

<<OpenCL编程指南>>

图书基本信息

书名：<<OpenCL编程指南>>

13位ISBN编号：9787111398493

10位ISBN编号：7111398491

出版时间：2012-12-1

出版时间：机械工业出版社华章公司

作者：Aaftab Munshi,Benedict R. Gaster,Timothy G. Mattson,James Fung,Dan Ginsburg

页数：404

译者：苏金国,李璜,杨健康

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<OpenCL编程指南>>

内容概要

OpenCL领域公认的权威著作，由OpenCL核心设计人员亲自执笔，不仅全面而深刻地解读了OpenCL规范和编程模型，而且通过大量案例和代码演示了基于OpenCL编写并程序序和实现各种并行算法的原理、方法、流程和最佳实践，以及如何对OpenCL进行性能优化，如何对硬件进行探测和调整。

蒙施、Benedict R.Gaster、Timothy

G.Mattson、James Fung、Dan Giburg所著的《OpenCL编程指南》分为两大部分：第一部分（1~13章），从介绍OpenCL的核心思想和编写OpenCL程序的基础知识开始，对枯燥的OpenCL规范进行了深刻而系统的解读，旨在帮助读者全面、正确地理解OpenCL规范及其编程模型；第二部分（14~22章），提供了一系列经典的案例，如图像直方图、Sobel边界检测过滤器、并行实现Dijkstra单源最短路径图算法、Bullet Physics SDK中的布模拟、用快速傅里叶变换模拟海洋、光流、OpenCL与PyOpenCL结合使用，使用OpenCL完成矩阵相乘与稀疏矩阵向量乘法等，目的是让读者通过案例熟练掌握编写复杂并程序序的方法和技巧

。《OpenCL编程指南》的附录收录了OpenCL规范定义的大量函数、命名常量和类型，可供程序员开发时查阅。

<<OpenCL编程指南>>

作者简介

作者：（美国）蒙施（Aaftab Munshi）译者：苏金国 Aaftab Munshi，OpenCL核心设计小组的灵魂人物之一，同时他也是OpenCL、OpenGL ES 1.1、OpenGL ES 2.0规范的编辑，与DanGinsburg和Dave Shreiner合著了《OpenGL ES 2.0 Programming Guide》。目前他任职于Apple公司。

