

飘扬的红领巾

[HOME](#)[CONTACT](#)[GALLERY](#)

深入理解JVM（五）——垃圾回收器

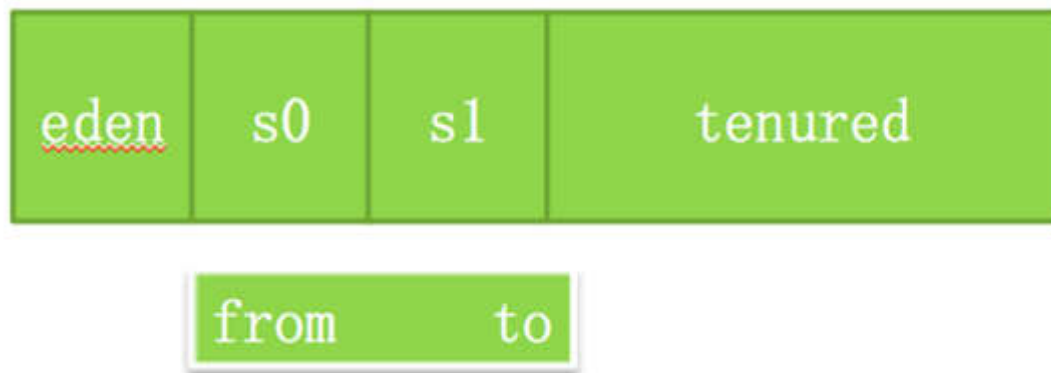
2017-08-21 09:46 by 飘扬的红领巾, 4108 阅读, 0 评论, 收藏, 编辑

轻松学习JVM（五）——垃圾回收器

上一篇我们介绍了常见的垃圾回收算法，不同的算法各有各的优缺点，在JVM中并不是单纯的使用某一种算法进行垃圾回收，而是将不同的垃圾回收算法包装在不同的垃圾回收器当中，用户可以根据自身的需求，使用不同的垃圾回收器，以便让自己的java程序性能到达最佳。

在介绍垃圾回收器之前，我们先回顾一下java堆的结构。

堆内存回顾



java堆内存结构包括：新生代和老年代，其中新生代由一个伊甸区和2个幸存区组成，2个幸存区是大小相同，完全对称的，没有任何差别。我们把它们称为S0区和S1区，也可

About



李平，目前在一家O2O互联网公司从事设计、开发工作。业余时间喜欢跑步、看书、游戏。

喜欢简单而高效的工作环境，熟悉JavaEE、SOA、数据库架构、优化、系统运维，有大型门户网站，金融系统建设经验。RHCE、MySQL OCP。MyCAT开源项目成员。

我的开源项目：

[mycat-eye](#)[nosql-eye](#)

昵称：[飘扬的红领巾](#)

园龄：[6年9个月](#)

荣誉：[推荐博客](#)

粉丝：[910](#)

关注：[0](#)

[+加关注](#)

