

《Java性能优化权威指南》

图书基本信息

书名：《Java性能优化权威指南》

13位ISBN编号：9787115342973

10位ISBN编号：7115342970

出版时间：2014-3

出版社：人民邮电出版社

作者：Charlie Hunt,Binu John

页数：540

译者：柳飞,陆明刚

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.tushu111.com

《Java性能优化权威指南》

内容概要

Java性能优化圣经！Java之父重磅推荐！

本书由曾任职于Oracle/Sun的性能优化专家编写，系统而详细地讲解了性能优化的各个方面，帮助你学习Java虚拟机的基本原理、掌握一些监控Java程序性能的工具，从而快速找到程序中的性能瓶颈，并有效改善程序的运行性能。

Java性能优化的任何问题，都可以从本书中找到答案！

《Java性能优化权威指南》

作者简介

Charlie Hunt现任Salesforce公司的性能工程架构师。曾任Oracle公司首席JVM性能工程师，负责HotSpot Java虚拟机和Java SE类库性能的改进。Charlie拥有美国伊利诺伊理工大学的计算机科学硕士学位、爱荷华州立大学的计算机科学学士学位。

Binu John是世界上最大的社交网站创建平台Ning.com的高级性能工程师。他目前的职责是着力改善Ning平台的性能和扩展性，以支持每月数百万PV的访问量。Binu拥有美国爱荷华大学生物医学工程和计算机科学硕士学位。

书籍目录

第1章 策略、方法和方法论	1
1.1 性能问题的现状	1
1.2 性能分析的两种方法：自顶向下和自底向上	4
1.2.1 自顶向下	4
1.2.2 自底向上	5
1.3 选择正确的平台并评估系统性能	5
1.3.1 选择正确的CPU架构	6
1.3.2 评估系统性能	7
1.4 参考资料	7
第2章 操作系统性能监控	8
2.1 定义	8
2.2 CPU使用率	9
2.2.1 监控CPU使用率：Windows	9
2.2.2 监控CPU使用率：Windows typeperf	12
2.2.3 监控CPU使用率：Linux	13
2.2.4 监控CPU使用率：Solaris	14
2.2.5 命令行监控CPU使用率：Linux和Solaris	16
2.3 CPU调度程序运行队列	19
2.3.1 监控CPU调度程序运行队列：Windows	19
2.3.2 监控CPU调度程序运行队列：Solaris	21
2.3.3 监控CPU调度程序运行队列：Linux	21
2.4 内存使用率	22
2.4.1 监控内存利用率：Windows	22
2.4.2 监控内存使用率：Solaris	23
2.4.3 监控内存使用率：Linux	24
2.4.4 监控锁竞争：Solaris	25
2.4.5 监控锁竞争：Linux	26
2.4.6 监控锁竞争：Windows	27
2.4.7 隔离竞争锁	27
2.4.8 监控抢占式上下文切换	27
2.4.9 监控线程迁移	28
2.5 网络I/O使用率	28
2.5.1 监控网络I/O使用率：Solaris	29
2.5.2 监控网络I/O使用率：Linux	30
2.5.3 监控网络I/O使用率：Windows	30
2.5.4 应用性能改进的考虑	31
2.6 磁盘I/O使用率	31
2.7 其他命令行工具	34
2.8 监控CPU使用率：SPARC T系列系统	35
2.9 参考资料	36
第3章 JVM概览	38
3.1 HotSpot VM的基本架构	38
3.2 HotSpot VM运行时	40
3.2.1 命令行选项	40
3.2.2 VM生命周期	41
3.2.3 VM类加载	44
3.2.4 字节码验证	46

