Московский Государственный Университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по второму практическому заданию
в рамках курса
«Суперкомпьютерное моделирование и технологии»

Выполнила:
Соколова Екатерина Витальевна,
627 группа
Вариант 13

Москва, 2022

Математическая постановка задачи

Функция $f(x, y, z)$ — непрерывна в ограниченой замкнутой области $G \subset R_3$. Требуется вычислить определённый интеграл:

$$I = \iiint\limits_{\text{G}} f(x, y, z)\, dx\, dy\, dz.$$

Для моего варианта №13 (интеграл №5) функция $f(x, y, z) = x^3 y^2 z$. Требуется вычислить интеграл:

$$I = \iiint\limits_{\text{G}} x^3 y^2 z\, dx\, dy\, dz, \tag{1}$$

где область $G = \{(x, y, z) : -1 \le x \le 0, -1 \le y \le 0, -1 \le z \le 0\}$.

Численный метод решения задачи

Будем производить вычисления методом Монте-Карло для численного интегрирования.

Пусть область $G \subset R_3$ ограничена параллелепипедом П: $\begin{cases} a_1 \le x \le b_1 \\ a_2 \le y \le b_2 \\ a_3 \le z \le b_3 \end{cases}$

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:

$$I = \iiint\limits_{\text{G}} f(x, y, z)\, dx\, dy\, dz = \iiint\limits_{\text{П}} F(x, y, z)\, dx\, dy\, dz.$$

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

$$I \approx |\Pi| \bullet \frac{1}{n} \sum_{i=1}^{n} F(p_i), \qquad (2)$$

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

Для заданного интеграла область ограничена кубом с вершинами (m_i, n_j), где $i, j = \overline{1, 8}$, $m_i \in \{0, -1\}$, $n_j \in \{0, -1\}$. Следовательно, $|\Pi| = 1$.

Обозначим искомое приближенное значение интеграла за $I_0 = \frac{1}{n} \sum_{i=1}^{n} F(p_i)$.

Аналитическое вычисление точного значения заданного интеграла

Область G задана таким образом, что ограничения на переменные x, y и z не зависят от этих переменных (заданы константно). Воспользуемся этим и вычислим интеграл следующим образом:

$$\iiint\limits_G x^3 y^2 z \, dx \, dy \, dz = \int_{-1}^{0}\int_{-1}^{0}\int_{-1}^{0} x^3 y^2 z \, dx \, dy \, dz = \int_{-1}^{0} dx \int_{-1}^{0} dy \int_{-1}^{0} x^3 y^2 z \, dz =$$

$$= \int_{-1}^{0} dx \int_{-1}^{0} dy \, (\frac{x^3 y^2 z^2}{2})|_{-1}^{0} = \int_{-1}^{0} dx \int_{-1}^{0} \frac{-x^3 y^2}{2} dy = \int_{-1}^{0} dx \, (\frac{-x^3 y^3}{6})|_{-1}^{0} =$$

$$= \int_{-1}^{0} \frac{-x^3}{6} dx = (\frac{-x^4}{24})|_{-1}^{0} = \frac{1}{24}.$$

Выразим значение вещественным числом: $\frac{1}{24} = 0{,}041(6)$.

Программная реализация

В рамках данного практического задания написана параллельная MPI программа, вычисляющая приближённое значение интеграла (1) с заданной точностью. Интерфейс программы соответствует предъявленным требованиям:

Ввод:
- требуемая точность (ε),

Вывод:
- посчитанное приближённое значение интеграла (I_0),
- ошибка посчитанного значения ($|I - I_0|$),
- количество сгенерированных случайных точек (n),
- время работы программы в секундах (t).

Время работы программы вычисляется как $t = max(t_i)$, где $0 \le i \le (p-1)$, p — количество используемых процессов, t_i — время работы i-го процесса.

Распараллеливание метода Монте-Карло осуществляется с использованием парадигмы «мастер-рабочие»: один из процессов («мастер») генерирует случайные точки и передаёт каждому (не считая себя) процессу — «рабочему» отдельный, предназначенный для него, набор сгенерированных случайных точек.

