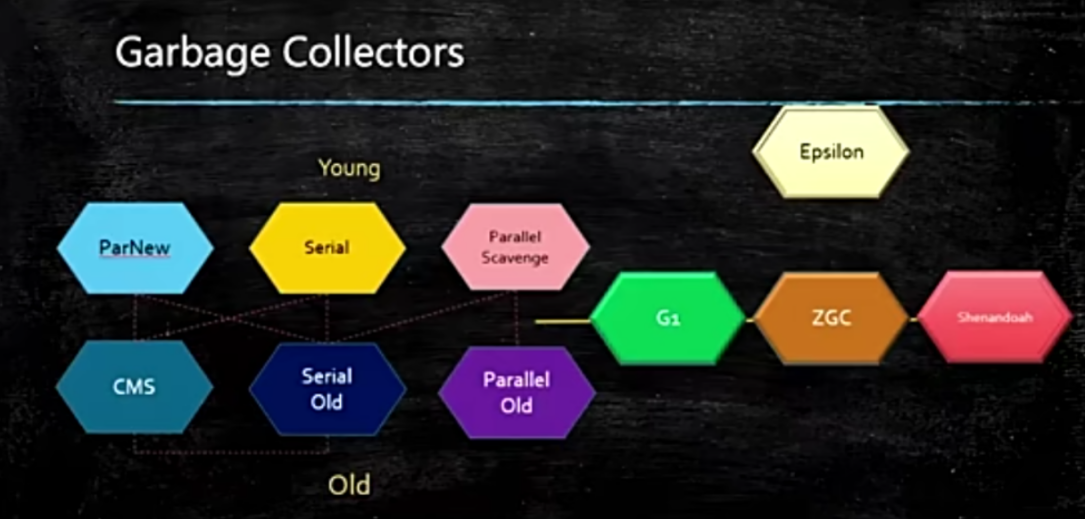


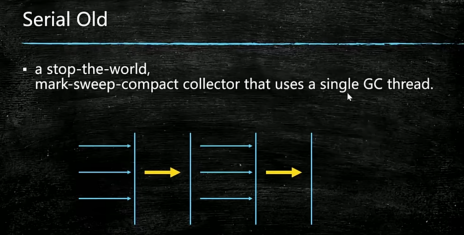
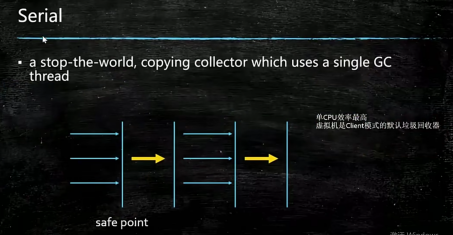
1、常见的垃圾回收器



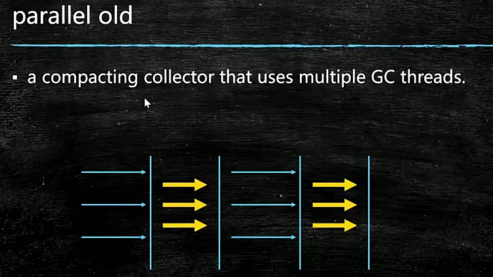
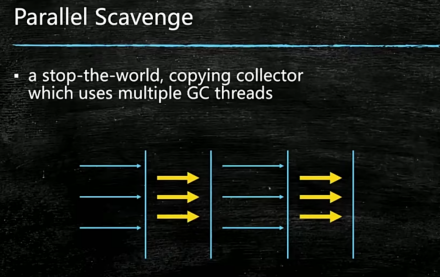
年轻代（S,PC,PN）：标记-复制算法

