09061 Abstracts Collection

Combinatorial Scientific Computing

— Dagstuhl Seminar —

Uwe Naumann[1], Olaf Schenk[2], Horst D. Simon[3], and Sivan Toledo[4]

[1] RWTH Aachen, D
naumann@stce.rwth-aachen.de
[2] University of Basel, CH
olaf.schenk@unibas.ch
[3] Lawrence National Lab. - Berkeley, USA
hdsimon@lbl.gov
[4] Tel Aviv University, IL
stoledo@tau.ac.il

Abstract. From 01.02.2009 to 06.02.2009, the Dagstuhl Seminar 09061 “Combinatorial Scientific Computing ” was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Graphs, combinatorics, high-performance scientific computing

1 Background

Motivation

The ongoing era of high-performance computing (HPC) is filled with enormous potential for scientific simulation, but also with daunting challenges. Architectures for HPC may have thousands of processors and complex memory hierarchies paired with a relatively poor interconnecting network performance. Thanks to the advances being made in computational science and engineering, the applications that run on these machines involve complex multiscale or multiphase physics, adaptive meshes and/or sophisticated numerical methods. A key challenge for scientific computing is obtaining high performance for these advanced applications on such complicated computers and, thus, to enable scientific simulations on a scale heretofore impossible.

One aim of the proposed seminar is to address these challenges by setting the stage for accelerated development and deployment of fundamental enabling

Dagstuhl Seminar Proceedings 09061
Combinatorial Scientific Computing
http://drops.dagstuhl.de/opus/volltexte/2009/2112

technologies in high performance computing. Relevant issues arise in load balancing and parallelization on high-performance computers, large-scale optimization, automatic differentiation of numerical simulation codes, and sparse matrix software tools. These seemingly disparate areas are unified by a common set of abstractions and algorithms based on combinatorics, graphs, and hypergraphs that represent the main focus of this seminar.

In-Depth Description of the Topic

The human desire for meaningful numerical simulation of physical, chemical, and biological phenomena in science and engineering has been increasing with the growing performance of the continuously improving computer systems. As a result of this development we are (and will always be) faced with a large (and growing) number of highly complex numerical simulation codes that run at the limit of the available high-performance computing resources. These codes often result from the discretization of systems of differential equations. The runtime correlates with the resolution that often needs to be very high in order to capture the real behavior of the underlying system. There is no doubt that the available hardware will always be used to the extreme. Improvements in the runtime of the simulations need to be sought through research in numerical algorithms and their efficient implementation on parallel architectures.

Many of the resulting problems are combinatorial in nature. Most of those are known to be computationally hard in the sense that efficient (polynomial in the required time and memory space) algorithms for their exact solution are unlikely to exist. For example, we have to deal with partitioning, elimination ordering, coloring, and matching problems for graphs and hypergraphs in various contexts. A good approximation of the solution to these abstract problems may lead to a significant decrease in the runtime of numerical programs that implement solvers for partial differential equations, nonlinear optimization algorithms, or solvers for generalized Eigenvalue problems. For example, the use of cutting-edge load balancing software (Zoltan[5]) enabled researchers at the Stanford Linear Accelerator Center to decrease the runtime of their pulse propagation and wakefield code Tau3P by up to 68% on a 512-processor IBM SP3.

Problem sizes are typically now in the millions of unknowns; and with emerging large-scale computing systems, this size is expected to increase by a factor of thousand over the next five years. Moreover, simulations are increasingly used in design optimization and parameter identification which is even more complex and requires the highest possible computational performance and fundamental enabling algorithmic technology. Derivatives of certain objectives of these numerical models with respect to a potentially very large number of model parameters are crucial for the highly desirable transition from pure simulation to optimization. Approximation of these derivatives via finite difference quotients often lacks the required accuracy. More importantly, it may be infeasible for a large parameter space in terms of its computational complexity. Adjoint numerical programs

[5] Zoltan is developed at Sandia National Laboratory

have until recently been written by hand to overcome this problem. Such programs compute large-scale gradients with machine accuracy at a small constant multiple of the computational complexity of the underlying forward simulation. Due to the enormous size of most numerical simulation codes the manual procedure may take up to several man years. Moreover manual adjoint codes are error-prone and hard to maintain as the forward simulation evolves. Special compilers are being developed based on the principles of Automatic Differentiation (AD) to generate adjoint code automatically. A large number of hard problems, many of them combinatorial in nature, need to be solved in this context. They range from classical covering and coloring to elimination and data-flow reversal problems. Graphs, in particular bipartite and directed graphs, play a central role in AD. Parallelism needs to be dealt with from two perspectives. First, the given numerical simulation program most likely runs on a parallel architecture, which must be taken into account by the semantic code transformation algorithms. Second, the adjoint code itself can be generated to use additionally available computing resources for an efficient solution of the data-flow reversal problem arising in this context. Last but not least, links between combinatorial problems in AD and those arising in large-scale optimization and sparse linear algebra have only recently started to be exploited.

What binds together the community of combinatorial scientific computing is the focus on practical use of graph algorithms and combinatorial algorithms to address a variety of different problems (load balancing, AD, matrix reordering) that all arise in scientific computing. This shared common denominator allows us to interact productively and to advance the state of the art in several different problem areas.

Objectives and Expected Results

The common combinatorial and graph-algorithmic aspects of the following questions will be addressed by the seminar.

1. How to provide advanced capabilities in load balancing and parallelization toolkits for high-performance computer architectures?
2. How to provide efficient directional derivatives and adjoints by Automatic Differentiation?
3. How to advance the state-of-the-art in sparse matrix algorithms and software?
4. How to provide innovative strategies to accelerate the algorithmic kernels arising most often used in simulations and large-scale optimizations?

The major focus is on a graph-based combinatorial perspective. Complete answers will have to consider additional issues in computer architecture, compiler technology, and software engineering. We hope to approach this kind of answers by seeding new future collaboration with representatives of the respective areas.

Outreach into areas of science and engineering (e.g. PDE-constrained optimization) that face similar combinatorial problems as the CSC community is a major objective. We consider Schloss Dagstuhl to be a perfect setting for such an ambitious undertaking. It provides a stimulating atmosphere for personal contact between distinguished experts in the respective fields supported by numerous high-quality presentations.

The purpose of the seminar is to bring together a diverse community of researcher working in different aspects in this exiting field. We will gather graduate students, young researcher, scientists in mid-career, and senior investigators from both academia and industry. Some are experts on graph combinatorial aspects, some are focusing on theoretical analysis, and some are more directed towards software development and concrete applications.

2 Executive Summary

The activities of the seminar focused on combinatorial issues in high-performance scientific computing. The activities included:

- eight one-hour invited talks
  - Bruce Hendrickson,[6] Sandia National Laboratory: Combinatorial Scientific Computing: A View to the Future
  - Alex Pothen,[7] Purdue University: Graph Matchings in CSC
  - Rob Bisseling,[8] Utrecht University: Combinatorial Problems in HPC
  - Paul Hovland,[9] Argonne National Laboratory: Combinatorial Problems in Automatic Differentiation
  - Iain Duff,[10] Rutherford Appleton Laboratory and CERFACS: Combinatorial Problems in Numerical Linear Algebra
  - Ruud van der Pas,[11] SUN Microsystems: Present and Future of High-Performance Scientific Computing
  - David Bader,[12] Georgia Institute of Technology: Emerging Applications in Combinatorial Scientific Computing
  - Michael Mahoney,[13] Stanford University: Combinatorial and scientific computing approaches to modern large-scale data analysis
- thirteen 20-minute contributed talks by participants from seven countries
- six software tutorials
  - Eric Boman, Sandia National Laboratory: Zoltan[14]
  - Francois Pellegrini, LABRI: Scotch and PT-Scotch[15]
  - Jean Utke, Argonne National Laboratory: OpenAD[16]
  - Andreas Wächter, IBM Research: Ipopt[17]
  - Andrea Walther, Technical University Dresden: ADOL-C[18]
  - John Gilbert, University of California Santa Barbara: Star-P[19]
- three round table discussions
  - Graph coloring for parallel computation (organizer: Assefaw Gebremedhin, Purdue University)
  - Multilevel algorithms for discrete problems (organizer: Eric Boman, Sandia National Laboratory)

[6] URL: `http://www.sandia.gov/~bahendr/`

[7] URL: `http://www.cs.purdue.edu/people/faculty/apothen/`

[8] `http://www.math.uu.nl/people/bisselin/`

[9] `http://www.mcs.anl.gov/~hovland`

[10] `http://www.numerical.rl.ac.uk/people/isd/isd.html`

[11] `http://blogs.sun.com/ruud/`

[12] `http://www.cc.gatech.edu/~bader/`

[13] `http://cs.stanford.edu/people/mmahoney/`

[14] `http://www.cs.sandia.gov/Zoltan/`

[15] `http://www.labri.fr/perso/pelegrin/scotch/`

[16] `http://www-unix.mcs.anl.gov/OpenAD/`

[17] `https://projects.coin-or.org/Ipopt`

[18] `http://www.math.tu-dresden.de/~adol-c/`

[19] `http://www.interactivesupercomputing.com/`

- Data-Flow Reversal in Adjoint Codes (organizer: Uwe Naumann, RWTH Aachen University)

Participants also enjoyed one of two recreational activities in a free afternoon (a long hike or a trip to Trier). As usual in Dagstuhl seminars, participants also engaged in lively informal professional (and personal) discussions during breaks, at meal times, and late into the night in the cafeteria / the wine cellar.

2.1 Invited Talks

Some of the invited talks surveyed different problem areas within combinatorial computing (all of these also described very recent research). These talks included Henrickson's talk, which surveyed the entire field and attempted to define it in both old and new ways. The talks by Pothen, Bisseling, Hovland, and Duff each focused on one problem area within the field. Pothen's talk focused mostly on linear-time, highly parallel approximation algorithms for weighted graph matching and for graph coloring. Bisseling's talk focused mostly on parallel graph partitioning. Hovland surveyed the area of automatic differentiation, with a focus, of course, on the combinatorial problems that arise in such computations. Duff talk focused on algorithms for sparse-matrix factorizations; it covered both traditional topics like the multifrontal method and elimination data structures, as well as on recent development, like the use of graph matching to enhance efficiency and parallelism in sparse factorizations. Bader focused on the use of high-performance accelerator processors (the Cell BE and GPUs) to speed up the solution of large-scale combinatorial problems in science.

The relatively large number of invited survey talks served as a community-building tool. The combinatorial scientific computing has been holding specialized meetings for less than 5 years, and it is still important for us to teach each other about the specific problems that each one of us address and about and techniques that we use to solve them. We have recognized that we share significant commonality, but we still make an effort to more precisely define the community and to enhance the scientific connections between its members. Our situation is quite different from that of computation geometry, say, a community that has been conducting Dagstuhl seminars for 18 years (and other meetings even earlier).

In fact, the common threads of combinatorial scientific computing revealed themselves in several ways during the seminar. Hendrickson, in the first talk of the seminar, suggested that the field is defined not only by the focus on combinatorial problems and algorithms in scientific computations, but also by a shared aesthetic; a common sense of what makes a problem important and beautiful, and what makes a proposed solution as a success. For example, researchers in combinatorial scientific computing favor problems that have a tangible impact on society (better science, better medicine, not just cleverer math); they therefore tend to assess results in a relatively practical way; they tend assess algorithms and implementations as a whole, not to focus on one or the other; and so on. Another common thread was the focus on high-performance computing.

It was widely recognized by participants that one cannot usually make a significant progress in scientific computing without paying attention to parallelism, because the only computers that can solve large-scale problems are highly parallel. This thread quickly led to another common concern, regarding the ability to implement our algorithms in a way that achieves high performance but without forcing us to spend significant amount of time tuning the implementation to each parallel machine (a fairly common behavior in high-performance computing).

Two long talks were given by people from outside the community. The talk by van der Pas focused on programming tools for high-performance computing. As mentioned above, this is an area that many members of the community feel passionately about, because they program high-performance machines and they struggle with the tools (compilers, profilers, programming languages). The talk was, therefore, very well received.

