



链滴

写缓冲 (change buffer), 这次彻底懂了! !!

作者: [noryar](#)

原文链接: <https://ld246.com/article/1619540075337>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<blockquote>
<p>本文摘录自架构师之路公众号! </p>
</blockquote>
<h2 id="毫无疑问-对于读请求-缓冲池能够减少磁盘IO-提升性能-问题来了-那写请求呢-">毫无疑问对于读请求, 缓冲池能够减少磁盘 IO, 提升性能。问题来了, 那写请求呢? </h2>
<h2 id="情况一">情况一</h2>
<p>假如要修改页号为 4 的索引页, 而这个页正好在缓冲池内。</p>
<p></p>
<p>如上图序号 1-2: </p>
<p>(1) 直接修改缓冲池中的页, 一次内存操作; </p>
<p>(2) 写入 redo log, 一次磁盘顺序写操作; </p>
<p>这样的效率是最高的。</p>
<p>画外音: 像写日志这种顺序写, 每秒几万次没问题。</p>
<p>是否会出现一致性问题呢? </p>
<p>并不会。</p>
<p>(1) 读取, 会命中缓冲池的页; </p>
<p>(2) 缓冲池 LRU 数据淘汰, 会将“脏页”刷回磁盘; </p>
<p>(3) 数据库异常奔溃, 能够从 redo log 中恢复数据; </p>
<p>什么时候缓冲池中的页, 会刷到磁盘上呢? </p>
<p>定期刷磁盘, 而不是每次刷磁盘, 能够降低磁盘 IO, 提升 MySQL 的性能。</p>
<p>画外音: 批量写, 是常见的优化手段。</p>
<h2 id="情况二">情况二</h2>
<p>假如要修改页号为 40 的索引页, 而这个页正好不在缓冲池内。</p>
<p></p>
<p>此时麻烦一点, 如上图需要 1-3: </p>
<p>(1) 先把需要为 40 的索引页, 从磁盘加载到缓冲池, 一次磁盘随机读操作; </p>
<p>(2) 修改缓冲池中的页, 一次内存操作; </p>
<p>(3) 写入 redo log, 一次磁盘顺序写操作; </p>
<p>没有命中缓冲池的时候, 至少产生一次磁盘 IO, 对于写多读少的业务场景, 是否还有优化的空呢? </p>
<p>这即是 InnoDB 考虑的问题, 又是本文将要讨论的写缓冲(change buffer)。</p>
<p>画外音: 从名字容易看出, 写缓冲是降低磁盘 IO, 提升数据库写性能的一种机制。</p>
<h2 id="什么是InnoDB的写缓冲-">什么是 InnoDB 的写缓冲? </h2>
<p>在 MySQL5.5 之前, 叫插入缓冲(insert buffer), 只针对 insert 做了优化; 现在对 delete 和 update 也有效, 叫做写缓冲(change buffer)。</p>
<p>它是一种应用在非唯一普通索引页(non-unique secondary index page)不在缓冲池中, 对页进行了写操作, 并不会立刻将磁盘页加载到缓冲池, 而仅仅记录缓冲变更(buffer changes), 等未来数据读取时, 再将数据合并(merge)恢复到缓冲池中的技术。写缓冲的目的是降低写操作的磁盘 IO, 提升数据库性能。</p>
<p>画外音: R 了狗了, 这个句子, 好长。</p>
<h2 id="InnoDB加入写缓冲优化-上文-情况二-流程会有什么变化-">InnoDB 加入写缓冲优化, 上“情况二”流程会有什么变化? </h2>
<p>假如要修改页号为 40 的索引页, 而这个页正好不在缓冲池内。</p>
<p></p>
<p>加入写缓冲优化后, 流程优化为: </p>
<p>(1) 在写缓冲中记录这个操作, 一次内存操作; </p>
<p>(2) 写入 redo log, 一次磁盘顺序写操作; </p>
<p>其性能与, 这个索引页在缓冲池中, 相近。</p>

