

CS185C: Introduction to NoSQL

Instructor: Dr. Kim

Sharding configuration on single machine

I. Set up shards

Test Configuration: a config replica set, a shard is not replicated

Service	Daemon	Port	DBpath
Shard Controller	mongos	27021	N/A
Config Server	mongod	27022	/db/config/data
Shard0	mongod	27023	/db/shard1/data
Shard1	mongod	27024	/db/shard2/data

[skim.Skim-It2] ► ssh 127.0.0.1 -p 2222 -l vagrant

1. Set up a config replica set

(a) Create a directory

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/config/data
```

(b) Start a mongod serving as a config server

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27022 --dbpath /db/config/data --configsvr -  
-replSet config
```

...

```
2016-12-20T02:33:51.482+0000 I NETWORK [HostnameCanonicalizationWorker] Starting hostname  
canonicalization worker
```

```
2016-12-20T02:33:51.503+0000 I NETWORK [initandlisten] waiting for connections on port 27022
```

(c) In a new window, initialize the replica set

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo --port 27022
```

```
MongoDB shell version: 3.2.9
```

```
connecting to: 127.0.0.1:27022/test
```

```
Server has startup warnings:
```

```
2016-12-20T02:33:51.437+0000 I CONTROL [initandlisten] ** WARNING: You are running this process as  
the root user, which is not recommended.
```

2016-12-20T02:33:51.438+0000 I CONTROL [initandlisten]

> rs.initiate()

```
{
  "info2" : "no configuration specified. Using a default configuration for the set",
  "me" : "vagrant-ubuntu-trusty-64:27022",
  "ok" : 1
}
```

config:OTHER>

2. Set up the shard controller (mongos) in a new window

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongos --configdb config/127.0.0.1:27022 --port 27021 --
chunkSize 1
```

The mongos is going to listen on port 27021 and also identifies the config server called config/127.0.0.1:27022

...

2016-12-20T02:43:00.514+0000 I SHARDING [Balancer] distributed lock 'balancer' acquired for 'doing balance round', ts : 58589ab4b4ccc8eb41074b0c

2016-12-20T02:43:00.518+0000 I SHARDING [Balancer] distributed lock with ts: 58589ab4b4ccc8eb41074b0c' unlocked.

2016-12-20T02:43:10.528+0000 I SHARDING [Balancer] distributed lock 'balancer' acquired for 'doing balance round', ts : 58589abeb4ccc8eb41074b0d

2016-12-20T02:43:10.534+0000 I SHARDING [Balancer] distributed lock with ts: 58589abeb4ccc8eb41074b0d' unlocked.

3. Bring up the two shard servers

(1) Open two new terminal, one for each server

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard0/data
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27023 --dbpath /db/shard0/data
```

```

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27023 --dbpath /db/shard0/data
2016-12-20T02:57:06.307+0000 I CONTROL [initandlisten] MongoDB starting : pid=2616 port
=27023 dbpath=/db/shard0/data 64-bit host=vagrant-ubuntu-trusty-64
2016-12-20T02:57:06.308+0000 I CONTROL [initandlisten] db version v3.2.9
2016-12-20T02:57:06.309+0000 I CONTROL [initandlisten] git version: 22ec9e93b40c85fc7ca
e7d56e7d6a02fd811088c
2016-12-20T02:57:06.309+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1f
6 Jan 2014
2016-12-20T02:57:06.311+0000 I CONTROL [initandlisten] allocator: tcmalloc
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] modules: none
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] build environment:
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] distmod: ubuntu1404
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] distarch: x86_64
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] target_arch: x86_64
2016-12-20T02:57:06.312+0000 I CONTROL [initandlisten] options: { net: { port: 27023 },
storage: { dbPath: "/db/shard0/data" } }
2016-12-20T02:57:06.404+0000 I - [initandlisten] Detected data files in /db/shard
0/data created by the 'wiredTiger' storage engine, so setting the active storage engine
to 'wiredTiger'.
2016-12-20T02:57:06.406+0000 I STORAGE [initandlisten] wiredtiger_open config: create,c
ache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fa
st),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_
idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),

```

vagrant@vagrant-ubuntu-trusty-64:~\$ sudo mkdir -p /db/shard1/data

vagrant@vagrant-ubuntu-trusty-64:~\$ sudo mongod --port 27024 --dbpath /db/shard1/data

