HZ BOOKS
华章教育

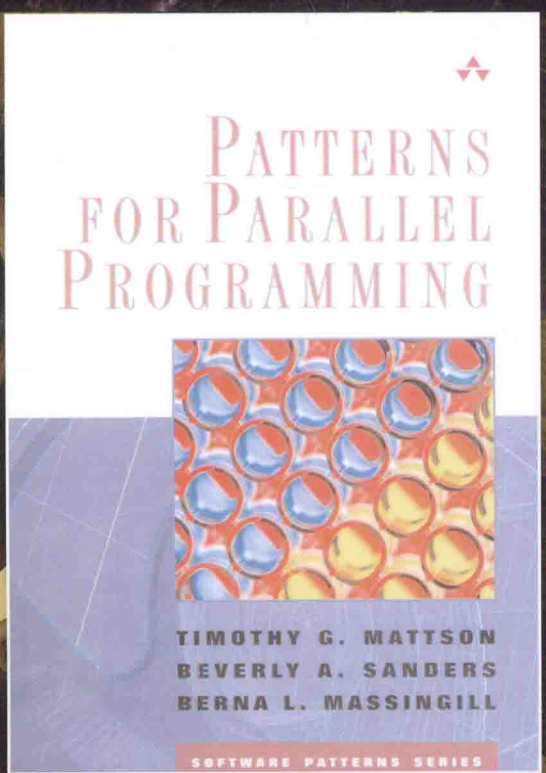
计 算 机 科 学 丛 书

PEARSON

并行编程模式

Timothy G. Mattson
(美) Beverly A. Sanders 著 张云泉 贾海鹏 袁良 译
Berna L. Massingill

Patterns for Parallel Programming



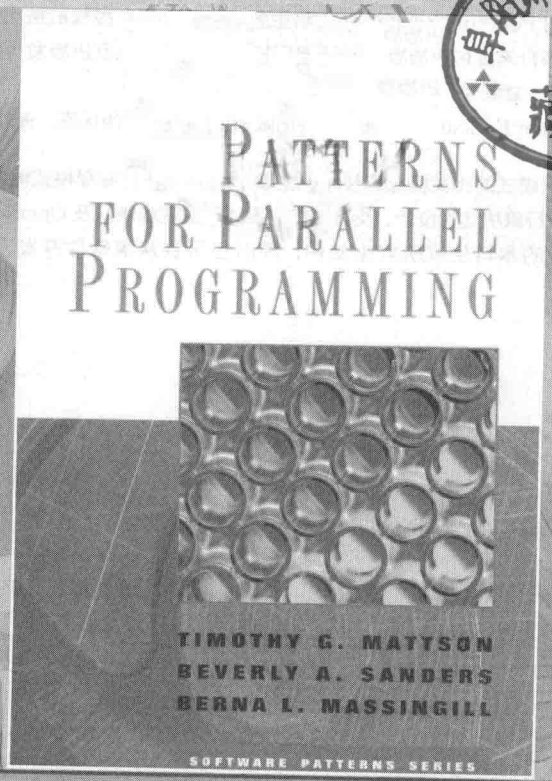
机械工业出版社
China Machine Press

计 算 机 科 学 丛

并行编程模式

(美) Timothy G. Mattson
Beverly A. Sanders
Berna L. Massingill
著
张云泉 贾海鹏 袁良 译

Patterns for Parallel Programming



机械工业出版社

图书在版编目（CIP）数据

并行编程模式 /（美）马特森（Mattson, T. G.），（美）桑德斯（Sanders, B. A.），（美）马森吉尔（Massingill, B. L.）著；张云泉，贾海鹏，袁良译 . —北京：机械工业出版社，2014.11
（计算机科学丛书）
书名原文：Patterns for Parallel Programming

