

Day 17: CS 1380

---

---

---

---



Today

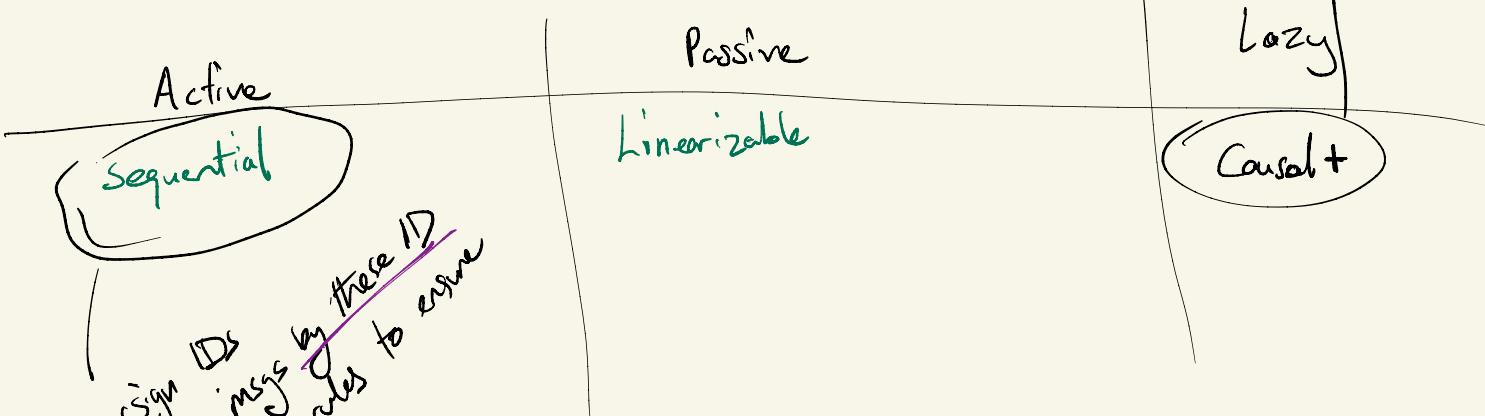
Top Hat Question  
Replication & Ordering  
CAP Theorem  
Practical Replication Service

$$\text{Linearizable} = \text{total} + \text{FIFO} + \text{head time}$$

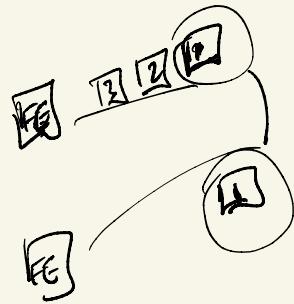
when leader receives / head time according to the leader

$$\text{Sequential} = \text{total} + \text{FIFO}$$

\* is because with lazy data or have conflicts & you can have conflicts



FC assign IDs  
 & RM orders msgs by these ID  
 & tie breaking rules to ensure  
 total ordering



## CAP Theorem

Consistency = reads will return the value of last/recent write according to some ordering constraint

