

MySQL跨行事务模型

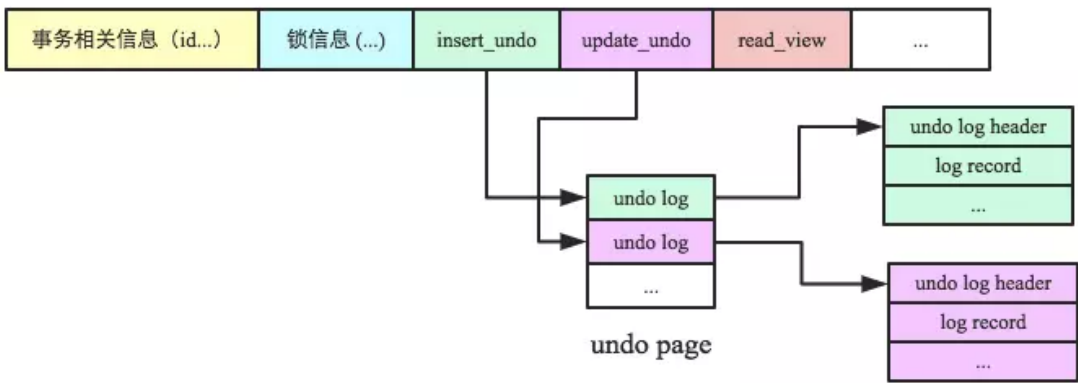


云狗狗狗狗 (/u/15cfc4b14afe) [+ 关注](#)
2017.10.10 16:39* 字数 3279 阅读 498 评论 1 喜欢 3
(/u/15cfc4b14afe)

MySQL事务原子性保证

事务原子性要求事务中的一系列操作要么全部完成，要么不做任何操作，不能只做一半。原子性对于原子操作很容易实现，就像HBase中行级事务的原子性实现就比较简单。但对于多条语句组成的事务来说，如果事务执行过程中发生异常，需要保证原子性就只能回滚，回滚到事务开始前的状态，就像这个事务根本没有发生过一样。如何实现呢？

MySQL实现回滚操作完全依赖于undo log，多说一句，undo log在MySQL除了用来实现原子性保证之外，还用来实现MVCC，下文也会涉及到。使用undo实现原子性在操作任何数据之前，首先会将修改前的数据记录到undo log中，再进行实际修改。如果出现异常需要回滚，系统可以利用undo中的备份将数据恢复到事务开始之前的状态。下图是MySQL中表示事务的基本数据结构，其中与undo相关的字段为insert_undo和update_undo，分别指向本次事务所产生的undo log。



MySQL事务基本数据结构

事务回滚根据update_undo (或者insert_undo) 找到对应的undo log，做逆向操作即可。对于已经标记删除的数据清理删除标记，对于更新数据直接回滚更新；插入操作稍微复杂一些，不仅需要删除数据，还需要删除相关的聚集索引以及二级索引记录。

undo log是MySQL内核中非常重要的一块内容，涉及知识比较多而且复杂，比如：

- 1. undo log必须在数据修改之前持久化，undo log持久化需不需要记录redo以防止宕机异常？如果需要就又涉及宕机恢复...



2. 通过undo log如何实现MVCC？

3. 那些undo log可以在什么场景下回收清理？如何清理？

MySQL事务一致性保证：强一致性事务保证

MySQL事务隔离级别

Read Uncommitted（RU技术解读：使用X锁实现写写并发）

Read Uncommitted只实现了写写并发控制，并没有有效的读写并发控制，导致当前事务可能读到其他事务中还未提交的修改数据，这些数据准确性并不靠谱（有可能被回滚掉），因此在此基础上作出的一切假设就都不靠谱的。在现实场景中很少有业务会选择该隔离级别。

写写并发实现机制和HBase并无两样，都是使用两阶段锁协议对相应记录加行锁实现。不过MySQL中行锁机制比较复杂，根据行记录是否是主键索引、唯一索引、非唯一索引或者无索引等分为多种加锁情况。这里举个简单例子做下说明：update user set userName = "libis" where id = 15。（如果大家有兴趣，可以参考登博的这篇文章：《MySQL加锁处理分析 (<https://link.jianshu.com?t=http://hedengcheng.com/?p=771>)》）

1. 如果id列是主键索引，MySQL只会为聚簇索引记录加锁。
2. 如果id列是唯一二级索引，MySQL会为二级索引叶子节点以及聚簇索引记录加锁。
3. 如果id列是非唯一索引，MySQL会为所有满足条件(id = 15)的二级索引叶子节点以及对应的聚簇索引记录加锁。
4. 如果id列是无索引的，SQL会走聚簇索引全表扫描，并将扫描结果加载到SQL Server层进行过滤，因此InnoDB会为扫描过的所有记录先加上锁，如果SQL Server层过滤不符合条件，InnoDB会释放该锁。因此InnoDB会为扫描到的所有记录都加锁，很恐怖吧！

