

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №1
по курсу «Параллельная обработка данных»**

Message Passing Interface (MPI)

Выполнил: Г.Н. Хренов

Группа: 8О-407Б

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

Москва, 2021

Условие

1. Цель работы: знакомство с технологией MPI. Реализация метода Якоби. Решение задачи Дирихле для уравнения Лапласа в трехмерной области с граничными условиями первого рода.
Вариант 6. обмен граничными слоями через bsend, контроль сходимости allreduce;

Программное и аппаратное обеспечение

GPU name: NVIDIA GeForce RTX 2060

compute capability 7:5

totalGlobalMem: 6442450944

sharedMemPerBlock: 49152

totalConstMem: 65536

regsPerBlock: 65536

maxThreadsDim: 1024 1024 64

maxGridSize: 2147483647 65535 65535

multiProcessorCount: 30

CPU name: AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx

MaxClockSpeed: 2300

NumberOfCourse: 4

RAM: 8

SSD: 256, HDD: 1024

OS: Windows10

Compiler: nvcc

Метод решения

Регулярная сетка делится на области, за каждую из которых отвечает отдельный процесс. Одна итерация решения исходной задачи состоит из трех этапов. На первом этапе происходит обмен граничными слоями между процессами. На втором этапе выполняется обновление значений во всех ячейках. И третий этап заключается в вычислении погрешности: сначала локально в рамках каждого процесса, а потом через обмены и во всей области.

Описание программы

Lab7.cpp:

#define _i(i, j, k) – переход из трехмерной сетки в линейную для элементов

#define _ib(i, j, k) - переход из трехмерной сетки в линейную для блоков

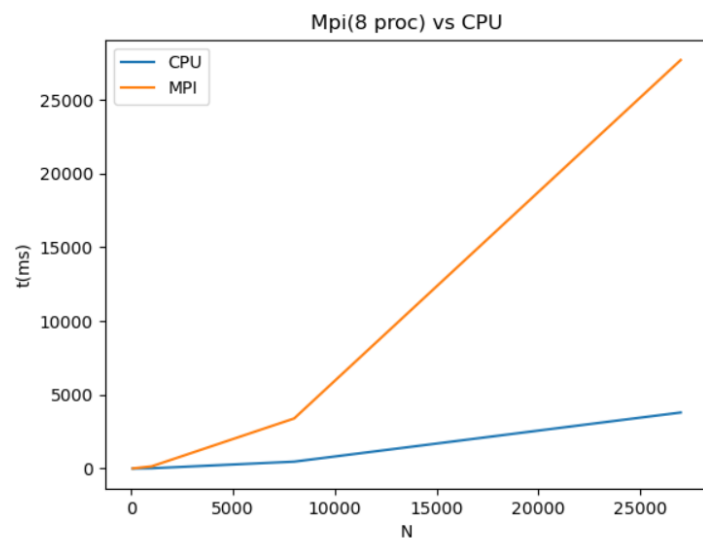
MPI_Bcast(&nx, 1, MPI_INT, 0, MPI_COMM_WORLD)-передача переменных всем процессам

MPI_Bsend(...) – передача данных между процессами

MPI_Allreduce(...) – контроль точности для всех процессов

Результаты

блоков\процессов	1*1*1	2*2*1	2*2*2
4*4*4	0.29ms	3.2ms	8.7ms
10*10*10	16.8ms	86.5ms	148.9ms
20*20*20	459ms	2115.1ms	3394.5ms
30*30*30	3566.2ms	15230.8ms	27.7s



