Final Report for UC Berkeley

Terascale Optimal PDE Solvers TOPS

DOE Award Number DE-FG02-01ER25478

9/15/2001 – 9/14/2006

PI: David Keyes

UC Berkeley co-PI: James Demmel

Funded Team Members: J. Demmel, E. J. Riedy (grad student), Y. Hida (grad student), J. Nie (grad student), Sonil Mukherjee (undergrad)

Unfunded collaborators: K. Yelick, R. Vuduc (grad student, later staff at LLNL), E.-J. Im (grad student, later faculty at Kookmin U. in Seoul), T. Katagiri (academic visitor from U. Electro-Communications, Japan), X. S. Li (funded through LBNL), L. Grigori (funded through LBNL), M. Hoemmen (grad student), H. Gahvari (grad student), B. Lee (undergrad), S. Kamil (undergrad), R. Nishtala (undergrad, later grad student), C. Hsu (undergrad), A. Gyulassy (undergrad), J. Moon (undergrad), W. Tu (undergrad), A. Jain (undergrad), S. Lindeneau (undergrad)

Executive Summary: In many areas of science, physical experimentation may be too dangerous, too expensive or even impossible. Instead, large-scale simulations, validated by comparison with related experiments in well-understood laboratory contexts, are used by scientists to gain insight and confirmation of existing theories in such areas, without benefit of full experimental verification. The goal of the TOPS ISIC was to develop and implement algorithms and support scientific investigations performed by DOE-sponsored researchers. A major component of this effort is to provide software for large scale parallel computers capable of efficiently solving the enormous systems of equations arising from the nonlinear PDEs underlying these simulations. Several TOPS supported packages where designed in part (ScaLAPACK) or in whole (SuperLU) at Berkeley, and are widely used beyond SciDAC and DOE. Beyond continuing to develop these codes, our main effort focused on automatic performance tuning of the sparse matrix kernels (eg sparse-matrix-vector-multiply, or SpMV) at the core of many TOPS iterative solvers. Based on the observation that the fastest implementation of SpMV (and other kernels) can depend dramatically both on the computer and the matrix (the latter of which is not known until run-time), we developed and released a system called OSKI (Optimized Sparse Kernel Interface) that will automatically produce optimized version of SpMV (and other kernels), hiding complicated implementation details from the user. OSKI led to a 2x speedup in SpMV in a DOE accelerator design code, a 2x speedup in a commercial lithography simulation, and has been downloaded over 500 times. In addition to a stand-alone version, OSKI was also integrated into the TOPS-supported PETSc system.

Comparison of Goals and Accomplishments. As described in the executive summary and in more detail below, TOPS' goal was to help DOE scientist perform their simulations more effectively, which means both faster and in an easy-to-use fashion. Our efforts in performance tuning successfully met the goal of ``faster'' with speedups of 4x for SpMV and similarly for many other widely used sparse matrix kernels. Ease-of-use was addressed by developing OSKI, which encapsulated many of the complicated tuning activities to a very simple user interface, and further "invisibly" integrating OSKI into PETSc. Impact may be measured by the 2x speedup in SpMV in a DOE accelerator design code, a 2x speedup in a commercial lithography simulation, being downloaded over 500 times.

Detailed Description of Activities. We begin with an overview of TOPS, and then describe UC Berkeley's contributions in more detail.

In many areas of science, physical experimentation is impossible, such as with cosmology; dangerous, as with manipulating the climate; or simply expensive, as with fusion reactor design. Large-scale simulations, validated by comparison with related experiments in well-understood laboratory contexts, are used by scientists to gain insight and confirmation of existing theories in such areas, without benefit of full experimental verification. But today's high-end computers, such as those at DOE's supercomputing centers, are one-of-a-kind, and come without all of the scientific software libraries that scientists expect to find on desktop workstations.

