

## <<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：9787111421900

10位ISBN编号：7111421906

出版时间：2013-6

出版时间：机械工业出版社

作者：周志明

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

前言Java是目前用户最多、使用范围最广的软件开发技术之一。

Java的技术体系主要由支撑Java程序运行的虚拟机、提供各开发领域接口支持的Java API、Java编程语言及许多第三方Java框架（如Spring、Struts等）构成。

在国内，有关Java API、Java语言语法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。

这种状况在很大程度上是由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的情况千差万别，而Java虚拟机则在千差万别的物理机上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。

在一般情况下，一个程序员只要了解了必要的Java API、Java语法，以及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。

程序很可能在10个人同时使用时完全正常，但是在10?000个人同时使用时就会缓慢、死锁，甚至崩溃。

毫无疑问，要满足10?000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运作性能和并发能力，甚至可能对程序运作状况完全没有任何改善。

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

更重要的是人为原因：如果开发人员不了解虚拟机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；如果用于生产开发，尤其是企业级生产开发，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。

本书可以使读者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。

第2版与第1版的区别JDK 1.7在2011年7月28日正式发布，相对于2006年发布的JDK 1.6，新版的JDK有了许多新的特性和改进。

本书的第2版也相应地进行了修改和升级，把讲解的技术平台从JDK 1.6提升至JDK 1.7。

例如，增加了对JDK 1.7中最新的G1收集器，以及JDK 1.7中JSR-292 InvokeDynamic（对非Java语言的调用支持）的分析讲解等内容。

在第1版出版后，笔者收到了许多热心读者的反馈意见，部分读者提出OpenJDK开源已久，第1版却很少有直接分析OpenJDK源码的内容，有点“视宝山而不见”的感觉。

因此，在本书第2版中，笔者特别加强了对这部分内容的讲解，其中在第1章中就介绍了如何分析、调试OpenJDK源码等。

在本书后续章节中，不少关于功能点的讲解都直接使用OpenJDK中的HotSpot源码或者JIT编译器生成

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

的本地代码作为论据。

如何把Java虚拟机原理中许多理论性很强的知识、特性应用于实践开发，是本书贯穿始终的主旨。由于笔者希望在本书第2版中进一步加强知识的实践性，因此增加了许多对处理JVM常见问题技能的讲解，包括如何分析GC日志、如何分析JIT编译器代码优化过程和生成代码等。

并且，在第1版的基础上，第2版中进一步增加了若干处理JVM问题的实践案例供读者参考。

另外，本书第2版还修正了第1版中多处错误的、有歧义的和不完整的描述。

有关勘误信息，可以参考第1版的勘误页面。

本书面向的读者（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的性能、并发和伸缩等能力是系统架构师的主要职责之一，而这部分与虚拟机的运作密不可分，本书可以作为他们制定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。

各部分基本上是互相独立的，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。

本书并没有假设读者在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的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。

笔者希望读者在阅读本书的同时，把本书中的实践内容亲自验证一遍，其中用到的代码清单可以从华章网站下载。

语言约定本书在语言和技术上有如下约定：本书中提到HotSpot、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2009年被Oracle公司收购，现在已经不存在这两个商标了，但毫无疑问的是，它们都是在Java领域中做出过卓越贡献的、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，只有程序员内部使用的开发版本号（Developer Version，例如java-version的输出）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，本书中所有讨论都是以Sun JDK 1.7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、现在和未来的一些发展趋势，并介绍了如何独立地编译一个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，不过也正是这个原因，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章讲解了虚拟机中内存是如何划分的，以及哪部分区域、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 1.7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虚拟机中自动内存分配及回收的主要规则。

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

第4章 介绍了随JDK发布的6个命令行工具与两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文件的数据是如何存储和访问的。

第7章 介绍了类加载过程的“加载”、“验证”、“准备”、“解析”和“初始化”5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4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10~11章：第10章 分析了Java语言中泛型、主动装箱和拆箱、条件编译等多种语法糖的前因后果，并通过实战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，这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并发相关的处理，了解并发的内幕也是成为一个高级程序员不可缺少的课程。

第五部分包括第12~13章：第12章 讲解了虚拟机Java内存模型的结构及操作，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的规则及使用，还了解了线程在Java语言中是如何实现的。

第13章 介绍了线程安全涉及的概念和分类、同步实现的方式及虚拟机的底层运作原理，并且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，读者可以通过以下信息找到更多关于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。

……勘误和支持在本书交稿的时候，我并不像想象中的那样兴奋或放松，写作之时那种“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，同时也在担心会不会由于自己理解有偏差而误导了读者。

由于写作水平和写作时间所限，书中难免存在不妥之处，大家如有任何意见或建议欢迎与我联系。

相信写书与写程序一样，作品一定都是不完美的，因为不完美，我们才有不断追求完美的动力。

本书第2版的勘误，将会在作者的博客中发布。

欢迎读者在博客上留言。

致谢首先要感谢我的家人，在本书写作期间全靠他们对我的悉心照顾，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，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Oracle公司虚拟机团队的莫枢，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。

。

<<深入理解Java虚拟机>>

最后，感谢机械工业出版社华章公司的编辑，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

## <<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。

除本书外，还著有经典著作《深入理解OSGi：Equinox原理、应用与最佳实践》，广获读者好评。

现任远光软件股份有限公司开发部总经理兼架构师，先后参与过国家电网、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 第一部分 走近Java 第1章 走近Java 21.1 概述 21.2 Java技术体系 31.3 Java发展史 51.4 Java虚拟机发展史 91.4.1 Sun Classic Exact VM 91.4.2 Sun HotSpot VM 111.4.3 Sun Mobile-Embedded VM Meta-Circular VM 121.4.4 BEA JRockit IBM J9 VM 131.4.5 Azul VM BEA Liquid VM 141.4.6 Apache Harmony Google Android Dalvik VM 141.4.7 Microsoft JVM及其他 151.5 展望Java技术的未来 161.5.1 模块化 171.5.2 混合语言 171.5.3 多核并行 191.5.4 进一步丰富语法 201.5.5 64位虚拟机 211.6 实战：自己编译JDK 221.6.1 获取JDK源码 221.6.2 系统需求 241.6.3 构建编译环境 251.6.4 进行编译 261.6.5 在IDE工具中进行源码调试 311.7 本章小结 35 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 382.1 概述 382.2 运行时数据区域 382.2.1 程序计数器 392.2.2 Java虚拟机栈 392.2.3 本地方法栈 402.2.4 Java堆 412.2.5 方法区 412.2.6 运行时常量池 422.2.7 直接内存 432.3 HotSpot虚拟机对象探秘 432.3.1 对象的创建 442.3.2 对象的内存布局 472.3.3 对象的访问定位 482.4 实战：OutOfMemoryError异常 502.4.1 Java堆溢出 512.4.2 虚拟机栈和本地方法栈溢出 532.4.3 方法区和运行时常量池溢出 562.4.4 本机直接内存溢出 592.5 本章小结 60 第3章 垃圾收集器与内存分配策略 613.1 概述 613.2 对象已死吗 623.2.1 引用计数算法 623.2.2 可达性分析算法 643.2.3 再谈引用 653.2.4 生存还是死亡 663.2.5 回收方法区 683.3 垃圾收集算法 693.3.1 标记-清除算法 693.3.2 复制算法 703.3.3 标记-整理算法 713.3.4 分代收集算法 723.4 HotSpot的算法实现 723.4.1 枚举根节点 723.4.2 安全点 733.4.3 安全区域 743.5 垃圾收集器 753.5.1 Serial收集器 763.5.2 ParNew收集器 773.5.3 Parallel Scavenge收集器 793.5.4 Serial Old收集器 803.5.5 Parallel Old收集器 803.5.6 CMS收集器 813.5.7 G1收集器 843.5.8 理解GC日志 893.5.9 垃圾收集器参数总结 903.6 内存分配与回收策略 913.6.1 对象优先在Eden分配 913.6.2 大对象直接进入老年代 933.6.3 长期存活的对象将进入老年代 953.6.4 动态对象年龄判定 973.6.5 空间分配担保 983.7 本章小结 100 第4章 虚拟机性能监控与故障处理工具 1014.1 概述 1014.2 JDK的命令行工具 1014.2.1 jps：虚拟机进程状况工具 1044.2.2 jstat：虚拟机统计信息监视工具 1054.2.3 jinfo：Java配置信息工具 1064.2.4 jmap：Java内存映像工具 1074.2.5 jhat：虚拟机堆转储快照分析工具 1084.2.6 jstack：Java堆栈跟踪工具 1094.2.7 HSDIS：JIT生成代码反汇编 1114.3 JDK的可视化工具 1144.3.1 JConsole：Java监视与管理控制台 1154.3.2 VisualVM：多合一故障处理工具 1224.4 本章小结 131 第5章 调优案例分析与实战 1325.1 概述 1325.2 案例分析 1325.2.1 高性能硬件上的程序部署策略 1325.2.2 集群间同步导致的内存溢出 1355.2.3 堆外内存导致的溢出错误 1365.2.4 外部命令导致系统缓慢 1375.2.5 服务器JVM进程崩溃 1385.2.6 不恰当数据结构导致内存占用过大 1395.2.7 由Windows虚拟内存导致的长时间停顿 1415.3 实战：Eclipse运行速度调优 1425.3.1 调优前的程序运行状态 1425.3.2 升级JDK 1.6的性能变化及兼容问题 1455.3.3 编译时间和类加载时间的优化 1505.3.4 调整内存设置控制垃圾收集频率 1535.3.5 选择收集器降低延迟 1575.4 本章小结 160 第三部分 虚拟机执行子系统 第6章 类文件结构 1626.1 概述 1626.2 无关性的基石 1626.3 Class类文件的结构 1646.3.1 魔数与Class文件的版本 1666.3.2 常量池 1676.3.3 访问标志 1736.3.4 类索引、父类索引与接口索引集合 1746.3.5 字段表集合 1756.3.6 方法表集合 1786.3.7 属性表集合 1806.4 字节码指令简介 1966.4.1 字节码与数据类型 1976.4.2 加载和存储指令 1996.4.3 运算指令 2006.4.4 类型转换指令 2026.4.5 对象创建与访问指令 2036.4.6 操作数栈管理指令 2036.4.7 控制转移指令 2046.4.8 方法调用和返回指令 2046.4.9 异常处理指令 2056.4.10 同步指令 2056.5 公有设计和私有实现 2066.6 Class文件结构的发展 2076.7 本章小结 208 第7章 虚拟机类加载机制 2097.1 概述 2097.2 类加载的时机 2107.3 类加载的过程 2147.3.1 加载 2147.3.2 验证 2167.3.3 准备 2197.3.4 解析 2207.3.5 初始化 2257.4 类加载器 2277.4.1 类与类加载器 2287.4.2 双亲委派模型 2297.4.3 破坏双亲委派模型 2337.5 本章小结 235 第8章 虚拟机字节码执行引擎 2368.1 概述 2368.2 运行时栈帧结构 2368.2.1 局部变量表 2388.2.2 操作数栈 2428.2.3 动态连接 2438.2.4 方法返回地址 2438.2.5 附加信息 2448.3 方法调用 2448.3.1 解析 2448.3.2 分派 2468.3.3 动态类型语言支持 2588.4 基于栈的字节码解释执行引擎 2698.4.1 解释执行 2698.4.2 基于栈的指令集与基于寄存器的指令集 2708.4.3 基于栈的解释器执行过程 2728.5 本章小结 275 第9章 类加载及执行子系统的案例与实战 2769.1 概述 2769.2 案例分析 2769.2.1 Tomcat：正统的类加载器架构 2769.2.2 OSGi：灵活的类加载器架构 2799.2.3 字节码生成技术与动态代理的实现 2829.2.4 Retrotranslator：跨越JDK版本 2869.3 实战：自己动手实现远程执行功能 2899.3.1 目标 2909.3.2 思路 2909.3.3 实现 2919.3.4 验证 2989.4 本章小结 299 第四部分 程序编译与代

