

ЭФФЕКТИВНОСТЬ ПОСРЕЗОВОЙ ОБРАБОТКИ ВЕКТОРНЫХ МАССИВОВ ДАННЫХ

*Винницкий национальный технический университет, Винница, Украина

**Киевский государственный экономико-технологический университет транспорта, Киев, Украина

***Анотація.** Наведено базові положення критерію ефективності для алгоритмів паралельної обробки масивів даних. Виконано порівняльний аналіз паралельних алгоритмів багатооперандного підсумовування. Показано переваги позрізової обробки векторних масивів даних.*

***Ключові слова:** критерій ефективності, коефіцієнт узгодженості, позрізова обробка, векторний масив даних.*

***Аннотация.** Приведены базовые положения критерия эффективности для алгоритмов параллельной обработки массивов данных. Выполнен сравнительный анализ параллельных алгоритмов многооперандного суммирования. Показаны преимущества посрезовой обработки векторных массивов данных.*

***Ключевые слова:** критерий эффективности, коэффициент согласованности, посрезовая обработка, векторный массив данных.*

***Abstract.** The basic provisions of the efficiency criterion for the parallel processing of data algorithms are presented. A comparative analysis of parallel algorithms of multi-operand summation is conducted. The advantages of the multi-sliced processing of vector data set are shown.*

***Keywords:** the criterion of effectiveness, consistency factor, sliced processing, a vector data set.*

1. Введение

Среди методов повышения производительности вычислительных средств, в частности, в составе интеллектуальных систем, например, в интеллектуальных робототехнических системах [1], среди первоочередных задач необходимо выделить создание адаптивных и параллельных методов обработки больших массивов данных, а также их адекватное отображение на параллельные архитектуры вычислительных средств.

Для сравнительного анализа известных и разрабатываемых параллельных методов обработки массивов данных обязательным является выбор показателей эффективности, которые учитывают характерный для конкретной задачи параллелизм, а также адекватность предложенной для нее структуры [2, 3].

2. Актуальность

Одной из наиболее часто используемых операций в процедурах обработки и анализа сигналов и изображений, а также при распознавании образов [4–7], является операция многооперандного суммирования или свертки элементов вектора [5], которая, в свою очередь, является наиболее сложной при аппаратном распараллеливании [6–8]. Широко известны два способа ускорения (распараллеливания) многооперандного суммирования: древовидный алгоритм или алгоритм логарифмического суммирования [5] и алгоритм рекурсивного суммирования или алгоритм рекурсивного удвоения [6]. Основные характеристики этих и последовательного алгоритма приведены в табл. 1.

Дальнейшее продвижение в процессе распараллеливания многооперандного суммирования возможно только при использовании распределённой обработки, в данном случае векторного массива данных. Одним из таких подходов к распараллеливанию является ме-

тод обработки векторного массива по разностным срезам [8–11], основные показатели которого также приведены в табл. 1.

При сравнительном анализе трех приведенных способов параллельной обработки векторных массивов в качестве критерия эффективности как наиболее приемлемый выбран коэффициент согласованности φ_A , основные сведения о котором изложены в работе [12]. Этот критерий эффективности позволяет учесть не только алгоритмический уровень параллелизма, но и соответствующие аппаратные затраты.

Таблица 1. Основные характеристики алгоритмов многооперандного суммирования

Параметры	Алгоритмы реализации оператора многооперандного суммирования			
	Последовательный алгоритм	Алгоритм логарифмического суммирования	Алгоритм рекурсивного суммирования	Алгоритм с посрезовой обработкой
Количество тактов, t	$n - 1$	$\log_2 n$	$\log_2 n$	$n \cdot N_y$
Количество процессоров (блоков), p	1	$n/2 \leq p \leq n - 1$	$n/2 \leq p \leq n/2$	n
Общее количество операций алгоритма, ω	$L(n - 1)$	$L(n - 1)$	$L(n/2) \log_2 n$	$3n \leq \omega \leq 3n^2$
Время выполнения алгоритма, T_p^n	$\tau(n - 1)$	$\tau \log_2 n$	$\tau \log_2 n$	$\tau n \leq T_p^n \leq \tau n^2$
Количество операций, L	1	1	1	4
Тип операций	Суммирование	Суммирование	Суммирование	Суммирование, вычитание, умножение, нахождение минимального элемента
τ – время выполнения одного такта алгоритма, N_y – количество циклов, n – размерность векторного массива				

Таким образом, целью данной работы является анализ эффективности способов параллельной обработки векторного массива данных на примере многооперандного суммирования его элементов.

