



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики



Суперкомпьютерное моделирование и технологии

**Отчет по заданию №3 «Задача для трёхмерного
гиперболического уравнения в прямоугольном
параллелепипеде»**

Вариант №1

студент 628 группы
Гугучкин Егор Павлович

2022 год

1. Математическая 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 $0 \leq t \leq T$ требуется найти решение $u(x, y, z, t)$ уравнения в

частных производных $\frac{\partial^2 u}{\partial t^2} = \Delta u$ с начальными условиями

$$u(t = 0) = \phi(x, y, z)$$

$$\frac{\partial u}{\partial t}(t = 0) = 0$$

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

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

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

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

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

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

2. Численный метод решения задачи

Введем на Ω сетку $\omega_{h\tau} = \overline{\omega_h} \times \omega_\tau$, где $T = T_0$,

$$L_x = L_{x0}, L_y = L_{y0}, L_z = L_{z0},$$

$$\begin{aligned} \overline{\omega_h} &= \{(x_i = ih_x, y_j = jh_y, z_k = kh_z), i, j, k = \overline{0, N}, h_x N = L_x, h_y N \\ &= L_y, h_z N = L_z\}, \end{aligned}$$

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

Через ω_h обозначим множество внутренних, а через γ_h – множество граничных узлов сетки $\overline{\omega_h}$.

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:

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

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

Для начала счета должны быть заданы значения:

$$\begin{aligned} u_{i,j,k}^0, u_{i,j,k}^1, (x_i, y_i, z_i) &\in \omega_h: \\ u_{i,j,k}^0 &= \phi(x_i, y_i, z_i), (x_i, y_i, z_i) \in \omega_h \\ u_{i,j,k}^1 &= u_{i,j,k}^0 + \frac{\tau^2}{2} \Delta_h \phi(x_i, y_i, z_i) \\ u_{i,j,0}^{n+1} &= u_{i,j,N}^{n+1} \\ u_{i,j,1}^{n+1} &= u_{i,j,N+1}^{n+1} \\ i, j, k &= \overline{0, N} \end{aligned}$$

3. Программная реализация

Реализовано две версии программы: последовательная и параллельная с использованием MPI + OpenMP. В качестве входных аргументов задаются следующие переменные: N – количество точек сетки вдоль одной оси, L – длина сетки вдоль одной оси, *filename* – имя выходного файла. На выходе программа выводит N , число MPI-процессов и погрешность полученного решения.

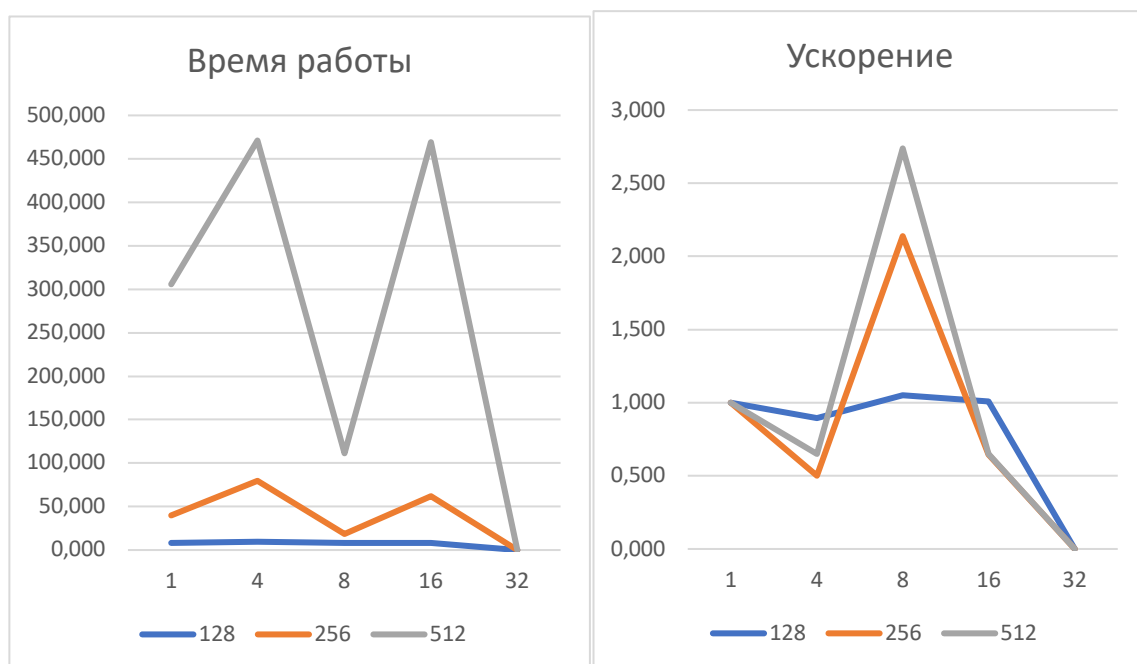
Параллельная версия программы выполнена следующим образом:

1. Сетка разделяется на $size$ блоков, где $size$ – число MPI-процессов. Каждому процессу выделяется свой блок.
2. Процессы находят ранги процессов-соседей и вычисляют координаты границ блока.
3. Процессы вычисляют u_0 и u_1 для своего блока.
4. Процессы вычисляют u_i и обмениваются граничными значениями.
5. Итоговая погрешность редуцируется с помощью оператора MPI_Reduce.

4. Результаты расчетов

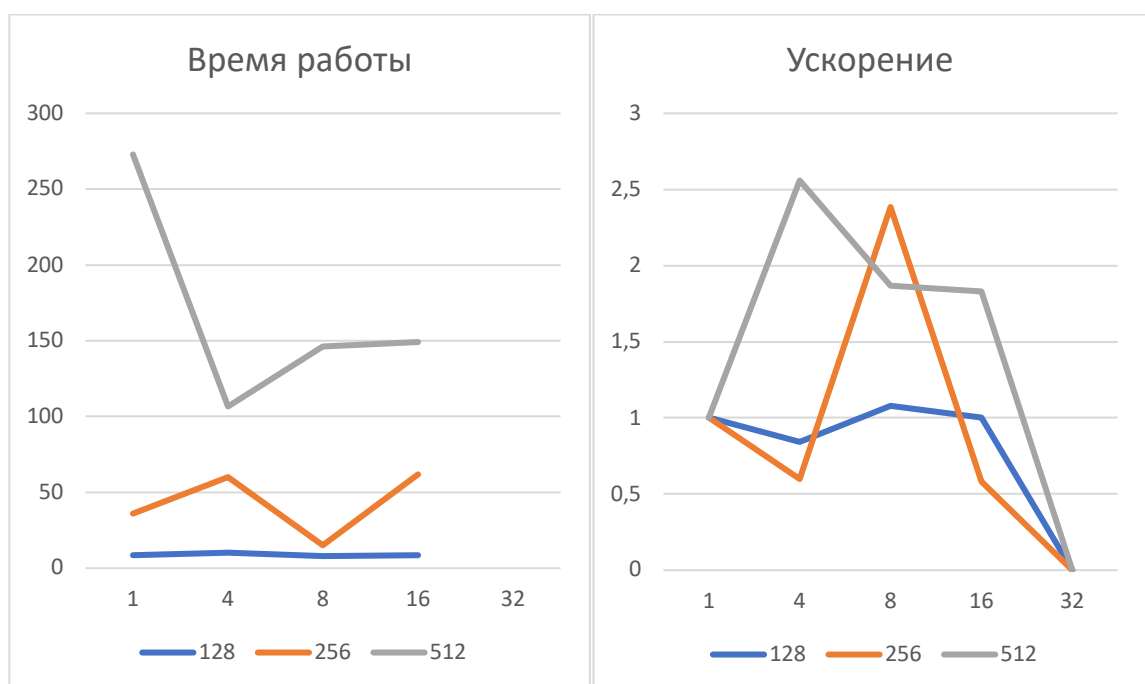
MPI программа; L = 1; сравнение с 1-процессной MPI программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	8,530	1,000	2.82095e-07
4	128^3	9,540	0,894	2.82095e-07
8	128^3	8,118	1,051	2.82095e-07
16	128^3	8,471	1,007	2.82095e-07
32	128^3			2.82095e-07
1	256^3	39,917	1,000	5.9588e-08
4	256^3	79,619	0,501	5.9588e-08
8	256^3	18,668	2,138	5.9588e-08
16	256^3	61,832	0,646	5.9588e-08
32	256^3			5.9588e-08
1	512^3	305,388	1,000	3.96013e-09
4	512^3	471,142	0,648	3.96013e-09
8	512^3	111,547	2,738	3.96013e-09
16	512^3	469,422	0,651	3.96013e-09
32	512^3			3.96013e-09



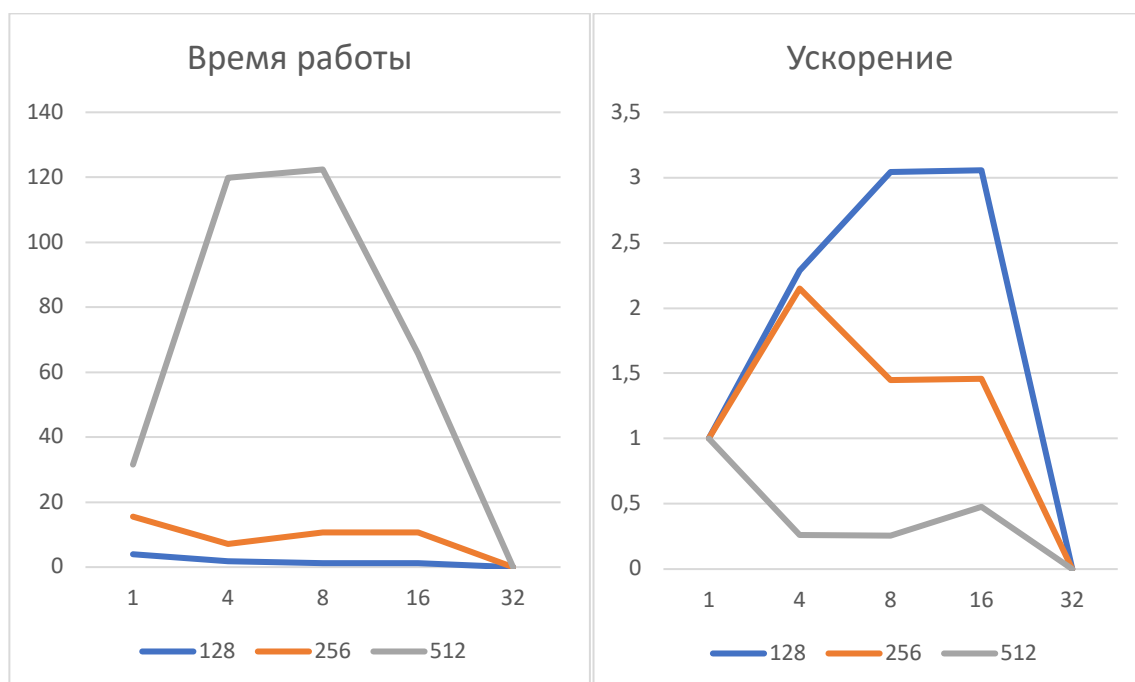
MPI программа; $L = \pi$; сравнение с 1-процессной MPI программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	8,598	1,000	$2.99631e-08$
4	128^3	10,220	0,841	$2.99631e-08$
8	128^3	7,970	1,079	$2.99631e-08$
16	128^3	8,588	1,001	$2.99631e-08$
32	128^3			$2.99631e-08$
1	256^3	36,009	1,000	$7.3784e-09$
4	256^3	60,202	0,598	$7.3784e-09$
8	256^3	15,098	2,385	$7.3784e-09$
16	256^3	61,877	0,582	$7.3784e-09$
32	256^3			$7.3784e-09$
1	512^3	272,890	1,000	$1.73213e-09$
4	512^3	106,651	2,559	$1.73213e-09$
8	512^3	146,194	1,867	$1.73213e-09$
16	512^3	148,981	1,832	$1.73213e-09$
32	512^3			$1.73213e-09$



