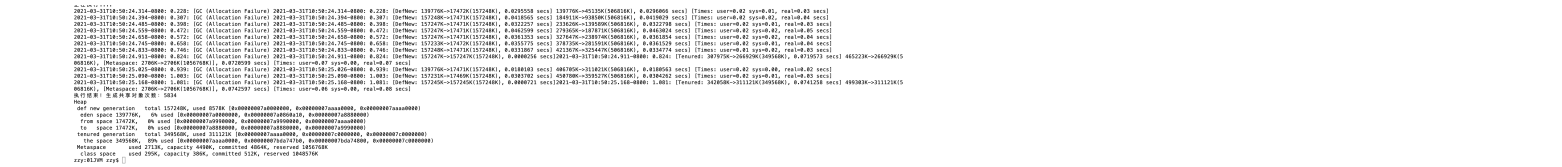
环境 JDK 1.8

1、串行GC

-Xmx 512m -Xms512



分析：

第一次Minor GC

```
2021-03-31T10:50:24.314-0800: 0.228: [GC (Allocation Failure) 2021-03-31T10:50:24.314-0800: 0.228: [DefNew: 139776K->17472K(157248K), 0.0295558 secs] 139776K->45135K(506816K), 0.0296066 secs] [Times: user=0.02 sys=0.01, real=0.03 secs]
```

第一条GC发生时间在程序启动后的0.228秒，是一次Minor GC，并且发生原因是Allocation Failure，表明本次引起GC的原因是因为在年轻代中没有足够的空间能够存储新的数据了。
本次Minor GC共回收掉了年轻代 139776K - 17472K = 122307K（约120M）的容量，耗时为0.0295558 secs。
同时堆内存从139776K 降低到 45135k，回收掉 94641K（约92.5M）。122307K - 94641K = 27666k（约27M），耗时为0.0296066 secs
本次总共有27M的数据从年轻代晋升到了老年代。
本次GC用户耗时为user=0.02秒，内核耗时为sys=0.01秒，总耗时为real=0.03秒。

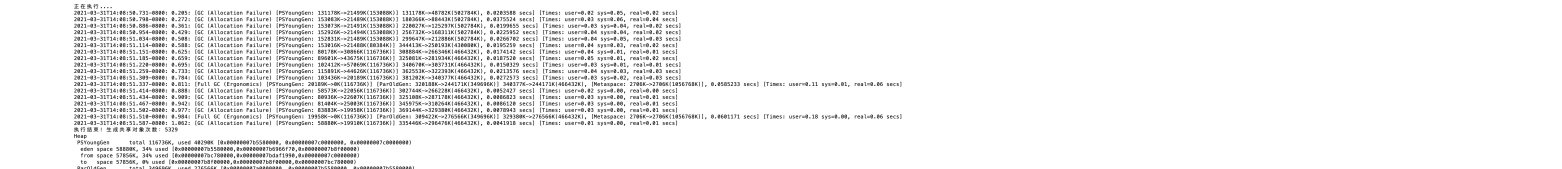
Minor GC

```
2021-03-31T10:50:24.911-0800: 0.824: [GC (Allocation Failure) 2021-03-31T10:50:24.911-0800: 0.824: [DefNew: 157247K->157247K(157248K), 0.0000256 secs]2021-03-31T10:50:24.911-0800: 0.824: [Tenured: 307975K->266929K(349568K), 0.0719573 secs] 465223K->266929K(506816K), [Metaspace: 2706K->2706K(1056768K)], 0.0720599 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
```

在0.824秒的时候发生了一次老年代的GC，从307975K-266929K = 41046K（约40M）。Metaspace没有2706K->2706K变化持续时间为0.07秒。

2、并行GC

-Xmx 512m -Xms512



第一次Minor GC

```
2021-03-31T14:08:50.731-0800: 0.205: [GC (Allocation Failure) [PSYoungGen: 131178K->21499K(153088K)] 131178K->48782K(502784K), 0.0203588 secs] [Times: user=0.02 sys=0.05, real=0.02 secs]
```

