12 | 为什么我的MySQL会“抖”一下？

2018-12-10 林晓斌

平时的工作中，不知道你有没有遇到过这样的场景，一条SQL语句，正常执行的时候特别快，但是有时也不知道怎么回事，它就会变得特别慢，并且这样的场景很难复现，它不只随机，而且持续时间还很短。

看上去，这就像是数据库“抖”了一下。今天，我们就一起来看一看这是什么原因。

你的SQL语句为什么变“慢”了

在前面第2篇文章《日志系统：一条SQL更新语句是如何执行的？》中，我为你介绍了WAL机制。现在你知道了，InnoDB在处理更新语句的时候，只做了写日志这一个磁盘操作。这个日志叫作redo log（重做日志），也就是《孔乙己》里咸亨酒店掌柜用来记账的粉板，在更新内存写完redo log后，就返回给客户端，本次更新成功。

做下类比的话，掌柜记账的账本是数据文件，记账用的粉板是日志文件（redo log），掌柜的记忆就是内存。

掌柜总要找时间把账本更新一下，这对应的就是把内存里的数据写入磁盘的过程，术语就是flush。在这个flush操作执行之前，孔乙己的赊账总额，其实跟掌柜手中账本里面的记录是不一致的。因为孔乙己今天的赊账金额还只在粉板上，而账本里的记录是老的，还没把今天的赊账算进去。

当内存数据页跟磁盘数据页内容不一致的时候，我们称这个内存页为“脏页”。内存数据写

入到磁盘后，内存和磁盘上的数据页的内容就一致了，称为“干净页”。

不论是脏页还是干净页，都在内存中。在这个例子里，内存对应的就是掌柜的记忆。

接下来，我们用一个示意图来展示一下“孔乙己赊账”的整个操作过程。假设原来孔乙己欠账10文，这次又要赊9文。

图1 “孔乙己赊账”更新和flush过程

回到文章开头的问题，你不难想象，平时执行很快的更新操作，其实就是在写内存和日志，而MySQL偶尔“抖”一下的那个瞬间，可能就是在刷脏页（flush）。

那么，什么情况会引发数据库的flush过程呢？

我们还是继续用咸亨酒店掌柜的这个例子，想一想：掌柜在什么情况下会把粉板上的赊账记录改到账本上？

- 第一种场景是，粉板满了，记不下了。这时候如果再有人来赊账，掌柜就只得放下手里的活儿，将粉板上的记录擦掉一些，留出空位以便继续记账。当然在擦掉之前，他必须先将正确的账目记录到账本中才行。
  这个场景，对应的就是InnoDB的redo log写满了。这时候系统会停止所有更新操作，把checkpoint往前推进，redo log留出空间可以继续写。我在第二讲画了一个redo log的示意图，这里我改成环形，便于大家理解。

