sciences communities, examine future trends in computing technology, and determine whether a software institute is needed to ensure the ability of these communities to take full advantage of current and future advances in theory, computational science, and digital computation. If the answer to this question was in the affirmative, the Workshop was also asked to define the institute's activities and mode of organization and operation.

The scientific activities targeted by the Workshop include molecular structure and dynamics; atomic, molecular, and condensed matter physics; and their applications to materials science and engineering and to biochemical processes. The basic mathematical problem is that of solving the Schrödinger equation with the Coulomb Hamiltonian for nuclear and electronic degrees of freedom. The tools for solving this high-dimensional partial differential equation range from independent-particle and mean-field approaches to full many-body and strongly-correlated methods. Time dependence is treated in a variety of ways ranging from Monte Carlo (quasi-random) simulations to classical dynamics to more fully quantum-mechanical formulations. In this report we will designate this large collection of activities "atomistic modeling and simulation" (AMS).

This report conveys to the intended audience that there is a well-defined challenge in the AMS community that can best be addressed by cyber infrastructure created and maintained under the leadership and support of an NSF-funded scientific software innovation institute (S2I2) as envisaged by NSF Program solicitation 10-551. It further specifies the form of such an institute as well as critical elements of its operation. Finally, the report illustrates the importance of modeling and simulation in advancing the molecular and materials sciences and the value of the cyber infrastructure to be created to both the users and the developers of computational software in this community. Some issues addressed in this report are directly related to the subject of the forthcoming report from the NSF-ACCI Task Force on Cyber Science and Engineering expected to be released in the fall of 2010.

Challenge

Atomistic modeling and simulation (AMS) studies over the past 25 years have become an indispensable component of investigations in physics, chemistry, and biology. Chemists routinely use AMS to predict chemical structures and energetics, characterize reaction mechanisms, and otherwise interpret the results of experimental studies. AMS has become a powerful tool in the quest for the understanding of biochemical processes ranging from genetics and protein folding to drug design, and is poised to make seminal contributions to our understanding of condensed-matter physics and materials science. These contributions have been made possible by the development of high-speed digital computers and through the development of new theoretical models, new computational algorithms and sophisticated software. Until recently, AMS has progressed rapidly because of increases in computer clock speeds, achieved mainly by miniaturization of the essential computer components. However, physical constraints, such as the leakage current from nanoscale wires, are now preventing further increases in clock speeds. Increases in computing speeds are currently being obtained mainly by producing microprocessors with multiple compute cores.

This trend in computer technology has brought the AMS community to a point of urgent need. The effective use of modern computer architectures, which is essential for advancing AMS, requires new algorithms and software components, but the software tools needed for these efforts are not at an optimally useful level. Much of the existing software is poorly documented and there are no common data structures or input/output formats. Thus, it is nearly impossible to combine modules from different sources to solve comprehensive problems, and it is also difficult to use existing work as a starting point for the development of new computational methods.

The main issues posed by new computer architectures involve algorithm and method designs that take advantage of the parallel processing that is now possible, and which retain efficiency when the computer resources scale to hundreds, thousand, or millions of parallel processors. The lack of scalability and interoperability of most existing software are stifling progress and innovation, not only for AMS but for the overall scientific community that relies on these capabilities. At this point, many scientists cannot use AMS to explore their ideas with sufficient rapidity and precision to make progress either toward better fundamental understanding or to contribute to the solution of some of the grand challenges facing modern society. In order to make progress, software standards must be developed so that the community can develop interoperable tools that in turn allow its members to construct solutions to more complex problems by building on each other's work instead of being forced to duplicate it.

Outcomes from a Scientific Software Innovation Institute

The Workshop participants have concluded that a more powerful and coherent cyber infrastructure is needed to support AMS activities, and that this could best be provided through the formation of an S2I2 institute. The immediate goal is to increase innovation and productivity in the AMS development community, with the ultimate goal of also increasing the level of software support for the broad range of scientific activities that use AMS.

To be effective, the Workshop participants determined that the following are **critical activities** for an S2I2 institute:

- Provide both leadership and service to the community.
- Provide a general software infrastructure that will support the use of massively parallel computer architectures and facilitate communication between software modules developed by others.
- Provide liaison with related software development efforts, including both those that arise pursuant to the NSF software innovation program (the SSE and SSI initiatives) and those that arise from other research groups or in the commercial sector.
- Provide expert consultation as well as continuous opportunity for education and training.
- Provide visiting scholarships to enable individuals or small workgroups to work with experts at the institute.
- Maintain liaison with other S2I2 institutes to benefit from their work and to avoid unnecessary duplication of effort.

