

МУЛЬТИОБРАБОТКА МАССИВОВ ДАННЫХ ПО РАЗНОСТНЫМ СРЕЗАМ

Ключевые слова: мультиобработка данных, пороговая обработка векторного массива данных, системы алгоритмических алгебр, разностный срез.

ВВЕДЕНИЕ

Вопросы структурного программирования алгоритмов сортировки достаточно подробно освещены в ряде работ [1–5]. Данные публикации подтвердили универсальность аппарата структурированных схем программ, его широкие возможности в плане анализа, классификации и создания новых алгоритмов с оптимальными характеристиками [2–4, 6]. Дальнейшее развитие алгебраической алгоритмики способствует разработке и исследованию семейств алгебр, адекватных основным парадигмам программирования и предназначенных для формализованного описания различных дискретных моделей — логических, теоретико-автоматных, грамматических, алгоритмических [7–9]. В известных работах системы алгоритмических алгебр (САА) В.М. Глушкова и их модификации в основном использованы для формализованного представления задач символьной мультиобработки, а именно различных методов сортировки — наиболее наглядных и разнообразных примеров мультиобработки.

Одной из распространенных задач мультиобработки массива данных является параллельное суммирование элементов векторного массива [10], которое также называют операцией свертки вектора [11] или оператором группового суммирования [12]. Один из вариантов реализации этой операции с использованием разностных срезов рассмотрен в [13]. Особенность такого подхода — дальнейшее развитие принципа многооперандности обработки векторного массива чисел, а также особое сочетание последовательного и параллельного выполнения базисных операций, что позволяет использовать специфические функциональные возможности разностно-срезовой обработки массивов данных [14, 15].

В настоящей статье представлены результаты описания в базисе модифицированной САА В.М. Глушкова алгоритмов параллельного суммирования элементов векторного массива данных.

МОДИФИКАЦИЯ БАЗИСА САА В.М. ГЛУШКОВА

Особенности операции суммирования элементов векторного массива M необходимо учитывать при дополнении аппарата модифицированных САА [1–3]. Здесь также определяющим будет понятие размеченного массива M [2, 3]. Пусть массив M имеет вид

$$M = \text{ПЛ} \uparrow a_1 a_2 \dots a_n *, \quad (1)$$

где ПЛ, * — специальные маркеры, отмечающие соответственно левый и правый концы размеченного массива; \uparrow — указатель; n — размерность (длина) массива.

По классификации способа обработки элементов массива [1, 3] далее будет рассмотрена левосторонняя обработка массива M в результате циклического сдвига указателя \uparrow слева направо.

Наряду с такими известными конструкциями структурного программирования, как композиция, альтернатива и цикл [2, 7], активно используется декомпозиция $(A+B)$ как операция параллельного применения операторов A и B , используемая при описании синхронных алгоритмов сортировки парным обменом [4]. Базисные условия (предикаты) включают известные по работам [2, 3] условия,

а именно: $l \leq r$ — истинное при выполнении указанного отношения для соседних элементов l и r массива; $d(*)$ — истинное при достижении указателем \uparrow маркера $*$, а также новое условие: γ — истинное при выполнении соотношения $\min = 0$ для минимального значения в массиве M .

На составе базисных операторов остановимся подробнее. Будут широко использоваться такие известные операторы, как оператор начальной расстановки указателей (НРУ), оператор завершения работы разностного среза (РС) (*ФИН*), оператор сдвига указателя \uparrow на один элемент вправо (\vec{C}). Для описания в терминах регулярных схем алгоритмов разностно-срезовой обработки и, в частности, для параллельного суммирования элементов векторного массива необходимо ввести следующие базисные операторы:

*СУМ*₂(b, c) — суммирование элементов b и c массива;

*ВЫЧ*₂(b, c) — вычитание (последовательное) элементов b и c массива;

ВЫЧ(M, a) — вычитание (параллельное) из всех элементов массива M элемента a ;

ВЫВ(R) — вывод результата.

Представленный математический аппарат позволяет описать в терминах операторных представлений САА и проанализировать все способы (алгоритмы) реализации обработки векторного массива данных с использованием разностных срезов.

РАЗНОСТНО-СРЕЗОВАЯ ОБРАБОТКА

Основным понятием при разностно-срезовой обработке является понятие РС [13], определяющее его в текущем цикле обработки как векторный массив. Каждый его элемент формируется как разность соответствующего элемента текущего массива и текущего внутреннего порога обработки, в качестве которого может использоваться элемент фиксированной позиции текущего массива, например наименьший элемент.

