

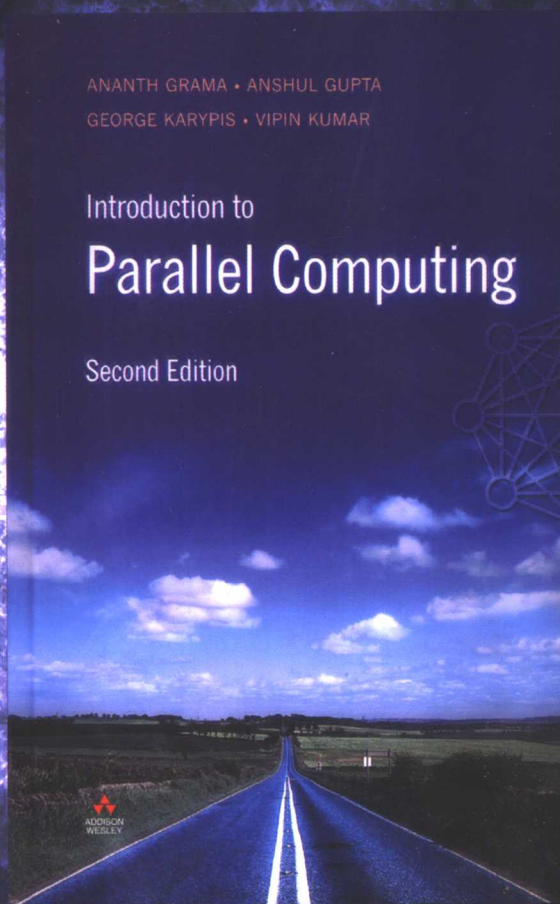
原书第2版

计 算 机 科 学 丛 书



# 并行计算导论

(美) Ananth Grama Anshul Gupta 著 张武 毛国勇 程海英 等译 李伯民 审校  
George Karypis Vipin Kumar



Introduction to Parallel Computing  
Second Edition



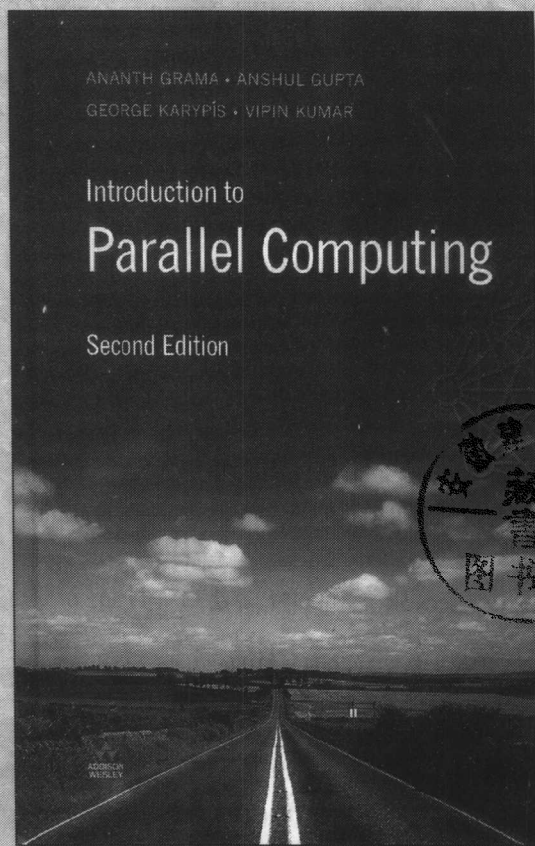
机械工业出版社  
China Machine Press

计 算 机 科 学 丛 书

原书第2版

# 并行计算导论

(美) Ananth Grama Anshul Gupta 著 张武 毛国勇 程海英 等译 李伯民 审校  
George Karypis Vipin Kumar



**Introduction to Parallel Computing**  
Second Edition



机械工业出版社  
China Machine Press

本书全面介绍并行计算的各个方面,包括体系结构、编程范型、算法和标准等,涉及并行计算中的新技术,也覆盖了较传统的算法,如排序、搜索、图和动态编程等。本书尽可能采用与底层平台无关的体系结构并且针对抽象模型来设计算法。书中选择MPI、POSIX线程和OpenMP作为编程模型,并在不同例子中反映了并行计算的不断变化的应用组合。

本书论述清晰,示例生动,并附有大量习题。适合作为高等院校计算机及相关专业本科生和研究生的教材或参考书。

Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: *Introduction to Parallel Computing (Second Edition)* (ISBN:0-201-64865-2).

