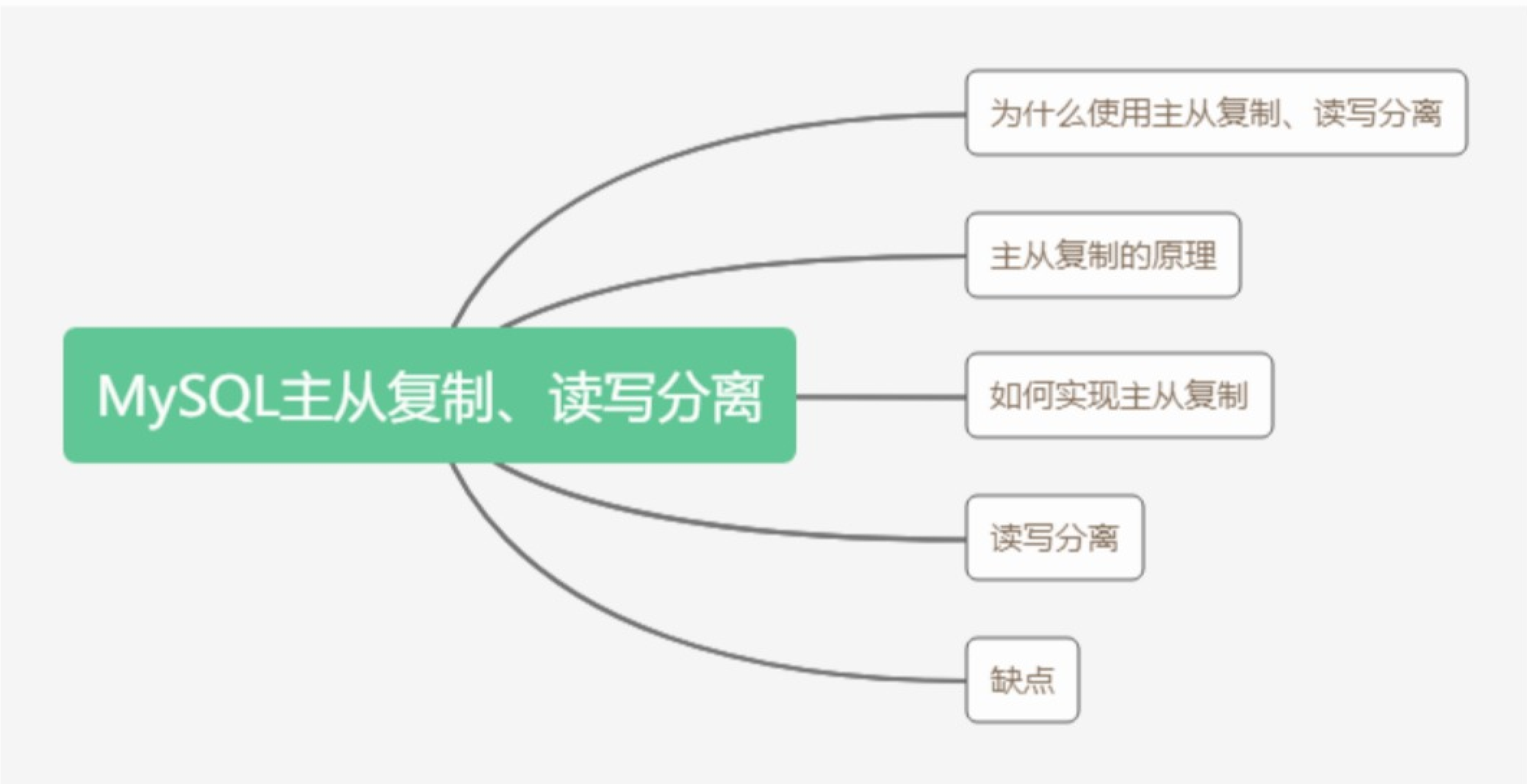
MySQL主从复制读写分离，看这篇就够了！

原创 阿里云开发者 于 2020-08-26 16:06:36 发布 13398 收藏 711 版权
文章标签： SQL druid 关系型数据库 MySQL Java 数据库连接 程序员 数据库 Maven 数据安全/隐私保护

