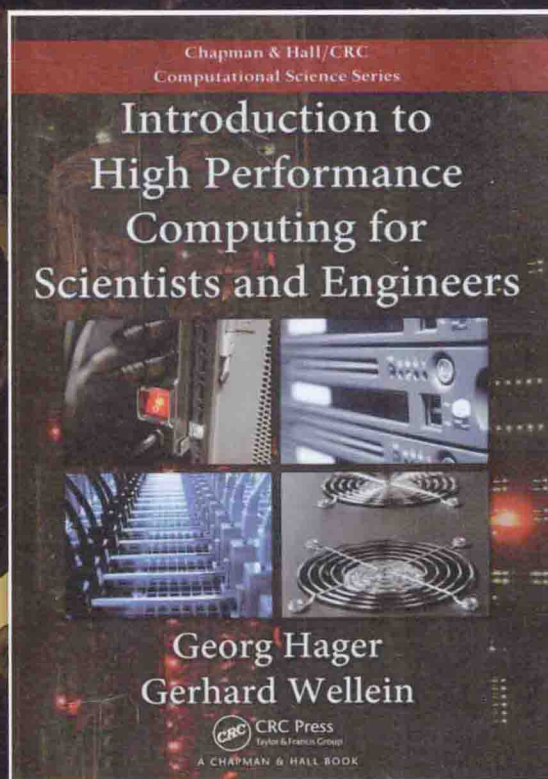


高性能科学与工程计算

(德) Georg Hager Gerhard Wellein 著

张云泉 袁良 贾海鹏 张先轶 译

Introduction to High Performance Computing
for Scientists and Engineers

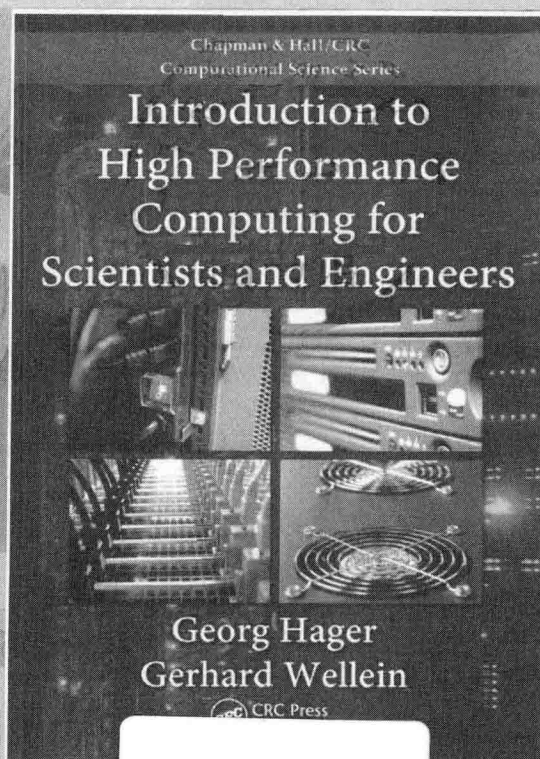


计 算 机 科 学 丛

高性能科学与工程计算

(德) Georg Hager Gerhard Wellein 著
张云泉 袁良 贾海鹏 张先轶 译

Introduction to High Performance Computing
for Scientists and Engineers



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

高性能科学与工程计算 / (德) 海格 (Hager, G.), (德) 韦雷因 (Wellein, G.) 著; 张云泉等译. —北京: 机械工业出版社, 2014.6

(计算机科学丛书)

书名原文: Introduction to High Performance Computing for Scientists and Engineers

ISBN 978-7-111-46652-9

I. 高… II. ①海… ②韦… ③张… III. 工程计算 IV. TB115

中国版本图书馆 CIP 数据核字 (2014) 第 093761 号

本书版权登记号: 图字: 01-2013-5984

Introduction to High Performance Computing for Scientists and Engineers, 1e, by Georg Hager and Gerhard Wellein (978-1-4398-1192-4).

Copyright © 2011 by Taylor & Francis Group, LLC.

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版, 并经过授权翻译出版。版权所有, 侵权必究。

本书中文简体字翻译版经机械工业出版社独家出版并限在中国大陆地区销售。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

本书封面贴有 Taylor & Francis 公司防伪标签, 无标签者不得销售。

本书主要面向科学家、研究者和工程师, 采取一种循序渐进的方式介绍高性能计算的基础知识, 即使不熟悉计算科学的读者也非常容易理解。本书主要覆盖了当代处理器体系结构基础知识、各种基本的程序串行优化策略和基本并行模式。此外, 在并行编程模型方面, 还包括基于 OpenMP 的共享并行和基于 MPI 的分布式并行编程, 并包含大量 Fortran 代码实例, 方便读者学习理解。总之, 本书以一种独特的视角介绍高性能计算知识, 适合各应用领域的专家和学生研读。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 朱 劼 肖晓慧

责任校对: 董纪丽

印刷: 北京瑞德印刷有限公司

版次: 2014 年 7 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 16

书号: ISBN 978-7-111-46652-9

定价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅开辟了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

