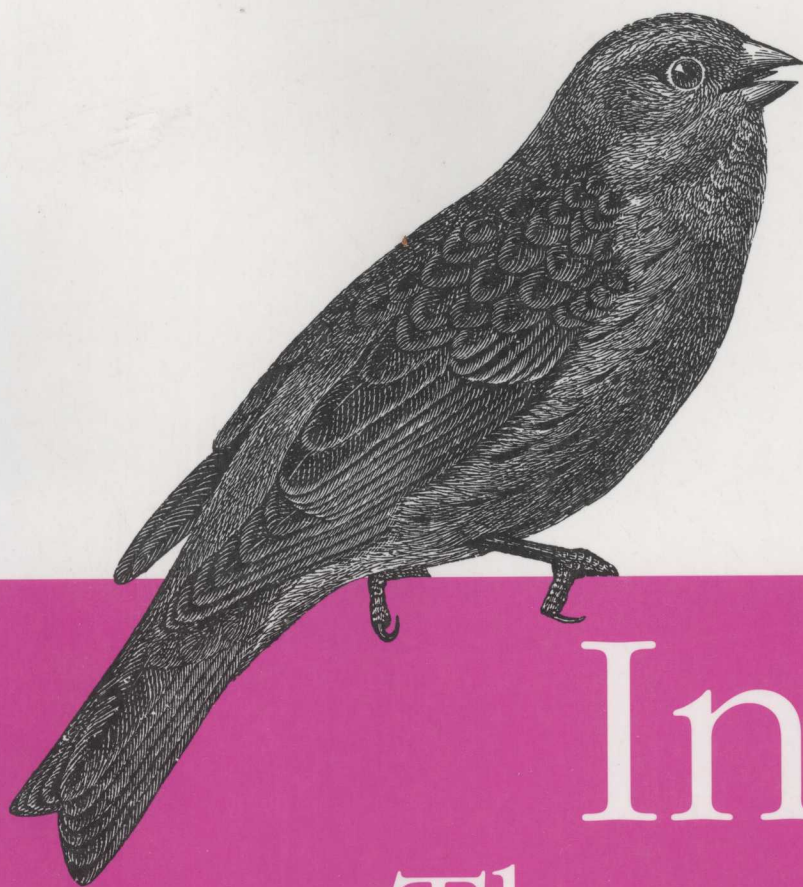


Intel Threading Building Blocks



Intel Threading Building Blocks

编程指南

James Reinders (仁达敬) 著

Alexander Stepanov 序

聂雪军 译

O'REILLY®



机械工业出版社
China Machine Press



Intel Threading Building Blocks 编程指南

James Reinders (仁达敬) 著
聂雪军 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

图书在版编目 (CIP) 数据

Intel Threading Building Blocks 编程指南: / (美) 仁达敬 (Reinders, J.) 著; 聂雪军等译. —北京: 机械工业出版社, 2009.1

书名原文: Intel Threading Building Blocks

ISBN 978-7-111-24231-4

I. I… II. ①仁…②聂… III. 并行程序—程序设计 IV. TP311.11

中国版本图书馆 CIP 数据核字 (2008) 第 103392 号

北京市版权局著作权合同登记

图字: 01-2007-4190 号

©2007 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2008. Authorized translation of the English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2007。

简体中文版由机械工业出版社出版 2008。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

本书法律顾问

北京市展达律师事务所

书 名 / Intel Threading Building Blocks 编程指南

书 号 / ISBN 978-7-111-24231-4

责任编辑 / 李东震

封面设计 / Edie Freedman, 张健

出版发行 / 机械工业出版社

地 址 / 北京市西城区百万庄大街 22 号 (邮编 100037)

印 刷 / 北京牛山世兴印刷厂

开 本 / 178 毫米 × 233 毫米 16 开本 21.5 印张

版 次 / 2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷

定 价 / 49.00 元 (册)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010) 68326294

对本书的赞誉

“串行计算的时代已经结束了。随着多核处理器的出现，过去只在大学和试验室里进行研究的并行计算技术现在已经作为主流出现了。Intel线程构建模块升级并极大地扩充了麻省理工大学的 Cilk 系统在 15 年前首创的‘工作窃取 (Work Stealing)’技术，它为并行编程提供了具有工业强度的现代 C++ 库。”

