
Contents at a Glance

Preface	xxi
I Programming Environments and Development Tools .	1
1 Parallel Programming Models and Paradigms	4
2 Parallel Programming Languages and Environments	28
3 MPI and PVM Programming	48
4 Linking Message-Passing Environments	87
5 Active Objects	101
6 Using Scoped Behavior to Optimize Data Sharing Idioms	113
7 Component-Based Development Approach	131
8 Hypercomputing with LiPS	153
9 An Efficient Tuple Space Programming Environment	175
10 Debugging Parallelized Code	197
11 WebOS: Operating System Services for Wide-Area Applications	225
II Java for High Performance Computing	247
12 Distributed-Object Computing	249
13 Java and Different Flavors of Parallel Programming Models	274
14 The HPspmd Model and its Java Binding	291
15 Web-Based Parallel Computing with Java	310
III Algorithms and Applications	327
16 Object-Oriented Implementation of Parallel Genetic Algorithms	331
17 Application-Specific Load Balancing on Heterogeneous Systems	350
18 Time Management in Parallel Simulation	375
19 Hardware System Simulation	395

20	Real-Time Resource Management Middleware: Open Systems and Applications	418
21	Data Placement in Shared-Nothing Database Systems	440
22	Parallel Inference with Very Large Knowledge Bases	455
23	MaRT: Lazy Evaluation for Parallel Ray Tracing	486
24	Fast Content-Based Image Retrieval	506
25	Climate Ocean Modeling	526
26	Computational Electromagnetics	540
27	CFD Simulation: A Case Study in Software Engineering	558
28	Quantum Reactive Scattering Calculations	580
29	Biomedical Applications Modeling	604
A	Glossary	627
	Index	659

Contents

Preface	xxi
I Programming Environments and Development Tools .	1
1 Parallel Programming Models and Paradigms	4
1.1 Introduction.....	4
1.2 A Cluster Computer and its Architecture	5
1.3 Parallel Applications and Their Development	7
1.3.1 Strategies for Developing Parallel Applications	9
1.4 Code Granularity and Levels of Parallelism	10
1.5 Parallel Programming Models and Tools	11
1.5.1 Parallelizing Compilers.....	11
1.5.2 Parallel Languages	12
1.5.3 High Performance Fortran	12
1.5.4 Message Passing	13
1.5.5 Virtual Shared Memory.....	13
1.5.6 Parallel Object-Oriented Programming.....	14
1.5.7 Programming Skeletons.....	14
1.6 Methodical Design of Parallel Algorithms	15
1.6.1 Partitioning	15
1.6.2 Communication.....	16
1.6.3 Agglomeration	16
1.6.4 Mapping.....	16
1.7 Parallel Programming Paradigms	16
1.7.1 Choice of Paradigms	17
1.7.2 Task-Farming (or Master/Slave)	19
1.7.3 Single-Program Multiple-Data (SPMD)	19
1.7.4 Data Pipelining.....	21
1.7.5 Divide and Conquer	21
	vii

1.7.6	Speculative Parallelism	23
1.7.7	Hybrid Models	23
1.8	Programming Skeletons and Templates	23
1.8.1	Programmability	25
1.8.2	Reusability	25
1.8.3	Portability	25
1.8.4	Efficiency	25
1.9	Conclusions	25
1.10	Bibliography	26
2	Parallel Programming Languages and Environments	28
2.1	Introduction	28
2.2	Early Mechanisms	30
2.2.1	Semaphores	30
2.2.2	Conditional Critical Regions	31
2.2.3	Monitors	31
2.2.4	Sockets	32
2.2.5	Remote Procedure Calls	32
2.2.6	Rendezvous	32
2.3	Shared Memory Environments	33
2.3.1	Pure Shared Memory Environments	34
2.3.2	Virtual Shared Memory Environments	35
2.4	Distributed Memory Environments	39
2.4.1	Ada	39
2.4.2	Message Passing Interface	39
2.4.3	Parallel Virtual Machine	40
2.4.4	Distributed Computing Environment	41
2.4.5	Distributed Java	41
2.5	Parallel Declarative Environments	44
2.5.1	Parallel Logic Languages	44
2.5.2	Parallel Functional Languages	45
2.6	Summary	46
2.7	Bibliography	46
3	MPI and PVM Programming	48
3.1	Introduction	48
3.2	Comparison of MPI and PVM	49
3.3	The All Pairs Shortest Path Problem	51
3.3.1	Description of the Problem	51
3.3.2	Dijkstra's Algorithm	51
3.3.3	Floyd's Algorithm	52
3.3.4	Parallel Algorithms	53
3.4	The MPI Programming Environment	56

