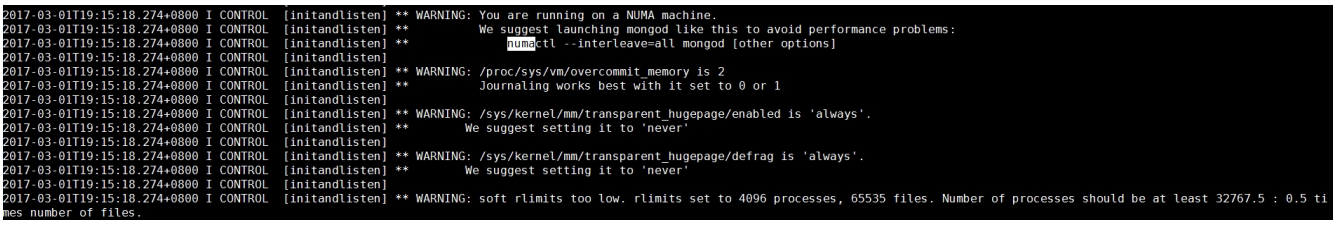
MongoDB

MongoDB

warning

mongodwarning



1mongodnuma
numactl --interleave=all ./bin/mongod -config ./conf/mongod.conf
2/proc/sys/vm/overcommit_memory
echo 1 > /proc/sys/vm/overcommit_memory
3
echo never >> /sys/kernel/mm/transparent_hugepage/enabled
echo never >> /sys/kernel/mm/transparent_hugepage/defrag
4processes
ulimit -u 32767/etc/security/limits.conf

1https://www.mongodb.com/download-center?jmp=nav#community
2tar -zxvf mongodb-linux-x86_64-amazon-3.2.12.tgz
3confmongod.conf

```
dbpath=/data/volume_b/mongodb/data/db
logpath=/data/volume_b/mongodb/logs/mongodb.log
pidfilepath=/data/volume_b/mongodb/logs/mongod.pid
port=27017
fork=true
logappend=true
journal=true
```

4dblogsmongodb
5mongodb

```
numactl --interleave=all ./bin/mongod -config ./conf/mongod.conf
```

1https://www.mongodb.com/download-center?jmp=nav#community
2tar -zxvf mongodb-linux-x86_64-amazon-3.2.12.tgz
3confmongod.conf

```
dbpath=/data/volume_b/mongodb/data/db
logpath=/data/volume_b/mongodb/logs/mongodb.log
pidfilepath = /data/volume_b/mongodb/logs/mongod.pid
port=27017
fork=true
logappend=true
journal=true
quiet=true
replSet=fbj
```

4dblogsmongodb
5mongodb

```
numactl --interleave=all ./bin/mongod -config ./conf/mongod.conf
```

1Mongo

```
./mongo

> use admin
switched to db admin

>
config=({_id:"fbj",members:[{_id:0,host:"192.168.20.100:27017"},{_id:1,host:"192.168.20.103:27017"},{_id
:2,host:"192.168.20.104:27017"}]})
{
        "_id" : "fbj",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "192.168.20.102:27017"
                },
                {
                        "_id" : 1,
                        "host" : "192.168.20.103:27017"
                },
                {
                        "_id" : 2,
                        "host" : "192.168.20.104:27017"
                }
        ]
}

> rs.initiate(config)
{ "ok" : 1 }
```

