mongodb sharding

目录

简介

> 副本集(replica)
>
> 副本集由若干台服务器组成，分为三种角色：主服务器、副服务器、仲裁服务器。根据集群搭建的需求，仲裁服务器不是必需的。主服务器提供主要的对外读写的功能，副服务器作为备份。当主服务器不可用时，其余服务器根据投票选出一个新的主服务器，提供读写功能。因此，副本集可以提高集群的可用性。
>
> 分片(sharding)
>
> 分片主要是为减小高数据量和高吞吐量的数据库应用对单机性能造成的压力。将大的数据分片存储在不同节点上，外部读写只操作相应的一个或一小部分节点，一次减少每个分片节点村春的数据量和处理的请求数 。
>
> 注意：在生产环境中，配置服务器务必使用三个，而不是一个；每个分片节点都部署成副本集，而不是一个单独的Mongo服务器

mongodb 集群版分片功能描述

是否启动库分片	是否启动集合分片	是否实现分片
是	创建片键—启动集合分片功能—插入数据	是
是	未创建片键—插入数据—创建片键—启动集合分片	新插入数据实现
否	插入数据—开启库分片—创建片键—启动集合分片	新插入数据实现

```
# 涉及三台服务器
[root@appserver ~]# cat /etc/hosts
1.1.1.10 appserver
1.1.1.11 mysql1
1.1.1.12 mysql2

[root@mysql1 ~]# cat /etc/hosts
1.1.1.10 appserver
1.1.1.11 mysql1
1.1.1.12 mysql2

[root@mysql2 ~]# cat /etc/hosts
1.1.1.10 appserver
1.1.1.11 mysql1
1.1.1.12 mysql2

# 系统信息
[root@mysql2 ~]# uname -a
Linux mysql2 3.10.0-123.el7.x86_64 #1 SMP Mon Jun 30 12:09:22 UTC 2014 x86_64 x86_64 x86_64
GNU/Linux/
```

一、安装准备

```
# 三台服务器部署相同
统一安装路径：/alidata/mogodb
mongodb数据库版本：3.2.16
从3.0 版本起，默认只有 local 库，没有admin 库，需要我们自己来创建。

# 安装脚本
http://git.jiagouyun.com/huangjx/huangjx/blob/master/mongodb%E9%9B%86%E7%BE%A4%E7%89%88/mongodb%E
7%AE%A1%E7%90%86/mongodb-3.2.16.sh

# 启动关闭脚本
http://git.jiagouyun.com/huangjx/huangjx/blob/master/mongodb%E9%9B%86%E7%BE%A4%E7%89%88/mongodb%E
7%AE%A1%E7%90%86/mongodb.server

cd /alidata/mongodb/
mkdir {conf,log,data}
useradd -u 600 -s /bin/false mongodb
chown -R mongodb.mongodb /alidata/mongodb/
mkdir -p data/shard/s0
mkdir -p data/configdb
```

二、configserver配置

```
# 配置服务器

[root@appserver conf]# cat mongod.conf
logpath=/alidata/mongodb/log/config.log
pidfilepath=/alidata/mongodb/config.pid
logappend=true
fork=true
port=27020
bind_ip=1.1.1.10 #每台服务器改成自己的IP即可
dbpath=/alidata/mongodb/data/configdb
configsvr=true


[root@appserver mongo]# bash mongodb.server start
about to fork child process, waiting until server is ready for connections.
forked process: 8943
child process started successfully, parent exiting

#注：三台都配置configserver
```

三、routeserver启动

```
[root@appserver mongo]# mongos --port 40000 --configdb appserver:27020,mysql1:27020,mysql2:27020 --fork --logpath=/alidata/mongodb/log/route.log --chunkSize 500 &
[1] 9077
[root@appserver mongo]# about to fork child process, waiting until server is ready for connections.
forked process: 9079
child process started successfully, parent exiting

[1]+  完成                  mongos --port 40000 --configdb appserver:27020,mysql1:27020,mysql2:27020 --fork --logpath=/alidata/mongodb/log/route.log --chunkSize 500

注：每台主机时间要一致。
```

四、shardserver配置

```
# shard配置文件
[root@appserver conf]# cat mongod_shard.conf
logpath=/alidata/mongodb/log/s0.log
pidfilepath=/alidata/mongodb/mongo.pid
logappend=true
fork=true
port=27017
bind_ip=1.1.1.10 ##每台服务器改成自己的IP即可
dbpath=/alidata/mongodb/data/shard/s0
shardsvr=true


# 启动shard
[root@appserver mongo]# bash mongodb_shard.server start
about to fork child process, waiting until server is ready for connections.
forked process: 9240
child process started successfully, parent exiting
```

