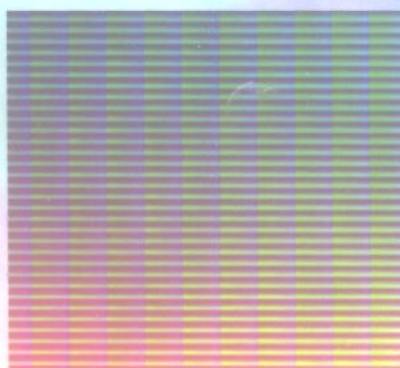


● 屈玉贵 梁晓雯 编著

并行处理 系统结构



中国科学技术大学出版社

TP338.6

428327

Q86

并行处理系统结构

屈玉贵 梁晓雯 编著

中国科学技术大学出版社

1999 · 合肥

JS134/17
内 容 简 介

本书是并行处理系统的硬件篇，主要包括并行处理系统结构的基本概念、基本原理及结构的设计和分析。

全书共分为十章，前六章系统地介绍了并行处理系统结构的基本概念及组成并行处理系统的各种构成元素，包括处理机、存储系统、流水线技术、互连网络及向量处理技术等。第七章至第九章着重讨论了共享存储器型多处理机系统、分布存储器型多处理机系统、阵列机和数据流计算机的构成与性能分析。

本书可作为高等院校计算机、信息系统及电子工程等相关专业的高年级本科生或研究生的教材，也可作为有关专业的教师、科研人员及工程技术人员的参考资料。

图书在版编目(CIP)数据

并行处理系统结构/屈玉贵 梁晓雯 编著. —合肥:

中国科学技术大学出版社, 1999 · 2

ISBN 7-312-01066-0

I. 并... II. 屈... III. 并行计算机-系统结构 IV. TP338 · 6

中国版本图书馆 CIP 数据核字(98)第 40712 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号 邮编: 230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本: 787 × 1092/16 印张: 27 字数: 640 千

1999 年 2 月第 1 版 1999 年 2 月第 1 次印刷

印数: 1-3000 册

ISBN7-312-01066-0/TP · 225 定价: 28.00 元

序

该书比较系统地阐述了构成并行处理系统的一些基本单元及由它们构造的系统，内容全面，可读性好。在以往研究的基础上，作者及时地添加了当今并行处理系统结构中的一些新的研究成果，使该书不仅可作为基础课程的教材，而且对科研人员也具有参考价值。从这本书的结构上看，该书首先讨论并行处理系统的各种组成技术，然后再分析整个系统性能，这种安排也是比较新颖独特的，有利于读者由浅入深地进行学习；从内容选取上看，本书加强了有关阵列并行机的内容，从而能较好地满足信号处理专业的学生学习的要求。我相信《并行处理系统结构》一书的出版，定能获得良好的教学效果，能使更多的读者受益。

陈国良

前 言

在计算机发展初期，并行处理就显示出了它在提高整个计算机系统性能方面的重要性。随着超大规模集成电路的不断发展，多处理机及多计算机系统已经实现。在计算机技术高速发展的今天，只要涉及到计算机系统结构的有关知识，就必然离不开并行处理的概念。

国内并行处理系统的研究始于八十年代初期，经过近二十年的发展，涌现了不少有关并行处理的著作。但是随着时间的不断推移，新的技术也不断涌现。为了更加系统全面地介绍并行处理系统结构，及时补充一些新的内容，我们编著了这本书。

本书集作者长期从事系统构成技术等有关课程的教学和科研经验，在查阅了大量国内外最新资料的基础上编写完成。全部内容分为两大部分，第一部分是并行处理系统构成元素的基本概念，这些构成元素包括处理机、存储系统、流水线技术、互连网络及向量处理技术等。第二部分是并行处理系统的构成与分析，阐述了并行处理系统的基本概念以及共享存储器型多处理机系统、分布存储器型多计算机系统、阵列机和非冯-诺依曼系统的基本构成与性能评价。本书的特点在于不是按类型讲述各类并行处理系统，而是首先讲述各种积木式的构成元素，然后讲述由这些构成元素组装的各类并行处理系统的结构特点和性能分析。

