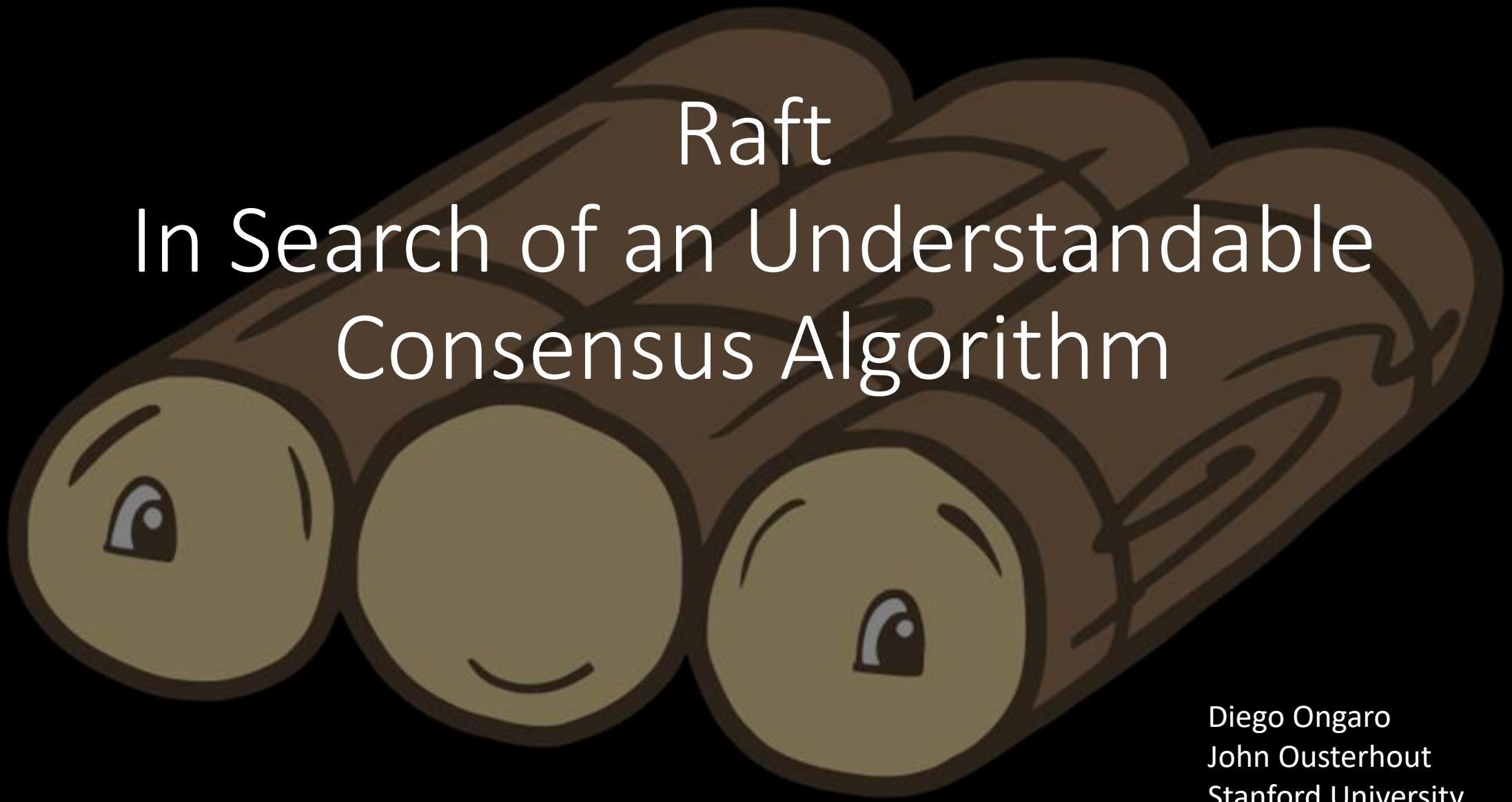


Raft
In Search of an Understandable Consensus Algorithm

Diego Ongaro
John Ousterhout
Stanford University
2014

About Me

Daniel Kiptoon

Senior Software Engineer

Vantage Intellica

Machine intelligence startup that aggregates open-source data to change the way data is consumed by impacting the way analytics are delivered

Interests

Distributed computing, Containerization, Automation

Outline

- Introduction
- Leader election
- Log replication
- Safety
- Follower and candidate failure
- Cluster membership changes
- Log compression
- Understandability

Introduction

- Replicated state machines
- What's wrong with Paxos?

