Chalmers Techniska Högskola — CTH TDA297
Data- och Informationsteknik — GU DIT290

Distribuerade system fk

Tentamen 2021-03-17

Dag, *Tid, Sal*: March 17th 2021, 08:30-12:30, Home sweet home

***Kursansvarig*:** Philippas Tsigas (Tel: 772 5409)

Totalt *Poängtal*: 60

***Betygsgränser*:**

CTH: 3:a 30 p, 4:a 38 p, 5:a 48 p

GU: Godkänd 30p, Väl godkänd 48 p

Instructions

- Please answer in English, if possible.
  If you have very big difficulty with that, though, you may answer in Swedish.
- **Do not forget to write your personal number and if you are a GU or CTH student and at which "linje".**
- Please start answering each assignment on a new page; number the pages and use only one side of each sheet of paper.
- Please write in a tidy manner and explain (Clearly) your answers.

LYCKA TILL !!!!

1. (15 points) Consider a synchronous network, in every round every process may send and receive a message to/from each of its neighbors. The network is organized as a 2-dimensional grid. The width of the grid is w, the height is h. Width and height are defined in terms of edges. All processes start at the same time. At the beginning we assume no process or link failures.

   A Assume that all processes know the grid dimensions. Design a protocol that reaches consensus.

   B Can you modify the protocol to tolerate process crashes, same assumptions, no partitions.

   C Assume that none of the processes knows the grid dimensions, no process failures. Design a protocol that reaches consensus. How many rounds does your algorithm require?

   D Assume a system with crash failures; i.e, nodes can crash at any time during the execution, no partitions. Does your algorithm from C work in these system settings? If your answer is yes calculate the number of rounds it takes. If your answer is no provide an execution that does not.

2. (5 points) For solving the dinning philosophers problem, in a system with $n + 1$ philosophers, the following solution has been proposed: All philosophers except from philosopher P_0 seek their right fork first. Philosopher P_0 seek her left fork first.

   Does this solution solve the dinning philosophers problem? If your answer is yes provide a proof and an analysis of the time complexity of the algorithm.

   If your answer is no provide a counterexample.

3. (15 points) Conflict-free Replicated Data Types, eventual consistency and strong eventual consistency:

   1. Define eventual consistency and strong eventual consistency in the context of Replicated Abstract Data Types.
   2. What was the motivation that lead to the introduction of Conflict-free Replicated Data Types?
   3. What is the consistency that they guarantee.
   4. Describe, by providing pseudo-code and an informal description, a CRDT implementation of a set. Provide also a proof of its correctness.
   5. Is your implementation sequential consistent? Provide a proof or give a counter example.

4. (10 points) Given the following code segments, how many and which results are not possible under i) sequential consistency and ii) linearizability? Each row represents a time interval, two events on the same row are concurrent, first row happens first. List all results of the three prints e.g., print x = 0, print y = 0, print z = 0, etc, that are not possible under i) sequential consistency and ii) linearizability. Assume that all variables, i.e., A, x, y and z are initialized to 0 before this code is reached. Show your thinking and do not forget to define sequential consistency and linerizability.

P1	P2
A=1	
x=A	print y
y=A	print x
z=y	print z

P1	P2	P3
A=1		
x=A	y=x	print y
	z=x	print x
		print z

5. (10 points) Moa is building a database that is replicated on N machines, using eager replication. To access the database, a client accesses any of the replicas. The communication between clients and the replicas and between the replicas themselves is reliable, point-to-point, and FIFO-ordered. However, the communication delays can vary significantly. She wants to design a totally-ordered causal multicast algorithm to be used on her system. She learned in the course the asynchronous and synchronous doorway mechanisms. She wants to make use of these mechanisms as key components in her design. Can you help her by designing such a protocol i.e. that implements totally-ordered causal multicast (i.e. satisfies the following requirements: 1. Reliability: Integrity, Validity, Agreement 2. Ordering: Causal, Total-order). Please prove the properties of your protocol.

6. (5 points) Consider a synchronous system with reliable point to point communication. Is it possible to solve the Byzantine Agreement problem and tolerate any number of byzantine faults using authentication? If yes describe a protocol and its proof of correctness. If not provide an impossibility proof.