

Contents

Preface xvii

Acknowledgments xix

CHAPTER 1

Introduction to Parallel Computing 1

- 1.1 Motivating Parallelism 2
 - 1.1.1 *The Computational Power Argument – from Transistors to FLOPS* 2
 - 1.1.2 *The Memory/Disk Speed Argument* 3
 - 1.1.3 *The Data Communication Argument* 4
- 1.2 Scope of Parallel Computing 4
 - 1.2.1 *Applications in Engineering and Design* 4
 - 1.2.2 *Scientific Applications* 5
 - 1.2.3 *Commercial Applications* 5
 - 1.2.4 *Applications in Computer Systems* 6
- 1.3 Organization and Contents of the Text 6
- 1.4 Bibliographic Remarks 8
 - Problems 9

CHAPTER 2

Parallel Programming Platforms 11

- 2.1 Implicit Parallelism: Trends in Microprocessor Architectures 12
 - 2.1.1 *Pipelining and Superscalar Execution* 12
 - 2.1.2 *Very Long Instruction Word Processors* 15
- 2.2 Limitations of Memory System Performance* 16
 - 2.2.1 *Improving Effective Memory Latency Using Caches* 17
 - 2.2.2 *Impact of Memory Bandwidth* 18
 - 2.2.3 *Alternate Approaches for Hiding Memory Latency* 21

| | | |
|-------|---|----|
| 2.2.4 | <i>Tradeoffs of Multithreading and Prefetching</i> | 23 |
| 2.3 | Dichotomy of Parallel Computing Platforms | 24 |
| 2.3.1 | <i>Control Structure of Parallel Platforms</i> | 25 |
| 2.3.2 | <i>Communication Model of Parallel Platforms</i> | 27 |
| 2.4 | Physical Organization of Parallel Platforms | 31 |
| 2.4.1 | <i>Architecture of an Ideal Parallel Computer</i> | 31 |
| 2.4.2 | <i>Interconnection Networks for Parallel Computers</i> | 32 |
| 2.4.3 | <i>Network Topologies</i> | 33 |
| 2.4.4 | <i>Evaluating Static Interconnection Networks</i> | 43 |
| 2.4.5 | <i>Evaluating Dynamic Interconnection Networks</i> | 44 |
| 2.4.6 | <i>Cache Coherence in Multiprocessor Systems</i> | 45 |
| 2.5 | Communication Costs in Parallel Machines | 53 |
| 2.5.1 | <i>Message Passing Costs in Parallel Computers</i> | 53 |
| 2.5.2 | <i>Communication Costs in Shared-Address-Space Machines</i> | 61 |
| 2.6 | Routing Mechanisms for Interconnection Networks | 63 |
| 2.7 | Impact of Process-Processor Mapping and Mapping Techniques | 65 |
| 2.7.1 | <i>Mapping Techniques for Graphs</i> | 66 |
| 2.7.2 | <i>Cost-Performance Tradeoffs</i> | 73 |
| 2.8 | Bibliographic Remarks | 74 |
| | Problems | 76 |

CHAPTER 3

Principles of Parallel Algorithm Design 85

| | | |
|-------|---|-----|
| 3.1 | Preliminaries | 86 |
| 3.1.1 | <i>Decomposition, Tasks, and Dependency Graphs</i> | 86 |
| 3.1.2 | <i>Granularity, Concurrency, and Task-Interaction</i> | 89 |
| 3.1.3 | <i>Processes and Mapping</i> | 93 |
| 3.1.4 | <i>Processes versus Processors</i> | 94 |
| 3.2 | Decomposition Techniques | 95 |
| 3.2.1 | <i>Recursive Decomposition</i> | 95 |
| 3.2.2 | <i>Data Decomposition</i> | 97 |
| 3.2.3 | <i>Exploratory Decomposition</i> | 105 |
| 3.2.4 | <i>Speculative Decomposition</i> | 107 |
| 3.2.5 | <i>Hybrid Decompositions</i> | 109 |
| 3.3 | Characteristics of Tasks and Interactions | 110 |
| 3.3.1 | <i>Characteristics of Tasks</i> | 110 |
| 3.3.2 | <i>Characteristics of Inter-Task Interactions</i> | 112 |
| 3.4 | Mapping Techniques for Load Balancing | 115 |
| 3.4.1 | <i>Schemes for Static Mapping</i> | 117 |
| 3.4.2 | <i>Schemes for Dynamic Mapping</i> | 130 |