译者序

Introduction to High Performance Computing for Scientists and Engineers

以高性能计算为核心的计算科学，已经广泛应用于科学研究和工程设计的各个层面，成为继理论研究和实验研究之外的第三种科学研究方法。美国总统信息技术顾问委员会在 2005 年发表题为“计算科学：确保美国竞争力”的报告中认为计算科学是 21 世纪最重要的技术领域之一，它对于整个社会的进步是不可或缺的，并建议美国政府制定规划，对计算科学进行长期资助。

高性能计算追求程序执行效率，因此并行程序的实现与优化成为计算机科学家的两大研究主题。本书也以并行编程模型和并行优化为主要内容，对程序串行优化、数据优化和并行性优化做了深入介绍，还对多处理器导致的 NUMA 问题以及多核环境下的进程亲缘性进行了深入讲解。在并行编程模型方面，OpenMP 和 MPI 分别是共享存储系统和分布式存储系统两大并行计算机体系架构上的事实标准。本书不仅详细介绍了这两种编程语言，并且还单独设立章节讲解了相应的关键优化技术。此外，本书配备大量程序和命令实例，方便读者掌握相关技巧。总之，本书作者通过自己多年的高性能计算领域编程和优化的实践经验，选取最为关键的基本知识和优化技术进行讲解，可供相关领域科学家作为参考。

本书作者 Georg Hager 和 Gerhard Wellein 均为物理学家，长期在 Erlangen Regional Computing Center (RRZE) 高性能计算组从事高性能计算研究，Gerhard Wellein 目前还担任该小组主任，主要研究兴趣包括现代微处理器的体系结构相关优化方法、处理器和系统层次的性能模型，以及异构系统优化。两位作者在高性能计算领域都有丰富的全方位用户支持和教学经验，例如讲座、报告、培训、代码并行化、代码分析和优化以及新颖计算机体系结构和工具的评估，其教学对象包括计算科学和工程编程领域的学生和科学家。

由于时间仓促，加上书中某些术语目前没有统一译法，所以我们对一些术语采取了保留其英文名称的方法。对书中翻译方面的错误和不妥之处，恳请广大读者不吝批评指正。

Georg Hager 和 Gerhard Wellein 在本书中为科学家和工程师循序渐进地介绍了高性能计算知识。本书的风格和讲解方法十分易读并容易理解。

二十多年前提出的计算建模和模拟的概念已经成为理论研究和实验研究之外的第三种科学研究方法，软件、数学模型和算法是科学计算领域的关键核心。并行硬件的发展十分迅猛，特别是以指数速度增长的处理器性能以及处理器体系结构和超级计算机系统设计的研 究。当计算建模和模拟成为科学研究的第三种方法后，复杂软件及其生态系统将会处于本研究领域的核心地位。

在应用层，科学必须展现为数学模型，并进一步表达为算法和对应的软件代码。相应地，大部分科学基金也在转向这样的项目，即需要领域科学家、计算机科学家及应用数学家协同合作，从原始科学构想到最终可执行软件的研究。这样的项目也需要数学库、协议和系统软件等这些需要花费数年开发并需要持久维护的大规模底层架构的支持，这些软件通常比最初设计的硬件平台甚至是设计和开发人员存活更长时间。

本书覆盖了当代处理器体系结构的基础知识，以及为科学计算程序有效地利用硬件特性进行串行优化的基本技术。作者讨论了数据移动中的关键问题并辅以实例，并以简单易读的方式介绍了高性能计算中的重要问题。书中还讨论了共享存储、非一致访问以及分布式存储的并行化方法。除此之外，还重点介绍了常用的并行编程模型，例如 OpenMP、MPI 以及混合编程方法。

我们生活在一个高效利用超级计算机系统进行大规模高性能计算的年代。本书对并行理论、优化技术、体系结构和现代高性能计算系统软件等方面进行了介绍，特别是对科学和工程问题的关注使得本书成为一本独特的教材，因此我强烈向科学家和工程师推荐此书，本书可以作为一本优秀的参考书，我相信本书会令大多数读者受益。

Jack Dongarra
美国田纳西大学

当 1941 年构建世界上第一台全自动可编程并具备二进制浮点运算能力的计算机时 [H129], Konrad Zuse 成功地预见了这种革命性设备不仅只应用于科学和工程领域, 还将对生活的各个方面产生深远影响 [H130]。计算机的高效运算、可视化和数据处理能力, 允许我们自动执行大量任务并实现无延迟通信, 这使得 Zuse 的预言成为现实, 计算已经彻底地改变了我们的生活方式和科研手段。

