Суперкомпьютерное моделирование и технологии

Отчёт

Численное интегрирование многомерных функций методом Монте-Карло.

Перминов Андрей Игоревич

Вариант 9

22.10.2021

Содержание

Математическая постановка задачи

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

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

В полученном варианте используется следующая функция и область: $f(x,y,z) = xy^2z^3$, G ограничена поверхностями $z = xy, y = x, x = 1, z = 0$.

Численный метод решения задачи

Пусть область G ограниченна параллелепипедом Π:

$$\Pi = \begin{cases} x_{min} \le x \le x_{max} \\ y_{min} \le y \le y_{max} \\ z_{min} \le z \le z_{max} \end{cases}$$

Рассмотрим функцию $F(x,y,z) = \begin{cases} f(x,y,z), & (x,y,z) \in G \\ 0, \text{иначе} \end{cases}$ и перепишем интеграл:

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

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.

$$|\Pi| = (x_{max} - x_{min}) \cdot (y_{max} - y_{min}) \cdot (z_{max} - z_{min})$$

Нахождение точного значения интеграла аналитически

Проанализировав ограниченную область, можно установить границы параллелепипеда: $0 \le x \le 1, 0 \le y \le x, 0 \le z \le xy$

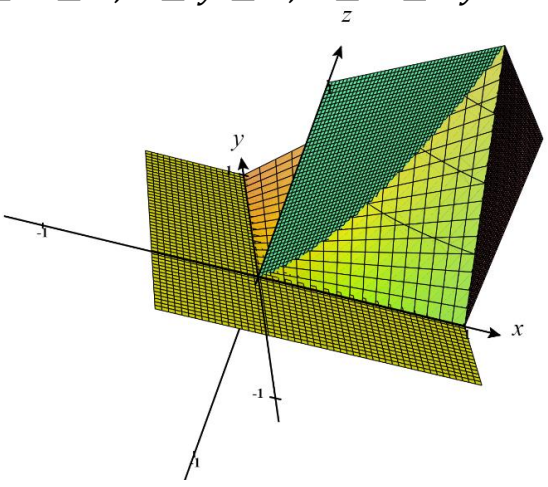

Рисунок 1. Область интегрирования

Посчитаем аналитически тройной интеграл:

$$I = \iiint\limits_G xy^2z^3\,dxdydz = \int\limits_0^1 \int\limits_0^x \int\limits_0^{xy} xy^2z^3\,dxdydz$$

$$\int\limits_0^1 \int\limits_0^x \int\limits_0^{xy} xy^2z^3\,dxdydz = \frac{1}{4}\int\limits_0^1 \int\limits_0^x x^5y^6\,dxdy = \frac{1}{4\cdot 7}\int\limits_0^1 x^{12}\,dx = \frac{1}{4\cdot 7\cdot 13} = \frac{1}{364}$$

Программная реализация

Реализована параллельная MPI-программа, принимающая от 0 до 9 аргументов (доступно при запуске с флагом `--help`):

- `-e` – необходимая точность вычисления интеграла;
- `-n` – количество точек, отдаваемых каждому процессу;
- `-d` – флаг отладки для вывода дополнительной информации;
- `-xmin` – нижняя граница изменения переменной x в параллелипипеде;
- `-xmax` – верхняя граница изменения переменной x в параллелипипеде;
- `-ymin` – нижняя граница изменения переменной y в параллелипипеде;
- `-ymax` – верхняя граница изменения переменной y в параллелипипеде;
- `-zmin` – нижняя граница изменения переменной z в параллелипипеде;
- `-zmax` – верхняя граница изменения переменной z в параллелипипеде;

При запуске программы все процессы считывают переданные аргументы, проверяя их на корректность, после чего в случае успешной проверки запускают основной цикл вычислений.

В качестве параллельной реализации используется парадигма «мастер–рабочие»: один из процессов («мастер») генерирует случайные точки и передаёт каждому из остальных процессов («рабочих») отдельный, предназначенный для него, набор сгенерированных случайных точек. Все процессы – рабочие вычисляют свою часть суммы. Затем вычисляется общая сумма с помощью операции редукции и итоговый интеграл.

Полученное значение интеграла сравнивается с заранее заданным эталоном и в случае достижения требуемой точности процессам рабочим передаётся флаг, сигнализирующий об окончании вычислений и завершении работы. В противном случае генерируется новая порция точек и процесс вычисления повторяется.

