



访问: 次

昵称: 那一叶随风
园龄: 2年8个月
粉丝: 119
关注: 15
+加关注

<	2018年4月						>
日	一	二	三	四	五	六	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	1	2	3	4	5	

搜索

找找看

谷歌搜索

我的标签

- php(16)
- mysql(10)
- 网络协议(5)
- Web安全(5)
- Memcache(5)
- Redis(4)
- Linux(3)
- apache(2)
- Web service(2)
- 设计模式(2)
- 更多

随笔分类(61)

- Apache(1)
- Linux(3)
- Memcached(5)
- MySQL(9)
- PHP(22)
- Redis(4)
- Smarty(1)
- Web安全(5)
- YII(1)
- 分布式
- 设计模式(2)
- 算法与数据结构
- 网络协议(5)
- 应用接口(3)

随笔档案(49)

- 2018年2月 (2)
- 2017年12月 (2)
- 2017年10月 (1)
- 2017年9月 (2)
- 2017年7月 (1)

MySQL主从复制与主主复制

阅读目录

- [1、简介](#)
- [2、环境说明](#)
- [3、主从复制](#)
 - [3.1、MySQL](#)
 - [3.2、配置文件](#)
 - [3.3、开始构建主从复制](#)
 - [3.4、测试主从复制](#)
- [4、MySql主主复制](#)
 - [4.1、实现原理](#)
 - [4.2、配置文件](#)
 - [4.3、开始构建主主复制](#)
 - [4.4、测试主主复制](#)
- [5、注意事项](#)

[回到顶部](#)

1、简介

MySQL作为世界上使用最为广泛的数据库之一，免费是其原因之一。但不可忽略的是它本身的功能的确很强大。随着技术的发展，在实际的生产环境中，由单台MySQL数据库服务器不能满足实际的需求。此时数据库集群就很好的解决了这个问题了。采用MySQL分布式集群，能够搭建一个高并发、负载均衡的集群服务器（这篇博客暂时不涉及）。在此之前我们必须要保证每台MySQL服务器里的数据同步。数据同步我们可以通过MySQL内部配置就可以轻松完成，主要有主从复制和主主复制。

[回到顶部](#)

2、环境说明

两台linux虚拟主机

Linux版本CentOS6.6、MySQL 5.5

ip: 192.168.95.11、192.168.95.12

[回到顶部](#)

3、主从复制

[回到顶部](#)

3.1、MySQL

已经安装好，并且没有任何数据



- 2016年12月 (3)
- 2016年3月 (2)
- 2015年12月 (2)

积分与排名

积分 - 67365
排名 - 5324

最新评论

1. Re:MySQL主从复制与主主复制
binlog-ignore 现在变成了binlog-ignore-db了
--Ze_wait
2. Re:OSI七层模型学习笔记
6
--剑仙6
3. Re:MySQL主从复制与主主复制
@那一叶随风再请教一个问题，在没有中间件的情况下直接调用双主做读写，会有数据一致性问题么？...
--showdbs
4. Re:MySQL主从复制与主主复制
@showdbs如果完全全按照我这篇博客里面来配置，肯定是不可能的。这篇博客里介绍的是最简单的主从主复制分布式配置。如果想要宕机自动切换一般需要中间件辅助加以实现，例如Keepalived中间件。我.....
--那一叶随风
5. Re:MySQL主从复制与主主复制
辛苦了
--矛盾gg

阅读排行榜

- 搭建MySQL高可用负载均衡集群(22956)
- MySQL主从复制与主主复制(19612)
- XSS跨站脚本攻击(16821)
- MySQL读写分离技术(8379)
- MySQL查询优化(5378)
- Session攻击（会话劫持+固定）与防御(4354)
- CSRF攻击与防御(4344)
- Apache URL重写规则(4288)
- Smarty的基本使用与总结(3855)
- memcached的安装以及php两个扩展软件安装（memcache、memcached）(3745)

评论排行榜

- MySQL读写分离技术(12)
- 搭建MySQL高可用负载均衡集群(11)
- MySQL主从复制与主主复制(9)
- Redis应用----消息传递(7)
- MySQL查询优化(5)

推荐排行榜

- MySQL读写分离技术(33)
- 搭建MySQL高可用负载均衡集群(21)
- XSS跨站脚本攻击(14)
- MySQL主从复制与主主复制(14)
- CSRF攻击与防御(8)
- memcached分布式缓存(8)
- Redis应用----消息传递(7)
- MySQL查询优化(7)
- Session攻击（会话劫持+固定）与防御(4)
- Redis构建分布式锁(4)

一般Linux中的MySQL配置文件都在/etc/my.cnf(windows中的配置文件为mysql.ini)

log-bin=mysql-bin 开启二进制日志

注意：二进制日志必须开启，因为数据的同步实质上就是其他的MySQL数据库服务器将这个数据变更的二进制日志在本机上再执行一遍。

192.168.95.11 为主数据库服务器

192.168.95.12 为从数据库服务器

[回到顶部](#)

