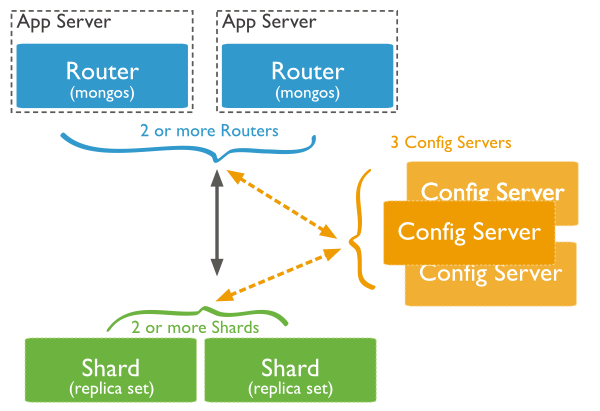
**MongoDB3集群搭建及验证**

**V3.0.7 (shard+replica)**

# 1、集群模块介绍

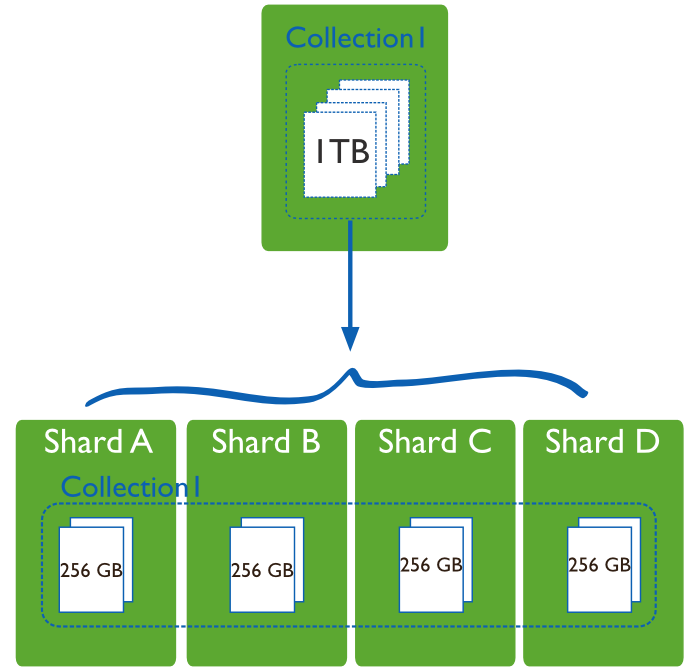
从MongoDB官方给的集群架构了解，整个集群主要有4个模块：Config Server、mongs、 shard、replica set。



**Config Server：**用来存放集群的元数据，也就是存放所有分片的配置数据，mongos第一次启动就需要连接configServer读取相关数据，当configServer有数据进行更新时，也会主动推送消息到所有的mongos上，在 3.0.7版本中，官方是建议配置3份的Config Server，以便挂掉两台时，业务还能够正常运转。