The **TOPS ISIC** was created to develop and implement algorithms and support scientific investigations performed by DOE-sponsored researchers. These simulations often involve the solution of partial differential equations (PDEs) on terascale computers. The TOPS Center researched, developed and deployed an integrated toolkit of open-source, optimal complexity solvers for the nonlinear partial differential equations that arise in many DOE application areas, including fusion, accelerator design, global climate change and reactive chemistry. The algorithms created as part of this project were also designed to reduce current computational bottlenecks by orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible.

Nonlinear PDEs give mathematical expression to many core DOE mission applications. PDE simulation codes require implicit solvers for the *multirate*, *multiscale*, *multicomponent*, *multiphysics* phenomena of hydrodynamics, electromagnetism, chemical reaction and radiation transport. Currently, such problems typically reach into the tens of millions of unknowns — and this size is expected to increase 100-fold in just a few years.

Unfortunately, the algorithms traditionally used to solve PDEs were designed to address smaller problems and become much less efficient as the size of the system being studied increases. This creates a double jeopardy for applications, particularly in the case when the algorithms are based on iterations — as it takes more computing resources to solve each step of the problem, and the number of steps also goes up. Fortunately, the physical origin of PDE problems often allows them to be addressed by using a sequence of approximations, each of which is smaller than the one before. The solutions to the approximations, which may be obtained more efficiently, are combined judiciously to provide the solution to the original problems. One well-known example of this approach is called the multigrid method, which solves a problem by tackling a coarse approximation and then using the solution to generate the solution of a better approximation, and so on. It can be shown that the performance of multigrid method is optimal for certain classes of problems. The underlying philosophy of this and other similar approaches is to make the majority of progress towards a high quality result through less complex intermediate steps.

The efforts defined for TOPS and its collaborations with other projects have been chosen to revolutionize large-scale simulation through incorporation of existing and new optimal algorithms and code interoperability. TOPS provides support for the software packages Hypre, PARPACK, PETSc, ScaLAPACK, Sundials, SuperLU and TAO, some of which are in the hands of thousands of users, who have created a valuable experience base on thousands of different computer systems.

Software developed and supported by the TOPS project is being used by scientists around the globe. In the past few years, researchers outside the SciDAC community authored more than two dozen scientific papers reporting results achieved using TOPS software. The papers cover many disciplines, from astronomy to chemistry to materials science to nanotechnology to optics.

TOPS solver software has also been incorporated into numerous other packages, some commercial and some freely available. Among the widely distributed packages maintained outside of the SciDAC program that employ or interface to TOPS software "under the hood" are Dspice, EMSolve, FEMLAB, FIDAP, Global Arrays, HP Mathematical Library, libMesh, Magpar, Mathematica, NIKE, Prometheus, SCIRun, SLEPc, Snark and Trilinos.

TOPS software is taught in many courses in the U.S. and abroad, and forms a core of the annual training workshop sponsored by DOE to introduce researchers to the Advanced CompuTational Software (ACTS) Collection. TOPS software is also regularly featured in short courses at professional meetings.

Next we give a **detailed description of UC Berkeley's activities and contributions**. Contributions fall into the following categories.

Automatic Performance Tuning of Sparse Matrix Kernels. This was our main area of effort, leading to publications [6,12,13,14,16,17,18,19,21,22,23] and a software release [24]. Building on on-going work on tuning sparse matrix kernels [6], we explored a variety of techniques to improve the performance of sparse matrix kernels like sparse-matrix-vector-multiplication (SpMV) depending both on the computer architecture and the matrix. Without optimizations, SpMV using the usual CSR sparse matrix format typically runs at 10% or less of machine peak, but using just one of our tuning techniques (Register Blocking, or RB) can raise performance up to 4x on some matrices, and up to 31% of peak on some architectures. Other optimizations include exploiting symmetry (another 2.6x faster than RB alone), variable block splitting (1.8x more than RB), diagonal representation (2x over CSR), reordering to create dense substructures (2x over CSR), cache blocking (3x over CSR), multiple vectors (7x over CSR). Other sparse matrix operations tuned included triangular solve (1.8x over CSR), AA^Tx and A^TAx (4x over CSR, 1.8x over RB), and A^2x (2x over CSR, 1.8x over RB). Because it the choice of best data structure and algorithm depends in a very non-obvious way on both the matrix and the architecture, a system called OSKI (Optimized Sparse Kernel Interface) was designed and released [24] to make it easy for users to get the benefit of our optimizations. In joint work with K. Ko, our optimizations were applied to two DOE accelerator design matrices, leading to a speed up of 1.68x on Itanium 2 for SpMV for the T3P matrix (4x for multiple vectors) and 2.1x on Power 4 for SpMV for the Omega3P matrix. Papers [12,13,17] were awarded prizes.

