

Microsoft

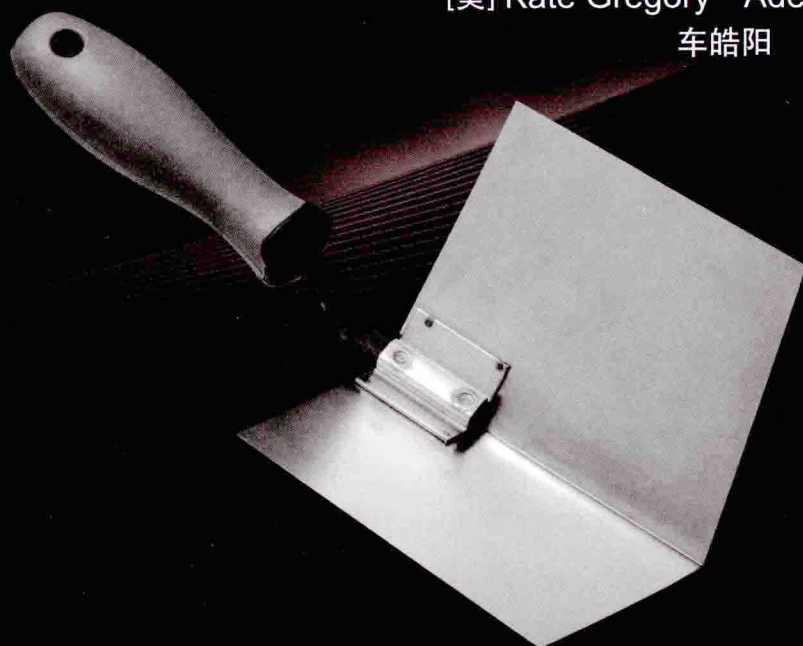
C++ AMP

Accelerated Massive Parallelism
with Microsoft Visual C++

C++ AMP

用Visual C++加速 大规模并行计算

[美] Kate Gregory Ade Miller 著
车皓阳 黄文龙 译

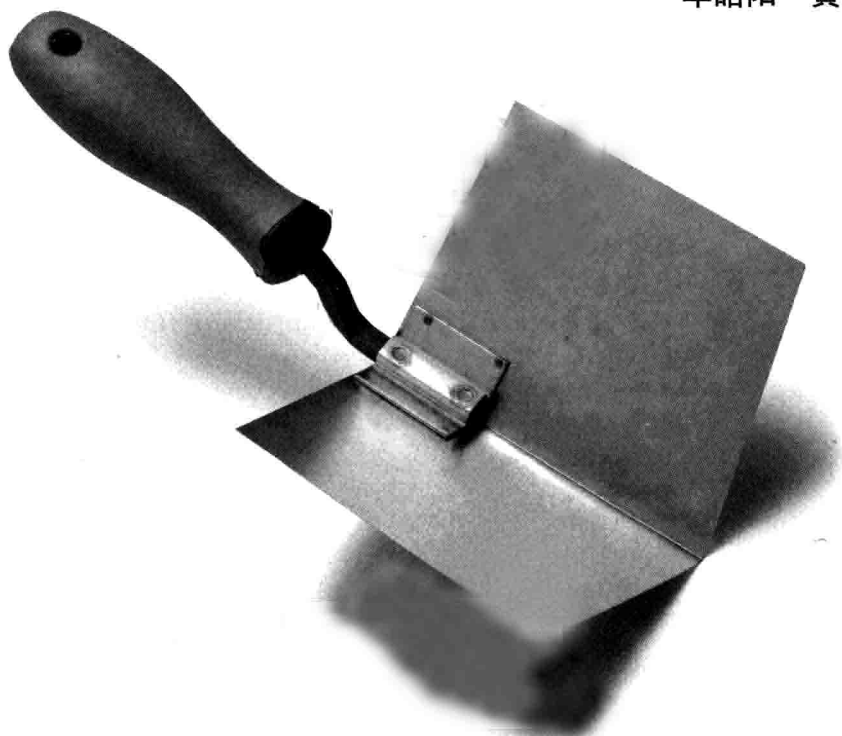


人民邮电出版社
POSTS & TELECOM PRESS

C++ AMP

用Visual C++加速 大规模并行计算

[美] Kate Gregory Ade Miller 著
车皓阳 黄文龙 译



人民邮电出版社
北京

图书在版编目(CIP)数据

C++ AMP : 用Visual C++加速大规模并行计算 /
(美) 格雷戈里 (Gregory, K.), (美) 米勒 (Miller, A.)
著; 车皓阳, 黄文龙译. — 北京: 人民邮电出版社,
2014. 4

