High-Performance Computing on the Intel® Xeon Phi™

Endong Wang · Qing Zhang · Bo Shen
Guangyong Zhang · Xiaowei Lu
Qing Wu · Yajuan Wang

High-Performance Computing on the Intel® Xeon Phi™

How to Fully Exploit MIC Architectures

Springer

Endong Wang
Qing Zhang
Bo Shen
Guangyong Zhang
Xiaowei Lu
Qing Wu
Yajuan Wang
Inspur, Beijing, China

Translators
Dave Yuen
Chuck Li
Larry Zheng
Sergei Zhang
Caroline Qian

中国水利水电出版社
www.waterpub.com.cn

Copyright © 2012 by China Water & Power Press, Beijing, China
Title of the Chinese original: MIC 高性能计算编程指南
ISBN: 978-7-5170-0338-0
All rights reserved

ISBN 978-3-319-06485-7 ISBN 978-3-319-06486-4 (eBook)
DOI 10.1007/978-3-319-06486-4
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014943522

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword by Dr. Rajeeb Hazra

Today high-performance computing (HPC), especially the latest massively supercomputers, has developed quickly in computing capacity and capability. These developments are due to several innovations. Firstly, Moore's law, named after the Intel founder Gordon Moore, predicts that the number of semiconductor transistors will double every 18–24 months. According to Moore's law, Intel continues to improve performance and shrink the size of transistors as well as to reduce power consumption, all at the same time. Another innovation is a series of CPU-improving microstructures which ensure that the performance of a single thread coincides with the parallelism in each successive CPU generation.

The development of HPC plays an important role in society. Although people are inclined to pay more attention to great scientific achievements such as the search for the Higg's boson or the cosmological model of cosmic expansion, the computing capability that everyone can now acquire is also impressive. A modern two-socket workstation based on the Intel® Xeon series processors could show the same performance as the top supercomputer from 15 years ago. In 1997, the fastest supercomputer in the world was ASCI Red, which was the first computing system to achieve over 1.0 teraflops. It had 9298 Intel Pentium Pro processors, and it cost $55,000,000 per teraflop. In 2011, the cost per teraflop was reduced to less than $1000. This reduction in cost makes high-performance computing accessible to a much larger group of researchers.

To sufficiently make use of the ever-improving CPU performance, the application itself must take advantage of the parallelism of today's microprocessors. Maximizing the application performance includes much more than simply tuning the code. Current parallel applications make use of many complicated nesting functions, from the message communication among processors to the parameters in threads. With Intel CPUs, in many instances we could achieve gains of more than ten times performance by exploiting the CPU's parallelism.

The new Intel® Xeon Phi™ coprocessor is built on the parallelism programming principle of the Intel® Xeon processor. It integrates many low-power consumption cores, and every core contains a 512-bit SIMD processing unit and many new vector instructions. This new CPU is also optimized for performance per watt. Due to a computing capability of over one billion times per second, the Intel® Xeon Phi™ delivers a supercomputer on a chip. This brand new microstructure delivers ground-breaking performance value per watt, but the delivered performance also relies on those applications being sufficiently parallelized and expanded to utilize many cores, threads, and vectors. Intel took a new measure to release this parallelism ability. Intel followed the common programming languages (including C, C++, and Fortran) and the current criteria. When readers and developers learn how to optimize and make use of these languages, they are not forced to adopt nonstandard or hardware-dependent programming modes. Furthermore, the method, based on the criteria, ensures the most code reuse, and could create the most rewards by compiling the transplantable, standardized language and applying it to present and future compatible parallel code.

In 2011, Intel developed a parallel computing lab with Inspur in Beijing. This new lab supplied the prior use and development environment of the Intel® Xeon processor and Intel® Xeon Phi™ coprocessor to Inspur Group and to some excellent application developers. Much programming experience can be found in this book. We hope to help developers to produce more scientific discovery and creation, and help the world to find more clean energy, more accurate weather forecasts, cures for diseases, develop more secure currency systems, or help corporations to market their products effectively.