Other performance tuning activities. [5,8,20] Since the search process within automatic performance tuning can potentially take a long time, in [5] we developed statistical techniques for assessing how close to the best performance we had found, and ending the search earlier. An overview of automatic tuning for sparse and dense linear algebra on a variety of (parallel) architectures appears in [8]. In [20] we discussed tuning for eigenvalue problems, showing how to get speedups on SIMD architectures.

Other sparse matrix algorithms. [4,10,11,15] We continued our collaboration with X. S. Li of LBNL on parallel sparse direct methods, with a description of our SuperLU_DIST solver in [4]. Postdoc Laura Grigori, who Demmel and Li co-advised, made further improvements on the symbolic factorization [10,11,15].

Improved algorithms and standards for new matrix kernels. [1,2,3,7,9] In a community-wide effort co-lead by collaborator J. Dongarra, we developed and released a new standard [2] for the widely used Basic Linear Algebra Subroutines (BLAS), which form the basis of most if not all dense and sparse linear algebra libraries. This work was meant to identify functions missing from the earlier BLAS standard, update the interface to multiple languages, include sparse matrices, and permit higher precision computations [1]. Part of the effort involved new algorithms for efficient high precision computations [3], which has applications in both computational geometry [7] and linear algebra [9].

Products developed (paper prizes shown in *italics*)

- **Journal Publications:**
  - [1] X. S. Li, J. Demmel, D. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S. Kang, A. Kapur, M. Martin, B. Thompson, T. Tung and D. Yoo, ”Design, implementation and testing the extended and mixed precision BLAS,” ACM Trans. Math. Software, 28(2) 2002 (also LBNL Tech Report LBNL-45991).
  - [2] L. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, R. Whaley, “An updated set of Basic Linear Algebra Subroutines (BLAS)”, ACM Trans. Math. Software, 28(2) 2002
  - [3] J. Demmel and Y. Hida, “Accurate floating point summation,” SIAM J. Sci. Comput., 25(4), pp1214-1248, 2003.
  - [4] X. S. Li and J. Demmel, “SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems,” ACM Trans. Math. Software, 29:110-140, 2003.
  - [5] R. Vuduc, J. Demmel and J. Bilmes, “Statistical Models for Empirical Search-based Performance Tuning,” Intern. J. High Performance Computing Applications, 18(1) pp 65-94, 2004.
  - [6] E.-J. Im, K. Yelick and R. Vuduc, “SPARSITY: An optimization framework for sparse matrix kernels,” Intern. J. High Performance Computing Applications, 18(1), pp 135-158, Feb 2004.
  - [7] J. Demmel and Y. Hida, “Fast and accurate floating point summation with application to computational geometry,” Numerical Algorithms 37(1-4) Dec 2004 (also presented at SCAN 2002)
  - [8] J. Demmel, J. Dongarra, V. Eijkhout, E. Fuentes, A. Petitet, R. Vuduc, R. Whaley and K. Yelick, “Self-adapting linear algebra algorithms and software,” Proc. of the IEEE, Special Issue on Program Generation, Optimization, and Platform Adaptation, 93(2) Feb 2005.
  - [9] J. Demmel, Y. Hida, W.Kahan, X. Li, S. Mukherjee, E. Riedy, ``Error bounds from extra precise iterative refinement,” ACM Trans. Math. Software, 32(2), 2006.
  - [10] L. Grigori, J. Demmel and X. S. Li, "Parallel Symbolic Factorization for Sparse LU with Static Pivoting," SIAM J. Scientific Computing, (accepted), 2006 (also appeared as LBNL Tech Report LBNL-59031, Oct 2005)
- **Conference Publications:**
  - [11] L. Grigori and X. S. Li, “A new scheduling algorithm for parallel sparse LU factorization with static pivoting,’’ Proc. Supercomputing’02, Baltimore, Nov 2002
  - [12] R. Vuduc, J. Demmel, K. Yelick, S. Kamil, R. Nishtala and B. Lee, ”Performance optimization and bounds for sparse matrix-vector multiply,’’ Proc. Supercomputing’02, Baltimore, Nov 2002 *(Finalist, Best Student Paper, to R. Vuduc)*

