

并行程序设计

Parallel Programming

Techniques and Applications
Using Networked Workstations
and Parallel Computers

Barry Wilkinson

(美) Michael Allen 著

北卡罗来纳大学

陆鑫达 等译

140

计算机科学丛书

并行程序设计

(美) Barry Wilkinson 著
Michael Allen

(北卡罗来纳大学)

陆鑫达 等译



机械工业出版社
China Machine Press

本书的主要内容是使用连网工作站和并行计算机并行编程的技术及应用。书中介绍了流水线、分治、同步、工作池等并行技术以及经典的排序、矩阵相乘、线性方程组求解、图像处理、搜索和优化算法的并行实现，并提供了大量的PVM和MPI伪代码及例程。

本书是计算机专业本科生、研究生并行程序设计课程的较好教材。

Barry Wilkinson & Michael Allen:Parallel Programming:Techniques and Applications Using Networked Workstation and Parallel Computers.

Authorized translation from the English language edition published by Prentice Hall.

Copyright © 1999 by Prentice Hall, Inc. All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由美国Prentice Hall公司授权机械工业出版社独家出版。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。

版权所有，侵权必究。

本书版权登记号：图字：01-2001-2203

图书在版编目（CIP）数据

并行程序设计/（美）威尔金森（Wilkinson, B.），（美）艾伦（Allen, M.）著；陆鑫达等译. -北京：机械工业出版社，2002.1

（计算机科学丛书）

书名原文：Parallel Programming:Techniques and Applications Using Networked Workstation and Parallel Computers

ISBN 7-111-09437-9

I.并… II.①威… ②艾… ③陆… III.并行程序-程序设计… IV.TP311.11

中国版本图书馆CIP数据核字（2001）第084566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：杨海玲

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002年1月第1版第1次印刷

787mm × 1092mm 1/16 · 23.5印张

印数：0 001-5 000册

定价：43.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

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，不仅擘划了研究的范畴，还揭橥了学术的源变，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、研究及收藏。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：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。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。

“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。

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，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

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
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
袁宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

译者序

本书是有关使用连网计算机来进行并行计算的一本很实用的教科书。本书是Barry Wilkinson和Michael Allen两位教授多年教学和科研的结晶。作者用此教材指导美国大学一年级学生进行并行编程实践，具有超前意识，为此获得美国国家科学基金会的资助，使此书得以出版。我们翻译此书的宗旨是想促使并行/分布计算，特别是使用连网计算机的并行/分布计算，在中国的普及和发展，尤其是在高等院校中的普及和发展。

本书介绍了常用的算法范例，包括分治、流水线、同步计算、主从方式及工作池等内容，并介绍了常用的经典算法，如排序、矩阵相乘、线性方程组求解、图像处理中的预处理和相应的变换、搜索和优化(包括遗传算法)等，这是本书的一大特色。读者掌握了这些算法范例和经典算法后就可为并行编程打下良好基础。书中每一章后面均附有习题，除了与每章内容有关的习题之外，还有不少习题源自现实生活，既富有启发性，又具有趣味性，应该说这是本书的另一特色。

本书的翻译工作由陆鑫达负责和组织，并翻译目录、前言、第1~3章，以及第11、12章，汤勇平翻译第4章，曾志勇翻译第5章，支小莉翻译第6章，张建翻译第7章，钟嵘翻译第8章，吴欣翻译第9章，徐蔚文翻译第10章。译稿全文由陆鑫达做了校对。

由于翻译时间仓促，再加上有的术语国内没有统一的译法，故翻译中的错误或不妥之处在所难免，恳请广大读者不吝指正。

陆鑫达

上海交通大学计算机科学与工程系

2001年6月8日

译者简介

陆鑫达 现为上海交通大学计算机科学与工程系教授、博士生导师、高性能计算研究室主任，中国计算机学会体系结构专委会副主任，中国计算机学会开放系统专委会高级委员，贵州大学兼职教授。1961年和1964年分别获哈尔滨大学计算机专业学士和硕士学位。1979—1981年为英国纽卡舍尔大学计算机系统访问学者，主要从事高度并行计算技术、VLSI芯片设计技术和处理机互连等技术研究。1987—1990年为德国GMD-FIRST柏林计算机研究所和柏林工业大学（TUB）的客座首席科学家，主要从事新型数据结构高性能计算系统研究，并负责国家自然科学基金会重大项目中的中德国际合作项目。曾任《中国大百科全书》“自动控制和系统工程”卷“信息处理”分卷副主编。主编教材《计算机系统结构》（高等教育出版社，1996年3月）。教材译著：《可扩展并行计算：技术、结构与编程》（机械工业出版社，2000年5月）。

目前的主要研究方向为高性能计算及应用、异构计算及元计算（包括Grid计算）、网络计算及其编程环境、机群计算（包括体系结构及中间件）、遗传和进化算法在映射和调度中的应用、分布计算及移动计算、智能代理等。

前 言

本书的目的是介绍并行编程技术。并行编程使用多计算机或多处理机的计算机来求解问题，它比使用单台计算机的计算速度要快得多。它也为求解更大规模的问题提供了机会，这些问题需要更多的计算步骤或更大存储容量需求，之所以能满足后一要求是因为多计算机和多处理机系统通常比单计算机有更大的总存储容量。在本书中，我们讨论的重点是使用多计算机进行并行编程，它们之间的通信是通过发送消息来完成的，从而出现了消息传递并行编程的术语。我们所使用的计算机可以是不同的类型(PC, SUN, SGI等)，但它们必须由网络进行互连，此外还必须有一个软件环境以在计算机间进行消息传递。适当连网的计算机可广泛地作为学生的基本计算平台，以便可以避免使用特殊设计的多处理机系统。为实现消息传递并行编程，可使用几种软件工具，其中包括PVM和几个MPI的实现方案，它们均可免费得到。这些软件也可在特殊设计的多处理机系统上使用，如果这些系统可以使用的话。就实用这一点而言，我们所讨论的技术和应用是独立于系统的。

本书分为两个部分。在第一部分中，将讨论并行编程的基本技术。第一部分中的各章涉及了所有的基本方面，通过简单问题的求解来说明技术，但是这些技术本身可在许多场合用来求解问题。通常我们先给出顺序的示例代码，然后再给出实际的并行伪代码。一般来讲，基本算法在本质上已经是并行的，而顺序版本不自然地使用循环将其串行化。当然某些算法为了高效地进行并行求解必须进行重构，而这种重构可能不是立即出现的。第一部分中第8章所介绍的并行编程类型并不是围绕消息传递多计算机的，而是围绕专门设计的共享存储多处理机系统的。在这一章内，叙述如何使用Pthreads，这是一个IEEE的多处理机标准系统，已广泛流传并可在单计算机上使用。

学习第一部分所需的准备是有关顺序编程的知识，如C语言的使用和相关数据结构的知识，在掌握了基本的顺序编程后可立即进行第一部分的学习。第一部分中许多指定的作业甚至无需专门的数学知识就可尝试求解。如果需要使用MPI或PVM，则可用具有消息传递库调用的C语言编写程序。在本书的附录中对如何进行具体的库调用做了说明。

许多并行计算问题具有一些专门开发的算法，第二部分中所研究的专用算法涉及非数值和数值范畴。在学习第二部分时，将需要一些数学概念，如矩阵。在第二部分中涉及的主题包括排序、矩阵乘法、线性方程组、偏微分方程、图像处理以及搜索和优化。图像处理特别适合于并行化，因此第二部分中将其作为饶有兴趣的一个应用，专门用一章加以介绍，这种应用对于多种项目具有重要的潜力。在图像处理这一章中，还讨论了所涉及的快速傅里叶变换，这一重要变换还应用在许多其他领域中，包括信号处理和语音识别。

在每章的末尾列出了许多“现实生活”习题，其中绝大部分源自实际生活。这些习题的求解无需专门的数学知识，因而是本书的特色之一；这些习题有助于开发使用并行编程技术的技巧，而不是简单地学习如何去求解像数的排序或矩阵相乘那样的专门问题。

第一部分中的主题宜作为一般顺序编程课程的补充。在北卡罗来纳夏洛特分校，我们以这种方式指导一年级学生进行并行编程。在这种环境下，本书便是对顺序编程教科书的补充。我们设想所使用的顺序编程语言为C或C++。第一部分和第二部分合在一起适合作为高年级并

行编程/并行计算课程，在UNCC我们以这种方式使用本书。

有关UNCC环境的全部细节以及具体网站的细节可以从http://www.cs.uncc.edu/par_prog中找到。在该网站中还有许多Web页面帮助学生如何对并行程序进行编译和运行，网站上还提供了一些示例程序，指导教师还可得到有关的教师指导手册。我们教授并行编程的工作是与北卡罗来纳州立大学的并行处理地区培训中心(Regional Training Center for Parallel Processing)所进行的培训工作相关联的，有关这个培训中心的更多情况可以从网站<http://renoir.csc.ncsu.edu/RTCPP>上找到。

