第01章：概述篇

第01章：概述篇 1

1 1-大厂面试题 3

2 2-背景说明 4

2.1 1-生产环境中的问题 4

2.1.1 生产环境发生了内存溢出该如何处理 4

2.1.2 副主题 4

生产环境应该给服务器分配多少内存合适？ 4

如何对垃圾回收器的性能进行调优？ 4

生产环境CPU负载飙高该如何处理？ 4

生产环境应该给应用分配多少线程合适？ 4

不加log，如何确定请求是否执行了某一行代码？ 4

不加log，如何实时查看某个方法的入参与返回值？ 4

2.2 2-为什么要调优 4

2.2.1 防止出现OOM 4

2.2.2 解决OOM 4

2.2.3 减少Full GC出现的频率 4

2.3 3-不同阶段的考虑 4

2.3.1 上线前 5

2.3.2 项目运行阶段 5

2.3.3 线上出现OOM 5

3 3-调优概述 5

3.1 1-监控的依据 5

3.1.1 运行日志 5

3.1.2 异常堆栈 5

3.1.3 GC日志 5

3.1.4 线程快照 5

3.1.5 堆转储快照 5

3.2 2-调优的大方向 5

3.2.1 合理地编写代码 5

3.2.2 充分并合理的使用硬件资源 5

3.2.3 合理地进行JVM调优 5

4 4-性能优化的步骤 5

4.1 第1步（发现问题）：性能监控 6

4.1.1 GC频繁 6

4.1.2 cpu load过高 6

4.1.3 OOM 6

4.1.4 内存泄露 6

4.1.5 死锁 6

4.1.6 程序响应时间较长 6

4.2 第2步（排查问题）：性能分析 6

4.2.1 打印GC日志，通过GCviewer或者 http://gceasy.io来分析异常信息 7

4.2.2 灵活运用 命令行工具、jstack、jmap、jinfo等 7

4.2.3 dump出堆文件，使用内存分析工具分析文件 7

4.2.4 使用阿里Arthas、jconsole、JVisualVM来实时查看JVM状态 7

4.2.5 jstack查看堆栈信息 7

4.3 第3步（解决问题）：性能调优 7

4.3.1 适当增加内存，根据业务背景选择垃圾回收器 8

4.3.2 优化代码，控制内存使用 8

4.3.3 增加机器，分散节点压力 8

4.3.4 合理设置线程池线程数量 8

4.3.5 使用中间件提高程序效率，比如缓存、消息队列等 8

4.3.6 其他…… 8

5 5-性能评价/测试指标 8

5.1 1-停顿时间（或响应时间） 8

5.2 2-吞吐量 9

5.2.1 对单位时间内完成的工作量（请求）的量度 9

5.2.2 在GC中：运行用户代码的事件占总运行时间的比例（总运行时间：程序的运行时间+内存回收的时间） 吞吐量为1-1/(1+n)，其中-XX::GCTimeRatio=n 9

