Chalmers Techniska Högskola	CTH TDA297
Data- och Informationsteknik	GU DIT290

Distribuerade system fk

Tentamen 2025-06-11

Dag, *Tid, Sal*: June 11th 2025, 08:30-12:30, HA4

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

Totalt *Poängtal*: 65

***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. (10 points) (Bitcoin)

   Bitcoin is a decentralized system that relies on replication to maintain consistency and fault tolerance across a distributed network of nodes.

   1. Describe the replication scheme used in Bitcoin. In your answer, explain:
      - How data (e.g., transactions and blocks) is replicated across nodes.
      - The role of mining and consensus in maintaining replicated state.
      - How consistency is eventually achieved despite network asynchrony and potential forks.
   2. Compare this replication approach with traditional primary-backup replication in distributed systems. Highlight at least two key differences in terms of fault tolerance, consistency, and coordination.

2. (15 points) (Quorum Systems)

   Quorum-based replication is a fundamental technique in distributed systems for ensuring consistency and availability.

   - Explain the concept of quorum consensus. In your answer, define what read and write quorums are, and describe the conditions they must satisfy to ensure consistency.
   - Consider a distributed key-value store replicated across 7 nodes. The system uses quorum consensus for reads and writes. What are the minimum sizes of read and write quorums that ensure consistency? Justify your answer.
   - Discuss how quorum consensus handles node failures and network partitions. What trade-offs does it make between consistency, availability, and partition tolerance?

   Moa is constructing a Quorum System with 9 nodes named: $11, 12, 13, 21, 22, 23, 31, 32, 33$. Moa defines a read-quorum to be any set of 3 nodes that have the same first character on their name ($11, 12, 13$ is an example of a possible read-quorum). Moa defines a write-quorum to be any set of 3 nodes that have the same second character on their name ($11, 21, 31$ is an example of a possible write-quorum).

   1. Is Moa's design correct? Explain why? What puzzles Moa is the fact that the cardinality of both read and write quorums is 3, while she is using 9 nodes.
   2. Calculate its resilience f. Give an example where this quorum system does not work anymore with $f + 1$ faulty nodes.
   3. What is the consistency that a quorum consensus replication guarantee?

3. (10 points) (2PC)

   - Describe the Two-Phase Commit protocol in detail. Your answer should include the roles of the coordinator and participants, the sequence of message exchanges, and the possible states each process can be in during the protocol.

- Prove that the 2PC protocol guarantees consistency, i.e., no two participants decide differently (one commits while another aborts), even in the presence of failures. Clearly state any assumptions you make (e.g., about failure models, reliable communication, etc.).
- Briefly discuss one limitation of 2PC in terms of availability or fault tolerance, and how this limitation is addressed in more advanced protocols like Three-Phase Commit (3PC).

4. (10 points) (Consistency Models)

   Consistency models define the guarantees provided by a distributed system regarding the visibility and ordering of updates.

   - Define linearizability and sequential consistency. Use examples to illustrate each model.
   - Compare the two models in terms of: real-time constraints, client-perceived operation order, implementation complexity.
   - Provide a scenario where a system satisfies sequential consistency but not linearizability. Explain why the scenario meets one model but not the other.

5. (10 points) (Distributed Algorithms)

   The Dining Philosophers problem is a classic synchronization problem in distributed systems, often used to illustrate issues of deadlock, fairness, and symmetry.

   - Define what is meant by a symmetric solution in the context of the Dining Philosophers problem.
   - Is it possible to design a symmetric and deterministic solution to the Dining Philosophers problem that avoids deadlock and ensures progress? Justify your answer with a formal argument or counterexample.
   - If symmetry must be broken, explain how it can be done and what trade-offs are introduced in terms of fairness or complexity.

6. (10 points) (Multiple Choice Questions)
   Each statement is either true or false. A correct answer gives 1 point, a wrong answer gives -1 point, no answer gives 0 points. Overall you cannot get less than 0 points for this question.

   1. A Reliable Broadcast satisfies also the properties a Best-effort broadcast.
      A. True B. False
   2. There is no symmetric solution to the leader election problem.
      A. True B. False
   3. The Gossip architecture for replication was designed to provide highly available service.
      A. True B. False
   4. The 3 Phase Commit protocol was introduced to improve the latency of the 2 Phase Commit protocol in executions where no faults take place.
      A. True B. False
   5. In an asynchronous system, where messages can be lost undetectably, consensus is solvable.
      A. True B. False

6. Quorum consensus replication is linerizable.
   A. True B. False
7. Bitcoin is linearizable.
   A. True B. False
8. Bitcoin is strongly eventual consistent.
   A. True B. False
9. A system can be sequential consistent but not linerizable.
   A. True B. False
10. Primary-backup replication is linerizable.
   A. True B. False