End Semester Examination
Distributed Systems
Monsoon 2025

Instructions

Please read the following instructions carefully.

- The exam is for a duration of 180 minutes and is for a maximum of 80 points.
- The question paper has **FOUR** printed pages. Check your copy to see that you have all the questions printed properly.
- Fully state any assumptions you make while answering the questions.
- Answer all parts of a question contiguously.
- Too verbose answers may attract a penalty.

Section A: Short Answer Questions

Answer the following questions in brief. Usually, one point requries about two to three sentences of written answer. This section is for a total of 20 points across eight questions.

1. Write about user-oriented design philosophy versus workload-oriented design philosophy of designing distributed systems. **(Points: 2.5) (COs: 3)**
2. Find the scalar time of events c, j, and k, from Figure 1. Show that scalar time is not strongly consistent by exhibiting a pair of events that violate strong consistency. You can use an increment of 1 to the time across all processors. **(Points: 1.5+1=2.5)(COs: 1)**
3. Define in symbols and words what is meant by termination detection. What are some difficulties of solving this problem in the distributed setting. **(Points: 1.5+1=2.5) (COs: 1)**
4. Explain the differences between causal and FIFO message delivery along with a suitable example. **(Points: 1.5+1=2.5) (COs: 1)**
5. State and explain the CAP theorem in brief. Notice that the authentication service for our email, and SSO, runs as a distribtued service. What aspects of the CAP theorem should it adhere to? Justifiy your answer. **(Points: 1.5+1=2.5)(COs: 6)**

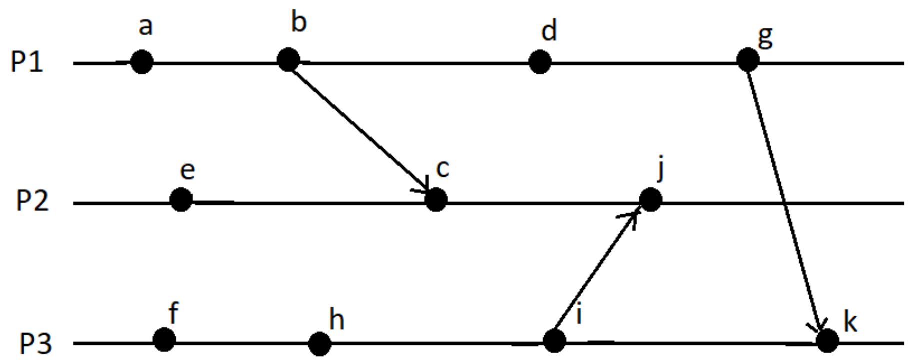

Figure 1: Figure for Question 2, Section A.

6. Explain the semantics that GFS ensures for the WriteAppend operation. Elaborate on aspects of these semantics are markedly different from a typical file system write operation? **(Points: 1+1.5=2.5) (COs: 3)**

7. Explain the notions of accidental compelxity and inherent complexity in the context of distributed programming. **(Points: 2.5) (COs: 2)**

8. What is meant by the global state of a distributed sytsem. When is a global state said to be consistent. **(Points: 1+1.5=2.5)(COs: 1)**

Section B: Long Answer Questions

Questions in this section require detailed answers. This section is for a total of 60 points.

1. Recall the problem of finding the kth smallest element in a given set of N elements. The algorithm groups the elements into groups of 5 elements each, finds the median element in each group, and finds the median of the $N/5$ elements recursively. This median of the medians is used as a pivot to rearrange/parition the input elements. Depending on the value of k, the algorithm then uses recursion to find the required element in either the set of elements less than the pivot or in the set of elements bigger than the pivot. If k equals the rank of the pivot, then the algorithm can finish without a further recursive call.

   Now, consider redesigning the above algorithm in the MPC model with each machine having a space of $S = N^\epsilon$ for a constant $0 < \epsilon < 1$. Answer the following question that help you along the solution.

   (a) How is the input stored?
   (b) Explain the mechanism needed to find the median in each group.
   (c) Write the pseudocode for the algorithm in the MPC setting.
   (d) Explain how to partition the input elements according to the pivot mentioned.
   (e) Write a recurrence relation that governs the number of rounds needed by the algorithm in the MPC setting.
   (f) Solve the recurrence relation.

(Points: 1+1+2+3+2+1=10) (COs: 5)

2. Consider a system running the algorithm of Ricart and Agarwala for mutual exclusion. Answer the following questions.

   (a) Compared to the algorithm of Luby, how does the algorithm obiviate the need for FIFO ordering of message delivery?

   (b) Does the algorithm ensure fairness? Justify your answer. Fairness in this context refers to the idea that access to mutual exclusion is in the order of time.

   (c) Does the algorithm ever get into a deadlock? Justify your answer.

   (Points: 4+3+3=10) (COs: 4)

3. Recall the phase-king algorithm for achieving distributed agreement. Answer the following questions.

   (a) Mention the properties requried of the solution to the agreement problem?

   (b) Does the algorithm satisfy the validity constraint of the soultion if the output is the tiebreak value proposed by the phase king? Justify your answer.

   (c) Does the algorithm have to really run for all $f+1$ rounds, or is it possible to stop the algorithm early? Justify your answer.

   (d) Show an instance where the algorithm fails to work correctly if the number of faulty processors, f, is such that $3f < n < 4f$, where n is the total number of processors.

   (Points: 1+2+3+4=10) (COs: 4)

4. Mention and explain in brief some fundamental design consdierations for distributed database systems? Map DynamoDB and BigTable according to the above considerations. Explain with the use of suitable examples, how does the support or the lack of support for transactions influence distributed database system design? **(Points: 4+1.5+1.5+3=10) (COs: 3)**

5. Recall the distributed algorithm to obtain the BFS traversal of a graph. Answer the following questions.

   (a) What is the number of rounds needed by the algorithm? Justify briefly.

   (b) Show examples of graphs where the algorithm takes exactly the above number of rounds. Your examples should be general enough in the sense that for every positive integer n, there exists a graph that meets the above bound on the number of rounds.

   (c) Extend the algorithm so that each node knows its set of children. What is the message complexity for this algorithm? Justify briefly.

   (d) Extend the algorithm so that the source node can terminate after all other nodes have done so.

   (Points: 1+3+3+3=10) (COs: 5)

6. Consider the Chord overlay network with function r_b and r_i mapping buckets and items to $[0, 1)$ respectively. We assume that r_b and r_i are drawn uniformly at random from a family of consistent hash functions. Answer the following questions.

(a) Describe the mechanism to place an item with id x in the appropriate bucket with a small example.

(b) Using the above mechanism, what is the expected number of items that are mapped to a given bucket when m items are placed in n buckets.

(c) How does the Chord mechanism ensure that the monotonicity property of the hashing is maintained? Explain with a small example.

(d) How many items are moved on expectation when a bucket leaves the system. Justify briefly.

(Points: 2+2.5+3+2.5=10) (COs: 3)