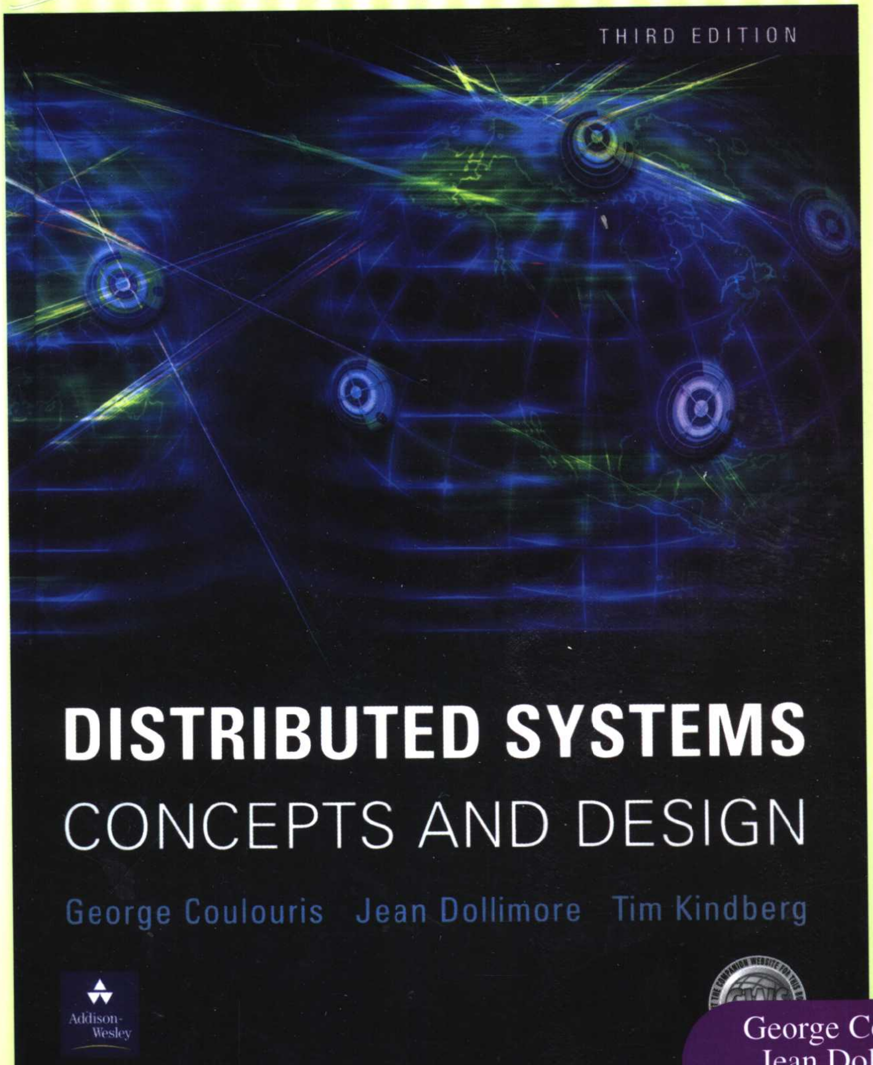
经典原版书库

分布式系统概念与设计

（英文版·第3版）



THIRD EDITION

DISTRIBUTED SYSTEMS

CONCEPTS AND DESIGN

George Coulouris Jean Dollimore Tim Kindberg

Addison-Wesley

George Coulouris
Jean Dollimore 著
Tim Kindberg

机械工业出版社
China Machine Press

经 典 原 版 书 库

分布式系统概念与设计

（英文版 · 第3版）

Distributed Systems

Concepts and Design

(Third Edition)

George Coulouris
Jean Dollimore 著
Tim Kindberg

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2003-1009

图书在版编目（CIP）数据

分布式系统概念与设计（英文版・第3版）/库勒瑞思（Coulouris, G.）等著. –北京：机械工业出版社，2003.3

（经典原版书库）

书名原文：Distributed Systems: Concepts and Design，Third Edition

ISBN 7-111-11749-2

Ⅰ.分… Ⅱ.库… Ⅲ.分布式系统－英文 Ⅳ.TP316.4

中国版本图书馆CIP数据核字（2003）第013558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：华 章
北京瑞德印刷有限公司印刷・新华书店北京发行所发行
2003年3月第1版第1次印刷
787mm×1092mm 1/16・49.75印张
印数：0 001-3 000册
定价：75.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

