并行计算技术及其应用

苏德富　梁正友　华　云　编著

重庆大学出版社

图书在版编目(CIP)数据

并行计算技术及其应用/苏德富,梁正友,华云编著.
重庆:重庆大学出版社,2001.9
计算机科学与技术本科系列教材
ISBN 7-5624-2358-X

Ⅰ.并... Ⅱ.①苏...②梁...③华... Ⅲ.并行计算机—高等学校—教材 Ⅳ.TP338.6

中国版本图书馆 CIP 数据核字(2001)第 053403 号

并行计算技术及其应用

苏德富 梁正友 华 云 编著
责任编辑 曾令维

*

重庆大学出版社出版发行
新 华 书 店 经 销
重庆大学建大印刷厂印刷

*

开本:787×1092 1/16 印张:15 字数:374 千
2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷
印数:1～5000
ISBN 7-5624-2358-X/TP·314 定价:22.00 元

前言

大规模并行处理是计算机未来发展的重要方向之一。计算机并行算法的研究可以追溯到上世纪 60 年代，并于后来的 70 年代诞生了第一代并行计算机。当时的机器主要是向量处理机和流水线处理机。

1980 年以来，国际上出现了除理论科学和实验科学以外的第三类科学，即计算科学。很多学科领域，用计算代替实验，省时、省力、省钱，取得了良好的效果。

高性能计算(High performance computing)是三大科学研究的一种重要手段。由于国防、科技、经济、教育和社会不断发展的需要促使它不断地发展。影响 21 世纪人类社会的一些关键性科学问题，称之为人类面临的“巨大挑战”课题。例如，全球气候模型、基因工程、湍流分析、流体力学和超导模型等。这些问题必须依靠高性能计算机平台美国能源部 ASCI 计划的目标是研制用于虚拟核试验和核武器存储失效仿真的超级计算机系统，浮点计算能力的目标要求达到每秒百万亿次。我国对高性能计算机的需求年增长率为 20%～30%。

高性能计算技术(HPCC, High performance computing technique)已成为体现一个国家综合国力的标志，对保障国家安全、促进科技进步、推动经济发展有十分重要的作用。因此，美国、日本和西欧等国都把发展高性能计算技术作为“国家行为”。美国的 HPCC 和加速战略计算计划(ASCI, Accelerated stategic computing Initiative)是在总统直接参与下制定的。

随着通信技术的进步，将成百上千个处理机联成一体组成大规模并行计算机(MPP)，美国万亿次级的高峰值超级计算机大多采用 MPP 结构。由于网络技术的快速发展，人们发现，使用网络把 PC 机或工作站联起来组成一个机群(cluster)系统，在机群系统上作并行计算，不仅造价低廉，结构灵活，且易于实现。特别是随着 Internet 网络的普及，人们需要对大量的数据信息进行处理，用集群系统来构造高性能 Web 服务器、Email 服务器等。

基于网络的应用模式已从客户/服务器方式、浏览器/服务器方式向多层分布式并行处理结构发展。利用 CORBA 技术实现多层分布式

并行处理是其中的一种。元计算技术是当前高性能计算研究的前沿课题之一，所谓元计算是通过广域网把不同性质的一组计算机资源集成起来，作为一个单独的计算环境向用户提供服务，基于 CORBA、JAVA 或 Web 等技术的分布式对象技术适合于构造复杂对象。利用计算机系统模拟现实世界，利用数字世界和现实世界进行商品展示、模拟军事实战演习等使得先进分布仿真（ADS，Advance Distributive simulation）技术成为研究的热点。

