**Теория Параллелизма**

**Отчет:**

“Уравнение теплопроводности”

(cublas + openacc)

Емельянов Алексей Алексеевич

Группа 21932

Новосибирск 2023

**Цель работы**

Цель данной работы состоит в том чтобы реализовать решение теплопроводности методом Якоба для двумерной сетки. Произвести оптимизацию предоставленного кода из лекции. Произвести сравнения по времени работы между CPU и GPU. ([ссылка на презентацию](https://classroom.google.com/u/1/c/NTg0Nzg0MTE5Mzgy/p/NTk0MTI0MzYzNDg5?pli=1)).

Для компиляции используются следующие команды:

* **CPUMylti -** *Pgc++ Paralel3.cpp p\_multiCore.pg -o -fast -acc=multicore -O2 -Mconcur=allcores*
* **CPU -** *Pgc++ Paralel3.cpp p\_oneCore.pg -o -fast -acc -O2*
* **GPU -***Pgc++ Paralel3.cpp p\_cpu.pg -o -fast -acc=gpu -O2 -D OPENACC\_\_*

Для компиляции cublas используем следующию команду

* **GPU -***Pgc++ Paralel3.cpp -o p\_cpu.pg -fast -acc=gpu -O2 -D OPENACC\_\_ -Mcudalib=cublas*

Для данной работы в качестве измерения времени выполнения программы использую команду при запуске программы “*time*” предоставленной на лекции.

Для запуска собранный программы используется следующая команда

*time (имя файлы) (значение ошибки) (размер сетки одной стороны) (максимальное число итераций)*

*пример time p.pg 0.0000001 128 10000000*

*ссылка на гитхаб - https://github.com/EmelyanovAlexey/Paralel/tree/master/task\_3*

**Таблица результатов выполнения программы**

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | | | |
| **Размер ячейки** | **Время выполнения** | **Результат ошибки** | **Количество итераций** |
| 128x128 | 1,389s | 9.7835e-07 | 11081 |
| 256x256 | 13,651s | 9.99204e-07 | 37301 |
| 512x512 | 199,024s | 9.99681e-07 | 120361 |
| 1024x1024 | 2254,125s | 9.99989e-06 | 1000000 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU MULTI** | | | |
| **Размер ячейки** | **Время выполнения** | **Результат ошибки** | **Количество итераций** |
| 128x128 | 1.364s | 9.93435e-07 | 11081 |
| 256x256 | 4.419s | 9.99204e-07 | 37301 |
| 512x512 | 26.541s | 9.99681e-07 | 120361 |
| 1024x1024 | 42.748s | 9.99989e-07 | 384341 |

|  |  |  |  |
| --- | --- | --- | --- |
| **GPU openacc** | | | |
| **Размер ячейки** | **Время выполнения** | **Результат ошибки** | **Количество итераций** |
| 128x128 | 4.398s | 9.93435e-07 | 11081 |
| 256x256 | 6.660s | 9.99204e-07 | 37301 |
| 512x512 | 8.634s | 9.99819e-07 | 120361 |
| 1024x1024 | 22.752s | 9.99989e-07 | 364621 |

|  |  |  |  |
| --- | --- | --- | --- |
| **GPU cublas** | | | |
| **Размер ячейки** | **Время выполнения** | **Результат ошибки** | **Количество итераций** |
| 128x128 | 0.937s | 9.7074e-07 | 11110 |
| 256x256 | 1.950s | 9.86088e-07 | 37370 |
| 512x512 | 4.876s | 9.98112e-07 | 120392 |
| 1024x1024 | 15.683s | 9.98906e-07 | 364711 |

**Графики**

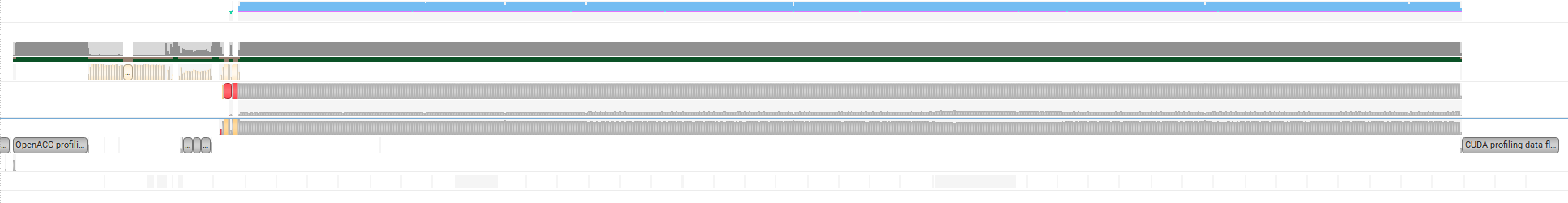
**Этапы оптимизации gpu cublas**

Входные параметры ошибка - 0.000001 сетка - 512 итерации - 1000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Время выполнения** | **Ошибки** | **Количество итераций** | **Комментарий** |
| **1** | 4.102s | 0.0104525 | 1000 | Выполнение программы с прошлого раза |
| **2** | 207.462ms | 0.0102564 | 1000 | Выполнение расчета ошибки через cublas |
| **3** | 17.856ms | 0.0142145 | 1000 | Обновление ошибки каждые 200 итерации |

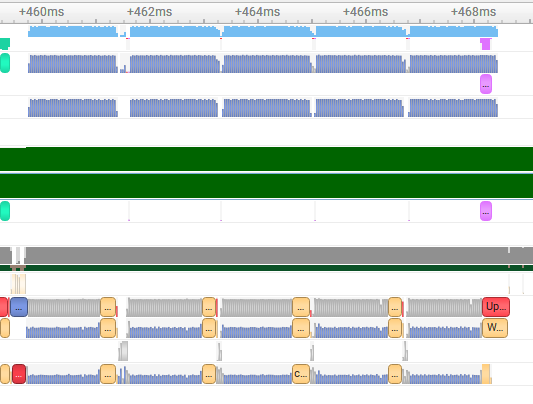
**Результат**

Исходя из полученных результатов видим, что благодаря cublas мы можем сократить сокращение времени выполнения программы. С помощью cublas мы реализуем синхронизацию потоков для вычислений меньше. Так же вычисление ошибки производится быстро. За счет этого наша программа стала работать быстрее.

**Профилировщик  
  
1 код со старого задания  
**

**2 cublas**

****

****