Los Alamos Computer Science Institute

FY00 and FY01 Project Report for Academic Participants

September 2001

Executive Summary:

After three years of activity, the Los Alamos Computer Science Institute has matured into a sophisticated research effort involving researchers at LANL and five academic institutions: Rice, University of Houston, University of Illinois, University of Tennessee, Knoxville, and University of New Mexico. During that time we have established a significant number of productive research collaborations between LANL and academic scientists, and have made significant progress toward realizing the original vision for LACSI - to create an effort that would preserve and foster high performance computing research, especially in areas of particular interest to LANL.

To date, LACSI has established a number of collaborations and carried out research (on the academic side) in three major areas:

1. Compilation, Systems, and Performance Evaluation of Large Scale Parallel Machines
2. Component Architectures for Rapid Application Development and Composition in a Networked Environment
3. Computational Mathematics

This work, described in detail below, has led to the publication of 85 papers and over 30 new algorithms and software tools.

In addition, LACSI has established the LACSI Annual Symposium that is beginning to be accepted as a major national meeting on high performance computing. LACSI has fostered over 40 collaborative visits among participants, of which six visits in the past year alone were for significant lengths of time. The LACSI Fellowship Program that honors top CS Graduate Students and involves summer internships at LANL or other labs has been expanded to all the ASC (DOE's Advanced Simulation and Computation Program, formerly known as ASC) labs. These activities would not have happened without the establishment of LACSI.

Finally, we have developed a plan to establish the Santa Fe Information Technology Laboratory, managed by Rice University, as a major research facility to foster open collaborations between LANL and academic partners within LACSI and other ASC-funded Centers.

In summary, we are making significant progress toward achieving the goals of LACSI. As you will find in the following report, working relationships among the partners are growing, high quality research is being conducted, and successful implementations of LACSI technology on DOE applications are beginning to take place, with more implementations in progress.

Part I: Overview and Management

The *Los Alamos Computer Science Institute (LACSI)* was created to foster world-class computer science and computational science research efforts that are relevant to the goals of the Los Alamos National Laboratory (LANL), both at LANL and at its academic partners. LACSI is a collaboration between LANL and the Rice University Center for Research on High Performance Software, along with partners at the University of Houston (UH), the University of Illinois at Urbana-Champaign (UIUC), the University of Tennessee at Knoxville, and the University of New Mexico in Albuquerque. LACSI has major components on site at LANL and at Rice University.

At the outset, the goals for LACSI were:

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

To accomplish these goals, LACSI has pursued research in a number of computer science and computational science areas of interest to Los Alamos and the ASC Program. The primary focus of this document is to present the results of the research conducted by the LACSI academic partners and to document the collaborative activities among the LACSI partners, particularly the collaborations between the academic partners and the LANL participants. The report presents research activities in three primary areas:

1. Compilation, Systems, and Performance Evaluation of Large Scale Parallel Machines
2. Component Architectures for Rapid Application Development and Composition in a Networked Environment
3. Computational Mathematics

Each of these primary research areas is further broken down into research "thrusts". For reporting purposes, these research thrusts are described separately, although many of them interact strongly with one another. The Year 3 budget for the LACSI academic partners (our first full 12 month project year) is $3.484 million, allocated as follows: 38% - Compilation, Systems, and Performance Evaluation of Large Scale Parallel Machines, 31% - Component Architectures for Rapid Application Development and Composition in a Networked Environment, 20% - Computational Mathematics and Algorithms. The remaining 11% is budgeted to support project administration, including symposium and workshop costs.

In these initial years of the Institute, we have experienced growing pains that have, in some ways, helped build the resolve of the project. Primary among these was a significant exodus from LANL of LACSI researchers with whom we were building collaborations. In some cases, this exodus meant dropping research activities for which there was now no counterpart at LANL. It also meant that new relationships and revised research activities had to be defined. This exodus led, in large part, to a joint effort by LANL and Rice University to establish the *Rice University Santa Fe Information Technology Laboratory* (SFITL) on a new campus in Santa Fe, New Mexico. Although last year's federal funding scenario delayed a Summer 2001 start, we are still hopeful the SFITL will be able to begin operations in 2002. The Santa Fe Information Technology Laboratory will conduct fundamental, open, unclassified research on information technology related to high-performance computing and networking, topics of long-term interest to the Department of Energy. It will provide a unique, world-class intellectual environment focused on computer science and information technology issues critical to the future of high performance computing. Strategically located in Santa Fe, New Mexico, SFITL will be close enough to the Los Alamos and Sandia National Laboratories to carry out intensive research collaborations, while maintaining strong ties to the emerging information technology community in Northern New Mexico. One of the goals for the SFITL will be to provide an exciting environment for DOE scientists to collaborate with academic and industrial counterparts, providing additional incentive for top young computer scientists to work for the DOE labs, and for existing scientists to remain at the labs.

Another challenge has been establishing appropriate activities and collaborations, and getting them implemented in a timely manner. Both Year one and Year two were each less than five months long, due to significant contractual delays. In addition, LACSI contracts to date have not been matched well to long-term research activities. We have been working diligently towards building an agreement more appropriate to the nature of the Institute, and have made significant progress. The next contract, for the period beginning October 1, 2001 will begin to establish the long-term nature of the project, and will include terms that are more appropriate to a research project instead of a manufacturing project.

A principal goal of LACSI is to foster collaborative relationships between LANL participants and the participating academic institutions. To date, we have been quite successful in pursuing this goal. More detail about the relationships that are continuing to build can be found in the technical reports for the various thrust areas later in this report. We provide a few highlights here. LACSI has fostered over 40 collaborative visits in the past two years, many of which involved technical talks to broader audiences. Within the past year, eight different academic participants visited LANL for significant lengths of time (in excess of a month per visit) collaborating on joint activities. Listed in this report are approximately 85 papers that have been (or soon will be) published, supported at least in part from LACSI funding. Also listed in this report are approximately 35 new algorithms or new (or improved versions of) software, libraries, or tools that have been produced at least in part as a result of LACSI efforts and funding.

It is also important to note that we are now beginning to see joint supervision of graduate students coming out of the Academic/LANL research collaborations. While we must admit this caught us a little by surprise, it is a natural metric indicating the growth of productive relationships. At present, two graduate students are under joint supervision:

(1) Student: V. Bokil (UH) - Advisors: Roland Glowinski (UH), Mac Hyman (LANL)

(2) Student: Jennifer Wightman (Rice) - Advisors: Petr Kloucek (Rice), Jerry Brackbill (LANL)

We will encourage more joint advising of graduate students through the future. A disappointment is that we are only aware of one joint publication so far. We fully expect more joint publications in the future and will take a more active, management role to help this occur.

Part of the initial proposal for forming LACSI was the creation of a LACSI Graduate Fellowship program to support students doing work in high performance computing. With the subsequent formation of the ASC institutes at Sandia (CSRI) and Livermore (ITS), the program was successfully expanded into the DOE Graduate Fellowship program.

Executive Committee:

An executive committee consisting of senior researchers from LANL, the academic partners, and other organizations manages LACSI. Current membership of the Executive Committee is as follows:

- Andy White, Institute Director, Chair, LANL
- Ken Kennedy, Academic Director, Vice-Chair, Rice
- Jack Dongarra, Univ. of Tennessee, Knoxville
- Lennart Johnsson, UH
- Deepak Kapur, UNM
- John Mellor-Crummey, Rice
- Rod Oldehoeft, LANL
- Dan Reed, UIUC
- Dan Sorensen, Rice
- Rick Stevens, ANL
- John Thorp, LANL
- Linda Torczon, Rice

This committee meets on a regular basis, with six face-to-face meetings in the past two project years, and five to six meetings each year in conference calls. It is the responsibility of this committee to review and plan activities for LACSI, ensuring that the project goals are clear, being implemented, and relevant to ASC/LANL program goals. It is also the responsibility of the Committee to provide counsel to the Institute Chair (currently Andy White) and the Academic Lead (currently Ken Kennedy). The relationship among the members of this committee is excellent.

Currently, an administrative committee (EC members White, Kennedy, Oldehoeft, Thorp, and Torczon, plus Rob Fowler and Danny Powell from Rice) meets every two weeks by teleconference to discuss management issues. Between face-to-face meetings, the entire EC holds a teleconference approximately six weeks to make major decisions and to coordinate activities. These meetings have been extremely effective in helping us manage this large and ambitious effort.

Annual Symposium:

The *LACSI Symposium* is an annual meeting that serves as a convocation of LACSI researchers to plan new research directions in workshops and in informal gatherings, as a showcase for

LACSI research efforts, and a forum for presenting outstanding research results from the national community in areas overlapping the LACSI technical vision. It is organized around traditional conference-style sessions with participation by LACSI researchers, outside scientists from academic partner institutions, and other scientists from the community at large.

The program committee for the 1st Annual LACSI Symposium was the LACSI Executive Committee. The goal for the 1st Symposium was to establish the visibility of the program, with invited talks on a variety of topics of interest to DOE and the ASC program. An impressive list of outside speakers were assembled, including Paul Messina, Larry Smarr, Steve Wallach, Gene Myers, John Reynders, Chris Johnson, Ed Seidel, and Tom Sterling. LACSI participants, including Andy White, Ken Kennedy, Ron Minnich, Dan Reed, and Bob Bixby, also gave invited talks. Approximately 160 people, from all over the country, attended the full Symposium. Additional LANL personnel (approximately 20) came for one or more of the five technical workshops on the last day.

The program committee for the 2nd Annual LACSI Symposium is chaired by Rod Oldehoeft, from LANL, and consists of senior LACSI researchers and several outside members. With the 2nd Symposium, the intent is to maintain a high quality program, while moving towards a program of peer-reviewed papers and talks, published proceedings, a new poster session activity, and a smaller number of invited talks. The final day of workshops was a particular success in 2000 and will continue in 2001. The program is a strong one, with 20 papers accepted out of a total of 60 submissions. Research sponsored through LACSI is heavily represented in the proceedings and the poster session. The workshops will not only spotlight work done under LACSI, but they will help to define future directions.

Other Global Interactions:

In addition to the activities discussed in the Part II of this report, which are associated with specific research activities, LACSI and the academic partners have participated in more general interactions with LANL and DOE in support of the goals listed above. Approximately 45 visits have taken place between lab and academic LACSI members, resulting in approximately 35 talks to audiences at those institutions that are broader than just the LACSI participants.

Rice University is running an on-going LACSI Colloquium Series, which invites LANL staff to visit and deliver lectures. To date, the following talks have been given:

Oct 20, 1999: Peter Beckman “Extreme Linux for High Performance Computing”

Dec 3, 1999: Suvas Vairacharya “SMARTS – Scalable Multithreaded Asynchronous Run-Time System for High Performance Computing”

Jan 26, 2000: Scott Haney “Parallel Object-Oriented Methods and Applications”

Feb 18, 2000: John Reynders “Components and Scalability”

April 13, 2000: Andy White “From Computer to Political Science: Societal Challenges in the 21st Century”

February 20, 2001: Jim Ahrens "Large Scale Data Visualization Using Parallel Data Streaming"

April 26, 2001: Ron Minnich "LOBOS, LinuxBIOS, and Why Clusters Need Them"

A sequence of circumstances, including the Cerro Grande fire, changes at LANL, and restrictions on DOE travel, made it very difficult to schedule as many talks as we would have liked during FY01. We plan for the level of activity will return to its intended level during the coming year.

Senior researchers from the academic partners, in particular Lennart Johnsson, Ken Kennedy, and Dan Reed, have been serving on several LANL and DOE advisory and review committees. In addition to serving on the LACSI Executive Committee, Danny Sorensen is also on the Sandia CSRI Executive Committee.

A Remaining Challenge

The principal remaining challenge in managing LACSI is to effect a transformation to a project that is managed as one effort rather than two—an effort at LANL and an academic partnership. We made significant progress toward this goal this summer by cooperating with LANL in the production of the "Institutes" part of the FY2002 ASC Implementation Plan (IP). As follow-on, we plan to produce a joint statement of work for the entire Institute by building on the current SOW for the academic partnership and the IP document from this summer. When these steps are completed, it will become possible for all the participants to see the entire scope of activities within LACSI and to more easily identify potential collaborations.

Part II: Research

A. Compilation, Systems, and Performance Evaluation of Large Scale Parallel Machines

The major goals of this research area are to advance the state of high performance computing by: contributing to our ability to measure and analyze the performance of large parallel programs; developing methods for improving such programs, either manually or through the use of new programming language constructs and compiler techniques; and having a practical impact by deploying these advances for use on important computations, especially at LANL.

Compilation strategies

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

Abstract: The focus of this project is to develop compiler technology and compiler-assisted tools that assist application developers in effectively exploiting current and future generation scalable parallel systems. To date, this effort has had three major research thrusts:

1. Short-term research focused on the design and development of compiler-assisted tools for pinpointing and helping users identify shortcomings in an application's node performance,
2. Medium-term research into compiler and run-time techniques for improving memory hierarchy utilization of node programs to help codes yield a higher fraction of peak processor performance, and

3. Long-term research on data-parallel compiler technology and scalable partitioning algorithms that aims to simplify the process of constructing high-performance codes for scalable parallel architectures by having a compiler optimize communication and management of off-processor data.

We discuss the motivation and accomplishments in each of these areas in the following sections, as well as point out interactions with LANL scientists in these efforts.

Tools for Performance Analysis of Node Programs

It is increasingly difficult for large scientific programs to attain a significant fraction of peak performance on systems based on microprocessors with substantial instruction level parallelism and with deep memory hierarchies. However, algorithm and application designers, in large part do still not use performance analysis and tuning tools on a day-to-day basis because they fail to address many of the key issues facing developers. To address many of the issues that have limited the usability and the utility of most existing tools, we developed HPCView – a toolkit for combining multiple sets of program profile data, correlating the data with source code, and generating a database that can be analyzed portably and collaboratively with commodity Web browsers. HPCView presents performance data in a hierarchical fashion that facilitates top-down analysis in a language- and platform-independent way. It supports comparison of multiple metrics to get a broad understanding of factors contributing to a code's performance. Rather than just presenting measured performance metrics, such as those gathered using hardware performance counters, HPCView supports the computation and presentation of derived metrics expressed equationally in terms of existing metrics.

The data reported by vendor tools are presented at the source line level and are summarized at the procedure level. Neither of these choices is the right level of aggregation. After aggressive compiler optimization on today's out-of-order processors with multiple functional units and non-blocking caches, it is very difficult to attribute costs at the individual line level. Furthermore, overall performance may depend more on issues of instruction balance and whether or not prefetching can be used to hide the latency of cache misses. These questions can only be addressed by examining performance measures at levels of aggregation intermediate between the line and procedure levels. We developed *bloop*, a tool for recovering loop structure by analyzing executable code, to enable loop-level aggregation of data in HPCView. Structure information collected with *bloop* is passed into HPCView, which uses this information to organize line-level performance information into more useful loop-level summaries. Currently, HPCView and *bloop* support the collection of hardware performance data and program analysis on SGI/MIPS and Compaq/Alpha platforms. A third related tool *xprof* is currently under development for translating performance measurements taken with Compaq's Dynamic Continuous Profiling Infrastructure (DCPI) into a form suitable for use by HPCView.

In August 2001, we installed version 2.0 of the HPCView toolkit at LANL, which we updated to version 2.01 in September. We have assisted application developers at LANL, including Randy Baker (PARTISN) and Jack Horner (SAGE), in applying HPCView to ASC codes. Rob Fowler from Rice University spent six weeks at LANL in August and September 2001, spending some of his time working with users to apply these tools to their applications.

