

Теория параллелизма

Отчет

“Уравнение теплопроводности на нескольких GPU”

Выполнила: Володина Софья, г. 21932

Дата: 31 мая 2023г

Цели работы:

Программа должна быть реализована на CUDA. Распараллеливание на несколько GPU должно производиться с использованием MPI. Операцию редукции(подсчет максимальной ошибки) в рамках одного MPI процесса реализовать с использованием библиотеки CUB. Подсчет глобального значения ошибки, обмен граничными условиями реализовать с использованием MPI.

Сравнить скорость работы и масштабирование для разных размеров сеток на разном количестве графических процессоров (1, 2, 4). Обратить внимание на корректность отображения MPI процессов на ядра центрального процессора и корректный выбор графических процессоров. Результаты, их анализ и выводы представить в виде отчета.

Используемый компилятор: mpic++

Используемый профилировщик: nsys

Как производили замер времени работы: time при запуске программы с помощью mpiexec, библиотека <chrono>

Данные из предыдущих задач.

CPU-onecore

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	1.49 s	0.000001	30074
256*256	20.3 s	0.000001	102885
512*512	269.4 s	0.000001	339599

CPU-multicore

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	0.35 s	0.000001	30074
256*256	3.57 s	0.000001	102885
512*512	41.6 s	0.000001	339599
1024*1024	483.7 s	0.000001	1000000

GPU

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	0.9 s	9.9998e-07	30080
256*256	2.4 s	9.99984e-07	102912
512*512	19.4 s	9.99984e-07	339968
1024*1024	46.8 s	1.36929e-06	1000000

GPU_cuBLAS

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	1.2 s	9.9998e-07	30074
256*256	3.1 s	9.99984e-07	102885
512*512	12.1 s	9.99984e-07	339599
1024*1024	40.4 s	0.00140936	1000000

