


JVM垃圾收集（三） — Appel式回收为什么使用两个Survivor

原创wxcperfect 于 2021-08-19 17:21:55 发布 248 收藏 版权

分类专栏: JVM 文章标签: jvm java jvm.gc

 JVM 专栏收录该内容

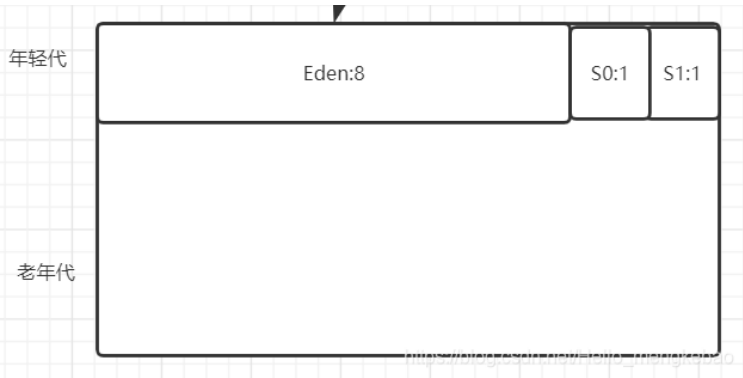
0 订阅 6 篇文章 订阅专栏

Appel式回收为什么使用两个Survivor

一、Appel式回收

Appel式回收的具体做法是把新生代分为一块较大的Eden空间和两块较小的Survivor空间，每次分配 **内存** 只使用Eden和其中一块Survivor。发生垃圾搜集时，将Eden和Survivor中仍然存活的对象一次性复制到另外一块Survivor空间上，然后直接清理掉Eden和已用过的那块Survivor空间。HotSpot虚拟机默认Eden和Survivor的大小比例是8 : 1，也即每次新生代中可用内存空间为整个新生代容量的90%（Eden的80%加上一个Survivor的10%），只有一个Survivor空间，即10%的新生代是会被“浪费”的。当然，98%的对象可被回收仅仅是“普通场景”下测得的数据，任何人都没有办法百分百保证每次回收都只有不多于10%的对象存活，因此Appel式回收还有一个充当罕见情况的“逃生门”的安全设计，当Survivor空间不足以容纳一次Minor GC之后存活的对象时，就需要依赖其他内存区域（实际上大多就是老年代）进行分配担保（Handle Promotion）。

Appel式回收示意图：



二、为什么要有survivor

如果没有survivor，即意味着只有新生代和老年代，新生代在进行一次minor gc后存活的对象无处安放，只能升级到老年代中，这样老年代很快就会被装满，进行full gc（或者major gc,目前只有CMS收集器会有单独收集老年代的行为。另外请注意“Major GC”这个说法现在有点混淆，在不同资料上常有不同所指，读者需按上下文区分到底是指老年代的收集还是整堆收集。），会出现stop the world，应用无法作出响应。这种情况下，要么增加老年代的大小，从而降低Full GC的频率，这样做的缺点就是：由于老年代空间过大，Full GC的执行时间会变长，系统的响应性急剧下降；要么减小老年代的大小，Full GC消耗的时间变短，系统的响应性增强，但是Full GC发生的频率增多，系统的吞吐量下降（吞吐量=代码执行时间/（代码执行时间+垃圾回收时间））

三、只有一个survivor会怎样

如果刚开始survivor为空，新生代的对象都存放在Eden中，此时发生了一次minor gc 将Eden中存活的对象复制到survivor中，过了一段时间Eden满了，这

时候无论是Eden中还是survivor中都存在需要被回收的对象，此时又发生了一次minor gc，则survivor中便出现了空间碎片。

如果试图将Eden区存活的对象转移到survivor中，努力适应这种不连续的空间。但是不连续的空间会导致再分配大对象的时候，由于没有连续的空间来分配，会导致提前垃圾回收。

如果将survivor中的所有存活对象向下移动来消除碎片，然后将所有的存活对象移入其中。这样做会降低效率。

如果把两个区域中的所有存活对象都复制到完全独立的空间中，也就是第二块Survivor中，这样就可以留出一块完全空着的Eden和Survivor了，下次GC的时候再重复这个流程

四、两个survivor的情况

如果存在两个survivor区，当Eden区满的时候，发生minor gc，有存活对象，将对象转移到S0中，当下次再发生minor gc的时候，将Eden区和S0区的存活对象复制到S1中（这种复制算法可以保证S1中来自Eden和S0中对象的地址是连续的），清空Eden区和S0的空间，然后交换S0和S1的角色，之后发生minor gc时，循环往复。直到存活对象old enough，升入老年代。这种情况下我们可以保证始终有一个Survivor的空间是没有碎片的,而另外一个Survivor是空着的。

JVM垃圾收集原因、算法 qq_44417051的博客 34
JVM垃圾收集原因、算法 垃圾收集Garbage Collection 简称GC 为什么要垃圾收集和内...

关于Java垃圾回收算法解析 lj961030的博客 150
首先什么样的能称为“垃圾呢”：在jvm的运行时内存区域中，一般来说垃圾回收就是针对...

参与评论

请发表有价值的评论， 博客评论不欢迎灌水，良好的社区氛围需要大家 评论

JVM基础(三):垃圾回收机制_一只修仙的猿 3-14
在c/c++中,开发者对内存有至高无上的权利,同时也需要自己对对象负责到底:创建和释放,...

Jvm之垃圾收集器与内存分配策略(二)_Penguin 8-17
Java虚拟机大多都优先采用了这种收集算法去回收新生代 Appel式回收 概念: Appel式回...

深入探究JVM之垃圾回收器 夜勿语的博客 635
文章目录前言正文一、垃圾收集算法标记-复制标记-清除标记-整理分代回收二、常用的...

JVM垃圾回收机制及算法 weixin_43125854的博客 30
栈：栈中的生命周期是跟随线程，所以不是关注的重点 堆：堆中的对象是共享的，也是...

【深入理解Java虚拟机】第3章:垃圾收集算法_乾坤未定... 3-10
“Appel式回收”:把新生代划分为一块较大的Eden空间和两块较小的Survivor。每次使用Ed...

jvm之垃圾收集算法_Scan688的博客 8-20
Appel式回收的具体做法是把新生代分为一块较大的Eden空间和两块较小的 Survivor空间...

给我一首歌的让你深入探究JVM之垃圾回... 程序员麦冬的博客（公众号同名） 136
前言 JVM的自动内存管理得益于不断发展的垃圾回收器，从最初的单线程收集到现在并...

Python 内存管理与垃圾回收机制 weixin_42618303的博客 106
python python内存的泄露和溢出前言一、内存泄露和溢出二、垃圾回收机制1.引用计数...

JVM--垃圾收集器与内存分配策略--大致总结_一只爱学习... 3-2
(2)内存缩小了一半,空间浪费大(可用Appel式回收,Eden与两个Survivor区比例8:1:1)。3....

JVM垃圾收集 — Appel式回收为什么使用两个S... 最新发布 THMAIL的博客 18
Appel式回收为什么使用两个Survivor 一、Appel式回收 Appel式回收的具体做法是把新...

JVM三种垃圾回收算法 asdhkai的博客 113
不同的垃圾收集算法 “标记-复制算法”、“标...

wxcp perfect

关注