

Отчет о выполнении MPI+OpenMP части задания по курсу "Суперкомпьютерное моделирование и технологии"

Мозговых Василий, 601 группа

Математическая постановка задачи

В треугольнике $D = \{(x, y) \mid y > 0, -x + 3y - 12 < 0, x + 3y - 12 < 0\}$, ограниченной кусочно-гладким контуром γ , рассматривается дифференциальное уравнение Пуассона

$$-\Delta u \equiv -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 1. \quad (1)$$

Для выделения единственного решения уравнение дополняется граничным условием Дирихле

$$u(x, y) = 0, \quad (x, y) \in \gamma. \quad (2)$$

Требуется найти функцию $u(x, y)$, удовлетворяющую уравнению (1) в области D и краевому условию (2) на ее границе.

Численный метод решения задачи

Для нахождения приближенного решения задачи (1), (2) используется метод фиктивных областей.

Метод фиктивных областей

Вводится прямоугольник $\Pi = \{(x, y) \mid -3,2 < x < 3,2, -0,2 < y < 4,2\}$ и его граница Γ . Далее фиксируется малое $\varepsilon > 0$.

В прямоугольнике Π рассматривается следующая задача Дирихле

$$-\frac{\partial}{\partial x} \left(k(x, y) \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y} \left(k(x, y) \frac{\partial v}{\partial y} \right) = F(x, y), \quad (x, y) \in \Pi \setminus \gamma, \quad (3)$$

с краевым условием

$$v(x, y) = 0, \quad (x, y) \in \Gamma,$$

где коэффициент

$$k(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 1/\varepsilon, & (x, y) \in \Pi \setminus \overline{D}, \end{cases}$$

и правая часть уравнения

$$F(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 0, & (x, y) \in \Pi \setminus \overline{D}. \end{cases}$$

Требуется найти непрерывную в $\bar{\Pi}$ функцию $v(x, y)$, удовлетворяющую дифференциальному уравнению задачи (3) всюду в $\Pi \setminus \gamma$, равную нулю на границе Γ прямоугольника, и такую, чтобы вектор потока

$$W(x, y) = -k(x, y) \left(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \right)$$

имел непрерывную нормальную компоненту на общей части границы области D и прямоугольника Π .

Известно, что функция $v(x, y)$ равномерно приближает решение $u(x, y)$ задачи (1), (2) в области D , а именно

$$\max_{(x,y) \in \bar{D}} |v(x, y) - u(x, y)| < C\varepsilon, \quad C > 0.$$

Это обстоятельство позволяет строить разностные схемы в прямоугольнике Π вместо исходной области D .

Разностная схема

Краевая задача для уравнения (3) решается численно методом конечных разностей. В замыкании прямоугольника $\bar{\Pi}$ определяется равномерная сетка

$$\bar{\omega}_h = \{x_i = -3, 2 + ih_1, i = 0, \dots, M, \quad y_j = -0, 2 + jh_2, j = 0, \dots, N\},$$

где $h_1 = 6, 4/M$, $h_2 = 4, 4/N$, $h = \max(h_1, h_2)$. Через ω_h обозначим множество узлов сетки, которые не лежат на границе Γ .

Рассмотрим линейное пространство H функций, заданных на сетке ω_h . Обозначим через w_{ij} значение сеточной функции $w \in H$ в узле сетки $(x_i, y_j) \in \omega_h$. Будем считать, что в пространстве H задано скалярное произведение и L_2 норма

$$(u, v) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} u_{ij} v_{ij} h_1 h_2, \quad \|u\|_{L_2} = \sqrt{(u, u)}.$$

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

$$Aw = B,$$

где $A : H \rightarrow H$ - оператор, действующий в пространстве сеточных функций, $B \in H$ - известная правая часть.

