

УДК 004.438C++AMP
ББК 32.973.202-018.2
Г79

Грегори, Кэйт.

Г79 С++ AMP: построение массивно параллельных программ с помощью Microsoft Visual C++ / К. Грегори, Э. Миллер ; пер. с англ. А. А. Слинкина. — 2-е изд., эл. — 1 файл pdf : 413 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-518-3

С++ Accelerated Massive Parallelism (С++ AMP) — разработанная корпорацией Microsoft технология ускорения написанных на С++ приложений за счет исполнения кода на оборудовании с распараллеливанием по данным, например, на графических процессорах. Модель программирования в С++ AMP основана на библиотеке, устройной по образцу STL, и двух расширениях языка С++, интегрированных в компилятор Visual C++ 2012. Она в полной мере поддерживается инструментами Visual Studio, в том числе IntelliSense, отладчиком и профилировщиком. Благодаря С++ AMP свойственная гетерогенному оборудованию производительность становится доступна широким кругам программистов.

В книге показано, как воспользоваться всеми преимуществами С++ AMP в собственных приложениях. Помимо описания различных черт С++ AMP, приведены примеры различных подходов к реализации различных алгоритмов в реальных приложениях.

Издание предназначено для программистов, уже работающих на С++ и стремящихся повысить производительность существующих приложений.

УДК 004.438C++AMP
ББК 32.973.202-018.2

Электронное издание на основе печатного издания: С++ AMP: построение массивно параллельных программ с помощью Microsoft Visual C++ / К. Грегори, Э. Миллер ; пер. с англ. А. А. Слинкина. — Москва : ДМК Пресс, 2013. — 412 с. — ISBN 978-5-94074-896-0. — Текст : непосредственный.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

ISBN 978-5-89818-518-3

© 2012 by Ade Miller, Gregory Consulting Limited

© Оформление, перевод на русский язык, ДМК Пресс, 2013



ОГЛАВЛЕНИЕ

Предисловие	13
Об авторах	15
Введение	16
Для кого предназначена эта книга	16
Предполагаемые знания	17
Для кого не предназначена эта книга.....	17
Организация материала	18
Принятые соглашения	19
Требования к системе	19
Примеры кода	20
Установка примеров кода.....	20
Использование примеров кода	21
Благодарности	21
Замеченные опечатки и поддержка книги	22
Нам важно ваше мнение	22
Оставайтесь на связи	23
Глава 1. Общие сведения и подход C++ AMP ...	24
Что означает GPGPU? Что такое гетерогенные вычисления? ...	24
История роста производительности.....	25
Гетерогенные платформы.....	26
Архитектура ГП.....	29
Кандидаты на повышение производительности за счет распараллеливания	30
Технологии распараллеливания вычислений на ЦП	34
Векторизация.....	34
OpenMP	37
Система Concurrency Runtime (ConcRT) и библиотека Parallel Patterns Library.....	39
Библиотека Task Parallel Library.....	41
WARP – Windows Advanced Rasterization Platform.....	41
Технологии распараллеливания вычислений на ГП.....	41
Что необходимо для успешного распараллеливания	43

Подход C++ AMP	45
C++ AMP вводит GPGPU (и не только) в обиход.....	45
C++ AMP – это C++, а не C	46
Для использования C++ AMP нужны только знакомые вам инструменты	47
C++ AMP почти целиком реализована на уровне библиотеки.....	48
C++ AMP порождает переносимые исполняемые файлы с прицелом на будущее	50
Резюме	52
Глава 2. Пример: программа NBody	53
Необходимые условия для запуска примера.....	53
Запуск программы NBody.....	55
Структура программы.....	59
Вычисления на ЦП	60
Структуры данных	60
Функция wWinMain	62
Обратный вызов OnFrameMove	62
Обратный вызов OnD3D11CreateDevice.....	63
Обратный вызов OnGUIEvent	65
Обратный вызов OnD3D11FrameRender	66
Классы NBody для вычислений на ЦП	66
Класс NBodySimpleInteractionEngine.....	67
Класс NBodySimpleSingleCore	67
Класс NBodySimpleMultiCore	68
Функция NBodySimpleInteractionEngine:: BodyBodyInteraction... ..	68
Вычисления с применением C++ AMP	70
Структуры данных	70
Функция CreateTasks	72
Классы NBody в версии для C++ AMP	74
Функция NBodyAmpSimple::Integrate	74
Функция BodyBodyInteraction	76
Резюме	77
Глава 3. Основы C++ AMP	79
Тип array<T, N>	79
accelerator и accelerator_view	82
index<N>	85
extent<N>	86
array_view<T, N>.....	86
parallel_for_each.....	91
Функции, помеченные признаком restrict(amp).....	94

Копирование между ЦП и ГП.....	96
Функции из математической библиотеки	98
Резюме	99

Глава 4. Разбиение на блоки 100

Назначение и преимущества блоков.....	101
Блочно-статическая память	102
Тип tiled_index<N1, N2, N3>	105
Преобразование простого алгоритма в блочный	106
Использование блочно-статической памяти	108
Барьеры и синхронизация	113
Окончательный вариант блочного алгоритма	116
Влияние размера блока.....	117
Выбор размера блока.....	120
Резюме	122

Глава 5. Пример: блочный вариант

программы NBody..... 124

Насколько разбиение на блоки повышает производительность программы NBody?.....	124
Блочный алгоритм решения задачи N тел	126
Класс NBodyAmpTiled.....	127
Метод NBodyAmpTiled::Integrate.....	127
Визуализатор параллелизма	133
Выбор размера блока.....	140
Резюме	144