Copyright © 2003 by Pearson Education Limited.

This translation of *Introduction to Parallel Computing (Second Edition)* (ISBN:0-201-64865-2) is published by arrangement with Pearson Education Limited.

本书中文简体字版由英国Pearson Education培生教育出版集团授权出版。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2003-5007

图书在版编目(CIP)数据

并行计算导论(原书第2版)/(美)格兰马(Grama, M.)等著;张武等译.-北京:机械工业出版社,2005.1

(计算机科学丛书)

书名原文:Introduction to Parallel Computing (Second Edition)

ISBN 7-111-14985-8

I. 并… II. ①格… ②张… III. 并行算法 IV. TP301.6

中国版本图书馆CIP数据核字(2004)第076848号

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:朱起飞

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2005年1月第1版第1次印刷

787mm×1092mm 1/16·28印张

印数:0 001-4000册

定价:49.00元

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

# 出版者的话

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，不仅筹划了研究的范畴，还揭开了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及度藏。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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业



的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

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,但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。教材的出版只是我们的后续服务的起点。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,我们的联系方法如下:

电子邮件: [hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范 明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅 宏	程 旭	程时端
谢希仁	裘宗燕	戴 葵		

## 秘 书 组

武卫东      温莉芳      刘 江      杨海玲

# 中文版序

我很高兴《并行计算导论》一书终于有了它的中文译本。在此，恭贺并感谢上海大学计算机学院张武教授承担了这项巨大的工程，以将此书呈献给了广大的中国读者。

并行计算是一个充满活力的领域，经过几十年的发展，这一领域的研究成果已在科学与技术的众多领域随处可见。现在，超级并行计算机在大规模科学与工程计算、电子商务、网络搜索、数据挖掘等领域都得到了广泛的应用。同时，利用现有的软硬件和网络技术制造机群式超级计算机也已变得越来越方便。此外，所有的超级计算机供应商都可为客户提供专用的超级并行计算机。重要的问题是如何充分利用这些超级并行计算机，这需要发展有效的并行算法和以适当的程序范例实施这些算法，这就是本书所讨论的重点。

我衷心希望本书能为读者在学习和应用并行计算这一重要的和不断发展着的最新技术的过程中提供有益的帮助。

Vipin Kumar

美国明尼苏达大学计算机科学与工程系教授

美国陆军高性能计算研究中心主任

Email: kumar@cs.umn.edu

URL: <http://www.cs.umn.edu/~kumar>

# 译者序

当代科学、技术和社会经济的发展对大规模科学与工程计算的需求是无止境的，诸如核反应模拟、数值天气预报、基因工程、城市交通、电子商务和网络搜索等问题都对计算提出了巨大的挑战。随着超级并行计算机的飞速发展，尤其是机群式超级计算机提供越来越方便的并行计算资源，如何充分利用这些并行计算机资源，已对并行计算及其应用构成严峻的考验。

《并行计算导论》的作者Vipin Kumar教授等是国际知名的并行计算专家，他们在并行算法设计与分析、并行计算应用等方面有很深的造诣。自本书第1版1994年出版到2003年出版第2版以来，已在世界范围内被广泛地采用为高等院校本科生和研究生的并行计算教材或参考书。本书系统介绍涉及并行计算的体系结构、编程范例、算法与应用和标准等。本书结构合理，可读性强，加之每章精心设计的习题集，形成了本书的主要特色。相信本书对培养国内急需的并行计算人才和推动并行计算课程建设与改革必将发挥积极的作用。

本书的前言、第1、8、9、13章由张武翻译，第2、10、11、12章由毛国勇翻译，第6、7章由程海英翻译，第3章由宋安平翻译，第4章由付朝江翻译，第5章由张衡翻译。全书由张武统稿。

本书主要作者Vipin Kumar教授应邀为本书中文版写了序，张武教授在美国伊利诺理工学院(IIT)计算机系工作期间，曾与Kumar教授有过交流与合作。美国IIT计算机系孙贤和教授、上海大学童维勤教授、支小莉博士和夏骄雄讲师等在本书的翻译过程中给予了帮助。

本书翻译期间正值全国抗击SARS，上海大学于2003年10月接受国家教育部“本科教学水平”评估。因此，翻译时间非常仓促。虽然在翻译过程中力求尊重原意和翻译准确，但由于水平有限及新术语的不断出现，不当和疏漏之处在所难免，敬请读者提出宝贵意见。

