

并行计算系列丛书

并行算法实践

陈国良 安虹 陈旻 编著
郑启龙 单久龙

高等教育出版社

(京)112号

内 容 提 要

本书是并行计算系列丛书之四,旨在介绍并行程序设计的有关知识和并行算法的具体编程实现。

本书从内容安排上分为上篇和下篇。其中,上篇为并行程序设计导论,主要包括并行程序设计基础(并行计算机系统与结构模型、PC机群搭建和并行程序设计简介等)、并行程序编程指南(MPI、PVM、HPF和OpenMP等)和并行程序开发方法(可视化并行程序设计环境、并行程序调试和并行程序性能分析及优化等);下篇为并行算法编程实现,主要包括非数值并行算法(排序、串匹配、图论、组合优化和计算几何等)及其MPI编程实现和数值计算并行算法(矩阵运算、线性方程组求解、矩阵特征值计算和傅氏及小波变换等)及其MPI编程实现。书后所附的光盘中包含了第 单元和第 单元中所有并行算法的MPI源程序。

书中内容精炼、实用,体现了并行算法的设计与实现相结合,可作为高等学校计算机及相关专业的本科高年级学生和研究生的教材,其中上篇和下篇也可分别作为“并行程序设计导论”和“并行算法编程指南”单独使用。本书也可供从事并行程序设计及其实现的科技人员参考阅读。

www.Chinaren.com

作者介绍

陈国良,中国科学技术大学教授,博士生导师,中国科学院院士,1938年6月生于安徽省颖上县,1961年毕业于西安交通大学无线电系计算机专业。中国科学院院士,1981—1983年在美国普度大学作访问学者,1984年至今曾多次应邀赴东京大学、普度大学、澳大利亚国立大学、新南威尔士大学、昆士兰大学、格里福斯大学、堪萨斯城市大学、依阿华大学、威斯康星大学、Maharish国际大学、香港理工大学、澳门大学、北京大学、国防科技大学等讲学交流。现任国家高性能计算中心(合肥)主任,国际高性能计算(亚洲)常务理事,中国计算机学会开放系统专业委员会副主任,中国数学会计算数学并行计算专业委员会委员。曾任国家教育部高等学校计算机科学与技术教学指导委员会副主任,全国高等教育电子、电工和信息类专业自考指导委员会副主任,安徽省高校计算机基础课程教学指导委员会副主任,中国计算机学会理事,安徽省计算机学会理事长,全国自然科学名词审定委员会委员,中国科学技术大学计算机系主任。

陈国良教授长期从事计算机科学技术的研究与教学工作。主要研究领域为并行算法、并行计算机体系结构和智能计算等。先后承担10多项国家863计划、国家攀登计划、国家自然科学基金、国家973计划、教育部博士点基金等科研项目。取得了多项被国内外广泛引用、达国际先进水平的科研成果,发表论文180多篇,出版著作9部、译著5部,参与主编计算机类辞典、词汇5部,主审、主编计算机类各种教材8部。曾获国家科技进步二等奖、国家级教学成果二等奖、国家教育部科技进步一等奖、中国科学院科技进步二等奖和自然科学三等奖、全国优秀教材一等奖、全国学术著作优秀奖、安徽省科技进步二等奖、国家科委高技术研究与发展计划三等奖、国家教委科技进步三等奖共18项,并获2001年度“国家863计划15周年先进个人重要贡献奖”。

陈国良教授在中国科学技术大学执教30年。长期以来,围绕着并行算法的教学与研究,逐渐形成了一套完整的“算法理论—算法设计—算法实现—算法应用”的并行算法学科体系,营造了我国并行算法类的教学基地。他先后指导培养研究生100多名,其中博士生60名,为我国培养了一批在国内外从事算法研究的高级人才。曾荣获1998年度安徽省教育系统劳动模范、安徽省优秀教师称号、2001年度宝钢教育基金优秀教师特等奖和2003年度全国第一届高等学校教学名师奖。

陈国良教授是我国非数值并行算法研究的学科带头人。他率先创建的我国第一个国家高性能计算中心是我国并行算法研究、环境科学与工程计算软件开发的重要基地,在学术界和教育界有一定的影响和地位。

序 言

高性能计算机是一个国家经济和科技实力的综合体现,也是促进经济、科技发展,社会进步和国防安全的重要工具,已成为世界各国竞相争夺的战略制高点。一些发达国家纷纷制定战略计划,提出很高目标,投入大量资金,加速研究开发步伐。多年来,随着大规模集成电路技术的不断进步,以多 CPU为基础的高性能并行计算机得到了迅速的发展,其高端系统正向百万亿次、千万亿次迈进。我国近十年来,对高性能并行计算的研究开发也给予了很大重视,取得了长足进步和可贵经验,研制出了具有相当水平的并行机系统,但与发达国家相比,差距仍然甚大,在高性能并行计算的应用开发与相关的人才培养教育方面尤显不足。如何使高性能并行机系统深入充分地在国民经济、科研和社会应用的发展中发挥作用,实为当务之急,引起人们的普遍关心。

