Investigate the balance between the consistency and the availability in your NoSQL system

The CAP theorem states that a distributed system cannot be simultaneously consistent, available and have partition tolerance.

Consistency happens when all the clients either receive the most recent read, or an error. Availability happens when the client always gets a response when it requests for data (without necessarily having to be consistent). Partition tolerance is when the cluster continues to work no matter the number of messages that are dropped by the network.

In the case of mongoDB, the system resolves partitions and maintains consistency, but it compromises on availability. The client only has one primary node that receives the all of the writes. Every other node (secondary nodes) will just copy it and apply the changes to their own datasets. When the primary node becomes unavailable, a secondary node will take its place and all the other secondary nodes will copy it. Once every secondary node has copied it, the cluster will become available again. During the time that the secondary nodes are copying the new primary node, the system will be unavailable (few seconds).