

<<释放多核潜能>>

图书基本信息

书名：<<释放多核潜能>>

13位ISBN编号：9787302235033

10位ISBN编号：7302235031

出版时间：2010-9

出版时间：清华大学出版社

作者：英特尔亚太研发有限公司、并行科技 编著

页数：287

字数：475000

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>

<<释放多核潜能>>

前言

随着CPU技术进入多核时代，如何有效利用多核并发运算优势，提升设备的处理能力，满足用户对高性能的需求，已成为业界关注的焦点。

在单核CPU时代，处理器芯片厂商提升CPU运算能力的主要途径是提高主频。

作为CPU的主要性能指标，主频标志着每单位时间内CPU能够执行运算指令的数量。

实际上，在单核CPU时代，处理器已经实现了多线程运算，通过在逻辑上模拟出多个CPU内核，以实现多任务调度和并发处理。

然而，这些处理过程始终由单个CPU以线程切换的方式完成的，运算负载由单个CPU承担。

而多核CPU则在真正意义上实现了内核级并行，与传统的单核CPU相比，多核CPU带来了更强的并行处理能力、更高的计算密度和更低的时钟频率，并大大减少了散热和功耗。

目前，越来越多用户的PC系统都具备了双核、四核的处理器。

随着处理器数的增加，程序的并行执行度可以更高，但是目前不少用户觉得多核没有带来很明显的性能提升，这是因为现在针对多核开发和优化的应用程序还比较少。

主要原因之一就是开发并行执行程序的难度非常大，程序员面临的巨大挑战就是如何把需要执行的任务并行化，而编写并行度超过4路以上的高效率程序。

程序员没有经过系统的专业学习和长期并行编程的实践经验，编写的程序就很难充分利用多核处理器带来的并行计算优势。

即使是并行编程经验丰富的程序员在编写并行度较高的程序时的效率，相比他编写并行度低的程序时的效率也要低得多。

<<释放多核潜能>>

内容概要

本书采用工程理论、工具详解和实际案例分析相结合的方式，全面介绍了英特尔Parallel Studio工具集的使用。

全书分三部分：基础部分（第1、2章）介绍了多核架构、并行编程的关键理论，Parallel Studio的特点以及一些简单案例；中级部分（第3~12章）详述了Parallel Studio各个组件的使用，是本书的重点；提高部分（第13章）选取了来自英特尔线程挑战赛的4个算例和1个商业软件并行优化案例，提供了从工程实际角度解决并行问题的视角。

本书适合所有对并行开发技术感兴趣的人员，包括具备一定编程经验的程序员、调试人员，计算密集型行业的高性能计算架构师、性能优化分析师，并行开发的研究人员，对英特尔Parallel Studio感兴趣的技术决策者等。

此外，本书也可作为高等院校计算机专业并行开发相关课程的培训及社会实践参考用书。

作者简介

并行科技

北京并行科技有限公司（以下简称并行科技），是一家专注于高性能计算软件与技术服务的高新技术企业，与英特尔等硬件厂商有着密切的合作伙伴关系，主要客户包括科研院所、能源、气象、制造、金融、互联网等计算密集型用户。

在程序并行化、优化领域，并行科技拥有自主知识产权的Paramon、Paraview专业工具，积累了从搜集应用特征、定位性能瓶颈到分级优化的系统方法，被誉为“性能专家”。

同时，并行科技作为英特尔软件代理商，也为用户提供卓越的英特尔软件技术支持、培训等。

英特尔高性能计算支持团队

英特尔高性能计算支持团队，专注于多核平台和服务器集群的并行应用优化，负责为国内高性能计算和互联网数据中心用户，提供应用性能优化、代码并行、应用特征分析、开发工具培训以及并行解决方案建议和评估等工程支持，在石油勘探、制造业、气候气象、生命科学和互联网搜索引擎等大规模并行应用领域积累了丰富的经验。

