Contents

Foreword by Grady Booch xix

Foreword by Paul Clements xxi

Preface xxv

Part I: Introduction and Overview

Chapter 1: Introduction 3

1.1 Software and Performance 3

1.1.1 Responsiveness 4

1.1.2 Scalability 5

1.2 The Importance of Performance 6

1.2.1 Consequences of Performance Failures 9

1.2.2 Causes of Performance Failures 9

1.2.3 Getting It Right 12

1.3 How Should You Manage Performance? 14

1.3.1 Reactive Performance Management 14

1.3.2 Proactive Performance Management 15

1.4 Software Performance Engineering 16

1.4.1 SPE Modeling Strategies 18

1.4.2 SPE Models 20

1.5 SPE for Object-Oriented Systems 21

1.5.1 What Does It Cost? 22

1.5.2 What Do You Need? 23

1.6 Summary 24

Chapter 2: SPE Quick View 27

2.1 SPE Process for Object-Oriented Systems 27

2.2 Case Study 32
2.2.1 Assess Performance Risk (Step 1) 33
2.2.2 Identify Critical Use Cases (Step 2) 33
2.2.3 Select Key Performance Scenarios (Step 3) 33
2.2.4 Establish Performance Objectives (Step 4) 35
2.2.5 Construct Performance Models (Step 5) 35
2.2.6 Determine Software Resource Requirements (Step 6) 36
2.2.7 Add Computer Resource Requirements (Step 7) 37
2.2.8 Evaluate the Models (Step 8) 39
2.2.9 Verify and Validate the Models (Step 9) 39
2.3 SPE in the Unified Software Process 40
2.4 Performance Solutions 41
2.4.1 Performance Principles 42
2.4.2 Performance Patterns 42
2.4.3 Performance Antipatterns 42
2.4.4 Implementation Solutions 43
2.5 Summary 44

Chapter 3: SPE and the UML 47
3.1 Overview 47
3.2 Extending the UML 48
3.2.1 Stereotypes 48
3.2.2 Tagged Values 49
3.2.3 Constraints 49
3.3 Use Cases and Scenarios 50
3.3.1 Use Cases 50
3.3.2 Scenarios 52
3.4 Extensions to Sequence Diagram Notation 55
3.4.1 Instance Decomposition 55
3.4.2 Looping, Alternation, and References 56
3.5 Specifying Time 59
3.5.1 Timing Marks 60
3.5.2 Time Expressions 60
3.5.3 Timing Constraints 60
3.5.4 Time in Sequence Diagrams 60
3.6 Concurrency 62
3.6.1 Threads and Processes 62
3.6.2 Coregions 63

3.6.3 Parallel Composition 64
3.6.4 Synchronization 65
3.7 Summary 67

Part II: SPE Models

Chapter 4: Software Execution Models 71
4.1 Purpose 71
4.2 Representing Software Execution Models 72
4.2.1 Execution Graphs 73
4.2.2 Execution Graph Restrictions 77
4.3 Model Solutions 79
4.3.1 Basic Solution Algorithms 80
4.3.2 More Advanced Solution Techniques 83
4.4 Analysis Procedures 83
4.5 Execution Graphs from Sequence Diagrams 88
4.6 ICAD Case Study 89
4.6.1 Architecture 1 90
4.6.2 Architecture 2 96
4.6.3 Analysis of Results 98
4.6.4 Architecture 3 98
4.7 Modeling Hints 102
4.8 Summary 103

Chapter 5: Web Applications and Other Distributed Systems 105
5.1 Introduction 105
5.2 Web Applications 106
5.3 Distributed Object Technology 109
5.3.1 Middleware 110
5.3.2 Limitations of Distributed Object Technology 111
5.3.3 Effective Development with Distributed Object Technology . 114
5.4 Modeling Distributed System Interactions 115
5.4.1 Types of System Interactions 115
5.4.2 Software Execution Model Representation 118
5.4.3 Representing Middleware Overhead 120
5.4.4 Software Model Solution Approximations 121
5.5 Example: Web e-Commerce Application 123
5.5.1 Database Scenario 129
5.5.2 Order Process Scenario 129

5.5.3 Example Summary 130
5.6 Modeling Hints 130
5.7 Summary 131

Chapter 6: System Execution Models 133
6.1 Introduction 133
6.2 System Model Basics 135
6.2.1 Performance Metrics 136
6.2.2 Solving the Queueing Model 139
6.2.3 Networks of Queues 141
6.3 Deriving System Model Parameters from Software Model Results . . . 146
6.4 Using the System Model for SPE 150
6.4.1 Advanced System Models 151
6.4.2 Alternate Solution Methods 152
6.4.3 Schedulability 153
6.5 Distributed System Case Study 153
6.5.1 Synchronization Model 160
6.6 Modeling Hints 165
6.7 Summary 166

