Суперкомпьютерное моделирование и технологии

Отчёт
Задача для трёхмерного гиперболического уравнения в прямоугольном параллелепипеде.

Перминов Андрей Игоревич

Вариант 3

31.10.2021

Содержание

Математическая постановка задачи

В трёхмерной замкнутой области

$$\Omega = [0 \le x \le L_x] \times [0 \le y \le L_y] \times [0 \le z \le L_z]$$

для $0 < t \le T$ требуется найти решение $u(x, y, z, t)$ уравнения в частных производных

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

с начальными условиями

$$\begin{cases} u|_{t=0} = \phi(x, y, z) \\ \left.\frac{\partial u}{\partial t}\right|_{t=0} = 0 \\ u(0, y, z, t) = 0 \\ u(L_x, y, z, t) = 0 \\ u(x, 0, z, t) = u(x, L_y, z, t) \\ u_y(x, 0, z, t) = u_y(x, L_y, z, t) \\ u(x, y, 0, t) = 0 \\ u(x, y, L_z, t) = 0 \end{cases}$$

Численный метод решения задачи

Введём на Ω сетку $\omega_{h\tau} = \bar{\omega}_h \times \omega_\tau$, где

$T = T_0$

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

$\bar{\omega}_h = \{(x_i = ih_x, y_j = jh_y, z_k = kh_z), i, j, k = \overline{0, N}, h_x N = L_x, h_y N = L_y, h_z N = L_z\}$

$\omega_\tau = \{t_n = n\tau, n = \overline{0, K}, \tau K = T$

Через ω_h обозначим множество внутренних, а через γ_h – множество граничных узлов сетки $\bar{\omega}_h$.

Для аппроксимации исходного уравнения воспользуемся следующей системой уравнений:

$$\frac{u_{i,j,k}^{n+1} - 2u_{i,j,k}^n + u_{i,j,k}^{n-1}}{\tau^2} = \Delta_h u^n, (x_i, y_i, z_i) \in \omega_h, n = \overline{1, K-1}$$

Здесь Δ_h – семиточечный разностный аналог оператора Лапласа:

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

Приведённая выше разностная схема является явной – значения $u_{i,j,k}^{n+1}$ на $(n+1)$-м шаге можно явным образом выразить через значения на предыдущих слоях.

Для начала счёта должны быть заданы значения $u^0_{i,j,k}, u^1_{i,j,k}, (x_i, y_i, z_i) \in \omega_h$:

$$u^0_{i,j,k} = \phi(x_i, y_i, z_i), \qquad (x_i, y_i, z_i) \in \omega_h$$

$$u^1_{i,j,k} = u^0_{i,j,k} + \frac{\tau^2}{2}\Delta_h \phi(x_i, y_i, z_i)$$

$$u^{n+1}_{i,0,k} = u^{n+1}_{i,N,k}$$

$$u^{n+1}_{i,1,k} = u^{n+1}_{i,N+1,k}$$

$i, j, k = \overline{0, N}$

Программная реализация

Реализовано две программы: последовательная и гибридная параллельная (MPI + OpenMP). Каждая из программ является консольным приложением и принимает входные данные в виде аргументов командной строки. Используются следующие аргументы (доступно при вызове с единственным аргументом `--help`):

- `-d` – отладочный режим (по умолчанию не используется)
- `-Lx` – длина параллелепипеда вдоль оси X (по умолчанию 1)
- `-Ly` – длина параллелепипеда вдоль оси Y (по умолчанию 1)
- `-Lz` – длина параллелепипеда вдоль оси Z (по умолчанию 1)
- `-T` – конечное время сетки (по умолчанию 1)
- `-N` – количество точек пространственной сетки (по умолчанию 40)
- `-K` – количество точек временной сетки (по умолчанию 100)
- `-steps` – количество шагов для решения (по умолчанию 20)
- `-btx` – тип граничного условия вдоль оси X (по умолчанию однородные первого рода)
- `-bty` – тип граничного условия вдоль оси Y (по умолчанию периодические-численные)
- `-btz` – тип граничного условия вдоль оси Z (по умолчанию однородные первого рода)
- `-s` – стратегия разбиения на блоки (по умолчанию блочное, доступно только для MPI версии)
- `-on` – путь к json файлу для сохранения численного решения (по умолчанию не используется)
- `-oa` – путь к json файлу для сохранения аналитического решения (по умолчанию не используется)
- `-od` – путь к json файлу для сохранения погрешности (по умолчанию не используется)
- `-o` – путь к текстовому файлу для вывода результатов (по умолчанию output.txt)