<p>画外音：可以看到，40 这一页，并没有加载到缓冲池中。</p>

<p>是否会出现一致性问题呢？</p>

<p>也不会。</p>

<p>（1）数据库异常奔溃，能够从 redo log 中恢复数据；</p>

<p>（2）写缓冲不只是一个内存结构，它也会被定期刷盘到写缓冲系统表空间；</p>

<p>（3）数据读取时，有另外的流程，将数据合并到缓冲池；</p>

<p>不妨设，稍后的一个时间，有请求查询索引页 40 的数据。</p>

<p></p>

<p>此时的流程如序号 1-3：</p>

<p>（1）载入索引页，缓冲池未命中，这次磁盘 IO 不可避免；</p>

<p>（2）从写缓冲读取相关信息；</p>

<p>（3）恢复索引页，放到缓冲池 LRU 里；</p>

<p>画外音：可以看到，40 这一页，在真正被读取时，才会被加载到缓冲池中。</p>

<h2 id="还有一个遗漏问题-为什么写缓冲优化-仅适用于非唯一普通索引页呢-">还有一个遗漏问题为什么写缓冲优化，仅适用于非唯一普通索引页呢？</h2>

<p>InnoDB 里，聚集索引(clustered index)和普通索引(secondary index)的异同，《1 分钟了解 MyISAM 与 InnoDB 的索引差异》有详尽的叙述，不再展开。</p>

<p>如果索引设置了唯一(unique)属性，在进行修改操作时，InnoDB 必须进行唯一性检查。也就是，索引页即使不在缓冲池，磁盘上的页读取无法避免(否则怎么校验是否唯一?)，此时就应该直接把应的页放入缓冲池再进行修改，而不应该再整写缓冲这个么蛾子。</p>

<h2 id="除了数据页被访问-还有哪些场景会触发刷写缓冲中的数据呢-">除了数据页被访问，还有哪些场景会触发刷写缓冲中的数据呢？</h2>

<p>还有这么几种情况，会刷写缓冲中的数据：</p>

<p>（1）有一个后台线程，会认为数据库空闲时；</p>

<p>（2）数据库缓冲池不够用时；</p>

<p>（3）数据库正常关闭时；</p>

<p>（4）redo log 写满时；</p>

<p>画外音：几乎不会出现 redo log 写满，此时整个数据库处于无法写入的不可用状态。</p>

<h2 id="什么业务场景-适合开启InnoDB的写缓冲机制-">什么业务场景，适合开启 InnoDB 的写缓冲机制？</h2>

<p>先说什么时候不适合，如上文分析，当：</p>

<p>（1）数据库都是唯一索引；</p>

<p>（2）或者，写入一个数据后，会立刻读取它；</p>

<p>这两类场景，在写操作进行时(进行后)，本来就要进行进行页读取，本来相应页面就要入缓冲，此时写缓存反倒成了负担，增加了复杂度。</p>

<p>什么时候适合使用写缓冲，如果：</p>

<p>（1）数据库大部分是非唯一索引；</p>

<p>（2）业务是写多读少，或者不是写后立刻读取；</p>

<p>可以使用写缓冲，将原本每次写入都需要进行磁盘 IO 的 SQL，优化定期批量写磁盘。</p>

<p>画外音：例如，账单流水业务。</p>

<h2 id="上述原理-对应InnoDB里哪些参数-">上述原理，对应 InnoDB 里哪些参数？</h2>

<p>有两个比较重要的参数。</p>

<p></p>

<p>参数：innodb_change_buffer_max_size</p>

<p>介绍：配置写缓冲的大小，占整个缓冲池的比例，默认值是 25%，最大值是 50%。</p>

<p>画外音：写多读少的业务，才需要调大这个值，读多写少的业务，25% 其实也多了。</p>

<p>参数: innodb_change_buffering</p>

<p>介绍: 配置哪些写操作启用写缓冲, 可以设置成 all/none/inserts/deletes 等。 </p>

<p>希望大家有收获, 思路比结论重要。 </p>