Homework/Lab 5

due Friday, March 21, 2003, 4 PM

This assignment includes items to be done individually and items that you may do in groups of two. You will need to submit two files. `hw05.txt` should include your answers to the first two questions and must be done individually. The program for the third item should be submitted as `hw05.tar`. If you work in a group on the program, only one group member needs to submit `hw05.tar`. *Please use the filenames specified* and be sure to include your name in each file.

1. Log in to each of your accounts on NCSA and NPACI systems. Run at least one relatively simple MPI program (something more substantial than "Hello, world" but nothing too complex) on each system. Briefly describe the process you had to use to compile and run your program on each system. This question is to be answered individually. Include your answer in a plain text file `hw05.txt`. (4 points)

2. Jordan and Alaghband Problem 5.4, p. 219. This question is to be answered individually. Include your answer in a plain text file `hw05.txt`. (6 points)

3. (10 points) Modify one of the matrix-matrix multiply examples from the lecture notes that uses MPI for communication so that all three matrices exist only on the master process at the start and finish of the computation. Use MPI functions to distribute the entries of `A` and `B`, as needed, to the processes (include the master in the computation), do the multiplication, then use MPI functions to gather the result `C` back to the master.

   - You will need a number of MPI communication steps. Explain each MPI call (what you need to do and why you chose that particular MPI call) in detailed comments in your source code.
   - Run your program using 1, 2, 4, 8, and 16 processes on the compute nodes of the bullpen cluster. Submit these runs through PBS. Include the timings and your analysis of those timings in the `README` file that you include with your submission. You may work in groups of two on this program and its writeup.
     Your submitted tar file should include a subdirectory `matmult` that includes your `Makefile`, your C source code, your PBS script(s), and a brief `README` file that expains how to run your program, describes how and why you chose to parallelize your program, and describes and analyzes your timing results. Please do *not* include object files or your executable in your tar file.

4. (20 points) Convert your OpenMP program from the previous lab to a distributed memory program, using MPI for communication.

   Most of the requirements for the program are the same as those from the previous lab.

   - Again, you may break down the computation any way you'd like. Justify your choice in the `README` file that you include in your submission.

- You will need a number of MPI communication steps. Explain each MPI call (what you need to do and why you chose that particular MPI call) in detailed comments in your source code.
- Overlap your computation and communication, where possible. Point out these areas of overlap in comments in your source code.
- Run your program using 1, 2, 4, 8, and 16 processes on the compute nodes of the bullpen cluster. Submit these runs through PBS. Include the timings and your analysis of those timings in the `README` file that you include with your submission.

  You may work in groups of two on this program and its writeup.

  Your submitted tar file should include a subdirectory `laplace` that includes your `Makefile`, your C source code, your PBS script(s), and a brief `README` file that expains how to run your program, describes how and why you chose to parallelize your program, and describes and analyzes your timing results. Please do *not* include object files or your executable in your tar file.

Honor code guidelines: While the programs are to be done only by you (meaning your group, if you choose to work in a group), along the lines of a **laboratory program**, I want to encourage you to ask questions and discuss the programs with me, our TA, and with classmates outside your group, as you develop it. However, no sharing of code between groups is permitted. If you have any doubts, please check first and avoid honor code problems later.

Grading guidelines: Your grade for the programs will be determined by correctness, design, documentation, and style, as well as the required items in the detailed comments and `README` file.