科学和工程研究从另一种特别的角度受益于计算能力的增长。早期的科研人员已经意识到计算机可以通过虚拟实验代替现实中那些太过复杂、昂贵或极端危险的真实实验, 或者进行以前无法手工完成的计算, 计算流体力学 (Computational Fluid Dynamics, CFD), 即模拟任意几何形状的流体就是一个这样典型的应用。飞机、汽车、高速列车以及涡轮设备的设计都离不开流体力学分析。以前的风洞和木质实体模型属于实验研究, 而在包括流体力学在内的几乎所有科学研究领域, 计算已成为理论分析和实验设计之外的第三种科学研究方法。近几年药物设计已成为高性能计算的新兴研究方向, 化学家可以利用软件 (只需点击鼠标) 快速发现化学反应机制, 模拟影响生命运行机制的大分子间的复杂动力学。理论固体物理通过在量子级别对组成成分、原子核和电子的相互作用进行建模来在微观级别研究物质结构 [A79], 需要的大量自由度使其无法在理论层次上进行分析, 而只能借助大规模计算资源实现。其他同样需要大规模计算的学科包括量子化学、物质科学、结构力学和医学图像处理等。

计算机模拟已成为学术和工业界大多数研究领域不可或缺的标准工具, 虽然科研人员可以利用个人计算机处理许多计算任务, 但是仍有一些任务需要个人计算机无法满足的大规模存储、内存以及计算速度, 而这正是高性能并行计算机系统的用武之地。

利用高性能计算 (High Performance Computing, HPC) 作为科研工具至少需要了解相关硬件和软件知识, 即使利用具有良好界面可直接运行的软件也是如此, 当需要编写代码时掌握这些知识就显得更为必要。但是我们与科学家和工程师多年的合作经验表明, 很难在不同研究组中建立和维护充足、完整的相关知识。陡峭的高性能计算学习曲线使得新的博士生难以独自快速掌握相关知识。虽然高性能计算具有基础性、难以掌握且极为昂贵等特点, 但是高性能计算毕竟只是一个工具, 而推进科学研究的进展才是终极目标。技术的进步使得高性能计算从院系级别扩展到了桌面级别, 在当前单处理器发展停滞以及通过增加并行性来提高性能的趋势下, 大量科学家和工程师必须关注性能和可扩展性, 这也是本书的主题。我们编写此书的目的为保持这些知识的新颖性。

事实上, 目前已经出版了多种计算机体系结构、优化和高性能计算方面的书籍 [S1, R34, S2, S3, S4], 虽然基本原理是一致的, 但是由于向量处理器的衰落、无处不在的 SIMD 处理能力、多核处理器的出现、ccNUMA 存储结构的发展以及高效能高性能网络互连系统的出现, 这些书中的许多内容都已过时。发展最为迅猛的要数运行 Linux 操作系统的基于 x86 结构的 Intel 或 AMD 处理器商业集群。最新的出版物更关注某个特定方面, 因此不适合作为学生教材或者相关科学家参考。本书的目标是从性能角度介绍体系结构和高性能计算编程的基础知识。我们的经验表明, 大多数用户经常无法定位性能因素以及是否需要考虑优化。本书的读者可以掌握如何确定性能限制因素等基础知识, 这为他们学习更为专业的技巧奠定了基础。因此本书也提供了一个详尽的参考书目列表, 可以在本书的网页中 (<http://>

www.hpc.rre.uni-erlangen.de/HPC4SE) 找到相应的具有超链接和附加评论的版本。

本书读者

在科学计算中心的多年工作经验使我们熟知用户和并行计算机厂商的需求，因此从事与高性能计算相关工作的人员都会从本书中获益。计算机科学、计算工程或更广泛的关心模拟领域的教师和学生可将本书作为教材。对于希望快速了解高性能计算基础知识的科学家和工程师，本书可以作为入门级参考书。最后，集群系统构建师可以利用本书的内容优化机器，为用户提供更好的服务。读者需要具有一定的编程和高级计算机体系结构知识。需要强调的是，本书只是一本入门级教材而不是一本详尽的参考书，因为到目前还没有一本高性能计算百科全书。

本书内容

高性能计算涉及给定算法（或者称为程序代码）的实现以及所运行的硬件平台。我们假定希望利用高性能计算资源的用户已经充分了解他们想要解决的问题的不同算法，因此我们并不全面介绍它们。本书中我们选取某些特定实例进行讲解，但是可能存在其他更为合适的算法，读者需要理解本书实例中的策略。

虽然我们努力保持本书的精简性，但是不可避免地包含了过多的内容。由于技术发展极为迅猛，我们忽略了一些近期的研究成果，例如现代加速技术（GPGPU、FPGA、Cell 处理器），因为这些内容很快将会变得过时。有些人认为高性能输入、输出技术应该包含在高性能计算书籍中，但是我们认为高效并行 I/O 技术是一个高级并且与硬件系统相关的课题，所以最好单独介绍。在软件方面，我们介绍了基本的程序串行优化策略和基本并行模式：基于 OpenMP 的共享存储并行化、基于 MPI 的分布式存储并行编程。其他一些模式，包括 Unified Parallel C (UPC 语言)、Co-Array Fortran (CAF 语言) 和其他一些现代编程方法，仍然需要检验其高效性，就像被广泛接受的 MPI 和 OpenMP 语言一样。

