UNIVERSITY of VIRGINIA

CS4740
CLOUD COMPUTING

Final Review

Prof. Chang Lou, UVA CS, Spring 2024

FINAL

- **Mon 5/6 2:00pm-3:30pm, Olsson Hall 011**
- Based upon lecture materials and projects (lab 2)
- No laptops, cellphones or other electronic devices.
- You are allowed to bring one US letter or similar size double-sided note.

- **Do not cheat**

COVERAGE

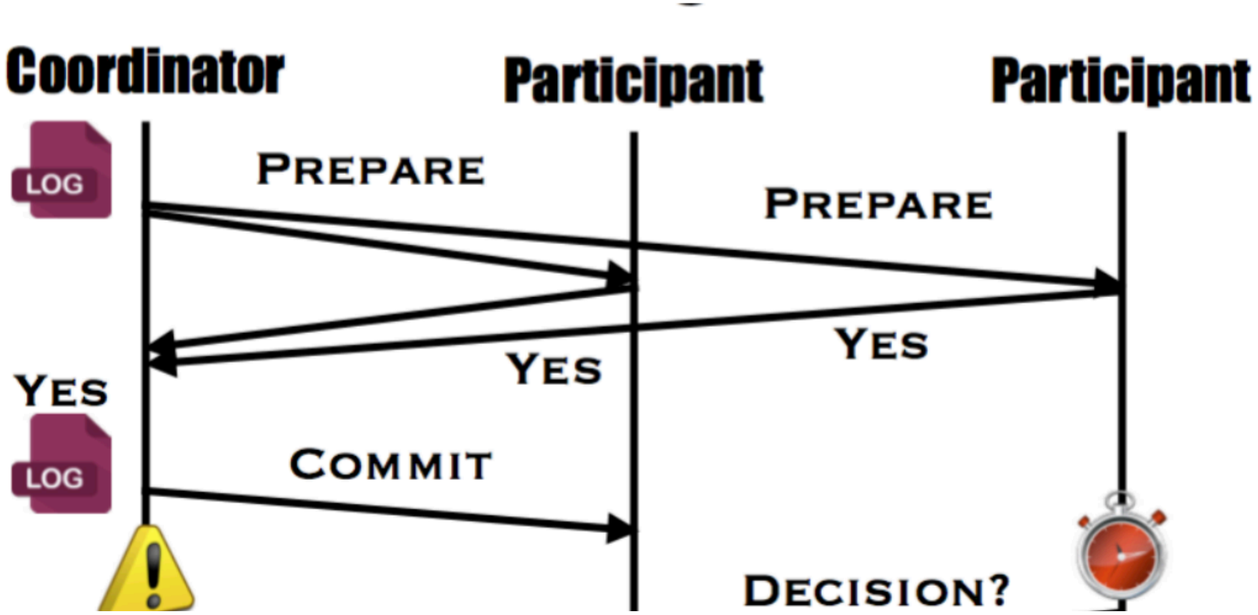
- Topics:
  - main focus: 2PC, consensus (Raft), isolation/consistency, GFS, ZooKeeper
  - only in true/false: virtualization, ML systems, reliability
  - not included: cloud infrastructure in industry

- How to use today's slides:
  - Use it as a basis to develop your cheatsheet
  - Use it to self-test