Improving Memory Hierarchy Performance

The gap between CPU speed and memory speed is widening as each new generation of computer hardware is introduced. For a program to achieve high performance on modern computer systems with multi-level memory hierarchies, it must effectively reuse data at each level of the memory hierarchy to overcome the impact of access latency and the limited bandwidths available from main memory. In this area, we have been investigating a variety of compile- and run-time strategies for improving memory hierarchy utilization for both regular and irregular problems. The focus of the work through mid-2001 was on designing and prototyping techniques. In the subsections below, we describe the significant efforts in this area.

Since mid-2001, we have been working to extend the SGI Pro64 open source compiler into a form suitable to serve as an infrastructure on which to build source-to-source transformation tools that embody these strategies for improving memory hierarchy performance. To date, the work with the SGI Pro64 infrastructure has focused on adapting the Cray FORTRAN 90 front-end (included in the release) and the intermediate representation into forms suitable for supporting source-to-source transformation tools capable of handling production FORTRAN 90 codes. This effort is being done in coordination with LACSI researchers at the University of Houston.

Compile-Time Data and Computation Reordering

Most studies of software techniques for memory hierarchy management have focused on reducing, or hiding the impact of, cache miss latency in loops. Because today's machines are increasingly limited by insufficient memory bandwidth, latency-oriented techniques can have only limited impact; they do not seek to minimize the total volume of data transferred to and from memory. To compensate for bandwidth limitations, this research explores techniques that increase global cache reuse, namely, reusing data across whole program and over the entire data space. We explored a two-step strategy for improving bandwidth utilization. The first step fuses computations on the same set of data to enable the caching of repeated accesses. The second step groups data used by each computation to make them contiguous. The first step reduces the frequency of memory access and the second step improves its efficiency. Experiments demonstrate the effectiveness of this strategy. This approach is suitable for automating in production compilers.

Run-time Data and Computation Reordering for Irregular Applications

Loop blocking and prefetching transformations have proven themselves for improving memory performance on regular applications; however, these techniques are not as effective for irregular applications. We therefore are investigating data and computation reordering techniques designed specifically for irregular applications. We have been focusing on coordinated data and computation reordering based on space-filling curves and we have introduced a new architecture-independent multi-level blocking strategy for irregular applications. We studied the effectiveness of these techniques applied to a molecular dynamics benchmark, an Air-Force particle hydrodynamics code, and a communication kernel from CHAD – a LANL ASC application (from Manjit Sahota and Bob Robey) that models combustion with fuel sprays on an unstructured mesh using hexahedral elements and their degenerate forms. For the Air-Force particle hydrodynamics code and the molecular dynamics benchmark, the most effective reordering reduced overall execution time by a factor of two for the first code and a factor of four for the second. For the communication kernel from CHAD, although we found that the space-filling curve ordering was no better than the original order given for test meshes, it was a factor of two better than a random ordering. This implies that it may prove useful for reordering data and computation after partitioning in a parallel system, after multiple steps of adaptive mesh refinement.

Automatic Recursive Blocking of Loop Nests

Recently, there have been several experimental and theoretical results showing the significant performance benefit of using recursive algorithms on both multi-level memory hierarchies and on shared-memory systems. This is because such algorithms have the data reuse characteristics of a blocked algorithm that is simultaneously blocked at many different levels. Most existing applications, however, are written using ordinary loops. We developed new compiler transformation converts loop nests with carried dependences into a form to which recursive partitioning can be applied automatically. This algorithm is fast and effective, handling loop nests with arbitrary nesting and control flow. The transformation results in substantial performance improvements for several linear algebra codes. As a side effect of this work, we developed an improved algorithm for transitive dependence analysis (a powerful technique used in the recursion transformation and other loop transformations) that is much faster in practice than the best previously known algorithm.

Improving Temporal Locality for Iterative Computations

Good spatial locality of memory accesses means that most or all data fetched into cache will be used. Many data layout and code reorganization transformations, either applied by a compiler or by hand, are known to improve spatial locality and mitigate interference. On systems with limited memory bandwidth, however, good spatial locality alone is not sufficient to achieve high performance. To achieve high performance on recent systems with typical memory hierarchies, programs must also exhibit substantial temporal locality. That is, on average, each data item must be accessed many times in cache before being evicted. Unfortunately, temporal reuse is low in iterative scientific programs in which each memory location is touched a small number of times per iteration, for which there are relatively few computations performed on each data element, and for which the data does not fit in cache.

Tiling loops over a program's data domain is a strategy commonly used to increase both spatial and temporal reuse in one or more levels of cache. Tiling reshapes an iteration space over a data domain by partitioning it into pieces that fit comfortably into cache and then completes all computations on each piece before moving to the next. Tiling rearranges the order of computation so that multiple references to a data element occur in inner loops while that element is still resident in cache. Recursive blocking generalizes multi-level tiling by providing locality at each level of the recursion, thus achieving good performance across a wide range of systems, independent of specific memory hierarchy implementations.

However, even if tiling and recursively blocking are successful at extracting maximal data reuse in each pass through the loops over the data domain, this often does not result in a sufficient reduction in memory bandwidth requirements. In this work, we therefore focus on techniques that improve temporal locality in iterative scientific applications.

For programs where the core computation consists of "stencil" computations in which each data element is updated based on values in its neighborhood, we have developed a technique we call *prismatic time skewing* that exploits not only reuse within an iteration over the data domain, but also across iterations in an enclosing convergence loop.

Prismatic time skewing partitions a multi-loop iteration space into prisms that are skewed with respect to both spatial and temporal (or convergence) dimensions. Novel aspects of this work include: multi-dimensional loop skewing; handling carried data dependences in the skewed loops without additional storage; bi-directional skewing to accommodate periodic boundary conditions; and an analysis and transformation strategy that works inter-procedurally. We combine prismatic skewing with a recursive blocking strategy to boost reuse at all levels in a memory hierarchy. A preliminary evaluation of these techniques shows significant performance improvements compared both to original codes and to methods described previously in the literature. With an inter-procedural application of our techniques to part of a large application, we were able to reduce overall primary cache misses by 27% and secondary cache misses by 119%.

Improving Scalarization

FORTRAN 90 array syntax adds expressive power to the language by enabling operations on, and assignments to, array sections. On conventional scalar processors, each array statement must be converted into a loop that maintains the correct semantics by a process called scalarization. The semantics of FORTRAN 90 array statements specify that the result must be consistent with fetching all inputs before any outputs are stored. Since an array variable may appear on both the right and left hand sides of an assignment statement, a scalarized loop for an array statement may require loads and stores to the same variable. It is simple to satisfy the required semantics by generating scalarized code that first computes the result into a temporary array, and then copies the result into the left-hand side array variable. Using extra temporary arrays, however, both increases memory traffic and increases pressure on caches. Thus, techniques to avoid using temporary arrays or minimize their size are important for performance. We developed a strategy for minimizing the size of temporary arrays needed by employing loop alignment and loop skewing. Our experiments with loop alignment show that it is extremely effective in improving memory hierarchy performance of FORTRAN 90 array code.

Data-parallel Compiler Technology

Hand-coding parallel programs to exploit parallel machines effectively is exceedingly difficult. Achieving top performance requires careful optimization to address a myriad of issues including data and computation partitioning, load-balance, serialization, communication, and memory hierarchy utilization, to name a few. High-level data-parallel languages such as High Performance FORTRAN (HPF) have been proposed and designed to simplify the task of developing high-performance parallel applications. HPF enables application developers to write programs using a global address space model in which parallelism and communication are largely implicit. While HPF compilers have become available for many parallel systems, they have failed to attract broad interest among application developers because commercial HPF compilers have largely failed to deliver application performance even roughly comparable to hand-coded implementations, especially for tightly coupled computations.

For the past five years, our research on data-parallel compilers has focused on developing compiler analysis and code generation strategies to enable compilers for data-parallel languages to generate code that delivers performance comparable to hand-coded implementations. To drive this effort, we have focused on data-parallel compilation techniques for tightly coupled applications. As part of our work, we have developed data-parallel compiler support for generalized multipartitioning, a sophisticated skewed block-cyclic distribution that leads to particularly efficient parallelization of line-sweep computations. (This partitioning strategy is described in more detail in the subsection below.) Much of this work entailed the development of a range of sophisticated analysis and code generation strategies. These optimizations include techniques for complex computation partitioning, communication vectorization to amortize fixed communication overheads by increasing communication granularity, communication factoring to generate simpler communication code, communication aggregation to coalesce multiple messages to the same destination, use of asynchronous communication for latency and asynchrony tolerance, and communication pipelining to reduce buffering requirements and overlap communication with computation. Using these techniques, in August 2001 we were able to parallelize a serial version of the NAS SP computational fluid dynamics application benchmark and achieve performance on 64 processors that is within 13% of a hand-coded parallel version developed by application scientists at NASA Ames Research Laboratory. For the NAS BT computational fluid dynamics benchmark, the code generated by our compiler (dHPF) outperformed, by a small margin, NASA's hand-coded version.

To transfer some of this technology into the hands of practitioners, we are, with additional funding from the DOE Office of Science, beginning a project to develop a robust open-source compiler for Co-array FORTRAN, which is a single-program, multiple-data (SPMD) parallel programming model based on FORTRAN 90. Co-array FORTRAN can be used to write parallel programs much in the same way that FORTRAN 90 is used with MPI right now; however, by including mechanisms for expressing synchronization and interprocessor data movement in the language, Co-array FORTRAN enables compiler technology to be brought to bear for optimizing communication.

Generalized Multipartitioning for Parallelizing Line Sweep Computations

Line sweeps are used to solve one-dimensional recurrences along each dimension of a multi-dimensional discretized domain. For example, this method is the basis for Alternating Direction Implicit (ADI) integration — a widely used numerical technique for solving partial differential equations such as the Navier-Stokes equation. It is also at the heart of a variety of other numerical methods and solution techniques. Parallelizing line sweep computations is important because they address important classes of problems and they are computationally intensive.

To support better parallelization of line sweep computations, a sophisticated strategy for partitioning data and computation known as multipartitioning was developed a decade ago. Multipartitioning distributes an array of two or more dimensions among a set of processors so that for computations performing a line sweep along any one of the array's data dimensions, (1) all processors are active in each step of the computation, (2) load-balance is nearly perfect, and (3) only a modest amount of coarse-grain communication is needed. These properties are achieved by carefully assigning each processor a balanced number of tiles between each pair of adjacent cuts (hyperplanes) along any partitioned data dimension.

Multipartitioning strategies described in the literature assign only one tile per processor per hyperplane across each distributed dimension. While this class of multipartitionings is optimal for two dimensions, it is optimal for three dimensions only when the number of processors is a perfect square or a prime. Our work considers the more general problem of computing optimal multipartitionings for *d*-dimensional data volumes on an arbitrary number of processors. We devised an algorithm that computes an optimal multipartitioning for this general case.

In particular, we developed a fast algorithm that uses a cost model to select the optimal data partitioning with the lowest communication cost and full processor utilization. The data array is multipartitioned in such a way that the number of tiles in each slice is a multiple of the number of processors. We show that having a partitioning in which the number of tiles in each slice is a multiple of the number of processors—an obvious necessary condition—is also a sufficient condition for creating a balanced mapping of tiles to processors. We give a constructive method for building this mapping using new techniques based on modular mappings. These techniques assign the optimal tile decomposition computed by the partitioning algorithm, to the physical processors that are going to compute on them.

Publications

1. Alain Darte, Daniel Chavarria-Miranda, Robert Fowler, and John Mellor-Crummey. On Efficient Parallelization of Line Sweep Computations. In *Proceedings of the International Symposium on Compilers for Parallel Computers*, Edinburgh, Scotland, June 2001.

2. Alain Darte, Daniel Chavarria-Miranda, Robert Fowler, and John Mellor-Crummey. Generalized Multipartitionings. In *Proceedings of Los Alamos Computer Science Institute Symposium*, Santa Fe, NM, Oct. 2001. (To appear.)

3. Chen Ding and Ken Kennedy. Improving effective bandwidth through compiler enhancement of global cache reuse. In International Parallel and Distributed Processing Symposium, San Francisco, CA, Apr. 2001. (Best Paper Award.).

4. Daniel Chavarria-Miranda, John Mellor-Crummey and Trushar Sarang. Data-parallel Compiler Support for Multipartitioning. In Proceedings of the European Conference on Parallel Processing (Europar), Manchester, UK. Lecture notes in Computer Science, 2150, Springer-Verlag, August 2001.

5. Daniel Chavarria-Miranda and J. Mellor-Crummey. Towards Compiler Support for Scalable Parallelism using Multipartitioning. In *Proceedings of LCR2000: 5th International Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers.* Lecture Notes in Computer Science 1915, Springer, 2000.

6. Guohua Jin, John Mellor-Crummey and Robert Fowler. Increasing Temporal Locality with Skewing and Recursive Blocking, In *Proceedings of SC2001*, Denver, CO. Nov. 2001. (To appear.)

7. Ken Kennedy. Fast Greedy Weighted Fusion. In *Proceedings of International Conference on Supercomputing*, pages 131-140 Santa Fe, NM, May 2000.

8. John Mellor-Crummey, Vikram Adve, Bradley Broom, Daniel Chavarria-Miranda, Robert Fowler, Guohua Jin, Ken Kennedy and Qing Yi. Advanced Optimization Strategies in the Rice dHPF Compiler. *Concurrency Practice and Experience: Special Issue on High Performance FORTRAN*, 2001. (To appear.)

9. John Mellor-Crummey, Robert Fowler, and David Whalley. Tools for Application-Oriented Performance Tuning. In Proceedings of the International Conference on Supercomputing (ICS 2001), pages 154-165, Sorrento, Italy, June 2001.

10. John Mellor-Crummey, Robert Fowler, and David Whalley. On providing useful information for analyzing and tuning applications. In Proceedings of the International Conference on Measurement and Modeling of Computer Systems (Sigmetrics 2001), pages 332-333, Cambridge, Mass, June 2001.

11. John Mellor-Crummey, Robert Fowler, and Gabriel Marin. HPCView: A Tool for Top-down Performance Analysis. In Proceedings of the Los Alamos Computer Science Institute Symposium, Santa Fe, NM. (To appear Oct. 2001.)

12. J. Mellor-Crummey, D. Whalley and K. Kennedy. "Improving Memory Hierarchy Performance for Irregular Applications Using Data and Computation Reorderings," *Intl. Journal of Parallel Programming*, 29(3), June 2001.

13. Qing Yi, Vikram Adve, Ken Kennedy, Transforming Loops to Recursion for Multi-Level Memory Hierarchies, In *Proceedings of ACM Conference on Programming Language Design and Implementation*. Vancouver, British Columbia, Canada. June 2000.

14. Yuan Zhao and Ken Kennedy. Scalarizing FORTRAN 90 Array Syntax. In *Proceedings of the Los Alamos Computer Science Institute Symposium*, Santa Fe, NM. Oct. 2001. (To appear.)

15. K. Zhang, J. Mellor-Crummey, R. Fowler. Compilation and Runtime Optimizations for Software Distributed Shared Memory. In *Proceedings of LCR2000: 5th Intl. Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers.* Lecture Notes in Computer Science 1915, Springer, 2000.

Visits

August 1999: John Mellor-Crummey spent three days at LANL talking with various LANL staff members including Yong Luo, Kirk Cameron, Matthe Madhave, Maya Gokhale, Harvey Wasserman, Federico Bassetti, and others about research challenges in high performance computing.

Summer 2000. Daniel Chavarria-Miranda visited LANL as a graduate student intern in the ACL.

August 2000: Rob Fowler visited Sandia National Laboratories and LANL for one day each, giving talks on HPCView to prospective users at each laboratory and working one-on-one with prospective users.

