Priorities and Strategies

Los Alamos Computer Science Institute

On March 18-19, 2002 the Los Alamos Computer Science Institute (LACSI) Executive Committee and Principal Investigators met to discuss methods of addressing issues raised in the 2001 LACSI Contract Review. The body was tasked to develop priorities and strategies to meet future programmatic and LANL computer science needs.

A framework was developed 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. On April 8-9, 2003, the LACSI Executive Committee and Principal Investigators met with senior LANL personnel to revise the framework, priorities, and strategies established at the planning meeting in 2002.

The revised framework outlined five strategic thrust areas:

- Components
- Systems
- Computational Science
- Application and System Performance
- Computer Science Community Interaction

This document presents the research vision and implementation strategy in each of these areas.

Components

Ken Kennedy (ken@rice.edu)
Jack Dongarra (dongarra@cs.utk.edu)
Lennart Johnsson (johnsson@cs.uh.edu)
Doug Kothe (dbk@lanl.gov)
Craig Rasmussen (crasmussen@lanl.gov)

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 a 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.

Short-term Goals

Components & Component Frameworks Review

Subproject Leads: Craig Rasmussen, LANL, crasmussen@lanl.gov; Ken Kennedy, Rice, ken@cs.rice.edu

The focus of this subproject is to conduct 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 ASCI 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).

LACSI Component Integration Challenge Problem

Subproject Leads: Craig Rasmussen, LANL, crasmussen@lanl.gov; Ken Kennedy, Rice, 713-348-5186, ken@cs.rice.edu

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 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).

Long-term R&D

Once the problems and current solution strategies are well understood, the Components research effort should focus on long-term research and development projects that will not only address the problem effectively when they mature many years in the future, but also provide important short- and medium-term payouts for ASCI and Los Alamos applications.

A major goal of this research should be to address the trade-off between generality of programming systems and the performance that applications written in them can deliver. Today, many high-level problem-solving systems exist, but the performance penalty for using them is severe. Is this situation an immutable law of nature, or merely an artifact of the implementation approaches we have pursued to date? The proposed research efforts on telescoping languages and high-performance Java for prototyping attempt to address this issue.

A second major question in this area is: Can we build components with the built-in ability to adapt with high performance to new computational platforms? One approach that has proved successful is the Atlas system, which uses substantive amounts of computation to provide versions of a computational linear algebra kernel that are highly tuned in advance to different machines. If this approach can be extended more generally to components of all kinds, it would help avoid the enormous costs involved in retargeting applications to different machines.

In the sections that follow, we will elaborate on some promising research directions addressing these issues.

Generation of Problem-Solving Systems Through Component Integration

Subproject Leads: Ken Kennedy, Rice, ken@cs.rice.edu; Jack Dongarra, Tennessee dongarra@cs.utk.edu

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 five 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 ASCI and LANL.

An important goal of this effort is to make it possible to build highly efficient applications from script-based integration of pre-defined components. Building on the component architecture efforts described 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).
- Deliver prototype performance modeler for heterogeneous components (Quarter 4)
- Produce a design strategy for the LACSI challenge problem (Quarter 4)
- Deliver the JaMake framework for use at Los Alamos. (Quarter 4)

Retargetable High-Performance Components and Libraries

Subproject Leads: Jack Dongarra, Tennessee dongarra@cs.utk.edu, Lennart Johnsson, Houston, johnsson@cs.uh.edu, Ken Kennedy, Rice, ken@cs.rice.edu,

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 ASCI program.

Tasks:

- Investigate optimization techniques for basic linear algebra for Intel Itanium and AMD Opteron processors. (Quarter 1)
- 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 2).
- 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. Make adjustments as necessary to the UHFFT API and internal optimization/adaptation procedures (Quarter 4).
- Explore performance evaluation and tuning of the UHFFT and an emerging Multi-Grid library on LANL platforms, Itanium systems with Myrinet and Opteron systems with Infiniband (Quarter 4).

Systems

Rod Oldehoeft (rro@lanl.gov)
Rob Fowler (rjf@rice.edu)
Edward Angel (angel@cs.unm.edu)
Thomas Caudell (tpc@eece.unm.edu)
Jack Dongarra (dongarra@cs.utk.edu)
Wu-Chun Feng (feng@lanl.gov)
Rich Graham (rlgraham@lanl.gov)
Lennart Johnsson (johnsson@cs.uh.edu)
Barney Maccabe (maccabe@cs.unm.edu)
John Mellor-Crummey (johnmc@rice.edu)
Dan Reed (reed@cs.uiuc.edu)
Scott Rixner (rixner@cs.rice.edu)

The computer systems research being addressed by LACSI is organized into six research thrust areas: Reliability, Adaptability, Commodity, Compiler/System Interfaces, Advanced Architectures, and Scalable Visualization.

