Практичне заняття № 6.
Розпаралелювання циклічних ділянок алгоритмів або програм: модифікації методу пірамід.

Як уже було зазначено на одному із попередніх практичних занять, *головним недоліком* методу пірамід є його *неекономність*, причиною якої є *дублювання* обчислень у різних пірамідах. Звичайно, у такому випадку виникає *проблема* зменшення частки ітерацій, які дублюються. Взагалі кажучи її можна вирішити двома шляхами. **Перший шлях** – це більш ретельне виділення пірамід для того, щоб в кожну з них увійшли лише ті ітерації, що дійсно потрібні для цієї піраміди. **Другий шлях** – звуження пірамід шляхом ігнорування деяких зв'язків, які найбільш «відхиляються», та реалізація цих зв'язків за допомогою спеціальних операторів обміну або шляхом організації такого порядку виконання пірамід, який гарантував би правильну передачу інформації по цих зв'язках. Перший підхід було розглянуто на одному з попередніх занять. Зараз зупинемось більш детально на другому підході.

Організація обмінів між гілками. Спеціальні оператори обміну.

З метою зменшення частки дубльованих ітерацій звуження пірамід може бути здійснене двома способами. Розглянемо перший спосіб – ігнорування найбільш «відхилюваних» векторів і реалізація відповідних їм зв'язків між пірамідами за допомогою вставлених у необхідні місця операторів обміну.

Почнемо викладки з простого прикладу циклу.

Приклад 1.

$$
\begin{aligned}
&FOR \quad i = 1,5 \quad DO \\
&FOR \quad j = 1,6 \quad DO \\
&\{ \ x(i,j) = f(x(i-2, j+1)) \\
&\quad y(i,j) = g(y(i-1, j)) \\
&\quad z(i,j) = h(z(i-1, j-2)) \ \ \}.
\end{aligned}
\tag{1}
$$

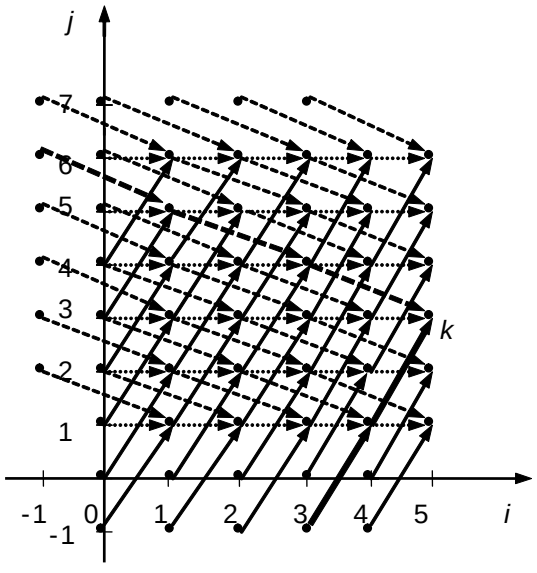

Рис. 1

Тут *f, g, h* – деякі явно задані функції.

Простір ітерацій і структура залежностей в ньому для даного циклу приведені на рис. 1. Найбільш «неприємним» вектором зв'язків у цьому прикладі є вектор (1, 2) – результат взаємодії ітерацій через масив *z*. Пряме застосування методу пірамід дало б досить незначну економію. Ми спробуємо отримати результуючу конструкцію з пірамід, утворених векторами, що залишилися: (1, 0) і (2, -1). Більш точніше, будемо мати на меті одержання наступної результуючої паралельної конструкції у формі:

$$
\begin{aligned}
&FOR \quad k = 1,6 \quad DO \quad AUTON \\
&FOR \quad i = 1,5 \quad DO \\
&FOR \quad j = k, k + [(5 - i)/2] \quad DO \\
&\{ \; x(i, j) = f(x(i-2, j+1)) \\
&\quad y(i, j) = g(y(i-1, j)) \\
&\quad z(i, j) = h(z(i-1, j-2)) \; \},
\end{aligned}
\tag{2}
$$

яка в наведеному тут вигляді поки що не є еквівалентною до (1). Крім цього, з метою технічних спрощень ми вважаємо, що ітерації пірамід, які не потрапляють в межі циклу, заміняються деякими фіктивними ітераціями, що не змінюють значення пам'яті. В нашому випадку такими є ітерація (1, 7) при k = 5, ітерації (1, 7), (1, 8), (2, 7), (3, 7) при k = 6. Можна вважати, що вони мають вигляд:

$$x(i, j) = x(i, j);\; y(i, j) = y(i, j);\; z(i, j) = z(i, j).$$

За таких припущень кожна з шести пірамід має по дев'ять ітерацій. На рис. 2 зображений граф інформаційних зв'язків між пірамідами, який необхідно реалізувати для того, щоб конструкції (1) і (2) стали еквівалентними. При цьому пунктиром показані зв'язки, які можуть бути реалізовані альтернативними способами. Так, наприклад, змінна *z*(1, 2), необхідна для 5-ої ітерації 3-ої піраміди, може бути отримана з 2-ої ітерації 1-ої піраміди або з 1-ої ітерації 2-ої піраміди (оскільки це одна і та ж ітерація (1, 2)). Альтернативні джерела для кожної з ітерацій можуть бути з'єднані, тоді отримаємо картину, приведену на рис. 3. Стрілки показують один із можливих способів обміну інформацією.