ISBN 978-7-115-34503-5

I. ①C… II. ①格… ②米… ③车… ④黄… III. ①
C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第023589号

内 容 提 要

C++ AMP 是 Microsoft Visual Studio 和 C++ 编程语言的新扩展, 用于帮助开发人员充分适应现在和未来的高度并行和异构计算环境。C++ AMP 更易用, 与 C++ 配合得更好。本书的目标是帮助 C++ 开发人员在理解 C++ AMP 核心概念的基础上, 掌握更高级的功能。书中包含了 C++ AMP 程序开发的设计思路、使用方法、真实案例及代码示例, 并提供了案例的源代码。通过学习本书, 读者能理解如何在应用程序中最好地使用 C++ AMP, 还能利用 Microsoft Visual Studio 2012 提供的调试和剖析工具来解决问题、优化性能。

本书的两位作者一位是有 20 多年 C++ 实践经验的讲师、演讲人和作家, 一位是微软工作室的首席软件架构师, 他们有丰富的实践经验, 因此, 本书极具实际应用参考价值。本书适合专业 C++ 异构编程人员及熟悉 C++ 并希望从事并行编程及异构编程的开发人员。

-
- ◆ 著 [美] Kate Gregory Ade Miller
 - 译 车皓阳 黄文龙
 - 责任编辑 杨海玲
 - 责任印制 程彦红 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 20
 - 字数: 419 千字 2014 年 4 月第 1 版
 - 印数: 1—3 000 册 2014 年 4 月河北第 1 次印刷
 - 著作权合同登记号 图字: 01-2013-2651 号
-

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

Kate Gregory and Ade Miller, *C++ AMP: Accelerated Massive Parallelism with Microsoft Visual C++*
(ISBN: 9780735664739)

Copyright 2014 by Posts & Telecom Press.

Original English language edition copyright © 2012 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A. All rights reserved.

本书中文简体字版由美国微软出版社授权人民邮电出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

谨将本书献给 Brian，你是促使我内心变得强大的秘密武器；同时献给我可爱的孩子们，你们俨然已是小大人了，妈妈感谢你们的理解和支持。

——Kate Gregory

谨将本书献给 Susan，她更实至名归。

——Ade Miller

序

从整个计算的历史看，收益多来自于标量处理器性能指数级的增长。但是，目前这种增长趋势已臻极致，异构计算时代的曙光乍现。我们身边的应用几乎都是功耗敏感型的，所有的计算系统几乎都会遇到功耗限制，从端到云，未来计算平台势必要拥抱异构模型。例如，一些飞速发展的世界超级计算机已经开始使用异构的 CPU+GPU 计算集群。虽然 CUDA 和 OpenCL 等一代编程接口针对这些系统研发了新的功能库和应用，但我们仍然可以清晰地看到，异构并行软件开发需要更高的生产率。

异构计算领域最大的挑战是，任何编程接口若要提升生产率，势必要给开发者足够多的控制权，这样他们才能达成自己的性能目标。Microsoft 推出的 C++ AMP 正是解决这种挑战的一大技术进步。C++ AMP 接口可以说是一种简单而优雅的 C++ 语言扩展，它解决了上述接口的两大弱点。第一，以往的方法和 C++ 软件工程实践适配得并不好。基于内核的并行编程模型往往会干扰应用自身的类与对象的组织结构。第二，动态分配数组时使用 C 类型的索引往往会使局部性控制代码变得复杂。

