КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 519.6

О.М. ХІМІЧ, В.А. СИДОРУК

ПЛИТКОВИЙ ГІБРИДНИЙ АЛГОРИТМ ФАКТОРИЗАЦІЇ РОЗРІДЖЕНИХ БЛОЧНО-ДІАГОНАЛЬНИХ МАТРИЦЬ З ОБРАМЛЕННЯМ

Розглядається новий гібридний алгоритм розв'язування систем лінійних алгебраїчних рівнянь з розрідженими симетричними додатно визначеними матрицями на комп'ютерах з графічними прискорювачами. Подано результати апробації алгоритму на багатоядерному комп'ютері з графічними прискорювачами Інпарком.

Вступ. При чисельному розв'язуванні науково-технічних задач у багатьох випадках виникає необхідність вирішувати задачу (або декілька задач) лінійної алгебри. Причому, як правило, розв'язування задач лінійної алгебри займає значну частину (50 % і більше) часу розв'язування всієї задачі загалом. Наприклад, задачі лінійної алгебри виникають при дискретизації крайових задач або задач на власні значення проекційно-різницевим методом (скінченних різниць, скінченних елементів).

Важливою особливістю задач лінійної алгебри, що виникають при дискретизації, є високий порядок їх матриць – до десятків мільйонів. Це спричинено бажанням оперувати більш точними дискретними моделями, що дозволяють отримувати наближені розв'язки більш близькі до розв'язків вихідних задач, краще враховувати локальні особливості даного процесу або явища. Також характерною особливістю задач лінійної алгебри, які виникають при дискретизації, являється те, що кількість ненульових елементів матриць таких задач складає kn, де $k << n$, а n – порядок матриці, тобто матриці є розрідженими [1].

Структура розрідженої матриці визначається нумерацією невідомих задачі і часто є стрічковою, блочно-діагональною з обрамленям, профільною і тому подібне.

У даній статті розглядаються симетричні додатно визначені розріджені матриці блочно-діагонального вигляду з обрамленням.

Постановка задачі. Розглянемо задачу знаходження розв'язку системи лінійних алгебраїчних рівнянь

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

з симетричною додатно-визначеною розрідженою матрицею порядку n.

Теоретичною передумовою методу розв'язання задачі (1) для розріджених матриць на комп'ютерах гібридної архітектури є попереднє застосування до вихідної матриці методу паралельних перерізів, який приводить вихідну матрицю до блочно-діагонального вигляду з обрамленням [1, 2]

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

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

Таким чином, задача розв'язування вихідної задачі (1) зводиться до розв'язування еквівалентної задачі

$$\tilde{A}\tilde{x} = \tilde{b}, \tag{2}$$

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

Найбільш ефективним прямим методом розв'язання такої задачі є, як відомо, метод Холецького [1, 2]. Розв'язання системи (2) полягає у розв'язанні підзадач: трикутне розвинення матриці, розв'язання двох СЛАР з трикутними матрицями

$$\tilde{A} = \tilde{L} * \tilde{L}^T, \tag{3}$$

$$\tilde{L}y = \tilde{b}, \tag{4}$$

$$\tilde{L}^T \tilde{x} = y. \tag{5}$$

Оскільки в прямому методі розв'язання СЛАР найбільшу обчислювальну складність має етап (3), то розробка ефективних алгоритмів факторизації є актуальною проблемою. В роботі [3] описано паралельний гібридний алгоритм розв'язання системи (1) на основі гібридного алгоритму факторизації Холецького розрідженої блочно-діагональної матриці з обрамленням. В даній роботі представлено гібридний плитковий алгоритм факторизації, який дозволяє максимально використати профільну або розріджену структуру блоків D_{ii}, C_{ip}, C_{pi}, D_{pp}.

Гібридний алгоритм. Розіб'ємо матрицю A на блоки розмірністю $s\times s$. Далі для факторизації блочно-діагональної матриці застосуємо алгоритм запропонований в [4] для щільних матриць.

Для факторизації матриці на *к*-му кроці використаємо наступне співвідношення

$$A^{\kappa} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{pmatrix}, \tag{6}$$

де розмірності блоків A_{11} – $s \times s$, A_{12} – $(n - ks)s$, A_{22} – $(n - ks)(n - ks)$, блоки A_{12} та A_{22} враховують структуру діагональних блоків та блоків обрамлення.

