К.В. Харченко

Модифікація паралельного блочно-діагонального методу розв'язання лінійних систем рівнянь для використання у САПР електронних схем

Розглянемо рішення несиметричної системи блочно-діагонального вигляду з обрамленням

$$\begin{matrix} A_{11} & 0 & \dots & A_{1p} & x_1 & b_1 \\ 0 & A_{22} & \dots & A_{2p} & x_2 & b_2 \\ \dots\dots & & .. = & \dots & & \\ A_{p1} & A_{p2} & \dots & A_{pp} & x_p & b_p \end{matrix} \qquad (1)$$

на p процесорах. Тоді використовуючи систему передачі повідомлень між паралельними процесорами та алгоритм [1] можливо знайти рішення цієї системи за час, менший ніж на однопроцесорній системі.

Розглянемо складність базового паралельно-послідовного алгоритму блочного LU-розкладу на одному процесорі та p процесорах для матриці $n \times n$ з обрамленням розміром m та p блоками розміром $k=(n-m)/(p-1)$, c - середня кількість ненульових елементів у строці.

Приймаючи складність послідовного алгоритму факторизації матриці розміром $n \times n$ за $O(n)$ та відповідного прямого та зворотнього ходу як $O_L(n)$ та $O_U(n)$, час обчислення операції множення t_m та операції додавання t_c отримаємо:

$$T(1) = (p-1)[O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k)] + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k), \qquad (2)$$

$$T(p) = O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k) + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k). \qquad (3)$$

Тоді прискорення базового паралельно-послідовного алгоритму блочного LU розкладу можливо обчислити [2] як

$$S_p = \frac{T(1)}{T(p)}. \quad (4)$$

Вважаючи, що розмір діагональних блоків перевищує розмір обрамлення $(k >> m)$ отримаємо складність паралельної частини $(O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k))$ буде значно більшою за складність послідовної частини $(pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k))$.

Тобто прискорення базового паралельно-послідовного алгоритму блочного LU розкладу приблизно дорівнює

$$S_p \approx p-1. \quad (5)$$

Тобто з (5) видно, що потужність головгого процесору не використовується у такому варіанті паралельного блочно-діагонального методу розв'язання лінійних систем рівнянь.

Для вирішення проблеми використання повної потужності головного процесору проведемо модифікацію цього алгоритму так, щоб головний процесор також виконував рішення одного з діагональних блоків, за такою схемою:

1. В процесори з номерами $i < p-1$ передаються $A_{ii}, A_{ip}, A_{pi}, b_i$
2. В цих процесорах виконуються такі дії:
   факторизація матриці $A_{ii}: A_{ii} = L_{ii}U_{ii}$;
   розв'язання систем $A_{ii}C_i = A_{ip}$;
   визначення матриці $W_i = A_{pi}C_i$.
3. Одночасно у головному процесорі виконуються такі дії:
   факторизація матриці $A_{p-1,p-1}: A_{p-1,p-1} = L_{p-1,p-1}U_{p-1,p-1}$;
   розв'язання систем $A_{p-1,p-1}C_{p-1} = A_{p-1,p}$;
   визначення матриці $W_{p-1} = A_{p,p-1}C_{p-1}$.
4. Після виконання цих дій *i-й* процесор передає матрицю Wi до p-го процесору та переходить до виконання прямого ходу неявної блочної схеми, яка приведена в дії 5.
5. Процесор з номером p виконує такі дії:
   в темпі надходження матриць W_i знаходить матрицю $\tilde{A}_{pp} = A_{pp} - \sum_{i=1}^{p-1} W_i$;
   факторизує матрицю $\tilde{A}_{pp}: \tilde{A}_{pp} = L_{pp}U_{pp}$;
6. Паралельно - послідовний алгоритм прямого ходу неявної блочної схеми полягає в тому, що в процесорах з номерами $i < p-1$ одночасно виконуються такі дії:

розв'язання системи $L_{ii}y_i = b_i$;

розв'язання системи $U_{ii}t_i = y_i$;

визначення вектору $\tilde{t}_i = A_{pi}t_i$ та передача його в процесор з номером *p*.