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）

PREFACE

This third edition of our textbook arrives at a time when distributed systems, particularly the Web and other Internet-based applications and services, are of unprecedented interest and importance. The book aims to convey insight into, and knowledge of, the principles and practice underlying the design of distributed systems, both Internet-based and otherwise. Information is provided in sufficient depth to allow readers to evaluate existing systems or design new ones. Detailed case studies illustrate the concepts for each major topic.

Distributed systems techniques developed over the last two to three decades, such as interprocess communication and remote invocation, distributed naming, cryptographic security, distributed file systems, data replication and distributed transaction mechanisms, provide the run-time infrastructure supporting today's networked computer applications.

Distributed system development relies increasingly on middleware support through the use of software frameworks that provide abstractions such as distributed shared objects, and services including secure communication, authentication and access control, mobile code, transactions and persistent storage mechanisms.

In the near future, distributed applications will enable closer cooperation between users through replicated data and multimedia data streams, and will support user and device mobility using wireless and spontaneous networking.

Developers of distributed systems and applications now benefit from a range of useful languages, tools and environments. These enable students as well as professional developers to construct working distributed applications.

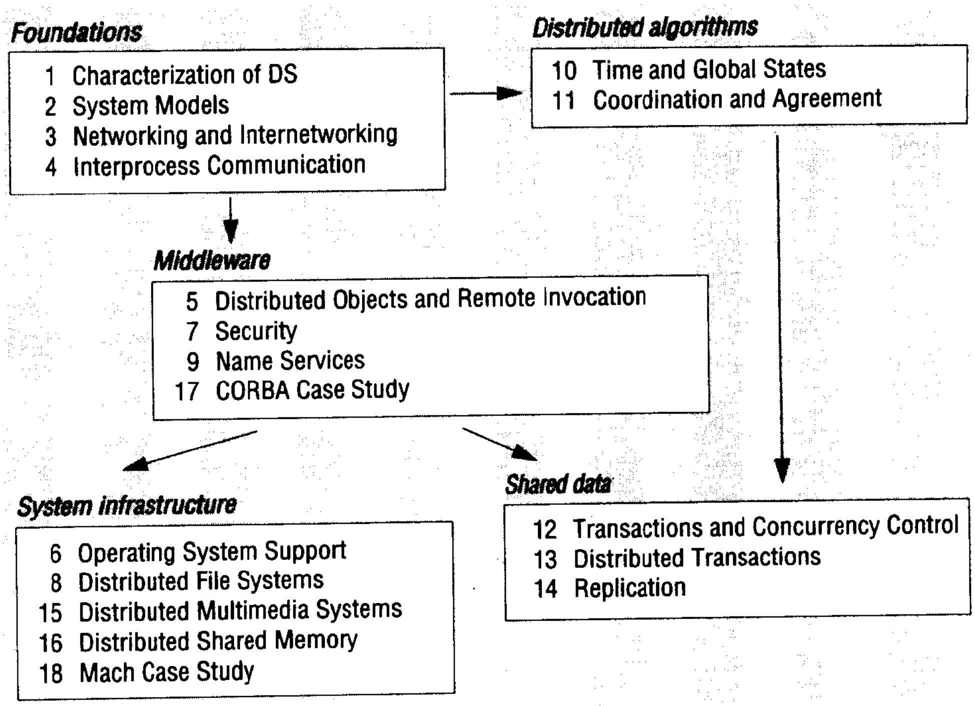
Purposes and readership

The book is intended for use in undergraduate and introductory postgraduate courses. It can equally be used for self-study. We take a top-down approach, addressing the issues to be resolved in the design of distributed systems and describing successful approaches in the form of abstract models, algorithms and detailed case studies of widely-used systems. We cover the field in sufficient depth and breadth to enable readers to go on to study most research papers in the literature on distributed systems.

