

第 2 节课作业实践

1、本机使用 G1 GC 启动一个程序，仿照课上案例分析一下 JVM 情况

启动

```
java -XX:+UseG1GC -XX:MaxGCPauseMillis=50 -jar
```

使用 jmap 查看堆栈情况

```
jmap -heap 15868
```

```
C:\Program Files\Java\jdk1.8.0_25\bin>jmap -heap 15868
Attaching to process ID 15868, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 25.25-b02

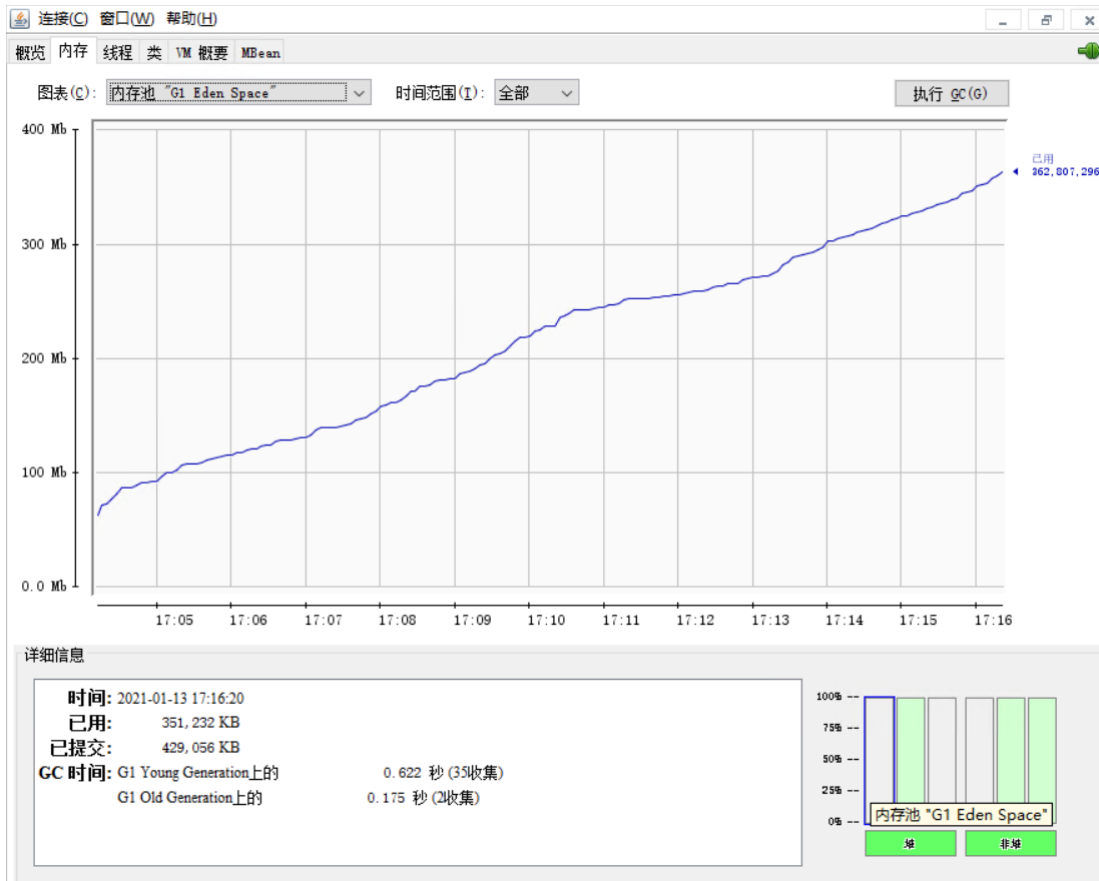
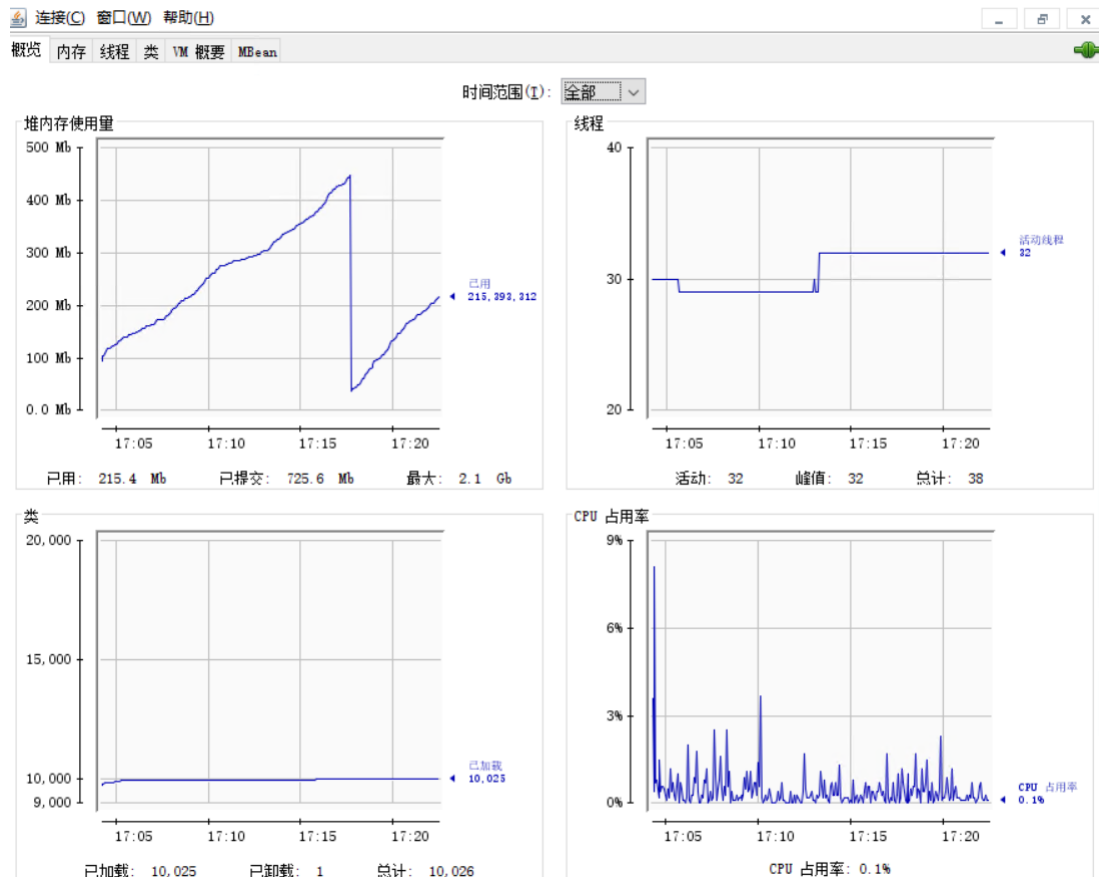
using thread-local object allocation.
Garbage-First (G1) GC with 4 thread(s)    使用G1 回收器 4线程

Heap Configuration:
  MinHeapFreeRatio          = 40
  MaxHeapFreeRatio          = 70
  MaxHeapSize                = 2147483648 (2048.0MB)    默认1/4的堆内存
  NewSize                   = 1363144 (1.2999954223632812MB)
  MaxNewSize                = 1287651328 (1228.0MB)
  OldSize                   = 5452592 (5.1999969482421875MB)    新生代占了一半多的空间
  NewRatio                  = 2
  SurvivorRatio             = 8
  MetaspaceSize             = 21807104 (20.796875MB)
  CompressedClassSpaceSize = 1073741824 (1024.0MB)
  MaxMetaspaceSize         = 17592186044415 MB
  G1HeapRegionSize          = 1048576 (1.0MB)

Heap Usage:
G1 Heap:
  regions = 2048    G1的regions 有2048个
  capacity = 725614592 (692.0MB)    一个占用1m
  used = 52339696 (49.91502380371094MB)
  free = 673274896 (642.0849761962891MB)
  7.213153728859962% used
G1 Young Generation:
Eden Space:
  regions = 19
  capacity = 439353344 (419.0MB)
  used = 19922944 (19.0MB)
  free = 419430400 (400.0MB)
  4.534606205250597% used
Survivor Space:    survivor 没有From 和 To之分了
  regions = 17
  capacity = 17825792 (17.0MB)
  used = 17825792 (17.0MB)
  free = 0 (0.0MB)
  100.0% used
G1 Old Generation:
  regions = 15
  capacity = 268435456 (256.0MB)
  used = 14590960 (13.915023803710938MB)
```

