

让普通程序员通晓GPU编程，让高性能计算不再高不可攀

GPU Programming and Code Optimization  
High Performance Computing for the Masses

# GPU

# 编程 与 优化

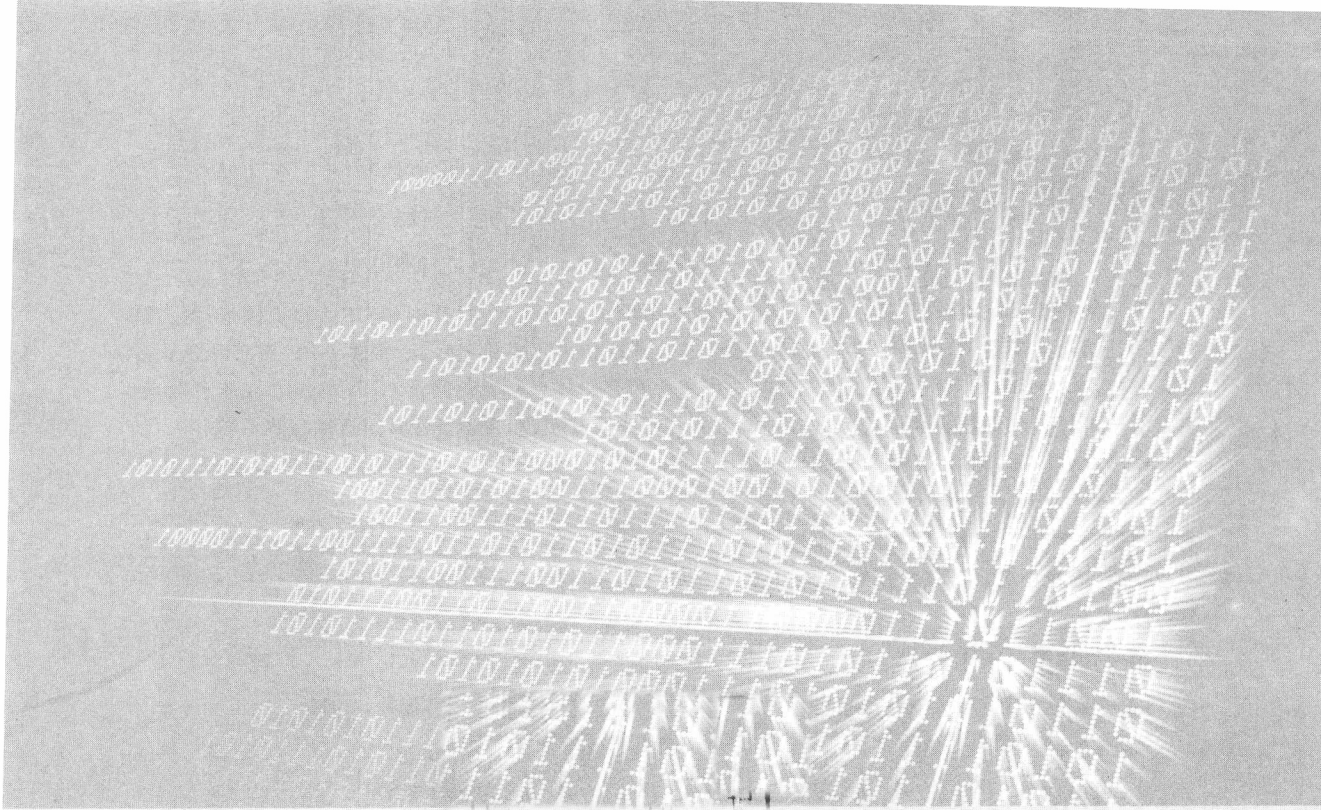
## ——大众高性能计算

方民权 张卫民  
方建滨 周海芳 著  
高 畅

- ◎ 系统全面的GPU知识体系
- ◎ 通俗翔实的异构协同并行
- ◎ 实践引导的有效优化方法
- ◎ 循序渐进的优化实例解析
- ◎ 切中要害的性能影响因素
- ◎ 精炼真实的GPU性能测评

