Midterm feedback (thanks)

- Points to need for more structure/signposting
  - Will try to start more classes with a few slides
- Maybe try some other communication tool [poll]
- Experiment: try to mitigate “race to answer” with breakout rooms (1–2 times per lecture)
  - Discuss among yourselves for 1 minute
  - One person reports back in main chat with member SUNet IDs
    E.g., [sunet1,sunet2,sunet3] Answer...
  - Jim will read out the best answer(s)
- A majority finds project discussions helpful
  - Clarification: fine to meet with me multiple times
  - Will have another round of meetings May 4–15

CS244b so far

- Seen how to transact atomically across systems using 2PC
  - E.g., lets you shard a database for scalability
- Seen how to replicate deterministic systems with consensus
  - Replication provides greater availability and reliability
  - Understand how at least one of (Raft, Paxos) works
- Seen an example replicated system: zookeeper
  - Nice, clean abstraction barrier between RSM and consensus (ZAB)
- Next week: Byzantine failure

Today’s learning goals

- Add a few more techniques to our arsenal
  - Primary copy replication, Witnesses
  - “Leases” (even though paper doesn’t use the term)
- More experience thinking about replication, consistency, logs
  - Reinforce concepts from multiple angles before doing Byzantine
- See a real system that is not perfectly clean, faces trade-offs
  - Violating abstraction barriers (RPC, NFS, replication)
  - Making hardware assumptions (clocks, batteries)
  - Failing to meet expected semantics (atime)

Who are the authors?

- Barbara Liskov
  - One of Stanford’s most distinguished CS Ph.D.s
  - Co-invented viewstamped replication (published before Paxos)
  - Co-invented practical Byzantine fault tolerant replication
  - Other contributions: parametric polymorphism, decentralized information flow control
  - ACM Turing award 2008 for inventing abstract data types
  - If Harp violates abstraction boundaries, probably a good reason!
- Sanjay Ghemawat
  - Highly respected engineer at Google
  - Numerous contributions including map-reduce, GFS, Spanner