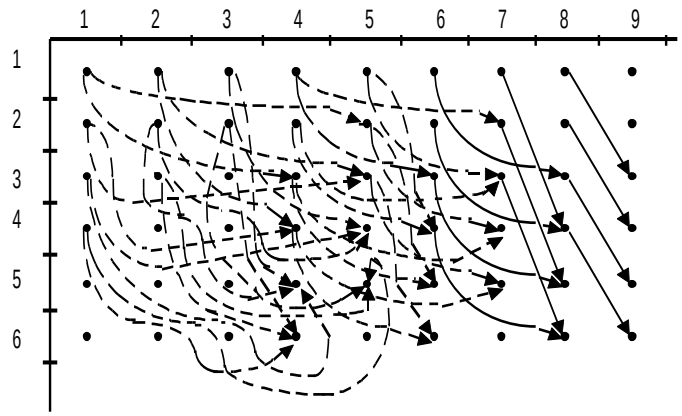

Рис. 2

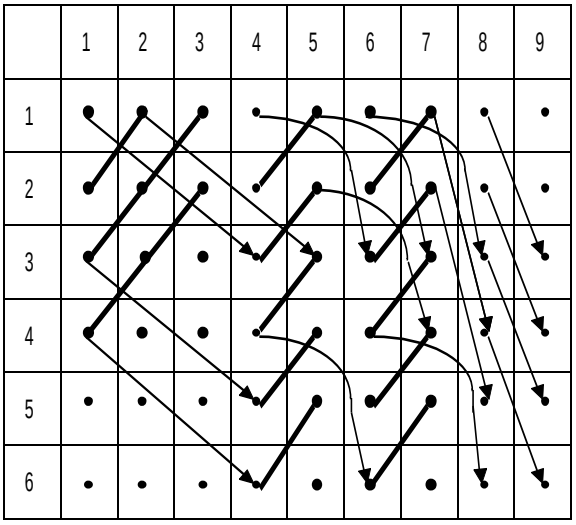

Рис. 3

Оскільки ми маємо певну свободу вибору конкретного режиму передачі інформації, то під час цього вибору може розв'язуватися деяка оптимізаційна задача, наприклад, задача **мінімізації кількості каналів** між гілками або задача **мінімізації кількості операторів** обміну.

Під час вибору тої чи іншої структури передачі інформації необхідно слідкувати, щоб інформація, яка передається, не затерла якої-небудь ще не використаної інформації, і сама не була затерта до моменту свого використання. Все це залежить від семантики оператора передачі, що використовується. Розглянемо можливі мовні способи опису актів передачі та прийому даних.

Найбільш розповсюдженими є ОККАМ-подібні оператори: $x \uparrow k$ – переслати змінну x у гілку (піраміду) з номером k, $x \downarrow k$ – зчитати змінну x із гілки з номером k. Семантика цих операторів, якщо їх розглядати кожен окремо, досить проста. Як тільки у гілку з номером l контроль доходить до оператора $x \uparrow k$, значення змінної x присвоюється одноіменній змінній у гілці з номером k. Якщо в цій гілці змінна x досі не використовувалась ні для читання, ні для запису, то вона заводиться як вхідна. Це є еквівалентним використанню оператора: ввід (x). Аналогічно, спрацьовування $x \downarrow k$ полягає в зчитуванні в комірку x значення із одноіменної комірки гілки з номером k.

Якщо значення j-ої координати, по якій ведеться перебір пірамід, для вектора, що відхиляється, дорівнює p (в нашому випадку $p = 2$), то зрозуміло, що при будь-якому k вся інформація, що передається по цьому вектору в k-ту піраміду, виробляється в (k-p)-ій піраміді. Таким чином, складається враження, що конструкція (2) могла б стати коректною, якщо переписати останній рядок у вигляді:

$$z(i,j) = h(z(i-1,j-2) \downarrow (k-2)) \quad \} \tag{3}$$

або у вигляді:

$$z(i,j) \uparrow (k+2) = h(z(i-1,j-2)) \quad \}. \tag{4}$$

Однак, кожен із цих операторів, взятий окремо, не забезпечує необхідного режиму обміну інформацією. В даному випадку в (3) присвоювання може почати вико-

