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

Суперкомпьютерное моделирование и технологии

Численное интегрирование многомерных функций методом Монте-Карло

Вариант 12

Зыкин Ярослав Ильич
628 группа

20 октября 2022 г.

Содержание

1 Постановка задачи

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

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

Требуется исследовать масштабируемость параллельной программы на параллельных вычислительных системах: IBM Polus, IBM Blue Gene/P.

Согласно предложенному варианту задания необходимо вычислить интеграл

$$I = \iiint\limits_G \exp^{x^2+y^2} * z^2 dxdydz$$

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

2 Аналитическое решение

$$\iiint\limits_G \exp^{x^2+y^2} * z^2 dxdydz = \begin{pmatrix} x = \rho\cos(\phi), y = \rho\sin(\phi), \\ x^2+y^2+z^2 = \rho^2+z^2 \leqslant 1 \end{pmatrix} =$$

$$= \iiint\limits_{G'} \exp^{\rho^2} * z * \rho d\rho d\phi dz = \int\limits_0^1 \exp^{\rho^2} \rho d\rho \int\limits_0^{\sqrt{1-\rho^2}} z dz \int\limits_0^{2\pi} d\phi =$$

$$2\pi \int\limits_0^1 \exp^{\rho^2} \rho \left.\frac{z^2}{2}\right|_0^{\sqrt{1-\rho^2}} d\rho = \pi \int\limits_0^1 \exp^{\rho^2}(1-\rho^2)\rho d\rho = \frac{\pi}{2}\int\limits_0^1 \exp^{\rho^2}(1-\rho^2)d\rho^2 =$$

$$= \begin{pmatrix} u = 1-\rho^2, du = -d\rho^2, \\ dv = \exp^{\rho^2} d\rho^2, v = \exp^{\rho^2} \end{pmatrix} = \frac{\pi}{2}\Big((1-\rho^2)\exp^{\rho^2}\Big|_0^1 - \int\limits_0^1 -\exp^{\rho^2} d\rho^2\Big) =$$

$$= \frac{\pi}{2}\Big(-1+\exp^{\rho^2}\Big|_0^1\Big) = \frac{\pi(\exp -2)}{2}$$

Таким образом, решением предложенного интеграла является число $\frac{\pi(\exp -2)}{2} \approx 1.1282744577$

3 Численный метод решения

Заданная область G ограничена параллелепипедом $\Pi : \begin{cases} -1 \leqslant x \leqslant 1 \\ -1 \leqslant y \leqslant 1 \\ 0 \leqslant z \leqslant 1 \end{cases}$

Рассмотрим функцию на области Π $F = \begin{cases} \exp^{x^2+y^2} * z^2 & (x,y,z) \in G \\ 0 & (x,y,z) \notin G \end{cases}$

Преобразуем искомый интеграл

$$I = \iiint\limits_{G} \exp^{x^2+y^2} * z^2 dxdydz = \iiint\limits_{\Pi} F(x,y,z)dxdydz$$

Пусть $p_1(x_1, y_1, z_1), p_2(x_2, y_2, z_2), \ldots$ – случайные равномерно распределённый в Π точки. В качестве приближённого значения интеграла воспользуемся выражением

$$I \approx |\Pi| \cdot \frac{1}{n} \sum_{i=1}^{n} F(p_i) \tag{1}$$

где $|\Pi|$ – объём параллелепипеда Π, $|\Pi| = (1-(-1)) * (1-(-1)) * (1-0) = 4$

4 Программная реализация

Для решения поставленной задачи была написана программа на языке C++ с использованием технологии параллельного программирования MPI.

В пердоставленном варианте предполагается решение задачи в парадигме «мастер - рабочие». Эта парадигма означает следующее:

1. «Мастер»-процесс (нулевой процесс в коммуникаторе) генерирует набор случайных точек всем процессам-«рабочим».
2. «Мастер»-процесс передаёт набор точек каждому «рабочему» процессу.
3. Процесс-«рабочий», получив набор точек, производит расчёт функции и суммирование результатов.
4. Закончив суммирование, процесс-«рабочий» передаёт полученный результат «мастер»-процессу.
5. «Мастер»-процесс с помощью полученных сумм считает новое приближённое значение интеграла по формуле 1.
6. Если полученный результат отличается от фактического (полученного аналитически) более, чем на ε, переходим к пункту 1.
7. Иначе процесс подсчёта значения интеграла считается законченным.

