

Московский Государственный Университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра суперкомпьютеров и квантовой информатики

Отчёт

Солвер BICGSTAB для СЛАУ

Работу выполнил:

Козлов Михаил Владимирович

523 группа

15.10.2018

Москва, 2018

1.	ОПИСАНИЕ ЗАДАНИЯ И ПРОГРАММНОЙ РЕАЛИЗАЦИИ	3
1.1.	Описание задания	3
1.2.	Описание программной реализации	3
2.	ИССЛЕДОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ	4
2.1.	Характеристики вычислительной системы	4
2.2.	Результаты измерений производительности.	5
3.	АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	9
3.1.	Процент от пика	9
3.2.	Процент от достижимой производительности	10
4.	ПРИЛОЖЕНИЕ	12

1. Описание задания и программной реализации

1.1. Описание задания

Реализовать решатель СЛАУ BICGSTAB для разреженных матриц в формате CSR. Применить библиотеку OpenMP для распараллеливания решателя, а также применить другие методы оптимизации.

1.2. Описание программной реализации

В рамках поставленной задачи был реализован решатель СЛАУ BICGSTAB. Для этого были реализованы следующие модели данных:

- класс `Matrix` – класс для хранения разреженной матрицы в формате CSR и выполнения операций с матрицей. В приватной области хранится структура матрицы: `double *A` – ненулевые элементы матрицы, `int *JA` – номера столбцов ненулевых элементов, `int *IA` – позиция начала данных для строки `i`, `int sizeIA` – размер вектора `IA`, `int sizeA` – размер векторов `A` и `JA`. Были реализованы различные конструкторы (`Matrix(int Nx, int Ny, int Nz)`, `Matrix(const Matrix &mat)`, `Matrix(const Matrix &mat, int k)`). Последний из них служит генератором диагональной матрицы, элементами которой являются обратные элементы диагонали входной матрицы. Также были реализованы функции вывода.
- класс `Vector` – класс для хранения вектора и выполнения операций над ним. В приватной области также хранится структура вектора: `double *A` – элементы вектора, `int size` – размер вектора. Были реализованы различные конструкторы (`Vector(int s)`, `Vector(const Vector &vec)`, `Vector(int s, double c)`, `Vector(int s, double *vec)`). Первый из них заполняет вектор синусами в цикле (`sin(i)`).

Были реализованы три базовые операции:

- `friend double dot(const Vector &vec1, const Vector &vec2)` – скалярное произведение двух векторов.
- `friend int axpby(Vector &vec1, const Vector &vec2, double a, double b)` – линейная комбинация двух векторов с коэффициентами `a`, `b`.
- `friend int SpMV(const Matrix &mat, const Vector &vec, Vector &res)` – умножение матрицы на вектор (`Ax`).

Функции решателей тестирования:

- `int solve(int N, Matrix &A, Vector &BB, double tol, int maxit, int debug)` – решатель СЛАУ $Ax = BB$ методом BICGSTAB. `N` – размерность матрицы, `tol` – невязка системы, `maxit` – максимальное количество итераций решателя, `debug` – флаг отладки.
- `int testFunc(int Nx, int Ny, int Nz, int N)` – функция, проводящая комплексное тестирование алгоритма и функций: тестирование на разных размерах и разном количестве потоков.

Были применены различные методы оптимизации, например, более оптимальный алгоритм для умножения разреженной матрицы на вектор, развертка циклов, многопоточность.

2. Исследования производительности

2.1. Характеристики вычислительной системы

Тестирование проводилось на кластере ВМК ИМВ Polus. Данная система содержит 5 вычислительных узлов, один из которых выполняет функцию фронтэнда. Основные характеристики узла можно посмотреть на сайте <http://hpc.cmc.msu.ru/polus>.

Пиковая производительность кластера 55.84 Tflop/s.

lscpu даёт следующий вывод:

```
Architecture:      ppc64le
Byte Order:        Little Endian
CPU(s):            160
On-line CPU(s) list: 0-159
Thread(s) per core: 8
Core(s) per socket: 10
Socket(s):         2
NUMA node(s):      2
Model:             1.0 (pvr 004c 0100)
Model name:        POWER8NVL (raw), altivec supported
CPU max MHz:       4023.0000
CPU min MHz:       2061.0000
Hypervisor vendor: (null)
Virtualization type: full
L1d cache:         64K
L1i cache:         32K
L2 cache:          512K
L3 cache:          8192K
NUMA node0 CPU(s): 0-79
NUMA node1 CPU(s): 80-159
```

Из информации из википедии (<https://ru.wikipedia.org/wiki/POWER8>) известно, что процессор POWER 8 имеет производительность 290 Gflop/s и 580 Gflop/s при обработке чисел двойной точности и одинарной точности соответственно. Максимальная пропускная способность памяти 230 GB/s.

