

Теория параллелизма

Отчёт

Уравнения теплопроводности на нескольких
GPU

Выполнил Солопов Илья

Группа 21932

26.05.2023

ВВЕДЕНИЕ	3
Цели работы.....	3
Используемый компилятор	3
Используемый профилировщик	3
Замер времени работы.....	3
ВЫПОЛНЕНИЕ НА GPU	4
GPU – вариант без использования MPI.....	4
GPU – вариант с использованием MPI	4
Диаграмма сравнения времени работы.....	5
Диаграмма сравнения времени работы.....	5
ВЫВОД	6
ПРИЛОЖЕНИЕ.....	7
Ссылка на репозиторий	7
Скриншоты из профилировщика.....	7

Введение

Цели работы

Реализовать решение уравнения теплопроводности (пятиточечный шаблон) в двумерной области на равномерных сетках. С условиями линейной интерполяции между углами области, а также ограниченными значениями точности и максимального числа итераций.

На вход программе через командную строку должны подаваться параметры: точность, размер сетки, количество итераций.

Вывод программы – количество итераций и достигнутое значение ошибки.

Перенести программу на GPU используя CUDA и произвести распараллеливание с помощью MPI. Операцию редукции (вычисление максимального значения ошибки) реализовать с использованием библиотеки CUB.

Сравнить скорость работы и масштабирование для разных размеров сеток на разном количестве графических процессоров (1, 2, 4).

Произвести профилирование программы и оптимизацию кода.

Используемый компилятор

Использовался компилятор `mpic++`.

Используемый профилировщик

Использовался профилировщик NVIDIA NsightSystems.

Замер времени работы

Замер времени работы программы производился при помощи библиотеки `chrono` языка `c++`.

Выполнение на GPU

GPU – вариант без использования MPI

Размер сетки	Время выполнения, мс	Точность	Количество итераций
128*128	3355	9.97535e-07	30081
256*256	3518	9.97732e-07	102913
512*512	4464	9.92997e-07	339969
1024*1024	26561	1.373e-06	1000000

GPU – вариант с использованием MPI

Размер сетки	Количество GPU	Время выполнения, мс	Точность	Количество итераций
128*128	1	601	9.05043e-07	30081
	2	1100	9.05043e-07	
	4	2095	9.05043e-07	
256*256	1	1200	9.92452e-07	102913
	2	2010	9.92452e-07	
	4	3341	9.92452e-07	
512*512	1	2466	9.97394e-07	339969
	2	6950	9.97394e-07	
	4	8200	9.97394e-07	
1024*1024	1	26989	1.36929e-06	1000000
	2	30145	1.36929e-06	
	4	22799	1.36929e-06	
2048*2048	1	95610	1.15583e-05	1000000
	2	73202	1.15583e-05	
	4	57141	1.15583e-05	
4096*4096	1	371231	9.82094e-06	1000000
	2	213451	9.82094e-06	
	4	129050	9.82094e-06	
8192*8192	1	1451140	1.03114e-05	1000000
	2	749432	1.03114e-05	
	4	438418	1.03114e-05	

