



Actividad 6 - Pruebas de particionamiento de bases de datos NoSQL

Holman Stiven Cruz Garcia

Corporación Universitaria Iberoamericana

Bases de Datos Avanzadas

Mary Luz Rubiano Acosta

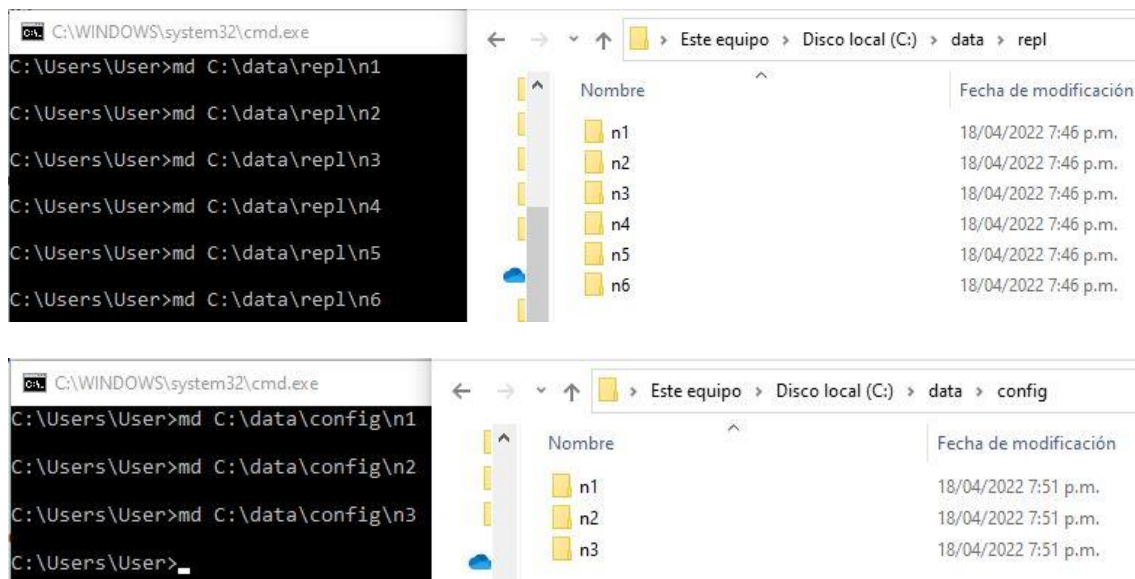
18 de abril de 2022

Pruebas e informe de particionamiento de Bases de Datos NoSQL

Casos de Prueba de Replicación de Bases de Datos NoSQL		
Requisitos	Prueba	Resultado Esperado
Escalamiento horizontal de datos.	Consulta sobre distribución de datos con el estado del clúster usando el comando <code>sh.status()</code>	Muestra la distribución de documentos en los shards implementados.
Fragmentación de bases de datos.	Consulta sobre distribución con el comando <code>getShardDsitribution()</code>	Indica el espacio en memoria asignado a cada shard y el peso que contienen.
La escritura se deben realizar desde un solo servidor.	Se intenta escribir sobre un nodo no primario al azar, luego se comprueba desde el primario.	No es posible escribir desde un nodo secundarios, solo el nodo primario tiene permiso de escritura.
El grupo de partición debe estar compuesto por servidores de configuración, enrutadores de consultas y shards.	Se verifica que los puertos asignados a los servidores de configuración, enrutador y shards, se encuentren activos usando el comando <code>netstat -a</code>	Todos los puertos asignados al conjunto de replicación se encuentran activos.
El sistema debe emplear particionamiento basado en hash.	Con <code>getShardDsitribution()</code> se revisan los pesos de los chunks, se adicionan 3 documentos de forma sucesiva y se verifica posteriormente los pesos de los chunks.	En la segunda prueba debe aumentar los pesos de los chunk uniformemente.