## &lt;&lt;深入理解Java虚拟机&gt;&gt;

码优化第10章 早期（编译期）优化 30210.1 概述 30210.2 Javac编译器 30310.2.1 Javac的源码与调试  
30310.2.2 解析与填充符号表 30510.2.3 注解处理器 30710.2.4 语义分析与字节码生成 30710.3 Java语法糖  
的味道 31110.3.1 泛型与类型擦除 31110.3.2 自动装箱、拆箱与遍历循环 31510.3.3 条件编译 31710.4 实  
战：插入式注解处理器 31810.4.1 实战目标 31810.4.2 代码实现 31910.4.3 运行与测试 32610.4.4 其他应用  
案例 32710.5 本章小结 328第11章 晚期（运行期）优化 32911.1 概述 32911.2 HotSpot虚拟机内的即时编  
译器 32911.2.1 解释器与编译器 33011.2.2 编译对象与触发条件 33211.2.3 编译过程 33711.2.4 查看及分析  
即时编译结果 33911.3 编译优化技术 34511.3.1 优化技术概览 34611.3.2 公共子表达式消除 35011.3.3 数  
组边界检查消除 35111.3.4 方法内联 35211.3.5 逃逸分析 35411.4 Java与C/C++的编译器对比 35611.5 本  
章小结 358第五部分 高效并发第12章 Java内存模型与线程 36012.1 概述 36012.2 硬件的效率与一致性  
36112.3 Java内存模型 36212.3.1 主内存与工作内存 36312.3.2 内存间交互操作 36412.3.3 对于volatile型变  
量的特殊规则 36612.3.4 对于long和double型变量的特殊规则 37212.3.5 原子性、可见性与有序性  
37312.3.6 先行发生原则 37512.4 Java与线程 37812.4.1 线程的实现 37812.4.2 Java线程调度 38112.4.3 状态  
转换 38312.5 本章小结 384第13章 线程安全与锁优化 38513.1 概述 38513.2 线程安全 38513.2.1 Java语言  
中的线程安全 38613.2.2 线程安全的实现方法 39013.3 锁优化 39713.3.1 自旋锁与自适应自旋 39813.3.2  
锁消除 39813.3.3 锁粗化 40013.3.4 轻量级锁 40013.3.5 偏向锁 40213.4 本章小结 403附录附录A 编  
译Windows版的OpenJDK 406附录B 虚拟机字节码指令表 414附录C HotSpot虚拟机主要参数表 420附  
录D 对象查询语言（OQL）简介 424附录E JDK历史版本轨迹 430

## <<深入理解Java虚拟机>>

### 编辑推荐

《深入理解Java虚拟机:JVM高级特性与最佳实践(第2版)》编辑推荐：超级畅销书全新升级，第1版两年内印刷近10次，Java图书领域公认的经典著作，繁体版台湾发行。基于最新JDK1.7，围绕内存管理、执行子系统、程序编译与优化、高效并发等核心主题对JVM进行全面而深入的分析，深刻揭示JVM的工作原理。以实践为导向，通过大量与实际生产环境相结合的案例展示了解决各种常见JVM问题的技巧和最佳实践。

<<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>