Availability = reads/writes will be processed by the systems no hanging

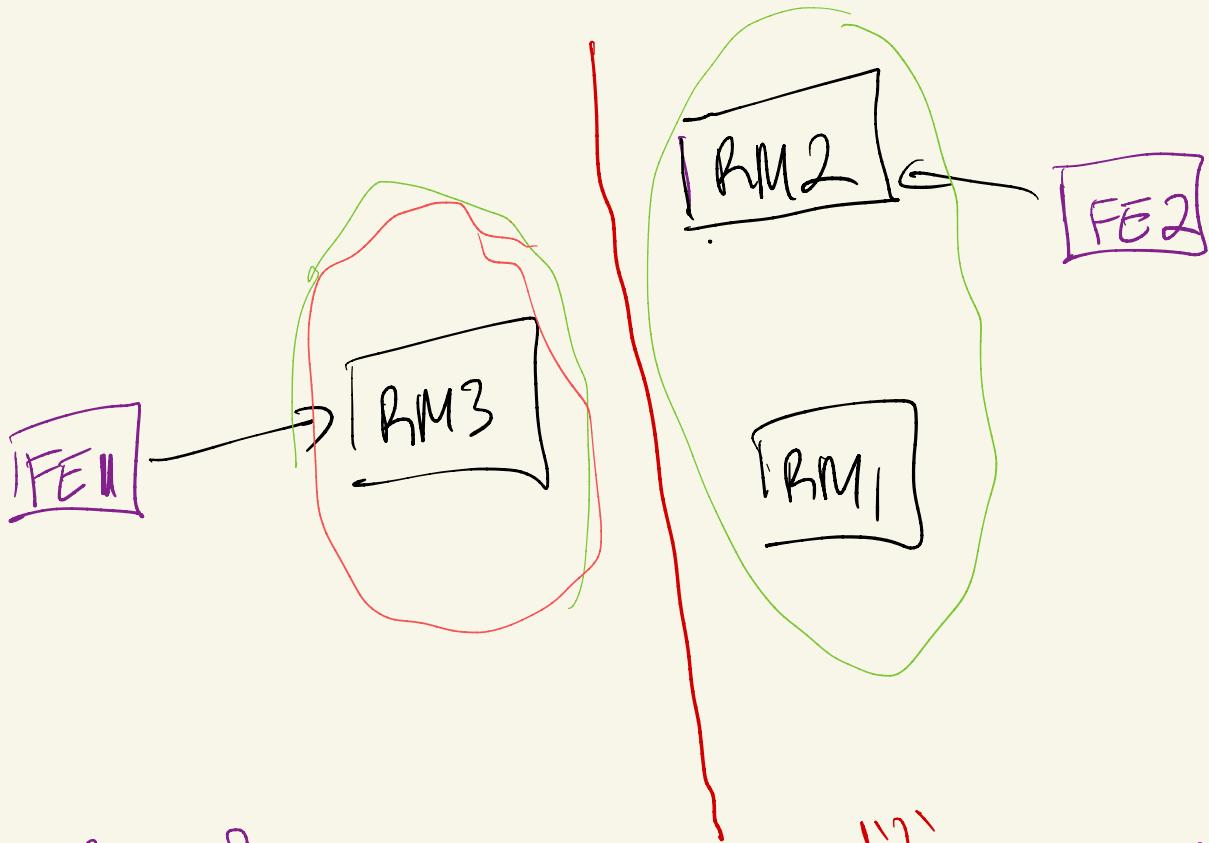
Partition = during a network partition some nodes in the system can make forward progress in protocol

