Actividad 3 - Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL

Anderson Devia Artunduaga.

IBEROAMERICANA CORPORACION UNIVERSITARIA

Ingeniería de software

Base de datos avanzadas

WILLIAM RUIZ

Diciembre 2023.

Tabla de Contenidos

Video Link: https://youtu.be/tHYSdzctlms

GitHub: https://github.com/Anderdevia/torneoFutbol.git

Requerimientos no funcionales

Vamos a presentar tres requerimientos no funcionales que son a mi parecer adecuados para realizar esta actividad de particionamiento de la base de datos MongoDB.

1. **Rendimiento**: Este requerimiento se refiere a la capacidad del sistema para procesar y responder a las solicitudes en un tiempo adecuado, manteniendo un alto nivel de eficiencia. En el caso del particionamiento de la base de datos MongoDB, se busca que el rendimiento sea óptimo al distribuir los datos entre múltiples nodos o fragmentos, evitando cuellos de botella en el acceso a la información y optimizando las consultas.

2. **Escalabilidad**: La escalabilidad se relaciona con la capacidad del sistema para adaptarse y manejar un aumento en la carga de trabajo o en la cantidad de datos sin comprometer su funcionamiento. En el contexto del particionamiento de MongoDB, es crucial que la base de datos pueda escalar horizontalmente, es decir, agregar más nodos o fragmentos para aumentar la capacidad de almacenamiento y procesamiento a medida que la aplicación crece.

3. **Disponibilidad**: La disponibilidad se refiere a la accesibilidad y la continuidad en el acceso a los datos y servicios del sistema. En el caso del particionamiento de MongoDB, se busca garantizar que la base de datos esté disponible en todo momento, incluso en situaciones de fallos o caídas parciales, mediante estrategias como la replicación de datos, el uso de réplicas y configuraciones adecuadas para mantener la disponibilidad.

Estrategia de Particionamiento

Para implementar esta estrategia de particionamiento, se pueden seguir los siguientes pasos:

Creación del índice: Se recomienda crear un índice en el campo _id de la colección equipos utilizando el comando db.equipos.createIndex({_id: 1}). Esta acción contribuirá a mejorar el rendimiento de las consultas relacionadas con el campo _id.

División de los datos de la colección: Para fragmentar los datos de la colección equipos, se puede utilizar el comando sh.shardCollection('TorneoDeportivo.equipos', {_id:1}). Esta acción

permite la distribución de los datos entre los servidores en el conjunto de réplicas, facilitando así la gestión de la información.

Monitoreo del rendimiento del clúster: Es esencial realizar un seguimiento continuo del rendimiento del clúster. Ajustar los fragmentos en función de las consultas y el tráfico de la base de datos permitirá mantener un equilibrio óptimo y eficiente en la distribución de los datos entre los servidores.

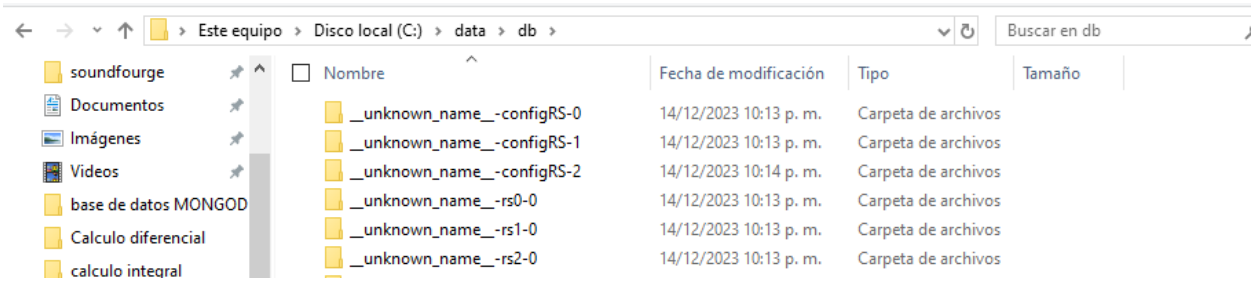
Comandos para realizar el particionamiento horizontal:

Arrancamos la consola de MOGODB

Se ejecuta el siguiente comando:

```
> cluster=new ShardingTest({shards:3, chunksize:1})
```

Podemos verificar que se crean las carpetas de configuración de las instancias en la carpeta \data\db.