Компиляция программы проводилась на фронтэнд узле командой `g++ -std=c++11 -g -O3 -fopenmp matrix.cpp -o matrix`, а постановка в очередь проводилась командой `mpisubmit.pl -t 20 matrix -- 100 100 100 0.0000001 1000 8 1`. В данном случае, все параметры кроме последнего не имеют смысла, так как флаг тестирования (последний параметр) равен

единице, и программа будет проводить тестирование. Вывод данных, полученных при таком запуске, будет показан в приложении в конце отчёта.

2.2. Результаты измерений производительности.

Тестирование проводилось следующим образом:

для каждого размера матрицы было произведено тестирование каждой базовой операции на различном количестве ядер (1, 2, 4, 8, 10, 16).

Размерности матрицы были следующими в формате (Nx, Ny, Nz) Nx*Ny*Nz:

- (10, 10, 10) 100,
- (10, 10, 100) 1000,
- (10, 100, 100) 10000,
- (100, 100, 100) 100000,
- (100, 100, 1000) 1000000.

Таблица времени выполнения базовых операций:

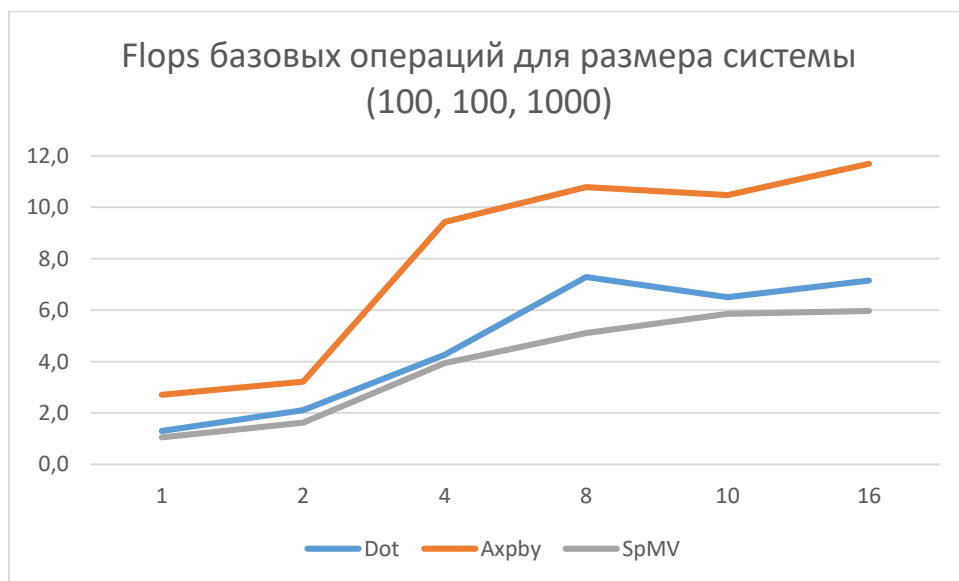
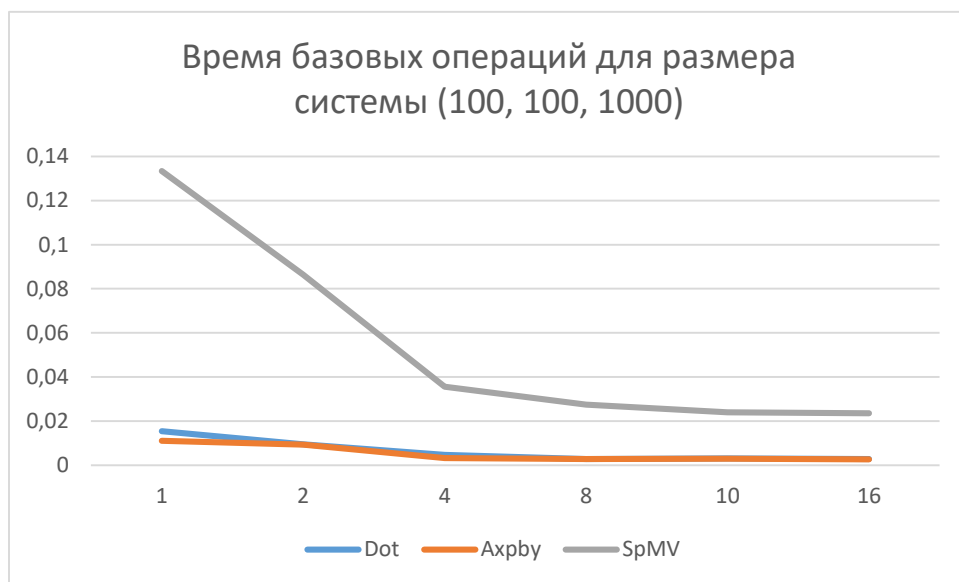
op	Nt	(10, 10, 10) time	(10, 10, 100) time	(10, 100, 100) time	(100, 100, 100) time	(100, 100, 1000) time
Dot	1	1.70525e-05	1.96993e-05	0.00016341	0.00163471	0.0153234
	2	1.44895e-05	3.74597e-05	0.000125093	0.00081804	0.00946328
	4	3.95626e-05	3.93521e-05	9.60529e-05	0.000476906	0.00469613
	8	4.88013e-05	5.34989e-05	9.51756e-05	0.000320572	0.00274481
	10	5.3281e-05	4.47407e-05	8.31168e-05	0.000334257	0.00307429
	16	0.00011529	0.000712642	0.000386907	0.000360645	0.00279637
Axpby	1	2.16626e-06	1.11051e-05	0.000103472	0.00163259	0.0110624
	2	2.14204e-06	1.83117e-05	7.04788e-05	0.000586953	0.00931855
	4	2.74181e-06	6.43544e-06	4.79091e-05	0.000271395	0.0031846
	8	4.81121e-06	6.56396e-06	4.22243e-05	0.000231985	0.00278291
	10	4.98071e-06	7.74302e-06	3.75044e-05	0.000323135	0.00286485
	16	3.97861e-06	1.35321e-05	3.38834e-05	0.00163259	0.00256709
SpMV	1	1.68793e-05	0.00013967	0.00131373	0.0130302	0.133388
	2	1.0388e-05	0.00011632	0.000677742	0.00675441	0.0863845
	4	1.16955e-05	5.22491e-05	0.000476131	0.00328081	0.0355777
	8	1.885e-05	5.0744e-05	0.00042399	0.00269281	0.0273881
	10	1.98167e-05	5.10532e-05	0.00037781	0.00315541	0.0239244
	16	2.39313e-05	5.60861e-05	0.000354204	0.00194852	0.0234475

