

通过200余个示例详解Java虚拟机的各种参数配置、故障排查、性能监控及优化  
技术全面，通俗易懂，实例典型，实用性强

**Broadview**<sup>®</sup>  
www.broadview.com.cn



# 实战 Java虚拟机

## JVM故障诊断与性能优化

(第2版)

葛一鸣 | 著

 中国工信出版集团

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

# 实战 Java 虚拟机

## JVM 故障诊断与性能优化

(第2版)

葛一鸣 | 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

不管技术如何发展, Java 依然是一个充满活力的生态圈, 学习 Java 的人也越来越多, 但多数人学习 Java 虚拟机 (JVM) 时都会遇到瓶颈。本书将通过 200 余个示例详细介绍 JVM 中的各种参数配置、故障排查、性能监控及性能优化, 帮助 Java 人突破瓶颈。

本书共 11 章, 修订后版本涵盖 Java 6~Java 10。第 1~3 章介绍了 Java 虚拟机的定义、总体架构、常用配置参数。第 4~5 章介绍了垃圾回收的算法和各种垃圾回收器。第 6 章介绍了 Java 虚拟机的性能监控和故障诊断工具。第 7 章详细介绍了对 Java 堆的分析方法和案例。第 8 章介绍了 Java 虚拟机对多线程, 尤其是对锁的支持。第 9~10 章介绍了 Java 虚拟机的核心——Class 文件结构, 以及 Java 虚拟机中类的装载系统。第 11 章介绍了 Java 虚拟机的执行系统和字节码, 并给出了通过 ASM 框架进行字节码注入的案例。

本书不仅适合 Java 程序员阅读, 还适合工作于 Java 虚拟机之上的研发人员、软件设计师、架构师阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

### 图书在版编目 (CIP) 数据

实战 Java 虚拟机: JVM 故障诊断与性能优化 / 葛一鸣著. — 2 版. — 北京: 电子工业出版社, 2019.7  
ISBN 978-7-121-36774-8

I. ①实… II. ①葛… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2019) 第 106658 号

责任编辑: 董 英

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16 印张: 28.5

字数: 654 千字

版 次: 2015 年 10 月第 1 版

2019 年 7 月第 2 版

印 次: 2019 年 7 月第 1 次印刷

定 价: 89.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

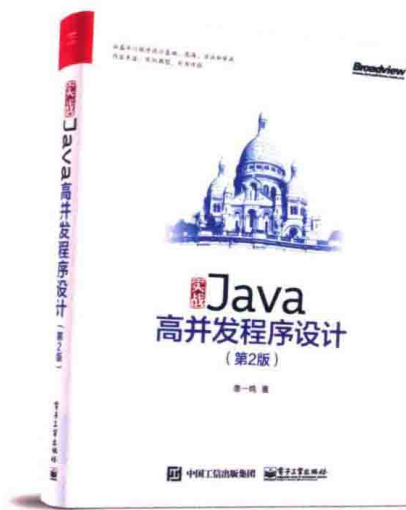
本书咨询联系方式: 010-51260888-819, [faq@phei.com.cn](mailto:faq@phei.com.cn)。

## 作者简介



葛一鸣

51CTO特约讲师，国家认证系统分析师，获得Oracle OCP认证。长期从事Java软件开发工作，对Java程序设计、JVM有深入的研究，对设计模式、人工智能、神经网络、数据挖掘等技术有浓厚兴趣，开设了在线的《深入浅出Java虚拟机——入门篇》培训课程，有问题可联系作者个人博客[www.uucode.net](http://www.uucode.net)。



КЕТУМРРОЕМ. КК. ККККК  
КККК КККК КККК

Dreamflow



# Java 高并发程序设计 (第2版)

郭剑强

中国工业出版社 清华大学出版社

# 前 言

## 关于 Java 生态圈

Java 是目前应用最广泛的软件开发平台之一。随着 Java 及 Java 社区的不断壮大，Java 也早已不再是简简单单的一门计算机语言了，它更是一个平台、一种文化、一个社区。

