HIGH PERFORMANCE COMPUTING
IN ENGINEERING SCIENCE

工程科学中的高性能计算

姜弘道　张健飞　秦忠国
陈　林　张　磊　编著

科学出版社

工程科学中的高性能计算

姜弘道　张健飞　秦忠国
陈　林　张　磊　编著

科学出版社
北京

内 容 简 介

在工程科学的各个领域，基于高性能计算的数值模拟（仿真）已得到越来越多的应用。本书围绕数值模拟工程科学问题中广泛用到的有限元法的并行计算，详细介绍所涉及的各方面知识与方法，包括并行计算概念、并行计算机体系结构、基于消息传递的并行编程环境 MPI、线性方程组的并行直接解法和并行迭代解法以及两个开源的并行解法软件包 PETs 和 Aztec，最后介绍并行有限元法中的区域分解法与 EBE 方法。为了便于理解与掌握，许多章节均带有算法或程序以及它们的详细说明，其中包括完整的线性方程组并行直接解、并行迭代解程序、基于 PCG-EBE 的并行有限元法程序以及基于 PETs 的有限元解法器。

本书可用作工科高等院校本科生、研究生相关课程的教材或自学参考书，也可供从事并行计算的工程技术人员学习使用。

图书在版编目(CIP)数据

工程科学中的高性能计算/姜弘道等编著. —北京：科学出版社，2013
ISBN 978-7-03-037336-6

Ⅰ.①工… Ⅱ.①姜… Ⅲ.①工程计算 Ⅳ.①TB115

中国版本图书馆 CIP 数据核字(2013)第 079550 号

责任编辑：童安齐 袁莉莉 / 责任校对：柏连海
责任印制：吕春珉 / 封面设计：耕者设计工作室

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
*
2013 年 5 月第 一 版 开本：B5（720×1000）
2013 年 5 月第一次印刷 印张：16
字数：310 000
定价：48.00 元
（如有印装质量问题，我社负责调换〈双青〉）
销售部电话 010-62136131 编辑部电话 010-62137026(BA08)

前　言

科学技术发展到今天，数值模拟（仿真）已成为人类认识世界、改造世界的新的重要手段，它主要用来解决不可能进行实验以及进行实验代价太大的问题，已被广泛用于科学、工程、生产领域。

数值模拟（仿真）是指利用对科学现象或工程系统建立的模型运用数值计算方法，在计算机上复现实际系统中发生的本质过程，并通过系统模型的数值实验研究存在的或设计中的系统。它将传统科学与工程领域的知识和技术与计算机科学、数学、物理以及社会科学领域中的知识和技术融合在一起，从而对影响工作和生活的所有方面进行更好的预测和优化。由于计算科学以及计算机系统的快速发展，模拟（仿真）的理论和技术已对科学、工程领域产生越来越大的影响，在不久的将来，将成为各领域科学技术发展的关键性因素。就工程科学而言，这种关键性因素主要表现在：

（1）计算机建模和仿真使我们能够探索包含挑战性分析、测量和实验方法的自然事件与工程系统，其结果必将是基于科学的计算模型取代经验主义的假设。

（2）建模和仿真将应用在从微处理器到城市基础设施、大型水利、土木工程和汽车、飞机制造的各类技术中，新的仿真方法必将为全部技术奠定基础。

（3）建模和仿真使我们能够以更多的科学基础，更少的试验和错误以及更短的设计周期设计和制造材料与产品。

（4）建模和仿真将极大地提高在特定设计和决策投入资源前预测结果和优化方案的能力。

（5）建模和仿真有助于拓宽处理相对传统方法十分复杂的问题的能力。例如，涉及时间和空间的多尺度问题，多物理过程问题以及包含不确定性的未知层次问题。

（6）建模和仿真将引进各种工具与方法，应用到所有工程学科中。所有工程学科可以从优化、控制、不确定性测量、验证和证明、设计决策和实时响应的进步中获益。

鉴于当今工程科学问题的复杂性，计算机模拟（仿真）的特点是模型复杂、规模宏大。因此，模型再好，不应用高性能计算，不可能实现真正意义上的仿真。

高性能计算，顾名思义，是在运算速度快、存储容量大、可靠性高的高性能计算机上为解决大规模科学与工程问题所实施的计算。目前任何高性能计算都离不开并行计算，高性能计算机几乎全是并行计算机。因此，工程科学中的高性能计算实