由中国科技大学陈国良教授主编的这套丛书,正适应了我国高性能并行计算研究、开发、应用、教育之需。本丛书由《并行算法的设计与分析》、《并行计算机体系结构》和《并行算法实践》三大部分组成,而以《并行计算——结构·算法·编程》为全丛书之提要。该丛书以并行计算为主题,对并行计算的硬件平台(当代主流并行计算机系统)、并行计算的理论基础(并行算法的设计与分析)和并行计算的软件支撑(并行程序设计)全面系统地展开了讨论,内容丰富,取材新近,具有相当的深度和广度,涵盖了并行计算机体系结构和并行算法的理论、设计和实践的各个方面,是国内外不多见的优秀著作。

陈国良教授是国家高性能计算中心(合肥)主任,长期从事并行算法和并行计算机体系结构的研究,本套丛书是作者几十年从事教学与科研工作的结晶,是目前国内该领域内容涵盖最为全面的系列著作。它的出版必将对进一步推动我国并行计算学科的发展与应用推广产生深远的影响。

前 言

并行计算系列丛书 并行计算的研究内容一般包括并行计算的理论基础 (即并行算法),实现它的硬件平台 (即并行计算机)和软件支撑 (即并行程序设计)。作者按此思想,曾于 1999年撰写了《并行计算——结构·算法·编程》一书,然后于 2002年又分别撰写和修订了《并行计算机体系结构》和《并行算法的设计与分析》两书。即将出版的《并行算法实践》介绍并行程序设计的有关知识和并行算法的具体编程实现,它与上述的三本书一起构成了并行计算系列丛书。该系列丛书是并行算法类教学体系 (如下图所示)中的主要内容,而《并行算法实践》处于该体系中的算法实现层次,是面向计算机专业研究生的。

章节内容 《并行算法实践》分为上篇和下篇。上篇为并行程序设计导论,共有三个单元:单元 为并行程序设计基础,包括并行计算机系统与结构模型 (第一章)、PC 机群的搭建 (第二章)和并行程序设计简介 (第三章);单元 为并行程序编程指南,包括 MPI编程指南 (第四章)、PVM 编程指南 (第五章)、HPF 编程指南 (第六章)和 OperMP 编程指南 (第七

章);单元 为并行程序开发方法,包括可视化并行程序设计环境(第八章)、并行程序的调试(第九章)、并行程序的性能分析(第十章)、并行程序的性能优化(第十一章)和图形化并行程序集成开发环境 GRADE简介(第十二章)。下篇为并行算法编程实现,共有两个单元:单元 为非数值并行算法 MPI编程实现,包括排序(第十三章)、串匹配(第十四章)、图论(第十五章)、组合优化(第十六章)和计算几何(第十七章)等问题的并行算法及其 MPI编程实现;单元 为数值并行算法 MPI编程实现,包括矩阵运算(第十八章)、线性方程组的直接解法(第十九章)、线性方程组的迭代解法(第二十章)、矩阵特征值计算(第二十一章)以及快速傅氏变换和离散小波变换(第二十二章)的并行算法及其 MPI编程实现。各章后注有主要参考文献(参数文献中所给出的网址可能会随时间变化,仅供参考),各单元后附有适量的习题,书末附有算法索引、MPI源程序清单和专业术语中英文对照及索引。

使用方法 《并行算法实践》是并行计算系列丛书之四,除了作为丛书的配套教材使用之外,也可以单独使用。当作为并行程序设计教材使用时,重点讲授上篇(30学时),选讲下篇部分内容(10学时);当作为并行算法类课程实验指导书使用时,可选讲上篇的第二章、第四章和第九章(10学时),重点是通过下篇的计算机科学中诸多常用计算问题的并行算法 MPI编程实现,来培养学生独立编写和调试并行程序的能力,一般上机时间可安排 20~30学时,上机的环境可以是超级并行机,也可以是自己搭建的 PC 机群(详见第二章)。

值得说明的是,下篇所选择的那些计算问题不一定最有代表性,相应的并行算法也并非一定最优,所编写的 MPI源程序并不推荐作为标准的并行算法库,只是期望起到示例作用,并且为了压缩篇幅,对于每一类计算问题,在正文中只择其一给出相应的 MPI源程序,其余者均放在随书的光盘中,同时也公布在中国科学技术大学国家高性能计算中心(合肥)的网站上:<http://www.nhpc.ustc.edu.cn>,欢迎读者丰富和优化它们。