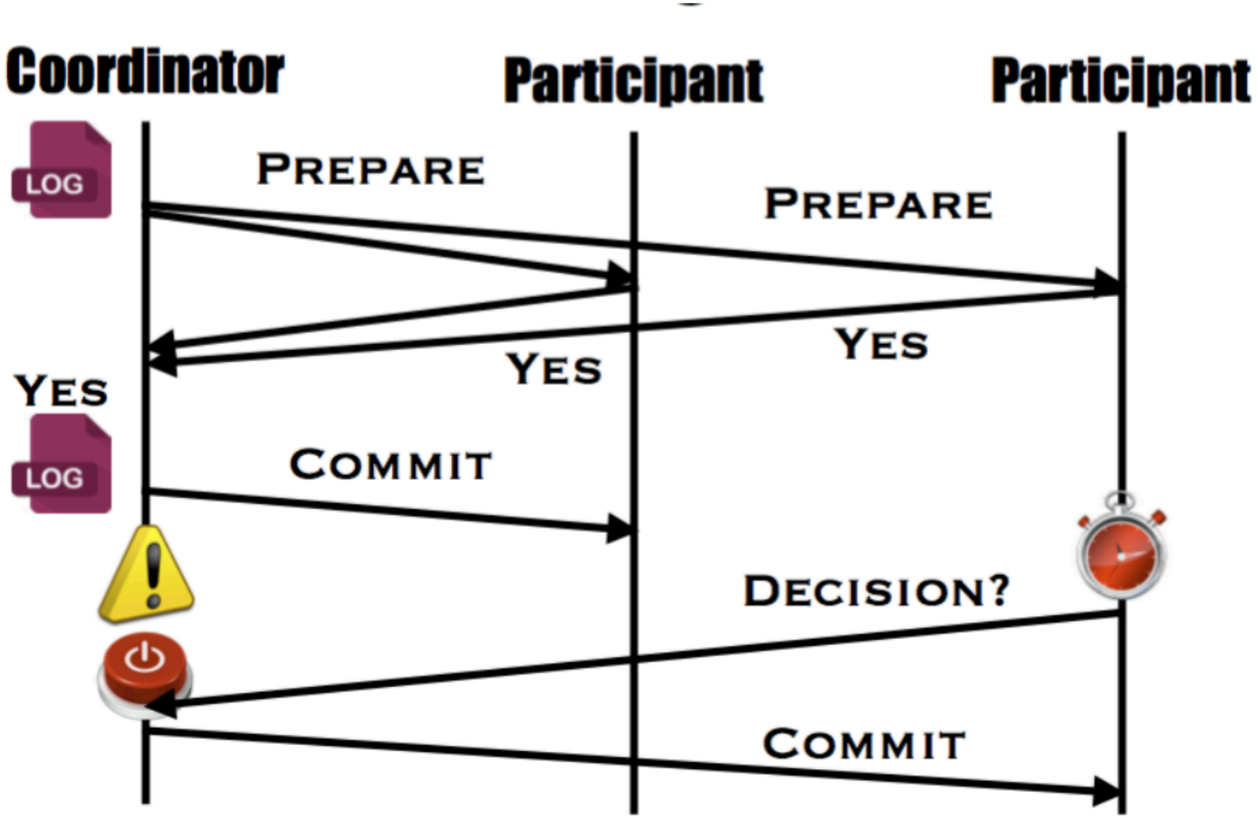
TWO-PHASE COMMIT

- What is 2PC for?
  - Sharding or replication?
- How does 2PC protocol work?
  - atomic commit protocol
  - server termination protocol
  - recovery protocol
- What is its key limitation? (why 2PC can't replace Paxos)?
- Different failure scenarios