老年代（SO,PO,CMS）：CMS标记-清除法，其他标记-整理法

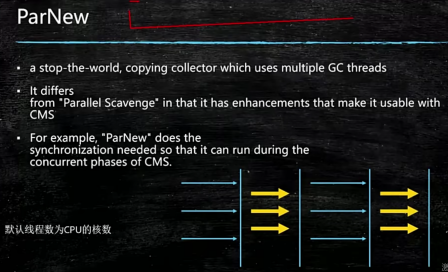
（1）组合1（STW）



1. 组合2（STW）

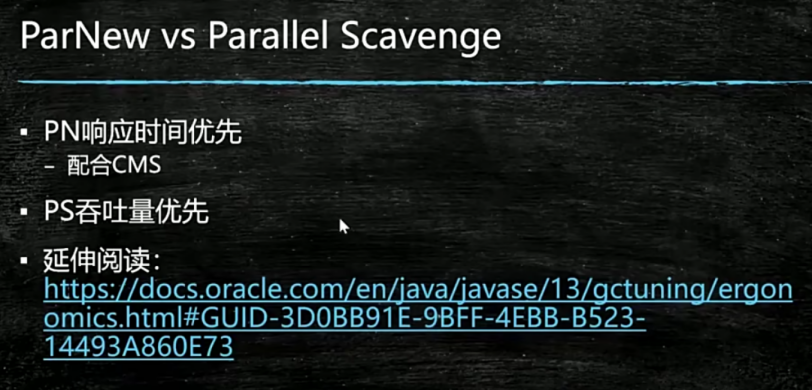


1. 组合3（部分STW）

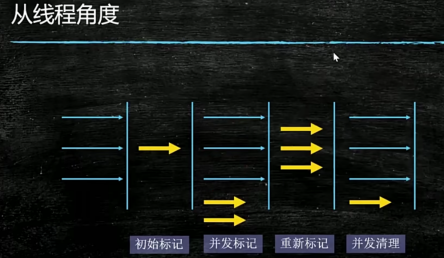


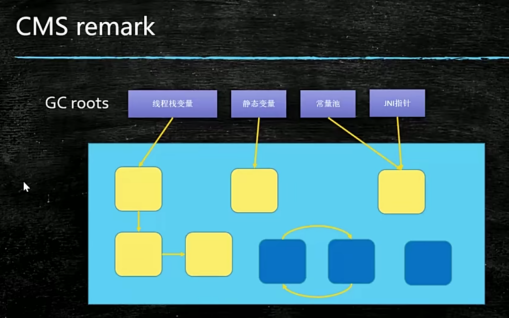
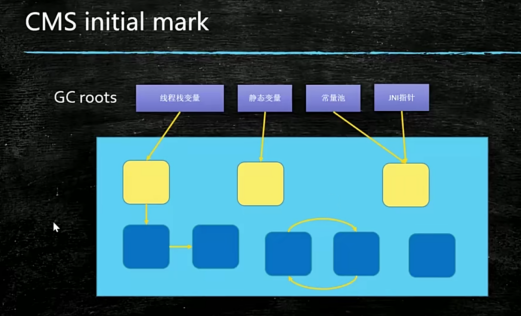
PC与PN对比

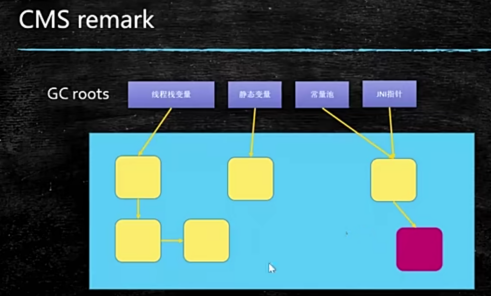
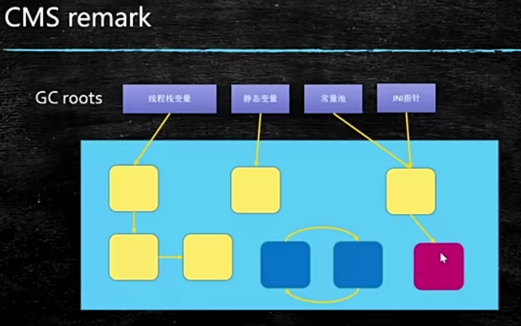
PN是为了配合CMS（降低STW的时间）而产生的



