



Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики

Фазлетдинов Рамиль Рустамович, 608

**ЗАДАНИЕ ПО КУРСУ «СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И
ТЕХНОЛОГИИ» ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МНОГОМЕРНЫХ
ФУНКЦИЙ МЕТОДОМ МОНТЕ-КАРЛО**

ВАРИАНТ 11

Москва
2022

Содержание

| | | |
|----------|--|----------|
| 1 | Постановка задачи | 3 |
| 1.1 | Введение | 3 |
| 1.2 | Математическая постановка | 3 |
| 1.3 | Численный метод решения задачи | 3 |
| 1.3.1 | Описание требований к программной реализации | 4 |
| 1.4 | Спецификация варианта | 4 |
| 2 | Ход работы | 4 |
| 2.1 | Аналитическое решение | 4 |
| 2.2 | Краткое описание программной реализации | 4 |
| 2.3 | Результаты запусков | 5 |

1 Постановка задачи

1.1 Введение

В качестве модельной задачи предлагается задача вычисления многомерного интеграла методом Монте-Карло. Программная реализация должна быть выполнена на языке С или С++ с использованием библиотеки параллельного программирования MPI. Требуется исследовать масштабируемость параллельной MPI-программы на следующих параллельных вычислительных системах ВМК МГУ:

1. IBM Blue Gene/P
2. IBM Polus

1.2 Математическая постановка

Функция $f(x, y, z)$ — непрерывна в ограниченной замкнутой области $G \subset R^3$. Требуется вычислить определённый интеграл:

$$I = \iiint_G f(x, y, z) dx dy dz$$

1.3 Численный метод решения задачи

Пусть область G ограничена параллелепипедом

$$\Pi : \begin{cases} a_1 \leq x \leq b_1, \\ a_2 \leq y \leq b_2, \\ a_3 \leq z \leq b_3 \end{cases}$$

Рассмотрим функцию:

$$F(x, y, z) = \begin{cases} f(x, y, z), & (x, y, z) \in G \\ 0, & (x, y, z) \notin G \end{cases}$$

Преобразуем искомый интеграл:

$$I = \iiint_G f(x, y, z) dx dy dz = \iiint_{\Pi} F(x, y, z) dx dy dz$$

Пусть $p_1(x_1, y_1, z_1), p_2(x_2, y_2, z_2), \dots$ — случайные точки, равномерно распределённые в Π . Возьмём n таких случайных точек. В качестве приближённого значения интеграла предлагается использовать выражение:

$$I \approx |\Pi| \cdot \frac{1}{n} \sum_{i=1}^n F(p_i),$$

где $|\Pi|$ - объём параллелепипеда Π . $|\Pi| = (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)$

1.3.1 Описание требований к программной реализации

Параллельная MPI-программа принимает на вход требуемую точность и генерирует случайные точки до тех пор, пока требуемая точность не будет достигнута. Программа вычисляет точность как модуль разности между приближённым значением, полученным методом Монте-Карло, и точным значением, вычисленным аналитически.

Программа считывает в качестве аргумента командной строки требуемую точность ϵ и выводит четыре числа:

- Посчитанное приближённое значение интеграла
- Ошибка посчитанного значения: модуль разности между приближённым и точным значениями интеграла
- Количество сгенерированных случайных точек
- Время работы программы в секундах

Время работы программы измеряется следующим образом. Каждый MPI-процесс измеряет своё время выполнения, затем среди полученных значений берётся максимум.

1.4 Спецификация варианта

Необходимо выполнить задачу в парадигме «мастер–рабочие»: один из процессов («мастер») генерирует случайные точки и передаёт каждому из остальных процессов («рабочих») отдельный, предназначенный для него, набор сгенерированных случайных точек.

Вариант интеграла:

$$\iiint_G \sqrt{x^2 + y^2} \, dx dy dz,$$

где область G ограничена поверхностями $x^2 + y^2 = z^2$, $z = 1$.

2 Ход работы

2.1 Аналитическое решение

$$\begin{aligned} \iiint_G \sqrt{x^2 + y^2} \, dx dy dz &= \iint_Q dx dy \int_{\sqrt{x^2 + y^2}}^1 \sqrt{x^2 + y^2} \, dz = \\ \iint_{x^2 + y^2 \leq z^2} (\sqrt{x^2 + y^2} - (x^2 + y^2)) \, dx dy &= \int_0^{2\pi} d\varphi \int_0^1 (r - r^2)r \, dr = 2\pi \left(\frac{r^3}{3} - \frac{r^4}{4} \right) \Big|_0^1 = 2\pi \left(\frac{1}{3} - \frac{1}{4} \right) = \frac{\pi}{6} \end{aligned}$$

2.2 Краткое описание программной реализации

В данной реализации мастер генерирует $\text{dots_each_proc} * \text{comm_size} * 3$ точек, где dots_each_proc - количество троек (x, y, z) , comm_size - количество процессов. Затем используется функция Scatter для того, чтобы каждый процесс забрал в свой буфер свою часть точек, то есть

`dots_each_proc * 3`. После подсчёта каждым процессом своей суммы происходит операция редукции, результат которой анализирует мастер для сравнения с заданной точностью.

2.3 Результаты запусков

Таблица 1: Таблица с результатами расчётов для системы Polus

| Точность ε | Число MPI-процессов | Время работы программы (с) | Ускорение | Ошибка |
|------------------------|---------------------|----------------------------|----------------|-------------|
| $3.0 \cdot 10^{-5}$ | 2 | 0.010139 | 1 | 9.78653e-06 |
| | 4 | 0.00626628 | 1.618025367523 | 2.72375e-05 |
| | 8 | 0.153313 | 0.066132682812 | 2.43043e-06 |
| | 16 | 0.210156 | 0.048245113154 | 1.6055e-05 |
| | 32 | | | |
| $5.0 \cdot 10^{-6}$ | 2 | 0.0432811 | 1 | 1.79392e-06 |
| | 4 | 0.154252 | 0.280586961595 | 2.96542e-06 |
| | 8 | 0.141004 | 0.306949448243 | 2.43043e-06 |
| | 16 | 0.215428 | 0.200907495776 | 2.73571e-06 |
| | 32 | | | |
| $1.5 \cdot 10^{-6}$ | 2 | 0.0307092 | 1 | 3.81489e-07 |
| | 4 | 0.156541 | 0.196173526424 | 1.34841e-06 |
| | 8 | 0.166259 | 0.184706993306 | 5.92816e-07 |
| | 16 | 0.340262 | 0.090251629627 | 1.27556e-06 |
| | 32 | | | |