

现代计算机 技术博览

A Broad Review of Modern Computer Technology

● 计算机科学技术系列丛书

● 第 1 集

● 中国计算机报社组编

内容提要

现代计算机技术的发展日新月异,它代表一种新的生产力,是世界新技术革命浪潮中的先导技术。本书由《中国计算机报》社聘请国内著名计算机专家撰写,涉及计算机领域最新技术的水平、现状和发展趋势,约有近 30 个专题,其中包括:图形处理、并行处理、网络技术、数据库标准与应用、光盘技术、计算机与信息安全、病毒的危害及防范、汉语语音识别、窗口技术等广大科技人员关心的热点技术。

读者对象:科技工作者、管理人员、大专院校师生。

现代计算机技术博览

中国计算机报社组编

责任编辑:路 石

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

铁指印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:40 字数:900 千字

1991 年 8 月第一版 1991 年 8 月第一次印刷

印数:1—8000 册 定价:平装 25 元 精装 30 元

书号:ISBN 7-5053-1415-7/TP·241

前 言

计算机科学技术正在以难以置信的方式改变着社会面貌,同时也在以惊人的速度改变着自身面貌。新技术、新概念、新门类、新系统,犹如汲不尽的泉水,源源不断地涌现、再涌现,就连最乐观的预言也在这种空前的发展速度面前黯然失色。

当人们还在对 386 超级微机的功能赞叹不已时,功能更强的 486 微机已突然出现在人们面前;而 3 英寸磁盘机才进入微机几天,2.5 英寸磁盘机又已悄悄登场。至于象工作站、小巨机、局域网与宽域网、并行处理、知识库、多媒质、病毒防治、图形图象、面向对象处理、人工智能等领域,则无日不在产生新的发展。新现象和新信息是如此之多,出现是如此之快,就连最博学多才的计算机专家也不敢自夸“无所不知、无所不晓”了。

因此,对计算机科学技术的发展提供最新的、全面的概括,帮助科技人员掌握最新技术和紧跟世界潮流,已成为广大计算机专业人员和计算机用户的迫切需要。本书的目的就是试图以专题选编的形式来满足这种需求。

本书共收入近 30 个技术专题 120 多篇专文,基本上概括了最近以来计算机科学技术领域中重大的新发展。这些专题和专文大多是国内从事计算机科研工作的专家撰写的,部分是根据国外资料编写的,在 1989~1990 年期间曾分别刊载于《中国计算机报》,在读者中产生了良好的反响。

中国计算机报是面向计算机科技人员和广大计算机用户的大篇幅报,每周出版一期,每期除刊出大量新闻和专文外,还刊出一个 2 万~4 万字的技术专题。这些专题均将以“博览”的形式陆续出书。但由于出版的滞后性,欲先睹为快的读者,不妨到邮局订一份《中国计算机报》。这样,您将每周都能看到一个对一门新技术发展的最新概括。

现代计算机技术博览编辑出版委员会

1991 年 7 月 10 日

《现代计算机技术博览》 编辑出版委员会

主 编:周慕昌

副主编:李超云

编辑委员:李 弘

王锡生

翁经午

孙敬芳

管 锐

国晓平

张卓明

梁祥丰

王春贵

董瑞玲

刘丽英

张碧薇

尹 荣

史新元

张鹏飞

张延淑

李云鹤

陈 艳

杨 红

曲 光

李 颖

张殿阁

陆孝如

李铭山

遇华宁

王 路

张慧英

孙毓林

目 录

I. 并行处理技术

并行处理技术综述.....	1
并行处理——先进技术.....	9
大型并行处理系统——巨型机技术	18
数组处理机的发展	24
超立方体并行多处理机系统	28
Alliant 并行多处理机系统	32

II. 网络技术

计算机网络的形成与发展	37
PC 机与主机的连接和通信.....	41
无线数字通信技术及其应用	50
Novell 与 3COM:微机 LAN 的现状、技术策略和选择	56
开放系统互连标准	62
Novell 网的发展历史和它的主要特点	66
Netware 的系统结构和基本工作原理.....	70
Novell 网的系统安全与数据保护	74
Novell 网文件服务器特点及性能分析	78
Novell 网应用前景展望	82
新一代窗口系统——Microsoft Windows 3.0	89

III. 计算机图形图象处理

图形技术的最新进展——三维图形处理器.....	100
视频信号处理系统结构的概况.....	103
计算机的智能接口.....	108
IBM PC 系列机显示系统的发展及性能比较	115
数字视频特技.....	123
数字特技基本原理及发展近况.....	134
录象采集处理分析系统.....	141
数字电视领域中的计算机技术.....	147
视频绘图系统与数字视频特技系统.....	151

