Problem Set 3

CSE 452 / CSE M552

May 28, 2017

Submit short, **typeset** answers to the following questions. Please work on this individually. You may not use skip days on this assignment.

Problem 1: Two-phase commit vs. Paxos

Imagine you have a birds-eye view of a system using two-phase commit. You can see all of the messages nodes send each other, but cannot examine any node's internal state. For a given transaction, at what point is it *durable*, that is, guaranteed to be committed? Give the earliest such point in time (hint: "every participant has received a COMMIT message" is too late).

Now, do the same thing for Paxos. When is a *value* (note: a value, not a proposal) durable?

Problem 2: Paxos acceptor state

Consider a deployment of Paxos (from Paxos Made Simple) with three acceptors. State whether each of these is a valid state at the three acceptors, where a state $n : (x, y)$ means the highest-numbered proposal accepted by acceptor n has number x and value y (and `nil` means the acceptor hasn't accepted any proposals). If the state is not valid, explain why in one sentence.

Hint: a state is valid if there is some sequence of message deliveries and message drops and node failures that leads to the state, assuming a correct implementation of proposers and acceptors.

- (a) $1 :$ `nil`, $2 :$ `nil`, $3 :$ `nil`
- (b) $1 : (1, A)$, $2 :$ `nil`, $3 :$ `nil`
- (c) $1 : (1, A)$, $2 : (2, B)$, $3 :$ `nil`
- (d) $1 : (1, A)$, $2 : (2, B)$, $3 : (3, C)$

Problem 3: A dubious Paxos execution

Consider another Paxos deployment with acceptors A, B, and C. A and B are also proposers, and there is a distinguished learner L. According to the Paxos paper, a value is chosen when a majority of acceptors accept it, and only a single value is chosen. How does Paxos ensure that the following sequence of events cannot happen? What actually happens, and which value is ultimately chosen?

- (a) A proposes sequence number 1, and gets responses from A, B, and C.
- (b) A sends `accept(1, "foo")` messages to A and C and gets responses from both. Because a majority accepted, A tells L that `"foo"` has been chosen. However, A crashes before sending an accept to B.
- (c) B proposes sequence number 2, and gets responses from B and C.
- (d) B sends `accept(2, "bar")` messages to B and C and gets responses from both, so B tells L that `"bar"` has been chosen.

Problem 4: Paxos liveness

In the absence of a designated proposer, it is possible for Paxos to fail to make progress even if no messages dropped and no nodes fail. Briefly describe how this can happen in a system with two proposers and three acceptors. Be specific about which messages are sent and in what order they are delivered.