Last 4-digits of Student ID: _______________

Northwestern University
Department of Electrical and Computer Engineering
EECS 358 Introduction to Parallel Computing

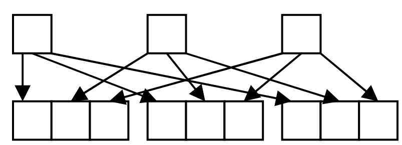
Question	Score
1	/ 20
2	/ 20
3	/ 30
4	/ 30
Total	/ 100

NOTE: Please show your work clearly for all the questions. Use the space provided for your answers as much as possible. If necessary feel free to use the back of the sheets and/or additional sheets.

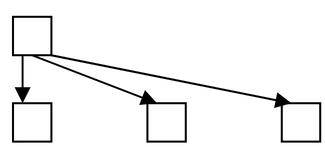
1) [20 pts]

a) [5 pts.] Match the MPI routines on the left side to the operations on the right:

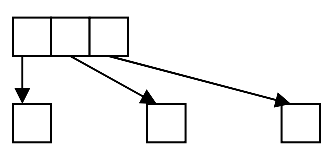
MPI_BCAST



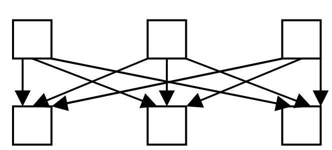
MPI_SCATTER



MPI_ALLREDUCE



MPI_ALLGATHER

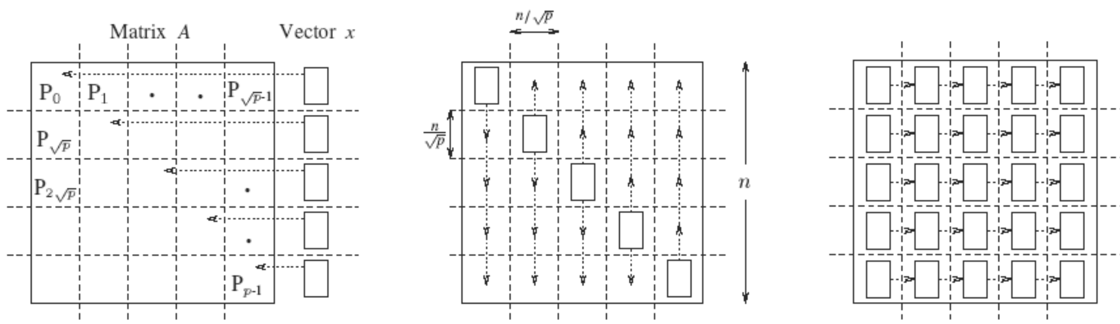


b) [5 pts] Define a Cartesian topology to match the physical layout of a machine with p processors and a torus interconnect structure.

c) [10 pts] Assume that a matrix is partitioned using a block-checkerboard distribution (slide 15.6-a). In this configuration, assume that we have to distribute each row in a cyclic manner among the processors in that row. Specifically, element (0,0) will be sent to P0, (0,1) will be sent to P1, (0,2) will be sent to P2, …, (0, $\sqrt{p}$) will be sent to P$\sqrt{p}$, (0, $\sqrt{p}$ +1) will be sent to P0, …. Note that the same destinations will be used for the following rows (i.e., (1,0) will be sent to P0, (1,1) will be sent to P1, etc.). Define a data type to perform the communication between processors more effectively.

Question 2) [20 pts.] Figure 1 presents a checkerboard data distribution for the matrix vector multiplication (Slide 15.13). In this distribution, every processor receives a block of size n/sqrt(p) by n/sqrt(p). Assuming that initially both the vector and the matrix is stored at processor 0, list the major communication tasks in terms of all_to_all broadcast, one_to_all broadcast, all_to_all scatter, …, etc. and find an estimate of the communication time in terms of p, n, t_s, and t_w if the operations are performed on an hypercube interconnect network using store-and-forward to do the following operations:

a) Distribute the data from processor 0 to all the other processors (Figure 1(a)),
b) Perform the vector distribution among the processors in each column as shown in Figure 1(b), and
c) Perform the single node accumulation of partial results (Figure 1(c)).



(a) (b) (c)

Figure 1. Matrix Vector Multiplication using Checkerboard Partitioning

$\frac{n^2}{p}$

Question 3) [30 pts.]

