HY590.45
Modern Topics in Scalable Storage Systems

Kostas Magoutis
magoutis@csd.uoc.gr
http://www.csd.uoc.gr/~magoutis
Order on state updates
Paxos algorithm

• Way to build fault-tolerant distributed systems
  – Replicated state machines (RSM)

• Consensus via message exchange
  – Asynchronous: no timing guarantees
  – Network can delay, reorder, lose (but not corrupt) packets

• Can guarantee safety
  – Replicas will agree on a single value

• Need additional assumptions to ensure progress
Informally

- Three roles: Proposer, acceptor, learner
- Simplest, but fault-intolerant solution: single acceptor
- With >1 acceptors, agreement by a majority required
- If single value proposed, that value should be chosen
  - Thus, an acceptor must accept the first value proposed to it
- However, this may lead to fragmented electorate
  - Multiple proposals by each proposer should be possible
  - Identify each proposal by a unique integer N
Informally

• After consensus, an acceptor cannot change its mind
  – A value is chosen when single proposal with that value
    accepted by a majority of the acceptors

• Allow multiple proposals to be chosen, but guarantee
  that all chosen proposals have the same value
Paxos setup

- Be able to agree in the presence of up to \( f \) failures
- \( 2f+1 \) nodes
- Agreement when majority \((f +1)\) agrees on a value
Need to try to get a majority to accept
Informally

- Allow multiple proposals to be chosen, but guarantee that all chosen proposals have the same value.
- If proposal $N$ with value $v$ is chosen, every higher numbered proposal issued by any proposer should have value $v$.
- A proposer wanting to issue a proposal numbered $N$ must learn the highest-numbered proposal $< N$ (if any) that has been or will be accepted by a majority.
Informally

• A proposer wanting to issue a proposal numbered $N$ must learn the highest-numbered proposal $< N$ (if any) that has been or will be accepted by a majority
  – Easy to learn about values already accepted
  – Hard to predict the future

• Control the future by extracting a promise that there will not be any acceptances of proposals $< N$
Paxos – phase 1

- Client initiates a prepare request for a value $v'$. 
- The prepare request is acknowledged by the proposer to an acceptor. 
- The acceptor then communicates this acknowledgment to the learner. 
- This process is repeated until a highest-numbered prepare request is accepted. 
- The accepted proposal is then written to a stable store.
Paxos – phase 2

propose $N, v$

client

propose $N, v$

proposer

acceptor

learner

value $v'$

client

propose $N, v$

proposer

acceptor

learner

propose $N, v$

proposer

acceptor

learner

propose $N, v$

proposer

acceptor

learner
Paxos – communicate agreement

decide $N, v$

decide $N, v$

decide $N, v$

value $v'$
Paxos – majority learns outcome