# Los Alamos Computer Science Institute

## Statement of Work for Academic Participants

The *Los Alamos Computer Science Institute (LACSI)* was created to foster internationally recognized computer science and computational science research efforts relevant to the goals of Los Alamos National Laboratory (LANL). LACSI is a collaborative effort between LANL and the Rice University Center for Research on High Performance Software (HiPerSoft), along with its partner institutions: The University of Houston (UH), the University of Illinois at Urbana-Champaign (UIUC), the University of New Mexico (UNM), and the University of Tennessee at Knoxville (UTK).

LACSI was founded with the following goals:

- To build a presence in computer science research at LANL commensurate with the strength of the physics community at LANL,
- To achieve a level of prestige in the computer science community on a par with the best computer science departments in the nation,
- To pursue computer science research relevant to the goals of High Performance Computing (HPC) programs at LANL, and
- To ensure that there remains a strong focus on high-performance computing in the academic computer science community.

In keeping with these goals, LACSI researchers engage in joint high-performance scalable computing research and in collaborative activities that foster a strong relationship between LANL and the participating academic institutions.

In the following section, we present the vision, implementation strategy, and tasks associated with LACSI projects at Rice and its partner academic institutions. In the final section, we describe the management and administrative plan for the LACSI academic partners.

# 1. Strategic Thrusts

In March 2002, the LACSI Executive Committee met with LACSI researchers at LANL to discuss methods of addressing issues raised in the 2001 LACSI contract review. The body was tasked to develop priorities and strategies to meet LANL's future programmatic and computer science needs. The group developed a framework to address long-term strategic thrust areas. Specific objectives were called out as near-term priorities. The objectives were folded into the framework to form a coherent planning view. A description of the long-term vision, framework, and objectives is available in a draft document titled *Priorities and Strategies*. The framework outlined five strategic thrust areas:

- Components
- Systems
- Foundations of Computational Science
- Application Performance
- Computer Science Community Interaction

The following five subsections describe the vision, implementation strategy, and tasks associated with the academic LACSI projects that fall under the five strategic thrust areas.

## 1.1. Components: Component Architectures for Rapid Application Development and Composition in a Networked Environment

### Investigators: Ken Kennedy, Zoran Budimlic, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, Charles Koelbel, Dan Reed, Jaspal Subhlok, Qing Yi

The goal of the component architectures effort is to make application development easier through the use of modular codes that integrate powerful components at a high level of abstraction. Such integration systems can be viewed as providing a "software backplane" into which various components can be plugged to add new strategies for solving fundamental problems such as hydrodynamics, mesh generation, and transport.

Through modularization and the existence of well-defined component boundaries (specified by programming interfaces), components allow scientists and software developers to focus on their own area of expertise. For example, components and modern scripting languages allow physicists to program at a high level of abstraction (by composing off-the-shelf components into an application), leaving the development of components to expert programmers. In addition, because components foster a higher level of code reuse, components provide an increased economy of scale, allowing resources to be shifted to areas such as performance, testing, and platform dependencies, thus improving software quality, portability, and application performance.

A fundamental problem with this vision is that Los Alamos application developers, and most others in science, cannot afford to sacrifice significant amounts of performance for this clearly useful functionality. Therefore, an important part of the effort will be the exploration of integration strategies that perform context-dependent optimizations automatically as a part of the

integration process. This theme defines a significant portion of the research content of the work described in later subsections.

The overarching goal of this activity is to develop component architectures that can be used to support rapid prototyping of portable parallel and distributed applications and rapid reconfiguration of existing applications. These architectures would be the basis for frameworks for applying advanced compilation techniques, run-time system elements, and programming tools to prepare applications for execution on scalable parallel computer systems and distributed heterogeneous grids.

To succeed, this effort will need to accomplish two long-term goals.

1. It must develop *frameworks for integrating existing components* rapidly and conveniently into complete applications. These frameworks must be able to produce efficient applications from scripts within reasonable compile times. In addition, they must be able to integrate components written in different languages, particularly Fortran and object-oriented languages like C++ and Java. Finally, the frameworks must support the generation of applications that execute with reasonable and reliable efficiency in a distributed computing environment.
2. It must develop a *collection of components for use in science and engineering applications.* This collection should be ideally suited for use in the rapid prototyping frameworks developed as part of this activity. The algorithms must be general, portable, and usable in a variety of situations.

### 1.1.1. Component Frameworks Review

**Investigators: Ken Kennedy, Jack Dongarra, Richard Hanson, Lennart Johnsson, Charles Koelbel**

We will assist the Los Alamos staff in conducting a study of available component frameworks. This will be done to support the goal of integrating high-performance component technology into strategic LANL applications and into the LANL software culture. A team of both LANL staff and academic partners will carry out this survey. This team will establish the requirements for component architectures that fit the needs of the Los Alamos code-development environment. This effort should be viewed as a precursor to a code project that would update a major hydrodynamics application to a more modern and maintainable form.

In conducting the study, the team should draw, to the maximum extent possible, on recent advances in the design and implementation of complex software systems. The approach should attempt to understand the value and impact of incorporating new languages, such as Java and Python, and new compiler strategies for addressing the problem. In addition, the connection to the Common Components Architecture (CCA) effort should be a major consideration. In examining recent advances, the team should consider the ease of adapting existing codes to take advantage of new software technologies. Issues such as language interoperability and the ability to create components out of existing codes should be carefully considered.

***FY03 Tasks:***

- Assemble team and establish LANL-specific requirements for component architectures (Quarter 1).

- Initiate review of existing component architectures (Quarter 2).
- Pick one or two of the most promising frameworks and test them on a range of code types.
- Produce report and recommendations (Quarter 4).

### 1.1.2. Component Architectures for Problem Solving Environments in a Networked Environment

**Investigators: Ken Kennedy, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, Dan Reed, Jaspal Subhlok**

#### *Generation of Problem-Solving Systems Through Component Integration*

The goal of this project is to develop compiler technologies and library designs that will make it possible to automatically construct domain-specific development environments for high-performance applications. This project will develop advanced compiler technology to construct high-level programming systems from domain-specific libraries. Programs would use a high-level scripting language such as Matlab to coordinate invocation of library operations. Scripting languages typically treat library operations as black boxes and thus fail to achieve acceptable performance levels for compute-intensive applications. Previously, researchers have improved performance by translating scripts to a conventional programming language and using whole-program analysis and optimization. Unfortunately, this approach leads to long script compilation times and has no provision to exploit the domain knowledge of library developers.

To address these issues we are pursuing a new approach called “telescoping languages”, in which libraries that provide component operations accessible from scripts are extensively analyzed and optimized in advance. In this scheme, language implementation would consist of two phases. The offline translator generation phase would digest annotations describing the semantics of library routines and combine them with its own analysis to generate an optimized version of the library, and produce a language translator that understands library entry points as language primitives. The script compilation phase would invoke the generated compiler to produce an optimized base language program. The generated compiler would (1) propagate variable property information throughout the script, (2) use a high-level “peephole” optimizer based on library annotations to replace sequences of calls with faster sequences, and (3) select specialized implementations for each library call based on parameter properties at the point of call.

We will use this strategy to construct a high-level application development environment for an application of interest to LANL based on Matlab. This system could be seen as a flexible replacement for POOMA.

We also plan to extend these programming systems to prepare applications for execution on computational grids. If this effort is to succeed, it must take into account two important realities. First, many components will be constructed using object-oriented languages, so techniques for optimizing such languages are critical. Second, the execution environments for the resulting programs are likely to be distributed, so the implementation must take into account the performance implications of distributed systems, even if the applications are compiled together. For these reasons, basing a significant portion of the work on the Java programming language

makes sense. Java is portable and includes distributed computing interfaces. However, we must overcome one major drawback of Java if it is to be used in scientific computation, namely its less-than optimal performance. Although we intend to focus on Java, many of the strategies developed for Java will extend to other object-oriented languages such as C++.

With these considerations in mind, we plan to pursue research in two important directions:

*Toolkits for Building Problem-Solving Systems:* This effort will focus on the production of tools for defining and building new domain specific PSEs, including:

- Tools for defining and building scripting languages.
- Translation of scripting languages to standard intermediate code.
- Frameworks for generating optimizers for scripting languages that treat invocations of components from known libraries as primitives in the base language.
- Optimizing translation of intermediate language to distributed and parallel target configurations.
- Tools for integrating existing code.
- Demonstration of these techniques in specific applications of interest to ASCI and LANL.

An important goal of this effort is to make it possible to build highly efficient applications from script-based integration of pre-defined components. Building on the component architecture efforts described above, we will pursue the novel strategy of "telescoping languages" to make it possible to extend existing languages through the use of software components.

***Component Architectures for Integration:*** This effort will focus on the design and specification of components that can be used in a PSE for high-performance computation. Significant issues will be flexibility and adaptability of the components to both the computations in which they are incorporated and the platforms on which they will be executed. In addition, these components must have architectures that permit the effective management of numerical accuracy.

***Tasks:***

*FY03:* Prototype telescoping languages framework based on Matlab delivered to LANL, demonstrated on a PSE derived from an application domain to be selected in collaboration with LANL.

*FY04:* Experimentation with PSE framework on a LANL application area.

### *Application Development Support for Distributed Computing*

The key challenge in building grid applications is to construct applications that are adaptive to changes in the execution environment and that can detect and correct performance problems automatically. In this activity, we will explore the meaning of network-aware adaptive applications and what the implementation and optimization challenges are for such applications. In addition, we will pursue research on middleware to support optimal resource selection in grid environments. The work will proceed in two major subactivities:

***Compilation of Configurable Object Programs:*** This research will explore the challenges in compiling programs to be dynamically configurable in the sense described previously. It will explore programming models and their translation to efficient collections of tasks that can be targeted to a variety of Grid computing platforms. The long-term goal is to support component-based implementation of applications for grids using PSEs of the sort described in the previous section. A critical technology will be adaptivity to changing performance characteristics of the components of a distributed computing platform. This adaptivity will require reoptimization and

migration of components and data. This work will amplify the NSF-funded GrADS effort and apply it to applications of interest to LANL. A significant component of this work will be the development of performance models and mapping strategies that can be used to automatically retarget applications to different parallel and distributed computing platforms.

