



普通高等教育“十三五”规划教材

PUTONG GAODENG JIAOYU “13·5” GUIHUA JIAOCAI

超算、云计算与大数据技术专业教程

多核并行高性能计算 OpenMP

雷洪 胡许冰 编著

Frotran C C++

Open Multi-Processing Message Passing Interface

Cache

Parallel Virtual Machine

High Performance

OMP_DYNAMIC OMP_SCHEDULE OMP_NUM_THREADS OMP_NESTED OMP_STACKSIZE

Cache

Parallel Virtual Machine

High Performance

Frotran C C++



冶金工业出版社
www.cnmp.com.cn



普通高等教育“十三五”规划教材

超算、云计算与大数据技术专业教程

多核并行高性能计算 OpenMP

雷 洪 胡许冰 编著

北 京

冶 金 工 业 出 版 社

2016

内 容 提 要

本书主要介绍了共享内存并行编程 OpenMP 的基本原理,采用实例方式讲解在 Fortran 语言环境中 OpenMP 并行程序的编写和运行,并综合评述了高性能计算编程中遇到的常见问题和解决方案。本书面向实际应用,简洁易学,使读者能够亲身感受到并行计算的魅力。

本书为计算机专业本科教材、高性能计算领域的理工科高年级本科生和研究生的并行计算课程教材,也可以作为从事并行计算研究、设计和开发的教师和工程师的专业参考书。

图书在版编目(CIP)数据

多核并行高性能计算: OpenMP/雷洪, 胡许冰编著. —北京:
冶金工业出版社, 2016. 5

普通高等教育“十三五”规划教材

ISBN 978-7-5024-7249-8

I. ①多… II. ①雷… ②胡… III. ①并行程序—程序设计—
高等学校—教材 IV. ①TP311. 11

中国版本图书馆 CIP 数据核字(2016) 第 105741 号

出 版 人 谭学余

地 址 北京市东城区嵩祝院北巷 39 号 邮编 100009 电话 (010)64027926

网 址 www.cnmp. com. cn 电子信箱 yjcs@cnmp. com. cn

责任编辑 刘小峰 杜婷婷 美术编辑 吕欣童 版式设计 吕欣童

责任校对 李 娜 责任印制 李玉山

ISBN 978-7-5024-7249-8

冶金工业出版社出版发行; 各地新华书店经销; 固安华明印业有限公司印刷

2016 年 5 月第 1 版, 2016 年 5 月第 1 次印刷

787mm × 1092mm 1/16; 17. 75 印张; 429 千字; 272 页

49. 00 元

冶金工业出版社 投稿电话 (010)64027932 投稿信箱 tougao@cnmp. com. cn

冶金工业出版社营销中心 电话 (010)64044283 传真 (010)64027893

冶金书店 地址 北京市东四西大街 46 号(100010) 电话 (010)65289081(兼传真)

冶金工业出版社天猫旗舰店 yjgycbs. tmall. com

(本书如有印装质量问题, 本社营销中心负责退换)

前 言

近年来，大规模集成电路的迅猛发展，促进了以多核处理器为基础的高性能计算机的飞速发展，与之休戚相关的并行计算技术也得到了长足进步。高性能计算机是一个国家经济和科技实力的综合体现，也是促进和提高经济、科技、国防安全水平的重要工具，已经成为世界强国争相发展的战略制高点。一个国家的高性能计算技术水平，取决于计算机专业人才，也取决于广大科技工作者的应用。现阶段计算应用和数据处理的状况是：大多数工程技术人员所面对的是计算服务器、多核微机这样的硬件系统，设备潜力没有被充分挖掘；同时，大多数工程计算是多物理场的耦合计算问题，计算量相对来说并不巨大。大多数科技工作者往往更加关注自身行业的技术发展，而对并行计算技术缺乏深入了解。充分发挥计算机技术的优势，让工程技术人员从高性能计算中获益，是一个急需解决的实际难题。因此，要求适合工程技术人员进行并行计算的软件系统必须简单、易学并且具有较高的并行效率。

作者长期从事钢铁冶金过程的数值模拟，由于科研工作需要，一直关注于并行计算技术的发展。由于现有相关著作大多数是由计算机专家编著，著作中大量的计算机专业术语难以被其他专业工科学生和广大科技工作者所理解，而且大多数著作是以 MPI 作为基本的并行计算语言进行推广。但因 MPI 存在学习困难、调试困难等诸多问题，使并行计算技术一直未能在工科专业普及。近年来，多核计算机和共享内存编程 OpenMP 的出现，使并行计算在工科专业的推广成为可能。为此，作者编写本书，目标是给从事较大规模工程计算的工程师提供一本多核心体系结构计算系统的实用参考书，为我国并行计算普及工作略尽绵薄之力。

