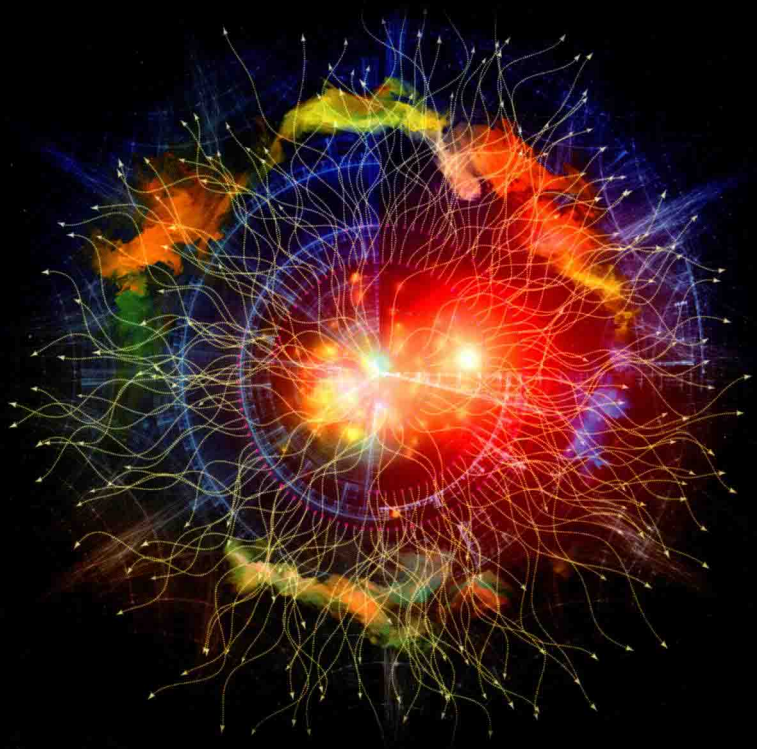
HZ BOOKS
华章IT

工业界和学术界联合鼎力推荐，国内顶级专家和实践者合力打造，所有案例均经过真机环境验证

基于最新的OpenCL 2.0标准，清晰地阐述了OpenCL的API、平台支持、编程模型、通信机制，以及与OpenGL的交互，尤其对OpenCL 2.0新引入的SVM机制、管道和新定义的原子操作机制等进行了深入讲解，辅以大量示例说明，并给出卷积、矩阵乘法等优化实践案例

高性能计算技术丛书



OpenCL Heterogeneous Parallel Computing: from Principle to Practice

OpenCL 异构并行计算

原理、机制与优化实践

刘文志 陈轶 吴长江 著

机械工业出版社
China Machine Press

高性能计算技术丛书

OpenCL Heterogeneous Parallel Computing: from Principle to Practice

OpenCL 异构并行计算

原理、机制与优化实践

刘文志 陈轶 吴长江 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

OpenCL 异构并行计算：原理、机制与优化实践 / 刘文志等著 . —北京：机械工业出版社，2015.12
（高性能计算技术丛书）

ISBN 978-7-111-51934-8

I. O… II. 刘… III. 图形软件－程序设计 IV. TP391.41

中国版本图书馆 CIP 数据核字（2015）第 255335 号

OpenCL 异构并行计算：原理、机制与优化实践

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：高婧雅 责任校对：董纪丽
印 刷：北京市荣盛彩色印刷有限公司 版 次：2016 年 1 月第 1 版第 1 次印刷
开 本：186mm×240mm 1/16 印 张：24.75
书 号：ISBN 978-7-111-51934-8 定 价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

Preface 序 一

随着计算能力的不断提高和可编程性的不断增强，GPU 受到越来越多厂商和开发人员的青睐，应用越来越广泛。无论是在科学计算等传统领域还是在多媒体计算等新兴互联网领域，在融合 CPU 和 GPU 构成的异构计算系统上使用 GPU 实现应用程序加速已经成为提高程序性能的主要模式。同时，主流芯片厂商根据实际计算需求，不断发展自己的 GPU 架构。例如：NVIDIA 的 Fermi、Kepler 和 Maxwell 架构；AMD 的 Cypress、Cayman、GCN 架构等。这些不同架构的 GPU 已经深入到从移动计算领域到超级计算领域的方方面面，异构计算正日益作为新的主流计算机体系结构。