```

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27024 --dbpath /db/shard1/data
2016-12-20T03:00:36.275+0000 I CONTROL [initandlisten] MongoDB starting : pid=2772 port
=27024 dbpath=/db/shard1/data 64-bit host=vagrant-ubuntu-trusty-64
2016-12-20T03:00:36.277+0000 I CONTROL [initandlisten] db version v3.2.9
2016-12-20T03:00:36.278+0000 I CONTROL [initandlisten] git version: 22ec9e93b40c85fc7ca
e7d56e7d6a02fd811088c
2016-12-20T03:00:36.280+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1f
6 Jan 2014
2016-12-20T03:00:36.282+0000 I CONTROL [initandlisten] allocator: tcmalloc
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] modules: none
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] build environment:
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] distmod: ubuntu1404
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] distarch: x86_64
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] target_arch: x86_64
2016-12-20T03:00:36.283+0000 I CONTROL [initandlisten] options: { net: { port: 27024 },
storage: { dbPath: "/db/shard1/data" } }
2016-12-20T03:00:36.380+0000 I STORAGE [initandlisten] wiredtiger_open config: create,c
ache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fa
st),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_
idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),

```

4. Tell the sharding system (mongos) where the shard servers are located.

```
Last login: Tue Dec 20 02:59:24 2016 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongo 127.0.0.1:27021
MongoDB shell version: 3.2.9
connecting to: 127.0.0.1:27021/test
mongos> sh.addShard("127.0.0.1:27023")
{ "shardAdded" : "shard0000", "ok" : 1 }
mongos> sh.addShard("127.0.0.1:27024")
{ "shardAdded" : "shard0001", "ok" : 1 }
mongos>
```

5. Check the shards (before populating data)

```
mongos> db.printShardingStatus()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("58589aaab4ccc8eb41074b09")
  }
  shards:
    { "_id" : "shard0000", "host" : "127.0.0.1:27023" }
    { "_id" : "shard0001", "host" : "127.0.0.1:27024" }
  active mongoses:
    "3.2.9" : 1
  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:
mongos>
```

```

vagrant@vagrant-ubuntu-trusty-64:~$ ps -af | grep mongo
root      1964   1936   0 04:00 pts/0    00:00:00 sudo mongod --port 27022 --dbpath /db/c
nfig/data --configsvr --replSet config
root      1965   1964   1 04:00 pts/0    00:00:53 mongod --port 27022 --dbpath /db/config,
data --configsvr --replSet config
vagrant   2118   2085   0 04:03 pts/1    00:00:00 mongo --port 27022
vagrant   2297   2261   0 04:09 pts/3    00:00:25 mongos --configdb config/127.0.0.1:27022
--port 27021 --chunkSize 1
root      2615   2565   0 04:23 pts/4    00:00:00 sudo mongod --port 27023 --dbpath /db/sh
ard0/data
root      2616   2615   0 04:23 pts/4    00:00:24 mongod --port 27023 --dbpath /db/shard0,
data
root      2771   2747   0 04:27 pts/5    00:00:00 sudo mongod --port 27024 --dbpath /db/sh
ard1/data
root      2772   2771   0 04:27 pts/5    00:00:21 mongod --port 27024 --dbpath /db/shard1,
data
root      2908   2886   0 04:33 pts/7    00:00:00 sudo mongo 127.0.0.1:27021
root      2909   2908   0 04:33 pts/7    00:00:00 mongo 127.0.0.1:27021
vagrant   3211   3191   0 05:09 pts/8    00:00:00 grep --color=auto mongo
vagrant@vagrant-ubuntu-trusty-64:~$

```

6. Explicitly tell both database and collection that you want them to be sharded.

(a) Enabling a database for sharding is a prerequisite to sharding on of its collection

```

mongos> sh.enableSharding("testdb")
{ "ok" : 1 }

```

(b) The given collection is going to be split into chunks.

