Mongodb 集群分片部署

分片部署是mongodb数据高可用的方案之一，还有就是master/slave模式，master/slave模式由下章节写出。这里将的分片就是集群中的单个节点或者节点的集合，只不过在mangodb中的叫法与关系型数据库不同。

mongoDB支持在多个机器中通过异步复制达到故障转移和实现冗余。多机器中同一时刻只有一台是用于写操作。正是由于这个情况，为mongoDB提供了数据一致性的保障。担当主角色的机器能把读操作分发给slaves/secondaries。

官网中为选择这两种架构提供了以下意见：  
\* if using <v1.6 : master/slave  
\* if need automatic fail-over and recovery (easy administration): replica sets  
\* if using --auth (security) : for now, master/slave  
\* if using sharding : either, but replica sets are best for clusters that are not small  
\* if risk averse : master/slave (replica sets are new to v1.6.0)

MongoDB 的数据分块称为 chunk。每个 chunk 都是 Collection 中一段连续的数据记录，通常最大尺寸是 200MB，超出则生成新的数据块。

要构建一个 MongoDB Sharding Cluster(集群分片部署)，需要三种角色：

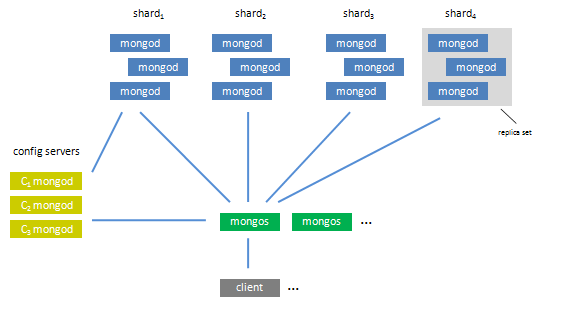
    Shard Server: mongod 实例，每个shard由一个或多个mongod进程组成，用于存储实际的数据块。

    Config Server: mongod 实例，存储了整个 Cluster Metadata，包括每个Shard的信息和chunks信息。

    Route Server: mongos 实例，前端路由，客户端由此接入，且让整个集群看上去像单一进程数据库。

Route 转发请求到实际的目标服务进程，并将多个结果合并回传给客户端。Route 本身并不存储任何数据和状态，仅在启动时从 Config Server 获取信息。Config Server 上的任何变动都会传递给所有的 Route Process。

在实际使用中，为了获取高可用、高性能的集群方案，我们会将 Shard Server 部署成 Replica Sets。



集群部署原理图

其中shard1,shard2…..这些就是上面说到的Shard Server，每个shard就是一个分片，分片最好部署到不同的机器上，这样才能有效解决单台服务器压力过大，或者宕机导致整个应用停止的问题。而且分片之间建议部署到同一个高速局域网中，这样才能有效提高数据在分片中的共享能力。如果分片之间的网络不是很理想，直接影响mongodb的查询能力。

其中 replica set就是多个shard的集合。

每台分片服务器需要个config server 即配置服务器

Mongos就是Route Server 即前端路由，接收来自客户端的所有数据请求，比如路由服务的IP是192.168.0.153:20000 这个应用程序中连接mongodb数据库的地址就是这个。

按照官方说的路由服务可以有多个，但是这个我没有亲自验证。

二．部署方案

关于部署方案在网上的资源也是很多。这里我只用一台机器模拟集群分片部署。

我只创建了一个分片，及set1，其中分片中由两个mongod进程组成。具体的命令说明请参考mongodb命令说明文档。

第一步：启动分片数据服务，

Shard1 （replSet set1）

192.168.0.153:10001 对应a.bat

mongod --shardsvr --dbpath E:\mongodbShared\a --logpath E:\mongodbShared\a\log\a.log --port 10001 --replSet set1

192.168.0.153:10002 对应 b.bat

mongod --shardsvr --dbpath E:\mongodbShared\b --logpath E:\mongodbShared\b\log\b.log --port 10002 --replSet set1

192.168.0.153:10003 决策服务 对应c.bat

只负责在某个node down掉后，进行vote选举新的master，它们本身并不存储数据备份

mongod --shardsvr --dbpath E:\mongodbShared\c --logpath E:\mongodbShared\c\log\c.log --port 10003 --replSet set1

第二步：配置10001 和 10002 这两个服务之间的关系及决策服务。

打开命令行

输入mongo --port 10001 （前提是需要把mongodb配置的环境变量中去）



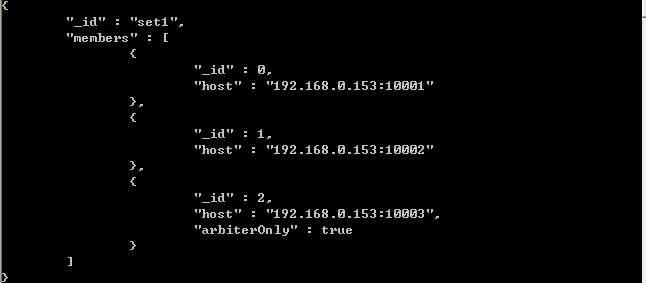


输入：

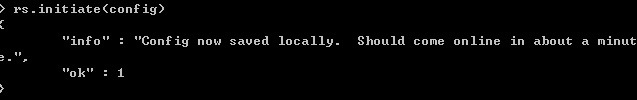
config={\_id:'set1',members:[{\_id:0,host:'192.168.0.153:10001'},{\_id:1,host:'192.168.0.153:10002'},{\_id:2,host:'192.168.0.153:10003',arbiterOnly:true}]}

注意：其中10003这里我加一个arbiterOnly:true 表明这是决策服务的意思

提示：



输入：rs.initiate(config) 进行配置的初始化

提示：

标明初始化成功，我们可以通过>rs.status()命令来查看这个分片的配置信息。



在其中咱们可以看见这个分片名称即“set1”及其成员的信息，其中需要咱们注意的是stateStr属性，可以看出在这个分片中，哪个服务是主服务primary哪个是辅助服务，哪个是决策服务。其中的health表示当前服务的状态，等于1就是正常状态，其他的我忘记了呵呵。

第三步：启动配置服务器

192.168.0.153:20000 对应 config.bat

mongod --configsvr --dbpath E:\mongodbShared\config --logpath E

:\mongodbShared\config\log\config.log --port 20000

第四步：启动路由服务

192.168.0.153:30000 对应 route.bat

mongos --chunkSize 1 --configdb "192.168.0.153:20000" --logpath

E:\mongodbShared\route\log\route.log –port 30000

其中 –chunkSize 块大小 1M 和配置服务器的地址

第五步：配置分片与分片的关系及配置集群部署哪个数据库

连接到路由服务上

命令行

输入：mongo –port 30000

  use admin  
    db.runCommand({addshard:'set1/192.168.0.153:10001,192.168.0.153:10002'})  
     db.runCommand({enablesharding:'test'})  
    db.runCommand({listshards:1})  
    printShardingStatus()  
    db.runCommand({shardcollection:'test.test', key:{\_id:1}, unique : true})

到此集群分片部署完毕。