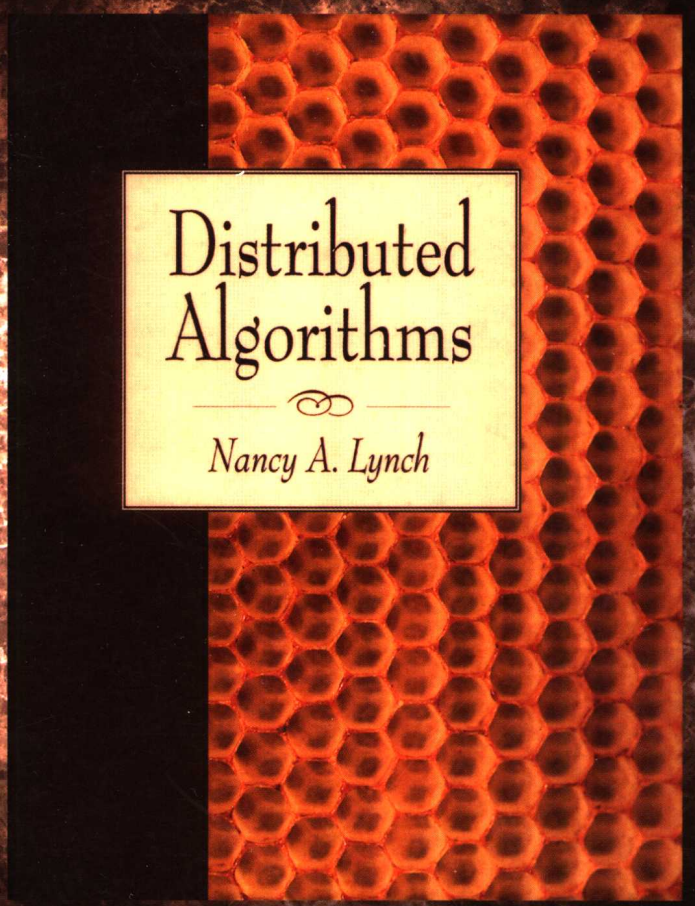
计 算 机 科 学 丛 书

分布式算法

(美) Nancy A. Lynch 著　舒继武 李国东 余华山 译



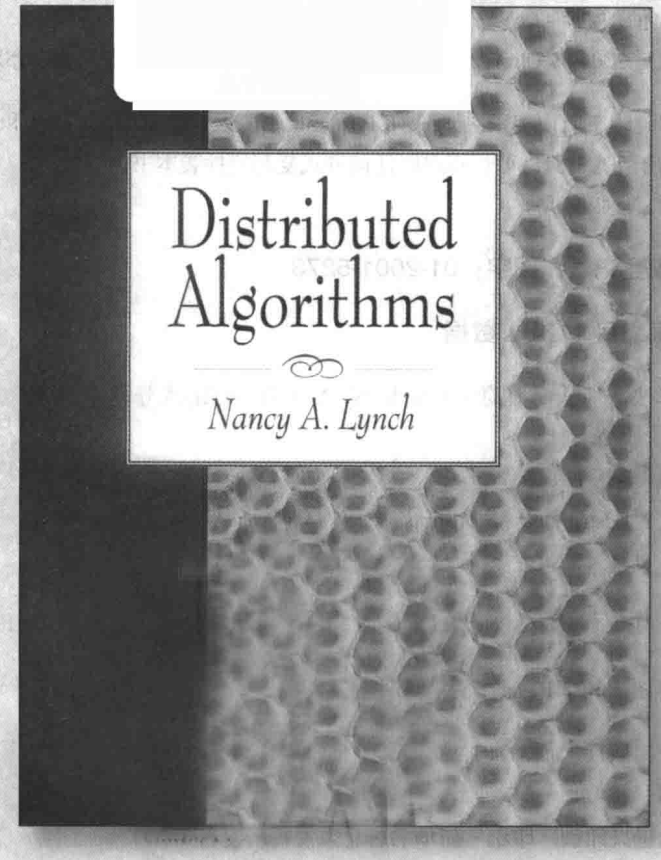
Distributed Algorithms

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

计 算 机 科 学 丛 书

分布式算法

（美） Nancy A. Lynch 著　　舒继武　李国东　余华山　译



Distributed Algorithms

机械工业出版社
China Machine Press
中信出版社
CITIC PUBLISHING HOUSE

本书对分布式算法进行全面介绍，包括最为重要的算法和不可能性结果。绝大部分的解都给出了数学证明。这些算法都根据精确定义的复杂度衡量方法进行分析。本书还讲述针对许多典型问题的算法、各类系统模型及其能力。章后提供大量习题并列出了详细的参考文献。