“本书很好地介绍了线程构建模块库，无论是初学者还是专家，都能通过本书清晰地理解并行的复杂性。”

——Charles E. Leiserson，麻省理工大学计算机与人工智能实验室

“我们通常习惯说，首先要使程序正确，然后再提高程序的速度。现在，我们无须这么做。TBB (Threading Building Block, 线程构建模块) 能够使我们在地地设计程序同时，还能提升 Maya 前台视图的速度。本书将向你展示如何通过在代码中使用 TBB 来获得最大的性能提升。”

——Martin Watt，Autodesk 公司高级软件工程师

“TBB 必将改变 C++ 编写并行程序的方式。本书对于任何程序员来说都是非常有用的。在本书中，James 实现了两个重要的目标：

- 详细地介绍了并行编程，阐述了最为常见的并行编程模式以及推动它们应用的动力；
- 详细地介绍了线程构建模块 C++ 库——在这个库中为这些并行编程模式提供了泛型算法。”

“TBB 融合了在过去二十年中面向对象并行计算领域的研究者们所获得的许多优秀思想。”

——Marc Snir，伊利诺大学香槟分校计算机系主任

“本书是我了解Intel线程构建模块的第一本书。它非常清晰地介绍了在TBB中实现的各种功能以及一些决策背后的决定因素,这进一步说明了在Intel线程构建模块中融合了目前一些最优秀的并行编程思想。在书中选择了一组为数不多但却功能强大的模式作为讲解示例,这使得本书非常易于学习和使用。我认为书中给出的代码片段和完整的应用程序都能非常有助于理解TBB库的主要功能,并且很好地说明了这个库在开发高效并行程序时的各种使用方式。”

——David Padua, 伊利诺大学

“多核芯片架构的出现给并行编程带来了极大的挑战,许多人都希望有一本好书以帮助并且指导用户来面对这些挑战。”

“本书介绍的是Intel线程构建模块,它为这些挑战提出了非常好的解决方案。在指导用户如何在多核处理器上并行编程方面,本书无疑是一份重量级的参考资料。”

“本书为读者们指出了一条独特的技术路线,即基于C++的并行编程模式——这是并行编程中一种功能强大并且切实可行的方法。本书在构思和编撰上都是严谨而细致的,它既可以作为培训的教材,也可以作为工程师们的参考资料。”

——Guang R.Gao 教授, 得拉瓦大学

“我很喜欢读这本书。它能够满足软件工程师们在开发下一代并行程序中对新技术思路的需求。当‘信息技术中最具爆炸性的技术’之一(这里指的是多核架构)出现时,开发人员对于这种新技术思路是非常渴望的。”

“本书还可以作为一本重要的指导资料,它能为软件工程师们指出新的机遇和挑战。”

“在线程构建模块中采用的基于库的方法是一种重要的全新举措,它很好地弥补了依赖高级编译器技术方法中的不足。”

——Rudolf Eigenmann, 普度大学电子与计算机工程系教授及计算研究中心代理主任

“多核系统的时代已经来临了。在编写利用多核功能的程序时,我们需要一些并行编程模型。此外,我们需要进行大量的培训以帮助编写串行程序的开发人员适应这种新技术的需求。本书代表了在这两个前沿方向上的进展。”

“线程构建模块(TBB)是在构造并行程序时一种灵活的、基于库的方法,这种方法能够很好地与其他编程解决方案结合起来使用。”

“本书在介绍 TBB 的方式上既有趣又易懂。而且，无论对于初学者还是有经验的程序员来说，书中丰富的信息都可以帮助他们将 TBB 应用于解决他们自己的编程问题。”

——Barbara Chapman, cOMPunity 公司 CEO, 休斯敦大学计算机科学系教授

“在未来的芯片中，将包含数十个乃至数百个核。在编写程序时，如果希望充分利用这些芯片所提供的强大功能，那么对于那些习惯使用串行算法而不是并行算法的程序员来说，并不是件容易的事情。”

“Intel 的 TBB 沿着并行编程这条漫长的道路前进了一大步，而更棒的是，所有功能都是用 C++ 实现的。”

