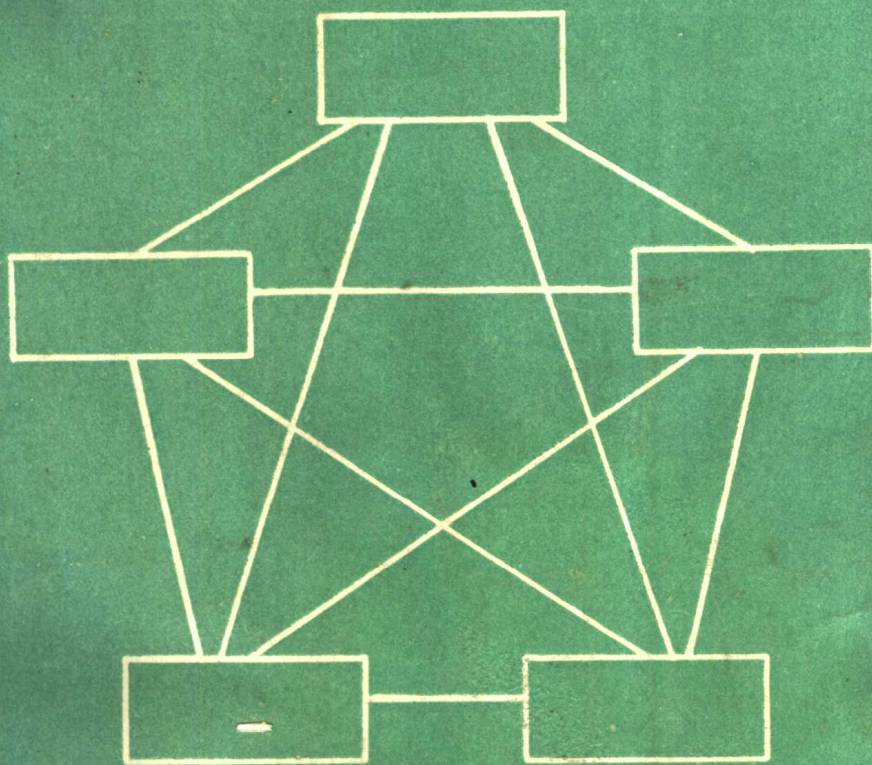


并行计算 与偏微分方程数值解

李晓梅 任兵 宋君强 编著



国防科技大学出版社

并行计算 与偏微分方程数值解

李晓梅 任 兵 宋君强 编著

国防科技大学出版社

内 容 简 介

本书综述了国内外巨型计算机的类型和特点；深入系统地介绍了并行算法的设计思想和发展；从实用角度出发，系统地介绍了偏微分方程离散化的基本原理及其并行计算的实现方法；提供了若干大型科学计算的并行处理实例。

本书选材新颖，结构严谨，把巨型计算机科学、偏微分方程的数值解法及并行算法三门重要学科有机地融为一体，既可作为大学高年级学生和研究生的一门教材，又可供从事计算数学、计算机研制和科学计算的人员参考。

并行计算与偏微分方程数值解

李晓梅 任 兵 宋君强 编著

责任编辑 徐后华 钟 平

*

国防科技大学出版社 出版发行

湖南省新华书店经销

国防科技大学印刷厂印装

*

开本：787×1092 1/16 印张：24¹/₁₆ 字数：550千字

1990年9月第1版第1次印刷 印数：1—1000

ISBN 7-81024-112-5
TP·21 定价：4.85元

序

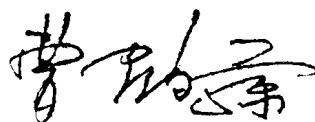
随着科学技术的迅猛发展，提出的计算问题愈来愈复杂，计算量和精度要求愈来愈高，大型和巨型计算机的问世，初步解决了这个问题。并行计算的成功实施，使大型和巨型机的计算速度得以倍增，大大提高了计算机的作用，这才真正发挥了计算机“大”和“巨”的特点。

把并行计算和求解偏微分方程结合起来，是本书的特色。求解任何科学计算问题，首先是把问题的主要因素提炼出来，建立起一个最能体现实际的数学模型——偏微分方程或偏微分方程组，建立起一个正确的微分方程或方程组，是获得问题正确解的一个关键。微分方程或方程组离散后，应用并行计算方法，又快又好地解决数学模型，是获得问题正确解的又一关键。

本书说理清楚，举例易懂，是一本很好的教材，适用大学高年级本科生或研究生，也是对工程技术人员或科学工作者的一本很好的参考书籍。

爰为序。

国际宇航科学院院士



1990.3.17于国防科技大学

前 言

许多科学与工程的问题,都可以用线性或非线性偏微分方程来描述。这些偏微分方程的数值求解,往往是大型或超大型计算课题,要求计算机具有极快的运算速度和极大的信息吞吐量。尽管从1944年到1964年的二十年间,计算机的运算速度提高了一百万倍,但离满足大型或超大型科学计算的需要还很远。在过去二十多年里,并行和向量流水线计算机的产生和发展,为解决大型和超大型科学计算问题提供了广阔的前景。但由于并行和向量流水线计算机的系统结构不同于传统计算机,要充分发挥并行机的效率,必须研究与之相适应的并行计算方法。

