Correction FlexRaft, ICPP 2023

"Minimizing Network and Storage Costs for Consensus with Flexible Erasure Coding."

November 12, 2026

We identified a subtle safety issue in the FlexRaft protocol involving partial re-encoding of an entry whose committed status is not yet known to a newly elected leader.

Sepecifically, suppose an entry x is committed in term 1, but the leader fails before propagating the updated *leaderCommit* value to the followers. A later leader may successfully reconstruct x during the LeaderPre phase while still considering it locally uncommitted. If the new leader then re-encodes (x) using a smaller coding parameter (k'), some replicas may overwrite their fragments from the previous encoding before the new encoding has been durably disseminated to (F+k') replicas. If the new leader subsequently fails, it is possible that neither the old encoding nor the new encoding retains enough mutually compatible fragments to reconstruct (x), even though (x) had already been committed.

The issue can be addressed by the following changes:

- **Non-destructive re-encoding.** During a transition from encoding with k to k', the old fragments should remain available until the new encoding has been durably disseminated to enough servers to satisfy the corresponding recovery condition. In other words, the encoding transition should be non-destructive until the new encoding becomes independently safe.
- **Encoding-aware LeaderPre recovery.** During LeaderPre recovery, fragments from different encoding schemes must not be mixed in the decoding process. Compatible fragments should correspond to the same Raft log entry and encoding scheme. Re-encoding an existing entry changes its coding parameter but does not create a new Raft log term.

We apologize for this issue and thank Prof. Dahlia Malkhi for bringing this issue to our attention.