译者

上海大学计算机学院

2004年1月



# 前 言

自1994年本书第1版问世以来，并行计算领域发生了巨大的变化。十多年前，紧耦合可扩展的信息传递平台是并行计算的主流模式，而今，主导地位则由大量成本较低的工作站、多处理机工作站以及服务器构成的机群式计算平台所占据。在此期间，这些平台的程序设计模型也得以发展。尽管十多年前绝大部分的机器还依赖于特定的应用程序编程接口（API）来实现信息交换与基于循环的并行，而现在的模型已在不同平台上将API标准化。一些程序设计模型已被接受为标准，比如消息传递库PVM和MPI、POSIX线程库、基于指令的OpenMP等，并被移植到多种平台上。

十年前，流体力学、结构力学和信号处理等是并行计算应用的主要领域，并且这些应用仍对当前并行计算平台构成挑战。但是，今天许多新的应用也正变得越来越重要，其中包括事务处理、信息检索、数据挖掘与分析 and 多媒体服务等数据密集型应用。新兴应用领域，如计算生物学与纳米技术，对并行算法和系统开发也提出了巨大的挑战。在并行体系结构、程序设计模型与应用上的变化，也伴随着如何使用户以基于网格服务的形式利用并行计算平台方面的改变。

以上发展对并行算法的设计、分析与实现过程产生了深刻的影响。十年前，并行算法设计主要强调的是如何将任务精确地映射到如格网和超立方体这样的特定拓扑结构上，而今天却是从算法设计与实现的角度来强调可编程性与可移植性。为此，本书尽可能采用与底层平台无关的体系结构，并针对抽象模型来设计算法。关于程序设计模型，本书选择消息传递接口（Message-Passing Interface, MPI）、POSIX线程和OpenMP。对于涉及并行计算的应用组合，也反映在本书的各种例子中。

本书构成一门专注于并行计算的课程的基础，也可分为两部分讲授。以下是对按两部分讲授的建议：

- 1) 并行计算导论：第1~6章。这个课程提供并行算法设计和并行编程的基础。
- 2) 并行算法设计与分析：第2、3章以及第8~12章。这个课程对各种并行算法的设计与分析进行深入的探讨。

本书材料曾在明尼苏达大学和普度大学的并行算法与并行计算课程中试用过。这些课程是为计算机专业低年级研究生与高年级本科生开设的。另外，对学习计算密集型问题的科学与工程专业的研究生，在他们选修的与科学计算相关的课程中也试用过这些材料。

本书多数章包括：1) 例子与图解；2) 教材的补充习题，用于测试学生对所学内容的理解程度；3) 书目评注，用于帮助有兴趣的学生和研究人员了解相关的和更高级的课题。本书的索引帮助读者快速查找到感兴趣的术语。术语所在的页号使用黑体排印。内容相对比较复杂的节上加了“\*”号。此外，从出版商（<http://www.booksites.net/kumar>）那里可以获得更多支持信息。

与本书前一版一样，我们认为本书也在不断发展之中。要感谢那些对我们第1版书提出批

评、建议、问题和提供代码或者其他信息的读者，并真诚地希望围绕着这本新版书，我们之间还能继续这种交流。我们鼓励读者通过电子邮件 [book-vk@cs.umn.edu](mailto:book-vk@cs.umn.edu) 与我们交流。所有相关的读者反馈将在征得同意后加到<http://www.cs.umn.edu/~parbook>下的归档信息中。该网站还维护着本书的在线勘误表。我们相信，在像并行计算这个高度动态变化的领域，以这种健全的方式进行思想和资料的交流将使我们受益良多。

## 致 谢

首先要感谢我们各自的妻子Joanna、Rinku、Krista和Renu，如果没有她们的包容与理解，这项工作是不可能如期完成的。我们也要感谢我们各自的父母及其他家庭成员，Akash、Avi、Chethan、Eleni、Larry、Mary-Jo、Naina、Petros、Samir、Subhasish、Varun、Vibhav和Vipasha，感谢他们在本书写作期间给予作者的热情支持与鼓励。

我们各自所属的机构，普度大学计算机科学与计算研究所（CRI）、明尼苏达大学计算机科学与工程系、美国陆军高性能计算研究中心（AHPARC）和数字技术中心（DTC），以及位于Yorktown Heights的IBM T. J. Watson研究中心，所具备的计算资源和活跃的学术氛围为我们提供了极大帮助。