并行算法是当前国际上的热门课题,国内亦已连续召开了两届全国并行算法学术会议。有关并行处理、并行算法和偏微分方程数值方法的文献很多,本书则是将并行计算与偏微分方程数值方法有机地结合在一起的一次尝试。编写本书的目的,是介绍并行计算的有关知识,充分挖掘各种偏微分方程数值方法的内在并行性。

本书是在原有讲义《向量机上偏微分方程数值解》的基础上编写而成的,参阅了大量国内外新近发表和出版的有关文献著作,并融合了作者近几年的科研成果。

为了使本书面向更多的读者,真正起到普及和推广并行计算的作用,我们在编写本书时,尽量避免抽象的数学理论分析。对于阅读本书的读者,仅要求有程序设计、计算机结构、线性代数及数学物理方程等方面的一般基础知识。设计有效的并行算法,必须熟悉并行计算机的结构特点,因此,本书用了一定篇幅介绍各种并行机的结构特点。由于偏微分方程离散化以后,主要的计算部分是解线性代数方程组,为此,本书重点介绍了离散方程的并行计算。

全书共分九章。

第一章论述并行计算机和大型科学计算,介绍了计算机的发展和计算机性能提高的主要手段;同时也介绍了并行算法发展的几个阶段,论述了研究并行算法的重要性。第二章在介绍串行计算机和向量流水线机的基础上,介绍了并行处理机、多处理机和多计算机模型。第三章论述并行算法的设计方法,介绍并行算法的有关概念和设计并行算法应注意的几个问题。

从第四章开始介绍偏微分方程的数值方法和并行计算方法。第四章作为后面几章的导论,介绍离散化方法——有限的数值模型的提出、建立、分类及特性等基本问题。第五、六、七章分别介绍双曲、抛物和椭圆型方程的数值离散化方法及有关的并行处理。第八章专门介绍离散方程的并行计算,既研究了一些典型算法的内在并行性,又介绍了近几年才出现的适合并行计算的新算法。由于篇幅所限,我们着重研究方法的内在并行性,没有详细讨论方法在并行和向量机上的具体实现。第九章列举了在不同并行或向量机上求解一些典型问题的计算实例。

本书第一、二、三章由李晓梅教授编写；第四、五、六章由任兵副教授编写；第七、八章由宋君强讲师编写；第九章由李晓梅、宋君强共同编写。

我们在编写过程中，得到了徐后华教授和吴其芬副教授的帮助，他们仔细审阅了全稿，提出了许多宝贵意见和建议。另外，在即将排版付印之际，国际宇航科学院院士曹鹤荪教授在百忙之中抽出时间审阅本书部分手稿，并为本书作序。在此一并表示衷心感谢。

编写本书时，尽管我们结合多年积累的教学科研经验，尽可能在书中反映最新的研究成果，但是，限于水平，错误和缺点在所难免，欢迎读者批评指正。

编著者

一九九〇年三月

目 录

1 并行计算机与大型科学计算

- 1.1 大型科学计算对超级计算机运算速度的需求 (1)
- 1.2 计算机达到高性能所使用的方法 (4)
- 1.3 计算机的发展 (10)
- 1.4 计算机系统分类 (14)
- 1.5 并行算法发展的几个阶段及研究并行算法的重要性 (16)
 - 1.5.1 并行算法发展的几个阶段 (16)
 - 1.5.2 研究并行算法的重要性 (20)

参考文献

2 并行计算机模型

- 2.1 串行计算机模型 (25)
 - 2.1.1 随机访问机RAM (25)
 - 2.1.2 阶的表示 (27)
- 2.2 向量计算机模型 (28)
 - 2.2.1 YH-1 向量处理机 (28)
 - 2.2.2 CDC CYBER-205向量处理机 (33)
- 2.3 并行处理机模型 (36)
 - 2.3.1 SIMD互连网络 (36)
 - 2.3.2 阵列处理机 (38)
 - 2.3.3 关联处理机 (44)
- 2.4 多处理机和多计算机模型 (44)
 - 2.4.1 多处理机的一般原理 (44)
 - 2.4.2 多处理机的分类 (45)
 - 2.4.3 二十种多处理机和多计算机简介 (46)

参考文献

3 并行算法的设计方法

- 3.1 并行算法的有关概念 (58)
- 3.2 设计并行算法应注意的几个问题 (60)
- 3.3 阵列处理机并行算法设计 (61)
- 3.4 MIMD计算机上算法设计 (65)
 - 3.4.1 MIMD计算机上的算法分类 (66)
 - 3.4.2 极限加速比 (68)
 - 3.4.3 MIMD计算机算法复杂性分析 (70)
- 3.5 MIMD计算机上的进程通信和同步化 (71)
- 3.6 死锁 (76)
- 3.7 多处理机上的任务调度 (77)