本书力求内容新颖全面，重点突出，语言精炼，通俗易懂。可作为高等院校计算机、自动化及电子工程等相关专业的高年级本科生、研究生的教材，也可作为有关专业的教师、研究人员及工程技术人员的参考书。

如果本书作为教材，参考学时为 80 ~ 100 学时。第一部分的 1 ~ 6 章是并行处理系统的基础，作为课程的必讲部分，用 40 ~ 60 学时讲授，第 7 ~ 10 章的几种并行处理系统可根据需要进行取舍，用 20 ~ 40 学时讲授。

作者非常感谢中国科学技术大学计算机系陈国良教授的关心和支持，特别感谢陈国良教授在百忙之中为本书写序。

由于作者的水平和经验有限，难免有疏漏及不当之处，希望读者批评指正。

作者

一九九九年一月

目 录

第一章 绪论	(1)
1.1 并行处理的概念	(2)
1.1.1 并行性的概念	(2)
1.1.2 并行等级的划分	(2)
1.1.3 实现并行处理的各种技术途径	(3)
1.2 并行处理的基础	(4)
1.2.1 并行处理的目的和效果	(4)
1.2.2 并行处理的应用和实现	(6)
1.2.3 并行计算模型和图形表示	(9)
1.2.4 并行算法的构成	(11)
1.2.5 处理机的分配	(12)
1.2.6 同步和通信	(13)
1.3 并行处理机的系统构成方式	(13)
1.3.1 并行处理机系统构成分类	(13)
1.3.2 处理单位	(14)
1.3.3 结构	(16)
1.3.4 处理方式	(17)
1.3.5 控制方式	(18)
1.3.6 连接方式	(19)
1.3.7 互连网络结构	(20)
1.3.8 通信方式	(21)
1.3.9 并行度	(22)
第二章 处理机	(23)
2.1 概述	(23)
2.2 应用对选择处理机的影响	(24)
2.3 微程序技术	(25)
2.3.1 微程序技术的发展	(25)
2.3.2 微程序技术中的一些基本概念	(26)
2.3.3 微程序设计技术	(28)
2.4 指令与寻址	(34)
2.4.1 指令系统	(34)
2.4.2 指令格式	(34)

2.4.3	寻址方式	(36)
2.4.4	指令系统举例	(39)
2.4.5	指令系统的改进	(46)
2.5	复杂指令集计算机(CISC)	(46)
2.6	精简指令集计算机(RISC)	(51)
2.6.1	RISC 提出的背景	(51)
2.6.2	RISC 的发展	(52)
2.6.3	RISC 的特征	(53)
2.6.4	RISC 的指令系统	(54)
2.6.5	RISC 的构成技术	(59)
第三章	存储系统	(63)
3.1	存储系统的层次结构技术	(63)
3.1.1	存储器的层次结构	(63)
3.1.2	程序的局部性	(63)
3.1.3	层次存储器的设计准则	(63)
3.2	虚拟存储器技术	(65)
3.2.1	虚拟存储器概念	(65)
3.2.2	分页存储管理	(65)
3.2.3	分段存储管理	(70)
3.2.4	段页式存储管理	(71)
3.2.5	虚拟存储管理中的替换算法	(72)
3.2.6	影响主存命中率的因素	(74)
3.3	高速缓冲存储器(Cache)	(77)
3.3.1	高速缓存的基本结构	(77)
3.3.2	高速缓存的地址映象与变换	(77)
3.3.3	高速缓存的替换算法	(83)
3.3.4	高速缓存的一致性	(85)
3.3.5	高速缓存的优化	(91)
第四章	流水线技术	(98)
4.1	流水线方式的基本原理	(98)
4.1.1	流水线方式的分类	(98)
4.1.2	线性流水线技术	(99)
4.1.3	非线性流水线技术	(102)
4.1.4	流水线的相关处理	(110)
4.2	超标量流水线和超流水线	(116)
4.2.1	超标量流水线	(116)
4.2.2	超流水线技术	(118)
4.2.3	超流水线超标量处理技术	(119)

