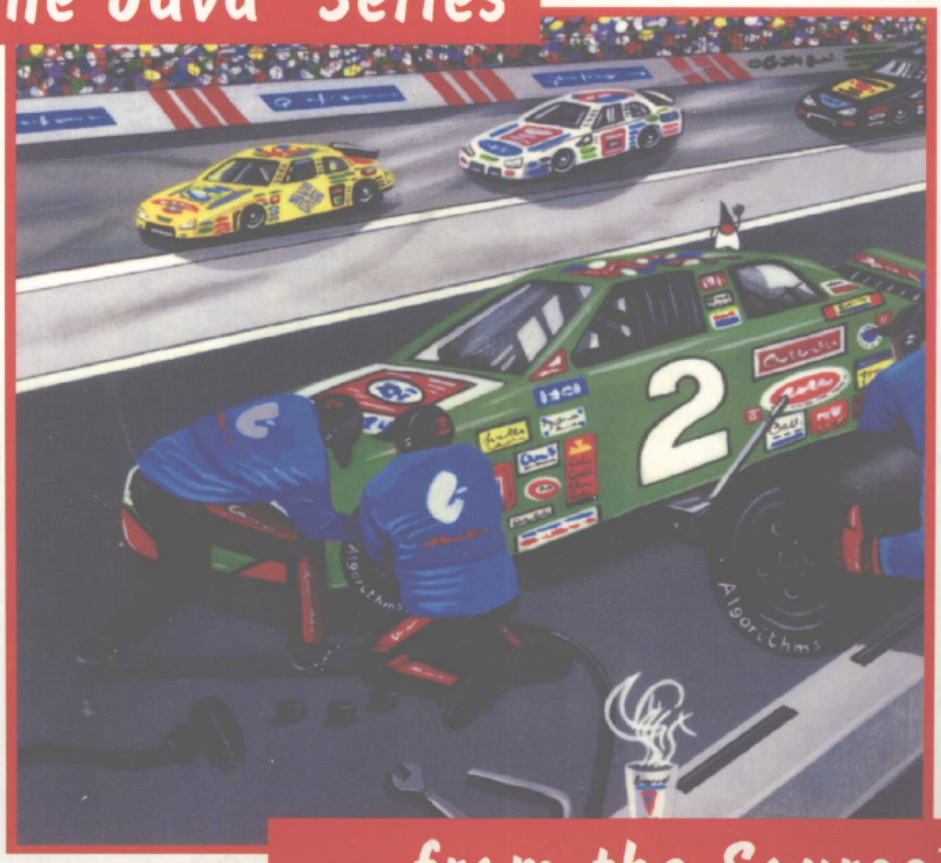
Steve Wilson · Jeff Kesselman 著
须晨 方梁 译

Java™ 平台性能
策略与技巧

The Java™ Series



... from the Source™

北京中软电子出版社

JAVA 平台性能

策略与技巧

Steve Wilson · Jeff Kesselman 著

须 晨 方 梁 译

北京中软电子出版社

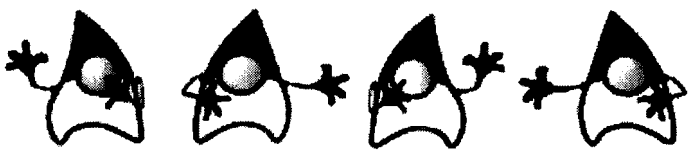
书　　名：**JAVA 平台性能策略与技巧**
作　　者：STEVE WILSON JEFF KESSELMAN
译　　者：须晨　方梁
责任编辑：高伟红
责任校对：方　梁
出版发行：北京中软电子出版社
地　　址：北京海淀区学院南路 55 号中软大厦 B 座 5 层
电　　话：010－62147079；51527258（传真）
电子邮件：susangwh@163.net
经　　销：各地新华书店
文本印刷：北京彩艺印刷有限公司
开本规格：787 毫米×1092 毫米　1/16 开本　13 印张　280 千字
版次印次：2003 年 1 月第一版　　2003 年 1 月第 1 次印刷
印　　数：1—3000 册
本 版 号：ISBN 7—90057—25—0
定　　价：37.70 元（含配套书、1CD）
说　　明：凡我社配套图书、光盘若有自然破损、缺页、脱页，本社负责调换。

简　介

自从 1995 年 Java 问世以来，因其能方便地开发和部署新软件而受到程序员和公司的极大欢迎。但是，Java 运行速度慢的抱怨也开始了。在 USENET 里搜索 comp.lang.jang.java.programmer 可以得到接近 3000 篇含有“java”和“slow”的文章。很明显，性能是许多开发者关注的问题。

在过去的几年里，技术的进步不断地改善着 Java 的性能。JIT(Just-in-time)编译器和先进的运行期系统，如 Sun 的 Java HotSpot 虚拟机显著地改善了 Java 的性能。另外，正如摩尔定律(Moore's Law)所言，计算机的功能每年都在不断增强。今天普通的 PC 已比 Java问世时的 PC 快了一个数量级。尽管有这些进步，Java系统运行速度慢的抱怨依然存在。

