



图文 40 重新回顾redo日志对于事务提交后，数据绝对不会丢失的意义

手机观看

464 人次阅读 2020-03-12 08:20:30

详情 评论

重新回顾redo日志对于事务提交后，数据绝对不会丢失的意义

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）

之前我们在给大家介绍了大量的MySQL底层原理知识之后，理论结合实践，给大家讲解了两个真实的生产环境的数据库优化案例，一个是数据库所在服务器的RAID存储系统的锂电池充放电导致的性能抖动问题，一个是数据库底层的linux操作系统的文件句柄限制导致的无法连接问题，相信大家在学习了理论知识之后，再来看这些真实的实战案例，会有不错的感觉。

那么接着在学习了两个实战案例之后，我们就要继续进行深入底层的MySQL原理剖析了，我们之前都知道，在我们执行增删改操作的时候，首先会在Buffer Pool中更新缓存页，那么缓存页和底层的物理磁盘上的数据页的原理，之前都已经详细讲解过了



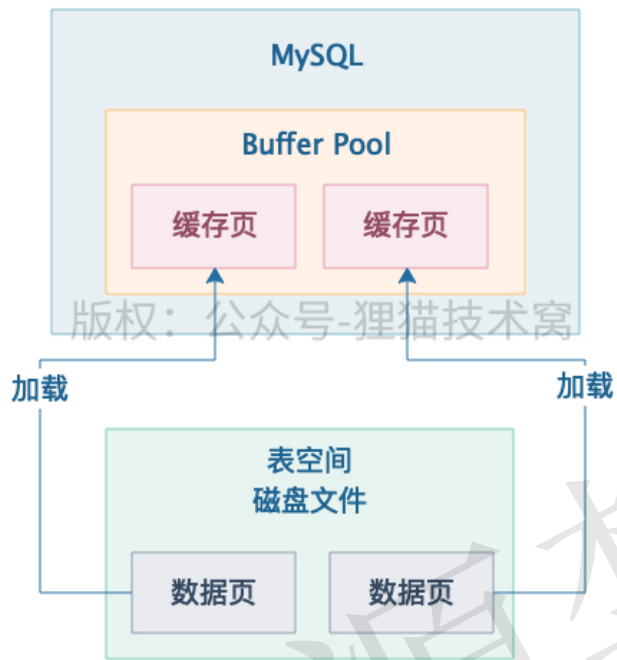
接着我们都知道，在更新完Buffer Pool中的缓存页之后，必须要写一条redo log，这样才能记录下来我们对数据库做的修改。

redo log可以保证我们事务提交之后，如果事务中的增删改SQL语句更新的缓存页还没刷到磁盘上去，此时MySQL宕机了，那么MySQL重启过后，就可以把redo log重做一遍，恢复出来事务当时更新的缓存页，然后再把缓存页刷到磁盘就可以了

redo log本质是保证事务提交之后，修改的数据绝对不会丢失的。

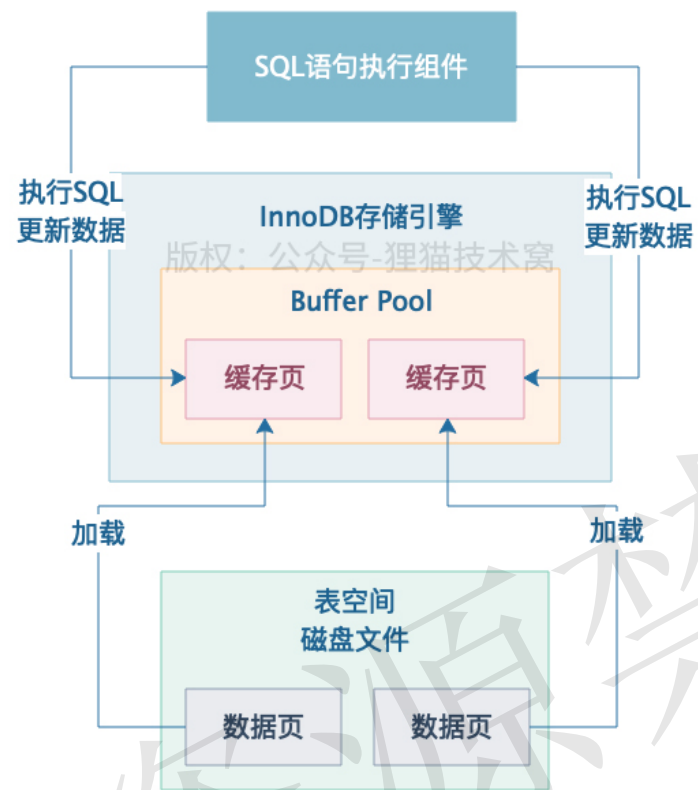
所以接下来一段时间我们会深入研究redo log的底层实现原理，今天我们就承上启下，给大家简单回顾一下redo log这个机制存在的意义。

首先我们都知道，执行增删改SQL语句的时候，都是针对一个表中的某些数据去执行的，此时的话，首先必须找到这个表对应的表空间，然后找到表空间对应的磁盘文件，接着从磁盘文件里把你要更新的那批数据所在的数据页从磁盘读取出来，放到Buffer Pool的缓存页里去，如下图所示。