本书内容将包含并行算法、并行计算机体系结构、编程和应用，主要介绍了并行计算基本理论和实现技术，以及网络环境下的并行分布式计算。全书共 17 章，可分为两个部分：第一部分（第 1～8 章）阐述并行计算基础知识和并行程序设计方法，其中第 1～3 章介绍了并行计算机体系结构的分类、常用的并行计算模型、并行程序基本特性和程序设计的基本方法，第 4～8 章分别讨论了并行排序与选择、数据传输与选路、伪随机序列的并行算法、并行串匹配和数值并行算法的基本思想和实现；第二部分（第 9～17 章）主要论述基于 CORBA 技术的网络环境下的并行分布式计算环境及其应用，其中第 9 章介绍了分布式技术的概念、发展以及应用和研究情况，第 10 章简要地介绍了新一代的分布式计算环境 CORBA 规范的基本概念，第 11～14 章介绍了 OMG IDL 语言以及 OMG IDL 语言到 C＋＋的映射并讨论了 CORBA 客户程序以及服务程序的编写方法，第 15 章讨论了 OMG 的命名服务，第 16 章介绍了 CORBA 产品 OmniORB2 及其构建的分布式计算环境的方法，第 17 章讨论了基于 CORBA 分布对象的并行计算技术以及在此基础上进行网络并行搜索的设计研究。

本书取材新颖，内容丰富，结构合理，注重基础，面向应用　可作为高等院校本科计算机科学与技术、电子信息工程、系统工程等专业的本科和研究生的教材或教学参考书，也可供其他专业的师生以及科研和工程技术人员自学或参考。

由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

对本文所用的 CORBA 产品 OmniORB2 以及所开发的源程序感兴趣的读者，可与出版社或作者联系。

作　者

2001 年 5 月

目录

第1章 引 论

随着信息时代的到来，需要处理的信息量越来越庞大、需要解决的问题越来越复杂，使得计算量剧增。通过提高单个处理器的运算速度和采用传统的“顺序(串行)”计算技术已难以胜任。因此，需要有功能更强大的新的计算机系统和计算技术来支撑。并行计算机及并行计算技术应运而生。

本章讨论并行处理技术基础，包括并行计算机结构、并行计算模型、并行计算的若干理论问题以及并行算法设计和分析的基本概念。

1.1 并行处理技术及其应用

计算机处理从数据和信息处理到知识处理，最终发展到今天的智能处理，每前进一步，都要求增强计算机的处理能力。计算机发展史表明，为了达到高性能计算的目的，除了必须提高计算机系统的 CPU 等元器件的速度外，其体系结构也必须不断改进，特别是当元器件的速度达到极限时，设计新的计算机系统结构成为问题的焦点。

另一方面，随着计算机性能的飞速发展与应用的日益广泛、深入，许多有识之士越来越清楚地认识到“计算”已经成为与理论分析和实验并列的第三种科学研究手段。为了保持在高性能计算和计算机通信技术领域中的世界领先地位，采用高性能计算机与网络技术(它们本身隐含着并行处理技术)来刺激设计与生产，以增强国民经济、国家安全、教育及整体环境的竞争能力，美国政府于 1991 年度开始执行“高性能计算与通信”(High Performance Computing and Communication，HPCC)项目即美国总统科学战略项目，目的是通过加强研究与开发解决一批重要的科学与技术挑战问题。该项目包括四部分：高性能计算机系统(HPCS)、先进软件技术与算法(ASTA)、国家科研与教育网(NREN)、基本研究与人类资源(BRHR)。HPCC 的具体目标是：对一批重要应用课题，计算能力要达到每秒万亿次运算(10^{12})即 Teraops；发展有关的系统软件工具，对处理一大批问题的算法进行改进；国家研究网能力要达到每秒十亿位(10^{9})即 Gigabit；充分保证计算机科学与工程领域科技人员的需求。

而要充分发挥新的计算机系统的高性能，必须提出新的计算机体系结构，发明新的计算方法，设计快速的算法，从而开发出先进、高效的系统软件和应用软件。并行计算机及并行处理技术正是实现这些目标的基础。

所谓并行处理(Parallel Processing)技术是指在同一时间间隔内增加操作数量的技术，所谓并行计算机(Parallel Computer)则是为并行处理所设计的计算机系统，相应地在并行计算机上求解问题称为并行计算(Parallel Computing)，在并行计算机上求解问题的算法称为并行

算法(Parallel Algorithms)。

并行处理是一种强调开发计算过程中并发事件(Concurrent Event)的有效的信息处理方式。并发性(Concurrency)包含并行性(Parallelism)、同时性(Simultaneity)和流水线(Pipelining)。并行事件可在同一时间间隔内发生在多个资源中,同时事件可在同一时刻发生,流水线事件可在部分重叠的时间内出现。