清华大学出版社





GPU Programming and Code Optimization  
High Performance Computing for the Masses

GPU

编程  
与  
优化

——大众高性能计算

方民权 张卫民 著  
方建滨 周海芳  
高畅

清华大学出版社  
北京



## 内 容 简 介

本书第一篇系统地介绍 GPU 编程的相关知识,帮助读者快速入门,并构建 GPU 知识体系;第二篇和第三篇给出大量实例,对每个实例进行循序渐进的并行和优化实践,为读者提供 GPU 编程和优化参考;第四篇总结影响 GPU 性能的关键要素(执行核心和存储体系),量化测评 GPU 的核心性能参数,给出 CPU/GPU 异构系统上覆盖完全的各种混合并行模式及其实践,帮助读者真正透彻理解 GPU。

本书适合作为计算机及相关专业的教材,也可作为 GPU 程序开发人员和科研人员的参考书。

本书封面贴有清华大学出版社防伪标签,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

### 图书在版编目(CIP)数据

GPU 编程与优化:大众高性能计算/方民权等著. --北京:清华大学出版社,2016

ISBN 978-7-302-44642-2

I. ①G… II. ①方… III. ①图像处理—程序设计 IV. ①TP391.41

中国版本图书馆 CIP 数据核字(2016)第 179437 号

责任编辑:白立军

封面设计:杨玉兰

责任校对:李建庄

责任印制:刘海龙

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>, 010-62795954

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:27.25 字 数:660千字

版 次:2016年9月第1版 印 次:2016年9月第1次印刷

印 数:1~2000

定 价:59.00元

---

产品编号:070097-01

---

---

# 前 言

---

多核与众核异构平台因其超强的浮点运算能力而成为当前高性能计算领域的新贵。2010年以来,已有3台异构超级计算机夺魁TOP 500,分别是搭载CPU/GPU异构系统的天河1A和泰坦超级计算机、搭载CPU/MIC异构系统的天河2号超级计算机。在这两类主流的多核与众核异构平台中,CPU/GPU异构平台在性价比、能耗比等方面表现尤为突出,例如,在Green500前10中有9台采用了这种架构。对于高性能计算用户而言,CPU/GPU异构系统无疑是一个良好的选择。

另一方面,当前PC已普遍装备GPU(独立显卡),使得这种CPU/GPU异构系统的硬件平台随处可见。尽管这类平台的GPU工作重心是游戏娱乐而非浮点计算,但在一些精度要求不高的领域仍然优势显著。此外,基于GPU编程的工具包是免费的,因此可用较低的成本构建合适的CPU/GPU异构并行平台。由于显卡的普及,CPU/GPU异构并行程序也能在几乎所有的PC中广泛应用。

然而,仅有硬件是没有应用价值的,异构系统上的程序开发是实现异构系统价值的直接且唯一的步骤。但是,异构并行软件开发面临着巨大挑战,主要包括异构数据通信、基于GPU体系结构的编程与优化、多编译器的联合编译等,具体到实践则难度更大。编写本书的目的就是辅助用户解决这些GPU异构并行软件开发的难题。

目前市面上已有很多GPU编程书籍,其中一些已经论述相当全面,为什么还要撰写本书呢?作为一名有多年开发经验的GPU程序员,阅读这些书籍总感觉有些不足。

首先,对于刚接触GPU的开发者,由于所要认识的GPU体系结构与常用的CPU体系结构差异巨大,相关的理论知识较难理解,而已有的一些论著为了增强理论性和学术性还对相关理论知识进行了抽象提升,因而不够通俗易懂,即使是GPU编程老手也未必能完全读懂。而本书将从GPU程序员的角度出发,通俗易懂地阐述GPU编程与优化相关的理论知识。

其次,从GPU理论到编程优化实践的过渡是非常关键的,但目前市面上的书籍重理论而轻实践。仅有理论而缺乏实践和直观实践效果,难以对程序员读者产生直接价值。况且很多没有实践论证的理论知识未必正确。本书将紧密结合理论和实践,并试图从实践中总结理论知识,从而帮助读者更好地理解。