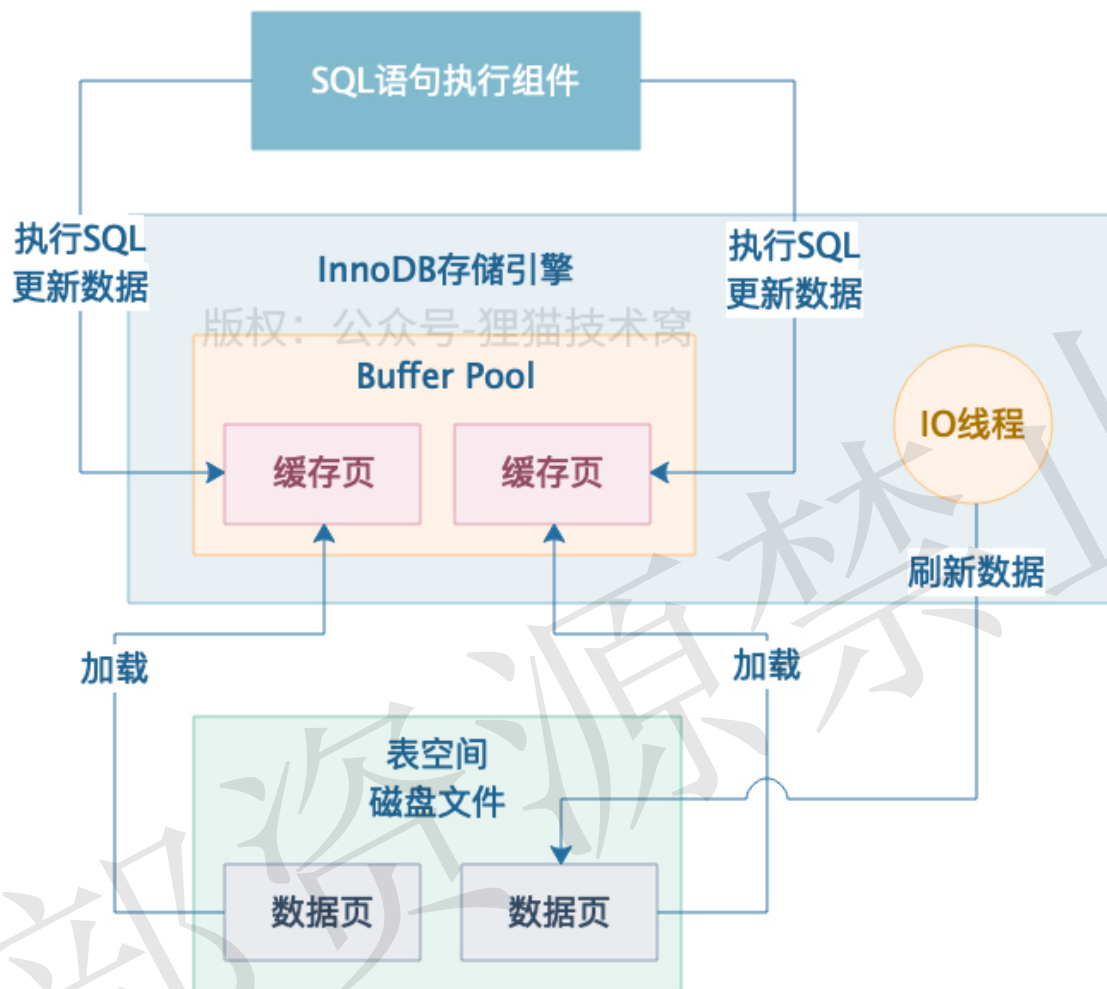


接着实际上你的增删改SQL语句就会针对Buffer Pool中的缓存页去执行你的更新逻辑，比如插入一行数据，或者更新一行数据，或者是删除一行数据。

当然，删除数据的逻辑我们还没讲，后续很快就要讲到了。至于说数据页和数据行的格式，就不用我多说了，其实都是MySQL自己定义的，之前都讲过了，大家现在对这些都知道了，如下图。



那么学习过之前的Buffer Pool底层原理之后都知道，其实你更新缓存页的时候，会更新free链表、flush链表、lru链表，然后有专门的后台IO线程，不定时的根据flush链表、lru链表，会把你更新过的缓存页刷新回磁盘文件的数据页里去，如下图所示。