Типы граничных условий:

- first-kind (f) – однородные граничные условия первого рода
- periodic-analytical (pa) – аналитические периодические граничные условия
- periodic-numerical (pn) – численные периодические граничные условия

Типы стратегий разбиения

- blocks (b) – блочное разбиение
- tapes (t) – ленточное разбиение

Для визуализации получаемых решений написан визуализатор (принимает json файлы, генерируемые программой): solve-visualizer

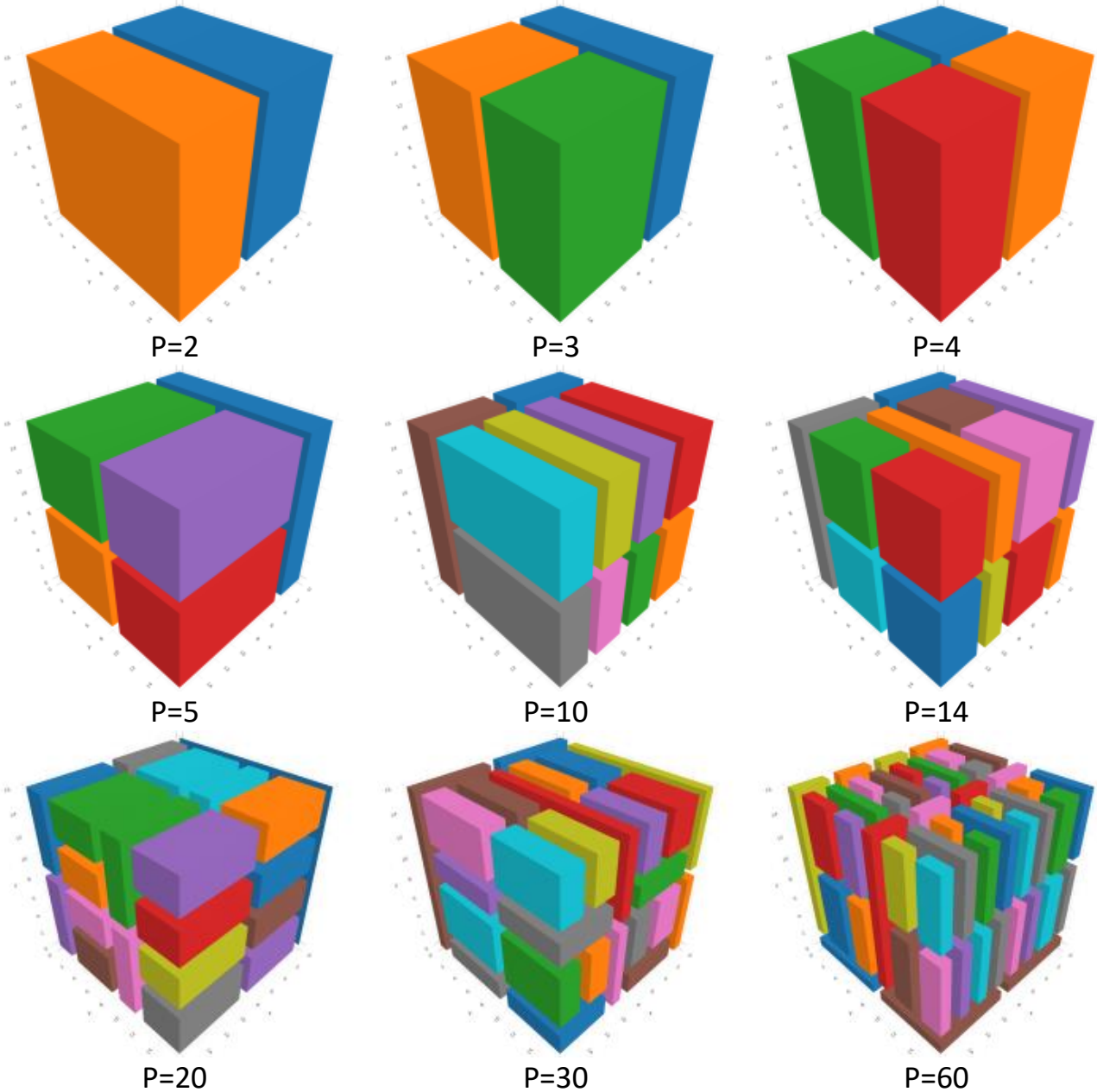
Особенности параллельной реализации

Для распараллеливания вся сетка разбивается на области (также прямоугольные параллелепипеды) в количестве используемых процессов по следующему алгоритму:

- начнём разбиение с параллелепипеда $[0, N]\ x\ [0, N]\ x\ [0, N]$, выберем начальную ось (X) и запустим рекурсивный процесс
- если текущее количество областей ($size$) равно 1, вернём обрабатываемый параллелепипед
- если размер нечётный, то по текущей оси выберем область $\frac{1}{size}$ и сделаем из неё параллелепипед, и продолжим разбивать область $1 - \frac{1}{size}$
- по выбранной оси делим область пополам и рекурсивно запускаем для этих подобластей переходя на следующую ось ($X \to Y,\ Y \to Z,\ Z \to X$).

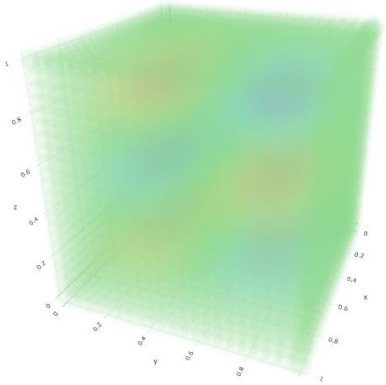
Для наглядности реализован визуализатор разбиения: split-visualizer

