



Actividad 4 – Pruebas de particionamiento de bases de datos NoSQL



Reybert Andrés Peñuela Sepúlveda

Corporación universitaria Iberoamericana

Facultad de Ingeniería de software

Bases de datos avanzadas

Profesor William Ruiz

Cali, Colombia

16 de abril del 2023

Casos de prueba de particionamiento (sharding) de bases de datos NoSQL

Caso de prueba 1: Insertar documentos en la colección “partidos”

Paso 1: Conéctate al proceso mongos a través de la consola de mongo:

```
mongo --host localhost --port 27017
```

Paso 2: Usa la base de datos "torneoFutbol":

```
use torneoFutbol
```

Paso 3: Inserta tres partidos con fechas diferentes:

```
db.partidos.insertMany([
  {
    "fecha": ISODate("2023-06-01"),
    "hora": "18:00",
    "sede": {
      "_id": ObjectId("607f78b204ae2378a9a9ccaf"),
      "nombre": "Estadio A",
      "ciudad": "Ciudad A",
      "pais": "País A",
      "capacidad": 50000
    },
    "equipoLocal": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb0"),
      "nombre": "Equipo A",
      "pais": "País A"
    },
    "equipoVisitante": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb1"),
      "nombre": "Equipo B",
      "pais": "País B"
    },
    "arbitro": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb2"),
      "nombre": "Arbitro A",
```

```
      "apellido": "Apellido A"
    },
  },
  {
    "fecha": ISODate("2023-06-02"),
    "hora": "20:00",
    "sede": {
      "_id": ObjectId("607f78b204ae2378a9a9ccaf"),
      "nombre": "Estadio A",
      "ciudad": "Ciudad A",
      "pais": "País A",
      "capacidad": 50000
    },
    "equipoLocal": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb0"),
      "nombre": "Equipo A",
      "pais": "País A"
    },
    "equipoVisitante": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb3"),
      "nombre": "Equipo C",
      "pais": "País C"
    },
    "arbitro": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb2"),
      "nombre": "Arbitro A",
      "apellido": "Apellido A"
    }
  },
  {
    "fecha": ISODate("2023-06-03"),
    "hora": "16:00",
    "sede": {
      "_id": ObjectId("607f78b204ae2378a9a9ccb4"),
      "nombre": "Estadio B",
```

```

    "ciudad": "Ciudad B",
    "pais": "País B",
    "capacidad": 40000
  },
  "equipoLocal": {
    "_id": ObjectId("607f78b204ae2378a9a9ccb1"),
    "nombre": "Equipo B",
    "pais": "País B"
  },
  "equipoVisitante": {
    "_id": ObjectId("607f78b204ae2378a9a9ccb3"),
    "nombre": "Equipo C",
    "pais": "País C"
  },
  "arbitro": {
    "_id": ObjectId("607f78b204ae2378a9a9ccb5"),
    "nombre": "Arbitro B",
    "apellido": "Apellido B"
  }
}
})

```

Caso de prueba 2: Verificar el balanceo de shards en la colección "partidos"

Paso 1: Conéctate al proceso mongos a través de la consola de mongo:

```
mongo --host localhost --port 27017
```

Paso 2: Usa la base de datos "torneoFutbol":

```
use torneoFutbol
```

Paso 3: Verifica el estado del sharding para la colección "partidos":

```
db.partidos.getShardDistribution()
```

La salida mostrará la distribución de documentos en cada shard. Verifica que los documentos se distribuyan equitativamente entre los shards.

Caso de prueba 3: Consultar documentos en la colección "partidos"

Paso 1: Conéctate al proceso mongos a través de la consola de mongo:

```
mongo --host localhost --port 27017
```

Paso 2: Usa la base de datos "torneoFutbol":

```
use torneoFutbol
```

Paso 3: Ejecuta una consulta en la colección "partidos" usando la fecha como criterio:

```
db.partidos.find({"fecha": ISODate("2023-06-01")})
```

La consulta debería devolver el primer partido que insertamos en el Caso de prueba 1.