——Eduard Ayguade, 巴塞罗那超级计算机中心，加泰罗尼亚理工大学

“Intel TBB 的应用领域是并行编程，而 STL 的应用领域只是普通的 C++。STL 中的范型编程极大地提高了 C++ 的编程效率。在 TBB 中提供了范型并行编程模型，它能够很好地屏蔽并发控制的复杂性。它不仅降低了开发并行代码的门槛，而且还实现了对多核的高效利用。”

——Lawrence Rauchwerger, 德克萨斯 A&M 大学，STAPL 的发明者

“最近 18 年，在超级计算机这个高端的研究领域中，一直都在寻找着一种能够通过 C++ 来编写高效并行的方式。我们知道模板和范型编程应该属于这个答案的一部分，但只有在多核（以及不久之后的百核）处理器出现之后，才给计算领域带来了根本性的变化。并行（Parallelism）将成为我们日常工作的一部分。”

“每个 C++ 程序员都开始考虑并发性和并行性，而线程构建模块正是提供了实现它们的正确抽象。”

“本书不仅仅只是讨论一个 C++ 模板库。它深入浅出地介绍了我们在过去 25 年中在并行计算领域中的研究工作。本书是学习并行编程的优秀教材。”

——Dennis Gannon, 印第安纳大学 Pervasive 技术实验室科学主任, DARPA 高性能计算 (HPC++) 项目前任主管以及全球网络论坛指导委员会成员

“TBB 中的众多优势深深地吸引了应用程序的开发人员，这些优势包括单处理器性能、并行可伸缩性、优于 OpenMP 的 C++ 编程能力、与 OpenMP 和底层原始线程的良好兼容性、能够优化性能和检测错误的 Intel 线程工具以及对于软件开发社群的开放性等。TBB 还消除了一些功能约束：语言扩展中的风险、特定的编译器依赖性以及底层原始线程的复杂性。”

“本书使得开发人员无需陡峭的学习曲线就可以提高编程效率,并且他们编写的程序都将是高质量和高性能的。”

——David Kuck, Intel 成员, KAI 公司的奠基者及超级计算研发中心的前任主管

Intel Threading Building Blocks 编程指南

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者序

计算机领域正在发生着一场计算方式的革命。Intel、AMD 以及其他芯片制造商不断推出在单晶片上集成多重处理单元的新型芯片，用于取代过去的单一中心处理器。多核处理器的问世给计算机工业带来更强大的能力，但同时也给软件产业带来更大的挑战：我们不得不面对并发和并行程序这些通常是并行计算的专业人员和高端用户才需要面对的问题。

传统的并行程序设计模型主要分为两大类，一类是共享存储模型，主要是采用多线程，例如目前已成为事实工业标准的 OpenMP；另一类是消息传递模型，例如 MPI (Message Passing Interface, MPI)。然而，这些方式都存在着一些缺陷，或者是需要特定的语言扩展，或者是在使用上存在困难，从而降低了并行程序的开发效率。

事实上，并行程序的设计目标是挖掘问题求解过程中的并行语义，寻求并行算法与并行机器体系结构的最佳匹配和映射，合理组织并行任务，减少额外消息传递和数据移动开销，而这正是 Intel 线程构建模块的设计初衷。

与 OpenMP、MPI 等传统技术相比，线程构建模块主要有以下特点：

1. 面向任务编程。任务是比线程更高级的并行抽象，任务与线程的关系就好比高级语言与汇编语言的关系。在使用线程构建模块时，需要设计的是任务而不是线程，这就使你无须关心一些具体的线程细节技术，例如负载均衡，优化调度等。线程构建模块的底层组件已经帮你实现了这些烦琐的工作，它将通过高效的方式将任务映射到线程，并通过任务窃取 (Task Stealing) 来实现自动调度。这正如在用 C++ 编程时无需编写 push/pop 之类的指令，C++ 编译器将自动实现这些工作。
2. 简单易用。线程构建模块是一个用标准 C++ 和模板实现的库，它既没有包含像 OpenMP 中的特殊语言扩展，也不是一种全新的编程语言。任何支持 ISO C++ 的编译器都可以直接编译 TBB 程序，而现有的程序也无需太多修改就可以通过线程构建模块获得并行的速度提升。计算机领域的发展历史已经证明，这一特点是决定技术生命力的重要因素。
3. 丰富的辅助开发工具。与线程构建模块结合使用的有许多辅助开发工具，例如 Intel 线程检查器，Intel 线程分析器等。这些辅助工具将极大地提高并行程序的开发效率和调试效率。

