

EECS 739: Homework 3

Due: Tuesday, April 5, 2016 (By 11am)

Reminder: Please review the homework policies on the syllabus before working on the assignment.

Questions:

1. (20 points) Consider the multiplication of two $n \times n$ dense matrices A and B with $A \times B = C$. Design and implement in MPI and C/C++ an efficient parallel matrix-matrix multiplication algorithm for the case where $n = cp$ and A is partitioned among p processors laid out in a logical square 2D mesh using 1D rowwise block cyclic partitioning.

Demonstrate that your code obtains correct results on 1 and 4 processors when multiplying two small dense matrices. **This verification should occur in your code. In addition, print A , B , and C for this small example.**

Generate timing results for performing the multiplication of two 1000×1000 matrices on 4 processors.

You must run your code on the slurm cluster as demonstrated in class.

Submit the following items electronically via e-mail to shontz@ku.edu: (1) your MPI and C/C++ code and (2) your PBS script.

In addition, submit the following items in hard copy: (1) your MPI and C/C++ code, (2) your PBS script, (3) output demonstrating your code works (as indicated above), and (4) output containing your timing results.

Both the electronic and hard copy submissions are required in order to obtain credit for this question.

2. (20 points) For this question, you will run scaling experiments using the code you implemented in the previous question.
 - (a) Perform a strong scaling experiment using fixed matrices that are somewhere between 1000×1000 and $10,000 \times 10,000$ in size. For this experiment, you should run your code on 1, 2, 4, 8, and 16 nodes and obtain timing results.
Generate a plot of the speedup versus the number of processors. In addition, plot the ideal speedup in the same figure.
 - (b) Perform a weak scaling experiment starting with matrices that are 1000×1000 in size. For this experiment, you should run your code on 1, 2, 4, 8, and 16 nodes and obtain timing results.
Generate a table with the following columns: n , p , and T_p summarizing your results.

Turn in the raw data for your timing results, your plot from (a), and your table from (b).

(OVER →)

3. (20 points; 4 points each) Consider an algorithm with problem size, $W = n^2$, serial run time, $T_s = n^2 t_c$, and parallel run time, $T_p = t_c(n^2/p) + t_s \log p + t_w n$, where t_c is the amount of time taken to perform a unit of computation, t_s is the start-up time of the communication network, and t_w is the per-word transfer time.
- (a) Determine an expression for the speedup, S .
 - (b) Determine an expression for the efficiency, E .
 - (c) Determine an expression for the total overhead, T_0 .
 - (d) Determine the isoefficiency function for this algorithm.
 - (e) Explain how to interpret the isoefficiency result.