Получающиеся разбиения

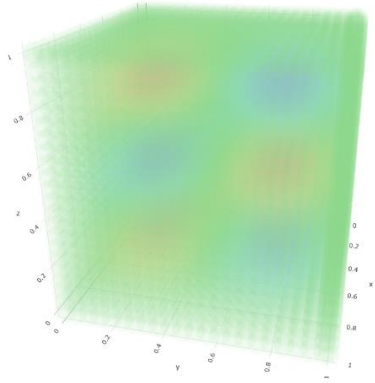


Графики аналитического и полученного решений

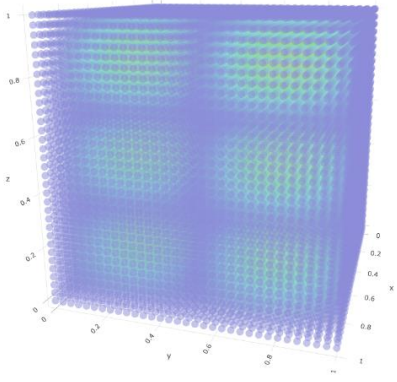
Решение при $L_x = L_y = L_z = 1$



аналитическое

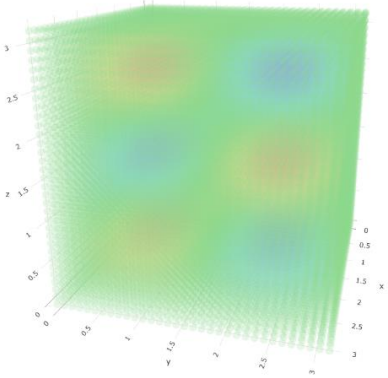


полученное

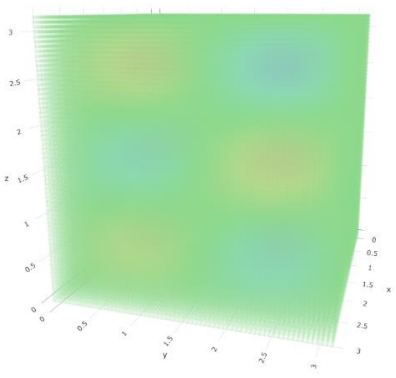


погрешность

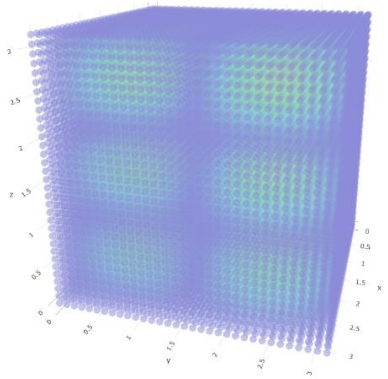
Решение при $L_x = L_y = L_z = \pi$



аналитическое



полученное



погрешность

Результаты запусков программы на различных кластерах

Polus (MPI версия)

Lx = Ly = Lz = 1, N = 128, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	3.37891	1.000	0.00541829	3.38371	1.000	0.00541829
2	1.97175	1.714	0.00541829	1.98044	1.709	0.00541829
4	1.13682	2.972	0.00541829	1.13015	2.994	0.00541829
8	0.761273	4.438	0.00541829	0.715661	4.728	0.00541829
10	0.697569	4.844	0.00541829	0.573838	5.897	0.00541829
16	0.430247	7.853	0.00541829	0.409231	8.268	0.00541829
20	0.389413	8.677	0.00541829	0.36037	9.390	0.00541829
32	0.302416	11.173	0.00541829	0.259181	13.055	0.00541829
40	0.24562	13.757	0.00541829	0.250944	13.484	0.00541829
64	0.175378	19.266	0.00541829	0.182393	18.552	0.00541829

Lx = Ly = Lz = 1, N = 256, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	26.8418	1.000	0.00542011	26.8246	1.000	0.00542011
2	15.5534	1.726	0.00542011	15.6095	1.718	0.00542011
4	8.5587	3.136	0.00542011	8.58564	3.124	0.00542011
8	5.74982	4.668	0.00542011	5.16543	5.193	0.00542011
10	5.12554	5.237	0.00542011	4.26427	6.291	0.00542011