作为一个平台，Java 虚拟机扮演着举足轻重的作用。除了 Java 语言，任何一种能够被编译成字节码的计算机语言都属于 Java 这个平台。Groovy、Scala、JRuby、Kotlin 等都是 Java 平台的一部分，它们依赖于 Java 虚拟机，同时，Java 平台也因为它们变得更加丰富多彩。

作为一种文化，Java 几乎成了“开源”的代名词。在 Java 程序中，有着数不清的开源软件和框架，如 Tomcat、Struts、Hibernate、Spring 等。就连 JDK 和 JVM 自身也有不少开源的实现，如 OpenJDK、Harmony。可以说，“共享”的精神在 Java 世界里体现得淋漓尽致。

作为一个社区，Java 拥有无数的开发人员，有数不清的论坛和资料。从桌面应用软件、嵌入式开发到企业级应用、后台服务器、中间件，都可以看到 Java 的身影。其应用形式之复杂、参与人数之众多也令人咋舌。可以说，Java 社区俨然已经成为一个良好而庞大的生态系统，而本书，将主要介绍这个生态系统的核心——**Java 虚拟机**。

## 第 2 版的重点修订

作者希望本书第 2 版的内容能够涵盖 JDK 7~JDK 10，所以特做如下修订：

- JDK 10 源码环境的下载与搭建。
- JDK 10 JVM 的调试方法介绍。
- JDK 10 运行参数的演变。
- JDK 10 用 G1 垃圾回收器替代 CMS 垃圾回收器，成为默认的垃圾回收器。
- JDK 10 调试工具的更新，废除功能类似的调试工具，重点推荐使用 Visual VM。

## 本书的体系结构

本书立足于实际开发，又不缺乏理论介绍，力求通俗易懂、循序渐进。全书共分为 11 章。

第 1 章综述，介绍了 Java 虚拟机的概念、定义，讲解了 Java 语言规范和 Java 虚拟机规范，最后还介绍了 OpenJDK 的调试方法。

第 2 章介绍了 Java 虚拟机的总体架构，说明了堆、栈、方法区等内存空间的作用和彼此之间的联系。

第 3 章介绍了 Java 虚拟机的常用配置参数，重点对垃圾回收跟踪参数、内存配置参数做了详细介绍，并给出了案例说明。

第 4 章从理论层面介绍了垃圾回收的算法，如引用计数、标记清除、标记压缩、复制算法等。本章是第 5 章的理论基础。

第 5 章基于垃圾回收的理论知识，进一步详细介绍了 Java 虚拟机中实际使用的各种垃圾回收器，包括串行回收器、并行回收器、CMS、G1 等。

第 6 章介绍了 Java 虚拟机的性能监控和故障诊断工具，考虑到实用性，也介绍了系统级性能监控工具的使用，两者结合，可以更好地帮助读者处理实际问题。

第 7 章详细介绍了对 Java 堆的分析方法和案例，主要讲解了 MAT 和 Visual VM 两款工具的使用，以及各自 OQL 的编写方式。

第 8 章介绍了 Java 虚拟机对多线程，尤其是对锁的支持。本章不仅介绍了虚拟机内部锁的实现、优化机制，也给出了 Java 语言层面的一些锁优化思路，最后还介绍了无锁的并行控制方法。

第 9 章介绍了 Java 虚拟机的核心——Class 文件结构，Class 文件作为 Java 虚拟机的基石，有着举足轻重的作用，对深入理解 Java 虚拟机有着不可忽视的作用。

第 10 章介绍了 Java 虚拟机中类的装载系统，其中着重介绍了 Java 虚拟机中 ClassLoader

的实现及设计模式。

第 11 章介绍了 Java 虚拟机的执行系统和字节码。为了帮助读者更快、更好地理解 Java 字节码，本章对字节码进行了分类讲解，并且理论联系实际，给出了通过 ASM 框架进行字节码注入的案例。

