[Тероия Параллелизма](https://classroom.google.com/u/1/c/NTg0Nzg0MTE5Mzgy" \t "_self)

Отчет

Уравнение теплопроводности

Выполнил группа 21933, Рюмин Вадим Юрьевич

Дата 07.03.2023

Цели работы

Реализовать решение уравнение теплопроводности (пятиточечный шаблон) в двумерной области на  равномерных сетках (128^2, 256^2,  
512^2, 1024^2). Граничные условия – линейная интерполяция между углами области. Значение в углах – 10, 20, 30, 20. Ограничить точность – 10^-6 и максимальное число итераций – 10^6.

Используемый компилятор

pgcc

Используемый профилировщик

Nsight Systems

Как производили замер времени работы

Для однопоточной программы, при помощи библиотеки “time.h”

Для многопоточной программы, при помощи nvprof

Для программы на, при помощи GPU PGI\_ACC\_TIME=1

Выполнение на CPU

CPU-onecore

|  |  |  |  |
| --- | --- | --- | --- |
| Размер сетки | Время  Выполнения(s) | Точность | Количество  итераций |
| 128\*128 | 2 | 10^-6 | 35085 |
| 256\*256 | 35 | 10^-6 | 123706 |
| 512\*512 | 404 | 10^-6 | 417181 |

CPU-multicore(./cpu\_m\_2)

|  |  |  |  |
| --- | --- | --- | --- |
| Размер сетки | Время  Выполнения(s) | Точность | Количество  итераций |
| 128\*128 | 1,5 | 10^-6 | 35085 |
| 256\*256 | 6 | 10^-6 | 123706 |
| 512\*512 | 41 | 10^-6 | 417181 |
| 1024\*1024 | 291 | 40^-6 | 10^6 |

Диаграмма сравнения время работы СPU-one и CPU-multi

Выполнение на GPU

Этапы оптимизации на сетке 512\*512

(количество итераций при профилировании 100)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап No | Время  Выполнения (ms) | Точность | Количество  итераций | Комментарии  (что было сделано) |
| 1 | 1,77 | 0.049 | 100 | Одномерный массив |
| 2 | 6,46 | 0.071 | 100 | Двумерный массив |

Диаграмма оптимизации

(по горизонтали номер этапа; по вертикали время работы)

GPU – оптимизированный вариант

|  |  |  |  |
| --- | --- | --- | --- |
| Размер сетки | Время  выполнения | Точность | Количество  итераций |
| 128\*128 | 1,44 | 10^-6 | 35945 |
| 256\*256 | 8,38 | 10^-6 | 125016 |
| 512\*512 | 25,6 | 10^-6 | 441293 |
| 1024\*1024 | 81,35 | 40^-6 | 10^6 |

Диаграмма сравнения времени работы CPU-one, CPU-multi, GPU(оптимизированный вариант) для разных размеров сеток

Вывод:

Двумерный массив работает быстрее чем одномерный.

На этапах оптимизации при 100 итерациях одномерный массив работает быстрее. Но при большем количестве итерации двумерный массив работает быстрее.

GPU имеет больше количество ядер чем CPU, поэтому расчёт на ней занимает намного меньше времени чем на CPU. Но время обмена данными CPU и GPU занимает время.

Поэтому расчёт матриц размером 128\*128 и 256\*256 быстрее происходит на многопоточном CPU. А расчёт матриц размером 512\*512 и 1024\*1024 быстрее происходит на GPU.

Приложение

Ссылка на GitHub

[CS/Two at main · RyuminVadim/CS (github.com)](https://github.com/RyuminVadim/CS/tree/main/Two)

Программы для CPU и GPU(скриншот)

Программа GPU

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, табличка

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, электроника, снимок экрана

Автоматически созданное описание

Программа CPU

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание