SOFT8023

Local Cluster Using Restaurant Dataset

Setting Up Our Cluster

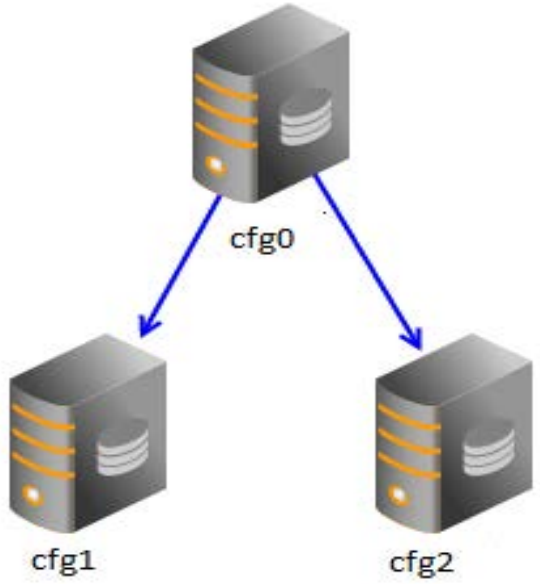
- We are going to create a cluster of 15 independent nodes on the local machine, each node will share same ip, but different port.
- See the cluster.zip file on Blackboard, which has a script-based procedure, so that you can create the cluster on your own machine as well. Follow the instructions in README.txt.

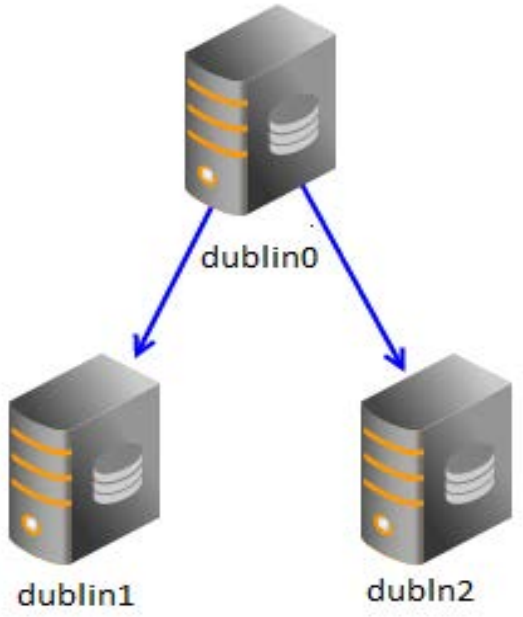
- In the simulation, the 15 nodes represents 15 servers, distributed over 4 small data centres: 6 nodes in Dublin and 9 nodes in Cork, Limerick and Galway (3 nodes each).
- The following picture represents the data centres and their associated nodes:

Setting Up Our Cluster

Setting Up Our Cluster

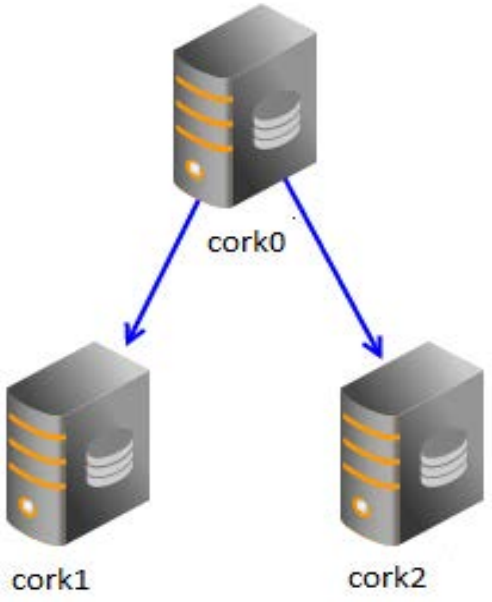
- Dublin data centre → Nodes:
  - Configuration nodes (or metadata): { cfg0, cfg1, cfg2 } in replica set “cfg”
  - First shard “dublin”. It is added from the replica set of the three nodes: { dublin0, dublin1, dublin2 }





