

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

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

**Лабораторная работа №2
по курсу «Параллельная обработка данных»**

Технология MPI и технология CUDA. MPI-IO

Выполнил: М.А.Трофимов

Группа: 8О-408Б

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

Москва, 2021

Условие

Цель работы. Совместное использование технологии MPI и технологии CUDA. Применение библиотеки алгоритмов для параллельных расчетов Thrust. Реализация метода Якоби. Решение задачи Дирихле для уравнения Лапласа в трехмерной области с граничными условиями первого рода. Использование механизмов MPI-IO и производных типов данных. **Вариант:** 1. конструктор типа MPI_Type_create_subarray.

Программное и аппаратное обеспечение

Характеристики GPU "NVIDIA GeForce GTX 950"

CUDA Driver Version / Runtime Version 11.4 / 11.4
CUDA Capability Major/Minor version number: 5.2
Total amount of global memory: 1997 MBytes (2094137344 bytes)
(006) Multiprocessors, (128) CUDA Cores/MP: 768 CUDA Cores
GPU Max Clock rate: 1278 MHz (1.28 GHz)
Memory Clock rate: 3305 Mhz
Memory Bus Width: 128-bit
L2 Cache Size: 1048576 bytes
Maximum Texture Dimension Size (x,y,z) 1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
Maximum Layered 1D Texture Size, (num) layers 1D=(16384), 2048 layers
Maximum Layered 2D Texture Size, (num) layers 2D=(16384, 16384), 2048 layers
Total amount of constant memory: 65536 bytes
Total amount of shared memory per block: 49152 bytes
Total shared memory per multiprocessor: 98304 bytes
Total number of registers available per block: 65536
Warp size: 32
Maximum number of threads per multiprocessor: 2048
Maximum number of threads per block: 1024
Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
Max dimension size of a grid size (x,y,z): (2147483647, 65535, 65535)
Maximum memory pitch: 2147483647 bytes
Texture alignment: 512 bytes

Характеристики CPU Intel i5-4460

of Cores 4
of Threads 4
Processor Base Frequency 3.20 GHz
Max Turbo Frequency 3.40 GHz
Cache 6 MB Intel® Smart Cache
Bus Speed 5 GT/s

Intel® Turbo Boost Technology 2.0 Frequency 3.40 GHz
TDP 84 W

Характеристики RAM

Total 15 Gi

Swap 2 Gi

Операционная система: Ubuntu 20.04 LTE

IDE Sublime Text 3

Compiler nvcc for cuda 11.4

Метод решения

На каждый процесс приходится один блок 3хмерной сетки. Блок тоже является 3хмерной сеткой. Процесс итерационно решает уравнение. В конце каждой итерации каждый процесс обменивается с “соседними” процессами граничными точками. Обмен происходит неблокирующими isend и igrsv, а проверка сходимости определяется с помощью allreduce. Максимальный элемент разностей двух итераций вычисляется с помощью thrust::max_element, а итерация решателя также распараллеливается на GPU. Печать ответа в файл производится параллельно всеми процессами.

Описание программы

В программе всё вынесено в разные функции:

read_Args - считать аргументы из stdin.

bcast_Args - разослать аргументы всем остальным процессам.

init - инициализировать массив точек блока.

swap - обмен местами двух указателей

print_data - печать данных в файл arg.name

do_math - одна итерация решателя на GPU.

{init/free}_buff - инициализация и очистка буферров, для общения между процессами.

{get/set}_{XOY/XOZ/YOZ}_slice - взятие “среза” куба области параллельно соответствующей плоскости.

pre_proc_max - создание массива разностей значений между итерациями и зануление границ, чтобы thrust::max_element нашёл именно максимальный элемент внутри сетки.

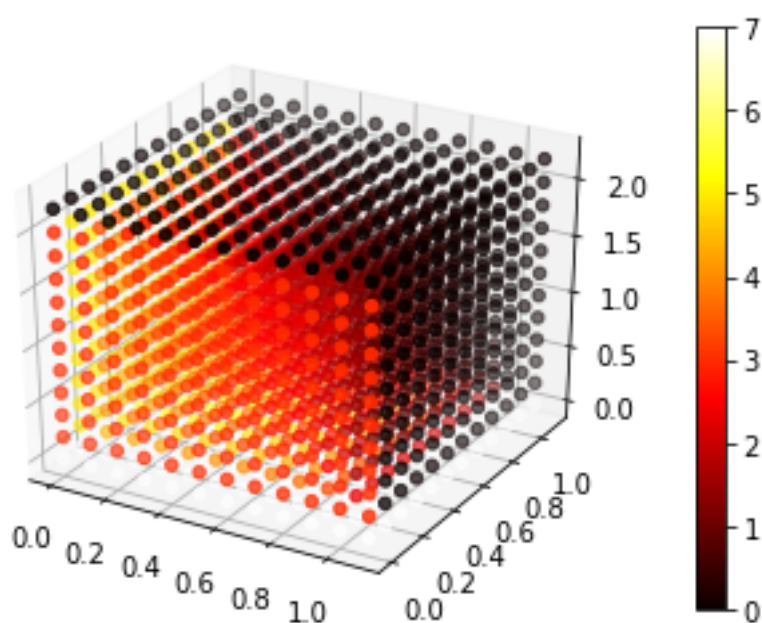
calc_eps - подсчёт ошибки на всех данных.

sync_edges - функция обмена границами между процессами.

Демонстрация работы

Процессы решают задачи на области от 0 до 1 по каждой оси и сеткой 10 точек на каждую ось, кроме oz, по oz 100 точек. Ядровые функции запускаются с конфигурациями <<<dim3(2,2,2), dim3(8,8,8)>>>

размер сетки процессов	время работы, ms
cpu	6734.14
1,1,1	2148.23
1,1,2	7388.99
1,2,1	8296.16
1,2,2	26894.51
2,2,1	27982.13



Выводы

Научился работать с технологией MPI-IO и “дружить” MPI и CUDA. Как видно, на компьютере с 4х ядерном процессоре и одной видеокарте увеличение скорости работы не наблюдается, т.к. не все процессы работают одновременно физически, а чередуют свою работу, а ядровые функции вообще ждут завершения работы других, т.к. на компьютере одна видеокарта, которая не может выполнять несколько таких функций одновременно.