SEARCH

最新评论

以称为from区和to区。

JVM的垃圾回收主要是针对以上堆空间的垃圾回收，当然其实也会针对元数据区（永久区）进行垃圾回收，在此我们主要介绍对堆空间的垃圾回收。

下面我们介绍几种垃圾回收器：

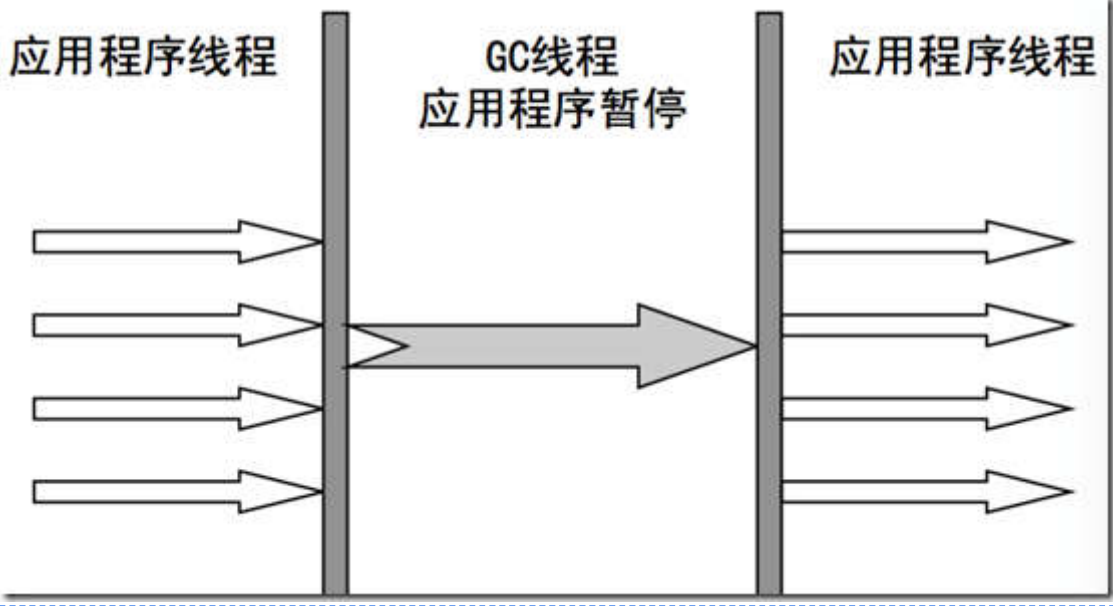
串行收集器

顾名思义，串行收集器就是使用单线程进行垃圾回收。对新生代的回收使用复制算法，对老年代使用标记压缩算法，这也和我们上一篇介绍的算法优势是相吻合的。

串行收集器是最古老最稳定的收集器，尽管它是串行回收，回收时间较长，但其稳定性是优于其他回收器的，综合来说是一个不错的选择。要使用串行收集器，可以在启动配置时加上以下参数：

```
-XX:+UseSerialGC
```

串行回收器的执行流程如下所示：



[Re:InnoDB一棵B+树可以存放多少行数据?](#)

好文 -- icycheng

[Re:深入理解JVM（八）——java堆分析](#)

@zhoumy 应该还有其他对象占用空间吧，比如这个类的一些元数据 -- xiaoli2333

[Re:大型网站的灵魂——性能](#)

mark -- xiaoli2333

[Re:深入理解JVM（七）——性能监控工具](#)

mark -- xiaoli2333

[Re:MySQL在并发场景下的问题及解决思路](#)

大神在吗，怎么联系你啊 -- duchaochen

日历							随笔档案	
< 2018年8月 >							2018年1月(2)	
日	一	二	三	四	五	六	2017年10月(1)	
29	30	31	1	2	3	4	2017年9月(4)	
5	6	7	8	9	10	11	2017年8月(7)	
12	13	14	15	16	17	18	2015年6月(1)	
19	20	21	22	23	24	25	2015年1月(2)	
26	27	28	29	30	31	1	2014年10月(2)	
2	3	4	5	6	7	8	2014年9月(2)	
我的标签							2014年5月(1)	
Maven(3)							2014年3月(2)	
Jenkins(2)							2014年1月(1)	
Nexus(2)							2013年9月(1)	
Sonar(2)							2013年8月(2)	
Svn(2)							2013年5月(1)	
Tomcat(2)							2013年4月(1)	
							2013年3月(1)	

执行垃圾回收时，应用程序线程暂停，GC线程开始（开始垃圾回收），垃圾回收完成后，应用程序线程继续执行。注意：在GC线程运行过程中使用单线程进行串行回收。

并行回收器

并行回收器你可能已经猜到就是使用多线程并行回收，不过这里需要注意的是，针对新生代和老年代，是否都使用并行，有不同的回收器选择：

1、 ParNew回收器

这个回收器只针对新生代进行并发回收，老年代依然使用串行回收。回收算法依然和串行回收一样，新生代使用复制算法，老年代使用标记压缩算法。在多核条件下，它的性能显然优于串行回收器，如果要使用这种回收器，可以在启动参数中配置：

```
-XX:+UseParNewGC
```