4.2.4	性能分析.....	(119)
4.3	VLIW 方式.....	(120)
4.3.1	VLIW 方式的基本工作原理.....	(120)
4.3.2	VLIW 方式的特征.....	(122)
4.3.3	VLIW 处理机的工作方式.....	(123)
4.3.4	VLIW 的优化编译.....	(126)
第五章	互连网络.....	(131)
5.1	互连网络的分类.....	(131)
5.2	互连网络的互连函数和结构参数.....	(132)
5.2.1	互连函数.....	(132)
5.2.2	互连网络的结构参数.....	(136)
5.3	静态互连网络.....	(137)
5.3.1	线性阵列.....	(138)
5.3.2	环网和带弦环网.....	(138)
5.3.3	全连接网.....	(139)
5.3.4	循环移数网.....	(139)
5.3.5	星形网络.....	(139)
5.3.6	网格形网络.....	(140)
5.3.7	树形网.....	(141)
5.3.8	立方体网络.....	(143)
5.3.9	Paradhan 网.....	(146)
5.3.10	总线.....	(146)
5.4	动态互连网络.....	(148)
5.4.1	单级互连网络.....	(149)
5.4.2	多级互连网络.....	(152)
5.5	互连网络中开关元件的设计与实现.....	(170)
5.5.1	2×2 交换开关的设计与实现.....	(170)
5.5.2	4×4 交叉开关的设计与实现.....	(176)
第六章	向量计算机.....	(181)
6.1	向量处理和向量指令.....	(181)
6.1.1	向量处理.....	(181)
6.1.2	向量指令.....	(181)
6.2	向量计算机的结构.....	(183)
6.2.1	存储器-存储器结构的向量计算机.....	(184)
6.2.2	寄存器-寄存器向量流水线结构.....	(187)
6.2.3	中间寄存器.....	(189)
6.3	向量计算机的数据结构.....	(192)
6.4	向量循环和链接.....	(196)

6.4.1	链接	(196)
6.4.2	向量循环	(198)
6.5	流水线网络	(199)
6.5.1	流水线网络模型	(199)
6.5.2	建立流水线网络的过程	(199)
6.6	向量处理机举例	(201)
6.6.1	向量协处理器	(201)
6.6.2	向量巨型计算机	(203)
6.7	向量计算机系统结构的设计准则	(208)
第七章	共享存储器型多处理机系统	(210)
7.1	基本模型	(210)
7.1.1	系统结构模型	(210)
7.1.2	PRAM 模型	(211)
7.1.3	存储器的一般模型	(212)
7.2	UMA 型多处理机系统	(214)
7.2.1	总线连接型多处理机系统	(214)
7.2.2	交叉开关型 UMA 多处理机系统	(246)
7.2.3	多级互连网络型 UMA 多处理机系统	(248)
7.3	NUMA 型多处理机系统	(270)
7.3.1	单总线连接型 NUMA 系统	(272)
7.3.2	群结构的 NUMA 多处理机系统	(276)
7.3.3	层次总线型 NUMA 多处理机系统	(284)
7.4	COMA 多处理机系统	(288)
第八章	分布存储器型多计算机系统	(290)
8.1	分布存储器型多计算机系统的模型	(290)
8.2	多计算机系统的消息格式	(291)
8.3	同步方式	(291)
8.4	寻径方式	(292)
8.4.1	维序寻径	(293)
8.4.2	自适应寻径	(294)
8.4.3	单播、选播、广播、会议通信模式	(295)
8.5	多计算机互连网	(296)
8.5.1	多计算机互连网的构成法	(296)
8.5.2	通信距离和中继量	(298)
8.5.3	中继规则	(299)
8.5.4	可靠性	(300)
8.5.5	互连网络的互连方式	(300)
8.5.6	互连线路	(309)

