Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

ОТЧЕТ

по курсу «Параллельное программирование»

Выполнил студент группы №932201

В. А. Викторов

Проверил старший преподаватель ММФ

В. И. Лаева

Томск-2024

# Задание 4.

# Используя MPI, реализовать программу для вычисления определенного интеграла от функции 𝑓(𝑥)= с точностью ε = 107 на отрезке [0.4; 9] с использованием метода правых треугольников. Провести параллельное вычисление интеграла на нескольких процессах и сравнить результаты.

Программа написана на языке C++ с использованием библиотеки MPI для параллельных вычислений. В программе используется функция f(x) для вычисления значения подынтегральной функции. Затем интеграл вычисляется методом правых треугольников на каждом процессе, результаты суммируются и выводятся на экран.

Приведём код написанной программы на языке C++:

#include <iostream>

#include <mpi.h>

#include <math.h>

#include <iomanip>

using namespace std;

double f(double x) {

return sqrt(x \* (3 + x)) / (x + 1);

}

int main(int argc, char \*\*argv)

{

int rank, size, rc;

const double a = 0.4, b = 9;

double I = 0, n = 10000000;

double h = (b - a) / n;

MPI\_Comm comm;

rc = MPI\_Init(&argc, &argv);

comm = MPI\_COMM\_WORLD;

rc = MPI\_Comm\_size(comm, &size);

rc = MPI\_Comm\_rank(comm, &rank);

double\* Result = new double[size];

double t = MPI\_Wtime();

for (double i = 0; i < n; i++)

{

int k = i;

if (rank == k % size)

I += f(a + (i + 1) \* h);

}

rc = MPI\_Gather(&I, 1, MPI\_DOUBLE, Result, 1, MPI\_DOUBLE, 0, comm);

if (rank == 0)

{

I = 0;

for (int j = 0; j < size; j++) I += Result[j];

I \*= h;

cout << "Result: " << setprecision(13) << I << endl;

cout << "Time: " << MPI\_Wtime() - t << endl;

}

delete[] Result;

rc = MPI\_Finalize();

return 0;

}

После выполнения программы на 4 процессах был получен результат:

Result: 8.956794773685

Time: 0.1945350170135

Error: 9.002834211458e-08

Вывод: Программа успешно реализует параллельное вычисление определенного интеграла с использованием MPI и метода правых треугольников. Проведенные вычисления показывают высокую точность и эффективность распараллеливания задачи.

Приведем результаты расчета программы для *n*  100000:

Количество процессоров size = 1

integral = 8.956794773687 time = 0.43743

Количество процессоров size = 2

integral = 8.956794773685 time = 0.27429

Количество процессоров size = 4

integral = 8.956794773685 time = 0 0.19453

Количество процессоров size = 5

integral = 8.956794773685 time = 0.27759

Количество процессоров size = 10

integral = 8.956794773685 time = 0.19232

Во всех запусках программа даёт верный результат вычисления интеграла.

Оценим ускорение

*SP*  *T*1 / *Tp* и эффективность

*EP*  *Sp* / *p* :

*S*  *T*1  0.43743  1.59 *E*  *S*2  1.59  0.795

2

2

*T*2 0.27429 2 2

*S*  *T*1  0.43743 2.25 *E*  *S*4  2.25  0.563

4

4

*T*4 0.19453 4 4

*S*  *T*1  0.43743  1.58 *E*  *S*5  1.58  0.315

5

5

*T*5 0.27759 5 5

*S*  *T*1  0.43743  2.27 *E*  *S*10  0.38  0.227

10 0.19232 10 10 10

*T*

10

Программа демонстрирует ускорение и эффективность при увеличении числа процессоров до определенного предела, после чего ускорение уменьшается. Например, при двух процессорах ускорение составляет примерно 1.53, что означает, что программа работает примерно в 1.59 раза быстрее, чем при одном процессоре. Однако при увеличении числа процессоров до 10 ускорение снижается до 2.27, а эффективность до 0.227. Это может быть связано с увеличением накладных расходов на управление процессами и коммуникацию между ними при большем числе процессоров.