The other talk from outside the community was by Mike Mahoney, a theoretical computer scientist who talked about new techniques in large-scale data analysis. The examples in the talk drew both from non-science applications (discovering online communities from records of personal interactions on the internet) and from scientific computations (discovering significant SNPs in gene databases). The talk was exciting in that some of the problems that Mahoney talked about were clearly related to scientific computations, they were clearly combinatorial, yet the techniques that he used were very different from those used by our community. This is to a large extent due to the fact that in biological data analysis the problems are not always well defined mathematically, as opposed to the physical sciences where problems are usually well defined. But the talk inspired some participants to search for applications of the techniques that Mahoney described.

Both Mahoney and van der Pas attended the whole seminar; they participated in many technical discussions with other participants and enriched our community and the seminar.

2.2 Contributed Talks

Virtually all the contributed talks described very recent reseach. Three were given by PhD students (Donfack, Daitch, and Langguth). They covered results in many areas of combinatorial scientific computing: automatic differentiation (Gebremedhin, Lyons, Steihaug), sparse factorizations (Davis and Li), ordering for sparse factorizations (Reid, Donfack, Scott), combinatorial preconditioning (Daitch), and huge-scale parallel PDE solvers (Arbenz, Ruede). Two interesting contributions came from long-time members of the community who are now working in new problem areas: genome sequencing (Catalyurek) and graph visualization (Hu).

2.3 Tutorials

The seminar included several hands-on tutorial on software packages that researchers in the community have been developing. These tutorials were intended to give other researchers first-hand experience in using the software, and in allowing them to continue using them on their own more easily. In other words, they were part demos to show the tools and part tutorials to make it easy to learn the tools.

The tutorials were conducted using a large cluster of laptops that were brought for this purpose from Aachen University by organizer Uwe Naumann and his colleagues from Aachen. All the relevant software packages have been installed on the laptops prior to the tutorials.

The demo part of the tutorials went fantastically well. It was a joy to see the developers of the software demonstrate it using simple examples in an interactive way. In many cases, the audience asked the tutorial presenter to try other things than he or she prepared, and the interactive nature of the demonstration was very lively. The fact that each participant had access to a laptop running the same software also helped, as participants were able to run simple examples and to examine the structure of the software and to look at a few source files.

The goal of actually teaching participants to use the software was perhaps a bit ambitious for 2-hour tutorials, but they certainly gave participants an opportunity to get started more easily than in their own offices back home, without the benefit of having the lead developer right there to answer questions. Feedback from participants and the organizers Feedback from participants has been very positive both in terms of the anonymous survey and in terms of what participants told the organizers. The results of the anonymous survey are largely consistent with those of other Dagstuhl seminars (as summarized in the last-60-days statistics), with many questions on which this seminar scored higher than average and a few on which it scored lower than average.

Many of the suggestions and constructive comments that participants filled in the survey reflect the wide range of expectations and wishes of participants. For example, some wished for fewer talks and more unstructured time for discussions; but many participants expressed a wish to give a talk. The organizing committee tried to balance these expectations. Similarly with respect to the length of talks: short talks are harder to follow, but leave more time for other activities.

2.4 Posters

Refer to Section 3.5 for details on the posters.

2.5 Post-Seminar Book

Two of the organizers, Uwe Naumann and Olaf Schenk, are working on a proposal to CRC Press for publication of a special collection of articles on Combinatorial Scientific Computing in their Chapman & Hall/CRC Computational Science series. The purpose of the book is to provide the first collection of references for a diverse community of researcher working in different aspects in the exiting field of CSC. The content is strongly motivated by this seminar including survey articles as well as tutorial-style software guides. Potential readers include graduate students, young researcher, scientists in mid-career, and senior investigators from both academia and industry. Some are experts on graph combinatorial aspects, some are focusing on theoretical analysis, and some are more directed towards software development and concrete applications. Outreach into areas of science and engineering that face similar combinatorial problems as the CSC community is a major objective.

2.6 A Suggestion

In the survey, some of our participants expressed a concern regarding the difficulty of traveling to Dagstuhl and back, and about some uncertainty regarding the accommodation. Even though all the information is available on the Dagstuhl web site, preparing for a trip like this is not a trivial task and it does involve some anxiety, especially to researchers from outside Germany, who are not familiar with the German rail and bus systems.

One way to alleviate some of this anxiety is to prepare a short on-line video that would quickly show visually how to get to Dagstuhl and back and what to expect once you are there. Pictures of the signs at Frankfurt airport to the train station, screenshot of the www.bahn.de web site showing which station to buy tickets to, telling people how to make a taxi reservation or check the bus schedule, show what the rooms at Dasgtuhl look like, and so on. This is clearly not essential, but it could make the trip just a bit easier.

The same video could also suggest some "Dagstuhl interaction strategies". The most valuable experiences in Dagstuhl seminars are often unplanned highly technical discussions, often by people who are quite far apart in terms of expertise. These discussions tend to bridge gaps between subcommunities and enable new connections within computer science. Such discussions are easier to have in Dagstuhl than anywhere else, but it is fairly easy to miss them. A stay at Dagstuhl can be enhanced significantly if a participant approaches other participants directly and ask "can you explain to me in more detail the technique you mentioned in your talk", or "I will graduate in the summer; can you tell me about postdoc opportunity in your lab?". If participants leave these interactions to chance, they sometimes happen but sometimes don't. It may be helpful to find ways to suggest these things to participants, especially junior ones; perhaps through the web site, or through the video, or even to ask organizers to talk about this in their opening remarks.

3 Abstract Collection

3.1 Monday, Feb 2, 2009

1. Plenary Talks

Combinatorial Scientific Computing: A View to the Future

Bruce Hendrickson (Sandia National Labs, Department of Discrete Algorithms & Mathematics, Albuquerque (NM), USA)

Since its emergence with the first SIAM workshop in 2004, Combinatorial Scientific Computing has become well established as an important component of the computational science landscape. As was foreseen at that first meeting, combinatorial insights and techniques have grown in importance and visibility as applications and computers have become increasingly complex.
In this talk, I'll present a personal perspective on future opportunities in CSC that emerge from the dramatic changes we are seeing in computer architectures, and also from the ongoing changes in scientific applications.

Keywords: Combinatorial scientific computing

Graph Matchings in CSC

Alex Pothen (Purdue University, Computing Research Institute, West Lafayette (IN), USA)

In the first part of this talk, I will discuss the CSCAPES Institute, a team of CSC researchers from several institutions, developing combinatorial algorithms to enable computational simulations at the frontiers of science and engineering. The research themes of the CSCAPES Institute include parallelization and load balancing toolkits, automatic differentiation technology, and parallel algorithms for graph coloring and graph matching.
In the second part of the talk, I will discuss our work on computing matchings in graphs on parallel computers. Matching is a fundamental problem in computer science and computational science and engineering, but it has proved to be hard to develop parallel algorithms for it. Recently several approximation algorithms have been developed for matching that are more amenable to parallelism. We describe scalable approximation algorithms for computing matchings on computers with thousands of processors.

Keywords: CSCAPES institute, parallel matchings

Combinatorial Problems in HPC

Rob Bisseling (Utrecht University, Department of Mathematics, Utrecht, the Netherlands)

This talk will present a survey of combinatorial problems encountered in scientific computations on today's high-performance architectures, with sophisticated memory hierarchies, multiple levels of cache, and multiple processors on chip as well as off-chip.
For parallelism, the most important problem is to partition sparse matrices, graph, or hypergraphs into nearly equal-sized parts while trying to reduce inter-processor communication.
For better cache performance, the problem is to reorder sparse matrices by suitable row and column permutations.
Common approaches to such problems involve multilevel methods based on coarsening and uncoarsening (hyper)graphs, matching of similar vertices, searching for good separator sets and good splittings, dynamical adjustment of load imbalance, and two-dimensional matrix splitting methods.

Keywords: HPC, hypergraphs

2. Contributed Talks

Node and Edge Label Overlap Removal in Visualizing Large Network

Yifan Hu (AT&T Research, Department of Information Visualization, Florham Park (NJ), USA)

Networks/graphs are increasingly used to encapsulate and to analyze vast amount of data available to us. Areas where networks are pervasive including biology (protein networks), social network and communication (Internet maps). Good visualization of networks can reveal structures intrinsic to the data, and lead to increasing understanding and even to discovery of new knowledge.
A number of algorithms have been proposed to visualize graphs as a collection of nodes and edges in two or three dimensional space. The force-directed model (Eades, 1984; Fruchterman and Reingold, 1991) is widely used for drawing undirected graphs. It is relatively easy to implement, and when combined with the multilevel approach and suitable data structures (e.g., quad-tree) to approximate long range repulsive forces, is very efficient and generally effective for large graphs (Hachul and Junger, 2005; Hu, 2005). Another popular algorithm is based on the stress model (Kamada and Kawai, 1989), which minimize a stress function to realize desired distances between vertices.
Most of these existing symmetric graph layout algorithms, however, treat nodes as points, and edges as a simple line. In practice, nodes and edges usually contain labels or graphics that need to be displayed. Naively incorporating this can lead to nodes and edges information that overlap, causing information of some nodes and edges to occlude that of others. If we assume

that the original layout conveys significant aggregate information such as clusters, the goal of any layout that avoids overlaps should be to retain the “shape” of the layout based on point nodes.

Node and edge label overlap removal algorithms have been proposed to address this need. For example, in node overlap removal, the Voronoi cluster busting algorithm (Gansner and North, 1998; Lyons, Meijer, and Rappaport, 1998) works by iteratively forming a Voronoi diagram from the current layout and moving each node to the center of its Voronoi cell until no overlaps remain. Another group of post-processing algorithms is based on maintaining the orthogonal ordering (Hayashi, Inoue, Masuzawa, and Fujiwara, 1998; Huang and Lai, 2003; Li, Eades, and Nikolov, 2005; Misue, 1995; Dwyer, Marriott, and Stuckey, 2005). However none of these existing algorithms are scalable for very large graphs. Furthermore, for small to medium graphs, many of the existing algorithms often produce overlap-free layout that do not resemble the original layout well.

We proposed new algorithms for node and edge label overlap removal. The new algorithms utilize a proximity graph of the nodes and edge labels in the original layout as a guide, and iteratively move the labels around while keeping the relative positions between them as close to those in the original layout as possible. The algorithms are efficient and can handle very large graphs. In contrast to existing algorithms, the new algorithms yield an overlap-free layout that preserves the structural information of the original layout better overall. We will compare the new algorithms with some existing algorithms using quantitative similarity measures, as well as illustrate the algorithms by application examples, including graphs from Mathematics Genealogy project.

In addition to label overlap removal, we will discuss challenges in dealing with large graphs in general, including how we approach the task of visualizing all the sparse matrices in the University Florida Sparse Matrix Collection. We will discuss open problems and issues in combinatorial computing uncovered in the project.

Keywords: Graph drawing, network visualization, multilevel, matching, maximal independent edge set, maximal independent vertex set

Micro-Finite Element Analysis of Human Bone Structures

Peter Arbenz (ETH Zurich, Institute of Computational Science, Zurich, Switzerland)

The investigation of the mechanical properties of trabecular bone presents a major challenge due to its high porosity and complex architecture, both of which vary substantially between anatomic sites and across individuals. A promising technique that takes bone microarchitecture into account is microstructural finite element (μFE) analysis. A very large number of finite elements is needed to accurately represent a human bone with its intricate microarchitecture; hence, the resulting μFE models possess a very large number of degrees of freedom. Detailed μFE models are obtained through high-resolution micro-computed tomography (μCT) of trabecular bone specimens allowing nondestructive imaging of the trabecular microstructure with resolutions on the order of 80 micrometer in living patients.

The discrete formulation is based on a standard finite element (voxel) discretization for linear elasticity. The degrees of freedom represent the displacements in the three axis directions at the voxel vertices.

A typical problem size is 100 million voxels corresponding to about 300 million degrees of freedom. Higher resolutions or larger bone specimens lead to larger models. We have simulated recently bones with up to 1.5 billion degrees of freedom [2]. A problem of 1 billion degrees of freedom generates a data volume – essentially the system matrix – of more than 1.25 TeraBytes that have to be stored in main memory.

