In Search of an Understandable Consensus Algorithm

Diego Ongaro and John Ousterhout

Stanford University

Presented by - Sarthak Moorjani
CS425 / ECE428 (Spring 2024)

“All distributed consensus algorithms are either Paxos, Paxos with an extra unnecessary cruft, or broken.”

-- Micheal Burrows, Google

Paxos

Basic Paxos or single-decree Paxos

- agree on a single value, not a log
- simplest form of consensus

Discussion: Can we build practical applications on top of this?

Not really! We need a replicated log for such purposes.

How to agree on a log?

- Run a Paxos on every log slot.
- Called Multi-Paxos.

- Very difficult and complicated!!

"The dirty little secret of the NSDI community is that at most five people really, truly understand every part of Paxos ;-)."

- Anonymous NSDI reviewer.

Raft: Key Takeaways

- Paxos is hard!
- With increasing systems which are distributed in nature, consensus is very important, therefore we need something else!
- Raft **clearly** separates log replication and leader election, which makes the algorithm very easy to understand.

Raft: *In Search of an Understandable Consensus*

Basics -

- Uses a leader-based approach

- Leader tells the other followers about what to do.

Leader based approach

Discussion: Advantages?

- EASY and less complex!!
- No conflicting commands from multiple leaders.
- Multiple Prepare-Accepts possible!

Leader based approach

Discussion: Disadvantages?

- Leader bottleneck – CPU Contention
- Geo-Replicated systems – Network Bandwidth

Raft Basics - Terms

- Time is divided into **terms.**
- Each term begins with a leader election.
- Continue operating in the same term till another election happens.'

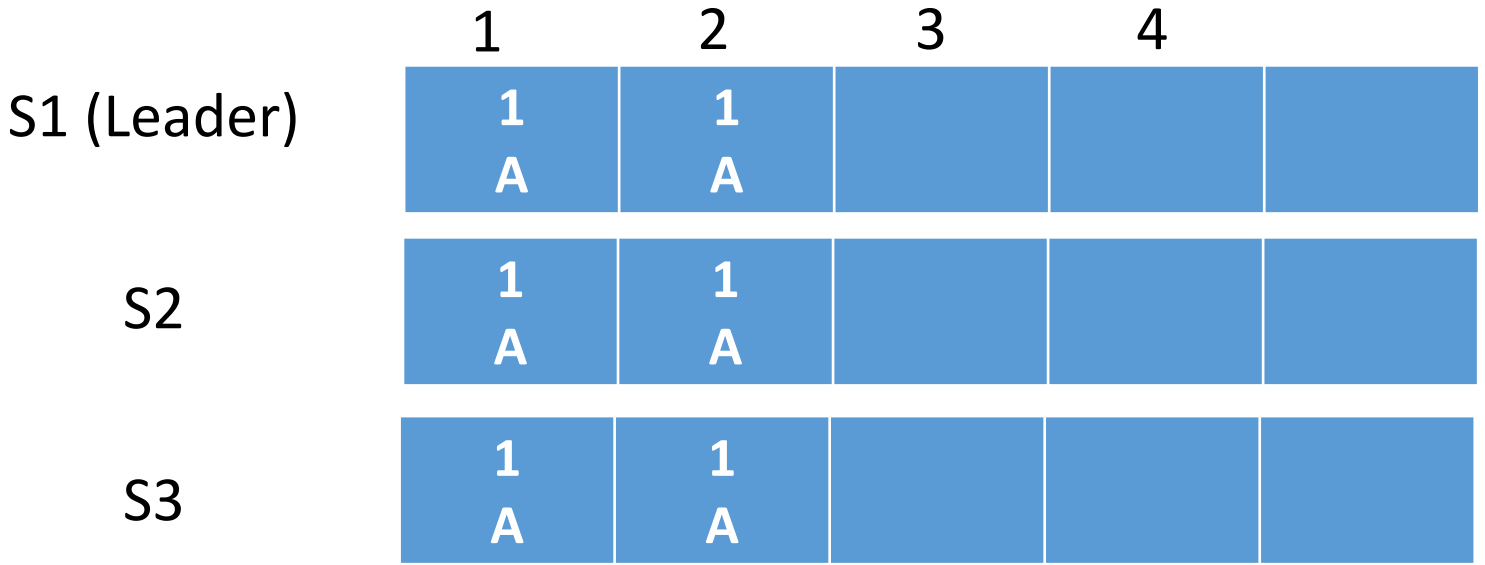
- Atmost one leader per term!