ISBN 978-7-111-49018-0

I. 并… II. ①马… ②桑… ③马… ④张… ⑤贾… ⑥袁… III. 并行程序 – 程序设计 IV. TP311.11

中国版本图书馆 CIP 数据核字（2014）第 304925 号

本书版权登记号：图字：01-2014-6662

本书介绍了并行编程模式的相关概念和技术，主要内容包括并行编程模式语言、并行计算的背景、软件开发中的并发性、并行算法结构设计、支持结构、设计的实现机制以及 OpenMP、MPI 等。

本书可供软件专业的本科生或研究生使用，同时也可供从事软件开发工作的广大技术人员参考。

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：谢晓芳　　责任校对：殷　虹
印　　刷：北京瑞德印刷有限公司　　版　　次：2015 年 2 月第 1 版第 1 次印刷
开　　本：185mm × 260mm　1/16　　印　　张：17.25
书　　号：ISBN 978-7-111-49018-0　　定　　价：75.00 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：hzjsj@hzbook.com

华章科技
HZBOOKS | Science & Technology

文艺复兴以来，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；也正是这样的优势，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，不仅擘划了研究的范畴，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，Brain W.Kernighan，Dennis Ritchie，Jim Gray，Afred 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相助，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，有的还专门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，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@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

华章教育
华章科技图书出版中心

译者序

Patterns for Parallel Programming

随着多核 / 众核处理器的普及以及并行计算集群应用的日益广泛，编写正确、高效的并行程序已经成为软件开发人员面临的一大挑战。并行程序可以归纳为一些具有明确定义的编程模式——用以描述并行编程的形式和方法，即并行编程模式。每一种模式都是具有相同控制结构的一类算法。并行模式的选择将直接影响并行程序的正确性和效率，从而影响整个系统的性能。因此，选择一种有效的并行编程模式对并行程序的编写及性能至关重要。

本书从寻找并发性、算法结构、支持结构和实现机制四个角度深入介绍了并行编程模式的分类、定义、实现、选择及应用。本书提供了从问题描述到最终编码的完整解决方案。首先从高层次的算法问题出发，介绍重组问题以开发算法的潜在并行性的策略及方法；在此基础上，从整体上讨论了如何利用算法的潜在并行性构造并行算法；接着从并行程序构造方法和共享数据结构的角度描述支持并行算法表达的软件构造；最后从进程 / 线程管理和进程 / 线程间交互两个方面，给出将上层空间的模式映射到特定编程环境的方法。本书不仅详细介绍了并行程序设计各个阶段的不同编程模式，讨论相应的关键技术，还详细介绍了 OpenMP、MPI、Java 这三种目前在并行编程社区常用的编程环境。此外，本书配备大量应用和程序实例，方便读者掌握相关技巧。总之，本书作者根据自己多年并行编程的实际经验，从并行算法的本质出发，以一种职业程序员易于掌握的方式对最为关键的基本知识和技术进行了细致讲解。本书可供具有一定并行编程经验的软件开发人员参考，为他们进行并行程序开发提供指导，从而降低并行编程的难度，提高并行程序的性能。

本书作者 Timothy G. Mattson、Beverly A. Sanders 和 Berna L. Massingill 从 1998 年就开始了模式语言和并行计算设计模式的相关工作，都具有丰富的并行编程经验。Timothy G. Mattson 在加州理工学院时，就将自己的分子散射软件移植到 Caltech/JPL 超立方体上。此外，Mattson 还参与了许多重要的并行计算项目，包括 ASCI Red 项目（第一个万亿次浮点运算大规模并行处理计算机）、OpenMP 开发以及 OSCAR（一种流行的集群计算包）。目前，他负责英特尔生命科学市场的战略决策，是英特尔生命科学的首席发言人。

由于时间仓促，加之书中个别术语目前没有统一译法，因此我们对一些术语采取了保留其英文名称的方法。书中翻译的错误和不妥之处，恳请广大读者不吝批评指正。

张云泉

“如果建好了它，他们就会到来。”[⊖]