相关著作 本书在撰写时除考虑到与丛书的整体内容协调外,还广泛地吸取了当今国内外相应著作中的精彩部分,以丰富和完善自身的内容。这些著作有: Raikumar Buyya 编,高性能集群计算,郑纬民等译,电子工业出版社,2001; Edited by Jose C Cunha, Peter Kasuk and Stephen C Winter. Parallel program development for cluster computing: methodology, tools and integrated environments. Nova Science Publishers, Inc., Huntington, New York, 2001; Barry Wilkinson, Michael Allen. Parallel programming: techniques and applications using networked workstations and parallel computers, Prentice-Hall, 1999(并行程序设计——技术与应用,影印版,高等教育出版社/Pearson Education 出版集团,2002),中译本:陆鑫达等译,并行程序设计,机械工业出版社,2002。所以读者在阅读本书时,如能配合阅读上述几本著作,将是非常有益的。

致谢 本书撰写时,曾直接或间接地引用许多专家、学者们的文献,有些内容也得益于上述几本著作,作者向他们深表谢意;但也有很多优秀文献未能被引用,作者也向他们深表歉意。书稿付梓前,承蒙北京大学李晓明教授进行了审校,提出了不少中肯的修改意见,作者尤为感谢。

陈国良教授根据并行计算丛书的总体规划,拟定了全书详细的章节内容。其中,中国

科学技术大学负责第 单元 (陈国良)、第 单元 (安虹)、第 单元 (郑启龙) 和第 单元 (单久龙) 的编写, 扬州大学负责第 单元 (陈旻) 的编写。全书成稿后, 经陈国良教授反复修改而定稿。参加编写工作的还有中国科学技术大学计算机科学技术系研究生万颖瑜、何家华、姚震、鄢超、李一哲、沈一飞、陈志辉、陈勇和宋彬等, 以及扬州大学计算机系研究生周解、秦玲和徐晓华等。对于这些年轻学生的辛勤劳动和他们对计算机事业的热爱, 作者深感欣慰并致谢意。

尽管本书的内容曾在中国科学技术大学计算机科学技术系的教学中多次使用过, 但书中错误和片面之处肯定还会存在, 恳请读者不吝批评指正。

作者

2003年 9月

目 录

上篇 并行程序设计导论

单元 I 并行程序设计基础

第一章 并行计算机系统与结构模型	(5)	2.4 机群系统性能评测	(49)
1.1 典型并行计算机系统简介	(5)	2.4.1 基准测试程序	(49)
1.1.1 阵列处理机	(5)	2.4.2 性能分析工具	(53)
1.1.2 向量处理机	(7)	2.5 小结	(56)
1.1.3 共享存储多处理机	(9)	参考文献	(56)
1.1.4 分布存储多计算机	(10)	第三章 并行程序设计简介	(58)
1.1.5 分布共享存储多处理机	(12)	3.1 并行程序开发方法	(58)
1.2 当代并行计算机体系结构	(14)	3.1.1 并行层次与代码粒度	(58)
1.2.1 并行计算机体系结构模型	(14)	3.1.2 并行程序开发策略	(59)
1.2.2 并行计算机存储结构模型	(17)	3.1.3 并行编程模式	(61)
1.2.3 分布式高速缓存与主存 体系结构	(19)	3.1.4 并行应用编程过程	(63)
1.3 小结	(22)	3.2 并行程序设计模型	(68)
参考文献	(23)	3.2.1 计算 样本程序	(68)
第二章 PC 机群的搭建	(24)	3.2.2 数据并行模型	(69)
2.1 机群系统概述	(24)	3.2.3 消息传递模型	(71)
2.1.1 机群系统原理与技术	(24)	3.2.4 共享变量模型	(72)
2.1.2 典型机群系统简介	(26)	3.3 并行编程语言和环境概述	(73)
2.2 硬件的选择与安装	(27)	3.3.1 早期并行编程语言	(74)
2.2.1 节点构建	(28)	3.3.2 近代并行编程语言与环境	(75)
2.2.2 系统构建	(31)	3.3.3 并行说明性语言环境	(78)
2.2.3 机群系统示例	(33)	3.4 循环程序并行化的一般方法	(78)
2.3 软件的选择与安装	(33)	3.4.1 数据相关分析	(78)
2.3.1 OS的选择	(33)	3.4.2 数据划分与处理器指派	(80)
2.3.2 SSI的构建	(35)	3.4.3 循环重构	(87)
2.3.3 编程环境的选择	(38)	3.5 小结	(96)
2.3.4 作业管理系统的选择	(42)	参考文献	(96)
		单元 习题	(98)

单元 并行程序编程指南

第四章 MPI编程指南	(105)	4.1 引言	(105)
-------------------	-------	--------------	-------