并行处理的四个级别是:作业或程序之间的并行,是指多个作业或多道程序的并行执行;任务或进程之间的并行,是指多个任务或程序段的并行执行;指令之间的并行,是指多条指令的并行执行;指令内部的并行,是指一条指令内部各微操作之间的并行执行。

开发问题的并行性包括开发计算并行性、搜索并行性和逻辑并行性。显然,计算并行性和算法的设计与分析有着密切关系。

并行处理和并行计算技术是一个年轻的领域。并行处理技术的发展最初主要是为了满足某些需要大量"计算"的领域,例如数值天气预报、石油勘探、空气动力学计算、天体物理的计算模拟、卫星图像处理、核武器的仿真与实验、社会发展和宏观经济模型的构造、人工智能研究等等。但是,今天的并行计算远非仅仅是常规的"数值并行计算",它包含更为广泛的意义:数值和非数值并行计算。事实上,现实世界中的一切活动都是并发的,如制造型企业里产品按流水线并行方式组装,由不同的设计小组同时设计同一个产品的各个部件,因此,将对现实世界问题的求解映射成计算机求解问题时,求解的方式应当也是并行的。

1.2 并行计算机分类

将并行计算机分类是为了方便讨论并行算法的设计与分析。虽然并行算法的设计与分析在一定程度上依赖于并行计算机体系结构,但是人们一开始总是希望设计出独立于某个具体机器结构的"通用"的并行算法,即希望所设计的算法至少能在一类并行计算机上运行。反过来,通过并行算法的设计与分析也有助于指导设计新的体系结构的并行计算机。

1.2.1 Flynn 分类法

966 年美国的 M. J. Flynn 教授提出一种按指令流和数据流将计算机系统进行分类的方法,将计算机系统分成 4 类:

①单指令流单数据流计算机:SISD(Single Instruction Stream Single Data Stream)计算机,即传统的只有一个处理机的顺序(串行)计算机。

②单指令流多数据流计算机:SIMD(Single Instruction Stream Multiple Data Stream)计算机,如阵列处理机、流水线处理机、关联处理机等。

③多指令流单数据流计算机:MISD(Multiple Instruction Stream Single Data Stream)计算机,此类计算机目前实际应用不多。

④多指令流多数据流计算机:MIMD(Multiple Instruction Stream Multiple Data Stream)计算机,如多处理机和多计算机均属此类计算机。这类计算机是目前并行计算机发展的主要方向。

1.2.2 Handler 分类法

1977 年，德国的 Handler 教授根据计算机系统中流水线和并行度出现的级别，将一台计算机表示为三对整数（三元组）。令 PCU 代表处理器控制单元，对应于一个处理器或 CPU；ALU 代表算术逻辑单元，对应于功能单元或处理单元；BLC 代表位一级电路。据此可将一台计算机表示成：

$$T(C)=<K\times K',D\times D',W\times W'>$$

其中：K 为 PCU 的数目，K' 为能够流水线执行的 PCU 的数目；

D 为每个 PCU 所控制的 ALU 数目，D' 为能够流水线执行的 ALU 数目；

W 为 ALU 或处理单元 PE 中的位数，W' 为在所有 ALU 或单一个 PE 中流水线的段数。

从研究并行算法的角度出发，人们一般采用 Flynn 分类法。

1.2.3 按机器体系结构分类

①同步并行计算机：它通过全局时钟、中央控制单元（或向量单元控制器）实现并行操作。其中，阵列处理机（Array Processor）每个 PE（处理器）均可和其上、下、左、右 4 个近邻相连，所有的 PE 在同一控制器控制之下按同一指令的要求，对同一块内存中的不同数据同步地进行操作，从而达到操作级并行。显然，阵列处理机实现空间上的并行。这样的机器对有限差分、矩阵运算和快速傅里叶变换（FFT）等计算具有很高的效率，在诸如图像信息处理等领域中得到重要应用。