We aim to make the subject accessible to students who have a basic knowledge of object oriented programming, operating systems and elementary computer architecture. The book includes coverage of those aspects of computer networks relevant to distributed systems, including the underlying technologies for the Internet, wide area, local area and wireless networks. Algorithms and interfaces are presented throughout the book in Java or, in a few cases, ANSI C. For brevity and clarity of presentation, a form of pseudo-code derived from Java/C is also used.

Organization of the book

The following diagram shows the book's chapters under five main topic areas. It is intended to provide a guide to the book's structure and to indicate recommended navigation routes for instructors wishing to provide, or readers wishing to achieve, understanding of the various subfields of distributed system design:


Foundations
1 Characterization of DS
2 System Models
3 Networking and Internetworking
4 Interprocess Communication
Distributed algorithms
10 Time and Global States
11 Coordination and Agreement
Middleware
5 Distributed Objects and Remote Invocation
7 Security
9 Name Services
17 CORBA Case Study
System infrastructure
6 Operating System Support
8 Distributed File Systems
15 Distributed Multimedia Systems
16 Distributed Shared Memory
18 Mach Case Study
Shared data
12 Transactions and Concurrency Control
13 Distributed Transactions
14 Replication

References

The existence of the World Wide Web has changed the way in which a book such as this can be linked to source material including research papers, technical specifications and standards. Many of the source documents are now available on the web; some are available only there. For reasons of brevity and readability, we employ a special form of reference to web material which loosely resembles a URL: references such as [www.omg.org] and [www.rsasecurity.com I] refer to documentation that is available only on the web. They can be looked up in the reference list at the end of the book, but the full URLs are given only in an online version of the reference list at the book's web site: www.cdk3.net/refs where they take the form of clickable links. Both versions of the reference list include a more detailed explanation of this scheme.

Rewritten and extended chapters:	1	Characterization of DS
	3	Networking and Internetworking
	4	Interprocess Communication
	5	Distributed Objects and Remote Invocation
	7	Security
	10	Time and Global States
	14	Replication
Entirely new chapters:	2	System Models
	11	Coordination and Agreement
	15	Distributed Multimedia Systems
	17	CORBA Case Study
Chapters that have been brought up-to-date:	6	Operating System Support
	8	Distributed File Systems
	9	Name Services
	12	Transactions and Concurrency Control
	13	Distributed Transactions
	16	Distributed Shared Memory
	18	Mach Case Study

Changes for this edition

This third edition appears some six years after the second edition. The work done to produce this edition is summarized in the table above. The introductory chapters and some others have been largely rewritten and several new chapters have been introduced to reflect new perspectives and technical directions. There has been much reorganization in other chapters, affecting the topics covered, the depth of coverage and the location of material. We have condensed older material to make space for new topics. Some material has been moved to a more prominent position to reflect its new significance. The case studies removed from the second edition can be found on the book's web site, as described below.

Acknowledgements

We are grateful to the many classes of students at Queen Mary and Westfield College whose contributions in seminars and lectures have helped us to produce this new edition, as have the helpful suggestions of many teachers elsewhere. Kohei Honda tested early drafts in classes at QMW and this resulted in some particularly valuable comments.

The following gave their time unstintingly to review substantial portions of the final draft, resulting in substantial improvements: Angel Alvarez, Dave Bakken, John Barton, Simon Boggis, Brent Callaghan, Keith Clarke, Kurt Jensen and Roger Prowse.

We should also like to thank the following, who either gave permission to use their material or suggested topics for inclusion in the book: Tom Berson, Antoon Bosselaers,

Ralph Herrtwitch, Frederick Hirsch, Bob Hopgood, Ajay Kshemkalyan, Roger Needham, Mikael Pettersson, Rick Schantz and David Wheeler.

We are grateful to the Department of Computer Science, Queen Mary and Westfield College, for hosting the companion web site and to Keith Clarke and Tom King for their support in setting it up.

Finally, we thank Keith Mansfield, Bridget Allen, Julie Knight and Kristin Erickson of Pearson Education/Addison-Wesley for essential support throughout the arduous process of getting the book into print.

Web site

We maintain a web site with a wide range of material designed to assist teachers and readers. It can be accessed via either of the URLs:

www.cdk3.net	www.booksites.net/cdkbook

The site includes:

Reference list: The list of references that can be found at the end of the book is replicated at the web site. The web version of the reference list includes active links for material that is available online.