16	3.6738	7.306	0.00542011	2.85054	9.410	0.00542011
20	3.17586	8.452	0.00542011	2.39508	11.200	0.00542011
32	2.13905	12.548	0.00542011	1.74538	15.369	0.00542011
40	1.84101	14.580	0.00542011	1.55653	17.234	0.00542011
64	1.23209	21.786	0.00542011	1.12446	23.856	0.00542011

Lx = Ly = Lz = 1, N = 512, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	212.339	1.000	0.00542056	212.174	1.000	0.00542056
2	119.572	1.776	0.00542056	120.541	1.760	0.00542056
4	66.7043	3.183	0.00542056	66.9953	3.167	0.00542056
8	38.7001	5.487	0.00542056	35.9597	5.900	0.00542056
10	38.6523	5.494	0.00542056	31.6109	6.712	0.00542056
16	22.3261	9.511	0.00542056	20.8502	10.176	0.00542056
20	23.1022	9.191	0.00542056	19.5098	10.875	0.00542056
32	14.4698	14.675	0.00542056	12.6782	16.735	0.00542056
40	13.1529	16.144	0.00542056	10.8413	19.571	0.00542056
64	8.52932	24.895	0.00542056	7.82263	27.123	0.00542056

Lx = Ly = Lz = π, N = 128, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	3.37747	1.000	0.000549676	3.38126	1.000	0.000549676
2	1.99226	1.695	0.000549676	1.98962	1.699	0.000549676
4	1.12936	2.991	0.000549676	1.13578	2.977	0.000549676
8	0.847514	3.985	0.000549676	0.579493	5.835	0.000549676
10	0.673376	5.016	0.000549676	0.552498	6.120	0.000549676
16	0.46852	7.209	0.000549676	0.402093	8.409	0.000549676
20	0.421829	8.007	0.000549676	0.378469	8.934	0.000549676
32	0.29494	11.451	0.000549676	0.257307	13.141	0.000549676
40	0.278028	12.148	0.000549676	0.227738	14.847	0.000549676
64	0.172591	19.569	0.000549676	0.195346	17.309	0.000549676

