●计算机新技术丛书●

PARALLEL PROGRAMMING PARALLEL PROGRAMMING

并行程序设计

沈志宇 廖湘科 胡子昂 编著

国防科技大学出版社

并行程序设计
沈志宇 廖湘科 胡子昂 编著
出版社

计算机新技术丛书

Parallel Programming

并行程序设计

沈志宇　廖湘科　胡子昂　编著

国防科技大学出版社
·长沙·

图书在版编目(CIP)数据

并行程序设计/沈志宇,廖湘科,胡子昂.—长沙:国防科技大学出版社,1997.9
ISBN 7—81024—450—7

Ⅰ 并行程序设计
Ⅱ 沈志宇 廖湘科 胡子昂
Ⅲ 计算机—并行程序设计—专著
Ⅳ TP311

责任编辑:胡见堂
责任校对:何 晋
封面设计:陆荣斌

国防科技大学出版社出版发行
电话:(0731)4555681 邮政编码:410073
E—mail:gfkdcbs@public.cs.hn.cn
新华书店总店北京发行所经销
湖南大学印刷厂印装
*
开本:787×1092 1/16 印张:12.25 字数:283千
1997年9月第1版第1次印刷 印数:1—2 000
*
ISBN 7—81024—450—7
TP·95 定价:16.00元

内 容 提 要

本书论述并行程序设计的理论和方法，侧重于支持大型科学与工程计算和大规模数据处理的 Fortran 语言并行程序设计。首先介绍了并行程序设计的基础知识和依赖关系分析技术，然后依次讲述了向量计算机并行程序设计、共享存储器多处理机系统并行程序设计、分布存储器并行计算机系统（包括工作站集群）的消息传递并行程序设计和数据并行程序设计，最后讨论了并行程序的调试和性能分析方法。

并行处理已是计算机发展的必然，并行程序设计已是许多学科领域的科技工作者都应掌握的一门技术。本书可作为高性能计算机研制人员和应用人员的参考书，也可作为计算机和有关专业的研究生教材。

前　言

并行处理已是计算机发展的必然。为了追求高性能，计算机厂商推出了越来越多的并行计算机，许多学科或应用领域也在越来越多地使用并行计算机。而高水平的并行程序设计是充分发挥并行计算机性能的关键，高水平的科学研究或工程设计也需要高效率地应用并行计算机。因此，并行程序设计已是许多学科领域的科技工作者都应掌握的一门技术。本书的目的就是介绍并行程序设计的理论和方法，侧重于支持大型科学与工程计算和大规模数据处理的 Fortran 语言并行程序设计。希望它能对学习或研究并行程序设计技术有所帮助。

本书首先介绍了并行程序设计的基础知识和依赖关系分析技术，然后论述了各种主流并行机型的并行程序设计技术，包括向量计算机并行程序设计、共享存储器多处理机系统并行程序设计、分布存储器并行计算机系统（或工作站集群）的消息传递并行程序设计和数据并行程序设计，最后讨论了并行程序设计的调试和性能分析方法。

本书既反映了国际上这一领域的最新研究成果，又总结了我们多年来从事并行程序设计语言、并行编译技术和并行程序设计技术研究的一些认识和经验。

本书注意了基础知识和专门技术并重，力求讲清楚并行程序设计的基本概念和基本方法，同时也具体介绍了典型系统的并行程序设计方法。

本书的读者对象是并行程序设计人员，高性能计算机研制人员和应用人员，以及计算机等有关专业的研究生。

本书的第 1、2、3、4、7 章由沈志宇撰写，第 5 章由廖湘科撰写，第 6 章由胡子昂撰写。全书由沈志宇统稿。

本书的撰写得到了胡守仁教授、李晓梅教授的关心和指教，也得到了国防科技大学计算机学院软件工程研究室许多同事们的帮助，特别是得到了国防科技大学计算机学院和国防科技大学出版社的大力支持，在此一并表示感谢。

