ТЕОРІЯ ОПТИМАЛЬНИХ РІШЕНЬ

УДК 519.6

О.М. ХІМІЧ, В.А. СИДОРУК

ПЛИТКОВИЙ ГІБРИДНИЙ АЛГОРИТМ ФАКТОРИЗАЦІЇ СТРУКТУРНО-СИМЕТРИЧНИХ МАТРИЦЬ

Запропоновано новий алгоритм отримання розв'язку систем лінійних рівнянь з розрідженими структурно-симетричними додатно-визначеними матрицями на комп'ютерах гібридної архітектури – комп'ютерах з багатоядерними процесорами і графічними прискорювачами.

Вступ. Значна кількість прикладних задач включає у себе як складову частину знаходження розв'язку системи лінійних алгебраїчних рівнянь (СЛАР) з розрідженими матрицями. Характерною особливістю СЛАР, що виникають в тих чи інших задачах є їх великий порядок. У деяких моделях порядок матриці СЛАР може досягати кількох мільйонів.

Разом з тим вимоги до високопродуктивних обчислень набагато випереджають можливості традиційних паралельних комп'ютерів, навіть не зважаючи на багатоядерність процесорів. Розв'язання проблеми прискорення обчислень на комп'ютерах з багатоядерними процесорами розглядається в площині використання для прискорення обчислень гібридних систем на основі багатоядерних CPU і GPU.

За останні роки у світі розроблено чимало алгоритмів розв'язування подібних СЛАР [1]. Серед розроблених методів немає універсального, який би ефективно розв'язував системи з матрицями будь-якого типу розрідженості, але є такі, що призначені для матриць певної структури [2, 3]. Тому при роз-в'язуванні СЛАР є попередня оптимізація структури розрідженої матриці. В даній статті будуть розглядатися розріджені структурно симетричні матриці, тобто матриці, що задовольняють наступним умовам:

- якщо $a_{ij} \neq 0$, то і $a_{ji} \neq 0$,
- якщо $a_{ij} = 0$, то і $a_{ji} = 0$.

Слід зазначити, що рівність значень a_{ij}, a_{ji} не вимагається. Прикладами структурно симетричних матриць можуть бути несиметричні стрічкові матриці, які зустрічаються, наприклад, при моделюванні процесів, що протікають при електрозварюванні та споріднених процесах.

У статті пропонується плиткових гібридний алгоритм факторизації таких матриць, який базується на приведенні матриці до блочно-діагонального вигляду з обрамленням.

Постановка задачі. Розглядається СЛАР з структурно симетричною невиродженою розрідженою матрицею

$$\tilde{A}\tilde{x} = \tilde{b}. \tag{1}$$

Ідеологічною передумовою використання гібридних обчислень при обробці розріджених матриць довільної структури є попереднє перевпорядкування структури вихідної матриці методом паралельних перерізів, що приводить матрицю до блочно-діагонального вигляду з обрамленням.

$$A = P^T \tilde{A} P = \begin{pmatrix} D_{11} & & & 0 & & C_{1p} \\ & D_{22} & & & & C_{2p} \\ & & D_{33} & & & C_{3p} \\ 0 & & & \ddots & & \vdots \\ & & & & D_{p-1,p-1} & C_{p-1,p} \\ C_{p1} & C_{p2} & C_{p3} & \cdots & C_{p,p-1} & D_{pp} \end{pmatrix},$$

$$x = P^T \tilde{x}, b = P^T \tilde{b}$$

де P – матриця перестановок, а блоки D_{ii}, C_{ip} та C_{pi} зберігають розріджену структуру, p – кількість діагональних блоків у матриці, $p \geq 3$.

Природнім для методу паралельних перерізів є те, що розмір останнього діагонального блоку набагато менший, ніж порядки інших діагональних блоків. Також структура блочно-діагональної матриці з обрамленням дозволяє проводити паралельну й незалежну обробку блоків D_{ii}, C_{ip} та C_{pi}, $i = \overline{1, p-1}$.

Одним із ефективних прямих методів для розв'язання системи (1) є метод, що базується на LU розвиненні матриці [4]. Алгоритм розв'язання складається з трьох етапів:

$$A = P_1 L U; \tag{2}$$

$$P_1 L y = P_1 b; \tag{3}$$

$$P_1 U x = y. \tag{4}$$

Тут P_1 – матриця перестановок рядків, яка отримується в процесі факторизації. Розглянемо плитковий гібридний алгоритм факторизації розрідженої структурно-симетричної блочно-діагональної матриці з обрамленням A.