Reliability

High-end systems of the future will be larger and more complex than those we use today, which are even now dangerously close to the limit of our ability to operate reliably. Extensive R&D is needed to tame this complexity so that the power of future systems will actually be available for productive work.

Short Term Goals

API to Reliability Capabilities

[LANL, UNM]
Today's hardware is being designed to monitor its own condition. For example, temperature and fan speed sensors are common on system boards, correctable memory and disk errors can be logged, and SmartDisks monitor their own operating conditions. We need expose this information to higher levels of the operating system, to middleware, and to applications to give them to react to impending problems. Research on, and development of, APIs for "reliability reflection" facilities is necessary to make the information truly useful.

Checkpointing

[LANL CCS-1]
We will pursue a more automated (compiler support) approach to application checkpointing to reduce the volume of saved data. We will research asynchronous forms

of checkpoints across an application for to reduce potential bottlenecks in networks and filesystems; this leads to transactional approaches.

WAN Reliability for Large Data Sets

[LANL]
Currently, transferring large data sets over wide-area networks is an all-or-nothing communication: if the transfer is interrupted, it must be restarted. We will pursue ways of providing persistence for partially transmitted data, so a transfer can resume, and perform research in using striping and parallel transfers for both throughput and reliability.

Long Term R&D

Compiler and Tool Support for Fault-Tolerant Programming

[Rice, LANL]
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.

This is a long-term effort; In FY04 we will focus on identifying long-term research directions.

Scalable Fault Tolerance

[UIUC, LANL]
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 dies, all MPI processes are killed, and the user must re-submit the job, which could incur a long wait in queue for large jobs). In addition, disk-based checkpointing requires high bandwidth I/O systems to record checkpoints.

To solve these problems, we propose an over provisioning scheme that incorporates in-memory, diskless checkpointing [1] with a fault-tolerant MPI implementation and real-time monitoring of system failure indicators (i.e., 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. This approach is described below.

Low-Latency Checkpointing: Our approach to diskless checkpointing is a software-based RAID5 technique applied to memory. Nodes are partitioned into equal-sized groups, and in each group one or two nodes are spares. The checkpoint data is written to memory, and parity is calculated and stored on the spares.

Diskless checkpointing is complementary to traditional disk checkpointing. 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 (e.g., via LANL's Supermon). We plan to implement this approach as a dynamically loaded user module that intercepts UNIX I/O calls and diverts them optionally to memory or disks.

Failure Models: We will continue analyzing the behavior of ASCI-scale systems in terms of their availability and their sensitivity to component failures. Using this data, we are developing 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. A preliminary time-to-solution performance analysis of the scheme described above shows that a system with 10,000 nodes can survive three times longer if each group has two spares.

Fault Tolerant MPI: Messaging systems will be developed for future interconnects to prevent errors (dropped or corrupted messages, loss of a link or a NIC) from terminating an application. Techniques, both manually programmed and automated by compilers, for applications to continue to function when a component computation fails will be researched.

We will extend an existing MPI implementation with APIs that let users register callback functions for application-specific recovery (e.g., loading the checkpoint data and re-initializing data structures and other resources). Another API will allow the user to register application-specific checkpoint routines. This approach is complementary to alternatives like LA-MPI by focusing on node status and failures, rather than communication links.

We will add health and heartbeat monitoring and failure recovery. When the MPI implementation recovers from a failure, it invokes the application-specific recovery routine. To exploit diskless checkpoints, we can invoke application-specific checkpoint routines and direct MPI synchronization calls such as MPI_Reduce. Checkpointing at a

higher frequency, which is possible due to the diskless feature, should greatly decrease the work to be re-done during recovery. We can also determine the checkpoint frequency intelligently by observing node health information (such as temperature, disk retries, soft memory errors) and past failure/recovery history.

Finally, we expect both of these schemes to include intelligent learning and adaptation. By monitoring and analyzing failure modes, the system can estimate the number of spare nodes needed for in-memory checkpointing to achieve a specified reliability. This will enable smoothly balancing performance and reliability.

We will develop systems that can monitor the status of various components and take action to keep applications running as failing parts are mapped out and replaced. This will require further development of technologies in monitoring, rapid migration of processes in any state, component virtualization and hot-swapability, and rapid resumption.

Adaptability

Systems and system software of the future will have to adapt themselves to complex environment. In addition to adaptation in the face of failure (preceding section), systems and their software will have to adapt to dynamic reconfiguration, to changes in their external environments, and to varying demands of high performance applications.

Long Term R&D

Network Protocols

[UNM]
Modern compute node architectures provide several contexts for execution, ranging from different address spaces on the host processor (operating system space versus application space), different execution contexts in the same address space (threads) to specialized processors on the node (e.g., network interface cards). Compute node operating systems must first be cognizant of the different places in which execution can occur and, second, be prepared to move functionality to different execution contexts to meet the needs of an application. In high-performance systems, the need to exploit different execution contexts is critical in the context of communication.

