

Отчет о выполненном практическом задании по курсу “Суперкомпьютерное моделирование и технологии”

Студентки 627 группы

Рагозиной Полины

(3 поток)

Постановка задачи

В трехмерной замкнутой области $\Omega = [0 \leq x \leq L_x] \times [0 \leq y \leq L_y] \times [0 \leq z \leq L_z]$ для $(0 < t \leq T]$ требуется найти решение $u(x, y, z, t)$ уравнения в частных производных $\frac{\partial^2 u}{\partial t^2} = \Delta u$ с начальными условиями $u_{t=0} = \varphi(x, y, z)$, $\frac{\partial u}{\partial t}_{t=0} = 0$, при условии, что на границах области заданы следующие граничные условия:

$$u(0, y, z, t) = 0,$$

$$u(L_x, y, z, t) = 0,$$

$$u(x, 0, z, t) = u(x, L_y, z, t),$$

$$u_y(x, 0, z, t) = u_y(x, L_y, z, t),$$

$$u(x, y, 0, t) = u(x, y, L_z, t),$$

$$u_z(x, y, 0, t) = u_z(x, y, L_z, t).$$

Численное решение

На Ω вводим сетку $\omega_{h\tau} = \overline{\omega}_h \times \omega_\tau$, где

$$T = T_0,$$

$$L_x = L_{x_0}, L_y = L_{y_0}, L_z = L_{z_0},$$

$$\overline{\omega}_h = \{(x_i = ih_x, y_j = jh_y, z_k = kh_z), i, j, k = 0, 1, \dots, N, h_x N = L_x, h_y N = L_y, h_z N = L_z\}, \omega_\tau = \{t_n = n\tau, n = 0, 1, \dots, K, \tau K = T\}.$$

Через ω_h обозначим множество внутренних узлов решётки, а через γ_h - граничных.

Аппроксимация - через систему уравнений:

$$\frac{u_{ijk}^{n+1} - 2u_{ijk}^n + u_{ijk}^{n-1}}{\tau^2} = \Delta_h u^n, (x_i, y_j, z_k) \in \omega_h, n = 1, 2, \dots, K-1,$$

где

$$\Delta_h u^n = \frac{u_{i-1,j,k}^n - 2u_{i,j,k}^n + u_{i+1,j,k}^n}{h_x^2} + \frac{u_{i,j-1,k}^n - 2u_{i,j,k}^n + u_{i,j+1,k}^n}{h_y^2} + \frac{u_{i,j,k-1}^n - 2u_{i,j,k}^n + u_{i,j,k+1}^n}{h_z^2}.$$

Зная u_{ijk}^0 и u_{ijk}^1 , можно посчитать аппроксимированные значения остальных u_{ijk}^n . Получим их так: $u_{ijk}^0 = \varphi(x_i, y_j, z_k)$, $u_{ijk}^1 = u_{ijk}^0 + \frac{\tau^2}{2} \Delta_h \varphi(x_i, y_j, z_k)$.

Последовательная реализация

Последовательная программа реализована очевидным образом. Производится итеративный проход в виде 4 вложенных циклов по матрице чисел с плавающей точкой $K \times N^3$, после каждой итерации поиск максимальной погрешности (эти значения хранятся в массиве длиной K). Так как вычисление значения в каждой ячейке использует значения предыдущих слоёв, все вычисления во время одной итерации независимы друг от друга.

Значения на границах вычисляются в соответствии с условием задачи.

На дальнейших этапах было решено отказаться от матрицы $K \times L^3$ в пользу матрицы $3 \times L^3$, так как в процессе вычисления в каждый момент времени требуется только текущий слой и два предыдущих. В момент t текущий слой имеет индекс $t\%3$, а предыдущие, соответственно, $(t-1)\%3$ и $(t-2)\%3$. В конце каждой итерации слои “смещаются”.

OpenMP

Параллельная OpenMP-программа получена простым добавлением omp-прагм: `#pragma omp for collapse(3)` в цикле обновления массива и omp-редукцией `#pragma omp for collapse(3) reduction(max:my_max)` при поиске максимальной погрешности.

MPI

Трёхмерная матрица на каждой итерации для удобства последующего добавления CUDA представлена в одномерном виде.

В этой части работы была использована топология “3D решётка”, таким образом если процессы составляют решётку $a \times b \times c$, то каждый процесс работает с частью решётки задачи размером $\frac{N}{a} \times \frac{N}{b} \times \frac{N}{c}$. Однако, чтобы избежать неоднозначности адресации, было решено создавать матрицу размером $3 \times N_{max}^3$, где N_{max} – максимальное из значений $\frac{N}{a}, \frac{N}{b}, \frac{N}{c}$. Адресация при этом выполняется так: $w[t][x][y][z] = arr[t\%3][z + y \cdot N_{max} + x \cdot N_{max} \cdot N_{max}]$. Помимо этого, на каждом процессе также создаётся 36 массивов-буферов размером N_{max}^2 – по два массива на отправку и приём данных в направлениях “от меньшего номера в топологии к большему” и “от большего к меньшему” в каждом из трёх измерений для каждой из трёх матриц. При добавлении нового значения в граничную ячейку решётки такое же значение добавляется в соответствующий буфер передачи. Затем на каждом этапе производится асинхронный обмен. Данные из буфера получения затем используются при подсчёте значений на новой итерации.