We solve the resulting symmetric positive definite linear systems by the preconditioned conjugate gradient algorithm. We employ a smoothed-aggregation based algebraic multigrid (AMG) preconditioner [1].

A particular strength of our parallel finite element solver, ParFE [3], is the capability to construct the preconditioner without actually forming the system matrix. This matrix-free procedure reduces the mentioned memory requirements by 75%. Only a quarter of the memory (and thus processors) is needed to solve the problem. Our rule of thumb is a 1/2 million degrees of freedom per GigaByte of main memory. We used ParFE on the Cray XT3/XT4 and on the IBM Blue Gene to solve real world problems in just a few minutes.

A crucial issue in large scale computations is load balancing. This holds true in particular for structures as complicated as trabecular bones. The layered data layout that is induced by the CT data is not appropriate for the structural analysis. The data has to be redistributed in order to reduce interprocessor communication. This portion of the code, i.e., the repartitioning, does not yet scale beyond 1000 processors [2]. We will have to resolve this issue before being able to efficiently tackle problem sizes of several billion degrees of freedom. We also work on an entirely new approach that exploits the underlying Cartesian grid in order to save more memory space.

[1] P. Arbenz, G. H. van Lenthe, U. Mennel, R. Müller, and M. Sala: A scalable multi-level preconditioner for matrix-free μ-finite element analysis of human bone structures. *Internat. J. Numer. Methods Engrg.* 73 (7): 927–947 (2008).

[2] C. Bekas, A. Curioni, P. Arbenz, C. Flaig, G. H. van Lenthe, R. Müller, and A. J. Wirth: Extreme scalability challenges in micro-finite element simulations of human bone. *Concurrency and Computation: Practice and Experience*, 2008 (Submitted for publication).

[3] The PARFE Project Home Page. http://parfe.sourceforge.net/.

PDEs with a Trillion Elements: When Theory Fails to Predict Performance

Ulrich Rüde (University of Erlangen-Nuremberg, Department of Computer Science 10, Erlangen, Germany)

The performance of scientific programs today is rarely dominated by the cost of performing floating point operations, but by the cost of local and remote memory access, and by communication. Efficient numerical programs require using the memory hierarchy consciously and to optimize for data access costs. With the advent of multicore technology, we must expect that future super computer systems will have in excess of a million of parallel cores.
In the talk we will discuss the design of two software packages that attempt to exploit current and future high performance architectures: The first is HHG, a prototype multigrid package capable to solve finite element systems with more than 300 billion degrees of freedom. Time permitting, we will also discuss the design of Walberla, a parallel framework for simulating complex multiphase flows with fluid-structure interaction and with moving obstacles.

Keywords: Numerical solution of PDEs parallel computing

3. Tutorial

Zoltan - A Parallel Toolkit for Partitioning, Ordering, and Coloring

Erik Boman (Sandia National Labs, Scalable Algorithms Department, Albuquerque (NM), USA)

Zoltan is a well established parallel toolkit for partitioning and load balancing, and is used in production codes on thousands of processors. Over the last few years, Zoltan has evolved to include algorithms for other combinatorial scientific computing problems, in particular, graph/matrix ordering, and graph coloring. In this tutorial, we give an overview of the capabilities and algorithms in Zoltan. We show examples of partitioning for parallel data distribution, ordering for sparse matrix factorization, and coloring for parallel scheduling. Although Zoltan was designed for parallel computing, it can also be used in serial. Recently, Zoltan has been integrated into the Trilinos framework, and the Isorropia package provides an alternative matrix-based interface to Zoltan.

Keywords: Partitioning, load balancing, parallel computing, software

Joint work of: Boman, Erik; Catalyurek, Umit; Chevalier, Cedric; Devine, Karen

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2088

3.2 Tuesday, Feb 3, 2009

1. Plenary Talks

Combinatorial Problems in Automatic Differentiation

Paul Hovland (Argonne National Laboratory, Argonne (IL), USA)

We describe many of the combinatorial problems that arise in the implementation and use of automatic differentiation tools. Among the problems considered are strategies for exploiting chain rule associativity to combine partial derivatives in an optimal order, ways to exploit low rank and sparse substructure in derivative computations (scarcity), and techniques to minimize computational costs in the presence of limited memory (including optimal checkpointing strategies). We also briefly discuss the role of coloring in derivative computations, especially in the context of nonlinear partial differential equations.

Keywords: Automatic differentiation

Combinatorial problems in Numerical Linear Algebra

Iain Duff (Rutherford Appleton Lab. - Didcot, Oxforshire, UK)

Numerical linear algebra and combinatorial optimization are vast subjects; as is their interaction. In virtually all cases there should be a notion of sparsity for a combinatorial problem to arise. Sparse matrices, therefore, form the basis of the interaction of these two seemingly disparate subjects. As the core of many of today's numerical linear algebra computations consists of sparse linear system solutions, we will cover combinatorial problems, notions, and algorithms relating to those computations.
This talk is thus concerned with direct and iterative methods for sparse linear systems and their intercation with combinatorial optimization.
On the direct methods side, we discuss matrix ordering; bipartite matching and matrix scaling for better pivoting; task assignment and scheduling for parallel multifrontal solvers. On the iterative method side, we discuss preconditioning techniques including incomplete factor preconditioners (notion of level of fill-in), support graph preconditioners (graph embedding concepts), and algebraic multigrids (independent sets in undirected graphs).
In a separate part of the talk, we discuss methods that aim to exploit sparsity during linear system solution. These methods include block diagonalization of the matrix; efficient triangular system solutions for right-hand side vectors of single nonzero entries. Towards the end, we mention, quite briefly as they are topics of other invited talks, some other areas whose interactions with combinatorial optimization are of great benefit to numerical linear algebra. These include graph and hypergraph partitioning for load balancing problems, and colouring problems in numerical optimization. On closing, we compile and list a set of open problems.

Keywords: Numerical linear algebra

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2077

2. Contributed Talks

The CPR method and Beyond

Trond Steihaug (University of Bergen, Institute of Computer Science, Bergen, Norway)

In 1974 A.R. Curtis, M.J.D. Powell, and J.K.Reid published a seminal paper on the estimation of Jacobian matrices which was later coined as the CPR method. Central to the CPR method is the effective utilization of a priori known sparsity information.
Unfortunately, despite the enormous practical implication of this simple but elegant method there had not been any significant published research until around 10 years later when T.F. Coleman and J.J. Moráe showed that the CPR method could be further improved by regarding the column partitioning as a graph coloring problem and an efficient software implementation of the graph coloring CPR method, the DSM, written in F77 was made available for general use. It should take a further two decades before the optimal CPR method in its general form was characterized and the theoretical underpinning for the optimality was shown.
In this talk we trace the development of techniques for the estimation of Jacobian and Hessian matrices in the past 35 years since the inception of the CPR-method. We will consider the complexity of representation of the problem of Jacobian estimation as the theme in the presentation.

Joint work of: Steihaug, Trond; Hossain, Shahadat (Department of Mathematics and Computer Science, University of Lethbridge, Canada)

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2079

The Enabling Power of Graph Coloring Algorithms in Automatic Differentiation and Parallel Processing

Assefaw H. Gebremedhin (Purdue University, College of Science, Department of Computer Science, West Lafayette (IN), USA)

Combinatorial scientific computing (CSC) is founded on the recognition of the enabling power of combinatorial algorithms in scientific and engineering computation and in high-performance computing.
The domain of CSC extends beyond traditional scientific computing—the three major branches of which are numerical linear algebra, numerical solution of differential equations, and numerical optimization—to include a

range of emerging and rapidly evolving computational and information science disciplines.

Examples of the latter include computational biology, chemistry, and physics; computational climate and material sciences; statistical physics; bioinformatics; and large-scale data management and analysis.

Orthogonally, CSC problems could also emanate from infrastructural technologies for supporting high-performance computing.

Among such technologies are algorithmic tools for data partitioning, load balancing, and task scheduling in parallel computing, and tools for data and iteration reordering in irregular computation.

Despite the apparent disparity in their origins, CSC problems and scenarios are unified by the following common features:

(A) The overarching goal is often to make computation efficient—by minimizing overall execution time, memory usage, and/or storage space—or to facilitate knowledge discovery or analysis.

(B) Identifying the most accurate combinatorial abstractions that help achieve this goal is usually a part of the challenge.

(C) The abstractions are often expressed, with advantage, as graph or hypergraph problems.

(D) The identified combinatorial problems are typically NP-hard to solve optimally. Thus, fast, often linear-time, approximation (or heuristic) algorithms are the methods of choice.

(E) The combinatorial algorithms themselves often need to be parallelized, to avoid their being bottlenecks within a larger parallel computation.

(F) Implementing the algorithms and deploying them via software toolkits is critical.

This talk attempts to illustrate the aforementioned features of CSC through an example: we consider the enabling role graph coloring models and their algorithms play in efficient computation of sparse derivative matrices via automatic differentiation (AD) and in discovering concurrency in parallel scientific computing.

The talk focuses on efforts being made on this topic within the SciDAC Institute for Combinatorial Scientific Computing and Petascale Simulations (CSCAPES).

Aiming at providing overview than details, we discuss the various coloring models used in sparse Jacobian and Hessian computation, the serial and parallel algorithms developed in CSCAPES for solving the coloring problems, and case studies that demonstrate the efficacy of the coloring techniques in the context of optimization problems in engineering applications.

Implementations of our serial algorithms for the coloring and related problems in derivative computation are assembled and made publicly available in a package called ColPack.

Implementations of our parallel coloring algorithms are incorporated into and deployed via the load-balancing toolkit Zoltan.

ColPack has been interfaced with ADOL-C, an operator overloading-based AD tool that has recently acquired improved capabilities for automatic de-

tection of sparsity patterns of Jacobians and Hessians; sparsity pattern detection is the first step in derivative matrix computation via coloring-based compression.
Further information on ColPack and Zoltan is available at their respective websites, which can be accessed via http://www.cscapes.org.

Keywords: Graph coloring, sparse derivative computation, automatic differentiation, parallel computing

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2093

Randomized Heuristics for Exploiting Jacobian Scarcity

Andrew Lyons (Argonne National Laboratory, Argonne (IL), USA)

Griewank and Vogel introduced the notion of Jacobian scarcity, which generalizes the properties of sparsity and rank to capture a kind of deficiency in the degrees of freedom of the Jacobian matrix $F'(\mathbf{x})$. We describe new randomized heuristics that exploit scarcity for the optimized evaluation of collections of Jacobian-vector or Jacobian-transpose-vector products.

Keywords: Jacobian, scarcity, accumulation, directed acyclic graph, heuristic, random

Joint work of: Lyons, Andrew; Safro, Ilya

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2086

Reducing the Total Bandwidth of a Sparse Unsymmetric Matrix

John K. Reid (Rutherford Appleton Lab., Didcot, Oxfordshire, UK)

For a sparse symmetric matrix, there has been much attention given to algorithms for reducing the bandwidth. As far as we can see, little has been done for the unsymmetric matrix A, which has distinct lower and upper bandwidths l and u. When Gaussian elimination with row interchanges is applied, the lower bandwidth is unaltered while the upper bandwidth becomes $l + u$. With column interchanges, the upper bandwidth is unaltered while the lower bandwidth becomes $l+u$. We therefore seek to reduce $\min(l, u)+l+u$, which we call the **total** bandwidth.
Band solvers are simple, avoid the need for indirect addressing, and can make good use of modern cache and vector hardware. If the matrix can be reordered to have small bandwidths, a band solver is likely to be faster than any other.
We consider applying the reverse Cuthill-McKee algorithm [2] to $A + A^T$, to the row graph of A, and to the bipartite graph of A. In addition, we introduce a variation that may be applied directly to A.

We have also adapted the hill-climbing algorithm of Lim, Rodrigues and Xiao [3] for improving a given ordering to the unsymmetric case and have proposed a variant of the node-centroid algorithm of Lim et al for unsymmetric A.
When solving linear systems, if the matrix is preordered to block triangular form, it suffices to apply the band-reducing method to the blocks on the diagonal. We have found that this is very beneficial for many matrices from actual applications.
Numerical results for a range of practical problems are presented and comparisons made with other possibilities, including the recent lexicographical method of Baumann, Fleishmann and Mutzbauer [1].