**mongs：**MongoDB集群的请求入口，能否自动实现数据的分布式分发，生产环境中建议部署在应用服务器上。

**shard：**分片就比如是将一张大表分散在几个不同的shard中，实现数据分布式存储。



**replica set：**主要是对每个分片进行冗余，生产环境中，一般将将副本集配置在三个节点上，两份副本、一份仲裁。

# 2、环境规划

服务器分别、端口及安装路径规划如下面两个表:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用途** | **IP** | **端口** | **备注** | **安装路径** |
| ConfigeServer | 172.16.16.120 | 30001 | 存放配置数据 | /db/configS |
| 172.16.16.121 | 30001 | /db/configS |
| 172.16.16.122 | 30001 | /db/configS |
| share1 | 172.16.16.124 | 40001 | Shard1主节点 | /db/shard1 |
| 172.16.16.125 | 40001 | Shard1副本节点 | /db/shard1 |
| 172.16.16.126 | 40001 | Shard1仲裁节点 | /db/shard1 |
| share2 | 172.16.16.125 | 40002 | Shard2主节点 | /db/shard2 |
| 172.16.16.126 | 40002 | Shard2副本节点 | /db/shard2 |
| 172.16.16.127 | 40002 | Shard2仲裁节点 | /db/shard2 |
| share3 | 172.16.16.126 | 40003 | Shard3主节点 | /db/shard3 |
| 172.16.16.127 | 40003 | Shard3副本节点 | /db/shard3 |
| 172.16.16.124 | 40003 | Shard3仲裁节点 | /db/shard3 |
| share4 | 172.16.16.127 | 40004 | Shard4主节点 | /db/shard4 |
| 172.16.16.124 | 40004 | Shard4副本节点 | /db/shard4 |
| 172.16.16.125 | 40004 | Shard4仲裁节点 | /db/shard4 |
| mongos | 172.16.16.124 | 50001 | 生产环境中一般直接部署在应用端 | /db/mongos |
| 172.16.16.125 | 50001 | /db/mongos |
| 172.16.16.126 | 50001 | /db/mongos |
| 172.16.16.127 | 50001 | /db/mongos |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Node** | **Config** | **Shard1** | | | **Shard2** | | | **Shard3** | | | **Shard4** | | | **Route** |
| **主** | **副** | **仲** | **主** | **副** | **仲** | **主** | **副** | **仲** | **主** | **副** | **仲** |
| 172.16.16.120 | ● |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 172.16.16.121 | ● |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 172.16.16.122 | ● |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 172.16.16.124 |  | ▲ |  |  |  |  |  |  |  | ◇ |  | ☆ |  | ■ |
| 172.16.16.125 |  |  | △ |  | ▼ |  |  |  |  |  |  |  | ☆ | ■ |
| 172.16.16.126 |  |  |  | △ |  | ▽ |  | ◆ |  |  |  |  |  | ■ |
| 172.16.16.127 |  |  |  |  |  |  | ▽ |  | ◇ |  | ★ |  |  | ■ |

# 3、搭建步骤

下载mongodb(<https://www.mongodb.org/downloads> ) ，这里使用的是 3.0.7版本。

|  |
| --- |
| # tar zxvf mongodb-linux-x86\_64-rhel55-3.0.7.gz  # mv mongodb-linux-x86\_64-rhel55-3.0.7 /usr/local/mongodb  # useradd mongo  # passwd mongo  Changing password for user mongo.  New UNIX password:  BAD PASSWORD: it is too simplistic/systematic  Retype new UNIX password:  passwd: all authentication tokens updated successfully.  # chown -R mongo:mongo /usr/local/mongodb/  # chown -R mongo:mongo /db |

## 3.1创建相关模块存放路径

#### 创建configeServer目录(172.16.16.120/121/122)

#mkdir -p /db/configS/data & mkdir -p /db/configS/log (存放ConfigServer的数据、日志)

#### 创建shard1目录(172.16.16.124/125/126)

#mkdir -p /db/shard1/data & mkdir -p /db/shard1/log (存放shard1的数据、日志)

#### 创建shard2目录(172.16.16.125/126/127 )

#mkdir -p /db/shard2/data & mkdir -p /db/shard2/log (存放shard2的数据、日志)

#### 创建shard3目录(172.16.16.126/127/124 )

#mkdir -p /db/shard3/data & mkdir -p /db/shard3/log (存放shard3的数据、日志)

#### 创建shard4目录(172.16.16.127/124/125 )

#mkdir -p /db/shard4/data & mkdir -p /db/shard4/log (存放shard4的数据、日志)

#### 创建mongos目录(172.16.16.124/125/126/127)

#mkdir -p /db/mongos/log (由于mongos只做路由使用，不存数据，所以只需要建立log目录)

## 3.2 模块配置及启动

#### configServer(172.16.16.120/121/122)配置及服务启动:

编写 /usr/local/mongodb/conf/configServer.conf，将参数都放在该文件中：

#vim /usr/local/mongodb/conf/configServer.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/configS/log/configServer.log" #日志存储位置

logAppend: true

storage:

journal: #journal配置

enabled: true

dbPath: "/db/configS/data" #数据文件存储位置

directoryPerDB: true #是否一个库一个文件夹

engine: wiredTiger #数据引擎

wiredTiger: #WT引擎配置

engineConfig:

cacheSizeGB: 6 #设置为6G,默认为物理内存的一半

directoryForIndexes: true #是否将索引也按数据库名单独存储

journalCompressor: zlib

collectionConfig: #表压缩配置

blockCompressor: zlib

indexConfig: #索引配置

prefixCompression: true

net: #端口配置

port: 30001 #另外两台需要分别修改为30002、30003

processManagement: #配置启动管理方式

fork: true

sharding: #分片配置

clusterRole: configsvr #分片角色

启动configServer：

$ /usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/configServer.conf

#### mongos(172.16.16.124/125/126/127)配置及服务启动

编写mongos.conf，将参数都放在该文件中(4台配置文件都一样)：

#vim /usr/local/mongodb/conf/mongos.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/mongos/log/mongos.log"

logAppend: true

net:

port: 50001

sharding:

configDB: 172.16.16.120:30001,172.16.16.121:30001,172.16.16.122:30001

processManagement:

fork: true

启动mongos：应保证集群中设备的时间都是一样的，否则启动mongos会报错，若不相同，可先搭建一套NTP服务器

$ /usr/local/mongodb/bin/mongos -f /usr/local/mongodb/conf/mongos.conf

#### ****shard1分片+副本集配置及服务启动(****172.16.16.124/125/126 ****)：****

#vim /usr/local/mongodb/conf/shard1.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/shard1/log/shard1.log" #日志存储位置

logAppend: true

storage:

journal: #journal配置

enabled: true

dbPath: "/db/shard1/data" #数据文件存储位置

directoryPerDB: true #是否一个库一个文件夹

engine: wiredTiger #数据引擎

wiredTiger: #WT引擎配置

engineConfig:

cacheSizeGB: 6 #设置为6G,默认为物理内存的一半

directoryForIndexes: true #是否将索引也按数据库名单独存储

journalCompressor: zlib

collectionConfig: #表压缩配置

blockCompressor: zlib

indexConfig: #索引配置

prefixCompression: true

net: #端口配置

port: 40001

processManagement: #配置启动管理方式

fork: true

sharding: #分片配置

clusterRole: shardsvr

**replication:**

replSetName: shard1 #配置副本集名称

启动shard1 mongod：

$ /usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/shard1.conf

#### ****shard2分片+副本集配置及服务启动(172.16.16.125/126/127 )：****

#vim /usr/local/mongodb/conf/shard2.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/shard2/log/shard2.log" #日志存储位置

logAppend: true

storage:

journal: #journal配置

enabled: true

dbPath: "/db/shard2/data" #数据文件存储位置

directoryPerDB: true #是否一个库一个文件夹

engine: wiredTiger #数据引擎

wiredTiger: #WT引擎配置

engineConfig:

cacheSizeGB: 6 #设置为6G,默认为物理内存的一半

directoryForIndexes: true #是否将索引也按数据库名单独存储

journalCompressor: zlib

collectionConfig: #表压缩配置

blockCompressor: zlib

indexConfig: #索引配置

prefixCompression: true

net: #端口配置

port: 40002

processManagement: #配置启动管理方式

fork: true

sharding: #分片配置

clusterRole: shardsvr

replication:

#oplogSizeMB:

replSetName: shard2 #配置副本集名称

启动shard2 mongod：

$ /usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/shard2.conf

#### shard3分片+副本集配置及服务启动****(1****72.16.16.126/127/124 ****)：****

#vim /usr/local/mongodb/conf/shard3.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/shard3/log/shard3.log" #日志存储位置

logAppend: true

storage:

journal: #journal配置

enabled: true

dbPath: "/db/shard3/data" #数据文件存储位置

directoryPerDB: true #是否一个库一个文件夹

engine: wiredTiger #数据引擎

wiredTiger: #WT引擎配置

engineConfig:

cacheSizeGB: 6 #设置为6G,默认为物理内存的一半

directoryForIndexes: true #是否将索引也按数据库名单独存储

journalCompressor: zlib

collectionConfig: #表压缩配置

blockCompressor: zlib

indexConfig: #索引配置

prefixCompression: true

net: #端口配置

port: 40003

processManagement: #配置启动管理方式

fork: true

sharding: #分片配置

clusterRole: shardsvr

replication:

#oplogSizeMB:

replSetName: shard3 #配置副本集名称

启动shara3 mongod：

conf]$ /usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/shard3.conf

#### ****shard4分片+副本集配置及服务启动(1****72.16.16.127/124/125 ****)：****

#vim /usr/local/mongodb/conf/shard4.conf

#!/bin/bash

systemLog:

destination: file

path: "/db/shard4/log/shard4.log" #日志存储位置

logAppend: true

storage:

journal: #journal配置

enabled: true

dbPath: "/db/shard4/data" #数据文件存储位置

directoryPerDB: true #是否一个库一个文件夹

engine: wiredTiger #数据引擎

wiredTiger: #WT引擎配置

engineConfig:

cacheSizeGB: 6 #设置为6G,默认为物理内存的一半

directoryForIndexes: true #是否将索引也按数据库名单独存储

journalCompressor: zlib

collectionConfig: #表压缩配置

blockCompressor: zlib

indexConfig: #索引配置

prefixCompression: true

net: #端口配置

port: 40004

processManagement: #配置启动管理方式

fork: true

sharding: #分片配置

clusterRole: shardsvr

replication:

#oplogSizeMB:

replSetName: shard4 #复制集名

启动shara4 mongod：

$ /usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/shard4.conf

## 3.2 集群副本集配置

副本集配置(在每个 shard的主节点上进行配置及初始化，否则会初始化失败 )：

#### shard1的副本集配置(主节点、副本节点、仲裁节点)：

$ ./mongo 172.16.16.124:40001

MongoDB shell version: 3.0.7

connecting to: 172.16.16.124:40001/test

> use admin

switched to db admin

> config = { \_id:"shard1", members:[

{\_id:0,host:"172.16.16.124:40001"},

{\_id:1,host:"172.16.16.125:40001"},

{\_id:2,host:"172.16.16.126:40001",arbiterOnly:true}]

}

#以下为输出

{

"\_id" : "shard1",

"members" : [

{

"\_id" : 0,

"host" : "172.16.16.124:40001"

},

{

"\_id" : 1,

"host" : "172.16.16.125:40001"

},

{

"\_id" : 2,

"host" : "172.16.16.126:40001",

"arbiterOnly" : true

}

]

}

> rs.initiate(config); #初始化配置

{ "ok" : 1 }

#### shard2的副本集配置(主节点、副本节点、仲裁节点)：

$ ./mongo 172.16.16.125:40002

MongoDB shell version: 3.0.7

connecting to: 172.16.16.125:40002/test

> use admin

switched to db admin

> config = { \_id:"shard2", members:[

{\_id:0,host:"172.16.16.125:40002"},

{\_id:1,host:"172.16.16.126:40002"},

{\_id:2,host:"172.16.16.127:40002",arbiterOnly:true}]

}

#以下为输出

{

"\_id" : "shard2",

"members" : [

{

"\_id" : 0,

"host" : "172.16.16.125:40002"

},

{

"\_id" : 1,

"host" : "172.16.16.126:40002"

},

{

"\_id" : 2,

"host" : "172.16.16.127:40002",

"arbiterOnly" : true

}

]

}

> rs.initiate(config); #初始化配置

{ "ok" : 1 }

#### shard3的副本集配置(主节点、副本节点、仲裁节点)：

$ ./mongo 172.16.16.126:40003

MongoDB shell version: 3.0.7

connecting to: 172.16.16.126:40003/test

> use admin

switched to db admin

> config = { \_id:"shard3", members:[

{\_id:0,host:"172.16.16.126:40003"},

{\_id:1,host:"172.16.16.127:40003"},

{\_id:2,host:"172.16.16.124:40003",arbiterOnly:true}]

}

#以下为输出

{

"\_id" : "shard3",

"members" : [

{

"\_id" : 0,

"host" : "172.16.16.126:40003"

},

{

"\_id" : 1,

"host" : "172.16.16.127:40003"

},

{

"\_id" : 2,

"host" : "172.16.16.124:40003",

"arbiterOnly" : true

}

]

}

> rs.initiate(config); #初始化配置

{ "ok" : 1 }

#### shard4的副本集配置(主节点、副本节点、仲裁节点)：

$ ./mongo 172.16.16.127:40004

MongoDB shell version: 3.0.7

connecting to: 172.16.16.127:40004/test

> use admin

switched to db admin

> config = { \_id:"shard4", members:[

{\_id:0,host:"172.16.16.127:40004"},

{\_id:1,host:"172.16.16.124:40004"},

{\_id:2,host:"172.16.16.125:40004",arbiterOnly:true}]

}

#以下为输出

{

"\_id" : "shard4",

"members" : [

{

"\_id" : 0,

"host" : "172.16.16.127:40004"

},

{

"\_id" : 1,

"host" : "172.16.16.124:40004"

},

{

"\_id" : 2,

"host" : "172.16.16.125:40004",

"arbiterOnly" : true

}

]

}

> rs.initiate(config); #初始化配置

{ "ok" : 1 }

## 3.4 集群分片配置

$ ./mongo 172.16.16.124:50001

mongos> use admin

switched to db admin

mongos> db.runCommand({addshard:"shard1/172.16.16.124:40001,172.16.16.125:40001,172.16.16.126:40001"});

{ "shardAdded" : "shard1", "ok" : 1 }

mongos>db.runCommand({addshard:"shard2/172.16.16.125:40002,172.16.16.126:40002,172.16.16.127:40002"});

{ "shardAdded" : "shard2", "ok" : 1 }

mongos>db.runCommand({addshard:"shard3/172.16.16.126:40003,172.16.16.127:40003,172.16.16.124:40003"});

