

УДК 519.688

ПАРАЛЛЕЛЬНОЕ УМНОЖЕНИЕ МАТРИЦЫ И МАТРИЧНОЙ СТЕПЕНИ НА ВЕКТОР

© Г.И. Малашонок, В.А. Лукашин

Ключевые слова: умножение разреженной матрицы на вектор; система компьютерной математики Mathpar.

Обсуждается проблема создания эффективного параллельного алгоритма умножения матрицы и матричной степени на вектор. Приводятся результаты экспериментов, проведенных на кластере МВС-100К Межведомственного суперкомпьютерного центра РАН.

1 Введение

Решение систем линейных уравнений, вычисление характеристического полинома, обращение матрицы — эти и другие задачи линейной алгебры могут быть решены с использованием пространства Крылова.

Пространство Крылова, порожденное вектором v и матрицей A , — это линейная оболочка, образованная m линейно независимыми векторами

$$K(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\},$$

при условии, что

$$A^m v \in K(A, v).$$

Если матрица A имеет размер $n \times n$, а размерность пространства Крылова существенно меньше, чем n , то можно получить алгоритмы, которые требуют меньше операций, чем классические алгоритмы. Обычно это наблюдается у разреженных матриц, у которых степень минимального полинома значительно меньше, чем размер матрицы.

Разреженные матрицы очень больших размеров появляются во многих прикладных задачах и это стимулирует исследование алгоритмов построения пространства Крылова для таких матриц. При этом центральной задачей является вычисление произведения матричной степени на вектор. Последовательный алгоритм вычисления такого произведения сводится просто к задаче эффективного вычисления скалярного произведения векторов, в то время как задача эффективного параллельного умножения остается открытой и требует специального изучения.

Мы предполагаем провести эксперименты с различными подходами к построению параллельного алгоритма.

В этой работе мы исследуем один алгоритм и приводим результаты экспериментов, в которых исследовалось общее время вычисления и эффективность вычислений.

2 Параллельный алгоритм

Пусть N — число процессоров в кластере, α — число ненулевых элементов в матрице A размера $n \times n$, $\alpha \leq n^2$. Для вычисления $A^m \cdot b$ будем пользоваться схемой

$$A(\dots A(A(A \cdot b))). \quad (1)$$

Алгоритм:

Матрица A разбивается на N блочных строк $A_i, i = 1, \dots, N$ таким образом, что каждый блок A_i имел примерно $\approx \alpha/N$ ненулевых элементов. Обозначим s_i номера строк, образующих блок A_i . Процессору с номером $i, i = 1, \dots, N$, пересылаются блок A_i , номера строк этого блока s_i и вектор b .

Цикл для $k = \overline{1, m-1}$ ($v_0 = b$):

- Процессор i вычисляет произведение $r_{i,k} = A_i * v_{k-1}$, составляющее i -ю часть вектора результата, затем посылает его всем N процессорам.
- Процессор i получает от каждого процессора j часть $r_{j,k}$ вектора v_k и составляет вектор $v_k = (r_{1,k}, r_{2,k}, \dots, r_{N,k})$. Он сохраняет вектор v_k у себя и использует его для вычисления следующей степени.

На последнем шаге цикла все части $r_{i,m}$ передаются только корневому процессору, который формирует вектор v_m .

Для пересылки между процессорами применяется упаковка объектов в байт-массивы и используются следующие методы библиотеки MPI:

- 1) Isend-Recv — для пересылки матриц A_i ;
- 2) Scatter — для пересылки массивов s_i ;
- 3) Bcast — для пересылки векторов b и $r_{i,k}$.

3 Результаты экспериментов

Эксперименты проводились на кластере МВС-100К Межведомственного суперкомпьютерного центра РАН. Объем оперативной памяти каждого процессора составлял 8 Гб. Во всех экспериментах участвовали случайные разреженные матрицы плотностью 10 %. Коэффициенты матриц и вектора b выбирались из простого поля Z/pZ . Число разрядов у p не превышало 31. Вычислялись все матричные степени $A^m \cdot b = v_m, m = 1, 2, \dots, 1000$. Изменялся размер n матрицы и число N процессоров в группе. Время работы параллельной программы сравнивалось со временем работы последовательной программы на одном процессоре. Результаты экспериментов приведены на рис. 1 и в табл. 1.

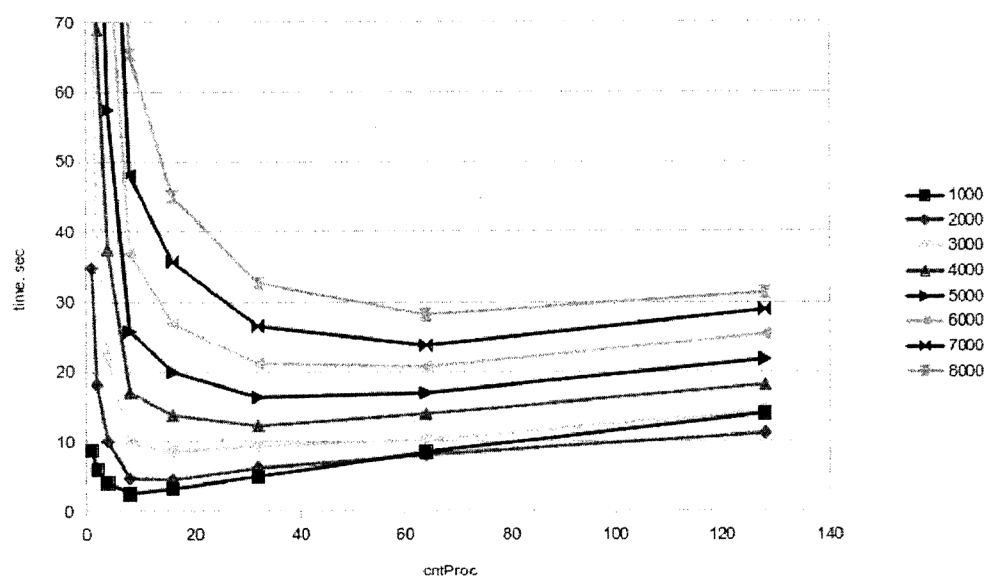


Рис. 1. Зависимость времени вычислений от количества процессоров $N = 1, 2, 4, \dots, 128$ для матриц размера $n \times n$, $n = 1000, 2000, \dots, 8000$

Таблица 1

Время, которое было затрачено на вычисление $m = 1000$ компонент пространства для матрицы размера $n \times n$ в группе из N процессоров (с)

$N \setminus n$	1000	2000	3000	4000	5000	6000	7000	8000
1	8,74	34,76	77,95	142,98	214,04	317,31	425,69	529,48
2	5,99	18,16	39,61	69,1	108,76	155,94	209,83	283,41
4	4,1	10,03	21,95	37,53	57,45	83,44	112,88	146,6
8	2,56	4,69	9,68	17,05	25,82	36,84	47,99	65,38
16	3,23	4,51	8,5	13,75	20	26,83	35,66	44,9
32	5	6,23	9,31	12,24	16,38	21,19	26,56	32,68
64	8,49	8,04	10,23	13,96	16,99	20,73	23,78	28,16
128	14,03	11,21	14,4	18,16	21,83	25,38	28,89	31,36

Если время вычислений в группе, содержащей N процессоров, равно t_N , а время таких же вычислений на одном процессоре t_1 , то коэффициентом ускорения, полученного на N процессорах будем считать величину

$$a_N = \frac{t_1}{t_N}. \quad (2)$$

В табл. 2 приведены коэффициенты ускорения, составленные для табл. 1.

Коэффициенты ускорения вычислений

N \ n	1000	2000	3000	4000	5000	6000	7000	8000
2	1,46	1,91	1,97	2,07	1,97	2,03	2,03	1,87
4	2,13	3,47	3,55	3,81	3,73	3,80	3,77	3,61
8	3,41	7,41	8,06	8,39	8,29	8,61	8,87	8,10
16	2,71	7,71	9,17	10,40	10,70	11,83	11,94	11,79
32	1,75	5,58	8,37	11,69	13,07	14,98	16,03	16,20
64	1,03	4,32	7,62	10,25	12,59	15,31	17,90	18,80
128	0,62	3,10	5,41	7,88	9,80	12,50	14,73	16,88

Из табл. 2 видно, что наибольший коэффициент ускорения для матриц размера 1000 достигается на 8-ми процессорах, для матриц размера 2000-3000 достигается на 16-ти процессорах, для матриц размера 4000-5000 достигается на 32-х процессорах, для матриц размера 6000-8000 достигается на 64-х процессорах. На 128 процессорах ни одну из этих задач не удается решить с большим ускорением.

Возможно, что не всегда требуется добиваться ускорения вычислений любой ценой, не взирая на эффективность. Представляет интерес и коэффициент эффективности:

$$f_i = \frac{a_i}{p_i} * 100 \%, \quad (3)$$

где a_i — коэффициент ускорения, p_i — количество процессоров. Коэффициент эффективности равен 100 %, если время вычислений на N процессорах в N раз меньше, чем на одном процессоре.

Коэффициенты эффективности представлены в табл. 3.

Таблица 3

Коэффициенты эффективности вычислений (%)

N \ n	1000	2000	3000	4000	5000	6000	7000	8000
2	73,00	95,71	98,41	103,46	98,40	101,74	101,44	93,41
4	53,27	86,68	88,79	95,24	93,14	95,07	94,28	90,30
8	42,66	92,62	100,72	104,84	103,60	107,66	110,88	101,24
16	16,91	48,19	57,31	65,00	66,90	73,92	74,62	73,70
32	5,46	17,44	26,17	36,52	40,84	46,80	50,08	50,63
64	1,61	6,76	11,91	16,01	19,68	23,92	27,97	29,38
128	0,49	2,42	4,23	6,15	7,66	9,77	11,51	13,19

Из табл. 3 видно, что наибольшая эффективность вычислений достигается на восьми процессорах для матриц размера 2000-8000, а для матриц размера 10000 эффективнее всего считать на одном процессоре.

4 Заключение

Приведено исследование одной параллельной схемы вычисления произведения матрицы и матричной степени на вектор на кластере МВС-100К Межведомственного суперкомпью-

терного центра РАН. Размеры матриц менялись в диапазоне 1000-8000. Плотность матрицы составляла 10 %. В дальнейшем предполагается исследовать другие параллельные схемы вычислений и провести эксперименты с матрицами других размеров.

БЛАГОДАРНОСТИ: Работа выполнена при поддержке гранта РФФИ № 12-07-00755-а.

Поступила в редакцию 20 декабря 2012 г.

Malaschonok G.I., Lukashin V.A. PARALLEL MULTIPLICATION ALGORITHMS FOR MATRIX AND MATRIX DEGREE BY THE VECTOR.

Problems of development of efficient parallel multiplication algorithms for matrix and matrix degree by the vector are discussed. The results of experiments carried out on a cluster MVS100K of JSCC RAS are presented.

Key words: sparse matrix multiplication by a vector, parallel matrix multiplication, computer mathematics Mathpar.