由于学识有限，错误或不足在所难免，欢迎读者批评指正。

作　者

1996 年 6 月

目　录

4 共享存储器多处理机系统并行程序设计

5 分布存储器并行系统的消息传递并行程序设计

1 并行程序设计基础知识

并行处理是70年代以来计算机发展的一个重要特征，也是计算机性能提高的一个关键原因。并行计算已经成为科学研究和工程设计的一个有力手段。并行处理是由并行体系结构技术和并行程序设计技术来支持的。广义地说，并行程序设计技术包括并行程序设计语言、并行编译技术、并行程序设计环境技术、并行算法，以及在上述技术支持下的并行编程技术。本书主要面向科学研究和工程技术人员，介绍在并行计算机上编写、调试和运行并行程序的基础知识和基本方法。

1.1 引　　言

1.1.1 现代科学技术对计算能力的需求

可以说，科学技术的发展和人类社会的进步是随着人类计算能力的提高而不断发展和进步的。在远古时代，人们只能结绳记事，对自然的认识就极其有限。后来人们逐渐认识了一些计算的规则，创立和发展了数学这门学问，发明了算盘等计算工具。到人类进入工业时代后，又发明了机械式计算器。这些都在不同程度上促进了社会的进步。1946年第一台电子计算机ENIAC问世，50年来计算机科学和技术可以说是自然科学中发展最快的一个领域。现在人类社会正逐渐步入信息时代，各个领域都在应用计算机。最快的计算机系统每秒已经可以完成一万亿次运算操作，科学家们正在用高性能计算机作出中长期天气预报、探测地下石油资源、预报慧星和木星的相撞、设计既美观又符合空气动力学原理的小轿车、研制更有效的药物等。高性能计算也与我们的社会生活息息相关。从电话、银行、航空，到动画、电子邮件，高性能计算正在改善我们的生活。

科学计算已经成为当代科学技术研究的重要方法之一。科学技术研究的两个传统方法是实验方法和理论方法。但这两个方法对所研究的问题的时间和空间尺度都受到很大限制，不能满足当代科学技术发展的需要。科学计算能在相当程度上突破这个限制，使人们得以研究过去无法研究的问题，并已获得重大成果。如长征二号捆绑式火箭的模拟、高精度π值的计算等。

科学技术的发展对计算能力有着巨大的、不断增长的需求。例如，中期数值天气预报的时效每增加一天，计算机的速度就要提高一个数量级左右。做三天的数值天气预报需要每秒百万次运算操作的计算机，四天预报要千万次的，五天预报就要一亿次的。世界各国一直把计算能力看作是关系到科技进步、经济增长和国家安全的重要问题。世界上第一台计算机就是为解决美国陆军弹道研究的急需而研制的。几十年来，不论哪个国家，最先进

的计算机总是先用于国防科研。高性能计算机和高性能计算机网络现在已经成为综合国力的重要标志之一。为了促进这些技术的发展，许多国家都投入了大量的人力物力。

美国从1992年开始实行为期五年的高性能计算与通信总统计划(HPCC)。实施这一计划的原因是因为美国认识到有许多"巨大挑战"(grand challenge)问题需要有高性能计算与通信能力才能解决。到近两年，这些问题在HPCC计划中又改称为"国家挑战"(National challenge)问题，更可见这些问题的解决对国家实力和地位的重要。除了有关国家安全的一些问题没有详细列出外，HPCC计划几年来分别列出的挑战性问题如下：气象模型、空气污染、臭氧消耗、海洋模型、海洋环流、流体湍性、燃烧系统、催化作用、半导体模型、蛋白质结构设计、数字解剖、人类器官和骨骼模型、细胞模型、酶活性研究、遗传工程、视觉与识别、高速民用运输机、先进工业制造技术、数字图书馆、高等教育和国民教育、卫生保健计划、危机和紧急事件处理、能源管理、电子商务等。以上问题涉及的学科包括量子化学、统计力学、相对论物理学、宇宙学、天体物理学、计算流体力学、材料学、生物学、药物学、医学、遗传学等。科学家们认为这些都需要具有每秒万亿次浮点操作、内存容量万亿字节和处理机与内存之间每秒万亿字节传输带宽性能的计算机(即所谓的3T(Trillion)机器)，甚至更高性能的计算机。

HPCC计划早期主要支持以下四方面的研究：高性能计算系统、先进的软件技术与算法、国家研究与教育网络、基础研究与人才资源。到1995年又增加了一个方面，即信息基础设施技术和应用。美国对HPCC计划的投资情况如表1.1所示。

