

SOURCEBOOK OF PARALLEL COMPUTING

JACK DONGARRA

University of Tennessee

IAN FOSTER

Argonne National Laboratory

GEOFFREY FOX

Indiana University

WILLIAM GROPP

Argonne National Laboratory

KEN KENNEDY

Rice University

LINDA TORCZON

Rice University

ANDY WHITE

Los Alamos National Laboratory



MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ELSEVIER SCIENCE

AMSTERDAM BOSTON LONDON NEW YORK
OXFORD PARIS SAN DIEGO SAN FRANCISCO
SINGAPORE SYDNEY TOKYO

Senior Editor Denise Penrose
Publishing Services Manager Edward Wade
Production Editor Howard Severson
Editorial Coordinator Emilia Thiuri
Cover Design Frances Baca
Text Design Detta Penna
Illustration Dartmouth Publishing, Inc.
Composition Windfall Software, using ZzT_EX
Copyeditor Barbara Kohl
Proofreader Carol Leyba
Indexer Steve Rath
Printer The Maple-Vail Book Manufacturing Group

Cover credit: Paul Klee, Green church steeple at center, 1917. © Nimatallah/Art Resource, NY.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Morgan Kaufmann Publishers
An imprint of Elsevier Science
340 Pine Street, Sixth Floor
San Francisco, CA 94104-3205
www.mkp.com

© 2003 by Elsevier Science (USA)
All rights reserved.
Printed in the United States of America

07 06 05 04 03 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, or otherwise—without the prior written permission of the publisher.

Library of Congress Control Number: 2002107244
ISBN: 1-55860-871-0

This book is printed on acid-free paper.

Part I Parallelism	1
Introduction	3
Parallel Computing Hardware	4
What Have We Learned from Applications?	8
Software and Algorithms	11
Toward a Science of Parallel Computation	13
Parallel Computer Architectures	15
Uniprocessor Architecture	16
<i>The CPU</i>	17
<i>Memory</i>	21
<i>I/O and Networking</i>	25
<i>Design Tradeoffs</i>	26
Parallel Architectures	26
<i>Memory Parallelism</i>	26
<i>Interconnects</i>	33
<i>CPU Parallelism</i>	35
<i>I/O and Networking for Parallel Processors</i>	38
<i>Support for Programming Models</i>	39
<i>Parallel Architecture Design Tradeoffs</i>	39
Future Directions for Parallel Architectures.....	40
Conclusion	41
Parallel Programming Considerations	43
Architectural Considerations	45
<i>Shared Memory</i>	45
<i>Distributed Memory</i>	46
<i>Hybrid Systems</i>	47
<i>Memory Hierarchy</i>	47
Decomposing Programs for Parallelism	49
<i>Identification of Parallelism</i>	49
<i>Decomposition Strategy</i>	50
<i>Programming Models</i>	51
<i>Implementation Styles</i>	51
<i>A Simple Example</i>	54
Enhancing Parallel Performance.....	56
<i>Scalability and Load Balance</i>	57
<i>Pipeline Parallelism</i>	59

<i>Regular versus Irregular Problems</i>	60
Memory-Hierarchy Management	63
<i>Uniprocessor Memory-Hierarchy Management</i> ...	63
<i>Multiprocessor Memory Hierarchies</i>	65
Parallel Debugging	66
Performance Analysis and Tuning	67
Parallel Input/Output	69
Conclusion	70
Further Reading	70
Part II Applications	73
General Application Issues	75
Application Characteristics in a Simple Example	75
Communication Structure in Jacobi's Method for Poisson's Equation	79
Communication Overhead for More General Update Stencils	82
Applications as Basic Complex Systems	84
Time-Stepped and Event-Driven Simulations	87
Temporal Structure of Applications	88
Summary of Parallelization of Basic Complex Systems	89
Meta-Problems	90
Conclusion	91
Parallel Computing in Computational Fluid Dynamics	93
Introduction to Computational Fluid Dynamics.....	94
<i>Basic Equations of Fluid Dynamics</i>	94
<i>Physical Regimes and Dimensionless Variables</i>	95
<i>The Role of High-Performance Computing</i>	97
Incompressible Flows	98
<i>Semi-discrete Formulation</i>	99
<i>Spectral Element Methods</i>	100
<i>Basic Operations</i>	108
<i>Global Matrix Operations</i>	111
<i>Data Structures</i>	113