- Develop connections with other disciplines in science and engineering to identify areas of beneficial overlap.
- Lead efforts toward the development and adoption of software standards relevant to AMS.

Critical requirements that were identified as necessary to ensure the success of the S2I2 institute include:

- Locating it at an institution with adequate computing expertise, computer equipment, and hardware and network infrastructure, and which is not unduly difficult to reach by public transportation.
- Enthusiastic support from physical science, computational science, and computer science faculty at the host institution.
- Strong and effective support from the administration of the host institution.
- Budgetary support sufficient to provide a reasonable environment for the development of software infrastructure for emerging hardware architectures.
- A governance structure that is responsive to the needs of the AMS community.

The **expected outcome** of a sustained and successful community effort led by a scientific software innovation institute is:

- a new generation of AMS software that takes full advantage of the most advanced computing technologies (multi-core and many-core processors[1], high-end computers in NSF's portfolio, etc.)
- a robust cyber infrastructure to support rapid integration and/or development of new software tools to explore innovative ideas,
- a process of communication, education and training that can maintain and further develop this cyber infrastructure, and
- an effective education program to prepare a new generation of researchers who can move the AMS community forward.

[1] A multi-core processor (common now in general computer CPUs) consists of several wide-functionality processors per computer chip (presently typically 2, 4, or 6, but in the foreseeable future 16 or more). A many-core processor refers to a chip with a much larger number of processors of somewhat more limited capacity and originally designed for graphics processing units (GPUs); these can contain 256 cores in currently available chips and are expected to have of the order of a thousand processors.

1. Scientific and engineering need

There is a clear and well-defined need for better software for AMS. The following is a list of AMS problems that would benefit from efficient and reliable tools that can be used by experimental and theoretical scientists and engineers. A number of recent reports to the National Science Foundation [1], to the Department of Energy [2, 3], and from the National Science Board [4] show the general context of these problems and point out how they connect to the needs of the Nation to foster and realize the continued growth of prosperity.

Recent reviews, including those by Sherrill [5] and by Hirata [6], show the state of the art in AMS. Within the scope of AMS, theoretical models, computational methods and computer software exist that can solve many important problems routinely. However, some of the problems listed in the above named reports pose grand challenges; they cannot now be solved with sufficient accuracy using reasonable human and compute resources to contribute effectively to addressing serious problems facing our society. Consider the following specific grand challenges:

- Scientists and engineers who seek to design new materials need to have the ability to generate complete and accurate spectra of molecules and condensed matter, including electronic and vibrational or phonon excitations. A paper by Csaszar et al. [7] shows that even to obtain the complete vibrational spectrum of water up to dissociation is still far from a routine calculation; see sidebar on software usability challenge. Results from such calculations for much more complex systems need to be readily available to meet the challenges of materials design for targeted applications. The calculation of free energies, crucial in modeling biological processes, depends on accurate frequencies including anharmonic effects.
- Many industrial chemical processes are energy intensive. Using catalysts, they can be replaced by much more energy-efficient pathways. However, the

The hardware challenge

Petascale computers are here. Blue Waters, which will arrive in 2011, is based on the 8-core Power7 and has more than 300,000 cores, 1.2 petabytes of memory and 18 petabytes of disk storage. To take full advantage of Blue Waters, AMS software must scale to tens if not hundreds of thousands of cores as well as make optimal use of the memory and disks. Systems of this extreme size will have component failures every day, so software must be fault-tolerant and have error-correction capability.

For exascale computers in the coming decade the challenges are truly daunting. Exascale computers will have on the order of 10 to 100 million cores. Although memory and disk sizes will grow, they will not maintain the same ratio to compute cores as in Blue Waters.

Evolution of many-core* (as opposed to multi-core) GPUs is also making programming harder for software developers. Nonetheless, they are attractive compute engines. The nVidia G80 had no double precision (DP) and the GT200 had DP speed 1/8 that of single precision (SP), but in the new Fermi GPU there is only a factor two in DP/SP speed, similar to the ratio in a CPU. This GPU, with a peak performance of 0.75 teraflops, is ready for more general use.

Intel has now announced a MIC (many integrated core) chip comparable to the nVidia Fermi. The Intel MIC uses POSIX threads instead of a CUDA kernel and it has virtual memory capability. AMD is working on the Fusion Application Processing Unit with integrated x86 CPU on the chip as well.

None of this new hardware can be used effectively with most current AMS software, and adapting the software to the hardware represents a major challenge.