Lx = Ly = Lz = π, N = 256, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	26.8546	1.000	0.000549861	26.8581	1.000	0.000549861
2	15.2069	1.766	0.000549861	14.7859	1.816	0.000549861
4	8.37291	3.207	0.000549861	9.6132	2.794	0.000549861
8	4.72224	5.687	0.000549861	4.50557	5.961	0.000549861
10	4.64202	5.785	0.000549861	4.07747	6.587	0.000549861
16	3.2318	8.309	0.000549861	2.55629	10.507	0.000549861
20	2.85524	9.405	0.000549861	2.35552	11.402	0.000549861
32	2.20191	12.196	0.000549861	1.76651	15.204	0.000549861
40	2.04728	13.117	0.000549861	1.57969	17.002	0.000549861
64	1.2482	21.515	0.000549861	1.11757	24.033	0.000549861

Lx = Ly = Lz = π, N = 512, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	212.134	1.000	0.000549907	212.326	1.000	0.000549907
2	119.237	1.779	0.000549907	119.366	1.779	0.000549907
4	66.5768	3.186	0.000549907	67.0074	3.169	0.000549907
8	40.2159	5.275	0.000549907	34.0599	6.234	0.000549907
10	36.5986	5.796	0.000549907	27.4475	7.736	0.000549907
16	23.8981	8.877	0.000549907	21.9308	9.682	0.000549907
20	23.9314	8.864	0.000549907	17.8668	11.884	0.000549907
32	14.2082	14.930	0.000549907	13.0011	16.331	0.000549907
40	13.68	15.507	0.000549907	10.1392	20.941	0.000549907
64	8.81245	24.072	0.000549907	7.60838	27.907	0.000549907

Blue Gene (MPI версия)

Lx = Ly = Lz = 1, N = 128, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	141.493	1.000	0.00541829	141.494	1.000	0.00541829
2	77.0068	1.837	0.00541829	77.0072	1.837	0.00541829
4	40.8615	3.463	0.00541829	40.0407	3.534	0.00541829
8	21.9844	6.436	0.00541829	20.883	6.776	0.00541829
16	11.5286	12.273	0.00541829	11.2159	12.615	0.00541829
32	6.04969	23.388	0.00541829	7.17778	19.713	0.00541829
64	3.23408	43.751	0.00541829	6.74018	20.993	0.00541829
128	1.69645	83.405	0.00541829	9.62115	14.707	0.00541829
256	0.901327	156.983	0.00541829	-	-	-

Lx = Ly = Lz = 1, N = 256, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	1110.44	1.000	0.00542011	1110.44	1.000	0.00542011
2	605.897	1.833	0.00542011	605.875	1.833	0.00542011
4	322.288	3.445	0.00542011	315.013	3.525	0.00542011
8	174.383	6.368	0.00542011	162.972	6.814	0.00542011
16	90.9261	12.213	0.00542011	84.5998	13.126	0.00542011
32	47.7614	23.250	0.00542011	48.3104	22.986	0.00542011
64	25.336	43.829	0.00542011	36.0454	30.807	0.00542011
128	13.4745	82.410	0.00542011	42.6008	26.066	0.00542011
256	6.70683	165.569	0.00542011	70.5973	15.729	0.00542011

Lx = Ly = Lz = 1, N = 512, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	8600.92	1.000	0.00542056	9423.54	1.000	0.00542056
2	4751.89	1.810	0.00542056	4759.36	1.980	0.00542056
4	2568.59	3.348	0.00542056	2491.81	3.782	0.00542056
8	1380.79	6.229	0.00542056	1284.44	7.337	0.00542056
16	722.163	11.910	0.00542056	656.146	14.362	0.00542056
32	380.695	22.593	0.00542056	352.157	26.759	0.00542056
64	202.1	42.558	0.00542056	223.976	42.074	0.00542056
128	104.098	82.623	0.00542056	210.467	44.774	0.00542056
256	52.0685	165.185	0.00542056	312.152	30.188	0.00542056