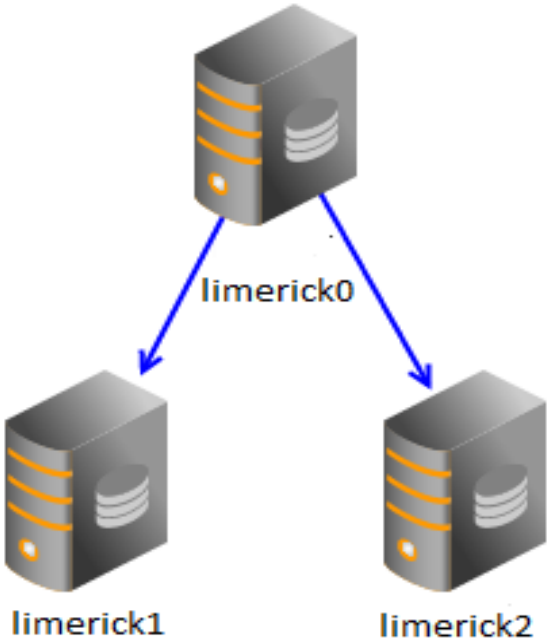
Setting Up Our Cluster

- Cork Data Centre → Nodes:
  - Second shard “cork”. It is added from the replica set of the three nodes: { cork0, cork1, cork2 }



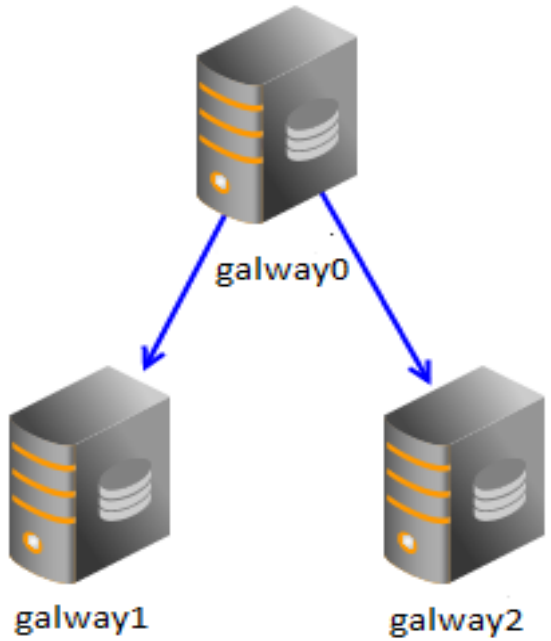
Setting Up Our Cluster

- Limerick Data Centre → Nodes:
  - Third shard “limerick”. It is added from the replica set of the three nodes: { limerick0, limerick1, limerick2 }



Setting Up Our Cluster

- Galway Data Centre → Nodes:
  - Fourth shard “galway”. It is added from the replica set of the three nodes: { galway0, galway1, galway2 }



Setting Up Our Cluster

- The script-based procedure to set up the cluster consists on:
  - 1.create_nodes.bat
  - 1.setup_config.js
  - 2.setup_cluster.bat
  - 2.setup_cluster.js
  - 3.insert_collection.bat
  - 3.restaurants_dataset.json
  - 4.shard_collection.bat
  - 4.shard_collection.js
  - 5.remove_cluster.bat
  - README.txt

Setting Up Our Cluster

- **1.create_nodes.bat**
  - It creates the data directories for the metadata and shard nodes.
  - It starts the mongod.exe processes of the config nodes of Dublin.
  - It creates the replica set for the config nodes (calling 1.setup_config.js).
  - It starts the four mongos.exe processes for interfacing the cluster, referencing the cfg replica set.
  - It starts the 12 mongod.exe processes for the 12 servers of Dublin, Cork, Limerick and Galway. (all with shardsvr parameter)

Setting Up Our Cluster

- **2.setup_cluster.bat:**
  It executes the JavaScript program 2.setup_cluster.js.
  - It sets up the replica set for dublin, cork, limerick and galway by running the rs.initiate command (it waits until each replica set has all nodes up and working as either primary or secondary).
  - It connects to the configuration node so as to set the chunk size to 1MB.
  - It also connects to the configuration node so as to add the four replica sets as new shards.

Setting Up Our Cluster

```
D:\OneDrive - Cork Institute of Technology\Modules\DATA8005 - Distributed Data Management\2. Code Examples\L07. Demonstr
ating Sharding\3. Cluster Environment>mongo.exe --shell 2.setup_cluster.js
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
type "help" for help
connecting to: mongodb://127.0.0.1:27000/test
MongoDB server version: 3.6.0
Dublin Replica Set Created!
Dublin Replica Set Up!
Dublin Shard Added!
connecting to: mongodb://127.0.0.1:27100/test
MongoDB server version: 3.6.0
Cork Replica Set Created!
Cork Replica Set Up!
Cork Shard Added!
connecting to: mongodb://127.0.0.1:27200/test
MongoDB server version: 3.6.0
Limerick Replica Set Created!
Limerick Replica Set Up!
Limerick Shard Added!
connecting to: mongodb://127.0.0.1:27300/test
MongoDB server version: 3.6.0
Galway Replica Set Created!
Galway Replica Set Up!
Galway Shard Added!

D:\OneDrive - Cork Institute of Technology\Modules\DATA8005 - Distributed Data Management\2. Code Examples\L07. Demonstr
ating Sharding\3. Cluster Environment>REM #
```

