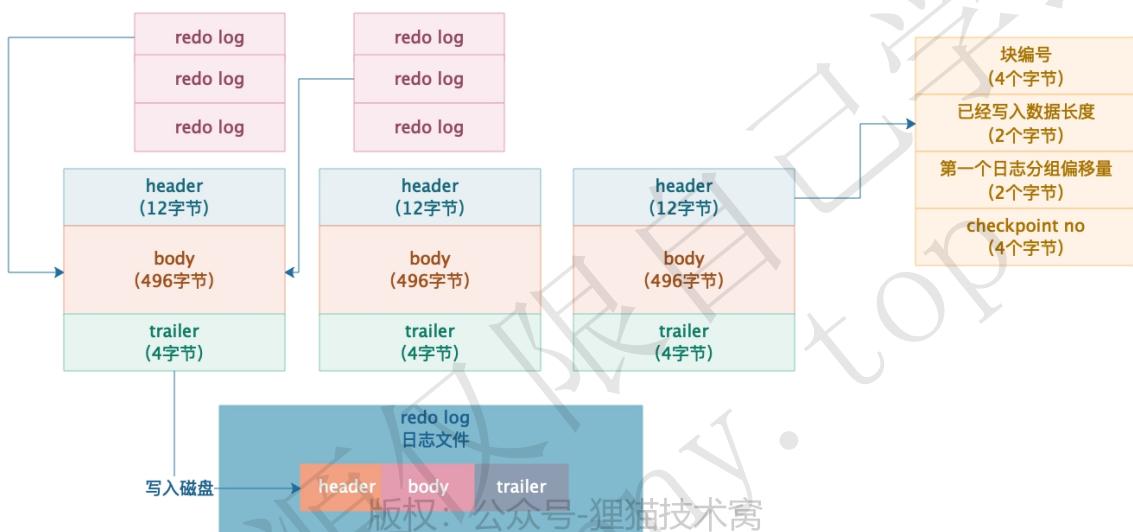


44 redo log buffer中的缓冲日志，到底什么时候可以写入磁盘？

之前我们给大家讲解了一下redo log buffer的缓冲机制，大家现在应该都知道了，redo log在写的时候，都是一个事务里的一组redo log，先暂存在一个地方，完事儿了以后把一组redo log写入redo log buffer。

写入redo log buffer的时候，是写入里面提前划分好的一个一个的redo log block的，选择有空闲空间的redo log block去写入，然后redo log block写满之后，其实会在某个时机刷入到磁盘里去，如下图。



那么今天我们就来研究一下，到底redo log buffer里的redo log block什么时候可以刷入到磁盘文件里去呢？

另外，磁盘上到底有几个redo log日志文件？不可能大量的redo log日志都放一个文件里吧？磁盘空间会占用的越来越多吗？

首先，我们先来看看redo log block是哪些时候会刷入到磁盘文件里去：

(1) 如果写入redo log buffer的日志已经占据了redo log buffer总容量的一半了，也就是超过了8MB的redo log在缓冲里了，此时就会把他们刷入到磁盘文件里去

(2) 一个事务提交的时候，必须把他的那些redo log所在的redo log block都刷入到磁盘文件里去，只有这样，当事务提交之后，他修改的数据绝对不会丢失，因为redo log里有重做日志，随时可以恢复事务做的修改

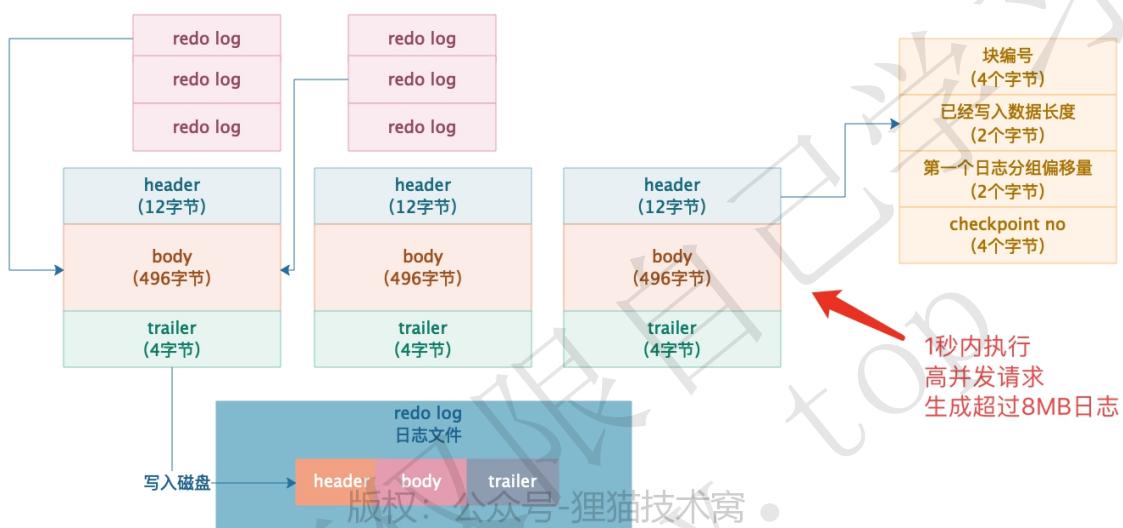
(PS：当然，之前最早最早的时候，我们讲过，这个先进入os cache的，进入os的文件缓冲区里，所以是否提交事务就强行把这个需要设置对应的参数，我们之前都讲过的

