

并行技术基础

主编 靳 鹏



1010101
01001111001
101010101
101011001
0011
001
1011001
011
111
1011001010100101010101

并行技术基础

主 编 靳 鹏
副主编 于 霞
田艳丰
段 勇

 吉林大学出版社

图书在版编目 (CIP) 数据

并行技术基础/ 靳鹏主编. —长春: 吉林大学出版社, 2011. 2

ISBN 978 - 7 - 5601 - 7027 - 5

I. ①并… II. ①靳… III. ①并行处理—基本知识 IV. ①TP274

中国版本图书馆 CIP 数据核字 (2011) 第 020548 号

书 名: 并行技术基础

作 者: 靳 鹏 主编

责任编辑、责任校对: 刘冠宏 卢 婵

吉林大学出版社出版、发行

开本: 880 × 1230 毫米 1/32

印张: 10.75 字数: 300 千字

ISBN 978 - 7 - 5601 - 7027 - 5

封面设计: 李岩冰

长春市利源彩印有限公司 印刷

2011 年 2 月 第 1 版

2010 年 2 月 第 1 次印刷

定价: 28.00 元

版权所有 翻印必究

社 址: 长春市明德路 421 号 邮编: 130021

发行部电话: 0431 - 88499826

网 址: <http://www.jlup.com.cn>

E-mail: jlup@mail.jlu.edu.cn

所谓并行技术，笼统地说是以并行计算机系统为核心的一系列科学技术的通称，包括并行计算机体系结构、并行操作系统、并行算法设计、并行程序设计语言、并行编程、并行数据库技术等。本书在上述几个方面对并行技术的基本框架进行了简要介绍。

此外，本书以较宽广的视角讨论了并行技术的发展轨迹，面向未来，提出了并行技术普及化的基本观点，并对此进行了初步论证，尤其是强调了并行操作系统的核心作用，给出了普通用户实现并行计算的几种途径，有一定的技术前瞻性，对普及并行技术有一定的帮助。

编者力求概念准确，论述严谨，内容新颖，图文并茂，可作为高等院校计算机专业本科生教材，也可供计算机专业研究生和相关专业的工程技术人员参考。

前 言

并行技术是当前计算科学领域最高端的学科之一。但近年来，随着工作站网络等技术的推广，以及消息传递技术的发展日益成熟，并行计算系统已经从传统的基础科学研究、国防等高端应用层面拓展开来，正逐渐走入大中型企事业单位甚至个人的视野中；随着 MPI 等基于消息传递的通用并行编程技术的广泛传播，介入该领域的应用门槛正逐渐走低，类似于个人电脑的普及，未来很可能催发出一个广阔的信息技术新市场；随着半导体集成度和功率之间矛盾的深化，导致摩尔定律的首次失效，当前处理器性能的评价标准转变为同等功率条件下的计算能力，进而随着计算芯片成本的不断降低，以及信息社会的计算需求和计算能力之间日益扩大的落差，并行技术向 IT 领域低端的发展动力十足，甚至多处理器技术在传统硬件上已经垄断了最低端的个人电脑领域，而且几十年以来一直在 IT 领域处于绝对主流地位的基于串行的软件开发体系，正面临颠覆性的挑战。

整个计算机世界面临再一次的重新洗牌，既是机遇又是挑战，在此背景下，普及并行技术正成为一种潮流和面向未来的选择。事实上，对信息技术比较关注的国家中，并行技术的普及工作早已开始，比较有代表性的如“将并行技术引入一年级课程”，该项目为美国国家科学基金项目，编号 DUE9554975。

本书的目标是，面向最广大普通计算机专业读者，用尽可能平实、通俗、简明的方式，把 IT 领域最高端的并行技术框架粗略地勾勒出来，为我国并行技术普及工作尽自己的力量。

