



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL



Presentado por:

Carlos Andrés Poveda

Diciembre del 2023

FACULTAD DE INGENIERIA DE SOFTWARE

Base de datos avanzada

Docente

William Ruiz



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL



Descripción de la actividad

▼ Detalles

Estimados estudiantes, Bienvenidos a la última actividad del curso Bases de datos Avanzadas.

La siguiente actividad pretende continuar apropiando de manera práctica los conceptos de particionamiento de bases de datos NoSQL a través de la continuación del caso práctico planteado en la actividades anteriores.

Para el desarrollo de la actividad se debe tener en cuenta lo siguiente:

1. La actividad se podrá realizar en **equipos de máximo 3 estudiantes**.
2. Explorar los recursos de conocimiento para el desarrollo de la actividad y los recursos de apoyo sugeridos. Se aconseja que realice las siguientes lecturas para complementar la información de la actividad: **(Clic para abrir la bibliografía de la unidad.)**
3. Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Editorial UOC.
1. Especificar a través de un documento, los casos de pruebas para validar que el particionamiento planteado en la actividad 3 funcione acorde al documento de requerimientos no funcionales.
2. Ejecutar los casos de prueba planteados y generar un reporte de los resultados y análisis **(Sugerencia apoyarse en pantallazos como evidencia)**.
3. Desarrollar un video de **(máximo 5 minutos)**, explicando los casos de prueba planteados y los resultados obtenidos.
4. Subir la actividad en el enlace dispuesto para la entrega de la tarea, indicando el link del repositorio Git y del video u otra opción es cargar los entregables en un repositorio creado en **OneDrive**.

Entregable

Un documento PDF con el enlace a los siguientes entregables que estará alojado en el repositorio GitHub o GitLab que ustedes ya deben tener creado u otra opción es cargar los entregables en un repositorio creado en **OneDrive**.

1. Documento con la descripción del escenario en que se necesitaría particionamiento y la especificación de los requerimientos no funcionales para dicho escenario.
2. Documento con los casos de pruebas para el particionamiento y los resultados obtenidos **(Sugerencia apoyarse en pantallazos como evidencia)**.
3. **Video(máximo de 5 minutos)**, explicando el proyecto, los casos de prueba realizados y los resultados obtenidos. El video debe ser claro y preciso, adicionalmente deben participar todos los integrantes del equipo.

Link de video : <https://youtu.be/v2JBfyVNFgg>

<https://github.com/CPOVEDAR/Actividad-4-Pruebas-de-particionamiento-de-bases-de-datos-NoSQL.git>

1 : Casos de prueba (Tener en cuenta que en estos casos de prueba de aplican el comando sharding)

Antes de proceder con la fragmentación de una colección en nuestro sistema de gestión de base de dato (Taller) del torneo deportivo, es necesario asegurarse de que la configuración y la implementación del particionamiento horizontal estén listas y operativas. Esto implica activar la funcionalidad de fragmentación tanto para la base de datos general del torneo como para la colección específica que hemos seleccionado para fragmentar. Es decir, se deben habilitar previamente las condiciones necesarias para que el sistema pueda llevar a cabo la tarea de fragmentación de datos de manera efectiva.



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

Después de este paso, será indispensable generar un índice para la clave de fragmentación. Esto se realiza con el propósito de fragmentar la colección y establecer la clave de fragmento, lo que permite la distribución de documentos entre los fragmentos que hemos configurado.

```
mongos> use Taller
switched to db Taller
mongos> db.arbitros.createIndex({number:1})

{ "ok" : 1 }
```

Ahora estamos listos para iniciar el proceso de fragmentación en nuestra colección "árbitros" dentro de la base de datos, asegurándonos de incluir siempre la clave de fragmento que hemos definido previamente.

```
mongos>
mongos> sh.shardCollection("Taller.arbitros",{ "number":1})

{"collectionsharded":"Taller.arbitros","ok" : 1}
```

El sistema habrá distribuido los documentos y podemos verificar la correcta ejecución del proceso de particionamiento en los datos actuales de nuestra colección "arbitros". En primer lugar, observaremos el equilibrio entre nuestras réplicas, nodos y fragmentos. Es así que:



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

```

mongos>
mongos> {
...   "raw" : {
...     "rs0/localhost:27017,127.0.0.1:27027,127.0.0.1:27037" : {
...       "db" : "Taller",
...       "collections" : 1,
...       "views" : 0,
...       "objects" : 640545,
...       "avgObjSize" : 70.83200339949052,
...       "dataSize" : 45370913,
...       "storageSize" : 50438144,
...       "numExtents" : 0,
...       "indexes" : 2,
...       "indexSize" : 54502272,
...       "ok" : 1,
...       "$gleStats" : {
...         "electionId" : ObjectId("41c783e7f5563ceee5727cfa")
...       }
...     },
...     "rs1/localhost:27018,localhost:27028,localhost:27038" : {
...       "db" : "TorneoDeportivoFutsal",
...       "collections" : 1,
...       "views" : 0,
...       "objects" : 359455,
...       "avgObjSize" : 80.73259935179947,
...       "dataSize" : 25461132,
...       "storageSize" : 6699272,
...       "numExtents" : 0,
...       "indexes" : 2,
...       "indexSize" : 6151040,
...       "ok" : 1,
...       "$gleStats" : {
...         "electionId" : ObjectId("6054763033c9f0547630c562")
...       }
...     }
...   },
...   "objects" : 1000000,
...   "avgObjSize" : 70,

```

Ahora explicaremos la distribución equitativa de nuestra colección utilizando otro comando, con el objetivo de comprender los diversos estados en los que se encuentra nuestra fragmentación. De esta manera, se evidencia que:



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

```

mongos>
mongos> --- Sharding Status ---
... sharding version: {
...   "_id" : 1,
...   "minCompatibleVersion" : 5,
...   "currentVersion" : 6,
...   "clusterId" : ObjectId("56f3563ad856566458254e93")
... }
... shards:
...   { "_id" : "rs0", "host" : "rs0/localhost:27017,127.0.0.1:27027,127.0.0.1:27037", "state" : 1 }
...   { "_id" : "rs1", "host" : "rs1/localhost:27018,localhost:27028,localhost:27038", "state" : 1 }
... active mongoses:
...   "4.2.6" : 1
... autosplit:
...   Currently enabled: yes
... balancer:
...   Currently enabled: yes
...   Currently running: yes
...   Collections with active migrations:
...     test.test_collection started at Mon Dic 16 2023 15:16:45 GMT-0500
... Failed balancer rounds in last 5 attempts: 0
... Migration Results for the last 24 hours:
...   1 : Success
... databases:
...   { "_id" : "TorneoDeportivoFutsal", "primary" : "rs0", "partitioned" : true }
...     TorneoDeportivoFutsal.arbitros
...       shard key: { "number" : 1 }
...       unique: false
...       balancing: true
...       chunks:
...         rs0    5
...         rs1    1
...       { "number" : { "$minKey" : 1 } } --> { "number" : 1195 } on : rs1 Timestamp(2, 0)
...       { "number" : 1195 } --> { "number" : 2394 } on : rs0 Timestamp(2, 1)
...       { "number" : 2394 } --> { "number" : 3596 } on : rs0 Timestamp(1, 0)
...       { "number" : 9588 } --> { "number" : { "$maxKey" : 1 } } on : rs0 Timestamp(1, 2)

```



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

2. Casos de prueba (Tener en cuenta que en estos casos de prueba de aplican el comando sharding)

Antes de proceder con la fragmentación de una colección en nuestro sistema de gestión de base de dato (Taller) del torneo deportivo, es necesario asegurarse de que la configuración y la implementación del particionamiento horizontal estén listas y operativas. Esto implica activar la funcionalidad de fragmentación tanto para la base de datos general del torneo como para la colección específica que hemos seleccionado para fragmentar. Es decir, se deben habilitar previamente las condiciones necesarias para que el sistema pueda llevar a cabo la tarea de fragmentación de datos de manera efectiva.

```
Administrador: Símbolo del sistema - mongo --port 27021 --host localhost
mongos> sh.enableSharding("Taller")
{ "ok" : 1 }
```