<<OpenCL编程指南>>

书籍目录

译者序

序

前言

致谢

第一部分 OpenCL 1.1语言与API

第1章 OpenCL介绍

1.1 什么是OpenCL, 或者为什么需要这本书

1.2 多核的未来: 异构平台

1.3 多核世界中的软件

1.4 OpenCL的概念基础

1.4.1 平台模型

1.4.2 执行模型

1.4.3 内存模型

1.4.4 编程模型

1.5 OpenCL与图形

1.6 OpenCL的内容

1.6.1 平台API

1.6.2 运行时API

1.6.3 内核编程语言

1.6.4 OpenCL小结

1.7 嵌入式简档

1.8 学习OpenCL

第2章 HelloWorld: 一个OpenCL例子

第3章 平台、上下文和设备

第4章 OpenCL C编程

第5章 OpenCL C内置函数

第6章 程序与内核

第7章 缓冲区和子缓冲区

第8章 图像和采样器

第9章 事件

第10章 与OpenCL的互操作

第11章 与Direct3D的互操作

第12章 C++包装器API

第13章 OpenCL嵌入式简档

第二部分 OpenCL 1.1案例研究

第14章 图像直方图

第15章 Sobel边界检测过滤器

第16章 并行实现Dijkstra单源最短路径图算法

第17章 Bullet Physics SDK中的布模拟

第18章 用快速傅里叶变换模拟海洋

第19章 光流

第20章 用PyOpenCL使用OpenCL

第21章 使用OpenCL完成矩阵乘法

第22章 稀疏矩阵向量乘法

附录A OpenCL 1.1小结

<<OpenCL编程指南>>

章节摘录

版权页：插图：再次说明，OpenCL并没有在宿主机上规定内存一致性模型。

先从离宿主机最远的内存（私有内存区域）开始考虑，逐步转向宿主机。

私有内存对宿主机是不可见的。

它只对相应的工作项可见。

这个内存采用顺序编程中很熟悉的加载 / 存储内存模型。

换句话说，对私有内存的加载和存储不能重新排序，即除了程序文本中定义的顺序外，不能以其他顺序出现。

对于局部内存，可以保证一个工作组中一组工作项能够看到的值在工作组同步点是一致的。

例如，一个工作组栅栏（work—group barrier）要求在栅栏之前定义的所有加载和存储必须先完成，工作组中这个栅栏之后的工作项才能继续。

换句话说，栅栏标记了一组工作项执行中的某一点，在这一点可以保证内存是一致的，在继续执行之前处于已知状态。

由于局部内存只是在一个工作组内共享，所以这对于定义局部内存区域的内存一致性就足够了。

对于一个工作组中的工作项，在工作组栅栏，全局内存也是一致的。

不过，尽管这个内存在工作组间共享，但无法强制执行一个内核的不同工作组之间全局内存的一致性。

对于内存对象，OpenCL定义了一个宽松一致性模型。

换句话说，单个工作项在内存中看到的值不能保证任何时刻在整个工作项集中都保持一致。

在给定的时刻，对于不同的工作项，对OpenCL内存对象的加载和存储可能以不同的顺序出现。

这称为宽松一致性（relaxed consistency）模型，因为它没有我们期望的加载 / 存储模型那么严格，即并发执行要与串行执行的顺序完全匹配。

最后一步是定义相对于命令队列中命令的内存对象一致性。

在这种情况下，我们使用了释放一致性的一个修改版本。

与一个内核相关联的所有工作项完成时，这个内核释放的内存对象的相应加载和存储完成后，内核命令才能标志为完成。

对于有序队列，这足以定义内核间的内存一致性。

对于乱序队列，则有两个选择（称为同步点）。

第一个选择是在特定的同步点（如命令队列栅栏）强制一致性。

第二个选择是通过事件机制（将在后面介绍）显式地管理一致性。

这些选择同样用于强制宿主机和OpenCL设备之间的一致性，也就是说，内存仅在命令队列中的同步点是一致的。

1.4.4编程模型 OpenCL执行模型定义了一个OpenCL应用如何映射到处理单元、内存区域和宿主机。

这是一个“以硬件为中心”的模型。

现在我们换个角度，介绍如何使用编程模型将并行算法映射到OpenCL。

编程模型实际上就是程序员如何考虑他们的算法。

因此，这些模型本质上比执行模型更为灵活。

OpenCL定义了两种不同的编程模型：任务并行和数据并行。

可以看到，还可以考虑一种混合模型：包含数据并行的任务。

程序员很有创造性，可以预期，过一段时间可能还会创建映射到OpenCL基本执行模型的另外的编程模型。

数据并行编程模型 前面内容描述了数据并行编程模型的基本思想（见图1—4）。

适合采用数据并行编程模型的问题都与数据结构有关，这些数据结构的元素可以并发更新。

基本上，就是将一个逻辑指令序列并发地应用到数据结构的元素上。

并行算法的结构被设计为一个序列，即对问题领域中数据结构并发更新的序列。

<<OpenCL编程指南>>

编辑推荐

《OpenCL编程指南》为OpenCL世界带来了福音。

《OpenCL编程指南》出自资深程序员之手，正是程序员需要的实用指南。

<<OpenCL编程指南>>

名人推荐

“欢迎来到异构并行编程的新世界，凭借这本通俗易懂的权威指南，你将深入了解完整的OpenCL编程模型。

”——Pat Hanrahan，斯坦福大学教授

<<OpenCL编程指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>