

Parallel Computer Architecture

A Hardware/Software Approach

David E. Culler Jaswinder Pal Singh

with Anoop Gupta



MORGAN KAUFMANN PUBLISHERS, INC.
An Imprint of Elsevier
San Francisco, California

Contents

Foreword ix

Preface xxi

1 Introduction 1

- 1.1 Why **Parallel Architecture** 4
 - 1.1.1 Application Trends 6
 - 1.1.2 Technology Trends 12
 - 1.1.3 Architectural Trends 14
 - 1.1.4 Supercomputers 21
 - 1.1.5 Summary 23
- 1.2 Convergence of **Parallel Architectures** 25
 - 1.2.1 Communication **Architecture** 25
 - 1.2.2 Shared Address Space 28
 - 1.2.3 Message Passing 37
 - 1.2.4 Convergence 42
 - 1.2.5 Data **Parallel Processing** 44
 - 1.2.6 Other **Parallel Architectures** 47
 - 1.2.7 **A Generic Parallel Architecture** 50
- 1.3 Fundamental Design Issues 52
 - 1.3.1 Communication Abstraction 53
 - 1.3.2 Programming Model Requirements 53
 - 1.3.3 Communication and Replication 58
 - 1.3.4 Performance 59
 - 1.3.5 Summary 63
- 1.4 Concluding Remarks 63
- 1.5 Historical References 66
- 1.6 Exercises 70

2 Parallel Programs 75

- 2.1 Parallel Application Case Studies 76
 - 2.1.1 Simulating Ocean Currents 77
 - 2.1.2 Simulating the Evolution of Galaxies 78
 - 2.1.3 Visualizing Complex Scenes Using Ray Tracing 79
 - 2.1.4 Mining Data for Associations 80
- 2.2 The Parallelization Process 81
 - 2.2.1 Steps in the Process 82
 - 2.2.2 Parallelizing Computation versus Data 90
 - 2.2.3 Goals of the Parallelization Process 91
- 2.3 Parallelization of an Example Program 92
 - 2.3.1 The Equation Solver Kernel 92
 - 2.3.2 Decomposition 93
 - 2.3.3 Assignment 98
 - 2.3.4 Orchestration under the Data Parallel Model 99
 - 2.3.5 Orchestration under the Shared Address Space Model 101
 - 2.3.6 Orchestration under the Message-Passing Model 108
- 2.4 Concluding Remarks 116
- 2.5 Exercises 117

3 Programming for Performance 121

- 3.1 Partitioning for Performance 123
 - 3.1.1 Load Balance and Synchronization Wait Time 123
 - 3.1.2 Reducing Inherent Communication 131
 - 3.1.3 Reducing the Extra Work 135
 - 3.1.4 Summary 136
- 3.2 Data Access and Communication in a Multimemory System 137
 - 3.2.1 A Multiprocessor as an Extended Memory Hierarchy 138
 - 3.2.2 Artifactual Communication in the Extended Memory Hierarchy 139
 - 3.2.3 Artifactual Communication and Replication: The Working Set Perspective 140
- 3.3 Orchestration for Performance 142
 - 3.3.1 Reducing Artifactual Communication 142
 - 3.3.2 Structuring Communication to Reduce Cost 150
- 3.4 Performance Factors from the Processor's Perspective 156
- 3.5 The Parallel Application Case Studies: An In-Depth Look 160
 - 3.5.1 Ocean 161
 - 3.5.2 Barnes-Hut 166
 - 3.5.3 Raytrace 174
 - 3.5.4 Data Mining 178

3.6	Implications for Programming Models	182
3.6.1	Naming	184
3.6.2	Replication	184
3.6.3	Overhead and Granularity of Communication	186
3.6.4	Block Data Transfer	187
3.6.5	Synchronization	188
3.6.6	Hardware Cost and Design Complexity	188
3.6.7	Performance Model	189
3.6.8	Summary	189
3.7	Concluding Remarks	190
3.8	Exercises	192
4	Workload-Driven Evaluation	199
4.1	Scaling Workloads and Machines	202
4.1.1	Basic Measures of Multiprocessor Performance	202
4.1.2	Why Worry about Scaling?	204
4.1.3	Key Issues in Scaling	206
4.1.4	Scaling Models and Speedup Measures	207
4.1.5	Impact of Scaling Models on the Equation Solver Kernel	211
4.1.6	Scaling Workload Parameters	213
4.2	Evaluating a Real Machine	215
4.2.1	Performance Isolation Using Microbenchmarks	215
4.2.2	Choosing Workloads	216
4.2.3	Evaluating a Fixed-Size Machine	221
4.2.4	Varying Machine Size	226
4.2.5	Choosing Performance Metrics	228
4.3	Evaluating an Architectural Idea or Trade-off	231
4.3.1	Multiprocessor Simulation	233
4.3.2	Scaling Down Problem and Machine Parameters for Simulation	234
4.3.3	Dealing with the Parameter Space: An Example Evaluation	238
4.3.4	Summary	243
4.4	Illustrating Workload Characterization	243
4.4.1	Workload Case Studies	244
4.4.2	Workload Characteristics	253
4.5	Concluding Remarks	262
4.6	Exercises	263
5	Shared Memory Multiprocessors	269
5.1	Cache Coherence	273
5.1.1	The Cache Coherence Problem	273
5.1.2	Cache Coherence through Bus Snooping	277

