链滴

Java 垃圾回收简介

作者：teneous
原文链接：https://ld246.com/article/1558103650838
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

HotspotGC收集器入门

@(syoka)

序言:本文参考自官方文档，截取了部分进行了翻译(Description Garbage Collection)

收获:读者看完本文能对GC有一个初步的理解

原文地址:https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html

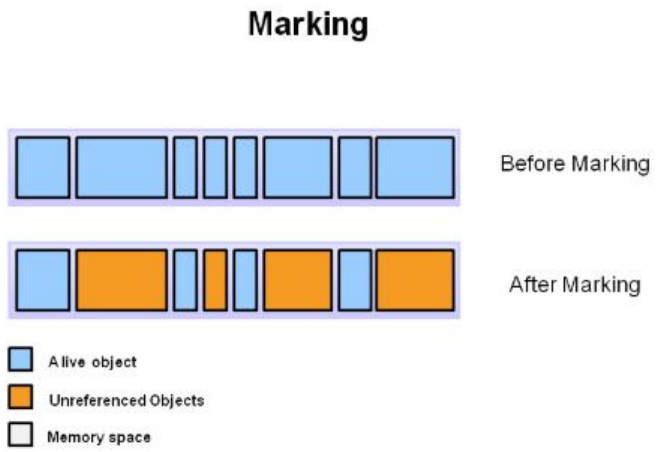
---

垃圾收集器

什么是自动垃圾回收器

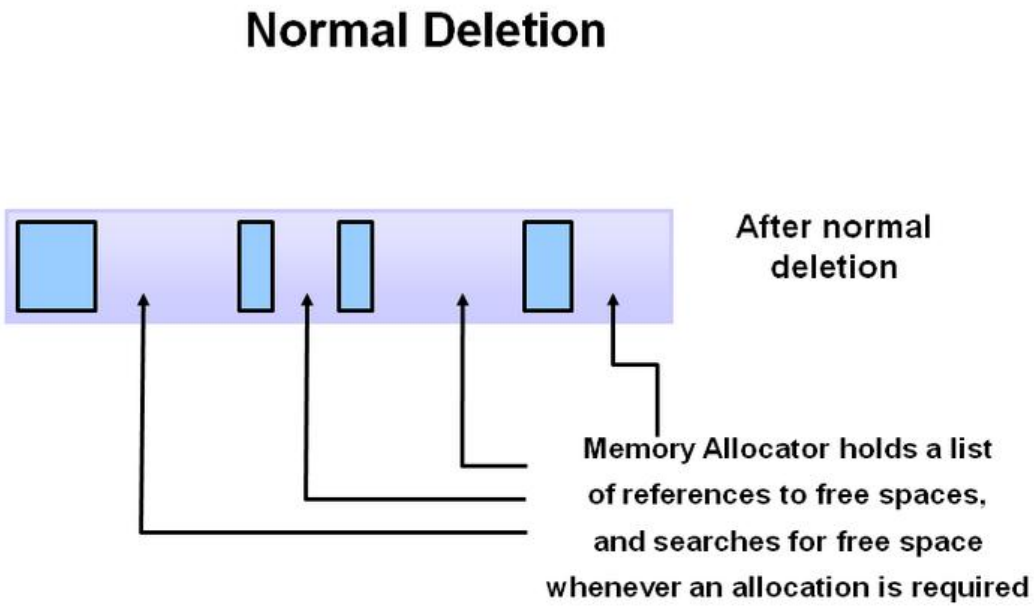
JAVA垃圾收集器会在运行中不断查看堆内存，确认哪些对象当前还有在使用，哪些已经没有使用了同时清理掉那些没有被使用的对象。 一个在使用的对象或者说一个被引用的对象，意味着在你程序某个地方依然保留着一个指针指向这个对象。如果是一个没有在使用的或 者已经没有任何引用指向对象的话，那么这样的对象就会被回收。 在某些编程语言中类似C，需要自己手动申请和释放内存，是在java中，释放内存是由GC收集器自动执行的，执行步骤如下