与 CUDA 只能运行在 NVIDIA GPU 上相比，OpenCL 由 Khronos 国际标准组织发布与维护，是一种针对通用并行计算的开放行业标准和跨厂商解决方案。到目前为止，OpenCL 已有包括 Intel、NVIDIA、AMD 在内的众多硬件厂商和软件厂商的支持与维护。随着异构计算的发展，OpenCL 的发展方兴未艾，正逐渐成为异构并行计算领域里异军突起的应用程序编程接口。OpenCL 定义了丰富的 API，应用程序开发人员可以通过使用 OpenCL，以最高效的方式充分利用计算系统中的各种异构计算资源，在实现性能目标的同时又可降低功耗。更重要的是，OpenCL 程序可以运行在不同厂商的各种处理器上，实现了高性能并行程序的可移植。

然而，为了实现上述目标，OpenCL 被设计成一个相对复杂的并行编程标准，其编程充满了各种困难和挑战。本书首先通过简洁、通俗的语句和丰富的代码示例清晰解释了 OpenCL 中比较晦涩难懂的各种概念以及 API 的使用；然后描述了 OpenCL 到主流 GPU 处理器的映射，最后通过二维卷积、矩阵乘法等实际案例的开发和优化，进一步帮助读者加深对 OpenCL 的概念和应用的理解。本书写作以最新的 OpenCL 2.0 为标准，对 SVM 机制、管道、原子操作等新概念进行了非常深入的描述，具有较强的前沿性，这为 OpenCL 开发人员理解、掌握和使用最先进的 OpenCL 技术提供了很大的帮助。

本书的作者是长期战斗在异构编程第一线的架构师和开发者，具有非常丰富的 OpenCL

使用和编程经验，本书正是他们多年OpenCL编程经验的总结。本书不仅详细描述了OpenCL的各种概念和特性；而且通过由浅入深的一系列实际应用案例，帮助读者掌握这个令人激动的新编程模型。本书内容充实，不仅适合不同经验水平的学生和开发者，而且对于致力于异构计算的研究人员，也是一本非常不错的OpenCL教科书。

张云泉　研究员

中国科学院计算技术研究所计算机体系结构国家重点实验室

Preface 序　二

计算机的基础组成在过去的十几年发生了很多变化，从单核处理器发展到多核处理器，然后发展到“众核”处理器，高效性的需求一直促进着处理器的发展，最终走到“异构处理器”，如 CPU 和 GPU 的结合，CPU 和 FPGA 的结合。这一阶段异构处理器的发展和先前的形式产生了本质的区别，先前的形式主要是将多个异构模块（在物理上）叠加到一颗处理器内，而各异构单元间的内存、数据处理和通信还是分开的，如内存控制器、通信机制还是各模块单独设计。现代的异构处理器着眼于将异构处理模块间的编程模型统一、内存统一、数据统一处理，甚至将各异构处理单元统一纳入操作系统的管理之下，最大限度地提升处理器的可编程性和能效比，为此各大芯片厂商也不遗余力地开发自己的异构片上系统（SoC）。

最近我们看到，由于人工智能和机器学习随着移动互联网的兴起，特别是对图片、视频、语音等非结构化数据的挖掘、识别，带动了以智能算法为核心的应用的兴起，“异构平台”成为各大互联网厂商追逐数据挖掘平台先进性的标志之一。这源于异构计算可使数据处理的特定性能实现成百上千倍提升的特点。

作为异构编程人员首选的编程语言模型 OpenCL（Open Computing Language），其将 GPU 计算的能力释放出来，带来了异构计算的新时代，在苹果公司的大力支持下，得到了包括 AMD、Intel 等主流处理器厂家的大力支持，也得到了如 Altera 等主流 FPGA 芯片公司的支持，OpenCL 由 Khronos Group 精心设计维护，由于其开放、高度通用的跨平台设计原则，正在成为异构处理器的性能调优利器和开发语言。试想，以后运行性能优化能够在不同厂家的异构处理器间兼容而无须重写，实现“一次编写，多环境运行”，则可以大大提高开发效率。

在该书出版之际，OpenCL 2.0 的规范也已经发布。本书不但介绍了 OpenCL 2.0 及硬件厂家（如 AMD 的 HSA 架构）对 OpenCL 2.0 的支持情况，还介绍了当前热点移动处理器对异构平台和 OpenCL 的支持情况。这几方面相信都是读者非常感兴趣的。