表1.1 HPCC计划的投资情况

财政年度	投资(亿美元)	增加幅度
1992	6.54	
1993	7.96	22%
1994	9.12	15%
1995	10.95	20%
1996	11.66(预计)	6%

但是，今天的高性能计算机从性能上说，仍然落后于需求；从使用的技术要求上说，又超前了目前普遍的程序设计技术水平。因此，并行程序设计技术和理论的研究成为一个当务之急的课题，成为一个对科学技术发展和社会进步有重大影响的课题。

1.1.2 并行处理的发展过程

和世界上许多事物一样，为了追求高性能，计算机及其程序设计也由简单走向了复杂。从标量机到大规模并行机，从串行程序设计到大规模并行程序设计的发展大致过程如图1—1所示。这

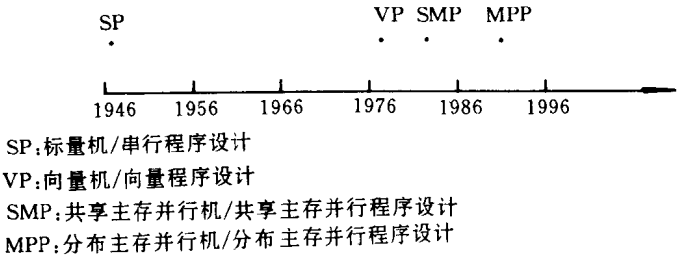

图1—1 程序设计的发展过程

个发展过程是以基本能实际应用的时间为标准的。计算机科学对并行处理的探索可以追溯到50年代。1950年，计算机之父Von Neumann提出了“自复制细胞自动机”的概念。1958年Steven Unger提出了构造两维SIMD阵列机的设想，1963年曾按这种构想提出了两种方案：Solomon系统以及Illiac Ⅲ，但均以失败而告终。1972年美国Illinois大学与Burroagh公司合作研制的Illiac Ⅳ系统可以说是分布主存大规模并行机的鼻祖。该机原拟做四个象限，每个象限由8×8个处理单元构成，每个处理单元可以和上下左右四个处理器通讯。这种设计思想对多处理机阵列结构的研究及设计产生了极大的影响。但由于当时硬件水平所限，Illiac Ⅳ只做了一个象限，且不太成功。此后，人们的注意力就转向了适度并行。

1976年美国Cray公司研制成功了世界上第一台实用的向量巨型机Cray—1。在此基础上，1983年Cray公司推出了四个向量处理机共享主存的Cray X—MP系统，成为世界上第一台成功的适度并行计算机。

80年代中期，随着超大规模集成电路技术的发展，单片微处理器性能的提高和并行处理技术的进步，就产生了新一代的分布主存大规模并行机。例如美国Thinking Machine公司1986年研制成功的CM—1和1987年的CM—2系统，美国nCUBE公司的nCUBE—1、nCUBE—2系统等。

90年代以来，分布主存并行发展很快，同时共享主存并行机也不断有进步，特别是多处理机服务器的推广应用增强了共享主存并行处理技术的生命力。1995年Convex公司推出的SPP 1200并行机采用了分布主存超级结点的体系结构，在每个超级结点上有四个共享主存的处理器，代表了共享主存和分布主存并行处理技术结合的一种趋势。

所以，并行处理技术经历了分布主存并行(不成功)—向量并行—共享主存并行—分布主存和共享主存并行并存这样一个历程。我们认为分布主存并行和共享主存并行这两类技术将继续共存，而这两类技术的结合将是一种有可能获得更高性能计算能力的方向。

现在的超级计算机和高性能计算机必定是并行机，虽然它们可能采用了不同的并行处理技术。这是因为串行计算机受到了主频、集成电路、组装工艺、冷却等方面的限制，计算速度难以得到令人满意的提高。也是因为人们对计算能力的需求是无止境的，尽管单机速度也一直在提高，将它们构成多机并行系统就能获得更快的速度。

并行处理、并行程序设计都还是一门发展中的技术，而且发展很快。目前的研究前沿是大规模并行处理和大规模并行程序设计。向量处理及其程序设计、共享主存多处理机并行处理及其程序设计已经基本成熟，但在共享主存多处理机并行处理及其程序设计技术方面人们在进一步努力，以求更好地实现自动并行化。共享主存和分布主存并行处理和并行程序设计技术的结合则是将来可能的发展方向。