IV. 汉字信息处理

字形信息技术开发的现状及展望.....	161
汉字点阵字形系列的研究.....	164
叠加法字形压缩技术.....	166
高精度汉字点阵的生成方法.....	169
汉字字形信息压缩方法探讨.....	171
汉字字形压缩技术的概况.....	175
电子印刷排版系统中的汉字字形问题.....	178
矢量字形技术标准化探讨.....	180
汉字字形缩小放大技术.....	182
电脑的听觉智能语音识别概述.....	186
连续语音识别 CSR	190
语音识别系统的小型化.....	194
应用神经网络方法进行语音识别.....	196
语音识别系统中的硬件部分.....	199
发音人识别.....	203
中文信息输入技术研究中的一些情况.....	206
“部首切去法”的一个外衍.....	210
汉字键盘输入技术评测方法的新发展.....	213
论计算机汉字输入技术的评估.....	215
全汉字编码输入与字形输出技术.....	220
论汉语键盘输入评测和竞赛的科学性.....	228
略论汉字键盘输入的评测与竞赛.....	231
电子印刷排版系统的发展与展望.....	232
电子排版系统的输出设备——激光印字机.....	239
What you see is what you get	243
台式印刷排版系统的扫描器.....	248
电子印刷排版系统的采购原则.....	251

V. 计算机安全与标准化

计算机病毒的危害及其防范.....	256
计算机病毒概述.....	265
计算机病毒和蠕虫对大学计算机系统的攻击.....	271
Mil 网/Arpa 网遭受攻击的有关细节	274
计算机病毒及防御措施.....	277

国外防范计算机病毒的疫苗产品.....	285
计算机病毒——理论与实践.....	288
计算机系统的脆弱性与信息安全.....	304
一种安全微机操作系统的实现方法.....	312
数据安全.....	316
网络安全与存取控制.....	320
计算机安全与立法概述.....	325
我国计算机设备产品安全标准的贯彻和实施.....	341
数据处理设备安全标准要点.....	353
微机产品电气安全部分的验证检查.....	359
安全认证概要.....	368
避免静电的设计——数字设备的 ESD 测试	370
信息泄露发射问题概述.....	378
EMI 屏蔽结构工程设计中的密封垫.....	382
计算机设备的 EMC 标准	386
EMI 屏蔽结构的工程设计.....	394
电源系统的电磁兼容性设计.....	398
计算机研制生产过程中所需的 EMI 测试工装	403
数据库标准化技术概述.....	405
信息处理系统数据管理参考模型.....	413
数据词典和信息资源管理.....	418

VI. CAD 与 DSS

CAD 的进展概况	423
计算机辅助工艺过程设计(CAPP)及在计算机集成制造系统(CIMS) 中的应用.....	429
CAD/CAM 集成系统中的数据库管理系统	432
计算机辅助工业设计(CAID)的时代即将到来	436
计算机辅助设计——CAD	440
几何选型原理方法和当前的新课题.....	444
DSS 的由来组成和发展.....	450
智能 DSS 的功能与结构	455
DSS 的分析设计.....	459
群体决策支持系统.....	462
面向财务的决策支持系统.....	465
开发 DSS 的语言工具选择	469
企业优化决策支持系统.....	471

VII. 计算机外存储技术

世界计算机外存贮设备市场述评.....	474
软盘驱动器现状与发展趋势.....	481
微小型温盘驱动器的技术现状及趋势.....	485
螺旋扫描技术的兴起及未来.....	488
国外光存贮技术的应用及发展趋势.....	492
光盘存储技术.....	498
光盘机开发现状.....	501
光盘在计算机领域中的应用.....	507
光盘存储技术的发展前景.....	512

VIII. 其 他

超高速 VLSI 技术的进展	515
ASIC 中的门阵列方法	519
ASIC 技术发展的新动向	525
ASIC 设计中的“标准单元法”	527
模拟专用集成电路及其设计自动化技术.....	532
ASIC 测试	536
综合业务数据网络由来和发展.....	540
综合业务数据网络技术发展概况.....	543
宽带 ISDN 现状	550
法国的综合业务数字网.....	554
日本未来的电信业务与网络发展.....	559
ISDN 用的中大规模集成电路的发展现状	563
未来的计算机展望.....	572
新元件的研究和未来计算机.....	576
整体性计算机的设想.....	578
超并行计算机.....	580
人工智能计算机.....	584
光计算机的前景.....	587
数据流计算机.....	589
计算机的进化概念.....	592
连接计算机.....	593
生物计算机概念.....	595

