



Бриан Тоуманнен

Программирование GPU при помощи Python и Cuda

Содержание

| | |
|--|----|
| Об авторе | 10 |
| О рецензенте | 11 |
| Предисловие | 12 |
| | |
| Глава 1. Почему программирование GPU? | 18 |
| Технические требования..... | 19 |
| Параллелизация и закон Амдала..... | 19 |
| Использование закона Амдала | 21 |
| Множество Мандельброта..... | 22 |
| Профилировка вашего кода..... | 25 |
| Использование модуля cProfile..... | 25 |
| Резюме..... | 26 |
| Вопросы..... | 27 |
| | |
| Глава 2. Настройка окружения для программирования GPU | 28 |
| Технические требования..... | 29 |
| Убедитесь, что у вас есть требуемое оборудование | 29 |
| Проверка вашего оборудования (Linux)..... | 30 |
| Проверка вашего оборудования (Windows) | 31 |
| Установка драйверов для GPU | 33 |
| Установка драйверов GPU (Linux)..... | 33 |
| Установка драйвера GPU (Windows) | 35 |
| Установка окружения для программирования на C++..... | 35 |
| Настройка GCC, Eclipse IDE и графических зависимостей (Linux)..... | 35 |
| Установка Visual Studio (Windows) | 36 |
| Установка CUDA Toolkit | 38 |
| Установка окружения Python для программирования GPU..... | 39 |
| Установка PyCUDA (Linux)..... | 40 |
| Создание скрипта для настройки окружения (Windows)..... | 40 |
| Установка PyCUDA (Windows) | 41 |
| Проверка PyCUDA | 42 |
| Резюме..... | 42 |
| Вопросы..... | 43 |

| | |
|--|-----|
| Глава 3. Начало работы с PyCUDA | 44 |
| Технические требования | 44 |
| Опрос вашего GPU | 45 |
| Опрос вашего GPU при помощи PyCUDA..... | 46 |
| Использование класса <code>gpuarray</code> модуля PyCUDA | 49 |
| Перенос данных в и из GPU при помощи <code>gpuarray</code> | 49 |
| Использование основных поэлементных операций через методы <code>gpuarray</code> | 50 |
| Использование <code>ElementWiseKernel</code> из PyCUDA для выполнения поэлементных операций..... | 55 |
| Возвращаемся к множеству Мандельброта | 58 |
| Краткая вылазка в функциональное программирование | 61 |
| Основа параллельного сканирования и редуцирования..... | 63 |
| Резюме..... | 64 |
| Вопросы..... | 65 |
| | |
| Глава 4. Ядра, нити, блоки и сетки | 66 |
| Технические требования | 67 |
| Ядра | 67 |
| Функция <code>SourceModule</code> из PyCUDA..... | 67 |
| Нити, блоки и сетки..... | 70 |
| Игра «Жизнь» Джона Конвея | 70 |
| Синхронизация и взаимодействие нитей | 77 |
| Использование функции устройства <code>__syncthreads()</code> | 77 |
| Использование разделяемой памяти..... | 80 |
| Алгоритм параллельной префиксной суммы..... | 82 |
| Алгоритм наивный параллельной префиксной суммы..... | 82 |
| Исключающая префиксная сумма и включающая префиксная сумма | 85 |
| Эффективный алгоритм параллельной префиксной суммы | 85 |
| Эффективный алгоритм параллельной префиксной суммы (реализация) | 87 |
| Резюме..... | 89 |
| Вопросы..... | 90 |
| | |
| Глава 5. Поток, события, контексты и одновременность | 91 |
| Технические требования | 92 |
| Синхронизация устройства CUDA | 92 |
| Использование класса <code>stream</code> из PyCUDA | 93 |
| Параллельная игра «Жизнь» Конвея при помощи потоков CUDA | 97 |
| События..... | 100 |
| События и потоки..... | 102 |
| Контексты..... | 103 |
| Синхронизация в текущем контексте..... | 104 |

| | |
|---|-----|
| Создание контекста | 105 |
| Многопроцессность и многонитиевость на стороне хоста | 106 |
| Различные контексты для параллельности на стороне хоста | 107 |
| Резюме | 110 |
| Вопросы | 111 |

Глава 6. Отладка и профилирование вашего кода на CUDA

| | |
|--|-----|
| Технические требования | 113 |
| Использование <code>printf</code> внутри ядер CUDA | 113 |
| Использование <code>printf</code> для отладки | 115 |
| Заполняем пробелы в CUDA C | 119 |
| Использование NSight IDE для разработки и отладки кода на CUDA C | 124 |
| Использование NSight с Visual Studio IDE под Windows | 125 |
| Использование NSight с Eclipse под Linux | 128 |
| Использование NSight для понимания варпа в CUDA | 131 |
| Использование профайлера <code>nvprof</code> и Visual Profiler | 134 |
| Резюме | 136 |
| Вопросы | 136 |