Part III: Data Collection

Chapter 7: SPE Data Collection 169
7.1 Introduction 169
7.2 SPE Data Requirements 170
7.2.1 Key Performance Scenarios 170
7.2.2 Performance Objectives 172
7.2.3 Execution Environment 174
7.2.4 Software Resource Requirements 174
7.2.5 Computer Resource Requirements 176
7.2.6 Data Gathering Issues 178
7.3 Performance Walkthrough 179
7.3.1 Topics 180
7.3.2 When to Conduct Performance Walkthroughs 182
7.3.3 Example 183
7.3.4 Tips for a Successful Performance Walkthrough 193
7.4 Resource Estimation Techniques 195
7.4.1 Use Measurements 195
7.4.2 Study Measurements 196

7.4.3 Use a Mentor 196
7.4.4 Best-Worst Case Estimates 197
7.4.5 What to Estimate 197
7.4.6 Estimating I/O Requirements 197
7.4.7 Estimating Network Messages 200
7.4.8 Obtaining Computer Resource Requirements 201
7.5 Summary 201

Chapter 8: Software Measurement and Instrumentation 203
8.1 Introduction 203
8.2 What Should You Measure? 204
8.2.1 Workload Data and Data Characteristics 207
8.2.2 Path Characteristics 207
8.2.3 Software Resources and Processing Overhead 207
8.2.4 Computer Resource Usage 208
8.3 Planning for Performance Measurement 208
8.3.1 Key Considerations 208
8.3.2 Performance Benchmarks 210
8.3.3 Designing and Conducting Measurement Studies 210
8.4 Performance Measurement Concepts 214
8.4.1 Terminology 215
8.4.2 Factors That May Affect Measurements 218
8.5 Data Collection Techniques and Tools 221
8.5.1 Data Collection Techniques 222
8.5.2 Measuring SPE Data 224
8.6 Instrumentation 227
8.6.1 Instrumentation Design Considerations 230
8.6.2 Implementation Alternatives 231
8.6.3 Data Reporting 233
8.7 Application Resource Measurement 233
8.8 Summary 236

Part IV: Performance Solutions

Chapter 9: Performance-Oriented Design 241
9.1 Principles for Performance-Oriented Design 241
9.2 Performance Control Principles 242
9.2.1 Performance Objectives Principle 242
9.2.2 Instrumenting Principle 243

9.3 Independent Principles 245
9.3.1 Centering Principle 245
9.3.2 Fixing-Point Principle 247
9.3.3 Locality Principle 249
9.3.4 Processing Versus Frequency Principle 251
9.4 Synergistic Principles 252
9.4.1 Shared Resources Principle 253
9.4.2 Parallel Processing Principle 254
9.4.3 Spread-the-Load Principle 256
9.5 Using the Principles 257
9.6 Summary 259

Chapter 10: Performance Patterns 261
10.1 Overview 261
10.2 Fast Path 263
10.2.1 Problem 264
10.2.2 Solution 264
10.2.3 Benefits 266
10.2.4 Consequences 266
10.3 First Things First 267
10.3.1 Problem 267
10.3.2 Solution 268
10.3.3 Benefits 269
10.3.4 Consequences 269
10.4 Coupling 269
10.4.1 Problem 270
10.4.2 Solution 271
10.4.3 Benefits 272
10.4.4 Consequences 272
10.5 Batching 272
10.5.1 Problem 272
10.5.2 Solution 274
10.5.3 Benefits 275
10.5.4 Consequences 275
10.6 Alternate Routes 276
10.6.1 Problem 276
10.6.2 Solution 277
10.6.3 Benefits 278

10.6.4 Consequences 279
10.7 Flex Time 279
10.7.1 Problem 279
10.7.2 Solution 280
10.7.3 Benefits 281
10.7.4 Consequences 281
10.8 Slender Cyclic Functions 282
10.8.1 Problem 282
10.8.2 Solution 283
10.8.3 Benefits 283
10.8.4 Consequences 284
10.9 Summary 284

Chapter 11: Performance Antipatterns 287
11.1 Overview 287
11.2 The "god" Class 288
11.2.1 Problem 289
11.2.2 Solution 292
11.3 Excessive Dynamic Allocation 293
11.3.1 Problem 293
11.3.2 Solution 295
11.4 Circuitous Treasure Hunt 296
11.4.1 Problem 297
11.4.2 Solution 299
11.5 The One-Lane Bridge 301
11.5.1 Problem 301
11.5.2 Solution 302
11.6 Traffic Jam 304
11.6.1 Problem 305
11.6.2 Solution 305
11.7 Summary 306

Chapter 12: Implementation Solutions 309
12.1 Overview 309
12.2 Performance Tuning 311
12.3 General Performance Solutions 316
12.3.1 Fast Path Speed-Up 316
12.3.2 Improving Scalability 318
12.3.3 Algorithm and Data Structure Choices 322

