

Mysql主从同步（复制）

目录：

mysql主从同步定义

主从同步机制

配置主从同步

配置主服务器

配置从服务器

使用主从同步来备份

使用mysqldump来备份

备份原始文件

主从同步的小技巧

排错

Slave_IO_Running: NO

Slave_SQL_Running: No

mysql主从同步定义

主从同步使得数据可以从一个数据库服务器复制到其他服务器上，在复制数据时，一个服务器充当主服务器（master），其余的服务器充当从服务器（slave）。因为复制是异步进行的，所以从服务器不需要一直连接着主服务器，从服务器甚至可以通过拨号断断续续地连接主服务器。通过配置文件，可以指定复制所有的数据库，某个数据库，甚至是某个数据库上的某个表。

使用主从同步的好处：

1. 通过增加从服务器来提高数据库的性能，在主服务器上执行写入和更新，在从服务器上向外提供读功能，可以动态地调整从服务器的数量，从而调整整个数据库的性能。

2. 提高数据安全-因为数据已复制到从服务器，从服务器可以终止复制进程，所以，可以在从服务器上备份而不破坏主服务器相应数据

3. 在主服务器上生成实时数据，而在从服务器上分析这些数据，从而提高主服务器的性能

注意，mysql是异步复制的，而MySQL Cluster是同步复制的。有很多种主从同步的方法，但核心的方法有两种，Statement Based Replication（SBR）基于SQL语句的复制，另一种是Row Based Replication（RBR）基于行的复制，也可以使用Mixed Based Replication（MBR）。在mysql5.6中，默认使用的是SBR。而mysql 5.6.5和往后的版本是基于global transaction identifiers(GTIDs)来进行事务复制。当使用GTIDs时可以大大简化复制过程，因为GTIDs完全基于事务，只要在主服务器上提交了事务，那么从服务器就一定会执行该事务。

通过设置服务器的系统变量binlog_format来指定要使用的格式：

1.SBR：当使用二进制日志时，主服务器会把SQL语句写入到日志中，然后从服务器会执行该日志，这就是SBR，在mysql5.1.4之前的版本都只能使用这种格式。使用SBR会有如下

长处：

1. 日志文件更小

2. 记录了所有的语句，可以用来日后审计

公告

昵称：kylinlin

园龄：3年1个月

粉丝：78

关注：2

+加关注

<2018年7, >

日	一	二	三
24	25	26	27
1	2	3	4
8	9	10	11
15	16	17	18
22	23	24	25
29	30	31	1

搜索

我的标签

- scrapy(8)
- Python(7)
- 爬虫(7)
- scrapy-redis(1)
- 分布式爬虫(1)
- collectd(1)
- grafana(1)
- influxdb(1)

弊端：

- 1. 使用如下函数的语句不能被正确地复制：load_file(); uuid(), uuid_short(); user(); found_rows(); sysdate(); get_lock(); is_free_lock(); is_used_lock(); master_pos_wait(); rand(); release_lock(); sleep(); version();
- 2. 在日志中出现如下警告信息的不能正确地复制：[Warning] Statement is not safe to log in statement format.
- 3. 或者在客户端中出现show warnings
- 4. Insert ... select语句会执行大量的行级锁表
- 5. Update语句会执行大量的行级锁表来扫描整个表

2.RBR：主服务器把表的行变化作为事件写入到二进制日志中，主服务器把代表了行变化的事件复制到从服务中，使用RBR的

长处：

- 1. 所有的数据变化都是被复制，这是最安全的复制方式
- 2. 更少的行级锁表

弊端：

- 1. 日志会很大
- 2. 不能通过查看日志来审计执行过的sql语句，不过可以通过使用mysqlbinlog
- 3. --base64-output=decode-rows --verbose来查看数据的 变动

3.MBR：既使用SBR也使用RBR，默认使用SBR

主从同步机制

Mysql服务器之间的主从同步是基于二进制日志机制，主服务器使用二进制日志来记录数据库的变动情况，从服务器通过读取和执行该日志文件来保持和主服务器的数据一致。

在使用二进制日志时，主服务器的所有操作都会被记录下来，然后从服务器会接收到该日志的一个副本。从服务器可以指定执行该日志中的哪一类事件（譬如只插入数据或者只更新数据），默认会执行日志中的所有语句。

每一个从服务器会记录关于二进制日志的信息：文件名和已经处理过的语句，这意味着不同的从服务器可以分别执行同一个二进制日志的不同部分，并且从服务器可以随时连接或者中断和服务器的连接。