第一次GC发生时间在程序启动后的0.205秒，是一次Minor GC，并且发生原因是Allocation Failure，表明本次引起GC的原因是因为在年轻代中没有足够的空间能够存储新的数据了。
本次Minor GC共回收掉了年轻代 131178K - 21499K = 109679K（约107M）的容量。
同时堆内存从131178K - 48782K = 82396K（约80M）。109679K - 82396K = 27283k（约27M），耗时为0.0203588 secs
本次总共有27M的数据从年轻代晋升到了老年代。
本次GC用户耗时为user=0.02秒，内核耗时为sys=0.05秒，总耗时为real=0.02秒。

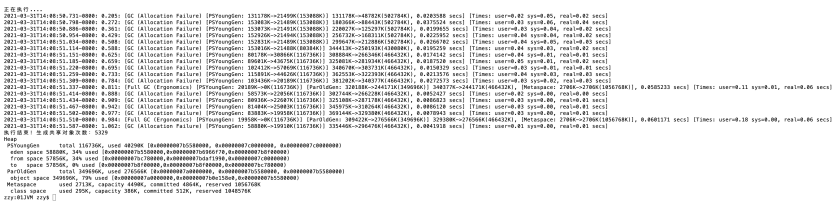
Full GC

```
2021-03-31T14:08:51.337-0800: 0.811: [Full GC (Ergonomics) [PSYoungGen: 20189K->0K(116736K)] [ParOldGen: 320188K->244171K(349696K)] 340377K->244171K(466432K), [Metaspace: 2706K->2706K(1056768K)], 0.0585233 secs] [Times: user=0.11 sys=0.01, real=0.06 secs]
```

FullGC发生在程序启动后的0.811s，新生代直接变成了0k，老年代，320188K->244171K = 76017k（74M）。Metaspace没有变化。总耗时为0.06。

Minor GC 仅仅年轻代区回收； Full GC 清理了不仅清理了年轻代对象（一般全部清理），还清理了老年代和Metaspace，所以Full GC的耗时远大与Minor GC。

3、CMS

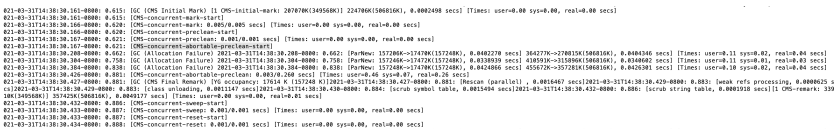


YOUNG GC

```
2021-03-31T14:38:29.775-0800: 0.229: [GC (Allocation Failure) 2021-03-31T14:38:29.775-0800: 0.229: [ParNew: 139776K->17471K(157248K), 0.0187305 secs] 139776K->41591K(506816K), 0.0188049 secs] [Times: user=0.02 sys=0.04, real=0.02 secs]
```

第一次GC发生时间在程序启动后的0.229秒，是一次Minor GC，并且发生原因是Allocation Failure。
本次Minor GC共回收掉了年轻代 139776K - 17471K = 122305（约120M）的容量。0.0187305 secs
同时堆内存从131178K - 48782K = 98185K（约96M）。122305- 48782K = 24120（约24M），耗时为0.0188049 secs
本次总共有24M的数据从年轻代晋升到了老年代。
本次GC用户耗时为user=0.02秒，内核耗时为sys=0.04秒，总耗时为real=0.02秒。

FULLGC



1.Initial Mark(初始标记)：当前步骤需要JVM暂停STW，标记的根对象用时0.0002498 secs 几乎没有任何停顿。

```
[GC (CMS Initial Mark) [1 CMS-initial-mark: 207070K(349568K)] 224706K(506816K), 0.0002498 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
```

2.并发标记。

```
2021-03-31T14:38:30.161-0800: 0.615: [CMS-concurrent-mark-start]
2021-03-31T14:38:30.166-0800: 0.620: [CMS-concurrent-mark: 0.005/0.005 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
```

3.并发预清理。

```
2021-03-31T14:38:30.166-0800: 0.620: [CMS-concurrent-preclean-start]
2021-03-31T14:38:30.167-0800: 0.621: [CMS-concurrent-preclean: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
2021-03-31T14:38:30.167-0800: 0.621: [CMS-concurrent-abortable-preclean-start]
2021-03-31T14:38:30.426-0800: 0.881: [CMS-concurrent-abortable-preclean: 0.003/0.260 secs] [Times: user=0.46 sys=0.07, real=0.26 secs]
```

