



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 алгоритма решения.

Последовательное решение представлено в файле prog.c. Входные данные заданы в виде констант N – количество точек в разбиении и L(Lx, Ly, Lz) – 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 разница 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, L=1		
N	Погрешность решения	Время, сек.
128	0,0000000007430611682	18,3456
256	0,0000000001856991228	138,8513
512	0.00000000004635691830	944,5499

Последовательная, L=pi		
N	Погрешность решения	Время, сек.
128	0.00000000007529632473	17,996
256	0.00000000001882494161	143,1511
512	0.00000000000470901096	929,4426

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 Polus для $L = 1$ 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	0.0000000004635691830	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

При различных запусках данной программы на машине Polus для $L = p_i$ были получены следующие значения:

OpenMP, L=pi				
N	Погрешность решения	Кол-во нитей	Время, сек.	Ускорение
128	0.00000000007529632473	1	17,0076	1,06
		8	6,2374	2,89
		16	3,7895	4,75
		32	5,5994	3,21
		64	8,5623	2,1
		128	12,7789	1,41
256	0.00000000001882494161	1	112,9403	1,27
		8	24,8339	5,76
		16	20,9453	6,83
		32	19,8634	7,21
		64	23,8523	6
		128	34,9875	4,09
512	0.00000000000470901096	1	845,3031	1,1
		8	154,7829	6
		16	141,6	6,56
		32	150,6932	6,17
		64	180,552	5,15
		128	209,6271	4,43

3. MPI версия алгоритма.

При различных запусках данной программы на машине IBM Polus для L=1 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

При различных запусках данной программы на машине IBM Polus для L=pi были получены следующие значения:

MPI, L=pi				
N	Погрешность решения	Кол-во ядер	Время, сек.	Ускорение
128	0.000000000007529632473	1	12,1323	1,51
		4	3,2986	5,56
		8	1,8804	9,76
		10	1,3075	14,03
		16	1,1036	16,62
		20	0,74	24,79
256	0.000000000001882494161	1	100,0015	1,39
		4	24,5897	5,65
		8	13,2199	10,5
		10	10,7678	12,9
		16	8,351	16,63
		20	9,1523	15,17
512	0.000000000000470901096	1	802,2568	1,18
		4	196,9542	4,8
		8	102,8645	9,18
		10	78,9103	11,97
		16	61,634	15,33
		20	45,6749	20,68

4. MPI + OpenMP версия алгоритма.

При различных запусках данной программы на машине IBM Polus для L=1 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
		8	4	3,522	5,21
		16	4	3,8743	4,74
256	0,0000000001856991228	1	4	11,3198	12,27
		4	4	4,271	32,51
		8	4	19,1762	7,24
		16	4	18,0402	7,7
512	0.00000000004635691830	1	4	106,1533	8,9
		4	4	34,0596	27,73
		8	4	61,4045	15,38
		16	4	58,1334	16,25
MPI+OpenMP					
N	Погрешность решения	Кол-во ядер	Кол-во нитей	Время, сек,	Ускорение
128	0,0000000007430611682	1	20	0,8988	20,41
		1	40	1,6568	11,07
		2	20	0,4593	39,94
		2	40	0,9581	19,15
256	0,0000000001856991228	1	20	6,8687	20,22
		1	40	11,3491	12,23
		2	20	4,9993	27,77
		2	40	6,7723	20,5
512	0.00000000004635691830	1	20	54,8015	17,24
		1	40	82,5759	11,44
		2	20	29,1649	32,39
		2	40	47,6268	19,83

5. MPI + CUDA версия алгоритма

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

CUDA				
128	0,0000000007430611682	1 mpi и 1 gpu	0,1761	104,18
		2 mpi и 2 gpu	0,0882	208
256	0,0000000001856991228	1 mpi и 1 gpu	2,27683	60,98
		2 mpi и 2 gpu	1,82257	76,18
512	0.00000000004635691830	1 mpi и 1 gpu	21,84649	43,24
		2 mpi и 2 gpu	16,5271	57,15

Для 2 GPU и различного количества ядер:

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	0.0000000004635691830	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