



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Аспирант 1 г.о.
Шведов Юрий Алексеевич

Отчет о проведенной работе по заданию курса
«Параллельные методы решения задач»

Москва, июнь 2016 г.

1 Постановка задачи

1. Разработать решение двумерного уравнения теплопроводности на прямоугольной сетке с вырезами.
2. Разработать параллельную гибридную программу с использованием технологий MPI и OpenMP, реализующую решение задачи.
3. Протестировать решение на вычислительной многопроцессорной системе, установить зависимость времени выполнения от числа процессоров.

2 Разностная схема для уравнения теплопроводности

Задача имеет следующий вид:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + f(x, y, t) \quad , \quad x \in [0, 1], y \in [0, 1], t \in [0, \infty) \quad ;$$

$$U(x, y, 0) = U_0(x, y) \quad ;$$

$$U(0, y, t) = U_l(y, t) \quad , \quad U(1, y, t) = U_r(y, t) \quad , \quad U(x, 0, t) = U_b(x, t) \quad , \quad U(x, 1, t) = U_t(x, t) \quad .$$

Разностная схема для этой задачи принимает вид:

$$\frac{U_{i,j}^{k+1} - U_{i,j}^k}{\tau} = \frac{U_{i-1,j}^k - 2U_{i,j}^k + U_{i+1,j}^k}{h_x^2} + \frac{U_{i,j-1}^k - 2U_{i,j}^k + U_{i,j+1}^k}{h_y^2} + f\left(\frac{i}{N}, \frac{j}{M}, \tau * k\right) \quad ;$$

$$h_x = \frac{1}{N}, h_y = \frac{1}{M}, \tau \leq \min\left(\frac{h_x^2}{2}, \frac{h_y^2}{2}\right) \quad , \quad U_{i,j}^0 = U_0\left(\frac{i}{N}, \frac{j}{M}\right) \quad ;$$

$$U_{0,j}^t = U_l\left(\frac{j}{M}, t\right) \quad , \quad U_{N,j}^t = U_r\left(\frac{j}{M}, t\right) \quad , \quad U_{j,0}^t = U_b\left(\frac{i}{N}, t\right) \quad , \quad U_{j,M}^t = U_t\left(\frac{i}{N}, t\right) \quad .$$

Здесь

- $U(x, y, t)$ — искомая функция, а $U_{i,j}^k$ - ее разностный аналог;
- h_x - шаг сетки по оси X;
- h_y - шаг сетки по оси Y;
- τ - шаг по времени;
- N, M - размеры сетки;
- должно выполняться следующее условие устойчивости $\tau < \frac{\min(h_x, h_y)^2}{2}$.

3 Описание решение

Поставленная в пп. 1 и 2 задача разбивается на две подзадачи:

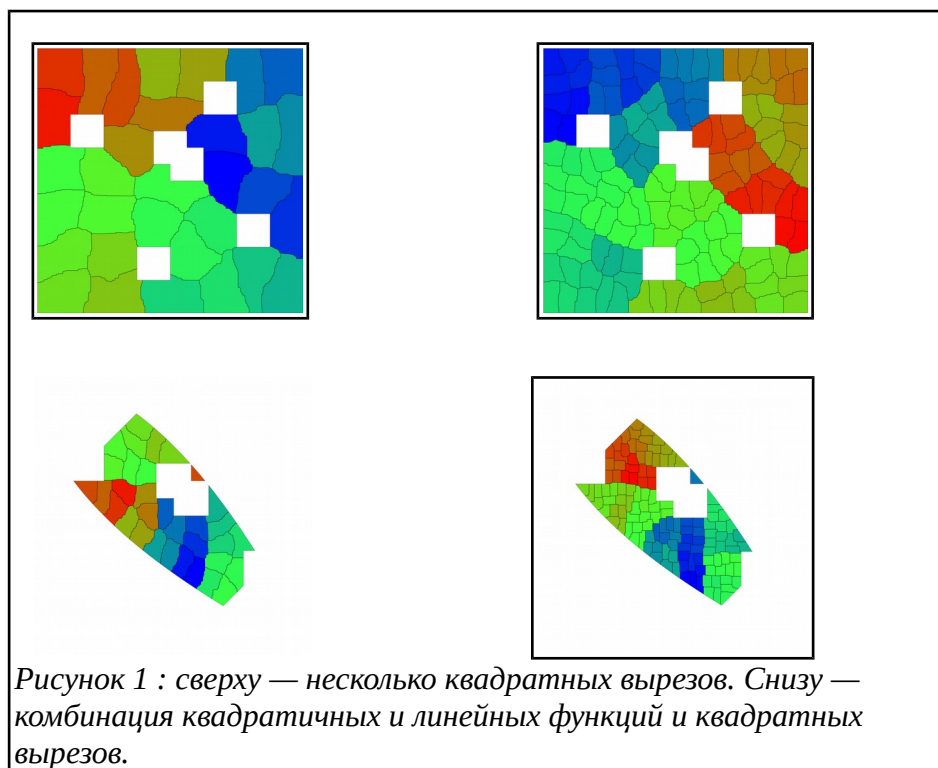
1. Декомпозиция сетки на домены по числу вычислительных узлов.
2. Организация вычисления по разностной схеме в пределах домена и обмена данными между доменами