4.最终标记。需要STW

```
2021-03-31T14:38:30.427-0800: 0.881: [GC (CMS Final Remark) [YG occupancy: 17614 K (157248 K)]2021-03-31T14:38:30.427-0800: 0.881: [Rescan (parallel) , 0.0016467 secs]2021-03-31T14:38:30.429-0800: 0.883: [weak refs processing, 0.0000625 secs]2021-03-31T14:38:30.429-0800: 0.883: [class unloading, 0.0011147 secs]2021-03-31T14:38:30.430-0800: 0.884: [scrub symbol table, 0.0015494 secs]2021-03-31T14:38:30.432-0800: 0.886: [scrub string table, 0.0001918 secs][1 CMS-remark: 339810K(349568K)] 357425K(506816K), 0.0049177 secs] [Times: user=0.00 sys=0.00, real=0.01 secs]
```

5.Concurrent Sweep(并发清除)

```
2021-03-31T14:38:30.432-0800: 0.886: [CMS-concurrent-sweep-start]
2021-03-31T14:38:30.433-0800: 0.887: [CMS-concurrent-sweep: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
```

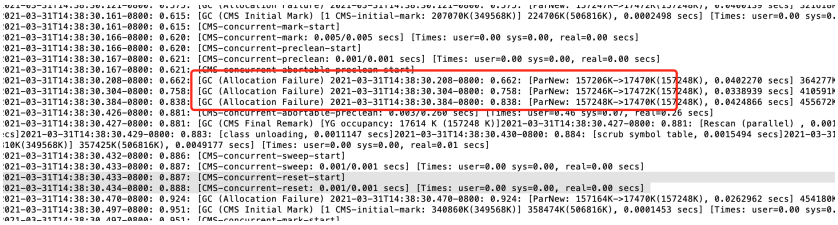
6.并发重置

```
2021-03-31T14:38:30.433-0800: 0.887: [CMS-concurrent-reset-start]
```

```
2021-03-31T14:38:30.434-0800: 0.888: [CMS-concurrent-reset: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
```

CMS垃圾回收主要分为初始标记，并发标记，并发预清除，最终标记，并发清除，并发重置。仅在初始标记, 最终标记会STW，且时间非常短。

值得注意的是在并发预清理的期间，发生了3次youngGC



4、G1GC

Minor GC

```
2021-03-31T15:56:47.128-0800: 0.198: [GC pause (G1 Evacuation Pause) (young), 0.0092168 secs]
   [Parallel Time: 8.7 ms, GC Workers: 4]
      [GC Worker Start (ms): Min: 198.5, Avg: 199.0, Max: 200.3, Diff: 1.7]
      [Ext Root Scanning (ms): Min: 0.0, Avg: 0.6, Max: 1.4, Diff: 1.4, Sum: 2.3]
      [Update RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
         [Processed Buffers: Min: 0, Avg: 0.0, Max: 0, Diff: 0, Sum: 0]
      [Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
      [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
      [Object Copy (ms): Min: 5.9, Avg: 6.8, Max: 7.4, Diff: 1.5, Sum: 27.3]
      [Termination (ms): Min: 0.0, Avg: 0.7, Max: 1.1, Diff: 1.1, Sum: 2.9]
         [Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
      [GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.1, Diff: 0.0, Sum: 0.1]
      [GC Worker Total (ms): Min: 6.9, Avg: 8.1, Max: 8.6, Diff: 1.7, Sum: 32.6]
      [GC Worker End (ms): Min: 207.1, Avg: 207.2, Max: 207.2, Diff: 0.0]
   [Code Root Fixup: 0.0 ms]
   [Code Root Purge: 0.0 ms]
   [Clear CT: 0.0 ms]
   [Other: 0.5 ms]
      [Choose CSet: 0.0 ms]
      [Ref Proc: 0.2 ms]
      [Ref Enq: 0.0 ms]
      [Redirty Cards: 0.0 ms]
      [Humongous Register: 0.1 ms]
      [Humongous Reclaim: 0.0 ms]
      [Free CSet: 0.0 ms]
   [Eden: 25.0M(25.0M)->0.0B(21.0M) Survivors: 0.0B->4096.0K Heap: 35.5M(512.0M)->13.7M(512.0M)]
 [Times: user=0.01 sys=0.00, real=0.01 secs]
```

