Contract No: 86192-001-04 49

Provide Medium to Long-term Computer Science Research Relevant to the Goals of the Advanced Simulation and Computing (ASC) Program

LACSI Task 3

Los Alamos Computer Science Institute External Computer Science Research

FY04 Statement of Work – REVISED

April 15, 2004

- **Components**

This effort will incrementally develop two long-term goals.

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.

Develop a *collection of components for use in science and engineering applications*. The algorithms should be general, portable, and usable in a variety of situations.

Specific tasks and deliverables are described in Appendix A, Section 1.1.

- **Systems**

Provide advanced development of computer subsystems, both hardware and software, of strategic interest to present and future ASC architectures. The specific tasks and deliverables are described in Appendix A, Section 1.2.

- **Computational Science**

Provide research and development in the areas of numerical methods for partial differential equations, linear and nonlinear solvers, and verification and validation methodologies to address the following goals:

Develop, analyze and implement numerical methods for coupled multi-physics. Particularly important is the discretization of partial differential equations by mixed and hybrid finite element type methods.

Develop linear and nonlinear solvers algorithms.

Develop software tools suitable for large-scale simulation codes and automation of the process of tuning those codes for efficiency on specific platforms.

The specific tasks and deliverables are described in Appendix A, Section 1.3.

- **Application and System Performance**

Develop compiler and run-time technology that will help application developers achieve a high fraction of peak performance on large-scale parallel computing systems. The specific tasks and deliverables are described in Appendix A, Section 1.4.

- **Computer Science Community Interaction**

Foster collaborative relationships between LACSI participants at LANL and at the LACSI academic sites using activities including:

- Host visitors and speakers at academic sites to encourage collaborative relationships between researchers at LANL and at the participating academic institutions.
- Support scientific collaboration visits to Los Alamos National Lab to integrate research products into operations.
- In conjunction with LANL, organize, host, and otherwise support a series of technical workshops on topics related to the LACSI technical vision.
- Host 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.

Specific tasks and deliverables are described in Appendix A, Section 1.5.

Project Management

The implementation strategy and tasks associated with the academic LACSI projects are detailed in Appendix A. Documentation regarding the objectives of the Institute may be found in the latest LACSI “Priorities and Strategies” document (LAUR # 03-7355).

Deliverables

- Quarterly status reports on milestones/deliverables

Other reports and briefings as required.

Appendix A

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 New Mexico (UNM), the University of Illinois at Urbana-Champaign (UIUC), the University of North Carolina at Chapel Hill (UNC), 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 University 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 (EC) 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 developed at the meeting is available in a document (LAUR #02-6613) titled *Priorities and Strategies*.

In April 2003, the LACSI EC met with senior LANL personnel to revise the framework, priorities, and strategies established at the planning meeting in 2002 and *Priorities and Strategies* was revised to incorporate the results of the April 2003 planning meeting (LAUR # 03-7355). In February 2004, the EC again met with senior LANL personnel to revise the framework, priorities, and strategies established in previous planning meetings. *Priorities and Strategies* is being revised to reflect the results of the February 2004 planning meeting. The current framework outlines five strategic thrust areas:

- Components
- Systems
- Computational Science
- Application and System 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, Bradley Broom, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, Charles Koelbel, John Mellor-Crummey, Dan Reed, Jaspal Subhlok

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.

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 areas of expertise. For example, components and modern scripting languages enable 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, making it possible for 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 is to explore 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 the remainder of this section.

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: <u>Ken Kennedy</u>, Jack Dongarra, Richard Hanson, Lennart Johnsson, Charles Koelbel

We will assist the LANL staff in conducting a study of high-performance components and associated frameworks available to scientific programmers. The goal of this review is to identify a set of components that can be effectively integrated into strategic LANL applications and into the LANL software culture. In particular, the new ASC weapons-code project, to be based on a component architecture, is identified as a key customer of this review.

In FY03, this subproject convened a components workshop that has illuminated some of the critical issues in integrating component methodologies and technologies into the laboratory codes. At this workshop, several high-performance applications were demonstrated, each of which were based on easily adaptable component architectures. A report from this workshop has been produced and the issue of high-performance components and component-based architectures are under continued study because of the importance of this subject to the future of software development at the laboratory.

This study will continue into the next year to produce a set of recommendations concerning components, component architectures, component integration frameworks, and any additional research that is needed to make these software systems practical to strategic LANL applications. 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, a major consideration should be given to projects in the DOE SciDAC (Scientific Discovery through Advanced Computing) initiative, such as the Common Components Architecture (CCA) effort. In examining recent advances, the team should consider the ease of adapting existing codes to take advantage of new software technologies.

Tasks:

- Produce a report summarizing findings and recommendations of the one-year study (Quarter 1).

1.1.2. LACSI Component Integration Challenge Problem

Investigators: Ken Kennedy, Jack Dongarra, Richard Hanson, Charles Koelbel, John Mellor-Crummey

One of the most difficult challenges for component integration is the problem of integrating data structure components (e.g., sparse matrices) with functional components (e.g., linear algebra). This problem is hard because the frequency of invocation of data access methods places a premium on high performance of the component interfaces. The long term-research section of the proposal has taken this as a major focus for the next several years.

To drive this research in directions that are most useful to LANL, we will work with LANL staff to define in the next year a challenge problem by specifying the interfaces to a data structure component that would be useful in LANL weapons codes. These interfaces will be developed through a joint study between code developers and computer and computational scientists within LACSI. A goal of this effort is to produce interfaces and define functionality that could be prototyped in the telescoping languages system for efficient component integration that is the subject of LACSI research. The ultimate goal is that the data structure component be made efficient enough for use in production weapons codes.

Tasks:

- Plan a series of meetings to be conducted through the second and third quarter (Quarter 1).
- Produce a report defining the interfaces to the LACSI challenge problem data structure (Quarter 4).

1.1.3. Component Architectures for Problem Solving Environments in a Networked Environment

Investigators: Ken Kennedy, Bradley Broom, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, John Mellor-Crummey, Dan Reed, Jaspal Subhlok

Generation of Problem-Solving Systems Through Component Integration

The goal of this research 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 effort 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 will pursue 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 attack the problem of making component integration efficient enough to be practical for high-performance scientific codes. Of particular importance in this context is the problem of efficiently integrating data structure components (e.g., sparse matrices) with functional components (e.g., linear algebra). This work will begin with a simple prototype of Matlab that includes arrays with data distribution. Specific array distributions for sparse matrices will be explored as a way of understanding the crucial performance issues. In the long term, this may lead to a new strategy for introducing parallelism into Matlab—by distributing the arrays across multiple processors and performing computations close to the data.

Once the Matlab array prototype has been explored, we will focus on the LACSI Challenge Problem data structure with the goal of demonstrating a prototype with adequate efficiency for use in production codes.

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 four fundamental directions:

Toolkits for Building Problem-Solving Systems: The 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 ASC 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 in this section, we will pursue the novel strategy of "telescoping languages" to make it possible to extend existing languages through the use of software components.

Advanced Component Integration Systems: This effort will explore the application of telescoping languages technology to the component integration problem, with a particular emphasis on integrating components that support data structures with those that implement functionality. The effort will begin with an emphasis on integration of distribution into Matlab arrays. If successful, this technology will open the door to high-level parallelization strategies for Matlab based on data distribution. The effort will also consider technologies for optimizing accesses to the component interfaces emerging from the LACSI challenge problem described earlier. The long-term goal of this research is to produce a component integration framework that is efficient enough to be accepted by developers of high-performance applications, such as LANL weapons code developers.

An additional topic of research in this area is the design of component integration systems for distributed computing environments or Grids. The goal of this effort is to make such component accesses efficient through high-level optimizations that minimize the effect of long and variable latencies.

Design for Efficient Component 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. A specific issue of importance is design strategies for efficient data structure components

Component Systems for Heterogeneous Computing Systems: The key challenge in this area is to construct applications that can be flexibly mapped to heterogeneous computing components and adapt to changes in the execution environment, detecting and correcting performance problems automatically. In this activity, we will explore the meaning of network-aware adaptive component frameworks and what the implementation and optimization challenges are for applications constructed from them. In addition, we will pursue research on middleware to support optimal resource selection in heterogeneous environments. A major byproduct of this work will be performance estimators (described in the section "Modeling Application and System Performance") and mappers that can be used to map applications efficiently to heterogeneous computing systems, such as distributed networks (e.g., grids) and single-box systems containing different computing components (e.g., vector processors and scalar processors).