本书可作为高等院校计算机系研究生的教材，尤其适合对计算机理论或体系结构感兴趣的学生学习，还适合分布式设计人员、研究人员及其相关技术人员参考。

本书版权登记号：图字：01-2001-5273

图书在版编目（CIP）数据

分布式算法 /（美）林奇（Lynch, N. A.）著；舒继武等译. - 北京：机械工业出版社，2004.1

（计算机科学丛书）

书名原文：Distributed Algorithms

ISBN 7-111-13127-4

Ⅰ. 分… Ⅱ. ① 林… ② 舒… Ⅲ. 电子计算机 - 算法理论 Ⅳ. TP301.6

中国版本图书馆CIP数据核字（2003）第087861号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：姚 蕾

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2004年 1月第1版第1次印刷

787mm × 1092mm 1/16 · 34印张

印数：0 001- 4 000册

定价：59.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线电话：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

译 者 序

分布式计算是随着计算机网络的发展而兴起的，现已成为提高问题求解规模和速度、提高系统可靠性的重要手段，在数值模拟和生物工程等应用领域中被广泛应用。随着网络技术的发展以及网络计算的兴起，分布式计算技术也在不断地发展和完善中，在计算机技术的发展和应用中发挥着越来越重要的作用。在我国科学工程计算部门和高等院校中，有越来越多的科技工作者开始学习和研究分布式计算技术。

分布式计算包括三个层面的内容：作为底层的分布式系统，作为理论指导的分布式算法，以及结合具体问题的程序实现。其中，分布式算法处于重要地位，它是分布式系统的体现，更是分布式程序设计的基础和灵魂。分布式算法的一个重要特点是，它并不仅是抽象的理论研究，而且是与具体的分布式系统和应用问题密切相关的。

然而，在分布式算法的研究中，国内的有关资料十分缺乏，因此我们翻译了本书，它有几个显著特点：

全面：本书分三部分，分别对同步算法、异步算法和部分同步算法进行全面的介绍，可以作为一本分布式算法的完全手册。

严谨：书中的算法和概念都给出准确的定义，性能的分析评价都给出严格的证明，可以作为进一步深入理论研究的基础。

深入浅出：虽然算法理论有很强的抽象性，但是本书能够用浅显的语言和大量的图示作出详尽的讲解，阅读本书只需要读者具备一些基本的离散数学和概率知识。因此，本书可以适合不同层次的读者。

本书的翻译是专门从事分布式算法的科研工作者通力合作的结果，其中清华大学计算机系舒继武副教授翻译了序言和第1章至第7章，南京大学计算机系李国东副教授翻译了第8章至第13章和第15章至第20章，北京大学计算机系余华山博士翻译了第14章和第21章至第25章，全书由舒继武和李国东统稿和审校。

此书的翻译过程中，我们深切体会到本书作者在分布式算法方面的造诣，自身也获得了提高。希望本书能使国内的学者共享本书作者的思想和结晶。

由于分布式算法是一个蓬勃发展的领域，译者水平有限，加上时间仓促，书中的错误在所难免，竭诚欢迎广大读者批评指正。

译 者

2003年9月

前　言

分布式算法是用于解决多个互连处理器运行问题的算法。分布式算法的各部分并发和独立地运行，每一部分只承载有限的信息。即使处理器和通信信道以不同的速度运作，或即使某些构件出了故障，这些算法仍然应该工作正常。

分布式算法有广泛的应用：电信、分布式信息处理、科学计算以及实时进程控制。例如，今天的电话系统、航班订票系统、银行系统、全球信息系统、天气预报系统以及飞机和核电站控制系统都严重依赖于分布式算法。很明显，确保分布式算法准确、高效地运行是非常重要的。然而，由于这种算法的执行环境很复杂，所以设计分布式算法就成为了一项极端困难的任务。

