

Pruebas de particionamiento de bases de datos NoSQL

Adriana L. Giraldo B.

Héctor F. Muñoz M.

Luis E. Salcedo A.

Corporación Universitaria Iberoamericana

Facultad de Ingeniería

Ingeniería de Software

Base de Datos Avanzada

William R. Martínez

Bogotá, 10 de abril de 2023

Pruebas de particionamiento de bases de datos NoSQL

Enlace para el video: <https://youtu.be/SUjrZagc7qA>

Enlace para GitHub:

1. Casos de Prueba.

Id	Caso de Prueba	Descripción	Fecha	Resultado Esperado
0001	Partición	Verificar que las instancias de los Shards sean creadas correctamente.	12/04/2023	Verificar los tres shard se encuentre creados correctamente.
0002	Accesibilidad	Ingresar documentos a las colecciones de Deportistas, Jugadores y Equipos.	12/04/2023	Ingresar 80.000 documentos en la colección de Deportistas, 6 documentos en la colección de Jugadores y 5 documentos en la colección de Equipos.
0003	Validación	Validar en que Shard se están almacenando los documentos ingresados.	12/04/2023	Verificar en cada Shard los documentos de cada una de las colecciones.
0004	Activación	Verificación del estado activo de los shards.	12/04/2023	Verificar que los shards se encuentren activos, si no lo están realizar la activación.
0005	Validación balancer	Validar que el balancer se encuentre en modo TRUE o activo.	12/04/2023	Validar la ejecución de la partición.

2. Ejecución de casos de prueba.

Ejecución caso de prueba # 0001.

Verificar los tres shard se encuentre creados correctamente.

```
> cluster=new ShardingTest ({shards: 3, chunksize:1})
Starting new replica set __unknown_name__-rs0
ReplSetTest starting set
ReplSetTest n is : 0
{
    "useHostName" : true,
    "oplogSize" : 16,
    "keyFile" : undefined,
    "port" : 20000,
    "replicaSet" : "__unknown_name__-rs0",
    "dbpath" : "$set-$node",
    "useHostname" : true,
    "shardsvr" : "",
    "pathOpts" : {
        "testName" : "__unknown_name__",
        "shard" : 0,
        "node" : 0,
        "set" : "__unknown_name__-rs0"
    },
    "setParameter" : {
```

```
81244//1, t: 1 }
s20006| 2023-04-11T15:26:16.513-0500 D1 NETWORK [shard-registry-reload] Started targeter for __unknown_name__-rs0/DESKTOP-KEHD2QP:20000
s20006| 2023-04-11T15:26:16.513-0500 D1 NETWORK [shard-registry-reload] Started targeter for __unknown_name__-rs1/DESKTOP-KEHD2QP:20001
s20006| 2023-04-11T15:26:16.513-0500 D1 NETWORK [shard-registry-reload] Started targeter for __unknown_name__-rs2/DESKTOP-KEHD2QP:20002
s20006| 2023-04-11T15:26:16.849-0500 D1 NETWORK [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_name__-configRS took 1ms
s20006| 2023-04-11T15:26:16.964-0500 D1 TRACKING [rep1SetDistLockPinger] Cmd: NotSet, TrackingId: 6435c268200749423c25d744
s20006| 2023-04-11T15:26:17.179-0500 D1 TRACKING [UserCacheInvalidator] Cmd: NotSet, TrackingId: 6435c269200749423c25d746
s20006| 2023-04-11T15:26:17.302-0500 D1 TRACKING [Uptime-reporter] Cmd: NotSet, TrackingId: 6435c269200749423c25d748
s20006| 2023-04-11T15:26:17.875-0500 D1 NETWORK [ReplicaSetMonitor-TaskExecutor] Refreshing replica set __unknown_name__-rs0 took 0ms
d20001| 2023-04-11T15:26:17.884-0500 I CONNPOOL [ShardRegistry] Ending idle connection to host DESKTOP-KEHD2QP:20005 because the pool meets co
```

 __unknown_name__-configRS-0	11/04/2023 3:26 p. m.	Carpeta de archivos
 __unknown_name__-configRS-1	11/04/2023 3:26 p. m.	Carpeta de archivos
 __unknown_name__-configRS-2	11/04/2023 3:26 p. m.	Carpeta de archivos
 __unknown_name__-rs0-0	11/04/2023 3:26 p. m.	Carpeta de archivos
 __unknown_name__-rs1-0	11/04/2023 3:26 p. m.	Carpeta de archivos
 __unknown_name__-rs2-0	11/04/2023 3:26 p. m.	Carpeta de archivos

Ejecución caso de prueba # 0002

Ingresar 80.000 documentos en la colección de Deportistas, 6 documentos en la colección de Jugadores y 5 documentos en la colección de Equipos.

