

HOMEWORK REVIEW

PERFORMANCE 1

基准测试

测试？

性能需求

以吞吐量和延迟性需求举例：

- 应用预期的吞吐量是多少？
- 请求和响应之间的延迟预期是多少？
- 应用支持多少并发用户或者并发任务？
- 当并发用户数或并发任务书达到最大时，可接受的吞吐量和延迟是多少？
- 最差情况下的延迟是多少？
- 要是垃圾收集引入的延迟在可容忍范围之内，垃圾收集的频率应该是多少？

查看启动参数

- `$ java -XX:+PrintCommandLineFlags -version`
- `$ java -XX:+PrintFlagsFinal -version`
- `$ jinfo -flag MaxHeapSize <pid>`

客户端还是服务器

- 至少2GB物理内存
- 至少2个虚拟处理器

预热阶段

JIT优化时机

Server	Client
--------	--------

10000	1500
-------	------

JAVA TIME

精度问题

- `System.currentTimeMillis()`
- `System.nanoTime()`

剔除无效代码

```
for (int i = 1; i <= 100; i++) {  
    student.say(i);  
}
```

注意事项

- 可重复

实验设计

- 操作系统
- CPU、内存等硬件
- JVM版本
- 参数
- SUT
- 期望

性能监控

工具

- JConsole
- VisualVM
- Java Mission Control

JAVA MISSION CONTROL

- 本地JVM
- 远程JVM

```
-Dcom.sun.management.jmxremote.port=8999  
-Dcom.sun.management.jmxremote.ssl=false  
-Dcom.sun.management.jmxremote.authenticate=false
```

EXERCISE 1

JMC监控

- 更新代码
- `$ cd 6-performance-1`
- `$ sbt -mem 512`
- `> test:run`

Windows

- exit all Java applications
- delete the %TMP%\hsperfdata_username directory
- create new %TMP%\hsperfdata_UserName directory.

性能分析

JAVA FLIGHT RECORDER

- `$ JAVA_OPTS="-XX:+UnlockCommercialFeatures -XX:+FlightRecorder -XX:StartFlightRecording=duration=300s,filename=myrecording.jfr"`
`sbt clean test:run -mem 512`
- `$ JAVA_OPTS="-XX:+UnlockCommercialFeatures -XX:+FlightRecorder"`
`sbt -mem 512`

性能调优

性能优化机会

- 使用更高效的算法
- 减少锁竞争
- 为算法生成更有效率的代码

- 内核态CPU使用
- 锁竞争
- 调整数据结构的大小

循环次数	线程数	改动	时间1	时间2	时间3	平均时间
100k	4	无	155	136	134	142
100k	4	BufferedWriter	150	120	125	132

EXERCISE 2

优化FizzGame性能

课后练习

- 优化FizzGame性能
- 包含优化前后对比文档
- 发Merge Request

参考资料

- [Java性能优化权威指南](#)
- [Java Mission Control User's Guide](#)
- [Java Flight Recorder Runtime Guide](#)
- [VisualVM Troubleshooting Guide](#)