

# Multi-Core Programming

Increasing Performance through Software  
Multi-threading

Shameem Akhter  
Jason Roberts

Intel  
PRESS

Copyright © 2006 Intel Corporation. All rights reserved.

ISBN 0-9764832-4-6

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Publisher, Intel Press, Intel Corporation, 2111 NE 25<sup>th</sup> Avenue, JF3-330, Hillsboro, OR 97124-5961. E-mail: intelpress@intel.com.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

Fictitious names of companies, products, people, characters, and/or data mentioned herein are not intended to represent any real individual, company, product, or event.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel, the Intel logo, Celeron, Intel Centrino, Intel NetBurst, Intel Xeon, Itanium, Pentium, MMX, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

† Other names and brands may be claimed as the property of others.

This book is printed on acid-free paper. ∞

Publisher: Richard Bowles

Editor: David J. Clark

Managing Editor: David B. Spencer

Content Architect: Stuart Goldstein

Text Design & Composition: Interactive Composition Corporation

Graphic Art: Kirsten Foote (illustrations), Ted Cyrek (cover)

***Library of Congress Cataloging in Publication Data:***

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

First printing, April 2006

# Contents

---

## **Preface xi**

## **Chapter 1 Introduction to Multi-Core Architecture 1**

- Motivation for Concurrency in Software 2
- Parallel Computing Platforms 5
  - Parallel Computing in Microprocessors 7
  - Differentiating Multi-Core Architectures from Hyper-Threading Technology 10
  - Multi-threading on Single-Core versus Multi-Core Platforms 11
- Understanding Performance 13
  - Amdahl's Law 14
  - Growing Returns: Gustafson's Law 18
- Key Points 19

## **Chapter 2 System Overview of Threading 21**

- Defining Threads 22
- System View of Threads 22
  - Threading above the Operating System 23
  - Threads inside the OS 26
  - Threads inside the Hardware 29
- What Happens When a Thread Is Created 30
- Application Programming Models and Threading 32
- Virtual Environment: VMs and Platforms 33
  - Runtime Virtualization 33

System Virtualization 33

Key Points 35

### **Chapter 3 Fundamental Concepts of Parallel Programming 37**

Designing for Threads 37

Task Decomposition 38

Data Decomposition 39

Data Flow Decomposition 40

Implications of Different Decompositions 41

Challenges You'll Face 42

Parallel Programming Patterns 42

A Motivating Problem: Error Diffusion 45

Analysis of the Error Diffusion Algorithm 48

An Alternate Approach: Parallel Error Diffusion 48

Other Alternatives 50

Key Points 51

### **Chapter 4 Threading and Parallel Programming Constructs 53**

Synchronization 53

Critical Sections 56

Deadlock 57

Synchronization Primitives 59

Semaphores 60

Locks 63

Condition Variables 66

Messages 68

Flow Control-based Concepts 71

Fence 71

Barrier 72

Implementation-dependent Threading Features 73

Key Points 74

### **Chapter 5 Threading APIs 75**

Threading APIs for Microsoft Windows 75

Win32/MFC Thread APIs 75

Threading APIs for Microsoft .NET Framework 107

Creating Threads 107

Managing Threads	110
Thread Pools	112
Thread Synchronization	117
POSIX Threads	120
Creating Threads	120
Managing Threads	122
Thread Synchronization	123
Signaling	124
Compilation and Linking	132
Key Points	132

## **Chapter 6 OpenMP<sup>†</sup>: A Portable Solution for Threading 135**

Challenges in Threading a Loop	137
Loop-carried Dependence	137
Data-race Conditions	140
Managing Shared and Private Data	141
Loop Scheduling and Partitioning	143
Effective Use of Reductions	147
Minimizing Threading Overhead	149
Work-sharing Sections	151
Performance-oriented Programming	152
Using Barrier and Nowait	152
Interleaving Single-thread and Multi-thread Execution	154
Data Copy-in and Copy-out	155
Protecting Updates of Shared Variables	157
Intel Taskqueuing Extension to OpenMP	160
OpenMP Library Functions	162
OpenMP Environment Variables	163
Compilation	164
Debugging	165
Performance	167
Key Points	169