3.2.5	类数据共享	47
3.2.6	解释器	48
3.2.7	异常处理	49
3.2.8	同步	50
3.2.9	线程管理	51
3.2.10	C++堆管理	53
3.2.11	Java本地接口	54
3.2.12	VM致命错误处理	55
3.3	HotSpot VM垃圾收集器	56
3.3.1	分代垃圾收集	56
3.3.2	新生代	58
3.3.3	快速内存分配	60
3.3.4	垃圾收集器	60
3.3.5	Serial收集器	61
3.3.6	Parallel收集器：吞吐量为先！	62
3.3.7	Mostly-Concurrent收集器：低延迟为先！	62
3.3.8	Garbage-First收集器：CMS替代者	64
3.3.9	垃圾收集器比较	64
3.3.10	应用程序对垃圾收集器的影响	65
3.3.11	简单回顾收集器历史	65
3.4	HotSpot VM JIT编译器	65
3.4.1	类型继承关系分析	67
3.4.2	编译策略	67
3.4.3	逆优化	68
3.4.4	Client JIT编译器概览	69
3.4.5	Server JIT编译器概览	69
3.4.6	静态单赋值——程序依赖图	69
3.4.7	未来增强展望	71
3.5	HotSpot VM自适应调优	71
3.5.1	Java 1.4.2的默认值	71
3.5.2	Java 5自动优化的默认值	71
3.5.3	Java 6 Update 18更新后的默认优化值	73
3.5.4	自适应Java堆调整	74
3.5.5	超越自动优化	75
3.6	参考资料	75
第4章	JVM性能监控	77
4.1	定义	77
4.2	垃圾收集	78
4.2.1	重要的垃圾收集数据	78
4.2.2	垃圾收集报告	78
4.2.3	垃圾收集数据的离线分析	86
4.2.4	图形化工具	89
4.3	JIT编译器	103
4.4	类加载	104
4.5	Java应用监控	106
4.6	参考资料	109
第5章	Java应用性能分析	110
5.1	术语	111
5.1.1	通用性能分析术语	111

5.1.2	Oracle Solaris Studio Performance Analyzer术语	112
5.1.3	NetBeans Profiler术语	112
5.2	Oracle Solaris Studio Performance Analyzer	112
5.2.1	支持平台	113
5.2.2	下载/安装Oracle Solaris Studio Performance Analyzer	114
5.2.3	使用Oracle Solaris Studio Performance Analyzer 抓取性能数据	114
5.2.4	查看性能数据	118
5.2.5	数据表示	125
5.2.6	过滤性能数据	128
5.2.7	命令行工具er_print	129
5.3	NetBeans Profiler	135
5.3.1	支持平台	136
5.3.2	下载安装NetBeans Profiler	136
5.3.3	开始方法分析会话	137
5.3.4	Controls子面板	143
5.3.5	Status子面板	143
5.3.6	Profiling Results子面板	143
5.3.7	Saved Snapshots子面板	144
5.3.8	View子面板	144
5.3.9	Basic Telemetry子面板	144
5.3.10	查看动态结果	145
5.3.11	对结果进行快照	145
5.3.12	启动内存分析会话	146
5.3.13	查看实时结果	148
5.3.14	对结果进行快照	150
5.3.15	定位内存泄漏	150
5.3.16	分析堆转储	151
5.4	参考资料	152
第6章	Java应用性能分析技巧	153
6.1	性能优化机会	153
6.2	系统或内核态CPU使用	154
6.3	锁竞争	161
6.4	Volatile的使用	171
6.5	调整数据结构的大小	172
6.5.1	StringBuilder或StringBuffer大小的调整	172
6.5.2	Java Collection类大小调整	175
6.6	增加并行性	179
6.7	过高的CPU使用率	181
6.8	其他有用的分析提示	182
6.9	参考资料	184
第7章	JVM性能调优入门	185
7.1	方法	185
7.1.1	假设条件	187
7.1.2	测试基础设施需求	188
7.2	应用程序的系统需求	188
7.2.1	可用性	188
7.2.2	可管理性	188
7.2.3	吞吐量	189
7.2.4	延迟及响应性	189

7.2.5	内存占用	189
7.2.6	启动时间	189
7.3	对系统需求分级	190
7.4	选择JVM部署模式	190
7.4.1	单JVM部署模式	190
7.4.2	多JVM部署模式	190
7.4.3	通用建议	191
7.5	选择JVM运行模式	191
7.5.1	Client模式或Server模式	191
7.5.2	32位/64位 JVM	192
7.5.3	垃圾收集器	192
7.6	垃圾收集调优基础	193
7.6.1	性能属性	193
7.6.2	原则	193
7.6.3	命令行选项及GC日志	194
7.7	确定内存占用	197
7.7.1	约束	197
7.7.2	HotSpot VM堆的布局	197
7.7.3	堆大小调优着眼点	200
7.7.4	计算活跃数据大小	201
7.7.5	初始堆空间大小配置	202
7.7.6	其他考量因素	203
7.8	调优延迟/响应性	204
7.8.1	输入	205
7.8.2	优化新生代的大小	205
7.8.3	优化老年代的大小	207
7.8.4	为CMS调优延迟	210
7.8.5	Survivor空间介绍	212
7.8.6	解析晋升阈值	214
7.8.7	监控晋升阈值	215
7.8.8	调整Survivor空间的容量	216
7.8.9	显式的垃圾收集	222
7.8.10	并发永久代垃圾收集	223
7.8.11	调优CMS停顿时间	224
7.8.12	下一步	225
7.9	应用程序吞吐量调优	225
7.9.1	CMS吞吐量调优	225
7.9.2	Throughput收集器调优	226
7.9.3	Survivor空间调优	228
7.9.4	调优并行垃圾收集线程	231
7.9.5	在NUMA系统上部署	231
7.9.6	下一步	232
7.10	极端示例	232
7.11	其他性能命令行选项	232
7.11.1	实验性(最近最大)优化	232
7.11.2	逃逸分析	233
7.11.3	偏向锁	233
7.11.4	大页面支持	234
7.12	参考资料	236

