

Heterogeneous Computing with OpenCL

# OpenCL

## 异构计算

Benedict R. Gaster Lee Howes David R. Kaeli 著  
Perhaad Mistry Dana Schaa  
张云泉 张先轶 龙国平 姚继锋 译



NLIC2970801772



**MK**  
MORGAN KAUFMANN

清华大学出版社

# OpenCL 异构计算

Benedict R. Gaster Lee Howes David R. Kaeli 著  
Perhaad Mistry Dana Schaa

张云泉 张先轶 龙国平 姚继锋 译



NLIC2970801772

清华大学出版社  
北京

## 内 容 简 介

《OpenCL 异构计算》提供 OpenCL 的第一手资料, 详尽阐述了如何在异构环境下进行并行编程。第 1 章首先介绍如何在并行系统下编程, 定义异构编程需要理解的概念。第 2~4 章循序渐进地介绍 OpenCL 的基本架构。第 5 章和第 6 章对这些概念加以扩展, 旨在帮助读者更好地理解。第 7~10 章提供 4 个更复杂的案例学习, 让读者理解到 OpenCL 具有广泛的应用。第 11~13 章锁定高级主题展开讨论。

本书可帮助学生和研究人员更好地理解通用异构计算(尤其是 OpenCL 提供的解决方案), 尤其是适合不同经验水平的学生, 可以作为 OpenCL 课程的教材或其他课程的参考, 例如并行编程课程和高级课程。

Heterogeneous Computing with OpenCL, 1st Edition  
Benedict R. Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, Dana Schaa  
ISBN: 978-0123877666

Copyright©2012 Advanced Micro Devices, Inc. Published by Elsevier(Inc). All rights reserved.  
Authorized Simplified Chinese translation edition published by Elsevier (Singapore) Pte Ltd Press and Tsinghua University.

Copyright©2012 by Elsevier (Singapore) Pte Ltd and Tsinghua University Press. All rights reserved.  
Published in China by Tsinghua University Press under special arrangement with Elsevier (Singapore) Pte Ltd.. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中国大陆地区(不包括香港、澳门特别行政区以及台湾地区)出版与发行。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。  
版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933  
北京市版权局著作权合同登记号 图字: 01-2012-2719

### 图书在版编目(CIP)数据

OpenCL 异构计算/贾斯特(Gaster,R. B.)等著; 张云泉等译. --北京: 清华大学出版社, 2012.6  
书名原文: Heterogeneous Computing with OpenCL  
ISBN 978-7-302-28685-1

I. ①O… II. ①贾… ②张… III. ①图形软件 IV. ①TP391.41

中国版本图书馆 CIP 数据核字(2012)第 083719 号

责任编辑: 文开琪  
装帧设计: 杨玉兰  
责任校对: 周剑云  
责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>  
地 址: 北京清华大学学研大厦 A 座 邮 编: 100084  
社 总 机: 010-62770175 邮 购: 010-62786544  
投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)  
质量反馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)  
课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者: 清华大学印刷厂  
装 订 者: 三河市新茂装订有限公司  
经 销: 全国新华书店  
开 本: 185mm×230mm  
版 次: 2012 年 6 月第 1 版  
印 数: 1~4000  
定 价: 76.00 元

印 张: 18.75

字 数: 390 千字  
印 次: 2012 年 6 月第 1 次印刷

# 推荐序 1

陈国良

中国科学院院士

我国超级计算机研制近几年奋起直追。2009年，天河一号(TH-1)的发布，使得我国成为继美国之后世界上第二个能够独立研制千万亿次峰值超级计算机的国家。2010年9月发布的国产天河一号A千万亿次超级计算机通过采用先进的CPU+GPGPU的异构混合加速体系架构，以2.56Pflops的Linpack性能夺取2010年11月国际HPC TOP500排行榜的第一名，创下中国国产超级计算机历史上首次夺得世界冠军的佳绩。同时，我国安装的超级计算机的上榜数量也首次超越日本和欧盟，排名世界第二，仅次于美国。目前，中国HPC TOP100排行榜中机器的平均Linpack性能为120Tflops，是2010年63Tflops的1.9倍，而国际上平均性能接近120Tflops的时间为2011年6月(117.76Tflops)，我国与世界平均性能差距从原来保持了很久的一年半差距，快速缩短为半年的差距！这标志着我国超级计算机系统的整机集成水平步入国际领先行列，且在一些核心硬件技术上取得了关键性的突破。2011年底的国际HPC TOP500排行榜中，前十名中的第二、四和第五名都采用了GPGPU异构加速体系架构——500台系统里有39台系统采用了GPGPU异构加速，由此可见，异构计算作为一种新的计算机体系架构逐渐成为主流。

