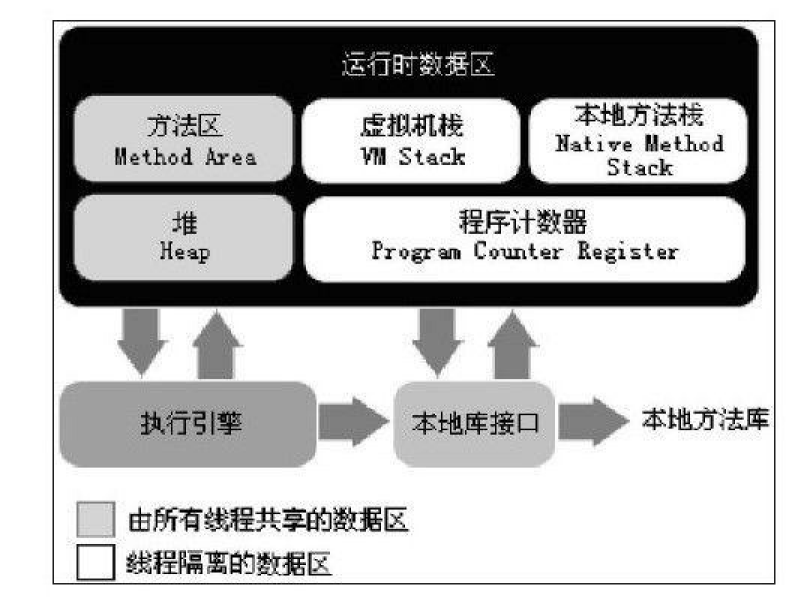
JVM学习笔记



JVM运行时数据区的调用关系

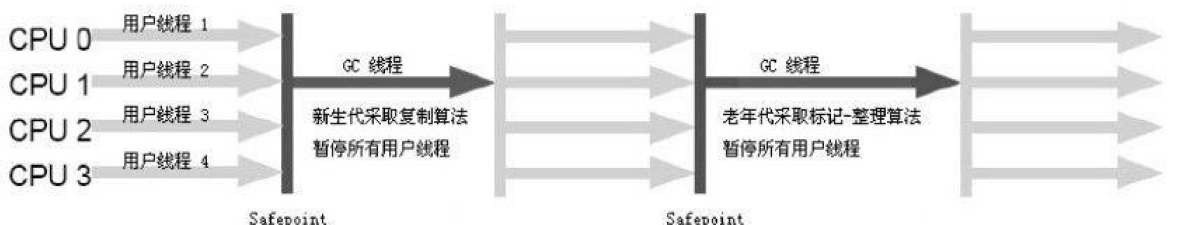
CAS （compare and swap） 如果内存位置的值与预期原值相匹配，那么处理器会自动将该位置值更新为新值。否则，处理器不做任何操作。

垃圾收集器（GC）

Serial收集器（JDK 1.3.1之前是唯一新生代的收集选择）

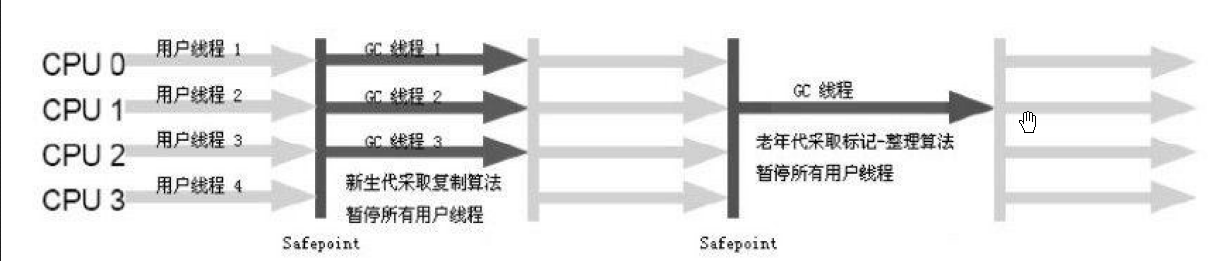
是一个单线程的收集器,在收集线程的过程中，会暂停所有其他工作线程，直到它收集结束。“Stop The World”暂停其他线程是虚拟机在后台自动发起和自动完成的，在其工作中用户是不可见的。它是虚拟机运行在Client模式下默认的新生代收集器，优点是简单而高效（与其他收集器的单线程比）对于限定单个CPU的环境来说，Serial收集器不会再线程交互上有开销，只要停顿时间不是很频繁发生，在Client模式下是一个比较好的选择。

