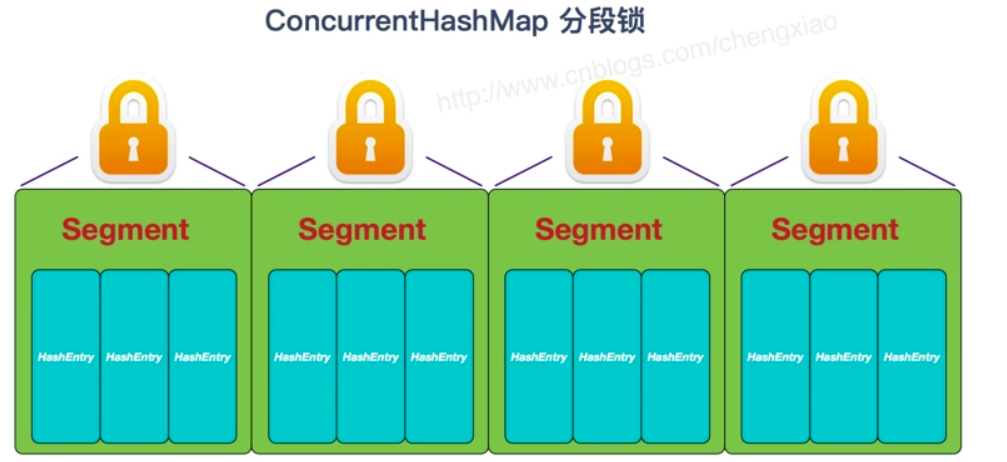
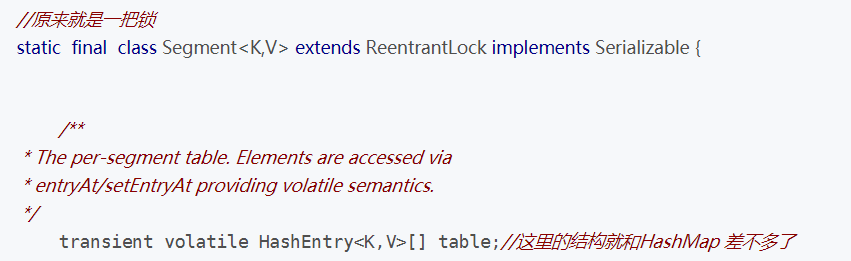


1.7 segments+HashEntry数组





Lock/unlock

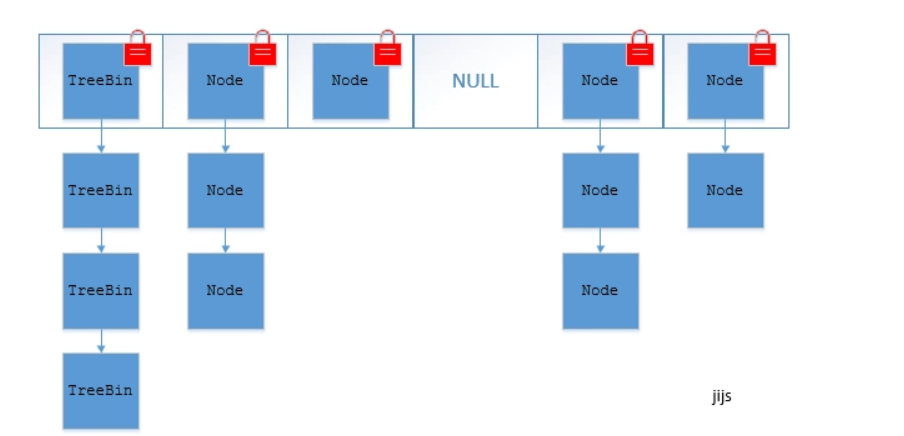
Segment 包含一个 HashEntry 数组， HashEntry 一个链表结构的元素，每个 Segment 守护着一个HashEntry数组里的元素，当对 HashEntry 数组的数据进行修改时，必须首先获得对应的 Segment的锁

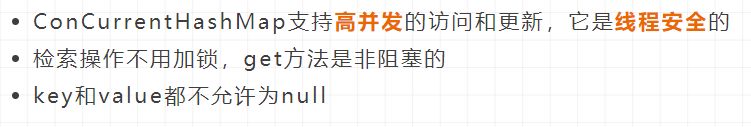
HashTable用的是synchronized，会锁住整个对象，一个HashTable上的操作都是并行的，连get方法都会阻塞其他操作

取消了Segment分段锁，采用CAS和synchronized来保证并发安全

CAS比较与交换，Compare and swap当多个线程尝试使用CAS同时更新同一个变量时，只有其中一个线程能更新变量的值(旧值和内存值相同时，将内存值修改为新值)

1.8 ConCurrentHashMap的底层是：Node数组+链表/红黑树 锁住每一个node节点





默认大小为16

扩容时新生成的数组，其大小为原数组的两倍

通过部分锁定(syn)+CAS算法来进行实现线程安全的

假设table已经初始化完成，put操作采用CAS+synchronized实现并发插入或更新操作

ConcurrentHashMap的三个原子操作