本书分析了当前流行的并行计算技术，从中遴选出适合大多数工程科技人员应用的 OpenMP 并行计算技术，从实例入手阐明程序运行过程，清晰而简洁地展示 OpenMP 并行计算原理、编程特点和方法。同时，结合工程实际，采用实例剖析高性能计算程序的完整实现途径。希望本书能够帮助工程技术人员了解并掌握并行计算知识，并在实际工作和学习中顺利运用，从而避免作者在获取这些经验时所犯的类似错误。

随着高性能计算技术迅速发展,国家急需具备并行计算专业技能的高科技人才,已有高校开设云计算与大数据处理技术专业方向。本书可以作为高等院校计算机专业 OpenMP 并行计算应用课程的教材,也可以作为其他理工科专业并行计算和数据处理技术的教材。

Fortran 是一门历史悠久的计算机语言,在数值计算中最为常用,已积累了非常丰富的计算程序。充分利用这些已有的计算资源,可以事半功倍地完成好高强度的科研任务。但相对于 C 语言而言,有关 Fortran 编程的著作较少,而 Fortran 方面的并行计算著作则更为罕见。鉴于此,本书以 Fortran 语言作为主要编程语言,兼顾 C/C++ 语言,介绍 OpenMP 的基本用法。实际上,OpenMP 编程在 Fortran 语言和 C/C++ 语言方面是互通的,本书的大多数程序稍加修改即可应用于 C/C++ 语言。

全书共分为 10 章和 7 个附录。主要内容如下:

第 1 章概述并行计算的发展历程,介绍并行计算概念和 Fortran 的发展历史。

第 2 章阐述 OpenMP 的语法,掌握 Fortran 程序编写、编译和执行的完整过程。

第 3 章阐明数据环境,研究共享变量和私有变量,全局变量和局部变量的联系和差异。

第 4 章讲解并行区域的构造方法,探讨线程组和子线程数量的确定方式。

第 5 章剖析不同并行结构的差异,实现负载平衡。

第 6 章揭示多线程同步的不同机制,防止数据竞争的出现。

第 7 章明晰运行环境要素,探究不同的时间函数和锁函数差异,避免死锁的发生。

第 8 章解析 OpenMP3.0 的重要指令 TASK,领悟 OpenMP3.0 的新特征。

第 9 章讨论循环的并行方案,体验提高任务粒度的途径。

第 10 章研讨高性能程序方案,商榷程序调试方法,确定程序优化措施。

附录给出 Linux 下常用命令及程序编辑和调试方法,了解 Fortran 的安装和编译优化。

本书各章节相互独立,部分内容略有重复,供读者根据需要选择阅读。阅读本书之前需对 Fortran 语言编程有所了解,才能深入体会 Fortran 语言在并行计算方面的优势。

阅读本书时,如果您对高性能计算感兴趣,建议阅读第 1 章。如果您是初

学者,那么需要关注第1章中编译器部分、第2章和附录1至附录4。

本书的核心部分是第3章到第9章,其中第3~6、8章是 OpenMP 的编程基础,学习之后基本能够编写大部分的 OpenMP 程序。第7章锁操作十分复杂,且 OpenMP2.5 和 OpenMP3.0 之间存在较大差异,因此读者需注意自己实际使用的编译器版本。第8章是 OpenMP3.0 的精华。而第9章则是 OpenMP 应用中常见问题的解决方案。

如果您想对串程序 and 并行程序进行优化,进行高性能计算,那么需要关注第10章及附录5和附录6。如果您是 C/C++ 用户,那么需要关注附录7,并根据其内容查阅本书相关章节。

东网科技有限公司胡许冰老师为本人学习和应用 OpenMP 提供了很多指导和帮助。胡许冰老师有超过20年的IT架构和系统设计经验,主导东网科技高性能计算、云计算和空间信息等平台产品的研发和东北区域超算中心的设计建设工作,是国内云计算和大数据领域的资深技术专家,在图像处理、分布式文件系统、数值计算以及机器学习领域有着深厚的技术积累。同时,深深感谢东北区域超算中心提供的计算平台及开发测试环境。