You can only HAVE TWO of the three

**GA** - if there partitions (failures) the protocol hangs

**CP** - during a partition, subset of the nodes will be unavailable because they want to provide consistency

**AP** - during partition, all nodes are available  
↳ data may get lost



Mongo DB      Level DB



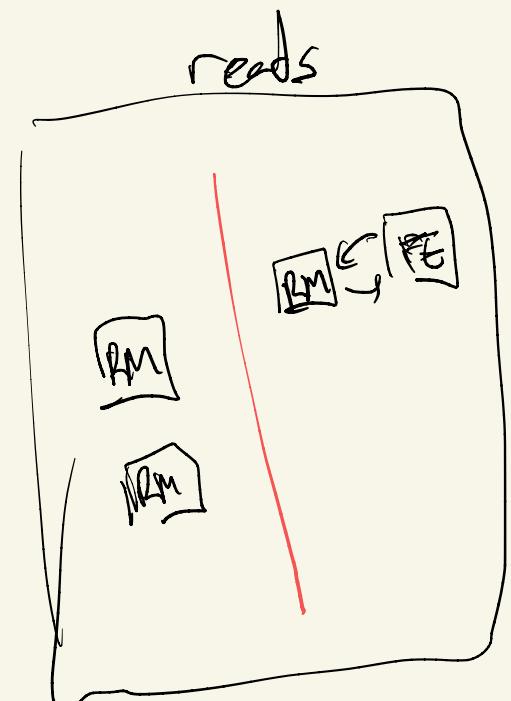
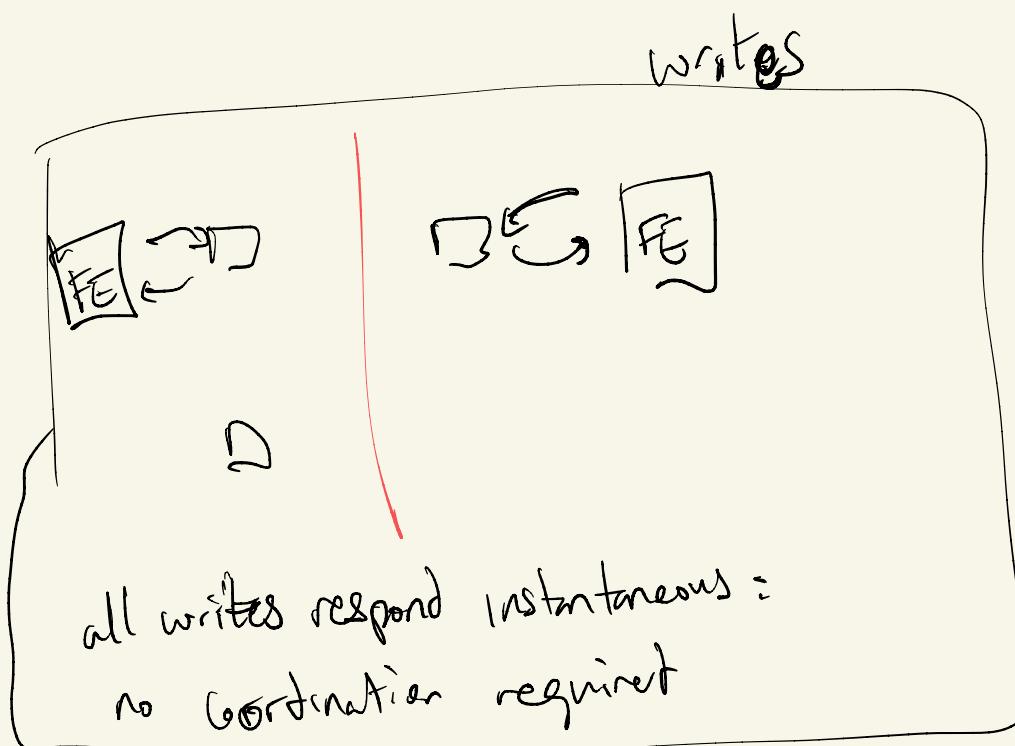
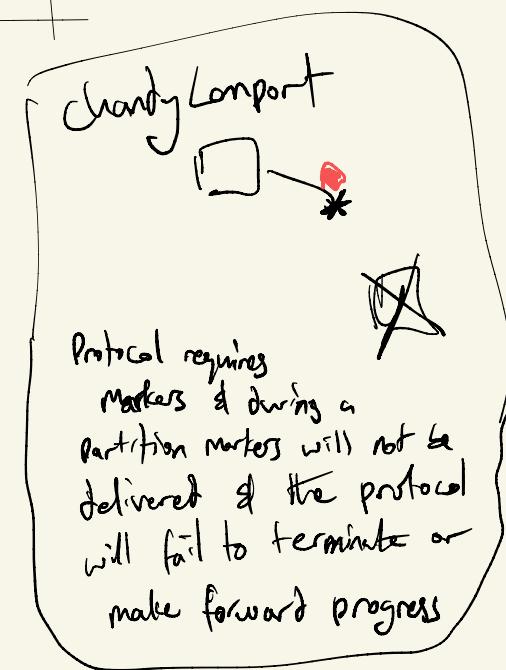
GP = maybe neither  
FE can read/write  
until the partition  
is over