本书对分布式算法这个领域做了全面的介绍——包括最为重要的算法和不可能性结果，且都是在一种简单的自动机理论环境中呈现。几乎所有的解都给出了数学证明（至少是粗略的）。这些算法都根据精确定义的复杂度衡量方法进行了分析。总之，这些材料为更深入地理解分布式算法打下了牢固的基础。

本书面向不同层次的读者。首先，本书可以作为计算机系一年级研究生的教材，尤其适合于对计算机系统、理论或两者怀有浓厚兴趣的学生。第二，本书可作为分布式系统设计人员的短期培训教材。最后，它也可作为参考手册，供设计人员、学生、研究人员以及任何对该领域感兴趣的个人使用。

本书包含了针对很多典型问题的算法，如在几种不同系统环境下的一致性(consensus)、通信、资源分配和同步问题。这些算法和结论基于分布式环境的基本假设来组织。组织的第一层基于时序模型（timing model）——同步、异步或部分同步；第二层基于进程间的通信机制——共享存储器或消息传递。每种系统模型都用数章来阐述：每一组的头一章提出所述系统类型的形式化模型，余下各章介绍了算法和不可能性结果。从头至尾，我们进行了严密的论述，然而非常简明易懂。

由于该领域很广阔也很活跃，因此本书不想去包罗万象。我们只是将最根本的结果选进了本书。若以复杂度来衡量，这些结果并不总是最优的；但它们比较简单且能够阐明重要的通用设计和推理方法。

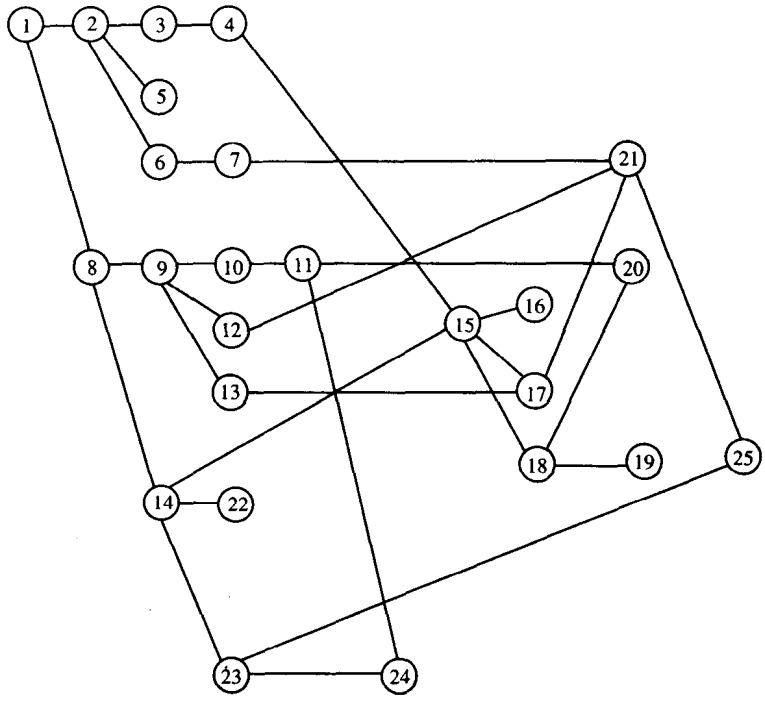
本书将会介绍分布式计算领域中许多最重要的问题、算法和不可能性结果。当实际系统中出现这些问题的时候，你就能将它们识别出来，并进而利用本书介绍的算法来解决它们，或者应用不可能性结果来证明它们是不可解的。本书还介绍各类系统模型及其能力。这样一来，你自己就可以设计出新算法（甚至还可以证明出新的不可能性结果）。最后，本书还会让你相信，严格推导分布式算法和系统是可行的：形式化建模，给出其所需行为的精确规格说明，严格证明它们符合规格说明，确定合适的复杂度衡量标准以及按照这些标准进行分析。