<i>Solution Techniques</i>	116
<i>Adaptive Mesh Re.nement</i>	121
<i>Implementation for Parallel Architectures</i>	126
<i>An Example The Cylinder Wake</i>	129
Compressible Flows	132
<i>Governing Equations of Motion</i>	133
<i>Numerical Methods for Hyperbolic</i>	
<i>Conservation Laws</i>	134
<i>An Application: The Richtmyer Meshkov</i>	
<i>Instability</i>	137
<i>Adaptive Mesh Re.nement</i>	138
Conclusion	144
Parallel Computing in Environment and	
Energy	145
Subsurface-Flow Modeling.....	146
<i>IPARS Motivation</i>	147
<i>IPARS Description</i>	148
IPARS and Grid Computing by NetSolve	152
<i>Integrating IPARS into NetSolve</i>	153
<i>Client-Side Web-Browser Interface</i>	154
Tracking and Interactive Simulation in IPARS	155
<i>An Interactive Computational Collaboration:</i>	
<i>DISCOVER</i>	157
<i>Integrating IPARS with DISCOVER</i>	158
Surface-Water Simulation	159
<i>A Water Quality Model: CE-QUAL-ICM</i>	159
<i>A Parallel Water-Quality Model:</i>	
<i>PCE-QUAL-ICM</i>	160
<i>Parallel Algorithm</i>	161
A Coupled Simulation of Flow and Transport with	
ADR.....	162
<i>The Active Data Repository</i>	163
<i>Implementation</i>	164
Conclusion	165
Parallel Computational Chemistry: An	
Overview of NWChem	167
Molecular Quantum Chemistry.....	168

The NWChem Architecture	171
NWChem Parallel Computing Support	174
<i>The Global Array Toolkit</i>	174
<i>Parallel Linear Algebra: PeIGS</i>	176
NWChem Chemistry Modules	178
<i>Hartree Fock Self-Consistent Field</i>	180
<i>Resolution of the Identity Second-Order, Many-Body Perturbation Theory</i>	182
NWChem's Place in the Computational Chemistry Community	186
A Larger Perspective: Common Features of Computational Chemistry Algorithms	188
Conclusion	192
Application Overviews	195
Numerical (General) Relativity	195
<i>Current Situation</i>	197
Numerical Simulations in Lattice Quantum Chromodynamics	199
<i>Lattice QCD Simulation Setup</i>	201
<i>Computational Requirements</i>	204
<i>Implementation Considerations</i>	205
<i>Recent Developments and Future Prospects</i>	206
Ocean Modeling	207
<i>Surface-Pressure Formulation of the Barotropic Mode</i>	208
<i>Free-Surface Formulation</i>	209
<i>Pressure Averaging</i>	210
<i>Latitudinal Scaling of Horizontal Diffusion</i>	210
<i>Code Designed for Parallel Computers</i>	211
<i>General Orthogonal Coordinates and the Displaced-Pole Grid</i>	211
<i>High-Resolution Simulations Enabled by POP</i>	211
Simulations of Earthquakes	212
<i>Typical Computational Problems</i>	215
<i>Computational Resource Requirements</i>	218
Cosmological Structure Formation	219
<i>The Problem to be Solved</i>	219

<i>Computational Issues</i>	219
<i>Parallel Unigrid Code: Kronos</i>	220
<i>Parallel AMR Code: Enzo</i>	222
<i>Parallelization of Enzo</i>	223
<i>Performance</i>	224
<i>Future Work</i>	227
Computational Electromagnetics	227
<i>Asymptotic Methods</i>	229
<i>Frequency-Domain Methods</i>	229
<i>Time-Domain Methods</i>	230
<i>Hybrid Methods</i>	231
<i>State of the Art</i>	232
Parallel Algorithms in Data Mining	232
<i>Parallel Algorithms for Discovering</i> <i>Associations</i>	233
<i>Parallel Algorithms for Induction of</i> <i>Decision-Tree Classifiers</i>	236
<i>State of the Art</i>	242
High-Performance Computing in Signal and Image Processing	243
<i>Examples of HPC Use in Signal and Image</i> <i>Processing</i>	244
<i>State of the Art</i>	249
Deterministic Monte Carlo Methods and Parallelism	249
<i>Motivation for Using Quasi-Random Numbers</i>	250
<i>Methods of Quasi-Random Number</i> <i>Generation</i>	252
<i>A Fundamental Problem with Quasi-Random</i> <i>Numbers</i>	255
<i>State-of-the-Art Quasi-Random Number</i> <i>Generators</i>	255
<i>A Parallel Quasi Monte Carlo Application</i>	256
<i>State of the Art</i>	258
Quasi Real Time Microtomography Experiments at Photon Sources	258

