Jonathan Roland

WEB 335 Introduction to NoSQL

Instructor: Prof. Krasso

12/02/2020

Discussion 8.1

1. **What are replica sets?**

Replica sets are a deliberately created set of servers which act as failovers for a specific MongoDB server. They mirror the data of the central server and can elect from amongst themselves a new primary if the existing primary goes down. They can also be used to restore corrupted documents or even a whole server if needed.

When a replica set is created, one server must be designated as primary. Secondaries are allowed to lag behind the primary and will refuse read and write requests to prevent reading of stale data and desynchronization with the primary.

1. **How does replica set members replicate new data?**

The primary maintains a capped collection in the *local* database named *oplog* which contains every write the primary performs. The secondaries also maintain an oplog and utilize a check every 2 seconds (known as a heartbeat) to stay synchronized with the primary. Every secondary’s goal is to stay synchronized with the primary, and they will communicate with each other to spread the work of synchronizing or electing a new primary in case of a failover incident.

If a secondary is trying to catch up their oplog, they will often check other secondaries for the next version rather than all requesting from the primary constantly. It is possible for a secondary to fall too far behind to catch up to the current oplog. It will then require a manual re-sync. To avoid this, make your oplog size as large as possible, ideally able to record two to three days worth of changes.

1. **How are failovers and rollbacks managed?**

If a primary goes down, the secondaries will negotiate an election for the new primary that will handle incoming read and write requests. There are several considerations that resolve which secondary will be elected primary including ping, network separation, and most importantly which of them is most up to date. Each secondary will evaluate it’s considerations and either submit itself for election or vote for the secondary it views as the best candidate. A majority of votes elects the new primary. The creator of the MongoDB replica set can also create an arbiter to break ties if constraints limit the number of secondaries to an odd number, such as a pair. The arbiter contains the evaluation capabilities of the secondaries without acting as a backup data store.

If a primary goes down and a write is not copied to any secondaries before a new secondary is elected to act as the primary, that issue must be resolved when the original primary comes back online. This resolution is called a rollback.

Rollback resolution is initiated when the original primary comes back online. It will request the latest oplog from the new primary and compare to its own oplog. If it finds a write that did not make it to the secondaries, it will do an editing process to stitch together the write activities of both. Note that in versions of MongoDB prior to 4.0, rollbacks could fail if the size of the synchronization was too large. This is no longer an issue past v4.0.

**Reference:**

Bradshaw, S., Brazil, E., & Chodorow, K. (2019). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage* (3rd ed.). O’Reilly Media.