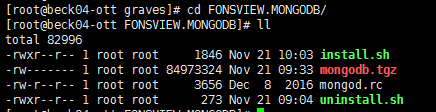
MongoDB 单机安装：

1.进入目录：cd /root/install

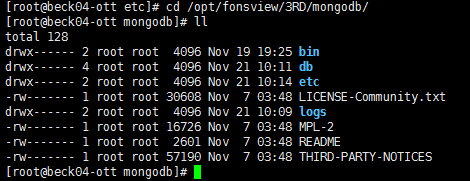
2.解压文件：tar -zxvf fonsview-mongodb-single.tgz

3.进入解压得到的文件夹：cd fonsview-mongodb-single



1. 执行安装脚本：./install.sh

默认安装位置： /opt/fonsview/3RD/mongodb



db: 存放数据

etc: 存放配置文件

logs：存放日志

1. 启动mongodb： service mongod start
2. 关闭mongodb: service mongod stop
3. 其他相关操作:

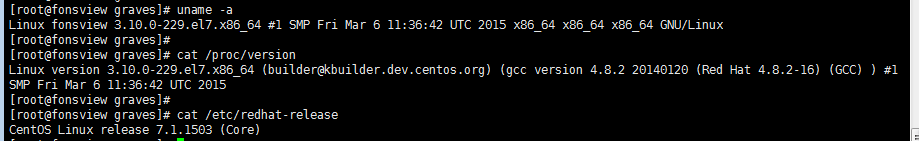
查看状态： service mongod status

重启： service mongod restart

MongoDB 集群安装：

1. 环境准备：

系统：



服务器：3台（172.16.6.92,172.16.6.95,172.16.6.103）

安装包：fonsview-mongodb-cluster.tgz

|  |  |  |
| --- | --- | --- |
| 服务器A（172.16.6.92） | 服务器B（172.16.6.95） | 服务器C（172.16.6.103） |
| mongos | mongos | mongos |
| config server | config server | config server |
| shard server1 主节点 | shard server1 副节点 | shard server1 仲裁 |
| shard server2 仲裁 | shard server2 主节点 | shard server2 副节点 |
| shard server3 副节点 | shard server3 仲裁 | shard server3 主节点 |

端口分配:

mongos:23000 （DB连接端口） config:24000 shard1:25001 shard2:25002 shard3:25003

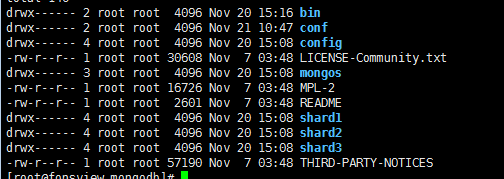
1. 安装mongodb：

进入默认目录: cd /opt/fonsview/3RD

将安装包放在默认目录下解压: tar -zxvf fonsview-mongodb-cluster.tgz

1. 分别在每台机器建立conf、mongos、config、shard1、shard2、shard3六个目录，因为mongos不存储数据，只需要建立日志文件目录即可

mkdir -p /opt/fonsview/3RD/mongodb/conf  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/mongos/log  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/config/data  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/config/log  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard1/data  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard1/log  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard2/data  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard2/log  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard3/data  
mkdir -p /mnt/vdb/opt/fonsview/3RD/mongodb/shard3/log



conf：存放所有服务器的配置文件

config：存放配置服务器的数据和日志信息

mongos：存放路由服务器的日志信息

shard1/shard2/shard3：存放各个分片服务器上的数据和日志信息

1. 配置服务器搭建副本集：

Mongodb3.4以后要求配置服务器也创建副本集，不然集群搭建不成功。

1. 添加配置文件:

vim /opt/fonsview/3RD/mongodb/conf/config.conf

1. 进入配置文件后添加如下信息：

## 配置文件内容  
pidfilepath=/opt/fonsview/3RD/mongodb/config/log/configsrv.pid  
dbpath=/opt/fonsview/3RD/mongodb/config/data  
logpath=/opt/fonsview/3RD/mongodb/config/log/configsrv.log  
logappend=true  
bind\_ip=0.0.0.0  
port=24000  
fork=true  
#declare this is a config db of a cluster;  
configsvr=true  
#副本集名称  
replSet=configs  
#设置最大连接数  
maxConns=20000