英特尔软件工具技术顾问团队

英特尔软件工具技术顾问团队，由英特尔各领域专家组成，为所有英特尔软件工具，如C/C++编译器、Fortran编译器、VTune性能分析器、核心数学库、英特尔Parallel Studio等，提供专业的售前、售后服务，并提供相关技术的高级培训。

<<释放多核潜能>>

书籍目录

目 录第1章 并行开发理论基础 1.1 并行相关概念 1.1.1 并发与并行、并行度 1.1.2 粒度 1.1.3 加速比及定律 1.1.4 可扩展性与并行效率 1.1.5 负载均衡 1.1.6 吞吐量与延迟 1.1.7 热点与瓶颈 1.2 多核并行 1.2.1 核软硬件现实 1.2.2 多核架构 1.2.3 多核并行手段 1.2.4 多核并行设计方法 1.2.5 多核多线程系统 1.2.6 多线程同步 1.2.7 多核多线程实现的问题 1.3 小结 第2章 英特尔Parallel Studio基础 2.1 英特尔Parallel Studio介绍 2.1.1 英特尔 Parallel Studio背景 2.1.2 英特尔 Parallel Studio的组成 2.1.3 英特尔 Parallel Studio的特色 2.1.4 英特尔 Parallel Studio的使用者 2.2 英特尔Parallel Studio快速上手 2.2.1 英特尔 Parallel Studio的下载安装 2.2.2 选择案例 2.2.3 实践动手第一步：采用Parallel Studio运行串行程序 2.2.4 实践动手第二步：选用合适的实现对代码并行化 2.2.5 实践动手第三步：定位错误 2.2.6 实践动手第四步：性能优化 2.3 小结 第3章 英特尔Parallel Composer详解 3.1 Composer概述 3.2 英特尔C/C++编译器 3.2.1 自动并行和OpenMP并行 3.2.2 过程间优化 3.2.3 档案导引优化 3.2.4 编译器向量化 3.3 英特尔并行调试器 3.3.1 英特尔并行调试器概述 3.3.2 线程数据共享侦测 3.3.3 可重入函数调用侦测 3.3.4 SSE寄存器窗口 3.3.5 OpenMP多线程试 3.3.6 并行区域的串行执行 3.4 英特尔TBB线程构建模块 3.4.1 英特尔TBB概述 3.4.2 功能模块分类与介绍 3.4.3 编译和运行TBB多线程程序 3.5 英特尔IPP性能基元 3.5.1 英特尔IPP概述 3.5.2 主要函数及其功能 3.5.3 编译和运行 3.6 小结 第4章 并行化方法 4.1 基本概念 4.1.1 Amdahl定律 4.1.2 进程与线程并行化方法 4.3 并行化设计 4.3.1 任务划分 4.3.2 功能划分 4.3.3 并行化开发中的一些思考 4.4 案例分析：特卡罗方法计算 值 4.5 小结 第5章 英特尔Parallel Composer案例分析 5.1 案例5-1：Composer的使用——向量化和自动并行化 5.2 案例5-2：并行调试器的使用 5.3 案例5-3：通过TBB进行字符串查找 5.4 案例5-4：IPP压缩和解压缩案例介绍 5.5 小结 第6章 英特尔Parallel Inspector详解 6.1 Inspector概述 6.2 启动Inspectort 6.2.1 工作流程 6.2.2 启动 6.3 配置查找错误的类型和粒度 6.3.1 基于线程的相关错误及粒度 6.3.2 基于内存的相关错误及粒度 6.4 定位和解决发现的错误 6.4.1 检查错误 6.4.2 查看和分析错误 小结 第7章 软件纠错方法 7.1 基本概念 7.1.1 软件查错或纠错 7.1.2 白盒测试 7.1.3 