

Distributed Consensus: Raft, Paxos & Linearizability Cheat Sheet

Leader election safety, log replication invariants, split-vote randomized timeouts, and linearizable reads.

EXECUTIVE SUMMARY

Deep dive into state-machine replication and consensus protocols powering etcd, CockroachDB, TiKV, and Spanner: Raft terms, Election Safety, Log Matching Property, and Lease Read optimizations.

KEY EQUATIONS & INVARIANTS

* Fault Tolerance Quorum Size (Crash-Recovery vs Byzantine):

$$\text{LaTeX: } N_{\{\text{CFT}\}} = 2f + 1, \quad N_{\{\text{BFT}\}} = 3f + 1$$

Note: Tolerating f crash failures requires $2f+1$ nodes (e.g., 3 or 5); Byzantine faults require $3f+1$.

* Raft Timing Invariant:

$$\text{LaTeX: } \text{broadcastTime} \lll \text{electionTimeout} \lll \text{MTBF}$$

Note: Typical values: broadcastTime ~ 0.5-20ms, electionTimeout ~ 150-300ms (randomized).

1. The Five Raft Safety Invariants

Raft decomposes consensus into Leader Election, Log Replication, and Safety. A candidate can only win an election if its log is at least as up-to-date as a majority of the cluster--compared first by `lastLogTerm`, then by `lastLogIndex`.

Invariant | Formal Guarantee

- Election Safety | At most one leader can be elected in a given term T .
- Leader Append-Only | A leader never overwrites or deletes its own entries; it only appends.
- Log Matching | If two logs contain an entry with the same index and term, they are identical up to that index.
- Leader Completeness | If a log entry is committed in term T , it will be present in the leaders of all higher terms $T' > T$.
- State Machine Safety | If a server applies log entry at index i , no other server will ever apply a different entry for index i .

COMMON PRODUCTION & INTERVIEW PITFALLS

- Serving reads directly from the leader's local memory without a `ReadIndex` quorum heartbeat check or bounded clock lease risks returning stale data during a network partition (split-brain leader).

FREQUENTLY ASKED QUESTIONS

Q: Why can't a new Raft leader immediately commit entries from previous terms by counting replicas?

A: Per Figure 8 of the Raft paper, an old-term entry stored on a majority could still be overwritten if a candidate with a higher lastLogTerm wins a subsequent election. Raft leaders only commit old entries indirectly by committing at least one entry from their current term.