C++ AMP 可以在并行代码里使用 C++ 循环结构和面向对象特性解决头一个问题，并且还可以使用 `array_view` 结构解决第二个问题，这真让人振奋。`array_view` 方法的设计比较有前瞻性，应用程序可以借其充分发挥即将来临的统一地址空间架构的优势。许多有经验的 CUDA 和 OpenCL 程序员都认为 C++ AMP 的程序设计风格更新颖、更优雅，也更高效。

依我看来，同样重要的是，C++ AMP 的出现让编译器上的许多调整从创新用法变为主流应用，C++ AMP 为其打开了方便之门，如数据布局调整和线程粒度调整。C++ AMP 还支持运行时数据迁移优化。程序员若想获得工作效率的巨大提升必然会需要这种技术进步。

虽然 C++ AMP 现在只有 Windows 实现版本，但它的接口是开放的，未来有可能会在其他平台上实现。如果其他平台供应商也可以提供它们的接口实现，那么 C++ AMP 就会产生更大的影响，而且这是非常有可能发生的事情。

本书的出版是异构并行计算的一个重要里程碑。我希望更多的开发者可以从本书中获益，以更有效率地开发异构并行应用程序。我很荣幸能为本书作序，并置身于这项伟大的进步中。更重要的是，我要向微软 C++ AMP 工程团队致敬，正是他们的辛勤劳作才使得这种进步成为可能。

——胡文美 (Wen-mei W. Hwu)

伊利诺伊大学香槟分校电子与计算机工程系教授兼 Sanders-AMD 中心主任
MulticoreWare 公司 CTO

前言

C++加速大规模并行计算(C++ Accelerated Massive Parallelism, C++ AMP)是微软公司发布的一套 C++应用程序加速技术,使用 C++ AMP 编写的代码可以有效地在图形处理器(Graphics-processing Unit, GPU)等支持数据并行的硬件上运行。C++ AMP 的定位不仅是要适应现代 GPU 和 APU 等并行硬件,而且还要保证应用代码未来的投资效用,对新近的并行硬件提供必要的支持。同时,C++ AMP 还是一种开放规范。微软的实现是基于 DirectX 构建的,支持跨不同硬件平台移植。其他实现版本也可以基于其他技术构建,因为规范本身并没有强制要求必须使用 DirectX。

C++ AMP 编程模型包含了类似现代 C++ STL 的模板库以及两项 C++语言扩展,它们都是 Visual C++ 2012 编译器的集成组件。Visual Studio 工具集对 C++ AMP 提供了全面支持,它的 IntelliSense 具有编辑、调试和剖析的功能。C++ AMP 使异构硬件成为主流,降低了它们的编程门槛,保证了程序员的开发效率。

本书要讲的是如何在应用程序中使用 C++ AMP。本书除了阐述 C++ AMP 的功能以外,还包含几个案例,演示了几个实际的应用程序实现,以及如何使用各种不同的方法去实现某些常见算法。读者可以下载这些案例的源代码以及每章的示例代码自己探索。

本书面向的读者

本书的目标是帮助 C++开发人员在理解 C++ AMP 核心概念的基础上再理解更高级的功能。如果你希望利用异构硬件来提升应用程序内既有功能的性能,或者增加以前由于性能限制而无法实现的全新功能,那么本书适合你。

读完本书以后,相信你能理解如何在应用程序中最好地使用 C++ AMP。我们还能利用 Microsoft Visual Studio 2012 提供的调试和剖析工具来解决问题、优化性能。

准备条件

在阅读本书之前,读者至少要有 Windows C++开发的实际经验,掌握面向对象编程的概念和 C++标准库(C++ Standard Library,经常被称作 STL,从先前的 Standard Template Library

而来)。了解一般的并行处理概念也会有所帮助，但这并不属于基本要求。虽然某些示例程序使用了 DirectX，但启用示例程序以及要理解其中的 C++ AMP 代码，并不需要具有任何 DirectX 的背景知识。