如果要进一步指定并发的线程数，可以配置一下参数：

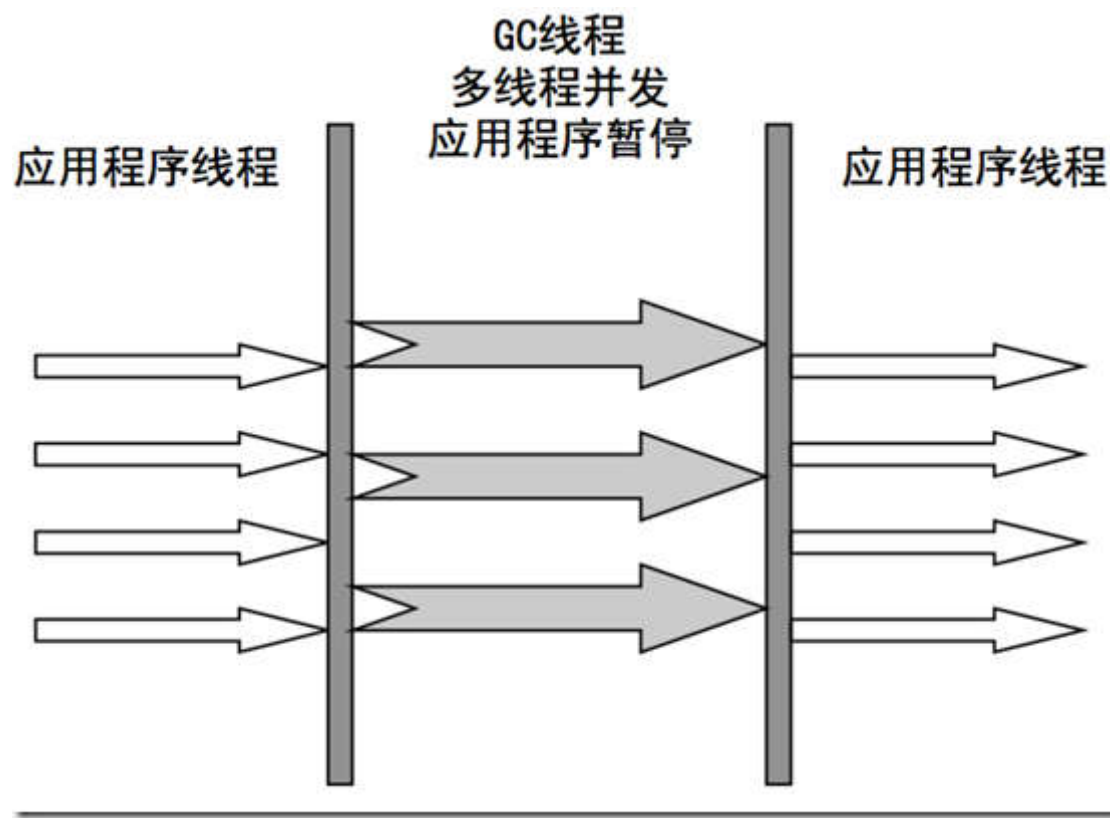
```
-XX:ParallelGCThreads
```

ParNew回收器的流程如下图所示：

并发(1)	2012年12月(1)
并发 乐观锁 悲观锁(1)	2012年11月(1)
大型网站(1)	2012年9月(1)
代码质量 Checkstyle PMD JDepend Eclemma Metric(1)	2012年6月(2)
更多	2012年5月(4)
	2012年3月(1)