并行处理技术综述

华北计算技术研究所 王 军

近十年来,并行处理技术的应用迅猛发展,这是因为并行处理固有的优越性。在新的微电子技术、并行处理技术的支持下形成的产品独树一帜,快步进入计算机大家庭。并行处理技术的最新发展包括新的并行硬件体系结构及新的不断涌现的产品。诸如小巨型机、脉动式计算机,互连技术拓扑结构(多级开关拓扑结构、超立方体结构等)的日趋完善以及程序设计环境和并行处理算法的突破(应用性程序设计)。本文就目前并行处理技术的发展做一概念性介绍,主要重点是针对并行处理体系结构而不是介绍某个特定的机器。

表 1 粒度的划分

粒 度	运行对象
大粒度	作 业
中粒度	任 务
中小粒度	子程序
小粒度	循 环
细粒度	指 令

一、并行处理的基本概念

由于并行体系结构的种类繁多,要给出并行结构的精确定义和分类标准是比较困难的。目前,按并行处理的粒度(细粒度并行,中、大粒度并行)划分并行处理系统的方法较为流行,表 1 就是一种粒度的划分法。

所谓粒度是指能够并行处理的单位的大小。但考虑到一些现代并行体系结构(如数据流机、归约机的结构)不能简单按粒度方法进行划分,而且诸多应用题例的不同阶段的粒度单位不定,因此,本文在定义和分类并行体系结构的过程中,着重考虑下列三个方面:

(1)包含有低级并行机制的结构(指令流水线、多个 CPU 功能部件、独立的 CPU 和 I/O 处理器),以便与最新发展的并行体系构成连续的整体;

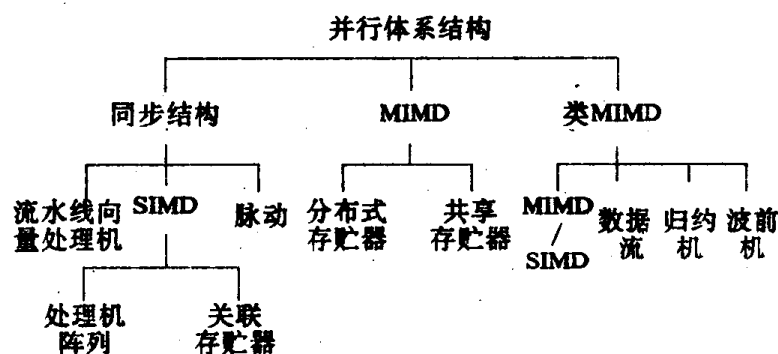


图1 并行计算机体系结构的分类

(2)仍尽量采用费林(Flynn)根据指令流和数据流的分类方法(SISD、SIMD、MISD、MIMD);

(3)包括流水线向量处理机和其它并行结构,这些结构难以按费林分类的结构做简单的划分。

我们给出并行体系结构的定义:它能提供一种进行

并行程序设计的结构,这种结构通过多个简单或复杂的处理机的并发执行,来协同解决问题。进而我们给出并行体系结构的分类(见图 1),从这种分类方法中,尽量反映出体系结构特点上的变化。

下面我们将简要介绍各种并行结构的主要特点,有关的互连技术,然后讨论一下并行

结构中相应的一些有关软件环境的问题。

二、并行计算机体系结构

1. 同步体系结构

同步并行体系结构通过全局时钟、中央控制部件控制器来协调一致的并行操作。

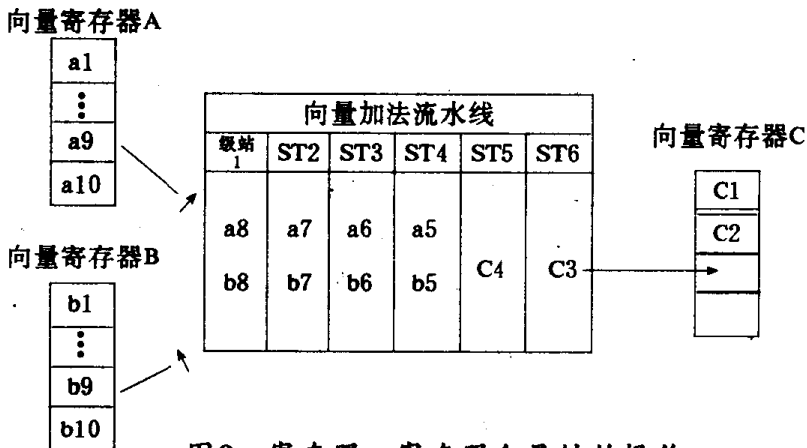


图2 寄存器-寄存器向量结构操作

(1) 流水线向量处理机

60年代末70年代初, 第一台向量处理机(Star100)问世。向量处理的特点是利用多个流水线功能部件, 并行地执行向量或标量的算术、布尔运算。图2给出了向量寄存器A、B的内容, 通过向量加法流水线得出向量元素C_i的示意图。向量加法部

