

---

# Intel Threading Building Blocks

*Outfitting C++ for Multi-Core  
Processor Parallelism*

*James Reinders*

**O'REILLY®**

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

---

# Table of Contents

<b>Foreword</b> .....	<b>xiii</b>
<b>Note from the Lead Developer of Intel Threading Building Blocks</b> .....	<b>xv</b>
<b>Preface</b> .....	<b>xix</b>
<b>1. Why Threading Building Blocks?</b> .....	<b>1</b>
Overview	2
Benefits	2
<b>2. Thinking Parallel</b> .....	<b>7</b>
Elements of Thinking Parallel	8
Decomposition	9
Scaling and Speedup	13
What Is a Thread?	19
Mutual Exclusion and Locks	22
Correctness	23
Abstraction	25
Patterns	25
Intuition	27
<b>3. Basic Algorithms</b> .....	<b>29</b>
Initializing and Terminating the Library	30
Loop Parallelization	32
Recursive Range Specifications	52
Summary of Loops	64

<b>4. Advanced Algorithms</b> .....	<b>65</b>
Parallel Algorithms for Streams	65
<b>5. Containers</b> .....	<b>80</b>
concurrent_queue	81
concurrent_vector	86
concurrent_hash_map	91
<b>6. Scalable Memory Allocation</b> .....	<b>101</b>
Limitations	101
Problems in Memory Allocation	101
Memory Allocators	103
Replacing malloc, new, and delete	104
<b>7. Mutual Exclusion</b> .....	<b>110</b>
When to Use Mutual Exclusion	111
Mutexes	112
Mutexes	118
Atomic Operations	122
<b>8. Timing</b> .....	<b>130</b>
<b>9. Task Scheduler</b> .....	<b>133</b>
When Task-Based Programming Is Inappropriate	133
Much Better Than Raw Native Threads	134
Initializing the Library Is Your Job	137
Example Program for Fibonacci Numbers	137
Task Scheduling Overview	140
How Task Scheduling Works	142
Recommended Task Recurrence Patterns	145
Making Best Use of the Scheduler	147
Task Scheduler Interfaces	153
Task Scheduler Summary	168
<b>10. Keys to Success</b> .....	<b>169</b>
Key Steps to Success	169
Relaxed Sequential Execution	170
Safe Concurrency for Methods and Libraries	171

Debug Versus Release	172
For Efficiency's Sake	172
Enabling Debugging Features	172
Mixing with Other Threading Packages	174
Naming Conventions	176
<b>11. Examples</b> .....	<b>177</b>
The Aha! Factor	177
A Few Other Key Points	179
parallel_for Examples	180
The Game of Life	190
Parallel_reduce Examples	199
CountStrings: Using concurrent_hash_map	209
Quicksort: Visualizing Task Stealing	215
A Better Matrix Multiply (Strassen)	223
Advanced Task Programming	230
Packet Processing Pipeline	237
Memory Allocation	257
Game Threading Example	262
Physics Interaction and Update Code	271
Open Dynamics Engine	275
<b>12. History and Related Projects</b> .....	<b>283</b>
Libraries	283
Languages	285
Pragmas	286
Generic Programming	286
Caches	289
Costs of Time Slicing	290
Quick Introduction to Lambda Functions	291
Further Reading	292
<b>Index</b> .....	<b>297</b>