----

* For definitions of multi-core and many-core processors, see footnote, page 9.

ab initio design of catalysts is still beyond the reach of current computational approaches. In biological processes, enzymes play the role of catalysts that enable many important processes to take place. The modeling of enzymes is crucial for gaining a proper understanding of many of the basic processes in living organisms as well as to explore alternate industrial chemical processes.

- Accurate and reliable microscopic description of superconductivity, specifically high-T_c superconductors based on cuprates and iron pnictides. Although possible mechanisms of the superconducting electron-pair formation in these materials have been proposed, the details have yet to be confirmed and certainly cannot yet be accurately described [8].
- Accurate description of high-energy, radiation-induced ionization is possible for one-electron systems, but is presently difficult to impossible for molecules where vibrational dynamics plays a role during the process as well.

The evolution of hardware (see the sidebar on “The hardware challenge”) is such that existing software will need major modifications to fully utilize the new capabilities provided by this hardware. In the previous decades the increasing speed of processors resulted from an increase in processor speed concomitant with architectural innovation and with the doubling of transistors per chip every 18 months according to Moore’s Law. This increase in capability allowed the AMS community to tackle more complex problems. Because the hardware evolution did not change the basic computer architecture, the software could readily evolve with the hardware. But hardware is now evolving in a very different way—increases in hardware performance are now a result of increases in the number of compute cores on a microprocessor chip. This has induced a disruptive change in the design and implementation of the software used by the AMS community. If the AMS community wants to reap benefits from the increasing computational power, this software must be re-engineered or, in some cases, totally rewritten. In fact, the change is so disruptive that the community believes that new theoretical and computational methodologies must be developed, not just new software implementations of existing methods.

The software usability challenge

The software in use in the AMS community reflects the underlying methods and theories at a high level of detail. This makes it difficult and error-prone to tackle the complex problems that are now calling for a solution. This challenge is conceptual rather than superficial and cannot be addressed by a simple graphic user interface (GUI), or even by the addition of an artificial intelligence (AI) expert system to guide the user. Therefore software usability is really a reflection of the usability of the underlying models as well.

For example, performing a calculation of the vibrational spectrum of water should be a routine task. However, as is discussed in detail in a recent paper [7], there is a daunting list of choices that must be made to perform the calculation at all. It is difficult to make these choices in a way that yields sufficient accuracy without making the calculation so arduous that it can no longer be carried out.

The electronic structure calculation requires the choice of a good basis set: a high-end correlation-consistent basis is chosen. Then a method to describe the electronic correlation must be selected: internally contracted multi-reference configuration interaction with a renormalized Davidson correction. Test calculations must be performed to assess the size of the active space. These calculations must be possible and be reliably accurate on a grid of nuclear geometries. Then a discrete-variable representation method is used to compute the nuclear wave functions on the potential energy surface obtained with the above step.

The fact that readers of this report who are not themselves quantum scientists do not know what the above technical terms mean proves the point that many may consider the software to be user-hostile beyond what a GUI or some AI will be able to fix.

Thus, the AMS community has a clear software need that has been growing for some time, but has been brought to critical stage by the shift in hardware evolution from “faster cores” to “more cores.” To tackle the more complex problems mentioned above, it is vital to be able to use to the fullest the increased performance offered by the new hardware. To meet this challenge, the AMS community needs to work with computer scientists, applied mathematicians, and software engineers to train a new generation of scientists and engineers in parallel computing and modern software engineering and to develop a new generation of software that is easier to use and provides reliable answers to today’s and tomorrow’s complex problems.

The good news is that the foundation to rework the existing theories and develop new theories to solve the Schrödinger equation exists. The existing theories and their implementation in existing software provide a solid base of experience and a reliable reference for validation to overcome the challenge.   However, this cannot be accomplished by small groups of individual investigators working in a random network of collaboration.   Although innovation is still expected to most often come from individual research groups and their students, to make progress an infrastructure must be put in place that will allow everyone to build efficiently on the work of others. This is where the S2I2 program of NSF will play a critical role.