接下来无论是RC、RR，抑或是Serialization，写写并发控制都使用上述机制，所以不再赘述。接下来会重点分析RC和RR隔离级别中的读写并发控制机制。

在详细介绍RC和RR之前，有必要在此先行介绍MySQL中MVCC机制，因为RC和RR都使用MVCC机制实现事务之间的读写并发。只不过两者在实现细节上有一些区别，具体区别接下来再聊。

MVCC in MySQL

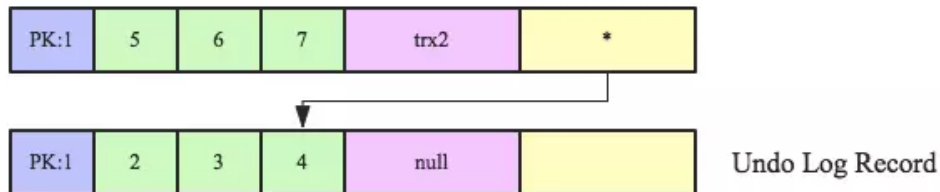
MySQL中MVCC机制相比HBase来说要复杂的多，涉及的数据结构也比较复杂。为了解释的比较清晰，以一个栗子为模版进行解释。比如当前有一行记录如下图所示：



PK:1	2	3	4	DB_TRX_ID	DB_ROLL_PTR
------	---	---	---	-----------	-------------

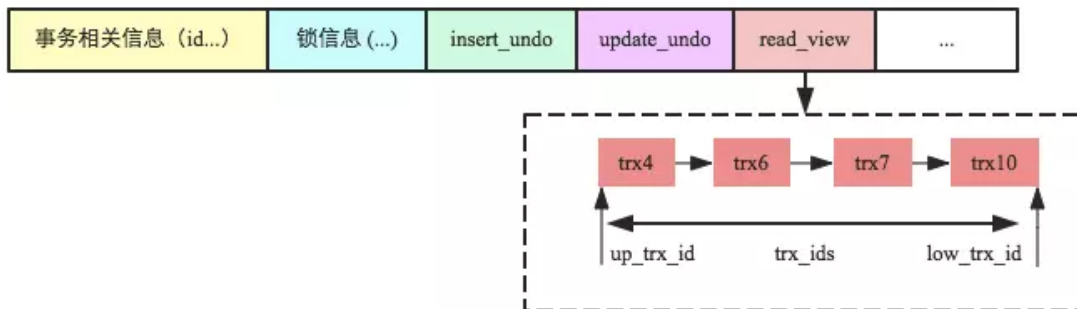
前面四列是该行记录的实际列值，需要重点关注的是DB_TRX_ID和DB_ROLL_PTR两个隐藏列（对用户不可见）。其中DB_TRX_ID表示修改该行事务的事务ID，而DB_ROLL_PTR表示指向该行回滚段的指针，该行记录上所有版本数据，在undo中都通过链表形式组织，该值实际指向undo中该行的历史记录链表。

现在假设有一个事务trx2修改了该行数据，该行记录就会变为下图形式，DB_TRX_ID为最近修改该行事务的事务ID(trx2)，DB_ROLL_PTR指向undo历史纪录链表：



了解了MySQL行记录之后，再来看看事务的基本结构，下图是MySQL的事务数据结构，上文我们提到过。事务在开启之后会创建一个数据结构存储事务相关信息、锁信息、undo log以及非常重要的read_view信息。

read_view保存了当前事务开启时整个MySQL中所有活跃事务列表，如下图所示，在当前事务开启的时候，系统中活跃的事务有trx4、trx6、trx7以及trx10。另外，up_trx_id表示当前事务启动时，当前事务链表中最小的事务ID；low_trx_id表示当前事务启动时，当前事务链表中的最大的事务ID。



read_view是实现MVCC的一个关键点，它用来判断记录的哪个版本对当前事务可见。如果当前事务要读取某行记录，该行记录的版本号（事务ID）为trxid，那么：

1. 如果 $trxid < up_trx_id$ ，说明该行记录所在的事务已经在当前事务创建之前就提交了，所以该行记录对当前事务可见。
2. 如果 $trxid > low_trx_id$ ，说明该行事务所在的事务是在当前事务创建之后才开启，所以该行记录对当前事务不可见。

