

Distribuerade system fk

Tentamen 2024-06-05

Dag, Tid, Sal: June 5th 2024, 08:30-12:30, SB

Kursansvarig: Philippos 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. (10 points) (Atomic Multicast)

Moa is building a database that is replicated on N machines. 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. Moa has designed the following variant of a totally-ordered multicast algorithm to be used on her system:

Each replica maintains Lamport's logical clock, and every inter-replica message is stamped with the unique id of the sender upon transmission. Whenever a replica receives a database update message from a client, it broadcasts the update in a message to all replicas (including itself). Whenever a replica receives an update message from another replica (or from itself) it puts the message into a local queue, and acknowledges the reception of the update by sending an acknowledgment message to all other replicas (including itself). The replica applies an update from its local queue to its local database if and only if:

- i) the update message has the lowest timestamp among the messages in the replica's local queue, and
- ii) the update message has been acknowledged by a quorum of N

After a replica has applied the update to the local database, the update message and its acknowledgments are removed from the local queue. If later another acknowledgment arrives for the message, such a late acknowledgment is simply dropped from the queue.

Prove that the above algorithm implements totally-ordered causal multicast (i.e. satisfies the following requirements: 1. Reliability: Integrity, Validity, Agreement 2. Ordering: Causal, Total-order) or produce a counter example.

2. (10 points) (Agreement Synchronous)

Suppose that we modify the problem of synchronous agreement with crash failures so that instead of crashing a process forever, the adversary may jam some or all of its outgoing messages for a single round. The adversary has limited batteries on its jamming equipment, and can only cause f such one-round faults. There is no restriction on when these one-round jamming faults occur: the adversary might jam f processes for one round, one process for f rounds, or anything in between, so long as the sum over all rounds of the number of processes jammed in each round is at most f . For the purposes of agreement and validity, assume that a process is non-faulty if it is never jammed. Can we modify the round based algorithm that we discussed in the course to work in this model? If yes, as a function of f and n , how many rounds does it take to reach agreement in the worst case in this model, under the usual assumptions that processes are deterministic and the algorithm must satisfy agreement, termination, and validity? If no, provide a proof.

3. (10 points) (Linearizability)

Consider a system with two processes, P_1 and P_2 , and two data objects, X and Y . The objects support only read and write operations, where $read(X) \rightarrow v$ denotes that a read operation on object X returns value v and $write(X, v)$, denotes a write operation of value v on object X .

1. Construct an execution scenario that is sequentially consistent, but not linearizable.

2. Explain why your scenario does not satisfy linearizability but does satisfy sequential consistency.

Your execution scenario should use only read and write operations running in these two processes accessing these two data objects and follow the notation described above.

4. (10 points) (Resource Allocation)

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 Misra's solution a Δ -coloring will result in an acyclic precedence graph. In contrast to the Chandy and Misra 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.
- Now consider the algorithm that is using only the asynchronous doorway together with the Δ -coloring conflict resolution scheme to solve the general resource allocation problem, i.e. the synchronous doorway is not used.
 1. Does this algorithm solve the general resource allocation problem? Provide a proof of the properties or a counterexample.
 2. If the algorithm is correct, what is the time complexity of this algorithm in an asynchronous system?

5. (10 points) (Byzantine Consensus)

Which type of faults are called Byzantine faults? Prove that it is impossible to reach binary consensus in a system with three processes if one of them is Byzantine faulty. How can the above proof be generalised for a system with n processes?

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 linearizable.
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 linearizable.
A. True B. False
10. Primary-backup replication is linearizable.
A. True B. False