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

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

Численное интегрирование многомерных функций методом Монте-Карло

Вариант 15

Выполнила: Рощина Ксения Андреевна, 627 группа

Математическая постановка задачи

Функция $f(x,y,z) = \sqrt{y^2+z^2}$ непрерывна в ограниченной замкнутой области $G \subset R^3$
Требуется вычислить определенный интеграл

$$I = \iiint\limits_G \sqrt{y^2+z^2}\, dxdydz$$

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

Численный метод решения задачи

Используем метод Монте-Карло для численного интегрирования. Область G из постановки задачи ограничена параллелепипедом $\Pi : \begin{cases} 0 \le x \le 2 \\ -1 \le y \le 1 \\ -1 \le z \le 1 \end{cases}$

Рассмотрим функцию $F(x,y,z) = \begin{cases} \sqrt{y^2+z^2}, & (x,y,z) \in G \\ 0, & \text{иначе} \end{cases}$

Преобразуем искомый интеграл:

$$I = \iiint\limits_G f(x,y,z)\, dxdydz = \iiint\limits_\Pi F(x,y,z)\, dxdydz$$

Пусть p1(x1, y1, z1), p2(x2, y2, z2), ... — случайные точки, равномерно распределённые в П. 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 |П| — объём параллелепипеда П. $|\Pi| = (b1 - a1)(b2 - a2)(b3 - a3)$

Нахождение точного значения интеграла аналитически

$$\iiint\limits_G \sqrt{y^2+z^2}dxdydz = \iiint\limits_{G_1} \rho^2 dxd\rho d\phi = \int\limits_0^{2\pi} d\phi \int\limits_0^1 \rho^2 d\rho \int\limits_0^2 dx = \frac{4\pi}{3}$$

$$y = \rho \sin\phi$$
$$z = \rho \cos\phi$$
$$G_1 : \{(x,\rho,\phi) : 0 \le x \le 2, 0 \le \rho \le 2, 0 \le \phi \le 2\pi\}$$
$$|I| = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \sin\phi & \rho\cos\phi \\ 0 & \cos\phi & \rho\sin\phi \end{vmatrix} = |-\rho\sin^2\phi - \rho\cos^2\phi| = \rho$$

Краткое описание программной реализации

В моем варианте задания была реализована парадигма взаимодействия процессов мастер-рабочие.

Процесс с ранком 0 (мастер, функция **master**) генерирует случайные точки порциями по 100 (чанк) и асинхронно рассылает их рабочим в цикле. После Isend сразу вызывается Irecv на получение локальной суммы, посчитанной соответствующим процессом.

В основном цикле сначала идет проверка, закончился ли какой-либо из Irecv на получение суммы, если да, то идет подсчет и проверка на выполнение условий выхода (полученное приближение < ε). Если условия выхода выполняются, мастер выводит результаты и посылает MPI_Abort, завершая выполнение всех процессов. Если нужно продолжать вычисления, то происходит проверка, закончился ли предыдущий Isend чанка для данного процесса (сделано для перекрытия обменов вычислениями) и отправка нового чанка, если нужно. Затем снова вызывается Irecv на получение следующей суммы.

Если на данной итерации цикла еще ни один из Irecv не закончился, то осуществляется попытка отправки дополнительного чанка на один из процессов, для которого завершился Isend. Если такого процесса тоже нет, то генерируется новый чанк и кладется в буфер, реализуемый структурой **ChunkBuf**. Он используется при посылке чанков процессам, чтобы по возможности избежать ожидания генерации нового набора точек.

Процессы-рабочие (функция **slave**) принимают от мастера свой чанк для вычислений и сразу же ставят Irecv на получение следующей порции данных, после чего начинают вычисления. Таким образом появляется возможность перекрытия обменов с мастером вычислениями. Затем они проверяют, что мастер принял их предыдущую сумму и отправляют только что посчитанную через Isend. Так как у процесса есть запасной чанк, то он может продолжить свои вычисления, не дожидаясь мастера.