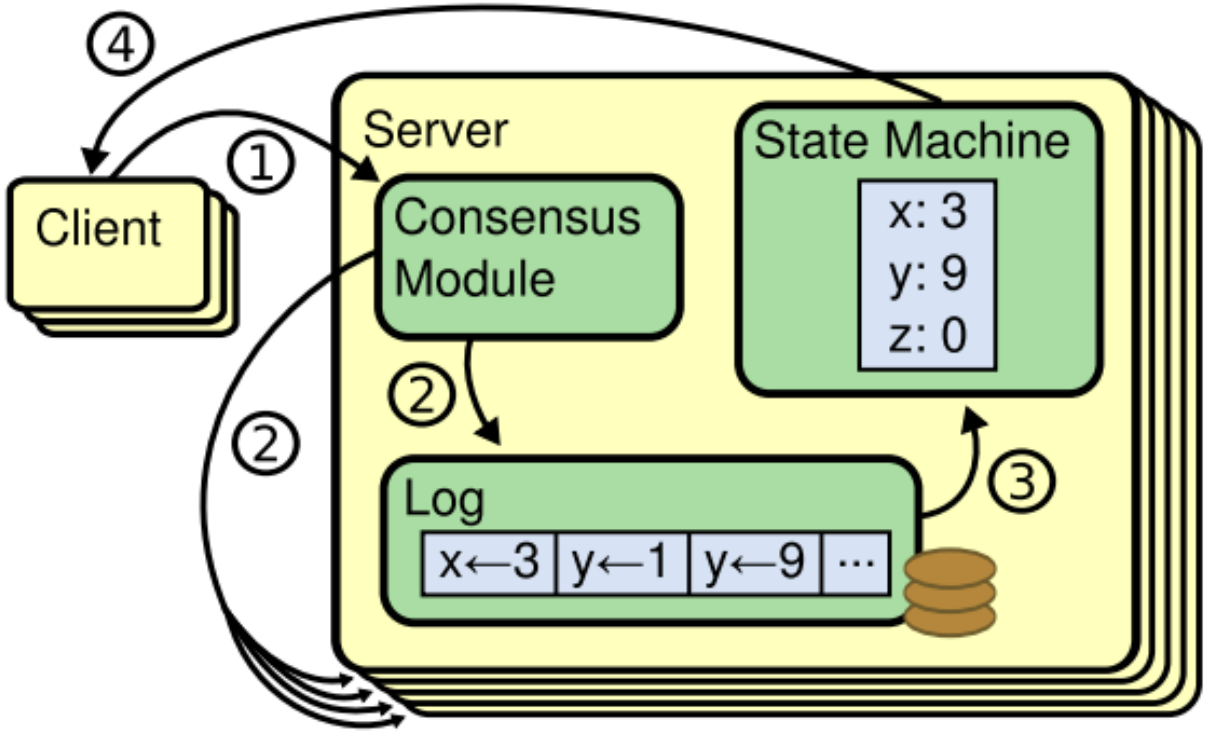


Figure 1: Replicated state machine architecture. The consensus algorithm manages a replicated log containing state machine commands from clients. The state machines process identical sequences of commands from the logs, so they produce the same outputs.