```

mongos> sh.shardCollection("testdb.testcollection", {testkey:1})
{ "collectionssharded" : "testdb.testcollection", "ok" : 1 }

```

7. Populate data in testdb.testCollection

username is not the shard key.

```
mongos> db = (new Mongo("localhost:27021")).getDB("testdb")
testdb
mongos> db.testcollection.count()
0
mongos> for (var i=0; i < 10; i++){
... db.testcollection.insert({"username": "user"+i})
... }
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 61,
    "errmsg" : "document { _id: ObjectId('5858c83eca20d545c504f3d2'), username: \"user9\" } does not contain shard key for pattern { testkey: 1.0 }"
  }
})
mongos> for (var i=0; i < 10; i++){ db.testcollection.insert({"testkey": "key"+i}) }
WriteResult({ "nInserted" : 1 })
mongos> db.testcollection.count()
10
mongos>
```

Let's insert 100,000 more documents.

```
mongos> for (var i = 0; i < 100000; i++){
... db.testcollection.insert({"testkey": "key"+i})
... }
WriteResult({ "nInserted" : 1 })
mongos>
```

8. Check the shard status (after populating data)

```
mongos> db.printShardingStatus()
--- Sharding Status ---
sharding version: {
  "_id" : 1,
  "minCompatibleVersion" : 5,
  "currentVersion" : 6,
  "clusterId" : ObjectId("58589aaab4ccc8eb41074b09")
}
shards:
  { "_id" : "shard0000", "host" : "127.0.0.1:27023" }
  { "_id" : "shard0001", "host" : "127.0.0.1:27024" }
active mongoses:
  "3.2.9" : 1
balancer:
  Currently enabled: yes
  Currently running: no
```

Failed balancer rounds in last 5 attempts: 0

Migration Results for the last 24 hours:

5 : Success

databases:

```
{ "_id" : "testdb", "primary" : "shard0001", "partitioned" : true }
```

testdb.testcollection

shard key: { "testkey" : 1 }

unique: false

balancing: true

chunks:

shard0000 5

shard0001 6

{ "testkey" : { "\$minKey" : 1 } } --> { "testkey" : "key0" } on : shard0000 Timestamp(2, 0)

{ "testkey" : "key0" } --> { "testkey" : "key19970" } on : shard0000 Timestamp(3, 0)

{ "testkey" : "key19970" } --> { "testkey" : "key30331" } on : shard0000 Timestamp(4, 0)

{ "testkey" : "key30331" } --> { "testkey" : "key41196" } on : shard0000 Timestamp(5, 0)

{ "testkey" : "key41196" } --> { "testkey" : "key5206" } on : shard0000 Timestamp(6, 0)

{ "testkey" : "key5206" } --> { "testkey" : "key5276" } on : shard0001 Timestamp(6, 1)

{ "testkey" : "key5276" } --> { "testkey" : "key6523" } on : shard0001 Timestamp(3, 4)

{ "testkey" : "key6523" } --> { "testkey" : "key76094" } on : shard0001 Timestamp(5, 2)

{ "testkey" : "key76094" } --> { "testkey" : "key8696" } on : shard0001 Timestamp(5, 3)

{ "testkey" : "key8696" } --> { "testkey" : "key9" } on : shard0001 Timestamp(5, 4)

{ "testkey" : "key9" } --> { "testkey" : { "\$maxKey" : 1 } } on : shard0001 Timestamp(1, 3)

9. Connect to the shards and see how many documents are stored in each shard.

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27021
MongoDB shell version: 3.2.9
connecting to: localhost:27021/test
mongos> use testdb
switched to db testdb
mongos> db.testcollection.count()
102567
```

Note: During population, I killed the loop execution after it partially created 2557 documents.

That's way the count reads 10 + 2557 + 100000.