参考文献

4 离散化方法——有限的数值模型

- 4.1 偏微分方程的特征理论 (82)

4.1.1	主特征	(83)
4.1.2	次特征	(85)
4.1.3	影响的形式	(88)
4.2	模型方程	(90)
4.3	过程的稳定性和定解条件的恰当性	(97)
4.4	数值方法的本质	(99)
4.4.1	任务	(99)
4.4.2	方法	(100)
4.4.3	推导离散化方程的常用方法	(101)
4.5	有限差分近似及其数学性质	(107)
4.5.1	有限差分方法的基本问题	(107)
4.5.2	离散近似与等价方程	(107)
4.5.3	截断误差与差分近似的相容性	(112)
4.5.4	离散化误差与差分解的收敛性	(113)
4.5.5	舍入误差与差分解的稳定性	(116)
4.5.6	差分余项效应及其修正	(124)

参考文献

5 双曲型方程

5.1	守恒型和特征型方程组	(131)
5.2	$\partial u/\partial t + a(\partial u/\partial x) = 0$ 的差分格式	(133)
5.2.1	迎风格式与 CFL 条件	(134)
5.2.2	利用特征线构造差分格式	(137)
5.2.3	MacCormack 显式格式 (1969)	(141)
5.2.4	Warming-Beam 格式 (1975)	(142)
5.2.5	隐式格式	(143)
5.3	非线性方程 $\partial u/\partial t + \partial F(u)/\partial x = 0$ 的常用差分格式	(146)
5.3.1	一阶显式格式	(146)
5.3.2	二阶显式格式	(147)
5.3.3	隐式格式	(150)
5.3.4	高分辨率差分格式简介	(152)
5.4	双曲型方程组 $\partial U/\partial t + \partial F(U)/\partial x = 0$ 的差分格式	(160)
5.4.1	守恒型差分格式	(160)
5.4.2	特征型差分格式	(163)
5.4.3	矢通量分裂算法	(164)
5.5	多维初值问题的分步方法和隐式因子分解法	(168)
5.5.1	数学原理与基本思想	(168)
5.5.2	分步差分格式	(171)
5.5.3	隐式因子分解法	(173)
5.6	双曲型方程 (组) 的初边值问题	(175)

参考文献

6 抛物型方程

6.1	初边值问题	(180)
6.2	常系数扩散方程 $\partial w / \partial t = a(\partial^2 u / \partial x^2)$ 的差分格式	(181)
6.2.1	显式格式	(181)
6.2.2	隐式格式	(184)
6.2.3	二步格式	(186)
6.2.4	差分格式的统一式	(187)
6.2.5	无条件稳定的组显式格式及其并行计算	(189)
6.3	二维问题及其并行计算	(197)
6.3.1	一维格式的直接推广	(197)
6.3.2	二维组显式格式	(200)
6.3.3	交替方向隐式格式 (ADI)	(206)
6.3.4	局部一维格式	(208)
6.3.5	预测-校正格式	(209)
6.3.6	跳点格式	(209)
6.4	非线性问题	(210)
6.4.1	MacCormack显式格式	(211)
6.4.2	隐式格式	(215)
6.4.3	线性化方法	(216)
6.5	可压缩流体的Navier-Stokes方程组的时间分裂算法	(217)
6.5.1	控制方程组	(217)
6.5.2	MacCormack显式时间分裂算法	(218)
6.5.3	Beam-Warming隐式格式	(224)

参考文献

7 椭圆型方程

7.1	引言	(229)
7.2	几种典型的椭圆型方程	(230)
7.2.1	几种典型的椭圆型偏微分方程	(230)
7.2.2	边界条件	(230)
7.2.3	椭圆型方程描述的一些实际问题	(231)
7.3	差分方法	(232)
7.3.1	平面问题的五点差分格式	(232)
7.3.2	边界条件的近似处理	(236)
7.3.3	差分方程的性质	(238)
7.3.4	差分问题的矩阵表示	(242)
7.3.5	高阶精度的差分格式	(245)
7.3.6	重Poisson方程的差分格式	(246)
7.4	变分原理	(248)
7.4.1	泛函分析基础	(248)
7.4.2	变分原理	(250)
7.4.3	Rayleigh-Ritz法与Galerkin法	(259)
7.5	有限元法	(261)
7.5.1	有限元法的一个实例	(262)

7.5.2	单元的形状函数	(269)
7.5.3	二维问题的矩形元	(270)
7.5.4	二维问题的三角形单元	(275)
7.5.5	二维问题的等参数单元	(281)
7.6	边界元法	(282)
7.6.1	边界积分方程	(284)
7.6.2	边界元公式	286)

