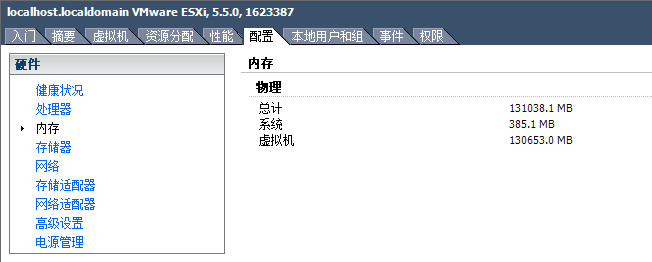
环境：浪潮服务器

内存：120G



硬盘：4T



总体架构：

分成6台服务器：

3台路由服务器

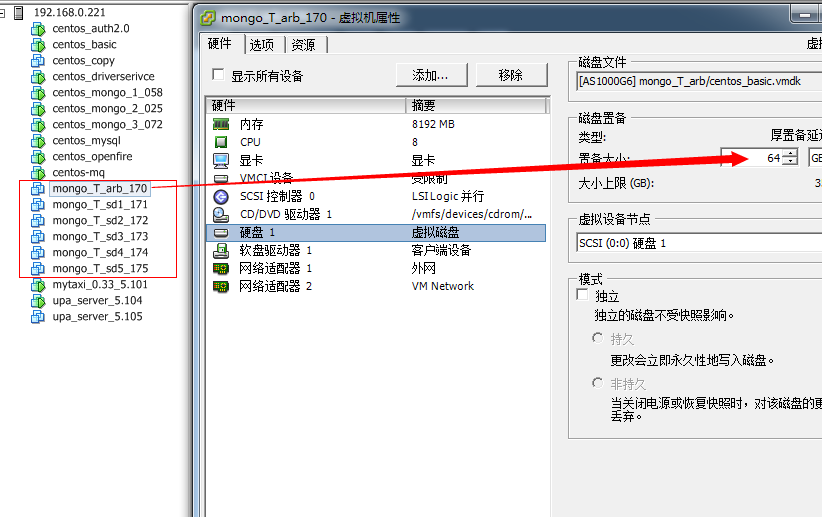
3台配置服务器

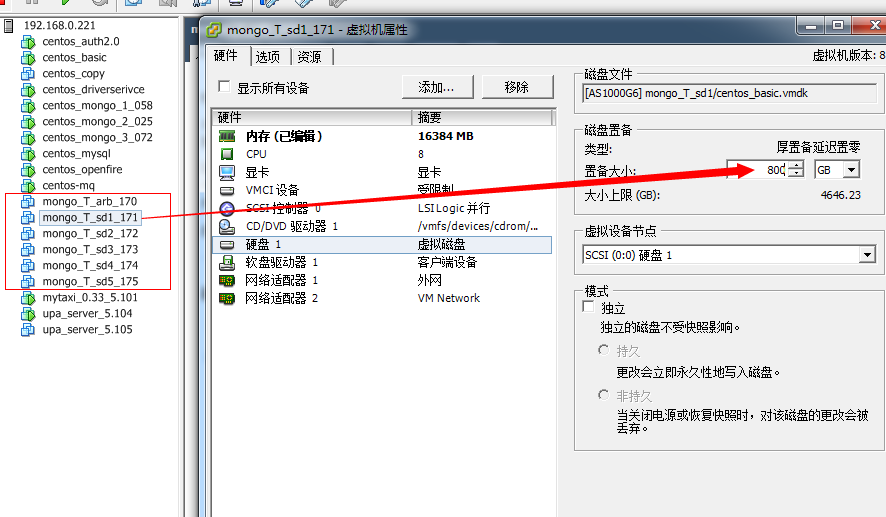
路由和配置服务器平分6台机器；

1台仲裁节点(64G)：用于某一主服务器挂掉，将可用服务自动升为主服务器的节点，制作投票，不进行数据的存储和读取

剩余5台(每台800G)；做5个分片；并将每个分片增加一个备份；

一：配置虚拟机；





二：配置网卡

170-175段

确保内网可以ping通

并开放端口如下：

-A INPUT -m state --state NEW -m tcp -p tcp --dport 27000 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 27001 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 27002 -j ACCEPT

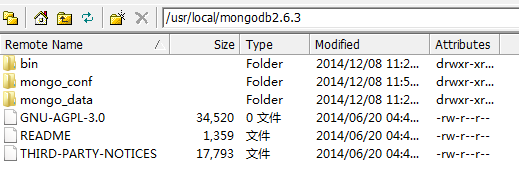
-A INPUT -m state --state NEW -m tcp -p tcp --dport 27003 -j ACCEPT