August 2000: After the LACSI Symposium, Rob Fowler, John Mellor-Crummey, and Richard Hanson met with Bob Malone and the Parallel Ocean modeling group to explore potential collaboration in parallel algorithms for ocean modeling. John Mellor-Crummey gave a talk entitled *Compilers, Tools, and Run-time Systems for High-Performance Computing.* John Mellor-Crummey and Rob Fowler spent a day meeting with Jeff Brown, Jack Horner, Michael Ham, Rich Graham, Brian Lally and Bob Robey to introduce them to the HPCView performance tool developed at Rice.

August 2000: John Mellor-Crummey gave a talk entitled *Compilers, Tools, and Run-time Systems for High-Performance Computing* as part of the LACSI Symposium workshop on systems software. Rob Fowler gave a talk on HPCView in the parallel software tools workshop.

January 2001: Rob Fowler visited LANL to work on research planning issues with John Thorp and others.

February 2001: Rob Fowler attended a meeting on the UNM campus in Albuquerque between LANL research staff and UNM Computer Science and Engineering faculty to discuss potential research areas to fund as part of UNM's membership in LACSI. Presented a talk on the research of the LACSI academic partners.

May 2001: Rob Fowler presented a talk on External Research to LANL's internal ASC/Institutes Project Review meeting.

August 2001 - September 2001: Robert Fowler visited LANL for six weeks to work on a variety of projects related to this research and to future directions for LACSI and scalable high performance computing at LANL.

Other Interactions

John Mellor-Crummey has been serving as a member of the program committee for *Second Annual LACSI Symposium*, to be held at Santa Fe in October 2001.

Software Distributions

HPCView has been installed at LANL since August 1999, and has been in use by LANL Application Scientists. HPCView 2.01 was installed at LANL and made available for use in September 2001.

Compilation Issues for High-Performance Microprocessors:

[Formerly called "IA-64 Compilation Issues"]

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

This project is focused on developing techniques to improve the performance of scientific codes on high-performance microprocessors, such as Intel's Itanium processor. Over the past two years, our work has gone in three distinct directions: 1) fundamental research on techniques that commercial compilers will need in order to generate high-quality code for modern, high-performance microprocessors; 2) fundamental research on how to improve the quality of optimization that a compiler can provide; and 3) efforts to transfer technology into compiler systems other than our research compiler.

Code generation techniques:

Next generation processors, like the Itanium, present a number of new challenges to code generation. We have been working on new algorithms for several problems that arise in code generation for these machines.

Scheduling for partitioned-register-set machines: On machines with partitioned register sets, the compiler must not only schedule each operation and allocate registers; it must also decide the placement, by cluster, of each operation and its operands. Good placements should increase functional-unit utilization, decrease running time, and decrease inter-cluster data traffic. Bad placements will increase running time, increase idle time on functional units, and increase inter-cluster traffic. We have implemented a new algorithm for integrated placement and scheduling. The current version operates inside an assembly-to-assembly translator for the TMS320C6200 microprocessor. When our Itanium code generator is finished, we will port the implementation into that framework. Early results show that the algorithm is effective at improving functional unit utilization and reducing inter-cluster traffic. On the TMS320C6200, this does not translate into faster execution, because of the long branch latencies—-4 delay slots per branch. Instead, the programs have a larger percentage of unfilled delay slots, so they execute NOPS in those positions. On machines with lower branch latencies, the improvement in running time should be larger.

Call-graph construction from compiled code: A number of new tools apply code-optimization or code-generation techniques to compiled code—that is, code that has already been scheduled and allocated. These tools arise in performance measurement and monitoring, dynamic compilation, and binary translation. However, some of the algorithms used to build the fundamental data-structures that underlie optimization and translation break down in the presence of features that can appear in compiled code. For example, the use of branch-to-register operations can severely degrade the precision of either a control-flow graph or a call graph. Similarly, the appearance of operations in delay slots—particularly labeled operations and branches—adds complexity to call graph and control-flow graph construction algorithms. We have developed a new algorithm for building control-flow graph that accurately and efficiently handles operations in delay slots, including branches in the delay slots of other branches. This kind of work is a necessary precursor to any optimization of compiled code.

Improvements to register allocation: We have developed a faster technique for copy coalescing—the phase of a graph-coloring register allocator that dominates total allocation cost. Our experimental results suggest decreases on the order of 50% for the compile-time used in coalescing. This should produce a corresponding decrease in total allocation time. This changes only the amount of time required for allocation. It does not change the resulting code. By lowering the cost of the best-practice register-allocation algorithm, we hope to make it applicable in situations where compile-time matters, such as dynamic compilation and optimization (Java JITs).

Itanium code generator

We are building a code generator for the Itanium ISA that will operate in our mainline research compiler. The code generator will allow us to experiment directly on Itanium systems. (Rice has recently obtained its first Itanium system, which we are using to debug the code generator.) We anticipate distributing the code generator in the same way that we distribute existing portions of the compiler—through our project web site.

Improving optimization

We have begun an effort aimed at understanding the fundamental tradeoffs in code optimization. Traditional compilers apply the same sequence of transformations to every program that they see. This practice ignores well-known facts. Programs differ in their basic properties—control-flow patterns, data sharing, length, and ratio of flops to memory references. Transformations both create and destroy opportunities for one another. Two transformations that purport to solve the same problem catch different subsets of the possible improvements. In truth, it is difficult to determine the "best" transformation sequence for a program.

This problem arises in classical scalar optimization. We do not know enough, today, about the interactions between transformations to propose good transformation sequences. We cannot relate transformation sequences to program properties. We do not know how far a given sequence is from optimal.

We have built a tool for exploring this problem—an adaptive compiler. It discovers, experimentally, the best transformation sequence for a given input program and a given set of input data. It takes an externally provided objective function and finds a transformation sequence to minimize the program's score under the objective function. The function can involve, essentially, any measurable parameter of the executable code—speed, space, power consumption, page faults, etc.—or combinations of them.

We are using this tool to explore adaptive compilation: to understand the tradeoffs between different objectives, to evaluate quantitatively the effectiveness of transformations and combinations of transformations, and to discover fundamental knowledge about the compilation process. We expect this work to produce new knowledge, new approaches to structuring compilers, and new ideas for transformations. We are working on ways to make these techniques practical.

A major goal of this research is to characterize the search spaces that the adaptive compiler explores. This knowledge will allow us to construct more effective search strategies, leading to faster compilations. We need such knowledge before we can use these ideas to attack harder problems that have higher payoff, such as distribution, blocking, and parallelization strategies. (We also need good estimators to optimize over those strategies, because the cost per experimental run rises significantly.)

Some of the preliminary results of this work will appear in a paper, titled "Adaptive Optimizing Compilers for the 21st Century" that will be presented at the LACSI 2001 Symposium on October 16, 2001.

Technology transfer:

At the urging of John Reynders (then at LANL's ACL), we looked at how to improve the quality of code generated by GCC. This involved discussions with Mark Mitchell of Code Sourcery and Steve Karmesin of LANL. We identified several specific areas for further work. These included development of a modern (i.e., Chaitin-Briggs) register allocator for GCC and improved scalar optimization, particularly a dead-code eliminator that was strong enough to remove the empty loops left behind on some POOMA codes.

We pursued this effort in two directions. We built an SSA-construction pass for GCC as a prelude for constructing both a register allocator and a dead-code eliminator, and we investigated the potential for using GCC in our adaptive compiler work (see 2 above). The former effort was halted when the GCC community released an incomplete, but officially-sanctioned SSA-construction pass. When that pass has become stable and bug-free, we may resume some of our work in GCC. The latter effort convinced us that GCC is an unsuitable target for the adaptive compiler work. The individual passes in GCC are so intertwined that reordering them is inconceivable.

In the last two months, we have taken a different approach to improving GCC. We have begun the legal work to provide the Free Software Foundation with the licenses that they will need to build a modern register allocator. At this point, it appears that both IBM and Rice University will provide licenses that allow inclusion of their patented techniques in GCC, for non-profit use. We have also offered our assistance to the GCC developer who will actually implement the new register allocator, including direct advice, code for our own implementation, and (as yet) unpublished bug fixes.

Visits:

Spring 2001: Keith Cooper visited LANL and the University of New Mexico to give talks on the use of genetic algorithms in the selection of compiler optimization strategies and to work on a collaboration on that subject.

Publications:

1. K.D. Cooper, D. Subramanian, and L. Torczon, "Adaptive Optimizing Compilers for the 21st Century", to appear at the 2001 LACSI Symposium. (Note: A paper that reports more quantitative results in preparation for the Nov 17 PLDI deadline.)
2. Z. Budlimlic, K.D. Cooper, T.J. Harvey, K. Kennedy, T. Oberg, and S. Reeves, Paper in preparation, on register coalescing from the SSA graph, as opposed to the interference graph. (Note: This new technique should radically reduce the cost of register coalescing, which is the most expensive part of register allocation. It should also allow the construction of an efficient, standalone coalescing phase--one impediment to quality code generation in JITs.)
3. K.D. Cooper, T.J. Harvey, and T. Waterman, Paper in preparation on control-flow graph reconstruction, for PLDI deadline, Nov 17.
4. Anshu Das Gupta, Technical report on cluster assignment and scheduling for partitioned register set machines. (Note: In this paper, we implemented a new algorithm for integrated cluster assignment and scheduling. We were able to obtain better functional unit utilization, but lost the benefits due to the high branch latency on the target machine. [This work begins from the output of another compiler and improves it, so it represents a lower bound on what we would expect in a from-scratch compiler.] Until we can

improve the branch-latency problem, this work will be difficult to publish in a first-class venue.)

Tools

Investigators: Dan Reed, Ruth Aydt, Barbara Chapman, Ken Kennedy, John Mellor-Crummey, Rob Fowler

Abstract: We are developing a new generation of software tools that integrate compilation research with real-time performance measurement, adaptive control systems, and intelligent data visualization and control. The goal of this work is to create an intelligent performance toolkit that allows software developers to create and runtime systems to negotiate "performance contracts" among software components and hardware/software systems.

During the reporting period, we made considerable progress toward fulfilling this goal on a variety of fronts. We have enhanced our tool infrastructure to support more dynamic instrumentation, control, and visualization. We have added the ability to capture hardware counter information, extended our system monitoring tools, explored data-compression techniques, and ported the tools to a wider range of platforms. Performance contracts have been more clearly defined and techniques for predicting application performance based on a combination of application resource stimuli and expected resource responses have been developed.

With additional support from other sources, a contract-monitoring infrastructure has been deployed both within clusters and across wide-area computational grids. With our fuzzy logic decision procedure component, the contract monitoring system has been used to detect unexpected decreases in application performance, and to identify the overloaded resource causing the problem. These efforts form the basis for the end-to-end intelligent performance toolkit where deep integration with compiler techniques and automatic adaptive control will obsolete the need for manual instrumentation and adaptation of an application. Individual efforts are described in detail in the following sections.

Objectives and Efforts

For performance tools to be useful, they must be available on a range of high-end computing platforms. In collaboration with VA Linux and IBM Research, we ported the SvPablo graphical user interface, instrumenting parsers, and data capture library to Linux and the IBM SP. SvPablo has been installed and tested on the Albuquerque High Performance Computing Center's (AHPCC) Linux cluster.

We are now working on porting SvPablo to IA-64 platforms, which will allow us to do performance analysis on this new architecture and conduct performance comparisons between the IA-64 and IA-32.

To gain additional insight into large-scale application performance and to identify research challenges for scalable performance tools, we integrated SvPablo with the publicly available Tennessee PAPI toolkit, a portable API to capture hardware performance counter information. This enables SvPablo to correlate application source code and hardware counter values, and to identify the dynamic patterns of software and hardware interactions occurring during the program execution. The SDDF-driven infrastructure of the SvPablo GUI made the addition of these new metrics possible without requiring a major rewrite of the display code.

To support more dynamic displays of real-time performance information, the SvPablo graphical user interface has been extended to provide a utility to generate call graphs for user applications. The call graph can be generated for all types of function calls, or for specific types, for example: MPI calls, I/O calls, HDF calls, etc. The call graph display correlates each stage of the application execution with the corresponding source code in real-time, allowing users to identify problems in the application and do real-time performance steering.

To support real-time adjustment of instrumentation points and provide a uniform interface for performance data browsing, we have begun integrating our SvPablo toolkit and the Dyninst dynamic instrumentation package, from the University of Maryland. We created a prototype tool that enabled us to dynamically instrument object codes originally created by C and FORTRAN compilers, and produce performance data in the same format as the data produced by SvPablo's instrumentation functions. Thus, it is possible to use SvPablo's GUI to browse the captured data containing information on function call durations and counts.

With this tool, one can also interactively stop program execution, remove or change the instrumentation points, and continue execution. This prototype was constructed and tested for sequential programs, and is now being extended for parallel programs based on MPI.

An important factor in overall application performance is system-level information. To allow easy access to this information in a uniform fashion across a range of platforms, we developed the Autopilot Performance Monitor component, which allows the capture of system-wide performance metrics. This component consists of two kinds of processes: collectors, which run on the machines to be monitored and retrieve statistics, and recorders, which take the data and process, record or output it. The set of available metrics includes metrics related to CPU activity, main and virtual memory, disk I/O and network traffic. Autopilot is used for communication between these processes. Collectors contain Autopilot sensors that capture the selected performance data and pass the data to Autopilot clients on recorders. Those recorders can print the incoming data, and optionally save that information in SDDF format for later analysis. A recorder could also be embedded in the contract-monitoring infrastructure, allowing decisions to be made on a combination of application and system-level measurements.

To address the volume of data potentially output by the hardware counter, application, and system-level sensors, we extended our clustering and projection pursuit analysis components. These two packages were originally developed to process performance data recorded in files. We created extensions that allowed both packages to accept input data from Autopilot sensors in real-time.

In a separate research effort on methodologies for reducing data volume, we have been working on characterization of performance data and its compression using different layouts and algorithms. In particular, we have been experimenting with both lossy and lossless methods by treating the performance data as an image using spatial (processor and metric) and temporal redundancy. Our goal is to come up with a hybrid algorithm that statistically achieves optimal compression.

Critical to our intelligent performance toolkit is the performance contract, which specifies the expected performance of an application on a particular set of resources. In conjunction with the NSF-funded Grid Application Development Software (GrADS) project, we have refined our concept of the performance contract and have developed an infrastructure for contract monitoring and violation detection.

Autopilot sensors inserted in the application capture performance measurements and make them available to contract monitor module(s) responsible for detecting performance problems. Fuzzy logic rules that are part of the contract monitor systems control when a contract is violated. The tolerance of these rules can be adjusted to accommodate some degree of deviation from the predicted performance without triggering a violation. In addition to flagging a violation, the rules are designed to help identify the cause of the violation (e.g., determining which processor is overloaded or which network is slow).

Work to date has focused primarily on formulating the performance contract specification language and building the infrastructure needed to verify the feasibility of this approach with real applications and systems. We have manually inserted probes, refined contracts, tuned tolerance levels, and considered appropriate actions when a violation was reported.

Based on this work we are confident that the approach has merit and have begun to take steps to automate the process, leading us toward our goal of developing software tools that integrate compilation research with real-time performance measurement, adaptive control systems, and intelligent data visualization and control.

Central to the performance contract is a performance model that predicts application performance based on a set of input parameters, such as resource capabilities. Models can be derived from a variety of sources, including application and library developer knowledge, compilers, and execution history. In addition, in conjunction with the NSF GrADS project, we have tested developer-supplied models, and have formulated an application signature performance model. This model uses historical information about application resource demands, together with estimates of resource responses to those demands, to predict execution metrics that express rates of progress. The application signature model allows one to predict performance without relying on intimate knowledge of the application.