Examples of challenges

A more extensive but still incomplete list of challenges in the scope of AMS is the following:

- Accurate descriptions of the electronic structure of increasingly complex molecules and materials, including extended and condensed-phase systems (solids, surface reactions, explicit solvation)
- Vibrational structure and dynamics of many-body systems
- Molecular dynamics of complex systems, including propagation on multiple excited surfaces
- Interactions with intense lasers (strong-field chemistry)
- Exploration of potential energy surfaces (structures, reaction paths, seams of intersection, conical intersections, conformational searching)
- Excited states (condensed and gas-phase spectroscopy, linear, non-linear and multi-dimensional spectroscopy)
- Ion and electron scattering (molecules and surfaces)
- Electron transfer (in renewable-energy research and in biological processes)
- Materials design, simulations for chemical engineering, drug design
- New algorithms to handle the complexity of biological systems and materials (length and time scales, sampling, rare events, coarse graining)
- Protein-protein interactions, protein folding, self-assembly of biological systems
- Intermolecular forces (accurate and cheap calculation of weak interactions like van der Waals and hydrogen bonding)

2. Institute Requirements

NSF has started a coordinated and long-term effort to build software infrastructure to assist the science and engineering communities, including the AMS community, in addressing the challenge precipitated by the change in direction of the hardware evolution from faster cores to many cores, many threads, and likely, many graphics processing units (GPUs). The solicitation for 2010 invited the creation of small groups building Scientific Software Elements (SSE) and small collaborations creating Scientific Software Integration (SSI). The final element of the new program is the creation of a small number of Scientific Software Innovation Institutes (S2I2). This report summarizes the findings of a Workshop organized for the AMS community to study and define the requirements for such an institute as well as the form an institute should have to effectively serve the needs of this community.

2.1. Simplify the conceptual framework

The conceptual framework of software for AMS must be simplified both for the user and the developer. The methodology to accomplish this exists and has proven to work in the past two decades in numerous complex software projects in industry and government: The use of **object oriented** design and the development of **standards** are the main ingredients of this successful approach to the control of software complexity.

Quantum Chemistry Program Exchange

The QCPE was started around 1960 with the mission to: (a) make computer programs from a variety of sources available through a single repository, and (b) provide a suitable interface between the developer and user communities to facilitate wider dissemination of computer codes. The project started with 3 years of funding from the US Air Force, secured by Harrison Shull. The funding was for 1 postdoc, who would manage all the codes coming in. The person would spend half the time managing QCPE and the other half performing research in Shull's group. Responsibilities included understanding (undocumented) codes in the repository and conveying to the users how such codes can be best utilized. This was in addition to making the software available to all.

Right from the start, QCPE was designed to be a self-sustaining organization, sustained through nominal charges ($35 per "download") to the user community. As it turned out, federal funding ran out after year 3 and never got renewed. Indiana University continued to support QCPE until it was disbanded, due to lack of funding, in the 1990s.

QCPE did not dictate standards for its programs or the data they generated, and this ultimately diminished the usefulness of the project. However, the most important development that contributed to the shutdown of QCPE was the advent of direct peer-to-peer FTP (file transfer protocol) technology in the 70s, where many of the users could directly download from developers and the utility of a mediating organization became fairly limited.

The design of software that can take full advantage of many compute cores on a chip (Intel's Nehalem, IBM's Power7 or NVIDIA's Fermi chips) or a compute system built using hundreds of thousands of compute cores, hundreds of petabytes of memory and tens of thousands of disk drives (Blue Waters) is a daunting task. **A major goal of the institute is to gather the needed expertise to enable the AMS community to create software that scales to thousands and to hundreds of thousands of cores, while making optimum use of the other resources available (memory, disks, etc.)**.

Method developers need to be presented with a conceptually simpler view of available methods and software components so that they can creatively use them to solve new, more complex problems. The development of the BLAS, MPI and LAPACK libraries has proven to be an enormous success. Application developers have a standard way to call a powerful and feature-rich set of software tools to solve linear algebra problems. Hardware and library developers at the same time have a clear set of software routines that they know are worth the effort of optimization because a large community of developers and much user software depends on them directly. Thus everyone benefits. **The development of the standards and a set of tools and building blocks in the form of a library of software components suitable for the community is one of the critical roles of the AMS institute.**

Finally, although this is a highly idealized goal, users of AMS software need to be able to specify a requested absolute accuracy of results they seek to compute, for example the vibrational spectrum of the water molecule. They also need to be able to obtain an estimate of the accuracy that can be obtained with a given set of computational resources. The software should report the results and in addition make available the methods and approximations that were used to obtain them with the requested accuracy. **The coordination of the method development that will lead to this capability is one of the expected requirements of the AMS institute.**

2.2. Leadership role

Previous efforts to create software infrastructure have not lived up to expectations; see side bars on “Quantum Chemistry Program Exchange” and “National Resource for Computational Chemistry”. The Workshop participants felt that these efforts fell short because, although they provided service for individual investigators, they lacked a more active channel through which the community could provide leadership and direction. The role of the individual investigators is crucial to provide innovation in new concepts and models. **The role of an institute should be to provide infrastructure that will both facilitate innovation and allow its fruits to be rapidly deployed to the entire AMS community to accelerate and stimulate further innovation.** The governance structure of the institute must carefully balance the inherently conflicting requirements of providing leadership while allowing individual innovation to flourish. One way to accomplish this is by having a well-functioning communication system that works both ways, from the institute to the community and from the community to the institute.