Плитковий алгоритм прямого методу. Розіб'ємо матрицю A на блоки розмірністю $c \times c$.

Далі для факторизації блочно-діагональної матриці застосуємо алгоритм запропонований в [4] для щільних матриць.

Для факторизації матриці на k-му кроці використаємо наступне співвідношення:

$$A^{\kappa} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = P_1 \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{21} \\ 0 & U_{22} \end{pmatrix}, \tag{5}$$

де розміри блоків A_{11} – $c \times c$, A_{12}, A_{21} – (n-kc)c, A_{22} – (n-kc)(n-kc), блоки A_{12}, A_{21} та A_{22} враховують структуру блоків D_{ii}, C_{pi}, C_{ip}, D_{pp}..

Звідси отримаємо алгоритм, за яким здійснюється розвинення на k-му кроці:

$$A_{11} = L_{11}U_{11}; \tag{6}$$

$$L_{21} = \left(U_{11}\right)^{-1} A_{21}; \tag{7}$$

$$U_{12} = \left(L_{11}\right)^{-1} A_{12}; \tag{8}$$

$$\tilde{A}_{22} = A_{22} - L_{21}U_{12}. \tag{9}$$

Зазначимо, що реалізація (6) – (9) на кожному кроці модифікує тільки блоки D_{ii}, C_{pi}, C_{ip}, $i = \overline{1, p-1}$, D_{pp}.

Для декомпозиції даних використовується блочний розподіл даних. У пам'яті кожного CPU(i) та GPU(i) зберігаються блоки D_{ii}, C_{pi}, C_{ip}, $i = \overline{1, p-1}$ та блок $A_{pp}^{(i)}$ того ж розміру, що й D_{pp}. В CPU(p) та GPU(p) зберігається блок D_{pp}.

На кожному кроці у всіх парах CPU(i) та GPU(i) $i = \overline{1, p-1}$ виконуємо:

- у CPU(i), $i = \overline{1, p-1}$ факторизуємо A_{11} із D_{ii}:

$$A_{11} = L_{11} * L_{11}^{T};$$

- у GPU(i), $i = \overline{1, p-1}$ модифікуємо стовпчик блоків L_{21}:

$$L_{21} = \left(U_{11}\right)^{-1} A_{21};$$

- у GPU(i), $i = \overline{1, p-1}$ модифікуємо рядок блоків U_{12}:

$$U_{12} = \left(L_{11}\right)^{-1} A_{12};$$

- у GPU(i), $i = \overline{1, p-1}$ модифікуємо блоки матриці A_{22} з $A_{pp}^{(i)}$ за формулою:

$$\tilde{A}_{22} = A_{22} - L_{21}U_{12}.$$

- У CPU(p), використовуючи мультизбирання, модифікуємо D_{pp}:

$$D_{pp}^{*} = D_{pp} - \sum_{i=1}^{p-1} A_{pp}^{(i)}.$$

Факторизуємо блок D_{pp}^{*}, тим самим завершуючи процес факторизації матриці.

На рисунку показано стан блоків на k-му кроці факторизації на GPU(i).

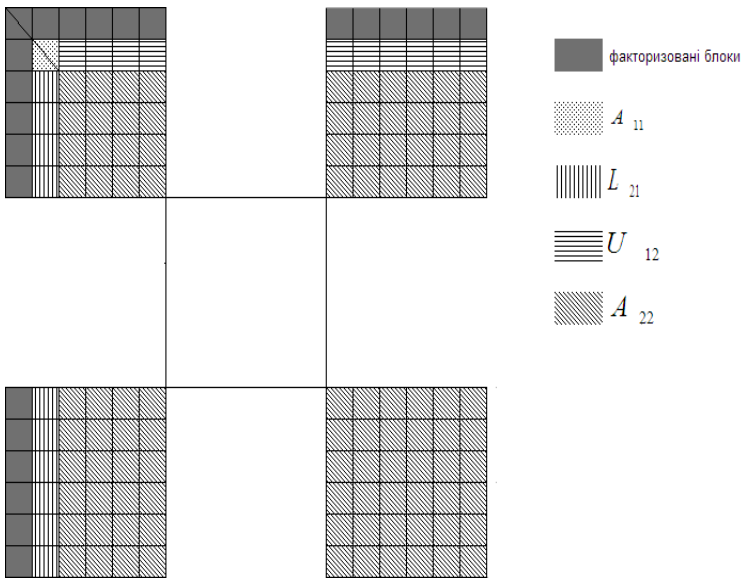

РИСУНОК. Декомпозиція даних на GPU(i)