第一步：标记 顾名思义,就是垃圾收集器会标记内存中哪些区块是在使用的哪些是没在使用的。



蓝色区块是当前还被引用到对象,而橙色则是已经没有被引用的对象。所有对象都会被扫描标记。如果描系统中的所有对象会是一个非常耗时的操作。

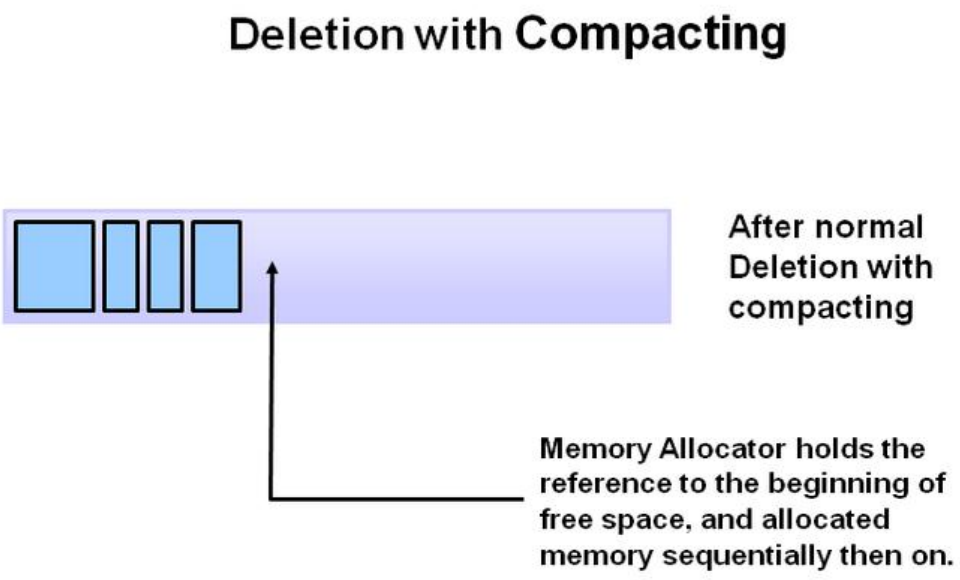
第二步: 通常删除 普通的删除会移除那些没有再被引用的对象、保留还有引用的对象然后使内存分配器的指针指向空闲空间。



内存分配器保持对可分配新对象的自由空间块的引用。 注:从以上理解中，如果仅仅使用标记法,在通删除后，内存中会出现很多零散的碎片,新分配的对象也会比较复杂(不是连续分配),因此我们有了以下应方案。