我们已建成了多处理器工作站、大规模并行超级计算机和集群，但利用这些机器编程的程序员却还没有出现。少数乐于迎接挑战的程序员已经证明，大多数问题可以利用并行计算机快速求解，但普通程序员，特别是那些生活安逸的职业程序员，却忽略了并行计算机。

他们这样做十分不明智，因为并行计算机即将成为主流。多线程微处理器、多核 CPU、多处理器 PC、集群、并行游戏控制台等并行计算机正在逐步占领整个计算市场，在计算机行业中，市场上到处是这样的硬件，这些硬件唯有借助并行程序才能全速运行。但谁将编写并行程序呢？

这是一个老生常谈的问题，甚至在 20 世纪 80 年代早期“killer micros”开始取代传统向量机时，我们就十分担心如何吸引普通程序员编写并行程序。我们尝试了能想到的所有方法，包括高级硬件抽象、隐式并行编程语言、并行语言扩展和可移植消息传递库。但是经过多年的努力之后，“他们”并没有出现，绝大多数程序员并没有致力于编写并行软件。

一个常见的观点是，你不能将新技巧告诉老程序员，因此直到老程序员逐渐退出、新一代程序员逐渐成长后，并行编程问题才能够得到解决。

但我们不认同这种悲观主义态度。多年来，程序员一直在采用新的软件技术方面表现出非凡能力，许多使用 Fortran 的老程序员现在正在编写完美的 Java 面向对象程序。因此问题并非在于老程序员，而在于并行计算专家如何培养并行程序员。

这正是本书的目标，我们希望把握优秀并行程序员思考并行算法本质的过程，并以一种职业程序员易于掌握的方式讲解。为此，我们利用*模式语言*来介绍并行编程。之所以这样选择，不是因为要利用设计模式解决新领域中的问题，而是因为模式已经被证明适用于并行编程。例如，模式在面向对象设计领域非常有效，它们提供了一种用于讨论设计元素的通用语言，并且能够有效地帮助程序员掌握面向对象的设计方法。

本书包含并行编程的模式语言。前两章将介绍并行计算的一些基础知识，包括并行计算的概念和术语，而不是详尽介绍整个领域。

后 4 章介绍了模式语言，对应于创建一个并行程序的 4 个阶段。

- **寻找并发性**。识别可用的并发性，并用于算法设计中。
- **算法结构**。用一种高级结构组织一个并行算法。
- **支持结构**。将算法转化为源代码，考虑如何组织并行程序以及如何管理共享数据。
- **实现机制**。寻找特定的软件构造，实现并行程序。

这些模式紧密相关，构成了 4 个设计空间。从顶部（寻找并发性）开始，依次经历每一种模式，到达底部（实现机制）时，就可以得到并行程序的一个详细设计。

如果目标是一个并行程序，那么所需要的除了一个并行算法之外，还包括编程环境和用

⊖ 这是电影《梦幻之地》中的对话。影片中无法完成梦想的农场主人公，有一天听到神秘声音说：“如果建好了它，他们就会到来。”于是他铲平了自己的玉米田建造了一座棒球场，最终他的棒球偶像真的来到这里打球。——译者注

于表示程序源代码中并发性的方法。程序员过去需要面对大量不同的并行编程环境。幸运的是，随着时间的推移，并行编程社区目前主要使用三种编程环境。

- OpenMP：扩展了 C、C++ 和 Fortran，主要用来在共享内存计算机上编写并行程序。
- MPI：用于集群和其他分布式存储计算机的一种消息传递库。
- Java：一种面向对象的编程语言，支持共享内存计算机上的并行编程，并支持分布式计算的标准类库。