1. 分别启动三台服务器的config server，进入 /opt/fonsview/3RD/mongdb/bin目录下

./mongod -f /opt/fonsview/3RD/mongodb/conf/config.conf

1. 登录任意一台服务器A，初始化配置副本集, 进入/opt/fonsview/3RD/mongdb/bin 目录下

./mongo --port 24000

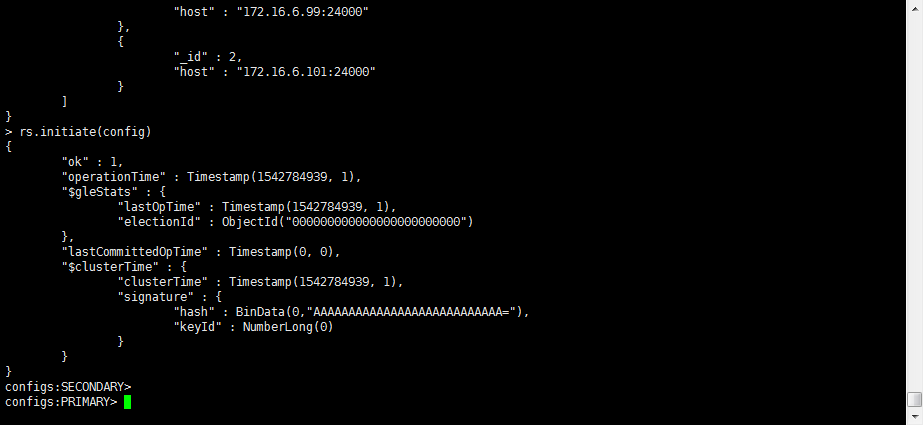
使用admin数据库,配置config变量，并初始化:

use admin;

config = {  
    \_id : "configs",  
     members : [  
         {\_id : 0, host : "172.16.6.92:24000" },  
         {\_id : 1, host : "172.16.6.95:24000" },  
         {\_id : 2, host : "172.16.6.103:24000" }  
     ]  
 };

rs.initiate(config)

初始化成功（此时 > 前已经出现 PRIMARY（主节点））：



其中的 \_id:”configs” 应与配置文件中的 replSet=congfigs 一致，host 为三个节点的 ip 和 port

如果初始化失败，会导致路由服务器和配置服务器连接不上。

1. 分片服务器搭建副本集：

配置分片副本集（三台服务器）

**设置第一个副本集：**

1. 添加配置文件:

vim /opt/fonsview/3RD/mongodb/conf/shard1.conf

1. 进入配置文件后添加如下信息：

## 配置文件内容  
pidfilepath = /opt/fonsview/3RD/mongodb/shard1/log/shard1.pid  
dbpath=/opt/fonsview/3RD/mongodb/shard1/data  
logpath=/opt/fonsview/3RD/mongodb/shard1/log/shard1.log  
logappend=true  
bind\_ip=0.0.0.0  
port=25001  
fork=true  
#副本集名称  
replSet=shard1

#声明集群分片数据库

shardsvr = true  
#设置最大连接数  
maxConns=20000

1. 分别启动三台服务器的shard1 server，进入 /opt/fonsview/3RD/mongdb/bin目录下

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard1.conf

1. 登录任意一台服务器A，初始化配置副本集, 进入/opt/fonsview/3RD/mongdb/bin 目录下

./mongo --port 25001

使用admin数据库,定义副本配置，并初始化:

use admin;

config = {  
    \_id : "shard1",  
     members : [  
         {\_id : 0, host : "172.16.6.92:25001" },  
         {\_id : 1, host : "172.16.6.95:25001" },  
         {\_id : 2, host : "172.16.6.103:25001", arbiterOnly: true }  
     ]  
 };

rs.initiate(config)

