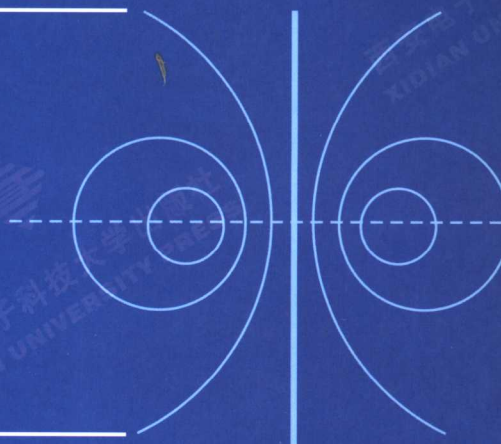


Parallel
Computation
in Electromagnetics



电磁场并行计算

张玉 著



西安电子科技大学出版社
<http://www.xduph.com>

电磁场并行计算

张 玉 著

西安电子科技大学出版社

2006

内 容 简 介

本书主要讨论电磁场数值分析方法的并行实现,依次介绍了一致性几何绕射理论、物理光学方法、矩量法、时域有限差分法和高低频混合方法等五种具有代表意义的算法及其并行计算方法。本书重视理论与实践的结合,不仅对每种并行电磁场计算算法提供了其在复杂电磁仿真工程中的应用实例,还在附录中给出了并行 FDTD 方法的源代码以及 MPI 并行环境设置、并行程序运行方法等内容,以便于读者理解和使用。

本书既可以作为电磁场专业研究生“电磁场数值分析”相关课程的教材或者参考书,也可供相关专业的科研人员参考。

图书在版编目(CIP)数据

电磁场并行计算/张玉著.

—西安:西安电子科技大学出版社,2006.1

ISBN 7-5606-1597-X

I. 电… II. 张… III. 电磁计算 IV. TM15

中国版本图书馆 CIP 数据核字(2005)第 127938 号

责任编辑 夏大平

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西光大印务有限责任公司

版 次 2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 11.25

字 数 254 千字

印 数 1~2000 册

定 价 22.00 元

ISBN 7-5606-1597-X/TN·0317

XDUP 1889001 - 1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

序

这是一本关于电磁场现代数值分析方法的交流著作。

在以计算机为载体的各领域、各学科计算和仿真研究方面，我国和世界先进国家之间尚存在着不小的差距。如何尽快缩小甚至填补这一差距是摆在我国学者，尤其是年青学者面前的重担和任务。

张玉博士在自己的科学研究中总结与提炼出“电磁场并行计算”这一分支，并整理成书，以期和前辈、同行交流学习，这无疑是一件好事。正因为如此，这方面内容尚属新颖，而新颖的内容在另一侧面则反映其还不够成熟。然而，“初生之犊不怕虎”，把这些内容及时整理出来并和大家切磋探讨正是共同促进这一领域工作的最好途径。

张玉博士的工作也反映了我们 AEMC Group 工作的一个侧面，其中每一部分都有他和年青同事们的辛勤汗水。我们期待听取更多批评指正的意见，也期待这些年青学者在大家的批评帮助下获得更快的提高，更期待这一领域通过开放交流，百花齐放，迎来无限美好的科学春天。

是为序。



2005年6月27日

前 言

本书是在梁昌洪教授的直接支持与指导下完成的。

半个世纪以来，随着计算机技术与计算数学的发展，计算电磁学迅速地成为电磁领域中不可缺少的重要研究方向之一。根据我们多年来从事计算电磁学的实践，即便是在计算机技术日新月异的今天，由于受到计算资源的限制，现实中不断涌现出的诸多大规模电磁仿真问题，依然难于在普通 PC 机上得到高效解决。显然，采用并行计算技术，将计算任务分解进行——电磁场并行计算，是一种有效扩大仿真规模，提高仿真效率的途径。

随着计算机群集技术的发展，并行计算已从曲高和寡的局面转变为越来越广泛地普及到各个领域，硬件支持也从原先只有国家和大公司才能负担得起的超级计算机发展到现在一个小实验室就可以组建的“个人超级计算机”——PC 集群系统或者工作站集群系统。实际上，集群系统可以由那些晚上或周末“空闲”的 PC 机或者工作站组成，由 PC 机构成这样的系统无需任何额外的投资。根据这种想法，可以实现利用网络连接的任意数目的异构计算机来联合求解一个问题。正是这一技术的出现，为我们运用并行计算技术进行大规模电磁仿真提供了较低成本的硬件基础，使得电磁场并行计算具有推广、普及的可能性。