National Resource for Computational Chemistry

Connected to the early success of QCPE (in the 70s), there was the development of a new organization, “a software institute,” to be funded by the DOE and NSF. To convey the pressing need for such an institute, there were several workshops conducted in the late 70s by: (a) Argonne, (b) Los Alamos, (c) Lawrence Berkeley Lab, (d) Utah, and (d) UC/Santa Cruz. A NRC panel also supported the concept. Important ideas discussed during the Argonne workshop included the development of a “quantum chemistry library.” Lawrence Berkeley Lab eventually got funded to create the NRCC, but it took a different direction, with the provision of computer capacity becoming the primary focus.

The institute should have a single physical location, where the majority of the permanent staff resides and where visitor and training programs are located, but the institute must also have strong ties to multiple distributed developer sites.

The current stage of software development for the transition to massively parallel machines is in some way analogous to the historical transition from coding in machine language on early computers to the development of compilers and languages like Fortran. The community gave up some execution speed in exchange for a large increase in ease of coding and for a much greater independence from specifics of the hardware. The hand-crafted massively-parallel implementations of some programs using MPI (message-passing interface) have been compared to coding in machine language. **The institute will need to lead in coordination with other S2I2 efforts to define the new high-level software development framework that will make programmers of massively-parallel computer systems more productive.**

A large collection of existent software tools enables scientists and engineers to compute numerous properties within the AMS scope. However, these software tools together are not capable of solving any of the grand challenge problems listed in the previous section. **A new synthesis of the various software tools must be forged, possibly with significant changes of the underlying algorithms to build a new coherent and significantly more potent software environment that can successfully address the grand challenges.** Such synthesis cannot be accomplished by a loose collection of individual investigators, but requires the coherent effort of the entire AMS community as well as collaborations with mathematicians and computer scientists. To guide this effort the leadership and coordination fostered by a well-designed software institute is crucial.

PSI4: Building software from components

PSI4 is an example of the need for new software: How did its developers make the decision that they needed a new software package with some radical changes?

The PSI3 Quantum Chemistry Suite started in 1977 at Berkeley in the Schaefer group and was renamed PSI in 1987 when he moved to Georgia. In 1990 it was ported to UNIX as PSI2 by Curtis Janssen and Ed Seidl. In 2000 David Sherrill, Ed Valeev, and Daniel Crawford started PSI3. It is written in C and C++, had a new integral code, CI, CC, and explicit correlation. It is available under GPL from SourceForge.

PSI3 has no infrastructure to take advantage of massive parallelism. All parallel development was ad hoc, non-systematic, and individually written for each module. For example, existing infrastructure made implementation of new density functionals or Cholesky decomposition of two-electron integrals overly complicated. This made it difficult to implement new methods and put too much chemistry out of reach.

Thus, the development of PSI4 was started with focus on a single executable environment most appropriate for massively parallel computing, with as design goals a fully integrated parallel execution stream, the avoidance of a need for rearrangement of data, and a fully object-oriented structure necessary for extending, sustaining, and reusing software. A C++/Python interface enables flexible user-defined computations such as surface scans and the building of composite methods. Its new tensor libraries include reduced-scaling approaches.

New infrastructure developments are possible and easy because of object-oriented design and open-source licensing: e.g. PSI4 uses the communicator code from MPQC, another package.

2.3. Required language in the NSF solicitation

The Workshop participants understand that the NSF solicitation will have to be formulated with sufficient generality that it will apply to any S2I2, not just one focused on the AMS community. It is recommended that the solicitation specify that the proposal must:

- Define and describe the mission and goals of the institute as well as how these goals will be achieved.
- Identify how the host institution will provide the necessary physical and personnel infrastructure to host the center and allow it to meet the above requirements.
- Explicitly describe the governance structure and the staffing of the institute, including the processes by which the different individuals and groups interact and communicate.
- Outline the communication mechanisms between the institute and the communities it serves.

