

Consistencia

Teniendo en cuenta el teorema CAP (Consistency Availability Partition tolerance), Cassandra sacrifica la consistencia para garantizar la disponibilidad (availability) y tolerancia a fallos (partitioning). Esto significa que los datos no se replican de forma síncrona en todos los nodos. En su lugar, se replican de forma asíncrona en los nodos que se le indiquen en la estrategia de replicación. Esto hace que los datos no estén disponibles en todos los nodos al mismo tiempo. Por lo tanto, si se realiza una lectura en un nodo, es posible que no se obtenga el dato más actualizado.

Se puede definir el grado de consistencia que deseamos a la hora de realizar una lectura o escritura. Pero hay que tener en cuenta que cuanto mayor sea el nivel de consistencia exigido peor serán las otras propiedades de Cassandra (disponibilidad y tolerancia a fallos). Es decir, cuanto mayor sea la consistencia requerida menor será el rendimiento. Si exigimos máxima consistencia en escritura el dato habrá que **escribirlo a todos los nodos** lo que será más lento y costoso. Del mismo modo, si exigimos máxima consistencia en la lectura habrá que **consultar a todos los nodos** para asegurarnos de que vamos a obtener el valor más reciente del dato.

Los niveles de consistencia que se pueden especificar son:

- **Any:** Sólo para escrituras. Se garantiza que la escritura se ha realizado en al menos un nodo.
- **One/Two/Three:** En escrituras se garantiza que el dato ha sido replicado en uno/dos/tres nodos. En lecturas se garantiza que se ha leído el dato más actual de uno/dos/tres nodos.
- **Quorum:** En escrituras se garantiza que el dato ha sido replicado en un *quorum* de nodos. En lecturas se garantiza que se ha leído el dato más actual de un *quorum* de nodos.
- **Local quorum:** En escrituras se garantiza que el dato ha sido replicado en el *quorum* del datacenter local. En lecturas se garantiza que se ha leído el dato más actual de un *quorum* del datacenter local.
- **Each quorum:** En escrituras se garantiza que el dato ha sido replicado en un *quorum* de veces de todos los datacenters. En lecturas se garantiza que se ha leído el dato más actual de un *quorum de nodos de todos los datacenters.
- **All:** En escrituras se garantiza que el dato ha sido replicado en todos los nodos. En lecturas se garantiza que se ha leído el dato más actual de todos los nodos.

El quorum se define según la siguiente fórmula:

$$\text{quorum} = (\text{factor de replicación}/2) + 1$$