***Frameworks for Building Adaptive Network Applications:*** This project will perform fundamental research and develop middleware for "optimal" resource selection in shared distributed environments. The key challenge is predicting the performance of an application on a set of nodes on a shared network with dynamically changing resource availability. The planned framework for resource selection will have the following main components:

- Development of the notion of a *performance signature*, which represents application resource requirements and predicts application performance under different network condition such as different speed links and processors and different competing loads and traffic.
- Development of a *resource graph* representation for the topology and status of currently available resources such as CPU and memory at nodes, and latency and bandwidth on network links.
- Development of algorithms for *graph-based node selection*, which map an application represented by its performance signature to the network represented by a resource graph.

***FY03 Tasks:***

- Demonstration of an application preparation and execution using the GrADS framework.
- Demonstration of a tool to generate the topology of simple networked clusters, including link capacities, traffic, and sharing.

### 1.1.3. Rapid Performance Prototyping of Algorithms

**Investigators: Ken Kennedy, Zoran Budimlic**

The goal of this sub-project is to develop tools for the rapid prototyping of algorithms and models for use in complex large-scale scientific simulation codes. The approach will be to capitalize on two successful efforts—one at Los Alamos and one at Rice University—involving the use of the Java programming language as a basis for rapid prototyping tools. Java has a number of attractive features for the development of rapid prototyping tools, including full support for software objects, parallel and networking operations, relative language simplicity, type-safety, portability, and a robust commercial marketplace presence leading to a wealth of programmer productivity tools.

***Compilation of Object-Oriented Languages:*** To make it possible to move rapidly from prototyping to high-performance application development, compilation strategies must be developed for object-oriented languages such as Java and C++. This should include interprocedural techniques such as inlining driven by global type analysis and analysis of multithreaded applications. This work would also include new programming support tools for high-performance environments. Initially, this work will focus on Java, through the use of the JaMake high-level Java transformation system developed at Rice. This system will include two novel whole-program optimizations, "class specialization" and "object inlining," which can improve the performance of high-level, object-oriented, scientific Java programs by up to two

orders of magnitude. Later we will consider extensions to other object-oriented languages. This effort will be undertaken in collaboration with the CartaBlanca group at LANL.

***Tasks:***

*FY03:* Test an automatic differentiation tool for Java supplied by Rice (Quarter 3). Demonstrate Rice JaMake compiler on Los Alamos particle code. (Quarter 4). Refine the JaMake high performance compilation system for Java and apply to components of the LANL CartaBlanca project on rapid prototyping of algorithms.

*FY04:* Deliver the JaMake framework for use at Los Alamos.

### 1.1.4. Retargetable High-Performance Applications

**Investigators: Ken Kennedy, Jack Dongarra, Lennart Johnsson, Charles Koelbel, John Mellor-Crummey, Qing Yi**

For many years, retargeting of applications for new architectures has been a major headache for high performance computation. As new architectures have emerged at dizzying speed, we have moved from uniprocessors, to vector machines, symmetric multiprocessors, synchronous parallel arrays, distributed-memory parallel computers, and scalable clusters. Each new architecture, and even each new model of a given architecture, has required retargeting and retuning every application, often at the cost of many person-months or years of effort.

Unfortunately, we have not yet been able to harness the power of high-performance computing itself to assist in this effort. We propose to change that by embarking on a project to use advanced compilation strategies along with extensive amounts of computing to accelerate the process of moving an application to a new high-performance architecture.

To address the problem of application retargeting, we must exploit some emerging ideas and develop several new technologies.

- First, we will exploit the recent work on automatically tuning computations for new machines of a given class. Examples of effective use of this approach include FFTW, Atlas, and UHFFT. The basic idea is to organize the computation so that it is structured to take advantage of a variety of parameterized degrees of freedom, including degree of parallelism and cache block size. Then, an automatically generated set of experiments picks the best parameters for a given new machine. This approach has been extremely successful in producing new versions of the LAPACK BLAS needed to port that linear algebra package to new systems.
- Recent work on cache-oblivious recursive algorithms has shown that it is possible to use recursion to avoid the need of selecting block sizes for memory hierarchy. Furthermore, work by faculty at Rice has established that excellent recursive algorithms can be automatically constructed by advanced compiler technologies, especially transitive dependence and iteration-space slicing. This work has succeeded in blocking LU decomposition without pivoting and simple matrix multiplication, achieving performance at least as good and usually much better than hand-blocked versions. In addition, it has generated a credible blocking for LU decomposition *with* pivoting, something that was until recently thought to be

impossible in a dependence-based system. This powerful analysis can be used to generate versions of numerical algorithms that are portable across a variety of architectures.

- New work is needed to extend the work on automatic tuning to arbitrary codes, so that programmers are not forced to laboriously convert each application to a format that is amenable to the Atlas-style experimental tuning strategy. Here we propose to develop a conversion tool that uses deep compiler analysis to translate critical loop nests to a form where automatic tuning can be effectively applied. This strategy will involve extensive use of symbolic strip-mining of loops. In addition, it will require exploitation of a new strategy called *overpartitioning* that can be used to provide flexible mapping of grid computations to machines of different configurations. The basic idea of overpartitioning is to tile a computation to a size that represents a minimum computation that makes sense to carry out on a single processor. There will usually be many more tiles than processors as a result, so each processor will be responsible for carrying out the computations associated with several tiles. The strategy for grouping tiles would be selected for each new architecture based on extensive tuning experiments, much in the way that Atlas tunes the BLAS.
- Finally, we will extend the Atlas technology so that it can gather key parameters of target hardware platforms and optimize selected kernel or software components across a range of platforms.

We propose to conduct research on the topics described in the previous sections and to use the results of this effort to construct at least one retargetable application of interest to DOE and the ASCI program.

***Tasks:***

*FY03:* Demonstrate loop-restructuring technologies on a LANL code. Prototype state space generation and exploration for simple kernels. Preliminary investigations into most, if not all, of: (a) generalization of ATLAS's configure routines, (b) code generation for GEMV and GER, (c) code generation for some Level 1 routines, (d) SMP support via pthreads, (e) packed storage optimizations, (f) wide and narrow band optimizations, (g) sparse optimizations, and (h) algorithmic tweaks and higher-level routines. Distribute the recursive LU, Cholesky and QR software.

*FY04:* Demonstration of kernel tuning on a benchmark application.

### 1.1.5. Numerical Component Libraries

**Investigators: Lennart Johnsson, Jack Dongarra, Richard Hanson**

The principal goal of this effort is to develop libraries of components that can be effectively used in rapid prototyping systems of the sort that this project proposes to produce. Key considerations will be production of scientific computing components that can be used reliably on a variety of computation platforms, especially parallel and distributed systems. This will entail establishing an architecture that provides a high degree of flexibility while maintaining efficiency and robustness across a broad spectrum of computational configurations. Important subactivities include:

- *Standards:* Initiate and lead community effort towards an interface standard that allows for the creation of efficient implementations of common library routines for equation solvers, BLAS, etc. on a broad spectrum of architectures and heterogeneous distributed environments.
- *Adaptable Libraries:* Creation of polyalgorithmic, parameterized library functions that allow for the run-time selection/optimization of algorithm selection and scheduling based on the problem at hand and the execution environment.
- *Fast Algorithms:* Several key ASCI applications are based on mathematical models requiring evaluation of interparticle interactions. We propose to develop library support for use of multipole and multipole-like O(N) methods in ASCI applications.
- *Algorithms for Graphics and Visualization:* Included are algorithms for the manipulation and visualization of extremely large data sets, generation and refinement of unstructured simplicial grids, along with multiresolution analysis over those grids, data management via wavelets, and feature definition and extraction.

***Tasks:*** The five-year outline of the project consists of four mutually dependent tracks:

- Development of new algorithms.
- Development of code generation and optimization infrastructure.
- Generation and implementation of adaptive software libraries on different parallel architectures.
- Demonstration of use of software libraries in important applications.

*FY03:*

- Initiate an attempt to develop a community consensus for APIs for FFTs.
- Release of a FFT library for MPI applications supporting complex-to-complex, real-to-complex, and complex-to-real transforms.
- Release of a FFT library for threaded applications supporting complex-to-complex, real-to-complex, and complex-to-real transforms.
- Development of a single processor adaptive library for Sine and Cosine transforms.
- Performance modeling for FFTs.
- Release of a code generation and optimization toolbox for FFTs.
- Demonstration of the parallel complex FFT library in at least one application.
- Release of a Sine and Cosine library for MPI applications.
- Release of a complex FFT library for applications using MPI+threads.
- Release of an OpenMP interface for the FFT library.
- Release of out-of-core FFT transforms.
- Performance modeling for complex and real transforms.
- Development of a code generation and optimization toolbox for finite element problems.

*FY04:*

- Demonstration of the Sine and Cosine library in at least one application.
- Release of a Sine and Cosine library for applications using MPI+threads.
- Release of an OpenMP interface for the Sine and Cosine library.
- Release of a Lagrangian finite element library of arbitrary dimension and order for simplexes and hypercubes for MPI applications.
- Development of a code generation and optimization toolbox for Legendre and Spherical transforms.
- Development of a code generation and optimization toolbox for finite difference (convolution) problems.
- Release of a library for Legendre and Spherical transforms for MPI applications.
- Release of a library for Legendre and Spherical transforms for threaded applications.
- Demonstration of the Lagrangian finite element library in at least on application.

*FY05–FY07:*

- Release of a Lagrangian finite element library of arbitrary dimension and order for simplexes and hypercubes for threaded and MPI+threads applications.
- Release of an OpenMP interface for the Lagrangian finite element library.
- Performance modeling for the finite element library functions.
- Demonstration of the Legendre and spherical transform library in at least one application.
- Release of a library for Legendre and Spherical transforms for applications using MPI+threads.
- Release of an OpenMP interface for the Legendre and Spherical transform library.
- Performance modeling for parallel spherical transforms.
- Release of a library for finite difference (convolution) problems.

- Demonstration of the finite difference (convolution) library in at least one application.
- Performance modeling for the finite difference library.

