计　算　机　科　学　丛　书

Scalable Parallel Computing

Technology, Architecture, Programming

可扩展并行计算

技术、结构与编程

（英文版）

Kai Hwang　Zhiwei Xu 著

机械工业出版社
China Machine Press

WCB
McGraw-Hill

计算机科学丛书

可扩展并行计算

技术、结构与编程

(英文版)

Scalable Parallel Computing

Technology, Architecture, Programming

Kai Hwang Zhiwei Xu 著

机械工业出版社
China Machine Press

WCB
Mc Graw Hill

RISBN: 007-1164456

本书版权登记号：图字：01-99-1094

图书在版编目(CIP)数据

可扩展并行计算：技术、结构与编程：黄铠 (Hwang. K)，徐志伟(Xu. Z)著. －影印本. －北京：机械工业出版社，1999.5

(计算机科学丛书)

书名原文：Scalable Parallel Computing: Technology, Architecture, Programming

ISBN 7-111-07176-X

Ⅰ.可… Ⅱ.①黄… ②徐… Ⅲ. 并行处理－英文 Ⅳ.TP274

中国版本图书馆CIP数据核字(1999)第09070号

[illegible]

出 版 人：马九荣 (北京市百万庄大街22号 邮政编码100037)

北京忠信诚胶印厂印刷 · 新华书店北京发行所发行

1999年5月第1版第1次印刷

787mm × 1092mm 1/16 · 51.5印张

印数：0 001-5 000册

定价：69.00元

Trademark Notice

About the Authors

Kai Hwang presently holds a Chair Professor of Computer Engineering at the University of Hong Kong (HKU), while taking a research leave from the University of Southern California (USC). He has been engaged in higher education and computer research for 26 years, after earning the Ph.D. in Electrical Engineering and Computer Science from the University of California at Berkeley. His work on this book started at the USC and was mostly completed at the HKU.

An IEEE Fellow, he has published extensively in the areas of computer architecture, digital arithmetic, parallel processing, and distributed computing. He is the founding Editor-in-Chief of the *Journal of Parallel and Distributed Computing*. He has chaired the international conferences: ICPP86, ARITH-7, IPPS96, ICAPP 96, and HPCA-4 in 1998. He has received several achievement awards for outstanding research and academic contributions to the field of parallel computing.

He has lectured worldwide and performed consulting and advisory work for US National Academy of Sciences, MIT Lincoln Laboratory, IBM Fishkill, TriTech in Singapore, Fujitsu and ETL in Japan, GMD in Germany, CERN School of Computing, and Academia Sinica in China. Presently, he leads a research group at HKU in developing an ATM-based multicomputer cluster for high-performance computing and distributed multimedia, Intranet, and Internet applications.

Zhiwei Xu is a Professor and Chief Architect at the National Center for Intelligent Computing Systems (NCIC), Chinese Academy of Sciences, Beijing, China. He is also a Honorary Research Fellow of HKU. He received a Ph.D. in Computer Engineering from the University of Southern California. He participated in the STAP/MPP benchmark projects led by Dr. Hwang at USC and HKU during the past 5 years.

He has taught at the Rutgers University and New York Polytechnic University. He has published in the areas of parallel languages, pipelined vector processing, and benchmark evaluation of massively parallel processors. Presently, he leads a design group at NCIC in building a series of cluster-based superservers in China. His current research interest lies mainly in network-based cluster computing and the software environments for parallel programming.

Dedication

To my family, for their love and support -- k.h.

To my high-school teacher,
Ms. Zhang Dahua, who taught me the
serenity of science at a tumultuous time -- z.x.

Foreword

Michael J. Flynn, Stanford University

Parallel processors are the future of computer design. The replicability of technology has proven an ever-increasing impetus towards parallel processor implementations. The demand for increasing performance in important applications is another persuasive argument. However, building efficient, scalable parallel processors has proven to be a very difficult problem.

