

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: Днепров И. С.

Группа: 8О-407Б-17

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

Москва, 2021

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. Требуется решить задачу, описанную в лабораторной работе №1, с использованием стандарта распараллеливания *openmp* в рамках одного процесса.

Математическая постановка:

$$\frac{\partial^2 u(x, y, z)}{\partial x^2} + \frac{\partial^2 u(x, y, z)}{\partial y^2} + \frac{\partial^2 u(x, y, z)}{\partial z^2} = 0,$$

$$u(x \leq 0, y, z) = u_{left},$$

$$u(x \geq l_x, y, z) = u_{right},$$

$$u(x, y \leq 0, z) = u_{front},$$

$$u(x, y \geq l_y, z) = u_{back},$$

$$u(x, y, z \leq 0) = u_{down},$$

$$u(x, y, z \geq l_z) = u_{up}.$$

Над пространством строится регулярная сетка. С каждой ячейкой сопоставляется значение функции u в точке соответствующей центру ячейки. Граничные условия реализуются через виртуальные ячейки, которые окружают рассматриваемую область.

Поиск решения сводится к итерационному процессу:

$$u_{i,j,k}^{(k+1)} = \frac{\frac{(u_{i+1,j,k}^{(k)} + u_{i-1,j,k}^{(k)})}{h_x^2} + \frac{(u_{i,j+1,k}^{(k)} + u_{i,j-1,k}^{(k)})}{h_y^2} + \frac{(u_{i,j,k+1}^{(k)} + u_{i,j,k-1}^{(k)})}{h_z^2}}{\frac{2}{h_x^2} + \frac{2}{h_y^2} + \frac{2}{h_z^2}}, \# \quad (1)$$

где

$$i = 1..n_x, j = 1..n_y, k = 1..n_z,$$

$$h_x = \frac{l_x}{n_x}, h_y = \frac{l_y}{n_y}, h_z = \frac{l_z}{n_z},$$

$$u_{0,j,k}^{(k)} = u_{left}, \quad u_{n_x+1,j,k}^{(k)} = u_{right},$$

$$u_{i,0,k}^{(k)} = u_{front}, \quad u_{i,n_y+1,k}^{(k)} = u_{back},$$

$$u_{i,j,0}^{(k)} = u_{down}, \quad u_{i,j,n_z+1}^{(k)} = u_{up},$$

$$u_{i,j,k}^{(0)} = u^0.$$

Процесс останавливается, как только

$$\max_{i,j,k} \left| u_{i,j,k}^{(k+1)} - u_{i,j,k}^{(k)} \right| < \varepsilon$$

Для записи результатов в файл использовать барьерную синхронизацию и последовательную пересылку по частям.

Вариант 1:

Распараллеливание основных циклов через *parallel for* (+директива *reduction* для вычисления погрешности).

Входные данные:

На первой строке заданы три числа: размер сетки процессов. Гарантируется, что при запуске программы количество процессов будет равно произведению этих трех чисел. На второй строке задается размер блока, который будет обрабатываться одним процессом: три числа. Далее, задается путь к выходному файлу, в который необходимо записать конечный результат работы программы и точность ε . на последующих строках описывается задача: задаются размеры области l_x , l_y и l_z , граничные условия: u_{down} , u_{up} , u_{left} , u_{right} , u_{front} и u_{back} , и начальное значение u^0 .

Выходные данные:

В файл, определённый во входных данных, необходимо напечатать построчно значения в ячейках сетки в формате с плавающей запятой с семью знаками мантиссы.

Пример:

[illegible]

| | |
|-------------------------|---------------------------|
| 1 1 2 | 4.627978e+00 4.194414e+00 |
| 3 3 3 | 3.239955e+00 |
| mpi.out | 4.728116e+00 4.177304e+00 |
| 1e-10 | 3.095109e+00 |
| 1.0 1.0 2.0 | 3.795164e+00 3.214610e+00 |
| 7.0 0.0 5.0 0.0 3.0 0.0 | 2.407141e+00 |
| 5.0 | |
| | 3.845336e+00 3.121249e+00 |
| | 2.150205e+00 |
| | 3.768252e+00 2.831573e+00 |
| | 1.746256e+00 |
| | 2.828257e+00 1.908052e+00 |
| | 1.133126e+00 |
| | |
| | 3.554538e+00 2.705966e+00 |
| | 1.793770e+00 |
| | 3.376228e+00 2.268328e+00 |
| | 1.267522e+00 |
| | 2.498077e+00 1.440742e+00 |
| | 7.373100e-01 |
| | |
| | 3.399697e+00 2.497911e+00 |
| | 1.638930e+00 |
| | 3.168173e+00 1.987935e+00 |
| | 1.059467e+00 |
| | 2.343237e+00 1.232688e+00 |
| | 5.824692e-01 |
| | |
| | 3.177560e+00 2.254941e+00 |
| | 1.482429e+00 |
| | 2.901943e+00 1.701042e+00 |
| | 8.799475e-01 |
| | 2.160481e+00 1.041743e+00 |
| | 4.653501e-01 |
| | |
| | 2.508778e+00 1.670702e+00 |
| | 1.120755e+00 |
| | 2.204404e+00 1.139741e+00 |
| | 5.713973e-01 |
| | 1.675964e+00 6.908981e-01 |
| | 2.879409e-01 |

Программное и аппаратное обеспечение

GPU

Название: GeForce GT 545

Размер глобальной памяти: 3150381056

Размер константной памяти : 65536

Размер разделяемой памяти: 49152

Регистров на блок: 32768

Максимум потоков на блок: 1024

Размер варпа: 32

Максимальные размеры блока: 1024 x 1024 x 64

Максимальные размеры сетки: 65535 x 65535 x 65535

Количество мультипроцессоров : 3

CPU

Название: Intel Core i7-3770

Частота: 3.40GHz

Размер кеша: 8192 KB

Количество ядер: 4

Количество потоков: 8

MEM

Размер: 16 GB

Тип: ddr3

Прочее

OS: Linux Ubuntu 16.04.6 Редактор: Atom

Выводы

Получен навык распараллеливания вычислений в рамках *MPI* и *OpenMP*, изучены вопросы обмена сообщений между процессами. Была численно решена задача Дирихле для уравнения Лапласа методом Якоби.

Уравнение Лапласа возникает во многих физических задачах механики, теплопроводности, электростатики, гидравлики. Большое значение оператор Лапласа имеет в квантовой физике, в частности в уравнении Шрёдингера.

Сложности возникли из-за наличия вложенных циклов.

В ходе лабораторной выяснилось, что технологии *MPI* и *OpenMP* хорошо подходят для решения параллельных задачах.