Compilation of Object-Oriented Languages: Object-oriented languages like Java have 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. However, it is still not considered efficient enough for most production applications. In this effort we are studying strategies for the elimination of impediments to performance in object-oriented systems.

To achieve this goal, we must develop new compilation strategies 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. In particular, we will explore some of the issues of compiling object-oriented features in Matlab.

Tasks:

- Produce a simple Matlab to C compiler with type disambiguation eliminated. (Quarter 1)
- Design the strategy for adding distributed matrices to Matlab and delivering a report on the design (Quarter 2, jointly between Rice and Tennessee)
- Finalize plan for compiling object-oriented features in Matlab (Quarter 3).
- Produce a design strategy for the LACSI challenge problem (Quarter 4)
- Deliver the JaMake framework for use at Los Alamos. (Quarter 4)

1.1.4. Retargetable High Performance Components and Libraries

Investigators: Jack Dongarra, Lennart Johnsson, Ken Kennedy

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.

Automatically Tuned Library Kernels: 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. We will extend this work to systems that can automatically generate the tuning search space for new libraries using automatic application tuning methodologies described in the "Application and System Performance" section of this document

Self-Adapting Numerical Software: We will explore new approaches to building adaptive numerical software that overcomes many of the deficiencies of current libraries. An adaptive software architecture has roughly three layers. First there is a layer of algorithmic decision making; the top level of an adaptive system concerns itself with the user data, and based on inspection of it, picks the most suitable algorithm, or parameterization of such algorithms. The component responsible for this decision process is an `Intelligent Agent' that probes the user data and based on heuristics chooses among available algorithms. Second there is the system layer; software on this level queries the state of the parallel resources and decides on a parallel layout based on this. There can be some amount of dialog between this level and the algorithmic level, since the amount of available parallelism can influence algorithm details. Finally, there is the optimized libraries level; here we have kernels that provide optimal realization of computational and communication operations. Details pertaining to the nature of the user data are unlikely to make it to this level. Implicit in this approach is a distinction of several kinds of adaptivity. First of all, there is static adaptivity, where adaptation happens during a one-time installation phase. Contrasting with this is dynamic adaptivity, where at run-time the nature of the problem and environment are taken into account. Orthogonal to this dichotomy is the distinction of adapting to the user data or the computational platform. We stress the obvious point that, in order to adapt to user data, a software system needs software that engages in discovery of properties of the input. Oftentimes, such discovery can only be done approximately and based on heuristics, rather than on an exact determination of numerical properties.

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 ASC program.

Tasks:

- Investigate optimization techniques for basic linear algebra for Intel Itanium and AMD Opteron processors. (Quarter 2)
- Construct algorithm selection based on input data with heuristic choice (Quarter 2)
- Plan to expand ATLAS-style tuning to sparse linear algebra and cluster numerical library (Quarter 2)
- Investigate the use of the methodology used for the UHFFT and the tools developed for creating the UHFFT library for MultiGrid applications at LANL (Quarter 3).
- Develop framework for self adaptation of numerical libraries on clusters (Quarter 3)
- Characterization of search space for self-adapting numerical software as a function of input data (Quarter 3)
- Begin automatically generate the tuning search space for new libraries using automatic application tuning methodologies (Quarter 4)
- Deploy and evaluate the UHFFT in at least two LANL applications in collaboration with LANL. (Quarter 4).
- Explore performance evaluation and tuning of the UHFFT on LANL platforms, Itanium systems with Myrinet and Opteron systems with Infiniband (Quarter 4).

1.2. Systems

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

1.2.1. Compiler Support for Scalable Parallel Programming

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

Today, MPI is the programming model used to write scalable parallel programs. However, as a programming model, MPI has several shortcomings. Explicitly coding communication using MPI primitives is tedious and error prone. Also, writing MPI programs to achieve top performance increases programming complexity: application developers must choreograph asynchronous communication and overlap it with computation. Furthermore, because of the explicit nature of MPI communication, significant compiler optimization of communication is impractical. Programming abstractions in which communication is not expressed in such a low-level form are better suited to having compiler optimization play a significant role in improving parallel performance.

SPMD programming models such as Co-array Fortran (CAF) and Unified Parallel C (UPC) offer promising near-term alternatives to MPI. Programming in these languages is simpler: one simply reads and writes shared variables. UPC and CAF provide programmers with virtually the same level of control over data distribution, parallelization, and communication placement as MPI, yet they boost programmer productivity by simplifying application development. Specifically, they free users from managing implementation details of communication; they make it practical to offload communication optimization to compilers by standardizing the expression of data movement and synchronization; and they improve performance portability by enabling compilers to tailor the details of communication implementation to the target platform at hand. Research into compiler optimizations for SPMD programming languages offers the potential of not only simplifying parallel programming, but also yielding superior performance because compilers are suited for performing pervasive optimizations that application programmers would not consider employing manually because of their complexity. Also, because CAF and UPC are based on a shared-memory programming paradigm, they naturally lead to implementations that avoid copies where possible; this is important because on modern computer systems, copies are costly. Today, UPC and CAF are relatively immature, as is compiler technology to support them. Research is needed to refine the language primitives (including language support for parallel I/O), to design effective compiler analysis and optimization strategies for SPMD programs (including strategies for generating latency-tolerant code), and efficient runtime systems.

Beyond explicitly parallel SPMD programming models, implicitly parallel programming models such as High Performance Fortran offer an even simpler programming paradigm, but require more sophisticated compilation techniques to yield high performance. Research into compiler technology to increase the performance and scalability of data-parallel programming languages as well as broaden their applicability is important if parallel programs are to be significantly simpler to write in the future.

FY04 Tasks:

- Demonstrate the Rice Co-Array Fortran compiler on HP AlphaSC systems. (Quarter 1)
- Refine data parallel code generation techniques for complex regular problems. Produce a report. (Quarter 1)
- Experiment and report on experiments with Co-Array Fortran version(s) of a LANL application on ASCI Q. (Quarters 2 and 3)
- Report on evaluation of compiler-generated parallel code on scalable systems. (Quarter 3)
- Report on definition of and experiments with Co-Array Fortran language refinements and optimization techniques. (Quarter 4)

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

Investigators: Scott Rixner, Alan Cox, Vijay Pai

The goal of this research is to investigate the performance tradeoffs of using Ethernet and TCP in cluster computing. Specialized networks, such as Quadrics and Myrinet, are typically used in cluster computing because of their higher bandwidth and lower latency. However, raw Ethernet is extremely competitive both in terms of bandwidth and latency when cost is considered. The drawbacks of Ethernet typically arise because of the way that it is used both by the operating system and the MPI library. With specialized networks, the protocol processing is usually handled directly in the MPI library. By doing so, the transport protocol can be tailored specifically to the cluster-computing domain, which reduces latency, and copying can be minimized by using such techniques as remote DMA.

In TCP, the protocol processing is handled within the operating system, which can be much more efficient. Currently, however, the use of TCP for MPI degrades performance, as TCP is designed to work over the Internet, rather than in the relatively controlled network of a computing cluster. Some of the most significant TCP overheads relate to copying between the application-level and kernel-level on network sends and receives. Although these problems have been solved in the past for network sends, generally applicable and easily programmable zero-copy receives remain an open problem.

Despite these drawbacks, using TCP over Ethernet has several advantages if its performance can be made competitive. First, Ethernet is clearly less expensive than specialized networks. Second, TCP provides reliability and easy portability across systems. Network servers are able to achieve extremely high performance levels with TCP, using scalable event notification systems, such as /dev/epoll in Linux, zero-copy I/O, and asynchronous I/O.

We intend to show that operating system advances for network servers and programmable network interfaces can be used to allow TCP over Ethernet to achieve competitive performance for cluster computing. Specifically, programmable network interfaces can be used as a mechanism for receive copy-avoidance. These network interfaces allow unexpected data to be buffered until they are needed by a receive posted by the application. Our goals are to encapsulate all the needed changes for high-performance MPI over TCP in the network interface hardware, the network interface firmware, and modified operating system drivers. We aim to avoid any changes to the MPI implementation itself or to the application software that are specific to our hardware, thus eliminating the need to continually re-implement MPI for each new class of systems.