{ "shardAdded" : "shard3", "ok" : 1 }

mongos>db.runCommand({addshard:"shard4/172.16.16.127:40004,172.16.16.124:40004,172.16.16.125:40004"});

{ "shardAdded" : "shard4", "ok" : 1 }

## 3.5 查看集群配置是否生效

注：查看集群配置是否生效时仲裁不被列出 。

mongos> db.runCommand( { listshards : 1 } );

{

"shards" : [

{

"\_id" : "shard1",

"host" : "shard1/172.16.16.124:40001,172.16.16.125:40001"

},

{

"\_id" : "shard2",

"host" : "shard2/172.16.16.125:40002,172.16.16.126:40002"

},

{

"\_id" : "shard3",

"host" : "shard3/172.16.16.126:40003,172.16.16.127:40003"

},

{

"\_id" : "shard4",

"host" : "shard4/172.16.16.124:40004,172.16.16.127:40004"

}

],

"ok" : 1

}

以上就完成了MongoDB shard+replica模式的集群搭建，接下来做业务测试。

## 3.6集群测试

默认情况下，库和集合是没有自动分片的，若有数据写入，只会往一个shard中存储，做个测试验证：

### 3.6.1数据写入测试：

$ ./mongo 172.16.16.127:50001

MongoDB shell version: 3.0.7

connecting to: 172.16.16.127:50001/test

mongos> use tangdb

switched to db tangdb

mongos> for (var i=1;i<=10000;i++) db.tangtable.save({"name":"ljai","age":27,"addr":"fuzhou"})

WriteResult({ "nInserted" : 1 })

mongos> db.tangtable.stats()

{

"sharded" : false,

"primary" : "shard1",

"ns" : "tangdb.tangtable",

"count" : 10000,

"size" : 670000,

"avgObjSize" : 67,

"storageSize" : 49152,

"capped" : false,

"wiredTiger" : {

"metadata" : {

"formatVersion" : 1

}

mongos> db.printShardingStatus()

--- Sharding Status ---

sharding version: {

"\_id" : 1,

"minCompatibleVersion" : 5,

"currentVersion" : 6,

"clusterId" : ObjectId("5625fc29e3c17fdff8517b73")

}

shards:

{ "\_id" : "shard1", "host" : "shard1/172.16.16.124:40001,172.16.16.125:40001" }

{ "\_id" : "shard2", "host" : "shard2/172.16.16.125:40002,172.16.16.126:40002" }

{ "\_id" : "shard3", "host" : "shard3/172.16.16.126:40003,172.16.16.127:40003" }

{ "\_id" : "shard4", "host" : "shard4/172.16.16.124:40004,172.16.16.127:40004" }

balancer:

Currently enabled: yes

Currently running: yes

Balancer lock taken at Tue Oct 20 2015 21:01:26 GMT+0800 (CST) by DataServer-04:50001:1445330413:1804289383:Balancer:846930886

Failed balancer rounds in last 5 attempts: 0

Migration Results for the last 24 hours:

No recent migrations

databases:

{ "\_id" : "admin", "partitioned" : false, "primary" : "config" }

{ "\_id" : "test", "partitioned" : false, "primary" : "shard1" }

{ "\_id" : "tangdb", "partitioned" : false, "primary" : "shard1" }

可以看出tangdb库并没有分片，且数据都在shard1上，登录其他shard1上查看：

$ ./mongo 172.16.16.124:40001

MongoDB shell version: 3.0.7

connecting to: 172.16.16.124:40001/test

shard1:PRIMARY> show dbs

tangdb 0.000GB

local 0.000GB

shard1:PRIMARY> use tangdb

switched to db tangdb

shard1:PRIMARY> show tables

tangtable

shard1:PRIMARY> db.tangtable.find().count()

10000

验证shard2、shard3、shard4上都没有tangdb这个库：

$ ./mongo 172.16.16.125:40002

MongoDB shell version: 3.0.7

connecting to: 172.16.16.125:40002/test

shard2:PRIMARY> show dbs

local 0.000GB

### 3.6.2指定数据库和集合进行分片：

为了让某个数据库与集合自动分片生效，对数据库(mydb)及数据库(mytable)中的表进行分片配置：

$ ./mongo 172.16.16.124:50001

MongoDB shell version: 3.0.7

connecting to: 172.16.16.124:50001/test

mongos> use admin

switched to db admin

mongos> db.runCommand( { enablesharding :"mydb"});

{ "ok" : 1 }

mongos> db.runCommand( { shardcollection : "mydb.mytable",key : {\_id: 1} } )

{ "collectionsharded" : "mydb.mytable", "ok" : 1 }