jconsole

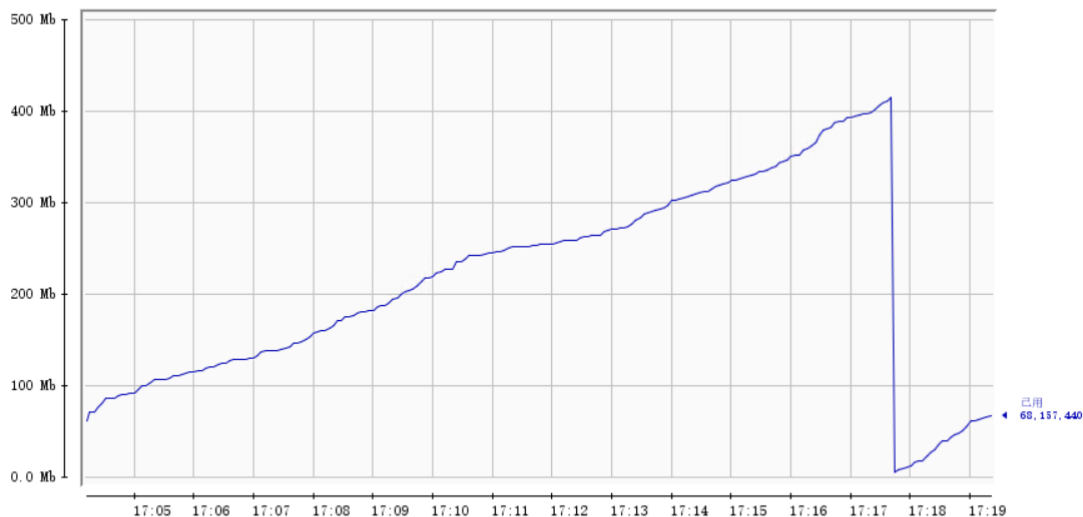
17:17的时候触发了full GC



图表(C): 内存池 "G1 Eden Space"

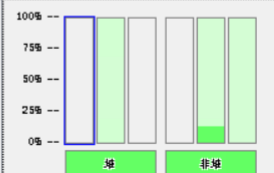
时间范围(T): 全部

执行 GC(G)