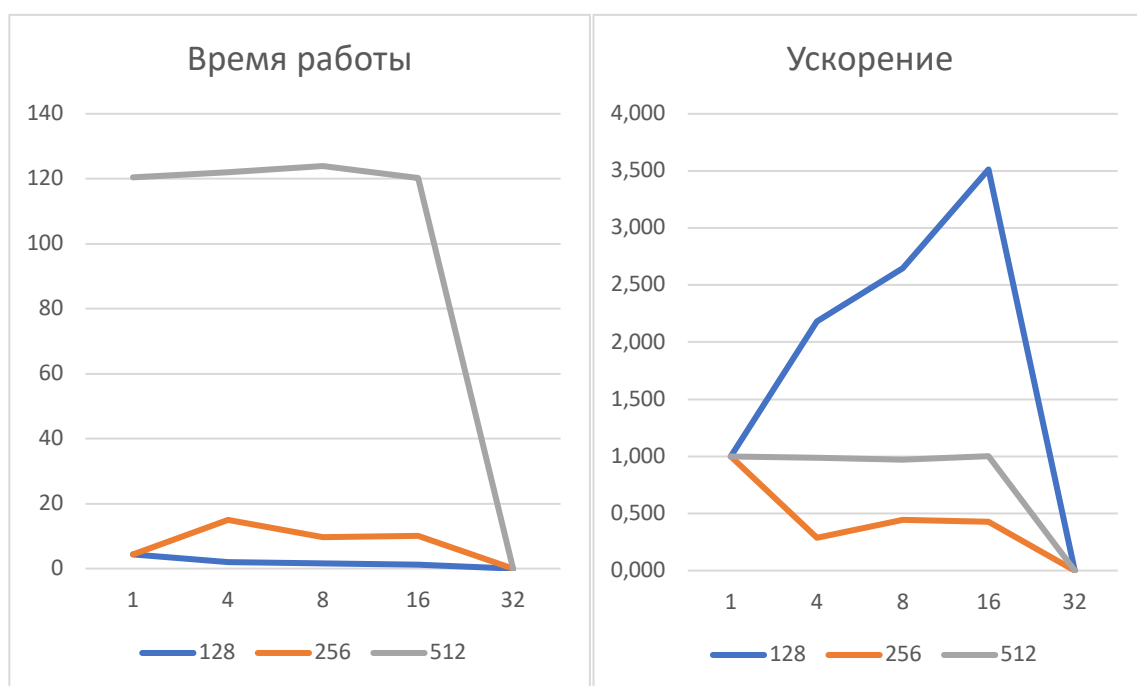
MPI+OpenMP (4 нити) программа; L = 1; сравнение с 1-процессной MPI программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	3,962	1,000	2.82095e-07
4	128^3	1,731	2,288	2.82095e-07
8	128^3	1,303	3,041	2.82095e-07
16	128^3	1,297	3,056	2.82095e-07
32	128^3			2.82095e-07
1	256^3	15,537	1,000	5.9588e-08
4	256^3	7,226	2,150	5.9588e-08
8	256^3	10,742	1,446	5.9588e-08
16	256^3	10,641	1,460	5.9588e-08
32	256^3			5.9588e-08
1	512^3	31,514	1,000	3.96013e-09
4	512^3	119,867	0,263	3.96013e-09
8	512^3	122,411	0,257	3.96013e-09
16	512^3	65,742	0,479	3.96013e-09
32	512^3			3.96013e-09