Дифференциальное уравнение задачи (3) во всех внутренних точках аппроксимируется разностным уравнением

$$-\frac{1}{h_1} \left(a_{i+1j} \frac{w_{i+1j} - w_{ij}}{h_1} - a_{ij} \frac{w_{ij} - w_{i-1j}}{h_1} \right) - \frac{1}{h_2} \left(b_{ij+1} \frac{w_{ij+1} - w_{ij}}{h_2} - b_{ij} \frac{w_{ij} - w_{ij-1}}{h_2} \right) = F_{ij},$$

$$i = 1, \dots, M-1, \quad j = 1, \dots, N-1,$$

в котором коэффициенты

$$a_{ij} = \frac{1}{h_2} \int_{y_{j-0,5}}^{y_{j+0,5}} k(x_{i-0,5}, \eta) d\eta, \quad b_{ij} = \frac{1}{h_1} \int_{x_{i-0,5}}^{x_{i+0,5}} k(\xi, y_{j-0,5}) d\xi$$

при всех $i = 1, \dots, M, j = 1, \dots, N$. Здесь полуцелые узлы

$$x_{i\pm 0,5} = x_i \pm 0,5h_1, \quad y_{j\pm 0,5} = y_j \pm 0,5h_2.$$

Правая часть разностного уравнения

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(\xi, \eta) d\xi d\eta, \quad \Pi_{ij} = \{(x, y) \mid x_{i-0,5} \leq x \leq x_{i+0,5}, y_{j-0,5} \leq y \leq y_{j+0,5}\}$$

при всех $i = 1, \dots, M - 1, j = 1, \dots, N - 1$.

Краевые условия задачи Дирихле аппроксимируются точно равенством

$$w_{ij} = w(x_i, y_j) = 0, \quad (x_i, y_j) \in \Gamma.$$

Переменные, заданные краевым условием, исключаются из системы уравнений $Aw = B$. Все коэффициенты a_{ij} , b_{ij} и F_{ij} вычисляются аналитически. В результате нужно решить систему из $(M - 1) \times (N - 1)$ уравнений с $(M - 1) \times (N - 1)$ неизвестными.

Метод решения системы линейных алгебраических уравнений

Приближенное решение разностной схемы получается итерационным методом сопряженных градиентов. Для ускорения сходимости метода применяется диагональное предобуславливание.

Пусть оператор $D : H \rightarrow H$ действует на сеточные функции $w \in H$ по правилу

$$(Dw)_{ij} = [(a_{i+1j} + a_{ij})/h_1^2 + (b_{ij+1} + b_{ij})/h_2^2]w_{ij}, \quad i = 1, \dots, M - 1, j = 1, \dots, N - 1.$$

Этот оператор легко обратим по элементарным формулам, т.к. по сути является поэлементным умножением на положительные числа (в матричном виде это соответствовало бы умножению на диагональную невырожденную матрицу).

Начальное приближение $w^{(0)}$ к решению разностной схемы выбирается равным нулю во всех точках расчётной сетки.

Нулевая итерация совершается по формулам скорейшего спуска. Пусть $r^{(0)} = B$ - невязка начального приближения, функция $z^{(0)} = D^{-1}r^{(0)}$. Тогда направление спуска $p^{(1)} = z^{(0)}$, $q^{(1)} = Ap^{(1)}$, шаг вдоль направления спуска определяется параметром

$$\alpha_1 = \frac{(z^{(0)}, r^{(0)})}{(q^{(1)}, p^{(1)})}.$$

Следующее приближение $w^{(1)}$ вычисляется согласно равенству

$$w^{(1)} = w^{(0)} + \alpha_1 p^{(1)}.$$

Дальнейшие вычисления проводятся по следующим формулам. Пусть выполнено k итераций метода и функции $r^{(k-1)}$, $z^{(k-1)}$, $p^{(k)}$, $q^{(k)}$, $w^{(k)}$, а также коэффициент α_k считаются известными. Тогда невязка последней итерации

$$r^{(k)} = r^{(k-1)} - \alpha_k q^{(k)},$$

сеточная функция $z^{(k)} = D^{-1}r^{(k)}$. Следующее направление спуска

$$p^{(k+1)} = z^{(k)} + \beta_{k+1} p^{(k)},$$

где коэффициент

$$\beta_{k+1} = \frac{(z^{(k)}, r^{(k)})}{(z^{(k-1)}, r^{(k-1)})}.$$

Шаг спуска определяется параметром

$$\alpha_{k+1} = \frac{(z^{(k)}, r^{(k)})}{(q^{(k+1)}, p^{(k+1)})},$$

где функция

$$q^{(k+1)} = Ap^{(k+1)}.$$

Следующее приближение к точному решению $w^{(k+1)}$ вычисляется согласно равенству

