

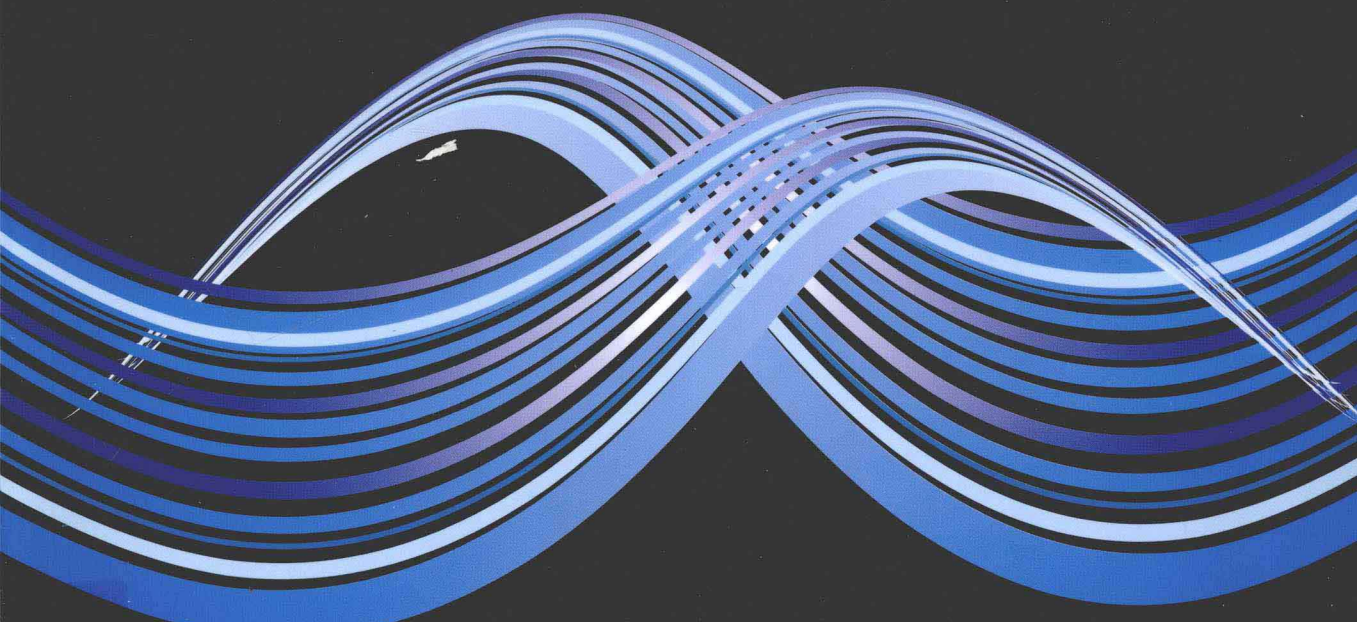
 WILEY
Publishers Since 1807

世界著名计算机教材精选

算法与并行计算

Fayez Gebali 著

都志辉 等 译



ALGORITHMS AND PARALLEL COMPUTING

清华大学出版社

世界著名计算机教材精选

Algorithms and Parallel Computing
算法与并行计算

Fayez Gebali 著
都志辉 等译

清华大学出版社
北 京

Fayez Gebali

Algorithms and Parallel Computing

EISBN: 978-0-470-90210-3

Copyright © 2011 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by John Wiley & Sons Inc. within the territory of the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国 John Wiley & Sons, Inc. 公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾)独家出版发行。未经许可之出口,视为违反著作权法,将受法律之制裁。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字 01-2011-4197 号

本书封面贴有 John Wiley & Sons 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

算法与并行计算/(美)格巴里(Gebali,F.)著;都志辉等译. —北京:清华大学出版社,2012.11

(世界著名计算机教材精选)

