Distributed Systems

Monsoon 2024

Mid Semester Examination

International Institute of Information Technology, Hyderabad

September 24, 2024. 8:30 AM to 10:00 AM

Instructions: Read the following points carefully.

1. The question paper has 10 questions spread across two sections for a total of 45 points.
2. The question paper has **THREE** pages. Ensure that your question paper is properly printed.
3. No clarifications shall be provided during the exam. If you need to make any assumptions to answer any question, state those assumptions explicitly. Answers arising out of incorrect or unsuitable assumptions may not get any points.
4. Answer all parts of each question contiguously.
5. Excessively verbose answers may attract negative points.

Section A -- Short Answer Questions. This section has five questions of 2 marks each. Answer each question in no more than two to three paragraphs.

Akhil 1. Define the problem of distributed agreement and explain the three conditions needed of the solution. give marks for consensus too (0.5) **(CO-4)**

lokesh 2. Define logical time. Is the following true? For any two events e_1 and e_2, if $e_1 \rightarrow e_2$, then $t(e_1) < t(e_2)$. Justify your answer. Is the converse true with scalar time? Is the converse true with vector time? Justify your answer. **(CO-1)**

Just saying strongly consistent −1/4

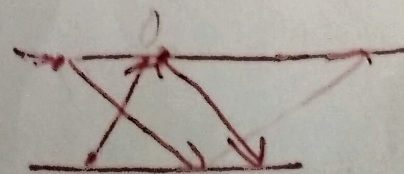
kishore sir 3. Explain the purpose of the Combiner and Partitioner in the context of Map-Reduce. **(CO-2)**

4. Recall the NFS protocol. Is the protocol stateful or stateless? What are some consequences of this characterization. How does it affect the system operations? Illustrate with suitable examples. **(CO-3)**

5. Write a small note on inherent complexity and accidental complexity. What aspects of these do MPI and gRPC suffer from or support. **(CO-2)**

↑ kishore

(Points: 5x2 = 10)



26/11/2024 08:04

Section B. Long Answer Questions. This section has 5 questions. Answer each question in enough detail. Each question is for 7 points.

1. Recall the algorithm discussed in class for arriving at distributed consensus in the presence of crash/fail-stop failures. Suppose that the algorithm tolerates f failures. Answer the following questions.

   a) What is the total number of messages exchanged during the algorithm? $O(n^2 f)$ or $n^2(f+1)$
   b) Under what conditions does the algorithm need all the f+1 rounds to arrive at the consensus? Explain. ~~The node leading consensus fails each round for f rounds~~
   c) Is it possible to stop the algorithm prior to f+1 rounds? How do you characterize such a stopping condition? Justify your answer. Yes. (1)

exactly 1 node ~~at least~~ fails per round for f rounds

(Points: 2+2+3=7) 1) Identify the 1st non faulty round **(CO-4)**

2) Count the no. of faulty nodes → $n-1-K$ total failures until now.

2. What is the metadata that a GFS master stores with respect to each file? If the master in GFS has 128 MB space for storing this metadata, for how many files can the master keep information of? Discuss some advantages and disadvantages of keeping the metadata in RAM vs in disk. Make and state any assumptions you need to arrive at this answer.

access speed
Stable storage / volatile storage
space limit / ∞
cost

(Points: 2+2+3=7) **(CO-3)**

Lokesh

3. Recall the algorithm of Lamport for guaranteeing mutual exclusion in a distributed system. Answer the following questions.

   a) What is the need for the reply messages in the algorithm? Explain. (Process not in CS / liveliness)
   b) Suppose that a site S_i is executing the critical section. Is the request of S_i at the top of the request queue at all sites? Justify your answer suitably. (No)
   c) Where does the algorithm require that messages are delivered in the FIFO order. Explain your answer. with FIFO, ME might not happen (security)

Fairness also ...

(Points:2+2+3=7) **(CO-4)**

Kishore

4. Define the notion of a consistent global state using both symbols and words. What are the typical challenges that algorithms for capturing a consistent global state must contend with. Compare and contrast the various algorithms for capturing a consistent global state based on various system properties and assumptions. You should use metrics such as the kind of data structures to be used, the purpose of any special messages or tokens, the nature of the algorithm with respect to guarantees of message delivery, and the like.

