ФУНДАМЕНТАЛЬНІ НАУКИ

УДК 681.3

МОЖЛИВОСТІ РОЗПАРАЛЕЛЮВАННЯ АЛГОРИТМУ БАГАТООПЕРАНДНОГО ДОДАВАННЯ

Канд. техн. наук, доц. Мартинюк Т. Б., Хомюк В. В., студ. Кухарчук Г. В.

Однією з найпоширеніших операцій більшості прикладних задач обробки сигналів та зображень є операція багатооперандного додавання або оператор групового підсумовуваня [1] виду

$$X_n = \sum_{i=1}^{n} a_i \,. \tag{1}$$

Ця операція є інтегральною операцією [2] і внаслідок цього має недостатній для алгоритмів обробки сигналів та зображень рівень паралелізму.

Метою даної роботи є дослідження можливостей розпаралелювання операції багатооперандного додавання (1).

Операція (1) є асоціативною бінарною операцією на довільній множині n, що відповідає орієнтованому ациклічному графу G, і є бінарним деревом (рис.1а), де прийняті такі позначення: t — кількість кроків в часі; p — кількість процесорів; ω — загальна кількість операцій алгоритму; n — кількість параметрів задачі; $T_p(n)$ — час виконання алгоритму; τ — час виконання одного кроку; L — кількість типів операцій.

До бінарного дерева використовують перетворення у відповідності із законами комутативності, асоціативності і дистрибутивності з метою мінімізації величини $T_p(n)$. В результаті бінарне дерево G трансформується в орієнтований ациклічний граф $\hat{G}$.

Відомо, що основний принцип розпаралелювання послідовних числових алгоритмів полягає в знаходжені способу перетворення виразу E, що відповідає послідовному алгоритму, в еквівалентний вираз $\hat{E}$, орієнтований ациклічний граф $\hat{G}$ якого мінімізовано за часовими параметрами в порівнянні з орієнтованим ациклічним графом G, що відповідає виразу E [3].

Розглянемо один із способів перетворення виразу E, що відповідає послідовному алгоритму (рис.1а), в еквівалентний вираз $\hat{E}$, відповідний паралельному алгоритму. Для цього скористаємося схемою Горнера, алгоритмом Винограда [3] та поняттями еквівалентних арифметичних виразів E і $\hat{E}$, а також поняттям альтернованого виразу n змінних, тобто виразом, що має такий вигляд

$$E_n\ (\ldots(a_1\upsilon_1 a_2)\ \upsilon_2 a_3)\ \upsilon_3 a_4 \ldots)\upsilon\ \upsilon_{n-1} a_n, \tag{2}$$

де υ_i — позначає одну із операцій $\theta = \{+,-,\times,/,\div\}$, причому υ_i, υ_{i+1} — операції різних типів; $i = \overline{1,\, n-1}$.

Послідовний алгоритм додавання n операндів у загальному вигляді можна записати як арифметичний вираз E

$$E = (\ldots(a_1 + a_2) + a_3 \ldots) + a_n. \tag{3}$$

Оскільки за визначенням [3] лінійною рекурентною задачею порядку m для n рівнянь називається задача R (n, m) вигляду

$$X_k = \begin{cases} 0, & k \le 0, \\ C_k + \sum\limits_{j=k-m}^{k-1} a_{kj} X_j, & 1 \le k \le n, \end{cases} \tag{4}$$

то задачу додавання n операндів можна записати у вигляді лінійної рекурентної задачі порядку 1 зі скалярним результатом

$$X_k = a_k + X_{k-1}, \qquad 1 \le k \le n, \tag{5}$$

де $X_0 = 0$; X_n — кінцевий результат.

Для розв'язання лінійної рекурентної задачі порядку m відомі три паралельних алгоритми [3], з яких алгоритм логарифмічного додавання або рекурсивного подвоєння найефективніший для задачі додавання n операндів [4]. Останній застосовується до асоціативних операцій: додавання і множення дійсних чисел, додавання і множення матриць, до операцій min та max з дійсними аргументами і відноситься до так званих алгоритмів «пошуку по дереву» [4].

