
High Performance Cluster Computing: Architectures and Systems, Volume 1

Edited by

Rajkumar Buyya
(rajkumar@dgs.monash.edu.au)

School of Computer Science and Software Engineering
Monash University
Melbourne, Australia

Contents at a Glance

Preface	xxix
I Requirements and General Issues	1
1 Cluster Computing at a Glance	3
2 Cluster Setup and its Administration	48
3 Constructing Scalable Services	68
4 Dependable Clustered Computing	94
5 Deploying a High Throughput Computing Cluster	116
6 Performance Models and Simulation	135
7 Metacomputing: Harnessing Informal Supercomputers	154
8 Specifying Resources and Services in Metacomputing Environments	186
II Networking, Protocols, and I/O	201
9 High Speed Networks	204
10 Lightweight Messaging Systems	246
11 Active Messages	270
12 Xpress Transport Protocol	301
13 Congestion Management in ATM Clusters	317
14 Load Balancing Over Networks	340
15 Multiple Path Communication	364
16 Network RAM	384
17 Distributed Shared Memory	410
18 Parallel I/O for Clusters: Methodologies and Systems	441
19 Software RAID and Parallel Filesystems	465

III	Process Scheduling, Load Sharing, and Balancing	499
20	Job and Resource Management Systems	501
21	Scheduling Parallel Jobs on Clusters	521
22	Load Sharing and Fault Tolerance Manager	536
23	Parallel Program Scheduling Techniques	556
24	Customized Dynamic Load Balancing	582
25	Mapping and Scheduling on Heterogeneous Systems	608
IV	Representative Cluster Systems	625
26	Beowulf	629
27	RWC PC Cluster II and SCore Cluster System Software	650
28	COMPaS: A Pentium Pro PC-Based SMP Cluster	665
29	The NanOS Cluster Operating System	686
30	BSP-Based Adaptive Parallel Processing	706
31	MARS: An Adaptive Parallel Programming Environment	726
32	The Gardens Approach to Adaptive Parallel Computing	744
33	The ParPar System: A Software MPP	758
34	Pitt Parallel Computer	775
35	The RS/6000 SP System: A Scalable Parallel Cluster	794
36	A Scalable and Highly Available Clustered Web Server	815
	Index	845

Contents

Preface	xxix
I Requirements and General Issues	1
1 Cluster Computing at a Glance	3
1.1 Introduction	3
1.1.1 Eras of Computing	4
1.2 Scalable Parallel Computer Architectures	5
1.3 Towards Low Cost Parallel Computing and Motivations	7
1.4 Windows of Opportunity	9
1.5 A Cluster Computer and its Architecture	10
1.6 Clusters Classifications	11
1.7 Commodity Components for Clusters	14
1.7.1 Processors	14
1.7.2 Memory and Cache	15
1.7.3 Disk and I/O	15
1.7.4 System Bus	16
1.7.5 Cluster Interconnects	16
1.7.6 Operating Systems	18
1.8 Network Services/Communication SW	21
1.9 Cluster Middleware	22
1.9.1 Middleware Layers	24
1.9.2 SSI Boundaries	25
1.9.3 Middleware Design Goals	26
1.9.4 Key Services of SSI and Availability Infrastructure	27
1.10 Resource Management and Scheduling (RMS)	28
1.11 Programming Environments and Tools	30
1.11.1 Threads	30
1.11.2 Message Passing Systems (MPI and PVM)	31
	vii

1.11.3	Distributed Shared Memory (DSM) Systems	31
1.11.4	Parallel Debuggers and Profilers	32
1.11.5	Performance Analysis Tools	33
1.11.6	Cluster Administration Tools	33
1.12	Cluster Applications	35
1.13	Representative Cluster Systems	35
1.13.1	The Berkeley Network Of Workstations (NOW) Project	35
1.13.2	The High Performance Virtual Machine (HPVM) Project	37
1.13.3	The Beowulf Project	38
1.13.4	Solaris MC: A High Performance Operating System for Clusters	39
1.13.5	A Comparison of the Four Cluster Environments	40
1.14	Cluster of SMPs (CLUMPS)	41
1.15	Summary and Conclusions	42
1.15.1	Hardware and Software Trends	42
1.15.2	Cluster Technology Trends	44
1.15.3	Future Cluster Technologies	44
1.15.4	Final Thoughts	45
1.16	Bibliography	46
2	Cluster Setup and its Administration	48
2.1	Introduction	48
2.2	Setting up the Cluster	49
2.2.1	Starting from Scratch	49
2.2.2	Directory Services inside the Cluster	51
2.2.3	DCE Integration	52
2.2.4	Global Clock Synchronization	53
2.2.5	Heterogeneous Clusters	53
2.2.6	Some Experiences with PoPC Clusters	54
2.3	Security	55
2.3.1	Security Policies	55
2.3.2	Finding the Weakest Point in NOWs and COWs	56
2.3.3	A Little Help from a Front-end	57
2.3.4	Security versus Performance Tradeoffs	57
2.3.5	Clusters of Clusters	58
2.4	System Monitoring	59
2.4.1	Unsuitability of General Purpose Monitoring Tools	59
2.4.2	Subjects of Monitoring	60
2.4.3	Self Diagnosis and Automatic Corrective Procedures	61
2.5	System Tuning	62
2.5.1	Developing Custom Models for Bottleneck Detection	62
2.5.2	Focusing on Throughput or Focusing on Latency	62
2.5.3	I/O Implications	62