На каждом шаге процесс «рабочий» считает свою часть суммы из формулы (2), затем, операцией редукции, вычисляется общая сумма и, следовательно, значение I_0. Если значение ошибки $|I - I_0| > \varepsilon$, то «мастер» генерирует новые точки и вычисление продолжается.

Исследование масштабируемости программы на системе Polus

Собираемая статистика сильно зависит от способа генерации точек. При некоторых видах рандомизации значения времени выполнения могут отличаться в несколько раз при двух запусках с одними и теми же параметрами. Это обусловлено различным числом сгенерированных точек (n).

Итоговая версия статистики собиралась на программе, генерирующей точки функцией $rand\_r()$ с зерном, равным $rank$. При таком подходе во время каждого запуска (совпадает число процессов и требуемая точность) будет совпадать количество сгенерированных точек. Тогда выявляется

закономерность по времени, но получаемое значение точности неизменно. Не исключены «скачки» ускорения, обусловленные случайно удачной генерацией точек (для достижения требуемой точности хватило количества точек, существенно меньшего, чем для запусков на другом числе процессов). Такие случаи часто возникают при запуске на 8 и 16 процессах, которые не требовалось вносить в таблицу, а также при запуске на 4 процессах для точности $3.0 \bullet 10^{-5}$. Данная точка присутствует на графике, но её не следует учитывать в расчётах.

Точность (ε)	Число MPI процессов (p)	Время работы программы (T_p), с	Ускорение (S_p)	Ошибка ($\lvert I - I_0 \rvert$)
$3.0 \bullet 10^{-5}$	2	0,1859738856	1	0.0000281419
	4	0,0007108916	261,6065312911	0.0000158572
	32	0,0350989980	5,2985525570	0.0000274365
$5.0 \bullet 10^{-6}$	2	0,1880635094	1	0.0000049043
	4	0,1990832045	0,9446477912	0.0000048734
	32	0,0422526981	4,4509230855	0.0000003324
$1.5 \bullet 10^{-6}$	2	0,1840278382	1	0.0000004894
	4	0,2015206053	0,9131961366	0.0000010100
	8	0,0854686329	2,1531623001	0.0000001840
	32	0,0360910601	5,0989867765	0.0000003324

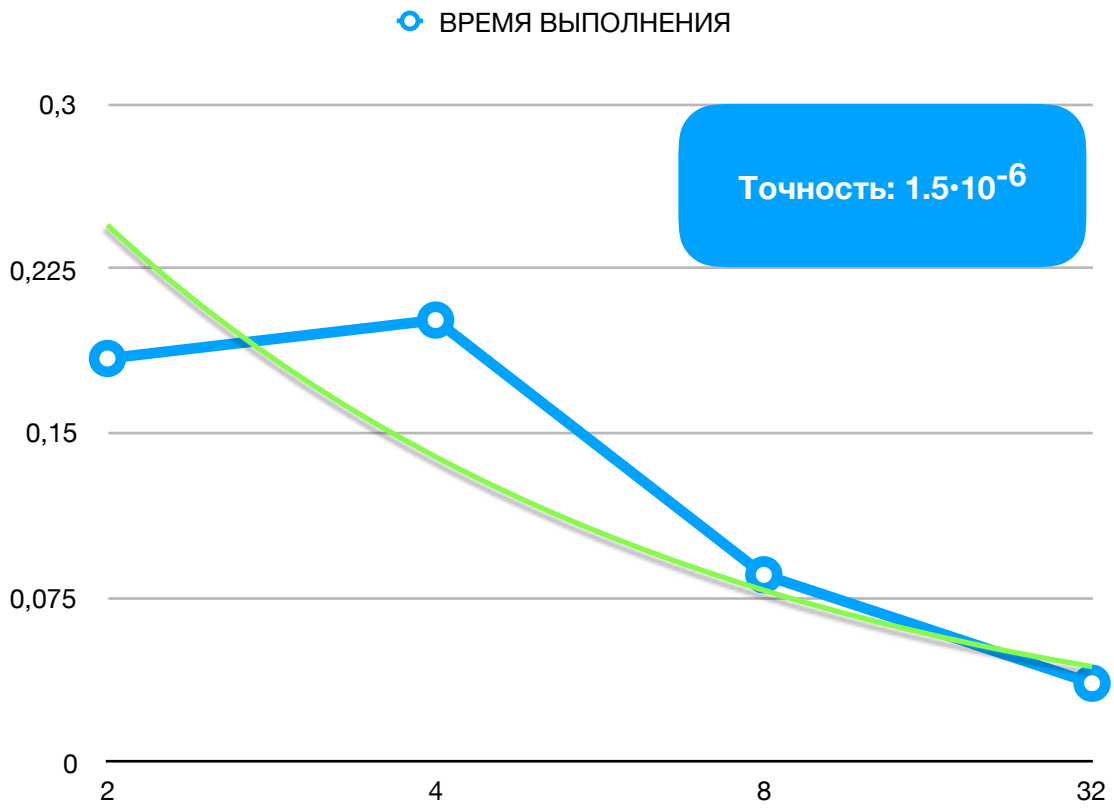
Время работы программы и значение ошибки в таблице являются усреднёнными значениями по 3 запускам для каждой пары требуемой точности и числа процессов (таблица с результатами запусков, в т.ч. на 8 и 16 процессах, зафиксирована в репозитории с исходным кодом).

