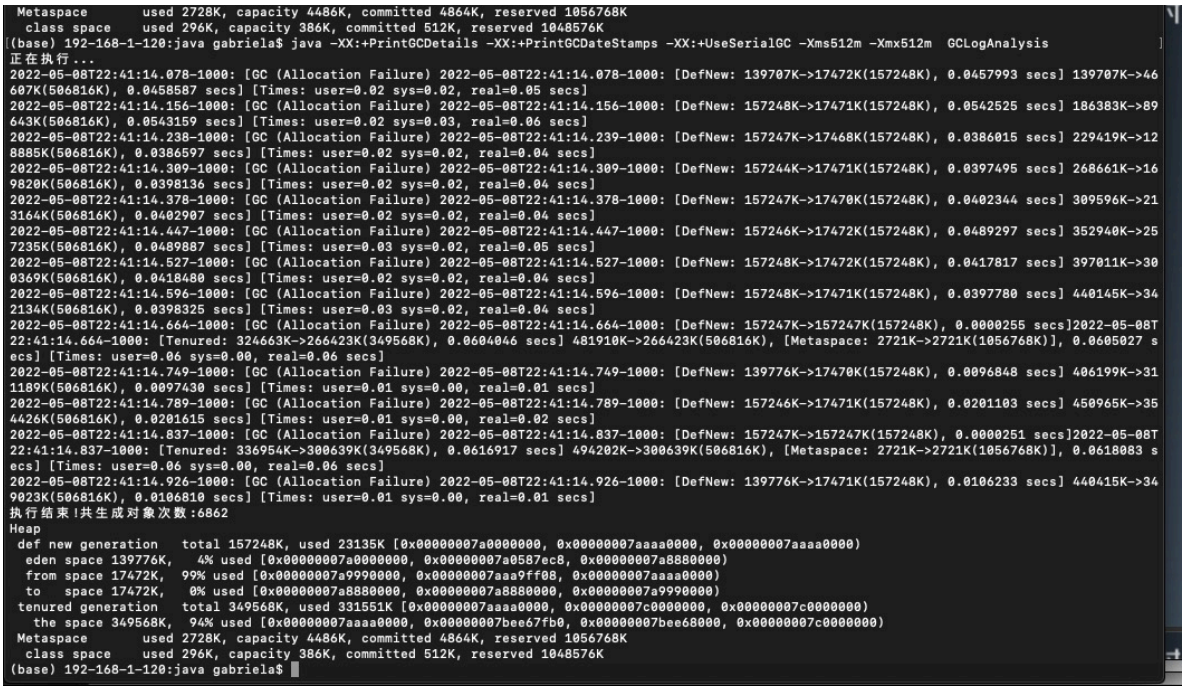
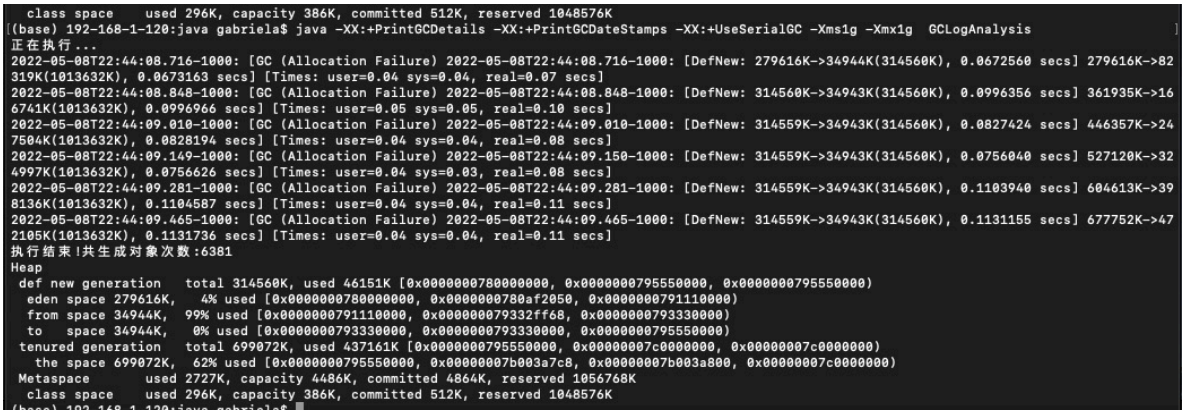
Module02

