# 概述

Buffer Pool 是什么？从字面上看是缓存池的意思，没错，它其实也就是缓存池的意思。它是 MySQL 当中至关重要的一个组件，可以这么说，MySQL的所有的增删改的操作都是在 Buffer Pool 中执行的。

但是数据不是在磁盘中的吗？怎么会和缓存池又有什么关系呢？那是因为如果 MySQL的操作都在磁盘中进行，那很显然效率是很低的，效率为什么低？因为数据库要从磁盘中拿数据，那肯定就需要IO，并且数据库并不知道它将要查找的数据是磁盘的哪个位置，所以这就需要进行随机IO，那样性能会很差。所以MySQL对数据的操作都是在内存中进行的，也就是在Buffer Pool 这个内存组件中。

实际上他就好比是Redis，因为Redis是一个内存是数据库，他的操作就都是在内存中进行的，并且会有一定的策略将其持久化到磁盘中。那Buffer Pool 的内存结构具体是什么样子的，那么多的增删改操作难道数据要一直在内存中吗？既然说类似 redis 缓存，那是不是也像 redis 一样也有一定的淘汰策略呢？

参考：

<https://mp.weixin.qq.com/s/u6J0xv1YmBjs8KywKW_3wg>

# 配置

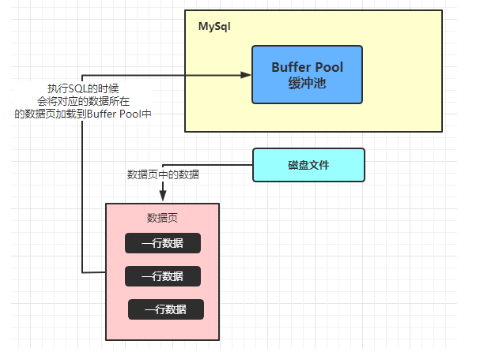
Buffer Pool 是 InnoDB 中的一块内存区域，他一定是有自己的大小的，且大小默认是 128M，不过这个容量似乎有点小了，大家的自己的生产环境可以根据实际的内存大小进行调整，参数为：innodb\_buffer\_pool\_size=2147483648 ，单位是字节。

# 查看和调整innodb\_buffer\_pool\_size  
 1、查看@@innodb\_buffer\_pool\_size大小，单位字节  
 SELECT @@innodb\_buffer\_pool\_size/1024/1024/1024; #字节转为G  
 2、在线调整InnoDB缓冲池大小，如果不设置，默认为128M  
 set global innodb\_buffer\_pool\_size = 4227858432; ##单位字节

# 原理

## 数据页

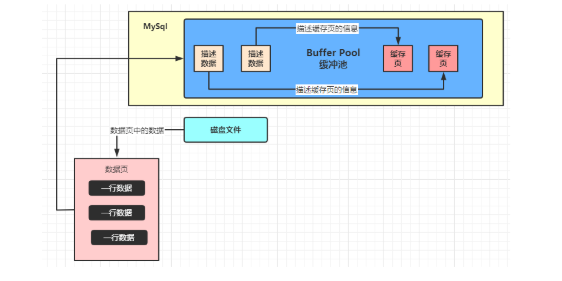
 MySQL在执行增删改的时候数据是会被加载到 Buffer Pool 中的，既然这样数据是怎么被加载进来的，是一条一条还是说是以其他的形式呢。我们操作的数据都是以表+行的方式，而表+行仅仅是逻辑上的概念，MySQL并不会像我们一样去操作行数据，而是抽象出来一个一个的数据页概念，每个数据页的大小默认是16KB，这些参数都是可以调整的。但是**建议使用默认的就好**，毕竟MySQL能做到极致的都已经做了。每个数据页存放着多条的数据，MySQL在执行增删改首先会定位到这条数据所在数据页，然后会将数据所在的数据页加载到Buffer Pool中。



## 缓存页

当数据页被加载到缓冲池中后，Buffer Pool中也有叫缓存页的概念与其一一对应，大小同样是16KB，但是 MySQL还为每个缓存也开辟额外的一些空间，用来描述对应的缓存页的一些信息，例如：数据页所属的表空间，数据页号，这些描述数据块的大小大概是缓存页的15%左右（约800B）。

