American University of Beirut
الجامعة الأميركية في بيروت

Faculty of Arts and Sciences *Department of Computer Science*

CMPS 297M – Parallel Computing
Assignment 03
Spring 2013-14

Due Date

Friday, March 28th, 5:00 pm

Grade distribution

40% correctness of output, 40% good parallel performance, 20% good programming style.

Question 1 (30%)

Recall that 2D arrays can be dynamically allocated in one of two ways. Given an *n* by *m* array, one can allocate it either:

1. using a single malloc which yields a large one dimensional array of values onto which we can map two dimensional accesses.
2. Or using *n* mallocs, one for each row, plus one malloc for an array of row arrays.

Examples can be found in many standard pages, see below as one out of many:

http://www.cs.swarthmore.edu/~newhall/unixhelp/C_arrays.html#dynamic2D

a. Write a BSPlib program where a 2D matrix is allocated using the first method. Processor *p(0)* then initializes the entries of the matrix and sends it to all other processors in action.

b. Can you repeat the above exercise but where the 2D matrix is allocated using the second method? Investigate the possibilities and pay attention to pitfallas associated with push-registering in BSPlib.

Question 2 (35%)

Write an MPI program where two processors *p(0)* and *p(1)* initialize an *n* dimensional row vector each, say *A* and *B* respectively. The program should then have *p(0)* and *p(1)*

swap the contents of *A* and *B*. Make sure to address the potential for deadlock.

Question 3 (35%)

Consider two *m* by m matrices, *A* and *B,* and *p* processors in 2D numbering.

a. Implement the naïve matrix multiplication discussed in class using MPI. This is the version where processor *p(0,0)* sends to each other processor *p(i.j)* the block of full rows indexed by $im/\sqrt{p},....,((i+1)m/\sqrt{p})-1$ and the block of full columns indexed by $jm/\sqrt{p},....,((j+1)m/\sqrt{p})-1$. Use MPI_Send/MPI_Recv.

b. Test your program in distributed setting mode on the hpc cluster.

c. Try input instances as large as your dynamic allocation will give you. For as many processing nodes as you can acquire on the cluster in distributed mode, plot the efficiency of your program.

d. Is efficiency near ideal? What could be hindering scalability? Discuss in light of communication volume as discussed in class.