Raft Reconfig Bug (Single Node Change)

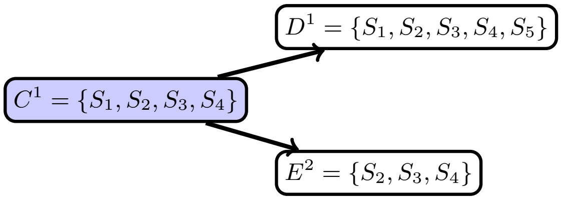
William Schultz

February 5, 2024

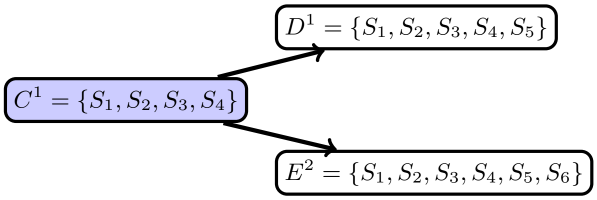
Raft Reconfig Bug Cases

High level overview of the Raft reconfiguration bug cases laid out in Diego's group post. Configs are annotated with their terms i.e., a config X in term t is shown as X^t.

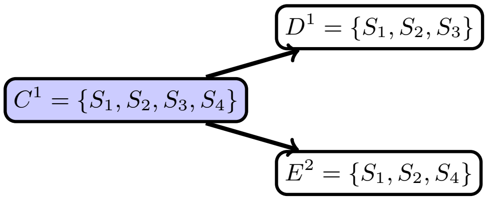
(1) **One add, one remove**:


$D^1 = \{S_1, S_2, S_3, S_4, S_5\}$
$C^1 = \{S_1, S_2, S_3, S_4\}$
$E^2 = \{S_2, S_3, S_4\}$

(2) **Two adds**:


$D^1 = \{S_1, S_2, S_3, S_4, S_5\}$
$C^1 = \{S_1, S_2, S_3, S_4\}$
$E^2 = \{S_1, S_2, S_3, S_4, S_5, S_6\}$

(3) **Two removes**:


$D^1 = \{S_1, S_2, S_3\}$
$C^1 = \{S_1, S_2, S_3, S_4\}$
$E^2 = \{S_1, S_2, S_4\}$

I think all of these bug cases can be viewed as instances of a common problem related to config management when logs diverge (i.e., when there are concurrent primaries in different terms). The bug arises in each case due to the fact that each child config (D and E) has quorum overlap with its parent C (due to the single node change condition), but the sibling configs don't have quorum overlap with each other. These scenarios are problematic because, for example, in case (1), config D could potentially commit writes in term 1 that are not known to leaders in config E in term 2 or higher (since D and E don't have quorum overlap), breaking the fundamental safety property that earlier committed entries are known to newer leaders. Note that this underlying problem should be avoided in joint consensus since in that case each child config will continue to contact a quorum in its parent config.

Proposed Fix

Diego proposes the following fix:

> The solution I'm proposing is exactly like the dissertation describes except that a leader may not append a new configuration entry until it has committed an entry from its current term.

As described above, the underlying bug can be seen as stemming from the fact that when log divergence occurs, even though each child config has quorum overlap with its parent (due to the single node change condition), the sibling configs do not necessarily have quorum overlap with each other.

So, upon election, before doing any reconfiguration, you actually need to be sure that any *sibling* configs are disabled i.e., prevented from committing writes, electing leaders, etc. You achieve this by committing a config in the parent config in your term, which disables all sibling configs in lower terms. Similarly to Diego's fix, we achieve this in MongoDB reconfig by rewriting config term on primary election, which then requires this config in the new term to become committed before further reconfigs can occur on that elected primary.