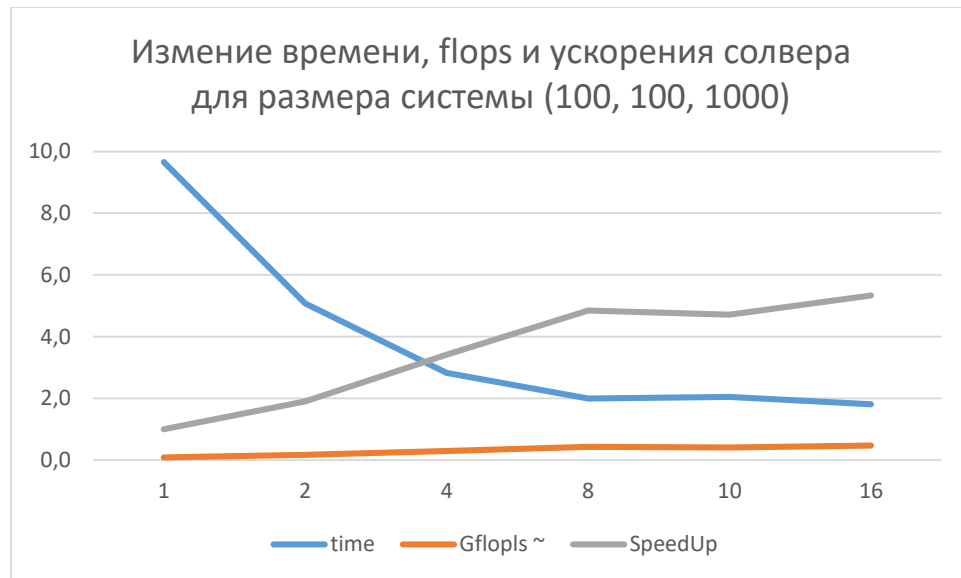
Таблица Gflops базовых операций:

op	Nt	(10, 10, 10) Gflops	(10, 10, 100) Gflops	(10, 100, 100) Gflops	(100, 100, 100) Gflops	(100, 100, 1000) Gflops
Dot	1	0.117285	1.01526	1.22392	1.22346	1.3052
	2	0.138031	0.533908	1.59881	2.44487	2.11343
	4	0.0505528	0.508232	2.08219	4.1937	4.25883
	8	0.0409825	0.37384	2.10138	6.23884	7.28647
	10	0.0375369	0.44702	2.40625	5.98342	6.50556
	16	0.0173475	0.0280646	0.51692	5.54561	7.15212
Ахрby	1	1.38488	2.70146	2.89934	1.83757	2.71189
	2	1.40053	1.6383	4.2566	5.11114	3.21938
	4	1.09417	4.66169	6.26186	11.054	9.42034
	8	0.623543	4.57041	7.10491	12.9319	10.7801
	10	0.602323	3.87446	7.99907	9.28403	10.4717
	16	0.754032	2.21695	2.89934	14.2317	11.6864
SpMV	1	0.829419	1.00236	1.06567	1.07443	1.04957
	2	1.34771	1.20357	2.06568	2.07272	1.62066
	4	1.19704	2.67947	2.94037	4.26724	3.93505
	8	0.742707	2.75894	3.30196	5.19903	5.11171
	10	0.706475	2.74224	3.70557	4.43682	5.85177
	16	0.585009	2.49616	3.95252	7.18494	5.97079