本书是美国国家科学基金会对我们作者在北卡罗来纳大学夏洛特分校为一年级学生介绍并行编程技术项目(美国国家科学基金项目“将并行编程技术引入一年级课程”，项目编号DUE 9554975)进行资助的直接结果。我们极其感谢国家科学基金会的计划主任M. Mulder博士对我们项目的支持，没有他的帮助，我们不可能去追求在教科书中提出的那些想法。我们也要感谢参与本项目的研究生和本科生。这个学生小组帮助开发了有关素材和作业习题。还要感谢计算机科学系的系统管理员James Robinson，他建成了我们的本地工作站集群，没有该集群我们不可能完成这一著作。

我们还要感谢UNCC的许多学生，他们在过去的几年中帮助我们改进了教材素材，特别是那些“远程课程”，在这些课程中，本书的材料成为以独特方式实验的课堂。这些远程课程除了UNCC外还向几所北卡罗来纳州立大学做了传播，其中包括北卡罗来纳大学阿什维尔分校，北卡罗来纳大学格林斯伯勒分校、北卡罗来纳大学威尔明顿分校和北卡罗来纳州立大学。我们还要向许多人致谢，其中特别要提及的是北卡罗来纳大学阿什维尔分校的Wayne Lang教授和北卡罗来纳州立大学的Mladen Vouk教授。Lang教授投身于课堂的教案开发，而Vouk教授除了以客座专家身份授课外，还设计了一个给人深刻印象的Web页面，其中包括了我们课程的“实音”和“自动翻转”的幻灯片。(这些授课情况可在<http://renoir.csc.ncsu.edu/CSC495A>上看到。)杜克大学的John Board教授以及北卡罗来纳大学查珀尔希尔分校的Jan Prins教授也欣然做了客座专家讲演。承蒙Raul Gallard教授的邀请，我们还在阿根廷的圣路易斯国立大学讲授了基于本书素材的并行编程课程——所有这一切活动均有益于我们编写本书。我们要向Prentice Hall出版社的Alan Apt和Laura Steele致谢，他们接受了我们编写书的提议并自始至终支持我们。我们还要向所有评阅人致谢，感谢他们非常有益的指点。

最后我们恳请读者对本书提出宝贵意见并不吝指正，并通过下列电子邮件地址传给我们：abw@uncc.edu(Barry Wilkinson)或cma@uncc.edu(Michael Allen)。

Barry Wilkinson

Michael Allen

于北卡罗来纳大学夏洛特分校

目 录

出版者的话
专家指导委员会
译者序
译者简介
前言
作者简介

第一部分 基本技术

第1章 并行计算机	2
1.1 对计算速度的需求	2
1.2 并行计算机的类型	4
1.2.1 共享存储器多处理机系统	4
1.2.2 消息传递多计算机系统	5
1.2.3 分布式共享存储器系统	6
1.2.4 MIMD和SIMD分类法	7
1.3 消息传递多计算机的体系结构特征	8
1.3.1 静态网络消息传递多计算机	8
1.3.2 嵌入	12
1.3.3 通信方法	15
1.3.4 输入/输出	17
1.4 用连网计算机作为多计算机平台	18
1.5 提高计算速度的潜力	21
1.6 小结	26
推荐读物	26
参考文献	27
习题	29
第2章 消息传递计算	31
2.1 消息传递编程基础	31
2.1.1 编程的选择	31
2.1.2 进程的创建	31
2.1.3 消息传递例程	33
2.2 使用工作站集群	38
2.2.1 软件工具	38
2.2.2 PVM	38
2.2.3 MPI	43
2.2.4 伪代码构造	49

2.3 并行程序的评估	50
2.3.1 并行执行时间	50
2.3.2 时间复杂性	52
2.3.3 对渐近分析的评注	55
2.3.4 广播/汇集的时间复杂性	55
2.4 并行程序的调试和评估	58
2.4.1 低层次调试	58
2.4.2 可视化工具	59
2.4.3 调试策略	60
2.4.4 用经验方法评估程序	60
2.4.5 对优化并行代码的评注	62
2.5 小结	63
推荐读物	63
参考文献	63
习题	65
第3章 易并行计算	67
3.1 理想的并行计算	67
3.2 易并行计算举例	68
3.2.1 图像的几何变换	68
3.2.2 曼德勃罗特集	72
3.2.3 蒙特卡罗法	78
3.3 小结	82
推荐读物	82
参考文献	82
习题	83
第4章 划分和分治策略	88
4.1 划分	88
4.1.1 划分策略	88
4.1.2 分治	91
4.1.3 M路分治	95
4.2 分治技术举例	97
4.2.1 使用桶排序法排序	97
4.2.2 数值积分	100
4.2.3 N体问题	103
4.3 小结	107

