Московский государственный университет имени М. В. Ломоносова

Задание №2

Решение краевой задачи для уравнения Пуассона с потенциалом методом конечных разностей на суперкомпьютерах Blue Gene/P и Polus.

Факультет: Вычислительной математики и кибернетики

Кафедра: Математической кибернетики

Группа: 618/1

Студент: Комаров Ярослав Борисович

Предмет: Суперкомпьютерное моделирование и технологии

Вариант: 1

Москва, 2019

1 Формулировка задания

Требуется методом конечных разностей приближенно решить краевую задачу для уравнения Пуассона с потенциалом в прямоугольной области. Для этого предлагается разработать параллельный код программы, вычисляющий приближенное решение разностной схемы методом наименьших невязок, а затем исследовать точность и масштабируемость метода на ПВС МГУ Blue Gene/P и Polus, выполнив расчеты на сетках разного размера с разным числом доступных процессов.

2 Математическая постановка задачи.

Задача практикума заключается в восстановлении известной гладкой функции $u(x, y)$ по ее образу $F(x, y) = -\Delta u + q(x, y)u$, где

$$\Delta u = \frac{\partial}{\partial x}(k(x,y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(k(x,y)\frac{\partial u}{\partial y}),$$

и ее граничным значениям. Из условий варианта:

$$u(x, y) = exp(1 - (x + y)^2), \Pi = [-1, 2] \times [-2, 2],$$

$$k(x, y) = 4 + x,$$

$$q(x, y) = (x + y)_+^2;$$

граничные условия для всех сторон являются граничными условиями Дирихле.

С помощью этих данных можно вычислить функцию F:

$$F(x, y) = -2exp(1 - (x + y)^2)(4(x + 4)y^2 + 8x(x + 4)y + x(4x(x + 4) - 3) - y - 8 + (x + y)_+^2))$$

Далее в расчетной области Π определяется равномерная прямоугольная сетка размера $N \times N$ с шагами h_1 и h_2 и рассматривается линейное пространство функций, заданных на ней. Обозначим через w_{ij} значение сеточной функции w в узле сетки (x_i, y_j). Будем считать, что в этом пространстве задано скалярное произведение и евклидова норма:

$$[u, v] = \sum_{i=0}^{N} h_1 \sum_{j=0}^{N} h_2 p_{ij} u_{ij} v_{ij}, \|u\|_E = \sqrt{[u, u]}.$$

Весовая функция $p_{ij} = p(x_i)p(y_j)$, где

$$p(x_i) = \left[\begin{array}{ll} 1, & 1 \le i \le N-1 \\ 1/2, & i = 0, i = N \end{array} \right.$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вида:

$$Aw = B,$$

где A – оператор, действующий в пространстве сеточных функций, а B – известная правая часть. Аппроксимируем уравнение $F(x, y) = -\Delta u + q(x, y)u$ во всех внутренних точках разностным уравнением $-\Delta_h w_{ij} + q_{ij} w_{ij} = F_{ij}$, где разностный оператор Лапласа:

$$\Delta_h w_{ij} = \frac{1}{h_1}\Big(k(x_i + 0.5h_1, y_j)\frac{w_{i+1j} - w_{ij}}{h_1} - k(x_i - 0.5h_1, y_j)\frac{w_{ij} - w_{i-1j}}{h_1}\Big) + \frac{1}{h_2}\Big(k(x_i, y_j + 0.5h_2)\frac{w_{ij+1} - w_{ij}}{h_2} - k(x_i, y_j - 0.5h_2)\frac{w_{ij} - w_{ij-1}}{h_2}\Big)$$

Краевые условия первого рода аппроксимируются точно равенством $w_{ij} = u(x_i, y_j)$, а значит эти переменные исключаются из разностной схемы, а соответствующие им узлы - из расчетной сетки. В результате получается система уравнений с неизвестными w_{ij}, которая решается методом наименьших невязок.

Этот метод позволяет получить последовательность сеточных функций $w^{(k)}$, сходящуюся по норме пространства к решению разностной схемы. При этом начальное приближение $w^{(0)}$ можно выбрать любым, например, равным нулю во всех точках расчетной сетки.

Метод является одношаговым:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \tau_{k+1} r_{ij}^{(k)},$$

где невязка $r^{(k)} = Aw^{(k)} - B$, итерационный параметр

$$\tau_{k+1} = \frac{[Ar^{(k)}, r^{(k)}]}{\|Ar^{(k)}\|_E^2}$$

В качестве критерия остановки итерационного процесса возьмем следующее неравенство:

$$\|w^{(k+1)} - w^{(k)}\|_E < 0.000001$$

3 Ресурс параллелизма.

Значения невязки в каждой точке сетки на каждой итерации независимы друг от друга, следовательно допустимо их параллельное вычисление. Далее для получения итерационного параметра $\tau^{(k+1)}$ на каждом шаге необходимо вычислить $Ar^{(k)}$, что также можно делать параллельно. Реализовать это можно, разделив сетку на ряды и распределив их между процессами так, чтобы каждый ряд обрабатывался одним MPI процессом. При использовании OpenMP вычисления можно ускорить, разделив циклы с генерацией массивов данных, так как вычисляемые значения для ряда не зависят от данных, полученных в текущем цикле.