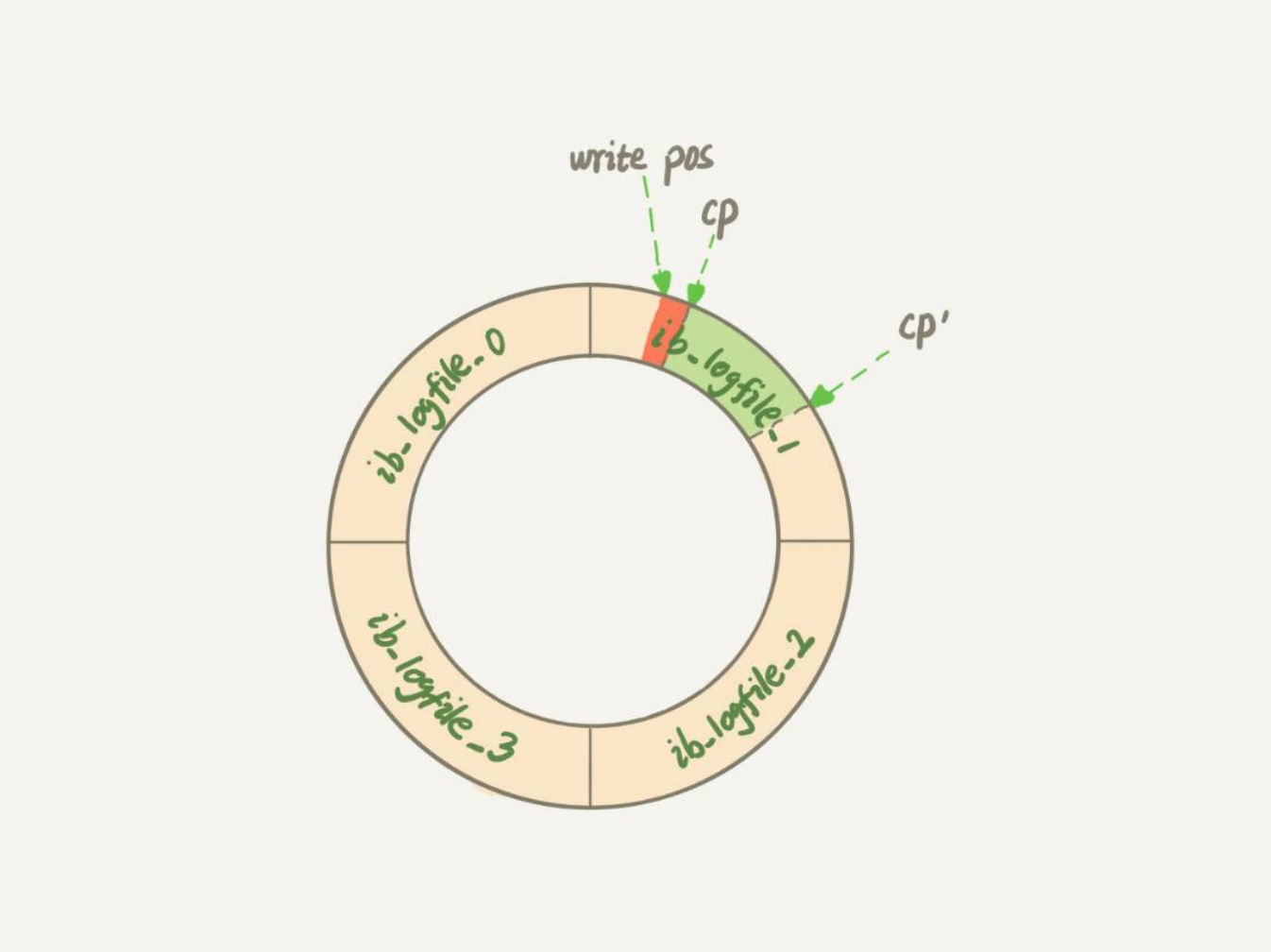

图2 redo log状态图

checkpoint可不是随便往前修改一下位置就可以的。比如图2中，把checkpoint位置从CP推进到CP'，就需要将两个点之间的日志（浅绿色部分），对应的所有脏页都flush到磁盘上。之后，图中从write pos到CP'之间就是可以再写入的redo log的区域。

- 第二种场景是，这一天生意太好，要记住的事情太多，掌柜发现自己快记不住了，赶紧找出账本把孔乙己这笔账先加进去。
  这种场景，对应的就是系统内存不足。当需要新的内存页，而内存不够用的时候，就要淘汰

一些数据页，空出内存给别的数据页使用。如果淘汰的是“脏页”，就要先将脏页写到磁盘。
你一定会说，这时候难道不能直接把内存淘汰掉，下次需要请求的时候，从磁盘读入数据页，然后拿redo log出来应用不就行了？这里其实是从性能考虑的。如果刷脏页一定会写盘，就保证了每个数据页有两种状态：

  - 一种是内存里存在，内存里就肯定是正确的结果，直接返回；
  - 另一种是内存里没有数据，就可以肯定数据文件上是正确的结果，读入内存后返回。这样的效率最高。

- 第三种场景是，生意不忙的时候，或者打烊之后。这时候柜台没事，掌柜闲着也是闲着，不如更新账本。
  这种场景，对应的就是MySQL认为系统“空闲”的时候。当然，MySQL“这家酒店”的生意好起来可是会很快就能把粉板记满的，所以“掌柜”要合理地安排时间，即使是“生意好”的时候，也要见缝插针地找时间，只要有机会就刷一点“脏页”。

- 第四种场景是，年底了咸亨酒店要关门几天，需要把账结清一下。这时候掌柜要把所有账都记到账本上，这样过完年重新开张的时候，就能就着账本明确账目情况了。
  这种场景，对应的就是MySQL正常关闭的情况。这时候，MySQL会把内存的脏页都flush到磁盘上，这样下次MySQL启动的时候，就可以直接从磁盘上读数据，启动速度会很快。

接下来，**你可以分析一下上面四种场景对性能的影响。**

其中，第三种情况是属于MySQL空闲时的操作，这时系统没什么压力，而第四种场景是数据库本来就要关闭了。这两种情况下，你不会太关注“性能”问题。所以这里，我们主要来分析一下前两种场景下的性能问题。

