



Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Параллельные вычисления

Многопоточная реализация солвера BiCGSTAB для
СЛАУ с разреженной матрицей, заданной в формате
CSR.

Выполнил:

студент 524 группы

Бикбулатов Тимур Русланович

14.10.2018

Москва, 2018

Содержание отчёта

1	Описание задания и программной реализации	3
1.1	Краткое описание задания	3
1.2	Краткое описание программной реализации - как организованы данные, какие функции реализованы (название, аргументы, на- значение)	3
2	Исследование производительности	6
2.1	Характеристики вычислительной системы:	6
2.2	Результаты измерений производительности	6
3	Анализ полученных результатов	10
4	Приложение1	11

1 Описание задания и программной реализации

1.1 Краткое описание задания

Многопоточная реализация солвера BiCGSTAB для СЛАУ с разреженной матрицей в формате CSR. Измерение производительности при параллельной и последовательной реализациях программы.

1.2 Краткое описание программной реализации - как организованы данные, какие функции реализованы (название, аргументы, назначение)

Запуск под OS X: `clang++ -Xpreprocessor -fopenmp pv.cpp -o pv.o -lomp`

Запуск в общем виде: `g++ pv.cpp -o pv -fopenmp`

Аргументы с командной строки: `Nx, Ny, Nz, tol, maxit, qa;`

Пример запуска: `./pv.o 100 100 100 0.000001 20 1`

Используемые функции:

1) `dot(X, Y, N)` – скалярное произведение векторов

X - вектор размера N

Y - вектор размера N

N - размер векторов

2) `axrby(X, Y, a, b, N)` – линейная комбинация векторов $X=a*X+b*Y$

X - вектор размера N

Y - вектор размера N

a - коэффициент для координат вектора X

b - коэффициент для координат вектора Y

N - размер векторов

3) `SpMV(Aval, AI, AJ, X, Y, N)` - матрично-векторное произв. $Y=AX$

Aval, AI, AJ - CSR представление матрицы размера в виде трех массивов

X - вектор размера N

Y - результирующий вектор размера N

N - размер векторов

SpMV(DD, X, Y, N) - матрично-векторное произведение $Y = AX$

DD - матрица размера N

X - вектор размера N

Y - результирующий вектор размера N

N - размер векторов

5) solver(N, Aval, AI, AJ, BB, tol, maxit) - алгоритм солвера

N - размер векторов

Aval, AI, AJ - CSR представление матрицы в виде трех массивов

BB - вектор правой части размера N

tol - критерий сходимости

maxit - максимальное число итераций

6) makediag(DD, Aval, AI, AJ, N) - создание диагональной матрицы

DD - матрица с диагональными элементами вида $dd=1/aa$

Aval, AI, AJ - CSR представление матрицы размера в виде трех массивов

N - размер векторов

7) repeatmatr(X, Y, N) – копирование вектора $X = Y$

X - вектор размера N

Y - вектор размера N

N - размер векторов

8) test_dot(N); test_axpby(N); test_spmv(N); test_function()

- функции для тестирования работы операций dot, axpby, spmv

- запуск функций тестирования, приведённых выше

N - размер векторов

2 Исследование производительности

2.1 Характеристики вычислительной системы:

MacBook Pro (Retina, 13-inch, Late 2013)

Процессор 2,6 GHz Intel Core i5

Графика Intel Iris 1536 МБ

Количество процессоров: 1

Общее количество ядер: 2

Общее количество логических ядер : 4

Кэш 2-го уровня (в каждом ядре): 256 КБ

Кэш L3: 3 МБ

Память: 8 ГБ

2.2 Результаты измерений производительности

Для $N = 1000$:

(DOT) testing sequential ops :

dot time= 0.002s GFLOPS= 0.39

(DOT) testing parallel ops for ntr= 2 :

dot time= 0.001s GFLOPS= 0.59 Speedup= 1.51X

(DOT) testing parallel ops for ntr= 4 :

dot time= 0.001s GFLOPS= 0.71 Speedup= 1.82X

(AXPBY) testing sequential ops :

axpby time= 0.002s GFLOPS= 0.65

(AXPBY) testing parallel ops for ntr= 2 :

axpby time= 0.001s GFLOPS= 0.81 Speedup= 1.25X

(AXPBY) testing parallel ops for ntr= 4 :

axpby time= 0.002s GFLOPS= 0.52 Speedup= 0.80X

(SPMV) testing sequential ops :

spmv time= 0.002s GFLOPS= 0.36

(SPMV) testing parallel ops for ntr= 2 :

spmv time= 0.002s GFLOPS= 0.48 Speedup= 1.33X

(SPMV) testing parallel ops for ntr= 4 :

spmv time= 0.002s GFLOPS= 0.50 Speedup= 1.39X

Для N = 10000 :

