



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

*Параллельные методы
решения задач.*

Практическое задание 1. Параллельная реализация операций с сеточными данными на неструктурированной смешанной сетке.

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 main” или “gcc -O3 -Wall -o main main.c lib.c -fopenmp”

Запуск:

```
main.exe [--help]
main.exe --Nx <Nx> --Ny <Ny> --K1 <K1> --K2 <K2> -n <n> --maxiter <N> --tol <tol> [--
file <filename>]
options:\n\
[--help]      Show this screen.
--Nx          Count of elements in horizontal
--Ny          Count of elements in vertical
--K1          Count of squares
--K2          Count of cut squares
-n            Count of threads
--maxiter     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

пиковая производительность:

базовая 2.6 / турбо 4.5GHz

3.81-5.11 GFLOPS / core

45.08-61.26 GFLOPS / computer

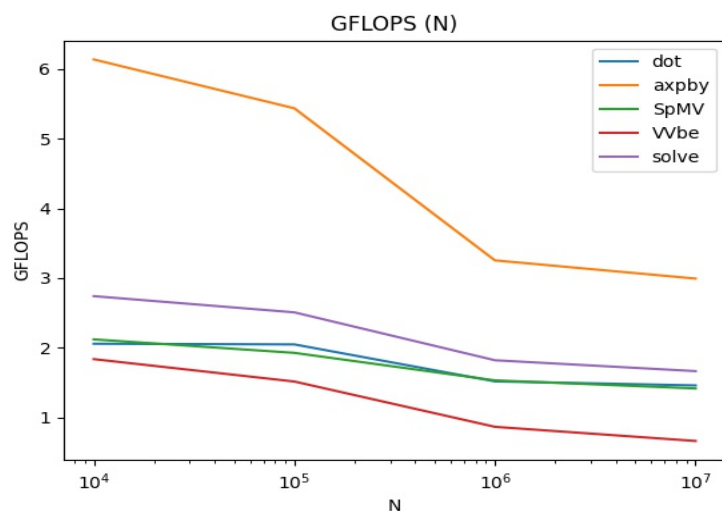
пиковая пропускная способность памяти: 41.8 GB/s

Параметры компиляции: -O3

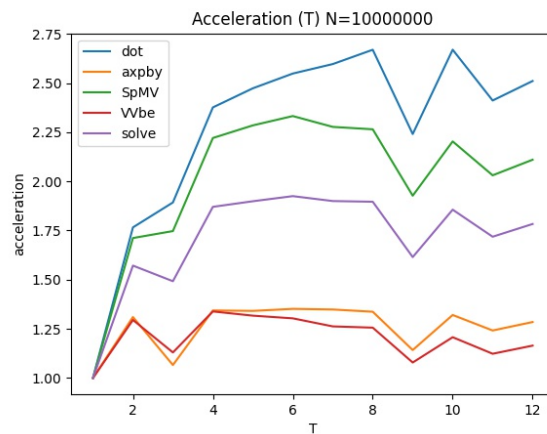
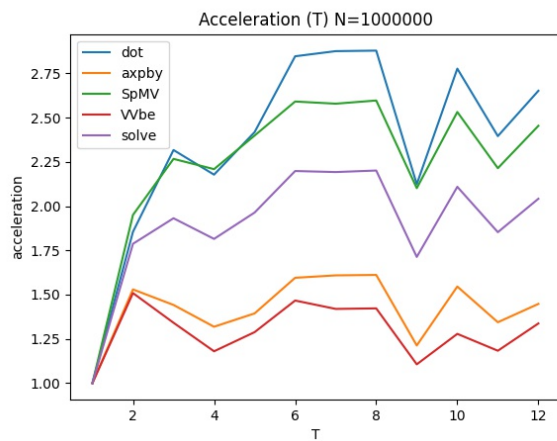
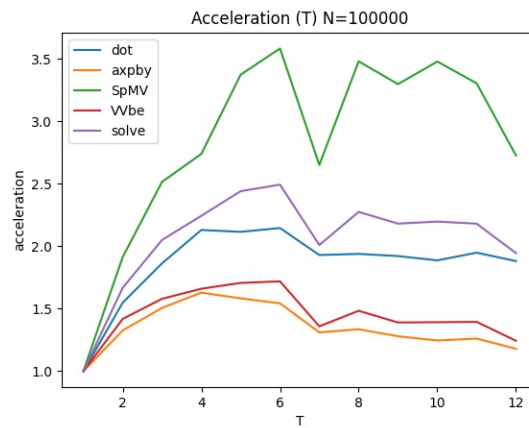
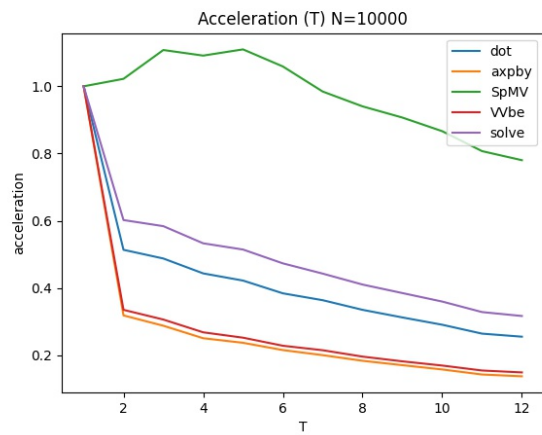
Результаты измерений производительности:

Последовательная реализация:

N	dot	axpby	SpMV	VVbe	solve
10000	2,060094	6,133742	2,122412	1,83925	2,741885
100000	2,050345	5,433809	1,929042	1,51834	2,509727
1000000	1,520486	3,254421	1,534913	0,869037	1,823153
10000000	1,462527	2,99372	1,419107	0,666467	1,667491



Параллельная реализация, ускорение:



dot	axpby	SpMV	VVbe	solve	T	N
1,00	1,00	1,00	1,00	1,00	1	10000000
1,77	1,31	1,71	1,30	1,57	2	10000000
1,89	1,07	1,75	1,13	1,49	3	10000000
2,38	1,34	2,22	1,34	1,87	4	10000000
2,47	1,34	2,28	1,32	1,90	5	10000000
2,55	1,35	2,33	1,30	1,93	6	10000000
2,60	1,35	2,28	1,26	1,90	7	10000000
2,67	1,34	2,27	1,26	1,90	8	10000000
2,24	1,14	1,93	1,08	1,61	9	10000000
2,67	1,32	2,20	1,21	1,86	10	10000000
2,41	1,24	2,03	1,12	1,72	11	10000000