主服务器和每一个从服务器都必须配置一个唯一的ID号（在my.cnf文件的[mysqld]模块下有一个server-id配置项），另外，每一个从服务器还需要通过CHANGE MASTER TO语句来配置它要连接的主服务器的ip地址，日志文件名称和该日志里面的位置（这些信息存储在主服务器的数据库里）

配置主从同步

有很多种配置主从同步的方法，可以总结为如下的步骤：

- 1. 在主服务器上，必须开启二进制日志机制和配置一个独立的ID
- 2. 在每一个从服务器上，配置一个唯一的ID，创建一个用来专门复制主服务器数据的账号
- 3. 在开始复制进程前，在主服务器上记录二进制文件的位置信息
- 4. 如果在开始复制之前，数据库中已经有数据，就必须先创建一个数据快照（可以使用mysqldump导出数据库，或者直接复制数据文件）
- 5. 配置从服务器要连接的主服务器的IP地址和登陆授权，二进制日志文件名和位置

配置主服务器

1.更改配置文件，首先检查你的主服务器上的my.cnf文件中是否已经在[mysqld]模块下配置了log-bin和server-id

```
[mysqld]

log-bin=mysql-bin

server-id=1
```

注意上面的log-bin和server-id的值都是可以改为其他值的，如果没有上面的配置，首先关闭mysql服务器，然后添加上去，接着重启服务器

2.创建用户，每一个从服务器都需要用到一个账户名和密码来连接主服务器，可以为每一个从服务器都创建一个账户，也可以让全部服务器使用同一个账户。下面就为同一个ip网段的所有从服务器创建一个只能进行主从同步的账户。

首先登陆mysql，然后创建一个用户名为rep，密码为123456的账户，该账户可以被192.168.253网段下的所有ip地址使用，且该账户只能进行主从同步

```
mysql > grant replication slave on *.* to 'rep'@'192.168.253.%' identified by '123456';
```

随笔分类(56)

Django(1)

Linux(5)

Matplotlib(1)

mysql(2)

nagios(1)

Pandas(8)

R(2)

scrapy(9)

Seaborn(1)

sqlite(1)

爬虫(3)

数据分析(15)

网络安全(2)

运维(5)

随笔档案(40)

2017年1月 (1)

2016年4月 (7)

2016年3月 (18)

2016年2月 (6)

2016年1月 (1)

2015年11月 (1)

2015年10月 (1)

2015年9月 (1)

2015年8月 (1)

2015年7月 (1)

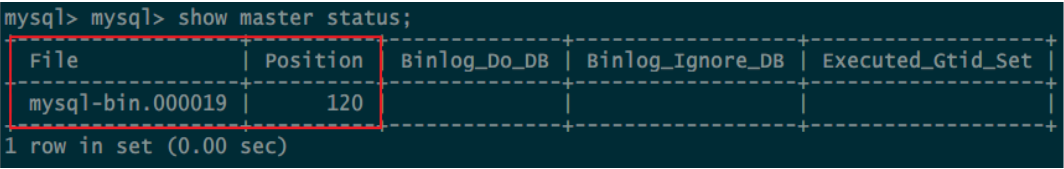
3.获取二进制日志的信息并导出数据库，步骤：

首先登陆数据库，然后刷新所有的表，同时给数据库加上一把锁，阻止对数据库进行任何的写操作

```
mysql > flush tables with read lock;
```

然后执行下面的语句获取二进制日志的信息

```
mysql > show master status;
```



File的值是当前使用的二进制日志的文件名，Position是该日志里面的位置信息（不需要纠结这个究竟代表什么），记住这两个值，会在下面配置从服务器时用到。

注意：如果之前的服务器并没有配置使用二进制日志，那么使用上面的sql语句会显示空，在锁表之后，再导出数据库里的数据（如果数据库里没有数据，可以忽略这一步）

```
[root@localhost backup]# mysqldump -uroot -p'123456' -S /data/3306/data/mysql.sock --all-databases > /server/backup/mysql_bak.%(date +%F).sql
```

如果数据量很大，可以在导出时就压缩为原来的大概三分之一

```
[root@localhost backup]# mysqldump -uroot -p'123456' -S /data/3306/data/mysql.sock --all-databases | gzip > /server/backup/mysql_bak.%(date +%F).sql.gz
```