(Points: 1.5+1.5+1.5+1.5+1=7) **(CO-1)**

1) a → $O(n^2 f)$ (1 mark), calculation (2 mark, f+1 round n^2 messages)

b → exactly 1 node failure each round. (1 mark)
→ better explain → (1 mark)

26/11/2024 08:04

Aditya

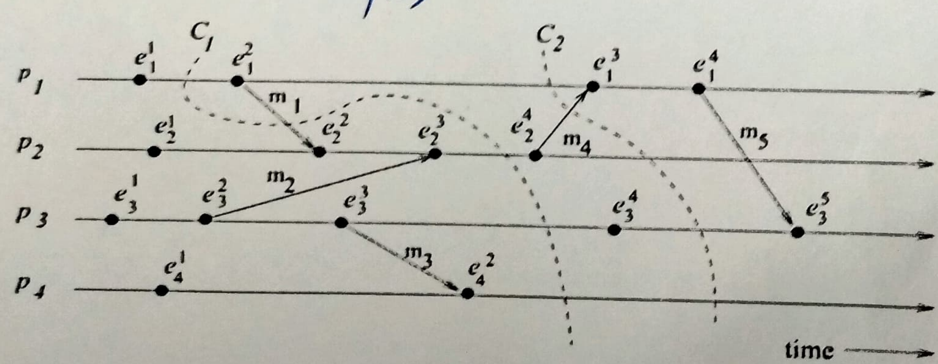
5. Define a cut in a distributed space-time diagram as a line joining an arbitrary point on the timeline of each process. This line therefore slices the space-time diagram into past and future with past referring to events that are to the left of the cut and future referring to events that are to the right of the cut. See Figure below for an example. Answer the following questions. It is impossible for a cut to violate the law of conservation of messages.

a) Is the following statement true? A consistent global state corresponds to a cut in which every message received in the past of the cut was also sent in the past of that cut. Justify your answer. Yes, every effect must have a cause so, ⇐ is also true

b) Suppose that a cut puts the event corresponding to the send of a message m_{ij} from process P_i to process P_j in the past and the receive event of the corresponding message in the future. In a consistent global state, how should m_{ij} be recorded? SC_{ij}

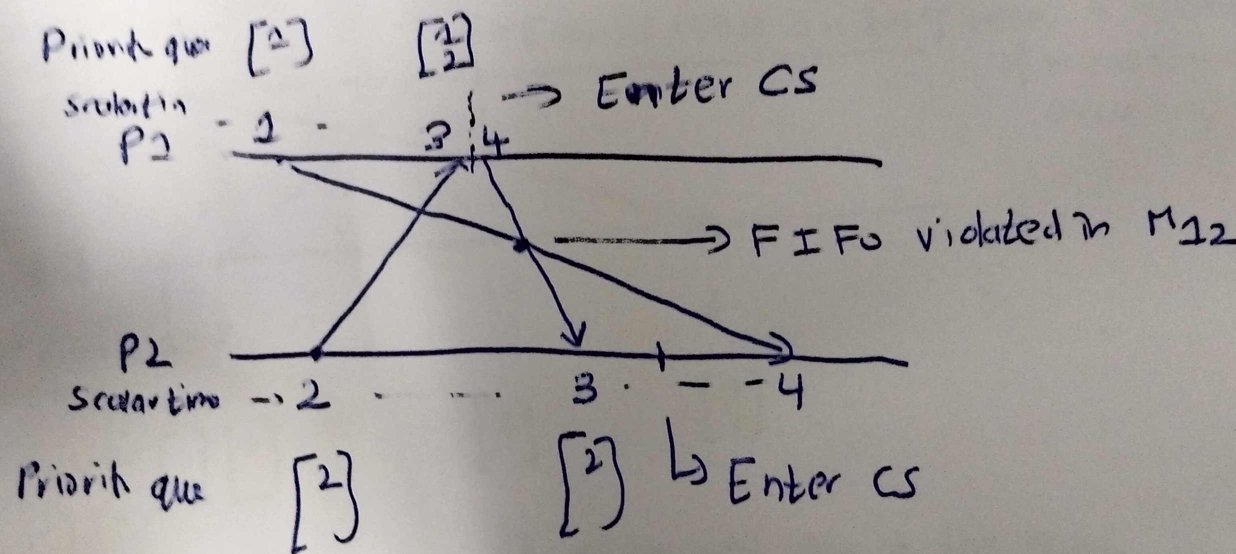
c) Consider a distributed system of n processes. Consider two cuts of a distributed computation C_1 and C_2 characterized by the following events. $C_1 = \{C_1(1), C_1(2), \cdots, C_1(n)\}$ and $C_2 = \{C_2(1), C_2(2), \cdots, C_2(n)\}$. The notation $C_i(j)$ refers to the point on the timeline of process P_j that the cut C_i slices. We call a cut C as a good cut if which every message received in the past of the cut was also sent in the past of that cut.

