

# 「一入 Java 深似海」系列课程 - 第六期

## 第四节：G1 垃圾收集器调优（下）

小马哥 @mercyblitz



# 「一入 Java 深似海」系列课程

- 讲师信息

小马哥，Java 劝退师，Apache 和 Spring Cloud 等知名开源架构成员。

- 微信：mercyblitz-1985
- GitHub： <https://github.com/mercyblitz>
- 课件资源： <https://github.com/mercyblitz/segmentfault-lessons/>
- 博客： <https://mercyblitz.github.io/>



小马哥 VIP 交流群

扫一扫二维码，加入群聊。



# 「一入 Java 深似海」系列课程

- 小马哥书籍预售 《Spring Boot 编程思想 - 核心篇》

以 Spring Boot 2.0 为讨论的主线，讨论的范围将涵盖 Spring Boot 1.x 的所有版本，以及所关联的 Spring Framework 版本

- 场景分析：掌握技术选型
- 系统学习：拒绝浅尝辄止
- 重视规范：了解发展趋势
- 源码解读：理解设计思想
- 实战演练：巩固学习成果





# 主要议题

- 延迟时间 G1 调优
- 吞吐量 G1 调优
- 下期预告



# 延迟时间 G1 调优



# 延迟时间 G1 调优

- 高延迟通用问题
  - 系统时间 (System Time) 占用过高
  - 引用对象处理时间过长
  - Young-Only 收集时间过长
  - Mixed 收集时间过长
  - Remembered Set 更新和扫描时间过高



# 延迟时间 G1 调优

- System-Time 和 Real-Time

每次收集停顿时，gc+cpu=info 日志将会记录操作系统停顿时间，比如 User=0.19s Sys=0.01s Real=0.01s，其中，

User 是用户时间，记录 JVM 代码所执行的时间

Sys 是系统时间，包含操作系统所花费的时间

Real 是停顿的绝对时间消耗



# 延迟时间 G1 调优

- 系统时间（System Time）占用过高通用场景一
  - JVM 从操作系统内存中分配或回收内存可能造成不必要的延迟
  - 解决方案 - 将内存操作提早到启动阶段
    - 将 VM 参数 `-Xms` 与 `-Xmx` 设置成相同数值
    - 并配合参数 `-XX:+AlwaysPreTouch`



# 延迟时间 G1 调优

- 系统时间 (System Time) 占用过高通用场景二
  - 在 Linux 中, Transparent Huge Pages (THP) 特性将小页面合并成大页面, 不仅仅是停顿, 它将随机地拖慢进程, 对于管理大量内存的 JVM, 这将是一个较高的风险, 因此, 关闭 THP 特性, 减少操作系统所带来的时间消耗
  - 解决方案 (CentOS)
    - `echo 'never' > /sys/kernel/mm/transparent_hugepage/enabled`
    - `echo 'never' > /sys/kernel/mm/transparent_hugepage/defrag`



# 延迟时间 G1 调优

- 系统时间（System Time）占用过高通用场景三
  - 不同应用进程向硬盘中写入大量的日志文件，可能造成 I/O 带宽瓶颈，造成一定的延迟
  - 解决方案
    - 设置合理的 Buffer 大小
    - 替换文件系统的日志存储方式



# 延迟时间 G1 调优

- 引用对象处理时间过长
  - 引用处理和入队所消耗的时间将显示 GC 日志信息中
  - 在引用处理阶段，G1 将更新引用对象（Reference）中被引用对象（根据它们特殊类型）
  - 在引用入队阶段，如果发现被引用已消亡，G1 将入队引用对象（Reference）到它们各自的引用队列
  - 如果以上过程处理时间较长，可以考虑增加 VM 参数 `-XX:+ParallelRefProcEnabled` 来处理



# 延迟时间 G1 调优

- Young-Only 收集时间过长
  - 通常，Young 收集所消耗的时间与 Young 生代的空间大小大致上成正比，尤其是，大量的存活对象需要被复制
  - 如果 Evacuate Collection Set 阶段执行时间较长的话，可以考虑减少新生代的 Heap 空间占比，即调低 -XX:G1NewSizePercent（实验性参数）
  - 同样地，调低 -XX:G1MaxNewSizePercent 也可以减缓 Young 收集所带来的时间消耗



# 延迟时间 G1 调优

- Mixed 收集时间过长
  - Mixed 收集在老生代回收空间，Mixed 收集包含新生代和老年代的 Regions，如果收集时间过长，首先调优 Young-Only 收集，再调优：
    - -XX:G1MixedGCCountTarget
    - -XX:G1MixedGCLiveThresholdPercent
    - -XX:G1HeapWastePercent



# 吞吐量 G1 调优



# 下期预告

第七期 *Java* 元编程



谢谢观看