1.1.3 并行程序设计概论

并行程序设计就是根据要解决的问题的有关并行特征和可能使用的并行计算机的特性，选用适当的并行算法和并行程序设计技术，编写出正确高效的并行程序。

实现并行程序设计有三个要素：

- 并行体系结构；

· 并行系统软件；

· 并行算法。

并行程序设计可以从四个层次来实施：

· 算法设计，对于一个特定的问题选择最适当的并行算法。

· 数据结构设计，对于一个特定的问题和算法选择最适当的数据结构。

· 语言选择，选择适合于特定机器、算法和数据结构的语言。

· 程序编写和优化，采用恰当的编程和优化技术。

并行程序设计面临的基本问题：

· 并行性分析

程序的控制流和数据流如何？

哪些操作可以并行执行？

并行操作的特性如何？

· 数据分布

如何分布才能最多地开发并行性？

如何分布才能保证负载平衡？

静态分布还是动态分布？

· 并行划分

如何组织处理机？

如何将并行任务分配到处理机上去执行？

静态调度还是动态调度？

· 同步通讯的组织

选择哪种同步、通讯机制？

如何使同步、通讯的开销最小？

如何保证程序的正确性、确定性？

各种实际应用问题都以不同形式存在着不同程度的并行性。典型的应用问题是由若干个相互作用的子问题组成的，每个子问题可以看作一个相对独立的模块。著名计算机应用专家 G. C. Fox 在 1989 年对 400 余篇论文中所研究的 84 个实际应用问题进行了分析归纳，得出如下五种类型的子问题：

1. 同步型：对大量数据做有规律的相同操作，如典型的矩阵运算、有限差分等。

2. 松散同步型：对大量无规则数据进行操作，而且计算过程按时间或步(step)分段进行，表现为宏观上的时间同步，如对不规则区域或异种数据区域的迭代计算。

3. 异步型：主要表现为时间不规则性，如用事件驱动模型实现的模拟程序。

4. 基本并行型：由不相关联的部分组成，每部分可独立进行，这类问题在数据采集及预处理阶段经常出现。

5. 松散同步复合型：由同步型模块组成，而模块之间的关系是异步的。

不同类型的应用问题适合于在不同类型的并行计算机上求解，这样才能充分开发每个应用问题内部所固有的并行性。

对并行计算机的体系结构可以从不同角度作出如下分类：

· 向量机和非向量机。
· 共享主存并行和分布主存并行机。
· 同构并行机和异构并行机。
· 紧耦合系统和松耦合系统。

并行算法则可从如下两个角度来分类：

· 数值并行算法和非数值并行算法。
· 同步并行算法和异步并行算法。

并行程序设计的模式和技术也可以从如下几个角度来分类：

· 数据并行和控制并行模式。
· SIMD 和 MIMD 模式。
· 向量、共享变量、消息传递、面向对象以及函数和逻辑并行程序设计技术，对于科学和工程计算而言，主要是应用前三种技术。

下面我们以应用问题的同步和通讯量两个特征为度量单位，将应用问题分为四个象限来讨论它们与体系结构、并行算法及并行程序设计之间的一般关系。

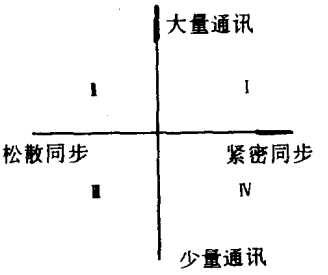

图 1－2　从同步和通讯两个特征来分析应用问题

第Ⅰ、Ⅳ象限的问题具有 SIMD 和数据并行的特征，而Ⅱ、Ⅲ象限的问题具有 MIMD 和控制并行的特征。

从并行化的效率，即并行开销的大小来看，无论哪个象限的问题，使用共享主存并行机都要优于分布主存并行机。但由于共享主存并行机受处理机数目限制，当应用问题的计算量相当大时，使用具有大量处理结点的分布主存并行机就能获得比共享主存并行机更快的运算速度。分布主存并行机对Ⅲ、Ⅳ象限的问题更适应一些，对Ⅰ、Ⅱ象限问题则通讯开销较大。

第Ⅰ、Ⅳ象限问题适应 SIMD 方式的并行机和并行程序设计，宜采用同步并行算法，第Ⅱ、Ⅲ象限的问题则适应 MIMD 方式的并行机和并行程度设计，宜采用异步并行算法。