第三个节点的 arbiterOnly: true 表示该节点为仲裁节点

**设置第二个副本集**：

1. 添加配置文件:

vim /opt/fonsview/3RD/mongodb/conf/shard2.conf

1. 进入配置文件后添加如下信息：

## 配置文件内容  
pidfilepath = /opt/fonsview/3RD/mongodb/shard2/log/shard2.pid  
dbpath = /opt/fonsview/3RD/mongodb/shard2/data  
logpath = /opt/fonsview/3RD/mongodb/shard2/log/shard2.log  
logappend = true  
bind\_ip = 0.0.0.0  
port = 25002  
fork = true

#副本集名称  
replSet=shard2

#声明集群分片数据库

shardsvr = true  
#设置最大连接数  
maxConns=20000

1. 分别启动三台服务器的shard2 server，进入 /opt/fonsview/3RD/mongdb/bin目录下

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard2.conf

1. 登录任意一台服务器B，初始化配置副本集, 进入/opt/fonsview/3RD/mongdb/bin 目录下

./mongo --port 25002

使用admin数据库,定义副本配置，并初始化:

use admin;

config = {  
    \_id : "shard2",  
     members : [  
         {\_id : 0, host : "172.16.6.92:25002", arbiterOnly: true  },  
         {\_id : 1, host : "172.16.6.95:25002" },  
         {\_id : 2, host : "172.16.6.103:25002" }  
     ]  
 };

rs.initiate(config)

**设置第三个副本集：**

1. 添加配置文件:

vim /opt/fonsview/3RD/mongodb/conf/shard3.conf

1. 进入配置文件后添加如下信息：

## 配置文件内容  
pidfilepath = /opt/fonsview/3RD/mongodb/shard3/log/shard3.pid  
dbpath = /opt/fonsview/3RD/mongodb/shard3/data  
logpath = /opt/fonsview/3RD/mongodb/shard3/log/shard3.log  
logappend = true  
bind\_ip = 0.0.0.0  
port = 25003  
fork = true

#副本集名称  
replSet=shard3

#声明集群分片数据库

shardsvr = true  
#设置最大连接数  
maxConns=20000

1. 分别启动三台服务器的shard3 server，进入 /opt/fonsview/3RD/mongdb/bin目录下

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard3.conf

1. 登录任意一台服务器C，初始化配置副本集, 进入/opt/fonsview/3RD/mongdb/bin 目录下

./mongo --port 25003

使用admin数据库,定义副本配置，并初始化:

use admin;

config = {  
    \_id : "shard3",  
     members : [  
         {\_id : 0, host : "172.16.6.92:25003" },  
         {\_id : 1, host : "172.16.6.95:25003", arbiterOnly: true },  
         {\_id : 2, host : "172.16.6.103:25003"}  
     ]  
 };

rs.initiate(config)

1. 配置路由服务器:
2. 添加配置文件:

vim /opt/fonsview/3RD/mongodb/conf/mongos.conf

1. 进入配置文件后添加如下信息：

#内容  
pidfilepath =  /opt/fonsview/3RD/mongodb/mongos/log/mongos.pid  
logpath =  /opt/fonsview/3RD/mongodb/mongos/log/mongos.log  
logappend = true  
bind\_ip = 0.0.0.0  
port = 23000  
fork = true  
#监听的配置服务器,只能有1个或者3个 configs为配置服务器的副本集名字  
configdb = configs/172.16.6.92:24000,172.16.6.95:24000,172.16.6.103:24000  
#设置最大连接数  
maxConns=20000