Network Partition

AP = but FE  
can keep read/write  
to their RIMs

Cassandra dynamic

	CA	CP	AP
Raft		✓	
Lazy Replication			✓ (circled)
Chandy-Lamport		✓	

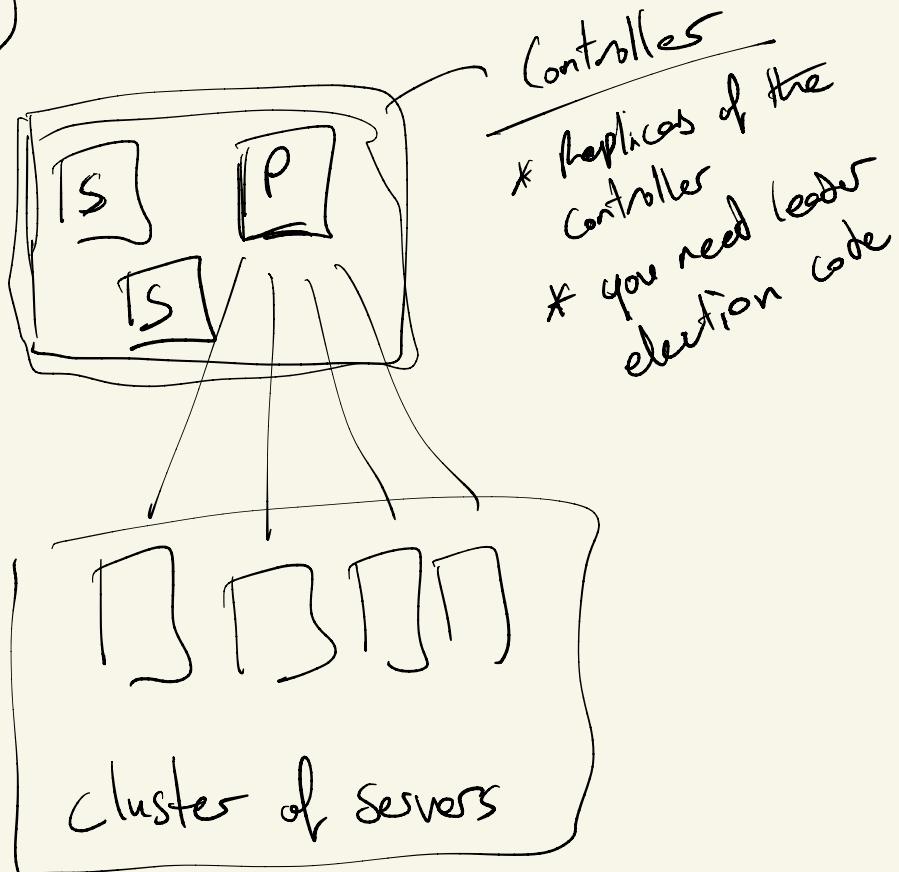


# Practical Consensus (Practical Passive Replication)

(Zookeeper)

