William Austin

07/11/2023

WEB 335 - Discussion 8.1 - Replica Sets

In most applications where data is being stored, written, deleted, and altered you want to make sure that your data isn’t all in one place, on one database. What would happen if that database crashed, or your application lost connection to that database? You’re app would crash. The way to avoid this is through ‘replica sets’. Replica sets are copies of your database on different databases, at different data centers. They are backups. They are identical to your primary database, and can take over as primary should the primary fail.

When you create your database, you want to create replica sets to protect the integrity of your data. When you do this, all the member sets elect a primary, and the other members become secondary. The primary is the only database that accepts new ‘writes’. Should that primary fail or crash, or lose network connection, the other members would immediately elect a new primary, and that member would then be the only one accepting new ‘writes’. Once, the former primary comes back up, it is now a secondary(Chodorow, 2020). This is a process called failover. Basically, failover is the automatic switching of s secondary to a primary if the primary fails for whatever reason.

Let’s say you have a blog that allows comments from readers. Those comments, as well as your blog posts, are stored in your database. As new comments come in, or as you write new blog posts, more data is added to your database. This data gets written into your primary database and replicated by the secondaries. As the primary database accepts new ‘writes’ it creates something called the ‘oplog’ (short for log of operations). The secondaries query this oplog to see which writes they have and which they need to replicate to be up to date with the primary. “Secondary members replicate data continuously” (Chodorow, 2020). As secondaries ‘write’ the data from the primary they too create their own oplog, this makes it possible for any secondary to query any other secondary to sync up. Meaning every secondary doesn’t have to query the primary, they can query each other as well. Database managers can build hierarchies of which secondaries talk directly to the primary and which secondaries talk only to other secondaries.

Now, a situation can arise where the primary has accepted a new ‘write’, but then crashes before that ‘write’ can be replicated among the secondaries. When the primary crashes, a new primary is elected from among the secondaries. The new primary’s oplog doesn’t have that last ‘write’, but as the new primary it is receiving new ‘writes’ and moves along without that piece of data. So now, when the crashed primary is back up, it queries the oplog to see what it has missed. It will see that it has data that no one else has, and because it is not a primary, it cannot share that data. That server will then begin a process called rollback. “Rollback is used to undo ops that were not replicated before failover”(Chodorow, 2020). It will take those rollbacks and put them in a separate file, that an administrator must manually deal with, either deleting them or adding them to the database.

Sources

Chodorow K. (2017). *MONGODB: The Definitive Guide*. O’REILLY MEDIA, INC, USA.