5.2	Memory Consistency	283
5.2.1	Sequential Consistency	286
5.2.2	Sufficient Conditions for Preserving Sequential Consistency	289
5.3	Design Space for Snooping Protocols	291
5.3.1	A Three-State (MSI) Write-Back Invalidation Protocol	293
5.3.2	A Four-State (MESI) Write-Back Invalidation Protocol	299
5.3.3	A Four-State (Dragon) Write-Back Update Protocol	301
5.4	Assessing Protocol Design Trade-offs	305
5.4.1	Methodology	306
5.4.2	Bandwidth Requirement under the MESI Protocol	307
5.4.3	Impact of Protocol Optimizations	311
5.4.4	Trade-Offs in Cache Block Size	313
5.4.5	Update-Based versus Invalidation-Based Protocols	329
5.5	Synchronization	334
5.5.1	Components of a Synchronization Event	335
5.5.2	Role of the User and System	336
5.5.3	Mutual Exclusion	337
5.5.4	Point-to-Point Event Synchronization	352
5.5.5	Global (Barrier) Event Synchronization	353
5.5.6	Synchronization Summary	358
5.6	Implications for Software	359
5.7	Concluding Remarks	366
5.8	Exercises	367

6 Snoop-Based Multiprocessor Design 377

6.1	Correctness Requirements	378
6.2	Base Design: Single-Level Caches with an Atomic Bus	380
6.2.1	Cache Controller and Tag Design	381
6.2.2	Reporting Snoop Results	382
6.2.3	Dealing with Write Backs	384
6.2.4	Base Organization	385
6.2.5	Nonatomic State Transitions	385
6.2.6	Serialization	388
6.2.7	Deadlock	390
6.2.8	Livelock and Starvation	390
6.2.9	Implementing Atomic Operations	391
6.3	Multilevel Cache Hierarchies	393
6.3.1	Maintaining Inclusion	394
6.3.2	Propagating Transactions for Coherence in the Hierarchy	396
6.4	Split-Transaction Bus	398
6.4.1	An Example Split-Transaction Design	400
6.4.2	Bus Design and Request-Response Matching	400

6.4.3	Snoop Results and Conflicting Requests	402
6.4.4	Flow Control	404
6.4.5	Path of a Cache Miss	404
6.4.6	Serialization and Sequential Consistency	406
6.4.7	Alternative Design Choices	409
6.4.8	Split-Transaction Bus with Multilevel Caches	410
6.4.9	Supporting Multiple Outstanding Misses from a Processor	413
6.5	Case Studies: SGI Challenge and Sun Enterprise 6000	415
6.5.1	SGI Powerpath-2 System Bus	417
6.5.2	SGI Processor and Memory Subsystems	420
6.5.3	SGI I/O Subsystem	422
6.5.4	SGI Challenge Memory System Performance	424
6.5.5	Sun Gigaplane System Bus	424
6.5.6	Sun Processor and Memory Subsystem	427
6.5.7	Sun I/O Subsystem	429
6.5.8	Sun Enterprise Memory System Performance	429
6.5.9	Application Performance	429
6.6	Extending Cache Coherence	433
6.6.1	Shared Cache Designs	434
6.6.2	Coherence for Virtually Indexed Caches	437
6.6.3	Translation Lookaside Buffer Coherence	439
6.6.4	Snoop-Based Cache Coherence on Rings	441
6.6.5	Scaling Data and Snoop Bandwidth in Bus-Based Systems	445
6.7	Concluding Remarks	446
6.8	Exercises	446

