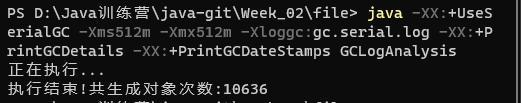
GCLogAnalysis.java GC分析

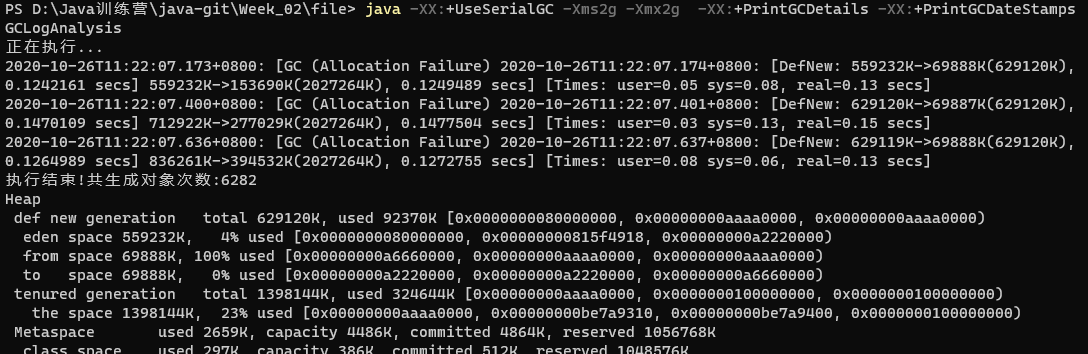
1. 串行GC

java -XX:+UseSerialGC -Xms512m -Xmx512m -Xloggc:gc.serial.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



堆大小改为128m,256m开始出现oom

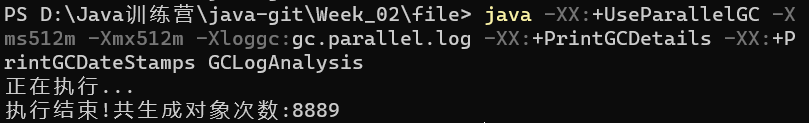
java -XX:+UseSerialGC -Xms2g -Xmx2g -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



内存大GC时间长，吞吐量减少

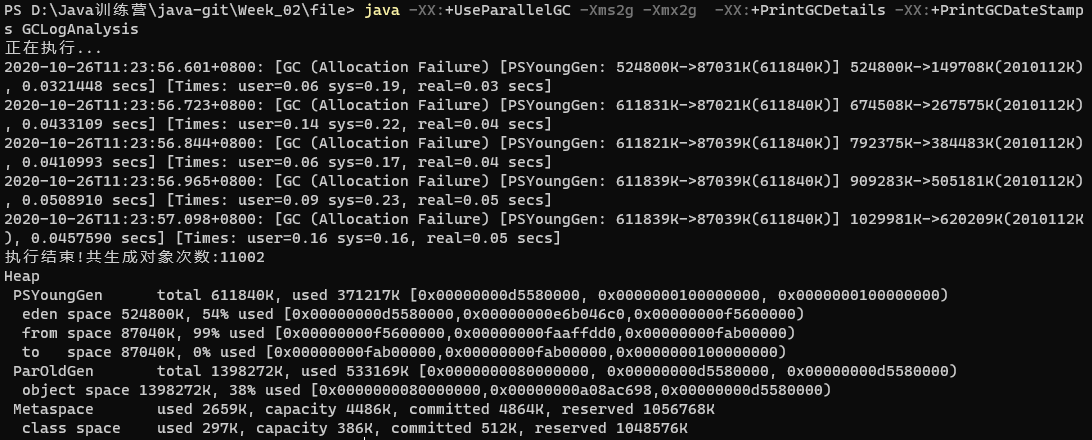
1. 并行GC

java -XX:+UseParallelGC -Xms512m -Xmx512m -Xloggc:gc.parallel.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



512内存，效率不高,应该是CPU相互抢资源。

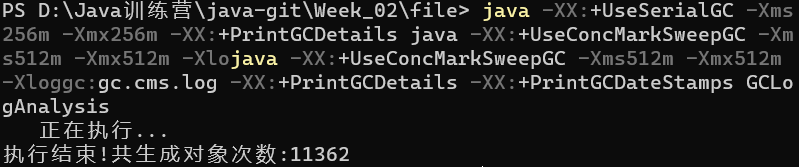
java -XX:+UseParallelGC -Xms2g -Xmx2g -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