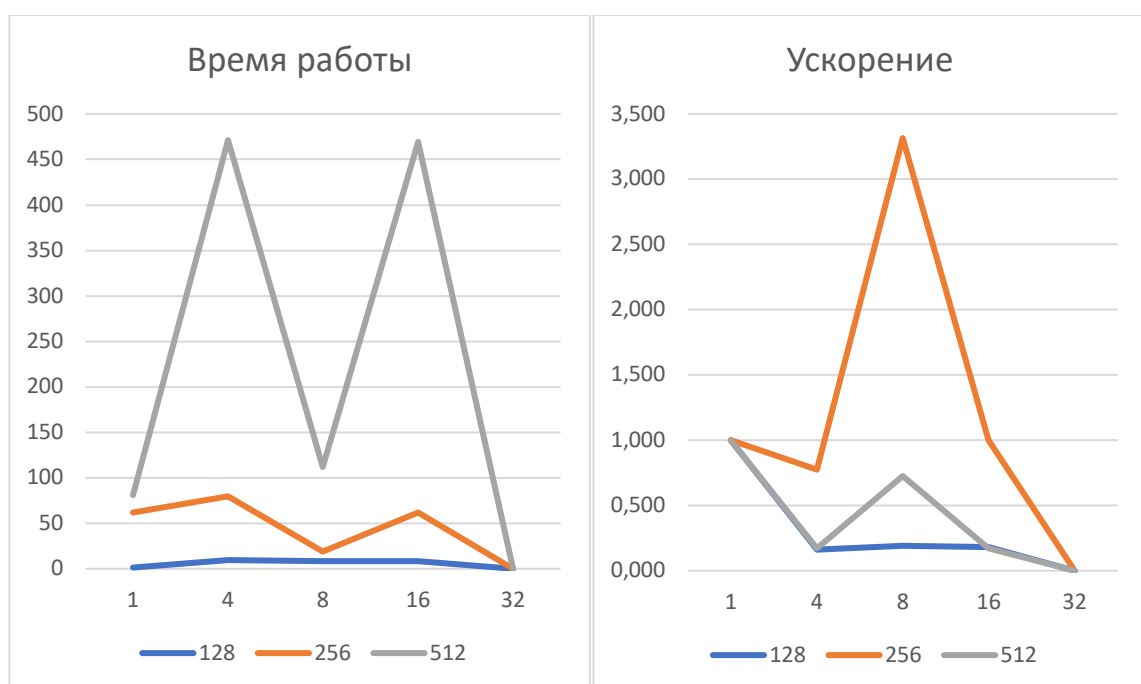
MPI+OpenMP (4 нити) программа; $L = \pi$; сравнение с 1-процессной MPI программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	4,371	1,000	$2.99631e-08$
4	128^3	2,002	2,183	$2.99631e-08$
8	128^3	1,652	2,645	$2.99631e-08$
16	128^3	1,244	3,513	$2.99631e-08$
32	128^3			$2.99631e-08$
1	256^3	4,336	1,000	$7.3784e-09$
4	256^3	14,992	0,289	$7.3784e-09$
8	256^3	9,691	0,447	$7.3784e-09$
16	256^3	10,145	0,427	$7.3784e-09$
32	256^3			$7.3784e-09$
1	512^3	120,372	1,000	$1.73213e-09$
4	512^3	121,940	0,987	$1.73213e-09$
8	512^3	123,924	0,971	$1.73213e-09$
16	512^3	120,149	1,002	$1.73213e-09$
32	512^3			$1.73213e-09$



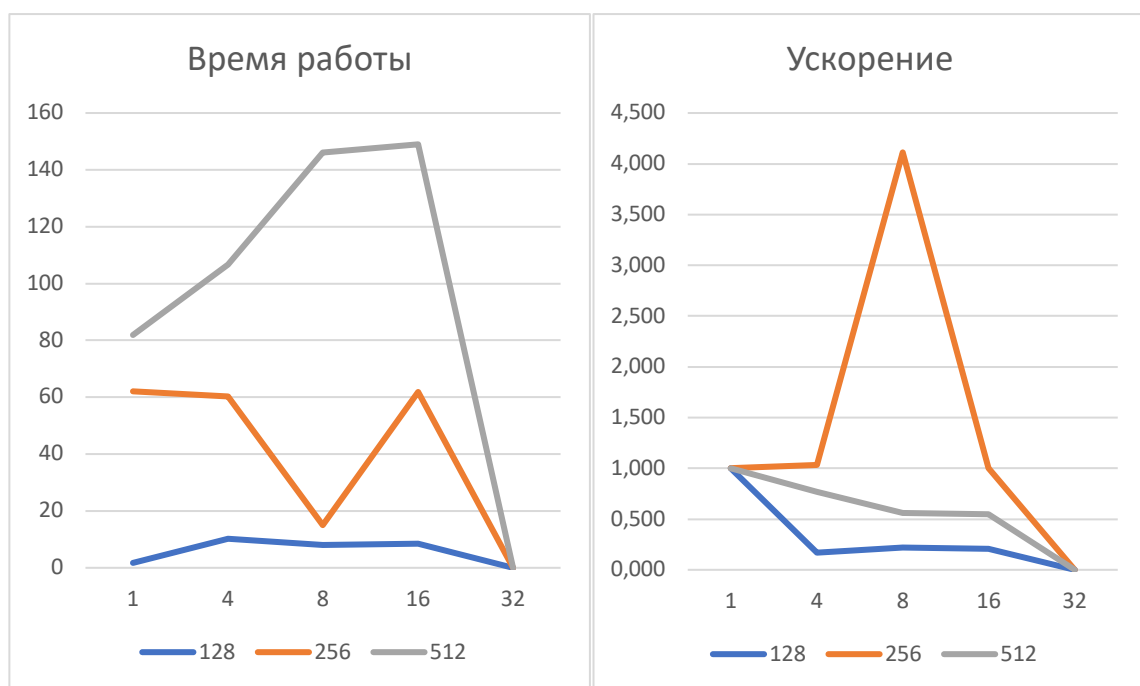
MPI программа; $L = 1$; сравнение с 1-процессной последовательной программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	1,553	1,000	2.82095e-07
4	128^3	9,540	0,163	2.82095e-07
8	128^3	8,118	0,191	2.82095e-07
16	128^3	8,471	0,183	2.82095e-07
32	128^3			2.82095e-07
1	256^3	61,881	1,000	5.9588e-08
4	256^3	79,619	0,777	5.9588e-08
8	256^3	18,668	3,315	5.9588e-08
16	256^3	61,832	1,001	5.9588e-08
32	256^3			5.9588e-08
1	512^3	80,990	1,000	3.96013e-09
4	512^3	471,142	0,172	3.96013e-09
8	512^3	111,547	0,726	3.96013e-09
16	512^3	469,422	0,173	3.96013e-09
32	512^3			3.96013e-09



