

图文 32 一文总结初步了解到的MySQL存储模型以及数据读写机制

108 人次阅读 2020-02-28 10:06:56

详情 评论

一文总结初步了解到的MySQL存储模型以及数据读写机制

- 如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑
- 如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）

今天我们来用一篇文章初步总结一下我们近期学习到的MySQL存储模型以及对应的读写机制，其实大家通过近期的学习也仅仅是初步了解了MySQL底层数据的存储模型而已，因为后续我们还要讲解MySQL的增删改查执行背后的深入底层的各种存储数据读写细节，现在仅仅是初步把存储模型的结构给建立起来罢了。

好，那么我们现在应该都知道了，最终MySQL的数据都是放在磁盘文件里的，这个大家应该都没什么问题吧

那么数据在磁盘文件里是怎么存放的呢？我们都知道我们平时数据都是插入一个一个的表中的，而表是个逻辑概念，其实在物理层面，他对应的是表空间这个概念。

所以其实在MySQL的磁盘上，表空间就对应着磁盘文件，在磁盘文件里就存放着数据！

那么这个表空间的磁盘文件里，数据是如何组织的呢？

这个就非常的复杂了！因为你可以想象一下，假如让你把数据直接一行一行的写入一个磁盘文件，当然很简单了！

但是问题是你现在要存储的是数据库里的如此复杂的数据！他里面是有各种字段类型的，还有索引这个概念，当后面我们讲到索引的时候，就会详细分析这个索引在磁盘里的数据组织结构，这也是相当的复杂。

所以其实在磁盘文件里存放的数据，他从最基本的角度来看的话，就是被拆分为一个一个的数据区（extent）分组，以后我们干脆就用他的英文名叫做extent组好了，每个extent组中包含256个extent，然后每个extent里包含64个数据页！然后每个数据页里都包含了一行一行的数据！如下图所示。