3.4.1	Communication Models in MPI	57
3.4.2	Creating an MPI Program.....	58
3.4.3	Running an MPI Program.....	60
3.4.4	PDIJK_MPI.....	61
3.4.5	PFLOYD_MPI.....	64
3.4.6	Tools for MPI Programs	68
3.5	The PVM Programming Environment	68
3.5.1	Communication Models in PVM	69
3.5.2	Creating a PVM Program	70
3.5.3	Running a PVM Program	73
3.5.4	PDIJK_PVM.....	73
3.5.5	PFLOYD_PVM.....	77
3.5.6	Tools for PVM Programs	82
3.6	Porting Hints	82
3.6.1	Initial Environment Setup	83
3.6.2	Parallel Tasks Setup	83
3.6.3	Group Management	84
3.6.4	Intertask Communication.....	84
3.6.5	Utility Functions	85
3.7	Summary	85
3.8	Bibliography	86
4	Linking Message-Passing Environments	87
4.1	Interoperability Between Message-Passing Interfaces	87
4.1.1	Message Passing Between Programming Environments	88
4.1.2	Access to Resource Management Systems.....	88
4.2	An Overview of the PLUS Library.....	88
4.2.1	Process Creation and Management	91
4.3	System Architecture.....	93
4.3.1	Library	93
4.3.2	Daemons	94
4.4	Adding New Message-Passing Environments	96
4.5	Performance Results.....	97
4.6	Related Work	97
4.7	Summary	98
4.8	Bibliography	99
5	Active Objects	101
5.1	Objects in Cluster-Based Parallel Systems	101
5.2	Active Versus Passive Objects.....	102
5.3	Objects and Atomicity	105
5.4	BaLinda K Objects	106
5.5	Speculative Processing.....	109

5.6	Summary	112
5.7	Bibliography	112
6	Using Scoped Behavior to Optimize Data Sharing Idioms	113
6.1	Introduction.....	113
6.2	Motivation: Data Sharing Idioms	114
6.3	Aurora: A Distributed Shared Data System	115
6.3.1	Process Models	115
6.3.2	Shared-Data Objects.....	115
6.3.3	Scoped Behavior.....	117
6.3.4	Illustrative Example: Matrix Multiplication	118
6.3.5	Discussion: Programming in Aurora.....	119
6.4	Implementation Overview.....	120
6.4.1	Handle-Body Composite Objects	120
6.4.2	Scoped Behavior Objects	121
6.5	Experience with Parallel Programs.....	122
6.5.1	Matrix Multiplication	122
6.5.2	2-D Diffusion	124
6.5.3	Parallel Sorting by Regular Sampling	126
6.6	Discussion and Related Work.....	128
6.7	Concluding Remarks	129
6.8	Bibliography	129
7	Component-Based Development Approach	131
7.1	Introduction.....	131
7.2	Component-Based Application Development	132
7.2.1	Overall Framework	132
7.2.2	Programmer Interface.....	133
7.2.3	Building Reusable Design Components.....	134
7.2.4	Assembling Reusable Design Components.....	136
7.2.5	Compiling and Running a Distributed Application.....	137
7.2.6	Alternative Design Styles	138
7.2.7	Consistency Checks	139
7.3	Advanced Features	140
7.3.1	Architecture Models as Reusable Components.....	140
7.3.2	Generation of Communication and Synchronization Code	142
7.3.3	Manipulating Messages	145
7.3.4	Repetitive Substructures in Architecture Models.....	146
7.4	Reusing Simulation Software in a Distributed Setting.....	148
7.5	Comparison Between Approaches	150
7.6	Concluding Remarks	151
7.7	Bibliography	152

