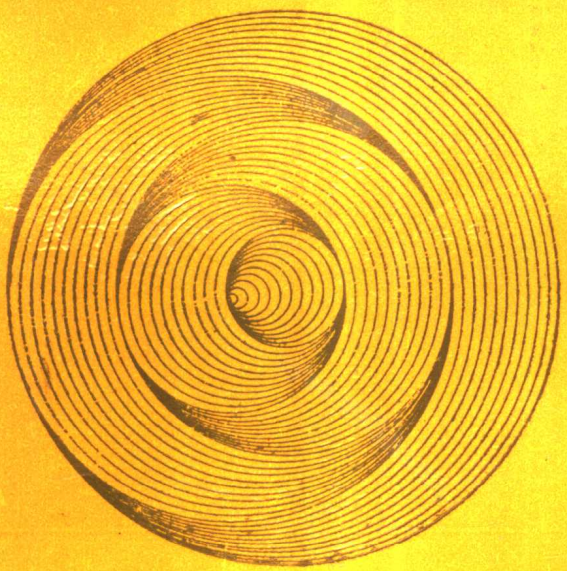
国家 863 计划资助项目

并行程序设计语言和模型

张德富　倪南　编著



南京大学出版社

并行程序设计语言和模型

张德富　倪　南　编著

国家 863 计划资助项目

南京大学出版社
1994·南京

(苏)新登字 011 号

并行程序设计语言和模型

张德富　倪　南　编著

*

南京大学出版社出版

南京大学校内，邮编 210008)

南京新利电脑照排中心照排

江苏省新华书店发行　　扬中印刷厂印刷

*

开本 850×1168　1/32　印张　6.25　字数　163　千

1994 年 7 月第 1 版　　1994 年 7 月第 1 次印刷

印数 1—6000

ISBN 7-305-02230-6/TP・70

定价 5.50 元

前　言

近 20 多年，并行处理技术一直是计算机科学技术领域内的重大研究课题之一。由于它对提高计算机的性能有着重大的作用，而且应用前景非常广泛，世界各国计算机工作者都非常重视这项技术的研究。

并行处理技术的难点在软件，并行计算机软件的质量对计算机系统性能的影响远比串行计算机的软件对计算机性能的影响大。在串行处理系统中，对于同一种给定的计算机，由于软件写得好坏而使计算机在运行中的性能差别只有 2－3 倍，最多 5－10 倍，但是在并行处理系统中，由于软件的好坏而表现出的性能差别可能有 50－100 倍。对于含有大量处理单元的并行处理计算机，软件对系统的影响更大，其性能差距可达几个数量级。因此对并行程序设计语言和模型的研究是十分重要的。

不仅在国内，即使在国际上，有关“并行程序设计语言和模型”的文献也是较少的，人们急需有关参考资料解决如何使用并行处理计算机、怎样进行并行程序设计等问题。在这时候，我们为能向国内广大读者奉献此书而感到高兴。

本书系统地介绍了 Occam、并行 C、并行 FORTRAN 和 LISP 等并行程序设计语言以及独立于结构的并行程序设计模型，后者主要是基于加里·萨博特著的有关文献而编写的。全书分绪论、并行程序设计语言（附有并行程序设计实例）、并行关系模型、基于并行关系模型的程序设计和并行关系模型的实现策略等。并行关系模型是一种独立于结构的并行程序设计模型，它为通用的并行程序设计提供了一种有效的方法。本书既可作为计算机专业高年级学生和研究生的教材，也可供有关科技人员参考。

本书的出版得到了863计划资助。在编著本书的过程中，孙钟秀教授等给予许多鼓励、支持和帮助，金凌紫副教授审阅了全书，并提出许多宝贵的修改意见，在此谨向他们致以诚挚的谢意。限于作者水平，书中欠妥及谬误之处敬请读者指正。