2.5.4	Caching Strategies	64
2.5.5	Fine-tuning the OS	64
2.6	Bibliography	66
3	Constructing Scalable Services	68
3.1	Introduction	68
3.2	Environment	69
3.2.1	Faults, Delays, and Mobility	69
3.2.2	Scalability Definition and Measurement	70
3.2.3	Weak Consistency	73
3.2.4	Assumptions Summary	74
3.2.5	Model Definition and Requirements	74
3.3	Resource Sharing	75
3.3.1	Introduction	75
3.3.2	Previous study	76
3.3.3	Flexible Load Sharing Algorithm	77
3.3.4	Resource Location Study	77
3.3.5	Algorithm Analysis	78
3.4	Resource Sharing Enhanced Locality	80
3.4.1	State Metric	81
3.4.2	Basic Algorithm Preserving Mutual Interests	81
3.4.3	Considering Proximity for Improved Performance	83
3.4.4	Estimating Proximity (latency)	85
3.4.5	Simulation Runs	85
3.4.6	Simulation Results	87
3.5	Prototype Implementation and Extension	89
3.5.1	PVM Resource Manager	89
3.5.2	Resource Manager Extension to Further Enhance Locality	89
3.5.3	Initial Performance Measurement Results	90
3.6	Conclusions and Future Study	91
3.7	Bibliography	92
4	Dependable Clustered Computing	94
4.1	Introduction	94
4.1.1	Structure	96
4.2	Two Worlds Converge	96
4.2.1	Dependable Parallel Computing	96
4.2.2	Mission/Business Critical Computing	98
4.3	Dependability Concepts	100
4.3.1	Faults, Errors, Failures	100
4.3.2	Dependability Attributes	101
4.3.3	Dependability Means	101
4.4	Cluster Architectures	101

4.4.1	Share-Nothing versus Shared-Storage	101
4.4.2	Active/Standby versus Active/Active	102
4.4.3	Interconnects	104
4.5	Detecting and Masking Faults	106
4.5.1	Self-Testing	106
4.5.2	Processor, Memory, and Buses	107
4.5.3	Watchdog Hardware Timers	107
4.5.4	Loosing the Software Watchdog	108
4.5.5	Assertions, Consistency Checking, and ABFT	109
4.6	Recovering from Faults	109
4.6.1	Checkpointing and Rollback	109
4.6.2	Transactions	110
4.6.3	Failover and Failback	111
4.6.4	Reconfiguration	111
4.7	The Practice of Dependable Clustered Computing	112
4.7.1	Microsoft Cluster Server	112
4.7.2	NCR LifeKeeper	113
4.7.3	Oracle Fail Safe and Parallel Server	113
4.8	Bibliography	114
5	Deploying a High Throughput Computing Cluster	116
5.1	Introduction	116
5.2	Condor Overview	117
5.3	Software Development	118
5.3.1	Layered Software Architecture	119
5.3.2	Layered Resource Management Architecture	120
5.3.3	Protocol Flexibility	120
5.3.4	Remote File Access	121
5.3.5	Checkpointing	123
5.4	System Administration	124
5.4.1	Access Policies	124
5.4.2	Reliability	126
5.4.3	Problem Diagnosis via System Logs	128
5.4.4	Monitoring and Accounting	129
5.4.5	Security	130
5.4.6	Remote Customers	132
5.5	Summary	133
5.6	Bibliography	133
6	Performance Models and Simulation	135
6.1	Introduction	135
6.2	New Performance Issues	136
6.2.1	Profit-Effective Parallel Computing.	136