虽然我们 cannot 忽视商业系统的统治性地位，但本书中许多概念都在与特定的体系结构无关的层次上进行介绍，因此当我们应用实例介绍并展示实际性能数值时，都是基于 x86 集群系统标准网络连接。几乎所有的代码都用 Fortran 编写，仅当需要与外界环境相关的特性时我们才用 C 或者 C++ 代码。本书中一些用来测试的代码可以在本书官方网站上下载：<http://www.hpc.rre.unierlangen.de/HPC4SE>。

本书按下述方式组织：第 1 章介绍现代基于高速缓存的微处理器体系结构，并讨论其内在性能瓶颈，对最新的发展（例如多核芯片和同时多线程（SMT））也进行了介绍。虽然向量处理器已不再在高性能计算市场中应用，但是我们还是对其进行了简要介绍。第 2 章和第 3 章讲解了基于高速缓存体系结构的串程序的一般优化方法，介绍了用来寻找循环程序最优性能的一些简单的模型，还展示了如何通过代码变换来改进其性能限制。实际上，我们认为在不同计算机系统层次上建立应用程序的性能模型是最为重要的工作，也是高性能计算领域中的引导性原理。

第 4 章介绍共享存储和分布式存储两种类型的并行计算机体系结构，以及相关网络拓扑结构。第 5 章在理论层次上介绍并行计算概念，首先介绍一些重要的并行编程模式，然后介绍能够解释并行可扩展性限制的性能模型，因此回答了利用低速处理器构建大规模并行计算机系统的原因和方式。第 6 章对 OpenMP 进行了介绍，OpenMP 是在共享存储系统上编写

科学计算应用的主流语言。第 7 章介绍了几种典型的与 OpenMP 相关的程序性能问题并给出了相应的避免或减轻问题的方法。由于缓存一致的非一致内存访问 (ccNUMA) 系统已经大规模应用在高性能计算的市场中 (这也是被忽视的事实, 甚至一些高性能计算专家也将其忽视), 相应的优化技术在第 8 章介绍。第 9 章和第 10 章介绍分布式系统上的消息传递接口语言 (Message Passing Interface, MPI) 以及如何编写高效 MPI 代码。最后, 第 11 章介绍 MPI 和 OpenMP 混合编程方法。每章最后都配备相应的习题, 我们推荐读者认真研究, 这些习题包括一些书中未介绍的零碎知识或者一些特定的主题。附录 B 提供了相应的答案。

我们建议读者从头到尾认真阅读本书, 因为本书的主题都极为重要。但是只想了解 OpenMP 和 MPI 的读者可以只从第 6 章和第 9 章开始阅读, 并从第 7 章、第 8 章和第 10 章中学习相应的优化技术。本书自引用比较多, 所以如果忽略了某些部分, 也可以在其他地方找到相关材料。

致谢

本书起源于为 Springer 出版社 “Lecture Notes in Physics” 卷编写的两章内容, 也即 2006 年计算多粒子物理夏季讨论班的内容 [A79]。我们感谢讨论班的组织者, 特别是 Holger Fehske、Ralf Schneider 和 Alexander Weisse, 使得我们有机会将自己在高性能计算领域的经验总结成文, 虽然本书增加了很多其他材料, 但是没有最初的两章内容我们就无法开始本书的撰写。

本书包括与用户、学生、算法、代码和工具一起工作的十几年经验, 因此许多人都直接或者间接 (甚至没有意识到) 为本书做出了贡献。我们特别要感谢 Erlangen Regional Computing Center (RRZE) 高性能计算服务中心的员工, 特别是 Thomas Zeiser、Jan Treibig、Michael Meier、Markus Wittmann、Johannes Habich、Gerald Schubert 和 Holger Stengel, 在与他们的讨论中得到了十分有用的知识。在十年中我们小组从 “Competence Network for Scientific High Performance Computing in Bavaria” (KONWIHR) 和弗里德里希 - 亚历山大 - 埃尔兰根 - 纽伦堡大学获得了极大的资助。

我们还要感谢 Uwe Küster (HLRS Stuttgart)、Matthias Müller (ZIH Dresden)、Reinhold Bader 和 Matthias Brehm (LRZ München), 感谢他们在服务中心与我们的高效合作。还要特别感谢 Darren Kerbyson (PNNL) 的鼓励和对我们的建议。最后我们要感谢 Rolf Rabenseifner (HLRS) 和 Gabriele Jost (TACC) 与我们在混合编程方面的合作, 本书第 11 章就是我们一起合作的成果。

一些公司的第一手技术支持及一些合作 (即使没有利润) 也值得我们感谢, Intel (Andrey Semin 和 Herbert Cornelius)、SGI (Reiner Vogelsang 和 Rüdiger Wolff)、NEC (Thomas Schönmeyer)、Sun Microsystems (Rick Hetherington、Ram Kunda 和 Constantin Gonzalez)、IBM (Klaus Gottschalk) 以及 Cray (Wilfried Oed)。

我们还要感谢 CRC 员工的支持, 特别是 Ari Silver、Karen Simon、Katy Smith 和 Kevin Craig。最后, 如果没有 Horst Simon (LBNL/NERSC) 和 Randi Cohen (Taylor & Francis) 的鼓励, 本书将不可能顺利完成, 他们使我们有动力开始撰写本书。

Georg Hager 与 Gerhard Wellein

埃尔兰根计算中心

埃尔兰根 - 纽伦堡大学

德国

ASCII (American Standard Code For Information Interchange) 美国标准信息交换代码

ASIC (Application-Specific Integrated Circuit) 专用集成电路

BIOS (Basic Input/Output System) 基本输入 / 输出系统

BLAS (Basic Linear Algebra Subroutines) 基本线性代数子程序

CAF (Co-Array Fortran) co-array Fortran

ccNUMA (cache-coherent NonUniform Memory Access) 缓存一致的非一致内存访问

CFD (Computational Fluid Dynamics) 计算流体力学

CISC (Complex Instruction Set Computer) 复杂指令集计算机

CL (Cache Line) 高速缓存行

CPI (Cycles Per Instruction) 指令周期

CPU (Central Processing Unit) 中央处理器

CRS (Compressed Row Storage) 压缩行存储

DDR (Double Data Rate) 双倍速率

DMA (Direct Memory Access) 直接内存访问

DP (Double Precision) 双精度

DRAM (Dynamic Random Access Memory) 动态随机存取内存

ED (Exact Diagonalization) 精确对角化

EPIC (Explicitly Parallel Instruction Computing) 显式并行指令计算

Flop (Floating-point operation) 浮点运算

FMA (Fused Multiply-Add) 混合乘法

FP (Floating Point) 浮点

FPGA (Field-Programmable Gate Array) 现场可编程逻辑门阵列

FS (File System) 文件系统

FSB (FrontSide Bus) 前端总线

GCC (GNU Compiler Collection) GNU 编译器集合

GE (Gigabit Ethernet) 千兆以太网

GigE (Gigabit Ethernet) 千兆以太网

GNU (GNU is not UNIX) GNU

GPU (Graphics Processing Unit) 图形处理单元

GUI (Graphical User Interface) 图形用户界面

HPC (High Performance Computing) 高性能计算

HPF (High Performance Fortran) 高性能 Fortran 语言

HT (HyperTransport) 超传输

IB (InfiniBand) infiniband

ILP (Instruction-Level Parallelism) 指令级并行

IMB (Intel Mpi Benchmarks) 英特尔 MPI 基准测试

I/O (Input/Output) 输入 / 输出

IP (Internet Protocol) 互联网协议

JDS (Jagged Diagonals Storage) 锯齿形对角线存储格式

L1D (Level 1 Data cache) 1 级数据缓存

L1I (Level 1 Instruction cache) 1 级指令缓存

L2 (Level 2 cache) 2 级缓存

L3 (Level 3 cache) 3 级缓存

LD (Locality Domain) 局部域

LD (Load) 加载

LIKWID (Like I Knew What I am Doing) 就像我知道我在做什么

LRU (Least Recently Used) 最近最少使用策略

LUP (Lattice site UPdate) 网格更新

MC (Monte Carlo) 蒙特卡罗方法

MESI (Modified/Exclusive/Shared/Invalid) 修改 / 专有 / 共享 / 无效 (缓存一致性协议状态)

MI (Memory Interface) 存储器接口

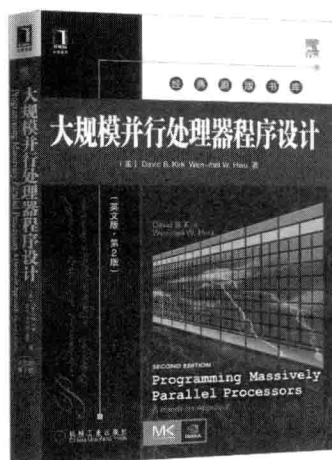
MIMD (Multiple Instruction Multiple Data) 多指令流多数据流