际上是工程科学中的并行计算，要实施工程科学的模拟（仿真）计算，必须对工程科学中的并行计算有所了解。本书编写的目的一是使读者对工程科学中的并行计算有一个概括而又清晰的了解；二是便于工科院校的本科生、研究生和工程技术人员应用并行计算实施工程科学的模拟（仿真）计算。

有限元法在工程科学的模拟（仿真）计算中占有重要地位，已得到广泛应用，因而本书围绕有限元法并行计算组织相关的各方面内容，这些内容对工程科学中模拟（仿真）计算的其他方法也是适用的。

本书共分七章。第一章介绍与并行计算相关的各方面基本概念与基本知识；第二章介绍目前最通用的消息传递型并行编程环境（MPI）；第三、四章介绍并行数值算法，重点是有限元法中的稀疏线性方程组的并行直接解法与迭代解法；第五章介绍一个功能强、效率高、开源的线性（非线性）方程组并行解法软件包 PETSc，并给出应用这个软件包的有限元并行求解器的要点及举例；第六章介绍另一个开源的并行线性解法器 Aztec；第七章进一步分析有限元法的并行性，并重点介绍有限元并行计算的区域分解法和 EBE 方法，给出基于 EBE 方法的程序。

要在一本不太厚的书中介绍并行计算涉及的多个学科的内容，是一件十分困难的事情。作者希望本书通过紧密围绕有限元并行计算的应用介绍各方面内容，对工科非计算机专业的学生与相关从业人员来说，可能是一个了解并进而掌握并行计算的一条较好的途径。

本书具体的编写分工是：张健飞编写第三、第四、第七章，秦忠国编写第二、第五章与第 3.7 节，陈林编写第一章，张磊编写第六章，姜弘道主持本书的策划与定稿工作。鉴于本书涉及多学科内容，作者们受学识所限，不免会有诸多不足与缺点，恳请读者不吝指正。

目　　录

第一章　并行计算概述

本章主要介绍并行计算的一些基础知识。首先，给出并行计算的一个简单定义，并讨论并行计算的主要研究目标和内容，以及推动并行计算发展的主要动力。其次，简要介绍并行计算赖以存在的硬件平台——并行计算机的体系结构以及运行在并行计算机上的操作系统和并行编程环境，使读者对并行计算机系统有一个大致的了解。最后，简要讨论并行算法，以并行计算力学为例说明并行计算在工程科学的应用[1~6]。

1.1　并行计算及其内容

并行计算是伴随并行计算机的出现在近30年来迅速发展的一门交叉学科，内容包括并行计算机体系结构、操作系统、并行算法、并行编程技术、并行性能优化与评价及并行应用等。因此，很难全面精确地给出并行计算的定义。但是，作为一门交叉学科，并行计算可以定位为连接并行计算机系统和实际应用问题之间的桥梁，它为科学研究、工程计算及商业应用等领域的专家在并行计算机上求解本领域问题提供具有共性的关键支撑。

1.1.1　什么是并行计算

我们考虑下述问题：

图书馆将一些书按类上架，而书架依据书的类别分组摆放，可以有多种上架方式。若由一工人单独完成，不能按某一要求的进度完成任务；若由多个工人完成，假定每次一人仅往书架上放一本书，可以有以下两种做法。

(1) 将所有书籍平均分配给每个人去完成。这种划分方法不是很有效，原因是每个工人为了将书上架必须走遍所有的书架。

(2) 将所有的书架分成一些组，且平均分配给各个工人负责，同时将所有图书平均分配给每个工人去上架。如果工人发现一本书属于自己所负责的书架，则将其放入书架，否则将这本书传给所在书架对应的工人。这种分法的效率比较高。