Our experiments to date have shown this method allows the contract monitoring system to detect declines in application performance and indicates the cause of the drop. This work will be presented at the *2nd International Workshop on Grid Computing*, and is the basis for Vraalsen's M.S. thesis.

The main Pablo Group website is at. Software distributions, publications, discussion of efforts are all available from this page.

Publications:

1. F. Vraalsen, R. Aydt, C. Mendes, and D. Reed, Performance Contracts: Predicting and Monitoring Grid Application Behavior, *Proceedings of the 2nd International Workshop on Grid Computing*, November 2001, *to appear*

2. F. Vraalsen, Performance Contracts: Predicting and Monitoring Grid Application Behavior, *M.S. Thesis*, University of Illinois at Urbana-Champaign, Fall 2001

Visits and Other Interactions

December 1999: Ruth Aydt, Dan Reed, and Dan Wells spent two days at LANL with John Reynders and others from various projects at the Institute. The visit covered multiple projects (ASC SIO, ASC DVC, and discussion of collaboration with the Institute, as well as participation in the open source meeting).

March 2000: Visit from LANL LACSI participants (John Reynders and others) to the Pablo group.

August 2000: Dan Reed presented an invited talk entitled *Grid Computing* at the LACSI Symposium, Santa Fe, NM,

August 2000: Ying Zhang and Celso Mendes, *Scalable Tools for High Performance Computing*, Parallel Software Tools Workshop, LACSI Symposium, Santa Fe, NM,

Celso Mendes has been participating in the preparation of the *Second Annual LACSI Symposium*, to be held at Santa Fe in October 2001.

Several members of the Pablo Group are active in the NSF-funded GrADS project and have attended multiple workshops and teleconferences for that project where LACSI tools development is also discussed with collaborators at Rice, Tennessee, and Houston.

Software Distributions:

Between October 1, 1999 and September 30, 2001 new releases of the SvPablo, Autopilot, Autopilot Performance Monitor, and SDDF performance tools have been distributed via the Pablo group website.

Parallel Application Development Tools

Investigator: Barbara Chapman

Abstract: This activity is focused on the goal of creating an interactive software tool that supports parallel application development. This includes practical support for a number of the standard tasks involved when creating MPI or OpenMP code. It also entails novel approaches to generate optimized parallel code for SMPs from “loop-level” OpenMP by translating the OpenMP code to a thread library under a number of compiler optimizations, and by exploiting user information in the form of additional directives. We chose LANL’s SMARTS runtime system to be the target thread library for this work. This work is being done on top of the Cougar compiler, which has been under construction at the University of Houston during the past two years. It provided several graphical displays of an input FORTRAN 77 code, and standard analyses and optimizations.

Our work focuses on tool support for programming in OpenMP and OpenMP + MPI, with the emphasis placed on OpenMP. We discuss the first three items in the summary and consider the last two separately below.

Most compilers now perform very simple translations of OpenMP to calls to thread routines. This provides considerable transparency of performance at the cost of raw performance, since many potential optimizations are ignored. (Notable exceptions are the Japanese RWCP Omni compiler and the Rice OpenMP/TreadMarks system (See below.), which target software DSM systems for which optimization is critical for even moderate performance levels.) One way to achieve performance under these circumstances is to create a well-optimized code in standard OpenMP before using a vendor compiler; another way is to optimize the code and directly target an efficient thread library. Our work considers support for both of these options.

Translation of OpenMP to SMARTS

Our main thrust so far is based upon the use of SMARTS as a target system for the translation of OpenMP programs. SMARTS was originally used as a runtime library for POOMA, where its procedures were inserted into C++ code. Our initial work evaluated SMARTS itself, in order to develop an understanding of the library and its use, as well as to understand the potential benefits of this thread library. Given the care with which functionality was initially chosen, and the sophisticated scheduling options it incorporates, we were convinced that this would provide us with significant advantages over a standard thread package.

First, we examined the SMARTS functionality in order to create an API and a basic translation strategy for use with FORTRAN programs. The resulting interfaces permit invocation of SMARTS routines from within the modified FORTRAN code. This API, and the strategy developed for its use, provides for the initialization of SMARTS at run time, the passing of compile-time information to SMARTS and the passing of iterates, pieces of code that are statically or dynamically scheduled for execution as individual units, together with the list of data objects read and written by them. SMARTS requires that the compiler provide it with a collection of iterates and the complete set of data dependences between those iterates. During the report period, we have developed the compiler algorithms needed to generate this information.

Next, we evaluated the features of SMARTS in order to understand how it might help us provide superior performance. In addition to helping reduce barrier synchronization, it has two specific benefits: it facilitates out-of-order execution of parallel loops and it permits the exploitation of temporal locality through schedule re-use across parallel loops. Synchronization remains in a SMARTS program only in the form of data dependence constraints on the order of execution of individual iterates. Thus if a loop nest requiring a barrier in OpenMP is parallelized this way, the barrier is required only between those iterates involved in a dependence relationship; other loop iterations may proceed. Similarly, the data flow execution model realized by SMARTS permits out of order execution, so that in the above case, iterations of a successor loop might also run, thus further reducing synchronization costs.

As a shared-memory programming model, OpenMP does not provide features for user control of data locality. However, cache interference is one of the major performance problems and it is the cause of much of the complexity of programming in this paradigm. SMARTS scheduling is useful for load balancing, but if it is too eagerly applied, it may lead to performance degradations when cached data is not reused. We can apply SMARTS so-called hard affinity scheduling, where iterates are mapped to specific threads statically by utilizing the iterate dependence graph constructed in the compiler to determine a mapping of iterates to threads. Our experiments show that the best payoff on the SMP platforms tested was obtained when cache reuse was given a higher priority than load balancing when the mapping was performed. In our current system, we expect the user to provide help to create iterates; because we need to emphasize the locality of data, we expect the user to provide information on how program data is to be decomposed via the

HPF data distribution syntax. The data decomposition drives the compiler decomposition of loop nests into “chunks” that become the SMARTS iterates. Thus, the data locality determines the structure of the parallelism in our system.

A number of experiments were performed to understand the impact of the overheads introduced by SMARTS, as well as the impact of several scheduling policies it provides. As a result of the overheads incurred, these experiments show that OpenMP code and code translated to SMARTS can be expected to have about the same execution time on two processors; after that, SMARTS begins to outperform other implementations and exhibits better scalability. A comparison of level 1 cache misses shows that our strategy for translating code to SMARTS and its scheduling options results in substantially fewer misses.

A paper reporting on this work is under preparation.

Performance Evaluation of OpenMP, MPI and pThreads

We have performed several experiments to better understand the performance problems associated with OpenMP. Initial work focused on comparing multiple versions of OpenMP programs, particularly on cinema platforms. Our results showed a clear superiority of the SPMD programming style, in which a user applies an MPI-like strategy to give each thread local data and work. Data is shared between threads via the construction and copying of buffers. This work has been reported on at several conferences and is the subject of two publications.

Recent work has begun a more extensive comparison of OpenMP as a programming model with other current programming paradigms. In addition to programming effort, which is to some extent dependent upon the programmer, considerations are the potential for resource utilization, potential for scalability and the cost of basic operations. Ongoing work creates a benchmarking suite to provide measurement and analysis of the above, but without considering programmer effort.

The latter area is the result of on-going collaboration with CCS at LANL and led to a summer internship for Ache Prabhakar from our research group. We plan to continue to build the benchmarking suite and to experiment with it during the coming project year in collaboration with CCS.

Cougar Compiler and Application Migration to Component Systems

We have been working to improve the usability of Cougar for on-going work at LANL and have added features that either support this goal or support our goal of enabling this system to be used in future for creating components from existing applications. Many bug fixes have been performed and work has also aimed to remove aspects of the implementation that might have limited the size of input code. We were able to find pilot users in our Physics department and NASA Johnson to help in the testing of some of its features.

We added features for examining the use of common blocks throughout the code, for printing out graphs, and for retrieving information on the definition of data structures. The GUI was extensively tested and inconsistencies were removed. The system was extended to recognize MPI calls; it is now able to perform program analysis and display for FORTRAN 77 programs that have been parallelized under MPI.

Discussion with NASA engineers using CORBA to develop what are essentially component systems revealed that they have substantial difficulties handling the input and output of the existing sequential source code. We have therefore begun by providing a number of features for locating, selecting and displaying I/O statements and their context. For future work with component systems, the biggest problem is the lack of a standard to target. Discussions with members of the CCA Forum have led us to believe that such a standard is still a long way off.

The major weakness of this system currently is its focus on FORTRAN 77 only. We have begun to work on using the SGI Pro64 compiler front end rather than our own in order to remedy this deficiency. The SGI software is available in source code and includes FORTRAN 77, FORTRAN 95, C and C++ front ends, thus giving us a path to supporting multi-language code in future. This work is being done in coordination with the compiler group at Rice.

This work has resulted in one publication and the production of a User Guide for Cougar. The latter is expected to be used primarily in its online version. It includes an installation procedure. The planned deployment of the new version of Cougar, originally planned for Summer 2001, has been delayed by the disruption of our systems by the recent flooding at the University.

Publications

1. Chapman, O. Hernandez, A. Patil and A. Prabhakar, Program Development Environment for OpenMP on ccNUMA and NUMA Platforms. *Proc. Large Scale Scientific Computations 2001*, Sozopol, LNCS, Springer Verlag, 2001 (to appear)
2. Chapman, A. Patil and A. Prabhakar, Performance Oriented Programming for NUMA Architectures, *Proc. WOMPAT 2001, LNCS 2104*, Springer Verlag, 137-154, 2001
3. B. Chapman. F. Bregier, A. Patil and A. Prabhakar, Achieving Performance under OpenMP on ccNUMA and Software Distributed Shared Memory Systems, Special Issue of *Concurrency Practice and Experience*. (To appear 2001.)

Visits

Summer 2001: Summer graduate student internship of Achal Prabhakar at LANL.

September 4, 2001:Visit by Barbara Chapman to CCS-3 with a presentation on OpenMP and Tools.

OpenMP

Investigators: Alan Cox, Y. Charlie Hu, Willy Zwaenepoel,

Abstract: The objective of this project is to develop portable shared memory programming support that scales from small clusters of workstations and SMPs to ASC supercomputers. Our approach is to support OpenMP — an emerging shared memory programming API standard — directly on top of the TreadMarks software distributed shared memory (DSM) system, which runs on networks of workstations and PCs and SMPs.

Accomplishments

The milestones for years 2 and 3 of the contract were as follows: "(Year 2) Assess performance impact of different optimizations and identify key obstacles to the scalability of software DSM systems. (Years 3-4) Develop integrated compiler-time/run-time optimizations to support efficient OpenMP programs on medium-scale parallel machines, and improve on the scalability of software DSM systems to sustain high efficiency on large-scale systems".

These milestones have been met. In year 2, we have investigated the performance of various applications running on OpenMP on networks of SMPs. The results were reported in a paper in JPDC. We have also investigated limits to the scalability of the OpenMP/NOW system, and we have found that contention at the end of sequential sections of the code is one the main impediment towards scalability. This work was reported at LCR-2000. In year 3, we have developed a multicast-based solution to this problem, which was reported at PPoPP-2001. Many of these results are included in Honghui Lu's Ph.D. thesis, which she completed at Rice in February 2001.

Status

We have been working with Kuck and Associates (KAI), now a division of Intel, to incorporate some of the ideas described above in a commercial offering by KAI. This work is now in alpha production stage.

Honghui Lu completed her Ph.D. thesis in February 2001, and will join the University of Pennsylvania as an assistant professor in Fall 2001. Charlie Hu, who worked on this project as a research scientist at Rice, will join Purdue University as an assistant professor in Spring 2002.

Publications

H. Lu, A.L. Cox, and W. Zwaenepoel, Contention Elimination by Replicating Sequential Sections in Distributed Shared Memory Programs, *Proceedings of the Eighth Conference on Principles and Practice of Parallel Programming*, June 2001.

Honghui Lu, OpenMP on Network of Workstations, Ph.D. Thesis, Department of Electrical and Computer Engineering, Rice University, February 2001.

A.L. Cox, Y.C. Hu, H. Lu and W. Zwaenepoel, OpenMP on Networks of SMPs, *Journal of Parallel and Distributed Computation*, Vol.60, No. 12, December 2000

Y.C. Hu, A.L. Cox and W. Zwaenepoel, Improving Fine-Grained Irregular Shared-Memory Benchmarks by Data Reordering, *Proceedings Supercomputing 2000*, November 2000.

A. Scherer, T. Gross and W. Zwaenepoel, An Evaluation of Adaptive Execution of OpenMP Task Parallel Programs, *Proceedings of Languages, Compilers, and Runtimes for Scalable Computing*, May 2000.

E. de Lara, Y.C. Hu, H. Lu, A.L. Cox and W. Zwaenepoel, The Effect of Memory Contention on the Scalability of Page-based Software Distributed Shared Memory Systems, *Proceedings of Languages, Compilers, and Runtimes for Scalable Computing*, May 2000.

Y.C. Hu, W. Yu, D. Wallach, A.L. Cox, and W. Zwaenepoel, Run-time Support for Distributed Sharing in Typed Languages, *Proceedings of Languages, Compilers, and Runtimes for Scalable Computing*, May 2000.

Visits and Other Interactions

Alan Cox attended the annual LACSI meeting in Santa Fe in 1999 and 2000. Charlie Hu could not obtain timely approval to visit LANL in conjunction with the 2000 meeting.

B. Component Architectures for Rapid Application Development and Composition in a Networked Environment

The goal of this group of projects is to develop compiler technologies and library designs that will make it possible to automatically construct domain-specific critical components for high-performance applications. To that end, we are developing advanced compiler technology to construct high-level programming systems from domain-specific libraries. Using such systems, programs will use a high-level scripting language to coordinate invocation of library operations. This will enable an application to remain readable but still perform at peak levels in the critical components.

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. Our effort has focused on overcoming this problem using extensive preliminary compilation (perhaps taking many hours) of libraries to produce high-performance scripting systems. This approach is called *telescoping languages* because it permits the construction of new languages on top of older ones by rerunning the language-generation process.

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, a significant portion of the work has focused on the Java programming language. 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 are pursuing research in three important directions:

Toolkits for Building Problem-Solving Environments (PSEs) and Systems. The effort focuses 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.
- Demonstration of these techniques in specific applications of interest to ASC and LANL.

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

Compilation of Object-Oriented Languages. As mentioned above, high-performance compilation strategies must be developed for object-oriented languages such as Java and C++. This should include interprocedural techniques such as inlining driven by global type analysis and analysis of multithreaded applications. This work would also include new programming support tools for high-performance environments. Initially, this work will focus on Java, using the JaMake high-level Java transformation system developed at Rice. Later, we will consider extensions to other object-oriented languages. This effort will be undertaken as a collaboration with the CartaBlanca group at LANL.

Numerical Component Libraries for Integration. This effort is focusing 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.

Application Development Support for Grid Computing. This effort leverages the GrADS project at Rice University, which involves a consortium of 8 universities and 13 principal investigators focusing on the construction of high-level programming systems that will make the computational Grid usable by ordinary scientists and engineers. This effort involves developing programs that are adaptive in that they can adapt to different resource configurations and changing loads on resources during execution. A fundamental strategy is to develop systems that integrate grid-aware library components into distributed applications via a high-level scripting language.