Алгоритм рекурсивного подвоєння використовує такий еквівалентний адитивний альтернований вираз $\hat{E}$

$$\hat{E} = (\ldots(((a_1 + a_2) + (a_3 + a_4)) + \ldots + ((a_{n-3} + a_{n-2}) + (a_{n-1} + a_n)))\ldots), \tag{6}$$

що не дає наглядного уявлення про структуру зв'язків між змінними виразу $\hat{E}$.

В загальному вигляді вираз (6), що використовує алгоритм Винограда, можна записати наступним чином

$$\hat{E} = (\ldots(F_1 \upsilon_1 F_2)\upsilon_2 F_3 \ldots)\upsilon_{k-1} F_k, \tag{7}$$

де F_i — підвираз E; $\upsilon_i \in \theta$.

Відомо, що найефективніше піддаються процесу разпаралелювання ті числові послідовні алгоритми, що містять векторні елементи, тобто операції з векторами та матрицями [3]. В цьому випадку процес виконання алгоритму рекурсивного подвоєння можна розглядати як додавання двох векторів, що формуються на кожному кроці додавання із парних та непарних елементів поточного вектора. Розглянемо цей процес з урахуванням виконання дій алгоритму Винограда [3].

Припустимо, що задано первісний n-вимірний вектор вигляду

$$X_0 = (a_1^{(0)}, a_2^{(0)}, \ \ldots\ , \ a_n^{(0)}), \tag{8}$$

з якого можна отримати два]n/2[— вимірних вектори, елементи яких складаються з непарних та парних елементів первісного вектора відповідно:

$$F_0' = \left(a_1^{(0)}, \ldots, a_{2m-1}^{(0)}, \ldots, a_{n-1}^{(0)}\right), \tag{9}$$

$$F_1 = \left(a_2^{(0)}, \ldots, a_{2m}^{(0)}, \ldots, a_n^{(0)}\right), \tag{10}$$

де m = 1, 2, ...]n/2[;]c[— округлення до більшого цілого числа c.

Таким чином, виконання алгоритму Винограда для даного випадку дозволяє визначити результат додавання двох векторів на першому кроці алгоритму рекурсивного подвоєння у вигляді]n/2[— вимірного вектора вигляду

$$X_1 = F_0' + F_1 = \left(a_1 + a_2, a_3 + a_4, \ldots, a_{2m-1} + a_m, \ldots, a_{n-1} + a_n\right) \tag{11}$$

або

$$X_1 = \left(a_1^{(1)}, a_2^{(1)}, \ldots, a_{]n/2[}^{(1)}\right). \tag{12}$$

Розкладемо вектор X_1 на два]$n/2^2$[-вимірних вектори F'_1 і F_2 способом, аналогічним для виразів (9) та (10), тобто

$$F_1' = \left(a_1^{(1)}, \ldots, a_{]n/2[-1}^{(1)}\right), \tag{13}$$

$$F_2 = \left(a_2^{(1)}, \ldots, a_{]n/2[}^{(1)}\right). \tag{14}$$

Тоді на другому кроці алгоритму рекурсивного подвоєння буде отримано]$n/2^2$[-вимірний вектор X_2 такого виду:

$$X_2 = F_1' + F = \left(a_1^{(1)} + a_2^{(1)}, \ldots, a^{(1)}_{]n/2[-1} + a^{(1)}_{]n/2[}\right), \tag{15}$$

або

$$X_2 = \left(a_1^{(2)}, a_2^{(2)}, \ldots, a^{(2)}_{]n/2^2[}\right). \tag{16}$$

За аналогією на k-му кроці алгоритму рекурсивного подвоєння ($k = \log_2 n$) буде отримано одновимірний вектор, тобто скалярний результат X_k виду

$$X_k = F'_{k-1} + F_k = \left(a_1^{(k-1)} + a_2^{(k-1)}\right), \tag{17}$$

або

$$X_k = a_1^{(k)}. \tag{18}$$

Таким чином, використовуючи наявність векторних елементів під час реалізації алгоритму рекурсивного подвоєння, еквівалентний адитивний вираз (6) з урахуванням виразу (7) стає можливим записати таким чином

$$\hat{E} = \varphi\left(\ldots\ \varphi\left(F_0^1 \upsilon_1 F_1\right) \upsilon_2 F_2\right) \upsilon_3 \ldots\Big) \upsilon_k F_k, \tag{19}$$

