Raft Vs MongoDB Primary Election

How is the raft consensus algorithm different from MongoDB's primary election process other than the fact that MongoDB takes other factors (priority, for example) into consideration while electing the primary?

mongodb consensus raft

asked Jan 9 '14 at 10:49

Chandra Sekar S
7,558 1 27 43

1 Answer

Some key differences on the consensus approach as at MongoDB 2.4 are:

- Raft uses a *strong leader* model. The leader has responsibility for managing replication and data flows from the leader to other servers. In MongoDB replica sets the secondaries follow the operation log (*oplog*) of an upstream host which can either be the primary or a secondary with a newer oplog.
- Raft only has three node states to consider: leader (primary), follower (secondary), or candidate (nominated primary). MongoDB has additional node states to consider including more potential error states such as `RECOVERING` or `SHUNNED` nodes, or delayed replica set members.
- In Raft each node can only vote for a candidate node once per election term. MongoDB allows votes per node to be adjusted as part of the replica set configuration, so some nodes may be non-voting or possibly have multiple votes (Note: multiple vote configuration has been deprecated as of the MongoDB 2.5 development branch).
- Raft uses a *joint consensus* approach which allows a cluster to continue operating during configuration changes. MongoDB requires a strict majority of voting nodes to elect a new primary; while an election is in progress the replica set has no primary and cannot accept writes.

For more detailed information you should compare the Raft paper In Search of an Understandable Consensus Algorithm with the documentation on MongoDB Replica Set Elections.

edited Jan 8 '16 at 6:47

answered Jan 10 '14 at 15:41

Stennie
35.5k 6 80 110