Component Toolkits for Building Problem-Solving Systems

Investigators: Ken Kennedy, Bradley Broom, Keith Cooper, Jack Dongarra, Rob Fowler, Lennart Johnsson, John Mellor-Crummey, and Linda Torczon

Telescoping languages. To address the problem of building high-level programming systems, we are pursuing a new approach called *telescoping languages*, in which libraries that provide component operations accessible from scripts are extensively analyzed and optimized in advance. In this scheme, language implementation would consist of two phases. The offline translator generation phase would digest annotations describing the semantics of library routines and combine them with its own analysis to generate an optimized version of the library, and produce a language translator that understands library entry points as language primitives. The script compilation phase would invoke the generated compiler to produce an optimized base language program. The generated compiler would (1) propagate variable property information throughout the script, (2) use a high-level "peephole" optimizer based on library annotations to replace sequences of calls with faster sequences, and (3) select specialized implementations for each library call based on parameter properties at the point of call. The fundamental ideas behind telescoping languages have been elaborated in a paper published in the proceedings of IPDPS 2000 and in a forthcoming paper to appear in JPDC (see below).

MATLAB prototype for signal processing. A principal focus of the work to date has been the development of the telescoping languages concept in a prototype based on MATLAB for signal processing applications. We have initiated a collaboration with the signal-processing group at Rice to conduct research on domain-specific languages for communications. The motivation for this effort is the widespread use of MATLAB in the signal processing community, coupled with the need to rewrite these applications in conventional programming languages to achieve high performance on embedded processors. A goal of the effort is to make it possible to compile the signal processing programs directly to these embedded processors without the need for additional human programming effort.

This work has led to the development of several new optimizations. A surprising discovery was that many signal processing applications make extensive use of procedures called within loops in which the only varying parameter is the loop index. This gives rise to two new optimizations. *Procedure vectorization* interchanges the loop inside the procedure, making it possible to vectorize many of the operations within. *Procedure reduction in strength* divides the procedure into a part that varies and one that does not, moving the non-varying component out of the loop. Both of these optimizations will be extensively used in the system we are constructing, though they are applicable to a broad variety of high-level domain-specific languages. While much of the same effect can be achieved by "inlining and hoisting" techniques, the analysis that enables application of these transformations is done during the pre-compilation phase, thus reducing the time to compile the script.

Plans. We are building our own MATLAB infrastructure to support this research. We expect to have it ready within the next fiscal year. When it is done, we will experiment with applications of higher interest to the ASC program, such as a replacement for the POOMA system.

Publications

1. Ken Kennedy. Telescoping languages: A compiler strategy for implementation of high-level domain-specific programming systems. *Proceedings of the 14th International Parallel and Distributed Processing Symposium (IPDPS 2000),* Cancun, Mexico, (May 2000).
2. Arun Chauhan and Ken Kennedy. Reduction in strength of procedures: An optimizing strategy for telescoping languages. *Proceedings of the 2001 International Conference on Supercomputing*, Sorrento, Italy (June 2001).
3. Ken Kennedy, Bradley Broom, Keith Cooper, Jack Dongarra, Rob Fowler, Dennis Gannon, Lennart Johnsson, John Mellor-Crummey, and Linda Torczon. Telescoping

languages: A strategy for automatic generation of scientific problem-solving systems from annotated libraries. To appear in *Journal of Parallel and Distributed Computing*.

Compilation of Object-Oriented Languages

Investigators: Ken Kennedy and Zoran Budimlić

Java is the first widely accepted language that addresses heterogeneous resources, security, and portability problems, making it attractive for scientific computation. It also encourages programmers to use object-oriented techniques in programming. Unfortunately, such object-oriented programs also incur unacceptable performance penalties. For example, using a polymorphic class hierarchy for numbers in a linear algebra package resulted in a code that is four times shorter, more extensible, and less bug-prone than the equivalent FORTRAN-style code, but also many times slower.

To address the poor performance problem, we developed several new compilation techniques that can improve the performance of scientific Java programs written in a polymorphic, object-oriented style to within a factor of two of the equivalent hand-coded FORTRAN-style programs. These techniques also maintain an acceptable level of Java bytecode portability and flexibility, thus rewarding, rather than penalizing, good object-oriented programming practice.

These optimizations for the first time address several important aspects of scientific programming in Java, such as optimization of arrays, program extensibility, and portability. They overcome the restrictions posed to the source code compiler by the Java bytecode execution model and maintain full security and portability of the Java bytecode.

Class specialization clones a class containing polymorphic fields based on the possible subtypes of those fields. It generates several specialized versions of the original class, exposing the subtype distinctions in the revised class hierarchy. The specialized classes are monomorphic with respect to the selected fields, enabling subsequent optimizations such as object inlining. This transformation extends the classical notion of cloning, where a function is cloned based on the values or types of its parameters, by allowing a class to be cloned based on the exact type of the polymorphic data it contains. Class specialization results in only modest direct performance benefit; its main result is that it enables subsequent object inlining and downstream optimization of the inlined code.

Object inlining is a novel program transformation that converts Java objects into *inlined objects*, a new data representation for the objects, and transforms one program into another that operates on this new representation. This transformation eliminates the indirection in object accesses by "flattening" the data structures in the program. It enables method inlining on the inlined objects by eliminating the privacy restrictions of the original code. Moreover, it improves the memory hierarchy performance by increasing the locality of the data. It reduces the overhead of dynamic memory management by allocating some objects on the stack and reducing the total number of objects in the program. In addition, it reduces the memory footprint of the program by reducing the number of objects among which the program data is distributed. Finally, it creates more space for local compiler optimizations by performing these transformations. Object inlining can improve the performance of polymorphic, object-oriented, scientific Java programs by up to two orders of magnitude.

Static Single Assignment (SSA) form, the standard intermediate representation for programs in a compiler, enables efficient implementation of many classical code optimization techniques. The SSA versions of many traditional optimization algorithms are both more efficient and easier to implement than their non-SSA equivalents. An efficient SSA representation is needed to improve the implementation of classical optimizations. We have developed a novel algorithm for converting SSA form into a control flow graph (CFG). We have proved various properties of the SSA form and then used those properties to develop this fast algorithm. This algorithm has a much wider applicability than just in compiling of Java for high performance, since it improves the compiler technology in general. Preliminary experiments with our implementation show that it eliminates almost as many copies as the $O(n^2)$ interference-based algorithms, while achieving near-linear complexity.

Almost-whole-program compilation is a novel compilation strategy in which the compiler assumes a static class hierarchy at compile time and the programmer specifies the classes that would be publicly visible. This strategy uses Java visibility rules, novel implementation techniques, and novel class packaging techniques to allow for extensive program optimization in Java. These techniques allow both good design *and* good performance by dividing the compilation process into two fundamentally different processes: development and distribution.

In the development phase, it provides the programmers with freedom to develop flexible, extensible object-oriented programs, without worrying about the performance. This enables the programmers to concentrate on the overall design and extensibility of the program without forcing them to introduce performance-design trade-offs.

On the other hand, the distribution phase is completely controlled by the programmer, allowing a balance between the performance of the generated program and its flexibility from the end-user viewpoint. This way, the programmer can first decide on the acceptable level of flexibility that the end-user needs (or solicit this information from the end-user), and then transform the program to allow for maximum performance under those flexibility restrictions. The programmer can completely restrict the flexibility of the produced program and thus enable full whole-program optimizations on one extreme, completely restrict the whole-program optimizations and thus enable full program flexibility to the end-user on the other extreme, and fine-tune the program to anywhere in between these two extremes in general. Regardless of the choice, the code that almost-whole-program compilation produces is always fully portable and verifiable Java bytecode.

Our implementation of the almost-whole-program framework enables us to pursue further research in type analysis for incomplete programs that will make possible creation of almost-whole programs with almost whole-program performance.

Publications:

1. Zoran Budimlić and Ken Kennedy. Prospects for scientific computing in polymorphic, object-oriented style. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, Texas, March 1999.
2. Zoran Budimlić, Ken Kennedy, and Jeff Piper. The cost of being object-oriented: A preliminary study. *Scientific Programming*, 7(2):87--95, 1999.

3. Zoran Budimlić. *Compiling Java for High Performance and the Internet*. Ph.D. thesis, Rice University, 2001.

4. Zoran Budimlić and Ken Kennedy. JaMake: A Java compiler environment. *Scientific Computing*}, 2001.

5. Zoran Budimlić and Ken Kennedy. Almost-whole-program compilation. In *Proceedings of the Third Workshop on Java for High Performance Computing*, Sorrento, Italy, June 2001.

6. Zoran Budimlić, Keith Cooper, Tim Harvey, Ken Kennedy, Tim Oberg and Steve Reeves. Fast SSA Reconstruction. To be submitted for PLDA 2002.

Visits

June 20-22. Ken Kennedy and Richard Hanson visited LANL to discuss research directions involving component architectures and advanced numerical algorithms.

We are waiting for approval for Zoran Budimlić to visit LANL to meet with W. Brian VanderHeyden.

Numerical Component Libraries for Integration

Investigators: Lennart Johnsson, Jack Dongarra, Roland Glowinski, Richard Hanson, Yuri Kuznetsov

The principal goal of this effort is the development of efficient library components for a broad spectrum of computational platforms and applications. The libraries should adapt easily to a broad range of architectures and configurations used in high performance applications, as well as to a wide range of application characteristics. To achieve this goal a novel approach to software library design and implementation has been pursued. The approach has emerged over the last few years, and is a major departure from how software libraries traditionally have been designed and implemented. Maintaining libraries with many versions is notoriously difficult to get right, so managing a single software version is the usual choice. Our main effort so far has been devoted to the construction of tools and infrastructure for automatic code generation and optimization. The efficiency of this approach has already been demonstrated in the case of the UHFFT library.

Fast Fourier Transforms.

We have developed a library that adapts automatically to the hardware it is running on by using a dynamic construction of the FFT algorithm. There is a comprehensive understanding of this problem, and there are high standards for its computational performance. The construction of the implementation is done through the selection of the fastest combination of component blocks of code called codelets, which are generated and optimized for the underlying architecture during the installation of the library. We have also developed an efficient automatic method of generating and optimizing the library modules by using a special-purpose compiler. The code generator is written in C and it generates a library of C codelets. The code generator is shown to be flexible and extensible and the entire library can be (re) generated in a matter of seconds taking into account a new set of optimization rules and user requirements. The code generator infrastructure developed for the FFT library was built general enough so that it can be easily extended to other well-structured problems. We have recently written a first set of test cases for the spherical transforms and finite element method.

The major novelty in our FFT library is that most of the code is automatically generated during the installation of the library with an attempt to tune the installation to the particular architecture. To our knowledge, this is the only FFT library with such capability. Although several other public domain libraries (such as FFTW from MIT) make use of automatic code generation techniques similar to ours, their code is usually pre-generated and fixed for all platforms. Even if they allow for possible modifications of the generated code, these modifications are cumbersome and not at all automatic. Our code is generated and optimized at the time of installation. We also have a small number of installation options that can be specified by the user. At present, these options are restricted to the range of sizes and dimensions for which the library should be optimized. We are planning to extend the range of options to include the interface, data distribution and parallelization methods in the next release. We may also include some application-specific options like known symmetries in the data, restrictions on the size of the library and the memory used by the code. The extent of the additional options will strongly depend on the feedback we get from the users. The idea is to exploit the flexibility of code generation and optimization tools that are built in the library for the benefit of the user and to allow for a significant and simple customization of the library.

The version 1.0 of the UHFFT library has been released for testing to groups helping us in code development and optimization. The full release to application-oriented groups is scheduled for September '01. This version of the library will support MPI applications in C and FORTRAN and have well-tuned code for several processors, including Intel Pentiums, Xeons, MIPS, IBM Power3, HP PA-RISC, SUN, etc. Tuning of the library for clusters and various forms of interconnection technologies will be carried out for the second release planned for the first quarter of 2002. Both complex and real transforms of arbitrary size and dimension are included with some minor restrictions on the parallel implementation.

Project URL: http://www.cs.uh.edu/~mirkovic/fft/parfft.htm

Portable, Adaptive Libraries for Linear Algebra

For the ATLAS adaptive linear algebra package, we produced a developer release (ATLAS 3.3) that possesses support for SSE2, allowing for maximal DGEMM performance of around 2Gflop on a 1.5Ghz Intel P4. Using SSE1 provides a roughly 4Gflop peak SGEMM on the same machine. Prebuilt archives are available for many architectures used widely within the DOE community, including well-tested version of the developer release. In particular, SSE2-enabled P4 libraries are available for both Linux and Windows.

In order to test iterative solvers for sparse linear systems, we built a test generator system. This work included implementing of some prototypical iterative methods, the writing of the test matrix generator, writing shell scripts for post-processing results, and instrumentation of the entire package using the PAPI portable interface to hardware performance counter instrumentation.

We have begun work on the extension of ATLAS to use iterative methods. In particular, this extension will embody several variants of conjugate gradient solvers and will make an adaptive choice among them based on the performance characteristics of the hardware on which it is run. Currently, some of the code has been written. We have rewritten some of the PETSc primitives into the required form. Thus far, we have tested the code on a small cluster at the University of Tennessee and we will soon begin testing on a large cluster at ORNL before attempting deployment at LANL.

In order to produce MPI applications that perform well on today's parallel architectures, programmers need effective tools for collecting and analyzing performance data. Because programmers typically work on more than one platform, cross-platform tools are highly desirable. A variety of such tools, both commercial and research, are becoming available. Because the tools use different approaches and provide different features, the user may be confused as to what tool will best suit his needs. We produced a review of the available cross-platform MPI performance analysis tools and evaluate them according to a set of criteria that includes robustness, usability, scalability, portability, and versatility. Issues pertaining to hybrid and heterogeneous, distributed programming environments and to MPI I/O performance analysis are also discussed.

Publications:

1. Jack Dongarra, Victor Eijkhout, and Henk van der Vorst, An Iterative Solver Benchmark, *LAPACK working note 152*, Department of Computer Science, University of Tennessee, (to appear in Scientific Programming), 2001.

2. Jack Dongarra, Victor Eijkhout, and Piotr Luszczek, Recursive approach in sparse matrix LU factorization, *Proceedings of the 1st SGI Users Conference*, ACC Cyfronet UMM, Cracow, Poland, October 2000, 409-418.

3. Mirkovic, D., Mahasoom R., Johnsson S.L. (2000) An Adaptive Software Library for Fast Fourier Transforms. *Proceedings of the 2000 International Conference on Supercomputing*, Santa Fe, NM, pp. 215-224, 2000.

4. Mirkovic, D., Johnsson S.L. (2001) Automatic Performance Tuning in UHFFT Library. *Proceedings of the 2001 International Conference on Computational Science, (ICCS 2001)*, May 2001, San Francisco, USA, Lecture Notes in Computer Science 2073, Vol. 1, pp. 71-80.

5. S. Moore, J. Dongarra, D. Cronk, and K. London. Review of MPI Performance Analysis Tools, *EuroPVM/MPI 2001*, Sept. 2001 (to appear).

Application Development Support for Distributed Computing

Investigators: Ken Kennedy, Keith Cooper, Lennart Johnsson, John Mellor-Crummey, Jaspal Subhlok, Linda Torczon

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

Compilation of Configurable Object Programs.

This research, which leverages the NSF funding for the GrADS (Grid Application Development Software) Project is exploring the challenges in compiling programs to be dynamically configurable to resources available on the Grid and to be reconfigurable in response to changing loads. It will explore programming models and their translation to efficient collections of tasks that can be targeted to a variety of Grid computing platforms. The long-term goal is to support component-based implementation of applications for grids using PSEs of the sort described in the toolkits effort described above.

