УДК 519.6

А.О. Чорна, І.А. Назарова
Донецький національний технічний університет, м. Донецьк
кафедра прикладної математики та інформатики

АНАЛІЗ МАСШТАБОВАНОСТІ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ МАТРИЧНОГО ДОБУТКУ НА РІЗНОМАНІТНИХ ТОПОЛОГІЯХ

Анотація

Чорна А.О., Назарова. І.А. Аналіз масштабованості паралельних алгоритмі вматричного добутку на різноманітних топологіях. *Розглянуто паралельні алгоритми матричного добутку Кеннона і Фокса. Проведена оцінка ефективності та масштабованості алгоритмів. Побудовано функції ізоефективності паралельних алгоритмів.*

Ключові слова*: паралельні алгоритми, матричний добуток, масштабованість, аналіз ізоефективності, алгоритм Кеннона, алгоритм Фокса.*

Постановка проблеми. При вирішенні різних математичних задач, наприклад, рішення систем лінійних алгебраїчних рівнянь або диференційних рівнянь, однією з основних та трудомістких операцій є матричний добуток. Тому для більш швидкого вирішення завдань більш високого рівня необхідні ефективні алгоритми добутку матриць. Для цього були розроблені паралельні алгоритми, використовувані на різних топологіях: 2D-тор і гіперкуб. За основу паралельних обчислень для матричного добутку при блочному розподілі даних прийнятий підхід, при якому базові під задачі відповідають за обчислення окремих блоків матриці Cі при цьому в під задачах на кожній ітерації розрахунків розташовується тільки по одному блоку вихідних матриць Aі B.У даній роботі розглядаються блокові алгоритми Кенона і Фокса.

Ізоефффективний аналіз

Для паралельних архітектур існує проблема, коли прискорення паралельного алгоритму лише продовжує зростати зі збільшенням числа процесорів, але має тенденцію до насичення або досягненню максимуму на певному значенні. Здатність паралельного алгоритму ефективно використовувати процесори при збільшенні складності розрахунків є важливою характеристикою паралельних обчислень і називається масштабованістю. Для оцінки масштабованості існують різні методи. Одним з таких є аналіз ізоефективності, який заснований на введенні функції ізоефективності. Для цієї функції використовується характеристика, звана загальні накладні витрати T_0. Вона містить у собі сумарні витрати всіх процесорів паралельної системи, враховуючи реалізацію обмінів та послідовну

частину в розпаралелених алгоритмах. Загальні накладні витрати обчислюються за формулою:

$$T_0 = pT_p - T_1,$$

де T_1 – час, необхідний для вирішення завдання заданого розміру на одному процесорі за допомогою найкращого послідовного алгоритму;

T_p – загальний час реалізації паралельного алгоритму на паралельній архітектурі:

$$T_p = T_{p,comp} + T_{p,comm}.$$

Для порівняннямасштабованостейалгоритмівдорівнюютьїхнакладні витратиі будуютьфункціюізоефективності. Функціяізоефективності– це залежністьрозмірурозв'язуваної задачіmвідкількості використовуваних процесорівpдля забезпеченняпостійного рівняефективності паралельнихобчислень:

$$m = f(p).$$

Для обчисленняT_p використовуються такі характеристики:

$T_{p,comp}$ – час вирішення завдання заданого розміру m з використанням паралельного алгоритму на паралельному комп'ютері з p процесорів без урахування обмінних операцій;

$T_{p,comm}$ – час виконання міжпроцесорних операцій обміну при реалізації паралельного алгоритму розв'язання задачі заданого розміру. Ця характеристика враховує t_s –латентність, тривалість підготовки повідомлення для передачі та t_w – час передачі одного байта.

Характеристикою продуктивності процесора є кількість операцій з плаваючою точкою, час виконання яких позначається t_{op}. Прийнято, що будь-який вид операції (додавання і добуток) виконуються за однаковий час.

$$t_{mal} = t_{add} = t_{op}$$

Залежно від топології і виду комунікаційної операції, для розрахунку $T_{p,comm}$ використовується модель Хокні.

Алгоритм Кеннона

У блочному алгоритмі матриці розбиваються на блоки, які становлять собою під матриці вихідних матриць. При цьому кожен блок C_{ij} матриці C визначається як добуток відповідних блоків матриць A і B.