Глава 6. Отладка 145

Первые шаги	145
Выбор режима отладки: на ЦП или на ГП	146
Эталонный ускоритель	150
Основы отладки на ГП.....	154
Знакомые окна и подсказки.....	154
Панель инструментов Debug Location.....	155
Обнаружение состояний гонки	156
Получение информации о нитях	158
Маркеры нитей	159
Окно GPU Threads.....	159
Окно Parallel Stacks	161
Окно Parallel Watch	163
Пометка, группировка и фильтрация нитей	165

Дополнительные способы контроля	168
Заморозка и разморозка нитей.....	168
Выполнение блока до текущей позиции	170
Резюме	172

Глава 7. Оптимизация..... 173

Подход к оптимизации производительности	173
Анализ производительности.....	174
Измерение производительности ядра.....	175
Использование визуализатора параллелизма.....	178
Использование пакета SDK визуализатора параллелизма	185
Способы оптимизации доступа к памяти	187
Совмещение и вызовы <code>parallel_for_each</code>	187
Эффективное копирование данных в память ГП и обратно	191
Эффективный доступ к глобальной памяти ускорителя	198
Массив структур или структура массивов.....	202
Эффективный доступ к блочно-статической памяти	205
Константная память	210
Текстурная память.....	211
Занятость и регистры.....	211
Оптимизация вычислений	213
Избегайте расходящегося кода.....	213
Выбор подходящей точности	218
Оценка стоимости математических операций	220
Развертывание циклов	220
Барьеры синхронизации	222
Режимы очереди	226
Резюме	227

Глава 8.Пример: программа Reduction 229

Постановка задачи	229
Отказ от ответственности.....	230
Структура программы.....	231
Инициализация и рабочая нагрузка.....	233
Маркеры визуализатора параллелизма.....	234
Функция <code>TimeFunc()</code>	235
Накладные расходы	237
Алгоритмы на ЦП	238
Последовательный алгоритм	238
Параллельный алгоритм	238
Алгоритмы с использованием C++ AMP	239
Простой алгоритм	240

Простой алгоритм с array_view	242
Простой оптимизированный алгоритм	244
Наивный блочный алгоритм	246
Блочный алгоритм с разделяемой памятью	248
Минимизация расхождения	254
Устранение конфликтов банков	256
Уменьшение числа простаивающих нитей	257
Развертывание цикла	258
Каскадная редукция	263
Каскадная редукция с развертыванием цикла	265
Резюме	266

Глава 9. Работа с несколькими ускорителями... 268

Выбор ускорителей	269
Перебор ускорителей	269
Ускоритель по умолчанию	272
Использование нескольких ГП	274
Обмен данными между ускорителями	279
Динамическое балансирование нагрузки	285
Комбинированный параллелизм	288
ЦП как последнее средство	290
Резюме	292

Глава 10. Пример: программа Cartoonizer 294

Необходимые условия	295
Запуск программы	295
Структура программы	299
Конвейер	301
Структуры данных	301
Метод CartoonizerDlg::OnBnClickedButtonStart()	303
Класс ImagePipeline	304
Стадия мультипликации	309
Класс ImageCartoonizerAgent	309
Реализации интерфейса IFrameProcessor	312
Использование нескольких ускорителей, совместимых с C++ AMP	321
Класс FrameProcessorAmpMulti	321
Разветвленный конвейер	324
Класс ImageCartoonizerAgentParallel	325
Производительность мультипликатора	328
Резюме	331

Глава 11. Интероперабельность с графикой ... 333

Основы	334
Типы norm и unorm	334
Типы коротких векторов	336
Тип texture<T, N>	340
Сравнение текстур и массивов	349
Использование текстур и коротких векторов	351
Встроенные функции HLSL	355
Интероперабельность с DirectX	356
Интероперабельность представления ускорителя и устройства Direct3D	357
Интероперабельность array и буфера Direct3D	358
Интероперабельность texture и текстурного ресурса Direct3D	359
Практическое использование интероперабельности с графикой	363
Резюме	365

Глава 12. Советы, хитрости и рекомендации... 367

Решение проблемы несоответствия размеру блока	368
Дополнение до кратного размеру блока	369
Отсечение блоков	371
Сравнение разных подходов	375
Инициализация массивов	376
Объекты-функции и лямбда-выражения	377
Атомарные операции	378
Дополнительные возможности C++ AMP Features в Windows 8	382
Обнаружение таймаутов и восстановление	384
Предотвращение TDR	385
Отключение TDR в Windows 8	386
Обнаружение TDR и восстановление	387
Поддержка вычислений с двойной точностью	388
Ограниченная поддержка двойной точности	388
Полная поддержка двойной точности	389
Отладка в Windows 7	389
Конфигурирование удаленной машины	390
Конфигурирование проекта	390
Развертывание и отладка проекта	392
Дополнительные отладочные функции	392
Развертывание	393

Развертывание приложения	393
Запуск C++ AMP на сервере	394
C++ AMP и приложения для Windows 8 в магазине Windows Store	397
Использование C++ AMP из управляемого кода	397
Из приложения .NET, приложения для Windows 7, Windows Store или библиотеки.....	397
Из приложения для C++ CLR.....	398
Из проекта для C++ CLR	398
Резюме	399
Приложение. Другие ресурсы	400
Другие публикации авторов этой книги	400
Сетевые ресурсы Microsoft	400
Скачивайте руководства по C++ AMP.....	401
Исходный код и поддержка.....	401
Обучение	402
Предметный указатель	403