To test the effectiveness of these techniques, we intend to add a TCP layer to LA-MPI. This will enable us to understand the bottlenecks related to networking and network interface and how the details of the network interface affect performance. We intend to use that information to develop modified MPI implementations that use TCP efficiently and can be well integrated with the newly designed network interfaces.

FY04 Tasks:

- Develop a TCP implementation of LA-MPI which uses many of the techniques from today's high-performance network servers (Quarter 1)
- Complete a detailed performance analysis comparing (1) TCP LA-MPI, (2) other TCP MPI implementations which all use TCP in a more naïve way, and (3) specialized, i.e. non-commodity, networks. (Quarter 4)

Long Term Tasks:

- Write a paper, based on the above performance analysis, on the problems and proposed solutions in these systems. (Quarter 1 FY05).
- Demonstrate and release a prototype system using LA-MPI/LAM, Linux, and a programmable NIC. (Quarter 4, FY05)
- Eliminate bottlenecks in TCP over Ethernet MPI implementation using a combination of advanced OS support and programmable network interfaces. Specifically, streamline the use of TCP over Ethernet for cluster computing and to eliminate copying on both message sends and receives.
- Show that MPI using TCP over Ethernet can be competitive with specialized networks.

1.2.3. Adaptability in Communication Structures for Commodity Clusters

Investigators: Arthur Maccabe and Patrick Bridges

This research addresses the development of adaptable communication structures for commodity clusters. In particular, we consider the adaptability in the implementation of two commonly used APIs: the standard Unix sockets API and MPI (Message Passing Interface).

In the past year, we have investigated improvements to TCP in the context of programmable NICs (Network Interface Controllers) and dual processors systems. The goal of this research is to investigate distribution of protocol processing to improve bandwidth, latency, and processor overhead. Prior work demonstrated the benefits to bandwidth and overhead to be gained by offloading IP fragmentation and re-assembly. In the past year, we used programmable NICs to investigate the benefits of ACK generation on the NIC. This research was aimed at improving the latency and gap for very short messages. We also developed a "virtual NIC" capability for dual processor systems that allows us to isolate communication activities on one of the two processors in a dual processor system.

We will build on our earlier efforts and address the need for adaptability in the implementation of communication APIs for computational clusters.

Our exploration of ACK generation was motivated by a desire to improve the performance of MPI over TCP, especially that seen by MPI applications. While this did not gain an appreciable improvement in latency, we were able to reduce the gap between MPI messages. In this task, we will consider a modified socket interface to support a method of caching connection information using equivalence classes, and TCP interoperability and fairness with respect to open connections.

Our use of the second processor on a dual processor system to emulate a virtual NIC, while intended to simplify the development of code to be run on a NIC, has lead us to consider the development of a general capability of partitioning the services provided by Linux among the processors of an SMP. The goal is to provide a highly predictable execution environment by controlling the overhead associated with providing these services.

We will begin our investigation of adaptation in placing network error checking and handling will be done by developing a very lightweight transport protocol that provides an unreliable byte stream in the Linux 2.6 kernel. We will then port LAMPI to take advantage of this transport and compare the performance of this LAMPI implementation to the traditional LAMPI implementation based on UDP.

FY04 Tasks:

- Implement a GUI to configure the placement of Linux 2.6 system services in an SMP node. (Q1)
- Deploy a Linux 2.6 kernel module that will provide lightweight transport layer for LAMPI. This module will provide no error checking or flow control, recognizing that these aspects of communication are handled by LAMPI. (Quarter 2)
- Release a NIC-caching-based implementation of TCP. This implementation will cache open socket connections. (Quarter3)
- Deliver a study comparing RMPP (a reliable message transport) with the lightweight transport developed for FY04-Q2. (Quarter 4)

1.2.4. Compiler and Tool Support for Fault-Tolerant Programming

Application software must contribute to the overall design for reliability as well as scalability on commodity clusters. For some algorithms, notably the “bag of tasks” model, recovery from failure is relatively easy, when unfinished tasks and collected partial results are saved to disk.

Algorithmic studies will be carried out to identify other paradigms that can be adapted to run through failure, and a significant application will be identified that can be made resilient.

Today, programmers typically write programs that include explicit checkpointing. For data-intensive programs, one promising approach is to store persistent data on disks and use dynamic assignment of coarse-grain tasks to collections of processors, coupled with transaction-like recording of computation results. Compiler support to simplify data management would help ease the transition from conventional checkpointing to fault-tolerant models for high-performance programming. Alternative techniques based on enhanced hardware support and smoothly degrading algorithms are also potentially applicable.

FY04 Tasks:

- Hold a workshop at LANL to identify the current state of the art and to identify the specific research directions to be taken in FY05-FY08. (Quarter 2).
- Report on workshop. (Quarter 3)

1.2.5. Scalable Fault Tolerance Task

Investigators: Dan Reed, Celso Mendes

As supercomputers scale to tens of thousands nodes, reliability and availability become increasingly critical. Both experimentation and theory have shown that the large component counts in very large-scale systems mean hardware faults are more likely to occur, especially for long-running jobs. The most popular parallel programming paradigm, MPI, has little support for reliability (i.e., when a node fails, all MPI processes are killed, and the user loses all computation since the last checkpoint). In addition, disk-based checkpointing requires high bandwidth I/O systems to record checkpoints.

To address these challenges, we will analyze the behavior of ASC-scale systems in terms of their availability and their sensitivity to component failures. Using this data, we will develop performability models that combine both fault-tolerance and performance for systems containing thousands of nodes. These models will include total time to solution as a function of failure modes and probabilities. They will build on analysis of common failure modes on large-scale systems, based on failure logs and component reliability. The results of these models will shape design and implementation of fault-tolerance checkpointing libraries.

We will also explore real-time monitoring of system failure indicators (e.g., temperature, soft memory errors and disk retries). The implementation estimates node failure probabilities and introduces enough redundancy to enable recovery. It complements disk-based checkpointing schemes to recover from failures between disk checkpoints. We envision it is a low overhead checkpoint alternative that can be performed much more often than disk checkpointing, triggered either periodically or via system measurements. Finally, we expect this approach to include intelligent learning and adaptation. By monitoring and analyzing failure modes, the system can estimate the requirements adaptation to achieve a specified reliability. This will enable smoothly balancing performance and reliability.

FY04 Tasks:

- Demonstrate measurement infrastructure for detecting component failure modes. (Quarter 4)

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

Investigator: Thomas P. Caudell

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 that 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 textural 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.

Based on the results from the 2003 work, this task will refine the design a set of pilot human subject experiments that attempt to quantify some subset of these issues. This will begin by revising the relevant parallel program information to be visualized, and devising methods for applying multiresolution techniques to that information. Monitoring and instrumentation of parallel programs will continue to be explored, using off-the-shelf tools as much as possible. As in 2003, either a new or revised human visualization task will be identified, in collaboration with LACSI colleagues, where performance is measured by a combination of time and accuracy. The process of designing and refining these tests will begin, and the supporting software will be revised. Finally, a pilot study will be performed. Results will be summarized in a report and paper intended to be presented at the LACSI Symposium.

The research conducted in 2003 laid the groundwork for the proposed 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.

FY04 Tasks:

- In conjunction with LANL collaborators, write a report on the design and codification of an appropriate experiment. (Quarter 1)
- Release a revision of UNM software to support new visualizations of large parallel programs, (Quarter 2)
- Collect experimental data. (Quarter 2)
- Analyze experimental data. (Quarter 3)
- Write a report on the results of the experiment. (Quarter 4)

1.3. Computational Science

Investigators: Mike Fagan, Bill Symes, Matthias Heinkenschloss, Yin Zhang, Petr Kloucek, Mark Embree, Dan Sorensen, Lennart Johnsson, Yuri Kuznetsov, Roland Glowinski

The *Computational Science* effort focuses on the development, analysis, and verification and validation (V&V) of numerical solution techniques for physical models embodied within large-scale multi-physics simulation tools designed to address today's leading problems in science and engineering. Key applications currently include the predictive simulation of weapons manufacturing and performance as supported by the DOE Advanced Simulation and Computing (ASC) Program and global climate modeling as supported by the DOE Scientific Discovery Through Advanced Computing (SciDAC) Program. The computational science effort can be divided into three principal research thrust areas: algorithms and models for specific physical phenomena of interest, numerical methods for the algorithmic coupling of these physical phenomena, and metrics for correctness and robustness of these models and algorithms. The thrust areas are:

1. Numerical solution of partial differential equations for continuum dynamics, energy transport, and materials science;
2. Linear and nonlinear solvers; and
3. Methodologies for V&V, Sensitivity, and Uncertainty Quantification.