Нехай вихідні матриці мають розмірність $m * m$ і розбиваються на квадратні блоки порядку q, де $q^2 = p$– кількість блоків та процесорів. Ідея алгоритму полягає в зміні схеми початкового розподілу блоків перемножуваних матриць між процесорами обчислювальної системи. Початкове розташування блоків в алгоритмі Кеннона підбирається таким чином, щоб розташовувані блоки на процесорах могли б бути перемножити без будь-яких додаткових передач даних між процесорами. При цьому подібний розподіл блоків може бути організовано таким чином, що переміщення блоків між процесорами в ході обчислень може здійснюватися з

використанням більш простих комунікаційних операцій. Алгоритм можна розділити на два етапи – це ініціалізація матриць і основний цикл алгоритму:

I. Ініціалізація матриць:

– для кожного рядка i, крім першого, циклічний зсув блоків на $i-1$ позицій вліво;

– для кожного стовпця j, крім першого, циклічний зсув блоків на $j-1$ позицій вгору.

II. Основний цикл:

– обчислюємо $C_{ij} = C_{ij} + A_{ij} * B_{ij}$;

– $q-1$ раз: для всіх рядків виконується зсув блоків вліво, для всіх стовпців –зсув блоків вправо, обчислюємо$C_{ij} = C_{ij} + A_{ij} * B_{ij}$.

Алгоритм Фокса

Початкові дані візьмемо такі ж, як і для алгоритму Кеннона. В ході обчислень на кожній базовій підзадачі (i,j) розташовується чотири матричних блоки:

– блок C_{ij} матриці C, обчислюваний підзадачею;

– блок A_{ij} матриці A, що розміщується в підзадачі перед початком обчислень;

– блоки $A\grave{}_{ij}$, $B\grave{}_{ij}$ матриць Aі B, одержувані підзадачею в ході виконання обчислень.

Алгоритм Фокса складається з таких етапів:

I. Етап ініціалізації: кожній підзадачі(i,j)передаються блоки A_{ij}, B_{ij}і обнуляються блоки C_{ij};

II. Етап обчислень: на кожній ітерації l, $0 \le l < q$, здійснюються такі операції:

– для кожного рядка i, $0 \le i < q$, блок A_{ij}підзадачі(i,j) пересилається на всі підзадачі того ж рядка решіткиiде індекс $j = (i+1)\ mod$;

– отримані в результати пересилань блоки $A\grave{}_{ij}$, $B\grave{}_{ij}$ кожної підзадачі(i,j)перемножуються і додаються до блоку C_{ij};

– блоки$C\grave{}_{ij}$кожної підзадачі(i,j) пересилаються підзадачам, що є сусідами зверху в стовбцях решітки підзадач(блоки підзадачз першого рядка решітки пересилаються підзадачам останнього рядка решітки).

Оцінка ефективності

Визначимообчислювальну складністьданих алгоритмів.Висхідні дані: матриця розміром $m \times m$, кількість блоків $q^2 = p$ для топології 2D – тор та $q^3 = p$ для топології гіперкуб, розмірність блоків $k = m/q$.Час послідовного алгоритму для обох алгоритмів однаковий и дорівнює:$T_1 = 2t_{op}m^3$.

Алгоритм Кеннона на топології 2D – тор.

Обчислювальна трудомісткість :$T_{p,comp}^{Canon} = t_{op}\left(\frac{2m^3}{p} + \frac{m^2}{\sqrt{p}}\right)$.

Час на обмінні операції:$T_{p,comp}^{Canon} = 4\left(\sqrt{p}-1\right)\left(t_s + \frac{m^2}{p}t_w\right)$.

Загальний час реалізації паралельного алгоритму на паралельній архітектурі:$T_p^{Canon} = t_{op}\left(\frac{2m^3}{p} + \frac{m^2}{\sqrt{p}}\right) + 4\left(\sqrt{p}-1\right)\left(t_s + \frac{m^2}{p}t_w\right)$.

Загальні накладні витрати:

$$T_0^{Canon} = m^2\left(t_{op}\sqrt{p} + 4\sqrt{p}t_w - 4t_w\right) + 4pt_s(\sqrt{p}-1).$$

Алгоритм Фокса на топології 2D – тор.

Обчислювальна трудомісткість :$T_{p,comp}^{Canon} = t_{op}\left(\frac{2m^3}{p} + \frac{m^2}{\sqrt{p}}\right)$.

Час на обмінні операції:$T_{p,comp}^{Fox} = t_s\left(2\sqrt{p}-1\right) + m^2t_w(\frac{1}{\sqrt{p}} + \frac{1}{2} - \frac{1}{p})$.

Загальний час реалізації паралельного алгоритму на паралельній архітектурі:$T_p^{Fox} = t_{op}\left(\frac{2m^3}{p} + \frac{m^2}{\sqrt{p}}\right) + t_s\left(2\sqrt{p}-1\right) + m^2t_w\left(\frac{1}{\sqrt{p}} + \frac{1}{2} - \frac{1}{p}\right)$.Загальні накладні витрати:$T_0^{Fox} = \left(t_{op}\sqrt{p} + t_w(\sqrt{p} + \frac{p}{2} - 1)\right) + pt_s(2\sqrt{p}-1)$.