很多读者可能熟悉其中的一种或几种编程语言，但为了方便那些并行计算的初学者，附录简要介绍了这几种编程环境。

我们研究模式语言多年，现在将其总结为一本书以便使用。但是这并非此项工作的终点。我们期望读者有自己的新想法，并设计更好的并行编程新模式。我们可能遗漏了模式语言的某些重要特征，希望并行计算社区能推广这种模式语言。我们将继续更新并改进这种模式语言，直到它成为并行计算社区的统一观点。我们将开展一些实际的工作，例如，使用模式语言来创建更好的并行编程环境，帮助人们使用这些模式来编写并行软件。我们将一直努力，直到并行软件全面代替串行软件的那一天。

致谢

我们从 1998 年开始研究模式语言，从如何设计并行算法的一个模糊概念开始到完成本书，已经走过了一段漫长而崎岖的道路。如果没有以下人员的帮助，我们不可能完成这项任务。

Mani Chandy 将 Tim 介绍给 Berverly 和 Berna，并坚信我们能成为一个优秀团队。美国国家科学基金、英特尔公司、三一大学多年来一直支持本项目的研究。每年夏天在伊利诺伊举办的 PLoP 会议参与者为本书的具体并行编程模式提供了大量帮助。该会议的组织和审稿过程十分具有挑战性，但没有这些经验我们无法完成模式语言的研究。同时感谢仔细阅读本书并指出大量错误的审稿专家。

最后，还要感谢我们的家人。感谢 Beverly 的家人（Daniel 和 Steve），Tim 的家人（Noah、August 和 Martha），以及 Berna 的家人（Billie），感谢他们的支持和付出。

Tim Mattson
Beverly Sanders
Berna Massingill

Timothy G. Mattson

Timothy G. Mattson 拥有加州大学圣克鲁兹分校的化学博士学位，研究方向为量子分子散射理论；之后在加州理工学院进行博士后研究，致力于将自己的分子散射软件移植到 Caltech/JPL 超立方体上。他曾担任多项与计算科学相关的商业和学术职务，参与了许多重要的并行计算项目，包括 ASCI Red 项目（第一个万亿次浮点运算大规模并行处理计算机）、OpenMP 开发以及 OSCAR（一种流行的集群计算包）。目前，他负责英特尔生命科学市场的战略决策，是英特尔生命科学社区的首席发言人。

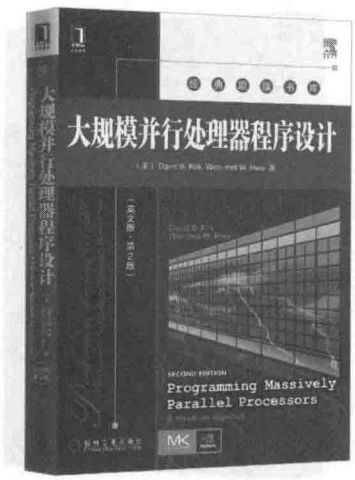
Beverly A. Sanders

Beverly A. Sanders 拥有哈佛大学的应用数学博士学位。她曾任教于马里兰大学、瑞士联邦理工学院（ETH Zürich）以及加州理工学院，目前任教于佛罗里达大学的计算机信息科学与工程学系。她的教学和科研一直围绕着设计模式、形式方法和编程语言思想等技术的开发与应用，以帮助程序员构建高质量、正确的程序，尤其是并发程序。

Berna L. Massingill

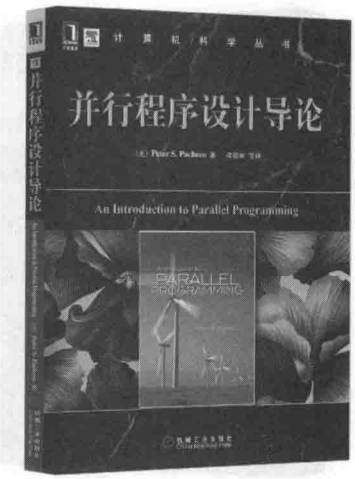
Berna L. Massingill 拥有加州理工学院的计算机科学博士学位，之后在佛罗里达大学进行博士后研究，和其他两位作者开始了关于并行计算设计模式的工作。她目前任教于三一大学（位于得克萨斯州圣安东尼奥市）计算机科学系。她拥有十余年的编程工作经验，最初从事主机系统编程工作，后来在一个软件公司担任开发人员。她的研究兴趣包括并行和分布式计算、设计模式以及形式方法。她的教学和研究目标之一是帮助程序员构建高质量、正确的程序。