For decades, it seemed that large-scale ($n > 100$ processors) parallel processors could be realized efficiently with just a little bit more effort and research. This has not proven so. It has been difficult to find the prerequisite parallelism in programs, and when this parallelism was found, it did not translate into a speedup of program execution proportional to the number of processing elements involved. This has been especially true when n exceeds 100.

One important problem in the realization of efficient parallel processors is the computational model that underlies the program model and language used to represent applications. This model was developed for and efficiently suits the single processor. Because our linguistic notions of sequential specification of actions are familiar, they are readily mapped onto a serial computational and programming model. Attempts to remap these representations onto the parallel model have proved so far quite inefficient.

There are three legs that support the understanding of efficient parallel processing: *computational models, underlying alternatives,* and the *programming paradigms.* This book is uniquely concerned about all three issues. Metrics for performance provide a quantitative basis for understanding basic models of computation.

Professors Hwang and Xu comprehensively analyze hardware models of processors, interconnection networks, and serial wide area networks giving a complete picture of the hardware state of the art. Their coverage extends from the hardware ILP (instructional level parallelism) to forms of parallelism achieved by NOW (networks of workstations). Their systems viewpoint integrates the computational model, the hardware, and the programming model in a review of major parallel processor implementation efforts that are currently underway.

It is in this integration of hardware and software that this book is most valuable. It is only by understanding the mapping from application onto system and into processor configuration that we can expect the consistent progress in building efficient scalable parallel processor systems.

Preface

Oh, A Digital World!

This book covers scalable architecture and parallel programming of multiprocessors, multicomputers, and network-based cluster platforms. Digital technology has made the computer industry. Now, digital technology is making another wave of fundamental impact to telecommunications and information industries. Converting everything to digital is the key to future success in a highly automated society.

The crossbreeding of technologies demands a new generation of computers that can adapt to scalable, parallel, and distributed computing. These changes in computer and information technologies has prompted computer professionals to study the material presented in this text. The ultimate goal is to become ready for new challenges in the 21st Century.

A Glance at the Book

The book consists of 14 chapters presented in four parts. We provide a balanced coverage of four aspects: *principles, technology, architecture,* and *programming*:

- In Part I, three chapters cover scalable computer platforms and models, basics of parallel programming, and parallel performance metrics.
- Part II assesses commodity microprocessors, distributed cache and memory architecture, switched interconnects, Gigabit networks, and communications.
- Part III covers *symmetric multiprocessors* (SMP) and *cache-coherent, nonuniform memory-access* (CC-NUMA) machines, *clusters of workstations* (COW), and *massively parallel processors* (MPP).
- Part IV presents parallel languages, programming models with emphasis on Unix programming environments, message passing, data parallelism, and the use of PVM, MPI, Fortran 90, and HPF on scalable computers.

A Trilogy on Computer Systems

Over a period of 15 years, Hwang and his associates have produced a trilogy on computer systems, all published by McGraw-Hill Book Company.

- ***Computer Architecture and Parallel Processing***,
  by K. Hwang and F. A. Briggs (1983)
- ***Advanced Computer Architecture***:
  Parallelism, Scalability, Programmability, by K. Hwang (1993)
- ***Scalable Parallel Computing***:
  Technology, Architecture, Programming, by K. Hwang and Z. Xu (1998)

More than 90% of the topics treated in this book are based on new technological advances and research development within the past 5 years. This book is newly written, not a revision of Hwang's previous books. Unique features are highlighted below:

Hot Chips and Interconnects

We assess commodity microprocessors and hot chips for building scalable multiprocessors and multicomputer clusters. Distributed cache/memory and Gigabit networks are studied along with latency hiding mechanisms. In particular, we study multiprocessor buses and crossbar switches, SAN (*System Area Network*), and LAN (*Local Area Network*) such as Gigabit Ethernet, SCI (*Scalable Coherence Interface*), and ATM (*Asynchronous Transfer Mode*) networks.

Scalable Platforms and Clusters

We focus on scalable architectures, fast messaging mechanisms, latency hiding, distributed shared memory, cache coherence protocols, and memory consistency models. We cover software extensions for higher availability, single system image, failure recovery systems, and job management in clusters of computers.