Разработанная программа принимает на вход параметр ε – необходимая точность расчёта интеграла.

Краткое описание работы параллельной программы

Логика работы программы разбивается на три части: работа в последовательном режиме, работа «мастер»-процесса в параллельном режиме и работа «рабочего» процесса в параллельном режиме.

Последовательная работа программы заключается в подсчёте значения по формуле 1 с добавлением на каждой новой итерации новой случайно сгенерированной точки.

«Мастер»-процесс случайным образом генерирует точки для всех «рабочих» процессов, передаёт их им, получает от них подсчитанную сумму и использует её для вычисления нового приближённого значения интеграла. Когда точность вычисления интеграла достигает необходимого заданного значения, происходит выход из цикла и отправка всем процессам-«рабочим» сообщения о завершении работы. В этот же момент происходит отсечка времени выполнения программы.

«Рабочие» процессы в цикле получают новую порцию точек от процесса «мастера», вычисляют значения функций, суммируют полученные значения и отправляют их «мастеру». Отсечка времени выполнения программы процесса-«рабочего» происходит после получения сообщения от «мастер»-процесса о завершении работы.

Итоговое время выполнения программы получается как максимум от времени выполнения всех процессов.

В программе используются следующие механизмы обмена сообщений и редукции MPI:

- MPI_Bcast – рассылка одно и того же сообщения всем процессам – с помощью этого обмена «мастер»-процесс передаёт «рабочим» процессам флаг остановки работы. Это необходимо для того, чтобы процессы-«рабочие» корректно завершались при выходе из цикла «мастер»-процесса.

- Связка MPI_Isend и MPI_Irecv – асинхронные (неблокирующие) отправка и получение сообщений – с помощью этих обменов «рабочие» получают очередной набор точек от «мастер»-процесса. Перед началом подсчёта суммы производится вызов фукнции MPI_Waitall, с помощью которой производится ожидание завершения всех начатых обменов (получение массива точек). В «мастер»-процессе в этом нет необходимости, потому что до синхронизации (операцией редукции) значения точек не изменяются.

- MPI_Reduce(MPI_SUM) – выполнение групповой операции поиска суммы – после подсчёта сумм на «рабочих» процессах с помощью этой операции производится получение и суммирование результатов в «мастер»-процессе для вычисления новой версии приближённого значения интеграла.

- MPI_Reduce(MPI_MAX) – выполнение групповой опреации поиска максимума – итоговое время программы вычисляется как максимальное время выполнения всех процессов.

Полный код программы доступен по ссылке monte-carlo

5 Полученные результаты и их анализ

В ходе выполнения практического задания были проведены запуски на суперкомпьютерах BlueGene и Polus.

На суперкомпьютере Polus эксперименты были проведены максимум на 32 процессах вместо 64, потому что, по информации от службы поддержки, на момент проведения экспериментов 64 процесса физически не были доступны.

Первой целью проведённых экспериментов было получение наиболее оптимального числа отправляемых «мастер»-процессом точек за один раз. Согласно полученным результатам для суперкомпьютере Polus это число равно 64 (см. Таблицы 5, 6, 7), для BlueGene – 8 (см. Таблицы 8, 9, 10). Это может быть связано с взаимоотношением трудоёмкости пересылок и трудоёмкости вычисления функции, а также с физическим устройством суперкомпьютеров и их сетей передачи данных.

В дальнейшем исследовании масштабируемости будут использоваться полученные оптимальные числа отправляемых за один раз точек. В результате экспериментов были получены (см. Таблицы 1, 2, Рисунки 1, 2) крайне низкие показатели ускорения и эффективности работы параллельной программы. Это связано с низкой вычислительной сложностью предложенной в варианте функции и с высокими накладным расходами по пересылке большого количества точек.

Этот вывод подтверждается и при сравнении работы параллельной программы в режиме «мастер-рабочие» с последовательной версией, время работы которой меньше на порядок (см. Таблицы 3, 4).