书名原文: Algorithms and Parallel Computing

ISBN 978-7-302-29009-4

I. ①算… II. ①格… ②都… III. ①计算机算法—教材 ②并行算法—教材 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2012)第 124045 号

责任编辑:龙启铭

封面设计:何凤霞

责任校对:梁毅

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:17

字 数:423千字

版 次:2012年11月第1版

印 次:2012年11月第1次印刷

印 数:1~3000

定 价:39.00元

产品编号:041695-01

译者序

受到能耗的限制,仅靠 CPU 提升工作频率便可直接提升处理性能的方式已经不再适用。因此设计更多但是频率更低的处理核心便成为处理器发展的方向。多核处理器成为趋势,为了充分利用这些处理核心,必须采用并行方式才可以发掘它们的计算能力。因此并行处理就成了未来提高处理器性能的一种必需的手段。

本书正好可以满足这样一种需求,它提供了关于并行计算与并行算法设计的基本知识与具体的实现案例,可以指导我们设计高效的并行算法。

本书大概可以分为两大部分,第 1~11 章侧重于基本的并行算法与知识的介绍,第 12~21 章侧重于如何利用前面介绍的知识来解决具体的应用问题。其中,第 1 章对本书中出现的几种重要算法:串行算法,并行算法,以及正则迭代算法等进行了介绍。第 2 章介绍了提升单处理器性能的手段。第 3 章介绍了并行计算机的主要类型,包括:共享内存,分布式内存,单指令多数据流,脉动阵列以及多核系统。第 4 章介绍了共享内存多核系统以及紧密相关的两个问题:高速缓存一致性和处理器同步。第 5 章介绍了并行处理器的集中内部互连网络:总线型,星型,环型,以及网络拓扑,探讨了几种更高效的网络类型:交换阵列,多层级网络。第 6 章介绍了用于开发并行应用的并发平台软件工具,包括 Cilk ++、OpenMP 以及统一计算设备架构(CUDA)。第 7 章介绍了特定算法在并行计算机上的实现方法、算法任务级别的流水线处理等。第 8 章介绍了 NSPA 算法的处理方法,NSPA 是指无法归类于串行、并行或者是串并行的算法。第 9 章介绍 z -变换的方法。第 10 章介绍构建迭代算法的依赖图。第 11 章介绍基于计算几何和线性代数概念的迭代算法分析技术。第 12 章探讨的是不同的一维(1D)有限脉冲响应(FIR)数码滤波器的并行处理架构。第 13 章探讨的是不同的二维(2D)和三维(3D)无限脉冲响应(IIR)数码滤波器的并行处理架构。第 14 章探讨的是不同的多采样率抽样器和中断器的并行处理架构。第 15 章探讨的是模式识别算法所需的并行处理架构。第 16 章探讨的是运用于视频压缩的运动估计并行处理架构。第 17 章探讨的是不同的有限域的 2^m 阶伽罗瓦域乘法的并行处理架构。第 18 章探讨的是不同的有限域的 2^m 阶伽罗瓦域除法的并行处理架构。第 19 章探讨的是不同的快速傅里叶变换算法的并行处理架构。第 20 章探讨了线性方程的求解系统。第 21 章讨论了使用有限差分法(FDM)来求解偏微分方程的不同的并行处理架构。

本书不是一本特别初级的入门教材,但是有一定计算机专业基础的读者阅读起来是不会有很大障碍的。另外对于需要直接了解某些具体案例的并行处理方法的读者,也可以从本书找到答案或者启发。推荐计算机、电子电气工程等相关专业的高年级本科生、研究生以及从事相关研究的科研人员阅读本书。

在翻译过程中我们尽量追求在忠于原著的同时符合中文的语言习惯,让读者在阅读时感到流畅和自然,对于书中出现的大量的专业术语我们尽量遵循标准的译法,在某些新的概念和名词出现之时我们尽量标注英文原文,方便读者的理解对照。为了尽快完成本书的翻译,有多位本研究组的人员并行参加了不同章节的翻译工作,他们是何禹、刘渊、邬凌超、王