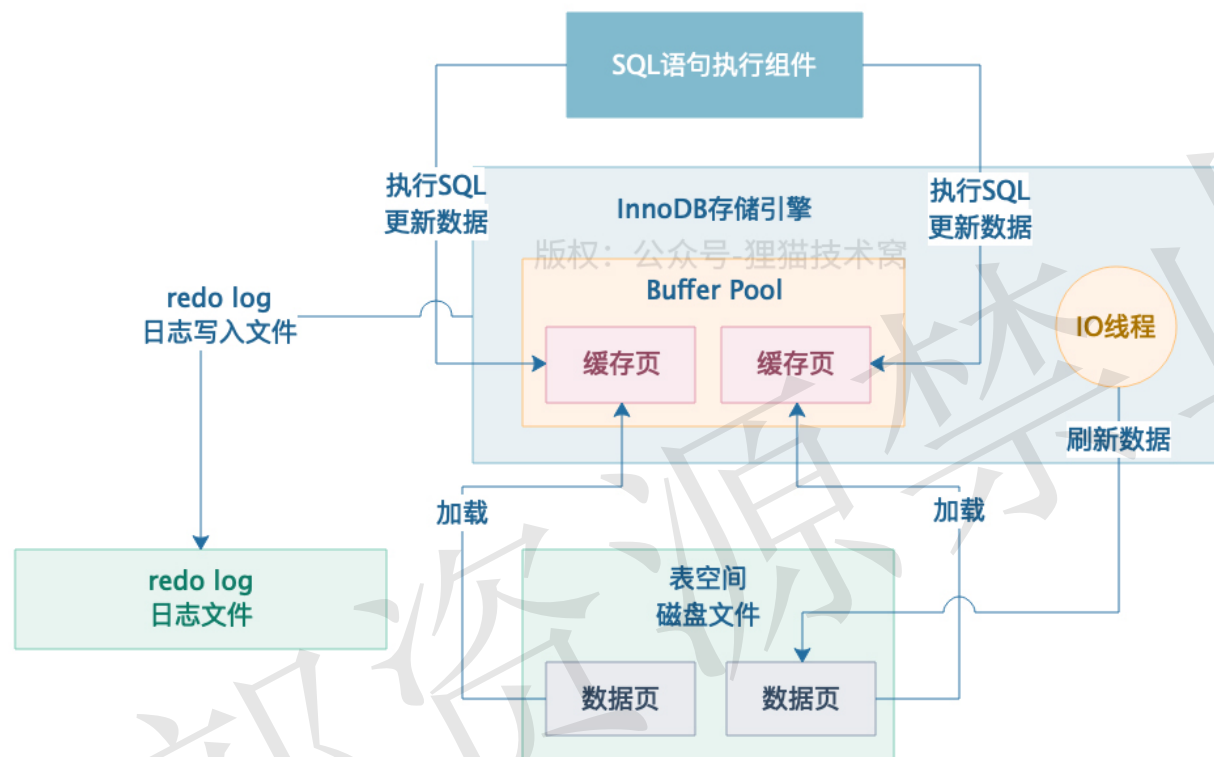


所以大家都知道这个机制里最大的漏洞就在于，万一你一个事务里有增删改SQL更新了缓存页，然后事务提交了，结果万一你还没来得及让IO线程把缓存页刷新到磁盘文件里，此时MySQL宕机了，然后内存数据丢失，你事务更新的数据就丢失了！

但是也不可能每次你事务一提交，就把你事务更新的缓存页都刷新回磁盘文件里去，因为大家之前也都知道，缓存页刷新到磁盘文件里，是随机磁盘读写，性能是相当的差！这会导致你数据库性能和并发能力都很弱的！

所以此时才会引入一个redo log机制，这个机制就是说，你提交事务的时候，绝对是保证把你对缓存页做的修改以日志的形式，写入到redo log日志文件里去的

这种日志大致的格式如下：对表空间XX中的数据页XX中的偏移量为XXXX的地方更新了数据XXX。如下图所示



只要你事务提交的时候保证你做的修改以日志形式写入redo log日志，那么哪怕你此时突然宕机了，也没关系！

因为你MySQL重启之后，把你之前事务更新过做的修改根据redo log在Buffer Pool里重做一遍就可以了，就可以恢复出来当时你事务对缓存页做的修改，然后找时机再把缓存页刷入磁盘文件里去。

☑ 那么有人会问了，你事务提交的时候把修改过的缓存页都刷入磁盘，跟你事务提交的时候把你做的修改的redo log都写入日志文件，他们不都是写磁盘么？差别在哪里？

这是本文一个关键的问题。

实际上，如果你把修改过的缓存页都刷入磁盘，这首先缓存页一个就是16kb，数据比较大，刷入磁盘比较耗时，而且你可能就修改了缓存页里的几个字节的数据，难道也把完整的缓存页刷入磁盘吗？

而且你缓存页刷入磁盘是随机写磁盘，性能是很差的，因为他一个缓存页对应的位置可能在磁盘文件的一个随机位置，比如偏移量为45336这个地方。

但是如果是写redo log，第一个一行redo log可能就占据几十个字节，就包含表空间号、数据页号、磁盘文件偏移量、更新值，这个写入磁盘速度很快。

此外，redo log写日志，是顺序写入磁盘文件，每次都是追加到磁盘文件末尾去，速度也是很快的。

所以你提交事务的时候，用redo log的形式记录下来你做的修改，性能会远远超过刷缓存页的方式，这也可以让你的数据库的并发能力更强。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

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》](#)



Copyright © 2015-2020 深圳小鹅网络技术有限公司 All Rights Reserved. [粤ICP备15020529号](#)

 小鹅通提供技术支持

内部资源禁止外传