详细信息

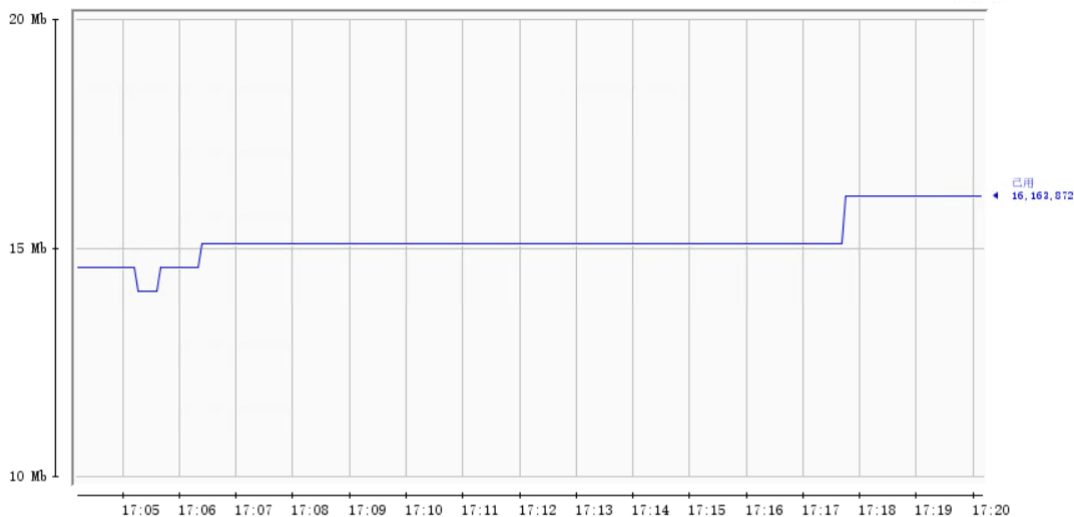
时间: 2021-01-13 17:19:20
已用: 65,536 KB
已提交: 431,104 KB
GC 时间: G1 Young Generation上的 0.646 秒 (36收集)
G1 Old Generation上的 0.175 秒 (2收集)



图表(C): 内存池 "G1 Old Gen"

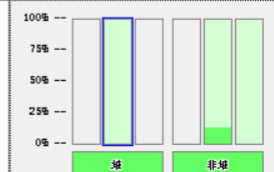
时间范围(T): 全部

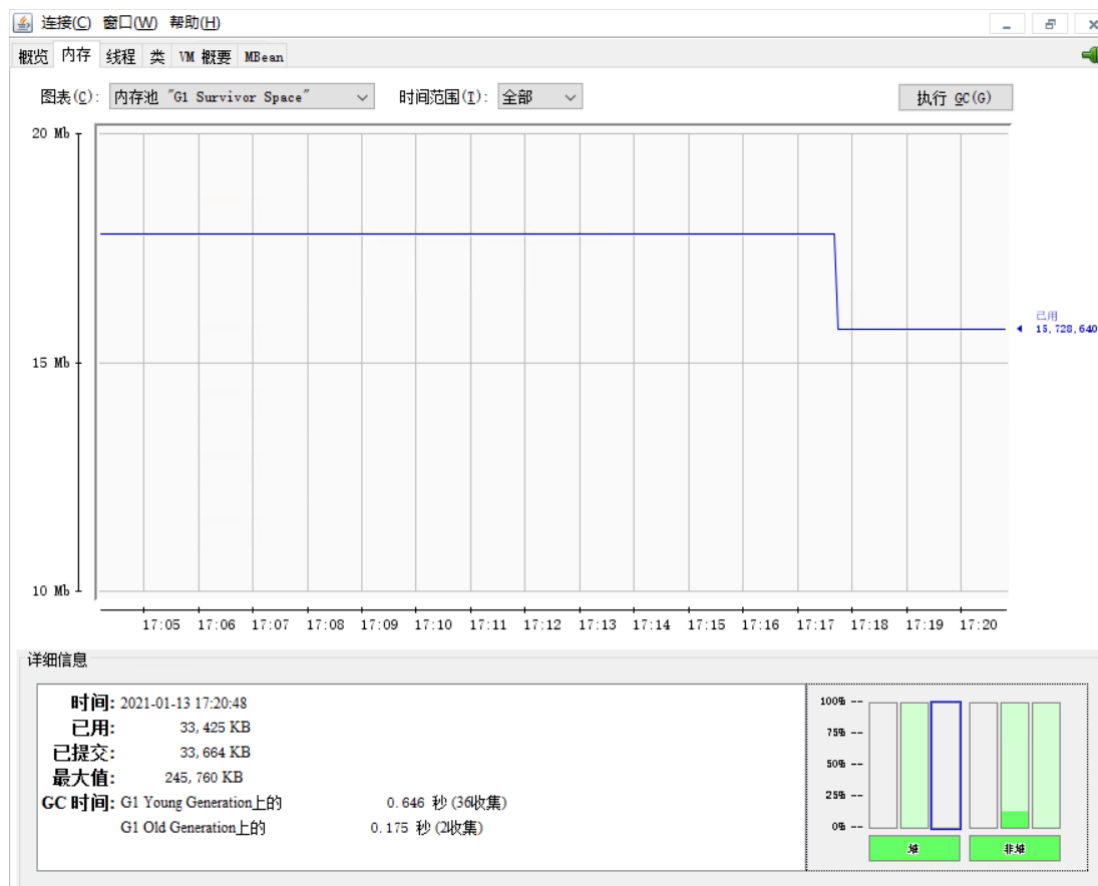
执行 GC(G)