## 1.2. Systems

### Investigators: John Mellor-Crummey, Ed Angel, Tom Caudell, Alan Cox, Rob Fowler, Guohua Jin, Ken Kennedy, Arthur Maccabe, Vijay Pai, Scott Rixner

### 1.2.1. Compiling high-level languages for advanced networks.

#### Investigators: John Mellor-Crummey, Rob Fowler, Guohua Jin, Ken Kennedy

Current methods for programming high-performance applications are still mired in an explicit message-passing style. While portable, the software is incredibly complex and error-prone, leading to low programmer productivity. We are developing compiler and operating system support for programming languages, such as Co-Array Fortran (CAF), based on explicit but higher-level constructs. Such languages enable programmer expression of algorithms in a more domain-specific way while allowing the programmer to retain flexibility and control.

Under other funding, we are currently developing a Co-Array Fortran compiler on top of an infrastructure derived from the Open64 compiler. This project will leverage that effort by extending the compiler to generate code to directly exploit performance and functionality for a variety of network interfaces that are important to LANL and ASC. These include the Quadrics communication sub-system, LA-MPI, and the asynchronous messaging system described in Section 1.2.2.

This project has extensive interaction with, and will provide infrastructure for, the "Multiprocessor Performance" sub-project described in Section 1.4.1 of the Application Performance thrust.

***FY03 Tasks:***

- *Q1:* Design target API for the Rice Co-Array Fortran compiler to several network interfaces.
- *Q2:* Design code generation enhancements for the compiler to use the API.
- *Q4:* Demonstrate code generation technology for data-parallel languages.

### 1.2.2. Efficient Zero-copy, Zero-mapped Asynchronous I/O

#### Investigators: Scott Rixner, Alan Cox, Vijay Pai

This project's objective is to demonstrate a new approach to zero-copy I/O. Previous attempts to build zero-copy I/O systems have generally fallen into one of two categories: those supporting zero-copy I/O transparently, using existing (Unix) interfaces, and those introducing new interfaces. Neither approach has seen widespread acceptance. Getting programmers to adopt new interfaces is difficult and has slowed the acceptance of newly proposed interfaces. Transparent approaches have traditionally been based on copy-on-write: the pages containing the I/O buffers are write-protected until the I/O is completed. This approach has problems as well. The VM manipulations are expensive, especially on an SMP; copies may still occur, if the

application writes into the pages containing the I/O buffers before the I/O completes. Our approach addresses all of these problems as follows.

We support the POSIX-standard asynchronous I/O interfaces. This has two benefits. First, much of the earlier work on zero-copy I/O has been done in the context of synchronous I/O. Modern high-performance applications, however, often use asynchronous I/O. Second, these I/O interfaces provide a way to transparently implement zero-copy without copy-on-write. In particular, they contain primitives for notifying an application that the I/O is complete and that the I/O buffers can be reused. Such primitives are essential because the application must avoid modifying the buffers between the time the I/O is initiated and the time that it is notified that the I/O is completed. If the application obeys this protocol, which is inherent in asynchronous I/O, then no copy-on-write is necessary.

An important optimization is the elimination of virtual memory manipulations that occur in conventional implementations of zero-copy I/O. These implementations map the buffers into the kernel address space before the physical I/O operation is initiated. With conventional network interfaces, this mapping is necessary to allow the kernel to perform operations such as check summing. With modern interfaces that offload such functions, there is seldom, if ever, any need for the kernel to touch or modify the data, and thus there is seldom any need to map the data in the kernel's address space. In those cases, the kernel can perform I/O straight from user space, without an intermediate mapping in the kernel. Although the pages still need to be pinned in memory for the duration of the I/O, this operation does not involve the TLB and the page tables, and is therefore much less costly, especially on an SMP.

***FY03 Tasks:***

- *Q1:* Design asynchronous I/O system.
- *Q4:* Demonstrate prototype implementation using LA-MPI.

### 1.2.3. High-performance Communication Based on Commodity Protocols

**Investigator: Arthur Maccabe**

The goal of this research is to investigate high-performance implementations of traditional and specialized communication protocols on commodity network fabrics. In recent years, there has been a marked trend toward the use of commodity components in building high-performance computing systems. Starting in the mid 1990's, Beowulf systems demonstrated that many computationally intense applications could be run on systems constructed from commodity components. While many applications can be run effectively on systems with commodity communication networks, most of the large, general-purpose computational clusters are based on specialized communication networks (e.g., Myrinet) to deliver the communication performance needed by other applications.

Identifying the commodity point in networking can be difficult. While 100 Mb Ethernet is certainly a commodity networking technology, it can be (and has been) argued that Myrinet and Gigabit Ethernet also represent commodity-networking technologies. To avoid this difficulty, we define commodity networking in terms of the protocol layer implemented in the network fabric. In particular, we focus on IP networks: networks that route traffic using the Internet Protocol.

Traditionally, communication performance has been defined in terms of bandwidth and latency. While these metrics are still important in the definition of communication performance, recently CPU utilization has emerged as another important metric. CPU utilization measures the percentage of the host processor cycles that are used in the implementation of a communication protocol. As network speeds have increased, percentage of processor cycles needed to implement communication protocols has also increased. In many instances, the bandwidth that can be attained is limited by the speed of the processor.

Knowing that processor cycle rates have increased at an astounding rate, it may be surprising to find that CPU utilization has also increased for high-speed networks. In the five year period from 1995 to 2000, processor cycle rates increased from approximately 100~MHz to approximately 1~GHz, a tenfold improvement. In the same time period, the bandwidth provided by high-performance networking technologies increased from 10~Mb/s to 1~Gb/s, an improvement of one hundred fold.

***Commodity Hardware:*** In commodity systems, the interface point between the network fabric and computational node is a network interface card (NIC) on the PCI bus. Using the PCI bus offers the benefits associated with commodity components, but introduces two significant problems. First, the interrupt latencies on a PCI bus are fairly high, 10-15 microseconds.

Second, access to main memory through the PCI bridge is fairly slow; latencies are on the order of a microsecond. It should be noted that the bandwidth of newer PCI buses is on the order of 4 Gb/s and is not significant problem for the current generation of gigabit networking technologies.

If we assume a gigabit networking fabric and a packet size of 1500 bytes, the inter-arrival time between network packets is 12 microseconds. When we also consider the 10-15-microsecond interrupt latency for the PCI bus, we can see the makings of a significant problem. Typically, this difficulty is handled by coalescing interrupts on the NIC. The NIC holds interrupts (and data packets) until either enough packets have arrived or until enough time has lapsed since the oldest packet arrived. Note: while this strategy avoids thrashing the PCI bus, it does not address the issue of CPU utilization.

The delay in access to primary memory through the PCI bus means that the data structures used to manage communications must be maintained on the NIC. The fact that the NICs have a limited amount of memory means that most of the local memory will be used to buffer packets and little is available for sophisticated data structures.

***Preliminary Results:*** Many vendors provide programmable NICs. Rather than providing a single function in hardware, these NICs have a processor that executes a control program. The control program on the NIC can be altered to implement different communication protocols or to improve the implementation of a communication protocol. While most vendors do not provide any information regarding the structure of their NICs, several vendors, including Myricom and Alteon, provide development support for writing alternate control programs. In a preliminary study, we wrote a control program for the Alteon Acenic, Gigabit Ethernet NIC. The default control program simply moves IP fragments between the host processor and the network. The altered control program implements IP fragmentation and reassembly on the NIC. Using netperf, we measured UDP delivery rates of 960 Mb/s for both control programs. However, while the CPU utilization for the default control program was approximately 80%, the CPU utilization for our control program was closer to 60%—a significant performance improvement.

In addition to the problems with the PCI bus indicated earlier, the processors on the NICs are slow. The speed of the NIC processor has a direct impact on the complexity of the control program. If we assume a 1500 byte packet, the inter-arrival time between successive packets on a gigabit network is 12 microseconds. If we further assume that the processor is able to complete the execution of an instruction on every cycle, this means that the control program only has 1200 instructions to complete the handling of a packet before the next packet arrives.

***Interactions with LANL:*** Regular interactions with Ron Minnich and other staff members of CCS will be critical to the success of this effort. Professor Maccabe and other members of the SSL will make regular visits to LANL (weekly visits during the summer and bi-weekly visits during the regular semesters). Whenever possible, we will also try to arranged to have a member of the SSL spend a summer in residence at LANL.

***FY03 Tasks:***

- A report describing functionality that could be offloaded in an implementation of TCP. In particular, we will consider offloading specific features like ACK generation and window management.
- Performance results obtained from offloading TCP features. Here, we will focus on traditional measures like bandwidth and latency as well as metrics like processor availability.
- A report describing how to shunt all OS functions to a single processor in an SMP system. This report will describe the measurement techniques used to evaluate the effectiveness of OS shunting.

### 1.2.4. Hierarchical Visualization of Large Data Sets and Visualization on PC Clusters with Programmable Pipelines

**Investigator: Ed Angel**

Much of the efforts in high performance computing have been oriented towards developing simulation codes. These codes generate huge data sets, which pose many problems in how they can be interpreted. Scientific visualization can provide at least part of the solution to these problems (which range from storage of the data sets to finding areas of importance in the data).

We are interested in methods that can be used interactively, probably in a hierarchical manner. Such a visualization would display information in a manner in which humans could decide which areas need detailed exploration and which can be discarded with a high probability that no significant information will be lost. We are also interested in methods that can be ported to a variety of computing/visualization environments and also be used for remote or shared visualization activities. We are working with methods that use clusters and can be used in conjunction with the physical simulation, removing the need for storage of large amounts of raw data. On the algorithmic side, we are working with particle systems. We have developed an object-oriented particle system, which has been used both for numerical simulations and visualization. Future work in this area will be on adding intelligence to the system so that it can be used to search out "interesting" areas in large data sets and communicate among them to keep the computational effort within predetermined bounds. The data sets with which we have been working have been supplied by Patrick McCormick from LANL's Advanced Computing Laboratory and are from the global climate simulations being run at LANL.

