

УДК 004.4(043.2)

Гамаюн В.П., д.т.н.

СОРТИРОВКА КВАЗИУПОРЯДОЧЕННЫХ МАССИВОВ**Національний авіаційний університет**

gamayun@nau.edu.ua

Введение

Обработка массивов, данных в компьютерной среде является актуальной задачей. Актуальность такого типа обработки возросла с развитием новых систем счисления и организации вычислительного процесса [1,2]. Так при разрядно-логарифмической системе счисления, где каждый значащий (ненулевой) разряд кодируется как логарифм от веса этого ненулевого разряда $N_i = \log_2 a_i p^i$,

При такой системе счисления операнд представим, как

$$A \rightarrow A^N = \{N_n \dots N_2, N_1\}$$

Например: 100011012 = 7.3.2.0РЛ
Увеличение точности обработки при РЛ кодировании на несколько порядков имеет особенность – появляется новый операционный базис на основе сортировки массивов [1,2,3,4]. Время реализации сортировки определяет и быстродействие таких систем в целом.

Сортировка по блокам и технология с перекрытием разрядов

Общая модель сортировки массивов с разрядно-логарифмическим представлением должна отражать особые признаки таких массивов – частичную упорядоченность составляющих операндов.

Части массивов могут упорядочены по контенту или формироваться по некоторым правилам упорядоченности. Пусть массив А из Т элементов упорядочен по убыванию (возрастанию), а массив В аналогично состоит из S элементов, упорядоченных по убыванию (возрастанию).

Задано правило формирования массива С как произведения А*В и определена задача упорядочения полученного массива. Основное предложение (решение) – любой массив после некоторой

трансформации становится квазиупорядоченным или полностью упорядоченным.

Такая трансформация может быть К – точечным фильтром, после которого имеем К блоков с неупорядоченными элементами – выполняется разделение по полосе фильтра. Каждый полученный блок подвергается К фильтрации, в результате которой получаем К*К новых неупорядоченных блоков в совокупности представляющих исходный массив. Причем блоки упорядочены относительно друг друга.

$$C = A * B = CK = CK-1 * K-1 = CK-2 * K-2 * K-2 \dots \dots$$

Процедуру такой фильтрации нужно продолжать до оптимальных размеров К блоков. Предельным значением блоков может быть «количество» шагов по определению параметров фильтра блока трансформации, что связано с анализом значений чисел и шага фильтрации – другими словами параметры гребенки фильтра. Если количество шагов по определению параметра соизмеримо с количеством трансформируемых элементов (первое должно быть больше второго), то процесс трансформации должен быть остановлен.

Относительно количества разрядов при сравнении фильтрации: при полной разрядной сетке – базовый подход, а если по нескольким (старшим) разрядам – то это метод с зацеплением.

При первом подходе сравнение выполняется медленнее, но нет необходимости применения общей процедуры упорядочивания. При втором методе более быстрая реализация, но затраты на общее сцепление и упорядочение. Выбор за реализацией – по оценке работы различных моделей на практических массивах. Общая модель ориентирована на различные разрядности и формы представления данных.

Пусть задан массив

$C = 15 \ 2 \ 18 \ 17 \ 6 \ 5 \ 4 \ 3 \ 1 \ 11 \ 8 \ 7 \ 12 \ 9 \ 13 \ 10 \ 14 \ 19 \ 20 \ 16$

Выбираем начальную систему K фильтров: фильтры для следующих значений

$K_1 = 1 - 5 \quad K_2 = 6 - 10 \quad K_3 = 11 - 15 \quad K_4 = 16 - 20$

После фильтрации всего массива получаем блоки со следующими элементами

$2 \ 5 \ 4 \ 3 \ 1 \quad 6 \ 8 \ 7 \ 9 \ 10 \ 15 \ 11 \ 12 \ 13 \ 14 \ 18 \ 17 \ 19 \ 20 \ 16$

После упорядочения внутри блоков формируем распределение всего массива

$12345 \quad 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20,$

что является искомым результатом.

Рассмотри другой алгоритм сортировки.

Алгоритм формирования массива при векторном умножении или умножении через матрицу частичных произведений определяет умножение каждый со всеми остальными элементами, начиная со старших значений. Если операнды заданы как $A_1 A_2 A_3 A_4 \dots A_i \dots A_T$ и $B_1 B_2 B_3 B_4 \dots B_j \dots B_K$ то получаемая матрица

$A_1 B_1$	$A_1 B_2$	$A_1 B_3$	\dots	$A_1 B_{K-1}$	$A_1 B_K$
$A_2 B_1$	$A_2 B_2$	$A_2 B_3$	\dots	$A_2 B_{K-1}$	$A_2 B_K$
$A_3 B_1$	$A_3 B_2$	$A_3 B_3$	\dots	$A_3 B_{K-1}$	$A_3 B_K$
\dots	\dots	\dots	\dots	\dots	\dots
$A_T B_1$	$A_T B_2$	\dots	\dots	$A_T B_{K-1}$	$A_T B_K$
\dots	$A_T B_2$	\dots	\dots	\dots	\dots

может быть квазиупорядочена по алгоритму ранжирования сумм индексов, т.е. записываются значения у которых суммы индексов $i+j$ меньше больше других.

Такой алгоритм базируется на упорядоченности исходных операндов – старшие значения имеют меньшие индексы. Таким образом представленная матрица преобразуется в квазиупорядоченный массив

$A_1 B_1 \ A_1 B_2 \quad A_2 B_1 \quad A_1 B_3 \quad A_2 B_2$
 $A_3 B_1 \quad \dots \quad A_T B_{K-1} \quad A_T B_K$

Например, для РЛ операндов (сомножителей) $A = 5 \ 4 \ 2 \ 1$ и $B = 6 \ 4 \ 1 \ 0$ получаем произведение (матрицу частичных произведений)

11	10	8	7		
	9	8	6	5	
		6	5	3	2
			5	4	2
					1,

которая преобразуется в массив, упорядоченный по сумме индексов

$11 \ - \ 10 \ 9 \ - \ 8 \ 8 \ 6 \ - \ 7 \ 6 \ 6 \ 5 \ - \ 5 \ 3 \ 4$
 $- \ 2 \ 2 \ - \ 1.$

После упорядочения внутри блоков получаем

$11 \ - \ 10 \ 9 \ - \ 8 \ 8 \ 6 \ - \ 7 \ 6 \ 6 \ 5 \ - \ 5 \ 4 \ 3$
 $2 \ 2 \ - \ 1,$ что является искомым результатом.

Далее для получения окончательного результата применим технологию сортировки по одному элементу.

Этапы (шаги) процедуры заключаются в следующем:

- формируется массив из квазиупорядоченных элементов;

- формируются блоки после фильтрации или по другому алгоритму (например, алгоритму суммы индексов);

- внутри блоков выполняется упорядочивание;

- выполняется формирование новых блоков по правилу: крайние элементы блоков заменяются на крайние элементы смежных блоков, если между ними выполняется соотношение ранжирования;

- внутри блоков выполняется упорядочивание;

- последние два пункта повторяются до получения результата сортировки в целом.

Массив 11 - 10 9 - 8 8 6 - 7 6 6 5 - 5 4 3- 2 2 - 1 преобразуется в следующий массив 11 - 10 9 - 8 8 7- 6 6 6 5 - 5 4 3- 2 2 - 1 что является конечным результатом.

Выводы

Моделирование показывает, что сортировка разрядно логарифмических массивов, данных по предложенной технологии увеличивает быстродействие (по количеству шагов для параллельной структуры) от единиц до нескольких порядков.

Развитие методов сортировки с перекрытием заключается в увеличении количества разрядов при перекрытии. Реализация моделей обработки с перекрытием по два разряда показывает пропорциональное увеличение скорости сортировки.

