



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	Исследование масштабируемости	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 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 на каждом шаге генерирует $(\text{comm_size} - 1) * \text{block}$ точек, где $\text{block} = \text{dots_each_iter} / \text{comm_size}$, dots_each_iter - количество троек (x, y, z) равное

1024, comm_size - количество процессов. Затем используется функция Scatter для того, чтобы каждый процесс забрал в свой буфер $3 * \text{block}$ точек. После подсчёта каждым процессом своей суммы происходит операция редукции, результатом которой анализирует мастер для сравнения с заданной точностью. Нахождение времени работы основывается на поиске максимума времени непосредственного подсчёта среди slave-процессов.

2.3 Исследование масштабируемости

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	0.0689793	1	2.84684e-05
	4	0.0275973	2.499	2.62617e-05
	8	0.0116789	5.906	2.97089e-05
	16	0.00624028	11.0538	2.97089e-05
$5.0 \cdot 10^{-6}$	2	0.0835625	1	4.40623e-06
	4	0.0236104	3.5392	3.79917e-06
	8	0.0111897	7.4678	4.66277e-06
	16	0.00539678	15.4837	3.06944e-06
$1.5 \cdot 10^{-6}$	2	0.0747723	1	5.52963e-07
	4	0.0304406	2.4563	5.91808e-07
	8	0.0128159	5.8343	6.53105e-07
	16	0.00606637	12.3257	8.00831e-07

Таблица 2: Таблица с результатами расчётов для системы Polus с разными зёрнами генерации при точности $5.0 \cdot 10^{-6}$

Зерно	Число MPI-процессов	Время работы программы (с)	Ускорение	
173	2	0.00796303	1	2.94507e-07
	4	0.00267344	2.9785	3.26349e-06
	8	0.000441078	18.0535	8.98286e-07
	16	0.000287857	27.6631	8.98286e-07
1731	2	0.00960783	1	2.94507e-07
	4	0.0233436	3.5895	3.26349e-06
	8	0.000446973	21.495	8.98286e-07
	16	0.000205957	46.6496	4.29615e-06
17311	2	0.00814864	1	2.94507e-07
	4	0.00338578	2.4067	3.26349e-06
	8	0.000447198	18.2215	8.98286e-07
	16	0.000225561	36.1261	4.29615e-06

Графики находятся на следующей странице.