1. CMS
2. 四个阶段

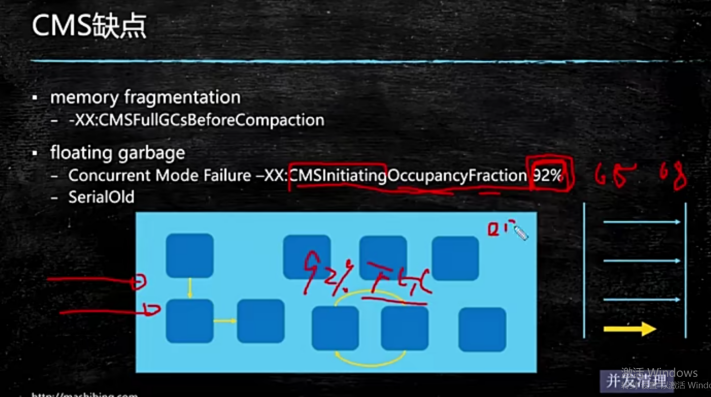






1. 缺点





* 内存碎片：

Promotion failure：当内存碎片很多时，新生代对象进入老年代时空间不足，由于CMS无法解决内存碎片问题，所以就会触发Serial Old垃圾回收器进行标记-整理，在内存很大的时候导致STW很严重。