We hope you enjoy and learn from this venerable book. This is the first book ever published on how to use the Intel® Xeon Phi™ coprocessor.

Santa Clara, CA Rajeeb Hazra

Foreword by Prof. Dr. Rainer Spurzem

Textbooks and teaching material for my Chinese students are often written in English, and sometimes we try to find or produce a Chinese translation. In the case of this textbook on high-performance computing with the Intel MIC, we now have a remarkable opposite example—the Chinese copy appeared first, by Chinese authors from Inspur Inc. led on by Wang Endong, and only some time later can all of us English speakers enjoy and benefit from its valuable contents. This occasion happens not by chance—two times in the past several years a Chinese supercomputer has been the certified fastest supercomputer in the world by the official Top500 list (http://www.top500.org). Both times Chinese computational scientists have found a special, innovative way to get to the top—once with NVIDIA's GPU accelerators (Tianhe-1A in 2010) and now, currently, with the Tianhe-2, which realizes its computing power through an enormous number of Intel Xeon Phi hardware, which is the topic of this book. China is rapidly ascending on the platform of supercomputing usage and technology at a much faster pace than the rest of the world. The new Intel Xeon Phi hardware, using the Intel MIC architecture, has its first massive installation in China, and it has the potential for yet another supercomputing revolution in the near future. The first revolution, in my opinion, has been the transition from traditional mainframe supercomputers to Beowulf PC clusters, and the second was the acceleration and parallelization of computations by general-purpose computing on graphical processing units (GPGPU). Now the stage is open for—possibly—another revolution by the advent of Intel MIC architecture. The past revolutions of accelerators comprised a huge qualitative step toward better price–performance ratio and better use of energy per floating point operation. In some ways they democratized supercomputing by making it possible for small teams or institutes to assemble supercomputers from off-the-shelf components, and later even (GPGPU) provide massively parallel computing in just a single desktop. The impact of Intel Xeon Phi and Intel MIC on the market and on scientific supercomputing has yet to be seen. However, already a few things can be anticipated; and let me add that I write this from the perspective of a current heavy user and provider of GPGPU capacity and capability. GPGPU architecture, while it provides outstanding performance for a fair range of applications, is still not as common as expected a few years ago. Intel MIC, if it fulfills the promise of top-class performance together with compatibility to a couple of standard programming paradigms (such as OpenMP as it works on standard Intel

CPUs, or MPI as it works on standard parallel computers) may quickly find a much larger user community than GPU. I hope very much that this very fine book can help students, staff, and faculty all over the world in achieving better results when implementing and accelerating their tasks on this interesting new piece of hardware, which will for sure appear on desktops, in institutional facilities, as well as in numerous future supercomputers.

Beijing, China
Rainer Spurzem

Foreword by Endong Wang

Currently scientists and engineers everywhere are relentlessly seeking more computing power. The capability of high-performance computing has become the competition among the few powerful countries in the world. After the "million millions flops" competition ended, the "trillion flops" contests have begun. The technology of semiconductors restricts the frequency of processors, but multi-processors and the many-integrated processors have become more and more important. When various kinds of many-integrated cores came out, we found that although the high point of computing has increased a lot, the compatibility of the applications became worse, and the development of applications has become more complicated. A lack of useful applications would render the supercomputer useless.

At the end of 2012, Intel corporation brought out the Intel® Xeon Phi™ coprocessor based on the many-integrated core. This product integrated more than 50 cores that were based on the x86 architecture into one PCI -Express interface card. It is a powerful supplement to the Intel® Xeon CPU, and brings a new performance experience for a highly parallelized workload. It is easy to program on this product, and there's almost no difference when compared with traditional programming. The code on the Intel® Xeon Phi™ coprocessor could be applied to a traditional platform based on CPU without any modifications, which protects the user's software investment. It can supply hundreds of running hardware threads, which could bring high parallelism and meet the current demands of high parallelization.