## 本书特色

本书的主要特点有：

- **结构清晰。**本书采用从整体到局部的视角，首先，第 1、2 章介绍了 Java 虚拟机的整体概况和结构。接着步步为营，每一章节对应单独的知识点，力求展示虚拟机的全貌。
- **理论结合实战。**本书不甘心于简单地枚举理论知识，在每一个理论背后，都给出了演示示例供读者参考，帮助读者更好地消化这些理论。比如，在对 Class 文件结构和字节码的介绍中，不仅仅简单地给出了理论说明，更是使用 ASM 框架将这些理论应用于实践，尽可能地做到理论与实践结合。
- **专注专业。**本书着眼于 Java 虚拟机，对 Java 虚拟机的原理和实践做了丰富的介绍，包括但不限于体系结构、虚拟机的调试方式、常用参数、垃圾回收系统、Class 文件结构、执行系统等，力求从多角度更专业地对 Java 虚拟机进行探讨。
- **通俗易懂。**本书依然服务于广大虚拟机初学者，尽量避免采用过于理论化的描述方式，简单的白话文风格贯穿全书，尽量使读者在阅读过程中少盲点、无盲点。
- **技术全面。**纵横 Windows 和 Linux 双系统下的性能诊断、涉及 32 位系统和 64 位系统的优化比较、贯穿从 JDK 1.5 到 JDK 10 的优化演进。

## 适合阅读人群

虽然本书讲解力求通俗，但要通读本书并取得良好的学习效果，要求读者需要具备基本的 Java 知识或者一定的编程经验。因此，本书适合以下读者：

- 拥有一定开发经验的 Java 平台开发人员（Java、Scala、JRuby 等）。
- 软件设计师、架构师。
- 系统调优人员。

- 有一定的 Java 编程基础并希望进一步理解 Java 的程序员。
- 虚拟机爱好者，JVM 实践者。

## 本书的约定

本书在叙述过程中，有如下约定：

- 本书中所述的 JDK 1.5、JDK 1.6、JDK 1.7、JDK 1.8、JDK 1.9、JDK 1.10 等同于 JDK 5、JDK 6、JDK 7、JDK 8、JDK 9、JDK 10。
- 如无特殊说明，Java 虚拟机均指 HotSpot 虚拟机。
- 如无特殊说明，本书的程序、示例均在 JDK 7~JDK 10 环境中运行。

## 联系作者

本书的写作过程远比想象的艰辛，为了让全书能够更清楚、更正确地表达和论述，我经历了好多个不眠之夜，即使现在回想起来，也忍不住会打个冷战。由于写作水平的限制，书中难免会有不妥之处，望读者谅解。

为此，如果读者有任何疑问或者建议，非常欢迎大家加入 QQ 群 254693571，一起探讨学习中的困难、分享学习的经验，我期待与大家一起交流、共同进步。同时，也希望大家可以关注我的博客 <http://www.uucode.net/>。

## 感谢

这本书能够面世，是因为得到了众人的支持。首先，要感谢我的妻子，她始终不辞辛劳，毫无怨言地对我照顾有加，才让我得以腾出大量时间，并可以安心工作。其次，要感谢编辑为我一次又一次地审稿改错、批评指正，才让本书逐步完善。最后，感谢我的母亲 30 年如一日对我的体贴和关心。

特别感谢网友千马奕在 JDK 1.9、JDK1.10 下对本书所做的测试和修订。

## 读者服务

轻松注册成为博文视点社区用户 ([www.broadview.com.cn](http://www.broadview.com.cn))，您即可享受以下服务：

- **下载资源：**本书如提供示例代码及资源文件，均可在 [下载资源](#) 处下载。
- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/36774>