Case studies include the HP/Convex Exemplar, Cray T3D/T3E, IBM SP2, Digital TruCluster, Microsoft Wolfpack, Sun Ultra Enterprise 10000, SGI Origin 2000, Sequent NUMA-Q, Intel/Sandia ASCI Option Red. We discuss the lessons learned from Stanford Dash, Berkeley NOW, Princeton SHRIMP, and Rice TreadMarks.

Parallel Software Environments

This book devotes more than half of the material to software tools and parallel programming systems. In the shared-memory approach, we study the ANSI X3H5, Pthreads, SGI Power C; OpenMP, and C// language. We study Solaris MC and LSF (*Load Sharing Facility*) for availability, single system image, and cluster job management.

For parallel programming, we study data-parallel, message-passing, shared-memory, and implicit paradigms. We study MPI (*Message-Passing Interface*), PVM (*Parallel Virtual Machine*), Fortran 90, and HPF (*High Performance Fortran*) for explicit parallelism; and languages and compliers for implicit parallelism.

Benchmark-based Evaluation

This book benefits from our benchmarking experience with six scalable computers: namely, the SP2 at Maui High Performance Computer Center, the T3D/T3E and Paragon at the San Diego Supercomputer Center, the T3D at Cray Eagan Data Center, the SP2, SGI server, and Pearl cluster at the University of Hong Kong.

Collective MPI communications in various machine platforms are evaluated with firsthand benchmark results. We reveal architectural implications from NASA parallel NAS and USC/HKU STAP benchmark results. These benchmark performance results are evaluated along with scalability analysis over machine sizes and problem sizes.

Web Resources

In a rapidly changing world, any computer book becomes obsolete in a few years. We have strived to prolong the life cycle of this book by selecting practical topics and discussing fundamental issues that can last over generations of computer systems. Examples and quantitative data are drawn from real designs or benchmark experiments.

We have complied at the end of the book an extensive ***Web Resources List***, linking to thousands of home pages of computer companies, research projects, information technology centers, and major application groups across academic, business, and government sectors. An on-line home site of this list is also maintained at HKU. See ***Guide for Instructors/Students*** to access our home pages.

Acknowledgments

We thank the professional reviews of our draft manuscript by six leading experts in this field. Their suggestions are very useful in revising the manuscript to its present form and contents. We appreciate Choming Wang for preparing the indices and the help from Dr. Cho-Li Wang in maintaining the book's Web site at HKU.

Intellectual exchanges with Dharma Agrawal, Jean-Loup Baer, Gordon Bell, David Culler, Jack Dongarra, Michael Flynn, Ian Foster, Jeffrey Fox, Mark Franklin, Wolfgang Giloi, Allan Gottlieb, Anoop Gupta, John Hennessy, Ken Kennedy, Duncan Lawrie, Charles Leiserson, Kai Li, Guojie Li, Lionel Ni, David Patterson, Gregory Pfister, John Rice, Sartaj Sahni, Chuck Seitz, Bruce Shriver, H. J. Siegel, Burton Smith, Daniel Tabak, H. C. Torng, and Ben Wah are always inspiring and appreciated.

We appreciate the sponsorship from McGraw-Hill editors, Eric Munson, Lynn Cox, and Betsy Jones. The production work from Richard DeVitto, Francis Owen, and Nina Kreiden and assistance from Polly Leung of HKU are gratefully acknowledged.

During the courses of writing this book, research funding supports from MIT Lincoln Laboratory, Hong Kong Research Grants Council, and the University of Hong Kong are appreciated. In particular, the excellent facilities and environment provided by HKU make the writing of this book a very pleasant undertakin,

Points of Feedback

For all technical contacts, suggestions, corrections, or exchanges of information, readers and university instructors are advised to contact either author via Email:

kaihwang@cs.hku.hk **zxu@apple.ncic.ac.cn**

We appreciate your feedback and hope you enjoy reading the book.

