

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа ТП.22Б07-мм

Экспериментальное исследование обработки изображений с использованием GPGU

БУРАШНИКОВ Артем Максимович

Отчёт по учебной практике
в форме «Эксперимент»

Научный руководитель:
доцент кафедры информатики, к. ф.-м. н., С. В. Григорьев

Санкт-Петербург
2023

Оглавление

| | |
|--|-----------|
| Введение | 3 |
| 1. Постановка задачи | 4 |
| 2. Обзор предметной области | 5 |
| 3. Детали реализации | 6 |
| 4. Подготовка тестового стенда | 7 |
| 4.1. Характеристики оборудования | 7 |
| 4.2. Исследовательские гипотезы | 7 |
| 4.3. Используемые метрики | 8 |
| 4.4. Инструменты для измерений | 8 |
| 4.5. Набор данных | 9 |
| 4.6. Эксперимент | 9 |
| Заключение | 11 |

Введение

1 Постановка задачи

Целью работы является проведение экспериментального исследования производительности наивной обработки изображений с использованием GPGPU и анализ следующего исследовательского вопроса.

- **RQ1**

При каких размерах изображения выгоднее проводить обработку на GPU, а при каких на CPU?

Для выполнения поставленной цели была сформулирована задача.

- Найти критическое значение размера изображения, при котором обработка на GPU демонстрирует ускорение.

2 Обзор предметной области

3 Детали реализации

4 Подготовка тестового стенда

В этом разделе указаны характеристики оборудования, на котором проводились исследования. Также были выдвинуты необходимые гипотезы, проверке которых посвящены проводимые эксперименты, и обозначены используемые метрики и инструменты для фиксирования измерений. Отдельно прокомментирован выбор набора тестируемых данных.

4.1 Характеристики оборудования

Аппаратная конфигурация обладает приведенными ниже характеристиками.

Операционная система

Operating System: Ubuntu 22.04.2 LTS

CPU

Architecture: x86_64
Model name: AMD Ryzen 5 4500U with Radeon Graphics
Thread(s) per core: 1
Core(s) per socket: 6

RAM

Total (MB): 9351

GPU

Device: [AMD/ATI] Renoir (rev c3)
Memory: 256M

4.2 Исследовательские гипотезы

Анализ поставленных задач позволил выдвинуть следующую гипотезу.

Гипотеза №1

Обработка изображений при помощи GPU должна демонстрировать ускорение даже для изображений сравнительно небольшого размера.

4.3 Используемые метрики

Для исследования *Гипотезы №1* измерено ускорение алгоритма (*англ.* — *Speedup*) применения фильтра с использованием GPU относительно алгоритма применения этого же фильтра с использованием CPU:

Speedup (S) вычисляется по формуле:

$$S = \frac{T_{old}}{T_{new}}, \quad (1)$$

где T_{cpu} — время применения фильтра на CPU, T_{gpu} — время применения фильтра на GPU.

4.4 Инструменты для измерений

Все замеры выполнены с использованием библиотеки для измерения производительности `BenchmarkDotNet v0.13.4`¹, разрабатываемой и поддерживаемой для платформы .NET. Ниже приведены настройки инструмента (для экспериментов использованы настройки по умолчанию).

- **launchCount** - общее количество запусков конкретного тестового метода. По умолчанию 1.
- **warmupCount** - количество итераций стадии разгрева. По умолчанию рассчитывается эвристикой.
- **iterationTime** - желаемое время исполнения итерации. По умолчанию 500 мс.
- **iterationCount** - количество итераций для тестового метода. По умолчанию рассчитывается эвристикой.

¹Библиотека .NET для замеров производительности. Дата посещения: 23 мая 2023 г.

- **invocationCount** - количество вызовов метода за одну итерацию. По умолчанию рассчитывается эвристикой.
- **unrollFactor** - сколько раз тестовый метод будет вызван за одну итерации в рамках одного тестового цикла. По умолчанию 16.
- **runOncePerIteration** - флаг, отвечающий за запуск тестового метода ровно один раз за итерацию. По умолчанию *False*.