Is the cut LATER defined by events $\{\max\{C_1(1), C_2(1)\}, \max\{C_1(2), C_2(2)\}, \cdots, \max\{C_1(n), C_2(n)\}\}$ good? Is the cut LATER good if the cuts C_1 and C_2 are good? Justify your answer. No, Yes

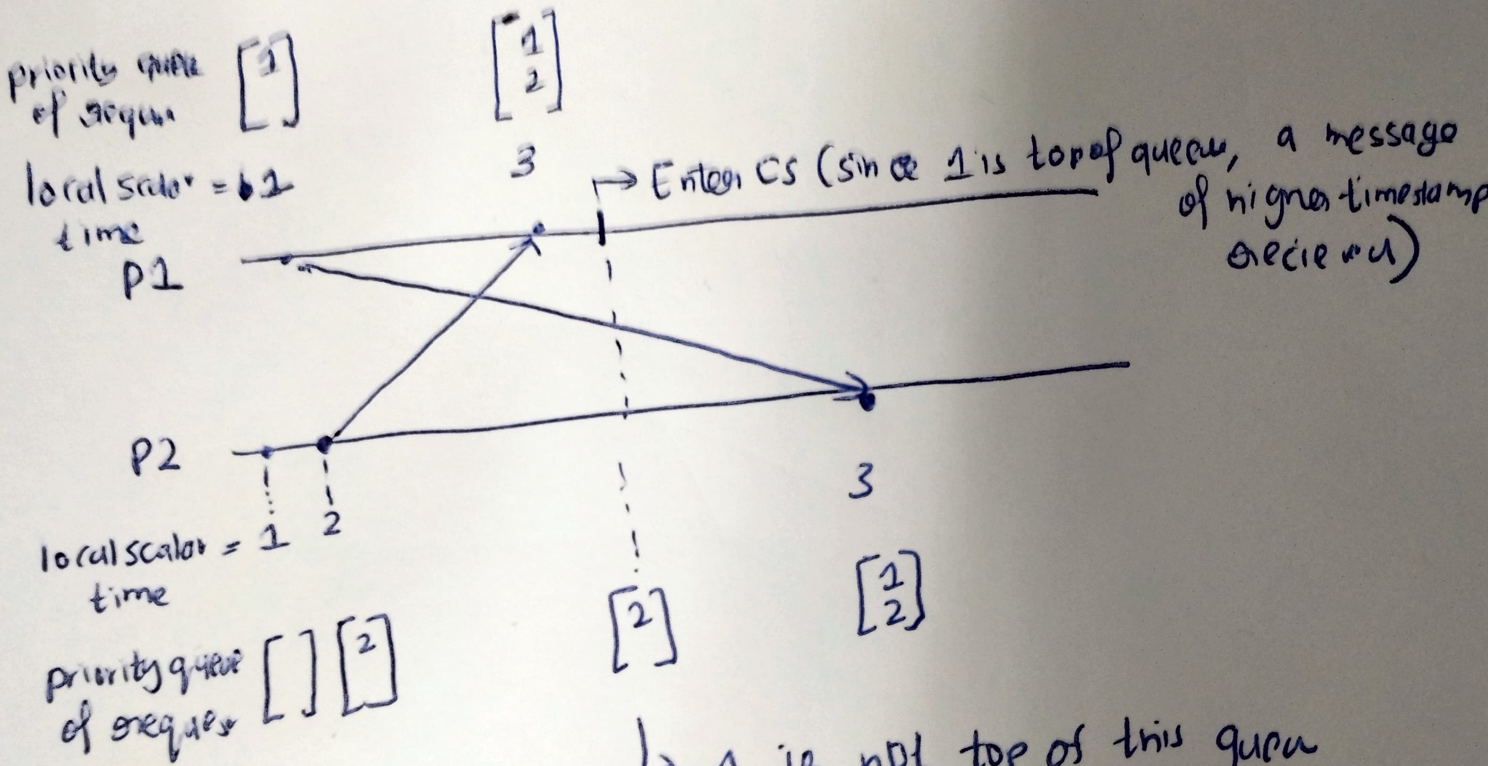


(Points: 2+2+3=7) (CO-1)

1



Both P1, P2 enter CS, as they are top of their queues and have received messages of higher timestamp => No FIFO, means no Mutual Exclusion (ME)



26/11/2024 08:09