Raft Basics - Terms

- Each node maintains the current term – Best guess of current term of the leader.
- Terms are monotonically increasing, and never re-used.
- Example, if S1 is in term 2 and S2 is in term 4, then S2 knows that S1 is in older state.

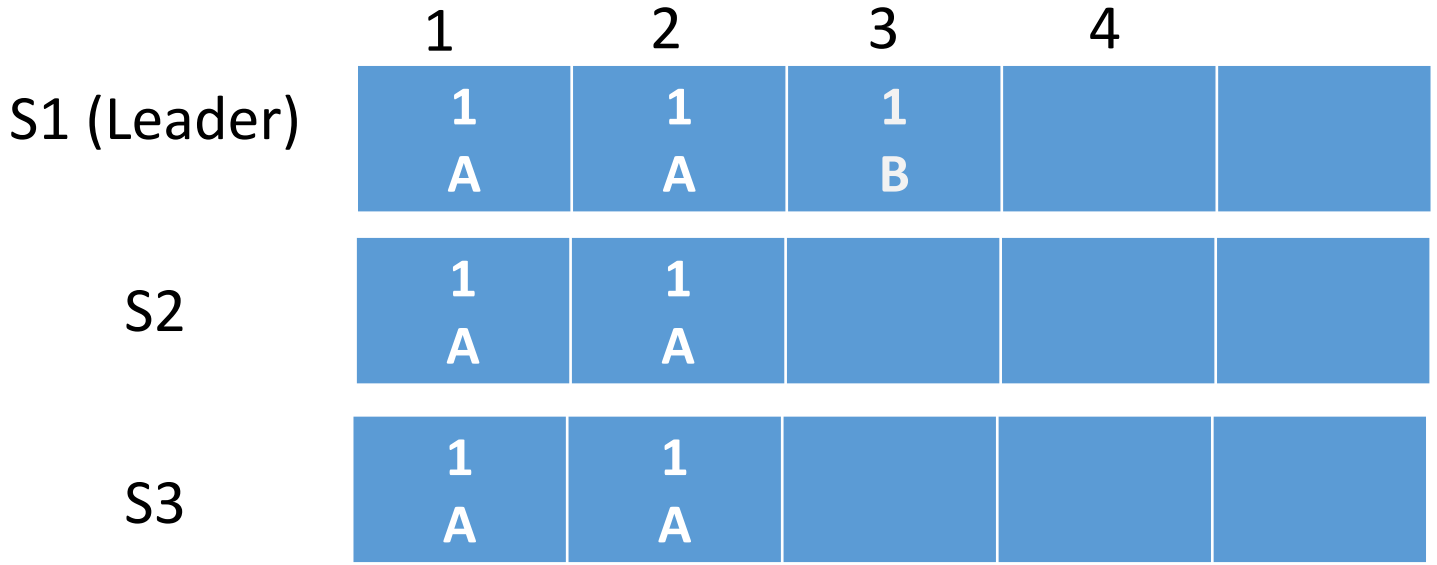
Raft Basics – Happy case

- Assume leader is elected