Дополнительно реализована «обычная» версия параллельной программы. В ней каждый процесс генерирует свои собственные точки независимо и затем считает общую сумму с помощью операции редукции. Все остальные действия (оценка достижения точности, передача сигнала завершения процессам и т.д. сделано полностью аналогично).

Результаты запусков программы на различных кластерах

Таблица 1. Результаты расчётов на Blue Gene/P (мастер-рабочий)

Точность (ε)	Число MPI процессов	Время работы программы (с)	Ускорение	Ошибка
1.0e-4	2	0.025181	1	5.94907e-5
	4	0.026026	0.968	5.12393e-5
	16	0.040886	0.616	5.37999e-5
	64	0.078336	0.321	5.72646e-5
2.0e-5	2	0.184750	1	1.24655e-5
	4	0.366492	0.504	1.20767e-5
	16	0.146726	1.259	1.12072e-5
	64	0.501224	0.369	1.18898e-5
8.0e-6	2	0.223830	1	4.46804e-6
	4	0.882050	0.254	4.47101e-6
	16	0.948025	0.236	4.83199e-6
	64	1.907600	0.117	4.72291e-6

Таблица 2. Результаты расчётов на Polus (мастер-рабочий)

Точность (ε)	Число MPI процессов	Время работы программы (с)	Ускорение	Ошибка
3.0e-5	2	0.015593	1	1.82846e-5
	4	0.010612	1.469	1.76684e-5
	16	0.016800	0.928	1.73457e-5
	64	0.028649	0.544	1.64332e-5
5.0e-6	2	0.050164	1	2.55487e-6
	4	0.180100	0.279	3.14434e-6
	16	0.167635	0.299	2.83515e-6
	64	0.321604	0.156	2.73644e-6
1.5e-6	2	0.104539	1	9.34065e-7
	4	0.691521	0.151	7.40848e-7
	16	1.413300	0.074	9.69010e-7
	64	1.487260	0.070	7.99376e-7

Запуски производились при N = 1000 (1000 точек на процесс). Время работы и ошибка вычислялись путём усреднения по 50 запускам.

Таблица 3. Результаты расчётов на Blue Gene/P (обычная схема)

Точность (ε)	Число MPI процессов	Время работы программы (с)	Ускорение	Ошибка
1.0e-4	1	0.013379	1	5.62939e-5
	4	0.006366	2.102	5.82478e-5
	16	0.002602	5.142	5.84753e-5
	64	0.001724	7.761	5.51759e-5
2.0e-5	1	0.117656	1	1.18561e-5
	4	0.048859	2.408	1.15702e-5
	16	0.011829	9.946	1.16796e-5
	64	0.009591	12.267	1.17729e-5
8.0e-6	1	0.353842	1	4.39475e-6
	4	0.446547	0.792	5.23333e-6
	16	0.054526	6.489	4.63279e-6
	64	0.035991	9.831	5.20942e-6

Таблица 4. Результаты расчётов на Polus (обычная схема)

Точность (ε)	Число MPI процессов	Время работы программы (с)	Ускорение	Ошибка
3.0e-5	1	0.007260	1	1.83298e-5
	4	0.004095	1.773	1.94462e-5
	16	0.002164	3.354	1.61581e-5
	64	0.002582	2.812	1.81731e-5
5.0e-6	1	0.072420	1	2.53375e-6
	4	0.035764	2.025	2.57642e-6
	16	0.014840	4.880	2.84998e-6
	64	0.013700	5.286	3.15469e-6
1.5e-6	1	0.768074	1	7.92767e-7
	4	0.424769	1.808	7.19423e-7
	16	0.053645	14.318	9.54345e-7
	64	0.061933	12.402	9.14685e-7

Запуски производились при N = 1000 (1000 точек на процесс). Время работы и ошибка вычислялись путём усреднения по 50 запускам.