每一次超级计算机体系结构的变化，都会伴随着并行程序设计环境和语言的进化。为了高效地为分布式存储和共享存储的并行计算机编写并行程序，工业界在总结众多各类并行语言的优点的基础上，分别提出了目前已经被广泛接受和使用的并行程序开发环境MPI和OpenMP工业界标准。这两个标准并行程序设计环境和语言的提出和推广普及，极大地提升了大规模并行应用软件的开发效率，降低了开发成本，确保了软件在不同厂家机器之间的可移植性和高性能。随着当前国际上主流体系架构逐渐向异构混合并行的转变，如何为这一新体系架构设计高效的且被广为接受的工业界标准并行程序开发环境和语言，成为一个迫切需要解决的问题。

OpenCL正是在这一背景下由Khronos国际组织提出，并得到众多硬件和软件厂商的支持，成为工业界异构计算的标准语言。虽然目前比较流行的异构计算的语言是CUDA语言，但是，我们相信，随着时间的推移，正如当年早期最流行的PVM分布式存储并

行环境逐渐退出历史舞台从而让位于工业界标准 MPI 一样, OpenCL 在融合了 CUDA 语言众多优点的基础上, 可能逐渐成为异构计算的工业界新标准语言, MPI+OpenMP+OpenCL 这样的并行编程环境亦有望成为未来的主流并行编程模型之一。

该书的翻译很及时, 是目前国内第一本全面介绍 OpenCL 的中文图书。书中既有对 OpenCL 基本概念和原理的介绍, 又有结合具体实例和特定硬件体系架构的深入浅出的讲解, 还探讨了 OpenCL 的一些扩展功能, 是一本难得的全面介绍 OpenCL 的教科书。相信该书的出版必将为广大异构计算爱好者和用户提供有价值的参考, 并对推动异构计算在我国的普及起着重要的作用。

## 推荐序 2

Wen-mei W. Hwu

伊利诺伊大学厄巴纳-香槟分校

电子与计算机工程系 AMD-Sanders 荣誉教授，并行计算所首席科学家

We are at the dawn of the heterogeneous computing era. With all applications being power limited, from mobile to supercomputing, all future computing platforms must embrace heterogeneity. With its adoption across major platforms of the computing industry today, OpenCL is emerging as a leading Application Programming Interface (API) for heterogeneous parallel computing. It allows software developers to optimally use the diverse processors in a heterogeneous system to achieve performance goals while reducing power consumption. It supports all major trends in heterogeneous computing platforms – the CPUs' evolution to multicores, the GPUs' rise in high-throughput computing, and the Accelerated Processing Units' (APUs) fast adoption in mobile devices.

我们正迎来异构计算时代！从移动到超级计算领域，当所有的应用都受限于能耗时，未来所有的计算平台都必须被设计为异构的。随着各大主要计算平台的先后采用，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(API)。它允许软件开发者以最优化的方式充分使用异构系统中多样化的处理器，以实现性能目标的同时又能降低能耗。它支持所有主流的异构计算平台及其发展势头——CPU 向多核化的演进，GPU 计算性能在高吞吐计算领域的崛起，以及 APU(AMD 加速处理器)在移动领域的迅速普及。

In 2011, OpenCL, Heterogeneous Computing with OpenCL was published as the first major textbook on OpenCL. I am glad to see that only half of a year later, the Chinese version is being published by the Tsinghua University Press. In China, experts and pioneers of the technology have already begun to popularize OpenCL to a broader community of software developers, from students to grass-root developers to IT professional. Many well-known manufacturers have committed to support OpenCL and work with early adopters from academia and industry to improve the OpenCL development environment with support

ranging from hardware and platform development, software and SDK release, higher-level programming models, to the initiatives regarding developer community and university curriculum. This textbook is timely in helping accelerate this movement.

