

如何做一个LRU Cache

这是容易被面试的内容

LRU Cache

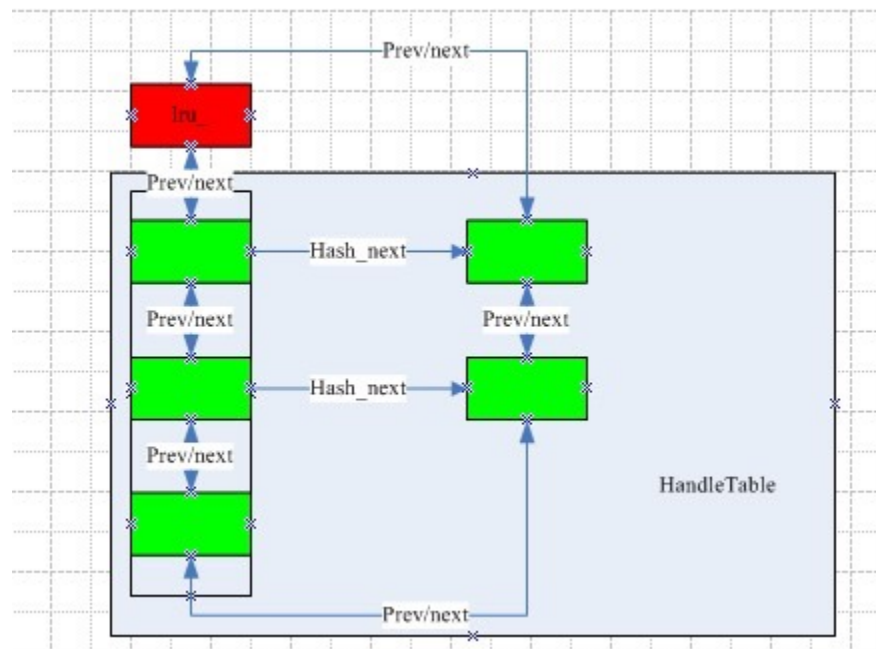
- 注意这里LevelDB 需要把一个Sstable的Indexblock或者Datablock加载到Cache中
- Cache的作用是什么？
 - 对最近经常访问的磁盘数据，加载到内存中，这样可以做到直接从内存中读取数据而无需访问磁盘
- Cache总是有用的么？
 - 对经常需要写盘的结构没有用
 - LevelDB中的Sstable几乎是不变的
- Cache的特性
 - 给一个数据，需要很快知道它是否在Cache中(命中)
 - 需要知道最近一直没有使用的数据，Least Recently Used的由来

LRU Cache的设计

- 请大家自己思考此处数据结构的设计
 - 高效的判定是否命中？
 - 哈希表
 - 如何筛选LRU的数据??
 - 链表，把新插入的数据放在表头，更新的数据也放到表头，这样不访问的数据慢慢会聚集到表尾
 - 所以是哈希表与链表的结合。
 - 用哈希链地址法，另外做一个链表，把哈希表中所有的节点链起来

LRU Cache的设计

- Lru : 链表的头指针
- Prev/next: 链表的前驱后继指针
- Hash_next哈希表的后继指针



LevelDB中的Cache

- 代码 cache.cc
- SharedLRUCache
 - 16个分区后的LRUCache
- LRU Cache
- HandleTable(哈希表)
- LRUHandle(哈希表的一个节点)

