Corporación universitaria iberoamericana

2022

Integrantes

Laura Viviana Rojas Henao  
Hernan Darío Seco Blanco  
Miguel Ángel Cifuentes

Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL

Actividad 5 y 6

# **Torneo de Micro Futbol 5**

Se va llevar a cabo un torneo de micro futbol en la vereda cruz verde, municipio de El Rosal Cundinamarca. Este torneo consta de 24 equipos, el cual tendrá una premiación de la siguiente manera.

* El equipo campeón: $2.000.000
* El equipo subcampeón: $1.000.000
* El tercer puesto: $400.000
* El cuarto puesto: $200.000
* La valla menos vencida: $70.000
* El goleador: $70.000

**Reglas del torneo.**

* El Equipo deberá tener mínimo 5 personas.
* Cada equipo deberá tener su capitán.
* En la fase 1 se eliminarán 4 equipos.
* En la fase 2 se eliminarán 4 equipos.
* En la fase 3 (muerte súbita) se eliminarán 8 equipos.
* En la fase 4 (muerte súbita) se eliminarán 4 equipos.
* En la fase 5 (cruzado) se eliminan 2 equipos.
* En la fase final se elimina 1 equipo.

**Requerimientos del torneo.**

* Registrar cada persona que hace parte del torneo diferenciándolas por su respectivo rol.
* Registrar todos los equipos que se disputaran en el torneo.
* Llevar un control de la tabla de los enfrentamientos.
* Llevar un control de la tabla de posiciones.
* Obtener el registro del goleador del torneo.
* Obtener el registro de la valla menos vencida del torneo.

**Actividad 3 - Requerimientos No Funcionales.**

* El torneo se llevará a cabo en el transcurso de 1 mes o 30 días.
* Garantizar el funcionamiento del sistema durante todo el torneo.
* El sistema deberá tener un rendimiento optimo en cualquier ejecución de los eventos.
* El sistema debe estar disponible para todos los usuarios.
* El sistema tendrá una adaptabilidad para visualizarse en diferentes dispositivos.
* El sistema deberá soportar la capacidad de varios usuarios o sesiones en simultaneo.
* Garantizar el buen manejo de la integridad de los datos.
* Llevar a cabo un buen manejo de la seguridad de datos.

**Estrategia de replicación**

Como estrategia visualizamos primero obtener las colecciones en orden de prioridad las cuales deberán presentar una disponibilidad 24x7, estas colecciones estarán recibiendo información constantemente. Estas tablas son:

* Resultado de partido.
* Tabla de posiciones.
* Premiación.
* Personas
* Equipos.

Esta secuencia está definida de acuerdo con que la colección de resultados de partido es la que más va a recibir información durante el torneo seguido de ella la colección de tabla de posiciones que se ira actualizando de acuerdo con los resultados del partido, la colección de premiación tiene mayor influencia al final del torneo y en el caso de las colecciones personas y equipos únicamente reciben información al inicio del torneo. Por esta razón la estrategia planteada es usar 2 nodos adicionales los cuales nos ayuden a garantizar el funcionamiento de la base de datos las 24x7.

**Diagrama

Descripción generada automáticamenteActividad 4 - Casos de uso**

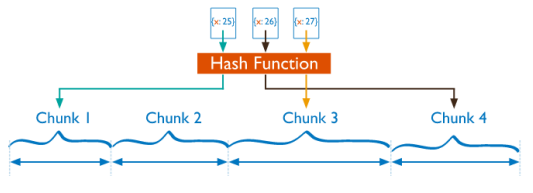
**Actividad 5 y 6 - Estrategia de fragmentado**

El escalado horizontal lo que implica es dividir el conjunto de datos del sistema y cargarlo en varios servidores, esto nos permitirá ir agregando servidores para aumentar la capacidad según la necesidad.

Existen dos estrategias de fragmentado para distribuir datos entre los clústeres:

Hashed Sharding y Ranged Sharding.

Teniendo en cuenta que nuestro torneo tiene una duración de 30 días consecutivos en los que va a recibir información constante y en simultáneo debemos garantizar el buen rendimiento del sistema en cada ejecución de sus tareas se visualiza una estrategia **Hashed Sharding** ya que facilita una distribución más uniforme de los datos, especialmente en conjuntos de datos en los que la clave del fragmento cambia de forma monótona.



