

Задание 1

Оптимизированное распараллеливание программы, реализующей метод Якоби в 3D пространстве Отчёт

Фролова О.В

1 Постановка задачи

Требуется реализовать максимально оптимизированное распараллеливание программы, которая реализует 3D метод Якоби.

- Менять программу можно (значения выходных массивов должны совпадать)
- Рекомендуется параметризовать тип данных с помощью typedef, чтобы можно было запустить программу как с использованием double, так и с использованием float типов для сравнения производительности.
- В выводе программы также должна содержаться информация о модели ГПУ и о количестве памяти, которое на нем доступно.

2 Makefile

```
NVCC      := nvcc

NVCCFLAGS := -O3 -arch=sm_60 -std=c++11 -Xcompiler -fopenmp

TARGET    := jacob
SRC       := jacob.cu

.PHONY: all float double clean

all: $(TARGET)

$(TARGET): $(SRC)
$(NVCC) $(NVCCFLAGS) $< -o $@

float: NVCCFLAGS += -DUSE_FLOAT
float: clean all
double: clean all

clean:
rm -f $(TARGET)
```

3 Результаты выполнения

3.1 Тип double

Были выбраны значения размеры сетки $L = 900$, количества итераций $i = 20$
Запуск производился на Polus
Результат выполнения команды

```
make
./jacob -L 900 -i 20 --compare
```

```

[edu-cmc-sqi24-14@polus-ib ~]$ ./jacob -L 900 -i 20 --compare
Jacobi3D: 900^3, 20 iterations
Data type: double
GPU: Tesla P100-SXM2-16GB, memory 16280 MB
CPU IT= 1 EPS=1.349500e+03
CPU IT= 2 EPS=5.245833e+02
CPU IT= 3 EPS=3.371435e+02
CPU IT= 4 EPS=2.746667e+02
CPU IT= 5 EPS=2.330247e+02
CPU IT= 6 EPS=1.967731e+02
CPU IT= 7 EPS=1.629500e+02
CPU IT= 8 EPS=1.392891e+02
CPU IT= 9 EPS=1.233136e+02
CPU IT= 10 EPS=1.122117e+02
CPU IT= 11 EPS=1.054653e+02
CPU IT= 12 EPS=9.813325e+01
CPU IT= 13 EPS=9.125502e+01
CPU IT= 14 EPS=8.440327e+01
CPU IT= 15 EPS=7.823711e+01
CPU IT= 16 EPS=7.229691e+01
CPU IT= 17 EPS=6.763436e+01
CPU IT= 18 EPS=6.371415e+01
CPU IT= 19 EPS=6.063202e+01
CPU IT= 20 EPS=5.815683e+01
CPU time = 10.912s
GPU IT= 1 EPS=1.349500e+03
GPU IT= 2 EPS=5.245833e+02
GPU IT= 3 EPS=3.371435e+02
GPU IT= 4 EPS=2.746667e+02
GPU IT= 5 EPS=2.330247e+02
GPU IT= 6 EPS=1.967731e+02
GPU IT= 7 EPS=1.629500e+02
GPU IT= 8 EPS=1.392891e+02
GPU IT= 9 EPS=1.233136e+02
GPU IT= 10 EPS=1.122117e+02
GPU IT= 11 EPS=1.054653e+02
GPU IT= 12 EPS=9.813325e+01
GPU IT= 13 EPS=9.125502e+01
GPU IT= 14 EPS=8.440327e+01
GPU IT= 15 EPS=7.823711e+01
GPU IT= 16 EPS=7.229691e+01
GPU IT= 17 EPS=6.763436e+01
GPU IT= 18 EPS=6.371415e+01
GPU IT= 19 EPS=6.063202e+01
GPU IT= 20 EPS=5.815683e+01
GPU time = 0.952s
Max diff = 0.000000e+00
Verification: SUCCESSFUL
Speedup: 11.46x

```

Результаты показывают значительное ускорение работы на GPU по сравнению с CPU. Максимальная разница (Max diff) между результатами CPU и GPU составила 0.000000e+00, что означает полное совпадение данных. Это подтверждает корректность работы программы (Verification: SUCCESSFUL)

3.2 Тип float

Были выбраны значения размеры сетки $L = 900$, количества итераций $i = 20$

Запуск производился на Polus

Результат выполнения команды

```
make float
./jacob -L 900 -i 20 --compare
```

```
[edu-cmc-sqi24-14@polus-ib ~]$ ./jacob -L 900 -i 20 --compare
Jacobi3D: 900^3, 20 iterations
Data type: float
GPU: Tesla P100-SXM2-16GB, memory 16280 MB
CPU IT= 1 EPS=1.349500e+03
CPU IT= 2 EPS=5.245833e+02
CPU IT= 3 EPS=3.371438e+02
CPU IT= 4 EPS=2.746666e+02
CPU IT= 5 EPS=2.330245e+02
CPU IT= 6 EPS=1.967731e+02
CPU IT= 7 EPS=1.629500e+02
CPU IT= 8 EPS=1.392892e+02
CPU IT= 9 EPS=1.233137e+02
CPU IT= 10 EPS=1.122117e+02
CPU IT= 11 EPS=1.054656e+02
CPU IT= 12 EPS=9.813293e+01
CPU IT= 13 EPS=9.125537e+01
CPU IT= 14 EPS=8.440295e+01
CPU IT= 15 EPS=7.823730e+01
CPU IT= 16 EPS=7.229688e+01
CPU IT= 17 EPS=6.763440e+01
CPU IT= 18 EPS=6.371423e+01
CPU IT= 19 EPS=6.063220e+01
CPU IT= 20 EPS=5.815710e+01
CPU time = 9.880s
GPU IT= 1 EPS=1.349500e+03
GPU IT= 2 EPS=5.245833e+02
GPU IT= 3 EPS=3.371438e+02
GPU IT= 4 EPS=2.746666e+02
GPU IT= 5 EPS=2.330245e+02
GPU IT= 6 EPS=1.967731e+02
GPU IT= 7 EPS=1.629500e+02
GPU IT= 8 EPS=1.392892e+02
GPU IT= 9 EPS=1.233137e+02
GPU IT= 10 EPS=1.122117e+02
GPU IT= 11 EPS=1.054656e+02
GPU IT= 12 EPS=9.813293e+01
GPU IT= 13 EPS=9.125537e+01
GPU IT= 14 EPS=8.440295e+01
GPU IT= 15 EPS=7.823730e+01
GPU IT= 16 EPS=7.229688e+01
GPU IT= 17 EPS=6.763440e+01
GPU IT= 18 EPS=6.371423e+01
GPU IT= 19 EPS=6.063220e+01
GPU IT= 20 EPS=5.815710e+01
GPU time = 0.805s
Max diff = 0.000000e+00
Verification: SUCCESSFUL
Speedup: 12.27x
```

GPU демонстрирует более значительный прирост скорости, что подтверждает его эффективность для задач с пониженной точностью. Результаты остались корректными

3.3 Вывод

Использование GPU позволило добиться 11-кратного ускорения вычислений для типа `double` и 12-кратного ускорения для типа `float` без потери точности, что демонстрирует эффективность применения графических процессоров для задач подобного типа (например, решения уравнений методом Якоби).