这时可以对数据库解锁，恢复对主数据库的操作

```
mysql > unlock tables;
```

配置从服务器

首先检查从服务器上的my.cnf文件中是否已经在[mysqld]模块下配置leserver-id

```
[mysqld]

server-id=2
```

注意上面的server-id的值都是可以改为其他值的（建议更改为ip地址的最后一个字段），如果没有上面的配置，首先关闭mysql服务器，然后添加上去，接着重启服务器

如果有多个从服务器上，那么每个服务器上配置的server-id都必须不一致。从服务器上没必要配置log-bin，当然也可以配置log-bin选项，因为可以在从服务器上进行数据备份和灾难恢复，或者某一天让这个从服务器变成一个主服务器

如果主服务器导出了数据，下面就导入该文件，如果主服务器没有数据，就忽略这一步

```
[root@localhost ~]# mysql -uroot -p'123456' -S /data/3306/data/mysql.sock < /server/backup/mysql_bak.2015-07-01.sql
```

如果从主服务器上拿过来的是压缩文件，就先解压再导入

配置同步参数，登陆mysql，输入如下信息：

```
mysql> CHANGE MASTER TO

-> MASTER_HOST='master_host_name',

-> MASTER_USER='replication_user_name',

-> MASTER_PASSWORD='replication_password',

-> MASTER_LOG_FILE='recorded_log_file_name',
```

如图所示：

文章分类

Linux

架构

网络安全

运维

最新评论

1. Re:Pandas简易入门

博主您好，我用data.co
略号，还是不能显示全部
总共有106列，请问您知

2. Re:scrapy-redis使用

还请楼主给出部分源码供
果您方便的话，麻烦联系
74866364,或者发博客I
谢谢您啦

3. Re:scrapy-redis使用

博主你好，文章有两点疑
master端的爬虫产生的I
edis中，这个过程这么身
获取到URL爬取到的内容
master端的Redis的呢？

4. Re:用pycharm开发C

good

5. Re:用pycharm开发C

good

阅读排行榜

1. 数据可视化（三） - S
(50206)

```
mysql> change master to
-> master_host='192.168.253.143',
-> master_port=3306,
-> master_user='rep',
-> master_password='123456',
-> master_log_file='mysql-bin.000019',
-> master_log_pos=120;
Query OK, 0 rows affected, 2 warnings (0.09 sec)
```

启动主从同步进程

```
mysql > start slave;
```

检查状态

```
mysql > show slave status \G
```

```
mysql> show slave status \G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.253.143
Master_User: rep
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000019
Read_Master_Log_Pos: 352
Relay_Log_File: mysql-relay-bin.000002
Relay_Log_Pos: 515
Relay_Master_Log_File: mysql-bin.000019
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 352
Relay_Log_Space: 688
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
```

上面的两个进程都显示YES则表示配置成功

使用主从同步来备份

把主服务器的数据复制到从服务器上，然后备份从服务器的数据，在数据量不是很大的时候使用mysqldump命令，对于很大的数据库，就直接备份数据文件。

使用mysqldump来备份

步骤：（以下的所有操作都在从服务器上进行）

1.首先暂停从服务器的复制进程

```
shell > mysqladmin stop-slave
```

或者只是暂停SQL进程（从服务器仍然能接收二进制日志的事件，但不会执行这些事件，这样能在重启SQL进程时加快复制进度）

```
shell > mysql -e 'stop slave sql_thread;'
```

2.使用mysqldump导出全部或部分的数据库

```
shell > mysqldump --all-databases > fulldb.dump
```

3.在导出数据库后，重启复制进程

```
shell > mysqladmin start-slave
```

备份原始文件

为了保证数据文件的完整性，在备份之前首先关闭从服务器，步骤：

1.关闭从服务器：

```
shell > mysqladmin shutdown
```

2. Pandas之容易让人混淆选择(28322)

3. Pandas简易入门（一

4. scrapy-redis使用详

5. Pandas简易入门（四

评论排行榜

1. scrapy-redis使用详

2. 机器学习简易入门（-（3）

3. 用pycharm开发djar

4. 机器学习简易入门（

5. Learning Scrapy笔记根据Excel文件运行多

推荐排行榜

1. 机器学习简易入门（归(4)

2. 机器学习简易入门（-（4）

3. 机器学习简易入门（

4. Pandas简易入门（一

5. Scrapy Learning笔记双向爬取(2)

2.复制数据文件，可以使用压缩命令，假如当前目录就是数据库的数据目录（在my.cnf文件中的配置项datadir的值就是该目录的位置）

```
shell > tar cf /tmp/dbbackup.tar ./data
```

