

命令行参数

1. -XX:+PrintGCDetails 打印gc日志
2. -XX:+PrintGCDateStamps 在打印gc日志的同时输出timestamp:2021-03-25T13:52:30.529-0800
3. -Xmx1g -Xms1g 最大堆1g,初始化堆1g

日志输出

| | |
|---|---|
| java -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xmx1g -Xms1g week_01_jvm.jmm.GCLogAnalysis | |
| 正在执行... | |
| 2021-03-25T13:59:22.894-0800: | [GC (Allocation Failure) [PSYoungGen: 261838K->43515K(305664K)] 261838K] |
| 2021-03-25T13:59:22.959-0800: | [GC (Allocation Failure) [PSYoungGen: 305659K->43518K(305664K)] 339665K] |
| 2021-03-25T13:59:23.031-0800: | [GC (Allocation Failure) [PSYoungGen: 305662K->43512K(305664K)] 408355K] |
| 2021-03-25T13:59:23.092-0800: | [GC (Allocation Failure) [PSYoungGen: 305656K->43504K(305664K)] 475406K] |
| 2021-03-25T13:59:23.161-0800: | [GC (Allocation Failure) [PSYoungGen: 305648K->43504K(305664K)] 540696K] |
| 2021-03-25T13:59:23.241-0800: | [GC (Allocation Failure) [PSYoungGen: 305648K->43513K(160256K)] 616080K] |
| 2021-03-25T13:59:23.294-0800: | [GC (Allocation Failure) [PSYoungGen: 160249K->80745K(232960K)] 536698K] |
| 2021-03-25T13:59:23.326-0800: | [GC (Allocation Failure) [PSYoungGen: 197481K->103523K(232960K)] 579785K] |
| 2021-03-25T13:59:23.363-0800: | [GC (Allocation Failure) [PSYoungGen: 220030K->111732K(232960K)] 611225K] |
| 2021-03-25T13:59:23.404-0800: | [GC (Allocation Failure) [PSYoungGen: 228468K->79083K(232960K)] 637631K] |
| 2021-03-25T13:59:23.473-0800: | [GC (Allocation Failure) [PSYoungGen: 195819K->35867K(232960K)] 670251K] |
| 2021-03-25T13:59:23.609-0800: | [GC (Allocation Failure) [PSYoungGen: 152603K->34640K(232960K)] 698828K] |
| 2021-03-25T13:59:23.714-0800: | [GC (Allocation Failure) [PSYoungGen: 151376K->37890K(232960K)] 730161K] |
| 2021-03-25T13:59:23.776-0800: | [Full GC (Ergonomics) [PSYoungGen: 37890K->0K(232960K)] [ParOldGen: 607392K->607392K(699392K)] 730161K] |
| 执行结束! 共生成对象次数: 8903 | |
| Heap | |
| PSYoungGen | total 232960K, used 5447K [0x00000007aab00000, 0x00000007c0000000, 0x00000007c0000000) |
| eden space | 116736K, 4% used [0x00000007aab00000, 0x00000007ab051f78, 0x00000007b1d00000) |
| from space | 116224K, 0% used [0x00000007b1d00000, 0x00000007b1d00000, 0x00000007b8e80000) |
| to space | 116224K, 0% used [0x00000007b8e80000, 0x00000007b8e80000, 0x00000007c0000000) |
| ParOldGen | total 699392K, used 323242K [0x0000000780000000, 0x00000007aab00000, 0x00000007aab00000) |
| object space | 699392K, 46% used [0x0000000780000000, 0x0000000793baa918, 0x00000007aab00000) |
| Metaspace | used 2572K, capacity 4486K, committed 4864K, reserved 1056768K |
| class space | used 277K, capacity 386K, committed 512K, reserved 1048576K |

日志解析

young gc 日志

如果新生代内存不够分配,则开始young gc,使用Parallel Scavenge策略

- 1 2021-03-25T13:59:22.894-0800: //时间戳
- 2 [GC (Allocation Failure) //gc原因, 分配内存失败
- 3 [PSYoungGen: 261838K->43515K(305664K)] //ygc内存从261838K清理后变成43515K,
- 4 261838K->77521K(1005056K), //整个堆(新+老年代) 从261838K->77521K(1005056K) 括号里代表堆的总大小
- 5 0.0238546 secs] // stw的时间
- 6 [Times: user=0.03 sys=0.04, real=0.02 secs] cpu使用的情况

问题1:

第一行日志为什么PSYoungGen的内存大小是261838kb,而整个堆此时也是261838kb为什么?

回答:

第一次jvm启动,old gen没有数据,数据都在young区.所以第一次gc的时候,参数上young区和整个堆区看上去大小是一致的

问题2:

第一次gc,在清理后,PSYoungGen清理了大约213MB(261838K->43515K)左右数据,而整个堆清理了大约180mb(261838K->77521K)为什么导致了整个young区减少了那么多,而整个堆并没有减少那么多呢?

回答:

到了old区. $213\text{MB}-180\text{mb} = 33\text{mb}$.通过计算可以得到富余的内存数量.那么应该是到了old区.

old gc 日志

如果晋升到老年代的对象大小大于了老年代剩余的容量,则触发full gc,使用Parallel Old策略

```
1 2021-03-25T13:59:23.776-0800:
2 [Full GC (Ergonomics) //Ergonomics就是负责自动的调解gc暂停时间和吞吐量之间的平衡
3 [PSYoungGen: 37890K->0K(232960K)] //PS代表Parallel Scavenge 收集器 新生代直接归0
4 [ParOldGen: 607579K->323242K(699392K)] //Parallel Old收集器 老年代从607579K->323242K
5 645470K->323242K(932352K), //整个堆从607579K->323242K
6 [Metaspace: 2566K->2566K(1056768K)], //metaspace 受到jvm的管理 metaspace属于堆
7 0.0909903 secs] // stw 0.0909903
8 [Times: user=0.14 sys=0.01, real=0.09 secs]
```