

<<深入理解Java虚拟机>>

图书基本信息

书名：<<深入理解Java虚拟机>>

13位ISBN编号：9787111349662

10位ISBN编号：7111349660

出版时间：2011-6

出版时间：机械工业出版社

作者：周志明

页数：387

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>

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

## 前言

Java是目前用户最多、使用范围最广的软件开发技术，Java的技术体系主要由支撑Java程序运行的虚拟机、为各开发领域提供接口支持的Java API、Java编程语言及许许多多的第三方Java框架（如Spring和Struts等）构成。

在国内，有关Java API、Java语言及第三方框架的技术资料和书籍非常丰富，相比之下，有关Java虚拟机的资料却显得异常贫乏。

这种状况很大程度上是由Java开发技术本身的一个重要优点导致的：在虚拟机层面隐藏了底层技术的复杂性以及机器与操作系统的差异性。

运行程序的物理机器情况千差万别，而Java虚拟机则在千差万别的物理机上面建立了统一的运行平台，实现了在任意一台虚拟机上编译的程序都能在任何一台虚拟机上正常运行。

这一极大的优势使得Java应用的开发比传统C/C++应用的开发更高效和快捷，程序员可以把主要精力集中在具体业务逻辑上，而不是物理硬件的兼容性上。

一般情况下，一个程序员只要了解了必要的Java API、Java语法并学习适当的第三方开发框架，就已经基本能满足日常开发的需要了，虚拟机会在用户不知不觉中完成对硬件平台的兼容以及对内存等资源的管理工作。

因此，了解虚拟机的运作并不是一般开发人员必须掌握的知识。

然而，凡事都具备两面性。

随着Java技术的不断发展，它被应用于越来越多的领域之中。

其中一些领域，如电力、金融、通信等，对程序的性能、稳定性和可扩展性方面都有极高的要求。

一个程序很可能在10个人同时使用时完全正常，但是在10000个人同时使用时就会变慢、死锁甚至崩溃。

毫无疑问，要满足10000个人同时使用需要更高性能的物理硬件，但是在绝大多数情况下，提升硬件效能无法等比例地提升程序的性能和并发能力，有时甚至可能对程序的性能没有任何改善作用。

这里面有Java虚拟机的原因：为了达到为所有硬件提供一致的虚拟平台的目的，牺牲了一些硬件相关的性能特性。

更重要的是人为原因：开发人员如果不了解虚拟机的一些技术特性的运行原理，就无法写出最适合虚拟机运行和可自优化的代码。

其实，目前商用的高性能Java虚拟机都提供了相当多的优化特性和调节手段，用于满足应用程序在实际生产环境中对性能和稳定性的要求。

如果只是为了入门学习，让程序在自己的机器上正常运行，那么这些特性可以说是可有可无的；如果用于生产环境，尤其是企业级应用开发中，就迫切需要开发人员中至少有一部分人对虚拟机的特性及调节方法具有很清晰的认识，所以在Java开发体系中，对架构师、系统调优师、高级程序员等角色的需求一直都非常大。

学习虚拟机中各种自动运作的特性的原理也成为了Java程序员成长道路上必然会接触到的一课。

通过本书，读者可以以一种相对轻松的方式学习虚拟机的运作原理，对Java程序员的成长也有较大的帮助。

## <<深入理解Java虚拟机>>

### 内容概要

作为一位Java程序员，你是否也曾经想深入理解Java虚拟机，但是却被它的复杂和深奥拒之门外？没关系，本书极尽化繁为简之妙，能带领你在轻松中领略Java虚拟机的奥秘。

本书是近年来国内出版的唯一一本与Java虚拟机相关的专著，也是唯一一本同时从核心理论和实际运用这两个角度去探讨Java虚拟机的著作，不仅理论分析得透彻，而且书中包含的典型案例和最佳实践也极具现实指导意义。

全书共分为五大部分。

第一部分从宏观的角度介绍了整个Java技术体系的过去、现在和未来，以及如何独立地编译一个OpenJDK7，这对理解后面的内容很有帮助。

第二部分讲解了JVM的自动内存管理，包括虚拟机内存区域的划分原理以及各种内存溢出异常产生的原因；常见的垃圾收集算法以及垃圾收集器的特点和工作原理；常见的虚拟机的监控与调试工具的原理和使用方法。