Después de este paso, será indispensable generar un índice para la clave de fragmentación. Esto se realiza con el propósito de fragmentar la colección y establecer la clave de fragmento, lo que permite la distribución de documentos entre los fragmentos que hemos configurado.

```
mongos> use Taller
switched to db Taller
mongos> db.arbitros.createIndex({number:1})
{ "ok" : 1 }
```



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

Ahora estamos listos para iniciar el proceso de fragmentación en nuestra colección "árbitros" dentro de la base de datos, asegurándonos de incluir siempre la clave de fragmento que hemos definido previamente.

```
mongos>  
mongos> sh.shardCollection("Taller.arbitros",{"number":1})  
  
{"collectionsharded":"Taller.arbitros","ok" : 1}
```

El sistema habrá distribuido los documentos y podemos verificar la correcta ejecución del proceso de particionamiento en los datos actuales de nuestra colección "árbitros". En primer lugar, observaremos el equilibrio entre nuestras réplicas, nodos y fragmentos. Es así que:



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

```

mongos>
mongos> {
...   "raw" : {
...     "rs0/localhost:27017,127.0.0.1:27027,127.0.0.1:27037" : {
...       "db" : "Taller",
...       "collections" : 1,
...       "views" : 0,
...       "objects" : 640545,
...       "avgObjSize" : 70.83200339949052,
...       "dataSize" : 45370913,
...       "storageSize" : 50438144,
...       "numExtents" : 0,
...       "indexes" : 2,
...       "indexSize" : 54502272,
...       "ok" : 1,
...       "$gleStats" : {
...         "electionId" : ObjectId("41c783e7f5563ceee5727cfa")
...       }
...     },
...     "rs1/localhost:27018,localhost:27028,localhost:27038" : {
...       "db" : "TorneoDeportivoFutsal",
...       "collections" : 1,
...       "views" : 0,
...       "objects" : 359455,
...       "avgObjSize" : 80.73259935179947,
...       "dataSize" : 25461132,
...       "storageSize" : 6699272,
...       "numExtents" : 0,
...       "indexes" : 2,
...       "indexSize" : 6151040,
...       "ok" : 1,
...       "$gleStats" : {
...         "electionId" : ObjectId("6054763033c9f0547630c562")
...       }
...     }
...   },
...   "objects" : 1000000,
...   "avgObjSize" : 70,

```

Ahora explicaremos la distribución equitativa de nuestra colección utilizando otro comando, con el objetivo de comprender los diversos estados en los que se encuentra nuestra fragmentación. De esta manera, se evidencia que:



Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

```

mongos>
mongos> --- Sharding Status ---
... sharding version: {
...   "_id" : 1,
...   "minCompatibleVersion" : 5,
...   "currentVersion" : 6,
...   "clusterId" : ObjectId("56f3563ad856566458254e93")
... }
... shards:
...   { "_id" : "rs0", "host" : "rs0/localhost:27017,127.0.0.1:27027,127.0.0.1:27037", "state" : 1 }
...   { "_id" : "rs1", "host" : "rs1/localhost:27018,localhost:27028,localhost:27038", "state" : 1 }
... active mongoses:
...   "4.2.6" : 1
... autosplit:
...   Currently enabled: yes
... balancer:
...   Currently enabled: yes
...   Currently running: yes
...   Collections with active migrations:
...     test.test_collection started at Mon Dic 16 2023 15:16:45 GMT-0500
... Failed balancer rounds in last 5 attempts: 0
... Migration Results for the last 24 hours:
...   1 : Success
... databases:
...   { "_id" : "TorneoDeportivoFutsal", "primary" : "rs0", "partitioned" : true }
...     TorneoDeportivoFutsal.arbitros
...       shard key: { "number" : 1 }
...       unique: false
...       balancing: true
...       chunks:
...         rs0    5
...         rs1    1
...       { "number" : { "$minKey" : 1 } } --> { "number" : 1195 } on : rs1 Timestamp(2, 0)
...       { "number" : 1195 } --> { "number" : 2394 } on : rs0 Timestamp(2, 1)
...       { "number" : 2394 } --> { "number" : 3596 } on : rs0 Timestamp(1, 0)
...       { "number" : 9588 } --> { "number" : { "$maxKey" : 1 } } on : rs0 Timestamp(1, 2)

```