随笔分类

Apache Mina(1)
Eclipse(1)
Hibernate(2)
Java(19)
JVM(8)
MongoDB(2)
MySQL(4)
RCP/SWT/Jface(1)
SOA(1)
Spring(3)
持续集成(4)
大型网站(3)
多线程(1)
开源项目(2)
敏捷(1)
其他(7)
设计模式(1)
数据结构/算法(1)
系统架构(3)
支付(1)



在进行垃圾回收时应用程序线程依然被暂停，GC线程并行开始执行垃圾回收，垃圾回收完成后，应用程序线程继续执行。

2、Parallel回收器

依然是并行回收器，但这种回收器有两种配置，一种类似于ParNEW：新生代使用并行回收、老年代使用串行回收。它与ParNew的不同在于它在设计目标上更重视吞吐量，可以认为在相同的条件下它比ParNew更优。要使用这种回收器可以在启动程序中配置：

```
-XX:+UseParallelGC
```

Parallel回收器另外一种配置则不同于ParNew，对于新生代和老年代均适应并行回收，要使用这种回收器可以在启动程序中配置：

重构(1)

推荐排行榜

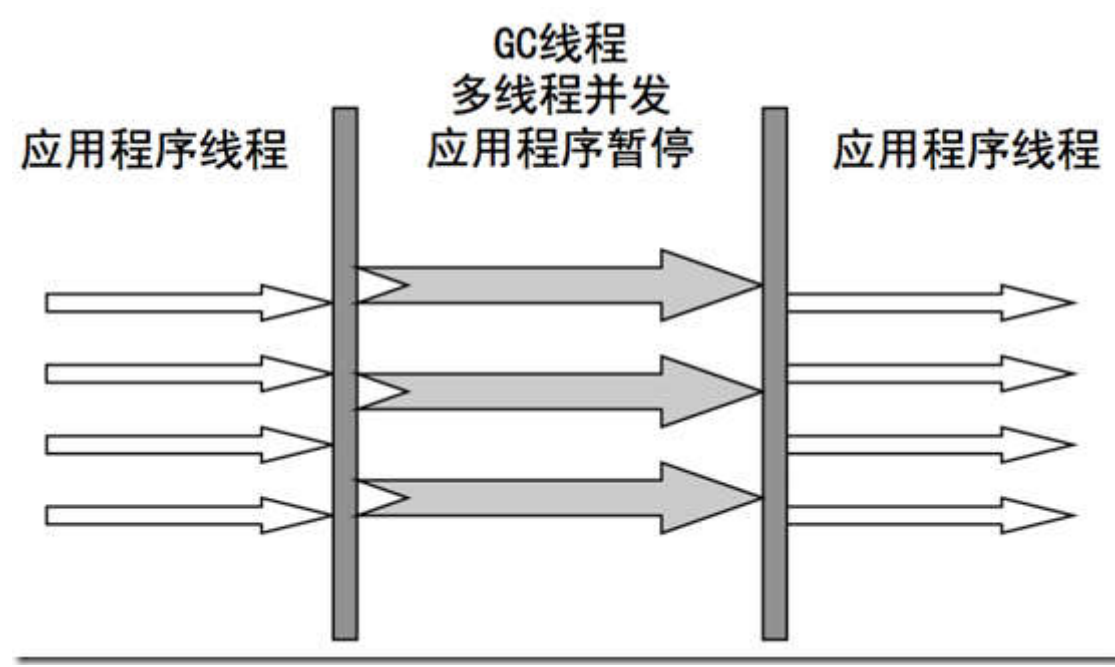
1. 大型网站系统架构的演化(211)
2. 大型网站的灵魂——性能(63)
3. 电商系统中的商品模型的分析与设计——续(51)
4. 电商系统中的商品模型的分析与设计(47)
5. 做了两款数据库监控工具，打算在近期开源(39)

