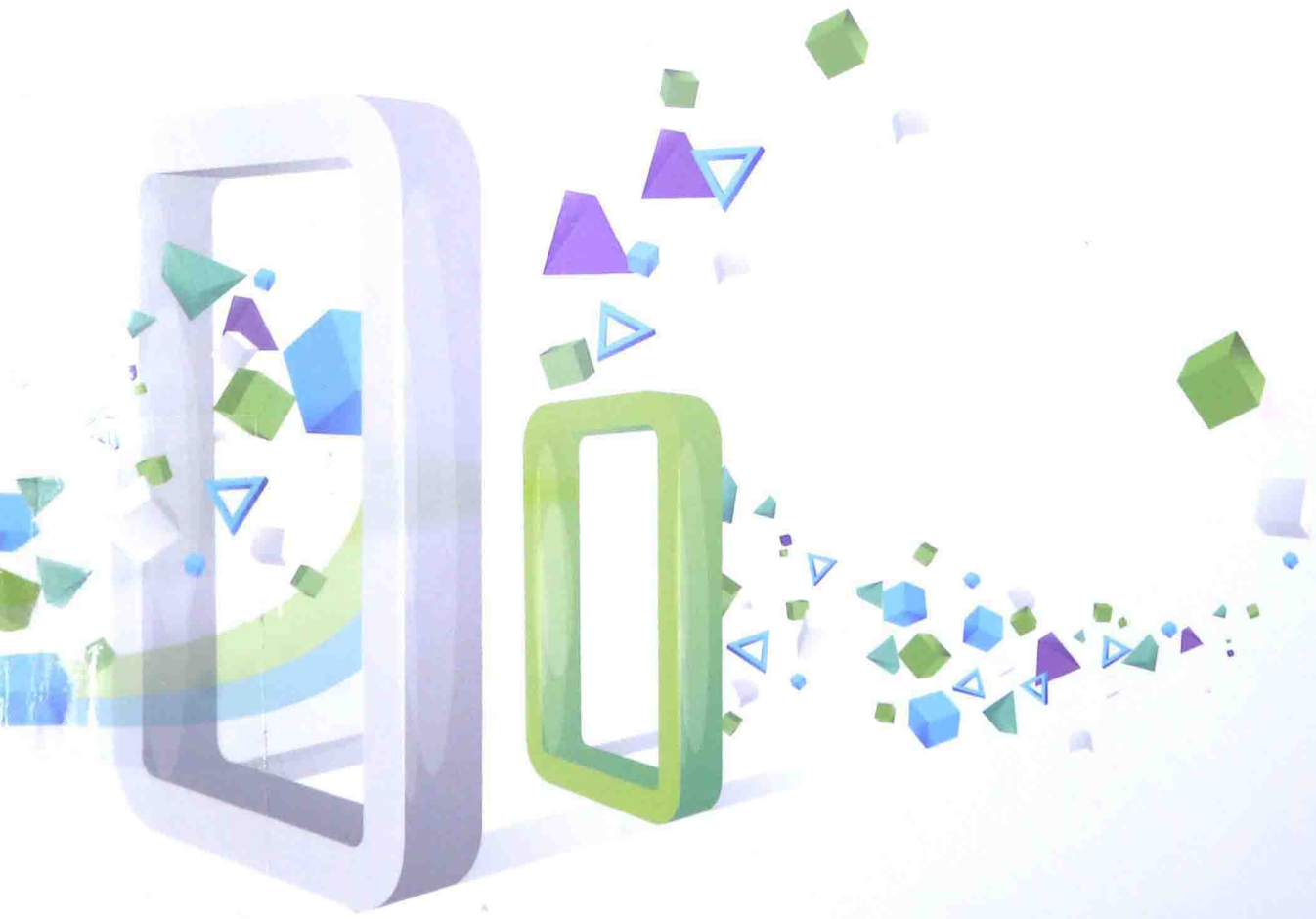
普通高等院校“十二五”规划教材

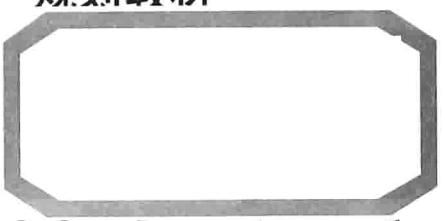
并行计算与程序设计

刘其成　胡佳男　孙雪姣　毕远伟　童向荣　编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校“十二五”规划教材



并行计算与程序设计

刘其成　胡佳男　孙雪姣　毕远伟　童向荣　编　著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书对并行计算的理论知识和并行程序设计方法进行了系统的介绍，包括并行计算基本概念、并行计算机基础、并行计算模型、并行计算性能评测、并行算法设计基础、OpenMP 多线程并行程序设计、MPI 消息传递并行程序设计、Windows 线程库并行程序设计、Java 多线程并行程序设计等内容。