推荐读物	107	7.2.2 分散式动态负载平衡	163
参考文献	108	7.2.3 使用线形结构的负载平衡	165
习题	109	7.3 分布式终止检测算法	167
第5章 流水线计算	114	7.3.1 终止条件	167
5.1 流水线技术	114	7.3.2 使用应答消息实现终止	167
5.2 流水线应用的计算平台	117	7.3.3 环形终止算法	168
5.3 流水线程序举例	118	7.3.4 固定能量分布式终止算法	170
5.3.1 数字相加	118	7.4 程序举例	170
5.3.2 数的排序	120	7.4.1 最短路径问题	170
5.3.3 生成质数	123	7.4.2 图表示	171
5.3.4 线性方程组求解——特殊案例	125	7.4.3 图的搜索	172
5.4 小结	127	7.5 小结	177
推荐读物	128	推荐读物	177
参考文献	128	参考文献	178
习题	128	习题	179
第6章 同步计算	132	第8章 共享存储器编程	184
6.1 同步	132	8.1 共享存储器多处理机	184
6.1.1 路障	132	8.2 说明并行性的结构	186
6.1.2 计数器实现	133	8.2.1 创建并发进程	186
6.1.3 树实现	135	8.2.2 线程	187
6.1.4 蝶形路障	136	8.3 共享数据	191
6.1.5 局部同步	136	8.3.1 创建共享数据	191
6.1.6 死锁	137	8.3.2 访问共享数据	192
6.2 同步计算	137	8.3.3 并行性的语言结构	198
6.2.1 数据并行计算	137	8.3.4 相关性分析	199
6.2.2 同步迭代	140	8.3.5 具有高速缓存的系统中 的共享数据	201
6.3 同步迭代程序举例	140	8.4 程序举例	203
6.3.1 用迭代法解线性方程组	140	8.4.1 UNIX进程	204
6.3.2 热分布问题	145	8.4.2 Pthreads的例子	206
6.3.3 细胞自动机	152	8.4.3 Java的例子	208
6.4 小结	153	8.5 小结	209
推荐读物	153	推荐读物	210
参考文献	154	参考文献	210
习题	154	习题	211
第7章 负载平衡与终止检测	160		
7.1 负载平衡	160		
7.2 动态负载平衡	161		
7.2.1 集中式动态负载平衡	162		
		第二部分 算法和应用	
		第9章 排序算法	216

9.1 概述	216	参考文献	264
9.1.1 排序	216	习题	265
9.1.2 可能的加速	216	第11章 图像处理	268
9.1.3 秩排序	217	11.1 低层图像处理	268
9.2 比较和交换排序算法	219	11.2 点处理	269
9.2.1 比较和交换	219	11.3 直方图	270
9.2.2 冒泡排序与奇偶互换排序	221	11.4 平滑、锐化和噪声消减	270
9.2.3 二维排序	224	11.4.1 平均值	271
9.2.4 归并排序	226	11.4.2 中值	272
9.2.5 快速排序	228	11.4.3 加权掩码	274
9.2.6 超立方体上的快速排序	230	11.5 边缘检测	275
9.2.7 奇偶归并排序	234	11.5.1 梯度和幅度	275
9.2.8 双调谐归并排序	235	11.5.2 边缘检测掩码	276
9.3 小结	238	11.6 霍夫变换	279
推荐读物	239	11.7 向频域的变换	282
参考文献	239	11.7.1 傅里叶级数	282
习题	240	11.7.2 傅里叶变换	282
第10章 数值算法	243	11.7.3 图像处理中的傅里叶变换	283
10.1 矩阵——回顾	243	11.7.4 离散傅里叶变换算法的并行化	285
10.1.1 矩阵相加	243	11.7.5 快速傅里叶变换	287
10.1.2 矩阵相乘	243	11.8 小结	293
10.1.3 矩阵 - 向量相乘	244	推荐读物	293
10.1.4 矩阵与线性方程组的关系	244	参考文献	293
10.2 矩阵乘法的实现	244	习题	295
10.2.1 算法	244	第12章 搜索和优化	298
10.2.2 直接实现	246	12.1 应用和技术	298
10.2.3 递归实现	248	12.2 分枝限界搜索	299
10.2.4 网格实现	249	12.2.1 顺序分枝限界	299
10.2.5 其他矩阵相乘方法	252	12.2.2 并行分枝限界	300
10.3 求解线性方程组	253	12.3 遗传算法	301
10.3.1 线性方程组	253	12.3.1 进化算法和遗传算法	301
10.3.2 高斯消去法	253	12.3.2 顺序遗传算法	303
10.3.3 并行实现	254	12.3.3 初始种群	303
10.4 迭代方法	256	12.3.4 选择过程	305
10.4.1 雅可比迭代	256	12.3.5 后代的生成	306
10.4.2 快速收敛方法	260	12.3.6 变异	307
10.5 小结	263	12.3.7 终止条件	307
推荐读物	263	12.3.8 并行遗传算法	308

