围绕内存管理、执行子系统、程序编译与优化、高效并发等核心内容对JVM进行了全面而深入的分析，深刻揭示JVM的工作原理

注重实战，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

JVM高级特性与最佳实践

Understanding the JVM

Advanced Features and Best Practices

周志明 著

机械工业出版社
China Machine Press

华章原创精品

深入理解Java虚拟机

JVM高级特性与最佳实践

Understanding the JVM

Advanced Features and Best Practices

周志明 著

机械工业出版社
China Machine Press

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。

图书在版编目（CIP）数据

深入理解 Java 虚拟机：JVM 高级特性与最佳实践 / 周志明著 . —北京：机械工业出版社，2011.6

ISBN 978-7-111-34966-2

I. 深…　II. 周…　III. JAVA 语言 – 程序设计　IV. TP312

中国版本图书馆 CIP 数据核字（2011）第 103021 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）
责任编辑：陈佳媛
北京京北印刷有限公司印刷
2011 年 9 月第 1 版第 3 次印刷
186mm×240mm • 25.5 印张
标准书号：ISBN 978-7-111-34966-2
定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

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。

本书读者对象

（1）使用 Java 技术体系的中、高级开发人员

Java 虚拟机作为中、高级开发人员必须修炼的知识，有着较高的学习门槛，本书可作为学习虚拟机的优秀教材。

（2）系统调优师

系统调优师是近几年才兴起的职业，本书中的大量案例、代码和调优实战将会对系统调优师的日常工作有直接的帮助。

（3）系统架构师

保障系统高效、稳定和可伸缩是系统架构师的主要职责之一，而这与虚拟机的运作密不可分，本书可以作为他们设计应用系统底层框架的参考资料。

如何阅读本书

本书一共分为五个部分：走近 Java、自动内存管理机制、虚拟机执行子系统、程序编译与代码优化、高效并发。各个部分基本上是相互独立的，没有必然的前后依赖关

系，读者可以从任何一个感兴趣的专题开始阅读，但是每个部分中的各个章节间有先后顺序。

本书并不假设读者在 Java 领域具备很专业的技术水平，因此在保证逻辑准确的前提下，尽量用通俗的语言和案例讲述虚拟机中与开发关系最为密切的内容。当然学习虚拟机技术本身就需要读者有一定的技术基础，且本书的读者定位是中、高级程序员，因此本书假设读者自己了解一些常用的开发框架、Java API 和 Java 语法等基础知识。

语言约定

本书在语言和技术上有如下的约定：

- 本书中提到 HotSpot 虚拟机、JRockit 虚拟机、WebLogic 服务器等产品的所有者时，仍然使用 Sun 和 BEA 公司的名称。实际上 BEA 和 Sun 分别于 2008 年和 2010 年被 Oracle 公司收购，现在已经不存在这两个商标了，但是毫无疑问它们都是对 Java 领域做出过卓越贡献的、值得程序员纪念的公司。
- JDK 从 1.5 版本开始，在官方的正式文档与宣传资料中已经不再使用类似“JDK 1.5”的名称，只有在程序员内部使用的开发版本号（Developer Version，例如 java -version 的输出）中才继续沿用 1.5、1.6 和 1.7 的版本号，而公开版本号（Product Version）则改为 JDK 5、JDK 6 和 JDK 7 的命名方式。为了行文一致，本书所有场合统一采用开发版本号的命名方式。
- 由于版面关系，本书中的许多示例代码都没有遵循最优的代码编写风格，如使用的流没有关闭流等，请读者在阅读时注意这一点。
- 如果没有特殊说明，本书中所有的讨论都是以 Sun JDK 1.6 为技术平台的。不过如果有某个特性在各个版本间的变化较大，一般都会说明它在各个版本间的差异。

内容特色

第一部分　走近 Java

本书的第一部分为后文的讲解建立了良好的基础。尽管了解 Java 技术的来龙去脉，以及编译自己的 OpenJDK 对于读者理解 Java 虚拟机并不是必需的，但是这些准备过程