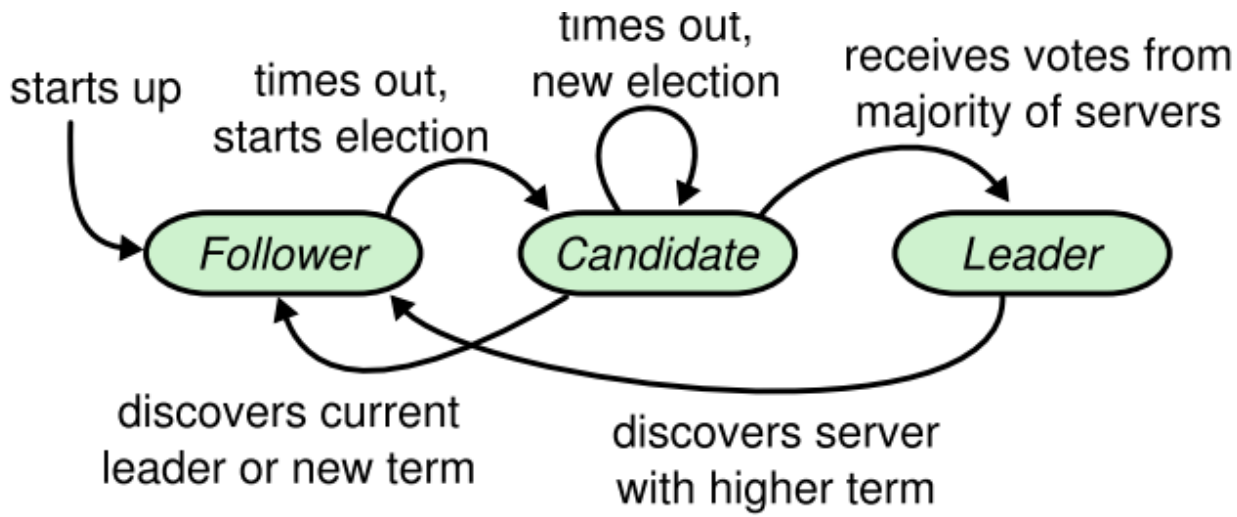


Figure 4: Server states. Followers only respond to requests from other servers. If a follower receives no communication, it becomes a candidate and initiates an election. A candidate that receives votes from a majority of the full cluster becomes the new leader. Leaders typically operate until they fail.

