INSTITUTO UNIVERSITÁRIO DE LISBOA

Armazenamento de Dados em Ambientes Distribuídos (ADAD)

Mestrado em Engenharia de Telecomunicações e Informática (METI)

Configuração Sharding

Catarina Ferreira da Silva
catarina.ferreira.silva@iscte-iul.pt

João Miguel Gonçalves Matos
joao_miguel_goncalves_matos@iscte-iul.pt

Configuração Sharding MongoDB

Os alunos deverão configurar um cluster com sharding manualmente, utilizando instâncias locais do MongoDB. (Nota: Devem ter o MongoDB instalado localmente e acessível via linha de comandos (`mongod`, `mongo` ou `mongosh`))

1.2.1 Criar diretorias para cada instância:

```
mkdir -p ~/mongo_cluster/configdb
mkdir -p ~/mongo_cluster/shard1
mkdir -p ~/mongo_cluster/shard2
mkdir -p ~/mongo_cluster/router
```

1.2.2 Iniciar o Config Server

- Em um novo terminal executar:

```
mongod --configsvr --replSet configReplSet --port 27019 --dbpath
~/mongo_cluster/configdb
```

> **Manter este processo em execução.**
> **Config Server usa a porta padrão 27019**

- Em um novo terminal:

```
mongosh --port 27019
```

- Inicializar o *replica set* do config server:

```
rs.initiate({
  _id: "configReplSet",
  configsvr: true,
  members: [{ _id: 0, host: "localhost:27019" }]
})
```

1.2.3 Iniciar os Shards

Shard 1

- Em um novo terminal:

```
mongod --shardsvr --replSet shard1ReplSet --port 27018 --dbpath
~/mongo_cluster/shard1
```

- Em um novo terminal:

```
mongosh --port 27018
```

```
rs.initiate({
  _id: "shard1ReplSet",
  members: [{ _id: 0, host: "localhost:27018" }]
})
```

Shard 2

- Em um novo terminal

```
mongod --shardsvr --replSet shard2ReplSet --port 27017 --dbpath
~/mongo_cluster/shard2
```

- Em um novo terminal

```
mongosh --port 27017
```

```
rs.initiate({
  _id: "shard2ReplSet",
  members: [{ _id: 0, host: "localhost:27017" }]
})
```

1.2.4 Iniciar o Router (mongos)

- Em um novo terminal

```
mongos --configdb configReplSet/localhost:27019 --port 27020 --
bind_ip_all
```

O router fica assim acessível na porta 27020
É este o endereço que a API Node.js deve usar para conectar ao MongoDB

1.2.5 Configurar o Cluster

- Em um novo terminal

```
mongosh --port 27020
```

Adicionar os shards ao cluster:

```
sh.addShard("shard1ReplSet/localhost:27018")
sh.addShard("shard2ReplSet/localhost:27017")
```

Ativar o sharding para o banco de dados do projeto: (Nota: primeiro conetar no Compass com a string de conexão: mongodb://localhost:27020/).

Dentro do MONGOSH no Compass executar:

```
use project
```

Criar o *index* necessário para a *shard key*:

```
db.users.createIndex({ partitionKey: 1, _id: 1 })
```

Ativar sharding para a base de dados "project":

```
sh.enableSharding("project")
```

Definir a chave de fragmentação. A chave será baseada em dois campos (`partitionKey` e `_id`):

```
sh.shardCollection("project.users", { partitionKey: 1, _id: 1 })
```

Criar tags para cada Shard:

```
sh.addShardTag("shard1ReplSet", "shardA")
```

```
sh.addShardTag("shard2ReplSet", "shardB")
```

Criar o intervalo correspondente com a chave a cada tag respectiva:

```
sh.addTagRange("project.users", { partitionKey: 0 }, { partitionKey:
1 }, "shardA")
```

```
sh.addTagRange("project.users", { partitionKey: 1 }, { partitionKey:
2 }, "shardB")
```

> Para o valor de partitionKey = 0, definimos o intervalo entre [0, 1];
> Para o valor de partitionKey = 1, definimos o intervalo entre [1, 2].

Verificar que os diferentes documentos e chunks foram divididos por cada Sharding:

```
db.chunks.find({ ns: "project.users" }).sort({min:1}).pretty()
db.getSiblingDB("project").users.getShardDistribution()
```

Documentos com o valor partitionKey: 0, deverão estar na shard1 (shard1ReplSet) e os documentos com o valor partitionKey: 1, deverão estar na shard2 (shard2ReplSet) como mostra a imagem:

```
> db.chunks.find({ ns: "project.users" }).sort({min:1}).pretty()
<
> db.getSiblingDB("project").users.getShardDistribution()
<
Shard shard1ReplSet at shard1ReplSet/localhost:27018

{
  data: '194B',
  docs: 3,
  chunks: 1,
  'estimated data per chunk': '194B',
  'estimated docs per chunk': 3
}
---
Shard shard2ReplSet at shard2ReplSet/localhost:27017

{
  data: '144B',
  docs: 2,
  chunks: 3,
  'estimated data per chunk': '48B',
  'estimated docs per chunk': 0
}
---
```

```
Totals

{
  data: '338B',
  docs: 5,
  chunks: 4,
  'Shard shard1ReplSet': [
    '57.39 % data',
    '60 % docs in cluster',
    '64B avg obj size on shard'
  ],
  'Shard shard2ReplSet': [
    '42.6 % data',
    '40 % docs in cluster',
    '72B avg obj size on shard'
  ]
}
```

1.3 Reiniciar Cluster

Após parar todos os serviços (fechando o terminal) o cluster deixa de funcionar. Não é necessário voltar a repetir toda a configuração anterior. Basta apenas reiniciarmos os servidores com os mesmos comandos usados antes:

1.3.1 Start Config Server

```
mongod --configsvr --replSet configReplSet --port 27019 --dbpath
~/mongo_cluster/configdb
```

1.3.2 Start Shard 1

```
mongod --shardsvr --replSet shard1ReplSet --port 27018 --dbpath
~/mongo_cluster/shard1
```

1.3.3 Start Shard 2

```
mongod --shardsvr --replSet shard2ReplSet --port 27017 --dbpath
~/mongo_cluster/shard2
```

1.3.4 Start Router (mongos)

```
mongos --configdb configReplSet/localhost:27019 --port 27020 --
bind_ip_all
```