# 目 录

第 1 章 初探 Java 虚拟机.....	1
1.1 知根知底：追溯 Java 的发展历程.....	2
1.1.1 那些依托 Java 虚拟机的语言大咖们.....	2
1.1.2 Java 发展史上的里程碑.....	2
1.2 跨平台的真相：Java 虚拟机做中介.....	4
1.2.1 理解 Java 虚拟机的原理.....	4
1.2.2 看清 Java 虚拟机的种类.....	5
1.3 一切看我的：Java 语言规范.....	6
1.3.1 词法的定义.....	6
1.3.2 语法的定义.....	7
1.3.3 数据类型的定义.....	8
1.3.4 Java 语言规范总结.....	9
1.4 一切听我的：Java 虚拟机规范.....	9
1.5 数字编码就是计算机世界的水和电.....	10
1.5.1 整数在 Java 虚拟机中的表示.....	10
1.5.2 浮点数在 Java 虚拟机中的表示.....	12
1.6 抛砖引玉：编译和调试虚拟机.....	14
1.7 小结.....	18
第 2 章 认识 Java 虚拟机的基本结构.....	19
2.1 谋全局者才能成大器：看穿 Java 虚拟机的架构.....	19

2.2	小参数能解决大问题：学会设置 Java 虚拟机的参数.....	21
2.3	对象去哪儿：辨清 Java 堆.....	22
2.4	函数如何调用：出入 Java 栈.....	24
2.4.1	局部变量表 .....	26
2.4.2	操作数栈 .....	31
2.4.3	帧数据区 .....	31
2.4.4	栈上分配 .....	32
2.5	类去哪儿了：识别方法区.....	34
2.6	小结 .....	36
<b>第 3 章</b>	<b>常用 Java 虚拟机参数 .....</b>	<b>37</b>
3.1	一切运行都有迹可循：掌握跟踪调试参数.....	37
3.1.1	跟踪垃圾回收——读懂虚拟机日志.....	38
3.1.2	类加载/卸载的跟踪 .....	42
3.1.3	查看系统参数.....	45
3.2	让性能飞起来：学习堆的配置参数.....	45
3.2.1	最大堆和初始堆的设置.....	46
3.2.2	新生代的配置.....	49
3.2.3	堆溢出处理 .....	53
3.3	别让性能有缺口：了解非堆内存的参数配置.....	54
3.3.1	方法区配置 .....	55
3.3.2	栈配置 .....	55
3.3.3	直接内存配置.....	55
3.4	Client 和 Server 二选一：虚拟机的工作模式.....	58
3.5	小结 .....	59
<b>第 4 章</b>	<b>垃圾回收的概念与算法 .....</b>	<b>60</b>
4.1	内存管理清洁工：认识垃圾回收.....	60
4.2	清洁工具大 PK：讨论常用的垃圾回收算法.....	61
4.2.1	引用计数法（Reference Counting） .....	62
4.2.2	标记清除法（Mark-Sweep） .....	63
4.2.3	复制算法（Copying） .....	64

4.2.4	标记压缩法（Mark-Compact） .....	66
4.2.5	分代算法（Generational Collecting） .....	67
4.2.6	分区算法（Region） .....	68
4.3	谁才是真正的垃圾：判断可触及性 .....	69
4.3.1	对象的复活 .....	69
4.3.2	引用和可触及性的强度 .....	71
4.3.3	软引用——可被回收的引用 .....	72
4.3.4	弱引用——发现即回收 .....	76
4.3.5	虚引用——对象回收跟踪 .....	77
4.4	垃圾回收时的停顿现象：Stop-The-World 案例实战 .....	79
4.5	小结 .....	83
<b>第 5 章</b>	<b>垃圾收集器和内存分配 .....</b>	<b>84</b>
5.1	一心一意一件事：串行回收器 .....	85
5.1.1	新生代串行回收器 .....	85
5.1.2	老年代串行回收器 .....	86
5.2	人多力量大：并行回收器 .....	86
5.2.1	新生代 ParNew 回收器 .....	87
5.2.2	新生代 ParallelGC 回收器 .....	88
5.2.3	老年代 ParallelOldGC 回收器 .....	89
5.3	一心多用都不落下：CMS 回收器（JDK 8 及之前的版本） .....	90
5.3.1	CMS 主要工作步骤 .....	90
5.3.2	CMS 主要的参数 .....	91
5.3.3	CMS 的日志分析 .....	93
5.3.4	有关 Class 的回收 .....	94
5.4	未来我做主：G1 回收器（JDK 9 及之后版本的默认回收器） .....	95
5.4.1	G1 的内存划分和主要收集过程 .....	95
5.4.2	G1 的新生代 GC .....	96
5.4.3	G1 的并发标记周期 .....	97
5.4.4	混合回收 .....	100
5.4.5	必要时的 Full GC .....	102
5.4.6	G1 的日志 .....	102

