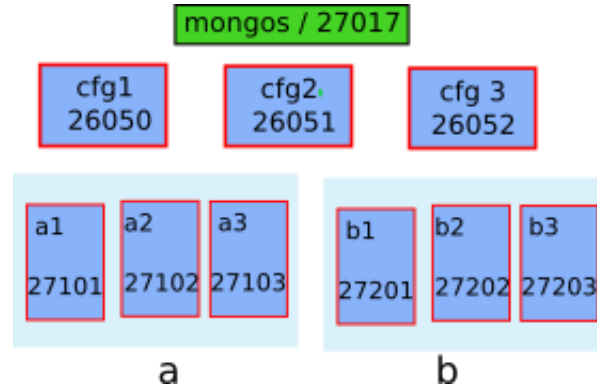


Práctica 9 - Particiones en MongoDB

Nuestro objetivo es crear un clúster en modo local con:

- 3 servidores de configuración con nombres `cfg1`, `cfg2`, `cfg3`
- un cliente `mongos` en el puerto por defecto (27017)
- 2 shards (particiones), `a` y `b`, cada una un conjunto réplica formado por 3 servidores

Los nombres y puertos vienen indicados en este gráfico:



1) Crear una carpeta *practshard*, dentro de la que vamos a trabajar

Nota: en los siguientes apartados utilizar la opción `--smallfiles` al final de cada creación de `mongod`

a) Las carpetas de datos necesarias (desde el terminal). Copiar en la solución el comando linux para crear las carpetas de datos (no la carpeta exterior “practshard”)

b) Crear los servidores de configuración. Copiar las instrucciones que permiten hacerlo a la solución.

c) Iniciar los servidores de datos que formarán las particiones. No olvidar poner el nombre del conjunto réplica (a o b). Copiar las instrucciones en la solución

d) Crear un servicio cliente `mongos`. Poner `mongoslog.txt` como nombre del fichero log (opción `logpath`).

Nota: Recordar que aquí se usan los nombres completos de los servidores de configuración. Estos nombres serán de la forma `nombrehost.local:puerto`. Para averiguar el nombre del host en un terminal de linux se puede teclear “hostname”. Copiar en la solución la instrucción para crear el servicio `mongos`.

Para asegurarnos de que todos los servidores y servicios se han iniciado, podemos teclear en el terminal

```
ps -A | grep "mongo"
```

deben aparecernos 10 líneas: 9 para servidores (`mongod`) y 1 para el cliente `mongos`

2) Activar los dos conjuntos réplica. Recordar que conviene hacerlo desde un cliente (`mongo`)

conectado a un miembro de la réplica. Copiar en la solución las instrucciones que se han utilizado dentro del cliente mongo. Escribirlas en secuencia.

Aviso: revisar bien los “corta-pegar”; los carga el diablo.

3) Crear las dos particiones (shards). Copiar en la solución las instrucciones que se han utilizado dentro del cliente mongo. Escribirlas en secuencia.

4) Conectar con mongo (teclea **mongo** desde el terminal sin número de puerto).

Escribir

`use pin-pon`

`for (i=0;i<1000;i++){db.cole.insert({"pin":i, "pon":1000-i})}`

`sh.status()`

¿Está particionada la base de datos pin-pon? ¿Qué parte de la respuesta que muestra sh.status lo indica?

5)[2] Teclear en mongos (**mongo** sin puerto desde el terminal):

`use aleatorio`

`db.dados.drop()`

La colección dados representa una simulación de muchas tiradas de un dado. Los documentos, que insertaremos luego, son de la forma `{tirada:0, dado: 5, acierto:"huy...casi adivinas el resultado"}` con “tirada” un valor único que indica el orden en la simulación, `dado` el valor que ha salido, y `acierto` un mensaje que indica si se ha adivinado el resultado de la tirada o no.

Haz que la colección *dados* admita “sharding”. Al hacerlo selecciona la clave “dado”

Copia en la solución las instrucciones que han permitido que la colección admita particiones.

6) Ahora ejecutamos (tardará un rato):

`use aleatorio`

`for (i=1;i<=20000;i++){tira= Math.floor(Math.random()*6)+1;`

`apuesta=Math.floor(Math.random()*6)+1;`

`db.dados.insert({tirada:i,`

`dado:tira,`

`acierto: (tira==apuesta ? "Has acertado" :`

`"Bertoldo, has perdido. Ten en cuenta que las vidas de la mayoría de la`

`gente están dirigidas por el deseo y el miedo. El deseo es la necesidad de añadirte algo para poder ser tú mismo más plenamente. Todo miedo es el miedo de perder algo y, por tanto, de sentirte reducido y de ser menos de lo que eres. Estos dos movimientos oscurecen el hecho de que el Ser no puede ser dado ni quitado. El Ser ya está en ti en toda su plenitud, ahora (Eckhart Tolle)")); }`

`s = db.dados.stats()`

¿Cuántos documentos tiene cada shard? Obtener esta información a partir de la variable s

Pista1: Investiga el formato del documento almacenado en s y utiliza la notación . :

s.XXXX.YYYY.....

Indica en la solución, la cantidad de documentos por partición el comando que has utilizado para obtener la información y las expresiones a partir de s que permiten obtener esta información

7) Ahora vamos a ver cómo afecta la existencia de las particiones a los planes de ejecución, y por tanto a la eficiencia.

```
exp = db.dados.explain("executionStats")
exp.find({tirada: {'$gte':9980}}, {_id:0,tirada:1})
```

- a) ¿Cuántos shards se han consultado? (consultar el array shards)
- b) ¿Cuántas claves?
- c) ¿Cuántos documentos?

Ayuda: el documento que devuelve puede ser muy largo, mejor copiarlo y verlo en un editor (por ejemplo gedit)

8) Análogo pero para la clave *dado*:

```
exp = db.dados.explain("executionStats")
exp.find({dado:5}, {_id:0,dado:1})
```

- a) ¿Cuántos shards se han consultado?
- b) ¿Cuántas claves?
- c) ¿Cuántos documentos?

9) Teclear: [use config](#)

Escribir una consulta que indique el shard del valor dado:4 . Copiarla en la solución.

Pista: [db.chunks.find\(...\)](#)