3. Базовые положения критерия эффективности

Для коэффициента согласованности φ_A характерно следующее соотношение [12]:

$$\varphi_A = \frac{0,5}{C} \sum_{i=1}^S c_i \cdot \left(\frac{T_A}{\sum_{l=1}^L \tau_{il} \cdot p_{il}} + \frac{\sum_{l=1}^L \tau_{il} \cdot p_{il}}{T_A} \right), \quad (1)$$

где C – цена структуры по формуле

$$C = \sum_{i=1}^S c_i \cdot N_i, \quad (2)$$

c_i – цена i -го блока, N_i – количество i -х блоков структуры, S – количество блоков в структуре, ν_i – коэффициент использования i -го блока, который можно представить следующим образом:

$$\nu_i = \frac{1}{T_A} \cdot \sum_{l=1}^L \tau_{il} \cdot p_{il}, \quad (3)$$

где T_A – время выполнения алгоритма, τ_{il} – время реализации на i -м блоке операции l -го типа, p_{il} – количество операций l -го типа, которые реализуются за время T_A на i -м блоке, L – количество типов операций алгоритма.

В дальнейшем для упрощения расчетов примем, что цена каждого i -го блока одинакова, то есть

$$c_i = c = const. \quad (4)$$

Кроме того, для алгоритмов логарифмического и рекурсивного суммирования при условии $L=1$ (табл. 1), учитывая, что время τ_{il} реализации операции суммирования на каждом i -м блоке одинаково и равно в среднем времени τ выполнения одного такта алгоритма суммирования, имеем

$$\tau_{il} = \tau = const. \quad (5)$$

Итак, можно выполнить следующие преобразования для коэффициента ν_i (3) использования i -го блока:

$$\nu_i = \frac{1}{T_A} \cdot \sum_{l=1}^L \tau_{il} \cdot p_{il} = \frac{\tau}{T_A} p_i = \frac{p_i}{t}, \quad (6)$$

где t – количество тактов, $T_A = t \cdot \tau$, p_i – количество операций, которые реализуются i -м блоком за время T_A , причем

$$p_i = \frac{\omega}{p}, \quad (7)$$

где ω – общее количество операций алгоритма, p – количество параллельных процессоров, в данном случае, параллельно функционирующих блоков (табл. 1).

В результате имеем

$$\nu_i = \frac{\omega}{t \cdot p}, \quad (8)$$

что приведет соотношение (1) для коэффициента согласованности φ_A с учетом выражения (2) для величины C к виду

$$\varphi_A = 0,5 \frac{1}{c \cdot S} \sum_{i=1}^S c' \left(\frac{t \cdot p}{\omega} + \frac{\omega}{t \cdot p} \right), \quad (9)$$

а с учетом количества S одинаковых блоков имеем

$$\varphi_A = 0,5 \left(\frac{t \cdot p}{\omega} + \frac{\omega}{t \cdot p} \right). \quad (10)$$

4. Сравнительный анализ параллельных алгоритмов многооперандного суммирования

Составляющие коэффициента согласованности φ_A (10) для трех рассматриваемых алгоритмов приведены в табл. 2.

Для алгоритмов логарифмического и рекурсивного суммирования рассчитаны по два варианта коэффициента согласованности соответственно: а) с учетом максимального количества параллельных блоков S (процессоров p) – $\varphi1(n)$, $\varphi3(n)$; б) с учетом их минимального количества – $\varphi2(n)$, $\varphi4(n)$ (табл. 2). Графики зависимостей $\varphi1(n)$ – $\varphi4(n)$ приведены на рис. 1 для размерностей массива чисел $n = 2, \dots, 64$.

Первоначальный упрощенный вариант расчета коэффициента согласованности φ_A для алгоритма с посрезовой обработкой приведен в работе [13]. В настоящей работе более детально учтены особенности посрезовой обработки векторного массива данных, а именно: количество параллельно функционирующих блоков S одинаково, но в этом случае варьируется количество F реализованных одновременно алгоритмов за время T_A . Поэтому в дальнейшем рассматриваются два варианта: а) с учетом совмещенных алгоритмов ($F = 4$), а именно, алгоритма параллельного суммирования, сортировки, пороговой обработки и формирования матрицы \mathbf{F} и вектора \mathbf{q} для дальнейшего восстановления элементов исходного массива – $\varphi5(n)$ [10]; б) без учета совмещенных алгоритмов ($F = 1$) – $\varphi6(n)$ (табл. 2).