The Inspur-Intel China Parallel Computing Joint Lab was found on August 24, 2011. This lab aims to promote the trillion-flops supercomputing system architecture and application innovation, establish the ecological condition of high-performance computing, and accelerate supercomputing in China into the trillion-flops era. The research and innovation in the Inspur-Intel China Parallel Computing Joint Lab will make a positive impact on the development of supercomputing in China in the next ten years, especially in the beginning of the trillion-flops era for the rest of the world. The Inspur-Intel China Parallel Computing Joint Lab contributed to the completion of the Intel® Xeon Phi™ coprocessor and made a tremendous effort to popularize it.

This book was finished by several dedicated members of the Inspur-Intel China Parallel Computing Joint Lab. In this book, relevant knowledge about the Intel® Xeon Phi™ coprocessor, programming methods in using the Intel® Xeon Phi™

coprocessor, optimizations for the program, and two successful cases of applying the Intel® Xeon Phi™ coprocessor in practical high-performance computing are introduced. This book has a clear structure and is easy to understand. It contains a programming basis, optimization, and specific development projects. At the same time, a lot of figures, diagrams, and segments of program were included to help readers understand the material. The authors of this book have plenty of project experience and have added their practical summaries of these projects. So this book not only introduces the theory, but it also connects more closely to actual programming. This book is also the first to introduce the Intel® Xeon Phi™ coprocessor and embodies the achievement of these authors. We hope to see China accumulate some great experience in the field of HPC. The authors and the members of the Inspur-Intel China Parallel Computing Joint Lab made great efforts to ensure the book publishing coincides with the Intel® Xeon Phi™ coprocessor, and they should be respected for this.

We hope the readers will grasp the full use of the Intel® Xeon Phi™ coprocessor quickly after reading this book, and gain achievements in their own fields of HPC application by making use of the Intel® Xeon Phi™ coprocessor. The Inspur Group hopes to dedicate themselves to HPC endeavors together with Intel Corporation.

Beijing, China Endong Wang

Preface

High-performance computing (HPC) is a recently developed technology in the field of computer science, and now computational science. HPC can secure a country's might, improve its national defense science, and promote the rapid development of highly sophisticated weapons. HPC is one of the most important measures of a country's overall prowess and economic strength. With the rapid growth of an information-based society, people are demanding more powerful capabilities in information processing. HPC is used not only for oil exploration, weather prediction, space technology, national defense, and scientific research, but also in finance, government, education, business, network games, and other fields that demand more computing capability. The drive in research to reach the goal of "trillion flops" computing has begun, and people are looking forward to solving larger scale and more complicated problems by using a trillion-flops supercomputer.

In this century, the many-integrated core (MIC) era has finally arrived. Today, the HPC industry is going through a revolution, and parallel computing will be the trend of the future as a prominent hot spot for scientific research. Current mainstream research has adopted the CPU-homogeneous architecture, in which there are dozens of cores in one node; this is not unusual. In large-scale computing, thousands of cores will be needed. Meanwhile, the CPU-homogeneous architecture faces a huge challenge because of its low performance-to-power ratio, performance-to-access memory ratio, and low parallel efficiency. When computing with the CPU+GPU heterogeneous architecture, the MIC acceleration technology of GPU is used. More and more developers have become dedicated to this field, but it also faces challenges such as fined-grained parallel algorithms, programming efficiency, and performance on a large scale. This book focuses on the central issues of how to improve the efficiency of large-scale computing, how to simultaneously shorten programming cycles and increase software productivity, and how to reduce power consumption.

Intel Corporation introduced the Intel® Xeon Phi™ series products, which are based on the MIC, to solve highly parallelized problems. The performance of the double-precision of this product has reached teraflop levels. It is based on the current x86 architecture, and supports OpenMP, pThread, MPI, and many parallel programming models. It also supports the traditional C/C++/Intel® Cilk™ Plus, Fortran, and many other programming languages. It is programmed easily, and many associated tools are supported. For applications that are difficult to realize by the traditional CPU platform, the MIC platform will greatly improve performance, and the source code can be shared by the CPU and MIC platform without any modifications. The combination of CPU and MIC in the x86 platform in heterogeneous computing provides HPC users with a new supercomputing solution.