Точность	Число процессов	Время работы	Ускорение	Эффективность	Ошибка
$3.0 \cdot 10^{-5}$	2	0.0318	1	1	$1.7 \cdot 10^{-5}$
	4	0.0390	0.82	0.41	$1.5 \cdot 10^{-5}$
	16	0.0243	1.31	0.16	$0.6 \cdot 10^{-5}$
	32	0.0376	0.85	0.05	$0.7 \cdot 10^{-5}$
$5.0 \cdot 10^{-6}$	2	0.0991	1	1	$3.8 \cdot 10^{-6}$
	4	0.1126	0.88	0.44	$2.2 \cdot 10^{-6}$
	16	0.0276	3.59	0.45	$0.5 \cdot 10^{-6}$
	32	0.0393	2.52	0.16	$4.2 \cdot 10^{-6}$
$1.5 \cdot 10^{-6}$	2	0.1858	1	1	$0.5 \cdot 10^{-6}$
	4	0.1399	1.33	0.66	$0.5 \cdot 10^{-6}$
	16	0.0407	4.57	0.57	$0.5 \cdot 10^{-6}$
	32	0.1780	1.04	0.07	$0.4 \cdot 10^{-6}$

Таблица 1: Таблица результатов для суперкомпьютера Polus.

Точность	Число процессов	Время работы	Ускорение	Эффективность	Ошибка
$1.0 \cdot 10^{-4}$	2	0.0054	1	1	$2.8 \cdot 10^{-5}$
	4	0.0118	0.46	0.229	$2.7 \cdot 10^{-5}$
	16	0.0096	0.56	0.070	$2.7 \cdot 10^{-5}$
	64	0.2768	0.02	0.001	$0.8 \cdot 10^{-5}$
$2.0 \cdot 10^{-5}$	2	0.1673	1	1	$1.5 \cdot 10^{-5}$
	4	0.1330	1.26	0.63	$0.7 \cdot 10^{-5}$
	16	0.2774	0.60	0.08	$0.6 \cdot 10^{-5}$
	32	0.2766	0.60	0.04	$0.8 \cdot 10^{-5}$
$2.0 \cdot 10^{-5}$	2	0.1743	1	1	$7.5 \cdot 10^{-6}$
	4	0.1330	1.31	0.66	$7.5 \cdot 10^{-6}$
	16	0.2774	0.63	0.08	$5.7 \cdot 10^{-6}$
	32	0.3756	0.46	0.03	$0.3 \cdot 10^{-6}$

Таблица 2: Таблица результатов для суперкомпьютера BlueGene/P.

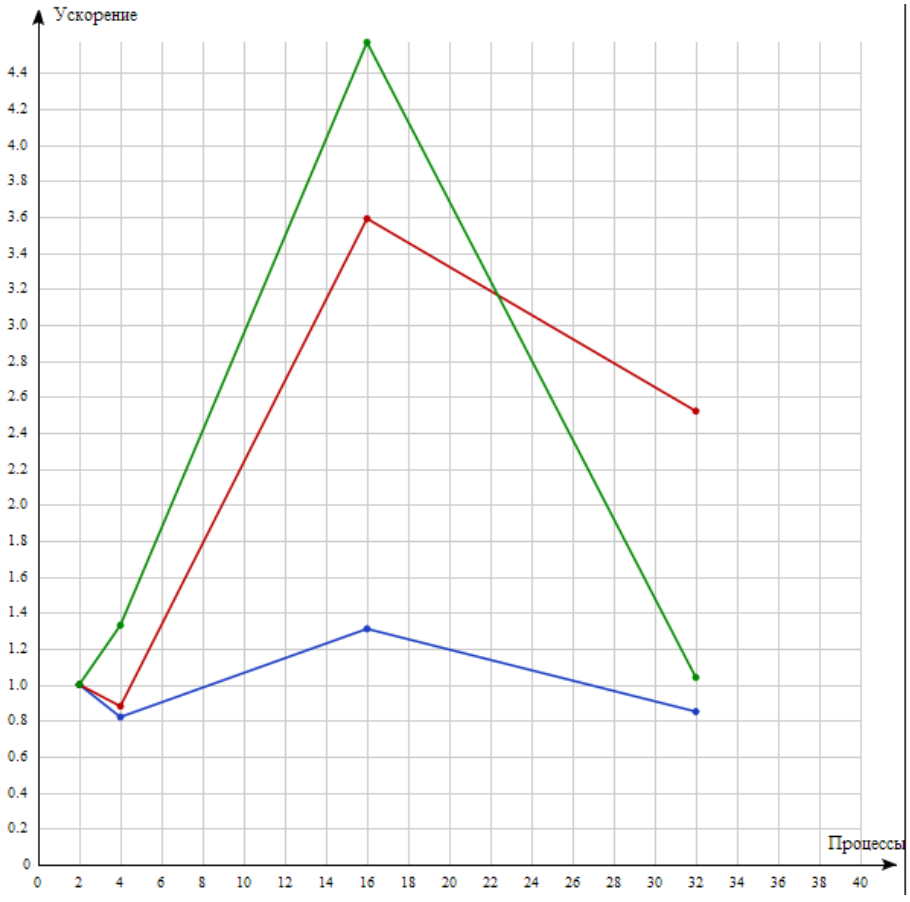

Рис. 1: График ускорения для с/к Polus ($3.0 \cdot 10^{-5}$, $5.0 \cdot 10^{-6}$, $1.5 \cdot 10^{-6}$)