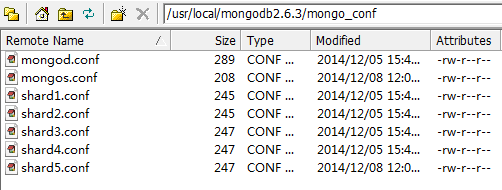
-A INPUT -m state --state NEW -m tcp -p tcp --dport 27004 -j ACCEPT

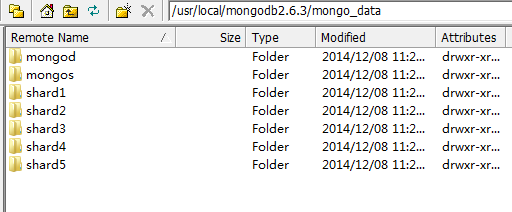
-A INPUT -m state --state NEW -m tcp -p tcp --dport 27005 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 27017 -j ACCEPT

三：安装mongodb







所有mongo\_data下的目录中都需要包含子目录data和logs

四、启动分片并配置

170、171、172为路由服务器

173、174、175为配置服务器

4.1分别启动配置服务：173、174、175

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/mongod.conf

mongod.conf内容如下：

configsvr = true

dbpath = /usr/local/mongodb2.6.3/mongodb\_data/mongod/data

logpath = /usr/local/mongodb2.6.3/mongodb\_data/mongod/logs/mongodb.log

logappend = true

directoryperdb = true

noauth = true

port = 27017

maxConns = 1024

fork = true

quota = true

quotaFiles = 1024

nssize = 16

4.2在每一台服务器分别启动mongos服务器：170、171、172

/usr/local/mongodb2.6.3/bin/mongos -f /usr/local/mongodb2.6.3/mongo\_conf/mongos.conf

mongos.conf内容如下：

configdb = 192.168.0.173:27017,192.168.0.174:27017,192.168.0.175:27017

logpath = /usr/local/mongodb2.6.3/mongo\_data/mongos/logs/mongodb.log

logappend = true

port = 27000

maxConns = 1024

fork = true

4.3启动各个分片的副本集

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard5.conf

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard1.conf

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard2.conf

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard3.conf

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard4.conf

/usr/local/mongodb2.6.3/bin/mongod -f /usr/local/mongodb2.6.3/mongo\_conf/shard5.conf

在仲裁节170点启动如下分片

sd1、sd2、sd3、sd4、sd5

在剩余5个分片个启动两个，如题结构图如下：

171：sd1、sd5副

172：sd2、sd1副

173：sd3、sd2副

174：sd4、sd3副

175：sd5、sd4副

4.4分别设置分片副本集：

分片1：

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.171:27001

#使用admin数据库

use admin

#定义副本集配置

config = { \_id:"shard1", members:[ {\_id:0,host:"192.168.0.171:27001",priority:2},{\_id:1,host:"192.168.0.172:27001",priority:1},{\_id:2,host:"192.168.0.170:27001",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

分片2：

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.172:27002

#使用admin数据库

use admin

#定义副本集配置

config = { \_id:"shard2", members:[ {\_id:0,host:"192.168.0.172:27002",priority:2},{\_id:1,host:"192.168.0.173:27002",priority:1},{\_id:2,host:"192.168.0.170:27002",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

分片3：

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.173:27003

#使用admin数据库

use admin

#定义副本集配置

config = { \_id:"shard3", members:[ {\_id:0,host:"192.168.0.173:27003",priority:2},{\_id:1,host:"192.168.0.174:27003",priority:1},{\_id:2,host:"192.168.0.170:27003",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

分片4：

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.174:27004

#使用admin数据库

use admin

#定义副本集配置

config = { \_id:"shard4", members:[ {\_id:0,host:"192.168.0.174:27004",priority:2},{\_id:1,host:"192.168.0.175:27004",priority:1},{\_id:2,host:"192.168.0.170:27004",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

分片5：

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.175:27005

#使用admin数据库

use admin

#定义副本集配置

config = { \_id:"shard5", members:[ {\_id:0,host:"192.168.0.175:27005",priority:2},{\_id:1,host:"192.168.0.171:27005",priority:1},{\_id:2,host:"192.168.0.170:27005",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

