mysql innodb 锁

作者：moloee
原文链接：https://ld246.com/article/1508749340708
来源网站：链滴

<h3 id="排他锁和共享锁">排他锁和共享锁</h3>
MySQL 锁机制分为表级锁和行级锁

共享锁又称为读锁，简称 S 锁，共享锁就是多个事务对于同一数据可以共享一把锁，都能访问到据，但是只能读不能修改。
排他锁又称为写锁，简称 X 锁，排他锁就是不能与其他所并存，如一个事务获取了一个数据行的他锁，其他事务就不能再获取该行的其他锁

<h4 id="申请排他锁的前提">申请排他锁的前提</h4>
没有线程对该结果集中的任何行数据使用排他锁或共享锁，否则申请会阻塞。
<h4 id="申请排他锁的方式">申请排他锁的方式</h4>

使用 <code>update/delete/insert</code> 修改数据的时候
使用 <code>select * from xxx for update</code>，for update 仅适用于 InnoDB，且必须事务块(BEGIN/COMMIT)中才能生效。在进行事务操作时，通过“for update”语句，MySQL 会对询结果集中每行数据都添加排他锁，其他线程对该记录的更新与删除操作都会阻塞。排他锁包含行锁表锁。

<h3 id="悲观锁和乐观锁">悲观锁和乐观锁</h3>

悲观锁：假定会发生并发冲突，屏蔽一切可能违反数据完整性的操作。
乐观锁：假设不会发生并发冲突，只在提交操作时检查是否违反数据完整性。

悲观锁（Pessimistic Lock），顾名思义，就是很悲观，每次去拿数据的时候都认为别人会修改所以每次在拿数据的时候都会上锁，这样别人想拿这个数据就会 block 直到它拿到锁。
乐观锁（Optimistic Lock），顾名思义，就是很乐观，每次去拿数据的时候都认为别人不会修改，以不会上锁，但是在提交更新的时候会判断一下在此期间别人有没有去更新这个数据。乐观锁适用于多写少的应用场景，这样可以提高吞吐量。
乐观锁的两种实现方式，两种机制的远离基本一致:

数据版本（Version）记录机制
时间戳（timestamp）

示例：
<blockquote>
假设有 A、B 两个用户同时各购买一件 id=1 的商品，用户 A 获取到的库存量为 1000，用户 B 取到的库存量也为 1000，用户 A 完成购买后修改该商品的库存量为 999，用户 B 完成购买后修改该品的库存量为 999，此时库存量数据产生了不一致。
</blockquote>
悲观锁方案：每次获取商品时，对该商品加排他锁
<pre><code class="highlight-chroma">begin;
select * from goos where id = 1 for update;
update goods set tock = stock - 1 where id = 1;
commit;
</code></pre>
乐观锁方案：每次获取商品时，不对该商品加锁。在更新数据的时候需要比较程序中库存量与数据库中的库存量是否相等
<pre><code class="highlight-chroma">#不加锁获取 id=1 的商品对象
select * from goos where id = 1
</code></pre>

begin;
#更新 stock 值，里需要注意 where 条件 “stock = cur_stock”，只有程序中获取到的库存量与数据库中的库存量相才执行更新
update goods set tock = stock - 1 where id = 1 and stock = cur_stock;
commit;
</code></pre>
<h2 id="总结">总结</h2>
1、InnoDB 行锁是通过给索引上的索引项加锁来实现的，只有通过索引条件检索数据，InnoDB 才使用行级锁，否则，InnoDB 将使用表锁。
2、由于 MySQL 的行锁是针对索引加的锁，不是针对记录加的锁，所以虽然是访问不同行的记录，是如果是使用相同的索引键，是会出现锁冲突的。应用设计的时候要注意这一点。
3、当表有多个索引的时候，不同的事务可以使用不同的索引锁定不同的行，另外，不论是使用主键引、唯一索引或普通索引，InnoDB 都会使用行锁来对数据加锁。
4、即便在条件中使用了索引字段，但是否使用索引来检索数据是由 MySQL 通过判断不同执行计划代价来决定的，如果 MySQL 认为全表扫描效率更高，比如对一些很小的表，它就不会使用索引，这情况下 InnoDB 将使用表锁，而不是行锁。因此，在分析锁冲突时，别忘了检查 SQL 的执行计划，确认是否真正使用了索引。
5、检索值的数据类型与索引字段不同，虽然 MySQL 能够进行数据类型转换，但却不会使用索引，而导致 InnoDB 使用表锁。通过用 explain 检查两条 SQL 的执行计划，我们可以清楚地看到了这一。
6、乐观锁适合读多写少的应用场景，这样可以提高吞吐量。