本书讲述面向科学和工程计算的并行程序设计的方法和技术。本章第 2 节介绍并行程序设计的基础知识，第 3 节讨论并行计算模型。第 2 章讲述依赖关系分析技术。第 3 章介绍向量计算机并行程序设计方法。共享存储器多处理机系统并行程序设计技术在第 4 章中论述。第 5 章的主题是分布存储器并行系统的消息传递并行程序设计。第 6 章是分布存储器并行系统的数据并行程序设计。最后，第 7 章介绍并行程序的调试和性能分析。

1.2　并行程序设计基础知识

1.2.1　并行体系结构简介

并行体系结构是并行处理的基础，各种并行程序设计机制也都是针对不同的并行体

系结构的。并行体系结构大致可分为向量处理计算机、共享主存多处理机系统、分布主存多处理机系统和群机并行系统四类，下面分别作一简单介绍。

·向量处理计算机(Vector Machine)

向量机除标量寄存器和标量功能部件外，还专门设有向量寄存器和向量流水功能部件，能对向量运算进行高速处理。如图 1—3 所示。

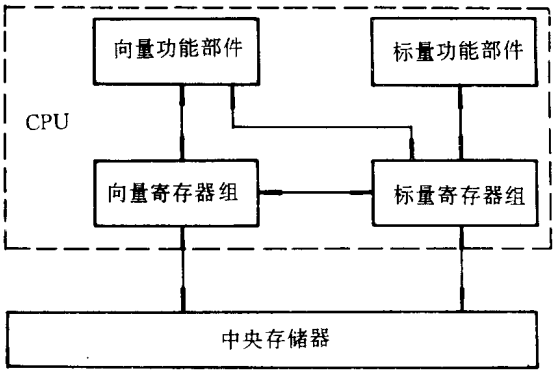

图 1—3　向量处理计算机

·共享主存多处理机系统(Shared Memory Multiprocessor System)

多个处理机共享一个中央存储器，一般还设有专门的多机同步通讯部件，能支持数据并行或控制并行的开发。处理机可以是向量机，数目常见的是 4～16，目前最多 64。处理机数目太多时，各处理机到中央存储器的通道成为瓶颈，这限制了这类机器的发展，也是分布主存大规模并行机发展起来的原因之一。如图 1—4 所示。

·分布主存大规模并行机(Distributed Memory Massively Parallel Processors)

具有很多个结点构成的并行机，每个结点有自己的处理机和存储器，结点之间以互连网络相连，主要支持数据并行开发，也能支持控制并行的开发。处理机采用微处理器，互连网络有二维网格、三维网格、树、环、超立方等。有些处理机作为控制和 I/O 结点，有完整的操作系统、网络软件和用户界面；多数处理机作为运算结点则只运行微内核操作系统。如图 1—5 所示。

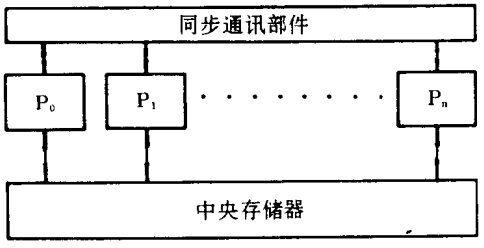

图 1—4　共享主存多处理机系统

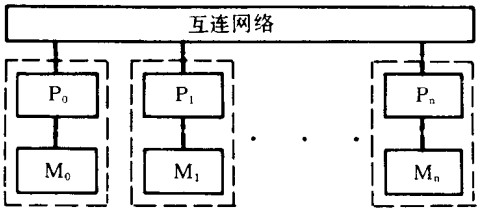

图 1—5　分布主存大规模并行机

·群机并行系统(Multicomputer Parallel System)

由多个计算机用网络或互连开关连接而成的系统，可以是同构群机，也可以是异构

的，计算机数目一般为几个至几十个，支持控制并行或数据并行的开发。结点一般是工作站，所以又称这类系统为工作站集群(workstation cluster)。每个结点上都有完整的操作系统、网络和用户界面，都可作控制结点或运算结点，即结点之间是平等的。如图1－6所示。

