CSCI 475/600, Fall 2001.

5th December 2001

Exploring the effect on performance of different patterns of data access via matrix multiplication

Assignment Overview (Due Mon. Dec. 10)

You will develop a program, in a high level language of your choice, which compares the performance of each of the possible loop orderings for matrix multiplication in single precision for several sizes of conformal matrices. The components of this comparison will include the following:

1. The generation of the elements, in double precision, of each of the matrix pairs from a specified random distribution.
   (a) if x and y are random numbers drawn from the distribution (0,1), each matrix entry will be: $\mathrm{sign}\,(y - \frac{1}{2}) * \exp(16 * (x - \frac{1}{2}))$.
2. The formation of double precision reference product matrices from the double precision matrix pairs using any loop ordering.
   (a) The input matrices will have a rectanguar shape with the sides having a ratio of 5 to 1. Use either two separate matrices forming AB^T and $B^T A$, or one matrix forming CC^T and $C^T C$.
   (b) For the smallest size, use the smallest matrix whose product timings are distinct. A default size can be (50,250).
   (c) For the largest size, use the largest matrix whose product timings can complete in a reasonable amount of time. A default size can be (200,1000).
3. The formation of single precision matrix pairs from the double precision pairs.

4. The CPU-timed generation of single precision product matrices, for all loop orderings, from the conformal single precision matrix pairs.
5. The calculation of the matrix 1, ∞, and F norms of the difference between the single and double precision products for each loop ordering. The explicit formation of error matrices embodying these differences should not be necessary.
6. Compilation and execution at both the lowest and highest (e.g. -fast) level of optimization.

To be submitted for grading:

1. Descriptively documented source code.
2. Informatively documented execution output for both optimization levels.
3. A cogent analysis of the results addressing:
   (a) a *ranking* of the accuracy and of the efficiency of the loop orderings for each product and for both optimization levels. (your program may perform this automatically)
   (b) a certification that the computed *norms* satisfy the relative *inequalities* as established by theory.
   (c) *the effect of the loop orderings on efficiency*. Include the discussion of how the patterns of data access of each ordering interact with the computer architecture.
   (d) *the effect of the size and shape of the matrices on the rankings* . Did the efficiency scale as exactly $O(m \times n \times m)$ or $O(n \times m \times n)$for the different matrix sizes?
   (e) *the effect of the optimization level on the rankings*. How did the accuracy and efficiency change?
   (f) *the effect of the loop orderings on accuracy*. Was there a significant difference between the accuracies of the different orderings? Did a high compiler optimization level affect the accuracy?

Why are we doing this?

CSCI475: This course includes an exploration of the factors which affect the quality and performance of numerical software with respect to both time and accuracy. This assignment examines performance issues, with respect to both time and accuracy, relating to: the scaling of problem size and shape, the permutation of statement order while maintaining mathematically, but not necessarily computationally, equivalent transformations, compiler optimization and the target computer language and architecture. Most large, computationally challenging, numerical problems are expressed as matrix equations, so the overwhelmingly dominant data structure in this context is the multidimensional array. Your facility with operations on this data structure will be exercised significantly in this assignment. The computation of several different error norms will enable you to check the reasonableness of the results of the matrix products and of the norms themselves.

Assessment:

You will be expected to perform the following:

- generate test data as per specification;
- document their source code and output descriptively;
- correctly implement all of the permutations of the loop ordering for the matrix product, and for the three specified matrix norms, with the specified precision, matrix sets, and optimization levels;
- infer the correctness of their program by using the norms, in part, as diagnostic instruments;
- cogently assess the influence that the underlying software and hardware architecture has in conjunction with the factors of data access patterns (generated by the respective loop orderings), relative matrix size, and optimization level;