The Poisson equation can be numerically solved over a region by discretizing it in the x and y directions to obtain a grid of points and then computing the approximate solution values at these points. In that case, the values at each matrix point can be iteratively found using the values of its neighbors (as shown in Figure 2, slide 12.4) from the previous iteration.

In this question, you are asked to compare two partitioning mechanisms (assuming that x and y are equal to n):

1) Row-wise: Each processor is assigned a number of rows (hence each processor gets 'n / p' rows).

2) Checkerboard: Each processor is assigned a block of size 'n/ $\sqrt{p}$ ' by 'n/ $\sqrt{p}$ '.

Find the cost of one iteration of the algorithm in terms of number of processors p, n, t_s, and t_w on a mesh structure using store-and-forward (omit the cost of computation). One iteration corresponds to calculating the values and sending/receiving them to the neighboring processors.

In a single step, a processor can only send and receive a single packet (i.e., you can send a message to a processor while receiving a message from another).

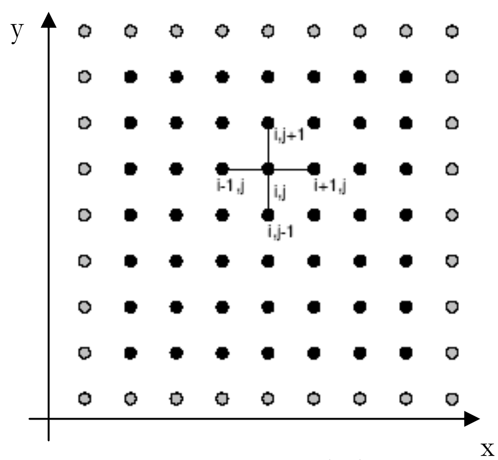
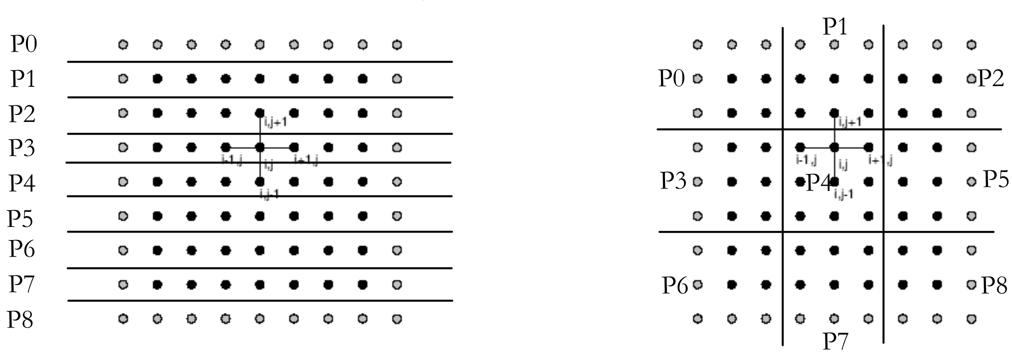


Figure 2. Poisson calculation.



(a) (b)

Figure 3. Two distributions: a) row-wise, b) checkerboard

Question 4) [30 points]

Halma is a board game invented in around 1880. The game is roughly based on Chinese Checkers.

In this game, each player uses a different colored set of marbles and places them into a checker in the checkerboard. The players have N marbles each and initially they are placed into opposite corners of the checkerboard. The idea of the game is to manipulate your marbles across the board to occupy the squares of your opponent. In other words, the objective is to move all your pieces from your corner into the opposing corner. Each turn, a piece is moved to the adjacent square or, if possible, a piece can be moved by hopping over an adjacent piece (your piece or opponent's piece) into the hole on the other side (jump). If possible, multiple jumps can be done in a single turn for the same marble, i.e. from the hole you jumped in you can hop over another adjacent piece if such exists and so on.

You *never* remove a marble; all marbles stay on playing board. The player getting all marbles across first wins.

One can move or jump in only "positive" directions. For example, if you have started from the lower-right corner, you can only move towards left or up. Similarly, you can only jump in the direction of left or up.

In this question, you are asked to write the computer program to play against a human player.

a). [25 pts.] Describe a parallel best first search algorithm that makes decisions for guiding a computer to play against a human player. *Note that, at each turn, a marble has at most two possible moves. In addition, some marbles will have no possible moves.* Even while considering the jumps, you can still assume that the marble has two possible moves (assume that if you make a jump in a direction only one other jump direction is available and you will always make the consecutive jump if it is available).

b) [5 pts.] What is the search overhead factor for the algorithm?