1. 分别启动三台服务器的mongos server，进入 /opt/fonsview/3RD/mongdb/bin目录下

./mongos -f /opt/fonsview/3RD/mongodb/conf/mongos.conf

1. 分片：

（1）mongos分片配置：

目前搭建了mongodb配置服务器、路由服务器，各个分片服务器，不过应用程序连接到mongos路由服务器并不能使用分片机制，还需要在程序里设置分片配置，让分片生效。

登录任意一台mongos，进入/opt/fonsview/3RD/mongdb/bin 目录下

./mongo --port 23000

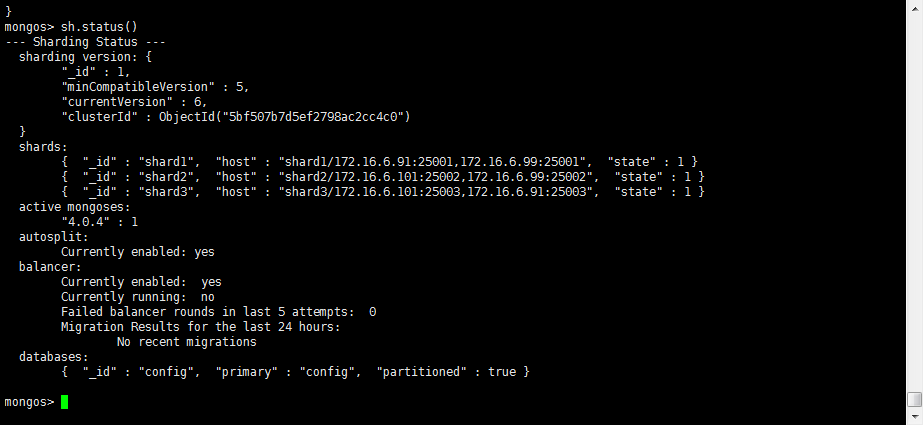
使用admin数据库，串联路由服务器与分配副本集：

use admin

sh.addShard("shard1/172.16.6.92:25001,172.16.6.95:25001,172.16.6.103:25001")  
sh.addShard("shard2/172.16.6.92:25002,172.16.6.95:25002,172.16.6.103:25002")  
sh.addShard("shard3/172.16.6.92:25003,172.16.6.95:25003,172.16.6.103:25003")

查看集群状态：

sh.status()



（2）指定需要分片的数据库：

db.runCommand( { enablesharding :"favorite"});

（3）指定数据库里需要分片的集合和片键：

db.runCommand( { shardcollection : "favorite.bookmark\_collection",key : {userId: "hashed" } } )

1. 将命令添加到系统路径中：

将命令添加到系统路径中，方便随处执行mongo命令：

1、在/etc/profile文件中，添加 export ：PATH=$PATH:/opt/fonsview/3RD/mongodb/bin

2、执行source /etc/profile，使系统环境变量立即生效

3、将mongo路径软链接到/usr/bin下：

ln -s /opt/fonsview/3RD/mongodb/bin/mongo  /usr/bin/mongo

1. 启动和关闭：

**安装部署过程中，已经执行了下列启动命令，故无需再次执行**

Mongodb的启动顺序是，先启动配置服务器，再启动分片，最后启动mongos.

./mongod -f /opt/fonsview/3RD/mongodb/conf/config.conf

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard1.conf

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard2.conf

./mongod -f /opt/fonsview/3RD/mongodb/conf/shard3.conf

./mongos -f /opt/fonsview/3RD/mongodb/conf/mongos.conf

关闭时，执行在进入mongos、分片和配置服务器分别执行如下命令：

use admin

db.shutdowmServer()

强烈不建议使用 kill -9 pid 方式关闭，这种方法容易造成数据丢失及服 务器瘫痪

1. 常用命令：

在数据库 admin 下的执行下列命令:

**db.runCommand( { movePrimary : "favorite", to : "shard1" } )** ：使用 movePrimary 命令变更数据库默认的Primary shard，非分片集合将会从当前shard移动到新的主分片。