此外,程序性能优化是GPU编程的重中之重。当前GPU书籍中提及了大量关于优化方法的理论知识,但很少针对每种优化方法和策略给出应用实例(即使有也可能只有一个,借鉴范围不够广),更没有针对某一个实例进行系统性循序渐进的优化。事实上,对于新手而言,优化时最大的困惑就是知道优化方法却不知道用到哪儿、怎么用。本书试图针对大量经典实例进行循序渐进式的优化,为读者提供详尽的优化参考。



最后, GPU 编程实践时不仅仅只是编程, 还涉及编译器、运行环境等相关配套知识, 若是没有这些配套知识, 即使看懂了、写出了相关代码又有什么意义呢? 这无异于纸上谈兵。本书将涉及代码编写、编译, 运行时需要涉及的所有配套知识, 包括系统环境、Linux 命令、编译选项、性能分析、并行计算相关常识等。

综上所述, 本书的定位是帮助 GPU 程序员从“零知识”入门到精通的书籍, 书中内容包含通俗易懂的 GPU 理论知识, 配套的知识体系, 大量代码实例及其循序渐进的优化过程、详细的性能分析和知识点总结, 与性能直接相关的 GPU 核心特征获取、分析和论述。对于 GPU 程序的开发人员, 本书具有较为全面的参考价值。

## 本书的结构和阅读建议

本书共计四篇: 第一篇共 5 章, 主要是 GPU 的理论知识, 包括 GPU 的领域背景(高性能计算概述)、GPU 概述、GPU 硬件架构、GPU 软件体系和 CUDA C 编程; 第二篇共 4 章, 基于 4 个入门级的 GPU 实例展示了其详细的并行和优化过程, 分别是向量加法、向量内积、矩阵乘法和矩阵转置; 第三篇共 5 章, 分别描述了 5 类不同应用的 GPU 编程和优化过程, 包括卷积、曼德博罗特集、前缀求和、排序和简单图像处理; 第四篇共 4 章, 阐述了影响 GPU 程序性能的核心因素, 分别从 GPU 执行核心、GPU 存储体系、影响 GPU 性能的关键因素、CPUs 和 GPU 的协同运算 4 个角度展开探讨。

首遍阅读时, 建议按行文顺序阅读, 本书已按知识难易程度做了梳理; 接着动手实践入门篇和提高篇; 若要进一步深入阅读则需结合核心篇章的 GPU 核心知识与入门篇和提高篇的实践, 逐步理解提升; 最终在本书基础上进一步优化各类应用, 进而开发出自己的优化方法。

对于急于从实例运行入手的读者, 可根据 4.3 节内容安装环境, 然后跳读入门篇和提高篇的实例章节, 在获得一定成功经验后再返回阅读理论篇和核心篇。

## 本书的特点和优势

本书内容丰富, 涵盖了系统全面的 GPU 知识体系、循序渐进的实例优化、从实践导出的真实有效的优化方法、影响 GPU 性能的核心因素、GPU 性能测评和 CPU/GPU 异构协同优化等内容。本书语言朴实, 通俗易懂, 对不易理解的概念定义, 通过笔者的理解重新进行阐述。本书还提供简单易读的实例代码, 详尽的编译命令和清晰的运行结果数据。

GPU 发展迅猛, GPU 架构几乎每两年就更新换代(就在本书第 3 轮修订期间, Pascal 架构发布, 笔者又重新修订了相应章节), 目前市面上的书籍暂时没有提供完整的 GPU 架构知识, 本书总结了所有的 GPU 架构, 阐述了更加系统完善的 GPU 知识体系。

GPU 优化是关键, 市面上同类书籍中阐述了很多优化方法, 但经历了循序渐进优化的具体实例相对缺乏, 甚至有些优化方法存在问题(本书有相应的实验佐证)。

## 致谢

本书大纲由方民权和方建滨共同商定; 方民权完成本书代码的编写、实验数据测试与

分析、本书初稿、首轮修订、重要修订(如错误修正、结构调整和章节内容增加等)以及后续修订内容的权衡和更新;张卫民、方建滨、周海芳等完成后续数轮修订,正是有他们的修订才使本书真正可读;高畅提供了 Linux 图形界面 CUDA 安装的文档和验证,验证了本书所有代码的结果正确性和性能准确性,以及进行一些关键的查漏补缺。正是所有作者共同努力,本书才能真正成稿。