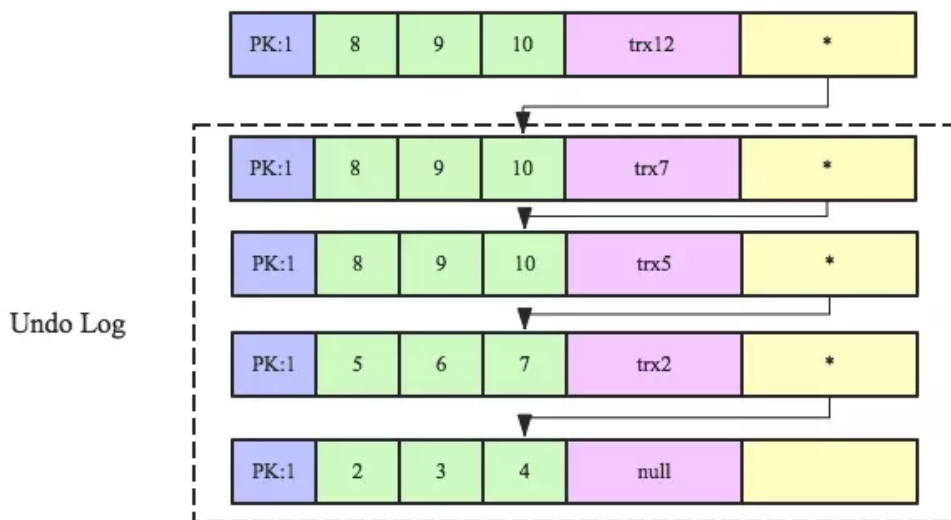
3. 如果 $up_trx_id < trxid < low_trx_id$ ，那么表明该行记录所在事务在本次新事务创建的时候处于活动状态。从 up_trx_id 到 low_trx_id 进行遍历，如果 $trxid$ 等于他们之中的某个事务id的话，那么不可见，否则可见。

以下面行记录为例，该行记录存在多个版本（ $trx2$ 、 $trx5$ 、 $trx7$ 以及 $trx12$ ），其中 $trx12$ 是最新版本。看看该行记录中哪个版本对当前事务可见。

1. 该行记录的最新版本为 $trx12$ ，与当前事务 $read_view$ 进行对比发现， $trx12$ 大于当前活跃事务列表中的最大事务 $trx10$ ，表示 $trx12$ 是在当前事务创建之后才开启的，因此不可见。

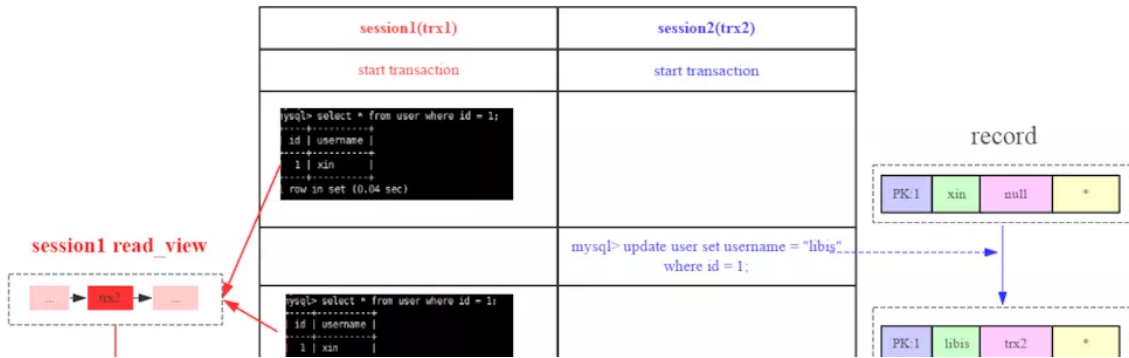
2. 再查看该行记录的第二个最新版本为 $trx7$ ，与当前事务 $read_view$ 对比发现， $trx7$ 介于当前活跃事务列表最小事务ID和最大事务ID之间，表明该行记录所在事务在当前事务创建的时候处于活动状态，在活跃列表中遍历发现 $trx7$ 确实存在，说明该事务还没有提交，所以对当前事务不可见。

3. 继续查看该记录的第三个最新版本 $trx5$ ，也介于当前活跃事务列表最小事务ID和最大事务ID之间，表明该行记录所在事务在当前事务创建的时候处于活动状态，但遍历发现该版本并不在活跃事务列表中，说明 $trx5$ 对应事务已经提交（注：事务提交时间与事务编号没有任何关联，有可能事务编号大的事务先提交，事务编号小的事务后提交），因此 $trx5$ 版本行记录对当前事务可见，直接返回。