On the hardware side, we have been working with a heterogeneous SP/Intel cluster (Vista-Azul) from IBM that is connected to a Scalable Graphics Engine (SGE). The Intel side of the cluster has a graphics board at each node of four processors. Vista-Azul allows us to experiment with a variety of rendering/simulation strategies, including running a simulation and visualization at the same time, thus avoiding the need for data storage. We have implemented sort first and sort last renderers in both hardware and software. We have also ported the Flatland virtual environment to the system. The SGE contains 16 Mpixels and can be configured by software in a variety of ways, such as a single 4K x 4K display or as multiple smaller displays. Likewise, we can obtain the output from the SGE in a variety of ways. The SGE is connected through a SP switch, hence access is very fast, However, it is only a frame buffer and cannot do computation or even read-backs.

Nevertheless, this unique configuration has generated a lot of interest in the large-scale data visualization community of which Los Alamos, Sandia, and Lawrence Livermore National Laboratories are major players. In particular, the SGE provides a straightforward way to drive a power wall or a high-density display such as the new IBM display in which DOE has shown great interest. We have been discussing joint projects with all three DOE labs. Mutual areas of interest center on rendering for such high-resolution displays. Open issues have to do with the difficulties of transferring high-resolution data to the SGE for display. For example, in multiframe images, how can we minimize the data transferred by compression algorithms? Can sort last work for high-resolution displays?

We have put together a commodity cluster of dual processor Intellistations with Nvidia G Force 3 cards connected by a gigabit ethernet. This cluster is located in the Scalable Systems Laboratory so that the graphics/visualization students can work with Professor Maccabe's high performance networking and systems group. One reason for this is that our recent work on Vista-Azul has confirmed what others are observing, which is that network issues can be the dominant factor in large-scale rendering. We hope to be able to surmount many of these problems by getting the two groups working together.

In addition, some of the features of the G Force 3 cards, such as pixel shaders and a programmable pipeline, open up many new algorithmic possibilities for visualization. Recently, NVidia has announced its new MV30 architecture, which represents a major architectural advance. This next generation of graphics processors will also have the ability to do numerical calculations and will support standard control structures. Thus, we will be able to build clusters in which the simulation, data analysis, and graphics can be on the graphics processor. In addition, the graphics architecture will lead to new algorithms in areas beyond graphics and visualization. We have been discussing a number of projects involving these processors with LANL, SNL, and LLNL. As these processors become available, probably in the first two quarters of FY03, we will replace the G Force processors and do our projects with the new architecture.

***FY03 Tasks:***

- Implement particle system used for feature extraction on ocean circulation data set.
- Benchmark cluster using G Force cards.

### 1.2.5. Studies of Visualization Tools for Viewing Dynamic Program Interaction in Parallel Systems

**Investigators: Thomas P. Caudell, Kenneth Summers**

Scientific computer users wanting to do numerical computation are often presented with programming interfaces that were designed for expert computer programmers. These interfaces tend to assume a large body of programming knowledge on the part of the user, knowledge the scientist does not have and does not wish to learn. These interfaces are often text based. Even large, integrated development environments are centered around a text editor. This textual code is very different from the scientist's mental model of the problem he is solving and the numerical calculation used to simulate the science, and requires a translation from the high-level mental-model level to the code level.

Program visualization can be used to help with this translation problem by representing a program in such a way that it is suggestive of the higher-level mental-models behind the code. Common information visualization and program cognition tools can be applied to this domain, including program slicing, pan and zoom, overview plus detail, and focus and context methods. A recently developed method called Continuous Semantic Zooming (CSZ) was designed primarily to address the problem of visualizing parallel programs. This method uses viewpoint proximity to trigger a change in detail in objects, or, in this case, program elements, being examined. This allows less detailed, higher-level views to coexist with more detailed, lower level views. The CSZ method allows for a continuous transition between these detail changes, permitting the user to concentrate on the material, rather than orientation in the code.

Preliminary human subject experiments have been conducted using this method for a static serial program. To use the method to its full capabilities, it can and should be applied to larger, more complex interactive problems, such as the interactions in a parallel program. Issues concerning multiple representations of trace data, real-time vs. stored data, display of multiple different types of processor data, and message size and content can all be explored using the CSZ method.

This task will design a set of pilot human subject experiments to attempt to quantify some subset of these issues. This will begin by identifying the relevant parallel program information, and devising methods for applying the CSZ technique to that information and displaying the results. Monitoring and instrumentation of parallel programs will be explored, using off-the-shelf tools as much as possible. Next, a relevant human visualization task will be identified, in collaboration with LACSI colleagues, where performance is measured by a combination of time and accuracy. A process of designing and refining a doable test will begin, and the supporting software will be developed. Finally, a single pilot study will be performed. Results will be summarized in a report and paper intended to be presented at the LACSI Symposium.

This task will lay the groundwork for many useful future studies of parallel program visualization, and will lead to the development of more useful tools for the understanding of complex parallel programs. The results may have a large impact on the future of visualization of parallel programs.

***FY03 Tasks:***

- *Q2:* UNM visualization software modified to support monitoring of large parallel programs.

- *Q3:* Experimental task defined and codified. Experiment conducted.
- *Q4:* Report written.

## 1.3. Foundations of Computational Science

**Investigators: Danny Sorensen, Liliana Borcea, John Dennis, Mark Embree, Mike Fagan, Roland Glowinski, Lennart Johnsson, Petr Kloucek, Yuri Kuznetsov, Matthias Heinkenschloss, William W. Symes, Richard Tapia, Frank Toffoletto, Yin Zhang**

Based on a contemporary assessment of LANL's needs, the *Foundations of Computational Science* effort encompasses two main aspects of computational mathematics:

1. Mathematical Methods and Algorithms
2. Verification (and Validation) of Simulation Codes

Generally, the team works on the development, analysis, and implementation of numerical methods that address leading problems of science and engineering. In addition, the team is interested in developing computational metrics and invariants that can confirm important properties of current and future simulation codes. We also have a keen interest in software design and practice suitable for large-scale simulation codes and in automating the process of tuning those codes for efficiency on specific platforms. These interests are very much in line with the current and future needs of ASCI. The text that follows describes the individual subprojects and how they address these research issues.

### 1.3.1. Parallel Tools for the Analysis and Optimization of Linked Subsystems

**Investigators: Matthias Heinkenschloss, John Dennis**

The objectives of this project are the development of parallel tools for the coupling of simulations of aspects of a problem to full system simulations on which optimization-enabled decisions can be based. This requires the development of a mathematical and algorithmic framework that supports the coupling of expensive subsystem simulations with varying accuracy, the development of optimization algorithms suitable to use with these simulations, the identification of parameters in these linked systems, and the error prediction in linked simulations.

***Methods and Techniques:*** Methods developed in this project are both non-Newton (direct search methods (DSMs)) and Newton (simultaneous analysis and design (SAND)) based to address different aspects of the problem. Both approaches can be combined. Algorithms are backed by rigorous convergence theory. Their implementations are tested on model problems and applied to "real-world" engineering analysis and design problems with relevance to LACSI.

***Verification and Validation:*** Methods and techniques developed in this subproject may be used within LANL simulation and optimization tools to improve robustness of subproblem solvers within these tools. The methods can also be used for nonlinear sensitivity studies (if the outputs are allowed to vary in a certain range, how accurate must the parameters be; if the parameters vary in a certain range, what is the variation in the outputs) and to aid validation (are there model parameters such that simulations "match" experimental data). Coupling optimization with simulation offers additional ways for verification and validation, which, although somewhat

speculative, have proven successful in our industrial experience. These are elaborated on in the following paragraphs.

Another approach to verification and validation is suggested by our industrial experience. Several times, engineers have given us simulation-based optimization problems that, when we run the optimization code, yield results that point to problems in the simulation. Of course, the problems are often in the formulation of the optimization problem as well, but inappropriate objective function models and missing or inconsistent constraints usually are easy to detect and eliminate from consideration. The most frequent location for such problems with the simulation is, not surprisingly, the boundary of the region where the user believes the simulation to be valid.

Another approach to uncovering errors in a simulation code is to use experimental design to choose a set of input vectors at which to test the simulation. The reason why optimization computations may be more effective at rooting out inconsistent simulation results than sampling based on experimental design is probably that the user has a better idea what to expect from an optimization run. For example, the user might have run the simulation with no problem with a certain set of inputs. If the user then poses an optimization problem solved by those inputs, and starts the optimization code not so far away, then convergence to a different solution to the optimization problem is very suspicious. Such a problem might be to minimize the distance from the output variables produced by the chosen input variables subject to the trial input variables being in the region of interest. These ideas are quite speculative, but they are based on some unplanned success in uncovering simulation bugs.

***Long-term Goals:*** Develop parallel tools for expensive engineering optimization problems, which may involve continuous and a moderate number of discrete or categorical variables, which may only allow access to simulations as a black box, and which can only be expected to generate inexact function evaluations.

***Short-term Goals:*** Explore the use of different surrogate models in DSMs, add grid-based parallelism and improve use of derivative information in DSM, add domain decomposition based parallelism into SAND approach, and explore additional applications of current algorithms.

***FY03 Tasks:***

- Identify example problems with interest to LANL.
- Formulate DD approaches for optimization.
- Compare and evaluate DD approaches for optimization on test problems.
- Refine pyramid direct search for dealing with larger numbers of design variables in surrogate optimization.
- Formulate a reduced basis/space mapping approach for dealing with larger numbers of design variables.
- Test space mapping against pyramid direct search for a hundred-design-variable test problem.

### 1.3.2. Code-based Sensitivity Subproject

**Investigator: Mike Fagan**

Users of complex, large-scale computer models often need to assess the sensitivity of model output to changes in model input. Such sensitivity calculations lie at the heart of methods for uncertainty quantification. Code-based sensitivity analysis combines compiler-based source code analysis and transformation with modern numerical algorithms to augment model output with sensitivities. The code-based sensitivity project researches methods and techniques for more effective derivative computation, methods for creating code-based sensitivity tools for programming languages with sophisticated features, and novel derivative applications.