简介：什么是主从复制，如何实现读写分离，看这篇你就懂了！

思维导图



文章已收录到我的Github精选，欢迎Star： https://github.com/yehongzhi/learningSummary

前言

在很多项目，特别是互联网项目，在使用MySQL时都会采用主从复制、读写分离的架构。

为什么要采用主从复制读写分离的架构？如何实现？有什么缺点？让我们带着这些问题开始这段学习之旅吧！

为什么使用主从复制、读写分离

主从复制、读写分离一般是一起使用的。目的很简单，就是**为了提高数据库的并发性能**。你想，假设是单机，读写都在一台MySQL上面完成，性能肯定不高。如果有三台MySQL，一台mater只负责写操作，两台salve只负责读操作，性能不就能大大提高了吗？

所以**主从复制、读写分离就是为了数据库能支持更大的并发。**

随着业务量的扩展、如果是单机部署的MySQL，会导致I/O频率过高。采用**主从复制、读写分离可以提高数据库的可用性。**

主从复制的原理

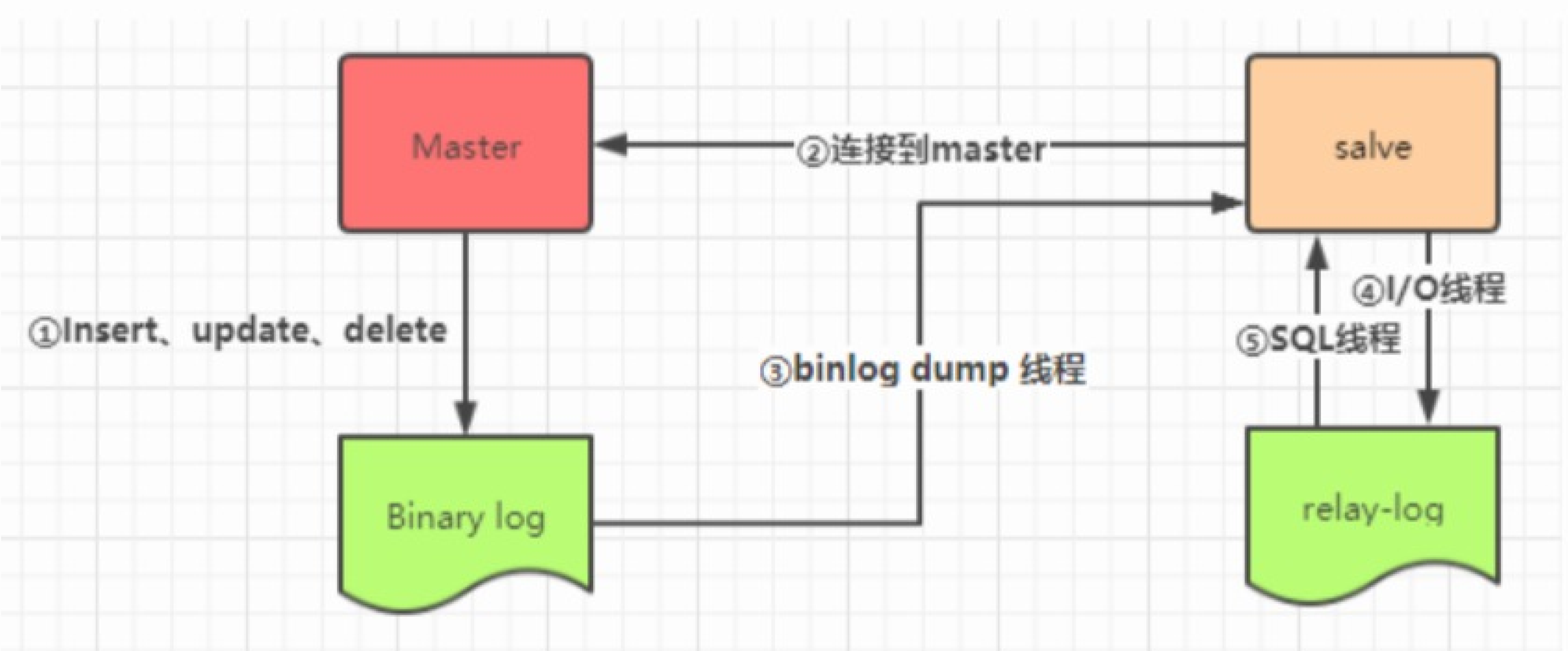
①当Master节点进行insert、update、delete操作时，会按顺序写入到binlog中。

