КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Оптимизация вычислений

Розглядається новий паралельний алгоритм розв'язування систем лінійних алгебраїчних рівнянь з розрідженими симетричними додатно визначеними матрицями на комп'ютерах з процесором Intel Xeon Phi другого покоління. Подано результати апробації алгоритму на комп'ютері Інпарком-хр.

УДК 519.6

В.А. СИДОРУК

ПАРАЛЕЛЬНИЙ АЛГОРИТМ РОЗВ'ЯЗУВАННЯ ЛІНІЙНИХ СИСТЕМ З РОЗРІДЖЕНИМИ МАТРИЦЯМИ ПОПЕРЕМІННО ТРИКУТНИМ МЕТОДОМ

Вступ. Інтерес до проблеми побудови ефективних методів розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) з розрідженими матрицями переважно обумовлений їх численними застосуваннями. Зокрема, системи рівнянь з розрідженими матрицями виникають у задачах аналізу міцності конструкцій у цивільному та промисловому будівництві, авіабудуванні, суднобудуванні тощо. Область застосування методів розв'язування СЛАР з розрідженими матрицями постійно розширюється.

Разом з тим вимоги до високопродуктивних обчислень набагато випереджають можливості традиційних паралельних комп'ютерів. Найбільш продуктивними на даний час є комп'ютери з гібридною архітектурою (поєднання багатоядерних процесорів (CPU), графічних прискорювачів (GPU)) та системи з процесорами Intel Xeon Phi.

Одним із напрямків розв'язування лінійних систем з розрідженими матрицями є ітераційні процеси на основі трикутних методів [1]: метод Зейделя, метод верхньої релаксації, метод симетричної верхньої релаксації, попеременно-трикутний метод. В роботі розглядається паралельний алгоритм попеременно трикутного методу для комп'ютерів з процесорами Intel Xeon Phi другого покоління. Формфактори для цього кристалу можуть бути двох видів: центральний процесор та сопроцесор (графічний прискорювач).

У даній статті розглядаються СЛАР з симетричними додатно визначеними розрідженими матрицями блочно-діагонального виду з обрамленням. Такі матриці можна отримати, застосувавши метод паралельних перерізів [2, 3].

1. Постановка задачі

Розглянемо задачу

$$Ax = b \tag{1}$$

з симетричною додатно-визначеною розрідженою матрицею порядку n.

$$A = \begin{pmatrix} A_{11} & 0 & 0 & \cdots & 0 & A_{1p} \\ 0 & A_{22} & 0 & \cdots & 0 & A_{2p} \\ 0 & 0 & A_{33} & \cdots & 0 & A_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{p-1,p-1} & A_{p-1,p} \\ A_{p1} & A_{p2} & A_{p3} & \cdots & A_{p,p-1} & A_{pp} \end{pmatrix},$$

де блоки A_{ii}, A_{ip}, A_{pi} зберігають розріджену структуру.

Для розв'язання системи (1) застосуємо неявний однокроковий метод [1].

$$B\frac{x_{k+1} - x_k}{\tau_k} + Ax_k = b. \tag{2}$$

Оператор B вибирається з умови мінімізації кількості ітерацій і його економічності. До економічних методів можна віднести поперемінно-трикутний метод, що описаний в [1].

Далі опишемо блочний варіант пропонованого алгоритму.

Блочний алгоритм. Розглянемо блочний алгоритм поперемінно трикутного методу. Для цього матрицю $\tilde{A}$ будемо представляти як

$$\tilde{A} = R + R^T.$$

Вимоги економічності задовольняє поперемінно-трикутний метод (ПТМ), суть якого полягає у побудові оператора B у спеціальному факторизованому вигляді

$$B = (E + \omega R)(E + \omega R), \tag{3}$$

де ω – ітераційний параметр, що буде визначений далі, R – оператор, якому відповідає трикутна матриця.

$$R = \begin{pmatrix} R_{11} & 0 & 0 & \cdots & 0 & 0 \\ 0 & R_{22} & 0 & \cdots & 0 & 0 \\ 0 & 0 & R_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & R_{p-1,p-1} & 0 \\ R_{p1} & R_{p2} & R_{p3} & \cdots & R_{p,p-1} & R_{pp} \end{pmatrix}. \tag{4}$$

Даний оператор зберігає блочну діагональну структуру $\tilde{A}$.

ПТМ належить до класу методів, ітераційні параметри для яких вибираються з урахуванням апріорної інформації про оператори ітераційної схеми. Для ПТМ ця інформація полягає у заданні величин δ та Δ, що задовольняють нерівності

$$\|x\|_A^2 \geq \delta\|x\|^2 \sqrt{b^2-4ac} \lim_{n\to\infty}\left\{\sup_{\|s-s\|\gamma} |G\right\},\ \left\|R^T x\right\|^2 \leq \Delta\|x\|_R^2 \Big/ 4.$$

Тоді $\omega = \dfrac{2}{\sqrt{\delta\Delta}}$.

