TURING 图灵程序设计丛书

[PACKT] PUBLISHING

Oracle JRockit
The Definitive Guide

JRockit权威指南
深入理解JVM

[瑞士] 马库斯·希尔特 [瑞典] 马库斯·拉杰格伦 著
曹旭东 译

- 深入JVM内部，剖析Java虚拟机原理，阐明Java性能提升关键
- 莫枢（RednaxelaFx）、阿里中间件团队推荐JVM参考书

中国工信出版集团 人民邮电出版社 POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Oracle JRockit
The Definitive Guide

JRockit权威指南
深入理解JVM

[瑞士] 马库斯·希尔特 [瑞典] 马库斯·拉杰格伦 著
曹旭东 译

人民邮电出版社
北京

图书在版编目（CIP）数据

JRockit权威指南 : 深入理解JVM / (瑞士) 马库斯·希尔特 (Marcus Hirt), (瑞典) 马库斯·拉杰格伦 (Marcus Lagergren) 著 ; 曹旭东译. -- 北京 : 人民邮电出版社, 2019.1
(图灵程序设计丛书)
ISBN 978-7-115-50045-8

Ⅰ. ①J… Ⅱ. ①马… ②马… ③曹… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第262524号

内 容 提 要

本书以 JRockit 为例深入剖析 JVM 工作原理，分为 3 大部分。第一部分着重介绍了 JVM 和自适应运行时的工作原理，并以 JRockit 为例专门介绍到底什么是好的 Java 代码。第二部分介绍 JRockit Mission Control 套件的具体功能，以及如何使用 JRockit Mission Control 套件来查找应用程序的性能瓶颈。第三部分介绍 Java 发展方向。

本书适合所有以 Java 编程语言为工作中心的开发人员和系统管理员。

◆ 著 [瑞士] 马库斯·希尔特 [瑞典]马库斯·拉杰格伦
译 曹旭东
责任编辑 朱 巍
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市君旺印务有限公司印刷

◆ 开本：800×1000 1/16
印张：21
字数：496千字 2019年1月第1版
印数：1－2 000册 2019年1月河北第1次印刷

著作权合同登记号 图字：01-2018-0945号

定价：99.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

版权声明

献给我的家人：Malin、Alexander和小Natalie。他们容忍我将无数个夜晚和周末花在发布新的主版本和撰写本书上。

——Marcus Hirt

献给我的家人：Klara、Alice和Ylva，特别是我可爱的妻子Klara。Klara要一个人应付两个孩子，非常辛苦。她不止一次表示过要买一本，然后烧了它。

——Marcus Lagergren

序

我至今仍然清楚地记得第一次遇到 JRockit 团队时的情形。那是在 1999 年，我代表 WebLogic 参加 JavaOne 大会，这些身着黑色 T 恤衫的瑞典大学生正在向大家介绍他们开发的、号称性能最强劲的服务器端虚拟机。那时，HotSpot 1.2 版本的发布再次延误，而我们也正被 Classic VM 中无穷无尽的伸缩性问题搞得焦头烂额，所以就对这些小伙子的演讲产生了兴趣，驻足倾听。在我离开他们的展台时，我认为这些聪明的小伙子还有些稚嫩，他们小看了虚拟机开发的复杂性。

时光飞逝，BEA 收购了 JRockit，而我成了 WebLogic 和 JRockit 两个团队间的技术沟通人。此时的 JRockit 已非吴下阿蒙，在服务器端表现出了强大的伸缩性和强劲的性能。随着工作的展开，我也很荣幸结识了本书的两位作者：Marcus Lagergren 和 Marcus Hirt。

当时负责编译器开发的 Lagergren 是一位非常高产的程序员。曾经有一段时间，我和他共同研究如何优化 WebLogic，以及探究为何某个方法没有被内联或去虚拟化。在这个过程中，我们（包括 WebLogic 团队和 JRockit 团队的其他成员）合力创造了 SPECjAppServer 的几项世界纪录，使得 JRockit 威名更盛。

本书的另一位作者 Hirt 则始终专注于性能剖析和诊断方面的工作。因此，他也顺理成章地成为了相关工具开发的领导者，这些工具就是 JRockit Mission Control 的前身。我们很早就注意到，若想扩大 JRockit 工程团队的规模，就必须投入资源来开发更好用的工具，以简化开发和调试工作。

转眼又过了几年，我加入了 Oracle，BEA 也被 Oracle 收购了。我怀着激动的心情再次迎接 JRockit 团队的加入，只不过这次的新东家换成了 Oracle。JRockit 的核心开发团队仍然是那些人，他们现在已经是虚拟机领域的专家了。