Matching of pre-posted receives in MPI might be done in an application-level thread, in the OS during interrupt handling, or by a specialized processor in a NIC. Execution handlers for active messages may impose more than one requirement on deadline processing, and so can be handled in different execution contexts.

Ultimately, the decision of where functionality should be placed is based on a large set of performance tradeoffs. In large part, these tradeoffs will depend on characteristics on the application. Initially, applications might explicitly declare where functionality should reside, more likely this would be hidden in a library. In the longer term, we need to investigate ways in which this can be automated through adaptation.

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.

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.

Commodity

While once the demands of high performance scientific computation was the dominant driver for new architectures, the much larger markets for commercial and home systems drive industry today. There will probably always be a demand for custom designed computation systems for certain very demanding, high value applications. Economics will require that most of the nation's capacity for high performance computing will need to be met with systems constructed almost entirely out of commodity components.

Long Term R&D

Operating Systems Kernels

[UNM]
The use of commodity operating system kernels is an attractive option. UNM will explore the suitability of Linux for scalable systems.

Son of Science Appliance Research

[LANL]
Currently, the Science Appliance software operates with a single master node; this has shown to be scalable to large numbers of worker nodes, but it presents a single failure point. Research will be pursued to build SA clusters with multiple master nodes that fail over seamlessly. Scheduling strategies will be adapted so multiple masters coordinate in the scheduling of worker nodes.

Protocols and Communications

[UNM]
The goal of this research is to investigate high-performance implementations of traditional and specialized communication protocols on commodity network fabrics. We define commodity networking in terms of the protocol layer implemented in the network fabric. In particular, we focus on IP networks: networks that route traffic using the Internet Protocol.

Traditionally, communication performance has been defined in terms of bandwidth and latency. While these metrics are still important, CPU overhead, the percentage of the host processor consumed in the implementation of a communication protocol, has recently emerged as another important metric. As network speeds have increased, so has CPU overhead. In many instances, the bandwidth that can be attained is limited by the speed of the processor.

In commodity systems, the interface point between the network fabric and computational node is a network interface card (NIC) on the PCI bus. Many vendors provide programmable NICs. In a preliminary study, we wrote a control program for the Alteon Acenic Gigabit Ethernet NIC. The default control program simply moves IP fragments between the host processor and the network. The altered control program implements IP fragmentation and reassembly on the NIC. We measured UDP delivery rates of 960 Mb/s for both control programs. However, while the CPU utilization for the default control program was about 80%, the CPU utilization for our control program was closer to 60%, a significant improvement. We plan to use the Myrinet interconnection network on the large LANL cluster for additional scaling studies.

We have two specific goals:

- Determine the performance benefits of moving the entire IP layer into a communication co-processor.
- Evaluate alternative implementations of the TCP sliding window and high-performance implementations of alternate communication protocols, including: VIA, Scheduled Transfer, and Portals 3.0.

Efficient Zero-copy, Zero-mapped Asynchronous I/O Subsystems

[Rice]
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. Furthermore, we are currently implementing a Gigabit Ethernet NIC using the Avnet Virtex II development board, which includes an FPGA and an SO-DIMM slot. This will enable us to show that the bottlenecks encountered by LA-MPI over TCP can be alleviated by intelligent and programmable NICs and improving the way the MPI library uses TCP. We expect the innovations we propose using our flexible NIC to migrate into mainstream network interfaces.

Milestones:

- Fall '03: TCP Implementation of LA-MPISummer '04: Paper on bottlenecks and proposed solutionsSummer '05: Prototype solution using LA-MPI, Linux, and our NIC

Compiler/System Interface

Long Term R&D

Compiler Support for Scalable Parallel Programming

[Rice]
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.

Asynchronous Interfaces (Disks, MPI)

[Rice]
A significant component of I/O and communication operations is processor overhead of marshalling and otherwise copying data. Synchronous zero-copy methods are not the answer. On the other hand, using asynchronous APIs to communication and I/O pose several difficulties:

- To be effective, asynchronous initiation and wait operations must be widely separated. Of particular importance, asynchronous receive operations must be posted sufficiently far in advance that they are guaranteed to have specified the final destination of any received data before that data arrives.
- Effective hand placement of asynchronous operations is notoriously difficult for non-specialists to perform. It requires analyzing data dependences to ensure that all data hazards have been respected.
- Compilers for high-level languages such as HPF, CAF, and UPC automatically place communication operations in their generated code. Not all programmers will use such languages models. It is important to support communication and I/O placement in lower-level languages.