- All followers are also up and running



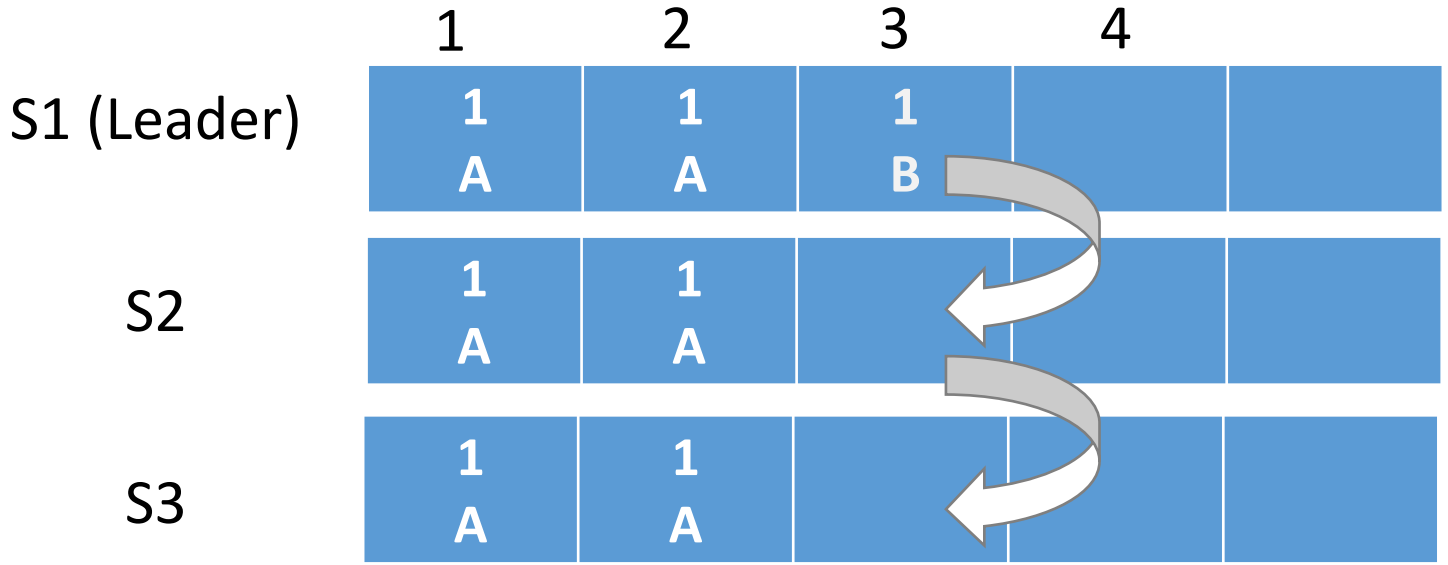
Raft Basics – Happy Case

Lets say a new request comes to the leader with command "B". The leader first appends it to its own log.



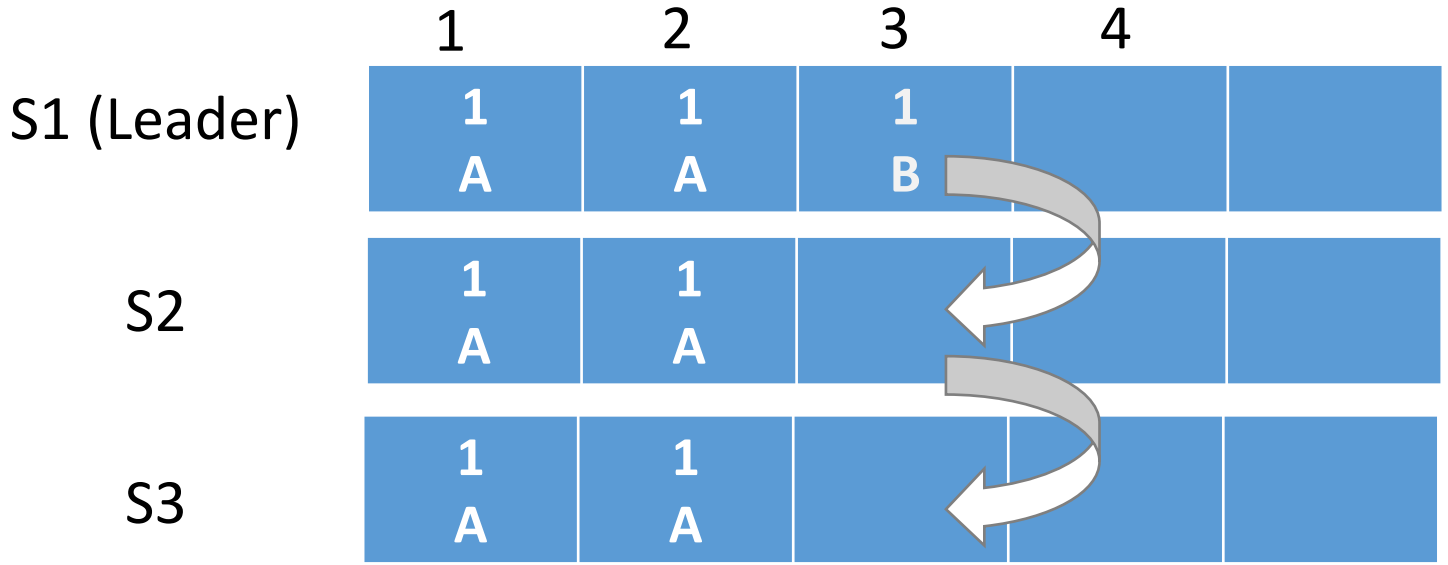
Raft Basics – Happy Case

Next, the leader sends the AppendEntry Request to the followers to append the entry in their logs.



