**目录**

[1. 前言 2](#_Toc455069886)

[2. 升级前注意 3](#_Toc455069887)

[3. 升级单个mongod实例 3](#_Toc455069888)

[3.1升级mongod实例 3](#_Toc455069889)

[3.2变更引擎 3](#_Toc455069890)

[4. 升级副本集 4](#_Toc455069891)

[4.1升级副本集各实例 4](#_Toc455069892)

[4.2变更引擎 5](#_Toc455069893)

[4.2.1变更前提示 5](#_Toc455069894)

[4.2.2更新引擎 5](#_Toc455069895)

[5. 升级分片集群 6](#_Toc455069896)

[5.1升级前注意 6](#_Toc455069897)

[5.2关闭均衡器 7](#_Toc455069898)

[5.3备份并更新元数据 8](#_Toc455069899)

[5.3.1备份元数据 8](#_Toc455069900)

[5.3.2关闭40002端口的mongos用其更新元数据 8](#_Toc455069901)

[5.4更新其他组件 9](#_Toc455069902)

[5.4.1更新mongos 9](#_Toc455069903)

[5.4.2更新3个mongod-config服务 10](#_Toc455069904)

[5.4.2.1将所有2.6的mongod-config更新为3.0 10](#_Toc455069905)

[(1)关闭并重新启动30003端口的mongod-config服务 10](#_Toc455069906)

[(2)关闭并重新启动30002端口的mongod-config服务 10](#_Toc455069907)

[(3)关闭并重新启动30001端口的mongod-config服务 11](#_Toc455069908)

[5.4.2.2将所有3.0普通引擎更新为wiredTiger 11](#_Toc455069909)

[(1)关闭均衡器 11](#_Toc455069910)

[(2)关闭--configdb第3个mongod-config（即30003端口）服务以确保元数据为只读模式 12](#_Toc455069911)

[(3)用mongodump导出--configdb第2个mongod-config（即30002端口）服务的数据 12](#_Toc455069912)

[(4)为--configdb第2个mongod-config（即30002端口）服务创建一个新的数据目录 13](#_Toc455069913)

[(5)关闭--configdb第2个mongod-config（即30002端口）服务 13](#_Toc455069914)

[(6)用wiredTiger和new\_dir启动--configdb第2个mongod-config（即30002端口）服务 13](#_Toc455069915)

[(7)用mongorestore给新启动的30002端口服务导入之前用mongodump导出的30002的数据 13](#_Toc455069916)

[(8)再次关闭--configdb第2个mongod-config（即30002端口）服务 13](#_Toc455069917)

[(9)重启--configdb第3个mongod-config（即30003端口）服务 14](#_Toc455069918)

[(10)用mongodump导出--configdb第3个mongod-config（即30003端口）服务的数据 14](#_Toc455069919)

[(11)为--configdb第3个mongod-config（即30003端口）服务创建一个新的数据目录 14](#_Toc455069920)

[(12)关闭--configdb第3个mongod-config（即30003端口）服务 14](#_Toc455069921)

[(13)用wiredTiger和new\_dir启动--configdb第3个mongod-config（即30003端口）服务 15](#_Toc455069922)

[(14)将刚刚导出的30003端口的数据导入到新启动的30003服务中来 15](#_Toc455069923)

[(15)用mongodump导出--configdb第1个mongod-config（即30001端口）服务的数据 15](#_Toc455069924)

[(16)为--configdb第1个mongod-config（即30001端口）服务创建一个新的数据目录 15](#_Toc455069925)

[(17)关闭--configdb第1个mongod-config（即30001端口）服务 15](#_Toc455069926)

[(18)用wiredTiger和new\_dir启动--configdb第1个mongod-config（即30001端口）服务 16](#_Toc455069927)

[(19)将刚刚导出的30001端口的数据导入到新启动的30001服务中来 16](#_Toc455069928)

[(20)关闭并重启--configdb第1个mongod-config（即30001端口）服务 16](#_Toc455069929)

[(21)重启打开均衡器 17](#_Toc455069930)

[(21)更新完成 17](#_Toc455069931)

[5.4.3更新分片副本集 17](#_Toc455069932)

[5.4.3.1更新lt01 17](#_Toc455069933)

[(1)更新lt01-10003端口的结点 18](#_Toc455069934)