现在市面上关于 OpenCL 编程的书籍无论翻译的还是国内自己编写的都很多，可见异构计算正在从先前的以科学计算为主导的领域走向更开放的生态系统和应用，这是非常令人欣喜的事情。本书的编写背景和风格与其他书籍稍显不同，本书的几位作者均来自社区，也是活跃在各大 GPU 厂家的资深技术人员和实际项目的开发工程师。他们从自己使用经验的角度来阐述如何构建一个合理优化的 OpenCL 程序。根据本书一步一步讲解的内容来进行 OpenCL 编程，无论你是一个 CPU 编程人员，还是一个 CUDA 编程人员，都能很快地学会 OpenCL 编程。本书中所涉及的案例讲解，作者都在 AMD 的 APU 上逐一编写验证过。本书介绍了完整的 OpenCL 编程模型，同时结合相关的知识、体系结构，形成一个完整的编程知识体系，非常适合工程师阅读和参考，尤其是对正在开发项目的工程师来说，一边阅读，一边动手实践，结合自己的项目实施，一定可以很快掌握。该书也推荐给对异构计算有所耳闻，希望了解它，以借力快速打开异构计算之门和实践的技术爱好者，本书可以帮助他们在计算编程领域更上一层楼。

最后对几位作者在工作之余付出的极大热情和心血表示感谢！欢迎体验异构计算之旅。

楚含进

AMD（中国）异构计算技术总监

Preface 前　言

为什么要写这本书

2007 年 NVIDIA 发布 CUDA，正式开启了利用 GPU 作为大规模数据并行计算的时代。而最近几年 GPU 计算已经完成了从实验室、研究所的研究对象到产业界提高生产力的实际工具的转变。但是 NVIDIA 的 CUDA 并没有得到其他厂商支持，CUDA 代码并不能在其他硬件厂商的产品上运行，而在实际中，用户更希望代码能够同时在多个平台上运行，以减少编码和优化代价。

2008 年，在苹果公司将自己撰写的 OpenCL 草案开放给 Khronos Group（开放标准组织）之后，Khronos Group 在 6 个月的时间内发布了 OpenCL 1.0 标准。这不仅引起了像 Intel、NVIDIA、AMD 这类传统 CPU 和 GPU 处理器厂商的关注，而且还吸引了像 TI 这类做 DSP 的公司，以及 Altera 这类做 FPGA 的公司。因为 OpenCL 将基于 GPU 的高性能计算概念做了更广范的延展，从 NVIDIA 扩展到几乎所有的硬件厂商，从 GPU 扩展到 CPU、DSP 和 FPGA 等，从高性能计算集群扩展到云、桌面和移动，我们称之为异构并行计算，而 GPU 计算是异构并行计算的一种。

目前，即便是 CPU 也能通过 OpenCL 实现其内部的 SIMD 操作，从而能达到更快速的数据处理。程序员通过 OpenCL 这样的编程工具就能达到加速原来数据密集型代码的目的，而无须过多关注底层的硬件特性（如指令集架构等）。OpenCL 给程序员带来了标准、统一的接口来实现任务级并行以及数据级并行的算法处理。

由于目前 OpenCL 编程环境最为成熟的还是 AMD 的 APP 和 NVIDIA 的 CUDA，因此本书主要基于 AMD APP 和 NVIDIA 的 CUDA 编程环境描述，考虑到移动端对 OpenCL 的支持也越来越多，尤其是 ARM 的 Mali GPU 已经引入了广泛的 OpenCL 支持，因此本书会简略介绍 OpenCL 在 Mali GPU 上的编程和优化。

由于 OpenCL 标准本身阅读起来比较晦涩，很多概念也没有完全解释清楚，因此我们写这本书的目的是以更简洁、通俗的语句来表达 OpenCL 中的各种概念，以及各种 API、各种语法的使用，使读者更易理解。同时加入了很多代码示例以及图表以进一步帮助读者加深对这些概念的理解。另外，在写这本书的时候 OpenCL 2.0 标准已经发布将近 1 年了，我们这本书也以最新的 OpenCL 2.0 标准为主，给读者呈现当前最先进的 OpenCL 技术。本书对 OpenCL 2.0 中所新引入的 SVM 机制、管道、原子操作等概念有着非常深入的描述，并且结合大量示例进行剖析。

读者对象

由于移动处理器和 GPU 已经非常便宜，而异构并行计算是未来的趋势，所有 IT 行业的从业者都应当收藏、阅读本书，以增加对 OpenCL 的了解。笔者认为下列人员更应当阅读本书：