- List measures of success and the means to measure that they are being achieved, including the process by which this measurement will take place and how the results will be judged and evaluated by the community.
- Outline the mechanism by which the community can redirect the institute to steer back to its agreed-upon goals when the evaluation shows this to be necessary.
- Specify how the institute will provide resources to the community. It is anticipated that every institute will have local control over a moderate-size compute resource such as a cluster with a few thousand cores as well as access to systems with novel hardware such as a cluster of nodes, each equipped with one or more GPUs or FPGAs, and that this hardware will be available (mainly for development activities) to users from the community. In addition, the proposal should specify what partnerships the institute has negotiated with national resource providers such as the NSF TeraGrid to ensure that developers in the community it serves will have an easy way to get quick access to top-of-the-line resources like Blue Waters for code development, validation, and performance tuning.
- Specify the license for all software created with support from the institute and show how it allows both wide dissemination to stimulate innovation and integration of commercial software companies into the developer community.

3. Institute Activities

It should not be the mission of the institute to create a monolithic software package that will be all things to all scientists. Rather, its activities should be designed to support software innovation that arises through the efforts of talented individuals, collaborations between small numbers of such individuals, and major software development projects. The appropriate role for an institute, then, is to cause rapid development of the infrastructure needed to foster innovation by:

- Developing general software tools that support computation on parallel platforms.
- Increasing the awareness of physical scientists about optimal software development processes and available tools, done through training, outreach, and consultation activities.
- Developing and encouraging the use of standards that will promote the interoperability and sharing of use of specialized software.
- Identifying and responding appropriately to unmet software infrastructure needs.
- Taking an active role in studying the use of innovative computer architecture for AMS problems (e.g., the use of graphics processor units for compute-intensive parallel computing).
- Improving communication and cooperation within the AMS development community, including its commercial members.

A detailed description of the way in which it is proposed that these activities will be carried out should be specified in proposals, but Workshop participant consensus opinion is that they could include elements such as:

- User meetings, to identify successes, failures, and possible new directions for the focus of effort by the institute's management and staff.
- Newsletters, web pages, or blogs that provide information to (and from) current and potential users.
- Training activities at levels suitable for students from the users' research groups.
- Topical workshops that have a direct effect in stimulating software development both by the institute's staff and by user groups.
- Workshops (in collaboration with appropriate computer scientists) on approaches that use innovative hardware components.

Institute use case

An individual researcher who is the PI of a grant for a project that involves development of software may have the following interaction with the institute and its members:

A graduate student of the PI develops a new method and implements it in functional software. The student starts by using wherever possible available components from the standards-based library made available and supported by the institute. Early on, the PI and the student contact the institute and possibly take a training class or workshop on software design and development. A permanent staff developer at the institute is assigned to assist with the design of the new software.

Then the student completes the work and finishes a thesis showing the value of the new method. After this validation is complete the software is handed to the developer at the institute for testing and quality assurance. Because the design was correct from the start, this work will not involve a complete rewrite, but rather becomes a maintenance task. The new software can readily be made to fit in the standards-based community library and framework. Finally, if interest and demand is sufficiently high, the component is released as an integral part of the community library and framework quickly after the publication of the work. Thus it is available to the community as a new building block.

- A visitor program that strengthens and benefits from relationships both with physical science users and with relevant members of the computer science community.
- Maintenance of one or more databases on data valuable to the AMS community such as available software tools and environments and theoretical methods, computational results for data mining or for validation and calibration of new methods, expertise of members of the community, etc.
- Management of development access to compute resources for members of the AMS community, including those of modest size at the institute location and others provided as national resources by the NSF TeraGrid and the DOE Leadership Computing Facilities. To develop state-of-the art software that scales optimally, continuous access to the real hardware is needed and the institute should manage an allocation at the TeraGrid for developers. This will simplify the management of access for both the individual research groups and the TeraGrid.
- The institute should play a key role as a provider of education in scientific computing to the next generation of scientists and engineers in the AMS community, a role that is notably absent from a majority of graduate education programs in our universities. This role will provide an important complement to the institute's leadership toward defining standards and catalyzing robust and high-performance implementation of innovative methods for AMS. This combination of infrastructure development and education is in good alignment with NSF's mission.

The benefit of standards

In recent years, Todd Martinez and his student, Ivan Ufimtsev, developed a compute kernel for evaluating *s* and *p* two-electron integrals efficiently on GPUs. If there was a standard for calling integrals in the AMS community, the benefit of the fast integral module would be available to all software in the community by simple relinking, as is the case with BLAS routines.