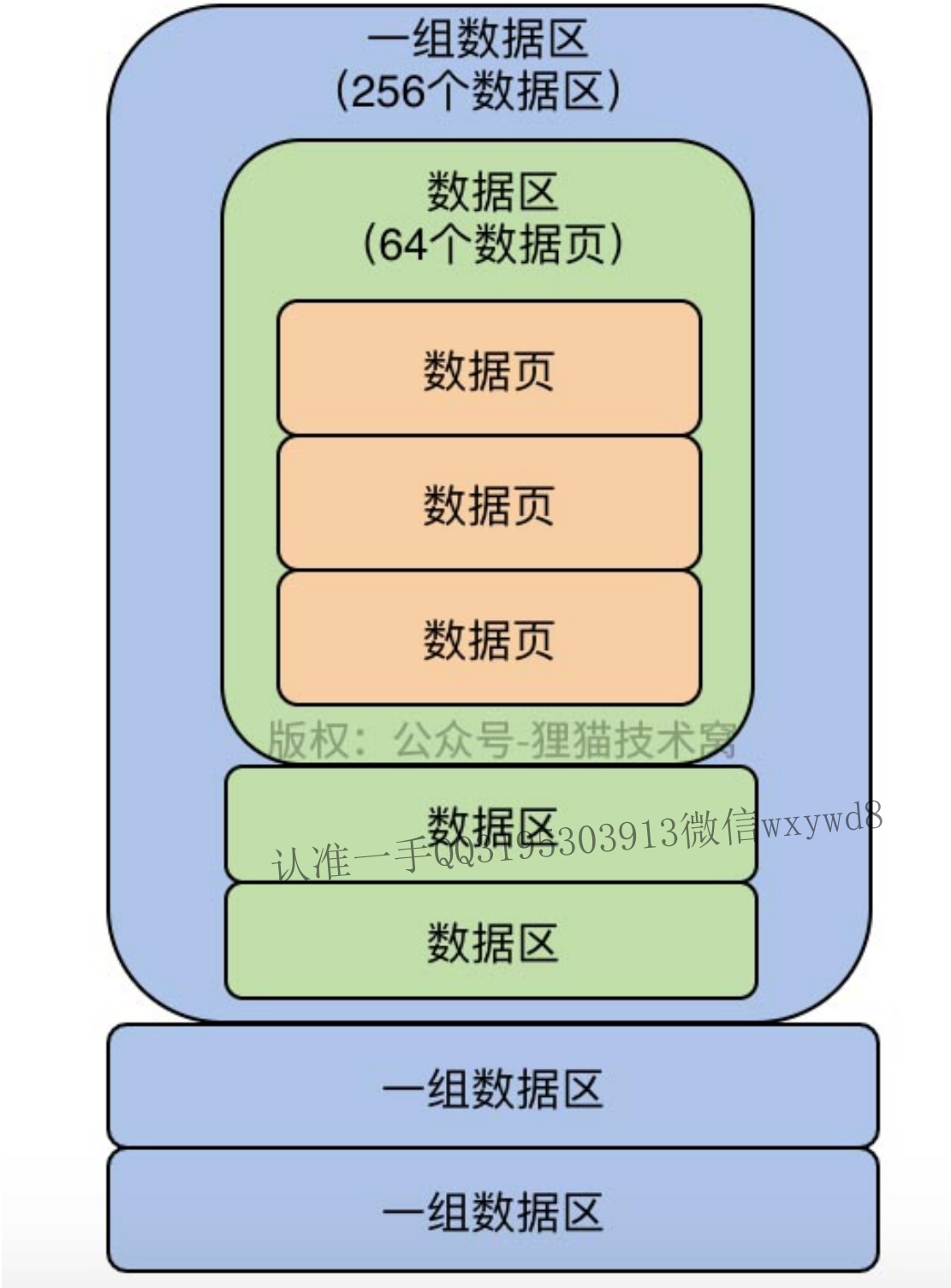
狸猫技术窝

进店逛

相关频道



从零开始
实战优化
已更新3



大家听到这里，是不是觉得特别的简单，其实绝对不是的

在实际存储的时候，我们之前稍微给大家介绍过一点点，在数据行里都有很多附加的信息，在数据页、数据区里，都有很多附加的特殊信息。各种各样的特殊信息，就可以让我们在简简单单的磁盘文件里实现B+树索引、事务之类的非常复杂的机制。

关于这些存储结构中的特殊信息是如何用于支撑实现很多高级的复杂功能的，我们后续会给大家讲解的！

那么现在问题来了，我们都知道，当我们在数据库中执行crud的时候，你必须先把磁盘文件里的一个数据页加载到内存的Buffer Pool的一个缓存页里去，然后我们增删改查都是针对缓存页里的数据来执行的！

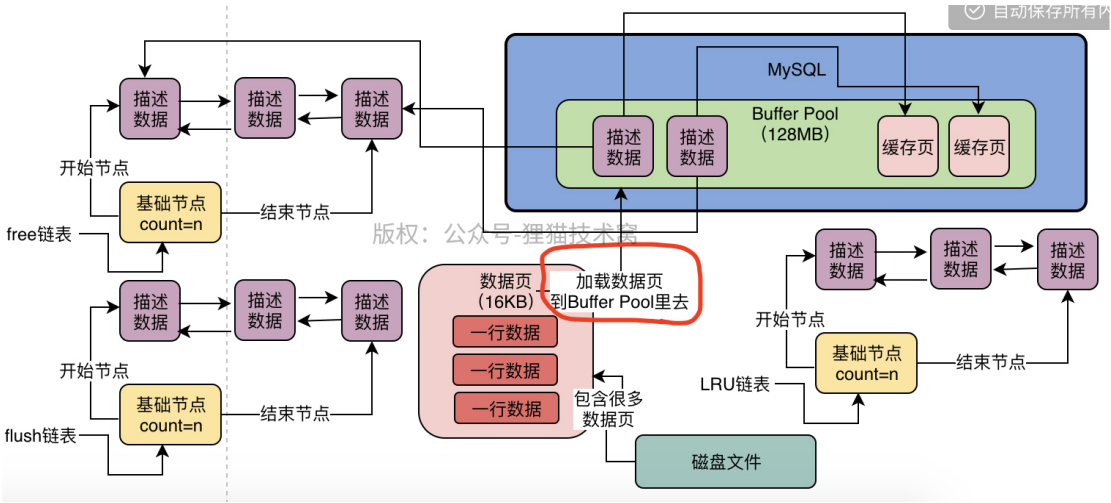
所以假设此时我们要插入一条数据，那么是选择磁盘文件里的哪个数据页加载到缓存页里去呢？