redo log哪怕事务提交的时候写入磁盘文件，也是redo log刷入物理磁盘文件，大家回过头去看看

(3) 后台线程定时刷新，有一个后台线程每隔1秒就会把redo log buffer里的redo log block刷到磁盘文件里去

(4) MySQL关闭的时候，redo log block都会刷入到磁盘里去

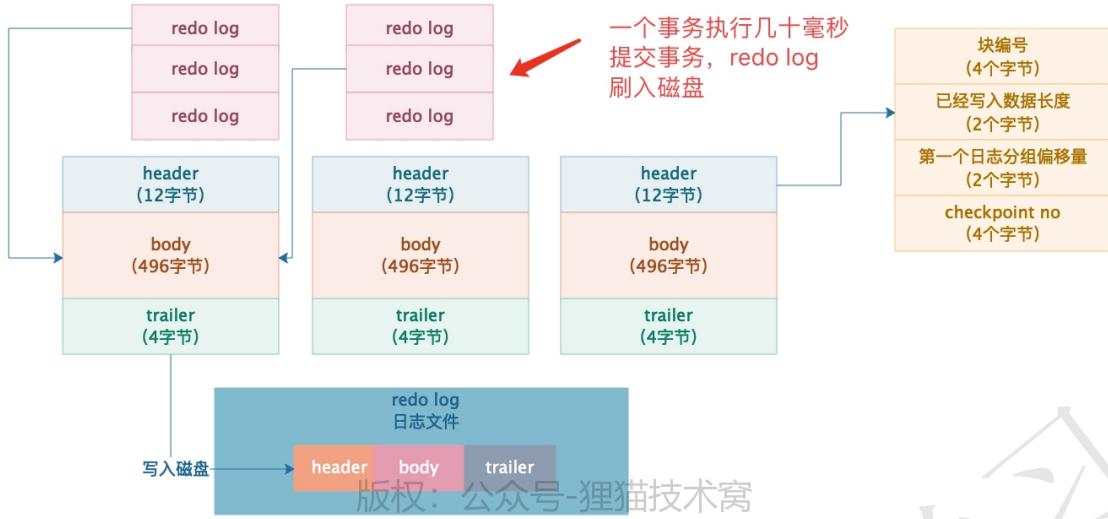
忽略上面的第四条不说，因为关闭MySQL的时候必然会刷redo log到磁盘，其他三条其实我们都看到了，也就是说，如果你瞬间执行大量的高并发的SQL语句，1秒内就产生了超过8MB的redo log，此时占据了redo log buffer一半的空间了，必然会直接把你的redo log刷入磁盘里去，如下图。



上面这种redo log刷盘的情况，在MySQL承载高并发请求的时候比较常见，比如每秒执行上万个增删改SQL语句，每个SQL产生的redo log假设有几百个字节，此时却是会在瞬间生成超过8MB的redo log日志，必然会触发立马刷新redo log到磁盘。

其次，第二种情况，其实就是平时执行一个事务，这个事务一般都是在几十毫秒到几百毫秒执行完毕的，说实在的，一般正常性能情况下，MySQL单事务性能一般不会超过1秒，否则数据库操作就太慢了。

那么如果在几十毫秒，或者几百毫秒的时候，执行完毕了一个事务，此时必然会立马把这个事务的redo log都刷入磁盘，如下图。



第一种情况其实是不常见的，第二种情况是比较常见的，往往redo log刷盘都是以一个短事务提交时候发生的，第三种情况就是后台线程每秒自动刷新redo log到磁盘去，这个就是说假设没有别的触发，后台线程自己都会不停的刷新redo log到磁盘。