Время запусков программы на различных кластерах

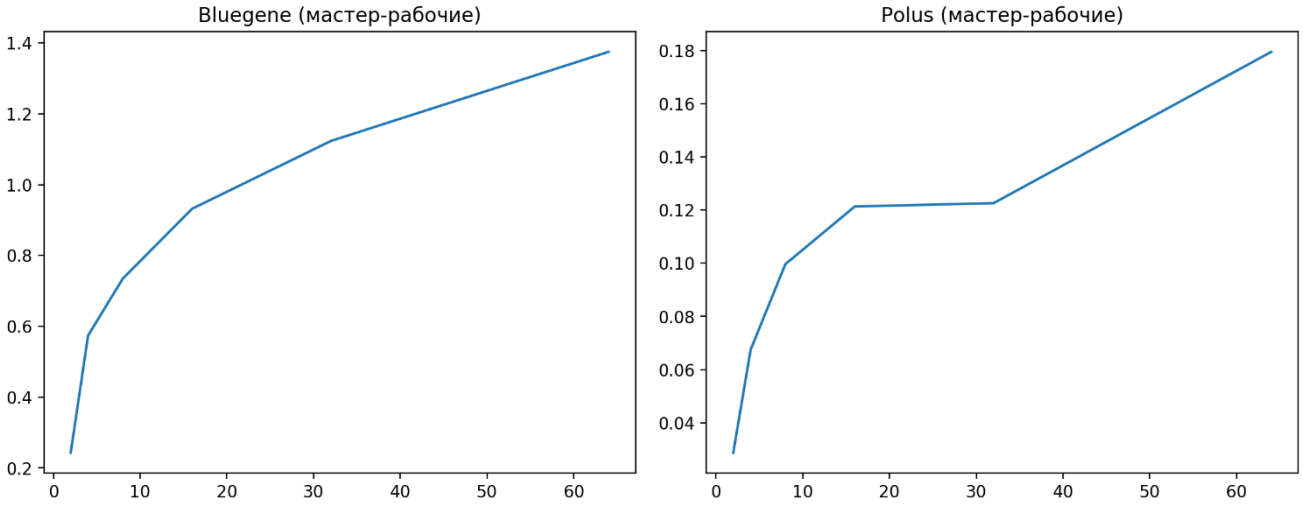

Рисунок 2. Время запусков парадигмы «мастер-рабочие»