A key product of this effort, both in the long and short term, is verified and validated software components constructed with defensible (demonstrable) software quality engineering practices. These components must instantiate robust and accurate solution techniques for the physical models required by the multi-physics simulation tools. The computational science effort devoted to "multi-physics coupling" algorithm research is necessary for the faithful simulation of multiple, simultaneously-occurring physical phenomena.

Long-term goals. Ensuring computational science follows the fundamental principles of the scientific method requires long term investigation of numerical methods and algorithms and careful software development. For example, a physicist or engineering analyst using these simulation tools should be able to generate high fidelity three-dimensional simulations, attain similar answers with two different numerical techniques, and be assured that each technique has been verified and validated. Because the transformation of physical principles into software can take many different paths, long-term research focuses on the investigation of new, possibly high-risk, methods along with new ideas for the improvement of classical methods that are parallel and scalable.

Experience shows investigation of new methods must be built upon the foundation of good software quality engineering. Unit-testing and component-based designs for even one-dimensional tests are necessary to assess the impact of this long-term research on next-generation simulation tools.

Long-term goals of the computational sciences effort include:

- Understanding the physics and mathematics of the phenomena to be simulated so that improved numerical methods can be devised that are both robust and accurate;
- Developing new algorithms for the resulting physical models that possess good single processor performance as well as being parallel and scalable;
- Instantiating these algorithms into component-based software as guided by sound software quality engineering practices. Unit-testing is of primary importance compared to reusability;
- Developing improved and automated methodologies for the verification of the algorithms and the software and the validation of the models; and
- Devising strategies for successful team software development of large-scale simulation tools.

Short-term goals. In the short term (< five years), the *Computational Science* effort must complement and deliver to the LANL ASC Computational Sciences Program Element (CompSci PE). As one of eight PEs within the LANL ASC Program, the principle mission of CompSci PE is to deploy verified and validated software components embodying shock hydrodynamics, radiative and neutron transport, and linear/nonlinear solvers. It must also deliver simulation tools for weapons performance (the Marmot Project) and weapons casting and welding processes (the Telluride Project). A notable short-term goal of the LACSI Computational Science effort is to deliver software components to the three critical ASC "code projects", known as collectively as the Crestone, Shavano, and Marmot Projects. Meeting these L2 Milestones within the CompSci PE is an absolute must, as they feed into and provide enabling technology for other, higher-level milestones within the LANL ASC Program. These technologies will also form the core constituents of next-generation LANL performance and manufacturing simulation tools.

The short-term goals for the Computational Science effort provides help to the Computational Science PE to meet its ASC goals.

Research Thrust Areas

For success in the long and short term the Computational Science effort has the following top three research priorities:

1. Numerical solution of partial differential equations for continuum dynamics, energy transport, and materials science;
2. Linear and nonlinear solvers; and
3. Methodologies for V&V, Sensitivity, and Uncertainty Quantification.

The following sections describe each piece in more detail.

1.3.1. Numerical solution of partial differential equations

Subproject leads: Lennart Johnsson, Yuri Kuznetsov, Roland Glowinski

LANL Contacts: J. Morel (CCS-4), J. Sicilian (CCS-2), W. Rider (CCS-2)

Here research is focused the development, analysis, and application of physical models and numerical methods for the simulation of key physical processes pertinent to LANL applications.

Continuum dynamics refers principally to the high deformation and high strain rate of shock-loaded fluids and solids that are bounded from other fluids and solids by complex topology interfaces. Two major computational techniques dominate methods research and development for continuum dynamics: Lagrangian (fluid reference frame) and Eulerian (fixed reference frame) methods. In Lagrangian methods research, importance is placed upon mimetic, conservative discretizations and the ability to model material interfaces as slide lines. In Eulerian methods research, priority is placed on high resolution, material discontinuity-preserving (i.e., interface tracking) methods working in conjunction with adaptive mesh refinement (AMR). Both of these methods could benefit from research covering their implementation on terascale parallel machines.

Energy transport refers principally to the understanding and development of deterministic, stochastic, and hybrid methods and algorithms for numerical neutron and radiative transport phenomena. Of interest are new, robust and accurate methods that exhibit increased fidelity on both unstructured and adaptively refined structured (and orthogonal) meshes.

Materials science research is itself a very broad field, but in this context the focus is on predictive, physics-based methods for the continuum (macroscopic) simulation of manufacturing processes such as casting and welding operations of interest to LANL and the DOE Complex. These operations are of interest because of current weapons remanufacturing requirements. Modeling such operations requires accounting for incompressible free surface fluid flow, radiative/conductive/inductive heat transfer, and solid/liquid phase change (including subgrid microstructure models).

1.3.1.1. *Parallel Numerical Methods for the Diffusion Equation in Heterogeneous Media on Strongly Distorted Meshes*

Investigators: Yuri Kuznetsov, Roland Glowinski, Lennart Johnsson

LANL Contacts: J. Morel (CCS-4), M. Shashkov (T-&), M. Berndt (T-7), D. Moulton (T-7)

Efficient parallel numerical methods for the diffusion equation 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, for instance, to the T-7 and CCS-4 groups at LANL. The researchers from the LANL part of the project are experienced in accurate and physically consistent approximations to the diffusion equations on strongly distorted meshes as well as in applications of advanced numerical methods to real-life scientific and engineering 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 designing of efficient parallel iterative solvers based on a combination of domain decomposition, fictitious domain, and multilevel techniques.

Short-term goal. The first main objective of the project is to develop and investigate new accurate, physically consistent, and convenient methods for applications approximations to the 3D diffusion and equations with heterogeneous anisotropic coefficients on strongly distorted, logically rectangular meshes. These techniques will be quantitatively compared against existing approaches at LANL (e.g., those devised by J. Morel and M. Shashkov) using standard LANL test problems.

Long-term goal. The second main objective of the project is to design, investigate, and implement on parallel computers new fast iterative solvers for large-scale algebraic systems resulting from mesh discretizations. The expected size of mesh problems is 10^7 - 10^8 degrees of freedom.

Milestones:

- To investigate new mimetic and mixed finite element discretizations on degenerate and nonconvex polyhedral macro-cells for 3D diffusion equations (March 2004);
- To develop and investigate new multilevel preconditioners for 3D diffusion equations in the case of strongly distorted polyhedral meshes (September 2004).

1.3.1.2. *Modeling and Simulation of Surface Tension Driven Flow. Efficient Methods for Strong Shock Calculations*

Investigators: Roland Glowinski, Yuri Kuznetsov, Lennart Johnsson, R. Sanders

LANL Contacts: M. François (CCS-2), D. Kothe (CCS-2)

Considering the new approach required from LACSI academic investigators, namely, their research has to be closely related to the one taking place at LANL, and following last spring joint meeting at Rice University, we have identified two main topics:

Modeling and Simulation of Surface Tension Driven Flow. Following the presentation of M. François (LANL) at USNCCM VII in Albuquerque (August 2003) and preliminary discussions we had with D. Kothe, U.H. researchers will investigate various issues concerning the modeling and simulation of surface tension driven flow. On the basis of recent results in two-dimension (article to appear in the CFD Journal) these investigators would like to develop a modular simulation methodology taking advantage of a time-discretization by operator splitting. The time-splitting will allow to use different approximations for the numerical treatment of advection, diffusion, incompressibility and free-surface; it will allow, for example, the coupling of finite element and finite volume methods, and taking advantage of existing modules. A crucial point is clearly the modeling and numerical treatment of the surface tension. As well known, the surface tension is proportional to $\mathbf{n}C_m$, where C_m is the mean curvature and $\mathbf{n}$ the normal vector at the free surface. Using appropriate variational techniques, generalizing those used in two-dimension, it is not necessary to compute explicitly the mean curvature to take the surface tension into account. One of the main objectives of these investigations is to develop a simulation methodology taking advantage of the above observation; the algorithmic gain should be substantial.

Strong shocks calculations. The simulation of phenomena involving strong shocks has been identified as a priority during last spring joint meeting at Rice University. Richard Sanders, at University of Houston, has developed computational methods for multi-dimensional shock computations. On the basis of numerous benchmark problems, R. Sanders methods have proved more accurate than ENO related schemes, as stable and no more costly. A first objective is to identify at those projects involving strong shock computations and develop a cooperative program from there.