(DOT) testing sequential ops :

dot time= 0.022s GFLOPS= 0.36

(DOT) testing parallel ops for ntr= 2 :

dot time= 0.014s GFLOPS= 0.58 Speedup= 1.60X

(DOT) testing parallel ops for ntr= 4 :

dot time= 0.021s GFLOPS= 0.39 Speedup= 1.07X

(AXPBY) testing sequential ops :

axpby time= 0.021s GFLOPS= 0.57

(AXPBY) testing parallel ops for ntr= 2 :

axpby time= 0.012s GFLOPS= 0.99 Speedup= 1.72X

(AXPBY) testing parallel ops for ntr= 4 :

axpby time= 0.007s GFLOPS= 1.83 Speedup= 3.20X

(SPMV) testing sequential ops :

spmv time=0.021s GFLOPS= 0.38

(SPMV) testing parallel ops for ntr= 2 :

spmv time=0.014s GFLOPS= 0.56 Speedup= 1.50X

(SPMV) testing parallel ops for ntr= 4 :

spmv time=0.008s GFLOPS= 0.98 Speedup= 2.64X

Для N = 100000 :

(DOT) testing sequential ops :

dot time= 0.199s GFLOPS= 0.40

(DOT) testing parallel ops for ntr= 2 :

dot time= 0.087s GFLOPS= 0.92 Speedup= 2.30X

(DOT) testing parallel ops for ntr= 4 :

dot time= 0.088s GFLOPS= 0.91 Speedup= 2.26X

(AXPBY) testing sequential ops :

axpby time= 0.156s GFLOPS= 0.77

(AXPBY) testing parallel ops for ntr= 2 :

axpby time= 0.077s GFLOPS= 1.57 Speedup= 2.04X

(AXPBY) testing parallel ops for ntr= 4 :

axpby time= 0.060s GFLOPS= 2.00 Speedup= 2.60X

(SPMV) testing sequential ops :

spmv time= 0.204s GFLOPS= 0.39

(SPMV) testing parallel ops for ntr= 2 :

spmv time= 0.087s GFLOPS= 0.92 Speedup= 2.34X

(SPMV) testing parallel ops for ntr= 4 :

spmv time= 0.080s GFLOPS= 1.00 Speedup= 2.55X

Для $N = 1000000$:

(DOT) testing sequential ops :

dot time= 1.489s GFLOPS= 0.54

(DOT) testing parallel ops for ntr= 2 :

dot time= 0.738s GFLOPS= 1.08 Speedup= 2.02X

(DOT) testing parallel ops for ntr= 4 :

dot time= 0.674s GFLOPS= 1.19 Speedup= 2.21X

(AXPBY) testing sequential ops :

axpby time= 1.383s GFLOPS= 0.87

(AXPBY) testing parallel ops for ntr= 2 :

axpby time= 0.707s GFLOPS= 1.70 Speedup= 1.96X

(AXPBY) testing parallel ops for ntr= 4 :

axpby time= 0.631s GFLOPS= 1.90 Speedup= 2.19X

(SPMV) testing sequential ops :

spmv time= 1.655s GFLOPS= 0.48

(SPMV) testing parallel ops for ntr= 2 :

spmv time= 0.840s GFLOPS= 0.95 Speedup= 1.97X

(SPMV) testing parallel ops for ntr= 4 :

spmv time= 0.801s GFLOPS= 1.00 Speedup= 2.07X

3 Анализ полученных результатов

dot:

Пиковая (достижимая) производительность: 25.32 GFLOPS

Максимальная в тесте производительность: 1.19 GFLOPS

Процент: 4.6998%

axrbv:

Пиковая (достижимая) производительность: 39.22 GFLOPS

Максимальная в тесте производительность: 2.00 GFLOPS

Процент: 5.0994%

srmv:

Пиковая (достижимая) производительность: 25.37 GFLOPS

Максимальная в тесте производительность: 1.00 GFLOPS

Процент: 3.9416%

4 Приложение1

Исходный текст программы в отдельном с++ файле с названием rv.cpp