此外，关联处理机（Associative Processor）则是一种特殊的阵列机，它按存储内容寻址，对于大规模信息检索之类的处理尤为合适。流水线处理机（Pipeline Processor）将工厂里的生产流水线装配技术应用于计算机结构中，把计算机的运算部件或控制部件等装配成一些有序的子部件，利用功能部件分离与时间重叠的办法，使每个被操作的对象处在整个操作流程的不同功能部件中，且保持在不同的阶段完成，从而达到操作级的并行。这种结构的计算机实现时间上的并行。

②MIMD 并行计算机：分为共享存储多处理机系统（MIMD-SM）、分布存储的多计算机系统（MIMD-DM）和分布虚拟共享存储多计算机系统（MIMD-DVSM）。多处理机和多计算机由一些可编程的且均可各自执行自己程序的多台处理器组成，其中：

MIMD-SM 多处理机以各处理器共享公共存储器进行数据通信为特征，它又称为紧耦合多处理机系统（Tightly Coupled Multiprocessor）。这类并行计算机编程较容易，但机器结构实现较为困难，且处理机数增长受限制，目前商用系统一般不超过 256 个处理机。

MIMD-DM 多计算机则以各处理器经物理通信链路传递消息进行数据通信为特征，它又称为松散耦合多处理机系统（Loosely Coupled Multiprocessor）。这类并行计算机编程较困难、速度较慢，但实现系统较容易且可扩展性较好。

MIMD-DVSM 多计算机物理上通过通信链路将各计算机连接起来，并将物理上分布的存储器构成一个统一的逻辑存储空间，各计算机通过共享这一逻辑存储空间进行数据通信为特征。这类并行计算机既吸取了 MIMD-SM 并行计算机的优点又吸取 MIMD-DM 并行计算机的优点，并抛弃它们的缺点，是目前最有前途的一种并行计算机。此外，还有一种 MIMD 变形

结构的并行计算机，它属于非冯·诺依曼计算机结构，例如数据流计算机(Data Flow Machine)，归约计算机(Reduction Machine)，推理计算机(Inference Machine)等。

③机群并行计算系统(Cluster)：这是一种基于网络的计算机机群并行计算系统。它将若干独立的计算机系统通过高速通信网络互联起来以支持并行计算。机群并行计算系统中的每个计算机称为结点机(可以是巨型机、大型机、中型机、小型机、工作站和高档微机等)。如果机群并行计算系统中所有的结点机都是工作站，那么称这样的机群并行计算系统为工作站机群。

机群并行计算系统具有如下特点：

①易于实现。只需将现有的计算机通过高速通信网络互联起来即可实现。

②可伸缩性强。在现有网络上增加新的计算机即可提高机群并行系统的处理能力。

③平台无关性。可以将各种不同体系结构的计算机互联起来构成一个异构并行计算环境。

④可重用性。因为通过机群并行计算系统中的结点机均是通用计算机，所以可以充分利用原有的成熟的程序代码进行并行程序设计。而且，可以借助分布式共享存储 DSM 技术，使得应用程序可以透明地访问所有结点机的内存，大大地方便用户的并行程序设计

⑤输入输出高度并行。可以利用数据分布技术，充分发挥机群系统中各输入输出的并行操作性能。例如，实现数据库系统的并行操作。

⑥性能/价格比高。机群并行计算系统中的结点机可以是价格便宜的微机或工作站，但是整个机群系统所能达到的性能却可以与大型机和巨型机媲美。

机群并行计算系统的不足之处是，连接入系统中的结点机达到一定规模时，系统可能需要花费大量的时间进行通信，从而降低了并行效率。换句话说，机群计算系统难以适应需要"大规模"并行处理的应用领域。

机群并行计算系统采用异步传输 ATM 模式以解决通信瓶颈问题，系统软件可以采用以 UNIX 为基础的 PVM(并行虚拟机器)环境或者 Linux 并行环境等。

1.3 并行计算机的处理器互联方式

如何将处理器与处理器或者处理器与存储器互联起来是并行计算机关键技术问题之一。这种互联结构除了对系统的效率有影响外，也对并行算法的设计与分析有影响。本节将讨论几种重要的并行计算机结构的互联方式。.

