

[您还未登录！登录](#)



[论坛首页](#) → [Java企业应用论坛](#) →

了解CMS(Concurrent Mark-Sweep)垃圾回收器

全部 [Hibernate](#) [Spring](#) [Struts](#) [iBATIS](#) [企业应用](#) [Lucene](#) [SOA](#) [Java综合](#) [Tomcat](#) [设计模式](#) [OO](#) [JBoss](#)
浏览 22927 次

锁定老帖子 主题: [了解CMS\(Concurrent Mark-Sweep\)垃圾回收器](#)
精华帖 (0) :: 良好帖 (0) :: 新手帖 (0) :: 隐藏帖 (0)

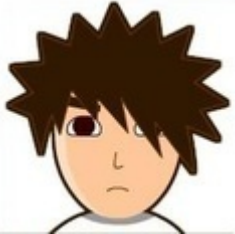
作者

正文

发表时间: 2012-01-01 最后修改: 2012-01-02

相关推荐: _

- ol_beta
- 等级: ☆



- 性别: ♂
- 文章: 150
- 积分: 110
- 来自: 天津
- 我现在离线

- [CMS gc实践总结](#)
- [JVM内存管理：深入垃圾收集器与内存分配策略](#)
- [作为java玩家应该了解的东西\(Java内存管理——垃圾收集概念及特点\)](#)
- [JVM 内存模型和垃圾回收（四）： 并发回收\(Concurrent Mark-Sweep Collector\)](#)
- [JVM --并发垃圾回收器CMS](#)
- [JVM垃圾回收CMS](#)
- [CMS垃圾收集器介绍](#)
- [JVM垃圾回收器常用参数介绍](#)

推荐群组: [高级语言虚拟机](#)

[更多相关推荐](#)

[Java](#) [JVM](#) [CMS](#) [垃圾回收](#) [jc](#)

1.总体介绍:

锁定老帖子 主题: [了解CMS\(Concurrent Mark-Sweep\)垃圾回收器](#)**精华帖 (0) :: 良好帖 (0) :: 新手帖 (0) :: 隐藏帖 (0)**

作者

正文

CMS(Concurrent Mark-Sweep)是以牺牲吞吐量为代价来获得最短回收停顿时间的垃圾回收器。对于要求服务器响应速度的应用上, 这种垃圾回收器非常适合。在启动JVM参数加上-XX:+UseConcMarkSweepGC, 这个参数表示对于老年代的回收采用CMS。CMS采用的基础算法是: 标记—清除。

2.CMS过程:

- 初始标记(STW initial mark)
- 并发标记(Concurrent marking)
- 并发预清理(Concurrent precleaning)
- 重新标记(STW remark)
- 并发清理(Concurrent sweeping)
- 并发重置(Concurrent reset)

初始标记：在这个阶段, 需要虚拟机停顿正在执行的任务, 官方的叫法STW(Stop The Word)。这个过程从垃圾回收的"根对象"开始, 只扫描到能够和"根对象"直接关联的对象, 并作标记。所以这个过程虽然暂停了整个JVM, 但是很快就完成了。

并发标记：这个阶段紧随初始标记阶段, 在初始标记的基础上继续向下追溯标记。并发标记阶段, 应用程序的线程和并发标记的线程并发执行, 所以用户不会感受到停顿。

并发预清理：并发预清理阶段仍然是并发的。在这个阶段, 虚拟机查找在执行并发标记阶段新进入老年代的对象(可能会有一些对象从新生代晋升到老年代, 或者有一些对象被分配到老年代)。通过重新扫描, 减少下一个阶段"重新标记"的工作, 因为下一个阶段会Stop The World。