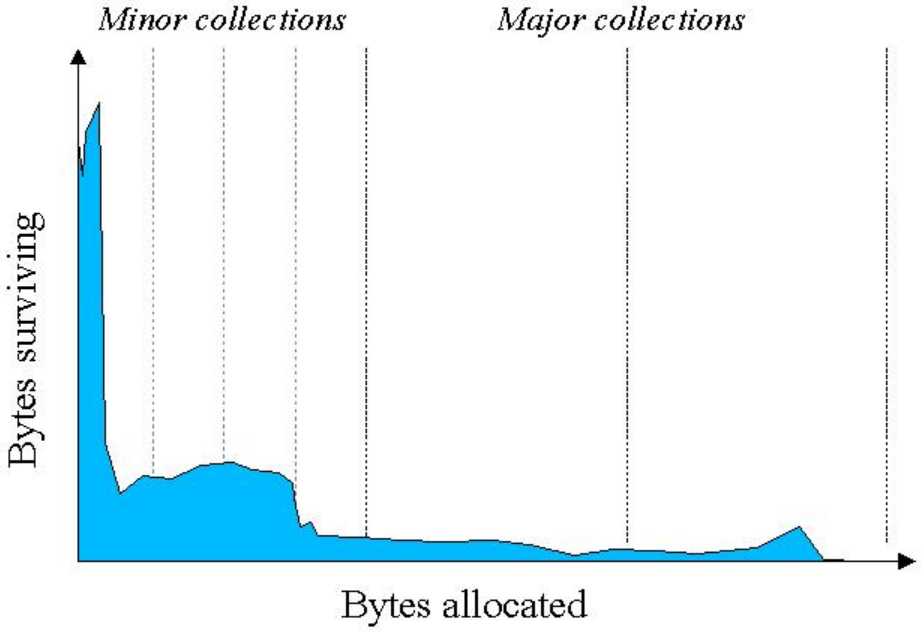
第二步A:删除与整理

为了进一步提高性能，删除未被引用的对象，你可以压缩整理内存中剩余的引用对象，通过将内存中用对象移动到一起，从而保证新内存在分配时会更加快速简单。



为什么使用分代垃圾回收

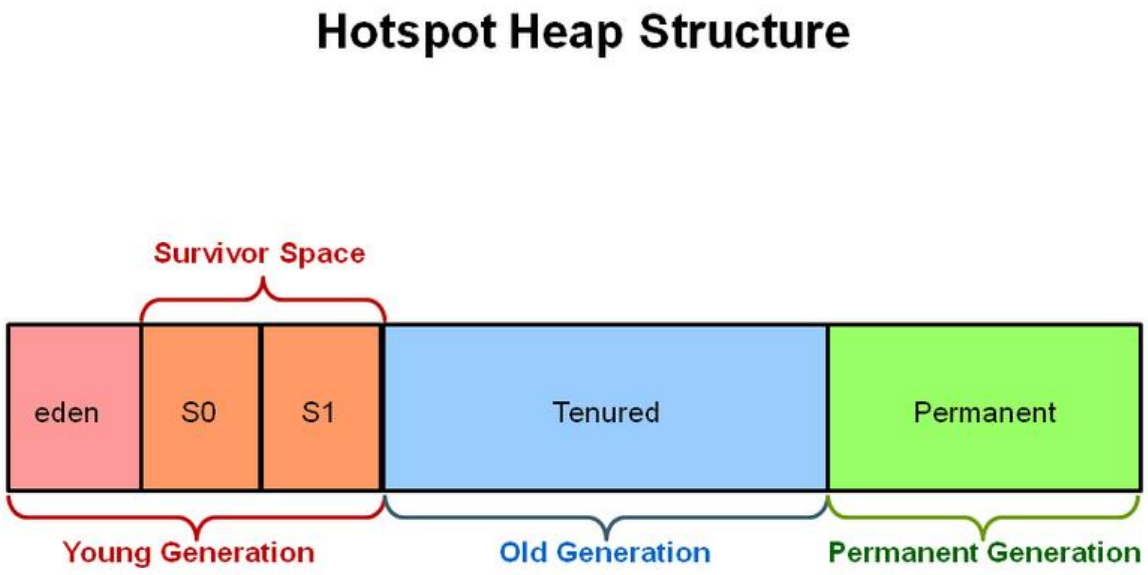
在早期，在JVM中使用标记和压缩的效率是非常低的，随着越来越多的对象被分配，保存引用对象的表会变的越来越大，垃圾收集的时间也会越来越长。然后，HotSpot的开发者根据分析出，大部分的象其实存活时间都非常的短。



正如你所见，只有很少很少的对象可以存活很长时间，大部分都是存活很短的时间。 这里是一个典的例子，Y轴为被分配的对象数量大小，X轴代表分配内存的时间。

JVM分代

学习JVM对内存对象的分配行为有助于提升JVM的性能。因此，我们将堆分配为几个小部分，它们分是:**年轻代**，**老年代**，**终生代** ，**持久代**



年轻代是新对象被分配和老化的场所，当青年代被堆满的时候，就会出现一次Minor garbage collect on(小型垃圾回收) Minor被优化为假定所有对象都是高死亡率。一个充满死亡对象年轻代的回收是常快的。一些存活者对象会逐渐老化最终移动到老年代

世界终止事件

所有的Minor回收都是世界终止事件，它意味这所有线程都会被暂停，直到完成收集操作。

老年代被用来存储长时间存活的对象（也有些大对象会直接分配到老年代），典型的来说，当年轻代客老化到一定程度的时候就会被移动到老年代，当老年代需要被回收的时候，就会发生一次Major Ga bage Collection(规模比较大的垃圾回收)

Major garbage collection当然也是世界终止事件，经常进行Major回收的话是非常的慢的，因为每都会检查全部活着的对象。对于响应式应用，Major垃圾回收应该尽可能的少 Major garbage collect on的世界停止事件的时间长度会受老年代所使用的垃圾收集器类型的影响 持久代包含JVM用于描述和方法所需的必要源信息，在运行时期，JVM在运行期使用的应用类会被放在持久代中。例如，JAVA SE包的类和方法就是存放在这里的