12.3.4 Time Versus Space Trade-Offs 323
12.3.5 Hardware/Software Platform Dependencies 326
12.4 Performance Solutions for Object-Oriented Software 327
12.4.1 Language-Independent Solutions 327
12.4.2 C++ Solutions 331
12.4.3 Java Solutions 333
12.5 Summary 337

Part V: Applications

Chapter 13: Web Applications 341
13.1 Introduction 341
13.2 Performance Issues 343
13.3 SPE Models for Web Applications 345
13.4 Case Study: Nachtfliegen.com 349
13.4.1 Plan Itinerary Scenario 350
13.4.2 Software Model 356
13.4.3 Hardware/Software Environment 356
13.4.4 Resource Requirements 358
13.4.5 Software Model Solution 360
13.4.6 Performance Improvements 362
13.4.7 System Execution Model 363
13.4.8 Sensitivity and Scalability Analysis 366
13.5 Typical Performance Problems 366
13.6 Summary 371

Chapter 14: Embedded Real-Time Systems 373
14.1 Introduction 373
14.2 Embedded Real-Time Systems Background 374
14.2.1 Timing Requirements 375
14.2.2 Hardware Constraints 378
14.2.3 Real-Time Operating Systems 378
14.2.4 Distributed Systems 379
14.2.5 Database 379
14.3 Performance Issues 379
14.3.1 Response Time and Throughput 380
14.3.2 Schedulability 381
14.4 SPE Models for Embedded Real-Time Systems 384
14.5 Case Study: Telephony Switching 385

14.5.1 Overview 385
14.5.2 Architecture and Design 388
14.6 Typical Performance Problems 396
14.7 Summary 402

Part VI: Making SPE Happen

Chapter 15: The SPE Process 407
15.1 Introduction 407
15.2 The SPE Process 408
15.2.1 Assess Performance Risk 410
15.2.2 Identify Critical Use Cases 411
15.2.3 Select Key Performance Scenarios 412
15.2.4 Establish Performance Objectives 412
15.2.5 Construct Performance Models 413
15.2.6 Determine Software Resource Requirements 415
15.2.7 Add Computer Resource Requirements 415
15.2.8 Evaluate the Models 416
15.2.9 Verify and Validate Models 418
15.3 Late Life Cycle SPE Activities 419
15.3.1 More Detailed Models 420
15.3.2 More Precise Data 421
15.3.3 Performance Testing 421
15.3.4 Baseline Models 422
15.4 Post-Deployment Performance Management 423
15.4.1 Evolutionary Changes 423
15.4.2 Capacity Management 423
15.5 SPE Artifacts 424
15.5.1 Performance Management Plans 425
15.5.2 Performance V&V Plan 427
15.5.3 SPE Configuration Management Plan 427
15.5.4 Performance Drivers 428
15.5.5 Performance Scenarios 428
15.5.6 Performance Objectives 428
15.5.7 Execution Environment Specifications 428
15.5.8 Performance Models 429
15.5.9 Model Results 429
15.5.10 Performance Instrumentation 429
15.5.11 Performance V&V Reports 429

15.5.12 Performance Test Plans 430
15.5.13 Performance Test Results 431
15.6 Integrating SPE Into Your Software Process 431
15.6.1 The Waterfall Model 432
15.6.2 The Spiral Model 432
15.6.3 SPE in the Unified Process 433
15.7 Summary 436

Chapter 16: Implementing SPE 439
16.1 Introduction 439
16.2 Tools 440
16.2.1 Modeling Tools 440
16.2.2 Development Tools 442
16.3 SPE Adoption and Use 444
16.3.1 Experience 444
16.3.2 Key Considerations 448
16.3.3 Pilot Projects 451
16.3.4 Critical Success Factors for Adoption and Use 452
16.4 SPE Implementation Strategies 455
16.4.1 Organizational Issues 455
16.4.2 Who Pays for SPE? 457
16.4.3 Costs 458
16.4.4 Risks 458
16.5 Critical Factors for Successful Projects 459
16.6 SPE Future 461
16.7 Summary 462

Part VII: Appendixes

Appendix A: UML Notation 467
A.1 Use Case Diagrams 468
A.2 Sequence Diagrams 469
A.2.1 Basic Sequence Diagrams 469
A.2.2 Augmented Sequence Diagrams 469
A.3 Deployment Diagrams 471
A.4 Stereotypes, Tagged Values, and Constraints 472
A.4.1 Stereotypes 472
A.4.2 Tagged Values 472
A.4.3 Constraints 472

Appendix B: SPE Modeling Notations 475
B.1 Execution Graph Notation 476
B.1.1 Basic Nodes 476
B.1.2 Synchronization Nodes 477
B.2 Information Processing Graph Notation 478

Bibliography 479

Index 489