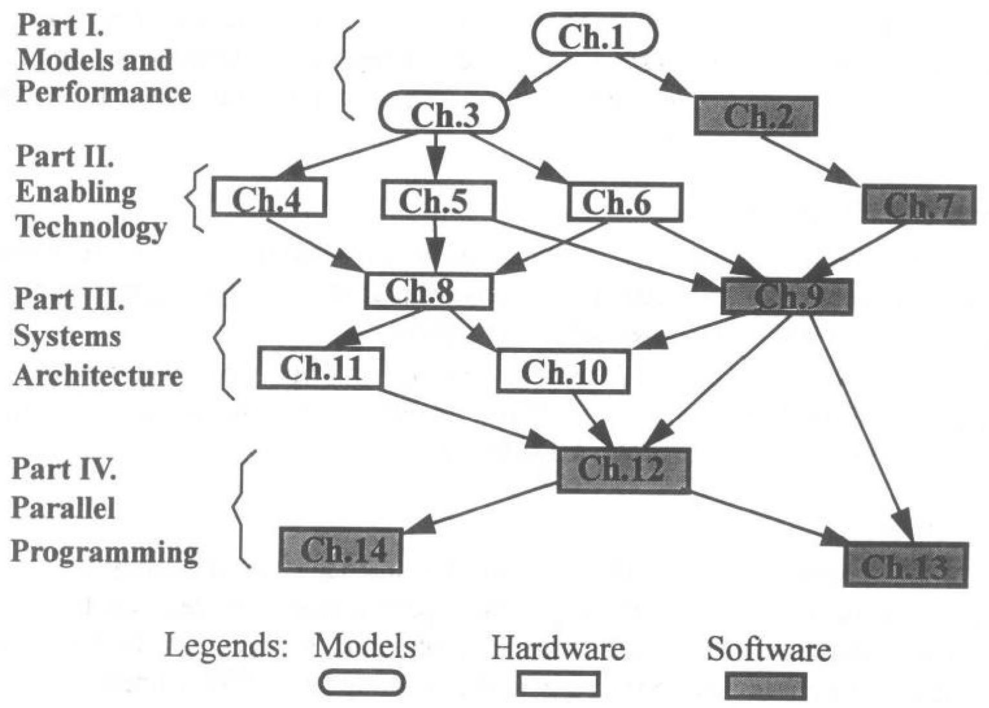
Kai Hwang
Zhiwei Xu

November 15, 1997
Hong Kong

Guide to Instructors/Readers

This book is designed as a standard text for classroom adoption in Computer Science or Computer Engineering curriculum at college/university levels. Suitable courses include: Computer Architecture, Parallel Processing, Distributed Computing, Concurrent Programming, Network-based Computing, Computer Engineering, etc.

Flowchart for Reading The flowchart shows the logical flow of the 14 chapters in this book. The four parts are indicated on the side labels.There are two chapters in theory and modeling (slashed boxes), six unshaded boxes for hardware and architecture chapters, and six shaded boxes for software and programming chapters.



Part I. Models and Performance
Part II. Enabling Technology
Part III. Systems Architecture
Part IV. Parallel Programming
Ch.1
Ch.3
Ch.2
Ch.4
Ch.5
Ch.6
Ch.7
Ch.8
Ch.9
Ch.11
Ch.10
Ch.12
Ch.14
Ch.13
Legends: Models Hardware Software

Course Offerings

Suggested below are five possible course offerings in adopting this book for use in an one-semester course with 45 hours of lectures:

- **Computer Architecture**: For hardware-oriented students in Electrical and Computer Engineering programs, cover Chapters 1, 3-6, and 8-11.
- **Parallel Programming:** For programming and software-oriented students in Computer Science programs, Chapters 1, 2, 7-10, and 12-14 are suitable.

- **Parallel Processing:** For mixed students from Computer Science/Engineering and Electrical Engineering programs, cover Chapters 1, 2, 4-6, 8-10, and 12.
- **Distributed Computing:** For mixed CS and EE students, Chapters 1, 2, 5-7, 9-10, and 13-14 are suitable for one-semester use.
- **Computer Engineering:** An advanced course in computer technology and software system design. Chapters 1, 2, 4-7, and 8-10 should be covered.

