

Parallel Programming **in C with MPI** **and OpenMP**

Michael J. Quinn

Oregon State University



Higher Education

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Kuala Lumpur Lisbon London Madrid Mexico City
Milan Montreal New Delhi Santiago Seoul Singapore Sydney Taipei Toronto



PARALLEL PROGRAMMING IN C WITH MPI AND OPENMP
International Edition 2003

Exclusive rights by McGraw-Hill Education (Asia), for manufacture and export. This book cannot be re-exported from the country to which it is sold by McGraw-Hill. The International Edition is not available in North America.

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2004 by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

10 09 08 07 06
20 09 08 07
CTF SLP

Library of Congress Cataloging-in-Publication Data

Quinn, Michael J. (Michael Jay)

Parallel programming in C with MPI and OpenMP / Michael J. Quinn. — 1st ed.

p. cm.

ISBN 007-282256-2

1. C (Computer program language). 2. Parallel programming (Computer science). I. Title.
QA76.73.C15Q55 2004

005.13'3—dc21

2003046371

CIP

When ordering this title, use ISBN 007-123265-6

Printed in Singapore

www.mhhe.com

BRIEF TABLE OF CONTENTS

Preface xiv

1	Motivation and History	1
2	Parallel Architectures	27
3	Parallel Algorithm Design	63
4	Message-Passing Programming	93
5	The Sieve of Eratosthenes	115
6	Floyd's Algorithm	137
7	Performance Analysis	159
8	Matrix-Vector Multiplication	178
9	Document Classification	216
10	Monte Carlo Methods	239
11	Matrix Multiplication	273
12	Solving Linear Systems	290
13	Finite Difference Methods	318
14	Sorting	338
15	The Fast Fourier Transform	353
16	Combinatorial Search	369
17	Shared-Memory Programming	404
18	Combining MPI and OpenMP	436
	Appendix A MPI Functions	450
	Appendix B Utility Functions	485
	Appendix C Debugging MPI Programs	505
	Appendix D Review of Complex Numbers	509
	Appendix E OpenMP Functions	513

Bibliography 515

Author Index 520

Subject Index 522

CONTENTS

Preface xiv

CHAPTER 1

Motivation and History 1

- 1.1 Introduction 1
- 1.2 Modern Scientific Method 3
- 1.3 Evolution of Supercomputing 4
- 1.4 Modern Parallel Computers 5
 - 1.4.1 *The Cosmic Cube* 6
 - 1.4.2 *Commercial Parallel Computers* 6
 - 1.4.3 *Beowulf* 7
 - 1.4.4 *Advanced Strategic Computing Initiative* 8
- 1.5 Seeking Concurrency 9
 - 1.5.1 *Data Dependence Graphs* 9
 - 1.5.2 *Data Parallelism* 10
 - 1.5.3 *Functional Parallelism* 10
 - 1.5.4 *Pipelining* 12
 - 1.5.5 *Size Considerations* 13
- 1.6 Data Clustering 14
- 1.7 Programming Parallel Computers 17
 - 1.7.1 *Extend a Compiler* 17
 - 1.7.2 *Extend a Sequential Programming Language* 18
 - 1.7.3 *Add a Parallel Programming Layer* 19
 - 1.7.4 *Create a Parallel Language* 19
 - 1.7.5 *Current Status* 21
- 1.8 Summary 21
- 1.9 Key Terms 22
- 1.10 Bibliographic Notes 22
- 1.11 Exercises 23