In the next few years, a large fraction of HPC capacity computing will be provided through commodity clusters based on X86 family (P4, Xeon, AMD64) processors. The Intel family in particular has resource bottlenecks that limit the utility of on-node SPMD-style data parallelism; processors contend for a single interface to memory, and “Hyper-threads” contend for on processor resources (memory interface, FPUs, etc.). An alternative is to use an asymmetric task-parallelism model of computation. We will investigate this strategy specifically as a means of accelerating large scientific codes. In particular, we will investigate the use of “program slicing” for scientific applications in which one thread

performs numeric calculations while one or more other threads are used to control communication, I/O, and other tasks. This will require an extension to existing thread scheduling mechanisms.

Advanced Architectures

Long Term R&D

Advanced Network Interfaces

[UNM]
In today's systems, network interfaces are connected through I/O busses that have significantly less bandwidth than processors and memories. We are entering into an era in which network speeds are increasing rapidly. While evolutionary improvements to I/O busses is one approach, *e.g.* PCI/X, we believe that more revolutionary approaches should also be evaluated. One approach is to put network interfaces much closer to CPUs and memory, for instance, by plugging a NIC into a CPU socket in a SMP system. This approach is being explored in industry and we will seek opportunities to pursue collaborations in this area.

Systems for Scalable Visualization

Short Term Goals

Improvements to Visualization Software

[UNM]
Modify UNM visualization software to support compound/composite displays; define and codify the experimental processes; execute the experiment, perform preliminary analysis, and report findings.

Long Term R&D

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

[UNM]
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.

Computational Science

Doug Kothe (dbk@lanl.gov)
Beth Wingate (wingate@lanl.gov)
Bill Symes (symes@rice.edu)
Dan Sorensen (sorensen@rice.edu)
Mike Fagan (mfagan@rice.edu)
Lennart Johnsson (johnsson@cs.uh.edu)
Deepak Kapur (kapur@cs.unm.edu)

Overview

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. Continuum Dynamics, Energy Transport, and Materials Science;
2. Multi-Physics Coupling; 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 ASCI "weapons performance code projects", known as collectively as the Crestone, Shavano, and Marmot Projects. Success also depends in part on helping the CompSci PE to meet its "Level 2" (L2) ASC milestones over the next three years, which are:

- Deliver and demonstrate a hybrid Monte Carlo deterministic transport capability (Q1/FY04);
- Simulate casting of the Qual Type 126 pit and compare the results of the simulation to the available experimental data for the same process (Q1/FY05);
- Deliver and demonstrate a Capsaicin Project transport capability to the Marmot Project (Q2/FY05) (Capsaicin is a software project for a verified deterministic transport capability.); and
- Deliver and demonstrate an interface tracking interface component to the Crestone Project (Q4/2005).

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 weapons performance and manufacturing simulation tools.

Research Thrust Areas

Continuum Dynamics, Energy Transport, and Materials Science

Subproject Leads: Lennart Johnsson (UH), Y. Kuznetsov (UH), R. Glowinski (UH), J. Morel (CCS-4), J. Sicilian (CCS-2), and 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).

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

UH: Y. Kuznetsov, R. Glowinski, and L. Johnsson
LANL: J. Morel, M. Shashkov, M. Berndt, and D. Moulton

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:
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:
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.

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

Rice: P. Kloucek and P. Solin
LANL: J. Brackbill, Marius Stan, and 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 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.

Short Term:

- 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 J. 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:

- *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.

Multi-Physics Coupling

Subproject Leads: D. C. Sorensen (Rice), Dana Knoll (T-3), and
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.

Numerical Linear Algebra for Large Systems
(Eigenvalue Methods and Software for ASCI-MPP Systems)

Rice: D.C. Sorensen and M. Embree
LANL: B. Nadiga, Jim Morel (CCS-4), Rob Lowrie (CCS-2), Dana Knoll (T-3),
John Turner (CCS-2), and Beth Wingate (CCS-2)

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. We are also concerned with steady state linear stability analysis on an idealized reduced-gravity quasigeostrophic ocean model. Part of this success has been the detection of Rossby waves, which can be indicators for the onset of chaotic behavior in the dynamical system.

Short term:

- Stability and sensitivity analysis of certain steady states of the OCM.
- Determination of Rossby waves, which are fundamental to the understanding of very long term cycles in ocean circulation.

Considerable algorithmic work is needed to accomplish this. We expect to improve eigenanalysis performance by orders of magnitude with an approximate shift invert scheme, a technique for introducing pre-conditioning in eigenvalue calculations. This scheme has been implemented and tested on the OCM and has successfully computed the desired Rossby waves and corresponding eigenvalues much faster than any competing schemes. We observe linear scaling with respect to the problem size (i.e. mesh independence for the required number of iterations).