汉、芮然、高鹏、杨全、权思及、王璟尧、于卓然。都志辉负责全书的通稿与审校,王汉在参与翻译工作之余还协助我做了大量的协调、组织、排版等方面的工作,在此特表示感谢。

本书的翻译工作受到了国家自然科学基金(61073008)和北京市自然科学基金(4122039和4082016)的支持,在此表示感谢。

特别感谢清华大学出版社的龙启铭编辑,为本书的出版做了大量具体细致的工作,也感谢他对于我一再推迟交稿日期的理解与宽容。

由于本书涉及的领域与内容比较广,虽然我们尽力做到翻译的精准,但是一定会存在不妥甚至错误的地方,恳请读者批评指正。

都志辉

2012年8月于清华园

作 者 序

关于本书

用如今的软件进行并行编程,开发工具所能达到的表现和硬件的潜在表现之间有一种“软件差距”。编程工具需要程序员的人工干预才能将代码并行化。本书的目的是为程序员提供在算法、串行代码以及迭代中寻求并行化时所需的技巧。并行计算在之前只是少数精英企业或集团才能支付得起的昂贵系统,如今已经几乎应用于每一台计算机上。我们可以在笔记本上,台式机上,甚至是智能手机的嵌入式系统中找到并行计算的身影。传统的并行计算旨在进行天气预报、风洞模拟,计算生物学以及信号处理等大型应用。如今,几乎所有的应用程序都涉及并行处理器的操作,而并行处理器已经成为现代计算机的标配。

并行算法可以是为了某一特殊目的特别定制的,也可以是通用的,并行算法的开发从技术角度上可以分为数个层次,如并行程序开发、编译器并行化、多线程操作系统或是超大规模多处理器系统。本书主要关注第一类:为了特殊目的定制的算法及其处理器架构的并行化。我们称此类系统为“核心加速器”。本书的第1~4章是关于上述内容的基础知识,随后的章节中将精选数个案例用于探讨分析。

虽然超大规模集成电路(VLSI)技术能够将更多的处理器整合在同一芯片上,但并行编程技术并没有跟上科技进步的脚步。例如,如今并行硬件系统的应用主要还是为加速特殊目的算法定制而成。这是出于两个实际问题:

(1) 在当前的计算机平台中多核系统是主流。

(2) 如在数据加密/解密、图形处理、数字信号处理、滤波等应用中使用的算法多数都是简单串行算法。

介绍本书更简单的方法是先说一说本书中没有涉及哪些内容。本书没有给出体系结构,并行计算机和通用并行算法的具体实现细节。这3个主题中的任何一个都需要很多的优秀的标准教材来详细讲解,例如“Computer Organization and Design”(D. A. Patterson and J. L. Hennessy),“Parallel Computer Architecture”(D. E. Culler, J. P. Singh, A. Gupta),“Introduction to Algorithms”(T. H. Cormen, C. E. Leiserson, R. L. Rivest)。我希望多数读者都能在阅读本书之前学习这些伟大的作品,若我之前列出的书单不够翔实明确也请大家见谅。

本书阐述了如何为实现特定的算法,系统地构建并行体系结构。此书中出现的算法也可以用于实现通用算法,无论并行与否。

本书适合计算机工程、电气工程和计算机科学行业的研究人员和研究生。本书需要用到线性代数和数字信号处理的基础知识。

本书的目的是:

(1) 对几种表述方法进行解释,如并行算法的依赖图或依赖矩阵。

(2) 探讨进程任务的调度方案,同时要符合输入和输出数据时序,可以流水线处理数

据,并能将数据广播到所有处理器。

(3) 探讨处理单元的任务分配方案。

章节预览

第 1 章主要内容包括:

- 定义了本书中出现的几种重要算法,即串行算法、并行算法以及正则迭代算法。
- 为并行计算机设计并行算法时需要考虑两者间的联系。
- 用加速因子和通信开销来量化并行化带来的优势。
- 举例说明并行计算机的两种应用。

第 2 章主要介绍提升单处理器性能的手段:

- 提升时钟频率。
- ALU(算术逻辑单元)结构并行化。
- 流水线作业。
- 超长指令字(VLIW)。
- 超大规模计算以及多线程。

第 3 章主要介绍并行计算机的主要类型:共享内存、分布式内存、单指令多数据流(SIMD)、脉动阵列以及多核系统。

第 4 章主要介绍共享内存多核系统以及紧密相关的两个问题,即高速缓存一致性和处理器同步。

第 5 章主要介绍并行处理器的内联网络的主要类型,即总线型、星型、环型以及网络拓扑,探讨了几种更高效的网络类型,如开关阵列、多层级网络。

第 6 章主要介绍了用于开发并行应用的并发平台软件工具,包括 Cilk++、OpenMP 以及统一计算设备架构(CUDA)。这些工具都只能解决相对简单的数据依赖问题,数据的完整性,正确性和任务执行的时间调度等问题都应该由编程者自己来决定。本书中出现的一些技巧能帮助编程者解决串行算法和正则递归算法中出现的一些问题。

第 7 章主要内容包括:

- 特定算法在并行计算机上的实现方法。包括独立环的调度,非独立环的传播,非独立环的展开,分治策略。
- 算法任务级别的流水线处理。
- 利用坐标旋转数码计算机(CORDIC)实现算法。

第 8 章主要介绍了非串行算法(NSPA)的处理方法,非线性算法是指无法归类于串行、并行或者是串并行的算法。多数通用算法都属于 NSPA,它们的并行性不明显,任务依赖图比较复杂。本章教授了提取此类算法中的并行部分的一种正规且有力的方法,此方法的好处在于将一个算法运行在一台并行计算机上时利用该方法可以有效提升算法的效率,并且此方法还可以估算出达到最大执行加速比所需的并行处理器的数量。

在该方法中,我们用任务 W 、并行度 P 和深度 D 来量化某个 NSPA 的并行化程度。

第 9 章主要内容包括:

- z -变换的方法。

- z -变换应用于数码滤波器以及不同并行平台上多采样率系统的实现。

此类应用一般都是在 z 域进行的,所以此类应用的软硬件实现也要在 z 域中进行研究。

第 10 章主要内容包括:

- 构建迭代算法的依赖图。
- 此方法应用于不超过索引数量不超过 3 个的迭代算法。
- 利用该依赖图,算法中的任务可以自动分配至软件线程和硬件处理器。

第 11 章主要介绍了基于计算几何和线性代数概念的迭代算法分析技术。此方法适用于当迭代算法的索引数多于 3 个的情况。例如二维(2D)和三维(3D)数码滤波器。对于此类算法,我们用一个多维空间中的凸壳和相关的依赖矩阵来表示算法中的每一个变量。该矩阵的零向量空间帮助我们对不同的并行软件线程和硬件处理单元进行分配和时间调度。

第 12 章探讨的是不同的一维(1D)有限脉冲响应(FIR)数码滤波器的并行处理架构。首先,运用 11 章中的计算几何的知识我们可以构建此系统的硬件结构。之后利用第 9 章中的 z -变换了实现可能的软件并行处理架构。

第 13 章探讨的是不同的二维(2D)和三维(3D)无限脉冲响应(IIR)数码滤波器的并行处理架构。我们运用 z -变换来实现此类滤波器。

第 14 章探讨的是不同的多采样率抽样器和中断器的并行处理架构。此类算法在无线通信中广泛应用。运用的技术是第 10 章的依赖图。

第 15 章探讨的是模式识别算法所需的并行处理架构。此类算法在无线通信中广泛应用。运用的技术是第 10 章的依赖图。

第 16 章探讨的是运用于视频压缩中的不同的运动估计的并行处理架构。为了适应这种复杂的算法,我们将问题简化,分为几个层次。运用的技术是第 10 章的依赖图。

第 17 章探讨的是不同的有限域的 2^m 阶伽罗瓦域乘法的并行处理架构。运用的技术是第 10 章的依赖图。

第 18 章探讨的是不同的有限域的 2^m 阶伽罗瓦域除法的并行处理架构。运用的技术是第 10 章的依赖图。

第 19 章探讨的是不同的快速傅里叶变换算法的并行处理架构。会运用流水线技术来实现该算法。

第 20 章探讨线性公式的求解系统。该系统有直接和间接两种实现方法。本章讨论的是如何将前置直接替换技术并行化。中间会涉及一个利用 Givens 公式将稠密矩阵转换为等价的三角矩阵的算法。本章也会讨论逐次超松弛(SOR)简介求解方法的并行化。