Keywords: Bandwidth of a sparse unsymmetric matrix

[1] M. Baumann, P. Fleishmann, and O. Mutzbauer. Double ordering and fill-in for the LU factorization. *SIAM J. Matrix Analysis and Applications*, 25:630–641, 2003.

[2] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 24th National Conference of the ACM*, 1969.

[3] A. Lim, B. Rodrigues, and F. Xiao. A centroid-blimased approach to solve the bandwidth minimization problem. *Proceedings of the 37th Hawaii international conference on system sciences, IEEE*, 2004.

Multifrontal Multithreaded Rank-Revealing Sparse QR Factorization

Tim Davis (University of Florida, Department of Computer and Information Science and Engineering, Florida, USA)

SuiteSparseQR is a sparse QR factorization package based on the multifrontal method. Within each frontal matrix, LAPACK and the multithreaded BLAS enable the method to obtain high performance on multicore architectures. Parallelism across different frontal matrices is handled with Intel's Threading Building Blocks library. The symbolic analysis and ordering phase pre-eliminates singletons by permuting the input matrix into the form $[R_{11}\, R_{12}\, ;\, 0\, A_{22}]$ where R_{11} is upper triangular with diagonal entries above a given tolerance. Next, the fill-reducing ordering, column elimination tree, and frontal matrix structures are found without requiring the formation of the pattern of $A^T A$. Rank-detection is performed within each frontal matrix using Heath's method, which does not require column pivoting.
The resulting sparse QR factorization obtains a substantial fraction of the theoretical peak performance of a multicore computer.

Keywords: Sparse QR, multifrontal, shared-memory parallelism

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2078

Evaluation of Sparse LU Factorization and Triangular Solution on Multicore Platforms

Sherry Li (Lawrence Berkeley National Laboratory, Computational Research Division, Berkeley (CA), USA)

The Chip Multiprocessor (CMP) will be the basic building block for computer systems ranging from laptops to supercomputers.
New software developments at all levels are needed to fully utilize these systems. In this work, we evaluate performance of different high-performance sparse LU factorization and triangular solution algorithms on several representative multicore machines. We include both Pthreads and MPI implementations in this study, and found that the pthreads implementation consistently delivers good performance and a left-looking algorithm is usually superior.

Keywords: Multicore architecture, sparse LU, triangular solve

3. Tutorial

Scotch and PT-Scotch

Francois Pellegrini (LaBRI, Bordeaux, France)

The design of the Scotch library for static mapping, graph partitioning and sparse matrix ordering is highly modular, so as to allow users and potential contributors to tweak it and add easily new static mapping, graph bipartitioning, vertex separation or graph ordering methods to match their particular needs.
The purpose of this tutorial is twofold. It will start with a description of the interface of Scotch, presenting its visible objects and data structures.
Then, we will step into the API mirror and have a look at the inside: the internal representation of graphs, mappings and orderings, and the basic sequential and parallel building blocks: graph induction, graph coarsening which can be re-used by third-party software. As an example, we will show how to add a simple genetic algorithm routine to the graph bipartitioning methods.

Keywords: Scotch, graph algorithms, data structures

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2091

3.3 Wednesday, Feb 4, 2009

1. Plenary Talks

Present and Future of High-Performance Scientific Computing

Ruud van der Pas (Sun Microsystems, Menlo Park (CA), USA)

In this overview paper we start by looking at the birth of what is called “High Performance Computing” today. It all began over 30 years ago when the Cray 1 and CDC Cyber 205 “supercomputers” were introduced. This had a huge impact on scientific computing. A very turbulent time at both the hardware and software level was to follow. Eventually the situation stabilized, but not for long.
Today, there are two different trends in hardware architectures and have created a bifurcation in the market. On one hand the GPGPU quickly found a place in the marketplace, but is still the domain of the expert. In contrast to this, multicore processors make hardware parallelism available to the masses. Each have their own set of issues to deal with.
In the last section we make an attempt to look into the future, but this is of course a highly personal opinion.

Keywords: High-performance scientific computing

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2083

2. Contributed Talks

Hypergraph-Based Unsymmetric Nested Dissection Ordering for Sparse LU Factorization

Simplice Donfack (Paris Sud University, Laboratory for Computer Science, Paris, France)

In this paper we present HUND, a hypergraph-based unsymmetric nested dissection ordering algorithm for reducing the fill-in incurred during Gaussian elimination. HUND has several important properties. It takes a global perspective of the entire matrix, as opposed to local heuristics.
It takes into account the asymmetry of the input matrix by using a hypergraph to represent its structure. It is suitable for performing Gaussian elimination in parallel, with partial pivoting. This is possible because the row permutations performed due to partial pivoting do not destroy the column separators identified by the nested dissection approach. Experimental results on 27 medium and large size highly unsymmetric matrices compare HUND to four other well-known reordering algorithms.
The results show that HUND provides a robust reordering algorithm, in the sense that it is the best or close to the best (often within 10 % of all the other methods).

Keywords: Sparse LU-factorization, reordering techniques, hypergraph partitioning, nested dissection

Joint work of: Grigori, Laura; Boman, Erik; Donfack, Simplice; Davis, Timothy A.

Heuristic Initialization of Bipartite Matching Algorithms

Johannes Langguth (University of Bergen, Department of Informatics, Bergen, Norway)

It is a well established result that improved pivoting in linear solvers can be achieved by computing a bipartite matching between matrix and diagonal entries. With the availability of increasingly faster linear solvers, the speed of bipartite matching computations must keep up to avoid slowing down these solvers. Fast algorithms for bipartite matching which are usually initialized with simple heuristics have been known for a long time. However, the performance of these algorithms is largely dependent on the result of the heuristic. We present a new heuristic aimed at obtaining high quality matchings.
Consider a connected bipartite graph $G = (V, E)$ in which all vertices of degree 1 have been matched and removed, which can be done easily. Our algorithm is based on exploiting two subgraphs G_A and G_B of G where $G_A = (V_A, E_A)$ is induced by all edges incident to a vertex of degree 2, and

$G_B = (V_B, E_B)$ is induced by all edges not incident to a vertex of degree 2. We use a minimum degree based heuristic to successively build a good matching in G_B, removing all matched vertices and their incident edges. Each time we match a vertex in $V_A \cap V_B$ this way, its adjacent vertices of degree 2 drop to degree 1 and can be matched immediately, which in turn implies that new vertices of degree 1 one appear. This continues until an entire connected component of G_A is matched.
We show how this strategy can be implemented in an efficient way and present experimental data on randomly generated and real world instances, comparing the heuristics running time and solution quality to other matching heuristics, as well as their impact on the running time of various exact bipartite matching algorithms when used as an initialization. Because some matrices turn out to be very hard to tackle with simple heuristics, exact algorithms that use them as initializations will generally perform poorly on these. Thus, for diverse sets of instances our improved initialization results in an improved average running time.

Keywords: Bipartite matching, heuristic

3. Tutorials

ADOL-C

Andrea Walther (TU Dresden, Department of Mathematics, Institute of Scientific Computing, Dresden, Germany)

The provision of derivatives for a function $f : R^n \to R^m, y = f(x)$ defined by an evaluation procedure in a high level computer language like Fortran or C forms an important task for numerous applications comprising for example optimization, parameter estimation, and data assimilation.
The technique of algorithmic differentiation (AD) [1] offers an opportunity to provide derivative information of any order for the given code segment by applying the chain rule systematically to statements of computer programs. The package ADOL-C [2] uses operator overloading for differentiating automatically C and C++ codes. For this purpose, each double variable on the path from the independent variables x to the dependents y is replaced by a variable of the type adouble introduced by ADOL-C. During an evaluation of the function to be differentiated the usage of the new data-type adouble causes the generation of an internal function representation. Afterwards several drivers allow a very flexible choice of the mode and order of differentiation to be performed.
Naturally, this approach works also for codes based on classes, templates and other C++-features. The resulting derivative evaluation routines may be called from C, C++, Fortran, or any other language that can be linked with C.
We will demonstrate the installation of ADOL-C and illustrate the application of ADOL-C using basic test suite. After this introduction we will

focus on advanced algorithmic differentiation provided by ADOL-C. This will cover the exploitation of sparsity by a combination of AD and graph coloring as well as the differentiation of time integrations using checkpointing facilities.

[1] A. Griewank and A. Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. *SIAM*, Philadelphia, PA, 2nd edition, 2008.

[2] A. Griewank, D. Juedes, and J. Utke. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. Math. Softw.*, 22:131–167, 1996.

Keywords: ADOL-C, algorithmic differentiation

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2084

NLP Solver Ipopt

Andreas Wächter (IBM T.J. Watson Research Center, Department of Mathematical Science, Yorktown, USA)

Ipopt [1] is an open source implementation of an algorithm for large-scale nonlinear nonconvex optimization, available at COIN-OR. It is a primal-dual interior point method with a line search procedure, ensuring convergence to local solutions under appropriate assumptions. A recently added feature is the possibility to use an iterative linear solver for the computation of search directions, allowing the solution of optimization problems arising in PDE-constrained optimization problems.
This tutorial will be focusing on practical issues. The participants will be giving a precompiled version of the program, but the installation procedure will be briefly demonstrated. We will present some details of the underlying mathematical algorithm, with the emphasis on aspects that help to understand the practical behavior of the method. The algorithmic options and the output of the program will be explained.
In practical exercises we will show how Ipopt can be used from the modeling language AMPL, via a Matlab interface, or directly from programming (FORTRAN, C, or C++) code. We will touch on issues of good modeling and what tricks are available in order to improve convergence. The participants will have the chance to experiment with provided examples.
[1] Homepage Ipopt: `http://projects.coin-or.org/Ipopt`.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2089

3.4 Thursday, Feb 5, 2009

1. Plenary Talks

Emerging Applications in Combinatorial Scientific Computing

David A. Bader (Georgia Institute of Technology, College of Computing, Computational Science and Engineering Division, Atlanta (GA), USA)

Computation is now recognized as the third pillar to scientific discovery; and combinatorial techniques are proving useful in solving real-world challenges in traditional and emerging computational sciences such as chemistry, biology, and medicine, as well as applications in national security. Yet they pose serious challenges for parallel machines due to non-contiguous, concurrent accesses to global data structures with low degrees of locality. For example, few parallel graph algorithms outperform their best sequential implementation due to long memory latencies and high synchronization costs. In this talk, we discuss emerging real-world applications that can be modeled using algorithms that process massive spatio-temporal complex networks, such as power grid stability, inference of gene function in protein interaction networks, and cancer research, as well as homeland security and national defense.

Our experimental studies use real-world graph instances with billions of elements and demonstrate superior performance on highly parallel systems such as the Cray XMT multithreaded architecture and the Sun's Niagara 2 Maramba system. We will also describe SNAP (Small-world Network Analysis and Partitioning), an open-source graph framework that we have developed for the exploration of massive complex networks and present new ideas on the exploration of the dynamic structure of massive spatio-temporal networks with billions of entities, such as understanding the genesis and dissipation of communities, allegiance switching, and source detection.

Keywords: Spatio-temporal complex networks

Combinatorial and Scientific Computing Approaches to Modern Large-Scale Data Analysis

Michael Mahoney (Stanford University, Department of Mathematics, Stanford (CA), USA)

In recent years, ideas from statistics and scientific computing have begun to interact in increasingly sophisticated and fruitful ways with ideas from the theory of algorithms to aid in the development of improved worst-case algorithms for very large-scale scientific and Internet data analysis problems. Although these data applications differ in several ways from traditional applications of combinatorial scientific computing (CSC), many of the ideas underlying them have a similar flavor, and this represent a nice venue for CSC researchers to make nontrivial contributions to real large-scale data analysis problems. After providing an overview of these ideas, two specific examples will be described in detail.

For the first example, consider the problem of selecting a “good” set of exactly k columns from an arbitrary $m \times n$ matrix. By randomly sampling columns from the matrix according to an importance sampling probability distribution that depends on the diagonal elements of the projection matrix onto the top-k right singular subspace, one can prove improved quality-of-approximation bounds in worst-case. In addition, one can also obtain improved results (i.e., faster and/or better and/or without the need for additional information such as ancestor history or haplotype phase information) in applications such as structure identification from large-scale DNA SNP data.

