

胡 玥 高庆狮 高小宇 著

串行算法 并行化基础



科学出版社
www.sciencep.com

串行算法并行化基础

胡 玥 高庆狮 高小宇 著

本专著的有关研究工作先后得到

973 计划资助(项目编号:2007CB311103)

863 基础研究基金资助(项目编号:863-306-05-01-1)

国家自然科学基金资助(项目编号:GJZRJJ-60083008)

科学出版社

北 京

内 容 简 介

引入并行是为了提高计算速度,到底能不能有效提高计算速度?如何度量计算速度的提高及其有效性?这些需要通过一些基本概念来刻画。本书第1章就是首先介绍这些有关串行算法并行化基本概念。并行计算是在一定的并行计算系统的类型上实现的,所以第2章介绍一些基本并行计算系统类型。多指令流多数据流巨型机是当今高性能计算机系统的主流,许多大部头的书都有详细论述,本专著就不重复。单指令流多数据流巨型机是20世纪60年代末到80年代并行计算的高性能计算机系统的主流,其中许多设计思路在当今仍然不失其价值。它们很容易使用的原因是对应的并行计算模式可以规范到十分自然的向量运算形式,即有一个理想的描述语言:向量语言。第3章就介绍一种向量语言。多指令流多数据流巨型机的并行计算模式目前难于规范到十分自然的运算形式,也就是尚不存在一个理想的描述语言。通过向量语言的了解,或许有助于今后多指令流多数据流高性能计算机系统理想的描述语言的诞生。第4章介绍串行算法并行化的各种类型。第5章到第7章介绍具体的、典型的串行算法的并行化,包括两路归并、多路归并、排序和广义一阶递推。最后一章(第8章)介绍一类广函数——纵横矩阵加工广数。

图书在版编目(CIP)数据

串行算法并行化基础/胡玥,高庆狮,高小宇著. —北京:科学出版社, 2008

ISBN 978-7-03-021720-2

I. 串… II. ①胡…②高…③高… III. 并行化算法 IV. TP301.6

中国版本图书馆CIP数据核字(2008)第056265号

责任编辑:王淑兰/责任校对:赵燕

责任印制:吕春珉/封面设计:三函设计

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂印刷

科学出版社发行 各地新华书店经销

*

2008年6月第一版 开本:B5(720×1000)

2008年6月第一次印刷 印张:8

印数:1—1 000 字数:180 000

定价:28.00元

(如有印装质量问题,我社负责调换〈新欣〉)

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

献给我的启蒙老师们：

母校北京大学数学力学系——

传授美丽简洁的仿射空间的江泽涵老师；
在万变中寻找不动点和不变性质的吴光磊老师；
用抽象难题严格训练学生的冷生明老师；
教学方法严格、精确、绝妙的丁石荪老师；
感人的热诚和严谨的陈杰老师；
讲授神奇宇宙的戴文赛老师；
才华横溢的段学复和程民德代课老师；
讲授让万物循规法则的钱敏老师；
揭示宏观和微观世界奥秘的钱尚武老师；
燃烧自己，照亮别人的张世龙老师。

百年母校漳州一中的覃景芬代数老师、陈“妈妈”常玉几何老师；
漳州钟芬小学我的数学老师杨校长；
永远感谢你们，永远怀念你们，我的启蒙老师们！

献给我有幸能够得到较长时间直接教诲的共和国泰斗们——

中国“巨型机”的指路人，把物理概念融化在数学方程中的“两弹一星”功勋科学家钱学森老师；

平时和颜悦色，关键时刻怒斥“四人帮”的“两弹一星”功勋科学家钱三强老师；

充满父辈关爱的“两弹一星”功勋科学家王淦昌老师和杨嘉墀老师；
生活十分简朴，平易近人，深受日本物理界敬重的周培源老师；
思维敏捷，精力充沛，永远在思考的华罗庚老师；