3.1 Декомпозиция сетки

Для минимальной по суммарной величине разреза декомпозиции сетки была написана утилита, использующая пакет METIS. Эта утилита принимает на вход размеры сетки и набор имен вырезов, которые предварительно запрограммированы. После этого строит сетку в формате представления разреженных графов CSR (compressed storage format) и отдает на обработку пакету METIS.

Результат работы записывается в простой текстовый файл, где в строке с номером $(i \times j)$ располагается номер домена, обрабатывающего конкретную вершину на позиции (i, j) , или отрицательное число, если вершина попала в вырез. Утилита так же строит графическое представление разбиения в файле формата ppm. Декомпозиция выполняется заранее на одном вычислительном узле.

Результаты декомпозиции для двух сеток размером 2048x2048 на 32 и 128 доменов представлены на рис.1.



3.2 Вычисление в разностной сетке

Процесс вычисления в разностной сетке состоит из следующих шагов.

1. Загрузка карты декомпозиции в процессоры.
2. Вычисление новых значений и обмен с соседними процессами.
3. Построение промежуточных результатов, если запрошено пользователем.
4. Проверка условия выхода и выход в случае их выполнения или, иначе, переход на шаг 2

На первом шаге создается класс `Chunk`, которому передаются файлы с описанием декомпозиции. Этот класс создает множество объектов класса `Vertex`, которые разделены на 4 категории:

- Внутренняя вершина — вершина, для которой процессор вычисляет значения непосредственно и которая не граничит с вершинами из соседнего процессора.
- Граничная вершина — вершина, для которой процессор вычисляет значения непосредственно и которая граничит хотя бы с одной вершиной из соседнего процессора.
- Вырезанная вершина — вершина, которая попала в вырез или на границу сетки, значение этой вершины берется из граничных условий.
- Соседняя вершина — вершина, принадлежащая соседнему процессору.

Для хранения вершин последней категории создается множество объектов класса `Neighbour`, который проксирует обмен с одним из соседних процессоров, хранит подмножество соседних вершин и ссылок на граничные вершины.

На втором шаге класс `Chunk` перебирает множество внутренних вершин, вычисляя новое значение для каждой. Этот процесс производится параллельно с помощью `OpenMP` и периодически прерывается для проверки состояния соседей. При проверки состояния, идет обращение к классу `Neighbour`, который проверяет статус асинхронного приема. В случае, если прием завершен, производится подсчет новых значений соответствующих граничных вершин и запускается новый процесс асинхронного приема/передачи свежих данных. В конце шага производится ожидание приема недостающих данных, подсчет и асинхронная отправка новых, не вычисленных ранее значений граничных вершин.

На третьем и четвертом шагах вычисляется значение $t = k \times \tau$, для которого получено численное значение функции $U(x, y, t)$. Условием выхода является достижение заданного значения времени $t = k \times \tau$. После чего генерируется тепловая карта в формате `ppm`. По достижении некоторых промежуточных значений $t = k \times \tau$, может быть так же сгенерировано промежуточно значение тепловой карты, если в опциях командной строки указан интервал генерации.

4 Система

Исследования производились на системе IBM Blue Gene/P [<http://hpc.cs.msu.su/bgp>]. Это массивно-параллельная вычислительная система, которая состоит из двух стоек, включающих 8192 процессорных ядер, с пиковой производительностью 27,9 терафлопс.

Характеристики системы:

- две стойки с вычислительными узлами и узлами ввода-вывода
- 1024 четырехъядерных вычислительных узла в каждой из стоек
- 16 узлов ввода-вывода в стойке (в текущей конфигурации активны 8, т.е. одна I/O-карта на 128 вычислительных узлов)
- выделенные коммуникационные сети для межпроцессорных обменов и глобальных операций
- программирование с использованием MPI, OpenMP/threads, POSIX I/O
- высокая энергоэффективность: ~ 372 MFlops/W
- система воздушного охлаждения

5 Входные данные

Для измерения производительности были использованы сетки различного размера двух типов, которые были изображены на рис.1

- Сетка с шестью квадратными вырезами, со сторонами 0.125, верхние левые углы которых расположены в точках (0.25, 0.125), (0.125, 0.625), (0.625, 0.75), (0.75, 0.375), (0.375, 0.5) и (0.3125, 0.4375).
- Сетка с вырезами, указанными в предыдущем пункте, а так же с вырезами, содержащие точки, удовлетворяющие одному из следующих условий: $y < x^2$, $x < y^2$, $y < 0.5 - x$ и $y > 1.5 - x$.