Raft basics

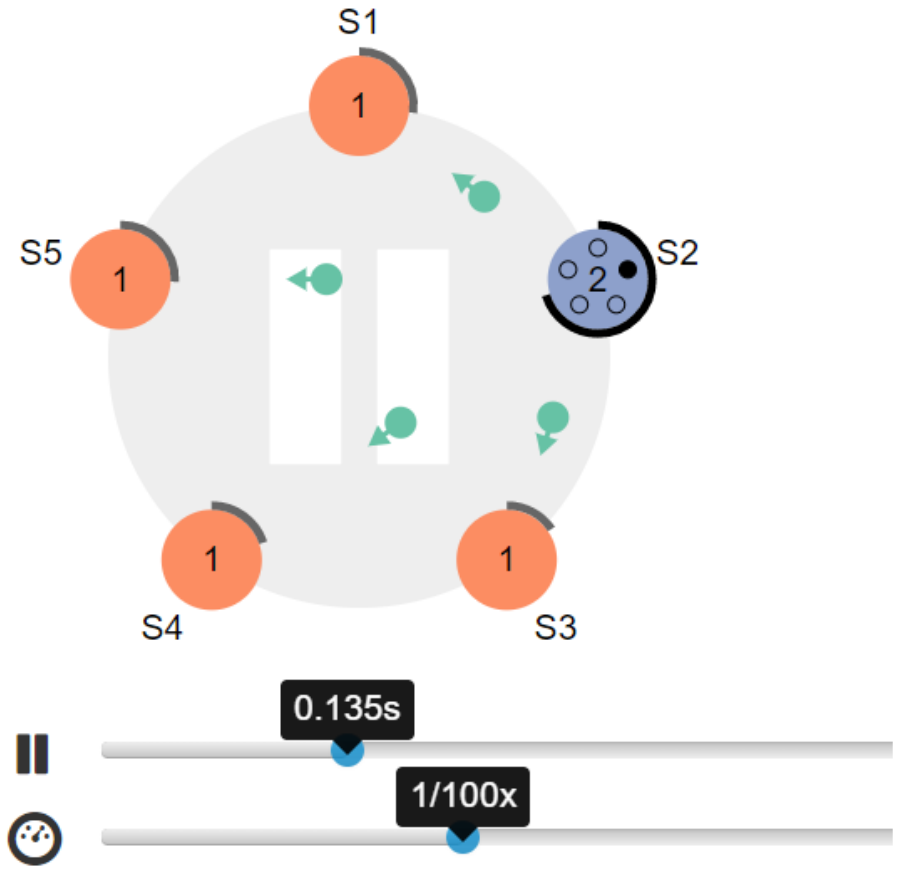
Election

Terms

RequestVote RPC

AppendEntries RPC

Leader Election



- Randomized election timeout 150-300ms
- Split votes result in election timeout

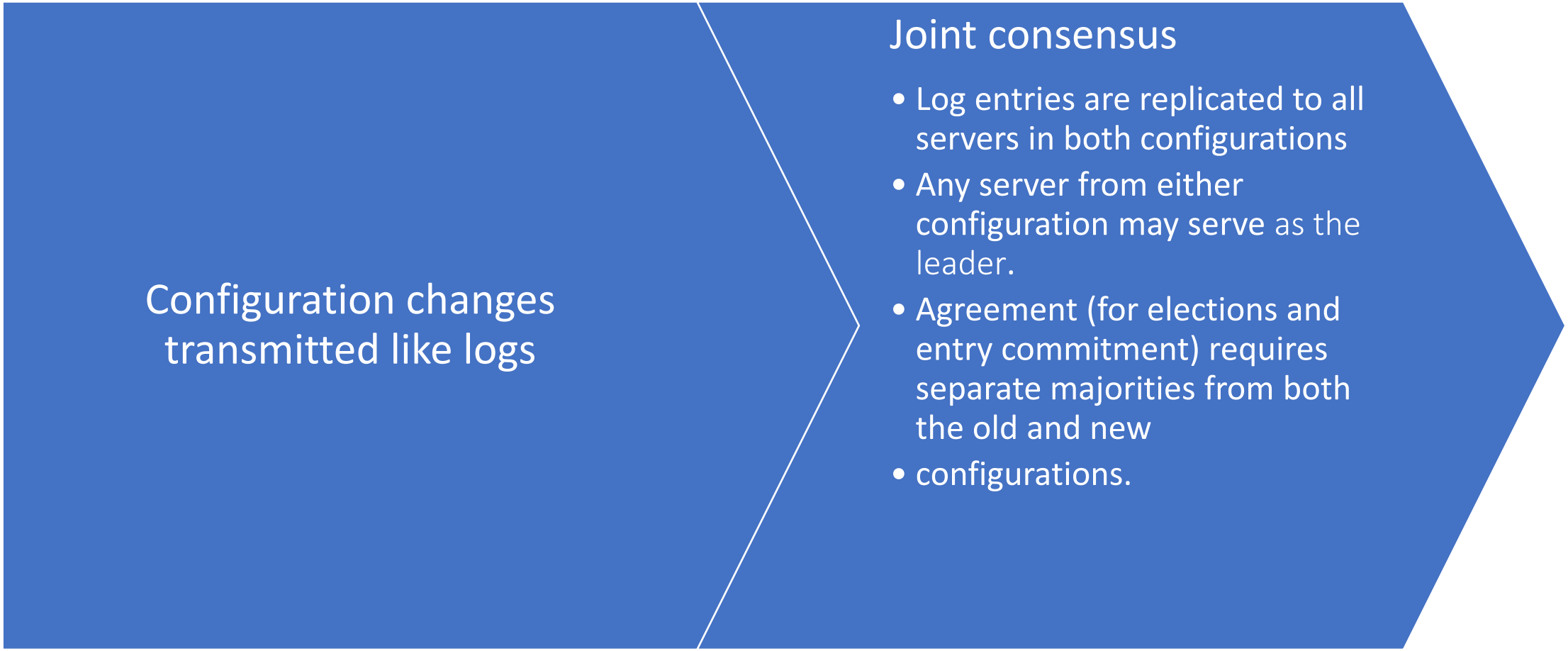
Log replication

- Leader receives request from client
- Sends out AppendEntries RPC in parallel to followers
- Committed entry
- Index of committed entry included in future heartbeats
- Maintains Log Matching property

Safety

- Election restriction using terms and log length
- Leader Completeness Property
- Follower and candidate crashes
  - Leader retries RPC indefinitely
  - AppendEntries RPC is idempotent
- Timing and availability
  - broadcastTime $\ll$ electionTimeout $\ll$ MTBF

Cluster Membership Changes



Log Compaction

- Each server takes log snapshot independently
  - last included index
  - last included term
  - latest configuration