Кроме того, примем, что общее количество операций для алгоритма с посрезовой обработкой (табл. 2) составляет

$$\omega = 3n \cdot N_u, \quad (11)$$

где N_u – количество циклов обработки ($1 \leq N_u \leq n$), а также учтем, что среди четырех операций ($L = 4$), выполняемых в каждом цикле, операцией бинарного умножения можно пренебречь, поскольку она реализуется в процессе коммутации соответствующих данных, и время ее реализации t_{com} значительно меньше, чем время других операций, то есть

$$t_{com} \leq t_{sm} \approx t_{sub} \approx t_{min}, \quad (12)$$

где t_{sm} , t_{sub} , t_{min} – время суммирования, вычитания и определения минимального числа (min) соответственно [10]. Тогда время выполнения алгоритма составляет

$$T_A = 2\tau \cdot n \cdot N_u, \quad (13)$$

где учтен последовательный характер выполнения операций min и суммирования частичных сумм для n операндов, но количество операций, выполняемых в каждом цикле, равняется двум, поскольку, уже начиная с первого цикла, можно совместить последовательное выполнение двух операций: min и суммирование частичных сумм [10].

Таблица 2. Составляющие коэффициента согласованности для параллельных алгоритмов

Составляющие коэффициента согласованности	Параллельные алгоритмы суммирования					
	Алгоритм логарифмического суммирования		Алгоритм рекурсивного суммирования		Алгоритм с посрезовой обработкой	
	$\varphi 1(n)$	$\varphi 2(n)$	$\varphi 3(n)$	$\varphi 4(n)$	$\varphi 5(n)$	$\varphi 6(n)$
Количество реализованных одновременно алгоритмов F	1	1	1	1	4	1
Количество параллельно функционирующих блоков S или процессоров p	$n-1$	$\frac{n}{2}$	$\frac{n}{2} \cdot \log_2 n$	$\frac{n}{2}$	n	n
Время выполнения алгоритма T_A	$\tau \cdot \log_2 n$	$\tau \cdot \log_2 n$	$\tau \cdot \log_2 n$	$\tau \cdot \log_2 n$	$\tau \cdot \log_2 n$	$2\tau n \cdot N_y$ $1 \leq N_y \leq n$
Общее количество операций ω	$n-1$	$n-1$	$\frac{n}{2} \cdot \log_2 n$	$\frac{n}{2} \cdot \log_2 n$	$3n \cdot N_y$	$3n \cdot N_y$
Коэффициент использования i -го блока v_i	$\frac{1}{\log_2 n}$	$\frac{2(n-1)}{n \cdot \log_2 n}$	$\frac{1}{\log_2 n}$	1	$\frac{6}{n}$	$\frac{3}{2n}$
Коэффициент согласованности φ_A	$0,5 \left(\log_2 n + \frac{1}{\log_2 n} \right)$	$0,5 \left(\frac{n \cdot \log_2 n}{2(n-1)} + \frac{2(n-1)}{n \cdot \log_2 n} \right)$	$0,5 \left(\log_2 n + \frac{1}{\log_2 n} \right)$	1	$0,5 \left(\frac{n}{6} + \frac{6}{n} \right)$	$0,5 \left(\frac{2n}{3} + \frac{3}{2n} \right)$

С учетом количества F реализованных одновременно алгоритмов с посрезовой обработкой, коэффициент v_i (3) использования i -го блока в общем случае можно представить так:

$$v_i = \frac{1}{T_A / F} \sum_{i=1}^L \tau_{ii} p_{ii} \cdot \quad (14)$$

А после преобразований этот коэффициент ν_i с учётом выражения (8) примет вид

$$\nu_i = \frac{F \omega}{t \cdot p}, \quad (15)$$

где, по данным табл. 2, $\omega = L \cdot n N_u$, $e = 2n N_u$, $p = n$.