To accelerate convergence for these problems, Embree and Sorensen are studying adaptive pre-conditioners based upon partial eigensystem information and effective restarting and deflation methods.

Long term:
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 the 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.

We plan to utilize a component framework for software integration within this project as a prototype for future projects that might require steady state calculations, stability, and bifurcation analysis. We intend to use the Python scripting language to implement our component framework. However, prior to a final decision, we shall also consider other options. Many of the components that we need are already available. We expect that integration of these components with Python will result in a framework that can be easily modified to accommodate improved components and future advances in the OCM.

Methodologies for V&V, Sensitivity, and Uncertainty Quantification

Subproject Leads: W. Symes (Rice), J. Kamm (CCS-2), Ken Hanson (CCS-2), and 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 ASCI 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.

Analysis and Optimization of Linked Subsystems

Rice: J. Dennis and M. Heinkenschloss
LANL: J. Kamm (CCS-2), Rudy Henninger (CCS-2), Ken Hanson (CCS-2), Dave Sharp (T-13), and 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.

One aspect of our research targets problems that involve complex multi-physics codes that are originally designed for simulations only, allow little intrusion by the optimizer, and, in most cases, are only available as black boxes. Frequently, traditional optimization approaches cannot be applied to these problems. In discussions with LANL staff, these were identified as the types of optimization problems that they encounter most often. Our surrogate management framework has been quite successful in solving several such complex design problems. Design parameters can be continuous or categorical. A

categorical variable is a variable that must always be a member of a finite set, or else the simulations on which the design is to be based cannot be run. For example, a categorical variable could represent the choice from a set of materials for a given insulation sector. We have even solved problems where the categorical variable determines its dimension. Software is available through the LACSI web site and a version of our surrogate management framework is implemented in the Boeing Design Explorer, used routinely in industrial MDO problems at Boeing.

Users are clamoring for several enhancements, most notably, that we be able to apply our techniques to larger problems on the order of 100-200 variables rather than the 20 or so variables we are handling for them now. The major obstacle here is the difficulty in building such high dimensional surrogates. Another extremely important question is how to handle mixtures of continuous and categorical design variables in black-box optimization problems. Traditional branch and bound techniques cannot even be applied to these problems. We have a successful algorithm developed with LACSI support, but we need to study how to handle larger numbers of variables.

Another aspect of research is the development of simultaneous analysis and design (SAND) tools for optimization problems with a large number of variables. These tools complement our surrogate management frameworks. They have the potential to solve the optimization problem in a time that is only a small multiple of the time needed for a single simulation, and they can handle very large numbers of continuous variables. However, they are tightly integrated with the simulation. Such methods have been used, e.g., for optimal design and control application in fluid flow. We have approached the T7 group to explore possible collaborations. Despite the success of SAND methods, several important questions remain. These include the integration of parallelism into the optimization, extension of our SAND approaches to better handle inexact problem information, improvements to handling a very large number of inequality constraints, and use of model hierarchies to enhance the convergence properties. We will investigate the possibility and viability of our SAND approach being integrated into the new CCS ASC code project.

<u>Short term:</u>

- Evaluate domain decomposition approaches to integrate parallelism into SAND optimization as well as using it for subproblem solves. Can the Zoltane SNL tool be used here? Addendum, June 03: The Zoltane SNL tool could be useful, but we have not yet investigated it. The goal of this part of our work is to promote DD parallelism into the optimization instead of exclusively using it in the PDE simulation.
- Tools developed to enable/improve DD parallelism in PDE simulation are potentially also useful in our context.
- Improve use of derivative information to direct search methods.

<u>Long term:</u>

- Extend surrogate management approaches to black-box MDO to handle larger numbers of variables, especially mixtures of continuous and categorical variables.

The major obstacle here is the difficulty in building such high dimensional surrogates. We plan to investigate decompositional and compositional approaches to overcome this obstacle. An approach that seems promising for the effective handling of larger numbers of categorical variables is to use directional derivative information.

- Extend SAND optimization methods to better handle inexact problem information and allow the use of model hierarchies to enhance the convergence properties.

Our completely rigorous approach is implemented in three codes: Boeing's Design Explorer, and the public domain codes NOMAD C++ and NOMADm (Matlab 6). Design Explorer is in production use at Boeing, including its role in planform design for the 7E7 program. Boeing is investigating commercialization of Design Explorer. Exxon Mobil is presently licensing NOMAD for distribution in an internal package for constituent analysis of crude oil. NOMAD is being considered for other internal Exxon Mobil software packages. Boeing is now putting an FTE into investigating the pyramid approach to parallelization of these algorithms. Presently, the parallelization is rather simple, but it is effective on problem functions that require several minutes to evaluate.