详细信息

时间: 2021-01-13 17:20:08
已用: 15,785 KB
已提交: 262,144 KB
最大值: 2,097,152 KB
GC 时间: G1 Young Generation上的 0.646 秒 (36收集)
G1 Old Generation上的 0.175 秒 (2收集)





连接名称: pid: 15868 ymjs-api-0.0.1-SNAPSHOT.jar 运行时间: 21 分钟

虚拟机: Java HotSpot(TM) 64-Bit Server VM 版本 25.25-b02 进程 CPU 时间: 48.906 秒

供应商: Oracle Corporation JIT 编译器: HotSpot 64-Bit Tiered Compilers

名称: 15868@ecs-42bc-0004 总编译时间: 30.098 秒

活动线程: 32 已加装当前类: 10,025

峰值: 32 已加载类总数: 10,026

守护程序线程: 28 已卸载类总数: 1

启动的线程总数: 38

当前堆大小: 244,137 KB 提交的内存: 708,608 KB

最大堆大小: 2,097,152 KB 暂挂最终处理: 0 对象

垃圾收集器: 名称 = 'G1 Young Generation', 收集 = 36, 总花费时间 = 0.646 秒

垃圾收集器: 名称 = 'G1 Old Generation', 收集 = 2, 总花费时间 = 0.175 秒

操作系统: Windows Server 2012 R2 6.3 总物理内存: 8,387,700 KB

体系结构: amd64 空闲物理内存: 2,942,092 KB

处理程序数: 4 总交换空间: 8,690,468 KB

提交的虚拟内存: 974,444 KB 空闲交换空间: 2,695,688 KB

VM 参数: -XX:+UseG1GC -XX:MaxGCPauseMillis=50

类路径: ymjs-api-0.0.1-SNAPSHOT.jar

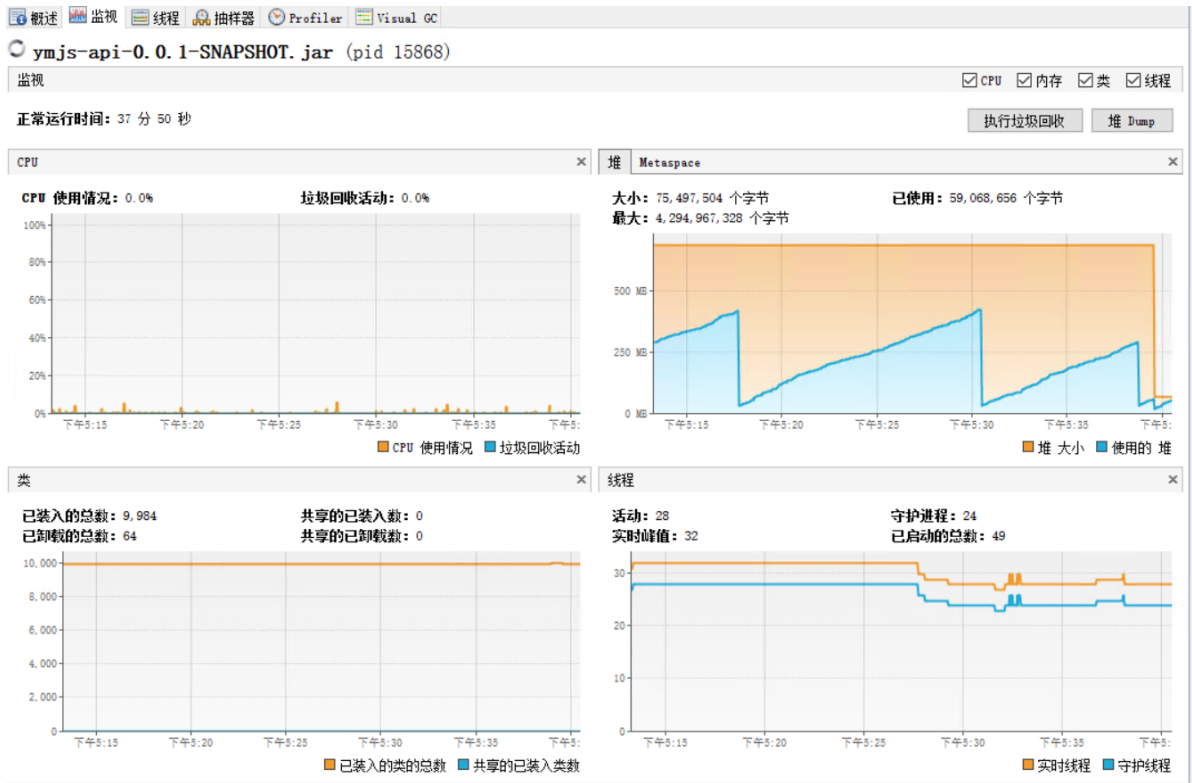
库路径: C:\ProgramData\Oracle\Java\javapath;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0;D:\soft\nodejs;D:\soft\mysql-5.7.24\bin;C:\Program Files\Java\jdk1.8.0_25\bin;C:\Users\Administrator\AppData\Local\Microsoft\WindowsApps;C:\Users\Administrator\AppData\Roaming\npm;D:\soft\mysql-5.7.24\bin;