Errata list: A list of errors that are discovered in the book, with corrections for them.

Supplementary material: We plan to maintain a set of supplementary material for each chapter. Initially, this consists of source code for the programs in the book and relevant reading material that was present in the previous edition of the book but was removed from this one for reasons of space. References to this supplementary material appear in the book with links such as www.cdk3.net/ipc.

Contributed teaching material: We hope to extend the supplementary material to cover new topics as they emerge during the lifetime of this edition. In order to do so, we invite teachers to submit teaching material, including lecture notes and laboratory projects. A procedure for the submission of supplementary material is described at the web site. Submissions will be reviewed by a team including several teachers who are users of the book.

Links to web sites for courses using the book: Teachers are asked to notify us of their courses with URLs for inclusion in the list.

Instructor's Guide: Comprising:

- Complete artwork of the book in a form suitable for use as slide masters.
- Solutions to the exercises (protected by a password available only to teachers).
- Chapter-by-chapter teaching hints.
- Suggested laboratory projects.

George Coulouris
Jean Dollimore
Tim Kindberg
London & Palo Alto, June 2000
<authors@cdk3.net>

CONTENTS

1

CHARACTERIZATION OF DISTRIBUTED SYSTEMS

A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. This definition leads to the following characteristics of distributed systems: concurrency of components, lack of a global clock and independent failures of components.

We give three examples of distributed systems:

- the Internet;
- an intranet, which is a portion of the Internet managed by an organization;
- mobile and ubiquitous computing.

The sharing of resources is a main motivation for constructing distributed systems. Resources may be managed by servers and accessed by clients or they may be encapsulated as objects and accessed by other client objects. The Web is discussed as an example of resource sharing and its main features are introduced.

The challenges arising from the construction of distributed systems are the heterogeneity of its components, openness, which allows components to be added or replaced, security, scalability – the ability to work well when the number of users increases – failure handling, concurrency of components and transparency.

1.1 Introduction

Networks of computers are everywhere. The Internet is one, as are the many networks of which it is composed. Mobile phone networks, corporate networks, factory networks, campus networks, home networks, in-car networks, all of these, both separately and in combination, share the essential characteristics that make them relevant subjects for study under the heading *distributed systems*. In this book we aim to explain the characteristics of networked computers that impact system designers and implementors and to present the main concepts and techniques that have been developed to help in the tasks of designing and implementing systems that are based on them.

We define a distributed system as one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages. This simple definition covers the entire range of systems in which networked computers can usefully be deployed.

Computers that are connected by a network may be spatially separated by any distance. They may be on separate continents, in the same building or the same room. Our definition of distributed systems has the following significant consequences:

Concurrency: In a network of computers, concurrent program execution is the norm. I can do my work on my computer while you do your work on yours, sharing resources such as web pages or files when necessary. The capacity of the system to handle shared resources can be increased by adding more resources (for example. computers) to the network. We will describe ways in which this extra capacity can be usefully deployed at many points in this book. The coordination of concurrently executing programs that share resources is also an important and recurring topic.

No global clock: When programs need to cooperate they coordinate their actions by exchanging messages. Close coordination often depends on a shared idea of the time at which the programs' actions occur. But it turns out that there are limits to the accuracy with which the computers in a network can synchronize their clocks – there is no single global notion of the correct time. This is a direct consequence of the fact that the *only* communication is by sending messages through a network. Examples of these timing problems and solutions to them will be described in Chapter 10.

Independent failures: All computer systems can fail and it is the responsibility of system designers to plan for the consequences of possible failures. Distributed systems can fail in new ways. Faults in the network result in the isolation of the computers that are connected to it, but that doesn't mean that they stop running. In fact the programs on them may not be able to detect whether the network has failed or has become unusually slow. Similarly, the failure of a computer, or the unexpected termination of a program somewhere in the system (a *crash*) is not immediately made known to the other components with which it communicates. Each component of the system can fail independently, leaving the others still running. The consequences of this characteristic of distributed systems will be a recurring theme throughout the book.

The motivation for constructing and using distributed systems stems from a desire to share resources. The term 'resource' is a rather abstract one, but it best characterizes the range of things that can usefully be shared in a networked computer system. It extends