参考文献

8 离散方程的并行计算

8.1	引言	(291)
8.2	直接法	(292)
8.2.1	稠密矩阵的Gauss消去法	(292)
8.2.2	Givens变换法	(296)
8.2.3	QIF算法	(298)
8.2.4	对称、正定、带状方程组的并行计算——算法1	(301)
8.2.5	对称、正定、带状方程组的并行计算——算法2	(306)
8.2.6	稀疏矩阵的排序方法	(310)
8.2.7	子结构法	(314)
8.2.8	三对角方程组	(315)
8.2.9	快速Poisson解法	(322)
8.3	迭代法	(325)
8.3.1	基本迭代法	(325)
8.3.2	Gauss Seidel法和SOR法的并行和向量计算	(332)
8.3.3	SSOR法的并行和向量计算	(340)
8.3.4	ADI法	(341)
8.3.5	多项式加速法	(342)
8.3.6	梯度法	(345)
8.3.7	预处理共轭梯度法及其并行和向量计算	(347)

参考文献

9 大型科学计算的并行计算实例

9.1	二维张量程序在YH-1机上实现	(354)
9.1.1	基本方程	(354)
9.1.2	定解条件	(359)
9.1.3	差分方程	(359)
9.1.4	计算网格	(365)
9.1.5	二维张量程序的并行处理	(366)
9.2	二维倾角校正CRAY X-MP/48上实现	(371)
9.3	中期数值天气预报谱模式的多任务化	(379)
9.3.1	控制方程	(379)
9.3.2	方程的解法——谱方法	(381)
9.3.3	谱模式的多任务化	(385)

参考文献

1 并行计算机与大型科学计算

并行计算机是一种精度高、速度快、容量大、吞吐率高、软件功能强的计算机。由于它的发展,极大地推动着科学和工程发展,而现代科学技术的发展,反过来又提出了许多大型和超大型的计算课题,如:数值天气预报、石油地震数据处理、核武器设计以及航空航天技术等。这些课题都要求有极高运算速度的计算机予以计算。并行计算机正是用来计算这些在一般计算机上难以计算、甚至不能计算的大型和超大型的科学计算和工程设计问题。

近廿年来,向量和并行计算机发展非常迅速,并逐步投放市场。我们知道,第一台阵列处理机ILLIAC IV是1975年投入使用,流水线处理机CRAY-1也是在1976年投放市场。由于超大规模集成电路(Very Large-Scale Integration)的出现,开辟了计算机的新时代,这不仅使个人计算机使用成为可能,而且对于那些由几十台、几百台、甚至上千台处理机所组成的多处理机系统,由它们共同协作来完成一个问题的计算也成为可能。这使人们看到:许多复杂过程的实时动态模拟已逐渐变成现实。从传统的诺依曼式的串行计算机发展到并行计算机,已被人们视为当代计算机科学最有意义的发展。

并行计算是一个相对年青的领域,它的产生和发展也不过是卅多年的历史,它是随着向量和并行计算机的广泛使用而发展起来的一门新学科。研究并行计算虽然是一种新的训练,但它却是非常重要的。许多在传统计算机上运行得很好的程序却不能很容易地转变成在并行机上有效运行的程序。相反,那些在串行机上并不有效的算法却常常隐含着并行性的特征,这些并行性特征吸引着人们将其编成并行程序。

当今,我们正进入并行计算机年代,许多应用领域的问题都要求计算机的运算能力比今天最快的计算机还要高出几个数量级。并行性是达到这种突破性的最好方式。现在西方各国正在发展具有更强能力的并行计算机,它们中有些价格是比较昂贵的。如计算能力超过100MFLOPS(每秒一亿浮点操作)的超级计算机,其价格是在500万美元到2500万美元之间。对于这些超级计算机来说,虽然在理论上可达到这样高的速度,但并不是说,程序均能以这样高速持续运行,这与应用问题及软件关系极大。

本章1.1节列举若干个例子说明大型科学计算对超级计算机运算速度的需求;1.2节介绍过去三十五年间计算机性能提高的主要原因,同时研究了流水线和并行性间差异;1.3节介绍计算机发展;1.4节介绍计算机系统的分类;最后1.5节介绍并行算法发展的几个阶段和研究并行算法重要性。

