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

Spring 1998

Machine Problem 2

Assigned: February 17, 1999
Due: March 1, 1999.

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.

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.`

Graduate students registered for 1 unit should also implement:
`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`

The OpenMP Fortran 77 compiler can be found in:
`/home/cse/cse302/cse302/fortran/guide36/bin/guidef77,`
the C++ compiler is in
`/home/cse/cse302/cse302/cc++/guide36/bin/guidec++`
and the C compiler is in:
`/home/cse/cse302/cse302/cc++/guide36/bin/guidec.`

You may want to set the path variable in your account to facilitate invoking the compiler of your choice. In the class web site, there are pointer to both the Fortran and the C/C++ manuals for OpenMP. Please make sure to consult the manuals before implementing. The syntax used in version 3.6 of OpenMP is somewhat different from the syntax discussed in class.