- [13] R. Vuduc, S. Mail, J. Hsu, R. Nishtala, J. Demmel and K. Yelick, “Automatic performance tuning and analysis of sparse triangular solve,” ICS 2002: Workshop on Performance Optimization via High-Level Languages and Libraries, 2002. (*Best Student Paper and Best Student Presentation Awards, to R. Vuduc)*
- [14] R. Vuduc, A. Gyulassy, J. Demmel and K Yelick, “Memory hierarchy optimizations and performance bounds for sparse A^TAx,” Proc. ICCS 2003: Workshop on Parallel Linear Algebra, Melbourne Australia, June 2003.
- [15] L. Grigori and X. S. Li, “Performance analysis of parallel right-looking sparse LU factorization on two-dimensional grids of processors,” Proc. PARA’04: Workshop on State-of-the-Art in Scientific Computing, Springer Lecture Notes in Computer Science, 2004.
- [16] E.-J. Im, I. Bustany, C. Ashcraft, J. Demmel and K. Yelick, “Toward automatic performance tuning of matrix triple products based on matrix structure,” Proc. PARA’04: Workshop on State-of-the-Art in Scientific Computing, 2004.
- [17] B. Lee, R. Vuduc, J. Demmel and K. Yelick, ”Performance models for evaluation and automatic tuning of symmetric sparse matrix-vector multiply,” Intern. Conf. on Parallel Processing, Montreal, Aug 2004. *(Best Paper Prize)*
- [18] R. Vuduc, J. Demmel, and K. Yelick, “OSKI: A library of automatically tuned sparse matrix kernels,” Proc. SciDAC 2005 meeting, San Francisco, June 2005.
- [19] R. Vuduc, H.-J. Moon, “Fast sparse matrix-vector multiplication by exploiting variable block structure,” Proc. Intern. Conf. High Performance Computing and Communications, Sorrento, Italy, Sept 2005.
- [20] J. Demmel, T. Katagiri, C. Voemel, ”Automatic Performance Tuning for Multisection with Multiple Eigenvalues Method for the Symmetric Eigenproblem,” Proc. PARA’06: Workshop on State-of-the-Art in Scientific and Parallel Computing, CP4, Umea, Sweden, June 2006 (to appear in Springer LNCS)

- **Other Publications:**
  - [21] T.-Y. Chen, “Preconditioning sparse matrices for computing eigenvalues and computing linear systems of equations,” PhD Dissertation, Computer Science Division, UC Berkeley, 2001
  - [22] R. Vuduc, “Automatic performance tuning of sparse matrix kernels,” PhD Dissertation, Computer Science Division, UC Berkeley, 2003
  - [23] R. Nishtala, R. Vuduc, J. Demmel, K. Yelick, ”Performance modeling and analysis of cache blocking in sparse matrix vector multiply,” Tech report UCB/CSD-04-1335, Computer Science Divsion, UC Berkeley, 2004
- **Software Releases:**
  - [24] OSKI (Optimized Sparse Kernel Interface) version 1.0.1b (March 2006), bebop.cs.berkeley.edu/oski
    - OSKI-PETSc (March 2006) bebop.cs.berkeley.edu/oski