使用本书

预备知识　阅读本书所需的预备知识是基本的本科离散数学（包括数学归纳和渐进分析）、一些编程技能以及对计算机系统相当熟悉。有关随机算法的部分还需要基本的概率知识。有关串行算法及其分析的本科课程对阅读本书有帮助，但并不是必需的。

章节关系　本书的编排原则是使读者能比较独立地阅读不同模型的各章。各章之间的依赖

关系如图A所示。例如，如果想尽快了解异步网络，就可以跳过第5~7章。还可以只读算法部分，而不必先阅读算法所依赖的建模部分。



图A 各章之间的依赖关系

带星号的小节 在本书目录中，有几个小节的标题打了星号。它们的内容不太基本或者说比其他部分更深。第一次阅读的时候可以忽略这些内容，不会有什么影响。

课程 本书的第1版已经在MIT（麻省理工学院）的研究生导论课中用了很多年，并且在一些计算机软件和应用公司的系统设计师夏季课程中用了三年。本书包括足够一年课程的内容，所以对一些短期课程来说必须有所取舍（注意看章节之间的关系）。

例如，在强调异步网络计算的一个学期的课程中，可以选择第3、4、6章、7.2节、第12章、和第14~21章，参考一些有关建模的章节（第2、8和9章），并根据需要加入第10、11和13章中的一些定义。在强调对分布一致性进行详细研习的一个学期的课程中，可以选择第2~9、12章、13.1节、第15、17、21、23和25章。还有其他多种可能组合。如果你是这个领域的研究者，你可以用所在领域的研究报告中更新或者更特别的结论来补充本书。

在为系统设计师提供的一两周的短期课程中，可以涉及全部章节的重点，在较高的层次上讨论关键结论和关键证明思想，而无需讲解太多细节。

错误 如果在本书中发现了错误，以及对本书有什么建设性建议，请告诉我。特别欢迎对额外问题的建议。请发送email到：distalgs@theory.lcs.mit.edu。

致谢

我们很难一一列举出所有对本书的出版做出贡献的人们，因为本书是多年教学和研究的成果，得到了许多学生和研究人员的帮助。即使这样，我还是想尽力而为。

本书是MIT的研究生课程6.852（分布式算法）讲稿的最终版本。在我早期组织材料的过程中，学生们学过这门课。这些学生在1990和1992年给予了特别的帮助，当时是他们帮助我完成了讲稿的在线版本。有几位课程助教对我整理笔记给予了极大的帮助，他们是：Ken Goldman、Isaac Saias和Boaz Patt-Shamir。助教Jennifer Welch 和Rainer Gawlick也帮了我很大的忙。

许多同事和学生与我一起研究本书中的一些结果，或者与我一起讨论其他人的工作，这对我充分理解资料帮助很大。其中包括：Yehuda Afek、Eshrat Arjomandi、Hagit Attiya、Baruch Awerbuch、Bard Bloom、Alan Borodin、James Burns、Soma Chaudhuri、Brian Coan、Harish Devarajan、Danny Dolev、Cynthia Dwork、Alan Fekete、Michael Fischer、Greg Frederickson、Eli Gafni、Rainer Gawlick、Ken Goldman、Art Harvey、Maurice Herlihy、Paul Jackson、Jon Kleinberg、Leslie Lamport、Butler Lampson、Victor Luchangco、Yishay Mansour、Michael Merritt、Michael Paterson、Boaz Patt-Shamir、Gary Peterson、Shlomit Pinter、Stephen Ponzio、Isaac Saias、Russel Schaffer、Roberto Segala、Nir Shavit、Liuba Shrira、Jørgen Sørgaard-Andersen、Eugene Stark、Larry Stockmeyer、Mark Tuttle、Frits Vaandrager、George Varghese、Bill Weihl、Jennifer Welch和Lenore Zuck。尤其感谢其中的两位：我的导师Michael 和我的学生Mark Tuttle。从1978年以来，Michael就开始与我一起致力于研究这个当时还很小但看起来很有前途的领域，而Mark Tuttle的硕士论文定义并发展了I/O自动机模型。