В дальнейшем будем рассматривать векторные массивы чисел в качестве РС, обозначать их как A_j , где j — j -й цикл обработки РС, а для исходного РС вида A_0 примем, что все его элементы $a_{i,0}$ — ненулевые и положительные числа.

Базовыми операциями для мультиобработки по РС являются [13, 14]:

а) формирование элементов вектора внутреннего порога обработки вида

$$\mathbf{q} = \{q_j\}_{j=1}^N, \quad (2)$$

где N — количество циклов обработки, причем элемент q_j определяется следующим образом:

$$q_j = \min_i A_{j-1} = \min_i \{a_{i,j-1}\}_{i=1}^n; \quad (3)$$

б) формирование текущего РС A_j вида

$$A_j = A_{j-1} - q_j = \{a_{i,j-1} - q_j\}_{i=1}^n; \quad (4)$$

в) накопление частичных сумм S_j вида [14]

$$S_N = \sum_{j=1}^N S_j = \sum_{j=1}^N q_j \left(\sum_{i=1}^n f_{i,j} \right), \quad (5)$$

где $f_{i,j}$ — i -й элемент j -го вектора бинарной матрицы \mathbf{F} масок, который формируется одновременно с РС A_j следующим образом [13, 14]:

$$f_{i,j} = \begin{cases} 1, & \text{если } a_{i,j} \geq 0, \\ 0, & \text{если } a_{i,j} < 0. \end{cases} \quad (6)$$

Заканчивается процесс многооперандного суммирования элементов исходного РС A_0 при выполнении условия

$$q_j = 0. \quad (7)$$

Последовательное применение указанных базовых алгоритмов позволяет:

а) сформировать окончательную сумму элементов исходного РС A_0 как результат мультисуммирования [10]

$$S_N = \sum_{i=1}^n a_{i,0} \quad (8)$$

в виде векторно-матричного умножения [13, 14]

$$S_N = (\mathbf{n}^T \cdot \mathbf{F}) \cdot \mathbf{q}, \quad (9)$$

где \mathbf{n} — единичный вектор; T — символ транспонирования;

б) отсортировать элементы исходного РС A_0 в порядке возрастания [14]:

$$A_0^s = \mathbf{G}^T \cdot A_0, \quad (10)$$

где \mathbf{G} — матрица сортировки, элементы $g_{i,j}$ которой определяются следующим образом:

$$g_{i,j} = \begin{cases} 1, & \text{если } a_{i,j} = 0, \\ 0, & \text{если } a_{i,j} \neq 0; \end{cases} \quad (11)$$

в) восстановить элементы исходного РС A_0 [14]:

$$A_0 = \mathbf{F} \cdot \mathbf{q}. \quad (12)$$

Кроме того, учитывая возможность совмещения в каждом j -м цикле операции формирования частичной суммы S_j и сравнения ее с текущим внешним порогом θ_j обработки [13], целесообразно ввести в рассмотрение пороговую функцию активации с формированием выходного сигнала Y :

$$\theta_j = \theta_{j-1} - S_j, \quad (13)$$

$$Y = \begin{cases} 1, & S_N \geq \theta_0, \\ 0, & S_N < \theta_0, \end{cases} \quad (14)$$

где θ_0 — начальный внешний порог обработки.

Исходя из этого, окончанием пороговой обработки считается выполнение условия

$$\theta_j \leq 0. \quad (15)$$

Таким образом, базовыми операциями разностно-срезовой обработки векторного массива данных являются операции вида (3)–(5), (13) и (14), а базовыми условиями — (7) и (15).

ОПИСАНИЕ АЛГОРИТМОВ РАЗНОСТНО-СРЕЗОВОЙ ОБРАБОТКИ В ТЕРМИНАХ САА В.М. ГЛУШКОВА

Учитывая, что в данной работе рассматриваются только процедуры мультисуммирования (8), мультисортировки (10) и пороговой обработки (14), а также базовая операция формирования РС (4), целесообразно определить соответствующие составные операторы, которые базируются на использовании уже выделенных базисных операторов и базисных предикатов. Далее необходимо ввести обозначение для массива M с указанием его крайних элементов в виде (a_1, a_n) для выделения зоны составных операторов.