计算电磁学体系庞大，电磁场数值算法种类繁多，显然应该结合不同数值算法的自身特点进行并行算法设计，以提高并行程序的性能。并行计算是一个强大的工具，电磁研究人员却往往缺乏并行算法设计方面的指导，因此迫切需要一本可以使电磁专业人员理解与应用的电磁场并行计算方面的专业书籍，以推动并行电磁计算在我国的进一步发展。这也是本书问世的直接原因。

想在一本书里对所有算法的并行计算都介绍清楚是很困难的，实际上也是没有必要的。根据我们的经验，只要能理解电磁场数值分析中典型算法的并行实现，就可以实现几乎全部电磁场数值算法的并行计算。鉴于此，本书并没有对所有电磁场数值分析方法进行逐一的讨论，而是选取了一致性几何绕射理论(UTD)、物理光学法(PO)、矩量法(MoM)、时域有限差分法(FDTD)和高低频混合方法这五种算法进行并行计算研究。从算法的适用范围来看，既有适合电大目标的高频方法(UTD、PO)，也有适合一般电尺寸问题的低频方法(MoM)；

从算法数学基础来看,既有基于积分方程的矩量法,也有基于差分方程的时域有限差分法;从时频域来看,既涵盖了频域方法(UTD/PO/MoM),也涉及了时域方法(FDTD);除了这些之外,本书还考虑了两种数值算法结合而成的混合方法。

在介绍每一种算法时,本书力求保持其完整性,但由于本书重点是电磁计算并行算法设计,不太适合将所有算法的全部理论都在此详加讨论,因此仅对近十年来发展比较迅速的算法进行了基础性介绍,对于较为成熟的UTD方法以及矩量法,则不深入讨论其数学理论基础。

在讨论各种算法的并行实现时,本书遵循由简单到复杂、由低级到高级的原则展开讨论,给出了示范性的关键程序伪代码,引导读者循序渐进地掌握电磁场并行计算算法设计。除了选择典型算法以外,为了便于读者理解并行计算的概念,本书也对电磁场并行计算的典型硬件平台、软件环境作了介绍,并具体给出了PC集群系统组建实例与并行FDTD计算源程序。考虑到这些因素后,全书在内容层次上可分为六个部分。

第一部分即第一章,介绍了并行计算的基本知识,并对国内外近年来的部分电磁场并行计算研究成果进行了简单回顾。

第二、三章构成第二部分,讨论计算电磁学中的高频UTD方法与PO方法。在这两种高频算法中,由于其方法本身的特点,使得按照计算角域进行计算任务分解可以获得较为理想的并行性能,但在并行实现时需要考虑遮挡判断的计算量,以获得负载均衡。

第三部分包括第四、五章,主要讨论矩量法的并行计算问题。所谓并行矩量法,就是运用并行技术填充矩量法的矩阵和求解相应的矩阵方程。在传统的串行机上,Linpack是有效求解线性方程组的软件包,然而在并行机上求解此问题,就需要设计出合适的并行算法,算法的优劣会对计算效率产生很大的影响。不同的求解矩量法矩阵方程的方法对应不同的矩阵并行填充方式,这一部分将针对并行高斯消去法和并行共轭梯度法两种矩阵方程求解方法,分别讨论矩阵的列块划分以及矩阵的棋盘状划分方式,并详细讨论针对典型基函数的高效矩阵填充方法。

第六、七、八章为第四部分,这部分以FDTD为主要研究内容。FDTD方法的MPI并行计算中,不同的通信方式会导致计算效率的极大不同。为了利用MPI并行环境中进程虚拟拓扑可能给并行FDTD方法带来的效益,在第六章概括介绍FDTD的基本概念、第七章讨论其几个关键问题的基础上,第八章深

入地研究了三种不同通信方式的并行 FDTD 算法, 实测结果表明选择恰当的 MPI 虚拟拓扑方式(维数与方向), 可以有效地节约计算时间。

第九章即第五部分, 重点讨论了高低频混合方法的并行计算。在具体的工程问题中, 如果将若干电磁场数值算法混合使用, 形成混合方法, 做到“人”尽其才, 扬长避短, 发挥出各自的优势, 那么对于任何一个单一算法难以解决的问题都可能很容易地找到高效的解决途径。混合方法的并行计算, 在原理上是与被混合各算法的并行计算相似或相同的, 本部分对 MoM 与 PO 混合方法的并行计算进行了详细讨论。

第六部分为两个附录。附录 A 介绍了 PC 集群系统的组建与 MPI 环境的设置, 附录 B 给出了一维 FDTD 在一维通信方式下的 MPI 并行程序源代码, 相信对于初学者是一个很好的具体向导。

本书的形成是与作者的科研工作密不可分的。多年来的研究工作得到了瑞典 Andersson U 博士的热情指导以及西安电子科技大学史小卫教授、谢拥军教授的大力支持。本书在编写过程中也得到了课题组许多研究生的热情帮助: 王楠、王萌参与了第二、三章的校正, 赵勋旺、党晓杰参与了第四、五、九章的校正, 丁伟参与了第六、七、八章的校正, 刘越东与作者一起整理了附录内容与示范程序, 在此作者对他们一并表示感谢!

本书在出版过程中得到了西安电子科技大学出版社的大力支持, 作者非常感谢出版社的陈宇光、夏大平等同志为本书所做的大量工作。

限于作者学识水平, 书中难免有不当或疏漏之处, 敬请专家、学者和读者对本书提出宝贵意见!

作 者

2005 年 6 月

于西安电子科技大学

目 录

第一章 电磁场并行计算基础	1
1.1 硬件平台	2
1.1.1 依据指令与数据划分	2
1.1.2 依据存储方式划分	3
1.2 软件环境	5
1.3 并行算法	6
1.4 并行性能评测与影响因素	7
1.5 电磁场并行计算	8
1.6 小结	10
参考文献	10
第二章 UTD 方法与并行计算	13
2.1 几何绕射理论简介	13
2.2 并行 UTD 方法与性能测试	14
2.2.1 并行 UTD 方法	14
2.2.2 性能测试与应用	17
2.3 小结	20
参考文献	21
第三章 PO 方法与并行计算	22
3.1 雷达散射截面与物理光学方法	22
3.2 目标几何建模简介	24
3.3 基于平面三角形的 PO 方法	25
3.4 并行 PO 方法与性能测试	29
3.4.1 并行 PO 方法	29
3.4.2 性能测试与应用	32
3.5 小结	35
参考文献	35
第四章 RWG 基矩量法与并行计算	37
4.1 电场积分方程	37
4.2 RWG 基函数	37
4.3 矩量法解电场积分方程	39
4.3.1 矩阵方程	39
4.3.2 矩阵元素计算	41
4.3.3 积分运算次数	45
4.4 RWG 基矩量法计算数据提取	46

4.5	列块矩阵填充与并行高斯消去法	47
4.5.1	列块矩阵填充	47
4.5.2	并行高斯消去法	49
4.5.3	性能测试与应用	52
4.6	棋盘块矩阵填充与并行 CGN 方法	55
4.6.1	矩阵、进程虚拟拓扑与 CGN 迭代算法	55
4.6.2	棋盘块矩阵填充	57
4.6.3	并行 CGN 算法	60
4.6.4	性能测试与应用	62
4.7	两种并行矩量法比较	63
4.8	小结	64
	参考文献	64
第五章	线面连接模型与并行矩量法应用	65
5.1	线面连接结构	65
5.2	电场积分方程	66
5.3	基函数	66
5.3.1	导体基函数	67
5.3.2	导线基函数	67
5.3.3	连接域基函数	68
5.3.4	矢位和标位	71
5.4	检验过程与矩阵方程	72
5.4.1	检验过程	72
5.4.2	矩阵方程	76
5.5	并行矩量法	79
5.6	性能测试与应用	81
5.7	小结	84
	参考文献	84
第六章	FDTD 方法的基本原理	85
6.1	FDTD 方法概况	85
6.2	差分方法	86
6.3	直角坐标系中的时域有限差分方法	88
6.3.1	麦克斯韦方程组	88
6.3.2	一维情形——Yee 算法	89
6.3.3	三维情形——Yee 算法	93
6.3.4	媒质参数处理	97
6.4	数值稳定性	99
6.4.1	时间离散间隔的稳定性要求	100
6.4.2	Courant 稳定性条件	100
6.4.3	数值色散对空间离散间隔的要求	101
6.5	小结	103
	参考文献	103