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

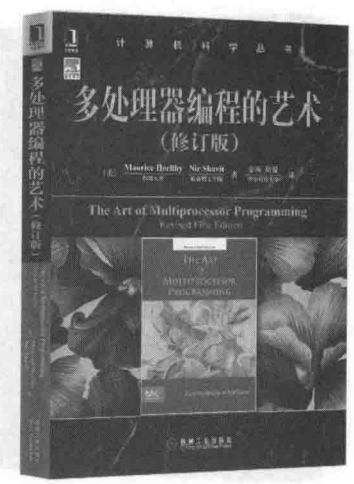


并行程序设计导论

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

高性能科学与工程计算

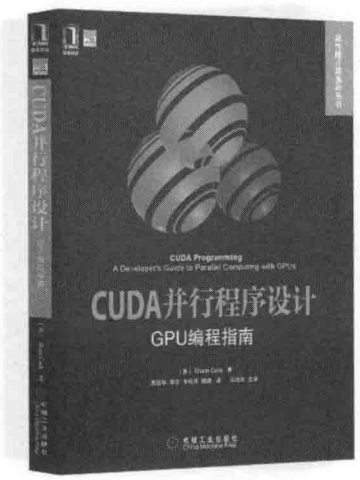
作者：Georg Hager 等 ISBN：978-7-111-46652-9 定价：69.00元



多处理器编程的艺术（修订版）

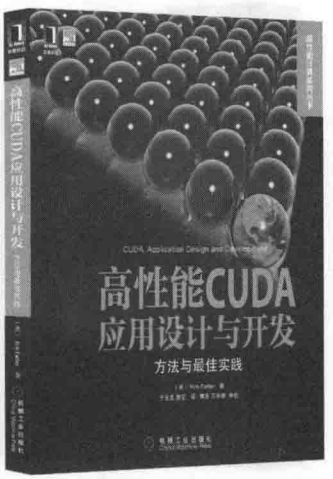
作者：Maurice Herlihy 等 ISBN：978-7-111-41858-0 定价：69.00元