平易近人，风雨无阻骑着车从中关村到端王府上班的“两弹一星”功勋科学家陈芳永老师；

八十多岁高龄时仍然热情豪放，热爱专业，身材魁梧，声音宏亮的汪德昭老师；

年迈仍然不知疲倦，白天参加“人大”会，晚上工作到深夜，日日夜夜抢时间的黄炳维老师。

永远感谢你们，永远怀念你们——我的老师，共和国的泰斗！

高庆狮

前 言

当今，高性能计算机系统令人注目。可惜高性能计算机系统两个重要难点至今没有解决：一是不好用，二是效率低。20 世纪 60 年代研制的“单指令流-多数据流”ILLIAC-4 机（最早的巨型机）就遇到了这样的问题，因为它在运行中的并行、存储和通信等全部需要用户进行人工处理，因此很不好用。科研人员通过改进系统结构、引入向量语言和高级编程语言自动向量化，解决了“不好用”的问题，同时“效率低”的问题也得到了缓解。而当今的高性能计算机系统是“多指令流-多数据流”，其在运行中的并行、存储和通信等还是由用户来人工处理，所以“不好用”的问题依然存在，“效率低”的问题由于并行台数激增而更加严重、更加突出了。

本书一方面借助于“单指令流-多数据流”巨型机的历史经验，有助于寻找“多指令流-多数据流”高性能计算机系统“不好用”的问题的解决方法；另一方面通过串行算法并行化的基本方法的介绍，希望有助于读者独立处理各种实际问题的并行化，进而有效地提高计算效率。

借本书出版介绍一下“为什么研究串行算法并行化”，和回答一下一些研究生提问的问题：“如何寻找科研课题”。

一、为什么研究串行算法并行化

为什么研究串行算法并行化呢？这要从我们接受亿次机设计任务说起。1973 年 3 月中国科学院计算所老所长阎沛霖带我到国防科工委钱学森那里接受亿次机设计任务开始，两个月后，也就是 1973 年 5 月我们提出了可行的解决方案，并正式承担了亿次巨型机设计任务及其模型机——中国第一台向量计算机 757 的研制任务。巨型机的实际解题能力取决于巨型机的峰值速度 V 与题目并行度 N ，而峰值速度 V 取决于基本处理机速度 V_0 与并行台数 M ，即

$$V = MV_0$$

例如，当 $V_0 = 1$ 亿次/秒时， $M = 100$ ， $V = 100$ 亿次/秒； $M = 2000$ ， $V = 2000$ 亿次/秒。当 $V_0 = 100$ 亿次/秒时， $M = 100$ ， $V = 1$ 万亿次/秒； $M = 2000$ ， $V = 20$ 万亿次/秒。

但是，峰值高不等于实际解题能力强。

例如，如果题目只能串行计算， $M = 2000$ 的 20 万亿次/秒的高性能计算机或者巨型机，其实际解题速度只有 100 亿次/秒，而不是 20 万亿次/秒。效率只

有 0.5‰，即 0.05%。这将造成极大的浪费！

又如，如果题目有 10% 只能串行计算， $M=2000$ 的 20 万亿次/秒的高性能计算机或者巨型机，其实际解题速度不可能超过 1000 亿次/秒，而不是 20 万亿次/秒。效率就不可能达到 $1/200$ ，即 0.5%。这也将造成极大的浪费！而且 M 越大，效率越低。

解决“效率低”的方法就是串行算法并行化。如果串行算法并行化了，那么，即使并行的效率只有一半，实际效率就会由 0.05% 提高到 50%，即实际计算能力提高了 1000 倍！即便并行化的效率只有十分之一，实际效率也会由 0.05% 提高到 10%，即实际计算能力提高了 200 倍！这就是我们在 1973 年承担亿次/秒巨型机任务的同时，开展了并行系统结构的研究，也开展了串行算法并行化的研究工作的原因。