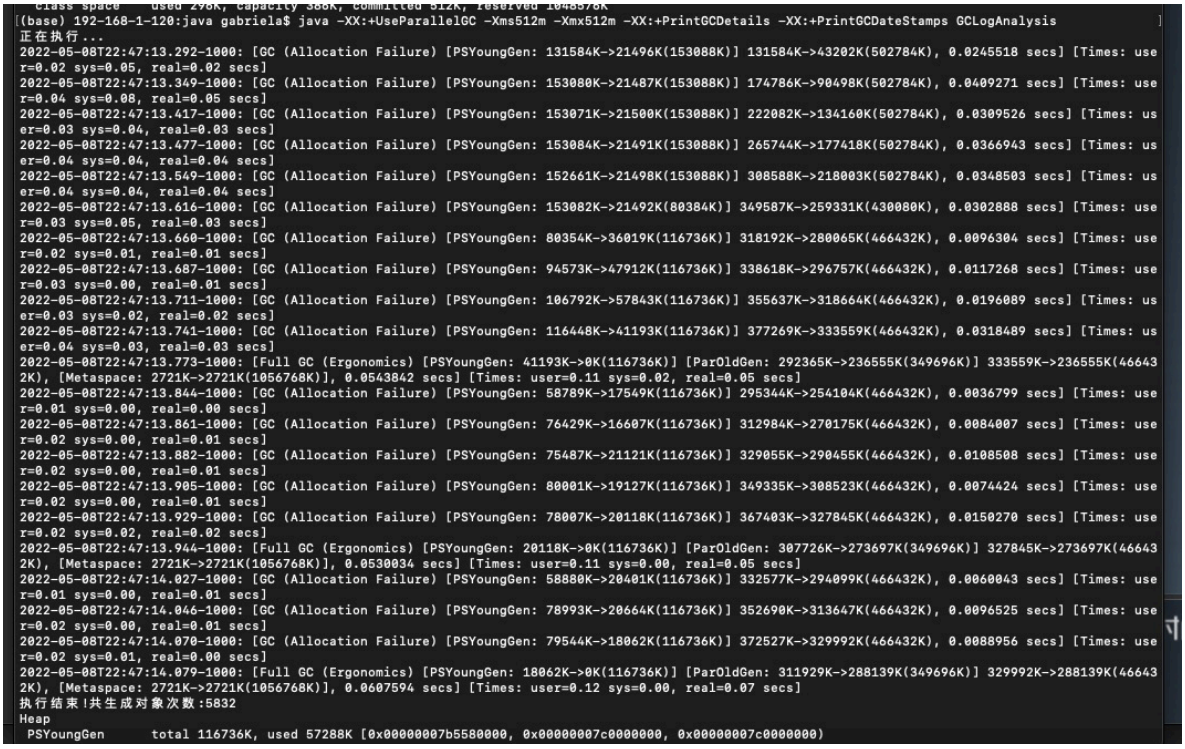




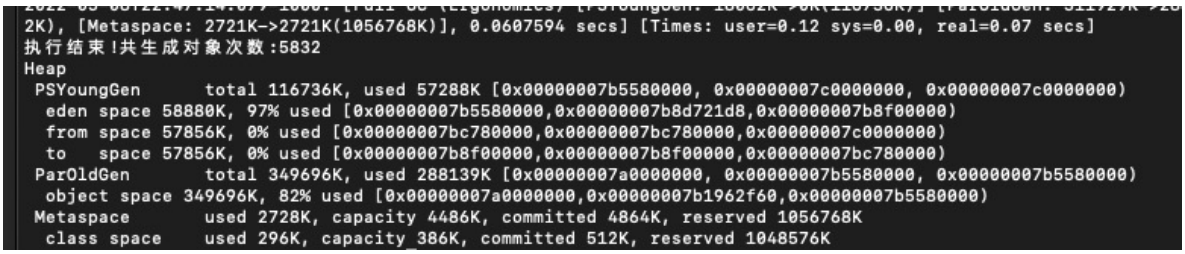
SerialGC

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。

调整xms和xmx以后明显GC次数减少，但是GC时间变长很多。



```
(base) 192-168-1-120:java gabriela$ java -XX:+UseParallelGC -Xms512m -Xmx512m -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis
正在执行...
2022-05-08T22:47:13.292-1000: [GC (Allocation Failure) [PSYoungGen: 131584K->21496K(153088K)] 131584K->43202K(502784K), 0.0245518 secs] [Times: user=0.02 sys=0.05, real=0.02 secs]
2022-05-08T22:47:13.349-1000: [GC (Allocation Failure) [PSYoungGen: 153080K->21487K(153088K)] 174786K->90498K(502784K), 0.0409271 secs] [Times: user=0.04 sys=0.08, real=0.05 secs]
2022-05-08T22:47:13.417-1000: [GC (Allocation Failure) [PSYoungGen: 153071K->21500K(153088K)] 222082K->134160K(502784K), 0.0309526 secs] [Times: user=0.03 sys=0.04, real=0.03 secs]
2022-05-08T22:47:13.477-1000: [GC (Allocation Failure) [PSYoungGen: 153084K->21491K(153088K)] 265744K->177418K(502784K), 0.0366943 secs] [Times: user=0.04 sys=0.04, real=0.04 secs]
2022-05-08T22:47:13.549-1000: [GC (Allocation Failure) [PSYoungGen: 152661K->21498K(153088K)] 308588K->218003K(502784K), 0.0348503 secs] [Times: user=0.04 sys=0.04, real=0.04 secs]
2022-05-08T22:47:13.616-1000: [GC (Allocation Failure) [PSYoungGen: 153082K->21492K(80384K)] 349587K->259331K(430080K), 0.0302888 secs] [Times: user=0.03 sys=0.05, real=0.03 secs]
2022-05-08T22:47:13.660-1000: [GC (Allocation Failure) [PSYoungGen: 80354K->36019K(116736K)] 318192K->280065K(466432K), 0.0096304 secs] [Times: user=0.02 sys=0.01, real=0.01 secs]
2022-05-08T22:47:13.687-1000: [GC (Allocation Failure) [PSYoungGen: 94573K->47912K(116736K)] 338618K->296757K(466432K), 0.0117268 secs] [Times: user=0.03 sys=0.00, real=0.01 secs]