4 Реализация программы

Была реализованна программа par. Размер расчетной сетки задается в самой программе, а количество доступных процессов она получает средствами MPI. Все узлы сетки распределялись по процессам последовательно так, чтобы каждый из них обрабатывал одинаковое количество узлов (возможно отклонение в 1 узел при отсутствии делимости нацело). Необходимые для работы этих процессов начальные данные передавались главным процессом в начале программы с помощью средств MPI - Send и Recv. Дальнейшее общение процессов между собой происходило посредством Allgatherv, Allreduce и Bcast. Также циклы, возникающие при применении оператора A к вектору и вычислении скалярного произведения, можно было, в силу независимости вычислений от данных полученных на предыдущих итерациях этих циклов, разделить с помощью OpenMP. В задании требовалось произвести вычисления на сетке размера 500 и сетке размера 1000 на устройствах Blue Gene/P и Polus, используя соответсвенно 128/256 и 10/20/40 процессов. Для Blue Gene/P также было необходимо использовать гибридную версию программы, которая содержала бы в себе не только MPI, но и OpenMP.

5 Параметры запуска

Компиляция производилась с помощью следующих команд:

BlueGene/P	**mpixlcxx par.cpp –o par**
Polus	**mpixlC Par.cpp -o par**

Запуск (помещение в очередь на выполнение) — с помощью следующих команд:

BlueGene/P	**mpisubmit.bg -n *proc1* -w 00:15:00 par**
Polus	**mpisubmit.pl -p *proc2* -w 00:15 ./par**

где $proc1 \in \{1, 128, 256\}$, а $proc2 \in \{1, 10, 20, 40\}$.

6 Результаты

Таблица 1: Таблица с результатами расчетов на ПВС IBM Blue Gene/P (MPI)

Число процессоров N_p	Число точек сетки N^2	Время решения T	Ускорение S
128	500×500	401.96	10.76
256	500×500	222.381	19.46
128	1000×1000	819.514668	9.28
256	1000×1000	410.632916	18.54

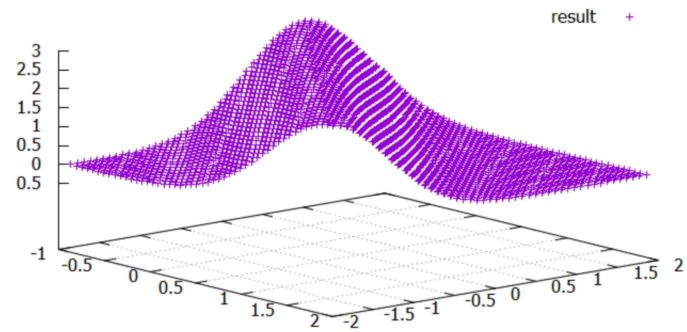
Таблица 2: Таблица с результатами расчетов на ПВС IBM Blue Gene/P (MPI/OpenMP)

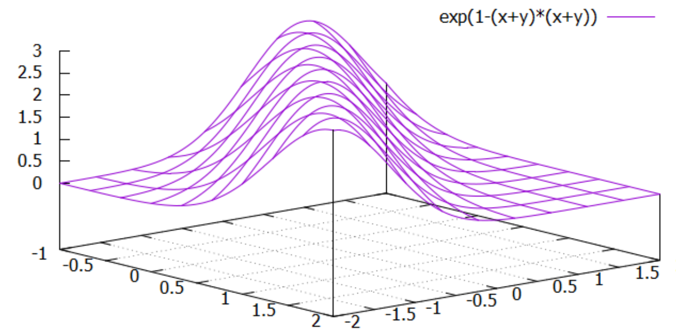
Число процессоров N_p	Число точек сетки N^2	Время решения T	Ускорение S
128	500×500	224.8962	19.24
256	500×500	138.675121	31.20
128	1000×1000	692.644114	10.99
256	1000×1000	373.825564	20.36

Таблица 3: Таблица с результатами расчетов на ПВС IBM Polus (MPI)

Число процессоров N_p	Число точек сетки N^2	Время решения T	Ускорение S
1	500×500	4328.003162	1
10	500×500	2881.910035	1.50
20	500×500	1770.249821	2.44
40	500×500	861.510442	5.02
1	1000×1000	7612.633169	1
10	1000×1000	3820.756881	1.99
20	1000×1000	1810.219586	4.20
40	1000×1000	1455.343334	5.23

Как видно из таблиц, благодаря огромному ресурсу параллелизма, имея достаточное количество процессоров, время выполнения задачи можно сократить в десятки раз. В качестве иллюстрации корректности работы программы приведем также график приближенного решения, полученного программой на сетке небольшого размера, и график функции, полученный с помощью утилиты gnuplot.





7 Список литературы

1. А.Н. Тихонов, А.А. Самарский. Уравнения математической физики. М. Изд. "Наука". 1977.

2. А.Н. Самарский, А.В. Гулин. Численные методы математической физики. М. Изд. "Научный мир". 2003.

3. Г.И. Марчук. Методы вычислительной математики. М. Изд. "Наука". 1989.

4. В.А. Ильин, Г.Д. Ким. Линейная алгебра и аналитическая геометрия. Изд. Московского университета. 2002.

5. IBM BlueGene/P, IBM Polus — http://hpc.cmc.msu.ru