

# 问了几人，MySQL changebuffer 这点都没答对

原创 是Yes呀 yes的练级攻略 2023-02-28 08:20 发表于浙江

收录于合集

#mysql 5 #Java 22 #面试 41 #MySQL 17

你好，我是yes。

有位同学问了我两个关于 `change buffer` 的问题：

星期四 10:01

大佬，讲 `change buffer` 的时候我有个疑问，如果 `insert` 一条语句先写入 `change buffer` 就返回了而没有读磁盘，怎么确定主键不冲突。如果 `update` 的时候只在 `change buffer` 里面记一条而不访问磁盘，怎么确定实际修改了多少行呢

随后我又挑了几个问答的用户提问这两个问题，发现对 `change buffer` 的理解都有偏差，于是决定操作一篇。

之前我写过一篇关于 `mysql innodb` 所有 `buffer` 的内容：[总结 `mysql` 的所有 `buffer`，一网打尽就这篇了！](#)

里面有提到 `change buffer`，这篇算是之前那篇的补充吧。

如果对 `mysql innodb buffer` 不太熟悉，建议先看下我之前总结的内容，这个还是需要点前置知识的。

## 简单回顾 `change buffer`

简单来说 `change buffer` 就是一块内存，存在于 `buffer pool` 中，当需要修改数据时候，如果这个数据所在页在 `buffer pool`，那么万事大吉，直接在内存也修改。

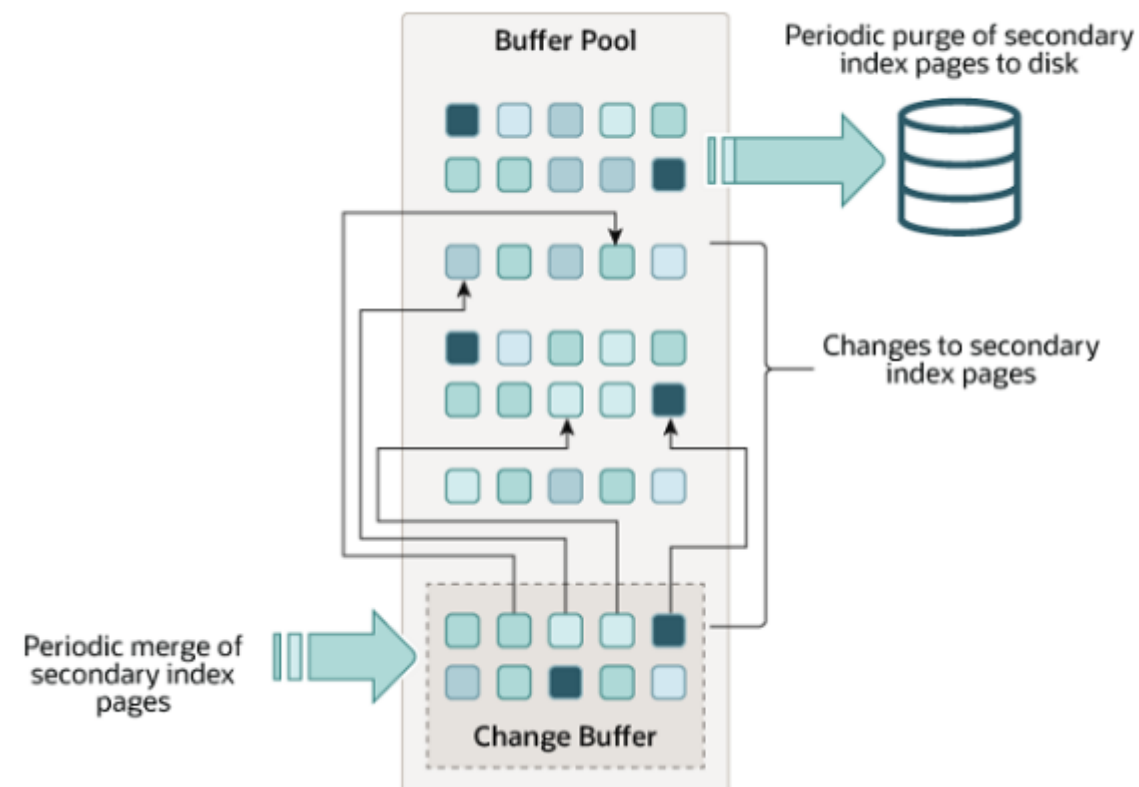
那假设 `buffer pool` 里没有要修改的那个数据页，这时候该修改怎么办呢？