Tasks:

- Develop a variational method for the modeling of the surface tension.
- Identify at LANL the projects where strong shock calculations are a priority.
- Implement a 3-D finite element simulator for surface tension driven for those situations where the equation of the free surface is of the form $z = f(x, y)$.
- Test shock calculation methodology on benchmark problems provided by LANL scientists.

1.3.2. Linear and nonlinear solvers

Subproject lead: D.C. Sorensen

LANL Contacts: Dana Knoll (T-3), Michael Pernice (CCS-3)

"Multi-physics coupling" refers to the algorithmic challenges posed when solutions to simultaneous sets of nonlinear PDEs (or even stochastic nonlinear systems), each representing a different physical process (e.g., advection, reaction, diffusion), must be found. Over the last several decades, numerical solutions to these coupled nonlinear systems of equations have been built up by "operator splitting" along physical processes. Not only is this practice poorly understood (in terms of its strengths and weaknesses), but the extent to which this approach approximates reality is sometime is question. In addition to this algorithm verification issue, the quest for algorithmic scalability remains. The solution of linear and nonlinear problems consumes a substantial fraction of computational time in physics-based simulation tools. Research focusing on scalable, parallel, linear and nonlinear solution algorithm development and deployment must therefore continue.

1.3.2.1. *Numerical Linear Algebra for Large Systems (Eigenvalue Methods and Software for ASC-MPP Systems)*

Investigators: D.C. Sorensen, Mark Embree

LANL Contacts: B. Nadiga (CCS-2), Jim Morel (CCS-4), Rob Lowrie (CCS-2), Dana Knoll (T-3), John Turner (CCS-2), Beth Wingate (CCS-2), Michael Pernice (CCS-3), Bobby Phillips (CCS-3), Jim Warsa (CCS-4)

We are working on Krylov and Newton-Krylov techniques for time integration and linear stability analysis of an ocean circulation model (OCM) developed by Nadiga. One goal of the project is to develop improved Matrix-free Newton-Krylov methods for solving these large-scale systems of nonlinear equations arising in the time integration of the dynamical system.

Preconditioners are an essential component to the efficient solution of large-scale multiphysics applications. Though theoretical properties of preconditioners are well understood, in practice it can be difficult to analyze the performance and scalability of preconditioners, for one typically does not explicitly construct the preconditioner operator, which could include multigrid cycles, domain decomposition, etc. However, Krylov subspace algorithms allow for the spectral properties of the preconditioned system to be approximated. We will investigate how this might be done on examples from LANL applications in magnetohydrodynamics and neutron transport. Such a study can reveal opportunities for solver acceleration, for example by using deflated GMRES algorithms.

Analysis of conditioning and nonnormality of preconditioners:

Short-term goals.

- Obtain access to the preconditioner used in Pernice's magnetohydrodynamic simulations.
- Perform an analysis of the conditioning and nonnormality of the preconditioned MHD system.
- Obtain from Jim Warsa sample linear systems arising in neutron transport k-eigenvalue calculations (performed by LANL's Attila software).
- Our recent results on the rate at which Krylov subspaces capture invariant subspaces can potentially contribute to our understanding of deflated restarted GMRES solvers. We will study the feasibility of incorporating our analysis in this context.

Long-term goals. The efficiency of a Newton-Krylov method is generally dependent upon the quality of the linear system preconditioner. The best preconditioners incorporate problem-specific information. We believe the information obtained from linear stability analysis can be used to build a better Newton-Krylov preconditioner.

We hope to use eigenanalysis to design improved linear system preconditioners that are adaptive. These preconditioners are to be integrated into the design of a Newton-Krylov solver for dynamics calculations in the OCM.

- Using approximations drawn from Krylov subspaces, we will investigate the spectral structure of the magnetohydrodynamics and neutron transport preconditioners. In particular, we will analyze eigenvector directions that are well approximated by Krylov subspaces and eigenvalue stability.
- Based on this analysis, we will analyze opportunities for effective design of a deflated restarted GMRES algorithm to accelerate solver performance. Deflation can potentially allow restarted GMRES to mimic the superlinear convergence observed for full GMRES.

Task:

- Produce a study of the possibility of extending our new understanding of the convergence behavior of restarting techniques for eigenvalue computations to better understand and design restarting techniques for iterative linear system solvers such as restarted GMRES.

1.3.2.2. *Methods and Tools for the Solution of Non-smooth, Multi-scale, Coupled Models*

Rice: Petr Kloucek, Pavel Solin

LANL: J. Brackbill (T-3), Marius Stan (MST-8), Chong Chang

We will concentrate on three areas of research in which we have ongoing collaboration with two groups in LANL. The first area is focused on developing coupling equations that can connect stochastic and deterministic models, the second area focuses on stochastic computational models of non-laminated microstructures, which drive phase changes in nonlinear materials, and the third area is centered on the development of efficient computational techniques for solving the Fokker-Planck equation.

The team members have developed a powerful stochastic approach to computational modeling of complicated crystalline materials. The computational methodology based on this theory allows for computer modeling of effective properties of composite materials, stochastic foams, and similar nano-to-micro-scale based materials. This project will focus upon the required solution techniques.

Short-term goals.

- Develop transmission conditions for exchange of information among stochastic and deterministic models of material behavior based on the overlapping domain decomposition method in which both deterministic and stochastic processes are active. The initial effort will be focused on one spatial dimension. We will extend our previous attempt done in collaboration with Dr. J. Brackbill during the summer internship of graduate student Jennifer Wightman.
- Develop a Fokker-Planck type equation modeling non-laminated microstructures in Martensitic materials. The derivation of the Fokker-Planck equation hinges on an appropriate Langevin system describing microscopic properties of atomic lattice.

Long-term goal.

- Mathematical and computer modeling of transformation toughening in zirconium-type ceramics. Transformation toughening is the increase in fracture toughness of a material that is the direct result of a phase transformation occurring at the tip of an advancing crack. The discovery of transformation toughening in zirconium ceramics indicates that traditionally brittle ceramics can reach fracture toughness four or more times higher when it undergoes the Martensitic transformation. Hence, zirconium ceramics are a good candidate for applications where toughness is required and where advantages of wear resistance, low density and of high melting point characterizing ceramics can be taken advantage of. There are no mathematical models capable of predicting such a phenomenon at the present time. There is a reasonable expectation that stochastic based models averaged to the meso-scale where the Fokker-Planck equation provides densities for the stress, conductivity, etc. can be successfully applied to design and evaluation of such materials. This methodology is widely applicable to variety of other situation such as casting that has identical phenomenological nature. The mathematical modeling of these processes requires the above-mentioned theory of transition from stochastic to field models as well as understanding which equations describe the microscopic behavior of atomic lattice.

Milestone:

- Based on the stochastic variational principle, derived Fokker-Planck equations describing averaged properties of complex materials. Provide analysis and computational verification of the proposed model.

1.3.3. Methodologies for V&V, Sensitivity, and Uncertainty Quantification

Subproject lead: W. W. Symes

LANL Contacts: J. Kamm (CCS-2), Ken Hanson (CCS-2), Rudy Henninger (CCS-2)

Predictive computational models used for stockpile stewardship studies require sophisticated models simulated on the world's largest computers. These models are complex, hence advanced verification ("solving the equations right") and validation ("solving the right equations") methodologies are needed to assess their accuracy and predictive capability. In every major ASC simulation code, complex subsystems interact in complex ways to form cohesive computer programs that predict important physical processes. Sophisticated, component-based software enables analysts to unit test and verify the codes even if there are major improvements and changes to the subsystems of the code. Finally, even with verified numerical algorithms (in the physics and software sense) and validated physical models, the uncertainty of the model/algorithm and its sensitivity to change must be better understood.

1.3.3.1. *Parallel Tools for the Analysis and Optimization of Linked Subsystems*

Investigators: M. Heinkenschloss

LANL contacts: J. Kamm (CCS-2), Rudy Henninger (CCS-2), Ken Hanson (CCS-2), Dave Sharp (T-13), Rob Lowrie (CCS-2)

We are working on surrogate management frameworks and simultaneous analysis and design approaches for design, parameter identification, or uncertainty analyses governed by complex simulations.

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.

