

《实战Java虚拟机》

图书基本信息

书名：《实战Java虚拟机》

13位ISBN编号：9787121256126

出版时间：2015-3

作者：葛一鸣

页数：452

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《实战Java虚拟机》

内容概要

随着越来越多的第三方语言（Groovy、Scala、JRuby等）在Java虚拟机上运行，Java也俨然成为了一个充满活力的生态圈。《实战Java虚拟机——JVM故障诊断与性能优化》将通过200余示例详细介绍Java虚拟机中的各种参数配置、故障排查、性能监控以及性能优化。

《实战Java虚拟机——JVM故障诊断与性能优化》共11章。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。

《实战Java虚拟机——JVM故障诊断与性能优化》不仅适合Java程序员，还适合任何一名工作于Java虚拟机之上的研发人员、软件设计师、架构师。

《实战Java虚拟机》

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，著有《自己动手写神经网络》电子书，也开设了在线的《深入浅出Java虚拟机——入门篇》培训课程。

书籍目录

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虚拟机来做中介

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

19

第2章 认识Java虚拟机的基本结构

20

2.1 谋全局者才能成大器：看穿Java虚拟机的架构

20

2.2 小参数能解决大问题：学会设置Java虚拟机的参数

22

2.3 对象去哪儿：辨清Java堆

23

2.4 函数如何调用：出入Java栈

25

2.4.1 局部变量表

27	
2.4.2	操作数栈
32	
2.4.3	帧数据区
32	
2.4.4	栈上分配
33	
2.5	类去哪儿了：识别方法区
35	
2.6	小结
37	
第3章	常用Java虚拟机参数
38	
3.1	一切运行都有迹可循：掌握跟踪调试参数
38	
3.1.1	跟踪垃圾回收——读懂虚拟机日志
39	
3.1.2	类加载/卸载的跟踪
42	
3.1.3	系统参数查看
44	
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
45	
3.2.2	新生代的配置
49	
3.2.3	堆溢出处理
52	
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	
第4章	垃圾回收概念与算法
60	
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
第5章 垃圾收集器和内存分配	84
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	90
5.3.1 CMS主要工作步骤	90
5.3.2 CMS主要的设置参数	91
5.3.3 CMS的日志分析	

92	
5.3.4	有关Class的回收
94	
5.4	未来我做主：G1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
107	
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
110	
5.5.5	在TLAB上分配对象
117	
5.5.6	方法finalize()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	配置进行性能测试的工具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 1.6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
第6章 性能监控工具	138
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IO使用——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	

167
6.3.7 远程主机信息收集——jstatd命令
170
6.3.8 多功能命令行——jcmd命令
172
6.3.9 性能统计工具——hprof
175
6.3.10 扩展jps命令
177
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
178
6.4.2 Java程序概况
179
6.4.3 内存监控
180
6.4.4 线程监控
180
6.4.5 类加载情况
182
6.4.6 虚拟机信息
182
6.5 一目了然：可视化性能监控工具Visual VM
183
6.5.1 Visual VM连接应用程序
184
6.5.2 监控应用程序概况
185
6.5.3 Thread Dump和分析
186
6.5.4 性能分析
187
6.5.5 内存快照分析
189
6.5.6 BTrace介绍
190
6.6 来自JRockit的礼物：虚拟机诊断工具Mission Control
198
6.6.1 MBean服务器
198
6.6.2 飞机记录器（Flight Recorder）
200
6.7 小结
203
第7章 分析Java堆
204
7.1 对症下药：找到内存溢出的原因
205

7.1.1 堆溢出	205
7.1.2 直接内存溢出	205
7.1.3 过多线程导致OOM	207
7.1.4 永久区溢出	209
7.1.5 GC效率低下引起的OOM	210
7.2 无处不在的字符串：String在虚拟机中的实现	210
7.2.1 String对象的特点	210
7.2.2 有关String的内存泄漏	212
7.2.3 有关String常量池的位置	215
7.3 虚拟机也有内窥镜：使用MAT分析Java堆	217
7.3.1 初识MAT	217
7.3.2 浅堆和深堆	220
7.3.3例解MAT堆分析	221
7.3.4支配树（Dominator Tree）	225
7.3.5 Tomcat堆溢出分析	226
7.4 筛选堆对象：MAT对OQL的支持	230
7.4.1 Select子句	230
7.4.2 From子句	232
7.4.3 Where子句	234
7.4.4 内置对象与方法	234
7.5 更精彩的查找：Visual VM对OQL的支持	239
7.5.1 Visual VM的OQL基本语法	239
7.5.2 内置heap对象	240
7.5.3 对象函数	242
7.5.4 集合/统计函数	

247	
7.5.5	程序化OQL分析Tomcat堆
252	
7.6	小结
255	
第8章	锁与并发
256	
8.1	安全就是锁存在的理由：锁的基本概念和实现
257	
8.1.1	理解线程安全
257	
8.1.2	对象头和锁
259	
8.2	避免残酷的竞争：锁在Java虚拟机中的实现和优化
260	
8.2.1	偏向锁
260	
8.2.2	轻量级锁
262	
8.2.3	锁膨胀
263	
8.2.4	自旋锁
264	
8.2.5	锁消除
264	
8.3	应对残酷的竞争：锁在应用层的优化思路
266	
8.3.1	减少锁持有时间
266	
8.3.2	减小锁粒度
267	
8.3.3	锁分离
269	
8.3.4	锁粗化
271	
8.4	无招胜有招：无锁
273	
8.4.1	理解CAS
273	
8.4.2	原子操作
274	
8.4.3	新宠儿LongAddr
277	
8.5	将随机变为可控：理解Java内存模型
280	
8.5.1	原子性
280	
8.5.2	有序性
282	