本项目源于我们的第1版书，因此要感谢所有在两个版本书的写作期间帮助过我们的人们。许多人以不同方式为这个项目做出了贡献。我们要感谢Ahmed Sameh一如既往的支持与鼓励，Dan Challou、Michael Heath、Dinesh Mehta、Tom Nurkkala、Paul Saylor和Shang-Hua Teng为本书的各个版本提供了有价值的输入工作。感谢明尼苏达大学与普度大学中选修了并行计算导论课程的学生们，他们找出了本书早期手稿中的各种错误。特别要提到的是Jim Diehl和Rasit Eskicioglu，他们极其耐心地为本书的手稿勘正了大量的错误。Ramesh Agarwal、David Bailey、Rupak Biswas、Jim Bottum、Thomas Downar、Rudolf Eigenmann、Sonia Fahmy、Greg Frederickson、John Gunnels、Fred Gustavson、Susanne Hambrusch、Bruce Hendrickson、Christoph Hoffmann、Kai Hwang、Ioannis Ioannidis、Chandrika Kamath、David Keyes、Mehmet Koyuturk、Piyush Mehrotra、Zhiyuan Li、Jens Palsberg、Voicu Popescu、Alex Pothen、Viktor Prasanna、Sanjay Ranka、Naren Ramakrishnan、Elisha Sacks、Vineet Singh、Sartaj Sahni、Vivek Sarin、Wojciech Szpankowski、Srikanth Thirumalai、Jan Vitek和David Yau为本书提供了宝贵的技术资料。与有合作精神和乐于助人的Pearson Education出版社的员工们一起工作是件非常愉快的事。特别地，我们要感谢Keith Mansfield与Mary Lince对本书的出版所进行的专业运作。

美国陆军研究实验室（The Army Research Laboratory, ARL）、陆军研究局（Army Research Office, ARO）、能源部（Department of Energy, DoE）、国家宇航局（National Aeronautics and Space Administration, NASA）、国家科学基金会（The National Science Foundation, NSF）为作者Ananth Grama、George Karypis与Vipin Kumar提供了并行计算研究的支持。我们还要特别感谢Kamal Abdali、Michael Coyle、Jagdish Chandra、Frederica Darema、Stephen Davis、Wm Randolph Franklin、Richard Hirsch、Charles Koelbel、Raju Namburu、N. Radhakrishnan、John Van Rosendale、Subhash Saini与Xiaodong Zhang在并行计算领域为我们的研究项目所提供的支持。感谢IBM公司的Andrew Conn、Brenda Dietrich、John Forrest、David Jensen与Bill Pulleyblank多年来对作者Anshul Gupta研究工作的一贯支持。

# 目 录

出版者的话	2.4.3 网络拓扑结构	24
专家指导委员会	2.4.4 静态互连网络评价	31
中文版序	2.4.5 动态互连网络评价	33
译者序	2.4.6 多处理器系统中的高速 缓存一致性	34
前言	2.5 并行计算机的通信成本	39
第1章 并行计算介绍	2.5.1 并行计算机的消息传递成本	39
1.1 推动并行化	2.5.2 共享地址空间计算机的通信成本	44
1.1.1 计算能力因素——从晶体管到 浮点运算速度	2.6 互连网络的路由选择机制	46
1.1.2 内存及磁盘速度的因素	2.7 进程-处理器映射的影响和映射技术	47
1.1.3 数据通信因素	2.7.1 图的映射技术	48
1.2 并行计算适用范围	2.7.2 成本-性能平衡	53
1.2.1 在工程及设计中的应用	2.8 书目评注	54
1.2.2 科学计算中的应用	习题	55
1.2.3 商业应用	第3章 并行算法设计原则	63
1.2.4 计算机系统中的应用	3.1 预备知识	63
1.3 本书的组织及内容	3.1.1 分解、任务与依赖图	63
1.4 书目评注	3.1.2 粒度、并发性与任务交互	65
习题	3.1.3 进程和映射	69
第2章 并行编程平台	3.1.4 进程与处理器	69
2.1 隐式并行:微处理器体系结构的 发展趋势*	3.2 分解技术	70
2.1.1 流水线与超标量执行	3.2.1 递归分解	70
2.1.2 超长指令字处理器	3.2.2 数据分解	72
2.2 内存系统性能的局限*	3.2.3 探测性分解	77
2.2.1 使用高速缓存改善有效内存延迟	3.2.4 推测性分解	79
2.2.2 内存带宽的影响	3.2.5 混合分解	80
2.2.3 躲避内存延迟的其他方法	3.3 任务和交互的特点	81
2.2.4 多线程与预取间的权衡	3.3.1 任务特性	81
2.3 并行计算平台剖析	3.3.2 任务间交互的特征	82
2.3.1 并行平台的控制结构	3.4 负载均衡的映射技术	84
2.3.2 并行平台的通信模型	3.4.1 静态映射方案	85
2.4 并行平台的物理组织	3.4.2 动态映射方案	95
2.4.1 理想并行计算机的体系结构	3.5 包含交互开销的方法	96
2.4.2 并行计算机互连网络	3.5.1 最大化数据本地性	97
	3.5.2 最小化争用与热点	98