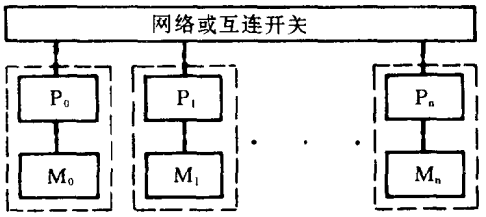

图1－6　群机并行系统

1.2.2　并行模式和依赖关系

·并行模式(Parallel Model)

先看看串行模式。串行模式即首先对第一个数据顺序地完成一组操作，再对第二个数据顺序地完成这组操作，依次直到对所有数据完成这组操作。设有一个装配线，需要完成A、B、C三种操作，每个操作需要单位时间t，则串行处理时，用一个工人，每3t时间装配完一个产品。见图1－7(a)。

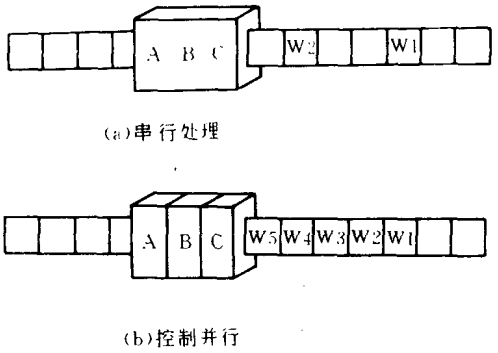

图1－7　并行模式

并行处理有控制并行和数据并行两种模式。

··控制并行　多指令流多数据流的并行执行，即对不同或相同数据执行不同操作的多个任务之间的并行。粒度较大，并行算法研究要较深入，并行流可能较少。再看上面的例子，若用三个工人分别完成A、B、C三个操作，则控制并行处理时，第一个产品要3t时间，以后每出一个产品就只要t时间。见图1－7(b)。在每一时刻装配线在对不同的产品进行不同的操作，因此称为控制并行。

··数据并行　单指令流多数据流的并行执行，即对不同数据执行相同操作的多个任务之间的并行。粒度一般较小，并行算法较直观，并行流可能较多。在上面的例子中，若用三个工人构成三条装配线，每人都完成

A、B、C 三个操作，则数据并行处理时，每 3t 时间出三个产品。在每一时刻每条装配线对不同的产品在进行相同操作，因此我们称之为数据并行。见图 1—7(c)。

对控制并行和数据并行可作如下比较：

表 1.2　控制并行和数据并行的比较

	控制并行	数据并行
控制流	多个	一个
数据流	多个、不规则	多个、规则
调度	不规则	规则
同步	显式	隐式
通讯	复杂	简单
确定性	较差	较好
可伸缩性	较差	较好
并行度	小	大
粒度	大	小
开销	小	大

多数实际问题同时具有控制并行性和数据并行性。一个问题中的某些子问题之间可能有一定的依赖关系，另一些子问题之间则没有依赖关系。考虑我们要整理一个花园，有用大剪草机剪中间的草，用小剪草机剪边沿的草，修剪花木和用喷泉浇灌整个花园四个任务，其中前三个任务是互相之间没有关系、可以用控制并行同时做的。第四个任务只能等前三个任务完成后才能做，否则就要把工人一身都淋湿了。如果我们对前三个任务每个任务派几个人去做，每个任务内则是数据并行。

· 数据依赖关系和控制依赖关系(Data Dependence and Control Dependence)

自然界的事物之间存在着一定的相互关系。在程序中不同的数据之间也可能存在一定的关系，例如：

$$A=B+C$$
$$E=F+G$$
$$D=A-E$$

则 A、E 的值没有算出来之前，无法确定 D 的值，我们称 D 依赖于 A 和 E。这是数据依赖关系。还可看出，A 和 E 的计算互相之间是没有关系的，因此可以同时计算，这就是所谓的并行性。

程序中一个语句执行与否可能取决于前一个语句的执行结果，例如：

```
S1:  IF(L .EQ. .TRUE.)THEN
S2:     CALL ABC
     ENDIF
```

这里 S2 是否执行依赖于 S1 判断的结果，这就是控制依赖关系，称 S2 控制依赖于 S1。

并行程序设计就是要在保持串行程序中原有的依赖关系不变的前提下，开发出尽可能多的并行性，使并行程序运行所需的墙上时间最短。