Кроме того, на каждой итерации каждый процесс считает наибольшую погрешность и из них выбирается самая большая с помощью `MPI_Reduce`.

MPI+OpenMP

Гибридная программа получена добавлением omp-прагм (`collapse` и `reduction`) к циклам заполнения ячеек и поиска погрешности. В остальном значительных изменений в MPI-программу внесено не было.

Таблицы результатов

Последовательная программа (L=1)		
N	T, с	δ
128	3,43	3,44E-06
256	27,1	6,88E-06
512	210	1,37E-05

Последовательная программа (L=pi)		
N	T, с	δ
128	3,46	3,44E-06
256	27,7	6,88E-06
512	213	1,37E-05

OpenMP (L=1)				
Число процессов	N	T, c	S	δ
1	128	3,77	1,0	3,44E-06
4	128	1,05	3,6	3,44E-06
8	128	0,528	7,1	3,44E-06
16	128	0,481	7,8	3,44E-06
32	128	0,331	11,4	3,44E-06
1	256	31	1,0	6,88E-06
4	256	7,69	4,0	6,88E-06
8	256	5,69	5,4	6,88E-06
16	256	3,2	9,7	6,88E-06
32	256	2,52	12,3	6,88E-06
1	512	255	1,0	1,37E-05
4	512	76	3,4	1,37E-05
8	512	53,2	4,8	1,37E-05
16	512	34,2	7,5	1,37E-05
32	512	27,8	9,2	1,37E-05

OpenMP (L=pi)				
Число процессов	N	T, c	S	δ
1	128	3,84	1,0	3,44E-06
4	128	1,4	2,7	3,44E-06
8	128	0,858	4,5	3,44E-06
16	128	0,541	7,1	3,44E-06
32	128	0,448	8,6	3,44E-06
1	256	30	1,0	6,88E-06
4	256	8,48	3,5	6,88E-06
8	256	6,54	4,6	6,88E-06
16	256	4,38	6,8	6,88E-06
32	256	3,13	9,6	6,88E-06
1	512	235	1,0	1,37E-05
4	512	64,3	3,7	1,37E-05
8	512	42,8	5,5	1,37E-05
16	512	30,2	7,8	1,37E-05
32	512	19,8	11,9	1,37E-05

MPI (L=1)				
Число процессов	N	T, с	S	δ
1	128	4,28	1,0	3,44E-06
4	128	1,3	3,3	3,46E-06
8	128	0,795	5,4	3,46E-06
16	128	0,664	6,4	3,46E-06
32	128	0,386	11,1	3,52E-06
1	256	34,2	1,0	6,88E-06
4	256	8,75	3,9	6,90E-06
8	256	5,43	6,3	6,90E-06
16	256	3,03	11,3	6,90E-06
32	256	2,43	14,1	6,96E-06
1	512	271	1,0	1,37E-05
4	512	79,8	3,4	1,38E-05
8	512	45,2	6,0	1,38E-05
16	512	20,8	13,0	1,38E-05
32	512	19,7	13,8	1,38E-05

MPI (L=pi)				
Число процессов	N	T, с	S	δ
1	128	4,3	1,0	3,48E-07
4	128	1,4	3,1	3,51E-07
8	128	0,899	4,8	3,51E-07
16	128	0,64	6,7	3,51E-07
32	128	0,491	8,8	3,56E-07
1	256	35,1	1,0	6,97E-07
4	256	10,6	3,3	6,99E-07
8	256	4,79	7,3	6,99E-07
16	256	3,59	9,8	6,99E-07
32	256	2,13	16,5	7,05E-07
1	512	296	1,0	1,39E-06
4	512	87,8	3,4	1,40E-06
8	512	40,3	7,3	1,40E-06
16	512	33,4	8,9	1,40E-06
32	512	10,4	28,5	1,40E-06

MPI+OpenMP (L=1)					
Число MPI процессов	Число OMP нитей	N	T, с	S	δ
1	4	128	1,34	1,0	3,44E-06
2	4	128	0,931	1,4	3,43E-06
1	4	256	10,5	1,0	6,90E-06
2	4	256	7	1,5	6,88E-06
1	4	512	83,8	1,0	1,37E-05
2	4	512	58,3	1,4	1,37E-05

MPI+OpenMP (L=pi)					
Число MPI процессов	Число OMP нитей	N	T, с	S	δ
1	4	128	1,34	1,0	3,48E-07
2	4	128	0,67	2,0	3,48E-07
1	4	256	10,5	1,0	6,97E-07
2	4	256	5,32	2,0	6,97E-07
1	4	512	84	1,0	1,39E-06
2	4	512	40,4	2,1	1,39E-06