CS320/CSE302/ECE392: INTRODUCTION TO PARALLEL PROGRAMMING FOR SCIENTISTS AND ENGINEERS

Spring 2000

Machine Problem 2

Assigned: February 21, 2000
Due: March 1, 2000.

The objective of this MP is to implement several subroutines using the OpenMP extensions. Each subroutine should execute as efficiently as possible in a multiprocessor environment containing a few processors (4 to 8 processors). Each subroutine corresponds to an operation in Fortran 90, but the implementation should be done in Fortran 77, C, or C++ using scalar operations only.
Use Guideview to visualize the performance of your implementations. For each routine operating on matrices, find the efficiency corresponding to n=10, n=100, and n=500. For each routine operating on vectors, find the efficiency corresponding to n=100,000 and =1,000,000. The performance of each function/subroutine should be measured separately. For the functions with vector parameters you should measure the execution time (using guideview) of a single invocation and of 50 consecutive invocations. A table with the computed efficiency for each of the cases just mentioned should be included in the final report together with an explanation of the values obtained. Use 4 processors to make the computation.

The subroutines are as follows (all vectors and matrices are `real*8`):
`add(a,b,c,n)`
This routine adds two vectors, `a` and `b` (both of length `n`), and stores the result in vector `c`.

`matmul(a,b,c,n)`
This routine multiplies two `n` by `n` matrices, `a` and `b`, and stores the result in matrix `c`.

`sum(a,b,n)`
This routine adds all the elements of a vector `a` (of length `n`) and returns the result in variable `b`.

`maxloc(a,n,m)`
This routine finds the first element of vector `a` (of length `n`) having the maximum value of the elements of `a`.

`prefix(a,b,n)`

This routine returns vector `b` with value $b(i) = \sum_{j=1}^{i} a(j)$. The length of vectors `a` and `b` is `n`.

`transpose(a,n)`

This routine transposes `a` (an `n` by `n` matrix) in place. That is, the resulting value of `a(i,j)` should be equal to the original value of `a(j,i)`

`firstnonzero(a,n,m)`
This routine finds the first nonzero element of `a` (a vector of length `n`) and stores its location in `m`

Graduate students registered for 1 unit should also implement:

Blocked versions of `transpose` and `matmul`. That is, `tranpose` and `matmul` should be implemented as submatrix operations where each submatrix is $\sqrt{n} \times \sqrt{n}$. For example, the obvious way to implement `matmul` would be to have a triply-nested loop control the operations on submatrices, and use another triply-nested loop for the submatrix operations. In other words, `matmul` could be implemented as a loop with six levels of nesting.