本书集作者多年的教学经验编写而成，语言通俗易懂，内容安排合理，讲解深入浅出。本书在介绍并行计算理论知识的基础上，特别注重并行程序设计的实践方法及实用性。书中含有大量精心设计并调试通过的程序实例，方便读者参考。

本书适合作为普通高等院校计算机科学与技术专业、软件工程专业以及信息类相关专业本科生和研究生的教材，也可作为社会培训教材或软件开发人员的参考用书。

图书在版编目（CIP）数据

并行计算与程序设计 / 刘其成等编著. — 北京：
中国铁道出版社，2014.6
普通高等院校“十二五”规划教材
ISBN 978-7-113-18390-5

Ⅰ. ①并… Ⅱ. ①刘… Ⅲ. ①并行算法－高等学校－
教材②并行程序－程序设计－高等学校－教材 Ⅳ.
①TP301.6②TP311.11

中国版本图书馆 CIP 数据核字(2014)第 107439 号

书　　名：并行计算与程序设计
作　　者：刘其成　胡佳男　孙雪姣　毕远伟　童向荣　编著

策　　划：刘丽丽　　**读者热线：**400-668-0820
责任编辑：周　欣　彭立辉
封面设计：刘　颖
责任校对：汤淑梅
责任印制：李　佳

出版发行：中国铁道出版社（100054，北京市西城区右安门西街 8 号）
网　　址：http://www.51eds.com
印　　刷：北京昌平百善印刷厂
版　　次：2014 年 6 月第 1 版　　2014 年 6 月第 1 次印刷
开　　本：787mm×1092mm　1/16　**印张：**13.5　**字数：**337 千
印　　数：1～2 000 册
书　　号：ISBN 978-7-113-18390-5
定　　价：29.00 元

FOREWORD

前 言

本书紧密结合相关专业规范，面向高等院校的学生和从事软件开发以及相关领域的工程技术人员，在充分考虑普通高等院校学生实际情况的基础上，覆盖并行计算课程要求的知识单元和知识点。

教学内容、教学方法、教学手段的改革是每个教育工作者的重要任务。编者根据多年的教学和软件开发经验，以使用多年的讲义为基础，对本书的内容取舍、组织编排和实例进行了精心设计。本书在难易程度上遵循由浅入深、循序渐进的原则，特别考虑到普通高等学校本科学生的实际理解和接受能力。与以往许多相关教材主要以理论为主不同，本书突出实用性，将复杂的理论融于具体的实例和程序中。书中的实例经过精心设计挑选，程序代码已认真调试，可以直接运行，为读者理解和使用提供了方便。同时，本书注重培养学生的自学能力和获取知识的能力。在编写过程中，力图在内容编排、叙述方法上为教师留有发挥空间的同时，也为学生留下自学空间。

全书共分 9 章：第 1 章通过一系列实例，介绍了并行计算和并行程序设计的基础知识，以及一些相关概念；第 2 章介绍了并行计算机的体系结构、并行计算机的分类，以及多核技术和 GPU 技术；第 3 章介绍了并行计算模型的概念，以及 PRAM 模型、LogP 模型和 BSP 模型，并且对几种常见模型进行了比较；第 4 章介绍了并行计算性能评测的基本概念、并行系统的性能分析方法、并行系统可扩展性度量指标；第 5 章介绍了并行算法的设计方法、并行算法的设计过程、并行算法设计技术；第 6 章介绍了 OpenMP 主要编译指导语句、OpenMP 主要运行时库函数、OpenMP 主要环境变量、OpenMP 多线程程序性能分析；第 7 章介绍了 MPI 消息传递接口及 MPICH 实现、MPI 编程基础、MPI 点对点通信和 MPI 群集通信方法；第 8 章介绍了 Windows 线程库基本知识、Win32 API 线程函数、MFC 和.NET Framework 线程处理相关类库；第 9 章介绍了 Java 线程的相关知识、Java Runnable 接口与 Thread 类实现多线程的方法，以及 Java 解决线程同步与死锁的方法。

本书由刘其成、胡佳男、孙雪姣、毕远伟、童向荣编著，其中第 1 章由毕远伟编写，第 2 章由童向荣编写，第 3~4 章、6~8 章由刘其成编写，第 5 章由孙雪姣编写，第 9 章由胡佳男编写。另外，张莹莹、邵珠方参与了部分内容的编写，郑文静绘制了部分图形。刘其成设计了全书的结构，并对全书进行了统稿。

