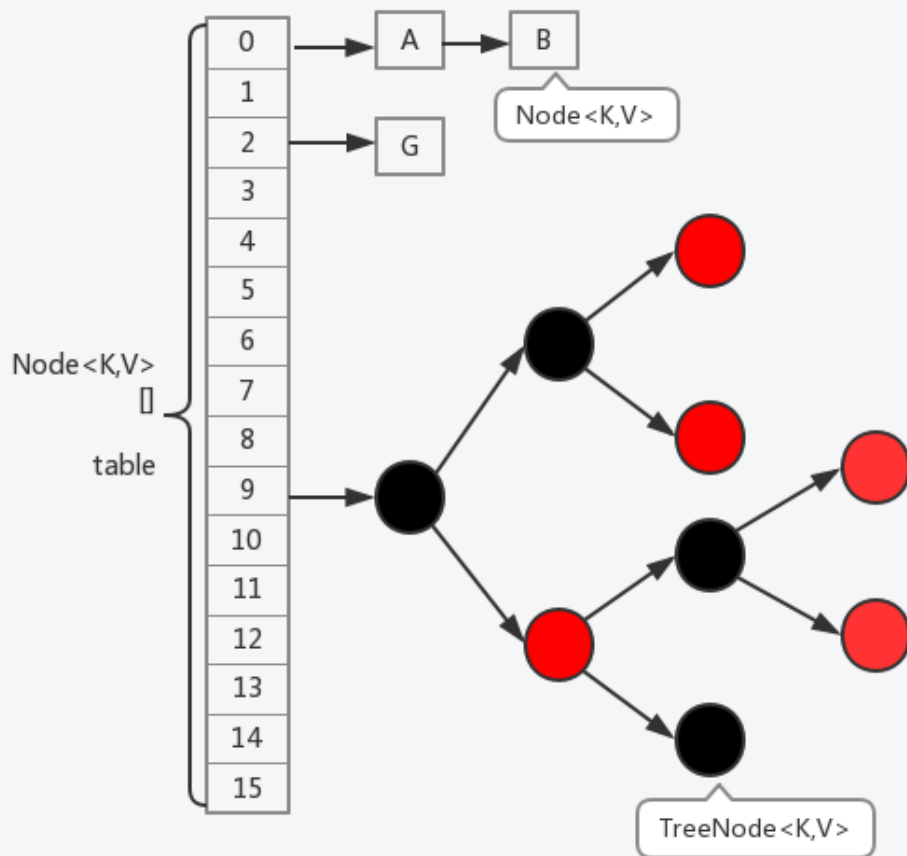


ConcurrentHashMap取消了Segment分段锁的数据结构，取而代之的是数组+链表（红黑树）的结构。而对于锁的粒度，调整为对每个数组元素加锁。定位节点的hash算法被简化了，这样带来的弊端是Hash冲突会加剧。因此在链表节点数量大于8时，会将链表转化为红黑树进行存储。这样一来，查询的时间复杂度就会由原来的 $O(n)$ 变成 $O(\log(n))$ 。



https://blog.csdn.net/fouy_yun/article/details/77816587

get()方法的流程相对简单一点，从上面代码可以看出以下步骤：

1. 首先定位到table[]中的i。
2. 若table[i]存在，则继续查找。
3. 首先比较链表头部，如果是则返回。
4. 然后如果为红黑树，查找树。
5. 最后再循环链表查找。

从上面代码可以看出，put的步骤大致如下：

1. 参数校验。
2. 若table[]未创建，则初始化。
3. 当table[i]后面无节点时，直接创建Node（无锁操作）。
4. 如果当前正在扩容，则帮助扩容并返回最新table[]。
5. 然后在链表或者红黑树中追加节点。

6. 最后还回去判断是否到达阈值，如到达变为红黑树结构