二、如何寻找研究课题

有一些研究生问我如何找研究课题，下面的轨迹或许是一个答复。

发现 Łukasiewicz 多值命题逻辑理论的错误和缺点是因为 Zadeh 先生的同僚喜欢用错误的 Łukasiewicz 多值命题逻辑理论的所谓“ $A \vee \neg A \neq \text{真}$ ，和 $A \vee \neg A \neq \text{假}$ ”来解释“ $A \cup \neg A \neq \Omega$ （全集）和 $A \cap \neg A \neq \emptyset$ （空集）”。而 Zadeh 模糊集合论缺点和错误是在给学外语专业跨大学科研究生开一门指导学习课程：“语言学进展”阅读模糊语言学时偶然发现的。从 20 世纪 80 年代在中科院计算技术研究所，90 年代在北京科技大学，到 21 世纪在大连理工大学召外语专业跨大学科研究生是因为发现机器翻译巨大经济效益和社会效益的前提是没有语无伦次、没有正错混杂，合乎语言表示规律。其关键是必须从自然科学的角度来研究语言学。这就需要培养一批跨学科的语言学人才。研究机器翻译是因为研究人类智能及其模拟和应用。智能是指能够自动学习知识和自动并且有效地利用学习到的知识去解决问题的能力。人类在数学、智力游戏、写论文等等活动表现出的能力都能够反映出智能。人类的语言能力同样能反映出人类的智能。一个小孩放在东京、伦敦、北京，他就会流利地讲日语、英语、汉语。不需要大人们干预，更不要修改他们的“程序”。由于 1980 年日本东京一本小册子，谈到自然语言之间的翻译未来市场很大，才意识到机器翻译未来是具有巨大的经济效益和社会效益。之后，才把它作为一个独立研究课题。研究人类智能是因为 1980 年为国防科工委的研制巨型机任务转到国防科技大学之后，科学院转向研究“未来面向智能领域应用的巨型机”（注：该项研究经过科学院、国家科委和国防科工委推荐，民口论证、军民联合论证和中南海论证和钱学森支持之后，被列入 863 项目）。之后发现全世界人工智能 50 多年来所研究的系统，没有一项有智能，因而重新研究人类的智能。而研究巨型机任务是从 1973 年 3 月中科院计算所刚刚“解放”的老所长

阎沛霖带我到国防科委钱学森主任接收亿次机设计任务开始。虽然之前的几年，钱老曾经布置过国防科委下属的两个研究所进行研究，但是，反馈回来只有一批调研资料和国内技术条件不成熟的意见。当时国内实际进行计划仍然是 200~500 万次，不能满足需要，国际上巨型计算机 ILLIAC-IV 因为不可靠及难于使用正处于一片批评声中，而 SRAR-100 的条件国内难于满足。因为我们在阅读国防科技情报所和计算所情报室所提供国外有关巨型机材料后，发现这两种绝然不同的两种巨型机是等价的。关键在于向量必须进行分段、流水线处理，使用必须依靠向量语言，所以两个月后的 1973 年 5 月提出了可行的解决方案，正式承担这项巨型机设计任务及其模型机——中国第一台向量计算机 757 的研制任务。并且得到两星期向钱老汇报一次和聆听钱老讲述他如何把物理概念和数学方程融合在一起解决问题的经验的机会。而有机会承担国防部门巨型机任务是因为 1957 年我从北京大学数学力学系毕业之后被分配到中国第一个计算机系统结构研究和设计小组，承担中国第一台自行设计大型电子管计算机、第一台自行设计大型晶体管计算机和专为两弹一星服务、被誉为“功勋计算机”的 109 丙计的系统结构设计。而 1957 年我被分配到中国第一个计算机系统结构研究和设计小组是因为我在 1955 年被动员改学计算数学，1956 年参加北大、清华与计算所筹备处合并第一届训练班的计算数学组，而且根据苏联计算机领域的领导人列贝捷夫院士的意见，要安排数学专业的人员从事计算机系统结构研究和设计工作（这点很重要，是关键。比具体承担者是谁更重要）。一切似乎是偶然，偶然背后又有必然。这是一名北京大学数学系学生毕业后众多的工作轨迹中的一个。20 世纪 50~70 年代，课题是国家给的，努力去完成就是。80 年代开始，国家只给意向，甚至不给，要靠自己独立思考。关键是：独立思考，不人云亦云。独立判断是非曲直，独立判断经济效益、社会效益和理论价值。冷对众说纷纭。或许以下三句话有参考价值：

“任何正常人都有优点和特长，检查检查你的优点和特长在哪里。人类的需要是阳光，你的优点和特长是水和土壤。有了阳光、水和土壤，你的兴趣就会带你事业上飞翔。没有阳光，或者没有水和土壤，兴趣只能给你幻想。”

“要特别注意那种经济效益和社会效益很大，人们认为做不到，难度很大但可能做到的事，因为这往往是重要的生长点和突破口。”

“不要幻想经过成千上万个聪明人没有搜索到的重要的科技宝藏，会突然从天上掉到你的口袋里。首先想一想解决它需要什么先决条件？例如跨学科知识，你是否具备？你是否有决心和有条件去具备？”