7. Процесор з номером *p* виконує такі дії:

прямий хід неявної блочної схеми для передостаннього блоку:

розв'язання системи $L_{p-1,p-1}y_{p-1} = b_{p-1}$;

розв'язання системи $U_{p-1,p-1}t_{p-1} = y_{p-1}$;

визначення вектору $\tilde{t}_{p-1} = A_{p,p-1}t_{p-1}$;

в темпі надходження векторів $\tilde{t}_i$ знаходить вектор $\tilde{b}_p = b_p - \sum_{i=1}^{p-1}\tilde{t}_i$;

розв'язує систему $L_{pp}y_p = \tilde{b}_p$ та переходить до виконання зворотнього ходу неявної блочної схеми.

8. Паралельно-послідовний алгоритм зворотнього ходу неявної блочної схеми. Він полягає у тому, що процесор з номером *p* розв'язує систему $U_{pp}x_p = y_p$ та передає вектор x_p до кожного процесору з номерами $i < p - 1$.

9. У процесорах з номерами $i < p - 1$ одночасно виконуються такі дії:

визначення вектору $u_i = A_{ip}\, x_p$;

розв'язання системи $L_{ii}\tilde{u}_i = u_i$;

знаходження $\tilde{y}_i = y_i - \tilde{u}_i$;

розв'язання системи $U_{ii}x_i = \tilde{y}_i$;

та головний процесор з номером *p* виконує такі дії:

визначення вектору $u_{p-1} = A_{p-1,p}\, x_p$;

розв'язання системи $L_{p-1,p-1}\tilde{u}_{p-1} = u_{p-1}$;

знаходження $\tilde{y}_{p-1} = y_{p-1} - \tilde{u}_{p-1}$;

розв'язання системи $U_{p-1,p-1}x_{p-1} = \tilde{y}_{p-1}$;

10. Після виконання алгоритму в *i*-м процесорі (*i=1,...,p-2*) буде знаходиться вектор x_i, який є відповідною частиною вектору *x* - рішення переупорядкованної системи.

Розглянемо складність модифікованого паралельно-послідовного алгоритму блочного LU-розкладу на одному процесорі та p процесорах:

$$T(1) = p[O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k)] + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k) \qquad (6)$$

$$T(p) = O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k) + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k) \qquad (7)$$

Тоді вважаючи, що розмір діагональних блоків перевищує розмір обрамлення $(k >> m)$, прискорення модифікованого паралельно-послідовного алгоритму блочного LU розкладу можливо обчислити як

$$S_p = \frac{T(1)}{T(p)} \approx p\,. \qquad (8)$$

Таким чином, у модифікованому алгоритмі потужність головгого процесору буде використовуватися повністю, що підвищило прискорення до p разів.

Згідно закону Амдала [3] залежність прискорення S_p від кількості процесорів для прискорення модифікованого паралельно-послідовного алгоритму блочного LU розкладу має вигляд графіку на Рис. 1. З графіку видно, що прискорення зменьшується нелінійно з зростанням p, та вже для кількості процесорів близько 8 прискорення становить меньше 6.9. З цього можна зробити висновок, що при використанні паралельно-послідовного алгоритму блочного LU розкладу можливо передбачити кількість процесорів, необхідних для розрахунків за певний час.

Прискорення модифікованого паралельно-послідовного алгоритму блочного LU розкладу (8) в великій мірі залежить від розміру обрамлення m. Евристичний кластерний алгоритм розділу великих мереж запропонований А. Санжованні-Винцентеллі [4] дає можливість отримати необхідну кількість діагональних блоків та обрамлення розміром 5-10% від розміру матриці у САПР електронних схем. Для (8) маємо таку залежність прискорення S_p від розміру обрамлення m (див. Рис. 2). Така залежність прискорення зростає для більшої кількості процесорів.