Since the Inspur-Intel China Parallel Computing Joint Lab was established on August 24, 2011, the members have dedicated themselves to HPC application programs on the MIC platform, and have ensured that the Intel® Xeon Phi™ series products would be released smoothly. We have accumulated a large amount of experience in exploring the software and hardware of MIC. It is a great honor for us to participate in the technology revolution of HPC and introduce this book to readers as pioneers. We hope more readers will make use of MIC technology and enjoy the benefits brought forth by the Intel® Xeon Phi™ series products.

Target Audience

The basic aim of this book is to help developers learn how to efficiently use the Intel® Xeon Phi™ series products, by which they can develop, transplant, and optimize their parallel programs. The general content of this book introduces some computing grammar, programming technology, and optimization methods in using MIC, and we also offer some solutions to the problems encountered during actual use based on our optimization experience.

We assume that readers already have some basic skills in parallel programming, but have a scant knowledge of MIC. This book does not intend to introduce the theory of parallel computing or algorithms, so we also assume that readers already have this knowledge. In spite of this, when faced with the parallel algorithm, we still describe it in a simple way. We assume that readers are familiar with OpenMP, MPI, and other parallel models, but we also state the basic grammar. We assume that readers can make use of any one of the C/C++/Fortran languages, and that C/C++ is preferred. However, the ideas and advice stated in this book are also adapted to other high-level languages. Moreover, when the Intel® Xeon Phi™ series of products support other languages in the future, most of the optimization methods and application experience will still be effective. Generally speaking, this book is for three types of computing-oriented people:

Students and professional scientists and engineers in colleges, universities, and research institutes, and developers engaged in parallel computing, multi-core, and many integrated core technology.

IT employees, especially those who develop HPC software, improve application performance by many-integrated cores, and pursue extreme performance in the HPC field.

HPC users in other fields, including oil exploration, biological genetics, medical imaging, finance, aerospace, meteorology, and materials chemistry. We hope to help them to improve the original CPU performance by means of MIC and ultimately increase productivity.

We wish to benefit more readers with this book. In the future we also hope to engage more and more readers around the world.

About This Book

Because of the diverse characteristics of MIC architecture, this book cannot be sorted strictly into well-defined sections. This book introduces the MIC programming language and Intel® Xeon Phi™ series products, and it also describes optimization in parallel programming. Through this book, we hope readers will fully understand MIC, and we expect readers to make good use of MIC technology in future practice.

This book includes three parts. The first one covers MIC basics, and includes Chaps. 1–7, in which fundamental knowledge about the MIC technology is introduced.

In Chap. 1, the development of parallel computing is recalled briefly. The current hardware characteristics of parallel computing are compared. Then MIC technology is introduced, and the advantages of MIC are stated.

In Chap. 2, the hardware and software architecture of MIC are introduced. Although there's no influence on programming on MIC in the absence of this background knowledge, exploring the MIC architecture deeply will help our programs become more adapted to MIC.

In Chap. 3, by computing the circumference ratio pi, the characteristics of MIC programming are directly demonstrated to readers. In addition, we introduce the background procedures of the program.

In Chap. 4, the background knowledge of MIC programming is discussed, including the basic grammar of OpenMP and MPI. If you have had this basic training, you can skip this chapter altogether.

In Chap. 5, the programming model, grammar, environment variables, and compilation options of MIC are introduced. You should be able to grasp the method of writing your own MIC program by this chapter.

In Chap. 6, some debugging and optimization tools and their usage are introduced. These tools bring a great deal of convenience to debugging and optimization.

In Chap. 7, some Intel mathematical libraries that have been adapted on MIC are discussed, including VML, FFT, and Blas.

The second section covers performance optimization, and comprises Chaps. 8 and 9.

In Chap. 8, the basic principles and strategy of MIC optimization are introduced, and then the methods and circumstance of MIC optimization are stated. The general methods of MIC optimization are covered. Moreover, most of the methods are applicable to the CPU platform, with a few exceptions.

In Chap. 9, through the classical example in parallel computing—the optimization of matrix multiplication—the optimization measures are stated step-by-step in the method of integrating theory with practice.