在 2011 年,《OpenCL 异构计算》英文版作为第一本 OpenCL 教科书出版。我非常欣喜地看到仅在半年之后,对应的中文译本便由清华大学出版社正式出版。在中国,我们的技术专家和先行者已经开始将 OpenCL 推广到更广阔的软件开发社区,去影响更多的学生、草根开发者和 IT 专业人员。许多知名厂商也纷纷承诺支持 OpenCL,与学术界和产业界携手改善 OpenCL 的开发环境,从硬件平台开发,软件和 SDK 的发布,更上层的编程模式的推广,等等,在诸多方面提供技术支持,并且针对开发者社区和高校课程实施一系列的举措以推动技术的普及。这本教科书的中文译本的出版无疑是非常及时的。

The translation team consists of faculty and PhD students from Parallel Computing Lab of the Chinese Academy of Sciences. With their many years of rich experience in parallel computing, they have become a major group of early adopters and evangelists of OpenCL in China. They are working hard to fulfill the mission of growing the Chinese heterogeneous computing community to embrace the new programming framework for heterogeneous computing.

《OpenCL 异构计算》中文译本的翻译团队来自中国科学院并行计算实验室的部分师生。凭借多年在并行计算领域积累的丰富的研究教学经验,他们已经成为 OpenCL 在中国早期采用和推广 OpenCL 的中坚力量。他们孜孜以求,勤勉地履行着发展中国本土异构计算社区的使命,拥抱这个崭新的开发架构和异构计算。

This book's publication marks a milestone of OpenCL's early journey in China. With this book, I expect to see many more people embrace OpenCL in China as well as to see China's contributions accelerating the worldwide progress of heterogeneous computing. I am honored to write this foreword and be part of this great movement. More importantly, I salute all those who labored to make this advancement possible.

《OpenCL 异构计算》的出版是 OpenCL 早期在中国落地扎根的一个里程碑。通过这本书的出版,我希望看到更多中国开发者能够掌握和应用 OpenCL,并为全球异构计算的发展做出贡献。我非常荣幸能为此书作序,亲历这个伟大的行动。更重要的是,我要在此向为之辛勤付出的各位中国同仁致敬!

# 译者序

张云泉博士

中国科学院软件研究所研究员

以超级计算为工具的计算科学，已经渗透到科学研究与工程设计的各个层面，成为一个具有战略意义的新兴学科。计算科学已成为理论和实验之外，人类进行科学探索和工程研究的第三种手段。为此，美国总统信息技术顾问委员会 PITAC 在 2005 年 6 月向时任总统布什提交的题为“计算科学：确保美国的竞争优势”的报告中，建议美国政府制定长期规划，对计算科学进行长期资助，以确保美国的竞争优势和国家安全。该委员会认为，计算科学是 21 世纪最重要的技术领域之一，因为它对整个社会的进步是不可或缺的。

在国内，随着天河一号 A 和曙光星云等国产千万亿次 CPU+GPGPU 异构超级计算机的陆续上线，并分别夺取 TOP500 世界第一和世界第二的好成绩，国产超级计算应用软件的研制水平和性能水平也取得了突飞猛进的进展，体现了计算平台的超前发展对应用的拉动作用。目前，国产超级计算应用能够利用的处理器核心数已经接近 10 万处理器核，但尚未突破 10 万核，且此类应用大部分都属于测试性应用；而国际上实际可有效利用的处理器核数已经突破 40 万处理器核，且出现了不少突破性的模拟成果和科学新发现。但是，不管国内还是国外，到目前为止，还很少有能够利用千万亿次超级计算机的全部处理器和 GPGPU 核的异构并行计算取得突破性应用成果的应用程序出现，CPU+GPGPU 异构并行应用软件开发面临瓶颈，亟需尽快取得突破。如何在不断突破处理器核心数可扩展性的前提下，尽快利用全机的 CPU+GPGPU 异构并行取得实际的突破性模拟应用成果，是当前国际计算科学领域面临的一个难题。

