



Московский Государственный Университет имени М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики

Курс “Современные высокопроизводительные вычисления”

Отчет по практическому заданию

Русецкий Илья Владиславович,
группа 628 (228_кт)

23 ноября 2024 года

1. Описание метода решения.

Решение представлено в файле `openmp_prog.c`. Входные данные заданы в виде констант N – количество точек в разбиении и $L(L_x, L_y, L_z)$ – размер области, в которой решается заданное уравнение. Алгоритм реализован в функции `main()`.

Данный алгоритм заключается в следующем: выделяется память для хранения трех массивов с различными состояниями решения ($Un-2$, $Un-1$, Un), вычисляются начальные состояния этого решения, а затем несколько итераций осуществляется пересчет решения согласно разностной схеме и вычисление ошибки для этой итерации.

Пересчет решения осуществляется функцией `step()`, которая представляет собой три вложенных цикла по i , j , k . Для каждой координаты (i, j, k) вычисляется значение элемента $Un[i][j][k]$, зависящее от полученных на предыдущих шагах значений: $Un[i][j][k] = 2 * Un1[i][j][k] - Un2[i][j][k] + \text{pow}(\tau, 2) * \text{laplas_operator}(Un1, i, j, k)$.

Вычисление ошибки осуществляется функцией `calculate_error`, которая представляет собой три вложенных цикла по i , j , k . Для каждой координаты (i, j, k) вычисляется значение элемента разницы между $Un[i][j][k]$ – текущим решением и аналитическим значением функции на данном шаге. Ошибка будет максимальным значением этой разницы по всем наборам i, j, k .

При запуске последовательных версий программы для $N=128, 256, 512$ были получены следующие результаты:

Последовательная		
N	Погрешность решения	Время, сек.
128	0,0000000007430611682	18,3456
256	0,0000000001856991228	138,8513
512	0.0000000004635691830	944,5499

2. OpenMP версия алгоритма.

Для параллельного вычисления решения с использованием технологии OpenMP к последовательной версии были добавлены следующие директивы:

1. `#pragma omp parallel for default(shared) private(i, j, k)` для трех вложенных циклов в функциях `init()` и `step()`. Переменные `i`, `j`, `k` сделаны приватными, так как это счетчики циклов и должны иметь разные копии у различных нитей. За остальные переменные нити не конкурируют поэтому они объявлены `shared`.
2. `#pragma omp parallel for default(shared) private(i, j, k) reduction(max: error)` для циклов в `calculate_error()`. В отличие от предыдущего случая тут добавляется переменная `error`, по которой по завершении цикла необходимо произвести редукцию: собрать `error` со всех нитей и присвоить в итоговый `error` максимальное значение.

При различных запусках данной программы на машине IBM Polus были получены следующие значения:

OpenMP				
N	Погрешность решения	Кол-во нитей	Время, сек.	Ускорение
128	0,0000000007430611682	1	17,5818	1,04
		8	5,8493	3,14
		16	3,9089	4,69
		32	5,6228	3,26
		64	8,0422	2,28
		128	12,4312	1,48
256	0,0000000001856991228	1	109,737	1,27
		8	26,9267	5,16
		16	21,4725	6,47
		32	23,466	5,92
		64	17,1714	8,09
		128	29,9271	4,64
512	0.00000000004635691830	1	860,7239	1,1
		8	160,5676	5,88
		16	145,5866	6,49
		32	154,9465	6,1
		64	180,7279	5,23
		128	219,6021	4,3

3. MPI версия алгоритма.

При различных запусках данной программы на машине IBM Polus были получены следующие значения:

MPI					
N	Погрешность решения	Кол-во ядер	Время, сек.	Ускорение	
128	0,0000000007430611682	1	11,9725	1,53	
		4	3,3042	5,55	
		8	1,7874	10,26	
		10	1,4169	12,95	
		16	1,0728	17,1	
		20	0,859	21,36	
256	0,0000000001856991228	1	97,9989	1,42	
		4	25,3111	5,49	
		8	13,3354	10,41	
		10	10,7671	12,9	
		16	7,888	17,6	
		20	9,1718	15,14	
512	0.00000000004635691830	1	794,1498	1,19	
		4	207,8048	4,55	
		8	110,1417	8,58	
		10	80,9568	11,67	
		16	62,7072	15,06	
		20	44,3977	21,27	

4. MPI + OpenMP версия алгоритма.

При различных запусках данной программы на машине IBM Polus на двух GPU были получены следующие значения:

MPI+OpenMP					
N	Погрешность решения	Кол-во ядер	Кол-во нитей	Время, сек,	Ускорение
128	0,0000000007430611682	1	4	1,4131	12,98
		4	4	0,5431	33,78
256	0,0000000001856991228	1	4	11,3198	12,27
		4	4	4,271	32,51
512	0.00000000004635691830	1	4	106,1533	8,9
		4	4	34,0596	27,73

5. MPI + CUDA версия алгоритма

При различных запусках данной программы на машине IBM Polus на двух GPU были получены следующие значения:

CUDA				
N	Погрешность решения	Кол-во ядер	Время, сек,	Ускорение
128	0,0000000007430611682	1	0,5342	34,34
		4	0,3816	48,08
		8	0,217	84,54
		10	0,1669	109,92
		16	0,3101	59,16
256	0,0000000001856991228	1	2,5774	53,87
		4	0,782	177,56
		8	0,478	290,48
		10	0,37	375,27
		16	0,5155	269,35
512	0.00000000004635691830	1	23,3902	40,38
		4	6,811	138,68
		8	3,8393	246,02
		10	3,0928	305,4
		16	2,5357	372,5