假设要执行这个 sql: `update table set a = 'yes' where b = 1`

第一直觉就是找到 `b = 1` 的数据所在的数据页，加载到 `buffer pool` 中，然后进行修改，这里就涉及了从磁盘拉取数据到内存这个动作，我们都知道读盘的开销是比较大的。

所以 `mysql innodb` 搞了个 `change buffer`，当数据页不存在 `buffer pool` 时，`mysql innodb` 先把更新动作缓存在 `change buffer` 中，当下次 `select b=1` 这个数据的时候，将数据加载到 `buffer pool` 中，同时应用上 `change buffer` 的修改，这就得到正确的数据了。

Figure 14.3 Change Buffer



图来自mysql官网

重点

上述的原理不难理解，不过我之前文章提到了一点：

看到我加粗的字体没，**二级索引页**，没错 **change buffer** 只能用于二级索引的更改，不适用于主键索引，空间索引以及全文索引。

还有，**唯一索引也不行，因为唯一索引需要读取数据然后检查数据的一致性。**

这就是重点，理解了这个问题就能解答上面那位同学的提问。

**change buffer** 只能应用非唯一二级索引

聚簇索引和唯一索引都需要检查数据的一致性，这就不好先把更新、插入、删除先缓存在 **change buffer** 中。

假设现在有一张名为 **yes** 表，里面有聚簇索引以及一个 (a,b) 的非唯一二级索引，现要执行如下语句：

```
update yes set b= 'yes的练级攻略' where a = 'yesyes'
```

思考下，此时能用上 **change buffer** 吗？

原则上这条更新语句，**mysql** 可以选择聚簇索引扫描，或者应用 (a,b) 这个索引。

假设走 (a,b) 索引。

那问题来了，要走 (a,b) 索引，不就是要扫描 (a,b) 的数据吗，也就是需加载 (a,b) 索引数据页到 **buffer pool** 中，那还用啥 **change buffer**？

听起来 **change buffer** 好像没啥用啊？

我再加个条件：假设这个表不仅仅有 (a,b) 这个索引，还有 (c,b)、(d,b) 等等这些索引，那么如果我们要修改 **b** 字段的值，是不是除了聚簇索引，还需要修改 (c,b)、(d,b) 等等这些索引？

那 **change buffer** 的用处就来了！我们可以缓存这些修改，后面再应用，这就大量减少了此次 **sql** 的随机磁盘 I/O，提高了性能。

看到这，相信你对 **change buffer** 应该已经有点感觉了，你可以借着这些感觉，去看下官网的介绍，理解起来更容易：

<https://dev.mysql.com/doc/refman/5.7/en/innodb-change-buffer.html>

第一个问题：`insert` 用 `chang buffer` 如何确定主键不冲突？

主键索引和唯一索引都需要扫描校验，`change buffer` 只会缓存那些非唯一二级索引的插入操作。

第二个问题：用 `chang buffer` 不访问磁盘，如何得到实际修改行数？

`chang buffer` 无法应用在主键索引(聚簇索引)和唯一二级索引上，所以这两个得改，主键索引都改了，影响行数就有了。

好了，今天的文章就到这了，如果有收获不妨点给赞。

关于问答我还提供 1v1 个人问答服务，详细可以看这里：[我的一对一专属问答服务](#)，简单来说就是你有任何面试、技术上的问题都可以直接微信向我提问。

我是yes，从一点点到亿点点我们下篇见~



yes的练级攻略

用接地气的话来分享一些后端技术或写一些想写的。

218篇原创内容

公众号

收录于合集 #mysql 5

上一篇 · 总结 mysql 的所有 buffer，一网打尽就这篇了！

分享 收藏