***Methods and Techniques:*** Ongoing research in code-based sensitivity methods and techniques is following two lines:

1. Extension of current AD algorithms to modern language features such as pointers, dynamic memory allocation. Current AD technology for Fortran 77 does not address pointers or dynamic memory allocation. To extend the AD techniques to Fortran 90 and C/C++, however, an AD tool must obviously address the more sophisticated memory models.
2. Investigation of recomputation vs. storage issues in automatic adjoint generation for general programs. Adjoint computation requires reversal of control flow, and preservation of intermediate state. Preservation of intermediate state, however, does not necessarily mean saving all of the intermediate state. Rather, an adjoint-computing program could save some intermediate state, and compute other intermediate state as needed. Naïve recomputation, however, is extremely expensive. Deciding what to save and when is an active area of AD research. To further complicate matters, advanced language features like dynamic memory allocation and pointers make the recomputation issue an order of magnitude more complicated.

***Verification of PDE-based Simulations:*** Most of the models used in computational science and engineering (CSE) are based on ordinary differential equations (ODEs) or partial differential equations (PDEs). Consequently, ensuring the correctness and accuracy of these ODE- and PDE-based computational models is of great importance to the CSE community. As part of the general task of confirming correctness and accuracy, the DE models must be verified. To verify a computational DE model means confirming that the computer program actually solves the DE in question. One class of techniques for verifying DE-based models involves using automatic differentiation (AD) to compute the derivatives a program purports to solve for a given DE. A DE solver can be verified at a given point by using AD to evaluate the derivatives of solution values produced by the solver at the given point, multiplying by the DE coefficients, summing the products and comparing the sum to the DE right-hand side. If the comparison is "close", then the DE is verified at the given point. The goal of the VDAD subproject is to develop and apply the AD technique to the verification

***Long-term Goals:*** Longer-term goals for the code-based sensitivity project include:

- Extending the automation of both discrete tangent-linear and discrete adjoint generation capability to sophisticated language facilities like pointers, exceptions, overloaded operators, threads, modules and other language features that appear in modern programming languages.

- Applying automatic differentiation for verification of ODE-based and PDE-based computer models.
- Applying automatic differentiation in the construction of Newton-Krylov solvers.
- Extending Adifor-style automatic differentiation to additional programming languages, notably Fortran 90, Java, and C/C++. The CartaBlanca project, in particular, has expressed interest in Java.
- Extending the augmentation paradigm to include other sensitivity measures such as intervals or probability distributions.

***Short-term Goals:*** The short-term goals for the code-based sensitivity effort are directly related to the ongoing collaboration with the Telluride Project directed by Doug Kothe (technical point-of-contact is Rudy Henninger). There are 3 major short-term goals:

1. Assist in the application of Adifor to current Fortran 77 codes.
2. Extend the Adifor technology to Fortran 90 so that accurate sensitivity calculations can be easily generated for the Truchas code.
3. Show proof-of-concept for verification of differential equation simulations by applying the technique to some small code previously validated by other means.

***FY03 Tasks:***

- Continue development of Open64-based Fortran 90 Automatic Differentiation tool (Adifor90).
- Continue factoring the Truchas code to yield a sequence of increasingly more complicated test cases for the Adifor90 AD tool.
- Locate a suitable code for proof-of-concept for AD verification technique.

### 1.3.3. Numerical Methods for Coupling Kinetic and Fluid Plasma Models

**Investigators: Petr Kloucek, Frank Toffoletto**

**Collaborator: Dr. J. Brackbill, LANL**

Magnetic reconnection, a process where the topological structure of magnetic fields is changed via violation of the frozen in flux condition, plays a major role in energy conversion in space and astrophysical plasmas. For the Earth's magnetosphere, it is generally believed to be the major process that supplies solar wind plasma and energy into the magnetosphere. For the sun, it is thought to play a major role in coronal mass ejections. In recent years, large-scale modeling of the Earth's magnetosphere has focused on models that solve the MHD equations on a global scale, thereby neglecting kinetic effects. Global-scale kinetic codes are beyond the reach of present day computer technology. Furthermore, to understand what effects that kinetic codes have on global scales, we need to explore new techniques for embedding the kinetic codes (such as the Vlasov-Maxwell solver), inside large-scale fluid codes. The problem of setting up boundary conditions so that a scale kinetic code can pass information to and from a large-scale fluid code in a physically and mathematically consistent fashion is a difficult challenge. These multi-physics techniques that we are developing are also applicable to other regions of the space environment where kinetic effects have global implications. For the Earth's magnetosphere such regions include the region above the Earth's auroral zone and the inner magnetosphere.

Successful completion of this project opens the possibility to build and use hybrid stochastic-deterministic models. Such models are desirable in a number of fields including materials science, biology, and social sciences.

***Methods and Techniques:*** We continue further development of Poincare-Steklov transition conditions between stochastic and deterministic models. We will focus on finding a suitable Fokker-Plank equation that can furnish transient probability density enabling the information propagation from deterministic to stochastic region.

***Validation:*** The proposed coupling of kinetic and fluid models may be used to identify limits of predictability properties of fluid models when applied in situations when very small-scale structures can develop. Fluid models average over a broad range of scales. Consequently, the computational results may not be reliable. Imbedding kinetic models in a small but important computational region may highlight the limits of applicability of the large-scale fluid models. The validation via comparison with kinetic codes is certainly important for global weather computational models as well as environmental models.

***Long Term Goals:***

- Accurate modeling and computer simulation of the plasma physics fluid-kinetic interaction applied to magnetic reconnection.
- Develop transition conditions for connection of certain types of stochastic partial differential equations and classical field-based MHD and fluid models.
- Develop homogenized version of the Landau-Vlasov-Maxwell (LVM) stochastic equations, and develop suitable finite element approximation, stabilization and parallelization techniques for these equations.

***Short Term Goals:***

- Extend to higher dimensions and more complex systems the one-dimensional stochastic-deterministic coupling conditions.
- Extend to higher dimensions the one-dimensional least-square-based stabilization of the Landau-Vlasov-Maxwell (LVM) equations.

***FY03 Tasks:***

- Investigate the use of a Fokker-Planck equation for the transient probability distribution between stochastic (kinetic) region and deterministic (fluid) region.
- Develop a computer program based on the overlapping domain decomposition in which both deterministic and stochastic processes are active.
- Develop a computer implementation of coupling conditions for Langmuir waves.
- Investigate computationally the sensitivity of the coupling conditions with respect to ghost values introduced to provide transient probability density in the deterministic region.
- Derive a suitable limit of homogenized Landau-Vlasov-Maxwell equations.
- Propose a parallelization of the resulting homogenized LVM equations.
- Develop a parallelized code in 2 spatial and 3 velocity dimensions of the LVM equations.

### 1.3.4. Large-Scale Linear Systems and Eigenvalue Methods

**Investigators: Danny Sorensen, Mark Embree**

This project is concerned with the development of methods and software for large eigenvalue problems and related applications. The project will extend the capabilities of the highly successful P_ARPACK eigenvalue software. This software executes on massively parallel systems and provides enabling technology in numerous application areas. It also can serve as an excellent test bed and point of interaction for new compiler technology.

We are also interested in the development of dimension (order) reduction techniques for dynamical systems and control. We have been developing projection methods that are closely related to the Krylov projection techniques used in large eigenvalue computations.

***Methods and Techniques:*** Within the LACSI project, we have developed a new method for approximating a shift-invert spectral transformation without direct matrix factorization. This transformation is required to accelerate convergence in difficult cases such as computing clustered or interior eigenvalues. We intend to improve this technique and apply it to stability and bifurcation analysis of certain dynamical systems.

Calculating the rightmost eigenvalues of large-scale dynamical models requires a spectral transformation to accelerate the convergence of most eigensolvers. While often used in practice, this technique can be computationally expensive. We have found a fixed-polynomial operator that can approximate the spectral transformation and reduce the computational cost. This fixed-polynomial operator has successfully been used to perform linear stability analysis on an idealized reduced-gravity quasigeostrophic ocean model. We have demonstrated the potential of this approach by successfully detecting Rossby waves that can be indicators for the onset of chaotic behavior in the dynamical system. We have been collaborating with LANL scientist Balu Nadiga on this application.

Matrix-free Newton-Krylov methods are preferred for solving these large-scale systems of nonlinear equations. However, the efficiency of this approach is dependent upon the quality of the preconditioner. The best preconditioners incorporate problem-specific information. We believe the information obtained from the linear stability analysis can be used to build a better Newton-Krylov preconditioner. We intend to investigate the development of adaptive

preconditioners that make effective use of eigeninformation. Many of the software components needed to implement this idea are readily available.

***Verification and Validation of Simulation Spectral Information:*** Simulations of dynamical systems are often concerned with analyzing the system for instabilities. Traditional stability analysis typically begins with a nonlinear PDE model, which is linearized about a steady-state solution and discretized. If all eigenvalues of the discretized formulation fall in the left half of the complex plane, one concludes that the steady state is stable.

Assessing the accuracy of numerically computed eigenvalues, and their physical relevance, is a central aspect of model validation. Software packages for solving large-scale eigenvalue problems, such as ARPACK, compute approximate eigenpairs that have small residuals. For hermitian problems, this guarantees that the computed eigenvalue is close to a true eigenvalue. For nonhermitian problems, as arise, e.g., in fluid dynamics and magnetohydrodynamics, a small residual can occur even if the computed eigenvalue is far from any true eigenvalue. Predictions of stability or instability may be incorrect.

One can identify situations where such errors are possible by inspecting pseudospectra, a generalized notion of eigenvalues that includes information about eigenvalue sensitivity. We propose to compute pseudospectra using the projection techniques of Wright and Trefethen. These techniques still rely upon ARPACK but utilize the process to compute pseudospectra instead of eigenvalues. We intend to apply this technique to obtain a pseudospectral analysis of certain steady state solutions to the OCM of Nadiga. Should that prove successful, we hope to apply the technique to other critical LANL simulations that require stability analysis.

Pseudospectra reveal more than whether a computed eigenvalue can be trusted or not; they also give insight into transient physical phenomena. Although all eigenvalues of a matrix may be located in the left half of the complex plane, indicating asymptotic stability, small perturbations to the steady state can be magnified before eventually receding. When combined with nonlinearity, this transient growth can cause instability in a system whose exact eigenvalues predict stability. Pseudospectra have proved an important tool for analyzing this behavior in transition to turbulence, where unstable flows are observed at lower Reynolds numbers than predicted by eigenvalue stability analysis.