在本书的编写过程中，参阅了大量书籍和其他相关资料，得到了张伟教授及中国铁道出版社的支持和帮助，在此表示衷心感谢。

尽管书稿几经修改，但由于作者学识有限，书中难免有疏漏与不当之处，恳请各位同仁、读者不吝赐教。

编 者

2014 年 4 月

CONTENTS

目 录

第1章 概述

学习目标

- 了解并行计算常见实例；
- 掌握并行计算定义等基础知识；
- 了解并行程序设计基础知识；
- 掌握并行计算相关概念。

本章首先从各个角度给出了并行计算的多个实例，通过实例讲述了并行计算的基础知识和相关概念。并行计算的基础知识包括并行计算的重要性、定义、应用分类、研究内容和并行设计的方法、应用系统的并行性。并行程序设计基础知识包括并行程序设计的思想、方法、并行编程环境和并行程序设计语言。并行计算包括顺序、并发和并行、进程和线程，以及并行算法等一些基本概念。

1.1 实 例

并行计算在许多计算机应用领域都产生了巨大的影响，使原来无法解决的应用问题成为可能解决的问题。例如，卫星数据处理、石油数据处理（连续优化问题）、调度问题、平面性问题及 VLSI 设计（离散优化问题）。现实生活中并行解决问题的方式有很多，例如手脚并用、边听边写等。

1.1.1 求和

下面给出几行求和代码。

```
int a[1000];
for(int sum=0, int i=0; i<1000; i++)
    sum=sum+a[i];
```

这段程序等价于：

```
int a[1000];              // (1)
for(int sum1=0, int i=0; i<1000; i=i+2)
    sum1=sum1+a[i];       // (2)
for(int sum2=0, int i=1; i<1000; i=i+2)
    sum2=sum2+a[i];       // (3)
int sum=sum1+sum2;        // (4)
```

显然，上述代码 4 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。

1.1.2 泡茶问题

想泡壶茶喝，茶叶有了，但是没有开水；同时水壶需要洗，茶壶茶杯也需要洗。生上火以后，需要做 4 项工作：洗好水壶、洗好茶壶茶杯、准备茶叶、冲开水泡茶。要完成这几项工作，下面三个人用了三种不同的方法：

甲：洗好水壶，灌上凉水，放在火上；在等待水开的时间里，洗茶壶、洗茶杯、拿茶叶；等水开了，泡茶喝。

乙：做好一些准备工作，洗水壶，洗茶壶、茶杯，拿茶叶；一切就绪，灌水烧水；坐待水开了泡茶喝。

丙：洗净水壶，灌上凉水，放在火上，坐待水开；水开了之后，急急忙忙找茶叶、洗茶壶茶杯，泡茶喝。

哪个人的做法节省时间？显然是甲，因为另外两个人的做法都“窝工”了。

假设洗水壶需要 1 min，把水烧开需要 10 min，洗茶壶、茶杯需要 2 min，拿茶叶需要 1 min，而泡茶需要 1 min。甲总共要 12 min（而乙、丙需要 15 min）。如果要缩短工时，提高效率，主要是烧开水这一环节，而不是拿茶叶这一环节；同时，洗茶壶和茶杯、拿茶叶总共需要 3 min，完全可以利用“等水开”的时间来做。

同时，如果有两个机器人，让它们泡茶，最好的方法显然是按照“甲”的做法分工：机器人 A 去烧水，机器人 B 洗茶具；等水开了，泡茶喝。这里应用了分块的思维——把不相关的事务分开给不同的处理器执行。

图 1-1 中，如果由甲一人来完成这个泡茶过程，图中 A 框部分可以并行。如果由两个机器人来完成，而且有不少于 2 个水龙头供机器人使用，那 B 框的部分也可以并行而且能取得更高的效率。可见能够合理利用的资源越多，并行的加速比率就越高。

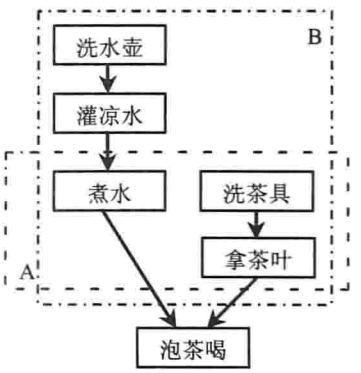


图 1-1 泡茶问题

1.1.3 图书馆新书上架

图书馆新书上架时，书按类上架，书架依据在书库中的位置分成一些组。在书非常多的情况下，如果一个人单独完成，不能按要求的时间完成任务。这时，考虑多个人来完成。如果每次一个人只往书架上放一本书，可以考虑下面两种方法：