При використанні паралельної віртуальної машини на мережі однопроцесорних комп'ютерів час передачі даних залежить від часу передачі повідомлення мінімального розміру t_{S0} та розміру полоси перепуску даних між процесорами t_0 (див. [5]). Тоді в середовищі розподілених обчислень з використанням локальної мережі на базі мережі Fast Ethernet функція часу передачі повідомленя з достатньою точністю може бути представлена як:

$$f(n,t_{S0},t_S) = t_{S0} + nt_S \qquad (9)$$

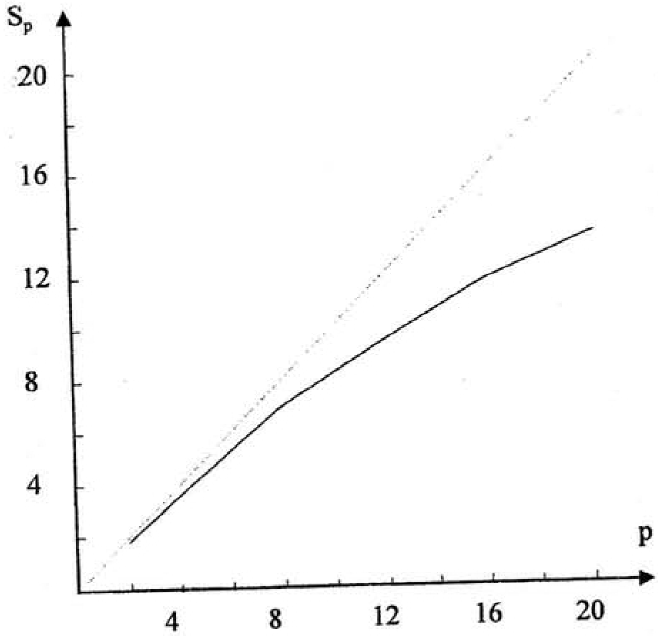

Рис.1

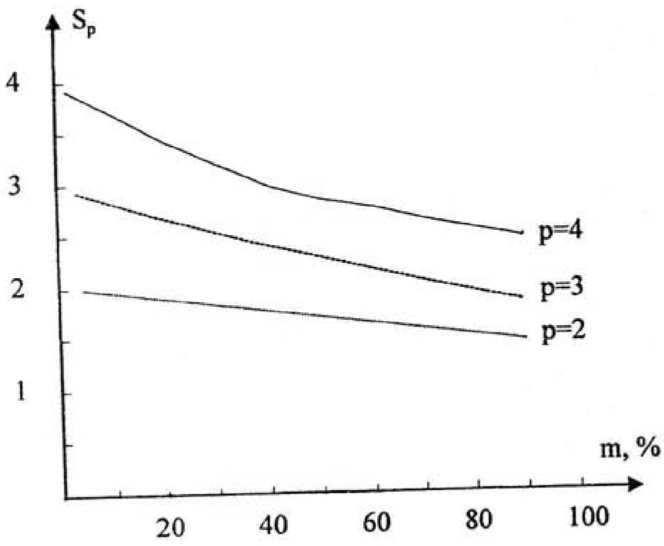

Рис. 2

Конкретні дані часу передачі повідомлення мінімального розміру t_{S0}, розміру полоси перепуску даних між процесорами t_0 для паралельної віртуальної машини на мережі Ethernet та FastEthernet можливо отримати з [5].

Тоді для модифікованого паралельно-послідовного алгоритму блочного LU розкладу необхідно виконати 5 пересилок даних між головним та керованними процесорами. Це можливо виконати за час $S(\omega) = f(ck + cm + ck + m) + f(m^2) + 3 \cdot f(m)$. Для складності модифікованого паралельно-послідовного алгоритму блочного LU-розкладу, отримаємо:

$$T(1) = p[O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k)] + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k) \quad (10)$$