Short-term goals. Develop new ways to generate search directions in direct search methods (DSM) and expand the functionality of the NOMAD meta-algorithm class, which is available through the LACSI software web page. Extend our existing DSM and parallel simultaneous analysis and design (SAND) methods to handle larger classes of constraints. Investigate application of parallel SAND methods to optimization problems governed by advection-dominated partial differential equations (PDEs). Explore additional applications of current algorithms.

Long-term goals. Develop parallel tools for expensive engineering optimization problems, which may involve continuous, and a moderate number 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. Such optimization problems arise in the form of, e.g., design, control or parameter identification problems, but also as subtasks in verification and validation approaches.

Tasks:

- Identify additional example problems with interest to LANL.
- Explore the use of parallel domain decomposition (DD) based preconditioners for large-scale optimization problems with inequality constraints.
- Apply optimization based DD preconditioner to advection dominated PDE constrained problems.

Milestones:

- Coupling of optimization based DD preconditioner with interior point methods.
- Extension and application of optimization based DD preconditioner advection dominated test problem with relevance to LACSI.

1.3.3.2. *Simulation driven optimization*

Investigator: W. W. Symes

LANL contacts: Ken Hanson (CCS-2), Rob Lowrie (CCS-2)

Simulation driven optimization poses a notorious problem: discretization and linearization do not commute when adaptive gridding is part of the simulation package. This leads to the "optimize then discretize" vs. "discretize then optimize" debate. The former approach, in which the various components of the problem are discretized independently, seems to have gained the upper hand recently. Our contribution so far (with LACSI-supported PhD student Eric Dussaud) is to identify a set of simple model problems that show this is the only mathematically consistent way to combine adaptivity and Newton-type optimization. The key issue then becomes control of simulation error during optimization. The interesting complication is that one seldom if ever actually has a constructive estimate of simulation error.

This problem is already acute for classic ODE-based control problems. The fundamental issue – control of error using only asymptotic information – has been treated in various ways by Carter, Toint, Polak and Pironneau, and others, with no completely satisfactory result. In collaboration with M. Heinkenschloss, we have begun to sketch out what we believe will be an optimal approach.

Short-term goals: To describe a scheme for regulating accuracy of function and gradient values during quasi-Newton iteration, not using explicit estimates, but closer to optimal than Polak's Master Algorithm.

Long-term goals: To verify theoretically optimality properties of scheme, extend to constrained problem formulations; to verify adequate performance in control of 3D Navier-Stokes.

FY04 Tasks:

- Complete algorithm description, implement as abstract algorithm class;
- Apply to ODE control problems with adaptive time stepping, explore performance issues;
- Identify design/control problems of interest to LANL involving adaptive simulation.

1.3.3.3. *Large-Scale Nonlinear Optimization Algorithms and Software*

Investigators: Yin Zhang

Computer simulations often involve system parameters whose values are not completely certain. The current focus of our project is to develop formulations, algorithms and prototype software for effectively handling system uncertainties in large-scale nonlinear optimization problems, such as those arising from optimal design and optimal control where parameter uncertainties may affect the critical behavior of the simulations.

We have proposed and are studying a class of robust formulations that take into account the worst-case scenarios under a set of parameter uncertainty models. These formulations introduce adjustable ``safety margins'' proportional to the system sensitivities with respect to uncertain parameters. They are sufficiently general and, more importantly, are not significantly more difficult to solve than the original formulations, hence potentially applicable to large-scale problems.

We are investigating the proposed robust formulations to determine their theoretical and computational properties and to identify efficient methods for solving the resulting robust optimization problems. A central issue in algorithmic research is the efficient evaluation of system sensitivities with respect to uncertain parameters. Another research issue arises from the fact that the robust formulations introduce safety-margin functions that are not everywhere differentiable, necessitating the study of their differentiable approximations. We are also studying software frameworks for incorporating robustification into optimization packages without altering the existing simulation codes.

By computing a robust solution and comparing it to an original one, the technologies developed in this project may serve as tools to verify or validate the stability of optimal solutions with respect to system parameter uncertainties.

Milestone:

- Produce a technical report to document a proposed algorithmic framework for performing large-scale nonlinear optimization under data uncertainty.

Task:

- Report on an investigation of the properties of our proposed algorithmic framework for nonlinear optimization under uncertainty.

1.3.3.4. *Component frameworks for simulation driven optimization*

Investigator: W. W. Symes

LANL contacts: Tom Evans (CCS-4), Rob Lowrie (CCS-2), John Turner (CCS-2)

This is a special case of the general component framework design problem, and does not have nearly the complexity implications of the applications contemplated by CCA, for example. However it has considerable scope and we believe that our solution will be useful to many other groups. The framework must couple a control process, typically an optimization algorithm, with a simulation process, typically a parallelized PDE solver, running in two or more different software environments. The construction of such a framework is a good deal easier if the interfaces defined by the control and simulation processes are limited in type. The Standard Vector Library ("SVL"), a C++ package for simulation-driven optimization developed at Rice, provides two class families, DataContainers and FunctionObjects, which completely encompass all interaction with the communications layer in the component framework. With several graduate students, we are working on making as transparent as possible the interaction of these two types with a simple communications layer built out of standard TCP/IP sockets. This socket layer is merely an easily accessible example of the sort of communications layer on which a framework can be built – we have done the same thing with CORBA in the past. Our aim is to identify features of OO numerics library design that render the transition between serial and client-server applications as simple as possible, independent of framework implementation.

A related project couples our package, SVL, with the Trilinos Solver Framework developed at Sandia and various other OO toolkits, which *a priori* are type-incompatible. We believe that this project will pioneer the identification of design features for OO numerics packages that (1) permit relatively easy interaction with other packages, and (2) componentize easily.

Short-term goals. Demonstrate library coupling using socket-based framework and simulation-based optimization problems. Complete a design document.

Long-term goals. Identify features of OO numerics library design that will effectively facilitate transition between serial and component-oriented applications independent of framework implementation.

FY04 Tasks:

- Complete the component version of SVL using Streamable classes.
- Demonstrate client-server application of componentized SVL by solving a seismic inverse problem with distributed, multi-platform simulation.
- Explore integration of this project with other LACSI components efforts.

1.3.3.5. *Code-Based Sensitivity Analysis*

Investigator: M. Fagan

LANL contacts: R. Henninger (CCS-2), Ken Hanson (CCS-2), Jim Sicilian (CCS-2), John Turner (CCS-2)

The main focus of the Code-Based sensitivity project is area 3: "Methodologies for V&V, Sensitivity, and Uncertainty Quantification".

To see how code-based sensitivity addresses this focus, note that code-based sensitivity *is* a methodology for sensitivity calculation. Code-based sensitivity research seeks efficient methods for computing sensitivities of physical models expressed as computer codes.

In addition to sensitivity calculation, code-based sensitivity methods can be useful in V&V as well. Current research is investigating ways of using code-based sensitivity to verify differential equation solvers.

Short-term goals. Short-term goals for the code-based sensitivity project are concerned with furthering the application and development of the code-based sensitivity tool Adifor. In addition, LANL has identified the Telluride project codes as an important short-term focus. Therefore, as outlined in the Priorities and Strategies document, there are 2 major short-term goals:

- Assist in the application of Adifor to current Telluride Fortran 77 codes.
- Extend the Adifor technology to Fortran 90, so that accurate sensitivity calculations can be easily generated for the Truchas code.

Tasks:

To address the short-term goals, the important tasks to perform are:

- Maintain compatibility with current LANL systems.
- Continue to generate unit tests for various Fortran 90 and Fortran 77 components.

Milestones:

The FY04 milestone resulting from these tasks will be an alpha-test version of the Adifor90 tool that meets the following criteria:

- The tool will work on current Linux-based LANL hardware.
- The tool will pass 90% of Fortran 77 unit test suite.
- The tool will pass at least 40% of the Fortran 90 unit test suite not covered in the Fortran 77 unit test.

1.4. Application and System Performance

Investigators: John Mellor-Crummey, Barbara Chapman, Keith Cooper, Jack Dongarra, Robert Fowler, Guohua Jin, Celso Mendes, Dan Reed, Linda Torczon

Building scientific applications that can effectively exploit extreme-scale parallel systems has proven to be incredibly difficult. The sheer level of parallelism in such systems poses a formidable challenge to achieving scalable performance. In addition, the architectural complexity of extreme-scale systems makes it hard to write programs that can fully exploit the capabilities of these systems. In today's extreme-scale systems, complex processors, deep memory hierarchies and heterogeneous interconnects require careful scheduling of an application's operations, data accesses and communication to enable the application to achieve a significant fraction of a system's potential performance. Furthermore, the large number of components in extreme-scale parallel systems makes component failure inevitable; therefore, long-running applications must be resilient to hardware faults or risk being unable to run to completion.

