HZ BOOKS
华章科技

ELSEVIER
爱思唯尔

CUDA开发者社区技术总监亲自撰写，英伟达中国首批CUDA官方认证工程师翻译，译著双馨

全面、详实地讲解了CUDA并行程序设计的技术知识点和编程方法，包含大量实用代码示例，是目前学习CUDA编程最权威的著作之一

高性能计算系列丛书

CUDA Programming
A Developer's Guide to Parallel Computing with GPUs

# CUDA并行程序设计
## GPU编程指南

（美）Shane Cook 著
苏统华 李东 李松泽 魏通 译 马培军 审校

机械工业出版社
China Machine Press

CUDA Programming
A Developer's Guide to Parallel Computing with GPUs

# CUDA并行程序设计

## GPU编程指南

（美）Shane Cook 著
苏统华 李东 李松泽 魏通 译 马培军 审校



机械工业出版社
China Machine Press

**图书在版编目（CIP）数据**

CUDA 并行程序设计：GPU 编程指南 /（美）库克（Cook, S.）著；苏统华等译；马培军审校 . —北京：机械工业出版社，2014.1
（高性能计算系列丛书）
书名原文：CUDA Programming：A Developer's Guide to Parallel Computing with GPUs

ISBN 978-7-111-44861-7

I. C… II. ①库… ②苏… ③马… III. 图像处理—程序设计 IV. TP391.41

中国版本图书馆 CIP 数据核字（2013）第 276497 号



本书是 CUDA 并行程序设计领域最全面、最详实和最具权威性的著作之一，由 CUDA 开发者社区技术总监亲自撰写，英伟达中国首批 CUDA 官方认证工程师翻译，详实地讲解了 CUDA 并行程序设计的技术知识点（平台、架构、硬件知识、开发工具和热点技术）和编程方法，包含大量实用代码示例，实践性非常强。

全书共分为 12 章。第 1 章从宏观上介绍流处理器演变历史。第 2 章详解 GPU 并行机制，深入理解串行与并行程序，以辩证地求解问题。第 3 章讲解 CUDA 设备及相关的硬件和体系结构，以实现最优 CUDA 程序性能。第 4 章介绍 CUDA 开发环境搭建和可用调试环境。第 5 章介绍与 CUDA 编程紧密相关的核心概念——网格、线程块与线程，并通过示例说明线程模型与性能的关系。第 6 章借助实例详细讲解了不同类型内存的工作机制，并指出实践中容易出现的误区。第 7 章细述多任务的 CPU 和 GPU 协同，并介绍多个 CPU/GPU 编程秘技。第 8 章介绍如何在应用程序中编写和使用多 GPU。第 9 章详述 CUDA 编程性能限制因素、分析 CUDA 代码的工具和技术。第 10 章介绍编程实践中的库与软件开发工具包。第 11 章讲解如何设计基于 GPU 的系统。第 12 章总结 CUDA 应用中易犯错误以及应对建议。

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：肖晓慧
藁城市京瑞印刷有限公司印刷
2014 年 1 月第 1 版第 1 次印刷
186mm × 240mm • 33.5 印张
标准书号：ISBN 978-7-111-44861-7
定 价：99.00 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）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

# 致中国读者

获悉本书将译为中文，传播于华夏大地，我欣喜不已。中国已跃居高性能计算的超级大国行列。中国的经济发展与GPU市场的发展有些相似：它没有被过去的束缚所羁绊，而是开启了一个新世界。中国建造高性能计算机的实力，足以问鼎美国大型计算机公司。中国的高性能计算明显优于其他对手的原因在于大量采用了GPU技术。GPU技术在中国广泛传播，主要得益于英伟达硬件和CUDA语言，而后者正是本书竭诚呈现的主题。

与很多国家一样，中国的计算领域也经历了变革。陈旧的串行编程模型正在被抛弃，计算领域正在拥抱并行体系结构。GPU拥有大规模并行的处理器核心，这是这一变革的重要组成部分。CPU领域的编程模型必然发生变化，但仅是微小的调整，以便利用GPU加速器的潜力。GPU目前应用到了各行各业。不论是学生还是专业人士，通过阅读本书均会收获良多。本书将帮助你在GPU和大规模并行处理器编程方面一路前行。