3.3、开始构建主从复制

第一步：

在192.168.95.11中创建一个192.168.95.12主机中可以登录的MySQL用户

用户：mysql12

密码：mysql12

```
mysql>GRANT REPLICATION SLAVE ON *.* TO 'mysql12'@'192.168.95.12' IDENTIFIED BY 'mysql12';
```

mysql>FLUSH PRIVILEGES;

第二步：

查看192.168.95.11MySQL服务器二进制文件名与位置

```
mysql>SHOW MASTER STATUS;
```

```
mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000048 | 432      |               |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

第三步：

告知二进制文件名与位置

在192.168.95.12中执行：

```
mysql>CHANGE MASTER TO
>MASTER_HOST='192.168.95.11',
>MASTER_USER='mysql12',
>MASTER_PASSWORD='mysql12',
>MASTER_LOG_FILE='mysql-bin.000048',
>MASTER_LOG_POS=432;
```

```
mysql> CHANGE MASTER TO
-> MASTER_HOST='192.168.95.11',
-> MASTER_USER='mysql12',
-> MASTER_PASSWORD='mysql12',
-> MASTER_LOG_FILE='mysql-bin.000048',
-> MASTER_LOG_POS=432;
Query OK, 0 rows affected (0.04 sec)
```

完成主从复制配置

[回到顶部](#)

```
mysql>SLAVE START; #启动复制
```

```
mysql>SHOW SLAVE STATUS\G #查看主从复制是否配置成功
```

```
mysql> SHOW SLAVE STATUS\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 192.168.95.11
      Master_User: mysql12
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000048
      Read_Master_Log_Pos: 432
      Relay_Log_File: localhost-relay-bin.000002
      Relay_Log_Pos: 253
      Relay_Master_Log_File: mysql-bin.000048
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
```

当看到Slave_IO_Running: YES、Slave_SQL_Running: YES才表明状态正常

实际测试:

--登陆192.168.95.11主MySQL

```
mysql>SHOW DATABASES;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)
```

--登陆192.168.95.12从MySQL

```
mysql>SHOW DATABASES;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)
```

192.168.95.11主MySQL操作:

```
mysql>create database aa;
```

```
mysql>use aa;
```

```
mysql>create table tab1(id int auto_increment,name varchar(10),primary key(id));
```

```
mysql>show databases;
```

```
mysql>show tables;
```

```
aa
mysql
performance_schema
test
+-----+
5 rows in set (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_aa |
+-----+
| tab1          |
+-----+
1 row in set (0.00 sec)
```

192.168.95.12从MySQL操作:

```
mysql> show databases;
```

```
mysql> show tables;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| aa        |
| mysql     |
| performance_schema |
| test      |
+-----+
5 rows in set (0.07 sec)

mysql> use aa
Database changed
mysql> show tables;
+-----+
| Tables_in_aa |
+-----+
| tab1          |
+-----+
1 row in set (0.00 sec)
```

由上面两个结果图可得知，两主机达到了数据同步。主从复制的配置就是如此的简单。

[回到顶部](#)

4、MySQL主主复制

[回到顶部](#)

4.1、实现原理

主主复制即在两台MySQL主机内都可以变更数据，而且另外一台主机也会做出相应的变更。聪明的你也许已经想到该怎么实现了。对，就是将两个主从复制有机合并起来就好了。只不过在配置的时候我们需要注意一些问题，例如，主键重复，server-id不能重复等等。

[回到顶部](#)

4.2、配置文件

--192.168.95.11

server-id=11 #任意自然数n，只要保证两台MySQL主机不重复就可以了。

log-bin=mysql-bin #开启二进制日志

auto_increment_increment=2 #步进值auto_increment。一般有n台主MySQL就填n

auto_increment_offset=1 #起始值。一般填第n台主MySQL。此时为第一台主MySQL

binlog-ignore=mysql #忽略mysql库【我一般都不写】

```
server-id=12

log-bin=mysql-bin

auto_increment_increment=2

auto_increment_offset=2

replicate-do-db=aa

配置好后重启MySQL
```

[回到顶部](#)

4.3、开始构建主主复制

因为主主复制是两个主从复制组合一起，所以我就接着上面主从复制接着配置。

第一步：

在192.168.95.12中创建一个192.168.95.11主机中可以登录的MySQL用户

用户：mysql11

密码：mysql11

```
mysql>GRANT REPLICATION SLAVE ON *.* TO 'mysql11'@'192.168.95.11' IDENTIFIED
BY 'mysql11';
```

```
mysql>FLUSH PRIVILEGES;
```

第二步：

在192.168.95.12查看二进制日志名和位置

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

```
mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000084 |      107 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