Setting Up Our Cluster

- **3.insert_collection.bat:**
  - It creates a new collection, *restaurants*, in the database *test*.
  - It fills the collection with the more than 20,000 documents from the file restaurants_dataset.json by using the mongoimport.exe tool.
  - Where are the restaurant collection's documents initially stored? We will see it in a minute. But the answer is clear, in the primary shard for the database test.

Setting Up Our Cluster

2.setup_cluster.js × | 3.restaurants_dataset.json × | 4.shard_collection.js ×


```
{"address": {"building": "1007", "coord": [-73.856077, 40.848447], "street": "Morris Park Ave", "zipcode": "10462"}
{"address": {"building": "469", "coord": [-73.961704, 40.662942], "street": "Flatbush Avenue", "zipcode": "11225"},
{"address": {"building": "351", "coord": [-73.98513559999999, 40.7676919], "street": "West   57 Street", "zipcode":
{"address": {"building": "2780", "coord": [-73.98241999999999, 40.579505], "street": "Stillwell Avenue", "zipcode":
{"address": {"building": "97-22", "coord": [-73.8601152, 40.7311739], "street": "63 Road", "zipcode": "11374"}, "bo
{"address": {"building": "8825", "coord": [-73.8803827, 40.7643124], "street": "Astoria Boulevard", "zipcode": "113
{"address": {"building": "2206", "coord": [-74.1377286, 40.6119572], "street": "Victory Boulevard", "zipcode": "103
{"address": {"building": "7114", "coord": [-73.9068506, 40.6199034], "street": "Avenue U", "zipcode": "11234"}, "bo
{"address": {"building": "6409", "coord": [-74.00528899999999, 40.628886], "street": "11 Avenue", "zipcode": "11219
{"address": {"building": "1839", "coord": [-73.9482609, 40.6408271], "street": "Nostrand Avenue", "zipcode": "11226
{"address": {"building": "2300", "coord": [-73.8786113, 40.8502883], "street": "Southern Boulevard", "zipcode": "10
{"address": {"building": "7715", "coord": [-73.9973325, 40.61174889999999], "street": "18 Avenue", "zipcode": "1121
{"address": {"building": "1269", "coord": [-73.871194, 40.6730975], "street": "Sutter Avenue", "zipcode": "11208"},
{"address": {"building": "1", "coord": [-73.96926909999999, 40.7685235], "street": "East   66 Street", "zipcode": "
{"address": {"building": "705", "coord": [-73.9653967, 40.6064339], "street": "Kings Highway", "zipcode": "11223"},
```

Setting Up Our Cluster

```
D:\OneDrive - Cork Institute of Technology\Modules\DATA8005 - Distributed Data Management\2. Code Examples\L07. Demonstr
ating Sharding\3. Cluster Environment>mongoimport.exe --db test --collection restaurants --drop --file 3.restaurants_dat
aset.json
2018-10-14T17:51:29.263+0100    connected to: localhost
2018-10-14T17:51:29.264+0100    dropping: test.restaurants
2018-10-14T17:51:31.258+0100    [#.......................] test.restaurants     545KB/11.3MB (4.7%)
2018-10-14T17:51:34.257+0100    [##......................] test.restaurants     1.05MB/11.3MB (9.2%)
2018-10-14T17:51:37.257+0100    [######..................] test.restaurants     3.10MB/11.3MB (27.4%)
2018-10-14T17:51:40.258+0100    [############............] test.restaurants     6.19MB/11.3MB (54.7%)
2018-10-14T17:51:43.257+0100    [###################.....] test.restaurants     9.03MB/11.3MB (79.7%)
2018-10-14T17:51:46.239+0100    [########################] test.restaurants     11.3MB/11.3MB (100.0%)
2018-10-14T17:51:46.240+0100    imported 25359 documents

D:\OneDrive - Cork Institute of Technology\Modules\DATA8005 - Distributed Data Management\2. Code Examples\L07. Demonstr
ating Sharding\3. Cluster Environment>REM #
```

Setting Up Our Cluster

- **3.insert_collection.bat:**

After completing the task, some issues might be highlighted:

- At this stage, the collection *restaurants* of the database *test* is still not sharded, as we have not explicitly asked for it.
- Thus, the documents of the collection are not divided into chunks.
- Instead, all the documents are placed in the primary shard of the test database: galway!
  - In the next slide we can see that there are 4 shards, but 0 chunks.
  - There are 2 databases, config and test.
  - The db *test* is not partitioned, and has galway as its primary shard.