最后简要介绍一下本书的内容及要点。首先，第一章是关于并行计算、并行处理、并行计算机的概括介绍，其核心内容是对 Amdahl 定律和 Gustafson 定律的理解。值得一提的是，这一章对于并行技术的发展历程和发展前景进行了讨论，在讨论之后，给出了本书的主旨，即并行技术即将迎来一次重要的发展机遇，正逐渐从绝对的尖端技术变成普及型技术，对于普通的计算机专业读者来说，普及并行技术有一定的必要性和可能性。第二章从计算机体系结构的角度对并行计算机的各种分类加以介绍，这是理解并行计算平台的基础。第三章介绍并行操作系统，实质上是从操作系统的角度介绍并行计算，具体来说是并行技术对于操作系统的要求，以及现代操作系统对于各类并行计算平台的支持。第四章介绍基础的并行算法知识。并行算法的内容在并行技术中占了很大的比例，而且并行算法的研究也是并行计算领域的研究前沿之一，但是本书作为一本入门教程，没有把重点放在这里，只是介绍了并行算法的一些基本设计思路。第五章并行编程技术是本书的重点内容之一，重点介绍了三种编程模式：共享主存、消息传递和数据并行。第六章以一种典型的并行程序开发方法，MPI 技术为核心，具体介绍了一些并行程序编程方法。到此为止，基本完成了最简单并行技术的知识框架，但是，计算机相关技术是高度实践性的知识，对于理论联系实际的要求很高，因此第七章为普通读者提供一种低成本的可以进行并行编程和实现并行计算的并行系统——Linux 机群，实质上是介绍了一种在普通微机环境中配置、安装 MPI 并行环境的一些方法。

本书编者靳鹏负责全书的统稿，并完成第一、二、三章，于霞负责第四章并行算法的编著，田艳丰负责第六章 MPI 编程部分的编写，段勇负责第五章并行算法和第七章 Linux 集群部分的编著。

目 录

第一章 并行技术概述	1
第一节 并行技术的定位	2
一、并行处理和并行计算	3
二、计算与计算科学	6
三、计算能力与计算需求	7
第二节 超级计算机	10
一、超级计算机的战略意义及其发展历史	10
二、全球超级计算机排名	11
三、超级计算机的发展史	13
第三节 并行计算的能力和效率	20
第四节 并行技术的主要实现形式	27
第五节 多核技术的挑战和机遇	35
第二章 并行计算系统的基本架构	41
第一节 计算机体系结构概述	42
一、计算机体系结构的概念	42
二、计算机组成的概念	43
三、计算机体系结构分类	43
四、SIMD 计算机	45
五、向量计算机	45
六、MIMD 计算机	47
第二节 MIMD 结构的并行计算机分类	49

并行技术基础

一、并行计算机组成的三要素	49
二、并行计算机互联网络的拓扑结构	54
三、MIMD 并行计算机的分类	60
第三节 并行计算机分类	61
一、访存模型	62
二、Cache 管理技术	64
三、并行计算机分类	66
 第三章 并行操作系统	74
第一节 操作系统回顾	74
一、操作系统	75
二、进程	76
三、线程	85
第二节 从操作系统角度看并行计算系统	89
一、网络操作系统	90
二、分布式操作系统	91
三、多处理器操作系统	93
第三节 多线程对多处理器并行系统的支持	96
一、多线程的实现	96
二、多线程的处理机调度	100
第四节 分布主存并行计算系统的任务分配和调度	102
一、典型的分布主存并行计算系统	103
二、分布主存并行系统处理器分配调度算法模型	108
三、进程间通信技术	108
第五节 虚拟存储技术	115
一、私有虚存	116
二、共享虚存	117
第六节 数据一致性管理	118
一、一致性的模式	118
二、存储器一致性模式与编程模型	125

三、DSM 系统的软件实现	126
第四章 并行算法基础	128
第一节 并行算法的度量	129
第二节 相关性分析	131
第三节 并行计算、并行编程模型与并行算法的关系	133
一、SPMD 和 MPMD 并行计算系统	133
二、具体问题的并行求解过程	134
三、并行编程模型简介	135
第四节 并行算法的设计	138
一、SPMD 和 MPMD 并行算法的设计	138
二、典型的并行算法设计思路	140
第五节 并行算法的分类及发展	143
第六节 关于矩阵乘法的典型并行算法	145
一、行列划分算法	146
二、Cannon 算法	147
第五章 并行编程基础	154
第一节 并行编程概述	155
一、并行编程的现状	155
二、并行编程为什么落后于串行编程的讨论	155
三、关于并行语言	156
四、并行编程模型和并行编程语言的分类及评价标准	157
第二节 进程的同构性、并行性	159
第三节 进程间的交互	160
一、进程交互简介	160
二、进程同步	161
三、进程通信的多种模式	162
第四节 共享存储器并行编程	163
一、Pthreads 简介	164