第七章 FDTD 方法的几个关键问题	104
7.1 吸收边界条件	104
7.2 激励源	107
7.2.1 激励源的时频域信号形式	107
7.2.2 FDTD 方法中激励源的实现	109
7.3 S 参数提取	113
7.4 近远场变换	114
7.4.1 惠更斯原理	114
7.4.2 远场计算	116
7.5 小结	118
参考文献	118
第八章 FDTD 方法与并行计算	120
8.1 并行 FDTD 方法	120
8.2 性能测试与应用	126
8.2.1 并行 FDTD 的仿真实例	126
8.2.2 并行 FDTD 的性能分析	130
8.2.3 并行 FDTD 的工程应用	133
8.3 小结	134
参考文献	134
第九章 高低频混合方法与并行计算	136
9.1 高低频混合方法简介	136
9.2 MoM - UTD 混合方法与并行计算	137
9.3 迭代 MoM - PO 混合方法与并行计算	138
9.3.1 电流定义与求场公式	138
9.3.2 迭代 MoM - PO 混合方法	139
9.3.3 并行迭代 MoM - PO 方法	141
9.4 性能测试与应用	142
9.5 小结	145
参考文献	146
附录 A 简单 PC Cluster 的组建与应用	148
A.1 PC Cluster 的一般架构	148
A.2 PC Cluster 的组建与应用实例	149
A.2.1 硬件平台	149
A.2.2 软件安装与设置	150
A.2.3 MPI 并行程序的运行方法	151
参考文献	154
附录 B 并行 FDTD 程序实例	155
参考文献	167

第一章 电磁场并行计算基础

随着计算机技术的飞速发展与应用日益广泛，人们已越来越清楚地认识到“计算”已经成为与“理论”和“实验”并列的第三种重要科学研究手段。除了计算数学以外，计算物理、计算力学、计算化学、计算电磁学等也都已有了长足的发展。计算机数值模拟与辅助设计已成为国防和很多尖端技术中必不可少的手段。例如，石油工业中的“油藏模拟”，就是流体力学与计算机及计算数学结合的产物。另一方面，计算机模拟与辅助设计在各个学科和工程领域中日益显示出它们的重要性，也对计算机的性能以及计算方法的效率提出了更高的要求。

在计算机计算技术中，并行计算是一种现代计算技术^[1,2]，它在科学研究、工程技术以及军事等方面的应用，已经取得了巨大的成就。所谓并行，是指有一个以上的事件在同一时刻或同一时间段内发生。并行计算就是利用多处理器或多计算机，将进程相对独立地分配于不同的节点上，由各自独立的操作系统调度享有独立的 CPU 和内存资源（内存可以共享），进程间通过消息传递相互交换信息。进程是运行在一个计算机节点上的一个程序，多个进程可以运行在同一个节点上。在消息传递系统中，即使是在同一个节点上运行的多个进程，它们之间的数据通信也是通过消息传输来实现的。为了提高效率，在消息传递系统中，一般一个节点只运行一个计算进程，本书即采用这种方式。

采用并行计算，是因为：

- (1) 可以加快计算速度，即在更短的时间内解决相同的问题或在相同的时间内解决更多更复杂的问题；
- (2) 可以扩大计算规模，特别是对一些新出现的有巨大挑战性的问题，不使用并行计算是无法解决的；
- (3) 可以节省投入，以较低的投入完成串行计算使用较高成本才能够完成的任务。

现代电磁仿真，通常面临一些计算量巨大的问题，比如飞机、军舰平台中天线的电磁特性分析。此时，把任务分解进行并行计算是人们自然想到的一种解决途径。实施电磁场并行计算通常需要考虑三个方面的因素，如图 1.0-1 所示。

由图 1.0-1 不难看出，要进行电磁场并行计算，我们需要硬件平台（比如 PC 集群系统^[3,4]），需要并行编程支持软件（比如信息传递接口——MPI 的具体实现^[5]），最后是进行电磁场数值方法的并行算法设计。接下来，我们将对此逐一进行介绍，并对本书中采用的具体硬件平台、软件环境给予说明。