我还要感谢Ajoy Datta、Roberto De Prisco、Alan Fekete、Faith Fich、Rainer Gawlick、Shai Halevi、Jon Kleinberg、Richard Ladner、John Leo、Victor Luchangco、Michael Melliar-Smith、Michael Merritt、Daniele Micciancio、Boaz Patt-Shamir、Anya Pogosyants、Stephen Ponzio、Sergio Rajsbaum、Roberto Segala、Nir Shavit、Mark Smith、Larry Stockmeyer、Mark Tuttle、George Varghese、Jennifer Welch和Lenore Zuck，他们审阅了全书的各部分草稿并提出了很多有用的建议。特别是Ajoy、Faith和George，他们使用本书的早期版本作为教材来教学，给出了很多宝贵的意见。此外，我要感谢 Joanne Talbot 不厌其烦的排版、画图、搜集参考文献，以及不停地打印文稿等。David Jones也参予了排版工作。在此，我还要感谢 John Guttag、Paul Penfield 和其他MIT EECS的成员，他们为我安排了写书的时间。Morgan Kaufmann的Bruce Spatz又一次鼓励并帮助我做这个艰巨的工作，他总能给我正确的建议。在本书最后成型阶段，Morgan Kaufmann的Julie Pabst和Diane Cerra给了我很大帮助。同时，也感谢Babel Press的Ed Sznyter的LaTeX技术。

最后也是最重要的，我要感谢体贴我的家人Dennis、Patrick和Mary Lynch，他们体谅我为本书所做的一切工作。同时为我料理了其他的事情。特别感谢Dennis，当我把大部分时间花在电脑前时，Dennis却在为我准备美味的海鲜晚餐，甚至把我的浴室和洗衣房翻修一新！

Nancy A. Lynch
Cambridge, Massachusetts

目　录

第二部分 异步算法

第二部分A 异步共享存储器算法

第二部分B 异步网络算法

第1章 引 言

1.1 相关主题

分布式算法（*distributed algorithm*）的概念包括大量并发算法，这些算法有着广泛的应用。最初，分布式算法这个术语用来指在那些分布在一个大的地理区域中的多个处理器上运行的算法。但多年之后，该术语的应用更为广泛。现在的分布式算法不仅包括运行在局域网上的算法，甚至还包括针对共享存储器多处理器的算法。造成这种状况的原因是：人们已经逐渐认识到，用在上述不同环境下的算法有许多共同之处。

分布式算法得到了广泛的应用：电信、分布式信息处理、科学计算以及实时进程控制。当我们为某项应用搭建一个系统时，其中一个重要的环节是设计、实现和分析分布式算法。这些算法和它们要解决的问题构成了本书中涉及到的研究领域的相关主题。

分布式算法有许多种类型。它们的分类所依据的属性包括：

- 进程间通信(IPC)的方法：分布式算法运行在一组处理器上，而这些处理器需要某种方式的通信。一些常规的通信方法包括访问共享存储器、发送点对点或广播的消息（在广域网或局域网上)以及执行远程过程调用。
- 时序模型：关于系统中事件的时序可作几种不同的假设，这反映了算法可能用到的不同时序信息类型。一种极端情况是处理器完全同步，通信和计算在完美的锁一步同步中进行。另一种极端情况是它们完全异步，以任意的速度和次序运行。两者之间有大量可能的假设，这些假设可以归为部分同步的，在这些情况下，处理器具有关于事件时序的部分信息。例如，处理器的相对速度可能有界限，或者处理器可以访问近似同步的时钟。
- 故障模型：算法运行时的底层硬件可能被假设为是完全可靠的。或者，算法可能需要容忍一定限度的故障行为。这些故障行为可能包括处理器故障，即处理器可能在给出或不给出警告的情况下停止工作；也可能暂时发生错误；或表现出严重的*Byzantine*故障，即一个出错的处理器可以作出任意动作。出错行为也包括通信机制的故障，包括消息丢失或消息重复。
- 需解决的问题：当然，算法也视它们试图解决的问题而相异。我们考虑的典型问题就是在上面提到的应用领域内产生的问题。这些问题包括资源分配、通信、分布式处理器之间的一致性、数据库并发控制、死锁检测、全局快照、同步以及各种对象类型的实现等。