$$w^{(k+1)} = w^{(k)} + \alpha_{k+1}p^{(k+1)}.$$

Метод сопряженных градиентов гарантирует, что при некотором k , не превосходящем $(M-1) \times (N-1)$ приближение $w^{(k)}$ станет равным точному решению разностной схемы. На практике это равенство нарушается из-за ошибок округлений, возникающих в процессе вычислений. Поэтому в качестве условия остановки итерационного процесса помимо максимально числа итераций используется неравенство

$$\|w^{(k+1)} - w^{(k)}\|_{L_2} < \delta.$$

Коэффициент δ выбран равным $3 \cdot 10^{-5}$, исходя из потребности в сходимости метода в отведенное время.

Работа по созданию MPI программы

В MPI программе исходная область разбивается на подобласти (домены), каждая из которых обрабатывается отдельным процессом. В задании требовалось разработать алгоритм такого разбиения, он реализован следующим образом.

- Пусть n - количество процессов, X и Y - размеры глобальной области
- Фиксируется $XY_{ratio} = X/Y$
- Далее для каждого делителя n_x числа n
 - $n_y = n \div n_x$
 - Если $2n_x < n_y$ или $2n_y < n_x$, то пропускаем делитель
 - $x = X \div n_x$, $y = Y \div n_y$ и $xy_{ratio} = x/y$
- Среди всех делителей, которые не пропустили, выбираем такой, при котором величина $|xy_{ratio} - XY_{ratio}|$ минимальна
- В силу целочисленного деления могли остаться нераспределенные узлы, число которых меньше $n_x \cdot n_y$. Их распределим по "принципу Дирихле" первым процессам по одному.

Указанных в алгоритме условий достаточно, чтобы найденное решение, если оно есть, удовлетворяло требованиям задания. Существование обеспечивается выбором числа процессов.

После определения оптимального разбиения каждый процесс получает свои локальные параметры сетки - размер сетки и смещение относительно глобальной сетки. С этими величинами процессы могут независимо инициализировать матрицы коэффициентов.

Далее код идейно повторяет последовательную версию, за исключением применения пяти-точечного шаблона. Здесь необходим обмен граничными значениями с соседними процессами. Для этого реализована функция `swap_values`, которая последовательно осуществляет обмены "снизу вверх" "сверху вниз" "справа налево" и "слева направо". Здесь послышки реализованы через неблокирующий `MPI_Isend`, а приёмы через блокирующий `MPI_Recv`, благодаря чему не происходит ситуации взаимной блокировки. После чтения процессы ожидают конца отправки при помощи `MPI_Wait` для синхронизации.

Также в конце каждого шага требуется вычислять нормы очередного шага. Для этого сначала вычисляется локальная норма, затем при помощи `MPI_Allgather` идёт пересылка всем процессам от всех.

После остановки итераций процессы синхронизируются, идёт пересылка локальных решений процессу с рангом 0. Процесс с рангом 0 записывает локальные решения в глобальную матрицу и выводит результат.

Работа по созданию MPI+OpenMP программы

Основная идея такая же, что и в обычной OpenMP программе - потоки работают независимо до тех пор, пока не возникает обменов информацией. В данном случае таких обменов всего 3 на каждом шаге:

1. Расчёт глобального скалярного произведения через `MPI_Allgather`.
2. Обмен граничными значениями для применения пятиточечного шаблона.
3. Очередной расчёт скалярного произведения.

Между этими шагами потоки работают с модификатором `nowait`, который убирает неявный барьер между циклами `for`.

Для корректной работы реализации MPI+OpenMP были написаны `.lsf`-скрипты для привязки потоков OpenMP к ядрам.

Прочее

При инициализации коэффициентов a_{ij} , b_{ij} , F_{ij} соответствующие интегралы вычисляются точно с использованием формулы Ньютона-Лейбница.