| | | |
|-------|--|-----|
| 3.5 | Methods for Containing Interaction Overheads | 132 |
| 3.5.1 | <i>Maximizing Data Locality</i> | 132 |
| 3.5.2 | <i>Minimizing Contention and Hot Spots</i> | 134 |
| 3.5.3 | <i>Overlapping Computations with Interactions</i> | 135 |
| 3.5.4 | <i>Replicating Data or Computations</i> | 136 |
| 3.5.5 | <i>Using Optimized Collective Interaction Operations</i> | 137 |
| 3.5.6 | <i>Overlapping Interactions with Other Interactions</i> | 138 |
| 3.6 | Parallel Algorithm Models | 139 |
| 3.6.1 | <i>The Data-Parallel Model</i> | 139 |
| 3.6.2 | <i>The Task Graph Model</i> | 140 |
| 3.6.3 | <i>The Work Pool Model</i> | 140 |
| 3.6.4 | <i>The Master-Slave Model</i> | 141 |
| 3.6.5 | <i>The Pipeline or Producer-Consumer Model</i> | 141 |
| 3.6.6 | <i>Hybrid Models</i> | 142 |
| 3.7 | Bibliographic Remarks | 142 |
| | Problems | 143 |

CHAPTER 4

Basic Communication Operations 147

| | | |
|-------|---|-----|
| 4.1 | One-to-All Broadcast and All-to-One Reduction | 149 |
| 4.1.1 | <i>Ring or Linear Array</i> | 149 |
| 4.1.2 | <i>Mesh</i> | 152 |
| 4.1.3 | <i>Hypercube</i> | 153 |
| 4.1.4 | <i>Balanced Binary Tree</i> | 153 |
| 4.1.5 | <i>Detailed Algorithms</i> | 154 |
| 4.1.6 | <i>Cost Analysis</i> | 156 |
| 4.2 | All-to-All Broadcast and Reduction | 157 |
| 4.2.1 | <i>Linear Array and Ring</i> | 158 |
| 4.2.2 | <i>Mesh</i> | 160 |
| 4.2.3 | <i>Hypercube</i> | 161 |
| 4.2.4 | <i>Cost Analysis</i> | 164 |
| 4.3 | All-Reduce and Prefix-Sum Operations | 166 |
| 4.4 | Scatter and Gather | 167 |
| 4.5 | All-to-All Personalized Communication | 170 |
| 4.5.1 | <i>Ring</i> | 173 |
| 4.5.2 | <i>Mesh</i> | 174 |
| 4.5.3 | <i>Hypercube</i> | 175 |
| 4.6 | Circular Shift | 179 |
| 4.6.1 | <i>Mesh</i> | 179 |
| 4.6.2 | <i>Hypercube</i> | 181 |

x Contents

| | | |
|-------|--|-----|
| 4.7 | Improving the Speed of Some Communication Operations | 184 |
| 4.7.1 | <i>Splitting and Routing Messages in Parts</i> | 184 |
| 4.7.2 | <i>All-Port Communication</i> | 186 |
| 4.8 | Summary | 187 |
| 4.9 | Bibliographic Remarks | 188 |
| | Problems | 190 |