若想对 C++语言有一般性的了解，请阅读 Bjarne Stroustrup 的《The C++ Programming Language》(Addison-Wesley, 2000)。本书使用了许多 C++11 规范中新增的语言功能和库函数，这些功能在此书出版时非常新，当时并没有太多介绍。Scott Meyers 的《Presentation Materials: Overview of the New C++ (C++11)》对 C++11 规范进行了很好的讲解。我们可以在 Artima Developer 网站上在线购买这份材料：http://www.artima.com/shop/overview_of_the_new_cpp。Nicolai M. Josuttis 的《The C++ Standard Library: A Tutorial and Reference (2nd Edition)》(Addison-Wesley Professional, 2012) 这本书也是非常不错的标准库资料。

本书示例还大量使用了并行模式库 (Parallel Patterns Library) 和异步代理库 (Asynchronous Agents Library)。Colin Campbell 和 Ade Miller 的《Parallel Programming with Microsoft Visual C++》(Microsoft Press, 2011) 很好地介绍了这两套代码库。这本书可以从 MSDN 上 (<http://msdn.microsoft.com/en-us/library/gg675934.aspx>) 免费下载。

本书不适合的读者

本书并不是要讲解 C++或者标准库。本书首先假定读者已经拥有 C++语言和标准库的实际使用经验。本书并非并行编程乃至多线程编程的普通读本。如果读者目前仍不熟悉这些主题，可以选择性地阅读上节列举的一些参考读物。

本书的组织结构

本书共由 12 章组成。每一章都重点关注 C++ AMP 编程的一个方面。除了介绍 C++ AMP 各个方面的章节以外，本书还设计了三个案例，用来讲解在实际工作的应用程序中会用到的那些 C++ AMP 核心功能。各案例的代码以及其他章节里面讲解的示例代码，均可在 CodePlex 上下载。

各章标题	内容简介
第 1 章 C++ AMP 概述	介绍 GPU、异构计算、CPU 并行等主题，以及如何使用 C++ AMP 才能使应用程序发挥当今异构系统的效能
第 2 章 NBody 案例	使用 C++ AMP 实现 N 体模拟问题
第 3 章 C++ AMP 基础	概要地介绍 C++ AMP 库和语言的变化以及编码规则

各章标题	内容简介
第 4 章 分组	介绍分组技术，这种技术可以将计算拆解成称为 tile 的线程组，tile 可以共享访问速度非常快的可编程高速缓存
第 5 章 分组 NBody 案例	讲解第 2 章 N 体问题的计算分组版本
第 6 章 调试	回顾 Visual Studio 中调试 C++ AMP 应用程序的技术和工具
第 7 章 优化	更详细地介绍了影响 C++ AMP 应用程序性能的因素，包括如何应用度量程序性能，如何优化代码获得最快的运行速度
第 8 章 性能案例——归约	本章讲解了一个简单计算的各种实现版本，以及每种实现方式带来的性能提升差异
第 9 章 使用多个加速器工作	如何利用多个 GPU 达到最佳性能、如何利用交织平行性、如何使用 CPU 才能使 GPU 尽可能地高效运行
第 10 章 Cartoonizer 案例	本章讲解了一个复杂的示例，该示例将 CPU 并行与 C++ AMP 并行结合起来，支持多种加速器
第 11 章 图形互操作	使用 C++ AMP 和 DirectX
第 12 章 提示、技巧与最佳实践	说明如何应对一些异常的情形和场合，解决一些常见的问题
附录 其他资源	在线资源、支持资源和培训资源，面向那些想对 C++ AMP 有更多了解的读者

本书约定和特色

为了提高可读性，本书遵循下述约定。

- “注意事项”标记表示完成某个步骤需要注意的信息和替代方案。
- 两个名字中间的“+”号表示要同时按下这两个键。例如，“Alt+Tab”表示要同时按下 Alt 键和 Tab 键。
- 两个或多个菜单项之间的竖线表示菜单项的选择顺序（例如，File | Close）。

系统需求

需要用下述硬件与软件以构建和运行书中示例。

- Microsoft Windows 7 with Service Pack 1 或 Windows 8（x86 或 x64）。本书示例也可以在 Windows Server 2008 R2（x64）和 Windows Server 2012（x64）上运行，但并没有在这些操作系统上测试。
- Visual Studio 2012 任何版本都支持，但演练第 7 章和第 8 章中的性能剖析代码需要