В результате коэффициент согласованности φ_A для данного алгоритма имеет вид

$$\varphi_A = 0,5 \left(\frac{t \cdot p}{F \omega} + \frac{F \omega}{t \cdot p} \right), \quad (16)$$

а с учетом данных табл. 1 и 2 имеем

$$\varphi_A = 0,5 \left(\frac{2n}{F \cdot L} + \frac{F \cdot L}{2n} \right). \quad (17)$$

Графики зависимостей $\varphi5(n)$ и $\varphi6(n)$ также приведены на рис. 1.

5. Анализ результатов моделирования

Полученные математические модели для трех алгоритмов параллельной обработки векторных массивов данных в виде соответствующих коэффициентов согласованности $\varphi1(n)$ – $\varphi6(n)$ (табл. 2) позволяют показать их графическую интерпретацию для любого значения размерности n векторного массива чисел (рис. 1).

Кроме того, выражение (17) позволяет определить оптимальную разрядность n_{opt} массива чисел, которая обеспечивает оптимальное значение коэффициента согласованности $\varphi_A = 1$ [12] при определенных величинах F и L , а именно:

$$n_{opt} = \frac{F \cdot L}{2}, \quad (18)$$

что при $F = 4$ и $L = 3$ (табл. 2) составляет $n_{opt} = 6$ для алгоритма с посрезовой обработкой. Полученный результат совпадает с графиком для зависимости $\varphi5(n)$ ($F = 4$) на рис. 1.

Из анализа полученных графиков (рис. 1) вытекает, что наиболее оптимальным по коэффициенту согласованности φ_A является алгоритм рекурсивного суммирования с минимальным количеством процессоров, для которого зависимость $\varphi4(n)$ совпадает с оптимальным значением $\varphi_A = 1$ для всех значений n . Следующим по оптимальности является алгоритм логарифмического суммирования с минимальным количеством процессоров, для которого зависимость $\varphi2(n)$ медленно возрастает, не превышая значения $\varphi_A = 4$ на отрезке n от 2 до 64. Для обоих приведенных алгоритмов с максимальным количеством процессоров зависимости $\varphi1(n)$ и $\varphi3(n)$ совпадают, возрастая на отрезке n от 2 до 64 без превышения значения $\varphi_A = 6$.

Для алгоритма с посрезовой обработкой без учета количества реализованных одновременно алгоритмов ($F = 1$) зависимость $\varphi6(n)$ резко возрастает уже на отрезке n от 2 до 20, что свидетельствует о неэффективном использовании в данном случае значительного количества блоков (процессоров).

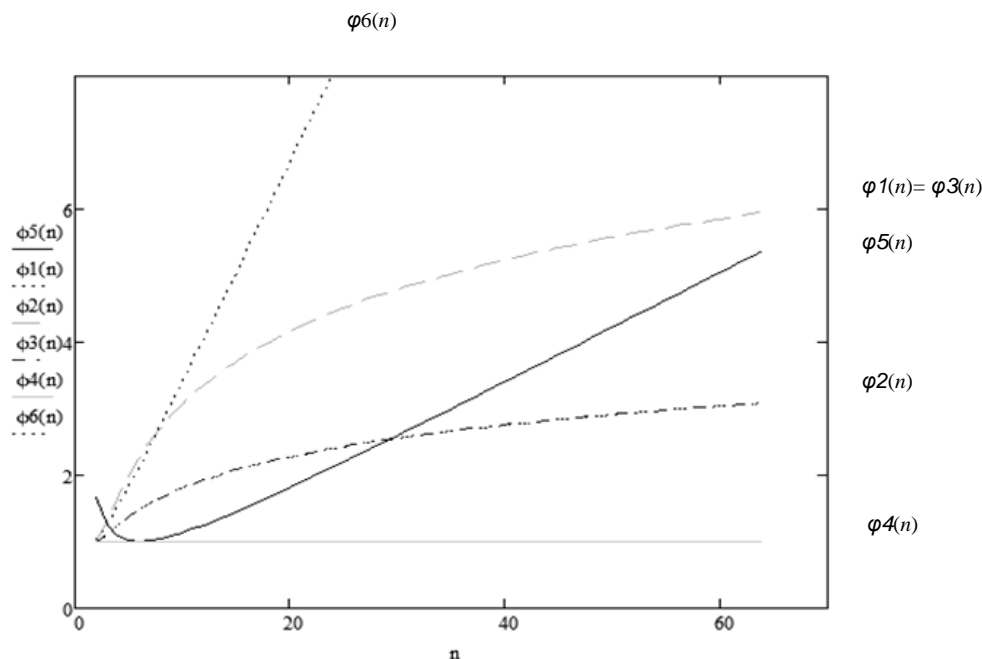


Рис. 1. Графики зависимости коэффициента согласованности от размерности векторного массива для параллельных алгоритмов суммирования

