National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Parallel and Distributing Computing	Course Code:	CS3006
Degree Program:	BS (CS)	Semester:	Spring 2022
Exam Duration:	60 Minutes	Total Marks:	30
Paper Date:	21/03/22	Weight	15
Exam Type:	Midterm I	Page(s):	4

Student : Name: ***SOLUTION*** **Roll No._____ Section:_____**

Instruction: Attempt all questions on the question paper. Rough sheets can be used but it should not be attached. If you think some information is missing then assume it and mention it clearly.

Question 1: MCQs & True/False **[10 marks]**

i. Single Program with Multiple Data (SPMD) is a close variant of _____ in Flynn taxonomy:

a. SIMD
b. MIMD *
c. MISD
d. SISD

ii. In case of _____ decomposition, divide and conquer strategy is used.

a. Recursive *
b. data
c. exploratory
d. Speculative

iii. In Bus network, the number of shared connections is equal to

a. 1
b. n *
c. $\log_2 n$
d. n^2

iv. The number of tasks that can be executed in parallel is called _____.

a. dependency
b. degree of concurrency *
c. granularity
d. All of the above

v. Which one is true about Amdahl's law formulation: -

a. It does not account for communication overheads
b. It provides an upper-bound on the achievable speedups
c. If the number of processors approaches infinity, Speedup is bounded by $\frac{1}{sequential\ fraction}$
d. All (a), (b), and (c) are correct *
e. Only (b) and (c) are correct

vi. Choose correct statement/s regarding NUMA architecture

a. Access time may vary depending on the data location *
b. Different processors interact with each other using message passing
c. The same logical address on different processors maps to the same physical memory location *
d. The same logical address on different processors maps to the different physical memory locations

vii. How is a network of workstations different from a cluster?

a. The workstations in such a network may have different operating systems
b. The nodes within a network will normally be co-located (located together)
c. Users normally have the power to login to individual workstations
d. Both (a) and (c) *

viii. In distributed Systems, processors can share logical address space

a. True
b. False *

ix. Mesh network provides better cost scalability but poor performance scalability than the bus network

a. True
b. False *

x. In case of Non-uniform Memory access, a processor cannot directly access the memory associated with another processor

a. True *
b. False

Question 2: Differentiate between **[4 marks]**

a. Exploratory and Speculative decomposition

Exploratory: Specially used to decompose the problems having underlying computation like search-space exploration.

We may use these steps:
(1) Partition the search space into smaller parts;
(2) Search each one of these parts concurrently, until the desired solutions are found

Speculative: Usually used in the problems where different input values or output of the previous stage causes many computationally intensive branches.

Possible steps:

(1) Speculate (guess) the output of previous stage
(2) Start performing computations in the next stage even before the completion of the previous stage.
(3) ***After the output of the previous stage is available, if the speculation was correct, then most of the computation for the next step would have already been done.***

b. Static and Dynamic task mapping

Static: when the size and number of tasks are known apriori, we prefer static mapping. This mapping of tasks to processes is done before program execution.

Dynamic: when the size and number of tasks are not known apriori, we prefer dynamic mapping. This mapping of tasks to processes is done during program execution.

Question 3: Assume a sequential program *S* has an execution time of *200* seconds. Further, assume that S_p is a parallel variant of *S*. After an experimental evaluation over different number of processors, the following running times were achieved: - [4+4 marks]

P	2	4	6	8
Execution Time (seconds)	107	61.92	48.31	42.46
Speedup	*1.87*	*3.23*	*4.14*	*4.71*
Karp-Flatt Metric	*0.07*	*0.08*	*0.09*	*0.10*

a) **Calculate Speedups for each of the experimental configurations in the space provided below and then write your answers in the table above.**

200 / 107 = 1.869158 = 1.87

200 / 61.92 = 3.229974 = 3.23

200 / 48.31 = 4.1399296 = 4.14

200 / 42.46 =4.710315 = 4.71

b) **Calculate the Karp-Flatt metric values in the space provided below and then write your answers in the table above. You also have to Interpret the results of Karp-Flatt metric and write your opinion below**

((1/1.87) – (1/2)) / (1 – 1/2) = 0.06951871657 = 0.07

((1/3.23) – (1/4)) / (1 – 1/4) = 0.07946336429= 0.08

((1/4.14) – (1/6)) / (1 – 1/6) = 0.08985507246= 0.09

((1/4.71) – (1/8)) / (1 – 1/8) = 0.09978768577 = 0.1

As 'e' is steadily increasing with p, it suggests that parallelization overhead is also contributing to poor speedup. Therefore, we need to reduce this overhead to improve speedups.

Question 4: For this question, you have a choice to attempt any one of the following questions: [8 marks]

a) Derive generalized expressions for arc-connectivity, diameter, bisection-width, and link cost of a 2D mesh (without wraparound) having M rows and N columns.

OR

b) Write a multithreaded program either using Open MP or Pthreads to find the maximum value in a matrix of size n x m. (Minimum number of threads should be 2 and maximum could be n)

(a)

Arc connectivity = 2 (each arc is connected to two nodes)

Diameter = (Row distance + Column distance) = (N-1) + (M-1) = M + N – 2

Bisection width = Minimum (M, N) (i.e., the min. number of links that must be removed to partition in equal halves)

Cost = (Links in a Row)(Total Rows) + (Links in a column)(Total Columns)

= (N-1)(M) + (M-1)(N) = MN – M + MN – N = 2MN – (M + N)

(b) *One Possible solution:*

```
pthread_mutex_lock maxlock; // global variables
int max -10000;
int** matrix;

void function(void* arg) {
  int row = * (int*) arg;
  int localMax = -1000;
  for(int i=0; i<M; i++) {
    if(matrix[row][i] > localMax) localMax = matrix[row][i];
  }
  mutex_lock(maxLock);
  if(localMax > max) max = localMax;
  mutex_unlock(maxLock);
  pthread_exit(NULL);
}

int main() {
  int N, M; // values from user
  matrix = new int*[N];
  for(int i=0; i<N; i++) matrix[i] = new int[M];
  // matrix is filled by user
  pthread_t* threads;
  threads = new pthread_t[N];

  for(int i=0; i<N; i++) pthread_create(threads[i], NULL, function, (void *) i);
  for(int i=0; i<N; i++) pthread_join(&threads[i], NULL);

  cout << "Max " << max << endl;
  delete [] threads;
  for(int i=0; i<N; i++) delete [] matrix[i];
  delete matrix;
  return 0;
}
```