复旦大学
张丽君
陈增荣
高坤敏
编著

电子计算机

并行算法的设计与分析

电子计算机
并行算法的设计与分析

复 旦 大 学
张丽君 陈增荣 高坤敏编著

湖南科学技术出版社

内 容 提 要

现代科学技术的发展向计算机科学提出了更高的要求，要求在尽可能短的时间内能处理尽可能多的问题。故从70年代开始出现并行处理机，而一门崭新的学科——并行算法亦随之问世。

本书以最基础、最常用、而又最具代表性的线代数问题为例，介绍了并行算法的一般特征及其设计基础、设计方法。全书共分四章。第一章介绍并行算法设计基础，主要讲述大、中、小型计算机及并行处理机模型，计算复杂性及浮点运算的舍入误差分析。第二章重点介绍基本算法。其中包括矩阵乘法、递推计算、多项式与算术表达式的并行计算及快速富里叶变换的并行处理。第三、四章就线代数方程组与特征值问题作专题探讨。是我国第一本阐述计算机并行计算的专著。

电 子 计 算 机

并行算法的设计与分析

复 旦 大 学

张丽君　陈增荣　高坤敏编著

责任编辑：周翰宗

*

湖南科学技术出版社出版

（长沙市展览馆路14号）

湖南省新华书店发行　湖南省新华印刷二厂印刷

*

1984年1月第1版第1次印刷

开本：787×1092毫米 1/32　印张：6.125　字数：137,000

印数：1——7,400

统一书号：15204·114　**定价：1.00元**

前　言

许多大型科学计算要求计算机具有尽可能高的速度和尽可能大的容量。当代大面积集成电路技术的迅速发展，为研制这类计算机创造了良好的条件。然而从速度上看，目前已接近物理极限，要从电路技术上进一步大幅度提高看来已不可能，所以必须从计算机的系统结构着手。于是从七十年代开始相继出现了许多并行处理机。

这类计算机有别于串行计算机的是能提供很强的并行处理能力。问题是传统的数值计算方法的不适应性，也许它会使这类计算机的效率大大降低。这就要求我们研究新的并行数值计算方法，并提高新算法在这类计算机上的运行效率。这是计算数学工作者面临的迫切任务。鉴于在科学计算上以后将主要使用这类计算机，我们必须作好准备，研究一切有关的问题。不仅要研究高度并行性的算法，还要对各种算法作出评价，并进行误差分析。正因为如此，国际上一门崭新的学科——并行算法（Parallel Algorithm，或叫平行算法）开始形成。数值算法从此不仅要区分为代数的和分析的，有限的和无限的，精确的和近似的，而且要区分为串行的和并行的。

目前我国已研制成并行处理机。757机已交付使用，785机亦已问世。所以研究并行算法不仅具有理论意义，也具有巨大的现实意义。本书拟介绍这门学科的概况，供从事计算数学和计算机软件的同志们参考。

各类计算问题都需要研制并行算法，但限于篇幅，本书只能选择有代表性的最基础而又常用的线代数问题为例，介绍并行算法的一般特征。为完备起见，对各种线代数问题，我们都给出了主要的并行求解算法。

书的内容力求通俗易懂。对涉及到的较难的基础知识都以浅显的语言作了简介。全书共分四章。第一章是基础，介绍并行算法设计中的四个重要问题：对硬件的依赖性，算法的计算复杂性，浮点运算的舍入误差分析，以及各组运算之间的数据依赖关系。第二章讨论基本算法，包括矩阵乘

法、线性递归计算、多项式和表达式求值以及FFT。这些算法在科学计算的各个领域都很有用，所以专列一章介绍。第三、四章讨论线代数问题的并行算法，包括各种线性方程组的典型求解方法以及特征值问题。

全书承国防科大陈火旺教授细心审阅，在此谨致谢忱。由于我们的水平有限，书中可能存在某些错误和缺点，欢迎批评指正。

编者　一九八一年五月

目　录

第一章　并行算法设计基础

算法的设计与分析涉及的基础知识有三个方面：计算机模型、计算复杂性以及算法的舍入误差分析。如果不考虑问题的数学求解，可以认为这是算法设计的三个要素。首先算法的设计与计算机的特征有关，不同类型计算机上的最优算法一般也不相同。其次，算法的计算复杂性是评价算法的最重要标志之一，计算复杂性很差的算法不能认为是好的算法。最后，算法须在计算机上运行。由于计算机的字长总是有限的，必然会遇到舍入误差问题。如果一个算法对舍入误差十分敏感，那么这一算法是毫无价值的。只有对舍入误差稳定的算法才是有用的算法。