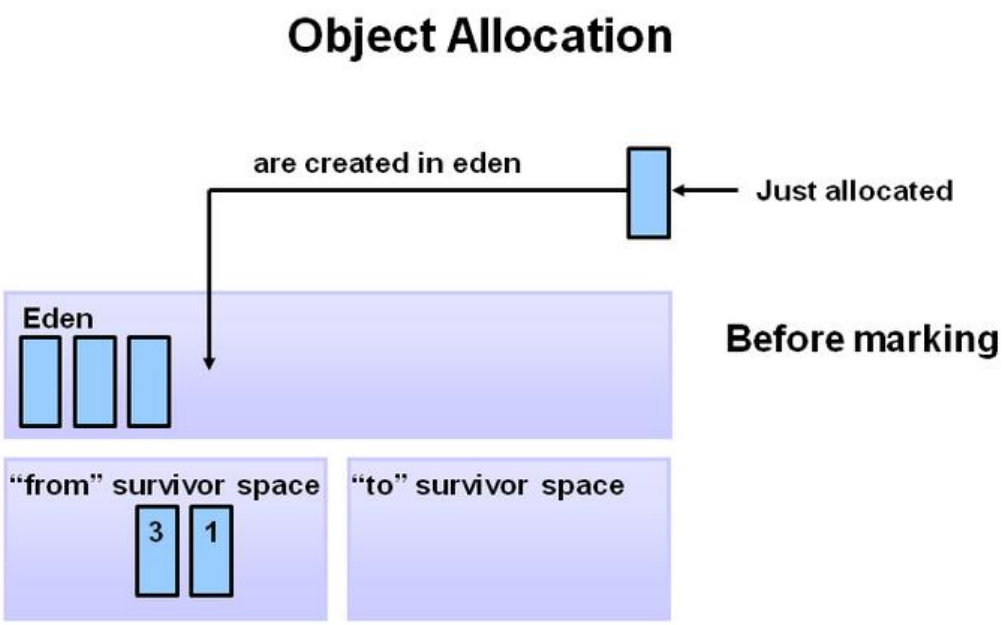
如果JVM发现某些类不会在被使用并且其他类可能需要空间的时候，不被需要的类就会被垃圾器回收卸载)。持久代会包含一次Full GC。

通常垃圾回收的工序

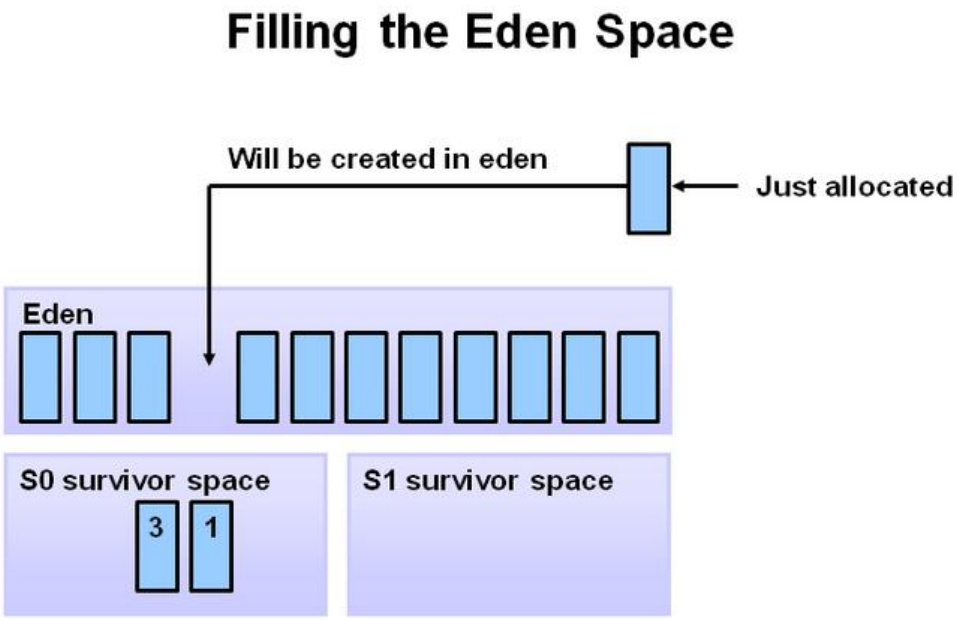
现在你可以一件知道为什么堆会被分成几块不同的代区了，是时候看看这些区块是如何关联的，接下