Shane Cook

# 译者序

我们正在由单核时代进入多核时代和众核时代。在单核时代，软件行业一直享用着“免费的午餐”。受益于CPU主频的指数级提速，开发软件无须任何代码修改，只要换上新一代的处理器，即可获得性能上的飞速提升。随着汹涌而来的众核时代，这里已经“不再有免费午餐”[⊖]。随着计算架构的不断演进，编程模型也发生着深刻的变化。计算机软件行业面临着最大的变迁问题——从串行、单线程的问题求解方式切换到大规模线程同时执行的问题求解方式。而CUDA提供了非常优秀的可扩展架构，以支持这种大规模并行程序设计需求。

本书是一本很出众的CUDA书籍，内容全面而又不落窠臼。全书可以分成四个部分。第一部分为背景篇，包括前4章。其中前两章简述流处理器历史和并行计算基本原理，第3～4章分别介绍了CUDA的硬件架构与计算能力和软件开发配置。第二部分为CUDA基本篇，包括第5～7章。第5、6章依次介绍了CUDA线程抽象模型和内存抽象模型，在此过程中，紧密结合直方图统计实例和样本排序实例进行讨论。为了更好地增进读者的实践经验，第7章全方位剖析了AES加密算法的CUDA实现过程。第三部分为CUDA扩展篇，包括第8～10章。其中第8、9章面向优化执行性能，而第10章为提升开发生产效率。第8章从充分利用多个硬件设备的角度，讲述了流的使用。相反的，第9章从程序优化角度，给出了CUDA性能调优的全方位指导。第10章介绍了一些常用的函数库和CUDA开发包中提供的优质SDK，为大型软件的快速发布提供了支持。第四部分为CUDA经验篇，包括最后的两章。这两章分别针对硬件系统搭建和软件生产过程中的共性问题提供建议，是作者多年CUDA开发经验的总结。

本书特色鲜明。作者在介绍CUDA时，仿佛在跟朋友聊天论道，谈论家常，讲着故事，娓娓道来。论到关键之处，却又语重心长，体贴备至。在不知不觉中，把CUDA的魅力展示得淋漓尽致，同时把CUDA程序设计的功力传授于你。

本书的翻译工作经过精心的组织，整个过程得到大批专业人士的帮助。在交付出版社之前，译者团队经过了全书讨论、初译、初核、再译、再核、审校等六个环节。很荣幸地邀请到在哈尔滨工

⊖ Herb Sutter在2005年发表的论文“The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software”中对这一问题作了前瞻性讨论。——译者注

业大学长年承担“并行程序设计”和“计算机体系结构”等课程教学工作的李东教授，加盟本书的翻译团队。李东教授翻译了第 1 ~ 3 章，对保证本书的翻译质量起到了重要作用。本书第 5 ~ 7 章的初译以及第 9 章部分小节的初译和再译由李松泽负责。本书第 10 章和第 12 章的初译以及第 9 章部分小节的初译和再译由魏通负责。苏统华负责了本书前言、第 4 章、第 8 章、第 9 章部分小节和第 11 章的初译和再译任务，除此之外，他还负责了全书的初核和再核任务。特别感谢哈尔滨工业大学软件学院院长马培军教授，他应邀审校了全部译文，提出了很多中肯的改进意见。本书在交到出版社之后，又得到了机械工业出版社编辑团队的大力帮助，他们的工作专业而细致，让人钦佩。另外还要感谢哈尔滨工业大学软件学院 2012 级数字媒体方向的硕士研究生，参与了部分内容的初译，特别是王烁行同学做了不少工作。如果没有这么多人的辛勤奉献，这本中译本很难如期呈现。另外，国家自然科学基金（资助号：61203260）对本书的翻译提供了部分资助，哈尔滨工业大学创新实验课《CUDA 高性能并行程序设计》也对本书的翻译提供了大力支持。

由于本书涉及面广，很多术语较新，目前尚无固定译法，翻译难度很大。有时，为一个术语选择一个恰当的中文译法，译者经常反复推敲、讨论。但由于译者水平有限，译文中难免存在一些问题，真诚地希望读者朋友们将您的意见发往 cudabook@gmail.com。