Setting Up Our Cluster

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("5bc3732c100b279a5ef52f07")
  }
  shards:
        {  "_id" : "cork",  "host" : "cork/127.0.0.1:27100,127.0.0.1:27101,127.0.0.1:27102",  "state" : 1 }
        {  "_id" : "dublin",  "host" : "dublin/127.0.0.1:27000,127.0.0.1:27001,127.0.0.1:27002",  "state" : 1 }
        {  "_id" : "galway",  "host" : "galway/127.0.0.1:27300,127.0.0.1:27301,127.0.0.1:27302",  "state" : 1 }
        {  "_id" : "limerick",  "host" : "limerick/127.0.0.1:27200,127.0.0.1:27201,127.0.0.1:27202",  "state" : 1 }
  active mongoses:
        "3.6.0" : 4
  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" : "config",  "primary" : "config",  "partitioned" : true }
                config.system.sessions
                        shard key: { "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                cork    1
                        { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : cork Timestamp(1, 0)
        {  "_id" : "test",  "primary" : "galway",  "partitioned" : false }

mongos>
```

Setting Up Our Cluster

- **4.shard_collection.bat:**
  It executes the JavaScript program 4.shard_collection.js.
  - It enables the database test for sharding. From that moment on, any collection of test can be sharded if we explicitly ask for it.
  - It creates the index { “cuisine”, “borough” } for the collection restaurants of the database test.
  - It explicitly asks to shard the collection restaurants, using the index { “cuisine”, “borough” } as the shard key.

Setting Up Our Cluster

- **4.shard_collection.bat:**
  It executes the JavaScript program 4.shard_collection.js.

By explicitly asking to shard the collection restaurants using the index { “cuisine”, “borough” } as the shard key…

- The metadata daemons split and migrate are activated.
- Split starts splitting the documents into manageable chunks (of at most 1MB of information).
  Given the size of the collection, it leads to 26 chunks.
- The migrator moves chunks from one shard to another to achieve a fair balance distribution of the chunks among the shards.

Setting Up Our Cluster

- Starting to rebalance “3 : Success”, i.e. 3 chunks migrated.

```
balancer:
      Currently enabled:  yes
      Currently running:  yes
      Collections with active migrations:
              test.restaurants started at Sun Oct 14 2018 17:58:18 GMT+0100 (GMT Standard Time)
      Failed balancer rounds in last 5 attempts:  0
      Migration Results for the last 24 hours:
              3 : Success
databases:
      {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
              config.system.sessions
                      shard key: { "_id" : 1 }
                      unique: false
                      balancing: true
                      chunks:
                              cork    1
                      { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : cork Timestamp(1, 0)
      {  "_id" : "test",  "primary" : "galway",  "partitioned" : true }
              test.restaurants
                      shard key: { "cuisine" : 1, "borough" : 1 }
                      unique: false
                      balancing: true
                      chunks:
                              cork    1
                              dublin  1
                              galway  17
                              limerick        1
                      too many chunks to print, use verbose if you want to force print
```

Setting Up Our Cluster

- Well underway “10 : Success”, i.e. 10 chunks migrated.

```
balancer:
	Currently enabled:  yes
	Currently running:  yes
	Collections with active migrations:
		test.restaurants started at Sun Oct 14 2018 17:59:04 GMT+0100 (GMT Standard Time)
	Failed balancer rounds in last 5 attempts:  0
	Migration Results for the last 24 hours:
		10 : Success
databases:
	{  "_id" : "config",  "primary" : "config",  "partitioned" : true }
		config.system.sessions
			shard key: { "_id" : 1 }
			unique: false
			balancing: true
			chunks:
				cork	1
			{ "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : cork Timestamp(1, 0)
	{  "_id" : "test",  "primary" : "galway",  "partitioned" : true }
		test.restaurants
			shard key: { "cuisine" : 1, "borough" : 1 }
			unique: false
			balancing: true
			chunks:
				cork	3
				dublin	4
				galway	10
				limerick	3
			too many chunks to print, use verbose if you want to force print
```

Setting Up Our Cluster

- Almost there…

