

03-JVM虚拟机

刘亚雄

极客时间-Java 讲师



今日目标

1. 熟悉JVM相关工具使用：JDK工具，第三方
 - jps、jstat、jinfo、jhat、jmap、VisualVM
 - GCEasy、MAT、Arthas
2. 掌握JVM调优基本理论
3. 理解GC日志含义
4. 掌握JVM调优实践案例
5. 掌握检测死锁与内存溢出的定位与分析方法

六、JVM相关工具

6.1 JDK工具包

- ① **jps: JVM Process status tool**: JVM进程状态工具, 查看进程基本信息
- ② **jstat: JVM statistics monitoring tool**: JVM统计监控工具, 查看堆, GC详细信息
- ③ **jinfo: Java Configuration Info**: 查看配置参数信息, 支持部分参数运行时修改
- ④ **jmap: Java Memory Map**: 分析堆内存工具, dump堆内存快照
- ⑤ **jhat: Java Heap Analysis Tool**: 堆内存dump文件解析工具
- ⑥ **jstack: Java Stack Trace**: Java堆栈跟踪工具

命令行

- ⑦ **VisualVM**: 性能分析可视化工具

GUI



做几个案例看一下

6.2 第三方工具

- ① **GCEasy**: 免费GC日志可视化分析Web工具
- ② **MAT: Memory Analyzer Tool** 可视化内存分析工具
- ③ **GCViewer**: 开源的GC日志分析工具
- ④ **Arthas**: 线上Java程序诊断工具, 功能非常强大



少年, 你想成为解Bug小王子吗?

6.3 JVM参数

JVM主要分三种：标准参数、非标准参数、不稳定参数

标准参数：不会随着JVM变化而变化

- 以**-开头**，如：-version、-jar
- 使用java -help查看

非标准参数：可能会随着JVM变化而变化，变化较小

- 以**-X开头**，如：-Xmx、-Xms、-Xmn、-Xss
- 使用java -X查看

不稳定参数：也是非标准的参数，主要用于JVM调优与Debug

- 以**-XX开头**，如：-XX:+ UseG1GC 、-XX:+UseParallelGC、-XX:+PrintGCDetails
- 分三类：性能参数、行为参数、和调试参数



下面我们进入调优环节

七、JVM调优实践

7.1 JVM调优

01-为什么JVM调优？

- 降本增效：**调优的最终目的都是为了应用程序使用最小的硬件消耗来承载更大的吞吐量。**

02-什么时候JVM调优？

- 系统吞吐量下降，或P90、P99响应增加
- 堆内存持续上涨，到出现OOM
- 频繁FullGC、GC停顿过长
- 堆内存占用过高

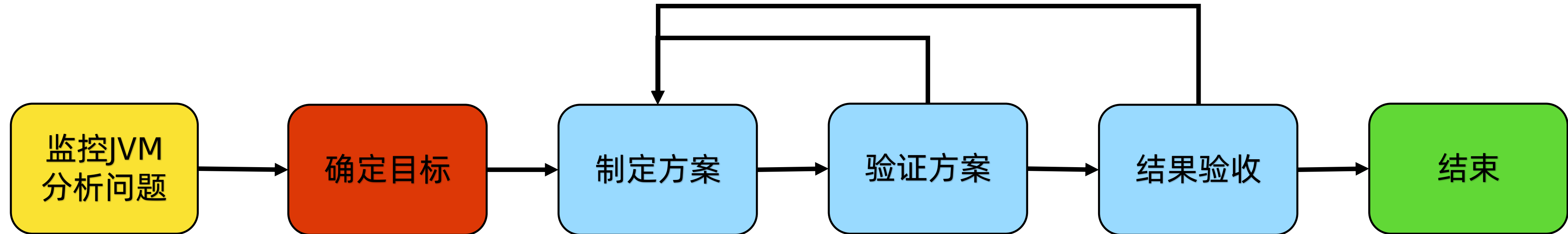
03-调优调个啥？

- 内存合理分配与使用、垃圾收集器的选配

04-调优的原则和目标如何确定？

- **优先原则：**产品、架构、代码、数据库优先，JVM是最后的手段
- **观测性原则：**发现问题解决问题，没问题不创造问题。

7.2 JVM调优主要步骤



- ① **监控JVM分析问题**: 评估必要性, 内存使用, GC频率, GC耗时
- ② **确定目标**: 内存占用, 响应延迟
- ③ **制定方案**: 配置内存及GC相关参数
- ④ **验证方案**: 测试环境对比方案前后差异, 确定是否生效
- ⑤ **结果验收**: 灰度测试, 全量发布



怎么监控JVM呢?

7.3 GC日志

JVM调优典型参数设置:

- -Xms、-Xmx、-Xss
- -XX:+UseParallelGC、-XX:UseParallelOldGC
- -XX:+UseParNewGC、-XX:+UseConcMarkSweepGC
- -XX:+UseG1GC



看日志很累?

JVM设置堆内存（Eden、S、Old），GC收集器，应该怎么配呢？

```
1 # 配置GC日志输出
2 JAVA_OPTS="${JAVA_OPTS} -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:+PrintHeapAtGC -
  Xloggc:${BASE_DIR}/logs/gc-default.log "
```



```
1 2021-05-18T14:31:11.340+0800: 2.340: [GC (Allocation Failure) [PSYoungGen: 896512K->41519K(1045504K)] 896512K-
  41543K(3435008K), 0.0931965 secs] [Times: user=0.14 sys=0.02, real=0.10 secs]
```



```
1 2021-05-19T14:46:07.367+0800: 1.562: [Full GC (Metadata GC Threshold)[PSYoungGen: 18640K->0K(1835008K)] [ParOldGen: 16K-
  >18327K(1538048K)] 18656K->18327K(3373056K), [Metaspace: 20401K->20398K(1069056K)], 0.0624559 secs] [Times: user=0.19
  sys=0.00, real=0.06 secs]
```

7.4 JVM调优-请看课堂笔记

01-堆内存与元空间优化

02-线程堆栈优化

03-垃圾收集器：吞吐量优先PS + PO

04-垃圾收集器：响应优先ParNew+CMS

05-垃圾收集器：全功能G1

06-案例01-内存溢出的定位与分析

07-案例02-检测死锁

THANKS

 极客时间 | 训练营

教育不是注满一桶水，而是点燃一把火