本书支撑课题包括国家自然科学基金项目(41375113 和 61272146)、湖南省研究生创新资助项目(CX2015B030)、国防科学技术大学计算机学院联合博导组项目(面向异构平台的海洋预报软件可移植性技术研究)。

文中涉及许多笔者的个人学术观点和经验总结,由于笔者水平有限,所有成果仅供参考,如有不准确甚至错误之处,望读者谅解并批评指正。另因编写时间仓促,尽管已进行 6 轮修订,但书中纰漏和瑕疵在所难免,若发现笔误或有其他意见建议,请致信 [admin@hpc6.com](mailto:admin@hpc6.com),万分感谢!

编者

2016 年 5 月

---

---

## 笔者的话

---

2016年1月20日是一个值得纪念的日子,这本书终于完成初稿,尽管有的章节还有待大幅修改。自2015年3月有了初步构思,我就义无反顾地开始撰写本书,到最终完稿,期间也多次调整论述结构。作为一名在校博士研究生,顶着博士毕业的压力,写一本对博士毕业“无用”的书,确实挺不容易的,不过我认为是值得的。

本科,我学的是机械专业,硕士转学计算机,2012年接触GPU编程,当时MIC已然兴起,却还能感受到GPU的“余热”。说实话,刚接触GPU时整个人是懵的,师兄已经毕业,师门就我一人研究GPU,当时只有两本GPU书——《GPU高性能运算之CUDA》和*CUDA by example*供参考,书中的知识还有些“过时”(Tesla架构和Fermi架构的区别)、“不系统”、甚至“错误”(见13.3.2节)(当然不可否认这两本书对我的帮助),网上资料就更少了。也许是我个人水平有限,仅Windows的GPU开发环境安装就耗费了半个多月时间,Linux的GPU环境更是失败了不知道多少次(当时网上资料很少,且很多方法尝试后均不可行);首个GPU程序开发完全不知道如何下手,头大了近20天才在某天灵光一闪想通,对GPU优化更是无从着手。其实上述过程只需要一位“老师”带着成功一遍,就能少走很多弯路。说这么多,其实想表达的意思是,新手时的我非常期盼一本系统的、全面的、循序渐进的GPU书籍,这也是我写本书的动因。

2015年,在学术圈GPU早已“过时”,为什么还致力于撰写本书呢?首先,GPU确实稳定好用:2014年也曾研究过MIC,“不可捉摸”的ICC编译器耗光了我的耐心,往往逻辑正确的代码编译器就是报错,明明简单向量化的代码编译器就是认为不可向量化;而CUDA代码只要编写正确,均能正常编译和执行,且性能取决于CUDA代码本身。其次,我认为GPU并行是一个大趋势大市场:GPU有3个层次可扩展性(见3.1节),能满足各种市场需求;GPU(显卡)已在PC普遍装备,性价比高,GPU平台已相当普及;GPU的性能不易受其他程序影响,相比之下CPU程序性能受其他程序影响程度较大;GPU发展迅猛,在未来量子计算机或生物计算机出现前(出现后也不可能立刻取代现有的计算机),CPU/GPU异构系统将会长期存在。

本书的定位是辅助“零知识”GPU程序开发人员从入门到精通的书籍,试图做到“知行合一”(最完美的想法是所有的理论知识均有实验结果佐证,但显然还有差距)、系统性强、知识全面、优化循序渐进。希望本书能够帮助每一位GPU程序开发人员。