Для оцінки якості гібридного алгоритму будемо використовувати кое-фіцієнт прискорення S_p та коефіцієнт ефективності E_p, що обчислюються за формулами:

$$S_p = T_1 / T_p, \qquad E_p = S_p / p,$$

де p – кількість використовуваних для розрахунків CPU та GPU, T_1 – час розв'язування задачі на гібридному комп'ютері з одним CPU та одним GPU, T_p – час розв'язування тієї ж задачі на гібридному комп'ютері з використанням p CPU і p GPU.

$$T_1 = \frac{Nt_g}{n_0},\ T_p = \frac{Nt_g}{n_0} + M_1 t_{opp} + M_2 t_{opg} + Q_1 t_{cpp} + Q_2 t_{cpg},$$

де t_g – середній час виконання однієї арифметичної операції на GPU; n_o – кількість арифметичних операцій, які можуть бути виконані одночасно на GPU; t_{opp} – час,
необхідний для обміну одним машинним словом між двома CPU; t_{opg} – час обміну між одним CPU та GPU; t_{cpp} – час, який потрібний для встановлення зв'язку між двома процесами; t_{cpg} – час, який потрібний для встановлення зв'язку між CPU і GPU; M_i, Q_i – кількість відповідних обмінів та синхронізацій; N – кількість виконуваних алгоритмом арифметичних операцій.

Оцінки коефіцієнта прискорення гібридного алгоритму отримані для топології комунікаційних зв'язків «кільце».

Будемо вважати, що порядки всіх діагональних блоків приблизно рівні

$$q_i \approx q = \frac{n-s}{p-1},$$

де s – порядок останнього діагонального блоку. Тоді справедливі наступні теореми.

Теорема 1. Кількість операцій, що виконуються на 1 GPU для знаходження факторизації структурно-симетричної матриці оцінюється величиною

$$N_p \approx \frac{2q^3}{3} + 2q^2 s = \frac{2q^2}{3}(q+3s).$$

Доведення. Введемо наступне позначення: $l = \frac{q}{c}$ – кількість рядків плиток у діагональному блоці.

Оскільки (6) – (9) виконуються паралельно і незалежно у всіх $p-1$ парах CPU та GPU і максимальна кількість операцій припадає на етап (9), то оцінка кількості операцій виконуваних на одному GPU визначається саме складністю етапу (9).

Кількість операцій необхідних для виконання (9) можна оцінити величиною

$$N_p \approx 2c\left(\sum_{k=1}^{l}(q-k)^2 + \sum_{k=1}^{l}2(q-kc)s + \sum_{k=1}^{l}s^2\right). \qquad (10)$$

Внаслідок нескладних перетворень (10) можна записати наступним чином:

$$N_p \approx 2c\left(\frac{q^3}{3c}+\frac{q^2 s}{c}+\frac{q^2}{2}+qs+\frac{qc}{6}+\frac{qs^2}{c}\right). \qquad (11)$$

Оскільки порядок останнього діагонального невеликий і порядок плитки досить малий відповідними доданками в (11) можна нехтувати. Розкривши дужки і знехтувавши відповідними доданками з (11) отримуємо

$$N_p \approx \frac{2q^3}{3}+2q^2 s=\frac{2q^2}{3}(q+3s).$$

Теорема доведена.

Теорема 2. Прискорення дрібно-плиткового гібридного алгоритму *LU*-розвинення структурно-симетричної матриці з обрамленням *A*, становить

$$S_p \approx (p-1)\left(1+\frac{3(p-1)s^2}{4q^2(q+3s)}\left(\tau_{opp}+\left(\frac{2}{(p-1)}+\frac{4qcs^2}{(p-1)}+\frac{4c}{(p-1)s}\right)\tau_{opg}\right)\right)^{-1},$$

де *c* – розмір плитки.

Доведення. Знайдемо об'єм даних, що передаються між процесорами підчас виконання алгоритму. Оскільки, використовується топологія комунікаційних зв'язків «кільце» і процесори обмінюються s^2 машинними словами – сумарний об'єм даних, яким обмінюються процесори становить $\frac{(p-1)s^2}{2}$.