7 Scalable Multiprocessors 453

7.1	Scalability	456
7.1.1	Bandwidth Scaling	457
7.1.2	Latency Scaling	460
7.1.3	Cost Scaling	461
7.1.4	Physical Scaling	462
7.1.5	Scaling in a Generic Parallel Architecture	467
7.2	Realizing Programming Models	468
7.2.1	Primitive Network Transactions	470
7.2.2	Shared Address Space	473
7.2.3	Message Passing	476
7.2.4	Active Messages	481
7.2.5	Common Challenges	482
7.2.6	Communication Architecture Design Space	485
7.3	Physical DMA	486
7.3.1	Node-to-Network Interface	486
7.3.2	Implementing Communication Abstractions	488

7.3.3	Case Study: nCUBE/2	488
7.3.4	Typical LAN Interfaces	490
7.4	User-Level Access	491
7.4.1	Node-to-Network Interface	491
7.4.2	Case Study: Thinking Machines CM-5	493
7.4.3	User-Level Handlers	494
7.5	Dedicated Message Processing	496
7.5.1	Case Study: Intel Paragon	499
7.5.2	Case Study: Meiko CS-2	503
7.6	Shared Physical Address Space	506
7.6.1	Case Study: CRAY T3D	508
7.6.2	Case Study: CRAY T3E	512
7.6.3	Summary	513
7.7	Clusters and Networks of Workstations	513
7.7.1	Case Study: Myrinet SBUS Lamai	516
7.7.2	Case Study: PCI Memory Channel	518
7.8	Implications for Parallel Software	522
7.8.1	Network Transaction Performance	522
7.8.2	Shared Address Space Operations	527
7.8.3	Message-Passing Operations	528
7.8.4	Application-Level Performance	531
7.9	Synchronization	538
7.9.1	Algorithms for Locks	538
7.9.2	Algorithms for Barriers	542
7.10	Concluding Remarks	548
7.11	Exercises	548
8	Directory-Based Cache Coherence	553
8.1	Scalable Cache Coherence	558
8.2	Overview of Directory-Based Approaches	559
8.2.1	Operation of a Simple Directory Scheme	560
8.2.2	Scaling	564
8.2.3	Alternatives for Organizing Directories	565
8.3	Assessing Directory Protocols and Trade-Offs	571
8.3.1	Data Sharing Patterns for Directory Schemes	571
8.3.2	Local versus Remote Traffic	578
8.3.3	Cache Block Size Effects	579
8.4	Design Challenges for Directory Protocols	579
8.4.1	Performance	584
8.4.2	Correctness	589

8.5	Memory-Based Directory Protocols: The SGI Origin System	596
8.5.1	Cache Coherence Protocol	597
8.5.2	Dealing with Correctness Issues	604
8.5.3	Details of Directory Structure	609
8.5.4	Protocol Extensions	610
8.5.5	Overview of the Origin2000 Hardware	612
8.5.6	Hub Implementation	614
8.5.7	Performance Characteristics	618
8.6	Cache-Based Directory Protocols: The Sequent NUMA-Q	622
8.6.1	Cache Coherence Protocol	624
8.6.2	Dealing with Correctness Issues	632
8.6.3	Protocol Extensions	634
8.6.4	Overview of NUMA-Q Hardware	635
8.6.5	Protocol Interactions with SMP Node	637
8.6.6	IQ-Link Implementation	639
8.6.7	Performance Characteristics	641
8.6.8	Comparison Case Study: The HAL S1 Multiprocessor	643
8.7	Performance Parameters and Protocol Performance	645
8.8	Synchronization	648
8.8.1	Performance of Synchronization Algorithms	649
8.8.2	Implementing Atomic Primitives	651
8.9	Implications for Parallel Software	652
8.10	Advanced Topics	655
8.10.1	Reducing Directory Storage Overhead	655
8.10.2	Hierarchical Coherence	659
8.11	Concluding Remarks	669
8.12	Exercises	672

9 Hardware/Software Trade-Offs 679

9.1	Relaxed Memory Consistency Models	681
9.1.1	The System Specification	686
9.1.2	The Programmer's Interface	694
9.1.3	The Translation Mechanism	698
9.1.4	Consistency Models in Real Multiprocessor Systems	698
9.2	Overcoming Capacity Limitations	700
9.2.1	Tertiary Caches	700
9.2.2	Cache-Only Memory Architectures (COMA)	701
9.3	Reducing Hardware Cost	705
9.3.1	Hardware Access Control with a Decoupled Assist	707
9.3.2	Access Control through Code Instrumentation	707
9.3.3	Page-Based Access Control: Shared Virtual Memory	709
9.3.4	Access Control through Language and Compiler Support	721