12.4 连续求精	311
12.5 爬山法	311
12.5.1 银行业务应用问题	312
12.5.2 爬山法在银行业务中的应用	314
12.5.3 并行化	315
12.6 小结	315
推荐读物	315
参考文献	315

习题	317
----------	-----

附 录

附录A 基本的PVM例程	323
附录B 基本的MPI例程	328
附录C 基本的Pthread例程	333
附录D 并行计算模型	337
索引	346

第一部分 基本技术

- 第1章 并行计算机
- 第2章 消息传递计算
- 第3章 易并行计算
- 第4章 划分和分治策略
- 第5章 流水线计算
- 第6章 同步计算
- 第7章 负载均衡与终止检测
- 第8章 共享存储器编程

第1章 并行计算机

本章首先叙述并行编程概念以及为实现此概念所必需的计算机系统的类型，然后将探讨并行计算机提高执行速度的前景。

1.1 对计算速度的需求

对计算机系统超过当前可能提供的计算速度的需求总是不断增长。需要高计算速度的领域包括科学和工程问题的数值建模和模拟，这样的问题常需要对大量数据进行很多次重复计算以得到有效结果。计算必须在“合理”的时间内完成。在制造业领域，工程计算和模拟如果可能的话必须在几秒或几分钟内完成。在设计环境中，如果进行一次模拟需要两星期才能得到结果，这通常是不可接受的。因为只有模拟完成时间足够短时，设计者方可高效地工作。当系统变得更复杂时，就需要增加更多时间对系统进行模拟。有一些应用问题的计算对时间有特定的期限(最著名的要数气象预报)，花两天时间来获取当地第二天精确的天气预报将使得这种预报毫无意义。某些研究领域，如对大型DNA结构建模以及进行全球天气预报均具有巨大挑战性问题。所谓巨大挑战性问题是指无法用当今计算机在合理的时间内完成求解的那些问题。显然，解决这一问题用10年的执行时间总是不合理的。

由计算机进行的天气预报(数值气象预报)是一个被广泛引用、需要功能非常强大计算机的一个典型例子。大气建模是通过将大气层划分成三维区域或单元完成的，该模型使用相当复杂的数学方程来估测各种影响。实质上，在某一时间间隔内的每个单元中的条件(温度、压力、湿度、风速和风向等)是通过使用在前一时间间隔中的条件进行计算得到的，每个单元的计算要重复许多次以模拟时间的推移。使此模拟具有有效性的关键在于必须有足够多的单元数。为预报数天的天气，由于大气层会受远距离事件的影响，因此必须有较大的覆盖范围。假定我们将整个地球大气层分成大小为1英里×1英里×1英里的许多单元，布满从地面至10英里的高度(即10个单元高)，粗略地估算约有 5×10^8 个单元。假设每个单元的计算需要200次浮点运算，则在一个时间步中就必须完成 10^{11} 次浮点运算。如果我们想预报10天以上的天气，使用的时间间隔为10分钟，此时就需要 10^4 个时间步和总计 10^{15} 次浮点运算，那么对一个运算速度为100 Mflops(10^8 浮点运算/秒)的计算机将需要 10^7 秒即超过100天才能完成上述计算。要想在10分钟内完成这一计算，我们就需要运算速度为1.7 Tflops(1.7×10^{12} 浮点运算/秒)的计算机。