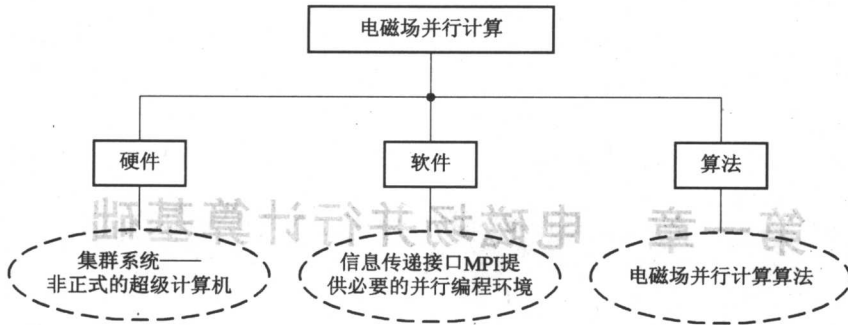


图 1.0-1 电磁场并行计算

1.1 硬件平台

并行计算机即能在同一时间内执行多条指令或处理多个数据的计算机，并行计算机是并行算法实现——并行计算的物理载体^[2]。下面简单介绍并行计算机的不同分类方式。

1.1.1 依据指令与数据划分

1972年，Micheal Flynn 根据一个并行计算机能够同时执行的指令与处理数据的多少对计算机的体系结构进行了分类，这就是所谓的 Flynn 分类法。Flynn 将冯·诺依曼型的计算机划分为四种基本类型，即 SISD、MIMD、SIMD、MISD，如图 1.1-1 所示^[3]。传统的顺序执行的计算机在同一时刻只能执行一条指令（即只有一个控制流）、处理一个数据（即只有一个数据流），因此被称为单指令流单数据流（SISD: Single Instruction Single Data）计算机。而对于大多数并行计算机而言，多个处理单元都是根据不同的控制流程执行不同的操作，处理不同的数据，因此，它们被称作是多指令流多数据流（MIMD: Multiple Instruction Multiple Data）计算机。

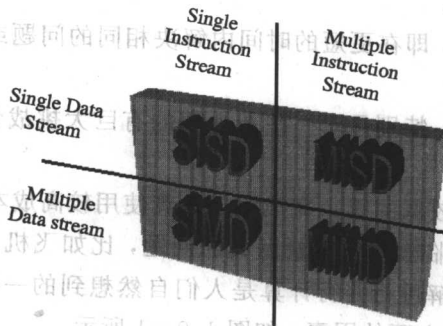


图 1.1-1 Flynn 分类法

曾经在很长一段时间内成为超级并行计算机主流的向量计算机除了有标量处理单元之外，最重要的是具有能进行向量计算的硬件单元。在执行向量操作时，一条指令可以同时多个数据（组成一个向量）进行运算，这就是单指令流多数据流（SIMD: Single Instruction

Multiple Data)的概念。

第四种类型即所谓的多指令流单数据流(MISD: Multiple Instruction Single Data)计算机。在这种计算机中,各个处理单元组成一个线性阵列,分别执行不同的指令流,而同一个数据流则顺次通过这个阵列中的各个处理单元。

这种系统结构只适用于某些特定的算法,相对而言,SIMD和MISD模型更适用于专用计算。在商用并行计算机中,MIMD模型最为通用,SIMD次之,而MISD最少用。

值得注意的是,这种分类方法基本上是概念上的,并不是绝对的。随着并行计算机组织方式的发展,人们又提出了单程序多数据(SPMD: Single Program Multiple Data)并行计算机和多程序多数据(MPMD: Multiple Program Multiple Data)并行计算机。这种划分方式所依据的执行单位不是指令而是程序,其划分粒度要大得多,是从更高的层面上来划分并行计算机的工作方式。

1.1.2 依据存储方式划分

从物理上划分,共享内存(见图 1.1-2)和分布式内存(见图 1.1-3)是两种基本的并行计算机存储方式。除此之外,分布式共享内存也是一种越来越重要的并行计算机存储方式。