本书是作者根据 1973 年 5 月开始到现在在串行算法并行化的研究成果基础上撰写的。主要内容取材于 20 世纪 90 年代后期和本世纪所取得的研究成果。参加写作的作者有胡玥，高小宇和我。本书旨在帮助读者掌握相关内容的完整的基

本概念、基本概念之间的相互联系，以及通过典型例子掌握基本方法，以便今后能更有效地独立进行与并行计算有关的工作。希望本书有抛砖引玉的作用。

计算机科学在半个世纪的发展中，其速度性能的提高、可靠性的提高、计算成本的降低、体积的缩小可以用“百亿倍”来刻画。它神奇地让时间变“快”，让距离变“短”，许许多多本来不可能实现的事成为可能和现实。计算机带来了巨大的产业革命和社会革命。支撑计算机半个世纪辉煌的基础是不超过 100 项的根源性创新。所谓根源性创新是指在根源部位能影响新的大产业的产生或者发展的科学或技术创新。例如：阴极射像管，锗半导体硅半导体集成电路（奔腾-4，4000 万个晶体管/片），磁心存储器（在美国使用了 10 多年，在中国使用了近 30 年），磁盘光盘，布尔代数（逻辑部件构造性描述），指令作为数据被加工，B-管（变址），中心的转移（由运算器转到存储器），分时中断多道程序，管理程序操作系统，一级存储器（即后来的虚拟存储器），微程序，关系数据库，流水线阵列结构，并行算法，并行分布式处理，模块化符号化编程技术，高级语言，结构程序化技术，可视化编程技术，面向对象的编程技术，软件工程，编程工具环境，用户图形界面，库函数，构件，激光打印机，互联网等。需要说明的是，“指令作为数据被加工”这种似乎是纯理论的东西怎么会与产业、经济效益挂钩？事实是，如果没有这个发明，高级语言及其编译系统不可能诞生，软件不可能神速发展，依赖于软件接口的互联网也不可能存在。一句话，没有它，半个世纪的辉煌就不可能发生。还应该指出，软件的研制效率与硬件相似，也是数十倍、数十倍地往前发展。软件的发展，往往需要牺牲硬件的运行效率为前提。例如，经过编译，编程的效率极大地提高，但是程序本身的运行效率也迅速下降。所以说，硬件的神速发展促使并支持软件发展，结果，计算机应用获得快速而广泛的普及，进而促进了硬件的发展（扩大了需求和资金）。

很遗憾，数十项的重大根源性创新中没有一项发明是中国籍华人的。磁心存储器发明人王安是个美籍华人。但是，相信随着中国经济的腾飞，在未来的半个世纪里，在各重要领域的创新发明榜上将会有中国人的名字写在上面。这个历史任务将落在谁的肩膀上？不是我们，而是你们——朝气蓬勃的年轻人！希望年轻的科技工作者能担当起这个历史使命！这是老一辈科学家对你们的殷切希望和诚恳的祝愿。

中国科学院院士

高庆狮

2007 年 9 月 25 日

目 录

第 0 章 绪论	1
0.1 计 算 科 学	1
0.2 为什么要并行计算	2
0.3 巨型机、高性能计算机本质特征:并行计算	3
0.4 巨型机、高性能计算机基本矛盾:台数与计算效率的矛盾	3
0.5 并行运算和并行数据传送	3
0.6 并行执行方式和重叠执行方式	4
0.7 并行算法与串行算法并行化	4
0.8 巨型机、高性能计算机的关键技术	4
0.9 数据相关和控制相关	5
第 1 章 串行算法并行化的基本概念	6
1.1 题目的规模与计算工作量 N	6
1.2 题目的计算时间 T	6
1.3 题目最快串行计算算法 C_0	7
1.4 题目在并行计算模型 $M(S)$ 下并行计算算法 B	7
1.5 题目在 $M(S)$ 下并行计算算法 B 的计算速度: $V_{B,M(S)}(N)$	7
1.6 在并行计算模型 $M(S)$ 下题目并行计算算法 B 的加速比	8
1.7 在并行计算模型 $M(S)$ 下题目并行计算算法 B 的效率	8
1.8 并行算法 B 的计算复杂性	8
1.9 常数效率并行算法	8
1.10 在某些讨论中的算法分类	8
1.11 并行计算台数 S 对并行计算速度的影响及串行算法并行化的意义	9
第 2 章 执行并行计算算法的并行计算机系统结构模型	10
2.1 并行算法实现的两要素之一:并行传送	11
2.2 单指令流-单数据流(SIMD)计算机	13
2.3 SIMD 二维阵列机	14
2.4 流水线向量机	15
2.5 第二代巨型机:纵横加工(分段处理)流水线向量机	15
2.6 细胞结构化虚共存纵横加工向量机	17

2.7	多维立方体机	19
2.8	多指令流-多数据流系统 MIMD	19
2.9	内部互联网络	19
2.10	通用或专用计算网络	19
2.11	PRAM 并行随机访问计算机	21
2.12	可变总线结构	22
2.13	素数存储系统	24
2.14	分段线性变换存储系统	25
第 3 章	向量语言	29
3.1	数据类型与数据结构	29
3.2	向量基本运算	29
3.3	向量或者数组中的向量	32
3.4	可以用硬件实现的控制向量	33
3.5	变长向量运算	38
3.6	向量语言的扩充	40
3.7	向量高级语言	41
第 4 章	串行算法并行化方法综述与比较	42
4.1	串行算法并行化之一:多分法方法	42
4.2	串行算法并行化之二:倍增法	43
4.3	串行算法并行化之三:纵横加工法	43
4.4	串行算法并行化效率比较	44
4.5	串行算法并行化之四:利用软件、硬件和软件硬件结合的优化方法	44
4.6	串行算法并行化之五:利用硬件直接实现的控制向量	45
第 5 章	两路归并与分类串行算法并行化	47
5.1	归并与排序的快速串行算法	47
5.2	归并基本定义与定理	50
5.3	K E Batcher 的 Odd-even 并行归并网络	53
5.4	根据归并基本定理所构造的快速并行归并算法	54
5.5	K E Batcher 的 Bitonic 归并算法	55
5.6	利用并行归并来实现并行排序	59
5.7	归并与排序串行算法并行化的 OPTIMAL 并行算法之一:纵横并行归并算法	60
5.8	归并与分类串行算法并行化的 OPTIMAL 并行算法之二:k-维并行归并算法	61

5.9 在理论模型上的排序.....	63
第6章 多路归并串行算法并行化	65
6.1 多路归并与排序的纵横斜并行算法.....	65
6.2 多路归并的 k-Bitonic 及 k-Bitonic 并行算法:Odd-even 并行算法与 Bitonic 并行算法的统一推广算法	65
第7章 一类一阶递推串行算法并行化	72
7.1 一类一阶递推的方程和最快串行算法.....	72
7.2 一类一阶递推方程的例子和应用背景.....	73
7.3 一阶递推和一类递推串行算法并行化的基本原理、思路和定理	74
7.4 一类一阶递推串行算法并行化的倍增(Doubling)算法	79
7.5 一类递推串行算法并行化的常数效率(OPTIMAL)并行算法之一: 一类递推纵横并行算法.....	82
7.6 一类递推串行算法并行化的常数效率(OPTIMAL)并行算法之二: 一类递推 k-维并行算法	84
7.7 一类递推串行算法并行化的 OPTIMAL 并行算法之三:一类递推 混合并行算法.....	88
7.8 几种算法的比较.....	91
7.9 一类递推的其他算法.....	92
第8章 一类广函数:纵横矩阵加工广函数.....	93
8.1 纵横矩阵加工广函数.....	93
8.2 纵横矩阵加工广算法的典型实例一:两路合并的 Bitonic 算法	100
8.3 纵横矩阵加工广算法的典型实例二:一类递推方程.....	102
8.4 纵横矩阵加工广函数的应用列举	107
8.5 结论	110
附录 (m, N)选择问题的纵横并行算法例子	111
参考文献.....	113

第 0 章 绪 论

0.1 计 算 科 学

0.1.1 计算科学是科学三大支柱之一

科学包括理论科学、实验科学和计算科学。理论科学和实验科学属于传统的科学。

自 1946 年数字计算机诞生以来,计算机以每 3~7 年计算速度和性能提高 10 倍,可靠性提高 10 倍,体积缩小 10 倍,单位计算的成本降低 10 倍的速度发展。速度与性能和可靠性的提高,体积和计算成本的下降在半个世纪中的发展可以用一个数字来刻画——百亿倍。

