GC事件的持续时间，通过三个部分来衡量:

user：表示GC线程所消耗的总CPU时间

sys：表示操作系统调用和系统等待事件所消耗的时间

real:表示应用程序实际暂停的时间

1. **串行GC**

1.1-XX：+UseSerialGC配置串行GC

1.2串行GC对年轻代使用mark-copy（标记-复制）算法，对老年代使用mark-sweep-compact（标记-清除-整理）算法。两者都是单线程的垃圾收集器，不能进行并行处理，所以都会触发全线暂停（STW），停止所有的应用线程。

因此这种GC算法不能充分利用多核CPU。不管有多少CPU内核，JVM在垃圾收集时都只能使用单个核心。CPU利用率高，暂停时间长。

1.3串行GC的垃圾回收影响三个区：Young区、Old区、Metaspace--元数据区

1.4逐步增加Xmx的值对比一下：

（1）256M:没有发生OOM，生成对象5150次，Full GC和Young GC都有发生;

（2）512M:生成对象7132次，Full GC和Young GC都有发生;

（3）1024M:生成对象9464次，没有发生Full GC；但是Young GC时，时间有明显增长，甚至达到0.1s，在512的时候，Full GC都没有这么长的时间；

（4）2048M：生成对象12180次，没有Full GC，且Young GC的时间更长。

并不是内存配置的越大越好，逐渐增大，效率提高，但垃圾回收时间也会变长。

1. **并行GC**

2.1配置并行GC

-XX：+UseParallelGC

-XX：+UseParallelOldGC

-XX：+UseParallelGC-XX:+UseParallelOldGC

-XX：ParallelGCThreads=N来指定GC线程数，其默认值为CPU核心数。

2.2并行GC在年轻代使用标记-复制（mark-copy）算法，在老年代使用标记-清除-整理（mark-sweepcompact）算法。年轻代和老年代的垃圾回收都会触发STW事件。

并行垃圾收集器适用于多核服务器，主要目标是增加吞吐量。因为对系统资源的有效使用，能达到更高的吞吐量:在GC期间，所有CPU内核都在并行清理垃圾，所以总暂停时间更短；在两次GC周期的间隔期，没有GC线程在运行，不会消耗任何系统资源。

2.3并行GC的垃圾回收影响三个区：Young区、Old区、Metaspace--元数据区

2.4逐步增加Xmx的值对比一下：

（1）256M：发生OOM，Full GC和Young GC都有发生；

（2）512M：没有发生OOM，生成对象7132次，Full GC和Young GC都有发生；

（3）1024M：生成对象8975次，有Full GC的发生。但是可以看到在进行Young GC的时候，时间有明显增长，甚至达到0.1s，在512的时候，Full GC都没有这么长的时间；

（4）2048M：生成对象10763次，没有Full GC，同样的，Young GC的时间更长。

随着内存配置的增加，系统调用时间也会增常，与串行GC相差不多。

1. **CMS**

##### 3.1-XX:+UseConcMarkSweepGC配置CMS GC

3.2CMS的目标是获取最短回收停顿时间的，基于标记清除的。分为6个阶段：InitialMark（初始标记）、ConcurrentMark（并发标记）、ConcurrentPreclean（并发预清理）、FinalRemark（最终标记）、ConcurrentSweep（并发清除）、ConcurrentReset（并发重置）；CMS在初始标记和最终标记的两个阶段会STW，会产生很多的内存碎片

3.3 CMS的垃圾回收：Young区、Metaspace--元数据区

1. **G1**

##### 4.1 -XX:+UseG1GC 配置G1 GC

4.2 G1 GC不再坚持固定大小以及固定数量的分代区域划分，而是分成大小相等的区域，也就是Region，每个区域在不同的时间可以扮演不同的角色。大对象放在特殊的区域Humongous区域，

4.3 CMS的垃圾回收：Young区、Mixed混合区

4.4逐步增加Xmx的值对比：

（1）256M：OOM

（2）512M：生成对象6004次

（3）1024M：生成对象8299次

（4）2048M：生成对象9191次

串行GC简单高效，是所有收集器里面内存消耗最小的；对于单核和核数比较少的，可以获得更高的单线程收集效率。

并行GC以高吞吐量为目标，即减少垃圾收集时间，让用户代码获得更长的运行时间。

CMS GC一种以获得最短停顿时间为目标的收集器，适合追求垃圾收集速度的服务器，是老年代垃圾收集器。

G1一种以获得最短停顿时间为目标的收集器，适合追求垃圾收集速度的服务器，是老年代垃圾收集器。