



Московский Государственный Университет им. М.В. Ломоносова Факультет Вычислительной
Математики и Кибернетики Кафедра Суперкомпьютеров и Квантовой Информатики

*Параллельные методы
решения задач.*

**Практическое задание 2.
Параллельная реализация операций с
сеточными данными на
неструктурированной смешанной
сетке с использованием MPI.**

Афанасьев Виталий Игоревич
510 группа

Москва, 2021

Описание задания и программной реализации

Описание задания: Реализация алгоритма построения двумерного графа и представления его в формате CSR. Реализация многопоточной версии солвера для СЛАУ с разреженной матрицей.

Описание программной реализации:

Функции графа:

`void gen_graph (int K, int *offset_elements, int *elements, int Nx, int Ny, int K1, int K2)` – генерирует граф из параметров Nx, Ny, K1, K2 (K – размер массива offset_elements, offset_elements, elements – представление графа, Nx – количество элементов по горизонтали, Ny – количество элементов по вертикали, K1 – количество неразделенных элементов, K2 – количество поделенных элементов)

`void graph2csr (int K, int offset_elements[], int elements[], int count_edges, int count_nodes, int IA[], int JA[])` – трансформация графа в CSR формат (K – размер массива offset_elements, offset_elements, elements – представление графа, count_edges – количество ребер, count_nodes – количество узлов, IA, JA – портрет разреженной матрицы)

Функции солвера:

`double dot(int N, double x[], double y[])` - скалярное произведение (N – размер векторов x и y)

`void axpby(int N, double a, double x[], double b, double y[]);` - линейная комбинация двух векторов – axpby ($x = ax + by$), x, y - вектора, a, b – скалярные значения (N – размер векторов x и y)

`void VWbe(int N, double x[], double y[], double z[])` - поэлементное умножение векторов $z[i] = x[i] * y[i]$ (N – размер векторов x, y и z)

`void SpMV(int N, int IA[], int JA[], double A[], double b[], double c[])` - матрично-векторное произведение с разреженной матрицей (N – размер массива IA, IA, JA – портрет матрицы, A – массив ненулевых коэффициентов матрицы (размера IA[N]), b – вектор, на который умножается матрица, c – результат умножения)

Компиляция:

“make task2” или “mpicc -O3 -Wall ...”

Запуск:

```
mpirun -np <np> task2 [--help]
mpirun -np <np> task2 --Nx <Nx> --Ny <Ny> --K1 <K1> --K2 <K2> -n <n> --maxiter <N> --tol <tol> [--file <filename>]
```

options:

<code>--help</code>	Show this screen.
<code>--Nx</code>	Count of elements in horizontal
<code>--Ny</code>	Count of elements in vertical
<code>--K1</code>	Count of squares
<code>--K2</code>	Count of cut squares
<code>--Px</code>	Count of parts by x
<code>--Py</code>	Count of parts by y
<code>--maxiter</code>	Maximum of count steps in solver

--tol Relative tolerance for termination.
[--filename] Debug mode

Исследование производительности

Характеристики вычислительной системы:

тип процессора: Intel Core i7-9750H

количество ядер: 6

количество потоков: 12

TPP:

Base:

- 2.6 GHz
- 20,8 GFLOPS / core
- 249,6 GFLOPS / computer

Turbo:

- 4.5GHz
- 36 GFLOPS / core
- 432 GFLOPS / computer

BW: 41.8 GB/s

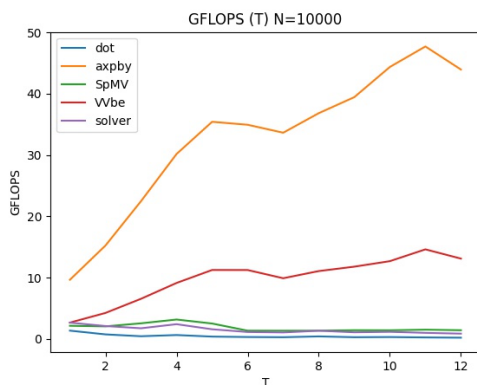
TBP: 1.74 GFLOP = 0.7% TPP

Параметры компиляции: -O3

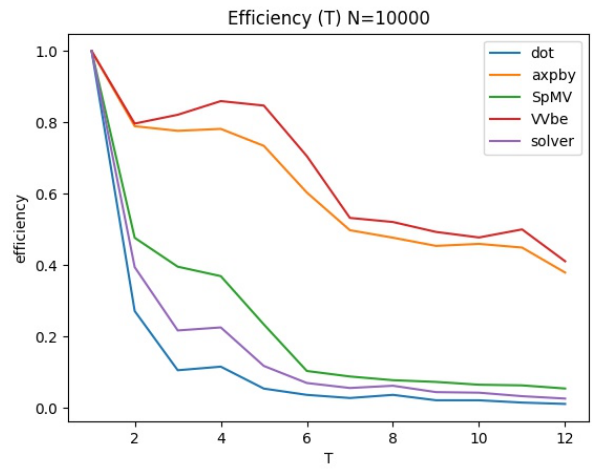
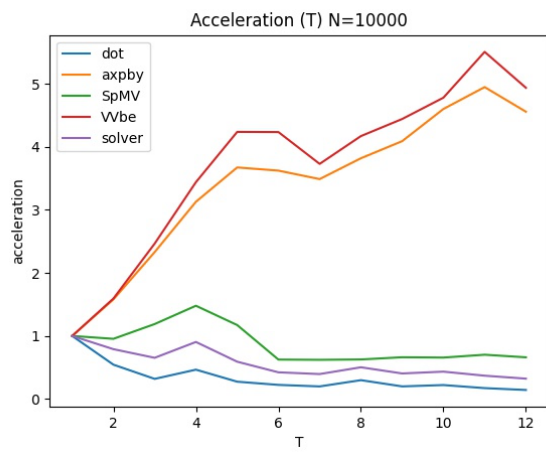
Результаты измерений производительности:

Параллельная реализация, ускорение:

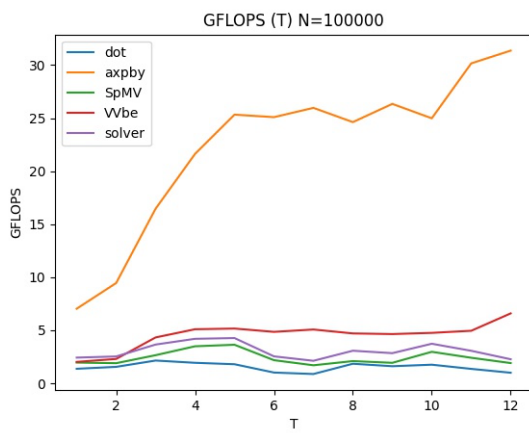
N = 10000



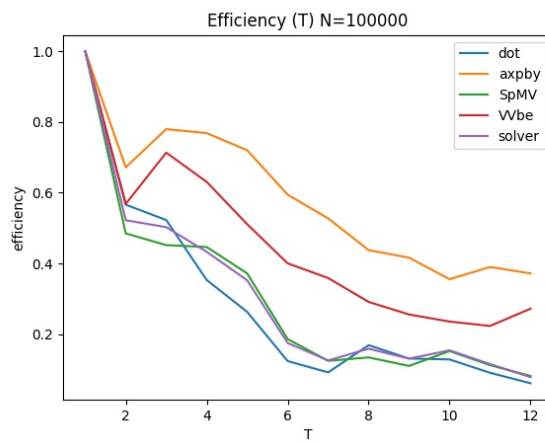
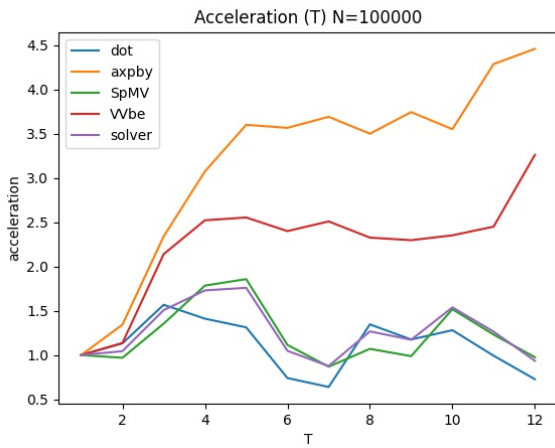
T	dot	axpby	SpMV	VVbe	solver
1	1,34	9,64	2,13	2,65	2,65
2	0,73	15,22	2,03	4,23	2,09
3	0,43	22,46	2,53	6,53	1,73
4	0,62	30,16	3,15	9,12	2,39
5	0,37	35,41	2,50	11,23	1,56
6	0,30	34,92	1,33	11,23	1,12
7	0,27	33,62	1,32	9,89	1,05
8	0,40	36,81	1,34	11,06	1,33
9	0,27	39,43	1,41	11,78	1,07
10	0,30	44,35	1,40	12,67	1,15
11	0,23	47,69	1,50	14,60	0,98
12	0,19	43,93	1,41	13,09	0,85



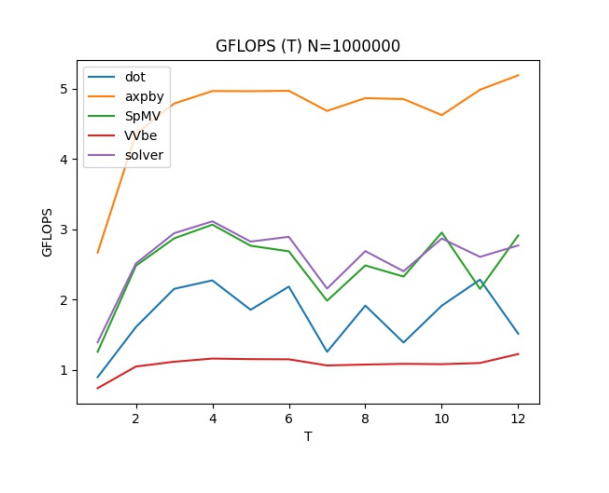
N = 100000



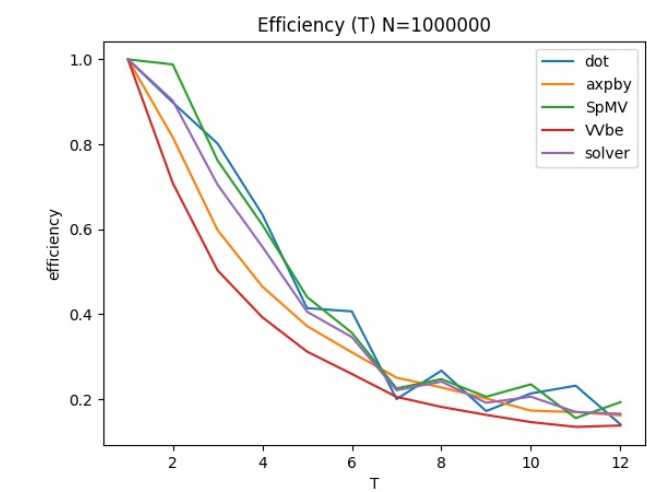
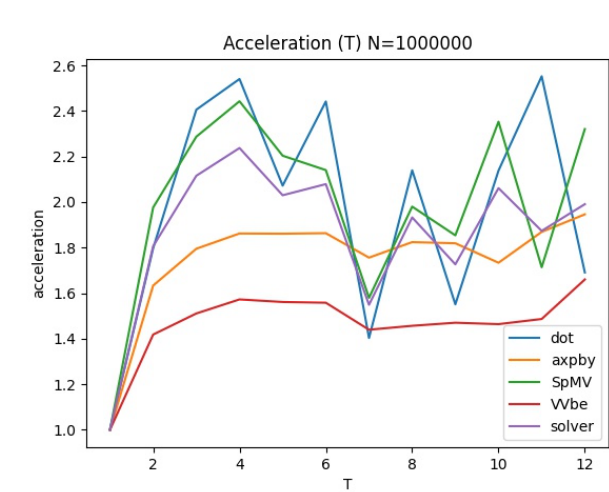
T	dot	axpby	SpMV	VVbe	solver
1	1,37	7,03	1,96	2,02	2,42
2	1,55	9,45	1,90	2,30	2,53
3	2,15	16,46	2,65	4,33	3,65
4	1,93	21,63	3,49	5,10	4,19
5	1,80	25,33	3,63	5,16	4,27
6	1,01	25,09	2,18	4,85	2,54
7	0,88	25,97	1,70	5,07	2,12
8	1,84	24,63	2,09	4,70	3,07
9	1,61	26,34	1,93	4,64	2,84
10	1,75	24,99	2,97	4,75	3,73
11	1,36	30,16	2,41	4,95	3,06
12	1,00	31,37	1,91	6,59	2,27



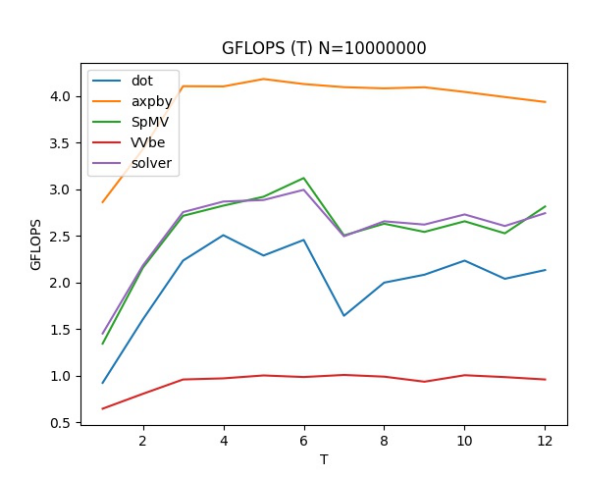
N = 1000000



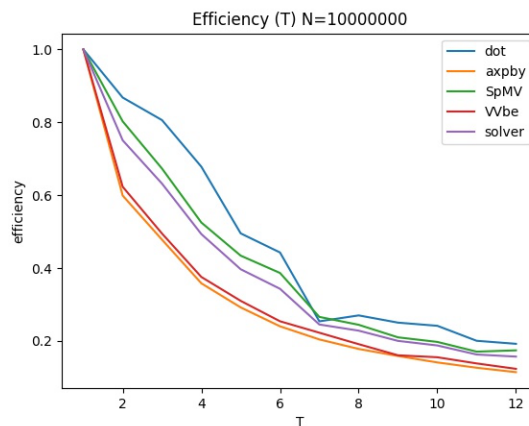
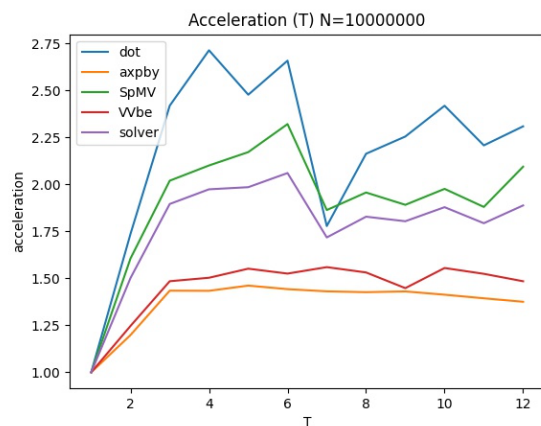
T	dot	axpby	SpMV	VVbe	solver
1	0,89	2,67	1,25	0,74	1,39
2	1,61	4,35	2,48	1,04	2,51
3	2,15	4,78	2,87	1,11	2,94
4	2,27	4,96	3,06	1,16	3,11
5	1,85	4,96	2,76	1,15	2,82
6	2,18	4,97	2,68	1,15	2,89
7	1,25	4,68	1,98	1,06	2,15
8	1,91	4,86	2,48	1,07	2,69
9	1,39	4,85	2,32	1,08	2,40
10	1,91	4,62	2,95	1,08	2,87
11	2,28	4,98	2,15	1,09	2,60
12	1,51	5,19	2,91	1,22	2,77



N = 10000000



T	dot	axpby	SpMV	VVbe	solver
1	0,92	2,86	1,34	0,65	1,45
2	1,60	3,43	2,16	0,81	2,18
3	2,24	4,10	2,72	0,96	2,75
4	2,51	4,10	2,82	0,97	2,87
5	2,29	4,18	2,92	1,00	2,88
6	2,46	4,13	3,12	0,99	2,99
7	1,64	4,09	2,51	1,01	2,50
8	2,00	4,08	2,63	0,99	2,66
9	2,08	4,09	2,54	0,94	2,62
10	2,23	4,04	2,66	1,01	2,73
11	2,04	3,99	2,53	0,99	2,61
12	2,13	3,94	2,82	0,96	2,74



Анализ полученных результатов

Максимальная достигаемая производительность для параллельной реализации для (N=10000000; T=12):

dot: 2.13 GFLOPS, 122% TBP

axpby: 3.94 GFLOPS, 226% TBP

SpMV: 2.82 GFLOPS, 162% TBP

VVbe: 0.96 GFLOPS, 55% TBP

Solver: 2.74 GFLOPS, 157% TBP