1.1 大型科学计算对超级计算机运算速度的需求

超级计算机也称为巨型机。它是一种在一定时期内可供使用的最快和最有效的科学

计算机系统。也就是说在一定时期内处理能力最强的一类计算机。而大型科学计算是应用超级计算机并通过适当地运用数值算法和技巧去求解或模拟科学和工程问题。今天超级计算机在许多科学、技术和工程应用中起着关键性的作用，社会需求对超级计算机性能的要求越来越高。很多领域的大型科学计算问题，如大气研究、天体物理学、分子生物学、计算空气动力学、聚合研究、核武器设计、密码分析、数值天气预报等。这些问题用现有的超级计算机无法解决，如果不尽早满足这些要求，将有碍于现代科学技术的发展。超大规模集成电路的成就和大型科学计算问题的挑战正使超级计算机的结构发生根本性变化。提高超级计算机性能的主要潜力在于大规模地利用并行性。但是高效地实现大规模的并行性是非常困难的，这是因为在今天的技术水平下，对通用计算机来说，大规模的并行性往往伴随着使人难以接受的复杂性和开销。因此，未来的超级计算机设计者所面临的问题是：如何提高并行性而又不增大或不显著增大复杂性，这是一个十分复杂而又十分关键的问题。解决这个问题需要从系统结构、软件、算法以及程序设计语言等各方面着手，掌握各种因素对系统性能的影响，以便为合理的设计提供科学依据。今天或明天的超级计算机使用对一个国家的生产正在作出或将作出实质性贡献。Hwang和Briggs在1984年曾指出：“当我们发展数据处理、知识处理中的信息处理，最后到人工智能处理，对计算机的使用正变得越来越深入，每深入一层就需要更强的计算机”。当今科学技术发展已对超级计算机提出了更高的要求。下面用几个具体的事例说明这一点。

(1) 数值天气预报

用电子计算机来预报全球气候需要求解一组球面坐标系下的一般环流模式方程。通常，用垂直高度、纬度和经度将大气划分为三维网格，时间作为第四维，用一个指定的时间增量将方程组离散。在一个半球上给定间距为270英里的网格，并给定一个适当的时间增量，24小时的预报需要1000亿次操作，在每秒可执行一亿次操作的计算机(如CRAY-1)上，需要花费大约100分钟。这种网格可以用于纽约和华盛顿的预报，但不能用于费城的预报，费城大约位于纽约和华盛顿的中间。要得到费城的更精确的预报，在所有四维上还必须将网格大小减半，其计算量将增加16倍，对于每秒100MFLOPS的计算机(CRAY-1)来说，完成24小时天气预报需要24小时计算。即使是这种新的网格，也不能有效地作长期预报，如果我们希望得到更精确的长期预报，必须发展更高性能的计算机。

(2) 计算空气动力学

今天用于空气动力学设计的两个基本手段是风洞和计算机。其中，风洞是飞行器设计中进行气流模拟的主要手段，目前计算机只能起辅助作用。但是风洞实验一方面要受模型规模、风速、密度、温度等干扰，同时还要受许多其他因素的限制。然而数值流动模拟却没有这些限制，它仅仅受计算机的计算速度和存储空间限制；另一方面建造和使用风洞的设施是十分昂贵的。据介绍^[1]，用一个现代化的风洞进行飞机翼型测试，每年要耗资1.5亿美元。如果能用计算机取代风洞的功能，则不但可以节省大量的费用，而且计算机可以实现许多风洞无法实现的空气动力学模拟，使空气动力学特性设计更为科学化。计算空气动力学的基本任务就是用高速计算机求解流体力学方程组，以

实现空气动力学数值模拟。在实际的空气动力学设计中,计算机只有在10到15分钟内完成模拟,它才能当作一种工程手段,否则只能作为程序的研究。比如要求解的问题是三维雷诺平均N-S方程,采用 10^6 个网格点,使用最新的算法,计算机也必须具有持续每秒十亿次以上的浮点运算速度,才能在10分钟内完成模拟^[2]。目前市场上的超级计算机的峰值速度大都在每秒10亿次以下,实际运行中还远远达不到这个峰值速度。一般仅能达到峰值速度的10%到40%^[3],而要达到持续每秒10亿次以上的浮点运算速度,计算机的峰值速度必须达到每秒80亿次浮点运算,这个速度大约相当于CRAY-2速度的四倍。达到这个目标后,计算机可以替代部分风洞实验,但还不能完全代替风洞实验。

目前由美国国家科学基金资助的ILLINOIS大学研究机构已使用超级计算机来研究风剪切,对遇到微爆炸的“飞行”模拟飞机作空气动力学计算。他们希望通过计算进一步研究微爆炸的危险出现。

用今天超级计算机设计整架飞机还不可能,如要对整架飞机进行设计,还需要更强大的计算机,其中包括复杂的三维流模拟和相应的化学反应模拟的发动机设计,这需要更大的计算机能力和容量,甚至至少要比今天的超级计算机高出三个数量级。

(3) 遥感 (Remote Sensing)

从卫星发射回来的地球资源数据的分析,它在农业、生态学、林业、地质以及土地使用规划方面都有许多应用。然而,卫星发回的图象数据通常都是如此之浩渺,以致于其中简单的计算也要花费大量CPU时间。例如美国从最近发射的Landsat卫星发回的一幅单个的Thematic Mapper图象,它是6000像素乘6000像素的点阵图。每个像素由8比特描述,整幅图由8个图象或象带组成。这样,单个一幅图就用288兆字节信息来描述。美国国家航空和航天管理局(NASA)已经安装了由Goodyear Aerospace制造的大规模并行处理机(MPP),使用它来执行卫星图象处理。