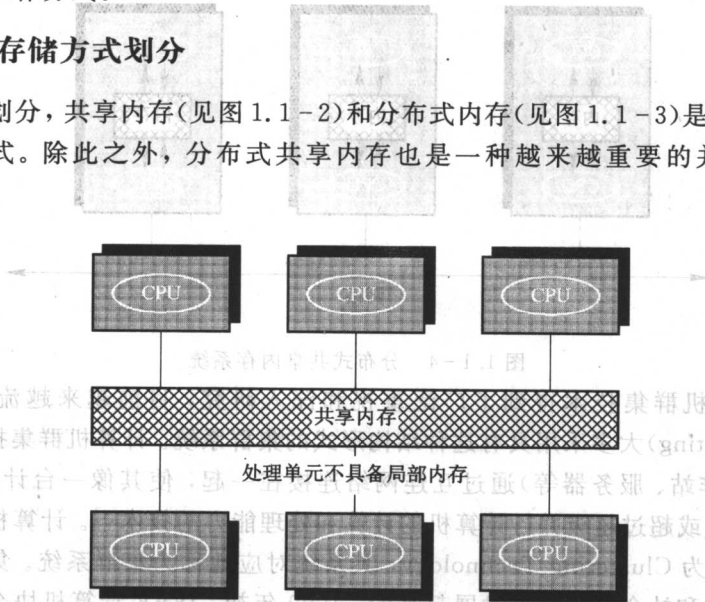


图 1.1-2 共享内存系统

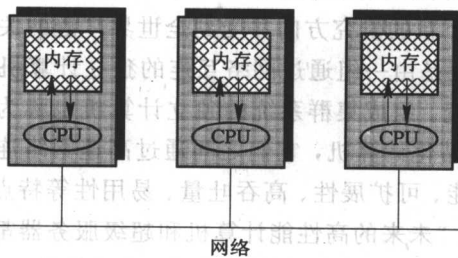


图 1.1-3 分布式内存系统

对于共享内存的并行计算机,各个处理单元通过对共享内存的访问来交换信息与协调各处理器对并行任务的处理。对这种共享内存的编程,实现起来相对简单,但共享内存往往成为并行性能特别是扩展性的重要瓶颈。

对于分布式内存的并行计算机，各个处理单元都拥有自己独立的局部存储器。由于不存在公共可用的存储单元，因此各个处理器之间通过消息传递来交换信息与协调和控制各个处理器的执行。

不难看出，通信对分布式内存并行计算机的性能有重要的影响。复杂的消息传递语句的编写成为在这种并行计算机上进行并行程序设计的难点所在，但是，由于这种类型的并行计算机具有很好的并行性能特别是扩展性，因此它的应用非常广泛。分布式共享内存的并行计算机结合了前两者的特点，是当今新一代并行计算机的一个重要发展方向。如图 1.1-4 所示，每个 CPU 都有自己的私有存储空间，数据从一个处理器到另一个处理器只有通过网络来传输，而共享存储中的多个 CPU 可以同时访问同一内存单元。

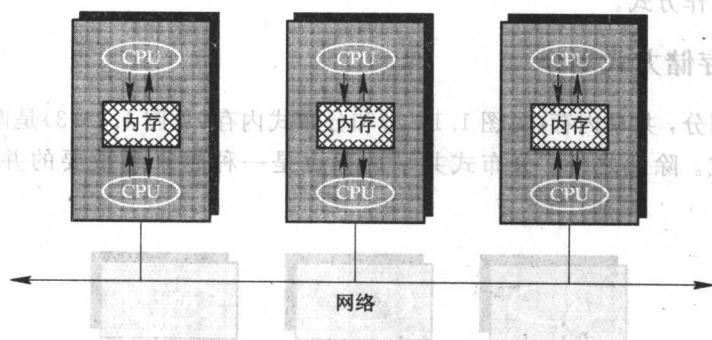


图 1.1-4 分布式共享内存系统

最近，计算机群集技术得到了广泛重视和深入研究，目前越来越流行的集群计算 (Cluster Computing) 大多采用具有这种结构形式的集群系统。计算机群集技术就是将若干台 PC 机(或工作站、服务器等)通过互连网络连接在一起，使其像一台计算机一样工作，从而获得接近于或超过超级并行计算机的计算和处理能力的技术^[4]。计算机群集技术简称集群技术，英文为 Clustering Technology，其系统对应地称为集群系统。集群技术是随着科学技术的发展和社会应用需求发展起来的，1999 年初，IEEE 计算机协会成立了集群计算任务组 (TFCC: the Task Force on Cluster Computing)，并于 1999 年 12 月在墨尔本举办了第一届集群计算国际会议 IEEE Cluster Computing 1999 (Workshop)。举办专门的国际会议表明它已经成为一个新的研究方向并受到全世界广泛的关注。

简单地说，集群系统就是由一组通过网络互连的独立计算机构成的、全部计算资源可一体化的并行或分布式系统。组成集群系统的独立计算机也称为节点。集群系统中，各个节点采用的都是标准的商品化计算机，它们之间通过高速网络连接起来，因此集群系统在较低的费用下，具有高性能、可扩展性、高吞吐量、易用性等特点，目前已成为并行处理的热点和主流。据专家预测：“未来的高性能计算机和超级服务器都将基于集群系统结构”。

集群系统的分类方式很多。按照节点硬件分类，由 PC 机、工作站、服务器或 SMP 组成的集群系统则分别称作 PC 集群 (COP: Cluster of PCs)、工作站集群 (COW: Cluster of Workstations)、服务器集群 (COS: Cluster of Servers) 和 SMP 集群 (CLUMP: CLUster of sMP)。其中工作站集群 (COW) 最初也称为工作站网络 (NOW: Networks of Workstations)。按照节点操作系统分类，集群可以分为 Linux 集群、NT 集群等。按照节点体系结