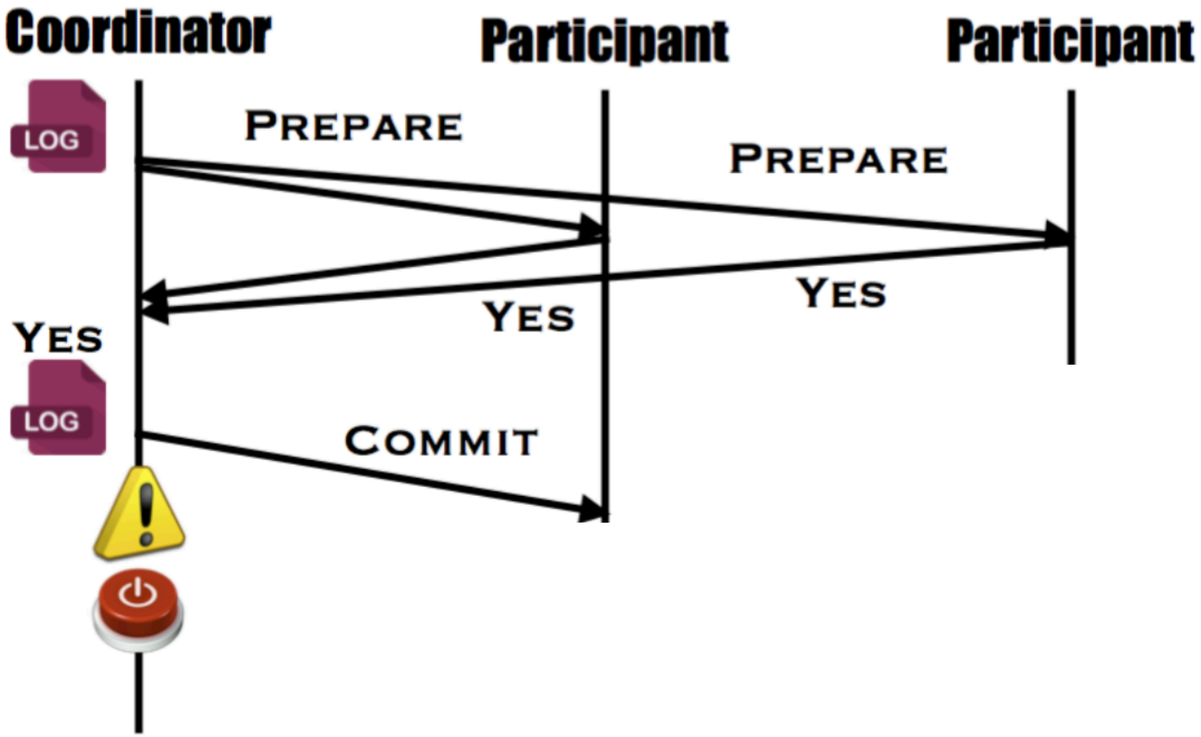
WHAT IF COORDINATOR FAILS AFTER SENDING DECISION?



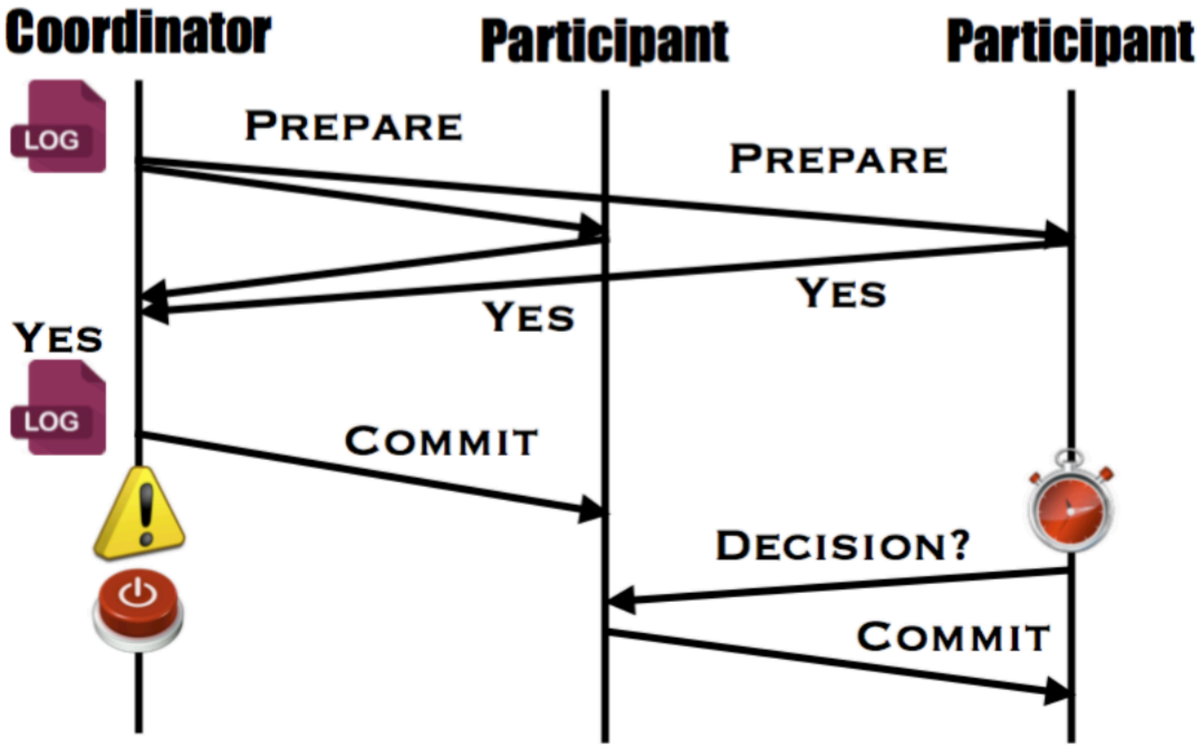
WHAT IF COORDINATOR FAILS AFTER SENDING DECISION?