件可包含6个流水级站。在使用寄存器-寄存器的结构中, 操作数和结构都放在指定的高速向量寄存器中(如Cray-1、富士VP-200)。流水线向量处理机可以获得较高的运算速度, 但由于其指令系统庞大, 必须使用大量的寄存器和缓冲器, 控制十分复杂。

近年来的巨型机(如Cyrax-MP/4和ETA-10)通过大的共享存储器, 将4~10个向量处理机连接起来, 由于这种结构可以支持任务级并行, 它们通常被认为是MIMD结构, 向量处理能力只被认为是它们的基本设计目标。

(2) 一般的SIMD结构

SIMD结构一般由中央控制部件、多个处理器以及用于处理器间通讯或处理器-存储器间通讯的互连网络(IN)组成。控制部件将每条指令都传送到所有的处理器上, 这些处理器以同样的方式, 对本地数据执行该操作, 当然也可以允许某个处理机不执行当前指令。SIMD的这种具有单个指令(I)部件和多个等同执行(E)部件的结构特征, 便于做向量等处理。

从原理上讲, 它和流水线向量处理机有较大的差别:

①利用大量重复的处理单元对向量所包含的各个分量同时进行计算, 而不同于流水线处理机的时间重叠;

②利用增多处理单元的个数, 提高运算速度, 不同于流水线处理机的缩短时钟周期。

所以, 只有在硬件价格大幅度下降的情况下, SIMD才能获得较好的性能价格比。实际上, 并行处理机产品往往兼有流水线和一般SIMD的特点。

• 阵列机体系结构

阵列机多用于大型科学计算。60年代末发展的阵列处理机(如ILLIAC-IV和BSP)都使用能容纳字长度操作数的处理器。操作数通常是浮点值(32~64), 不同的IN(互连网)

用来提供处理器-处理器、处理器-存储器之间的通信,其中最为常用的是网状(mesh)和交叉总线(crossbar)。

另一种阵列处理机是利用大量的1位处理器。在位平面式结构中,处理机的阵列是按对称格状形式排列的(如 64×64),与之相关的还有与处理机格状阵列大小相对应的多个存储器位平面。位于处理器格(X,Y)处的处理器,在所有有关的存储器平面(X,Y)处的存储器位上进行操作。通常的操作有拷贝屏蔽,以及在整个平面或一个平面内的列、行上的算术操作。

Loral 的 MPP 和 ICL 的 DAP 就是这种结构的典型例子,适用于图象处理。

• 关联存储器处理机体系结构

围绕关联存储器建立的计算机形成了另一种类型的 SIMD 结构,但造价较高。它使用特定的比较逻辑,按照数据的内容,而不是按寻址的方式,并行地访问存储的数据。关联存储器的研究开始于 50 年代末,开始是用于并行查找与指定的内容相匹配的数据。关联存储器处理机起始于 70 年代初期(如 Bell 的 PEPE),后来又有 Loral 的 Aspro,这类处理机适合面向数据库的应用,如跟踪检索和监视。

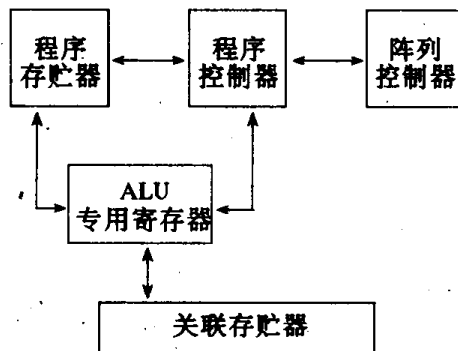


图3 关联存储器处理组织

图 3 给出了关联存储器处理机的功能部件,程序控制器仍按常规读和执行指令,当遇到需访问关联存储器指令时,就激活专用阵列控制器。专用寄存器允许程序控制器和关联存储器共享数据。

目前的大多数关联存储器处理机采用位串行组织,可以引发关联存储器中的所有字的某个位片(位列)上的并行操作。每个关

联存储器字在一般情况下有很多位(如 32768),该字与专用寄存器和比较逻辑有关,因此,一个有 4096 个字的关联处理机可以有 4096 个处理单元。

(3) Systolic(脉动)结构

80 年代初期,卡内基梅隆大学的 H. T. Kung 提出了 Systolic 结构,以解决专用系统存在的大量计算与 I/O 频带宽度之间的平衡问题。Systolic 结构是流水线的多处理机结构的一种,但数据被有节奏地从存储器中以脉动的形式产生,经过处理机网络后,返回存储器(如图 4 所示)。