Lagergren 仍然负责底层实现（即 JRockit Virtual Edition）的开发，生产力依然很高。在 Hirt 的带领下，Mission Control 已经从内部开发工具成长为最受用户喜爱的 JRockit 开发套件之一。两位作者对 JRockit 的方方面面都非常了解，很难想象会有比他们两位更适合撰写本书的人了。

因此，正如前面所说，能够结识 JRockit 开发团队并与其合作，我感到非常高兴。我相信，你会享受阅读本书的过程。多年以来，我觉得本书主题非常有趣，希望你也会有这样的体会。

Adam Messinger
Oracle Fusion Middelware 项目组，开发副总裁
2010 年 2 月 14 日
加州旧金山

前　言

机缘巧合促成了本书的出版。

那时，互联网还没有在世界范围内普及，我们也还只是高中生，经常混迹于同一个 BBS，在讨论数学问题的过程中结识了对方，成为了好友，并将这份友情延伸到了生活和合作的软件项目中。后来，我们又共同进入了位于斯德哥尔摩的瑞典皇家理工学院（KTH）学习。

在 KTH，我们结识了更多的朋友。在第三学年的数据库系统课程中，我们找到了足够多志同道合的人，准备干点事业。最终，我们决定成立一家名为 Appeal Software Solutions（其首字母缩写为 A.S.S.，当时看来绝对是一个完美的名字）的咨询公司。我们中有些人是半工半读的，所以预留了部分收入，以便当所有成员毕业后可以使公司步入正轨。我们的长期目标是公司可以开发产品，而不仅仅是做咨询，但当时我们还不知道到底要开发什么。

1997 年，由于在 Sun 公司赞助的大学生竞赛中胜出，Joakim Dahlstedt、Fredrik Stridsman 和 Mattias Joëlson 得以参加当年的 JavaOne 大会。有意思的是，第二年，他们又胜出了。

一切都源于我们的 3 位英雄在 1997 年和 1998 年参加的两届 JavaOne 大会。在会上，他们注意到，Sun 公司的自适应 JVM——HotSpot 虽然在当时被誉为能够彻底解决 Java 性能问题的终极 JVM，但在这两年里却没有什么实质性的进步。那时的 Java 主要是解释执行的，市场上有一些针对 Java 的静态编译器，可以生成运行速度快于字节码的静态代码，但是这从根本上违反了 Java 的语义。正如本书反复强调的，到目前为止，自适应解决方案在运行时具有远超静态解决方案的潜力，但实现起来也更困难。

1998 年，HotSpot 没什么动作，年轻气盛的我们不禁问道："这很难吗？看我们做一个更好、更快的自适应虚拟机出来！"我们专业背景不错，而且认为有了明确的方向，于是就开工了。尽管后来的实践证明了挑战比我们预期的更大，但我们想提醒读者的是，当时是 1998 年，Java 在服务器端的腾飞才刚刚开始，J2EE 刚刚出现，几乎没人听说过 JSP。因此，我们所涉及的问题领域小得多。

我们最初计划用一年时间实现一个 JVM 的预览版，同时继续提供咨询服务来保证 JVM 的持续开发。最初，新 JVM 的名字是 RockIT，结合了 Rock and Roll（摇滚）、Rock Solid（坚如磐石）和 IT 三者的意思。后来由于注册商标的原因，又在名字前面加了一个字母 J。

在经历了初期的几次失败后，我们需要寻找风投。当然，向投资人解释清楚为什么投资一款自适应 JVM 能够赚钱（同时期的其他竞争对手都是免费提供的），是一大难题。这不仅仅因为当时是 1998 年，更重要的因素是，投资人还无法理解这种既不需要给用户发广告短信，也不需要

发送电子邮件订单的商业模式。

最终，我们获得了风投，并在2000年初发布了JRockit 1.0版本的第一个原型。尽管只是1.0版本（网上有人说它“非常1.0”，不够成熟），但是它应用于多线程服务器程序时性能优异，风光一时。以此为契机，我们获得了更多的投资，并将咨询业务拆分为一个独立的分公司，公司的名字也从Appeal Software Solutions变成了Appeal Virtual Machines。我们又雇用了一些销售人员，并就Java许可证的问题开始与Sun公司协商。

JRockit的相关工作越来越多。2001年，处理咨询业务的工程师都转入了与JVM相关的项目中，咨询公司宣告停业。这时，我们清楚地知道如何将JRockit的性能再提升一步，同时也意识到在这个过程中我们消耗资源的速度太快了。于是，管理层开始寻找合适的大公司，以实现整体收购。