Для простоты назовем первый тип сеток *квадратным*, а второй — *круглым*.

Начальные условия имеют следующий вид $U_0(x, y) = x \times y$.

Граничные условия имеют следующий вид $U_l(y, t) = U_r(y, t) = \sin(\pi y)$, $U_t(x, t) = U_b(x, t) = \sin(\pi x)$.

Граничные условия вырезов имеют следующий вид $U_h(x, y, t) = \sin(100\pi t)$.

А функция в правой части уравнения имеет вид $f(x, y, t) = \sin(100\pi t)$.

6 Аналитическое решение

Корректность программы проверялась на сетке размером 256x256 с аналитическим решением следующей задачи:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - 4 - 100 \pi \cos(100 \pi t)$$

$$U_0(x, y) = x^2 + y^2 \quad U_l(y, t) = y^2 + \sin(100 \pi y) \quad U_r(y, t) = 1 + y^2 + \sin(100 \pi y)$$

$$U_t(x, t) = x^2 + \sin(100 \pi x) \quad U_b(x, t) = 1 + x^2 + \sin(100 \pi x)$$

Аналитическое решение этой задачи имеет вид: $U(x, y, t) = x^2 + y^2 + \sin(100 \pi t)$. Его максимальное отклонение от численного решения в периоде $t \in [0, 0.003]$ составляет 0.108483 при $t = 0.012$. Визуальное сравнение аналитического и числового решений при $t = 0.012$ представлено на рис.2

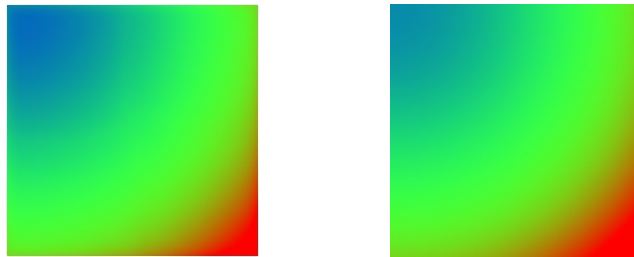


Рисунок 2 слева — числовое решение, справа — аналитическое

7 Результаты

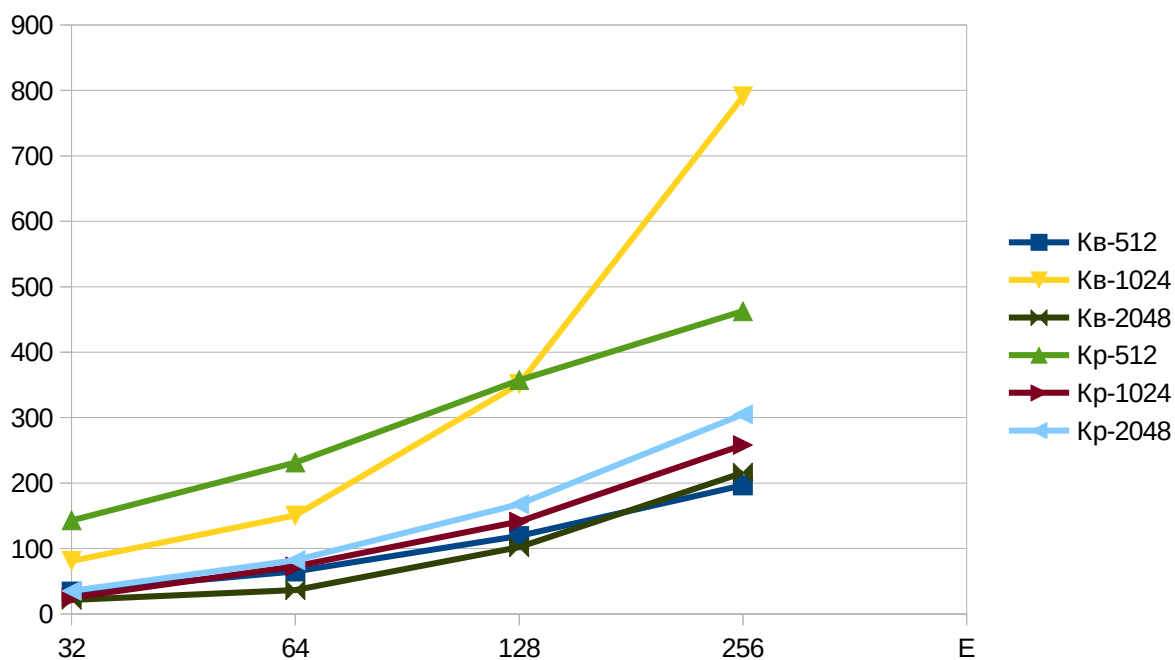
Исследования проводились на круглых и квадратных сетках размерами 512x512, 1024x1024 и 2048x2048. В таблице приведены данные по размерам декомпозиции для 32, 64, 128 и 256 процессоров: минимальное количество вершин, максимальное и среднее.