[(2)更新lt01-10002端口的结点 18](#_Toc455069935)

[(3)切换Primary 19](#_Toc455069936)

[(4)更新lt01-10001端口的结点 19](#_Toc455069937)

[5.4.3.2更新lt02 20](#_Toc455069938)

[(1)更新lt02-20003端口结点 20](#_Toc455069939)

[(2)更新lt02-20001端口结点 20](#_Toc455069940)

[(3)切换Primary 21](#_Toc455069941)

[(4)更新lt02-20002端口结点 21](#_Toc455069942)

[5.5总结 22](#_Toc455069943)

[6. Q&A 23](#_Toc455069944)

[7. 参考资料 23](#_Toc455069945)

**jar cvfm MyWordCount.jar manifest.data \*.class**

**关键数据升级务必进行升级安全评估，避免升级过程出现问题对业务造成致命影响！！！**

# 前言

和前面升级类似，能直接升级到3.0的版本是2.6+，其他低版本必须依次进行，本篇介绍**如何从2.6升级到3.0**。（至于2.8版本，可能没有太大变化，官方没有给出专门文档来说明如何从2.8升级）

# 升级前注意

从2.6升级到3.0需要注意入下问题：

1. 必须从2.6+版本升级，2.6-均不能直接升级到3.0；
2. 如果待升级版本使用了授权认证，那么用户数据模型至少需要是v3版本，即2.6版本的授权认证模型;

可以参考https://docs.mongodb.com/manual/release-notes/3.0-compatibility/#legacy-auth-model-removed来处理2.4的用户认证模型，参考该链接将其更新到2.6的用户授权认证模式https://docs.mongodb.com/manual/release-notes/2.6-upgrade-authorization/.

1. 在该链接https://docs.mongodb.com/manual/release-notes/3.0-compatibility/处查看关于MongoDB3.0有关兼容性的问题，如果现有版本存在这些问题，那么请处理之后再进行升级；

# 升级单个mongod实例

## 3.1升级mongod实例

2.6+版本有资格直接升级到3.0。

（1）在http://www.mongodb.org/downloads?jmp=docs下载MongoDB3.0版本；

（2）关闭2.6版本，然后用3.0的直接启动即可；

## 3.2变更引擎

升级到3.0的一个重要原因就是使用3.0的WiredTiger存储引擎，它可以大大缩减磁盘使用率，但是要在启动3.0时使用WiredTiger引擎，还需要进行如下过程：

1. 用mongodump导出2.6中的数据；
2. 创建一个全新的数据目录new\_dir为新启动的mongod3.0备用；
3. 用mongod3.0加上WiredTiger，加上new\_dir启动，如下：

**mongod --storageEngine wiredTiger --dbpath <newWiredTigerDBPath>**

1. 最后将之前导出的数据重新导入到新启动的mongod3.0中来：

**mongorestore <exportDataDestination>**

使用新的引擎略显麻烦。

# 升级副本集

## 4.1升级副本集各实例

为了方便描述，我们假定升级如下模型的副本集：



1. 首先更新各个secondary结点（这是必须的），我们先关闭Secondary-2结点的mongod，然后用mongod3.0替换重启。等待Secondary-2恢复到Secondary状态，再继续进行；

**J:\mongodb-64-2.6.12\bin\mongo ip:port**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

**J:\mongodb-64-3.0.10\bin\mongod --port=port --dbpath="…"**

1. 等到Secondary-2恢复到Secondary状态之后，我们用同样的方法升级一下Secondary-1结点；
2. 此时，我们切换Primary到另外两个结点中的一个。登录到Primary上，执行切换动作，然后用3.0重启该节点；

**J:\mongodb-64-2.6.12\bin\mongo ip:port**

**use admin;**

**rs.stepDown(); # 切换Primay结点，这个动作会强迫执行一个故障转移failover**

**db.shutdownServer(); # 如果不放心，执行rs.status()查看结点状态确认已经关闭**

**J:\mongodb-64-3.0.10\bin\mongod --port=port --dbpath="…"**

由此可见，副本集的更新到3.0时是允许热更新的，而且注意到允许存在不同版本的mongod实例运行。

副本集的故障转移（failover，失效备援、故障转移）并不是立即完成的，在故障转移完成之前是不能进行写操作的，通常需要等待30s或者更多。

## 4.2变更引擎

类似于单个实例，我们可以将副本集中的mongod引擎也更新为WiredTiger，详细过程如下：

### 4.2.1变更前提示

好在副本集允许各个成员直接存在不同的存储引擎，因此我们采用滚动的方式来让新成员使用WiredTiger。再将所有的成员更新为WiredTiger之前，注意让混合引擎运行一段时间（官方也没有给出一个参考时间），此时负载方法可能会受到一定的影响。 **待更新的版本必须2.6.8+**。