2

```
fbj:PRIMARY>rs.status()

fbj:PRIMARY> con = new Mongo("localhost:27017")
connection to localhost:27017

fbj:PRIMARY> priDB = con.getDB("test")
test

fbj:PRIMARY> priDB.isMaster()
{
        "hosts" : [
                "192.168.20.102:27017",
                "192.168.20.103:27017",
                "192.168.20.104:27017"
        ],
        "setName" : "fbj",
        "setVersion" : 1,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "192.168.20.102:27017",
        "me" : "192.168.20.102:27017",
        "electionId" : ObjectId("7fffffff0000000000000001"),
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 1000,
        "localTime" : ISODate("2017-04-11T02:31:12.112Z"),
        "maxWireVersion" : 4,
        "minWireVersion" : 0,
        "ok" : 1
}

fbj:PRIMARY> for(i=0;i<1000;i++){priDB.coll.insert({count:i})}
WriteResult({ "nInserted" : 1 })

fbj:PRIMARY> priDB.coll.count()
1000

fbj:PRIMARY> use test
switched to db test

fbj:PRIMARY> show tables
coll

fbj:PRIMARY> db.coll.find()
{ "_id" : ObjectId("58ec402c14ae530934f46135"), "count" : 0 }
{ "_id" : ObjectId("58ec402c14ae530934f46136"), "count" : 1 }
{ "_id" : ObjectId("58ec402c14ae530934f46137"), "count" : 2 }
{ "_id" : ObjectId("58ec402c14ae530934f46138"), "count" : 3 }
{ "_id" : ObjectId("58ec402c14ae530934f46139"), "count" : 4 }
{ "_id" : ObjectId("58ec402c14ae530934f4613a"), "count" : 5 }
{ "_id" : ObjectId("58ec402c14ae530934f4613b"), "count" : 6 }
{ "_id" : ObjectId("58ec402c14ae530934f4613c"), "count" : 7 }
{ "_id" : ObjectId("58ec402c14ae530934f4613d"), "count" : 8 }
{ "_id" : ObjectId("58ec402c14ae530934f4613e"), "count" : 9 }
{ "_id" : ObjectId("58ec402c14ae530934f4613f"), "count" : 10 }
{ "_id" : ObjectId("58ec402c14ae530934f46140"), "count" : 11 }
{ "_id" : ObjectId("58ec402c14ae530934f46141"), "count" : 12 }
{ "_id" : ObjectId("58ec402c14ae530934f46142"), "count" : 13 }
{ "_id" : ObjectId("58ec402c14ae530934f46143"), "count" : 14 }
{ "_id" : ObjectId("58ec402c14ae530934f46144"), "count" : 15 }
{ "_id" : ObjectId("58ec402c14ae530934f46145"), "count" : 16 }
{ "_id" : ObjectId("58ec402c14ae530934f46146"), "count" : 17 }
{ "_id" : ObjectId("58ec402c14ae530934f46147"), "count" : 18 }
{ "_id" : ObjectId("58ec402c14ae530934f46148"), "count" : 19 }
Type "it" for more
```

```
./mongo

fbj:SECONDARY> show dbs
2017-04-11T10:35:51.680+0800 E QUERY    [thread1] Error: listDatabases failed:{ "ok" : 0, "errmsg" :
"not master and slaveOk=false", "code" : 13435 } :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:62:1
shellHelper.show@src/mongo/shell/utils.js:761:19
shellHelper@src/mongo/shell/utils.js:651:15
@(shellhelp2):1:1

fbj:SECONDARY> db.getMongo().setSlaveOk();(use adminrs.slaveOk())
fbj:SECONDARY> show dbs
local  0.000GB
test   0.000GB

fbj:SECONDARY> use test
switched to db test
fbj:SECONDARY> show tables
coll
fbj:SECONDARY> db.coll.find()
{ "_id" : ObjectId("58ec402c14ae530934f46135"), "count" : 0 }
{ "_id" : ObjectId("58ec402c14ae530934f46136"), "count" : 1 }
{ "_id" : ObjectId("58ec402c14ae530934f46137"), "count" : 2 }
{ "_id" : ObjectId("58ec402c14ae530934f46138"), "count" : 3 }
{ "_id" : ObjectId("58ec402c14ae530934f4613a"), "count" : 5 }
{ "_id" : ObjectId("58ec402c14ae530934f46139"), "count" : 4 }
{ "_id" : ObjectId("58ec402c14ae530934f4613b"), "count" : 6 }
{ "_id" : ObjectId("58ec402c14ae530934f4613d"), "count" : 8 }
{ "_id" : ObjectId("58ec402c14ae530934f4613c"), "count" : 7 }
{ "_id" : ObjectId("58ec402c14ae530934f4613e"), "count" : 9 }
{ "_id" : ObjectId("58ec402c14ae530934f4613f"), "count" : 10 }
{ "_id" : ObjectId("58ec402c14ae530934f46141"), "count" : 12 }
{ "_id" : ObjectId("58ec402c14ae530934f46140"), "count" : 11 }
{ "_id" : ObjectId("58ec402c14ae530934f46144"), "count" : 15 }
{ "_id" : ObjectId("58ec402c14ae530934f46143"), "count" : 14 }
{ "_id" : ObjectId("58ec402c14ae530934f46145"), "count" : 16 }
{ "_id" : ObjectId("58ec402c14ae530934f46142"), "count" : 13 }
{ "_id" : ObjectId("58ec402c14ae530934f46146"), "count" : 17 }
{ "_id" : ObjectId("58ec402c14ae530934f46147"), "count" : 18 }
{ "_id" : ObjectId("58ec402c14ae530934f46148"), "count" : 19 }
Type "it" for more

4

> use ycsb

//
> db.usertable.ensureIndex({field0:1})
{
        "createdCollectionAutomatically" : true,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

1https://www.mongodb.com/download-center?jmp=nav#community
2tar -zxvf mongodb-linux-x86_64-amazon-3.2.12.tgz
3conf

mongod1.conf