## MapReduce (Primary)

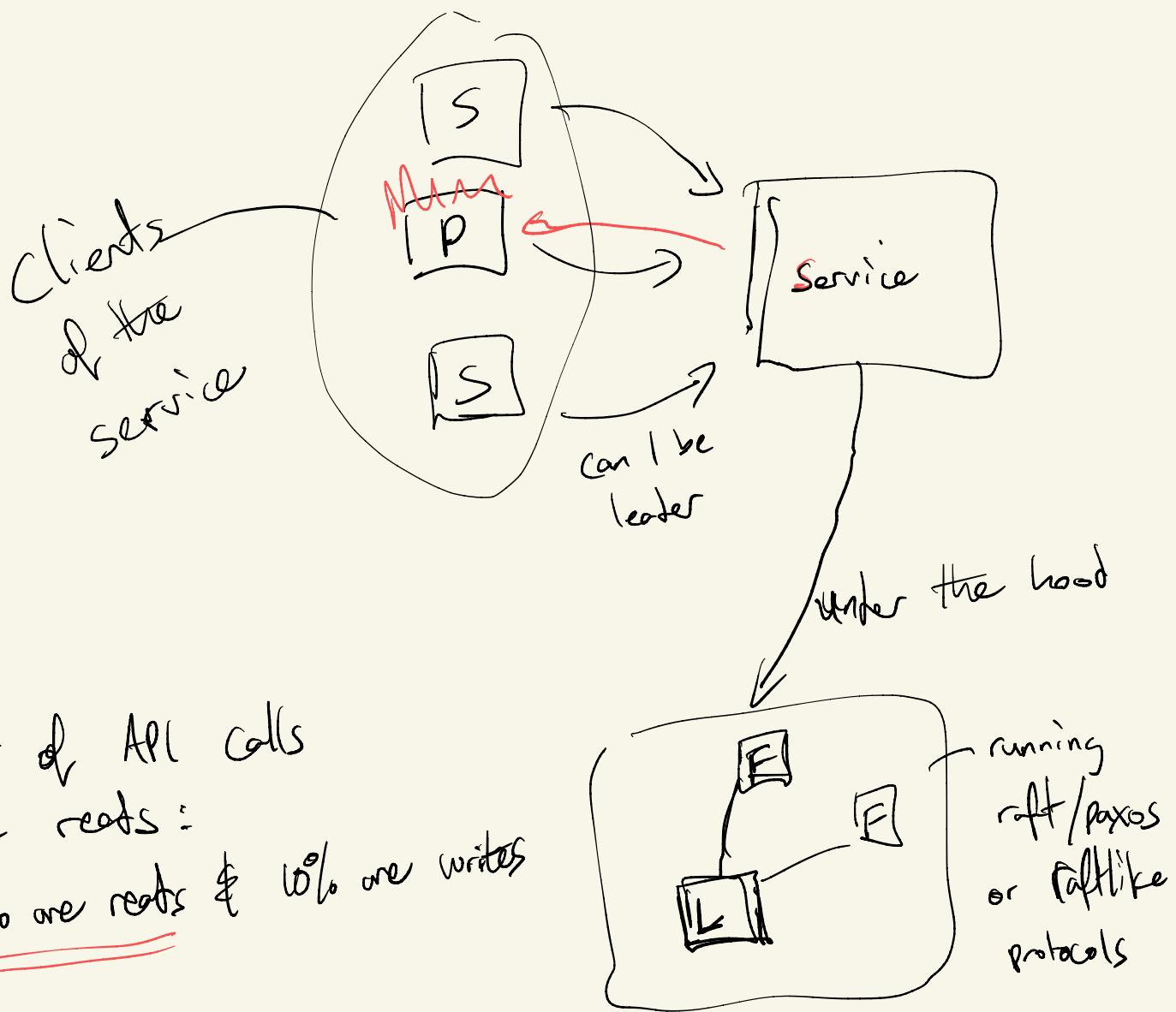
- \* placing map or reduce on servers
- \* detecting server failure & restarting map/reduce tasks
- \* using progress reports