Таблица времени, Gflops и ускорения солвера для размера системы (100, 100, 1000)

	1	2	4	8	10	16
time	9.66218	5.07783	2.82449	1.99228	2.04838	1.81236
Gflops ~	0.0879719	0.167394	0.300939	0.426646	0.414962	0.469002
SpeedUp	1	1.90282	3.42085	4.8498	4.71699	5.33127





3. Анализ полученных результатов

3.1. Процент от пика

Для оценки процента производительность от пиковой, возьмём величину Gflops каждой операции и поделим на максимальную величину Gflops процессора. Из пункта 2.1 имеем максимальную производительность процессора POWER 8 для чисел двойной точности 290 Gflops. Если принять это за производительность 10 ядер по 8 потоков, то на один поток приходится 3.625 Gflops ($290 / 80$).

Таким образом, следующая таблица показывает процент от пиковой производительность для каждой операции для разных размеров и количества ядер.

op	Nt	(10, 10, 10) %	(10, 10, 100) %	(10, 100, 100) %	(100, 100, 100) %	(100, 100, 1000) %
Dot	1	3,235448	28,007172	33,763310	33,750621	36,005517
	2	1,903876	7,364248	22,052552	33,722345	29,150759
	4	0,348640	3,505048	14,359931	28,922069	29,371241
	8	0,141319	1,289103	7,246138	21,513241	25,125759
	10	0,103550	1,233159	6,637931	16,505986	17,946372
	16	0,029909	0,048387	0,891241	9,561397	12,331241
Axpby	1	38,203586	74,523034	79,981793	50,691586	74,810759
	2	19,317655	22,597241	58,711724	70,498483	44,405241
	4	7,546000	32,149586	43,185241	76,234483	64,967862
	8	2,150148	15,760034	24,499690	44,592759	37,172759
	10	1,661581	10,688166	22,066400	25,611117	28,887448
	16	1,300055	3,822328	4,998862	24,537414	20,148966
SpMV	1	22,880524	27,651310	29,397793	29,639448	28,953655
	2	18,589103	16,600966	28,492138	28,589241	22,353931
	4	8,255448	18,479103	20,278414	29,429241	27,138276
	8	2,561059	9,513586	11,386069	17,927690	17,626586
	10	1,948897	7,564800	10,222262	12,239503	16,142814
	16	1,008636	4,303724	6,814690	12,387828	10,294466

3.2. Процент от достижимой производительности

Для оценки процента от достижимой производительности с учётом величины пропускной способности памяти возьмём Gflops каждой операции и поделим на достижимую производительность с учётом пропускной способности памяти.

Эта величина равна $OpGflops / \min(tpp, bw * ai)$, где tpp – пиковая производительность, ai – арифметическая интенсивность алгоритма, bw – пропускная способность памяти. Из пункта 2.1 имеем 290 Gflops пиковой производительности, максимальная пропускная способность памяти 230 GB/s. Арифметическую интенсивность можно посчитать как отношение FLOP к числу байтов обращений в память. Значит, для базовых операций получим следующее количество байт обращений в память (double = 8 байт): 24N (8 * 3), 40N (8 * 5), 168N (8 * 3 * 7, оценка сверху) соответственно для dot, axpby, spmv. Следовательно, арифметическая интенсивность равна 1 / 12, 3 / 40, 1 / 12 соответственно для dot, axpby, spmv.

Таким образом, получаем следующую таблицу процента от достижимой производительности для разных размеров и разного количества ядер.