第一种是“redo log写满了，要flush脏页”，这种情况是InnoDB要尽量避免的。因为出现这种情况的时候，整个系统就不能再接受更新了，所有的更新都必须堵住。如果你从监控上看，这时候更新数会跌为0。

第二种是“内存不够用了，要先将脏页写到磁盘”，这种情况其实是常态。**InnoDB用缓冲池（buffer pool）管理内存，缓冲池中的内存页有三种状态：**

- 第一种是，还没有使用的；
- 第二种是，使用了并且是干净页；
- 第三种是，使用了并且是脏页。

InnoDB的策略是尽量使用内存，因此对于一个长时间运行的库来说，未被使用的页面很少。

而当要读入的数据页没有在内存的时候，就必须到缓冲池中申请一个数据页。这时候只能把最久不使用的数据页从内存中淘汰掉：如果要淘汰的是一个干净页，就直接释放出来复用；但如果是

脏页呢，就必须将脏页先刷到磁盘，变成干净页后才能复用。

所以，刷脏页虽然是常态，但是出现以下这两种情况，都是会明显影响性能的：

1. 一个查询要淘汰的脏页个数太多，会导致查询的响应时间明显变长；
2. 日志写满，更新全部堵住，写性能跌为0，这种情况对敏感业务来说，是不能接受的。

所以，InnoDB需要有控制脏页比例的机制，来尽量避免上面的这两种情况。

InnoDB刷脏页的控制策略

接下来，我就来和你说说InnoDB脏页的控制策略，以及和这些策略相关的参数。

首先，你要正确地告诉InnoDB所在主机的IO能力，这样InnoDB才能知道需要全力刷脏页的时候，可以刷多快。

这就要用到innodb_io_capacity这个参数了，它会告诉InnoDB你的磁盘能力。这个值我建议你设置成磁盘的IOPS。磁盘的IOPS可以通过fio这个工具来测试，下面的语句是我用来测试磁盘随机读写的命令：

```
fio -filename=$filename -direct=1 -iodepth 1 -thread -rw=randrw -ioengine=psync -bs=16k -size=500M -numjobs=1
```

其实，因为没能正确地设置innodb_io_capacity参数，而导致的性能问题也比比皆是。之前，就曾有其他公司的开发负责人找我看一个库的性能问题，说MySQL的写入速度很慢，TPS很低，但是数据库主机的IO压力并不大。经过一番排查，发现罪魁祸首就是这个参数的设置出了问题。

他的主机磁盘用的是SSD，但是innodb_io_capacity的值设置的是300。于是，InnoDB认为这个系统的能力就这么差，所以刷脏页刷得特别慢，甚至比脏页生成的速度还慢，这样就造成了脏页累积，影响了查询和更新性能。

虽然我们现在已经定义了“全力刷脏页”的行为，但平时总不能一直是全力刷吧？毕竟磁盘能力不能只用来刷脏页，还需要服务用户请求。所以接下来，我们就一起看看InnoDB怎么控制引擎按照“全力”的百分比来刷脏页。

根据我前面提到的知识点，试想一下，**如果你来设计策略控制刷脏页的速度，会参考哪些因素呢？**

这个问题可以这么想，如果刷太慢，会出现什么情况？首先是内存脏页太多，其次是redo log写满。

所以，InnoDB的刷盘速度就是要参考这两个因素：一个是脏页比例，一个是redo log写盘速度。

InnoDB会根据这两个因素先单独算出两个数字。

参数innodb_max_dirty_pages_pct是脏页比例上限，默认值是75%。InnoDB会根据当前的脏页比例（假设为M），算出一个范围在0到100之间的数字，计算这个数字的伪代码类似这样：

```
F1(M)
{
  if M>=innodb_max_dirty_pages_pct then
      return 100;
  return 100*M/innodb_max_dirty_pages_pct;
}
```

InnoDB每次写入的日志都有一个序号，当前写入的序号跟checkpoint对应的序号之间的差值，我们假设为N。InnoDB会根据这个N算出一个范围在0到100之间的数字，这个计算公式可以记为F2(N)。F2(N)算法比较复杂，你只要知道N越大，算出来的值越大就好了。

然后，**根据上述算得的F1(M)和F2(N)两个值，取其中较大的值记为R，之后引擎就可以按照innodb_io_capacity定义的能力乘以R%来控制刷脏页的速度。**

