PARALLEL PROGRAMMING AND OPTIMIZATION WITH INTEL® XEON PHI™ COPROCESSORS

HANDBOOK ON THE DEVELOPMENT AND OPTIMIZATION OF PARALLEL APPLICATIONS FOR INTEL® XEON® PROCESSORS AND INTEL® XEON PHI™ COPROCESSORS

SECOND EDITION

COLFAX INTERNATIONAL

ANDREY VLADIMIROV | RYO ASAI | VADIM KARPUSENKO

PARALLEL PROGRAMMING AND OPTIMIZATION WITH INTEL® XEON PHI™ COPROCESSORS

HANDBOOK ON THE DEVELOPMENT AND OPTIMIZATION OF PARALLEL APPLICATIONS FOR INTEL® XEON® PROCESSORS AND INTEL® XEON PHI™ COPROCESSORS

Second Edition

Andrey Vladimirov, Ryo Asai and Vadim Karpusenko

Electronic book built: May 14, 2015
Last revision date: May 13, 2015

Copyrighted Material

Published by Colfax International, 750 Palomar Ave, Sunnyvale, CA 94085, USA.

Terms of Use

This book is available in the electronic version and in the printed version. Both versions are accompanied by a set of practical exercises available as an electronic archive. The book and the practical exercises may be used under the following terms:

1. You may use book and the code of the practical exercises for your own education.
2. If you use this book and/or practical exercises to teach a course,
   a) every student must purchase their own copy of the book, OR
   b) you must obtain written authorization from the copyright holder.
3. If you wish to use significant portions of the code of the practical exercises for derivative works, you must obtain written authorization from us.
4. You MAY NOT distribute the electronic version of the book or the source code of the "labs".
5. If you own a printed version of the book, you may lend it to other people, and the borrowers of the book may download the labs as described in Section 6.2 use them under the terms described here. This applies to individual book owners and to libraries (i.e., institutional book owners).

Disclaimer and Legal Notices

While best efforts have been used in preparing this book, the publisher makes no representations or warranties of any kind and assumes no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaims any implied warranties of merchantability or fitness of use for a particular purpose. The publisher shall not be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information or programs contained herein. No warranty may be created or extended by sales representatives or written sales materials.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Results have been simulated and are provided for informational purposes only. Results were derived using simulations run on an architecture simulator or model. Any difference in system hardware or software design or configuration may affect actual performance.

Because of the evolutionary nature of technology, knowledge and best practices described at the time of this writing, may become outdated or simply inapplicable at a later date. Summaries, strategies, tips and tricks are only recommendations by the publisher, and reading this eBook does not guarantee that one's results will exactly mirror our own results. Every company is different and the advice and strategies contained herein may not be suitable for your situation. References are provided for informational purposes only and do not constitute endorsement of any websites or other sources.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

ISBN: 978-0-9885234-2-5

About the Authors

Andrey Vladimirov, PhD, is Head of HPC Research at Colfax International. His primary interest is the application of modern computing technologies to computationally demanding scientific problems. Prior to joining Colfax, A. Vladimirov was involved in computational astrophysics research at Stanford University, North Carolina State University, and the Ioffe Institute in Russia, where he studied cosmic rays, collisionless plasmas and the interstellar medium using computer simulations.

Ryo Asai, is a Researcher at Colfax International. He develops optimization methods for scientific applications targeting emerging parallel computing platforms, computing accelerators and interconnect technologies. Ryo holds a B.S. degree in Physics from University of California, Berkeley.

Vadim Karpusenko, PhD, is Principal HPC Research Engineer at Colfax International involved in training and consultancy projects on data mining, software development and statistical analysis of complex systems. His research interests are in the area of physical modeling with HPC clusters, highly parallel architectures, and code optimization. Vadim holds a PhD from North Carolina State University for his research in in the field of computational biophysics on the free energy and stability of helical secondary structures of proteins.

Additional publications by these authors
related to Intel MIC architecture programming
may be found at
http://research.colfaxinternational.com/

Acknowledgements

Second Edition

We cannot thank enough the people who have contributed their valuable time and expertise to write technical reviews of the 2nd edition of this book. They have provided guidance, fixed misconceptions, future-proofed the messages and caught countless bugs: **Ilya Burylov, Gennady Fedorov, Alexandr Kalinkin, Alexandr Kobotov, Vadim Pirogov** (Intel/MKL), **Joseph Curley** (Intel), **Rob Farber** (TechEnablement.com), **Rakesh Krishnaiyer** (Intel), **Lawrence Meadows** (Intel), **John Pennycook** (Intel), **Troy Porter** (Stanford University), **Frances Roth** (Intel), **Jason Sewall** (Intel), **Gergana Slavova** (Intel). Thank you all very much!