第8章	Java应用的基准测试	237
8.1	基准测试所面临的挑战	237
8.1.1	基准测试的预热阶段	238
8.1.2	垃圾收集	240
8.1.3	使用Java Time接口	240
8.1.4	剔除无效代码	241
8.1.5	内联	247
8.1.6	逆优化	251
8.1.7	创建微基准测试的注意事项	256
8.2	实验设计	257
8.3	使用统计方法	258
8.3.1	计算均值	258
8.3.2	计算标准差	258
8.3.3	计算置信区间	259
8.3.4	使用假设测试	260
8.3.5	使用统计方法的注意事项	262
8.4	参考文献	263
8.5	参考资料	263
第9章	多层应用的基准测试	264
9.1	基准测试难题	264
9.2	企业级应用基准测试的考量	266
9.2.1	定义被测系统	266
9.2.2	制定微基准测试	266
9.2.3	定义用户交互模型	267
9.2.4	定义性能指标	270
9.2.5	扩展基准测试	273
9.2.6	用利特尔法则验证	274
9.2.7	思考时间	275
9.2.8	扩展性分析	278
9.2.9	运行基准测试	278
9.3	应用服务器监控	281
9.3.1	GlassFish监控	281
9.3.2	监控子系统	286
9.3.3	Solaris	287
9.3.4	Linux	288
9.3.5	Windows	288
9.3.6	外部系统的性能	289
9.3.7	磁盘I/O	292
9.3.8	监控和调优资源池	293
9.4	企业级应用性能分析	294
9.5	参考资料	295
第10章	Web应用的性能调优	297
10.1	Web应用的基准测试	298
10.2	Web容器的组件	298
10.2.1	HTTP连接器	299
10.2.2	Servlet引擎	300
10.3	Web容器的监控和性能调优	300
10.3.1	容器的开发和生产模式	300
10.3.2	安全管理器	301

10.3.3	JVM调优	301	
10.3.4	HTTP服务和Web容器	303	
10.3.5	HTTP监听器	303	
10.4	最佳实践	315	
10.4.1	Servlet和JSP最佳实践	315	
10.4.2	内容缓存	324	
10.4.3	会话持久化	328	
10.4.4	HTTP服务器文件缓存	329	
10.5	参考资料	333	
第11章	Web Service的性能	334	
11.1	XML的性能	334	
11.1.1	XML处理的生命周期	335	
11.1.2	解析/解编组	335	
11.1.3	访问	338	
11.1.4	修改	338	
11.1.5	序列化/编组	339	
11.2	验证	339	
11.3	解析外部实体	341	
11.4	XML文档的局部处理	343	
11.5	选择合适的API	346	
11.6	JAX-WS参考实现栈	349	
11.7	Web Service基准测试	350	
11.8	影响Web Service性能的因素	353	
11.8.1	消息大小的影响	353	
11.8.2	不同Schema类型的性能特征	355	
11.8.3	终端服务器的实现	358	
11.8.4	处理程序的性能	359	
11.9	最佳性能实践	361	
11.9.1	二进制负载的处理	361	
11.9.2	处理XML文档	365	
11.9.3	使用MTOM发送XML文档	365	
11.9.4	使用Provider接口	368	
11.9.5	快速信息集	370	
11.9.6	HTTP压缩	372	
11.9.7	Web Service客户端的性能	373	
11.10	参考资料	374	
第12章	Java持久化及Enterprise Java Bean的性能	375	
12.1	EJB编程模型	376	
12.2	Java持久化API及其参考实现	376	
12.3	监控及调优EJB容器	379	
12.3.1	线程池	380	
12.3.2	Bean池和缓存	382	
12.3.3	EclipseLink会话缓存	385	
12.4	事务隔离级	386	
12.5	Enterprise Java Bean的最佳实践	387	
12.5.1	简要说明使用的EJB基准测试	387	
12.5.2	EJB 2.1	388	
12.5.3	EJB 3.0	400	
12.6	Java持久化最佳实践	403	

