Українська Федерація Інформатики

Інститут кібернетики імені В. М. Глушкова НАН України

Вищий навчальний заклад Укоопспілки
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
(ПУЕТ)

ІНФОРМАТИКА ТА СИСТЕМНІ НАУКИ (ІСН-2015)

МАТЕРІАЛИ
VI ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ ЗА МІЖНАРОДНОЮ УЧАСТЮ

(м. Полтава, 19–21 березня 2015 року)

За редакцією професора О. О. Ємця

Полтава
ПУЕТ
2015

УДК 519.6

ФАКТОРИЗАЦІЯ НЕСИМЕТРИЧНИХ СТРІЧКОВИХ МАТРИЦЬ НА КОМП'ЮТЕРАХ З ГРАФІЧНИМИ ПРИСКОРЮВАЧАМИ

***А. Ю. Баранов**, аспірант*
Інститут кібернетики імені В.М. Глушкова НАН України
abaranov.ua@gmail.com

При чисельному розв'язанні задач в багатьох випадках виникає необхідність розв'язувати систему лінійних алгебраїчних рівнянь (СЛАР). Важливою 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 обрамленням, профільною і тому подібне. В даній роботі розглядається несиметрична стрічкова структура.

Отже, розглянемо систему лінійних алгебраїчних рівнянь:

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

де матриця A – стрічкова несиметрична, n – порядок матриці A, $k1$ – кількість нижніх діагоналей, $k2$ – кількість верхніх діагоналей.

Найбільш ефективним прямим методом розв'язання такої задачі є, як відомо, метод Гауса [2]. Розв'язання системи (1) полягає в розв'язанні підзадач: трикутне розвинення матриці системи (2), розв'язання двох СЛАР з трикутними матрицями (3) та (4):

$$A = PLU \tag{2}$$

$$Ly = b \tag{3}$$

$$Ux = y \tag{4}$$

Обчислювальна складність задачі (1) визначається складністю розв'язання задачі (2). Тому надалі будемо розглядати задачу факторизації стрічкової несиметричної матриці за формулою (2).

У роботі [3] показано, що кількість верхніх діагоналей матриці U дорівнює $k1+k2$. Тому, для опису алгоритму будемо розглядати матрицю A з $k1+k2$ верхніми діагоналями. Нехай порядок стрічкової несиметричної матриці A дорівнює $n=ps$, $k1=ls, k1+k2=us$, де s – це деяке наперед задане натуральне число, як правило вибирається таким чином, щоб блок міг розміститися в кеші як CPU так і GPU. Розіб'ємо матрицю A на квадратні блоки таким чином, щоб в блоці A_{ij} зберігались елементи матриці, які знаходяться рядках $\overline{is, is+s-1}$ та стовбцях $\overline{js, js+s-1}$.

Розглянемо алгоритм факторизації несиметричної стрічкової матриці для гібридних комп'ютерів з графічним прискорювачами. Нехай кількість доступних графічних прискорювачів дорівнює g, будемо їх нумерувати починаючи з нуля. На i-му кроці алгоритму факторизації нам потрібні тільки блоки, які знаходяться в i-му стовбчику, та в u стовбчиках, що слідують за ним. При цьому, в кожному стовбчику знаходиться щонайбільше $u+l+1$ блоків. Таким чином, на кожному кроці алгоритму, достатньо зберігати в GPU тільки ці блоки. Перейдемо до опису гібридного алгоритму факторизації. Блоки матриці A, що знаходяться на GPU, будемо позначати використовуючи верхній індекс d.

Перед початком реалізації алгоритму виконується копіювання блоків матриці A, що знаходяться в перших $u+1$ стовпчиках, в пам'ять GPU. При цьому, блоки, які знаходяться в стовпчику з номером t, копіюються в GPU з номером $(t-1) \bmod g$.

Для кожного $i=\overline{1, p}$ виконати наступні кроки:

1. Копіювання діагонального блоку A^d_{ii} та блоків що знаходяться нижче його в стовбчику i – $A^d_{(i+1)i}, A^d_{(i+2)i}, \ldots, A^d_{qi}$ в пам'ять CPU;
2. Факторизації прямокутної матриці, елементами якої є блоки $A_{ii}, A_{(i+1i}, A_{(i+2)i}, \ldots, A_{q,i}$ на CPU;

3. Копіювання отриманих на попередньому кроці блоків $L_{ii},\ldots,L_{qi}$, U_{ii}, P_i в пам'ять всіх GPU;
4. В кожному GPU виконується перестановки рядків в тих блоках, де це необхідно, використовуючи матриці P_i^d;
5. Обчислення блоків $U^d_{i(i+1)},U^d_{i(i+2)},\ldots,U^d_{ir}$, де $r=\min(i+u,p)$ за формулою:

$$U^d_{i(i+1)} \quad \cdots \quad U^d_{ir} \;=\; L^d_{ii}{}^{-1} \; A^d_{i(i+1)} \quad \cdots \quad A^d_{ir} \tag{5}$$

6. Модифікація блоків A^d_{jt}, де $j=\overline{i+1,q}$ та $t=\overline{i+1,r}$ за формулою:

$$A^d_{jt} \leftarrow A^d_{jt} - L^d_{ji}U^d_{it} \tag{6}$$

7. Якщо $i+u+2<p$, то копіюємо блоки матриці в стовбчику $i+u+2$ в пам'ять GPU з номером $(i+u+1)\bmod g$.

Запропонований паралельний алгоритм реалізовано на комп'ютерах з гібридною (MIMD, SIMD) архітектурою: комп'ютері з графічними прискорювачами Інпарком (Tesla M2090, 2 CPU Intel(R) Xeon(R) E5606 @ 2.13GHz), та СКІТ-4 (Tesla M2050, 4 CPU Intel(R) Xeon(R) X5675 @ 3.07GHz).

Література

1. А. Н. Химич, А. В. Попов, В. В. Полянко: Алгоритмы параллельных вычислений для задач линейной алгебры с матрицами нерегулярной структуры // Кибернетика и систем. аналіз. – 2011. – 47, № 6. – С.159 – 174.
2. Уилкинсон Дж. Х., Райнш К. Справочник алгоритмов на языке Алгол. Линейная алгебра. – М.: Машиностроение, 1976. – 389 с.
3. Gene H. Golub and Charles F. Van Loan. 1996. Matrix Computations (3rd Ed.). Johns Hopkins University Press, Baltimore, MD, USA.