随着硬件的平均性能不断提高和核心运行期系统技术的不断改善，为什么还有性能问题困扰着如此多的开发者呢？答案显而易见，今天用 Java 开发的软件比一两年前已经复杂得多。开发者不断面临挑战——每次随着工具的更新，应用程序的复杂性和范围就不断地扩展。总之，技术实现的承诺不再是“舞蹈的公爵”一类的小程序(applet)了。



跳舞的公爵

目 录

译者序

许多程序员常常担心 Java 的性能，但很少去思考 Java 程序为什么慢以及有什么方法能提高 Java 程序的性能。本书正是回答了这些问题，既提供了有关 Java 垃圾回收和 HotSpot 虚拟机的基础知识，也提供了众多的实用技巧，如：如何减少 Java 程序的内存驻留，如何提高 I/O 性能，如何应用可变或不可变对象，如何控制类装载，如何应用本地方法，如何选择合适的数据结构和算法，如何编写响应灵敏的用户界面等。这些技巧建立在作者深刻理解 Java 语言的基础上，通过学习这些技巧程序员可加深对 Java 技术的理解，编写出性能更好的 Java 程序。总之，这是一本 Java 性能方面有价值的参考书。

一般地，调试 Java 程序的性能涉及四个方面：第一，Java 语言本身的特点，如垃圾回收机制和 HotSpot 虚拟机，以及 Java 各个 API 有关性能方面的考虑。我们可逐步总结出应用 Java 编程接口的性能规则，编码时遵循这些规则，就有助于提高程序的性能。例如，Vector 对象的构造函数有两个性能调节参数：容量和装载因子，同时，Vector 对象的每个方法都是同步的。从性能的观点看，ArrayList 类要更好一些。第二，程序的设计。如果程序是一个分布式的应用，从程序性能上考虑，就应考虑粗粒度的方法和数据。另外，应掌握一些提高程序性能的基本技术，如缓冲池。第三，策略。在整个开发过程中必须将性能验证融入开发过程，在整个项目过程中，不断跟踪程序性能的变化，及早发现程序性能的问题。第四，应用性能调试工具。性能调试离不开工具的支持，调试复杂的应用时尤其如此。现在已有很多商业化的性能调试工具，如 JProbe，optimizeit 等。总而言之，Java 程序的性能是一个程序的整体表现，应谨慎全面地考虑程序的多个方面。

本书是 Sun 公司 Java 系列丛书之一，作者是 Sun Microsystems 的 Swing 开发小组成员，他们是 Steve Wilson 和 Jeff Kesselman。

本书的中文简体版是须晨、方梁翻译的；李晓霞和洪伟提供了帮助，在此深表感谢。翻译不当之处，请不吝指教，译者 EMAIL：fangbook@yahoo.com。

须晨、方梁　谨识

2002 年 12 月

序　言

1997年我受雇从事Java基础类(Java Foundation Classes,简称JFC)Swing工具包的研发工作。这是一个雄心勃勃的举动——Swing将成为Java开发图形用户界面(GUIs)的新标准。Swing发布之前,Java平台唯一的GUI工具包是基于90年代标准、相当原始的抽象窗口工具包(Abstract Windows Toolkit,简称AWT)。与AWT"最小公分母"的笨拙设计相比,Swing是一个充满艺术性设计的工具包。它完全由Java实现,具有功能强大的模型-视图架构(Model-View architecture),先进的组件集和革命性的嵌入式外观系统(pluggable look-and-feel,简称PLAF)。当JFC在1998年年中发布后,它立即被成千上万热切的开发者迅速应用。

像任何成功的新产品一样,成功总伴随着一些尖锐的抱怨。一些开发者抱怨架构和设计的哲学问题,一些开发者则抱怨设计错误或缺少某个特性。但是,令我个人最烦恼的是对Swing程序很慢的抱怨。

在说服经理让我花一个星期来研究Swing的性能问题后,我下载了一个资源分布分析(profiling)工具包的试用版,开始深入研究Swing工具包里的各个部分。

结果证明,在多个领域里提高Swing性能相对容易。一周后,我提交了一个结果报告并发送给其余Swing开发组成员。小组的其他成员开始认识到性能调试的重要性,并开始了自己的性能分析。以后的几个月里,我越来越多地从事性能分析和调试的工作,同时Swing小组逐步改进了Swing的性能。书中的知识是我们调试Swing过程中获得的。

1998年末我们发布了一个Swing的新版本,其典型任务的运行速度超过前一个版本的两倍。但是,在一些开发者满意这些改进的同时,我们依然陷在各种性能抱怨的困境里。很明显,性能问题的复杂性已超过了我们当初的想象。

1998年末我加入Sun Java软件部门的性能小组,担心解决性能问题将成为我的全职工作。为了更好地理解我们和开发者面临的性能问题,我花了大量时间跟那些严谨的、开

发实际应用的 Java 技术开发者交流。有时开发者指出 Java 类库或虚拟机(VM)的更新提高了他们程序的性能。性能小组的宗旨是确定何时更新这些新变化并将它们融入到运行环境(runtime)中。