The third and last section covers project development, and includes Chaps. 10 and 11.

In Chap. 10, we propose a set of methods to apply parallel computing to project applications by summarizing our experiences on development and optimization of our own projects. We also discuss how to determine if a serial or parallel CPU program is suitable for MIC, and how to transplant the program onto MIC.

In Chap. 11, we show, using two actual cases of how the MIC technology influences an actual project.

In the early stages, this book was initiated by Endong Wang, the director of the State Key Laboratory of high-efficiency server and storage technology at the Inspur-Intel China Parallel Computing Joint Lab, and the senior vice president of Inspur Group Co., Ltd. Qing Zhang, the lead engineer of the Inspur-Intel China Parallel Computing Joint Lab, formulated the plan, outline, structure, and content of every chapter. Then, in the middle stage, Qing Zhang organized and led the team for this book, checking and approving it regularly. He examined and verified the accuracy of the content, the depth of the technology stated, and the readability of this book, and gave feedback for revisions. This book was actually written by five engineers in the Inspur-Intel China Parallel Computing Joint Lab: Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. The first chapter was written by Bo Shen. The second chapter was written by Qing Wu and Bo Shen. The third through fifth chapters were written by Bo Shen, and Yajuan Wang participated. The sixth chapter was written by Qing Wu. The seventh chapter was written by Xiaowei Lu. The eighth chapter was written by Guangyong Zhang, and Bo Shen and Yajuan Wang participated. The ninth chapter was written by Guangyong Zhang. The tenth chapter was written by Bo Shen. The eleventh chapter was written by Xiaowei Lu and Guangyong Zhang. In the later stage, this book was finally approved by Endong Wang, Qing Zhang, Dr. Warren from Intel, and Dr. Victor Lee.

The whole source code has been tested by the authors of this book, but because of the initial stage of MIC technology, we cannot ensure that these codes will be applicable in the latest release. Hence, if any updates come out for the compiler and the execution environment of MIC, please consult the corresponding version manual by Intel.

Acknowledgments

The publication of this book is the result of group cooperation. We would like to show our respect to the people who gave their full support to the composition and publication.

We must express our heartfelt thanks to Inspur Group and Intel Corporation, who gave us such a good platform and offered the working opportunity in the

Inspur-Intel China Parallel Computing Joint Lab. We are fortunate to be able to do research on MIC technology.

We are grateful for the support of the leadership of Inspur Group, especially to the director of the HPC Center, Inspur Group, Jun Liu, who supplied us with financial support and solicitude.

We are grateful to Michael Casscles, Dr. Wanqing He, Hongchang Guo, Dr. David Scott, Xiaoping Duan, and Dr. Victor Lee for their support of the technology and resources for our daily work in the parallel computing joint lab. We especially can't forget Wanqing! He supplied us with plenty of guidance from experience before writing this book. We are also grateful to Dr. Raj Hazra, GM Technical Computing in Intel Corporation, and Joe Curley, MD Technical Computing in Intel Corporation, for their support of the Inspur-Intel China Parallel Computing Joint Lab.

We are grateful to our application users: BGP Inc., China National Petroleum Corp, Institute of Biophysics, Chinese Academy of Sciences, Northwestern Polytechnical University, Chinese Academy of Meteorological Sciences, and Shandong University—especially Prof. Fei Sun and Dr. Kai Zhang from the Institute of Biophysics Chinese Academy of Sciences—and Profs. Chengwen Zhong and Qinjian Li from Northwestern Polytechnical University. The cases in this book come from them.

We are grateful to Inspur Group and Intel Corporation for their support, especially the managers Yongchang Jiang and Ying Zhang from the High-Efficiency Server Department, Inspur Group, who were able to save us a great deal of time.

We thank very much Dr. Haibo Xie and Xiaozhe Yang; we are unable to forget this pleasant time.

We are grateful to the families of the authors for their consideration and patience.

We thank the editors from China WaterPower Press, especially to the editor Chunyuan Zhou and editor Yan Li for their tolerance of our demands. This book could not possibly be published without their hard work.