Необхідно зазначити, що в загальному випадку отримання теоретично обґрунтованих виразів для констант δ та Δ є самостійною складною задачею. Тому викликають інтерес алгоритми, які потребують мінімум апріорної інформації про задачу.

Пропонується нестаціонарна схема [1, с. 260]

$$B_{k+1}\frac{x_{k+1}-x_k}{\tau_k} + Ax_k = b. \tag{5}$$

Як і раніше B_{k+1} буде вибиратись у спеціальному факторизованому вигляді [1, с. 264]

$$B_{k+1} = \left(E+\omega_{k+1}R\right)\left(E+\omega_{k+1}R^T\right).$$

Для обчислення ω_{k+1} використаємо наступну процедуру:

$$\omega_{k+1} = \|w_k\| / \left\|R^T w_k\right\|,\ \ \omega_0 = 0.$$

Ітераційний процес будемо реалізовувати за наступними формулами:

$$r_k = Ax_k - b. \tag{6}$$

$$\left(E+\omega_k R\right)\overline{w}_k = r_k. \tag{7}$$

$$\left(E+\omega_k R^T\right)w_k = \overline{w}_k. \tag{8}$$

$$x_{k+1} = x_k - \tau_{k+1}w_k. \tag{9}$$

Вибір ітераційного параметра τ_{k+1} виконується за формулою методу скорішого спуску.

$$\tau_{k+1} = \frac{\left(w_k, r_k\right)}{\left(Aw_k, w_k\right)}. \tag{10}$$

У подальшій викладці будемо вважати, що матриця E та вектори b, x_k, x_{k+1}, w_k, $\overline{w}_k$, r_k теж мають блочну структуру, а величина i блоку відповідає порядку блоку R_{ii}. Позначення:

- $b_i, x_{k_i}, x_{k+i_i}, w_{k_i}, \overline{w_{k_i}}, r_{k_i},$ означатимуть, що ми працюємо з i частиною відповідних векторів;
- $b_p, x_{k_p}, x_{k+i_p}, w_{k_p}, \overline{w_{k_p}}, r_{k_p},$ означатимуть, що ми працюємо з частинами векторів, що відповідають останньому процесору.

Паралельний алгоритм. Припустимо, що для обчислень нам доступна система архітектури *p* CPU. Також будемо вважати, що пам’яті *p* CPU нам достатньо для збереження необхідних даних.

Реалізуємо наступну декомпозицію даних:

- у пам’яті CPU з номерами *i*, $1 \le i < p$ містяться блоки R_{ii}, R_{ip}, а також частини векторів r_{k_i}, x_{k_i}, x_{k+1_i}, x_{k_p}, b_i, $\overline{w}_{k_i}$, w_{k_i}, $w_{k\,p}$, y_i, z_i;
- у пам’яті CPU з номером *p*, містяться блок R_{pp}, а також частини векторів r_{k_p}, x_{k_p}, x_{k+1_p}, b_p, $\overline{w}_{k\,p}$, $w_{k\,p}$, y_p, z_p.

Запишемо паралельний гібридний алгоритм ПТМ.

Етап (6) реалізується співвідношеннями:

$$r_{k_i} = R_{ii}x_{k_i} + R_{ii}^T x_{k_i} + R_{pi}^T x_{k_p} - b_i, \tag{11}$$

для всіх CPU з номером *i* $1 \le i < p$,

$$r_{k_p} = \sum_{i=1}^{p-1} R_{pi}x_{k_i} + R_{pp}x_{kp} + R_{pp}^T x_{k_p} - b_p, \tag{12}$$

для CPU з номером *p*.

Формула (7) реалізується співвідношеннями:

$$\overline{w}_{k} = \left(E_{ii} + \omega_k R_{ii}\right)^{-1} r_{k_i}, \tag{13}$$

для всіх CPU з номером *i* $1 \le i < p$,

$$\overline{w}_{kp} = \left(E_{pp} + \omega_k R_{pp}\right)^{-1} \left(r_{k_p} - \sum_{i=1}^{p-1} \omega_k R_{pi} \overline{w}_{k} \right), \tag{14}$$

для CPU з номером *p*.

Для формули (8) діють співвідношення:

$$w_{k\,p} = \left(E_{pp} + \omega_k R_{pp}^T\right)^{-1} \overline{w}_{kp}, \tag{15}$$

для CPU з номером *p*,

$$w_{k_i} = \left(E_{ii} + \omega_k R_{ii}^T\right)^{-1} \left(\overline{w}_{k} - \omega_k R_{pi}^T w_{k\,p}\right), \; 1 \le i < p, \tag{16}$$

для всіх CPU з номером *i* $1 \le i < p$.

Запишемо блочний алгоритм для (10).

Введемо два вектори *z*, *y*:

$$y = \left(w_k, r_k\right),$$

$$z = \left(Aw_k, w_k\right).$$

Елементи векторів обчислюються наступним чином:

- у всіх процесорах виконується співвідношення

$$y_i = \left(w_{k\,i}, r_{k\,i}\right),\ \ i = \overline{1, 2, ..., p}; \qquad 17)$$

- у всіх CPU з номерами $i\ \ 1 \le i < p,$ незалежно виконуємо

$$z_i = \left(\left(R_{ii} w_{k\,i} + R_{ii}^T w_{k\,i} + R_{pi}^T w_{k\,p}\right),\ w_{k\,i}\right) \qquad (18)$$

та знаходимо добутки $R_{pi} w_{k_i}$;

- у останьому CPU виконується мультизбирання та формула приведена далі

$$z_p = \left(\left(\sum_{i=1}^{p-1} R_{pi} w_{k_i} + R_{pp} w_{k_p} + R_{pp}^T w_{k_p}\right), w_{k_p}\right). \qquad (19)$$

У одному з процесорів виконуємо мультизбирання і модифікацію параметра τ_{k+1}

$$\tau_{k+1} = \sum_{i=1}^{p} y_i \Big/ \sum_{i=1}^{p} z_i\,. \qquad (20)$$

Виконуємо мультирозсилку τ_{k+1} та у всіх процесорах реалізуємо

$$x_{k+1i} = x_{k_i} - \tau_{k+1} w_{k\,i},\ \ 1 \le i < p. \qquad (21)$$

Кількість операцій, виконуваних одним CPU, при реалізації паралельного алгоритму поперемінно трикутного методу для розрідженої матриці блочно-діагональної структури з обрамленням оцінюється величиною

$$N_p \approx \max_{1 < i \le p-1} \alpha_i,$$

де $a_i = 12nz\left(\tilde{D}_i\right) + 12nz\left(C_{pi}\right) + 3n_i,$ $nz()$ – кількість ненульових елементів в матриці, n_i – довжина і частини вектора, що відповідає діагональному блоку з номером i.

Для паралельного алгоритму поперемінно-трикутного методу коефіцієнт прискорення оцінюється наступним співвідношенням:

$$S_p \approx (p-1)\left(\left(\frac{1}{\beta}\right)\left(\max_{1 \le i < p-1} \alpha_i t_c + 2(p-1)(n_p t_{opp} + t_{cpp})\right)\right)^{-1},$$

де

$$a_i = 12nz\left(\tilde{D}_i\right) + 12nz\left(C_{pi}\right) + 3n_i,\ \ \beta = \frac{\sum_{i=1}^{p-1} \alpha_i}{\left(p-1\right)},$$

де t_c – середній час виконання однієї арифметичної операції на CPU, t_{opp} – час, необхідний для обміну одним машинним словом між двома процесами, t_{cpp} – час, який потрібний для встановлення зв'язку між двома процесами.

Програмна реалізація і чисельний експеримент.

Розрахунки проводились на комп'ютері Інпарком_*xp* [4], який має наступні характеристики:

- процесори: Intel Xeon Phi 7210 (64 ядра) з частотою 1.3 ГГц;
- обсяг швидкої пам'яті MCDRAM: 16 Гб;
- обсяг оперативної пам'яті: 192 Гб;
- обсяг SSD накопичувача: 240 Гб.

При програмній реалізації паралельного алгоритму попеременінно трикутного методу використовувались функції з бібліотек Intel MPI [5], OpenMP 4.0 [6], а також функції з бібліотеки Intel Math Kernel Library [7], що реалізують BLAS операції над розрідженими матрицями. Для роботи з швидкою пам'яттю MCDRAM використовуються функції з бібліотеки MemKind [8].

Чисельні експерименти проводились на розріджених матрицях, що наведені в таблиці. Також у таблиці наведені характеристики матриці такі як: порядок матриці, кількість ненульових елементів і проблемна область.

ТАБЛИЦЯ. Набір тестових матриць з Флоридської колекції розріджених матриць

Назва	Проблемна область	Порядок	Кількість ненульових елементів
G3_circuit	circuit simulation problem	1 585 478	7 660 826
G2_circuit	circuit simulation problem	150 102	726 624
parabolic_fem	computational fluid dynamics problem	525 825	3 674 625
apache2	structural problem	715 176	4 817 870
Dubcova3	2D/3D Problem	146 689	3 636 643

На рис. 1 показано портрети наповненості ненульовими елементами тестових матриць. Графіки зміни часів розв'язання систем рівнянь з використанням різної кількості процесорів показано на рис. 2 і 3. На рис. 3 програмна реалізація алгоритму реалізована з підтримкою роботи з надшвидкою пам'яттю MCDRAM.