3.然后再启动mysql服务器

主从同步的小技巧

主服务器第一次导入数据，如果你从其他地方拿来了要导入到主服务器中的数据，此时只要在主服务器中导入一次即可，因为这些数据会自动发送到从服务器中，在主服务器上使用命令

```
shell > mysql -h master < other_data.sql
```

增加从服务器，本来已经至少有一个从服务器时（暂时命名为slave1），决定再添加其余的从服务器（slave2），此时就不需要像上面那样去操作主服务器，只要复制一个已经存在的从服务器就可以了

排错

Slave_IO_Running: NO

这是一个很常见的错误（我也曾对这个错误咬牙切齿），总结起来就三个原因：

1. 主服务器的网络不通，或者主服务器的防火墙拒绝了外部连接3306端口
2. 在配置从服务器时，输错了ip地址和密码，或者主服务器在创建用户时写错了用户名和密码
3. 在配置从服务器时，输错了主服务器的二进制日志信息

排错过程：（主服务器ip: 192.168.1.139, 从服务器ip: 192.168.1.204）

第0步就是检查错误日志，如果不能快速排错，可以按我的步骤试试：

1. 首先在从服务器上执行ping程序，确定能ping通主服务器

在从服务器上执行mysql的远程连接

```
[root@slave204 log]# mysql -urep -p -h 192.168.1.139 -P3306
```

如果显示ERROR 1045 (28000): Access denied for user 'test'@'192.168.1.204' (using password: YES)则跳转到第3

2. 登陆主服务器的mysql，查看所有的用户

```
mysql > select user,host from mysql.user;
```

```
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| rep@192.168.1.% | %             |
| ttl@%         | %             |
| tt@192.168.1.204 | %             |
| root          | 127.0.0.1     |
| root          | ::1           |
| root          | firstcentos   |
| root          | firstcentos   |
| root          | localhost     |
| root          | localhost     |
+-----+-----+
9 rows in set (0.00 sec)
```

我想要的就是rep用户名

上图就是我的错误根源，可以看到用户名完全写错了，先删除错误的用户：

```
mysql > drop user "rep@192.168.1.%"@ "%";
```

再重新创建用户：

```
mysql > grant replication slave on *.* to 'rep'@'192.168.1.%' identified by '123456';
```

```
mysql > flush privileges;
```

3. 假如用户名没有错，那么如何排除是否是输入的密码错误呢？

额，我也想知道方法。最好就是多输入几遍，或者重新创建用户名和密码来测试。问题还没有解决，转到4

4. 在你的防火墙中添加3306端口

```
[root@localhost mysql]# firewall-cmd --zone=public --add-port=3306/tcp --permanent
```

```
[root@localhost mysql]# firewall-cmd --reload
```

再关闭selinux

```
[root@slave204 log]# vi /etc/sysconfig/selinux
```

把SELINUX=enforcing改为SELINUX=disabled

```
[root@slave204 log]# source /etc/sysconfig/selinux
```

登录主服务器，查看服务器状态

```
mysql > show master status \G
```

然后重新配置一次从服务器，在配置之前首先关闭主从同步进程

```
mysql > stop slave;
```

之外的方法，我也没试过

Slave_SQL_Running: No

把上面的Slave_IO_Running调试成YES后，就轮到这个小样了。我根据这个博客的内容来解决的：
<http://kerry.blog.51cto.com/172631/277414>

分类： 运维, mysql

好文要顶

关注我

收藏该文



kylinlin
关注 - 2
粉丝 - 78

+加关注

« 上一篇: Mysql备份与恢复

» 下一篇: R简易入门（一）

posted @ 2016-03-09 16:32 kylinlin 阅读(11887) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- Timeline登陆Microsoft Launcher还要再等段时间
 - 外媒分析小米为何首日破发：估值还是有些高
 - 红帽实现业务转型 帮助企业进行数字化转型
 - “共享单车第一镇”凉了：千元单车2折就卖，车厂转型房地产
 - React技能图 2018 目前在GitHub上已获2千多Star
- » 更多新闻...

最新知识库文章:

- 断点单步跟踪是一种低效的调试方法
 - 测试 | 让每一粒尘埃有的放矢
 - 从Excel到微服务
 - 如何提升你的能力？给年轻程序员的几条建议
 - 程序员的那些反模式
- » 更多知识库文章...