We are very grateful for the English translation made by Professor David A. Yuen and his team from the University of Minnesota, Twin Cities, China University of Geosciences, Wuhan, consisting of Qiang (Chuck) Li, Liang (Larry Beng) Zheng, Siqi (Sergei) Zhang, and Caroline Qian. Jed Brown and Karli Rupp from Argonne National Laboratory also gave very useful advice, and finally, Prof. Xiaowen Chu and Dr. Kayiyong Zhao from Hong Kong Baptist University are to be thanked for their help in proofreading of the last few chapters.

Lastly, we are grateful to all the others whom we have not acknowledged.

MIC technology has just come out, so there are undoubtedly some mistakes to be found in this book. We apologize for this and look forward to any suggestions from our readers. This is the first book ever written in any language on MIC technology; it was published in the fall of 2012, and is to be contrasted with the newer books coming out from the USA in 2013 bearing the names of Intel Xeon Phi coprocessor.

Beijing, China Qing Zhang

Contents

Part I Fundamental Concepts of MIC

1 High-Performance Computing with MIC . . . 3
1.1 A History of the Development of Multi-core and Many-Core Technology . . . 3
1.2 An Introduction to MIC Technology . . . 7
1.3 Why Does One Choose MIC? . . . 8
1.3.1 SMP . . . 9
1.3.2 Cluster . . . 9
1.3.3 GPGPU . . . 9

2 MIC Hardware and Software Architecture . . . 13
2.1 MIC Hardware Architecture . . . 14
2.1.1 Definitions . . . 14
2.1.2 Overview of MIC Hardware Architecture . . . 14
2.1.3 The MIC Core . . . 18
2.1.4 Ring . . . 30
2.1.5 Clock . . . 30
2.1.6 Page Tables . . . 31
2.1.7 System Interface . . . 32
2.1.8 Performance Monitoring Unit and Event Manager . . . 38
2.1.9 Power Management . . . 39
2.2 Software Architecture of MIC . . . 39
2.2.1 Overview . . . 39
2.2.2 Bootstrap . . . 41
2.2.3 Linux Loader . . . 43
2.2.4 μOS . . . 43
2.2.5 Symmetric Communication Interface . . . 45
2.2.6 Host Driver . . . 45
2.2.7 Sysfs Node . . . 48
2.2.8 MIC Software Stack of MPI Applications . . . 49
2.2.9 Application Programming Interfaces . . . 56

3 The First MIC Example: Computing Π . . . 57

4 Fundamentals of OpenMP and MPI Programming . . . 61
4.1 OpenMP Foundation . . . 61
4.1.1 A Brief Introduction to OpenMP . . . 62
4.1.2 OpenMP Programming Module . . . 62
4.1.3 Brief Introduction to OpenMP Grammar . . . 62
4.2 Message-Passing Interface Basics . . . 67
4.2.1 Start and End MPI Library . . . 69
4.2.2 Getting Information About the Environment . . . 70
4.2.3 Send and Receive Messages . . . 70

5 Programming the MIC . . . 73
5.1 MIC Programming Models . . . 73
5.2 Application Modes . . . 74
5.2.1 CPU in Native Mode . . . 76
5.2.2 CPU Primary, MIC Secondary Mode . . . 76
5.2.3 CPU and MIC “Peer-to-Peer” Mode . . . 77
5.2.4 MIC Primary, CPU Secondary Mode . . . 77
5.2.5 MIC-Native Mode . . . 78
5.3 Basic Syntax of MIC . . . 80
5.3.1 Offload . . . 80
5.3.2 Declarations of Variables and Functions . . . 100
5.3.3 Header File . . . 102
5.3.4 Environment Variables . . . 102
5.3.5 Compiling Options . . . 103
5.3.6 Other Questions . . . 105
5.4 MPI on MIC . . . 105
5.4.1 MPI on MIC . . . 105
5.4.2 MPI Programming on MIC . . . 106
5.4.3 MPI Environment Setting on MIC . . . 108
5.4.4 Compile and Run . . . 111
5.4.5 MPI Examples on MIC . . . 111
5.5 SCIF Programming . . . 114
5.5.1 What Is SCIF? . . . 114
5.5.2 Basic Concepts of SCIF . . . 114
5.5.3 Communication Principles of SCIF . . . 116
5.5.4 SCIF’s API Functions . . . 118