本书作者 James Reinders (仁达敬) 是 Intel 软件开发产品部门的首席讲师, 他在并行程序设计领域有着丰富的经验。本书融合了作者和 Intel 线程构建模块开发团队在并行程序设计领域和线程构建模块中的宝贵经验, 既有一定的理论深度, 又包含了丰富的编程示例。

本书对读者的技术水平要求不高, 对于各个层次的读者, 本书都将是非常有用的。但如果读者具备一些 C++ 模板以及模板库等方面的知识, 那么就更容易地理解本书中的一些概念。多核时代已经来临, 并行程序已经被推向了主流地位, 这既带来了挑战, 也带来了机遇。衷心希望本书能够为并行程序开发人员带来帮助。

参与本书翻译工作的还有李杨、吴汉平、徐光景、童胜汉、陈军、胡凯、刘红、张玮、陈红和李斌。由于译者的时间和水平有限, 翻译中的疏漏之处在所难免, 还望读者和同行不吝指正。

致谢

感谢妻子云兰和女儿彤彤给我的鼓励, 使我能够保持愉快的工作状态。感谢我的父母一如既往地支持着我。

聂雪军

2008年3月于武汉

作者简介

James Reinders (仁达敬), Intel 软件开发产品部门的首席讲师, 他是一位高级工程师, 于1989年加入Intel公司。他在许多项目上做出了贡献, 包括世界上第一台万亿次浮点运算超级计算机 (ASCI Red) 以及 iWarp、Pentium Pro、Pentium II、Itanium、Pentium 4 等处理器上的编译器和架构工作。他在处理器架构、编译器优化、并行计算机架构以及为软件开发人员编写工具产品上有着多年的经验。

Reinders 还是 www.goparallel.com 网站上月刊 “The Gauntlet” 的专栏编辑, 他还编写了《VTune Performance Analyzer Essentials》(Intel Press) 一书, 并参与编写了《Multi-Core Programming》(Intel Press) 一书。

封面介绍

本书封面上的动物是一只野生金丝雀, 它是雀科中的一种小型鸣禽。它的别名也叫作加那利金丝雀或者大西洋金丝雀, 因为它原产自位于西欧的岛屿, 尤其是马德拉群岛、亚述尔群岛和加那利群岛等。金丝雀这个名字来源于拉丁文中的 canaria (意指“狗”), Pliny the Elder 在他的《Naturalis Historia》中最早使用了这个名字, 得源于生活在这些岛上的巨型狗。金丝雀通常生活在果园、农田以及灌木丛, 它们在灌木和树林中筑巢。

虽然野生金丝雀比驯养的金丝雀要稍大一些, 它们在外表上是非常相似的。它的胸脯为黄绿色, 背上有褐色斑纹。与许多其他物种一样, 雄性金丝雀的色泽比雌性金丝雀的色泽要更为鲜明。雄性金丝雀还有一副好嗓子。当西班牙人在15世纪征服这些群岛时, 他们驯养了金丝雀并且开始繁殖它们。到了16世纪, 金丝雀在整个欧洲都被当做一种宠物。(Samuel Pepys 在1661年的一篇日记中记述了他的“金丝雀”)。在有选择性地繁殖了五百年之后, 进化出了许多金丝雀品种, 包括现在常见的亮黄色种类的金丝雀。金丝雀是一种大受欢迎的宠物, 因为它可以存活10年左右, 不需要特殊的照料, 而且人们认为它的叫声是所有鸟类中最悦耳的。

在20世纪80年代, 煤矿工人将金丝雀用作一种报警工具, 在每个矿井中放两只金丝雀。根据美国矿产局的报告, 金丝雀的报警效果比老鼠要好, 因为它们对气体更敏感, 并且在出现瓦斯时更显得不安。矿井中的金丝雀可以叫一整天, 但如果一氧化碳浓度上升, 它将停止鸣叫并且在栖枝上抖动, 直至死亡——这可以警告矿工们赶快离开。

目录

致中文版读者	1
序	5
Intel 线程构建模块首席开发师的笔记	7
前言	11
第 1 章 为什么需要线程构建模块	19
简介	20
优势	20
第 2 章 并行思维	25
并行思维的要素	25
分解	27
扩展性与加速性	31

什么是线程	37
互斥和锁	39
正确性	40
抽象	43
模式	43
直觉思维	45
第 3 章 基本算法	46
库的初始化和结束	47
循环的并行化	49
递归区间的定义	69
对循环的小结	83
第 4 章 高级算法	84
用于流的并行算法	85
第 5 章 容器	100
concurrent_queue	101
concurrent_vector	106
concurrent_hash_map	112
第 6 章 可扩展的内存分配	123
限制	123
内存分配中的若干问题	124
内存分配器	124
替换 malloc、new 和 delete	126
第 7 章 互斥	133
何时使用互斥	133
Mutexes	135

互斥量	142
原子操作	147
第 8 章 计时	155
第 9 章 任务调度器	159
基于任务的编程在什么情况下是不合适的	159
远远好于本地裸线程	160
初始化库是必需的	163
斐波纳契 (Fibonacci) 数列示例	163
任务调度简介	166
任务调度的工作原理	168
任务重现模式	171
高效利用调度器	173
任务调度器的接口	179
任务调度器小结	196
第 10 章 成功的关键因素	197
通向成功的关键步骤	197
松弛串行执行	198
方法和库的安全并发性	199
调试版本与发布版本	200
为了提高效率	200
支持调试功能	200
与其他线程软件包混合使用	203
命名约定	204
第 11 章 编程示例	206
Aha!	206
其他一些关键点	208

parallel_for 示例	209
生命游戏	219
parallel_reduce 示例	227
统计字符串: 使用 concurrent_hash_map	237
Quicksort: 可视的任务密取	243
一种更优的矩阵乘法 (Strassen 算法)	251
高级任务编程	257
网络数据包处理流水线	266
内存分配	285
游戏线程示例	289
物理相互作用的代码和更新的代码	299
Open Dynamics Engine	303

第 12 章 发展历程及相关项目

库	311
语言	312
编译器指令	313
泛型编程	313
快存	316
时间分片的开销	316
Lambda 函数简介	317
阅读参考	318

致中文版读者

距本书第一版（英文版）正式推出不到一个月，Autodesk 就发布了新版本的 Maya，这个版本包含了千万行程序代码，其核心部件利用了线程构建模块 TBB 来增强并发性。他们告诉我说在他们的应用程序中 TBB 是如此易于实现，并且能带来如此出众的效果性能，使他们感到了强烈的震撼。这仅仅是在本书及开源发布不到半年时间内一系列所公布的使用了 TBB 的商业产品和开源项目的一个开始。而现在，TBB 已经发展到 2.1 版本。当前，TBB 随着红旗 Linux、OpenSolaris、Solaris、Red Hat Fedora 8 和 9、OpenSUSE、Ubuntu、TuboLinux、Haansoft、Micacle Linux 和 VietSoftware 一起发布。TBB 与英特尔编译器等一系列产品一起，被称为英特尔并行工作室（Intel Parallel Studio）新产品系列的核心。

我们这个圈儿里的其他开发者已经把 TBB 移植到更多的平台上了。新近出现的有 Sun Solaris 系统上的 Sun Studio，运行在 PowerPC 平台上的 Microsoft Xbox* 360 以及 Apple Mac OS X。其中 Apple Mac OS X 的出现扩充了操作系统大家庭，这些系统包括 Windows、Linux、Intel 平台上的 Mac OS X，以及 Intel 最先参与开发的 FreeBSD 系统的各个版本。

我们知道许多公司已经发布了捆绑使用 TBB 的产品，包括 Autodesk Maya，Pixar Renderman，Real Image AvidXpress，Autodesk REVIT，Mentor Graphics Calibre，National Instruments Vision Software，Knowledge Miner，Deep Shadows，Weeden PriceServer，Honeywell SMARTlab，用于数字成像的 Ricoh 图像处理引擎，以及 Canon 的成像和光学系统。我知道还有很多在主流应用产品使用了 TBB，非常遗憾在这里我未能一一列出。