op	Nt	(10, 10, 10) %	(10, 10, 100) %	(10, 100, 100) %	(100, 100, 100) %	(100, 100, 1000) %
Dot	1	3,235448	28,007172	33,763310	33,750621	36,005517
	2	1,903876	7,364248	22,052552	33,722345	29,150759
	4	0,348640	3,505048	14,359931	28,922069	29,371241
	8	0,213822	1,950470	10,963722	32,550470	38,016365
	10	0,195845	2,332278	12,554348	31,217843	33,942052
	16	0,090509	0,146424	2,696974	28,933617	37,315409
Axpby	1	38,203586	74,523034	79,981793	50,691586	74,810759
	2	19,317655	22,597241	58,711724	70,498483	44,405241
	4	7,546000	32,149586	43,185241	76,234483	64,967862
	8	3,614742	26,495130	41,187884	74,967536	62,493333
	10	3,491728	22,460638	46,371420	53,820464	60,705507
	16	4,371200	12,851884	16,807768	82,502609	67,747246
SpMV	1	22,880524	27,651310	29,397793	29,639448	28,953655
	2	18,589103	16,600966	28,492138	28,589241	22,353931
	4	8,255448	18,479103	20,278414	29,429241	27,138276
	8	3,874993	14,394470	17,227617	27,125374	26,669791
	10	3,685957	14,307339	19,333409	23,148626	30,530974
	16	3,052221	13,023443	20,621843	37,486643	31,151948

4. Приложение

Вывод теста, запущенный на IBM Polus:

10 10 10 1000

> Threads = 1

> DOT = 499.509

> Time of DOT = 1.70525e-05

> GFLOPS = 0.117285

> SpeedUp = 1

> AXPBY L2 norm = 67.0491

> Time of AXPBY = 2.16626e-06

> GFLOPS = 1.38488

> SpeedUp = 1

> SpMV L2 norm = 48.7737

> Time of SpMV = 1.68793e-05

> GFLOPS = 0.829419

> SpeedUp = 1

> Threads = 2

> DOT = 499.509

> Time of DOT = 1.44895e-05

> GFLOPS = 0.138031

> SpeedUp = 1.17689

> AXPBY L2 norm = 67.0491

> Time of AXPBY = 2.14204e-06

> GFLOPS = 1.40053

> SpeedUp = 1.0113

> SpMV L2 norm = 48.7737
> Time of SpMV = 1.0388e-05
> GFLOPS = 1.34771
> SpeedUp = 1.62489

> Threads = 4

> DOT = 499.509
> Time of DOT = 3.95626e-05
> GFLOPS = 0.0505528
> SpeedUp = 0.431026

> AXPBY L2 norm = 67.0491
> Time of AXPBY = 2.74181e-06
> GFLOPS = 1.09417
> SpeedUp = 0.790082

> SpMV L2 norm = 48.7737
> Time of SpMV = 1.16955e-05
> GFLOPS = 1.19704
> SpeedUp = 1.44322

> Threads = 8

> DOT = 499.509
> Time of DOT = 4.88013e-05
> GFLOPS = 0.0409825
> SpeedUp = 0.349427

> AXPBY L2 norm = 67.0491
> Time of AXPBY = 4.81121e-06
> GFLOPS = 0.623543
> SpeedUp = 0.450252

> SpMV L2 norm = 48.7737
> Time of SpMV = 1.885e-05
> GFLOPS = 0.742707
> SpeedUp = 0.895455

> Threads = 10

> DOT = 499.509
> Time of DOT = 5.3281e-05
> GFLOPS = 0.0375369
> SpeedUp = 0.320049

> AXPBY L2 norm = 67.0491
> Time of AXPBY = 4.98071e-06
> GFLOPS = 0.602323
> SpeedUp = 0.434929

> SpMV L2 norm = 48.7737
> Time of SpMV = 1.98167e-05
> GFLOPS = 0.706475
> SpeedUp = 0.851772

> Threads = 16

> DOT = 499.509
> Time of DOT = 0.00011529
> GFLOPS = 0.0173475
> SpeedUp = 0.147909

> AXPBY L2 norm = 67.0491
> Time of AXPBY = 3.97861e-06
> GFLOPS = 0.754032

> SpeedUp = 0.544476

> SpMV L2 norm = 48.7737

> Time of SpMV = 2.39313e-05

> GFLOPS = 0.585009

> SpeedUp = 0.705324

10 10 100 10000

> Threads = 1

> DOT = 4999.86

> Time of DOT = 1.96993e-05

> GFLOPS = 1.01526

> SpeedUp = 1

> AXPBY L2 norm = 212.129

> Time of AXPBY = 1.11051e-05

> GFLOPS = 2.70146

> SpeedUp = 1

> SpMV L2 norm = 158.553

> Time of SpMV = 0.00013967

> GFLOPS = 1.00236

> SpeedUp = 1

> Threads = 2

> DOT = 4999.86

> Time of DOT = 3.74597e-05

> GFLOPS = 0.533908

> SpeedUp = 0.525881

> AXPBY L2 norm = 212.129

> Time of AXPBY = 1.83117e-05
> GFLOPS = 1.6383
> SpeedUp = 0.606449

> SpMV L2 norm = 158.553
> Time of SpMV = 0.00011632
> GFLOPS = 1.20357
> SpeedUp = 1.20074

> Threads = 4

> DOT = 4999.86
> Time of DOT = 3.93521e-05
> GFLOPS = 0.508232
> SpeedUp = 0.500592