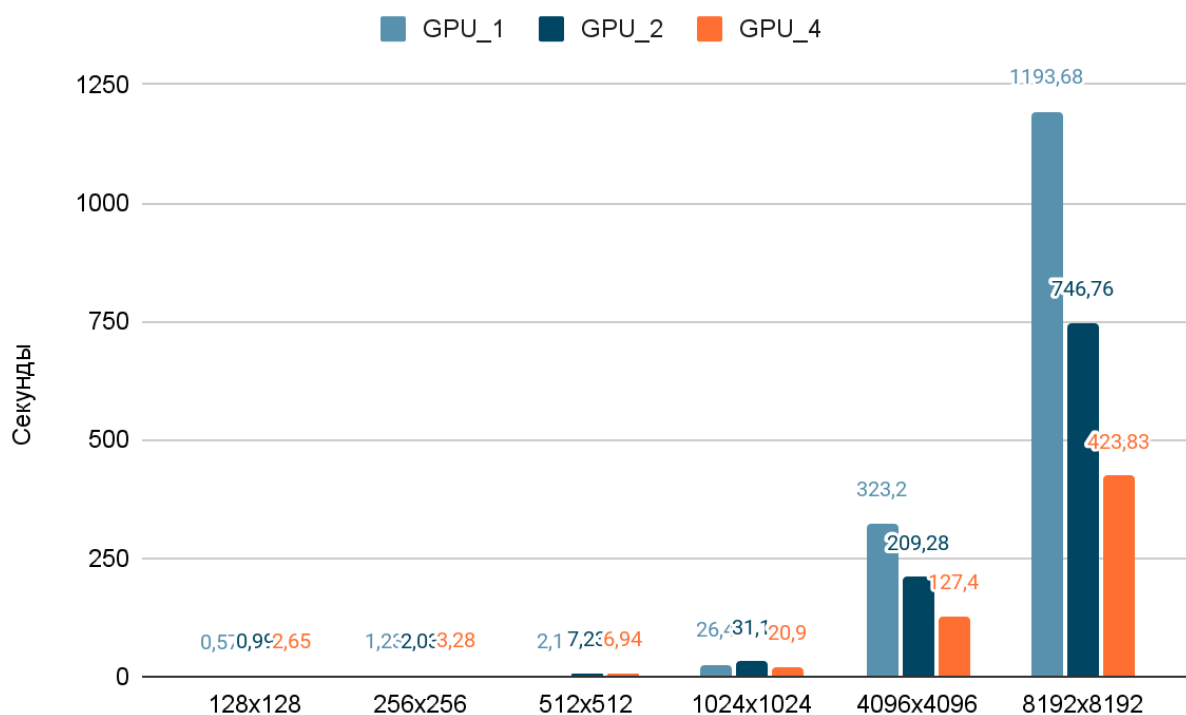
GPU_CUDA

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	0.24 s	9.47552e-07	30500
256*256	0.54 s	9.91241e-07	103000
512*512	2.9 s	9.97135e-07	340000
1024*1024	29.5 s	1.36929e-06	1000000

Данные работы с MPI

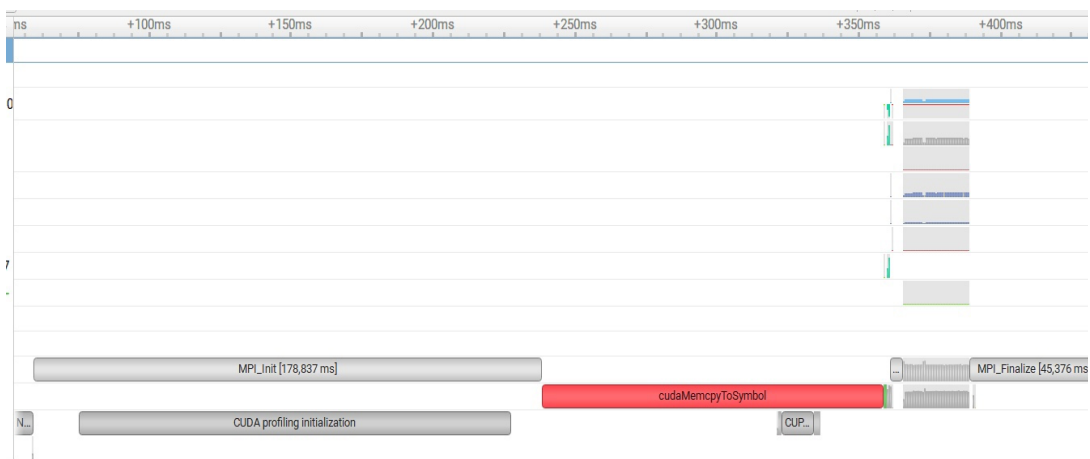
Количество процессов	Размер сетки	Время выполнения	Точность	Количество итераций
1	128x128	0.57 s	8.77769e-07	30500
	256x256	1.23 s	9.91241e-07	103000
	512x512	2.1 s	9.91241e-07	340000
	1024x1024	26.4 s	9.91241e-07	1000000
	4096x4096	323.2 s	9.91241e-07	1000000
	8192x8192	1193.68 s	9.91241e-07	1000000
2	128x128	0.99 s	8.77769e-07	30500
	256x256	2.03 s	9.91241e-07	103000
	512x512	7.23 s	9.91241e-07	340000
	1024x1024	31.1 s	9.91241e-07	1000000
	4096x4096	209.28 s	9.91241e-07	1000000
	8192x8192	746.76 s	9.91241e-07	1000000
4	128x128	2.65 s	8.77769e-07	30500
	256x256	3.28 s	9.91241e-07	103000
	512x512	6.94 s	9.91241e-07	340000
	1024x1024	20.9 s	9.91241e-07	1000000
	4096x4096	127.4 s	9.91241e-07	1000000
	8192x8192	423.83 s	9.91241e-07	1000000

Скорость работы для разных размеров сеток на разном количестве графических процессоров (1, 2, 4).