6.2.2	Impact of Heterogeneity and Nondedication	137
6.2.3	Communication Interactions	139
6.3	A Cost Model for Effective Parallel Computing	139
6.3.1	The Memory Hierarchy	140
6.3.2	Parallel Program Structures	142
6.3.3	The Cost Model and Memory Access Time Prediction	143
6.3.4	Validation of the Framework and its Models	146
6.4	Conclusions	152
6.5	Bibliography	152
7	Metacomputing: Harnessing Informal Supercomputers	154
7.1	General Introduction	154
7.1.1	Why Do We Need Metacomputing?	155
7.1.2	What Is a Metacomputer?	156
7.1.3	The Parts of a Metacomputer	156
7.2	The Evolution of Metacomputing	157
7.2.1	Introduction	157
7.2.2	Some Early Examples	158
7.3	Metacomputer Design Objectives and Issues	161
7.3.1	General Principles	162
7.3.2	Underlying Hardware and Software Infrastructure	162
7.3.3	Middleware – The Metacomputing Environment	163
7.4	Metacomputing Projects	166
7.4.1	Introduction	166
7.4.2	Globus	166
7.4.3	Legion	170
7.4.4	WebFlow	175
7.5	Emerging Metacomputing Environments	181
7.5.1	Introduction	181
7.5.2	Summary	181
7.6	Summary and Conclusions	181
7.6.1	Introduction	181
7.6.2	Summary of the Reviewed Metacomputing Environments	182
7.6.3	Some Observations	183
7.6.4	Metacomputing Trends	183
7.6.5	The Impact of Metacomputing	184
7.7	Bibliography	184
8	Specifying Resources and Services in Metacomputing Environments	186
8.1	The Need for Resource Description Tools	186
8.2	Schemes for Specifying Computer Components, Software, and Services	187
8.2.1	Resource Specification in Local HPC-Systems	187
8.2.2	Resource Specification in Distributed Client-Server Systems	188

8.2.3	The Metacomputing Directory Service (MDS)	188
8.2.4	The Resource Description Language (RDL)	189
8.3	Resource and Service Description (RSD)	190
8.3.1	Requirements	190
8.3.2	Architecture	191
8.3.3	Graphical Interface	192
8.3.4	Language Interface	194
8.3.5	Internal Data Representation	195
8.3.6	Implementation	198
8.4	Summary	199
8.5	Bibliography	199
II Networking, Protocols, and I/O		201
9	High Speed Networks	204
9.1	Introduction	204
9.1.1	Choice of High Speed Networks	204
9.1.2	Evolution in Interconnect Trends	206
9.2	Design Issues	207
9.2.1	Goals	207
9.2.2	General Architecture	208
9.2.3	Design Details	211
9.3	Fast Ethernet	217
9.3.1	Fast Ethernet Migration	218
9.4	High Performance Parallel Interface (HiPPI)	219
9.4.1	HiPPI-SC (Switch Control)	219
9.4.2	Serial HiPPI	220
9.4.3	High Speed SONET Extensions	220
9.4.4	HiPPI Connection Management	221
9.4.5	HiPPI Interfaces	221
9.4.6	Array System: The HiPPI Interconnect	223
9.5	Asynchronous Transfer Mode (ATM)	224
9.5.1	Concepts	224
9.5.2	ATM adapter	225
9.5.3	ATM API basics	226
9.5.4	Performance Evaluation of ATM	227
9.5.5	Issues in Distributed Networks for ATM Networks	229
9.6	Scalable Coherent Interface (SCI)	229
9.6.1	Data Transfer via SCI	229
9.6.2	Advantages of SCI	231
9.7	ServerNet	232
9.7.1	Scalability and Reliability as Main Goals	232

9.7.2	Driver and Management Software	234
9.7.3	Remarks	234
9.8	Myrinet	235
9.8.1	Fitting Everybodys Needs	235
9.8.2	Software and Performance	237
9.8.3	Remarks	238
9.9	Memory Channel	238
9.9.1	Bringing together Simplicity and Performance	238
9.9.2	Software and Performance	240
9.9.3	Remarks	240
9.10	Synfinity	241
9.10.1	Pushing Networking to the Technological Limits	241
9.10.2	Remarks	243
9.11	Bibliography	243
10	Lightweight Messaging Systems	246
10.1	Introduction	246
10.2	Latency/Bandwidth Evaluation of Communication Performance	247
10.3	Traditional Communication Mechanisms for Clusters	249
10.3.1	TCP, UDP, IP, and Sockets	249
10.3.2	RPC	250
10.3.3	MPI and PVM	250
10.3.4	Active Messages	251
10.4	Lightweight Communication Mechanisms	251
10.4.1	What We Need for Efficient Cluster Computing	252
10.4.2	Typical Techniques to Optimize Communication	254
10.4.3	The Importance of Efficient Collective Communications	256
10.4.4	A Classification of Lightweight Communication Systems	257
10.5	Kernel-Level Lightweight Communications	258
10.5.1	Industry-Standard API Systems	258
10.5.2	Best-Performance Systems	260
10.6	User-Level Lightweight Communications	262
10.6.1	BIP	263
10.6.2	Fast Messages	264
10.6.3	Hewlett-Packard Active Messages (HPAM)	264
10.6.4	U-Net for ATM	264
10.6.5	Virtual Interface Architecture (VIA)	265
10.7	A Comparison Among Message Passing Systems	266
10.7.1	Clusters Versus MPPs	267
10.7.2	Standard Interface Approach Versus Other Approaches	267
10.7.3	User-Level Versus Kernel-Level	267
10.8	Bibliography	268