First Edition

Authors are sincerely grateful to **James Reinders** for supervising and directing the creation of this book, **Albert Lee** for his help with editing and error checking, to specialists at Intel Corporation who contributed their time and shared with the authors their expertise on the MIC architecture programming: **Bob Davies**, **Shannon Cepeda**, **Pradeep Dubey**, **Ronald Green**, **James Jeffers**, **Taylor Kidd**, **Rakesh Krishnaiyer**, **Chris (CJ) Newburn**, **Kevin O'Leary**, **Zhang Zhang**, and to a great number of people, mostly from Colfax International and Intel, who have ensured that gears were turning and bits were churning during the production of the book, including **Rajesh Agny**, **Mani Anandan**, **Joe Curley**, **Roger Herrick**, **Richard Jackson**, **Mike Lafferty**, **Thomas Lee**, **Belinda Liviero**, **Gary Paek**, **Troy Porter**, **Tim Puett**, **John Rinehimer**, **Gautam Shah**, **Manish Shah**, **Bruce Shiu**, **Jimmy Tran**, **Achim Wengeler**, and **Desmond Yuen**.

Contents

Foreword to the First Edition

We live in exciting times; the amount of computing power available for sciences and engineering is reaching enormous heights through parallel computing. Parallel computing is driving discovery in many endeavors, but remains a relatively new area of computing. As such, software developers are part of an industry that is still growing and evolving as parallel computing becomes more commonplace.

The added challenges involved in parallel programming are being eased by four key trends in the industry: emergence of better tools, wide-spread usage of better programming models, availability of significantly more hardware parallelism, and more teaching material promising to yield better-educated programmers. We have seen recent innovations in tools and programming models including OpenMP and Intel Threading Building Blocks. Now, the Intel® Xeon Phi™ coprocessor certainly provides a huge leap in hardware parallelism with its general purpose hardware thread counts being as high as 244 (up to 61 cores, 4 threads each).

This leaves the challenge of creating better-educated programmers. This handbook from Colfax, with a subtitle of "Handbook on the Development and Optimization of Parallel Applications for Intel Xeon Processors and Intel Xeon Phi Coprocessors" is an example-based course for the optimization of parallel applications for platforms with Intel Xeon processors and Intel Xeon Phi coprocessors.

This handbook serves as practical training covering understandable computing problems for C and C++ programmers. The authors at Colfax have developed sample problems to illustrate key challenges and offer their own guidelines to assist in optimization work. They provide easy to follow instructions that allow the reader to understand solutions to the problems posed as well as inviting the reader to experiment further. Colfax's examples and guidelines complement those found in our recent book on programming the Intel Xeon Phi Coprocessor by Jim Jeffers and myself by adding another perspective to the teaching materials available from which to learn.

In the quest to learn, it takes multiple teaching methods to reach everyone. I applaud these authors in their efforts to bring forth more examples to enable either self-directed or classroom oriented hands-on learning of the joys of parallel programming.

James R. Reinders
Co-author of "Intel® Xeon Phi™ Coprocessor High Performance Programming"

Intel Corporation
March 2013

Preface to the Second Edition

A lot has happened in Intel's "parallel universe" since the publication of the first edition of this book in March 2013. The family of Intel Xeon Phi coprocessors has grown to three series: 3100, 5100 and 7100, offering a range of performance tiers and prices. Active-cooling Intel Xeon Phi coprocessors were introduced, allowing workstation users to take advantage of the Intel Many Integrated Core (MIC) architecture. Plans were released for future Intel MIC architecture products, based on the Knights Landing chip, and capable of acting as a stand-alone CPU. In the CPU domain, Intel Xeon processors based on the Haswell architecture were released, supporting a new instruction set AVX2 and new functionality.