WHAT IF COORDINATOR FAILS AFTER SENDING ONLY ONE DECISION?



WHAT IF COORDINATOR FAILS AFTER SENDING ONLY ONE DECISION?

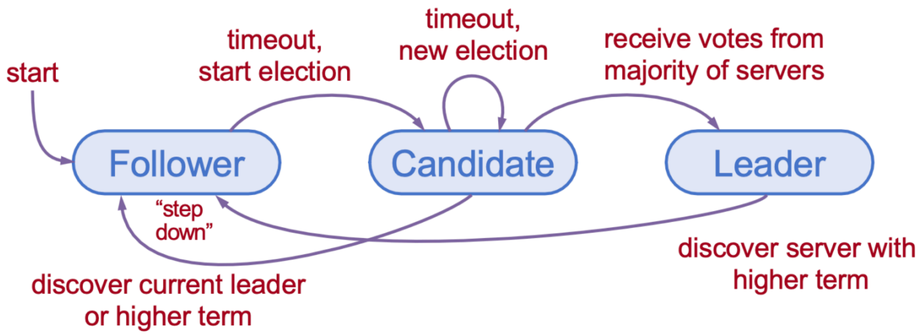


CONSENSUS

- Motivation for replication
- How to replicate shards across servers? (WAL)
- How to resolve dead coordinator problem ("majority")
- What motivates the development of Raft?

CONSENSUS (RAFT)

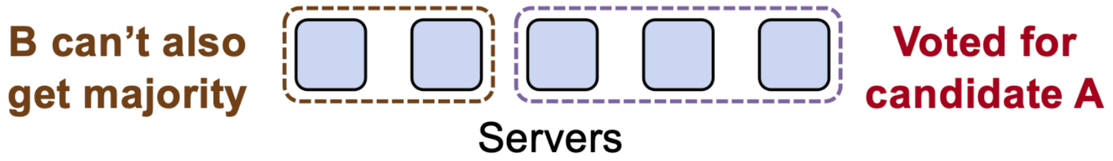
- What are the three states of each server?
  - how they transit to others?



- What does "term" mean in Raft?
  - why we need "term"?

CONSENSUS (RAFT)

- How to start a election for a node?
- What are three possible results for an election?
  - votes from majority, RPC from valid leader, split vote
- How does Raft leader election ensure safety and liveness?



CONSENSUS (RAFT)

AppendEntries RPC

Invoked by leader to replicate log entries (§5.3); also used as heartbeat (§5.2).

Arguments:

term	leader's term
leaderId	so follower can redirect clients
prevLogIndex	index of log entry immediately preceding new ones
prevLogTerm	term of prevLogIndex entry
entries[]	log entries to store (empty for heartbeat; may send more than one for efficiency)
leaderCommit	leader's commitIndex

Results:

term	currentTerm, for leader to update itself
success	true if follower contained entry matching prevLogIndex and prevLogTerm

Receiver implementation:

1. Reply false if term < currentTerm (§5.1)
2. Reply false if log doesn't contain an entry at prevLogIndex whose term matches prevLogTerm (§5.3)
3. If an existing entry conflicts with a new one (same index but different terms), delete the existing entry and all that follow it (§5.3)
4. Append any new entries not already in the log
5. If leaderCommit > commitIndex, set commitIndex = min(leaderCommit, index of last new entry)

RequestVote RPC

Invoked by candidates to gather votes (§5.2).

