

mycat数据库代理配置

一：介绍

mycat的具体概念以及作用这里不作概述，相关内容的了解和学习可以参考

- mycat开源社区：<http://www.mycat.io/>
- mycat文档：<http://www.mycat.io/document/mycat-definitive-guide.pdf>

这里介绍怎么使用mycat配置两个mysql数据库节点，并且在其中一个节点宕机的时候自动切换到另外的mysql数据库节点。

二：Mysql部署

使用docker创建两个双机热备的mysql数据库，具体如何设置主从互备参考另一份文档《mysql双机热备》。

分别是：

- mysqltest1: 10.18.139.81:3307
- mysqltest2: 10.18.139.81:3308

为了后续方便查看具体的sql执行在哪个数据库，可以在配置文件里打开数据库的general_log，加上如下配置

```
general_log = 1
```

```
[mysqld]
server-id          = 1
log_bin            = /var/lib/mysql/mysql-bin.log
general_log        = 1
```

三：mycat的部署和配置

使用docker安装mycat。

先拉取镜像：

```
docker pull abulo/docker-mycat
```

然后创建容器，命令如下：

```
docker run -d --name mycat -v
/data/mycat/server.xml:/usr/local/mycat/conf/server.xml -v
/data/mycat/rule.xml:/usr/local/mycat/conf/rule.xml -v
/data/mycat/schema.xml:/usr/local/mycat/conf/schema.xml -p 8066:8066 -p 9066:9066
abulo/docker-mycat
```

这里映射了三份配置文件和两个端口。数据库的自动切换都是在这些配置文件当中配置的。接下来介绍一下。

1: server.xml

server.xml主要保存了mycat需要的系统配置信息，具体配置内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mycat:server SYSTEM "server.dtd">
<mycat:server xmlns:mycat="http://io.mycat/">
    <system>
        <property name="useSqlStat">1</property>
        <property name="useGlobleTableCheck">0</property>
        <property name="defaultSqlParser">druidparser</property>
        <property name="sequenceHandlerType">2</property>
        <property name="processorBufferPoolType">0</property>

        <property name="serverPort">8066</property>
        <property name="managerPort">9066</property>
        <property name="handleDistributedTransactions">0</property>
        <property name="useOffHeapForMerge">1</property>
        <property name="memoryPageSize">1m</property>
        <property name="spillsFileBufferSize">1k</property>
        <property name="useStreamOutput">0</property>
        <property name="systemReserveMemorySize">389m</property>
    </system>
    <user name="root">
        <property name="password">123456</property>
        <property name="schemas">testdb</property>
    </user>
</mycat:server>
```

- system: 这个标签下的配置都是mycat的系统全局配置。
 - serverPort: 定义 mycat 的使用端口,默认值为 8066。也就是程序连接mycat时的端口。
 - managerPort: 定义 mycat 的管理端口,默认值为 9066。
 - 这两个端口也是我们docker容器映射到宿主机的端口。system的其他标签含义参考文档开头给出的文档。
- user: 这个标签用于定义登录mycat的用户和权限。
 - 上述配置可阐述为，定义了一个名为 root 的用户，密码为 123456，可访问的schema为testdb。testdb是在schema.xml中定义的一个逻辑数据库。

2: schema.xml

Schema.xml 作为 MyCat 中重要的配置文件之一,管理着 MyCat 的逻辑库、表、分片规则、DataNode 以及 DataSource，具体配置内容如下：

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">
    <!-- 数据库配置，与server.xml中的数据库对应 -->
    <schema name="testdb" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1" />
    <!-- 分片配置 -->
    <dataNode name="dn1" dataHost="shard1" database="test" />
    <dataHost name="shard1" maxCon="1000" minCon="10" balance="0" writeType="0"
dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
        <heartbeat>select user();</heartbeat>
```

```

        <writeHost host="db1-M1" url="10.18.139.81:3307" user="root"
password="123456">
        </writeHost>
        <!-- 配置standby writeHost -->
        <writeHost host="db2-M1" url="10.18.139.81:3308" user="root"
password="123456">
        </writeHost>
    </dataHost>
</mycat:schema>