де $\upsilon_1 = \ldots = \upsilon_k$ — операція додавання, φ — неарифметична операція формування векторів, елементи яких є непарними елементами векторів, отриманих на попередніх кроках алгоритму, тобто

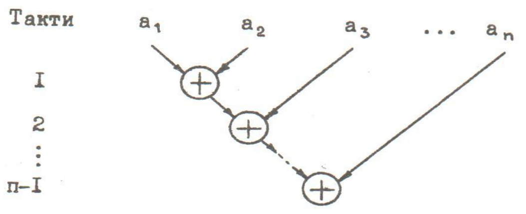
$$\varphi(X_{i-1}) = F'_{i-1} = \left(a_1^{i-1}, \ldots, a_{2m-1}^{i-1}, \ldots, a^{i-1}_{]n/2^i[-1}\right), \tag{20}$$

де $i = \overline{1, k}$; $F_1, \ldots, F_k$ — підвирази, які визначаються в загальному вигляді у відповідності з формулою:

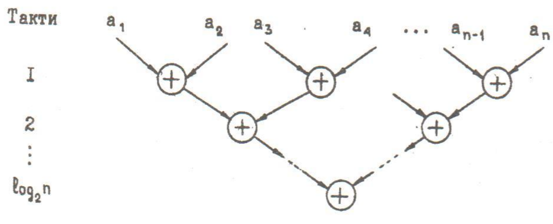
$$F_i = \left(a_2^{i-1}, \ldots, a_{2m}^{i-1}, \ldots, a^{i-1}_{]n/2^{i-1}[}\right). \tag{21}$$

Таким чином, у виразі (7) в алгоритмі Винограда для випадку розпаралелювання задачі додавання n операндів необхідно ввести неарифметичну операцію φ, яка враховує наявність векторних елементів в даній задачі, тобто $\theta = \{+, \times, /, \div, \varphi\}$.

Відповідний еквівалентному виразу $\hat{E}$ (19) орієнтований ациклічний граф $\hat{G}$ зображено на рис.1б.



а)



б)

Рис.1. а) $p = 1$, $\omega = (n - 1)L$, $t = n - 1$, $T_1 = (n - 1)\tau$, $L = 1$;
б) $n/2 \le p \le n - 1$, $\omega = (n - 1)L$, $t = \log_2 n$, $T_p = \tau \log_2 n$, $L = 1$

Аналіз двох наданих алгоритмів (рис.1а, 1б) свідчить про те, що хоча загальна кількість обрахувань (ω) складає $0(n)$ для двох варіантів, час виконання (T_p) для паралельного алгоритму складає $0(\log_2 n)$ у порівнянні з $0(n)$ для послідовного алгоритму. Таким чином основна мета мінімізації часових параметрів послідовного алгоритму додавання n операндів досягнута у алгоритмі рекурсивного подвоєння, який являє собою «деревовидний» алгоритм, оскільки залежності, які входять в даний алгоритм, утворюють «деревовидну» структуру (рис.1б), кожний вузол якої являє собою двовхідний суматор [4]. Використання алгоритму Винограда, врахування векторних елементів, а також розширення *списку операцій за рахунок неарифметичних операцій* дає можливість отримати ще один вираз $\hat{E}$, еквівалентний виразу E (3).

Відомий алгоритм паралельного додавання n операндів базується на принципі, який визначений авторами як формування зрізів різниць [5]. Суть цього алгоритму полягає в тому, що кожна дія (крок) включає в себе виділення загальної значущої частини серед n величин, яке виконується в процесі операції порівняння, та формування кратного цієї загальної частини, причому кратність визначається кількістю величин даної дії. Кожна наступна дія починається з виділення значущих різниць шляхом порівняння всіх величин з загальною частиною, які і є вихідними величинами для цієї дії. Кінцевий результат отримується в процесі накопичення кратних всіх n дій [6]. Таким чином, в основу алгоритму багатооперандного додавання покладено виконання таких простих алгоритмів: 1) порівняння між собою вхідних числових величин, які складають первісну множину; 2) виділення загальної значущої (мінімальної) величини поточної множини; 3) формування зрізів різниць, які являють собою проміжні множини числових величин, тобто сукупність величин різниці всіх числових величин з загальною частиною поточної множини.