Deportistas:

```
mongos> for (i= 0; i < 80000; i++) {  
... db.Deportistas.insert({author : "author" +i, post_title : "Futbolistas de alto rendimiento "  
... +i, date: new Date() });  
... }  
WriteResult({ "nInserted" : 1 })  
mongos> -
```

Jugadores:

```
mongos> db.Jugadores.insertMany([  
... {id:1, nombre: 'Macnelly Torres', equipo: 'A.Nacional', posicion: 'Delantero'},  
... {id:2, nombre: 'Jason Zapata', equipo: 'D. Pasto', posicion: 'Medio Campo'},  
... {id:3, nombre: 'Antonio Belez', equipo: 'A. de Cali', posicion: 'Defensa'},  
... {id:4, nombre: 'Mario Benitez', equipo: 'D. Tolima', posicion: 'Portero'},  
... {id:5, nombre: 'Camilo Vargas', equipo: 'Junior', posicion: 'Lateral'},  
... {id:6, nombre: 'Luis Diaz', equipo: 'Millonarios', posicion: 'Delantero'}  
... ]);  
{  
    "acknowledged" : true,  
    "insertedIds" : [  
        ObjectId("6436b89aca6ccec077ad053d"),  
        ObjectId("6436b89aca6ccec077ad053e"),  
        ObjectId("6436b89aca6ccec077ad053f"),  
        ObjectId("6436b89aca6ccec077ad0540"),  
        ObjectId("6436b89aca6ccec077ad0541"),  
        ObjectId("6436b89aca6ccec077ad0542")  
    ]  
}
```

Equipos:

```
}
```

```
mongos> db.Equipos.insertMany([  
... {id:1, nombre: 'D. Cali', titulos: '12'},  
... {id:2, nombre: 'A. Nacional', titulos: '22'},  
... {id:3, nombre: 'A. de Cali', titulos: '9'},  
... {id:4, nombre: 'Junior', titulos: '8'},  
... {id:5, nombre: 'Millonarios', titulos: '12'}  
... ]);  
{  
    "acknowledged" : true,  
    "insertedIds" : [  
        ObjectId("6436190977d705d2e652e9f7"),  
        ObjectId("6436190977d705d2e652e9f8"),  
        ObjectId("6436190977d705d2e652e9f9"),  
        ObjectId("6436190977d705d2e652e9fa"),  
        ObjectId("6436190977d705d2e652e9fb")  
    ]  
}
```

```
mongos> -
```

Ejecución caso de prueba # 0003

Verificar en cada Shard los documentos de cada una de las colecciones

Deportistas:

```
> shard1 = new Mongo("DESKTOP-KEHD2QP:20000")
connection to DESKTOP-KEHD2QP:20000
> shard1DB = shard1.getDB("eventoDeportivo")
eventoDeportivo
> shard1DB.Deportistas.count()
0
> shard2 = new Mongo("DESKTOP-KEHD2QP:20001")
connection to DESKTOP-KEHD2QP:20001
> shard2DB = shard2.getDB("eventoDeportivo")
eventoDeportivo
> shard2DB.Deportistas.count()
80000
> shard3 = new Mongo("DESKTOP-KEHD2QP:20002")
connection to DESKTOP-KEHD2QP:20002
> shard3DB = shard3.getDB("eventoDeportivo")
eventoDeportivo
> shard3DB.Deportistas.count()
0
```