***Long-term Goals:***

- The ultimate goal currently lies with the full ocean model. We hope to find a parameter regime where the model goes to a steady state, and then look into the stability of that state and compute the most unstable eigenmodes.
- Develop a methodology for making pseudospectral analysis available to appropriate simulations of important dynamical systems.

***Short-term Goals:***

- Construction of linear system preconditioners based on adaptive use of spectral information.
- Develop a pseudo spectral analysis methodology for the OCM.

***FY03 Tasks:***

- Write a report on a scalable method for eigenvalue problems. The report will show the application of the method to large-scale stability analysis and sensitivity analysis in an ocean circulation model (OCM).
- Develop an approximate shift-invert spectral transformation (ASI preconditioner) for determining Rossby waves in Nadiga's OCM.
- Develop serial code for determining Rossby waves using ARPACK with the ASI preconditioner.
- Parallelize the serial code and perform some large-scale test cases.
- Obtain sample matrices from LANL and test performance of linear system solvers with conventional incomplete factorization and sparse approximate inverse preconditioners.
- Design novel spectral-based preconditioners for simple test problems (e.g. 5-pt/7-pt regular grid discretizations of non-self-adjoint PDEs).
- Experiment with new algorithms on LANL test matrices; comparison with conventional preconditioners.

### 1.3.5. Large-Scale Nonlinear Optimization Algorithms and Software

**Investigators: Yin Zhang, Richard Tapia**

Mathematical optimization is a fundamental tool for enhancing performance of complex systems. In recent years, it has been applied to more and more areas. As a result, the demand for efficient optimization algorithms has become ever greater, especially for large-scale problems where off-the-shelf algorithms are often ineffective, necessitating the research and development of special-purpose algorithms. The goal of this project is to develop algorithms and software for a number of large-scale optimization problems. While we continue our research in semidefinite programming and its applications, we will concentrate on developing novel algorithms for several classes of graph-partitioning problems based on a unified nonlinear optimization framework. These graph-partitioning problems have wide ranges of applications in, for example, VLSI circuit design, statistical physics, machine scheduling, data mining, sparse matrix computation, and parallel computation.

***Methods and Techniques:*** Our preliminary results have indicated that partitioning algorithms based on our nonlinear-optimization framework can produce high-quality partitions, very often better than sophisticated existing algorithms. However, our algorithms are still considerably slower than the best existing algorithms, and the quality advantage seems to deteriorate as the problem size increases. In light of these preliminary results, our research will focus on efforts to improve the speed of the proposed approach to make it comparable to that of the best existing algorithms. This will be done via two routes. Firstly, we need to devise a more efficient continuous optimization algorithm for our special optimization problems at hand; secondly and more importantly, the most significant speed-up must come from an effective combination of our algorithm with a multi-level strategy (which is used in all of the state-of-the-art partitioning algorithms). A multi-level strategy will not only help speed up our partitioning algorithms, but also improve the scalability of our algorithms in terms of partitioning quality.

***Validation of Mesh Partitioning:*** Solving large systems of partial differential equations (PDE) in the three-dimensional space on parallel computers usually requires partitioning of the

underlying computational mesh. If a system of PDEs is solved repeatedly on the same mesh, then the quality of the mesh partitioning becomes a critical factor in determining the overall computational time. Today the state-of-the-art partitioning algorithms can partition meshes of tens of millions of nodes in seconds. However, the quality of such partitions is essentially unknown. We believe that there may be a need for some partitioning algorithms that are not the fastest, but capable of generating better partitions if given a larger, but still affordable, amount of time. Such algorithms can be used to do off-line validation for mesh partition. In addition, such algorithms can provide improvements to mesh partitions generated by faster algorithms. The algorithms studied in our subproject fit well into this category.

***Long-term goals:*** For the subproject of graph-partitioning algorithms, the long-term goals include:

- Develop a set of well-founded and well-tested algorithms for different classes of graph partitioning problems suitable to our nonlinear-optimization framework, and
- Develop a compact research software package capable of producing high-quality partitions for different classes of problems in time comparable to that of the best existing algorithms.

We expect that the partitioning quality of our algorithms will be better, at least for some classes of problems, than the best existing algorithms.

***Short-term Goals:*** In the short-term, we will focus on a particular graph-partitioning problem—the min-bisection problem, which is one of the most important graph-partitioning problems with the most applications. We will first develop a prototype algorithm and a research code for this problem. Results obtained for this problem will prove to be useful for other graph partitioning problems. We continue to try to identify possible research interests that fit our research expertise and are in common with those of some LANL scientists working in optimization related projects. The aim is to form new joint projects on optimization algorithms.

***FY03 Tasks:***

- Conduct preliminary theoretical analysis to gain insight into the possible behavior of the algorithms in our unified framework.
- Design, develop and test prototype research codes that implement the basic versions of the algorithms.
- Collect sets of representative test problems for different classes of graph partitioning problems.

### 1.3.6. Parallel Numerical Methods for the Diffusion and Maxwell Equations in Heterogeneous Media of Strongly Distorted Meshes

**Investigators: Yuri Kuznetsov, Roland Glowinski, Lennart Johnsson**

Efficient parallel numerical methods for the diffusion and Maxwell equations in highly heterogeneous anisotropic media is an important topic for scientists and engineers working in computer simulation of complex physical phenomena. This statement is very relevant to several research groups at LANL and UH.

***Algorithms and Methods:*** The researchers from the LANL part of the project are very experienced in accurate and physically consistent approximations to the diffusion and Maxwell

equations on strongly distorted meshes as well as in applications of advanced numerical methods to real life problems.

The researchers from UH have long-term experience in discretization of partial differential equations by mixed and hybrid finite element methods. They also hold the worldwide leading positions in the design of efficient parallel iterative solvers based on a combination of domain decomposition, fictitious domain, and multilevel techniques.

***FY03 Tasks:***

- Develop Jacobian (RJ) Rezoning Algorithms for the shape optimization and constrained smoothing of the unstructured 3D grids for the Arbitrary Lagrangian-Eulerian (ALE) methods.
- Develop, investigate, and validate mixed/hybrid and operator support methods on the locally refined strongly distorted grids for the 3D diffusion equations.
- Investigate optimization and parallelization of multilevel/multigrid preconditioners for the 3D diffusion equations with non-diagonal diffusion tensors.

### 1.3.7. Methods and Software for Inverse and Control Problems

**Investigator: William W. Symes**

Solution of simulation-driven optimization problems, of which control and inverse problems represent two important subcategories, invariably involves integrating several levels of abstraction, and often several computing environments. These features cause conventional, procedural approaches to code development to bog down in design and maintenance nightmares. Object and component programming paradigms provide natural means to express multilevel abstraction and multiplatform computation. These paradigms are near universal in commercial programming, but are still gaining acceptance in scientific computing. This project demonstrates the application of object and component programming methods to a particular scientific programming task type to produce applications that attain the same performance levels as conventional procedural methods but exhibit clear-cut superiority in maintainability, extensibility, and code reuse.

***Methods and Techniques:*** The C++ Standard Vector Library ("SVL") provides an object-oriented environment for expressing algorithms based on calculus in Hilbert space. The ongoing project, of which SVL is the latest product, has introduced several innovations into OO numerics:

- vector spaces as objects
- evaluation objects for functions and operators
- validation methods implemented in base classes

which will become common features of all high-performance OO optimization libraries, and perhaps eventually of a standard set of interface classes. Component programming is designed into SVL from the beginning: communication is localized in a few class subtrees, which minimizes the work required to implement new component-based applications. The communications paradigm is modular and flexible; the current SVL communications classes,

which wrap low-level socket calls, may be exchanged in the future for libraries based on the emerging CCA standard or another component framework.

The FDTD library is an SVL-based framework for time-dependent simulation-driven optimization. FDTD takes advantage of the features shared by all such problems to reduce the code generation task for derivatives and adjoints to a level easily handled by several automatic differentiation ("AD") packages. AD is quite successful in this structured application. The component design of SVL is realized in FDTD, creating natural distributed implementations of compute-intensive applications.

The SVL team views SVL as a prototype for exploration of the applicability of OO and CO programming methods to scientific computing and as an exceptionally powerful tool for research on various control and inverse problems. Our intent is that the SVL experience will be helpful to the scientific computing community in designing future OO/CO standards with larger or different scope.

***Validation and Verification:*** SVL and FDTD address code validation and verification for simulation-driven optimization in several ways. First, careful design of the object library leads to extensive reuse of optimization and utility implementations, dramatically decreasing the likelihood of bug introductions due to code modification. This is the reason for the complete dominance of OO in commercial programming, and has been very evident in our experience with earlier versions of SVL. Second, component design permits simulation modules, for instance, to be designed and tested independently, and therefore more thoroughly than would be possible in monolithic designs. Third, the SVL base classes for functions and operators have standard validity tests for derivatives ("sensitivities") and other auxiliary objects that are immediately available for any SVL-implemented simulators, as implemented (non-virtual) base class methods. These tests flag many common errors, and dramatically reduce the likelihood that erroneous code survives. Fourth, the FDTD framework requires only that the user implement code for single time steps, and provides methods, implemented in the base classes, for testing the single time step modules. Once these (intrinsically lightweight) tests are passed, the FDTD design provides a guarantee of correctness for the entire application. Fifth, FDTD provides "sensitivity" (derivative) computations that can be used in the fashion envisioned in the Code-based Sensitivity Subproject to assess the validity of simulators. Sixth, the several related modules in the single-step code may be generated by (easy) AD from the basic simulator code, which provides its own guarantee that at least the relationships between these modules are likely to be correct.

***Long-term Goals:*** The overarching goal of this project is to contribute to the evolution of a standard interface model for optimization and linear algebra applications. ESI is a step in that direction, but has a variety of drawbacks, notably its exclusive linear system orientation and the appearance of low-level task distribution details in the high-level interface, which SVL avoids. Many other OO/CO projects (TAO, RSQP++, Dakota, Trilinos, OOQP,…) implement parts of this vision. Carefully defined, minimal interfaces can accommodate a maximal scope of applications while permitting high quality implementations of important and useful algorithms, not limited to a single domain, family of data structures, or computing platform. SVL aims to demonstrate this proposition within the scope of simulation-driven optimization.