Другая картина наблюдается для этого алгоритма с учетом количества реализованных одновременно алгоритмов ($F = 4$), где зависимость $\varphi5(n)$ резко падает к значению $\varphi_A = 1$ на отрезке n от 2 до 6, что подтверждает расчеты $n_{\text{опт}}$ из выражения (18), а потом начинает возрастать, но при этом на отрезке n от 6 до 28 $\varphi5(n)$ не превышает значений $\varphi2(n)$, а на отрезке n от 6 до 60 – значений $\varphi1(n) = \varphi3(n)$.

6. Выводы

Итак, можно считать, что предложенный многофункциональный метод посрезовой обработки может быть реализован на соответствующей многопроцессорной структуре с приемлемой эффективностью, аналогичной для известных параллельных алгоритмов суммирования в пределах размерности n входного массива данных от 4 до 28. При этом надо заметить, что для алгоритмов логарифмического и рекурсивного суммирования с минимальным количеством процессоров p не учитываются аппаратные и временные затраты на перекоммутацию данных между процессорами, что в результате может увеличить обоснованно приемлемую размерность n входного массива (больше 28) для суммирования с посрезовой обработкой.

СПИСОК ЛИТЕРАТУРЫ

1. Архитектура гибридной интеллектуальной системы управления мобильного робота / А.В. Гаврилов, В.В. Губарев, К.Х. Джо [и др.] // Вестник Новосибирского ГТУ. – 2004. – № 2. – С. 3 – 13.
2. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – СПб.: БХВ – Петербург, 2004. – 608 с.
3. Гергель В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем: учеб. пособ. / В.П. Гергель, Р.Г. Стронгин. – Нижний Новгород: Изд-во ННГУ им. Н. Лобачевского, 2003. – 184 с.

4. Прэтт У. Цифровая обработка изображений: в 2-х кн. / Прэтт У.; пер. с англ. – М.: Мир, 1982. – Кн. 1. – 312 с.; Кн. 2. – 480 с.
5. Системы параллельной обработки / Под ред. Д. Ивенса. – М.: Мир, 1985. – 416 с.
6. Хохлюк В.И. Параллельные алгоритмы целочисленной оптимизации / Хохлюк В.И. – М.: Радио и связь, 1987. – 224 с.
7. Гамаюн В.П. Организация макрооператорной обработки в многооперандных вычислительных структурах / В.П. Гамаюн // Управляющие системы и машины. – 1995. – № 4/5. – С. 10 – 14.
8. Мартинюк Т.Б. Рекурсивні алгоритми багатооперандної обробки інформації / Мартинюк Т.Б. – Вінниця: УНІВЕРСУМ – Вінниця, 2000. – 216 с.
9. Мартинюк Т.Б. Методи та засоби паралельних перетворень векторних масивів даних / Т.Б. Мартинюк, В.В. Хом'юк. – Вінниця: УНІВЕРСУМ – Вінниця, 2005. – 203 с.
10. Мартинюк Т.Б. Особенности математической модели дискретного SM-преобразования / Т.Б. Мартинюк, В.В. Хом'юк // Математичні машини і системи. – 2010. – № 4. – С. 145 – 155.
11. Мартинюк Т.Б. Модель порогового нейрона на основе параллельной обработки по разностным срезам / Т.Б. Мартинюк // Кибернетика и системный анализ. – 2005. – № 4. – С. 78 – 89.
12. Трухний В.Д. Структурно-информационные оценки качества цифровых вычислительных машин / В.Д. Трухний // Экономика и математические методы. – 1969. – Вып. 1. – С. 79 – 89.
13. Мартинюк Т.Б. Оцінювання ефективності алгоритмів мультиобробки масивів даних / Т.Б. Мартинюк, В.В. Хом'юк // Вісник Вінницького політехнічного інституту. – 2005. – № 5. – С. 76 – 82.

Стаття надійшла до редакції 10.03.2017