(4) 重整化技术

重整化技术是一个理论上的重要突破,它为理解一系列尚未解决的科学与工程问题提供新框架。而这些科学与工程问题涉及到量子场理论、材料相转化的起源、湍流的发展、金属裂缝的传播及油田开发等问题,其中只有一小部份能够用解析方法解决。大型计算对重整化方法的应用是必不可少的,即使今天最大的计算机也不足以解决这些问题。

(5) 人工智能

大多数通用计算机都有一个相对输入/输出(I/O)相互作用面。但是,如果计算机在最高层次上通过说话,图和自然语言使人类与计算机进行交互作用,这就需要对声音、图和自然语言的实时输入进行处理。这种处理的数据量是相当大的,决非今天的计算机能够解决得了。日本已开始了第五代计算机研究工作,其目标之一就是建立一种新的计算机,其能力每秒可执行一亿次到十亿次(100 Million to 1 Billion)的逻辑判断。由于一个逻辑判断要执行100到1000条机器指令,因此所设计的新一代计算机必须每秒完成一百亿到一千亿条指令。

(6) 军事使用

目前美国现有的超级计算机正为军事研究机构使用。在先进的武器设计中,特别是核武器设计、核武器效应的模拟等需要进行大型计算模拟。这种大型计算模拟可以真正

取代实验方法。类似地，未来核电站的设计、情报的收集、为自动绘制地图而用的图象数据处理等运算都需要比今天的超级计算机大几个数量级才能满足军事上的要求。

当然科学计算的意义不在于解决一些老问题，而是通过数值模拟和实验发现新现象。象孤立子的形成、奇异吸引子的出现以及一大类非线性系统所共有的普适性等一些非遍历性质的发展是这个科学过程实例。很显然，这些发现需要极强能力的超级计算机的支持。

1.2 计算机达到高性能所使用的方法

今天计算机工程已使用了两种方法使计算机达到高性能。第一种方法是提高线路的速度，第二种方法则是通过流水线或并行性来增加可并行执行的操作数目。怎样使用流水线和并行性来设计现代高性能计算机呢？Hockney和Jesshope^[4]提供了许多有关先进计算机方面资料。

图1.1解释了过去三十多年来如何提高计算机的性能。从图中我们可以粗略地看到，每五年计算机的算术运算速度增加10倍。比如1950年到1980年间，浮点乘的速度大约增加一千万倍。

其原因部分归因于元器件速度的增长，但结构先进性却是增加速度的另一个重要因素。如果我们把某一个时期器件性能的提高（用器件级延迟时间表示）和同一时期的计算机系统性能的提高（用平均指令时间表示）作一比较，可以发现，计算机性能提高指数要比器件级性能提高指数大得多。Enslow统计了1965年~1975年这十年间，器件性能提高近10倍，但计算机系统的性能却提高近百倍。又如MPP的时钟比EDSACI的时钟快一个数量级，但其他加速度的提高则是由于MPP并行完成了16384个乘法的原因。因此在多级上引入并行性可以提高计算机的性能。在过去的30多年中，各种不同的新技术都得到了发展，下面我们将加以叙述。

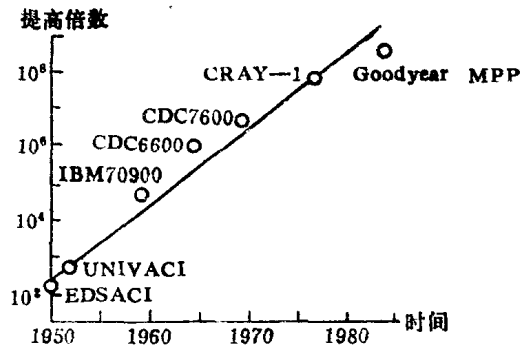


图 1-1 计算机的性能每 5 年提高 10 倍

量级，但其他加速度的提高则是由于 MPP 并行完成了 16384 个乘法的原因。因此在多级上引入并行性可以提高计算机的性能。在过去的 30 多年中，各种不同的新技术都得到了发展，下面我们将加以叙述。

(1) 位并行存储器和位并行运算

第一台电子数字计算机采用了位串行主存储器，我们可从主存储器中单独地读字的每一位。比如 EDSAC, SEAC, PILOT, ACE, EDVAC 和 UNIVAC 等均设有水银延迟线（超声速的）的位串行存储器。

并行访问字的所有位的第一个存储器是一个阴极射线管（CRT）系统，它是以英国 Manchester 大学的 Williams 命名的。尽管 Williams 是将阴极射线管使用于位串行模型上，但英国的数值分析学院的 SWAC 计算机却将 Williams 管使用于并行模型上，即字的第 K 位存放在第 K 个 CRT 中。类似地 IBM701 也是将 Williams 管使用在并行模型上。只要位并行存储器是可用的，则位并行运算也成为可能，IBM701 就是具有位并行运算的第一台商用计算机。IBM704 则是第一台使用原型铁酸盐核心存储器（Ferrite Core Mem-

ory)，它不仅允许位并行存储访问，而且还具有价格合适、无易变性等许多其他优点。

(2) I/O处理机(通道)