### 4.2.2更新引擎

（1）关闭副本集Secondary-2，如下：

**J:\mongodb-64-2.6.12\bin\mongo ip:port**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

（2）为即将使用的WiredTiger引擎创建一个新的存储目录

为即将使用的WiredTiger引擎创建一个单独的新目录new\_dir，或者删除已经停掉的Secondary-2使用的数据目录old\_dir中的全部内容，让WiredTiger继续使用这个目录old\_dir。

注意：mongod-WiredTiger使用不同引擎创建的数据文件将无法启动或者启动之后无法识别数据文件，用2.4的数据让3.0wiredTiger启动之后无法找到数据文件。

1. 使用3.0加上wiredTiger存储引擎，加上新目录new\_dir，以副本方式启动Secondary-2结点：

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --dbpath=<newWiredTigerDBPath> --replSet <replSetName>**

以新的方式启动之后，Secondary-2会同步Primary结点的数据过来，至于时间长短这依赖于网络质量和数据量的大小。

1. 以同样的方式对剩下两个结点进行操作，注意rs.stepDown()切换Primary结点。

# 升级分片集群

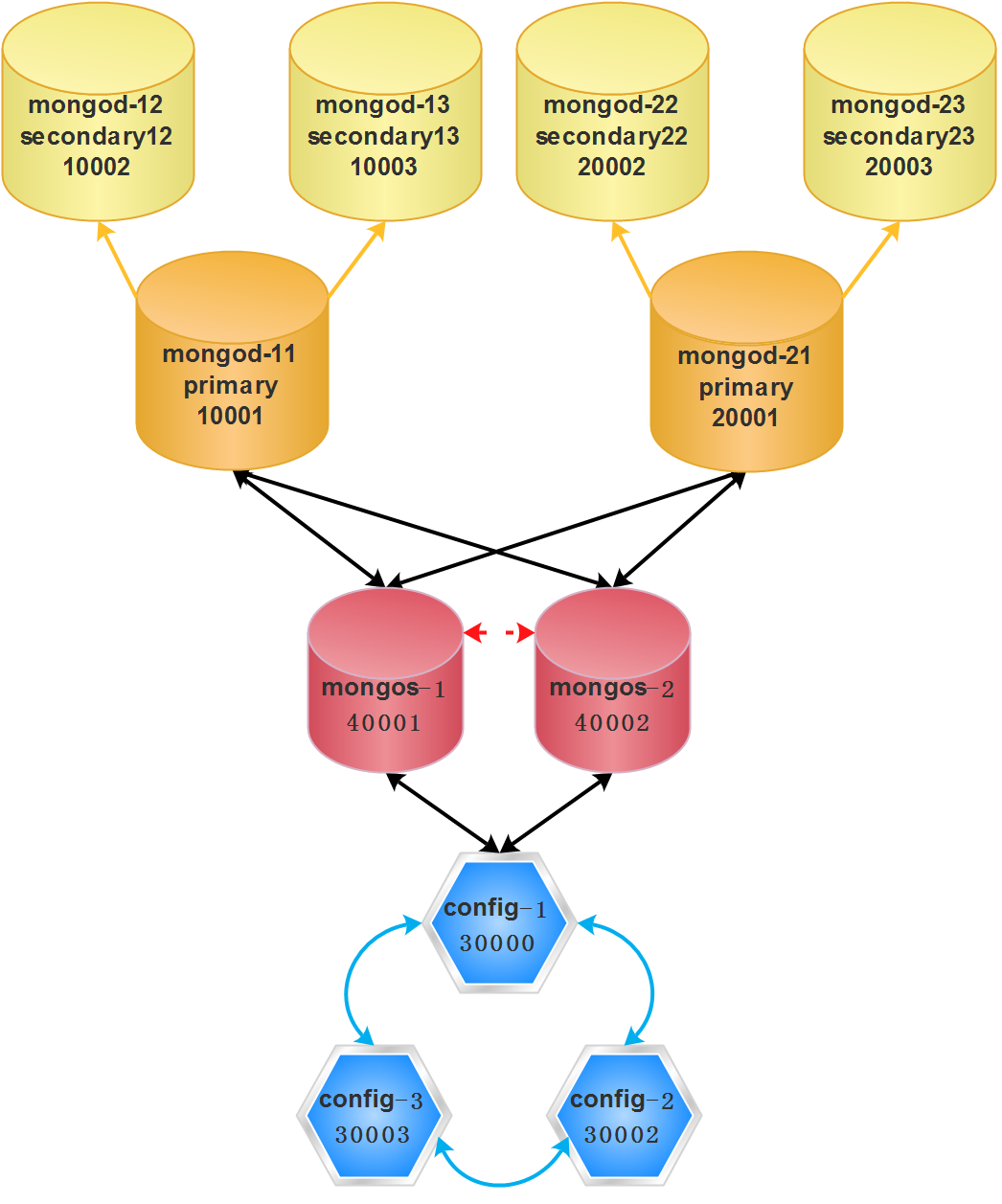
同样也必须是2.6+才能直接升级到3.0。

## 5.1升级前注意

升级集群可以不下线，但是必须确保在升级过程中不能有修改元数据的操作，比如：

* [sh.enableSharding()](https://docs.mongodb.com/manual/reference/method/sh.enableSharding/#sh.enableSharding)；
* [sh.shardCollection()](https://docs.mongodb.com/manual/reference/method/sh.shardCollection/#sh.shardCollection)；
* [sh.addShard()](https://docs.mongodb.com/manual/reference/method/sh.addShard/#sh.addShard)；
* [db.createCollection()](https://docs.mongodb.com/manual/reference/method/db.createCollection/#db.createCollection)；