CHAPTER 5

Analytical Modeling of Parallel Programs 195

| | | |
|-------|---|-----|
| 5.1 | Sources of Overhead in Parallel Programs | 195 |
| 5.2 | Performance Metrics for Parallel Systems | 197 |
| 5.2.1 | <i>Execution Time</i> | 197 |
| 5.2.2 | <i>Total Parallel Overhead</i> | 197 |
| 5.2.3 | <i>Speedup</i> | 198 |
| 5.2.4 | <i>Efficiency</i> | 202 |
| 5.2.5 | <i>Cost</i> | 203 |
| 5.3 | The Effect of Granularity on Performance | 205 |
| 5.4 | Scalability of Parallel Systems | 208 |
| 5.4.1 | <i>Scaling Characteristics of Parallel Programs</i> | 209 |
| 5.4.2 | <i>The Isoefficiency Metric of Scalability</i> | 212 |
| 5.4.3 | <i>Cost-Optimality and the Isoefficiency Function</i> | 217 |
| 5.4.4 | <i>A Lower Bound on the Isoefficiency Function</i> | 217 |
| 5.4.5 | <i>The Degree of Concurrency and the Isoefficiency Function</i> | 218 |
| 5.5 | Minimum Execution Time and Minimum Cost-Optimal Execution Time | 218 |
| 5.6 | Asymptotic Analysis of Parallel Programs | 221 |
| 5.7 | Other Scalability Metrics | 222 |
| 5.8 | Bibliographic Remarks | 226 |
| | Problems | 228 |

CHAPTER 6

Programming Using the Message-Passing Paradigm 233

| | | |
|-------|--|-----|
| 6.1 | Principles of Message-Passing Programming | 233 |
| 6.2 | The Building Blocks: Send and Receive Operations | 235 |
| 6.2.1 | <i>Blocking Message Passing Operations</i> | 236 |
| 6.2.2 | <i>Non-Blocking Message Passing Operations</i> | 239 |
| 6.3 | MPI: the Message Passing Interface | 240 |

| | | |
|--------|--|-----|
| 6.3.1 | <i>Starting and Terminating the MPI Library</i> | 242 |
| 6.3.2 | <i>Communicators</i> | 242 |
| 6.3.3 | <i>Getting Information</i> | 243 |
| 6.3.4 | <i>Sending and Receiving Messages</i> | 244 |
| 6.3.5 | <i>Example: Odd-Even Sort</i> | 248 |
| 6.4 | Topologies and Embedding | 250 |
| 6.4.1 | <i>Creating and Using Cartesian Topologies</i> | 251 |
| 6.4.2 | <i>Example: Cannon's Matrix-Matrix Multiplication</i> | 253 |
| 6.5 | Overlapping Communication with Computation | 255 |
| 6.5.1 | <i>Non-Blocking Communication Operations</i> | 255 |
| 6.6 | Collective Communication and Computation Operations | 260 |
| 6.6.1 | <i>Barrier</i> | 260 |
| 6.6.2 | <i>Broadcast</i> | 260 |
| 6.6.3 | <i>Reduction</i> | 261 |
| 6.6.4 | <i>Prefix</i> | 263 |
| 6.6.5 | <i>Gather</i> | 263 |
| 6.6.6 | <i>Scatter</i> | 264 |
| 6.6.7 | <i>All-to-All</i> | 265 |
| 6.6.8 | <i>Example: One-Dimensional Matrix-Vector Multiplication</i> | 266 |
| 6.6.9 | <i>Example: Single-Source Shortest-Path</i> | 268 |
| 6.6.10 | <i>Example: Sample Sort</i> | 270 |
| 6.7 | Groups and Communicators | 272 |
| 6.7.1 | <i>Example: Two-Dimensional Matrix-Vector Multiplication</i> | 274 |
| 6.8 | Bibliographic Remarks | 276 |
| | Problems | 277 |