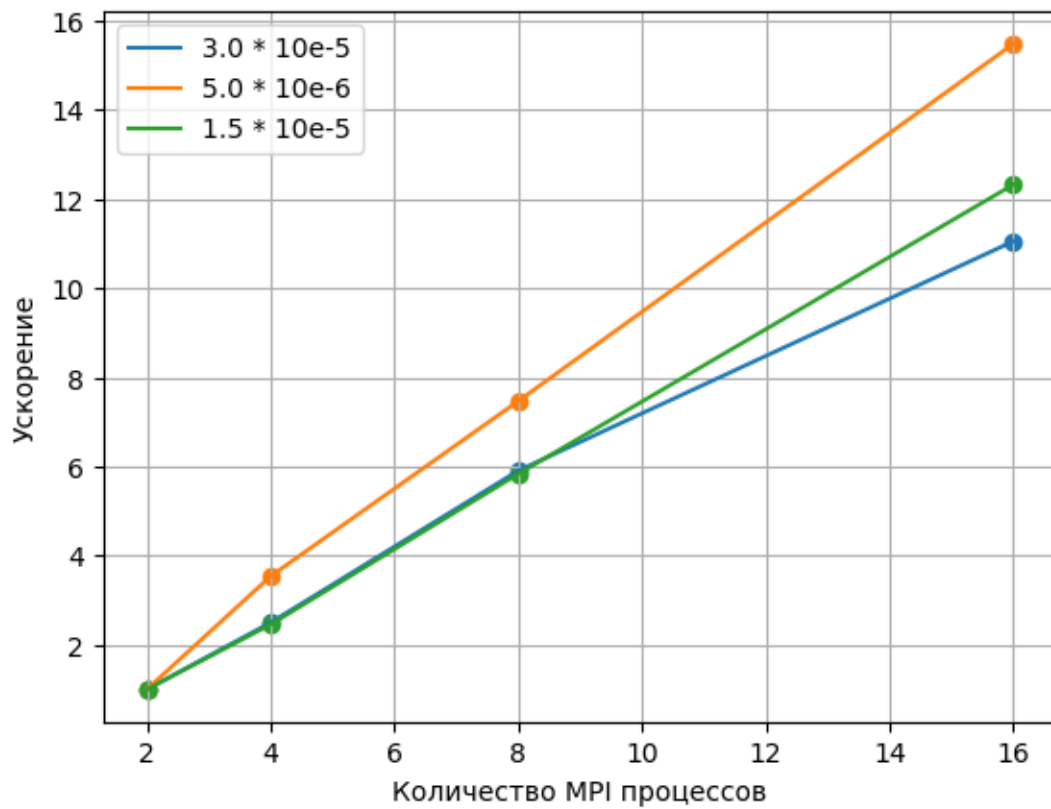


Рис. 1: Зависимость ускорения от количества процессов

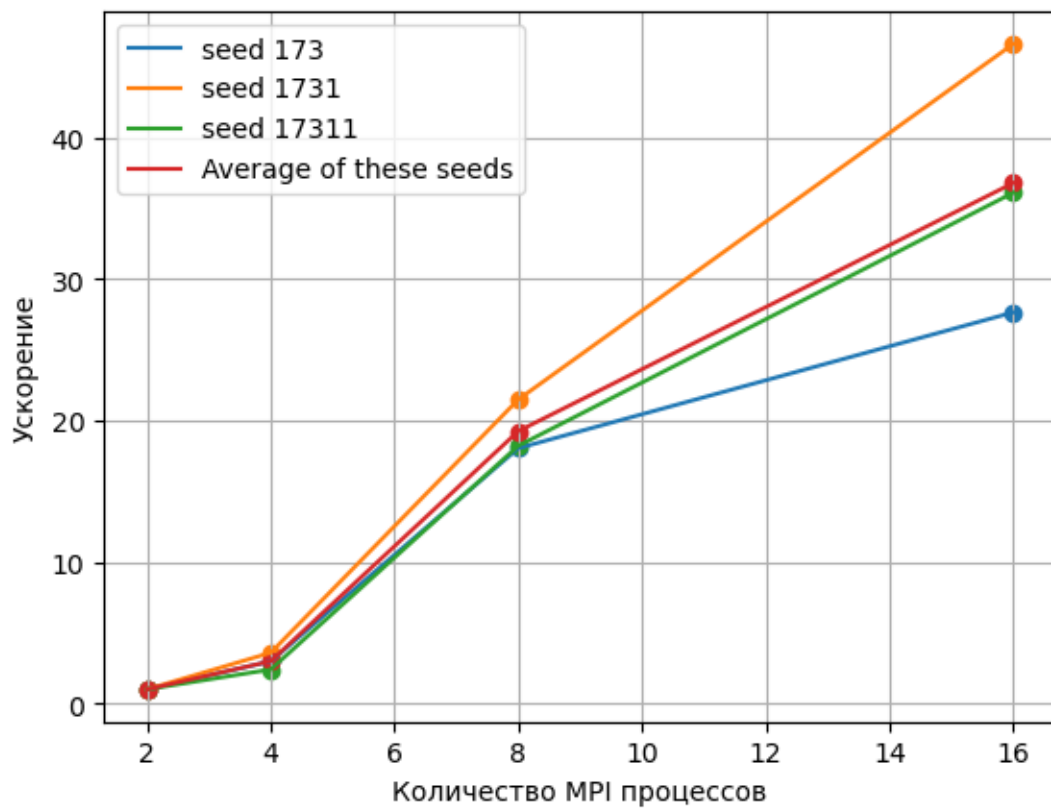


Рис. 2: Зависимость ускорения при различных зернах и фиксированной точности $5.0 \cdot 10^{-6}$