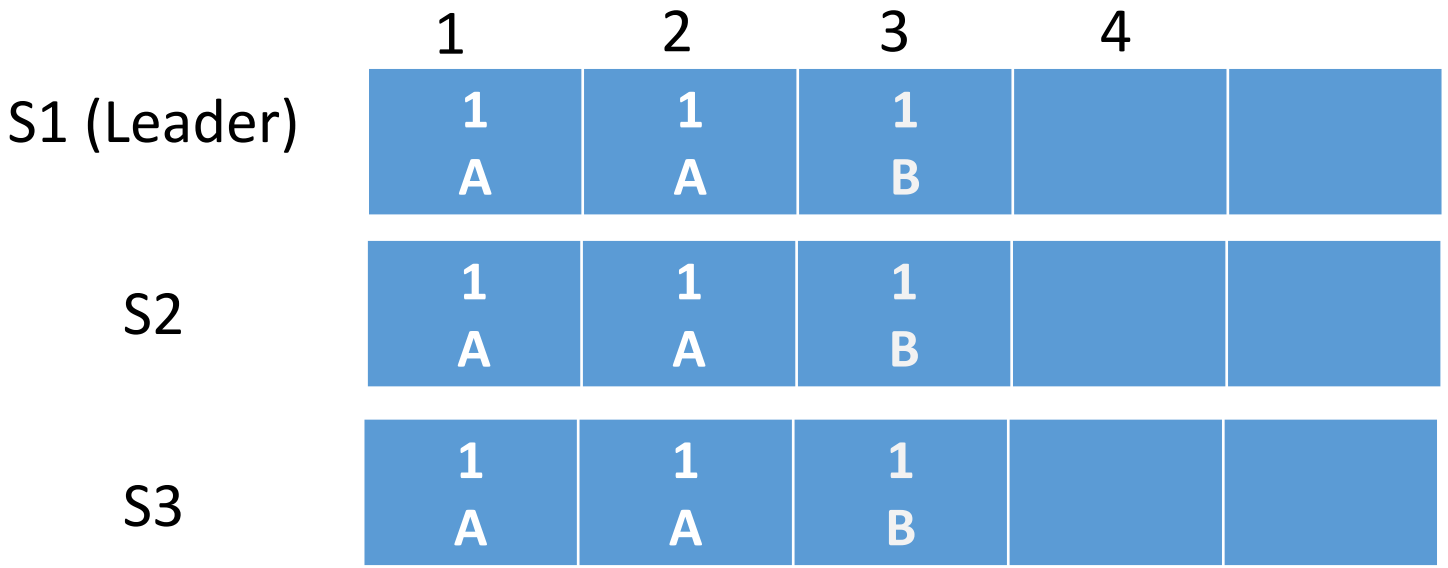
Raft Basics – Happy Case

Next, the followers check if the prefix of their logs matches the prefix of the leader's log