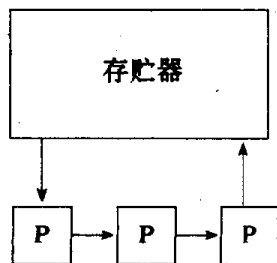


图4 进出存储器的Systolic数据流

此数据流含有从存储器中获得的操作数和每个处理器要用的中间结果。由规则的局部互连网连接的模块处理机,为各种各样的专用系统提供基本的构造块,在系统运行的每个时间间隔内,这些处理机执行一个短的、不变的指令序列。

Systolic 通过高度的并行计算能力和避免 I/O 与

存贮器之间的瓶颈,使得专用系统有较高的性能。数据在多个处理器中流动,获得了高并行度。数据从存贮器或外部设备中获得,一旦它进入 Systolic,就被传递给需要它的处理器,不再受存贮器存取的影响。只有阵列拓扑边界的处理器才执行 I/O 操作。越来越多的专用系统采用 Systolic 结构,特别是在信号处理方面。此外,可编程的(可重置的)Systolic 结构已经出现,如卡内基梅隆大学的 Warp 和 Saxpy 的 Matrix-1,它们不仅仅局限于只实现一种算法。尽管 Systolic 的概念最初是为了实现基于 VLSI 芯片的系统而提出的,但目前已在多种物理级上实现。

2、MIMD 结构

MIMD 结构采用多处理机,可以各自执行独立的指令流并使用局部数据。MIMD 结构是异步计算机,硬件控制是非集中式的,但各处理机在执行进程中,可通过 IN 传递信息,达到同步目的。

促使发展 MIMD 结构的因素有几个:一是它能支持更高的并行度(子程序、任务级或更高),二是 n 处理机系统的性能价格比远比由 n 个单处理机组成的系统的性能价格比高。目前流行的小巨型机多用 MIMD 结构。

(1) 分布式存贮器结构

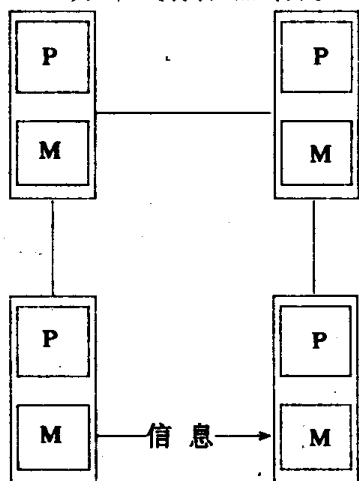


图5 MIMD分布式存贮体系结构

如图 5 所示,利用 IN 将处理机结点连接起来(包括自主处理机和它的局部存贮器),通过 IN 可传递信息。结点可共享数据,但没有任务共享存贮器。

• 环形拓扑结构

环的通信直径($N/2$)可以通过增加弦连接来降低,使用弦连接成多个环还可增加环形结构的容错性能。通常使用固定大小的信息包,其中包含结点目的地字段。环形适用于小数目的处理机和执行数据通讯不大的算法。

• 网状拓扑结构

二维网状(或格状)有 n^2 个结点,每个均与 4 个相邻结点直接连接。有时在边缘提供卷绕连接,可将通讯直径从 $2 \times (n-1)$ 降至 $2 \times (n/2 \text{ 的整数部分})$ 。通过提供附加的对角线连接,或对连接成行(或列)的结点使用总线,还可增大通讯量。这种结构适用于面向网、矩阵的算法。

• 树形拓扑结构

如哥伦比亚大学的 DADO2 和 Non-Von,树形支持查找和排序的征服算法、图象处理算法、数据流以及归约程序设计。尽管有不同的树形结构拓扑,但主要是完全二叉树,其通信直径为 $2 \times (n-1)$ 。

• 超立方体拓扑结构

n 维超立方体,使用 $N=2^n$ 个处理机,每个结点由 n 位 $0-N-1$ 个数值做为该点的地址进行唯一标识,要保证相邻结点的值有且只有一位不同。这种结构的出现是为了满足三

维科学应用的需要。Cosmic Cube, Ametek Series20/0, 及 Ncube/10 均属此结构。

• 可重配置拓扑结构

它具有分布式存贮器体系结构的基本物理拓扑结构, 但又提供可编程开关, 允许用户选择与应用通信形式匹配的逻辑拓扑结构。目前有两种类型: 特定不同的拓扑结构(如 Lawrence Snyder 的可配置高并行计算机, 即 Chip)和在基本的拓扑结构中加入同一类型的多个互连拓扑(如 Howard J. Siegal 的可加入的 SIMD/MIMD 系统, 即 Pasm)。研究此种结构的动机是寻求一种作为许多专用结构的特定结构, 以便有效支持特定算法或其特有的通信方式。

(2) 共享存贮器体系结构

通过提供全局、可共享的存贮器, 来实现多处理机间的协调, 这类体系结构机器有 80 年代的 Flex/32 和 Multimax。它包含多个共享存贮器的通用处理机, 而不是一个 CPU 加上外围 I/O 处理机。共享存贮器计算机避开了信息传递遇到的一些问题, 但又必须解决象数据访问同步化和高速缓存连续性问题。