可以为走近 Java 技术和 Java 虚拟机提供很好的引导。第一部分只有第 1 章：

第 1 章　介绍了 Java 技术体系的过去、现在和未来的发展趋势，并介绍了如何独立编译一个 OpenJDK 7。

第二部分　自动内存管理机制

因为程序员把内存控制的权力交给了 Java 虚拟机，所以可以在编码的时候享受自动内存管理的诸多优势，不过也正因为这个原因，一旦出现内存泄漏和溢出方面的问题，如果不了解虚拟机是怎样使用内存的，那么排查错误将会成为一项异常艰难的工作。第二部分包括第 2 ～ 5 章：

第 2 章　讲解了虚拟机中的内存是如何划分的，哪部分区域、什么样的代码和操作可能导致内存溢出异常，并讲解了各个区域出现内存溢出异常的常见原因。

第 3 章　分析了垃圾收集的算法和 JDK 1.6 中提供的几款垃圾收集器的特点及运作原理，通过代码实例验证了 Java 虚拟机中的自动内存分配及回收的主要规则。

第 4 章　介绍了随 JDK 发布的 6 个命令行工具与 2 个可视化的故障处理工具的使用方法。

第 5 章　与读者分享了几个比较有代表性的实际案例，还准备了一个所有开发人员都能“亲身实战”的练习，读者可通过实践来获得故障处理和调优的经验。

第三部分　虚拟机执行子系统

执行子系统是虚拟机中必不可少的组成部分，了解了虚拟机如何执行程序，才能写出更优秀的代码。第三部分包括第 6 ～ 9 章：

第 6 章　讲解了 Class 文件结构中的各个组成部分，以及每个部分的定义、数据结构和使用方法，以实战的方式演示了 Class 的数据是如何存储和访问的。

第 7 章　介绍了在类加载过程的“加载”、“验证”、“准备”、“解析”和“初始化”这五个阶段中虚拟机分别执行了哪些动作，还介绍了类加载器的工作原理及其对虚拟机的意义。

第 8 章　分析了虚拟机在执行代码时如何找到正确的方法，如何执行方法内的字节码，以及执行代码时涉及的内存结构。

第 9 章　通过四个类加载及执行子系统的案例，分享了使用类加载器和处理字节码的一些值得欣赏和借鉴的思路，并通过一个实战练习来加深对前面理论知识的理解。

第四部分　程序编译与代码优化

Java 程序从源码编译成字节码和从字节码编译成本地机器码的这两个过程，合并起来其实就等同于一个传统编译器所执行的编译过程。第四部分包括第 10 和 11 章：

第 10 章　分析了 Java 语言中的泛型、自动装箱拆箱、条件编译等多种语法糖的前因后果，并通过实战案例演示了如何使用插入式注解处理器来实现一个检查程序命名规范的编译器插件。

第 11 章　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。此外，还讲解了几种常见的编译期优化技术。

第五部分　高效并发

Java 语言和虚拟机提供了原生的、完善的多线程支持，使得它天生就适合开发多线程并发的应用程序。不过我们不能期望系统来完成所有与并发相关的处理，了解并发的内幕也是一个高级程序员不可缺少的课程。第五部分包括第 12 和 13 章：

第 12 章　讲解了虚拟机的 Java 内存模型的结构和操作，以及原子性、可见性和有序性在 Java 内存模型中的体现，介绍了先行发生原则及使用，还讲解了线程在 Java 语言中是如何实现的。

第 13 章　介绍了线程安全所涉及的概念和分类、同步实现的方式以及虚拟机的底层运作原理，并且还介绍了虚拟机实现高效并发所采取的一系列锁优化措施。

参考资料

本书名为“深入理解 Java 虚拟机”，但要想真的深入理解虚拟机，仅凭一本书肯定是远远不够的，读者可以通过下面的信息找到更多关于 Java 虚拟机方面的资料。我在写作此书的时候，也从下面这些参考资料中获得了很大的帮助。