- 互联网及传统行业的 IT 从业者；
- 希望将应用移植到移动处理器或 GPU 的开发人员；
- 对向量化和并行化感兴趣的职业工作者；
- 大中专院校、研究所的学生及教授。

如何阅读本书

本书大致的目录结构如下：

第 1 章主要介绍并行计算的发展历程以及 OpenCL 在其中所扮演的角色；

第 2 章和第 3 章介绍了 OpenCL 的大体概念以及它在主机端上 API 的功能和说明；

第 4 章和第 5 章主要描述 OpenCL C 语言的概念以及相关语法点；

第 6 章对 OpenCL 整个同步机制做了一个总结性的整体深入介绍，从主机端的事件同步到内核程序的原子操作，每一种同步方式都做了非常详细的介绍；

第 7 章详细描述了 OpenCL 与 OpenGL 之间的交互；

第 8 章介绍了当前 OpenCL 各大实现厂商对 OpenCL 的各自硬件实现，同时也讲解了各种不同硬件平台上如何有针对性地对 OpenCL 程序做进一步优化；

第 9 章和第 10 章通过几个实际例子为读者展示了 OpenCL 的优化实践以及在实际工程项目中的使用技巧。

阅读本书的读者应当对 C 编程语言有一定程度的理解，最好同时能熟悉基本的计算机

体系结构方面的理论知识。当然，考虑到很多工程学、经济学等方面的专家对计算机相关的理论知识掌握有限，而且此类读者往往更偏向于 OpenCL 工具的运用，因此我们建议这类读者可以略过整个 5.6 节以及整个第 8 章。虽然本书尽可能通俗而又简洁地去介绍大部分 OpenCL 2.0 标准中所涉及的概念，但是很多概念没有一定基础仍然会比较难以理解，因此读者在遇到这样的概念时可以先实践，然后慢慢消化。

勘误和支持

由于笔者的水平有限、工作繁忙、编写时间仓促，而异构并行计算领域正在高速发展中，OpenCL 的标准内容也越来越多，笔者虽已努力确认很多细节，但书中难免会出现一些不准确的地方甚至是错误，恳请读者批评指正。另外，由于我们目前手头上支持 OpenCL 2.0 标准的设备有限，本书中难免还会有一些错误、瑕疵。读者若发现一些明显的错误或者对我们书写的内容有任何疑问、建议，欢迎与我们一同讨论。我们的联系方式为：ly152832912@163.com。

书中有些完整的工程代码可在以下地址下载：www.hzbook.com。

致谢

本书能够出版不仅仅出自我们自己对 OpenCL 的热情和执着，在这里我们还要感谢机械工业出版社华章公司高婧雅对我们的大力支持，没有她，我们也不会想到要编写此书。

感谢赵成龙（网名龙猫）认真帮我们审查草案，以及对草案细节的校正。在此书成书的过程中，如果没有赵成龙的无私帮助，本书可能会晚半年出版。

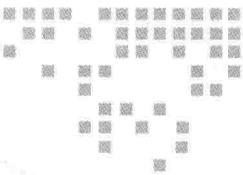
感谢 AMD 大中华区软件合作及解决方案高级经理时昕对整本书的审阅，时总不但名字与时俱进，技术能力也与时俱进，有点让我们这些一线的程序员汗颜了；我们还要感谢 AMD（中国）有限公司免费为我们提供实验设备。

感谢家人对我们写作的支持，感谢他们容忍了我们晚上加班回家后，还要在电脑前写作！

谨以此书献给我最爱的家人，以及众多热爱异构并行计算的朋友们！愿你们快乐地阅读本书！

风辰

目　录 Contents



第 1 章 Chapter 1

异构并行计算的过去、现状和未来

在正式进入本章的主题前，先让我们重温一下异构并行计算的概念。异构并行计算包含两个方面的内容：异构和并行。异构是指：计算单元由不同的多种处理器组成，如 X86 CPU+GPU、ARM CPU+GPU、X86 CPU+FPGA、ARM CPU+DSP 等。并行是指：要发挥异构硬件平台的全部性能必须要使用并行的编程方式。这通常包含两个层次的内容：

1）多个不同架构的处理器同时计算，要发挥异构系统中所有处理器的性能，可通过并行编程使每个处理器都参与运算，避免处理器闲置。相比于只让某一种类型的处理器参与工作，这种方式提高了性能上限，简单举例来说，在 X86 CPU+GPU 平台上，X86 CPU 的计算能力为 1TFLOPS，GPU 的计算能力为 4TFLOPS，如果只使用 GPU，那么最大可发挥的性能是 4TFLOPS，而如果加上 X86 CPU，则最大可发挥的性能是 5TFLOPS。

