Tolerância a Faltas Distribuída

2015/16

MEI/MI/MSI

Problem Sets

1. Explain what are the similarities and differences of the following two means to achieve dependability: fault prevention and fault tolerance.
2. Why does the class of timing faults include the class of omission faults?
3. If two components, A and B, fail independently and each one has a reliability of 0.9, what is the reliability of a system composed of A and B that requires both components in order to function correctly?
4. Consider a hard disk as a system. What of the following is a failure, a fault, an error?
    a. A sudden, abrupt movement causes a head crash
    b. Data stored on the disk is destroyed and forever lost due to the head crash
    c. A read operation that pretends retrieving previously stored data fails.
5. A system has on average one failure every 4000 hours and the repair takes 4 hours. Calculate:
    a. its reliability in terms of Mean Time To Failure (MTTF)
    b. its reliability in terms of Mean Time Between Failures (MTBF)
    c. its maintainability in terms of Mean Time To Repair (MTTR)
    d. its availability based on the MTBF and the MTTR
6. Provide two examples of each of the following 3 forms of redundancy: space redundancy, time redundancy and value redundancy.
7. Why with 2-MR (2-modular redundancy) we can only achieve fail-silent behavior, in the sense that the resulting system cannot tolerate a faulty component?
8. What do we mean when we say that there is no single correct model for an object?
9. What do we mean when we say that answering different types of questions about an object usually requires different models?
10. What do we mean when we say that choosing a model implies making assumptions about the environment in which the system will run?
11. What does it mean to say that a model is accurate? And tractable?
12. Why do we say that postulating that a system is asynchronous is a non-assumption?

13. What does it mean to say that a process fails by crashing?
14. Why is important to study distributed algorithms rigorously, defining a system model, stating the problem in terms of a set of properties, writing the algorithm in unambiguous pseudo-code, and providing proofs of correctness?
15. Explain informally what the reliable broadcast problem is.
16. Consider the following system model: point-to-point network, reliable channels, no time assumptions (asynchronous system), and crash faults. Why simply sending a message to all processes using these (reliable) channels does not solve reliable broadcast in this model?
17. Consider the same system model. Explain informally an algorithm that solves the problem of reliable broadcast in this system model.
18. Consider the same system model and the reliable broadcast algorithm we studied. Write a short but rigorous proof that the algorithm satisfies the Validity property.
19. Explain the problem of FIFO broadcast.
20. Given a reliable broadcast algorithm, how can we enforce FIFO order?
21. What do we mean when we say that an event E causally precedes an event F?
22. Explain the problem of causal broadcast.
23. Explain the problem of total order (or atomic) broadcast.
24. For each of the following properties, say which one is a Safety property, a Liveness property of both a Safety and Liveness property. Justify.
    P1: All correct processes deliver the same messages.
    P2: If a correct sender broadcasts a message m then all correct processes eventually deliver m.
    P3: There is a known constant T such that if a message m is broadcast at real time t, then no correct process delivers m after real time t+T.
25. What is a perfect failure detector? How does it differ from an eventually perfect failure detector?
26. Why a perfect failure detector in not implementable in a partially synchronous model defined this way: For every run S there is a Global Stabilization Time (GST) after which there are bounds on the relative process speeds and communication delays. Both GST and the bounds are unknown.
27. Consider the implementation of a failure detector in slide 18 of lecture note 05 (“Failure Detectors and Consensus”). Explain why does it implement an eventually strong failure detector ($\diamond\mathcal{S}$) in the partially synchronous system model considered (no omissions in the network).
28. What is the weakest class of failure detectors that allows solving consensus? Why is it the same to consider this class of failure detectors or

an eventually strong failure detector in a system model that otherwise is asynchronous and has only crash faults?

29. The consensus algorithm by Mostefaoui and Raynal in slide 20 of lecture note 05 ("Failure Detectors and Consensus") solves consensus if f<=n-1 in the following system model: point-to-point network, reliable channels, asynchronous, crash faults and a strong failure detector (S). Prove that this is true considering the specification of consensus in terms of the 3 properties in slide 19.
30. Explain the basic functioning of active replication (or state machine approach). Say explicitly why replicas need to start in the same state, replicas must be deterministic and atomic broadcast is needed.
31. Discuss which is the minimum number of replicas to implement a replicated service using active replication with crash faults.
32. Explain the basic functioning of passive replication (or primary-approach). Say explicitly why replicas do not need to be deterministic and preemption is supported.
33. Answer the same but for semi-active replication.
34. What is the main benefit of Paxos as an algorithm to implement active replication?
35. Why the following design principle of Paxos is important: algorithm is safe even if system is asynchronous; partial synchrony needed only for liveness.
36. Consider the following situation: a set of servers is running Paxos to order clients' requests and momentarily two servers S1 and S2 believe they are the leader and assign the same sequence number 123 to two requests, r1 and r2, respectively. Explain why Paxos does not let two servers accept these two requests (i.e., either r1 or r2 are assigned that sequence number). Consider a single view (i.e., no view change).
    a. What is the role of prepare and prepare-ok messages in Paxos?
    b. In what conditions the liveness of Paxos is ensured?
    c. What is aggregation (or batching) in an algorithm like Paxos, and why is it an important mechanism?
    d. What are the main differences between Viewstamped Replication and Paxos?
37. Consider the Paxos and Viewstamped Replication protocols used to implement replicated services.
38. How many replicas are needed to ensure that the replicated service is able to tolerate 3 faults (f=3) using these protocols? Justify your answer.
39. Why the following design principles of Paxos are important: algorithm is safe even if system is asynchronous; partial synchrony needed only for liveness.

40. Explain carefully how Viewstamped Replication ensures that client requests are executed in total order by all server replicas.
41. Explain why the durability mechanisms applied to state machine replication (logging, checkpoints and state transfer) can affect the performance of the system.
42. What is the difference between RAID4 and RAID5? Why the latter is considered more efficient? (You can draw a picture if you think it would help explain.)
43. Explain the write and read protocols for implementing a regular register using asynchronous majority quorum systems.
44. Considering concurrent objects, answer the following questions.

a. What does it mean to say that a concurrent object is linearizable and wait-free?
b. Is there a wait-free implementation of a *compare&swap* object by a *test&set* object? Justify.

45. What is the consensus number of DepSpace? Justify.
46. Explain the reasons why a coordination service is considered much easy to use and maintain than implementing distributed (message passing) algorithms.