* [db.collection.drop()](https://docs.mongodb.com/manual/reference/method/db.collection.drop/#db.collection.drop)；
* [db.dropDatabase()](https://docs.mongodb.com/manual/reference/method/db.dropDatabase/#db.dropDatabase)；
* any operation that creates a database；
* <https://docs.mongodb.com/manual/reference/sharding/>这个地方的所有命令，注意并不是所有命令都会导致修改元数据，但是为了确保最好在升级元数据的过程不要使用这个链接出的所有命令；

更新过程同样分为：元数据更新、其他组件更新。假定我们要更新的集群模型如下：



详细更新过程如下：

## 5.2关闭均衡器

登录到40001端口的mongos上关闭均衡器，如下操作：

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:40001**

**MongoDB shell version: 2.6.12**

**connecting to: 127.0.0.1:40001/test**

**mongos> sh.stopBalancer();**

**Waiting for active hosts...**

**Waiting for the balancer lock...**

**Waiting again for active hosts after balancer is off...**

**mongos> sh.getBalancerState();**

**false**

**mongos>**

## 5.3备份并更新元数据

### 5.3.1备份元数据

为了安全起见，我们备份一下配置元数据：

**J:\mongodb-64-2.6.12\bin\mongodump -host 127.0.0.1:30001 -d config -o "G:\mongodb\_upgrade\_backup"**

### 5.3.2关闭40002端口的mongos用其更新元数据

注意，升级元数据的时候，mongod-config的服务一定要处于开启状态，否则升级谁呢，对吧！

我们的集群中有2个mongos，我们逐个升级，这样可以避免业务受牵连（前提是业务采取了轮询mongos的方式访问）。在此我们先关闭40002端口的mongos服务，利用40002端口的mongos服务来升级元数据，如下进行。