При решении разностной схемы количество итераций ограничено сверху количеством переменных, таким образом, программа гарантированно завершится. Помимо L_2 -нормы шага метода сопряженных градиентов отслеживаются L_1 и max нормы:

$$\|u\|_{L_1} = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} |u_{ij}| h_1 h_2, \quad \|u\|_{max} = \max_{\substack{i=1,\dots,M-1, \\ j=1,\dots,N-1}} |u_{ij}|.$$

Время работы MPI программы в секундах получается при помощи функции `MPI_Wtime`.

Результаты расчетов

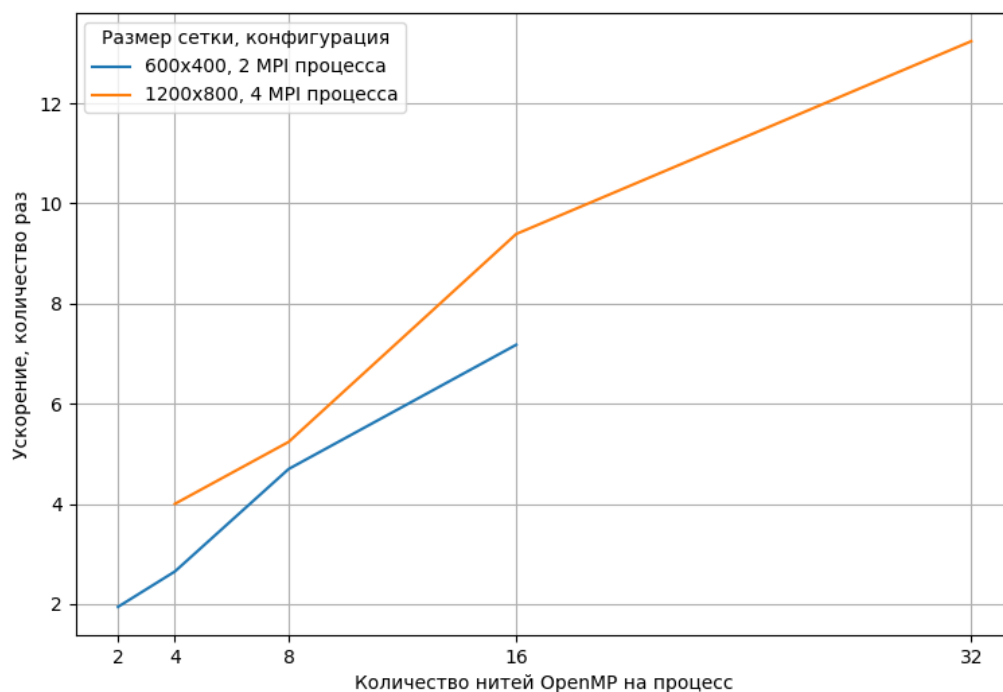
В таблице 1 представлены ускорения относительно CPU версии программы для двух расчетных сеток и разного количества процессов MPI и нитей OpenMP.

| Количество процессов MPI | Количество OpenMP-нитей в процессе | Число точек сетки ($M \times N$) | Число итераций | Время решения в секундах | Ускорение |
|--------------------------|------------------------------------|------------------------------------|----------------|--------------------------|-----------|
| 2 | 1 | 600x400 | 477 | 1.872 | 1.94 |
| 2 | 2 | 600x400 | 477 | 1.373 | 2.65 |
| 2 | 4 | 600x400 | 477 | 0.775 | 4.69 |
| 2 | 8 | 600x400 | 477 | 0.507 | 7.17 |
| 4 | 1 | 1200x800 | 718 | 5.816 | 4.00 |
| 4 | 2 | 1200x800 | 718 | 4.442 | 5.23 |
| 4 | 4 | 1200x800 | 718 | 2.479 | 9.38 |
| 4 | 8 | 1200x800 | 718 | 1.758 | 13.23 |

Таблица 1: Таблица с результатами расчетов на ПВС IBM Polus (MPI+OpenMP код).

Далее изображен график ускорения, построенный по табличным данным.

График ускорений MPI+OpenMP на разных расчетных сетках



Ниже представлен график численного решения задачи на сетке с параметрами $M = 1200$, $N = 800$.

Численное решение задачи на сетке 1200x800