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27023
MongoDB shell version: 3.2.9
connecting to: localhost:27023/test
Server has startup warnings:
2016-12-20T02:57:08.276+0000 I CONTROL [initandlisten] ** WARNING: You are running th
process as the root user, which is not recommended.
2016-12-20T02:57:08.278+0000 I CONTROL [initandlisten]
> use testdb
switched to db testdb
> db.testcollection.count()
48769
```

```

connecting to: localhost:27024/test
Server has startup warnings:
2016-12-20T03:00:36.492+0000 I CONTROL [initandlisten] ** WARNING: You are running
process as the root user, which is not recommended.
2016-12-20T03:00:36.493+0000 I CONTROL [initandlisten]
> use testdb
switched to db testdb
> db.testcollection.count()
53798
>

```

II. Adding a new shard to the cluster

1. Make a directory and launch mongod for the shard 2

```

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard2/data
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27025 --dbpath /db/shard2/data
2016-12-20T06:45:24.586+0000 I CONTROL [initandlisten] MongoDB starting : pid=5370 port
=27025 dbpath=/db/shard2/data 64-bit host=vagrant-ubuntu-trusty-64
2016-12-20T06:45:24.588+0000 I CONTROL [initandlisten] db version v3.2.9
2016-12-20T06:45:24.589+0000 I CONTROL [initandlisten] git version: 22ec9e93b40c85fc7ca
e7d56e7d6a02fd811088c
2016-12-20T06:45:24.590+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1f
6 Jan 2014
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] allocator: tcmalloc
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] modules: none
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] build environment:
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] distmod: ubuntu1404
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] distarch: x86_64
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] target_arch: x86_64
2016-12-20T06:45:24.592+0000 I CONTROL [initandlisten] options: { net: { port: 27025 },
storage: { dbPath: "/db/shard2/data" } }

```

2. Add the new shard to the cluster. Connect to mongos and addShard.

```

mongos> sh.addShard("localhost:27025")
{ "shardAdded" : "shard0002", "ok" : 1 }
mongos>

```

3. Check the shard status after adding shard 2.

```

mongos> db.printShardingStatus()
--- Sharding Status ---
sharding version: {
  "_id" : 1,
  "minCompatibleVersion" : 5,
  "currentVersion" : 6,
  "clusterId" : ObjectId("58589aaab4ccc8eb41074b09")
}
shards:

```



```
{ "_id" : "shard0000", "host" : "127.0.0.1:27023" }
{ "_id" : "shard0001", "host" : "127.0.0.1:27024" }
{ "_id" : "shard0002", "host" : "localhost:27025" }
```

active mongoses:

"3.2.9" : 1

balancer:

Currently enabled: yes

Currently running: no

Failed balancer rounds in last 5 attempts: 0

Migration Results for the last 24 hours:

8 : Success

databases:

```
{ "_id" : "testdb", "primary" : "shard0001", "partitioned" : true }
```

testdb.testcollection

shard key: { "testkey" : 1 }

unique: false

balancing: true

chunks:

shard0000 4

shard0001 4

shard0002 3