```
dbpath=/data/volume_b/mongodb/data/db1
logpath=/data/volume_b/mongodb/logs/mongodb1.log
pidfilepath = /data/volume_b/mongodb/logs/mongod1.pid
port=27017
fork=true
logappend=true
journal=true
quiet=true
replSet=fbj1
oplogSize=102400
```

mongod2.conf

```
dbpath=/data/volume_b/mongodb/data/db2
logpath=/data/volume_b/mongodb/logs/mongodb2.log
pidfilepath = /data/volume_b/mongodb/logs/mongod2.pid
port=27018
fork=true
logappend=true
journal=true
quiet=true
replSet=fbj2
oplogSize=102400
```

mongod3.conf

```
dbpath=/data/volume_b/mongodb/data/db3
logpath=/data/volume_b/mongodb/logs/mongodb3.log
pidfilepath = /data/volume_b/mongodb/logs/mongod3.pid
port=27019
fork=true
logappend=true
journal=true
quiet=true
replSet=fbj3
oplogSize=102400
```

config.conf

```
dbpath=/data/volume_b/mongodb/data/config
logpath=/data/volume_b/mongodb/logs/config.log
pidfilepath = /data/volume_b/mongodb/logs/config.pid
port=27027
fork=true
logappend=true
journal=true
quiet=true
```

mongos.conf

```
logpath=/data/volume_b/mongodb/logs/mongos.log
pidfilepath = /data/volume_b/mongodb/logs/mongos.pid
port=27037
fork=true
configdb=192.168.20.100:27027,192.168.20.103:27027,192.168.20.104:27027
```

4dblogsmongodb
5

```
numactl --interleave=all ./bin/mongod -config ./conf/mongod1.conf --setParameter
replWriterThreadCount=32
numactl --interleave=all ./bin/mongod -config ./conf/mongod2.conf --setParameter
replWriterThreadCount=32
numactl --interleave=all ./bin/mongod -config ./conf/mongod3.conf --setParameter
replWriterThreadCount=32
```

6

```
./mongo 192.168.20.100:27017

> use admin
switched to db admin

>
config=({_id:"fbj1",members:[{_id:0,host:"192.168.20.100:27017"},{_id:1,host:"192.168.20.103:27017"},{_i
d:2,host:"192.168.20.104:27017"}]})
{
        "_id" : "fbj1",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "192.168.20.100:27017"
                },
                {
                        "_id" : 1,
                        "host" : "192.168.20.103:27017"
                },
                {
                        "_id" : 2,
                        "host" : "192.168.20.104:27017"
                }
        ]
}