11 Active Messages	270
11.1 Introduction	270
11.2 Requirements	271
11.2.1 Top-down Requirement	271
11.2.2 Bottom-up Requirement	272
11.2.3 Architecture and Implementation	273
11.2.4 Summary	274
11.3 AM Programming Model	274
11.3.1 Endpoints and Bundles	275
11.3.2 Transport Operations	277
11.3.3 Error Model	278
11.3.4 Programming Examples	279
11.4 AM Implementation	280
11.4.1 Endpoints and Bundles	282
11.4.2 Transport Operations	285
11.4.3 NIC Firmware	287
11.4.4 Message Delivery and Flow Control	288
11.4.5 Events and Error handling	289
11.4.6 Virtual Networks	290
11.5 Analysis	291
11.5.1 Meeting the Requirements	292
11.6 Programming Models on AM	294
11.6.1 Message Passing Interface (MPI)	294
11.6.2 Fast Sockets	296
11.7 Future Work	297
11.7.1 Bandwidth Performance	297
11.7.2 Flow Control and Error Recovery	297
11.7.3 Shared Memory Protocol	298
11.7.4 Endpoint Scheduling	298
11.7.5 Multidevice Support	298
11.7.6 Memory Management on NIC	299
11.8 Bibliography	299
12 Xpress Transport Protocol	301
12.1 Network Services for Cluster Computing	301
12.2 A New Approach	303
12.3 XTP Functionality	304
12.3.1 Multicast	304
12.3.2 Multicast Group Management (MGM)	305
12.3.3 Priority	306
12.3.4 Rate and Burst Control	307
12.3.5 Connection Management	308

12.3.6	Selectable Error Control	308
12.3.7	Selectable Flow Control	309
12.3.8	Selective Retransmission	309
12.3.9	Selective Acknowledgment	309
12.3.10	Maximum Transmission Unit (MTU) Detection	310
12.3.11	Out-of-band Data	310
12.3.12	Alignment	310
12.3.13	Traffic Descriptors	310
12.4	Performance	310
12.4.1	Throughput	311
12.4.2	Message Throughput	311
12.4.3	End-to-end Latency	312
12.5	Applications	313
12.5.1	Multicast	313
12.5.2	Gigabyte Files	313
12.5.3	High Performance	313
12.5.4	Image Distribution	314
12.5.5	Digital Telephone	314
12.5.6	Video File Server	314
12.5.7	Priority Support	314
12.5.8	Real-time Systems	314
12.5.9	Interoperability	314
12.6	XTP's Future in Cluster Computing	315
12.7	Bibliography	315
13	Congestion Management in ATM Clusters	317
13.1	Introduction to ATM Networking	317
13.1.1	Integrated Broadband Solution	318
13.1.2	Virtual Connection Setup	319
13.1.3	Quality of Service	320
13.1.4	Traffic and Congestion Management	320
13.2	Existing Methodologies	321
13.3	Simulation of ATM on LAN	323
13.3.1	Different Types of Traffic	324
13.3.2	Analysis of Results	325
13.3.3	Heterogeneous Traffic Condition	329
13.3.4	Summary	330
13.4	Migration Planning	331
13.4.1	LAN to Directed Graph	331
13.4.2	A Congestion Locator Algorithm	333
13.4.3	An Illustration	334
13.5	Conclusions	337

13.6 Bibliography	338
14 Load Balancing Over Networks	340
14.1 Introduction	340
14.2 Methods	341
14.2.1 Factors Affecting Balancing Methods	341
14.2.2 Simple Balancing Methods	345
14.2.3 Advanced Balancing Methods	347
14.3 Common Errors	353
14.3.1 Overflow	353
14.3.2 Underflow	353
14.3.3 Routing Errors	354
14.3.4 Induced Network Errors	354
14.4 Practical Implementations	355
14.4.1 General Network Traffic Implementations	355
14.4.2 Web-specific Implementations	359
14.4.3 Other Application Specific Implementations	360
14.5 Summary	362
14.6 Bibliography	362
15 Multiple Path Communication	364
15.1 Introduction	364
15.2 Heterogeneity in Networks and Applications	365
15.2.1 Varieties of Communication Networks	366
15.2.2 Exploiting Multiple Communication Paths	366
15.3 Multiple Path Communication	367
15.3.1 Performance-Based Path Selection	368
15.3.2 Performance-Based Path Aggregation	369
15.3.3 PBPD Library	370
15.4 Case Study	371
15.4.1 Multiple Path Characteristics	371
15.4.2 Communication Patterns of Parallel Applications	375
15.4.3 Experiments and Results	378
15.5 Summary and Conclusion	381
15.6 Bibliography	381
16 Network RAM	384
16.1 Introduction	384
16.1.1 Issues in Using Network RAM	387
16.2 Remote Memory Paging	387
16.2.1 Implementation Alternatives	388
16.2.2 Reliability	395
16.2.3 Remote Paging Prototypes	400