The principal goals of the application performance research thrust are

- understanding application and system performance on present-day extreme-scale architectures through the development and application of technologies for measurement and modeling of program and system behavior,
- devising software strategies to ameliorate application performance bottlenecks on today's architectures,
- modeling the behavior of applications to understand factors affecting their scalability on future generations of extreme-scale systems, and
- investigating software technology that will enable higher performance on next-generation, extreme-scale parallel systems.

A broad spectrum of issues affects application performance. A multitude of challenging problems must be solved to understand how to best implement scientific applications so that they can achieve scalable high performance on extreme-scale parallel systems. As part of this research thrust, the project team will explore the topic of application performance on many fronts and undertake a program of research that aims to develop technologies to support measuring, modeling, understanding, tuning, and steering application performance on current and future generations of extreme-scale parallel architectures. This work will address all aspects of performance and reliability spanning system architecture, network, and applications. Our investigation will include work on both scalability and node performance.

The findings from this research, as well as tools and software infrastructure developed as products of this effort, are expected to benefit all ASC application teams by providing them with more efficient programming models, technology for compiler-assisted tuning of applications, better performance instrumentation and diagnostic capabilities, improved algorithm-architecture mapping, and better performing extreme-scale parallel architectures.

1.4.1. Tools for Performance Measurement and Analysis

Investigators: Robert Fowler, John Mellor-Crummey, Celso Mendes, Dan Reed, Jack Dongarra

A key goal of performance measurement and analysis tools is to help to understand the behavior of applications running on extreme-scale parallel systems by quantifying inefficiencies that arise from a variety of factors such as load imbalance, serialization, underutilization of processor functional units, data copying, poor temporal and spatial locality of data accesses, exposed communication latency, high communication frequency, and large communication bandwidth requirements. A quantitative assessment of factors limiting application performance on current-generation architectures will help to focus long-term research on software and hardware technologies that hold the most promise for improving application performance and scalability on future systems.

On terascale systems, performance problems are varied and complex and thus a wide range of performance evaluation methods need to be supported. Large-scale applications often have irregular behavior, with time varying resource demands. The appropriate data collection strategy depends on the aspect of program performance under study. Key strategies for gathering performance data include statistical sampling of program events, inserting instrumentation into the program via source code transformations, link time rewriting of object code or binary modification before or during execution. Capturing traces of program events such as message communication helps to characterize the temporal dynamics of application performance; however, the scale of these systems implies that a large volume of performance data must be collected and digested. Improved data collection strategies are needed for collecting more useful information and reducing the volume of information that must be collected. Statistical sampling provides a formal basis to achieve a desired estimation accuracy under a certain measurement cost. We will investigate the feasibility of using statistical sampling techniques to characterize performance on large systems. Research problems to be addressed include determining the appropriate level for implementing different instrumentation and measurement strategies, how to support a modular and extensible framework for performance evaluation as well as the appropriate compromise between instrumentation cost, the level of detail of measurements and the volume of data to be gathered.

Current tools for analysis of application performance on extreme-scale systems suffer from numerous shortcomings. Typically, they provide a myopic view of performance; they provide only descriptive rather than prescriptive information; and they fail to support effective analysis and presentation of data for extreme-scale systems. To help users cope with the overwhelming volume of information about application behavior on extreme-scale systems, more sophisticated analysis strategies are needed for automatically identifying and isolating key phenomena of interest, distilling and presenting application performance data in ways that provide insight into performance bottlenecks, and providing application developers with guidance about where and how their programs can be improved. Comparing profiles based on different events, computing derived metrics such as event ratios and correlating profile data with routines, loops and statements in application code can provide application developers with insight into performance problems. However, better statistical techniques are needed for analyzing performance data and for understanding the causes and effects of differences among process performance. Instead of modeling each system component, these techniques select a statistically valid subset of the components, and model the members of that subset in detail. Properties of the subset are used as a basis in estimates for the entire system. Our research in this area, so far, has focused on system availability. We plan to expand that scope and apply these techniques to study application performance. The main goal is to evaluate how well application performance can be characterized and understood, based on a more efficient data collection scheme.

Tasks:

- Refine HPCToolkit binary analysis tools to improve efficiency on large applications and improve source-level mapping of program structure for optimized programs; robustify hpcviewer user interface. (Quarter 1)
- Develop HPCToolkit prototype for Opteron systems. (Quarter 2)
- Construct a tool for collecting call-graph style profiles for unmodified, optimized application binaries on Alpha+Tru64 platform. (Quarter 3)
- Explore strategies for analyzing and presenting call-graph profiles of large application programs. (Quarter 4)
- Refine the SvPablo toolkit to better support instrumentation and collection of performance data for applications of interest to LACSI (Ongoing)
- Interact with LANL application researchers to characterize the behavior of their applications executed on large-scale systems, and to explore opportunities for performance improvements based on the findings produced by this characterization (Ongoing).

- Continue refinement of the PAPI interface for accessing hardware performance counters. The goal of this effort is to provide a robust implementation of PAPI including features such as thread safety, counter multiplexing, and counter-driven user callbacks on important computing platforms. (Ongoing)
- The academic performance analysis team will hold a third performance tools workshop at LANL if the applications teams or LANL management believe another such workshop would be productive.

1.4.2. Modeling of Application and System Performance

Investigators: John Mellor-Crummey, Robert Fowler

The modeling of high performance software and hardware systems is highly complex requiring the encapsulation of key processing structures and characteristics. This is a direct result of the performance space being multi-dimensional and highly non-linear in any of its dimensions. As a result, accurate models such as the ones developed by the Performance and Architecture Laboratory (PAL) at Los Alamos are the performance tool of choice in gaining insight into the performance of applications and systems.

As a consequence of the complex and interacting factors affecting program and system performance research in performance modeling must necessarily span a wide range of topics. At Los Alamos, PAL's focus is on building holistic models of parallel system and application performance. However, at present, construction of these models is a manual, painstaking process. An important challenge for modeling research is to simplify the task of model creation. A key task of the academic partners will be to research strategies to simplify model construction. The focus of this aspect of the research will be on designing, building and evaluating semi-automatic tools for synthesizing models and model components as well as to explore how to integrate model components synthesized automatically into hand-crafted model frameworks. A second area of research will explore the applications of performance models to a variety of problem domains including scalability analysis of applications and architectures, as well as evaluating the potential of future architectures.

Tasks:

- Complete basic IA64-based processor architecture modeling infrastructure for predicting performance of loop nests. (Quarter 1)
- Refine instruction schedule modeler to better support modeling of VLIW architectures and to incorporate memory hierarchy performance model results. (Quarter 2)
- Evaluate and extend modeling approaches to more accurately model node performance. (Quarter 3)
- Explore combining single-processor/node modeling efforts of Rice with scalability models for entire applications developed at LANL to yield composite models reflecting both serial and parallel performance. (Quarter 4)

1.4.3. Automatic application tuning

Investigators: Keith Cooper, Jack Dongarra, Robert Fowler, Ken Kennedy, Guohua Jin, John Mellor-Crummey, Linda Torczon

The architectural complexity of modern computer systems makes it very difficult to manually tune codes so that they achieve top performance. Furthermore, the rapidly changing landscape of computer platforms means that any investment in manual tuning for a particular platform soon becomes obsolete. A promising approach that has emerged for automatically tuning library codes is embodied by the ATLAS and UHFFT projects. At library generation time, these systems generate a multitude of variants of a pre-determined set of library primitives, run a collection of experiments to empirically evaluate the performance of each variant on the target platform, and then select the variants with the best performance for inclusion in the run-time library. This search-based tuning strategy virtually eliminates the manual effort of tuning for particular target platforms and generates highly efficient code. The tuning strategy can be re-executed when a version is needed for a new platform. Research is needed to devise search algorithms that make it practical to apply this style of automatic empirically driven tuning to whole applications.