2）每个处理器都是多核向量处理器[⊖]，这要求使用并行编程以发挥每个处理器的计算能力。通常每个处理器包括多个核心，每个核心包含一个或多个长向量，如 AMD GCN GPU 中就包含数量不等的核心，每个核心包含 4 个向量，每个向量能够同时处理 16 个 4 字节长度的数据。如果没能很好地并行，则可能不能完美地发挥多核和向量化的性能。

作为本书的开篇，本章将主要介绍异构并行计算的历史、现状和未来：

1）异构并行计算的历史。即在异构并行计算出现之前，处理器是如何提升性能，使用了哪些提升性能的方法，这些方法为什么又遇到困难了。读史使人明智，通过了解异构并行计算的历史，读者可以了解到为什么异构并行计算会大行其道，也了解了为什么笔者会编写本书。

2）异构并行计算的现状。今天异构并行计算已经得到充分的发展并且还在进一步快速

⊖ 本书的向量处理器指支持 MIMD 执行或 SIMD 指令集的处理器。

发展中，OpenCL 和其他的异构并行计算工具已经应用到许多图像处理、视频处理及科学计算项目上，而这些工具自身也在快速进化中。近两年，许多科学计算以外的行业和领域（如互联网行业）正在应用异构并行计算来加快研究和产品化的步伐。

3）异构并行计算的未来。计算的未来是异构并行的，异构并行的概念、应用在计算机及相关领域会越来越广。任何参与计算机及相关行业的人员都应当了解并学习异构并行相关的内容。在不久的将来，不懂异构并行计算就意味着不懂计算机。

在具体介绍异构并行计算的历史、现状和未来之前，笔者想介绍几个始终贯穿本书的相关概念：

1）向量化。向量化是一种一条指令同时处理多个数据的方法，从这一点来说，它是一种数据并行技术。主流的向量化技术有两种：SIMD（Single Instruction Multiple Data，单指令多数据）和 SIMT（Single Instruction Multiple Thread，单指令多线程），大多数 CPU（如 AMD Zen 处理器）都使用 SIMD 向量化技术，而大多数 GPU（如 AMD GCN）都使用 SIMT 向量化技术。关于 SIMD 的具体描述请参看图 1-1。

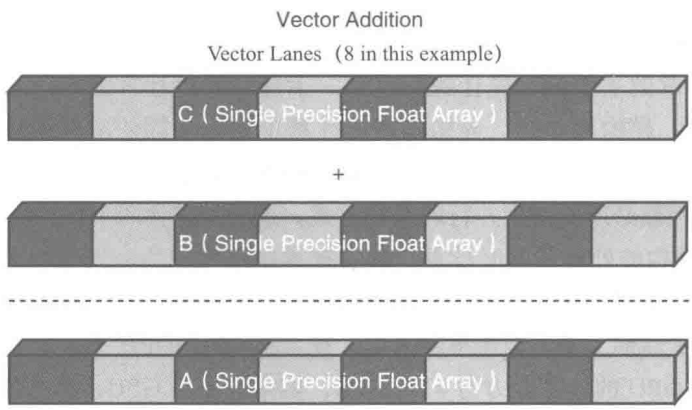


图 1-1　向量化示例

SIMD 操作可简单描述为一些具有如下特点的操作：对两个长向量寄存器中的数据按元素进行操作，结果向量寄存器和源向量寄存器长度相同。例如，对两个长度为 512 位的向量进行 SIMD 操作，按照单精度进行浮点加操作，假设单精度浮点类型占用空间大小为 4 个字节，那么 512 位向量可一次同时处理 16（512 位 /8 位每字节 /4 字节）个单精度浮点数据得到 16 个结果，其中第一个向量的第 1 个元素和第二个向量的第 1 个元素相加产生结果向量的第 1 个元素，第一个向量的第 15 个元素和第二个向量的第 15 个元素相加产生结果向量的第 15 个元素，其余类推。

2）多核。多核是指：在一块芯片上，集成多个处理器核心，这多个处理器核心共享或不共享缓存层次结构。图 1-2 是 ARM 公司设计的 ARM Cortex-A72 多核处理器，从中可以看出其最多具有 4 个核心（为了应对不同细分市场的需求，ARM 处理器核心数量通常可调