УДК 658.012.011.56:681.3.06
Я.Кінах
Тернопільська академія народного господарства

УДОСКОНАЛЕННЯ РОЗПАРАЛЕЛЕННЯ МАТРИЧНИХ ОПЕРАЦІЙ АЛГОРИТМУ ЗАГАЛЬНОГО РЕШЕТА ЧИСЛОВОГО ПОЛЯ

В статті подано розв'язок задачі удосконалення алгоритму загального решета числового поля для етапу матричних операцій, що дозволяє на практиці уникнути необхідності використовувати суперкомп'ютери, та скоротити час виконання криптоаналітичного алгоритму.

Умовні позначення

ЗРЧП – загальне решето числового поля;
СЛАР – система лінійних алгебраїчних рівнянь;
СУБД – система управління базою даних;
ТАС – типова алгоритмічна структура.

Перед тим, як переробляти код для переходу на паралельний комп'ютер, треба грунтовно проаналізувати код програми. Частка послідовних операцій алгоритму ЗРЧП незначна, тому прискорення буде малим. Отже, потрібно замінювати окремі компоненти алгоритму ЗРЧП.

Розглянемо організацію процесу блокового розпаралелення розв'язання СЛАР великої розмірності на основі прямих декомпозиційних методів, використовуючи викладене в [1].

Нехай $HX=Q$ - розріджена система розмірності $n \times n$, яка подана у вигляді

$$\begin{bmatrix} H_1 & \dots & & \Phi_1 \\ \dots & \dots & \dots & \dots \\ & \dots & H_N & \Phi_N \\ \theta_1 & \dots & \theta_N & W_N \end{bmatrix} \begin{bmatrix} X_1 \\ \dots \\ X_N \\ X_C \end{bmatrix} = \begin{bmatrix} Q_1 \\ \dots \\ Q_N \\ Q_C \end{bmatrix}, \qquad (1)$$

де H_i, X_i і Q_i - неособливі підматриці коефіцієнтів розмірності $n_i \times n_i$;
n_i - підвектори невідомих змінних і правих частин i-ї підсистеми рівнянь відповідно;
Φ_i і θ_i - прямокутні підматриці розмірності $n_i \times n_c$ і $n_c \times n_i$ відповідно;
W_C, X_C і Q_C - квадратні підматриці зв'язку підсистем розмірності $n_c \times n_c$;
n_c - підвектори невідомих змінних і правої частини підсистеми рівнянь зв'язку, причому $i = \overline{1,N}$, $n_i << n$, $n_c << n$, $\sum_{i=1}^{N} n_i + n_c$.

Підматриця Φ_i характеризує вплив підвектора змінних зв'язку X_C на рівняння кожної підматриці, а підматриці θ_i і W_C - зв'язок підвекторів невідомих змінних X_i і підвектора змінних зв'язку X_C між собою.

Завдяки особливостям структури блокової матриці коефіцієнтів, обробка кожного блоку або підсистеми рівнянь може виконуватись незалежно на одному чи декількох процесорах [2].

Процес блокового розв'язання СЛАР методом Гауса складається з чотирьох послідовних етапів [1].

На першому етапі при W_C=0 виконуються паралельно часткові виключення в i-тих підсистемах рівнянь вихідної системи

$$\begin{bmatrix} H_i & \Phi_i \\ \theta_i & \end{bmatrix}\begin{bmatrix} X_i \\ X_c \end{bmatrix} = \begin{bmatrix} Q_i \\ Q_c \end{bmatrix},\ i = \overline{1,N}, \tag{2}$$

виділених з (1), після чого вони набудуть форми

$$\begin{bmatrix} H_{mi} & \Phi_{mi} \\ & W_{ci} \end{bmatrix}\begin{bmatrix} X_i \\ C_i \end{bmatrix} = \begin{bmatrix} Q_{mi} \\ Q_{ci} \end{bmatrix}, \quad i = \overline{1,N}, \tag{3}$$

де перетворені підматриці H_{mi}, Φ_{mi} і підвектор Q_{mi}, а також сформована в процесі розв'язку підматриця W_{ci} і підвектор правої частини Q_{ci} рівняння підсистеми зв'язку визначаються вибраним методом розв'язку СЛАР.