本章前三节分别介绍这三个方面最基础的内容。第四节简略介绍并行算法的设计与分析。

§1　各类计算机的主要特征

我们研究的对象是各类科学计算的算法，而算法的优劣与计算机的结构特征密切相关。在某种计算机上运行得很好的算法，在另一类计算机上也许很差。所以我们必须讨论计算机的结构特征对算法的影响。对此国际上召开过专题会议，这就说明了问题的重要性。当然，这里的结构特征有别于计算机的系统结构。因为系统结构关心的是计算机各部件协调的细节，而我们并不关心这些细节。我们给出的特征模型是相当简单、相

当粗糙的。它离计算机的系统结构还有很大距离。

本节主要介绍中、小型计算机，大型计算机，巨型计算机的特征模型。重点放在最后一类。

1.1 中小型计算机的模型

自四十年代初期至六十年代初设计的计算机，以及当前流行的单微处理机都是规模较小的计算机。从系统结构上看，这些机器称为第一、二代计算机（早期对计算机曾以电子管、晶体管、集成电路和大规模集成电路来划分，但现在更倾向于按结构来划分）。它们的运算器比较简单，主要核心是加法器。它们的特征模型如图1.1所示。

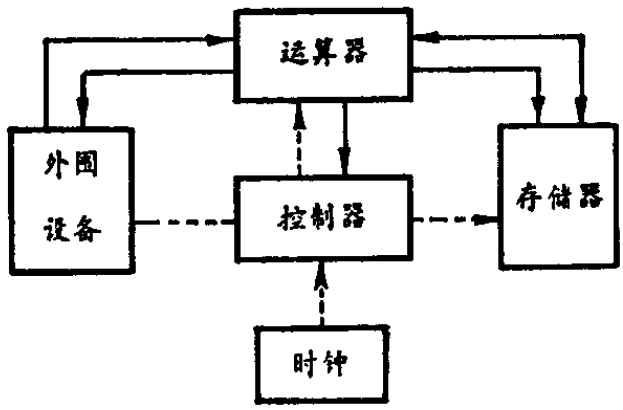

图1.1 中小型计算机模型

图中的时钟隔一定的周期发出一个脉冲。这个周期称为计算机的时钟周期。它是计算机运行的最基本的时间单位。存储器分存储介质部分和存储控制部分。存储介质部分用来存储数据，包括数和指令。存储控制部分接受控制器送来的信号，管理存数（即写数）和取数（即读数）。每存储一个数据或取出一个数据，都需占用存储器一个存取周期时间。一个存取周期通常等于8个时钟周期，也有计算机采用4个或 6 个时钟周期为一个存取周期的。

运算器接受控制器送来的命令，对存储器送来的数据进行加工。存储在存储器中的指令通过运算器传送至控制器，使控制器发出执行该指令的有关命令。运算器主要由三、四个寄存器和加法器组成。寄存器存放存储器送来的数据或加工好的结

果。通常取指令需一个存取周期，一对数相加需一个存取周期。如果需要变址或间接访问，还要增加一个存取周期。乘法和除法也是在加法器上实现的，方法是多次相加或相减。这就要耗费较长时间。除一般运算都需要的若干个存取周期外，按乘除法的实现方案，另要加上t或$t/2$个时钟周期。这里t是尾数长度。

外围设备包括输入、输出、外存储器等设备，以及对这些设备的控制部分。这一部分与算法的关系不大。只有当问题规模很大、外存储器作为第二存储时才与某些算法有关。

控制器是整个计算机的核心。它负责各个部件之间的协调，给这些部件发出命令。

在图1.1中实线表示有关部件之间数据的流向，虚线则指出信号的流向。

从中小型计算机的特征模型可以看到，对这类计算机，加法运算所需的时间比乘法和除法少得多。通常对中型计算机$t_{*}\doteq t_{+}\doteq 3t_{+}$。对小型计算机约为$t_{*}=t_{+}\doteq 5t_{+}$。这里$t_{+}$、$t_{*}$、$t_{+}$分别表示加减法、乘法和除法所需的时间。传统的计算方法主要考虑乘除法的运算次数，因此主要适用于这类计算机。

1.2 大型计算机的模型

从六十年代中叶起，出现了大型计算机，也有人称为第三代计算机。它的特征模型如图1.2所示。

就算法而言，大型计算机与中小型计算机的主要区别有三。其一是运算器中增加了乘法部件，所需的元件个数比加法部件要多得多。由于有了高速乘法部件，乘法的执行时间比加减法只多几个时钟周期。然而因没有除法部件，除法的执行时间仍很长。在有倒数部件的计算机里，除法的算法如下：