CHAPTER 7

Programming Shared Address Space

Platforms 279

| | | |
|-------|---|-----|
| 7.1 | Thread Basics | 280 |
| 7.2 | Why Threads? | 281 |
| 7.3 | The POSIX Thread API | 282 |
| 7.4 | Thread Basics: Creation and Termination | 282 |
| 7.5 | Synchronization Primitives in Pthreads | 287 |
| 7.5.1 | <i>Mutual Exclusion for Shared Variables</i> | 287 |
| 7.5.2 | <i>Condition Variables for Synchronization</i> | 294 |
| 7.6 | Controlling Thread and Synchronization Attributes | 298 |
| 7.6.1 | <i>Attributes Objects for Threads</i> | 299 |
| 7.6.2 | <i>Attributes Objects for Mutexes</i> | 300 |

| | | |
|--------|---|-----|
| 7.7 | Thread Cancellation | 301 |
| 7.8 | Composite Synchronization Constructs | 302 |
| 7.8.1 | <i>Read-Write Locks</i> | 302 |
| 7.8.2 | <i>Barriers</i> | 307 |
| 7.9 | Tips for Designing Asynchronous Programs | 310 |
| 7.10 | OpenMP: a Standard for Directive Based Parallel Programming | 311 |
| 7.10.1 | <i>The OpenMP Programming Model</i> | 312 |
| 7.10.2 | <i>Specifying Concurrent Tasks in OpenMP</i> | 315 |
| 7.10.3 | <i>Synchronization Constructs in OpenMP</i> | 322 |
| 7.10.4 | <i>Data Handling in OpenMP</i> | 327 |
| 7.10.5 | <i>OpenMP Library Functions</i> | 328 |
| 7.10.6 | <i>Environment Variables in OpenMP</i> | 330 |
| 7.10.7 | <i>Explicit Threads versus OpenMP Based Programming</i> | 331 |
| 7.11 | Bibliographic Remarks | 332 |
| | Problems | 332 |

CHAPTER 8

Dense Matrix Algorithms 337

| | | |
|-------|--|-----|
| 8.1 | Matrix-Vector Multiplication | 337 |
| 8.1.1 | <i>Rowwise 1-D Partitioning</i> | 338 |
| 8.1.2 | <i>2-D Partitioning</i> | 341 |
| 8.2 | Matrix-Matrix Multiplication | 345 |
| 8.2.1 | <i>A Simple Parallel Algorithm</i> | 346 |
| 8.2.2 | <i>Cannon's Algorithm</i> | 347 |
| 8.2.3 | <i>The DNS Algorithm</i> | 349 |
| 8.3 | Solving a System of Linear Equations | 352 |
| 8.3.1 | <i>A Simple Gaussian Elimination Algorithm</i> | 353 |
| 8.3.2 | <i>Gaussian Elimination with Partial Pivoting</i> | 366 |
| 8.3.3 | <i>Solving a Triangular System: Back-Substitution</i> | 369 |
| 8.3.4 | <i>Numerical Considerations in Solving Systems of Linear Equations</i> | 370 |
| 8.4 | Bibliographic Remarks | 371 |
| | Problems | 372 |

CHAPTER 9

Sorting 379

| | | |
|-------|--|-----|
| 9.1 | Issues in Sorting on Parallel Computers | 380 |
| 9.1.1 | <i>Where the Input and Output Sequences are Stored</i> | 380 |
| 9.1.2 | <i>How Comparisons are Performed</i> | 380 |

| | | |
|-----|---|-----|
| 9.2 | Sorting Networks | 382 |
| | 9.2.1 <i>Bitonic Sort</i> | 384 |
| | 9.2.2 <i>Mapping Bitonic Sort to a Hypercube and a Mesh</i> | 387 |
| 9.3 | Bubble Sort and its Variants | 394 |
| | 9.3.1 <i>Odd-Even Transposition</i> | 395 |
| | 9.3.2 <i>Shellsort</i> | 398 |
| 9.4 | Quicksort | 399 |
| | 9.4.1 <i>Parallelizing Quicksort</i> | 401 |
| | 9.4.2 <i>Parallel Formulation for a CRCW PRAM</i> | 402 |
| | 9.4.3 <i>Parallel Formulation for Practical Architectures</i> | 404 |
| | 9.4.4 <i>Pivot Selection</i> | 411 |
| 9.5 | Bucket and Sample Sort | 412 |
| 9.6 | Other Sorting Algorithms | 414 |
| | 9.6.1 <i>Enumeration Sort</i> | 414 |
| | 9.6.2 <i>Radix Sort</i> | 415 |
| 9.7 | Bibliographic Remarks | 416 |
| | Problems | 419 |