第三部分分析了虚拟机的执行子系统，包括Class的文件结构以及如何存储和访问Class中的数据；虚拟机的类创建机制以及类加载器的工作原理和它对虚拟机的意义；虚拟机字节码的执行引擎以及它在实行代码时涉及的内存结构。

第四部分讲解了程序的编译与代码的优化，阐述了泛型、自动装箱拆箱、条件编译等语法糖的原理；讲解了虚拟机的热点探测方法、HotSpot的即时编译器、编译触发条件，以及如何从虚拟机外部观察和分析JIT编译的数据和结果。

第五部分探讨了Java实现高效并发的原理，包括JVM内存模型的结构和操作；原子性、可见性和有序性在Java内存模型中的体现；先行发生原则的规则和使用；线程在Java语言中的实现原理；虚拟机实现高效并发所做的一系列锁优化措施。

本书适合所有Java程序员、系统调优师和系统架构师阅读。

## <<深入理解Java虚拟机>>

### 作者简介

周志明，资深Java技术专家，对JavaEE企业级应用开发、OSGi、Java虚拟机和工作流等都有深入的研究，并在大量的实践中积累了丰富的经验。

尤其精通Java虚拟机，撰写了大量与JVM相关的经典文章，被各大技术社区争相转载，是ITeye等技术社区公认的Java虚拟机方面的领袖人物之一。

现任远光软件股份有限公司平台开发部经理兼平台架构师，先后参加与过国家电网、南方电网等多个国家级大型ERP项目的平台架构工作，对软件系统架构也有深刻的认识和体会。

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

## 书籍目录

前言

致谢

第一部分 走近Java

第1章 走近Java

1.1 概述

1.2 Java技术体系

1.3 Java发展史

1.4 展望Java技术的未来

1.4.1 模块化

1.4.2 混合语言

1.4.3 多核并行

1.4.4 进一步丰富语法

1.4.5 64位虚拟机

1.5 实战：自己编译JDK

1.5.1 获取JDK源码

1.5.2 系统需求

1.5.3 构建编译环境

1.5.4 准备依赖项

1.5.5 进行编译

1.6 本章小结

第二部分 自动内存管理机制

第2章 Java内存区域与内存溢出异常

2.1 概述

2.2 运行时数据区域

2.2.1 程序计数器

2.2.2 Java虚拟机栈

2.2.3 本地方法栈

2.2.4 Java堆

2.2.5 方法区

2.2.6 运行时常量池

2.2.7 直接内存

2.3 对象访问

2.4 实战：OutOfMemoryError异常

2.4.1 Java堆溢出

2.4.2 虚拟机栈和本地方法栈溢出

2.4.3 运行时常量池溢出

2.4.4 方法区溢出

2.4.5 本机直接内存溢出

2.5 本章小结

第3章 垃圾收集器与内存分配策略

3.1 概述

3.2 对象已死？

3.2.1 引用计数算法

3.2.2 根搜索算法

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

- 3.2.3 再谈引用
- 3.2.4 生存还是死亡？
- 3.2.5 回收方法区
- 3.3 垃圾收集算法
  - 3.3.1 标记-清除算法
  - 3.3.2 复制算法
  - 3.3.3 标记-整理算法
  - 3.3.4 分代收集算法
- 3.4 垃圾收集器
  - 3.4.1 Serial收集器
  - 3.4.2 ParNew收集器
  - 3.4.3 Parallel Scavenge收集器
  - 3.4.4 Serial Old收集器
  - 3.4.5 Parallel Old收集器
  - 3.4.6 CMS收集器
  - 3.4.7 G1收集器
  - 3.4.8 垃圾收集器参数总结
- 3.5 内存分配与回收策略
  - 3.5.1 对象优先在Eden分配
  - 3.5.2 大对象直接进入老年代
  - 3.5.3 长期存活的对象将进入老年代
  - 3.5.4 动态对象年龄判定
  - 3.5.5 空间分配担保
- 3.6 本章小结
- 第4章 虚拟机性能监控与故障处理工具
  - 4.1 概述
  - 4.2 JDK的命令行工具
    - 4.2.1 jps：虚拟机进程状况工具
    - 4.2.2 jstat：虚拟机统计信息监视工具
    - 4.2.3 jinfo：Java配置信息工具
    - 4.2.4 jmap：Java内存映像工具
    - 4.2.5 jhat：虚拟机堆转储快照分析工具
    - 4.2.6 jstack：Java堆栈跟踪工具
  - 4.3 JDK的可视化工具
    - 4.3.1 JConsole：Java监视与管理控制台
    - 4.3.2 VisualVM：多合一故障处理工具
  - 4.4 本章小结
- 第5章 调优案例分析与实战
  - 5.1 概述
  - 5.2 案例分析
    - 5.2.1 高性能硬件上的程序部署策略
    - 5.2.2 集群间同步导致的内存溢出
    - 5.2.3 堆外内存导致的溢出错误
    - 5.2.4 外部命令导致系统缓慢
    - 5.2.5 服务器JVM进程崩溃
  - 5.3 实战：Eclipse运行速度调优