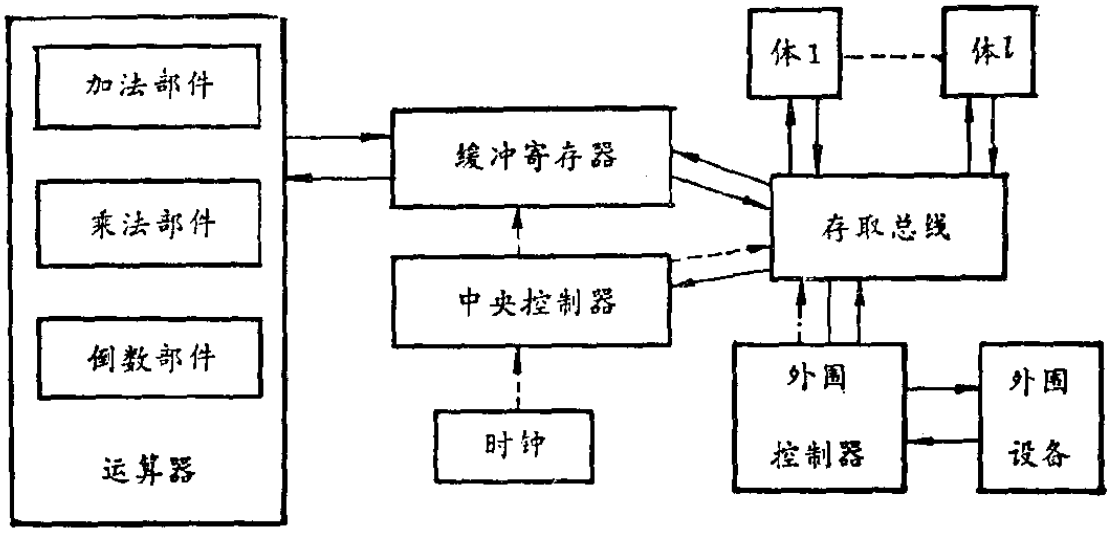

图1.2 大型计算机的模型

1.求出倒数近似值；

2.利用牛顿法对倒数进行迭代

$$y=f(x)=1/x; \qquad y^{(n+1)}=y^{(n)}(2-y^{(n)}\cdot x)$$

3.乘上分子得到商。

对半精度的倒数部件，牛顿迭代只需进行一次。若倒数部件的精度较差，迭代次数还需增加。如果没有倒数部件，执行除法所需的时间更长。通常除法是加法所需时间的五至十倍。

其二是主存储器的设计。中小型计算机里的存储器在这里相当于一个体。现在的主存储器由多个体组成。由于不同的体可以同时存取（在硬件中称为体的交错访问），总的传输速率大大提高。

区别之三是高速缓冲器的使用。加工的数据如能在高速缓冲区中取到，则运行时间亦大大缩短。

当然还有其它差别。如运算时间不再是若干个存取周期，只要几个时钟周期就行了。这使加法和乘法的执行时间大为缩短。

从上可以看到，在大型计算机上的算法，不再关心乘除法的总次数。它关心的应是除法的次数。

对于存储器，一般说来，它的传输速率跟不上运算的速度。因此要充分发挥各个体的效率。由于同一个体在一个存取周期内不能存取两次，所以我们希望数据的访问能均匀地遍及各个体上。这就存在一个数据在存储器内的排列问题。对间接访问等地址随机的一批数据，这个问题无法解决。我们关心的是特殊类型的数据——线性向量。所谓线性向量就是任意两个连续分量的地址差是常数的向量。对这类数据，只要地址差与体的个数互质，数据就均匀地分散到各个体上。从这个要求可以看出，体的个数应该是质数。只有这样，才能使尽可能多的线性向量达到上述要求。由于大型计算机中体的个数较少，这个问题尚不突出。算法主要应考虑尽可能减少对存储器的平均访问次数。

1.3 并行处理机的模型

七十年代初开始出现巨型机，现在称为并行处理机。也有人称之为第四代计算机。到目前为止，并行处理机已有多种。从系统结构来看，大致可以区分为阵列式、向量流水线和多处理机这三种类型（尽管目前研究分布式计算机的人很多，但对它的算法研究尚处于探索阶段）。