苏统华

哈尔滨工业大学软件学院

# 前　言

过去的五年中，计算领域目睹了英伟达（NVIDIA）公司带来的变革。随后的几年，英伟达公司异军突起，逐渐成长为最知名的游戏硬件制造商之一。计算统一设备架构（Compute Unified Device Architecture，CUDA）编程语言的引入，第一次使这些非常强大的图形协处理器为 C 程序员日常所用，以应对日益复杂的计算工作。从嵌入式设备行业到家庭用户，再到超级计算机，所有的一切都因此而改变。

计算机软件界最大的变迁是从串行编程转向并行编程。其中，CUDA 起到了重要的作用。究其本质，图形处理单元（Graphics Processor Unit，GPU）是为高速图形处理而设计的，它具有天然的并行性。CUDA 采用一种简单的数据并行模型，再结合编程模型，从而无须操纵复杂的图形基元。

实际上，CUDA 与之前的架构不同。它不要求程序员对图形或者图形基元有所了解，也不用程序员有任何这方面的知识。你也不一定要成为游戏开发人员。CUDA 语言使得 GPU 看起来与别的可编程设备一样。

本书并不假定读者有 CUDA 或者并行编程的任何经验，仅假定读者有一定的 C/C++ 语言编程知识。随着本书的不断深入，读者将越来越胜任 CUDA 的编程工作。本书包含更高级的主题，帮助你从不知晓并行编程的程序员成长为能够全方位发掘 CUDA 潜力的专家。

对已经熟悉并行编程概念和 CUDA 的程序员来说，本书包含丰富的学习资料。专设章节详细讨论 GPU 的架构，包括最新的费米（Fermi）和开普勒（Kepler）硬件，以及如何将它们的效能发挥到极致。任何可以编写 C 或 C++ 程序的程序员都可以在经过几个小时的简单训练后编写 CUDA 程序。通过对本书的完整学习，你将从仅能得到数倍程序加速的 CUDA 编程新手成长为能得到数十倍程序加速的高手。

本书特别针对 CUDA 学习者而写。在保证程序正确性的前提下，侧重于程序性能的调优。本书将大大扩展你的技能水平和对编写高性能代码的认识，特别是 GPU 方面。

本书是实践者在实际应用程序中使用 CUDA 编程的实用指南。同时我们将提供所需的理论知识和背景介绍。因此，任何人（不管有无基础）都可以使用本书，从中学习如何进行 CUDA 编程。综上，本书是专业人士和 GPU 或并行编程学习者的理想之选。

本书编排如下：

**第 1 章** 从宏观上介绍流处理器（streaming processor）的演变历史，涉及几个重要的发展历程，正是它们把我们带入今天的 GPU 处理世界。

**第 2 章** 介绍并行编程的概念。例如，串行与并行程序的区别，以及如何采用不同的策略寻找求解问题之道。本章意在为既有串行程序员建立一个基本的认识，这里的概念将在后面进一步展开。

**第 3 章** 详尽地讲解 CUDA 设备及与其紧密相关的硬件和架构。为了编写最优性能的 CUDA 程序，适当了解设备硬件的相关知识是必要的。

**第 4 章** 介绍了如何在 Windows、Mac 和 Linux 等不同操作系统上安装和配置 CUDA 软件开发工具包，另外介绍可用于 CUDA 的主要调试环境。

**第 5 章** 介绍 CUDA 线程模型，并通过一些示例来说明线程模型是如何影响程序性能的。

**第 6 章** 我们需要了解不同的内存类型，它们在 CUDA 中的使用方式是影响性能的最大因素。本章借助实例详细讲解了不同类型内存的工作机制，并指出实践中容易出现的误区。

**第 7 章** 主要详述了如何在若干任务中恰当地协同 CPU 和 GPU，并讨论了几个有关 CPU/GPU 编程的议题。

**第 8 章** 介绍如何在应用程序中编写和使用多 GPU。

**第 9 章** 对 CUDA 编程中限制性能的主要因素予以详解，考察可以用来分析 CUDA 代码的工具和技术。

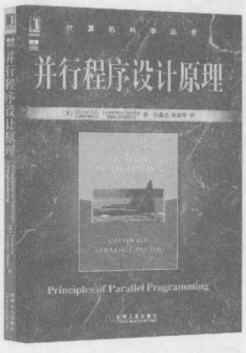
**第 10 章** 介绍了 CUDA 软件开发工具包的示例和 CUDA 提供的库文件，并介绍如何在应用程序中使用它们。