На другому етапі формується підматриця K_c і підвектор правої частини b_c підсистеми рівнянь зв'язку:

$$K_c = W_c + \sum_{i=1}^{N} W_{ci},$$
$$b_c = Q_c + \sum_{i=1}^{N} Q_{ci}. \tag{4}$$

На третьому етапі розв'язуємо перетворену підсистему рівнянь зв'язку $K_cX_c=b_c$ та обчислюємо підвектор невідомих змінних зв'язку X_c.

На четвертому етапі визначаються підвектори невідомих змінних X_i з рівняння блоку $H_{mi}X_i=Q_{mi}-\Phi_{mi}X_c,\ i = \overline{1,N}$.

В результаті отримаємо паралельно на N процесорах значення всіх компонентів вектора невідомих змінних для вихідної СЛАР $X=(X_1, X_2,...,X_N,X_c)^T=(x_1,x_2,...,x_i,...,x_n)^T$. Верхній індекс T означає операцію транспонування матриці.

Наведена вище обчислювальна схема організації процесу розв'язування СЛАР великої розмірності дозволяє виділити в алгоритмах розв'язку основний рівень перетворення алгоритмів, тобто блоки обчислень. На першому та четвертому етапах - $2N$ блоків, на другому - два блоки і на третьому - один блок. Якщо кількість процесорів або блоків i не обмежувати числом N, то можна здійснити розпаралелення розв'язання підсистеми на третьому етепі, а перетворення підсистем рівнянь на першому.

Обчислення всередині окремих блоків з перетворення підсистем рівнянь (2), (3) володіють природнім паралелизмом, оскільки всі операції над компонентами підматриць і підвекторів виконуються незалежно один від одного.

Точність отриманого розв'язку за такої організації обчислень на рівні крупномодульного розпаралелювання блоків не буде відрізнятись від точності послідовних алгоритмів даного методу, якщо найбільші за абсолютним значенням елементи в підсистемах рівнянь групуватимуться в підматрицях коефіцієнтів H_i.

Одним з ключових моментів при проведенні трудомістких обчислювальних експериментів з обробки розрідженої матриці H великої розмірності методом Гауса, що виникає під час роботи алгоритму ЗРЧП, є зменшення часу розробки паралельних додатків. Для цього можна використати обмежені форми паралелизму, подані у вигляді ТАС з масивним паралелизмом, та розробці механізмів їх повторного використання. В програмуванні ці структури відомі під різними назвами: алгоритмічні шаблони, родові алгоритми, архітипи, типові проектні рішення.

Терміном ТАС позначають сімейство однотипних параметризованих алгоритмів

$$S=A(P_S, P_F), \tag{5}$$

де S – множина алгоритмів A з набором P_S структурних та P_F функціональних параметрів.

Підхід передбачає реалізацію ТАС у вигляді шаблонів – заготовок, в яких зафіксована інваріантна частина, і є розподіленою комунікаційною структурою. Настроювання шаблону на конкретну область застосування здійснюється фіксацією структурних та функціональних параметрів. Структурні параметри визначають топологію паралельної програми, функціональна частина визначає обчислювальний зміст алгоритму.

Множину ТАС можна розбити на базові та проблемно – орієнтовані структури. Базові алгоритмічні структури не орієнтовані на конкретну область застосування і мають універсальний характер. Проблемно – орієнтовані алгоритмічні структури – це високорівневі конструкції, які характеризуються вужчою областю застосування, що виникають в результаті узагальнення групи подібних обчислювальних методів.

За традиційного розпаралелення обчислення виконує один процес, а результат розсилається іншим процесам, що беруть участь в роботі алгоритму. В цілій низці випадків роботи алгоритму ЗРЧП ефективнішою є схема обчислень і відповідна до неї структура, в основу котрої покладене дублювання послідовних обчислень, оскільки економиться час на виконання обміну інформацією. В цій структурі відсутні витрати на декомпозицію та збір даних, відповідні проектори є тотожніми функціями, а послідовна функція співпадає з паралельною.