作 者

1992年8月

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

试读结束，需要全本PDF请购买 www.ertongbook.com

第一章 绪 论

§1.1 并行处理计算机

1.1.1 并行处理

随着电子管、晶体管和集成度日益提高的集成电路的出现，元器件技术的不断进步，计算机的速度不断提高，性能不断增强。但是，由于电子信号的最大传输速度是有限的，仅靠加速电子部件的速度来改善计算机的性能以满足用户对计算机愈来愈高的要求是不可能的。提高计算机性能的另一重要途径就是采用并行处理技术。

并行处理是一种有效的强调开发计算过程中并行事件的信息处理方式。它是一门综合性较强的学科，涉及到算法、语言、操作系统和系统结构等方面的知识，这些方面相互联系，互为条件，互为保证。

并行性有三种含义：一是同时性，指两个或多个事件在同一时刻发生在多个资源中；二是并发性，指两个或多个事件在同一时间间隔内发生在多个资源中；三是流水线，指两个或多个事件发生在可能重叠的时间段内。

由于并行性蕴涵着提高信息处理速度和解决大规模问题的巨大潜力，展现出由多个处理机组成更大的计算机系统以及由超大规模集成电路(VLSI)组成计算机的前景，因此对并行处理技术的研究在最近 20 多年一直是计算机科学技术领域内的重大课题之一。美国国防部尖端研究规划总署 1982 年初在纽约召开的一次会

议上明确地提出提高目前计算机系统性能的办法是“大量地并行”。尽可能地从一切方面引入并行处理技术是计算机科学技术及其应用技术发展的必然趋势。

1.1.2 并行处理计算机

并行处理计算机通常是指关联处理机，并行处理机，流水线处理机和多处理机。按照指令流和数据流的费林分类法，关联处理机、并行处理机、流水线处理机为单指令多数据流(SIMD)计算机，多处理机为多指令多数据流(MIMD)计算机。

商业化并行处理计算机越来越强调使用流水线操作和多处理机处理，这主要是由于它们的性能价格比高。

流水线计算机是指在中央处理器，输入输出处理器和存储器层次结构中具有数据重叠(时间)处理功能的那一类计算机。它有附属数组处理机如 IBM3838 和向量巨型机两种类型。通常附属数组处理机作为后端机附加到主计算机上，加强主计算机的浮点运算和向量处理能力。开发向量巨型计算机的主要目的是为了增加向量处理能力和提高向量处理速度。向量巨型机与附属数组处理机相比具有运算速度高，存储容量大，软件并行结构化，便于执行大规模的向量和矩阵计算等特点。第一代向量机产生于 60 年代末，如 CDC7600，其速度为每秒 5 兆次浮点运算。第二代向量机产生于 70 年代末，如 Cray-1，Cyber205，其速度分别为每秒 160 和 800 兆次浮点运算。现在向量巨型机趋于由多台向量机组成超级计算机，如美国 CRAY 公司正在研制的 Cray-3(最多有 16 台处理机)，它与 Cray-2 基本兼容。多流水线的向量处理机和多向量机是今后超级计算机主要发展方向之一。

现代并行处理系统——多处理机系统是当今计算机研究和开发的热门。近几年在这方面取得了重大成就。多处理机已开始从实验室研究走向批量生产，产品已进入市场。1986 年思维机器公司宣告连接机研制成功并投入运行，这个由 65536 台处理器构成的数据级并行处理计算机，每秒运算 10 亿次，问世后第一年就生