8.6	并行处理方式.....	(311)
8.6.1	数据分割解法.....	(311)
8.6.2	处理分割方式.....	(314)
8.7	二叉树网计算机 Coral 68k.....	(316)
8.7.1	系统构成.....	(316)
8.7.2	处理机.....	(316)
8.7.3	程序方式.....	(317)
8.7.4	处理机消息.....	(318)
第九章	阵列并行计算机.....	(321)
9.1	超并行计算机系统的发展.....	(321)
9.1.1	冯-诺依曼结构.....	(321)
9.1.2	SIMD 阵列机模型.....	(322)
9.1.3	网格体系结构.....	(323)
9.1.4	并行处理术语.....	(324)
9.2	脉动阵列.....	(325)
9.2.1	脉动阵列的结构和特征.....	(325)
9.2.2	在脉动阵列上实现的算法举例.....	(326)
9.2.3	脉动数据结构.....	(327)
9.3	细胞自动机.....	(332)
9.3.1	细胞自动机.....	(332)
9.3.2	点灯问题.....	(334)
9.4	带全局总线的网格阵列机.....	(338)
9.4.1	一维广播阵列机.....	(338)
9.4.2	二维广播阵列机.....	(339)
9.4.3	全局总线在脉动阵列上的应用.....	(341)
9.4.4	全局总线的省略问题.....	(343)
9.5	锥形阵列机.....	(343)
9.5.1	二维锥形阵列机.....	(343)
9.5.2	锥形阵列机及算法举例.....	(345)
9.6	树网阵列机.....	(346)
9.6.1	树网阵列机.....	(346)
9.6.2	基本运算.....	(347)
9.6.3	MOT 上的算法.....	(348)
9.7	排序阵列机.....	(350)
9.7.1	排序网络.....	(351)
9.7.2	排序网的构成方法.....	(352)
9.7.3	双调排序网.....	(354)
9.7.4	Batcher 的奇偶归并排序网络.....	(360)

9.8	洗牌交换阵列机.....	(364)
9.8.1	洗牌交换网络.....	(364)
9.8.2	多项式求解.....	(366)
9.8.3	洗牌交换阵列机上的排序.....	(367)
9.9	超立方体阵列机.....	(369)
9.9.1	超立方体结构.....	(369)
9.9.2	超立方体阵列机算法.....	(372)
9.10	阵列计算机举例.....	(376)
9.10.1	AAP-2 的硬件结构.....	(376)
9.10.2	AAP-2 的软件.....	(381)
第十章	数据流计算机.....	(384)
10.1	数据流驱动的概念.....	(384)
10.2	数据流语言与数据流程图.....	(385)
10.2.1	数据流语言.....	(385)
10.2.2	数据流程图.....	(390)
10.3	数据流计算机结构.....	(396)
10.3.1	静态数据流计算机.....	(397)
10.3.2	动态数据流计算机结构.....	(402)
10.3.3	其他类型的数据流计算机.....	(406)
10.3.4	数据流计算机的设计标准.....	(410)
10.4	数据流计算机的评价.....	(411)
	主要参考文献.....	(413)

第一章 绪 论

电子计算机从 1946 年问世以来,在五十年左右的时间里,取得了及其惊人的发展。以器件的发展为标志,计算机经历了电子管、晶体管、中小规模集成电路、大规模和超大规模集成电路四代。目前,在世界范围内正在广泛研究第五代计算机。由于器件的迅速发展,使得计算机硬件在体积、重量、速度、可靠性、稳定性等性能上有了极大的改善,而且价格也在不断降低。可以说,器件技术的突飞猛进,为计算机的发展提供了必不可少的物质基础,它是推动计算机发展的重要因素之一。

但是,还应该看到,仅有器件的发展是远远不够的。人们还致力于研究如何最合理地组织这些器件,如何最大限度地发挥这些器件的作用,如何构成综合性能最佳的系统,这就是计算机系统结构要研究的问题。恩斯洛(P.H.Enslow)曾经做了如下比较:在 1965 年至 1975 年的十年中,器件的延迟时间降至原来的十分之一,而计算机指令时间却降至原来的百分之一。由此可见,在这十年中,计算机性能提高的幅度比器件性能提高的幅度大得多。这种情况在近年的计算机发展中也尤为明显。因此,除器件的发展以外,计算机体系结构的发展也是推动计算机发展的一个重要因素。