（1）所有的书籍平均分配给每个人去完成，每个人将书上架必须走遍所有的书架，这种划分方法不是太有效。

（2）将所有书架分组，且平均分配给各个人负责，同时将所有图书平均分配给每个人去上架。如果某人发现书属于自己所负责书架，则将其放入书架。否则，将这本书传给它所在书架对应的人。这种分法对应的效率比较高。

上述例子说明，将一个任务划分成一些子任务，并分配给多个人去完成，多个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。

上面的图书馆新书上架的例子涉及并行计算相关的两个概念：（1）任务划分，将图书平均分配给所有人。（2）通信，多个人之间传递图书就是子任务通信。并行计算就是严格按照上述原理来完成的。

1.1.4　天气预报

考虑 3 000 km×3 000 km 的范围，垂直方向的考虑高度为 11 km。将 3 000 km×3 000 km×11 km 的区域分成若干 0.1 km×0.1 km×0.1 km 的小区域，则将近有 10^{11} 个不同的小区域。另外，还需考虑时间因素，将时间参数量化。假定考虑 48 h 天气预报。

每一小区域的计算包括参数的初始化及与其他区域的数据交换。若每一小区域计算的操作指令为 100 条，则整个范围一次计算的指令为 $10^{11}\times100=10^{13}$，两天的计算次数将近 100 次，因此，指令总数为 10^{15} 条。用一台 10 亿次/秒计算机进行计算，将大约需要 280 h。如果使用 100 个 10 亿次/秒的处理器构成一台并行计算机，每个处理器计算的区域为 10^{8} 个，不同的处理器通过通信来传输参数，若每个处理器的计算能力得到充分利用，则整个问题的计算时间不超过 3 h。

这个例子说明两个问题：①并行计算可以解决原先不能解决的问题。②并行计算可进行更准确的天气预报。

1.1.5　美国 HPCC 计划

美国 HPCC（High Performance Computing and Communication）计划提出了重大挑战性问题，以及对科学与工程具有重大的经济与科学意义的一些基础课题，它们的解可通过高性能并行计算技术得到。

美国 HPCC 计划公布的重大挑战性应用包括：①磁记录技术：研究静磁和交互感应以降低高密度磁盘的噪声。②新药设计：通过抑制人的免疫故障病毒蛋白酶的作用研制治疗癌症与艾滋病药物。③高速民航：用计算流体动力学来研制超音速喷气发动机。④催化作用：仿生催化剂计算机建模，分析合成过程中的酶作用。⑤燃料燃烧：通过化学动力学计算，揭示流体力学的作用，设计新型发动机。⑥海洋建模：对海洋活动与大气流的热交换进行整体海洋模拟。⑦臭氧耗损：研究控制臭氧损耗过程中的化学与动力学机制。⑧数字解析：用计算机研究适时临床成像、计算层析术、磁共振成像。⑨大气污染：对大气质量模型进行模拟研究，控制污染传播，揭示其物理/化学机理。⑩蛋白质结构设计：对蛋白质组成的三维结构进行计算机模拟研究。⑪图像理解：实时绘制图像或动态。⑫密码破译：破译由长位数组成的密码，寻找该数的两个乘积因子。

1.1.6　教务管理系统

很多情况下开发环境和运行环境已经解决了系统的并行问题，下面的教务管理系统是一个典型的例子。但是，需要注意的是，许多系统的并行问题需要在软件的设计和实现中采取一定的措施。

（1）用 C 语言编写一个 Windows 程序，供教务员登记、查阅和统计学生考试成绩，所有课程的成绩表保存在一个文件中。程序是一个顺序程序，每执行一次程序就显示一个对话窗口，教务员可通过多次启动该程序而在多个窗口中进行不同的操作。

程序启动一次，就在 Windows 操作系统的支持下创建一个进程。多次启动就创建多个进程，这个程序以及它使用的文件构成了一个并行的应用系统。因为有特定运行环境的支持，程序员没有解决与进程的并行执行有关的任何问题，Windows 操作系统解决了进程的创建、撤销、调度、资源分配等问题。

（2）用 C 语言编写两个 Windows 程序，一个程序供教务员登记成绩表，另一个程序供本单位的任何教师或学生查阅成绩，两个程序共享同一个保存成绩表的文件。

两个顺序程序连同它们所使用的文件共同构成了一个并行系统。每个程序定义一类进程，每一次启动就创建一个相应的进程。教务员可以在一台终端上把负责登记成绩表的程序启动两次，在两个窗口中分别登记不同课程的成绩表。与此同时，可能有 3 个学生各在一台终端上启动另一个程序来查阅成绩。此时，系统中同时有五个进程在并行运行。进程之间所有的并行处理问题都由 Windows 操作系统解决了，不需要程序员为此做更多工作。