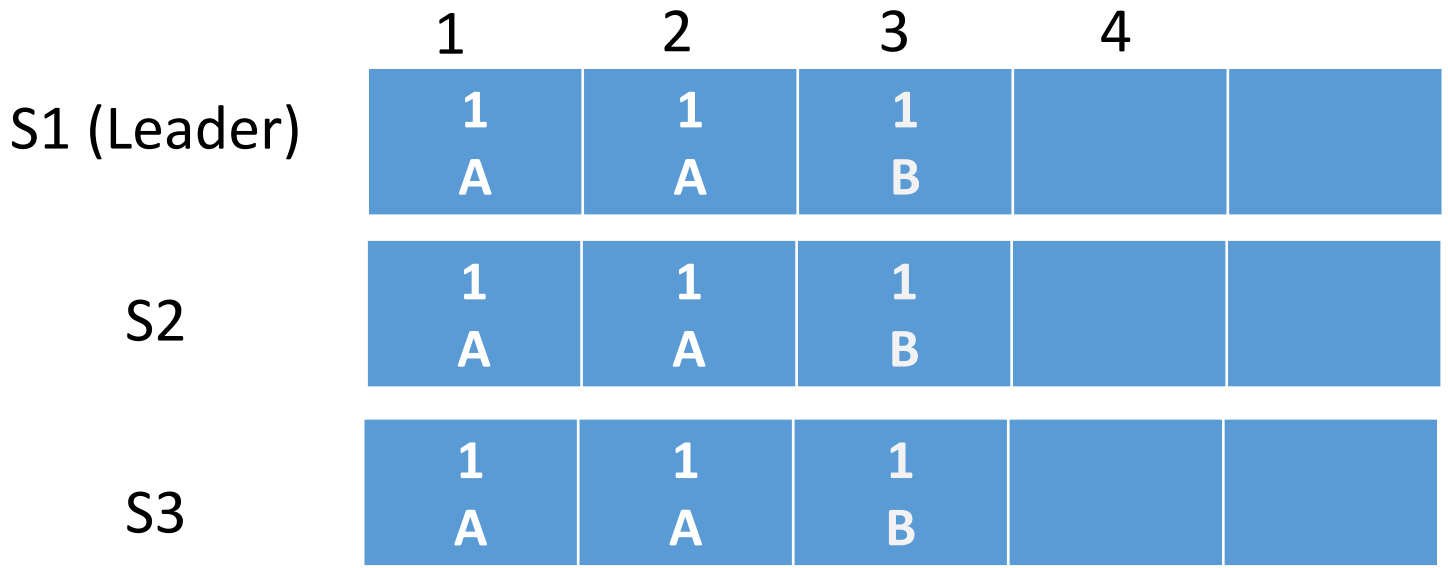
Raft Basics – Happy Case

Next, the followers append their logs and acknowledge the leader.



Raft Basics – Happy Case

When leader receives majority acknowledgements, it commits into the state machine and replies to client.



Raft Basics – Happy Case

Discussion/Question: Can the followers apply the command to the state machine as soon as they append to the log?

How to detect failures

Heartbeats
What to do when heartbeats stop?

Absence of heartbeats can mean crash/partitioned.

Call an election after the "timeout".

Leader Elections

- The follower runs an election after the timer goes off.
- Converts itself to candidate and does a RequestVote RPC.

Leader Elections

Election Restrictions -

- Vote yes only if the candidate has a higher term in the last entry, or -
- It has the same term and a greater log term.

- Leader completeness: If a log entry is committed, that entry must be present in the logs of all the future leaders.

Discussion

Can a server give vote to more than one candidate?

No! Because, we cant have more than one leader.

How is it ensured in Raft?

The server **persists** the information about who it **votedFor** in the current term.