8.5.3 可见性	284
8.5.4 Happens-Before原则	286
8.6 小结	286
第9章 Class文件结构	287
9.1 不仅跨平台，还能跨语言：语言无关性	287
9.2 虚拟机的基石：Class文件	289
9.2.1 Class文件的标志——魔数	290
9.2.2 Class文件的版本	292
9.2.3 存放所有常数——常量池	293
9.2.4 Class的访问标记（Access Flag）	300
9.2.5 当前类、父类和接口	301
9.2.6 Class文件的字段	302
9.2.7 Class文件的方法基本结构	304
9.2.8 方法的执行主体——Code属性	306
9.2.9 记录行号——LineNumberTable属性	307
9.2.10 保存局部变量和参数——LocalVariableTable属性	308
9.2.11 加快字节码校验——StackMapTable属性	308
9.2.12 Code属性总结	313
9.2.13 抛出异常——Exceptions属性	314
9.2.14 用实例分析Class的方法结构	315
9.2.15 我来自哪里——SourceFile属性	318
9.2.16 强大的动态调用——BootstrapMethods属性	319
9.2.17 内部类——InnerClasses属性	320
9.2.18 将要废弃的通知——Deprecated属性	321
9.2.19 Class文件总结	

322	
9.3 操作字节码：走进ASM	
322	
9.3.1 ASM体系结构	
322	
9.3.2 ASM之Hello World	
324	
9.4 小结	
325	
第10章 Class装载系统	
326	
10.1 来去都有序：看懂Class文件的装载流程	
326	
10.1.1 类装载的条件	
327	
10.1.2 加载类	
330	
10.1.3 验证类	
332	
10.1.4 准备	
333	
10.1.5 解析类	
334	
10.1.6 初始化	
336	
10.2 一切Class从这里开始：掌握ClassLoader	
340	
10.2.1 认识ClassLoader，看懂类加载	
341	
10.2.2 ClassLoader的分类	
341	
10.2.3 ClassLoader的双亲委托模式	
343	
10.2.4 双亲委托模式的弊端	
347	
10.2.5 双亲委托模式的补充	
348	
10.2.6 突破双亲模式	
350	
10.2.7 热替换的实现	
353	
10.3 小结	
357	
第11章 字节码执行	
358	
11.1 代码如何执行：字节码执行案例	
359	
11.2 执行的基础：Java虚拟机常用指令介绍	
369	

11.2.1 常量入栈指令	369
11.2.2 局部变量压栈指令	370
11.2.3 出栈装入局部变量表指令	371
11.2.4 通用型操作	372
11.2.5 类型转换指令	373
11.2.6 运算指令	375
11.2.7 对象/数组操作指令	377
11.2.8 比较控制指令	379
11.2.9 函数调用与返回指令	386
11.2.10 同步控制	389
11.2.11 再看Class的方法结构	391
11.3 更上一层楼：再看ASM	393
11.3.1 为类增加安全控制	393
11.3.2 统计函数执行时间	396
11.4 谁说Java太刻板：Java Agent运行时修改类	399
11.4.1 使用-javaagent参数启动Java虚拟机	400
11.4.2 使用Java Agent为函数增加计时功能	402
11.4.3 动态重转换类	404
11.4.4 有关Java Agent的总结	407
11.5 与时俱进：动态函数调用	407
11.5.1 方法句柄使用实例	407
11.5.2 调用点使用实例	411
11.5.3 反射和方法句柄	412
11.5.4 指令invokedynamic使用实例	414
11.6 跑得再快点：静态编译优化	

418	
11.6.1	编译时计算
419	
11.6.2	变量字符串的连接
421	
11.6.3	基于常量的条件语句裁剪
422	
11.6.4	switch语句的优化
423	
11.7	提高虚拟机的执行效率：JIT及其相关参数
424	
11.7.1	开启JIT编译
425	
11.7.2	JIT编译阈值
426	
11.7.3	多级编译器
427	
11.7.4	OSR栈上替换
430	
11.7.5	方法内联
431	
11.7.6	设置代码缓存大小
432	
11.8	小结
436	

《实战Java虚拟机》

精彩短评

- 1、这个书比较容易入门 适合 jvm初学者，然后和<深入理解JVM>这本书结合 那就更好了，形成高低搭配。
- 2、这本书写得非常朴实，作者有很强的实践功底，每一部分内容除了有原理还有案例，非常适合入门学习
- 3、感觉还是不够深入，相对于前辈《深入理解Java虚拟机：JVM高级特性与最佳实践》，多了些JDK新知识，比如G1，实战部分如果更贴合实际生产需求会更好，而不是做实验性质的调优。
- 4、有所获.
- 5、前面几章非常不错，可以了解jvm的内存分配，后面就有些啰嗦点，值得一看。
- 6、前面讲有关jvm的一些概念，最后讲class结构和指令的有些偏题，和副标题不太符合
- 7、快速复习一下Java虚拟机，垃圾回收，堆，类加载...
- 8、整体上可能没有周志明的全和准确，但也有很多周没有讲到的。可以和深入理解jvm一起看。
- 9、前半部分值得一看
- 10、值得一读

章节试读

1、《实战Java虚拟机》的笔记-第58页

从发展趋势上看，未来64位系统必然会逐步取代32位系统，而64位系统中虚拟机更倾向于使用Server模式。

应该说 目前64位只支持server模式。

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:www.tushu000.com