2022-05-08T22:47:13.711-1000: [GC (Allocation Failure) [PSYoungGen: 106792K->57843K(116736K)] 355637K->318664K(466432K), 0.0196089 secs] [Times: user=0.03 sys=0.02, real=0.02 secs]
2022-05-08T22:47:13.741-1000: [GC (Allocation Failure) [PSYoungGen: 116448K->41193K(116736K)] 377269K->333559K(466432K), 0.0318489 secs] [Times: user=0.04 sys=0.03, real=0.03 secs]
2022-05-08T22:47:13.773-1000: [Full GC (Ergonomics) [PSYoungGen: 41193K->0K(116736K)] [ParOldGen: 292365K->236555K(349696K)] 333559K->236555K(466432K), [Metaspace: 2721K->2721K(1056768K)], 0.0543842 secs] [Times: user=0.11 sys=0.02, real=0.05 secs]
2022-05-08T22:47:13.844-1000: [GC (Allocation Failure) [PSYoungGen: 58789K->17549K(116736K)] 295344K->254104K(466432K), 0.0036799 secs] [Times: user=0.01 sys=0.00, real=0.00 secs]
2022-05-08T22:47:13.861-1000: [GC (Allocation Failure) [PSYoungGen: 76429K->16607K(116736K)] 312984K->270175K(466432K), 0.0084007 secs] [Times: user=0.02 sys=0.00, real=0.01 secs]
2022-05-08T22:47:13.882-1000: [GC (Allocation Failure) [PSYoungGen: 75487K->21121K(116736K)] 329055K->290455K(466432K), 0.0108508 secs] [Times: user=0.02 sys=0.00, real=0.01 secs]
2022-05-08T22:47:13.905-1000: [GC (Allocation Failure) [PSYoungGen: 80001K->19127K(116736K)] 349335K->308523K(466432K), 0.0074424 secs] [Times: user=0.02 sys=0.00, real=0.01 secs]
2022-05-08T22:47:13.929-1000: [GC (Allocation Failure) [PSYoungGen: 78007K->20118K(116736K)] 367403K->327845K(466432K), 0.0150270 secs] [Times: user=0.02 sys=0.02, real=0.02 secs]
2022-05-08T22:47:13.944-1000: [Full GC (Ergonomics) [PSYoungGen: 20118K->0K(116736K)] [ParOldGen: 307726K->273697K(349696K)] 327845K->273697K(466432K), [Metaspace: 2721K->2721K(1056768K)], 0.0530034 secs] [Times: user=0.11 sys=0.00, real=0.05 secs]
