National University of Computer and Emerging Sciences
Lahore Campus

Parallel and Distributed Computing (CS3006)

Date: February 24th, 2025

Sessional-I Exam

Total Time (Hrs.): 1
Total Marks: 60
Total Questions: 6

Course Instructor(s)

Dr. Abdul Qadeer
Mr. Danyal Farhat
Mr. Raja Muzammil Muneer
Ms. Saba Tariq
Dr. Syed Muhammad Irteza
Dr. Zeeshan Ali Khan

_____________ _____________ _________________________

Roll No Section Student Signature

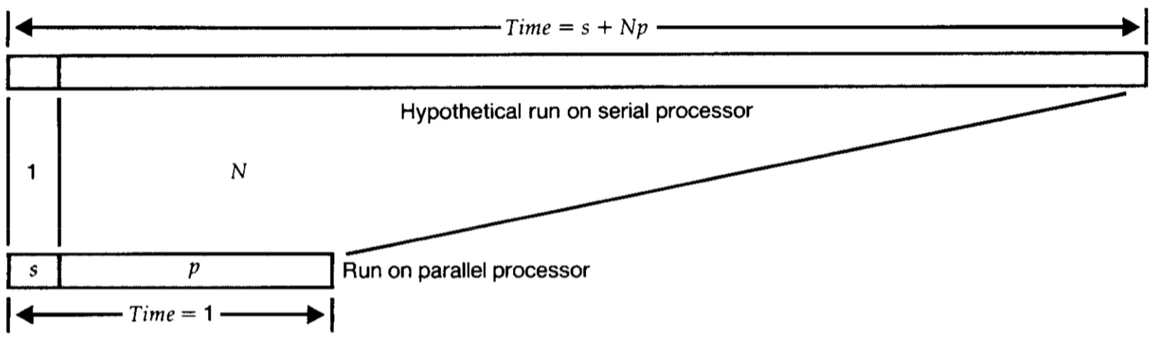
Instructions / notes:

1. Attempt all questions on the answer sheet.
2. If a question is ambiguous, note it, assume clarifying details, and solve it.
3. Include all work with your answer; do not use a separate rough sheet.
4. We estimate that you need no more than 10 minutes on any question. Pace yourself accordingly.

CLO 3: **Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.**

Q1a: [2 marks] How does Gustafson's law differ from Amdahl's law?

Q1b: [4 marks] Derive Gustafson's law using the symbols provided in the diagram below. Clearly mention what each symbol means. Your final expression should be in terms of *N* and *s* only.



Q1c: [2 marks] What are the two assumptions made in the above derivation?

Solution for Q1:

Q2a: In Gustafson's Law we increase the input data with increasing processors. In Amdahl's law we keep the input data same across runs with increasing processors.

Q2b: We are increasing data that can be used in parallelization in proportion to the number of processors (N). Assuming a single processor will take N times more time to finish the work, the time of a hypothetical serial run = s (serial proportion of time) + N*p
We assume that using N processors the time will be: s + p = 1

Scaled speed-up is as follows:

$$\text{Scaled speedup} = (s + p \times N)/(s + p)$$

$$= s + p \times N \quad \text{Because } s+p = 1$$

This is a line having slope (1-N)

$$= N + (1 - N) \times s. \quad \text{Put } p = (1-s) \text{ in above equation and simplify.}$$

s = serial fraction
p = parallel fraction
N = number of processors

Q2c: We assumed that each processor is equally capable and can divide and finish its allocated work at the same rate.
We assumed there are no overheads and data-based scaling will perfectly continue with increasing processors.

CLO 3: **Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.**

Q2a: [2 marks] Write Amdahl's law and Karp-Flat metric (K-F metric) formula (no derivation needed).

Q2b: [4 marks] Assume that 90% of the algorithm can be parallelized while the remaining 10% must be executed sequentially. We have partially filled the following table for you with 4 missing values (A to D). Calculate these four missing values using formulae you mentioned above.

Number of Processors used	2	4	6	8
Amdahl's speedup	**A**	3.077	4	4.706
Gustafson's speedup	**B**	3.7	5.5	7.3
K-F metric using Amdahl's speedup	**C**	0.1	0.1	0.1
K-F metric using Gustafson's speedup	**D**	0.027	0.018	0.014

Q2c: [2 marks] Review the K-F metric derived from Amdahl's speedup in your completed table. What conclusions can you draw, and why?

Q2d: [2 marks] Review the K-F metric derived from Gustafson's speedup in your completed table. What conclusions can you draw, and why?

Q2e: [2 marks] Above we calculated K-F metric values using speed-up values provided by Amdahl's or Gustafson's instead of getting the speed-up values by empirically running the code. If we had experimentally computed K-F metric values what trend of K-F values will you expect and why (only for Amdahl's case)?

