**Отчёт по выполненному заданию**

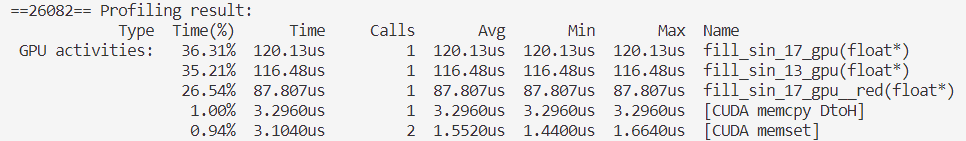
Шитенко Алина гр.21932

С помощью Visual Studio я заполнила на графическом процессе массив типа float/ double и посчитала сумму всех его элементов. При сборке программы для исполнения на GPU использовала компилятор PGI.

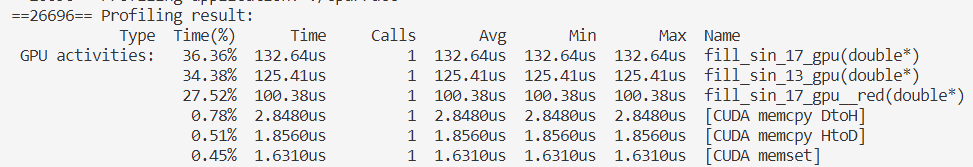


Скриншот Фрагмент результата сборки программы на GPU

Также произвела профилирование программы, с помощью которого узнала, сколько времени тратится на выполнение циклов для заполнения массива и подсчета суммы. Видно, что передача данных занимает меньше времени, чем время исполнения ядер. В программе 2 цикла, но ядер (по выводу профилировщика) три, последнее ядро с суффиксом \_red обозначает, что после функции fill\_sin для окончательного вычисления результата редукции вызывается еще одно ядро.



Скриншот Фрагмент результата профилирования с типом данных float



Скриншот Фрагмент результата профилирования с типом данных double

Я посчитала время, затраченное на CPU и GPU, результаты отобразила в таблице и на диаграмме. Оказалось, что быстрее работают программы с типом float и выполненные на GPU.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CPU-1(одно ядро) | CPU-multicore (несколько ядер) | GPU (общее время работы) |
| Double | 72000 us | 45400 us | 3351 us |
| Float | 50500 us | 40400 us | 3338 us |

*Таблица, в которой указаны различные времена работы программы*

В задании было указано посчитать сумму всех элементов массива, заполненном на графическом процессоре, и сравнить с результатом на центральном. Значения на процессорах не равны, так как на CPU суммирование последовательное, на GPU – параллельное, а параллельное исполнение может менять порядок выполнения операций:

* CPU float: -0.26838809251785278320
* CPU double: 0.00000188488826168203
* GPU float: -0.12359619140625000000
* GPU double: 0.00001157482826123765

|  |
| --- |
| Код программы для типа double (для float просто заменить все упоминания double) |
| CPU |
|  |
| GPU |
|  |

Я считаю, что оптимальнее для моей задачи использовать GPU, так как вычислений много: размер массива – 10^7 элементов, и в отличие от CPU, GPU может поддерживать несколько тысяч потоков, время работы программы значительно меньше.

Ссылка на github: https://github.com/alincnl/Task1Parallelism