При мультисуммировании (8) и мультисортировке (10) составные операторы должны иметь следующий вид:

а) составной оператор выделения минимального числа среди элементов $\overline{(a_1, a_n)}$ РС A_j

$$МИН_2(\overline{a_1, a_n}) ::= [d(*)] \{ [l \leq r] (УСТ(\min = l) \vee УСТ(\min = r) \vec{C}) \}, \quad (16)$$

с помощью которого реализуется последовательное выделение минимального элемента в каждой паре элементов массива (a_1, a_n) со сдвигом на один элемент вправо по массиву и с установкой (назначением) элементу \min одного из двух значений;

б) составной оператор накопления частичных сумм элементов $\overline{(a_1, a_n)}$ РС A_j

$$НАКОП_2(\overline{a_1, a_n}) ::= [d(*)] \{ [f_i = 0] (E \vee СУМ_2(S, \min) \vec{C}) \}, \quad (17)$$

с помощью которого реализуется последовательное накопление частичных сумм S_j вида (5), для чего используется проверка равенства нулю соответствующего элемента f_i (6) в текущем векторе бинарной матрицы \mathbf{F} масок;

в) составной оператор формирования текущего вектора бинарной матрицы \mathbf{F} масок

$$МАСКА(\overline{a_1, a_n}) ::= [dif(r_i) < 0] (УСТ(f_i = 0) \vee УСТ(f_i = 1)), \quad (18)$$

включающий проверку условия отрицательного результата r_i вычитания (4) и соответствующую установку элемента f_i (6).

Для пороговой обработки (14) необходимо ввести составной оператор для реализации пороговой функции активации вида

$$АКТИВ(\theta, S) ::= [dif(r) < 0] (УСТ(Y = 1) \vee УСТ(Y = 0)), \quad (19)$$

включающий проверку условия отрицательного результата r вычитания (13) и соответствующую установку выходного сигнала Y (14).

Поскольку формирование РС A_j в каждом j -м цикле мультисуммирования является базовой операцией, запишем ее с помощью составных и базовых операторов

$$СПЕЗ(\overline{a_1, a_n}) ::= НРУ \times МИН_2(\overline{a_1, a_n}) \times (ВЫЧ(M, \min) + МАСКА(\overline{a_1, a_n})) \quad (20)$$

с учетом одного оператора последовательного действия (16) и двух операторов параллельного действия — $ВЫЧ(M, \min)$ и (18), которые выполняются одновременно.

Алгоритм сортировки элементов исходного РС A_0 в процессе мультисуммирования, формирующий дополнительный результат (отсортированный массив) по отношению к основному результату (сумме элементов массива), представим следующим образом:

$$\begin{aligned} СОРТ^n(\overline{a_1, a_n}) ::= & [\gamma] \{ НРУ \times МИН_2(\overline{a_1, a_n}) \times НРУ \times (ВЫЧ(M, \min) + \\ & + МАСКА(\overline{a_1, a_n}) + НАКОП_2(\overline{a_1, a_n})) \times ВЫВ(S) \} \times ФИИ. \end{aligned} \quad (21)$$

В записи (21) используется оператор $ВЫВ(S)$ для вывода результата операции накопления в каждом j -м цикле обработки. Выходом из процесса сортировки является выполнение условия (7), представленного базовым предикатом γ .

Базовый алгоритм формирования суммы элементов исходного РС A_0 запишем в таком виде:

$$\begin{aligned} СУММА^n(\overline{a_1, a_n}) ::= & [\gamma] \{ НРУ \times МИН_2(\overline{a_1, a_n}) \times \\ & \times (ВЫЧ(M, \min) + МАСКА(\overline{a_1, a_n})) \times НРУ \times \\ & \times НАКОП_2(\overline{a_1, a_n}) \times СУМ_2(R, S) \} \times ВЫВ(R) \times ФИИ, \end{aligned} \quad (22)$$

которую можно сократить, воспользовавшись операцией $СПЕЗ(\overline{a_1, a_n})$ вида (20):

$$\begin{aligned} СУММА^n(\overline{a_1, a_n}) ::= & [\gamma] \{ СПЕЗ(\overline{a_1, a_n}) \times НРУ \times \\ & \times НАКОП_2(\overline{a_1, a_n}) \times СУМ_2(R, S) \} \times ВЫВ(R) \times ФИИ. \end{aligned} \quad (23)$$

Это первый вариант алгоритма суммирования, в котором постепенно за N циклов накапливаются текущие частичные суммы S_j вида (5).