However, correctly packaging the library is a relevant issue: The communication between GPU and CPU over the PCIe bus is too slow to perform only the integral evaluation on the GPU. The current version of the software also assembles the Hamiltonian matrix on the GPU.

An important and proven tool to advance progress of a community is the development of standards. Everyone recognizes the benefit of the BLAS standard for linear algebra. A computational kernel that has been suggested as a good candidate in the AMS community is the evaluation of integrals; see side bar "The benefit of standards". Standards have been proposed [9], but their widespread adoption requires a critical mass of interoperable software that adheres to the standard and can easily be extended by researchers to explore and implement their own ideas.

Obviously computer science and applied mathematics (especially in numerical analysis and algorithm development) must be deeply involved in this activity. To obtain maximum benefit from the effort, the scope should be a broad as possible, including quantum chemistry, electronic structure with ab-initio and semi-empirical Hamiltonians for molecules, clusters and condensed matter, molecular dynamics with force fields and ab initio potentials, classical and quantum dynamics, electron scattering and ionization processes, etc. However, other fields where many-body methods are crucial, such as nuclear physics, may also need to be included in the effort to optimize return on investment for NSF, to ensure that standards are developed with sufficiently broad scope, and to avoid unnecessary duplication of software creation. The inclusive nature of the institute activities, encompassing physics, chemistry, mathematics and computer science and engineering, is nowhere more important than in the education and training mission. Without a comprehensive plan to address effectively the education and training needs of the AMS community the institute requirements listed in Section 2 cannot be satisfied. The institute must show leadership to guide the active researchers in the community to a common goal, but it must also provide the infrastructure to provide students the education they need to play an effective role in this effort.

Many contributions from the different fields of physics, chemistry, mathematics and computer science and engineering will be required to find the correct mixture of theory, model, algorithm, and software implementation that

- produces results with accuracy and reliability useful and acceptable to physicists, chemists, and materials scientists,
- has the convergence and stability properties that allow mathematicians to assure reliable results, and
- executes efficiently, using a significant fraction of available resources, on massively parallel computers employing methodology developed by computer scientists and engineers.

To be an effective and productive player in such collaborations, students must be educated on the basic concepts and language of their partners. Thus chemistry students must learn about algorithms, object-oriented software engineering, and modern programming languages and libraries. Computer science students and mathematicians need to be educated about the role of basis set expansions in solving partial differential equations with large numbers of dimensions.

Most likely this cannot be accomplished by a few graduate-level classes. A comprehensive framework of background material must be made available online so that it can be accessed anytime during the student’s study. The material must be organized in a thoughtfully designed way to be supportive of the institute’s multi-disciplinary effort. A simple encyclopedic collection of topics will not lead to the desired learning process. Workshops, schools, and visitor programs must be organized to provide focused and intense learning environments to jumpstart the process of mastering the extensive relevant material.

4. Organizational Structure

The workshop participants had many discussions on how the institute should function. The following roles were considered relevant and important to the success of a software institute as a long-term structure and activity:

Advisory board

- broad range of people
- some from industry
- initial selection of advisory board members should be in proposal
- maybe some board members elected and others appointed by director
- representation from user group
- some connection to host University's administration

Group of principal investigators, local and remote

- provides sense of direction
- explores, clarifies, defines, and expresses the wish of the community
- at least monthly conference calls and quarterly meetings in person

Forum for standardization

- open for everyone to join, like MPI and OpenCL forums
- should meet regularly, at least once per quarter
- has a travel budget for participants

User group

- holds regular meetings
- maintains email list

Visitor program

- coordinated with workshops and school
- for students, postdocs, and faculty
- for individuals and groups focused on a special topic

PhD level staff

- should be allowed to have an independent research activity and professional development activity to make the position attractive
- faculty affiliation in suitable department

Support staff

- for administrative work such as for workshop and visitor programs
- coding and web development for online education and training activities

Policy for resource management and allocation

- manage access to TeraGrid resources for development and debugging
- committee for review of proposals from community members to use local or remote compute resources, attend or hold workshops, schools, and visits

The institute should have an explicit plan to build and manage relationships with other S2I2 institutes, SSE and SSI projects, as well as with other national resources such as TeraGrid and DOE and DOD facilities for hardware access and for software collaboration and synergy.

5. Institute Resources

The institute must have a central location where the bulk of the staff is co-located. This is considered crucial to allow the professional staff to have synergistic collaborations. It also makes the institute a desired destination for short- and long-term visits by researchers, students, and research groups. Thus, facilities to accommodate visitors must be available with easy access to travel to and from the institute location. Training facilities must be on site in addition to a web presence for distance learning and software and training dissemination.