阅读排行榜

1. 大型网站系统架构的演化(51203)
2. 深入理解JVM（一）——基本原理(34609)
3. 电商系统中的商品模型的分析与设计(16808)
4. 大型网站的灵魂——性能(15776)
5. 使用Maven+Nexus+Jenkins+Svn+Tomcat+Sonar搭建持续集成环境（一）(15207)

XX:+UseParallelOldGC

Parallel回收器的流程和ParNew的流程是一致的：



在进行回收时，应用程序暂停，GC使用多线程并发回收，回收完成后应用程序线程继续运行。

CMS回收器

CMS回收器：Concurrent Mark Sweep，并发标记清除。注意这里注意两个词：并发、标记清除。

并发表示它可以与应用程序并发执行、交替执行；标记清除表示这种回收器不是使用的是标记压缩算法，这和前面介绍的串行回收器和并发回收器有所不同。需要注意的是CMS回收器是一种针对老年代的回收器，不对新生代产生作用。这种回收器优点在于减少了

应用程序停顿的时间，因为它不需要应用程序完成暂定等待垃圾回收，而是与垃圾回收并发执行。要执行这种垃圾回收器可以在启动参数中配置：

```
-XX:+UseConcMarkSweepGC
```

CMS回收机运行机制非常复杂，我们简单的将他的运行流程分为以下几步：

初始标记

标记从GC Root可以直接可达的对象；

并发标记（和应用程序线程一起）

主要标记过程，标记全部对象；

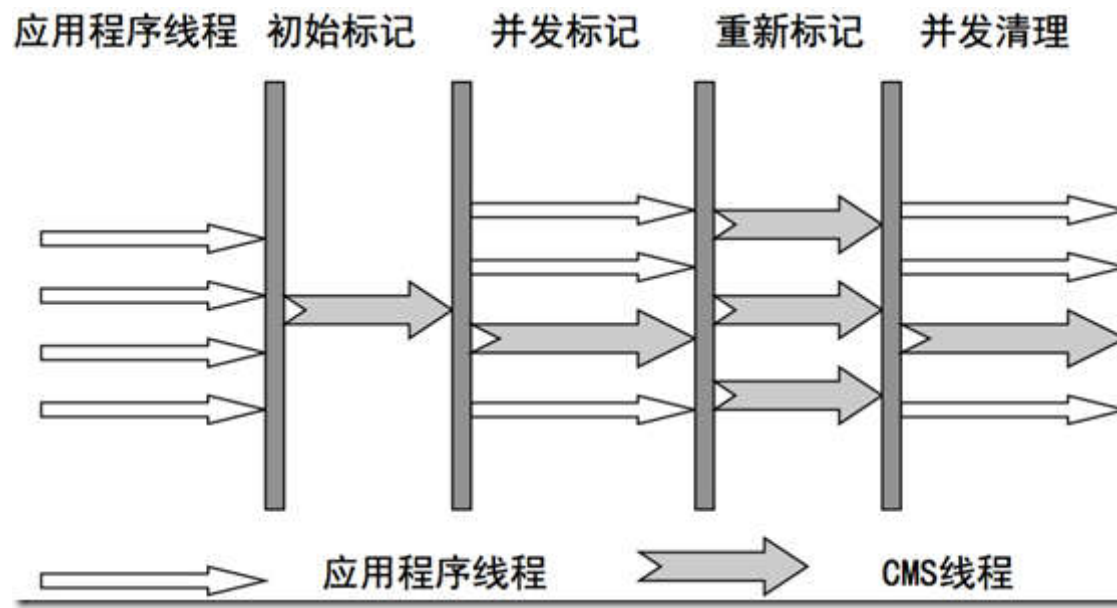
重新标记

由于并发标记时，用户线程依然运行，因此在正式清理前，再做依次重新标记，进行修正。

并发清除（和用户线程一起）

基于标记结果，直接清理对象。

流程如下图所示：



从上图可以看到标记过程分三步：初始标记、并发标记、重新标记，并发标记是最主要的标记过程，而这个过程是并发执行的，可以与应用程序线程同时进行，初始标记和重新标记虽然不能和应用程序并发执行，但这两个过程标记速度快，时间短，所以对应用程序不会产生太大的影响。最后并发清除的过程，也是和应用程序同时进行的，避免了应用程序的停顿。