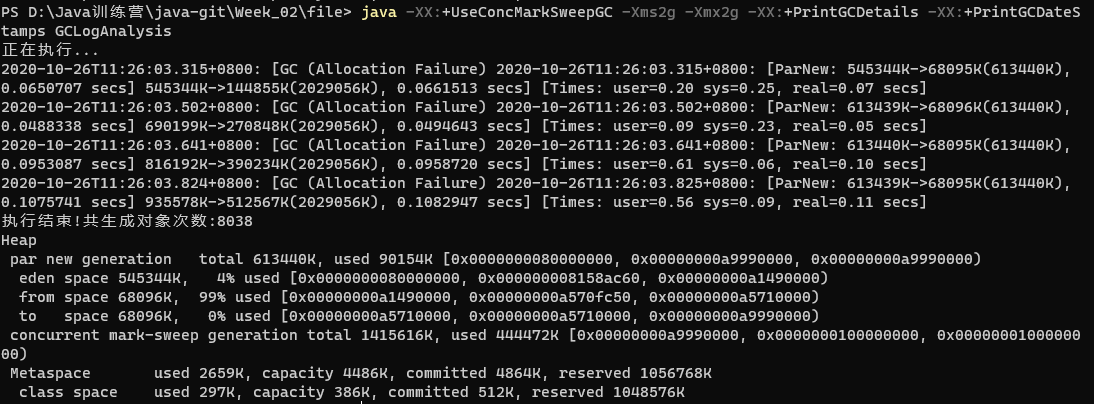
内存大优势提升

1. CMS GC

java -XX:+UseConcMarkSweepGC -Xms512m -Xmx512m -Xloggc:gc.cms.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



java -XX:+UseConcMarkSweepGC -Xms2g -Xmx2g -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



CPU争抢资源

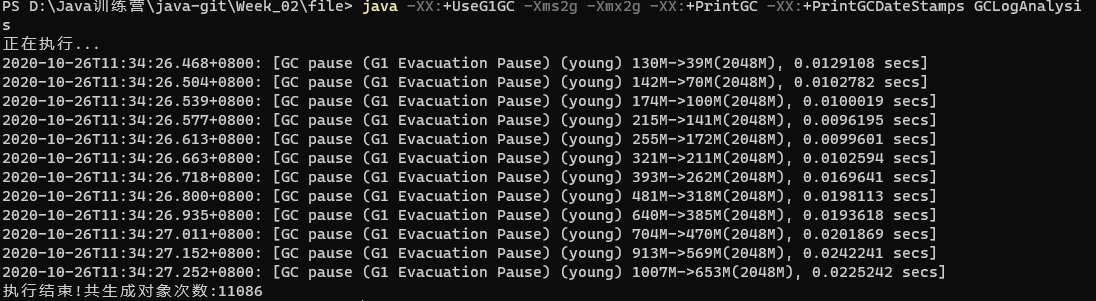
1. G1 GC

java -XX:+UseG1GC -Xms512m -Xmx512m -Xloggc:gc.g1.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



所有gc在多核服务器的条件下，相差不大

java -XX:+UseG1GC -Xms2g -Xmx2g -XX:+PrintGCDetails -XX:+PrintGCDateStamps GCLogAnalysis



内存越大，效率越高

正常情况，串行的效率最低，其他效率会更高

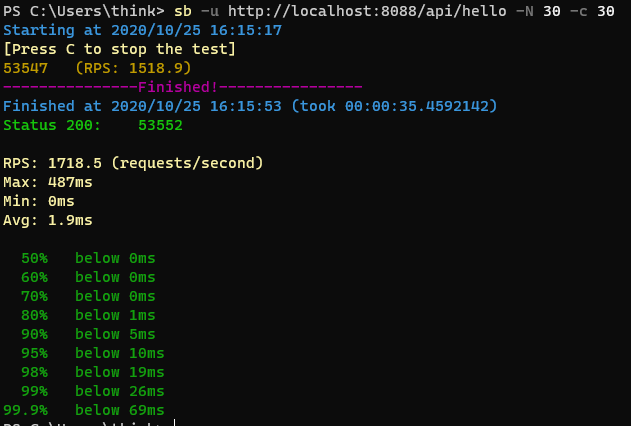
压测

启动gw

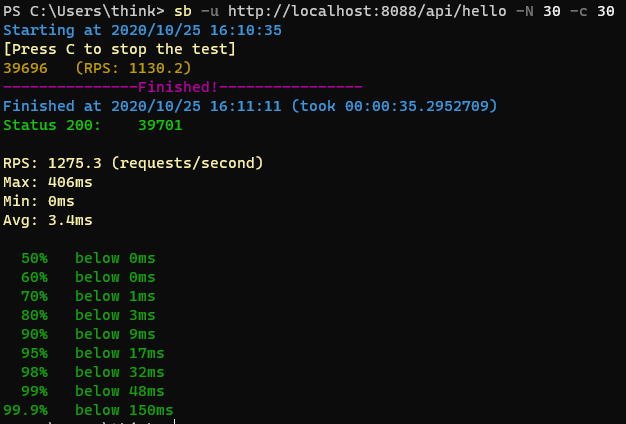
Java -jar -Xms512m -Xmx512m gateway-server-0.0.1-SNAPSHOT.jar