众所周知，开发一个好的并行应用软件，首先要选择合适的并行程序设计语言和环境，这关系到程序的生产效率、运行性能和可移植性等重要方面。在异构并行体系架构出现之前，工业界普遍公认的并行程序设计标准是支持分布式存储并行的 MPI 和支持共享存储并行的 OpenMP。随着 GPGPU 在超级计算环境中的日渐普及，加之它在若干应用领域所表现出的优秀的性能功耗比和性价比优势，越来越多的开发人员开始关注 GPGPU 的并行程序设计问题，迫切希望工业界尽快制定一个统一的为广大厂商和用户普遍接受的 GPGPU 并行编程语言或环境，以降低用户的开发成本和学习成本，确保程序



在不同厂家提供的异构加速部件间的可移植性。OpenCL(开放计算语言)异构计算语言正是在这一背景之下,在吸收了 CAL、CTM、Brook++和 CUDA 等语言和开发环境精华的基础上,由 OpenGL 标准的提出者 Khronos Group 国际标准化组织提出的。

在 OpenCL 出现之前,工业界占据主流的是 NVIDIA 公司为自己生产的 GPGPU 量身定做的 CUDA 语言。由于 NVIDIA 公司投入大量的人力和物力配合其 GPGPU 的销售对 CUDA 进行培训,目前 CUDA 语言已经有了大量的中文教科书可以利用。但是,由于 OpenCL 作为业界公认的异构计算程序设计语言标准刚推出不久,到目前为止的英文教科书资料都很稀少,更别说中文教科书了。为了尽快将这一工业标准的异构计算并行编程语言介绍给广大的科研人员和学生,在清华大学出版社和 AMD 公司的支持下,我们组织翻译了本书,以便让中国读者第一时间掌握国际上关于异构计算并程序设计的最新技术动向和编程标准,确保所编写出的异构并程序的可移植性,推动异构计算在中国特别是高等院校的进一步普及和发展。

本书作者 Benedict Gaster 博士和 Lee Howes 博士都就职于 AMD 公司的异构计算编程模型的研究和开发部门,是工作在第一线的 OpenCL 语言架构师和开发者。Benedict Gaster 博士还是 AMD 公司在 OpenCL 语言标准制定的国际标准化组织 Khronos Group 的代表。本书的另外三位作者 David R. Kaeli 教授、Perhaad Mistry 和 Dana Schaa 都来自美国东北大学。Perhaad Mistry 和 Dana Schaa 是 David R. Kaeli 教授的博士研究生。David R. Kaeli 教授从事高性能计算机系统和软件的性能和设计研究已有多年历史,近期主要从事包括微架构、后端编译器和 GPGPU 等方面的研究工作,指导了 GPU 计算领域方面的多个研究项目。在加盟美国东北大学之前,他曾在 IBM 工作 12 年,期间在 IBM 的 T. J. Watson 研究中心工作 7 年,合作发表了 200 多篇学术论文。他还曾担任 IEEE 计算机系统结构技术委员会主席,并被选举为 IEEE 会士。

在本书翻译的过程中,中国科学院软件研究所并行软件与计算科学实验室的博士研究生李焱和颜深根,中国海洋大学的博士研究生贾海鹏分别参与了书中部分章节的翻译和审校工作,在此一并表示感谢。全书由张云泉、张先轶、龙国平和姚继锋负责统一审校和修订。

由于时间仓促,再加上有些专业术语目前国内没有统一的译法,所以我们结合读者的知识背景和工作环境,对一些术语采用了保留其英文名称的做法,以帮助大家达成共识,促进沟通。如果发现有翻译方面的错误或不妥之处,恳请广大读者不吝指正。