（3）在上例的基础上，采用 C/S 体系结构，允许教务员或者任何一位任课教师通过网络在不同的计算机上登记成绩表，也允许学生在网络的任何结点上查阅成绩。设计方案：成绩表存放在服务器端，由一个数据服务器提供数据查询、数据更新等基本服务；安装在每台客户机上的程序与上例一样，也是一个负责成绩登记的程序和一个负责成绩查询的程序，但它们不是访问本机的文件，而是访问远程的数据库。

服务器上提供的服务以及每台客户机上运行的程序是相互并行的，每一台计算机上的不同进程也是相互并行的。多个客户端的进程并行地访问同一个数据库时，数据库管理系统解决共享、互斥、数据完整性等问题；每台客户机上多个进程之间的并行处理由 Windows 操作系统解决。

（4）在上例的基础上，采用 B/S 体系结构。设计方案：负责成绩登记和成绩查看等功能的所有程序都在服务器上运行，客户机上只剩下一个产生浏览界面的程序。服务器上完成各项功能的进程是并行执行的；一台客户机上也可以开出多个浏览界面，多个进程并行执行。并行处理由数据库管理系统和 Windows 操作系统解决。

1.1.7 地球物理石油勘探数据处理系统

从分配处理器资源这个角度看，无论一个进程还是一个线程都是一个独立的处理器分配单位，都是一个控制流。在设计开始时，问题的关键在于从逻辑上理清这些控制流。此时，可以暂时忽略进程和线程之间的区别，只强调它们都是控制流。在深入考虑实现细节时，再根据操作系统、编程语言等条件决定把它们设计成进程还是线程。

下面的地球物理石油勘探数据处理系统中，把通过数据采集设备获取的数据输入系统，经过数据处理，将地质信息显示在屏幕上。

（1）3 个进程分别负责数据的输入、数据处理和显示。编写 3 个 C 程序，分别用于输入进程、数据处理进程和显示进程的创建，3 个进程并行执行完成系统功能，如图 1-2 所示。

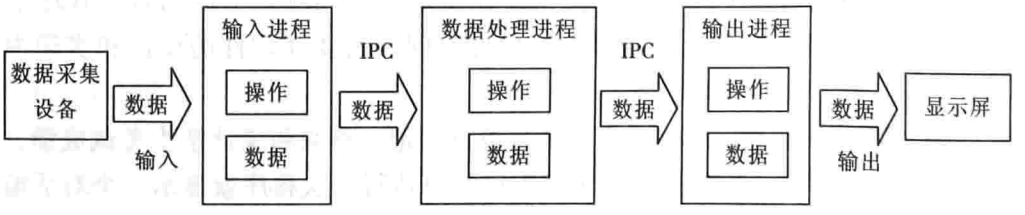


图 1-2　用多进程实现数据的输入、处理和显示

这种设计方案的问题是用每个 C 程序所创建的进程不能共享同一片内存空间中的数据，每个进程能够操作的数据只能是在它分配的数据空间中的私有数据。于是图 1-2 中 3 个进程间只能通过使用进程间的通信（IPC）传送数据信息。由于数据量很大，将使系统性能受到严重影响，成为实时要求较高的系统运行时的瓶颈。

（2）设计方案利用线程实现 3 个控制流。只设计一个进程，用一个 C 程序来实现。在这个程序中定义了 3 个线程，分别负责输入、数据处理和显示，通过线程与线程之间的并行体现系统的并行性，如图 1-3 所示。

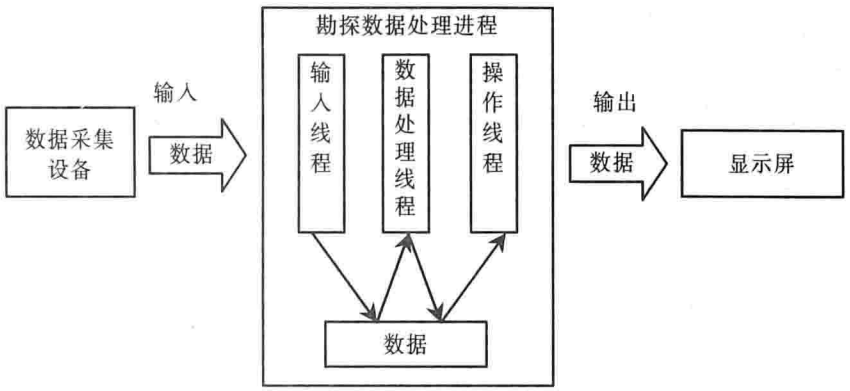


图 1-3　用多线程实现数据的输入、处理和显示

这个方案在一个进程内部这 3 个线程都共享该进程所定义的数据，不需要在不同控制流间传送数据。但是，要考虑多个线程之间数据互访问的实现，处理之前和处理之后的数据要分别用两个数据结构来描述。

