# 1.准备

1.一台本机 安装mysql

2.一台虚拟机 安装mysql 需要两台机器互相可以ping通

# 2.主数据库配置

1.打开my.ini (linux下为 /etc/my.cnf)

2.在mysqld下增加如下配置

log-bin=mysql-bin #slave会基于此log-bin来做replication

server-id=1 #master的标示

binlog-do-db = ms\_test #用于master-slave的具体数据库

3.进入mysql

增加一个从数据库连接主数据库的用户名和密码

repl为用户名 10.20.147.111 为 从数据库IP

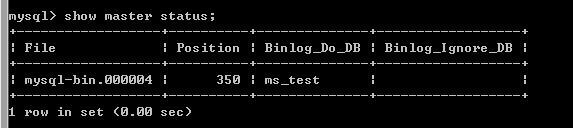
111111为密码

GRANT REPLICATION SLAVE ON \*.\* TO repl@10.20.147.111 IDENTIFIED BY '111111';

4.重启mysql

5.进行mysql 输入 show master status;

查看主数据库状态



# 3.配置从数据库

1.编辑 mysql配置文件 /etc/my.cnf

2.在文件最后 增加一行

server-id=2 #slave 从数据库标识

保存退出 并且重启mysql

3.进入数据库

输入

10.20.147.110 为主数据库IP

repl 连接主数据库的用户名

111111 连接主数据库的密码

mysql-bin.000003 是 show master status 中file字段的值

161261 是show master status 中position字段的值

CHANGE MASTER TO

MASTER\_HOST='10.20.147.110',

MASTER\_USER='repl',

MASTER\_PASSWORD='111111',

MASTER\_LOG\_FILE='mysql-bin.000003',

MASTER\_LOG\_POS=161261;

4.启动从数据库

start slave;

5.查看从数据库状态

show slave status \G;

# 4.安装amoeba 代理数据库

1.下载压缩包

2.解压到本地

3.进入conf文件夹，编辑amoeba.xml配置文件

4.按照注释进行配置

<?xml version="1.0" encoding="gbk"?>

<!DOCTYPE amoeba:configuration SYSTEM "amoeba.dtd">

<amoeba:configuration xmlns:amoeba="http://amoeba.meidusa.com/">

<server>

<!-- proxy server绑定的端口 -->

<property name="port">8066</property>

<!-- proxy server绑定的IP -->

<!--

<property name="ipAddress">127.0.0.1</property>

-->

<!-- proxy server net IO Read thread size -->

<property name="readThreadPoolSize">20</property>

<!-- proxy server client process thread size -->

<property name="clientSideThreadPoolSize">30</property>

<!-- mysql server data packet process thread size -->

<property name="serverSideThreadPoolSize">30</property>

<!-- socket Send and receive BufferSize(unit:K) -->

<property name="netBufferSize">128</property>

<!-- Enable/disable TCP\_NODELAY (disable/enable Nagle's algorithm). -->

<property name="tcpNoDelay">true</property>

<!-- 对外验证的用户名 -->

<property name="user">root</property>

<!-- 对外验证的密码 -->

<property name="password">root</property>

</server>

<!--

每个ConnectionManager都将作为一个线程启动。

manager负责Connection IO读写/死亡检测

-->

<connectionManagerList>

<connectionManager name="defaultManager" class="com.meidusa.amoeba.net.MultiConnectionManagerWrapper">

<property name="subManagerClassName">com.meidusa.amoeba.net.AuthingableConnectionManager</property>

<!--

default value is avaliable Processors

<property name="processors">5</property>

-->

</connectionManager>

</connectionManagerList>

<dbServerList>

<!--

一台mysqlServer 需要配置一个pool，

如果多台 平等的mysql需要进行loadBalance，

平台已经提供一个具有负载均衡能力的objectPool：com.meidusa.amoeba.mysql.server.MultipleServerPool

简单的配置是属性加上 virtual="true",该Pool 不允许配置factoryConfig

或者自己写一个ObjectPool。

-->

<!--主数据库配置-->

<dbServer name="server1">

<!-- PoolableObjectFactory实现类 -->

<factoryConfig class="com.meidusa.amoeba.mysql.net.MysqlServerConnectionFactory">

<property name="manager">defaultManager</property>

<!-- 真实mysql数据库端口 -->

<!--主数据库端口 -->

<property name="port">3306</property>

<!-- 真实mysql数据库IP -->

<!--主数据库IP -->

<property name="ipAddress">127.0.0.1</property>

<!--需要做主从的数据库名字-->

<property name="schema">ms\_test</property>

<!-- 用于登陆mysql的用户名 -->

<!--主数据库用户名 -->

<property name="user">root</property>

<!-- 用于登陆mysql的密码 -->

<!--主数据库密码 -->

<property name="password">root</property>

</factoryConfig>

<!-- ObjectPool实现类 -->

<poolConfig class="com.meidusa.amoeba.net.poolable.PoolableObjectPool">

<property name="maxActive">200</property>

<property name="maxIdle">200</property>

<property name="minIdle">10</property>

<property name="minEvictableIdleTimeMillis">600000</property>

<property name="timeBetweenEvictionRunsMillis">600000</property>

<property name="testOnBorrow">true</property>

<property name="testWhileIdle">true</property>

</poolConfig>

</dbServer>

<!--从数据库配置-->

<dbServer name="server2">

<!-- PoolableObjectFactory实现类 -->

<factoryConfig class="com.meidusa.amoeba.mysql.net.MysqlServerConnectionFactory">

