

<<Java程序性能优化>>

图书基本信息

书名：<<Java程序性能优化>>

13位ISBN编号：9787302296256

10位ISBN编号：7302296251

出版时间：2012-9-1

出版时间：清华大学出版社

作者：葛一鸣

页数：400

字数：649000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Java程序性能优化>>

前言

关于JavaJava是目前应用最为广泛的软件开发平台之一。

随着Java以及Java社区的不断壮大，Java也早已不再是一门计算机语言了，它更是一个平台、一种文化、一个社区。

作为一个平台，JVM虚拟机扮演着举足轻重的作用。

除了Java语言，任何一种能够被编译成字节码的计算机语言都属于Java这个平台。

Groovy、Scala、JRuby等都是Java平台的一个部分，它们依赖于JVM虚拟机，同时，Java平台也因为它们变得更加丰富多彩。

作为一种文化，Java几乎成为了“开源”的代名词。

在Java程序中，有着数不清的开源软件和框架，如Tomcat、Struts、Hibernate、Spring等等。

就连JDK和JVM自身也有不少开源的实现如OpenJDK、Harmony。

可以说，“共享”的精神在Java世界里体现得淋漓尽致。

作为一个社区，Java拥有无数的开发人员，数不清的论坛、资料。

从桌面应用软件、嵌入式开发到企业级应用、后台服务器、中间件，都可以看到Java的身影。

其应用形式之复杂、参与人数之庞大也令人咋舌。

可以说，Java社区已经俨然成为了一个良好而庞大的生态系统。

此外，纯粹作为一门软件开发语言，Java非常容易学习，其学习曲线较C++等老牌计算机语言相比，也比较平缓。

因为它尽力简化或去除了C++中许多晦涩、多余和难以理解的部分，如指针、虚函数、多继承等。

本书架构本书主要介绍Java应用程序的优化方法和技巧，总共分为6章。

第1章是综述，介绍了性能的基本概念，两个重要的定律（木桶原理和Amdahl定律），以及系统调优的一般过程与注意事项。

第2章从设计层面，介绍了与性能相关的设计模式、组件以及有助于改善性能的软件设计思想。

第3章从代码层面介绍了如何编写高性能的Java代码。

涉及的主要内容有：字符串的优化处理、文件I/O的优化、核心数据库结构的使用、Java的引用类型以及一些常用的惯例。

第4章介绍了并程序开发的相关内容，以及如何通过多线程提高系统性能。

先后介绍了并发设计模式、线程池、并发数据结构的使用、并发控制方法、“锁”的优化、无锁的使用以及协程。

第5章立足于JVM虚拟机层面，介绍了如何通过设置合理的JVM参数提升Java程序的性能。

第6章为工具篇，主要介绍了获取和监控程序或系统性能指标的各种工具，以及Java应用程序相关的故障排查工具。

本书特点本书的主要特点有：· 专注于介绍Java应用程序的优化方法、技巧和思想，并深度剖析JDK的部分实现。

· 具有较强的层次性和连贯性，依次介绍了在软件设计层面、代码层面、JVM虚拟机层面的优化方法。

· 理论结合实际，使用丰富的示例帮助读者理解理论知识。

阅读人群要通读本书并取得良好的学习效果，要求读者需要具备Java的基本知识。

本书不是一本帮助初学者入门的书籍。

因此，本书适合以下读者：· 拥有一定开发经验的Java开发人员· Java软件设计师、架构师· 系统调优人员· 有一定的Java基础并希望更进一步的程序员

本书的约定本书在叙述过程中，有如下约定：· 本书中所述的JDK 1.5、JDK 1.6等同于JDK 5、JDK 6。

· 如无特殊说明，JVM虚拟机均指Hot Spot虚拟机。

· 如无特殊说明，本书的程序、示例均在JDK 1.6环境中运行。

下载提示本书涉及的源程序请读者直接登录清华大学出版社网站，搜索到本书页面后按照提示进行下载。

<<Java程序性能优化>>

感谢在本书的写作过程中，我充满着感激之情。

首先是对我的家人，在本书完稿前，父亲病重，但我由于工作上的繁忙未能抽出太多时间照顾他，幸好得到了母亲的大力支持和父亲的谅解，我才能够鼓足勇气，全身心投入到写作之中。