Второй вариант для алгоритма суммирования запишем в следующем виде:

$$\begin{aligned} \overline{СУММА}^n(a_1, a_n) ::= [\gamma] \{ & \overline{НРУ} \times (\overline{МИН}_2(a_1, a_n) + \overline{НАКОП}_2(a_1, a_n)) \times \\ & \times (\overline{ВЫЧ}(M, \min) + \overline{МАСКА}(a_1, a_n) + \overline{СУМ}_2(R, S)) \} \times \overline{ВЫВ}(R) \times \overline{ФИН}. \end{aligned} \quad (24)$$

Для второго варианта характерным является наличие в первом цикле двух «пустых» операций: $\overline{НАКОП}_2(a_1, a_n)$ и $\overline{СУМ}_2(R, S)$, поскольку они выполняются в данном случае с опережением.

Для третьего варианта алгоритма суммирования характерна запись вида

$$\begin{aligned} \overline{СУММА}^n(a_1, a_n) ::= \overline{СПЕЗ}(a_1, a_n) \times [\gamma] \{ & \overline{НРУ} \times \overline{НАКОП}_2(a_1, a_n) \times \\ & \times \overline{СУМ}_2(R, S) \times \overline{СПЕЗ}(a_1, a_n) \} \times \overline{ВЫВ}(R) \times \overline{ФИН}, \end{aligned} \quad (25)$$

в которой операция формирования первого РС выведена за пределы цикла.

Четвертый вариант записи алгоритма суммирования имеет вид

$$\begin{aligned} \overline{СУММА}^n(a_1, a_n) ::= \overline{НРУ} \times \overline{МИН}_2(a_1, a_n) \times [\gamma] \{ & (\overline{ВЫЧ}(M, \min) + \\ & + \overline{МАСКА}(a_1, a_n) + \overline{СУМ}_2(R, S)) \times \overline{НРУ} \times (\overline{МИН}_2(a_1, a_n) + \\ & + \overline{НАКОП}_2(a_1, a_n)) \} \times \overline{ВЫВ}(R) \times \overline{ФИН}, \end{aligned} \quad (26)$$

где за пределы цикла выведена операция определения минимального элемента в исходном РС A_0 .

Таким образом, четыре варианта записи в терминах САА В.М. Глушкова алгоритма суммирования с использованием РС отличаются только последовательностью применения введенных базисных и составных операторов и предикатов в пределах и за пределами цикла обработки.

Для пороговой обработки по РС первый вариант соответствующего алгоритма имеет вид

$$\begin{aligned} \overline{ПОРОГ}^n(a_1, a_n) ::= \overline{СУММА}^n(a_1, a_n) \times \\ \times \overline{ВЫЧ}_2(\theta, R) \times \overline{АКТИВ}(\theta, R) \times \overline{ВЫВ}(Y) \times \overline{ФИН} \end{aligned} \quad (27)$$

или, при детальном описании с использованием записи (23),

$$\begin{aligned} \overline{ПОРОГ}^n(a_1, a_n) ::= [\gamma] \{ \overline{СПЕЗ}(a_1, a_n) \times \overline{НРУ} \times \overline{НАКОП}_2(a_1, a_n) \times \\ \times \overline{СУМ}_2(R, S) \} \times \overline{ВЫЧ}_2(\theta, R) \times \overline{АКТИВ}(\theta, R) \times \overline{ВЫВ}(Y) \times \overline{ФИН}. \end{aligned} \quad (28)$$

Поскольку при обработке по РС возможно совмещенное выполнение некоторых операций пороговой обработки, ниже приведены еще три возможных варианта представления алгоритма пороговой обработки.

Второй вариант:

$$\begin{aligned} \overline{ПОРОГ}^n(a_1, a_n) ::= [\gamma] \{ \overline{НРУ} \times (\overline{МИН}_2(a_1, a_n) \times \overline{НАКОП}_2(a_1, a_n)) \times \\ \times (\overline{ВЫЧ}(M, \min) + \overline{МАСКА}(a_1, a_n) + \overline{СУМ}_2(R, S) + \\ + \overline{ВЫЧ}_2(\theta, R)) \times \overline{АКТИВ}(\theta, R) \} \times \overline{ВЫВ}(Y) \times \overline{ФИН}; \end{aligned} \quad (29)$$

третий вариант:

$$\begin{aligned} \overline{ПОРОГ}^n(a_1, a_n) ::= \overline{СПЕЗ}(a_1, a_n) \times [\gamma] \{ \overline{НРУ} \times \overline{НАКОП}_2(a_1, a_n) \times \\ \times (\overline{СУМ}_2(R, S) \times \overline{ВЫЧ}_2(\theta, S)) \times (\overline{СПЕЗ}(a_1, a_n) + \\ + \overline{АКТИВ}(\theta, S)) \} \times \overline{ВЫВ}(Y) \times \overline{ФИН}; \end{aligned} \quad (30)$$