<property name="manager">defaultManager</property>

<!-- 真实mysql数据库端口 -->

<!--从数据库端口-->

<property name="port">3306</property>

<!-- 真实mysql数据库IP -->

<!--从数据库IP-->

<property name="ipAddress">192.168.11.151</property>

<property name="schema">ms\_test</property>

<!-- 用于登陆mysql的用户名 -->

<!--从数据库用户名-->

<property name="user">root</property>

<!-- 用于登陆mysql的密码 -->

<!--从数据库密码-->

<property name="password">MyNewPass1!</property>

</factoryConfig>

<!-- ObjectPool实现类 -->

<poolConfig class="com.meidusa.amoeba.net.poolable.PoolableObjectPool">

<property name="maxActive">200</property>

<property name="maxIdle">200</property>

<property name="minIdle">10</property>

<property name="minEvictableIdleTimeMillis">600000</property>

<property name="timeBetweenEvictionRunsMillis">600000</property>

<property name="testOnBorrow">true</property>

<property name="testWhileIdle">true</property>

</poolConfig>

</dbServer>

<dbServer name="multiPool" virtual="true">

<poolConfig class="com.meidusa.amoeba.server.MultipleServerPool">

<!-- 负载均衡参数 1=ROUNDROBIN , 2=WEIGHTBASED , 3=HA-->

<property name="loadbalance">1</property>

<!-- 参与该pool负载均衡的poolName列表以逗号分割 -->

<property name="poolNames">server1</property>

</poolConfig>

</dbServer>

</dbServerList>

<queryRouter class="com.meidusa.amoeba.mysql.parser.MysqlQueryRouter">

<!--

<property name="ruleConfig">${amoeba.home}/conf/rule.xml</property>

<property name="functionConfig">${amoeba.home}/conf/functionMap.xml</property>

<property name="ruleFunctionConfig">${amoeba.home}/conf/ruleFunctionMap.xml</property>

<property name="LRUMapSize">1500</property>

<property name="defaultPool">server1</property>

<property name="writePool">server1</property>

<property name="readPool">server1</property>

<property name="needParse">true</property>

-->

<property name="LRUMapSize">1500</property>

<property name="defaultPool">server1</property>

<property name="writePool">server1</property>

<property name="readPool">server2</property>

<property name="needParse">true</property>

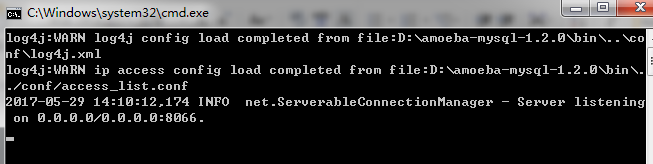
</queryRouter>

</amoeba:configuration>

5.进入amoeba bin文件夹

双击amoeba.bat启动

显示



证明启动成功

# 6.测试

1.在主数据库建立一个表，从数据库无需建立表

2.编写一段JDBC代码测试

3. **package** msTest.test;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

/\*\*

\* 数据库工具类

\* **@author** muty

\*

\*/

**public** **class** DBUtil {

//amoeba数据库用户名

**private** **static** **final** String ***username*** = "root";

//amoeba数据库密码

**private** **static** **final** String ***password*** = "root";

//连接到哪一个数据库

**private** **static** **final** String ***db\_name*** = "ms\_test";

/\*\*

\* Connection 数据库连接对象

\* java.sql.Connection

\* 获取数据库连接

\*/

**public** **static** Connection getConnection(){

Connection conn = **null**;

**try**{

//加载数据库驱动

Class.*forName*("com.mysql.jdbc.Driver");

//拼接连接数据的地址和参数

String url = "jdbc:mysql://localhost:8066/"+***db\_name***

+"?user="+***username***+"&password="+***password***+"&characterEncoding=utf8";

System.***out***.println(url);

//获取连接

conn = DriverManager.*getConnection*(url);

}**catch**(Exception e){

e.printStackTrace();

}

**return** conn;

}

**public** **static** **void** main(String[] args){

**try**{

Connection conn = DBUtil.*getConnection*();

PreparedStatement ps = conn.prepareStatement("insert into stu(name,age) values(?,?)");

ps.setString(1,"测试学生");

ps.setInt(2, 25);

**int** r = ps.executeUpdate();

System.***out***.println(r);

}**catch**(Exception e){

e.printStackTrace();

}

}

}

4.执行代码，查看主数据库和从数据库，数据相同则证明配置成功