6 Debugging and Profiling Tools for the MIC . . . 123
6.1 Intel’s MIC-Supported Tool Chains . . . 123
6.2 MIC Debugging Tool IDB . . . 124
6.2.1 Overview of IDB . . . 124
6.2.2 IDB Interface . . . 124
6.2.3 IDB Support and Requirements for MIC . . . 125
6.2.4 Debugging MIC Programs Using IDB . . . 125
6.3 MIC Profiling Tool VTune . . . 149

7 Intel Math Kernel Library 167
7.1 Introduction to the Intel Math Kernel Library 167
7.2 Using Intel MKL on MIC 169
7.2.1 Compiler-Aided Offload 169
7.2.2 Automatic Offload Mode 171
7.3 Using FFT on the MIC 175
7.3.1 Introduction to FFT 175
7.3.2 A Method to Use FFT on the MIC 175
7.3.3 Another Method to Use FFT on the MIC 178
7.4 Use BLAS on the MIC 184
7.4.1 A Brief Introduction to BLAS 184
7.4.2 How to Call BLAS on the MIC 185

Part II Performance Optimization

8 Performance Optimization on MIC 191
8.1 MIC Performance Optimization Strategy 191
8.2 MIC Optimization Methods 193
8.2.1 Parallelism Optimization 193
8.2.2 Memory Management Optimization 196
8.2.3 Data Transfer Optimization 199
8.2.4 Memory Access Optimization 212
8.2.5 Vectorization Optimization 216
8.2.6 Load Balance Optimization 225
8.2.7 Extensibility of MIC Threads Optimization 228

9 MIC Optimization Example: Matrix Multiplication 231
9.1 Series Algorithm of Matrix Multiplication 231
9.2 Multi-thread Matrix Multiplication Based on OpenMP 233
9.3 Multi-thread Matrix Multiplication Based on MIC 234
9.3.1 Basic Version 234
9.3.2 Vectorization Optimization 235
9.3.3 SIMD Instruction Optimization 236
9.3.4 Block Matrix Multiplication 237

Part III Project Development

10 Developing HPC Applications Based on the MIC 259
10.1 Hotspot Testing 260
10.1.1 Preparation 260
10.1.2 Hotspot Locating and Testing 261
10.2 Program Analysis 264
10.2.1 Analysis of Program Port Modes 264
10.2.2 Analysis of Size of the Computation 264
10.2.3 Characteristic Analysis 265
10.2.4 Parallel Analysis of Hotspots 267

10.2.5 Vectorization Analysis 270
10.2.6 MIC Memory Analysis 270
10.2.7 Program Analysis Summary 271
10.3 MIC Program Development 271
10.3.1 OpenMP Parallelism Based on the CPU 272
10.3.2 Thread Extension Based on MIC 273
10.3.3 Coordination Parallelism Based on Single-Node CPU+MIC Mode 273
10.3.4 MIC Cluster Parallelism 274

11 HPC Applications Based on MIC 277
11.1 Parallel Algorithms of Electron Tomography Three-Dimensional Reconstruction Based on Single-Node CPU+MIC Mode 278
11.1.1 Electron Tomography Three-Dimensional Reconstruction Technology and Introduction of SIRT Algorithms 278
11.1.2 Analysis of the Sequential SIRT Program 281
11.1.3 Development of a Parallel SIRT Program Based on OpenMP 285
11.1.4 Development of Parallel SIRT Programs Based on the MIC 287
11.1.5 Design of the Heterogeneous and Hybrid Architecture of CPU+MIC Mode Based on Single Nodes and Multiple Cards 291
11.2 Parallel Algorithms of Large Eddy Simulation Based on the Multi-node CPU+MIC Mode 296
11.2.1 Large Eddy Simulation Based on the Lattice Boltzmann Method 296
11.2.2 Analysis of Large Eddy Simulation Sequential (Serial) Program 301
11.2.3 Parallel Algorithm of Large Eddy Simulation Based on OpenMP 303
11.2.4 Parallel Algorithm of Large Eddy Simulation Based on MIC 306
11.2.5 Parallel Algorithm of Large Eddy Simulation Based on Multi-nodes and CPU+MIC Hybrid Platform 309