Arguments:

term	candidate's term
candidateId	candidate requesting vote
lastLogIndex	index of candidate's last log entry (§5.4)
lastLogTerm	term of candidate's last log entry (§5.4)

Results:

term	currentTerm, for candidate to update itself
voteGranted	true means candidate received vote

Receiver implementation:

1. Reply false if term < currentTerm (§5.1)
2. If votedFor is null or candidateId, and candidate's log is at least as up-to-date as receiver's log, grant vote (§5.2, §5.4)

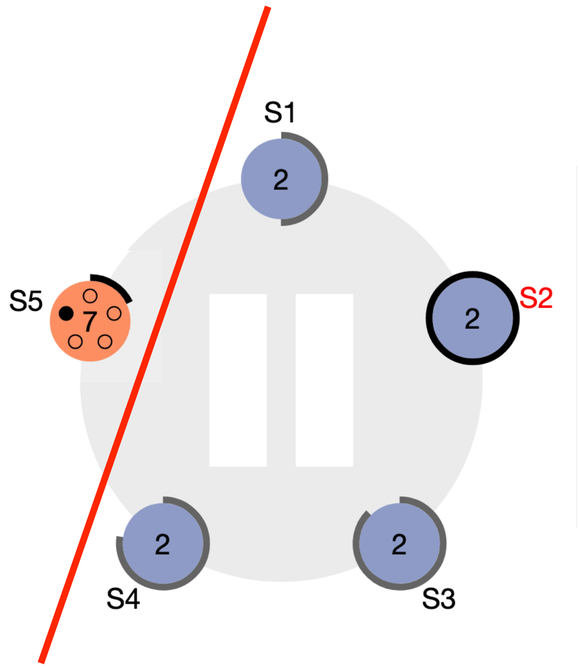
CONSENSUS (RAFT)

- Why sometimes log entries on different server have different index and term?
- Why Raft can guarantee safety even though leaders can often change?
- Why the elected candidate most likely contains all committed entries?
- Why storing on a majority does not mean the entry is committed?
- How to repair followers' logs?

CONSENSUS (RAFT)

- How to deal with old leaders? What if it tries to commit?
- Exercise

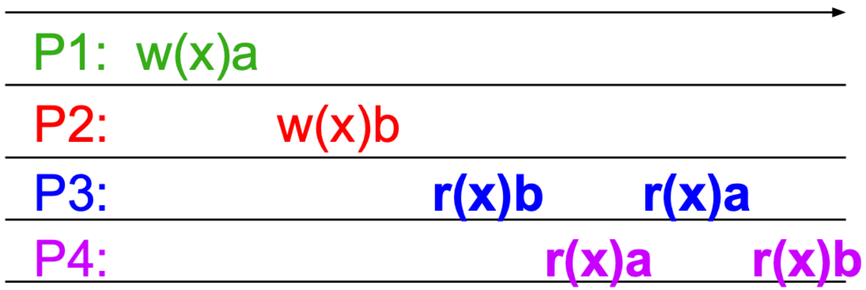
What happens after the network partition heals? Who will become the leader?