5.4.7	G1 相关的参数.....	106
5.5	回眸：有关对象内存分配和回收的一些细节问题.....	106
5.5.1	禁用 System.gc() .....	107
5.5.2	System.gc()使用并发回收 .....	107
5.5.3	并行 GC 前额外触发的新生代 GC.....	109
5.5.4	对象何时进入老年代.....	109
5.5.5	在 TLAB 上分配对象 .....	117
5.5.6	finalize()函数对垃圾回收的影响 .....	120
5.6	温故又知新：常用的 GC 参数 .....	125
5.7	动手才是真英雄：垃圾回收器对 Tomcat 性能影响的实验.....	127
5.7.1	配置实验环境.....	127
5.7.2	配置性能测试工具 JMeter.....	128
5.7.3	配置 Web 应用服务器 Tomcat.....	131
5.7.4	实战案例 1——初试串行回收器.....	133
5.7.5	实战案例 2——扩大堆以提升系统性能.....	133
5.7.6	实战案例 3——调整初始堆大小.....	134
5.7.7	实战案例 4——使用 ParrelOldGC 回收器.....	135
5.7.8	实战案例 5——使用较小堆提高 GC 压力 .....	135
5.7.9	实战案例 6——测试 ParallelOldGC 的表现 .....	135
5.7.10	实战案例 7——测试 ParNew 回收器的表现 .....	136
5.7.11	实战案例 8——测试 JDK 1.8 的表现 .....	136
5.7.12	实战案例 9——使用高版本虚拟机提升性能.....	137
5.8	小结 .....	137
<b>第 6 章</b>	<b>性能监控工具 .....</b>	<b>138</b>
6.1	有我更高效：Linux 下的性能监控工具 .....	139
6.1.1	显示系统整体资源使用情况——top 命令.....	139
6.1.2	监控内存和 CPU——vmstat 命令 .....	140
6.1.3	监控 I/O——iostat 命令.....	142
6.1.4	多功能诊断器——pidstat 工具 .....	143
6.2	用我更高效：Windows 下的性能监控工具 .....	148
6.2.1	任务管理器 .....	148

6.2.2	perfmon 性能监控工具 .....	150
6.2.3	Process Explorer 进程管理工具 .....	153
6.2.4	pslist 命令——Windows 下也有命令行工具 .....	155
6.3	外科手术刀：JDK 性能监控工具 .....	157
6.3.1	查看 Java 进程——jps 命令 .....	158
6.3.2	查看虚拟机运行时信息——jstat 命令 .....	159
6.3.3	查看虚拟机参数——jinfo 命令 .....	162
6.3.4	导出堆到文件——jmap 命令 .....	163
6.3.5	JDK 自带的堆分析工具——jhat 命令 .....	165
6.3.6	查看线程堆栈——jstack 命令 .....	168
6.3.7	远程主机信息收集——jstatd 命令 .....	171
6.3.8	多功能命令行——jcmd 命令 .....	173
6.3.9	性能统计工具——hprof .....	175
6.3.10	扩展 jps 命令 .....	178
6.4	我是你的眼：图形化虚拟机监控工具 JConsole .....	178
6.4.1	JConsole 连接 Java 程序 .....	179
6.4.2	Java 程序概况 .....	180
6.4.3	内存监控 .....	180
6.4.4	线程监控 .....	181
6.4.5	类加载情况 .....	183
6.4.6	虚拟机信息 .....	183
6.5	一目了然：可视化性能监控工具 Visual VM .....	184
6.5.1	Visual VM 连接应用程序 .....	185
6.5.2	监控应用程序概况 .....	186
6.5.3	Thread Dump 和分析 .....	187
6.5.4	性能分析 .....	188
6.5.5	内存快照分析 .....	190
6.5.6	BTrace 介绍 .....	191
6.6	来自 JRockit 的礼物：虚拟机诊断工具 Mission Control .....	199
6.6.1	MBean 服务器 .....	199
6.6.2	飞行记录器（Flight Recorder） .....	201
6.7	小结 .....	204