нуватись раніше, ніж в (*k*-2)-ій гілці виробиться змінна $z(i-1, j-2)$. Аналогічно, в (4) передача значення змінної $z(i, j)$ може здійснюватися пізніше за її використання в (*k*-2)-ій гілці. Таким чином, в обох випадках використовувана в (*k*+2)-ій гілці змінна зчитається із власної пам'яті, а не із пам'яті *k*-ої гілки, як це вимагається. Виходом із такої ситуації є спільне використання операторів $x \uparrow k$ та $x \downarrow k$, тобто подання останнього рядка у вигляді:

$$z(i, j) \uparrow (k+2) = h(z(i-1, j-2) \downarrow (k-2)) \quad \}.$$

Семантика взаємодії цієї пари операторів може бути описана таким чином. При зустрічі оператора $z(i, j) \uparrow (k+2)$ змінна $z(i, j)$ викидається в канал-чергу типу FIFO, який з'єднує *k*-ту і (*k*+2)-гу гілки, а зчитування $z(i-1, j-2) \downarrow (k-2)$ здійснюється виключно з цього каналу. Слід відзначити, що одержана конструкція не буде повністю правильною, оскільки тут має місце «крайовий ефект»: перша і друга піраміди не мають джерел на рівні (*k*-2)-ої піраміди. Тому для правильності згаданої конструкції будемо вважати, що до моменту її спрацьовування у відповідні канали здійснене засилання значень потрібних нам змінних. Тобто паралельна конструкція доповнюється фіктивними каналами на рівні перших двох пірамід. Крім цього, в кожному конкретному випадку необхідно перевіряти оператори обміну на коректність використання: передача даних із *l*-ої гілки в *j*-ту повинна здійснюватись у тій же послідовності, що і їх використання.

Конвеєрний запуск «обрізаних» пірамід.

Розглянемо **інший спосіб організації обмінів** між паралельними гілками, що полягає в конвейєрному запуску «обрізаних» конструкцій (пірамід). Даний спосіб грунтується на використанні мовних засобів високого рівня. Зупинемось на використанні службових слів, що задають взаємодію між гілками через спільну пам'ять (*SYNCH*(*j*), *PIPE*(*i*, *j*), *SIM* і т. д.).

Приклад 2.

Нехай вихідний цикл має вигляд:

$$\begin{aligned} &FOR \quad i = 1,6 \quad DO \\ &FOR \quad j = 1,8 \quad DO \\ &x(i, j) = f(x(i-1, j), x(i-1, j-1)). \end{aligned} \tag{5}$$

Тут *f* – функція, задана явно.

Граф залежності між ітераціями цього циклу поданий на рис. 4. За методом пірамід одержимо наступну конструкцію паралельного виконання (5):

$$\begin{aligned} &FOR \quad k = 1,8 \quad DO \quad AUTON \\ &FOR \quad i = 1,6 \quad DO \\ &FOR \quad j = \max\{1, i+k-6\}, k \quad DO \\ &x(i, j) = f(x(i-1, j), x(i-1, j-1)). \end{aligned} \tag{6}$$

Перехід до «обрізаних» пірамід дає можливість отримати ціле сімейство паралельних конструкцій. Так, одна із них може бути записана таким чином:

$$
\begin{aligned}
&FOR \;\; k = 1,8 \;\; DO \;\; SYNCH(1)\\
&FOR \;\; i = 1,6 \;\; DO\\
&FOR \;\; j = \max\{1, k-2, i+k-6\}, k \;\; DO\\
&x(i, j) = f(x(i-1, j), x(i-1, j-1)).
\end{aligned} \qquad (7)
$$

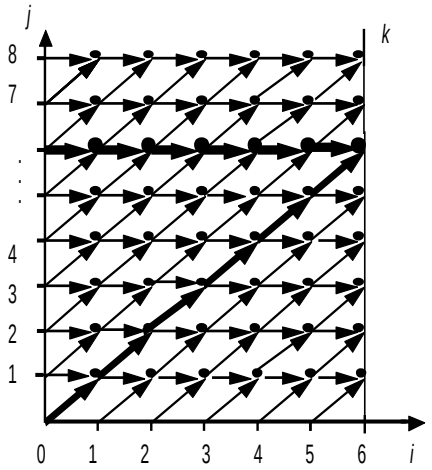

Рис. 4

Запис $DO \;\; SYNCH(1)$ означає, що в кожній гілці $k \in [1,8]$ після виконання однієї ітерації по i здійснюється синхронізація. Відповідні «обрізані» піраміди зображені на рис. 5. Порівнюючи рис. 4 і рис. 5, бачимо, що при використанні конструкції (7) довжина кожної з гілок зменшилася на шість ітерацій (якщо не враховувати «крайових ефектів») у порівнянні з відповідними гілками, побудованими за (6).