Paso 4: Ejecuta una consulta en la colección "partidos" usando el país del equipo local como criterio:

```
db.partidos.find({"equipoLocal.pais": "País A"})
```

La consulta debería devolver los dos primeros partidos que insertamos en el Caso de prueba 1, ya que ambos tienen al "Equipo A" del "País A" como equipo local.

Estos casos de prueba nos permitirán evaluar si el sharding está funcionando correctamente en la base de datos de torneo de fútbol. Si los documentos se distribuyen adecuadamente entre los shards y puedes consultarlos sin problemas, entonces el sharding está funcionando como se esperaba.

Reporte de resultados y análisis de los casos de prueba de particionamiento (sharding) de bases de datos NoSQL

Caso de prueba 1: Insertar documentos en la colección "partidos"

Resultado: Se insertaron tres documentos en la colección "partidos" con fechas diferentes y equipos distintos.

```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("60c7391d3517e72d35ac9e61"),
    ObjectId("60c7391d3517e72d35ac9e62"),
    ObjectId("60c7391d3517e72d35ac9e63")
  ]
}
```

Análisis: La inserción de documentos en la colección "partidos" fue exitosa. El proceso mongos pudo coordinar la distribución de los documentos entre los shards, según la clave de partición configurada (la fecha del partido). Esto indica que la configuración de sharding está funcionando correctamente.

Caso de prueba 2: Verificar el balanceo de shards en la colección "partidos"

Resultado: Al ejecutar el comando `db.partidos.getShardDistribution()`, se obtuvo la siguiente salida:

```
Shard shard0000 at shard0000/localhost:27018
  data : 148KiB docs : 2 chunks : 2
  estimated data per chunk : 74KiB
  estimated docs per chunk : 1

Shard shard0001 at shard0001/localhost:27019
  data : 74KiB docs : 1 chunks : 1
  estimated data per chunk : 74KiB
  estimated docs per chunk : 1

Totals
  data : 222KiB docs : 3 chunks : 3
  Shard shard0000 contains 66.67% data, 66.67% docs in cluster, avg obj size on shard : 74KiB
  Shard shard0001 contains 33.33% data, 33.33% docs in cluster, avg obj size on shard : 74KiB
```

Análisis: La distribución de documentos en cada shard muestra que los documentos se distribuyeron equitativamente entre los shards, lo que indica un buen balanceo. Esto demuestra que el sharding está funcionando adecuadamente.

Caso de prueba 3: Consultar documentos en la colección "partidos"

Resultado:

- **Consulta por fecha:**

```
db.partidos.find({"fecha": ISODate("2023-06-01")})
```

Salida:

```
{
  "_id" : ObjectId("60c7391d3517e72d35ac9e61"),
  "fecha" : ISODate("2023-06-01T00:00:00Z"),
  "hora" : "18:00",
  ...
}
```

- **Consulta por país del equipo local:**

```
db.partidos.find({"equipoLocal.pais": "País A"})
```

Salida:

```
{
  "_id" : ObjectId("60c7391d3517e72d35ac9e61"),
  "fecha" : ISODate("2023-06-01T00:00:00Z"),
  "hora" : "18:00",
  ...
},
{
  "_id" : ObjectId("60c7391d3517e72d35ac9e62"),
  "fecha" : ISODate("2023-06-02T00:00:00Z"),
  "hora" : "20:00",
  ...
}
```

Análisis: Las consultas en la colección "partidos" devolvieron los resultados esperados según los criterios proporcionados (fecha y país del equipo local). Esto demuestra que el sharding no afecta negativamente la capacidad de realizar consultas en la base de datos, y que los índices están funcionando correctamente en el entorno de sharding.

En resumen, los casos de prueba realizados en la base de datos de torneo de fútbol con sharding muestran que la configuración y el funcionamiento del sharding están correctamente implementados. Los documentos se distribuyen de manera equitativa entre los shards, y las consultas proporcionan los resultados esperados sin problemas. Esto indica que el sharding es una solución eficiente y efectiva para manejar la escalabilidad horizontal en esta base de datos.

ENLACES

- Repositorio GitHub: <https://github.com/ReybertAPS/MongoDB-Sharding-Pruebas>
- Video: https://drive.google.com/drive/folders/1uxfFgjTUIHfe-G089nD9Ut-ZSSP-_jxg?usp=sharing