链滴

MySQL 学习笔记 ----- 全局锁和表锁

作者：wky181

原文链接：https://ld246.com/article/1580804696863

来源网站：链滴

全局锁

全局锁就是对**整个数据库实例加锁。**MySQL提供了一个加全局读锁的方法，命令是 Flush tables with ead lock (FTWRL)，之后其它线程的所有语句都会被阻塞。全局锁很少使用，因为它要对整个库加锁，**型使用场景是，做全库逻辑备份。**

那么最直观的全库备份的做法，就是直接执行FTWRL，使库处于只读状态，然后进行备份，这样就有个缺点：

1、如果对主库备份，期间都不能对主库执行增删改操作，业务几乎处于暂停状态。

2、如果对从库备份，从库不能接受主库传过来的binlong，导致主从延迟高。

其实我们可以让数据库在执行备份的时候，同时又可以对其进行增删改操作。但是这样有一个问题，如银行系统要维护，需要将数据库备份，假设备份期间，有一个用户A用系统给用户B转账了3000元如果时间顺序上是先备份A的余额，然后用户A先转钱，B收到钱，再备份B的余额，那么当用备份表复数据时，A用户的余额没有减少！用户B白白多了3000块！

这样银行可就亏大发了，所以显然是不行的。所以官方自带的逻辑备份工具mysqldump，当mysqldmp使用参数--single-transaction的时候，会启动一个事务，确保拿到一致性视图。而由于**MVCC的持**，这个过程中数据是可以**正常更新的。**参数--single-transaction　导数据之前就会启动一个事务隔离级别为**可重复读**），来确保拿到一致性视图。

解释：因为备份库，就是把库中数据都查一遍，所以在查询前，开启事务，保证查询的数据都是在一视图里面，这就是一致性视图，主要应用了MVCC机制，查询事务开启后会生成一个事务id，之后查的数据行的事务id，都要小于等于这个事务id，保证数据的一致性。所以备份期间的增删改的数据，会对备份造成影响。（备份期间的增删改的数据不会在备份的库中）。

业务的更新不只是**增删改数据（DML)**，还有可能是加字段等修改表结构的**操作（DDL）**。不论是哪方法，一个库被全局锁上以后，你要对里面任何一个表做加字段操作，都是会被锁住的。

表级锁

MySQL里面表级别的锁有两种：一种是表锁，一种是元数据锁（meta data lock，MDL)。

表锁的语法是 lock tables … read/write。与FTWRL类似，可以用unlock tables主动释放锁，也以在客户端断开的时候自动释放。需要注意，lock tables语法除了会限制别的线程的读写外，也限定本线程接下来的操作对象。

另一类表级的锁是MDL（metadata lock)。MDL不需要显式使用，在访问一个表的时候被自动加上。MDL的作用是，保证读写的正确性。你可以想象一下，如果一个查询正在遍历一个表中数据，而执行期间另一个线程对这个表结构做变更，删了一列，那么查询线程拿到的结果跟表结构对上，肯定是不行的。

在MySQL 5.5版本中引入了MDL，当对一个表做增删改查操作的时候，加MDL读锁；当要对表做结变更操作的时候，加MDL写锁。这意味着，**多个线程**可以对**同一张表**进行增删改查。如果有两个线程同时给一个表加字段，其中一个要等另一个执行完才能开始执行。

所以在更改表结构时要格外小心，如果是一个热门表，更改表结构要加MDL写锁，这时对热门表的增改查就要阻塞，，这个库的线程很快就会爆满，对用户体验也不好。而且一旦完成更改，释放掉MDL锁，阻塞的线程同时运行，也容易造成行锁死锁。

上一篇：MySQL学习笔记-----深入浅出索引（下）