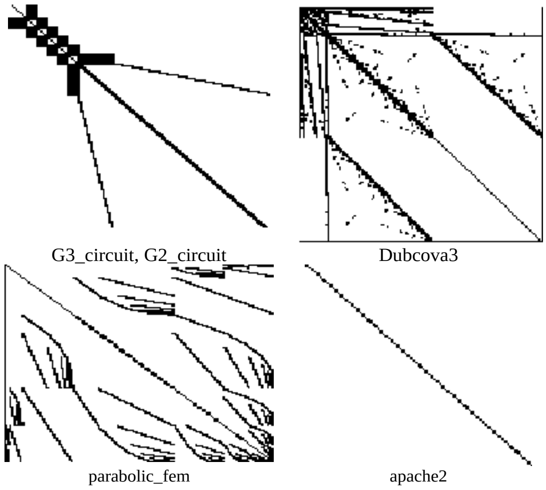

РИС. 1. Профіль наповненості ненульовими елементами вихідних матриць

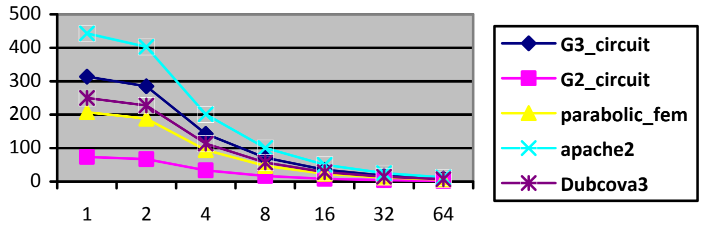

РИС. 2. Час розрахунків на різній кількості ядер

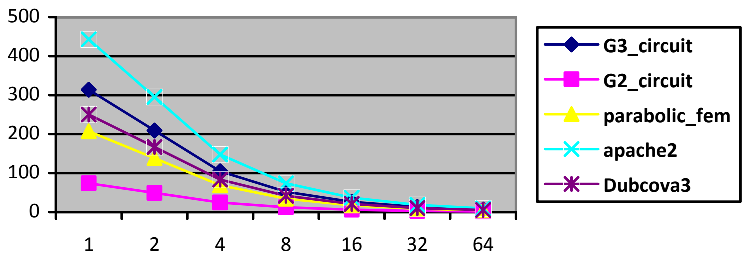

РИС. 3. Час розрахунків на різній кількості ядер з використанням MCDRAM

Висновки. Розроблено та експериментально досліджено паралельний алгоритм поперемінно трикутного методу розв’язування систем лінійних алгебраїчних рівнянь з розрідженими матрицями нерегулярної структури на комп’ютерах з процесорами Intel Xeon Phi другого покоління. Основою пропонованого підходу є структурна регуляризація матриць (перевпорядкування елементів матриці до блочно-діагонального вигляду з обрамленням) та застосування відомих ітераційних процедур на основі трикутних методів.

Показано, що виконання структурної регуляризації, використання на всіх етапах обчислень функцій з ефективних програмних бібліотек, а також урахування особливостей архітектури комп'ютера дають можливість отримати суттєве прискорення.

В.А. Сидорук

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЛИНЕЙНЫХ СИСТЕМ С РАЗРЕЖЕННЫМИ МАТРИЦАМИ ПОПЕРЕМЕННО-ТРЕУГОЛЬНЫМ МЕТОДОМ

Рассматривается новый параллельный алгоритм решения систем линейных алгебраических уравнений с разреженными симметричными додатно определенными матрицами на компьютерах с процессором Intel Xeon Phi второго поколения. Представлены результаты апробации алгоритма на компьютере Инпарком-хр.

V.A. Sydoruk

PARALLEL ALGORITHM FOR SOLVING LINEAR SYSTEMS WITH SPARSE MATRIX BY ALTERNATELY-TRIANGULAR METHOD

A new parallel algorithm for solving systems of linear algebraic equations with sparse symmetric positively-defined matrices on computers with Intel Xeon Phi processor of the second generation is considered. The results of testing the algorithm on the Inparcom-xp computer are presented.

Список літератури

1. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. М.: Наука, 1978. 592 с.
2. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. М.: Мир, 1984. 334 с.
3. Писанецки С. Технология разреженных матриц. М.: Мир, 1988.
4. Химич А.Н., Молчанов И.Н., Мова В.И. и др. Численное программное обеспечение MIMD-компьютера Инпарком. Киев: Наукова думка, 2007. 222 с.
5. https://software.intel.com/en-us/intel-mpi-library
6. https://www.openmp.org/
7. https://software.intel.com/en-us/mkl
8. https://github.com/memkind/memkind

Одержано 07.06.2018

Про автора:

Сидорук Володимир Антонович,
кандидат фізико-математичних наук, науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України.