3.5.3 使计算与交互重叠	98	4.8 小结	137
3.5.4 复制数据或计算	99	4.9 书目评注	138
3.5.5 使用最优聚合交互操作	100	习题	139
3.5.6 一些交互与另一些交互的重叠	100	第5章 并行程序的解析建模	143
3.6 并行算法模型	101	5.1 并行程序中的开销来源	143
3.6.1 数据并行模型	101	5.2 并行系统的性能度量	144
3.6.2 任务图模型	101	5.2.1 执行时间	144
3.6.3 工作池模型	102	5.2.2 总并行开销	144
3.6.4 主-从模型	102	5.2.3 加速比	144
3.6.5 流水线模型或生产者-消费者模型	103	5.2.4 效率	147
3.6.6 混合模型	103	5.2.5 成本	149
3.7 书目评注	103	5.3 粒度对性能的影响	149
习题	103	5.4 并行系统的可扩展性	152
第4章 基本通信操作	107	5.4.1 并行程序的扩展特性	153
4.1 一对多广播以及多对一归约	108	5.4.2 可扩展性的等效率度量	155
4.1.1 环或线性阵列	108	5.4.3 成本最优性和等效率函数	158
4.1.2 格网	110	5.4.4 等效率函数的下界	159
4.1.3 超立方体	111	5.4.5 并发度和等效率函数	159
4.1.4 平衡二叉树	111	5.5 最小执行时间和最小成本	
4.1.5 算法细节	112	最优执行时间	159
4.1.6 成本分析	114	5.6 并行程序渐近分析	162
4.2 多对多广播和归约	114	5.7 其他可扩展性的度量	162
4.2.1 线性阵列和环	115	5.8 书目评注	165
4.2.2 格网	117	习题	166
4.2.3 超立方体	117	第6章 使用消息传递模式编程	171
4.2.4 成本分析	120	6.1 消息传递编程的原理	171
4.3 全归约与前缀和操作	121	6.2 操作构件: 发送和接收操作	172
4.4 散发和收集	123	6.2.1 阻塞式消息传递操作	172
4.5 多对多私有通信	125	6.2.2 无阻塞式消息传递操作	175
4.5.1 环	126	6.3 MPI: 消息传递接口	176
4.5.2 格网	128	6.3.1 启动和终止MPI库	177
4.5.3 超立方体	128	6.3.2 通信器	177
4.6 循环移位	131	6.3.3 获取信息	178
4.6.1 格网	131	6.3.4 发送和接收消息	178
4.6.2 超立方体	133	6.3.5 实例: 奇偶排序	182
4.7 提高某些通信操作的速度	135	6.4 拓扑结构与嵌入	184
4.7.1 消息分裂和路由选择	135	6.4.1 创建和使用笛卡儿拓扑结构	184
4.7.2 全端口通信	136	6.4.2 实例: Cannon的矩阵与矩阵相乘	185
		6.5 计算与通信重叠	187