②salve从库连接master主库，Master有多少个slave就会创建多少个binlog dump线程。

③当Master节点的binlog发生变化时，binlog dump 线程会通知所有的salve节点，并将相应的binlog内容推送给slave节点。

④I/O线程接收到 binlog 内容后，将内容写入到本地的 relay-log。

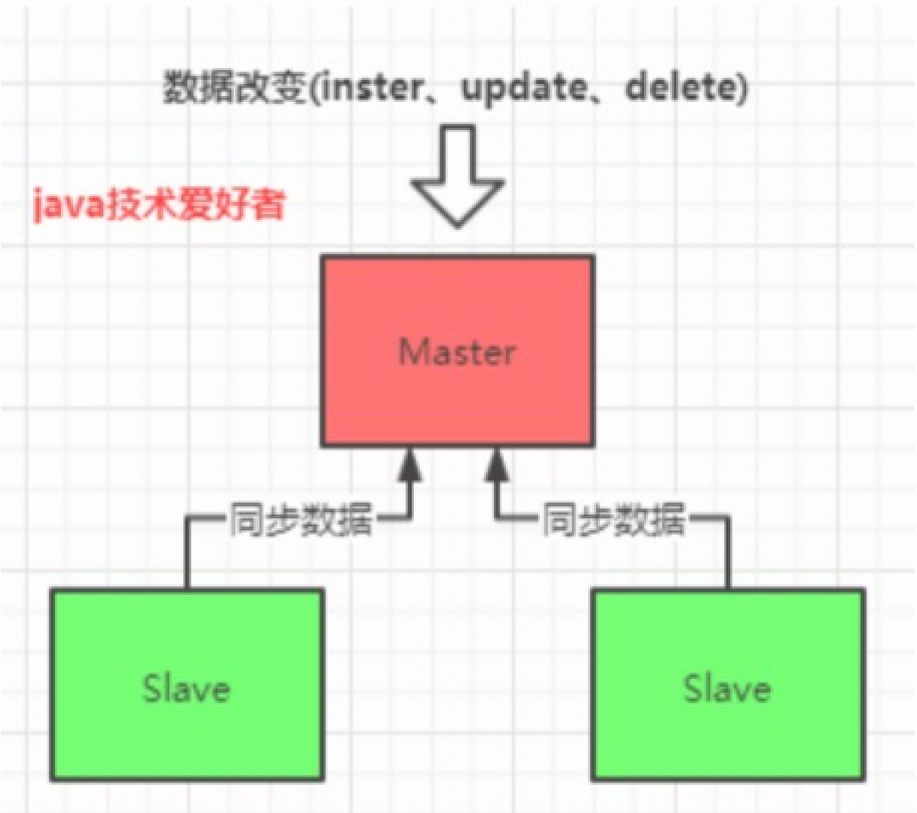
⑤SQL线程读取I/O线程写入的relay-log，并且根据 relay-log 的内容对从数据库做对应的操作。



如何实现主从复制

我这里用三台虚拟机(Linux)演示，IP分别是104(Master)，106(Slave)，107(Slave)。

预期的效果是一主二从，如下图所示：



Master配置

使用命令行进入mysql：

```
mysql -u root -p
```

接着输入root用户的密码(密码忘记的话就网上查一下重置密码吧~)，然后创建用户：

```
//192.168.0.106是slave从机的IP
GRANT REPLICATION SLAVE ON *.* to 'root'@'192.168.0.106' identified by 'Java@1234';
//192.168.0.107是slave从机的IP
GRANT REPLICATION SLAVE ON *.* to 'root'@'192.168.0.107' identified by 'Java@1234';
//刷新系统权限表的配置
FLUSH PRIVILEGES;
```