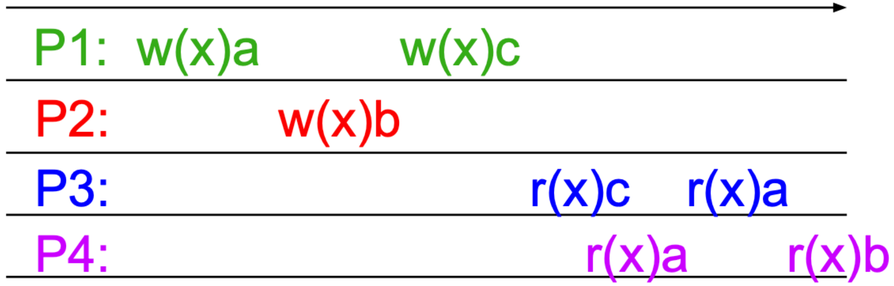


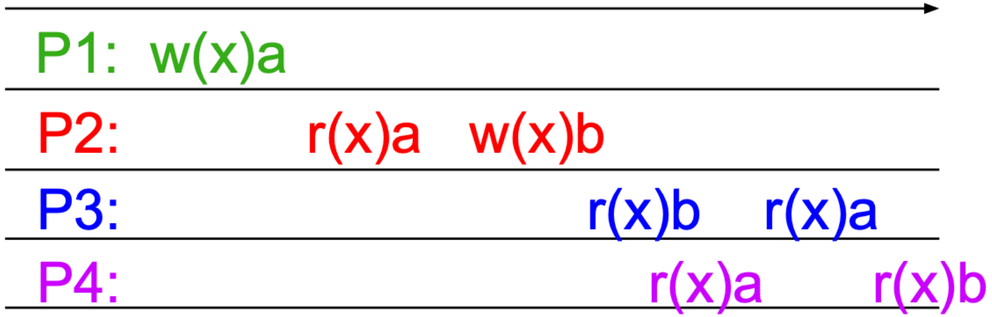
ISOLATION AND CONSISTENCY

- What are the common isolation levels? Difference?
  - why we need different levels?
- Give examples on dirty/fuzzy/phantom reads? Difference?
- How to implement different isolation levels?
- What are the common consistency models? Difference?
- Can you judge if a DSM execution satisfies semantic?

ISOLATION AND CONSISTENCY







ISOLATION AND CONSISTENCY

- Tips for different consistency models:
  - strict consistency: does the result match the physical time sequence?
  - sequential consistency: is there a global sequential order that can explain?
  - causal consistency: which writes are concurrent? their order can be seen differently on different nodes

GOOGLE FILE SYSTEM

- What motivates the development of GFS?
  - capacity, performance, fault-tolerance..
- Why GFS splits big files into chunks?
- Why GFS can have high parallel throughput?
- Workflow for basic ops
  - read, write
  - how to handle concurrent writes

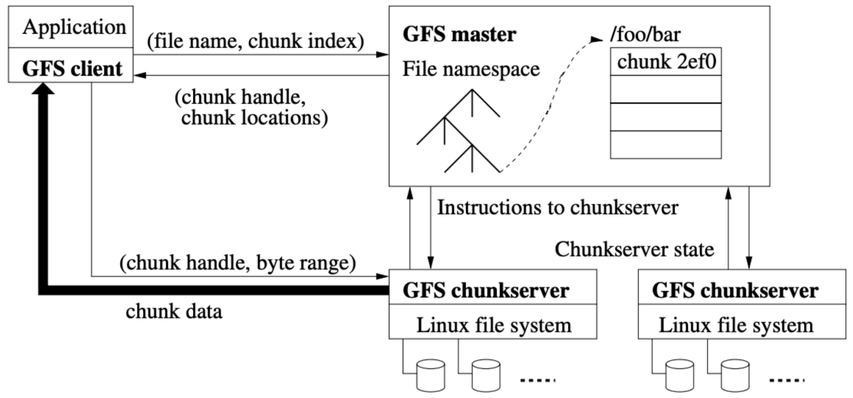


Figure 1: GFS Architecture

GOOGLE FILE SYSTEM

- How to handle different failure cases
  - client crashes?
  - a secondary crashes?
  - a primary crashes?
  - coordinator crashes?
- Design review
  - good: separation of naming (chunk server), primary, lease
  - not so great: single coordinator bottleneck, small file support, consistency

ZOOKEEPER

- What are coordination services for
- ZooKeeper data model
  - tree, different types of znodes, operations
  - session management
- Programming with ZooKeeper
  - simple lock? lock without herd effect?
  - how to implement leader election?

VIRTUALIZATION

- What are benefits of virtualization
- What are popular VMM softwares
- Difference between VMM and containers

ML SYSTEMS

- Workflow of a typical ML job
- What ML systems can provide for ML tasks?
- Difficulty of designing and implementing ML systems

RELIABILITY

- How fuzz testing works?
  - key idea?
- How static analysis works?
  - what does soundness and completeness mean
- How model checking works?
  - can you draw a state-transition diagram for a program?
- Advantages and disadvantages of different approaches

TAKEAWAYS

- Office hours this week:
  - Monday: 4pm-5pm (4/29)
  - Wednesday: 4pm-5pm (5/1)
  - Friday: 4pm-5pm (5/3)
- Next class: **final exam**
  - Time is not enough? Complete the SET to get a head start :)

CS 4740 - 001 Cloud Computing

Deadline: this Friday