Read Committed（技术解读：写写并发使用X锁，读写并发使用MVCC避免脏读）

上文介绍了MySQL中MVCC技术实现机制，但要明白RC隔离级别下事务可见性，还需要get一个核心点：RC隔离级别下的事务在每次执行select时都会生成一个最新的 $read_view$ 代替原有的 $read_view$ 。



如上图所示，左侧为1号事务，在不同时间点对id=1的记录分别查询了三次。右侧为2号事务，对id=1的记录进行了更新。更新前该记录只有一个版本，更新好变成了两个版本。

1号事务在RC隔离级别下每次执行select请求都会生成一个最新的read_view，前两次查询生成的全局事务活跃列表中包含trx2，因此根据MVCC规定查到的记录为老版本；最后一次查询的时间点位于2号事务提交之后，因此生成的全局活跃事务列表中不包含trx2，此时在根据MVCC规定查到的记录就是最新版本记录。

Repeatable Read (

技术解读：写写并发使用X锁，读写并发使用MVCC避免不可重复读；当前读使用Gap锁避免幻读)

和RC模式不同，RR模式下事务不会再每次执行select的时候生成最新的read_view，而是在事务第一次select时就生成read_view，后续不会再变更，直至当前事务结束。这样可以有效避免不可重复读，使得当前事务在整个事务过程中读到的数据都保持一致。示意图如下所示：



这个就很容易理解，三次查询所使用的全局活跃事务列表都一样，且都是第一次生成的read_view，那之后查到的记录必然和第一次查到的记录一致。



RR隔离级别能够避免幻读吗？

如果对幻读还不了解的话，可以参考该系列的第一篇文章。如下图所示，1号事务针对id>1的过滤条件执行了三次查询，2号事务执行了一次插入，插入的记录刚好符合id>1这个条件。可以看出，三次查询得到的数据是一致的，这个是由RR隔离级别的MVCC机制保证的。这么看来，是避免了幻读，但是在最后1号事务在id=2处插入一条记录，MySQL会返回Duplicate entry的错误，可见避免了幻读是一种假象。

session1(trx1)	session2(trx2)
start transaction	start transaction
<pre>mysql> select * from user where id > 0; +-----+ id username +-----+ 1 libis 2 fanny +-----+ 2 rows in set (0.00 sec)</pre>	
<pre>mysql> select * from user where id > 0; +-----+ id username +-----+ 1 libis 2 fanny +-----+ 2 rows in set (0.00 sec)</pre>	insert into user values(3, "xunxing");
<pre>mysql> select * from user where id > 0; +-----+ id username +-----+ 1 libis 2 fanny +-----+ 2 rows in set (0.00 sec)</pre>	commit
<pre>l> insert into user values (3,"xunxing"); R 1062 (23000): Duplicate entry '3' for key 'PRIM</pre>	

窃以为避免了幻读!!!

实际上并没有!!!

严格意义避免幻读（技术解读：当前读使用Gap锁避免幻读）

之前提到的所有RR级别的select语句我们称为快照读，快照读能够保证不可重复读，但并不能避免幻读。于是MySQL又提出“当前读”的概念，常见的当前读语句有：

1. select for update
2. select lock in share mode
3. update / delete

并且规定，RR级别下当前读语句会给记录加上一种特殊的锁 - Gap锁，Gap锁并不锁定某个具体的记录，而是锁定记录与记录之间的间隔，保证这个间隔中不会插入新的其他记录。下图是一个示意图：



上图中1号事务首先执行了一个当前读的select语句，这个语句会在 id > 0的所有间隔加上Gap锁，接下来2号事务在id = 3处执行插入时系统就会返回Lock wait timeout execcded的异常。当然，其他事务可以在id <= 0的条件下插入成功，这没问题。

小礼物走一走，来简书关注我

赞赏支持

mysql (/nb/17473016)

举报文章 © 著作权归作者所有



云狗狗狗狗狗 (/u/15cfc4b14afe)

写了 39766 字，被 7 人关注，获得了 24 个喜欢 (/u/15cfc4b14afe)

+ 关注


一个程序狗

喜欢 | 3

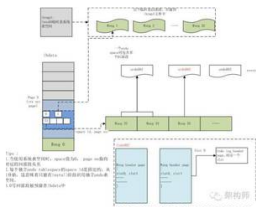


更多分享

《MySQL技术内幕：InnoDB存储引擎(第2版)》书摘 (/p/3eca0b18cf51?ut...
MySQL技术内幕：InnoDB存储引擎(第2版) 姜承尧 第1章 MySQL体系结构和存储引擎 >> 在上述例子中使用了mysqld_safe命令来启动数据库，当然启动MySQL实例的方法还有很多，在各种平台下的方式可能会...


