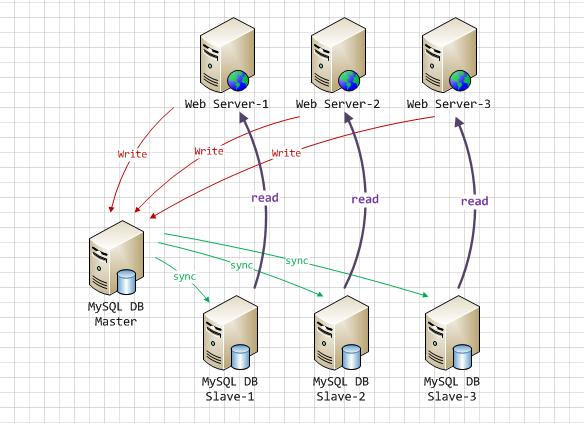
**[MySQL](https://www.cnblogs.com/gl-developer/p/6170423.html)**[主从复制](https://www.cnblogs.com/gl-developer/p/6170423.html)**[(Master-Slave)](https://www.cnblogs.com/gl-developer/p/6170423.html)**[实践](https://www.cnblogs.com/gl-developer/p/6170423.html)

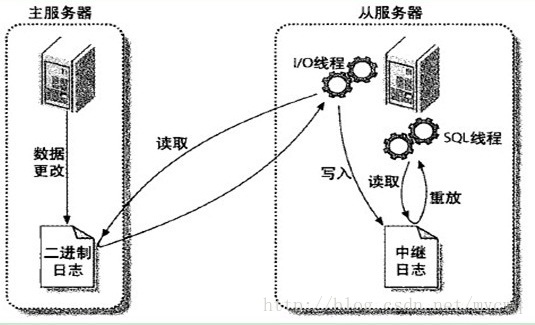
MySQL数据库自身提供的主从复制功能可以方便的实现数据的多处自动备份，实现数据库的拓展。多个数据备份不仅可以加强数据的安全性，通过实现读写分离还能进一步提升数据库的负载性能。

下图就描述了一个多个数据库间主从复制与读写分离的模型(来源网络)：



在一主多从的数据库体系中，多个从服务器采用异步的方式更新主数据库的变化，业务服务器在执行写或者相关修改数据库的操作是在主服务器上进行的，读操作则是在各从服务器上进行。如果配置了多个从服务器或者多个主服务器又涉及到相应的负载均衡问题，关于负载均衡具体的技术细节还没有研究过，今天就先简单的实现一主一从的主从复制功能。

Mysql主从复制的实现原理图大致如下(来源网络)：



MySQL之间数据复制的基础是二进制日志文件（binary log file）。一台MySQL数据库一旦启用二进制日志后，其作为master，它的数据库中所有操作都会以“事件”的方式记录在二进制日志中，其他数据库作为slave通过一个I/O线程与主服务器保持通信，并监控master的二进制日志文件的变化，如果发现master二进制日志文件发生变化，则会把变化复制到自己的中继日志中，然后slave的一个SQL线程会把相关的“事件”执行到自己的数据库中，以此实现从数据库和主数据库的一致性，也就实现了主从复制。

实现**MySQL**主从复制需要进行的配置：

* 主服务器：
  + 开启二进制日志
  + 配置唯一的server-id
  + 获得master二进制日志文件名及位置
  + 创建一个用于slave和master通信的用户账号
* 从服务器：
  + 配置唯一的server-id
  + 使用master分配的用户账号读取master二进制日志
  + 启用slave服务

具体实现过程如下：

一、准备工作：

1.主从数据库版本最好一致

2.主从数据库内数据保持一致

主数据库：182.92.172.80 /linux

从数据库：123.57.44.85 /linux

二、主数据库master修改：

**1.**修改**mysql**配置

找到主数据库的配置文件my.cnf(或者my.ini)，我的在/etc/mysql/my.cnf,在[mysqld]部分插入如下两行：

[mysqld]

log-bin=mysql-bin #开启二进制日志

server-id=**1** #设置server-id

**2.**重启**mysql**，创建用于同步的用户账号

打开mysql会话shell>mysql -hlocalhost -uname -ppassword

创建用户并授权：用户：rel1密码：slavepass

mysql> CREATE USER 'repl'@'123.57.44.85' IDENTIFIED BY 'slavepass';#创建用户

mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'repl'@'123.57.44.85';#分配权限

mysql>flush privileges; #刷新权限

**3.**查看**master**状态，记录二进制文件名**(mysql-bin.000003)**和位置**(73)**：

mysql > SHOW MASTER STATUS;

+------------------+----------+--------------+------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------+----------+--------------+------------------+

| mysql-bin.**000003** | **73** | test | manual,mysql |

+------------------+----------+--------------+------------------+

二、从服务器slave修改：

**1.**修改**mysql**配置

同样找到my.cnf配置文件，添加server-id

[mysqld]

server-id=**2** #设置server-id，必须唯一

**2.**重启**mysql**，打开**mysql**会话，执行同步**SQL**语句(需要主服务器主机名，登陆凭据，二进制文件的名称和位置)：

mysql> CHANGE MASTER TO

-> MASTER\_HOST='182.92.172.80',

-> MASTER\_USER='rep1',

-> MASTER\_PASSWORD='slavepass',

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

-> MASTER\_LOG\_POS=**73**;

**3.**启动**slave**同步进程：

mysql>start slave;

4.查看slave状态：



mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **1**. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: **182.92**.**172.80**

Master\_User: rep1

Master\_Port: **3306**

Connect\_Retry: **60**

Master\_Log\_File: mysql-bin.**000013**

Read\_Master\_Log\_Pos: **11662**

Relay\_Log\_File: mysqld-relay-bin.**000022**

Relay\_Log\_Pos: **11765**

Relay\_Master\_Log\_File: mysql-bin.**000013**

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB:

Replicate\_Ignore\_DB:

...



当Slave\_IO\_Running和Slave\_SQL\_Running都为YES的时候就表示主从同步设置成功了。接下来就可以进行一些验证了，比如在主master数据库的test数据库的一张表中插入一条数据，在slave的test库的相同数据表中查看是否有新增的数据即可验证主从复制功能是否有效，还可以关闭slave（mysql>stop slave;）,然后再修改master，看slave是否也相应修改（停止slave后，master的修改不会同步到slave），就可以完成主从复制功能的验证了。

还可以用到的其他相关参数：

master开启二进制日志后默认记录所有库所有表的操作，可以通过配置来指定只记录指定的数据库甚至指定的表的操作，具体在mysql配置文件的[mysqld]可添加修改如下选项：



# 不同步哪些数据库

binlog-ignore-db = mysql

binlog-ignore-db = test

binlog-ignore-db = information\_schema

# 只同步哪些数据库，除此之外，其他不同步

binlog-do-db = game



如之前查看master状态时就可以看到只记录了test库，忽略了manual和mysql库。