

Mongodb 配置 Master-Slave

编写者：何永安

时间：2016.05.26

目 录

| | |
|--|---|
| 1.规划如下..... | 3 |
| 2.配置主从..... | 3 |
| 2.1 编辑 Master 配置文件..... | 3 |
| 2.2 编辑 Slave1 配置文件..... | 3 |
| 2.3 编辑 Slave2 配置文件..... | 4 |
| 2.4 编辑 Slave3 配置文件..... | 4 |
| 3.启动实例..... | 5 |
| 4.测试..... | 5 |
| 4.1 登录主库..... | 5 |
| 4.2 登录从库 slave1，执行..... | 5 |
| 4.3 登录从库 slave2，执行..... | 6 |
| 4.4 登录从库 slave3，执行..... | 6 |
| 4.5 尝试在从库 slave1 插入数据库..... | 6 |
| 4.6 先把 slave1 进程结束，再在 master 插入数据，然后在启动 slave1，查看是否能得到 slave1 不在线时，master 插入的数据能否同步..... | 7 |

1.规划如下

| 服务器 IP | 服务器角色 | 端口 | 数据目录 | 日志目录 |
|------------------|--------|-------|---|--|
| 192. 168. 5. 201 | Master | 27017 | /usr/local/mongodb/mongodb-2. 4. 9-instance1/dbdata/master-data | /usr/local/mongodb/mongodb-2. 4. 9-instance1/dblogs/master-log |
| 192. 168. 5. 201 | Slave1 | 27018 | /usr/local/mongodb/mongodb-2. 4. 9-instance2/dbdata/slave1-data | /usr/local/mongodb/mongodb-2. 4. 9-instance2/dblogs/slave1-log |
| 192. 168. 5. 201 | Slave2 | 27019 | /usr/local/mongodb/mongodb-2. 4. 9-instance3/dbdata/slave2-data | /usr/local/mongodb/mongodb-2. 4. 9-instance3/dblogs/slave2-log |
| 192. 168. 5. 201 | Slave3 | 27020 | /usr/local/mongodb/mongodb-2. 4. 9-instance4/dbdata/slave3-data | /usr/local/mongodb/mongodb-2. 4. 9-instance4/dblogs/slave3-log |

2.配置主从

2.1 编辑 Master 配置文件

```
cd /usr/local/mongodb/mongodb-2. 4. 9-instance1/conf
vi MasterSlaveModule-Master.conf
内容如下：
logappend=true
fork=true
port=27017
dbpath=/usr/local/mongodb/mongodb-2. 4. 9-instance1/dbdata/master-data
logpath=/usr/local/mongodb/mongodb-2. 4. 9-instance1/dblogs/master/log.out
pidfilepath=/usr/local/mongodb/mongodb-2. 4. 9-instance1/pid/master/MasterSlaveModule-Master.
pid
rest=true
maxConns=1024
master=true          #将指定其为 master
oplogSize=2048
```

2.2 编辑 Slave1 配置文件

```
cd /usr/local/mongodb/mongodb-2. 4. 9-instance2/conf
vi MasterSlaveModule-Slave1.conf
内容如下：
```

```
logappend=true
fork=true
port=27018
dbpath=/usr/local/mongodb/mongodb-2.4.9-instance2/dbdata/slave1-data
logpath=/usr/local/mongodb/mongodb-2.4.9-instance2/dblogs/slave1-log/log.out
pidfilepath=/usr/local/mongodb/mongodb-2.4.9-instance2/pid/slave1/MasterSlaveModule-Slave1.pid
rest=true
maxConns=1024
slave=true
source=192.168.5.201:27017 #指定 master 的主机地址及端口号
autoresync=true
oplogSize=2048
```