MIPS (Million Instructions Per Second) 每秒百万指令

MMM (Matrix-Matrix Multiplication) 矩阵矩阵乘

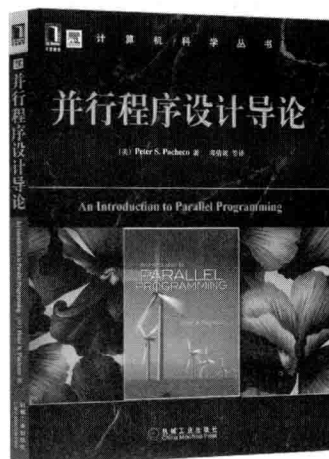
- MPI (Message Passing Interface) 消息传递接口
- MPMD (MultiPle Program Multiple Data) 多程序流多数据流
- MPP (Massively Parallel Processing) 大规模并行处理
- MVM (Matrix-Vector Multiplication) 矩阵向量乘
- NORMA (NO Remote Memory Access) 非远程存储访问
- NRU (Not Recently Used) 最近未使用算法
- NUMA (NonUniform Memory Access) 非一致内存访问
- OLC (Outer-Level Cache) 外级缓存
- OS (Operating System) 操作系统
- PAPI (Performance Application Programming Interface) 应用程序性能编程接口
- PC (Personal Computer) 个人电脑
- PCI (Peripheral Component Interconnect) 外围组件互连
- PDE (Partial Differential Equation) 偏微分方程
- PGAS (Partitioned Global Address Space) 分割全局地址空间
- PLPA (Portable Linux Processor Affinity) 可移植 Linux 处理器亲缘性
- POSIX (Portable Operating System Interface for uniX) 可移植操作系统接口 (UNIX)
- PPP (Pipeline Parallel Processing) 流水线并行处理
- PVM (Parallel Virtual Machine) 并行虚拟机
- QDR (Quad Data Rate) 4 倍数据倍率
- QPI (Quick Path Interconnect) 快速路径互连
- RAM (Random Access Memory) 随机访问内存
- RISC (Reduced Instruction Set Computer) 精简指令集计算机
- RHS (Right Hand Side) 右端项
- RFO (Read For Ownership) 所有者处理
- SDR (Single Data Rate) 单倍数据速率
- SIMD (Single Instruction Multiple Data) 单指令流多数据流
- SISD (Single Instruction Single Data) 单指令流单数据流
- SMP (Symmetric MultiProcessing) 对称多处理
- SMT (Simultaneous MultiThreading) 同步多线程
- SP (Single Precision) 单精度
- SPMD (Single Program Multiple Data) 单程序流多数据流
- SSE (Streaming SIMD Extensions) SIMD 流指令扩展
- ST (STore) 存储
- STL (Standard Template Library) 标准模板库
- SYSV (UNIX System V) UNIX 系统 V
- TBB (Threading Building Blocks) TBB 语言
- TCP (Transmission Control Protocol) 传输控制协议
- TLB (Translation Lookaside Buffer) 旁路缓冲器
- UMA (Uniform Memory Access) 一致内存访问
- UPC (Unified Parallel C) UPC 语言

推荐阅读



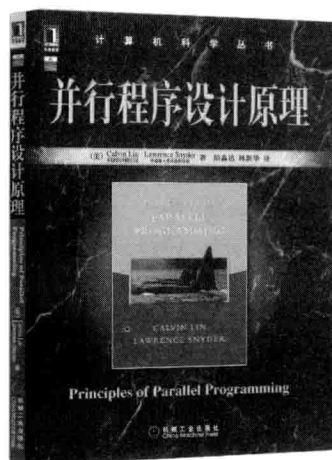
大规模并行处理器程序设计 (英文版·第2版)

作者: David B. Kirk 等 ISBN: 978-7-111-41629-6 定价: 79.00元



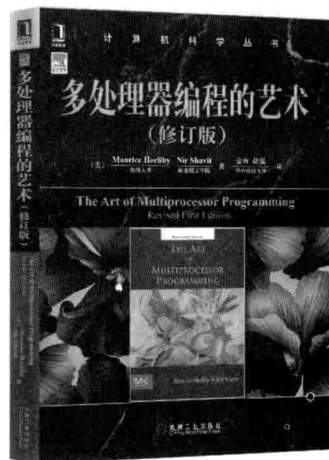
并行程序设计导论

作者: Peter S. Pacheco ISBN: 978-7-111-39284-2 定价: 49.00元



并行程序设计原理

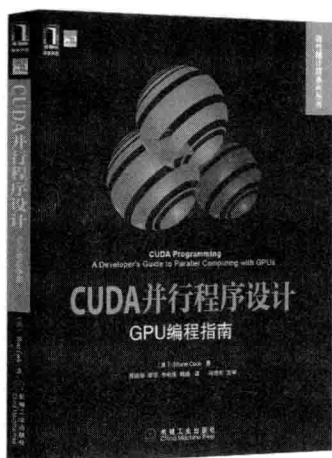
作者: Larry Snyder 等 ISBN: 978-7-111-27075-1 定价: 45.00元



多处理器编程的艺术 (修订版)

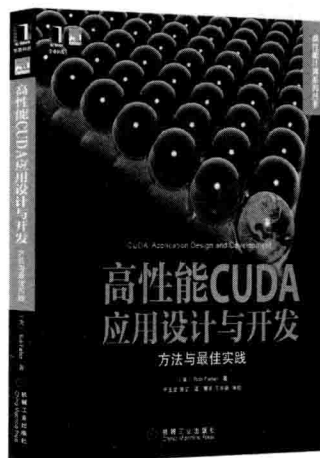
作者: Maurice Herlihy 等 ISBN: 978-7-111-41858-0 定价: 69.00元