**# 关闭旧的40002端口mongos**

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:40002**

**use admin;**

**db.shutdownServer();**

**# 在40002端口启动一个新3.0版本的mongos加上--upgrade用来更新元数据**

**J:\mongodb-64-3.0.10\bin\mongos.exe --port=40002 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003" --upgrade**

执行之后出现如下信息表明更新成功：

**2016-06-30T10:03:03.865+0800 I SHARDING [mongosMain] upgrade of config server to v6 successful**

**2016-06-30T10:03:03.990+0800 I SHARDING [mongosMain] distributed lock 'configUpgrade/LIULEI26:40002:14672**

**52182:41' unlocked.**

**2016-06-30T10:03:03.991+0800 I - [mongosMain] Config database is at version v6**

之后再用不带--upgrade启动40002端口的mongos3.0，如下：

**J:\mongodb-64-3.0.10\bin\mongos.exe --port=40002 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003"**

## 5.4更新其他组件

### 5.4.1更新mongos

我们更新另一个mongos，现在我们更新40001端口的mongos，此时就不再需要--upgrade选项了，如下操作即可：

**# 关闭旧的40001端口mongos**

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:40001**

**use admin;**

**db.shutdownServer();**

**# 在40001端口启动一个新3.0版本的mongos用来更新元数据**

**J:\mongodb-64-3.0.10\bin\mongos.exe --port=40001 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003"**

### 5.4.2更新3个mongod-config服务

按照官方建议按照mongos --configdb后面参数顺序从后往前更新，我们的更新顺序是30003、30002、30001。我们在此也将mongod-config使用新的存储引擎，我们分为两步进行，（1）将所有2.6引擎以3.0替换，按原始引擎工作；（3）将3.0引擎重启为wiredTiger。

#### 5.4.2.1将所有2.6的mongod-config更新为3.0

如下更新：

##### (1)关闭并重新启动30003端口的mongod-config服务

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:30003**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-3.0.10\bin\mongod --dbpath="G:\mongodb-config3-30003\data" --port=30003 --configsvr**

##### (2)关闭并重新启动30002端口的mongod-config服务

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:30002**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-3.0.10\bin\mongod --dbpath="G:\mongodb-config2-30002\data" --port=30002 --configsvr**

##### (3)关闭并重新启动30001端口的mongod-config服务

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:30001**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-3.0.10\bin\mongod --dbpath="G:\mongodb-config1-30001\data" --port=30001 --configsvr**

至此，所有的mongod-config配置服务器已经更新为普通引擎3.0完毕。

#### 5.4.2.2将所有3.0普通引擎更新为wiredTiger

下面我们将mongod-config全部更新为wiredTiger引擎，如下操作：

##### (1)关闭均衡器

**C:\Users\Administrator>J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:40002**

**2016-06-30T10:28:51.725+0800 I CONTROL Hotfix KB2731284 or later upda**

**data files**

**MongoDB shell version: 3.0.10**

**connecting to: 127.0.0.1:40002/test**

**mongos> sh.stopBalancer();**

**Waiting for active hosts...**

**Waiting for the balancer lock...**

**Waiting again for active hosts after balancer is off...**

**mongos> sh.getBalancerState();**

**false**

**mongos>**

##### (2)关闭--configdb第3个mongod-config（即30003端口）服务以确保元数据为只读模式

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30003**

**use admin;**

**db.shutdownServer();**

##### (3)用mongodump导出--configdb第2个mongod-config（即30002端口）服务的数据

**J:\mongodb-64-3.0.10\bin\mongodump --host=127.0.0.1:30002 --out=G:\mongodb\_config\_2d\_exportdata**

一定确保成功。

##### (4)为--configdb第2个mongod-config（即30002端口）服务创建一个新的数据目录

G:\mongodb-config2-30002\_new\_dir，而且必须确保该目录为空。

##### (5)关闭--configdb第2个mongod-config（即30002端口）服务

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30002**

**use admin;**

**db.shutdownServer();**

##### (6)用wiredTiger和new\_dir启动--configdb第2个mongod-config（即30002端口）服务

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine wiredTiger --dbpath="G:\mongodb-config2-30002\_new\_dir" --port=30002 --configsvr**

##### (7)用mongorestore给新启动的30002端口服务导入之前用mongodump导出的30002的数据

**J:\mongodb-64-3.0.10\bin\mongorestore --host=127.0.0.1:30002 G:\mongodb\_config\_2d\_exportdata**

##### (8)再次关闭--configdb第2个mongod-config（即30002端口）服务

完成了--configdb第2个mongod-config（即30002端口）服务的更新，我们再次关闭这个服务。提示：长时间内，我们需要注意必须保证有2个配置服务处于active状态并且保持集群配置元数据为只读状态（没有写入更新操作即可）。