同时，母亲对我的悉心照料也让我更加能够专注到工作之中。

同时，我要感谢我的工作单位UT斯达康以及两位前辈Rex Zhu和Tao Tao。

正是他们在平时工作中对我的细心指导，才能让我有所进步和积累。

而这一切，正是本书的基础。

最后，再次感谢我的母亲，祝她身体健康。

葛一鸣

<<Java程序性能优化>>

内容概要

Java是目前应用最为广泛的软件开发平台，学习针对Java程序的优化方法有重要的现实意义。

《Java程序性能优化：让你的Java程序更快、更稳定》以Java性能调优为主线，系统地阐述了与Java性能优化相关的知识与技巧。

《Java程序性能优化：让你的Java程序更快、更稳定》共6章，先后从软件设计、软件编码、JVM调优以及程序故障排斥等方面介绍针对Java程序的优化方法。

第1章介绍性能的基本概念、定律、系统调优的过程和注意事项。

第2章从设计层面介绍与性能相关的设计模式、组件。

第3章从代码层面介绍如何编写高性能的Java程序。

第4章介绍了并行开发和如何通过多线程提高系统性能。

第5章立足于JVM虚拟机层面，介绍如何通过设置合理的JVM参数提升Java程序的性能。

第6章为工具篇，介绍了获取和监控程序或系统性能指标的各种工具，包括相关的故障排查工具。

本书适合所有Java程序员、软件设计师、架构师以及软件开发爱好者，对于有一定经验的Java工程师，本书更能帮助他突破技术瓶颈，深入Java内核开发！

<<Java程序性能优化>>

作者简介

葛一鸣，毕业于浙江工业大学，获得了计算机软件与理论专业硕士学位。

取得了国家认证系统分析师职称、OCP（Oracle数据库认证专家）。

现就职于UT斯达康通讯有限公司，从事Java软件开发。

有多年的软件开发经验，参与开发过多个软件项目。

熟悉Java、PHP、MySQL、Oracle等开发语言和数据库，对Java软件开发、性能优化、分布式软件有着浓厚的兴趣和深入的研究。

在校期间曾经发表过多篇技术论文。

<<Java程序性能优化>>

书籍目录

第1章 Java性能调优概述

1.1 性能概述

1.1.1 看懂程序的性能

1.1.2 性能的参考指标

1.1.3 木桶原理与性能瓶颈

1.1.4 Amdahl定律

1.2 性能调优的层次

1.2.1 设计调优

1.2.2 代码调优

1.2.3 JVM调优

1.2.4 数据库调优

1.2.5 操作系统调优

1.3 基本调优策略和手段

1.3.1 优化的一般步骤

1.3.2 系统优化注意事项

1.4 小结

第2章 设计优化

2.1 善用设计模式

2.1.1 单例模式

2.1.2 代理模式

2.1.3 享元模式

2.1.4 装饰者模式

2.1.5 观察者模式

2.1.6 Value Object模式

2.1.7 业务代理模式

2.2 常用优化组件和方法

2.2.1 缓冲 (Buffer)

2.2.2 缓存 (Cache)

2.2.3 对象复用——“池”