```

- schema: schema 标签用于定义 MyCat 实例中的逻辑库。
 - name: 此处我们定义了一个名为testdb的逻辑库,也就是我们在server.xml中配置的root用户可访问的 schema。
 - dataNode: 指定schema对应的dataNode
- dataNode: dataNode 标签定义了 MyCat 中的数据节点,也就是我们通常所说的数据分片。一个 dataNode 标签就是一个独立的数据分片。
 - name: 定义数据节点的名字,这个名字需要是唯一的。对应schema节点中配置的数据Node属性。
 - dataHost: 该属性用于定义该分片属于哪个数据库实例的,属性值是引用 dataHost 标签上定义的 name 属性。
 - database: 该属性用于定义该分片属于哪个具体数据库实例上的具体库。这个对应我们数据库中的具体数据库名称。我在mysqltest1和mysqltest2中创建了一个名为test的database。
- dataHost: 作为 Schema.xml 中最后的一个标签,该标签在 mycat 逻辑库中也是作为最底层的标签存在,直接定义了具体的数据库实例、读写分离配置和心跳语句。
 - name: 唯一标识 dataHost 标签,供上层的标签使用。对应dataNode中的dataHost属性
 - balance: 决定了哪些MySQL服务器参与到读操作的负载均衡中。
 1. balance="0", 不开启读写分离机制,所有读操作都发送到当前可用的 writeHost 上。
 2. balance="1",全部的 readHost 与 stand by writeHost 参与 select 语句的负载均衡,简单的说,当双主双从模式(M1->S1,M2->S2,并且 M1 与 M2 互为备),正常情况下,M2,S1,S2 都参与 select 语句的负载均衡。
 3. balance="2",所有读操作都随机的在 writeHost、readhost 上分发。
 4. balance="3",所有读请求随机的分发到 writerHost 对应的 readhost 执行,writerHost 不承担读压力,注意 balance=3 只在 1.4 及其以后版本有,1.3 没有。

此处暂未考虑负载均衡的问题,设为0。

- writeType:
 1. writeType="0", 所有写操作发送到配置的第一个 writeHost,第一个挂了切到还生存的第二个 writeHost,重新启动后已切换后的为准,切换记录在配置文件中:dnindex.properties。
 2. writeType="1",所有写操作都随机的发送到配置的 writeHost,1.5 以后废弃不推荐。
- switchType:
 1. switchType="-1": 表示不自动切换
 2. switchType="1": 默认值, 自动切换
 3. switchType="2": 基于MySQL主从同步的状态决定是否切换

数据库的自动切换主要依赖以上两个参数writeType, switchType。

writeType="0" switchType="1" 这样配置, mycat会自动检测数据库是否可用,不可用后自动切换。

- heartbeat: 配置检测数据库节点是否可用的命令。mycat定期会发送该命令给每个数据库节点用于检测节点是否可用。
- writeHost: 指定后端数据库的相关配置给 mycat,用于实例化后端连接池。此处我们配了两个实例, 分别就是我们的mysqltest1和mysqltest2。

3: rule.xml

rule.xml 里面就定义了我们表进行拆分所涉及到的规则定义。这里我们暂时没用到, 可自行查询文档了解其用法。

给一个空的配置文档

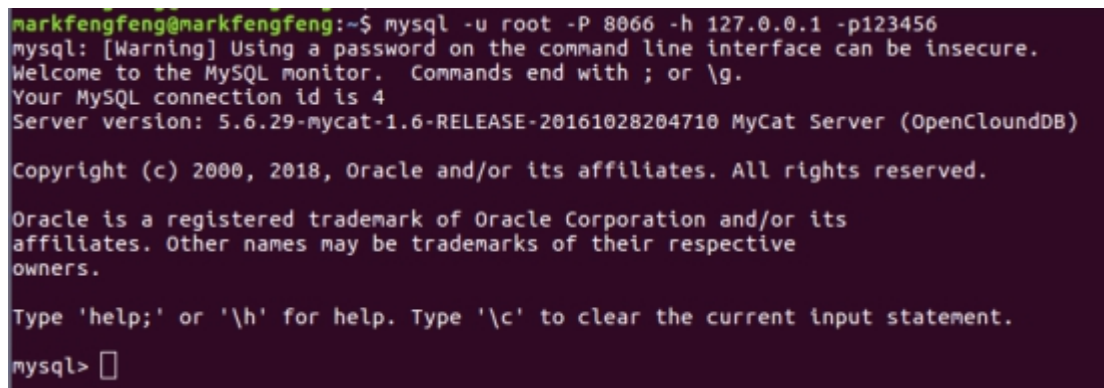
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mycat:rule SYSTEM "rule.dtd">
<mycat:rule xmlns:mycat="http://io.mycat/">
</mycat:rule>
```

四: 测试

以上介绍了mycat的部署和配置。这样就完成了一个简单的数据库自动切换配置。接下来测试一下。

启动mycat, mysqltest1, mysqltest2。

根据在server.xml中的配置mycat的用户名root, 密码123456, 端口为8066, 通过这些信息可以使用mysql客户端连接mycat。



```
markfengfeng@markfengfeng:~$ mysql -u root -P 8066 -h 127.0.0.1 -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloudDB)