第三步：

告知二进制文件名与位置

在192.168.95.11中执行：

```
mysql>CHANGE MASTER TO
MASTER_HOST='192.168.95.12',
MASTER_USER='mysql11',
MASTER_PASSWORD='mysql11',
MASTER_LOG_FILE='mysql-bin.000084',
MASTER_LOG_POS=107;
```

```
mysql> CHANGE MASTER TO
-> MASTER_HOST='192.168.95.12',
-> MASTER_USER='mysql11',
-> MASTER_PASSWORD='mysql11',
-> MASTER_LOG_FILE='mysql-bin.000084',
-> MASTER_LOG_POS=107;
Query OK, 0 rows affected (0.04 sec)
```

分别开启slave start;

mysql>SHOW SLAVE STATUS\G #查看主从复制是否配置成功

192.168.95.11

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 192.168.95.11
        Master_User: mysql12
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000050
        Read_Master_Log_Pos: 107
        Relay_Log_File: localhost-relay-bin.000008
        Relay_Log_Pos: 253
        Relay_Master_Log_File: mysql-bin.000050
        Slave_IO_Running: Yes
        Slave_SQL_Running: Yes
        Replicate_Do_DB: aa
        Replicate_Ignore_DB:
        Replicate_Do_Table:
```

192.168.95.12

当看到Slave_IO_Running: YES、Slave_SQL_Running: YES才表明状态正常

测试:

--192.168.95.11

mysql>use aa;

mysql>select*from tab1;

tab1无数据

--192.168.95.12

mysql>use aa;

tab1无数据

--192.168.95.11插入数据

mysql>insert into tab1 (name) value('11'),('11'),('11');

--192.168.95.12插入数据

mysql>insert into tab1 (name) value('22'),('22'),('22');

查看数据:

两个主机数据结果一样!

主主复制配置成功!



信息中有错误提示，可根据错误提示进行更正。

3、Slave_IO_Running、Slave_SQL_Running不全为YES时，大多数问题都是数据不统一导致。

常见出错点：

- 1、两台数据库都存在db数据库，而第一台MySQL db中有tab1，第二台MySQL db中没有tab1，那肯定不能成功。
 - 2、已经获取了数据的二进制日志名和位置，又进行了数据操作，导致POS发生变更。在配置CHANGE MASTER时还是用到之前的POS。
 - 3、stop slave后，数据变更，再start slave。出错。
- 终极更正法：重新执行一遍CHANGE MASTER就好了。

(以上是自己的一些见解，若有不足或者错误的地方请各位指出)

作者：[那一叶随风](#)

声明：本博客文章为原创，只代表本人在工作学习中某一时间内总结的观点或结论。转载时请在文章页面明显位置给出原文链接。

执着

分类: MySQL

标签: mysql

好文要顶

关注我

收藏该文



那一叶随风

关注 - 15

粉丝 - 119

+加关注

- « 上一篇: [MySQL常用语句](#)
- » 下一篇: [MySQL查询优化](#)

posted @ 2017-03-01 19:20 那一叶随风 阅读(19622) 评论(9) 编辑 收藏

评论列表

- #1楼 2017-04-12 10:28 Crazy_Yang

```
mysql>GRANT REPLICATION SLAVE ON *.* TO 'mysql11'@'192.168.95.12' IDENTIFIED BY 'mysql11';
```

主主复制, 这个地方应该是192.168.95.11吧

支持(0) 反对(0)
- #2楼[楼主] 2017-04-12 11:53 那一叶随风

@ Crazy_Yang

是11，谢谢提醒哈。已修改了

支持(0) 反对(0)
- #3楼 2017-12-04 09:21 夏天公子

值得学习

支持(0) 反对(0)



#5楼 2018-03-20 10:03 showdbs

有个问题，这种方式是实现的主从复制，在master挂了的情况下slave会提升为master么？

支持(0) 反对(0)

#6楼 2018-03-21 15:04 矛盾gg

辛苦了

支持(0) 反对(0)

#7楼[楼主] 2018-03-21 20:38 那一叶随风

@ showdbs
如果完完全全按照我这篇博客里面来配置，肯定是不能的。这篇博客里介绍的是最简单的主从主主复制分布式配置。如果想要宕机自动切换一般需要中间件辅助加以实现，例如Keepalived中间件。我的另一篇博客有使用到这个软件，有兴趣可以去看看。

支持(0) 反对(0)

#8楼 2018-03-22 10:39 showdbs

@ 那一叶随风
再请教一个问题，在没有中间件的情况下直接调用双主做读写，会有数据一致性问题么？

支持(0) 反对(0)

#9楼 2018-03-31 16:38 Ze_wait

binlog-ignore 现在变成了binlog-ignore-db了

支持(0) 反对(0)

[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

- 谁能救得了中兴通讯？
- 中美贸易战升温，华为暂别美国梦，转战其它市场
- 马斯克致信解释Model3暂停原由：提高产量至每周6000辆
- FB微软等34家科技公司联合声明：不帮政府发动网络攻击
- 苹果iPhone有个设计问题越来越难解决：外部空间已耗尽
- » 更多新闻...

最新知识库文章

- 如何识别人的技术能力和水平？
- 写给自学者的入门指南
- 和程序员谈恋爱
- 学会学习
- 优秀技术人的管理陷阱
- » 更多知识库文章...