Звідси отримаємо алгоритм, за яким проводиться розвинення на *к* кроці:

$$A_{11} = L_{11} * L_{11}^T; \tag{7}$$

$$L_{21} = A_{21} * \left(L_{11}^T\right)^{-1}; \tag{8}$$

$$\tilde{A}_{22} = A_{22} - L_{21} * L_{21}^T. \tag{9}$$

Зазначимо, що реалізація (7) – (9) на кожному кроці модифікує тільки блоки D_{ii}, C_{pi}, $i = \overline{1, p-1}$, D_{pp}.

Нехай для розв'язування задачі на комп'ютері гібридної архітектури маємо *p* CPU і *p* GPU. Реалізуємо наступний розподіл даних: в пам'яті CPU(*i*) та GPU(*i*) $i = \overline{1, p-1}$ містяться блоки D_{ii}, C_{pi} та блок $A_{pp}^{(i)}$ того ж розміру, що й D_{pp}; в CPU(*p*) та GPU(*p*) – блок D_{pp}. Тут і надалі, під CPU(*i*) та GPU(*i*) розумітимемо CPU та GPU, що відповідають процесу з номером *i*, $i = \overline{1, p}$.

На рис. 1 показано блочний розподіл даних на *к*-му кроці факторизації блочно-діагональної матриці з обрамленням, враховуючи запропоновану вище декомпозицію.

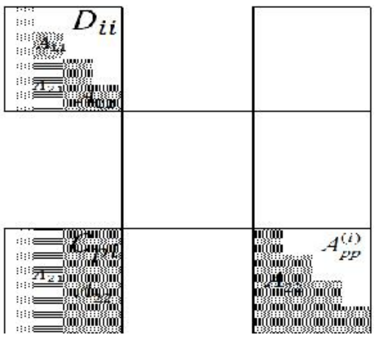


РИС. 1. Декомпозиція даних в GPU(*i*) на *к*-му кроці факторизації

Використовуючи декомпозицію даних, наведену вище, плитковий гібридний алгоритм факторизації записується у наступній формі:

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

- 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}

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

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

$$\tilde{A}_{22}=A_{22}-L_{21}L_{21}^{T};$$

2) використовуючи мультизбирання (одночасна пересилка даних і знаходження сум), модифікуємо D_{pp}

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

3) факторизуємо блок D_{pp}^{*}, тим самим завершуючи процес факторизації матриці A.

Оцінки прискорення і ефективності алгоритму. Для оцінки якості гібридних алгоритмів будемо використовувати коефіцієнт прискорення S_p та коефіцієнт ефективності E_p, що обчислюються за формулами

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

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.

Введемо наступні позначення: N_p – кількість операцій, виконуваних алгоритмом на гібридному комп'ютері з використанням p CPU і p GPU; t_g – середній час виконання однієї арифметичної операції 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 CPU; t_{cpg} – час, який потрібний для встановлення зв'язку між CPU і GPU.

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

Обчислимо кількість операцій виконуваних алгоритмом факторизації. Введемо наступні позначення $m=\frac{n}{p}$ – порядок діагонального блоку матриці A та $l=\frac{m}{s}$ – кількість плиток у стовпці діагонально блоку. При підрахунку кількості операцій будемо вважати, що матриця A має діагональні блоки однакових порядків.

Оскільки в алгоритмі на $p-1$ парі CPU та GPU одночасно і незалежно виконуються l раз етапи (7) – (9) і один раз факторизується останній діагональний блок, то кількість арифметичних операцій виконуваних операцій можна обчислити за наступною формулою:

$$N_1 \approx (p-1)\alpha + \beta.$$

Для обчислення кількості операції необхідних для факторизації останнього діагонального блоку будемо вважати, що останній діагональний блок щільна матриця. Тоді $\beta = \frac{m^3}{3}$.

Розглянемо етапи (7) – (9). Тут найбільше арифметичних операцій припадає на виконання етапу (9). Обчислимо їх кількість.

На к-му кроці факторизації, як видно з рис. 1, нам потрібно модифікувати $\left\lceil \frac{(m-ks)^2}{2} \right\rceil$ елементів з D_{ii}, $(m-ks)m$ – з C_{pi} та $\left\lceil \frac{m^2}{2} \right\rceil$ – з $A_{pp}^{(i)}$.