推荐阅读



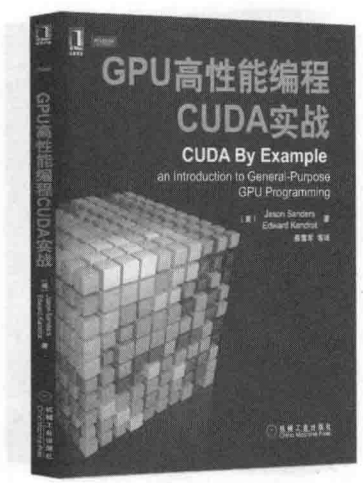
CUDA并行程序设计：GPU编程指南

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



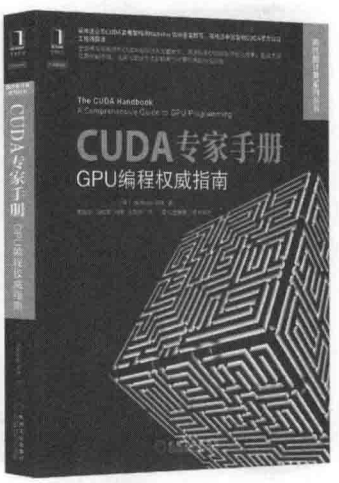
高性能CUDA应用设计与开发：方法与最佳实践

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



CUDA专家手册：GPU编程权威指南

作者：Nicholas Wilt ISBN：978-7-111-47265-0 定价：85.00元

目 录

Patterns for Parallel Programming

第 1 章

Patterns for Parallel Programming

并行编程的模式语言

1.1 引言

计算机被广泛用于模拟自然科学、医学和工程学等领域中的真实物理系统，包括天气预报模拟系统、震撼电影的场景模拟系统，并且可以使用任意的计算能力来完成更加精细的模拟。无论是顾客购物模式，还是来自于宇宙的遥测数据或者 DNA 序列，它们需要分析的数据量都十分巨大。为了提供这种计算能力，设计者将多个处理单元融入一个较大的系统中。这就是所谓的*并行计算机*，它能够同时运行多个任务，在更短的时间内解决规模更大的问题。

过去，并行计算机仅用于解决一些最重要的问题，但是从 20 世纪 90 年代中期开始，随着微处理器内部开始支持多线程技术，并且在单个硅片上可容纳多个处理器内核，这种情况已经发生了根本性变化。现在，并行计算机随处可见，几乎每所大学计算机系都至少有一台并行计算机。几乎所有的石油公司、汽车制造商、药品开发公司和特效工作室都已经使用了并行计算。

例如，计算机动画的生成过程是将动画文件的信息（如光、纹理和阴影）应用于 3D 模型以生成 2D 图像，再用这些 2D 图像组成电影的帧。为较长的电影生成所需的帧时（每秒 24 帧），并行计算是非常必要的。1995 年，由 Pixar 公司发行的《玩具总动员》是第一个完全通过计算机软件制作的电影，该电影由一台包括 100 个双处理器机器的名为“renderfarm”[PS00] 的机器处理。Pixar 公司在 1999 年制作《玩具总动员 2》时利用了一台具有 1400 个处理器的系统。由于提高了处理能力，影片的效果（包括纹理、服饰和艺术效果）有大幅度提升。Monsters 公司 2001 年使用了一个拥有 250 个企业级服务器。（每个服务器包含 14 个处理器，共 3500 个处理器）的系统。然而利用更高的计算性能，包括处理器数目以及单个处理器计算性能，来提升动画效果，因此生成一幅帧需要的时间仍保持不变。

生物科学在能够从多种生物体（包括人）中获得 DNA 序列信息后实现了跨越式的发展。由 Celera 公司提出并成功使用的一种称为基因组鸟枪的排序算法，可将基因组划分为多个子段，首先通过实验确定每个子段的 DNA 序列，然后利用计算机通过重构子段间的重叠区域来重组整个序列。Celera 公司排序人类基因组时使用的计算机包括 150 台四路服务器以及一台具有 16 个处理器和 64GB 内存的服务器，实际计算包括 5 兆亿次基对基比较 [Ein00]。

SETI@home 项目 [SET，ACK+02] 的目的是寻找地外智能存在的证据，它是另一个展示并行计算能力的实例。该项目利用位于波多黎各的世界上最大的阿雷卡纳特无线望远镜搜索外太空，分析收集到的数据，搜索智能信号源。该项目的计算需求超过了世界上最大超级计算机的性能，因此只能利用*公共计算资源*满足性能需求，即将通过 Internet 互联的世界范围内的 PC 整合为一台并行计算机。项目将收集的数据划分为一些子任务，并将子任务通过 Internet 发送到每个客户端计算机上，利用这些分布的个人计算机的空闲时间进行计算。每个

客户端定期连接到 SETI@home 服务器，下载数据并执行分析计算，最后将结果发送回服务器。客户端程序被设计为一个屏幕保护程序，仅当计算机处于空闲状态时才将 CPU 贡献给 SETI 问题执行计算任务。目前，一个工作子任务平均需要一台计算机 7 ~ 8 个小时的 CPU 时间，从项目启动到现在已有 2.05 亿个工作单元得以处理。最近，也出现了一批利用公共计算资源的新项目，并且一些大公司也利用其内部个人计算机的空闲计时解决从新药筛选到芯片设计验证等问题。