Функція ізоефективності алгоритмів Кеннона і Фокса для 2D-тор:

$T_0^{Canon} = T_0^{Fox}$, $m^2\left(t_{op}\sqrt{p} + 4\sqrt{p}t_w - 4t_w\right) + 4pt_s\left(\sqrt{p}-1\right) =$

$m^2\left(t_{op}\sqrt{p} + t_w(\sqrt{p} + \frac{p}{2} - 1)\right) + pt_s\left(2\sqrt{p}-1\right)$,

$m = \sqrt{\frac{t_s p(2\sqrt{p}-3)}{t_w(\frac{p}{2}+3-3\sqrt{p})}}$.

Алгоритм Кеннона для топологіїгіперкуб.

Обчислювальна трудомісткість :

$T_{p,comp}^{Canon} = t_{op}\left(\frac{2m^3}{\sqrt[3]{p^2}} + \frac{m^2}{\sqrt[3]{p}}\right)$.

Час на обмінні операції:

$T_{p,comp}^{Canon} = 4\left(\sqrt[3]{p}-1\right)\left(t_s + \left(\frac{m}{\sqrt[3]{p}}\right)^2 t_w \log_2 \sqrt[3]{p}\right)$.

Загальний час реалізації паралельного алгоритму на паралельній архітектурі:

$$T_p^{Canon} = t_{op}\left(\frac{2m^3}{\sqrt[3]{p^2}} + \frac{m^2}{\sqrt[3]{p}}\right) + 4\left(\sqrt[3]{p}-1\right)\left(t_s + \frac{m^2}{\sqrt[3]{p^2}}t_w \log_2 \sqrt[3]{p}\right).$$

Загальні накладні витрати:

$$T_0^{Canon} = 2t_{op}m^3\left(\sqrt[3]{p}-1\right) + m^2\left(t_{op}\sqrt[3]{p^2} + 4\sqrt[3]{p^2}t_w \log_2 \sqrt[3]{p} - 4t_w\sqrt[3]{p}\log_2 \sqrt[3]{p}\right) + 4pt_s(\sqrt[3]{p}-1) \quad (17).$$

Алгоритм Фокса на топології гіперкуб.

Обчислювальна трудомісткість:

$$T_{p,comp}^{Fox} = q\left(2t_{op}k^3 + t_{op}k^2\right) = t_{op}\left(\frac{2m^3}{\sqrt[3]{p^2}} + \frac{m^2}{\sqrt[3]{p}}\right).$$

Час на обмінні операції:

$$T_{p,comp}^{Fox} = t_s\left(\sqrt[3]{p}\log_2\sqrt[3]{p} + \sqrt[3]{p} - 1\right) + m^2 t_w \log_2\sqrt[3]{p}\,(\frac{2}{\sqrt[3]{p}} - \frac{1}{\sqrt[3]{p^2}}).$$

Загальний час реалізації паралельного алгоритму на паралельній архітектурі:

$$T_p^{Fox} = t_{op}\left(\frac{2m^3}{\sqrt[3]{p^2}} + \frac{m^2}{\sqrt[3]{p}}\right) + t_s\left(\sqrt[3]{p}\log_2\sqrt[3]{p} + \sqrt[3]{p} - 1\right) + m^2 t_w \log_2\sqrt[3]{p}\,(\frac{2}{\sqrt[3]{p}} - \frac{1}{\sqrt[3]{p^2}}).$$

Загальні накладні витрати:

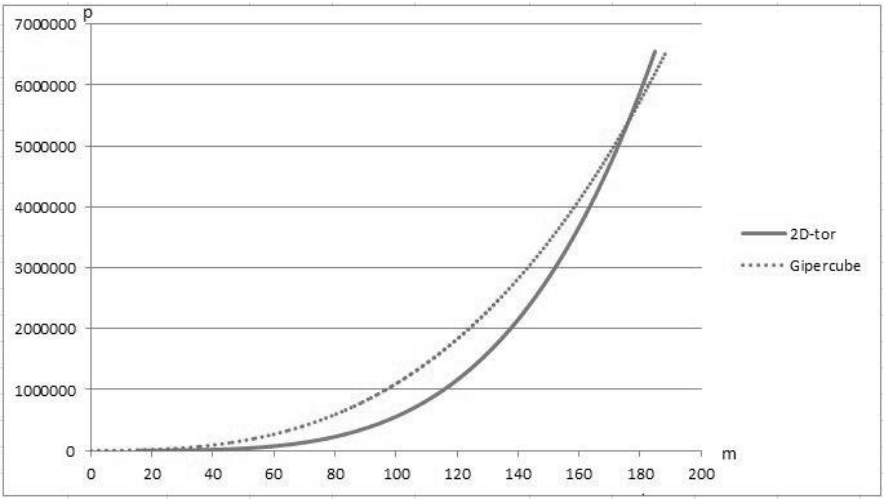
$$T_0^{Fox} = 2t_{op}m^3\left(\sqrt[3]{p} - 1\right) + m^2\left(t_{op}\sqrt[3]{p^2} + t_w\log_2\sqrt[3]{p}\,(\frac{2}{\sqrt[3]{p}} - \frac{1}{\sqrt[3]{p^2}})\right) + t_s p\sqrt[3]{p}(\log_2\sqrt[3]{p} - 1).$$

Функція ізоефективності алгоритмів Кенноа і Фокса для топології 2D – тор:

$$m^2\left(t_{op}\sqrt[3]{p^2} + 4\sqrt[3]{p^2}t_w\log_2\sqrt[3]{p} - 4t_w\sqrt[3]{p}\log_2\sqrt[3]{p}\right) + 4pt_s(\sqrt[3]{p} - 1) = m^2\left(t_{op}\sqrt[3]{p^2} + t_w\log_2\sqrt[3]{p}\,(\frac{2}{\sqrt[3]{p}} - \frac{1}{\sqrt[3]{p^2}})\right) + t_s p\sqrt[3]{p}(\log_2\sqrt[3]{p} - 1),$$

$$m = \sqrt{\frac{t_s p\left(\sqrt[3]{p}(\log_2\sqrt[3]{p} - 3) + 3\right)}{t_w\left(2\sqrt[3]{p^2}\log_2\sqrt[3]{p} - 3\sqrt[3]{p}\log_2\sqrt[3]{p}\right)}}.$$

Отримуємо наступні графіки функцій ізоефективності алгоритмів Кеннона і Фокса для топологій 2D - тор гіперкуб (мал. 1)



Малюнок 1 – Графіки функцій ізоефективності алгоритмів Кеннона і Фокса

При побудові графіків коефіцієнти приймали такі значення: $t_{op} = 1$, $t_s = 10$, $t_w = 3$. Як бачимо з графіків, для матриць однакової розмірності топологія

2D – тор потребує меншу кількість процесорів. Це виконується для розмірності матриць менше 180. Але для малої кількості процесорів (менш, ніж 20) топологію гіперкуб для цих алгоритмів застосувати не можна. Для топології 2D – тор вище графіка ефективніший алгоритм Фокса (його накладні витраті менші за витрати алгоритмаКеннона), а для топології гіперкуб вище графіка краще алгоритм Кеннона.

Висновки

У роботі наведені результати досліджень авторів, присвячені оцінці ефективності та масштабованості паралельних алгоритмів матричного добутку таких, як Кеннона і Фокса на різноманітних топологіях. Проведено аналітичний аналіз масштабованності на основі динамічних характеристик паралельних алгоритмів, отримані експериментальні результати та виявлені переваги та недоліки відображення на топології гіперкуб та тор. Перспективним напрямом дослідження є застосування апарату ізоефективного аналізу для визначення ступеню масштабованності та отримання пріоритетних областей застосування більш складних алгоритмів, наприклад, методів паралельного розв’язання багатовимірної задачі Коші.

Список літератури

1. Gupta A., Kumar V. Scalability of parallel algorithm for matrix multiplication // Technical report TR-91-54, Departmentof CSU ofMinneapolis, 2001. – P. 1-211.
2. Фельдман Л.П., Назарова И.А. Современные параллельные методы численного решения задачи Коши. - Донецк: ГВУЗ “ДонНТУ”, 2013. – 206с.
3. Назарова І.А., Фельдман Л.П. Масштабованість паралельного розв'язання систем звичайних диференційних рівнянь для мультикомп'ютерів із розподіленою пам'яттю//Материалы международной научно-технической конференции «Искусственныйинтеллект. Интеллектуальные системы: ИИ-2010».–Таганрог-Донецк-Минск:Изд-во ИПИИ, т.1, 2010. – С. 159-163.
4. Фельдман Л.П., Назарова И.А., Хорошилов А.В. Параллельные блочные алгоритмы умножения матриц для мультикомпьютеров с распределенной памятью.// Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка, випуск 8(120): – Донецьк, ДонНТУ, 2007. - С. 297-309.