方民权



---

---

# 目 录

---

## 第一篇 理 论 篇

<b>第 1 章 高性能计算概述</b> .....	3
1.1 高性能计算概念辨析 .....	3
1.1.1 并行计算、高性能计算和超级计算 .....	3
1.1.2 超级计算机与超级计算中心.....	4
1.2 计算科学 .....	5
1.3 高性能计算发展史 .....	5
1.4 高性能计算简介 .....	6
1.5 向量机与阵列机 .....	8
1.6 本章小结 .....	9
<b>第 2 章 GPU 概述</b> .....	10
2.1 GPU 是什么 .....	10
2.2 协处理器.....	10
2.3 GPU 与显卡的关系 .....	11
2.4 GPU/显卡购买注意事项 .....	11
2.5 为什么要学 GPU 编程 .....	12
2.6 GPU 与 CPU 辨析.....	13
2.7 GPU 发展简史 .....	14
2.8 GPU 编程方法 .....	14
2.9 CPU/GPU 异构系统.....	16
<b>第 3 章 GPU 硬件架构</b> .....	17
3.1 GPU 架构 .....	17
3.1.1 Tesla 架构 .....	18
3.1.2 Fermi 架构 .....	20
3.1.3 Kepler 架构 .....	21
3.1.4 Maxwell 架构 .....	23
3.1.5 Pascal 架构 .....	24

3.2	Kernel 的硬件映射 .....	28
3.3	GPU 存储体系 .....	29
3.4	GPU 计算能力 .....	30
<b>第 4 章</b>	<b>GPU 软件体系 .....</b>	<b>33</b>
4.1	GPU 软件生态系统 .....	33
4.2	CUDA Toolkit .....	34
4.2.1	NVCC 编译器 .....	34
4.2.2	cuobjdump .....	35
4.3	CUDA 环境安装 .....	36
4.3.1	Windows 7 安装 CUDA 4.2 .....	36
4.3.2	Linux 下安装 CUDA .....	38
<b>第 5 章</b>	<b>CUDA C 编程 .....</b>	<b>41</b>
5.1	CUDA 编程模型 .....	41
5.2	CUDA 编程七步曲 .....	42
5.3	驱动 API 与运行时 API .....	42
5.4	CUDA 运行时函数 .....	43
5.4.1	设备管理函数 .....	43
5.4.2	存储管理函数 .....	45
5.4.3	数据传输函数 .....	48
5.4.4	线程管理函数 .....	51
5.4.5	流管理函数 .....	52
5.4.6	事件管理函数 .....	52
5.4.7	纹理管理函数 .....	53
5.4.8	执行控制函数 .....	55
5.4.9	错误处理函数 .....	55
5.4.10	图形学互操作函数 .....	57
5.4.11	OpenGL 互操作函数 .....	58
5.4.12	Direct3D 互操作函数 .....	59
5.5	CUDA C 语言扩展 .....	60
5.6	grid-block-thread 三维模型 .....	61

## 第二篇 入门篇

<b>第 6 章</b>	<b>向量加法 .....</b>	<b>67</b>
6.1	向量加法及其串行代码 .....	67
6.2	单 block 单 thread 向量加 .....	68

6.3	单 block 多 thread 向量加 .....	68
6.4	多 block 多 thread 向量加 .....	69
6.5	CUBLAS 库向量加法 .....	70
6.6	实验结果分析与结论 .....	71
6.6.1	本书实验平台 .....	71
6.6.2	实验结果 .....	71
6.6.3	结论 .....	71
6.7	知识点总结 .....	72
6.8	扩展练习 .....	75
<b>第 7 章</b>	<b>归约：向量内积 .....</b>	<b>76</b>
7.1	向量内积及其串行代码 .....	76
7.2	单 block 分散归约向量内积 .....	77
7.3	单 block 低线程归约向量内积 .....	78
7.4	多 block 向量内积(CPU 二次归约) .....	79
7.5	多 block 向量内积(GPU 二次归约) .....	81
7.6	基于原子操作的多 block 向量内积 .....	81
7.7	计数法实现多 block 向量内积 .....	84
7.8	CUBLAS 库向量内积 .....	85
7.9	实验结果与结论 .....	86
7.9.1	实验结果 .....	86
7.9.2	结论 .....	86
7.10	归约的深入优化探讨 .....	87
7.10.1	block 数量和 thread 数量对归约性能的影响 .....	87
7.10.2	算术运算优化 .....	88
7.10.3	减少同步开销 .....	89
7.10.4	循环展开 .....	90
7.10.5	总结 .....	91
7.11	知识点总结 .....	91
7.12	扩展练习 .....	94
<b>第 8 章</b>	<b>矩阵乘法 .....</b>	<b>95</b>
8.1	矩阵乘法及其 3 种串行代码 .....	95
8.1.1	一般矩阵乘法 .....	95
8.1.2	循环交换矩阵乘法 .....	97
8.1.3	转置矩阵乘法 .....	98
8.1.4	实验结果与最优串行矩阵乘 .....	99
8.2	grid 线程循环矩阵乘法 .....	100