Jugadores:

```
> shard1 = new Mongo("DESKTOP-KEHD2QP:20000")
connection to DESKTOP-KEHD2QP:20000
> shard1DB = shard1.getDB("eventoDeportivo")
eventoDeportivo
> shard1DB.Jugadores.count()
0
> shard2 = new Mongo("DESKTOP-KEHD2QP:20001")
connection to DESKTOP-KEHD2QP:20001
> shard2DB = shard2.getDB("eventoDeportivo")
eventoDeportivo
> shard2DB.Jugadores.count()
6
> shard3 = new Mongo("DESKTOP-KEHD2QP:20002")
connection to DESKTOP-KEHD2QP:20002
> shard3DB = shard3.getDB("eventoDeportivo")
eventoDeportivo
> shard3DB.Jugadores.count()
0
```

Equipos:

```
> shard1 = new Mongo("DESKTOP-KEHD2QP:20000")
connection to DESKTOP-KEHD2QP:20000
> shard1DB = shard1.getDB("eventoDeportivo")
eventoDeportivo
> shard1DB.Equipos.count()
0
> shard2 = new Mongo("DESKTOP-KEHD2QP:20001")
connection to DESKTOP-KEHD2QP:20001
> shard2DB = shard2.getDB("eventoDeportivo")
eventoDeportivo
> shard2DB.Equipos.count()
5
> shard3 = new Mongo("DESKTOP-KEHD2QP:20002")
connection to DESKTOP-KEHD2QP:20002
> shard3DB = shard3.getDB("eventoDeportivo")
eventoDeportivo
> shard3DB.Equipos.count()
0
.
```

Ejecución caso de prueba # 0004

Verificar que los shards se encuentren activos, si no lo están realizar la activación.

```
mongos> shard1 = new Mongo("DESKTOP-KEHD2QP:20006")
connection to DESKTOP-KEHD2QP:20006
mongos> sh.status()
--- Sharding Status ---
sharding version: {
    "id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("6436b634a46f7e89a0f6e962")
}
shards:
{ "_id" : "__unknown_name__-rs0", "host" : "__unknown_name__-rs0/DESKTOP-KEHD2QP:20000", "state" : 1 }
{ "_id" : "__unknown_name__-rs1", "host" : "__unknown_name__-rs1/DESKTOP-KEHD2QP:20001", "state" : 1 }
{ "_id" : "__unknown_name__-rs2", "host" : "__unknown_name__-rs2/DESKTOP-KEHD2QP:20002", "state" : 1 }
active mongoses:
"4.2.23-rc0" : 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" : "config", "primary" : "config", "partitioned" : true }
    config.system.sessions
        shard key: { "_id" : 1 }
        unique: false
        balancing: true
        chunks:
            { "__unknown_name__-rs0" : 1
                { "_id" : { "$minKey" : 1 } } --> { "_id" : { "$maxKey" : 1 } } on : __unknown_name__-rs0
{ "_id" : "eventoDeportivo", "primary" : "__unknown_name__-rs1", "partitioned" : false, "version" : {
eb32e62c2"}, "lastMod" : 1 } }
```

```
mongos> sh.setBalancerState(true)
{
    "ok" : 1,
    "operationTime" : Timestamp(1681313731, 3),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1681313731, 3),
        "signature" : {
            "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAA="),
            "keyId" : NumberLong(0)
        }
    }
}
```

Ejecución caso de prueba # 0005

Validar la ejecución de la partición.

```
active mongoses:
    "4.2.23-rc0" : 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" : "Biblioteca", "primary" : "__unknown_name__-rs2", "partitioned" : true, "version" : { "uuid" : "ff2"), "lastMod" : 1 } }
    { "_id" : "config", "primary" : "config", "partitioned" : true }
        config.system.sessions
            shard key: { "_id" : 1 }
            unique: false
                balancing: true
                chunks:
                    __unknown_name__-rs0 1
                    { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : __unknown_name__-rs0 Ti
    { "_id" : "eventoDeportivo", "primary" : "__unknown_name__-rs1", "partitioned" : true, "version" : { "u
b32e62c2"), "lastMod" : 1 } }
        eventoDeportivo.Deportistas
            shard key: { "author" : 1 }
            unique: false
            balancing: true
            chunks:
                __unknown_name__-rs1 1
                { "author" : { "$minKey" : 1 } } -->> { "author" : { "$maxKey" : 1 } } on : __unknown_name__-
```