Отчет по Теории Параллелизма

Ссылка на гитхаб: https://github.com/ViktoriaTix/TP
Для создания программы на видеокарте pgcc -Minfo=all -fast -acc main.c -o cparracc -pg
Для создания программы на процессоре gcc -o main main.c -lm

Результат работы

Тип данных	GPU - видеокарта	CPU(One-core) - процессор
double	-0.000000000003211653165635 7	0.000000000048958196545031 2
float	-0.028533935546875000000000	-0.027786219492554664611816 4

Время выполнения цикла на заполнение массива

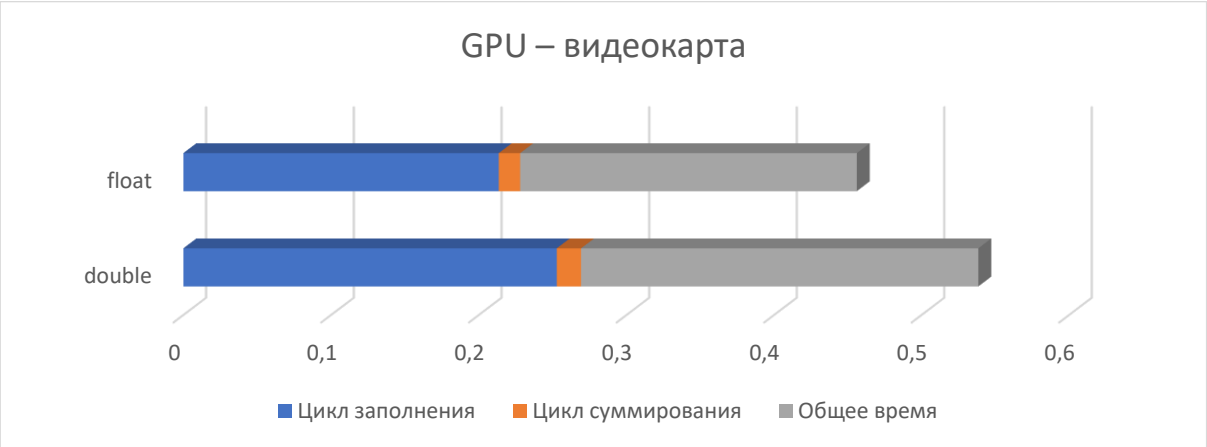
Тип данных	GPU – видеокарта (сек)	CPU(One-core) – процессор (сек)
double	0.252780	0.224502
float	0.213535	0.204939

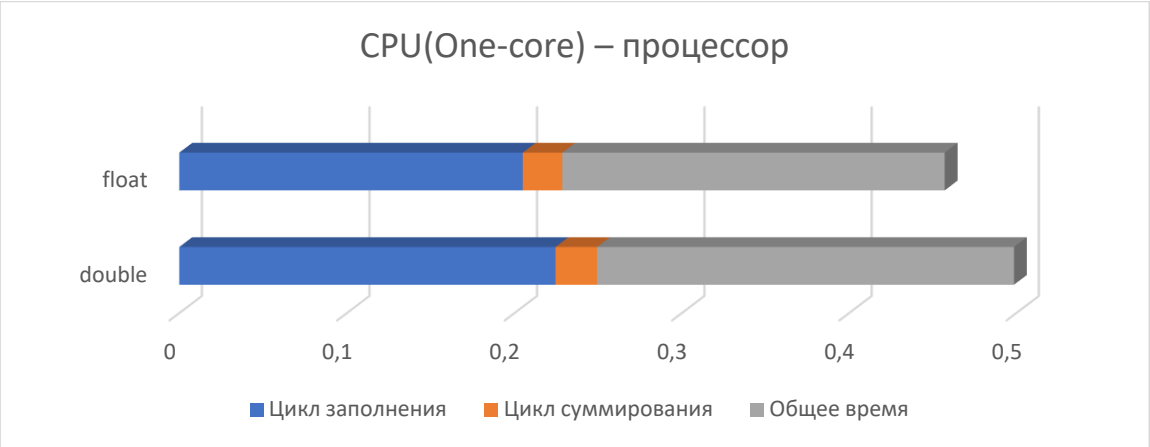
Время выполнения цикла на суммирование

Тип данных	GPU – видеокарта (сек)	CPU(One-core) – процессор (сек)
double	0.016438	0.024754
float	0.014495	0.023618

Общее время выполнения программы

Тип данных	GPU – видеокарта (сек)	CPU(One-core) – процессор (сек)
double	0.269264	0.249355
float	0.228080	0.228642





Код для double:

```
#include <stdio.h>
#include <math.h>
#include <time.h>      // for clock_t, clock(), CLOCKS_PER_SEC

#define _USE_MATH_DEFINES
#define N 10000000
#define M_PI 3.14159265358979323846 // pi

double Sum=0;
double arr[N];

int main()
{
        double time_all = 0.0;
        clock_t begin_all = clock();

        double time_spent1 = 0.0;

        clock_t begin1 = clock();

#pragma acc kernels
        for (int i = 0; i < N; i++)
```

```
	{
		arr[i] = sin(i * M_PI *2 / N);
	}

	clock_t end1 = clock();
	time_spent1 += (double)(end1 - begin1) / CLOCKS_PER_SEC;

	double time_spent2 = 0.0;

	clock_t begin2 = clock();

#pragma acc kernels
	for (int i = 0; i < N; i++)
	{
		Sum += arr[i];
	}

	clock_t end2 = clock();
	time_spent2 += (double)(end2 - begin2) / CLOCKS_PER_SEC;

	printf("%f\n", time_spent1);
	printf("%f\n", time_spent2);
	printf("%0.25f\n", Sum);
	printf("\n");

	clock_t end_all = clock();
	time_all += (double)(end_all - begin_all) / CLOCKS_PER_SEC;
	printf("%f\n", time_all);

	return 0;
}
```

Вывод: программа работает быстрее на GPU – видеокарте. И в каждом случае тип данных float обрабатывается быстрее, чем double.