```
balancer:
      Currently enabled:  yes
      Currently running:  yes
      Collections with active migrations:
              test.restaurants started at Sun Oct 14 2018 17:59:35 GMT+0100 (GMT Standard Time)
      Failed balancer rounds in last 5 attempts:  0
      Migration Results for the last 24 hours:
              14 : Success
databases:
      {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
              config.system.sessions
                      shard key: { "_id" : 1 }
                      unique: false
                      balancing: true
                      chunks:
                              cork    1
                      { "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : cork Timestamp(1, 0)
      {  "_id" : "test",  "primary" : "galway",  "partitioned" : true }
              test.restaurants
                      shard key: { "cuisine" : 1, "borough" : 1 }
                      unique: false
                      balancing: true
                      chunks:
                              cork    5
                              dublin  5
                              galway  6
                              limerick        4
                      too many chunks to print, use verbose if you want to force print
```

Setting Up Our Cluster

- And then… perfect balance, 15 migrations and 5 chunks (5MB) in each shard

```
balancer:
	Currently enabled:  yes
	Currently running:  no
	Failed balancer rounds in last 5 attempts:  0
	Migration Results for the last 24 hours:
		15 : Success
databases:
	{  "_id" : "config",  "primary" : "config",  "partitioned" : true }
		config.system.sessions
			shard key: { "_id" : 1 }
			unique: false
			balancing: true
			chunks:
				cork	1
			{ "_id" : { "$minKey" : 1 } } -->> { "_id" : { "$maxKey" : 1 } } on : cork Timestamp(1, 0)
	{  "_id" : "test",  "primary" : "galway",  "partitioned" : true }
		test.restaurants
			shard key: { "cuisine" : 1, "borough" : 1 }
			unique: false
			balancing: true
			chunks:
				cork	5
				dublin	5
				galway	5
				limerick	5
			too many chunks to print, use verbose if you want to force print
```

Setting Up Our Cluster

- **4.shard_collection.bat:**
  - Now we can connect to the database *config* to know:
    1. How many chunks there are.
    2. What is the shard key range of each of them.
    3. In which shard is each chunk hosted.

```
mongos> use config
switched to db config
mongos> db.chunks.count()
21
mongos>
```

```
mongos> db.chunks.find().pretty()
{
        "_id" : "config.system.sessions-_id_MinKey",
        "ns" : "config.system.sessions",
        "min" : {
                "_id" : { "$minKey" : 1 }
        },
        "max" : {
                "_id" : { "$maxKey" : 1 }
        },
        "shard" : "cork",
        "lastmod" : Timestamp(1, 0),
        "lastmodEpoch" : ObjectId("5bc37441100b279a5ef53a59")
}
{
        "_id" : "test.restaurants-cuisine_MinKeyborough_MinKey",
        "lastmod" : Timestamp(2, 0),
        "lastmodEpoch" : ObjectId("5bc37587100b279a5ef54787"),
        "ns" : "test.restaurants",
        "min" : {
                "cuisine" : { "$minKey" : 1 },
                "borough" : { "$minKey" : 1 }
        },
        "max" : {
                "cuisine" : "American ",
                "borough" : "Brooklyn"
        },
        "shard" : "dublin"
}
{
        "_id" : "test.restaurants-cuisine_\"American \"borough_\"Brooklyn\"",
        "lastmod" : Timestamp(3, 0),
        "lastmodEpoch" : ObjectId("5bc37587100b279a5ef54787"),
        "ns" : "test.restaurants",
        "min" : {
                "cuisine" : "American ",
                "borough" : "Brooklyn"
        },
        "max" : {
                "cuisine" : "American ",
                "borough" : "Manhattan"
        },
        "shard" : "cork"
}
{
        "_id" : "test.restaurants-cuisine_\"American \"borough_\"Manhattan\"",
        "lastmod" : Timestamp(4, 0),
        "lastmodEpoch" : ObjectId("5bc37587100b279a5ef54787"),
        "ns" : "test.restaurants",
        "min" : {
                "cuisine" : "American ",
                "borough" : "Manhattan"
        },
        "max" : {
                "cuisine" : "American ",
                "borough" : "Missing"
        },
        "shard" : "limerick"
```

Setting Up Our Cluster

2. What is the shard key range of each of them. → db.chunks.find()
3. In which shard is each chunk hosted. → also db.chunks.find()

Thank you for your attention!