

## Задание 2

# Оптимальное ускорение программы, реализующей метод ADI в 3D пространстве

## Отчёт

Фролова О.В

### 1 Постановка задачи

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

- 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    := adi3d
SRC       := adi3d.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  $i = 10$   
Запуск производился на Polus  
Результат выполнения команды

```
make
./adi3d -L 900 -i 10 --compare
```

```

[edu-cmc-sqi24-14@polus-ib ~]$ ./adi3d -L 900 -i 10 --compare
ADI3D: 900^3, 10 iterations
Data type: double
GPU: Tesla P100-SXM2-16GB, free memory 16017 MiB / 16280 MiB
CPU IT= 1 EPS=1.498331e+01
CPU IT= 2 EPS=1.498331e+01
CPU IT= 3 EPS=1.870133e+01
CPU IT= 4 EPS=1.870133e+01
CPU IT= 5 EPS=2.054088e+01
CPU IT= 6 EPS=2.054088e+01
CPU IT= 7 EPS=2.167565e+01
CPU IT= 8 EPS=2.167565e+01
CPU IT= 9 EPS=2.248184e+01
CPU IT= 10 EPS=2.248184e+01
CPU time = 5.279s
GPU IT= 1 EPS=1.498331e+01
GPU IT= 2 EPS=1.498331e+01
GPU IT= 3 EPS=1.870133e+01
GPU IT= 4 EPS=1.870133e+01
GPU IT= 5 EPS=2.054088e+01
GPU IT= 6 EPS=2.054088e+01
GPU IT= 7 EPS=2.167565e+01
GPU IT= 8 EPS=2.167565e+01
GPU IT= 9 EPS=2.248184e+01
GPU IT= 10 EPS=2.248184e+01
GPU time = 1.222s
Max diff = 0.000000e+00
Verification: SUCCESSFUL
Speedup: 4.32x

```

GPU (Tesla P100-SXM2-16GB) показал 4.32-кратное ускорение по сравнению с CPU. Результаты CPU и GPU полностью совпадают ( $\text{maxdiff} = 0$ ), что подтверждает отсутствие ошибок в реализации (Verification: SUCCESSFUL)

### 3.2 Тип float

Были выбраны значения размеры сетки  $L = 900$ , количества итераций  $i = 10$

Запуск производился на Polus

Результат выполнения команды

```

make float
./adi3d -L 900 -i 10 --compare

```

```

[edu-cmc-sqi24-14@polus-ib ~]$ ./adi3d -L 900 -i 10 --compare
ADI3D: 900^3, 10 iterations
Data type: float
GPU: Tesla P100-SXM2-16GB, free memory 16017 MiB / 16280 MiB
CPU IT= 1 EPS=1.498331e+01
CPU IT= 2 EPS=1.498331e+01
CPU IT= 3 EPS=1.870134e+01
CPU IT= 4 EPS=1.870134e+01
CPU IT= 5 EPS=2.054088e+01
CPU IT= 6 EPS=2.054088e+01
CPU IT= 7 EPS=2.167565e+01
CPU IT= 8 EPS=2.167565e+01
CPU IT= 9 EPS=2.248183e+01
CPU IT= 10 EPS=2.248184e+01
CPU time = 2.856s
GPU IT= 1 EPS=1.498331e+01
GPU IT= 2 EPS=1.498331e+01
GPU IT= 3 EPS=1.870134e+01
GPU IT= 4 EPS=1.870134e+01
GPU IT= 5 EPS=2.054088e+01
GPU IT= 6 EPS=2.054088e+01
GPU IT= 7 EPS=2.167565e+01
GPU IT= 8 EPS=2.167565e+01
GPU IT= 9 EPS=2.248183e+01
GPU IT= 10 EPS=2.248184e+01
GPU time = 1.079s
Max diff = 0.000000e+00
Verification: SUCCESSFUL
Speedup: 2.65x

```

GPU демонстрирует прирост скорости, что подтверждает его эффективность для задач с пониженной точностью. Результаты CPU и GPU идентичны ( $\text{maxdiff} = 0$ ), что подтверждает отсутствие ошибок.

### 3.3 Вывод

Использование GPU позволило добиться ускорения вычислений без потери точности. Он эффективен для задач с большим объемом вычислений (много итераций/крупная сетка).