专业版或旗舰版。

- 构建 NBody 案例需要安装 DirectX SDK (2010 年 6 月)。
- 计算机需要 1.6 GHz 或更高主频的处理器。推荐使用四核处理器。
- 1 GB (32 位) 或 2 GB (64 位) RAM。
- 10 GB 可用硬盘空间 (用于安装 Visual Studio 2012)。
- 5 400 转硬盘。
- 支持 DirectX 11 的显卡 (用于运行 C++ AMP 示例) 或者 1024×768 或更高分辨率的显示器 (用于 Visual Studio 2012)。
- DVD-ROM 驱动器 (如果用 DVD 安装 Visual Studio 2012 的话)。
- 互联网连接, 用于下载软件或各章示例。

代码示例

本书多数章节都有示例, 读者可以亲手试验, 更好地理解 and 掌握学到的新内容。这些示例可以在下述网址中找到:

<http://go.microsoft.com/fwlink/?Linkid=260980>

读者可以根据说明下载 zip 压缩源文件。



注意事项 除了代码示例以外, 你的系统还要安装 Visual Studio 2012 和 DirectX SDK (2010 年 6 月)。如果已经安装, 请补充安装各产品的最新服务安装包。

安装代码示例

请遵循下述步骤安装代码示例。

- (1) 请从本书配套网站 CodePlex (<http://ampbook.codeplex.com/>) 下载 zip 压缩源文件。读者可以在 Downloads 页签处下载最新版本。请选择最新的推荐下载 (recommended download) 选项。

- (2) 如遇提示, 请仔细阅读用户授权协议书的内容。如果选择接受协议内容, 请选择 Choose 选项, 然后单击 Next。
- (3) 解压压缩文件, 使用 Visual Studio 2012 打开文件 BookSamples.sln。



注意事项 如果授权协议书没有出现, 读者可以从 CodePlex (<http://ampbook.codeplex.com/license>) 网站上下载授权协议书。示例代码中也有一份协议书。

使用代码示例

解压示例后产生的 Samples 目录包含以下三类子目录。

- **CaseStudies:** 本目录包含了第 2 章、第 8 章和第 10 章中讲解的三个案例。每个案例各有一个目录。
 - **Nbody:** N 体引力模型。
 - **Reduction:** 用来演示性能取舍的一组归约求解算法实现。
 - **Cartoonizer:** 一套图像处理应用程序, 可以将从磁盘数据源加载或从视频摄像头捕捉的图像序列制作成卡通片。
- **Chapter 4、Chapter 7、Chapter 9、Chapter 11、Chapter 12:** 这些目录包含了对应章节的代码。
- **ShowAmpDevices:** 一个用来枚举 C++ AMP 本地支持设备的实用程序。

顶层 Samples 目录包含一个 Visual Studio 2012 解决方案文件 BookSamples.sln, 它包含了上述所有项目。可在 Debug 和 Release 模式下编译, 无警告或错误信息, 运行目标平台可以是 Win32 和 x64。每个项目还有自己的解决方案文件, 满足读者分别加载它们的需要。

致谢

没人可以独立完成一本书的所有内容。本书有两位作者, 同时, 许多人也都给予了无私的帮助。在这里作者要特别感谢下面这些人。

微软 C++ AMP 产品团队所做的工作远远超出了提供初稿评审反馈意见以及解答大量问题；感谢所有微软的同事们：Amit Agarwal、David Callahan、Charles Fu、Jerry Higgins、Yossi Levanoni、Don McCrady、Łukasz Mendakiewicz、Daniel Moth、Bharath Mysore Nanjundappa、Pooja Nagpal、James Rapp、Simon Wybranski、Lingli Zhang、Weirong Zhu。

微软 C++ AMP 团队还维护了一个微博，发布了许多珍贵的资料。上述 C++ AMP 产品团队的许多评审人员都在微博上发表了自己的观点。此外，下面这些人提供的素材也很有价值：微软同事 Steve Deitz、Kevin Gao、Pavan Kumar、Paul Maybee、Joe Mayo、Igor Ostrovsky。