For the second example, recall that the concept of regularization, often included as a smoothness assumption or a norm bound, is implicitly included in certain spectral approximation algorithms for intractable combinatorial problems. For example, for the minimum conductance cut problem, widely-used in clustering and community detection in large sparse graphs, recently-developed local spectral methods can be used not only to obtain an efficient approximation algorithm but also to “probe” the graph for regularized clusters that are more meaningful in applications. This leads to a very precise characterization of the presence and absence of small-scale and large-scale community structure in modern social and information networks, a ubiquitous problem in many Internet applications, that currently-existing network generation models fail to reproduce in even a qualitative manner.

Keywords: Large-scale data analysis

2. Contributed Talks

Parallel Short Sequence Mapping for High Throughput Genome Sequencing

Umit Catalyurek (Ohio State University, Department of Biomedical Informatics, Columbus (OH), USA)

With the advent of next-generation high throughput sequencing instruments, large volumes of short sequence data are generated at an unprecedented rate. Processing and analyzing these massive data requires overcoming several challenges. A particular challenge addressed in this abstract is the mapping of short sequences (reads) to a reference genome by allowing mismatches. This is a significantly time consuming combinatorial problem in many applications including whole-genome resequencing, targeted sequencing, transcriptome/small RNA, DNA methylation and ChiP sequencing, and takes time on the order of days using existing sequential techniques on large scale datasets. In this work, we introduce six parallelization methods each having

different scalability characteristics to speedup short sequence mapping. We also address an associated load balancing problem that involves grouping nodes of a tree from different levels. This problem arises due to a trade-off between computational cost and granularity while partitioning the workload. We comparatively present the proposed parallelization methods and give theoretical cost models for each of them. Experimental results on real datasets demonstrate the effectiveness of the methods and indicate that they are successful at reducing the execution time from the order of days to under just a few hours for large datasets.
To the best of our knowledge this is the first study on parallelization of short sequence mapping problem.

Keywords: Genome sequencing, sequence mapping

Joint work of: Bozdag, Doruk; Catalyurek, Umit

Fast AMD orderings for Matrices with Some Dense Rows

Jennifer Scott (Rutherford Appleton Lab., Didcot, Computational Science and Engineering Department, Oxfordshire, UK)

The efficiency of sparse direct solvers for the solution of symmetric linear systems $Ax = b$ is dependent upon the order in which the variables are eliminated, that is, the order in which the pivots are selected. Many solvers include a preordering step that aims to use information on the sparsity pattern of A to find a permutation so that, if the pivots are chosen in order from the diagonal of the permuted matrix, the computed factors are sparser than if the pivots were chosen in order from the diagonal of the original matrix.
The problem of finding a permutation that results in the smallest amount of fill-in for a Cholesky factorization is NP-complete and so heuristics are used to find a good ordering.
Two main classes of methods are widely used: those based on nested dissection and those based on the minimum degree algorithm.
Nested dissection is often the method of choice for very large problems but it can be more expensive than the most efficient implementations of the minimum degree algorithm, which is preferred for more modest size problems and for very sparse problems. Currently, the most successful variant of the minimum degree algorithm is the approximate minimum degree algorithm (AMD) [1][2].
Although AMD generally produces high quality orderings, it can be prohibitively expensive when A has some dense (or almost dense) rows and columns.
AMD uses the undirected graph of the matrix and selects each node in turn to have minimum (approximate) degree. Once a node is selected, it is eliminated from the graph and replaced by adding edges between its neighbours so that the neighbours become a clique. If a row is full, the corresponding node

will always be adjacent to the eliminated node so its adjacency list has to be scanned and updated, requiring $\mathcal{O}(n^2)$ operations for a problem with n variables. This results in a very high run time.
Recently a number of variants of AMD have been proposed that aim to overcome this limitation [3][4][5].
We compare the performance of these variants on a range of problems from practical applications and illustrate how they can sometimes perform poorly. This leads us to propose a new variant [6] that offers both speed and robustness.

Keywords: AMD ordering

[1] P. Amestoy, T. Davis, and I. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Appl.*, 17 (1996), pp. 886 – 905.

[2] P. Amestoy, T. Davis, and I. Duff. Algorithm 837: AMD, an approximate minimum degree ordering algorithm, *ACM Trans. Mathematical Software*, 30 (2004), pp. 381 – 388.

[3] P. R. Amestoy, H. S. Dollar, J. K. Reid, and J. A. Scott. An approximate minimum degree algorithm for matrices with dense rows. *Technical Report RAL-TR-2007-020*, Rutherford Appleton Laboratory, 2007.

[4] J. Carmen. An AMD preprocessor for matrices with some dense rows and columns. http://www.netlib.org/linalg/amd/amdpre.ps.

[5] T. A. Davis. Direct methods for sparse linear systems, volume 2 of Fundamentals of Algorithms. *SIAM*, Philadelphia, PA, 2006.

[6] H. S. Dollar and J. A. Scott. A note on fast approximate minimum degree orderings for symmetric matrices with some dense rows. *Technical Report RAL-TR-2007-022*, Rutherford Appleton Laboratory, 2007.

A Nearly-Linear Time Algorithm for Approximately Solving Linear Systems in a Symmetric M-Matrix

Samuel Daitsch (Yale University, Department of Computer Science, New Haven (CT), USA)

A symmetric M-matrix is a positive-definite matrix with non-positive off-diagonal entries. If M is a symmetric M-matrix, then there exists a factorization $M = AA^T$, where A has at most 2 non-zero entries per column. In this talk we present an algorithm for approximately solving a linear system in a symmetric M-matrix M, when we are given such a factorization.
The quality of approximation is measured in the matrix norm, defined as $\|\mathbf{x}_M\| = \sqrt{\mathbf{x}^T M \mathbf{x}}$.

The algorithm finds a solution within relative error ϵ in the matrix norm in expected time $\mathcal{O}(m \log \frac{\kappa}{\epsilon})$, where m is the number of non-zeros in M, and κ is the condition number of M.
The algorithm works by reducing the problem to solving $\mathcal{O}(\log n)$ linear systems in diagonally-dominant matrices. which can each be solved in nearly-linear time using the algorithm of Spielman and Teng.
An interesting example of where symmetric M-matrix systems occur is in solving generalized network flow problems. Using our M-matrix algorithm in an interior-point method yields the fastest known algorithm for a certain subset of such problems.

Joint work of: Daitch, Samuel; Spielman, Daniel

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2080

Full Paper:
http://arxiv.org/abs/0803.0988

Star-P: A Tool for Interactive Numerical and Combinatorial Computing

John Gilbert (University of California, Department of Computer Science, Santa Barbara (CA), USA)

This tutorial presents Star-P, a parallel implementation of the Matlab programming language, and KDT, a suite of tools designed for combinatorial, numerical, and hybrid scientific computing in Star-P.
We are building a flexible, scalable, interactive environment for high-performance computation on discrete structures that can be used both as a rapid-prototyping tool for exploring and experimenting with different approaches to analysis, and as a scalable system for performing analysis on real, dynamic, discrete data.
High-level scientific language systems such as Matlab, Python, and R have become quite popular among computational scientists and engineers.
These systems include interactive, interpreted environments for rapid prototyping, debugging, and data visualization, as well as support for matrix data types and comprehensive numerical libraries. As computing hardware continues to become cheaper relative to programmer effort, high-level interpreted languages are increasingly used for serious large-scale scientific computation.
These high-level systems typically provide a rich infrastructure for numerical algorithms, but support graphs and graph operations only as an afterthought. Our work on KDT unifies numerical and combinatorical computation in terms of a common infrastructure for sparse arrays, graphs, and matrices. Graphs have long been used to analyze sparse matrices, and many sparse matrix algorithms are built with graph algorithms.
We turn this relationship around, showing that graph algorithms can be efficiently designed and implemented using methods and systems originally

developed for sparse linear algebra. The sparse matrix abstraction allows structured representation of irregular data structures and access patterns in many parallel combinatorial applications.

Keywords: Combinatorial, numerical, hybrid scientific computing

Joint work of: Gilbert, John; Reinhardt, Steven; Shah, Viral

3. Tutorials

Automatic Differentiation with OpenAD and Combinatorial Problems

Jean Utke (Argonne National Laboratory, Argonne (IL), USA)

Computing derivatives of numerical models $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ given as a computer program P is an important but also compute-intensive task. Automatic differentiation (AD) [1] provides the means to obtain such derivatives. OpenAD [3] implements AD as a source transformation applied to Fortran programs.

We will give an overview of the components of OpenAD, illustrate how to install the tool and verify the installation using some examples from the regression test suite. Given the complexity of implementing a source transformation tool vs. an operator overloading library we will briefly compare the benefits of either approach. Considering certain sets of semantically valid transformations we can impose efficiency criteria on the transformed program. Consequently there are various choices given to the user of the AD tool. While the efficiency considerations may not be important for small scale problems they typically become a crucial aspect of differentiating a large scale application.

A major goal of the tutorial is to characterize the choices offered by OpenAD, to provide a rationale for the decisions and illustrate the respective use patterns in OpenAD. Among these choices are forward vs. reverse mode, split/joint/hybrid reversal and the implied checkpointing schemes. While the above questions reflect the coarse-grained choices to be made for a given transformation, there is also a set of problems at the level of the computational graph most of which are combinatorial in nature. In OpenAD, considerable effort [2] has been put into the efficient preaccumulation of local Jacobians. It relies on heuristics that operate on a structural representation of the computational graphs adorned only with generic data about the kind of certain partial derivatives on the graph edges. Because the elimination heuristics are implemented in their own component and are open source one can experiment with new heuristics in the context of practical test cases with relative ease.

A second goal of the tutorial is to illustrate how one might go about implementing new heuristics and evaluate the results in the OpenAD framework. The last part of the tutorial will be devoted to other possible uses of OpenAD such as non-smooth model tracing and efficient computation of higher-order tensors using generated libraries from the Rapsodia tool [4].

Keywords: Automatic differentiation, combinatorial problem, tool tutorial

[1] Andreas Griewank and Andrea Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Number 105 in Other Titles in Applied Mathematics. *SIAM*, Philadelphia, PA, 2nd edition, 2008.

[2] Andrew Lyons and Jean Utke. On the practical exploitation of scarsity. *In Christian H. Bischof, H. Martin Bucker, Paul D. Hovland, Uwe Naumann, and J. Utke, editors, Advances in Automatic Differentiation, volume 64 of Lecture Notes in Computational Science and Engineering*, pages 103–114, Springer, Berlin, 2008.

[3] OpenAD website. http://www.mcs.anl.gov/openad.

[4] Rapsodia website. http://www.mcs.anl.gov/rapsodia.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/2095

3.5 Posters

On Out-of-Core Solution of Sparse Linear Systems for Sparse Right Hand-Side Vectors

Bora Ucar (ENS Lyon, Lyon, France)

We consider efficient solution of sparse linear systems for multiple sparse right hand-side (rhs) vectors, each containing only a single nonzero. We address the problem in the context of out-of-core direct solvers, where the factors are stored on disk.

An application which requires solving such a linear system is the computation of the variance matrix (V), or some elements of it, under the standard linear regression model. In this setting, V is the inverse of a symmetric positive-definite matrix, say A. Suppose we are concerned with the computation of the ith diagonal element of V, given a suitable factorization of A. This can be cast as a linear system with A and the ith column of the identity matrix as the rhs vector. The requested entry can be found by traversing the nodes of the assembly tree from a node to a root node (forward substitution), and then by descending from that root to the starting node (a partial backward substitution).

Given a large number of right-hand side vectors, the computations should proceed in epochs—a time slot in which solutions for a reasonable number of rhs vectors take place. An obvious problem then arises: how to partition the rhs vectors into sets in order to optimize system resources. Since we are in the out-of-core context, we aim to minimize the total cost of the factor loading operations. We have successfully modelled the partitioning problem for the application mentioned above in terms of the hypergraph partitioning problem. We will present the model and a few results.