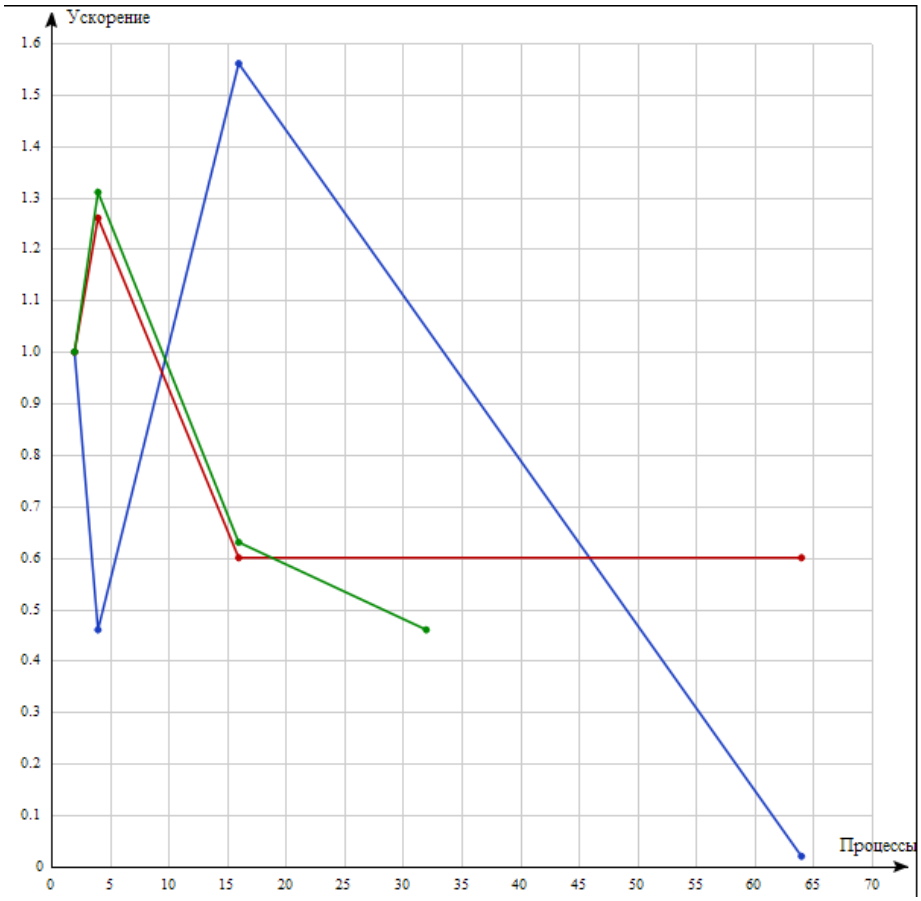

Рис. 2: График ускорения для с/к Polus ($1.0 \cdot 10^{-4}$, $2.0 \cdot 10^{-5}$, $2.0 \cdot 10^{-5}$)

6 Приложения

Точность	Время работы (последовательное)	Время работы (лучшее параллельное)	Ошибка
$3.0 \cdot 10^{-5}$	0.0009	0.0027	$0.8 \cdot 10^{-5}$
$5.0 \cdot 10^{-6}$	0.0032	0.0276	$2.5 \cdot 10^{-6}$
$1.5 \cdot 10^{-6}$	0.0076	0.0407	$1.5 \cdot 10^{-6}$

Таблица 3: Таблица результатов для Polus для последовательных запусков.