第 21 章讨论使用有限差分法(FDM)来求解偏微分方程的不同的并行处理架构。在许多工程和科学应用中都对此类方程的求解有很大的需求。

鸣 谢

我希望在此表达我深深的谢意。

感谢埃及的 Ain Shams 大学的 M. W. El-Kharashi 博士在本书筹备过程中提供的建议和鼓励。我也在此对下述的同僚表达我诚挚的谢意,没有你们的帮助我不可能完成此书:

Dr. Esam Abdel-Raheem

Dr. Turki Al-Somani

University of Windsor, Canada
Dr. Atef Ibrahim
Electronics Research Institute, Egypt
Mr. Brian McKinney
ICESoft, Canada
Dr. Mohamed Rehan
British University, Egypt

Al-Baha University, Saudi Arabia
Dr. Mohamed Fayed
Al-Azhar University, Egypt
Dr. Newaz Raiq
ParetoLogic, Inc., Canada
Dr. Ayman Tawik
Ajman University, United Arab Emirates

忠告与建议

本书覆盖了关于并行计算的许多方面。很有可能在本书中出现各种错误与疏漏。万望其他领域的学者和工程师能就本书的内容,行文和章节布置等方面提出宝贵意见。同时也欢迎来信提出问题和案例(若非原创请标明引用出处)。

请您将您的宝贵意见发送电邮至:fayez@uvic。

来电、来信或传真请按照以下方式联络:

Dr. Fayez Gebali

Electrical and Computer Engineering Department

University of Victoria, Victoria, B. C., Canada V8W 3P6

电话: 250-721-6509

传真: 250-721-6052

缩 写 表

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
ALU	arithmetic and logic unit
AMP	asymmetric multiprocessing system
API	application program interface
ASA	acyclic sequential algorithm
ASIC	application-specific integrated circuit
ASMP	asymmetric multiprocessor
CAD	computer-aided design
CFD	computational fluid dynamics
CMP	chip multiprocessor
CORDIC	coordinate rotation digital computer
CPI	clock cycles per instruction
CPU	central processing unit
CRC	cyclic redundancy check
CT	computerized tomography
CUDA	compute unified device architecture
DAG	directed acyclic graph
DBMS	database management system
DCG	directed cyclic graph
DFT	discrete Fourier transform
DG	directed graph
DHT	discrete Hilbert transform
DRAM	dynamic random access memory
DSP	digital signal processing
FBMA	full-search block matching algorithm
FDM	finite difference method
FDM	frequency division multiplexing
FFT	fast Fourier transform
FIR	finite impulse response
FLOPS	floating point operations per second
FPGA	field-programmable gate array
GF(2^m)	Galois field with 2^m elements

GFLOPS	giga floating point operations per second
GPGPU	general purpose graphics processor unit
GPU	graphics processing unit
HCORDIC	high-performance coordinate rotation digital computer
HDL	hardware description language
HDTV	high-definition TV
HRCT	high-resolution computerized tomography
HTM	hardware-based transactional memory
IA	iterative algorithm
IDHT	inverse discrete Hilbert transform
IEEE	Institute of Electrical and Electronic Engineers
IIR	infinite impulse response
ILP	instruction-level parallelism
I/O	input/output
IP	intellectual property modules
IP	Internet protocol
IR	instruction register
ISA	instruction set architecture
JVM	Java virtual machine
LAN	local area network
LCA	linear cellular automaton
LFSR	linear feedback shift register
LHS	left-hand side
LSB	least-significant bit
MAC	medium access control
MAC	multiply/accumulate
MCAPI	Multicore Communications Management API
MIMD	multiple instruction multiple data
MIMO	multiple-input multiple-output
MIN	multistage interconnection networks
MISD	multiple instruction single data stream
MIMD	multiple instruction multiple data
MPI	message passing interface
MRAPI	Multicore Resource Management API
MRI	magnetic resonance imaging
MSB	most significant bit
MTAPI	Multicore Task Management API
NIST	National Institute for Standards and Technology
NoC	network-on-chip