- Leader occasionally send snapshot to followers (InstallSnapshot) RPC

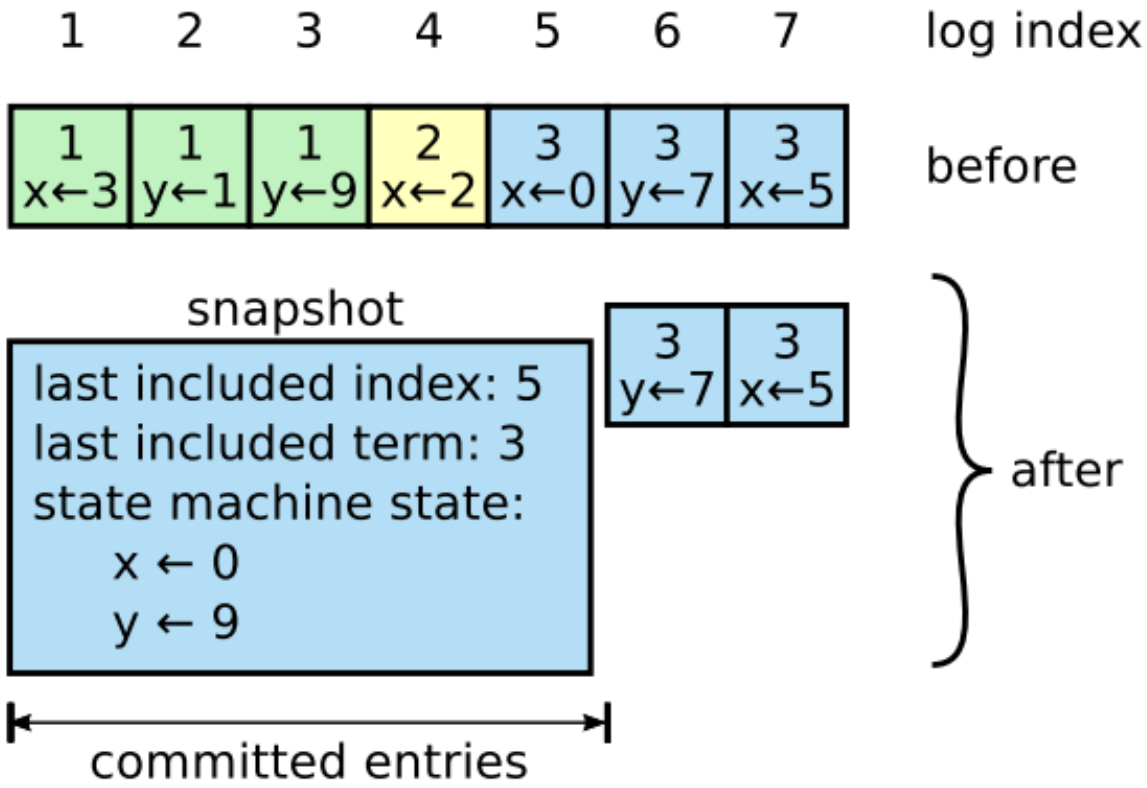


Figure 12: A server replaces the committed entries in its log (indexes 1 through 5) with a new snapshot, which stores just the current state (variables *x* and *y* in this example). The snapshot's last included index and term serve to position the snapshot in the log preceding entry 6.

Understandability

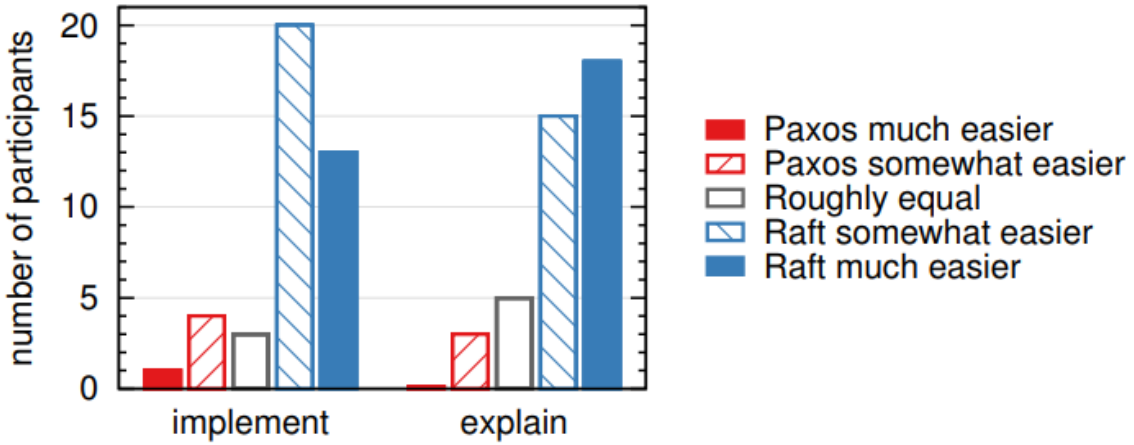


Figure 15: Using a 5-point scale, participants were asked (left) which algorithm they felt would be easier to implement in a functioning, correct, and efficient system, and (right) which would be easier to explain to a CS graduate student.

Peaking your interest....

References

- https://raft.github.io/
- https://raft.github.io/raftscope-replay/index.html
- http://thesecretlivesofdata.com/raft/