Keywords: Direct methods, hypergraph partitioning, out-of-core solution

Joint work of: Ucar, Bora; Amestoy, Patrick A.; Duff, Iain S.; Slavova, Tzvetomila

Assessing an approximation algorithm for the minimum fill-in problem in practice

Martin Bücker (RWTH Aachen, DE)

During a Cholesky factorization, a zero matrix element may become nonzero, a phenomenon called fill-in. This can increase the storage requirement and computing time by orders of magnitude. It is well-known that a permutation of the coefficient matrix influences the size of the fill-in. So, one is interested in the minimum fill-in problem, finding a permutation that leads to the least possible fill-in over all permutations.

We investigate an implementation of an approximation algorithm for the minimum fill-in problem. The algorithm has some degree of freedom since it is composed of several subtasks for which one can choose between different algorithms. The goal of the present work is to study the impact of theses components and carefully examine the practicability of the overall approximation algorithm by a set of numerical examples.

Keywords: Sparse linear algebra

Joint work of: Bücker, H. Martin; Lülfesmann, Michael; Rasch, Arno

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2119

Large-Scale PDE-Constrained Optimization in Biomedical Hyperthermia Treatment Planning

Olaf Schenk (University of Basel, Department of Computer Science, Basel, Switzerland)

Biomedical hyperthermia cancer treatment is a promising therapeutical option in oncology . Various types of cancer can be treated by heating the tumor to about 45^0C using non-ionizing radiation (microwaves). It makes the tumor more susceptible to an accompanying radio or chemotherapy. The optimal hyperthermia treatment planning can be formulated as a nonlinear optimization problem, in which the Pennes Bio-heat equations appear as important constraints — a mathematical task known as *PDE-constrained optimization*. Solving the PDE constrained optimization problem presents a frontier problem in scientific computing. The size, complexity and infinite-dimensional nature of PDE-constrained optimization problems present significant challenges for general-purpose optimization algorithms and, typically, Tikhonov regularization, iterative solvers, preconditioning, inexactness and parallel implementations are necessary to cope with the numerical challenges.

Interior-point optimization methods and software are used to solve the discretized nonconvex PDE-constrained optimization problem. By using a full space approach, optimality and feasibility are reached simultaneously. The infinite-dimensional differential equations are discretized, and only linearized state equations have to be solved in every optimization step, which can improve the overall performance tremendously. The ultimate advantage of the full space approach is that the underlying Newton method typically can converge very fast and even superlinearly close the the solution.

The combinatorial scientific computing task is related to the solution of highly ill-conditioned symmetric KKT matrices that arise in the interior-point optimization process.

The solution process is stabilized by using variants of symmetric weighted matchings. These weighted matchings compute a static approximation of the pivoting order in Gaussian elimination. The subsequent fast algorithms for sparse matrices triggered a dramatic improvement of modern sparse direct solvers. In particular, borrowing from classical Bunch and Kaufmann pivoting for symmetric matrices, sparse direct solvers for symmetric indefinite systems have been developed , that are orders of magnitude faster and more memory efficient than previous methods. Their positive impact on preconditioning has also been recognized both in the unsymmetric and in the symmetric highly indefinite case.

For the KKT matrices we use a fast and robust algebraic multilevel preconditioner, which combines two key ideas: a graph–pivoting strategy based on weighted graph matchings and an inverse based coarsening process, which drives the algebraic multilevel factorization. By using the symmetrized version of maximum weight matchings, we are able to optimize several large-scale three-dimensional examples of PDE-constrained optimizations in the full space of states, control and adjoint variables with equality and non-equality constraints.

The largest nonconvex optimization problem from three-dimensional PDE-constrained optimization approach has more than thousands of control variables and one million state variables with both lower and upper bound.

Keywords: PDE-Constrained optimization, weighted graph matchings, interior-point optimization, biomedical application

Joint work of: Christen, Matthias; Neufeld, Esra; Schenk, Olaf; Wächter, Andreas

Adaptive Cycling and Block Smoothers in Aggregation AMG

Frank Hülsemann (EDF, Clamart, France)

Aggregation based algebraic multigrid methods with piecewise constant restriction and prolongation operators allow for a fast construction of the operator hierarchy but without further measures, they are slower to converge than other AMG variants with more elaborate set up routines.

However, different options to improve the performance of the non-smoothed aggregation approach are available. Y. Notay proposed the K-cycle in conjunction with non-smoothed aggregation which adds few steps of a flexible Krylov method at selected levels.

This poster describes an alternative approach which also includes some Krylov-like projection but complements it with an adaptive cycling strategy and block smoothers of Gauss-Seidel or column projection type.

Beyond smoothing, the block column projection method can be used to detect strong connections between unknowns in non-symmetric matrices.

The proposed method is tested on sparse symmetric and non-symmetric positive definite matrices arising from compactly supported discretizations in one, two and three spatial dimensions. The numerical results show that the proposed version is clearly superior to the original non-smoothed aggregation scheme.

However, the results also illustrate that in the case of anisotropies in either the underlying grid or the material parameters, the robustness of the approach needs to be improved further.

Keywords: Aggregation AMG, block smoothers

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2094

Scalable Stencil Computations on Modern Chip-Multithreading Architectures

Matthias Christen (University of Basel, Department of Computer Science, Basel, Switzerland)

Novel micro-architectures such as the Cell Broadband Engine Architecture (Sony, Toshiba, IBM) and general purpose graphics processing units are just two examples for the hardware diversity found in nowadays high performance computing landscape. Modern micro-architectures are typically inherently highly parallel and therefore offer fantastic peak performances, which make them attractive platforms for compute-intensive simulations.

Partial differential equation (PDE) solvers constitute a large fraction of scientific applications in such diverse areas as heat diffusion, astrophysics, biomechanics, electromagnetics, and fluid dynamics. These applications are often implemented using iterative finite-difference or finite-volume techniques that sweep over a spatial grid, performing nearest neighbor computations called *stencils.* In a stencil operation, each point in a multidimensional grid is updated with weighted contributions from a subset of its neighbors in both time and space — thereby representing the coefficients of the PDE for that grid point. These operations are then used to build solvers that range from simple Jacobi iterations to complex multigrid and adaptive mesh refinement methods. Stencil calculations perform global sweeps through data structures that are typically much larger than the capacity of the available data caches. In addition, the amount of data reuse is limited to the number of points in a stencil, which is typically small. As a result, these computations generally achieve a low fraction of theoretical peak performance, since data from main memory cannot be transferred fast enough to avoid stalling the computational units on modern microprocessors. Reorganizing these stencil calculations to take full advantage of memory hierarchies has been the subject of much investigation over the years. These have principally focused on tiling optimizations that attempt to exploit locality by performing operations on cache-sized blocks of data before moving on to the next block.

On this poster we give an architectural overview over the Cell processor and NVIDIA GPUs and show some performance numbers that were obtained for a stencil code used in the simulation of thermal behavior of the human body during biomedical hyperthermia cancer treatment. Hyperthermia is a promising treatment modality for various types of cancer. The technique involves heating the tumour with electromagnetic fields, generally using antenna arrays to focus the energy. In planning the therapy, the therapeutically optimal antenna parameters for the applicator have to be determined for each patient individually given the patient's geometry. The temperature distribution is predicted by solving the 3D Pennes bio-heat transfer equation. Although this can be a demanding task, a planning and simulation tool can greatly help clinical researchers to model and simulate the medical treatment.

We are addressing the practical concerns of fitting into a clinician's standard work-flow, and the question of how to use optimal algorithms and relevant HPC architectures to maximize a biomedical application performance.

Keywords: Stencil computation, Cell processor, GPU, bandwidth-saving algorithm, hyperthermia cancer treatment

Joint work of: Christen, Matthias; Schenk, Olaf

Weighted Aggregation for Multi-Level Graph Partitioning

Cedric Chevalier Sandia National Labs, Department of Energy, Albuquerque (NM), USA)

Graph partitioning is a well-known optimization problem of great interest in theoretical and applied studies. Since the 1990s, many multilevel schemes have been introduced as a practical tool to solve this problem. A multilevel algorithm may be viewed as a process of graph topology learning at different scales in order to generate a better approximation for any approximation method incorporated at the uncoarsening stage in the framework. In this work we compare two multilevel frameworks based on the geometric and the algebraic multigrid schemes for the partitioning problem.

Keywords: Graph partitioning, multilevel, coarsening, weighted aggregation

Joint work of: Chevalier, Cedric; Safro, Ilya

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2082

See also: Learning and Intelligent Optimization - LION, 3 Jan 14-18, 2009, Trento, Italy

Structure-Preserving Multilevel ILUs for KKT Systems

Matthias Bollhöfer (TU Braunschweig, Faculty of Mathematics and Computer Science, Braunschweig, Germany)

We present a multilevel incomplete factorization approach for systems of type

$$K = \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix},$$

where A is nonsingular real symmetric matrix. We discuss a multilevel approach based on two aspects.

1. On each level, every coarse grid matrix will again be a KKT system like K. To do this, a pivoting strategy based on [2] is used, where 2×2 pivots are built using one entry from A together with one entry from the second block. Single entries from the A block are eliminated as a 1×1 pivot only, if no connection to an element in B is present. A preprocessing phase is run prior to elimination to transform K into the desired form.

   However, if one of these two choices of pivots is not available, the associated row and column are permuted to the Schur complement.
2. Beside structure-preserving elimination, the ILU is supplemented with an inverse-based [1] strategy, where pivots (1×1 or 2×2) are skipped whenever the scaled lower triangular factor L exceeds $\|L^{-1}\| > \kappa$ for some predefined bound κ.

For the symmetric indefinite case the second aspect has been discussed in [3] and furthermore it has been successfully applied ton non-convex optimization problems [4] without explicitly using the KKT-structure.

We will also discuss the impact of this kind of factorization with respect to the eigenvalue perturbation. The latter is important in particular for preserving the inertia.

Keywords: KKT systems, multilevel ILU, iterative methods

Joint work of: Bollhöfer, Matthias; Wubs, Fred (Groningen University)

[1] M. Bollhöfer, Y. Saad. Multilevel preconditioners constructed from inverse–based ILUs. *SIAM J. Sci. Comput.* 27:5, 1627–1650, 2006.

[2] Arie C. De Niet and Fred W. Wubs. Numerically stable LDL^T-factorization of $\mathcal{F}$-type saddle point matrices. *IMA Journal of Numerical Analysis*, 2008.

[3] O. Schenk, M. Bollhöfer, R. A. Römer. On large scale diagonalization techniques for the Anderson model of localization. *SIAM* Review 50 (2008), 91–112.

[4] O. Schenk, A. Wächter, and M. Weiser. Inertia Revealing Preconditioning For Large-Scale Nonconvex Constrained Optimization. *SIAM J. Sci. Comput.*, 2008 (to appear).

Task Repartitioning under Data Replication with Memory Constraints

Cevdet Aykanat (Bilkent University, Department of Computer Engineering, Ankara, Turkey)

Many parallel scientific computing applications require repeating the same computation over a problem instance with different parameters. In general, the quality of the initial task-to-processor or data-to-processor mapping in such applications tends to deteriorate in terms of computational load balance and communication requirements, as the computational structure of the problem or application parameters change between successive computational instances. A promising solution to this problem is to rebalance the load distribution in the parallel system as needed by rearranging the assignment of tasks to processors via a process known as repartitioning (remapping). Novel repartitioning models are essential for efficient parallelization.
The success of a repartitioning model depends on its ability to rebalance the load distribution as well as to minimize the overheads introduced due to the repartitioning process itself. Although it is problem-dependent, most typical repartitioning overheads are due to task migration, data replication, and repartitioning computations. Recently, successful combinatorial models [1], [2], [3], which are based on graph and hypergraph partitioning by fixed vertices, are proposed as solutions to the repartitioning problems arising in different types of applications. In all three models, the computational structure of an underlying application is represented by a graph/hypergraph, where vertices represent tasks and edges/hyperedges represent the interaction between/ among the tasks.
In [1] and [3], vertices are fixed to the parts according to the initial task-to-processor mapping, whereas in [2], the hyperedges, which represent the data primitives, are fixed to the parts according to the initial data-to-processor mapping. The fixed vertex formulations adopted in these models encapsulate the cost of task and/or data migration. The focus of this work is on parallel scientific

computing applications in which similar type of computations are successively repeated over the same dataset instance for many times with different parameters. There is no dependency between tasks and the only reason for inter-task interaction is the existence of data primitives that are inputs to several tasks. Both computational structure and expected task execution times may change during successive computational instances. Change in computational structure means change in the data primitive requirements of tasks.

