

# 1 准备

```
版本 : mongodb-linux-x86_64-rhel62-3.2.10.tgz
tar -xvf mongodb-linux-x86_64-rhel62-3.2.10.tgz
mv mongo-linux-x86_64-rhel62-3.2.10 mongo3.2
scp -r mongo3.2 mongo@nam1:/data/mongocluster
scp -r mongo3.2 mongo@name2:/data/mongocluster
scp -r mongo3.2 mongo@admin:/data/mongocluster
```

```
vi ~/.bashrc
```

```
MONGO=/data/mongocluster/mongodb3.0.13
export PATH=$PATH:$MONGO/bin
```

分别在下面 10.10.11.11 10.10.11.12 10.10.11.13 创建下面目录

```
mkdir -p /data/mongocluster/config
Mkdir -p /data/mongocluster/data/share1
Mkdir -p /data/mongocluster/data/share2
Mkdir -p /data/mongocluster/data/share3
```

在 10.10.11.13 节点还需创建

```
Mkdir -p /data/mongocluster/data/config-server
Mkdir -p /data/mongocluster/data/mongos
```

## 2 主机/进程/端口说明

No.	host	port mongod listens on			cfgsvr port	mongos port
		Share1	Share2	Share3		
1	10.10.11.11	20001	20002	20003		---
2	10.10.11.12	20001	20002	20003		---
3	10.10.11.13	20001	20002	20003	30000	40000

### 3 服务 说明

No.	name	role	hostname	port	dbPath
1	Shard 1	primary	10.10.11.11	20001	/data/mongocluster/data/share1
		secondary	10.10.11.12	20001	/data/mongocluster/data/share1
		arbiter	10.10.11.13	20001	/data/mongocluster/data/share1
2	Shard 2	primary	10.10.11.12	20002	/data/mongocluster/data/share2
		secondary	10.10.11.11	20002	/data/mongocluster/data/share2
		arbiter	10.10.11.13	20002	/data/mongocluster/data/share2
3	Shard 3	primary	10.10.11.13	20003	/data/mongocluster/data/share3
		secondary	10.10.11.12	20003	/data/mongocluster/data/share3
		arbiter	10.10.11.11	20003	/data/mongocluster/data/share3

### 4 配置文件

分别在 10.10.11.11 10.10.11.12 10.10.11.13 节点的/data/mongocluster/data/config 目录下 创建下面三个配置文件

Shard1.conf 如下

```
dbpath=/data/mongocluster/data/share1
directoryperdb = true
shardsvr = true
replSet = shard1
port = 20001
oplogSize = 100
pidfilepath = /data/mongocluster/data/share1/mongodb.pid
logpath = /data/mongocluster/data/share1/shard1.log
logappend = true
profile = 1
slowms = 5
fork = true
~
```

Shard2.conf.conf 如下

```
dbpath=/data/mongocluster/data/share2
directoryperdb = true
shardsvr = true
replSet = shard2
port = 20002
oplogSize = 100
```

```
pidfilepath = /data/mongocluster/data/share2/mongodb.pid
logpath = /data/mongocluster/data/share2/shard1.log
logappend = true
profile = 1
slowms = 5
fork = true
```

Shard3.conf 如下

```
dbpath=/data/mongocluster/data/share3
directoryperdb = true
shardsvr = true
replSet = shard3
port = 20003
oplogSize = 100
pidfilepath = /data/mongocluster/data/share3/mongodb.pid
logpath = /data/mongocluster/data/share3/shard3.log
logappend = true
profile = 1
slowms = 5
fork = true
```

在 10.10.11.13 （mongos ConfigSvr 服务） /data/mongocluster/config 创建下面脚本  
Config-server.sh mongos.sh

Config-server.sh 脚本如下

```
pidfilepath = /data/mongocluster/config-server/mongodb.pid
dbpath = /data/mongocluster/config-server
directoryperdb = true
configsvr = true
port = 20000
logpath = /data/mongocluster/config-server/config-server.log
logappend = true
fork = true
```