并行技术基础

二、程序举例	171
第五节 消息传递并行编程基础	176
第六节 数据并行编程模型	179
一、HPF 简介	180
二、HPF 的数据并行和数据映射	181
第七节 并行程序性能评价	184
一、浮点峰值性能与实际浮点性能	185
二、数值效率和并行效率	186
三、可扩展分析	188
四、串行程序的优化	189
五、并行程序性能优化	194
 第六章 MPI 并行编程	197
第一节 两个 MPI 程序及其相关分析	198
一、MPI 并行编程的三种模式	198
二、两段 MPI 例程	200
第二节 MPI 基础知识	208
一、MPI 的目标	208
二、MPI 的发展	210
三、MPI 的绑定	210
四、目前主要的 MPI 实现	211
五、MPI 程序的一些惯例	211
第三节 MPI 核心接口	212
一、MPI 接口参数说明	212
二、MPI 核心接口及相关示例	214
三、MPI 预定义数据类型	219
四、MPI 数据类型匹配	221
五、MPI 数据转换	223
第四节 MPI 消息	224
一、MPI 消息的组成	224

二、任意源和任意标识	225
三、MPI 通信域	225
第五节 简单 MPI 示例	226
一、用 MPI 实现计时功能	226
二、获取机器的名字和 MPI 版本号	229
三、是否初始化及错误退出	231
四、数据接力传送	233
五、任意进程间相互问候	236
六、任意源和任意标识的使用	239
七、编写安全的 MPI 程序	240
八、阶段小结	243
第六节 点对点通信函数与通信模式	245
第七节 聚合通信与同步	248
第八节 自定义数据类型	249
第九节 进程组与通信器	254
第十节 进程拓扑结构	255
第十一节 文件输入/输出	256
 第七章 Linux 机群	 262
第一节 构建 Linux 机群的要素	264
第二节 Linux 操作系统的安装与基本操作	266
一、Linux 系统的安装	267
二、基本使用与管理	270
第三节 Linux 下的程序开发环境	278
一、第一个 C 程序	278
二、Fortran 程序的开发	287
三、软件开发	293
第四节 MPICH 安装与程序编译、运行、调试	309
一、单机环境下 MPICH 的安装	309
二、机群环境下 MPICH 的安装	315

并行技术基础

第五节 安装配置基于 Linux 的并行计算集群.....	321
一、在单机上安装、配置 MPI 并行环境	321
二、在联网的多台机器上安装、配置 MPI 并行环境	323
三、专用并行机群系统的配置	328
参考文献	331

第一章 并行技术概述

并行技术是当前计算科学领域最高端的学科之一。但近年来,随着工作站网络等技术的推广及消息传递技术的发展,并行计算系统已经从传统的基础科学研究、国防等高端应用层面拓展开来,正逐渐走入大中型企事业单位甚至个人的视野中;随着 MPI 等基于消息传递的通用并行编程技术的广泛传播,介入该领域的应用门槛正逐渐走低,类似于个人电脑的普及,未来很可能催发出一个广阔的信息技术新市场;随着半导体集成度和功率之间矛盾的深化,导致摩尔定律的首次失效,当前处理器性能的评价标准转变为同等功率条件下的计算能力,进而随着计算芯片成本的不断降低,以及信息社会的计算需求和计算能力之间日益扩大的落差,并行技术向 IT 领域低端的发展动力十足,甚至多处理器技术在传统硬件上已经垄断了最低端的个人电脑领域,而且几十年以来一直在 IT 领域处于绝对主流地位的基于串行的软件开发体系,正面临颠覆性的挑战。