在本人学习和应用 OpenMP 的过程中,得到了许多老师和同行的帮助,在此特向东北大学计算中心刘小峰老师、大连理工大学张永彬博士、东网科技有限公司王兴老师、英特尔亚太研发有限公司黄飞龙工程师和周姗工程师表示由衷的感谢。在本书的撰写过程中,参考了其他同行的文章、课件和研究资料,还引用了许多专家 and 学者的工作。在此一并感谢并向他们所做的工作表示深深的敬意。本书中的所有程序代码可登录冶金工业出版社网站 www.cnmp.com.cn 下载。

本书的出版得到了辽宁省百千万人才工程培养经费(2013921073)和国家自然科学基金委员会-宝钢集团有限公司钢铁联合研究基金(U1460108)的资助。在书稿准备与出版过程中,冶金工业出版社的编辑人员也给予了大力支持,在此一并表示感谢。

尽管作者在编写本书的过程中投入了大量的精力,但受计算机专业水平所限,书中难免存在不当之处,恳请专家和读者给予批评指正。

雷洪于东北大学
2016年4月

目 录

1 并行计算概论	1
1.1 多核 CPU	2
1.2 并行计算与分布式计算	2
1.3 并行计算机的种类	3
1.3.1 CPU 与存储器的连接方式	3
1.3.2 数据的通信方式	4
1.3.3 指令和数据之间的工作方式	5
1.4 并行编程模式	5
1.4.1 共享内存模式	6
1.4.2 消息传递模式	7
1.4.3 数据并行模式	8
1.5 OpenMP 和 MPI 的特点	10
1.6 并行计算中常用概念	11
1.6.1 程序、线程、进程和超线程	11
1.6.2 单核编程和多核编程	12
1.6.3 多线程编程和多进程编程	13
1.6.4 并行算法评价	13
1.7 OpenMP 多核编程	15
1.7.1 OpenMP 的历史	15
1.7.2 OpenMP 的特点	16
1.8 科学计算领域语言的选取	16
1.9 Fortran 发展历史	18
1.9.1 Windows 系统	18
1.9.2 Linux 系统	18
1.9.3 Fortran 程序的编译和执行	19
1.10 小结	21
练习题	21
2 OpenMP 编程简介	22
2.1 编译指导语句	23
2.2 并行执行模式	23
2.2.1 编译指导语句格式	24

2.2.2 主要指令	25
2.2.3 主要子句	26
2.2.4 指令和子句的配套使用	27
2.3 头文件	28
2.4 常用库函数	28
2.5 最简单的并行程序	29
2.6 小结	33
练习题	33
3 数据环境	34
3.1 PRIVATE 子句、SHARED 子句和 DEFAULT 子句	34
3.2 FIRSTPRIVATE 子句和 LASTPRIVATE 子句	38
3.3 THREADPRIVATE 子句	41
3.4 COPYIN 子句和 COPYPRIVATE 子句	45
3.5 REDUCTION 子句	49
3.6 伪共享	53
3.7 小结	54
练习题	54
4 并行控制	56
4.1 PARALLEL 指令	56
4.2 设定线程数量	59
4.3 默认模式	59
4.4 静态模式	60
4.5 动态模式	61
4.6 嵌套模式与 NUM_THREADS 子句	63
4.7 IF 子句 (条件并行)	65
4.8 小结	67
练习题	68
5 并行构造	69
5.1 负载均衡	69
5.1.1 静态负载均衡	69
5.1.2 动态负载均衡	70
5.2 DO 指令	71
5.2.1 循环依赖	73
5.2.2 单重循环	78
5.2.3 嵌套循环	80
5.2.4 循环工作量的划分与调度	83

5.3	SECTIONS 指令	90
5.4	WORKSHARE 指令	93
5.5	SINGLE 指令	96
5.6	小结	98
	练习题	98
6	线程同步	99
6.1	互斥锁机制	99
6.2	事件同步机制	100
6.3	BARRIER 指令	100
6.4	NOWAIT 指令	102
6.5	MASTER 指令	104
6.6	CRITICAL 指令	106
6.7	ATOMIC 指令	108
6.8	ORDERED 指令	110
6.9	FLUSH 指令	113
6.10	小结	115
	练习题	115
7	运行环境	117
7.1	环境变量	117
7.1.1	OMP_DYNAMIC	117
7.1.2	OMP_SCHEDULE	118
7.1.3	OMP_NUM_THREADS	118
7.1.4	OMP_NESTED	118
7.1.5	OMP_STACKSIZE	118
7.1.6	环境变量的设置方法	119
7.2	库函数	120
7.2.1	运行环境操作函数	120
7.2.2	OpenMP 时间函数	121
7.2.3	Fortran 常用时间函数	123
7.2.4	锁函数	129
7.3	小结	139
	练习题	139
8	OpenMP 3.0 新特征	140
8.1	任务	140
8.1.1	任务结构	140
8.1.2	任务特征	141