CHAPTER 2

Parallel Architectures 27

- 2.1 Introduction 27
- 2.2 Interconnection Networks 28
 - 2.2.1 *Shared versus Switched Media* 28
 - 2.2.2 *Switch Network Topologies* 29
 - 2.2.3 *2-D Mesh Network* 29
 - 2.2.4 *Binary Tree Network* 30
 - 2.2.5 *Hypertree Network* 31
 - 2.2.6 *Butterfly Network* 32
 - 2.2.7 *Hypercube Network* 33
 - 2.2.8 *Shuffle-exchange Network* 35
 - 2.2.9 *Summary* 36
- 2.3 Processor Arrays 37
 - 2.3.1 *Architecture and Data-parallel Operations* 37
 - 2.3.2 *Processor Array Performance* 39
 - 2.3.3 *Processor Interconnection Network* 40
 - 2.3.4 *Enabling and Disabling Processors* 40
 - 2.3.5 *Additional Architectural Features* 42
 - 2.3.6 *Shortcomings of Processor Arrays* 42
- 2.4 Multiprocessors 43
 - 2.4.1 *Centralized Multiprocessors* 43
 - 2.4.2 *Distributed Multiprocessors* 45
- 2.5 Multicomputers 49
 - 2.5.1 *Asymmetrical Multicomputers* 49
 - 2.5.2 *Symmetrical Multicomputers* 51
 - 2.5.3 *Which Model Is Best for a Commodity Cluster?* 52
 - 2.5.4 *Differences between Clusters and Networks of Workstations* 53
- 2.6 Flynn's Taxonomy 54
 - 2.6.1 *SISD* 54
 - 2.6.2 *SIMD* 55

- 2.6.3 MISD 55
- 2.6.4 MIMD 56
- 2.7 Summary 58
- 2.8 Key Terms 59
- 2.9 Bibliographic Notes 59
- 2.10 Exercises 60

CHAPTER 3

Parallel Algorithm Design 63

- 3.1 Introduction 63
- 3.2 The Task/Chamel Model 63
- 3.3 Foster's Design Methodology 64
 - 3.3.1 Partitioning 65
 - 3.3.2 Communication 67
 - 3.3.3 Agglomeration 68
 - 3.3.4 Mapping 70
- 3.4 Boundary Value Problem 73
 - 3.4.1 Introduction 73
 - 3.4.2 Partitioning 75
 - 3.4.3 Communication 75
 - 3.4.4 Agglomeration and Mapping 76
 - 3.4.5 Analysis 76
- 3.5 Finding the Maximum 77
 - 3.5.1 Introduction 77
 - 3.5.2 Partitioning 77
 - 3.5.3 Communication 77
 - 3.5.4 Agglomeration and Mapping 81
 - 3.5.5 Analysis 82
- 3.6 The n -Body Problem 82
 - 3.6.1 Introduction 82
 - 3.6.2 Partitioning 83
 - 3.6.3 Communication 83
 - 3.6.4 Agglomeration and Mapping 85
 - 3.6.5 Analysis 85
- 3.7 Adding Data Input 86
 - 3.7.1 Introduction 86
 - 3.7.2 Communication 87
 - 3.7.3 Analysis 88
- 3.8 Summary 89

- 3.9 Key Terms 90
- 3.10 Bibliographic Notes 90
- 3.11 Exercises 90

CHAPTER 4

Message-Passing Programming 93

- 4.1 Introduction 93
- 4.2 The Message-Passing Model 94
- 4.3 The Message-Passing Interface 95
- 4.4 Circuit Satisfiability 96
 - 4.4.1 Function `MPI_Init` 99
 - 4.4.2 Functions `MPI_Comm_rank` and `MPI_Comm_size` 99
 - 4.4.3 Function `MPI_Finalize` 101
 - 4.4.4 Compiling MPI Programs 102
 - 4.4.5 Running MPI Programs 102
- 4.5 Introducing Collective Communication 104
 - 4.5.1 Function `MPI_Reduce` 105
- 4.6 Benchmarking Parallel Performance 108
 - 4.6.1 Functions `MPI_Wtime` and `MPI_Wtick` 108
 - 4.6.2 Function `MPI_Barrier` 108
- 4.7 Summary 110
- 4.8 Key Terms 110
- 4.9 Bibliographic Notes 110
- 4.10 Exercises 111

CHAPTER 5

The Sieve of Eratosthenes 115

- 5.1 Introduction 115
- 5.2 Sequential Algorithm 115
- 5.3 Sources of Parallelism 117
- 5.4 Data Decomposition Options 117
 - 5.4.1 Interleaved Data Decomposition 118
 - 5.4.2 Block Data Decomposition 118
 - 5.4.3 Block Decomposition Macros 120
 - 5.4.4 Local Index versus Global Index 120