6.6 聚合的通信和计算操作	191	的比较	243
6.6.1 障碍	191	7.11 书目评注	243
6.6.2 广播	191	习题	244
6.6.3 归约	191	第8章 稠密矩阵算法	247
6.6.4 前缀	193	8.1 矩阵向量乘法	247
6.6.5 收集	193	8.1.1 一维行划分	247
6.6.6 散发	194	8.1.2 二维划分	250
6.6.7 多对多	195	8.2 矩阵与矩阵的乘法	253
6.6.8 实例: 一维矩阵与向量相乘	195	8.2.1 简单的并行算法	254
6.6.9 实例: 单源最短路径	197	8.2.2 Cannon算法	254
6.6.10 实例: 样本排序	199	8.2.3 DNS算法	256
6.7 进程组和通信器	200	8.3 线性方程组求解	258
6.8 书目评注	203	8.3.1 简单高斯消元算法	259
习题	204	8.3.2 带部分主元选择的高斯消元算法	269
第7章 共享地址空间平台的编程	205	8.3.3 求解三角系统: 回代法	271
7.1 线程基础	205	8.3.4 求解线性方程组时的数值因素	272
7.2 为什么要用线程	206	8.4 书目评注	272
7.3 POSIX 线程API	207	习题	273
7.4 线程基础: 创建和终止	207	第9章 排序	279
7.5 Pthreads中的同步原语	210	9.1 并行计算机中的排序问题	279
7.5.1 共享变量的互斥	210	9.1.1 输入输出序列的存放位置	279
7.5.2 用于同步的条件变量	216	9.1.2 如何进行比较	280
7.6 控制线程及同步的属性	219	9.2 排序网络	281
7.6.1 线程的属性对象	219	9.2.1 双调排序	282
7.6.2 互斥锁的属性对象	220	9.2.2 将双调排序映射到超立方体 和格网	285
7.7 线程注销	221	9.3 冒泡排序及其变体	290
7.8 复合同步结构	221	9.3.1 奇偶转换	290
7.8.1 读-写锁	222	9.3.2 希尔排序	293
7.8.2 障碍	225	9.4 快速排序	294
7.9 设计异步程序的技巧	228	9.4.1 并行快速排序	295
7.10 OpenMP: 基于命令的并行 编程标准	229	9.4.2 用于CRCW PRAM的并行形式	296
7.10.1 OpenMP编程模型	229	9.4.3 用于实际体系结构的并行形式	298
7.10.2 在OpenMP中指定并发任务	232	9.4.4 主元选择	302
7.10.3 OpenMP中的同步结构	237	9.5 桶和样本排序	303
7.10.4 OpenMP中的数据处理	240	9.6 其他排序算法	305
7.10.5 OpenMP库函数	241	9.6.1 枚举排序	305
7.10.6 OpenMP中的环境变量	242	9.6.2 基数排序	306
7.10.7 显式线程与基于OpenMP编程			

9.7 书目评注	307	11.4.7 IDA*的并行形式	365
习题	308	11.5 并行最佳优先搜索	365
第10章 图算法	315	11.6 并行搜索算法的加速比异常	368
10.1 定义和表示	315	11.7 书目评注	371
10.2 最小生成树: Prim 算法	317	习题	374
10.3 单源最短路径: Dijkstra算法	321	第12章 动态规划	379
10.4 全部顶点对间的最短路径	321	12.1 动态规划概述	379
10.4.1 Dijkstra算法	322	12.2 串行一元DP形式	381
10.4.2 Floyd算法	323	12.2.1 最短路径问题	381
10.4.3 性能比较	327	12.2.2 0/1背包问题	382
10.5 传递闭包	327	12.3 非串行一元DP形式	384
10.6 连通分量	328	12.4 串行多元DP形式	387
10.7 稀疏图算法	331	12.5 非串行多元DP形式	387
10.7.1 查找最大独立集	332	12.6 综述与讨论	390
10.7.2 单源最短路径	334	12.7 书目评注	391
10.8 书目评注	340	习题	391
习题	341	第13章 快速傅里叶变换	395
第11章 离散优化问题的搜索算法	345	13.1 串行算法	395
11.1 定义与实例	345	13.2 二进制交换算法	398
11.2 顺序搜索算法	349	13.2.1 全带宽网络	399
11.2.1 深度优先搜索算法	349	13.2.2 有限带宽网络	404
11.2.2 最佳优先搜索算法	351	13.2.3 并行快速傅里叶变换中的 额外计算	405
11.3 搜索开销因子	353	13.3 转置算法	407
11.4 并行深度优先搜索	353	13.3.1 二维转置算法	407
11.4.1 并行DFS的重要参数	355	13.3.2 转置算法的推广	410
11.4.2 并行DFS分析的一般框架	357	13.4 书目评注	413
11.4.3 负载均衡方案分析	359	习题	414
11.4.4 终止检测	360	附录A 函数的复杂度与阶次分析	417
11.4.5 试验结果	362	索引	419
11.4.6 深度优先分支定界搜索的 并行形式	364		

# 第1章 并行计算介绍

在过去的十年里，微处理器技术有了巨大的发展。处理器的时钟频率从大约40 MHz（例如，1988年前后的MIPS R3000）增加到超过2 GHz（例如，2002年前后的奔腾IV）。同时，处理器现在能够在同一周期里执行多条指令。高端处理器的平均指令周期数（CPI）在过去的十年里提高了大约一个数量级。所有这一切导致峰值浮点运算执行速度（浮点运算次数每秒或FLOPS）增加了几个数量级。除此之外，同一时期计算机在其他方面也取得了重要的进展，其中最重要的进展可能要算内存系统向处理器提供数据速度的提高。计算机硬件和软件方面的重大革新有效地缓解了由数据通道和内存导致的瓶颈。