这些图片将会告诉JVM中的对象分配和老化是如何进行的

1.首先所有新对象都会被分配在Eden区，幸存者0和1区域都是空的

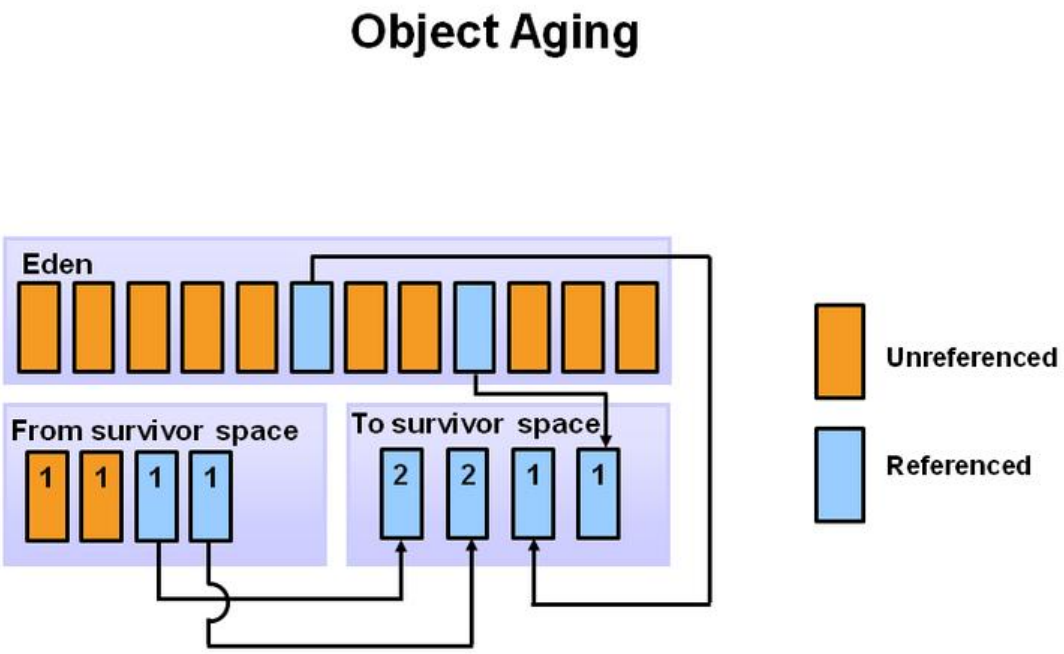


2.随着对象不断的被分配，最终Eden区会被充满，minor gc就会被触发

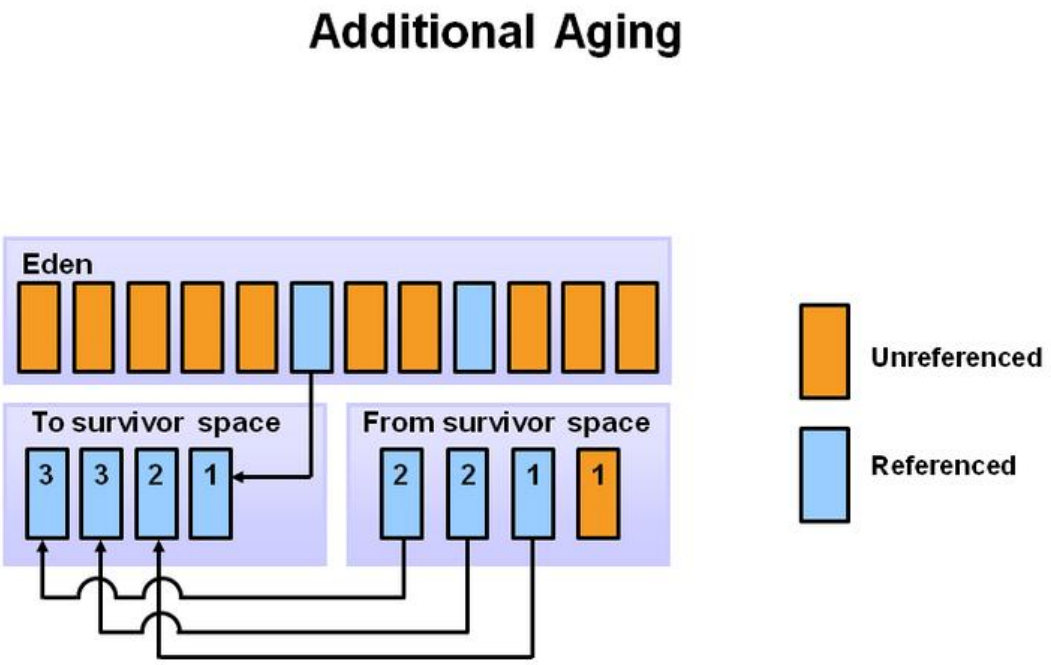