	32			64			128			256		
	min	avr	max	min	avr	max	min	avr	max	min	avr	max
КВ-512	7365	7393	7413	3640	3697	3733	1795	1849	1893	898	925	952
КВ-1024	29655	29697	29745	14815	14849	14876	7379	7425	7469	3680	3713	3752
КВ-2048	118953	119041	119088	59417	59521	59620	29705	29761	29803	14836	14881	14929
Кр-512	1454	1477	1517	724	739	761	359	370	392	180	185	191
Кр-1024	5838	5905	5947	2905	2953	2992	1434	1477	1560	717	739	761
Кр-2048	23595	23612	23656	11779	11806	11825	5734	5903	6228	2913	2952	3008

Исследование эффективности программы проводилось на 64, 128, 256 и 512 потоках (2 OpenMP потока на процессор). В таблице ниже представлены результаты измерений времени работы программы. Для сеток размером 512x512 условием выхода было $t = 0.03$, а для сеток размером 1024x1024 и 2048x2048 — $t = 0.0003$. Сравнение эффективности производилось относительно версии запущенной на 1 процессоре и 1 потоке.

Процессоры/Потоки	1	32	2	64	2	128	2	256	2	E
Размер сетки	T	T	S	T	S	T	S	T	S	
Кв-512	7864	228	34.49	121	64.99	66	119.2	40	196.6	0.47
Кв-1024	3164	39	81.13	21	150.7	9	351.6	4	791	1.34
Кв-2048	27861	1296	21.5	765	36.42	273	102.1	129	216	0.72
Кр-512	7864	55	143	34	231.3	22	357.5	17	462.6	1.59
Кр-1024	284	11.36	25	3.9	72.82	2.01	141.3	1.1	258.2	0.50
Кр-2048	4912	139	35.34	59.8	82.14	29.3	167.6	16.1	305.1	0.61
Total										0.872585
T = time (s) S = speedup E= efficiency										

Ниже изображены графики ускорения от количества процессоров.



На рис. 3 изображены некоторые результаты решения уравнений

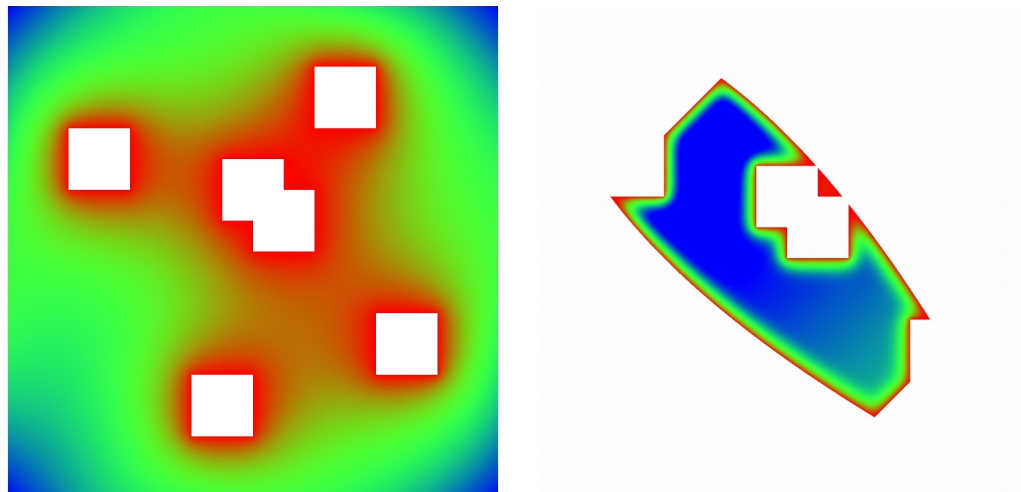


Рисунок 3 Слева — значение сетки с вырезами квадратного типа при $t=0.03$, справа - значение сетки с вырезами квадратного типа при $t=0.0003$

8 Исходный код

Исходный код программы доступен на GitHub: <https://github.com/ein-shved/superpc>. Для сборки проекта на системе Blue Gene/P достаточно запустить make на ветке bg.

9 Заключение

Была реализована параллельная программа решения двумерного уравнения теплопроводности на прямоугольной сетке с вырезами. При реализации использовались технологии OpenMP и MPI, а так же макет METIS для декомпозиции сетки. Для достижения более высокой временной эффективности была реализована асинхронная схема обмена сообщениями между процессорами. Проведены исследования производительности программы, которые показали ее среднюю эффективность относительно однопоточной программы.