第三课作业

A	B	C	D	E	F
java +GC策略 -Xmx128m -XX:+PrintGCDetails GCLogAnalysis	Xmx=128M	Xmx=512M	Xmx=1024M	Xmx=2048M	Xmx=4096M
串行GC，-XX:+UseSerialGC	连续的Full GC 最后直接OOM	生成对象：7891 15 次MinorGC 6 次Major GC Major暂停时间 0.058	生成对象：7796 14 次MinorGC 3 次MajorGC Major暂停时间 0.07	生成对象：7968 15 次MinorGC 3 次MajorGC Major暂停时间 0.063	生成对象：7646 15 次MinorGC 3 次MajorGC Major暂停时间 0.073
并行GC，-XX:+UseParallelGC	连续的Full GC 最后直接OOM	生成对象：8249 32 次MinorGC 7 次MajorGC Major暂停时间 0.0314	生成对象：8893 19 次MinorGC 3 次MajorGC Major暂停时间 0.036	生成对象：9214 10 次MinorGC 3 次MajorGC Major暂停时间 0.036	生成对象：9628 8 次MinorGC 3次MajorGC Major暂停时间 0.04
CMS GC，-XX:+UseConcMarkSweepGC	连续的Full GC 最后直接OOM	生成对象次数：8210 18 次MinorGC 3次MajorGC Major暂停时间 0.0533	生成对象：8034 30 次MinorGC 1 次MajorGC Major暂停时间 0.01	生成对象：7919 29 次MinorGC 1 次MajorGC Major暂停时间 0.01	生成对象：7783 29 次MinorGC 1 次MajorGC Major暂停时间 0.04
G1 GC，-XX:+UseG1GC	连续的Full GC 最后直接OOM	生成对象次数：8097 37 次MinorGC 15 次MajorGC Major暂停时间 0.0027	生成对象：7569 13 次MinorGC 3 次MajorGC Major暂停时间 0.0039	生成对象：5557 12 次MinorGC 0次MajorGC Major暂停时间0	生成对象：4841 10次MinorGC 0次MajorGC Major暂停时间0

串行的吞吐量【626/秒】：

```
java -XX:+UseSerialGC -jar gateway-server-0.0.1-
SNAPSHOT.jar
```

```
Transfer/sec:      1.89MB
(base) haowang@192 ~ $ wrk -t40 -c200 -d60s --latency "http://127.0.0.1:8088/api/hello"
Running 1m test @ http://127.0.0.1:8088/api/hello
  40 threads and 200 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency    14.11ms   21.28ms 264.93ms   89.86%
    Req/Sec   626.08    206.83     2.86k    75.11%
  Latency Distribution
     50%    6.31ms
     75%   14.26ms
     90%   35.88ms
     99%  106.80ms
  1495511 requests in 1.00m, 178.55MB read
Requests/sec:  24882.27
Transfer/sec:      2.97MB
```