上述例子说明，将一个任务划分成一些子任务，分配给多个工人去完成，工人之间相互合作、并在需要时相互传递图书，这种和谐协调的工作方式可较快地完成任务。

并行计算是按照上述原理来完成的，它涉及与并行计算相关的两个概念。

(1) 任务划分。将书架与图书平均分配给所有工人为任务划分的一个例子。

(2) 通信。工人之间传递图书为子任务间通信的一个例子。

并行计算(Parallel Computing)是指在并行计算机上,将一个应用分解成多个子任务,分配给并行计算机上不同的处理器,各个处理器之间相互协同,并行地执行子任务,从而达到加快求解速度,或者提高求解应用问题规模的目的。

因此,为了成功开展并行计算,必须具备以下两个基本条件:

(1) 硬件资源,包括并行计算机和高速网络。并行计算机至少包含两台或两台以上处理机,这些处理机通过高速网络相互连接,相互通信。

(2) 软件资源,包括操作系统、并行编程技术和并行算法。并行计算的操作系统需要具有较高的稳定性和可靠性;并行编程技术使并行计算机可以快速地求解应用问题;并行算法使应用问题在并行计算机上的求解更有效。

此外,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。

对于具体的应用问题,采用并行计算技术的主要目的在于以下两个方面:

(1) 加快求解问题的速度。

(2) 扩大求解问题的规模。

并行计算之所以必需,主要在于串行计算无法满足大规模科学与工程计算及商业应用的需求,而并行计算是目前唯一能满足实际大规模计算需求的支撑技术。并行计算之所以可行,主要在于具有实际应用背景的计算问题在许多情况下都可以分解为能并行计算的多个子任务。

综上所述,并行计算的主要目标在于在并行计算机上解决一批具有重大挑战性计算任务的科学、工程及商业计算问题,满足不断增长的应用问题对速度和内存资源的需求。

1.1.2 并行计算的内容

由并行计算的两个必备条件可知,并行计算的主要研究内容大致可分为以下四个方面:

(1) 并行计算机的高性能特征抽取。其主要任务在于充分理解和抽取当前并行计算机体系结构的高性能特征,提出实用的并行计算模型和并行性能评价方法,指导并行算法的设计和并行程序的实现。

(2) 有效的并行算法设计与分析。其包括并行计算模型、并行算法的一般设计方法、基本设计技术、一般设计过程,以及对给定的并行计算机及运行在其上的并行算法进行运行性能评价。

(3) 并行计算机语言。并行计算机语言必须简洁,编程容易,可以有效地实

现。目前，并行编程环境主要有消息传递平台 MPI、共享存储平台 OpenMP 及高性能 FORTRAN(HPF)等，并且新的编程语言与编程模式正在不断出现。结合具体并行算法，可以开发求解应用问题的并行程序。同时，结合并行计算机的高性能计算特征，不断优化并行程序的性能，保证并行程序具有良好的可移植性。

(4)并行应用。这是并行计算研究的最终目的。开发的并行计算程序需要通过实际应用的验证和确认，从而保证并行程序的正确性和效率。同时，结合实际应用出现的各种问题，不断改进并行算法和并行程序。

以上四个方面相互耦合，缺一不可。(1)是开展并行计算研究的基础；(2)、(3)是并行计算研究的核心，也是在并行计算机上高效求解应用问题的基本保证；(4)是并行计算研究的目的，也是验证和确认并行计算研究成果的最有效的途径，同时为(2)、(3)的研究提供取之不尽的源泉。

需要说明的是，并行计算不同于分布式计算(Distributed Computing)。后者主要是指通过网络相互连接的两个以上的处理机相互协调、各自执行相互依赖的不同应用，从而达到协调资源访问、提高资源使用效率的目的，如观众点播、远程驾驭式可视化及电视会议等。但是，它无法达到并行计算所倡导的加快求解同一个应用的速度，或者扩大求解同一个应用的问题规模的目的。对于一些复杂应用系统，分布式计算和并行计算通常相互配合，既要通过分布式计算协调不同应用之间的关系，又要通过并行计算提高单个应用的求解能力。