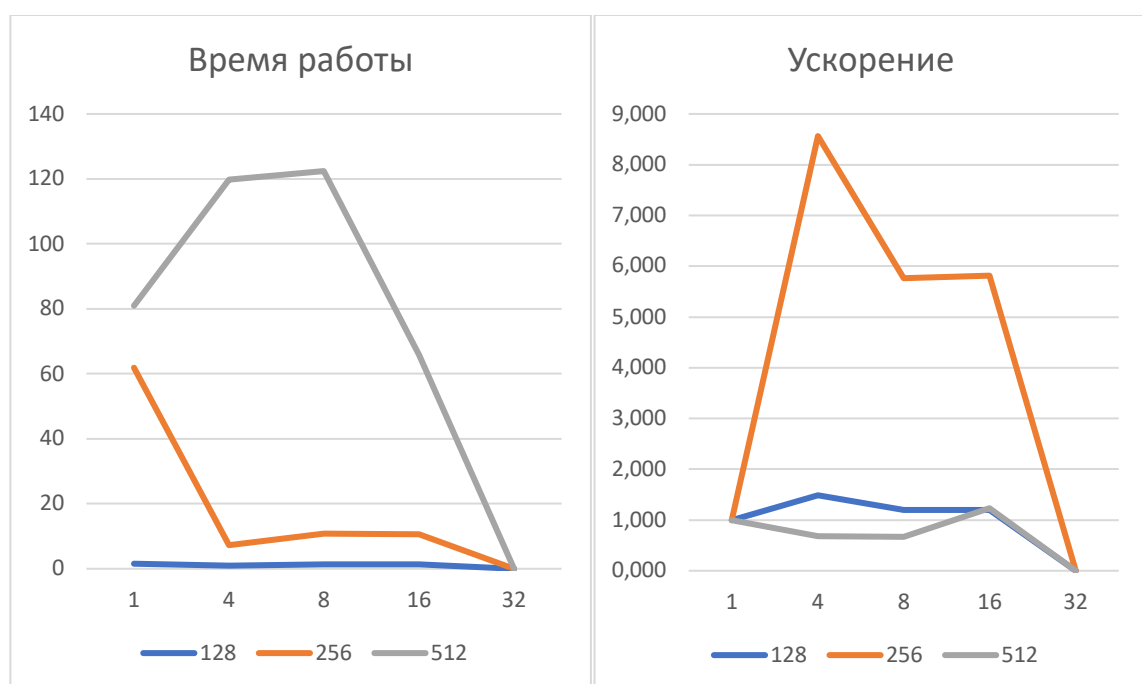
MPI программа; $L = \pi$; сравнение с 1-процессной последовательной программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	1,749	1,000	$2.99631e-08$
4	128^3	10,220	0,171	$2.99631e-08$
8	128^3	7,970	0,219	$2.99631e-08$
16	128^3	8,588	0,204	$2.99631e-08$
32	128^3			$2.99631e-08$
1	256^3	62,084	1,000	$7.3784e-09$
4	256^3	60,202	1,031	$7.3784e-09$
8	256^3	15,098	4,112	$7.3784e-09$
16	256^3	61,877	1,003	$7.3784e-09$
32	256^3			$7.3784e-09$
1	512^3	81,881	1,000	$1.73213e-09$
4	512^3	106,651	0,768	$1.73213e-09$
8	512^3	146,194	0,560	$1.73213e-09$
16	512^3	148,981	0,550	$1.73213e-09$
32	512^3			$1.73213e-09$



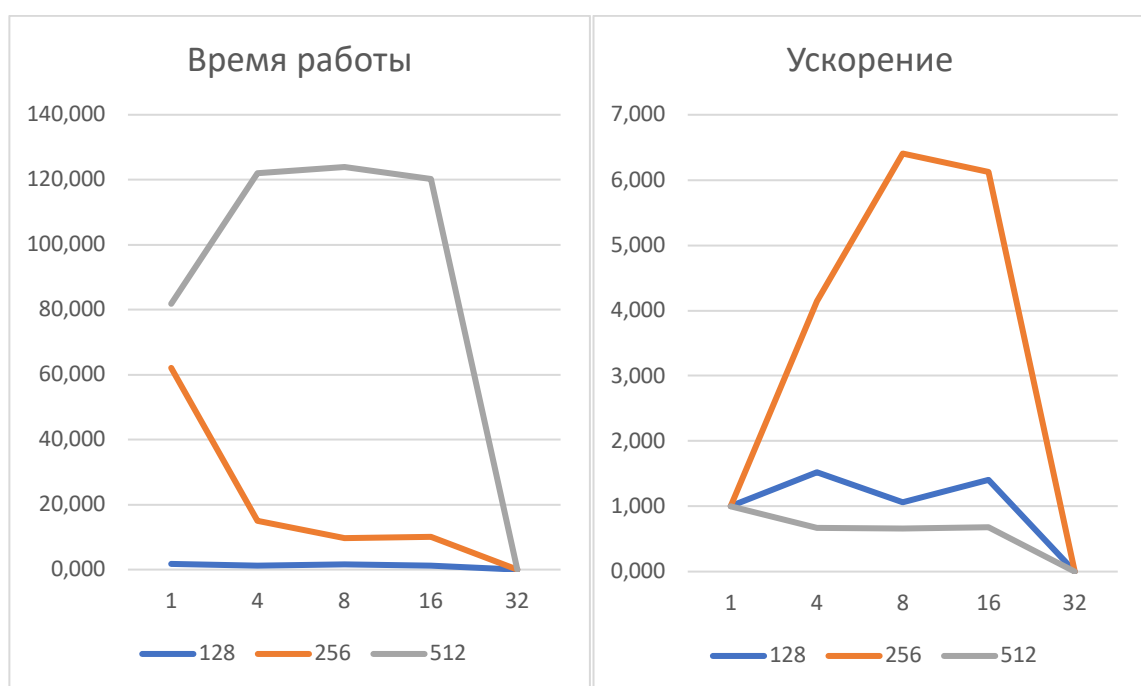
MPI+OpenMP (4 нити) программа; L = 1; сравнение с 1-процессной последовательной программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	1,553	1,000	2.82095e-07
4	128^3	1,045	1,487	2.82095e-07
8	128^3	1,303	1,192	2.82095e-07
16	128^3	1,297	1,198	2.82095e-07
32	128^3			2.82095e-07
1	256^3	61,881	1,000	5.9588e-08
4	256^3	7,226	8,564	5.9588e-08
8	256^3	10,742	5,761	5.9588e-08
16	256^3	10,641	5,815	5.9588e-08
32	256^3			5.9588e-08
1	512^3	80,990	1,000	3.96013e-09
4	512^3	119,867	0,676	3.96013e-09
8	512^3	122,411	0,662	3.96013e-09
16	512^3	65,742	1,232	3.96013e-09
32	512^3			3.96013e-09



