Yii2.0+ubuntu下mysql读写分离

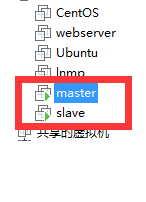
今天是2016年上班第一天，调整调整状态，总不能闲着，于是乎吧很久之前就像实施的事情就在今天把它做了！

数据库读写分离是大型互联网架构的常用技术，主服务器负责写操作，从服务器负责读操作，一般读写分离架构分为：

1. 单主单从（单台主服务器对单台从服务器）
2. 单主多从
3. 多主多从

其原理是通过mysql运行产生的二进制日志，任意相通版本的数据库拿到二级制日志并运行即可得到数据库结构和数据。（这里我们主要介绍怎么操作，原理部分请各位同学再去查询！）

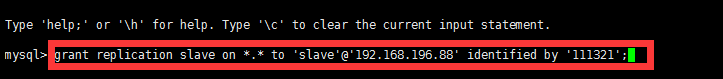
我们新建两台虚拟机，（juzi习惯用ubuntu，你也可以用其它linux发行版，原理一样）一台master放主数据库，一台slave放从数据库。



一、首先我们进入master数据库：



输入密码就进入了，接下来我们创建一个slave用户并给它授权，让它可以远程访问



（你可以将 ‘slave’@’192.168.196.88’换成’slave’@’%’）

最后，我们进行数据库的配置：

1. vim（vi） /etc/mysql/my.conf
2. server-id = 1 #主机标示，整数  
   log\_bin = mysql-bin #确保此文件可写  
   read-only =0 #主机，读写都可以  
   binlog-do-db =test #需要备份数据，多个写多行  
   binlog-ignore-db=mysql #不需要备份的数据库，多个写多行
3. 保存配置并重启。
4. 查看master是否建立完成： show master status;  
   +------------------+----------+--------------+------------------+  
   | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |  
   +------------------+----------+--------------+------------------+  
   | mysql-bin.000002 | 218 | test | mysql |  
   +------------------+----------+--------------+------------------+  
   1 row in set (0.00 sec)  
   Master基本完成；

二、接下来我们进入从服务器：

数据库配置：

1. server-id = 2#标示，整数（一定要与mater不同）  
   log\_bin = mysql-bin #确保此文件可写  
   replicate-do-db=test #需要备份数据，多个写多行  
   replicate-ignore\_db=mysql #不需要备份的数据库，多个写多行

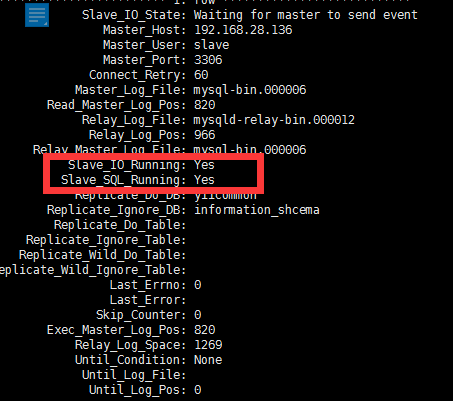
保存并重启mysql。

进入到slave数据库然后设置从master复制日志配置：

change master to master\_host='192.168.196.68', master\_user='slave', master\_password='111321',master\_port=3306;

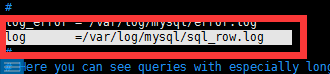
启动Slave:start slave;

查看状态：show slave status\G;



Slave\_IO\_Running: Yes  
Slave\_SQL\_Running: Yes 如果这两处都为yes 则配置没有问题。

接下来你可以配置my.conf 来检测sql语句的执行



在mysqld节点下进行配置！

三、Yii2.0的主从配置：

'components' => [

'db' => [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=192.168.28.136;dbname=yiicommon',

'username' => 'root',

'password' => 'sadamu',

'charset' => 'utf8',

'tablePrefix' => 'le\_',

// 配置从服务器

'slaveConfig' => [

'username' => 'root',

'password' => 'sadamu',

'charset' => 'utf8',

'tablePrefix' => 'le\_',

'attributes' => [

// use a smaller connection timeout

PDO::ATTR\_TIMEOUT => 10,

],

],

// 配置从服务器组

'slaves' => [

['dsn' => 'mysql:host=192.168.28.137;dbname=yiicommon'],

// ['dsn' => 'dsn for slave server 2'],

// ['dsn' => 'dsn for slave server 3'],

// ['dsn' => 'dsn for slave server 4'],

],

],

然后执行程序，你就可以在配置文件中所定义的文件中看到执行的sql语句了！