

Оглавление

Предисловие	12
Введение	13
1. От графических процессоров к GPGPU	14
1.1. Производительность и параллелизм	14
1.2. Эволюция GPU	15
1.3. Сравнение архитектуры CPU и GPU	18
2. Программная модель CUDA	21
2.1. Основные принципы	21
2.2. Нити и блоки	22
2.3. Расширения языка	29
2.3.1. Атрибуты функций и переменных	30
2.3.2. Встроенные типы	31
2.3.3. Встроенные переменные	32
2.3.4. Оператор вызова GPU-ядра	32
2.3.5. Встроенные функции	33
2.4. CUDA runtime API	33
2.4.1. Асинхронное исполнение	34
2.4.2. Обработка ошибок в CUDA	36
2.4.3. Доступ к свойствам установленных GPU	37
2.5. Атомарные операции	39
2.5.1. Атомарные арифметические операции	40
2.5.2. Атомарные побитовые операции	42

2.5.3. Проверка статуса нитей варпа	42
2.5.4. Доступность и производительность атомарных операций . . .	43
3. Иерархия памяти	45
3.1. Константная память	47
3.2. Глобальная память	48
3.2.1. Кэширование	53
3.2.2. Пример: транспонирование матрицы	55
3.2.3. Пример: перемножение двух матриц	56
3.2.4. Оптимизация работы с глобальной памятью	57
3.3. Текстовая память	63
3.4. Общее виртуальное адресное пространство (UVA)	66
3.4.1. Пример: использование pinned-памяти хоста в GPU-ядре . . .	67
3.4.2. Пример: обмен данными напрямую между GPU	69
3.5. Разделяемая память	70
3.5.1. Пример: перемножение матриц	72
3.5.2. Эффективный доступ к разделяемой памяти	76
3.5.3. Пример: умножение матрицы на транспонированную	79
4. Взаимодействие CUDA и Fortran	84
4.1. Введение в CUDA Fortran	91
4.1.1. Элементы host-части программы	91
4.1.2. Программирование GPU-ядер	93
4.1.3. Правила передачи аргументов	94
4.1.4. Правила видимости	95
4.1.5. CUDA Fortran и CUDA C	95
4.1.6. Компиляция	96
4.1.7. Компактная форма записи	96
5. Некоторые алгоритмы обработки массивов	98
5.1. Параллельная редукция	98
5.2. Префиксная сумма (scan)	110
5.2.1. Реализация с помощью CUDA	111

5.2.2. Реализация с помощью CUDPP	118
6. Архитектура GPU	123
6.1. Архитектура GPU	123
6.2. Общие методы оптимизации CUDA-программ	130
7. Прикладные математические библиотеки	138
7.1. CUBLAS	139
7.1.1. Пример: <i>matmul</i>	140
7.1.2. Пример: степенной метод	141
7.2. CUSPARSE	148
7.2.1. Пример: решение треугольной линейной системы уравнений	150
7.3. CUFFT	153
7.3.1. Пример: решение уравнение Пуассона	156
7.4. CURAND	162
7.4.1. Пример: генерация показаний распределенных датчиков	164
8. Технологии для разработки на основе CUDA	167
8.1. Thrust	167
8.1.1. Простейшее преобразование на примере сложения векторов	167
8.1.2. Функторы на примере операции SAXPY	169
8.1.3. Трансформации общего вида, <i>zip_iterator</i>	171
8.1.4. Редукция	173
8.1.5. Производительность	176
8.1.6. Взаимодействие Thrust и CUDA C	183
8.1.7. Пример: расчет общего количества осадков	184
8.1.8. Переключение целевой платформы Thrust (backend)	186
8.1.9. Вызов Thrust из Fortran	190
8.2. PyCUDA	198
8.2.1. Введение	198
8.2.2. Простой пример работы с PyCUDA	199
8.2.3. Модуль <i>gpgpu</i> и взаимодействие с NumPy	200

9. Анализ работы приложений на GPU	204
9.1. Профилирование	204
9.1.1. CUDA events	204
9.1.2. CUDA profiler	205
9.2. Отладка	207
9.2.1. Принципы и терминология	207
9.2.2. GDB	208
9.2.3. CUDA-GDB	214
9.3. Диагностика	217
9.3.1. CUDA-MEMCHECK	217
10. Использование нескольких GPU	218
10.1. Контекст устройства	219
10.2. Fork	223
10.3. MPI	225
10.4. POSIX-потoki	227
10.5. Boost.Thread	233
10.6. OpenMP	238
11. CUDA Streams	241
11.1. Пример: перемножение матриц	242
11.2. Пример: взаимодействие между CUDA-ядром и хостом	246
11.3. Пример: использование нескольких устройств и асинхронное копирование	253
12. Решение уравнений Навье – Стокса на GPU	257
12.1. Метод покоординатного расщепления и соответствующий разностный метод первого порядка	258
12.1.1. Реализация метода прогонок на одном GPU	260
12.1.2. Реализации метода прогонок на нескольких GPU	262
12.2. Метод погруженной границы	265
12.2.1. Реализация для кластера с множеством GPU	268
12.2.2. Оптимизация метода сопряженных градиентов	269

13. Методы трассировки лучей на GPU	278
13.1. Обратная трассировка лучей	279
13.2. Поиск пересечений	282
13.3. Ускорение поиска пересечений	285
13.3.1. Регулярная сетка	285
13.3.2. Kd-дерево	289
13.4. Советы по оптимизации	292
Ссылки на источники	297
Приложение А. Установка и настройка CUDA	301
A.1. Windows 7	301
A.1.1. PGI Visual Fortran	313
A.2. Linux	314
A.3. Использование визуальной среды Eclipse совместно с CUDA	322
Приложение В. Счетчики профилирования	324