

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ.
WORK12**

ОТЧЕТ О ПРАКТИКЕ

студента 3 курса 311 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Вильцева Данила Денисовича

Проверил

Старший преподаватель

М. С. Портенко

Саратов 2023

СОДЕРЖАНИЕ

1	Work 12.....	3
1.1	Условие задачи.....	3
1.2	Описание алгоритма	3
1.3	Последовательная реализация.....	3
1.4	Параллельная реализация	4
2	Результат работы	7
3	Характеристики компьютера.....	8

1 Work 12

1.1 Условие задачи

Аналогично работе с OMP выполните следующее задание через MPI.

В качестве методов приближенного вычисления двойных интегралов рассмотрим параллельные реализации:

3. метода статистических испытаний.

Номер задания для MPI = (номер задания для OMP+1) mod 3

1.2 Описание алгоритма

Предположим, требуется вычислить определённый интеграл: $\int_a^b f(x) dx$

Рассмотрим случайную величину u , равномерно распределённую на отрезке интегрирования $[a, b]$. Тогда $f(u)$ также будет случайной величиной, причём её математическое ожидание выражается как

$$\mathbb{E}f(u) = \int_a^b f(x) \varphi(x) dx,$$

где $\varphi(x)$ — плотность распределения случайной величины u , равная $\frac{1}{b-a}$ на участке $[a, b]$. Таким образом, искомый интеграл выражается как

$$\int_a^b f(x) dx = (b-a) \mathbb{E}f(u),$$

но математическое ожидание случайной величины $f(u)$ можно легко оценить, смоделировав эту случайную величину и посчитав выборочное среднее.

Итак, бросаем N точек, равномерно распределённых на $[a, b]$, для каждой точки u_i вычисляем $f(u_i)$. Затем вычисляем выборочное среднее: $\frac{1}{N} \sum_{i=1}^N f(u_i)$.

В итоге получаем оценку интеграла:

$$\int_a^b f(x) dx \approx \frac{b-a}{N} \sum_{i=1}^N f(u_i).$$

Точность оценки зависит только от количества точек N .

Этот метод имеет и геометрическую интерпретацию. Он очень похож на описанный выше детерминистический метод, с той разницей, что вместо равномерного разделения области интегрирования на маленькие интервалы и суммирования площадей получившихся «столбиков» мы забрасываем область интегрирования случайными точками, на каждой из которых строим такой же «столбик», определяя его ширину как $\frac{b-a}{N}$, и суммируем их площади.

1.3 Последовательная реализация

```
#include <iostream>
#include <mpi.h>
#define _USE_MATH_DEFINES
#include <math.h>
#include <time.h>
#include <vector>
#include <iomanip>
using namespace std;

const double fraction = 1.0 / (static_cast<double>(RAND_MAX) + 1.0);
```

```

double integral(const double a, const double b, const double h)
{
    int i, n;
    double sum; // Локальная переменная для подсчета
    ↪ интеграла
    double x; // Координата точки сетки
    n = (int)((b - a) / h); // Количество точек сетки интегрирования
    sum = 0.0;

    for (i = 0; i < n; i++)
    {
        x = rand() * fraction * (b - a) + a;
        if (x != 0)
            sum += exp(-5 * x);
    }

    return sum * (b - a) / n;
}

int main()
{
    double res;
    double stime, ftime; // Время начала и конца расчета
    double a = 0.0; // Левая граница интегрирования
    double b = 1000.0; // Правая граница интегрирования
    double h = 0.001; // Шаг интегрирования
    stime = clock();
    res = integral(a, b, h); // вызов функции интегрирования
    ftime = clock();
    cout << "Integral value: " << res << endl;
    cout << "Time: " << (ftime - stime) / CLOCKS_PER_SEC << endl;

    system("pause");
    return 0;
}

```

1.4 Параллельная реализация

Распараллелим программу так чтобы переменные x и y были локальными для каждого потока, и общий результат будем получать в sum аналогично задаче 2.

```

#include <iostream>
#include <mpi.h>
#define _USE_MATH_DEFINES

```

```

#include <math.h>
#include <time.h>
#include <vector>
#include <iomanip>
using namespace std;

const double fraction = 1.0 / (static_cast<double>(RAND_MAX) + 1.0);

double integral(const double a, const double b, const double h)
{
    int i, n;
    double sum; // Локальная переменная для подсчета
    ↪ интеграла
    double x; // Координата точки сетки
    n = (int)((b - a) / h); // Количество точек сетки интегрирования
    sum = 0.0;

    // mpi part
    int commsize;
    int rank;
    double Result;
    MPI_Init(NULL, NULL);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &commsize);
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    for (i = 0; i < n; i++)
    {
        x = rand() * fraction * (b - a) + a;
        if (x != 0)
            sum += exp(-5 * x);
    }

    // mpi part
    MPI_Reduce(&sum, &Result, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
    MPI_Finalize();

    return Result * (b - a) / n;
}

int main()
{
    double res;
    double stime, ftime; // Время начала и конца расчета
    double a = 0.0; // Левая граница интегрирования

```

```

double b = 1000.0;                                // Правая граница интегрирования
double h = 0.001;                                // Шаг интегрирования
stime = clock();
res = integral(a, b, h); // вызов функции интегрирования
ftime = clock();
cout << "Integral value: " << res << endl;
cout << "Time: " << (ftime - stime) / CLOCKS_PER_SEC << endl;

system("pause");
return 0;
}

```

2 Результат работы

```
Integral value: 0.189752  
Time: 0.156
```

Рисунок 1 – Результат последовательной реализации

```
Integral value: 0.189752  
Time: 0.124
```

Рисунок 2 – Результат параллельной реализации

Ускорение: Переход к многопоточной версии дает ускорение в 1.23 раза

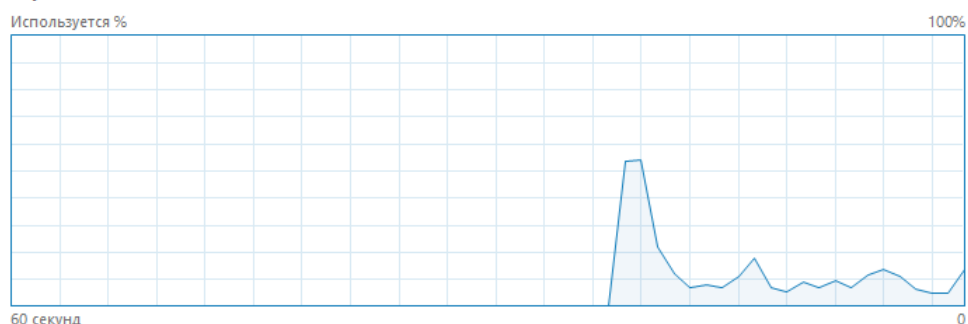
3 Характеристики компьютера

Характеристики устройства

Имя устройства	DESKTOP-MSS8D39
Процессор	Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.20 GHz
Оперативная память	8,00 ГБ
Код устройства	E3BB953D-13B0-42A7-944B-1ED9FD0E C328
Код продукта	00330-80000-00000-AA153
Тип системы	64-разрядная операционная система, процессор x64

ЦП

Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz



Использование	Скорость	Базовая скорости:	3,20 ГГц
14%	3,43 ГГц	Сокетов:	1
Процессы	Потоки	Ядра:	4
220	3285	Логических процессоров:	4
Время работы	Дескрипторы	Виртуализация:	Включено
100:23:51:24	170005	Кэш L1:	256 КБ
		Кэш L2:	1,0 МБ
		Кэш L3:	6,0 МБ