（3）石油勘探数据处理系统不仅需要将勘探数据实时地显示出来，而且需要进行更多的处理。可以设计若干进行各种专业化处理的进程，进程内部含有多个线程。各个进程之间的数据交换不太频繁，可以通过文件系统或者数据库管理系统来实现数据共享，也可以在进程之间通过 IPC 或 RPC 传送数据。进程内部各个线程仍然共享进程私有数据空间的数据。系统中既有进程与进程之间的并行，又有线程与线程之间的并行。

1.2　并行计算基础知识

1.2.1　并行计算的重要性

随着科学技术的发展，计算技术取得了前所未有的进展，为各个学科的研究定量化与精确化描述创造了有利条件，并逐渐形成了一门计算性的学科分支——计算科学与工程，如计算物理、计算化学、计算生物学、计算地质学、计算气象学、计算材料科学等。

计算科学与工程、理论科学、实验科学一起成为当今的三门重要科学。这三门科学彼此相辅相成，推动科学发展与社会进步，而并行计算又是计算科学与工程的核心。

并行计算不仅和国家的科技和经济发展密切相关，而且直接影响到国防能力和国家安全，如核爆炸模拟、复杂系统精确解算、基因研究和国家机要通信的加密与解密等。并行计算能力是衡量国家实力的重要标志。

并行计算技术的发展规划一直是美国、欧洲许多国家的一种“国家行为”。美国的 HPCC 和 ASCI 计划分别是在布什和克林顿直接参与下制定的。美国曾列出科学家普遍认为是 21 世纪影响人类社会的一些关键性科学问题，称之为人类面临的“巨大挑战”，这些问题必须依靠高性能计算机平台，采用大规模科学与工程计算来解决，例如分子动力学、遗传基因、海洋循环、飞行物设计等。

当前，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。并行技术向 IT 低端的发展动力十足，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 的层面。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。美国等国家已经开始进行并行计算技术的普及工作：编号为 DUE9554975 的美国国家科学基金项目，题目就是“将并行技术引入一年级课程”。并行计算技术逐渐开始从“阳春白雪”成为大众化技术。

1.2.2 并行计算的定义、并行计算机系统及软件

1. 并行计算的定义

并行计算，又称高性能计算、超级计算，是指同时对多个任务、多条指令，或对多个数据项进行处理。

并行计算的主要目的是提供比传统计算快的计算速度，以及解决传统计算无法解决的问题。也就是说，通过并行计算，可以在要求的合理时限内完成计算任务。例如，制造业一般要求的时限是秒级，短时天气预报（当天）一般要求的时限是分钟级，中期天气预报（3~10 日）一般要求的时限是小时级，长期天气预报（气候）一般要求的时限是尽可能快，而对于湍流模拟可通过并行计算来完成。

2. 并行计算机系统

完成并行计算的计算机系统称为并行计算机系统，它是将多个处理器（可以几个、几十个、几千个、几万个等）通过网络连接以一定的方式有序地组织起来。一定的连接方式涉及网络的互联拓扑、通信协议等，而有序的组织则涉及操作系统、中间件软件等。

并行计算机是由一组处理单元组成的，这组处理单元通过相互之间的通信与协作，以更快的速度共同完成一项大规模的计算任务。因此，并行计算机的两个最主要的组成部分是计算结点和结点间的通信与协作机制。并行计算机体系结构的发展也主要体现在计算结点性能的提高以及结点间通信技术的改进两方面。

3. 并行软件

并行软件的一般开发过程是给定并行算法，采用并行程序设计平台，通过并行实现获得实际可运行的并行程序。在并行计算机上运行该程序，评价该程序的实际性能，揭示性能瓶颈，指导程序的性能优化。

1.2.3 并行计算的应用分类

科学与工程计算对并行计算的需求是十分广泛的，但所有的应用可概括为三方面：

（1）计算密集型：这一类型的应用问题主要集中在大型科学工程计算与数值模拟（气象预报、地球物理勘探等）方面。

（2）数据密集型：Internet 的发展，提供了海量的数据资源，甚至于大数据。但有效地利用这些资源，需要进行并行处理，对计算机的要求也相当高。这些应用包括数字图书馆、数据仓库、数据挖掘、计算可视化等。

