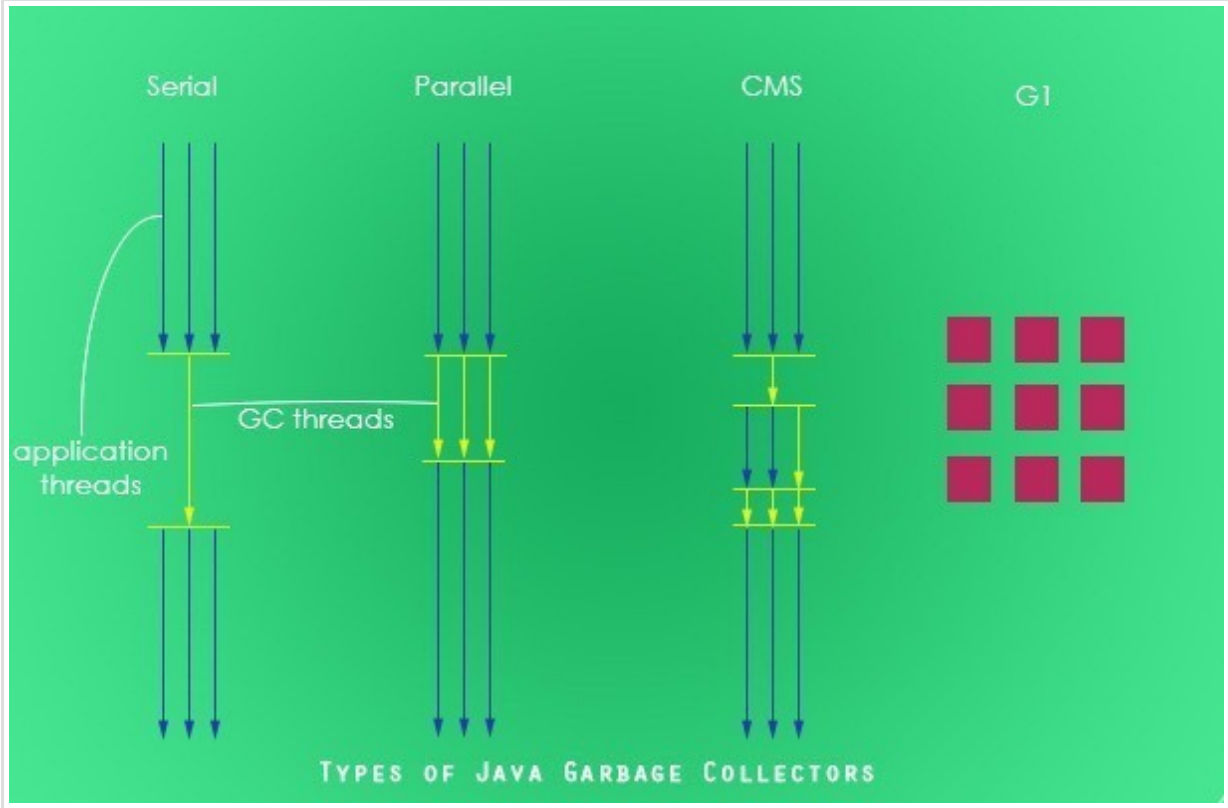


# 细述 Java垃圾回收机制→Types of Java Garbage Collectors

本文非原创，翻译自[Types of Java Garbage Collectors](#)  
在Java中为对象分配和释放内存空间都是由垃圾回收线程自动执行完成的。和C语言不一样的是Java程序员不需要手动写垃圾回收相关的代码。这是使得Java如此流行，同时也是Java能帮助程序员写出更好的Java应用的优点之一。

本文将会介绍各种不同类型的Java垃圾回收器。垃圾回收是Java用来将程序员从分配和释放内存的琐事中解放出来的自动过程。Java有四种类型的垃圾回收器，

- 1. Serial Garbage Collector
- 2. Parallel Garbage Collector
- 3. CMS Garbage Collector
- 4. G1 Garbage Collector



这四种类型的垃圾回收器都有各自的优

点和缺点。最重要的是程序员可以选择JVM使用哪种类型的垃圾回收器。我们可以通过传递不同的JVM参数来设置使用哪一个。各个垃圾回收器在不同应用场景下的效率会有很大的差异。因此了解各种不同类型的垃圾回收器以及它们的应用场景是非常重要的。

## Serial Garbage Collector

串行垃圾回收器控制所有的应用线程。它是为单线程场景设计的，只使用一个线程来执行垃圾回收工作。它暂停所有应用线程来执行垃圾回收工作的方式不适用于服务器的应用环境。它最适用的是简单的命令行程序。

使用-XX:+UseSerialGCJVM参数来开启使用串行垃圾回收器。

## Parallel Garbage Collector

并行垃圾回收器也称作基于吞吐量的回收器。它是JVM的默认垃圾回收器。与Serial不同的是，它使用多个线程来执行垃圾回收工作。和Serial回收器一样，它在执行垃圾回收工作是也需要暂停所有应用线程。

## CMS Garbage Collector

并发标记清除(Concurrent Mark Sweep,CMS)垃圾回收器，使用多个线程来扫描堆内存并标记可被清除的对象，然后清除标记的对象。CMS垃圾回收器只在下面这两种情形下暂停工作线程，

- 1. 在老年代中标记引用对象的时候
- 2. 在做垃圾回收的过程中堆内存中有变化发生

对比与并行垃圾回收器，CMS回收器使用更多的CPU来保证更高的吞吐量。如果我们可以有更多的CPU用来提升性能，那么CMS垃圾回收器是比并行回收器更好的选择。

使用-XX:+UseParNew GCJVM参数来开启使用CMS垃圾回收器。

# G1 Garbage Collector

G1垃圾回收器应用于大的堆内存空间。它将堆内存空间划分为不同的区域，对各个区域并行地做回收工作。G1在回收内存空间后还立即堆空闲空间做整合工作以减少碎片。CMS却是在全部停止(stop the world,STW)时执行内存整合工作。对于不同的区域G1根据垃圾的数量决定优先级。

使用-XX:UseG1GCJVM参数来开启使用G1垃圾回收器。

## Java 8 的优化

在使用G1垃圾回收器是，开启使用-XX:+UseStringDeduplacatonJVM参数。它会通过把重复的String值移动到同一个char[]数组来优化堆内存占用。这是Java 8 u 20引入的选项。

以上给出的四个Java垃圾回收器，在什么时候使用哪一个去决于应用场景，硬件配置和吞吐量要求。

# Garbage Collection JVM Options

下面是些主要的与Java垃圾回收相关的JVM选项。

### Type of Garbage Collector to run

选项	描述
-XX:+UseSerialGC	串行垃圾回收器
-XX:+UseParallelGC	并行垃圾回收器
-XX:+UseConcMarkSw eepGC	CMS垃圾回收器
-XX:ParallesCMSThread=	CMS垃圾回收器–使用的线程数量
-XX:UseG1GC	G1垃圾回收器

## GC 优化选项

选项	描述
-Xms	初始堆内存大小
-Xmx	最大堆内存大小
-Xmn	年轻代的大小
-XX:PermSize	初始永久代的大小
-XX:MaxPermSize	最大的永久代的大小

### Example Usage of JVM GC Options

```
java -Xmx12m -Xms3m -Xmn1m -XX:PermSize=20m -XX:MaxPermSize=20m -XX:+UseSerialGC -jar java-application.jar
```