2022-05-08T22:47:14.027-1000: [GC (Allocation Failure) [PSYoungGen: 58880K->20401K(116736K)] 332577K->294099K(466432K), 0.0060043 secs] [Times: user=0.01 sys=0.00, real=0.01 secs]
2022-05-08T22:47:14.046-1000: [GC (Allocation Failure) [PSYoungGen: 78993K->20664K(116736K)] 352690K->313647K(466432K), 0.0096525 secs] [Times: user=0.02 sys=0.00, real=0.01 secs]
2022-05-08T22:47:14.070-1000: [GC (Allocation Failure) [PSYoungGen: 79544K->18062K(116736K)] 372527K->329992K(466432K), 0.0088956 secs] [Times: user=0.02 sys=0.01, real=0.00 secs]
2022-05-08T22:47:14.079-1000: [Full GC (Ergonomics) [PSYoungGen: 18062K->0K(116736K)] [ParOldGen: 311929K->288139K(349696K)] 329992K->288139K(466432K), [Metaspace: 2721K->2721K(1056768K)], 0.0607594 secs] [Times: user=0.12 sys=0.00, real=0.07 secs]
执行结束!共生成对象次数:5832
Heap
 PSYoungGen      total 116736K, used 57288K [0x00000007b5580000, 0x00000007c0000000, 0x00000007c0000000)
```



```
2K), [Metaspace: 2721K->2721K(1056768K)], 0.0607594 secs] [Times: user=0.12 sys=0.00, real=0.07 secs]
执行结束!共生成对象次数:5832
Heap
 PSYoungGen      total 116736K, used 57288K [0x00000007b5580000, 0x00000007c0000000, 0x00000007c0000000)
  eden space 58880K, 97% used [0x00000007b5580000,0x00000007b8d721d8,0x00000007b8f00000)
  from space 57856K, 0% used [0x00000007bc780000,0x00000007bc780000,0x00000007c0000000)
  to   space 57856K, 0% used [0x00000007b8f00000,0x00000007b8f00000,0x00000007bc780000)
 ParOldGen       total 349696K, used 288139K [0x00000007a0000000, 0x00000007b5580000, 0x00000007b5580000)
  object space 349696K, 82% used [0x00000007a0000000,0x00000007b1962f60,0x00000007b5580000)
 Metaspace       used 2728K, capacity 4486K, committed 4864K, reserved 1056768K
  class space    used 296K, capacity 386K, committed 512K, reserved 1048576K
```

