

Contents

Preface xi

Acknowledgements xiii

Part I Getting Started 1

Chapter 1 Fishing for Things to Tune 3

All Roads Lead to the Source 4

When to Use the VTune™ Performance Analyzer 5

 A Development Cycle Approach 5

 A Performance-centric Approach 5

 Some Case Histories 6

How to Find Bottlenecks 8

What Expertise Is Required? 8

Learning to Fish: A Tuning Methodology 10

 Different Fishing Rods and Bait 10

 No Matter What Gear You Choose, You Need a Method 10

 Decide What to Catch: Your Tuning Goals 12

 System-level Tuning 14

 Application-level Tuning 17

 Computer Architecture-level Tuning 19

Recap 20

Chapter 2 Computer and Application Architecture Basics 21

Computer Architecture in a Nutshell	22
The Clock	22
Caches	23
Cache Coherency	25
Cache Line	25
Cache Hit or Miss	25
Dirty Cache	26
Multi-Way Caches Such As 8-Way Cache	26
Replacement Policies Such As LRU	27
Memory Management, Addressing, and TLBs	27
Inside the CPU	28
Processor Differences That Matter	29
Intel XScale® Technology	30
P6 Microarchitecture	30
Intel NetBurst® Microarchitecture	30
EPIC Technology	31
Recurring Themes When Tuning Software	31
Threading and Hyper-Threading Technology (HT Technology)	32
Dynamic Libraries	33
What Lies Ahead	33
Recap	34

Chapter 3 Projects, Activities, GUI, and Command Line 35

A Quick Overview	35
GUI and Command Line Interface	36
Projects and Activities	38
Tuning Environments	39
Creating Projects	39
Graphical User Interface (GUI)	43
The Tuning Browser	43
Sampling Data Views	47
Call Graph Data Views	56
Counter Monitor Data Views	58
Static Module Analysis View	59
Output Window	61
VTune™ Analyzer's Command Line	61
Creating Projects in the Visual Studio .NET Environment	63
Recap	66

Part II Examining Your System and Applications 67

Chapter 4 Hotspot Hunting 101: Sampling 69

- What Is a Hotspot? 71
- Before You Begin Sampling 73
- Types of Sampling 74
- Time-Based Sampling 74
 - Using the Program Example 74
 - Using Your Own or Another Program 75
 - Step-by-Step Procedure 75
 - Which Application Took the Most Time? 77
 - Drilling Down to the Source 81
 - Step-by-Step Procedure 86
 - Drilling Down 88
- Event-Based Sampling 91
 - Step-by-Step Procedure 91
 - How Often Did the *Split Loads* and *Split Stores* Occur? 94
 - Drilling Down to the Source 95
 - Step-by-Step Procedure 99
 - Drilling Down 101
- Sampling Rates 104
- Event Ratios 106
- More About Sampling Events and Event Ratios 107
- Sampling Over Time View 109
 - Configuring Sampling Over Time 112
 - Generating Over Time Views 113
 - Using Zoom 118
 - Saving a Zoomed-in Time Range 120
- Sampling Usage Tips 124
- How Sampling Analysis Works 125
- Recap 126

Chapter 5 Hotspot Hunting 102: Call Graph 129

- Call Graph Overview 130
 - To Sample or Not to Sample 131
 - In What Environments Does Call Graph Work? 131
 - What About Managed Code? 132
- Before You Begin Profiling 133
 - Generating Base Relocations 134
 - Enabling Source Code View 136

Configuring for Call Graph Data Collection	137
Using the Call Graph Wizard	137
Advanced Configuration Options	140
Viewing Call Graph Results	145
Function Summary	146
Graph	146
Call List	147
Hands-on Call Graph Profiling	149
Choosing a Program Example	149
The Linux Call Graph Viewer	159
Blocked Time on a Thread/Function Basis	167
Does My System Show Wait Time?	167
Why Do Sampling and Call Graph Have Different Hotspots?	167
Call Graph Usage Tips	168
How Call Graph Analysis Works	170
Recap	171

Chapter 6 Hotspot Hunting 103: Counter Monitor 173

Performance Counters, Objects, and Triggers	174
Counter Monitor Data Collection	176
Choosing a Program Example	177
Collecting Counter Data for a Running Thread/Process	193
Identifying Bottlenecks Using Counters	195
Developing Custom DLLs, Counters, and Triggers	197
Counter Monitor Usage Tips	198
How Counter Monitor Analysis Works	199
Recap	200

Chapter 7 Automatic Hotspot Analysis 201

Types of Tuning Advice	202
Counter Monitor-based Advice	203
Sampling-based Advice	203
Source-based Tuning Advice	204
Static and Disassembly-based Advice	205
Before You Begin Using the Intel® Tuning Assistant	205
Using the Tuning Assistant	205
Choosing a Program Example	206
Selecting Events for Which Advice Is Available	220
Recap	221