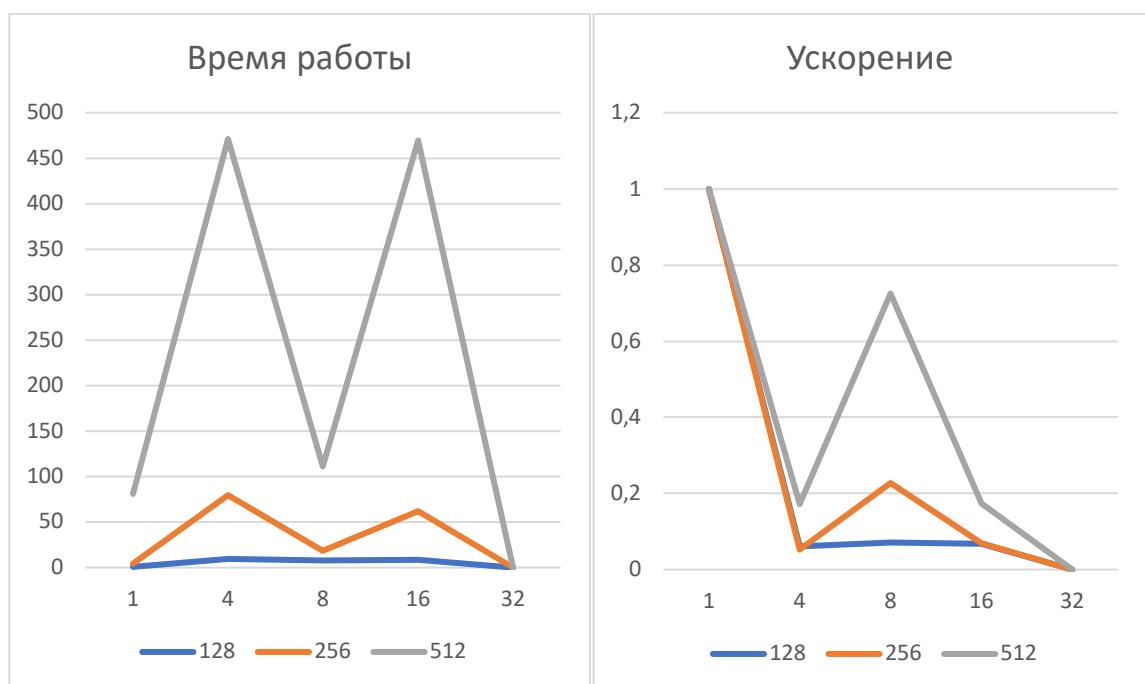
MPI+OpenMP (4 нити) программа; $L = \pi$; сравнение с 1-процессной последовательной программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	1,749	1,000	2.99631e-08
4	128^3	1,149	1,522	2.99631e-08
8	128^3	1,652	1,058	2.99631e-08
16	128^3	1,244	1,406	2.99631e-08
32	128^3			2.99631e-08
1	256^3	62,084	1,000	7.3784e-09
4	256^3	14,992	4,141	7.3784e-09
8	256^3	9,691	6,406	7.3784e-09
16	256^3	10,145	6,120	7.3784e-09
32	256^3			7.3784e-09
1	512^3	81,881	1,000	1.73213e-09
4	512^3	121,940	0,671	1.73213e-09
8	512^3	123,924	0,661	1.73213e-09
16	512^3	120,149	0,681	1.73213e-09
32	512^3			1.73213e-09