CMS的优点显而易见，就是减少了应用程序的停顿时间，让回收线程和应用程序线程可以并发执行。但它也不是完美的，从他的运行机制可以看出，因为它不像其他回收器一样集中一段时间对垃圾进行回收，并且在回收时应用程序还是运行，因此它的回收并不彻底。这也导致了CMS回收的频率相较于其他回收器要高，频繁的回收将影响应用程序的吞吐量。

G1回收器

G1回收器是jdk1.7以后推出的回收器，试图取代CMS回收器。

不同于其他的回收器、G1将堆空间划分成了互相独立的区块。每块区域既有可能属于老年代、也有可能是新生代，并且每类区域空间可以是不连续的（对比CMS的老年代和新生代都必须是连续的）。这种将老年代区划分成多块的理念源于：当并发后台线程寻找可回收的对象时、有些区块包含可回收的对象要比其他区块多很多。虽然在清理这些区块时G1仍然需要暂停应用线程、但可以用相对较少的时间优先回收包含垃圾最多区块。这也是为什么G1命名为Garbage First的原因：第一时间处理垃圾最多的区块。要使用G1回收器需要在启动是配置以下参数：

`-XX:+UseG1GC`

G1相对CMS回收器来说优点在于：

- 1、因为划分了很多区块，回收时减小了内存碎片的产生；
- 2、G1适用于新生代和老年代，而CMS只适用于老年代。

小结

本文简要介绍了JVM中的垃圾回收器，主要包括串行回收器、并行回收器以及CMS回收器、G1回收器。他们各自都有优缺点，通常来说你需要根据你的业务，进行基于垃圾回收器的性能测试，然后再做选择。下面给出配置回收器时，经常使用的参数：

`-XX:+UseSerialGC`：在新生代和老年代使用串行收集器

`-XX:+UseParNewGC`：在新生代使用并行收集器

`-XX:+UseParallelGC`：新生代使用并行回收收集器，更加关注吞吐量

`-XX:+UseParallelOldGC`：老年代使用并行回收收集器

`-XX:ParallelGCThreads`：设置用于垃圾回收的线程数

`-XX:+UseConcMarkSweepGC`：新生代使用并行收集器，老年代使用CMS+串行收集器

`-XX:ParallelCMSThreads`：设定CMS的线程数量

-XX:+UseG1GC: 启用G1垃圾回收器



本文基于署名 2.5 中国大陆许可协议发布，欢迎转载，演绎或用于商业目的，但是必须保留本文的署名李平（包含链接），具体操作方式可参考此处。如您有任何疑问或者授权方面的协商，请给我留言。

[好文要顶](#)[关注我](#)[收藏该文](#)

飘扬的红领巾

关注 - 0

粉丝 - 910

0

0

荣誉: [推荐博客](#)

[+加关注](#)

« 上一篇: [深入理解JVM（四）——垃圾回收算法](#)

» 下一篇: [深入理解JVM（六）——类加载器原理](#)

分类: [Java](#), [JVM](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。



最新IT新闻：

- OpenBSD停用英特尔处理器的超线程功能
 - 2018年最受欢迎的VS Code扩展插件合集
 - 抖音视频热搜榜上线，“眉毛怪”登榜首
 - 阿里发布第一季财报，开新公司整合饿了么口碑业务
 - 荣耀Note 10是一台应该「大材小用」的手机 | 验货
- » 更多新闻...



最新知识库文章：

- 一个故事看懂“区块链”
 - 被踢出去的用户
 - 成为一个有目标的学习者
 - 历史转折中的“杭派工程师”
 - 如何提高代码质量？
- » 更多知识库文章...