（3）网络密集型：通过网络进行远距离信息交互，来解决用传统方法很难解决的一些应用问题。如协同工作、遥控与远程医疗诊断等。

另外，也有的应用属于上面 3 种类型的混合类型。

1.2.4　并行设计的方法

简单的问题可以采用分块的思维，如果是复杂的任务，就不可能容易地找出分块的方案，所以需要并行设计的方法来指导我们。

可以并行解决的问题具有以下特征：①具有内在并行性；②任务或数据可以分割。

并行性的含义包括：①同时性，两个或两个以上事件在同一时刻发生；②并发性，两个或两个以上事件在同一时间间隔发生；③流水线，两个或两个以上事件在可能重叠的时间段内。

并行程序必须以某种形式给出任务的并行性，它们可以以下面的方式给出：

1．数据并行

考虑下面的矩阵加法：A、B、C 都是 $n \times n$ 的矩阵，计算 $C=A+B$。

为了计算 C 的每个元素，需要进行如下计算：$C_{i,j}=A_{i,j}+B_{i,j}$。

注意：计算 C 的每个元素的操作都是一次加法，只是操作数不同而已，而且，每个元素的计算都是独立的，它们可以并行执行。

这种类型的并行性表现为：在不同的数据上进行相同的操作，称为数据并行性。表现出这种并行性的问题通常称为数据并行问题。数据并行问题的一个突出特点是，对大多数的这类问题，数据并行性的程度（可以并行进行数据并行的操作数目）随着问题规模的增加而增大。这意味着对于这类问题，可以用较多的处理器来有效地处理更大规模的问题。这是一种比较简单的并行方式，在实际应用中，有很多问题都是数据并行的。

例如，园艺工作，工人在一大块草坪上工作，可以把草坪分成两半，一个人负责一半。每个人在自己负责的草坪上，既翻地又除草。

又如，课代表收作业，有数学、语文、英语 3 门课，分别有 3 个课代表。但是，3 个课代表一起收数学、语文和英语作业，各自负责 1/3 的工作。

2．任务并行

将系统按照功能进行分解，称为任务分解。例如，可以设计多个线程分别完成输入、图形化界面、计算、输出等功能。任务分解较简单，因为功能重叠的机会较少。通常来说，很多子任务都可以并行执行。这种并行性表现为子任务的并行执行，因此被称为任务并行性。

例如，园艺工作，可以分解为翻地和除草两个任务，两个人可以按照任务分配工作，一个人翻地，一个人除草。

又如，课代表收作业，有数学、语文、英语 3 门课，分别有 3 个课代表。数学课代表收数学作业、语文课代表收语文作业、英语课代表收英语作业。

再如，Word 等文字处理软件，用户输入文字的同时，文档分页在后台发生。

3．流水并行

流水并行性是指在同一个数据流上同时执行多个程序，后续的程序处理的是前面程序处理过的数据流。流水并行性通常也被称为流水线。在流水线上，计算的并行性表现为：每个处理器上运行一个不同的程序，它们构成一个完整的处理流程，每个处理器把自己处理完的数据马上传递给逻辑上的下一个处理器。这样，实际上形成了处理器间的一个流水线，因此称为流水并行性。

汽车制造流程包括冲压汽车外形、焊接、涂装喷漆、安装外观和内置物品、装发动机、调试

等环节，不能颠倒顺序。但制造多辆汽车时，很多环节可以并行。例如，在冲压第三辆车的外形的同时，可以焊接第二辆车，同时给第一辆车喷漆。

4．混合并行

很多问题中的并行性表现为数据并行性、任务并行性和流水并行性的混合。某个问题表现出的流水并行性的数量，通常独立于问题的规模；而任务和数据并行性则相反，它们通常会随问题规模的增长而增长。通常情况下，任务并行性可以用来开发粗粒度的并行性，而数据并行性用来开发细粒度的并行性。因此，任务并行性和数据并行性的组合，可以有效地用于开发大量处理器上的并行算法。

1.2.5 应用系统的并行性

随着计算机网络、多处理机系统、分布式处理、并行计算等计算机软硬件技术的发展以及计算机应用领域的扩大，大量的应用系统都需要被设计成并行系统。

从网络、硬件平台的角度看，应用系统的并行性主要有以下几种情况：①分布在通过网络相连的不同计算机上的进程之间的并行；②在多个 CPU 的计算机上运行的多个进程或线程之间的并行；③在多核 CPU 的计算机上运行的多个进程或线程之间的并行。

从应用系统的需求看，在以下几种情况下系统是并行的：①用户需要跨地域进行业务处理的系统；②同时使用多台计算机或者一台计算机的多个 CPU 或多核 CPU 进行处理的系统；③同时供多个用户或者一个用户的多个操作者使用的系统；④在同一时间提供多项功能，或者说对多项功能的处理发生在同一时间的系统；⑤通过多个对外接口与系统外部的多个人员、设备或其他系统同时进行交互的系统。