Точность	Время работы (последовательное)	Время работы (лучшее параллельное)	Ошибка
$1.0 \cdot 10^{-4}$	0.0013	0.0054	$9.7 \cdot 10^{-5}$
$2.0 \cdot 10^{-5}$	0.0014	0.0188	$8.2 \cdot 10^{-6}$
$0.8 \cdot 10^{-5}$	0.0056	0.1330	$7.9 \cdot 10^{-6}$

Таблица 4: Таблица результатов для BlueGene/P для последовательных запусков.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.0177	$0.8 \cdot 10^{-5}$
	8	0.0036	$2.8 \cdot 10^{-5}$
	64	0.0318	$1.7 \cdot 10^{-5}$
	256	0.0434	$1.5 \cdot 10^{-5}$
	1024	0.0994	$2.8 \cdot 10^{-5}$
4	1	0.0082	$0.8 \cdot 10^{-5}$
	8	0.0628	$2.3 \cdot 10^{-5}$
	64	0.0390	$1.5 \cdot 10^{-5}$
	256	0.0339	$1.5 \cdot 10^{-5}$
	1024	0.0814	$2.8 \cdot 10^{-5}$
16	1	0.0027	$0.8 \cdot 10^{-5}$
	8	0.0158	$1.9 \cdot 10^{-5}$
	64	0.0243	$0.6 \cdot 10^{-5}$
	256	0.0270	$2.2 \cdot 10^{-5}$
	1024	0.0497	$0.7 \cdot 10^{-5}$
32	1	0.0146	$2.7 \cdot 10^{-5}$
	8	0.0223	$1.3 \cdot 10^{-5}$
	64	0.0376	$0.7 \cdot 10^{-5}$
	256	0.0663	$0.8 \cdot 10^{-5}$
	1024	0.3185	$0.7 \cdot 10^{-5}$

Таблица 5: Таблица результатов для Polus для точности $3.0 \cdot 10^{-5}$.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.0637	$2.6 \cdot 10^{-6}$
	8	0.3822	$1.1 \cdot 10^{-6}$
	64	0.0991	$3.8 \cdot 10^{-6}$
	256	0.1517	$2.6 \cdot 10^{-6}$
	1024	0.1142	$2.3 \cdot 10^{-6}$
4	1	0.0309	$2.6 \cdot 10^{-6}$
	8	0.2104	$1.1 \cdot 10^{-6}$
	64	0.1126	$2.2 \cdot 10^{-6}$
	256	0.1287	$2.3 \cdot 10^{-6}$
	1024	0.1157	$2.3 \cdot 10^{-6}$
16	1	0.1086	$4.4 \cdot 10^{-6}$
	8	0.0571	$3.4 \cdot 10^{-6}$
	64	0.0276	$0.5 \cdot 10^{-6}$
	256	0.3275	$2.2 \cdot 10^{-6}$
	1024	0.3302	$2.2 \cdot 10^{-6}$
32	1	0.2149	$3.3 \cdot 10^{-6}$
	8	0.0522	$3.3 \cdot 10^{-6}$
	64	0.0393	$4.2 \cdot 10^{-6}$
	256	0.3369	$3.8 \cdot 10^{-6}$
	1024	1.1589	$1.0 \cdot 10^{-6}$

Таблица 6: Таблица результатов для Polus для точности $5.0 \cdot 10^{-6}$.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.9032	$1.5 \cdot 10^{-6}$
	8	0.3822	$1.1 \cdot 10^{-6}$
	64	0.1858	$0.5 \cdot 10^{-6}$
	256	0.5564	$1.1 \cdot 10^{-6}$
	1024	0.7672	$1.1 \cdot 10^{-6}$
4	1	0.8720	$0.6 \cdot 10^{-6}$
	8	0.1831	$1.1 \cdot 10^{-6}$
	64	0.1399	$0.5 \cdot 10^{-6}$
	256	1.1698	$0.2 \cdot 10^{-6}$
	1024	3.8224	$0.6 \cdot 10^{-6}$
16	1	0.1804	$1.0 \cdot 10^{-6}$
	8	0.0721	$0.6 \cdot 10^{-6}$
	64	0.0407	$0.5 \cdot 10^{-6}$
	256	0.2777	$0.9 \cdot 10^{-6}$
	1024	1.2274	$0.5 \cdot 10^{-6}$
32	1	0.2095	$0.7 \cdot 10^{-7}$
	8	0.3253	$0.4 \cdot 10^{-6}$
	64	0.2078	$0.4 \cdot 10^{-6}$
	256	1.1016	$0.9 \cdot 10^{-6}$
	1024	1.0893	$1.0 \cdot 10^{-6}$

