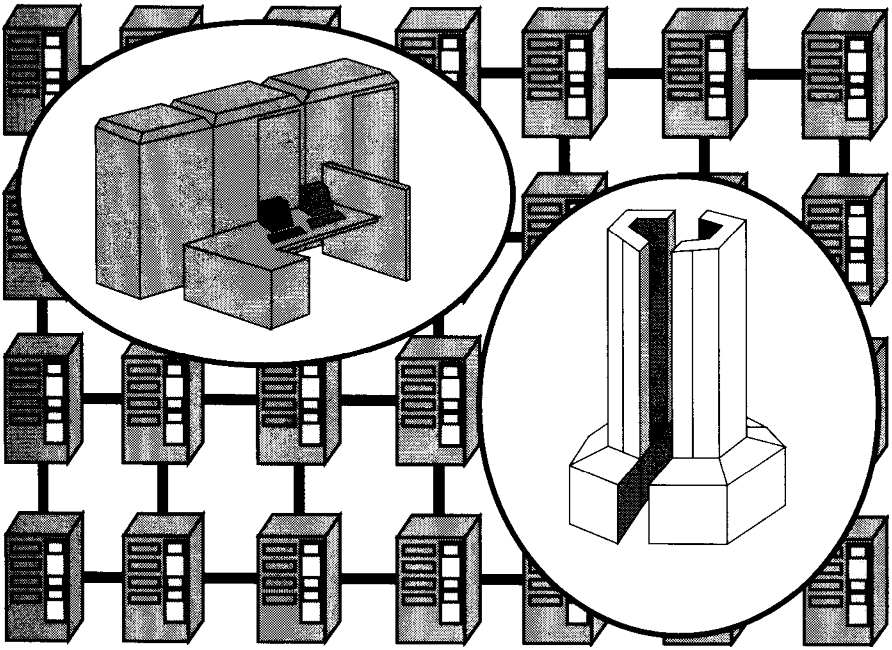
3700

681.31.02
P481.
Ed.2

In Search of Clusters
Second Edition

Gregory F. Pfister



To join a Prentice Hall PTR Internet mailing list, point to http://www.prenhall.com/mail_lists.

ISBN 0-13-899709-8

Prentice Hall PTR
Upper Saddle River, NJ 07458

Table of Contents

List of Figures - xiii

List of Tables - xvii

Preface to the Second Edition - - - - - - - - - - - - xix

Preface- xxiii

PART 1 WHAT ARE CLUSTERS, AND WHY USE THEM? - - - - - 1

Chapter 1 Introduction - 3

1.1 Working Harder 4

1.2 Working Smarter 5

1.3 Getting Help 6

1.4 The Road to Lowly Parallel Processing 10

1.5 A Neglected Paradigm 14

1.6 What is to Come 17

Chapter 2 **Examples - 19**

2.1 Beer & Subpoenas 19

2.2 Serving the Web 21

2.3 The Farm 23

2.4 Fermilab 25

2.5 Other Compute Clusters 28

2.6 Full System Clusters 31

2.6.1 DEC OpenVMS Cluster 31

2.6.2 Tandem Himalaya 34

2.6.3 IBM S/390 Parallel Sysplex 36

2.6.4 Future Candidates 40

2.7 Cluster Software Products 40

2.7.1 Batch and Login 41

2.7.2 Database Systems 42

2.7.3 Parallel Programming Facilities 43

2.7.4 Operating Systems 45

2.8 Basic (Availability) Clusters 47

2.9 Not the End 51

Chapter 3 **Why Clusters? -** *53*

3.1 The Standard Litany 54

3.2 Why Now? 56

3.3 Why Not Now? 58

3.4 Commercial Node Performance 60

3.5 The Need for High Availability 64

Chapter 4 **Definition, Distinctions, and Initial Comparisons- - - 71**

4.1 Definition 72

4.2 Distinction from Parallel Systems 72

4.3 Distinctions from Distributed Systems 75

4.3.1 Internal Anonymity 75

4.3.2 Peer Relationship 76

4.3.3 Clusters within a Client-Server Organization 78

4.3.4 Clusters and Three-Tiered Applications 80

4.4 Concerning “Single System Image” 81

4.5 Other Comparisons 81

4.6 Reactions 83

PART 2 HARDWARE - 85

Chapter 5 A Cluster Bestiary - - - - - - - - - - - - - - - - - - 87

5.1 Exposed vs. Enclosed 90

5.2 “Glass-House” vs. “Campus-Wide” Cluster 93

5.3 Cluster Hardware Structures 97

5.3.1 The Four Categories 97
5.3.2 I/O-Attached Message-Based 99
5.3.3 I/O-Attached Shared Storage 100
5.3.4 Memory-Attached Shared Storage 100
5.3.5 Memory-Attached Message-Based 101
5.3.6 Discussion 10 1