4.5 Набор данных

Для фиксации исследуемых величин выбраны 5 изображений различного размера, имеющих квадратную матрицу пикселей. Такой выбор обусловлен тем, что для проверки гипотезы 4.2 достаточно провести измерения на одинаковых изображениях, причем форма матрицы на производительность алгоритмов не оказывает влияния. Выбранные изображения представлены в таблице 1. В качестве фильтра зафиксировано применение гауссовского размытия. Выбор данного фильтра обусловлен его базовыми свойствами, при которых можно провести качественные выводы.

| Изображение | Ширина, пикселей | Высота, пикселей |
|-------------|------------------|------------------|
| 100x100.jpg | 100 | 100 |
| 200x200.jpg | 200 | 200 |
| 300x300.jpg | 300 | 300 |
| 474x474.jpg | 400 | 400 |

Таблица 1: Набор изображений для проведения экспериментального исследования

4.6 Эксперимент

Проверка инструмента измерения производительности

Измерения, полученные с помощью **BenchmarkDotNet**, проверены на нормальность при помощи библиотек языка Python: **SciPy v1.10.1**²,

²Библиотека языка Python для проведения научных и инженерных расчетов. Дата посещения: 23 мая 2023 г.

NumPy v1.24.3³ и Matplotlib v3.7.1⁴. Для этого отдельно выполнены 100 измерений производительности алгоритма на графе `mycielskian12` с параметром `parallelLevel = 3`. Граф выбран как предположительно обладающий достаточным для эффективного применения асинхронных вычислений количеством вершин. Значение `parallelLevel` выбиралось как предположительно оптимальное. Результаты измерений зафиксированы в таблице 2.

| Время 1 (мс) | Время 2 (мс) | Время 3 (мс) | Время 4 (мс) | Время 5 (мс) | Время 6 (мс) | Время 7 (мс) | Время 8 (мс) | Время 9 (мс) | Время 10 (мс) |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|

Таблица 2: Измерения, округленные до сотых значений, полученные во время работы параллельной версии алгоритма на графе `mycielskian12`, `parallelLevel = 3`

С помощью библиотеки `scipy` вызовом функций `stats.shapiro()` и `stats.normaltest()` проверено соответствие полученных величин нормальному распределению. Значение `pvalue` соответствовало 0.15 (Шapiro) и 0.13 (Пирсон), что удовлетворяет критерию нормальности. По данным в таблице 2 построена гистограмма `??`. Вид гистограммы соответствует нормальному распределению.

Анализ результатов

Ниже даны ответы на вопросы исследования. Результаты измерения ускорения представлены на рис 3.

Из графика видно, что наивное применение фильтра с использованием GPU становится быстрее для изображения 474×474 . Это объясняется тем, что использование параллельных вычислений с помощью GPU даёт выигрыш в производительности пре идентичных операциях, которые необходимо применять на массив пикселей.

³Библиотека языка Python для математических вычислений. Дата посещения: 23 мая 2023 г.

⁴Библиотека языка Python для визуализации данных. Дата посещения: 23 мая 2023 г.

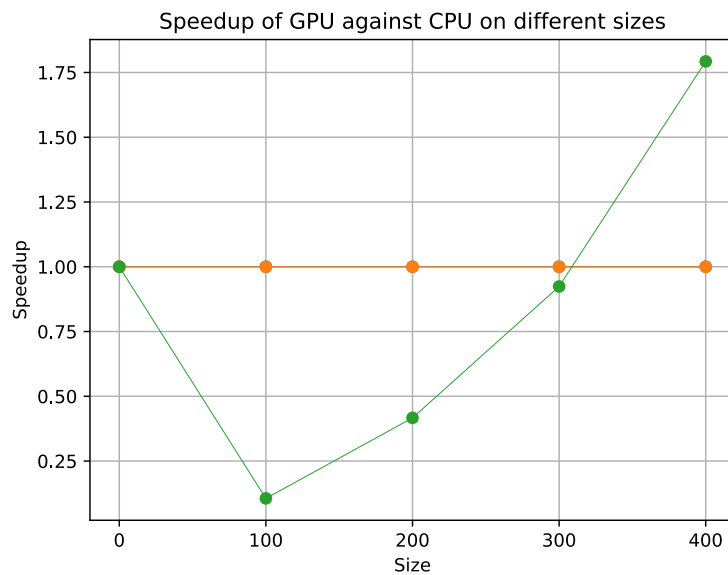


Таблица 3: Ускорение применения фильтра на GPU относительно CPU для изображения 100x100

Заключение

В рамках выполнения данной работы проведено экспериментальное исследование. Получены следующие результаты.

- Оценена скорость применения фильтра с использованием графического процессора и найдено критическое значение размера, при котором использование GPU даёт ускорение.