Перед виконанням операції мультизбирання необхідно у всіх парах CPU та GPU, крім останньої, виконати обмін даними об'ємом s^2. Також аналогічний обмін проводить і в останній парі CPU та GPU.

Також у процесі факторизації діагональних блоків D_{ii} всі пари CPU і GPU, крім останньої, обмінюються даними об'ємом $2qc$. Остання пара CPU і GPU обмінюється об'ємом даних рівним $2sc$.

Сумарний об'єм даних, яким обмінюються всі пари CPU та GPU рівний $2qc+s^2+2sc$.

При обчисленні T_1 та T_p будемо використовувати величину N_p з теореми 1.

Підставимо всі знайдені величини у формулу для обчислення коефіцієнта прискорення S_p отримаємо

$$S_p \approx \frac{(p-1)\frac{N_p}{n_o}t_g}{\frac{N_p}{n_o}t_g + \frac{(p-1)s^2}{2}t_{opp} + (2qc + s^2 + 2sc)t_{opg}}.$$

Розділивши чисельник та знаменник на $\frac{N_p}{n_o}t_g$ отримаємо

$$S_p \approx \frac{(p-1)}{1 + \frac{1}{N_p}\left(\frac{(p-1)s^2}{2}\tau_{opp} + (2qc + s^2 + 2sc)\tau_{opg}\right)}.$$

Винесемо за дужки $\frac{(p-1)s^2}{2}$ та підставимо значення N_p

$$S_p \approx \frac{(p-1)}{1 + \frac{3(p-1)s^2}{4q^2(q+3s)}\left(\tau_{opp} + \left(\frac{2}{(p-1)} + \frac{4qcs^2}{(p-1)} + \frac{4c}{(p-1)s}\right)\tau_{opg}\right)}$$

або

$$S_p \approx (p-1)\left(1 + \frac{3(p-1)s^2}{4q^2(q+3s)}\left(\tau_{opp} + \left(\frac{2}{(p-1)} + \frac{4qcs^2}{(p-1)} + \frac{4c}{(p-1)s}\right)\tau_{opg}\right)\right)^{-1}.$$

Теорема доведена.

Висновки. Досліджено гібридний алгоритм прямого методу розв'язання систем лінійних рівнянь з структурно-симетричними матрицями. В основі алгоритму лежить ідея приведення матриці до блочно-діагонального вигляду з обрамленням. Дана операція дозволяє досягти високого рівня паралелізму. Плитковість дозволяє краще враховувати профіль матриці й ущільнити обчислення на GPU.

Запропонований підхід дозволяє розв'язувати велику кількість прикладних задач. Зокрема, можливе застосування даного алгоритму у розв'язанні задач, що виникають при моделюванні процесів електрозварювання та споріднених технологій.

А.Н. Химич, В.А. Сидорук

ПЛИТОЧНЫЙ ГИБРИДНЫЙ АЛГОРИТМ ФАКТОРИЗАЦИИ СТРУКТУРНО-СИММЕТРИЧНЫХ МАТРИЦ

Предложен новый алгоритм получения решения систем линейных уравнений с разреженными структурно-симметричными положительно-определенными матрицами на компьютерах гибридной архитектуры – компьютерах с многоядерными процессорами и графическими ускорителями.

A.N. Khimich, V.A. Sydoruk

TILED HYBRID ALGORITHM FOR FACTORIZATION OF STRUCTURALLY SYMMETRIC MATRICES

Proposed a new algorithm to obtain the solution of linear equations with sparse structurally symmetric positively-defined matrix for hybrid architecture computers - computers with multicore processors and graphics accelerators.

1. *Джордж А., Лю Дж.* Численное решение больших разреженных систем уравнений. М.: Мир, 1984. 334 с.
2. *Химич А.Н., Попов А.В., Полянко В.В.* Алгоритмы параллельных вычислений для задач линейной алгебры с матрицами нерегулярной структуры. *Кибернетика и системный анализ.* 2011. 47, № 6. С. 159 – 174.
3. *Хіміч О.М., Сидорук В.А.* Гібридний алгоритм розв'язування лінійних систем з розрідженими матрицями на основі блочного LL^T методу. *Компьютерная математика*. 2015. Вип. 1. С. 67 – 74.
4. *Alfredo Buttari, Julien Langou, Jakub Kurzak and Jack Dongarra*. A Class of Parallel Tiled Linear Algebra Algorithms for Multicore Architectures. Parallel Computing. 2009. Vol. 35, Issue 1. P. 38 – 53.

Одержано 20.03.2017