创建的这两个用户在配置slave从机时要用到。

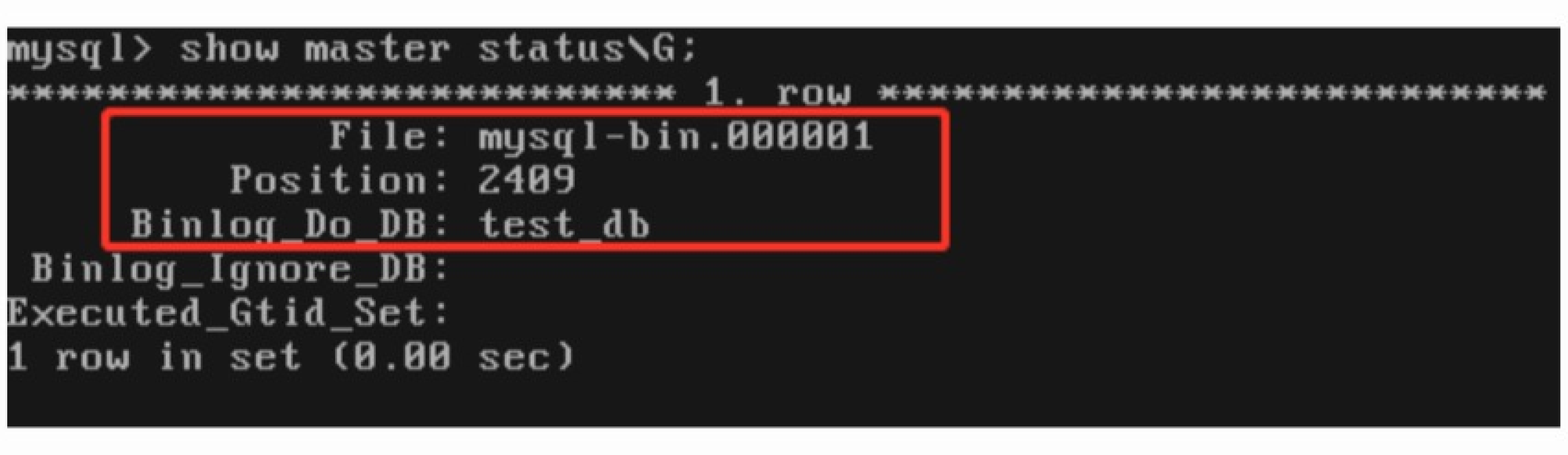
接下来在找到mysql的配置文件/etc/my.cnf，增加以下配置：

```
# 开启binlog
log-bin=mysql-bin
server-id=104
# 需要同步的数据库，如果不配置则同步全部数据库
binlog-do-db=test_db
# binlog日志保留的天数，清除超过10天的日志
# 防止日志文件过大，导致磁盘空间不足
expire-logs-days=10
```

配置完成后，重启mysql：

```
service mysql restart
```

可以通过命令行 `show master status\G;` 查看当前binlog日志的信息(后面有用)：


```
mysql> show master status\G;
*************************** 1. row ***************************
             File: mysql-bin.000001
         Position: 2409
     Binlog_Do_DB: test_db
 Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)
```

Slave配置

Slave配置相对简单一点。从机肯定也是一台MySQL服务器，所以和Master一样，找到/etc/my.cnf配置文件，增加以下配置：

```
# 不要和其他mysql服务id重复即可
server-id=106
```

接着使用命令行登录到mysql服务器：

```
mysql -u root -p
```

然后输入密码登录进去。

进入到mysql后，再输入以下命令：

```
CHANGE MASTER TO
MASTER_HOST='192.168.0.104',//主机IP
MASTER_USER='root',//之前创建的用户账号
MASTER_PASSWORD='Java@1234',//之前创建的用户密码
MASTER_LOG_FILE='mysql-bin.000001',//master主机的binlog日志名称
MASTER_LOG_POS=862,//binlog日志偏移量
master_port=3306;//端口
```

还没完，设置完之后需要启动：

```
# 启动slave服务
start slave;
```

启动完之后怎么校验是否启动成功呢？使用以下命令：

```
show slave status\G;
```

可以看到如下信息（摘取部分关键信息）：

```
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.0.104
                  Master_User: root
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 619
               Relay_Log_File: mysqld-relay-bin.000001
                Relay_Log_Pos: 782
        Relay_Master_Log_File: mysql-bin.000001 //binlog日志文件名称
             Slave_IO_Running: Yes //Slave_IO线程、SQL线程都在运行
            Slave_SQL_Running: Yes
             Master_Server_Id: 104 //master主机的服务id
                  Master_UUID: 0ab6b3a6-e21d-11ea-aaa3-080027f8d62
```

另一台slave从机配置一样，不再赘述。

测试主从复制

在master主机执行sql：

```
CREATE TABLE `tb_commodity_info` (
  `id` varchar(32) NOT NULL,
  `commodity_name` varchar(512) DEFAULT NULL COMMENT '商品名称',
  `commodity_price` varchar(36) DEFAULT '0' COMMENT '商品价格',
  `number` int(10) DEFAULT '0' COMMENT '商品数量',
  `description` varchar(2048) DEFAULT '' COMMENT '商品描述',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='商品信息表';
```

接着我们可以看到两台slave从机同步也创建了商品信息表：

在这里插入图片描述

主从复制就完成了！java技术爱好者有点东西哦~