ParallelGC

并行垃圾收集器这一类组合，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-sweep-compact）算法”。年轻代和老年代的垃圾回收都会触发 STW 事件，暂停所有的应用线程来执行垃圾收集。两者在执行“标记和复制/整理”阶段时都使用多个线程。通过并行执行，使得 GC 时间大幅减少。

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。

与SerialGC相比，GC暂停时间大幅减少。

```
 class space    used 296K, capacity 386K, committed 512K, reserved 1048576K
(base) 192-168-1-120:java gabriela$ java -XX:+UseConcMarkSweepGC -Xms512m -Xmx512m -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis
正在执行...
2022-05-08T22:57:21.364-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.364-1000: [ParNew: 139776K->17471K(157248K), 0.0752469 secs] 139776K->46
409K(506816K), 0.0757740 secs] [Times: user=0.04 sys=0.04, real=0.08 secs]
2022-05-08T22:57:21.473-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.473-1000: [ParNew: 157247K->17472K(157248K), 0.1095741 secs] 186185K->88
874K(506816K), 0.1096371 secs] [Times: user=0.05 sys=0.05, real=0.11 secs]
2022-05-08T22:57:21.615-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.615-1000: [ParNew: 157136K->17472K(157248K), 0.0580305 secs] 228539K->13
0303K(506816K), 0.0580960 secs] [Times: user=0.10 sys=0.03, real=0.06 secs]
2022-05-08T22:57:21.704-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.704-1000: [ParNew: 157248K->17472K(157248K), 0.0649501 secs] 270079K->17
4249K(506816K), 0.0650135 secs] [Times: user=0.11 sys=0.03, real=0.06 secs]
2022-05-08T22:57:21.800-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.800-1000: [ParNew: 157182K->17472K(157248K), 0.0632711 secs] 313959K->22
0789K(506816K), 0.0633369 secs] [Times: user=0.09 sys=0.03, real=0.06 secs]
2022-05-08T22:57:21.864-1000: [GC (CMS Initial Mark) [1 CMS-initial-mark: 203317K(349568K)] 223927K(506816K), 0.0049326 secs] [Times: user=0.00 sys
=0.00, real=0.00 secs]
2022-05-08T22:57:21.869-1000: [CMS-concurrent-mark-start]
2022-05-08T22:57:21.880-1000: [CMS-concurrent-mark: 0.011/0.011 secs] [Times: user=0.01 sys=0.01, real=0.02 secs]
2022-05-08T22:57:21.880-1000: [CMS-concurrent-preclean-start]
2022-05-08T22:57:21.881-1000: [CMS-concurrent-preclean: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
2022-05-08T22:57:21.881-1000: [CMS-concurrent-abortable-preclean-start]
2022-05-08T22:57:21.905-1000: [GC (Allocation Failure) 2022-05-08T22:57:21.905-1000: [ParNew: 157248K->17470K(157248K), 0.0675343 secs] 360565K->26
1348K(506816K), 0.0677017 secs] [Times: user=0.12 sys=0.03, real=0.07 secs]
2022-05-08T22:57:22.005-1000: [GC (Allocation Failure) 2022-05-08T22:57:22.005-1000: [ParNew: 157246K->17470K(157248K), 0.0614996 secs] 401124K->30
4789K(506816K), 0.0615616 secs] [Times: user=0.08 sys=0.03, real=0.06 secs]
2022-05-08T22:57:22.103-1000: [GC (Allocation Failure) 2022-05-08T22:57:22.103-1000: [ParNew2022-05-08T22:57:22.171-1000: [CMS-concurrent-abortable
-preclean: 0.007/0.290 secs] [Times: user=0.41 sys=0.08, real=0.29 secs]
: 157246K->17471K(157248K), 0.0798562 secs] 444565K->348206K(506816K), 0.0799175 secs] [Times: user=0.14 sys=0.03, real=0.08 secs]
2022-05-08T22:57:22.183-1000: [GC (CMS Final Remark) [YG occupancy: 17979 K (157248 K)]2022-05-08T22:57:22.183-1000: [Rescan (parallel) , 0.0006981
 secs]2022-05-08T22:57:22.184-1000: [weak refs processing, 0.0000306 secs]2022-05-08T22:57:22.184-1000: [class unloading, 0.0003823 secs]2022-05-08
T22:57:22.184-1000: [scrub symbol table, 0.0006223 secs]2022-05-08T22:57:22.185-1000: [scrub string table, 0.0002426 secs][1 CMS-remark: 330735K(34
9568K)] 348714K(506816K), 0.0021130 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
2022-05-08T22:57:22.186-1000: [CMS-concurrent-sweep-start]
2022-05-08T22:57:22.187-1000: [CMS-concurrent-sweep: 0.001/0.001 secs] [Times: user=0.01 sys=0.00, real=0.00 secs]
2022-05-08T22:57:22.187-1000: [CMS-concurrent-reset-start]
2022-05-08T22:57:22.188-1000: [CMS-concurrent-reset: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
执行结束!共生成对象次数:4579
Heap
 par new generation   total 157248K, used 134996K [0x00000007a0000000, 0x00000007aaaa0000, 0x00000007aaaa0000)
  eden space 139776K,  84% used [0x00000007a0000000, 0x00000007a72c5470, 0x00000007a8880000)
  from space 17472K,  99% used [0x00000007a8880000, 0x00000007a998fc88, 0x00000007a9990000)
  to   space 17472K,   0% used [0x00000007a9990000, 0x00000007a9990000, 0x00000007aaaa0000)
 concurrent mark-sweep generation total 349568K, used 288088K [0x00000007aaaa0000, 0x00000007c0000000, 0x00000007c0000000)
 Metaspace       used 2728K, capacity 4486K, committed 4864K, reserved 1056768K
  class space    used 296K, capacity 386K, committed 512K, reserved 1048576K
```