1.3.1 一维线性阵列结构

一维线性阵列(Linear Array)简记为 LA，其连接方式是系统中的每个处理器(首、尾处理器除外)只与其左、右近邻的处理器相连，这种连接方式是 Systolic(心动阵列)结构的最基本形式。

设处理器的数目 $n=2^m$，分别记为 $P_0, P_1, P_2, \cdots, p_{n-1}$；其地址(编号)的二进制表示式为：$P=p_{m-1}p_{m-2}\cdots p_0$，则线性阵列的连接函数 LC 可定义如下：

$$LC_{-1}(P)=P-1\ ,1\leqslant P\leqslant n-1\text{(逆向连接)}$$

$$LC_{+1}(P)=P+1,0\leqslant P\leqslant n-2\text{(正向连接)}$$

例如，8个处理器的线性阵列结构如图1.1所示。

一维线性阵列连接的变种是循环移位连接方式：

P_0----P_1 --P_2 P_3 - P_4----P_5----P_6----P_7

图1.1 8个处理器的线性阵列

1.3.2 二维网格结构

在网格连接(Mesh-Connected)方式中，每个处理器都排在二维网格中的一个格点上。二维网格连接结构简记为 MC^2，其连接函数定义如下：

$MC^2_{-1}(P)=P-1 \bmod n$

$MC^2_{+1}(P)=P+1 \bmod n$

$MC^2_{-\sqrt{n}}(P)=P-\sqrt{n} \bmod n$

$MC^2_{+\sqrt{n}}(P)=P+\sqrt{n} \bmod n$

其中 $0 \leqslant P \leqslant n-1$。

例如，当处理器数 $n=16$ 时，其二维网格连接函数表示法为：

$MC^2_{-1}(P)=(15,14,13,\ldots,3,2,1,0)$

$MC^2_{+1}(P)=(0,1,2,3,\ldots,13,14,15)$

$MC^2_{-4}(P)=(15,11,7,3)\ (14,10,6,2)\ (13,9,5,1)\ (12,8,4,0)$

$MC^2_{+4}(P)=(0,4,8,12)\ (1,5,9,13)\ (2,6,10,14)\ (3,7,11,15)$

其相应的二维网格结构如图1.2所示。

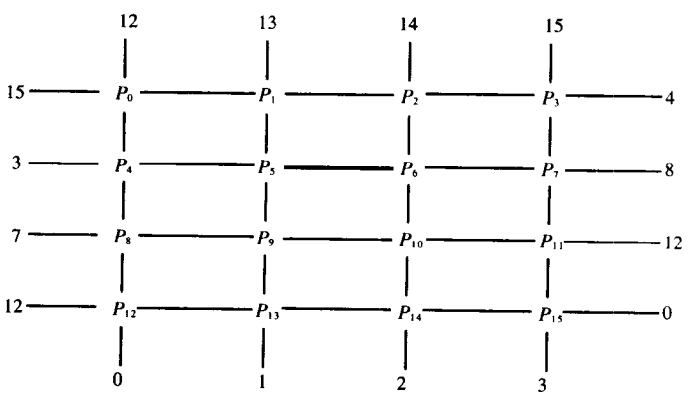

图1.2 $n=16$ 个处理器的二维网格结构

二维网格结构是一种常用的并行机，特别适合于处理二维问题，已有许多有效的排序、选择、矩阵运算以及图像处理等方面的并行算法在其上运行。网格结构的缺点是通信能力有限，这是因为在最坏情形下，网格机器上任意两个处理器之间的信息交换步数(选路数)需要 $\sqrt{n}-1$ 步。

1.3.3 树结构

我们都知道，二叉树是一种具有很多优良性质的、得到广泛应用的数据结构。将处理器按树结构方式进行连接(Tree-Connected)称为树机器，简记为 TC。树机器中除了根结点和叶结