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

- 5.3.1 调优前的程序运行状态
- 5.3.2 升级JDK 1.6的性能变化及兼容问题
- 5.3.3 编译时间和类加载时间的优化
- 5.3.4 调整内存设置控制垃圾收集频率
- 5.3.5 选择收集器降低延迟
- 5.4 本章小结
- 第三部分 虚拟机执行子系统
- 第6章 类文件结构
  - 6.1 概述
  - 6.2 无关性的基石
  - 6.3 Class类文件的结构
    - 6.3.1 魔数与Class文件的版本
    - 6.3.2 常量池
    - 6.3.3 访问标志
    - 6.3.4 类索引、父类索引与接口索引集合
    - 6.3.5 字段表集合
    - 6.3.6 方法表集合
    - 6.3.7 属性表集合
  - 6.4 Class文件结构的发展
  - 6.5 本章小结
- 第7章 虚拟机类加载机制
  - 7.1 概述
  - 7.2 类加载的时机
  - 7.3 类加载的过程
    - 7.3.1 加载
    - 7.3.2 验证
    - 7.3.3 准备
    - 7.3.4 解析
    - 7.3.5 初始化
  - 7.4 类加载器
    - 7.4.1 类与类加载器
    - 7.4.2 双亲委派模型
    - 7.4.3 破坏双亲委派模型
  - 7.5 本章小结
- 第8章 虚拟机字节码执行引擎
  - 8.1 概述
  - 8.2 运行时栈帧结构
    - 8.2.1 局部变量表
    - 8.2.2 操作数栈
    - 8.2.3 动态连接
    - 8.2.4 方法返回地址
    - 8.2.5 附加信息
  - 8.3 方法调用
    - 8.3.1 解析
    - 8.3.2 分派
  - 8.4 基于栈的字节码解释执行引擎
    - 8.4.1 解释执行

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

- 8.4.2 基于栈的指令集与基于寄存器的指令集
- 8.4.3 基于栈的解释器执行过程
- 8.5 本章小结
- 第9章 类加载及执行子系统的案例与实战
  - 9.1 概述
  - 9.2 案例分析
    - 9.2.1 Tomcat：正統的类加载器架构
    - 9.2.2 OSGi：灵活的类加载器架构
    - 9.2.3 字节码生成技术与动态代理的实现
    - 9.2.4 Retrotranslator：跨越JDK版本
  - 9.3 实战：自己动手实现远程执行功能
    - 9.3.1 目标
    - 9.3.2 思路
    - 9.3.3 实现
    - 9.3.4 验证
  - 9.4 本章小结
- 第四部分 程序编译与代码优化
  - 第10章 早期（编译期）优化
    - 10.1 概述
    - 10.2 Javac编译器
      - 10.2.1 Javac的源码与调试
      - 10.2.2 解析与填充符号表
      - 10.2.3 注解处理器
      - 10.2.4 语义分析与字节码生成
    - 10.3 Java语法糖的味道
      - 10.3.1 泛型与类型擦除
      - 10.3.2 自动装箱、拆箱与遍历循环
      - 10.3.3 条件编译
    - 10.4 实战：插入式注解处理器
      - 10.4.1 实战目标
      - 10.4.2 代码实现
      - 10.4.3 运行与测试
      - 10.4.4 其他应用案例
    - 10.5 本章小结
  - 第11章 晚期（运行期）优化
    - 11.1 概述
    - 11.2 HotSpot虚拟机内的即时编译器
      - 11.2.1 解释器与编译器
      - 11.2.2 编译对象与触发条件
      - 11.2.3 编译过程
      - 11.2.4 查看与分析即时编译结果
    - 11.3 编译优化技术
      - 11.3.1 优化技术概览
      - 11.3.2 公共子表达式消除
      - 11.3.3 数组边界检查消除
      - 11.3.4 方法内联
      - 11.3.5 逃逸分析