整个计算机世界面临再一次的重新洗牌,既是机遇又是挑战,在此背景下,普及并行技术正成为一种潮流和面向未来的选择。

2008 年 10 月在北京举行的第 13 届国际计算机棋类奥林匹克大赛上,经过激烈的角逐,程序 Many Faces of Go 以全胜战绩囊括了围棋项目 19×19 和 9×9 两个棋盘规模级别的冠军。而为这一世界冠军围棋程序提供超强马力的,正是运行在四台八核计算机上的 Windows HPC Server 2008 系统(HPC: High Performance Computing, 高性能计算)。2008 年 9 月 22 日,Windows HPC Server 2008 在华尔街 HPC 大会上发布。微软副总裁 Bill Laing 在会上发表主题讲话,向大家说明

并行技术基础

微软正努力致力于把高性能计算平民化、主流化,在众多合作伙伴的支持下也有实力完成这个目标。

回顾二十年前,活跃在市场上的超级计算机公司开发定制化的硬件、软件及开发工具,高昂的技术门槛令人望而却步。之后的几年间,计算机主流公司用精简指令芯片和自开发的定制化网络开始了竞争。1998年之后,商品化的芯片、标准网络再次把成本降低,直到现在多核技术已经普及到 PC 机的层面。

本书想说明的是,有一只看不见的手,正把高性能计算机从国家实验室里推到各个行业、企业,而且这个潮流越来越迅猛,对整个信息社会来说,挑战也越来越紧迫。

事实上,对信息技术比较关注的国家中,并行技术的普及工作早已开始,比较有代表性的如“将并行技术引入一年级课程”,该项目为美国国家科学基金项目,编号 DUE9554975。可以说,把握了并行技术的今天,就是把握了计算科学的未来。并行技术从“阳春白雪”成为大众化技术,对于我国这样的信息技术后发国家是一个难得的战略性机遇,抓住这个机遇,可以有效缩短我国与发达国家在信息技术上的代差,进而迎头赶上。本书希望在我国并行技术的普及化方面做出自己的贡献。

第一节 并行技术的定位

所谓并行技术,笼统地说是以计算机为核心的一系列科学技术的通称,包括并行计算机体系结构、并行操作系统、并行算法设计、并行程序设计语言、并行编程、并行数据库技术等。考察并行技术的定位,通常从计算科学发展的角度,考察并行处理技术在科学技术领域的定位。下面,本节从计算的角度出发,梳理其脉络,这也可以看做是并行技术的宏观定位。

一、并行处理和并行计算

并行处理(Parallel Processing),是计算科学的重要内容,是实现高性能计算(High Performance Computing)的重要途径,通常,高性能计算或超级计算(Super Computing)也称为并行计算(Parralled Computing),与之相关的,高性能计算机或超级计算机,也被称为并行计算机。

因此可以说,并行处理是并行技术的主体部分,它涉及的范围很广,包括并行结构、并行算法、并行操作系统、并行语言及其编译系统等,其研究目标是实现并行计算。下面把它们作为本节的考察目标,进行初步研究。

概况地说,并行处理是计算机科学与其他工程学科相结合的处于高速发展过程的交叉学科,其研究重点在于研究并挖掘各个计算层次的并行性。实质就是在并行计算平台上实现的信息的并行处理和计算。

从算法的角度看,我国高校计算机专业通常介绍的算法都是串行的,算法按照计算步骤进行组织,程序按照逐条指令的方式进行编写,而计算机则按照一系列的微操作指令执行。通常上述的串行模式可以解决问题,但是以并行方式让计算机同时执行多个操作可以极大地获得性能上的提高,这种提高在很多环境下是非常重要的。

通常提高计算性能(或称提高计算能力)的途径有两条:提高串行操作时硬件潜在的时钟频率,或让多个操作在多个处理器上并行执行。由于功率问题,在新的计算器件(如光信息处理技术)大规模应用之前,摩尔定律不可能永远执行下去,因此,显然后者更有前景。这一问题在后面的章节将得到更详细的说明,详见图1-1。

实现并行性必须有下列几个条件:计算方法必须涉及多个相互独立的操作;编程语言必须可以定义并行操作,或自动找出可以并行执行的操作;运行并行程序的计算机的体系结构必须能够同时执行多个并发操作。

可以举个例子说明上面的问题,求长度为1000的数据元素的和,代码片段如下:

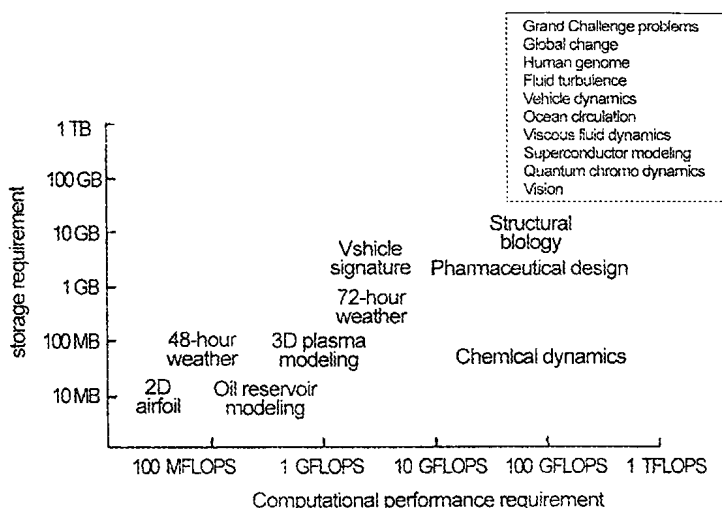


图 1-1 典型的计算需求

```
int a[ 1000 ];
for( int sum = 0, inti = 0; i < 1000; i + + )
    sum = sum + a[ i ];
这段程序等价于:
int a[ 1000 ]; //①
for( int sum1 = 0, inti = 0; i < 1000; i = i + 2 )
    sum1 = sum1 + a[ i ]; //②
for( int sum2 = 0, inti = 1; i < 1000; i = i + 2 )
    sum2 = sum2 + a[ i ]; //③
int sum = sum1 + sum2; //④
```

显然上述代码四条指令中的②和③没有相关性,可以独立执行,它们是算法开销的绝对主体,如果分别在两个处理器上并行工作,不考虑数据传输、同步等开销,理论上可使算法的时间复杂度降低约一半。当然,②和③并发执行算法还需并行计算系统和编程语言的支持。

因此,并行处理是并行计算机体系结构、并行算法设计、并行程序

设计语言相结合的产物,其目标是使计算机系统能够更快完成对应用问题的计算。更具体地说,就是要把算法的需求与计算机体系结构的性能结合起来,即实现并行计算。

描述并行处理是一个困难的工作,综合来看:高性能并行计算机的处理器数以千百计,处理器个数的增加为并行处理带来很多技术难题,再考虑并行计算机 CPU 速度、网络结构与性能、多级存储结构、操作系统性能等特征,研究适合某种并行计算的系统平台,并能在其上高效执行的并行新应用、新算法、新技术、新理论,充分提高解决问题的精度和速度,充分扩大求解的规模,充分降低求解的软硬件条件,对于大量的挑战性问题的数据处理、数值模拟、理论研究具有非常重要而紧迫的应用需求。

因此,构造适合当前计算系统平台的,可以并发执行的并行算法在计算科学和计算理论中非常重要,也可以说并行处理技术在提高计算性能、解决计算能力问题上具有决定性意义。

上文所述,并行处理的目标是实现并行计算。所谓并行计算(parallel computing)是相对于串行计算来说的,在并行计算机上,将一个应用分解成多个子任务,分配给不同的处理器,各个处理器之间相互协同,并行地执行子任务,从而达到加速求解速度,或者减小求解应用问题规模的目的。

并行计算分为时间上的并行和空间上的并行。时间上的并行就是指流水线技术,本书在后面章节将有介绍。而空间上的并行则是指用多个处理器并发地执行计算任务,为了成功开展并行计算,必须具备三个基本条件:

首先,使用并行计算机。并行计算机至少包含两台或两台以上处理机,这些处理机通过互连网络相互连接,相互通信。

其次,应用问题必须具有并行度。也就是说,应用可以分解为多个子任务,这些子任务可以并行地执行。将一个应用分解为多个子任务的过程,称为并行算法的设计。这是并行处理的核心任务之一。

最后,并行编程。在并行计算机提供的并行编程环境上,具体实现并行算法,编制并程序,并运行该程序,从而达到并行求解应用问