Визуализация

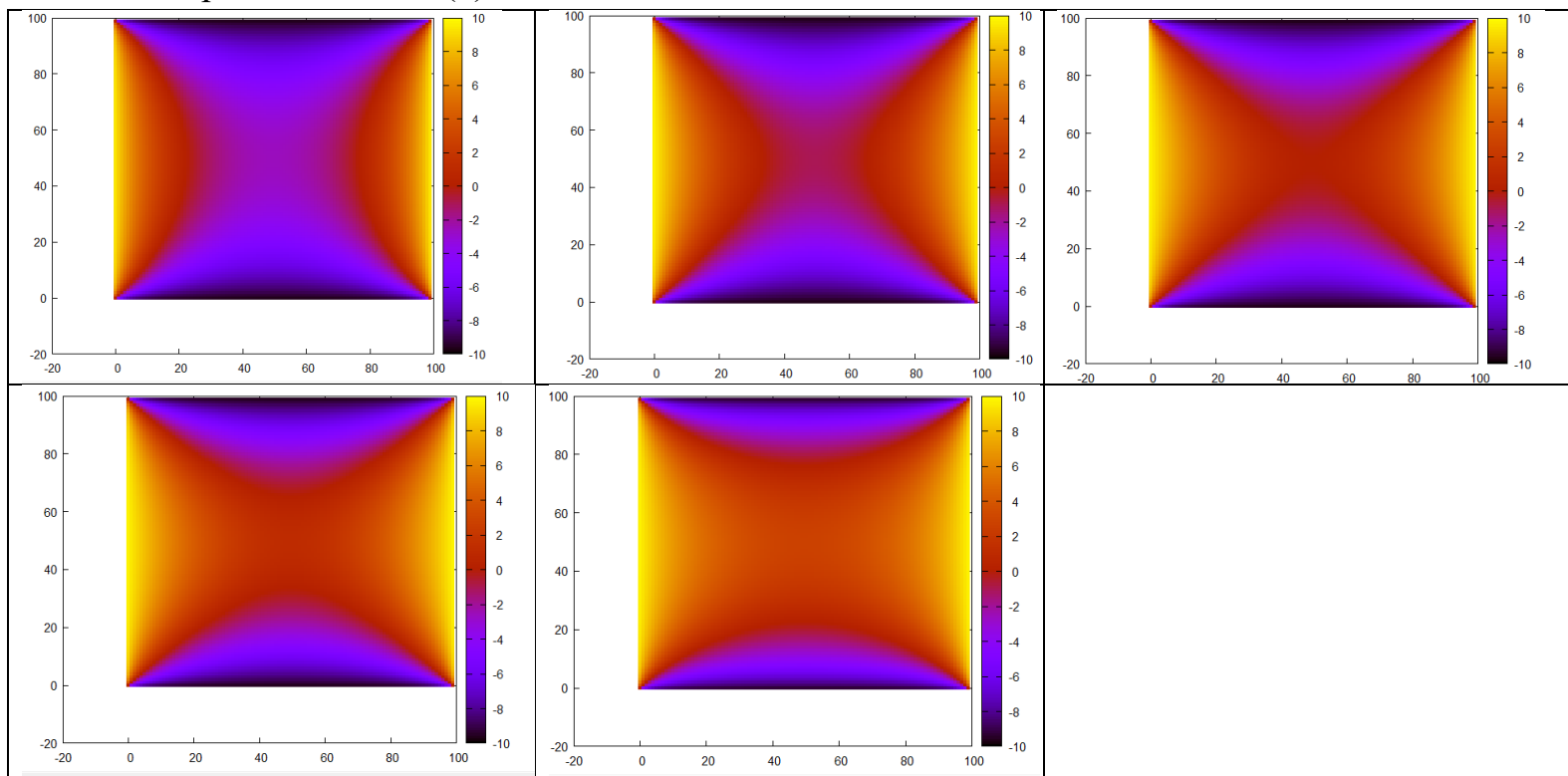
```
2 2 2
100 100 5
mpi.out
1e-10
1.0 1.0 1.0
-5 5 10.0 10.0 -10.0 -10.0
```

Внизу-холодно, вверху-жарко,

лево-право-жарко

зад-перед-холодно

Срезы по высоте(к)



Выводы

Технология MPI дает возможность выполнять параллельные вычисления за счет обеспечения параллельной работы нескольких процессов. Однако общение процессов требует написания большого объема кода, к тому же обслуживание всех процессов занимает немало времени. Bsend хорош тем, что он является неблокирующей операцией. Это достигается благодаря выделению буфера, достаточно большого для того, чтобы при отправке одним процессом всей необходимой информации память буфера не была переиспользована.