3.还在被引用的对象就会被移动到第一个幸存者区（s0），没有被引用的对象就会被清除

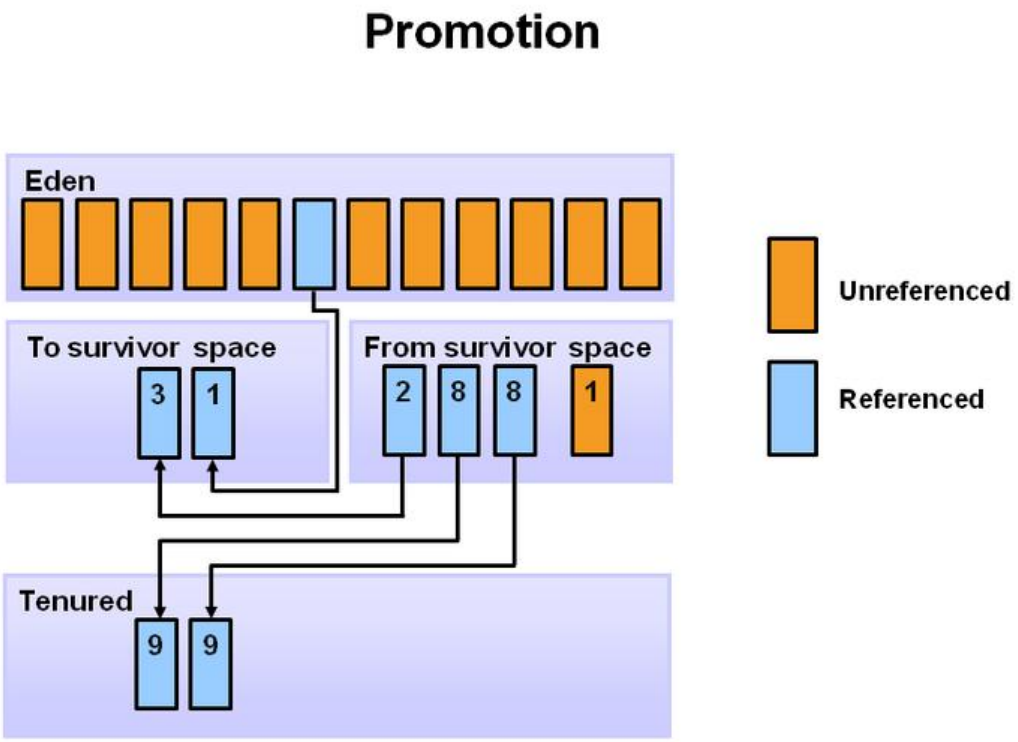
4.在下一个minor gc后，Eden会发生相同的操作，没有引用的对象会被删除，有引用的对象会被移到s1区(to)，另外，s0中还有引用的对象也会被移动到s1中(复制算法) 并且年龄+1,然后会清空Eden s0区，注意我们现在的s1区中存在着不同年龄段的对象。



