



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

Филиппов Андрей Александрович
608 группа
7 вариант

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МНОГОМЕРНЫХ ФУНКЦИЙ МЕТОДОМ МОНТЕ-КАРЛО

Москва
2022

1 Введение

В качестве модельной задачи предлагается задача вычисления многомерного интеграла методом Монте-Карло.

Программная реализация должна быть выполнена на языке C или C++ с использованием библиотеки параллельного программирования MPI.

Требуется исследовать масштабируемость параллельной MPI-программы на следующих параллельных вычислительных системах ВМК МГУ:

1. IBM Polus

2 Математическая постановка задачи

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

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

где функция $f(x, y, z) = \sqrt{y^2 + z^2}$,

область $G = \{(x, y, z) : 0 \leq x \leq 2, y^2 + z^2 \leq 1\}$

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 . $|\Pi| = (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)$

4 Нахождение точного значения интеграла аналитически

$$I = \iiint_G \sqrt{y^2 + z^2} dx dy dz = \{x = x, y = r \cos \phi, z = r \sin \phi\} = \int_0^2 dx \int_0^{2\pi} d\phi \int_0^1 r^2 dr = \frac{4\pi}{3}$$

5 Описание 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.

В моём варианте реализована независимая генерация точек MPI-процессом. Каждый процесс генерирует одинаковое кол-во точек, вычисляют свою часть суммы, затем операцией Allreduce каждый процесс получает общую сумму и считает разность между получившимся значением интеграла и истинным значением (ошибку), затем каждый процесс проверяет достижение нужной точности и если точность не достигнута, то уходит на следующую итерацию.

6 Исследование масштабируемости программы на системе Polus

Таблица 1: Таблица с результатами расчётов для системы Polus

Точность ϵ	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	1	3.3168	1	2.798e-05
	4	0.882723	3.757	2.800e-05
	16	0.276318	12.003	2.799e-05
$5.0 \cdot 10^{-6}$	1	59.6326	1	2.1861e-06
	4	15.2831	3.901	2.1861e-06
	16	4.27734	13.941	2.1861e-06
$1.5 \cdot 10^{-6}$	1	132.146	1	1.25052e-07
	4	33.628	3.929	1.25215e-07
	16	9.236	14.307	1.24612e-07

Таблица 2: Таблица с результатами расчётов для различных сидов на точности $5e - 06$

Seed	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$rank + 2$	1	39.6867	1	3.85744e-06
	4	9.9513	3.988	3.85744e-06
	16	2.84224	13.963	3.85744e-06
$rank + 20$	1	59.5916	1	3.78056e-07
	4	14.9225	3.993	3.78056e-07
	16	4.28267	13.914	3.78056e-07
$rank + 200$	1	30.1624	1	1.50095e-06
	4	7.64959	3.943	1.50095e-06
	16	2.17981	13.837	1.50094e-06
$rank + 2000$	1	10.1949	1	3.61216e-06
	4	2.55248	3.994	3.61216e-06
	16	0.723238	14.096	3.61216e-06
$rank + 20000$	1	99.4006	1	2.92548e-06
	4	25.5332	3.892	2.92548e-06
	16	7.14777	13.906	2.92548e-06