在第一代电子计算机中，使用CPU执行I/O指令，这样不仅浪费了CPU的时间，而且I/O传输率也因受CPU测试I/O设备状态和为I/O设备服务的速度所限制。在这以后采用磁带驱动使I/O传输变得比较快(磁带驱动大约比卡片输入快100倍)。但是磁带驱动数据传输的速度也远远慢于处理机的数据处理速度，这是因为电机I/O设备比电子CPU慢。因此在执行I/O指令时，CPU大部分时间处于空闲状态，这种低效率大大降低了计算机的性能[5]。

解决这种低效率的方法是引进I/O处理机来进行I/O操作。这种I/O处理机方式也称为通道方式，它接收CPU发来的指令，然后按指令要求与指定的I/O设备进行联系，并独立工作，释放出的CPU将继续进行运算处理，通道有它自己的指令集，也就是常规的I/O操作。例如IBM709就是在IBM704上增加了6个通道而形成I/O处理机，它是一种结构先进的有特色的第二代计算机(见图1-2)。这种I/O处理机方式仅仅是面向外设的控制和数据的传输，其通道指令也仅仅是一些简单的与I/O操作有关的命令，并在CPU的I/O指令控制下工作。由于它还不能进行码制变换、数据格式加工、数据块的检测与校正等，因此这种方式仍然要影响整机处理速度。

为解决I/O处理机方式的不足，目前是在高速计算机系统中采用外围处理机方式。由于这种外围处理机结构更接近于一般的处理机，甚至可以使用一般中、小型机作为外围处理机，其处理功能比I/O处理机强，从而可以更进一步提高整机系统的速度。例如CYBER170就是采用这种方式。图1-3表示了它的外围处理机系统框图。

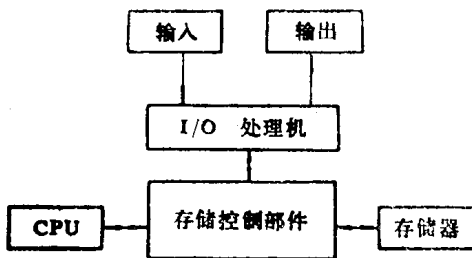


图 1-2 I/O处理机是第二代计算机中结构先进的有特色的处理机

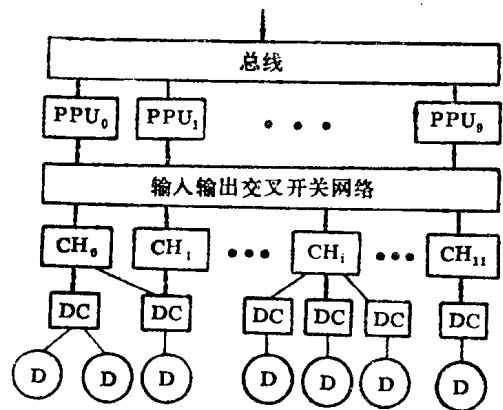


图 1-3 CYBER 170外围系统框图

(3) 交叉存储器(Interleaved Memory)

交叉存储器是一种含多个存储体，并能同时访问的存储器。每一个体有它自己的寻址线路，指令和数据被交叉可获得并行取的能力。当使用低阶交叉(Low-order Interleaving)时，地址的低阶位决定了含该地址的存储体，当使用高阶交叉(High-order In-

terleaving) 时, 地址高阶层决定了存储体。图1-4 说明了低阶交叉和高阶交叉之间的差异。

在设计一台计算机时, 各个部件的速度配置是很重要的。例如, 如果存储器部件不能保持提供指令和数据的话, 就不可能很好的做成高速CPU。IBM STRETCH 计算机(1961) 是一台具有交叉存储器的计算机, 由于采用了交叉存储器, 使得相对慢的磁心存储器能够赶上快速处理机。比如, 当我们将存储器分成二个体时, 则从存储器中取或送入存储器的传输率可以提高 2 倍。

ATLAS计算机(1963) 采用四体交叉存储器, 而CDC6600(1964)则采用了32个体的交叉存储器。实际上目前高速计算机均使用了交叉存储器。

(4) 高速缓冲存储器

高速缓冲存储器 (Cache Memory) 是一种小的快速存储器。它是为了弥补主存速度不足, 在处理机和主存之间设置了一个高速、小容量的缓冲存储器, 构成“Cache—主存”层次, 这样就可减少处理机等待主存数据传输时间。见图1-5。