В результаті виконується не тільки арифметична операція багатооперандного додавання чисел, але й з'являється можливість реалізації задач асоціативного пошуку інформації, до яких відносяться задачі визначення екстремальних чисел та упорядкована вибірка чисел. Крім того, існує можливість відновлення множин числових величин.

Таким чином, можна вважати, що алгоритм багатооперандного додавання на основі формування зрізів різниць володіє поширеними функціональними можливостями, які вказані вище, справедливість виконання яких доведена в процесі розробки математичної моделі багатооперандної (багатомісної або групової) обробки числової інформації [5, 7].

Еквівалентний адитивний альтернований вираз $\hat{E}$ для алгоритму, який використовує зрізи різниць, можна записати у наступному вигляді

$$\hat{E} = (\ldots \quad (F_1 + F_2) + F_3) + \quad \ldots) + F_n, \tag{22}$$

розглядаючи вихідні та проміжні дані у вигляді n-вимірних векторів X_j. В цьому випадку підвираз F_j можна зобразити у вигляді

$$F_j = q_j p_j\,, \tag{23}$$

де q_j — найменший ненульовий елемент вектора X_j; p_j — кількість ненульових елементів вектора X_j; $j = \overline{1,\ n}$; причому

$$q_j = \min(X_{j-1}). \tag{24}$$

Крім того, кожний елемент $a_i^{(j)}$ n-вимірного вектора X_j формується наступним чином

$$a_i^{(j)} = a_i^{(j-1)} - q_j\,, \tag{25}$$

причому первісний n-вимірний вектор X_0 має вигляд (8); $q_0 = 0$; $i = \overline{1,\ n}$.

Таким чином, необхідно розширити список операцій θ алгоритму Винограда шляхом введення в нього операції min над елементами первісного та проміжних векторів.

Орієнтований ациклічний граф $\hat{G}$, відповідний виразу $\hat{E}$ (22) для алгоритму на базі зрізів різниць представлено на рис. 2. Аналіз параметрів наданого алгоритму багатооперандного додавання свідчить про те, що час його виконання T_p складає $0(n)$, загальне число обчислень ω — $0(n^2)$, а число процесорів $p = n$. Використання відомої методики для визначення ефективності паралельних алгоритмів за такими параметрами як

$$S_p(n) = \frac{T_1(n)}{T_p(n)} \le p\,; \tag{26}$$

$$E_p(n) = \frac{S_p(n)}{p} \le 1\,, \tag{27}$$

де $S_p(n)$, $E_p(n)$, — прискорення та ефективність паралельного алгоритму відповідно, $T_1(n)$ — час виконання послідовного алгоритму ($p = 1$), не дозволяє визначити алгоритм на базі зрізів різниць ефективним паралельним алгоритмом. При цьому така унікальна властивість цього алгоритму як можливість одночасної реалізації декількох алгоритмів на одній структурі не враховується. В той же час методика оцінювання ефективності паралельних алгоритмів з використанням коефіцієнта узгодження [8] дозволяє врахувати багатофункційність структури для даного алгоритму і дає можливість розглядати алгоритм багатоперандного додавання на базі зрізів різниць як один з найпристосованіших до реалізувальної його структури [9]. При цьому належність його до класу регулярних ітераційних алгоритмів дозволяє реалізувати цей алгоритм на перспективній елементній базі, тобто на лінійній систолічній структурі [9].

