1. 不同GC在512M,1G,4G内存下的执行次数:

图表, 条形图

描述已自动生成

整体来看 G1收集器>CMS收集器>Parallel收集器>Serial收集器。

内存在256M 四个垃圾收集器均出现OOM异常

在512M内存下，G1>Serial=Parallel>CMS。初步判断为在内存有限的情况下，G1略好于其他三种，并行在此情况下并未体现出相对于串行的优势。而CMS进行老年代的GC步骤略多于串行，并行，则其执行次数最低。

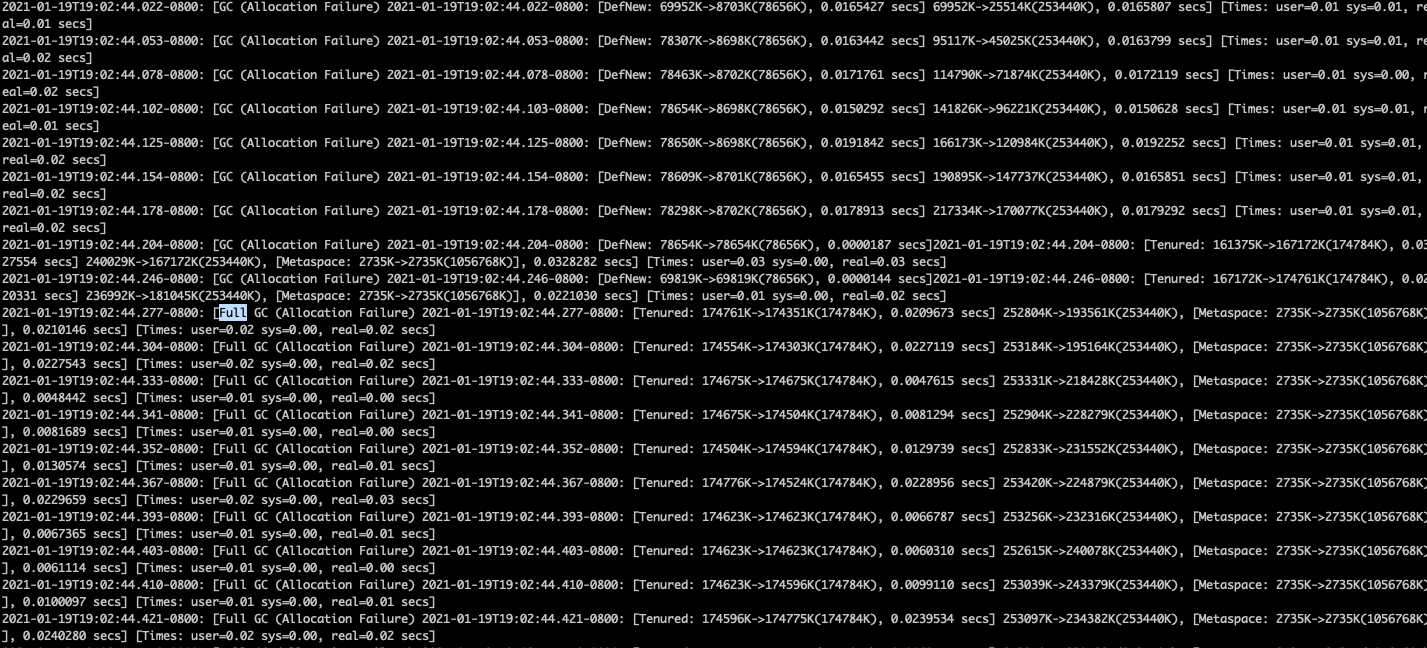
在1G内存下，与512M的内存下执行次数横向对比，均有不同程度的提升。其中CMS依旧处于执行次数较低的地位。

在4G内存在，由于内存足够，则可以看出在执行次数上G1依旧最优，CMS在大内存的执行上展现其优势，高于串行以及并行。并行在4G内存的执行下，执行次数也优串行。但是在4G的内存下执行 串行，并行，G1反而不如在1G内存下执行次数多。初步考虑是因为因为Young区内存增大，导致YoungGC耗时增加。

1. 详细分析在不同

串行收集器：

在256M内存下执行

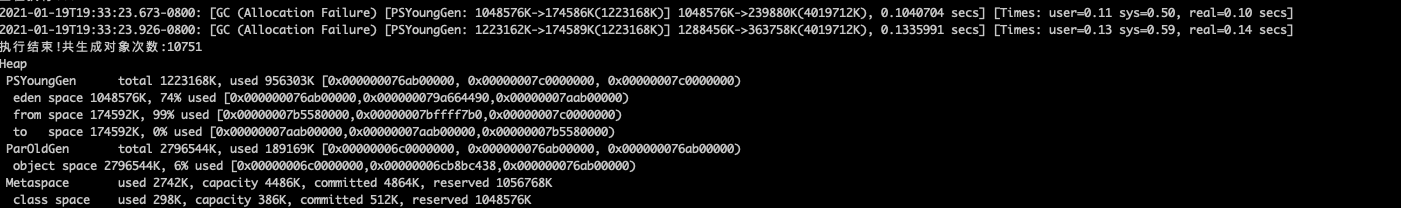


起初YoungGc回收正常，年轻代内存回收内存大小略多于整体内存回收大小，有一部分内存成功进入了老年代。但是在进行了几次GC 之后 年轻代的垃圾回收并未会回收掉任何内存，但是整体的堆内存还是在增长的，说明部分对象创建后直接进入了年老代，同时经过FullGC并未回收掉任何内存，直至最终内存溢出。

由于是串行垃圾回收在最后STW时间占比过大，导致系统实际执行时间其实并没有多少。

并行回收期:

在4G内存下

未产生FullGC，YuongGC回收的内存有一部分进入了老年代。但是未触发FullGC。

在1G内存下：

文本

描述已自动生成

在经过5次左右的yuongGC后触发FullGC，在FullGC时，直接清空掉了年轻代的所有对象，转移进入了老年代。