All readers should start with the material in chapter 1. Engineering students may read more of the technology and architecture chapters on the left side and software-oriented students on the right subtree of the flowchart. Logically, the reading of this book should flow from the top to the bottom chapters shown by the arrows.

The six hardware and architecture chapters form the core of the course in *Computer Architecture* or in *Computer Engineering*. The six chapters in software and programming form the core for the *Parallel Programming* course. The *Parallel Processing* course covers all scalable parallel systems with more emphasis on shared-memory multiprocessing. The *Distributed Computing* course emphasizes message-passing systems and network-based cluster computing.

Instructor's Manual

An *Instructor's Manual* is available to proven instructors without charge. The Manual provides solutions to selected homework problems, viewing-graph masters, some sample tests, and topics suggested for term projects.

Instructors can request the Manual by writing to: Lynn Cox, McGraw-Hill College Division, 55 Francisco Street, Suite 200, San Franscico, CA. 94133-2117, U.S.A. or submit a written request to Fax No. 415-989-7702.

Web Site Access

As an updated reference, the book is also intended for self-study use by system designers, academic researchers, application programmers, system analysts, resource managers, solution providers, and computer professionals in general. To avoid becoming obsolete, readers are invited to visit our Web site for updated WWW links.

http://www.cs.hku.hk/~kaihwang/book98.html

This Web site is updated dynamically. If you want your organization or your projects to be added into the list. Contact Dr. C. L. Wang of the University of Hong Kong by Email: **clwang@cs.hku.hk**.

Table of Contents

About the Authors IV

Foreword XV

Preface XVI

Guide to Instructors/Readers XIX

Part I Scalability and Clustering 1

Chapter 1 Scalable Computer Platforms and Models 3

- **1.1 Evolution of Computer Architecture** 5
  - 1.1.1 Computer Generations 5
  - 1.1.2 Scalable Computer Architectures 6
  - 1.1.3 Converging System Architectures 8
- **1.2 Dimensions of Scalability** 9
  - 1.2.1 Resource Scalability 9
  - 1.2.2 Application Scalability 11
  - 1.2.3 Technology Scalability 12
- **1.3 Parallel Computer Models** 13
  - 1.3.1 Semantic Attributes 14
  - 1.3.2 Performance Attributes 17
  - 1.3.3 Abstract Machine Models 18
  - 1.3.4 Physical Machine Models 26
- **1.4 Basic Concepts of Clustering** 30
  - 1.4.1 Cluster Characteristics 30
  - 1.4.2 Architectural Comparisons 31
  - 1.4.3 Benefits and Difficulties of Clusters 32
- **1.5 Scalable Design Principles** 37
  - 1.5.1 Principle of Independence 37
  - 1.5.2 Principle of Balanced Design 39
  - 1.5.3 Design for Scalability 44
- **1.6 Bibliographic Notes and Problems** 47

Chapter 2 Basics of Parallel Programming 51

- **2.1 Parallel Programming Overview** 51
  - 2.1.1 Why Is Parallel Programming Difficult? 52
  - 2.1.2 Parallel Programming Environments 55
  - 2.1.3 Parallel Programming Approaches 56
- **2.2 Processes, Tasks, and Threads** 59

2.2.1 Definitions of an Abstract Process 59
2.2.2 Execution Mode 62
2.2.3 Address Space 63
2.2.4 Process Context 65
2.2.5 Process Descriptor 66
2.2.6 Process Control 67
2.2.7 Variations of Process 70

2.3 Parallelism Issues 71
2.3.1 Homogeneity in Processes 72
2.3.2 Static versus Dynamic Parallelism 74
2.3.3 Process Grouping 75
2.3.4 Allocation Issues 76

2.4 Interaction/Communication Issues 77
2.4.1 Interaction Operations 77
2.4.2 Interaction Modes 80
2.4.3 Interaction Patterns 82
2.4.4 Cooperative versus Competitive Interactions 84