8.1.3 任务类别	141
8.1.4 任务同步	143
8.1.5 DO 指令、SECTIONS 指令和 TASK 指令	143
8.1.6 TASK 指令与递归算法	151
8.2 COLLAPSE 子句	154
8.3 锁拥有者的变迁	156
8.4 小结	157
练习题	157
9 应用实例	158
9.1 循环的并行	158
9.1.1 单重循环	159
9.1.2 多维数组和嵌套循环	167
9.2 粗粒度的设置	171
9.2.1 IF 子句	171
9.2.2 if 语句	173
9.2.3 方案总结	175
9.3 全局变量和局部变量	175
9.3.1 common 定义	176
9.3.2 module 定义	179
9.3.3 全局变量和局部变量、共享变量和私有变量	181
9.3.4 私有变量和段错误	182
9.4 小结	183
练习题	183
10 高性能计算程序的实现途径	185
10.1 硬件条件和操作系统	186
10.2 科学问题算法的优化	186
10.2.1 数学模型	187
10.2.2 求解方法	188
10.3 串程序的编写	190
10.4 常见的调试器	191
10.5 高性能程序的优化步骤	191
10.6 串程序的正确性调试	193
10.6.1 程序的错误类型	193
10.6.2 计算程序中常见错误	193
10.6.3 静态安全检查	194
10.6.4 动态安全检查	198
10.6.5 IDBC 串行调试	200

10.7 程序热点的确定	206
10.7.1 编译器热点分析报告	206
10.7.2 手动热点分析报告	211
10.8 串行程序的优化	215
10.8.1 循环变换	216
10.8.2 向量化	219
10.8.3 Intel Fortran 常用优化策略	230
10.9 并行程序的优化	236
10.9.1 性能提升的预估	236
10.9.2 并行优化步骤	236
10.9.3 向导自动并行化 (GAP)	237
10.9.4 优化技术	238
10.9.5 自动并行化	238
10.9.6 并行调试策略	239
10.9.7 IDBC 并行调试	240
10.10 小结	245
练习题	245
附录	246
附录 1 常用的 Linux 命令	246
附录 2 Linux 下的文本编辑器 vi	249
附录 2.1 vi 的基本概念	249
附录 2.2 vi 的主要操作	249
附录 3 Intel Fortran 安装	252
附录 4 常用的 GDB 命令	256
附录 5 Linux 环境下 Intel Fortran 常用编译方案	259
附录 6 Intel Fortran 常用编译开关	261
附录 7 C 和 C++ 语言中 OpenMP 常见用法	266
附录 7.1 语法格式	266
附录 7.2 头文件	266
附录 7.3 指令类	266
附录 7.4 子句类	268
附录 7.5 库函数	269
附录 7.6 最简单的 C/C++ 并行程序	270
附录 7.7 OpenMP 并行程序的编译和执行	270
参考文献	271

计算机是人类解决大负载科学计算问题任务的强有力工具。随着微电子工业的发展,计算机中央处理器(CPU)的运行速度越来越快,一方面,随着计算机硬件的不断提高,人们解决问题所需的时间消耗越来越少;另一方面,人们希望求解的问题规模^[1]不断扩大,现有的计算能力远远不能满足需要。这样,单纯依靠一颗CPU来解决这些计算问题是不现实的,必须使用多颗CPU来共同完成一个计算任务。这就是并行计算的由来。

并行计算的优点是具有强大的数值计算和数据处理能力,能够被广泛地应用于国民经济、国防建设及科技发展中具有深远影响的重大课题,如石油勘探、地震预测、天气预报、新型武器设计、天体和地球科学等等。而并行计算系统既可以是专门设计的、含有多颗CPU的超级计算机,也可以是以某种方式互连的若干台独立计算机构成的集群。在这样的背景下,对编程人员提出了更高的要求。通常,编程人员需要对现有的串行程序进行修改,对CPU之间的通信和控制进行协调从而解决并行程序所带来的数据竞争、同步等潜在问题,实现并行程序的高稳定性和高并行加速比。

