Contents

Foreword ix
Preface xi

Chapter One
Introduction to Virtual Machines 1

1.1 Computer Architecture 6
1.2 Virtual Machine Basics 9
1.3 Process Virtual Machines 13
1.4 System Virtual Machines 17
1.5 A Taxonomy 22
1.6 Summary: The Versatility of Virtual Machines 23
1.7 The Rest of the Book 24

Chapter Two
Emulation: Interpretation and Binary Translation 27

2.1 Basic Interpretation 29
2.2 Threaded Interpretation 32
2.3 Predecoding and Direct Threaded Interpretation 34
2.4 Interpreting a Complex Instruction Set 38
2.5 Binary Translation 49
2.6 Code Discovery and Dynamic Translation 52
2.7 Control Transfer Optimizations 64
2.8 Instruction Set Issues 68
2.9 Case Study: Shade and the Role of Emulation During Simulation 77
2.10 Summary: Performance Tradeoffs 80

Chapter Three
Process Virtual Machines 83

3.1 Virtual Machine Implementation 85
3.2 Compatibility 87
3.3 State Mapping 95
3.4 Memory Architecture Emulation 102
3.5 Instruction Emulation 114
3.6 Exception Emulation 119
3.7 Operating System Emulation 128
3.8 Code Cache Management 133
3.9 System Environment 140
3.10 Case Study: FX!32 142
3.11 Summary 145

Chapter Four
Dynamic Binary Optimization 147

4.1 Dynamic Program Behavior 153
4.2 Profiling 156
4.3 Optimizing Translation Blocks 167
4.4 Optimization Framework 180
4.5 Code Reordering 186
4.6 Code Optimizations 201
4.7 Same-ISA Optimization Systems: Special-Case Process Virtual Machines 208
4.8 Summary 218

Chapter Five
High-Level Language Virtual Machine Architecture 221

5.1 The Pascal P-Code Virtual Machine 225
5.2 Object-Oriented High-Level Language Virtual Machines 228
5.3 The Java Virtual Machine Architecture 241
5.4 Completing the Platform: APIs 261
5.5 The Microsoft Common Language Infrastructure: A Flexible High-Level Language Virtual Machine 267
5.6 Summary: Virtual ISA Features 275

Chapter Six
High-Level Language Virtual Machine Implementation 281

6.1 Dynamic Class Loading 284
6.2 Implementing Security 286

6.3 Garbage Collection 294
6.4 Java Native Interface 304
6.5 Basic Emulation 305
6.6 High-Performance Emulation 306
6.7 Case Study: The Jikes Research Virtual Machine 320
6.8 Summary 327

Chapter Seven
Codesigned Virtual Machines 329

7.1 Memory and Register State Mapping 333
7.2 Self-Modifying and Self-Referencing Code 337
7.3 Support for Code Caching 339
7.4 Implementing Precise Traps 344
7.5 Input/Output 351
7.6 Applying Codesigned Virtual Machines 352
7.7 Case Study: Transmeta Crusoe 354
7.8 Case Study: IBM AS/400 357
7.9 Summary 367

Chapter Eight
System Virtual Machines 369

8.1 Key Concepts 373
8.2 Resource Virtualization — Processors 382
8.3 Resource Virtualization — Memory 396
8.4 Resource Virtualization — Input/Output 404
8.5 Performance Enhancement of System Virtual Machines 415
8.6 Case Study: VMware Virtual Platform 426
8.7 Case Study: The Intel VT-x (Vanderpool) Technology 436
8.8 Summary 442

Chapter Nine
Multiprocessor Virtualization 445

9.1 Partitioning of Multiprocessor Systems 445
9.2 Physical Partitioning 455
9.3 Logical Partitioning 458
9.4 Case Study: Cellular Disco System Virtual Machine-Based Partitioning 475
9.5 Virtualization with Different Host and Guest ISAs 485
9.6 Summary 496

Chapter Ten
Emerging Applications 499

10.1 Security 501
10.2 Migration of Computing Environments 520
10.3 Grids: Virtual Organizations 535
10.4 Summary 552

Appendix A
Real Machines 553

A.1 Computer System Hardware 554
A.2 The User ISA: Computation 561
A.3 The System ISA: Resource Management 566
A.4 Operating System Organization 580
A.5 The Operating System Interface 583
A.6 System Initialization 586
A.7 Multiprocessor Architecture 588
A.8 Example Instruction Set Architectures 600

References 613

Index 629