（1）书籍

❑《The Java Virtual Machine Specification, Second Edition》[①]

《Java 虚拟机规范（第 2 版）》，1999 年 4 月出版。国内并没有引进这本书，

① 官方地址：*http://java.sun.com/docs/books/jvms/second_edition/html/VMSpecTOC.doc.html*。

自然也就没有中文译本，但全书的电子版是免费发布的，在IT书籍中它已经非常“高寿”了[①]（这本书的第3版已处于基本完成的草稿状态，在JDK 1.7正式版发布后这本书应该就会推出第3版）。要学习虚拟机，虚拟机规范无论如何都是必须读的。这本书的概念和细节描述与Sun的早期虚拟机（Sun Classic VM）高度吻合，不过，随着技术的发展，高性能虚拟机真正的细节实现方式与虚拟机规范所描述的差距已经越来越大。但是，如果只能选择一本参考书来了解虚拟机的话，那仍然是这本书。

- 《The Java Language Specification, Third Edition》[②]

  《Java语言规范（第3版）》，2005年7月由机械工业出版社出版，不过出版的是影印版，没有中文译本。虽然Java虚拟机并不是Java语言专有的，但是了解Java语言的各种细节规定对虚拟机的行为也是很有帮助的，它与《Java虚拟机规范（第2版）》都是Sun官方出品的书籍，而且这本书还是由Java之父James Gosling亲自撰写的。

- 《Oracle JRockit The Definitive Guide》

  《Oracle JRockit权威指南》，2010年7月出版，国内也没有（可能是尚未）引进这本书，它是由JRockit的两位资深开发人员（其中一位是JRockit Mission Control团队的TeamLeader）撰写的高级JRockit虚拟机使用指南。虽然JRockit的用户量可能不如HotSpot多，但也是最流行的三大商业虚拟机之一，并且不同虚拟机中的很多实现思路都是可以对比参照的。这本书是了解现代高性能虚拟机的很好的途径。

- 《Inside the Java 2 Virtual Machine, Second Edition》

  《深入Java虚拟机（第2版）》，2000年1月出版，2003年机械工业出版社出版了中文译本。在相当长的时间里，这本书是唯一一本关于Java虚拟机的中文图书。

---

① 在这十多年间虚拟机规范的更新可以通过JSR-924规范来跟踪，JSR-924规范的地址为：*http://jcp.org/aboutJava/communityprocess/maintenance/jsr924/index3.html*。

② 官方地址：*http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf*。

（2）网站资源

- 高级语言虚拟机圈子：*http://hllvm.group.iteye.com/*

  里面有一些国内关于虚拟机的讨论，并不只限于JVM，而是涉及对所有的高级语言虚拟机（High-Level Language Virtual Machine）的讨论，但该网站建立在ITEye[①]上，自然还是以讨论Java虚拟机为主。圈主撒迦（莫枢）的博客（*http://rednaxelafx.iteye.com/*）是另外一个非常有价值的虚拟机及编译原理等资料的分享园地。

- HotSpot Internals：*http://wikis.sun.com/display/HotSpotInternals/Home*

  一个关于OpenJDK的Wiki网站，许多文章都由JDK的开发团队编写，更新很慢，但是仍然有很大的参考价值。

- The HotSpot Group：*http://openjdk.java.net/groups/hotspot/*

  HotSpot组群，包含虚拟机开发、编译器、垃圾收集和运行时四个邮件组，其中有关于HotSpot虚拟机的最新讨论。

联系作者

在本书完稿时，我并没有像想象中那样兴奋或放松，写作时的那种“战战兢兢、如履薄冰”的感觉依然萦绕在心头。在每一章、每一节落笔之时，我都在考虑如何才能把各个知识点更有条理地讲述出来，都在担心会不会由于自己理解有偏差而误导了大家。囿于我的写作水平和写作时间，书中难免存在不妥之处，所以特地开通了一个读者邮箱（understandingjvm@gmail.com）与大家交流，大家如有任何意见或建议都欢迎与我联系。此外，大家也可以通过我的微博（http://t.sina.com.cn//icyfenix）与我取得联系。