1.1.3　推动并行计算发展的主要动力

大规模科学与工程计算等应用对并行计算的需求是推动并行计算快速发展的主要动力。长期以来，它们对并行计算的需求是无止境的。例如，我国在拱坝静动力分析的有限元法计算上，经历了从 20 世纪 70 年代数千自由度到八九十年代数万自由度，再到 21 世纪头十年数十、上百万自由度的发展过程，这是与坝体越来越高、地质条件越来越复杂、地震因素越来越突出等各种情况产生的需要相适应的。又如，全球气象预报中期天气预报模式要求在 24h 内完成 48h 天气预测数值模拟，此时内存需求大于 1TB，计算性能要求高达 25 万亿次/s。此外，在天体物理、流体力学、密码破译、海洋大气环境、石油勘探、地震数据处理、生物信息处理、新药研制、湍流直接数值模拟、燃料燃烧、工业制造、图像处理等领域，以及大量基础理论的研究领域，都存在计算挑战性问题，均需要并行计算的技术支持。

除了大规模科学与工程计算应用，微电子技术与大规模集成电路发展是推进并行计算发展的另一个主要动力。从 1986 年到 2002 年，微处理器的性能平均每年提高 50%。这意味着，用户和软件开发人员为了使应用程序性能提高，常常只需等待下一代微处理器的诞生。然而，从 2002 年起，单个处理器性能的改进速度变慢，每年仅改进约 20%。这一不同是戏剧性的，如果微处理器每年性能改进

50%，则在 10 年内性能将提高 60 倍；而当性能改进 20%，则 10 年内性能只能提高 6 倍。这一性能提高上的不同伴随着处理器设计的戏剧性的变化。到 2005 年，大多数微处理器的主要制造商决定将快速提高性能的道路放在并行化方向上，不再试图开发更快的单片处理器，而是开始将多个完整的处理器放在一个单独的集成电路上，并且试图生产更多的具有大量处理器的各类并行计算机，并使其得到广泛应用。这些改变无疑会改变并行计算机的体系结构、操作系统和编译系统及并行编程环境，从而对并行算法和并行编程模式带来新的挑战。当前，如何快速开发并行应用程序，高效率地发挥当前并行计算机的峰值性能，已经成为并行计算研究面临的一个挑战性问题。

1.2 并行计算机体系结构

本节首先简要介绍单处理机体系结构，它是理解并行计算机体系结构的基础，然后转入对并行计算机体系结构以及并行体系结构中一些重要问题的描述。

1.2.1 单处理机体系结构

为了更好地理解并行计算机的体系结构，下面简单介绍单处理机的两个主要部件，即中央处理器(Central Processing Unit，CPU)和存储器(Memory)。

1. 中央处理器

CPU 是电子计算机的核心部件，其功能主要是解释计算机指令以及处理计算机软件中的数据，它具有以下四个方面的基本功能：

(1) 指令控制。程序的顺序控制称为指令控制。由于程序是一个指令序列，这些指令的相互顺序不能任意颠倒，必须严格按照程序规定的顺序进行。

(2) 操作控制。一条指令的功能往往是由若干个操作信号的组合来实现的。因此，CPU 管理并产生由内存取出的每条指令的操作信号，把各种操作信号送往相应的部件，从而控制这些部件按照指令的要求进行动作。

(3) 时间控制。对各种操作实施时间上的定时称为时间控制。在计算机中，各种指令的操作信号以及一条指令的整个执行过程都受到时间的严格定时。

(4) 数据加工。数据加工是对数据进行算术运算和逻辑运算处理。

CPU 的基本部分主要由控制器和运算器组成：

(1) 控制器。其由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成。它是发布命令的“决策机构”，完成协调和指挥整个计算机系统的操作。它的主要功能有：