Sb -u <http://localhost:8088/api/hello> -N 30 -c 30

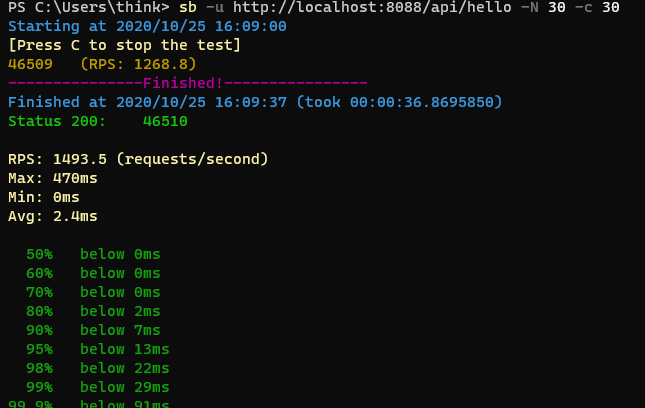
SerialGC



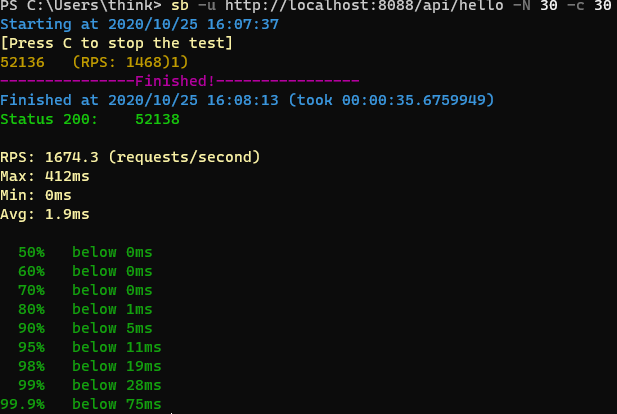
CMSGC



并行gc



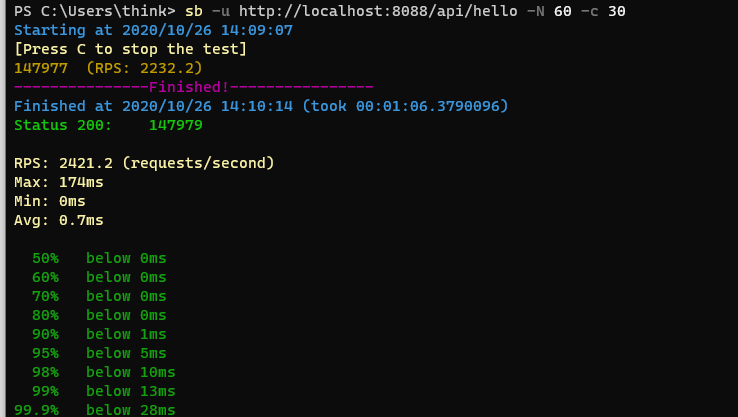
G1GC



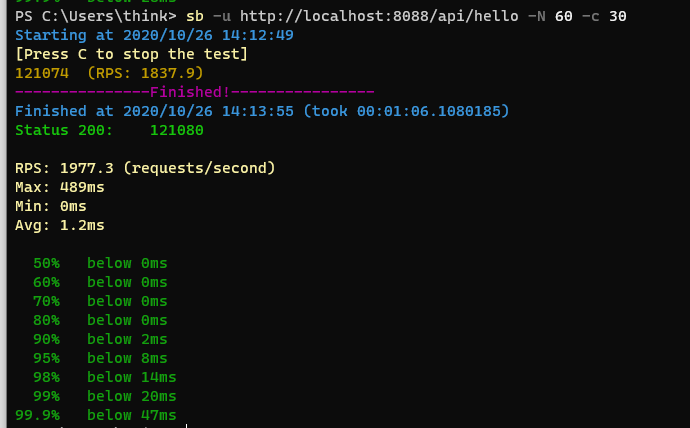
Java -jar -Xms1g -Xmx1g gateway-server-0.0.1-SNAPSHOT.jar

sb -u <http://localhost:8088/api/hello> -c 30 -N 60

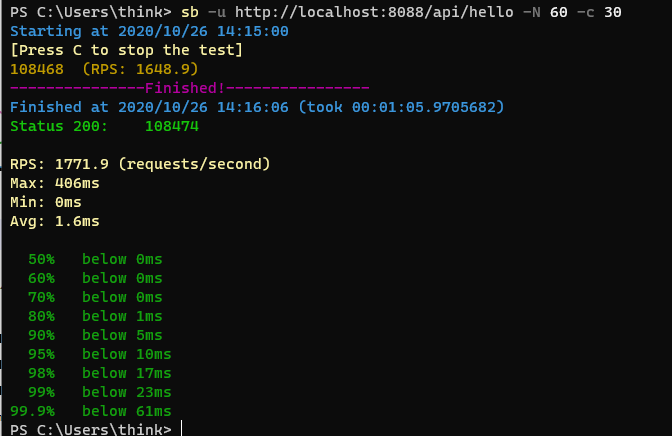
Parallel GC



Java -jar -Xms1g -Xmx1g -XX:+UseSerialGC gateway-server-0.0.1-SNAPSHOT.jar



Java -jar -Xms1g -Xmx1g -XX:+UseConsMarkSweepGC gateway-server-0.0.1-SNAPSHOT.jar



Java -jar -Xms1g -Xmx1g -XX:+UseG1GC gateway-server-0.0.1-SNAPSHOT.jar