 沉默剑士 (/u/6ca93a173ea2?)
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendenc

(/p/a4680c8c5435?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend MySQL : InnoDB undo log 漫游 (/p/a4680c8c5435?utm_campaign=m...

本文是对整个Undo生命周期过程的阐述，代码分析基于当前最新的MySQL5.7版本。本文也可以作为了解整个Undo模块的代码导读。由于涉及到的模块众多，因此部分细节并未深入。前言 Undo log是InnoDB MV...

 LeiLv (/u/f796a54054d1?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

(/p/9523669cc82e?

last_name	first_name
stark	tony
tom	hiddleston
morgan	freeman
jeff	dean
donald	trump

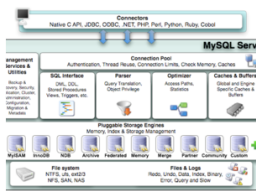
utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend 事务及锁的实现 (/p/9523669cc82e?utm_campaign=maleskine&utm_con...

什么是事务 事务是一条或多条数据库操作语句的组合，具备ACID，4个特点。原子性：要不全部成功，要不全部撤销 隔离性：事务之间相互独立，互不干扰 一致性：数据库正确地改变状态后，数据库的一致性约束...

 jiangmo (/u/de31051e96e1?


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

(/p/bd8675e5c7b2?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend 《高性能MySQL》&《MySQL技术内幕 InnoDB存储引擎》笔记 (/p/bd867...

《高性能MySQL》&《MySQL技术内幕 InnoDB存储引擎》笔记 第一章 MySQL架构与历史 MySQL的架构 从上图可以看出，MySQL数据库区别于其他数据库的最重要的一个特点就是其插件式的表存储引擎。需要...

 xiaogmail (/u/59cac9fff87c?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

浅薄的深刻原因 (/p/af44c6fec474?utm_campaign=maleskine&utm_cont...

半夜失眠，刷了N多年前说说和老照片 感慨万千：当年觉得醍醐灌顶的观点现在觉得人之常情，反倒是当时不以为然的小事回想起来越来越有趣，当年视为金科玉律的道理现在看来大谬不然，反倒是视为理所当然的...

 杨文肃璞 (/u/e5bce6345255?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

(/p/b871ba69850f?

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend

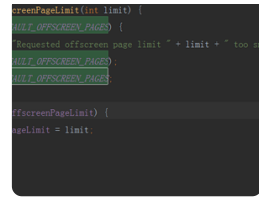
viewpager+fragment 预加载问题 (/p/b871ba69850f?...

viewpager特性：我们在开发过程中经常会用到viewpager+fragment组合来实现页面的切换，谷歌为了增强用户体验，ViewPager加载页面时默认会预加载当...

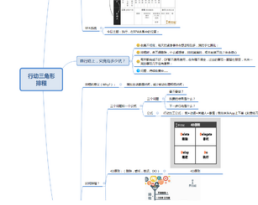


西貝武 (/u/850e51ee6944?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend



(/p/a9dcf2ed1964?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend
G173第三次#锦囊大讲堂班级晨会纪要 (/p/a9dcf2ed1964?utm_campaign=...

【初阶线上】第三次会议锦囊 会议目的：学习用OmniFocus进行排程，觉察践行易效能生活带给自己的喜悦与收获。 班级主 教 练：王怀亮教练 轮值教练：王艺璇教练，李鑫教练 班 长：万 璐 主 持 人：袁怡秀 ...



胡豆儿 (/u/ba9c68f287a9?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

(/p/113581d5f440?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend
“中国国际债券投资论坛2017”在港举行 (/p/113581d5f440?utm_campaign=...

2017年10月26日—27日，由通达金融研究院主办，上海浦东金融学会联合主办，Tradeweb、Intex、Framesoft大力支持的“中国国际债券投资论坛2017”在香港荃湾悦来酒店成功举行。活动两天围绕探讨中国...



中国生活消费 (/u/a4ba04e4ce4d?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

秋月情思 (/p/6a2a49acd3e3?utm_campaign=maleskine&utm_content=...

有句古诗，说 秦时明月汉时关 有首老歌，唱 明月千里寄相思 诗的意韵，歌的情怀， 让这思乡之情 渐浓，渐浓 嫦娥奔月给儿时的梦想插上了翅膀， 飞啊飞，飞离了家乡 长大了，现实了 人类的一大步却将儿时的梦...



锦心明道 (/u/c177f2406e9a?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