CHAPTER 10

Graph Algorithms 429

| | | |
|------|--|-----|
| 10.1 | Definitions and Representation | 429 |
| 10.2 | Minimum Spanning Tree: Prim's Algorithm | 432 |
| 10.3 | Single-Source Shortest Paths: Dijkstra's Algorithm | 436 |
| 10.4 | All-Pairs Shortest Paths | 437 |
| | 10.4.1 <i>Dijkstra's Algorithm</i> | 438 |
| | 10.4.2 <i>Floyd's Algorithm</i> | 440 |
| | 10.4.3 <i>Performance Comparisons</i> | 445 |
| 10.5 | Transitive Closure | 445 |
| 10.6 | Connected Components | 446 |
| | 10.6.1 <i>A Depth-First Search Based Algorithm</i> | 446 |
| 10.7 | Algorithms for Sparse Graphs | 450 |
| | 10.7.1 <i>Finding a Maximal Independent Set</i> | 451 |
| | 10.7.2 <i>Single-Source Shortest Paths</i> | 455 |
| 10.8 | Bibliographic Remarks | 462 |
| | Problems | 465 |

CHAPTER 11

Search Algorithms for Discrete Optimization Problems 469

- 11.1 Definitions and Examples 469
- 11.2 Sequential Search Algorithms 474
 - 11.2.1 *Depth-First Search Algorithms* 474
 - 11.2.2 *Best-First Search Algorithms* 478
- 11.3 Search Overhead Factor 478
- 11.4 Parallel Depth-First Search 480
 - 11.4.1 *Important Parameters of Parallel DFS* 482
 - 11.4.2 *A General Framework for Analysis of Parallel DFS* 485
 - 11.4.3 *Analysis of Load-Balancing Schemes* 488
 - 11.4.4 *Termination Detection* 490
 - 11.4.5 *Experimental Results* 492
 - 11.4.6 *Parallel Formulations of Depth-First Branch-and-Bound Search* 495
 - 11.4.7 *Parallel Formulations of IDA** 496
- 11.5 Parallel Best-First Search 496
- 11.6 Speedup Anomalies in Parallel Search Algorithms 501
 - 11.6.1 *Analysis of Average Speedup in Parallel DFS* 502
- 11.7 Bibliographic Remarks 505
 - Problems 510

CHAPTER 12

Dynamic Programming 515

- 12.1 Overview of Dynamic Programming 515
- 12.2 Serial Monadic DP Formulations 518
 - 12.2.1 *The Shortest-Path Problem* 518
 - 12.2.2 *The 0/1 Knapsack Problem* 520
- 12.3 Nonserial Monadic DP Formulations 523
 - 12.3.1 *The Longest-Common-Subsequence Problem* 523
- 12.4 Serial Polyadic DP Formulations 526
 - 12.4.1 *Floyd's All-Pairs Shortest-Paths Algorithm* 526
- 12.5 Nonserial Polyadic DP Formulations 527
 - 12.5.1 *The Optimal Matrix-Parenthesization Problem* 527
- 12.6 Summary and Discussion 530
- 12.7 Bibliographic Remarks 531
 - Problems 532

CHAPTER 13

Fast Fourier Transform 537

- 13.1 The Serial Algorithm 538
- 13.2 The Binary-Exchange Algorithm 541
 - 13.2.1 *A Full Bandwidth Network* 541
 - 13.2.2 *Limited Bandwidth Network* 548
 - 13.2.3 *Extra Computations in Parallel FFT* 551
- 13.3 The Transpose Algorithm 553
 - 13.3.1 *Two-Dimensional Transpose Algorithm* 553
 - 13.3.2 *The Generalized Transpose Algorithm* 556
- 13.4 Bibliographic Remarks 560
 - Problems 562

APPENDIX A

Complexity of Functions and Order Analysis 565

- A.1 Complexity of Functions 565
- A.2 Order Analysis of Functions 566

Bibliography 569**Author Index 611****Subject Index 621**