Since the individual processors of the parallel system have a limited storage capacity, we can reserve a limited amount of storage for holding replicas at each processor. For the parallelization of a particular computational stage, the repartitioning model should utilize the replication pattern of the previous computational stage(s) for reducing the communication overhead due to the data replication requirement of the current stage. We propose a two-phase model for solving this problem.

The hypergraph-partitioning-based model proposed for the first phase is an enhanced version of our previous model in [2] and it aims to minimize the total message volume that will be incurred due to the replication/migration of input data while maintaining balance on computational and receive-volume loads of processors. The network-flow-based model proposed for the second phase is an adaptation of [4] and it aims to minimize the maximum message volume handled by processors via utilizing the flexibility in assigning send-communication tasks to processors, which is introduced by data replication. The validity of our proposed model is verified on parallelization of a direct volume rendering algorithm.

Keywords: Task repartitioning, data replication, hypergraph partitioning with fixed vertices, assignment flow network

Joint work of: Aykanat, Cevdet; Erkan, Okuyan; B. Barla, Cambazoglu

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2090

[1] C. Aykanat, B.B. Cambazoglu, F. Findik, and T. Kurc. Adaptive Decomposition and Remapping Algorithms for Object-Space-Parallel Direct Volume Rendering of Unstructured Grids. *J. Parallel and Distributed Computing*, vol. 67, no. 1, pp. 77–99, Jan. 2007.

[2] B.B. Cambazoglu and C. Aykanat. Hypergraph-Partitioning-Based Remapping Models for Image-Space-Parallel Direct Volume Rendering of Unstructured Grids. *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 1, pp. 1–14, Jan. 2007.

[3] U.V. Catalyurek, E.G. Boman, K.D. Devine, D. Bozdag, R.T. Heaphy, and L.A. Riesen. Hypergraph-based Dynamic Load Balancing for Adaptive Scientific Computations. *In Proc. IPDPS'07*, pp. 1–11, Mar. 2007.

[4] A. Pinar and B. Hendrickson. Improving Load Balance with Flexibly Assignable Tasks. *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 10, pp. 956–965, Oct. 2005.

Low-Memory Tour Reversal in Directed Graphs

Viktor Mosenkis (LuFG Informatik 12, RWTH Aachen University, Germany)

We consider the problem of reversing a *tour* $(i_1, i_2, \ldots, i_l)$ in a directed graph $G = (V, E)$ with positive integer vertices V and edges $E \subseteq V \times V$, where $i_j \in V$ and $(i_j, i_{j+1}) \in E$ for all $j = 1, \ldots, l-1$. The tour can be processed in last-in-first-out order as long as the size of the corresponding stack does not exceed the available memory. This constraint is violated in most cases when considering control-flow graphs of large-scale numerical simulation programs. The tour reversal problem also arises in adjoint programs used, for example, in the context of derivative-based nonlinear optimization, sensitivity analysis, or other, often inverse, problems. The intention is to compress the tour in order not to run out of memory. As the general optimal compression problem was proven to be NP-hard and big control-flow graphs results from loops in programs we do not want to use general purpose algorithms to compress the tour. We want rather to compress the tour by finding loops and replace the redundant information by proper representation of the loops.

Keywords: Directed graph, tour reversal, offline algorithm, dynamic programming, online algorithm, control-flow reversal, adjoint program

Joint work of: Mosenkis, Viktor; Naumann, Uwe

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2092

What Color is the Non-Constant Part of Your Jacobian?

Ebadollah Varnik (LuFG Informatik 12, RWTH Aachen University, Germany)

The context of our work is *Automatic Differentiation* [1](AD) of vector functions $F : I\!R^n \supseteq D \rightarrow I\!R^m, \quad \mathbf{y} = F(\mathbf{x})$ implemented as computer programs. The Jacobian $F' = F'(\mathbf{x})$ can be computed using vector forward [reverse] mode AD as $\dot{Y} = F' \cdot \dot{X}$ [$\bar{X} = F'^T \cdot \bar{Y}$] with $\dot{X}$ [$\bar{Y}$] initialized to the identity $I \in R^{n \times n}$ [$R^{m \times m}$]. Sparsity can be exploited to reduce the runtime and the storage requirement for Jacobian accumulation. Therefore, the compressed [transposed] Jacobian is computed as $I\!R^{m \times \dot{p}} \ni \dot{B} = F' \cdot \dot{S}$ [$I\!R^{n \times \bar{p}} \ni \bar{B} = F'^T \cdot \bar{S}$], where the *seed matrix* [3] $\dot{S} \in I\!R^{n \times \dot{p}}$ [$\bar{S} \in I\!R^{m \times \bar{p}}$] is the result of partitioning of the Jacobian's columns [rows] into $\dot{p}$ [$\bar{p}$] groups of *structurally orthogonal* columns [rows]. The minimization of $\dot{p}$ [$\bar{p}$] requires the solution of a coloring problem [2] on some graph representation $G(F')$ (e.g. bipartite graph) of F'. A recovery step is needed to compute the nonzero entries of the Jacobian from $\dot{B}$ [$\bar{B}$]. The goal is to minimize $\dot{p}$ [$\bar{p}$]. Therefore, we consider the Jacobian $F' = F'_v + F'_c$ as the sum of its *variable* F'_v and *constant* F'_c entries. We compute F'_c and the sparsity pattern of F'_v and obtain $\tilde{\dot{S}} \in I\!R^{m \times \tilde{\dot{p}}}$ [$\tilde{\bar{S}} \in I\!R^{n \times \tilde{\bar{p}}}$] by coloring $G(F'_v)$. We obtain F'_v by solving the linear system $\tilde{\dot{B}} - F'_c \cdot \tilde{\dot{S}} = F'_v \cdot \tilde{\dot{S}}$ [$\tilde{\bar{B}} - {F'_c}^T \cdot \tilde{\bar{S}} = {F'_v}^T \cdot \tilde{\bar{S}}$] with $\tilde{\dot{B}} \in I\!R^{m \times \tilde{\dot{p}}}$ [$\tilde{\bar{B}} \in I\!R^{n \times \tilde{\bar{p}}}$] using a simple substitution procedure.

Keywords: Sparse Jacobian Accumulation, Graph Coloring

Joint work of: Varnik, Ebadollah; Naumann, Uwe

[1] A. Griewank and A. Walther, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation,*SIAM, Other Titles in Applied Mathematics, 2nd (2008), Philadelphia, PA.*
[2] A. H. Gebremedhin and F. Manne, and A. Pothen, What Color is Your Jacobian? Graph Coloring for Computing Derivatives. *SIAM Review, 47 (2005), pp. 629-705.*
[3] A. R. Curtis and Powell, M. J. D. and J. K. Reid, On the Estimation of Sparse Jacobian Matrices, *Journal of the Institute of Mathematics and Applications, 13 (1974), pp. 117–119.*

Algorithmic Differentiation Through Automatic Graph Elimination Ordering (ADTAGEO)

Jan Riehme (Department of Computer Science, University of Hertfordshire, Hatfield, UK)

Algorithmic Differentiation Through Automatic Graph Elimination Ordering (ADTAGEO) is based on the principle of Instant Elimination: At runtime we dynamically maintain a DAG representing only active variables that are alive at any time. Whenever an active variable is deallocated or its value is overwritten the corresponding vertex in the Live-DAG will be eliminated immediately by the well known vertex elimination rule [1].

Consequently, the total memory requirement is equal to that of the sparse forward mode. Assuming that local variables are destructed in the opposite order of their construction (as in C++), a single assignment code is in effect differentiated in reverse mode. If compiler-generated temporaries are destroyed in reverse order too, then Instant Elimination yields the statement level reverse mode of ADIFOR [2] naturally.

The user determines the elimination order intentionally (or unintentionally) by the order in which he declares variables, which makes hybrid modes of AD possible by combining forward and reverse differentiated parts.

By annotating the Live-DAG with local Hessians and applying second order elimination rules, Hessian-vector products can be computed efficiently since the annotated Live-DAG stores one half of the symmetric Hessian graph only (as suggested in [1]).

Nested automatic differentiation is done easily by subsequent propagations, since sensitivities between variables alive can be obtained at any point in time within the Live-DAG.

The concept of maintaining a Live-DAG fits optimally into the strategy of overloaded operators for classes, it is a very natural example of Object Oriented Programming. A proof-of-concept implementation in C++ is available (contact the first author).

Keywords: Automatic Differentiation, Instant Elimination, Live-DAG, symmetric Hessian-DAG, forward mode, reverse mode, checkpointing, ADTAGEO

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2085

[1]. Griewank, A.: Evaluating Derivatives. *Principles and Techniques of Algorithmic Differentiation. SIAM (2000)*
[2]. Bischof, C.H., Carle, A., Khademi, P., Mauer, A.: ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science & Engineering 3 (1996) 18-32*

Joint work of: Riehme, Jan; Griewank, Andreas

3.6 Round Tables

Multilevel algorithms for discrete problems

Eric Boman (Sandia National Laboratory, USA)

Multilevel algorithms have proved to be an efficient framework for solving many problems, for example, linear systems of equations, graph partitioning, graph layout, and clustering. The multilevel approach has enabled fast solution of large problems, which would otherwise not have been possible. In this roundtable discussion, we discussed the state of the art and future developments.

The coarsening phase, where a smaller representation of the original problem is generated, is often critical. It is interesting that different application areas have chosen different approaches. In algebraic multigrid (AMG), a set of coarse vertices is selected and aggregates formed around these. In graph partitioning, typically pairs of vertices are matched and contracted. There is no fundamental reason for this difference. In recent work (presented at a poster here at Dagstuhl), Ilya Safro and Cedric Chevalier have shown that AMG-style coarsening also works well for graph partitioning, and may both be faster and give better quality solutions. Some customization depending on the objective seems to be helpful. Yifan Hu pointed out a similar approach has been deve;loped earlier for envelope (wave front) reduction and graph layout. One problem is that standard AMG methods may produce too large aggregates, giving too small coarse graphs.

The discussion then touched on other applications, like social networks. Multilevel methods may also be used to compute metrics like centrality and modularity. Clustering for dynamic networks or graphs is a new challenge.

To advance research in this area, it would be helpful to obtain metrics of what is a good "coarse" graph that can easily be evaluated. Currently, we have to implement a coarsening method and run the whole multilevel algorithm to observe the result for a specific application.

Keywords: Multilevel algorithms

Graph coloring for parallel computation

Assefaw Gebremedhin (Purdue University, USA)

Graph coloring can be used to discover concurrency in parallel scientific computing. The goal of this round table was to identify specific applications within this general area where further investigation might be needed. Three applications were discussed in some detail: adaptive mesh refinement, preconditioned iterative methods for sparse linear systems (where multi-coloring is used), and full sparse tiling. Of these, full sparse tiling and multi-coloring in the blocked case were considered worth investigating.

Keywords: Graph coloring, parallel computation

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2083

Data-Flow Reversal in Adjoint Codes

Uwe Naumann (LuFG Informatik 12, RWTH Aachen University, Germany)

Adjoint numerical simulation codes are widely used in computational science and engineering to compute the sensitivity of certain objectives with respect to potentially very large numbers of parameters efficiently. The data flow of the original simulation needs to be reversed. Intermediate values of the original computation need to be accessed in reverse order. Their storage is infeasible for all but the smallest problems. Combinations of storage (checkpointing) and recomputation yield the DATA FLOW REVERSAL (DFR) problem that is to minimize the computational effort for a given upper bound on the available memory. DFR could be shown to be NP-complete in [1].