- 5.4.5 *Ramifications of Block Decomposition* 121
- 5.5 Developing the Parallel Algorithm 121
 - 5.5.1 *Function MPI_Bcast* 122
- 5.6 Analysis of Parallel Sieve Algorithm 122
- 5.7 Documenting the Parallel Program 123
- 5.8 Benchmarking 128
- 5.9 Improvements 129
 - 5.9.1 *Delete Even Integers* 129
 - 5.9.2 *Eliminate Broadcast* 130
 - 5.9.3 *Reorganize Loops* 131
 - 5.9.4 *Benchmarking* 131
- 5.10 Summary 133
- 5.11 Key Terms 134
- 5.12 Bibliographic Notes 134
- 5.13 Exercises 134

CHAPTER 6

- Floyd's Algorithm** 137
 - 6.1 Introduction 137
 - 6.2 The All-Pairs Shortest-Path Problem 137
 - 6.3 Creating Arrays at Run Time 139
 - 6.4 Designing the Parallel Algorithm 140
 - 6.4.1 *Partitioning* 140
 - 6.4.2 *Communication* 141
 - 6.4.3 *Agglomeration and Mapping* 142
 - 6.4.4 *Matrix Input/Output* 143
 - 6.5 Point-to-Point Communication 145
 - 6.5.1 *Function MPI_Send* 146
 - 6.5.2 *Function MPI_Recv* 147
 - 6.5.3 *Deadlock* 148
 - 6.6 Documenting the Parallel Program 149
 - 6.7 Analysis and Benchmarking 151
 - 6.8 Summary 154
 - 6.9 Key Terms 154
 - 6.10 Bibliographic Notes 154
 - 6.11 Exercises 154

CHAPTER 7

Performance Analysis 159

- 7.1 Introduction 159
- 7.2 Speedup and Efficiency 159
- 7.3 Amdahl's Law 161
 - 7.3.1 *Limitations of Amdahl's Law* 164
 - 7.3.2 *The Amdahl Effect* 164
- 7.4 Gustafson-Barsis's Law 164
- 7.5 The Karp-Flatt Metric 167
- 7.6 The Isoefficiency Metric 170
- 7.7 Summary 174
- 7.8 Key Terms 175
- 7.9 Bibliographic Notes 175
- 7.10 Exercises 176

CHAPTER 8

Matrix-Vector Multiplication 178

- 8.1 Introduction 178
- 8.2 Sequential Algorithm 179
- 8.3 Data Decomposition Options 180
- 8.4 Rowwise Block-Striped Decomposition 181
 - 8.4.1 *Design and Analysis* 181
 - 8.4.2 *Replicating a Block-Mapped Vector* 183
 - 8.4.3 *Function MPI_Allgatherv* 184
 - 8.4.4 *Replicated Vector Input/Output* 186
 - 8.4.5 *Documenting the Parallel Program* 187
 - 8.4.6 *Benchmarking* 187
- 8.5 Columnwise Block-Striped Decomposition 189
 - 8.5.1 *Design and Analysis* 189
 - 8.5.2 *Reading a Columnwise Block-Striped Matrix* 191
 - 8.5.3 *Function MPI_Scatterv* 191
 - 8.5.4 *Printing a Columnwise Block-Striped Matrix* 193
 - 8.5.5 *Function MPI_Gatherv* 193
 - 8.5.6 *Distributing Partial Results* 195