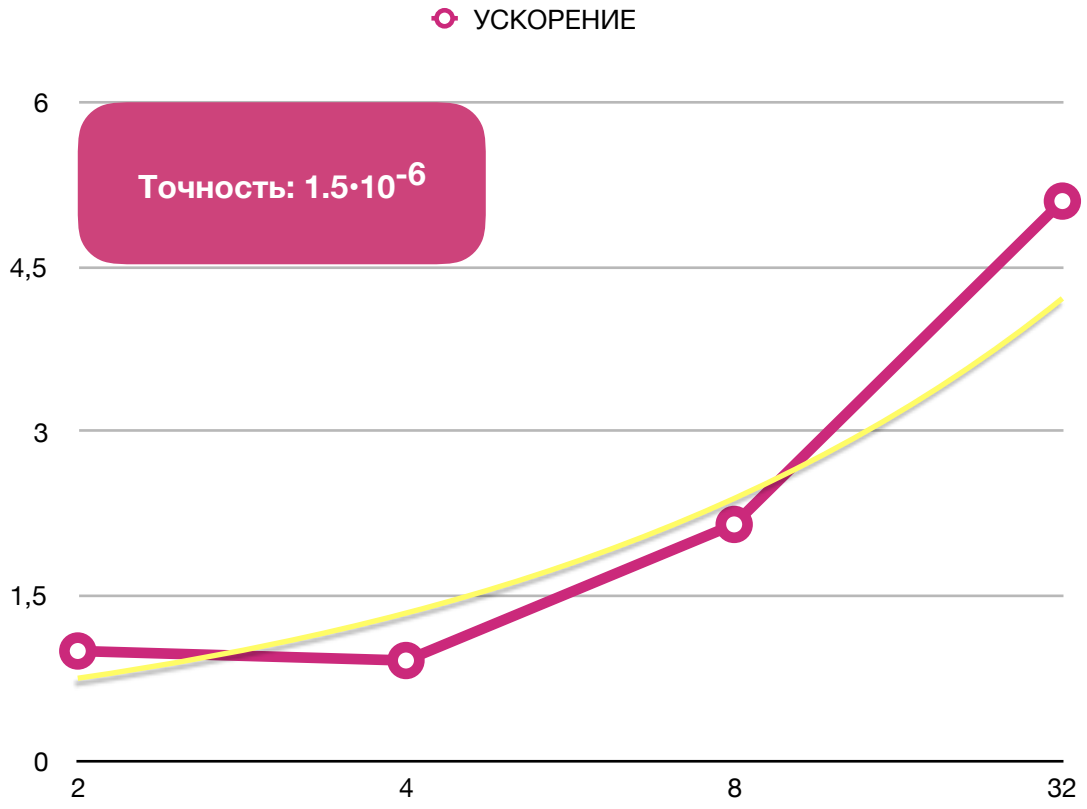
Ускорение программы, запущенной на p MPI-процессах, описывается величиной