Исследование масштабируемости на Blue Gene/P и Polus

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка	Кол-во обработанных точек
$1.0 \cdot 10^{-4}$	2	0.126954	1	4.78028e-05	31200
	4	0.0209018	6.07	5.31765e-05	5800
	16	0.140019	0.9	2.10825e-05	39100
	64	0.191834	0.66	5.73759e-05	38400
$2.0 \cdot 10^{-5}$	2	0.312243	1	7.33339e-08	77000
	4	0.187607	1.66	1.86089e-05	53000
	16	0.168621	1.85	1.06656e-05	47400
	64	0.251706	1.24	1.94774e-05	68900
$0.8 \cdot 10^{-5}$	2	0.312122	1	7.33339e-08	77000
	4	0.48775	0.63	5.06753e-06	138300
	16	0.434776	0.71	2.04469e-06	122400
	64	0.485897	0.64	7.98423e-06	134000

Таблица 1. Таблица с результатами расчётов для системы Blue Gene/P

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка	Кол-во обработанных точек
$3.0 \cdot 10^{-5}$	2	0.0162122	1	2.7218e-05	74500
	4	0.0214739	0.75	2.7218e-05	74500
	16	0.0251895	0.64	2.7218e-05	74500
	64	0.0361725	0.44	7.23032e-06	81500
$5.0 \cdot 10^{-6}$	2	0.0207858	1	7.33339e-08	77000
	4	0.0208235	0.99	3.62912e-06	76800
	16	0.0948687	0.21	1.7702e-06	435000
	64	0.0391246	0.53	2.36102e-06	144300
$1.5 \cdot 10^{-6}$	2	0.0206593	1	1.37543e-06	89900
	4	0.125862	0.16	3.56181e-07	435100
	16	0.020409	1.01	9.88169e-07	86300
	64	0.114234	0.18	1.46737e-06	299500

Таблица 2. Таблица с результатами расчётов для системы Polus

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка	Кол-во обработанных точек
$1.5 \cdot 10^{-6}$	2	13.5338	1	7.33339e-08	77000
	4	4.07879	3.31	6.80729e-07	80000
	16	1.49993	9.02	2.006e-07	146600
	64	1.18021	11.46	4.95454e-07	181200

Таблица 3. Ускорение с искусственным увеличением объема вычислений на рабочих процессах (Blue Gene/P)