在讨论计算机系统结构的同时,我们不能忽视计算机并行技术的发展。在器件技术发展的任何一个水平上,由于物理器件本身在速度、尺寸、功能上存在着无法克服的限制,使得并行处理技术的发展就成为同一时期提高系统性能的关键手段。

纵观计算机体系结构从低级向高级发展的过程,我们可以看到,这一过程也是并行处理技术不断发展的过程。最早的计算机并行是位并行方式。1952 年美国新技术研究所(IAS)研制了第一台位并行运算样机,1953 年完成了具有位并行运算的第一台商业计算机 IBM701。同一时期也出现了 CPU 与 I/O 设备在一定程度上的并行。到六十年代,并行性进一步发展,研制出了流水线单处理机系统,如 1964 年完成的 CDC6600 机,设有乘、除、加、长加、移位、布尔运算、递增等操作的 10 个独立的功能部件,它们可以被多个外围处理机分时复用。1967 年的 IBM360/91 计算机指令的取指、译码、地址计算等均采用流水线方式重叠地工作。在 1970 年至 1980 年这段时间里,大规模集成电路的快速发展,导致了向量计算机、阵列计算机、相联处理机等多种多样的并行处理系统。1976 年由 CRAY 公司研制的 CRAY-1 就是比较成功的向量流水线计算机,运算速度可达每秒一亿三千万次浮点运算。八十年代以来,计算机的体系结构又有了突破性的进展,最具代表性的是精简指令系统计算机(RISC)、数据流计算机和智能计算机等。这些系统结构的出现,使得计算机的并行处理技术向着更高的方向迈进。

1.1 并行处理的概念

1.1.1 并行性的概念

无论是数值计算、数据处理、信息处理还是人工智能问题求解，其中都可能包含有能同时进行运算或操作的成分。我们把问题中具有可以同时进行运算或操作的特性，称之为并行性(Parallelism)。开发并行性的目的是为了能予以并行处理，以提高计算机求解问题的效率。

例如，在使用具有相同延迟时间的元器件条件下，采用 n 位计算器进行 n 位并行运算的速度几乎是用一位运算器进行 n 位串行运算的 n 倍，这就是传统机器由串行到并行的典型事例。但是，对并行性的理解不能只限于这种靠器件资源的简单重复实现的并行，应当有更广义的理解。比如，单处理机内采用的重叠、流水方式工作，操作系统中采用的多道程序分时共行，都是更广义上的并行。也就是说，只要在同一时刻或是同一时间间隔内完成两种或两种以上性质相同或不同的工作，它们在时间上相互重叠，都体现了并行性。所以，并行性包括同时性和并发性二重含义。同时性(Simultaneity)指的是二个或多个事件在同一时刻发生。并发性(Concurrency)指的是二个或多个事件在同一时间间隔内发生。

1.1.2 并行等级的划分

并行性可以划分为不同的等级，而且从不同的角度出发，等级的划分方法也不一样。

1. 从计算机系统内部执行程序的角度来划分，可分为四级，从低到高分别是：

- 指令内部——一条指令内部各种微操作之间的并行。
- 指令之间——多条指令在某一时刻或同一时间间隔内并行执行。
- 任务或进程之间——多个任务或程序段之间的并行执行。
- 作业或程序之间——多个作业或多道程序之间的并行执行。

作业或程序的并行关键在于并行算法，即怎样将有限的硬件、软件资源有效地同时分配给正在用于求解一个大题目的多个程序。任务或进程级的并行主要讨论如何进行任务分解。指令之间的并行主要要解决和处理好指令之间存在的相互关联。指令内部的并行主要取决硬件和组成的设计。总之，并行性由高到低反映了硬件实现的比例在增大。所以，并行性的实现也是一个软硬功能分配问题。往往需要折中权衡。随着硬件成本的不断下降和软件成本的相对上升，硬件实现的比例还有增大的趋势。例如，过去在单处理机中并行性升到任务、作业级时，主要是通过操作系统中的进程管理、作业管理、并行语言、并发程序设计等软的方法实现，而现在在多处理机系统中，由于已有完成各个任务或作业的硬处理机，其并行性主要是靠硬件来提供的。