Lx = Ly = Lz = π, N = 128, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	142.496	1.000	0.000549676	142.496	1.000	0.000549676
2	77.4895	1.839	0.000549676	77.4894	1.839	0.000549676
4	41.1135	3.466	0.000549676	40.3135	3.535	0.000549676
8	22.1162	6.443	0.000549676	20.9752	6.794	0.000549676
16	11.6038	12.280	0.000549676	11.2888	12.623	0.000549676
32	6.08839	23.405	0.000549676	7.16593	19.885	0.000549676
64	3.25425	43.788	0.000549676	6.74699	21.120	0.000549676
128	1.70468	83.591	0.000549676	9.62002	14.812	0.000549676
256	1.50589	94.626	0.000549676	-	-	-

Lx = Ly = Lz = π, N = 256, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	1120.28	1.000	0.000549861	1120.28	1.000	0.000549861
2	610.702	1.834	0.000549861	610.702	1.834	0.000549861
4	324.514	3.452	0.000549861	317.597	3.527	0.000549861
8	175.456	6.385	0.000549861	163.753	6.841	0.000549861
16	91.5347	12.239	0.000549861	85.2042	13.148	0.000549861
32	48.1049	23.288	0.000549861	48.4248	23.134	0.000549861
64	25.5064	43.922	0.000549861	36.1936	30.952	0.000549861
128	13.1408	85.252	0.000549861	42.6262	26.281	0.000549861
256	7.2389	154.758	0.000549861	70.9196	15.796	0.000549861

Lx = Ly = Lz = π, N = 512, K = 2000

Число MPI процессов (P)	Блочное разбиение			Ленточное разбиение		
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность
1	8605.17	1.000	0.000549907	9431.81	1.000	0.000549907
2	4754.13	1.810	0.000549907	4765.72	1.979	0.000549907
4	2574.72	3.342	0.000549907	2497.13	3.777	0.000549907
8	1392.04	6.182	0.000549907	1293.15	7.294	0.000549907
16	728.384	11.814	0.000549907	661.571	14.257	0.000549907
32	383.046	22.465	0.000549907	353.86	26.654	0.000549907
64	202.853	42.421	0.000549907	225.365	41.851	0.000549907
128	103.621	83.045	0.000549907	211.249	44.648	0.000549907
256	52.5081	163.883	0.000549907	316.831	29.769	0.000549907

Blue Gene (гибридная версия, блочное разбиение)

Lx = Ly = Lz = 1, N = 128, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	141.493	1.000	0.00541829	87.9429	1.000	0.00541829	1.609
2	77.0068	1.837	0.00541829	47.2583	1.861	0.00541829	1.629
4	40.8615	3.463	0.00541829	24.6753	3.564	0.00541829	1.656
8	21.9844	6.436	0.00541829	12.9544	6.789	0.00541829	1.697
16	11.5286	12.273	0.00541829	6.78328	12.965	0.00541829	1.700
32	6.04969	23.388	0.00541829	3.49826	25.139	0.00541829	1.729
64	3.23408	43.751	0.00541829	1.83381	47.956	0.00541829	1.764
128	1.69645	83.405	0.00541829	1.02677	85.650	0.00541829	1.652
256	0.901327	156.983	0.00541829	0.603308	145.768	0.00541829	1.494

Lx = Ly = Lz = 1, N = 256, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	1110.44	1.000	0.00542011	682.316	1.000	0.00542011	1.627
2	605.897	1.833	0.00542011	366.57	1.861	0.00542011	1.653
4	322.288	3.445	0.00542011	192.21	3.550	0.00542011	1.677
8	174.383	6.368	0.00542011	100.854	6.765	0.00542011	1.729
16	90.9261	12.213	0.00542011	52.1929	13.073	0.00542011	1.742
32	47.7614	23.250	0.00542011	26.8175	25.443	0.00542011	1.781
64	25.336	43.829	0.00542011	13.8613	49.225	0.00542011	1.828
128	13.4745	82.410	0.00542011	7.21106	94.621	0.00542011	1.869
256	6.70683	165.569	0.00542011	4.45263	153.239	0.00542011	1.506

