Structured Parallel Programming
Patterns for Efficient Computation

Michael McCool Software Architect, Intel Corporation
James Reinders Director, Chief Evangelist, Intel Software, Intel Corporation
Arch Robison Senior Principal Engineer, Intel Corporation

ISBN: 978-0-12-415993-8

PUB DATE: late June 2012

LIST PRICE: $59.95

FORMAT: Paperback

PAGES: c. 406

TRIM: 7.5w x 9.25h

AUDIENCE
Software Developers, Computer Programmers, Software Architects

TABLE OF CONTENTS

The simplest way for developers to learn patterns for high-performance parallel programming

"I've been dreaming for a while of a modern accessible book that I could recommend to my threading-deprived colleagues and assorted enquirers to get them up to speed with the core concepts of multithreading as well as something that covers all the major current interesting implementations. Finally I have that book."

— Martin Watt, Principal Engineer, Dreamworks Animation

KEY FEATURES
- The patterns-based approach offers structure and insight that developers can apply to a variety of parallel programming models
- Develops a composable, structured, scalable, and machine-independent approach to parallel computing
- Includes detailed examples in both Cilk Plus and the latest Threading Building Blocks, which support a wide variety of computers

DESCRIPTION
Programming is now parallel programming. Much as structured programming revolutionized traditional serial programming decades ago, a new kind of structured programming, based on patterns, is relevant to parallel programming today. Parallel computing experts and industry insiders Michael McCool, Arch Robison, and James Reinders describe how to design and implement maintainable and efficient parallel algorithms using a pattern-based approach. They present both theory and practice, and give detailed concrete examples using multiple programming models. Examples are primarily given using two of the most popular and cutting edge programming models for parallel programming: Threading Building Blocks, and Cilk Plus. These architecture-independent models enable easy integration into existing applications, preserve investments in existing code, and speed the development of parallel applications. Examples from realistic contexts illustrate patterns and themes in parallel algorithm design that are widely applicable regardless of implementation technology.