Mongos.sh 脚本如下

```
configdb = 10.10.11.13:20000
port = 40000
logpath = /data/mongocluster/mongos/mongos.log
logappend = true
fork = true
```

## 5 启动脚本

分别在 10.10.11.11 10.10.11.12 /data/mongocluster/config 目录下创建 start.sh

Start.sh 脚本如下

```
#!/bin/bash
mongod -f /data/mongocluster/config/share1.conf
mongod -f /data/mongocluster/config/share2.conf
mongod -f /data/mongocluster/config/share3.conf
```

在 10.10.11.13 (mongos configSvr) /data/mongocluster/config 目录下创建 start.sh

```
#!/bin/bash
mongod -f /data/mongocluster/config/share1.conf
mongod -f /data/mongocluster/config/share2.conf
mongod -f /data/mongocluster/config/share3.conf
mongod -f /data/mongocluster/config/config-server.conf
sleep 10s
mongos -f /data/mongocluster/config/mongos.conf
```

## 6 设置路由，分片

分别在 10.10.11.11 10.10.11.12 10.10.11.13 执行下面命令

```
bash /data/mongocluster/config/start.sh
```

验证服务已经启动

```
ps -aux | grep mongo
```

```
mongo 85517 0.0 0.0 103248 888 pts/0 S+ 09:58 0:00 grep mongo
mongo 108864 0.0 0.1 549860 60684 ? S1 Mar28 0:56 mongod -f /data/mongocluster/config/share1.conf
mongo 108923 0.0 0.1 548832 61408 ? S1 Mar28 0:55 mongod -f /data/mongocluster/config/share2.conf
mongo 109106 0.4 1.6 2161816 553320 ? S1 Mar28 5:42 mongod -f /data/mongocluster/config/share3.conf
```

(10.10.11.11 10.10.11.12)

```
mongo 85517 0.0 0.0 103248 888 pts/0 S+ 09:58 0:00 grep mongo
mongo 108864 0.0 0.1 549860 60684 ? S1 Mar28 0:56 mongod -f /data/mongocluster/config/share1.conf
mongo 108923 0.0 0.1 548832 61408 ? S1 Mar28 0:55 mongod -f /data/mongocluster/config/share2.conf
mongo 109106 0.4 1.6 2161816 553320 ? S1 Mar28 5:42 mongod -f /data/mongocluster/config/share3.conf
mongo 109200 0.0 0.2 506912 84428 ? S1 Mar28 1:10 mongod -f /data/mongocluster/config/config-server.conf
mongo 109330 0.1 0.0 265260 9816 ? S1 Mar28 2:40 mongos -f /data/mongocluster/config/mongos.conf
```

(10.10.11.13)

### 6.2 配置分片

在 10.10.11.11 节点上

执行 mongo --host 10.10.11.11:20001 连接 shard1 的端口

注意：特别注意的是，对于仲裁节点，需要有个特别的配置——`arbiterOnly:true`。这个千万不能少了，不然主备模式就不能生效。

```
use admin
config = { _id:"shard1",members:[
    { id: 0, host:"10.10.11.11:20001",priority:1},
    { id: 1, host:"10.10.11.12:20001",priority:2},
    { id: 2, host:"10.10.11.13:20001",arbiterOnly:true },
]
};
rs.initiate(config)
rs.status()
```

```
shard1:SECONDARY> rs.status()
{
  "set" : "shard1",
  "date" : ISODate("2017-03-29T05:13:38.681Z"),
  "myState" : 2,
  "term" : NumberLong(2),
  "syncingTo" : "10.10.11.12:20001",
  "heartbeatIntervalMillis" : NumberLong(2000),
  "members" : [
    {
      "_id" : 0,
      "name" : "10.10.11.11:20001",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 377,
      "optime" : {
        "ts" : Timestamp(1490764404, 2),
        "t" : NumberLong(2)
      },
      "optimeDate" : ISODate("2017-03-29T05:13:24Z"),
      "syncingTo" : "10.10.11.12:20001",
      "configVersion" : 1,
      "self" : true
    },
    {
      "_id" : 1,
      "name" : "10.10.11.12:20001",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 38,
      "optime" : {
        "ts" : Timestamp(1490764404, 2),
        "t" : NumberLong(2)
      },
      "optimeDate" : ISODate("2017-03-29T05:13:24Z"),
      "lastHeartbeat" : ISODate("2017-03-29T05:13:38.480Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:13:38.408Z"),
      "pingMs" : NumberLong(0),
      "electionTime" : Timestamp(1490764404, 1),
      "electionDate" : ISODate("2017-03-29T05:13:24Z"),
      "configVersion" : 1
    },
    {
      "_id" : 2,
      "name" : "10.10.11.13:20001",
      "health" : 1,
      "state" : 7,
      "stateStr" : "ARBITER",
      "uptime" : 38,
      "lastHeartbeat" : ISODate("2017-03-29T05:13:38.480Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:13:37.417Z"),
      "pingMs" : NumberLong(0),
      "configVersion" : 1
    }
  ],
  "ok" : 1
}
```