2.2.4 并行替代串行

2.2.5 负载均衡

2.2.6 时间换空间

2.2.7 空间换时间

2.3 小结

第3章 Java程序优化

3.1 字符串优化处理

3.1.1 String对象及其特点

3.1.2 subString()方法的内存泄漏

3.1.3 字符串分割和查找

3.1.4 StringBuffer和StringBuilder

3.2 核心数据结构

3.2.1 List接口

3.2.2 Map接口

3.2.3 Set接口

3.2.4 优化集合访问代码

<<Java程序性能优化>>

- 3.2.5 RandomAccess接口
- 3.3 使用NIO提升性能
 - 3.3.1 NIO的Buffer类族和Channel
 - 3.3.2 Buffer的基本原理
 - 3.3.3 Buffer的相关操作
 - 3.3.4 MappedByteBuffer性能评估
 - 3.3.5 直接内存访问
- 3.4 引用类型
 - 3.4.1 强引用
 - 3.4.2 软引用
 - 3.4.3 弱引用
 - 3.4.4 虚引用
 - 3.4.5 WeakHashMap类及其实现
- 3.5 有助于改善性能的技巧
 - 3.5.1 慎用异常
 - 3.5.2 使用局部变量
 - 3.5.3 位运算代替乘除法
 - 3.5.4 替换switch
 - 3.5.5 一维数组代替二维数组
 - 3.5.6 提取表达式
 - 3.5.7 展开循环
 - 3.5.8 布尔运算代替位运算
 - 3.5.9 使用arrayCopy ()
 - 3.5.10 使用Buffer进行I/O操作
 - 3.5.11 使用clone()代替new
 - 3.5.12 静态方法替代实例方法
- 3.6 小结
- 第4章 并行程序开发及优化
 - 4.1 并行程序设计模式
 - 4.1.1 Future模式
 - 4.1.2 Master-Worker模式
 - 4.1.3 Guarded Suspension模式
 - 4.1.4 不变模式
 - 4.1.5 生产者-消费者模式
 - 4.2 JDK多任务执行框架
 - 4.2.1 无限制线程的缺陷
 - 4.2.2 简单的线程池实现
 - 4.2.3 Executor框架
 - 4.2.4 自定义线程池
 - 4.2.5 优化线程池大小
 - 4.2.6 扩展ThreadPoolExecutor
 - 4.3 JDK并发数据结构
 - 4.3.1 并发List
 - 4.3.2 并发Set
 - 4.3.3 并发Map
 - 4.3.4 并发Queue
 - 4.3.5 并发Deque

<<Java程序性能优化>>

4.4 并发控制方法

4.4.1 Java内存模型与volatile

4.4.2 同步关键字synchronized

4.4.3 ReentrantLock重入锁

4.4.4 ReadWriteLock读写锁

4.4.5 Condition对象

4.4.6 Semaphore信号量

4.4.7 ThreadLocal线程局部变量

4.5 “锁”的性能和优化

4.5.1 线程的开销

4.5.2 避免死锁

4.5.3 减小锁持有时间

4.5.4 减小锁粒度

4.5.5 读写分离锁来替换独占锁

4.5.6 锁分离

4.5.7 重入锁ReentrantLock和内部锁synchronized

4.5.8 锁粗化(Lock Coarsening)

4.5.9 自旋锁 (Spinning Lock)

4.5.10 锁消除 (Lock Elimination)

4.5.11 锁偏向(Biased Lock)