16.3	Network Memory File Systems	402
16.3.1	Using Network Memory as a File Cache	402
16.3.2	Network RamDisks	403
16.4	Applications of Network RAM in Databases	404
16.4.1	Transaction-Based Systems	404
16.5	Summary	406
16.5.1	Conclusions	406
16.5.2	Future Trends	407
16.6	Bibliography	408
17	Distributed Shared Memory	410
17.1	Introduction	410
17.2	Data Consistency	411
17.2.1	Data Location	412
17.2.2	Write Synchronization	415
17.2.3	Double Faulting	416
17.2.4	Relaxing Consistency	417
17.2.5	Application/Type-specific Consistency	421
17.3	Network Performance Issues	423
17.4	Other Design Issues	424
17.4.1	Synchronization	424
17.4.2	Granularity	426
17.4.3	Address-Space Structure	427
17.4.4	Replacement Policy and Secondary Storage	428
17.4.5	Heterogeneity Support	429
17.4.6	Fault Tolerance	431
17.4.7	Memory Allocation	433
17.4.8	Data Persistence	433
17.5	Conclusions	434
17.6	Bibliography	435
18	Parallel I/O for Clusters: Methodologies and Systems	441
18.1	Introduction	441
18.2	A Case for Cluster I/O Systems	442
18.3	The Parallel I/O Problem	444
18.3.1	Regular Problems	444
18.3.2	Irregular Problems	445
18.3.3	Out-of-Core Computation	445
18.4	File Abstraction	445
18.5	Methods and Techniques	446
18.5.1	Two-Phase Method	448
18.5.2	Disk-Directed I/O	448
18.5.3	Two-Phase Data Administration	449

18.6	Architectures and Systems	450
18.6.1	Runtime Modules and Libraries	450
18.6.2	MPI-IO	452
18.6.3	Parallel File Systems	453
18.6.4	Parallel Database Systems	453
18.7	The ViPIOS Approach	455
18.7.1	Design Principles	455
18.7.2	System Architecture	456
18.7.3	Data Administration	460
18.8	Conclusions and Future Trends	462
18.9	Bibliography	463
19	Software RAID and Parallel Filesystems	465
19.1	Introduction	465
19.1.1	I/O Problems	465
19.1.2	Using Clusters to Increase the I/O Performance	466
19.2	Physical Placement of Data	466
19.2.1	Increasing the Visibility of the Filesystems	467
19.2.2	Data Striping	469
19.2.3	Log-Structured Filesystems	474
19.2.4	Solving the Small-Write Problem in Clusters of Workstations	476
19.2.5	Network-Attached Devices	477
19.3	Caching	478
19.3.1	Multilevel Caching	479
19.3.2	Cache-Coherence Problems	480
19.3.3	Cooperative Caching	483
19.4	Prefetching	485
19.4.1	“Parallel” Prefetching	485
19.4.2	Transparent Informed Prefetching	486
19.4.3	Scheduling Parallel Prefetching and Caching	486
19.5	Interfaces	489
19.5.1	Traditional Interface	490
19.5.2	Shared File Pointers	490
19.5.3	Access Methods	491
19.5.4	Data Distribution	494
19.5.5	Collective I/O	496
19.5.6	Extensible Systems	496
19.6	Bibliography	497
III	Process Scheduling, Load Sharing, and Balancing	499
20	Job and Resource Management Systems	501

20.1	Motivation and Historical Evolution	501
20.1.1	A Need for Job Management	501
20.1.2	Job Management Systems on Workstation Clusters	502
20.1.3	Primary Application Fields	503
20.2	Components and Architecture of Job- and Resource Management Systems	503
20.2.1	Prerequisites	503
20.2.2	User Interface	504
20.2.3	Administrative Environment	504
20.2.4	Managed Objects: Queues, Hosts, Resources, Jobs, Policies	504
20.2.5	A Modern Architectural Approach	507
20.3	The State-of-the-Art in RMS	508
20.3.1	Automated Policy Based Resource Management	508
20.3.2	The State-of-the-Art of Job Support	510
20.4	Challenges for the Present and the Future	515
20.4.1	Open Interfaces	515
20.4.2	Resource Control and Mainframe-Like Batch Processing	516
20.4.3	Heterogeneous Parallel Environments	517
20.4.4	RMS in a WAN Environment	519
20.5	Summary	519
20.6	Bibliography	519
21	Scheduling Parallel Jobs on Clusters	521
21.1	Introduction	521
21.2	Background	522
21.2.1	Cluster Usage Modes	522
21.2.2	Job Types and Requirements	522
21.3	Rigid Jobs with Process Migration	523
21.3.1	Process Migration	523
21.3.2	Case Study: PVM with Migration	524
21.3.3	Case Study: MOSIX	525
21.4	Malleable Jobs with Dynamic Parallelism	526
21.4.1	Identifying Idle Workstations	526
21.4.2	Case Study: Condor and WoDi	526
21.4.3	Case Study: Piranha and Linda	528
21.5	Communication-Based Coscheduling	529
21.5.1	Demand-Based Coscheduling	529
21.5.2	Implicit Coscheduling	530
21.6	Batch Scheduling	531
21.6.1	Admission Controls	531
21.6.2	Case Study: Utopia/LSF	531
21.7	Summary	533