A critical technology will be adaptivity to changing performance characteristics of the components of a distributed computing platform. This adaptivity will require reoptimization and migration of components and data. To date, the GrADS project has developed a framework for execution of Grid programs that supports this kind of adaptivity. It is described in the paper by Berman, *et al.* (See below.) We have also produced a detailed definition of the program execution process in the GrADS framework in a technical report available on the GrADS web site: (http://hipersoft.cs.rice.edu/grads/)

A significant component of this work is the development of performance models and mapping strategies that can be used to automatically retarget applications to different parallel and distributed computing platforms. The efforts to date have defined preliminary strategies based on trial executions for performance modeling of composite applications.

Another effort has been to develop fundamental communication primitives that can be used as building blocks to construct Grid applications. Broadcast is an operation that is frequently used but difficult to implement optimally in a Grid environment. Recently researchers on our project have proved the problem to be NP-complete and developed a fast and effective heuristic to plan Grid broadcast operations. Mandal, Kennedy, and Mellor-Crummey will publish this work in a report.

Because this effort receives principal funding from the NSF, the efforts within LACSI are focused on finding suitable LANL applications for implementation in a Grid environment. This work is only now getting started.

Frameworks for Building Adaptive Network Applications.

This project is performing fundamental research and developing middleware for “optimal” resource selection in shared distributed environments. The key challenge is predicting the performance of an application on a set of nodes on a shared network with dynamically changing resource availability. The planned framework for resource selection will have the following main components:

Development of the notion of a performance signature, which represents application resource requirements and predicts application performance under different network conditions, such as different speed links and processors and different competing loads and traffic.

Development of a resource graph representation for the topology and status of currently available resources such as CPU and memory at nodes and latency and bandwidth on network links.

Development of algorithms for graph based node selection, which map an application represented by its performance signature to the network represented by a resource graph.

To date, we have developed a framework for invoking remote MPI code from a CORBA distributed program development environment and analyzed the overheads involved. The goal of this subproject is to make high performance numerical codes easily accessible to programmers using CORBA without additional programming. Experiments were conducted with a compute server at Rice and a CORBA environment at University of Houston. The general conclusion is that the overheads are small enough that such remote invocation is profitable for medium to large problems and the process can be made largely transparent to the user by using utilities for automatic generation of "glue" to connect the two environments.

The broad goal of this project is to build adaptive network applications by providing support for scheduling and resource management on computation clusters and distributed grid environments. This phase of the research focused on measuring the resources used by applications during execution to infer their performance on shared networks. Since the resource needs of different applications are very different, this is an important component of resource selection on busy networks. The major research achievements were as follows:

- We built a framework to measure system activity composed of processor utilization and traffic on network links for parallel and distributed applications.
- We experimented with NAS benchmarks to demonstrate that the basic computation and communication structure of these programs can be accurately inferred by system level measurements.
- We demonstrated with NAS benchmarks that analysis of resource measurements on a dedicated testbed can provide parameters of a model to estimate performance when the network is shared.
- We developed a tool set to infer program level communication activity, that is, message exchange sequence, from measurements on the network.
- We developed a framework to select nodes for execution using NWS (Network Weather Service) to measure and predict CPU and bandwidth availability.

Visits:

May 2001: Jaspal Subhlok visited the University of New Mexico on May 6-8 to give a talk titled "Automatic Resource Selection on Shared Networks" and discuss research collaboration in this area.

Research in this area was also a topic of discussion during visits listed in other sections of this report.

Publications

1. Francine Berman, Andrew Chien, Keith Cooper, Jack Dongarra, Ian Foster, Dennis Gannon, Lennart Johnsson, Ken Kennedy, Carl Kesselman, John Mellor-Crummey, Dan Reed, Linda Torczon, and Rich Wolski. The GrADS Project: Software support for high-level grid application development. To appear in *International Journal of Supercomputer Applications*.
2. Yu-Ren Chung. Web mirror selection by extraction of long-term throughput from short HTTP transfers. Master's thesis, Dept. of Computer Science, University of Houston, May 2001.
3. Amit Lodh. Reuse of high performance MPI programs using CORBA interface. Master's thesis, Dept. of Computer Science, University of Houston, May 2001.
4. J. Subhlok, Shreenivasa Venkataramaiah, Amitoj Singh and Shrikanth Goteti. Characterizing application performance on shared networks based on resource usage. *Poster for presentation at the LACSI Symposium 2001*.
5. J. Subhlok and P. Steenkiste. Airshed pollution modeling in an HPF style environment. *Journal of Parallel and Distributed Computing*, 60(6):690--715, June 2000.
6. J. Subhlok and G. Vondran. Optimal use of mixed task and data parallelism for pipelined computations. *Journal of Parallel and Distributed Computing,* 60(3):295–319, March 2000.

C. Computational Mathematics

Large-Scale Nonlinear Optimization Algorithms and Software

Investigators: Yin Zhang, Richard Tapia

So far, three research projects headed by the PI's have been supported in part by the LACSI funds. We outline progress achieved in the three projects below.

Computational Biology: Dr. Zhijun Wu received 50% support from our LACSI project. He worked on computational methods for X-Ray Crystallography, in collaboration with a biochemist George Phillips, under the supervision of the PI's. The main results of this project are contained in two papers (see Publications), where a fast Newton method is devised for a maximum entropy problem arising from a promising approach in X-Ray Crystallography.

Algorithms for large-scale semidefinite programs: This is a joint project between Zhang and his collaborators at Georgia Institute of Technology. Semidefinite programming has been one of the most active research areas in optimization with its many newly found applications. However, the capacity of solving large-scale semidefinite programs (SDPs) has been severely limited by the extraordinarily high demand on computational resources. We have proposed novel algorithms for solving large-scale SDPs and developed an experimental software package called BMZ. (See the list below.) Theoretical and computational results are summarized in two recent papers. (See publications – TR01-11 and TR99-27 in the Department of Applied Mathematics at Rice University.)

Continuous optimization techniques for discrete problems: This again is a collaboration between Zhang and his co-workers at Georgia Institute of Technology, in which we study continuous optimization techniques for solving graph-partitioning problems. Two prototype software packages have been developed for solving several types of graph-partitioning problems (see the list below). Theoretical and computational results are reported in two recent papers (see publications – TR00-33 and TR00-34 in the Department of Applied Mathematics at Rice University).

These three software packages developed under partial support from the LACSI project.

> CirCut: A fast and scalable implementation of Goemans-Williamson quality heuristics for the maximum cut problem, maximum bisection problem, and other graph partitioning problems, code written in FORTRAN 90. http://www.caam.rice.edu/~zhang/circut
>
> Max-AO: High-quality heuristics for the maximum stable set, maximum clique, and related problems based on continuous optimization formulations, written in ANSI C. http://www.math.gatech.edu/~burer/Max-AO
>
> BMZ: An algorithm for solving a class of large-scale semidefinite programs via nonlinear transformation and a first-order log-barrier algorithm, written in ANSI C. To be released soon.

Publications

1. Sam Burer, Renato Monteiro and Yin Zhang. A Computational Study of a Gradient-Based Log-Barrier Algorithm for a Class of Large-Scale SDPs, Technical Report TR01-11, Department of Computational and Applied Mathematics, Rice University, Houston, Texas. 2001. Submitted to *Mathematical Programming*, 2001.
2. Sam Burer, Renato Monteiro and Yin Zhang. Interior-Point Algorithms for Semidefinite Programming Based on a Nonlinear Programming Formulation, Technical Report TR99-27, Department of Computational and Applied Mathematics, Rice University, Houston, Texas. 1999. To appear in *Computational Optimization and Applications*, 2000.
3. Sam Burer, Renato Monteiro and Yin Zhang. Rank-Two Relaxation Heuristics for Max-Cut and Other Binary Quadratic Programs. Technical Report TR00-33, Department of Computational and Applied Mathematics, Rice University, Houston, Texas. Submitted to *SIAM Journal on Optimization*, 2001.
4. Sam Burer, Renato Monteiro and Yin Zhang. Maximum Stable Set Formulations and Heuristics Based on Continuous Optimization. Technical Report TR00-34, Department of Computational and Applied Mathematics, Rice University, Houston, Texas. Submitted to *Mathematical Programming*, 2001.
5. Zhijun Wu, George Phillips, Richard Tapia and Yin Zhang. A Fast Newton's Method for Entropy Maximization in Phase Determination. To appear in *SIAM Review*. 2001.

6. Zhijun Wu, George Phillips, Richard Tapia and Yin Zhang. A Fast Newton's Method for Entropy Maximization in Statistical Phase Estimation. To appear in *ACTA Crystallographica*.

Parallel Tools for the Analysis and Optimization of Linked Subsystems

Investigators: Matthias Heinkenschloss, John Dennis

Abstract: We are developing methods and parallel tools for the coupling of subsystem simulations that are expensive to evaluate and the optimization of the resulting coupled system.

We refined our mathematical framework for the coupling of systems arising in the time-decomposition of optimal control and design problems. The basic ideas of this approach and preliminary numerical results for linear quadratic problems are described in (Heinkenschloss, 2000). Implementation and testing of above the framework for control of turbulent flows is in progress.

We explored solution approaches for the optimal control of aero-acoustic noise. This model problem involves a linked subsystem consisting of the compressible Navier-Stokes or Euler equations for the near field and a wave equation for the far field. Two recent publications by Collis *et al*. report results for the optimal control of the near-field problem. Work on the coupling is in progress.

We deployed the filter approach to constrained optimization to the Rice - Boeing collaboration on engineering design. We hope to extend this to a collaboration with the LANL work on simulation predictability. (The connection can be seen as LANL studying ways to increase the predictability of simulations and Rice as studying ways to incorporate that particular class of simulations into decision-making in a mathematically rigorous and computationally effective way.) More detail is given in four recent papers. (See the Audet *et al*. papers.)

Software development

The current implementation, FOCUS, of the general pattern search algorithm is available through the LACSI Computing Resources Web site (http://lacsi.lanl.gov/resources). A new implementation of NOMAD is currently under in its second development cycle http://www.crt.umontreal.ca~couturgi/NOMAD); it will include several algorithmic extensions and refinements to be made available though this site. Boeing has incorporated the algorithms involved in its Design Explorer toolkit. It is now the production tool for Boeing planform design projects. More detail is given in the paper ""A Surrogate-Model-Based Method for Constrained Optimization" in the AIAA MDO Conference Proceedings, September 2000. (See publication list.)

Publications:

1. Audet C., Dennis J.E.Jr. (2000), On the convergence of pattern search algorithms with mixed variables, *SIAM Journal on Optimization*, Vol.11 No.3, 573--594.

2. Audet C., Dennis J.E.Jr. (2000), A Pattern Search Filter Method for Nonlinear Programming without Derivatives, TR00-09, Department of Computational and Applied Mathematics, Rice University, Houston Texas.

3. Audet C., Dennis J.E.Jr, (2000), Analysis of Generalized Pattern Searches, TR99-07, Department of Computational and Applied Mathematics, Rice University, Houston Texas.

4. Audet C., Booker A.J., Frank P.D. and Moore D.M., (2000), A Surrogate-Model-Based Method for Constrained Optimization, *AIAA MDO Conference Proceedings*, Long Beach. September 2000

5. S.S. Collis, K.Ghayour, M.Heinkenschloss, M.Ulbrich, and S.Ulbrich, Optimal control of unsteady compressible viscous flow", Technical report, Department of Computational and Applied Mathematics, Rice University, 2001. (Submitted for publication.)

6. S.S. Collis, K.Ghayour, M.Heinkenschloss, M.Ulbrich, and S. Ulbrich, Towards adjoint-based methods for aeroacoustic control, In *39th Aerospace Science Meeting & Exhibit*, January 8--11, 2001, Reno, Nevada, AIAA Paper 2001--0821, 2001.

7. M. Heinkenschloss, Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problems, Technical Report TR00-31, Department of Computational and Applied Mathematics, 2000.

Visits and Interactions

Matthias Heinkenschloss visited LANL in August 2001 to explore collaborations with Mac Hyman of the Mathematical Modeling and Analysis Group, T-7.

The next visit involving both investigators is planned for the October 2001 in conjunction with the LACSI Symposium.

Code-Based Sensitivity Analysis

Investigator: Mike Fagan

Abstract: The main goal of the Code-Based Sensitivity Project is to produce both mathematical techniques and usable software tools for augmenting computer models with sensitivity calculations. Our main vehicle for disseminating our knowledge is the ADIFOR (Automatic Differentiation of FORTRAN) suite of tools. The latest version is ADIFOR 3.0. A description of the features and techniques for ADIFOR 3.0 appear in CAAM-TR-00-02.

To summarize, progress in the Code-Based Sensitivity Analysis Project has been in four areas:

1. Improved FORTRAN 77 Differentiation
2. Upgraded MPI coverage for Parallel Sensitivity Computation
3. New Sensitivity Computation Algorithms
4. Generalizing the ADIFOR Tool to Other Programming Languages

Improved FORTRAN 77 Differentiation

We have improved the effectiveness of the ADIFOR tool on two classes of FORTRAN 77 programs:

Non-holomorphic complex-valued programs: Some programs use the intrinsic complex arithmetic of FORTRAN 77 in ways that are non-differentiable when viewed as complex-valued functions. In other words, the program is **non-holomorphic**. The program **is** differentiable (holomorphic), however, when the complex values are viewed as pairs of real numbers. ADIFOR can now be advised to treat complex variables as a pair of Reals, thereby computing correct sensitivities for non-holomorphic (but still R2 differentiable) functions. As a side benefit, a user could **verify** that a given complex-valued program is holomorphic by using ADIFOR to compute R2 derivatives and then verifying the Cauchy conditions by inspection. See CAAM-TR00-04 (ADIFOR 3.0 working note 3) for details.

Nonstandard F77 programs that use pointers and polymorphic addresses: Some F77 programs make use of the F77 pass-by-address subprogram parameter mechanism to create polymorphic subroutines. While this use of the mechanism is technically forbidden by the F77 standard, no compilers we have used actually enforce this standard, even though ADIFOR does. We have, however, developed a tool set for correcting the polymorphic type mismatches introduced by nonstandard pass-by-address. Currently, this tool set must be applied as a separate phase before the main ADIFOR processing. Future versions of ADIFOR, however, will apply this processing invisibly to the user. The technique embodied in the tool will also be used more general versions of ADIFOR that supply sensitivity calculations for programming languages, such as FORTRAN90, that include pointers.

Details of this tool set and techniques will be detailed in the forthcoming ADIFOR 3.0 working note 4.

Upgraded MPI Coverage

Prior to year 2000, ADIFOR supported a few MPI operations: `send, recv, wait, isend, irecv, barrier,` and `bcast`. During 2000-2001, we added reverse mode and 2nd order forward mode coverage for all these operations. In addition, we added full support for `waitall, waitsome`, and the system-defined reduction operations.

Furthermore, preliminary support for user-defined reduction operations and user-defined data types is currently being added. Details of the MPI effort are covered in conference paper "Automatically Differentiating MPI-1: The Complete Story". Recent improvements in the treatment user-defined reductions will appear in the forthcoming "Forward and Reverse Mode Differentiation Techniques for User-Defined Reduction Operations".

New Algorithms for Sensitivity Analysis

We have recently worked out a forward mode method of computing derivatives for iterative procedures that incorporates a restart feature. The efficiency of this technique is comparable to restarted finite differences. Details of the new method will appear in the forthcoming "Projection-based Derivatives for Restartable Iterative Procedures".

Generalization to Other Programming Languages

In order to make Code-Based Sensitivity techniques more accessible to the scientific community, we are embarking on a generalization program to ensure that users of languages other than FORTRAN 77 will be able to apply our techniques. As part of that program, we are designing an XML-based intermediate language specific to the needs of a general Automatic Differentiation engine. We are calling this intermediate language XAIF.

Design is still ongoing, but a preliminary description appears in "A Component-based Software Architecture for Semantic Transformation of Scientific Software". There is a preprint, but the paper and the design are still in preparation.