产了 16 台。另一方面，由多台至十几台，几十台以至上百台处理机组成的中小规模并行处理计算机也进入实用化、商品化阶段。

多处理机按照其拓扑结构分为开关连接结构和网络连接结构两类(图 1.1)，图左边是开关连接结构，图中结点是开关，端点是

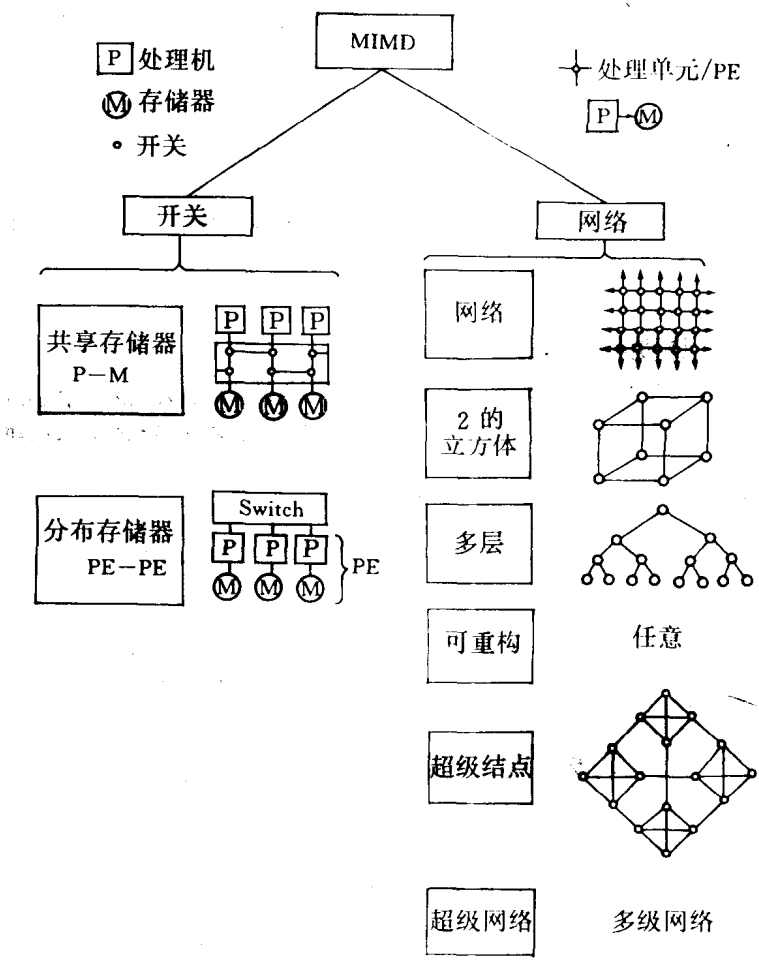


图 1.1　多处理机系统结构的分类

处理机或存储器。该类多处理机有两种。一是直接耦合(紧密耦合)的多处理机系统,系统中各处理机共享主存储器,通过公共变量进行机间通信,并受统一的操作系统管理,如 Fx/8。另一是非直接耦合(松散耦合)多处理机系统,系统中各处理机均有自己的主存储器(分布存储),各处理机由各自的操作系统管理,它们通过共享的输入输出系统进行通信。开关单元的结构是复杂的,它有总线结构,交叉开关结构和多端口存储器结构。图右边是网络连接结构,图中结点是包含主存储器的处理机,它有网格,立方体,多层,可重构,超级结点和超级网格等六种网络结构,是另一类松散耦合的多处理机系统。松散耦合的多处理机系统就是分布式计算机系统。开关和网络连接统称互连网络。

由工业化生产的高性能低价格微处理机组成的多处理机系统具有处理功能强,可靠性高,可维护性和可用性好,灵活性大,良好的性能价格比等特点,尤其是性能价格比特别吸引人。目前中小规模的多处理机系统已在市场上出售,其性能价格比,比超级小型机及大型机高出几倍甚至一个数量级。Intel 公司科学部预测九十年代超级计算机将是包括几百,几千甚至几万台微处理机构成的高度并行处理的机器,其性能将比现在业已研制成的超级计算机提高 1000 倍。

多处理机包括多向量机系统是今后并行处理计算机发展的主流,而软件和算法方面的突破是发展多处理机的关键。

§1.2 并行程序设计语言和模型

1.2.1 并行程序设计存在的问题

前面介绍了多种构造并行处理计算机的方法,它们对程序设计技术有不同的要求。在串行计算机中所有机器都可用 PASCAL,FORTRAN,COBOL 和 LISP 等高级语言编程。语言的高级属性掩盖了具体串行机的属性。在并行处理计算机上也能达到类