четвертый вариант:

$$\begin{aligned} \text{ПОРОГ}^n(\overline{a_1, a_n}) ::= & \text{HPY} \times \text{МИН}_2(\overline{a_1, a_n}) \times [\gamma] \{ (\text{ВЫЧ}(M, \min) + \\ & + \text{МАСКА}(\overline{a_1, a_n}) + \text{ВЫЧ}_2(\theta, S) + \text{СУМ}_2(R, S)) \times \text{HPY} \times (\text{МИН}_2(\overline{a_1, a_n}) + \\ & + \text{НАКОП}_2(\overline{a_1, a_n}) + \text{АКТИВ}(\theta, R)) \} \times \text{ВЫВ}(Y) \times \text{ФИН}. \end{aligned} \quad (31)$$

Анализ вариантов алгоритма пороговой обработки (27)–(31) свидетельствует о том, что в них использованы варианты (22)–(26) алгоритма суммирования как основной составляющей мультиобработки по РС.

ЗАКЛЮЧЕНИЕ

Функциональная мощность базиса САА В.М. Глушкова позволяет на его основе, дополнив базисными и составными операторами и предикатами, учитывающими специфику обработки данных, представить в терминах регулярных схем любой алгоритм мультиобработки, в частности с использованием разностных срезов.

Все многообразие реализаций любого алгоритма мультиобработки в модифицированном базисе САА В.М. Глушкова можно изложить в компактной и наглядной форме, что не только упрощает процесс анализа таких представлений, но и способствует их дальнейшему развитию (оптимизации и модернизации).

СПИСОК ЛИТЕРАТУРЫ

1. Цейтлин Г.Е. Структурное программирование задач символьной мультиобработки // Кибернетика. — 1983. — № 5. — С. 22–30.
2. Цейтлин Г.Е. Проектирование последовательных алгоритмов сортировки: классификация, трансформация, синтез // Программирование. — 1989. — № 3. — С. 3–24.
3. Цейтлин Г.Е. Распараллеливание алгоритмов сортировки // Кибернетика. — 1989. — № 6. — С. 67–74.
4. Кожемяко В.П., Мартынюк Т.Б., Хомяк В.В. Особенности структурного программирования синхронных алгоритмов сортировки // Кибернетика и системный анализ. — 2006. — № 5. — С. 122–133.
5. Яценко Е.А. Регулярные схемы алгоритмов адресной сортировки и поиска // Управляющие системы и машины. — 2004. — № 5. — С. 61–66.
6. Цейтлин Г.Е., Амонс А.А., Головин О.В., Зубцов А.Ю. Интегрированный проектирования и синтеза классов алгоритмов и программ // Кибернетика и системный анализ. — 2000. — № 3. — С. 165–169.
7. Цейтлин Г.Е. Алгебры Глушкова и теория клонов // Там же. — 2003. — № 4. — С. 48–58.
8. Борисов Е.С. Полуавтоматическая система декомпозиции последовательных программ для параллельных вычислителей с распределенной памятью // Там же. — 2004. — № 3. — С. 139–150.
9. Цейтлин Г.Е., Иванов Е.А. Специализированные информационные технологии для лиц с физическими ограничениями // Управляющие системы и машины. — 2008. — № 5. — С. 62–69, 74.
10. Мартинюк Т.Б., Хомяк В.В. Методи та засоби паралельних перетворень векторних масивів даних. — Вінниця: УНІВЕРСУМ – Вінниця, 2005. — 203 с.
11. Ни Л.М., Джейн А.К. Двухуровневый конвейерный систолический массив для кластерного анализа // СБИС для распознавания образов и обработки изображений: Пер. с англ. / Под ред. К. Фу. — М.: Мир, 1988. — 248 с.
12. Справочник по цифровой вычислительной технике / Под ред. В.Н. Малиновского. — К.: Техніка, 1980. — 320 с.
13. Мартынюк Т.Б. Модель порогового нейрона на основе параллельной обработки по разностным срезам // Кибернетика и системный анализ. — 2005. — № 4. — С. 78–89.
14. Мартинюк Т.Б. Рекурсивні алгоритми багатоперандної обробки інформації. — Вінниця: УНІВЕРСУМ – Вінниця, 2000. — 216 с.
15. Васюра А.С., Мартинюк Т.Б., Куперштейн Л.М. Методи та засоби нейроподібної обробки даних для систем керування. — Вінниця: УНІВЕРСУМ – Вінниця, 2008. — 175 с.

Поступила 06.11.2009