解决方法：设置参数提前压缩

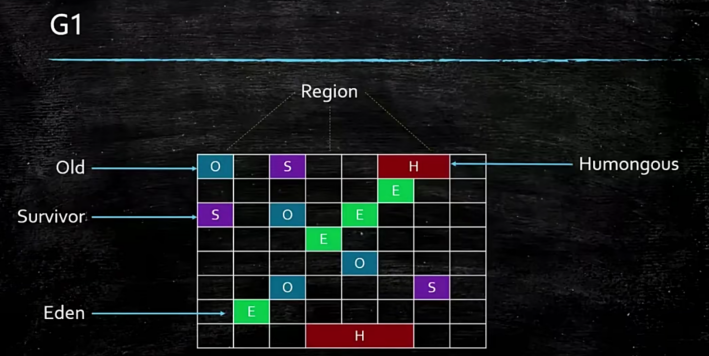
* 浮动垃圾：

Concurrent Mode failure：产生的浮动垃圾没有清理，导致老年代空间不足，还是触发Serial Old。

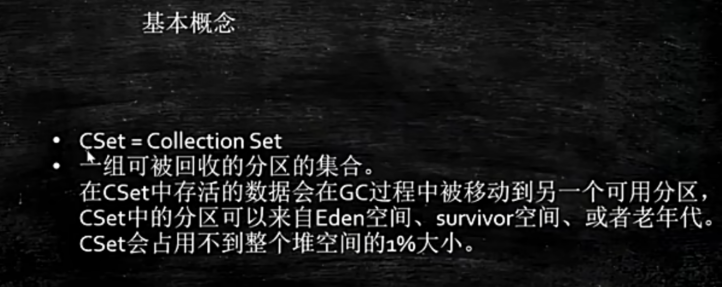
解决方法：降低CMS的FGC阈值，提前FGC，保证足够空间

1. G1垃圾回收器
2. 基本概念

逻辑分区，物理不分区

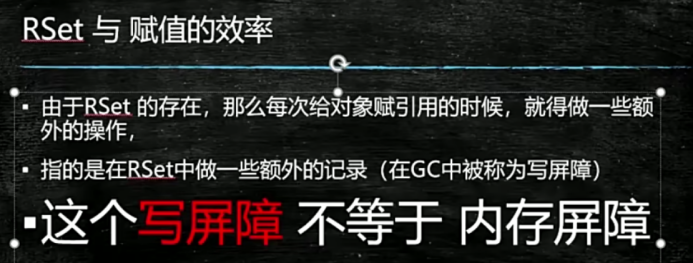


Cset：记录可以被回收的分区



Rset：记录其他分区的对象到当前分区的引用，所有对本分区回收不需要扫描整个堆区，只需要扫描当前分区即可。



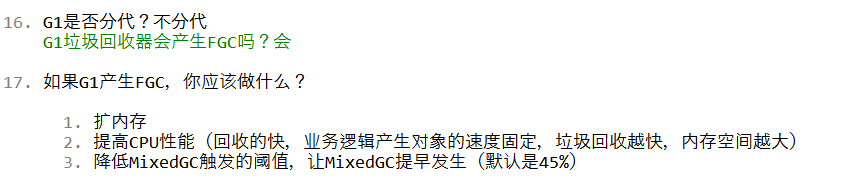


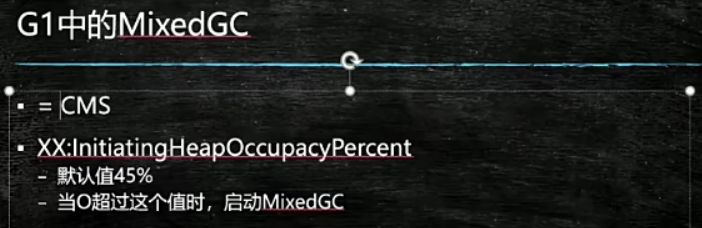
1. 三种GC

Yong GC：Yong GC【年轻代不足】

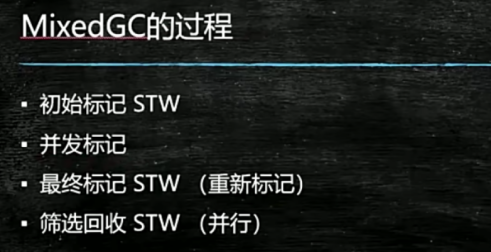
Mixed GC：Yong GC+部分Old GC【老年代空间占有率到达一定值，只有G1回收器】

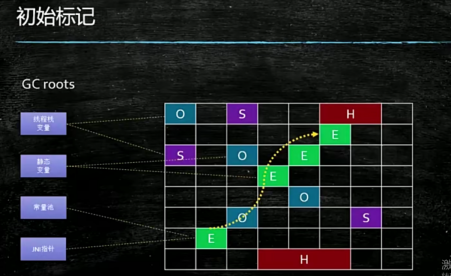
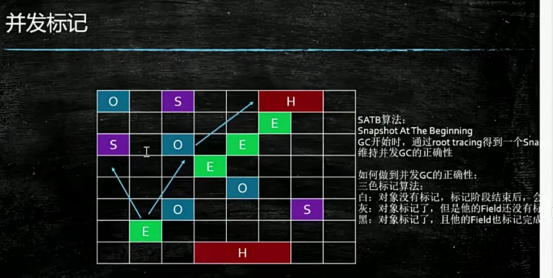
Full GC：STW单线程回收【老年代不足】

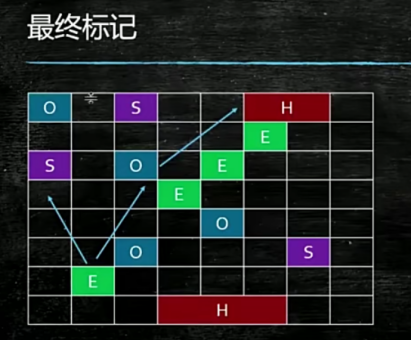
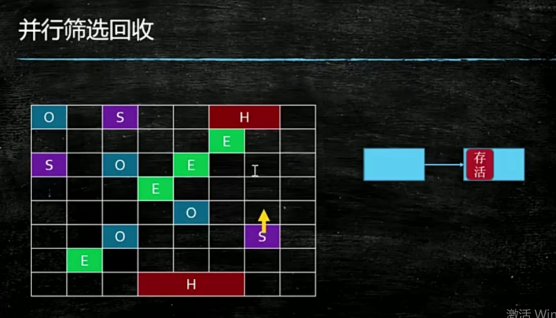




