Preview Session
Given a raft cluster with 7 nodes, if 2 nodes fail.

Thus only 5 nodes are currently send/receive heartbeats, when you determine majority for election/log-commitments.

How many nodes are required for a majority?

( ) 1
( ) 3
(X) 4
( ) 7

In your raft clusters, you have 7 nodes, what is the maximum number of simultaneous node failures you can tolerate?

( ) 1
(X) 3
( ) 5
( ) 7
Which of these provides safety (at most one viable leader)?

- Randomized timeouts
- Protocol ensure that each node votes at most once
- Clients retry lost requests
- Followers redirect clients to leaders.

Which of these provides liveness (eventually a leader is elected)?

- Randomized timeouts
- Protocol ensure that each node votes at most once
- Clients retry lost requests
- Followers redirect clients to leaders.
In the gossip update protocol, there are several ways to speed up propagation to all nodes. In a correctly implemented gossip protocol, which of these is the best candidate for improving the time to propagate a message to all nodes.

(X) Increase the number of random nodes being contacted by each node.
( ) Increase the time interval for generating gossip messages
( ) Use a better random algorithm for selecting nodes.
( ) All of the above.

In a cluster designed to use lazy replication, which of these is true about update request messages?

✓) All update requests for a Key must be preceded by at least one query request (for the Key).
✓) Update requests are processed by RMs without a delay.
( ) Updated values may be lost while RMs are offline.
(X) All of the above.

In the gossip update protocol, there are several ways to speed up propagation to all nodes. In a correctly implemented gossip protocol, which of these is the best candidate for improving the time to propagate a message to all nodes.

(X) Increase the number of random nodes being contacted by each node.
( ) Increase the time interval for generating gossip messages
( ) Use a better random algorithm for selecting nodes.
( ) All of the above.
You are evaluating a protocol's guarantees according to the CAP theorem, which of these guarantees is *NOT* possible:

- (X) CAP
- ( ) CA
- ( ) AP
- ( ) CP

You designed a new distributed systems protocol, you observe that whenever there is a network partition the protocol stops working (fails make any progress towards termination). Which of these guarantees does your system provide?

- ( ) CAP
- (X) CA
- ( ) AP
- ( ) CP

You can pick and have at most two:

During partition there is no way to provide both Availability & consistency.
Many systems store data to disks (often in a log). Which of the ACID properties are these systems trying to preserve?

( ) Atomicity
( ) Isolation
( ) Transaction
(X) Durability

Which of the CAP properties does 2PC provide? (hint: consider the scenario where the coordinator is partitioned from the participants)

(X) CA
( ) CP
( ) PA
( ) CAP!
Which of these is an example of meta-data?

(X) File attributes (creation date, size, access control, locks)
( ) File contents (the bytes/bits in the file)
( ) All of the above.
( ) None of the above.

In terms of data versus meta/data split, which of these statements is correct?

✓ Metadata is small, involve minimal client interactions, and stored on a small number of servers 3 servers
✓ Data involves a bulk of the client interactions, involve massive amount of bytes, and are stored on a large number of servers 1000 of servers
(X) All of the above.
( ) None of the above.
Load Balancing (Global/local)
Performance (Mephedone/liteminer)
Time/ordering (Vector clocks/logical clocks/Global snapshots)
Distributed Hash Table (Topestry/Chord/Consistent hashing)
Replication (Active/Passive/Lazy)
Transactions (ACID/2PC/Locks)
Distributed File Systems (meta/data split)
Cassandra/Dynamo
Ordering guarantees for protocols

Raft (passive replication)

Linearizable = Total + FIFO + Real time

Active replication

Sequential = FIFO + Total

Leaders decide order: follows the leader's order
Based on leader's time

Each FE assigns clock to msg
Each sorts the msg by clock & indicates a tie
Lazy replication

Based on vector clocks = causal ordering of events
Raft

leader election = safety / liveness

log safety = when a log committed and at least one entry in current term

majority of nodes

who can be a leader?

node with most up to date log

myTerm > voteRPLterm

reject

myTerm = voteRPLterm

check index

myTerm < voteRPLterm

vote (yes)

leader for term 4

S1

1 2 2 2 4

S2

1 1 3

S3

1 1 2 2

S4

0

S5

1 1 2 2

follower

who is the leader for term 3?

S2

\* S4 has no supporters

\* S5 can become leader

\* S2 can become leader

\* S3 can't become leader (only S4 will vote yes)
Final (L1 - L20)

Out 11:59 pm (EST) ~ 19 April

Due 11:59 pm (EST) ~ 21 April