

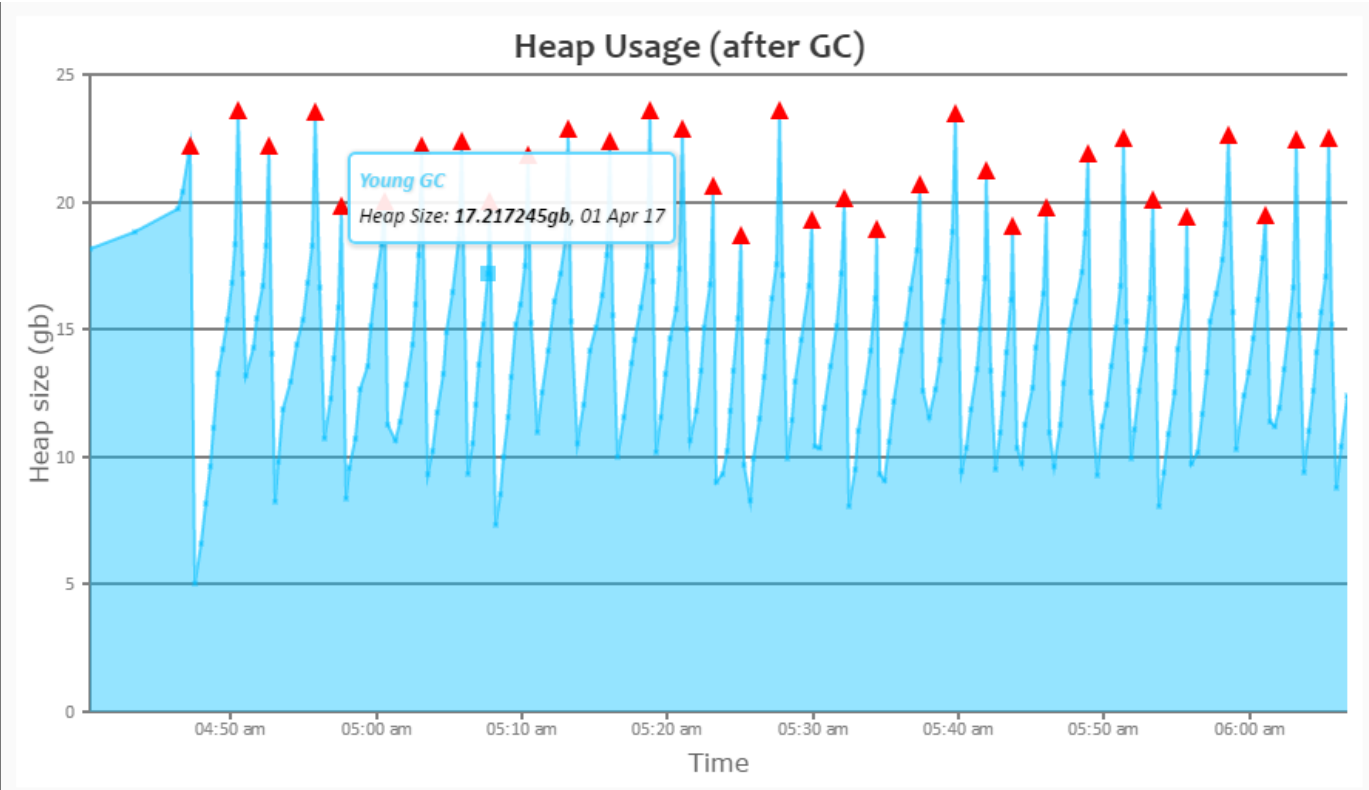
调优笔记

描述

2017年4月25日 线上发生报警，现状是 服务间接性质无响应, 预警策略 1分钟内无响应，3次超过2次报警；

分析：

通过 gc.log 分析 得出FullGC发生的频率很高,具体发生的频率如下图：



反映说明了GC回收出现了问题，具体细节表现在，GC的创建对象频率太高，其实主要的是 promoted bytes 这里的数字有点大，证明老年代的空间越来越小

所以触发的FULL GC。

Total created bytes ?	1.63 tb
Total promoted bytes ?	339.02 gb
Avg creation rate ?	330.48 mb/sec
Avg promotion rate ?	67.03 mb/sec

但是这里并不能 直接 反映问题，离线分析日志得知；

```
17403.109: [GC17403.109: [ParNew
Desired survivor size 536870912 bytes, new threshold 1 (max 6)
- age 1: 1073740712 bytes, 1073740712 total
: 9437184K->1048576K(9437184K), 2.8208850 secs]
24481244K->17676490K(32505856K), 2.8210520 secs] [Times: user=9.14
sys=0.02, real=2.82 secs]
17405.930: Total time for which application threads were stopped: 2.8215420
seconds
```

1073740712 > 536870912

这里的数据需要拷贝 survivor，已经不能满足，所以会有一部分溢出到老年代，升级的 age 已经被强制的调整成为了 1；这里要说明下之前用的 CMS 垃圾回收器，具体的 flag 为；

```
JAVA_OPTS=" -Xms32g -Xmx32g -Xmn10g -Xss256k -server -XX:PermSize=128M
-XX:MaxPermSize=128M -XX:-UseSplitVerifier -XX:+UseConcMarkSweepGC
-XX:+UseParNewGC -XX:+CMSClassUnloadingEnabled
-XX:+CMSParallelRemarkEnabled -XX:+UseCMSCompactAtFullCollection
-XX:CMSFullGCsBeforeCompaction=5 -XX:CMSMaxAbortablePrecleanTime=5
-XX:MaxGCPauseMillis=300 -XX:+DisableExplicitGC
-XX:+HeapDumpOnOutOfMemoryError -XX:+PrintTenuringDistribution -verbose:gc
-XX:+PrintGCDetails -XX:+PrintGCApplicationStoppedTime
-XX:+PrintGCApplicationConcurrentTime -XX:+PrintGCTimeStamps
-Xloggc:/tmp/doss.gc.log -Duser.home=/home/work"
```

所以从 2 个方面来考虑

1. 增加 GC thread，带来的因素是 CPU 也会升高；
2. 降低创建对象的速率；

所以本次调整为升级 GC 垃圾回收为 G1，设定相关参数 GC thread；与此同时在调低线程数来降低创建对象的频率；

具体 GC 参数配置为 tomcat/bin/catalina.sh：

```
JAVA_OPTS=" -server -Xms32G -Xmx32G -XX:PermSize=512m -XX:+UseG1GC
-XX:MaxGCPauseMillis=200 -XX:ParallelGCThreads=20 -XX:ConcGCThreads=10
-XX:InitiatingHeapOccupancyPercent=70 -XX:-UseSplitVerifier
-XX:+DisableExplicitGC -XX:+HeapDumpOnOutOfMemoryError
-XX:+PrintTenuringDistribution -verbose:gc -XX:+PrintGCDetails
-XX:+PrintGCApplicationStoppedTime -XX:+PrintGCApplicationConcurrentTime
-XX:+PrintGCDateStamps -Xloggc:/tmp/gc.log -Duser.home=/home/work "
```

注：

线上 APK 打包和配置升级都有发生；

升级线程数有原来的 10 降低为 3；

```
job.upgrade.task.thread.pool.size
```

打包有原来的20 降低为10;

```
pack.thread.pool.size=10
```

这些数据都是参考，具体细节都需要在线下验证，线上APK的打包数为 5000;