2. 从计算机系统中处理数据的并行性角度来看，并行等级也可分为四级，由低到高分别是：

- 位串字串——同时只对一个字的一位进行处理，通常指传统的串行单处理机，没有并行性。
- 位并字串——同时对一个字的全部位进行并行处理，通常指传统的并行单处理机，

开始出现并行性。

- 位片串字并——同时对许多字的同一位（称位片）进行处理，通常指传统的并行单处理机，开始进入并行处理领域。

- 全并行——同时对许多字的全部或部分位组进行处理。

3.从计算机信息加工的各个步骤和阶段来看，并行等级可分为：

- 存储器操作并行——可以采用单体多字、多体单字或多体多字方式在一个存储周期内访问多个字，进而采用按内容访问方式在一个存储周期内用位片串字并或全并行方式实现对存储器中大量的字的高速并行比较、检索、更新、变换等操作，典型的例子就是以并行存储器系统和相联存储器系统为核心构成的相联处理机。

- 处理器操作步骤并行——处理器操作步骤可以指一条指令的取指、分析、执行等操作步骤，也可以指如浮点加法的求阶差、对阶、尾加、舍入、规格化等具体操作的执行步骤。将操作步骤和具体操作的执行步骤在时间上重叠流水地进行，典型的例子就是流水线处理机。

- 处理器操作并行——为支持向量、数组运算，可以通过重复设置大量处理单元，让它们在同一个控制器控制下，按照同一条指令的要求对多个数据组同时操作，典型的例子就是并行处理机。

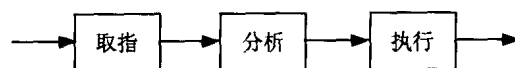
- 指令、任务、作业并行——这是较高级的并行，虽然它也可包含如操作、操作步骤等较低等级的并行，但原则上与操作级并行是不同的。指令级以上的并行是多个处理机同时对多条指令及有关的多数据组进行处理，而操作数并行是对同一条指令及其有关的多数据组进行处理。因此，前者构成的是多指令流多数据流(MIMD)计算机，后者构成的是单指令流多数据流(SIMD)计算机，典型的例子就是多处理机。

1.1.3 实现并行处理的各种技术途径

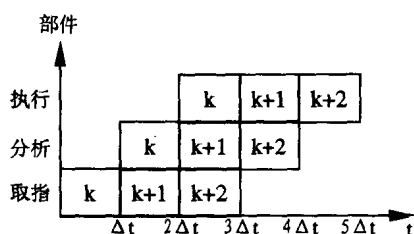
实现并行处理的技术途径是各种各样的，其中主要有时间重叠、资源重复和资源共享等方法。

时间重叠(Time Inteleaving)是在并行性概念中引入时间因素，使多个处理过程在时间上错开，轮流重叠地使用同一套硬件设备的不同部件，以加快硬件周转而赢得速度。最典型的时间重叠就是流水线工作方式，如图 1-1 所示。图中将一条指令的执行过程分解成取指、分析和执行三大部分，每一部分由相应部件完成。假定每一个部件的通过时间为 Δt ，那么在 $2\Delta t$ 到 $3\Delta t$ 的时间间隔里，第 $k+2$ 条指令正在取指令，第 $k+1$ 条指令正在分析，而第 k 条指令已经在具体执行了。从图中可看出，完成三条指令只需 $5\Delta t$ ，显然比指令串行执行的速度要快了许多。时间重叠原则上不需要重复增加硬件设备就可提高系统的性能。

