1. Buffer Pool
2. 概述

对于InnoDB，为了缓存磁盘中的页，向操作系统申请了一块连续的内存称为缓冲池，默认是128MB。

1. 内部组成

Buffer Pool被划分为若干个页面，页面大小与InnoDB表空间使用的页面大小一致，默认都是16KB，这些页称为缓冲页。为了更好地管理这些缓冲页，每个缓冲页都配置了一个控制块，存储着一些控制信息，如该页所属的表空间编号，页号等等，控制块和缓冲页都在缓冲池中，控制块放在缓冲池的前面，缓冲页放在缓冲池的后面。

1. free链表

既然缓冲页是用来缓存磁盘的页面的，那么有些缓冲页已经被使用了，有些缓冲页还没有使用，我们需要快速找出空闲的缓冲页，因此将所有空间的缓冲页对应的控制块放到一个free链表中。该链表的基节点中包含了链表的头节点地址，尾节点地址，以及当前链表中节点的数量，其余的每个控制块中包含着链表的前后指针。

1. 判断缓冲页存在于缓冲池

如何判断某个缓冲页是否存在于缓冲池中？

我们都是通过表空间号+页号来定位一个页的，因此以表空间号+页号作为一个哈希表的key，以缓冲页控制块的地址作为value。这样我们就能够快速地判断出某个缓冲页是否存在于缓冲池中，如果没有就从free链表中选出一个空闲的缓冲页，将磁盘中对应的页加载到缓冲页中。

1. flush链表

如果我们修改了缓冲池中某个缓冲页的数据，它就与磁盘上的页不一致了，这种页称为脏页。如果我们修改一次就刷新到磁盘一次，性能会受到影响，因此我们选择将这些脏页使用链表存起来，称为flush链表。

1. LRU链表
2. 简单的LRU

缓冲池的大小是有限的，如果需要缓存的页的大小超过了缓冲池的大小，就会将某些旧的缓冲页从缓冲池中移除，再将新的页放进来。为了提高缓冲池的命中率，我们需要淘汰掉最近很少使用的部分缓冲页，这就需要创建一个链表，按照最近最少使用的原则淘汰缓冲页，所以这个链表称为LRU（Least Recently Used）链表。

当需要访问某个页时，如果该页不在缓冲池中，在把该页从磁盘加载到缓冲池中的缓冲页时，就把该页对应的控制块作为节点塞到LRU链表的头部。这样，LRU链表的尾部就是最近最少使用的缓冲页了，当缓冲池中的空闲缓冲页使用完时，淘汰LRU链表尾部的一些缓冲页即可。

1. 划分区域的LRU

有些情况下，简单的LRU反而会造成性能的损耗，如对很大的表进行全表扫描时，可能会把缓冲池中的缓冲区都更换一遍，会将使用频率很高的页从缓冲池中淘汰掉；又或者加载到缓冲池中的页不一定用到，都会导致缓冲池命中率降低。

因此，将LRU链表划分为两部分，一部分存储使用频率很高的缓冲页，称为热数据，一部分存储使用频率低的缓冲页，称为冷数据。两部分的比例并不是对半分，可以通过查询系统变量 innodb\_old\_blocks\_pct得知。当磁盘上的某个页面初次加载到缓冲池中的某个缓冲页时，该缓冲页对应的控制块会放到冷数据区的头部，不会影响到热数据区使用频繁的缓冲页。如果该缓冲页第二次被访问，就将其控制块移动到热数据的头部。对于热点数据，可能会频繁访问，不需要频繁地移动热数据的节点到头部，只有在位于热数据区域1/4的后面时，才会被移动到LRU链表的头部。