Previous work on self-tuning libraries required library developers to write a code generator to enumerate each interesting variant of a library procedure for empirical evaluation. Generalizing this automatic tuning approach to whole programs requires developing algorithms for identifying critical loop nests that could potentially benefit from tuning. A conversion tool could use deep compiler analysis to restructure critical loop nests into a form to which automatic tuning can be effectively applied by making extensive use of symbolic strip mining. For tuning parallel loop nests, a strategy called *overpartitioning* that can be used to facilitate mapping of chunks of computation to different processors or machines. Exhaustive search through all combinations of variants for each loop nest in a program is impractical. The challenge is to develop intelligent search algorithms that use program semantic information to guide the selection of promising implementation variants that merit empirical evaluation. Moreover, these techniques must include alternatives and solutions that can respond to hardware and software faults present in extreme scale systems.

A promising direction is to use data from hardware performance counters to guide empirically driven tuning approaches. Hardware performance counters can help identify and quantify performance bottlenecks. We expect this knowledge will be useful for helping to identify promising candidate transformations for improving performance. Also, hardware performance counters can be used to measure the benefits and costs of transformations, which can help guide their use.

Tasks:

- Extend experiments with adaptive choice of blocking factors from DMXPY to other dense linear-algebra codes. (Quarter 2). Investigate adaptive choice for other transformations such as inline substitution. (Quarters 3 & 4)
- Continue refinement of and experimentation with tools for performing complex sets of interacting program transformations (directed by manual annotations) to improve program performance. (Quarters 1& 2)
- Explore strategies for model-guided, empirically based application of single program transformations to improve performance of scientific programs. (Quarters 1& 2)

- Explore decision algorithms for applying transformations effectively in concert (Quarters 3 & 4).

1.4.4. Compiler technology for exploiting modern processors

Investigators: Keith Cooper, Ken Kennedy, John Mellor-Crummey, Linda Torczon

Keeping pace with the Moore's law curve and delivering 60% annual increases in processor performance has come at the expense of increasing complexity of processor architectures. Exploiting modern processor architectures to achieve a significant fraction of peak performance with code compiled from conventional programming languages such as Fortran, C, and C++ requires compiler innovation for each new architectural feature. For example, the IA-64 introduces a set of new architectural features that pose a considerable set of challenges for compilers. First, the functional units must be kept busy. This requires the compiler to transform the input program so that it has enough instruction-level parallelism (ILP) to sustain the computation rate. It requires instruction-scheduling techniques that can convert available ILP into dense schedules - for simple loops, for loops with control flow, and for straight-line code. Second, operands must be ready for each instruction. This will involve transforming programs to match their locality to the memory hierarchy of the target system, including real applications of blocking, prefetching, and (perhaps) streaming. Once data is on-chip, a compiler may need to manage instruction and data placement with respect to the clustered register file, along with the classic problems of allocation and scheduling. Third, predication must be handled with a holistic approach. If-conversion is not the whole answer. Open issues include understanding the tradeoff between underutilization of instruction issue slots with predication versus branching to out-of-band denser (unpredicated) code, predicate register management, the interaction between predicate lifetimes and instruction placement in the scheduler, and minimizing the impact of predicate evaluation on overall application performance.

Tasks:

- Examine assembly code produced by the ORC compiler for IA-64 to determine where performance is being lost. (Quarters 1 & 2). Develop a strategy for improving the performance of code produced by the ORC compiler (Quarters 3 & 4).
- Investigate opportunity for improving scientific code on x86-style machines using low overhead dynamic optimization techniques. (Quarters 3 & 4)

1.4.5. Compiler-based tools and run-time technology for parallel programming

Investigator: Barbara Chapman

The objective of this project is enhance programmer productivity with compiler-based tools and to develop compiler and run-time technology that will help application developers achieve better application performance with OpenMP, alone and in conjunction with MPI.

- *Productivity Enhancing Tools:* 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

OpenMP: 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.

Task:

- Deliver a working open source OpenMP compiler for the Itanium architecture. That would include front ends for Fortran, C and C++, the lowering and generation of object code. However, few if any optimizations will be included, the emphasis being a working (i.e. robust) compiler.

1.4.6. Dynamic adaptation and steering

Investigators: Dan Reed, Celso Mendes, John Mellor-Crummey

Today, most applications simply run to completion on the resources that they acquire at program launch. However, large-scale systems built from commodity components are prone to failure and long-running applications for such systems must sense and respond to component failure.

Performance steering offers an opportunity to adjust a running program for more efficient execution and to adapt to changing resource availability (e.g., due to component failures or resource sharing). A challenge is to develop strategies that enable applications to monitor their own behavior and reactively adjust their behavior to optimize performance according to one or more metrics. More generally, better techniques for dynamic adaptation and control for complex, ASC-scale systems are needed. Our goal is to develop tools and approaches that can help applications achieve high performance even when system components fail or applications are subject to other system constraints. Strategies for automatic performance steering based on performance and fault models offer the potential to enable long-running programs to repeatedly adjust themselves to changes in the execution environment – perhaps to opportunistically acquire more resources as they become available, to rebalance load, adapt to component failures, or bound power consumption. Validated performance "contracts" among applications, systems, and users that combine temporal and behavioral reasoning from performance predictions, previous executions, and compile-time analyses are one promising approach. This work will explore using performance contracts to guide the monitoring of application and resource behavior; contracts will include dynamic performance signatures and techniques for locally (per process) and globally (per application and per system) evaluating observed behavior relative to that expected.

Tasks:

- Explore techniques for "performance contracts" (i.e., application and system level techniques for closed loop performance monitoring and adaptation to approximate fixed performance levels).
- Demonstrate adaptation techniques for multi-attribute system behavior.

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 FY04 project year.

In addition to hosting visitors and speakers, the LACSI academic partners in conjunction with LANL organized, hosted, and documented three technical workshops held at LANL during FY03 on topics related to the LACSI technical vision. Specifically, performance workshops were held in February 2003 and July 2003 and a components workshop was held in April 2003. During the FY04 project year, the LACSI academic partners in conjunction with LANL will organize, host, and otherwise support technical workshops on topics related to the LACSI technical vision. We are actively seeking additional funds to support larger meetings on topics relevant to LACSI.

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 FY04 LACSI Symposium was held October 27-29, 2003 in Santa Fe, New Mexico. In addition, the LACSI academic partners in conjunction with LANL disseminated information about LACSI and its research results at Supercomputing 2003. Plans are underway to hold the FY05 LACSI Symposium in Santa Fe, New Mexico, October 12-14, 2004. Details related to the LACSI symposium are available at http://lacsi.lanl.gov/symposium.

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. Rod Oldhoeft (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 personnel, 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, Chair, *Los Alamos National Laboratory*
- Ken Kennedy, co-Chair, *Rice University*
- Jeff Brown, *Los Alamos National Laboratory*
- Jack Dongarra, *University of Tennessee at Knoxville*
- Bill Feiereisen, *Los Alamos National Laboratory*
- Rob Fowler, *Rice University*
- Adolfy Hoisie, *Los Alamos National Laboratory*
- 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 North Carolina at Chapel Hill*
- 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 of the research performed and its relevance to LACSI goals. Based on the outcomes of its reviews of LACSI research and other LACSI activities, the EC might identify projects to phase out and propose a collection of projects to be undertaken, along with goals for those projects. 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 document (LAUR # 02-6613) titled *Priorities and Strategies*.

In April 2003, the EC met with senior LANL personnel to revise the framework, priorities, and strategies established at the planning meeting in 2002 and *Priorities and Strategies* was revised to incorporate the results of the April 2003 planning meeting (LAUR # 03-7355). In February 2004, the EC again met with senior LANL personnel to revise the framework, priorities, and strategies established in previous planning meetings. *Priorities and Strategies* is being revised to reflect the results of the February 2004 planning meeting. Relevance to the LACSI priorities and strategies outlined in the document continues to 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 fall and at the spring planning meeting. In FY03, LACSI EC meetings were held on October 15, 2002 (Santa Fe, NM) and April 8, 2003 (Houston, TX). In FY04, the LACSI EC meetings were held on October 28, 2003 (Santa Fe, NM) and February 19, 2004 (Houston, TX).

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), Deepak Kapur (University of New Mexico), Dan Reed (University of North Carolina at Chapel Hill), 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 (Computational Science), John Mellor-Crummey (Application and System Performance), and Linda Torczon (Computer Science 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 monthly 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 ASC 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.