协调带有共享变量的处理机要求具有很好的同步机制, 以防止进程在另一个进程完成修改数据之前, 访问该数据。

通常, 共享结构中的每个处理机有一个局部存贮器用作高速缓存。因此, 同一共享存贮器数据就可能有多份拷贝在多个处理机的 cache 中, 必须保证这些数据在 cache 中的版本一致(即相关性问题)。无论何时当一个处理机更新此数据的拷贝时, 每个处理机的 cache 中的此数据也要更新。在处理机数目较少的系统中, 可以使用硬件“侦听”机制确定何时更新了可共享存贮器数据; 较大的系统则通常依靠软件来解决。

• 总线互连

分时共享总线为每个处理机访问可共享存贮器提供了公正简单的方法, 单总线可接纳中等数目的处理机(从 4 至 20 个), 一次只有一个处理机访问总线。还可采用两种总线: 连接一组处理机的局部总线和连接与每组有关的特定服务处理机的更高层的系统总线。

• Crossbar 互连

使用 n^2 个交叉点的 crossbar 开关, 将 n 个处理机与 n 个存贮器互连。在每个可能的 P/M 对之间提供专用通路, 从而防止出现竞争冲突。但因这种互连配置的价格较贵, 处理机数目一般多在 4 至 16 个之间。Alliant FX/8 就是采用此种互连。

• 多级互连网络(MIN)

是 Crossbar 和总线的性能/价格比的折衷。MIN 是通过多级开关互连网将 n 个处理机与 n 个存贮器连接。进行存贮器访问请求的处理机通过发出含有每级控制位的位值, 来指定所希望的目的地, 在级 i 上的开关检查第 i 位, 以决定输入请求是被连至上输出, 还是下输出。MIN 的最大特点是可扩充。

3. 类 MIMD 体系结构

有些并行系统的体系结构类型既以异步操作和多指令流多数据流并行操作的 MIMD 原则为基础, 又不同于一般 MIMD 的组织原则, 在本文中, 我们称之为类 MIMD 结构。这些结构有 MIMD/SIMD 混合结构、数据流结构、归约机以及波前(wavefront)阵列。

(1) MIMD/SIMD 结构

80年代计算机界出现的 MIMD/SIMD 混合结构允许 MIMD 结构中的被选定部分按 SIMD 的控制方式工作(如 DADO、Non-Von、Pasm 和 TRAC)。这种结构和控制 SIMD 执行的实现机制是完全不同的。我们通过使用树结构、信息传递计算机作为 MIMD/SIMD 的基本结构,来说明这种结构的一般概念。

SIMD 控制器和处理机之间的主/从关系可以映射为树状结点及其子孙之间的关系。子树的根处理机结点作为 SIMD 的控制器,它将指令传送至子孙结上,在子孙结点上,根据局部存储器中的数据执行指令。MIMD/SIMD 的结构灵活,已成为人们研究的重点,适合于支持并行图象处理和专家系统。

(2)数据流结构

数据流计算机与冯·诺依曼计算机不同,它是数据驱动计算机。其特点是用操作数驱动操作,而不是由指令驱动操作。也就是说,当操作数全部到齐后,按操作数的操作指示开始执行操作。这样,如果有很多个操作均满足上述条件,它们就可以彼此并行执行而不受指令顺序执行的限制,从而有可能充分开拓并行度,实现任务、子程序和指令级的并行操作。虽然数据流机适用于大并行度,但是存在以下几个问题:

①由于使用数据驱动的概念,控制数据流在流动时需要相当多的辅助信息,从而使得数据流计算机对存储器的要求远比冯·诺依曼机大(约二、三倍)。

②由于数据的相关性,使其在操作级的并行度并不太理想。数据流原理对标量运算可能是吸引人的,而对数组、递归等却难以管理。

③数据流机的互连网络的设计仍是一个难题。

④由于数据流计算机中,不存在程序计数器,为诊断和维护带来了困难。因此,尚未有实用的数据流计算机应市。

(3)归约体系结构

归约体系结构的特点是要求驱动,即当一条指令的结果要求作为另一条已允许执行的指令的操作数时,才允许执行该指令。这种体系结构的研究起始于 70 年代末,为的是支持功能程序设计语言。

归约体系结构执行含有嵌套表达式的程序,表达式是递归定义的,程序可以引用命名的表达式。归约程序可识别可归约的表达式,然后用计算出来的值代替它们,归约程序最后归约出它的结果。归约体系结构能进行串归约和图归约。