> AXPBY L2 norm = 212.129
> Time of AXPBY = 6.43544e-06
> GFLOPS = 4.66169
> SpeedUp = 1.72562

> SpMV L2 norm = 158.553
> Time of SpMV = 5.22491e-05
> GFLOPS = 2.67947
> SpeedUp = 2.67317

> Threads = 8

> DOT = 4999.86
> Time of DOT = 5.34989e-05
> GFLOPS = 0.37384
> SpeedUp = 0.368219

> AXPBY L2 norm = 212.129
> Time of AXPBY = 6.56396e-06
> GFLOPS = 4.57041
> SpeedUp = 1.69183

> SpMV L2 norm = 158.553
> Time of SpMV = 5.0744e-05
> GFLOPS = 2.75894
> SpeedUp = 2.75245

> Threads = 10

> DOT = 4999.86
> Time of DOT = 4.47407e-05
> GFLOPS = 0.44702
> SpeedUp = 0.4403

> AXPBY L2 norm = 212.129
> Time of AXPBY = 7.74302e-06
> GFLOPS = 3.87446
> SpeedUp = 1.43421

> SpMV L2 norm = 158.553
> Time of SpMV = 5.10532e-05
> GFLOPS = 2.74224
> SpeedUp = 2.73578

> Threads = 16

> DOT = 4999.86
> Time of DOT = 0.000712642
> GFLOPS = 0.0280646
> SpeedUp = 0.0276427

> AXPBY L2 norm = 212.129
> Time of AXPBY = 1.35321e-05
> GFLOPS = 2.21695
> SpeedUp = 0.820647

> SpMV L2 norm = 158.553
> Time of SpMV = 5.60861e-05
> GFLOPS = 2.49616
> SpeedUp = 2.49029

10 100 100 100000
> Threads = 1

> DOT = 50000
> Time of DOT = 0.00016341
> GFLOPS = 1.22392
> SpeedUp = 1

> AXPBY L2 norm = 670.82
> Time of AXPBY = 0.000103472
> GFLOPS = 2.89934
> SpeedUp = 1

> SpMV L2 norm = 511.255
> Time of SpMV = 0.00131373
> GFLOPS = 1.06567
> SpeedUp = 1

> Threads = 2

> DOT = 50000
> Time of DOT = 0.000125093

> GFLOPS = 1.59881

> SpeedUp = 1.3063

> AXPBY L2 norm = 670.82

> Time of AXPBY = 7.04788e-05

> GFLOPS = 4.2566

> SpeedUp = 1.46813

> SpMV L2 norm = 511.255

> Time of SpMV = 0.000677742

> GFLOPS = 2.06568

> SpeedUp = 1.93839

> Threads = 4

> DOT = 50000

> Time of DOT = 9.60529e-05

> GFLOPS = 2.08219

> SpeedUp = 1.70125

> AXPBY L2 norm = 670.82

> Time of AXPBY = 4.79091e-05

> GFLOPS = 6.26186

> SpeedUp = 2.15975

> SpMV L2 norm = 511.255

> Time of SpMV = 0.000476131

> GFLOPS = 2.94037

> SpeedUp = 2.75917

> Threads = 8

> DOT = 50000

> Time of DOT = 9.51756e-05

> GFLOPS = 2.10138

> SpeedUp = 1.71693

> AXPBY L2 norm = 670.82

> Time of AXPBY = 4.22243e-05

> GFLOPS = 7.10491

> SpeedUp = 2.45053

> SpMV L2 norm = 511.255

> Time of SpMV = 0.00042399

> GFLOPS = 3.30196

> SpeedUp = 3.09849

> Threads = 10

> DOT = 50000

> Time of DOT = 8.31168e-05

> GFLOPS = 2.40625

> SpeedUp = 1.96603

> AXPBY L2 norm = 670.82

> Time of AXPBY = 3.75044e-05

> GFLOPS = 7.99907

> SpeedUp = 2.75893

> SpMV L2 norm = 511.255

> Time of SpMV = 0.00037781

> GFLOPS = 3.70557

> SpeedUp = 3.47722

> Threads = 16

> DOT = 50000
> Time of DOT = 0.000386907
> GFLOPS = 0.51692
> SpeedUp = 0.422349

> AXPBY L2 norm = 670.82
> Time of AXPBY = 3.38834e-05
> GFLOPS = 8.8539
> SpeedUp = 3.05376

> SpMV L2 norm = 511.255
> Time of SpMV = 0.000354204
> GFLOPS = 3.95252
> SpeedUp = 3.70895

100 100 100 1000000

> Threads = 1

> DOT = 500000
> Time of DOT = 0.00163471
> GFLOPS = 1.22346
> SpeedUp = 1

> AXPBY L2 norm = 2121.32
> Time of AXPBY = 0.00163259
> GFLOPS = 1.83757
> SpeedUp = 1