工作过程如下：



ParNew收集器

ParNew收集器其实就是Serial收集器的多线程版本，除了使用多线程进行垃圾收集之外，其余行为几乎一样。它在Server模式下的虚拟机中以新生代收集器为首选，ParNew和Serial能支持CMS, ParNew收集器工作过程如下图



Parallel Scavenge 收集器

Parallel Scavenge 是一个新生代收集器，在ParNew的基础上，Parallel Scavenge 收集器的目标是达到一个可控制的吞吐量 ，Parallel Scavenge收集器提供两个参数用户精确控制吞吐量，分别是控制最大垃圾收集停顿时间的-XX:MaxGCPauseMillis 以及吞吐量大小的 –XX：GCTimeRatio

MaxGCPauseMillis 参数允许的值是一个大于

吞吐量公式

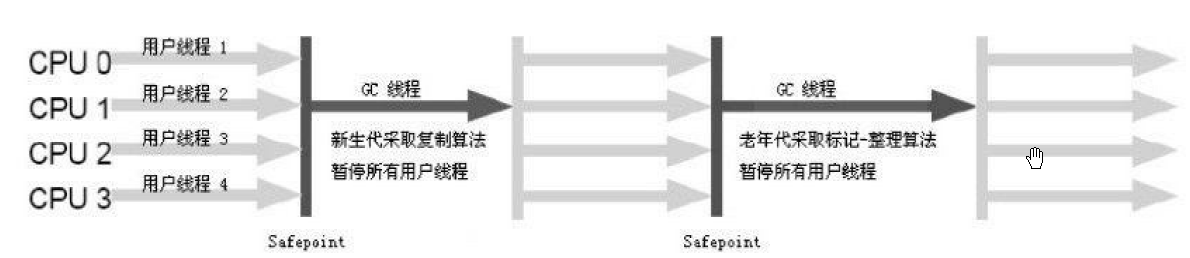
**吞吐量=运行用户代码时间/（运行用户代码时间+垃圾收集时间）**

**如 99%=99min/100min**

Serial Old收集器

Serial Old是Serial的收集器的老年代版本，同样是单线程收集器，使用“标记-整理”算法。 这个收集器**在于给Client模式下的虚拟机使用**。如果用在Server模式下，name它主要用途是：JDK1.5以及之前的版本用来与Parallel Scavenge收集器搭配使用，另一种用途就是作为CMS收集器的后备预案 ，在并发收集发生Concurrent Mode Failure时使用

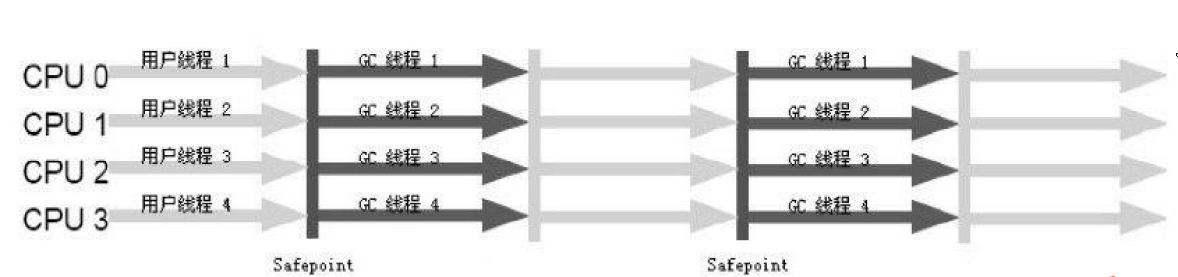
流程图如图



**Parallel Old收集器**

**是Parallel Scavenge 收集器的老年代版本，使用多线程和“标记-整理”算法。JDK1.6才开始提供的，Parallel Old 能与CMS组合使用，其组合使得“吞吐量优先”收集器终于有了比较名副其实的应用组合，在注重吞吐量以及CPU资源敏感的场合，都可以优先考虑Parallel Scavenge 加 Parallel Old收集器**

**工作过程如图**



**CMS收集器**

**CMS收集器是一种以获取最短回收停顿时间为目标的收集器。目前很大一部分的Java应用集中在互联网站或者B/S系统的服务端上。**