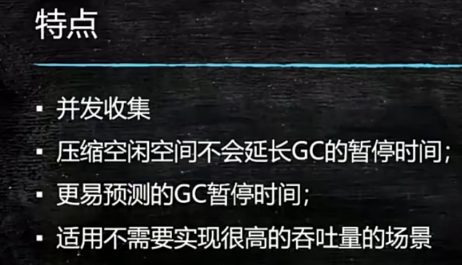
1. Mixed GC



 （复制）

1. 特点

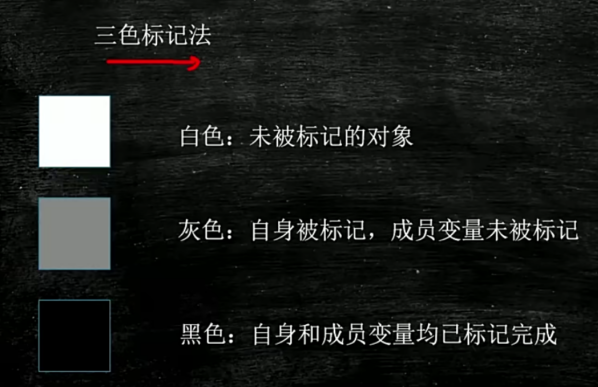


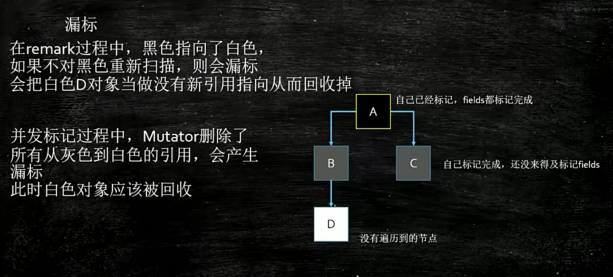
1. 算法-三色标记，解决漏标

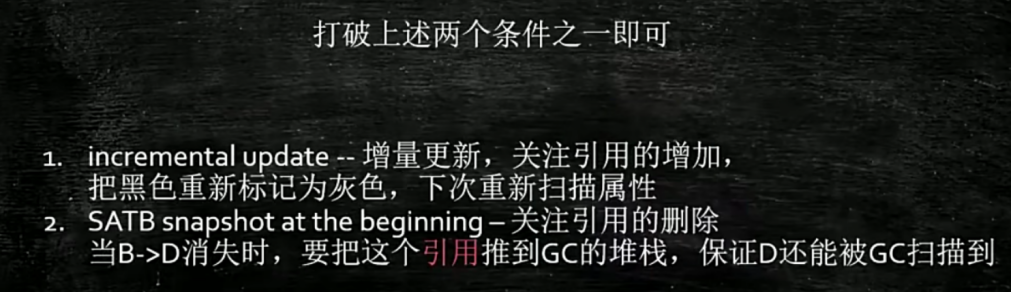
解决并发标记过程中，漏标

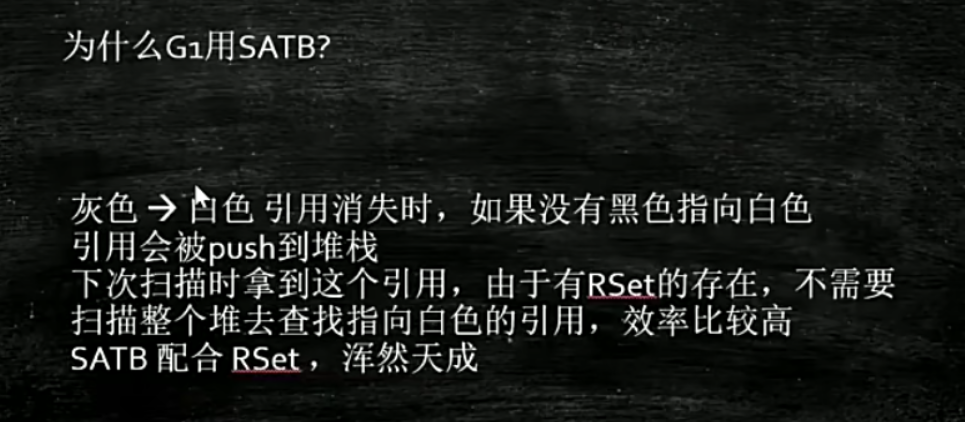
增量更新：需要重新遍历灰色对象，效率不高。CMS。

STAB：关注被删除的引用，首先将删除的引用关系压倒GC堆栈，GC扫描的GC堆栈时候，会结合Rset判断是否有引用指向此白色对象，再继续垃圾回收。G1.



，





1. 算法-Card Table