8.3	block 线程循环矩阵乘法 .....	101
8.4	行共享存储矩阵乘法 .....	101
8.5	棋盘阵列矩阵乘法 .....	103
8.6	判断移除 .....	105
8.7	CUBLAS 矩阵乘法 .....	106
8.8	实验结果分析与结论 .....	108
8.8.1	矩阵乘精度分析 .....	108
8.8.2	实验结果分析 .....	110
8.8.3	浮点运算能力分析 .....	111
8.9	行共享存储矩阵乘法改进 .....	111
8.10	知识点总结 .....	113
8.11	扩展练习 .....	115
<b>第 9 章</b>	<b>矩阵转置 .....</b>	<b>116</b>
9.1	矩阵转置及其串行代码 .....	116
9.2	1D 矩阵转置 .....	117
9.3	2D 矩阵转置 .....	118
9.4	共享存储 2D 矩阵转置 .....	119
9.5	共享存储 2D 矩阵转置 diagonal 优化 .....	120
9.6	实验结果分析与结论 .....	121
9.7	共享存储 2D 矩阵转置的深入优化 .....	122
9.8	知识点总结 .....	124
9.9	扩展练习 .....	125
<b>第三篇 提高篇</b>		
<b>第 10 章</b>	<b>卷积 .....</b>	<b>129</b>
10.1	卷积及其串行实现 .....	129
10.1.1	一维卷积 .....	129
10.1.2	二维卷积 .....	131
10.2	GPU 上 1D 卷积 .....	134
10.3	$M$ 常量 1D 卷积 .....	135
10.4	$M$ 共享 1D 卷积 .....	136
10.5	$N$ 共享 1D 卷积 .....	137
10.6	实验结果分析 .....	139
10.6.1	扩展法 1D 卷积实验结果分析 .....	139
10.6.2	判断法与扩展法 1D 卷积对比 .....	140
10.6.3	加速比分析 .....	141

10.6.4	线程维度对性能的影响	141
10.7	2D 卷积的 GPU 移植与优化	142
10.7.1	GPU 上 2D 卷积	142
10.7.2	M 常量 2D 卷积	143
10.7.3	M 常量 N 共享 2D 卷积	143
10.7.4	2D 卷积实验结果分析	145
10.8	知识点总结	145
10.9	扩展练习	147
<b>第 11 章</b>	<b>曼德博罗特集</b>	<b>148</b>
11.1	曼德博罗特集及其串行实现	148
11.2	曼德博罗特集的 GPU 映射	150
11.3	一些优化尝试及效果	152
11.3.1	访存连续	152
11.3.2	uchar4 访存合并	153
11.3.3	4 种零拷贝	153
11.3.4	总结分析	155
11.4	计算通信重叠优化	156
11.5	突破 kernel 执行时间限制	159
11.6	知识点总结	160
11.7	扩展练习	162
<b>第 12 章</b>	<b>扫描：前缀求和</b>	<b>163</b>
12.1	前缀求和及其串行代码	163
12.2	Kogge-Stone 并行前缀和	164
12.2.1	直接 Kogge-Stone 分段前缀和	164
12.2.2	交错 Kogge-Stone 分段前缀和	165
12.2.3	完整 Kogge-Stone 前缀和	166
12.3	Brent-Kung 并行前缀和	168
12.3.1	Brent-Kung 分段前缀和	169
12.3.2	两倍数据的 Brent-Kung 分段前缀和	170
12.3.3	避免 bank conflict 的两倍数据 Brent-Kung 分段前缀和	171
12.3.4	完整 Brent-Kung 前缀和	173
12.4	warp 分段的 Kogge-Stone 前缀求和	174
12.5	实验结果分析与结论	177
12.6	知识点总结	179
12.7	扩展练习	180