- 8.5.7 *Function MPI_Alltoallv* 195
 - 8.5.8 *Documenting the Parallel Program* 196
 - 8.5.9 *Benchmarking* 198
 - 8.6 **Checkerboard Block Decomposition** 199
 - 8.6.1 *Design and Analysis* 199
 - 8.6.2 *Creating a Communicator* 202
 - 8.6.3 *Function MPI_Dims_create* 203
 - 8.6.4 *Function MPI_Cart_create* 204
 - 8.6.5 *Reading a Checkerboard Matrix* 205
 - 8.6.6 *Function MPI_Cart_rank* 205
 - 8.6.7 *Function MPI_Cart_coords* 207
 - 8.6.8 *Function MPI_Comm_split* 207
 - 8.6.9 *Benchmarking* 208
 - 8.7 **Summary** 210
 - 8.8 **Key Terms** 211
 - 8.9 **Bibliographic Notes** 211
 - 8.10 **Exercises** 211
- CHAPTER 9**
- Document Classification** 216
- 9.1 **Introduction** 216
 - 9.2 **Parallel Algorithm Design** 217
 - 9.2.1 *Partitioning and Communication* 217
 - 9.2.2 *Agglomeration and Mapping* 217
 - 9.2.3 *Manager/Worker Paradigm* 218
 - 9.2.4 *Manager Process* 219
 - 9.2.5 *Function MPI_Abort* 220
 - 9.2.6 *Worker Process* 221
 - 9.2.7 *Creating a Workers-only Communicator* 223
 - 9.3 **Nonblocking Communications** 223
 - 9.3.1 *Manager's Communication* 224
 - 9.3.2 *Function MPI_Irecv* 224
 - 9.3.3 *Function MPI_wait* 225
 - 9.3.4 *Workers' Communications* 225
 - 9.3.5 *Function MPI_Isend* 225
 - 9.3.6 *Function MPI_Probe* 225
 - 9.3.7 *Function MPI_Get_count* 226
 - 9.4 **Documenting the Parallel Program** 226
 - 9.5 **Enhancements** 232
- 9.5.1 *Assigning Groups of Documents* 232
 - 9.5.2 *Pipelining* 232
 - 9.5.3 *Function MPI_Testsome* 234
- 9.6 **Summary** 235
 - 9.7 **Key Terms** 236
 - 9.8 **Bibliographic Notes** 236
 - 9.9 **Exercises** 236
- CHAPTER 10**
- Monte Carlo Methods** 239
- 10.1 **Introduction** 239
 - 10.1.1 *Why Monte Carlo Works* 240
 - 10.1.2 *Monte Carlo and Parallel Computing* 243
 - 10.2 **Sequential Random Number Generators** 243
 - 10.2.1 *Linear Congruential* 244
 - 10.2.2 *Lagged Fibonacci* 245
 - 10.3 **Parallel Random Number Generators** 245
 - 10.3.1 *Manager-Worker Method* 246
 - 10.3.2 *Leapfrog Method* 246
 - 10.3.3 *Sequence Splitting* 247
 - 10.3.4 *Parameterization* 248
 - 10.4 **Other Random Number Distributions** 248
 - 10.4.1 *Inverse Cumulative Distribution Function Transformation* 249
 - 10.4.2 *Box-Muller Transformation* 250
 - 10.4.3 *The Rejection Method* 251
 - 10.5 **Case Studies** 253
 - 10.5.1 *Neutron Transport* 253
 - 10.5.2 *Temperature at a Point Inside a 2-D Plate* 255
 - 10.5.3 *Two-Dimensional Ising Model* 257
 - 10.5.4 *Room Assignment Problem* 259
 - 10.5.5 *Parking Garage* 262
 - 10.5.6 *Traffic Circle* 264
 - 10.6 **Summary** 268
 - 10.7 **Key Terms** 269
 - 10.8 **Bibliographic Notes** 269
 - 10.9 **Exercises** 270

CHAPTER 11

Matrix Multiplication 273

- 11.1 Introduction 273
- 11.2 Sequential Matrix Multiplication 274
 - 11.2.1 *Iterative, Row-Oriented Algorithm* 274
 - 11.2.2 *Recursive, Block-Oriented Algorithm* 275
- 11.3 Rowwise Block-Striped Parallel Algorithm 277
 - 11.3.1 *Identifying Primitive Tasks* 277
 - 11.3.2 *Agglomeration* 278
 - 11.3.3 *Communication and Further Agglomeration* 279
 - 11.3.4 *Analysis* 279
- 11.4 Cannon's Algorithm 281
 - 11.4.1 *Agglomeration* 281
 - 11.4.2 *Communication* 283
 - 11.4.3 *Analysis* 284
- 11.5 Summary 286
- 11.6 Key Terms 287
- 11.7 Bibliographic Notes 287
- 11.8 Exercises 287