# 英文版序

Avi Mendelson 博士  
微软(以色列)研发中心  
以色列理工学院兼职教授

二十多年以来,戈登·摩尔(Gordon Moore)提出的摩尔定律一直鼓舞和激励着计算机行业的发展,即片上晶体管密度每 18 个月翻一番;因此可以预测特定应用在两年之内在新一代处理器上取得的性能将是上一代处理性能的两倍。新一代处理器上以“黄金因子”(golden factor)缩小所有晶体管尺寸,0.3(理想缩减比率),并相应减少功耗,制造工艺和处理器技术的持续改进是推动这一发展趋势的主要动力。因此,任何新一代处理器上晶体管密度每增加一倍,从而获得 50%的速度提升(频率),并保持功耗和功耗密度相同。当需要更好的性能时,计算机架构师便主要通过使用额外的晶体管来提升频率和增加新的架构特点,从而提升现有应用和新应用的性能。

在 21 世纪中期,晶体管的尺寸变得如此之小,以至于“physics of small devices”开始支配着整个芯片的特性。因此,在没有显著增加功耗和功耗密度的情况下,可能无法实现频率的提高和密度的增加。最近国际半导体科技蓝图(International Technology Roadmap for Semiconductors, ITRS)的一份报告中也支持这一看法,并指出这一趋势在可预见的未来中继续存在,最有可能成为影响技术发展和计算机系统未来的最重要因素。

为达到性能每隔一段时间(不再是两年)翻一倍的期望,业内发生了两个主要的变化。第一,在现代处理器上,增加每个芯片上的核数而不是频率。这种趋势迫使软件也做出相应的改变。既然不能指望硬件为给定的应用程序带来更显著的性能,我们需要利用多核架构的优势重新实现同一应用。第二,散热和功耗成为未来架构设计首要考虑的问题。这种趋势鼓励社会开始寻找异构解决方案:系统由不同的子系统组成,优化每个子系统来获取不同的优化点,以满足不同的工作负载。例如,许多系统由“传统”的 CPU 架构和特殊用途的 FPGA 或图形处理器(GPU)组成。这种集成可以在不同层次上实现,例如,在系统级、主板级以及(最近在)CPU 核级。

在同构并行和分布式系统上开发软件被认为是一个艰巨的任务,即便使用知名的范式以及良好的编程语言,开发方法,算法和调试工具等。开发支持通用异构系统的软件相对较新,所以不太成熟并且困难重重。异构系统正成为大势所趋,许多主要的软件和

硬件厂商开始创建软件环境来支持它们。AMD 提议使用美国斯坦福大学开发的 Brook 语言来处理流计算,后来为了访问其低级硬件原语流而对 SW 环境进行扩展,将 Close to Metal(CTM)和 Compute Abstraction Layer(CAL)包含在内,目的是为了充分利用其高线程并行架构的优势。NVIDIA 采取了类似的方法,近几代 GPU 的设计和相应的 CUDA 编程环境利用了 GPU 上高度并行线程的特点。Intel 通过扩展多核编程的使用方法来在 Larrabee 架构上进行编程。IBM 提出使用基于消息传递的软件以便利用其异构和非一致的 Cell 架构及基于 FPGA 的解决方案来整合 VHDL 和 C/C++语言编写的库,以充分利用这两种环境。这些编程环境为特定领域都提供了应用范围,但在某种程度上,都不满足通用软件运行在不同硬件架构上的需求(如 Java 代码可以运行在差异巨大的 ISA 架构上)。OpenCL 是为满足这一重要需求而设计的。它由非营利性组织 Khronos 定义和管理,该语言和开发环境“借用”了许多非常成功的基本概念和特定的硬件环境,如 CUDA, CAL 和 CTM,并将它们融合在一起,创造了一个独立于硬件的软件开发环境。它支持不同层次的并行性,并能高效映射到由 CPU, GPU, FPGA 和其他潜在的未来设备组成的单一或多设备的同构或异构系统。为了支持未来的设备,OpenCL 定义了一套机制,如果满足机制,设备可以进行无缝地融入 OpenCL 环境。OpenCL 定义的运行时可以进行资源管理,在相同的执行环境下融合不同类型的硬件,并有望在未来使用更通用的方式允许动态平衡计算、功耗和其他资源,如内存层次。