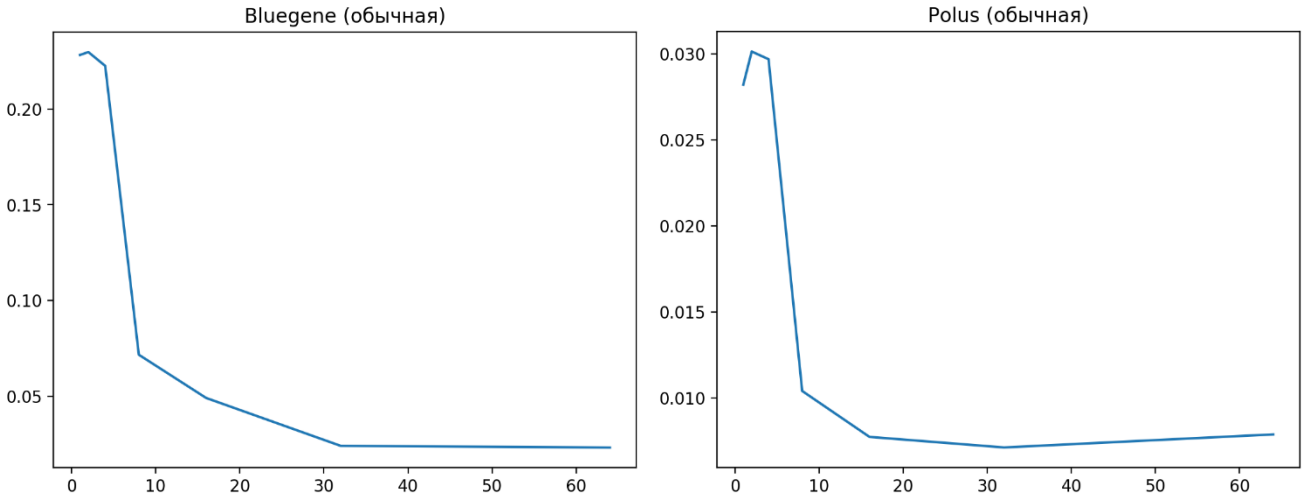

Рисунок 3. Время запусков обычной версии

Ускорение на различных кластерах

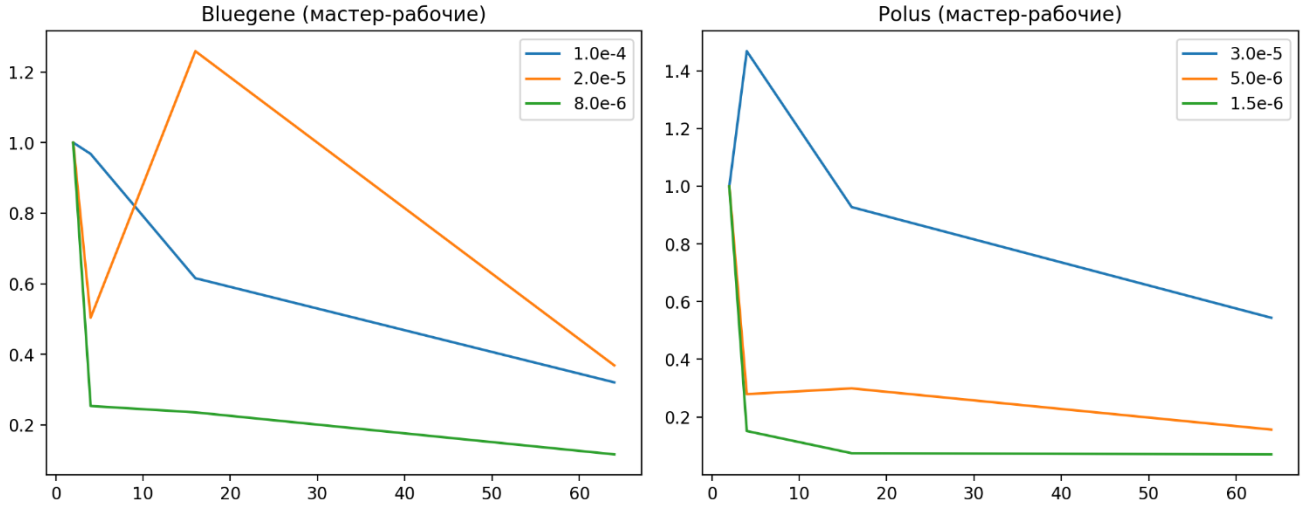

Рисунок 4. Ускорение парадигмы «мастер-рабочие»

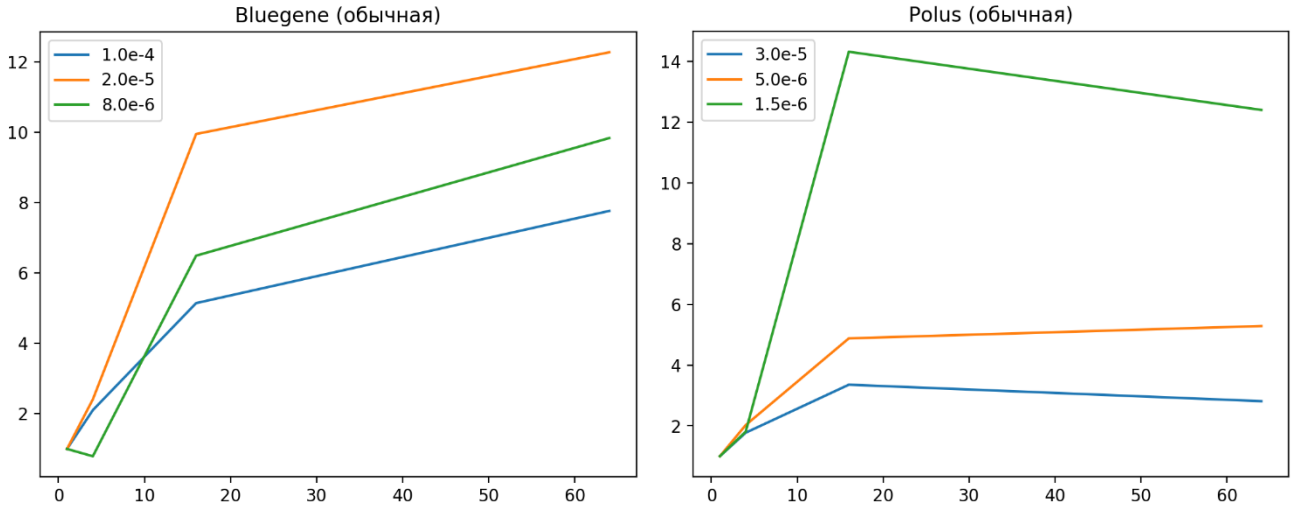

Рисунок 5. Ускорение обычной версии

Выводы

Как следует из приведённых выше таблиц, парадигма «мастер-рабочие» плохо подходит для решения задачи численного интегрирования методом Монте-Карло. Основная причина кроется в том, что самым узким местом парадигмы является рассылка данных другим процессам, в то время как операции редукции занимают гораздо меньшее время. Тем не менее, данная парадигма имеет право на жизнь, поскольку существует немалое количество задач, в которых вычисления, которые производятся каждым из «рабочих» занимают значительное время, превосходящее время на передачу данных для расчётов.