图文 47 简单回顾一下， MySQL运行时多个事务同时执行是什么场景？

手机观看

528 人次阅读 2020-03-26 07:00:00

详情 评论

简单回顾一下， MySQL运行时多个事务同时执行是什么场景?

- **如何提问**：每篇文章都有评论区，大家在评论区留言提问
- **如何加入狸猫技术交流群**：
  - 添加微信号：Lvgu0715_ （微信名：绿小九），狸猫技术窝的管理员
  - 发送专栏购买截图
  - 2小时内管理员会拉群，人工操作请耐心等待

到目前为止，我们已经给大家深入讲解了MySQL的buffer pool机制、redo log机制和undo log机制，相信大家现在对我们平时执行一些增删改语句的实现原理，都有了一定较为深入的理解了！

因为平时我们执行增删改的时候，无非就是从磁盘加载数据页到buffer pool的缓存页里去，对缓存页进行更新，同时记录下来undo log回滚日志和redo log重做日志，应对的是事务提交之后MySQL挂了恢复数据的场景，以及事务回滚的场景。

那么接下来其实我们要理解的东西，就是要提高一个层次了，我们要理解到事务这个层面了

所谓的事务呢，其实或多或少每个人都是知道一点的，我们今天只不过是站在MySQL内核原理的角度，拔高到事务的层面给大家回顾一下。

其实大家可以思考，平时我们是不是一般都是写一个业务系统，然后业务系统会去对数据库执行增删改查？

好，我们看下图

然后我们应该都知道一件事情，通常而言，我们都是在业务系统里会开启事务来执行增删改操作的，我随便给大家举个例子，下面的代码大家看看。

```java
@Transactional
public void doService() {
    // 增加一条数据
    addUser();
    // 修改一条数据
    updateUser();
    // 删除一条数据
    deleteUser();
}
```

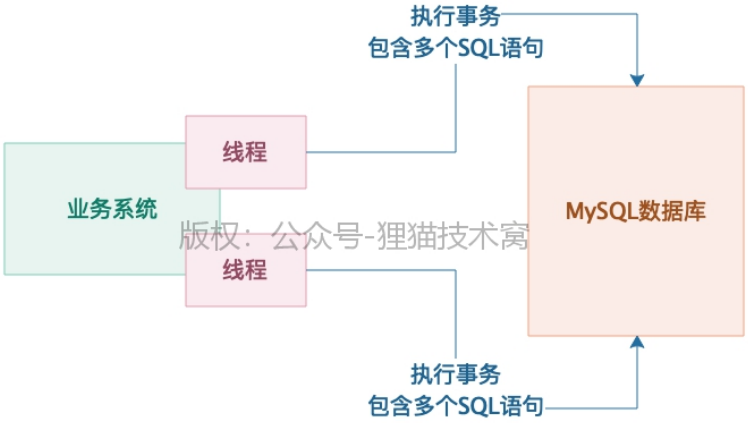
所以一般来说，业务系统是执行一个一个的事务，每个事务里可能是一个或者多个增删改查的SQL语句

这个事务的概念想必不用我多说了，其实就是一个事务里的SQL要不然一起成功就提交了，要不然有一个SQL失败，那么事务就回滚了，所有SQL做的修改都撤销了！我们看下图。

接着问题来了，这个业务系统他可不是一个单线程系统啊！他是有很多线程的！

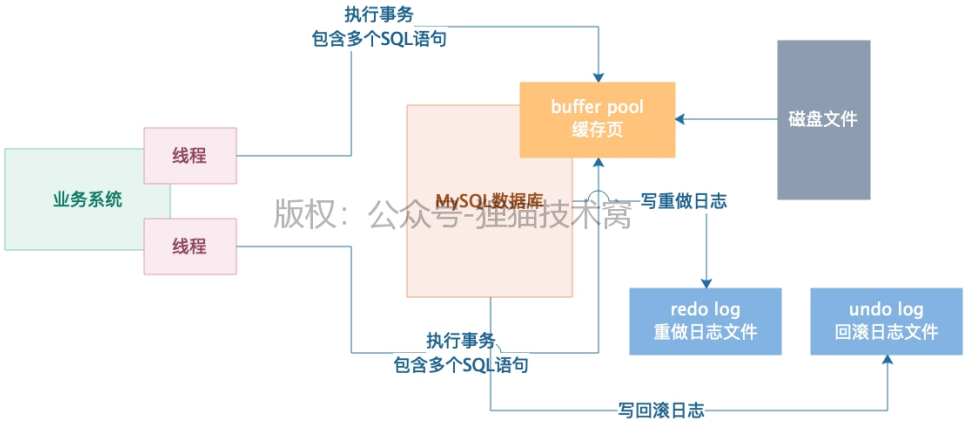
因为他面向的终端用户是有很多人的，可能会同时发起请求，所以他需要多个线程并发来处理多个请求的

于是，这个业务系统很可能是基于多线程并发的对MySQL数据库去执行多个事务的！看下图。



那么每个事务里面的多个SQL语句都是如何执行的呢？

其实就是我们之前给大家讲过的那一套原理了，包括从磁盘加载数据页到buffer pool的缓存页里去，然后更新buffer pool里的缓存页，同时记录redo log和undo log，如下图所示。



每个事务如果提交了，那么就皆大欢喜，这个提交的过程我之前最早就讲过了，他有一些步骤，包括在redo log里记录事务提交标识之类的。

如果事务提交之后，redo log刷入磁盘，结果MySQL宕机了，是可以根据redo log恢复事务修改过的缓存数据的。

如果要回滚事务，那么就基于undo log来回滚就可以了，把之前对缓存页做的修改都给回滚了就可以了。

这就是在MySQL内核层面，把多个事务和我们之前讲解的buffer pool、redo log、undo log几个机制都结合在一起的一个场景讲解。想必大家都是可以理解的。

但是这里就有很多问题了：

- 多个事务并发执行的时候，可能会同时对缓存页里的一行数据进行更新，这个冲突怎么处理？是否要加锁？
- 可能有的事务在对一行数据做更新，有的事务在查询这行数据，这里的冲突怎么处理？

所以接下来，我们要给大家讲解的，**就是解决多个事务并发运行的时候，同时写和同时读写的一些并发冲突的处理机制，包括了MySQL事务的隔离级别、MVCC多版本隔离、锁机制，等等。**

End

狸猫技术窝精品专栏及课程推荐：

- 《从零开始带你成为消息中间件实战高手》
- 《21天互联网Java进阶面试训练营》（分布式篇）
- 《互联网Java工程师面试突击》（第1季）
- 《互联网Java工程师面试突击》（第3季）
- 《从零开始带你成为JVM实战高手》