本书是一本教科书,旨在教学生如何在异构环境下进行编程。本书始于一个非常重要的讨论(即如何编写并行系统),定义了学生在异构系统下开始编程之前需要理解的概念。同时,本书通过使用类比来理解并行和分布式系统中使用的不同模型。第 2~4 章循序渐进地为学生介绍了 OpenCL 的基本架构(第 2 章),包括主机和设备架构(第 3 章)。第 4 章提供一个重要实例将这些概念融合在一起。第 5 章和第 6 章对前几章所介绍的概念进行扩展,帮助读者进一步理解 OpenCL 中并发的概念和运行时执行(第 5 章),剖析 CPU 与 GPU(第 6 章)。OpenCL 基础搭建之后,本书有 4 章(第 7 章~第 10 章)内容致力于更复杂的实例,这些内容对学生理解 OpenCL 的广泛应用是至关重要的,这些应用超越了任何领域的具体操作模式。本书还演示了同一程序如何在不同平台上运行,如 NVIDIA 或 AMD。本书最后三章致力于高级主题。

毫无疑问,这是一本非常重要的书,便于学生和研究人员更好地理解通用异构计算领域,尤其是 OpenCL 提供的解决方案。这本书写得很好,适合不同经验水平的学生,可以用来作为 OpenCL 课程的教科书,书中部分内容可以作为其他课程的扩展,例如,前两章非常适合并行编程课程,而有一些例子可作为高级课程的一部分。

# 前言

## 我们身处异构世界

我们所在的世界本质上是异构的。如此多元的丰富性和细节难以用语言描述。与此同时，广泛不同的实体针对具体任务和环境进行的优化带来了某种程度的复杂性和相互影响。

在计算领域，丰富的异构计算系统有利于程序员选择最佳的硬件架构来执行手边的任务，或者选择恰当的任务来优化使用特定的硬件架构。在解决包含各种不同任务的计算问题时，异构系统在这两方面的灵活性表现得尤为明显。最近，计算机设计社区出现了一股构建异构系统的热潮。市面上正在涌现出一些新型的计算系统，它们由不同类别的架构组成。但这一进程的发展受阻，因为缺乏一个标准化的编程环境来统管通用框架内呈多样化态势的各种资源。

## 有关 OpenCL

OpenCL 的发展旨在减轻程序员为异构系统编写应用程序的负担。OpenCL 迎合了当前为给定架构上增加处理器核数这一发展趋势。OpenCL 框架能够运行在多核 CPU、DSP、FPGA、GPU 以及异构加速处理单元上。该架构提供了广泛的方法来抽取内存系统和指令流中的并行性和效率。OpenCL 框架的多样性为设计人员提供了解决问题的方法——如果采用 OpenCL 规范进行设计，该解决方案的更新便能够与硬件架构的种类增多和发展保持同步。OpenCL 标准中抽象和接口允许程序员使应用程序无缝地运行在一个或多个厂商的多种异构设备上。

## 本书的目标

目前为止，没有一本合适的教材能够帮助程序员和软件工程师利用 OpenCL 编程标准所具有的能力和灵活性。因此，我们尝试填补空白。为了这个目标，我们不想编写一本语法手册——有大量好的资源，程序员可以从中找开完整、最新的 OpenCL 语法描述。

本书试图向开发者或学生说明如何利用 OpenCL 框架来构建有趣且有用的应用。我们通过提供大量实际应用案例来展示该编程标准具有的魅力。

我们希望读者能够拥抱这个新的编程框架，并探究它为异构系统带来的所有好处。我们欢迎对本书提出改进意见，同时也希望本书能够帮助你构建下一个异构应用程序。

## 致谢

我们感谢 Manju Hegde 提议本书写作项目，感谢 BaoHuong Phan 和 Todd Green 从 AMD 和 Morgan Kaufmann 对项目的管理和支持。同时感谢 Jay Owen 为本项目参与者之间互相交流所提供的帮助。在技术方面，我们感谢 Jay Cornwall 为编辑本书所做的深入细致的工作，感谢 Joachim Deguara, Takahiro Harada, Justin Hensley, Marc Romankewicz 和 Byunghyun Jang 对各章所做的重要贡献，尤其是大量的案例学习，没有他们的帮助，这些不可能完成。来自诺基亚研究中心的 Jari Nikara, Tomi Aarnio 和坦佩雷理工大学的 Janne Pietiäinen 也为本书做出了贡献。

