Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра интеллектуальных информационных технологий

Отчет о выполнении задания по курсу «Суперкомпьютерное моделирование и технологии»

Выполнил:
Багамаев Мурад Аммаевич
студент 622 группы
вариант №7

Москва, 2024

Содержание

1 Постановка задачи

В задании требуется реализовать решение трехмерного гиперболического уравнения :

$$\frac{\partial^2 u}{\partial t^2} = \Delta u \tag{1}$$

в трехмерной замкнутой области

$$\Omega = [0 \le x \le L_x] \times [0 \le y \le L_y] \times [0 \le z \le L_z] \tag{2}$$

с начальными и граничными условиями:

$$\begin{aligned}
u|_{t=0} &= \varphi(x, y, z), \\
\left.\frac{\partial u}{\partial t}\right|_{t=0} &= 0, \\
u(0, y, z, t) &= u(L_x, y, z, t), \quad u_x(0, y, z, t) = u_x(L_x, y, z, t), \\
u(x, 0, z, t) &= u(x, L_y, z, t), \quad u_y(x, 0, z, t) = u_y(x, L_y, z, t), \\
u(x, y, 0, t) &= 0, \quad u(x, y, L_z, t) = 0.
\end{aligned}$$

2 Описание численной схемы

Для численного решения используется двухшаговая явная разностная схема:

$$u_{ijk}^{n+1} = \tau^2 \Delta_h u^n + 2u_{ijk}^n - u_{ijk}^{n-1}, \tag{3}$$

где $\Delta_h u^n$ — семиточечный оператор Лапласа:

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

Начальные условия задаются как:

$$\begin{aligned}
u_{ijk}^0 &= \varphi(x_i, y_j, z_k), \\
u_{ijk}^1 &= u_{ijk}^0 + \frac{\tau^2}{2} \Delta_h \varphi(x_i, y_j, z_k).
\end{aligned}$$

Аналитическое решение имеет вид:

$$u_{\text{analytical}} = \sin\left(\frac{2\pi}{L_x} x + 3\pi\right) \sin\left(\frac{2\pi}{L_y} y + 2\pi\right) \sin\left(\frac{\pi}{L_z} z\right) \cos(a_t \cdot t + \pi), \tag{4}$$

где $a_t = \pi\sqrt{\frac{4}{L_x^2} + \frac{4}{L_y^2} + \frac{1}{L_z^2}}$.

3 Создание гибридной реализации MPI/OpenMP

Для данной работы была разработана гибридная реализация алгоритма решения трехмерного гиперболического уравнения с использованием MPI и OpenMP. Основной подход заключался в разбиении пространственной сетки между MPI процессами и использовании директив OpenMP для ускорения расчётов внутри каждого процесса. Проведённый анализ показал, что:

- MPI обеспечивает масштабирование за счёт распределения сетки между узлами;
- OpenMP снижает время выполнения благодаря параллелизации операций в рамках одного процесса с помощью директивы `omp parallel for`;
- Использовались неблокирующие команды `MPI_Isend` и `MPI_Irecv` для передачи данных между процессами, что позволило сократить накладные расходы на обмен сообщениями;
- Эффективность комбинации MPI и OpenMP зависит от размера задачи и количества вычислительных ресурсов.

4 Результаты расчетов

Число MPI процессов N_p	**Число точек сетки** N^3	**Время решения** T	**Ускорение** S	**Погрешность** δ
1	128^3	2619.7	1	7.40813e-07
4	128^3	664.204	3.94	7.40813e-07
8	128^3	452.345	5.79	7.40813e-07
16	128^3	304.195	8.6	7.40813e-07
32	128^3	380.909	6.88	7.40813e-07
1	256^3	21055.2	1	1.78227e-07
4	256^3	5706.1	3.69	1.78227e-07
8	256^3	3087.52	6.82	1.78227e-07
16	256^3	1959.56	10.74	1.78227e-07
32	256^3	1364.12	15.44	1.78227e-07
1	512^3	178412	1	3.88221e-08
4	512^3	47166.7	3.78	3.88221e-08
8	512^3	32953.5	5.4	3.88221e-08
16	512^3	18526.7	9.63	3.88221e-08
32	512^3	12198.4	14.63	3.88221e-08

Таблица 1: MPI код ($L_x = L_y = L_z = 1$)

Число MPI процессов N_p	**Число точек сетки** N^3	**Время решения** T	**Ускорение** S	**Погрешность** δ
1	128^3	2618.91	1	7.58252e-08
4	128^3	778.221	3.37	7.58252e-08
8	128^3	469.684	5.58	7.58252e-08
16	128^3	340.561	7.69	7.58252e-08
32	128^3	268.42	9.76	7.58252e-08
1	256^3	21078.4	1	1.8755e-08
4	256^3	5783.56	3.64	1.8755e-08
8	256^3	3112.9	6.77	1.8755e-08
16	256^3	1914.73	11	1.8755e-08
32	256^3	1430.48	14.74	1.8755e-08
1	512^3	168836	1	4.61352e-09
4	512^3	44440.2	3.8	4.61352e-09
8	512^3	23491.9	7.2	4.61352e-09
16	512^3	11807.7	14.3	4.61352e-09
32	512^3	7927	21.3	4.61352e-09

Таблица 2: MPI код ($L_x = L_y = L_z = \pi$)

Число MPI процессов N_p	**Число OpenMP нитей в процессе**	**Число точек сетки** N^3	**Время решения** T	**Ускорение** S	**Погрешность** δ
1	4	128^3	2814.81	1	7.40813e-07
2	4	128^3	2707.63	1.04	7.40813e-07
4	4	128^3	2450.71	1.15	7.40813e-07
8	4	128^3	2362.41	1.19	7.40813e-07
1	4	256^3	15012.6	1	1.78227e-07
2	4	256^3	8810.61	1.7	1.78227e-07
4	4	256^3	5763.87	2.6	1.78227e-07
8	4	256^3	4111.97	3.65	1.78227e-07
1	4	512^3	244478	1	3.88221e-08
2	4	512^3	173071	1.41	3.88221e-08
4	4	512^3	126849	1.93	3.88221e-08
8	4	512^3	104798	2.33	3.88221e-08

Таблица 3: MPI + OpenMP код ($L_x = L_y = L_z = 1$)

Число MPI процессов N_p	**Число OpenMP нитей в процессе**	**Число точек сетки** N^3	**Время решения** T	**Ускорение** S	**Погрешность** δ
1	4	128^3	2826.05	1	7.58252e-08
2	4	128^3	2743.27	1.03	7.58252e-08
4	4	128^3	2603.44	1.09	7.58252e-08
8	4	128^3	2382.8	1.19	7.58252e-08
1	4	256^3	14918.1	1	1.8755e-08
2	4	256^3	9036.27	1.65	1.8755e-08
4	4	256^3	6324.23	2.36	1.8755e-08
8	4	256^3	4459.97	3.34	1.8755e-08
1	4	512^3	135200	1	4.61352e-09
2	4	512^3	71576.8	1.89	4.61352e-09
4	4	512^3	36125.9	3.74	4.61352e-09
8	4	512^3	19661.6	6.88	4.61352e-09

Таблица 4: MPI + OpenMP код ($L_x = L_y = L_z = \pi$)