21.8 Bibliography	534
22 Load Sharing and Fault Tolerance Manager	536
22.1 Introduction	536
22.2 Load Sharing in Cluster Computing	537
22.3 Fault Tolerance by Means of Checkpointing	538
22.3.1 Checkpointing a Single Process	538
22.3.2 Checkpointing of Communicating Processes	539
22.4 Integration of Load Sharing and Fault Tolerance	540
22.4.1 Environment and Architecture	540
22.4.2 Process Allocation	542
22.4.3 Failure Management	545
22.4.4 Performance Study	546
22.5 Related Works	552
22.6 Conclusion	553
22.7 Bibliography	554
23 Parallel Program Scheduling Techniques	556
23.1 Introduction	556
23.2 The Scheduling Problem for Network Computing Environments	558
23.2.1 The DAG Model	559
23.2.2 Generation of a DAG	559
23.2.3 The Cluster Model	560
23.2.4 NP-Completeness of the DAG Scheduling Problem	561
23.2.5 Basic Techniques in DAG Scheduling	561
23.3 Scheduling Tasks to Machines Connected via Fast Networks	564
23.3.1 The ISH Algorithm	564
23.3.2 The MCP Algorithm	565
23.3.3 The ETF Algorithm	566
23.3.4 Analytical Performance Bounds	567
23.4 Scheduling Tasks to Arbitrary Processors Networks	569
23.4.1 The Message Routing Issue	569
23.4.2 The MH Algorithm	569
23.4.3 The DLS Algorithm	570
23.4.4 The BSA Algorithm	572
23.5 CASCH: A Parallelization and Scheduling Tool	574
23.5.1 User Programs	575
23.5.2 Lexical Analyzer and Parser	575
23.5.3 Weight Estimator	575
23.5.4 DAG Generation	577
23.5.5 Scheduling/Mapping Tool	577
23.5.6 Communication Inserter	578
23.5.7 Code Generation	578

23.5.8 Graphical User Interface	578
23.6 Summary and Concluding Remarks	579
23.7 Bibliography	580
24 Customized Dynamic Load Balancing	582
24.1 Introduction	582
24.1.1 Related Work	583
24.2 Dynamic Load Balancing (DLB)	585
24.2.1 Load Balancing Strategies	586
24.2.2 Discussion	587
24.3 DLB Modeling and Decision Process	588
24.3.1 Modeling Parameters	588
24.3.2 Modeling the Strategies – Total Cost Derivation	591
24.3.3 Decision Process – Using the Model	595
24.4 Compiler and Runtime Systems	596
24.4.1 Runtime System	596
24.4.2 Code Generation	596
24.5 Experimental Results	598
24.5.1 Network Characterization	598
24.5.2 MXM: Matrix Multiplication	599
24.5.3 TRFD	601
24.5.4 AC: Adjoint Convolution	603
24.5.5 Modeling Results: MXM, TRFD, and AC	604
24.6 Summary	606
24.7 Bibliography	606
25 Mapping and Scheduling on Heterogeneous Systems	608
25.1 Introduction	608
25.2 Mapping and Scheduling	609
25.2.1 The Mapping Problem	609
25.2.2 The Scheduling Problem	611
25.3 The Issues of Task Granularity and Partitioning	613
25.3.1 Two Strategies of Scheduling in Clustering	613
25.3.2 Some Effective Partitioning Algorithms	614
25.4 Static Scheduling and Dynamic Scheduling	617
25.4.1 Related Work in Homogeneous Systems	617
25.4.2 Further Work Relating to Heterogeneous Systems	618
25.5 Load Balancing Issues	619
25.5.1 Load Balancing in Homogeneous Environment	619
25.5.2 Heterogeneous Computing Environment (HCE)	621
25.6 Summary	622
25.7 Bibliography	622