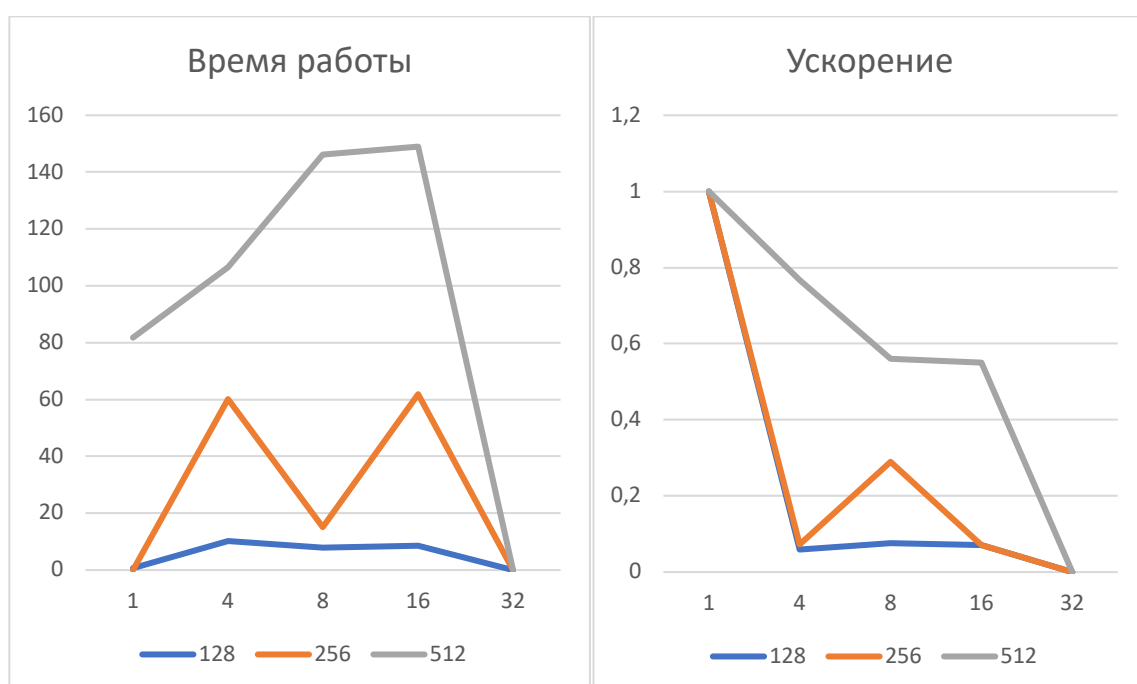
MPI программа; $L = 1$; сравнение с последовательной OpenMP (4 нити) программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	0,578	1,000	2.82095e-07
4	128^3	9,540	0,061	2.82095e-07
8	128^3	8,118	0,071	2.82095e-07
16	128^3	8,471	0,068	2.82095e-07
32	128^3			2.82095e-07
1	256^3	4,255	1,000	5.9588e-08
4	256^3	79,619	0,053	5.9588e-08
8	256^3	18,668	0,228	5.9588e-08
16	256^3	61,832	0,069	5.9588e-08
32	256^3			5.9588e-08
1	512^3	80,990	1,000	3.96013e-09
4	512^3	471,142	0,172	3.96013e-09
8	512^3	111,547	0,726	3.96013e-09
16	512^3	469,422	0,173	3.96013e-09
32	512^3			3.96013e-09



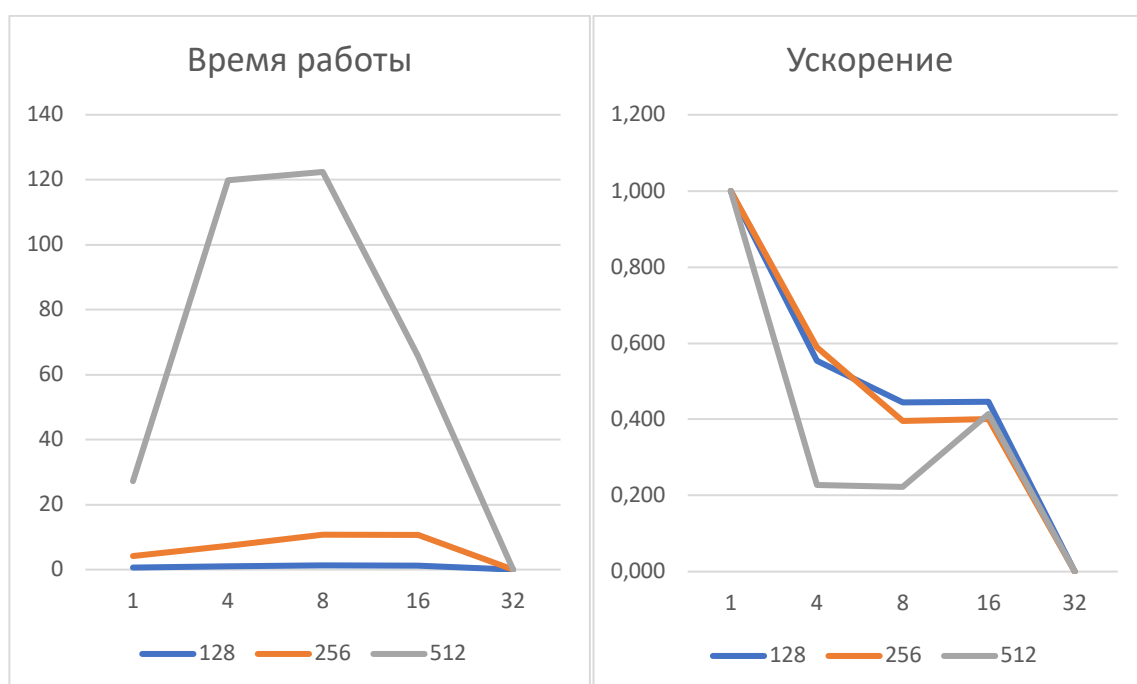
MPI программа; $L = \pi$; сравнение с последовательной OpenMP (4 нити) программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	0,609	1,000	$2.99631e-08$
4	128^3	10,220	0,060	$2.99631e-08$
8	128^3	7,970	0,076	$2.99631e-08$
16	128^3	8,588	0,071	$2.99631e-08$
32	128^3			$2.99631e-08$
1	256^3	4,369	1,000	$7.3784e-09$
4	256^3	60,202	0,073	$7.3784e-09$
8	256^3	15,098	0,289	$7.3784e-09$
16	256^3	61,877	0,071	$7.3784e-09$
32	256^3			$7.3784e-09$
1	512^3	81,881	1,000	$1.73213e-09$
4	512^3	106,651	0,768	$1.73213e-09$
8	512^3	146,194	0,560	$1.73213e-09$
16	512^3	148,981	0,550	$1.73213e-09$
32	512^3			$1.73213e-09$