但是不管怎么说，主要是保证一个事务执行的时候，redo log都进入redo log buffer，提交事务的时候，事务对应的redo log必须是刷入磁盘文件，接着才算是事务提交成功，否则事务提交就是失败，保证这一点，就能确保事务提交之后，数据不会丢，有redo log在磁盘里就行了。

当然，绝对保证数据不丢，还得配置一个参数，提交事务把redo log刷入磁盘文件的os cache之后，还得强行从os cache刷入物理磁盘。

最后给大家说一下redo log日志文件的问题，我们都知道平时不停的执行增删改，那么MySQL会不停的产生大量的redo log写入日志文件，那么日志文件就用一个写入全部的redo log？对磁盘占用空间越来越大怎么办？

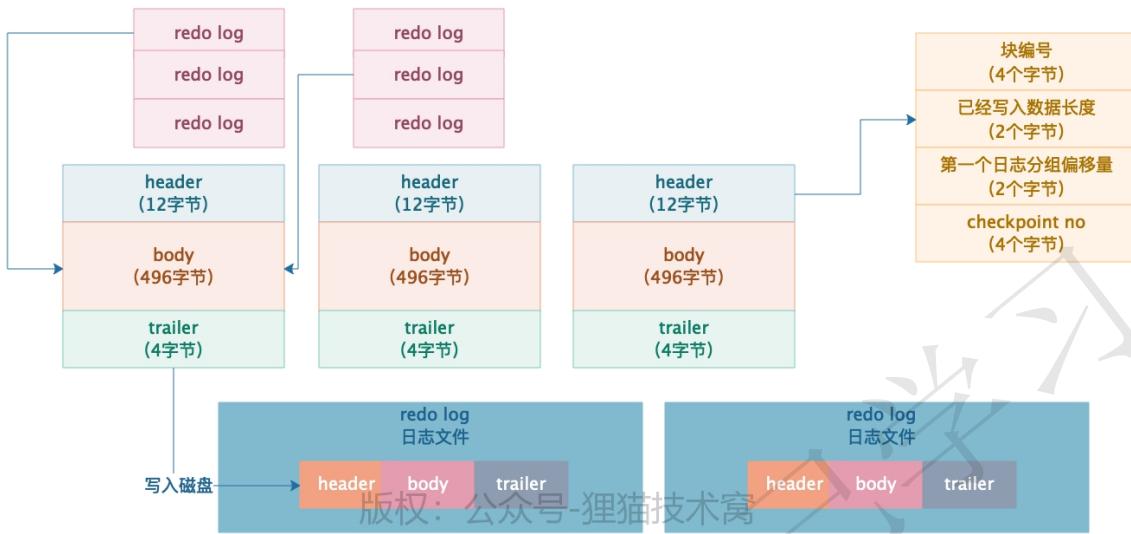
别担心，这些问题都可以解决，实际上默认情况下，redo log都会写入一个目录中的文件里，这个目录可以通过show variables like 'datadir'来查看，可以通过innodb_log_group_home_dir参数来设置这个目录的。

然后redo log是有多个的，写满了一个就会写下一个redo log，而且可以限制redo log文件的数量，通过innodb_log_file_size可以指定每个redo log文件的大小，默认是48MB，通过innodb_log_files_in_group可以指定日志文件的数量，默认就2个。

所以默认情况下，目录里就两个日志文件，分别为ib_logfile0和ib_logfile1，每个48MB，最多就这2个日志文件，就是先写第一个，写满了写第二个。那么如果第二个也写满了呢？别担心，继续写第一个，覆盖第一个日志文件里原来的redo log就可以了。

所以最多这个redo log，mysql就给你保留了最近的96MB的redo log而已，不过这其实已经很多了，毕竟redo log真的很小，一条通常就几个字节到几十个字节不等，96MB足够你存储上百万条redo log了！

如果你还想保留更多的redo log，其实调节上述两个参数就可以了，比如每个redo log文件是96MB，最多保留100个redo log文件。下面图里，给大家展示出来了多个redo log文件循环写入的示意。



我想讲到这里，大家对redo log机制就理解更加深刻了，对于事务产生的redo log如何进入内存缓冲，如何进入block，什么时候刷入磁盘，磁盘上有几个redo log日志文件，这些机制都了解的很清晰了。

End