Homework 5 Parallel Programming, Algorithms, and Architectures Fall 2000

Due date: 5pm Tuesday, November 21, 2000

Problem 1

Use OPENMP to write a code that allows you to select the scheduling strategy of a do loop and determine which iterations of the loop are actually scheduled on each thread. Your code should allow you to vary the work amount per iteration (without involving substantial memory traffic so as not to introduce extra effects into the experiment), the number of threads, and the number of iterations in the loop. You should also not perform I/O during the parallel execution in order to avoid significant effects on the scheduling. Check the effects of the various scheduling parameters on the assignment of iterations to threads. Do they all perform as advertised? (It is always a good idea to perform such simple tests to get used to the language and to check function of the supplied runtime library.)

Problem 2

Recall the simple bubble sort algorithm that was discussed in terms of Cedar Fortran in an earlier homework. Implement this algorithm on a single SGI O200 using OPENMP. Your code should run on 2,3, and 4 processors correctly with an arbitrary length vector of numbers to be sorted.

Use simple reads and writes to memory for synchronization. Recalling our discussions yesterday in class concerning potential problems with compilers and code generators on memory based shared memory synchronization you probably should look into the documentation on the OPENMP directive FLUSH in the insight online documentation and man pages. Also not one of the solutions advocated assumed a doacross loop. Therefore you should consider carefully based on your work in problem 1 the actual scheduling (static, dynamic, etc) that is appropriate for this type of implementation.