执行 `mongo --host 10.10.11.11:20002` 连接 shard2 的端口

```

use admin
config = { _id:"shard2",members:[
    { _id: 0, host:"10.10.11.11:20002",priority:2},
    { _id: 1, host:"10.10.11.12:20002",priority:1},
    { _id: 2, host:"10.10.11.13:20002",arbiterOnly:true},
]
};
rs.initiate(config)
rs.status()

```

```

shard2:PRIMARY> rs.status()
{
  "set" : "shard2",
  "date" : ISODate("2017-03-29T05:21:37.087Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "members" : [
    {
      "_id" : 0,
      "name" : "10.10.11.11:20002",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 16,
      "optime" : {
        "ts" : Timestamp(1490764892, 2),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2017-03-29T05:21:32Z"),
      "lastHeartbeat" : ISODate("2017-03-29T05:21:36.562Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:21:34.637Z"),
      "pingMs" : NumberLong(0),
      "syncingTo" : "10.10.11.12:20002",
      "configVersion" : 1
    },
    {
      "_id" : 1,
      "name" : "10.10.11.12:20002",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 846,
      "optime" : {
        "ts" : Timestamp(1490764892, 2),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2017-03-29T05:21:32Z"),
      "infoMessage" : "could not find member to sync from",
      "electionTime" : Timestamp(1490764892, 1),
      "electionDate" : ISODate("2017-03-29T05:21:32Z"),
      "configVersion" : 1,
      "self" : true
    },
    {
      "_id" : 2,
      "name" : "10.10.11.13:20002",
      "health" : 1,
      "state" : 7,
      "stateStr" : "ARBITER",
      "uptime" : 16,
      "lastHeartbeat" : ISODate("2017-03-29T05:21:36.562Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:21:32.747Z"),
      "pingMs" : NumberLong(0),
      "configVersion" : 1
    }
  ],
  "ok" : 1
}

```

执行 `mongo --host 10.10.11.11:20003` 连接 shard3 的端口

```

use admin
config = { _id:"shard3",members:[
    { _id: 0, host:"10.10.11.11:20003",arbiterOnly:true},
    { _id: 1, host:"10.10.11.12:20003",priority:2},
    { _id: 2, host:"10.10.11.13:20003",priority:1},
]
};