日志分析:

```
2021-03-31T15:56:47.128-0800: 0.198: [GC pause (G1 Evacuation Pause) (young), 0.0092168 secs]
```

发生在程序启动后的0.198秒，只执行了young区回收，耗时0.0092168 秒。STW时间0.0092168秒。

```
[Parallel Time: 8.7 ms, GC Workers: 4]
```

GC线程为4个，时间为8.7ms

```
GC Worker Start (ms): Min: 198.5, Avg: 199.0, Max: 200.3, Diff: 1.7]
```

GC线程工作的启动时间（相对JVM的启动时间），如果Min与Max相差较 大，说明有其他线程抢占CPU资源

```
 [Ext Root Scanning (ms): Min: 0.0, Avg: 0.6, Max: 1.4, Diff: 1.4, Sum: 2.3]
```

扫描对外内存的时间

```
[Update RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
```

线程更新remember set的时间

```
[Processed Buffers: Min: 0, Avg: 0.0, Max: 0, Diff: 0, Sum: 0]
[Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
```

扫描CSet中的region对应的RSet的时间。

```
  [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
```

扫描实际代码中Root的时间

```
 [Object Copy (ms): Min: 5.9, Avg: 6.8, Max: 7.4, Diff: 1.5, Sum: 27.3]
```

拷贝回收集中的对象用时

```
[Termination (ms): Min: 0.0, Avg: 0.7, Max: 1.1, Diff: 1.1, Sum: 2.9]
```

确认GC线程可以终止的耗时

```
[Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
```

有多少GC线程尝试中止

```
[GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.1, Diff: 0.0, Sum: 0.1]
```

其他任务的耗时。

```
[GC Worker Total (ms): Min: 6.9, Avg: 8.1, Max: 8.6, Diff: 1.7, Sum: 32.6]
```

GC线程总耗时。

```
[GC Worker End (ms): Min: 207.1, Avg: 207.2, Max: 207.2, Diff: 0.0]
```

GC作业完成时刻。

```
[Code Root Fixup: 0.0 ms]
```

释放用于管理并行活动的内部数据时间

```
[Code Root Purge: 0.0 ms]
```

请求其他部分数据时间

```
[Clear CT: 0.0 ms]
```

```
[Other: 0.5 ms]
```

其他活动耗时

```
[Choose CSet: 0.0 ms]、
```

选择CSet的耗时时间

```
[Ref Proc: 0.2 ms]
```

处理非强引用的时间

```
[Ref Enq: 0.0 ms]
 [Redirty Cards: 0.0 ms]
 [Humongous Register: 0.1 ms]
 [Humongous Reclaim: 0.0 ms]
```

```
[Free CSet: 0.0 ms]
```

释放regions的时间。

```
[Eden: 25.0M(25.0M)->0.0B(21.0M) Survivors: 0.0B->4096.0K
Heap: 35.5M(512.0M)->13.7M(512.0M)]
```

Eden区GC使用前后大小 25.0M(25.0M)->0.0B(21.0M)。Survivors区GC 前后的使用: 0.0B->4096.0K。暂停前后整个堆内存使用总量。

```
 [Times: user=0.01 sys=0.00, real=0.01 secs]
```