上述的计算流程比较抽象，不容易理解，所以我画了一个简单的流程图。图中的F1、F2就是上面我们通过脏页比例和redo log写入速度算出来的两个值。

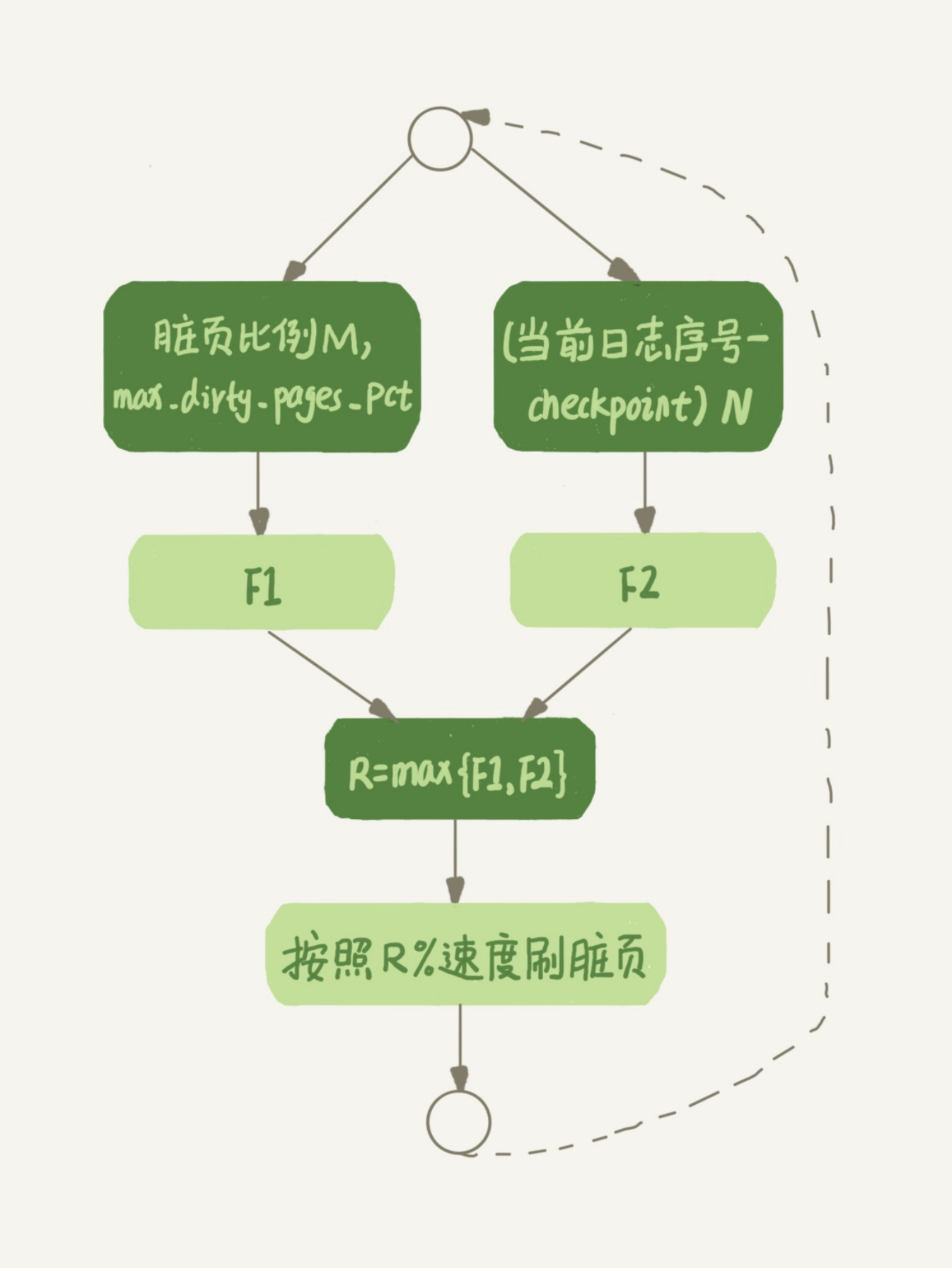

图3 InnoDB刷脏页速度策略

现在你知道了，InnoDB会在后台刷脏页，而刷脏页的过程是要将内存页写入磁盘。所以，无论是你的查询语句在需要内存的时候可能要求淘汰一个脏页，还是由于刷脏页的逻辑会占用IO资源并可能影响到了你的更新语句，都可能是造成你从业务端感知到MySQL“抖”了一下的原因。

