Chalmers Techniska Högskola
Data- och Informationsteknik

CTH TDA297
GU DIT290

Distribuerade system fk

Tentamen 2014-08-22

Dag, Tid, Sal: August 22th 2014, 14:00-18:00, V 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. *15 marks*

   (a) Define the specification of the i) Reliable Broadcast, ii) FIFO Broadcast and iii) Causal Broadcast.

   (b) Describe three algorithms that implement respectively the three different types of Broadcast mentioned above in an asynchronous system, with process crash failures.

   (c) What is the time and communication complexity of your FIFO Broadcast Algorithm? Provide a complexity proof.

2. *10 marks*

   Describe an algorithm that computes a spanning tree of a network $G(V, E)$. How a node of the network can use the existence of such a spanning tree in order to broadcast information to all nodes of the network?

3. *10 marks*

   Describe the differences between the three-phase commit protocol and the two-phase commit protocol. Draw the three-phase commit protocol as a state machine where you also describe the behavior of the protocol when time-outs are triggered and processes are recovered after crashing.

4. *10 marks*

   Describe the two generals problem. Can you find a solution to the problem? If yes describe your solution and provide a proof of its time and communication complexity. If not provide a proof that the problem is not solvable.

5. *15 marks*

   a) Eric wants to build a replicated storage system. In his system there is only one client. The client performs just one storage operation (read or write) at a time, waiting for each operation to complete before starting the next. Eric wants availability even in the case of one server failure. Because of that he decides to store the data on two servers. Eric wants to ensure that the data on the two servers stay identical all the time. In order to guarantee that he is thinking making the writes at the two servers atomic using three-phase commit, to ensure both-or-nothing behavior. In the design he has in mind, the client acts as the transaction coordinator. The client would execute a write as described in the three-phase commit protocol. The system uses the timeout recovery scheme explained in the lectures. Eric thinks about this design for a while, and eventually realizes that three-phase commit is fundamentally not suited to providing ***availability*** via replication. Please explain why. Eric wants to formalize the consistency properties of the replication system. Is it sequential consistent? Is it a linearizable one? Please explain your answer.

   b) Eric decides to change his design and uses now the gossip architecture that lazily synchronizes the two servers. The single client contacts any server that is available and gets the value that this server has, updates are also performed on the first availiable server first and then lazily propagated on the next server. Each client request uses a unique id. What is the availability that he can achieve? Eric wants to formalize the consistency properties of the replication system in case where there are no failures. Is it sequential consistent? Is it a linearizable one? Please explain your answer.

c) Eric decides to change his design and uses now nine replicas and quorums to ensure strong consistency and availability. What are the constraints on the sizes of read and write quorums? What is the availability that he can achieve? Give an example of quorum processing involving a write, two reads, then a write. Eric wants to formalize the consistency properties of the replication system. Is it sequential consistent? Is it a linearizable one? Please explain why?