点的处理器之外，其他每个结点的处理器都与其父结点的处理器及其两个子结点的处理器相连，即具有 n 个处理器的树机器的连接函数为：

$$\mathrm{TREE}(P_i)=\begin{cases}P_{2I+1}\\P_{2I+2}\end{cases}$$

其中 $i=0,1,2,\ldots,\lfloor\frac{n}{2}\rfloor-1$。树中只有根结点和叶结点的处理器具有输入输出处理能力。树机器的典型应用是叶结点的处理器对数据执行并行运算，而内结点的处理器则负责为叶结点处理器提供通信，其最长通信路径约为树的高度，因此树根结点的处理器很容易成为系统通信的瓶颈。图 1.3 示出了 15 个处理器构成的二叉树机器结构。

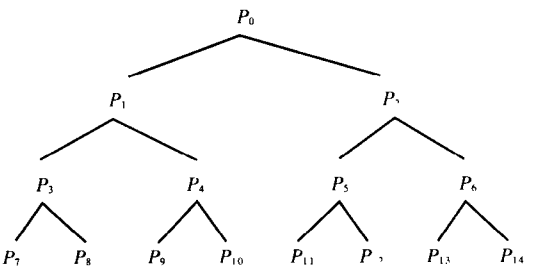

图 1.3　15 个处理器的二叉树机器

为了减少处理器之间的通信瓶颈问题，可对上述的二叉树机器结构进行改进。一种方式是采用如图 1.4 所示的 X 树机结构，它实际上是一棵将同层中兄弟结点的处理器彼此进行相连的完全二叉树。

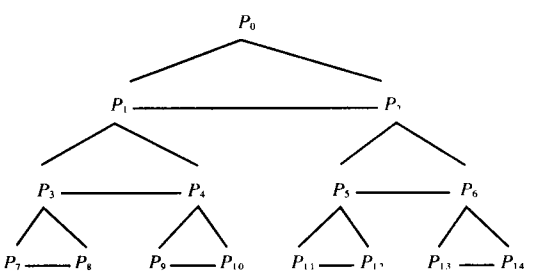

图 1.4　15 个处理器的 X 树机器

由于平方根函数大于对数函数，因此二叉树结构机器的通信能力要优于网格结构机器的通信能力。若将它们的优点结合起来，则可以导出一种称为树网结构的并行计算机

1.3.4　树网结构

树网机器(Mesh-Tree)是将二叉树机器和网格机器相结合起来而产生的一种并行计算机结构，简记为 MT，树网机器也称为正交树机器(Orthogonal Tree)。对于网格结构，如果去掉其水平和垂直连接，并将其各行和各列代之以二叉树，那么可得到树网结构。显然，这时树网结构的行和列中格点上的处理器变成完全二叉树相应的叶结点处理器，它们通过相应的行树和列树进行通信，数据的输入和输出操作由树根结点处理器完成。树网结构既具有良好的通信功能又继承了网格结构的许多有效算法，同时也非常适合于 VLSI 的集成化。

1.3.5　超立方连接结构

超立方连接机器是一个具有 $n=2^k$ 个结点的网络，这些结点组织成一个 k 维的超立方结构(Hypercube)。特别地，当 $k=3$ 时即是人们熟知的立方连接机器(Cube Connected，简记为 CC)。在立方连接机器中，每个处理器都分布在 3 维立方体的各个顶点中。如果采用合适的标记方法，那么每个处理器都可以直接与它的三个近邻相连，而且每维中各处理器的地址的二进制表示只有一个二进制位不同。值得注意的是：处理器的数目 $n=2^m$，其地址 P 的二进制表示为 $P=p_{m-1}\cdots p_1 p_0$，记 p_i 为 p_i 的补，定义立方连接函数 CC 为：

$$CC_i(P)=CC_i(p_{m-1}\cdots\cdots p_{i+1}\ p_i\ p_{i-1}\cdots\cdots p_0)=p_{m-1}\cdots\cdots p_{i+1}\ \overline{p}_i\ p_{i-1}\cdots\cdots p_0$$

其中 $0\leqslant i<m$。

处理器数 $n=8$ 的立方连接结构如图 1.5 所示，其中 CC_0 将地址第 0 位不同的那些处理器相连，CC_1 将地址第 1 位不同的那些处理器相连，CC_2 将地址第 2 位不同的那些处理器相连实际上，将 3 个连接函数组合在一起即得到一个立方体。