要尽量避免这种情况，你就要合理地设置innodb_io_capacity的值，并且**平时要多关注脏页比例，不要让它经常接近75%**。

其中，脏页比例是通过Innodb_buffer_pool_pages_dirty/Innodb_buffer_pool_pages_total得到的，具体的命令参考下面的代码：

```
mysql> select VARIABLE_VALUE into @a from global_status where VARIABLE_NAME = 'Innodb_buffer_pool_pa
select VARIABLE_VALUE into @b from global_status where VARIABLE_NAME = 'Innodb_buffer_pool_pages_tota
select @a/@b;
```

接下来，我们再看一个有趣的策略。

一旦一个查询请求需要在执行过程中先flush掉一个脏页时，这个查询就可能要比平时慢了。而MySQL中的一个机制，可能让你的查询会更慢：在准备刷一个脏页的时候，如果这个数据页旁边的数据页刚好是脏页，就会把这个“邻居”也带着一起刷掉；而且这个把“邻居”拖下水的逻辑还可以继续蔓延，也就是对于每个邻居数据页，如果跟它相邻的数据页也还是脏页的话，也会被放到一起刷。

在InnoDB中，innodb_flush_neighbors 参数就是用来控制这个行为的，值为1的时候会有上述的“连坐”机制，值为0时表示不找邻居，自己刷自己的。

找“邻居”这个优化在机械硬盘时代是很有意义的，可以减少很多随机IO。机械硬盘的随机IOPS一般只有几百，相同的逻辑操作减少随机IO就意味着系统性能的大幅度提升。

而如果使用的是SSD这类IOPS比较高的设备的话，我就建议你把innodb_flush_neighbors的值设置成0。因为这时候IOPS往往不是瓶颈，而“只刷自己”，就能更快地执行完必要的刷脏页操作，减少SQL语句响应时间。

在MySQL 8.0中，innodb_flush_neighbors参数的默认值已经是0了。

小结

今天这篇文章，我延续第2篇中介绍的WAL的概念，和你解释了这个机制后续需要的刷脏页操作和执行时机。利用WAL技术，数据库将随机写转换成了顺序写，大大提升了数据库的性能。

但是，由此也带来了内存脏页的问题。脏页会被后台线程自动flush，也会由于数据页淘汰而触发flush，而刷脏页的过程由于会占用资源，可能会让你的更新和查询语句的响应时间长一些。在文章里，我也给你介绍了控制刷脏页的方法和对应的监控方式。

文章最后，我给你留下一个思考题吧。

一个内存配置为128GB、innodb_io_capacity设置为20000的大规格实例，正常会建议你将redo log设置成4个1GB的文件。

但如果你在配置的时候不慎将redo log设置成了1个100M的文件，会发生什么情况呢？又为什么会出现这样的情况呢？

你可以把你的分析结论写在留言区里，我会在下一篇文章的末尾和你讨论这个问题。感谢你的收听，也欢迎你把这篇文章分享给更多的朋友一起阅读。

上期问题时间

上期我留给你的问题是，给一个学号字段创建索引，有哪些方法。

由于这个学号的规则，无论是正向还是反向的前缀索引，重复度都比较高。因为维护的只是一个学校的，因此前面6位（其中，前三位是所在城市编号、第四到第六位是学校编号）其实是固定的，邮箱后缀都是@gamil.com，因此可以只存入学年份加顺序编号，它们的长度是9位。

而其实在此基础上，可以用数字类型来存这9位数字。比如201100001，这样只需要占4个字节。其实这个就是一种hash，只是它用了最简单的转换规则：字符串转数字的规则，而刚好我们设定的这个背景，可以保证这个转换后结果的唯一性。

评论区中，也有其他一些很不错的见解。

评论用户@封建的风 说，一个学校的总人数这种数据量，50年才100万学生，这个表肯定是小表。为了业务简单，直接存原来的字符串。这个答复里面包含了“优化成本和收益”的思想，我觉得值得at出来。

@小潘 同学提了另外一个极致的方向。如果碰到表数据量特别大的场景，通过这种方式的收益是很不错的。

评论区留言点赞板：

> @lttzzlll ，提到了用整型存“四位年份+五位编号”的方法；
> 由于整个学号的值超过了int上限，@老杨同志 也提到了用8个字节的bigint来存的方法。