Further Reading 323

Appendix: Installation and Environment Configuration of MIC 325

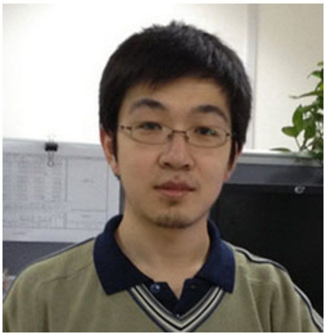
Index 335

Introduction to the Authors

Endong Wang is both a Director and Professor of the Inspur-Intel China Parallel Computing Joint Lab in Beijing, China. He has received a special award from the China State Council, and is also a member of a national advanced computing technology group of 863 experts, the director of the State Key Laboratory for high-efficiency server and storage technology, Senior Vice President of the Inspur group, the chairman of the Chinese Committee of the International Federation for Information Processing (IFIP), and Vice President of the China Computer Industry Association. He is the winner of the National Scientific and Technology Progress Award as the first inventor in three projects, the winner of the Ho Leung Ho Lee Science and Technology innovation award in 2009, and has garnered 26 national invention patents.

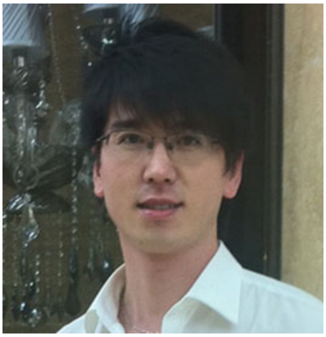
Qing Zhang has a master's degree in computer science from Huazhong Technical University in Wuhan and is now a chief engineer of the Inspur-Intel China Parallel Computing Joint Lab. He is manager of HPC application technology in Inspur Group—which engages in HPC, parallel computing, CPU multi-core, GPU, and MIC technology—and is in charge of many heterogeneous parallel computing projects in life sciences, petroleum, meteorology, and finance.

Bo Shen is a senior engineer of the Inspur-Intel China Parallel Computing Joint Lab, and is engaged in high-performance algorithms, research, and application of software development and optimization. He has many years of experience concerning the development and optimization in life sciences, petroleum, and meteorology.

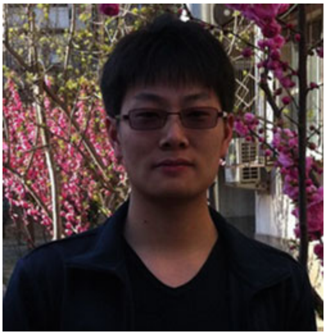


Guangyong Zhang has a master's degree from Inner Mongolia University, majoring in computer architecture, and is now an R&D engineer of the Inspur-Intel China Parallel Computing Joint Lab, engaged in the development and optimization of GPU/MIC HPC application software. He has published many papers in key conference proceedings and journals.

Xiaowei Lu received a master's degree from Dalian University of Technology, where he studied computer application technology, and is now a senior engineer of the Inspur-Intel China Parallel Computing Joint Lab, engaged in the algorithm transplantation and optimization in many fields. He is experienced in high-performance heterogeneous coordinate computing development.



Qing Wu has a master's degree from Jilin University in Changchun and is now a senior engineer of the Inspur-Intel China Parallel Computing Joint Lab, engaged in high-performance parallel computing algorithm and hardware architecture as well as software development and optimization. He led many transplantation and optimization projects concerning the heterogeneous coordinate computing platform in petroleum.



Yajuan Wang has a master's degree from the Catholic University of Louvain, majoring in artificial intelligence. She is now a senior engineer of the Inspur-Intel China Parallel Computing Joint Lab, and is heavily involved in artificial intelligence and password cracking.