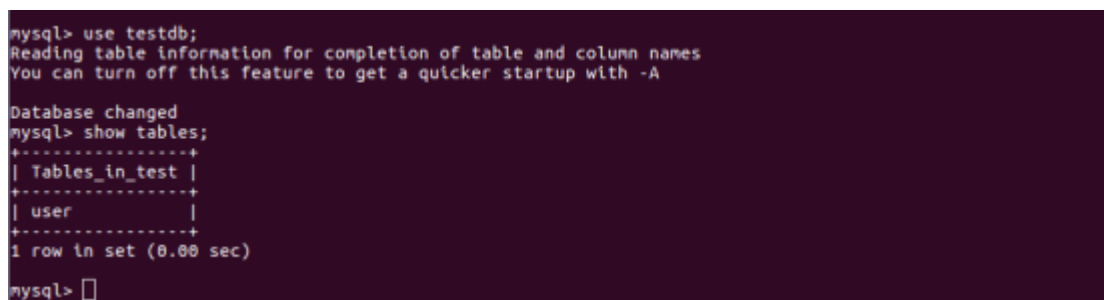
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

我们配置的逻辑数据库为testdb。如下图, 进入数据库我们可以看见在mysqltest1和mysqltest2中定义的数据表user。



```
mysql> use testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| user            |
+-----+
1 row in set (0.00 sec)

mysql>
```

打开mysqltest1和mysqltest2的日志。如下图, 左侧为mycat, 右上角为mysqltest1, 右下角为mysqltest2。可以看见一直有执行select user();这个就是我们在schema.xml中配置的heartbeat。mycat正在检测数据库是否可用。

```
markfengfeng@markfengfeng:~$ mysql -u root -P 8066 -h 127.0.0.1 -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloudDB)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| user            |
+-----+
1 row in set (0.00 sec)

mysql>
```

执行sql。如下图，可以看见所有的sql都在mysqltest1上执行，因为配置了两个数据库的互备，所以mysqltest2上在同步数据。

```
markfengfeng@markfengfeng:~$ mysql -u root -P 8066 -h 127.0.0.1 -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloudDB)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| user            |
+-----+
1 row in set (0.00 sec)

mysql> delete from user;
Query OK, 1 row affected (0.02 sec)

mysql> insert into user values('xiehuafeng');
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+-----+
| name |
+-----+
| xiehuafeng |
+-----+
1 row in set (0.03 sec)

mysql>
```