2002年2月，BEA公司收购Appeal Virtual Machines公司，这让投资人松了一口气，同时也保证了我们有足够的资源做进一步的研究和开发。为了配合测试，BEA建立了一个宽敞的服务器机房，加固了地板，保证了电力供应。那时，有一根电缆从街上的接线盒通过服务器机房的窗户连进来。过了一段时间，这个服务器机房已经无法放下开发测试所需的全部服务器了，于是我们又租了一个机房来放置服务器。

作为BEA平台的一部分，JRockit的发展相当理想。在BEA的前两年，我们为JRockit开发了很多区别于其他Java解决方案的新特性，例如后来发展成为JRockit Mission Control的开发框架。此后，新闻发布、世界级的测试跑分和虚拟化平台随之而来。在拥有了JRockit后，BEA与Sun、IBM并列为三大JVM厂商，成为了拥有数千用户的平台。JRockit产生的利润，首先是来自工具套件，然后是产品JRockit Real Time提供的无比强大的GC性能。

2008年，Oracle收购BEA，这一事件起初令人感到不安，但是JRockit和相关团队最终获得了更多的关注和赞誉。

经过这些年的发展，令我们引以为荣的是，JRockit的用户遍布全球，它为关键应用的稳定运行保驾护航。同样令我们感到骄傲的是，当初6个少年在斯德哥尔摩老城区的一个小破屋中的设计已经成长为世界级产品。

本书的内容是我们十多年来与自适应运行时，尤其是JRockit，打交道的经验总结。据我们所知，其中的很多内容之前还没有发表过。

希望本书能对你有所帮助和启发。

内容概述

第1章：起步。这一章对JRockit JVM和JRockit Mission Control做了简要介绍，内容包括如何获得相关软件及软件对各平台的支持情况，在切换JVM厂商的产品时需要注意的问题，JRockit和JRockit Mission Control版本号的命名规则，以及如何获取更多有关JRockit JVM的内容。

第2章：自适应代码生成。这一章对自适应运行时环境中的代码生成做了简要介绍。具体说来，解释了为什么在JVM中实现自适应代码生成比在静态环境中更有难度，而其实现所能发挥

的效用却更加强大；介绍了赌博式的性能优化技术；通过一个例子介绍了 JRockit 的代码生成和优化流水线；讨论了自适应代码优化和传统代码优化；介绍了如何使用标志和指令文件来控制 JRockit 的代码生成。

第 3 章：自适应内存管理。这一章对自适应运行时环境中的内存管理做了介绍。通过介绍自动内存管理的相关概念和算法，解释了垃圾回收器的工作机制。详细介绍了 JVM 在为对象分配内存时所做的具体工作，以及为便于执行垃圾回收所需记录的元数据信息。后半部分主要介绍用于控制内存管理的最重要的 Java API，以及可在 Java 应用程序中生成确定性延迟的 JRockit Real Time 产品。最后，介绍了如何使用标志来控制 JRockit JVM 的内存管理系统。

第 4 章：线程与同步。这一章介绍了 Java 和 JVM 中非常重要的线程与同步的概念及其在 JVM 中的简要实现，并深入讨论了 Java 内存模型及其内在的复杂性。简单介绍了基于运行时信息反馈的自适应优化对线程和同步机制的实现的影响。此外，还以双检查锁失效为例对多线程编程中常见的一些错误做了介绍。最后讲解了如何分析 JRockit 中的锁，以及如何通过标志控制线程的部分行为。

第 5 章：基准测试与性能调优。这一章讨论了基准测试的相关性，以及制定性能目标和指标的重要性；阐释了如何针对特定问题设计适合的基准测试；介绍了一些针对 Java 的工业级基准测试套件；详细讨论了如何根据基准测试的结果优化应用程序和 JVM；以 JRockit JVM 为介绍了相关命令行参数的使用。

第 6 章：JRockit Mission Control 套件。这一章介绍了 JRockit Mission Control 工具套件，包括启动和各种详细配置等内容。解释了如何在 Eclipse 中运行 JRockit Mission Control，以及如何配置 JRockit 以使 Eclipse 在 JRockit 上运行。介绍了几种不同的工具，统一了相关术语的使用。讲解了如何使用 JRockit Mission Control 远程访问 JRockit JVM，以及与故障处理相关的内容。

第 7 章：Management Console。这一章介绍了 JRockit Mission Control 中的 Management Console 组件，讲解了诊断命令的概念以及如何在线监控 JVM 实例，还介绍了触发器规则的设置和事件通知的机制，最后讲解了如何利用自定义组件扩展 Management Console。