5.4 Communication Requirements 104

5.4.1 Computational/y Intensive Workloads 104
5.4.2 //O-Intensive Workloads 105
5.4.3 What Can Be Done? 107

5.5 Cluster Acceleration Techniques 108

5.5.1 Tandem’s ServerNet System Area Network (SAN) 108
5.5.2 *Intel’s Virtual Interface Architecture (VIA) 113*
5.5.3 DEC Memory Channel 118
5.5.4 *The S/390 Parallel Sysplex Coupling Facility 121*
5.5.5 High Function, Low Function 127

Chapter 6 Symmetric Multiprocessors - - - - - - - - - - - - - 131

6.1 What is an SMP? 132

6.2 What is a Cache, and Why Is It Necessary? 135

6.2.1 *The Processor-Memory Speed Problem 136*
6.2.2 *Why Faster Main Memory A/one Doesn’t Work 138*
6.2.3 *Locality of Reference and Caches 140*
6.2.4 Level 2 Caches 143
6.2.5 *Cache Lines 144*
6.2.6 Now, Let’s Simplify Things 145
6.2.7 *The Cache as a Messy Desk 146*
6.2.8 Caches *do it at Random 148*
6.2.9 An Aside for the *Incurably Precise 150*

6.3 Memory Contention 152

6.3.1 Great Big Caches 153
6.3.2 Line Size, Memory Banks, and Pipelining 155

6.4 Cache Coherence 158

6.4.1 *“Fixing” it in Software 161*
6.4.2 *Central Directory 162*
6.4.3 *Snoopy Bus 164*
6.4.4 *Snoopy Bus and Switch 170*
6.4.5 *Commercial Processing: Cross-Interrogation and Shared Caches 171*

6.5 Sequential and Other Consistencies 174

6.6 Input/Output 182

6.7 Summary 184

Chapter 7 NUMA and Friends - - - - - - - - - - - - - - - - - 185

7.1 UMA, NUMA, NORMA, and CC-NUMA 186

7.2 How CC-NUMA Works 188

7.2.1 Two CC-NUMA Implementation Styles 192

7.3 The “N” in CC-NUMA 199

7.4 Software Implications 200

7.5 Other CC-NUMA Implications 203

7.5.1 Single Operating System 203

7.5.2 CC-NUJMA and High Availability 203

7.6 Is “NUMA” Inevitable? 205

7.6.1 The Four Basic Arguments 206

7.6.2 The Overwhelming Memory Argument 207

7.6.3 So, Is NUMA Inevitable? 208

7.7 Great Big CC-NUMA 209

7.8 Simple COMA 210

PART 3 SOFTWARE - 217

Chapter 8 Workloads - 219

8.1 Why Discuss Workloads? 220

8.2 Serial: Throughput 220

8.2.1 Batch 222

8.2.2 Interactive 223

8.2.3 Multijob Parallel 224

8.3 Parallel 224

8.3.1 Large Effort Justified 225

8.3.2 Minimal Effort Justified 227

8.4 Amdahl’s Law 231

8.5 The Point of All This 233

Chapter 9 Basic Programming Models and Issues- - - - - - - - - 235

9.1 What is a Programming Model? 236

9.2 The Sample Problem 239

9.3 Uniprocessor 241

9.4 Shared Memory 247

9.4.1 Races and Locks 249

9.4.2 Serialization 253

9.4.3 Consistency and Chaos 256

9.4.4 Many Locks and Deadly Embraces (Deadlocks) 260

9.4.5 Alternation Schemes 264

9.4.6 What Went Wrong? 266
9.4.7 The Wavefront (Hyperplane) Method 269
9.4.8 Let "forall" do the Work 270
9.4.9 Chunking and Blocking 271
9.4.10 Load Balancing and a Global Queue 276
9.4.11 Concerning Global Queues 279
9.4.12 Not the End, Again 281

9.5 Message-Passing 282
9.5.1 Jacobi/Seidel/Chaotic "Mixed" Relaxation 283
9.5.2 Implicit Synchronization and Alternation Schemes 287
9.5.3 Wavefronts and Multiple Waves 288
9.5.4 Wavefronts in Strips 291
9.5.5 Communication Overhead in Message-Passing 292
9.5.6 Clusters of SMPs 294

9.6 CC-NUMA 295

9.7 SIMD and All That 299
9.7.7 SIMD, MIMD, and SPMD 299
9.7.2 Distributed Virtual Memory and COMA (ALLCACHE) 301
9.7.3 Dataflow and Reduction 302
9.7.4 Linda 303
9.7.5 General Comments on Other Programming Models 303

9.8 Importance 304
9.8.1 Ease of Use 305
9.8.2 Portability 307

Chapter 10 Commercial Programming Models- --------- 311

10.1 Small N vs. Large N 312

10.2 Small N Programming Models 314
10.2.1 Threads 314
10.2.2 User-Mode Threads 315
10.2.3 SMP with Threads 320
10.2.4 SMP with Shared Segments 322
10.2.5 OS Interprocess 324