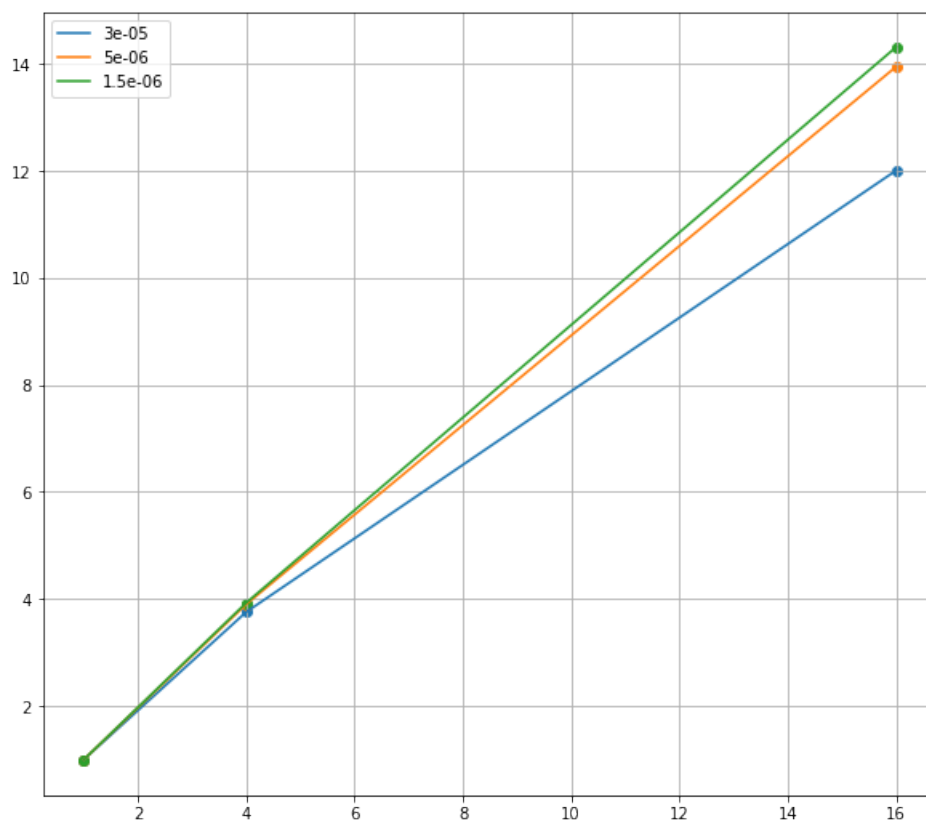


Рис. 1: График ускорения

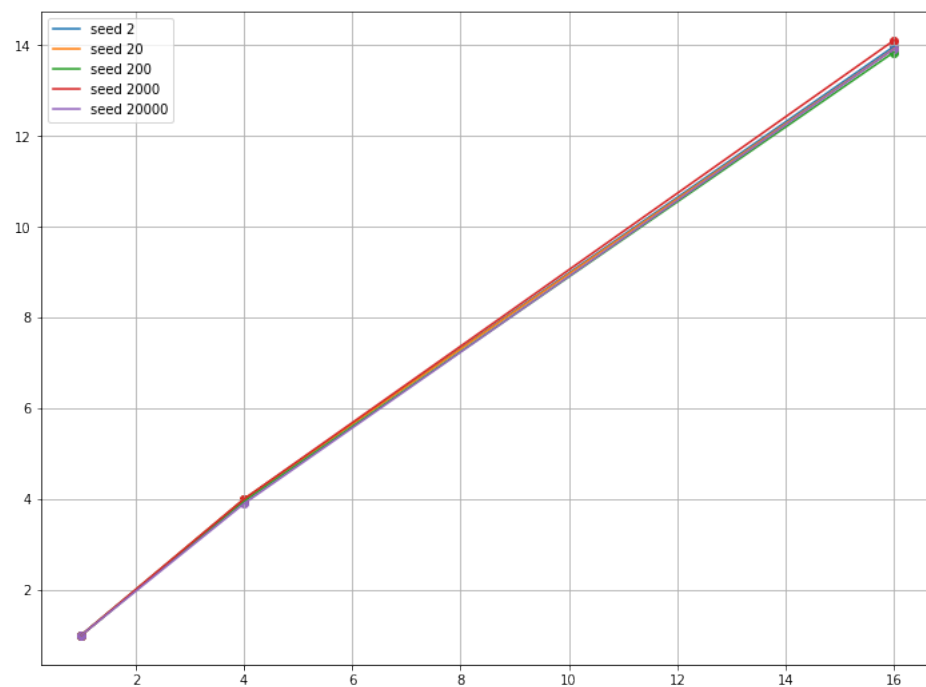


Рис. 2: Графики ускорения с различными seed