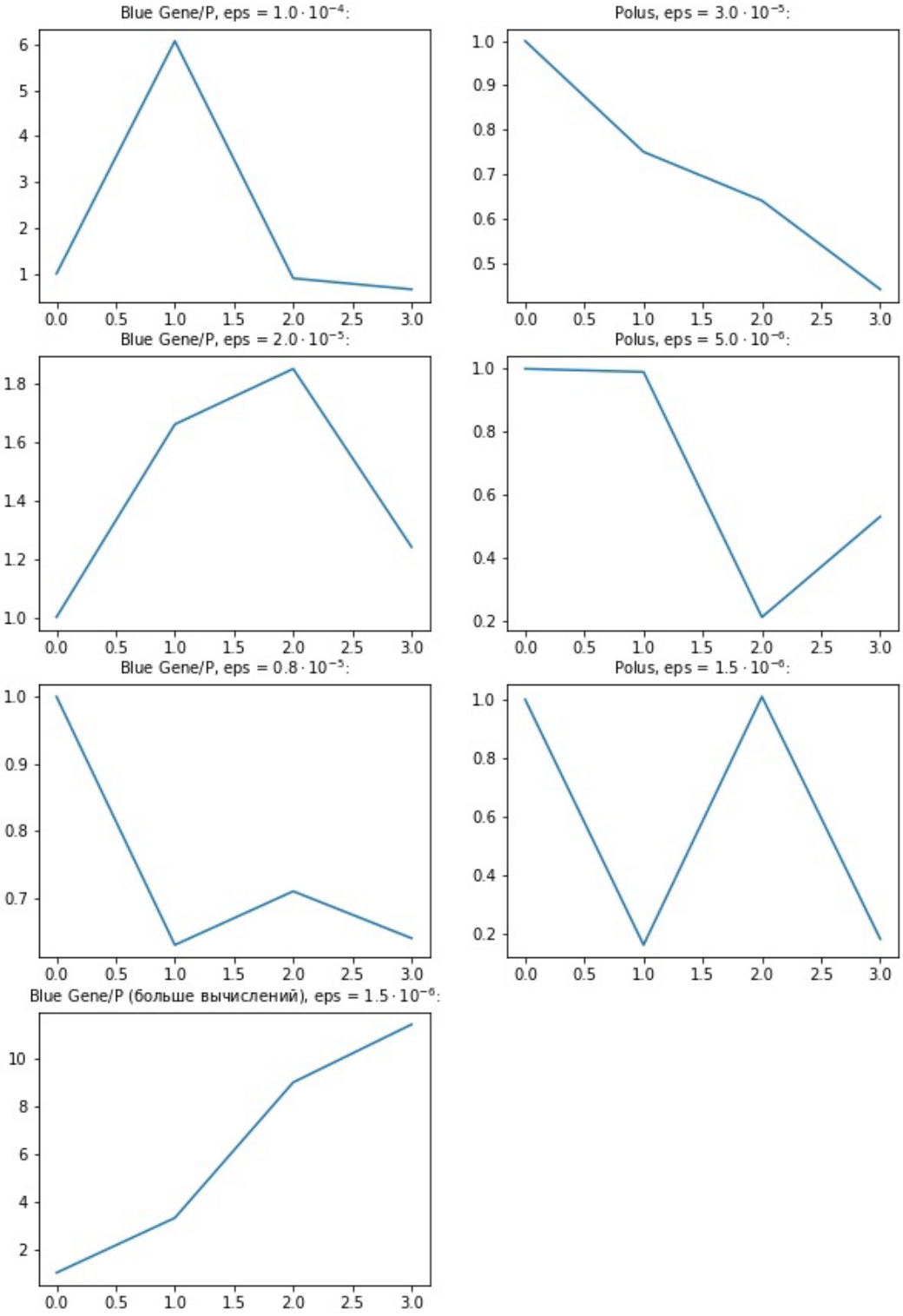

Рисунок 1. Графики ускорения

Выводы

Реализованная программа показала не лучшие результаты в масштабируемости. Из-за очень легковесных вычислений на процессах-рабочих бОльшая часть времени тратилась на ожидание завершения обменов.

Моя реализация мастер-процесса работает с каждым из подчиненных процессов по отдельности, что могло бы позволить избежать простоев при работе с процессами различной производительности (например, когда один из процессов слишком медленный и из-за него другим приходится простаивать в ожидании завершения коллективных операций). Но в ситуации, когда превалируют обмены, мастер может не успевать распределять нагрузку между некоторыми процессами, и они остаются незадействованными. Это послужило причиной остановки роста производительности при исполнении на большом числе процессов. Также в этой ситуации мастер не успевал заполнять буфер или отправлять дополнительные чанки процессам.

Поэтому, чтобы посмотреть, как поведет себя моя программа при более тяжелых вычислениях на рабочих процессах, я искусственно увеличила объем вычислений. В предыдущем разделе присутствуют дополнительные таблица и график для этого случая. На них видно, что программа хорошо ускоряется. Используя логи я убедилась, что мастер-процесс стабильно оперировал буфером и осуществлял дополнительные посылки чанков на процессы, занятые вычислениями. Все процессы-рабочие были загружены и не простаивали в ожидании завершения обменов.

Таким образом, я делаю вывод, что моя реализация взаимодействия мастер-рабочие отлично бы подошла для задач с более сложными вычислениями. Но так как в данном случае вычисления были очень простыми, программа показала слабое ускорение, а в некоторых случаях и замедление.

Хочется дополнительно отметить, что из-за элемента случайности в методе Монте-Карло количество точек, необходимых для получения нужной погрешности, могло очень сильно варьироваться в зависимости от порядка их обработки, что влияло на время выполнения программы.