МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №1
по курсу «Параллельная обработка данных»

Message Passing Interface (MPI)

Выполнил: К.М. Воронов
Группа: 8О-407Б
Преподаватель: А.Ю. Морозов

Москва, 2022

Условие

Цель работы. Знакомство с технологией MPI. Реализация метода Якоби.
Решение задачи Дирихле для уравнения Лапласа в трехмерной области с граничными условиями первого рода.

Программное и аппаратное обеспечение

GPU:

- Название NVIDIA GeForce RTX 3050
- Compute capability: 8.6
- Графическая память: 4100456448
- Разделяемая память: 49152
- Константная память: 65536
- Количество регистров на блок: 65536
- Максимальное количество нитей: (1024, 1024, 64)
- Максимальное количество блоков: (2147483647, 65535, 65535)
- Количество мультипроцессоров: 16

Сведения о системе:

- Процессор: Intel Core i5-11400H 2.70GHz
- ОЗУ: 32 ГБ
- SSD 1ТБ

Программное обеспечение:

- OS: Linux Mint 21
- Текстовый редактор: Sublime text
- Компилятор: mpic++

Метод решения

Для решения этой задачи я разбиваю исходную сетку на блоки, каждый блок обрабатывается отдельным процессом. После завершения, все блоки собираются в 0 процессе и выводятся.

Описание программы

Взаимодействие процессов происходит через MPI_Send / MPI_Recv, отсылка основных и начальных параметров от нулевого процесса всем остальным через MPI_Bcast, а обработка ошибки через MPI_Allreduce. Так как MPI_Send и MPI_Recv блокируются при отправке/приёме больших данных, то сначала каждый процесс отправляет вверх, направо, назад, а принимает снизу, слева, впереди, а потом наоборот.

Результаты работы

Размер сетки 10x10x10

Количество процессоров	Время, мс
1	8.957 ms
2	7.298 ms
4	5.387 ms
8	6.508 ms

Размер сетки 30x30x30

Количество процессоров	Время, мс
1	1.83798e+03 ms
2	932.837 ms
4	509.457 ms
8	525.460 ms

Размер сетки 50x50x50

Количество процессоров	Время, мс
1	2.01878e+04 ms
2	1.05403e+04 ms
4	5.69327e+03 ms
8	6.07056e+03 ms

Выводы

Выполнив данную лабораторную, я научился параллелить программы с помощью MPI на примере решения дифференциального уравнения. Во время выполнения столкнулся с проблемой блокировки. В коде была ошибочно поставлена синхронизация между отправкой и получением для первого этапа, поэтому при больших размерах сетки программа зависала. Сначала тестировал на 30x30x30 и 50x50x50, но блокировки не происходило. Она была на сетке 100x100x100. Несмотря на сложность написания и тестирования, MPI преподносит себя как очень мощный аппарат.