<b>第 7 章 分析 Java 堆</b> .....	<b>205</b>
7.1 对症下药：找到内存溢出的原因.....	206
7.1.1 堆溢出.....	206
7.1.2 直接内存溢出.....	206
7.1.3 过多线程导致 OOM.....	208
7.1.4 永久区溢出.....	210
7.1.5 GC 效率低下引起的 OOM.....	211
7.2 无处不在的字符串：String 在虚拟机中的实现.....	211
7.2.1 String 对象的特点.....	211
7.2.2 有关 String 的内存泄漏.....	213
7.2.3 有关 String 常量池的位置.....	216
7.3 虚拟机也有内窥镜：使用 MAT 分析 Java 堆.....	218
7.3.1 初识 MAT.....	218
7.3.2 浅堆和深堆.....	221
7.3.3 MAT 堆分析案例解析.....	222
7.3.4 支配树（Dominator Tree）.....	226
7.3.5 Tomcat 堆溢出分析.....	227
7.4 筛选堆对象：MAT 对 OQL 的支持.....	231
7.4.1 Select 子句.....	231
7.4.2 From 子句.....	233
7.4.3 Where 子句.....	235
7.4.4 内置对象与方法.....	235
7.5 更精彩的查找：Visual VM 对 OQL 的支持.....	240
7.5.1 Visual VM 的 OQL 基本语法.....	240
7.5.2 内置 heap 对象.....	241
7.5.3 对象函数.....	243
7.5.4 集合/统计函数.....	247
7.5.5 程序化 OQL 分析 Tomcat 堆.....	253
7.6 小结.....	256
<b>第 8 章 锁与并发</b> .....	<b>257</b>
8.1 安全就是锁存在的理由：锁的基本概念和实现.....	258

8.1.1	理解线程安全.....	258
8.1.2	对象头和锁.....	260
8.2	避免残酷的竞争：锁在 Java 虚拟机中的实现和优化.....	261
8.2.1	偏向锁.....	261
8.2.2	轻量级锁.....	263
8.2.3	锁膨胀.....	264
8.2.4	自旋锁.....	265
8.2.5	锁消除.....	265
8.3	应对残酷的竞争：锁在应用层的优化思路.....	267
8.3.1	减少锁持有时间.....	267
8.3.2	减小锁粒度.....	268
8.3.3	锁分离.....	270
8.3.4	锁粗化.....	272
8.4	无招胜有招：无锁.....	274
8.4.1	理解 CAS.....	274
8.4.2	原子操作.....	275
8.4.3	新宠儿 LongAdder.....	278
8.5	将随机变为可控：理解 Java 内存模型.....	281
8.5.1	原子性.....	281
8.5.2	有序性.....	283
8.5.3	可见性.....	285
8.5.4	Happens-Before 原则.....	287
8.6	小结.....	287

## 第 9 章 Class 文件结构.....288

9.1	不仅跨平台，还能跨语言：语言无关性.....	288
9.2	虚拟机的基石：Class 文件.....	290
9.2.1	Class 文件的标志——魔数.....	291
9.2.2	Class 文件的版本.....	293
9.2.3	存放所有常数——常量池.....	294
9.2.4	Class 的访问标记（Access Flag）.....	301
9.2.5	当前类、父类和接口.....	302