2.5 Semantic Issues in Parallel Programs 85
2.5.1 Program Termination 85
2.5.2 Determinacy of Programs 86

2.6 Bibliographic Notes and Problems 87

Chapter 3 Performance Metrics and Benchmarks 91

3.1 System and Application Benchmarks 91
3.1.1 Micro Benchmarks 92
3.1.2 Parallel Computing Benchmarks 96
3.1.3 Business and TPC Benchmarks 98
3.1.4 SPEC Benchmark Family 100

3.2 Performance versus Cost 102
3.2.1 Execution Time and Throughput 103
3.2.2 Utilization and Cost-Effectiveness 104

3.3 Basic Performance Metrics 108
3.3.1 Workload and Speed Metrics 108
3.3.2 Caveats in Sequential Performance 111

3.4 Performance of Parallel Computers 113
3.4.1 Computational Characteristics 113
3.4.2 Parallelism and Interaction Overheads 115
3.4.3 Overhead Quantification 118

3.5 Performance of Parallel Programs 126
3.5.1 Performance Metrics 126
3.5.2 Available Parallelism in Benchmarks 131

3.6 Scalability and Speedup Analysis 134
3.6.1 Amdahl's Law: Fixed Problem Size 134
3.6.2 Gustafson's Law: Fixed Time 136
3.6.3 Sun and Ni's Law: Memory Bounding 139
3.6.4 Isoperformance Models 144

3.7 Bibliographic Notes and Problems 148

Part II Enabling Technologies 153

Chapter 4 Microprocessors as Building Blocks 155

4.1 System Development Trends 155
4.1.1 Advances in Hardware 156
4.1.2 Advances in Software 159
4.1.3 Advances in Applications 160

4.2 Principles of Processor Design 164
4.2.1 Basics of Instruction Pipeline 164
4.2.2 From CISC to RISC and Beyond 169
4.2.3 Architectural Enhancement Approaches 172

4.3 Microprocessor Architecture Families 174
4.3.1 Major Architecture Families 174
4.3.2 Superscalar versus Superpipelined Processors 175
4.3.3 Embedded Microprocessors 180

4.4 Case Studies of Microprocessors 182
4.4.1 Digital's Alpha 21164 Microprocessor 182
4.4.2 Intel Pentium Pro Processor 186

4.5 Post-RISC, Multimedia, and VLIW 191
4.5.1 Post-RISC Processor Features 191
4.5.2 Multimedia Extensions 195
4.5.3 The VLIW Architecture 199

4.6 The Future of Microprocessors 201
4.6.1 Hardware Trends and Physical Limits 201
4.6.2 Future Workloads and Challenges 203
4.6.3 Future Microprocessor Architectures 204

4.7 Bibliographic Notes and Problems 206

Chapter 5 Distributed Memory and Latency Tolerance 211

5.1 Hierarchical Memory Technology 211
5.1.1 Characteristics of Storage Devices 211
5.1.2 Memory Hierarchy Properties 214
5.1.3 Memory Capacity Planning 217

5.2 Cache Coherence Protocols 220
5.2.1 Cache Coherency Problem 220
5.2.2 Snoopy Coherency Protocols 222
5.2.3 The MESI Snoopy Protocol 224

5.3 Shared-Memory Consistency 228
5.3.1 Memory Event Ordering 228
5.3.2 Memory Consistency Models 231
5.3.3 Relaxed Memory Models 234

5.4 Distributed Cache/Memory Architecture 237
5.4.1 NORMA, NUMA, COMA, and DSM Models 237
5.4.2 Directory-Based Coherency Protocol 243
5.4.3 The Stanford Dash Multiprocessor 245
5.4.4 Directory-Based Protocol in Dash 248

5.5 Latency Tolerance Techniques 250
5.5.1 Latency Avoidance, Reduction, and Hiding 250
5.5.2 Distributed Coherent Caches 253
5.5.3 Data Prefetching Strategies 255
5.5.4 Effects of Relaxed Memory Consistency 257
5.6 Multithreaded Latency Hiding 257
5.6.1 Multithreaded Processor Model 258
5.6.2 Context-Switching Policies 260
5.6.3 Combining Latency Hiding Mechanisms 265
5.7 Bibliographic Notes and Problems 266

