

# **Большое домашнее задание 3**

## **Отчет**

Чубий Савва Андреевич, БПИ233

## Изменения кодовой базы

### Параллелизм

- Написан `ThreadPool` (см. файлы `ThreadPool.hpp` и `ThreadPool.cpp`)
- Распараллелены все обходы матрицы.

Это сделано в два этапа:

- Написан метод `void Fluid::forall(const F& f)` (см. файл `Fluid.hpp`), который принимает функцию вида `void f(size_t x, size_t y)` и выполняет её в нескольких потоках для каждой клетки вида  $\{(x, y) \in \mathbb{N}^2 \mid 0 \leq x < n \wedge 0 \leq y < m\}$ .

Причем, обход матрицы сделан в таком порядке, что, если функция `f` модифицирует только клетку  $(x, y)$  и её соседей  $(x \pm 1, y \pm 1)$ , то два потока никогда не будут одновременно оперировать над одной и той же клеткой, а значит дополнительные средства синхронизации (такие как `std::mutex` или `std::atomic`) не требуются, что дополнительно ускоряет алгоритм.

- Метод `forall` использован в соответствующих местах программы

### Не-мультипоточные оптимизации

- Использована статическая реализация Матрицы (из ДЗ 2)
- Реализованы «правильные» сравнения вещественных чисел, а именно, сравнения вида `lhs == rhs` заменены на `boolean::eq(lhs, rhs)`, где функция `eq` объявлена следующим образом:

```
namespace boolean {  
    template<typename T, typename U>  
    bool eq(T lhs, U rhs) {  
        return is_zero(lhs - rhs);  
    }  
  
    template<typename T>  
    bool is_zero(T v) {  
        return std::abs(v) < eps;  
    }  
}
```

- Ускорен поиск по массиву `deltas`. Так как значения массива `deltas` заранее известны, и их количество крайне ограничено, то линейный поиск был заменен на несколько `if`-ов.

### Другое (не оптимизации)

- Добавлен аргумент командной строки `--threads=N`

## Замеры

№ конфиг.	Кол-во поток	Наличие не-мультипоточных оптимизаций	Коммит	Время (секунды)	Ускорение <sup>1</sup>
1	1	Нет	4e93e5d	31.4	1.000
2	8	Нет	4e93e5d	13.2	2.379
3	1	Есть	b97f64e	15.8	1.987
4	8	Есть	b97f64e	10.5	2.990

**Вывод:** как видно из замеров, и параллелизм, и не-мультипоточные оптимизации дают значительное ускорение.

## — Параметры —

### Параметры устройства

CPU:	Intel(R) Pentium(R) Gold G5400 CPU @ 3.70GHz
RAM:	8 GB

### Тест

- Флаги компиляции:
  - DSIZES=S(36, 84),S(36, 85)
  - DTYPES=FIXED(32, 16)
  - g0
  - O2
  - DNDEBUG
  - std=gnu++20
- Аргументы командной строки:
  - p-type=FIXED(32, 16)
  - v-type=FIXED(32, 16)
  - v-flow-type=FIXED(32, 16)
  - ticks=400
  - threads=\*число потоков\*
  - quiet=true
  - data\_heavy.in
- Используется тест из условия:

---

<sup>1</sup>Относительно первого замера

```

// N, M
36 84
// Field
#####
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#.....#                #                .....#
#.....#                #                .....#
#.....#                #                .....#
#.....#                #                .....#
#.....#                #                .....#
#.....#                #                .....#
#.....#                #                .....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#.....#.....#.....#
#####
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#                                                                 #
#####
// G
0.1
// Rho
0.1
. 1000

```