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30002**

**use admin;**

**db.shutdownServer();**

##### (9)重启--configdb第3个mongod-config（即30003端口）服务

用原始选项启动这个服务：

**J:\mongodb-64-3.0.10\bin\mongod --dbpath="G:\mongodb-config3-30003\data" --port=30003 --configsvr**

##### (10)用mongodump导出--configdb第3个mongod-config（即30003端口）服务的数据

**J:\mongodb-64-3.0.10\bin\mongodump --host=127.0.0.1:30003 --out=G:\mongodb\_config\_3rd\_exportdata**

##### (11)为--configdb第3个mongod-config（即30003端口）服务创建一个新的数据目录

G:\mongodb-config3-30003\_new\_dir，而且必须确保该目录为空。

##### (12)关闭--configdb第3个mongod-config（即30003端口）服务

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30003**

**use admin;**

**db.shutdownServer();**

##### (13)用wiredTiger和new\_dir启动--configdb第3个mongod-config（即30003端口）服务

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine wiredTiger --dbpath="G:\mongodb-config3-30003\_new\_dir" --port=30003 --configsvr**

##### (14)将刚刚导出的30003端口的数据导入到新启动的30003服务中来

**J:\mongodb-64-3.0.10\bin\mongorestore --host=127.0.0.1:30003 G:\mongodb\_config\_3rd\_exportdata**

##### (15)用mongodump导出--configdb第1个mongod-config（即30001端口）服务的数据

**J:\mongodb-64-3.0.10\bin\mongodump --host=127.0.0.1:30001 --out=G:\mongodb\_config\_1st\_exportdata**

##### (16)为--configdb第1个mongod-config（即30001端口）服务创建一个新的数据目录

G:\mongodb-config1-30001\_new\_dir，而且必须确保该目录为空。

##### (17)关闭--configdb第1个mongod-config（即30001端口）服务

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30001**

**use admin;**

**db.shutdownServer();**

##### (18)用wiredTiger和new\_dir启动--configdb第1个mongod-config（即30001端口）服务

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine wiredTiger --dbpath="G:\mongodb-config1-30001\_new\_dir" --port=30001 --configsvr**

##### (19)将刚刚导出的30001端口的数据导入到新启动的30001服务中来

**J:\mongodb-64-3.0.10\bin\mongorestore --host=127.0.0.1:30001 G:\mongodb\_config\_1st\_exportdata**

##### (20)关闭并重启--configdb第1个mongod-config（即30001端口）服务

这个动作用来激活元数据的写入功能：

**J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:30002**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine wiredTiger --dbpath="G:\mongodb-config2-30002\_new\_dir" --port=30002 --configsvr**

##### (21)重启打开均衡器

**C:\Users\Administrator>J:\mongodb-64-3.0.10\bin\mongo 127.0.0.1:40002**

**2016-06-30T10:28:51.725+0800 I CONTROL Hotfix KB2731284 or later upda**

**data files**

**MongoDB shell version: 3.0.10**

**connecting to: 127.0.0.1:40002/test**

**mongos> sh.setBalancerState(true);**

##### (21)更新完成

至此整个mongod-config使用wiredTiger引擎更新完成，过程还是颇为麻烦的。

### 5.4.3更新分片副本集

分片的更新过程和之前类似，但是我们要使用新的wiredTiger引擎。更新分片为3.0并且使用wiredTiger引擎，相比跟新mongod-config要简单一点，但是就是在数据量较大的情况下可能会比较缓慢，因为有一个数据同步的过程。下面我们开始更新：

我们按照这样一个步骤更新：lt01——》lt02。每个分片先更新副本结点、然后切换Primay、最后更新切换后的结点。注意找到Primary结点。

#### 5.4.3.1更新lt01

登录到10001机器上，rs.status()查看当前Primay结点，lt01分片的当前Primary结点为10001端口，因此我们最后更新这个结点。

##### (1)更新lt01-10003端口的结点

* 关闭lt01-10003结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:10003**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

* 为lt01-10003结点创建一个新的数据目录

G:\mongodb-shard01repl-mongod3-10003\_new\_dir\data，这是lt分片的10003端口新数据目录。

* 以wiredTiger+new\_dir选项、使用10003端口启动一个mongod3.0

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=10003 --dbpath="G:\mongodb-shard01repl-mongod3-10003\_new\_dir\data" --replSet=lt01 --shardsvr**

##### (2)更新lt01-10002端口的结点

* 关闭lt01-10002结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:10002**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