Publications

1. A. Carle and M. Fagan, Automatic Differentiation for Non-holomorphic Complex-Valued FORTRAN Functions, *ADIFOR 3.0 Working Note #3*, CAAM-TR00-04
2. A. Carle and M. Fagan, Automatic Differentiation of FORTRAN Intrinsic Functions and Operators, *ADIFOR 3.0 Working Note #2*, CAAM-TR00-03, (September, 2000).
3. A. Carle and M. Fagan, ADIFOR 3.0 Overview, Rice Technical Report CAAM-TR-00-02, February, 2000
4. Carle and M. Fagan, "Automatically Differentiating MPI-1: The Complete Story", to be published in *Proceedings of the 3rd International Conference/Workshop on Automatic Differentiation*, June 2000. Proceedings to be published by Springer-Verlag (December, 2001).

(In preparation)

5. M. Fagan. Projection-based Derivatives for Restartable Iterative Procedures.
6. J. Abate, M. Fagan, P. Hovland, B. Norris, and L. Roh, A Component-based Software Architecture for Semantic Transformation of Scientific Software.
7. M. Fagan and P. Hovland. Forward and Reverse Mode Differentiation Techniques for User-Defined Reduction Operations.

Visits and Other Interactions

August 2000: M. Fagan attended the LACSI Symposium.

July 23-26, 2001: M. Fagan visited LANL CCS-2 to present a talk and to discuss research directions. Hosted by D. Kothe and R. Henniger.

Eigenvalue Methods and Software for ASC-MPP systems

Investigators: Danny Sorensen

Progress has been made in two general areas of this project. The first of these is the development of an approximate spectral transformation suitable for eigenvalue calculations.

The second area concerns the development of numerical methods for constructing reduced models for linear time invariant (LTI) control systems through low rank approximation of certain system Grammians. Large systems of this form arise in circuit simulation; they also arise through spatial discretization of certain time-dependent PDE control systems (such as parabolic equations subject to boundary control).

Our (Sorensen and Rice graduate students H. Thornquist and Y. Zhou) work is devoted to approaches to model reduction that have a high potential for addressing some fundamental difficulties with existing dimension reduction techniques. These issues are central to the potential development of robust and widely applicable software. We are striving to develop a computational methodology that will:

- provide rigorous bounds on the error in the response of the reduced system;
- naturally preserve fundamental system properties such as stability;
- and be fully automatic once a desired error tolerance is specified.

We are investigating subspace projection methods of Krylov and non-Krylov type. Our primary goal has been to develop implicit restarting methods that will iteratively produce approximations of specified rank k to controllability-, observability-, and cross- Grammians that are best approximations of rank k to the full Grammians. We are also investigating other techniques such as cyclic low-rank Smith methods.

The methods we are developing produce low rank approximate solutions to these system Grammians in factored form. From these factors, approximate balancing transformations are obtained. These balancing transformations amount to projection methods that produce reduced order models and a reduced basis set in which states that are simultaneously difficult to reach and observe are eliminated to retain only the dominant features of the system.

A Pre-conditioning Scheme for Large Eigenvalue Problems.

Preliminary work completed includes updating ARPACK and P_ARPACK codes to be compatible with the latest version of LAPACK. We have also completed preliminary work on the development an approximate shift-invert spectral transformation that will accelerate convergence of eigenvalue calculations without requiring a sparse-direct factorization and without requiring very accurate iterative solves. This will enable the solution of much larger problems in a matrix-free manner.

We have developed a new matrix-free algorithm for computing the smallest eigenvalues of a self-adjoint operator. This algorithm first constructs a fixed polynomial operator using pre-conditioned GMRES to approximate the shift-invert operator. This operator is then used to with the Implicitly Restarted Arnoldi Method (IRAM) to achieve the effect of shift-invert without factoring a matrix. Our numerical results indicate that it is just as accurate as IRAM in regular mode but uses far fewer matrix-vector products. As the problem size increases, this algorithm proves to be far less expensive. Moreover, it is also more accurate and considerably less expensive than solving the shift-invert equation using pre-conditioned GMRES started anew for each new right-hand side. In the future, we hope to improve the fixed polynomial operator to enable computation of clustered eigenvalues in any user-specified portion of the spectrum. This work will be presented at the LACSI symposium Oct. 2001.

We eventually plan to apply this procedure within P_ARPACK to find the most unstable modes of an ocean model. We intend to work with Balu Nadiga (LANL) to apply this to stability and bifurcation analysis of both idealized ocean models and full ocean models. The ultimate goal currently lies with the full ocean model. We hope to find a parameter regime where the model goes to a steady state, and then look into the stability of that state and get the most unstable eigenmodes.

Foundations of reduced basis methods:

The general theme of this research involves dimension reduction in the representation of trajectories of dynamical systems. The following work investigates theoretical developments in dimension reduction for linear time invariant dynamical systems. The concept of principal component analysis is fundamental to our reduced basis methods. This amounts to computing a reduced basis set from the eigensystem of the Grammian of the trajectory. In linear time invariant systems, this is related to various projection methods for a technique called balanced model reduction. We have made a considerable advance in the understanding of decay rates for Hankel singular values and for eigenvalues of the solutions to Lyapunov equations. These results have direct bearing on the ability to achieve very low dimensional balanced model reduction. We also have developed some fundamental relations between Lyapunov equations and the Lanczos process.

Fundamental relations between the Lanczos process and the Lyapunov inertia theorem. We (Antoulas and Sorensen) present a new proof of the inertia result associated with Lyapunov equations. Furthermore, we present a connection between the Lyapunov equation and the Lanczos process, which is closely related to the Schwarz form of a matrix. We provide a method for reducing a general matrix to Schwarz form in a finite number of steps (O(n3)). Hence, we provide a finite method for computing inertia without computing eigenvalues. This scheme is unstable numerically and hence is primarily of theoretical interest.

Theoretical justification of low dimensional approximations. This work (Antoulas, Sorensen and Zhou) investigates the decay rate of the Hankel singular values of linear dynamical systems. This issue is of considerable interest in model reduction by means of balanced truncation, for instance, since the sum of the neglected singular values provides an upper bound for an appropriate norm of the approximation error. The decay rate involves a new set of invariants associated with a linear system, which are obtained by evaluating a modified transfer function at the poles of the system. These considerations are equivalent to studying the decay rate of the eigenvalues of the product of the solutions of two Lyapunov equations. The related problem of determining the decay rate of the eigenvalues of the solution to one Lyapunov equation is also be addressed. Understanding the nature of these decay rates provides a foundation for the expectation of accurate low dimensional representations of large systems.

Approximation of large-scale dynamical systems. This work (Antoulas and Sorensen, Feb 2001) reviews the state of affairs in the area of approximation of large-scale systems. We distinguish among three basic categories, namely the SVD-based, the Krylov-based and the SVD-Krylov-based approximation methods. The first two were developed independently of each other and have distinct sets of attributes and drawbacks. The third approach seeks to combine the best attributes of the first two.

Projection methods for model reduction

We have developed several new algorithms based upon subspace projection for model reduction. Our methods produce low rank approximate solutions to system Grammians in factored form. From these factors, approximate balancing transformations are obtained. These balancing transformations amount to projection methods that produce reduced order models and a reduced basis set in which states that are simultaneously difficult to reach and observe are eliminated to retain only the dominant features of the system.

A modified low-rank Smith method for large-scale Lyapunov equations. In this note (Antoulas et al. 2001) we present a modified cyclic low-rank Smith method to compute low-rank approximations to solutions of Lyapunov equations arising from large-scale dynamical systems. Unlike the original cyclic low-rank Smith method introduced by Penzl, the number of the columns in the approximate solutions does not necessarily increase at each step. The number of columns required by the modified method is usually much lower than the original cyclic low-rank Smith method. The modified method never requires more columns than the original. Upper bounds are established for the errors in the low-rank approximate solutions and also for the errors in the resulting approximate Hankel singular values. Numerical results are given to verify the efficiency and accuracy of the new algorithm.

Low dimensional Lanczos-Gauss quadrature method for computing the heat capacity of a large-scale molecular system. This work (Yang *et al.)* develops an efficient algorithm for computing the heat capacity of a large-scale molecular system. The new algorithm is based on a special Gaussian quadrature whose abscissas and weights are obtained by a simple Lanczos iteration. Our numerical results have indicated that this new computational scheme is quite accurate. We have also shown that this method is at least a hundred times faster than the earlier approach that is based on estimating the density of states and integrating with a simple quadrature formula.

Projection methods for balanced model reduction. The work (Sorensen and Antoulas) investigates projection methods for the iterative computation of partially balanced reduced order systems. This approach is completely automatic once an error tolerance is specified and yields an error bound. Our approach is based on the computation and approximation of certain system Grammians and in particular on the cross Grammian.

Comparison of various computational methods for model reduction. An overview of model reduction methods and a comparison of the resulting algorithms are presented (Antoulas *et al.* 2000). These approaches are divided into two broad categories, namely SVD based and moment matching based methods. It turns out that the approximation error in the former case behaves better globally in frequency while in the latter case the local behavior is better.

Publications

1. A.C. Antoulas and D.C. Sorensen, Lyapunov, Lanczos, and Inertia, *Linear Algebra and its Applications*, 326, 137-150, 2001.
2. A.C. Antoulas and D. C. Sorensen, Approximation of large-scale dynamical systems: An overview, CAAM TR01-01, Feb. 2001.
3. A.C. Antoulas, D. C. Sorensen, and S. Gugercin, A survey of model reduction methods for large-scale systems, CAAM TR00-38, December 2000.
4. A.C. Antoulas, D. C. Sorensen, and S. Gugercin, A modified low-rank Smith method for large-scale Lyapunov equations, CAAM TR01-10, May 2001.
5. A.C. Antoulas, D. C. Sorensen, and Y. Zhou, On the decay rate of Hankel singular values and related issues, CAAM TR01-09, May 2001.
6. D. C. Sorensen and A. C. Antoulas, Projection Methods for Balanced Model Reduction, CAAM TR01-03, March 2001.
7. C. Yang, D. W. Noid, B. G. Sumpter, D. C. Sorensen, and R. E. Tuzun, An Efficient Algorithm for Calculating the Heat Capacity of a Large-scale Molecular System, CAAM TR01-02, February 2001.

All CAAM technical reports are available at
http://www.caam.rice.edu/caam/caam-techrep.html

Presentations by D.C. Sorensen

"Tutorial on Reduced Basis Methods in Dynamical Systems and Control," Sandia National Laboratory, Albuquerque, NM, August 2000.

"Methods for Large Scale Eigenvalue Problems," (Part of Numerical Linear Algebra for High Performance Computers Tutorial - Dongarra, Duff, Sorensen, Van der Vorst), VECPAR 2000, Porto, Portugal, June 2000.

"Balanced Model Reduction by Projection Methods," MTNS, Perpignan, France, June 2000.

"Balanced Model Reduction by Projection Methods," SIAM 2000 National Meeting, Rio Grande, Puerto Rico, July 2000.

"Methods for Large Scale Eigenvalue Problems," (Part of Numerical Linear Algebra for High Performance Computers Tutorial - Dongarra, Duff, Sorensen, Van der Vorst), Euro-Par 2000, Munich, Germany, August 2000.

"Balanced Model Reduction by Projection Methods," CCA 2000, Anchorage, Alaska, September 2000.

"Low Rank Approximate Solutions to Large Scale Lyapunov Equations," Meeting in honor of Pete Stewart, University of Maryland, October 2000.

"Low Rank Approximate Solutions to Large Scale Lyapunov Equations," SIAM Applied Linear Algebra Conference, Raleigh, NC, October 2000.

"Methods for Large Scale Eigenvalue Problems," (Part of Numerical Linear Algebra for High Performance Computers Tutorial - Dongarra, Duff, Sorensen, Van der Vorst), SC2000, Dallas, TX, November 2000.

Integer/Mixed Integer Programming and Large Scale Problems

Investigators: Nate Dean, Bob Bixby

Abstract: Our main objective during this period was to study applications and computational aspects of large-scale problems involving discrete optimization and networks and to develop algorithms for their solution. We focused primarily on techniques for mining existing data and solving network layout problems. For that purpose, we model objects derived from a large dataset as vertices in a network where an edge between two objects is weighted according to the similarity between them. Techniques for uncovering structure in networks are used together with some software that helps us view and understand such structure in terms of the original data. One of our main software tools was a computer implementation of an approach used by statisticians for data analysis, which was adapted to optimize network layouts. We acquired and installed some existing software (Xgobi, Xgvis, and Link) to explore this approach and its application to a variety of areas such as nano-technology, scheduling, telephone call data, market basket analysis, and DNA sequencing.

The following paragraphs summarize our recent papers.

In a rectilinear drawing of a simple graph G, each vertex is mapped to a distinct point in the plane and a straight-line segment with appropriate ends represents each edge. The goal of rectilinear crossing minimization is to find a rectilinear drawing of G with as few edge crossings as possible. A new approach to rectilinear crossing minimization is presented including a formulation of the problem as a mathematical program with a linear objective function and simple quadratic constraints. Then we size the problem for various graph families. These sizings provide a mechanism for ranking crossing number problems according to their apparent complexity. (See Dean, AMS 2001.)

Many different sizes and shapes of nanotubes have been found experimentally, and many others have been proposed because of their expected energetic, structural, and electronic properties. Even with the most advanced algorithms, sophisticated data structures, and parallel computing, directly determining the energetic stability of nanotubes is a quantum mechanical problem that is still beyond our reach. This paper attacks the problem of determining minimum energy configurations of single-walled carbon nanotubes using a mathematical programming model. The model includes a potential energy function that is minimized subject to constraints on the angular resolution and bond lengths. This approach seems to consistently produce stable configurations. (See Dean, IEEE-NANO 2001.)

Summary: Tim Redl is a graduate student, and this article (See Redl.) summarizes some of his work towards completing his PhD thesis. The article begins with a study of the major research regarding problems in timetabling, in particular two closely related problems that colleges and universities face each semester. These problems are (1) to construct a semester-long schedule of classes and (2) to create a conflict-free timetable for end-of-semester final examinations. He analyzes various existing theoretical and computational results (for example, alternative problem formulations, algorithms and commercial software) that have been and continue to be used in attempting to solve problems of this nature. The article concludes with an analysis of current class and final exam scheduling at Rice University and the University of Houston and presents results obtained by the author in attempting to solve a sample scheduling problem using actual data from Rice University and using implementations of several algorithms discussed earlier in the paper.

Publications

1. N. Dean, Mathematical-programming formulation of rectilinear crossing minimization, to be presented at a meeting of the American Mathematical Society, October 5-6, 2001.
2. N. Dean, Mathematical-programming model of bond length and angular resolution for minimum energy carbon nanotubes, to be presented at the IEEE-NANO Conference, October 28-30, 2001.
3. T. Redl, Color me a schedule: Timetabling as a graph coloring problem - theory and practice, 2001 (First draft under revision.)

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

Investigators: Petr Kloucek, Frank Toffoletto

Abstract: We address applied research in the area of the meso-scale numerics. The mesoscale is a physical range interpolating between the microscale, which is inherently, stochastic in the sense of Heisenberg, and the macroscale that is deterministic and described using the continuum. This activity is relevant to a number of applications currently being pursued in LANL. These applications include modeling of elastic properties of foams, thermodynamics of twinned shape memory alloys, and the study of the origin of instabilities in the Earth's Magnetosphere.

Our objective is to develop comprehensive computational tools for modeling of the kinetics of microstructural evolution coupled with large-scale continuum phenomena under thermo-mechanical loading in compound structures. We strive to approximate the microstructural kinetics using the first order implicit partial differential equations with a certain specific type of Dirichlet boundary conditions. We plan to implement these tools using the FEAT operator calculus within the Java high performance framework.

