

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №3**  
**по курсу «Параллельная обработка данных»**

**Технология MPI и технология OpenMP**

Выполнил: А. О. Тояков

Группа: М8О-407Б-18

Преподаватели: К. Г. Крашенинников,  
А. Ю. Морозов

Москва, 2022

## УСЛОВИЕ

**Цель работы:** Совместное использование технологии MPI и технологии OpenMP. Реализация метода Якоби. Решение задачи Дирихле для уравнения Лапласа в трехмерной области с граничными условиями первого рода.

Вариант № 1: Распараллеливание основных циклов через parallel for (+директива reduction для вычисления погрешности);

## ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Device: GeForce MX250

Размер глобальной памяти: 3150381056

Размер константной памяти : 65536

Размер разделяемой памяти: 49152

Регистров на блок: 32768

Максимум потоков на блок: 1024

Количество мультипроцессоров : 3

OS: Linux Ubuntu 18.04

Редактор: VSCode

Компилятор: nvcc версии 11.4 (g++ версии 7.5.0)

## МЕТОД РЕШЕНИЯ

Задача решается аналогично предыдущей, однако для переноса граничных значений между процессами мы будем использовать не буфер, а производные типы данных, которые определяют расположения границ. Основные вычисления будем распараллеливать с помощью OpenMP.

## ОПИСАНИЕ ПРОГРАММЫ

#define \_i(i, j, k) – переход из трёхмерной сетки в одномерную для элементов.

#define \_ib(i, j, k) – переход из трёхмерной сетки в одномерную для блоков.

MPI\_Bcast() – передача данных всем процессам.

MPI\_Sendrecv() – передача данных между процессами.

MPI\_Allreduce() – контроль сходимости всех процессов.

MPI\_Type\_create\_hvector() – создание производного типа данных.

MPI\_Type\_commit() – регистрация производного типа данных.

class Processor – класс для хранения всех функций и переменных для решения задачи.

int main() – создание экземпляра класса Processor, а также инициализация и завершение работы MPI.

## ТЕСТЫ ПРОИЗВОДИТЕЛЬНОСТИ

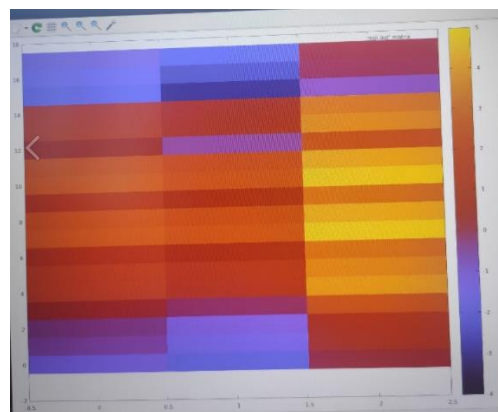
блоков\процессов	1*1*1	2*2*1	2*2*2
4*4*4	0.85ms	3200.4ms	4105.61ms
10*10*10	50.3ms	18116.6ms	23992ms
20*20*20	814.7ms	51562ms	70187.3ms
30*30*30	5564.2ms	128801ms	168309ms

## ВИЗУАЛИЗАЦИЯ

Тест:

```
test.txt
1  1 1 1
2  3 3 6
3  mpi.out
4  1e-10
5  1.0 1.0 2.0
6  -7.0 -10.0 5.0 10.0 -3.0 0.0
7  7.0
```

Срез:



## **ВЫВОДЫ**

Выполнив лабораторную работу № 9, я познакомился с технологией OpenMP, которая является очень простым инструментом для распараллеливания вычислений. Производные типы данных в MPI – мощный и удобный инструмент. Он позволяет процессам не использовать доп. память для пересылки и получения сообщений, а также с помощью него можно распараллеливать запись в файл.