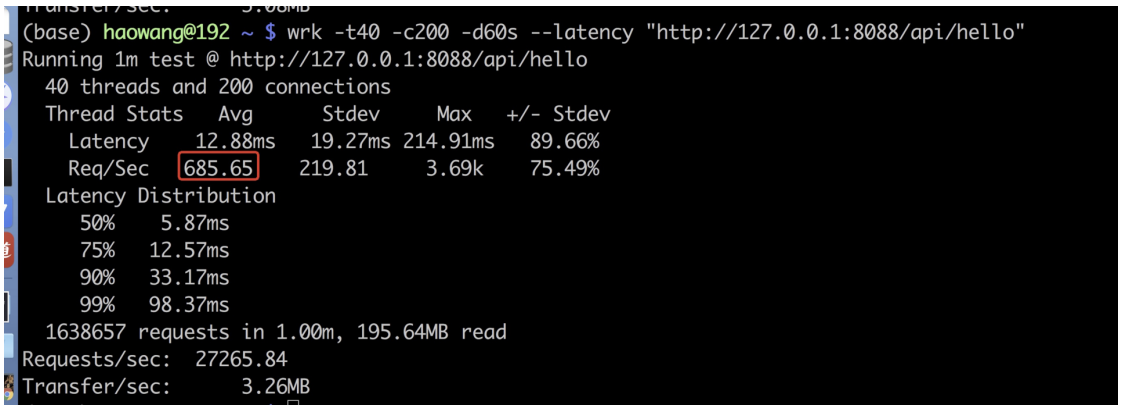
并行的吞吐量【722/秒】：

```
java -XX:+UseParallelGC -jar gateway-server-0.0.1-
SNAPSHOT.jar
```

```
Transfer/sec:      3.27MB
(base) haowang@192 ~ $ wrk -t40 -c200 -d60s --latency "http://127.0.0.1:8088/api/hello"
Running 1m test @ http://127.0.0.1:8088/api/hello
  40 threads and 200 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency    14.85ms   29.05ms 458.83ms   91.26%
    Req/Sec   722.11    230.16     3.29k    76.21%
  Latency Distribution
     50%    5.32ms
     75%   12.51ms
     90%   38.40ms
     99%  120.77ms
  1714457 requests in 1.00m, 204.69MB read
Requests/sec:  28531.01
Transfer/sec:      3.41MB
```

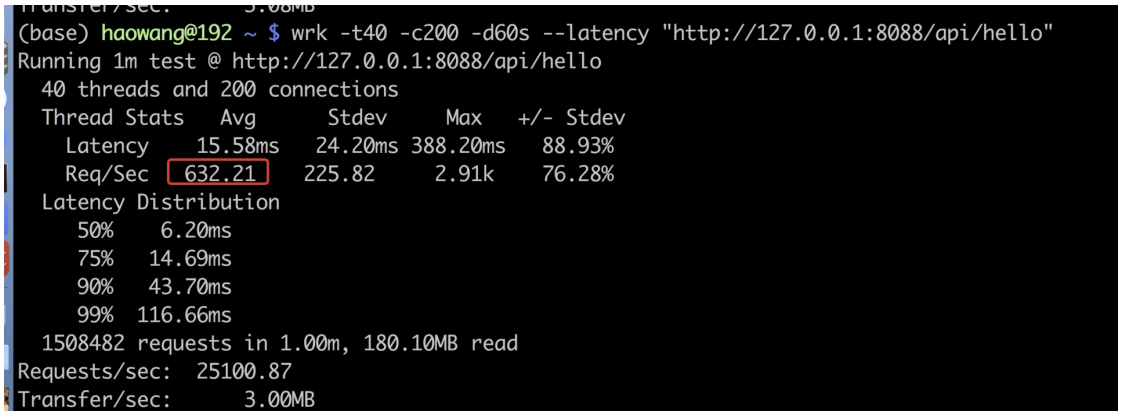
CMS的吞吐量【685/秒】

```
java -XX:+UseConcMarkSweepGC -jar gateway-server-0.0.1-
SNAPSHOT.jar
```



G1的吞吐量【632/秒】

```
java -XX:+UseG1GC -jar gateway-server-0.0.1-SNAPSHOT.jar
```



共同的特征：

1、堆内存为128M的时候，都发生了OOM。

2、随着堆内存的增加，MinorGC的次数减少，Major的次数也减少。

3、随着堆内存的增加，除了并行GC之外，其余的策略，随着内存的增加，生成的对象不变，或对象减少。

独有的特征：

1、串行GC生成对象，MinorGC，MajorGC，暂停时间，并没有随着堆内存的增大而发生明显变化。

2、并行GC随着堆内存的增加，生成对象增加，吞吐量增加，且并行GC在这些GC策略当中吞吐量是最高的。MinorGC和MajorGC都在明显增加。

生成对象增加意味着业务保证能力增加。

所以这也是在JDK1.8之前默认使用并行GC策略的标准。

3、CMS MinorGC随着堆内存增加，而增加，MajorGC随着堆内存的增加，而减少。

4、G1 GC的暂停时间，随着堆内存的增加，暂停时间减少。且生成的对象个数急剧减少。