黑盒测试 7.2 并行的纠错 7.3 线程并行的常见错误 7.3.1 线程间死锁 7.3.2 线程间竞争 7.3.3 内存泄露 7.4 小结 第8章 软件纠错案例 8.1 案例8-1：线程间相互作用导致的死锁问题 8.2 案例8-2：线程竞争 8.3 案例8-3：内存泄露 8.4 小结 第9章 英特尔Parallel Amplifier详解 9.1 Amplifier概述 9.1.1 如何开始Amplifier 9.1.2 如何使用信号信息 9.1.3 环境和对象 9.2 Amplifier的几个概念 9.3 Amplifier的分析运行 9.3.1 分析运行的几个选项 选择分析模式 9.3.3 如何选择分析模式 9.3.4 如何在命令行下运行分析模式 9.3.5 热点：分析程序哪里耗时 9.3.6 并行度：展现并行程程序的另外一个特点 9.3.7 锁和等待：分析程序在哪里等待 9.3.8 选择数据采集的时段 9.4 Amplifier中浏览性能数据结果 9.4.1 总览 9.4.2 在Bottom-up和Top-down中切换 9.4.3 选择和管理栈类型 9.4.4 选择颜色方案 9.4.5 按照不同类型划分组 9.4.6 在命令行模式下查看性能数据 9.5 Amplifier解释性能数据结果 9.5.1 总览 9.5.2 解释热点分析结果 9.5.3 解释并行度分析结果 9.5.4 解释锁等待分析结果 9.6 Amplifier中的源代码 9.7 Amplifier中对比性能数据结果 9.8 Amplifier中管理结果文件 小结 第10章 性能优化方法 10.1 性能优化概述 10.1.1 性能和性能优化是计算机领域不变的主题 10.1.2 性能优化的定义 10.2 性能优化通用方法 10.2.1 性能优化的顺序 10.2.2 系统级别的性能优化 10.2.3 应用级别的性能优化 10.2.4 微架构级别的性能优化 10.2.5 性能优化工作循环 10.2.6 性能优化循环的常见问题 10.3 并行应用性能优化方法 10.3.1 概述 10.3.2 减少关键路径上的时间 10.3.3 检查是否选择最优的并行方法 10.3.4 检查是否选择合适的层级开始并行 10.3.5 Amdahl定律的检查：减少串行部分的比例 10.3.6 检查程序的负载均衡问题 10.3.7 检查程序的粒度问题 10.3.8 采用合适的线程库 10.3.9 检查同步性能问题 10.3.10 检查硬件导致的扩展性问题 10.4 小结 第11章 性能优化案例 11.1 IO并行：系统级优化案例 锁的实现：锁优化案例 11.3 同步与负载均衡：生产消费类型的优化案例 11.4 优化临界区：WinThread循环计算型优化案例 11.5 负载均衡与归约：OpenMP循环计算型优化案例 11.6 线程数，桶数与锁：Hash表与TBB优化案例 11.7 选择合适的层级并行：任务与数据并行优化案例 11.8 避免硬件性能瓶颈：内存与高速缓存优化案例 11.9 算法选择：排序优化与TBB案例 11.10 内存操作TBB优化案例 11.11 小结 第12章 英特尔Parallel Advisor详解 12.1 Advisor基础 12.1.1 Advisor总览 12.1.2 如何开始Advisor 12.2 Advisor工作流程 12.3 Annotations 12.4 Advisor 工具 12.4.1 Survey 12.4.2 Suitability 12.4.3 Correctness 1 用案例 12.5.1 SpMV并行化 12.5.2 DGEMM并行化 12.6 小结 第13章 总体系统化案例 13.1 数独 13.1