五、注册分片

```
#查看状态
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("59a68f4f2c9536c6111a4cf8")
}
  shards:
  active mongoses:
        "3.2.16" : 1
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:

注册分片1
mongos> use admin;
switched to db admin
mongos> db.runCommand({addshard:"1.1.1.10:27017"});
{ "shardAdded" : "shard0000", "ok" : 1 }

注册分片2
mongos> db.runCommand({addshard:"1.1.1.11:27017"});
{ "shardAdded" : "shard0001", "ok" : 1 }

注册分片3
mongos> db.runCommand({addshard:"1.1.1.12:27017"});
{ "shardAdded" : "shard0002", "ok" : 1 }

mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("59a68f4f2c9536c6111a4cf8")
}
  shards:
        {  "_id" : "shard0000",  "host" : "1.1.1.10:27017" }
        {  "_id" : "shard0001",  "host" : "1.1.1.11:27017" }
        {  "_id" : "shard0002",  "host" : "1.1.1.12:27017" }
  active mongoses:
        "3.2.16" : 1
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
```

```
                No recent migrations
  databases:

# 添加完成
```

六、查看系统相关信息

```
mongos> show dbs
config  0.000GB
mongos> use config
switched to db config

mongos> show collections;
changelog
chunks
lockpings
locks
mongos
settings
shards
tags
version

mongos> db.mongos.find()
{ "_id" : "appserver:40000", "ping" : ISODate("2017-08-30T10:41:15.943Z"), "up" :
NumberLong(1788), "waiting" : true, "mongoVersion" : "3.2.16" }

mongos> db.shards.find()
{ "_id" : "shard0000", "host" : "1.1.1.10:27017" }
{ "_id" : "shard0001", "host" : "1.1.1.11:27017" }
{ "_id" : "shard0002", "host" : "1.1.1.12:27017" }
```

七、分片

```
#启动数据库分片功能
mongos> use admin
mongos> sh.enableSharding("hjxdb");
 ##或者
mongos> db.runCommand( { enableSharding: "hjxdb"} )

#创建数据库
mongos> use hjxdb

#创建集合
mongos> db.createCollection("user");

#创建索引
mongos> use hjxdb
mongos> db.user.ensureIndex({"username" : 1})

 ##循环插入数据
mongos> for(var i=0;i<1000;i++){ db.user.insert({"username" : "user"+i,"create_at" : new Date()});
}

mongos> sh.status()
--- Sharding Status ---
       省略
  databases:
        {  "_id" : "hjxdb",  "primary" : "shard0000",  "partitioned" : true }
                hjxdb.user
                        shard key: { "username" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                shard0000       1
                        { "username" : { "$minKey" : 1 } } -->> { "username" : { "$maxKey" : 1 } }
on : shard0000 Timestamp(1, 0)

#启动集合分片功能
mongos> use admin
mongos> db.runCommand({ shardcollection: "hjxdb.user", key: {username:1}});

 ##循环插入数据
mongos> for(var i=1000;i<100000;i++){ db.user.insert({"username" : "user"+i,"create_at" : new
Date()}); }

mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("59a61ef8e2d56af97541e121")
}
  shards:
        {  "_id" : "shard0000",  "host" : "localhost:27000" }
        {  "_id" : "shard0001",  "host" : "localhost:27001" }
```

```
        {  "_id" : "shard0002",  "host" : "localhost:27002" }
  active mongoses:
        "3.2.16" : 1
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                26 : Success
  databases:
        {  "_id" : "test",  "primary" : "shard0000",  "partitioned" : true }
                hjxdb.user
                        shard key: { "username" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                shard0000       1088
                                shard0001       12
                                shard0002       12
                        too many chunks to print, use verbose if you want to force print
```

八、删除分片

```
#删除分片
mongos> db.runCommand( { removeshard: "shard0000" } ) #持续执行查看chunk删除状态
mongos> db.runCommand( { movePrimary: "test", to: "shard0001" }) #启动集合到其他分片
{ "primary" : "shard0001:localhost:27001", "ok" : 1 }
mongos> db.runCommand({removeshard:"shard0000"})
{
        "msg" : "removeshard completed successfully",
        "state" : "completed",
        "shard" : "shard0000",
        "ok" : 1
}
```