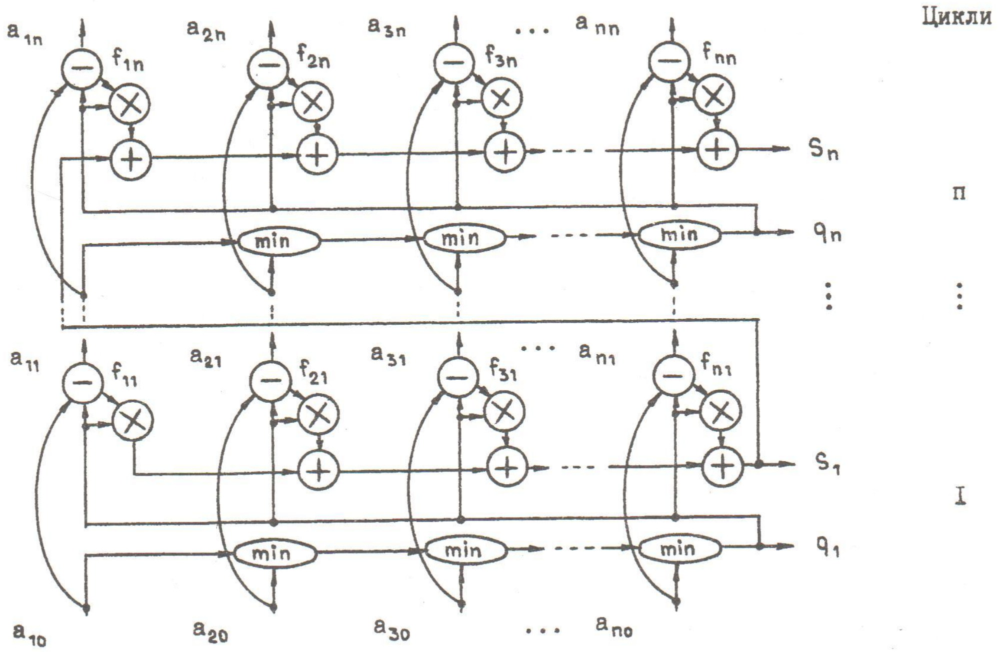


Рис. 2. $P = n$, $\omega = n^2L$, $t = n$, $T_p = n\tau$, $L = 4$

ВИСНОВОК

Використання алгоритму Винограда та наявність векторних елементів дозволило обгрунтувати алгоритм багатоперандного додавання на базі зрізів різниць як один з можливих варіантів розпаралелювання послідовного алгоритму додавання n операндів поряд з відомим алгоритмом рекурсивного подвоєння.

ЛІТЕРАТУРА

1. Справочник по цифровой вычислительной технике /Под ред. Б. Н. Малиновского. — К.: Техніка, 1980. — 320 с.
2. Карцев М. А., Брик В. А. Вычислительные системы и синхронная арифметика. — М.: Радио и связь, 1981. —360 с.
3. Фейлмейер М. Параллельные численные алгоритмы // Системы параллельной обработки / Под ред. Д. Ивенса. — М.: Мир, 1985. — С 285—337.
4. Джагадиш Х. В., Рао С. К., Кайлат Т. Матричные структуры для реализации итерационных алгоритмов // ТИИЭР. — 1987. — Т. 75. — № 9. — С 184—203.
5. Кожемяко В. П., Мартынюк Т. Б. и др. Параллельная обработка изображения. — Ужгород: изд-во Ужгород. гос. Ун-та, 1993. — 89 с.
6. Квазиимпульсно-потенциальные оптоэлектронные элементы и устройства логико-временного типа / Свечников С. В., Кожемяко В. П., Тимченко Л. И. — К.: Наук. думка, 1987. — 256 с.
7. А. с. 1119035 СССР. Способ параллельного сложения длительностей временных интервалов / Кожемяко В. П. и др. // Бюл. изобр. — 1984. — №38.
8. Трухний В. Д. Структурно-информационные оценки качества цифровых вычислительных машин // Экономика и математические методы. — 1969. — Вып. 1. — 79—89 с.
9. Оцінювання структурно-інформаційної складності паралельних алго-ритмів додавання / Т. Б. Мартинюк, Н. І. Заболотна, В. В. Шолота // Вісник ВПІ. — 1996. — № 4. — С. 21—26.

Кафедра оптоелектронного приладобудування