Глобальна структура програми цілком визначається алгоритмічною структурою методу ЗРЧП. В ієрархічному плані програма – це дворівневе дерево, в корені якого знаходиться ТАС, а листки дерева сформовані з послідовних модулів – параметрів ТАС.

Таким чином, в найпростішому випадку ТАС виступають операторами композиції або конструкторами, що дозволяють збирати складні програми з послідовних модулів. Використання ТАС як єдиного способу композиції накладає жорсткі обмеження на структуру програми.

Якщо функціональні параметри ТАС не лише послідовні модулі, то можна будувати програми з вкладеною ієрархічною структурою. Ієрархічна структура програми – це дерево, в корені котрого знаходиться ТАС, що визначає глобальну структуру програми, нетермінальні вершини складаються з ТАС, листки дерева відповідають послідовним модулям.

Вкладеність ідеально відповідає методу розробки програм “зверху – вниз”, що дозволяє проводити ієрархічну декомпозицію матриці H на окремі підматриці H_i.

Очевидним є факт: чим складніша конфігурація, що забезпечує доступ до даних робочої станції, тим частіше відбуваються порушення в її роботі. Є ще один чинник, котрий пов'язаний із тим, що багато засобів розробки використовують ті самі стандартні бібліотеки доступу до даних (у випадку Windows це BDE і ODBC). На сьогодні є чимала кількість різноманітних програмних продуктів (особливо енциклопедій і довідників), що містять дані, при установці яких встановлюються і ці бібліотеки (подібні дії дозволені за певних умов виробниками цих бібліотек).

Нерідко застосовуються так звані екстенсивні заходи (нарощування апаратної частини робочих станцій, збільшення пропускної здатності мережі, прокладання нових ліній зв'язку, перенесення застарілих даних в архіви з метою зменшення обсягу бази даних), що звичайно вимагають чималих технічних витрат, особливо для великої кількості користувачів і високої швидкості росту обсягу бази даних.

Є інші, інтенсивні способи розв'язання подібних задач. Вони можуть мати різноманітну реалізацію, але в цілому використовують одну загальну ідею. Ця ідея полягає в створенні нових сервісів, загальних для користувачів інформаційної системи.

Такі сервіси в загальному є сервісами проміжного прошарку (middleware services), оскільки займають проміжний рівень між даними і сервісами, що їх обслуговують, з одного боку, і користувальними додатками, орієнтованими на конкретну предметну область, з іншого боку. Ці сервіси звичайно мають мінімальний користувальний інтерфейс або не мають його зовсім. Нерідко вони можуть бути

реалізовані для декількох різноманітних платформ, тому що є сервісами більш високого рівня, ніж сервіси, специфічні для даної операційної системи або СУБД. Такі сервіси можна реалізувати всередині додатків або бібліотек, так званих - Application Server, а також у вигляді служб операційних систем.

Таким чином, наведене удосконалення забезпечить паралельну обробку матриці H, що значно зменшує час виконання матричних операцій методу ЗРЧП, а також час виконання алгоритму ЗРЧП загалом на 8%.

Their decomposition is done and the optimization rulers of decomposition are formulated. The calculated system is offered which allows the algorithm NFS to be applied parallel. It shortens the full filament time of the algorithm in order to define the level of the reliability of the algorithm RSA, El-Gamala and and use the safe keys in practice.

Література

1. Нагорный Л.Я. Жуков И.А. Метод решения систем больших размерностей на многопроцессорных структурах // Автоматизация проектирования в электронике. - Киев: Техніка.-1979. - Вып.20. - С. 76 - 80.
2. Нагорный Л.Я., Жуков И.А. О формировании блочно-диагональных матриц систем линейных уравнений высокого порядка для решения их на ЦВМ // Электроника и моделирование.-Киев: Наукова думка.- 1976.-Вып.12. - С. 50 - 53.
3. Нагорный Л.Я., Жуков И.А Решение больших систем нелинейных уравнений методом декомпозиции с использованием операции развёртывания матриц // Вопросы вычислительной и прикладной математики. - Ташкент: Ин-т кибернетики с ВЦ АН Уз.ССР.-1979. - Вып.58. - С.39 - 45.

Одержано 21.11.2003 р.