## 关于作者

**Benedict R. Gaster** 是一位致力于研究下一代异构处理器编程模型的软件架构师，具体研究新一代处理器(同时包含 CPU 和 GPU 加速器)进行并行编程的高层次抽象。他对 OpenCL 的设计做出了广泛的贡献，并在 Khronos Group(科纳斯组织)的开放标准联盟中代表 AMD。他因为类型系统的可扩展标记和变量的研究成果而获得计算机科学博士学位。

**Lee Howes** 已经在 AMD 工作两年多，目前的研究集中于未来异构计算的编程模型。他的研究举在于如何以声明方式表示迭代域到数据的映射，用通俗易懂的方式向开发人员说明复杂的架构概念和优化技术，这两者都通过编程模型的改进和教育来实现。他因为这一领域的研究成果而获得伦敦帝国学院计算机科学的博士学位。

**David Kaeli** 获得罗格斯大学电气工程专业的学士和博士学位，雪城大学计算机工程专业专业的硕士学位。他是东北大学工程学院本科课程的副院长兼欧洲经委会学院的全职教授，东北大学计算机系统结构研究实验室主管(NUCAR 负责人)。在 1993 年加入东北大学之前，他曾在 IBM 工作 12 年。在这 12 年当中，后 7 年在 T. J. Watson 研究中心(位于纽约市 Yorktown Heights)工作。作为合著者发表了 200 多篇经过严格评审的文献。他的研究范围很广，包括微架构到后端编译器和软件工程。他指导了 GPU 计算领域方面的很多研究项目。目前，他担任计算机系统结构 IEEE 技术委员会主席。他是 IEEE 会士和 ACM 成员。

**Perhaad Mistry** 目前在东北大学攻读博士学位。他获得孟买大学电子工程专业学士学位和东北大学计算机工程专业硕士学位。他目前是东北大学计算机系统结构实验室的一员，导师是 David Kaeli。他做过很多并行计算项目。他曾为 GPGPU 平台的物理模拟设计了可扩展数据结构，还针对异构设备实施了医疗重建算法。目前的研究重点在于异构计算分析工具的设计。他正在研究采用诸如 OpenCL 之类的标准是否能够用于构建出合适的工具来简化当下跨大量异构设备运行的并行编程和性能分析。

**Dana Schaa** 拥有加州理工大学圣路易奥比斯波分校计算机工程学士和东北大学电气和计算机工程的硕士学位，他目前也在东北大学攻读博士学位。他的研究兴趣包括并行编程模型和抽象，特别针对 GPU 架构。他已经开发了基于 GPU 实现的几个医疗成像研究项目，从实时可视化到分布式异构环境的图像重建。他在 2010 年娶了一位出色的妻子 Jenny，他们与他们可爱的猫一起生活在波士顿。

## 关于译者

**张云泉博士**，中国科学院软件研究所并行软件与计算科学实验室执行主任，中科院计算机科学国家重点实验室研究员，博士生导师。

1995年北京理工大学计算机科学与技术系计算机应用专业，获工学学士学位；2000年中国科学院软件研究所计算机软件与理论专业硕博连读，获工学博士学位。主要研究方向为大型并行数值软件、并行程序设计和性能评价、并行计算和并行编程模型，等。

已在国内外学术刊物上发表论文一百余篇，出版译著一本，论著章节三章，申请专利两项，软件著作权三项。

获得中科院软件所首批杰出青年人才专项计划支持。曾获国家科技进步奖二等奖一项，获中科院科技进步二等奖一项，获2000年度中科院院长奖学金优秀奖。连读七年(2004—2010)获中国软件行业协会全国先进个人。