<i>The Computational Processing Pipeline</i>	
<i>Framework</i>	259
<i>Scientific Challenges</i>	260
<i>Benefits of Real-Time X-Ray Microtomography</i>	
<i>Experiments</i>	263
<i>Future Work</i>	265
WebHLA-Based Meta-Computing Environment	
for Forces Modeling and Simulation.....	265
<i>DoD Modeling and Simulation</i>	266
<i>Forces Modeling and Simulation</i>	266
<i>High-Level Architecture</i>	267
<i>WebHLA</i>	268
<i>Example WebHLA Application:</i>	
<i>Parallel/Meta-Computing CMS</i>	272
<i>Next Steps</i>	278
Computational Structure of Applications.....	280
<i>Applications from This Book</i>	280
<i>Applications from</i>	284
<i>[573]</i>	284
<i>Applications from</i>	287
Conclusion.....	290

Part III Software Technologies..... 291

Software Technologies.....	293
Selecting a Parallel Program Technology.....	294
<i>Parallel Programming Models</i>	295
<i>Parallel Programming Technologies</i>	297
<i>Decision Rules</i>	308
Achieving Correct and Efficient Execution.....	308
<i>Dealing with Nondeterminism</i>	309
<i>Performance Modeling</i>	309
Conclusion.....	310
<i>Clusters and DSM</i>	310
<i>Grids</i>	310
<i>Ultra-Scale Computers</i>	311
<i>Programming Productivity</i>	311
<i>Further Reading</i>	312

Message Passing and Threads	313
Message-Passing Programming Model	314
<i>The Message Passing Interface Standard</i>	315
<i>Parallel Virtual Machine</i>	322
<i>Extensions to the Message Passing Interface</i>	322
<i>State of the Art</i>	323
Multithreaded Programming.....	323
<i>POSIX Threads</i>	325
<i>OpenMP</i>	327
Conclusion	329
Parallel I/O	331
Parallel I/O Infrastructure	333
<i>Basic Disk Architecture</i>	333
<i>Parallel I/O Architecture</i>	334
<i>File Systems</i>	335
<i>The API Problem</i>	336
<i>I/O Libraries</i>	338
<i>Language-Based Parallel I/O</i>	339
Overview of MPI-IO	339
<i>Simple MPI-IO Example</i>	340
<i>Main Features of MPI-IO</i>	342
<i>Noncontiguous Accesses in MPI-IO</i>	343
<i>MPI-IO Implementations</i>	343
Parallel I/O Optimizations.....	344
<i>Data Sieving</i>	344
<i>Collective I/O</i>	346
<i>Hints and Adaptive File-System Policies</i>	347
How Can Users Achieve High I/O Performance?	348
<i>General Guidelines</i>	348
<i>Achieving High Performance with MPI-IO</i>	349
Conclusion	355
Languages and Compilers	357
Automatic Parallelization.....	359
Data-Parallel Programming in High Performance Fortran.....	361
Shared-Memory Parallel Programming in OpenMP	366

Single-Program, Multiple-Data Programming in Co- Array Fortran	371
Supporting Technologies	377
<i>Programming Support Tools</i>	377
<i>Libraries</i>	378
Future Trends.....	378
Conclusion	379
Further Reading	380
Parallel Object-Oriented Libraries	383
Object-Oriented Parallel Libraries	384
<i>Abstraction</i>	384
<i>Parallelism</i>	386
<i>Encapsulation</i>	387
<i>Generic Programming</i>	388
<i>A POOMA Example</i>	389
Object-Oriented Parallel Programming in Java.....	391
Multithreaded Computation in C++	396
<i>The Execution Model</i>	397
<i>Thread and Synchronization</i>	398
Remote Function Calls, Global Pointers, and Java RMI	401
Component-Based Software Design	403
<i>The DOE Common Component Architecture</i>	404
Conclusion	406
Problem-Solving Environments.....	409
NetSolve: Network-Enabled Solvers	411
<i>The NetSolve Philosophy</i>	412
<i>NetSolve Infrastructure</i>	412
<i>Some Applications of NetSolve</i>	416
<i>Current Developments and Future Research</i>	417
WebFlow-Object Web Computing	418
<i>WebFlow Architecture</i>	421
<i>WebFlow Applications</i>	424
WebPDELab	429
<i>The WebPDELab Server</i>	429
<i>WebPDELab Security Issues</i>	437
<i>WebPDELab Features and Issues</i>	438