12.6.1	JPA查询语言中的查询	403
12.6.2	查询结果缓存	405
12.6.3	FetchType	406
12.6.4	连接池	408
12.6.5	批量更新	409
12.6.6	选择正确的数据库锁策略	411
12.6.7	不带事务的读取	411
12.6.8	继承	411
12.7	参考资料	412
附录A	重要的HotSpot VM选项	413
附录B	性能分析技巧示例源代码	429
B.1	锁竞争实现1	429
B.2	锁竞争实现2	439
B.3	锁竞争实现3	449
B.4	锁竞争实现4	459
B.5	锁竞争实现5	469
B.6	调整容量变化1	481
B.7	调整容量变化2	492
B.8	增加并发性的单线程实现	504
B.9	增加并发性的多线程实现	514

精彩短评

- 1、书是好书，但是年代太久了，早两年出版就好了
- 2、总体很好的一本JVM的专业书籍！花了零零碎碎的时间看完了，有一小部分章节跳过了。目前记住的倒不多，但是已经不觉得这块儿像之前那么神秘了，对于堆空间大小比例(新生代、老生代以及新生代中Eden和Survivor)的分配和优化，讲述得很深刻，针对不同的场景提出不同的优化策略，覆盖面很广，另外还有举例描述，很容易理解记忆这些知识点，而不需要强行记忆。对各种GC日志的解读也很清晰明了，后边还有对web service的性能解析和优化。不满意的是这本书的最后接近100页全是之前例子中的Java代码，其实可以用一个链接到类似github这样的代码仓库就足够了，不过依然还是感谢译者这么辛苦。总体而言，这本书值得推荐，我个人还会再看看的。
- 3、针对web比较有用。
- 4、重点读的
- 5、前几张可以多看看几遍..后面有些过时了，有些工具介绍不常用也可以跳过比如 Oracle 的性能监控工具..
- 6、前十章不错，受益匪浅，JVM调优终于有套路了，后边章节感觉用处不太大
- 7、一口气看完《java性能优化权威指南》、《java性能权威指南》和《深入理解java虚拟机》，强烈推荐《java性能优化权威指南》，后面两本也很不错，不过相对于第一本，第二本在讲解jvm的同时会讲一些API实践，第三本太深入jvm，对于普通程序员来说，Java性能优化权威指南是首选，作者不光专业知识深厚，写着能力还特别强，循序渐进，而且具有很强的可操作性。
- 8、内容有点老了，另外无用的章节有点多，以及大段的贴代码。瑕不掩瑜，Java程序员还是应该好好看一下
- 9、写的不错就是有点跟不上jdk的更新速度了。毕竟很多参数都是和不同版本绑定了。不过还是很不错的书籍！
- 10、great book
- 11、读完前7章，有一定价值，不过略显啰嗦。后面的章节不建议阅读。。。。
- 12、搞Java的人都应该读一读，译者非常用心，谢谢
- 13、java的圣经级别的书籍
- 14、有感悟，有所得。但其中部分技巧以不能满足生产环境的需要
- 15、常翻常新，6，7两章可读
- 16、java调优，讲了比较多jvm方面的内容，收获挺大
- 17、很多细节，还是不错的。但是很多东西感觉已经过时了。
- 18、GC调优
- 19、性能优化的标准教材
- 20、并发编程实战，性能优化权威指南，java编程思想，基本可以搞定java了。
- 21、感受到了译者的用心，性能优化是个大话题，一是要讲明白深层次的原理，二是要结合工具和实践，尽管这本书这么厚，还是很难全面的去兼顾这两方面。另外内容略偏JavaEE和Oracle/SUN的技术栈了。
- 22、直接跳到第六章，比较爽，以后写程序得先看看类库实现了。
- 23、前面很精彩，后面技术基本没有用过，被忽略了
- 24、读的时候感觉有蛮大的收获，但是读完之后，发现什么也记不得了。说明作者对观点的整理和灌输还是不足够，或者是自己还是太弱。
- 25、感觉不出来什么
- 26、书倒是很不错，只是有些啰嗦，缺少抽象性的总结，比较偏向实践。书的内容也有点老了，不过优化思路都是一致的。比《深入理解Java虚拟机》更详尽。其实基础扎实，读完4，6，7章就够了。
- 27、非常好的一本书，从理论到操作都非常详细，叙述的很透彻。这本书的翻译个人也觉得很不错。
- 28、介绍的很详细，细读地话需要很多知识的积累，作为工具书留着细细品味
- 29、看完也没有很深印象，主要是平常用的太少了了吧。也算了解了思路，值得一看的书。
- 30、介绍了性能分析技巧与工具,性能调优方法,内存管理等多个角度切入.详细介绍了java性能调优的心得和方法论.应作为工具书随时查阅.