***Short-term Goals:***

- Complete development of integrated component framework in SVL.
- Canonical implementation of adaptive time stepping, integration with spatial AMR.
- Develop reliable integration of optimization with adaptive simulation.
- Develop methodology for integrating complex simulators, possibly without derivative capability, into simulation-driven optimization organized by SVL.
- Exploration of automatic FDTD wrapper generation as part of AD.

***FY03 Tasks:***

- Client-server implementation of seismic simulation and inversion using various server platforms (cluster, SMP).
- FDTD-based data assimilation for satellite particle flux methods, using adaptive semi-Lagrangian integration of approximate plasma models (joint with Rice Space Physics group).
- SVL implementation of line-search and trust-region SQP, application to time-decomposed flow control problems (joint with M. Heinkenschloss).
- Identify partner project within LANL as test bed for extension from simulation to simulation-driven optimization using SVL toolkit.

### 1.3.8. Stochastic Aspects of Inverse Problems and Uncertainty Assessment

**Investigator: Liliana Borcea**

Inverse problems form a particular subclass of simulation-driven optimization problems, in which model parameters are sought to fit observed data. A wide variety of ASCI-related problems, and even such disparate tasks as crack detection and wildfire management, involve inverse problems. In practice, all inverse problems have a stochastic component, originating both in data noise and in model uncertainty or underdetermination. For example, in nondestructive testing, one seeks large defects in the materials, but smaller inhomogeneities are present; these influence the data but are important (and even determinable) only in a statistical sense.

This subproject seeks to understand the effect of modeling uncertainty on the quality of the inversion results. Progress requires a combination of stochastic analysis and numerical simulations. This research seriously stresses the throughput of simulators, which must conform to all the relevant physics at all scales of importance to the data, but be sufficiently fast to generate many realizations needed for developing and validating inversion approaches.

We will build on recent progress in developing explicit and accurate models for quantifying loss of resolution in the inverse scattering problem of imaging small objects in noisy media. These models show that the effect of the (unaccounted for) inhomogeneities in the medium can be quantified in terms of just a few parameters, which one can estimate from the data.

The work of the coming year will provide extensions to this recent progress, generalizing the underlying physical assumptions and the amount and type of data assumed. Software development, leveraging work done in other parts of this project, will focus on parallelization and componentization of PDE simulators coupled with optimization software.

***FY03 Tasks:***

- *Q1 and Q2*: Using stochastic analysis, develop explicit and accurate model for estimating resolution loss in imaging small objects in noisy media. Develop software-enabling assessment of model accuracy.
- *Q3*: Extension of stochastic inverse scattering to imaging larger (extended) objects or surfaces, in noisy environments. Development of an image deblurring strategy, based on our understanding of resolution loss in noisy media.
- *Q4*: Development of optimization algorithms and software for imaging the reflectivity of extended targets in noisy environments. A key component of this work is the coupling of the optimization software with parallelized PDE solvers for the forward problem.

## 1.4. Application Performance

**Investigators: John Mellor-Crummey, Barbara Chapman, Keith D. Cooper, Rob Fowler, Guohua Jin, Ken Kennedy, Celso Mendes, Dan Reed, Scott Rixner, Linda Torczon**

### 1.4.1. High-level Compilation and Compiler-based Tools

**Investigators: John Mellor-Crummey, Barbara Chapman, Rob Fowler, Guohua Jin, Ken Kennedy**

The objective of this project is to develop compiler and run-time technology that will help application developers achieve a high fraction of peak performance on large-scale parallel computing systems.

Principal areas of focus within this project include:

- *Productivity Enhancing Tools (Chapman):* Many problems that arise in application development for parallel architectures are not specific to a code but are instances of a kind of porting problem. Increasing programmer productivity may go hand-in-hand with superior programming if such problems can be recognized and the programmer presented with a strategy for remedying it. We will create a tool that provides the developer with a variety of features to study an application and collect specific performance problems, devise solution strategies and develop techniques that will "match" application code regions to these problems.
- *Compiler-assisted development tools (Kennedy, Jin, Fowler, Mellor-Crummey):* Achieving the highest possible performance with programs may require complex restructuring that is best performed by a tool, but with user guidance to control its actions. Our goal is to capitalize on user knowledge that is not (or not easily) derivable by a compiler. In this effort, we will develop tools to assist users in creating high performance codes by choreographing compiler-based code restructuring transformations with simple directives that enable fine-grain control of the process. We will continue research on adaptive compilation for performance tuning. This research is a generalization of the "self tuning" approach employed by libraries such as ATLAS. Our adaptive compilation framework calls for a tool to analyze a program and generate a parameterized version of the same program. The range of parameters corresponds to the solution space of various program transformations combined. A run-time system then tunes the application with different parameter instantiations and collects performance data associated with these instantiations. At the end, the compiler uses the parameter values that have best performance to generate a clean code for a tuned application. This automatic tuning strategy can help a program achieve high performance across different architectures, more importantly, help legacy code adapt to modern machines.
- *Node Performance: (Kennedy, Jin, Fowler, Mellor-Crummey):* A principal problem facing scientific codes on systems composed of commodity microprocessors is ineffective use of multi-level memory hierarchies. We will investigate strategies for improving utilization of multi-level memory hierarchies within a single thread of control using a combination of computation restructuring to improve temporal reuse along with data restructuring to improve spatial locality and reduce conflicts. This effort will include compile-time

approaches such as loop fusion and program transformations to improve register allocation of array values, along with run-time approaches such as sparse matrix dissection to improve temporal locality in irregular computations.

- *Multiprocessor Performance: (Kennedy, Jin, Fowler, Mellor-Crummey, Chapman):* Challenges for achieving high performance include analyzing and transforming programs within and across procedures to improve parallelism, partitioning data and computation to exploit parallelism effectively, and optimizing communication to minimize volume, frequency, and exposed latency. Topics we will investigate in this area will include data-parallel compiler technology, multiprocessor time skewing – a partitioning strategy that improves node locality and reduces communication, and compiler technology for explicitly parallel languages such as Co-array Fortran and UPC. We will investigate alternative execution strategies for OpenMP that might permit this language to scale up to large numbers of threads and will also explore compiler technology to support data parallel programming under this programming model (with suitable extensions) and the hybrid OpenMP+MPI model.

This research will consider scientific applications that employ sophisticated methods for processing large-scale data sets, including the use of unstructured meshes and dynamic adaptation of data structures.

Procedurally, this project effort will involve experimentation (including simulation, measurement, and analysis of applications) to identify the most significant performance bottlenecks, developing and prototyping techniques for improving performance in such applications, developing compiler analyses and transformations to automate application of these techniques to the extent possible, and incorporating these techniques into compilers and tools that can be used by application scientists.

In addition to technology for conventional large-scale parallel systems, we will begin to explore compilation technology appropriate for emerging architectures for the later half of the decade, including processor-in-memory systems.

In FY02, our studies of node performance with Sage led to manual and semi-automated restructuring transformations that improved the sparse-matrix vector multiply performance of its conjugant gradient solver by well over a factor of 2. In Rice's Fortran 77-based compiler infrastructure, we built a prototype of a sophisticated tool for radical restructuring of scientific codes through massive loop fusion, iteration space reordering, storage reduction, and register-blocking. Also, we made substantial progress in creating a source-to-source infrastructure for manipulation production Fortran 90 programs based on the Open64 compiler infrastructure.

***FY03 Tasks:***

- Develop compiler run-time libraries and tools technology for improving the performance of ASCI workloads.
- Complete development of Open64 source-to-source infrastructure to support manipulation of full-featured Fortran 90 codes.
- Construct prototype program restructuring tools to improve performance of production ASCI codes using Open64 compiler infrastructure.
- Experiment with Open64-based compiler tools on LANL codes.
- Create a GUI for Open64 environment to support interactive compiler-based code analysis.

- Develop strategy and basic implementation plan for automatic improvement of scalability of OpenMP programs.
- Realize implementation of OpenMP with execution strategy that enforces barriers between threads only where needed to overcome heavyweight "all" barriers.
- Deploy initial Dragon tool with basic functionality for navigating Fortran/C/C++ code and analyzing its structure, assist LANL scientists in using the tool, and develop next phase of tool work.
- Explore generation of parameterized codes for using adaptive compilation and search strategies for exploring parameter space. Investigate applicability of processor-in-memory systems for ASCI problems.

### 1.4.2. Compilation Issues for High-Performance Microprocessors

**Investigators: Keith D. Cooper, Ken Kennedy, John Mellor-Crummey, Scott Rixner, Linda Torczon, Barbara Chapman**

This investigation focuses on developing the compiler technology required to obtain a reasonable fraction of the performance available on high-end processors, such as the Itanium, the PowerPC, or the high-end Pentium machines. The premise of the research is that code written in high-level languages, such as Fortran, C, C++, or Java, should execute well on these machines. This requires that the compiler manage many different aspects of program execution. Among the areas that this project will explore are:

- *Instruction-level parallelism (ILP):* The compiler must transform the input program so that it contains enough ILP to keep the functional units busy. It must schedule the code so that the processor finds those opportunities. It may need to manage placement of data and operations on the chip (i.e., in distinct clusters or even distinct processors) as well.
- *Memory hierarchy management:* The compiler must ensure that operands are available, in registers or cache, when an operation needs them. To do this, the compiler must transform the program to ensure that its locality matches the memory hierarchy of the target system. Example transformations include blocking, prefetching, and streaming.
- *Handling new architectural features:* As microprocessors evolve, new features appear that are intended to improve performance. Recent examples include predication, speculation, and clustered register sets. Before applications see the benefits of these features, compilers must have effective mechanisms for using them.

The primary goals of this project are to develop practical compilation techniques to address the problems that keep current compilers from generating efficient code for high-performance microprocessors, and to work at transferring these techniques, and other best-practice techniques, into commercial and open-source compilers.

To address these issues, we have brought together a team that includes expertise in analysis and translation for parallel systems (Kennedy, Mellor-Crummey), managing memory hierarchies and locality (Cooper, Kennedy, Mellor-Crummey, Rixner), and low-level code generation issues (Cooper, Rixner, Torczon).