精选留言

Tony Du 74

当内存不够用了，要将脏页写到磁盘，会有一个数据页淘汰机制（最久不使用），假设淘汰的是脏页，则此时脏页所对应的redo log的位置是随机的，当有多个不同的脏页需要刷，则对应的redo log可能在不同的位置，这样就需要把redo log的多个不同位置刷掉，这样对于redo log的处理不是就会很麻烦吗？（合并间隙，移动位置？）

另外，redo log的优势在于将磁盘随机写转换成了顺序写，如果需要将redo log的不同部分刷掉（刷脏页），不是就在redo log里随机读写了么？

2018-12-10

作者回复

好问题。

其实由于淘汰的时候，刷脏页过程不用动redo log文件的。

这个有个额外的保证，是redo log在“重放”的时候，如果一个数据页已经是刷过的，会识别出来并跳过。

2018-12-10

某、人 38

redo log是关系型数据库的核心啊,保证了ACID里的D。所以redo log是牵一发而动全身的操作
按照老师说的当内存数据页跟磁盘数据页不一致的时候,把内存页称为'脏页'。如果redo log设置得太小,redo log写满.那么会涉及到哪些操作呢,我认为是以下几点:

1.把相对应的数据页中的脏页持久化到磁盘,checkpoint往前推

2.由于redo log还记录了undo的变化,undo log buffer也要持久化进undo log
3.当innodb_flush_log_at_trx_commit设置为非1,还要把内存里的redo log持久化到磁盘上
4.redo log还记录了change buffer的改变,那么还要把change buffer purge到idb
以及merge change buffer.merge生成的数据页也是脏页,也要持久化到磁盘
上述4种操作,都是占用系统I/O,影响DML,如果操作频繁,会导致'抖'得向现在我们过冬一样。
但是对于select操作来说,查询时间相对会更快。因为系统脏页变少了,不用去淘汰脏页,直接复用干净页即可。还有就是对于宕机恢复,速度也更快,因为checkpoint很接近LSN,恢复的数据页相对较少
所以要控制刷脏的频率,频率快了,影响DML I/O,频率慢了,会导致读操作耗时长。
我是这样想的这个问题,有可能不太对,特别是对于第4点是否会merge以及purge,还需要老师的解答

2018-12-10

作者回复

抖得像过冬一样，

你说得很对，第4点没错的，出现这种情况的时候，连change buffer的优化也没意义了

2018-12-11

yesir 13

我观察了下公司的数据库确实发现了抖动现象，有几个问题，
1）Innodb_buffer_pool_pages_total这个值很大，百万级别的，而且数值不像是人为设置上去的，是怎么来的呢？
2）Innodb_buffer_pool_pages_dirty达到4万多的时候就开始flush了，脏页比例是75，这肯定是远达不到的，ssd磁盘，innodb_io_capacity是200，肯定可以提高。文章中说flush的触发条件有2个，一个是内存不够了，一个是redo log 满了，那么我这个场景是哪种情况呢

2018-12-11

作者回复

1） 这个是innodb 数据页总是，过百万是正常的，16K一个，Bufree pool size 16G 就是100万了

2）你这个例子就是io_capacity设太小了...

2018-12-12

Ryoma 8

关于粉板和redo log的类比我觉得有一点不太合适：redo log记录的是实时欠款，比如账本中是10文，又欠了9文，此时redo log 记录的是19；而粉板的话，只会追加某人欠款+9文，不会关注原来已欠款多少（不然某人赊账时，我还需要找到账本中的这个人，才知道他之前欠款多少，我觉得这个场景跟MySQL中的场景还是有区别的）

2018-12-12

作者回复

Redo log里也是记的+9哦

2018-12-12

melon 24

melon

24

又思考了一下，请老师帮忙看一下理解的对不对：buffer pool里维护着一个脏页列表，假设现在redo log 的 checkpoint 记录的 LSN 为 10，现在内存中的一干净页有修改，修改后该页的LSN为12，大于 checkpoint 的LSN，则在写redo log的同时该页也会被标记为脏页记录到脏页列表中，现在内存不足，该页需要被淘汰掉，该页会被刷到磁盘，磁盘中该页的LSN为12，该页也从脏页列表中移除，现在redo log 需要往前推进checkpoint，到LSN为12的这条log时，发现内存中的脏页列表里没有该页，且磁盘上该页的LSN也已经为12，则该页已刷脏，已为干净页，跳过。