NSPA	nonserial-parallel algorithm
NUMA	nonuniform memory access
NVCC	NVIDIA C compiler
OFDM	orthogonal frequency division multiplexing
OFDMA	orthogonal frequency division multiple access
OS	operating system
P2P	peer-to-peer
PA	processor array
PE	processing element
PRAM	parallel random access machine
QoS	quality of service
RAID	redundant array of inexpensive disks
RAM	random access memory
RAW	read after write
RHS	right-hand side
RIA	regular iterative algorithm
RTL	register transfer language
SE	switching element
SF	switch fabric
SFG	signal flow graph
SIMD	single instruction multiple data stream
SIMP	single instruction multiple program
SISD	single instruction single data stream
SLA	service-level agreement
SM	streaming multiprocessor
SMP	symmetric multiprocessor
SMT	simultaneous multithreading
SoC	system-on-chip
SOR	successive over-relaxation
SP	streaming processor
SPA	serial-parallel algorithm
SPMD	single program multiple data stream
SRAM	static random access memory
STM	software-based transactional memory
TCP	transfer control protocol
TFLOPS	tera floating point operations per second
TLP	thread-level parallelism
TM	transactional memory
UMA	uniform memory access

VHDL	very high-speed integrated circuit hardware description language
VHSIC	very high-speed integrated circuit
VIQ	virtual input queuing
VLIW	very long instruction word
VLSI	very large-scale integration
VOQ	virtual output queuing
VRQ	virtual routing/virtual queuing
WAN	wide area network
WAR	write after read
WAW	write after write
WiFi	wireless fidelity

目 录

第 1 章 引言	1
1.1 概述	1
1.2 自动并行编程	1
1.3 算法	3
1.3.1 算法的有向图.....	3
1.3.2 算法的邻接矩阵 A	4
1.3.3 基于子任务的依赖关系对算法进行分类.....	5
1.3.4 串行算法.....	6
1.3.5 并行算法.....	6
1.3.6 SPA	6
1.3.7 NSPA	7
1.3.8 RIA	8
1.3.9 并行算法实现.....	8
1.4 设计并行计算系统	9
1.5 并行算法和并行体系结构.....	10
1.6 并行算法与并行体系结构相关.....	10
1.7 算法的实现：两个方面的问题	11
1.8 衡量并行计算的优势.....	11
1.8.1 加速比	11
1.8.2 通信开销	12
1.8.3 计算加速比和通信开销	12
1.9 针对多处理器系统的 Amdahl 法则	14
1.10 Gustafson-Barsis 法则.....	15
1.11 并行计算的应用	16
1.11.1 气象建模	16
1.11.2 CT	17
1.11.3 计算流体力学(CFD)	18
1.12 习题	18
第 2 章 增强单处理器的性能	21
2.1 概述.....	21
2.2 提高处理器的时钟频率.....	21
2.3 ALU 的并行化	22
2.4 使用分级存储器体系.....	24
2.4.1 内存-高速缓存之间的操作	25