Pruebas de conversión de un replica set en un sharded clúster replicado

Creación de directorios:



Inicio de servidor primario

```
C:\Program Files\MongoDB\Server\4.0\bin>mongod --port 27017 --dbpath \data\repl\n1 --replSet rs0
```

Activación shell servidor primario

```
C:\Program Files\MongoDB\Server\4.0\bin>mongo --port 27017
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("573fb3bd-1691-4248-994d-1aea8584cecf") }
MongoDB server version: 4.0.28
```

Inicio de servidores y activación de shell secundarios

```
C:\Program Files\MongoDB\Server\4.0\bin>mongod --port 27018 --dbpath \data\repl\n2 --replSet rs0
2022-04-18T19:58:10.670-0500 I NETWORK [initandlisten] waiting for connections on port 27018
```

```
C:\Program Files\MongoDB\Server\4.0\bin>mongod --port 27019 --dbpath \data\repl\n3 --replSet rs0
2022-04-18T19:58:49.461-0500 I NETWORK [initandlisten] waiting for connections on port 27019
```

Creación de conjunto de replicas

```
> rs.initiate()
{
  "info2" : "no configuration specified. Using a default configuration for the set",
  "me" : "DESKTOP-FS2OKCO:27018",
  "ok" : 1
}
rs0:SECONDARY> rs.add("192.168.0.18:27019")
{
  "ok" : 1,
  "operationTime" : Timestamp(1650332359, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1650332359, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
rs0:PRIMARY> rs.add("192.168.0.18:27020")
{
  "ok" : 1,
  "operationTime" : Timestamp(1650332376, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1650332376, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
```

Llamado a la base de datos

```
rs0:PRIMARY> use myfootballnine
switched to db myfootballnine
```

Inicio de servidores de configuración

```
C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27022 --dbpath \data\config\n1
--configsvr --bind_ip 192.168.0.18

C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27023 --dbpath \data\config\n2
--configsvr --bind_ip 192.168.0.18

C:\Program Files\MongoDB\Server\4.0\bin>start mongod --port 27024 --dbpath \data\config\n3
--configsvr --bind_ip 192.168.0.18
```

Activación de servidores de configuración

```
2022-04-18T20:43:54.877-0500 I NETWORK [initandlisten] waiting for connections on port 27022
2022-04-18T20:44:27.541-0500 I NETWORK [initandlisten] waiting for connections on port 27023
2022-04-18T20:44:38.126-0500 I NETWORK [initandlisten] waiting for connections on port 27024
```

Inicio enrutador

```
C:\Program Files\MongoDB\Server\4.0\bin>mongos --port 27021 --configdb
rs0/192.168.0.18:27022,192.168.0.18:27023,192.168.0.18:27024
```

Respuesta enrutador

```
C:\Program Files\MongoDB\Server\4.0\bin>mongos --configdb 192.168.0.18:27022,192.168.0.18:27023
,192.168.0.18:27024 --port 27021 --bind_ip 192.168.0.18
FailedToParse: mirrored config server connections are not supported; for config server replica
sets be sure to use the replica set connection string
try 'mongos --help' for more information

C:\Program Files\MongoDB\Server\4.0\bin>mongos --configdb "192.168.0.18:27022","192.168.0.18:27
023","192.168.0.18:27024" --port 27021 --bind_ip 192.168.0.18
FailedToParse: mirrored config server connections are not supported; for config server replica
sets be sure to use the replica set connection string
try 'mongos --help' for more information

C:\Program Files\MongoDB\Server\4.0\bin>mongos --configdb localhost:27022,localhost:27023,local
host:27024 --port 27021 --bind_ip 192.168.0.18
FailedToParse: mirrored config server connections are not supported; for config server replica
sets be sure to use the replica set connection string
try 'mongos --help' for more information
```

Nota

Siguiendo el paso a paso del capítulo VIII Sharding, del libro introducción a las bases de datos nosql usando mongodb, logré llegar solo hasta el script que inicia el enrutador, probé con varios ajustes sobre la sintaxis, sin embargo, obtuve la misma respuesta “las conexiones del servidor de configuración reflejado no son compatibles”.

Conclusiones

- El particionamiento en bases de datos no relacionales permite que las consultas puedan ser gestionadas por rango o por hash.
- La capacidad de los shards puede ser asignada editando los espacios dedicados a cada chunk.
- Al particionar bases de datos se reducen costos en recursos de máquinas, puesto que su escalamiento es horizontal e ilimitado.

- El almacenamiento de datos en particionamiento basado en rango es mejor organizado que el basado en hash, sin embargo, el basado en hash proporciona un mejor balanceo sobre el conjunto de shards.

Repositorio en GitHub:

<https://github.com/HolmanCruz/particionamiento-db-nosql-mongodb>

Video explicativo:

<https://youtu.be/MgpoVpKq36o>

Referencias Bibliográficas

Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Editorial UOC. <https://elibro.net/es/lc/biblioibero/titulos/58524>

MongoDB. (2008). *MongoDB Sharding*. Sharding
<https://www.mongodb.com/docs/v4.0/sharding/>