**db.adminCommand({"flushRouterConfig":1})**：在使用movePrimary命令变更数据库的主分片之后，config server中的配置信息是最新的，mongos缓存的配置信息变得过时了。MongoDB提供命令：flushRouterConfig 强制mongos从config server获取最新的配置信息，刷新mongos的缓存。

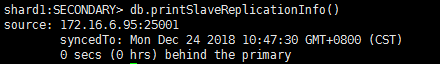
在数据库 testdb 下的执行下列命令:

**help**: 查看帮助

**rs.status()**：查看复制集状态

**rs.printReplicationInfo()**:查看oplog状态

**rs.printSlaveReplicationInfo()**:查看复制集延迟



* source——从库的IP及端口
* syncedTo——当前的同步情况，延迟了多久等信息

**use testdb** :切换 testdb 数据库

**show collections** : 查看当前数据库下的所有集合

**db** : 查看当前数据库名称

**db.serverStatus()**:查看服务状态详情

**db.stats()**:查询当前数据库统计信息

在数据库 testdb 中的集合 testcollection 下执行下列命令：

**db.testcollection.count()** : 获取当前集合文档总数

**db.testcollection.find().size(m).limit(n)** : 从第m条开始查询n条记录

**db.testcollection.getShardDistribution()** : 查看分片情况

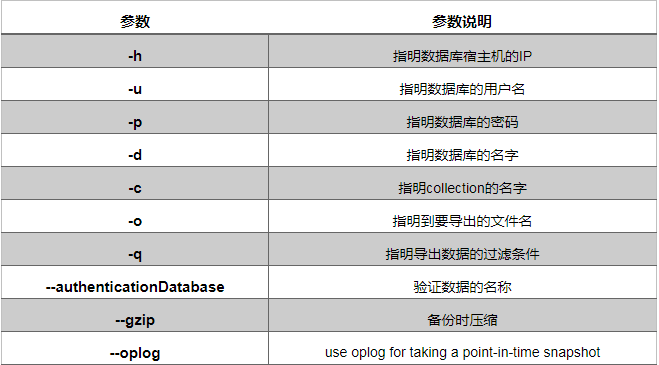
1. 数据备份与恢复：

1 mongodump/mongorestore：

1. 利用mongodump进行备份:

mongodump -h **172.16**.6**.92**:**23000** -u root -p root --authenticationDatabase favorite -o /opt/fonsview/3RD/dbBackup

若无密码，则 -u -p 省略

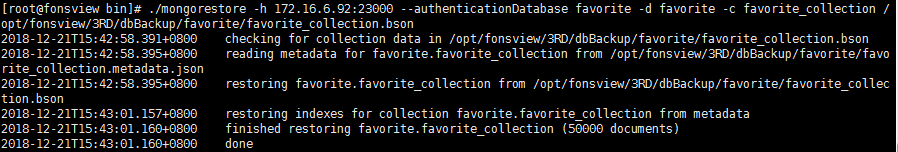
命令说明： 

1. 利用mongorestore进行恢复：

mongorestore -h **172.16**.6**.92**:**23000** -u root -p root --authenticationDatabase favorite -d test -c favorite\_collection /opt/fonsview/3RD/dbBackup/favorite\_collection.bson

命令说明：

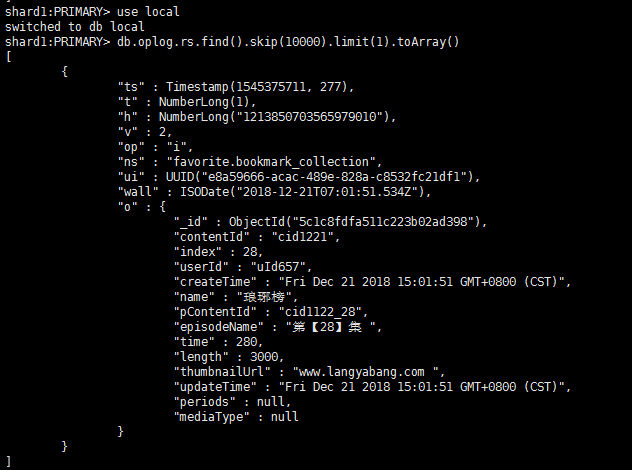
成功恢复结果如下图所示：



需要重启mongos

2 oplog:

MongoDB 的Replication是通过一个日志来存储写操作的，这个日志就叫做oplog。Secondary就是通过查看Primary 的oplog这个集合来进行复制的。每个节点都有oplog，记录这从主节点复制过来的信息，这样每个成员都可以作为同步源给其他节点。

oplog的数据结构如下：

* ts: 8字节的时间戳，由4字节unix timestamp + 4字节自增计数表示。
* op：1字节的操作类型
  + "i"： insert
  + "u"： update
  + "d"： delete
  + "c"： db cmd
  + "db"：声明当前数据库 (其中ns 被设置成为=>数据库名称+ '.')
  + "n": no op,即空操作，其会定期执行以确保时效性
* ns：操作所在的namespace
* o：操作所对应的document，即当前操作的内容（比如更新操作时要更新的的字段和值）
* o2: 在执行更新操作时的where条件，仅限于update时才有该属性
* **全量恢复：**

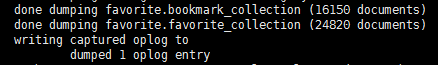
首先分别导出各分区的 oplog

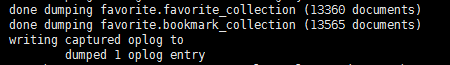
mongodump -h 127.0.0.1 --port 25001 --oplog -o /home/mongod/backup/oplog

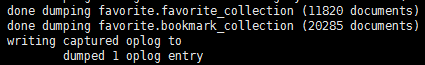
mongodump -h 127.0.0.1 --port 25002 --oplog -o /home/mongod/backup/oplog

mongodump -h 127.0.0.1 --port 25003 --oplog -o /home/mongod/backup/oplog

操作成功：

Shard1：

Shard2：

Shard3：

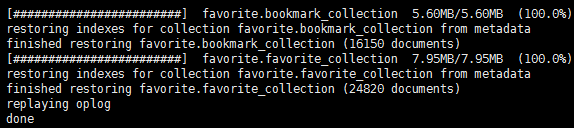
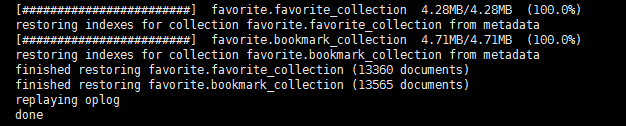
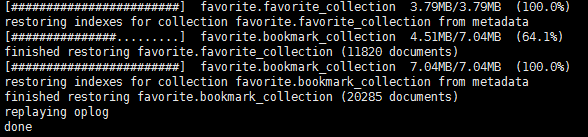
操作日志获取成功后，使用 mongorestore 命令即可恢复：

mongorestore -h 127.0.0.1 --port 25001 --oplogReplay --drop /home/mongod/backup/oplog

mongorestore -h 127.0.0.1 --port 25002 --oplogReplay --drop /home/mongod/backup/oplog

mongorestore -h 127.0.0.1 --port 25003 --oplogReplay --drop /home/mongod/backup/oplog

恢复成功截图如下：

Shard1:Shard2:Shard3:

* **即时恢复PITR（Point-In-Time-Recovery）：**

前提：

需要恢复的时间点在 oplog 时间域中

模拟错误操作删除集合：

db.bookmark\_collection.drop()