Lx = Ly = Lz = 1, N = 512, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	8600.92	1.000	0.00542056	5531.18	1.000	0.00542056	1.555
2	4751.89	1.810	0.00542056	2895.61	1.910	0.00542056	1.641
4	2568.59	3.348	0.00542056	1517.01	3.646	0.00542056	1.693
8	1380.79	6.229	0.00542056	794.502	6.962	0.00542056	1.738
16	722.163	11.910	0.00542056	407.951	13.558	0.00542056	1.770
32	380.695	22.593	0.00542056	210.244	26.308	0.00542056	1.811
64	202.1	42.558	0.00542056	108.913	50.785	0.00542056	1.856
128	104.098	82.623	0.00542056	55.9623	98.838	0.00542056	1.860
256	52.0685	165.185	0.00542056	29.8807	185.109	0.00542056	1.743

Lx = Ly = Lz = π, N = 128, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	142.496	1.000	0.000549676	88.0216	1.000	0.000549676	1.619
2	77.4895	1.839	0.000549676	47.2734	1.862	0.000549676	1.639
4	41.1135	3.466	0.000549676	24.6981	3.564	0.000549676	1.665
8	22.1162	6.443	0.000549676	12.9627	6.790	0.000549676	1.706
16	11.6038	12.280	0.000549676	6.7755	12.991	0.000549676	1.713
32	6.08839	23.405	0.000549676	3.49033	25.219	0.000549676	1.744
64	3.25425	43.788	0.000549676	1.8336	48.005	0.000549676	1.775
128	1.70468	83.591	0.000549676	1.02571	85.815	0.000549676	1.662
256	1.50589	94.626	0.000549676	1.15484	76.220	0.000549676	1.304

Lx = Ly = Lz = π, N = 256, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	1120.28	1.000	0.000549861	684.851	1.000	0.000549861	1.636
2	610.702	1.834	0.000549861	367.672	1.863	0.000549861	1.661
4	324.514	3.452	0.000549861	192.997	3.549	0.000549861	1.681
8	175.456	6.385	0.000549861	101.139	6.771	0.000549861	1.735
16	91.5347	12.239	0.000549861	52.277	13.100	0.000549861	1.751
32	48.1049	23.288	0.000549861	26.8123	25.542	0.000549861	1.794
64	25.5064	43.922	0.000549861	13.8648	49.395	0.000549861	1.840
128	13.1408	85.252	0.000549861	7.21582	94.910	0.000549861	1.821
256	7.2389	154.758	0.000549861	4.64308	147.499	0.000549861	1.559

Lx = Ly = Lz = π, N = 512, K = 2000

Число MPI процессов (P)	MPI			MPI+OpenMP			Ускорение
	Время решения (с)	Ускорение	Погрешность	Время решения (с)	Ускорение	Погрешность	
1	8605.17	1.000	0.000549907	5541.26	1.000	0.000549907	1.553
2	4754.13	1.810	0.000549907	2894.92	1.914	0.000549907	1.642
4	2574.72	3.342	0.000549907	1519.12	3.648	0.000549907	1.695
8	1392.04	6.182	0.000549907	798.428	6.940	0.000549907	1.743
16	728.384	11.814	0.000549907	408.788	13.555	0.000549907	1.782
32	383.046	22.465	0.000549907	210.271	26.353	0.000549907	1.822
64	202.853	42.421	0.000549907	108.978	50.848	0.000549907	1.861
128	103.621	83.045	0.000549907	56.0311	98.896	0.000549907	1.849
256	52.5081	163.883	0.000549907	29.3193	188.997	0.000549907	1.791

Выводы

Задача для трёхмерного гиперболического уравнения в прямоугольном параллелепипеде отлично подходит для распараллеливания. В результате получены программные средства, решающие поставленную задачу как средствами MPI, OpenMP, так и гибридным способом – MPI+OpenMP.

Важную роль в MPI распараллеливании задачи сеточного метода играет способ разбиения на блоки. Проведены эксперименты как для блочного, так и для ленточного способов. Ленточный способ показал хорошие результаты, но не лучшее масштабирование по сравнению с блочным способом. Имеет смысл проверить и другие варианты блочного разбиения, достаточно добавить обработчик разбиения и соответствующий тип.

Увы, передача данных, и, особенно, работа с памятью, по-прежнему остаётся камнем преткновения при распараллеливании программ, что особенно хорошо ощущается при запуске на суперкомпьютере Blue Gene.