

Теория параллелизма

Отчёт

Уравнение теплопроводности на нескольких  
GPU

Выполнил Грищенко Александр Михайлович, 21932

05.2023

## 1 Цели работы

Реализовать решение уравнение теплопроводности (пятиточечный шаблон) в двумерной области на равномерных сетках.

Перенести программу на GPU используя CUDA. Распараллеливание на несколько GPU произвести 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). Проверить корректность отображения MPI процессов на ядра центрального процессора и корректный выбор графических процессоров.

Сравнить скорость работы для разных размеров сеток на графическом процессоре с предыдущими реализациями на OpenACC и OpenACC с cuBLAS.

Произвести профилирование программы и оптимизацию кода.

## 2 Используемые ускорители

Tesla V100-SXM2-32GB

## 3 Используемый компилятор

Open MPI 4.0.7rc2

Для компиляции:

```
mpic++ task_5.cu -o t5-mpi
```

Для запуска:

```
mpiexec -n 2, ./t5-mpi
```

Порядок параметров не важен, как и сами параметры. По умолчанию размер сетки – 128, количество итераций –  $10^6$ , точность –  $10^{-6}$

## 4 Используемый профилировщик

nsys: NVIDIA Nsight Systems version 2022.4.1.21-0db2c85 с флагом *--trace cuda*

```
nsys profile -t cuda mpiexec -n 4  
./t5-mpi -s 2048 -i 4000
```

## 5 Замер времени работы

Для замера времени работы использовалась библиотека *chrono*. Замер времени производился несколько раз, затем бралось среднее время.

## 6 Профилирование





# 7 Сравнение времени работы с предыдущими реализациями

Количество операций во всех реализациях кратно размеру сетки.

## 7.1 CPU-onecore

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.2	9.9e-07	30080
256*256	3.3	9.9e-07	102912
512*512	61	9.9e-07	339968

## 7.2 CPU-multicore

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.9	9.9e-07	30080
256*256	2.6	1e-07	102912
512*512	21	9.9e-07	339968
1024*1024	205	9.89e-07	1000448

## 7.3 GPU (OpenACC)

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.13	9.9e-07	30080
256*256	0.37	9.9e-07	102912
512*512	1.8	9.9e-07	339968
1024*1024	25.7	1.4e-06	1000448

#### 7.4 GPU (OpenACC + cuBLAS)

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.09	9.9e-07	30080
256*256	0.33	9.9e-07	102912
512*512	1.8	9.9e-07	339968
1024*1024	25.8	1.4e-06	1000448

#### 7.5 GPU (CUDA)

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.06	9.9e-07	30080
256*256	0.23	9.9e-07	102912
512*512	1.3	9.9e-07	339968
1024*1024	24.1	1.4e-06	1000448

#### 7.6 MPI

Размер сетки	Количество GPU	Время выполнения, с	Коэфф. эффективности	Точность	Количество операций
128*128	1	0.2	1	9.9e-07	30080
	2	0.9	0.11		
	4	1.5	0.03		
256*256	1	0.75	1	9.9e-07	102912
	2	3.2	0.117		
	4	5.3	0.035		
512*512	1	7.1	1	9.9e-07	339968
	2	12.9	0.27		
	4	18.5	0.096		
1024*1024	1	64.7	1	1.4e-06	1000448
	2	217	0.15		
	4	54.8	0.29		
2048*2048	1	247	1	1.2e-05	1000448
	2	141.9	0.87		
	4	94.7	0.65		
4096*4096	1	1056	1	9.8e-06	1000448
	2	538	0.98		
	4	277	0.95		

## 8 Вывод

MPI позволяет решать задачу одновременно на нескольких GPU, однако это эффективно только на больших сетках. На маленьких же стоит отдать предпочтение решению на одной GPU.

## 9 Приложение

[Ссылка на GitHub](#)