IV Representative Cluster Systems	625
26 Beowulf	629
26.1 Searching for Beowulf	629
26.1.1 The Beowulf Model: Satisfying a Critical Need	630
26.1.2 A Short History of Large Achievements	630
26.1.3 Application Domains	632
26.1.4 Other Sources of Information	633
26.2 System Architecture Evolution	634
26.2.1 The Processor	634
26.2.2 The Network	635
26.2.3 Putting It All Together	637
26.3 Prevailing Software Practices	638
26.3.1 Small Scale Software Provides Big Scale Performance	638
26.3.2 The Linux Operating System	640
26.4 Next Steps in Beowulf-Class Computing	641
26.4.1 Grendel - Towards Uniform System Software	641
26.4.2 Large System Scaling	643
26.4.3 Data-Intensive Computation	645
26.5 Beowulf in the 21st Century	646
26.5.1 Processing Nodes	646
26.5.2 Storage	646
26.5.3 System Area Networks	647
26.5.4 The \$1M TFLOPS Beowulf	647
26.5.5 The Software Barrier	648
26.5.6 Not the Final Word	648
26.6 Bibliography	649
27 RWC PC Cluster II and SCore Cluster System Software	650
27.1 Introduction	650
27.2 Building a Compact PC Cluster Using Commodity Hardware	651
27.2.1 Overview	651
27.2.2 Networks	653
27.2.3 Processor Card	655
27.2.4 Chassis Design	655
27.2.5 Cooling System	657
27.3 SCore Parallel Operating System Environment on Top of Unix	657
27.3.1 Software Overview	657
27.3.2 PM High Performance Communication Driver and Library	658
27.3.3 MPI on PM	658
27.3.4 SCore-D Parallel Operating System	659
27.3.5 MPC++ Multi-Thread Template Library	659
27.4 Performance Evaluation	660

27.4.1	PM Basic Performance	660
27.4.2	MPI Basic Performance	660
27.4.3	NAS Parallel Benchmarks Result	660
27.4.4	SCore-D Gang Scheduling Overhead	661
27.5	Concluding Remarks	663
27.6	Bibliography	664
28	COMPaS: A Pentium Pro PC-Based SMP Cluster	665
28.1	COMPaS: A Pentium Pro PC-Based SMP Cluster	665
28.2	Building PC-Based SMP Cluster	666
28.2.1	Pentium Pro PC-Based SMP Node	666
28.2.2	Inter-Node Communication on 100Base-T Ethernet	668
28.2.3	NICAM: User-Level Communication Layer of Myrinet for SMP Cluster	668
28.3	Programming for SMP Cluster	673
28.3.1	All Message Passing Programming	674
28.3.2	All Shared Memory Programming	674
28.3.3	Hybrid Shared Memory/Distributed Memory Programming	674
28.4	Case Studies – Benchmarks Results on COMPaS	676
28.4.1	Explicit Laplace Equation Solver	676
28.4.2	Matrix-Matrix Multiplication	678
28.4.3	Sparse Matrix Conjugate Gradient Kernel	679
28.4.4	Radix Sort	680
28.5	Guidelines for Programming in PC-Based SMP Cluster	682
28.6	Summary	684
28.7	Bibliography	684
29	The NanOS Cluster Operating System	686
29.1	Introduction	686
29.1.1	Design Objectives	687
29.2	Architecture Overview	688
29.2.1	NanOS Microkernel	689
29.2.2	Membership Service	690
29.2.3	Object Request Broker	691
29.2.4	HIDRA Support for High Availability	691
29.3	NanOS	692
29.3.1	An Object-Oriented Microkernel	692
29.3.2	Microkernel Architecture	693
29.4	MCMM	696
29.4.1	MCMM Protocol	696
29.5	HIDRA	698
29.5.1	Overview of HIDRA	699
29.5.2	Replication Models	699

29.5.3	Object Request Broker	700
29.5.4	Coordinator-Cohort Replication Model	701
29.6	Summary	703
29.7	Bibliography	704
30	BSP-Based Adaptive Parallel Processing	706
30.1	Introduction	706
30.2	The Bulk-Synchronous Parallel Model	706
30.2.1	Cluster of Workstations as a BSP Computer	708
30.2.2	Program Reorganization for Parallel Computing on Dedicated Clusters: Plasma Simulation	709
30.3	Parallel Computing on Nondedicated Workstations	709
30.3.1	Nondedicated Workstations as Transient Processors	709
30.3.2	Approaches to Adaptive Parallelism	710
30.4	Adaptive Parallelism in the BSP Model	712
30.4.1	Protocol for Replication and Recovery	712
30.4.2	Performance of Adaptive Replication	714
30.5	A Programming Environment for Adaptive BSP	714
30.5.1	Dynamic Extensions to the Oxford BSP Library	715
30.5.2	The Replication Layer	715
30.5.3	The User Layer	716
30.6	Application of A-BSP to Parallel Computations	718
30.6.1	Maximum Independent Set	719
30.6.2	Plasma Simulation	719
30.6.3	Results	719
30.7	Application of A-BSP to Nondedicated Workstations	721
30.8	Conclusions	723
30.9	Bibliography	723
31	MARS: An Adaptive Parallel Programming Environment	726
31.1	Motivation and Goals	726
31.2	Related Work	728
31.2.1	Exploiting Idle Time	728
31.2.2	Adaptive Schedulers	729
31.3	The Available Capacity of NOWs	729
31.3.1	Node Idleness	729
31.3.2	Aggregate Idle Time	731
31.4	The MARS Approach	731
31.4.1	MARS Infrastructure	731
31.4.2	Parallel Programming Methodology	733
31.4.3	The MARS Scheduler	737
31.5	Experimental Results	738
31.5.1	Efficiency and Adaptability	740