第 8 章：JRockit Runtime Analyzer。这一章介绍了 JRockit 运行时分析器（JRockit Runtime Analyzer，JRA），它是一款可以定制的按需分析框架，用于详细记录 JRockit 以及运行在其中的应用程序的执行状况，以便进行离线分析。其记录内容包括方法和锁的性能分析、垃圾回收信息、优化决策信息、对象统计信息以及延迟事件等。这一章最后介绍了如何根据这些记录信息来判别常见问题以及如何延迟分析。

第 9 章：JRockit Flight Recorder。这一章详细介绍了 JFR（JRockit Flight Recorder）。新版本 JRockit Mission Control 套件使用 JFR 取代了 JRA。这一章讲解了 JFR 与 JRA 的区别，最后介绍了如何扩展 JFR。

第 10 章：Memory Leak Detector。这一章介绍了 JRockit Mission Control 套件中的最后一个工具 JRockit Memory Leak Detector。其中介绍了具有垃圾回收功能的编程语言中内存泄漏的概念，以及 Memory Leak Detector 的一些用例。Memory Leak Detector 不仅可以用来找出 Java 应用程序中意外持有的对象，还可以对 Java 堆做通用分析。此外还介绍了 Memory Leak Detector 的一些内部实现机制，以及它能保持很低的运行开销的原因。

第 11 章：JRCMD。这一章介绍了命令行工具 JRCMD。用户可以通过 JRCMD 与目标机器上的 JVM 交互，并发送诊断命令。这一章按字母表顺序列出了 JRCMD 中最重要的诊断命令，并通过示例讲解了如何使用这些命令来检测或修改 JRockit JVM 的状态。

第 12 章：JRockit Management API。这一章介绍了如何编程实现对 JRockit JVM 内部功能的访问，如 JRockit Mission Control 套件就是基于 Management API 来实现的。尽管这一章介绍的 JMAPI 和 JMXMAPI 并未得到完整的官方支持，但从中可以了解到一些 JVM 的工作机制。希望读者可以实际动手操作一下，以加深理解。

第 13 章：JRockit Virtual Edition。这一章介绍了现代云环境中的虚拟化，其中包括了 JRockit Virtual Edition 产品的相关概念和具体细节。通常来说，操作系统很重要，但对于 JRockit Virtual Edition 来说，移除软件栈中的操作系统层并不是什么大问题，而且移除之后还可以降低操作系统层所带来的性能开销，降低的程度甚至在物理硬件上也达不到。

阅读前提

请正确安装 JRockit JVM 和运行时环境。为了更好地理解本书的内容，请使用 JRockit R28 或其之后的版本，不过使用 JRockit R27 也是可以的。此外，正确安装 Eclipse for RCP/Plug-in Developer 也很有必要，尤其是尝试用不同的方法扩展 JRockit Mission Control 以及使用源码包中的程序时。

目标读者

本书主要面向以 Java 为工作中心，并已具备一定知识技能的人员，例如对 Java 开发或安装管理有相关工作经验的开发人员或系统管理员。书中内容分为 3 大部分。

第一部分着重介绍了 JVM 和自适应运行时的作用及工作原理，还指出了自适应运行时以及 JRockit 的优势和劣势，以便在适当的时候解释什么是良好的 Java 编码实践。深入到 JVM 这个黑盒中，探查运行 Java 应用程序时到底发生了什么。理解第一部分的内容可以帮助开发人员和架构师理解某些设计决策的后果，进而做出更好的决策。这部分也可作为高校自适应运行时课程的学习资料。

第二部分着重介绍了 JRockit Mission Control 套件的具体功能，以及如何使用它来查找应用程序的性能瓶颈。对于想要对 JRockit 系统做性能调优以运行特定程序的系统管理员和开发人员来说，这部分内容非常有用。对于希望优化 Java 应用程序以提高资源利用率、优化性能的开发人员来说，这部分内容也很有用。但应该记住的是，对 JVM 层面的调优也只有这么多了，对应用程序本身的业务逻辑和具体实现做调优其实是更简单、更有效的。本书将会介绍如何使用 JRockit Mission Control 套件来查找应用程序的瓶颈，以及如何控制硬件和程序运行的成本。

第三部分介绍了新近和即将发布的重要的 JRockit 相关技术，主要面向对 Java 技术发展方向比较感兴趣的读者。这部分内容着重讲解了 Java 虚拟化。

最后，列出了本书的参考文献和术语表。

排版约定

本书中会包含一些代码，包括 Java 代码、命令行和伪代码等。Java 代码以等宽字体表示，并按照标准 Java 格式显示。命令行和参数也会以等宽字体显示。类似地，段落中引用的文件名、代码片段和 Java 包名也会使用等宽字体表示。