2,51	1,28	2,11	1,16	1,78	12	10000000
1,00	1,00	1,00	1,00	1,00	1	1000000
1,85	1,53	1,95	1,51	1,79	2	1000000
2,32	1,44	2,27	1,34	1,93	3	1000000
2,18	1,32	2,21	1,18	1,81	4	1000000
2,42	1,39	2,40	1,29	1,96	5	1000000
2,85	1,60	2,59	1,47	2,20	6	1000000
2,88	1,61	2,58	1,42	2,19	7	1000000
2,88	1,61	2,60	1,42	2,20	8	1000000
2,12	1,21	2,10	1,11	1,71	9	1000000
2,78	1,55	2,53	1,28	2,11	10	1000000
2,40	1,34	2,21	1,18	1,85	11	1000000
2,65	1,45	2,45	1,34	2,04	12	1000000
1,00	1,00	1,00	1,00	1,00	1	100000
1,55	1,33	1,91	1,42	1,67	2	100000
1,86	1,51	2,51	1,58	2,05	3	100000
2,13	1,63	2,74	1,66	2,24	4	100000
2,11	1,58	3,37	1,71	2,44	5	100000
2,15	1,54	3,58	1,72	2,49	6	100000
1,93	1,31	2,65	1,36	2,01	7	100000
1,94	1,34	3,48	1,48	2,27	8	100000
1,92	1,28	3,30	1,39	2,18	9	100000
1,89	1,25	3,48	1,39	2,20	10	100000
1,95	1,26	3,30	1,39	2,18	11	100000
1,88	1,18	2,73	1,24	1,94	12	100000
1,00	1,00	1,00	1,00	1,00	1	10000
0,51	0,32	1,02	0,33	0,60	2	10000
0,49	0,29	1,11	0,31	0,58	3	10000
0,44	0,25	1,09	0,27	0,53	4	10000
0,42	0,24	1,11	0,25	0,51	5	10000
0,38	0,22	1,06	0,23	0,47	6	10000
0,36	0,20	0,98	0,21	0,44	7	10000
0,33	0,18	0,94	0,20	0,41	8	10000
0,31	0,17	0,91	0,18	0,38	9	10000
0,29	0,16	0,87	0,17	0,36	10	10000
0,26	0,14	0,81	0,15	0,33	11	10000
0,26	0,14	0,78	0,15	0,32	12	10000

Анализ полученных результатов

Максимальная достигаемая производительность для последовательной реализации для максимального N:

dot: 1.46 GFLOPS, 28-38% от пика / ядро

axpby: 2.99 GFLOPS, 58-78% от пика / ядро

SpMV: 1.42 GFLOPS, 28-37% от пика / ядро

VVbe: 0.66 GFLOPS, 13-17% от пика / ядро

Solve: 1.66 GFLOPS, 32-43% от пика / ядро

Максимальная достигаемая производительность для параллельной реализации для максимального N:

dot: 1.46 GFLOPS, 6-8% от пика / компьютер

axpby: 2.99 GFLOPS, 6-8% от пика / компьютер

SpMV: 1.42 GFLOPS, 5-7% от пика / компьютер

VVbe: 0.66 GFLOPS, 1-2% от пика / компьютер

Solve: 1.66 GFLOPS, 5-7% от пика / компьютер