## &lt;&lt;深入理解Java虚拟机&gt;&gt;

- 11.4 Java与C/C++的编译器对比
- 11.5 本章小结
- 第五部分 高效并发
- 第12章 Java内存模型与线程
  - 12.1 概述
  - 12.2 硬件的效率与一致性
  - 12.3 Java内存模型
    - 12.3.1 主内存与工作内存
    - 12.3.2 内存间交互操作
    - 12.3.3 对于volatile型变量的特殊规则
    - 12.3.4 对于long和double型变量的特殊规则
    - 12.3.5 原子性、可见性与有序性
    - 12.3.6 先行发生原则
  - 12.4 Java与线程
    - 12.4.1 线程的实现
    - 12.4.2 Java线程调度
    - 12.4.3 状态转换
  - 12.5 本章小结
- 第13章 线程安全与锁优化
  - 13.1 概述
  - 13.2 线程安全
    - 13.2.1 Java语言中的线程安全
    - 13.2.2 线程安全的实现方法
  - 13.3 锁优化
    - 13.3.1 自旋锁与自适应自旋
    - 13.3.2 锁消除
    - 13.3.3 锁粗化
    - 13.3.4 轻量级锁
    - 13.3.5 偏向锁
  - 13.4 本章小结
- 附录A Java虚拟机家族
- 附录B 虚拟机字节码指令表
- 附录C HotSpot虚拟机主要参数表
- 附录D 对象查询语言(OQL)简介
- 附录E JDK历史版本轨迹

## <<深入理解Java虚拟机>>

### 媒体关注与评论

Java 程序是如何运行的？

Java虚拟机在其中扮演了怎样的角色？

如何让Java程序具有更高的并发性？

许多Java程序员都会诸如此类的疑问。

无奈，国内在很长一段时间里都没有一本从实际应用的角度讲解Java虚拟机的著作，《深入理解Java虚拟机：JVM高级特性与最佳实践》的出版可谓填补了这个空白。

它从Java程序员的角度出发，系统地将Java程序运行过程中涉及的各种知识整合到了一起，并配以日常工作中可能会碰到的疑难案例，引领读者轻松踏上探索Java虚拟机的旅途，是广大对Java虚拟机感兴趣的读者的福音！

——莫枢（RednaxelaFX）虚拟机和编程语言爱好者 在武侠的世界里，无论是至刚至强的《易筋经》，还是阴柔无比的《葵花宝典》，都离不开内功修炼。

没有了内功心法，这些武术只是花拳绣腿的拙劣表演而已。

软件业是武林江湖的一个翻版，也有着大量的模式、套路、规范等外功，但“外功修行，内功修神”，要想成为“扫地僧”一样的绝世高人，此书是必备的。

——秦小波 资深Java技术专家/著有畅销书《设计模式之禅》 对Java程序员来说，Java虚拟机可以说是既熟悉又神秘，很少Java程序员能够抑制自己探究它的冲动。

可惜，分析JVM实现原理的书籍（特别是国内作者出版的）是少之又少。

《深入理解Java虚拟机：JVM高级特性与最佳实践》的出版可谓Java程序员的福音，作者将自己多年来在Java虚拟机领域的实践经验和研究心得呈现在了这《深入理解Java虚拟机：JVM高级特性与最佳实践》中，不仅系统地讲解了Java虚拟机工作机制和底层原理，而且更难能可贵的是与实践很好地结合了起来，具有非常强的实践指导意义，强烈推荐！

——计文柯 资深Java技术专家/著有畅销书《Spring技术内幕：深入解析Spring架构设计与实现原理》

## <<深入理解Java虚拟机>>

### 编辑推荐

围绕内存管理、执行子系统、编程编译与优化、高效并发等核心内容对JVM进行全面而深入的分析，深刻揭示JVM的工作原理；注重实现，以解决实践中的疑难问题为首要目的，包含大量经典案例和最佳实践。

<<深入理解Java虚拟机>>

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>