推荐阅读



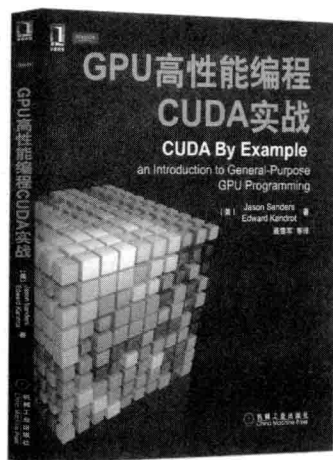
CUDA并程序序设计：GPU编程指南

作者：Shane Cook ISBN: 978-7-111-44861-7 定价：99.00元



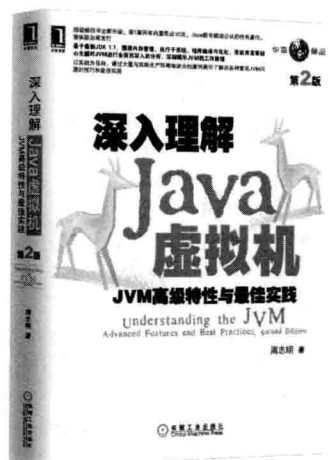
高性能CUDA应用设计与开发：方法与最佳实践

作者：Rob Farber ISBN: 978-7-111-40446-0 定价：59.00元



GPU高性能编程CUDA实战

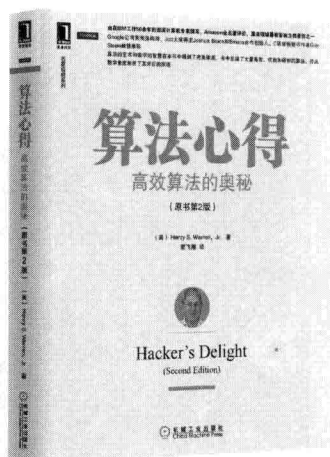
作者：Jason Sanders 等 ISBN: 978-7-111-32679-3 定价：39.00元



深入理解Java虚拟机：JVM高级特性与最佳实践（第2版）

作者：周志明 ISBN: 978-7-111-42190-0 定价：79.00元

推荐阅读



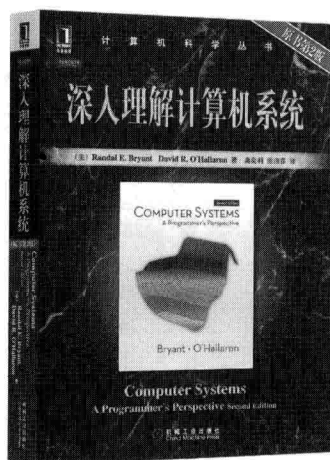
算法心得：高效算法的奥秘（原书第2版）

作者：Henry S. Warren ISBN: 978-7-111-45356-7 定价：89.00元



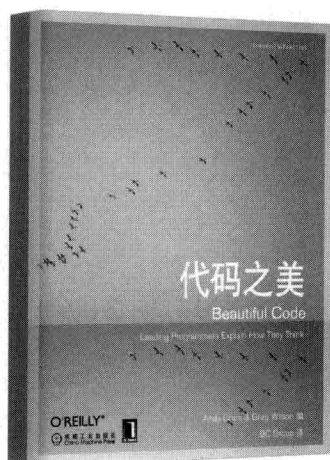
算法导论（原书第3版）

作者：Thomas H. Cormen 等 ISBN: 978-7-111-40701-0 定价：128.00元



深入理解计算机系统（原书第2版）

作者：Randal E. Bryant 等 ISBN: 978-7-111-32133-0 定价：99.00元



代码之美

作者：Grey Wilson ISBN: 978-7-111-25133-0 定价：99.00元

出版者的话	2.2.1 少做工作	32
译者序	2.2.2 避免耗时运算	32
序	2.2.3 缩减工作集	33
前言	2.3 小方法，大改进	33
缩略词	2.3.1 消除常用子表达式	33
	2.3.2 避免分支	34
	2.3.3 使用SIMD指令集	34
第1章 当代处理器	2.4 编译器作用	36
1.1 存储程序的计算机体系结构	2.4.1 通用优化选项	37
1.2 基于高速缓存的通用微处理器	2.4.2 内联	37
体系结构	2.4.3 别名	37
1.2.1 性能指标和基准测试	2.4.4 计算准确性	38
1.2.2 晶体管：摩尔定律	2.4.5 寄存器优化	39
1.2.3 流水线	2.4.6 利用编译日志	39
1.2.4 超标量	2.5 C++优化	40
1.2.5 SIMD	2.5.1 临时变量	40
1.3 存储层次	2.5.2 动态内存管理	42
1.3.1 高速缓存	2.5.3 循环与迭代器	43
1.3.2 高速缓存映射	习题	43
1.3.3 预取	第3章 数据访存优化	45
1.4 多核处理器	3.1 平衡分析与lightspeed评估	45
1.5 多线程处理器	3.1.1 基于带宽的性能建模	45
1.6 向量处理器	3.1.2 STREAM 基准测试	47
1.6.1 设计原理	3.2 存储顺序	49
1.6.2 最高性能估计	3.3 案例分析：Jacobi算法	50
1.6.3 程序设计	3.4 案例分析：稠密矩阵转置	53
习题	3.5 算法分类和访存优化	56
第2章 串行代码基本优化技术	3.5.1 $O(N)/O(N)$	56
2.1 标量剖析	3.5.2 $O(N^2)/O(N^2)$	57
2.1.1 基于函数和代码行的程序剖析	3.5.3 $O(N^3)/O(N^2)$	60
2.1.2 硬件性能计数器	3.6 案例分析：稀疏矩阵向量乘	61
2.1.3 手工代码插入	3.6.1 稀疏矩阵的存储机制	62
2.2 优化常识	3.6.2 JDS sMVM优化	64