勘误

写书和写代码一样，刚开始都是不完美的，需要不断地修正和重构，本书也不例外。如果大家在阅读本书的过程中发现了本书中存在的任何问题，都欢迎反馈给我们。我们会把本书的勘误集中公布在icyfenx.iteye.com/blog/1119214。

---

① 该网站原名为JavaEye，近期更名为ITEye。

致　谢

首先要感谢我的家人，全靠家人在本书写作期间对我的悉心照顾，才让我能够全身心地投入到写作之中，而无后顾之忧。

同时要感谢我的工作单位远光软件，公司为我提供了宝贵的工作、学习和实践的环境，本书中的许多知识点都来自于工作中的实践；也感谢与我一起工作的同事们，非常荣幸能与你们一起在这个富有激情的团队中共同奋斗。

还要感谢淘宝网的莫枢（rednaxelafx.iteye.com）在百忙之中抽空审阅了本书，提出了许多宝贵的建议和意见。

最后，感谢机械工业出版社华章公司的编辑们，本书能够顺利出版，离不开他们的敬业精神和一丝不苟的工作态度。

周志明

2011 年 5 月

目　录

前　言
致　谢

第一部分　走近 Java

第 1 章　走近 Java / 2

1.1　概述 / 2
1.2　Java 技术体系 / 3
1.3　Java 发展史 / 5
1.4　展望 Java 技术的未来 / 9
　1.4.1　模块化 / 9
　1.4.2　混合语言 / 9
　1.4.3　多核并行 / 11
　1.4.4　进一步丰富语法 / 12
　1.4.5　64 位虚拟机 / 13
1.5　实战：自己编译 JDK / 13
　1.5.1　获取 JDK 源码 / 13
　1.5.2　系统需求 / 14

1.5.3 构建编译环境 / 15
1.5.4 准备依赖项 / 17
1.5.5 进行编译 / 18
1.6 本章小结 / 21

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 / 24

2.1 概述 / 24
2.2 运行时数据区域 / 25
2.2.1 程序计数器 / 25
2.2.2 Java 虚拟机栈 / 26
2.2.3 本地方法栈 / 27
2.2.4 Java 堆 / 27
2.2.5 方法区 / 28
2.2.6 运行时常量池 / 29
2.2.7 直接内存 / 29
2.3 对象访问 / 30
2.4 实战：OutOfMemoryError 异常 / 32
2.4.1 Java 堆溢出 / 32
2.4.2 虚拟机栈和本地方法栈溢出 / 35
2.4.3 运行时常量池溢出 / 38
2.4.4 方法区溢出 / 39
2.4.5 本机直接内存溢出 / 41
2.5 本章小结 / 42

第 3 章 垃圾收集器与内存分配策略 / 43

3.1 概述 / 43
3.2 对象已死？ / 44
3.2.1 引用计数算法 / 44

3.2.2 根搜索算法 / 46
3.2.3 再谈引用 / 47
3.2.4 生存还是死亡？ / 48
3.2.5 回收方法区 / 50
3.3 垃圾收集算法 / 51
3.3.1 标记－清除算法 / 52
3.3.2 复制算法 / 52
3.3.3 标记－整理算法 / 54
3.3.4 分代收集算法 / 54
3.4 垃圾收集器 / 55
3.4.1 Serial 收集器 / 56
3.4.2 ParNew 收集器 / 57
3.4.3 Parallel Scavenge 收集器 / 59
3.4.4 Serial Old 收集器 / 60
3.4.5 Parallel Old 收集器 / 61
3.4.6 CMS 收集器 / 61
3.4.7 G1 收集器 / 64
3.4.8 垃圾收集器参数总结 / 64
3.5 内存分配与回收策略 / 65
3.5.1 对象优先在 Eden 分配 / 66
3.5.2 大对象直接进入老年代 / 68
3.5.3 长期存活的对象将进入老年代 / 69
3.5.4 动态对象年龄判定 / 71
3.5.5 空间分配担保 / 73
3.6 本章小结 / 75