停止mysqltest1，模拟宕机。再执行插入一条数据。Mysqttest1停止了工作，mycat自动切换到了mysqltest2。通过日志可以看到sql在mysqltest2上执行。


```
markfengfeng@markfengfeng:~$ mysql -u root -P 8066 -h 127.0.0.1 -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.29-mycat-1.6-RELEASE-2016102804710 MyCat Server (OpenCloudDB)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb;
Reading table information for completion of table and column names
You can turn off this feature with 'set -A'

Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| user            |
+-----+
1 row in set (0.00 sec)

mysql> delete from user;
Query OK, 1 row affected (0.02 sec)

mysql> insert into user values("xiehuafeng");
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+-----+
| name      |
+-----+
| xiehuafeng |
+-----+
1 row in set (0.03 sec)

mysql> insert into user values("张三");
Query OK, 1 row affected (0.02 sec)

mysql>
```

```
markfengfeng@markfengfeng:/data/test1/mysql$
2018-12-12T09:36:37.952474Z 22 Query select user()
2018-12-12T09:36:47.952506Z 26 Query select user()
2018-12-12T09:36:57.952391Z 17 Query select user()
2018-12-12T09:37:07.952447Z 24 Query select user()
2018-12-12T09:37:17.952507Z 26 Query select user()
2018-12-12T09:37:27.952368Z 25 Query select user()
2018-12-12T09:37:37.952397Z 21 Query select user()
2018-12-12T09:37:47.952111Z 19 Query select user()
2018-12-12T09:37:57.952215Z 23 Query select user()
2018-12-12T09:38:07.952198Z 18 Query select user()
2018-12-12T09:38:17.952438Z 22 Query select user()
2018-12-12T09:38:27.952528Z 26 Query select user()
2018-12-12T09:38:37.952325Z 17 Query select user()
2018-12-12T09:38:47.952532Z 24 Query select user()
2018-12-12T09:38:57.952480Z 20 Query select user()
2018-12-12T09:39:07.952405Z 25 Query select user()
2018-12-12T09:39:17.952520Z 21 Query select user()
2018-12-12T09:39:27.952252Z 19 Query select user()
2018-12-12T09:39:37.952416Z 23 Query select user()
2018-12-12T09:39:47.952495Z 18 Query select user()
2018-12-12T09:39:57.952191Z 22 Query select user()
2018-12-12T09:40:07.952381Z 26 Query select user()
2018-12-12T09:40:17.952504Z 17 Query select user()
2018-12-12T09:40:27.952387Z 24 Query select user()
2018-12-12T09:40:37.952378Z 20 Query select user()
2018-12-12T09:40:47.952464Z 25 Query select user()

markfengfeng@markfengfeng:/data/test2/mysql$
2018-12-12T09:36:37.952497Z 18 Query select user()
2018-12-12T09:36:47.952668Z 18 Query select user()
2018-12-12T09:36:57.952401Z 18 Query select user()
2018-12-12T09:37:07.952558Z 18 Query select user()
2018-12-12T09:37:17.952629Z 18 Query select user()
2018-12-12T09:37:27.952411Z 18 Query select user()
2018-12-12T09:37:37.952521Z 18 Query select user()
2018-12-12T09:37:47.952149Z 18 Query select user()
2018-12-12T09:37:57.952286Z 18 Query select user()
2018-12-12T09:38:07.952198Z 18 Query select user()
2018-12-12T09:38:17.952446Z 18 Query select user()
2018-12-12T09:38:27.952588Z 18 Query select user()
2018-12-12T09:38:37.952357Z 18 Query select user()
2018-12-12T09:38:47.952545Z 18 Query select user()
2018-12-12T09:38:57.952486Z 18 Query select user()
2018-12-12T09:39:07.952491Z 18 Query select user()
2018-12-12T09:39:17.952580Z 18 Query select user()
2018-12-12T09:39:27.952252Z 18 Query select user()
2018-12-12T09:39:37.952493Z 18 Query select user()
2018-12-12T09:39:47.952530Z 18 Query select user()
2018-12-12T09:39:57.952197Z 18 Query select user()
2018-12-12T09:40:07.952447Z 18 Query select user()
2018-12-12T09:40:17.952635Z 18 Query select user()
2018-12-12T09:40:27.952492Z 18 Query select user()
2018-12-12T09:40:37.952499Z 18 Query select user()
2018-12-12T09:40:47.952486Z 18 Query select user()
2018-12-12T09:40:57.952318Z 18 Query select user()
2018-12-12T09:41:07.952786Z 18 Query select user()
2018-12-12T09:41:09.742901Z 18 Query SET names utf8;
2018-12-12T09:41:09.743009Z 18 Query insert into user values("张三")
2018-12-12T09:41:17.952717Z 18 Query select user()
```

启动mysqltest1，会自动去mysqltest2上同步数据。

```
markfengfeng@markfengfeng:/data/test1/mysql$
2018-12-12T09:38:37.952325Z 17 Query select user()
2018-12-12T09:38:47.952532Z 24 Query select user()
2018-12-12T09:38:57.952486Z 20 Query select user()
2018-12-12T09:39:07.952405Z 25 Query select user()
2018-12-12T09:39:17.952520Z 21 Query select user()
2018-12-12T09:39:27.952252Z 19 Query select user()
2018-12-12T09:39:37.952416Z 23 Query select user()
2018-12-12T09:39:47.952495Z 18 Query select user()
2018-12-12T09:39:57.952191Z 22 Query select user()
2018-12-12T09:40:07.952381Z 26 Query select user()
2018-12-12T09:40:17.952504Z 17 Query select user()
2018-12-12T09:40:27.952387Z 24 Query select user()
2018-12-12T09:40:37.952378Z 20 Query select user()
2018-12-12T09:40:47.952464Z 25 Query select user()
mysqld, Version: 5.7.19-log (MySQL Community Server (GPL)). started with:
Tcp port: 0 Unix socket: (null)
Time Id Command Argument
2018-12-12T09:47:15.832573Z 1 Connect Out rep@10.18.139.81:3308
2018-12-12T09:47:15.903630Z 2 Query BEGIN
2018-12-12T09:47:15.911436Z 2 Query COMMIT /* implicit, from Xid_log_event */
2018-12-12T09:47:15.934107Z 4 Query SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE CREATE_OPTIONS LIKE '%partitioned%';
2018-12-12T09:47:23.439819Z 5 Connect root@172.17.0.1 on test using TCP/IP
2018-12-12T09:47:23.440577Z 5 Query select user()
2018-12-12T09:47:27.952498Z 5 Query select user()
2018-12-12T09:47:37.952393Z 5 Query select user()
```

就这样，通过数据库之间的数据同步以及mycat自动切换可用数据库节点。就做到了数据库的容灾切换。