Таблица 7: Таблица результатов для Polus для точности $1.5 \cdot 10^{-6}$.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.0185	$9.7 \cdot 10^{-5}$
	8	0.0054	$2.8 \cdot 10^{-5}$
	64	0.0803	$9.1 \cdot 10^{-5}$
	256	0.0752	$4.9 \cdot 10^{-5}$
	1024	0.1862	$4.0 \cdot 10^{-5}$
4	1	0.0137	$0.8 \cdot 10^{-5}$
	8	0.0118	$2.7 \cdot 10^{-5}$
	64	0.0690	$8.8 \cdot 10^{-5}$
	256	0.0637	$4.9 \cdot 10^{-5}$
	1024	0.1596	$4.0 \cdot 10^{-5}$
16	1	0.0101	$0.8 \cdot 10^{-5}$
	8	0.0096	$2.7 \cdot 10^{-5}$
	64	0.1586	$0.6 \cdot 10^{-5}$
	256	0.1541	$2.2 \cdot 10^{-5}$
	1024	0.4118	$1.1 \cdot 10^{-5}$
64	1	0.0444	$1.7 \cdot 10^{-5}$
	8	0.2768	$0.8 \cdot 10^{-5}$
	64	0.2169	$1.9 \cdot 10^{-5}$
	256	0.2211	$8.5 \cdot 10^{-5}$
	1024	0.4063	$0.2 \cdot 10^{-5}$

Таблица 8: Таблица результатов для BlueGene/P для точности $1.0 \cdot 10^{-4}$.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.0188	$0.8 \cdot 10^{-5}$
	8	0.1673	$1.5 \cdot 10^{-5}$
	64	0.0806	$1.5 \cdot 10^{-5}$
	256	0.1927	$1.3 \cdot 10^{-5}$
	1024	0.2553	$0.3 \cdot 10^{-5}$
4	1	0.0137	$0.8 \cdot 10^{-5}$
	8	0.1330	$0.7 \cdot 10^{-5}$
	64	0.1801	$0.6 \cdot 10^{-5}$
	256	0.1857	$1.2 \cdot 10^{-5}$
	1024	0.2188	$0.3 \cdot 10^{-5}$
16	1	0.0102	$0.8 \cdot 10^{-5}$
	8	0.2774	$0.6 \cdot 10^{-5}$
	64	0.1586	$0.6 \cdot 10^{-5}$
	256	0.4159	$1.1 \cdot 10^{-5}$
	1024	0.4118	$1.1 \cdot 10^{-5}$
64	1	0.4443	$1.7 \cdot 10^{-5}$
	8	0.2766	$0.8 \cdot 10^{-5}$
	64	0.2169	$1.9 \cdot 10^{-5}$
	256	0.4076	$0.2 \cdot 10^{-5}$
	1024	1.4063	$0.2 \cdot 10^{-5}$

Таблица 9: Таблица результатов для BlueGene/P для точности $2.0 \cdot 10^{-5}$.

Число процессов	Кол-во передаваемых точек	Время работы	Ошибка
2	1	0.7609	$7.9 \cdot 10^{-6}$
	8	0.1743	$7.5 \cdot 10^{-6}$
	64	0.2193	$5.9 \cdot 10^{-6}$
	256	0.2583	$2.6 \cdot 10^{-6}$
	1024	0.2554	$3.3 \cdot 10^{-6}$
4	1	0.6272	$6.0 \cdot 10^{-6}$
	8	0.1330	$7.5 \cdot 10^{-6}$
	64	0.1801	$5.9 \cdot 10^{-6}$
	256	0.2268	$3.3 \cdot 10^{-6}$
	1024	0.2188	$3.3 \cdot 10^{-6}$
16	1	0.4799	$6.0 \cdot 10^{-6}$
	8	0.2774	$5.7 \cdot 10^{-6}$
	64	0.1585	$5.9 \cdot 10^{-6}$
	256	1.1260	$7.1 \cdot 10^{-6}$
	1024	2.1877	$2.2 \cdot 10^{-6}$
64	1		
	8	0.3756	$0.3 \cdot 10^{-6}$
	64	0.4621	$2.1 \cdot 10^{-6}$
	256	0.4076	$2.1 \cdot 10^{-6}$
	1024	0.4063	$1.0 \cdot 10^{-6}$

Таблица 10: Таблица результатов для BlueGene/P для точности $0.8 \cdot 10^{-5}$.