当今的个人计算机,其速度远远超过早期(20 世纪 60 年代末 70 年代初)的巨型机数十倍,甚至上百倍;而逻辑门数目超过数百倍,甚至上千倍(上亿个门:数十万个门);内存容量同样也超过上百倍,甚至上千倍;外存的容量超过数百倍,甚至数千倍。百亿倍的发展变化使得许多不可能的事成为可能,许多没有经济效益的事成为有巨大经济效益。这就是导致了整个社会的各个领域(生产、科学、技术、教育、生活、……,几乎是所有的领域)产生巨大的革命性的变化的重要动力。科学工程领域也不例外,计算科学因此而诞生:计算数学、计算物理学、计算化学、计算流体力学、计算生命科学……

0.1.2 计算的好处

【例】 计算流体力学^①。

1973 年 3 月,当科学院计算所到国防科委接受巨型机设计任务时,国防科委副主任钱学森指出:由于飞行器性能不断提高,复杂性不断增加,所以对风洞实验的要求也在不断增加。

波音-747 的风洞实验花了 1 万小时,超过 1 年。耗资为当时的 1000 万美元。

在 20 世纪 70 年代早期,计划中的 B-1 轰炸机需要风洞实验 4 万个小时,约 5 年。实验费用为当时的 4000 万美元(后来,70 年代末的航天飞行器的设计所需的风洞实验,时间估计需要超过 10 年,费用估算需要当时的两亿美元,情况相似)。

计算机计算可以代替风洞实验的大部分工作,既节省时间,又节省费用。何况

^① 参考国防情报(资料)所和中科院计算所情报(资料)室当时及后来根据钱学森指示所提供的材料。

每 3~7 年计算成本约下降 10 倍,计算能力提高 10 倍。这就使得计算流体力学成为具有巨大经济效益和社会效益的学科。

0.2 为什么要并行计算

0.2.1 计算的需求

20 世纪 70 年代,要用计算机对飞行器作比较理想的模拟计算,就需要解完全黏性的不定长的 Navier-Stokes 方程,这个计算需要用每秒运行 1 万亿次左右、内存容量数十亿字的巨型机。

许多物理问题、工程问题最后归结为解三维、四维到七维偏微分方程问题。如果三维问题每维取 1000 个点,每个点有 10 个参数,就有 10^{10} 个参数,这就要有 100 亿字存储容量。如果每点执行 20 个运算,再加上时间步 5000 步,这就需要 10^{15} 次操作,在 25 亿次/秒的计算机上需要计算 100 个小时。如果七维问题每维取 100 个点,就有 10^{14} 个点,每个点一个参数,存储容量就要 100 万亿字。如果每点执行 20 个操作,时间方向取 5000 步,这就需要 10^{19} 个操作,即 1 万亿次/秒的计算机要算 2500 小时。

处理复杂的大气模型,需要 10^{25} 次操作;下国际象棋,如果用简单方法去分析各种可能走法,仅不同顺序走法的组合就约有 10^{120} 种。

1985 年,石油勘探、地震信息处理仅一项,就要求计算机的总处理能力为

$$\sum_i V_i \times M_i \quad (10^{18} \text{ 字} \cdot \text{次} / \text{秒})$$

其中, V_i, M_i 分别为第 i 台计算机的速度(次/秒)及内存容量(字)。

0.2.2 器件水平满足不了需求时的解决方法:并行处理

有三种不同层次的并行处理。初层次:逻辑级并行;中层次:部件级并行;高层次:计算级并行,即并行计算。

- 逻辑级并行:从全字并行运算(不是逐个二进制位的串行运算或者逐个字节的并串行运算)发展到多位一乘,多位一除等。即部件内部高度逻辑并行。
- 部件级并行:在单指令流单数据流控制下多个部件同时运算。例如,多(运算)功能部件并行工作形式,运算部件—指令部件—存储部件高度重叠的流水线形式。
- 计算级并行:非单指令流单数据流的多计算部件或者多计算机同时计算,这就是计算级的并行。实现计算级并行的计算机系统称为并行计算(机)系统。计算级并行就是并行计算。实现并行计算的计算机系统的结构有

