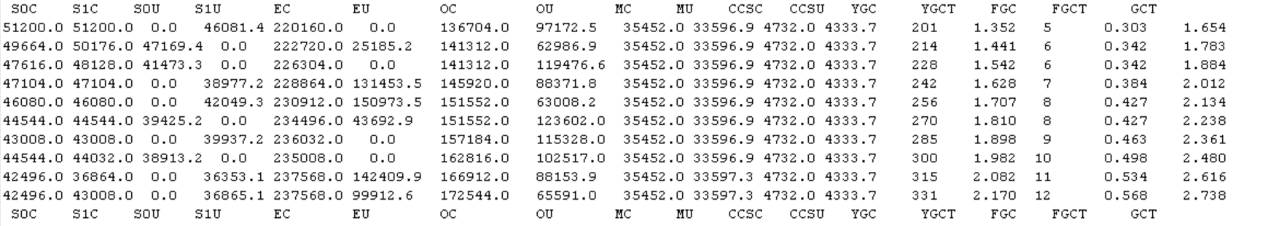
**GC日志监控**

1000毫秒打印一次gc日志，每10笔显示表头

jstat -gc -h10 $(jcmd | grep "performance-1.1.0.jar" | awk '{print $1}') 1000

1. 默认值

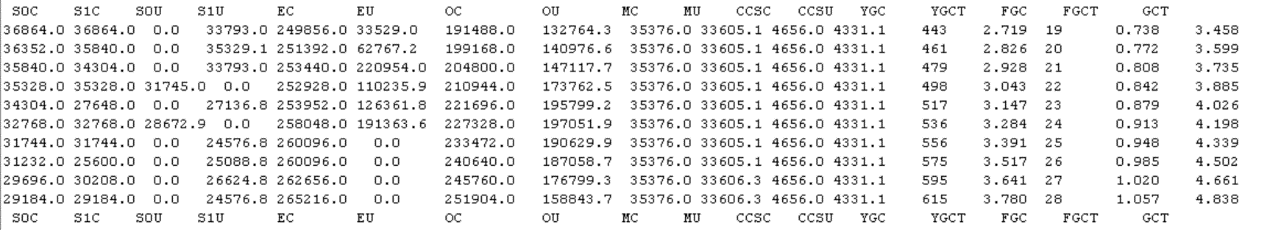
java -Xloggc:/usr/local/myapp/gc0.log -jar performance-1.1.0.jar



监控结果：10秒内130YGC(新生代GC活动的数量)，7次FGC,总耗时1.08秒

2、调大-XX:ParallelGCThreads=4

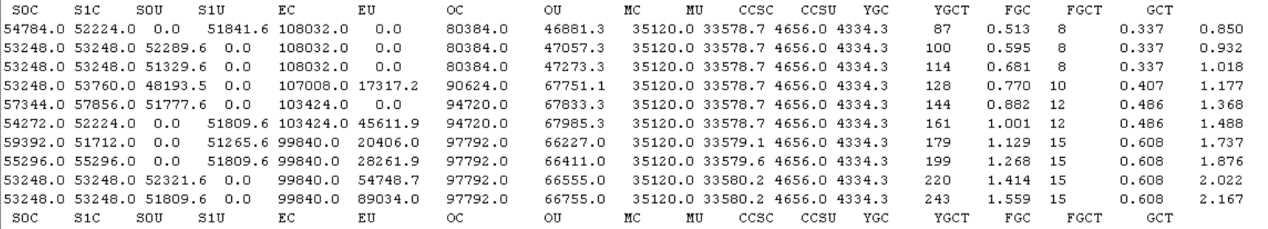
java -Xloggc:/usr/local/myapp/gc1.log -XX:ParallelGCThreads=4 -jar performance-1.1.0.jar



监控结果：10秒内172YGC(新生代GC活动的数量)，9次FGC,总耗时1.38秒

3、降低耗时，设置-XX:MaxGCPauseMills=10

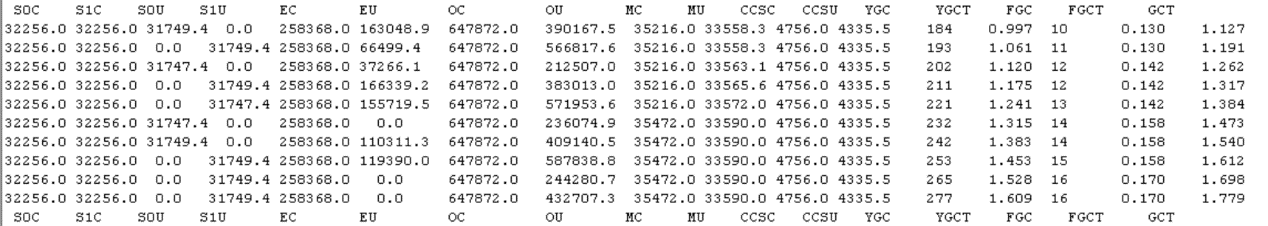
java -Xloggc:/usr/local/myapp/gc3.log -XX:MaxGCPauseMillis=10 -jar performance-1.1.0.jar