并行计算离不开硬件和软件两大系统,如图1-1所示。硬件系统中CPU和存储器的连接形式决定了科技人员采用的并行方式。绝大多数科技人员使用的是个人计算机、服务器和 workstation,这一类的硬件系统均可采用 OpenMP 进行并行。

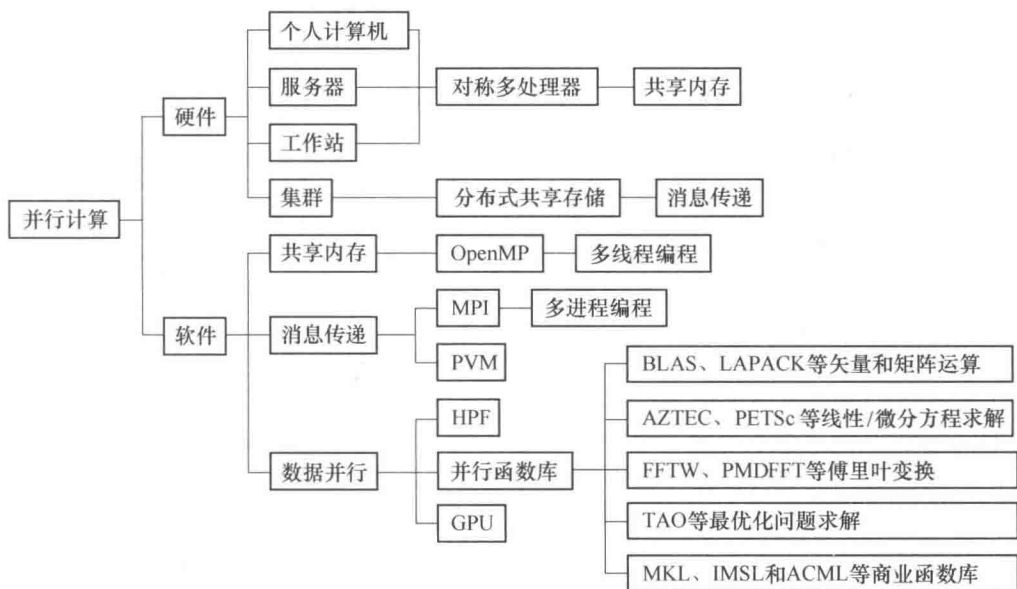


图 1-1 并行计算所需硬件和软件

1.1 多核 CPU

计算机运算速度的提高能够有效地提高科研人员的工作效率。在过去的几十年里,个人计算机 CPU 的主频一直依照摩尔定律发展。但是当单核 CPU 的主频达到 3GHz 以后,过高的功耗和高散热问题成为瓶颈限制了 CPU 频率的提高。虽然单核 CPU 的性能可能接近了极限 4GHz,但是多媒体、大规模科学计算等多个应用领域却对处理器性能不断地提出了更高的要求。现代电子工业的发展使芯片上晶体管的密度仍可以不断增加,于是各主流处理器厂商将产品战略从提高芯片的时钟频率转到了多内核的研发。因此,多核处理器的出现是应用需求和科技进步的时代产物。

多核处理器,又称为片上多处理器或单芯片多处理器(Chip Multi-Processor,简称为 CMP),是指在一个芯片上集成多个处理器核,而各种处理器核一般都具有固定的逻辑结构:指令级单元、执行单元、一级缓存(L1)、二级缓存(L2)、存储器及其控制单元、总线接口等。多核处理器的出现和发展十分迅猛,仅经历了 20 年的历史。1996 年,美国 Stanford 大学首先提出片上多处理器和首个多核结构原型。2001 年,IBM 公司推出第一个商用多核处理器 POWER4。2005 年,Intel 和 AMD 多核处理器的大规模应用,使多核成为市场主流。目前常见的多核芯片有 2 个、4 个、6 个或 8 个核,并且核的数目随着新一代 CPU 的出现而不断增加^[2-4]。

多核处理器利用在一个芯片上集成的多个功能完整的核心可以对外提供统一服务并提高程序的并行性,它具有如下优点:

(1) 多核 CPU 利用集成的多个 CPU 核心可以成倍地提高整个处理器同时执行的线程数,从而提升处理器的并行性能,更有效地执行各项计算任务,提高计算性能。

(2) 多个核心集成在一个芯片上,极大地缩短了核心之间的互连线,降低了核心之间的通信延迟,同时提高了通信效率和数据传输带宽。

(3) 多核结构可以有效地共享芯片上的资源,动态地调整频率和电压,提高资源利用率,降低功耗。

(4) 多核处理器结构比较简单,采用成熟的单核处理器作为处理器核心,设计和验证周期短,优化容易,扩展性强,可以有效地降低研发成本。

1.2 并行计算与分布式计算

并行计算的目的是利用多个 CPU(或多个 CPU 核)的协同来求解同一个问题,从而实现由于单核 CPU 计算能力或内存容量限制而无法提供的性能。它的实质是将一个待求解的问题分解成若干个子问题,各个子问题均由独立的 CPU 同时进行计算。这样,各 CPU(或 CPU 核)在并行计算过程中往往需要频繁地交换数据,具有细粒度和低开销的特征。并行计算的重要特征是短的执行时间和高的可靠性,它主要是指以高精度浮点运算为主的科学计算。

而分布式计算的目的是提供方便,这种方便性主要体现在可用性、可靠性以及物理分布三个方面。在分布式计算中,CPU(或 CPU 核)之间并不需要频繁地交换数据,具有

粗粒度的特征。分布式计算的重要特征是长的正常运行时间，它主要是指以整数运算为主同时具有少量简单的浮点运算的事务处理型计算。

在实际应用中，并行计算与分布式计算是十分相似的概念。它们之间的界限十分模糊，并行计算与分布式计算都是指在同一时刻同时有若干个指令序列（或指令集合）在运行。在单核单处理器计算机设备中，这种同时处理实际上取决于操作系统的调度，它是通过时间片轮转执行机制来实现的。而在多核心或多处理器设备中，操作系统通过将不同的任务调度到不同的 CPU（或 CPU 核心）上从而实现真正的同时执行^[5]。

1.3 并行计算机的种类

并行计算机通常包含多颗自带高速缓存（Cache）的 CPU，而且这些 CPU 需要一定数量的内存才能工作。同时，这些 CPU 通常需要借助于网络传递数据从而实现 CPU 之间的协同工作。因此，并行计算机通常可分为三大部件：CPU、存储器和网络。通常，并行计算机的种类可通过 CPU 与存储器的连接方式、数据的通信方式以及指令和数据之间的工作方式来进行划分^[3,6-8]。

1.3.1 CPU 与存储器的连接方式

根据存储器与 CPU 的连接方式可分为共享存储系统和分布存储系统^[2,9]。在共享存储系统中，所有 CPU 共同使用同一个存储器和输入输出（I/O）设备，并且一般通过总线连接，如图 1-2 所示。这种方式适合于实验室常见的计算服务器系统。目前一般采用 2 颗或 4 颗多核 CPU。在共享存储系统中，内存空间是统一编址的，可以被 CPU 所共享；CPU 之间数据通信依靠 CPU 对具有相同地址的内存单元的访问来实现。但是当多颗 CPU 对同一地址的内存单元进行读写操作时，会出现访问冲突，即数据竞争。这种连接方式一般用于小型多 CPU 系统。如果采用多总线，CPU 的数量一般不超过 16 个。

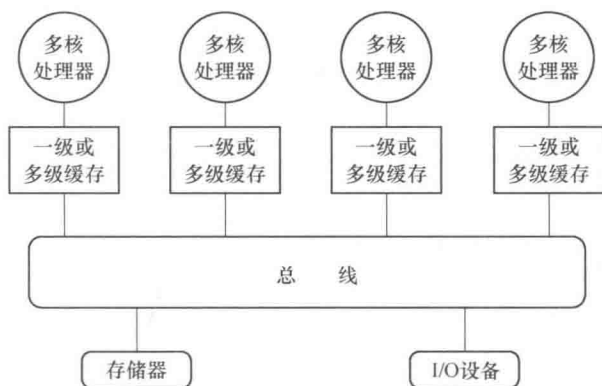


图 1-2 共享存储系统的基本结构

图 1-3 给出了分布存储系统的结构。通常，每颗 CPU 均具有各自的存储器和输入输出设备，它们组成了一个计算节点；多个计算节点通过网络相互连接形成了分布存储系统。这种方式适合于实验室常用的集群系统。由于每颗 CPU 都有自己的存储器，因此可以保

证 CPU 访问存储器速度, 不会出现访问冲突。另外, 在网络中增加计算节点比较方便, 即系统的可扩展性能好。但是各节点间必须借助于网络相互通信, 因此数据通信比较困难, 必须借助于专门的通信方法。