重新标记：这个阶段会暂停虚拟机, 收集器线程扫描在CMS堆中剩余的对象。扫描从"跟对象"开始向下追溯, 并处理对象关联。

并发清理：清理垃圾对象, 这个阶段收集器线程和应用程序线程并发执行。

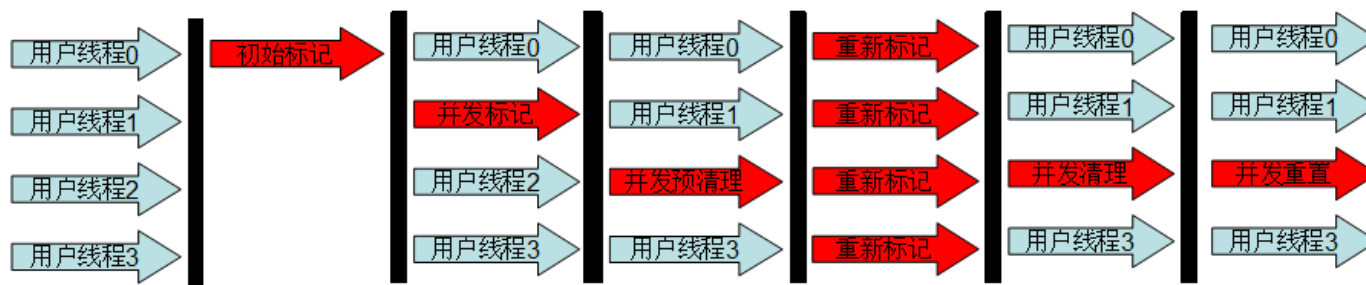
并发重置：这个阶段, 重置CMS收集器的数据结构, 等待下一次垃圾回收。

锁定老帖子 主题: 了解CMS(Concurrent Mark-Sweep)垃圾回收器**精华帖 (0) :: 良好帖 (0) :: 新手帖 (0) :: 隐藏帖 (0)**

作者

正文

CSM执行过程:



3.CMS缺点

- CMS回收器采用的基础算法是Mark-Sweep。所有CMS不会整理、压缩堆空间。这样就会有一个问题：经过CMS收集的堆会产生空间碎片。CMS不对堆空间整理压缩节约了垃圾回收的停顿时间，但也带来的堆空间的浪费。为了解决堆空间浪费问题，CMS回收器不再采用简单的指针指向一块可用堆空间来为下次对象分配使用。而是把一些未分配的空间汇总成一个列表，当JVM分配对象空间的时候，会搜索这个列表找到足够大的空间来hold住这个对象。
- 需要更多的CPU资源。从上面的图可以看到，为了让应用程序不停顿，CMS线程和应用程序线程并发执行，这样就需要有更多的CPU，单纯靠线程切换是不靠谱的。并且，重新标记阶段，为空保证STW快速完成，也要用到更多的甚至所有的CPU资源。当然，多核多CPU也是未来的趋势！
- CMS的另一个缺点是它需要更大的堆空间。因为CMS标记阶段应用程序的线程还是在执行的，那么就会有堆空间继续分配的情况，为了保证在CMS回收完堆之前还有空间分配给正在运行的应用程序，必须预留一部分空间。也就是说，CMS不会在老年代满的时候才开始收集。相反，它会尝试更早的开始收集，已避免上面提到的情况：在回收完成之前，堆没有足够空间分配！默认当老年代使用68%的时候，CMS就开始行动了。 - XX:CMSInitiatingOccupancyFraction =n 来设置这个阈值。

总得来说，CMS回收器减少了回收的停顿时间，但是降低了堆空间的利用率。

4.啥时候用CMS

锁定老帖子 主题: [了解CMS\(Concurrent Mark-Sweep\)垃圾回收器](#)**精华帖 (0) :: 良好帖 (0) :: 新手帖 (0) :: 隐藏帖 (0)**

作者

正文

如果你的应用程序对停顿比较敏感，并且在应用程序运行的时候可以提供更大的内存和更多的CPU(也就是硬件牛逼)，那么使用CMS来收集会给你带来好处。还有，如果在JVM中，有相对较多存活时间较长的对象(老年代比较大)会更适合使用CMS。

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。

推荐链接

[返回顶楼](#)

[论坛首页](#) → [Java企业应用版](#)跳转论坛: ▼

- [首页](#)
- [资讯](#)
- [精华](#)
- [论坛](#)
- [问答](#)
- [博客](#)
- [专栏](#)
- [群组](#)
- [搜索](#)

- [广告服务](#)
- [ITeye黑板报](#)
- [联系我们](#)
- [友情链接](#)

© 2003-2018 ITeye.com. [[京ICP证070598号](#) 京公网安备11010502027441]

北京创新乐知信息技术有限公司 版权所有