2目 录

4.1.1 MPI的产生	(105)	5.5.4 通信函数	(144)
4.1.2 MPI的语言绑定	(106)	5.5.5 计算的 PVM 程序	(145)
4.1.3 MPI的实现	(107)	5.6 PVM 虚拟机结构	(146)
4.2 6个基本函数组成的 MPI子集	(107)	5.6.1 结构分析	(147)
4.3 MPI消息	(109)	5.6.2 动态配置	(147)
4.3.1 消息数据类型	(110)	5.6.3 PVM 虚拟机的构建过程	(148)
4.3.2 消息标签	(113)	5.7 小结	(149)
4.3.3 通信域	(115)	参考文献	(149)
4.3.4 消息状态	(117)	第六章 HPF 编程指南	(150)
4.4 点对点通信	(117)	6.1 HPF概述	(151)
4.4.1 MPI通信模式	(118)	6.1.1 引言	(151)
4.4.2 阻塞和非阻塞通信	(119)	6.1.2 HPF的语言特点	(152)
4.4.3 通信和计算的重叠	(119)	6.1.3 HPF的语言模型	(153)
4.5 群集通信	(121)	6.2 HPF编程简介	(154)
4.5.1 群集通信的通信功能	(122)	6.2.1 一个简单的 HPF程序实例	(154)
4.5.2 群集通信的同步功能	(125)	6.2.2 HPF的基本特性	(155)
4.5.3 群集通信的聚合功能	(125)	6.3 数据映射	(159)
4.5.4 群集通信例程的共同特点	(126)	6.3.1 数据映射说明语句	(159)
4.5.5 计算的 MPI程序	(127)	6.3.2 一个数据映射的 HPF程序 段分析	(164)
4.6 MPI扩展	(127)	6.4 数据并行结构	(165)
4.6.1 动态进程	(128)	6.4.1 数组运算	(165)
4.6.2 远程存储访问	(129)	6.4.2 FORALL语句和 FORALL结构	(167)
4.6.3 并行 IO	(131)	6.4.3 INDEPENDENT指示	(169)
4.7 小结	(132)	6.5 HPF语言的过程	(171)
参考文献	(132)	6.5.1 HPF语言的一般函数和子程序	(171)
第五章 PVM 编程指南	(133)	6.5.2 HPF语言内部函数	(172)
5.1 引言	(133)	6.5.3 HPF语言的库函数	(174)
5.1.1 开发历史	(133)	6.6 HPF实例分析 :2D FFT	(176)
5.1.2 PVM的特点	(134)	6.7 HPF语言其他特性	(178)
5.1.3 与 MPI的比较	(134)	6.7.1 HPF对 FORTRAN 90语言 的限制	(178)
5.2 PVM 的启动和命令	(135)	6.7.2 HPF 1.1子集	(179)
5.3 一个简单的 PVM程序	(136)	6.7.3 HPF 2.0与 HPF 1.1的不同点	(180)
5.3.1 程序介绍	(136)	6.8 小结	(180)
5.3.2 编译和运行	(138)	参考文献	(181)
5.4 PVM 任务	(139)	附录一 HPF 指令语法	(182)
5.4.1 任务派生	(139)	附录二 一般的内部函数及库函数	(184)
5.4.2 任务组	(140)	附录三 HPF 网络资源	(187)
5.4.3 任务标识符	(141)	第七章 OpenMP 编程指南	(189)
5.4.4 任务管理	(141)	7.1 OpenMP概述	(189)
5.5 PVM 通信	(142)	7.1.1 什么是 OpenMP	(190)
5.5.1 点对点通信	(142)	7.1.2 OpenMP的历史	(191)
5.5.2 群集通信	(142)		
5.5.3 消息的打包/解包	(143)		

7.1.3 OpenMP的目标	(191)	7.3.7 threadprivate编译制导语句.....	(205)
7.2 OpenMP编程风格	(191)	7.3.8 数据域属性子句	(206)
7.2.1 OpenMP并行编程模型	(191)	7.3.9 子句 编译制导语句总结	(208)
7.2.2 OpenMP程序结构	(192)	7.3.10 语句的绑定和嵌套规则	(209)
7.3 OpenMP编程简介	(193)	7.4 运行库例程与环境变量	(210)
7.3.1 一个简单的OpenMP程序实例	(194)	7.5 OpenMP计算实例	(210)
7.3.2 编译制导	(195)	7.6 小结	(213)
7.3.3 并行域结构	(196)	参考文献	(214)
7.3.4 共享任务结构	(197)	附录 运行库例程	(215)
7.3.5 组合的并行共享任务结构	(201)	单元 习题	(217)
7.3.6 同步结构	(202)		
单元 并程序开发方法			
第八章 可视化并行程序设计环境	(223)	9.2.1 前期设计	(254)
8.1 引言	(223)	9.2.2 初步实现	(256)
8.1.1 并行软件工程	(223)	9.2.3 功能开发	(258)
8.1.2 并行程序开发环境的要求	(224)	9.2.4 维护	(260)
8.1.3 工具集成	(225)	9.3 高级并行调试技术简介	(260)
8.2 SEPP/HPCTI简介	(225)	9.3.1 全局断点	(261)
8.2.1 SEPP/HPCTI方法	(225)	9.3.2 渐增检查点	(261)
8.2.2 SEPP/HPCTI组成	(226)	9.3.3 事件分析	(261)
8.3 可视化并行语言的分类	(228)	9.3.4 静态分析	(262)
8.3.1 通用编程模型	(229)	9.4 并行程序的性能调试	(262)
8.3.2 进程模型	(232)	9.4.1 性能调试的一般步骤	(262)
8.3.3 进程交互	(233)	9.4.2 性能分析工具举例 :VAMPR 和 GuideView	(263)
8.3.4 正则并行结构	(234)	9.5 小结	(267)
8.3.5 分层设计和代码复用	(235)	参考文献	(267)
8.4 可视化环境实例	(236)	第十章 并行程序的性能分析	(269)
8.4.1 FrameWorks系统	(236)	10.1 并行程序性能监控	(269)
8.4.2 Enterprise并行编程系统	(237)	10.1.1 监控的应用和分类	(269)
8.4.3 CODE 2.0语言	(238)	10.1.2 并行跟踪的实现	(271)
8.4.4 HeNCE编程环境	(239)	10.1.3 侵扰的模型和补偿处理	(273)
8.4.5 TRAPPER编程环境	(240)	10.1.4 并行监控和操作系统的结合 与交互	(275)
8.4.6 Meander环境	(241)	10.2 并行程序性能预测	(275)
8.5 小结	(243)	10.2.1 并行系统中的性能预测	(275)
参考文献	(244)	10.2.2 并行系统建模	(278)
第九章 并行程序的调试	(245)	10.2.3 并行系统模拟仿真	(281)
9.1 并行调试的方法与步骤	(245)	10.3 性能可视化	(282)
9.1.1 并行调试的困难	(245)	10.3.1 可视化的概念	(283)
9.1.2 并行调试的方法	(246)	10.3.2 数据生成	(284)
9.1.3 并行调试的步骤	(247)		
9.2 并行调试器的设计与实现	(254)		