微软同事 Ed Essey 和 Daniel Moth 帮助我们完成了整个项目的启动工作，接洽 O'Reilly 和有编写 C++ AMP 图书想法的作者们。他们还参与协调我们与 C++ AMP 产品团队之间的工作。

还要感谢负责编辑和制作的 Russell Jones、Holly Bauer 和 Carol Whitney，以及本书配图员 Rebecca Demarest。

我们非常幸运，通过 O'Reilly 的 Rough Cuts 项目，能够在 Safari 浏览器上提前对外发布本书初稿。许多人对本书早期初稿提供了自己的反馈意见，我们感谢他们的参与和兴趣。Bruno Boucard 和 Veikko Eeva 是其中对我们最有帮助的、最热忱的审稿人。

本书勘误与支持

我们付出全力保证本书和本书配套资料的准确性，本书出版后发现的错误都会张贴我们在 oreilly.com 网站上的微软出版社网站：

<http://go.microsoft.com/fwlink/?Linkid=260979>。

如果发现任何未列出的错误，可以通过同一网页报告我们。

如需额外支持，请发电子邮件给微软出版社图书支持：msspinput@microsoft.com。

请注意上述地址并不提供微软软件的产品支持。

我们期待你的反馈

微软出版社视读者满意为己任，视读者的反馈意见为无价之宝。请告知对于本书的看法：

<http://www.microsoft.com/learning/booksurvey>。

这个调查问卷非常短，我们会阅读你的品评和意见。提前感谢你的指导！

保持联系

我们可以在 Twitter 上继续交流，我们的 Twitter 网址是 <http://twitter.com/MicrosoftPress>。

作者介绍

Kate Gregory 有 20 多年的 C++ 应用经验，她是知名的讲师、演讲人和作家。管理、指导、技术写作、技术演讲占据了她许多的时间，但她仍然坚持每周都写程序。**Kate** 是十几本图书的作者，主要在 DevTeach、TechEd（美国、欧洲、非洲）和 TechDays 发表演讲。她是 C++ 的最有价值专家（Most Valuable Professional, MVP），多伦多 .NET 用户组的创始赞助者，多伦多东部 .NET 用户组的创始人，加拿大彼得伯勒市特伦特大学（Trent University）兼职教授。从 2002 年 1 月起，她就一直担任多伦多市的微软技术代言人，并于 2004 年 1 月，获得 Visual C++ 方向的微软最有价值专家称号。2005 年 6 月，她获得年度技术代言人（Regional Director）大奖。2011 年 2 月，她被授予 2010 年度 Visual C++ 最有价值专家称号。她的公司——Gregory Consulting Limited，总部设在安大略乡间，主要帮助客户使用新技术，适应不断变化的业务环境。

Ade Miller 目前是 Microsoft Studios 的首席软件架构师。他曾在微软公司担任过多种职务，包括：在 Windows HPC Server 团队担任程序经理，从事大数据平台方面的工作；担任过模式与实践组的敏捷工程团队的开发主管。他的主要兴趣是并行和分布式计算，以及通过工程化领导力提升团队的软件交付效率。

他是 *Parallel Programming with Microsoft .NET* 和 *Parallel Programming with Microsoft Visual C++* 两本书的作者之一。他还曾在微软以及其他地方发表和演讲过他关于并行计算以及敏捷软件开发等方面的经验。