Se ingresan datos sobre el balanceador
Se debe iniciar una segunda consola cliente de MongoDB y corremos el siguiente comando contra el balanceador

```
> db = (new Mongo("localhost:20006")).getDB("TorneoDeportivo")
TorneoDeportivo
mongos>
```

Para realizar pruebas de particionamiento vamos a ejecutar el siguiente comando el cual creara un millón de registros

```
> db=(new Mongo("localhost:20006")).getDB("Torneo")
Torneo
mongos> for (i=0; i< 1000000; i++){
... db.equipos.insert({_id: "EQU" +i, nombre: "Equipo numero"+i, ciudad: "ciudad-" +1, año: "2023-"+i});}
WriteResult({ "nInserted" : 1 })
```

Comprobar

```
mongos> db.equipos.count()
1000000
mongos>
```

Tenemos el millón de registros

Ahora vamos a comprobar la distribución de los datos en los nodos se abre otra ventana para ejecutar lo siguiente:

Para conectarnos con el puerto:

```
> shard1 = new Mongo("localhost:20000")
connection to localhost:20000
>
```

Verifico la conexión:

```
> shard1DB = shard1.getDB("TorneoDeportivo")
TorneoDeportivo
>
```

Ahora verifico el número de registros:

```
TorneoDeportivo
> shard1DB.equipos.count()
0
>
```

Realizamos los pasos anteriores, pero con el siguiente puerto:

```
mongos> shard2=new Mongo("localhost:20001")
connection to localhost:20001
mongos> shard2DB = shard2.getDB("TorneoDeportivo")
TorneoDeportivo
mongos> shard1DB.equipos.count()
1000000
mongos>
```

```
> shard3 = new Mongo("localhost:20002")
connection to localhost:20002
> shard3DB = shard3.getDB("TorneoDeportivo")
TorneoDeportivo
> shard3DB.equipos.count()
0
>
```

Ahora no vamos al primer puerto y ejecutamos:

```
mongos> shard1= new Mongo("localhost:20006")
connection to localhost:20006
mongos>
```

Ahora revisamos el estado con el siguiente comando:

```
mongos> sh.status()
```

```
1000000
mongos> shard1= new Mongo("localhost:20006")
connection to localhost:20006
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("657bc3f74c7c7b4ade27b779")
  }
  shards:
        {  "_id" : "__unknown_name__-rs0",  "host" : "__unknown_name__-rs0/DESKTOP-N2RUEH8:20000",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs1",  "host" : "__unknown_name__-rs1/DESKTOP-N2RUEH8:20001",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs2",  "host" : "__unknown_name__-rs2/DESKTOP-N2RUEH8:20002",  "state" : 1 }
  active mongoses:
        "4.4.25" : 1
  autosplit:
        Currently enabled: no
  balancer:
        Currently enabled:  no
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "Torneo",  "primary" : "__unknown_name__-rs0",  "partitioned" : false,  "version" : {  "uuid" : U
"2179955c-eaa0-4c09-948c-afa111c9579f"),  "lastMod" : 1 } }
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos>
```

Ahora procedemos a activarlo con el siguiente comando:

```
mongos> sh.enableSharding("TorneoDeportivo")
```

```
mongos> sh.enableSharding("TorneoDeportivo")
{
        "ok" : 1,
        "operationTime" : Timestamp(1702616168, 5),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1702616168, 5),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Verificamos su estado con el siguiente comando:

