CISC 372: Introduction to Parallel Programming
Fall 2007
Individual Programming Assignment 3
"Where does the Sun Shine? Where are the Dark Shadows?"

Due dates:
Part (a) program and script only: Tuesday, October 2, 2007
Full Hand In assignment below including part (a) again: Friday, October 5, 2007

Individual Work: You should be doing this work for this lab on your own. You may ask the TA or instructor for help, but there should be no collaboration among students in this class or outside this class (with the exception of sharing and comparing input files or output generated by your programs.) **For this assignment, you may discuss with others how to best decompose the input matrix to minimize communication during the computation phase and delay time waiting on data. You need to write your own code in the end though!**

Objectives: The objective of this assignment is to write an MPI program that gives you experience using collective communication and explores different data decompositions and their resulting overheads.

Procedure:

Do exercise 6.12 on page 156 of your textbook. In this exercise, you will write an initial program that takes an input file with an N by N matrix of integers ranging from values of 0 through 100, representing points on a square piece of land. Each value represents the altitude of the land at the corresponding point. The output is a file with an N by N matrix of 0's and 1's, where a 0 indicates the point is in the sunlight and 1 indicates a shady spot.

Part (a) has the sun coming in directly from the west of the parcel of land, so shadows are all directly towards the east. Part (b) is a bit more complex as the sun is in the northwest, and so the shadows are headed toward the southeast.

Notes:

1. Process 0 should be the only process reading from the input file and printing to the output file.
2. All processes, including process 0, should be working on computation once they receive their data from process 0 (process 0 should not explicitly send to itself!). You may assume that N is evenly divisible by the number of processes, and you will only be running this program on 1, 2, 4, 8, and 16 processes.
3. Your goal should be to minimize the amount of communication between processes so your program can benefit from parallelism without undue overheads. To achieve this, you need to think about the best way to distribute the data among the processes so they don't have to send much data to each other once they have their own data.

4. Your goal should be to minimize the amount of space that is used by each process to only what they need to perform their computation. Each process does not need the entire matrix!
5. To really test how well your parallelism is doing, you should run this on a matrix of a very large size. We will provide you with some very large data sets for this. Initially, to check your correctness and not slow down our fast cars, use a data set of 16 by 16 or 32 by 32.
6. **HINTS:** Do program (a) first and get that working without thinking about program (b). Then, copy program (a) and revise the decomposition parts and then also the communication between processes during computation to achieve part (b). Draw pictures of what values are needed to compute the final output value for a given point before you start coding!

Hand In Electronically to the TA as a tar file:
To tar: tar cvf lastname.tar names of files to be submitted go here

a. Your 2 programs
b. Scripts of runs of each program on the data sets we will provide you soon (showing the compilation of your programs and the runs with 4 and 16 processes each). Be sure that your output goes to a file, and not the screen as it is too big!
c. A text file in which you write a paragraph of your justification for your data distribution choices for each program, and your observations from running your programs.

The TA will be compiling and running your program on a data set so he can see the output file that is generated and easily check correctness of output files.

Criteria for Evaluation:
a. Program 1 code to spec: 25 points
b. Program 2 code to spec: 25 points
c. Materials handed in as requested, and TA able to run the code: 2 points
d. Discussion: 3 points