We have developed *Subgrid Projection Method* which allows for the first time to approximate stochastic objects such as the probability measures by deterministic processes including gradient flows or visco-elastic oscillations

We developed new computational approach to mesoscale physics problems based on the *adaptive nonconforming domain decomposition technique* using Dunford-Pettis equi-integrability condition. The method is designed to theoretically capture solutions with unbounded internal scales, which are described, by the solutions of the implicit partial differential equations.

Collaborations in Progress

Coupling of stochastic and deterministic processes: We work with the group of Dr. J. Brackbill on the development of the coupling conditions for the Landau-Vlasov equations with the standard Magneto-hydrodynamic solvers. The collaboration led to the internship of Jennifer Wightman, a Rice University graduate student.

Plasma simulations: Jennifer Wightman worked with Dr. J. Brackbill on a problem in plasma simulation, the finite grid instability. The new idea is that the results of another LANL-T3 project show that the instability does not occur for total free energy conserving methods. Jennifer looked at an electrostatic particle-in-cell problem in 1-D, and compared the stability of explicit and implicit solutions. She will continue this work within the framework of her PhD program with J. Brackbill as her co-advisor. We are planning another internship for the next summer at LANL.

*Java high performance computing***:** Given the common interests and complementary capabilities of the CartaBlanca team at Los Alamos led by W. Brian VanderHeyden, Group-T3, and the Computer Science Institute at Rice, we pursue to advance the development of Rapid Prototyping Environment (RPE) and Java high performance computing capabilities. The following are areas of existing and future collaboration:

- use of fast array packages for improved speed of Java high performance computing applications;
- development of adaptive non-conforming Finite-Element and Rough Finite Element branch of CartaBlanca;
- incorporation of FEAT-based operator calculus into CartaBlanca;
- and incorporation of mesoscale numerical module into CartaBlanca RPE in addition to the multiphase flow package for examination and modeling of thermodynamics of structured materials, coupling of stochastic and deterministic models such as Fokker-Planck or Landau-Vlasov-Maxwell equations with Magneto-hydrodynamics models to investigate instabilities in aurora and magnetosphere

URLs:

http://www.caam.rice.edu/~luism/

http://www.caam.rice.edu/~reynoldd/

http://www.caam.rice.edu/~wightman/

Publications

1. P. Kloucek and M. V. Romerio, The first Order Phase Transitions of Bubbles at Solid-Liquid Interface, Technical Report #1, January 2001, Swiss Federal Institute of Technology, submitted to Arch. Rat. Mech. Anal., 2001
2. D. Cox, P. Kloucek and D. R. Reynolds, The Non-Local Relaxation of Non-Attainable Differential Inclusions Using Subgrid Projection Method - One Dimensional Theory and

Computations, Technical Report #10, July 2001, Swiss Federal Institute of Technology, submitted to J. Nonlin. Sci., 2001

3. P. Kloucek and L. A. Melara, The computational modeling of incoherent phase boundaries, Technical Report #13, August 2001, Swiss Federal Institute of Technology, submitted to J. Comp. Phys., 2001
4. P. Kloucek, F. Toffoletto, and J. Wightman, Rough Finite Element Solver for the Magnetosphere-Ionosphere Coupling Equation, *Poster Presentation at the National Science Foundation-GEM Annual Meeting*, June 2001
5. P. Kloucek and D. R. Reynolds, A Mesoscale Model for the Phase Transformation Kinetics of Shape Memory Alloys, with Application to Vibration Damping, *Proceedings of The Fourth European Conference on Numerical and Applied Mathematics*, Ischia Porto, Italy, July 2001
6. P. Kloucek and L. A. Melara, The computational modeling of multiscale phase boundaries, *Proceedings of The Fourth European Conference on Numerical and Applied Mathematics*, Ischia Porto, Italy, July 2001

Visits and Other Interactions

April 12-16, 2000: Dr. Petr Kloucek and Dr. Michel Romerio (Swiss Institute of Technology), LANL materials group, "*The first order phase transition of bubbles at solid-liquid interface."*

September 2000: *Dr. Petr Kloucek and Dr. Frank Toffoletto*. The first LACSI conference, mini-symposium on The *Mesoscale Numerics.*

February 5-7, 2001: *Dr. Jeremiah Brackbill (LANL*), Rice University, Dept. of Computational and Applied Mathematics Colloquium, *Transient Magnetic Reconnection and Unstable Shear Layers.*

February 5-7, 2001: *Dr. Joachim Birn (LANL*), Rice University, Dept. of Computational and Applied Mathematics, Mesoscale Numerics Talk, *Magnetic Reconnection and Magnetospheric Substorms.*

May 2-4, 2001: *Dr. Petr Kloucek,* LANL T-3 talk, *Modeling of Mesoscale Physics Problems*

June 27 – July 24, 2001, Jennifer Wightman, graduate summer internship with LANL T-3, Dr. J. Brackbill, *Solving Grid Instability in Plasma Simulations.*

September 2001: *Dr. Giovanni Lapenta (*LANL) Rice University, Dept. of Computational and Applied Mathematics Colloquium, *Computational Stochastic Physics Models.*

Methods and Software for Inverse and Control Problems

Investigators: Bill Symes

Abstract: Characterization and control of complex dynamical systems, such as wave propagation in heterogeneous materials, requires coupling of very expensive and inexact simulations, often involving sub-simulations interacting at vastly different scales, with appropriate optimization and linear algebra algorithms. These algorithms must incorporate

features not found in contemporary library code, such as management of the interaction of simulation and optimization errors, scalability with maximal architecture neutrality, and transparent management of data storage amongst devices of varying latency. It is virtually inconceivable that such tools could be built, maintained, and successfully transferred to end users without the employment of object-oriented programming methodology.

Progress

In summer 2000, we (Symes and then Rice graduate student Mark Gockenbach, now at Michigan Tech) declared the Hilbert Class Library, version1.0, open for business and put a notice in NADIGEST to that effect. We got a dozen or so email inquiries, including some reports of ports to Windows NT. More recently, a group at Sandia/Livermore (Paul Boggs, Juan Meza, and coworkers) has decided to adopt HCL as the basis for their next generation optimization and control software. (See http://www.trip.caam.rice.edu/txt/hcldoc/html/index.html)

Last year we began a discussion with a number of people at various national laboratories and universities about the design of standard numerical classes. A critical issue is the flexibility of method definition for classes, for example those defining the vector abstraction. It is simply impossible to provide all conceivable functions explicitly that algorithm writers might need, and it is also unacceptable to require algorithm writers and users to hack the basic vector interface to include whatever they need that isn't there. Roscoe Bartlett, who has now graduated from Larry Biegler's group at CMU and works at Sandia/Albuquerque, came up with a beautiful solution: method forwarding. This is not a new idea (both the STL and PETSc use similar ideas), but it really does solve the problem, to the extent that it can be solved in contemporary computing environments. The algorithm writer provides any necessary functions in the form of function object classes; objects of these classes can be passed to an evaluation method of a vector object.

After a lot of discussion with Shannon Scott, a Rice M.A. student who graduated Spring 2001, we made a proposal for a standard vector class design, which we believe to be somewhat more flexible and powerful than Bartlett's. The paper and some code fragments are available through the TRIP web page, www.trip.caam.rice.edu. We continue to develop these ideas in C++, which will become HCL2.0 or the basis for a Standard Vector Library. We also plan to develop a parallel set of Java interfaces. As compiler technology has caught up with more of the ISO C++ standard, we're using templates, exception handling, and standard library classes extensively.

One of the key difficulties in this enterprise is provision for parallel execution (and efficiency, generally). Scott's thesis demonstrated a client-server design using CORBA. He built HCL classes with both client-side proxy and server-side remote components, for vectors and linear operators; only the server code is distributed. The linear operator figured in a simple boundary control problem for the acoustic wave equation. He was able to solve the control problem with a parallel domain decomposition solution of the wave equation, but with the optimization client consisting entirely of undistributed, standard serial HCL code. Thus, his middleware effectively made HCL and the MPI-based wave equation solver into components, and provided a framework for their interaction, entirely using off-the-shelf commercial parts.

Scott's thesis is available on the Rice/CAAM web site, http://www.caam.rice.edu.

We have proposed that all services requiring special environments, such as parallel execution, be implemented in a client server framework like the one demonstrated in Scott's thesis. That way, the special features needed in part of the program (like MPI for the parallel PDE solver in Scott's thesis) do not pervade the rest of the code. We believe that this use of component design is essential to the creation of a standard set of numerical classes.

Scott's work is being followed up both here and at Michigan Tech. Mark Gockenbach and one of his students have formalized his constructions in a remote data container class which will be used to build remote/distributed vector classes. Rice graduate student Eric Dissuade has used our prior work on generic implementation of the adjoint state method to attack a satellite data assimilation problem proposed by our colleagues in Space Physics. This project is just starting; Eric carried out data assimilation for a simple model as a class project this spring, and then converted his MATLAB code to a HCL application suitable for models that are more useful. Rice graduate student Hala Dajani has taken an interest in component frameworks and distributed computing in the context of HCL, and has been working with me this summer on the new class library. She is also interested in developing the Java analog. Graduate student Tony Padula has been at Sandia this summer working with Roscoe Bartlett's rSQP++, and will be involved with us when he returns.

We plan to convene a workshop on OO standard interfaces for optimization and its applications at Rice in January 2002. We had a first informal workshop at Argonne in Fall 2000; the interested group of academic, laboratory, and industry people has grown since then.

Personnel changes at LANL have made it difficult to maintain enough continuity to foster visits between the institutions. Ironically, the work initiated under LACSI has catalyzed (See above.) active collaborations with researchers at Sandia Albuquerque, Sandia Livermore, and Argonne National Laboratories. We will attempt to rekindle interest from LANL at the LACSI Symposium.

Publications

Shannon Scott, Software Components for Simulation and Optimization, Masters Thesis, Rice University, CAAM-T01-06, April 2001.

Efficient Parallel Solvers for Multiphase Flow in Strongly Heterogeneous Porous Media

Investigators: Roland Glowinski, Yuri Kuznetsov, and Lennart Johnsson.

Abstract: The main goal of this project is to investigate efficient parallel solvers for multiphase flow in strongly heterogeneous porous media, a related goal being the simulation of wave propagation in heterogeneous media, another high-priority problem for LANL scientists. Indeed, the main trust of the research effort during these last two years has been the investigation of computational methods for the numerical simulation of electro-magnetic and acoustic waves in unbounded media with obstacles. Two major difficulties with these problems are:

1. The implementation of appropriate boundary conditions transparent to the outgoing waves, so that the computations can take place on a bounded domain containing the region of practical interest.

2. The computational treatment of the obstacles using finite volume or finite element meshes as uniformly as possible.

The first difficulty has been overcome using the PML methodology (PML stands for Perfectly Matched Layer). Concerning the treatment of obstacles we are currently investigating and testing a combination of fictitious domain (domain embedding) and time discretization by operator- splitting methods; the results obtained by our Ph.D. students (V. Bokil and S. Lapin) look promising. Some remarks are in order:

The combination domain embedding and operator splitting should be an important ingredient when addressing the simulation of multiphase flow in heterogeneous and porous media, since these techniques are, in particular, well suited to parallelization.

We found it convenient to reformulate the staggered finite volume and finite difference schemes advocated by the LANL scientists in the mixed finite element “language”. Through this reformulation, the practitioner has access to the greater flexibility and well-documented results and algorithms associated with variational methods.

V. Bokil is a graduate student at University of Houston preparing a Ph.D. under the dual supervision of R. Glowinski (UH) and M. Hyman (LANL).

Publications

A lecture on results obtained in this project has been given at the July 2001 SIAM Annual Meeting in San Diego, California. Its title was: “A 2-D finite element formulation of the uniaxial perfectly matched layer”; an article on the above topics is in preparation.

Visits and Other Interactions

November 1999: R. Glowinski and L. Johnsson visited LANL, had discussions with local scientists, and R. Glowinski gave a lecture at the Scientific Computing seminar.

February 2000: R. Glowinski visited LANL to discuss the joint LACSI project with M. Hyman and L. Winter.

May 15--August 10, 2000: V. Bokil (UH Ph.D. student) visited LANL T-7 to work under the supervision of M. Buksas and M. Hyman.

August 2000: Participation by V. Bokil, R. Glowinski, and Y. Kuznetsov in the LACSI Symposium.

November 2000: V. Bokil and R. Glowinski met with M. Buksas and M. Hyman to discuss results and objectives.

May 14--July 13,2001: V. Bokil visited LANL-T 7 to work under the supervision of M. Buksas and M. Hyman.

PARALLEL NUMERICAL METHODS FOR THE DIFFUSION AND MAXWELL EQUATIONS IN HETEROGENEOUS MEDIA ON STRONGLY DISTORTED MESHES

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

Abstract: The main objective of the project is to develop efficient parallel solvers for large-scale algebraic saddle point problems that arise from special finite difference/finite element approximations of the diffusion and Maxwell equations in highly heterogeneous media on strongly distorted meshes. This problem is a particular interest for several research groups in LANL and UH. During the reporting period, the main efforts have been concentrated in the area of diffusion equations. A new algebraic preconditioned iterative solver has been designed, implemented, and evaluated, both theoretically and by numerical experiments.

Efficient numerical methods (discretization and algebraic solvers) for the diffusion equations in highly heterogeneous an isotropic media on strongly distorted meshes are in the focus of research interest of scientists and engineers working in the area of computer simulation of complex physical phenomena. For the challenging applications, the algebraic mesh systems are extremely large scale (up to 100 million unknowns) and always very ill-conditioned. It is still unclear for numerical mathematicians how to overcome the difficulties resulting from anisotropic coefficients and strongly distorted meshes.

Summary of Results

A new approach based on a combination of mixed hybrid finite element technique and mimetic finite difference schemes has been developed for the accurate approximations of the diffusion equations for the case of strongly distorted meshes.

A new algebraic multilevel preconditioner with projectors on the mean values of iterated mesh functions has been proposed and investigated. Numerical experiments with a number of selected real life test cases have demonstrated very good arithmetic complexity (efficiency) of the proposed preconditioner.

Experimental codes have been developed for 2D and 3D diffusion equations for the case of logically rectangular distorted meshes.

Further plans:

A parallel version of the developed codes is planned to be ready for numerical experiments by September 2001. This may be delayed due to the recent flooding in Houston.

In January 2001, we started to work for the Maxwell equations in heterogeneous media for the case of rectangular prismatic meshes. We expect the first results in November 2001.

M. Brendt, Y. Kuznetsov, K. Lipnikov, D. Moulton, and M. Shashkov are preparing a paper on mimetic/mixed hybrid discretization of diffusion equations.

Visits and Other Interactions

October 1999: M. Shashkov (LANL) visited UH, presented a talk at the seminar on Scientific Computing, and discussed the problems of common interest.

February 2000: Y. Kuznetsov visited Santa Fe to discuss the joint LACSI project with M. Shashkov.

August 2000: The team (M. Shashkov, Y. Kuznetsov, R. Glowinski, K. Lipnikov, A. Hayrapetyan) held a meeting at the LACSI Symposium in Santa Fe.

January 2001: M. Shashkov and D. Moulton visited Houston (UH) to discuss the scientific results of the projects and further research plans.

May-August, 2001: UH graduate students K. Lipnikov and A. Hayrapetyan visited LANL as graduate student interns.

August 2001: Y. Kuznetsov visited LANL. He presented a lecture at the seminar on scientific computing and discussed the recent results and further developments of the project with M. Hyman, M. Shashkov, J. Dendy, and D. Moulton.