NOMAD has been successfully tested by United Technologies as a part of its design package. No decision has been made on whether they will approach Rice for a license. It has also done very well on the community problem test set in groundwater remediation being developed at SAMSI and the US Army research station in Mississippi.

The Matlab version of NOMAD is in production use at Siemens for designing engine control laws to be implemented in silicon for BMW automobiles.

A predecessor of NOMAD, FOCUS, has been freely available on the LANL software web page for a couple of years. Since no licensing is required, we do not have a list of LANL users. John Dennis has participated in the LANL Workshops on Uncertainty Estimation (last one in Dec 2002) to discuss the use of surrogate management frameworks, especially the NOMAD tool, in this context. In addition, we have given presentations at LANL and at the LACSI symposia to introduce the background behind our optimization approaches and our tools. The basic versions of NOMAD are relatively easy to interface even with complex simulation codes. Therefore, often little communication takes place between NOMAD developers and users. It would be excellent to be directly involved in the application of NOMAD on LANL problems. We are ready to travel to LANL at the drop of a hat to meet with anyone interested.

The SAND tools target problems where the number of design/control variables is of the order O(n) (distributed design/control, identification problems), O(n^(2/3)) (boundary control), or only O(1), where n is the number of mesh points in the 3D PDE simulation. Since the PDEs are included as constraints into the optimization problem, the PDE solutions are also considered optimization variables.

We have discussed application of the SAND technique to a design problem related to uncertainty estimation, posed by M. Berndt and D. Tartakovsky (both LANL T7).

Preliminary work by M. Berndt and D. Tartakovsky on model problems has shown the potential of their approach, but has also revealed the need for faster optimization than the nested analysis and design (NAND) framework they have used in their tests so far. Our SAND approaches can remove this bottleneck.

We will arrange a meeting with Jim Kamm this summer/fall to see how our work on analysis and simulation of linked subsystems can be used in his V&V project.

Simulation driven optimization

Rice: W. W. Symes
LANL: Ken Hanson (CCS-2) and 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 that 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.

We have mostly considered tree-based or multiresolution AMR, following the lead of Harten and Cohen, for PDEs. 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.

Code-Based Sensitivity Analysis

Rice: M. Fagan
LANL: R. Henninger (CCS-2), Ken Hanson (CCS-2), Jim Sicilian (CCS-2), and John Turner (CCS-2)

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

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

Long term:
Longer-term goals for the code-based sensitivity project go in four different directions:

1. Applying automatic differentiation for verification of ODE-based and PDE-based computer models.
2. Applying automatic differentiation in the construction of Newton-Krylov solvers. Newton-Krylov solvers need directional derivatives. Typically, these directional derivatives are computed using a matrix-free finite difference method. Our proposed alternative is to use automatic differentiation to compute the directional derivatives.
3. Extending Adifor-style automatic differentiation to additional programming languages, notably Fortran 90, Java, and C/C++.
   The CartaBlanca project, in particular, has expressed interest in Java.
   Leverage Note: The language processing infrastructure currently used supports Fortran 90, C and C++. Consequently, (current) short-term work on Fortran 90 should be reusable for C and C++.
4. Adapting automatic differentiation to scripting languages such as Python (and possibly Perl and Ruby).
5. Extending the augmentation paradigm to include other sensitivity measures such as intervals or probability distributions.

Large-Scale Nonlinear Optimization Algorithms and Software

Rice: R. Tapia and Y. 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.

Component frameworks for simulation driven optimization

Rice: W. W. Symes
LANL: Tom Evans (CCS-4), Rob Lowrie (CCS-2), and 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 I 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, I am 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.

Imaging and time reversal in random media

Rice: L. Borcea
LANL: Jim Kamm (CCS-2), Ken Hanson (CCS-2), and Jim Sicilian (CCS-2)

We are interested in the inverse scattering problem of target identification in a random medium, in a regime with significant multipathing. Specifically, we seek to develop imaging algorithms that are statistically stable, such that the resulting images are reliable and independent of the realizations of the random medium. We have already been successful in imaging small scatterers buried in infinite random media, via active arrays of transducers (antennas). Our algorithms combine state of the art signal processing techniques with a careful statistical analysis of time reversal in random media and they are proven theoretically and numerically to be statistically stable. We wish to extend our work to finite size targets and to media with interfaces (such as the sea surface or the earth surface). This work will consist of both analysis and extensive numerical calculations, especially for the case of data gathered with very large, synthetic aperture arrays of antennas.

Application and System Performance

Adolphy Hoisie (hoisie@lanl.gov)
John Mellor-Crummey (johnmc@rice.edu)
Barbara Chapman (chapman@cs.uh.edu)
Keith Cooper (keith@rice.edu)
Jack Dongarra (dongarra@cs.utk.edu)
Robert Fowler (rjf@rice.edu)
Guohua Jin (jin@rice.edu)
Celso Mendes (cmendes@cs.uiuc.edu)
Dan Reed (reed@cs.uiuc.edu)
Linda Torczon (linda@rice.edu)

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 including operating system activity, 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. 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 ASCI 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.