<b>第 13 章 排序</b> .....	181
13.1 串行排序及其性能 .....	181
13.1.1 选择排序 .....	181
13.1.2 冒泡排序 .....	182
13.1.3 快速排序 .....	182
13.1.4 基数排序 .....	183
13.1.5 双调排序网络 .....	185
13.1.6 合并排序 .....	186
13.1.7 串行排序性能对比 .....	187
13.2 基数排序 .....	188
13.2.1 基数排序概述 .....	188
13.2.2 单 block 基数排序 .....	189
13.2.3 基于 thrust 库的基数排序 .....	196
13.3 双调排序网络 .....	197
13.3.1 双调排序网络概述 .....	197
13.3.2 单 block 双调排序网络 .....	199
13.3.3 多 block 双调排序网络 .....	202
13.4 快速排序 .....	206
13.5 合并排序 .....	207
13.6 实验结果分析与结论 .....	208
13.7 知识点总结 .....	209
13.8 扩展练习 .....	210
<b>第 14 章 几种简单图像处理</b> .....	211
14.1 图像直方图统计 .....	211
14.1.1 串行直方图统计 .....	211
14.1.2 并行直方图统计 .....	211
14.1.3 实验结果与分析 .....	212
14.2 中值滤波 .....	213
14.2.1 串行中值滤波 .....	214
14.2.2 1D 并行中值滤波 .....	215
14.2.3 共享 1D 中值滤波 .....	216
14.2.4 双重共享 1D 中值滤波 .....	218
14.2.5 2D 并行中值滤波 .....	221
14.2.6 共享 2D 中值滤波 .....	222
14.2.7 共享 2D 中值滤波的改进 .....	227
14.2.8 实验结果与分析 .....	229
14.3 均值滤波 .....	231



14.3.1 串行均值滤波	231
14.3.2 并行均值滤波	232
14.3.3 实验结果与分析	233

## 第四篇 核 心 篇

<b>第 15 章 GPU 执行核心</b>	237
15.1 概述	237
15.2 算术运算支持	238
15.2.1 整数运算	238
15.2.2 浮点运算	239
15.3 算术运算性能	240
15.4 分支处理	242
15.5 同步与测时	246
15.5.1 同步	246
15.5.2 测时	247
15.6 数学函数	247
15.7 warp 与 block 原语	249
15.7.1 warp 原语	249
15.7.2 block 原语	250
15.8 kernel 启动、线程切换和循环处理	251
<b>第 16 章 GPU 存储体系</b>	254
16.1 概述	254
16.2 寄存器	259
16.3 局部存储	261
16.4 共享存储器	264
16.4.1 共享存储使用	264
16.4.2 bank conflict	265
16.4.3 volatile 关键字	266
16.4.4 共享存储原子操作	267
16.5 常量存储	268
16.6 全局存储	269
16.6.1 全局存储的使用	269
16.6.2 全局存储的合并访问	271
16.6.3 利用纹理缓存通道访问全局存储	271
16.7 纹理存储	273
16.7.1 CUDA 数组	273

