

主从同步

C-2 创建: 张林伟, 最后修改: 张林伟 昨天 14:34

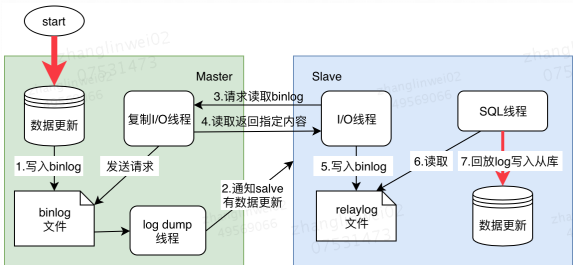
主从复制是一个异步过程，从 Master 到 Slave 的复制过程需要三个线程完成。

mysql主从复制需要三个线程：**master（binlog dump thread）、slave（I/O thread、SQL thread）**

binlog dump线程：主库中有数据更新时，根据设置的binlog格式，将更新的事件类型写入到主库的binlog文件中，并创建log dump线程通知slave有数据更新。当I/O线程请求日志内容时，将此时的binlog名称和当前更新的位置同时传给slave的I/O线程。

I/O线程：该线程会连接到master，向log dump线程请求一份指定binlog文件位置的副本，并将请求回来的binlog存到本地的relay log中。

SQL线程：该线程检测到relay log有更新后，会读取并在本地做redo操作，将发生在主库的事件在本地重新执行一遍，来保证主从数据同步。



整个复制过程实际上就是 Slave 从 Master 端获取该日志然后再在自己身上完全**顺序地**执行日志中所记录的各种操作。

过程解析

- 主库写入数据并且生成binlog文件。该过程中MySQL将事务串行的写入二进制日志，即使事务中的语句都是交叉执行的。
- 在事件写入二进制日志完成后，master通知存储引擎提交事务。
- 从库服务器上的IO线程连接Master服务器，请求从执行binlog日志文件中的指定位置开始读取binlog至从库。
- 主库接收到从库的IO线程请求后，其上复制的IO线程会根据Slave的请求信息分批读取binlog文件然后返回给从库的IO线程。（master第一次接受slave同步请求后，之后都是master dump thread检测到binary log变化后持续推送）
- Slave服务器的IO线程获取到Master服务器上IO线程发送的日志内容、日志文件及位置点后，会将binlog日志内容依次写到Slave端自身的Relay Log（即中继日志）文件的最末端，并将新的binlog文件名和位置记录到master-info文件中，以便下一次读取master端新binlog日志时能告诉Master服务器从新binlog日志的指定文件及位置开始读取新的binlog日志内容。
- 从库服务器的SQL线程会实时监测到本地Relay Log中新增了日志内容，然后把RelayLog中的日志翻译成SQL并且按照顺序执行SQL来更新从库的数据。
- 从库在relay-log.info中记录当前应用中继日志的文件名和位置点以便下一次数据复制。

高并发情况下，单台 MySQL 数据库承载的连接数多、读写压力大，MySQL系统瓶颈凸显。因此，一般来说都是通过 主从复制（Master-Slave）的方式来同步数据，再通过读写分离（MySQL-Proxy）来提升数据库的并发负载能力进行部署与实施。

大部分互联网场景，数据模型「一写多读」。

读次数（read_num）一般是写次数（write_num）的 10 倍以上

一般主库负责写，从库负责读。

主从延迟：新增一条记录，首先写入主库，主库记录信息完成后同步给从库，主库完成记录插入和从库成功同步主库信息的时间差就是DB主从延迟。

影响主从延迟的主要因素是：从库消费relaylog

- 从库的机器性能比主库要差
- 从库的压力大（读）
- 大事务的执行（产生几百G的binlog）
- DDL（alter、drop、repair、create）耗时长
- 锁冲突
- 从库的复制能力（mysql在5.6版本之前，是只支持单线程）

MySQL 支持的复制类型

- 基于语句的复制：在主服务器上执行的SQL语句，在从服务器上执行同样的语句。MySQL默认采用基于语句的复制，效率比较高。一旦发现没法精确复制时，会自动选着基于行的复制
- 基于行的复制：把改变的内容复制过去，而不是把命令在从服务器上执行一遍

参考资料：

DB主从延迟

<https://blog.csdn.net/hguisu/article/details/7325124>

<https://www.jianshu.com/p/0d07b446ae33>

仅供内部使用，未经授权，切勿外传