



Содержание

Глава 1. Существующие многоядерные системы.

Эволюция GPU. GPGPU	7
1.1. Многоядерные системы	8
1.1.1. Intel Core 2 Duo и Intel Core i7	8
1.1.2. Архитектура SMP	9
1.1.3. BlueGene/L	10
1.1.4. Архитектура GPU	11
1.2. Эволюция GPU	11

Глава 2. Модель программирования в CUDA.

Программно-аппаратный стек CUDA	17
2.1. Основные понятия	17
2.2. Расширения языка C	22
2.2.1. Спецификаторы функций и переменных	22
2.2.2. Добавленные типы	23
2.2.3. Добавленные переменные	23
2.2.4. Директива вызова ядра	23
2.2.5. Добавленные функции	24
2.3. Основы CUDA host API	26
2.3.1. CUDA driver API	27
2.3.2. CUDA runtime API	27
2.3.3. Основы работы с CUDA runtime API	31
2.3.4. Получение информации об имеющихся GPU и их возможностях	31
2.4. Установка CUDA на компьютер	34
2.5. Компиляция программ на CUDA	35
2.6. Замеры времени на GPU, CUDA events	41
2.7. Атомарные операции в CUDA	42
2.7.1. Атомарные арифметические операции	42
2.7.2. Атомарные побитовые операции	44
2.7.3. Проверка статуса нитей warp'a	44

Глава 3. Иерархия памяти в CUDA.

Работа с глобальной памятью	45
3.1. Типы памяти в CUDA	45
3.2. Работа с константной памятью	46
3.3. Работа с глобальной памятью	47
3.3.1. Пример: построение таблицы значений функции с заданным шагом	49
3.3.2. Пример: транспонирование матрицы	49
3.3.3. Пример: перемножение двух матриц	50

3.4. Оптимизация работы с глобальной памятью	51
3.4.1. Задача об N-телах	55

Глава 4. Разделяемая память в CUDA

и ее эффективное использование	59
4.1. Работа с разделяемой памятью	59
4.1.1. Оптимизация задачи об N телах	60
4.1.2. Пример: перемножение матриц	62
4.2. Паттерны доступа к разделяемой памяти	66
4.2.1. Пример: умножение матрицы на транспонированную	69

Глава 5. Реализация на CUDA базовых операций над массивами – reduce, scan, построения

гистограмм и сортировки	72
5.1. Параллельная редукция	72
5.2. Нахождение префиксной суммы (scan)	79
5.2.1. Реализация нахождения префиксной суммы на CUDA	80
5.2.2. Использование библиотеки CUDPP для нахождения префиксной суммы	86
5.3. Построение гистограммы	88
5.4. Сортировка	98
5.4.1. Битоническая сортировка	98
5.4.2. Поразрядная сортировка	101
5.4.3. Использование библиотеки CUDPP	102

Глава 6. Архитектура GPU, основы PTX

6.1. Архитектура GPU Tesla 8 и Tesla 10	106
6.2. Введение в PTX	108
6.2.1. Типы данных	111
6.2.2. Переменные	112
6.2.3. Основные команды	114

Глава 7. Иерархия памяти в CUDA.

Работа с текстурной памятью	121
7.1. Текстурная память в CUDA	122
7.2. Обработка цифровых сигналов	123
7.2.1. Простые преобразования цвета	124
7.2.2. Фильтрация. Свертка	128
7.2.3. Обнаружение границ	134
7.2.4. Масштабирование изображений	137

Глава 8. Взаимодействие с OpenGL

8.1. Создание буферного объекта в OpenGL	142
8.2. Использование классов	143

8.3. Пример шума Перлина	147
8.3.1. Применение	150
Глава 9. Оптимизации	152
9.1. PTX-ассемблер	155
9.1.1. Занятость мультипроцессора	156
9.1.2. Анализ PTX-ассемблера	157
9.2. Использование CUDA-профайлера	161
Приложение 1. Искусственные нейронные сети	163
П1.1. Введение	163
П1.1.1. Задачи классификации (Classification)	163
П1.1.2. Задачи кластеризации (Clustering)	164
П1.1.3. Задачи регрессии и прогнозирования	164
П1.2. Модель нейрона	165
П1.3. Архитектуры нейронных сетей	166
П1.4. Многослойный перцептрон	166
П1.4.1. Работа с многослойным перцептроном	167
П1.4.2. Алгоритм обратного распространения ошибки	169
П1.4.3. Предобработка данных	171
П1.4.4. Адекватность данных	171
П1.4.5. Разбиение на наборы	171
П1.4.6. Порядок действий при работе с многослойным перцептроном	172
П1.5. Перцептроны и CUDA	173
П1.5.1. Пример задачи реального мира	174
П1.6. Литература	178
Приложение 2. Моделирование распространения волн цунами на GPU	179
П2.1. Введение	179
П2.2. Математическая постановка задачи	181
П2.3. Программная модель	183
П2.4. Адаптация алгоритма под GPU	186
П2.5. Заключение	191
П2.6. Литература	191
Приложение 3. Применение технологии NVIDIA CUDA для решения задач гидродинамики	193
П3.1. Введение	193
П3.2. Сеточные методы	194
П3.2.1. Геометрический многосеточный метод	195
П3.2.2. Алгебраический многосеточный метод	197
П3.2.3. Метод редукции	198

П3.2.4. Оценка эффективности	199
П3.3. Метод частиц	200
П3.4. Статистическая обработка результатов	201
П3.5. Обсуждение	202
П3.6. Литература	203

Приложение 4. Использование технологии CUDA при моделировании динамики пучков

в ускорителях заряженных частиц	205
П4.1. Введение	205
П4.2. Особенности задачи	205
П4.3. Использование многоядерных процессоров	208
П4.4. Реализация на графических процессорах	210
П4.5. Результаты	214
П4.6. Литература	216

Приложение 5. Трассировка лучей

П5.1. Обратная трассировка лучей	219
П5.1.1. Поиск пересечений	221
П5.1.2. Проблемы трассировки лучей на GPU	222
П5.1.3. Ускорение поиска пересечений	223
П5.2. Оптимизация трассировки лучей для GPU	228
П5.2.1. Экономия регистров	228
П5.2.2. Удаление динамической индексации	229
П5.3. Литература	230