大家注意，这里要划重点了，其实这个时候会看看你往哪个表里插入数据？然后肯定得根据表找到一个表空间啊！

找到表空间之后，就可以定位到对应的磁盘文件啊！有了磁盘文件之后，就可以从里面找一个extent组，找一个extent，接着从里面找一个数据页出来！这个数据也可能是空的，也可能已经放了一些数据行了！

然后就可以把这个数据页从磁盘里完整加载出来，放入Buffer Pool的缓存页里了！

我们看下图的示意



当然，这个时候有人会问了，这个从磁盘文件里读取一个数据页，是怎么读取的啊？

其实这个很简单了，你可以想一下，磁盘文件里放的数据都是紧挨在一起的，类似于下面的那种样子。

0xdf3439399abc0sfdkslf9sdfpsfds0xdf3439399abc0sfdkslf9sdfpsfds

0xdf3439399abc0sfdkslf9sdfpsfds0xdf3439399abc0sfdkslf9sdfpsfds

其实上述字符完全无任何意义，就是我为演示随便搞出来的一段东西而已，但是大致来说磁盘里存放的数据看起来就是那样的，可能先是有个extent组开始的一些东西，然后里面是一个一个的extent，每个extent开始的时候会写一些特殊的信息，然后再是一个一个的数据页，里面是一个一个的数据行。

那么在读取一个数据页的时候，你可以通过随机读写的方式来了，举个例子，我们下面有一个伪代码，大家看看。就是设置一下要从一个数据文件的哪个位置开始读取，一直到哪个位置就结束。

```
dataFile.setStartPosition(25347)
dataFile.setEndPosition(28890)
dataPage = dataFile.read()
```

通过上面伪代码那种方式，你指定磁盘文件里的开始和截止的位置，就能读取出来指定位置的一段数据，比如读取出来一大坨东西：psfds0xdf343939。也许这坨东西就是一个数据页包含的内容了。

然后把数据页放到内存的缓存页里即可。

接着crud操作都可以直接针对缓存页去执行了，会自动把更新的缓存页加入flush链表，然后更新他在lru链表里的位置，包括更新过的缓存页会从free链表里拿出来，等等，后续一系列操作，都是之前我们分析过的了。

此时对于那些被更新过的缓存页来说，都会由后台线程刷入磁盘的，那么刷磁盘的时候是怎么刷呢？我们也是写一段伪代码给大家看看。

```
dataFile.setStartPosition(25347)
dataFile.setEndPosition(28890)
dataFile.write(cachePage)
```

因为一个数据页的大小其实是固定的，所以一个数据页固定就是可能在一个磁盘文件里占据了某个开始位置到结束位置的一段数据，此时你写回去的时候也是一样的，选择好固定的一段位置的数据，直接把缓存页的数据写回去，就覆盖掉了原来的那个数据页了，就如上面的伪代码示意。

今天通过一篇总结文章，就给大家讲清楚了我们初步了解到的存储模型是如何跟Buffer Pool缓存机制配合起来实现crud的，接着我们会给大家讲解几个数据库优化的案例了！

我们的风格就是讲一些理论的知识，就配合一些实战案例，让大家理论和实战相结合起来。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为消息中间件实战高手》](#)
[《21天互联网Java进阶面试训练营》（分布式篇）](#)
[《互联网Java工程师面试突击》（第1季）](#)
[《互联网Java工程师面试突击》（第3季）](#)
[《从零开始带你成为JVM实战高手》](#)

认准一手QQ3195303913微信wxywd8