- 31、有点厚，但内容没有觉得很权威哈
- 32、略读，看不大明白，估计是还没到相应的阶段。

精彩书评

- 1、这本书介绍了Java程序员需要掌握的OS、JVM层面的很多知识，系统详尽地介绍了性能优化的各个方面。并且作者结合自己的经验总结了一些性能优化的模式，例如CPU利用率高的时候需要怎么进行优化等，这些模式让人受益匪浅，相信读过后能对大家日常工作中做性能优化提供不小的帮助。
- 2、内容很精彩，忍不住晚上熬夜看完了。对于系统监控介绍了不少好工具，还有详细的使用方法。JVM介绍内容也不错，很好的快速了解gc概况。因为本书不是专门讲JVM的书，所以这块内容比较概括，深入的细节建议去关注Oracle官方的博客。JVM监控也介绍了不少好工具，忍不住下载尝试一把，非常不错。MAT,JPprofiler和BTrace是我平时用的比较多的工具，而本书并没有介绍这两个工具，而是从另一个视角介绍了oracle solaris studio和NetBeans Profiler，正好作为一个不错的补充。对于JVM的调优，从理论结合实践，从问题入手，步步优化，最终给出优化结果。其中的方法值得好好学习。如何发现问题，如何入手，怎么进行下一步，够给出了不错的指导。后面关于web调优主要讲GlassFish，平时基本不涉及到，后面EJB和持久化也用的比较少。所以这两章略读。总的来说还是不错的，推荐。
- 3、前面8章还不错，第9章之后一般般...第9章之后的内容大概浏览了一下，基本上都是泛泛的说了一下，不足以指导实践，还要去参考其他书籍...个人感觉，在实践中很少用到GlassFish、WebService、EJB这些东东...所以9章之后的内容，基本没有看的必要了...
- 4、"他目前的职责是着力改善Ning平台的性能和扩展性，以支持每月数百万PV的访问量。"摊下来一天一二十万pv, uv可能只有几千到几万了,就这点访问量还需要性能优化么!抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了抱歉，你的评论太短了
- 5、如题。各大网站都已经售了，已经冲到畅销榜前茅。欢迎读者到这里提交勘误和意见：<http://www.ituring.com.cn/book/852?q=Java%E6%80%A7%E8%83%BD>

章节试读

1、《Java性能优化权威指南》的笔记-第41页

3.2.1 命令行选项

java的命令行分为三种：标准的（写在jvm标准中得。可惜没有找到），稳定的以 -X开头，不稳定的以 -XX开头。

参数设定的方法：

1. 打开布尔值 -XX:+<option>; 关闭布尔值 -XX:-<option>;
2. 设定数值型。 -XX:<option>=<number>; 数值可以包括 k,K,m,M,g,G
3. 字符串型的数值方案 -XX:<option>=<string>;

在命令行中通过命令查看正在运行的java的参数的方法：

1. jinfo -flags pid
2. jinfo -sysprops pid
3. jps -lv

通过代码来操作mbean就可以动态在运行期获取到这些参数

本章延伸阅读：

<http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>

http://publib.boulder.ibm.com/infocenter/realtime/v1r0/index.jsp?topic=%2Fcom.ibm.rt.doc.10%2Fuser%2Fstandard_options.html

<http://stackoverflow.com/questions/8343099/is-there-a-java-api-to-get-the-vm-arguments>

<http://stackoverflow.com/questions/1490869/how-to-get-vm-arguments-from-inside-of-java-application>

2、《Java性能优化权威指南》的笔记-第42页

（3）设定环境变量如：LD_LIBRARY_PATH和CLASSPATH

这里的LD_LIBRARY_PATH是什么？

3、《Java性能优化权威指南》的笔记-第61页

第一行，脚注放错了位置，脚注2用来说明新生代采用复制的方式实现垃圾收集，但是放在了老年代后面，不合适。

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.tushu111.com