Диаграмма сравнения времени работы на сетках больших размерностей

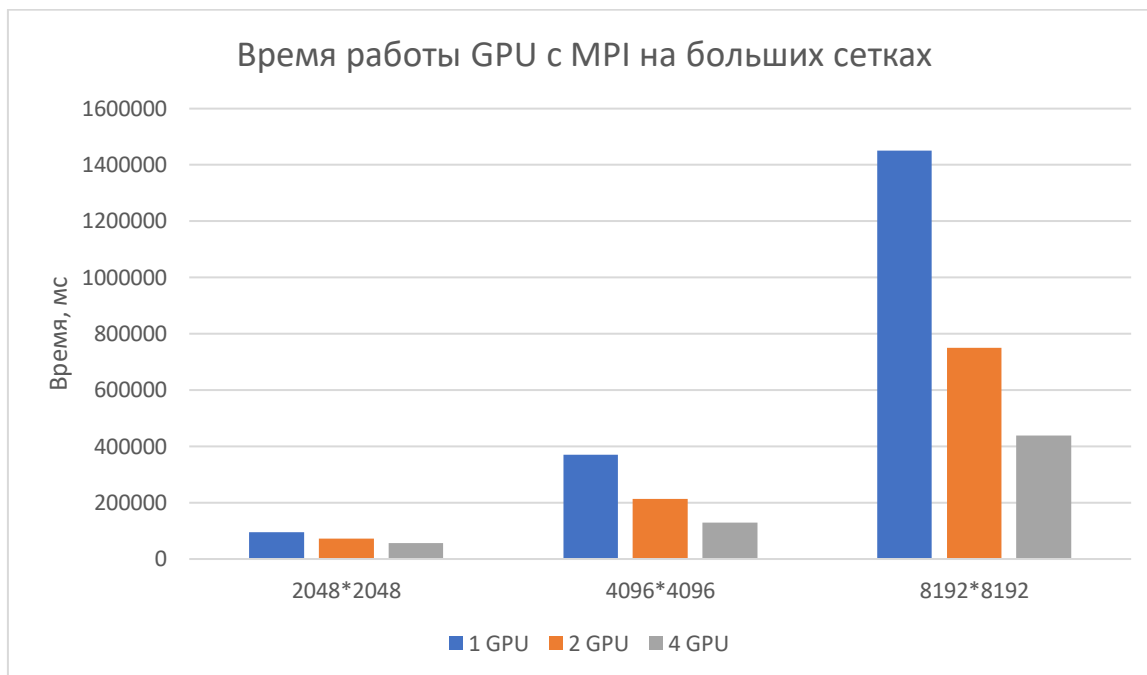
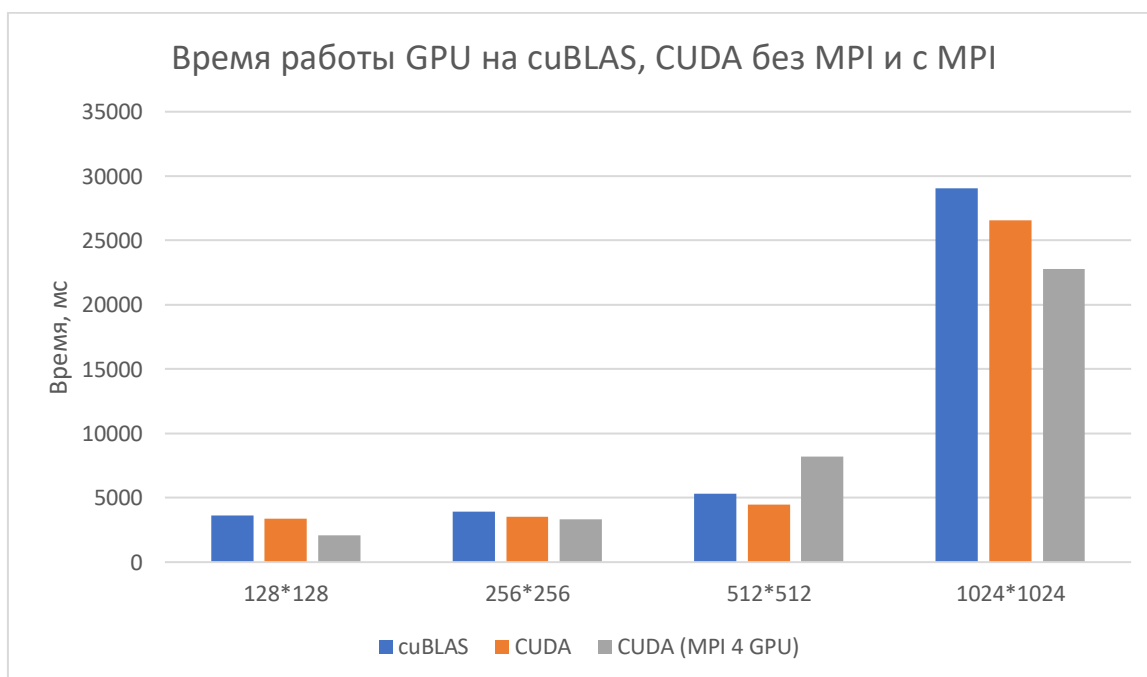


Диаграмма сравнения времени работы



Вывод

Полученные результаты говорят нам о том, что использование MPI для распараллеливания программы на несколько графических процессоров благотворно влияет на производительность программы, но эффект заметен лучше только на больших сетках. Для сеток малого размера использование нецелесообразно.

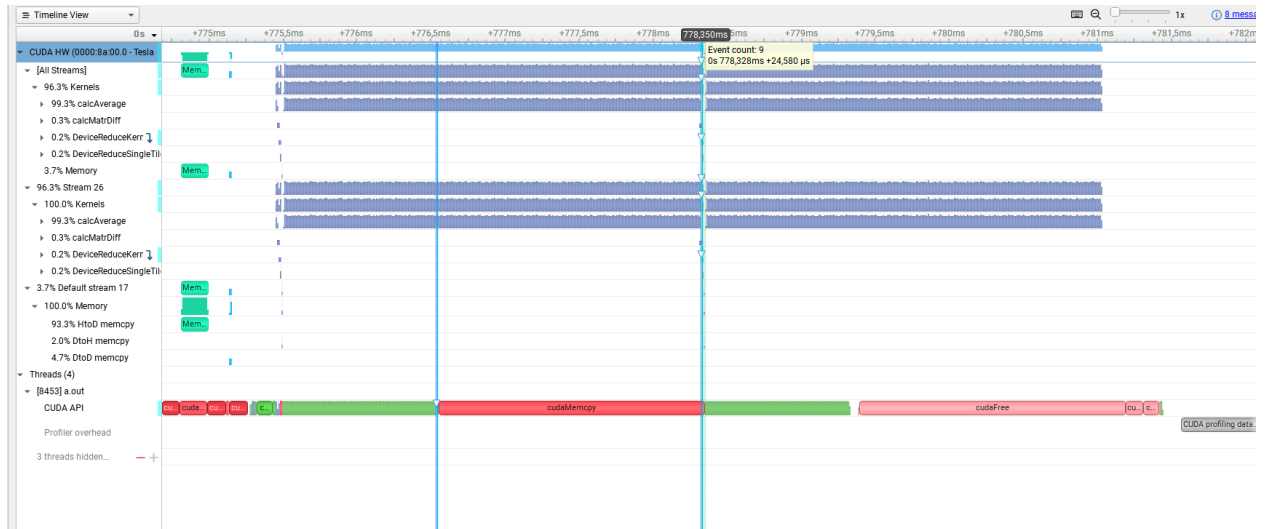
Приложение

Ссылка на репозиторий

<https://github.com/ILSO/Parallelism-tasks>

Скриншоты из профилировщика

Этап 1



Этап 2

Этап 3

Этап 4