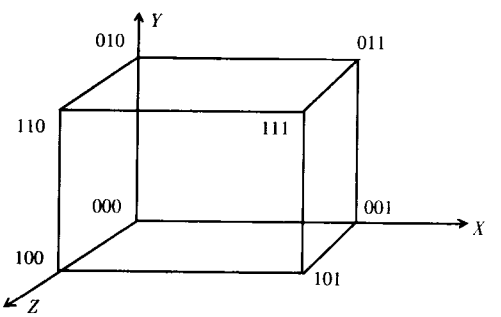

图 1.5　8 个处理器的立方连接机器结构

立方连接网络是一种非常通用的互联结构，它可以模拟很多其他形式的互联网络。但是，由于 k 维超立方连接网络中的每个顶点都与 k 个近邻相连，所以当问题规模变大时将导致顶点之间的连线非常复杂，这无疑限制了它的应用范围。

1.3.6　q 维网格机器

q 维网格(q-Dimensional Lattice)机器将其每个处理器均置于 q 维空间的 $(i_1, i_2, \cdots, i_q)$ 格点上(i_k 是整数，$1\leqslant k\leqslant q$)，每一格点只放置一个处理器。位于 $(i_1, i_2, \cdots, i_q)$ 上的处理器 P_i 连接位于 $(j_1, j_2, \cdots, j_q)$ 上的 P_j，当且仅当 P_i 和 P_j 之间按如下定义的距离等于 1：

$$d(P_i, P_j)=\sum_{k=1}^{q}\mathrm{abs}(i_k-j_k)$$

其中函数 abs()为绝对值函数。q 维网格机器中按此方式相连的两个处理器称为是相邻的。

由此可见，一维线性连接实际上就是一个 1 维网格机器，二维网孔连接即是一个 2 维网格机器，q 维立方连接则是一个具有 2^q 个处理器的特殊的 q 维网格机器。q 维网格的一个重要应用是用于证明并行排序时间复杂性的下界。

1.3.7　洗牌交换网络

洗牌-交换(Shuffle-Exchange，简记为 SE)网络是一类非常有用的互联结构。洗牌的名字

来源于日常生活中的“洗牌”游戏，即将一叠牌对分成两半，然后分别对这两部分牌再进行同样方式的洗牌，依次类推。这种洗牌方法称为“均匀洗牌”(Perfect Shuffle)。在实际应用中，一般将它与“交换”连接联合使用，构成所谓的“洗牌-交换”网络。

洗牌和交换连接函数分别定义如下：

$$SH(P)=SH(p_{m-1}\ p_{m-2}\cdots p_1\ p_0)=p_{m-2}\ p_{m-3}\cdots\ p_1\ p_0\ p_{m-1}$$

$$EX(P)=EX(\ p_{m-1}\ p_{m-2}\cdots p_1\ p_0)=\ p_{m-1}p_{m-2}\cdots p_1\overline{p_0}$$

由上述定义可知，洗牌的功能实际上是将处理器地址循环左移一位，交换功能则是将奇、偶相邻的两个处理器地址中的数据进行交换。

处理器数 $n=8$ 的洗牌-交换网络如图 1.6 所示，其中虚线表示洗牌，实线表示交换。

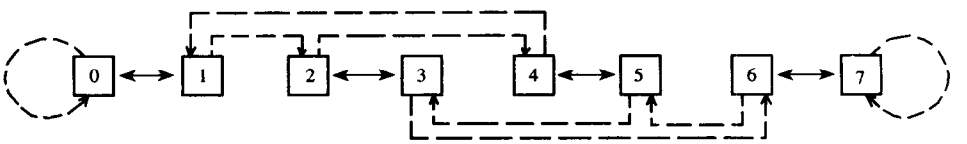

图 1.6　8 个处理器的洗牌 - 交换网络

由于洗牌-交换网络可以方便地模拟立方网络的功能，而且其每个顶点的度数为 3，并与问题的规模无关，因此是一种应用较广泛的并行结构。其不足之处是网络结构缺乏规律性，因此难以模块化。