Chapter 6 System Interconnects and Gigabit Networks 273
6.1 Basics of Interconnection Network 273
6.1.1 Interconnection Environments 273
6.1.2 Network Components 276
6.1.3 Network Characteristics 277
6.1.4 Network Performance Metrics 280
6.2 Network Topologies and Properties 281
6.2.1 Topological and Functional Properties 281
6.2.2 Routing Schemes and Functions 283
6.2.3 Networking Topologies 286
6.3 Buses, Crossbar, and Multistage Switches 294
6.3.1 Multiprocessor Buses 294
6.3.2 Crossbar Switches 298
6.3.3 Multistage Interconnection Networks 301
6.3.4 Comparison of Switched Interconnects 305
6.4 Gigabit Network Technologies 307
6.4.1 Fiber Channel and FDDI Rings 307
6.4.2 Fast Ethernet and Gigabit Ethernet 310
6.4.3 Myrinet for SAN/LAN Construction 313
6.4.4 HiPPI and SuperHiPPI 314
6.5 ATM Switches and Networks 318
6.5.1 ATM Technology 318
6.5.2 ATM Network Interfaces 320
6.5.3 Four Layers of ATM Architecture 321
6.5.4 ATM Internetwork Connectivity 324
6.6 Scalable Coherence Interface 326
6.6.1 SCI Interconnects 327
6.6.2 Implementation Issues 329
6.6.3 SCI Coherence Protocol 332
6.7 Comparison of Network Technologies 334
6.7.1 Standard Networks and Perspectives 334
6.7.2 Network Performance and Applications 335
6.8 Bibliographic Notes and Problems 337

Chapter 7 Threading, Synchronization, and Communication 343
7.1 Software Multithreading 343

7.1.1 The Thread Concept 344
7.1.2 Threads Management 346
7.1.3 Thread Synchronization 348

7.2 Synchronization Mechanisms 349
7.2.1 Atomicity versus Mutual Exclusion 349
7.2.2 High-Level Synchronization Constructs 355
7.2.3 Low-Level Synchronization Primitives 360
7.2.4 Fast Locking Mechanisms 364

7.3 The TCP/IP Communication Protocol Suite 366
7.3.1 Features of The TCP/IP Suite 367
7.3.2 UDP, TCP, and IP 371
7.3.3 The Sockets Interface 375

7.4 Fast and Efficient Communication 376
7.4.1 Key Problems in Communication 377
7.4.2 The Log P Communication Model 384
7.4.3 Low-Level Communications Support 386
7.4.4 Communication Algorithms 396

7.5 Bibliographic Notes and Problems 398

Part III Systems Architecture 403

Chapter 8 Symmetric and CC-NUMA Multiprocessors 407

8.1 SMP and CC-NUMA Technology 407
8.1.1 Multiprocessor Architecture 407
8.1.2 Commercial SMP Servers 412
8.1.3 The Intel SHV Server Board 413

8.2 Sun Ultra Enterprise 10000 System 416
8.2.1 The Ultra E-10000 Architecture 416
8.2.2 System Board Architecture 418
8.2.3 Scalability and Availability Support 418
8.2.4 Dynamic Domains and Performance 420

8.3 HP/Convex Exemplar X-Class 421
8.3.1 The Exemplar X System Architecture 421
8.3.2 Exemplar Software Environment 424

8.4 The Sequent NUMA-Q 2000 425
8.4.1 The NUMA-Q 2000 Architecture 426
8.4.2 Software Environment of NUMA-Q 430
8.4.3 Performance of the NUMA-Q 431

8.5 The SGI/Cray Origin 2000 Superserver 434
8.5.1 Design Goals of Origin 2000 Series 434
8.5.2 The Origin 2000 Architecture 435
8.5.3 The Cellular IRIX Environment 443
8.5.4 Performance of the Origin 2000 447

8.6 Comparison of CC-NUMA Architectures 447

8.7 Bibliographic Notes and Problems 451