5.3 3-并发数 9

5.3.1 同一时刻，对服务器有实际交互的请求数 9

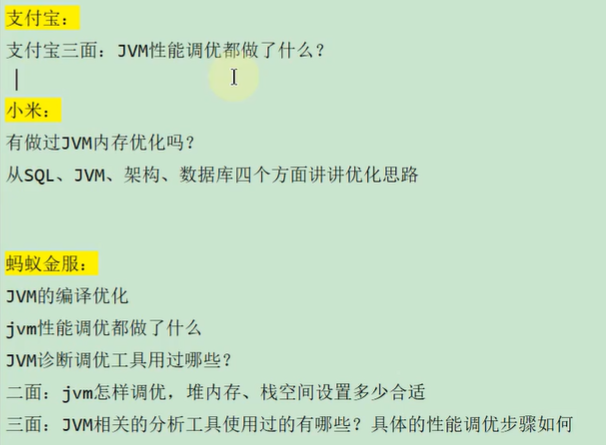
5.4 4-内存占用 9

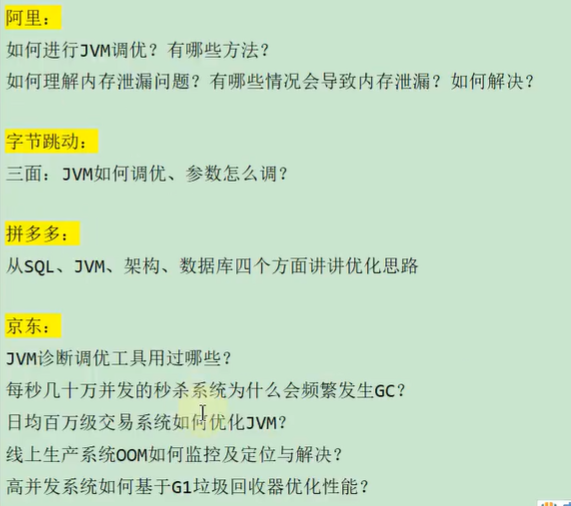
5.4.1 Java堆区所占的内存大小 9

5.5 5-相互间的关系 9

5.5.1 以高速公路通行状况为例 9

1. 1-大厂面试题





1. 2-背景说明
   1. 1-生产环境中的问题
      1. 生产环境发生了内存溢出该如何处理
      2. 副主题

生产环境应该给服务器分配多少内存合适？

如何对垃圾回收器的性能进行调优？

生产环境CPU负载飙高该如何处理？

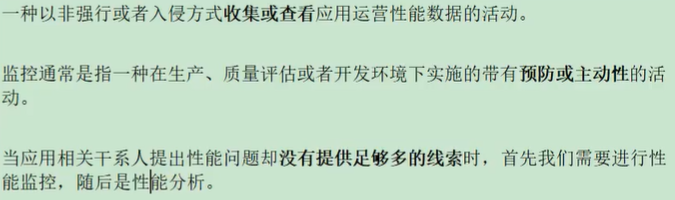
生产环境应该给应用分配多少线程合适？

不加log，如何确定请求是否执行了某一行代码？

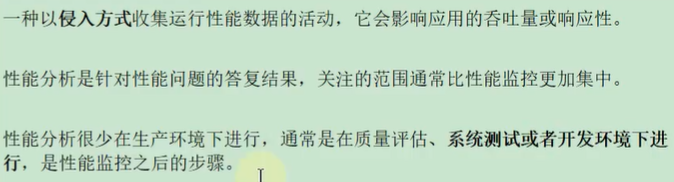
不加log，如何实时查看某个方法的入参与返回值？

* 1. 2-为什么要调优
     1. 防止出现OOM
     2. 解决OOM
     3. 减少Full GC出现的频率
  2. 3-不同阶段的考虑
     1. 上线前
     2. 项目运行阶段
     3. 线上出现OOM

1. 3-调优概述
   1. 1-监控的依据
      1. 运行日志
      2. 异常堆栈
      3. GC日志
      4. 线程快照
      5. 堆转储快照
   2. 2-调优的大方向
      1. 合理地编写代码
      2. 充分并合理的使用硬件资源
      3. 合理地进行JVM调优
2. 4-性能优化的步骤
   1. 第1步（发现问题）：性能监控



* + 1. GC频繁
    2. cpu load过高
    3. OOM
    4. 内存泄露
    5. 死锁
    6. 程序响应时间较长
  1. 第2步（排查问题）：性能分析



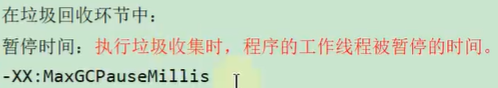
* + 1. 打印GC日志，通过GCviewer或者 http://gceasy.io来分析异常信息
    2. 灵活运用 命令行工具、jstack、jmap、jinfo等
    3. dump出堆文件，使用内存分析工具分析文件
    4. 使用阿里Arthas、jconsole、JVisualVM来实时查看JVM状态
    5. jstack查看堆栈信息
  1. 第3步（解决问题）：性能调优

graphic

* + 1. 适当增加内存，根据业务背景选择垃圾回收器
    2. 优化代码，控制内存使用
    3. 增加机器，分散节点压力
    4. 合理设置线程池线程数量
    5. 使用中间件提高程序效率，比如缓存、消息队列等
    6. 其他……

1. 5-性能评价/测试指标
   1. 1-停顿时间（或响应时间）





* 1. 2-吞吐量
     1. 对单位时间内完成的工作量（请求）的量度
     2. 在GC中：运行用户代码的事件占总运行时间的比例（总运行时间：程序的运行时间+内存回收的时间） 吞吐量为1-1/(1+n)，其中-XX::GCTimeRatio=n
  2. 3-并发数
     1. 同一时刻，对服务器有实际交互的请求数

graphic

* 1. 4-内存占用
     1. Java堆区所占的内存大小
  2. 5-相互间的关系
     1. 以高速公路通行状况为例