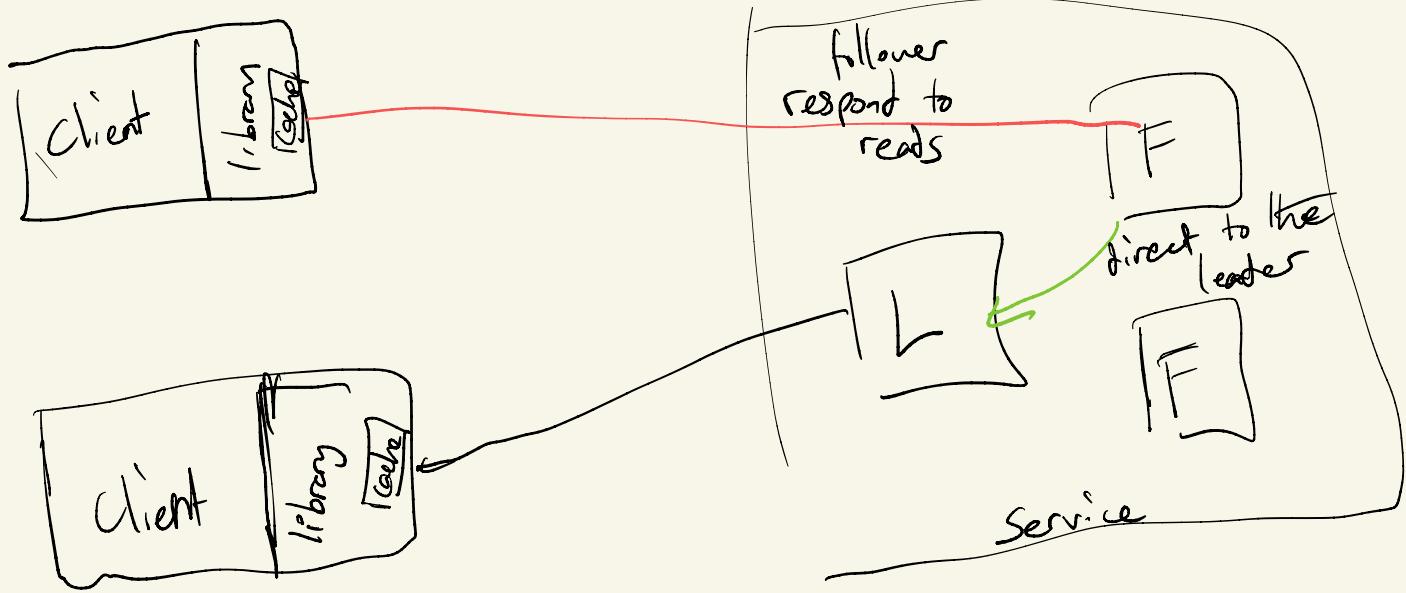
raft = leader election

\* log

Consensus / Coordination Services = primitives to automate  
leader election &  
techniques to maintain  
persistent state (log or KV)