2.4.2	高速缓存的设计	26
2.4.3	分层高速缓存	26
2.4.4	将内存块映射到高速缓存行	26
2.4.5	关联映射	27
2.4.6	组相关映射	28
2.4.7	缓存容量对缓存命中率的影响	28
2.5	流水线作业	28
	估算流水线作业的速度	29
2.6	超长指令字(VLIW)处理器	32
2.7	指令级并行(ILP)和超标量处理器	33
	2.7.1 真实数据依赖:写后读(RAW)	34
	2.7.2 程序的依赖关系	35
	2.7.3 资源冲突	35
	2.7.4 输出依赖性:写后写(WAW)	35
	2.7.5 反依赖:读后写(WAR)	36
2.8	多线程处理器	36
2.9	习题	37
第3章	并行计算机	39
3.1	概述	39
3.2	并行计算	39
3.3	共享内存的多处理器(统一内存访问 UMA)	40
3.4	分布式内存多处理器(非统一内存访问 NUMA)	41
3.5	SIMD 处理器	41
3.6	脉动式处理器	42
3.7	集群计算	44
3.8	网格计算(云计算)	44
3.9	多核系统	44
3.10	流多处理器	46
3.11	并行处理器之间的通信	48
	3.11.1 通信类型	48
	3.11.2 消息传递(MP)通信机制	49
3.12	并行体系结构总结	50
3.13	习题	50
第4章	共享内存多处理器	52
4.1	概述	52
4.2	高速缓存一致性和内存一致性	53
	4.2.1 目录协议	56
	4.2.2 Snoopy 协议	57

4.3	同步和互斥	57
4.3.1	同步: 锁机制	58
4.3.2	同步: 互斥量	59
4.3.3	同步: 栅栏	60
4.3.4	同步原语的对比	61
4.4	习题	62
第5章	互连网络	63
5.1	概述	63
5.2	逻辑拓扑结构中互连网络的分类	63
5.2.1	总线型	63
5.2.2	星型	64
5.2.3	环型	64
5.2.4	网型	64
5.2.5	交叉开关网络	65
5.2.6	交叉开关网络的连接及仲裁	66
5.2.7	多级互连网络	66
5.2.8	榕树(Banyan)网络	66
5.2.9	树型网络	67
5.2.10	随机拓扑网络	68
5.3	互联网络交换架构	68
5.3.1	输入队列交换器	69
5.3.2	输出队列交换器	70
5.3.3	共享缓冲区交换器	71
5.3.4	多输入队列交换器	73
5.3.5	多输出队列交换器	73
5.3.6	多输入输出队列交换器	74
5.3.7	VRQ 交换器	75
5.4	习题	76
第6章	并发平台	78
6.1	概述	78
6.2	并发平台	78
6.3	Cilk++	78
6.3.1	Cilk++ 并行循环: cilk_for	79
6.3.2	数据竞争和程序不确定性	80
6.3.3	将串行代码并行化的 Cilk++ 组件	82
6.3.4	使用 Cilk++ 实现矩阵乘法	82
6.4	OpenMP	84
6.4.1	OpenMP 编译指导语句	85
6.4.2	编译指导语句子句	86

6.4.3	OpenMP 负载分配	87
6.4.4	循环指导语句: for	87
6.4.5	循环指导语句: sections	89
6.4.6	运行时库函数	90
6.4.7	环境变量	90
6.4.8	OpenMP 同步	90
6.5	统一计算设备架构(CUDA)	91
6.5.1	定义 CUDA 中的线程、块和网格	93
6.5.2	将函数交付内核执行	94
6.5.3	主机与 CUDA 设备间的通信	95
6.5.4	CUDA 线程的同步与通信	95
6.5.5	内核和网格	95
6.5.6	块	97
6.5.7	线程	97
6.5.8	CUDA C 语言扩展	97
第 7 章	针对并行算法的特别技术	98
7.1	概述	98
7.2	定义算法变量	99
7.3	独立循环调度	99
7.4	依赖循环	100
7.5	针对简单依赖循环的循环分发方法	100
7.6	循环展开	101
7.7	问题划分	101
7.8	分而治之(递归划分)策略	102
7.9	流水线	104
7.10	习题	106
第 8 章	非串行-并行算法	107
8.1	概述	107
8.2	并行化用 DAG 表示的 NSPA 算法	108
8.3	分析 NSPA 的形式化方法	109
	矩阵的幂的意义: 矩阵的连通性	110
8.4	辨别算法中的环	112
8.5	提取串行及并行算法的性能参数	113
8.6	相关定理	114
8.7	串行和并行算法在并行计算机上的性能	116
8.8	习题	116
第 9 章	z-变换分析	118
9.1	概述	118
9.2	z -变换的定义	118