4.6 无锁的并行计算

4.6.1 非阻塞的同步/无锁

4.6.2 原子操作

4.6.3 Amino框架介绍

4.6.4 Amino集合

4.6.5 Amino树

4.6.6 Amino图

4.6.7 Amino简单调度模式

4.7 协程

4.7.1 协程的概念

4.7.2 Kilim框架简介

4.7.3 Task及其状态

4.7.4 Fiber及其状态

4.7.5 Kilim开发环境配置

4.7.6 Kilim之Hello World

4.7.7 多任务通信

4.7.8 Kilim实例及性能评估

4.8 小结

第5章 JVM调优

5.1 Java虚拟机内存模型

5.1.1 程序计数器

5.1.2 Java虚拟机栈

5.1.3 本地方法栈

5.1.4 Java堆

5.1.5 方法区

5.2 JVM内存分配参数

5.2.1 设置最大堆内存

<<Java程序性能优化>>

- 5.2.2 设置最小堆内存
- 5.2.3 设置新生代
- 5.2.4 设置持久代
- 5.2.5 设置线程栈
- 5.2.6 堆的比例分配
- 5.2.7 堆分配参数总结
- 5.3 垃圾收集基础
 - 5.3.1 垃圾收集的作用
 - 5.3.2 垃圾回收算法与思想
 - 5.3.3 垃圾收集器的类型
 - 5.3.4 评价GC策略的指标
 - 5.3.5 新生代串行收集器
 - 5.3.6 老年代串行收集器
 - 5.3.7 并行收集器
 - 5.3.8 新生代并行回收 (Parallel Scavenge) 收集器
 - 5.3.9 老年代并行回收收集器
 - 5.3.10 CMS收集器
 - 5.3.11 G1收集器 (Garbage First)
 - 5.3.12 Stop the World案例
 - 5.3.13 收集器对系统性能的影响
 - 5.3.14 GC相关参数总结
- 5.4 常用调优案例和方法
 - 5.4.1 将新对象预留在新生代
 - 5.4.2 大对象进入老年代
 - 5.4.3 设置对象进入老年代的年龄
 - 5.4.4 稳定与震荡的堆大小
 - 5.4.5 吞吐量优先案例
 - 5.4.6 使用大页案例
 - 5.4.7 降低停顿案例
- 5.5 实用JVM参数
 - 5.5.1 JIT编译参数
 - 5.5.2 堆快照 (堆Dump)
 - 5.5.3 错误处理
 - 5.5.4 取得GC信息
 - 5.5.5 类和对象跟踪
 - 5.5.6 控制GC
 - 5.5.7 选择类校验器
 - 5.5.8 Solaris下线程控制
 - 5.5.9 使用大页
 - 5.5.10 压缩指针
- 5.6 实战JVM调优
 - 5.6.1 Tomcat简介与启动加速
 - 5.6.2 Web应用程序介绍
 - 5.6.3 JMeter介绍与使用
 - 5.6.4 调优前Web应用运行状况
 - 5.6.5 调优过程
- 5.7 总结

<<Java程序性能优化>>

第6章 Java性能调优工具

6.1 Linux命令行工具

6.1.1 top命令

6.1.2 sar命令

6.1.3 vmstat命令

6.1.4 iostat命令

6.1.5 pidstat工具

6.2 Windows工具

6.2.1 任务管理器

6.2.2 perfmon性能监控工具

6.2.3 Process Explorer

6.2.4 pslist命令行

6.3 JDK命令行工具

6.3.1 jps命令

6.3.2 jstat命令

6.3.3 jinfo命令

6.3.4 jmap命令

6.3.5 jhat命令

6.3.6 jstack命令

6.3.7 jstatd命令

6.3.8 hprof工具

6.4 JConsole工具

6.4.1 JConsole连接Java程序

6.4.2 Java程序概况

6.4.3 内存监控

6.4.4 线程监控

6.4.5 类加载情况

6.4.6 虚拟机信息

6.4.7 MBean管理

6.4.8 使用插件

6.5 Visual VM多合一工具

6.5.1 Visual VM连接应用程序

6.5.2 监控应用程序概况

6.5.3 Thread Dump和分析

6.5.4 性能分析

6.5.5 快照

6.5.6 内存快照分析

6.5.7 MBean管理

6.5.8 TDA使用

6.5.9 BTrace介绍

6.6 Visual VM对OQL的支持

6.6.1 Visual VM的OQL基本语法

6.6.2 内置heap对象

6.6.3 对象函数

6.6.4 集合/统计函数

6.6.5 程序化OQL

6.7 MAT内存分析工具

<<Java程序性能优化>>

- 6.7.1 初识MAT
- 6.7.2 浅堆和深堆
- 6.7.3 支配树 (Dominator Tree)
- 6.7.4 垃圾回收根
- 6.7.5 内存泄露检测
- 6.7.6 最大对象报告
- 6.7.7 查找支配者
- 6.7.8 线程分析
- 6.7.9 集合使用情况分析
- 6.7.10 扩展MAT
- 6.8 MAT对OQL的支持
 - 6.8.1 Select子句
 - 6.8.2 From子句
 - 6.8.3 Where子句
 - 6.8.4 内置对象与方法
- 6.9 JProfile简介
 - 6.9.1 JProfile使用配置
 - 6.9.2 内存视图
 - 6.9.3 堆快照
 - 6.9.4 CPU视图
 - 6.9.5 线程视图
 - 6.9.6 JVM统计信息
 - 6.9.7 触发器
- 6.10 小结

<<Java程序性能优化>>

编辑推荐

一个优秀的程序员，不仅要会编写程序，更要会编写高质量的程序感受Java开发中的大智慧，让你的Java程序更优美专注于Java应用程序的优化方法、技巧和思想深入剖析软件设计层面、代码层面、JVM虚拟机层面的优化方法理论结合实际，使用丰富的示例帮助读者理解理论知识

<<Java程序性能优化>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>