8	Hypercomputing with LiPS	153
8.1	Generative Communication	153
8.1.1	Tuple Space	154
8.1.2	Benefits of Generative Communication	155
8.1.3	Example Applications	156
8.2	Using LiPS	160
8.2.1	Writing and Generating a LiPS Application	160
8.2.2	Starting the System and its Application	161
8.2.3	Monitoring the System and its Applications	162
8.2.4	Some Remarks on Fault-Tolerant Applications	163
8.3	The LiPS Runtime Systems	164
8.3.1	The Fault-Tolerant Tuple Space Machine	165
8.3.2	Performance	167
8.3.3	The LiPS Daemon (<code>lipsd()</code>)	168
8.3.4	The LiPS Daemon Controller (<code>lipsdc()</code>)	169
8.4	The LiPS Development System	170
8.4.1	Programming in CWEB	170
8.4.2	The LiPS Test Environment	172
8.5	Bibliography	173
9	An Efficient Tuple Space Programming Environment	175
9.1	Introduction	175
9.2	Tuple Space Programming	176
9.2.1	Fundamentals	177
9.2.2	Example Linda Program	179
9.2.3	Associative Memory Analysis	179
9.3	Compilation Environment	180
9.3.1	Basic Translation	180
9.3.2	Optimizing Compilers	181
9.4	Run-time Environment	187
9.4.1	Processor Location of Data	188
9.4.2	Data Structures for Efficient Data Access	190
9.4.3	Data Transfer Protocol	191
9.4.4	Process Creation	191
9.4.5	Cluster Execution Environment	192
9.4.6	Run-time Optimizations	193
9.5	Extensions	193
9.6	Conclusions	194
9.7	Bibliography	194
10	Debugging Parallelized Code	197
10.1	Introduction	197
10.2	Automatic Parallelization	198

10.3	The Debugging Problem	199
10.4	Debugging with Code Liberation.....	201
10.4.1	Overview of the Technique	201
10.4.2	The Single Assignment Form via Global Renaming	203
10.4.3	Renaming Simple Variables in Structured Code	204
10.4.4	Array Renaming.....	209
10.4.5	Renaming Unstructured Code.....	213
10.4.6	Name Reclamation	214
10.4.7	Reclaiming the Names	217
10.4.8	The Debugging Interface.....	219
10.5	Experimental Results.....	220
10.6	Conclusions	222
10.7	Bibliography	222
11	WebOS: Operating System Services for Wide-Area Applications	225
11.1	Introduction.....	225
11.2	WebOS Overview.....	227
11.3	Naming.....	229
11.4	Persistent Shared State	233
11.5	Security and Authentication.....	235
11.5.1	Validating and Revoking Statements	235
11.5.2	Processes and Roles	236
11.5.3	Authorization.....	236
11.6	Process Control.....	237
11.7	Rent-A-Server.....	237
11.7.1	Current Approaches	238
11.7.2	System Design	238
11.7.3	Performance	240
11.8	Related Work	243
11.9	Conclusions	243
11.10	Bibliography	244
II	Java for High Performance Computing	247
12	Distributed-Object Computing	249
12.1	Introduction.....	249
12.2	CORBA	250
12.2.1	Basic Model – CORBA 2.0 Architecture	251
12.2.2	OMG IDL(Interface Definition Language) and its Mapping ...	252
12.2.3	CORBA Object Services.....	253
12.2.4	A Matrix Multiplication Example.....	253
12.3	Java RMI.....	257