GC时间的持续时间。0.01秒

Full GC

日志:

```
2021-03-31T15:56:47.714-0800: 0.784: [GC pause (G1 Humongous Allocation) (young)
(initial-mark), 0.0048943 secs]
   [Parallel Time: 4.5 ms, GC Workers: 4]
      [GC Worker Start (ms): Min: 784.2, Avg: 784.3, Max: 784.4, Diff: 0.2]
      [Ext Root Scanning (ms): Min: 0.0, Avg: 0.1, Max: 0.2, Diff: 0.2, Sum: 0.6]
      [Update RS (ms): Min: 0.1, Avg: 0.1, Max: 0.1, Diff: 0.0, Sum: 0.6]
         [Processed Buffers: Min: 1, Avg: 2.5, Max: 4, Diff: 3, Sum: 10]
      [Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
      [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
      [Object Copy (ms): Min: 4.0, Avg: 4.0, Max: 4.0, Diff: 0.1, Sum: 16.0]
      [Termination (ms): Min: 0.0, Avg: 0.1, Max: 0.1, Diff: 0.1, Sum: 0.2]
         [Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
      [GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.1]
      [GC Worker Total (ms): Min: 4.2, Avg: 4.3, Max: 4.4, Diff: 0.2, Sum: 17.4]
      [GC Worker End (ms): Min: 788.6, Avg: 788.7, Max: 788.7, Diff: 0.0]
   [Code Root Fixup: 0.0 ms]
   [Code Root Purge: 0.0 ms]
   [Clear CT: 0.0 ms]
   [Other: 0.4 ms]
      [Choose CSet: 0.0 ms]
      [Ref Proc: 0.1 ms]
      [Ref Enq: 0.0 ms]
      [Redirty Cards: 0.0 ms]
      [Humongous Register: 0.1 ms]
      [Humongous Reclaim: 0.0 ms]
      [Free CSet: 0.0 ms]
   [Eden: 3072.0K(55.0M)->0.0B(54.0M) Survivors: 20.0M->2048.0K Heap: 360.4M(512.0M)-
>359.8M(512.0M)]
 [Times: user=0.01 sys=0.01, real=0.00 secs]
2021-03-31T15:56:47.719-0800: 0.789: [GC concurrent-root-region-scan-start]
2021-03-31T15:56:47.719-0800: 0.789: [GC concurrent-root-region-scan-end, 0.0001759
secs]
2021-03-31T15:56:47.719-0800: 0.789: [GC concurrent-mark-start]
2021-03-31T15:56:47.724-0800: 0.795: [GC concurrent-mark-end, 0.0054434 secs]
2021-03-31T15:56:47.725-0800: 0.795: [GC remark 2021-03-31T15:56:47.725-0800: 0.795:
[Finalize Marking, 0.0002283 secs] 2021-03-31T15:56:47.725-0800: 0.795: [GC ref-proc,
0.0000430 secs] 2021-03-31T15:56:47.725-0800: 0.795: [Unloading, 0.0008065 secs],
0.0018361 secs]
 [Times: user=0.01 sys=0.00, real=0.00 secs]
2021-03-31T15:56:47.727-0800: 0.797: [GC cleanup 373M->373M(512M), 0.0005796 secs]
```

```
35  [Times: user=0.00 sys=0.00, real=0.00 secs]
```

上一次youngGC

```
1  [Eden: 300.0M(300.0M)->0.0B(55.0M) Survivors: 7168.0K->20.0M Heap: 440.8M(512.0M)-
   >358.0M(512.0M)]
```

显示分配的Yong 区将要耗尽，在次Minor GC耗时0.014s,之后紧接着 发生了G1的并发标记清理垃圾。对空间的使用量 69.9%=358M/512.0M。

第一阶段：initial-mark，初始标记[Times: user=0.01 sys=0.01, real=0.00 secs]，耗时0.0048943 secs 发生STW

第二阶段：concurrent-root-region-scan ， 从root regions中扫存活对象

第三阶段：GC concurrent-mark-并发标记，递归扫描整个堆里的对象图， 找出要回收的对象， 但可与用户程序并发执行。

第四阶段：remark，对用户线程做另一个短暂的暂停， 用于处理并发阶段结束后仍遗留下来的最后那少量的SATB记录。

第五阶段：cleanup;这一阶段主要为接下来即将要进行的对象转移阶段做准备。统计出所有小堆区中的存活对象，并且对这些小堆区按存活对象数进行排序。也为下一次标记阶段作必要的整理工作。

```
1  [Times: user=0.00 sys=0.00, real=0.00 secs]
```

该阶段耗时