

Mysql悲观锁和乐观锁

参考<https://bestzuo.cn/posts/1344475084.html>

一、基本介绍

1、定义

1) 悲观锁

总是假设最坏的情况，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阻塞直到它拿到锁（共享资源每次只给一个线程使用，其它线程阻塞，用完后再把资源转让给其它线程）。传统的关系型数据库里边就用到了很多这种锁机制，比如行锁，表锁等，读锁，写锁等，都是在做操作之前先上锁。Java中synchronized和ReentrantLock等独占锁就是悲观锁思想的实现。

2) 乐观锁

总是假设最好的情况，每次去拿数据的时候都认为别人不会修改，所以不会上锁，但是在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，可以使用版本号机制和CAS算法实现。乐观锁适用于多读的应用类型，这样可以提高吞吐量，像数据库提供的类似于write_condition机制，其实都是提供的乐观锁。在Java中java.util.concurrent.atomic包下面的原子变量类就是使用了乐观锁的一种实现方式CAS实现的。

2、使用场景

乐观锁适用于写比较少的情况下（多读场景），即冲突真的很少发生的时候，这样可以省去了锁的开销，加大了系统的整个吞吐量。

但如果是多写的情况，一般会经常产生冲突，这就会导致上层应用会不断的进行retry，这样反倒是降低了性能，所以一般多写的场景下用悲观锁就比较合适。

二、Mysql的乐观锁和悲观锁

MySQL InnoDB中要使用悲观锁，我们必须关闭mysql数据库的自动提交属性，即**MySQL默认使用autocommit模式**，

1、MySQL 中悲观锁的实现

1)概念

在 MySQL 中一般指排他锁，当事务在操作数据时将这部分数据进行锁定，直到操作完毕后再解锁，解锁后才能够由其它事务操作该部分数据。

MySQL 中的悲观锁大部分情况下依赖数据库的锁机制实现，一般使用 **select...for update**对选择的数据进行加锁处理

2) 示例:

```
SELECT * FROM account WHERE name = "MAX" for update
```

这条 SQL 语句锁定了 account 表中所有符合检索条件（name=" MAX" ）的记录。本次事务提交之前（事务提交时会释放事务过程中的锁），外界无法修改这些记录

2、MySQL 中乐观锁的实现

1) 定义

乐观锁相对悲观锁而言，乐观锁假设认为数据一般情况下不会造成冲突，所以数据进行提交更新的时候，才会对数据是否冲突进行检测，如果冲突了，就返回用户错误的信息，让用户决定如何去做。一般都是使用版本号机制来实现MySQL 数据库中的乐观锁。

版本号的实现方式有两种，一个是数据版本机制，一个是时间戳机制

1) 数据版本机制

为数据增加一个版本标识，一般是通过为数据库表增加一个数字类型的 **version** 字段来实现。当读取数据时，将 **version** 字段的值一同读出，数据每更新一次，对 **version** 的值加 1。当我们提交更新时，判断数据库表对应的当前版本信息与第一次取出来的 **version** 值进行比对，如果数据库表当前版本号与第一次取出来的 **version** 值相等，则予以更新，否则认为是过期数据。

```
mysql> select * from t_goods;
```

id	status	name	version
1	1	道具	1
2	2	装备	2

```
update t_goods set status=2,version=version+1 where id=#{id} and version=#{version};
```