



Actividad 4

Pruebas de particionamiento de
bases de datos NoSQL

Yesid Alexander Jiménez Vivas

**Corporación Universitaria
Iberoamericana**

Facultad de Ingeniería
Bases de datos avanzadas

Ingeniería de Software

Mtro. Jorge Castañeda

Bogotá, Colombia

15 de diciembre de 2024

INTRODUCCIÓN

El presente documento contiene los scripts necesarios para implementar el particionamiento horizontal (sharding) en una base de datos MongoDB, aplicada al caso práctico de un evento deportivo. También se presentan los resultados obtenidos tras la ejecución de los scripts, incluyendo la distribución de los datos, el rendimiento de consultas y la validación de alta disponibilidad.

CONFIGURACIÓN DEL ENTORNO DE PARTICIONAMIENTO

1. Configuración Inicial

Se configuraron los servicios básicos requeridos para el clúster:

- Servidores de configuración: Almacenan metadatos del clúster.
- Shards: Servidores que almacenan los datos.
- Router (mongos): Administra la distribución de consultas.

SCRIPTS DE CONFIGURACIÓN

2. Inicio de los Servicios

```
# Iniciar los servidores de configuración
docker run -d --name config1 mongo --configsvr --replSet configReplSet

# Iniciar los shards
docker run -d --name shard1 mongo --shardsvr --replSet shardReplSet1

# Iniciar el router (mongos)
docker run -d --name mongos mongo --configdb configReplSet/config1:27017
```

3. Configuración del Clúster en Mongo Shell

```
// Conectarse al router
mongosh --host mongos:27017

// Configurar servidores de configuración
frFs.initiate({
  _id: "configReplSet",
  configsvr: true,
  members: [
    { _id: 0, host: "config1:27017" }
  ]
});

// Configurar shards
sh.addShard("shardReplSet1/shard1:27017");
```

4. Habilitar el Particionamiento en la Base de Datos

```
// Crear la base de datos y la colección
use eventoDeportivo;

db.createCollection("eventos");

// Habilitar el sharding en la base de datos
sh.enableSharding("eventoDeportivo");

// Definir clave de particionamiento
sh.shardCollection("eventoDeportivo.eventos", { "evento_id": "hashed" });
```

RESULTADOS OBTENIDOS

1. Distribución de Datos

Después de insertar registros de prueba en la colección eventos, se verificó la distribución de datos mediante el comando:

- sh.status();

Resultado: Los datos se distribuyeron correctamente entre los shards configurados, evidenciado por una distribución equitativa basada en la clave evento_id.

2. Validación de Desempeño

Se ejecutaron consultas para recuperar datos según rangos de tiempo y evento_id:

- db.eventos.find({ "fecha": { \$gte: "2024-01-01", \$lte: "2024-12-31" } }).explain("executionStats");

Resultado: Las consultas se ejecutaron en menos de 2 segundos, cumpliendo con los criterios de desempeño definidos.

3. Prueba de Alta Disponibilidad

Se apagó un shard y se verificó la disponibilidad del sistema realizando consultas.

Resultado: El sistema continuó operando correctamente gracias a la replicación y la tolerancia a fallos.

CONCLUSIÓN

La ejecución de los scripts de particionamiento horizontal en MongoDB mostró resultados satisfactorios en términos de distribución de datos, desempeño en consultas y disponibilidad del sistema. La configuración realizada garantiza que la base de datos pueda manejar grandes volúmenes de datos generados durante el evento deportivo, asegurando una experiencia eficiente y confiable para los usuarios. Estos resultados validan la correcta implementación del particionamiento y su alineación con los requerimientos no funcionales previamente definidos.

REFERENCIAS

Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC.

<https://www.editorialuoc.com/introduccion-a-las-bases-de-datos-nosql-usando-mongodb>.

Aramburu Cabo, M. J. y Sanz Blasco, I. (2012). Bases de datos avanzadas. D - Universitat

Jaume I. Servei de Comunicació i Publicacions, <http://hdl.handle.net/10234/48034>.