# Read-specific optimizations

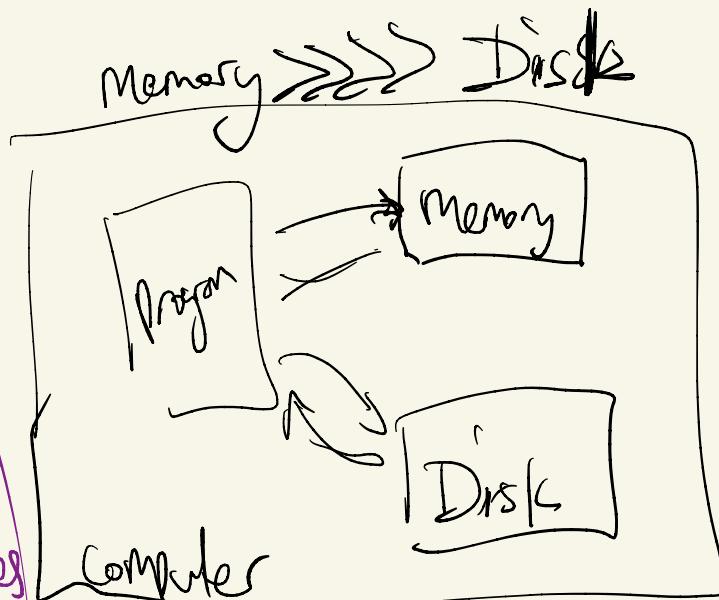


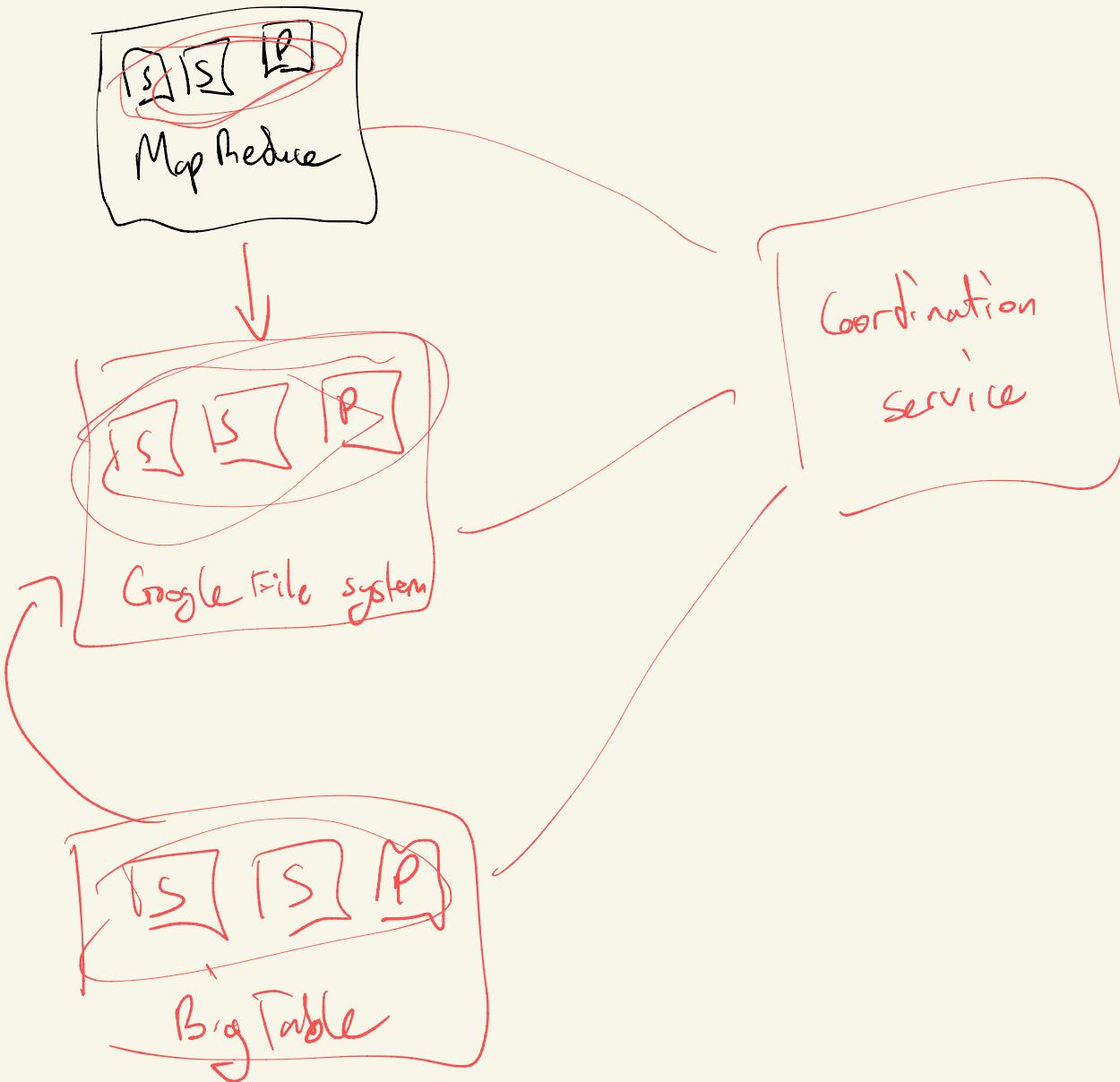
- Techniques
- ① Caching
  - ② have followers respond to read requests
- ↳ Statelessness ↳ when note gets update it sends notification to Cache  
 can get stale bolt information!!!  
 provide a "synch" command to force followers to synchronize with leader