CHAPTER 12

Solving Linear Systems 290

- 12.1 Introduction 290
- 12.2 Terminology 291
- 12.3 Back Substitution 292
 - 12.3.1 *Sequential Algorithm* 292
 - 12.3.2 *Row-Oriented Parallel Algorithm* 293
 - 12.3.3 *Column-Oriented Parallel Algorithm* 295
 - 12.3.4 *Comparison* 295
- 12.4 Gaussian Elimination 296
 - 12.4.1 *Sequential Algorithm* 296
 - 12.4.2 *Parallel Algorithms* 298
 - 12.4.3 *Row-Oriented Algorithm* 299
 - 12.4.4 *Column-Oriented Algorithm* 303

12.4.5 *Comparison* 30312.4.6 *Pipelined, Row-Oriented Algorithm* 304

- 12.5 Iterative Methods 306
- 12.6 The Conjugate Gradient Method 309
 - 12.6.1 *Sequential Algorithm* 309
 - 12.6.2 *Parallel Implementation* 310
- 12.7 Summary 313
- 12.8 Key Terms 314
- 12.9 Bibliographic Notes 314
- 12.10 Exercises 314

CHAPTER 13

Finite Difference Methods 318

- 13.1 Introduction 318
- 13.2 Partial Differential Equations 320
 - 13.2.1 *Categorizing PDEs* 320
 - 13.2.2 *Difference Quotients* 321
- 13.3 Vibrating String 322
 - 13.3.1 *Deriving Equations* 322
 - 13.3.2 *Deriving the Sequential Program* 323
 - 13.3.3 *Parallel Program Design* 324
 - 13.3.4 *Isoefficiency Analysis* 327
 - 13.3.5 *Replicating Computations* 327
- 13.4 Steady-State Heat Distribution 329
 - 13.4.1 *Deriving Equations* 329
 - 13.4.2 *Deriving the Sequential Program* 330
 - 13.4.3 *Parallel Program Design* 332
 - 13.4.4 *Isoefficiency Analysis* 332
 - 13.4.5 *Implementation Details* 334
- 13.5 Summary 334
- 13.6 Key Terms 335
- 13.7 Bibliographic Notes 335
- 13.8 Exercises 335

CHAPTER 14

Sorting 338

- 14.1 Introduction 338
- 14.2 Quicksort 339

- 14.3 A Parallel Quicksort Algorithm 340
 - 14.3.1 Definition of Sorted 340
 - 14.3.2 Algorithm Development 341
 - 14.3.3 Analysis 341
- 14.4 Hyperquicksort 343
 - 14.4.1 Algorithm Description 343
 - 14.4.2 Isoefficiency Analysis 345
- 14.5 Parallel Sorting by Regular Sampling 346
 - 14.5.1 Algorithm Description 346
 - 14.5.2 Isoefficiency Analysis 347
- 14.6 Summary 349
- 14.7 Key Terms 349
- 14.8 Bibliographic Notes 350
- 14.9 Exercises 350

CHAPTER 15

The Fast Fourier Transform 353

- 15.1 Introduction 353
- 15.2 Fourier Analysis 353
- 15.3 The Discrete Fourier Transform 355
 - 15.3.1 Inverse Discrete Fourier Transform 357
 - 15.3.2 Sample Application: Polynomial Multiplication 357
- 15.4 The Fast Fourier Transform 360
- 15.5 Parallel Program Design 363
 - 15.5.1 Partitioning and Communication 363
 - 15.5.2 Agglomeration and Mapping 365
 - 15.5.3 Isoefficiency Analysis 365
- 15.6 Summary 367
- 15.7 Key Terms 367
- 15.8 Bibliographic Notes 367
- 15.9 Exercises 367