Other Grid-Computing Environments	440
<i>Meta-Computing Systems</i>	440
<i>Seamless Access and Application Integration</i>	441
Conclusion	442
Tools for Performance Tuning and Debugging ..	443
Correctness and Performance Monitoring Basics	444
<i>Profiling and Program-Counter Sampling</i>	445
<i>Event Counting</i>	446
<i>Interval Timing</i>	448
<i>Event Tracing</i>	448
<i>Control Breakpoints</i>	450
Measurement and Debugging Implementation	
Challenges	451
<i>Clocks and Timing</i>	451
<i>Event Orders and Time</i>	451
Deep Compiler Integration	453
<i>A Motivating Example</i>	453
<i>Performance Modeling and Prediction</i>	455
Software Tool Interfaces and Usability	456
<i>Tool Scalability</i>	457
<i>User Expectations and Recommendations</i>	457
Software Tool Examples	459
<i>Jumpshot Event Visualization</i>	459
<i>SvPablo Source Code Correlation</i>	460
<i>Thinking Machines Prism</i>	462
<i>Etnus TotalView</i>	465
Challenges and Open Problems	466
Conclusion	466
Further Reading	467
The 2-D Poisson Problem	469
The Mathematical Model	469
A Simple Algorithm.....	470
Parallel Solution of Poisson s Equation	470
<i>Message Passing and the Distributed-Memory</i> <i>Model</i>	470
<i>The Single Name-Space, Distributed-Memory</i> <i>Model</i>	472

<i>The Shared-Memory Model</i>	475
<i>Comments</i>	476
Adding Global Operations	477
<i>Collective Operations in MPI</i>	477
<i>Reductions in HPF</i>	478
<i>Reductions in OpenMP</i>	478
<i>Conclusion</i>	480

Part IV Enabling Technologies and Algorithms 481

Reusable Software and Algorithms	483
Templates: Design Patterns for Parallel Software ...	483
Communicators and Data Structure Neutrality.....	484
Standard Libraries and Components	485
<i>Load Balancing and Grid Generation</i>	485
<i>Mesh Generation</i>	486
<i>Software for Scalable Solution of PDEs</i>	486
<i>Parallel Continuous Optimization</i>	486
Automatic Differentiation	486
Templates and Numerical Linear Algebra	487
Conclusion	489
Graph Partitioning for High-Performance Scientific Simulations	491
Modeling Mesh-Based Computations as Graphs.....	493
<i>Computing a</i>	495
<i>Way Partitioning via Recursive Bisection</i>	495
Static Graph-Partitioning Techniques	495
<i>Geometric Techniques</i>	496
<i>Combinatorial Techniques</i>	501
<i>Spectral Methods</i>	506
<i>Multilevel Schemes</i>	509
<i>Combined Schemes</i>	513
<i>Qualitative Comparison of Graph Partitioning Schemes</i>	513
Load Balancing of Adaptive Computations	516
<i>Scratch-Remap Repartitioners</i>	518
<i>Diffusion-Based Repartitioners</i>	522

Parallel Graph Partitioning	525
Multiconstraint, Multiobjective Graph Partitioning	526
<i>A Generalized Formulation for Graph Partitioning</i>	531
Conclusion	538
<i>Limitations of the Graph-Partitioning Problem Formulation</i>	538
<i>Other Application Modeling Limitations</i>	539
<i>Architecture Modeling Limitations</i>	539
<i>Functionality of Available Graph Partitioning Packages</i>	540
Mesh Generation	543
Mesh-Generation Strategies and Techniques.....	544
<i>Cartesian Meshes</i>	544
<i>Structured Meshes</i>	544
<i>Unstructured Meshes</i>	547
<i>Hybrid/Generalized Meshes</i>	549
<i>Meshless Methods</i>	550
Mesh-Generation Process and Geometry Preparation.....	550
Adaptive Mesh Generation.....	552
<i>Structured Mesh Adaptation</i>	552
<i>Generalized Mesh Adaptation</i>	555
Parallel Mesh Generation.....	560
Mesh Software	561
Mesh Configurations	564
Mesh Web Sites	567
The Pacing Obstacle: Geometry/Mesh Generation	569
Desiderata	571
Conclusion	572
Templates and Numerical Linear Algebra.....	575
Dense Linear Algebra Algorithms	576
<i>Loop Rearranging</i>	577
<i>Uses of LU Factorization in Science and Engineering</i>	577
<i>Block Algorithms and Their Derivation</i>	578