Гамаюн В.П.

СОРТИРОВКА КВАЗИУПОРЯДОЧЕННЫХ МАССИВОВ

В организации вычислительного процесса при задачно-алгоритмическом методе проектирования основной этап при шивке набора алгоритмов в обобщенную блок-схему заключается в выборе часто используемого алгоритма, который затем структурно реализуется в операционном автомате. В классическом построении компьютерных средств таким общим алгоритмом является сложение, другие алгоритмы реализуются при поддержке этого базового. Математическое обеспечение совместно со операционным (структурным) автоматом составляет операционный базис.

В статье рассмотрено предложение построения нового операционного базиса компьютерных средств для высокоточных по обработке компьютерных средств и систем. Высокоточная обработка реализуется через разрядно-логарифмическое представление. Базовой операцией при котором является сортировка массивов, элементами которых являются короткие цифровые коды. Сортировка выполняется для упорядочения массивов. При выполнении операций при разрядно логарифмическом кодировании массивы некоторым образом (не полностью) упорядочены. Поэтому постановка задачи выбрана именно при таких условиях. Главное решение задачи предлагается в несколько этапов. Упорядочение массива по блокам с граничными значениями, упорядочение внутри блоков и стыковка блоков между собой с «перекрытием» граничных элементов блоков. Затем повторение сортировки внутри блоков и последующие «перекрытие». Сортировка заканчивается при отсутствии неупорядоченных элементов. Моделирование определяет сокращение шагов сортировки на один-два порядка по сравнению с сортировкой методом «пузырька» на гипотетической параллельной структуре. Улучшение базовой операции определяет положительное быстродействие нового операционного базиса в целом.

Ключевые слова: сортировка псевдо упорядоченных массивов, разрядно-логарифмическое кодирование, сортировка по блокам.

Литература

1. Гамаюн В.П. Организация обработки в многооперандных вычислительных структурах. / Гамаюн В.П.// – Киев, 1996. – 20 с. – (Препр./ НАН Украины. Ин-т кибернетики; 96 - 3).
2. Гамаюн В.П. Разрядно-логарифмическая арифметика. Методы и алгоритмы. / Гамаюн В.П.// – К.:Книжное изд. НАУ,2007. –272с.
3. Гамаюн В.П. Моделирование багаторазрядных компьютерных систем: Навч.посібник. / Гамаюн В.П.// – К.:Книжкове вид-во НАУ,2007, – 112с.
1. Лорин Г. Сортировка и системы сортировки. / Лорин Г.// – М.: Наука, 1983. – 384 с.

Gamayun V.P.

SORTING THE QUASI ORDERED ARRAY

In organizing a computational process with a task-algorithmic design method, the main step in stitching a set of algorithms into a generalized block scheme is to select a frequently used algorithm, which is then structurally implemented in an operational machine. In the classical construction of computer tools, such a general algorithm is addition, other algorithms are implemented with the support of this basic one. The software together with the operational (structural) machine makes up the operational basis.

The article considers the proposal to build a new operational basis of computer tools for high-precision processing of computer tools and systems. High-precision processing is implemented through the bit-logarithmic representation. The basic operation in which is the sorting of arrays whose elements are short digital codes. Sorting is done to arrange arrays. When performing operations in bitwise logarithmic coding, arrays are in some way (not completely) ordered. Therefore, the statement of the problem is chosen precisely under such conditions. The main solution to the problem is proposed in several stages. Array ordering by blocks with boundary values, ordering inside blocks and docking blocks with each other with "overlapping" boundary elements of blocks. Then repeat the sorting within the blocks and the subsequent "overlap". Sorting ends when there are no unordered items. Modeling determines the reduction of sorting steps by one or two orders of magnitude compared to sorting by the "bubble" method on a hypothetical parallel structure. Improving the basic operation determines the positive performance of the new operational basis as a whole.

Keywords: *sorting pseudo-ordered arrays, bit-logarithmic coding, sorting by blocks.*