

# Contents

---

**Preface**   xiii

**Part I Performance Tools and Concepts**   1

**Chapter 1 Introduction**   3

    Building an Application   4  
    Optimization Pitfalls   4  
    The Software Optimization Process   6  
    Key Point   7

**Chapter 2 The Benchmark**   9

    The Attributes of the Benchmark   10  
        Repeatable (Required)   10  
        Representative (Required)   11  
        Easy to Run (Required)   11  
        Verifiable (Required)   12  
        Measure Elapsed Time (Optional)   12  
        Complete Coverage (Situation-dependent)   12  
        Precision (Situation-dependent)   12  
        Quality Assurance and Testing (Desirable, but Optional)   13  
    Automobile Fuel Economy   13  
    Key Points   15

**Chapter 3 Performance Tools**   17

    Timing Tools   17  
    Optimizing Compilers   19  
        Using the Intel® C++ Compiler   19

Optimizing for Specific Processors	21
Writing Functions Specific to One Processor	22
Using SIMD Instructions	23
Automatic Vectorization	24
C++ Class Libraries for SIMD Operations	24
Intrinsics	26
Inline Assembly Language	26
Other Compiler Optimizations	27
Types of Software Profilers	28
Performance Monitor	28
VTune™ Performance Analyzer	29
Sampling	29
Call Graph Profiling	31
Source Code Analysis	33
Microsoft Visual C++ Profiler	34
Sampling Versus Instrumentation	35
Human Brain: Trial and Error, Common Sense, and Patience	36
Key Points	37

## **Chapter 4 The Hotspot 39**

What Causes Hotspots and Cold-spots?	40
More Than Just Time	41
Uniform Execution and No Hotspots	43
Key Points	45

## **Chapter 5 Processor Architecture 47**

Functional Blocks	48
Two Cheeseburgers Please!	49
Instruction Fetch and Decode	52
Instruction Execution	53
Retirement	55
Memory	55
Key Points	57

## **Part II Performance Issues 59**

### **Chapter 6 Algorithms 61**

Computational Complexity	61
Choice of Instructions	62
Data Dependencies and Instruction Parallelism	66
Memory Requirements	68
Detecting Algorithm Issues	69
Key Points	71

### **Chapter 7 Branching 73**

Finding the Critical Mis-predicted Branches	76
---	----

Step 1: Find the Mis-predicted Branches	76
Step 2: Find the Time-consuming Hotspots	77
Step 3: Determine the Percentage of Mis-predicted Branches	78
Final Sanity Check	79
The Different Types of Branches	80
Removing Branches with CMOV	82
Removing Branches with Masks	83
Removing Branches with Min/Max Instructions	84
Removing Branches By Doing Extra Work	85
Improving Branches	86
Key Points	87

## **Chapter 8 Memory 89**

Memory Overview	90
Main Memory and Virtual Memory	91
Processor Caches	91
L1 Cache Details	93
Software Prefetch	94
Writing Data Without the Cache: Non-temporal Writes	96
Issues Affecting Memory Performance	97
Cache Compulsory Loads	98
Cache Capacity Loads	98
Cache Conflict Loads	99
Cache Efficiency	100
Data Alignment	100
Compilers and Data Alignment	102

Detecting Memory Issues	102
Finding Page Misses	103
Finding L1 Cache Misses	106
Understanding Potential Improvement	107
Fixing Memory Problems	109
Key Points	115

## **Chapter 9 Loops 117**

Common Loop Problems	118
Loop Unrolling	119
Loop Invariant Work	123
Loop Invariant Branches	124
Iteration Dependencies	125
Memory Address Dependencies	126
Key Points	127

## **Chapter 10 Slow Operations 129**

Slow Instructions	130
Lookup Tables	131
System Calls	134
System Idle Process	137
Key Points	141

## **Chapter 11 Floating Point 143**

Numeric Exceptions	144
Flush-to-Zero and Denormals are Zero	147
Precision	148
Scalar-SIMD Floating Point	152
Float-to-Integer Conversions, Rounding	152
Using the Intel C++ Compiler's Round Instead of Truncate Switch	154
Assembly Language	154
SIMD Convert with Truncation Instructions	157
Direct Bit Manipulation	157
Floating-Point Manipulation Tricks	158
Square Root	158
Reciprocal Square Root	158
Key Points	159

**Chapter 12 SIMD 161**

- Using the SIMD Instructions 162
- SIMD Instruction Issues 164
  - Data Alignment 164
  - Compatibility of SIMD and x87 FPU Floating-Point Calculations 165
  - Data Simplifying Buffer Lengths/Padding 166
- Integer SIMD 166
- Single-Precision Floating-Point SIMD 167
- Reciprocal Approximations Accuracy 168
- Double-Precision Floating-Point SIMD 168
- SIMD Data Organization 169
- Determining Where to Use SIMD 176
- Key Points 177

**Chapter 13 Processor-Specific Optimizations 179**

- 32-bit Intel Architectures 179
- The Pentium III Processor 182
- L1 Instruction Cache 182
- Instruction Decoding 183
- Instruction Latencies 184
- Instruction Set 185
- Floating-Point Control Register 185
- L1 Data Cache 186
- Memory Prefetch 186
- Processor Events 187
- Partial Register Stalls 187
- Partial Flag Stall 189
- Pause Instruction 189
- Key Points 189

**Chapter 14 Introduction to Multiprocessing 191**

- Parallel Programming 192
- Thread Management 193
  - Low-level Thread Libraries 194
  - High-Level Thread Management with OpenMP 194
- Threading Goals 197
- Threading Issues 198
- Tools 202
- Key Points 203

**Part III DESIGN AND APPLICATION OPTIMIZATION 205****Chapter 15 Design For Performance 207**

- Data Movement 208

Performance Experiments for Design	210
Algorithms	211
Key Points	213

## **Chapter 16 Putting It Together: Basic Optimizations 215**

The Sample Application	215
Quick Review of the Algorithms	220
Bilinear Pixel Interpolation	220
Alpha Blending	222
Image Rotation	223
Let The Optimizations Begin	224
Compilation	224
The Benchmark	225
Locate the Hotspots	228
Removing Calls To <code>_ftol</code>	229
Algorithm Issues	233
Investigation and Thought	234
Performance Experiments	236
Removing Work	237
Calling Functions Differently	238
Summary of Optimizations	239
Key Points	241

## **Chapter 17 Putting It Together: More Optimizations 243**

Additional Analysis	243
Writing a Specialized Merged Function	247
Using SIMD Technology	250
More Analysis, Reduce MemCopyRect	254
Pick a Different Algorithm	256
Improving the Algorithm	257
Pre-calculating Values	259
Write-Combining Memory	260
More Analysis, Remove Multiplies	263
Knowing When to Stop Optimizing	265
Summary of Optimizations	266
Key Points	266

## **References 267**

## **Index 269**