目前支持串归约和图归约的机器,如 Newcastle 归约机、北卡罗林纳的 Cellular Tree 机;Utah 的应用多处理系统均属试验机。

(4)波前阵列体系结构

波前阵列处理机将 Systolic 的数据流水线与异步数据流操作结合起来。S·Y. Kung 于 80 年代初提出波前阵列的概念,其目的是为了给专用系统(大计算量与 I/O 带宽之间的平衡)提供有效的结构。波前和 Systolic 结构的特点都具有模块处理机和规则的局部互连网。但波前阵列用异步的“握手”作为协调进程间数据运动的机制,从而代替了同步 Systolic 数据流水的全局时钟。因此,当处理机执行计算并准备将数据传送给它的后继者时,就通知后继者,一旦后继者表示准备好,该处理机就发送数据,并接收来自后继者的回答信号。“握手”机制使得波前阵列处理机没有交叉地平缓前进。

Kung 认为波前阵列处理机有几个特点优于 Systolic, 即更大的可量测性, 简单的程序设计及更大的容错性能。霍普金斯大学和斯坦福大学正在研制波前阵列处理机。

三、软件环境

实现高并行度的并行处理不仅依赖于体系结构、硬件上的改进和发展, 更依赖于软件产品的支持。现就并行处理所需软件环境中的共存问题做一介绍。

1、并行操作系统

并行操作系统从多机操作系统发展而来, 目前比较成功的是卡内基梅隆大学研究的 Mach 操作系统。

多机操作系统要解决处理机间的通信问题以及避免系统出现死锁状态。在共享存贮器 MIMD 系统中, 需要通信的进程或处理器都可以直接去存取共享存贮器的数据, 且保证共享数据的完整性。解决的办法是使有关的各个进程同步地去存取共享数据, 所采用的技术有信号灯同步技术、监督体同步技术及互容方式的监督体同步技术等。在分布式 MIMD 结构中, 主要是利用互连网, 以信息形式进行通讯, 各处理机要遵守通信协议。

多机系统的 OS 在解决死锁问题上有三种办法:

- ①预防算法: 用于系统的设计阶段;
- ②避免算法: 用于系统运行期间;
- ③检查恢复算法: 也用于系统运行期间, 通过超时与否来判断是否出现死锁。

并行 OS 除了要解决上述问题外, 还要进行并行进程的控制和调度, 这包括并行任务的派生与汇合、并行任务的同步与互斥以及资源分配和进程调度。并行 OS 以充分发挥特定并行体系结构的效能为目标, 因此, 采用并行编译系统、相应的并行算法, 才可能获得该并行体系结构的最大并行度。

2、并行化编译系统

编译系统是计算机用户求解问题的一个有利工具。SIMD 和 MIMD 的出现又对编译系统提出了不同于标量机编译系统的新要求, 即应能够完成程序向量化及并行化。从本质上说, 程序向量化是一种程序变换, 也就是从一段给定的标量程序生成向量程序的过程。一般向量编译系统应有四方面功能: 相关分析、和量化(程序变换)、特殊指令识别和语言的完整性。但是, 实际的并行编译系统是复杂的, 与向量编译不同的是并行化常需考虑同步问题, 对不具共享存贮的并行编译还要考虑存储分配。Convex C 系列的 FORTRAN 5.0 编译系统不仅有较强的向量化能力, 也有很强的并行功能, 特别是具有处理机动态分配功能。

利用并行化编译系统, 程序员只需按以前的常规方法编写非并行的程序。并行化的工作由编译程序和运行的 OS 来完成。并行编译系统可以自动地将程序中能够并行化或向量化的部分抽取出来, 或标入并行指令, 或按子块将其分配到并行处理部件上去。这种自动转换功能虽然使用户旧有程序可以继续使用, 但程序员无法按并行概念去更好地安排自己的程序, 解决这个问题的办法就是采用并行语言编程。

目前流行的小巨型机上都配有向量化并行编译系统, 如 Alliant FX/2800 就采用了并

行化编译技术,能自动使用并行处理硬件执行 FORTRAN 语言。

并行编译系统目前还没有统一标准,不同的并行机系统应提出适合自己结构特点的并行机制。任何编译系统对用户程序的优化、向量化和并行化都是有限的,为了提高具体应用中实际程序的运行效率,有时需要人工干预改变算法。

3. 并行语言

并行处理系统上使用的高级程序语言可分为二类:一类是传统的为广大程序员所熟悉的串行语言;另一类是专为并行处理目的设计的新语言——并行语言。直接使用传统串行语言的好处是现有程序不需改动即可使用,程序员也不必重新训练,但是用来描述大量数组运算时的效率却很低。而全新的并行语言,不仅要描述并行进程,还要考虑程序分支、程序循环、并行进程间的通信与同步、数组和进程组的处理等。目前的并行语言有面向并行进程通信的语言 CSP,及广泛用于小巨型机中的 Occam 语言。Occam 语言结构简洁,它通过一些结构符来描述串行进程、并行进程、程序分循环、并行进程间的通信与同步,拥有很好的并行处理能力和同步通信能力。

