

## Serial收集器

Serial（串行）垃圾收集器是最基本、发展历史最悠久的收集器；

JDK1.3.1前是HotSpot新生代收集的唯一选择；、

1、特点

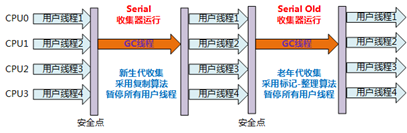
（1）针对新生代；

（2）采用复制算法；

（3）单线程收集；

（4）进行垃圾收集时，必须暂停所有工作线程，直到完成；

即会"Stop The World"；

Serial/Serial Old组合收集器运行示意图如下：

## ParNew收集器

ParNew垃圾收集器是Serial收集器的多线程版本。

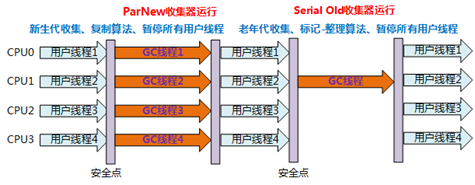
1、特点

除了多线程外，其余的行为、特点和Serial收集器一样；

如Serial收集器可用控制参数、收集算法、Stop The World、内存分配规则、回收策略等；

两个收集器共用了不少代码；

ParNew/Serial Old组合收集器运行示意图如下：



## Parallel Scavenge收集器

Parallel Scavenge垃圾收集器因为与吞吐量关系密切，也称为吞吐量收集器（Throughput Collector）。

1、特点

（1）新生代收集器；

（2）采用复制算法；

（3）多线程收集；

（4）主要特点是：它的关注点与其他收集器不同

CMS等收集器的关注点是尽可能地缩短垃圾收集时用户线程的停顿时间；

而Parallel Scavenge收集器的目标则是达一个可控制的吞吐量（Throughput）；

注：吞吐量：CPU用于用户代码的时间/CPU总消耗时间的比值，即=运行用户代码的时间/(运行用户代码时间+垃圾收集时间)。比如，虚拟机总共运行了100分钟，其中垃圾收集花掉1分钟，那吞吐量就是99%。

## Serial Old收集器

Serial Old是 Serial收集器的老年代版本；

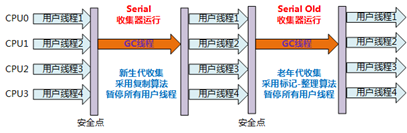
1、特点

（1）针对老年代；

（2）采用"标记-整理"算法（还有压缩，Mark-Sweep-Compact）；

（3）单线程收集；

Serial/Serial Old收集器运行示意图如下：



## Parallel Old收集器

Parallel Old垃圾收集器是Parallel Scavenge收集器的老年代版本；

JDK1.6中才开始提供；

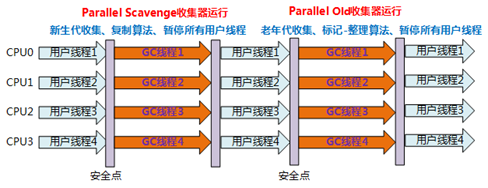
1、特点

（1）针对老年代；

（2）采用"标记-整理"算法

（3）多线程收集；

Parallel Scavenge/Parallel Old收集器运行示意图如下：



## CMS收集器

并发标记清理（Concurrent Mark Sweep，CMS）收集器也称为并发低停顿收集器（Concurrent Low Pause Collector）或低延迟（low-latency）垃圾收集器；

在前面ParNew收集器曾简单介绍过其特点；

此外CMS只能与ParNew一起使用。

1、特点

（1）针对老年代；

（2）基于"标记-清除"算法(不进行压缩操作，产生内存碎片)；

（3）以获取最短回收停顿时间为目标；

（4）并发收集、低停顿；

（5）需要更多的内存（看后面的缺点）；

（6）是HotSpot在JDK1.5推出的第一款真正意义上的并发（Concurrent）收集器；

第一次实现了让垃圾收集线程与用户线程（基本上）同时工作；

## G1收集器

G1（Garbage-First）是JDK7-u4才推出商用的收集器；

1、特点

（1）并行与并发

能充分利用多CPU、多核环境下的硬件优势；

可以并行来缩短"Stop The World"停顿时间；

也可以并发让垃圾收集与用户程序同时进行；

（2）分代收集，收集范围包括新生代和老年代

能独立管理整个GC堆（新生代和老年代），而不需要与其他收集器搭配；

能够采用不同方式处理不同时期的对象；

虽然保留分代概念，但Java堆的内存布局有很大差别；

将整个堆划分为多个大小相等的独立区域（Region）；

新生代和老年代不再是物理隔离，它们都是一部分Region（不需要连续）的集合；