兼任中国计算机学会 YOCSEF 学术委员会主席(2010.5—2011.5)，中国软件行业协会常务理事/数学软件分会秘书长，中国计算机学会理事/高性能计算专业委员会秘书长，国家863“高性能计算机评测中心”技术委员会委员，CCF《计算机学会通讯》、《计算机科学》和中国科学院《科研信息化技术及应用》编委。中国高性能计算机TOP100排行榜的主要组织者和发布者。2008年—2011年HPC China程序委员会共同副主席。IEEE IWPAPS08和IWPAPS09程序委员会共同主席，IEEE ICPADS2008、PAAP09、FCST09、FutureTech 2010、ACM ICS 2010、IEEE IPDPS2012、IEEE CCGrid 2012、FutureTech 2012程序委员会委员。IEEE CSE 2010程序委员会共同主席。ISC 2012 Steering Committee成员；SC 2012程序委员会委员。中国软件行业协会常务理事，中国计算机学会理事。

**张先轶**，中国科学院软件研究所并行软件与计算科学实验室助理研究员。2007年硕士毕业于北京理工大学计算机学院，2005年本科毕业于北京理工大学计算机科学与工程系。主要研究兴趣包括，面向GPU等异构系统的并行算法与软件性能优化，自适应性能调优等。

**姚继锋博士**，1998年清华大学应用数学系获学士学位，2004年中国科学院软件研究所获计算机软件与理论专业博士学位。曾担任上海超级计算中心研发部经理，高性能计算技术总监，现担任中科院软件所高级工程师，硕士生导师，并行软件与计算科学实验

室主任助理，中科嘉速(北京)并行软件有限公司总经理。

主要研究方向并行软件与算法、大规模数值计算方法、海量数据处理和可视化技术。先后主持和参与了多个国家和地方重大科研和软件研发项目，其中包括国家 863 重大项目《天体大规模并行数值计算软件平台的研制》(课题副组长)、863 重点项目《新概念高效能计算机体系结构及系统研究开发》(总体组成员，子课题负责人)、上海市信息化专项资金项目《并行程序集成开发环境》、《基于网格技术的企业虚拟计算平台》、《材料模拟高性能集成平台》(均为课题负责人)。

**龙国平博士**，2000 年 9 月到 2004 年 6 月，中南大学计算机科学与技术专业毕业，获工学学士学位；2004 年 9 月到 2010 年 6 月，中国科学院计算技术研究所体系结构方向硕博连读，获工学博士学位。博士论文题目《局部性制导的多核/多线程处理器片上资源共享研究》，获 2011 年 CCF 博士论文提名奖。现为中科院软件研究所助理研究员。2011 年 6 月到 2012 年 7 月，美国加州大学圣巴巴拉分校系统结构实验室访问研究。主要研究方向为计算机体系结构、并行计算以及大规模数据处理，发表论文 14 篇，授权专利 2 项。



# 目 录

<b>第 1 章 并行编程入门</b> .....1	新建一个 OpenCL 程序对象.....26
引言.....1	OpenCL 的 kernel.....27
OpenCL.....1	内存模型.....29
本书目标.....2	编写 kernel.....31
并行思维.....2	向量相加实例的完整代码.....32
并发编程模型和并行编程模型.....6	小结.....39
线程和共享内存.....9	参考文献.....39
消息传递通信.....9	
不同的并行粒度.....10	<b>第 3 章 OpenCL 设备架构</b> .....41
数据共享和同步.....11	引言.....41
本书结构.....11	硬件权衡.....41
参考文献.....12	性能随频率的提升及其限制.....43
扩展阅读和相关网站.....13	超标量执行.....44
	VLIW.....44
<b>第 2 章 OpenCL 简介</b> .....15	SIMD 和向量处理.....47
引言.....15	硬件多线程.....48
OpenCL 标准.....15	多核架构.....51
OpenCL 规范.....15	集成：片上系统和 APU.....53
kernel 和 OpenCL 执行模型.....16	高速缓存层次和内存系统.....54
平台和设备.....19	架构设计空间.....55
主机-设备交互.....19	CPU 设计.....56
执行环境.....21	GPU 体系结构.....60
上下文.....22	APU 和类 APU 的设计.....63
命令队列.....22	小结.....64
事件.....23	参考文献.....65
内存对象.....23	
flush 命令和 finish 命令.....26	<b>第 4 章 OpenCL 基本实例</b> .....67
	引言.....67