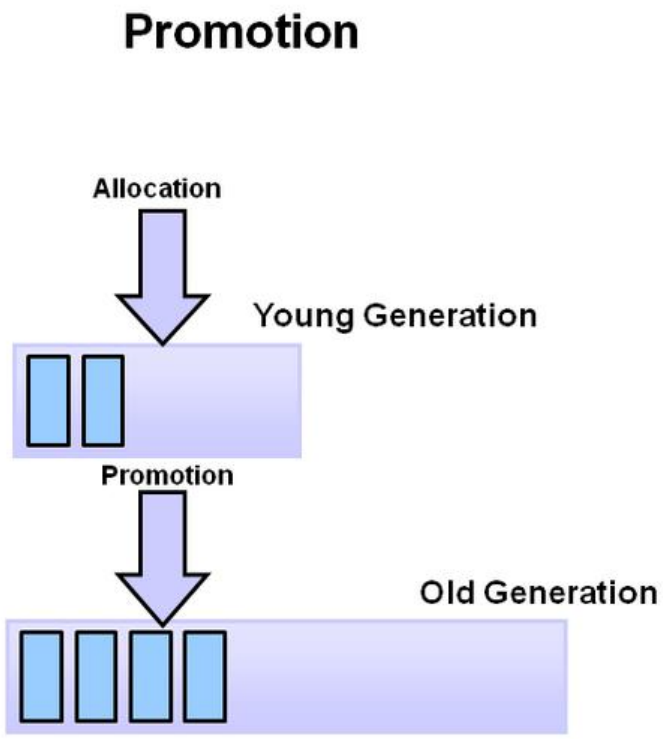
5.在下一次minor gc后，执行同样的操作，但是此时注意幸存者的复制角度发生了改变(原:s0->s1)现(1->s0)，然后对象年龄+1，s1被清除



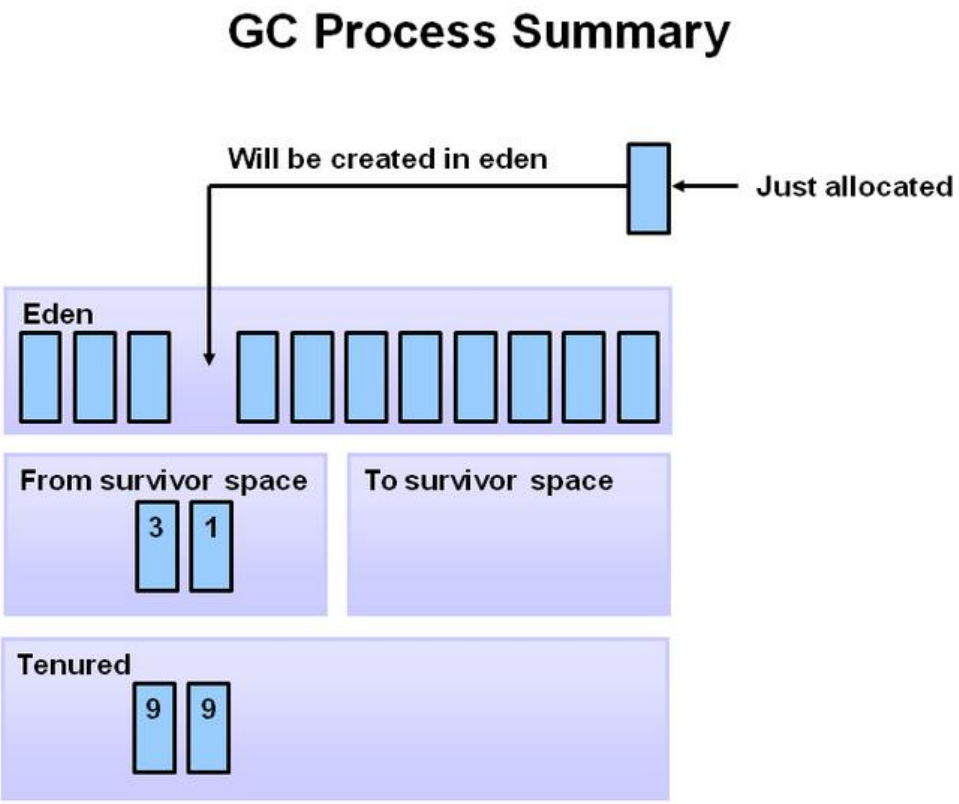
6.照这样下去，当对象的年龄到达一个阈值的时候（本例是8），如果此时对象依然存活就会被移动老年代去



7.随着minor gc的发生，到了一定年龄的幸存者对象会被移动到老年代中



8.上述几乎涵盖了年轻代的整个过程，最终，major GC会在老年代中执行清理无用对象并且压缩空间。



JAVA垃圾回收器

常见参数设置:

参数	描述
-Xms	JVM启动时的堆初始化大小
-Xmx	堆的最大值
-Xmn	青年代的大小
-XX:PermSize	永久代的初始化大小
-XX:MaxPermSize	永久代的最大值

The Serial GC(串行GC)