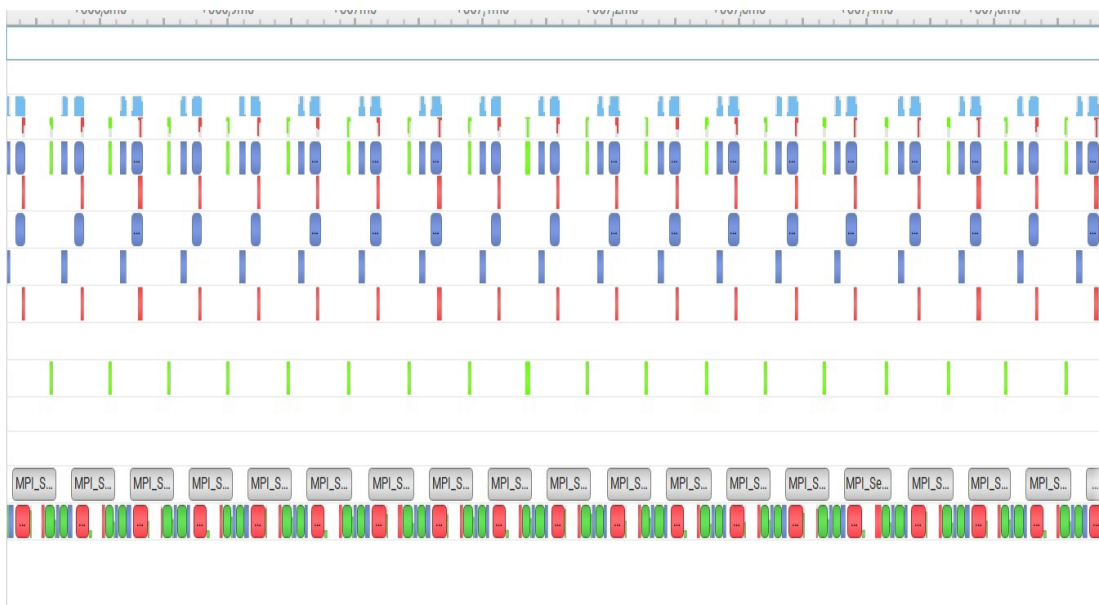


Скриншоты с результатами профилирования в NSightSystems

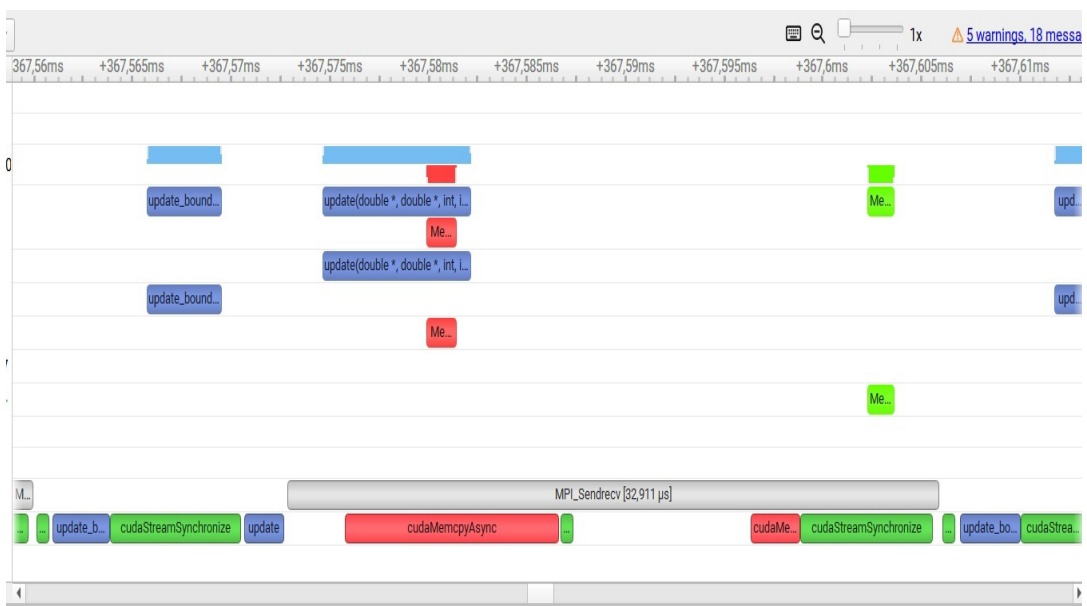
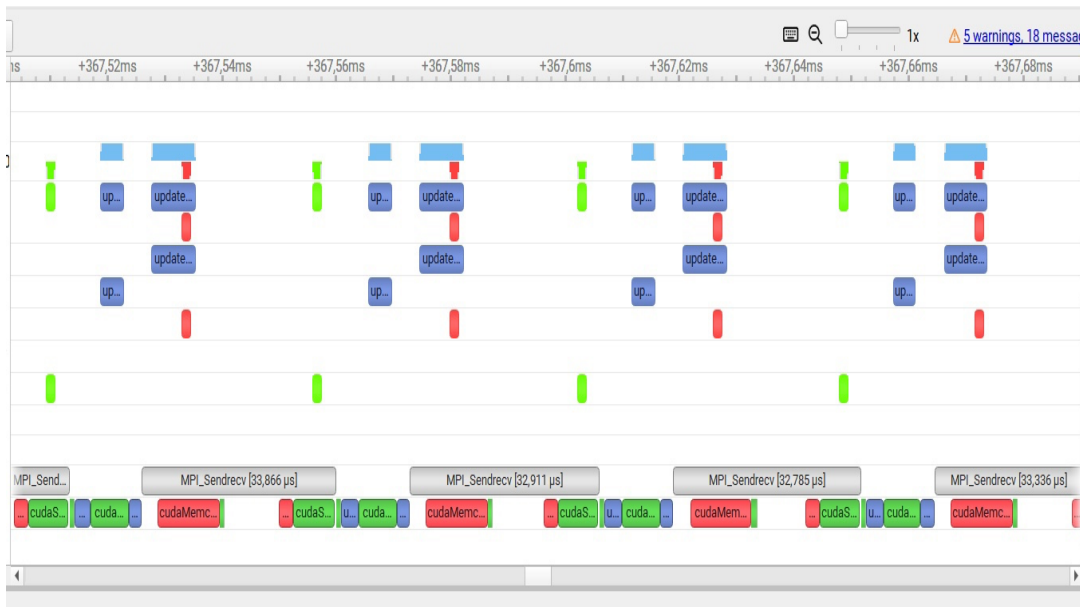
Полная визуализация программы.



Главный цикл программы 372ms.



**Итерации: обновление границ, обновление внутренних значений,
обновление ошибки, обмен границами.**



Вывод решения сетки 11 на 11

```
10 11 12 13 14 15 16 17 18 19 20
11 12 13 14 15 16 17 18 19 20 21
12 13 14 15 16 17 18 19 20 21 22
13 14 15 16 17 18 19 20 21 22 23
14 15 16 17 18 19 20 21 22 23 24
15 16 17 18 19 20 21 22 23 24 25
16 17 18 19 20 21 22 23 24 25 26
17 18 19 20 21 22 23 24 25 26 27
18 19 20 21 22 23 24 25 26 27 28
19 20 21 22 23 24 25 26 27 28 29
20 21 22 23 24 25 26 27 28 29 30
Error: 2.02931e-11
time: 193866 mcs
Iterations: 500
```

Вывод

Было реализовано решение уравнения теплопроводности в двумерной области на равномерных сетках с использованием функций CUDA и библиотеки CUB. Распараллеливание на несколько GPU производилось с использованием MPI. Проведено сравнение скорости работы данной программы на разном количестве процессов. Из данных наблюдений можно сделать вывод, что данный подход решения задачи будет эффективен только при больших размерах матрицы и на большем количестве GPU. Если говорить обратно, при меньших матрицах результат по времени гораздо хуже, чем у предыдущих решений задачи, при этом число процессов не поможет улучшить скорость расчетов.

Приложение

Ссылка на git: https://github.com/PiroJOJO/Parallel_5