

<<Intel Threading Buil>>

图书基本信息

书名：<<Intel Threading Building Blocks编程指南>>

13位ISBN编号：9787111242314

10位ISBN编号：7111242319

出版时间：2009-1

出版时间：机械工业出版社

作者：仁达敬

页数：323

译者：聂雪军

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：<http://www.tushu007.com>

前言

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。

<<Intel Threading Build>>

内容概要

多核处理器的出现使得并行编程成为了每个程序员关注的主题。

不具备多处理器核的计算机系统已经变得越来越少。

本书为c++程序给出了并行编程的解决方案，它既不要求放弃c++，也不要求使用那些裸线程或者本地线程。

本书介绍的是Intel线程构建模块。

线程构建模块是一个用来实现并行语义的c++模板库，它对c++进行了扩展，抽象出了线程管理机制并支持简明的并行编程。

在使用这个库时，你所要指定的是任务而不是线程，然后库将高效自动地将这些任务映射到线程上。

如果希望程序在处理器核不断增加的情况下也能够继续运行，那么这种方式将带来许多好处。

与裸线程相比，线程构建模块能够使你更方便地表达并行语义，同时还能够提高程序的性能，可移植性和可扩展性。

在使用本书时，你无需任何并行编程经验或者在多核处理器上的编程经验。

无论你是拥有丰富的并行编程经验，还是对并行编程一无所知，本书的内容都将是非常有用的。

此外，本书也不要求之前是否对线程有所了解。

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、Itanium、Pentium 4等处理器上的编译器和

<<Intel Threading Buil>>

书籍目录

致中文版读者序 Intel线程构建模块首席开发师的笔记前言第1章 为什么需要线程构建模块 简介 优势第2章 并行思维 并行思维的要素 分解 扩展性与加速性 什么是线程 互斥和锁 正确性 抽象模式 直觉思维第3章 基本算法 库的初始化和结束 循环的并行化 递归区间的定义 对循环的小结第4章 高级算法 用于流的并行算法第5章 容器 concurrent_queue ConCUrrent_VeCtOr concurrent_hash_map第6章 可扩展的内存分配 . 限制 内存分配中的若干问题 内存分配器 替换malloc、new和delete第7章 互斥 何时使用互斥 Mutexes 互斥量 原子操作第8章 计量第9章 任务调度器第10章 成功的关键因素第11章 编程示例第12章 发展历程及相关项目

章节摘录

第1章 为什么需要线程构建模块 简介 多核处理器正在变得日益普及，而利用现有的线程软件包来编写一个简单的parallel_for循环却是一项乏味的工作。

要编写一个高效的可扩展(scalable)程序则更为困难。

可扩展性体现的是这样的概念：随着处理器中核数量的增加，程序的性能能够得到相应的提升。

随着在新推出的处理器中有着越来越多的核，使用线程构建模块来编写程序将帮助你从中获得许多好处。

线程构建模块是一个用来支撑可扩展并行编程的库，它是用标准的C++代码实现的。

它不需要专门的语言或者编译器。

线程构建模块可以几乎应用于所有处理器或者操作系统上的任何一种c++编译器中，这种强大的功能对于程序员来说是非常有吸引力的。

线程构建模块通过模板来提供常见的并行迭代模式，这使得程序员即使在不具备同步、负载均衡以及快存优化等专门知识的情况下，也能够从多核处理器中获得性能提升。

使用线程构建模块编写的程序既能够运行在单核处理器的系统上，也能够运行在多核处理器的系统上。

线程构建模块促进了可扩展数据并行编程技术的发展。

此外，它还支持嵌套并行，因此你可以很容易地从小型的并行组件中构建出大型的并行组件。

在使用这个库时，你需要指定的是任务，而不是线程，然后再通过这个库以高效的方式将任务自动映射到线程。

因此，与裸线程(Raw Thread)相比，线程构建模块能够更方便地实现并行，并获得更优的结果。

<<Intel Threading Buil>>

媒体关注与评论

“ 串行计算的时代已经结束了……《Intel Threading Building Blocks编程指南》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 Intel Threading Building Blocks编程指南》清晰地理解并发的复杂性。

” ——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，我们无需这么做。

TBB使我们在正确地设计程序的同时，还能提升Maya前台视图的速度。

Intel Threading Building Blocks编程指南》将向你展示如何通过代码中使用TBB来获得最大的性能提升。

” ——Martin Watt , Autodesk公司高级软件工程师

编辑推荐

“ 串行计算的时代已经结束了……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发的复杂性。

” ——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，我们无需这么做。

TBB使我们在正确地设计程序的同时，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本书介绍了如何通过Intel线程构建模块来最大限度地利用多核处理器的强大性能。

TBB是一个可移植的C++库，能够运行在Windows、Linux、Macintosh以及uNix等系统上。

本书阐述了在多线程编程中的关键任务以及如何通过TBB上可移植的和稳定的方式来实现它们。

书中包含了大量的示例和丰富的资料，给出了常见的使用模式，揭示了TBB中的精彩之处，并且还提供了一些指导意见，用来在不同的并行编程方式上进行选择以获得最佳的性能。

本书的内容包括：? 充分利用多核系统中并行语义的简单且高效的方式。

- ? 编写并行程序的关键问题。
- ? 多线程中的常见模式。
- ? 用于高效处理的线程安全容器。
- ? 任务调度。
- ? 在线程环境中的内存管理。

本书说明了TBB能够比原始线程更方便地在c++中指定并行语义，同时还可以提升程序的性能、可移植性以及可伸缩性。

在开始学习TBB时，并不要求具备并行编程或者多核处理器的经验。

任何希望在多核系统上编写软件的程序员，都能够从本书中获益。

<<Intel Threading Buil>>

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:<http://www.tushu007.com>