In FY02, we developed new algorithms to reduce memory traffic by eliminating redundant loads (up to 35% of all loads) and stores (miniscule numbers). We developed a prototype assembly-

level vectorizer for the Pentium 4's SIMD unit. We used simulation studies to investigate the speed and power implications of new cache structures. We performed a large (14 CPU-month) experiment to begin characterizing the mathematical spaces in which adaptive compilers will work. This year, we will continue on several of these fronts. We will also move into working on the Open 64 compiler infrastructure.

***FY03 Tasks:***

- Investigate improvements to the register allocation and scheduling algorithms in the Open 64 compiler.
- Evaluate potential for improvement in the performance of applications compiled with Open 64.
- Work with the gcc development team to improve code generation in gcc.
- Further improve our assembly level optimizer/vectorizer for the Pentium 4.
- Work LANL benchmarks and abstracts of those benchmarks into our experiments on adaptive compilation.
- Investigate strategies for using thread programming models to exploit simultaneous multithreading within a processor.

### 1.4.3. Intelligent Performance Analysis and Adaptability

**Investigators: Dan Reed, Celso Mendes**

Parallel applications are becoming increasingly multidisciplinary, with libraries and other application components implemented using diverse programming languages, models, and parallelization strategies. Moreover, large-scale systems now have thousands and soon will have tens of thousands of processors. Measuring and improving application performance on these "extreme scale" systems, while not unduly perturbing system behavior, poses a plethora of new challenges. The large number of components in extreme scale parallel systems also makes component failure inevitable; meaning long-running applications must be resilient to hardware faults or risk being unable to run to completion. In consequence, it is now extraordinarily difficult to achieve and sustain high fractions of peak hardware performance on today's extreme scale systems for multidisciplinary applications.

To optimize the behavior of complex applications, performance analysis software must also evolve, replacing simple measurement tools with tools that provide deep integration of compile-time transformations, measurement, and analysis while also treating thousands or tens of thousands of processors and tasks. Moreover, time varying resource availability and demands will require increasing use of real-time, adaptive performance optimization, while maintaining computation progress and fidelity in the face of component failures. This integration, based on user-specified, compiler-synthesized, and measurement-validated performance contracts, will enable creation of a new generation of nimble, high-performance applications.

We will explore new techniques for scalable, statistical characterization of extreme scale systems, enabling low overhead performance measurement and prediction. We will also develop compact, scalable representations of dynamic execution behavior that can be used for performance contract specification and evaluation. In addition, we will explore the efficacy of automatic, multiversion code selection for performance remediation in the face of behavioral

changes. Finally, we will explore common failure modes on extreme scale systems, with a goal of extending performance contracts to treat system and application failures. The overall goal of this work is to create an intelligent performance toolkit that allows software developers to create and runtime systems to negotiate "performance contracts" among software components and hardware/software systems. The result will be tested and deployed in collaboration with LANL researchers.

***FY03 Tasks:***

- Extend performance contracts to reason about global (multiple task) behavior.
- Explore scalable measurement and prediction techniques that statistically characterize the behavior of systems with thousands of processors.
- Extend compact representations of dynamic behavior and failure modes.
- Deploy and test extended performance toolkit with LANL researchers.

### 1.4.4. Performance Tools

**Investigators: Rob Fowler, John Mellor-Crummey**

Parallel applications are becoming increasingly multidisciplinary, with libraries and other application components implemented using diverse programming languages, models, and parallelization strategies. In consequence, it is now extraordinarily difficult to achieve high fractions of peak hardware performance on large-scale parallel systems, emerging networks of workstations. This project will focus on refining and deploying the HPCView toolkit – a set of compiler-assisted performance tools that help users understand where and how an application's demands do not match an architecture's capabilities. This toolkit helps explain program performance by collecting, correlating, and displaying information about program structure, transformations by optimizing compilers, and dynamic performance measurements.

***FY03 Tasks:***

- Deploy open-source HPCView toolkit for measurement and analysis of application node performance for use on ASCI workloads on Pentium, IA64, MIPS, and Alpha architectures.
- Work with LANL application groups to evaluate performance and tuning opportunities in open ASCI workload codes accessible to academic partners (including Truchas).
- Explore enhancements to measurement and presentation tools in HPCView toolkit to reflect dynamic execution structure and parallel overhead.

## 1.5. Computer Science Community Interactions

Fostering collaborative relationships between LACSI participants at LANL and at the LACSI academic sites is a principal LACSI goal. Because LACSI is a collaborative research effort, effective means of supporting collaborations are important to LACSI's success. To encourage collaboration, the LACSI academic institutions support a variety of opportunities for researchers and students from Los Alamos and the academic partner sites to visit each other, to share ideas, and to actively collaborate on technical projects. LANL has also hosted speakers from the LACSI academic sites as part of the ACL Seminar Series. Researchers from the LACSI academic sites are available to speak in the ACL Seminar Series during the FY03 project year.

In 1999, Rice University established the LACSI Speaker Series as another means of encouraging collaborative relationships between researchers at LANL and at the participating academic institutions, particularly Rice and UH. Since its inception, Rice has hosted twelve LANL speakers. Speakers during the last project year included Wu Feng, Brian VanderHeyden, Doug Kothe, Dan Quinlan, and Rich Graham. Adolfy Hoisie is scheduled to speak on November 12, 2002. We plan to host additional LANL speakers during this project year. For more details on the LACSI Speaker Series at Rice, visit http://hipersoft.rice.edu/lacsi/speaker_series.htm.

In addition to hosting visitors and speakers, the LACSI academic partners, in conjunction with LANL, will organize, host, and otherwise support a series of technical workshops on topics related to the LACSI technical vision. Planned activities include a series of workshops at LANL targeted at exposing application researchers to emerging technologies.

To reach a broader community, LACSI hosts an annual symposium to showcase LACSI results and to provide a forum for presenting outstanding research results from the national community in areas overlapping the LACSI technical vision. This is a traditional conference-style meeting with participation by both LACSI members and scientists from the community at large. The LACSI Symposium will be held October 14 – 16, 2002 in Santa Fe, New Mexico. Details related to the LACSI symposium are available at http://hipersoft.rice.edu/lacsi/symposium_2002.htm. In addition, the LACSI academic partners in conjunction with LANL will disseminate information about LACSI and its research results at Supercomputing 2002.

Finally, Rice will also coordinate a technical infrastructure between Los Alamos and the academic partners, enabling web broadcasting of local technical talks, workshops, and the LACSI Symposium to an off-site audience.

# 2. Management and Administration

Andy White (LANL) directs LACSI in conjunction with Ken Kennedy (Rice), who serves as co-director of LACSI and director of the academic portion of the LACSI effort. John Thorpe (LANL) and Linda Torczon (Rice) assist the directors as executive directors. The directors make significant decisions with the advice of the LACSI Executive Committee (EC), which includes the site director for each of the six LACSI sites, key LANL researchers, the project directors for the academic portion of each of the five strategic thrusts, and the executive directors. The EC currently consists of the following members:

- Andy White, LACSI Director, Chair, *Los Alamos National Laboratory*
- Ken Kennedy, LACSI co-Director, co-Chair, *Rice University*
- Jack Dongarra, *University of Tennessee at Knoxville*
- Bill Feiereisen, *Los Alamos National Laboratory*
- Rob Fowler, *Rice University*
- Lennart Johnsson, *University of Houston*
- Deepak Kapur, *University of New Mexico*
- Doug Kothe, *Los Alamos National Laboratory*
- John Mellor-Crummey, *Rice University*
- Rod Oldehoeft, *Los Alamos National Laboratory*
- Dan Reed, *University of Illinois at Urbana-Champaign*
- Dan Sorensen, *Rice University*
- John Thorp, *Los Alamos National Laboratory*
- Linda Torczon, *Rice University*

The EC is responsible for planning and reviewing LACSI activities on a regular basis and establishing new directions, along with new goals and modified milestones. The EC evaluates progress based on the quality and relevance of the research being done, and how the goals of LACSI are being met. Based on the outcomes of its reviews of the research and the other LACSI activities, the EC might propose a collection of projects to be undertaken, along with goals for those projects, and identify projects to phase out. LACSI researchers to lead the new efforts would be identified and work would be initiated. The resulting work would be evaluated in subsequent reviews.

In March 2002, the EC met with LACSI researchers at LANL to discuss methods of addressing issues raised in the 2001 LACSI contract review. The group developed a framework to address long-term strategic thrust areas. Specific objectives were called out as near-term priorities. The objectives were folded into the framework to form a coherent planning view. A description of the long-term vision, framework, and objectives is available in a draft document titled *Priorities and Strategies*. The EC will review *Priorities and Strategies* annually and update the document as appropriate to reflect LANL's long-term priorities. Relevance to the LACSI priorities and strategies outlined in the document will be a key evaluation criterion used when the EC evaluates progress on LACSI projects.

The EC meets by teleconference bi-monthly and communicates regularly by e-mail. The EC also meets in person at the LACSI Symposium every October and, typically, at another point in the spring.

## 2.1. Management of Academic Subcontracts

Rice is the lead site on the contract for all academic partners, with Ken Kennedy serving as director. Linda Torczon assists him as executive director. Rana Darmara assists him as senior project administrator. Each academic site has a site director: Ken Kennedy (Rice University), Lennart Johnsson (University of Houston), Dan Reed (University of Illinois at Urbana-Champaign), Deepak Kapur (University of New Mexico), and Jack Dongarra (University of Tennessee at Knoxville). Each of the five strategic thrust areas has a project director: Ken Kennedy (Components), Rob Fowler (Systems), Dan Sorensen (Foundations of Computational Science), John Mellor-Crummey (Application Performance), and Linda Torczon (Computer Science and Community Interaction). Significant decisions related to the management of the academic subcontracts are made by the Director with the advice of the academic site directors and the academic project directors.

The LACSI directors, the LACSI executive directors, and key research and administrative personnel from LANL and Rice meet bi-weekly by teleconference to handle administrative matters related to contracts, invoicing, meeting arrangements, reporting requirements, and other administrative issues that arise

## 2.2. Computational Resources

The academic partners will be provided with access to ASCI computing platforms at LANL on a predetermined basis for development and testing. The process will make it possible to allocate a small cluster of nodes each week and a larger cluster of nodes once a month. It is understood that dedicated access may be needed for key tests and performance analyses.