* 为lt01-10002结点创建一个新的数据目录

G:\mongodb-shard01repl-mongod2-10002\_new\_dir\data，这是lt01分片的10002端口新数据目录。

* 以wiredTiger+new\_dir选项、使用10002端口启动一个mongod3.0

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=10002 --dbpath="G:\mongodb-shard01repl-mongod2-10002\_new\_dir\data" --replSet=lt01 --shardsvr**

##### (3)切换Primary

注意：再切换Primary结点之前，必须等待上面两个结点将数据同步完成。关注两个副本结点是健康状态的。

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:10001**

**rs.stepDown()**

**rs.status();**

##### (4)更新lt01-10001端口的结点

* 关闭lt01-10001结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:10001**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

* 为lt01-10001结点创建一个新的数据目录

G:\mongodb-shard01repl-mongod1-10001\_new\_dir\data，这是lt分片的10001端口新数据目录。

* 以wiredTiger+new\_dir选项、使用10001端口启动一个mongod3.0

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=10001 --dbpath="G:\mongodb-shard01repl-mongod1-10001\_new\_dir\data" --replSet=lt01 --shardsvr**

好，至此lt01分片更新完成，下面更新lt02分片。

#### 5.4.3.2更新lt02

lt02分片的Primary结点是20002，我们按照20003、20001、切换Primary、20002的顺序更新。

##### (1)更新lt02-20003端口结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:20003**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

**G:\mongodb-shard02repl-mongod3-20003\_new\_dir\data，这是lt分片的20003端口新数据目录。**

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=20003 --dbpath="G:\mongodb-shard02repl-mongod3-20003\_new\_dir\data" --replSet=lt02 --shardsvr**

##### (2)更新lt02-20001端口结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:20001**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

**G:\mongodb-shard02repl-mongod1-20001\_new\_dir\data，这是lt分片的20001端口新数据目录。**

**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=20001 --dbpath="G:\mongodb-shard02repl-mongod1-20001\_new\_dir\data" --replSet=lt02 --shardsvr**

##### (3)切换Primary

注意：再切换Primary结点之前，必须等待上面两个结点将数据同步完成。关注两个副本结点是健康状态的。

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:20002**

**rs.stepDown()**

**rs.status();**

##### (4)更新lt02-20002端口结点

**J:\mongodb-64-2.6.12\bin\mongo 127.0.0.1:20002**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

**G:\mongodb-shard02repl-mongod2-20002\_new\_dir\data，这是lt分片的20002端口新数据目录。**

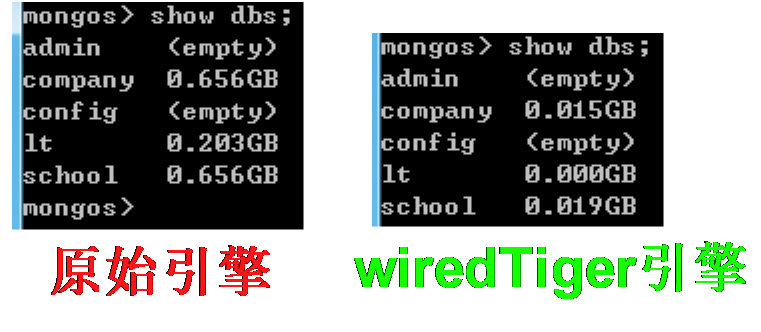
**J:\mongodb-64-3.0.10\bin\mongod --storageEngine=wiredTiger --port=20002 --dbpath="G:\mongodb-shard02repl-mongod2-20002\_new\_dir\data" --replSet=lt02 --shardsvr**

