

Contents

Preface i

Why All the Effort for Hyper-Threading Technology? ii

Why This Book? iii

A Note about Gender iv

Chapter 1 An Introduction to Hyper-Threading Technology 1

More on Hyper-Threading Technology 3

Why Hyper-Threading Technology Is Important 4

Early Days 4

Then, Modern Operating Systems 5

Today: Parallel Processing with Multiple Threads 6

Additional Benefits of Hyper-Threading Technology 6

Performance Scalability 7

Cleaner Software Architecture 7

Key Points 8

Chapter 2 Fundamental Concepts of Parallel Programming 9

Designing for Threads 9

Functional Decomposition 10

Producer/Consumer 11

Data Decomposition 12

Implications of Different Decompositions 12

Challenges You'll Face	14
Programming a Thread	15
Windows Code	15
Pthreads and Linux Code	19
OpenMP Code	20
What Happens When a Thread Is Created	22
Thread Execution at the Processor Level	25
Hyper-Threading Technology	25
Key Points	29

Chapter 3 Elements of Parallel Programming 31

Coordinating Threads	31
The Spin Wait	32
Waiting for Events in Windows	34
Waiting for Events (in Pthreads)	35
Mutexes	37
Mutexes in Pthreads	43
Critical Sections	44
Tips on Using Critical Sections	48
Interlocked Functions	48
Comparative Timings	51
Waiting for Specific Events	56
Windows Events	57
Events in Pthreads	61
Recap	69
Key Points	70

Chapter 4 Basic Parallel Programming 71

Queues	76
Implementing the Queue in Code	80
Other Queue Implementations	85
Thread Pooling	85
Many Threads, Similar Work	85
Many Threads, Dissimilar Work	87
Key Points	89

Chapter 5 Threading APIs 91

Windows APIs	92
Compilation and Linking	92
Managing Threads	93
Thread Pools	95

Thread Priority	97
Processor Affinity	99
Fibers: The Do-it-Yourself Threads	102
Pthreads	107
Compilation and Linking	107
Key Points	110

Chapter 6 OpenMP: The Portable Solution 111

Challenges in Loop Parallelization	112
Flow Dependencies	113
Race Conditions	113
Managing Shared and Private Data	114
Loop Scheduling and Load Balancing	115
Array Reductions Using Loops	117
Improving Loop Performance with OpenMP	119
Work-Sharing Sections	120
Other Performance-Oriented Pragmas	121
Barriers and the <code>nowait</code> Clause	121
Single-Thread Execution	122
<code>firstprivate</code> and <code>lastprivate</code>	123
Critical Sections	124
Atomic Operations	125
Run-time OpenMP Functions	126
OpenMP Environment Variables	128
Compilation	128
Debugging	129
Performance	131
Key Points	132

Chapter 7 Performance 133

Starting Right	134
Cold Spots	134
Maximum Theoretical Performance	136
Typical Performance	137
Parallel Performance on Hyper-Threading Technology	139
Load Balancing	140
Scalability	143
How Many Threads?	143
Memory Constraints on Performance	149
Allocating from the Heap	149
False Sharing	151

The 64-Kilobyte Aliasing Problem 152
Key Points 153

Chapter 8 Parallel Programming Headaches 155

Passing Variables to Threads 155
Data Races, Deadlocks and Live Locks 158
 Ordered Lock Acquisition 161
 Live Locks 164
Heavily Contended Locks 166
Recursive Locks 169
Memory Issues 171
 Memory Sharing 172
Thread-safe Functions and Libraries 179
Key Points 180

Chapter 9 Hyper-Threading Technology and Processor-level Issues 183

Why Hyper-Threading Technology Was Developed 184
Instruction Execution on Hyper-Threading Technology 186
Riding the Bus—A Serializing Effect 187
The Question of Caches 188
 Write-Combining Buffers 192
Avoiding Pipeline Stalls 193
 Spin Waits and the `pause` Instruction 193
 Spin-wait Instructions on the Prescott Architecture 195
 Serializing Events 196
Other Considerations 197
Key Points 198

Glossary 199

References and Further Reading 207

Other Useful Resources 208
 Books 208
 Web Sites 209

Index 211