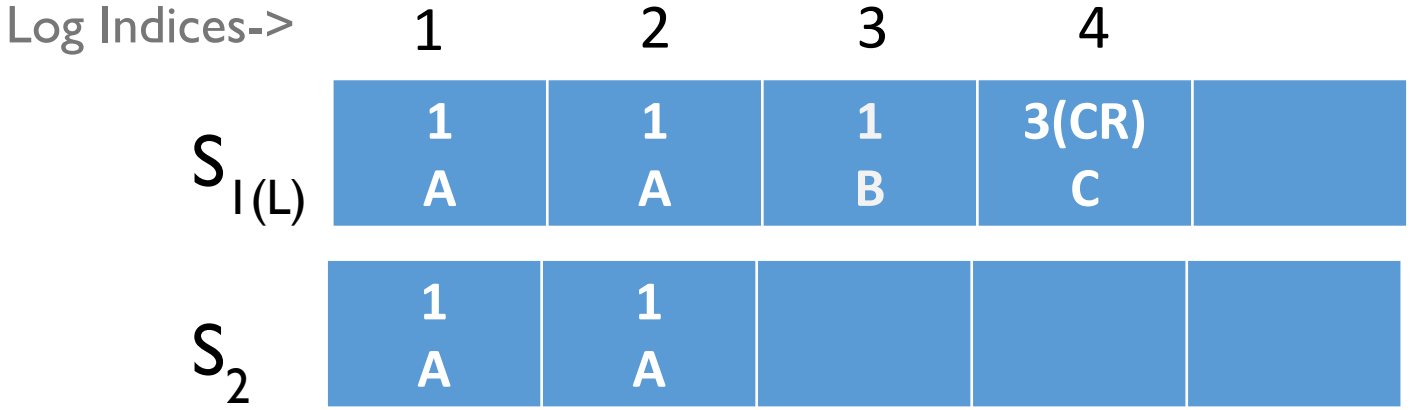
Discussion

Can an election candidate requesting for votes actually receive an AppendEntry RPC?

Yes! That means a new leader was already elected and the candidate steps down and becomes a follower.

Catching up with Leader (1)

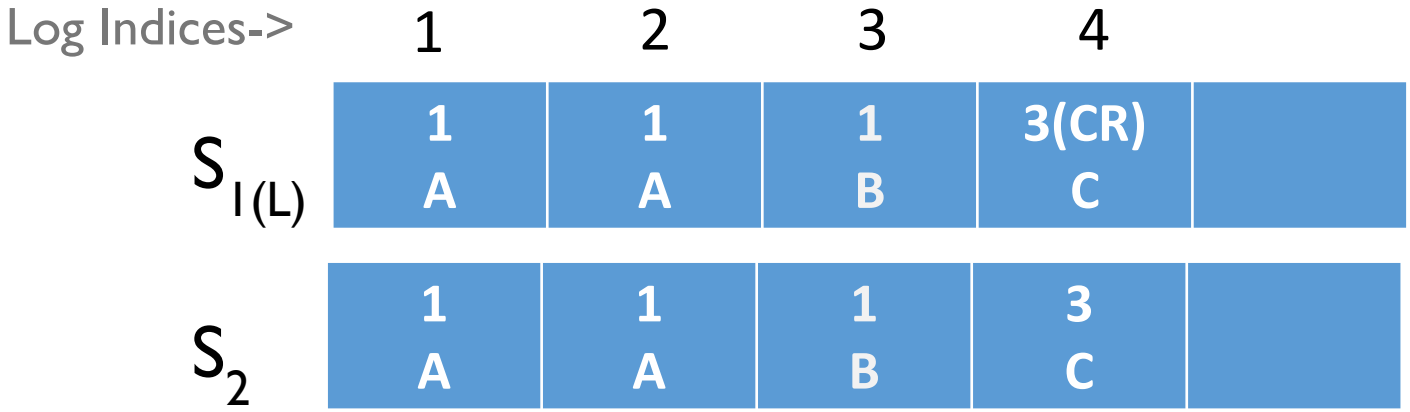
Current term is 3, S1 is the leader and is trying to replicate log index 4.



Follower is missing entries --> Crashed before seeing <3, 1B>

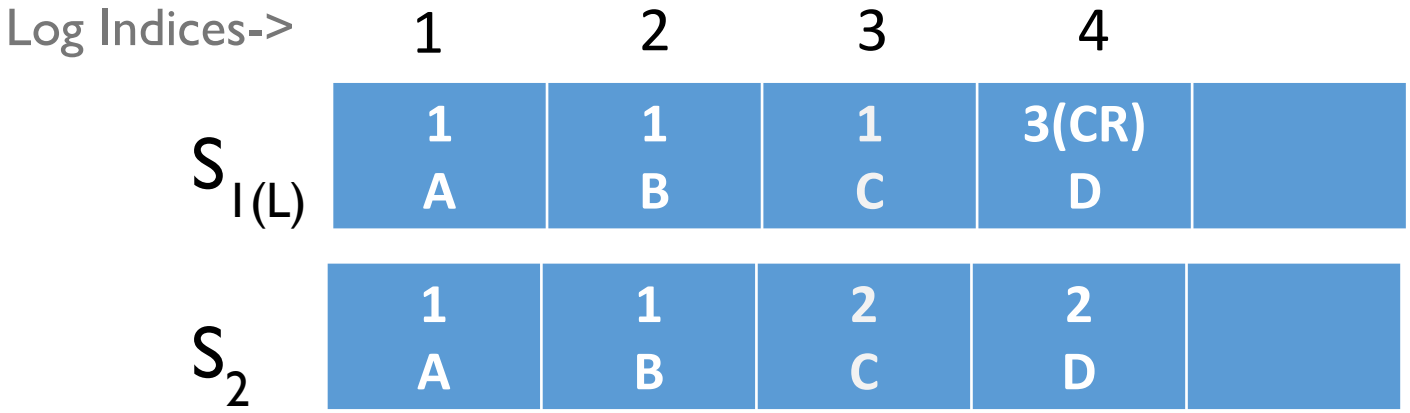
Catching up with Leader (1)

