国外计算机科学教材系列

分布式算法导论

（第二版）

Introduction to Distributed Algorithms

Second Edition

英文版

［荷］Gerard Tel　著

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

国外计算机科学教材系列

分布式算法导论

（第二版）

（英文版）

Introduction to Distributed Algorithms

Second Edition

[荷] Gerard Tel 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

分布式算法20多年来一直是倍受关注的主流方向。本书第二版不仅给出了算法的最新进展，还深入探讨了与之相关的理论知识。这本教材适合本科高年级和研究生使用，同时，本书所覆盖的广度和深度也十分适合从事实际工作的工程师和研究人员参考。书中重点讨论了点对点消息传递模型上的算法，也包括计算机通信网络的实现算法。其他重点讨论的内容包括分布式应用的控制算法（如波算法、广播算法、选举算法、终止检测算法、匿名网络的随机算法、快照算法、死锁检测算法、同步系统算法等），还涉及了利用分布式算法实现容错计算。第二版新增的关于方向感和故障检测器的内容都代表了当今最新技术发展水平，为在这些方向上从事研究的人员提供了很好的帮助。

Originally published by Cambridge University Press in 2000.
This reprint edition is published with the permission of the Syndicate of the Press of the University of Cambridge, Cambridge, England.
本书原版由 Cambridge University Press 2000年出版。
本书英文影印版权得到英国剑桥大学出版社授权。
© Cambridge University Press 2000.
This edition is licensed for distribution and sale in the People's Republic of China only excluding Hong Kong, Taiwan and Macau and may not be distributed and sold elsewhere.
本书英文影印版权（仅限中国大陆）由 Cambridge University Press 授予电子工业出版社。其原文版权受法律保护。未经许可，不得以任何形式或手段复制或抄袭本书内容。
本书英文影印版仅限于在中华人民共和国境内（不包括香港、澳门特别行政区以及台湾地区）发行与销售，并不得在其他地区发行与销售。

版权贸易合同登记号：图字：01-2003-2426

图书在版编目（CIP）数据

分布式算法导论 = Introduction to Distributed Algorithms, Second Edition：第二版 /（荷）特尔（Tel, G.）著. -北京：电子工业出版社，2003.7
（国外计算机科学教材系列）
ISBN 7-5053-8810-X

Ⅰ.分… Ⅱ.特… Ⅲ.电子计算机－算法理论－教材－英文 Ⅳ.TP301.6

中国版本图书馆CIP数据核字（2003）第046941号

责任编辑：马 岚
印 刷 者：北京兴华印刷厂
出版发行：电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路173信箱 邮编：100036
经 销：各地新华书店
开 本：787 × 980 1/16 印张：38.25 字数：857千字
版 次：2003年7月第1版 2003年7月第1次印刷
定 价：59.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔（Harvey M. Deitel)、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

主　任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委　员	王　珊	中国人民大学信息学院院长、教授
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授 中国计算机学会多媒体专业委员会主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹　鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础课程教学指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

此为试读，需要完整PDF请访问：www.ertongbook.com

Preface

Distributed systems and distributed information processing have received considerable attention in the past few years, and almost every university offers at least one course on the design of distributed algorithms. There exist a large number of books about principles of distributed systems; see for example Tanenbaum [Tan96] or Sloman and Kramer [SK87], but these concentrate on architectural aspects rather than on algorithms. Since the first edition of this book, other texts on distributed algorithms have been published by Barbosa [Bar96], Lynch [Lyn96], and Attiya and Welch [AW98].

It has been remarked that algorithms are the backbone of every computer application; therefore a text devoted solely to distributed algorithms seems to be justified. The aim of this book is to present a large body of theory about distributed algorithms, which has been developed over the past twenty years or so. This book can be used as a textbook for a one- or two-semester course on distributed algorithms; the teacher of a one-semester course may select topics to his own liking.

The book will also provide useful background and reference information for professional engineers and researchers working with distributed systems.

