Performance Optimization of Sparse Kernels in TOPS

PIs: J. Demmel[4], J. Dongarra[5], J. Ding[5], V. Eijkhout[5], D. Keyes[3], S. Li[2,4], B. Smith[1], R. Vuduc[4]

[1]Argonne National Lab, [2]Lawrence Berkeley National Lab, [3]Old Dominion U., [4]U. California-Berkeley, [5]U. Tennessee

Summary

To deliver as high a percentage of per-processor peak performance as is practical on the hierarchical memory architectures on which most SciDAC scientists make their production runs, the Terascale Optimal PDE Simulations (TOPS) Center is researching innovative strategies for tuning the kernels that arise most often in solving sparse linear systems from DOE simulations.

TOPS is automatically tuning performance of sparse matrix kernels that dominate many scientific and engineering applications. Given a sparse matrix (i.e., its sparsity pattern and other properties like symmetry), the operation to be performed (e.g., sparse matrix vector multiply or SpMV, triangular solve) and the processor architecture, it is possible to build a custom data structure and implementation that can substantially increase performance. The speedups from uniprocessor tuning depend strongly on the matrix, the operation to be performed and architecture, but can range up to a factor of seven or more for sparse-matrix-multiple-vector multiplication. Composite operations, like A^T*A*x, are also amenable to high speedups when performed atomically.

Straightforward implementations of standard operations on large data objects can run slowly on contemporary hierarchical memory processors, where the main memory latency is 100 or more times greater than the processor clock period. For operations with no cache locality whatsoever, this bounds performance at 1% of peak or less.

The Basic Linear Algebra Subroutines (BLAS) invented for dense linear algebraic operations are universally used, because well-tuned BLAS break the work into cache-sized blocks, augmenting reuse, and often achieve 80% or more of peak. Furthermore, the BLAS can be tuned automatically using packages like ATLAS or PHiPAC. The FFTW package takes a similar approach for the FFT.

Local discretizations (e.g., finite elements) of partial differential equations (PDEs) typically give rise to sparse linear systems that are much less amenable than dense systems to obtaining a high percentage of peak. SpMV is the most important kernel in iterative methods for these PDEs. Its memory traffic pattern is also analogous to evaluating a differential operator on a grid function, a grid transfer operation in a multilevel method, or accumulating the right-hand side in a triangular solve: each matrix element is used only once, magnifying the importance of tuning these operations.

Multicomponent systems of PDEs, when ordered most rapidly by unknowns at a gridpoint, have a sparse structure of dense blocks. Furthermore, density increases through fill-in in incomplete factors often employed as preconditioners, or in exact LU factorizations. By exploiting effects like these through hand-tuning, researchers using the now TOPS-supported PETSc software were able to obtain up to 25% of peak uniprocessor performance on hierarchical memory machines in an implicit aerodynamics computation en route to a Bell Prize in 1999, for a code whose original uniprocessor performance was less than 5%. We are automating such optimizations in TOPS.

Whereas matrices arising in PDE simulations are sparse, vectors representing gridfunctions are dense. Some linear algebraic methods operate on multiple vector columns simultaneously, providing another source of density to exploit. Historically, block iterative methods of this type have not enjoyed great favor. However, tests performed by TOPS researchers show improved computation rates for block algorithms due to their superior memory locality—an effect that may overcome convergence rate disadvantages in many problem-architecture combinations.

TOPS researchers have identified six common kernels arising in sparse linear algebraic computations and have subjected them to exhaustive performance tuning on seven commercially important uniprocessors (including the Power3 and Power4 systems at NERSC and ORNL's CCS).

R. Vuduc, in work that was honored with "Best Student Paper" at an ICS'02 workshop, has also done a theoretical performance analysis for one optimization technique, register blocking, that provides an upper performance bound for this operation on a given matrix and architecture obtained from modeling the memory traffic alone. On a test suite of 44 matrices from applications and four architectures, we are within 20% of optimal for many matrices, especially those from finite element modeling. (We use PAPI data to validate the predictions of our model.) On non-FEM matrices, speedups of a factor of two are still possible. In addition to register blocking, we use "switch-to-dense", which recognizes that triangular matrices resulting from (I)LU factorizations are often quite dense in their trailing submatrix, so that a dense BLAS implementation can be used there. Similar speedups are obtained, as well as agreements between actual performance and memory-based performance modeling. Furthermore, on the most recent Intel Itanium 2 architecture, we can achieve up to 30% of peak machine speed on SpMV for matrices from FEM and protein modeling codes, and speedups of up to two on the challenging web connectivity matrix used by the Google search engine. This demonstrates the relevance of our work on future architectures and on current and new applications alike.

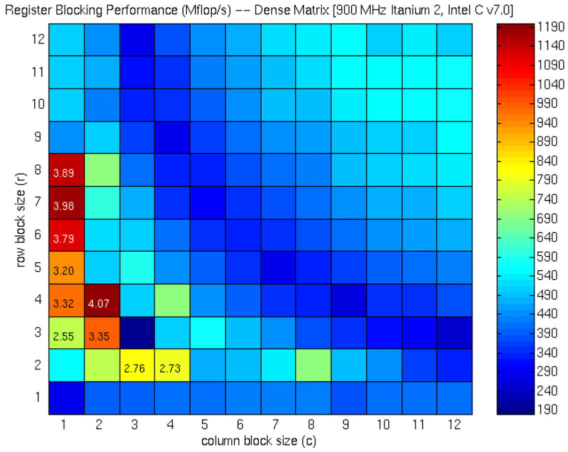

Figure 1. SpMV register blocking performance, 900 MHz Itanium 2 for a dense matrix stored in sparse format. Block sizes r×c are shown up to 12×12. A judicious choice of block size (here, 4×2) leads to 1.2 Gflop/s performance, or 33% of peak speed—a 4x increase over the conventional (1×1) code. This picture varies dramatically across platforms and matrices.

TOPS researchers are also developing automatically tuned implementations of the smoother component of multigrid, obtaining up to a factor of three over the simple 3-loop implementation of a natural ordering smoother.

The TOPS project webpage may be found at http://www.tops-scidac.org.

For further information on this subject contact:
Prof. David E. Keyes, Project Lead
Old Dominion University
Phone: 757-683-3882
dkeyes@odu.edu