

CS 695 High Performance Computing

Fall 2007

Homework 4

Due: Thurs Nov 15, 2007

Note. There are two pages (and four questions) on this assignment.

In this programming assignment, your task is to design a parallel algorithm for doing (dense) matrix-vector multiplication (MatVec) $AB = C$, and to implement your algorithm using C (or C++) and the MPI communication library. To help you get started, Mahantesh Halappanavar has written a simple C code for performing MatVec when the matrix A is partitioned by rows. (The code is available in the course directory.) A master-slave computing model is employed, in which process 0 is the Master, and it generates the matrix A by rows and sends blocks (of rows) to the processors (Slaves), broadcasts B to all other processors, receives the vector C from the Slaves, and prints it.

1. Modify the code (or write your own) to incorporate timing routines so that you can report the time taken to do the computation. When you run on large problems, turn off the printing of the vector C (but use it for debugging purposes). Run this code on four processes all on one processor (so you could use the `qssh` command to run interactively). Run your code on matrices of order (i.e., number of rows or columns) equal to 2048, 4096, 8192, and 16,384, and report the results.
2. Now write an algorithm that does the matrix vector multiplication by partitioning the *columns* of A onto the processes. The vector B needs to be distributed by the Master process to the Slaves as needed; each process computes a partial product, and then use a reduction operation (`MPI_Reduce` for the Master process to compute the product C .

Submit your code electronically or put it in a directory from where I can copy it.

Time your code on four processes all running on one processor, and compare the times for matrices of the same size as in the previous part.

3. Now run your code on 2, 4, and 6 processes (running on different processors) using the `qsub` command, using the submission script that you have used in an earlier assignment. Report the times on *both* the Infiniband and Gigabit ethernet networks.

4. Try to improve the performance of your code by identifying any inefficiencies (the code provided by Mahantesh is not efficient, so you should be able to do much better). List these improvements and report the best times you are able to achieve.