尽管通过并行计算可以以更短的时间解决一些串行无法完成的问题，但实现并行也需要付出一些代价。编写并行计算软件是十分困难的，因此只有少量程序员具备丰富的并行编程经验。如果要利用并行计算机带来的并行性潜能，大部分程序员都需要学习如何编写并行程序。

本书将为有串行程序设计经验的程序员介绍编写并行程序的设计方法。虽然已经有一些优秀的书籍介绍过特殊的并行编程环境，但本书不同之处在于，我们主要介绍并行算法的构造和设计思路。因此，我们将使用*模式语言*的概念，在面向对象社区中已经大量使用了这种包含专家设计经验的高度结构化表示。

本书前两章将介绍基础知识，其中第 1 章概述理解和使用模式语言所必需的并行计算相关概念及其背景，第 2 章将更深入地讨论并行程序员所使用的基本概念和术语，其余章节介绍具体的模式语言。

1.2 并行编程

并行计算的关键是*可挖掘的并发性*。如果一个计算问题能够被分解成多个子问题，并且所有子问题可在相同时间内同时、安全地解决，则该问题就存在并发性。必须分析问题并发性并在代码中显示开发性，使得子问题能够并行执行，也就是说，问题解决方案必须具备并发性。

大部分大规模计算问题具备一定的并发性。程序员通过建立并行算法并在并行编程环境中实现来挖掘问题的并发性。这样，当并行程序在多处理器系统上运行时，才能在更短的时间内完成计算。此外，相对于单处理器系统，多处理器系统能处理规模更大的问题。

例如，假设计算包括求一个大型数据集的和时，串行计算将所有的值按顺序相加；如果使用多处理器，则需划分数据集，并在不同处理器上计算每个子集的和，同时完成计算，最后求所有子集之和。这样利用多处理器并行计算能更快地完成任务。若每个处理器都有私有内存，将数据分布到多个处理器上即可处理更大规模的问题。

上述简单示例展示了并行计算的本质。并行计算的目标是利用多处理器在更短时间内解决问题，以及处理规模更大的问题（与单处理所能处理的问题规模相比）。程序员需要鉴别出问题的并发性，并设计并行算法来挖掘问题中的并发性，然后在合适的并行编程环境中编写并行代码，最后在并行系统上运行。

并行编程也面临一些挑战。通常，问题所包含的并发任务间具有一定依赖性，这些子计算以不同顺序完成可能会影响程序的运行结果。例如，在上述并行求和示例中，某一子集求和计算完成后，才能与其他子集的和相加。该算法对所有任务强加了一种偏序顺序（即所有子任务完成后才能够被组合在一起计算最终结果）。此外，由于浮点运算具有非结合性和非

交换性，因此当求和运算顺序不同时，计算结果可能会有微小的数值差。设计安全的并行程序需要付出大量努力，优秀并行程序员必须非常谨慎以保证这些不确定性不会影响最终计算结果。

即使一个并行程序是“正确”的，它也可能无法通过挖掘并发性来实现性能提高。必须确保挖掘并发性而导致的额外开销不会严重影响程序运行时间，并且在其他问题中实现负载平衡通常不像求和问题那样简单。并行算法效率依赖于它与底层硬件体系结构间的映射关系，一个并行体系结构上执行效率非常高的并行算法在另一个体系结构上执行的性能可能很差。

第 2 章将更为量化地介绍并行计算并重新讨论这个问题。

1.3 设计模式和模式语言

设计模式描述在特定上下文中类似问题的有效解决方法。模式具有预定的格式，包括模式名称、上下文的描述、目标和限制以及相应的解决方案。其理念是记录专家经验，供他人遇到类似问题时参考借鉴。除了解决方法本身外，模式名称也是非常重要的，它构成了领域专用词汇的基础，有效加强同一领域设计者之间的交流。