2.3 编辑 Slave2 配置文件

```
cd /usr/local/mongodb/mongodb-2.4.9-instance3/conf
vi MasterSlaveModule-Slave2.conf
内容如下：
logappend=true
fork=true
port=27019
dbpath=/usr/local/mongodb/mongodb-2.4.9-instance3/dbdata/slave2-data
logpath=/usr/local/mongodb/mongodb-2.4.9-instance3/dblogs/slave2-log/log.out
pidfilepath=/usr/local/mongodb/mongodb-2.4.9-instance3/pid/slave2/MasterSlaveModule-Slave2.pid
rest=true
maxConns=1024
slave=true
source=192.168.5.201:27017 #指定 master 的主机地址及端口号
autoresync=true
oplogSize=2048
```

2.4 编辑 Slave3 配置文件

```
cd /usr/local/mongodb/mongodb-2.4.9-instance4/conf
vi MasterSlaveModule-Slave3.conf
内容如下：
logappend=true
fork=true
port=27020
```

```
dbpath=/usr/local/mongodb/mongodb-2.4.9-instance4/dbdata/slave3-data
logpath=/usr/local/mongodb/mongodb-2.4.9-instance4/dblogs/slave3-log/log.out
pidfilepath=/usr/local/mongodb/mongodb-2.4.9-instance4/pid/slave3/MasterSlaveModule-Slave3.pid
rest=true
maxConns=1024
slave=true
source=192.168.5.201:27017 #指定 master 的主机地址及端口号
autoresync=true
oplogSize=2048
```

3.启动实例

```
./mongod -f /usr/local/mongodb/mongodb-2.4.9-instance1/conf/MasterSlaveModule-Master.conf
./mongod -f /usr/local/mongodb/mongodb-2.4.9-instance2/conf/MasterSlaveModule-Slave1.conf
./mongod -f /usr/local/mongodb/mongodb-2.4.9-instance3/conf/MasterSlaveModule-Slave2.conf
./mongod -f /usr/local/mongodb/mongodb-2.4.9-instance4/conf/MasterSlaveModule-Slave3.conf
```

4.测试

4.1 登录主库

```
./mongo 192.168.5.201:27017
执行：
use mydb
db.mydb.insert({"name":"ABCEDFG"})
```

4.2 登录从库 slave1，执行

```
use mydb
db.mydb.find()
结果：
/* 0 */
{
  "_id" : ObjectId("5745cdd12780cf69714e0b52"),
  "name" : "ABCEDFG"
}
```

4.3 登录从库 slave2，执行

```
use mydb
db.mydb.find()
结果：
/* 0 */
{
  "_id" : ObjectId("5745cdd12780cf69714e0b52"),
  "name" : "ABCEDFG"
}
```

4.4 登录从库 slave3，执行

```
use mydb
db.mydb.find()
结果：
{
  "_id" : ObjectId("5745cdd12780cf69714e0b52"),
  "name" : "ABCEDFG"
}
```

从 4.1 - 4.4 可以看出同步成功！

4.5 尝试在从库 slave1 插入数据库

```
db.mydb.insert({"name":"AAAA"});
not master
```

从 4.5 可以看出，无法在从库插入数据！

4.6 先把 slave1 进程结束，再在 master 插入数据，然后在启动 slave1，查看是否能得到 slave1 不在线时，master 插入的数据能否同步

(1) 结束 slave

(2) 在主库插入数据

```
db.mydb.insert({"name":"DDDDDDDDDDDDDD"});
```

(3) 启动从库 slave1

```
./mongod -f /usr/local/mongodb/mongodb-2.4.9-instance2/conf/MasterSlaveModule-Slave1.conf
```

(4) 连接从库 slave1，执行

```
db.mydb.find();
```

结果：

| Key | Value | Type |
|--|--------------------------------------|----------|
| ▼ (1) ObjectId("5745c917a4e72ce0a738c4f2") | { 2 fields } | Object |
| _id | ObjectId("5745c917a4e72ce0a738c4f2") | ObjectId |
| name | tom | String |
| ▼ (2) ObjectId("5745d07e2780cf69714e0b54") | { 2 fields } | Object |
| _id | ObjectId("5745d07e2780cf69714e0b54") | ObjectId |
| name | DDDDDDDDDDDDDD | String |

由此可见：从库掉线期间，主库插入的数据，等从库在线时，会自动再同步！