> rs.initiate(config)
{ "ok" : 1 }

./mongo 192.168.20.103:27018

> use admin
switched to db admin

>
config=({_id:"fbj2",members:[{_id:0,host:"192.168.20.100:27018"},{_id:1,host:"192.168.20.103:27018"},{_i
d:2,host:"192.168.20.104:27018"}]})
{
        "_id" : "fbj2",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "192.168.20.100:27018"
                },
                {
                        "_id" : 1,
                        "host" : "192.168.20.103:27018"
                },
                {
                        "_id" : 2,
                        "host" : "192.168.20.104:27018"
                }
        ]
}

> rs.initiate(config)
{ "ok" : 1 }
```

```
./mongo 192.168.20.104:27019

> use admin
switched to db admin

>
config=({_id:"fbj3",members:[{_id:0,host:"192.168.20.100:27019"},{_id:1,host:"192.168.20.103:27019"},{_i
d:2,host:"192.168.20.104:27019"}]})
{
        "_id" : "fbj3",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "192.168.20.100:27019"
                },
                {
                        "_id" : 1,
                        "host" : "192.168.20.103:27019"
                },
                {
                        "_id" : 2,
                        "host" : "192.168.20.104:27019"
                }
        ]
}

> rs.initiate(config)
{ "ok" : 1 }
```

7configdb

```
numactl --interleave=all ./bin/mongod --configsvr -config ./conf/config.conf
```

8mongos

```
numactl --interleave=all ./bin/mongos -config ./conf/mongos.conf
```

9mongos

```
./bin/mongo 192.168.20.100:27037

mongos> use admin
switched to db admin
mongos> db.runCommand( { addshard :
"fbj1/192.168.20.100:27017,192.168.20.103:27017,192.168.20.104:27017"});
{ "shardAdded" : "fbj1", "ok" : 1 }
mongos> db.runCommand( { addshard :
"fbj2/192.168.20.100:27018,192.168.20.103:27018,192.168.20.104:27018"});
{ "shardAdded" : "fbj2", "ok" : 1 }
mongos> db.runCommand( { addshard :
"fbj3/192.168.20.100:27019,192.168.20.103:27019,192.168.20.104:27019"});
{ "shardAdded" : "fbj3", "ok" : 1 }
mongos> db.runCommand( { listshards : 1 } );
{
        "shards" : [
                {
                        "_id" : "fbj1",
                        "host" : "fbj1/192.168.20.100:27017,192.168.20.103:27017,192.168.20.104:27017"
                },
                {
                        "_id" : "fbj2",
                        "host" : "fbj2/192.168.20.100:27018,192.168.20.103:27018,192.168.20.104:27018"
                },
                {
                        "_id" : "fbj3",
                        "host" : "fbj3/192.168.20.100:27019,192.168.20.103:27019,192.168.20.104:27019"
                }
        ],
        "ok" : 1
}




./bin/mongo 192.168.20.100:27037

mongos> use admin
switched to db admin
```

```
mongos> db.runCommand( { enablesharding :"testdb"});
{ "ok" : 1 }
mongos> db.runCommand( { shardcollection : "testdb.table1",key : {id: 1} } )
{ "collectionsharded" : "testdb.table1", "ok" : 1 }

mongos> db.printShardingStatus()    //
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("58f42ee684fe49830067fdad")
}
  shards:
        {  "_id" : "fbj1",  "host" :
"fbj1/192.168.20.100:27017,192.168.20.103:27017,192.168.20.104:27017" }
        {  "_id" : "fbj2",  "host" :
"fbj2/192.168.20.100:27018,192.168.20.103:27018,192.168.20.104:27018" }
        {  "_id" : "fbj3",  "host" :
"fbj3/192.168.20.100:27019,192.168.20.103:27019,192.168.20.104:27019" }
  active mongoses:
        "3.2.12" : 3
  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" : "testdb",  "primary" : "fbj1",  "partitioned" : true }
                testdb.table1
                        shard key: { "id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                fbj1    1
                        { "id" : { "$minKey" : 1 } } -->> { "id" : { "$maxKey" : 1 } } on : fbj1
Timestamp(1, 0)

