МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ СИСТЕМ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Отчет по лабораторной №5

по дисциплине «Системное программирование»

**Технология OpenMP**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил: | студент гр. ФИБ-3301-51-01 |  | / В. Р. Кочкин / |
| Проверил: | к.ф.-м.н. доцент каф. ПМиИ |  | / В. А. Бызов / |

Киров 2022

Цель работы

Получить навыки разработки параллельных алгоритмов с использованием технологии OpenMP.

Задания

**Задание 0**

Проверить, поддерживается ли OpenMP. Если нет, то включить поддержку в

параметрах проекта.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 0](#_Задание_0.).

**Задание 1**

Скопировать фрагмент кода в программу. Запустить на выполнение.

#include <omp.h>

int main () {

int nthreads, tid;

// Создание параллельной области

#pragma omp parallel private(tid)

{

// печать номера потока

tid = omp\_get\_thread\_num();

printf("Hello World from thread = %d\n", tid);

// Печать количества потоков – только master

if (tid == 0) {

nthreads = omp\_get\_num\_threads();

printf("Number of threads = %d\n", nthreads);

}

15

} // Завершение параллельной области

}

Ответить на вопросы:

а) Зачем нужна директива parallel?

б) Сколько потоков было запущено? Почему?

в) Сколько потоков одновременно работают с переменной tid? Почему?

г) Поток с каким tid останется после завершения параллельной области?

Запросить требуемое количество потоков у пользователя. Задать количество потоков для параллельной области.

**Полученные результаты**

**Ответы на вопросы**

Листинг программы приведен в [приложении А задание 1](#_Задание_1.).

**Задание 2**

Написать программу, задающую работу двух потоков. Первый поток в цикле

выводит последовательно числа от 1 до N, а второй – N раз выводит слово

«HELLO». Число N задаётся пользователем.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 2](#_Задание_2.).

**Задание 3**

Написать параллельную программу, находящую поэлементное произведение

двух массивов размера N. Задать параметр schedule, попробовать разные

аргументы.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 3](#_Задание_3.).

**Задание 4**

Написать параллельную программу, вычисляющую максимальное значение

среди элементов вектора, используя директивы critical.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 4](#_Задание_4.).

**Задание 5**

Написать программу, задающую работу 𝑀 + 𝐾 потоков. Первые 𝑀 потоков вычисляют сумму от 1 до 𝑁, а оставшиеся 𝐾 потоков вычисляют длину 𝑁-мерного вектора. Число 𝑁 задаётся пользователем.

Указание. Использовать секции и вложенный параллелизм.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 5](#_Задание_5.).

**Задание 6**

Написать параллельную программу, которая каждый элемент вектора размера N заменяет на его наибольший простой делитель. Число N задается пользователем. Элементы вектора – случайные натуральные числа из диапазона [105, 106].

Замерить среднее время выполнения программы для N = 2\*107, 5\*107 и 108 на 1, 2, 4 и 8 потоках. Вычислить среднее ускорение для 2, 4 и 8 потоков. Построить диаграмму зависимости ускорения от числа потоков для каждого размера вектора (3 графика на одной диаграмме).

**Полученные результаты**

*Проверка работы алгоритма.*

*Таблица 5 – Время вычисление суммы, с*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размерность | Число потоков | | | |
| 1 | 2 | 4 | 8 |
| 107 | 0,0023571 | 0,0014051 | 0,001276 | 0,0011502 |
| 108 | 0,0144072 | 0,0077773 | 0,005149 | 0,004044 |
| 109 | 0,135902 | 0,0690642 | 0,05326 | 0,0318545 |

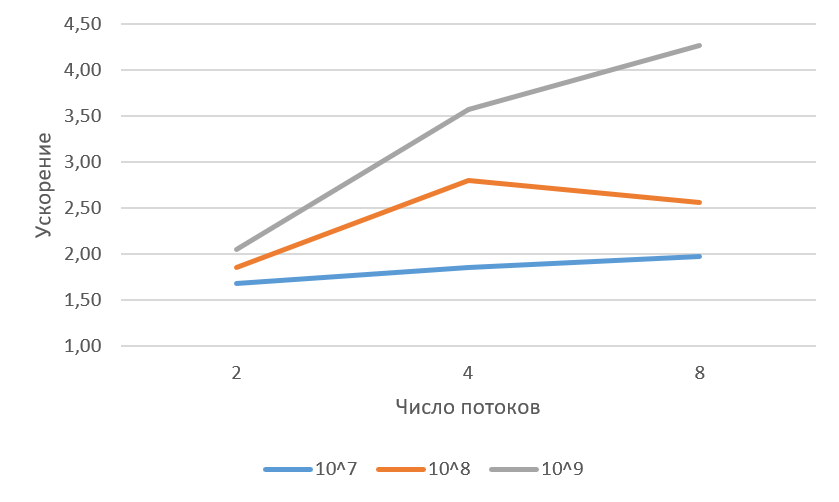
Из таблицы видно, что программа быстрее всего исполнялась на 8 потоках.

*Таблица 6 – Ускорение вычисления суммы, с*

|  |  |  |  |
| --- | --- | --- | --- |
| Размерность | Число потоков | | |
| 2 | 4 | 8 |
| 107 | 1,68 | 1,8472571 | 2,05 |
| 108 | 1,85 | 2,7980035 | 3,56 |
| 109 | 1,97 | 2,551671 | 4,27 |

Из таблицы видно, что максимальное ускорение при 8 потоках, однако ускорение не достигает 8.

*График 3 – Ускорение вычисления суммы, с*



Листинг программы приведен в [приложении А задание 6](#_Задание_6.).

**Задание 7**

Написать параллельную программу, выполняющую умножение двух матриц размера N × N. Разработать программы с использованием распараллеливания циклов разного уровня вложенности.

Замерить среднее время выполнения программ для N = 500, 1000, 2000 на 1, 2, 4 и 8 потоках. Сравнить полученные результаты. Оцените величину накладных расходов на создание и завершение потоков.

Для оптимального варианта вычислить среднее ускорение на 2, 4 и 8 потоках. Построить диаграмму зависимости ускорения от числа потоков для каждого размера матриц (3 графика на одной диаграмме).

**Полученные результаты**

Листинг программы приведен в [приложении А задание 7](#_Задание_7.).

**Задание 8**

Написать параллельную программу, выполняющую поиск максимального значения среди минимальных элементов строк матрицы размера N × N.

Обосновать выбор средств и методов для распараллеливания.

Замерить среднее время выполнения программ для N = 500, 1000, 2000 на 1, 2, 4 и 8 потоках. Вычислить среднее ускорение на 2, 4 и 8 потоках.

**Полученные результаты**

Листинг программы приведен в [приложении А задание 8](#_Задание_8.).

Вывод

В ходе лабораторной работы я получил навыки разработки параллельных алгоритмов с использованием технологии OpenMP. Были произведены замеры времени выполнения программ и на их основании сделаны выводы. Все тесты проводились на процессоре AMD Ryzen 5 3500U 4/8 2.1ГГц.

# **Приложения**

## **Приложение А. Листинги программ**

### Задание 0.

### Задание 1.

### Задание 2.

### Задание 3.

### Задание 4.

### Задание 5.

### Задание 6.

### Задание 7.

### Задание 8.