1.3.8　蝶形结构

蝶形(Butterfly)结构从拓扑结构上看像蝶形，它与多级立方网络、二叉树网络和归并网络相似，它共有 $(k+1)2^k$ 个结点、$k+1$ 行(第 0 行到第 k 行)，其中第 0 行和第 k 行逻辑上视为同一行，每行中有 $n=2^k$ 个结点。$k=3, n=8$ 的蝶形结构如图 1.7 所示。

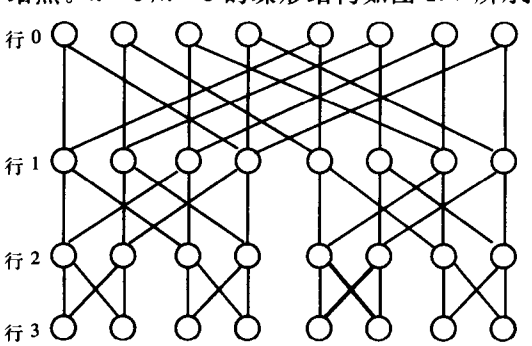

图 1.7　$k=3, n=8$ 的蝶形结构

设 $P_{r,i}(0\leqslant i<n, 0\leqslant r\leqslant k)$ 为位于第 r 行上的第 i 个处理器，则在第 $r(r>0)$ 行上的处理器 $P_{r,i}$ 连接 $P_{r-1,j}$，这里的 j 或者等于 i 或者 j 与 i 的二进制表示从左边数起只有第 r 位不同。这种网络由很多蝶状的图形组成，蝴蝶的翅膀宽度由下而上呈指数增大。

蝶形结构的并行计算机非常适合于处理快速傅里叶变换(FFT)等一类问题的算法。

1.4 并行计算模型

虽然并行计算机体系结构是并行算法运行的物质基础，但是并行算法的设计与分析不能局限于某种具体结构的并行计算机。因此，必须对各种具体的并行计算机抽象化，抽象出具有一般意义的并行计算模型，然后在此模型上研究并行算法。因此，需要对并行计算机系统作一些假设：

①处理器数目可以是无限的和有限的。所谓无限是指在任何时间内系统可使用的处理器数目随问题的规模 n 增长而增长，所谓有限是指用固定的 $p(p<n)$ 台处理器去求解规模为 n 的问题。

②每台处理器执行操作的时间均相同并且和处理器的数目无关。

③共享存储器的容量无限，而且允许或者不允许多个处理器同时读/写同一个共享存储单元。

④计算机指令系统仅具有最基本的算术和逻辑运算、访存、输入/输出和转移等指令（因为功能齐全的指令系统也不会改变算法复杂性的数量级）。

1.4.1 SIMD 互联网络模型

不同互联网络连接的 SIMD 机器简记之为 SIMD-IN，它分为处理器-处理器的组织方式和处理器-存储器的组织方式，前者适用于并行计算机处理器和存储器数目相等的场合，后者则适用于存储器数目多于处理器数目的场合。从设计并行算法的角度来看，两者没有本质的差别。

SIMD 互联网络模型有：

①一维线性连接的 SIMD 模型（SIMD-LC）
②网孔连接的 SIMD 模型（SIMD-MC）
③树形连接的 SIMD 模型（SIMD-TC）
④树网连接的 SIMD 模型（SIMD-MT）
⑤金字塔连接的 SIMD 模型（SIMD-PC）
⑥超立方连接的 SIMD 模型（SIMD-HC）
⑦网格连接的 SIMD 模型（SIMD-LA）
⑧立方环连接的 SIMD 模型（SIMD-CCC）
⑨洗牌交换连接的 SIMD 模型（SIMD-SE）
⑩蝶形结构连接的 SIMD 模型（SIMD-BF）

1.4.2 共享存储的 SIMD 模型

在共享存储器（Shared Memory）的 SIMD 模型（简记为 SIMD-SM）中，处理器之间通过采取读/写一个共享存储器的办法来进行数据通信。当共享存储器的容量有限时，在同一时刻，多个处理器访问共享存储器会受到限制，按照受限制的程度可将 SIMD-SM 模型分为：

①不允许同时读写同一存储单元的 SIMD 机器模型（Exclusive Read and Exclusive