目录

第 1 章 C++ AMP 概述	1
1.1 为什么选择 GPGPU? 什么是异构计算?	1
1.1.1 性能提升史	1
1.1.2 异构平台	2
1.1.3 GPU 架构	3
1.1.4 通过并行性提升性能的候选方案	4
1.2 CPU 并行技术	7
1.2.1 向量化	7
1.2.2 OpenMP	9
1.2.3 并发运行时库 (ConcRT) 和并行模式库	10
1.2.4 任务并行库	11
1.2.5 WARP——Windows 高级栅格化平台	11
1.2.6 GPU 并行技术	12
1.2.7 成功并行化的要求	13
1.3 C++ AMP 方法	14
1.3.1 C++ AMP 将 GPGPU (以及更多) 带进主流	14
1.3.2 C++ AMP 是 C++, 而不是 C	14
1.3.3 C++ AMP 使用了我们熟识的工具	15
1.3.4 C++ AMP 是一个近乎全面的代码库	15
1.3.5 C++ AMP 可以生成可移植的、不会过时的执行代码	17
1.4 小结	18
第 2 章 Nbody 案例	19
2.1 运行示例前的准备工作	19
2.2 运行 Nbody 示例	20
2.3 示例的结构	24
2.4 CPU 计算	25
2.4.1 数据结构	25
2.4.2 wWinMain 函数	26
2.4.3 OnFrameMove 回调函数	26
2.4.4 OnD3D11CreateDevice 回调函数	27
2.4.5 OnGUIEvent 回调函数	29
2.4.6 OnD3D11FrameRender 回调函数	30

2.5	CPU NBody 类	30
2.5.1	NBodySimpleInteractionEngine	30
2.5.2	NBodySimpleSingleCore	31
2.5.3	NBodySimpleMultiCore	31
2.5.4	NBodySimpleInteractionEngine::BodyBodyInteraction	32
2.6	C++ AMP 计算	33
2.6.1	数据结构	33
2.6.2	CreateTasks	35
2.7	C++ AMP NBody 类	37
2.7.1	NBodyAmpSimple::Integrate	37
2.7.2	BodyBodyInteraction	38
2.8	小结	40
第 3 章	C++ AMP 基础	41
3.1	array<T, N>	41
3.2	accelerator 与 accelerator_view	43
3.3	index<N>	45
3.4	extent<N>	46
3.5	array_view<T, N>	46
3.6	parallel_for_each	50
3.7	使用 restrict(amp) 标记的函数	52
3.8	在 CPU 和 GPU 之间复制数据	54
3.9	数学库函数	56
3.10	小结	57
第 4 章	分组	58
4.1	使用分组的目的和好处	58
4.2	tile_static 内存	60
4.3	tiled_extent	60
4.4	tiled_index<N1, N2, N3>	62
4.5	将简单算法改成分组算法	63
4.5.1	使用 tile_static 内存	64
4.5.2	分组栅和同步	68
4.5.3	完成简单版本到分组版本的修改	70
4.6	分组大小的影响	71
4.7	选择分组大小	74
4.8	小结	75

第 5 章	分组 NBody 案例	76
5.1	分组功能对 NBody 的提速有多大帮助	76
5.2	N 体问题的分组算法	77
5.2.1	NBodyAmpTiled 类	78
5.2.2	NBodyAmpTiled::Integrate	79
5.3	使用 Concurrency Visualizer	83
5.4	选择分组大小	88
5.5	小结	92
第 6 章	调试	93
6.1	第一步	93
6.1.1	选择 GPU 调试还是 CPU 调试	94
6.1.2	参考加速器	97
6.2	GPU 调试基础	100
6.2.1	熟悉的窗口和技巧	100
6.2.2	Debug Location 工具栏	101
6.2.3	检测竞态	101
6.3	检查线程运行状况	103
6.3.1	线程标记	104
6.3.2	GPU Threads 窗口	105
6.3.3	Parallel Stacks 窗口	107
6.3.4	Parallel Watch 窗口	108
6.3.5	对线程置标记、分组和过滤	110
6.4	施加更多的控制	112
6.4.1	冻结与解冻线程	113
6.4.2	Run Tile to Cursor	114
6.5	小结	116
第 7 章	优化	117
7.1	一种性能优化方法	117
7.2	分析性能	118
7.2.1	测量内核性能	118
7.2.2	使用 Concurrency Visualizer	121
7.2.3	使用 Concurrency Visualizer SDK	126
7.3	优化内存访问模式	127
7.3.1	别名和 parallel_for_each 调用	127
7.3.2	往返 GPU 的高效数据复制	131