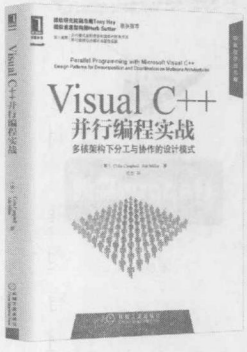
**第 11 章** 关注构建自己的 GPU 服务器或者 GPU 集群时的几个相关议题。

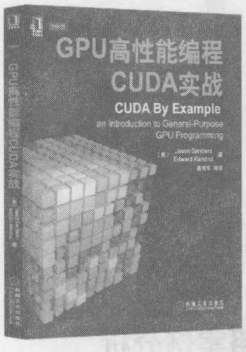
**第 12 章** 检视多数程序员在开发 CUDA 应用程序时易犯的错误类型，并对如何检测和避免这些错误给出了建议。

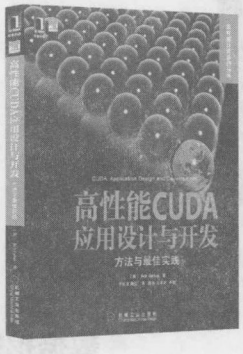
# 推荐阅读


并行程序设计导论
An Introduction to Parallel Programming


并行程序设计原理
Principles of Parallel Programming


Visual C++
并行编程实战
多核架构下分工与协作的设计模式


GPU高性能编程
CUDA实战
CUDA By Example
an Introduction to General-Purpose
GPU Programming


高性能CUDA
应用设计与开发
方法与最佳实践


CUDA并行程序设计
GPU编程指南

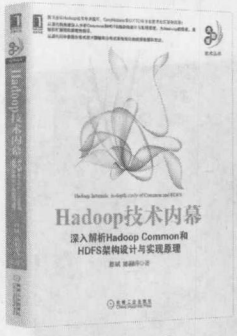

OpenCL编程指南


云计算与分布式系统
从并行处理到物联网
Distributed and Cloud Computing


大规模分布式存储系统
原理解析与架构实战

# 推荐阅读






- **Hadoop实战（第2版）**
  作者：陆嘉恒
  ISBN：978-7-111-39583-6
  定价：79.00元

- **Hadoop技术内幕：深入解析MapReduce架构设计与实现原理**
  作者：董西成
  ISBN：978-7-111-42226-6
  定价：69.00元

- **数据挖掘与数据化运营实战：思路、方法、技巧与应用**
  作者：卢辉
  ISBN：978-7-111-42650-9
  定价：59.00元

- **数据挖掘：实用案例分析**
  作者：张良均
  ISBN：978-7-111-42591-5
  定价：79.00元

- **Hadoop技术内幕：深入解析Hadoop Common和HDFS架构设计与实现原理**
  作者：蔡斌 等
  ISBN：978-7-111-41766-8
  定价：89.00元

- **网站数据分析：数据驱动的网站管理、优化和运营**
  作者：张洪举
  ISBN：978-7-111-43514-3
  定价：69.00元

# 目　录

# 第1章 超级计算简史

## 1.1 简介

为什么我们会在一本关于CUDA的书籍中谈论超级计算机呢？超级计算机通常走在技术发展的最前沿。我们在这里看到的技术，在未来的5 ~ 10年内，将是桌面计算机中很普通的技术。2010年，在德国汉堡举行的一年一度的国际超级计算机大会上宣布，根据500强名单（http://www.top500.org），英伟达基于GPU的机器在世界最强大的计算机列表中位列第二。从理论上讲，它的峰值性能比强大的IBM Roadrunner和当时的第一名Cray Jaguar的性能还要高。当时Cray Jaguar的性能峰值接近3千万亿次。2011年，采用CUDA技术的英伟达GPU仍然是世界上最快的超级计算机。这时大家突然清楚地认识到，与简陋的桌面PC一起，GPU已经在高性能计算领域达到了很高的地位。

超级计算是我们在现代处理器中看到的许多技术的发展动力。由于对用更快的处理器来处理更大数据集的需求，工业界不断生产出更快的计算机。正是在这些发展变化中，GPU的CUDA技术走到了今天。