Тобто

$$\sim \approx 2s\left(\sum_1^l \frac{(m-ks)^2}{2} + \sum_1^l (m-ks)m + \sum_1^l \frac{m^2}{2} \right).$$

Розкриємо дужки та розпишемо суми

$$\alpha \approx \left(s\sum_1^l \left(m^2 - 2mks + 2k^2s^2\right) + 2s\sum_1^l (m-ks)m + s\sum_1^l m^2 \right).$$

Після проведення спрощень і підстановки значень отримаємо наступну формулу для обчислення α:

$$\alpha \approx \frac{8m^3}{3}.$$

Звідси

$$N_1 \approx (p-1)\frac{8m^3}{3} + \frac{m^3}{3}.$$

Кількість операцій виконуваних при реалізації паралельного варіанту гібридного алгоритму обчислюється за формулою

$$N_p \approx 3m^3.$$

Знехтувавши доданками, що відповідають за синхронізації між процесорами та процесорами і графічними прискорювачами, прискорення дрібно-плиткового гібридного алгоритму LL^T-розвинення розрідженої блочно-діагональної матриці з обрамленням A, становить

$$S_p \approx \frac{\left((p-1)\frac{8m^3}{3} + \frac{m^3}{3}\right)t_g}{3m^3 t_g + \mathrm{T}},$$

де $T = \frac{(p-1)m^2}{2} t_{opp} + (4ms + m^2)t_{opg}$.

Результати чисельних експериментів. Розрахунки проводились на вузлах кластеру Інпарком-G [5], які мають наступні характеристики:

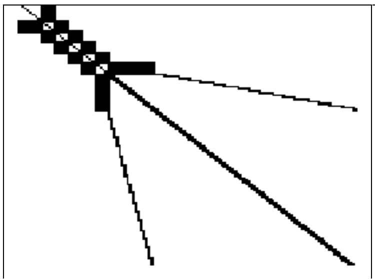
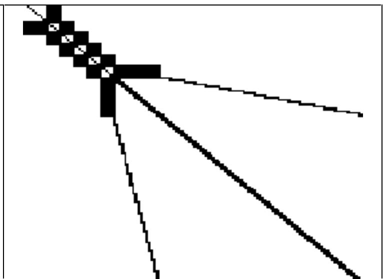
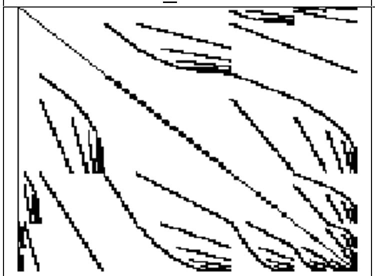
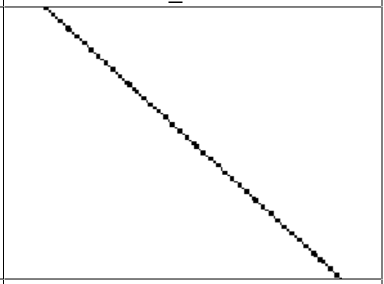
- процесори: 2 Xeon 5606 (8 ядер) з частотою 2.13 ГГц;
- графічні прискорювачі: 2 Tesla M2090;
- обсяг оперативної пам’яті: 24 Гб;
- комунікаційне середовище: InfiniBand 40 Гбіт/с (з підтримкою GPUDirect), Gigabit Ethernet;
- також на вузлах встановлена бібліотека MKL 10.2.6 та CUDA починаючи з версії 3.2.

Чисельні експерименти здійснювались на розріджених матрицях, що приведені в табл. 1. Також тут приведені такі характеристики матриці як її порядок, кількість ненульових елементів. В табл. 2 приведені профілі заповненості вхідних розріджених матриць. На рис. 2 показано залежність часу виконання програми від кількості використовуваних GPU. Причому слід зазначити, що для архітектури *n* CPU + *n* GPU параметр *n* рівний кількості діагональних блоків.