CMSGC

以获取最短回收停顿时间为目标的收集器，基于并发“标记清理”实现。

```
Error: A fatal exception has occurred. Program will exit.
(base) 192-168-1-120:java gabriela$ java -XX:+UseG1GC -Xms512m -Xmx512m -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis
正在执行...
2022-05-08T23:00:52.012-1000: [GC pause (G1 Evacuation Pause) (young), 0.0108998 secs]
   [Parallel Time: 7.1 ms, GC Workers: 4]
      [GC Worker Start (ms): Min: 145.1, Avg: 145.9, Max: 147.0, Diff: 2.0]
      [Ext Root Scanning (ms): Min: 0.0, Avg: 0.2, Max: 0.7, Diff: 0.7, Sum: 0.9]
      [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: 4.6, Avg: 5.7, Max: 6.3, Diff: 1.7, Sum: 22.6]
      [Termination (ms): Min: 0.0, Avg: 0.2, Max: 0.3, Diff: 0.3, Sum: 0.7]
         [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: 0.1, Sum: 0.2]
      [GC Worker Total (ms): Min: 5.0, Avg: 6.1, Max: 7.0, Diff: 2.0, Sum: 24.5]
      [GC Worker End (ms): Min: 152.0, Avg: 152.1, Max: 152.1, Diff: 0.0]
   [Code Root Fixup: 0.0 ms]
   [Code Root Purge: 0.0 ms]
   [Clear CT: 0.1 ms]
   [Other: 3.7 ms]
      [Choose CSet: 0.0 ms]
      [Ref Proc: 0.1 ms]
      [Ref Enq: 0.0 ms]
      [Redirty Cards: 0.1 ms]
      [Humongous Register: 0.0 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: 34.7M(512.0M)->14.2M(512.0M)]
 [Times: user=0.02 sys=0.01, real=0.01 secs]
2022-05-08T23:00:52.038-1000: [GC pause (G1 Evacuation Pause) (young), 0.0049180 secs]
   [Parallel Time: 4.4 ms, GC Workers: 4]
      [GC Worker Start (ms): Min: 167.1, Avg: 167.4, Max: 168.1, Diff: 1.0]
      [Ext Root Scanning (ms): Min: 0.0, Avg: 0.2, Max: 0.4, Diff: 0.4, Sum: 0.7]
      [Update RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
         [Processed Buffers: Min: 0, Avg: 0.5, Max: 2, Diff: 2, Sum: 2]
      [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: 3.3, Avg: 3.6, Max: 3.7, Diff: 0.5, Sum: 14.2]
      [Termination (ms): Min: 0.0, Avg: 0.3, Max: 0.5, Diff: 0.5, Sum: 1.1]
         [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: 3.3, Avg: 4.1, Max: 4.4, Diff: 1.0, Sum: 16.3]
```

G1GC

G1 GC 这是一种兼顾吞吐量和停顿时间的 GC 实现，是 Oracle JDK 9 以后的默认 GC 选项。G1 可以直观的设定停顿时间的目标，相比于 CMS GC，G1 未必能做到

CMS 在最好情况下的延时停顿，但是最差情况要好很多。