$$T(p) = O(k) + m(O_L(k)+O_U(k)) + m^2(ct_m+(c-1)t_c) + O_L(k) + O_U(k) + ct_m+(c-1)t_c + k(ct_m+(c-1)t_c) + O_L(k) + kt_c + O_U(k) + pm^2 t_c + O(m) + pm\, t_c + O_L(k) + O_U(k)+ f(ck + cm + ck + m) + f(m^2) + 3*f(m) \quad (11)$$

$$S_p = \frac{T(1)}{T(p) + S(\omega)} \quad (12)$$

Для конкретної реалізації модифікованого паралельно-послідовного алгоритму блочного LU розкладу маємо такі значання $t_S = 200;\ t_{S0} = 5000;\ t_m = 10;\ t_c = 1$, а використовуючи як тестову задачу матрицю розміром 1100x1100, що має 5166 ненульових елементів відповідну для схеми 128-ми розрядного суматора, $m = 5\%;\ n = 1100.$

Побудувавши графік прискорення залежно від кількості процесорів p згідно формули (12) отримаємо криву, що зображена на Рис. 3. Як і вважалось вище, на початку кривої маємо зростання прискорення. Та починаючи з деякого $p_{опт}$ отримаємо зменьшення прискорення, так як час на пересилку повідомлень буде значно перевищувати час розрахунків (10). Ці теоретичні результати співпадають з практичними, і для даної задачі для схеми 128-ми розрядного суматора $p_{опт}$ дорівнює 4.

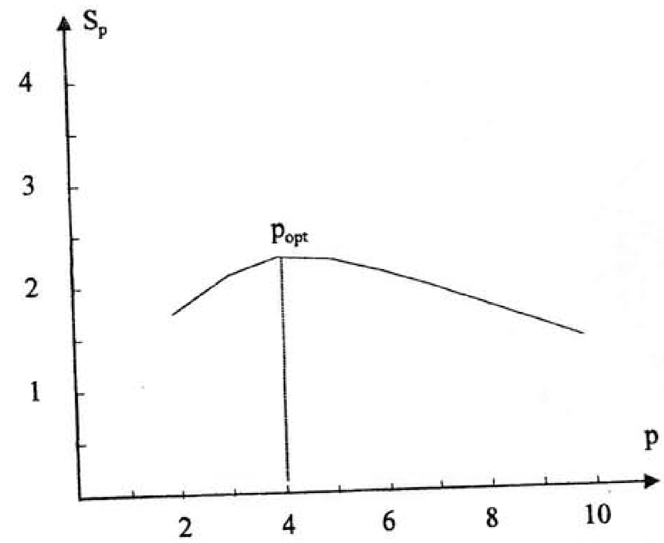

Рис. 3.

З цого зробимо висновок, що для вирішення лінійної системи рівнянь у паралельному середовищі необхідно враховувати: час передічі повідомлення на мережі; достатню кількість процесорів; кількість діагональних блоків; наявний розмір обрамлення;

розрідженість матриць діагональних блоків; складність факторизації матриць; складність обчислення прямого і зворотнього кроків LU метода.

Всі ці параматри можуть бути оптимізовані з метою знаходження максимального прискорення, яке відповідає найменьшому часу обчислення за модифікованим паралельно-послідовним алгоритмом блочного LU розкладу.

Література

1. Молчанов И.Н. "Введение в алгоритмы параллельных вычислений", К. Наукова Думка, 1990, 202 с.
2. Ортега, Дж., "Введение в параллельные и векторные методы решения линейных систем", пер.с англ., М., Мир 1991, 364 с.
3. L. Baker, B.J. Smith, Parallel Programming, McGraw-Hill, 1996, 381 pp.
4. A. Sangiovanni-Vincentelli, L.K. Chen, L.O. Chua, "An Efficient Heuristic Cluster Algorithm for Tearing Large-Scale Networks," IEEE Transactions on circuit and systems, VOL. CAS-24, NO. 12, December 1977, pp.709-717
5. Ладогубец В.В., Харченко К.В., "Средства параллельных вычислений в САПР", Радиоэлектроника №2, 1999, с. 51-60