On the software tools side, the Intel Parallel Studio XE 2015 suite was improved to accommodate the new parallel framework standards: OpenMP 4.0 and MPI 3.0. The evolution of Intel VTune Amplifier XE has added many useful functions for automated diagnostics of performance issues. Intel compilers produce more user-friendly optimization reports than before, and have become even smarter about automatic vectorization and other optimizations.

The work in the users' domain did not stand still, either. With a large number of case studies and research articles on applications for the Intel MIC architecture, it is accurate to say that the developer ecosystem has been established. We are proud to say that Colfax has made a considerable contribution to this progress with the first edition of "Parallel Programmin and Optimization with Intel Xeon Phi Coprocessors". In the years 2013 and 2014, over 1000 science and industry experts at tens of locations across North America have been students of the Colfax Developer Training based on this book. Their experience and feedback, along with the innovations in the Intel tools, have built a solid case for the publication of the second edition of "Parallel Programming and Optimization with Intel Xeon Phi Coprocessors".

Among the numerous new features of the second edition, the ones that stand out are:

1. The details unveiled by Intel of the present and future MIC processors, including Knights Landing;

2. Discussion of configuration and system administration of clusters with Intel Xeon Phi coprocessors, including InfiniBand support, bridged network configuration and storage setup;

3. Additional applications based on case studies of our research in 2013–2014 included in the text as references, as well as practical exercises;

4. Console listings, example codes and hyperlinks to online manuals accurate as of Intel Parallel Studio XE 2015, Intel MPSS 3.4.1 and CentOS 7.0 Linux;

5. New programming models made available in OpenMP 4.0;

6. Deeper review of the Intel Math Kernel Library support for the MIC architecture;

7. More convenient page format and font size for on-screen reading, and

8. Numerous updates to the text improving the clarity and depth of the discussion.

We hope that you find this book to be a valuable resource on "all things Xeon Phi", and, as always, we value your feedback. The HPC research department of Colfax International can be reached by email at phi@colfax-intl.com, and the latest updates on our work can be found at research.colfaxinternational.com.

Preface to the First Edition

Welcome to the Colfax Developer Training! You are holding in your hands or browsing on your computer screen a comprehensive set of training materials for this training program. This document will guide you to the mastery of parallel programming with Intel® Xeon® family products: Intel® Xeon® processors and Intel® Xeon Phi™ coprocessors. The curriculum includes a detailed presentation of the programming paradigm for Intel Xeon product family, optimization guidelines, and hands-on exercises on systems equipped with Intel Xeon Phi coprocessors, as well as instructions on using Intel® software development tools and libraries included in Intel® Parallel Studio XE.

These training materials are targeted toward developers familiar with C/C++ programming in Linux. Developers with little parallel programming experience will be able to grasp the core concepts of this subject from the detailed commentary in Chapter 3. For advanced developers familiar with multi-core and/or GPU programming, the training offers materials specific to the Intel compilers and Intel Xeon family products, as well as optimization advice pertinent to the Many Integrated Core (MIC) architecture.

We have written these materials relying on key elements for efficient learning: practice and repetition. As a consequence, the reader will find a large number of code listings in the main section of these materials. In the extended Appendix, we provided numerous hands-on exercises that one can complete either under an instructor's supervision, or autonomously in a self-study training.

This document is different from a typical book on computer science, because we intended it to be used as a lecture plan in an intensive learning course. Speaking in programming terms, a typical book traverses material with a "depth-first algorithm", describing every detail of each method or concept before moving on to the next method. In contrast, this document traverses the scope of material with a "breadth-first" algorithm. First, we give an overview of multiple methods to address a certain issue. In the subsequent chapter, we re-visit these methods, this time in greater detail. We may go into even more depth down the line. In this way, we expect that students will have enough time to absorb and comprehend the variety of programming and optimization methods presented here. The course road map is outlined in the following list.

- Chapter 1 presents the Intel Xeon Phi architecture overview and the environment provided by the MIC Platform Software Stack (MPSS) and Intel Parallel Studio XE on Many Integrated Core architecture (MIC). The purpose of Chapter 1 is

to outline what users may expect from Intel Xeon Phi coprocessors (technical specifications, software stack, application domain).

- Chapter 2 allows the reader to experience the simplicity of Intel Xeon Phi usage early on in the program. It describes the operating system running on the coprocessor, with the compilation of native applications, and with the language extensions and CPU-centric codes that utilize Intel Xeon Phi coprocessors: offload and virtual-shared memory programming models. In a nutshell, Chapter 2 demonstrates how to write serial code that executes on Intel Xeon Phi coprocessors.