> SpMV L2 norm = 1712.5
> Time of SpMV = 0.0130302
> GFLOPS = 1.07443
> SpeedUp = 1

> Threads = 2

> DOT = 500000

> Time of DOT = 0.00081804

> GFLOPS = 2.44487

> SpeedUp = 1.99832

> AXPBY L2 norm = 2121.32

> Time of AXPBY = 0.000586953

> GFLOPS = 5.11114

> SpeedUp = 2.78146

> SpMV L2 norm = 1712.5

> Time of SpMV = 0.00675441

> GFLOPS = 2.07272

> SpeedUp = 1.92913

> Threads = 4

> DOT = 500000

> Time of DOT = 0.000476906

> GFLOPS = 4.1937

> SpeedUp = 3.42774

> AXPBY L2 norm = 2121.32

> Time of AXPBY = 0.000271395

> GFLOPS = 11.054

> SpeedUp = 6.01555

> SpMV L2 norm = 1712.5

> Time of SpMV = 0.00328081

> GFLOPS = 4.26724

> SpeedUp = 3.97163

> Threads = 8

> DOT = 500000

> Time of DOT = 0.000320572

> GFLOPS = 6.23884

> SpeedUp = 5.09935

> AXPBY L2 norm = 2121.32

> Time of AXPBY = 0.000231985

> GFLOPS = 12.9319

> SpeedUp = 7.03747

> SpMV L2 norm = 1712.5

> Time of SpMV = 0.00269281

> GFLOPS = 5.19903

> SpeedUp = 4.83888

> Threads = 10

> DOT = 500000

> Time of DOT = 0.000334257

> GFLOPS = 5.98342

> SpeedUp = 4.89057

> AXPBY L2 norm = 2121.32

> Time of AXPBY = 0.000323135

> GFLOPS = 9.28403

> SpeedUp = 5.05233

> SpMV L2 norm = 1712.5

> Time of SpMV = 0.00315541

> GFLOPS = 4.43682

> SpeedUp = 4.12947

> Threads = 16

> DOT = 500000

> Time of DOT = 0.000360645

> GFLOPS = 5.54561

> SpeedUp = 4.53273

> AXPBY L2 norm = 2121.32

> Time of AXPBY = 0.000210797

> GFLOPS = 14.2317

> SpeedUp = 7.74482

> SpMV L2 norm = 1712.5

> Time of SpMV = 0.00194852

> GFLOPS = 7.18494

> SpeedUp = 6.68721

100 100 1000 10000000

> Threads = 1

> DOT = 5e+06

> Time of DOT = 0.0153234

> GFLOPS = 1.3052

> SpeedUp = 1

> AXPBY L2 norm = 6708.2

> Time of AXPBY = 0.0110624

> GFLOPS = 2.71189

> SpeedUp = 1

> SpMV L2 norm = 5431.56

> Time of SpMV = 0.133388
> GFLOPS = 1.04957
> SpeedUp = 1

> Threads = 2

> DOT = 5e+06
> Time of DOT = 0.00946328
> GFLOPS = 2.11343
> SpeedUp = 1.61924

> AXPBY L2 norm = 6708.2
> Time of AXPBY = 0.00931855
> GFLOPS = 3.21938
> SpeedUp = 1.18714

> SpMV L2 norm = 5431.56
> Time of SpMV = 0.0863845
> GFLOPS = 1.62066
> SpeedUp = 1.54412

> Threads = 4

> DOT = 5e+06
> Time of DOT = 0.00469613
> GFLOPS = 4.25883
> SpeedUp = 3.26298

> AXPBY L2 norm = 6708.2
> Time of AXPBY = 0.0031846
> GFLOPS = 9.42034
> SpeedUp = 3.47372

> SpMV L2 norm = 5431.56
> Time of SpMV = 0.0355777
> GFLOPS = 3.93505
> SpeedUp = 3.74921

> Threads = 8

> DOT = 5e+06
> Time of DOT = 0.00274481
> GFLOPS = 7.28647
> SpeedUp = 5.58266

> AXPBY L2 norm = 6708.2
> Time of AXPBY = 0.00278291
> GFLOPS = 10.7801
> SpeedUp = 3.97512

> SpMV L2 norm = 5431.56
> Time of SpMV = 0.0273881
> GFLOPS = 5.11171
> SpeedUp = 4.8703

> Threads = 10

> DOT = 5e+06
> Time of DOT = 0.00307429
> GFLOPS = 6.50556
> SpeedUp = 4.98436

> AXPBY L2 norm = 6708.2
> Time of AXPBY = 0.00286485
> GFLOPS = 10.4717
> SpeedUp = 3.86142

> SpMV L2 norm = 5431.56
> Time of SpMV = 0.0239244
> GFLOPS = 5.85177
> SpeedUp = 5.57541

> Threads = 16

> DOT = 5e+06
> Time of DOT = 0.00279637
> GFLOPS = 7.15212
> SpeedUp = 5.47973

> AXPBY L2 norm = 6708.2
> Time of AXPBY = 0.00256709
> GFLOPS = 11.6864
> SpeedUp = 4.30932

> SpMV L2 norm = 5431.56
> Time of SpMV = 0.0234475
> GFLOPS = 5.97079
> SpeedUp = 5.6888

> Solver test...