The institute staff must include PhD-level personnel with the background and experience needed to provide effective support to the necessary software infrastructure development and use.

The institute must own and operate local compute resources of midrange capacity with easy access as well as early-access special machines for exploration by the institute and its community.

In addition the institute should have local machines with novel architectures such as a small cluster with GPUs in every node for exploratory development.

For larger compute projects, the institute should leverage the NSF funded TeraGrid and other resources managed and operated by federal and state entities and universities. The access to these resources should be negotiated by the institute on behalf of the community members, so that they do not have to go through a lengthy process to get allocations for access. These allocations are primarily for development access with the need for quick turnaround on many small jobs.

Members of the AMS community who want to carry out big computational projects should secure a separate allocation for their project using the established TeraGrid allocation process.

References

- NSF report NSF-2007-28: “Cyberinfrastructure Vision for 21st Century Discovery”, Cyberinfrastructure Council, March 2007, http://www.nsf.gov/pubs/2007/nsf0728/nsf0728.pdf .

1. DOE report: “Directing Matter and Energy: Five Challenges for Science and the imagination”, Basic Energy Sciences Advisory Committee, Co-Chairs G. Flaming and M. Ratner, December 2007, http://www.er.doe.gov/bes/reports/files/GC_rpt.pdf.
2. DOE report: “Opportunities for Discovery: Theory and Computation in Basic Energy Sciences”, Subcommittee on Theory and Computation of the Basic Energy Sciences Advisory Committee, Co-Chairs B. Harmon, K. Kirby, C. W. McCurdy, January 2005, http://www.er.doe.gov/bes/reports/files/OD_rpt.pdf .
3. NSB report NSB-09-55: “Building a Sustainable Energy Future: U.S. Actions for an Effective Energy Economy Transformation”, National Science Board, August 2009, www.nsf.gov/pubs/2009/nsb0955/nsb0955.pdf .
4. S. Hirata, Quantum chemistry of macromolecules and solids. Phys. Chem. Chem. Phys. **11** (2009) 8397-8412.
5. C. D. Sherrill, Frontiers in electronic structure theory. J. Chem. Phys. **132** (2010) 110902.
6. A. G. Csaszar, E. Matyrus, T. Szidarovszky, L. Lodi, N. F. Zobov, S. V. Shirin, O. L. Polyansky, J. Tennyson, First-principles prediction and partial characterization of the vibrational states of water up to dissociation, J. Quant. Spectroscopy and Radiative Transfer **111** (2010) 1043-1064.
7. Jian-Xin Zhu, Rong Yu, Hangdong Wang, Liang L. Zhao, M. D. Jones, Jianhui Dai, Elihu Abrahams, E. Morosan, Minghu Fang, and Qimiao Si , Band Narrowing and Mott Localization in Iron Oxychalcogenides $La_2O_2Fe_2O(Se,S)_2$, Phys. Rev. Lett. **104** (2010) 216405.
8. J. P. Kenny, C. L. Janssen, E. F. Valeev, and T. L. Windus, Components for Integral Evaluation in Quantum Chemistry, J. Comp. Chem. **29** (2008) 562.

List of Abbreviations

AI	artificial intelligence
AIMD	ab-initio molecular dynamics
AMD	Advanced Micro Devices, computer chip manufacturer
AMS	atomic modeling and simulation
CC	coupled cluster
CI	configuration interaction
CPU	central processing unit
CUDA	compute unified device architecture, development environment for GPU programming
DFT	density functional theory
DMFT	dynamic mean-field theory
DMR	density matrix renormalization
DP	double precision
FFMD	force field molecular dynamics
FPGA	field-programmable gate array
FTP	file-transfer protocol
GPL	GNU Public License
GPU	graphics processor unit
GUI	graphical user interface
MBPT	many-body perturbation theory
MD	molecular dynamics
MIC	many integrated core, many-core chip made by Intel similar to a GPU
MPI	message passing interface
NRCC	National Resource for Computational Chemistry
PB	Petabyte
PCIe	Peripheral Component Interconnect Express, a computer standard
QCPE	Quantum Chemistry Program Exchange
QMC	quantum Monte-Carlo
QSAR	quantitative structure-activity relationship
RAM	random access memory
S2I2	scientific software innovation institute, proposal category in SI^2 solicitation
SP	single precision
SSE	scientific software elements, proposal category in SI^2 solicitation
SSI	scientific software integration, proposal category in SI^2 solicitation