Exercises. Each chapter (with the exception of Chapters 1 and 13) ends with a list of exercises and small projects. The projects usually require the reader to develop a small but non-trivial extension or application of the material treated in the chapter, and in most cases I do not have a "solution". If the reader succeeds in working out one of these small projects, I would be pleased to have a copy of the result.

A list of answers (sometimes partial) to most of the exercises is available for teachers; it can be obtained from the author or by anonymous ftp.

Corrections and suggestions. If the reader finds errors or omissions in this book, please inform the author (preferably by electronic mail). All constructive criticism, including suggestions for more exercises, is most welcome.

Acknowledgements. Draft versions of this book were proofread carefully by the following: Erwin Bakker, Hans Bodlaender, Stefan Dobrev, Petra van Haaften, Ted Herman, Jan van Leeuwen, Patrick Lentfert, Friedemann Mattern, Pascale van der Put, Peter Ružička, Martin Rudalics, Anneke Schoone, and Kaisa Sere. Their comments were very helpful in improving the quality of the manuscript. Also, some students of the fall courses on "Gedistribueerde Algoritmen" at Utrecht University provided me with helpful suggestions. The Department of Computer Science provided the technical support necessary for text processing and printing. Linguistic editing was performed by Susan Parkinson.

Gerard Tel, April 1994/February 2000.

目 录 概 览

第一部分　协议（Protocols）

第二部分　基础算法（Fundamental Algorithms）

第三部分　容错（Fault Tolerance）

第四部分　附录（Appendices）

Contents

此为试读，需要完整PDF请访问： www.ertongbook.com

Chapter 1

Introduction: Distributed Systems

This chapter gives reasons for the study of distributed algorithms by briefly introducing the types of hardware and software systems for which distributed algorithms have been developed. By a distributed system we mean all computer applications where several computers or processors cooperate in some way. This definition includes wide-area computer communication networks, but also local-area networks, multiprocessor computers in which each processor has its own control unit, and systems of cooperating processes.

The different types of distributed system and the reasons why distributed systems are used are discussed in Section 1.1. Some examples of existing systems will be given. The main topic of this book, however, is not what these systems look like, or how they are used, but how they can be made to work. And even that topic will be further specialized towards the treatment of the *algorithms* used in the systems.

Of course, the entire structure and operation of a distributed system is not fully understood by a study of its algorithms alone. To understand such a system fully one must also study the complete architecture of its hardware and software, that is, the partition of the entire functionality into modules. Also, there are many important questions related to properties of the programming languages used to build the software of distributed systems. These subjects will be discussed in Section 1.2.

It is the case, however, that there are already in existence excellent books about distributed systems, which concentrate on the architectural and language aspects; see, e.g., Tanenbaum [Tan96], Sloman and Kramer [SK87], Bal [Bal90], Coulouris and Dollimore [CD88] or Goscinski [Gos91]. As already mentioned, the present text concentrates on algorithms for distributed systems. Section 1.3 explains why the design of distributed algorithms differs from the design of centralized algorithms, sketches the research field of distributed algorithms, and outlines the remainder of the book.

1.1 What is a Distributed System?

In this chapter we shall use the term "distributed system" to mean an interconnected collection of autonomous computers, processes, or processors. The computers, processes, or processors are referred to as the *nodes* of the distributed system. (In the subsequent chapters we shall use a more technical notion, see Definition 2.6.) To be qualified as "autonomous", the nodes must at least be equipped with their own private control; thus, a parallel computer of the single-instruction, multiple-data (SIMD) model does not qualify as a distributed system. To be qualified as "interconnected", the nodes must be able to exchange information.

As (software) processes can play the role of nodes of a system, the definition includes software systems built as a collection of communicating processes, even when running on a single hardware installation. In most cases, however, a distributed system will at least contain several processors, interconnected by communication hardware.

More restrictive definitions of distributed systems are also found in the literature. Tanenbaum [Tan96], for example, considers a system to be distributed only if the existence of autonomous nodes is *transparent* to users of the system. A system distributed in this sense behaves like a virtual, stand-alone computer system, but the implementation of this transparency requires the development of intricate distributed control algorithms.