习题	66	6.1.4 同步	107
第4章 并行计算机	68	6.1.5 归约	108
4.1 并行计算模式分类	69	6.1.6 循环调度	109
4.2 共享存储计算机	69	6.1.7 任务	110
4.2.1 cache一致性	69	6.1.8 其他方面	111
4.2.2 UMA	71	6.2 案例分析: OpenMP并行 实现Jacobi算法	112
4.2.3 ccNUMA	71	6.3 高级OpenMP: 波前并行化	114
4.3 分布式存储计算机	73	习题	116
4.4 混合型系统	74	第7章 高效OpenMP编程	119
4.5 网络	75	7.1 OpenMP程序性能分析	119
4.5.1 网络的基本性能特征	75	7.2 性能缺陷	120
4.5.2 总线	78	7.2.1 减轻Open MP共享区开销	121
4.5.3 交换网络和胖树网络	79	7.2.2 决定短循环的OpenMP开销	126
4.5.4 Mesh 网络	81	7.2.3 串行化	128
4.5.5 混合网络	82	7.2.4 伪共享	129
习题	82	7.3 案例分析: 并行稀疏矩阵向 量乘	130
第5章 并行性基础	83	习题	133
5.1 为什么并行化	83	第8章 ccNUMA体系结构的局部性 优化	134
5.2 并行性	83	8.1 ccNUMA的局部访问	134
5.2.1 数据并行性	84	8.1.1 首次访问方式分配页面	135
5.2.2 功能并行性	86	8.1.2 通过其他方式的局部性 访问	137
5.3 并行扩展性	87	8.2 案例分析: 稀疏MVM的 ccNUMA优化	138
5.3.1 限制并行执行的因素	87	8.3 页面布局缺陷	139
5.3.2 可扩展性指标	88	8.3.1 非NUMA友好的OpenMP 调度	139
5.3.3 简单可扩展性定律	89	8.3.2 文件系统高速缓存	140
5.3.4 并行效率	90	8.4 C++中的ccNUMA问题	142
5.3.5 串行性能与强可扩展性	91	8.4.1 对象数组	142
5.3.6 改进的性能模型	92	8.4.2 标准模板库	144
5.3.7 选择正确的扩展性基准	94	习题	146
5.3.8 案例分析: 低速处理器 计算机能否变得更快	95	第9章 使用MPI进行分布式存储并行 内存编程	147
5.3.9 负载不均衡	98	9.1 消息传递	147
习题	101	9.2 MPI简介	148
第6章 使用OpenMP进行共享存储 并行编程	103		
6.1 OpenMP简介	103		
6.1.1 并行执行	103		
6.1.2 数据作用域	105		
6.1.3 循环的OpenMP工作共享	106		

9.2.1 一个简单例子	148	11.1.2 任务模式实现	191
9.2.2 消息和点对点通信	150	11.1.3 案例分析: 混合Jacobi解 法器	192
9.2.3 集合通信	154	11.2 MPI线程交互分类	193
9.2.4 非阻塞点对点通信	157	11.3 混合分解及映射	195
9.2.5 虚拟拓扑	160	11.3.1 每个节点一个MPI进程	195
9.3 实例: Jacobi解法器的MPI 并行	162	11.3.2 每个插槽一个MPI进程	196
9.3.1 MPI实现	162	11.3.3 每个插槽多个MPI进程	196
9.3.2 性能特征	167	11.4 混合编程的优势和劣势	197
习题	170	11.4.1 改善的收敛速度	197
第10章 高效MPI编程	171	11.4.2 共享高速缓存中的数据 重用	197
10.1 MPI性能工具	171	11.4.3 利用额外级别的并行性	198
10.2 通信参数	174	11.4.4 重叠MPI通信和计算	198
10.3 同步、串行化和竞争	174	11.4.5 减少MPI开销	198
10.3.1 隐式串行化和同步	174	11.4.6 多级别开销	198
10.3.2 竞争	176	11.4.7 向量模式下批量同步 通信	198
10.4 降低通信开销	177	附录A 多核环境中的拓扑和亲缘性	199
10.4.1 最优化区域分解	177	A.1 拓扑	200
10.4.2 聚合消息	180	A.2 线程和进程分布	201
10.4.3 非阻塞与异步通信	181	A.2.1 外部亲缘性工具	201
10.4.4 集合通信	183	A.2.2 程序控制亲缘性	203
10.5 理解节点内点对点通信	184	A.3 非页面首次访问分配策略	204
习题	189	附录B 习题解答	206
第11章 MPI与OpenMP混合编程	190	参考文献	221
11.1 基本MPI/OpenMP混合编程 模型	190	索引	232
11.1.1 向量模式实现	191		