与正文相关的重要一些信息，或是补充说明，会使用中括号括起来。

此部分内容很重要！

技术名词和基本概念会作为**关键字**用黑体字表示。为便于查询，这些技术名词会列在术语表中。

在本书中，JROCKIT_HOME 和 JAVA_HOME 表示 JRockit JDK/JRE 的安装目录。例如，默认安装 JRockit 之后，Java 命令的位置是：

C:\jrockits\jrockit-jdk1.5.0_17\bin\java.exe

而 JROCKIT_HOME 和 JAVA_HOME 的值则为：

C:\jrockits\jrockit-jdk1.5.0_17\

JRockit JVM 有其自己的版本号规则，目前最新的主版本是 R28。JRockit 的次版本号表示在发行主版本后第几次发行小版本，例如 R27.1 和 R27.2。本书中使用 R27.*x* 表示所有的 R27 版本，R28.*x* 表示所有的 R28 版本。

默认情况下，本书所介绍的内容是以 R28 版本为基础的，针对之前版本的内容会特别说明。

JRockit Mission Control 客户端使用了更加标准的版本号规则，例如 4.0。在介绍 JRockit Mission Control 的相关工具时，工具的版本号 3.*x* 和 4.0 也分别对应了 JRockit Mission Control 客户端的版本。在写作本书时，JRockit Mission Control 客户端的最新版本是 4.0，除非特别指明，所有内容均是以此版本为基础来讲解的。

书中内容有时会涉及一些第三方产品。阅读本书时无须十分了解这些产品。涉及的第三方产品如下。

- Oracle WebLogic Server：Oracle J2EE 应用服务器
- Oracle Coherence：Oracle 内存型分布式缓存技术
- Oracle Enterprise Manager：Oracle 应用程序管理套件
- Eclipse：Java IDE（也可用于其他语言的开发）
- HotSpot™：HotSpot™JVM

读者反馈

欢迎读者反馈对本书的看法，喜欢什么、不喜欢什么，这对我们开发读者真正需要的选题来说非常重要。

要发送反馈信息，可以直接发邮件到 feedback@packtpub.com，并在邮件主题中注明书名。

如果你需要某本书，希望我们出版的话，请在 PacktPub 的官网 www.packtpub.com 中填写表单，或者发邮件到 suggest@packtpub.com 来说明。

如果读者精通某个领域，并且想要撰写或参与写作一本书的话，请阅读 www.packtpub.com/authors 中的作者指南。

客户支持

针对购买了 Packt 图书的读者，我们提供了很多周边内容，帮助你更好地理解书中内容。

下载本书示例代码

可从 http://www.ituring.com.cn/book/2491 下载本书中的示例代码，以及代码的使用说明。

勘误

尽管我们尽力确保书中内容无误，但错误在所难免。如果读者发现了错误，不管是文字错误还是代码错误，敬请告知，我们将感激不尽。这不仅可使其他读者免受错误困扰，还可以帮助我们完善本书后续的版本。如果读者发现了任何错误，请访问 http://www.packtpub.com/support，选择书名，然后点击 let us know 链接，输入勘误的具体内容。[①]当勘误通过验证后，内容将被接受，而且该勘误信息将上传到我们的网站，或者添加到该书下面 Errata 部分的已有勘误表列表当中。在 http://www.packtpub.com/support 可以看到目前已有的勘误表。

盗版问题

对所有媒体来说，互联网盗版都是一个长期存在的问题。Packt 公司对自己的版权和许可证的保护非常严格。如果你在互联网上遇到以任何形式非法复制我们作品的行为，请立刻向我们提供具体地址或网站名称，以帮助我们采取补救措施。

请通过 copyright@packtpub.com 联系我们，并且附上可疑盗版资料的链接。

感谢你帮助我们保护作者，使我们能够带给你更有价值的内容。

---

① 针对本书中文版的勘误，请到 http://www.ituring.com.cn/book/2491 查看和提交。——编者注

疑问

如果读者对本书有任何疑问，请发邮件至 questions@packtpub.com 说明，我们会尽力解决。

致谢

感谢这些年一直陪伴在我们身边的富有创造力的人们。特别是 Appeal 的同事，你们已经成为我们生活的一部分，我们很荣幸能与如此卓越的团队分享这段历程。

此外，非常感谢我们的家人，感谢你们在本书写作期间给予我们的耐心和支持。

电子书

扫描如下二维码，即可购买本书电子版。

目　录