构的类型, 集群可分为同构集群、异构集群等。按照应用目的(科学计算/商业计算)分类, 集群可分为高性能集群和高可用性集群。根据与外部网络的连接关系, 可以将集群简单地划分为封闭式集群和开放式集群两类^[3]。封闭式集群将节点隐藏于网关之外, 因此不需要 IP 地址, 具有较高的网络安全性, 这更有利于计算任务的完成。开放式集群系统对集群外部而言是可见的, 它需要更多的 IP 地址, 网络安全性需要特别注意, 但它具有比较灵活的特性, 这有利于网络与信息服务等任务。

集群系统的应用领域非常广泛, 几乎所有传统的并行计算领域, 比如涉及区域分解、解线性方程组、有限差分等都可以应用。尽管对普通用户来说, 集群技术的应用还不像把主机与显示器连在一起那么容易, 但对计算中心与研究部门来说已不是像制造计算机那样遥不可及。在计算资源成本有限的情况下, 本书研究工作的硬件平台采用的是 10 台 Dell PC 机用 1000M(B)交换机与高速网卡连接起来的 Windows 集群系统, 我们将在附录 A 中对其进行具体说明。书中计算结果除特殊说明外均是在这个封闭式的 PC 集群系统中获得的。

1.2 软件环境

为了了解并行软件环境, 我们首先介绍并行编程模型^[2]。如果能对计算机的系统结构进行高度的抽象, 给出一个简洁的概念模型, 那么, 程序员在编写程序时, 就不需要了解硬件结构的具体细节, 这种抽象模型就是我们所说的编程模型。共享地址空间、消息传递以及数据并行是最常见的三种并行编程模型。

我们可以将共享地址空间模型看作一个公告牌, 各个节点上的多个进程共享它们的一部分地址空间, 并通过简单的读/写指令(Load/Store)来存取其中的数据。

消息传递模型则有点像邮政系统, 节点之间是通过一条条的消息来协同工作的, 每条消息都明确地标识出发送进程和接收进程的地址(或编号)。在这种方式下, 各进程之间并没有能共同访问的全局共享地址空间。

第三类是所谓的数据并行模型。在这类计算机中包含有较多的处理单元, 它们首先分别对同一个数据集中的不同数据进行并行的计算, 相互交换计算的结果并进行协调, 然后再继续做下一步运算。

这几种并行模型中, 消息传递模型是目前大量的并行程序设计的主要编程模式。目前已经存在许多通用且成熟的消息传递软件包, 其中应用比较广泛的并行程序开发环境是并行虚拟机(PVM: Parallel Virtual Machine)和信息传递接口(MPI: Message Passing Interface)。

PVM 是由美国田纳西大学、奥克里季国家实验室等研制的并行程序开发环境。它可以把多个异构的计算机组织起来使之成为一个易于管理的、可扩展的、易编程使用的并行计算资源。它的各个计算节点可以是共享存储或分布式存储的多处理机或者是向量超级计算机, 专用的图形、标量工作站。这些异构的计算节点可以通过多种网络(比如 Ethernet、FDDI 等等)互联, 成为一个网络计算虚拟机。用户的计算任务被分配到各个计算节点上, 多个节点并行运算, 从而实现粗粒度的并行。PVM 的免费、开放以及易用使其成为一个被广泛接受的并行程序开发环境, 有很多并行机公司都宣布支持 PVM, PVM 可以安装到各

种 Unix、Windows 操作系统上运行，而所有这些又都有力地促进了 PVM 的推广。

MPI 是并行计算机的消息传递接口标准，是全球工业、政府和科研部门联合推出的适合进程间进行标准消息传递的并行程序设计平台。最初版本 MPI 1.0 于 1993 年 2 月推出，目前较新的为 MPI-2，是在对原来的 MPI 作了重大扩充的基础上，于 1997 年 7 月推出的。它实际上是一个消息传递函数库的标准说明，是目前国际上最流行的并行编程环境之一，是分布式存储的可扩展并行计算机和 workstation 网络的一种编程范例。

MPI 没有简单地指定某系统为标准，而是吸取了许多已经广泛应用的消息传递系统的特点。它在采纳已有系统优点的基础上，增加了许多新的特点，从而使其成为非常有吸引力的标准。同 PVM 相比，MPI 有下列特点：

