COSC 6422: Parallel and Distributed Computing
Assignment 1
Handout date: January 6, 1998
Due date: Tuesday February 10, 1998

1. Purpose

The object of this assignment is to write and evaluate the performance of a simple (but not too simple) parallel program. You are to choose an application and implement it using both a distributed shared memory and message passing interface. The distributed shared memory system you use will be Treadmarks. The message passing system you will use will be PVM. Both are installed in the LCSR (see the LCSR section of the course home page for links to more information). After each parallel program has been tested, debugged and tuned, you are to measure, evaluate and compare the performance of each implementation. Be sure to explain why you think they obtain the performance that they do and whether or not their performance could be improved (and how).

2. List of Applications

The following is a sample list of applications you may chose from If you have another application in mind, talk to the instructor about it. You may not, however, implement the same application as someone else in the class. A list of students and the applications chosen will be maintained on the web page for the assignment. Once you have an application in mind send me email telling me which application you'd like to implement and you are going to use an existing implementation or not and I'll hopefully approve the application and add you to the list.

LU Decomposition, Successive Over Relaxation*, Fast Fourier Transform*, Quicksort*, Traveling salesman problem*, Simplex, Water*, Jacobi*, Mean Value Analysis, Neural Networks and Cholesky Decomposition.

Note that most of these applications have already been implemented on a parallel system in one form or another. You may start with existing implementations and work from there. A few of these applications are provided with the Treadmarks distribution (the ones marked *). For those applications that are provided with the Treadmarks distribution, you will be expected to do more extensive experimental analysis of the application to compensate.

I'd really prefer that you don't use applications for which there are both Treadmarks and PVM implementation since that doesn't leave you with much to do except for conducting experiments and analyzing the performance. This might be slightly more acceptable if you then implement a version that uses a third parallel programming system called CRL, but I'll warn you that CRL hasn't been used that much here at York and it might be difficult to get help with if you run into difficulties. If you are concerned about not doing enough or doing too much please talk to me about your application and your approach.

3. Performance Analysis

The performance of the application you chose should be measured and evaluated using a number of criterion. One of these is the speedup of these applications on two, three, and four processors. However, some of these applications will not, for various reasons, obtain any speedup at all. Regardless of whether speedup is obtained or not, you should show performance measurements that show where the performance loss is occurring for multiple processors. This may be due to startup costs, synchronization costs, the computation being too fine-grained, etc.

4. What to Hand In

Hand in a report that describes the application in detail, including all of the major data structures and how these have been parallelized. Also describe the two different implementations of the application - the shared memory version and the message passing version. Finally, and most importantly, include in the report a measurement study of the application that describes the experimental design, how the application performs, what kind of speedup it obtains, and which components of the application are the cause of any loss in performance.