资源重复(Resource Replication)是在并行性概念中引入空间因素，通过重复设置硬件资源，如处理器或外围设备等，来达到大幅度提高可靠性和处理速度的目的。图 1-2 为资源重复的例子，设置 N 个完全相同的处理器(PE)，让它们受同一个控制器(CU)控制，控制器每执行一条指令就可以同时让各个处理器对各自分配到的数据完成同一种运算。并行处理发展的早期，由于硬件价格较贵，因此，资源重复主要用来提高可靠性。现在硬件价格不断下降，资源重复被大量用于提高系统的处理速度。



(a)指令流水线



(b)指令在流水线各部件中流过的时间关系

图 1-1 流水线工作方式

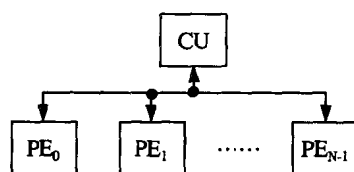


图 1-2 资源重复的例子

资源共享(Resource Sharing)就是让多个用户按一定时间间隔顺序轮流地使用同一套资源，以提高系统的整体性能。例如，多道程序分时系统就是利用共享 CPU、主存资源，以降低系统价格，提高设备利用率的例子。当然共享资源不仅限于硬件资源的共享，也应包括软件、信息资源的共享。

1.2 并行处理的基础

1.2.1 并行处理的目的和效果

计算机性能的提高和应用领域的扩大互相促进形成了一个计算机发展的良性循环。在今后的发展中将会遇到计算量更大的应用问题，这就要求计算机能进行更高速的处理工作。另一方面，如果要使一个处理机或少量共享存储器型的处理机能在广阔的领域里更加灵活地应用，那么其吞吐量、应答时间等都是要改进的内容。

随着半导体集成电路的进步，复杂的处理器和大容量的存储器已能实现小型化和低价格，特别是处理器和存储器都能用同样的材料制作，产量越大，生产成本就越低，因此，由多处理器组成的并行处理系统开始引人注目。在多处理机系统中，每个处理机都可带有自己的存储器，它们有机地结合起来，同时工作，共同解决一个大问题，组成了一个并行计算机系统。

并行处理的思想在 1960 年前后就通过实验样机实现了。现在，这方面的研究和实用化都已有了很大的飞跃。原因之一是元件技术、电路技术及安装技术都有了发展，处理机、存储器及连接回路都可小型化且造价低。原因二是在不少领域中，为了突破单处理机处

理能力的局限性，人们在问题分析及算法上都下了一番功夫进行研究，积累了专用并行机制作的经验。另外，新的非诺依曼结构的数据驱动、需求驱动等机型的理论研究和试验样机的制作技术也已成熟。

在这样的背景条件下，并行处理的研究和并行处理的开发都发展起来了，研究并行处理有三个目的：

(1) 处理时间缩短

若一个问题的处理过程以并行方式实现，则其处理时间就可缩短，而且能使问题应答的实时性得到保证。

(2) 可靠性和效率提高

通过多个处理机进行多重处理，系统的一部分误动作就可以被检出并得到纠正，使系统的可靠性提高。由于系统是由多个处理机组成，一个处理机发生故障时，可以马上用其他处理机代替，从而确保系统工作的连续性，很适合应用于实时处理或可靠性要求高的领域。

(3) 算法简明化

算法简明化本质上也含有需并行处理的问题，以前都是用串行算法来解决这类问题，但在大型问题中，还是要通过并行处理来解决。例如对自然现象的模型化和模拟分析就是用精巧洁的并行处理来实现的。

为了达到上述目的，使处理机系统发挥更大的效率，首先就要明确应用对象，设计出合适的算法和处理机结构，把问题内在的并行性有效地反映在物理级的处理机系统中。为此就要很好地把握对象的数据结构、处理过程的功能关系等。例如，在图象处理中，如果求相邻元素的亮度的平均值时，用二维格子状并行，如果要求整个画面的平均亮度，则可设计成树状结构。另外，求解问题的算法只有和系统的结构相适应，才能充分发挥系统的作用。

通用多处理机系统一般都有处理时间短、可靠性高、效率高、负载变动容易、易扩充等优点。但是，要想在解一个问题时，最大限度地发挥所有处理机的作用，这样的程序设计是很困难的，因为处理机的空闲时间及伴随并行处理而产生的多余操作的比率不能太高。