4.5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

连接到mongos

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.170:27000

使用admin数据库

use admin

串联路由服务器与分配副本集

　　db.runCommand( { addshard : "shard1/192.168.0.171:27001,192.168.0.172:27001,192.168.0.170:27001"});

　　db.runCommand( { addshard : "shard2/192.168.0.172:27002,192.168.0.173:27002,192.168.0.170:27002"});

　　db.runCommand( { addshard : "shard3/192.168.0.173:27003,192.168.0.174:27003,192.168.0.170:27003"});

　　db.runCommand( { addshard : "shard4/192.168.0.174:27004,192.168.0.175:27004,192.168.0.170:27004"});

　　db.runCommand( { addshard : "shard5/192.168.0.175:27005,192.168.0.171:27005,192.168.0.170:27005"});

db.runCommand( { listshards : 1 } );

4.6指定需要分片的数据库

目前配置服务、路由服务、分片服务、副本集服务都已经串联起来了，但我们的目的是希望插入数据，数据能够自动分片，就差那么一点点，一点点……

连接在mongos上，准备让指定的数据库、指定的集合分片生效。

指定testshard分片生效

db.runCommand( { enablesharding :"testshard1"});

指定数据库里需要分片的集合和片键（片键是为了设置当前记录分布在哪个分片中；这里采用hashed是尽量保证各个分片的数据量平均）

db.runCommand({shardcollection : "testshard1.t\_count",key : {"\_id": "hashed"}});

我们设置testshard的 t\_count 表需要分片，根据“\_id”自动分片到 shard1 ，shard2，shard3，shard4，shard5上面去。要这样设置是因为不是所有mongodb 的数据库和表 都需要分片!这里的\_id是mongodb自动生成的，不需要人为指定

4.7进行测试；测试分片配置结果

连接mongos服务器

/usr/local/mongodb2.6.3/bin/mongo 192.168.0.170:27000

使用testshard

use testshard;

插入测试数据

for (var i = 1; i <= 100000; i++) db.t\_count.save({id:i,"test1":"testval1"});

查看分片情况如下，部分无关信息省掉了

db.t\_count.stats();

{

"sharded" : true,

"systemFlags" : 1,

"userFlags" : 1,

"ns" : "testshard.t\_count",

"count" : 100000,

"numExtents" : 25,

"size" : 11200000,

"storageSize" : 13967360,

"totalIndexSize" : 8290464,

"indexSizes" : {

"\_id\_" : 3286752,

"\_id\_hashed" : 5003712

},

"avgObjSize" : 112,

"nindexes" : 2,

"nchunks" : 5,

"shards" : {

"shard1" : {

"ns" : "testshard.t\_count",

"count" : 19959,

"size" : 2235408,

"avgObjSize" : 112,

"storageSize" : 2793472,

"numExtents" : 5,

"nindexes" : 2,

"lastExtentSize" : 2097152,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 1635200,

"indexSizes" : {

"\_id\_" : 654080,

"\_id\_hashed" : 981120

},

"ok" : 1

},

"shard2" : {

"ns" : "testshard.t\_count",

"count" : 20289,

"size" : 2272368,

"avgObjSize" : 112,

"storageSize" : 2793472,

"numExtents" : 5,

"nindexes" : 2,

"lastExtentSize" : 2097152,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 1708784,

"indexSizes" : {

"\_id\_" : 670432,

"\_id\_hashed" : 1038352

},

"ok" : 1

},

"shard3" : {

"ns" : "testshard.t\_count",

"count" : 19872,

"size" : 2225664,

"avgObjSize" : 112,

"storageSize" : 2793472,

"numExtents" : 5,

"nindexes" : 2,

"lastExtentSize" : 2097152,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 1602496,

"indexSizes" : {

"\_id\_" : 654080,

"\_id\_hashed" : 948416

},

"ok" : 1

},

"shard4" : {

"ns" : "testshard.t\_count",

"count" : 19915,

"size" : 2230480,

"avgObjSize" : 112,

"storageSize" : 2793472,

"numExtents" : 5,

"nindexes" : 2,

"lastExtentSize" : 2097152,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 1651552,

"indexSizes" : {

"\_id\_" : 654080,

"\_id\_hashed" : 997472

},

"ok" : 1

},

"shard5" : {

"ns" : "testshard.t\_count",

"count" : 19965,

"size" : 2236080,

"avgObjSize" : 112,

"storageSize" : 2793472,

"numExtents" : 5,

"nindexes" : 2,

"lastExtentSize" : 2097152,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 1692432,

"indexSizes" : {

"\_id\_" : 654080,

"\_id\_hashed" : 1038352

},

"ok" : 1

}

},

"ok" : 1

}

可以看到数据分到5个分片，各自分片数量。已经成功了!

断电重启：

按照4.1、4.2、4.3的顺序

先启动三台配置服务器

再启动三台路由服务器

最后启动各个分片