

**Fall 2005 CS 441/632/732 Parallel Computing
Homework-4**

Individual work only. 200 points. Due Dec 5, 2005 before 5:00 p.m.

1. Implement the parallel version of the Mandelbrot program using MPI message passing primitives and 2-D data distribution for the following load balancing schemes:
 - a. static load balancing – divide the computation into fixed blocks based on the process grid layout. If the process grid layout is $P \times Q$, image size is $M \times N$, then each process is assigned $M/P * N/Q$ (if P and Q are not divisible exactly by M and N , respectively, then distribute the additional $R=M\%P$ rows and $C=N\%Q$ columns evenly among the first R process rows and C process columns). Measure the performance for the following process grid layouts: 2×8 , 4×4 , and 8×2 .
 - b. dynamic load balancing – assign each process computation of $K * M/P$ rows and $K * N/Q$ columns, where $K = 0.25, 0.5, 0.75$, and 1 ; $M \times N$ is image size; and $P \times Q$ is the process grid layout. Measure the performance for different task assignments and the following process grid layouts: 2×8 , 4×4 , and 8×2 [Note: When $K=1.0$, problem 1(b) reduces to problem 1(a)].

First download the sequential program provided, follow the instructions given to compile and run it, and use this program to develop the parallel versions. Compare the performance of static load balancing and dynamic load balancing schemes for different number of processes. Complete the following table:

| Process Grid Layout | Time Taken (seconds) | | | | |
|---------------------|-----------------------|------------------------|-------|--------|-------|
| | Static Load Balancing | Dynamic Load Balancing | | | |
| | | K=0.25 | K=0.5 | K=0.75 | K=1.0 |
| 2x3 | | | | | |
| 3x2 | | | | | |
| 2x8 | | | | | |
| 4x4 | | | | | |
| 8x2 | | | | | |

2. Download, compile, and run the MPI-IO example. Modify program 1(a) to include MPI I/O function calls to write the image to the disk directly from each process instead of sending individual pieces to process with rank 0 and then writing the image to a file. Measure the time taken for the following process grid layouts: 2×1 , 4×1 , 6×1 , 8×1 , and 10×1 . Also compare the performance results between program 1(a) and program 2 (you have to run program 1(a) for the corresponding process grid layouts). Complete the following table:

| Process Grid Layout | Time Taken (seconds) | |
|---------------------|----------------------|--------------|
| | Without MPI I/O | With MPI I/O |
| 2x1 | | |
| 4x1 | | |
| 6x1 | | |
| 8x1 | | |
| 10x1 | | |

3. Derive expressions to compute the communication cost involved in the implementation of broadcast function on a 2-D mesh and a 2-D torus for the pipelined and recursive doubling implementations [This question is optional for undergraduate students and **required** for both M.S. and Ph.D. students].

**Fall 2005 CS 441/632/732 Parallel Computing
Homework-4**

4. [BONUS QUESTION]

Implement the parallel version of the Jacobi method to solve a system of linear equations using 2-D data distribution. First write the sequential program using the pseudo code provided in the text book. Compile and test it, and then use the driver program provided to develop the parallel version. Test the program for matrix size 1000x1000 and 5000x5000 for the following process grid layouts: 2x2, 2x3, 2x8, 3x2, 3x3, 4x4, and 8x2. Complete the following table:

| Process Grid Layout | Time Taken (seconds) | |
|---------------------|----------------------|-----------|
| | 1000x1000 | 5000x5000 |
| 2x2 | | |
| 2x3 | | |
| 2x8 | | |
| 3x2 | | |
| 3x3 | | |
| 4x4 | | |
| 8x2 | | |

General Comments:

You must implement and test programs (1) and (2) on the CIS cluster (Everest) and use MPI for communication. Instructions for using the CIS cluster and submitting jobs to SGE can be found at: <http://www.cis.uab.edu/ccl/resources/everest/EverestGridNodeUserGuide.php>.

Submission:

Email the source code along with any Makefile and scripts as a single tar file attachment to puri@cis.uab.edu with the subject "CS 441/632/732 Homework-4." Turn-in a printed report in class using the format provided at <http://www.cis.uab.edu/cs441/report.html>. After submission, do not make any changes to your source code on Everest, you will be asked to demonstrate your program on Everest and the timestamp of the files will be used to determine late submissions.

Grading:

| | |
|---|------------|
| Correct implementation and testing of the programs (including collecting timing information for the tables above) | 150 points |
| Performance Analysis | 35 points |
| Lab report format/presentation | 15 points |