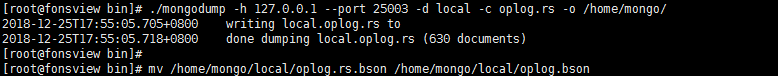
集合删除后，原有的分片信息也会被删除，因此需要重新指定数据库里需要分片的集合和片键：

db.runCommand( { shardcollection : "favorite.bookmark\_collection",key : {userId: "hashed" } } )

导出local 库下的 oplog.rs 集合并对 oplog 重命名：

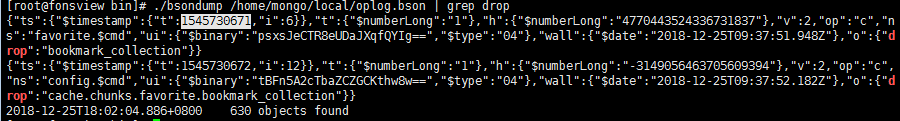
./mongodump -h 127.0.0.1 --port 25003 -d local -c oplog.rs -o /home/mongo/

mv /home/mongo/local/oplog.rs.bson /home/mongo/local/oplog.bson



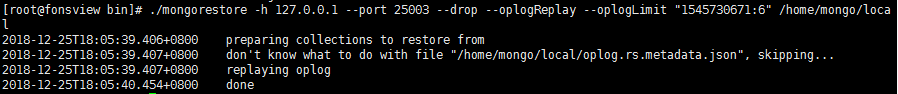
在bson文件中查找时间点：

./bsondump /home/mongo/local/oplog.bson | grep drop



执行还原操作：

./mongorestore -h 127.0.0.1 --port 25001 --oplogReplay --oplogLimit "1545787867:4" /home/mongo/local/



还原成功：



1. 常见问题：

若三台分片所在的一台发生故障（如：重启），不会影响数据库的运行。在发生故障的那台重新启动conf服务、mongod服务和mongos服务后按照如下步骤设置主节点优先级即可（如主、副节点未发生变化请忽略）。

在设置mongodb副本集时，Primary节点，second节点，仲裁节点，有可能资源配置（CPU或者内存）不均衡，所以要求某些节点不能成为Primary  
我们知道mongodb的设置：  
  除了仲裁节点，其他每个节点都有个优先权，可以手动设置优先权来决定谁的成为primay的权重最大。  
  副本集中通过设置priority的值来决定优先权的大小，这个值的范围是0--100，值越大，优先权越高。  
默认的值是1，rs.conf是不显示的；  
如果值是0，那么不能成为primay。  
1.规划时直接设置，这个就略过了  
2.在线加入的节点配置：  
配置过程：  
通过修改priority的值来实现（默认的优先级是1（0-100），priority的值设的越大，就优先成为主）  
1)PRIMARY> cfg=rs.conf()  
2)PRIMARY>cfg.members[0].priority = 2  
3)PRIMARY> rs.reconfig(cfg)  
注意：第2步members大括号里面的成员和\_id是没有关系的，而是rs.conf查出来节点的数值的顺序；

这些操作必须在Primary上进程。

for (var userId = 1; userId <= 100; userId++) {

var arr1 = new Array();

var arr2 = new Array();

for (var index = 1; index <= 5; index++) {

arr1.push({

"contentId": "cid1225",

"index": index,

"userId": "uId" + userId,

"createTime": Date(),

"name": "琅琊榜",

"pContentId": "cid1225\_" + index,

"episodeName": "第【" + index + "】集 ",

"time": index \* 10,

"length": 3000,

"thumbnailUrl": "www.langyabang.com ",

"updateTime": Date(),

"periods": null,

"mediaType": "2"

});

arr2.push({

"contentId" : "cid"+index,

"userId" : "uId"+userId,

"createTime" : Date(),

"name" : "亮剑"+index,

"thumbnailUrl" : "www.liangjian.com",

"updateTime" : Date(),

"mediaType" : "2",

"reserved" : "none",

"subName" : "意大利炮",

"cornerMark" : "cm001",

"programType" : "战争"

});

}

db.bookmark\_collection.insertMany(arr1);

}