一、阵列式模型 最早出现的并行处理机 IlliacIV 就是阵列式的。64 台处理机构成 8×8 阵列。每台处理机都有自己的存储器，并可以直接访问上下左右邻接处理机的存储器。因为数据从任一处理机传送到另一处理机，要通过不少中间处理机，所以要花费较长时间。可以证明，对矩阵计算，数据传送时间将超过算术运算所花费的时间。而且为缩短算法的运行时

间，数据的组织、调度都将遇到很多麻烦。为避免这些问题，D.J.Kuck研制了BSP（结构如图1.3所示），16台处理机通过十

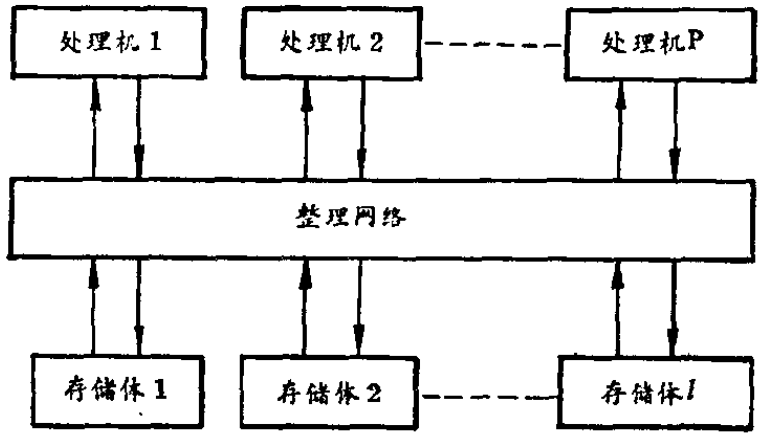

图1.3 BSP的结构

字交叉的整理网络（alignment network）可以直接与17个存储体相连。这样就不存在处理机之间的数据传送问题。

整理网络是并行处理中很重要的一个课题。处理机之间完全传送的整理网络所需元件的复杂性阶数非常高。当处理机台数增加时，元件数量急剧增加。当处理机台数超过15时，完全传送实际上已成为不可能。

为了增加并行运行的处理机台数，我们只能限制处理机可以直接访问的范围。但我们希望有较好的整理网络，使数据传送时间和整理网络所需元件数量之间有好的折衷。现在已经提出了不少好的整理网络方案。因与此有关的并行算法尚需探索，故本书对整理网络不作介绍。

我们采用图1.3作为阵列式的模型。这里每台处理机相当于上节中的大型计算机。它们同时执行中央处理机送来的指令。由于对若干数据执行同一指令，这类计算机属于单指令流多数据流类型（简记为SIMD）。

目前文献中常用的阵列式模型均作如下假定：

1.每台处理机的运算执行时间相同，且与处理机的台数无关；

2.每台处理机对存取信息不加限制；

3.处理机的台数不受限制。

在这些假定中2、3两条与实际的差距很大，且两者不能兼顾，即处理机台数的增加必然大大限制数据访问的能力。

二、流水线模型 大多数第三代计算机已使用流水线技术。并行处理机在这点上与第三代计算机的差别在于将流水线技术用于向量计算。由于不是按指令流设计流水线，避免了流水线中断，提高了效率。但这要求硬件提供费用昂贵的向量寄存器。

流水线技术与工厂里流水线生产的基本思想是一样的。它将复杂的运算分解为基本的子运算。运算对象逐个（或逐对）经过各子运算站，最后完成复杂的运算。由于各子运算站可以同时工作，所以运算速度大大提高。如浮点加法可分解为：求阶差；尾数按阶差移位，即对阶；按数符改变尾数（因减法用加法器实现，所以当两个异号数相加或两个同号数相减时需将负数的尾数改为反码或补码）；尾数相加；按双符号位处理尾数（若为负数将结果转换为原码）；规格化和舍入等六个子运算。相应的流水线就有六个子运算站。当第一对数在第一站加工完毕送入第二站时，第一站又可加工第二对数。当第一对数到达第六站时，第一至五站分别是第六至二对操作数。一般流水线还要加上取操作数和送出结果这两个站。

若某流水线有l站，则它的结构如图1.4所示。设子运算站i加工一对数所需的时间为t_i，则$t_{max}=\max_i t_i$称为流水线的周期。大多数计算机的流水线周期等于时钟周期。第一对操作数从送入流水线到完成为止共需$l*t_{max}$时间。以后每隔t_{max}时间完