<<释放多核潜能>>

算法 13.1.2 并行优化 13.1.3 小结 13.2 最短路径 13.2.1 串行算法 13.2.2 并行优化 13.2.3 小结 13.3
序 13.3.1 串行算法 13.3.2 并行优化 13.3.3 小结 13.4 骑士巡游 13.4.1 串行算法 13.4.2 并行优化 13.4
结 13.5 商业软件Paraview 13.5.1 问题描述 13.5.2 并行优化 13.5.3 小结 附录A 英文术语表

章节摘录

插图：1.1.1并发与并行、并行度并行计算中常见的两个名词：并行与并发。

二者有微妙的不同。

并发是指从广义上讲只要同时多任务处理，就是并发；但是并发并不能保证在每一个时刻都有多个任务（线程，进程）同时工作。

例如，一个单核处理器平台并且这个平台没有或者没有打开硬件超线程或者多线程功能，在上面运行一个多线程程序，虽然这个程序是并程序，但是由于处理器资源有限，在任意一个时刻最多只有一个线程是工作的。

并行就不同。

并行是指在同一时刻，有多个任务同时运行，才称之为并行。

从性能上讲，它体现了并行的计算能力。

例如，在一个双核处理器的平台上运行多线程程序，如果并程序设计得好，在某些时刻，同时运行的线程数应该可以达到二，从而充分利用了处理器多核资源。

因此，并行是指无论从微观还是宏观上看，都是多个线程或者进程同时执行，而并发在微观上不是同时执行的，只是把时间分成若干段，使多个进程或线程快速交替地执行。

接下来还有并行度这个概念：在某个时刻同时工作的任务（线程，进程）的数量，称之为并行度。

并行度越高，说明同时运行的任务数越多，如果相应的核数合适，可以提高整体的计算能力，充分发挥并程序的效率。

当然有时候也叫并发度，但是基本上都是在微观上表示同时进行的线程 / 进程的个数。

<<释放多核潜能>>

媒体关注与评论

当它向我正确指出占用时间最长的源代码行时，我非常高兴。
修改之后，应用程序的速度几乎提高了10倍。

——一位匿名beta版测试人员这样描述英特尔 Parallel Amplifier英特尔Parallel Studio中新的分析和评测工具，使得开发新的Envivio 4Caster系列代码转换器变得更快更有效。

尤其是通过使用英特尔Parallel Inspector和英特尔Parallel Amplifier，提高了代码的可靠性及其在多核多线程环境下的性能，从而缩短了总体开发周期。

在合格性检查阶段，由于安全性提高，减少了紊乱代码的数量，错误跟踪也变得更加容易。

英特尔ParallelStudio全面优化了我们的软件产品，并缩短了上市周期。

——Eric Rosier 工程副总裁 Envivio公司英特尔Parallel Inspector和英特尔Parallel Amplifier极大地简化了查找热点和内存泄露的任务，整体性能提升两倍，同时也消除了过去几处未发现的内存泄露，我们非常满意。

——Vlad Romashko软件开发经理 OpenCascade S.A.S使用英特尔Parallel Studio能帮助我们对游戏进行性能分析，找到热点，并能定位内存错误和线程错误。

Parallel Studio与IDE完美整合，方便易用，使我们的开发效率和开发质量大幅度提高，降低了风险和成本。

——徐鸿 开发主管 第九城市信息技术有限公司处理器从单核到多核的变迁，使得软件的并行或者并发成为必然。

“工欲善其事，必先利其器”，并程序的调试和优化尤其需要出色的开发工具。

英特尔Parallel Studio方便、高效，在传统的高性能计算领域已经发挥了巨大的作用，而对每一个有志于进行并行软件开发的程序员来说，它都应该是首选的开发工具。

——姚继锋 原上海超级计算机中心首席架构师 现中科嘉速（北京）并行软件有限公司总经理在TechED 2009的英特尔展台，我们向用户推荐了英特尔Parallel Studio，他们都认为并行开发是未来的发展方向。

Parallel Studio面向Windows用户，4个组件涵盖查找热点、调试编译、验证和调优，与项目开发过程相吻合，很容易上手。

这本书可以说是英特尔ParallelStudio并行开发的“红宝书”值得一读。

——颜伟 技术工程师 北京并行科技有限公司

<<释放多核潜能>>

编辑推荐

《释放多核潜能:英特尔Parallel Studio并行开发指南》：多核时代，从串行到并行，你的编程思想和开发技术必修升级！

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>