Current term is 3, S1 is the leader and is trying to replicate log index 4.



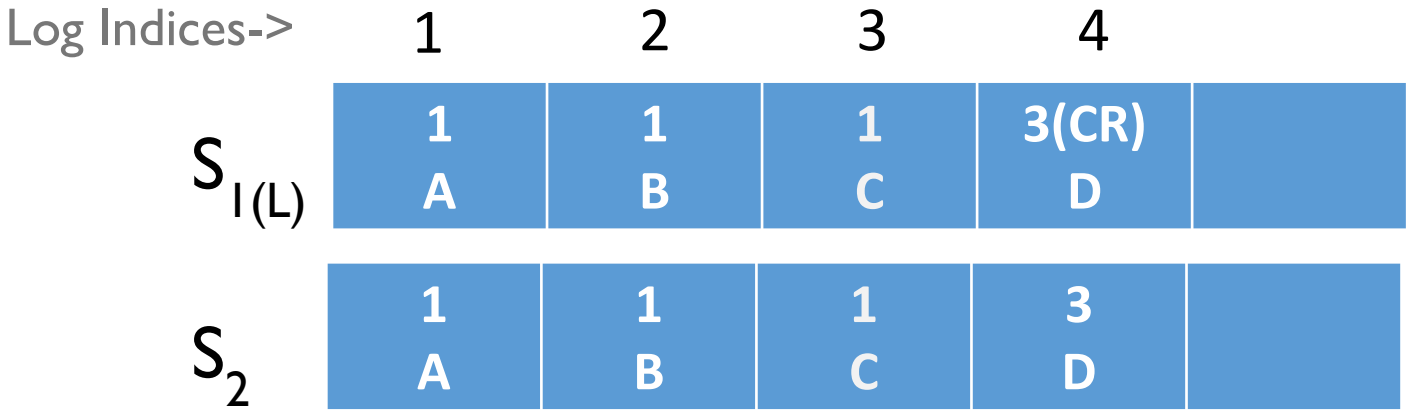
Leader decrements next index and tries to replicate again.

Catching up with Leader (2)



Some missing entries, some extra entries.

Catching up with Leader (2)

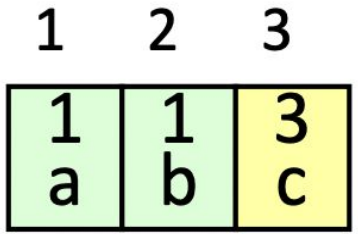


Similarly, remove tail and replicate.

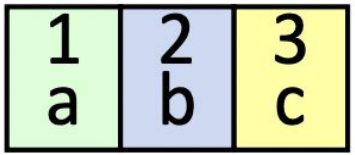
Log States – Valid or not?

Discussion / Question

Let the leader log be -->



Can the follower log be -->?



Question!

Is this a valid state? Assume S1 is current leader for term 3 and is trying to replicate <4, 3> (LogIndex, Term)

Log Indices->	1	2	3	4	
S1 (Leader)	1	1	1	3(CL)	
S2	1	1	1		
S3	1	1	1		
S4	2	2			
S5	2	2			

Take Home Question/Discussion

Consider the following log states (This is valid in Raft). Which all entries can be safely applied to the state machine by S1 (current leader) (Current term is 4)

Log Indices->	1	2	3	4	5	6
S1 (Leader)	1	1	2	2	2	4
S2	1	1	1	3		
S3	1	1	2	2	2	
S4	1					
S5	1	1	2	2	2	2

Ack: Prof. Aishwarya Ganesan

Questions?