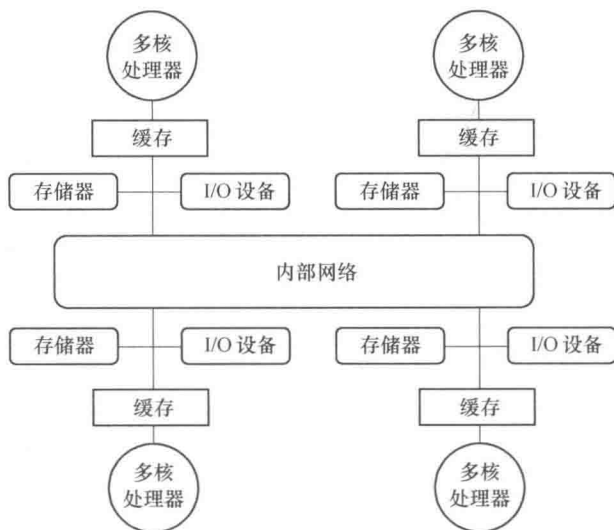


图 1-3 分布存储系统的基本结构

1.3.2 数据的通信方式

根据数据通信方式, 可以将并行计算系统分为共享地址空间系统和消息传递系统两大系统。

在共享地址空间系统中, 存储器的地址空间是统一的, 因此可称为单地址系统或共享存储多处理器 (Shared Memory Multiprocessors, 简称 SMM) 系统。根据 CPU 与存储器的连接方式可将共享存储器多处理器系统进一步进行分类。如果存储器是集中式的, 那么所有的处理器能够以相同的速度访问内存, 这种系统称为对称共享内存多处理器系统 (Symmetric Shared-memory Multiprocessors 或 Symmetric Multiprocessors, 简称 SMP) 或均匀存储访问系统 (Uniform Memory Access, 简称 UMA)。如果内存是分布式的, 那么 CPU 访问内存的速度就与内存的位置有关。毫无疑问, CPU 访问本地内存的速度最快。换言之, 由于处理器访问内存的速度是不一样的, 因此称为分布式共享内存系统 (Distributed Shared-Memory, 简称 DSM) 或非均匀存储访问系统 (Nonuniform Memory Access, 简称 NUMA)。

在消息传递系统中, 每个计算节点都是一个独立的计算机系统, 而每个节点的存储器均单独编址, 因此同一个地址对应于多个存储器。这样, 节点间数据的传递不能通过本地节点的处理器直接访问其他节点的存储器来实现, 而必须通过节点之间相互发送含有数据信息的信息来实现。这种通过发送包含数据的信息来实现数据通信的系统称为消息传递系统。它可分为大规模并行处理机系统 (Massively Parallel Processor, MPP) 和集群系统 (Cluster)。大规模并行处理机系统是指由几百或几千台处理机组成的规模并行计算系统。此系统的很多硬件设备是专门设计制造的, 它的网络传输速度较高但扩展性稍差, 开

发十分困难,通常标志着一个国家的综合实力。而集群系统是相互连接的多个同构或异构的独立计算机的集合体,节点之间通过高性能互联网相连接。每个节点都有自己的存储器、I/O 设备和操作系统,可以作为单机使用;并行任务的完成则需通过各节点之间的相互协同工作来完成。近 10 年以来,集群系统以高性价比、高可扩展性和结构的灵活性在多个领域得到了广泛应用。

1.3.3 指令和数据之间的工作方式

根据指令和数据之间的工作方式可分为四大类^[2,3]。第一类是单指令流单数据流系统 (Single Instruction Single Data, SISD), 具有一个单处理器核的个人计算机可归为此类。第二类是单指令多数据流系统 (Single Instruction Multiple Data, SIMD), 它是指在多颗 CPU 上运行相同的指令,但是每颗 CPU 所处理的数据对象并不相同。今天的图形处理器 (Graphics Processing Unit, GPU) 已经不再局限于 3D 图形处理了,它在通用计算技术的发展可归为 SIMD 类。第三类是多指令单数据流系统 (Multiple Instruction Single Data, MISD), 在实际应用中,这种系统是不存在的。第四类是多指令多数据流系统 (Multiple Instruction Multiple Data, MIMD), 它是指每颗 CPU 上执行的指令和处理的数据各不相同。目前常见的多核个人计算机和集群计算机可归为此类。

1.4 并行编程模式

并行编程是使用程序语言显式地进行说明,从而实现将计算任务中不同部分分配给不同的 CPU 同时执行。并行编程模式按通信方法可分为共享内存模式、消息传递模式和数据并行模式^[2,10]。

