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»**

**Изучение технологии MPI, CUDA и OpenMP**

Выполнил: Гамов П.А.

Группа: 8О-407Б-18

Преподаватели: К.Г. Крашенинников,

А.Ю. Морозов

Москва, 2021

**Условие**

Знакомство с технологией MPI. Реализация метода Якоби.

Решение задачи Дирихле для уравнения Лапласа в трехмерной области с граничными

условиями первого рода.

Вариант 2 - обмен граничными слоями через bsend, контроль сходимости allgather;

**Программное и аппаратное обеспечение**

nvcc 7.0

Ubuntu 14.04 LTS

|  |  |
| --- | --- |
| Compute capability | 6.1 |
| Name | GeForce GTX 1050 |
| Total Global Memory | 2096103424 |
| Shared Mem per block | 49152 |
| Registers per block | 65534 |
| Max thread per block | (1024,1024,64) |
| Max block | (2147483647, 65535, 65535) |
| Total constant memory | 65536 |
| Multiprocessor’s count | 5 |

**Метод решения**

Использовал технологию MPI для организации пересылки значений между процессорами.

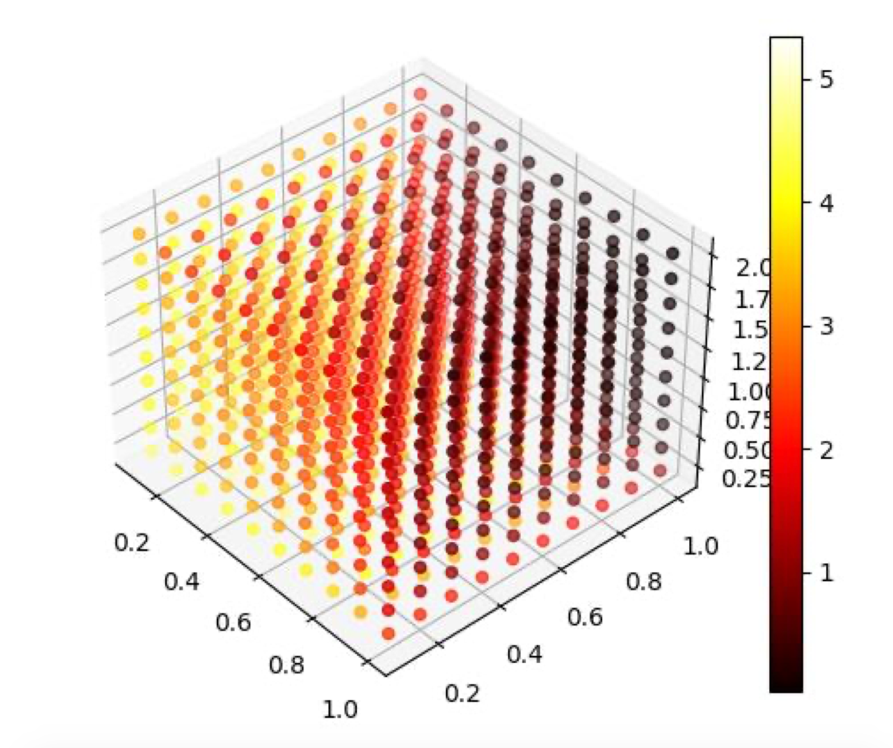
**Описание программы**

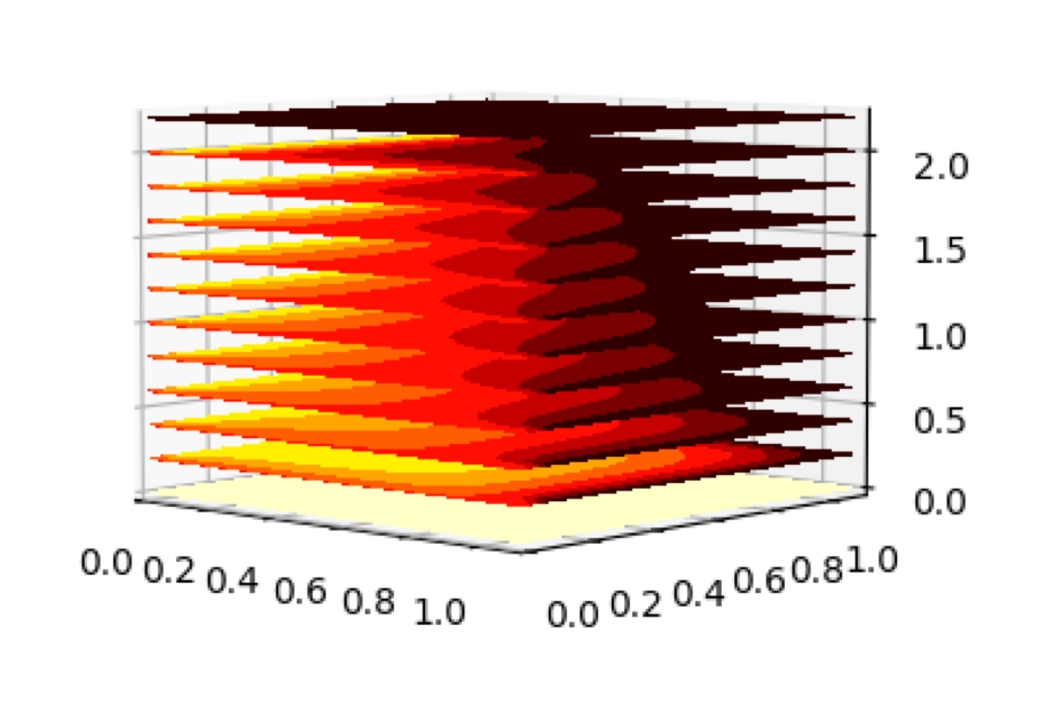
Сначала использовал MPI\_Bcast для передачи всех нужных стартовых параметров всем процессорам. Далее узнав ранг каждого процессора в сетке, организую буффер для функции Bsend используя MPI\_Buffer\_attach.

Заполняю начальными значениями и краевыми. Далее огранизую главный цикл, в котором будет осуществляться основная работа по пересылке данных. Так как функция пересылки B\_send является не блокирующей, я сразу могу отправить все нужные данные и потом уже заняться их приемом. После, использую MPI\_Barrier для того чтобы быть уверенным в том, что все процессоры завершили пересылку данных и в цикле начинаю считать значение новой сетки используя пересланные данные. Ранее был организован массив, в который функцией MPI\_Allgather все посчитанные разницы оказываются в массиве общих погрешностей и каждый процессор проверяет нужно ли ему выходить из цикла.

**Результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| Сетка процессоров | Размер сетки | MPI | Single |
| 2 1 2 | 20 20 20 | 1.3 sec | 1.6 sec |
| 2 2 2 | 40 40 40 | 34.2 sec | 49 sec |

****

****

**Выводы**

Я научился использовать примитивные функции передачи данных для написания довольно сложного алгоритма с технологией MPI, я считаю, что данная технология должна быть использована на действительно сложных проектах и суперкомпьютерах, так как решение уравнения не стоит того времени и сложности с отладкой, с которой придется столкнуться студенту. Очень трудно отлаживать и улавливать ошибки, но технология действительно уникальная и очень интересная.