The aim of this round table discussion was to identify an appropriate formalism and related data structures for the investigation and approximate solution of the DFR problem. Reversal trees [2] turned out to be the favorite choice. The related CALL TREE REVERSAL problem is also NP complete [3]. Ongoing research focuses on variants of call trees and heuristics for their efficient reversal. The integration of such techniques into tools for automatic differentiation (AD) remains one of the major aims of AD software developers.

[1] Uwe Naumann: DAG Reversal is NP-complete. To appear in J. Discr. Alg., Elsevier, 2009.

[2] Andreas Griewank and Andrea Walther: Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation (Second Edition), SIAM, 2008.

[3] Uwe Naumann: Call Tree Reversal is NP-Complete. Pages 13-22 in Christian H. Bischof, H. Martin Buecker, Paul D. Hovland, Uwe Naumann, and Jean Utke, eds.: Advances in Automatic Differentiation, Springer, 2008.

Keywords: Data flow reversal, adjoint codes

4 Short Bios of the Plenary Speakers

4.1 Bruce Hendrikson

Bruce Hendrickson is Senior Manager for Computer Science and Mathematics at Sandia National Laboratories. He is also an affiliated faculty in the Computer Science Department at the University of New Mexico. In his 18 year career at Sandia, Bruce has worked on software, algorithms, and analysis for a wide range of combinatorial and scientific applications. Bruce received his PhD in Computer Science from Cornell, after obtaining degrees in Mathematics and Physics from Brown. He is the author of nearly 100 scientific papers, has served on the editorial boards of a range of journals in parallel and scientific computing, and has helped organize a number of international meetings. His research interests include combinatorial scientific computing, parallel algorithms, linear algebra, graph algorithms, scientific software, data mining and computer architecture.

4.2 Alex Pothen

Alex Pothen is a Professor of Computer Science and Director of the Computing Research Institute at Purdue University, and Director of the CSCAPES Institute. Alex received an undergraduate degree from the Indian Institute of Technology in New Delhi, and his PhD from Cornell University. He has held earlier academic appointments at the University of Waterloo, Penn State, Old Dominion University, and ICASE, a research institute at NASA. Alex has had the privilege of being involved with the CSC research community since its inception, and was co-chair of the first three international workshops in CSC.

4.3 Rob Bisseling

Rob Bisseling is an associate professor in scientific computing at the Mathematics Department of Utrecht University, the Netherlands. He is leader of the Master's programme in Scientific Computing at Utrecht University. He received BSc and MSc degrees in mathematics from Nijmegen University in the Netherlands, and a PhD degree in theoretical chemistry from the Hebrew university in Jerusalem. Before his current position, he worked for six years in industry, at the Shell research laboratory in Amsterdam. His main research fields are parallel computing and combinatorial scientific computing. His current research interests are sparse matrix computations, information retrieval, bioinformatics, and fast Fourier transforms. He was co-organizer of the 2005 workshop on combinatorial scientific computing in Toulouse. He is author of the book, "Parallel Scientific Computation: A Structured Approach using BSP and MPI", Oxford University Press, March 2004.

4.4 Paul Hovland

Paul Hovland received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1997. He has been conducting research in automatic differentiation for over 15 years and is one of the original contributors to the ADIFOR system for differentiating Fortran 77 programs. He is currently a Computer Scientist at Argonne National Laboratory and directs research projects developing new algorithms and tools for Fortran 95 (OpenAD/F) and C/C++ (ADIC). Other research interests include performance engineering and tools to support development of high performance computational science applications.

4.5 Iain Duff

Iain Duff is a CCLRC Senior Fellow in the Computational Science and Engineering Department. After completing his D Phil at Oxford, he was a Harkness Fellow in the United States visiting Stony Brook and Stanford. He then spent two years lecturing at the University of Newcastle upon Tyne before joining the Harwell Laboratory where he became Group Leader of Numerical Analysis in 1986. In 1990, the Group moved to the Atlas Centre at the Rutherford Appleton Laboratory. He has had several extended visits to Argonne National Laboratory, the Australian National University, the University of Colorado at Boulder, Stanford University, and the University of Umeå. He is the Project Leader for the Parallel Algorithms Group at CERFACS in Toulouse and is a Visiting Professor at the University of Strathclyde. He is a life Fellow of the Institute of Mathematics and its Applications. He is a member of SIAM (USA), SMAI (France), SBMAC (Brazil) and is Editor or Associate Editor of twelve journals in numerical analysis or scientific computation. He is on the Council of the Institute of Mathematics and its Applications and is an IMA representative on the Board of ICIAM. He is chairman of the Board of Trustees of SIAM. He was elected as a Fellow of the Royal Society of Edinburgh in 2006. His current research interest cover numerical linear algebra, sparse matrices, parallel computing, scientific computation and mathematical software.

4.6 Ruud van der Pas

Ruud van der Pas is a Senior Staff Engineer in the Sun Developer Tools organization at Sun Microsystems and operates on a world-wide basis. Ruud has studied mathematics and physics and has been involved with the performance aspects of High Performance and Technical computing since 1985. Prior to joining Sun he worked at SGI, Convex Computer Corporation, the University of Utrecht and Philips. His experiences over the past 23 years cover a great variety of architectures, including the Cray-1 and Cyber 205 early vector supercomputers, clusters of workstations, vector shared memory systems, plus cache based SMP, cc-NUMA and CMT architectures. He has been with Sun since 1998. His expertise is in serial performance, shared memory parallelization using OpenMP and application performance analysis. Ruud is a board member of community, the community of OpenMP researchers and developers in academia and industry. At Sun, his main responsibility is to help defining the specifications for future products and features regarding the Sun compilers and tools, with a focus on shared memory parallelization. Ruud is also responsible for the Sun Application Tuning Seminar and works closely with several other engineering groups within Sun to discuss enhancements to current and future products. Additionally, he regularly gives technical presentations and workshops at conferences and seminars all over the world. He is also on the organizing committee of several conferences and workshops. Ruud closely works with several of Sun's leading High-Performance Computing customers all over the world, including the RWTH Aachen in Germany, the Danish Technical University (DTU) in Denmark, the Tokyo Institute of Technology (TiTech) in Japan, and the University of Houston and Stanford University, both in the USA. He is also involved in the area of Reliable Computing, with a focus on Interval Analysis and Interval Arithmetic. Ruud is co-author of the book “Using OpenMP", published by MIT Press in October 2007.

4.7 David Bader

David A. Bader is Executive Director of High-Performance Computing and a Full Professor in Computational Science and Engineering, a division within the College of Computing, at Georgia Institute of Technology. Dr. Bader also serves as Director of the Sony-Toshiba-IBM Center of Competence for the Cell Broadband Engine Processor located at Georgia Tech. He received his Ph.D. in 1996 from The University of Maryland, was awarded a National Science Foundation (NSF) Postdoctoral Research Associateship in Experimental Computer Science. He is an NSF CAREER Award recipient, an investigator on several NSF awards, was a distinguished speaker in the IEEE Computer Society Distinguished Visitors Program, and a member of the IBM PERCS team for the DARPA High Productivity Computing Systems program. Dr. Bader serves on the Research Advisory Council of Internet2 and the Steering Committees of the IPDPS and HiPC conferences, and was the General co-Chair for IPDPS (2004–2005), and Vice General Chair for HiPC (2002–2004). David has chaired several major conference program committees: Program Chair for HiPC 2005, Program Vice-Chair for IPDPS 2006 and Program Vice-Chair for ICPP 2006, and has served on numerous conference program committees related to parallel processing and computational science & engineering, is an associate editor for several high impact publications including the IEEE Transactions on Parallel and Distributed Systems (TPDS), the ACM Journal of Experimental Algorithmics (JEA), IEEE DSOnline, and Parallel Computing, is a Senior Member of the IEEE Computer Society and a Member of the ACM. Dr. Bader has been a pioneer in the field of high-performance computing for problems in bioinformatics and computational genomics. He has co-chaired a series of meetings, the IEEE International Workshop on High-Performance Computational Biology (HiCOMB), written several book chapters, and co-edited special issues of the Journal of Parallel and Distributed Computing (JPDC) and IEEE TPDS on high-performance computational biology. He has co-authored over 90 articles in peer-reviewed journals and conferences, and his main areas of research are in parallel algorithms, combinatorial optimization, and computational biology and genomics.

4.8 Michael Mahoney

Michael Mahoney is currently a research scientist at Stanford University. He is currently working on geometric network analysis; developing approximate computation and regularization methods for large informatics graphs; and applications to community detection, clustering, and information dynamics in large social and information networks. His research interests also include randomized algorithms for matrices and regression problems, applications to DNA single nucleotide polymorphism data, and large-scale statistical data analysis more generally. He has been a faculty member at Yale University and a researcher at Yahoo.

Day	Chair	Time	Talk	Title	Speaker
Sunday		after 3pm		Arrival	
Monday	U. Naumann	9am–9.15am	Opening		Organizers
		9.15am–10.15am	Plenary Talks	Combinatorial Scientific Computing: A View to the Future	B. Hendrickson
		11am-noon		The Dance of a Thousand Processors: Parallel Algorithms in CSC	A. Pothen
	O. Schenk	2pm-3pm		Combinatorial Problems in HPC	R. Bisseling
		3.30pm-4.30pm	Contributed Talks	Node and Edge Label Overlap Removal in Visualizing Large Networks	Y. Hu
				Micro-Finite Element Analysis of Human Bone Structures	P. Arbenz
				PDEs with a trillion elements – When theory fails to predict performance	U. Rüde
		4.30pm-6.30pm	Tutorial	Zoltan	E. Boman
Tuesday	U. Naumann	9am-10am	Plenary Talk	Combinatorial Problems in Automatic Differentiation	P. Hovland
		11am-noon	Contributed Talks	The CPR Method and Beyond	T. Steihaug
				The Enabling Power of Graph Coloring Algorithms in Automatic Differentiation & Parallel Proc.	A. Gebremedhin
				Randomized Heuristics for Exploiting Jacobian Scarcity	A. Lyons
	S. Toledo	2pm-3pm	Plenary Talk	Combinatorial Problems in Numerical Linear Algebra	I. Duff
		3.30pm-4.30pm	Contributed Talks	Reducing the total bandwidth of a sparse unsymmetric matrix	J. Reid
				Multifrontal multithreaded rank-revealing sparse QR factorization	T. Davis
				Evaluation of Sparse LU Factorization and Triangular Solution on Multicore Platforms	X. Li
		4.30pm-6.30pm	Tutorial	Scotch and PT-Scotch	F. Pellegrini
		7.30pm-9.30pm	Tutorial	OpenAD	J. Utke
Wednesday	U. Rüde	9am-10am	Plenary Talk	Present and Future of High-Performance Scientific Computing	R. van der Pas
		11am-noon	Contributed Talks	Hypergraphbased Unsymmetric Nested Dissection Ordering for Sparse LU	S. Donfack
				Heuristic initialization of bipartite matching algorithms	J. Langguth
		Afternoon		Hiking / Trip to Trier and wine tasting	
Thursday	A. Pothen	9am-10am	Plenary Talk	Emerging Applications in Combinatorial Scientific Computing	D. A. Bader
		11am-noon	Contributed Talks	Parallel Short Sequence Mapping for High Throughput Genome Sequencing	U. Catalyurek
				Fast AMD orderings for matrices with some dense rows	J. Scott
				A Nearly-Linear Time Algorithm for Solving Linear Systems in a Symmetric M-Matrix	S. Daitch
		2pm-4pm	Tutorials	Ipopt	A. Wächter
		4pm-6pm		ADOL-C	A. Walther
		2pm-6pm	Round Tables	Data-Flow Reversal in Adjoint Codes	U. Naumann
				Graph coloring for parallel computation	A. Gebremedhin
				Multilevel algorithms for discrete problems	E. Boman
	J. Gilbert	Evening	Plenary Talk	Combinatorial and scientific computing approaches to modern large-scale data analysis	M. Mahoney
Friday		9am-11am	Tutorial	Star-P	J. Gilbert
	A. Pothen	11am-noon	CSC Community Meeting		
		noon	Closing	Departure	Organizers

Table 1. Timetable for Dagstuhl Seminar 09061 in Combinatorial Scientific Computing held from 01.02.2009 to 06.02.2009.