The Influence of Computer Architecture on Performance.....	580
<i>Discussion of Architectural Features</i>	580
<i>Target Architectures</i>	582
Dense Linear Algebra Libraries.....	583
<i>The BLAS as the Key to Portability</i>	583
<i>Overview of Dense Linear Algebra Libraries</i>	587
<i>Available Software</i>	589
Sparse Linear Algebra Methods.....	590
<i>Origin of Sparse Linear Systems</i>	590
<i>Basic Elements in Sparse Linear Algebra Methods</i>	591
Direct Solution Methods	591
<i>Matrix Orderings</i>	592
<i>Use of Level-3 BLAS Kernels</i>	594
<i>Available Software</i>	594
Iterative Solution Methods.....	596
<i>Stationary Iterative Methods</i>	596
<i>Krylov Space Methods</i>	596
<i>Preconditioners</i>	598
<i>Libraries and Standards in Sparse Methods</i>	600
<i>Available Software</i>	602
Sparse Eigenvalue Problems.....	603
<i>Algorithms and Software for Large Eigenvalue Problems</i>	603
=	603
<i>Additional Available Software and Future Directions</i>	618
Conclusion	619
<i>Future Research Directions in Dense Algorithms</i>	619
Software for the Scalable Solution of Partial Differential Equations	621
PDE Background.....	622
Challenges in Parallel PDE Computations.....	623
<i>Software Complexity</i>	624
<i>Data Distribution and Access</i>	624

<i>Portability, Algorithms, and Data Redistribution ..</i>	626
Parallel Solution Strategies	627
PETSc Approach to Parallel Software for PDEs	628
<i>Sample Applications</i>	629
<i>Mathematical Formulation</i>	632
<i>Composability and Interoperability</i>	639
<i>Performance Issues.....</i>	640
Software for PDEs.....	645
Conclusion	647
Parallel Continuous Optimization	649
Local Optimization.....	651
Global Optimization	653
<i>Protein Folding</i>	655
<i>Cluster Simulation</i>	656
<i>Distance Geometry.....</i>	656
<i>Stochastic Global Optimization.....</i>	657
<i>Effective-Energy Simulated Annealing</i>	657
<i>Global Continuation.....</i>	658
Direct Search Methods.....	659
<i>The Surrogate Management Framework.....</i>	661
<i>Asynchronous Parallel Search</i>	662
Optimization of Linked Subsystems	663
Variable and Constraint Distribution.....	666
<i>Variable Distribution</i>	667
<i>Constraint Distribution</i>	668
Conclusion	669
Path Following in Scientific Computing and Its Implementation in AUTO.....	671
Local Continuation	673
Global Continuation and Degree Theory.....	675
Folds and Bifurcations.....	677
Practical Path Following.....	679
Branch Switching at Bifurcations.....	683
Computational Examples: AUTO	686
<i>Bursting Oscillations</i>	688
<i>Some Navier Stokes Flows</i>	690
<i>Kolmogorov Flows</i>	691

Parallel AUTO	694
<i>Parallel Implementation</i>	695
Conclusion	699
Automatic Differentiation	701
Overview of Automatic Differentiation	703
<i>How Automatic Differentiation Works</i>	704
<i>When Automatic Differentiation Works</i>	706
Automatic-Differentiation Implementation Techniques	707
<i>AD via Operator Overloading</i>	708
<i>AD via Source-to-Source Transformation</i>	708
Automatic-Differentiation Software	709
<i>ADOL-C</i>	709
<i>Adifor 3.0</i>	710
Automatic Differentiation of Message-Passing Parallel Codes	711
<i>Activity Analysis</i>	711
<i>Differentiation of Communication Operations</i>	711
<i>Differentiation of Reduction Operations</i>	714
Advanced Use of Automatic Differentiation	714
<i>Computing Sparse Jacobian Matrices with Known Sparsity</i>	714
<i>Computing Sparse Jacobian Matrices with Unknown Sparsity</i>	716
<i>Strip-Mining of Derivative Computations</i>	716
<i>Exploiting Coarse-Grained Chain Rule Associativity</i>	717
<i>Checkpointing for the Reverse Mode</i>	717
Conclusion	719

V Conclusion	V
Wrap-Up and Signposts to the Future	723
Computational Resources	723
Applications	724
Software	725
Templates, Algorithms, and Technologies	727
Signposts	727

REFERENCES	729
INDEX	791
ABOUT THE AUTHORS.....	833