Research Topics

Modeling of Application and System Performance

Investigators: Adolphy Hoisie, John Mellor-Crummey and 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 PAL at Los Alamos are the performance tool of choice in gaining insight into the performance of applications and systems.

The research in this area will span a wide range of topics. First, we will continue the application modeling work so that all important types of computations (and their associated application software) are accurately modeled. The next step in this endeavor is modeling of non-deterministic applications such as Monte-Carlo transport. Second, a major thrust will be in the enhancement of the models to include detailed effects of OS, architectural features and system activity. Based on a novel methodology developed by PAL (which led to major performance improvements on the ASCI Q machine) we will include this capability directly into our application models. Third, we will look into using our models to do application steering, hence complementing other proposed steering approaches to be explored by Rice and UIUC. For this to be possible, models will have to become dynamic and incorporate runtime information from the hardware performance monitors and other sources (e.g., NICs) as the application executes.

Fourth, we will undertake the task of trying to simplify model creation. We will attempt to provide a mechanism for: aiding model creation, enabling model description, undertaking model evaluation, and allowing for model incorporation into code. These capabilities will provide a "tool" basis for performance modeling, easing their creation and use, while at the same time allowing performance models to accumulate within a coherent structure rather than to have a sequence of one-off studies. 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 exploring how to integrate model components synthesized automatically into hand-crafted model frameworks.

Fifth, we will concentrate on the numerous applications of the models we developed. The resulting performance models can be used for scalability analysis on both existing and proposed future architectures, in procurement to compare proposed alternatives, in software development to ascertain the performance impact of code re-configuration prior to implementation, and in real-time to steer the processing of code to increase processing efficiency. Accurate models are a unique tool for architecture design. We plan to apply models of the ASCI workload to propose and design advanced architectures that maximize the performance of this workload.

Short-term tasks:

- The LANL PAL team will continue to work on analysis and modeling of applications in the ASCI workload. That work will be broadened to include other ASCI types of computations such as non-deterministic applications.
- PAL has begun an active research effort in expanding the models to include an accurate account of system effects.
- Develop infrastructure for modeling and prediction of node performance of code for IA64-based architectures.
- Refine instruction schedule modeler to better support modeling of VLIW architectures and to incorporate memory hierarchy performance model results.
- 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.

Better tools for measurement and analysis of application performance

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

On terascale systems, performance problems are varied and complex and thus a wide range of performance evaluation methods need to be supported. 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.

Short term 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.
- Explore performance data collection based on stack sampling to provide better information about dynamic context for performance measurements. Construct a tool for collecting call-graph style profiles for unmodified, optimized application binaries and explore strategies for analyzing and presenting such profiles for large applications.
- Refine the SvPablo toolkit to better support instrumentation and collection of performance data for applications of interest to LACSI.
- 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.
- 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.

- The academic performance analysis team will continue to hold performance tools workshops at LANL if the applications teams or LANL management believe additional such workshops would be productive. These workshops serve three purposes. First, they help application teams use tools developed by the academic members of LACSI to understand how their choices of data structures and algorithms affect performance. Second, they provide an opportunity for cross-disciplinary working groups to examine ASCI workload exemplars and exchange ideas. Third, they provide valuable feedback to the performance team about opportunities for enhancing tool capabilities.

Scaling the Memory Wall

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

There is a large mismatch between processor and memory speeds in today's computer systems. Consequently, typical scientific computations are starved for memory bandwidth in systems based on modern microprocessors. A multi-pronged approach is needed to address this problem. Radical changes in hardware and software design offer the most potential for overcoming the memory bottleneck. The team will explore both hardware and software strategies that have the potential for ameliorating this bottleneck. To influence the development of alternative architectures, design alternatives must be evaluated with ASCI workloads and feedback must be provided to computer architects about the findings. Efficient software for such radically different, next-generation computer systems will have very different organizing principles at the machine level. Research is needed to improve programming models so that applications can be expressed more naturally, yet mapped onto current and next-generation architectures more efficiently. For high performance on present day architectures and future systems, better compiler technology is needed for transforming applications from an organization appropriate for humans to one that is well matched to exploiting an architecture's capabilities. In particular, typical applications fail to adequately exploit temporal reuse of data at all levels of the memory hierarchy. Compiler technology for reorganizing programs to exploit latent opportunities for temporal reuse can substantially improve performance. However, compiler technology is not omniscient. Choice of application data structures can aid or thwart compiler transformation capabilities. Further investigation is needed to understand the impact that algorithms and data structures have on the compiler's ability to transform programs for high performance and to determine what kinds of data structure representations are appropriate for organizing ASCI applications so that they can be mapped efficiently to present-day and future computer systems.