另一个需要巨大计算量的应用问题是预测太空中天体的运动。每个天体由于万有引力而互相吸引，这些远距离的力可用简单公式加以计算(参见第4章)，而每个天体的运动可以通过计算天体所受合力的计算加以预测。如果共有 N 个天体，则每个天体将总共需计算 $N-1$ 个力，约 N^2 次计算。在确定这些天体的新位置后，必须重复这种计算。图1-1中示出了一个快照，是一名大学生求解此问题所得到的结果，由于该问题是作为指定的编程习题的，故而只给定了几个天体。但是实际上可能要考虑具有大量天体的情况。例如，银河系可能有 10^{11} 个星体，此时就需重复计算 10^{22} 次。即使使用一个仅需 $N \log_2 N$ 次计算的高效的近似算法(但每次计算中含有更多的计算量)，计算的总量仍异常巨大($10^{11} \log_2 10^{11}$)。若在单处理器系统上运算将需要很

长时间,即使每次计算仅需1微秒(10^{-6} 秒,这是非常乐观的估计,因为一次计算中含有多次乘法和除法),则使用 N^2 算法时,迭代就需要 10^9 年,而用 $N \log_2 N$ 算法时,一次迭代也需几乎是一年的时间。

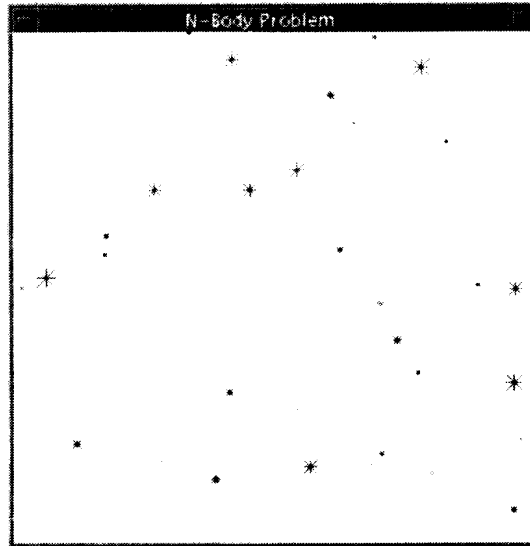


图1-1 由Scott Linsen (北卡罗来纳大学夏洛特分校本科生)完成的天体物理学 N 体模拟

最后,也就是人的本能,不断想像那些超过计算机系统能力的新的应用,从而需要比目前可提供的更高速度。近期的一些应用,如虚拟现实,需要很高的计算速度以实时地获得所需结果。看来不论当前的处理器达到什么样的计算速度,总是会有应用需要更高的计算能力。

提高计算速度的一种方法是用多个处理器协同求解一个问题;这种方法实际上好多年来一直在进行研究。在这种方法中,整个求解问题被分成若干部分,然后每个部分各由一个处理器并行地计算。编写这种形式的程序被称为是并行编程。计算平台,即一台并行计算机,可以是专门设计的、含有多个处理器的计算机或是以某种方式互连的若干台独立的计算机。这种方法将显著地提高性能。基本的想法是, n 台计算机应能提供 n 倍的单机速度,不论当前计算机的速度为多少,可以期待求解的问题将以 $1/n$ 时间完成。当然这是一个理想情况,实际上很难达到。这是因为求解问题经常不能完全分解为各个独立的部分,同时各部分之间必须进行交互,包括计算中的数据传送和同步。尽管如此,仍可达到实质性的改进,这取决于欲求解问题和问题中的并行性程度。对并行计算的要求永远不会终止,这一方面是因为单处理器执行速度的不断升级使并行计算机速度越来越快,另一方面总是有一些具有巨大挑战性的问题,这些问题在当前计算机上求解不可能在合理的时间内完成。

除了可使对现有的求解问题得到加速以外,多计算机/多处理机的使用常常可使一个更大的求解问题或更精确的求解问题能在合理的时间内完成。例如,许多物理现象的计算涉及到将求解问题分成为离散的求解点。天气预报计算包含着将大气层分为三维的求解格点。二维和三维的求解格点出现在许多其他应用中。用多处理机求解时,常允许在给定的时间内计算更多的求解点,从而获得更精确的解。一个相关的因素是多计算机常比单机有更大的总主存储器容量,从而使需要较大主存储器容量的求解问题得到解决。

并行计算机不是新设想,事实上这是一个很古老的想法。例如, Gill 在1958年就写出了有