31.5.2	Fault Tolerance and Intrusion	740
31.6	Conclusion and Future Work	742
31.7	Bibliography	742
32	The Gardens Approach to Adaptive Parallel Computing	744
32.1	Introduction	744
32.2	Related Work	746
32.3	Communication	748
32.3.1	Active Messages	748
32.3.2	Global Objects	749
32.3.3	Poll Procedure Annotations	750
32.4	Adaptation and Tasking	751
32.4.1	Multitasking	752
32.4.2	Blocking	753
32.4.3	Task Migration	754
32.4.4	Gardens Screen Saver	754
32.5	Performance Results	755
32.6	Summary	755
32.7	Bibliography	756
33	The ParPar System: A Software MPP	758
33.1	Introduction	758
33.2	The ParPar System	759
33.2.1	Hardware Base	759
33.2.2	Software Structure	760
33.2.3	Design Principles	761
33.2.4	Control Protocols	761
33.2.5	Data Network	763
33.3	System Configuration and Control	764
33.3.1	Dynamic Reconfiguration	764
33.3.2	Reliability and Availability	764
33.3.3	The Master Control	765
33.4	Job Control	765
33.4.1	Job Initiation	765
33.4.2	Job Termination	766
33.4.3	Debugging	767
33.5	Scheduling	768
33.5.1	Adaptive Partitioning	768
33.5.2	Gang Scheduling	769
33.6	Parallel I/O	770
33.6.1	Terminal I/O	770
33.6.2	Parallel Files	771
33.7	Project Status	773

33.8 Bibliography	773
34 Pitt Parallel Computer	775
34.1 Introduction	775
34.2 The Operating System	776
34.2.1 Internode Communication	778
34.2.2 Typical Usage	779
34.2.3 A Problem Suite for Research	780
34.3 The Laplace Problem	780
34.3.1 A One-Dimensional Example	780
34.3.2 A Two-Dimensional Example	782
34.4 Technical Description of the Laplace Program	783
34.5 User Description of the Laplace Operating System	784
34.6 Linear Simultaneous Equations	785
34.6.1 A Calculation Example	788
34.6.2 Technical Description of the Linear Simultaneous Equation Program	789
34.6.3 User Description of the Linear Simultaneous Equation Program	790
34.7 An Example Application	792
34.8 Summary	793
34.9 Bibliography	793
35 The RS/6000 SP System: A Scalable Parallel Cluster	794
35.1 Dual Personalities	794
35.2 SP System Architecture	796
35.3 SP System Structure	800
35.3.1 SP Communications Services	802
35.3.2 SP System Management	804
35.3.3 SP Globalized Resources	806
35.3.4 SP Availability Services	809
35.3.5 SP Programming Model and Environment	810
35.4 Concluding Remarks	812
35.5 Bibliography	813
36 A Scalable and Highly Available Clustered Web Server	815
36.1 Introduction	815
36.1.1 The Internet and the Need for Clustered Web Servers	816
36.1.2 Availability	816
36.1.3 Scalability	817
36.2 Web Servers and Dynamic Content	818
36.2.1 Introduction	818
36.2.2 Static Files on the Web	818
36.2.3 Common Gateway Interface	818

36.2.4	Web Server Application Programming Interfaces	819
36.2.5	FastCGI	820
36.2.6	Servlets	822
36.2.7	Summary	823
36.3	Fine-Grain Load Balancing	823
36.3.1	Introduction	823
36.3.2	Domain Name System (DNS)	823
36.3.3	Round-Robin DNS	824
36.3.4	Load Imbalances with Round-Robin DNS	825
36.3.5	Packet Forwarding for Fine-Grain Load Balancing	825
36.3.6	Summary	826
36.4	Shared Filesystems and Scalable I/O	827
36.4.1	Introduction	827
36.4.2	Shared Fileservers	828
36.4.3	Wide Striping	829
36.4.4	Scalable I/O - Virtual Shared Disk Architecture	830
36.4.5	Real-Time Support for Multimedia Content	831
36.4.6	Summary	831
36.5	Scalable Database Access on the Web	832
36.5.1	Introduction	832
36.5.2	On-Line Commerce and Databases	832
36.5.3	Connection Management for Scalability	832
36.5.4	Java Database Connectivity (JDBC)	834
36.5.5	Caching	834
36.5.6	Parallel Databases	835
36.5.7	Advanced Metadata Management	835
36.5.8	Summary	837
36.6	High Availability	837
36.6.1	Introduction	837
36.6.2	High Availability Infrastructure	837
36.6.3	Web Server and Router Recovery	839
36.6.4	Filesystem and I/O System Recovery	841
36.6.5	Database Recovery	841
36.6.6	Summary	841
36.7	Conclusions	842
36.8	Bibliography	843
	Index	845