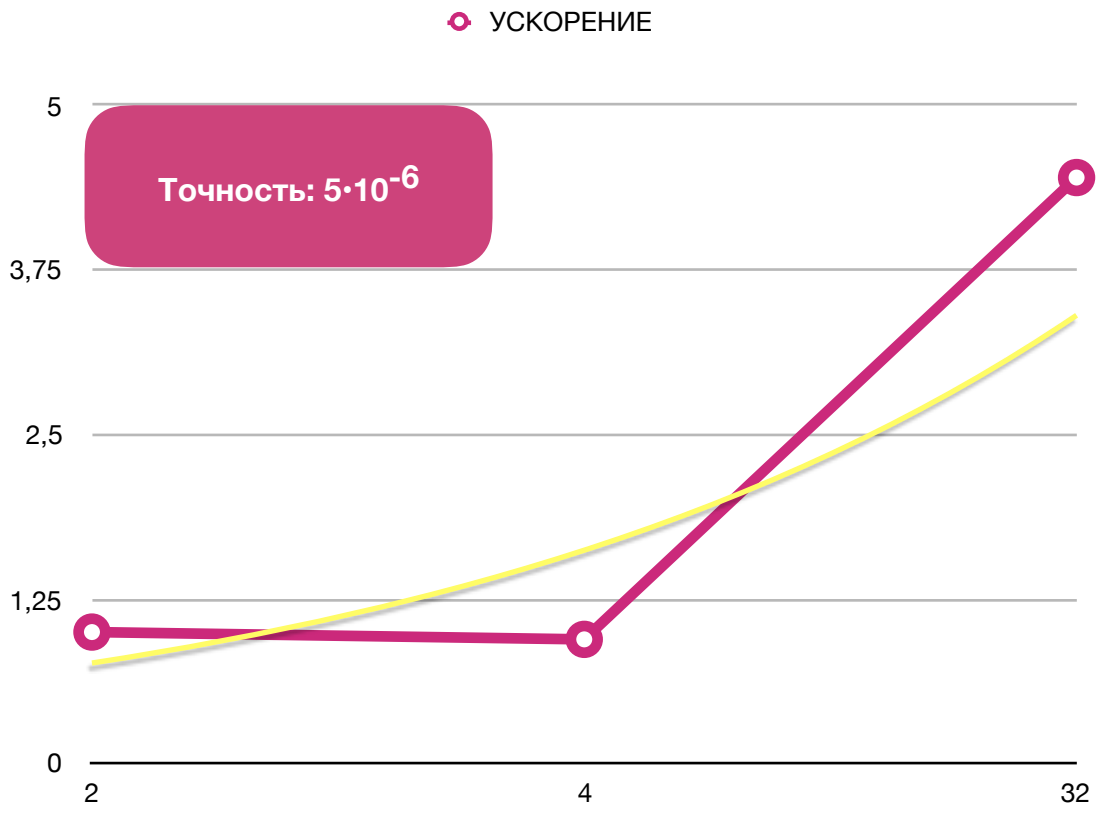
$$S_p = \frac{T_1}{T_p},$$

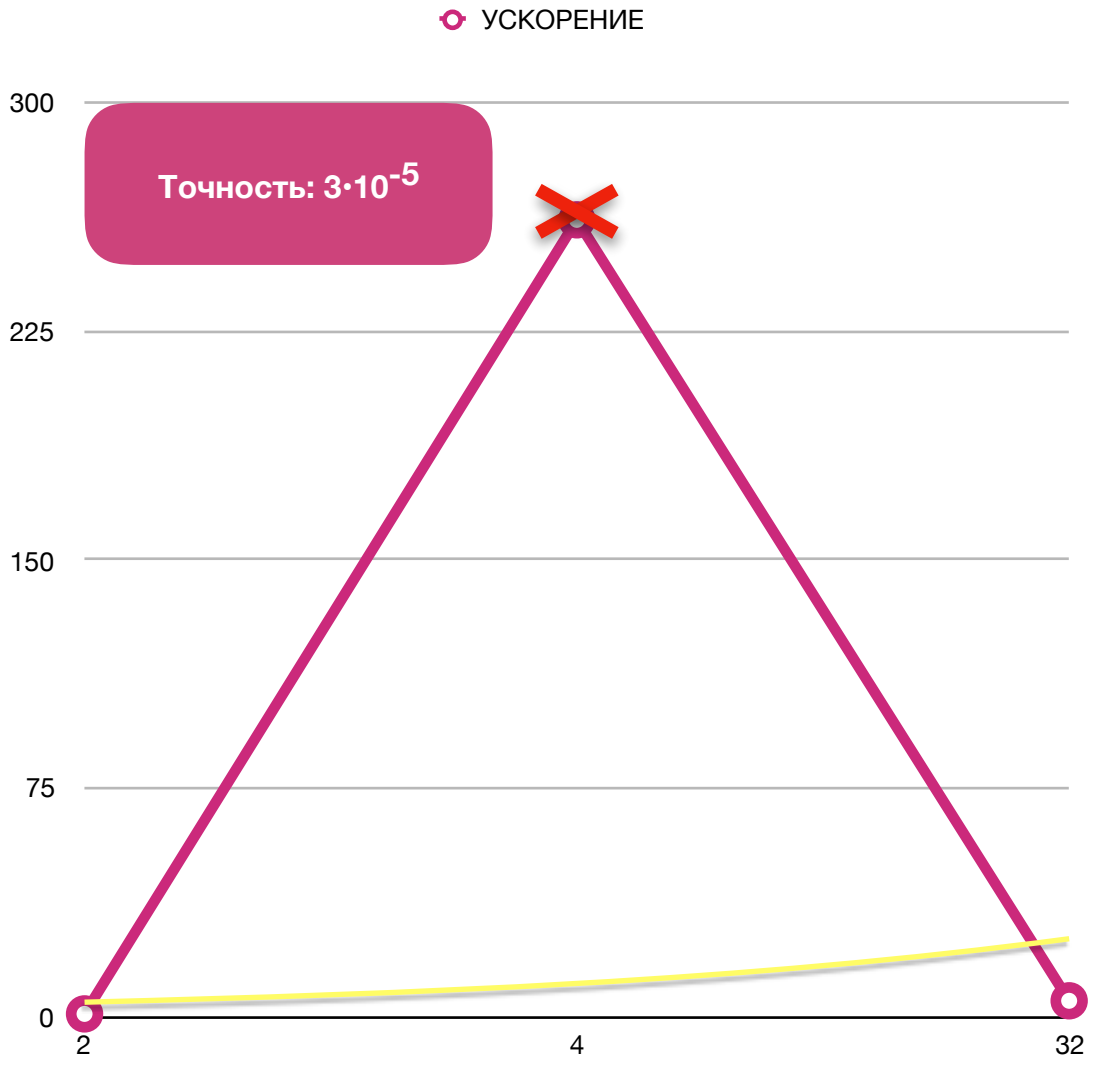
где T_1 — время работы программы на 1 MPI-процессе, а T_p — время работы программы на p MPI-процессах.

Выразим зависимость ускорения программы от количества используемых MPI-процессов на графиках.






УСКОРЕНИЕ
5
3,75
2,5
1,25
0
Точность: 5·10-6
2
4
32


УСКОРЕНИЕ
300
225
150
75
0
Точность: 3·10-5
2
4
32

В среднем, ускорение, получаемое относительно запуска на 2 процессах (мастер + 1 рабочий) и запуска на 32 процессах (мастер + 31 рабочий) приблизительно равно 5, что является весомой причиной для применения распараллеливания.