USING OPENCL

Advances in Parallel Computing

This book series publishes research and development results on all aspects of parallel computing. Topics may include one or more of the following: high-speed computing architectures (Grids, clusters, Service Oriented Architectures, etc.), network technology, performance measurement, system software, middleware, algorithm design, development tools, software engineering, services and applications.

Series Editor:
Professor Dr. Gerhard R. Joubert

Volume 21

Recently published in this series

Vol. 20. I. Foster, W. Gentzsch, L. Grandinetti and G.R. Joubert (Eds.), High Performance Computing: From Grids and Clouds to Exascale
Vol. 19. B. Chapman, F. Desprez, G.R. Joubert, A. Lichnewsky, F. Peters and T. Priol (Eds.), Parallel Computing: From Multicores and GPU's to Petascale
Vol. 18. W. Gentzsch, L. Grandinetti and G. Joubert (Eds.), High Speed and Large Scale Scientific Computing
Vol. 17. F. Xhafa (Ed.), Parallel Programming, Models and Applications in Grid and P2P Systems
Vol. 16. L. Grandinetti (Ed.), High Performance Computing and Grids in Action
Vol. 15. C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr and F. Peters (Eds.), Parallel Computing: Architectures, Algorithms and Applications

Volumes 1–14 published by Elsevier Science.

ISSN 0927-5452 (print)
ISSN 1879-808X (online)

Using OpenCL

Programming Massively Parallel Computers

Janusz Kowalik

16477-107th PL NE, Bothell, WA 98011, USA

and

Tadeusz Puźniakowski

UG, MFI, Wit Stwosz Street 57, 80-952 Gdańsk, Poland

IOS Press

Amsterdam • Berlin • Tokyo • Washington, DC

ISBN 978-1-61499-029-1 (print)
ISBN 978-1-61499-030-7 (online)
Library of Congress Control Number: 2012932792
doi:10.3233/978-1-61499-030-7-i

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

This book is dedicated to Alex, Bogdan and Gabriela
with love and consideration.

Preface

This book contains the most important and essential information required for designing correct and efficient OpenCL programs. Some details have been omitted but can be found in the provided references. The authors assume that readers are familiar with basic concepts of parallel computation, have some programming experience with C or C++ and have a fundamental understanding of computer architecture. In the book, all terms, definitions and function signatures have been copied from official API documents available on the page of the OpenCL standards creators.

The book was written in 2011, when OpenCL was in transition from its infancy to maturity as a practical programming tool for solving real-life problems in science and engineering. Earlier, the Khronos Group successfully defined OpenCL specifications, and several companies developed stable OpenCL implementations ready for learning and testing. A significant contribution to programming heterogeneous computers was made by NVIDIA which created one of the first working systems for programming massively parallel computers – CUDA. OpenCL has borrowed from CUDA several key concepts. At this time (fall 2011), one can install OpenCL on a heterogeneous computer and perform meaningful computing experiments. Since OpenCL is relatively new, there are not many experienced users or sources of practical information. One can find on the Web some helpful publications about OpenCL, but there is still a shortage of complete descriptions of the system suitable for students and potential users from the scientific and engineering application communities.

Chapter 1 provides short but realistic examples of codes using MPI and OpenMP in order for readers to compare these two mature and very successful systems with the fledgling OpenCL. MPI used for programming clusters and OpenMP for shared memory computers, have achieved remarkable worldwide success for several reasons. Both have been designed by groups of parallel computing specialists that perfectly understood scientific and engineering applications and software development tools. Both MPI and OpenMP are very compact and easy to learn. Our experience indicates that it is possible to teach scientists or students whose disciplines are other than computer science how to use MPI and OpenMP in a several hours time. We hope that OpenCL will benefit from this experience and achieve, in the near future, a similar success.

Paraphrasing the wisdom of Albert Einstein, we need to simplify OpenCL as much as possible but not more. The reader should keep in mind that OpenCL will be evolving and that pioneer users always have to pay an additional price in terms of initially longer program development time and suboptimal performance before they gain experience. The goal of achieving simplicity for OpenCL programming requires an additional comment. OpenCL supporting heterogeneous computing offers us opportunities to select diverse parallel processing devices manufactured by different vendors in order to achieve near-optimal or optimal performance. We can select multi-core CPUs, GPUs, FPGAs and other parallel processing devices to fit the problem we want to solve. This flexibility is welcomed by many users of HPC technology, but it has a price.

Programming heterogeneous computers is somewhat more complicated than writing programs in conventional MPI and OpenMP. We hope this gap will disappear as OpenCL matures and is universally used for solving large scientific and engineering problems.

Acknowledgements

It is our pleasure to acknowledge assistance and contributions made by several persons who helped us in writing and publishing the book.

First of all, we express our deep gratitude to Prof. Gerhard Joubert who has accepted the book as a volume in the book series he is editing, Advances in Parallel Computing. We are proud to have our book in his very prestigious book series.

Two members of the Khronos organization, Elizabeth Riegel and Neil Trevett, helped us with evaluating the initial draft of Chapter 2 Fundamentals and provided valuable feedback. We thank them for the feedback and for their offer of promoting the book among the Khronos Group member companies.

Our thanks are due to NVIDIA for two hardware grants that enabled our computing work related to the book.

Our thanks are due to Piotr Arłukowicz, who contributed two sections to the book and helped us with editorial issues related to using LaTeX and the Blender^{3D} modeling open-source program.

We thank two persons who helped us improve the book structure and the language. They are Dominic Eschweiler from FIAS, Germany and Roberta Scholz from Redmond, USA.

We also thank several friends and family members who helped us indirectly by supporting in various ways our book writing effort.

Janusz Kowalik
Tadeusz Puźniakowski

How to read this book

The text and the source code presented in this book are written using different text fonts. Here are some examples of diffrent typography styles collected.

variable – for example:
... the variable *platform* represents an object of class ...

type or class name – for example:
... the value is always of type cl_ulong...
... is an object of class cl::Platform...

constant or macro – for example:
... the value CL_PLATFORM_EXTENSIONS means that...

function, **method** or **constructor** – for example:
... the host program has to execute **clGetPlatformIDs**...
... can be retrieved using **cl::Platform::getInfo** method...
... the context is created by **cl::Context** construcor...

file name – for example:
... the cl.h header file contains...

keyword – for example:
... identified by the **__kernel** qualifier...

Contents