2018-12-11

作者回复

对的。

2018-12-11

岁月安然

18

“内存不够用了，要先将脏页写到磁盘“和“redo log 写满了，要 flush 脏页”可以理解为一个脏页本身占用内存，释放内存需要将脏页写入到磁盘才能释放。而redo log写满只有当redo log对应的脏页flush到磁盘上才能释放对应空间。有几个问题：

1、“内存不够用了，要先将脏页写到磁盘“redo log对应的空间会释放嘛？“redo log 写满了，要 flush 脏页”对应的内存页会释放嘛？

2、将脏页flush到磁盘上是直接将脏页数据覆盖到对应磁盘上的数据？还是从磁盘上取到数据后取根据redo log记录进行更新后再写入到磁盘？

3、redo log是怎么记录对应脏页是否已经flush了？如果断电了重启导致内存丢失，前面几章说通过redo log进行数据恢复那redo log又怎么去释放空间？

2018-12-10

作者回复

1. Redolog 的空间是循环使用的，无所谓释放。 对应的内存页会变成干净页。但是等淘汰的时候才会逐出内存

2. 好问题，前者

3. 不用记，重启了就从checkpoint 的位置往后扫。 如果已经之前刷过盘的, 不会重复应用redo log。 好问题

2018-12-10

jimmy

16

老师，我想问一下，innodb是如何知道一个页是不是脏页的，是有标记位还是通过redolog的checkpoint来确定的？

2018-12-10

作者回复

每个数据页头部有LSN，8字节，每次修改都会变大。

对比这个LSN跟checkpoint 的LSN，比checkpoint小的一定是干净页

2018-12-10

老鱼头 8

那个说fio命令会破坏硬盘的兄弟，是没用对命令。估计把-filename=/dev/sdb1 。。。这个的意思是从 分区 sdb1 的第一个扇区开始写入随机数据，去判断这个磁盘的写入速度。如果指定路径+文件名就不会出这事了~比如老师给的例子~

2018-12-12

作者回复

嗯嗯，你说的对

写文章的时候，我还故意用变量，这样直接拷贝会出错，然后自己再写个路径，已经考虑了安全了

2018-12-12

WL 6

把该讲内容总结为几个问题, 大家复习的时候可以先尝试回答这些问题检查自己的掌握程度:

1.
脏页和干净页的定义是什么?
2.
引发数据库flush脏页的四种典型场景各是什么?对mysql性能的影响各是怎样的?
3.
缓存池中的内存页有哪三种状态? 哪两种刷脏页的情况会比较影响性能?
4.
innodb_io_capacity这个参数的作用是什么,这个参数设置错误可能导致什么样的后果, 如何正确的设置这个参数?
5.
mysql如何通过innodb_io_capacity, 脏页比例(M), 当前日志序号(N)这三个指标来控制以什么样的速度去刷脏页的?
6.
innodb_flush_neighbor这个参数表示什么意思,应该如何设置?

2018-12-16

godtrue 5

1：MySQL抖一下是什么意思？

抖我认为就是不稳定的意思，一个SQL语句平时速度都挺快的，偶尔会慢一下且没啥规律，就是抖啦！

2：MySQL为啥会抖一下？

因为运行的不正常了，或者不稳定了，需要花费更多的资源处理别的事情，会使SQL语句的执行效率明显变慢。针对innoDB导致MySQL抖的原因，主要是InnoDB 会在后台刷脏页，而刷脏页的过程是要将内存页写入磁盘。所以，无论是你的查询语句在需要内存的时候可能要求淘汰