```

```

    ]
};
rs.initiate(config)
rs.status()
shard3:PRIMARY> rs.status()
{
  "set" : "shard3",
  "date" : ISODate("2017-03-29T05:22:13.450Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "members" : [
    {
      "_id" : 0,
      "name" : "10.10.11.11:20003",
      "health" : 1,
      "state" : 7,
      "stateStr" : "ARBITER",
      "uptime" : 19,
      "lastHeartbeat" : ISODate("2017-03-29T05:22:13.105Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:22:10.308Z"),
      "pingMs" : NumberLong(0),
      "configVersion" : 1
    },
    {
      "_id" : 1,
      "name" : "10.10.11.12:20003",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 19,
      "optime" : {
        "ts" : Timestamp(1490764925, 2),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2017-03-29T05:22:05Z"),
      "lastHeartbeat" : ISODate("2017-03-29T05:22:13.104Z"),
      "lastHeartbeatRecv" : ISODate("2017-03-29T05:22:12.318Z"),
      "pingMs" : NumberLong(0),
      "syncingTo" : "10.10.11.13:20003",
      "configVersion" : 1
    },
    {
      "_id" : 2,
      "name" : "10.10.11.13:20003",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 705,
      "optime" : {
        "ts" : Timestamp(1490764925, 2),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2017-03-29T05:22:05Z"),
      "infoMessage" : "could not find member to sync from",
      "electionTime" : Timestamp(1490764925, 1),
      "electionDate" : ISODate("2017-03-29T05:22:05Z"),
      "configVersion" : 1,
      "self" : true
    }
  ],
  "ok" : 1
}

```

## 6.3 配置路由

在 10.10.11.13 节点 执行 如下命令

```

mongo --port 40000
use admin
db.runCommand({addshard:"shard1/10.10.11.11:20001,10.10.11.12:20001,10.10.11.13:20001",
name:"shard1",maxsize:20480});
db.runCommand({addshard:"shard2/10.10.11.11:20002,10.10.11.12:20002,10.10.11.13:20002",
name:"shard2",maxsize:20480});

```

```
db.runCommand({addshard:"shard3/10.10.11.11:20003,10.10.11.12:20003,10.10.11.13:20003",
name:"shard3",maxsize:20480});
db.runCommand({listshards : 1});
```

```
mongos> db.runCommand({listshards : 1});
{
  "shards" : [
    {
      "_id" : "shard1",
      "host" : "shard1/10.10.11.11:20001,10.10.11.12:20001"
    },
    {
      "_id" : "shard2",
      "host" : "shard2/10.10.11.11:20002,10.10.11.12:20002"
    },
    {
      "_id" : "shard3",
      "host" : "shard3/10.10.11.12:20003,10.10.11.13:20003"
    }
  ],
  "ok" : 1
}
```

```
printShardingStatus()
```

```
mongos> printShardingStatus()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("58db41cfb715665c6ccb9a80")
  }
  shards:
    { "_id" : "shard1", "host" : "shard1/10.10.11.11:20001,10.10.11.12:20001" }
    { "_id" : "shard2", "host" : "shard2/10.10.11.11:20002,10.10.11.12:20002" }
    { "_id" : "shard3", "host" : "shard3/10.10.11.12:20003,10.10.11.13:20003" }
  active mongoses:
    "3.2.10" : 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:
```

## 7 创建 test 库 users 表 启动分片

连接上 mongos 对外路由

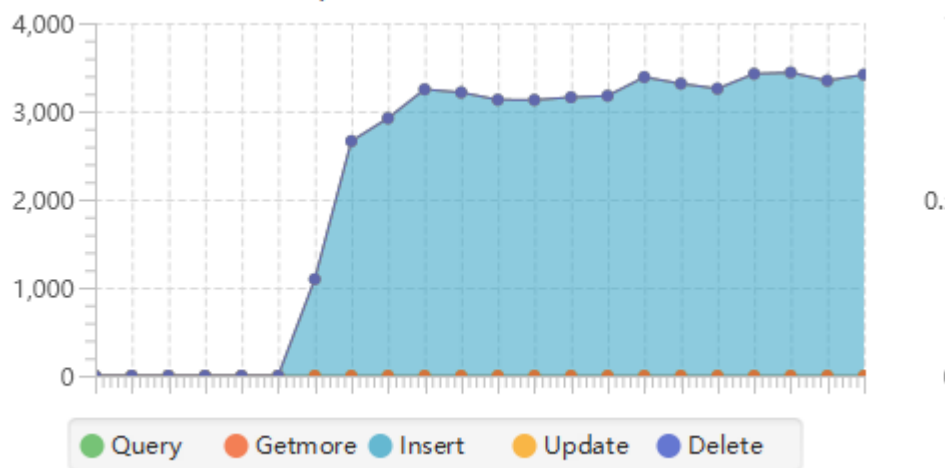
```
mongos --host 10.10.11.13 --port 4000
Use test
db.createCollection("users")
Use admin
db.runCommand({enablesharding:"test"});
b.runCommand({shardcollection:"test.users",key:{id:1}})
```