三种方式:重叠方式(例如,流水线)、并行方式(例如,二维阵列)或混合方式。

0.3 巨型机、高性能计算机本质特征:并行计算

所有的巨型机、高性能计算机都是建立在并行计算的基础上。结构上是流水线(重叠)方式,或者并行方式,或者流水线与并行混合方式。

单指令流多数据流(SIMD)计算机系统(无论是基于并行还是重叠)对应的并行计算模式可以规范到十分自然的向量运算形式。

多指令流多数据流(MIMD)计算机系统对应的并行计算模式和编程语言至今仍然处于很不成熟状态,类似于早期单指令流多数据流(SIMD)计算机系统中的ILLIAC-IV,把麻烦推给用户,至今没有适用的、自然的运算模式和编程语言。

混合方式(单指令流与多指令流混合计算机系统)对应的并行计算模式之一,将在第2章的2.6节中提到。

有限制的通用计算网络形式(例如,脉动计算网络、波前计算网络、排序计算网络等)是并行计算的具体算法计算的网络化。

0.4 巨型机、高性能计算机基本矛盾:

台数与计算效率的矛盾

巨型机、高性能计算机基本矛盾就是并行计算的基本矛盾。并行计算系统的效率取决于并行计算的台数 S (系统的并行度)与算法并行程度 P 之间的关系。在题目算法规模和并行程度 P 确定之后,台数 S 越大,效率越低。对流水线结构的并行计算系统而言, S 是等效台数。

0.5 并行运算和并行数据传送

并行运算和并行数据传送是进行并行计算,实现并行计算算法的两要素。

例如,计算 $a_1 + a_2 + \cdots + a_n$ 时,第 i 台处理器计算完 $b_i = a_{2i} + a_{2i+1}$ 后,在计算 $c_i = b_{2i} + b_{2i+1}$ 之前,必须把在不同处理器的数据传送到相同处理器上。

因此,实现并行计算的计算机系统不仅需要考虑并行运算,还必须考虑并行传送。后者需要依靠内部互连网络。

0.6 并行执行方式和重叠执行方式

实现并行运算有两种不同方式:并行执行方式和重叠执行(流水线)方式。

这好比在某工厂车间里有 8 个工人,每个产品需要进行 8 道加工。并行执行方式相当于每个工人都对一个工件进行 8 道加工,不同个人加工不同工件。重叠方式,即流水线方式,相当于每个工人对所有的工件只进行同一道加工,不同工人进行不同道的加工。

这两种方式功能是等价的,即从计算的角度没有区别,也就是能够达到相同的并行计算的目的。并行方式结构简单,但是设备效率低,即元器件利用率低;流水线结构复杂,但是设备效率高,即元器件利用率高。

0.7 并行算法与串行算法并行化

并行计算讨论的范围比较宽,包括各种计算或者应用问题在各种理论的或者实际的并行计算系统上的各种快速的或者高效的并行计算算法或者程序;各种计算或者应用问题,包括本来就可以并行计算的或者本来只能串行计算的问题。也就是,除了串行算法并行化之外,大量的基本可并行计算的算法在具体的并行计算系统上的具体计算算法。

基本可并行的算法常见的有矩阵运算、线性代数方程组求解、偏微分方程求解、FFT、卷积和滤波等。讨论它们的并行算法基本上在两个方面,这两方面常常融合在一起:

① 满足误差要求,又适合于并行计算的快速解法。这一部分可以划归数值计算(计算数学),也可以划归并行计算。

② 适合在某种并行计算机系统类型上实现的并行算法。

难以并行计算的问题的并行化,包括串行算法并行化和创造出新的并行计算系统或者计算网络来实现原来需要串行计算的问题,以及应用计算数学(包括数值计算)构造出新的可并行计算的数值计算方法。本书重点介绍串行算法并行化,顺带介绍一点所涉及的并行计算网络。凡是需要与计算数学结合在一起的并行计算及其网络,不在本书讨论范围。

0.8 巨型机、高性能计算机的关键技术

巨型机、高性能计算机的四个关键技术是:提高每个处理单元的水平有关的技术、串行算法并行化技术、有效处理计算目标的系统结构设计技术和使用户好用的

技术。

巨型机、高性能计算机的速度(即峰值) V

$$V = S \times V_0$$

其中, V_0 是每个基本计算单元的计算速度, S 是处理单元台数或者等效台数。

显然,巨型机、高性能计算机的速度 V 在并行计算台数 S 不变时取决于每个处理单元的速度 V_0 。每个处理单元的速度 V_0 越大, V 就越大。例如 $S=100$, $V_0=1$ 亿次/秒, $V=100$ 亿次/秒; $V_0=100$ 亿次/秒, $V=1$ 万亿次/秒;水涨船高。当巨型机、高性能计算机重点服务于确定的专门目标时,每个处理单元如何有效地处理确定的专门目标就成为关键。