设计模式最早由 Christopher Alexander 提出，其应用的领域是城市规划和建筑学[AIS77]。Beck 和 Cunningham[BC87] 最早将设计模式概念引入软件工程社区，随着 Gamma、Helm、Johnson 和 Vlissides[GHJV95] 的名为 GoF（Gang of Four 的缩写）的书籍的出版，这一概念在面向对象编程中占据了重要地位。该书收集了大量面向对象编程的设计模式。例如，Visitor 模式描述了一种组织类的方式，能独立实现不同数据结构的代码与遍历它的代码，因此遍历仅仅依赖于每个节点的类型和实现该遍历的类。这能够在不改变数据结构类的情况下灵活地添加新功能，为数据结构的不同遍历方法实现提供了便利。GoF 书中介绍的设计模式已经进入了面向对象编程词典，并已在学术文章、商业出版物和系统文档中广泛应用，这些设计模式已成为软件工程师所必需的知识。

一个组建于 1993 年的名为 Hillside Group[Hil] 的非盈利性教育组织促进了模式和模式语言的使用，更进一步地说，它鼓励人们编写通用的编程和设计实践方面的规范，因此促进了人们在计算机领域的交流。为了设计新的模式并帮助模式编写者提高技能，Hillside Group 每年举办一次编程模式语言研讨会（Pattern Languages of Programs，PLoP），并在其他地方举办了分会，例如 ChiliPLoP（在美国西部）、KoalaPLoP（在澳大利亚）、EuroPLoP（在欧洲）和 Mensore PLOP（在日本）。这些研讨会的论文集 [Pat] 包含大量模式资源，覆盖了大多数软件应用领域，并成为几本书的 [CS95、VCK96、MRB97、HFR99] 主要素材。

Alexander 最初研究模式时，不仅提出了一个模式分类方法，还提出了一种模式语言，从而引入了一种新的设计方法。在模式语言中，所有模式组织为一个特殊结构，用户遍历模式集，并选择具体模式来设计复杂系统。设计者在每个决策点上选择一个适合的模式，该模式可以导出其他多个模式，最终通过一个模式网络完成设计。因此，模式语言包括了一种设计方法学，并向应用程序开发人员提供了特定领域建议（尽管都称为语言，但模式语言并不是一种编程语言）。

1.4 关于并行编程的模式语言

本书给出了用于并行编程的模式语言，它具有许多优势。最直接的好处是它可以通过提

供重要问题解决方法目录、扩展的词汇表和方法学来传播专家经验。我们希望在并行程序开发的全过程中提供指导来降低并行编程的难度。程序员首先应深入理解要解决的实际问题，然后利用模式语言，最终得到详细的并行设计或代码。我们的长期目标是希望模式语言成为定性评估不同编程模型、促进并行编程工具开发的基础。

模式语言由4个设计空间组成，包括*寻找并发性*、*算法结构*、*支持结构*和*实现机制*，如图1-1所示，4个空间构成一个线性层次，其中寻找并发性位于顶部，而实现机制位于底部。

图1-1 模式语言概况

*寻找并发性*设计空间的目标是重组问题以揭示其可开发的并
5 发性。设计者在这个层次中主要面对高层次算法问题，并揭示问题的潜在并发性。*算法结构*设计空间利用潜在并发性构造算法，设计者在这个层次上考虑如何利用*寻找并发性*模式中的并发性。*算法结构*模式描述开发并发性的整体策略。*支持结构*设计空间为*算法结构*设计空间和*实现机制*设计空间提供了一个中间层。支持结构设计空间有两组重要的模式，一组程序构造方法的模式，另一组是通用共享数据结构模式。*实现机制*设计空间将上层空间的模式映射到特定的编程环境中。它描述进程/线程管理（例如，创建或销毁进程/线程）和进程/线程间交互（例如信号量、栅栏或消息传递）的通用机制。*实现机制*设计空间中的条目直接被映射到特定并行编程环境中的元素，因此不表示为模式。但它们都包含在模式语言中，
6 以便提供从问题描述到最终编码的完整解决方案。