Chalmers Techniska Högskola Data- och Informationsteknik	CTH TDA297 GU DIT290

Distribuerade system fk

Tentamen 2017-03-15

Dag, *Tid, Sal*: March 15th 2017, 08:30-12:30, H building

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

***Hjälpmedel*:** Inga

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 (briefly) your answers.
- Students must **not** write their personal number on the answer sheets since the exam is anonymous; they shall write that **only** on the name slip area that they will seal.

LYCKA TILL !!!!

1. (3 points) How do the replica managers in a passive replication scheme reach consensus on the effect of a request?

2. (4 points) Is Total Order Broadcast (TOB) equivalent to Consensus? If no, explain why not. If yes, explain (with pseudo code/high level description) how the two are equivalent, and how can one be implemented using the other.

3. (3 points) Suppose you have an asynchronous message-passing system with a complete communication graph, unique node identities, and no failures. Show that any deterministic shared-memory object (an ADT that can be accessed concurrently be parallel processes) can be simulated in this model, or give an example of a shared-memory object that can't be simulated.

4. (13 points) Moa is building a counter that is replicated on N machines. To access the counter, a client can accesses all 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. Moa has designed the following algorithm to be used on her shared counter system:

   > Each replica maintains a local counter. To execute an increment operation, a client needs only one machine available, it just increases the value of the local counters of this replica by 1. To read the counter, a client needs access to all N machines, it asks from all the machines the values of their respective local counters and sums up their values.

   Is this counter linearizable? Provide a proof sketch if it is or a counterexample otherwise.

   Suppose we use the same mechanism to build a generalized counter, where an increment operation may increase or decrease the value of the counter by an arbitrary amount, not just +1. Is the resulting object linearizable? Provide a proof sketch if it is or a counterexample otherwise.

5. (15 points) The algorithm by Choy and Singh for resource allocation that we discussed in the class, starts with the idea of using the Δ-coloring conflict solution for keeping low the maximum access time. As described in Chandy and Mistra's solution a Δ-coloring will result in an acyclic precedence graph. In contrast to teh Chandy and Mistra algorithm, in the Choy and Singh algorithm the precedences will remain static and hence keep the longest chain of processes waiting for resources to be proportional to Δ. The unfairness problem of the static precedence solution is avoided by a mechanism called double doorway.

   - Provide and describe the double doorway mechanism.
   - Provide a complexity analysis of the complete Choy and Singh algorithm.

6. (12 points) Byzantine Agreement:

   1. Prove that it is impossible to reach agreement in a system with three processes if one of them is Byzantine faulty.
   2. How can the above proof be generalised for a system with n processes?
   3. Is it possible to reach agreement in a system with three processes if one of them is Byzantine faulty by using authentication (unforgeable signatures)? If your answer is yes, describe an algorithm. If your answer is no give a proof to support your answer.

7. (10 points) 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. Assuming only crash failures, 2PC is a good choice to handle coordinated progress.
      A. True B. False
   2. 3PC was introduced to handle undetected message losses.
      A. True B. False
   3. Any transactional service is linearizable if each request is executed at most once.
      A. True B. False
   4. A system can be linearizable but not sequential consistent.
      A. True B. False
   5. Strong eventual consistency guarantees sequential consistency.
      A. True B. False
   6. Sequential consistency is composable.
      A. True B. False
   7. The Gossip architecture for replication was designed to provide highly available services.
      A. True B. False
   8. Strong eventual consistency is guaranteed by CRDTs.
      A. True B. False
   9. There exist a consensus solution able to handle up to $f < n/3$ Byzantine failures.
      A. True B. False
   10. CRDTs optimistically execute an operation and check later for inconsistencies.
       A. True B. False