超级计算机和桌面计算正在向着异构计算发展——人们试图通过将中央处理器（Central Processor Unit，CPU）和图形处理器（Graphics Processor Unit，GPU）技术混合在一起来实现更高的性能。使用GPU的两个最大的国际项目是BOINC和Folding @ Home，它们都是分布式计算的项目。这两个项目使得普通人也能为具体的科学项目做出真正的贡献。在项目中，采用GPU加速器的CPU/GPU主机的贡献远远超过了仅装备CPU主机的贡献。截至2011年11月，大约550万台主机提供了约5.3千万亿次的计算性能，这将近是2011年世界上最快的超级计算机（日本富士通的“京（K）计算机”）计算性能的一半。

作为美国最快的超级计算机Jaguar的升级换代产品，命名为Titan的超级计算机计划于2013年问世。它将用近30万个CPU核和高达18 000个GPU板卡达到每秒10 ~ 20千万亿次的性能。正是由于有像Titan这样的来自世界各地的大力支持，无论是在HPC（高性能计算）行业，还是在桌面电脑领域，GPU编程已经成为主流。

现在，你可以自己“攒”或者购买一台具有数万亿次运算性能的桌面超级计算机了。在21世纪初期，这将会使你跻身500强的首位，击败拥有9632奔腾处理器的IBM ASCI Red。

这不仅部分地展现了过去十几年计算机技术取得的巨大进步，更向我们提出了从现在开始的未来十几年，计算机技术将发展到何种水平这个问题。你可以完全相信在未来一段时间内，GPU将位于技术发展的前沿。因此，掌握GPU编程将是任何一个优秀开发人员必备的重要技能。

## 1.2 冯·诺依曼计算机架构

几乎所有处理器都以冯·诺伊曼提出的处理结构为工作基础，冯·诺伊曼被认为是计算之父之一。在该结构中，处理器从存储器中取出指令、解码，然后执行该指令。

现代处理器的运行速度通常高达4GHz。现代DDR-3内存，与标准的英特尔I7设备配合使用时，可以在运行任何程序时最高达到2GHz的速度。然而，在一个I7设备中至少具有四个处理器或内核。如果你认为超线程能力只能作为一个真正的处理器，那么一个I7设备中也是两个处理器。

在一个I7 Nehalem系统中，三通道DDR-3内存具有的理论带宽如表1-1所示。受主板和确切的内存模式影响，实际带宽可能要小很多。

表1-1 I7 Nehalem处理器带宽

| QPI时钟频率 | 理论带宽 | 单核带宽 |
|---|---|---|
| 4.8GT/s（标准配置） | 19.2GB/s | 4.8GB/s |
| 6.4GT/s（最大配置） | 25.6GB/s | 6.4GB/s |

注意：QPI指快速通道互联（Quick Path Interconnect）

当考虑处理器的时钟速度时，你遇到的第一个问题是关于内存带宽的。用速度为4GHz的处理器，你可能每个时钟周期需要取来一条指令（操作码）和某个数据（操作数）。

通常，每个指令长32位。所以，假设你在每个核上只执行一组不带数据的、顺序执行的指令，则每秒钟你需要取来4.8GB/s ÷ 4=1.2GB条指令。这是假设处理器平均每个时钟周期执行一条指令的情况[㊀]。不过，通常你还需要读取和写回数据，这里假设数据与指令的比例是1∶1，那么这意味着我们的实际吞吐量将减少一半。

内存速度和时钟速度的比率是限制CPU和GPU吞吐量的一个重要因素，这一点我们将在后续的章节中讨论。深入分析你就会发现，除了CPU和GPU中的一些例外，大多数应用程序属于“内存受限型”[㊁]，而不是“处理器时钟周期或负载受限型”。

CPU厂商试图通过使用缓存和突发内存访问来解决这个问题，这利用了程序的局部性原理。下面是一个典型的C程序，请仔细观察该函数中相关操作的类型：

```
void some_function
{
 int array[100];
 int i = 0;
```

㊀ 实际达到的执行速度可能大于或小于1，这里我们做了简化处理。

㊁ 即性能提高受到内存速度的限制。——译者注