一个脏页，还是由于刷脏页的逻辑会占用 IO 资源并可能影响到了你的更新语句，都可能是造成你从业务端感知MySQL“抖”了一下的原因。

3：MySQL抖一下有啥问题？

很明显系统不稳定，性能突然下降对业务端是很不友好的。

4：怎么让MySQL不抖？

设置合理参数配配置，尤其是设置 好innodb_io_capacity 的值，并且平时要多关注脏页比例，不要让它经常接近 75%

5：啥是脏页？

当内存数据页跟磁盘数据页内容不一致的时候，我们称这个内存页为“脏页”。

按照这个定义感觉脏页是不可避免的，写的时候总会先写内存再写磁盘和有没有用WAL没啥关系？

6：啥是干净页？

内存数据写入到磁盘后，内存和磁盘上的数据页的内容就一致了，称为“干净页”。

7：脏页是咋产生的？

因为使用了WAL技术，这个技术会把数据库的随机写转化为顺序写，但副作用就是会产生脏页。

8：啥是随机写？为啥那么耗性能？

随机写我的理解是，这次写磁盘的那个扇区和上一次没啥关系，需要重新定位位置，机械运动是很慢的即使不是机械运动重新定位写磁盘的位置也是很耗时的。

9：啥是顺序写？

顺序写我的理解是，这次写磁盘那个扇区就在上一次的下一个位置，不需要重新定位写磁盘的位置速度当然会快一些。

10：WAL怎么把随机写转化为顺序写的？

写redolog是顺序写的，先写redolog等合适的时候再写磁盘，间接的将随机写变成了顺序写，性能确实会提高不少。

2019-06-30

喔～ 5

老师，请问下访问某条记录时，存储引擎是如何判断这条记录所在的数据页是否在内存当中，这个查内存机制是如何实现的？

2018-12-20

作者回复

每个页面有编号的。拿着编号去内存看，没有，就去磁盘

2018-12-20

阿建 5

redo-log内存满了，不停的要刷脏页回磁盘。现象就会是发现机器io不高，但是mysql明显的卡顿。

2018-12-10

Tony Du 4

当内存不够用了，要将脏页写到磁盘，会有一个数据页淘汰机制（最久不使用），假设淘汰的是脏页，则此时脏页所对应的redo log的位置是随机的，当有多个不同的脏页需要刷，则对应的redo log可能在不同的位置，这样就需要把redo log的多个不同位置刷掉，这样对于redo log的处理不是就会很麻烦吗？（合并间隙，移动位置？）
另外，redo log的优势在于将磁盘随机写转换成了顺序写，如果需要将redo log的不同部分刷掉（刷脏页），不是就在redo log里随机读写了么？

作者回复
好问题。

其实由于淘汰的时候，刷脏页过程不用动redo log文件的。

这个有个额外的保证，是redo log在“重放”的时候，如果一个数据页已经是刷过的，会识别出来并跳过。

我的回复
这个额外保证是如何做到的？能不能稍微解释下
通过刷脏页时数据页更新的timestamp来对比redo log的timestamp？

2018-12-11

作者回复

LSN，每次写redo log都带的一个数字， 数据页上也有，对比大小的，因为太细节没有写到文章中。

2018-12-11

董航 4

怎么说呢，越看越明白，哈哈，前面不懂的，后面就懂了

2018-12-10

作者回复

□□

你领会到实践篇的“奥义”了

一边引入新知识点，一边应用前面的

2018-12-10

greatcl 3

二刷中。

我觉得Ryoma 的疑问可能是因为《图 1 “孔乙己赊账”更新和 flush 过程》中，日志/粉板的内容写成了“10改成19”，其实应该是“+9”。而“内存/掌柜记忆”的内容是否是19要根据原有内容是否在内存中来区别。如果掌柜记忆够好或者孔乙己赊账前是否先问了下：我现在欠多少钱？老板查了下回答：10文，那么此时内存的值是会被直接改为19，否则内存中只是change buffer的记录+9。

不知这样理解是否正确

2019-04-24

Geek_477c02 3

InnoDB 认为这个系统的能力就这么差，所以刷脏页刷得特别慢，甚至比脏页生成的速度还慢，这样就造成了脏页累积，影响了查询和更新性能。为什么会影响性能，是因为要读取的数据不在内存中，但是脏页过多导致新页必须等待脏页刷盘导致的么？

2019-01-29

作者回复

理解准确

2019-01-29

lionetes 3

很多测试人员再做压力测试的时候 出现刚开始 insert update 很快 一会 就出现很慢,并且延迟很大，大部分是因为redo log 设置太小 引起的,完美诠释

2018-12-11

作者回复

常见的误用场景

2018-12-11

Ying 3

redolog 设置过小，会导致频繁刷脏页，还可能引发连坐，这样抖的频率可能会明显变高，系统会不断卡死的

2018-12-10

Lucus 2

思考题：

redo log设置的太小，应该是很容易引发redo log满了要flush，这时候所有的更新都会阻塞。内存的优势没有利用起来，如果是更新频繁的库会导致整个库都不可用。

p.s. 林老师能不能详细讲一下innodb_io_capacity这个参数与系统的配置关系，如何根据系统调整最佳值？

2019-04-01

克己过

2

老师！！！fio这条命令是会破坏硬盘的！而且百度搜出来不加硬盘坏关键词去搜，搜出来的文章没有一篇会告诉你这个事情！！！不说了，我去恢复数据了

2018-12-10

作者回复

没有吧，怎么会破坏硬盘？我和以前同事一直这么用的呀...

你确定是这个命令导致的吗

2018-12-12