第 4 章 虚拟机性能监控与故障处理工具 / 76

4.1 概述 / 76

4.2 JDK 的命令行工具 / 76
4.2.1 jps：虚拟机进程状况工具 / 79
4.2.2 jstat：虚拟机统计信息监视工具 / 80
4.2.3 jinfo：Java 配置信息工具 / 82
4.2.4 jmap：Java 内存映像工具 / 82
4.2.5 jhat：虚拟机堆转储快照分析工具 / 84
4.2.6 jstack：Java 堆栈跟踪工具 / 85
4.3 JDK 的可视化工具 / 87
4.3.1 JConsole：Java 监视与管理控制台 / 88
4.3.2 VisualVM：多合一故障处理工具 / 96
4.4 本章小结 / 105

第 5 章 调优案例分析与实战 / 106

5.1 概述 / 106
5.2 案例分析 / 106
5.2.1 高性能硬件上的程序部署策略 / 106
5.2.2 集群间同步导致的内存溢出 / 109
5.2.3 堆外内存导致的溢出错误 / 110
5.2.4 外部命令导致系统缓慢 / 112
5.2.5 服务器 JVM 进程崩溃 / 113
5.3 实战：Eclipse 运行速度调优 / 114
5.3.1 调优前的程序运行状态 / 114
5.3.2 升级 JDK 1.6 的性能变化及兼容问题 / 117
5.3.3 编译时间和类加载时间的优化 / 122
5.3.4 调整内存设置控制垃圾收集频率 / 126
5.3.5 选择收集器降低延迟 / 130
5.4 本章小结 / 133

第三部分　虚拟机执行子系统

第 6 章　类文件结构 / 136

6.1　概述 / 136

6.2　无关性的基石 / 136

6.3　Class 类文件的结构 / 138

6.3.1　魔数与 Class 文件的版本 / 139

6.3.2　常量池 / 141

6.3.3　访问标志 / 147

6.3.4　类索引、父类索引与接口索引集合 / 148

6.3.5　字段表集合 / 149

6.3.6　方法表集合 / 153

6.3.7　属性表集合 / 155

6.4　Class 文件结构的发展 / 168

6.5　本章小结 / 170

第 7 章　虚拟机类加载机制 / 171

7.1　概述 / 171

7.2　类加载的时机 / 172

7.3　类加载的过程 / 176

7.3.1　加载 / 176

7.3.2　验证 / 178

7.3.3　准备 / 181

7.3.4　解析 / 182

7.3.5　初始化 / 186

7.4　类加载器 / 189

7.4.1　类与类加载器 / 189

7.4.2　双亲委派模型 / 191

7.4.3　破坏双亲委派模型 / 194

7.5 本章小结 / 197

第 8 章 虚拟机字节码执行引擎 / 198

8.1 概述 / 198

8.2 运行时栈帧结构 / 199

8.2.1 局部变量表 / 199

8.2.2 操作数栈 / 204

8.2.3 动态连接 / 206

8.2.4 方法返回地址 / 206

8.2.5 附加信息 / 207

8.3 方法调用 / 207

8.3.1 解析 / 207

8.3.2 分派 / 209

8.4 基于栈的字节码解释执行引擎 / 221

8.4.1 解释执行 / 221

8.4.2 基于栈的指令集与基于寄存器的指令集 / 223

8.4.3 基于栈的解释器执行过程 / 224

8.5 本章小结 / 230

第 9 章 类加载及执行子系统的案例与实战 / 231

9.1 概述 / 231

9.2 案例分析 / 231

9.2.1 Tomcat：正统的类加载器架构 / 232

9.2.2 OSGi：灵活的类加载器架构 / 235

9.2.3 字节码生成技术与动态代理的实现 / 238

9.2.4 Retrotranslator：跨越 JDK 版本 / 242

9.3 实战：自己动手实现远程执行功能 / 246

9.3.1 目标 / 246

9.3.2 思路 / 247