

Materia Bases de datos avanzadas

PROFESOR: WILLIAM RUIZ

Universidad Iberoamericana

Ingeniería en Ciencia de Datos

Actividad 2 - Conceptos y
comandos básicos de la replicación en bases de datos
NoSQL

Nelson Fernando Nopssa Castro

ID 100169351

Documento de Requerimientos No Funcionales para la Base de Datos del Torneo de Fórmula 1

1. Introducción

Este documento especifica los criterios de calidad necesarios para asegurar la redundancia y disponibilidad 24x7 de la base de datos del torneo de Fórmula 1 utilizando MongoDB. Estos aspectos son cruciales para garantizar que la información esté siempre accesible y que el sistema sea robusto y confiable.

2. Criterios de Calidad: Redundancia y Disponibilidad 24x7

2.1 Redundancia

2.1.1 Replicación de Datos

Objetivo: Garantizar que los datos estén disponibles en múltiples ubicaciones para protegerse contra fallos de hardware y otros desastres.

Implementación:

Clúster de Réplica: Se ha configurado un clúster de réplica de MongoDB con tres nodos.

Nodo Primario: Acepta todas las operaciones de escritura.

Nodos Secundarios: Replican los datos del nodo primario y pueden aceptar operaciones de lectura, mejorando la disponibilidad y redundancia.

Configuración Geográfica: Los nodos del clúster se pueden distribuir en diferentes centros de datos o regiones geográficas para mayor resiliencia contra desastres locales.

Gestión de Permisos: Cada servidor o nodo maneja los permisos necesarios para acceder a la información de la base de datos y réplica.

2.2 Disponibilidad 24x7

2.2.1 Alta Disponibilidad (HA)

Objetivo: Garantizar que el sistema esté disponible para los usuarios en todo momento, sin interrupciones.

Implementación:

Failover Automático: Se ha configurado el failover automático en el clúster de réplica de MongoDB para que, en caso de fallo del nodo primario, uno de los nodos secundarios sea promovido automáticamente a nodo primario.

Heartbeat Monitoring: Se utiliza el mecanismo de heartbeat de MongoDB para monitorear continuamente el estado de los nodos y detectar fallos rápidamente.

Operaciones Sin Contratiempos: El sistema garantiza que el ingreso, consulta y modificación de la información se realice de forma ágil y sin contratiempos.

3. Conclusión

Implementar los criterios de calidad especificados en este documento garantizará que la base de datos del torneo de Fórmula 1 utilizando MongoDB sea robusta, confiable y esté siempre disponible para los usuarios. La redundancia y alta disponibilidad son componentes clave para asegurar que la información esté protegida contra fallos y accesible en todo momento, proporcionando una experiencia continua y sin interrupciones. Estas medidas asegurarán que la base de datos pueda manejar eficientemente el alto volumen de datos y las demandas operacionales del torneo de Fórmula 1.

Implementación en Práctica

La implementación práctica de estos criterios incluyó la creación de un clúster de réplica con tres nodos, configurando un nodo primario para aceptar operaciones de escritura y nodos secundarios para replicar los datos y aceptar operaciones de lectura. Se configuró el failover automático para asegurar la alta disponibilidad, permitiendo que en caso de fallo del nodo primario, un nodo secundario sea promovido automáticamente a primario. Además, se implementó un monitoreo continuo del estado de los nodos para detectar y manejar fallos rápidamente. Todo esto garantiza que las operaciones de ingreso, consulta y modificación de datos se realicen de manera eficiente y sin interrupciones.

Estas acciones aseguran la redundancia y disponibilidad 24x7 de la base de datos del torneo de Fórmula 1 utilizando MongoDB, cumpliendo con los criterios de calidad necesarios.

Implementación:

Clúster de Réplica: Se ha configurado un clúster de réplica de MongoDB con tres nodos utilizando los comandos siguientes:

```
MiejReplicaSet=new ReplSetTest({name:"MireplicaSet",nodes:3})
```

```
MiejReplicaSet.startSet()
```

```
MiejReplicaSet.initiate()
```

2.2 Disponibilidad 24x7

2.2.1 Alta Disponibilidad (HA)

Objetivo: Garantizar que el sistema esté disponible para los usuarios en todo momento, sin interrupciones.

Implementación:

Failover Automático: Se ha configurado el failover automático en el clúster de réplica de MongoDB para que, en caso de fallo del nodo primario, uno de los nodos secundarios sea promovido automáticamente a nodo primario. Esta configuración se realizó mediante los siguientes comandos:

```
connPrimary= new Mongo("localhost:20000")
```

```
primaryDB=connPrimary.getDB("Formula1")
```

```
primaryDB.isMaster()
```

```
primaryDB.adminCommand({shutdown:1})
```

```
connNewPrimary=new Mongo ("localhost:20001")
```

```
newPrimaryDB=connNewPrimary.getDB("Formula1")
```

```
newPrimaryDB.isMaster()
```