4目 录

10.3.3 数据显示	(285)	GRADE简介	(320)
10.3.4 数据分析和用户交互	(289)	12.1 GRADE并行程序集成开发环境	(320)
10.3.5 用户界面	(291)	12.1.1 GRADE的组成	(321)
10.4 小结	(291)	12.1.2 在GRADE环境中开发并行程序 的步骤	(321)
参考文献	(292)	12.2 可视化并行程序设计	(323)
第十一章 并行程序的性能优化	(294)	12.2.1 可视化并行程序设计语言 GRAPNEL	(323)
11.1 引言	(294)	12.2.2 图形编辑器 GRED	(328)
11.1.1 调度问题的一般模型	(294)	12.3 映射和调度以及负载平衡工具	(330)
11.1.2 并行计算中的任务调度	(295)	12.3.1 DSM&S和DLB与GRADE环境 的集成	(330)
11.1.3 并行计算中任务调度的分类	(297)	12.3.2 调度和映射工具	(331)
11.1.4 并行计算中任务调度的模型	(299)	12.3.3 动态负载平衡系统	(332)
11.2 静态任务调度的NP完全性及其 最优算法	(302)	12.4 并行分布式程序调试器	(333)
11.2.1 静态任务调度的NP完全性	(302)	12.4.1 DDBG与GRADE的集成	(334)
11.2.2 静态任务调度的最优算法	(304)	12.4.2 DDBG的体系结构与接口库	(334)
11.3 静态任务调度的启发式算法	(305)	12.4.3 GRED与DDBG的集成	(336)
11.3.1 贪心算法	(305)	12.5 Tape/PVM监控器和PROVE可视化 工具	(337)
11.3.2 随机算法	(306)	12.5.1 源代码插桩	(337)
11.3.3 聚簇策略	(307)	12.5.2 数据获取和跟踪分析	(339)
11.4 动态负载平衡	(308)	12.5.3 可视化	(340)
11.4.1 基本概念	(308)	12.6 小结	(341)
11.4.2 负载信息收集	(309)	参考文献	(341)
11.4.3 负载迁移决策	(310)	单元 习题	(343)
11.4.4 负载迁移执行	(315)		
11.5 小结	(317)		
参考文献	(318)		
第十二章 图形化并行程序集成开发环境			

下篇 并行算法编程实现

单元 非数值并行算法MPI编程实现

第十三章 排序	(351)	参考文献	(357)
13.1 枚举排序	(351)	附录 PSRS算法MPI源程序	(358)
13.1.1 枚举排序及其串行算法	(351)	第十四章 串匹配	(364)
13.1.2 枚举排序的并行算法	(352)	14.1 KMP串匹配算法	(364)
13.2 快速排序	(353)	14.1.1 KMP串匹配及其串行算法	(364)
13.2.1 快速排序及其串行算法	(353)	14.1.2 KMP串匹配的并行算法	(368)
13.2.2 快速排序的并行算法	(354)	14.2 随机串匹配算法	(372)
13.3 并行正则采样排序 PSRS	(356)	14.2.1 随机串匹配及其串行算法	(372)
13.3.1 PSRS算法原理	(356)	14.2.2 随机串匹配的并行算法	(374)
13.3.2 PSRS算法形式化描述	(356)	14.3 近似串匹配算法	(375)
13.4 小结	(357)	14.3.1 近似串匹配及其串行算法	(375)