人们认识并发性对于加速计算单元的作用已经几十年了。然而，它们在提供多样性的数据通道、日益增加的存储单元（包括内存和外存）访问、可扩展性能以及较低成本方面的作用，只反映在并行计算的各种应用中。以2个、4个甚至8个处理器连接在一起的桌面计算机、工程用工作站以及计算服务器正成为通用的设计应用平台。大规模科学与工程计算要依靠更多处理器的并行计算机，其处理器数量通常达到几百台。像数据库或网络服务器这样的数据密集型平台，或者像事务处理及数据挖掘这样的应用，通常使用提供高聚合磁盘带宽的工作站机群。图形图像方面的应用往往用多个渲染管道和多个处理单元来计算及实时地渲染由几百万个多边形构成的现实场景。一些要求高可用性的应用程序要依靠并行或分布式平台提供冗余。因此，无论从性能、价格还是应用程序的需求来看，理解目前可用的各种各样并行计算平台的机理、工具以及编程技术，都是非常重要的。

1

## 1.1 推动并行化

并行软件的设计一直被认为要耗费大量的时间和精力，这可能主要由于说明和协调并行任务固有的复杂性、缺少可移植的算法、标准化的环境和软件开发工具包。当看到微处理器的快速发展时，人们不禁对花大力气研究并行化来加速应用程序的必要性产生怀疑。毕竟，如果花两年时间去开发一个并程序，而在这段时间里硬件或软件平台已变得过时，显然这两年的开发就白费了。然而，硬件设计方面的某些明显趋势表明，单一处理器（或隐式并行）体系结构不能保证未来提升性能的速度，因为缺少隐式并行化以及存在数据通道及内存方面的其他瓶颈。同时，标准化的硬件接口减少了从设计微处理器到设计基于微处理器并行计算机的过渡时间。此外，在编程环境标准化方面取得的可观进展增加了并行应用程序的寿命。所有这一切都为发展并行计算平台提供了有力的支持。

### 1.1.1 计算能力因素——从晶体管到浮点运算速度

1965年，戈登·摩尔提出如下的观察结果：

“半导体上的晶体管数量大约每年增长了一倍。在短时间内这个增长速度可望保持下去。虽然没有理由相信在至少10年的时间内增长速度不能继续稳定，但对较长时间而言，多少难



以确定其增长速度。也就是说，到1975年，每块集成电路上的晶体管数将达到65 000个。”

2 他的推理是根据观察基于元件复杂度和时间之间的对数-线性经验关系中的三个数据点得到的。他由此证明，到1975年，在只有四分之一平方英寸的单一芯片上含有65 000个晶体管是可能的。当1975年拥有65 000个晶体管的16K CCD存储器制造出来后，他的预测被证明是准确的。他在1975年接下来的文章里，把对数-线性关系改成了芯片尺寸、晶体管最小尺寸以及“电路与元件合理排列”之间的指数关系。接下来他写道：

“空间太少挤不进任何东西，除非布局更加合理化，我们只能依靠两个尺寸方面的因素——更大的芯片及更小的晶体管。”

他把元件数量每增长一倍的时间修改为18个月并预测从1975年起按这个降低后的速度增长。这就是著名的“摩尔定律”，也就是芯片的晶体管数量每18个月增长一倍。许多年以来，这条经验定律令人惊奇地在微处理器及动态随机存取存储器上反复地得到验证。通过将元件密度及芯片大小与计算能力相联系，摩尔定律推断出给定价格下计算能力每隔大约18个月翻一番。

在过去几年里，摩尔定律的局限性受到了广泛的争议。避开争议不谈，把晶体管数量转换成每秒运算次数（Operations Per Second, OPS）是至关重要的。虽然制造拥有巨大晶体管数量的元件已经成为可能，但如何利用这些晶体管以提高运算能力是对体系结构的挑战。一个合理的解决方案就是依靠并行化——隐式及显式并行。2.1节将简单地讨论隐式并行，本书的其他部分都将探讨显式并行。

### 1.1.2 内存及磁盘速度的因素

总体计算速度并非仅仅由处理器一个因素所决定，它也取决于内存系统向处理器提供数据的能力。在过去的十年里，当高端处理器的时钟速度大约以每年40%速度增长的同时，此期间动态随机存取存储器（DRAM）访问速度的年增长率仅为10%左右。伴随每个时钟周期处理指令数量的增加，处理器速度与访问内存之间的差异带来了巨大的性能瓶颈。这种处理器速度和动态随机存取存储器延迟之间越来越大的差距通常通过一种称为高速缓存的快速存储器来弥补，它依靠数据访问的局部性来提升内存系统的性能。除了延迟时间外，处理器与动态随机存取存储器间的净有效带宽也对持续计算速度产生影响。