12.3.1	Basic Model.....	258
12.3.2	RMI Features	258
12.3.3	Matrix Multiplication	259
12.4	DCOM.....	262
12.4.1	Basic Model.....	262
12.4.2	Identification of DCOM Interfaces and Classes	263
12.4.3	IDL and its Mapping.....	263
12.4.4	Creation of Objects	264
12.4.5	Using COM Interfaces and Object Lifetime.....	264
12.4.6	DCOM Programming in Java	265
12.4.7	Matrix Multiplication Example	265
12.5	Voyager	268
12.5.1	Basic Model.....	268
12.5.2	Voyager Features	268
12.6	A Simple Performance Measurement.....	271
12.6.1	Matrix Multiplication	271
12.6.2	Which Paradigm is Superior?	272
12.7	Bibliography	273
13	Java and Different Flavors of Parallel Programming Models	274
13.1	Introduction.....	274
13.2	Java Threads—Built-in Support for Parallelism and Concurrency	275
13.2.1	Are Java Threads the Correct Model?.....	280
13.2.2	Java Support for Parallelism	281
13.3	Parallel Programming Models	282
13.3.1	Functional Models.....	283
13.3.2	Object-Oriented Models	284
13.3.3	Data Parallel Models.....	284
13.3.4	Message Passing Models	285
13.3.5	Shared Memory Models.....	287
13.4	Summary	289
13.5	Bibliography	289
14	The HPspmd Model and its Java Binding	291
14.1	Introduction.....	291
14.2	Java Language Binding	292
14.2.1	Basic Concepts	293
14.2.2	Global Variables	294
14.2.3	Program Execution Control	298
14.2.4	Communication Library Functions.....	299
14.3	Java Packages for HPspmd Programming	300
14.4	Programming Examples.....	302
14.5	Issues in the Language Design.....	302

14.5.1	Extending the Java Language	304
14.5.2	Why not HPF?	304
14.5.3	Datatypes in HPJava	305
14.6	Projects in Progress	306
14.7	Summary	307
14.8	Bibliography	308
15	Web-Based Parallel Computing with Java	310
15.1	Introduction	310
15.2	Web-Based Parallel Computing	311
15.3	Comparing Cluster with Web-Based Parallel Computing	312
15.4	Examples of Internet-Based Parallel Computing	313
15.5	Can Java be Used for Web-Based Parallel Computing?	315
15.6	Problems to be Solved in Web-Based Parallel Computing	315
15.7	A Case Study: The JET Platform	317
15.8	Some Performance Results	321
15.9	Conclusions	324
15.10	Bibliography	325
III	Algorithms and Applications	327
16	Object-Oriented Implementation of Parallel Genetic Algorithms	331
16.1	Introduction	331
16.2	Short Overview of GA Systems	332
16.3	Object-Oriented Approach to PGAs	334
16.4	Classes Representing Individuals	336
16.5	Local Genetic Operations	337
16.6	Island Model	338
16.7	Global Population Model	340
16.8	Load Balancing	341
16.9	File and I/O Operations	343
16.10	PGA Application Framework	343
16.11	Sample Results	344
16.12	Concluding Remarks	346
16.13	Bibliography	348
17	Application-Specific Load Balancing on Heterogeneous Systems	350
17.1	Introduction	350
17.2	System Overview	351
17.3	Implementation of a Complex FDTD Equation	353
17.3.1	Implementation Details	355
17.3.2	Experimental Result	355

17.4	Load Balancing	359
17.4.1	Methodology	359
17.4.2	Processor Weights	360
17.4.3	Application Data Movement (ADM)	361
17.5	Analysis	368
17.5.1	Perfect Load Balancing System	368
17.5.2	Experimental Results	369
17.5.3	Future Improvements	372
17.6	Conclusions	372
17.7	Bibliography	374
18	Time Management in Parallel Simulation	375
18.1	Introduction	375
18.1.1	Different Types of Simulations	375
18.2	Major Issues of Parallel Simulation	376
18.3	Principles of Parallel Simulation	377
18.3.1	Distributed Programming: A Simulation Example	377
18.3.2	The Apparently Sequential Nature of Simulation	377
18.3.3	A Natural Correspondence Between Problem and Solution	378
18.3.4	Logical Process Simulation	380
18.3.5	Conservative Vs. Optimistic Simulation	381
18.4	Conservative Synchronization Protocols	383
18.4.1	Variants of the Chandy-Misra Null Message Protocol	383
18.4.2	Deadlock Detection and Recovery	385
18.4.3	Global Vs. Local Lookahead	385
18.4.4	Conservative Time Windows	391
18.5	Conclusion	393
18.6	Bibliography	393
19	Hardware System Simulation	395
19.1	Introduction	395
19.2	NEPSi	396
19.2.1	ASIMUT	396
19.2.2	NEPSi as a Client Server Architecture	396
19.2.3	Overview of the NEPSi Network	398
19.2.4	Communication Model	401
19.2.5	VHDL Models Used	403
19.2.6	Tests for TNP	404
19.2.7	Initial Tests	405
19.2.8	Varying Numbers of PEs and Simulators	407
19.2.9	Matrix Multiplication on TNP	408
19.2.10	Variable Machines, Processes and PEs	411
19.2.11	Rollback	413