**Ventaja**

La división de hash es complementaria a la división de rango, que puede distribuir documentos aleatoriamente a varios fragmentos, expandir completamente la capacidad de escritura y compensar la falta de división de rango

En el caso del  fragmentado este distribuye datos a través de los fragmentos de cada clúster, permitiendo que cada fragmento tenga un subconjunto del total de datos del clúster. Cada vez que el conjunto de datos crece, los fragmentos adicionales aumentan la capacidad de almacenamiento del clúster.

**Comandos Usados para realizar la Partición**

# Se crea cada servidor de configuración, para lo que por cada servidor de

# Configuración se abre una instancia de mongod en un puerto con la opción --port,

# Se indica el directorio dónde se almacena los datos con la opción –dbpath,

# Se indica la opción --configsvr

# Instancia mongod para el servidor

**mongod --dbpath C:\data\config1 --configsvr --port 27022  --replSet replica1**

**mongod --dbpath C:\data\config2 --configsvr --port 27023 --replSet replica1**

**mongod --dbpath C:\data\config3 --configsvr --port 27024 --replSet replica1**

#A continuación, inicie los fragmentos como se muestra aquí.

**mongod --dbpath C:\data\shard1\data1 --configsvr --replSet replica1 --port 27022**

**mongod --dbpath C:\data\shard1\data2 --configsvr --replSet replica1 --port 27022**

**mongod --dbpath C:\data\shard1\data3 --configsvr --replSet replica1 --port 27022**

# Ahora, conéctese a uno de los servidores de fragmentos para habilitar un conjunto

# De réplicas como mostrado aquí y en

**mongo hostname:26017**

# Enrutadores

# Para crear un enrutador se abre una instancia de mongos

# Este programa se puede encontrar en el directorio bin de la instalación de MongoDB.

# Se invoca indicando el puerto donde se va a ejecutar con la opción --port, se listan los

# Servidores de configuración indicando el nombre del equipo y el puerto donde se ejecutan,

# y se usa la opción --configdb para indicar que es un enrutador.

**mongos --configdb localhost:27022,localhost:27023,localhost:27024  --port 27021**

**mongos --configdb replica1/localhost:27022,localhost:27023,localhost:27024  --port 27021**

# Hay que observar que también es posible en este paso indicar

# El tamaño de los trozos con la opción --chunkSize. Si no se indica esta opción,

# Por defecto toma el valor de 64 MB.

# Shards

#  Se van a crear, por ejemplo, dos shards, por lo que es necesa- rio disponer de dos

#  Directorios para cada uno de los servidores

# Para crear un shard se abre una instancia de mongod indi- cando

#  El puerto donde se va a ejecutar con la opción –port, se indica

#  El directorio donde se almacenan los datos con la opción –dbpath,

#  Se indica la opción --shardsvr

**mongod --dbpath C:\data\shards\n1 --shardsvr --replSet replica1 --port 27026**

**mongod --dbpath C:\data\shards\n2 --shardsvr --replSet replica1 --port 27027**

**mongod --dbpath C:\data\shards\n3 --shardsvr --replSet replica1 --port 27028**

# A continuación se agregan los shards al clúster (figura 160). Para ello se inicia una instancia de mongo

#  En el puerto donde se está ejecutando el enrutador, y cada shard se añade con el comando sh.addShard

#  («Nombre del equipo: puerto»), indicando el equipo y el puerto donde se ejecuta el shard.

**sh.addShard(«Nombre del equipo: puerto»)**

**sh.addShard("replica1/localhost:27026")**

**sh.status()**

# Ahora, para habilitar la fragmentación para una base de datos, use este comando.

**mongos> sh.enableSharding("demos")**

# Para habilitar la fragmentación de una colección, use este código.

**mongos> sh.shardCollection("demos.users",{"id":1}**

**{**

**"ok" : 0,**

**"errmsg" : "no such command: 'addShard'",**

**"code" : 59,**

**"codeName" : "CommandNotFound",**

**"lastCommittedOpTime" : Timestamp(1650504898, 1),**

**"$clusterTime" : {**

**"clusterTime" : Timestamp(1650504898, 1),**

**"signature" : {**

**"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),**

**"keyId" : NumberLong(0)**

**}**

**},**

**"operationTime" : Timestamp(1650504898, 1)**

**}**

**Link de github:**

[**https://github.com/vinan1709/torneo\_micro\_tarea**](https://github.com/vinan1709/torneo_micro_tarea)

**Link del video:**

[**https://youtu.be/XgYnGJhhsRU**](https://youtu.be/XgYnGJhhsRU)

Referencias:

* <https://dit.gonzalonazareno.org/gestiona/proyectos/2016-17/Cluster_Sharding_MongoDB_Carlos_Garcia.pdf>
* https://programmerclick.com/article/8971439868/