16.7.2	纹理存储的操作和限制	274
16.7.3	读取模式、纹理坐标、滤波模式和寻址模式	276
16.7.4	表面存储	278
16.8	主机端内存	281
16.9	零拷贝操作	283
<b>第 17 章</b>	<b>GPU 关键性能测评</b>	284
17.1	GPU 性能测评概述	284
17.2	GPU 参数获取	286
17.2.1	GPU 选择	286
17.2.2	详细设备参数获取	287
17.3	精确测时方法汇总	288
17.3.1	clock 测时	289
17.3.2	gettimeofday 测时	289
17.3.3	CUDA 事件测时	289
17.3.4	cutil 库函数测时	290
17.4	GPU 预热与启动开销	290
17.5	GPU 浮点运算能力	291
17.6	GPU 访存带宽	293
17.7	GPU 通信带宽	295
17.8	NVIDIA Visual Profiler	296
17.9	程序性能对比约定	298
<b>第 18 章</b>	<b>CPUs 和 GPUs 协同</b>	299
18.1	协同优化基点	299
18.1.1	CPU 并行矩阵乘基点	299
18.1.2	GPU 并行矩阵乘基点	300
18.2	CPU/GPU 协同	300
18.3	多 GPU 协同	305
18.3.1	CUDA 版本	306
18.3.2	OpenMP+CUDA	308
18.3.3	MPI+CUDA	311
18.4	CPUs/GPUs 协同	314
18.4.1	CUDA 版本	314
18.4.2	OpenMP+CUDA	319
18.4.3	MPI+OpenMP+CUDA	324
18.5	本章小结	329

## 附 录

附录 A 判断法 1D 卷积代码 .....	333
附录 A.1 判断法 1D 卷积 basic 版 .....	333
附录 A.2 判断法 1D 卷积 constant 版 .....	334
附录 A.3 判断法 1D 卷积 shared 版 .....	336
附录 A.4 判断法 1D 卷积 cache 版 .....	337
附录 B 曼德博罗特集的系列优化代码 .....	340
附录 B.1 完整版串行 C 代码 .....	340
附录 B.2 cuda_1_0 .....	343
附录 B.3 cuda_0_2 .....	345
附录 B.4 cuda_zerocopy .....	346
附录 B.5 cuda_1_0_zerocopy .....	348
附录 B.6 cuda_0_0_zerocopy .....	349
附录 B.7 cuda_0_2_zerocopy .....	351
附录 B.8 cuda_2 .....	352
附录 B.9 cuda_1_2 .....	354
附录 C 几种图像处理完整源码 .....	357
附录 C.1 BMP 图像读写头文件 .....	357
附录 C.2 图像直方图串行代码 .....	373
附录 C.3 串行中值滤波代码 .....	374
附录 C.4 并行均值滤波相关代码 .....	376
附录 D nvprof 帮助菜单 .....	383
附录 E NVCC 帮助菜单 .....	388
附录 F 几种排序算法源代码 .....	399
附录 F.1 bitonic_sort_block 函数 .....	399
附录 F.2 GPU 快速排序完整代码 .....	400
附录 F.3 GPU 合并排序完整代码 .....	408
参考文献 .....	417



---

---

# 第一篇 理论篇

---

- 第 1 章 高性能计算概述
- 第 2 章 GPU 概述
- 第 3 章 GPU 硬件架构
- 第 4 章 GPU 软件体系
- 第 5 章 CUDA C 编程

理论篇主要阐述系统全面的 GPU 知识体系,包括 GPU 并行计算的领域背景(高性能计算概述)、GPU 概述、GPU 硬件架构、GPU 软件体系和 GPU 编程方法(CUDA C 编程)。通过阅读本篇,读者能够较为系统全面地掌握 GPU 相关的基础理论和技术知识,建立必要的知识结构。

本篇内容丰富,笔者推荐几项值得重点关注的内容:①向量机和阵列机的辨别,深入浅出地对比了向量机(CPU)和阵列机(GPU)的区别(1.5节);②GPU 购买注意事项(2.4节);③全面的 GPU 架构知识,包括 Tesla、Fermi、Kepler、Maxwell 和 Pascal 架构(3.1节);④实测可行的 CUDA 环境安装指南,包括 Windows、Linux 命令行和 Linux 图形界面等<sup>①</sup>(4.3节);⑤CUDA 编程七步曲,简述了 CUDA 编程的基本步骤(5.2节)。

---

① 同类 CUDA 书籍也有环境安装内容,但笔者亲测大都不可用,也许是笔者水平有限吧。