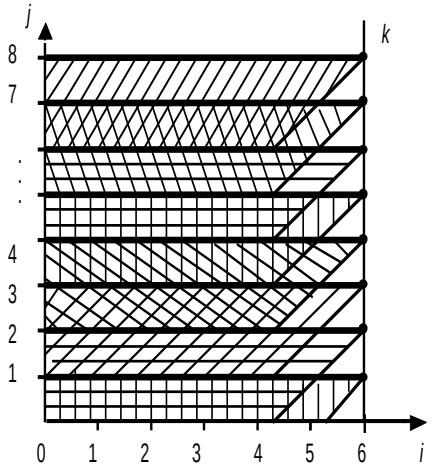

Рис. 5

Приклад 3.

Розглянемо інший цикл:

$$
\begin{aligned}
&FOR \;\; i = 1,7 \;\; DO\\
&FOR \;\; j = 1,6 \;\; DO\\
&\{ \;\; y(i, j) = g(y(i-2, j-1), y(i-1, j+1))\\
&\quad x(i, j) = f(x(i, j-1), x(i-1, j)) \;\; \}.
\end{aligned} \qquad (8)
$$

Тут g, f – функції, задані явно.

Граф залежностей між ітераціями даного циклу приведений на рис. 6. Оскільки ми маємо одну результуючу ітерацію, то є і одна піраміда, а отже, застосування звичайного методу пірамід не призведе до прискорення обчислень.

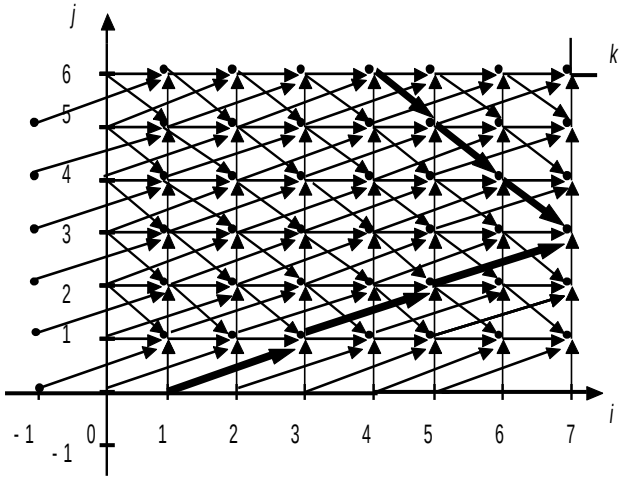

Рис. 6

Однак, якщо дану піраміду розбити на декілька пірамід із вершинами на $(7, j)$, $j = \overline{1,6}$, зсунувши їх одна відносно одної по осі i на одну координату, то можна записати деяку паралельну форму виконання циклу (8):

$$
\begin{aligned}
&FOR \quad k = 1,6 \quad DO \quad PIPE(i,1)\\
&FOR \quad i = 1,7 \quad DO\\
&FOR \quad j = \max\{1, [(i-1)/2] + k - 3\}, \min\{6, k - i + 7\} \quad DO\\
&\{ \ y(i, j) = g(y(i-2, j-1), y(i-1, j+1))\\
&\quad x(i, j) = f(x(i, j-1), x(i-1, j)) \ \}.
\end{aligned}
\tag{9}
$$

Запис $DO \quad PIPE(i,1)$ означає, що наступна ітерація циклу по k почне виконуватись, коли виконається ітерація циклу по i в попередній ітерації циклу по k. Для (8) можна навести й більш прийнятну паралельну конструкцію, якщо в (9) межі зміни індекса j визначити таким чином:

$$FOR \quad j = \max\{1, \ k-1, [(i-1)/2] + k - 3\}, \ \min\{6, \ k - i + 7\} \quad DO \, .$$

В результаті отримаємо паралельні гілки, в яких у середньому на три ітерації менше порівняно з гілками, описаними в (9).

Слід відзначити, що на практиці зустрічаються цикли, для яких характерним є те, що на одній і тій же ітерації змінна може приймати декілька значень. Тому використання в даному випадку паралельних конструкцій із взаємодією між гілками через спільну пам'ять є недоцільним, оскільки виникає можливість затирання потрібних на інших ітераціях значень змінних.

Вправа для самостійної роботи.

Використовуючи метод пірамід та його модифікації, побудувати відповідні паралельні конструкції для циклу:

$$FOR \;\; i = 1, 6$$
$$FOR \;\; j = 1, 7 \;\; DO$$
$$x(i, j) = x(i-1, j) + x(i-1, j-3) + x(i-2, j+1).$$