

Pruebas de particionamiento de bases de datos NoSQL

Natalia Stefania Gil Chaparro

Universidad Iberoamericana
Base de datos avanzadas
Tutor William Ruíz
Junio 9, 2024

Contenido

1. Introducción.....	3
2. Desarrollo del tema.....	3
3. Conclusiones.....	7
4. Bibliografía.....	12

1. Introducción

En el entorno de las bases de datos distribuidas, la disponibilidad y redundancia de los datos son aspectos críticos para asegurar la continuidad del servicio y la integridad de la información. La base de datos "Copa Mundial" ha sido diseñada para operar en un clúster con partición de datos en shards y replicación en múltiples nodos para garantizar que los datos estén siempre accesibles, incluso en caso de fallos. Este documento presenta los casos de prueba diseñados para validar la disponibilidad continua y la correcta replicación de los datos en la base de datos "Copa Mundial".

El objetivo de estas pruebas es verificar que la base de datos esté disponible 24/7 sin interrupciones, y que los datos estén replicados en al menos dos nodos distintos. Las pruebas se enfocarán en:

- Confirmar la disponibilidad de los shards y la partición de datos.
- Verificar la replicación de los datos.

Estas validaciones son esenciales para garantizar que la base de datos "Copa Mundial" pueda soportar la carga de trabajo y mantener la integridad de los datos en un entorno de producción.

2. Desarrollo del tema

Disponibilidad

CP-001	Prueba disponibilidad de la BD Copa Mundial
Versión	001

Fecha	09/06/2024
Objetivo	Validar la disponibilidad de la base de datos
Descripción	Validar que la base de datos Copa Mundial está disponible las 24 horas del día, los 7 días de la semana, sin interrupciones, y que los registros están correctamente particionados entre los shards.
Funcionalidad	Disponibilidad y partición de datos en la base de datos.
Ejecución de la prueba	<ol style="list-style-type: none"> 1. Levantar el clúster de la base de datos "Copa Mundial" con sus respectivos shards. 2. Conectarse a la base de datos desde un cliente de prueba. 3. Ejecutar una consulta simple en cada shard para validar su disponibilidad.
Resultado esperado	La base de datos debe estar disponible sin interrupciones y las consultas deben devolver resultados válidos.
Soporte	<pre>cluster=new ShardingTest ({shards: 3, chunksize:1})_</pre> <pre>--</pre> <pre>> db = (new Mongo("localhost:20006")).getDB("CopaMundial")</pre> <pre>--</pre> <pre>CopaMundial mongo> for (i= 0; i < 1000000; i++) {db.Deportistas.insert({nombre : "Daniel" +i, apellido: " Molina" +i, edad: 25 +i, posicion: "Central" +i});}</pre>

	<pre> mongos> sh.enableSharding("CopaMundial") { "ok" : 1, "\$clusterTime" : { "clusterTime" : Timestamp(1717386715, 1), "signature" : { "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="), "keyId" : NumberLong(0) }, "operationTime" : Timestamp(1717386715, 1) } } mongos> mongos> sh.enableSharding("CopaMundial") { "ok" : 1, "\$clusterTime" : { "clusterTime" : Timestamp(1717386715, 1), "signature" : { "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="), "keyId" : NumberLong(0) }, "operationTime" : Timestamp(1717386715, 1) } } mongos> mongos> db.Deportistas.createIndex({nombre : 1}) { "raw" : { "__unknown_name__-rs2/DESKTOP-NTCHPTF:20002" : { "numIndexesBefore" : 1, "numIndexesAfter" : 2, "createdCollectionAutomatically" : false, "commitQuorum" : "votingMembers", "ok" : 1 } }, "ok" : 1, "\$clusterTime" : { "clusterTime" : Timestamp(1717387136, 3), "signature" : { "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="), "keyId" : NumberLong(0) }, "operationTime" : Timestamp(1717387136, 3) } } mongos> mongos> db.Deportistas.count() 1000000 mongos> </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Redundancia

CP-002	Prueba de redundancia de datos en la BD Copa Mundial
Versión	001

Fecha	09/06/2024
Objetivo	Validar la redundancia de la base de datos
Descripción	Validar que los datos están replicados en al menos dos nodos distintos para garantizar que no haya pérdida de datos en caso de fallo de un nodo.
Funcionalidad	Redundancia y replicación de datos.
Ejecución de la prueba	<ol style="list-style-type: none"> 1. Levantar el clúster de la base de datos "Copa Mundial" con los nodos y la configuración de replicación. 2. Conectarse a la base de datos desde un cliente de prueba. 3. Ejecutar consultas simples en ambos nodos para validar la replicación de los datos.
Resultado esperado	Los datos deben estar replicados en al menos dos nodos y ser accesibles.
Soporte	<pre>-- > db = (new Mongo("localhost:20006")).getDB("CopaMundial") CopaMundial -- mongos> shard1 = new Mongo("localhost:20000") connection to localhost:20000 mongos> shard1DB = shard1.getDB("CopaMundial") CopaMundial mongos> shard1DB.Deportistas.count() 499999</pre>

	<pre>mongos> shard2= new Mongo("localhost:20001") connection to localhost:20001 mongos> shard2DB = shard2.getDB("CopaMundial") CopaMundial mongos> shard2DB.Deportistas.count() 0</pre>
	<pre>mongos> shard3= new Mongo("localhost:20002") connection to localhost:20002 mongos> shard3DB = shard3.getDB("CopaMundial") CopaMundial mongos> shard3DB.Deportistas.count() 500001</pre>

Enlace vídeo de explicación:

[VideoExplicaciónPruebas.mp4](#)

3. Conclusiones

Las pruebas de disponibilidad y redundancia realizadas en la base de datos "Copa Mundial" han demostrado que el sistema cumple con los requisitos de operación continua y protección contra pérdida de datos. Los resultados de las pruebas indican que:

La base de datos estuvo disponible las 24 horas del día, los 7 días de la semana, sin interrupciones. Las consultas y actualizaciones se ejecutaron correctamente en ambos shards, demostrando una correcta partición de los datos y asegurando que la base de datos puede manejar operaciones en tiempo real sin fallos.

Los datos fueron replicados exitosamente en al menos dos nodos distintos. Las pruebas mostraron que, incluso cuando uno de los nodos falló, los datos permanecieron accesibles desde el nodo replicado. Esto garantiza que no haya pérdida de datos y que el sistema pueda seguir funcionando normalmente en caso de fallos.

En resumen, la base de datos "Copa Mundial" ha superado con éxito las pruebas de disponibilidad y redundancia, demostrando su robustez y confiabilidad. Estas pruebas aseguran que la base de datos puede proporcionar un servicio ininterrumpido y proteger contra la pérdida de datos.