串行GC收集器是JAVA SE5和SE6的默认客户端类型机器,在串行下,minor(作用于年轻代)和major(老代)收集器都是以串行方式执行的(单个虚拟CPU)， 另外，它也可以用于标记压缩的收集方法，它将内存中的对象移动到堆首(FILO)，这样的话，以便新的内存分配在堆末尾成为单个连续的内存块。样经过整理的堆在分配新内存块的时候会相当的快

Usage Case(适用场景)

串行GC收集器对于没有短暂时间暂停硬性要求的客户端类型机器来说是一个不错的选择。充分利用单虚拟垃圾处理器的性能。

对于今天的硬件来说，串行GC可以仅用堆中几百MB来有效的管理大量的琐碎应用。即使在最坏情况就几次短暂暂停(在进行full gc情况下）。

另一个受欢迎使用串行GC的场景就是大量的JVM运行在相同的机器上(在某些场景，JVM超出了可用理器)。在这种情况下，如果一个JVM正在GC的时候，最好只使用一个进程处理，这样 可以将对其他VM的干扰最小化，即使gc持续时间比较长，串行GC在这方面权衡的很不错 。

最后，随着嵌入式硬件的增加，内存最少，内核很少，串行GC可能会卷土重来。

命令行变更

想要开启串行收集

-XX:+UseSerialGC

这里是一个命令用例

java -Xmx12m -Xms3m -Xmn1m -XX:PermSize=20m -XX:MaxPermSize=20m -XX:+UseSerialC -jar c:\javademos\demo\jfc\Java2D\Java2demo.jar

---

The Parallel GC(并行GC)

并行GC也可以叫做吞吐量收集器，它使用多个CPU作用于新生代

默认你有几个CPU它就会启动几个垃圾收集器，控制并行个数可以使用以下命令：

-XX:ParallelGCThreads=期望数值

在单个CPU下，哪怕设置了并行回收器，虚拟机依然会采用默认方式。

在2核CPU下，并行收集器的性能与默认垃圾回收器的性能一样，在超过2核CPU以上的时候，年轻代圾回收的暂停时间明显的减少了。

Usage Case(适用场景)

这种收集器适用于处理大量任务并且可以接受长停顿，比如batch进程，类似打印报表，或者从数据查询大量数据并显示出来

-XX:+UseParalleGC

使用这条命令行，你可以得到一个多线程年轻代收集器和一个单线程的老年代收集器，该选项还可以单线程下对老年代压缩

-XX:+UseParallelOldGC

使用这条命令行，你可以得到一个多线程年轻代收集器和一个多线程老年代收集器。它也是一个多线压缩收集器。HotSpot只会在老年代进行压缩，而青年代只是被看作一个副本收集器，因此它不需要缩

压缩描述了对象之间以一种无缝方式进行移动。在GC打扫后，2个存活对象之间可能存在空洞，压缩会移动对象，这样对象间就没有空洞了。垃圾收集器也可能不支持压缩。因此，**并行收集器和并行压**

收集器区别就在于后者会在GC打扫完后进行空间压缩，而前者不会。

The Concurrnet Mark Sweep(CMS)Collector

并发标记打扫(Eclipse默认使用)(也是一个并行短时暂停的收集器)收集老年代。它尝试通过与应用程线程同时执行大量垃圾收集工作来最小化由于垃圾收集而导致的暂停，通常来说，并行短时暂停的收器不会复制或者压缩存活对象。没有存活对象的时候垃圾收集器就不再工作了。如果碎片成为一个问，那么就会分配一个更大的堆。(因为不会压缩空间)

注:用于年轻代的CMS收集器与并行收集器的算法相同

Usage Case(适用场景)

CMS收集器用于短时暂停并且可以与垃圾收集器间共享资源的应用。举个栗子，响应事件的桌面UI应，响应请求的网络服务器，或者一个响应查询的数据库。

Command Line Switches

开启CMS收集器

-XX:+UseConcMarkSweepGC

设置线程数量

-XX:ParallelCMSThreads=<n>

The G1 Garbage Collector

G1收集器用于JAVA7,是用来代替CMS的，G1是并行并发逐渐压缩且短时暂停的垃圾收集器，同时它对于之前描述的收集器还有不同的布局，然而，详细的内容已经超过了本次的分享内容

命令行变更

要开启G1处理器

-XX:+UseG1GC

鉴于篇幅过长，G1的话，后面我分享一个别人的或者自己写一篇