成一对操作数的加工。n对操作数的加工共需时间

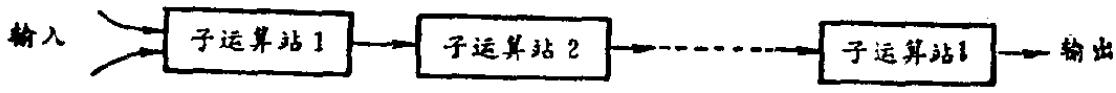

图1.4 流水线的结构

$$T_1=[(l-1)+n]t_{max}=t_0+n*t_{max}①$$

其中$(l-1)t_{max}$称为流水线的起步时间②，以后简记为t_0或σ_0。

流水线计算机速度的提高受到流水线站数的限制。为了进一步提高速度，不少计算机采用多重流水线和向量链接技术。若有p条流水线，则加工n对数所需的时间为

$$I_p=t_0+\lceil n/p\rceil*t_{max}③$$

不同运算的向量流水线满足一定条件时可以链接。如：x*y +z，这里x*y是两个向量的分量乘，即为

$$(x_1y_1,\ x_2y_2,\ \cdots,\ x_ny_n)$$

类似地，$x\div y$表示向量的分量除法。这些向量运算记号现已广泛采用。x*y +z使用的两条流水线是 A*B 和A +B，它们可以如下链接（见图1.5）：当x和y的第一对分量x_1、y_1之积从部件A*B流出，部件A +B就可以开始工作。于是 x*y +z 的执行时间为

$$T_l=t_0^*+t_0^++(n+1)\tilde{t}_{max},\quad \tilde{t}_{max}=\max(t_{max}^*,t_{max}^+)$$

式中右上角的 * 和 + 标记该量是流水部件 A*B和A +B 的有关量。可以看到，向量链接技术使执行时间大大减少。

通常的链接条件有三：流水线部件要有空；一个操作数向

注① t_{max}今后记为τ，有时为方便起见可取为1。

注② 若从指令流出算起，起步时间尚需再增加2τ。但后面的链接只需考虑流水线本身。

注③ x表示不小于x的最小整数。

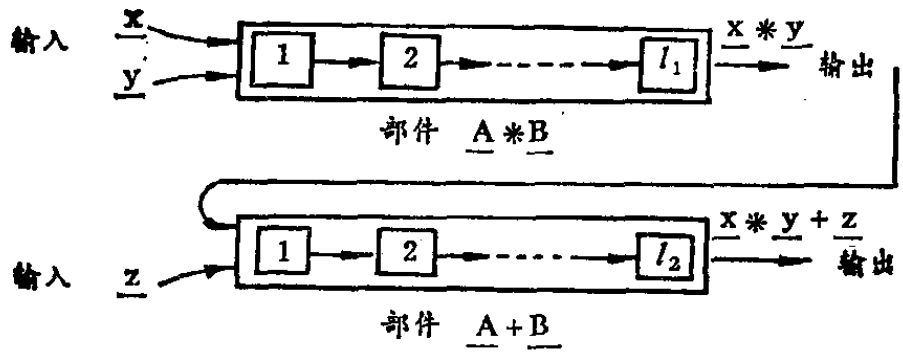

图1.5 向量的链接

量事先已准备好；操作数向量寄存器要有空。这三条缺一不可。如 x + y + z 在只有一个加法部件的计算机里不能链接，因为加法部件没有空。又如$z_i = x_i y_i + z_{i-1}$, $i = 1, 2, \cdots n$; 由于z_i依赖于z_{i-1}，操作数向量事先未准备好，也不能链接。又如x*y + x，因x不能同时用于两条不同的流水线而不能链接。

在有向量链接设备的计算机中除法可如下执行(参见1.2中的除法算法)：

$1/V_2 \Rightarrow V_4$　　半精度倒数近似值

$V_1 * V_4 \Rightarrow V_5$　　半精度商

$2 - V_2 * V_4 \Rightarrow V_6$　牛顿迭代的校正因子

$V_5 * V_6 \Rightarrow V_3$　　得全精度商$V_3 = V_1/V_2$

这里只使用一次链接技术，即倒数部件与乘法部件链接。第三条指令中的减法是用求补码的办法实现的。

从上面的分析可以看到，对流水线模型，计算时间与运算次数之间有着线性依赖关系，或者说两者的时间复杂性是同阶的。这一点使流水线模型的算法和阵列式理论模型的算法有本质上的差别。流水线计算机不希望算法运算次数的复杂性阶增加，而阵列式理论模型则不然。尽管两者有差别，但都是针对向量运算设计的，所以均称为向量计算机。

三、多处理机模型　多处理机系统由多台处理机联合而