> Number of threads = 1
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753
It = 6; res = 0.447256; tol = 0.000200019

It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08
It = 14; res = 2.48026e-05; tol = 1.10921e-08
It = 15; res = 7.72044e-06; tol = 3.45269e-09
It = 16; res = 2.70016e-06; tol = 1.20755e-09
It = 17; res = 9.65432e-07; tol = 4.31754e-10
> Final discrepancy = 9.65432e-07
> Numder of iters = 17
> Final time of computation = 9.66218
> GFLOPS = 0.0879719
> SpeedUp = 1

> Number of threads = 2
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753
It = 6; res = 0.447256; tol = 0.000200019
It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08
It = 14; res = 2.48026e-05; tol = 1.10921e-08

It = 15; res = 7.72044e-06; tol = 3.45269e-09
It = 16; res = 2.70016e-06; tol = 1.20755e-09
It = 17; res = 9.65432e-07; tol = 4.31754e-10
> Final discrepancy = 9.65432e-07
> Numder of iters = 17
> Final time of computation = 5.07783
> GFLOPS = 0.167394
> SpeedUp = 1.90282

> Number of threads = 4
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753
It = 6; res = 0.447256; tol = 0.000200019
It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08
It = 14; res = 2.48026e-05; tol = 1.10921e-08
It = 15; res = 7.72044e-06; tol = 3.45269e-09
It = 16; res = 2.70016e-06; tol = 1.20755e-09
It = 17; res = 9.65432e-07; tol = 4.31754e-10
> Final discrepancy = 9.65432e-07
> Numder of iters = 17
> Final time of computation = 2.82449
> GFLOPS = 0.300939
> SpeedUp = 3.42085

> Number of threads = 8
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753
It = 6; res = 0.447256; tol = 0.000200019
It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08
It = 14; res = 2.48026e-05; tol = 1.10921e-08
It = 15; res = 7.72044e-06; tol = 3.45269e-09
It = 16; res = 2.70016e-06; tol = 1.20755e-09
It = 17; res = 9.65432e-07; tol = 4.31754e-10
> Final discrepancy = 9.65432e-07
> Numder of iters = 17
> Final time of computation = 1.99228
> GFLOPS = 0.426646
> SpeedUp = 4.8498

> Number of threads = 10
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753

It = 6; res = 0.447256; tol = 0.000200019
It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08
It = 14; res = 2.48026e-05; tol = 1.10921e-08
It = 15; res = 7.72044e-06; tol = 3.45269e-09
It = 16; res = 2.70016e-06; tol = 1.20755e-09
It = 17; res = 9.65432e-07; tol = 4.31754e-10
> Final discrepancy = 9.65432e-07
> Number of iters = 17
> Final time of computation = 2.04838
> GFLOPS = 0.414962
> SpeedUp = 4.71699

> Number of threads = 16
It = 0; res = 2236.07; tol = 1
It = 1; res = 837.045; tol = 0.374338
It = 2; res = 90.2838; tol = 0.0403762
It = 3; res = 19.6953; tol = 0.00880802
It = 4; res = 5.82113; tol = 0.00260329
It = 5; res = 1.54234; tol = 0.000689753
It = 6; res = 0.447256; tol = 0.000200019
It = 7; res = 0.106756; tol = 4.77425e-05
It = 8; res = 0.0284587; tol = 1.27271e-05
It = 9; res = 0.0075798; tol = 3.38979e-06
It = 10; res = 0.00235927; tol = 1.0551e-06
It = 11; res = 0.000887506; tol = 3.96905e-07
It = 12; res = 0.000252407; tol = 1.1288e-07
It = 13; res = 7.53324e-05; tol = 3.36897e-08

It = 14; res = 2.48026e-05; tol = 1.10921e-08

It = 15; res = 7.72044e-06; tol = 3.45269e-09

It = 16; res = 2.70016e-06; tol = 1.20755e-09

It = 17; res = 9.65432e-07; tol = 4.31754e-10

> Final discrepancy = 9.65432e-07

> Number of iters = 17

> Final time of computation = 1.81236

> GFLOPS = 0.469002

> SpeedUp = 5.33127