To draw concrete conclusions in this area, our research plan calls for using application-driven analysis. Activity in this area will be limited until tools for helping to automate construction of detailed application-centric performance models (described in the previous subsection) become more mature.

Automatic application tuning

Investigators: Keith Cooper, Jack Dongarra, Robert Fowler, Ken Kennedy, Guohua Jin, John Mellor-Crummey, and 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* 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.

Short term tasks:

- Explore the application of adaptive search strategies to achieve ATLAS-like automatic tuning without programmer intervention by manipulating parameters to the compiler, observing the results, and adjusting the compilation parameters.

Extend strategies explored for tuning matrix multiply on the MIPS R12000 to other dense linear-algebra codes and investigate the application of these strategies to non-numerical code.
- Continue refinement of and experimentation with tools for performing complex sets of interacting program transformations to improve program performance. The focus will be on applying transformations such as fusion, tiling, time skewing, scalar replacement and storage reduction to enhance memory hierarchy utilization in modern computer systems.
- Explore strategies for model-guided, empirically based application of single program transformations to improve performance of scientific programs.
- Explore decision algorithms for applying transformations effectively in concert.

Compiler technology for exploiting modern processors

Investigators: Keith Cooper, Ken Kennedy, John Mellor-Crummey, and 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.

Short-term tasks:
- Examine assembly code produced by the ORC compiler for IA-64 to determine why the retired IPC is low and develop systematic approaches to increase retired IPC and overall application performance.
- Rewrite portions of the Vizer system, which vectorizes simple loops from x86 object code, to improve efficiency, to make it more robust, and to simplify extending it with new transformations. Extend it to handle whole programs.

Application mapping, dynamic adaptation and steering

Investigators: Dan Reed, Ken Kennedy, Celso Mendes and John Mellor-Crummey.

As computer systems grow in size and complexity, tool support is needed to facilitate the efficient mapping of large-scale applications onto these systems. Today, most applications are mapped to a set of resources at program launch and then run to completion using these resources. 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. Another issue of growing importance as the scale of systems increases is that they consume prodigious amounts of power for operation and cooling.

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 running on ASCI-scale systems to monitor their own behavior and reactively adjust their behavior to optimize performance according to one or more metrics. For this purpose, performance analysis tools must provide robust performance observation capabilities at all levels of the system and the ability to map low-level behavior to high-level program constructs.

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. For instance, dynamically managing power consumption is one way to make the operation of large-scale systems more cost effective. However, such control must not unduly sacrifice performance, otherwise the primary reason for parallel systems will be lost. Strategies for automatic performance steering based on power, 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.

Short Term Tasks:

- Explore failure modes of large-scale systems from workload traces, develop statistical characterization techniques to quantify system behavior more inexpensively and explore using application and system level "performance contracts" as the basis for monitoring and adapting behavior.
- The UIUC infrastructure for performance contracts is currently based on performance data collected by each application task captured using the PAPI interface to hardware performance counters and using the MPI profiling interface to

capture communication event data. This infrastructure will be extended incorporate contracts based on more general, system-level data such as that gathered using LANL's Supermon and MAGNET toolkits. Data gathered with these toolkits will be made available through our Autopilot sensors so that system-wide contracts can be established to guarantee a certain level of resource availability.

- The UIUC team will collaborate with the Green Destiny project at LANL on performance and intelligent power management.
- Demonstrate adaptation techniques for multi-attribute system behavior, including power management.

Computer Science Community Interaction

Linda Torczon (linda@rice.edu)
Jack Dongarra (dongarra@cs.utk.edu)
Rob Fowler (rjf@rice.edu)
Lennart Johnsson (johnsson@cs.uh.edu)
Deepak Kapur (kapur@cs.unm.edu)
Ken Kennedy (ken@cs.lanl.gov)
Rod Oldehoeft (rro@lanl.gov)
Dan Reed (reed@ncsa.uiuc.edu)

LACSI is a collaborative research effort between Los Alamos National Laboratory, Rice University, the University of Houston, the University of Illinois at Urbana-Champaign, the University of New Mexico, and the University of Tennessee at Knoxville. Effective means of supporting collaborations are important to the success of LACSI. To support collaboration, LACSI will provide a variety of opportunities for researchers from Los Alamos and the academic partner sites to visit each other, to share ideas, and to actively collaborate on technical projects.

In addition, we will organize, host, and otherwise support a series of technical workshops on topics related to the LACSI technical vision. This will include a series of workshops at LANL targeted at exposing application researchers to emerging technologies.

LACSI will also 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. This will be a traditional conference-style meeting with participation by both LACSI members and scientists from the community at large.

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