(1) MPI 只是一个支持并行计算的程序库，并不是一个并行操作系统。

(2) 比起 PVM，MPI 更模块化。

(3) PVM 支持异构的集群，而 MPI 标准虽然也可支持异构集群，但它并没有强制性地要求每个厂商都提供支持异构的实现。

(4) MPI 的语义更精确。我们在 MPI 程序库中，能够找到各种各样的发送和接收释义。

(5) 作为一种消息传递接口标准，MPI 没有严格要求底层的通信协议。它只是一种针对应用程序的通信程序库，至于底层的硬件、协议完全由厂商自己决定。

概括起来，MPI 具有功能强大、性能高、适应面广、使用方便以及可扩展性好等优点，其缺点是进程个数不能动态地改变。

MPI 本身是一个规格很严密的通信标准，主要的功能是处理各个进程之间的信息交换。需要注意的是，MPI 并不是一套软件，而 MPICH 才是符合 MPI 这个通信协议标准的一套软件。MPICH^[5] 软件由 Mathematics and Computer Science Division 的 Argonne 实验室研发，详细资料可以参考：<http://www-unix.mcs.anl.gov/mpi/mpich/>。我们可以经由 MPICH 这个软件提供的 MPI 函数库（可连接到用户应用程序中实现消息发送、消息接收以及其他消息传递操作的一组函数的集合）来达成并行运算的功能，这使得程序开发者无需考虑通信节点上面的问题，而可以将程序开发的重心放在程序本身的问题上面。

本书的研究工作将基于 MPI 并行环境，其具体实现采用 MPICH1.2.5 版本。当然，本书中针对不同电磁场数值分析方法给出的并行计算方案，也同样适用于 PVM 并行环境。此外，由于 MPI 自身的特点，本文的研究工作可直接移植到其它操作系统的集群或者高性能工作站中。

1.3 并行算法

算法是解题的精确描述，是一组有穷的规则，它规定了解决某一特定类型问题的一系列运算。并行计算是可同时求解的各进程的集合，这些进程相互作用和协调动作，并最终获得问题的求解。并行算法就是对并行计算过程的精确描述。

并行算法可以从不同的角度分为数值计算并行算法和非数值计算并行算法，同步并行算法和异步并行算法，共享存储并行算法和分布存储并行算法。

数值计算是指基于代数关系运算的计算问题,如矩阵运算、多项式求值、线性代数方程组求解等。科学与工程中的计算问题如计算力学、计算物理、计算化学等一般是数值计算问题。求解数值计算问题的算法称为数值算法(Numerical Algorithm)。非数值计算是指基于比较关系运算的一类诸如排序、选择、搜索、匹配等符号处理,相应的算法称为非数值算法(Non-numerical Algorithm)。非数值计算在符号类信息处理中获得了广泛应用,如数据库领域的计算问题、海量数据挖掘等。

本书介绍的电磁场并行计算属于数值并行计算,后续章节将结合不同的电磁场数值分析方法,具体介绍基于消息传递并行编程模型的并行算法设计以及性能测试情况。

1.4 并行性能评测与影响因素

我们知道,并行计算性能评测与并行计算机体系结构、并行算法、并行程序设计一起构成了“并行计算”研究的四大分支。为了方便地、可比较地评价并行计算系统的性能,人们提出了许多测试基准。了解这些基准对于客观公正地评价并行计算系统非常重要,其中主要基准有以下三种。

1. 加速比

在并行计算系统上进行计算的主要目标就是要加速整个计算过程,所以研究并行系统(并行算法、并行程序)的加速性能十分重要。理想情况下,若程序的每一部分均能完全并行,使用 P 个处理器的并行算法的运算速度应是使用一个处理器时的 P 倍。但实际上这是不可能的,因此需要用加速比去衡量程序并行执行效率。加速比是指采用多个处理器进行计算时计算速度所能得到加速的倍数,以下分别为 P 个处理器加速比与并行加速比的定义。

定义 1: P 个处理器加速比:

$$S_p = \frac{t_1}{t_p}$$

其中, t_1 是用单个处理器在并行环境下求解某个问题所需的时间; t_p 是 P 个处理器并行求解某个问题所需的时间。

定义 2: 并行加速比:

$$S'_p = \frac{t_{\text{seq}}}{t_p}$$

其中, t_{seq} 是用单个处理器在串行环境下求解某个计算问题所需的时间。

通常, $S'_p < S_p$,因为单个处理器在并行环境下的时间开销一般大于该处理器在串行环境下的时间开销。

2. 并行效率

并行效率是与加速比相关的概念。一个并行程序的效率定义为 $E_p = S_p / P$,其中, P 为处理器的个数。当加速比 S_p 接近 P 时,效率 E_p 接近于1。

影响并行效率的因素很多。首先,不能期望一个程序的所有部分都能完全并行,例如,