Part III Usage Models 223

Chapter 8 Comparing Multiple Activities 225

- Comparing Performance Across Target Systems 226
 - Transporting Projects to Different Systems 226
 - Statically Analyzing Your Code for Specific Processors 229
- Comparing Versions of a Developing Program 231
- Comparing Activity Results 231
 - Comparing Sampling Results 231
 - Comparing Source View Results 236
 - Comparing Call Graph Results 237
 - Comparing Call Graph and Sampling Results 239
 - Merging Activity Results of Different Workloads 241
- Getting Help from the Intel® Tuning Assistant 243
 - Comparing Up to Three Binaries on the Same System 243
 - Comparing Binaries Across Two or Three Systems 246
- Computer Architecture-level Tuning 248
 - General Methodology 248
 - Using Sampling 248
 - Tuning for Specific Goals 249
 - Tuning Single-Processor Systems 250
 - Tuning Dual and Multiprocessor Systems 252
 - Tuning Systems with Hyper-Threading Technology 253
- Automating Multiple Tasks 255
 - Collecting Performance Data in Batch Mode 255
 - Using the VTune™ Analyzer's Batch Mode API 256
 - Preparing Linux-based Scripts 263
 - The VTune™ Analyzer as a Tool for Quality Assurance Groups 263
- Recap 264

Chapter 9 Analyzing Interpreted Languages 265

- Analysis of Java Code 267
 - Supported Java Environments 268
 - Configuring Java Sampling: Command Line 268
 - Configuring Java Sampling: GUI 270
 - Call Graph Profiling for Java Applications 282
 - Configuring Java Call Graph Profiling 283

- Analysis of .NET Code (Including C#) 291
 - JIT-Compiled Versus Interpreted MSIL 292
 - Before You Begin 292
 - Configuring .NET Sampling 294
 - Call Graph Profiling for .NET Applications 302
 - Configuring .NET Call Graph Profiling 303
- Analysis of Visual Basic Code 310
- Recap 311

Part IV Advanced Topics 313

Chapter 10 Remote Analysis 315

- How Remote Data Collection Works 316
 - Linux-based Remote Agents 317
 - Windows-based Remote Agents 318
 - WinCE-based Devices with Intel XScale® Technology 319
- Configuring Remote Agents for Data Collection 322
 - Installing VTune™ Analyzer Software 323
 - Preparing Remote Agents 323
 - Before You Begin Remote Data Collection 327
 - Configuring a Wizard on the Controlling System 328
 - Viewing Remotely Collected Data 330
- More About Sampling on Handheld Devices 331
 - Starting Up the Data Collector 333
 - Configuring Transport Media Settings (SRDC Only) 334
 - Creating an Activity 336
 - Configuring Other Sampling Settings 338
 - Starting Sampling 341
 - Locating, Converting, and Viewing Result Files 345
- Recap 347

Chapter 11 Handling Some Special Needs 349

- Analyzing Only Parts of Your Application 350
 - Start Delays 350
 - Pause/Resume Buttons 351
 - Program Control: Pause/Resume API 352
- Analyzing Code Incorporated in the OS 358
 - Issues with Ring 0, Device Drivers, and Kernels 359
 - Limiting Event Sampling to Ring 0 or Ring 3 Only 360
 - A Few Caveats 361
- Recap 361

Chapter 12 Threaded Programs: Correctness 363

- About Intel® Thread Checker 365
- Terminology 366
- Building Code for Use with Intel Thread Checker 367
 - Binary Instrumentation Mode 367
 - Source Instrumentation Mode 368
 - Mixed Mode 368
- Recap 373

Appendix A Types of Line Number and Symbol File Information 375

- Intel® and Microsoft† Compilers 379
- Intel or Microsoft Compilers with Visual Studio† 379
- Intel or Microsoft Compilers with Visual Studio .NET 380
- Microsoft Compilers from the Windows Command Line 384
- Intel Compilers from the Command Line 385

Appendix B Sampling Events and Event Ratios 391

- Event Counters and Event Ratios 392
- Advice for All Processors 393
 - Evaluating the CPI Ratio 393
 - Compilation Tips 394
 - Disabling Power Management 394
 - Optimizing Branch Predictability 394
 - Reducing the Working Data Set to Avoid Cache Misses 396
 - General Coding Tips 397
- Common Event Concepts 397
 - Clockticks: Does Time Exist If You Are Not Moving? 398
 - Many Branches 398
 - Call Overhead (High CALL Event Counts) 398
 - Misprediction, Speculation, Retired, and Executed 399
 - MMX™ Instructions and Streaming SIMD Extensions (SSE)
 - Events 400
 - Cache and Memory Events 401

x ■ VTune™ Performance Analyzer Essentials

- Intel® Pentium® M Processors 402
 - Pentium M Processor Events 402
 - Pentium M Processor Event Ratios 404
- Intel Pentium 4 and Intel Xeon™ Processors 405
 - Hyper-Threading Technology 406
 - Pentium 4 and Intel Xeon Processor Events 407
 - Pentium® 4 and Intel® Xeon™ Processor Event Ratios 420
- Intel Itanium® 2 Processors 429
 - Itanium 2 Processor Events 430
 - Itanium 2 Processor Event Ratios and Other Derived Events 434
- Intel XScale® Technology 439
 - Intel XScale Technology Events 440

Index 443