19.2.12 Multi Campus Simulation	413
19.2.13 Effect of Migration / Load Balancing	414
19.2.14 Network Traffic Effects	415
19.3 Discussion	415
19.4 Bibliography	417
20 Real-Time Resource Management Middleware: Open Systems and Applications	418
20.1 Introduction.....	418
20.2 Architecture of Dynamic QoS Management Middleware	421
20.3 Adaptive Resource Allocation	423
20.3.1 Automatic Survivability	424
20.3.2 Automatic Scalability	425
20.3.3 Assessment of Real-Time QoS.....	425
20.3.4 Resource Allocation and Management.....	426
20.4 Experiences with the Adaptive Resource Management Services	429
20.4.1 The Navy Testbed.....	429
20.4.2 A Real-Time Control System Benchmark	431
20.4.3 Experiments	433
20.5 Conclusions	438
20.6 Bibliography	439
21 Data Placement in Shared-Nothing Database Systems	440
21.1 Introduction.....	440
21.2 Data Placement.....	441
21.3 Declustering.....	443
21.3.1 Evaluation Time Reduction for Select and Project	444
21.3.2 Relation Co-Location for Join	445
21.4 Placement	446
21.5 Re-Distribution	448
21.5.1 Update Re-Distribution.....	450
21.5.2 Query Frequency Change Re-Distribution	450
21.6 Dynamic Re-Organization	451
21.7 Summary	452
21.8 Bibliography	453
22 Parallel Inference with Very Large Knowledge Bases	455
22.1 Introduction.....	455
22.1.1 Parallel Reflexive Reasoning on Cluster Computers.....	456
22.1.2 Contributions	457
22.2 SHRUTI: A Structured Connectionist Reasoning System	457
22.2.1 Terminology	458
22.2.2 Knowledge Encoding and Inference in Backward Reasoning...	459

22.2.3	Constraints on Rules and Inferences	461
22.3	Mapping SHRUTI onto Parallel Machines	461
22.4	SHRUTI on the CM-5—Design and Implementation	462
22.4.1	The Connection Machine CM-5	463
22.4.2	The SHRUTI-CM5 Knowledge Base	463
22.4.3	Design Considerations	464
22.4.4	Encoding the Knowledge Base	466
22.4.5	Spreading Activation and Inference	468
22.5	SHRUTI-CM5—A Mathematical Analysis	471
22.5.1	Motivation	471
22.5.2	A Summary of the Analysis	472
22.5.3	Implications of the Analysis	475
22.6	SHRUTI-CM5—Experiments with Large Knowledge Bases	477
22.6.1	Experiments with Random Knowledge Bases	477
22.6.2	Experiments with WordNet	482
22.6.3	Empirical Validation of the Analysis	483
22.7	Conclusion	483
22.8	Bibliography	484
23	MaRT: Lazy Evaluation for Parallel Ray Tracing	486
23.1	Introduction	486
23.1.1	Principles of Ray Tracing	486
23.1.2	Accelerated Ray Tracing Techniques	488
23.1.3	The Time-Memory Balance	489
23.2	On Ray Tracing Parallelization Techniques	489
23.2.1	The Image Parallel Approach	490
23.2.2	The Object Parallel Approach	490
23.2.3	Parallel Computation Analysis	491
23.2.4	Synthesis	492
23.3	MaRT: a Lazy Ray Tracer	493
23.3.1	On Lazy Evaluation	493
23.3.2	The Lazy Octree	495
23.3.3	Lazy Construction of Polygon Data	496
23.4	Parallel MaRT	497
23.4.1	Implementation of Parallel MaRT	497
23.4.2	Results	498
23.5	Concluding Remarks	503
23.6	Bibliography	505
24	Fast Content-Based Image Retrieval	506
24.1	Introduction	506
24.2	Image Feature Extraction	508
24.2.1	An Overview	509