1.1.1 Motivation

Distributed computer systems may be preferred over sequential systems, or their use may simply be unavoidable, for various reasons, some of which are discussed below. This list is not meant to be exhaustive. The choice of a distributed system may be motivated by more than one of the arguments listed below, and some of the advantages may come as a spin-off after the choice has been made for another reason. The characteristics of a distributed system may vary also, depending on the reason for its existence, but this will be discussed in more detail in Subsections 1.1.2 through 1.1.6.

(1) *Information exchange.* The need to exchange data between different computers arose in the sixties, when most major universities and companies started to have their own mainframe computer. Cooperation between people of different organizations was facilitated by

the exchange of data between the computers of these organizations, and this gave rise to the development of so-called *wide-area networks* (WANs). ARPANET, the predecessor of the current Internet, went on-air in December, 1969. A computer installation connected in a wide-area network (sometimes called a *long-haul network*) is typically equipped with everything a user needs, such as backup storage, disks, many application programs, and printers.

Later computers became smaller and cheaper, and soon each single organization had a multitude of computers, nowadays often a computer for each person (a personal computer or workstation). In this case also the (electronic) exchange of information between personnel of one organization already required that the autonomous computers were connected. It is even not uncommon for a single person or family to have multiple computers in the home, and connect these in a small personal home-network.

(2) *Resource sharing.* Although with cheaper computers it became feasible to equip each employee of an organization with a private computer, the same is not the case for the peripherals (such as printers, backup storage, and disk units). On this smaller scale each computer may rely on dedicated servers to supply it with compilers and other application programs. Also, it is not effective to replicate all application programs and related file resources in all computers; apart from the waste of disk space, this would create unnecessary maintenance problems. So the computers may rely on dedicated nodes for printer and disk service. A network connecting computers on an organization-wide scale is referred to as a *local-area network* (LAN).

The reasons for an organization to install a network of small computers rather than a mainframe are cost reduction and extensibility. First, smaller computers have a better price–performance ratio than large computers; a typical mainframe computer may perform 50 times faster than a typical personal computer, but cost 500 times more. Second, if the capacity of a system is no longer sufficient, a network can be made to fit the organization's needs by adding more machines (file servers, printers, and workstations). If the capacity of a stand-alone system is no longer sufficient, replacement is the only option.

(3) *Increased reliability through replication.* Distributed systems have the potential to be more reliable than stand-alone systems because they have a *partial-failure* property. By this it is meant that some nodes of the system may fail, while others are still operating correctly and can

take over the tasks of the failed components. The failure of a stand-alone computer affects the entire system and there is no possibility of continuing the operation in this case. For this reason distributed architectures are a traditional concern in the design of highly reliable computer systems.

A highly reliable system typically consists of a two, three, or four times replicated uniprocessor that runs an application program and is supplemented with a voting mechanism to filter the outputs of the machines. The correct operation of a distributed system in the presence of failures of components requires rather complicated algorithmical support.

(4) *Increased performance through parallelization.* The presence of multiple processors in a distributed system opens up the possibility of decreasing the turn-around time for a computation-intensive job by splitting the job over several processors.

Parallel computers are designed specifically with this objective in mind, but users of a local-area network may also profit from parallelism by shunting tasks to other workstations.

(5) *Simplification of design through specialization.* The design of a computer system can be very complicated, especially if considerable functionality is required. The design can often be simplified by splitting the system into modules, each of which implements part of the functionality and communicates with the other modules.

On the level of a single program modularity is obtained by defining abstract data types and procedures for different tasks. A larger system may be defined as a collection of cooperating processes. In both cases, the modules may all be executed on a single computer. But it is also possible to have a local-area network with different types of computers, one equipped with dedicated hardware for number crunching, another with graphical hardware, a third with disks, etc.

1.1.2 Computer Networks

By a computer network we mean a collection of computers, connected by communication mechanisms by means of which the computers can exchange information. This exchange takes place by sending and receiving messages. Computer networks fit our definition of distributed systems. Depending on the distance between the computers and their ownership, computer networks are called either *wide-area networks* or *local-area networks.*

此为试读,需要完整PDF请访问: www.ertongbook.com