1.2.6 并行计算的研究内容

并行计算的研究内容广泛，包括并行计算机系统结构、并行算法设计、并行编程环境等，主要表现在以下几方面：

（1）并行计算机的设计：包括并行计算机的结构设计、互联拓扑、网络通信等。设计并行计算机重要的一点是要考虑处理器数目的按比例增长（即可扩展性），以及支持快速通信及处理器间的数据共享等。

（2）有效算法的设计：如果没有有效的并行算法，并行计算机无法使用。而并行算法的设计完全不同于串行算法的设计；不同的并行计算机的算法设计不同，只有将不同的并行计算机与不同的实际问题相结合，才能设计出有效的并行算法。这方面的主要研究内容包括并行计算模型、并行算法的一般设计方法、基本设计技术和一般设计过程，以及常用数值并行算法与非数值并行算法的设计。

（3）评价并行算法的方法：对于给定的并行计算机及运行在上面的并行算法，需要评价运行性能。性能分析需解决的问题是，如何利用基于并行计算机及其相适应的并行算法去快速地解决问题，以及如何有效地利用各个处理器。这方面的主要研究内容包括结合计算机与算法，提出相应的性能评测指标，为设计高效的并行算法提供依据。

（4）并行计算机语言：与传统的计算机语言不同，并行语言依赖于并行计算机。并行计算机语言必须简洁，编程容易，可以有效地实现。目前，相关的技术有 MPI、OpenMP 等，而新的技

术正在不断出现。

（5）并行程序的可移植性：可移植性是指在一台并行机上开发的程序，不加修改或进行少量修改，即可在另一台计算机上运行。可移植性是并行程序设计的主要问题之一。

（6）并行计算机的自动编程：通过并行化编译器编译，用户的串行程序可以直接在并行计算机上并行执行。目前这种编译器还不存在，而仅有一些半自动并行化编译器，需要进一步进行并行化编译器设计方面的研究。

（7）并行编程环境与工具：一个综合的编程环境与工具可以使编程容易；同时，编程环境与工具可以使并行计算机的底层机构对用户透明，也可以为用户提供设计与开发程序所需要的调试器与模拟器等工具。

1.3　并行程序设计策略和模型

1.3.1　并行程序设计策略

目前，比较流行的并行程序设计策略有扩展编译器、扩展串行编程语言和创造一个并行语言 3 种。

（1）扩展编译器是开发并行化编译器，使其能够发现和表达现有串行语言程序中的并行性，例如 Intel C++ Compiler 就有自动并行循环和向量化数据操作的功能。这种把并行化的工作留给编译器的方法虽然降低了编写并行程序的成本，但因为循环和分支等控制语句的复杂组合，编译器不能识别相当多的可并行代码，而错误地编译成了串行版本。

（2）扩展串行编程语言是当前最为流行的方式，通过增加函数调用或者编译指令来表示低层语言以获取并行程序。用户能够创建和结束并行进程或线程，并提供同步与通信的功能函数等。这方面较好的库有 MPI 和 OpenMP 等；解释型脚本 Parallel Python 也有许多人在使用。

（3）创造一个并行语言是一个疯狂的想法，但近几十年来一直有人在做这样的事情，如 HPF（High Performance Fortran）是 Fortran 90 的扩展，用多种方式支持数据并行程序。

基于上述策略，并行程序设计方法主要有显式线程、利用编译器指导、利用并行应用库、并行程序语言和消息传递等方法。显式线程方法包括微软 Windows 线程 API、Pthreads、Java 线程类等；利用编译器指导的方法包括自动并行、OpenMP、Intel Threading Building Blocks 等；利用并行应用库方法包括 Intel IPP/MKL、ScaLAPACK、PARDISO、PLAPACK 等；并行程序语言有很多种；消息传递的方法包括 MPI、PVM 等。

在 Intel 提供的各种工具的支持下，并行程序设计流程如下：运用 VTune Performance Analyzer 进行分析，运用 Intel Performance libraries（IPP，MKL）、OpenMP（Intel Compiler）、Explicit threading（Win32，Pthreads）进行设计，运用 Intel Thread Checker、Intel Debugger 调试错误，运用 Intel Thread Profiler、VTune Performance Analyzer 进行性能分析和调整。

1.3.2　并行程序设计模型

并行程序设计模型是一种程序抽象的集合，程序员利用这些模型就可以为多处理机、多计算机和集群设计并行程序。并行程序设计模型逐渐形成消息传递、共享存储和数据并行 3 种标准模型，它们的特点如表 1-1 所示。