同样,巨型机、高性能计算机的速度 V 在每个处理单元的速度 V_0 不变时取决于并行计算台数 S 。并行计算台数 S 越大, V 就越大。例如 $S=100$, $V_0=5$ 亿次/秒, $V=500$ 亿次/秒; $S=4000$, $V=2$ 万亿次/秒;同样是水涨船高。

不过, V_0 的提高,解题的速度实实在在地、有效地提高了。对解题的效率没有任何不良影响。而并行计算台数 S 的增加,解题的速度未必提高,提高不提高取决于计算的算法的并行程度。一般来说,解题的效率随着并行计算台数 S 的增加而下降,但是下降多少,取决于计算的算法的并行程度。而题目的并行程度有两类,一类自然并行度高,另一类本来并行度不高,依靠串行算法并行化技术来提高算法的并行度。也就是取决于算法并行化技术。

一个 S 台或者等效 S 台的并行计算(巨型机、高性能计算机)系统,串行算法的执行效率 $=1/S$ 。因为,无论多少台,始终只有一台工作。又如,如果题目有10%只能串行计算,无论 S 多大(2000台,20000台),计算其中的10%仍然是一台工作。效率就永远不可能达到 $10/S$, S 越大效率越低($S=2000$,效率低于0.5%; $S=20000$,效率低于0.05%)。极大浪费!解决的唯一方法是,把串行算法变成并行算法,即串行算法并行化。

S 台处理单元如何组织成为有效处理计算目标也是重要的。用户使用困难是当前高性能计算机的通病。

0.9 数据相关和控制相关

数据相关和控制相关是并行计算的两个主要障碍。

数据相关指后面的计算依赖于前面的计算结果。

控制相关指后面执行的指令依赖于前面计算的结果。例如,条件转移。

第 1 章 串行算法并行化的基本概念

1.1 题目的规模与计算工作量 N

(1) 题目的规模 N

例如, N 皇后问题求解的 N , 两路归并问题的被归并的总项数 N , 排序问题的被排序的序列的项数 N 等。但是可能出现多个规模参数。例如, 行数列数不相等的矩阵相乘: L 行 M 列乘 M 行 N 列 (L, M, N)。多规模参数有时也可以用一个参数来代表, 此例中用 $N' = LMN$ 来代表。

(2) 题目计算工作量 $W(N)$

需要注意的是, 题目计算工作量 $W(N)$ 一般与规模 N 有关。

(3) 题目按算法 C 计算的附加计算工作量 $W_{c\pm}(N)$

题目按算法 C 计算的计算工作量可能比最快的串行算法的计算量有所增减。 $W_{c\pm}(N)$ 是其增减部分。倒过来, 可以定义题目计算工作量 $W(N)$ 为串行最快算法 C_0 的计算工作量 $W_{c_0}(N)$ 。

(4) 题目按算法 C 计算的总计算工作量 $W_c(N)$

题目按算法 C 计算的总计算工作量

$$W_c(N) = W(N) + W_{c\pm}(N)$$

其中 $W(N)$ 为题目计算工作量, $W_{c\pm}(N)$ 为题目按算法 C 计算的附加计算工作量。

1.2 题目的计算时间 T

(1) 在计算机 M 条件下的题目按算法 C 计算的计算时间: $T_{c,M}(N)$

在计算机 M 条件下的题目按算法 C 计算的计算时间显然与计算机模型 M , 计算算法 C 以及题目的规模 N 有关。

(2) 在计算机 M 条件下的题目按算法 C 计算的计算速度 $V_{c,M}(N)$

在计算机 M 条件下的题目按算法 C 计算的计算速度

$$V_{c,M}(N) = W(N) / T_{c,M}(N)$$

注意, 这里的计算工作量是 $W(N)$, 而不是 $W_c(N)$ 。计算机 M 必定属于一种计算机模型。计算时间和速度只指出计算机模型是不够的, 还需要指出计算单位工作量所需的单位时间 t_0 。而计算单位工作量的单位时间 t_0 与具体的计算机有关。