14.3.2 近似串匹配的并行算法	(381)	16.3 装箱问题	(418)
14.4 小结	(383)	16.3.1 装箱问题及其串行算法	(418)
参考文献	(383)	16.3.2 装箱问题的并行算法	(419)
附录 KMP串匹配并行算法的 MPI源程序	(385)	16.4 背包问题	(420)
第十五章 图论	(392)	16.4.1 背包问题及其串行算法	(420)
15.1 传递闭包	(392)	16.4.2 背包问题的并行算法	(422)
15.1.1 传递闭包串行算法	(392)	16.5 TSP问题	(423)
15.1.2 传递闭包并行算法	(394)	16.5.1 TSP问题及其串行算法	(423)
15.2 连通分量	(396)	16.5.2 TSP问题的并行算法	(423)
15.2.1 顶点倒塌法算法原理描述	(396)	16.6 小结	(425)
15.2.2 连通分量并行算法	(396)	参考文献	(426)
15.3 单源最短路径	(398)	附录 八皇后问题并行算法的 MPI源程序 ...	(427)
15.3.1 最短路径串行算法	(398)	第十七章 计算几何	(432)
15.3.2 最短路径并行算法	(399)	17.1 包含问题	(432)
15.4 最小生成树	(402)	17.1.1 包含问题及其串行算法	(432)
15.4.1 最小生成树串行算法	(402)	17.1.2 包含问题并行算法	(433)
15.4.2 最小生成树并行算法	(403)	17.2 相交问题	(435)
15.5 小结	(406)	17.2.1 两多边形相交问题及其串行 算法	(435)
参考文献	(406)	17.2.2 相交问题的并行算法	(436)
附录 连通分量并行算法的 MPI源程序	(407)	17.3 凸壳问题	(437)
第十六章 组合优化	(411)	17.3.1 凸壳问题及其串行算法	(438)
16.1 八皇后问题	(411)	17.3.2 凸壳问题并行算法	(439)
16.1.1 八皇后问题及其串行算法	(411)	17.4 小结	(440)
16.1.2 八皇后问题的并行算法	(412)	参考文献	(440)
16.2 SAT问题	(414)	附录 包含问题并行算法的 MPI源程序	(441)
16.2.1 SAT问题及其串行算法	(414)	单元 习题	(444)
16.2.2 SAT问题的并行算法	(415)		

单元 数值并行算法 MPI编程实现

第十八章 矩阵运算	(455)	18.4.2 Cannon乘法的并行算法	(462)
18.1 矩阵转置	(455)	18.5 LU分解	(466)
18.1.1 矩阵转置及其串行算法	(455)	18.5.1 矩阵的 LU分解及其串行算法	(466)
18.1.2 矩阵转置并行算法	(456)	18.5.2 矩阵 LU分解的并行算法	(467)
18.2 矩阵 - 向量乘法	(458)	18.6 QR分解	(469)
18.2.1 矩阵 - 向量乘法及其串行算法 ...	(458)	18.6.1 矩阵 QR分解的串行算法	(469)
18.2.2 矩阵 - 向量乘法的并行算法	(458)	18.6.2 矩阵 QR分解的并行算法	(471)
18.3 行列划分矩阵乘法	(459)	18.7 奇异值分解	(474)
18.3.1 矩阵相乘及其串行算法	(459)	18.7.1 矩阵奇异值分解的串行算法	(474)
18.3.2 简单的矩阵并行分块乘法算法 ...	(460)	18.7.2 矩阵奇异值分解的并行算法	(477)
18.4 Cannon乘法	(462)	18.8 Cholesky分解	(480)
18.4.1 Cannon乘法的原理	(462)	18.8.1 矩阵 Cholesky分解的串行算法 ...	(480)