Cache的命中率对机器性能影响很大。采用什么方法提高Cache的命中率是Cache设计中的很重要问题, Cache的命中率与信息块大小、Cache的容量、所

采用的主存到Cache的地址映象方法以及替换策略等密切相关, 它们可以通过模拟方法加以分析。一般情况下, “Cache—主存”层次等效速度与命中率的关系为:

$$t_e = H_c t_c + (1 - H_c) t_m \tag{1.2.1}$$

其中 t_e 为“Cache—主存”层次等效速度, H_c 为访问Cache的命中率, t_c 为Cache的访问时间, t_m 为主存周期。因此采用Cache比处理机直接访问主存, 其速度提高倍数为:

$$\rho = t_m / t_e = 1 / [1 - (1 - t_c / t_m) H_c] \tag{1.2.2}$$

令 $H_c = \alpha / (\alpha + 1)$, 则(1.2.2)式变为:

$$\rho = 1 / \left[1 - (1 - t_c / t_m) \frac{\alpha}{\alpha + 1} \right] = (\alpha + 1) t_m / (t_m + \alpha t_c) \tag{1.2.3}$$

因为 $t_c / (t_m + \alpha t_c) < 1$, 所以 $\rho < (\alpha + 1)$, 若 $H_c = 0.5$, 相当于 $\alpha = 1$, ρ 的最大值比 2 小; 若 $H_c = 0.75$, 相当于 $\alpha = 3$, ρ 的最大值比 4 小; 若 $H_c = 1$, $\rho = \rho_{max} = t_m / t_c$, 这是 ρ 的可能最大值。 ρ 的期望值与命中率 H_c 的关系如图1-6所示。

存储体

0	1	2	3
0 (0000)	1 (0001)	2 (0010)	3 (0011)
4 (0100)	5 (0101)	6 (0110)	7 (0111)
8 (1000)	9 (1001)	10 (1010)	11 (1011)
12 (1100)	13 (1101)	14 (1110)	15 (1111)

(a)

存储体

0	1	2	3
0 (0000)	4 (0100)	8 (1000)	12 (1100)
1 (0001)	5 (0101)	9 (1001)	13 (1101)
2 (0010)	6 (0110)	10 (1010)	14 (1110)
3 (0011)	7 (0111)	11 (1011)	15 (1111)

(b)

图 1-4 交叉存储

(a) 低阶交叉允许地址的低阶位决定存储体

(b) 高阶交叉允许地址的高阶位决定存储体

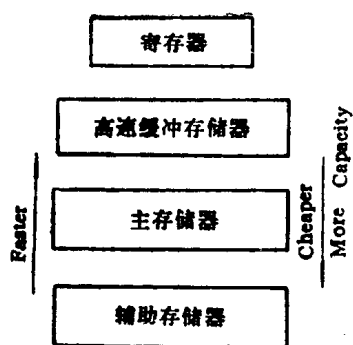
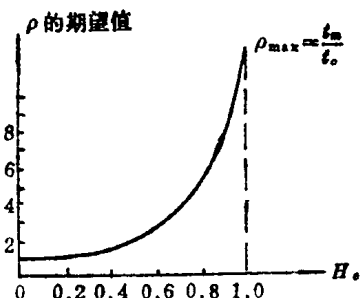


图 1-5 一个典型存储器分级结构

图 1-6 ρ 的期望值与 H_c 关系

(5) 指令预估 (Instruction Look-Ahead)

高速缓冲存储器的使用能使快速指令部件处于忙的工作状态。但另一个问题是指令预估，也称为指令缓冲。在这种方式下，少量要执行的指令对于指令部件来说总是有效的。因此指令的预先取出要象其他操作（如指令译码）一样快速得到。

IBM STRETCH 计算机设有挑选指令、译码指令、计算操作地址和预先取出操作数等若干指令。IBM360/91 也具有同样的预估能力，此外，它还能预取分支目标指令和一些后续指令，以保持管道不被占用。在有流水线功能部件下，指令预估也是需要的。如 CRAY-1 机和 YH-1 机均设有指令缓冲站。当计算机工作时，系统首先将要执行的目标程序存放在主储里，当执行该程序时，依次将程序逐段地从主储调到指缓站，指缓站的指令依次逐条译码，并发送出去执行。

(6) 多功能部件

获得指令流和预取操作数的一种方法是审查一组要实现的指令，并决定在没有改变程序语义下，什么样的操作可以并行实现。在给定一组功能部件（加、乘等）和一组操作间的优先关系下，我们的目的是要寻找这样一种方式，使得在这组功能部件上来处理这些操作，所需要的总时间为最小。

CDC6600 使用 10 个功能部件来进行实际计算，其中二个功能部件实现增量，二个功能部件实现浮乘。对于每一个操作，如移位、布尔运算、分支、浮点加、整加和浮点除均提供一个功能部件实现。一当功能部件开始处理指令时，它就与其他功能部件一样独立工作。因此，许多功能部件可以同时执行不同指令，至少可以希望二个或三个功能部件在同一时刻，在执行一个典型程序任何点上进行运算。图 1-7 是 CDC6600 多功能部件区图解。

YH-1 计算机设有 18 个独立的专用功能部件。其中包括一组地址功能部件、一组标量功能部件、二组向量功能部件和二组浮点功能部件。每组又有若干个功能部件，各功能部件可以独立地执行相应的运算。因此，在不同功能部件的指令之间彼此无关情况下，它们可以并行地执行运算操作。

(7) 指令流