引导类路径: C:\Program Files\Java\jre1.8.0_25\lib\resources.jar;C:\Program Files\Java\jre1.8.0_25\lib\rt.jar;C:\Program Files\Java\jre1.8.0_25\lib\sunrsasign.jar;C:\Program Files\Java\jre1.8.0_25\lib\jsse.jar;C:\Program Files\Java\jre1.8.0_25\lib\jce.jar;C:\Program Files\Java\jre1.8.0_25\lib\charsets.jar;C:\Program Files\Java\jre1.8.0_25\lib\jfr.jar;C:\Program Files\Java\jre1.8.0_25\classes

jvisualvm

可以看到 eden区 survivor区 Old区 的大小都是一样的

GC 触发的很频繁，Eden区到400M的时候触发了一次fullGC



抽样器 创建的最多的类是 Object int[] char[] 和 TreeMap

ymjs-api-0.0.1-SNAPSHOT.jar (pid 15532)

抽样器

抽样: CPU 内存 停止

状态: 抽样处于非活动状态

堆柱状图 PermGen 柱状图 每个线程分配

类: 3,787 实例: 1,689,328 字节: 75,367,936

类名	字节 [K]	字节	实例数
java.lang.Object[]	14,...	(19.3%)	443,...
int[]	9,9...	(13.2%)	11,...
char[]	9,7...	(12.9%)	100,...
java.util.TreeMap\$Entry	7,1...	(9.4%)	178,...
java.io.ObjectStreamClass\$WeakClassKey	5,6...	(7.4%)	175,...
byte[]	3,3...	(4.4%)	15,...
java.lang.String	1,9...	(2.5%)	81,...

老师没有演示，不大会用，只能大概看到都是绿色的

