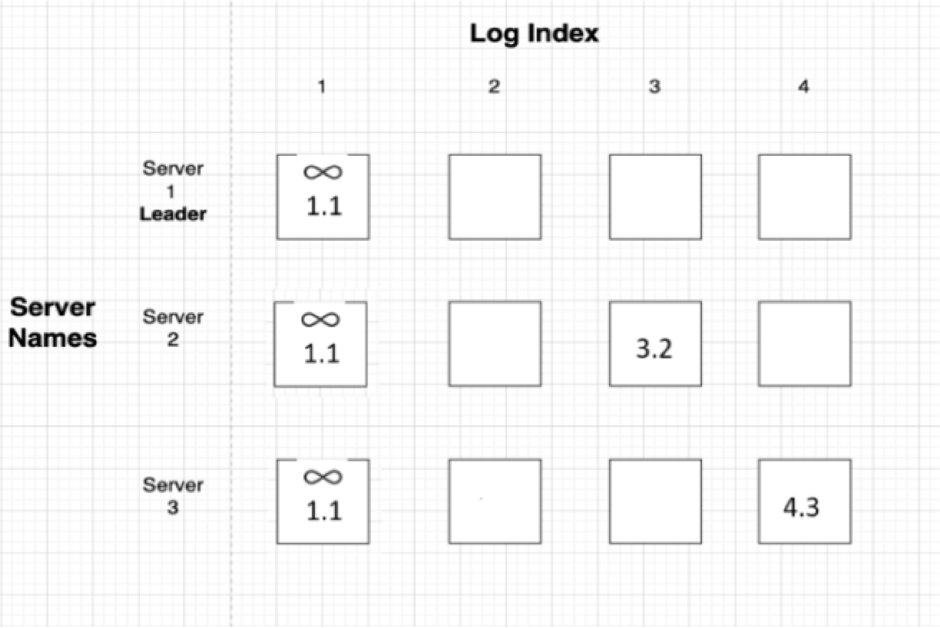
CS582 Distributed Systems

Quiz 6 A Key

Student Name:		**Student ID:**
Time Allowed: 20 mins.	**Total Marks:** 13	**Marks Obtained:**

1. (6 marks) Log Entry Selection and Contention Resolution

   Scenario: Suppose we have a replicated key-value storage service that uses Multi-Paxos to ensure consistency across three servers. The current logs are as follows:



Log Index

Server Names	1	2	3	4
Server 1 Leader	∞ 1.1			
Server 2	∞ 1.1		3.2	
Server 3	∞ 1.1			4.3

The current round number for Server 1 is 5. A new command, `Put (X,1)`, is received for inclusion.

- **Identify which log entry Server 1 would initially choose for Put (X,1) and explain how it determines this index.**
  Server 1 will select index 2 for `Put (X,1)`, as it is the firstUnchosenIndex (the lowest index with no chosen value). Server 1 identifies this by scanning its own log entries for the first empty slot.
- **Describe the Prepare RPC message Server 1 would send to other servers, specifying the proposal number and the index.**
  Server 1 sends Prepare RPC messages with:
  - Proposal Number: 5.1
  - Index: 2

  Server 1 sends this to Servers 2 and 3. Since only the leader can propose, other servers will wait for this proposal before attempting any values at index 2.
- **After receiving all responses, determine the firstUnchosenIndex for Server 1 and state whether the server will skip future prepare RPCs.**
  Upon successful acceptance of `Put (X,1)`, the firstUnchosenIndex for Server 1 would move to index 3. The NoMoreAccepted flag would remain unset. The NoMoreAccepted flag should only be set if Server 1 learns that other servers do not have accepted values for subsequent entries, allowing it to skip proposing for future entries until needed. Since there's no indication that other servers are missing further entries, the NoMoreAccepted flag will not be set, and the proposer cannot skip future Prepare RPCs.
- **If Server 1 successfully logs CMD_A and it gets chosen on majority servers, would it send a Success RPC? Explain why or why not.**
  The Success RPC is sent when the proposer receives an accept reply with an older firstUnchosenIndex than its own. In this case, Server 2 will send 3 and Server 3 will send 3 while the proposer's own firstUnchosenIndex is also 3. Therefore, no success RPC will be sent.

2. (6 marks) Understanding Configuration Changes in Multi-Paxos

   A Multi-Paxos cluster currently has four servers (Servers A, B, C, and D). The quorum requires any three servers for consensus. To handle increased load, a new server, Server E, is scheduled to join the cluster. The cluster uses the log to record configuration changes, with a configuration delay parameter, $\alpha = 2$. This means configuration changes take effect only after two additional log entries are chosen.

   - **Explain how the configuration change for Server E is managed in the Multi-Paxos log with $\alpha = 2$. When will Server E become part of the quorum, and what are the quorum requirements for entries at indexes 5, 6, 7, and 8?**

     The configuration change for Server E with $\alpha = 2$ means that Server E will not become part of the quorum immediately after the change is logged. Instead, two additional log entries must be chosen after the entry recording the configuration change.

     – Index 5: Only the original servers (A, B, C, and D) are part of the quorum. Any three out of these four servers are needed for consensus.

     – Index 6: Since this is the entry recording the configuration change, it also requires any three out of the original four servers (A, B, C, and D).

     – Index 7: This is the first entry after the configuration change, but Server E is not yet included in the quorum because two entries must be chosen after the configuration change.

     – Index 8: After two additional entries (indexes 6 and 7) have been chosen, Server E officially joins the quorum starting with index 8. From this point, any three out of five servers (A, B, C, D, and E) are needed for consensus.

   - **Small α Value: What would be the effect if α were set to a very small value (e.g., $\alpha = 1$)? How might this impact consistency in quorum changes?**

     Setting α to a very small value (e.g., $\alpha = 1$) would mean that configuration changes take effect almost immediately, with just one additional log entry needing to be chosen. This approach reduces concurrency, as each configuration change must be finalized before moving on to subsequent entries.

   - **Large α Value: Describe the impact of setting α to a very large value (e.g., $\alpha = 100$). How would this affect the system's ability to respond to configuration changes?**

     A very large α value, such as 100, would mean that many log entries must be chosen before a configuration change, such as adding a new server, takes effect. This delay in adopting new configurations can reduce the system's responsiveness to changes, as it slows down the integration of new nodes into the quorum.

3. (3 marks) PBFT

   - **In a PBFT network with 7 replicas, the primary sends a pre-prepare message with sequence number 10 for a client request. Backup Replica 3 receives the pre-prepare message but notices that Replica 5 received a different sequence number for the same request. What should Replica 3 do next?**

     Replica 3 should recognize this as Byzantine behavior from the primary (sending inconsistent pre-prepare messages) and initiate a view change. Replica 3 would broadcast a view change request to other replicas, indicating the primary's misbehavior. Each replica then acknowledges this request. The new primary gathers a quorum of $2f + 1$ responses and sends a new-view message with this certificate.

   - **In a PBFT system with 4 replicas ($f = 1$), the primary initiates a request at sequence number 15. After receiving the pre-prepare message, each replica sends a prepare message. How many prepare messages must each replica collect to move to the commit phase?**

     Each replica must collect $2f + 1 = 3$ prepare messages to proceed to the commit phase. This ensures that a majority of honest replicas agree on the ordering before committing.