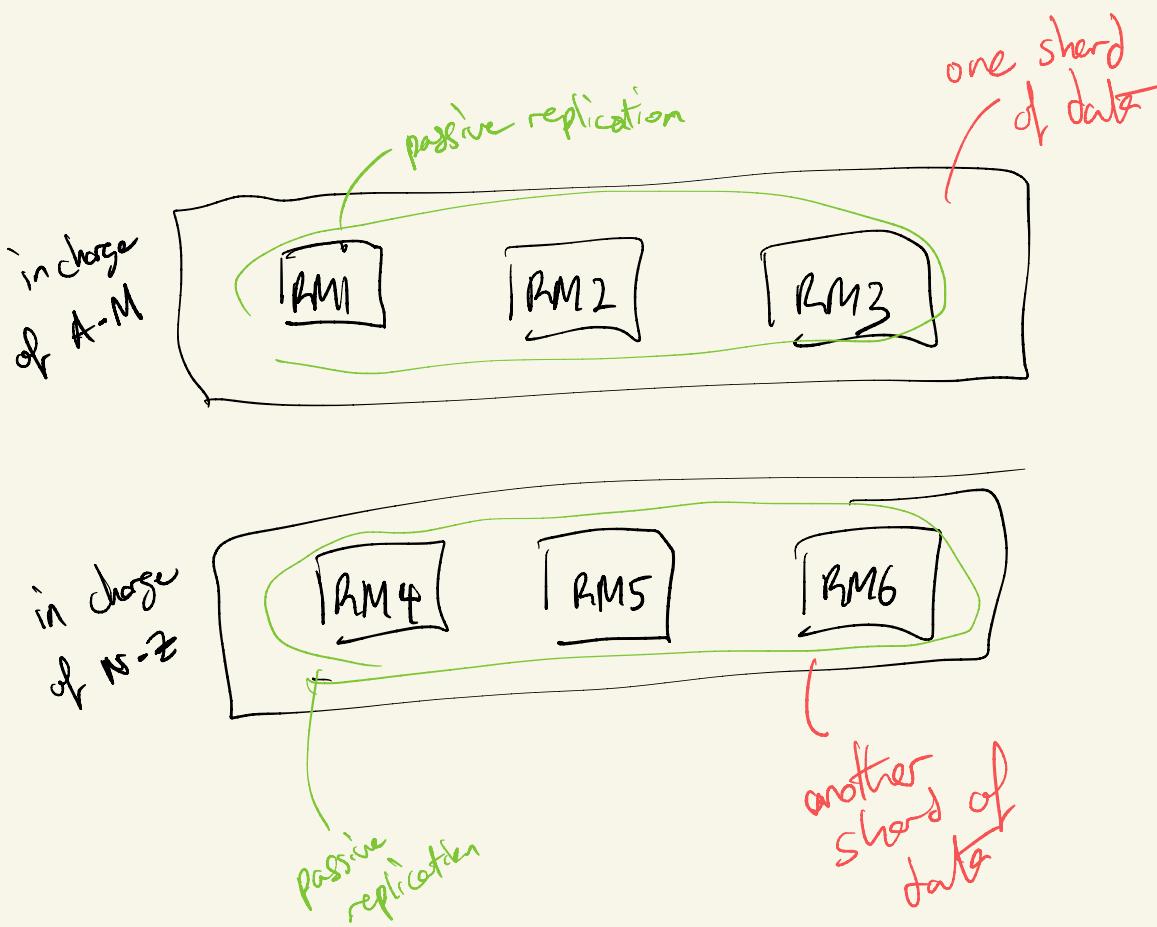
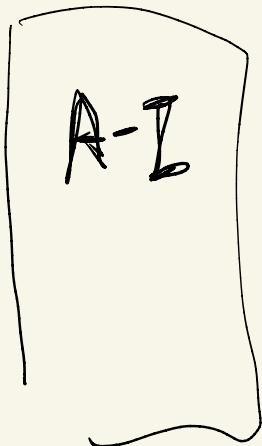
③ Make all data in-memory

④ Leader / followers can respond to read request

only leader processes writes







all replication schemes provide "consistency" guarantee with a shard!

These no guarantees for changes across these shards

# This class

- ① advanced ordering
- ② CAP Theorem
- ③ Zookeeper ( motivation  
& how it scales  
to needs )

