国外经典教材

分布式计算原理与应用

（影印版）

(美) M. L. Liu 著

PEARSON
Addison Wesley

Distributed Computing

PRINCIPLES and APPLICATIONS

M. L. Liu

Pearson Education
清华大学出版社

国外经典教材

分布式计算原理与应用

（影印版）

（美）　M. L. Liu　著

清华大学出版社

北　京

Distributed Computing

PRINCIPLES and APPLICATIONS

M.L. Liu
California Polytechnic State University,
San Luis Obispo

PEARSON
Addison
Wesley

Boston San Francisco New York
London Toronto Sydney Tokyo Singapore Madrid
Mexico City Munich Paris Cape Town Hong Kong Montreal

内 容 简 介

这是专门为本科生和研究生设计的教材，供分布式计算课程使用，不仅详细介绍了分布式程序设计技术的核心概念和原理，还采用 learn by doing 方法，使读者能通过程序设计和实验练习来巩固所学知识。

图书在版编目(CIP)数据

分布式计算原理与应用＝Distributed Computing: Principles and Applications /（美）刘 （Liu，M. L.）著. —影印版. —北京：清华大学出版社，2004.4
（国外经典教材）
ISBN 7-302-08268-5
Ⅰ. 分… Ⅱ. 刘… Ⅲ. 分布式操作系统－程序设计－教材－英文 Ⅳ. TP316.4
中国版本图书馆 CIP 数据核字（2004）第 018434 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
文稿编辑：文开棋
封面设计：陈刘源
印 刷 者：北京牛山世兴印刷厂
装 订 者：北京国马印刷厂
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：28.25
版　　次：2004 年 4 月第 1 版　2004 年 4 月第 1 次印刷
书　　号：ISBN 7-302-08268-5/TP · 5964
印　　数：1～3000
定　　价：45.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770175-3103 或(010)62795704

Senior Acquisitions Editor: *Maite Suarez-Rivas*
Marketing Manager: *Nathan Schultz*
Marketing Coordinator: *Lesly Hershman*
Text Design, Composition and Art: *Gillian Hall, The Aardvark Group*
Cover Design: *Gina Hagen Kolenda*
Cover Image: *Dorling Kindersley, Ltd.*
Prepress and Manufacturing: *Caroline Fell/Hugh Crawford*

Access the latest information about Addison-Wesley titles from our World Wide Web site:
http://www.aw.com/cs

The programs and applications presented in this book have been included for their instructional value. They have been tested with care, but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, not does it accept any liabilities with respect to the programs or applications.

Library of Congress Cataloging-in-Publication Data

Liu, Mei-Ling L.
Distributed computing : principles & applications / Mei-Ling L. Liu.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-79644-9
1. Electronic data processing--Distributed processing. I. Title.

QA76.9.D5L58 2003
004'.36--dc21
2002034187

ISBN 0-201-79644-9
1 2 3 4 5 6 7 8 9 10-DOC-06050403

Distributed Computing

PRINCIPLES and APPLICATIONS

Preface

In the *Year 2001 Model Curricula for Computing (Computing Curricula 2001) (http://www.computer.org/education/cc2001/report/index.html)* developed by the Joint IEEE Computer Society/ACM Task Force, net-centric computing is included as a key area in the Computer Science body of knowledge:

> *Recent advances in computer and telecommunications networking, particularly those based on TCP/IP, have increased the importance of networking technologies in the computing discipline. Net-centric computing covers a range of subspecialties, including computer communication network concepts and protocols, multimedia systems, Web standards and technologies, network security, wireless and mobile computing, and distributed systems.*
>
> *Mastery of this subject area involves both theory and practice. Learning experiences that involve hands-on experimentation and analysis are strongly recommended as they reinforce student understanding of concepts and their application to real-world problems. Laboratory experiments should involve data collection and synthesis, empirical modeling, protocol analysis at the source code level, network packet monitoring, software construction, and evaluation of alternative design models. All of these are important concepts that can be best understood by laboratory experimentation.*

The ACM model curricula lists a number of topics in this area, specifying a minimum of 15 hours of core topics and additional elective topics. Many of these topics are covered in a series of courses in distributed computing that I initiated and have taught at California Polytechnic State University (Cal Poly), San Luis Obispo, since 1996. For these courses I employed excerpts from various publications, as well as materials I developed, including overhead slides, code samples, and laboratory/problem/research assignments, which I provided as a course package to my students.