4. 非数值计算的并行算法

在并行处理系统中,除了需要上述 OS、编译系统、并行语言的支持外,还需有并行算法的支持。并行算法与并行处理系统的结构有密切关系。

非数值计算的提法是相对于数值计算而言的,数值计算是指基于代数运算的一类,诸如矩阵等运算的计算问题。非数值计算指基于关系运算的一类,诸如排序、选择等的计算问题,图论算法也是一大类非数值计算的问题。非数值计算有着广泛的应用前景,智能体系结构、数据库、知识库管理系统、并行与分布式处理系统等均用到它们。

并行算法可分成同步运行和异步运行,在 SIMD 机器上和 MIMD 机器上的以及 VLSI 的并行算法。其中运行在 MIMD 机器上的算法又常分成多处理机(紧耦合)系统中的异步并行算法和分布式算法(松耦合)。

通用并行结构(SIMD, MIMD)上的算法常用运行时间 $T(n)$ 和所使用处理机数目 $P(n)$ 来衡量,成本函数 $C(n) = T(n) \times P(n)$ 。

VLSI 上的并行算法常用芯片面积 A 和电路速度 T (计算时间)来衡量, AT^2 是衡量算法的主要标准。

四、结束语

随着对高并行度的迫切要求,并行处理技术也在不断发展。当前 RISC 技术已应用在全联盟的新一代小巨型机 FX/2800 上。此外,将通信与处理功能集成在一块处理器芯片的趋势,也为发展海量并行处理系统提供了基础。在软件环境中,人们希望能遵循一些共同的原则和标准,以便于各种应用程序的移植,为此,并行处理系统的软件环境也将逐步走向开放式计算技术软件标准的道路。

并行处理——先进技术

华北计算技术研究所 尹祯祚

并行处理从道理上讲很简单。如果 32 位微处理器能够抵得上超级小型机,将许多这样的处理器组合在一起工作,足可抵得上一台巨型机。实际上,有些厂家把他们的并行机称做巨型机,的确以处理机的数量和处理速度来提高功能。

小的开发公司试图在已经拥挤的计算机商业领域中打开市场,并行处理系统在需求巨型机功能的用户当中已很时髦。但这些用户财力有限,只能买得起超小型或中小型机。目前流行的许多并行系统的价格在 5 万美元的小型机和 100 万美元的超小型机之间。

首批并行系统已装在大学、公司和政府级研究机构,正当它们在这些机构中发挥作用时,许多厂家看到了它们的潜在市场。开始,虽然技术上没有经验,但商业应用已显示出良好的前景。值得一提的是,有的厂家刻意追求事务处理、相关数据库管理以及正文检索,而这些应用又使一般系统力所不及。

并行系统分为三组:“粗粒”、“中粒”和“细粒”系统,2~30 个处理器的为“粗粒”系统,500 或更多个处理器为“中粒”系统,达数千个处理器的为“细粒”系统,三组基本都采用不同的处理器互连结构。在粗粒系统中,处理器通过公共的面向总线结构交换信息;在“中粒”和“细粒”并行系统中,数以百计的处理器通过相当于微型化了的电话网络进行通信。

什么叫并行处理

多年来,并行处理和多处理一直混淆不清,更不必说阵列和向量处理。但是,有时它们又有相似之处,六十年代初已经认识到它们结构上的不同。厂家经常阐述它们的差别,以便说明优于其它技术。计算机系统可以归并为如下四种体系结构:

- ① 单指令单数据流(SISD)
- ② 多指令多数据流(MIMD)
- ③ 单指令多数据流(SIMD)
- ④ 多指令单数据流(MISD)

许多普通单处理机称为 SISD 系统,它们采用标量处理的方法,一次取出并执行一条指令,许多微型和大型商用系统都采用这种方法。尽管是串行处理,采用先进的多用户系统常常被视为同时处理多个问题。实际上,这些系统一直通过多年来使用的分时技术,在单处理机上分时完成多个计算任务。

并行计算机通常被称之为 MIMD 系统,因为这种系统可使两台或多台处理机同时从事同样的处理工作。随着处理机速度的提高,已使任务的处理速度成线性上升。真正采用并行的机器不会和多处理系统混淆。多处理属于使用两个或多个处理机的通用系统。当一种多处理机结构令其系统处理多用户作业时,每个计算任务只用一个处理机进行处理。

单指令多数据流(SIMD)系统主要使用多个处理单元根据浮点加和多数据的阵列同