```
{ "testkey" : { "$minKey" : 1 } } --> { "testkey" : "key0" } on : shard0002 Timestamp(8, 0)
{ "testkey" : "key0" } --> { "testkey" : "key19970" } on : shard0000 Timestamp(8, 1)
{ "testkey" : "key19970" } --> { "testkey" : "key30331" } on : shard0000 Timestamp(4, 0)
{ "testkey" : "key30331" } --> { "testkey" : "key41196" } on : shard0000 Timestamp(5, 0)
{ "testkey" : "key41196" } --> { "testkey" : "key5206" } on : shard0000 Timestamp(6, 0)
{ "testkey" : "key5206" } --> { "testkey" : "key5276" } on : shard0002 Timestamp(7, 0)
{ "testkey" : "key5276" } --> { "testkey" : "key6523" } on : shard0002 Timestamp(9, 0)
{ "testkey" : "key6523" } --> { "testkey" : "key76094" } on : shard0001 Timestamp(9, 1)
{ "testkey" : "key76094" } --> { "testkey" : "key8696" } on : shard0001 Timestamp(5, 3)
{ "testkey" : "key8696" } --> { "testkey" : "key9" } on : shard0001 Timestamp(5, 4)
{ "testkey" : "key9" } --> { "testkey" : { "$maxKey" : 1 } } on shard0001 Timestamp(1, 3)
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27023
MongoDB shell version: 3.2.9
connecting to: localhost:27023/test
Server has startup warnings:
2016-12-20T06:37:00.348+0000 I CONTROL [initandlisten] ** WARNING
  the root user, which is not recommended.
2016-12-20T06:37:00.350+0000 I CONTROL [initandlisten]
> use testdb
switched to db testdb
> db.testcollection.count()
48769
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27024
MongoDB shell version: 3.2.9
connecting to: localhost:27024/test
Server has startup warnings:
2016-12-20T06:38:27.918+0000 I CONTROL [initandlisten] ** WARNING: You are running
the root user, which is not recommended.
2016-12-20T06:38:27.919+0000 I CONTROL [initandlisten]
> use testdb
switched to db testdb
> db.testcollection.count()
39018
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27025
MongoDB shell version: 3.2.9
connecting to: localhost:27025/test
Server has startup warnings:
2016-12-20T06:45:24.763+0000 I CONTROL [initandlisten] ** WARNING: You are running
the root user, which is not recommended.
2016-12-20T06:45:24.764+0000 I CONTROL [initandlisten]
> use testdb
switched to db testdb
> db.testcollection.count()
14780
```

III. Removing a shard from the cluster

Draining the removed shard process: mongos relocates the chunks on the target shard server to the other shard servers in the cluster.

```

connecting to: localhost:27021/test
mongos> use admin
switched to db admin
mongos> db.runCommand({removeShard:"localhost:27025"})
{
  "msg" : "draining started successfully",
  "state" : "started",
  "shard" : "shard0002",
  "note" : "you need to drop or movePrimary these databases",
  "dbsToMove" : [ ],
  "ok" : 1
}
mongos> db.runCommand({removeShard:"localhost:27025"})
{
  "msg" : "draining ongoing",
  "state" : "ongoing",
  "remaining" : {
    "chunks" : NumberLong(3),
    "dbs" : NumberLong(0)
  },
  "note" : "you need to drop or movePrimary these databases",
  "dbsToMove" : [ ],
  "ok" : 1
}
mongos> db.runCommand({removeShard:"localhost:27025"})
{
  "msg" : "removeshard completed successfully",
  "state" : "completed",
  "shard" : "shard0002",
  "ok" : 1
}
mongos>

```

```

mongos> db.runCommand({listshards:1})
{
  "shards" : [
    {
      "_id" : "shard0000",
      "host" : "127.0.0.1:27023"
    },
    {
      "_id" : "shard0001",
      "host" : "127.0.0.1:27024"
    }
  ],
  "ok" : 1
}
mongos>

```

Note: the corresponding daemon and directory should be manually killed and removed, respectively.

Note: To verify that a process is a [mongos](#)

```
mongos> db.runCommand({ isdbgrid:1 })
{ "isdbgrid" : 1, "hostname" : "vagrant-ubuntu-trusty-64", "ok" : 1 }
```

If it is not a mongos, an error is generated, for example as shown below:

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo localhost:27022
MongoDB shell version: 3.2.9
connecting to: localhost:27022/test
Server has startup warnings:
2016-12-20T06:36:24.087+0000 I CONTROL [initandlisten] ** WARNING: You
are running this process as the root user, which is not recommended.
2016-12-20T06:36:24.088+0000 I CONTROL [initandlisten]
config:PRIMARY> db.runCommand({isdbgrid:1})
{
  "ok" : 0,
  "errmsg" : "no such command: 'isdbgrid', bad cmd: '{ isdbgrid: 1.0 }'",
  "code" : 59
}
```