监控结果：10秒内172YGC(新生代GC活动的数量)，7次FGC,总耗时1.3秒

4、改用CMS回收器

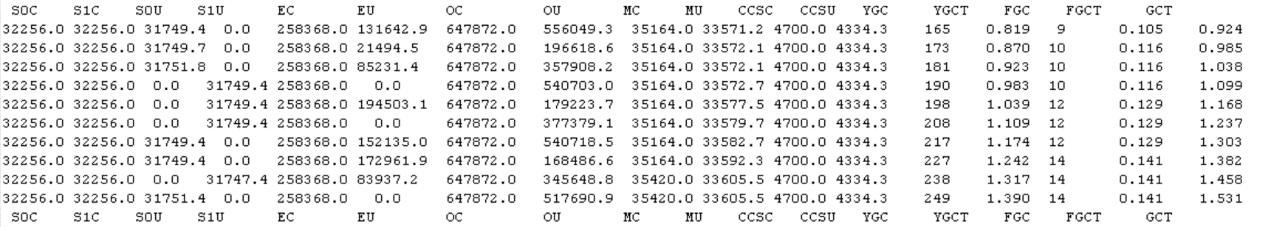
java -Xloggc:/usr/local/myapp/gc4.log -XX:+UseConcMarkSweepGC -jar performance-1.1.0.jar



监控结果：10秒内93YGC(新生代GC活动的数量)，6次FGC,总耗时0.65秒

5、增加线程

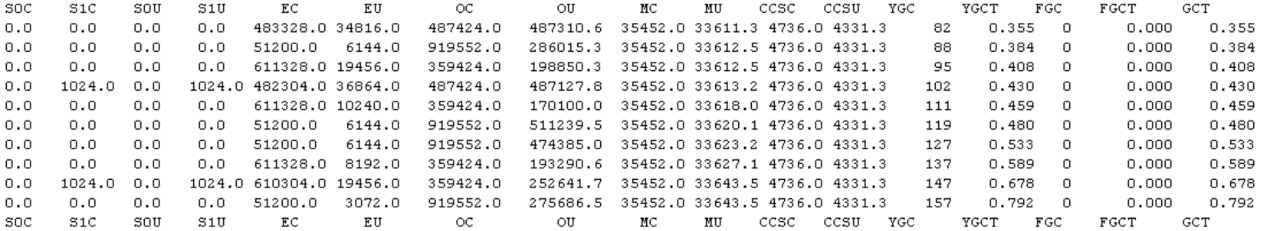
java -Xloggc:/usr/local/myapp/gc5.log -XX:+UseConcMarkSweepGC -XX:+ConcGCThreads=3 -jar performance-1.1.0.jar



监控结果：10秒内84YGC(新生代GC活动的数量)，5次FGC,总耗时0.61秒

6、改用G1

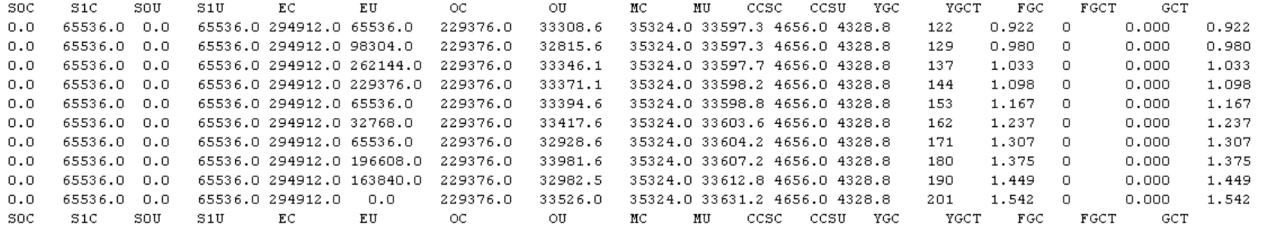
java -Xloggc:/usr/local/myapp/gc6.log -XX:+UseG1GC -jar performance-1.1.0.jar



监控结果：10秒内75YGC(新生代GC活动的数量)，0次FGC,总耗时0.34秒

7、增加分区大小

java -Xloggc:/usr/local/myapp/gc7.log -XX:+UseG1GC -XX:G1HeapRegionSize=64m -jar performance-1.1.0.jar



监控结果：10秒内79YGC(新生代GC活动的数量)，0次FGC,总耗时0.64秒

**结论：响应优先的话，使用G1回收器。**