```
mongos> sh.status()
```

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("657bc3f74c7c7b4ade27b779")
  }
  shards:
        {  "_id" : "__unknown_name__-rs0",  "host" : "__unknown_name__-rs0/DESKTOP-N2RUEH8:20000",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs1",  "host" : "__unknown_name__-rs1/DESKTOP-N2RUEH8:20001",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs2",  "host" : "__unknown_name__-rs2/DESKTOP-N2RUEH8:20002",  "state" : 1 }
  active mongoses:
        "4.4.25" : 1
  autosplit:
        Currently enabled: no
  balancer:
        Currently enabled:  no
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "Torneo",  "primary" : "__unknown_name__-rs0",  "partitioned" : false,  "version" : {  "uuid" : UU
"2179955c-eaa0-4c09-948c-afa111c9579f"),  "lastMod" : 1 } }
        {  "_id" : "TorneoDeportivo",  "primary" : "__unknown_name__-rs1",  "partitioned" : true,  "version" : {  "uu
 : UUID("aa766c4d-8574-4a17-8b93-ee6932beef9a"),  "lastMod" : 1 } }
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos>
```

Ahora vamos a crear un índice en la colección equipos con la clave _id para esto ejecutamos el siguiente comando:

```
mongos> db.equipos.ensureIndex({_id:1})
```

```
mongos> db.equipos.ensureIndex({_id:1})
{
        "raw" : {
                "__unknown_name__-rs0/DESKTOP-N2RUEH8:20000" : {
                        "numIndexesBefore" : 1,
                        "numIndexesAfter" : 1,
                        "note" : "all indexes already exist",
                        "ok" : 1
                }
        },
        "ok" : 1,
        "operationTime" : Timestamp(1702613509, 12),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1702616511, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Ahora distribuimos los datos de la colección equipos de la base de datos TorneoDeportivo con lo cual nos permite escalar horizontalmente con el siguiente comando:

```
mongos> sh.shardCollection("TorneoDeportivo.equipos", {_id:1})
```

```
mongos> sh.shardCollection("TorneoDeportivo.equipos", {_id:1})
{
        "collectionsharded" : "TorneoDeportivo.equipos",
        "collectionUUID" : UUID("0c18b8c6-0453-44ae-a252-e86f46891c1e"),
        "ok" : 1,
        "operationTime" : Timestamp(1702616914, 11),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1702616914, 11),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Ahora verificamos la grabación de carga cuál es su estado con el siguiente comando:

```
mongos> sh.getBalancerState()
false
mongos>
```

Ahora procedemos a activarlo con el siguiente comando:

```
mongos> sh.setBalancerState(true)
```

```
mongos> sh.setBalancerState(true)
{
        "ok" : 1,
        "operationTime" : Timestamp(1702617282, 3),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1702617282, 3),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Ahora verificamos su estado:

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("657bc3f74c7c7b4ade27b779")
  }
  shards:
        {  "_id" : "__unknown_name__-rs0",  "host" : "__unknown_name__-rs0/DESKTOP-N2RUEH8:20000",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs1",  "host" : "__unknown_name__-rs1/DESKTOP-N2RUEH8:20001",  "state" : 1 }
        {  "_id" : "__unknown_name__-rs2",  "host" : "__unknown_name__-rs2/DESKTOP-N2RUEH8:20002",  "state" : 1 }
  active mongoses:
        "4.4.25" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "Torneo",  "primary" : "__unknown_name__-rs0",  "partitioned" : false,  "version" : {  "uuid" : UUID
"2179955c-eaa0-4c09-948c-afa111c9579f"),  "lastMod" : 1 } }
        {  "_id" : "TorneoDeportivo",  "primary" : "__unknown_name__-rs1",  "partitioned" : true,  "version" : {  "uuid
 : UUID("aa766c4d-8574-4a17-8b93-ee6932beef9a"),  "lastMod" : 1 } }
                TorneoDeportivo.equipos
                        shard key: { "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                __unknown_name__-rs1    1
                        { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : __unknown_name__-rs1 Time
tamp(1, 0)
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos>
```

Y comprobamos que Currently enable: yes
Esta activo

BILIOGRAFIAS

Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC. (Capitulo 9- (*Sharding*))