## 8 性能测试

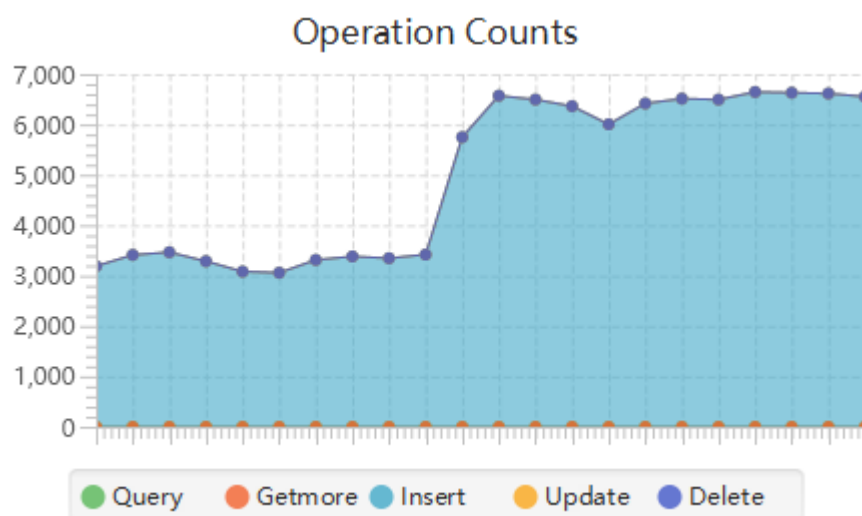
### 8.1 单线程 insert Value < 1K

插入 200W 条数据



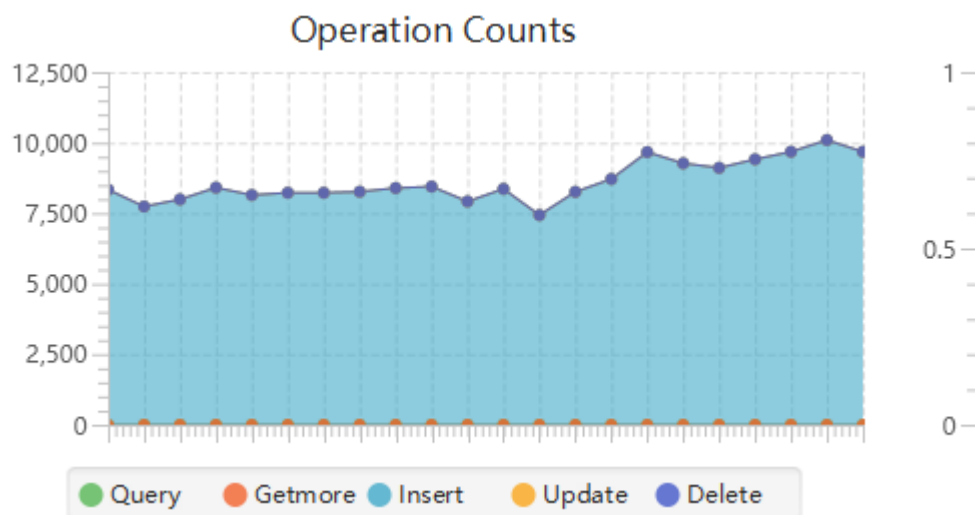
单线程 3 机器 3 个分片 大约 3500 Opts

### 8.2 两个线程 Insert 分别 200W



2 个线程 3 个机器 3 个分片 大约 接近 7 千 ops

### 8.3 四个线程 Insert 分别 200W



4 个线程 3 个机器 3 个分片 大约解决 1 万 1 千 ops

机器 未出现内存暴涨 Swap 上升 整体还算稳定 有关读需要代码进行测试