## **Chapter 7 Solutions to Common Parallel Programming Problems 171**

Too Many Threads	171
Data Races, Deadlocks, and Live Locks	174

- Deadlock 177
- Heavily Contended Locks 181
  - Priority Inversion 181
  - Solutions for Heavily Contended Locks 183
- Non-blocking Algorithms 186
  - ABA Problem 188
  - Cache Line Ping-ponging 190
  - Memory Reclamation Problem 190
  - Recommendations 191
- Thread-safe Functions and Libraries 192
- Memory Issues 193
  - Bandwidth 193
  - Working in the Cache 194
  - Memory Contention 197
- Cache-related Issues 200
  - False Sharing 200
  - Memory Consistency 204
    - Current IA-32 Architecture 204
    - Itanium® Architecture 207
  - High-level Languages 210
- Avoiding Pipeline Stalls on IA-32 211
- Data Organization for High Performance 212
- Key Points 213

## **Chapter 8 Multi-threaded Debugging Techniques 215**

- General Debug Techniques 215
  - Designing with Debugging in Mind 216
  - Extending your Application—Using Trace Buffers 219
- Debugging Multi-threaded Applications in Windows 224
  - Threads Window 225
  - Tracepoints 225
  - Breakpoint Filters 226
  - Naming Threads 227
  - Putting it All Together 228
- Multi-threaded Debugging Using GDB 232
  - Notification on Thread Creation 233
  - Getting a List of All Threads in the Application 233
  - Setting Thread-specific Breakpoints 233

- Switching between Threads 235
- Applying a Command to a Group of Threads 235
- Key Points 236

## **Chapter 9 Single-Core Processor Fundamentals 237**

- Processor Architecture Fundamentals 237
- Comparing Superscalar and EPIC Architecture 245
- Key Points 246

## **Chapter 10 Threading on Intel® Multi-Core Processors 247**

- Hardware-based Threading 247
  - Threading from Intel 251
- Hyper-Threading Technology 252
  - Difference between Multiprocessor and Hyper-Threading Technology 254
  - Hyper-Threading Technology Architecture 254
- Multi-Core Processors 257
  - Architectural Details 257
  - Comparison between Multiprocessors and Multi-Core Processors 260
  - Multi-Core for Itanium® Architecture 261
- Multiple Processor Interaction 266
  - Inter-Processor Communication and Multi-threaded Programming 266
- Power Consumption 268
  - Power Metrics 268
  - Reducing Power Consumption 270
- Beyond Multi-Core Processor Architecture 271
- Key Points 272

## **Chapter 11 Intel® Software Development Products 275**

- Overview 275
  - Investigate 276
  - Create/Express 276
  - Debugging 277
  - Tuning 277
- Intel® Thread Checker 277
  - How It Works 278
  - Usage Tips 280

- Using Intel® Thread Checker with OpenMP 281
- Intel Compilers 281
  - OpenMP† 282
  - Software-based Speculative Precomputation 286
  - Compiler Optimization and Cache Optimization 287
- Intel® Debugger 288
- Intel Libraries 289
  - Intel® Math Kernel Library 289
  - Intel® Integrated Performance Primitives 290
  - Parallel Program Issues When Using Parallel Libraries 290
  - The Future 291
  - Intel® Threading Building Blocks 292
- Intel® VTune™ Performance Analyzer 292
  - Find the Hotspot 293
  - Using Call Graph for Finding a Threading Point 294
  - Check the Load Balancing 295
- Intel® Thread Profiler 295
- MPI Programming 296
  - Intel Support for MPI 297
- Key Points 300

**Glossary 303**

**References 317**

**Index 323**