本书不讲述某些并发算法，如并行随机存取机(PRAM)算法和针对固定连接网络(如数组、树和超立方体)的算法。与大部分并发算法不同，本书提到的算法具有更高程度的不确定性（*uncertainty*）和行为独立性（*independence of activity*）。本书中的算法必须面对的几种不确定性和行为独立性包括：

- 处理器数目未知
- 网络拓扑结构未知
- 不同位置上的独立输入
- 几个程序立即运行，在不同时间开始，以不同速度运行
- 处理器的不确定性

- 不确定的消息传递次数
- 不确定的消息顺序
- 处理器和通信故障

幸运的是，并不是每个算法都要面对所有这些不确定性！

由于这些不确定性，分布式算法的行为经常很难理解。即使算法的代码很短，由于多个处理器并行执行代码，其执行步骤以某种不确定的方式相互交错，也会意味着即使在输入相同的情况下，算法有多种不同的表现。因此，通过准确预测算法到底如何执行来理解算法通常是不可能的。这可以对比其他并行算法，例如，在PRAM算法中，我们通常能够预测某一时刻算法将要做什么。对于分布式算法，我们最好是理解它的行为中某些已选定的属性，而不是去理解它的行为的一切。

在过去的15年里，分布式算法的研究已经发展成一个相当一致的领域。该领域的研究风格大致如下：首先，确认在实际分布式计算中的重要问题，并定义适合对问题进行数学研究的抽象版本。然后，开发出解决问题的算法。精确地描述这些算法，证明它们能解决出现的问题，并且根据不同度量标准来分析其复杂度。算法的设计者通常会试图降低算法的复杂度。同时，要证明不可能性结果和下限，给出问题如何才能可解的限制以及其求解代价。所有这些工作的基础是分布式系统的数学模型。

这些结论构成了十分有趣的数学理论。但是它们不仅仅是数学理论：这里讲到的问题声明可用于对实际系统的某些部分建立形式化规格说明；这里讲到的算法（很多情况下）可用于实际的设计；这里给出的不可能性结果会告诉设计者应在何时停止做一些事情。所有这些结果，加上底层的数学模型，有助于设计者理解他们构建的系统。

3

1.2 我们的观点

本书研究的领域是分布式算法。因为这是一个非常广阔和活跃的领域，我们不能作全面的研究。必须有所取舍，所以我们试图选出本领域在理论上和实际上最基本的一些结果。就复杂度来讲，它们不一定是最优的结果，但我们更倾向于选出那些简单且体现了重要的设计和推理方法的结论。我们给出的结果包括一部分在本领域内相当典型的问题，如领导者选举、网络搜索、构造生成树、分布一致性、互斥、资源分配、构造对象、同步、全局快照以及可靠通信。这些问题在不同应用中重复出现。我们会在几个不同的系统模型中考虑这些问题。

本书的一大特点是，我们根据一个几乎统一的形式化框架来给出所有的算法、不可能性结果和下限。这个框架包括少量针对不同类型分布式系统的形式化自动机理论模型，以及使用这些模型来对系统进行推理的标准方法。我们的框架基于自动机理论，而不是基于某种特定的形式语言或形式证明逻辑；这使得我们可以用基本的集合论数学来表达结论，而不用过多地担心语言细节。这样也有灵活性，因为可以在同一框架内运用多种语言和逻辑来对算法进行描述和推理。使用形式化框架可以对所有的结论进行严密的处理。

对于严密性还需要多说几句。在分布式算法领域内，严密的处理十分重要，因为其中有许多微妙的因素。如果不注意这一点，就很难避免错误。然而，很难说清怎样作出完全严密的表达既足够简练又易于直观理解。在本书中，我们综合使用了直观的和严密的推理。也就是说，我们给出相应形式化模型的精确描述。对于算法，我们有时用形式化模型精确描述，有时用文字描述，有时两者都用。在讨论算法正确性时，严密的程度可能变化很大：有时给出很形式化的证明，有时仅仅是直观的概述。然而，我们希望提供足够的工具，以便在需要时可以将直观概述扩展为形式化证明。我们一般根据形式化模型来给出严密的不可能性证明。