**# 缓存页是什么时候被创建的？** 当MySql启动的时候，就会初始化Buffer Pool，这个时候MySQL会根据系统中设置的innodb\_buffer\_pool\_size大小去内存中申请一块连续的内存空间，实际上在这个内存区域比配置的值稍微大一些，因为【描述数据】也是占用一定的内存空间的，当在内存区域申请完毕之后，MySql会根据默认的缓存页的大小（16KB）和对应`缓存页\*15%`大小(800B左右)的数据描述的大小，将内存区域划分为一个个的缓存页和对应的描述数据。

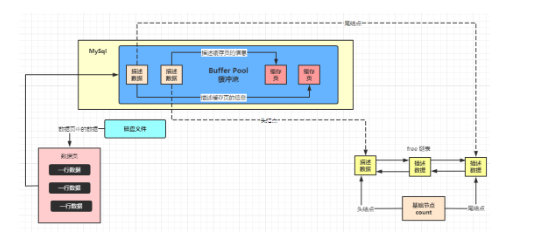


## Free链表

上面是说了每个数据页会被加载到一个缓存页中，但是加载的时候 MySQL是如何知道那个缓存页有数据，那个缓存页没有数据呢？换句话说， MySQL是怎么区分哪些缓存页是空闲的状态，是可以用来存放数据页的。

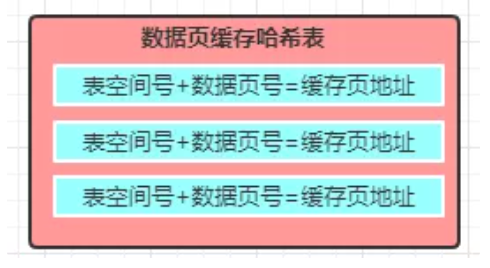
为了解决这个问题，MySQL为Buffer Pool设计了一个双向链表—free链表，这个free链表的作用就是用来保存空闲缓存页的描述块（这句话这么说其实不严谨，换句话：每个空闲缓存页的描述数据组成一个双向链表，这个链表就是free链表）。之所以说free链表的作用就是用来保存空闲缓存页的描述数据是为了先让大家明白free链表的作用，另外free链表还会有一个基础节点，他会引用该链表的头结点和尾结点，还会记录节点的个数（也就是可用的空闲的缓存页的个数）。

这个时候，他可以用下面的图片来描述：



当加载数据页到缓存池中的时候，MySQL会从free链表中获取一个描述数据的信息，根据描述节点的信息拿到其对应的缓存页，然后将数据页信息放到该缓存页中，同时将链表中的该描述数据的节点移除。这就是数据页被读取Buffer Pool中的缓存页的过程。

但MySQL是怎么知道哪些数据页已经被缓存了，哪些没有被缓存呢。实际上数据库中还有后一个哈希表结构，他的作用是用来存储表空间号+数据页号作为数据页的key，缓存页对应的地址作为其value，这样数据在加载的时候就会通过哈希表中的key来确定数据页是否被缓存了。



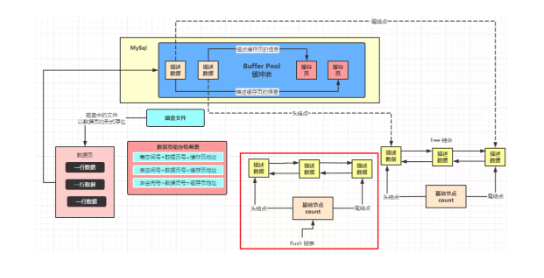
## Flush链表

MySql在执行增删改的时候会一直将数据以数据页的形式加载到Buffer Pool的缓存页中，增删改的操作都是在内存中执行的，然后会有一个后台的线程数将脏数据刷新到磁盘中，但是后台的线程肯定是需要知道应该刷新哪些啊。

针对这个问题，MySQL设计出了Flush链表，他的作用就是记录被修改过的脏数据所在的缓存页对应的描述数据。如果内存中的数据和数据库和数据库中的数据不一样，那这些数据我们就称之为脏数据，脏数据之所以叫脏数据，本质上就是被缓存到缓存池中的数据被修改了，但是还没有刷新到磁盘中。

同样的这些已经被修改了的数据所在的缓存页的描述数据会被维护到Flush中（其实结构和free链表是一样的），所以Flush中维护的是一些脏数据数据描述（准确地说是脏数据的所在的缓存页的数据描述）

另外，当某个脏缓存页被刷新到磁盘后，其空间就腾出来了，然后又会跑到Free链表中了。



## LRU链表

# 总结