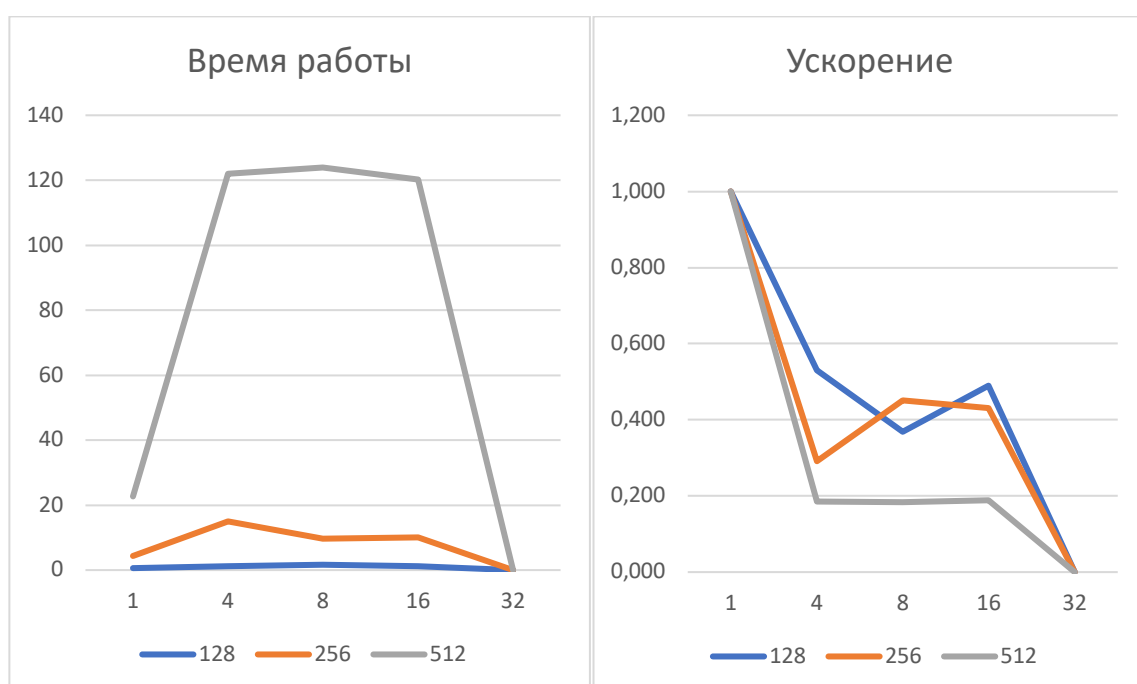
MPI+OpenMP (4 нити) программа; L = 1; сравнение с последовательной OpenMP (4 нити) программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	0,578	1,000	2.82095e-07
4	128^3	1,045	0,553	2.82095e-07
8	128^3	1,303	0,444	2.82095e-07
16	128^3	1,297	0,446	2.82095e-07
32	128^3			2.82095e-07
1	256^3	4,255	1,000	5.9588e-08
4	256^3	7,226	0,589	5.9588e-08
8	256^3	10,742	0,396	5.9588e-08
16	256^3	10,641	0,400	5.9588e-08
32	256^3			5.9588e-08
1	512^3	27,244	1,000	3.96013e-09
4	512^3	119,867	0,227	3.96013e-09
8	512^3	122,411	0,223	3.96013e-09
16	512^3	65,742	0,414	3.96013e-09
32	512^3			3.96013e-09



MPI+OpenMP (4 нити) программа; $L = \pi$; сравнение с последовательной OpenMP (4 нити) программой

Число MPI-процессов N_p	Число точек сетки N^3	Время решения T	Ускорение S	Погрешность σ
1	128^3	0,609	1,000	2.99631e-08
4	128^3	1,149	0,530	2.99631e-08
8	128^3	1,652	0,369	2.99631e-08
16	128^3	1,244	0,490	2.99631e-08
32	128^3			2.99631e-08
1	256^3	4,369	1,000	7.3784e-09
4	256^3	14,992	0,291	7.3784e-09
8	256^3	9,691	0,451	7.3784e-09
16	256^3	10,145	0,431	7.3784e-09
32	256^3			7.3784e-09
1	512^3	22,725	1,000	1.73213e-09
4	512^3	121,940	0,186	1.73213e-09
8	512^3	123,924	0,183	1.73213e-09
16	512^3	120,149	0,189	1.73213e-09
32	512^3			1.73213e-09



5. Выводы

По полученным результатам, можно сделать вывод, что последовательная + OpenMP программа работает лучше, чем MPI+OpenMP программа. Возможно, это связано с тем, что при таком объеме данных тестирование уязвимо к выбросам.

При этом не получилось добиться однозначных результатов для распараллеливания MPI программы. Для разных ситуаций самыми эффективными могут быть распараллеливания на 4, 8, или 16 процессов.