Solution for Q2:

Q2a: Amdahl's law: Speed-up = 1/(F + (1-F)/N). where F = serial fraction, N = no of processors
K-P Metric = f = (N/speed-up – 1) /(N – 1), N = number of processors

Q2b:
A = 1.818
B = 1.9
C = 0.1
D = 0.053

(Q2c) K-F metric using Amdahl's speedup: serial fraction remains constant as we increase the number of processors as the problem size is fixed. Because we used speed-up calculated vis Amdahl's (and not the empirical speed-up observed), K-F is not able to show any overheads.

(Q2d) K-F metric using Gustafson's speedup: serial fraction decreases with scaled speedup since parallelizable workload increases in fraction.

Q2e: Experimental speedups consider overheads i.e. data parallelization overheads which theoretical speedup do not. So we expect that K-F values will increase.

CLO 3: **Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.**

Q3a: [2 marks] You are designing a cloud-based image processing service where each uploaded image is independently transformed (e.g., resized and enhanced). With thousands of concurrent uploads, which Flynn's Taxonomy architecture best supports parallelism and scalability? (Just name it.) Provide rationale for your answer in a line.

Q3b: [8 marks] You're optimizing a scientific computation program and profiling reveals that 70% of the work is parallelizable while 30% remains sequential. You plan to rent a server from a cloud provider that offers machines ranging from 2 to 500 cores. While you might be tempted to get a server with 500 cores assuming each extra core will give you some speed-up, but you want an informed decision. You only like speed-ups in whole numbers (such as 5, 6, etc. and not fractions like 5.3 etc.). What core count should you get?

Solution for Q3:

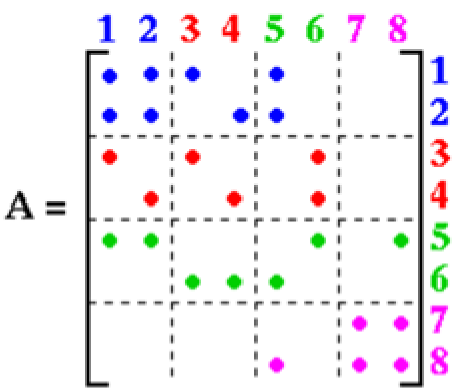
Q3a) Multiple answers are possible:
MIMD: Use multiple independent processors working on independent pictures for best performance and scalability

SPMD or SIMD: Since Flynn's taxonomy is quite loose, students can say that it resembles SIMD because all processing elements are doing the same operation but on independent data.

Q3b): We know from Amdahl's that the max speed-up can be 1/F even if we use infinite processors. So that number turns out to be 3.33333… Now as per question, we want to target speed up of 3. That means by using that value we get answer of 21 cores.

CLO 3: **Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.**

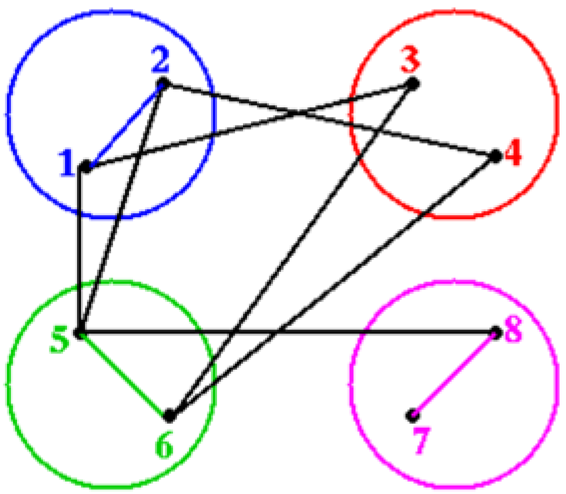
Q4a: [6 marks] Consider the problem of computing the product y = Ab, where A is a sparse 8 x 8 matrix and b is a dense 8 x 1 vector. A matrix is considered sparse when a significant number of its entries are zero, and the locations of the non-zero entries do not conform to a predefined structure or pattern. The matric A is shown on the right, with the non-zero elements represented by dots. Create a task interaction graph for this problem, such that i^{th} task computes entry of y[i] and owns the i^{th} row of A and the element b[i].



Q4b: [4 marks] Map the tasks created in part (a) to four processes, such that two consecutive tasks are assigned to one process. Calculate the list C_i, where i = 1 to 4, which contains the indices of b that the tasks on Process i need to access from tasks mapped to other processes.