18.8.2 矩阵 Cholesky 分解的并行算法 ...	(481)	第二十一章 矩阵特征值计算	(540)
18.9 矩阵求逆	(483)	21.1 求解矩阵最大特征值的乘幂法	(540)
18.9.1 求矩阵的逆的串行算法	(483)	21.1.1 乘幂法及其串行算法	(540)
18.9.2 矩阵求逆的并行算法	(484)	21.1.2 乘幂法并行算法	(541)
18.10 小结	(486)	21.2 求对称矩阵特征值的雅可比法	(543)
参考文献	(486)	21.2.1 雅可比法求对称矩阵特征值 的串行算法	(543)
附录一 Cannon乘法并行算法的 MPI源程序	(488)	21.2.2 雅可比法求对称矩阵特征值 的并行算法	(546)
附录二 矩阵 LU 分解并行算法的 MPI源程序	(494)	21.3 求对称矩阵特征值的单侧旋转法	(556)
附录三 矩阵求逆并行算法的 MPI源程序 ...	(498)	21.3.1 单侧旋转法的算法描述	(556)
第十九章 线性方程组的直接解法	(504)	21.3.2 求对称矩阵特征值的单侧旋转法 的并行计算	(559)
19.1 高斯消去法解线性方程组	(504)	21.4 求一般矩阵全部特征值的 QR方法 ...	(562)
19.1.1 高斯消去及其串行算法	(504)	21.4.1 QR方法求一般矩阵全部特征值 的串行算法	(562)
19.1.2 并行高斯消去算法	(507)	21.4.2 QR方法求一般矩阵全部特征值 的并行算法	(563)
19.2 约当消去法解线性方程组	(512)	21.5 小结	(565)
19.2.1 约当消去及其串行算法	(512)	参考文献	(566)
19.2.2 约当消去法的并行算法	(514)	附录 求对称矩阵特征值的雅可比并行算法 MPI源程序	(567)
19.3 小结	(517)	第二十二章 快速傅氏变换和离散小波变换 ...	(581)
参考文献	(518)	22.1 快速傅里叶变换 FFT	(581)
附录 全主元高斯消去法并行算法的 MPI源程序	(519)	22.1.1 串行 FFT迭代算法	(581)
第二十章 线性方程组的迭代解法	(524)	22.1.2 并行 FFT算法	(583)
20.1 雅可比迭代	(524)	22.2 离散小波变换 DWT	(585)
20.1.1 雅可比迭代及其串行算法	(524)	22.2.1 离散小波变换 DWT及其串行 算法	(585)
20.1.2 雅可比迭代并行算法	(526)	22.2.2 离散小波变换并行算法	(588)
20.2 高斯-塞德尔迭代	(527)	22.3 小结	(590)
20.2.1 高斯-塞德尔迭代及其串行 算法	(527)	参考文献	(590)
20.2.2 高斯-塞德尔迭代并行算法	(528)	附录 FFT并行算法的 MPI源程序	(591)
20.3 松弛法	(531)	单元 V 习题	(598)
20.3.1 松弛法及其串行算法	(531)	算法索引	(607)
20.3.2 松弛法并行算法	(532)	MPI源程序清单	(609)
20.4 小结	(534)	专业术语中英文对照及索引	(611)
参考文献	(534)		
附录 高斯-塞德尔迭代并行算法的 MPI源程序	(535)		

单元 并行程序开发方法

第八章 可视化并行程序设计环境

第九章 并行程序的调试

第十章 并行程序的性能分析

第十一章 并行程序的性能优化

第十二章 图形化并行程序集成开发环境 GRADE 简介

这一单元主要研究并行应用程序的开发方法,包括可视化的并行程序设计环境(第八章)、并行程序的调试(第九章)、并行程序的性能分析(第十章)、并行程序的性能优化(第十一章)和图形化并行程序集成开发环境 GRADE 简介(第十二章)。

本单元的第八章介绍可视化的并行程序设计环境,包括可视化并行程序的开发、可视化并行语言和可视化编程环境实例介绍;第九章介绍并行程序的调试,包括并行调试的方法和步骤、并行调试技术和并行调试器的设计与实现;第十章介绍并行程序的性能分析,包括并行程序性能监控、并行程序性能预测和并行程序性能可视化;第十一章介绍并行程序的性能优化,包括并行任务调度模型和算法,分布式环境下动态负载平衡方法;第十二章综合介绍一个图形化并行程序集成开发环境实例(GRADE),包括可视化并行程序设计语言 GRAPNEL、图形编辑器 GRED、分布式并行程序调试器 DDBG、映射调度工具 DSM&S 与负载平衡工具 DLB、可视化监测工具 Tape/PVM 与可视化工具 PROVE 以及系统模拟工具等。

第八章 可视化并行程序设计环境

在硬件技术高速发展的今天,支持图形化的程序设计越来越重要。使用可视化编程来代替亲自动手写代码是很自然的想法。本章首先简要介绍并行编程环境的基本概况,接着讨论可视化编程语言分类,最后介绍当今流行的几种可视化并行开发环境。

8.1 引言

在过去十几年中计算机硬件技术飞速发展,出现了大量的并行计算平台,这就大大提高了许多应用程序的运行速度。特别是随着机群计算系统的出现,人们可以低价搭建并行计算系统,这也使得并行分布式应用程序开发环境的重要性随之不断增加。

随着并行应用的发展,人们对并行程序设计模型和支持它们的工具与环境提出了更高的要求。特别是由于在现实中缺乏合适的软件开发方法和工具,现在还有许多并行应用的性能无法充分地提高。

8.1.1 并行软件工程

软件工程界目前正在出现第二次严重软件危机。20世纪60年代,高速发展的体系结构使得有可能开发庞大复杂的应用程序,但是软件不断增长的需求与开发技术的落后之间的巨大矛盾不可避免地引发了软件危机。现在随着并行计算平台的普及,不论是高性能超级计算机还是工作站机群,针对这些平台的程序开发要比以前的串行程序更复杂,而且最糟糕的是程序员们一直使用简单的冯·诺伊曼机器编程模型,这在并行计算机上并不是完全正确

的。大量不同的并行硬件拓扑结构还会导致程序映射问题。以前程序员们仅需掌握普通串行软件的一维设计模式,而现在却被迫要掌握更复杂的设计空间!