广义上讲，并行处理就是在具有相同目标的活动中，可以同时进行二个以上的动作或操作。推而广之，在自然界和人类社会中同样也存在这种现象。

(1) 技术

从计算机技术方面看，并行计算就是把串行计算的“时间”变换成“空间”。这里，不只是把计算分解成在时间上连续的基本操作序列，还要把计算分解成在空间上分散进行操作的集。如果并行计算能简单进行，那么由串行计算出发，把时间变换成空间就可以了。如果并行计算不能简单进行，则从开始就必须在并行算法上下功夫。

(2) 自然

并行计算方法是数学、物理等对象的自然规律。从万有引力及电磁学的基本法则看，这个世界上没有孤立的东西，事物都是互相依存的，即所有的事物都是连续地同时变化的。所以，对自然界最适合用并行计算的方法进行模拟。

(3) 社会

从社会现象方面看,社会组织都是分阶层(级别)的,有上级给下级分配工作,下级向上级汇报的组织形式,还有以部分工作为主体来集中整体工作的组织形式。前者对应各个单位算法的抽象化,后者对应部分主导型并行算法。

(4) 人类

如果考虑一下人的动作和思维,可以发现这正是并行处理的宝库。人的手足的动作及手指的动作都作为机器人的动作实现了,这就是并行性的具体化。人的大脑中进行着错综复杂的并行处理过程,人通过感官去认识事物、思考问题的形态等都是将来的研究课题。

人脑中进行的并行处理不是单纯追求高速性,人在处理问题时是以所看到的自然界的并行处理为目标的。当并行度大时,我们的思考能否跟的上对象的变化还是个问题。例如,百足虫的足的运动规则,千手观音的手的多样运动,其整体的规律是不一样的。因此,要能跳出串行计算中的决定性、规则性和秩序性的框框,扩展到非决定性、不规则性和无秩序性的领域中去。在串行运算中,性能、吞吐量、应答时间、收敛性都是可预测的,而在并行计算中,就很难预测了。

今后,并行处理要从以下方面去研究开发:

(1) 应用:探索适合于并行处理的领域,开发并行算法。

(2) 语言:设计并实现并行程序设计语言。

(3) 机器:研究灵活的并行处理系统构成法,以及如何实现系统间的互换。

首先,对于应用来说,就是要实现前述的目的和效果,并为此在并行算法上下功夫。这就要求弄明白具体应用中并行运行的单位的大小(粒度)和并行处理的数量(并行度),并要能进行多样组合,开扩并行处理的接口,然后通过整理分类,来考虑语言的设计和机器的组成。

其次,程序设计语言在并行处理中比在串行处理中更重要。如何定义并行操作进程,如何定义进程间相互干涉及关系,以及达到什么样的透明度等,这都是设计的分歧点。从普遍的并行性原理上讲,并行处理不但要通过设计通用的并行程序语言来简明地记述并行算法,而且要有并行思考问题的新方法,要丢掉串行计算的规范,从一开始就引入并行性。

对于并行机的构成、处理机的连接形式、存储器的配置和互连、通信和同步技术、处理器功能大小、处理机个数等方面,都要全面加以考虑。

在并行机中,任务的分割、处理机的分配、任务间的相互协调都是串行机不存在的问题。处理的粒度和处理机的粒度要相对应,程序的并行度和处理机的个数要相对应,任务间的关系要和处理机的连接相对应,这都是要研究的重要课题。

对于应用,要由语言和机器来支持,语言在机器上实现。但对于机器的结构来讲,应用、语言、机器这三者的顺序是不能忽视的。

1.2.2 并行处理的应用和实现

1. 并行处理的对象分类

在进行并行处理时,以什么为中心来并行地组织任务呢?这是因应用对象的特征不同而不同的。现从数据、装置及查找解的运行手段三个因素来考虑,如图 1-3 所示。 x 轴表示数据规模,数据越多,并行度越大。 y 轴表示功能模块个数(这里的个数对应于并行度的