

# Paxos

CS262, Introduction to Distributed Computing  
Spring, 2008

# Administrivia I

- Four weeks to go...
- Programming Assignments
  - Six of you have not completed Assignment 1
  - Only four of you have completed Assignment 2
- Projects
  - Send your proposals by April 11

# Administrivia II

- Exams
  - About half graded
  - Will be returned Wednesday
  - Good, bad, ugly
- Remember, you can re-take

# Paxos

- Agreement problem
  - Want to make sure everyone agrees (eventually)
  - Make progress
  - Allow missing members to catch up
- An alternative to causal atomic broadcast

# Context

- Each participant has
  - Messengers for communication
    - Messages may be lost or delayed
  - Persistent storage (ledgers)
  - Local storage (notes)
- Goal is eventual agreement

# Synod

- Each ballot has a unique number
  - The numbering provides an ordering
- The quorums of any two ballots must have a common member

# Ballots

- For every ballot B
  - if any member in B's quorum voted on an earlier ballot, then the decree in B is the decree of the latest of those earlier ballots
- This means any successful vote will be not be lost (why?)

# Majority

- We think of a majority as more than half
  - This will work
  - It isn't needed
- The real purpose of a majority
  - Insure that there is overlap between any two majorities
  - Could be weighted in various ways

# Leaders

- The synod algorithm requires no leader
  - If multiple proposals are made, take the one with the highest sequence number
  - Does not allow multiple votes at the same time
- In the synod algorithm, sequence numbers tell you what vote you are on

# With a Leader

- Multiple votes can be taken
  - Leader determines which value for which sequence
  - Everyone keeps track of what votes have been taken
- Sequence numbers disambiguate values

# Leader Election

- Leaders can change
  - When a new leader is elected,
    - Find out what the last ballot was (from a quorum)
    - Start after that
  - If unknown values are needed, use olive day decree

# Catching up

- Ask the leader where we are
  - Can send all decrees
  - Can send latest, and you can ask for gaps
  - Can piggy-back information on other messages

# Various Optimizations

- Piggy-backing messages
- Authorities
- Time-stamping/leasing
- Timeouts

# Changing Legislators

- Legislators for decree  $n$  determined by  $n-3$ 
  - Why  $n-3$ ?

# Final Projects

- You should be thinking about them
- You can come and talk to me
- You need to write a one or two page description

# What I'm Looking For

- It has to involve multiple machines
- There should be some well-defined question
- What you implement should have some bearing on the question
- You will need to give a 10 minute presentation the last week of class