CISC 372: INTRODUCTION TO PARALLEL PROGRAMMING
Spring 2007
(Group) Lab 5
Due Date: October 22, 2007

1 Objectives

The main objectives of this project are to tackle a data decomposition problem in MPI parallel programming and to gain the experience of the project group environment common in the computer industry.

2 Problem Description

Your group has been charged with improving the runtime performance of an image processing program to enable faster computer visualization and object recognition. You should start with the sequential program that when given an image, returns an altered version of that image that delineates all the edges within the original image. This program is called an edge detection processor.

To get a feel for how the edge processor works, try running the edge detection processor located at: /usa/pollock/372porsche/public/edgedir07/ on the fast car cluster. In that same directory, there is a README file that explains how to get this edge detector up and running. There are also test files in the same directory, but you can have some fun with creating your own test files (read the README to see how).

It is important that your parallel version of this code be fast and correct; lives depend on your work! The specifications for your parallel program are as follows:

1. Your program should accept any arbitrary size image, for which the matrix to store the image is evenly divisible by 16. Process 0 should be the only process performing input/output to files.
2. Your program must perform parallel computations with necessary communications with an effective use of time. You are welcome to use any MPI command you find to achieve these goals, but it is perfectly fine to use only those we have discussed in class so far.
3. Your program should be as efficient as possible in space used by each process.
4. Your software must perform all computations correctly for 1, 4, 8 and 16 processes.
5. You must clearly and thoroughly document your code.

After you have a version of your program that you believe is the most efficient parallel program, also code an alternate data decomposition for the parallel program for comparison. You may choose any alternate that is very different from the one you already used.

3 What to Hand In

A tarfile that includes the following items, sent electronically to the TA rjk@udel.edu and me:

1. Your parallel code that uses good data decomposition and tries to achieve the best performance.
2. Your parallel code that has the alternate data decomposition.
3. A short written report that answers the questions:
   (a) What overall data distribution did you choose and why do you think this achieves good performance?
   (b) How did you handle the borders of the image and what solutions do you see?
   (c) How did you test your parallel program?

(d) What performance problems did you observe throughout your design and development phases? How did you try to address these challenges?

(e) How do the runs of your two different data decompositions compare? Explain the difference in the performance based on the data decompositions chosen. Do they perform relatively as you expected? Why or why not?

(f) How does load balancing come into play in this task?

(g) What characteristics are desirable in a good suite of inputs for testing your code thoroughly for both correctness and performance?

(h) What responsibility did each of your group members take on for this first deliverable?

(i) How do you compile and run your program?

(j) How did your group share the load of work for this project?

4. One script of a compile and run of your program on a data set that is included with your hand in.

5. One script of a compile and run of your alternate program on the same data set that is included with your hand in.

6. The input and output of the data set for which you scripted the run.

4 Criteria for Evaluation

Your project will be evaluated according to the following criteria:

1. (40 pts) Parallel program to spec above.
   25-correctly working parallel program
   15-both time and space efficient

2. (15 pts) An alternative data decomposition implementation.

3. (13 pts) Written report.

4. (2 pts) Materials handed in as requested, and TA able to run the code

5 Getting Help and Group Work

Your group should strive for a balanced workload among the group members in this project. Your code should reflect the efforts of your group members only. You can get help from the TA or instructor, and the MPI manual and textbook.