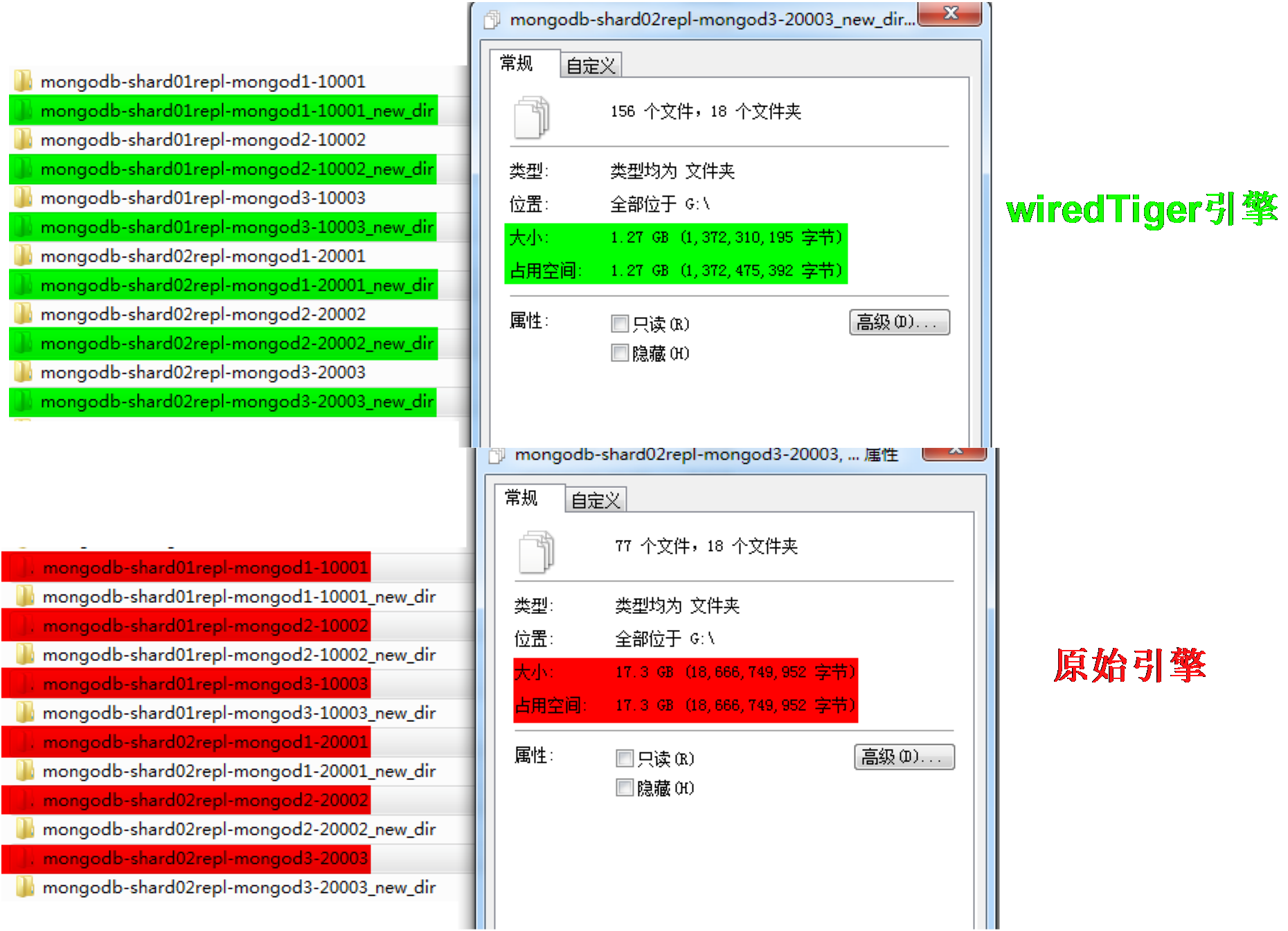
## 5.5总结

至此整个集群更新完毕，并且使用了新的存储引擎wiredTiger。使用新引擎更新过程颇为麻烦，需要非常小心！！！



如下是使用引擎前后数据库大小的变化：





# Q&A

带有身份认证的访问升级方式有待探索！

# 参考资料

1. https://docs.mongodb.com/manual/release-notes/3.0-compatibility/#legacy-auth-model-removed
2. https://docs.mongodb.com/manual/release-notes/3.0-upgrade/
3. https://docs.mongodb.com/manual/release-notes/2.6-upgrade-authorization/
4. https://docs.mongodb.com/manual/release-notes/3.0-compatibility/
5. https://docs.mongodb.com/manual/tutorial/change-standalone-wiredtiger/
6. https://docs.mongodb.com/manual/tutorial/change-replica-set-wiredtiger/
7. https://docs.mongodb.com/manual/tutorial/change-sharded-cluster-wiredtiger/
8. https://docs.mongodb.com/manual/tutorial/change-config-server-wiredtiger/