This textbook is a synthesis of the course materials I accumulated over a span of six years, designed for a sequence of technical elective courses for upper-division undergraduates.

Reasons for the Book

Traditionally, distributed computing courses are offered at the graduate level. With the growth of Internet and intranet applications, more and more undergraduate students are engaged in net-centric computing, either at the workplace or on their own initiative. Distributed computing is distinct from (1) communications and networks and (2) distributed operating systems. It operates at a higher level of abstraction than the network layer and the operating systems layer and deals with programming paradigms, application program interfaces (APIs) or toolkits, and protocols and standards in the context of net-centric computing. Although there are numerous books available on network programming and technologies, there is a lack of books written in a textbook style, combining the theory and practice of distributed computing.

This book has the following distinctive features:

- It is designed to introduce **undergraduate students** to the principles of distributed computing, topics formerly reserved for graduate students.
- It focuses on the upper layers of the architecture for net-centric computing, specifically on **computing abstractions** and **paradigms**.
- It incorporates both conceptual topics and practical topics, using program samples and hands-on exercises to illustrate and reinforce the concepts presented.
- It is designed as a textbook, with a narrative style suitable for academic settings, diagrams to illustrate the topics, exercises at the end of each chapter, and a list of references for student research.
- It is designed for **learn-by-doing teaching**: programming samples are used to reinforce the topics presented, and **laboratory** activities are incorporated in the exercises at the end of each chapter.
- **Supplementary teaching materials**, including presentation slides, program samples, a Web site, and an instructor's manual, will be provided by the author.
- In addition to printed books and articles, this book cites reliable references that are accessible via the World Wide Web. For example, the references include a link to the archival sites where Internet Requests for Comments (RFCs) can be looked up online. It is the author's belief that undergraduate students are more inclined to look up references that are readily available on the Web. (*Note*: Although I have chosen to include as references Web links that are reliable and stable, it is possible that some of the links may become obsolete over time. I would appreciate receiving reports of inactive links.)

What This Book Is Not About

- **This book is not about networking.** In a general sense, networking encompasses distributed computing in as much as distributed computing involves networked computers. But in academia, courses in networking typically focus on the lower layers of the network architecture and address topics such as signal transmission, error corrections, data link layer protocols, transport layer protocols, and the Internet layer protocol. By comparison, this book deals with the uppermost layers of the network architecture, namely the application, presentation, and session layers, and more from the viewpoint of computing paradigms and abstractions than from the viewpoint of system architecture.

- **This book is not about distributed systems.** Our focus is not on system architecture or system resources.
- **This book is not about Web application development.** Although the Internet is the most popular network, programming for the Internet is a specialized form of distributed computing. This book addresses distributed computing for networks in general, including the Internet, intranet, and local area networks.
- **This book is not about application program interfaces (APIs) or technologies.** Although a number of APIs are introduced in the book, they are presented as representative toolkits that support particular paradigms; the introduction of these APIs is meant to allow students to write programs for hands-on laboratory exercises that reinforce the concepts and principles.

A Word to Fellow Instructors

The book is designed for use in a technical elective course during an academic term. The entire twelve chapters of the book can be covered in a quarter at a fast pace or in a semester at a more leisurely pace. The material requires no advanced knowledge of networks, operating systems, or software engineering. A course taught using the book can be taken by an upper-division undergraduate student.

With a field as wide as distributed computing, it is impossible for one book to cover every corner of the discipline. In particular, there is no intention here to cover the latest technologies. The book is meant to convey fundamental concepts in interprocess communication.

A common thread that runs through this book is the idea of abstraction, in the sense of detail encapsulation—how that idea applies in the various paradigms of distributed computing and the trade-offs between tools that provide different levels of abstraction. It is my firm belief that these concepts and ideas are important to each student in computer science and computer engineering, regardless of the student's chosen area of specialization. With an understanding of these fundamental concepts, students should be well equipped to explore new tools and technologies on their own, as they will be expected to do throughout their career.