与这些开发者一起工作时,我们经常发现修改他们编写的代码就能提高程序的性能,同时,我们发现他们错误地理解了 Java 技术性能特点,甚至对更一般的性能调试也是如此。

本书的目的是与读者分享我们在调试基于 Java 技术的系统的性能过程中所获得的经验和知识,希望本书能成为有价值的性能参考指引。

Steve Wilson
Sun Microsystems

关于本书

本书能帮助读者编写 Java 平台的高性能软件,包括将性能调试整合到软件开发过程的高端策略和代码级性能调试的技巧。

本书分为两部分,从不同的视角来讨论性能调试,提供了性能调试过程的整体视图。

第一部分:策略,概要描述了性能调试过程,重点是将一般策略融入开发过程,改善基于 Java 技术系统的性能。

第二部分:实用技巧,重点讨论在找出程序热点(hot spots)和瓶颈后提高性能的具体技术。

策略部分的更高端的知识针对广泛的读者,包括软件工程师,项目经理,技术经理和基于 Java 技术解决方案的质量保证专家。技巧部分的知识适合寻求提高软件运行速度的具体编码技术并熟悉 Java 的中级和高级开发者。

策略的章节最好作为一个整体来阅读,而技巧的章节不必逐一阅读,可以直接从你最感兴趣的任何主题开始学习。

书末的两个附录提供了垃圾回收(garbage collection)和热点虚拟机(HotSpot VM)的有关信息,并介绍了它们如何影响性能。

性能测量

除非另外提示,本书描述的所有性能测量的示例在 Java2 标准版(J2SE)v1.3 的预发布版上运行,其测试环境是微软视窗操作系统上的 HotSpot 客户端虚拟机(HotSpot Client VM)。

具体的性能测量结果仅是具体测试配置环境的反映。诸多因素,如 CPU,硬盘,操作系统和 Java 运行环境(Java runtime environment,简称 JRE)都可影响程序的性能。注意:在不同配置的环境里相同的基准测试可能产生截然不同的结果。

代码示例

本书的代码片段,示例程序、公用代码和基准测试代码可在 *http://java.sun.com/*

docs/books/performance/找到。

致谢

我们对为本书的出版作出贡献的人们表示感谢。

Jon Kannegaard、Larry Abrahams 和 Graham Hamilton 在初期推动了项目的开展。特别感谢 Larry,他在整个项目过程中提供了必要的管理支持。

Java 系列丛书的编辑 Lisa Friendly 和 Tim Lindholm,在本书编写的全过程中耐心地指导了两个第一次写书的作者。

Addison-Wesley 的 Mike Hendrickson 和 Julie DiNicola 在整个过程中给予了作者巨大的帮助。

The Design Cage 的 Deborah Adair 既是编辑,又是图形设计者,还担任我们的写作辅导。没有她的帮助我们不可能完成本书。

Hans Muller, Swing 项目的技术负责人和 Swing 线程模型的顶级专家提供了第十一章.用 Swing 编写反应灵敏的用户界面。他花费了许多晚上和周末编写第十一章,使我们能更好地理解在 Swing 程序中如何使用线程。

Alan Sommerer 筹划了初稿的组织和概要,从而保证没有遗漏关键概念。

David Wilson 和 Doris Chen 在我们编写本书时开始编写一个两天的性能调优培训课程。我们与他们交流了许多想法,使双方受益非浅。

过去的一年里,Agnes Jacob 为我们介绍了许多从事性能相关工作的开发者。他们宝贵的经验决定了本书的内容。

许多人审阅了本书的初稿并与作者进行了有益的技术交流,本书的很多内容来源于这些交流的结果。他们是:Eric Armstrong, Tom Ball, Clifford Crick, Mark Davidson, Joshua Engel, Peter Haggar, Howard Harkness, Cay Horstmann, Peter Kessler, Gary Little, Mike Martak, Mike McCloskey, Dave Mendenhall, Philip Milne, Srdjan Mitrovic, Bill Pataky, Nancy Schorr, 和 David Stoutamire。

第一部分

策　略

策略，1.计划和指导大规模军事行动的方式和艺术。
2.达到目标的计划或方法。
——兰登出版社《韦伯词典》

本部分的章节介绍了在开发过程中提高 Java 系统性能的策略。无论是软件工程师，项目经理，技术经理或质量保证专家，这些内容能帮助他们更好地理解如何达到程序性能目标。至于用于改善 Java 程序性能的具体编程技巧，请参见本书的第二部分。

第 1 章　性能是什么

性能是什么？这看起来是一个简单的问题，但是，实际上性能有多个重要的方面。本章讨论有关性能的一般问题以及分析影响 Java 系统性能的不同因素。

第 2 章　性能过程

我们不仅需要了解影响性能的因素，还要知道如何将性能调试融入开发过程。本章说明了如何将性能调试融入软件开发周期并成为其中的一部分。

第 3 章　测量是一切

如果不评估变化的影响就不可能有效地调试软件。本章讨论了测量软件性能的不同方法以及如何分析和应用这些结果。