CHAPTER 16

Combinatorial Search 369

- 16.1 Introduction 369
- 16.2 Divide and Conquer 370

- 16.3 Backtrack Search 371
 - 16.3.1 Example 371
 - 16.3.2 Time and Space Complexity 374
- 16.4 Parallel Backtrack Search 374
- 16.5 Distributed Termination Detection 377
- 16.6 Branch and Bound 380
 - 16.6.1 Example 380
 - 16.6.2 Sequential Algorithm 382
 - 16.6.3 Analysis 385
- 16.7 Parallel Branch and Bound 385
 - 16.7.1 Storing and Sharing Unexamined Subproblems 386
 - 16.7.2 Efficiency 387
 - 16.7.3 Halting Conditions 387
- 16.8 Searching Game Trees 388
 - 16.8.1 Minimax Algorithm 388
 - 16.8.2 Alpha-Beta Pruning 392
 - 16.8.3 Enhancements to Alpha-Beta Pruning 395
- 16.9 Parallel Alpha-Beta Search 395
 - 16.9.1 Parallel Aspiration Search 396
 - 16.9.2 Parallel Subtree Evaluation 396
 - 16.9.3 Distributed Tree Search 397
- 16.10 Summary 399
- 16.11 Key Terms 400
- 16.12 Bibliographic Notes 400
- 16.13 Exercises 401

CHAPTER 17

Shared-Memory Programming 404

- 17.1 Introduction 404
- 17.2 The Shared-Memory Model 405
- 17.3 Parallel for Loops 407
 - 17.3.1 `parallel for` Pragma 408
 - 17.3.2 Function `omp_get_num_procs` 410
 - 17.3.3 Function `omp_set_num_threads` 410
- 17.4 Declaring Private Variables 410
 - 17.4.1 `private` Clause 411

- 17.4.2 *firstprivate* Clause 412
 - 17.4.3 *lastprivate* Clause 412
 - 17.5 Critical Sections 413
 - 17.5.1 *critical* Pragma 415
 - 17.6 Reductions 415
 - 17.7 Performance Improvements 417
 - 17.7.1 *Inverting Loops* 417
 - 17.7.2 *Conditionally Executing Loops* 418
 - 17.7.3 *Scheduling Loops* 419
 - 17.8 More General Data Parallelism 421
 - 17.8.1 *parallel* Pragma 422
 - 17.8.2 Function *omp_get_thread_num* 423
 - 17.8.3 Function *omp_get_num_threads* 425
 - 17.8.4 *for* Pragma 425
 - 17.8.5 *single* Pragma 427
 - 17.8.6 *nowait* Clause 427
 - 17.9 Functional Parallelism 428
 - 17.9.1 *parallel sections* Pragma 429
 - 17.9.2 *section* Pragma 429
 - 17.9.3 *sections* Pragma 429
 - 17.10 Summary 430
 - 17.11 Key Terms 432
 - 17.12 Bibliographic Notes 432
 - 17.13 Exercises 433
- CHAPTER 18**
- Combining MPI and OpenMP** 436
- 18.1 Introduction 436
 - 18.2 Conjugate Gradient Method 438
 - 18.2.1 *MPI Program* 438
 - 18.2.2 *Functional Profiling* 442
 - 18.2.3 *Parallelizing Function*
matrix_vector_product 442
 - 18.2.4 *Benchmarking* 443
 - 18.3 Jacobi Method 444
 - 18.3.1 *Profiling MPI Program* 444
 - 18.3.2 *Parallelizing Function*
find_steady_state 444
 - 18.3.3 *Benchmarking* 446
- 18.4 Summary 448
- 18.5 Exercises 448
- APPENDIX A**
- MPI Functions** 450
- APPENDIX B**
- Utility Functions** 485
- B.1 Header File *MyMPI.h* 485
 - B.2 Source File *MyMPI.c* 486
- APPENDIX C**
- Debugging MPI Programs** 505
- C.1 Introduction 505
 - C.2 Typical Bugs in MPI Programs 505
 - C.2.1 *Bugs Resulting in Deadlock* 505
 - C.2.2 *Bugs Resulting in Incorrect Results* 506
 - C.2.3 *Advantages of Collective Communications* 507
 - C.3 Practical Debugging Strategies 507
- APPENDIX D**
- Review of Complex Numbers** 509
- APPENDIX E**
- OpenMP Functions** 513
- Bibliography** 515
- Author Index** 520
- Subject Index** 522