3 内存系统的总体性能是由高速缓存能够满足总内存需求的部分所决定的。在2.2节对内存系统的性能做更详细的介绍。并行平台通常能提供更好的内存系统性能，因为它能提供（1）更大的集合高速缓存，（2）更高的内存系统的集合带宽（两者通常与处理器个数都是线性关系）。此外，并行算法的核心原则，即数据访问的局部性原则，也适用于便利高速缓存实现的串行算法。这个论点可推广到用并行平台能对二级存储达到高集合带宽的磁盘。由此，并行算法产生发展非核心计算的见解。事实上，在一些发展最快的并行计算应用领域，如数据服务器（数据库服务器，网络服务器），已经不再过多地依靠它们的聚合计算能力，而更多的是依靠它们更快取出数据的能力。

### 1.1.3 数据通信因素

随着网络基础设施的不断发展，将因特网作为一个巨大的异构并行/分布式计算环境的构

想开始形成。许多应用很自然地采用了这种计算形式，一些大规模并行计算最令人关注的应用就属于这种广域网分布式平台范畴。外星智能探索（SETI）项目就使用许多家用计算机来分析来自外层空间的电磁信号，人们也尝试用这种方法对极大的整数进行因式分解和求解大规模的离散优化问题。

在许多应用程序中，因特网上的数据或资源的位置是有约束的，对分布在相对低带宽网络上大型商业数据集进行数据挖掘就是这样的例子。在这样的应用中，即使无需重新安排并行计算，计算能力也可用来完成所需的任务，但想要在一个中心位置收集数据也许是不可行的。在这些情况下，并行化的动机不仅来自对计算资源的要求，而且也来自改变（集中化）方法的不可行性或不必要性。

## 1.2 并行计算适用范围

从科学及工程应用的计算模拟，到商业应用的数据挖掘及事务处理等许多领域，并行计算已经产生了巨大的影响。并行化的成本优势与应用对性能上的需求相结合，为促进并行计算提供了令人信服的论据。下面给出并行计算在诸多应用方面的例子。

### 1.2.1 在工程及设计中的应用

并行计算传统上已成功地应用于机翼设计（优化升力、阻力、稳定性），内燃机设计（优化负载分布、燃烧），高速电路设计（设计时延、电容以及电感效应）以及结构设计（优化结构完整性、设计参数、成本等）等方面。最近，微电机系统及毫微电机系统（MEMS及NEMS）的设计引起了人们极大的重视。虽然绝大多数工程和设计中的应用提出了多空间与多时间尺度以及耦合物理现象的问题，但对MEMS/NEMS设计，这些问题更加突出。即使在一个单一的系统里，也经常要处理与量子现象、分子动力学以及像传导、对流、辐射和结构力学等物理过程的随机模型及连续模型有关的混合问题。这对几何建模、数学建模、建立算法以及所有与并行计算机有关的方面都提出了巨大的挑战。

4

在工程与设计方面的其他应用则着重对各种过程进行优化。并行计算机已被用来解决许多离散及连续优化问题。如像线性最优化和分支定界中的单纯型法和内点法以及离散最优化中的遗传规划这样一些算法，都已被有效地并行化和频繁使用。

### 1.2.2 科学计算中的应用

过去几年，在高性能科学计算应用方面取得了革命性的进展。国际人类基因测序联盟和Celera公司对人类基因的测序，在生物信息学方面开辟了令人激动的新领域。理解基因及蛋白质的功能及结构上的特点能使人类理解并根本性地影响生物学过程。以开发新的药品、治疗疾病及改善医疗条件为目的的生物序列分析既需要大规模计算能力，也需要创新的算法。实际上，一些最新的并行计算技术就是专门针对生物信息学方面的应用而发展的。

在计算物理学及计算化学方面的进展强调对尺度从量子现象到大分子结构之间变化过程的理解。这已导致设计新材料、理解化学反应途径和更有效的反应过程。在天体物理学方面的应用包括对银河系的演化过程及热核反应过程的研究，以及对来自天文望远镜的巨大数据集的分析等。并行计算还广泛应用于气象模型、矿物勘探以及洪水预测等领域并对人类日常生活产生极大的影响。