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

1.1 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 Blue Gene/P
2. IBM Polus

1.2 Математическя 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 ограниченна параллелепипедом

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

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 $p_1(x_1,\, y_1,\, z_1)$, $p_2(x_2,\, y_2,\, z_2)$, ... — 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:

$$\iiint_G \sqrt{x^2+y^2}\,dxdydz,$$

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

2 Ход работы

2.1 Аналитическое решение

$$\iiint_G \sqrt{x^2+y^2}\,dxdydz = \iint_Q dxdy \int_{\sqrt{x^2+y^2}}^{1} \sqrt{x^2+y^2}\,dz =$$

$$\iint_{x^2+y^2\le z^2} (\sqrt{x^2+y^2} - (x^2+y^2))\,dxdy = \int_0^{2\pi} d\varphi \int_0^1 (r-r^2)r\,dr = 2\pi(\frac{r^3}{3} - \frac{r^4}{4})|_0^1 = 2\pi(\frac{1}{3} - \frac{1}{4}) = \frac{\pi}{6}$$

2.2 Краткое описание программной реализации

В данной реализации мастер на каждом шаге генерирует dots_each_proc * (comm_size - 1) * 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.

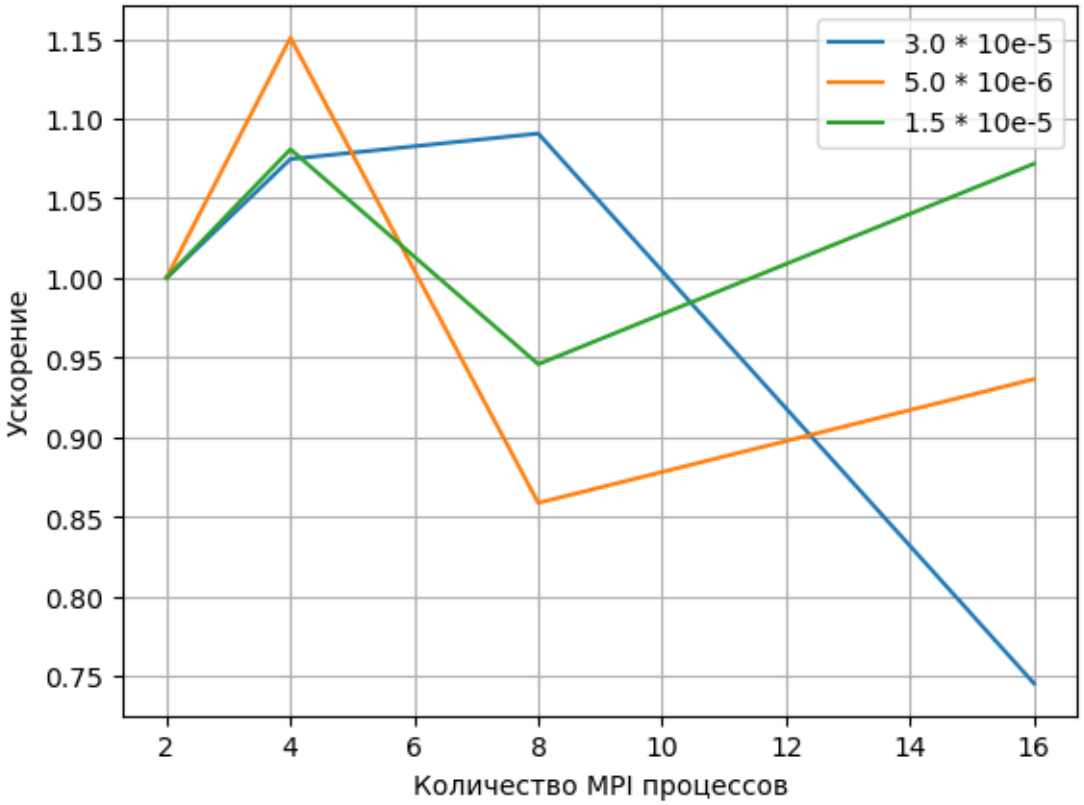
2.3 Результаты запусков

В процессе запусков были протестированы несколько различных зерен генерации и найдены несколько вариантов с довольно большим количеством точек для сходимости (2-15 миллионов), а так же такие зерна, при которых сходимости не наблюдалось в течение 10 минут на самый низкой точности вычислений. В конечном счёте был выбран вариант, сходимость которого достигалась примерно за 5.5 млн точек.

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	0.907691	1	2.85806e-05
	4	0.844668	1.0746	2.81817e-05
	8	0.832212	1.0906	2.74798e-05
	16	1.21767	0.7454	2.55414e-05
	32			
$5.0 \cdot 10^{-6}$	2	0.987908	1	4.13782e-06
	4	0.858466	1.1507	4.77977e-06
	8	1.15035	0.8587	4.61699e-06
	16	1.05498	0.9364	4.05108e-06
	32			
$1.5 \cdot 10^{-6}$	2	1.33261	1	2.25045e-07
	4	1.23299	1.0807	4.85203e-07
	8	1.40894	0.9458	8.69409e-07
	16	1.24351	1.0716	2.35053e-07
	32			

Для варианта 32 не удалось получить результаты, поскольку планировщик не хотел выделять ресурсы и задача просто выходила из очереди по истечению выделенного времени.


3.0 * 10e-5
5.0 * 10e-6
1.5 * 10e-5
Ускорение
1.15
1.10
1.05
1.00
0.95
0.90
0.85
0.80
0.75
2
4
6
8
10
12
14
16
Количество MPI процессов