目前,工程技术人员常用的编程语言是消息传递接口 (Message Passing Interface, 简称 MPI) 和直接控制共享内存式并行编程的应用程序接口 (Open Multi-Processing, 简称 OpenMP)。当采用 MPI 进行并行化计算时,每个进程都有各自独立的存储器。当进行全局共享数据的读写操作时,需进行计算机之间的通信从而实现数据的搬运。因此 MPI 需要明确划分数据结构并重构源程序,编程困难并且开发周期长;但是它具有较好的可移植性和扩展性,以及较高的并行效率。OpenMP 是 1997 年 10 月由计算机硬件和软件厂商联合发表的共享内存编程应用程序接口的工业标准协议。它主要是针对循环进行并行,能有效地克服并行编程的可移植性和扩展性能差的缺点。近年来,实验室用微机、工作站和服务器纷纷采用多 CPU 共享内存技术,为研究者使用以 OpenMP 为基础的并行计算方法提供了必要的硬件条件。

一个成功的并行程序应具备如下特征:

(1) 正确性。如果将串行程序并行后得到的结果与串行结果存在较大的差异,则串行程序的并行化也就失去了意义。当然在计算过程中,由于截断误差、随机数的调用等因素在某些情况下会造成并行程序与串行程序结果存在细小差异,这是允许的。因此,并行计算结果与串行计算结果的比较是并行编程中的重要一环。

(2) 高性能。并行计算的一个重要目标是追求较短的计算时间。如果并行计算时间大于串行计算时间,并行计算也就失去了意义。并行程序的计算性能的衡量指标一般采用并行加速比和并行效率两个重要参数。

(3) 可扩展性。以前,用户是针对个人使用的双核 CPU 微机进行编程。但是随着硬件的发展,八核 CPU 或更多核的 CPU 也会相继出现。在硬件快速变化的情况下,用户迫切希望不要因为新硬件的出现而不得不大幅度地修改并行程序。要避免此状况的出现,在程序设计开始就应考虑到可扩展性。这样,所编写的并行程序将具有较长的生命周期,从而极大地节省人力和财力。

目前在实验室比较常见的计算系统,大体上可以分为两类:一类是共享内存系统(SMP),例如个人计算机、工作站和服务器的,其特点是多颗 CPU 拥有物理上共享的内存;一类是分布存储系统(DMP),如集群系统,其特点是系统由多个物理上分布的计算结点组成,每个计算结点拥有自己的内存,结点之间通过高速以太网或专用高速网络连接。它们各自的特点见表 1-1。

表 1-1 实验室常见的计算系统特点及并行计算方式

计算系统	个人计算机	集 群	服务器和工作站
硬件系统	单一主机,单颗 CPU,每颗 CPU 有多个核心	多台主机,有各自的一颗或多颗 CPU,每颗 CPU 有多个核心	单一主机,多颗 CPU,每颗 CPU 有多个核心
操作系统	单一	多个	单一
高性能计算系统	对称多处理器(SMP)	分布式共享存储(DMP)	对称多处理器(SMP)
常用并行开发模式	共享内存模式	消息传递模式	共享内存模式

并行程序的编程模型、运行环境、调试环境等都要比串行程序复杂得多。提供良好的高性能计算开发环境,一直是学术界和工业界所追求的目标。

1.4.1 共享内存模式

共享内存存储,是指对多颗 CPU 都访问一个共享存储器。在图 1-4 中,计算系统中共有 4 颗 CPU,每颗 CPU 有 4 个核心。因此,整个计算系统共有 16 个核心,这些核心均能够访问(进行读定操作)内存中的同一个位置(变量的值)。

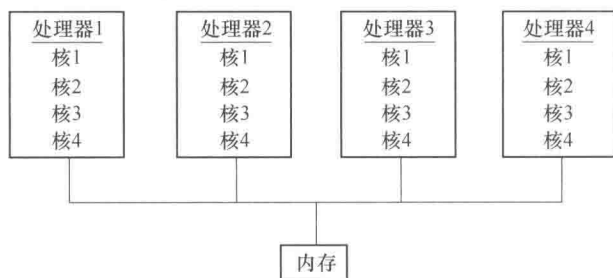


图 1-4 共享内存存储系统示意图

在共享内存模型中,一个并行程序由多个共享内存的并行任务组成,数据的交换通过隐式地使用共享数据(即线程间的通信通过对共享内存的读写操作)来完成。在大多数情况下,此编程模式的主要任务是对循环进行并行处理,而计算与数据的划分和任务之间的通信则由编译器自动完成。