10.3 Large-N I/O Programming Models 324
10.3.1 Global I/O 324
10.3.2 Local I/O 326

10.4 Large-N Processor-Memory Models 326
10.4.1 Message-Passing 327
10.4.2 Remote Procedure Calls (RPC and DCE RPC) 327
10.4.3 Global Locks 328
10.4.4 The SMP Variation (*SMP) 329

10.5 Shared Disk or not Shared Disk? 330
10.5.1 Physically Shared Disks 330
10.5.2 I/O Shipping 333
10.5.3 Function Shipping 335
10.5.4 Cache Coherence Again, and IBM's Parallel Sysplex 336

10.5.5 Further Comparisons 342
10.5.6 The Ultimate Direction 344

Chapter 11 Single System Image----------------347

11.1 Single System Image Boundaries 348
11.2 Single System Image Levels 350
11.3 The Application and Subsystem Levels 352
11.3.1 The Application Level 352
11.3.2 The Subsystem Level 353
11.3.3 The File System Layer 355
11.3.4 The Toolkit Layer 356
11.4 The Operating System Kernel Levels 358
11.4.1 The Kernel API Layer 359
11.4.2 The Level of the Kernel Interfaces 366
11.4.3 The Distributed Virtual Memory Level 367
11.4.4 The Microkernel Level 367
11.5 Hardware Levels 369
11.6 SSI and System Management 370
11.6.1 Over-Kernel SSI System Management 371
11.6.2 Kernel-Level SSI System Management 373

PART 4 SYSTEMS - 377

Chapter 12 High Availability- 379

12.1 What Does “High Availability” Mean? 381
12.1.1 How High is “High”? 383
12.1.2 Facets of Availability 385
12.1.3 Outage Duration 392
12.1.4 So, OK, What Really is High Availability? 393
12.2 The Basic Idea: Failover 394
12.2.1 First, Do No Harm 395
12.2.2 Avoiding Planned Outages 395
12.2.3 Some Questions 397
12.3 Resources 398
12.4 Failing Over Data 400
12.4.7 Replication and Switchover 401
12.4.2 Avoiding Toxic Data: Transactions 406
12.5 Failing Over Communications 411
12.6 Towards Instant Failover 414
12.7 Failover to Where? 416
12.8 Lock Data Reconstruction 418
12.9 Heartbeats, Events, and Failover Processing 419
12.10 System Structure 424
12.10.1 IBM’s RS/6000 Cluster Technology (“Phoenix’) 424

12.10.2 Microsoft's Cluster Services *("Wolfpack")* 430

12.11 Related Issues 434

12.11.1 The Link To Scaling: Cost 435

12.11.2 Disaster Recovery 436

12.11.3 Hardware Fault Tolerance & Repair Techniques 440

12.11.4 Lowly Available Systems SMP CC-NUMA 445

Chapter 13 Symmetric Multiprocessors, "NUMA," and Clusters- 453

13.1 Preliminaries 454

13.1.1 What's Being Compared? 455

13.1.2 Good Twin, Evil Twin 456

13.1.3 Architecture vs. Implementation 458

13.2 Performance 459

13.2.1 Workload Characteristics 460

13.2.2 Workload-Relative Performance 464

13.2.3 Projecting to a Single Dimension 467

13.3 Cost 469

13.3.1 Does Processor Complex Cost *Matter? 469*

13.3.2 The Cost of Scaling Up SMPs 472

13.3.3 The Production Volume Issue 476

13.4 High Availability 477

13.5 Other Issues 478

13.5.1 Memory Utilization 478

13.5.2 Load Balancing 480

13.5.3 Application and Subsystem Support 481

13.5.4 Administration 481

13.5.5 Programming Model 482

13.6 Partitioning 482

13.7 Conclusion 486

Chapter 14 Why We Need the Concept of *Cluster* - - - - - - - - *487*

14.1 Benchmarks 488

14.1.1 The TPC Benchmarks 488

14.1.2 Parallel Execution 489

14.1.3 Difficulties 492

14.1.4 Why Am I Doing This? 496

14.2 Development Directions 498

14.3 Confusion of Issues 500

14.4 The Lure of Large Numbers 501

Chapter 15 Conclusion ---------------------- 505

15.1 Cluster Operating Systems 507

15.2 Exploitation 507

15.3 Standards 508

15.3.1 Common Functions 509

15.3.2 Intra-Cluster Communication 509
15.3.3 Transactions 510
15.3.4 Comments on Standards 510

15.4 Software Pricing 511

15.5 What About 201 O? 512

15.6 Coda: The End of Parallel Computer Architecture 515

Annotated Bibliography - - - - - - - - - - - - - - - - 517

Index - ***549***

About the Author - - - - - - - - - - - - - - - - - - - 575