① 从内存中取出一条指令，并指出下一条指令在内存中的位置。

② 对指令进行译码或测试，产生相应的操作控制信号，以便启动规定的动作。

③ 指挥并控制 CPU、内存和输入/输出(Input/Output，I/O)设备之间数据流动的方向。

(2) 运算器。由算术逻辑单元(Arithmetic Logic Unit，ALU)、累加寄存器、数据缓冲寄存器和状态条件寄存器组成，它是数据加工处理部件。相对控制器而言，运算器接受控制器的命令进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。运算器有两个主要功能：

① 执行所有的算术运算。

② 执行所有的逻辑运算，并进行逻辑测试，如零值测试或两个值的比较。

2. 存储器

存储器是计算机系统中的记忆设备，用来存放程序和数据。计算机中的全部信息，包括输入的原始数据、计算程序、中间运行结果和最终运行结果都保存在存储器中。它根据控制器指定的位置存入和取出信息。有了存储器，计算机才有记忆功能，才能保证正常工作。按用途，存储器可分为主存储器(内存)和辅助存储器(外存)，也可分为外部存储器和内部存储器。外存通常是磁性介质或光盘等，能长期保存信息。内存指主板上的存储部件，用来存放当前正在执行的数据和程序，但仅用于暂时存放程序和数据，关闭电源或断电，数据会丢失。

当计算机运行一个程序时，程序和数据保存在存储器中，由于存储器的访问时间与 CPU 的速度极不匹配，从而提出多级存储结构的概念，即在计算机体系结构中引入容量更小、速度更快的存储层，即高速缓冲存储器(Cache)。它采用速度很快、价格更高的半导体静态存储器，与微处理器存放在一起，存放当前使用最频繁的指令和数据。当 CPU 从内存中读取指令与数据时，同时访问高速缓存与主存。如果所需内容在高速缓存中，就能立即获取；如果没有，再从主存中读取。高速缓存中的内容是根据实际情况及时更换的。这样，通过增加少量成本即可获得很高的速度。Cache 一般分为三级，一级 Cache 嵌入 CPU 芯片内，为寄存器提供数据，它的容量最小，多数系统还有二级或三级 Cache。

Cache 的容量远小于主存的容量，因而不可能将一个进程所使用的数据全部驻留在一级或二级 Cache 中，所以程序在执行过程中，必须由存储硬件来确定哪些存储单元上的数据需要复制到 Cache 中。如果 Cache 已经写满，但还需要复制其他存储单元上的数据，则必须从 Cache 中删除一些数据项(必要时还要写回主存)。若 CPU 请求一个数据，该数据不在 Cache 中，这时产生一次 Cache 失效事件，失效事件发生的概率称为 Cache 失效率，存储系统设计的主要目标是使 Cache 失效率达到极小。同时，失效率依赖于程序的行为和对应的算法。从编程和算法设计的角度来看，为了降低失效率，需要在程序中开发“时间局部性”，即在一个短时间内

重复使用同一个数据，也就是说，在数据被剔出 Cache 之前，充分使用该数据，这就要求编程和算法设计人员认真考虑数据的使用方式。

数据在 Cache 和主存中通常以 64b、128b 和 256b 为单位进行成组传输，其传输单位称为 Cache 线。在一个时刻中移动的是整个 Cache 线，从而允许主存更有效地提供一组数据。如果程序访问第一个数据后紧接着访问相邻的数据，它将发现该数据已在 Cache 中。对于这样的程序，较大的 Cache 线能提高程序的性能。如果程序采用非结构化的访存方式，那么数据读入 Cache 将占据大部分时间，而实际上这些数据并不被使用。对于这样的程序，较大的 Cache 线反而降低程序的性能，这时应选择相对较小的 Cache 线。

1.2.2 并行计算机的基本概念及其分类

1. 并行计算机的基本概念

什么是并行计算机？简单地说，并行计算机是指两台或两台以上处理机，通过高速网络连接起来而形成的并行计算机系统，处理机间相互通信与协作，以便高效、快速地求解大型应用问题。