- Chapter 3 introduces Single Instruction Multiple Data (SIMD) parallelism and automatic vectorization, thread parallelism with OpenMP and Intel Cilk Plus, and distributed-memory parallelization with MPI. In brief, Chapter 3 shows how to write parallel code (vectorization, OpenMP, Intel Cilk Plus, MPI).

- Chapter 4 re-iterates the material of Chapter 3, this time delving deeper into the topics of parallel programming and providing example-based optimization advice, including the usage of the Intel Math Kernel Library. This chapter is the core of the training. The topics discussed in this Chapter 4 include:

  i) scalar optimizations;
  ii) improving data structures for streaming, unit-stride, local memory access;
  iii) guiding automatic vectorization with language constructs and compiler hints;
  iv) reducing synchronization in task-parallel algorithms by the use of reduction;
  v) avoiding false sharing;
  vi) increasing arithmetic intensity and reducing cache misses by loop blocking and recursion;
  vii) exposing the full scope of available parallelism;
  viii) controlling process and thread affinity in OpenMP and MPI;
  ix) reducing communication through data persistence on coprocessor;
  x) scheduling practices for load balancing across cores and MPI processes;
  xi) optimized Intel Math Kernel Library function usage, and other.

  If Chapter 3 demonstrated how to write parallel code for Intel Xeon Phi coprocessors, then Chapter 4 shows how to make this parallel code run fast.

- Chapter 6 summarizes the course and provides pointers to additional resources.

Throughout the training, we emphasize the concept of portable parallel code. Portable parallelism can be achieved by designing codes in a way that exposes the data and task

parallelism of the underlying algorithm, and by using language extensions such as OpenMP pragmas and Intel Cilk Plus. The resulting code can be run on processors as well as on coprocessors, and can be ported with only recompilation to future generations of multi- and many-core processors with SIMD capabilities. Even though the Colfax Developer Training program touches on low-level programming using intrinsic functions, it focuses on achieving high performance by writing highly parallel code and utilizing the Intel compiler's automatic vectorization functionality and parallel frameworks.

The handbook of the Colfax Developer Training is an essential component of a comprehensive, hands-on course. While the handbook has value outside a training environment as a reference guide, the full utility of the training is greatly enhanced by students' access to individual computing systems equipped with Intel Xeon processors, Intel Xeon Phi coprocessors and Intel software development tools. Please check the Web page of the Colfax Developer training for additional information: http://www.colfax-intl.com/xeonphi/

Welcome to the exciting world of parallel programming!

IT IS ALL ABOUT OPTIMIZING PARALLELISM

Parallelism has long been a nonnegotiable requirement of all high performance computing applications in supercomputer sites. These days, parallel computing is also becoming commonplace in smaller computing environments: private clusters, workstations and portable computers. Computer architectures grow in size (more compute nodes in a cluster, more cores on a chip) and evolve in depth (wider SIMD registers, deeper pipelines). Harnessing this rocketing growth of hardware capabilities to tackle new fascinating computing problems requires that software developers continually learn to optimize their applications to utilize all available levels and scope of hardware parallelism.

In Parallel Computing and Optimization with Intel Xeon Phi Coprocessors, Colfax International presents to high performance application developers the state-of-the-art programming paradigms and best optimization practices for modern computing platforms based on the Intel multi-core and Many Integrated Core (MIC) architectures.

In this example-based intensive guide to programming Intel Xeon Phi coprocessors, you will find:

- An overview of Intel MIC processors of the first generation (Knights Corner) and second generation (Knights Landing);

- Introduction to task- and data-parallel programming with MPI, OpenMP, Intel Cilk Plus, and automatic vectorization with Intel C and C++ compiler;

- Extensive discussion of high performance application optimization on the Intel Xeon and Intel Xeon Phi platforms, including scalar optimizations, improvement of SIMD operations, multithreading, efficient cache utilization, communication control, and scaling across heterogeneous distributed-memory computing systems;

- A discussion of system administration tasks for workstations and clusters with Intel Xeon Phi coprocessors;

- Supplementary code for practical exercises (self-study "labs") comprising 30 guided exercises with solutions, also used in the Colfax Developer Training program.

www.colfax-intl.com