似的效果吗？

在很大程度上程序如同菜谱而处理器就是厨师。厨师熟悉的格言是："太多厨师会搅混了清汤"。而大多餐厅确实需要很多厨师，问题不是厨师太多，而是要求他们按预先为单一厨师制定的菜谱去做菜。在特殊情况下，菜谱或许会叫一个厨师作切100片葱这样简单的操作，这样100名厨师可以合作完成。但在通常情况下，必须根据要做的菜，生成一个新的并行菜谱，使得厨师之间互不冲突。对串行菜谱风格的简单模仿生成的还是串行菜谱，只有用一种新的语言、风格、表示法才能写成并行菜谱。这种新语言的重点不在于如何指导厨师去完成单独的任务，而是使他们相互通信和协调，充分发挥各自的效用。

1.2.2 传统串行语言的扩展

由于传统串行语言不满足通用并行设计的需要，最佳方法是扩展现有语言，使之能描述并行的问题。已经有了很多这方面的工作，如PASCAL的并行版本等，但许多并行语言都存在如下一些问题。

首先，扩展语言通常与特定的并行结构紧密联系，是按作者的需要产生的。就好像一个设计者对某种厨房较有经验，为它设计了菜谱语言，但它不能在设备不同的厨房中使用。

其次，许多扩展语言难以推广使用，所提出的概念是基于计算机的某一特定部件，而不是程序员所遇到的问题。如在烹饪例子中，使用扩展语言的程序只是指明将葱插入某个斜槽，然后按红色开关，而不是给出高级的，面向问题的、独立于机器的指令：切葱。

最后，扩展单一语言的观念会把我们束缚在某个基语言的缺陷上。更通用的方法是设计独立于表示的扩展集合，它可用于不同的基语言。

1.2.3 新的并行程序设计语言

现在已提出许多新语言，能够像传统语言一样被运用，但前提是在程序员未明确指明并行的情况下，编译程序能自动地从程序

中抽取可并行的部分。这种方法是可行的，因为某些传统语言的特征(如副作用)已被排除，使得语言易于编译。

这个方法的问题在于原始应用问题的并行性，未必在最终的程序中表示出来。“好”的串行程序员能够写出在单处理机上快速执行的程序，但这种程序无内在并行性。这种在串行机上提高算法和程序速度的技巧和技术无法在并行机上使用。

例如，一个程序要排除数组样本中的干扰，办法是将数组元素的值，用它本身与邻近元素的平均值来替代。这个操作当然可用并行算法：每个元素将它本身邻近元素加起来，除以相应元素的个数。但串行程序未必这样做，而是在算法开始时计算一次所有元素的和，以后的计算，可用增加一个新值再减去一个旧值的方法得到，就像一个掩模沿着数组滑动一样。这个算法比用并行算法在串行机上仿真速度快，因为加法次数减少了。但在串行算法中找不出一点并行性，不能利用并行硬件。所以，虽然串行语言与人们所熟悉的风格相符，但与通用并行硬件不匹配，使得程序不能充分利用并行硬件。

1.2.4 独立于结构的并行程序设计模型

本书重点介绍了一种独立于结构的并行程序设计模型，即“并行关系模型”。它的主要目的之一是解决通用性的问题。程序员可描述任何类型的应用问题，而不必了解程序将在何种并行处理计算机上运行。该模型必须提供工具来描述程序员的问题。另一个方面，基本模型的语言，要便于用透明、有效的方式来编译，以适用于各种并行处理计算机结构。当然其效率是重要的，效率是提高速度不可缺少的因素，而提高速度是编制并行程序的动力。

试读结束，需要全本PDF请购买 www.ertongbook.com