并行计算机的出现、发展以及广泛应用是基于人们在两方面的认识。第一，单处理机性能不能满足大规模科学与工程问题的计算需求，而并行计算机是实现高性能计算、解决挑战性计算问题的唯一途径；第二，同时性和并行性是物质世界的一种普遍属性，例如对几十份常规应用软件的统计表明 90%左右的串行计算均可以并行化。实际上，针对某一具体应用问题，可以利用它们内部的并行性，设计并行算法，将其分解为相互独立、但彼此有一定联系的若干子问题，分别将各子问题交给各台处理机进行处理，而所有处理机按所设计的并行算法相互通信与协调，共同完成对给定应用问题的求解。

2. 并行计算机的分类方法

对并行计算机的分类有多种方法，其中最著名的是 1966 年由 M. J. Flynn 提出的分类法，称为 Flynn 分类法。Flynn 分类法是根据计算机的运行机制进行分类的，分为以下两类：

(1) 指令流(Instruction Stream)：机器执行的指令序列。

(2) 数据流(Data Stream)：由指令流调用的数据序列，包括输入数据和中间结果。

Flynn 根据指令流和数据流的不同组织方式，把计算机系统的结构分为以下四类：

(1) 单指令流单数据流(Single Instruction Single Data，SISD)。

(2) 单指令流多数据流(Single Instruction Multiple Data,SIMD)。

(3) 多指令流单数据流(Multiple Instruction Single Data,MISD)。

(4) 多指令流多数据流(Multiple Instruction Multiple Data,MIMD)。

其中,SISD 计算机是传统的单处理机,也称为串行计算机。

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,可以把并行计算机分为 SIMD 并行计算机和 MIMD 并行计算机。

SIMD 并行计算机在同一时刻,各处理机(或处理器)执行相同的指令,处理不同的数据。SIMD 并行计算机对并行计算机的发展起到重要的推动作用,由于微处理器芯片技术的发展,单处理机性能非常强大。同时,90 年代后并行计算机均朝着 MIMD 并行计算机方向发展,所以目前 SIMD 并行计算机已退出历史舞台。

MIMD 并行计算机中各处理机(或处理器)可同时执行不同的指令,处理不同的数据。按照存储方式的不同可分为共享存储多处理机[对称多处理机(Symmetric Multiprocessors,SMP)]、分布存储多处理机[大规模并行处理机(Massively Parallel Processer,MPP)和集群(Cluster)]和分布共享存储多处理机(Distributed Shared Memory,DSM)三种类型。

SIMD 并行计算机通常在实际问题包含大量的对不同数据的相同运算(向量运算和矩阵运算)的情况下才发挥其优势。而 MIMD 并行计算机无此要求,它可以适应更多的并行算法,因而可以更加充分地开掘实际问题的并行性。SIMD 并行计算机所使用的 CPU 通常是专门设计的,而 MIMD 并行计算机可以使用通用 CPU。

1.2.3　共享存储与分布存储

不同类型的并行计算机实现多机并行工作的方式不同。其中,按照通信方式进行划分,采用共享公共存储器中数据的方式实现通信的并行计算机称为多处理机(Multiprocessors),通过消息传递的方式实现通信的并行计算机称为多计算机(Multicomputers)。与之相对应,按照组织结构和存储方式进行划分,可将并行计算机分为两种基本的类型,即共享存储多处理机系统和分布存储多计算机系统。

共享存储是具有所有处理器都能访问的物理内存,地址空间统一编码,各处理机采用读写内存中共享数据的方式进行交互的结构模型。在存储器硬件结构的实现方法上,可将其分为集中式共享存储器和分布式共享存储器两种,前者由单一的存储器构成,后者由统一地址空间编码的多个存储器构成。共享存储的多处理机是一种紧耦合型的系统,其优点在于通信机制简单,使程序开发更加简便,可移植性更好。但是,由于共享访问存储介质,处理器的访存过程需要经历竞争和延迟,这将在一定程度上制约共享存储多处理机的速度。

分布存储是多个处理机拥有自己独立的存储器,彼此之间不共享,处理机之间