Solution of Q4:

Solution for Q1, parts (a) and (b)



C1 (blue colored) = (3,4,5)
C2 (red colored) = (1,2,6)
C3 (green colored) = (1,2,3,4,8)
C4 (purple colored) = (5)

CLO 1: **Demonstrate understanding of various concepts involved in parallel and distributed computer architectures.**

Q5 a: [2 marks] Draw a 4x4 Omega network for processors P0–P3 and memory modules M0–M3.

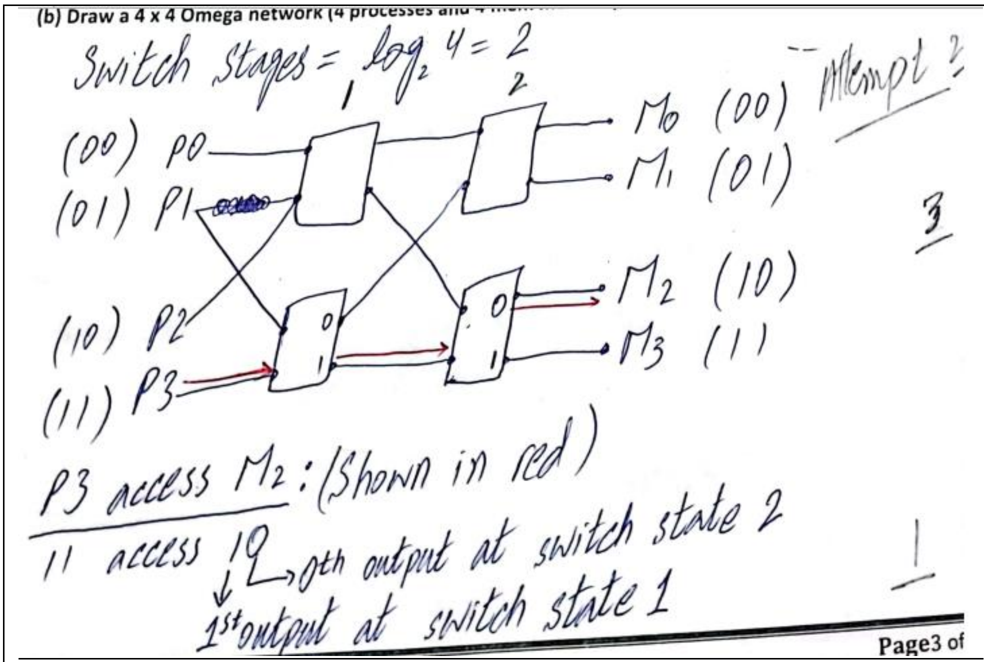
Q5 b: [2 marks] Explain how P3 will access M2 in terms of routing?

Q5 c: [2 marks] How many switching stages will be required for this network?

Q5 d: [2 marks] Can P1 to M2 and P3 to M3 communications happen concurrently? Why or why not?

Q5 e: [2 marks] How many switching nodes will be required for this network?

Solution of Q5:

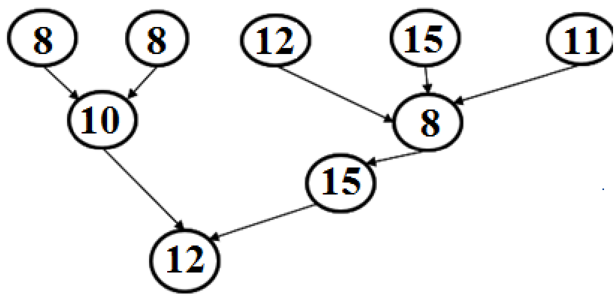


(c) Switching Stages = 2
(d) Not possible. There is a conflicting link on the path between bottom two switches.
(e)Switching Nodes = (p / 2) x log (p) = (4/2) x (log (4)) = 2 x 2 = 4

CLO 3: **Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.**

Q6: [2+2+2+2+2 marks] Examine the task-dependence graph on the right and compute:

(a) Maximum degree of concurrency
(b) Any Single Critical Path Work
(c) Average degree of concurrency
(d) Minimum number of processors needed to obtain the maximum possible speedup
(e) Maximum speed up if number of processors are limited to 4



Solution of Q6:

(i) Maximum degree of concurrency
Maximum degree of concurrency = 5

(ii) Any Single Critical Path Length
Critical Path Length = 15 + 8 + 15 + 12 = 50

(iii) Average degree of concurrency

Average degree of concurrency = (8 + 8 + 12 + 15 + 11 + 10 + 8 + 15 + 12)/50 = 99/50 = 1.98

(iv) Minimum number of processors needed to obtain the maximum possible speedup

Minimum number of processors needed to obtain the maximum possible speedup = 5

(v) Maximum speedup if number of processors are limited to 4

Maximum speedup if number of processors are limited to 4 = 9 / 5 = 1.80