9.4	Putting It All Together: A Taxonomy and Simple COMA	724
9.4.1	Putting It All Together: Simple COMA and Stache	726
9.5	Implications for Parallel Software	729
9.6	Advanced Topics	730
9.6.1	Flexibility and Address Constraints in CC-NUMA Systems	730
9.6.2	Implementing Relaxed Memory Consistency in Software	732
9.7	Concluding Remarks	739
9.8	Exercises	740
10	Interconnection Network Design	749
10.1	Basic Definitions	750
10.2	Basic Communication Performance	755
10.2.1	Latency	755
10.2.2	Bandwidth	761
10.3	Organizational Structure	764
10.3.1	Links	764
10.3.2	Switches	767
10.3.3	Network Interfaces	768
10.4	Interconnection Topologies	768
10.4.1	Fully Connected Network	768
10.4.2	Linear Arrays and Rings	769
10.4.3	Multidimensional Meshes and Tori	769
10.4.4	Trees	772
10.4.5	Butterflies	774
10.4.6	Hypercubes	778
10.5	Evaluating Design Trade-Offs in Network Topology	779
10.5.1	Unloaded Latency	780
10.5.2	Latency under Load	785
10.6	Routing	789
10.6.1	Routing Mechanisms	789
10.6.2	Deterministic Routing	790
10.6.3	Deadlock Freedom	791
10.6.4	Virtual Channels	795
10.6.5	Up*-Down* Routing	796
10.6.6	Turn-Model Routing	797
10.6.7	Adaptive Routing	799
10.7	Switch Design	801
10.7.1	Ports	802
10.7.2	Internal Datapath	802
10.7.3	Channel Buffers	804

10.7.4	Output Scheduling	808
10.7.5	Stacked Dimension Switches	810
10.8	Flow Control	811
10.8.1	Parallel Computer Networks versus LANs and WANs	811
10.8.2	Link-Level Flow Control	813
10.8.3	End-to-End Flow Control	816
10.9	Case Studies	818
10.9.1	CRAY T3D Network	818
10.9.2	IBM SP-1, SP-2 Network	820
10.9.3	Scalable Coherent Interface	822
10.9.4	SGI Origin Network	825
10.9.5	Myricom Network	826
10.10	Concluding Remarks	827
10.11	Exercises	828
11	Latency Tolerance	831
11.1	Overview of Latency Tolerance	834
11.1.1	Latency Tolerance and the Communication Pipeline	836
11.1.2	Approaches	837
11.1.3	Fundamental Requirements, Benefits, and Limitations	840
11.2	Latency Tolerance in Explicit Message Passing	847
11.2.1	Structure of Communication	848
11.2.2	Block Data Transfer	848
11.2.3	Precommunication	848
11.2.4	Proceeding Past Communication in the Same Thread	850
11.2.5	Multithreading	850
11.3	Latency Tolerance in a Shared Address Space	851
11.3.1	Structure of Communication	852
11.4	Block Data Transfer in a Shared Address Space	853
11.4.1	Techniques and Mechanisms	853
11.4.2	Policy Issues and Trade-Offs	854
11.4.3	Performance Benefits	856
11.5	Proceeding Past Long-Latency Events	863
11.5.1	Proceeding Past Writes	864
11.5.2	Proceeding Past Reads	868
11.5.3	Summary	876
11.6	Precommunication in a Shared Address Space	877
11.6.1	Shared Address Space without Caching of Shared Data	877
11.6.2	Cache-Coherent Shared Address Space	879
11.6.3	Performance Benefits	891
11.6.4	Summary	896

- 11.7 Multithreading in a Shared Address Space 896
 - 11.7.1 Techniques and Mechanisms 898
 - 11.7.2 Performance Benefits 910
 - 11.7.3 Implementation Issues for the Blocked Scheme 914
 - 11.7.4 Implementation Issues for the Interleaved Scheme 917
 - 11.7.5 Integrating Multithreading with Multiple-Issue Processors 920
- 11.8 Lockup-Free Cache Design 922
- 11.9 Concluding Remarks 926
- 11.10 Exercises 927

12 Future Directions 935

- 12.1 Technology and Architecture 936
 - 12.1.1 Evolutionary Scenario 937
 - 12.1.2 Hitting a Wall 940
 - 12.1.3 Potential Breakthroughs 944
- 12.2 Applications and System Software 955
 - 12.2.1 Evolutionary Scenario 955
 - 12.2.2 Hitting a Wall 960
 - 12.2.3 Potential Breakthroughs 961

Appendix: Parallel Benchmark Suites 963

- A.1 ScaLapack 963
- A.2 TPC 963
- A.3 SPLASH 965
- A.4 NAS Parallel Benchmarks 966
- A.5 PARBENCH 967
- A.6 Other Ongoing Efforts 968

References 969

Index 995