24.2.2	Wavelet Based Multiple Image Feature Extraction	512
24.3	Dynamic Image Indexing	515
24.4	Image Similarity Measurement	516
24.5	Image Searching	517
24.6	Parallel Implementation	517
24.6.1	Divide-and-Conquer Parallel Algorithms	518
24.7	Experimental Results	519
24.7.1	Parallel Wavelet Transform Using PVM	519
24.8	Parallel Image Feature Extraction – PVM Vs. DSM	520
24.8.1	Hierarchical Image Matching in a PVM Environment	520
24.9	Conclusion	524
24.10	Bibliography	525
25	Climate Ocean Modeling	526
25.1	Introduction	526
25.2	Model Description	527
25.3	Parallel Partition on Irregular Geometries	528
25.3.1	A Simple Partition	528
25.3.2	An Efficient Partition	528
25.4	Ocean Modeling on Various Systems	531
25.4.1	Overview of Parallel Systems	532
25.4.2	Cray T3D	533
25.4.3	Cray T3E	534
25.4.4	HP SPP2000	534
25.4.5	Beowulf System-PC Clusters	535
25.4.6	Code Performance and Intercomparisons	535
25.5	Scientific Results	536
25.6	Summary	537
25.7	Bibliography	539
26	Computational Electromagnetics	540
26.1	Introduction	540
26.2	Physical Optics Method	543
26.3	Finite-Difference Time-Domain Method	548
26.4	Finite-Element Integral-Equation Coupled Method	550
26.5	Conclusions	554
26.6	Bibliography	556
27	CFD Simulation: A Case Study in Software Engineering	558
27.1	Introduction	558
27.2	TfC – a State-of-the-Art Industrial CFD Package	559
27.3	Requirements for Parallel CFD Simulation	563
27.4	Design and Implementation of ParTfC	567

27.5	Object Oriented Design of Scientific Software.....	574
27.6	Productive Use of Parallel Scientific Computing Software	575
27.6.1	Resource Management in Workstation Clusters	575
27.6.2	The Pre- and Post-Processing Bottlenecks	577
27.6.3	Network Based Collaborative Design Environments.....	578
27.7	Bibliography	579
28	Quantum Reactive Scattering Calculations	580
28.1	Introduction.....	580
28.2	The Many Body Problem: Description, Decomposition, and Solutions	581
28.2.1	The Decomposition of the Problem and the Mathematical Formalism.....	582
28.2.2	The Integration of Scattering Equations	584
28.3	Parallelization Strategies.....	586
28.3.1	Task Farm Model.....	588
28.3.2	Pipeline Model.....	590
28.4	Parallel Implementations on CRAY T3E.....	595
28.4.1	Task Farm Model.....	595
28.4.2	Pipeline Model.....	595
28.4.3	Performance Evaluation	596
28.5	Parallel Implementation on SGI Origin 2000	598
28.5.1	Task Farm Model.....	598
28.5.2	Performance Evaluation	598
28.6	Parallel Implementation on a Metacomputer	598
28.6.1	Implementation Issues	599
28.6.2	Performance Evaluation	600
28.7	Concluding Remarks	601
28.8	Bibliography	602
29	Biomedical Applications Modeling	604
29.1	Introduction.....	604
29.2	The Chromosome Reconstruction Problem.....	605
29.2.1	Physical Mapping via Clone Ordering	605
29.3	PVM Algorithms for Chromosome Reconstruction	606
29.3.1	Simulated Annealing and Microcanonical Annealing	606
29.3.2	Parallel SA and MCA Using PVM.....	609
29.4	Heart Rate Variability and Kolmogorov Entropy	613
29.4.1	K_2 Entropy	613
29.4.2	Serial Computation of the Correlation Integral	616
29.5	A Parallel Algorithm for K_2 Entropy Computation using PVM.....	616
29.6	Optimal Scaling Region Determination Algorithm	618
29.7	Experimental Results.....	620
29.7.1	Chromosome Reconstruction	620

29.7.2 K_2 Entropy Computation.....	621
29.8 Conclusions	623
29.9 Bibliography	624
A Glossary	627
Index	659