ТАБЛИЦЯ 1. 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

ТАБЛИЦЯ 2. Профіль наповненості ненульовими елементами вихідних матриць

	
G3_circuit	G2_circuit
	
parabolic_fem	apache2

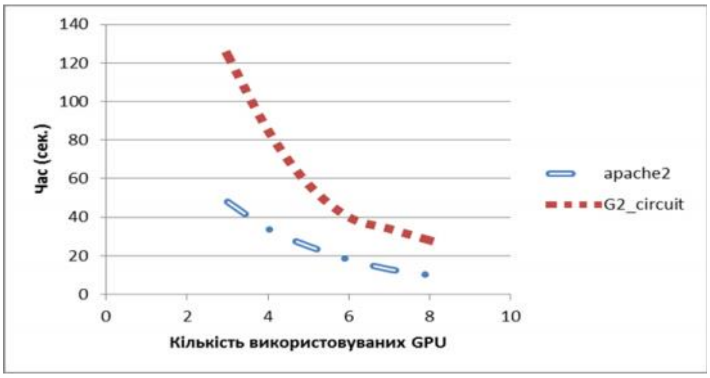


РИС. 2. Часи розв’язання систем з відповідними матрицями

Для програмної реалізації етапів (8), (9) використовувались функції бібліотеки CUBLAS. Для плиток з порядками 8 – 32 використовувались функції cublasDtrsmBatched та cublasDgemmBatched. Для плиток з порядками 64 – 1024 використовувались виклики в циклі функції cublasDtrsm та cublasDgemm, виконання кожної з функцій проходило в окремому потоці cudaStream.

Копіювання даних проходило в асинхронному режимі.

На рис. 3 показано графік залежності продуктивності від використовуваного розміру плитки на прикладі виконання факторизації матриці G2_circuit на гібридній архітектурі 1 CPU та 1 GPU.

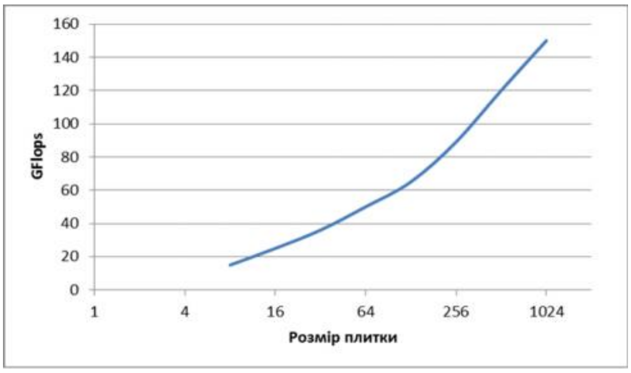


РИС. 3. Продуктивність алгоритму

Висновки. Цей алгоритм має ряд переваг над запропонованим у [3] алгоритмом. Зокрема, алгоритм краще враховує профільну, або розріджену структуру діагональних блоків і матриці в цілому. Можна регулювати розмірність блоку з яким проводяться обчислення на кожному кроці алгоритму, за рахунок цього може досягатись ефект кешизації обчислень, коли блоки поміщаються в швидкій пам’яті GPU. Також така блочна структура дозволяє працювати з нерозривними масивами даних на GPU, що зменшує кількість індексних операцій і перевірок які на графічному прискорювачі є досить затратними.

А.Н. Химич, В.А. Сидорук

ПЛИТОЧНЫЙ ГИБРИДНЫЙ АЛГОРИТМ ФАКТОРИЗАЦИИ
РАЗРЕЖЕННЫХ БЛОЧНО-ДИАГОНАЛЬНЫХ МАТРИЦ С ОБРАМЛЕНИЕМ

Рассматривается новый гибридный алгоритм решения систем линейных алгебраических уравнений с разрежеными симметричными положительно определенными матрицами на компьютерах с графическими ускорителями. Представлены результаты апробации алгоритма на многоядерном компьютере с графическими ускорителями Инпарком

A.N. Khimich, V.A. Sidoruk

HYBRID ALGORITHM FOR SOLVING LINEAR SYSTEMS WITH SPARSE MATRIX
BY DIRECT METHODS

A new hybrid algorithm for solving systems of linear algebraic equations with sparse symmetric positive definite matrices on computers with GPU is considered. The results of testing the algorithm on Inparcom multicore computer are presented.

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 LLT методу // Компьютерная математика. – 2015. – Вып. 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, N 1. – P. 38 – 53.
5. *Химич А.Н., Молчанов И.Н., Мова В.И. и др*. Численное программное обеспечение MIMD-компьютера Инпарком. – Киев: Наук. думка, 2007. – 222 с.

Одержано 10.02.2016

Про авторів:

Хіміч Олександр Миколайович,
доктор фізико-математичних наук, член-кореспондент НАН України,
завідуючий відділом Інституту кібернетики імені В.М. Глушкова НАН України,

Сидорук Володимир Антонович,
молодший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України.