The first three chapters of the book contain introductory background material that can be covered in the first one or two weeks of an academic term, during which students are gently introduced to a multitude of issues to which they may or may not have been exposed already. Subsequent chapters are more technical and detailed and can be covered at roughly a chapter each week.

Broad as the subject of distributed computing is, you will likely feel compelled to supplement this text with additional material that you deem important. For example, you may be inclined to add an introduction to distributed algorithms, or to go into more depth in the area of security. To allow time for such additions, you may consider omitting selected chapters from this book.

It is not assumed that readers of this book have had prior expertise in distributed computing. At Cal Poly, I have used the material to teach students with diverse backgrounds, ranging from some students who had no experience with multiprocess programming, to those who were already sophisticated network software developers. While the material worked best with the former, those with advanced backgrounds would find the topics of interest nevertheless.

Readers, Please Note

A word on the use of typefaces in the narrative of this book:

- Key terms and phrases are emphasized through the use of boldface; for example: This book addresses **distributed computing**.
- A special word, such as one used as a program identifier, nonstandard protocol name, or operation name, is expressed in italic to distinguish it from the rest of the text in a sentence; for example: What do you expect the outcome to be when *RunThread3* is executed? Compile and run it.
- Reserved words and identifiers, such as those stipulated by the Java Language or by a well-known protocol, appear in italic; for example: To support threading in a program, Java provides a class named *Thread* as well as an interface named *Runnable* interface.

A word on the article inserts and Web links:

Throughout the text of this book, excerpts from articles previously published in various media are inserted. These articles were chosen for their relevance to the topics, and for the interest that they may bring to the readers.

Many of the references listed at the end of each chapter are Web links. This is a deliberate choice, because the author believes that the accessibility of Web-available contents will enhance a student's initiative to pursue research in a topic. The Web links chosen are those that the author deemed to be reliable and stable. But obsolescence of some links is inevitable, in which case the author offers an apology and welcomes your notification of such an occurrence.

Contacting the Author

The compilation of a textbook is a painstaking and consuming task. To the best of my ability, I have attempted to maximize the accuracy of the materials presented within the pages of this book. If you discover any errors or inaccuracy, or if you have suggestions for improvements, I shall be grateful to hear from you. Please email to *mliu@csc.calpoly.edu.*

Supplementary Materials

Supplementary materials, including the source files for programming samples and overhead slides, are available at URL www.aw.com.

Additional supplementary materials for instructors only are available by contacting your local Addison Wesley sales representative.

Acknowledgments

I thank the generosity of the various authors and publishers who granted reprint permission to allow the inclusion of previously published work in the text of this book.

I am forever indebted to my doctoral program advisors, Dr. Divyakant Agrawal and Dr. Amr El Abbadi, Computer Science Department, University of California at Santa Barbara, who introduced me to the field of distributed computing.

Gratitude is due to Cal Poly students Jared Smith, Domingo Colon, Vinh Pham, Hafeez Jaffer, Erik Buchholz, and Lori Sawdey for the gift of their time and efforts in reviewing the book. Thanks are also due to the students who enrolled in these courses at Cal Poly in school year 2001–2003: Computer Engineering 369, Computer Engineering 469, and Computer Science 569, for their patience with the numerous errors in the initial drafts of the manuscript for this book.

I thank my colleagues in the Computer Science Department and the College of Engineering at Cal Poly, who provided the opportunities for me to teach the courses that inspired this book, and without whose encouragement the book would not have been possible.

I thank the staff of Addison Wesley for their invaluable guidance and assistance that allows the book to come to fruition, and the following reviewers for the generosity of sharing their time and wisdom:

Anup Talukdar
Motorola Labs, Schaumburg

Dr. Ray Toal
Loyola Marymount University

Mr. David Russo
Computer Science and Engineering
Senior Lecturer, University of Texas at Dallas

Dr. Alvin Lim
Computer Science and Software Engineering Department
Auburn University

Isaac Ghansah, Professor
Computer Science and Computer Engineering
California State University, Sacramento

Bruce Char
Department of Computer Science
Drexel University

Finally, I thank my family and in particular my son, Marlin, for providing a reason for me to strive.

M. L. Liu
Cal Poly, San Luis Obispo
January, 2003

Contents

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5
The Client-Server Paradigm 133

CHAPTER 6

CHAPTER 7