深入理解Java虚拟机：JVM高级特性与最佳实践（第2版）

作者：周志明

深入理解Java虚拟机

——JVM高级特性与最佳实践

周志明　著

ISBN：978-7-111-42190-0

本书纸版由机械工业出版社于2013年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目　录

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10000个人同时使用时就会缓慢、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（http://icyfenix.iteye.com/blog/1119214）。

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（http://www.hzbook.com）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。