Глава 7. Использование библиотек CUDA

вместе со Scikit-CUDA

| | |
|--|-----|
| Технические требования | 137 |
| Установка Scikit-CUDA | 138 |
| Базовая линейная алгебра при помощи <code>cuBLAS</code> | 139 |
| Функции 1-го уровня AXPY в <code>cuBLAS</code> | 139 |
| Другие функции <code>cuBLAS</code> 1-го уровня | 141 |
| GEMV 2-го уровня в <code>cuBLAS</code> | 142 |
| Функции 3-го уровня GEMM в <code>cuBLAS</code> для измерения производительности GPU | 144 |
| Быстрое преобразование Фурье при помощи <code>cuFFT</code> | 147 |
| Простое одномерное FFT | 148 |
| Использование FFT для свертки | 149 |
| Использование <code>cuFFT</code> для двумерной свертки | 150 |
| Использование <code>cuSolver</code> из Scikit-CUDA | 155 |
| Сингулярное разложение (SVD) | 155 |
| Использование SVD для анализа методом главных компонент (PCA) | 156 |
| Резюме | 158 |
| Вопросы | 158 |

Глава 8. Библиотеки функций для GPU CUDA и Thrust

| | |
|--|-----|
| Технические требования | 160 |
| Библиотека функций GPU <code>cuRAND</code> | 160 |

| | |
|---|------------|
| Оценка π при помощи метода Монте-Карло..... | 161 |
| CUDA Math API | 165 |
| Краткий обзор определенных интегралов | 165 |
| Вычисление определенного интеграла при помощи метода Монте-Карло | 166 |
| Пишем тесты | 172 |
| Библиотека CUDA Thrust | 174 |
| Использование функторов в Thrust | 176 |
| Резюме..... | 178 |
| Вопросы..... | 178 |
| Глава 9. Реализация глубокой нейросети..... | 180 |
| Технические требования | 181 |
| Искусственные нейроны и нейросети | 181 |
| Реализация плотного слоя искусственных нейронов | 182 |
| Реализация слоя мягкого максимума | 187 |
| Реализация функции потери перекрестной энтропии..... | 189 |
| Реализация последовательной сети..... | 189 |
| Реализация методов вывода..... | 191 |
| Градиентный спуск..... | 193 |
| Подготовка и нормализация данных | 197 |
| Данные Iris | 197 |
| Резюме..... | 200 |
| Вопросы..... | 200 |
| Глава 10. Работа с компилированным кодом для GPU | 201 |
| Запуск откомпилированного кода при помощи Stypes..... | 202 |
| Снова возвращаемся к вычислению множества Мандельброта | 202 |
| Компиляция и запуск PTX-кода..... | 208 |
| Написание «обертки» для CUDA Driver API..... | 209 |
| Использование CUDA Driver API | 213 |
| Резюме..... | 216 |
| Вопросы..... | 217 |
| Глава 11. Оптимизация быстродействия в CUDA | 218 |
| Динамический параллелизм | 219 |
| Быстрая сортировка при помощи динамического параллелизма | 220 |
| Векторные типы данных и доступ к памяти..... | 222 |
| Потокобезопасные атомарные операции..... | 224 |
| Перестановки в пределах варпа | 225 |
| Вставка PTX-ассемблера прямо в код..... | 228 |

| | |
|--|------------|
| Оптимизированная по быстродействию версия суммирования элементов массива | 232 |
| Резюме | 235 |
| Вопросы | 235 |
| Глава 12. Куда идти далее? | 237 |
| Расширение знаний о CUDA и программировании GPGPU | 238 |
| Системы из нескольких GPU | 238 |
| Кластерные вычисления и MPI | 238 |
| OpenCL PyOpenCL | 239 |
| Графика | 239 |
| OpenGL | 240 |
| DirectX12 | 240 |
| Vulkan | 240 |
| Машинное обучение и компьютерное зрение | 241 |
| Основы | 241 |
| cuDNN | 241 |
| Tensorflow и Keras | 242 |
| Chainer | 242 |
| OpenCV | 242 |
| Технология блокчейн | 242 |
| Резюме | 243 |
| Вопросы | 243 |
| Ответы на вопросы | 244 |
| Предметный указатель | 250 |