在解决第一次软件危机的过程中出现了现代软件工程(Software Engineering),其中包括了如结构化编程、正规化定义等许多开发技术。人们希望把这些串行程序开发技术扩展之后引入到并行程序开发中来,显然许多针对串行软件工程的设想在并行开发中不再适用。例如在传统生命周期(Life cycle)模型中,如果程序性能评价被放在生命周期的末尾这个相对优先级不高的位置上,那么软件设计者不需要投入很大的精力在上面。但是在高性能计算中,一个应用如果需要使用并行处理,毫无疑问性能是决定性的要素。这就意味着传统串行软件开发中的生命周期模型不能完全满足并行软件工程的需要^[1]。

像这类新出现的问题,许多学者已经开始着手研究新的设计工具和方法,以适用于并行编程。第一代典型的并行开发工具大部分是由硬件厂商开发的,因此只能在厂商特定的硬件上使用。这些工具仅仅可以简单编程,显然不能满足大型项目的需要。而新的设计工具和方法的主要问题是,程序员们心理上无法马上接受不熟悉的编程模式。因此实际所使用的工具一定要考虑到心理上的惯性和对变化的抗拒,也就是说,在运用新技术的同时一定要兼顾以前的编程模式。因此,这就要求工具既要使硬件可用,又要使程序员们的知识和技巧不需要做大的改变。

8.1.2 并行程序开发环境的要求

简单来说,对并行程序开发环境的要求可分为两个方面:第一是正确性(Correctness)问题,它要保证设计出来的并行分布式程序对要实现的应用是绝对可靠的。第二是性能(Performance)问题,也就是如何在给定硬件结构上使得程序执行时获得可以接受的性能。

在传统的程序开发中,程序员们总喜欢将应用分成一系列定义清楚的子阶段:从问题分析与定义、程序设计与实现、到实际执行。并行程序开发中主要困难是在开发生命周期的不同阶段之间有着不可避免的交互。对于正确性而言,并行分布式处理器之间错综复杂的交互造成了分布共享资源的存取一致性(Consistency)问题和并行分布计算的不确定性(Indeterminacy)问题。而性能与问题的定义和分解以及要映射的具体并行体系结构硬件之间有密切关系。为了更好地分解问题,必须细致地了解并行体系结构的特性和限制。例如,共享存储器和分布存储器模型之间的区别、或者处理器间的通信延迟的不同都会给最后的性能带来巨大影响。

要使并行程序员更好地编程,就要提供合适的工具与集成开发环境,用以支持问题定义与分解以及代码生成与执行等不同阶段。当然,所有阶段都要有合适的

工具来帮助用户更好地理解程序的行为和关注程序的正确性与性能,只有这样的工具才能帮助编写出优质的并行软件。

8.1.3 工具集成

过去十年中,研究者们开发了大量不同的并行开发工具。但是用户们在实际并行程序开发时并不愿意使用它们。造成这种情况的主要原因是,这些绝大部分工具的学习与安装都很困难,而且它们通常无法简化所有的用户任务。当然,这并不是说没有成功的商业和学术工具的例子。

现在一个很重要的问题是,单独一个工具只支持某些特定功能,如调试或性能分析等。要在并行软件生命周期的不同阶段帮助用户处理各式各样的任务,这些单独工具是不够的,因此我们要把这些不同工具集成到一个完整的开发环境中。而这就需要对提供的并行开发软件提出附加的要求,因为工具集成要求在环境中的不同工具有着一致连续的视角。

集成环境中各种工具有着各自不同的用户界面、行为和抽象层次,一个集成环境应该对软件的整个生命周期的各种活动提供一个统一的视角,这种视角应该集中在应用级概念的高层抽象上。而且只要有可能,一个环境的所有功能应该提供风格统一的用户界面。开发环境应该引导用户根据软件生命周期中的当前阶段,来理顺不同操作之间的前后关系。任何时候环境都要考虑底层的特性,包括中间件层、操作系统、硬件体系结构层。环境还应该帮助用户建立抽象概念之间的联系。

8.2 SEPP/HPCTI 简介

8.2.1 SEPP/HPCTI 方法

1994年,欧洲24所大学、研究所和公司联合建立起一个组织,旨在研究图形化并行程序设计环境和并行程序开发的整个生命周期,他们主要关注基于消息传递的机群系统上的并行开发。该协会主持研究了两个颇有影响的项目:SEPP(Software Engineering for Parallel Processing)和HPCTI(High Performance Computing Tools for Industry)。他们在研究中积累了大量的如并行程序开发方法、支持工具和应用这些并行软件工程的重要知识。下面将介绍一下SEPP/HPCTI方法^[2]的基本概念与思想^[2]。