mongos> for (var i = 1; i <= 10000; i++)db.table1.save({id:i,"test1":"testval1"});
WriteResult({ "nInserted" : 1 })
mongos> db.table1.stats();

mongos> db.runCommand( { enablesharding :"ycsb"});
{ "ok" : 1 }
mongos> db.runCommand( { shardcollection : "ycsb.usertable",key : {_id: "hashed"} } )
{ "collectionsharded" : "ycsb.usertable", "ok" : 1 }

mongos> db.printShardingStatus()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("58f42ee684fe49830067fdad")
}
  shards:
        {  "_id" : "fbj1",  "host" :
"fbj1/192.168.20.100:27017,192.168.20.103:27017,192.168.20.104:27017" }
        {  "_id" : "fbj2",  "host" :
"fbj2/192.168.20.100:27018,192.168.20.103:27018,192.168.20.104:27018" }
        {  "_id" : "fbj3",  "host" :
"fbj3/192.168.20.100:27019,192.168.20.103:27019,192.168.20.104:27019" }
  active mongoses:
        "3.2.12" : 3
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  1
        Last reported error:  Couldn't get a connection within the time limit
        Time of Reported error:  Tue Apr 18 2017 10:15:11 GMT+0800 (CST)
        Migration Results for the last 24 hours:
                106 : Success
                7 : Failed with error 'data transfer error', from fbj3 to fbj2
                2 : Failed with error 'data transfer error', from fbj3 to fbj1
                9 : Failed with error 'chunk too big to move', from fbj3 to fbj1
                10 : Failed with error 'chunk too big to move', from fbj3 to fbj2
```

```
                89 : Failed with error 'aborted', from fbj3 to fbj2
                7904 : Failed with error 'aborted', from fbj3 to fbj1
  databases:
        {  "_id" : "testdb",  "primary" : "fbj1",  "partitioned" : true }
                testdb.table1
                        shard key: { "id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                fbj1    1
                                fbj2    1
                                fbj3    1
                        { "id" : { "$minKey" : 1 } } -->> { "id" : 2 } on : fbj2 Timestamp(2, 0)
                        { "id" : 2 } -->> { "id" : 20 } on : fbj3 Timestamp(3, 0)
                        { "id" : 20 } -->> { "id" : { "$maxKey" : 1 } } on : fbj1 Timestamp(3, 1)
        {  "_id" : "ycsb",  "primary" : "fbj3",  "partitioned" : true }
                ycsb.usertable
                        shard key: { "_id" : "hashed" }
                        unique: false
                        balancing: true
                        chunks:
                                fbj1    2
                                fbj2    2
                                fbj3    2
                        { "_id" : { "$minKey" : 1 } } -->> { "_id" : NumberLong("-6148914691236517204")
} on : fbj1 Timestamp(3, 2)
                        { "_id" : NumberLong("-6148914691236517204") } -->> { "_id" :
NumberLong("-3074457345618258602") } on : fbj1 Timestamp(3, 3)
                        { "_id" : NumberLong("-3074457345618258602") } -->> { "_id" : NumberLong(0) } on
: fbj2 Timestamp(3, 4)
                        { "_id" : NumberLong(0) } -->> { "_id" : NumberLong("3074457345618258602") } on
: fbj2 Timestamp(3, 5)
                        { "_id" : NumberLong("3074457345618258602") } -->> { "_id" :
NumberLong("6148914691236517204") } on : fbj3 Timestamp(3, 6)
                        { "_id" : NumberLong("6148914691236517204") } -->> { "_id" : { "$maxKey" : 1 } }
on : fbj3 Timestamp(3, 7)
```