Практичне заняття № 5.
Розв'язання одновимірної задачі фільтрації з використанням методу пірамід для розпаралелювання циклів.

Послідовний алгоритм розв'язання задачі фільтрації.

Послідовний аналог паралельного алгоритму розв'язання одновимірної задачі цифрової фільтрації, що реалізує синхронну схему обчислень і розглянутий на практичному занятті № 2, має вигляд:

$$
\begin{aligned}
&FOR\ \ j=1,k_1\ \ DO\\
&\{\ FOR\ i=1,n\ \ DO\\
&\quad\{\ p=0\\
&\quad\ FOR\ s=-m,m\ \ DO\\
&\quad\ \{\ p=p+x_{i+s}^{j-1}*f_s\ \}\\
&\quad\quad x_i^j=p\ \}\ \}.
\end{aligned}
\tag{1}
$$

Результатом роботи алгоритму (1) є $x_i^{k_1},\ i=\overline{1,n}$. Тут вважаємо, що $x_{i'}^0=x_{i'}$ для всіх $i'=\overline{1-m,n+m}$. Тобто це є початкові значення змінних (до виконання першого переобчислення).

Застосування методу пірамід.

Розглянемо алгоритм (1) за таких значень параметрів задачі цифрової фільтрації: $m=1, n=10, k_1=5$. Для організації паралельного виконання його ітерацій $(i,\ j)$ використаємо метод пірамід. У цьому випадку простір ітерацій для (1) подано на рис. 1. Крім простору ітерацій, на цьому рисунку зображено залежності між окремими ітераціями (з допомогою стрілок) та трикутник (піраміду) для результуючої ітерації **(5, 5)**.

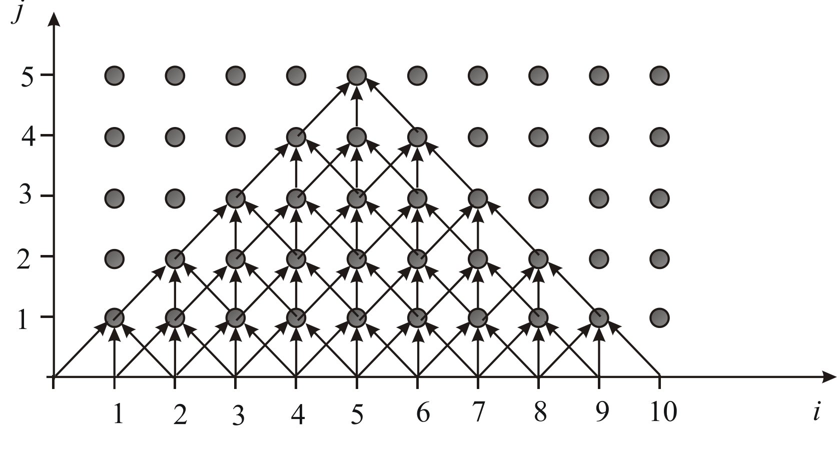

Рис. 1

Легко здогадатися, що у цьому випадку всі результуючі ітерації будуть розташовані на прямій $j=5$. Отже, кількість таких ітерацій для цього варіанту розв'язання задачі цифрової фільтрації дорівнює **10**, тобто всіх трикутників (пірамід) буде **10**. Відповідна паралельна конструкція для (1) за наведених вище значень параметрів задачі фільтрації матиме вигляд (2).

$$
\begin{aligned}
& FOR\ \ t = 1, 10\quad DO\quad PAR\\
\{\ \ & FOR\ \ j = 1,\ 5\ \ DO\\
\{\ \ & FOR\ \ i = \max\{1,\ \ j + t - 5\}, \min\{10,\ \ t - j + 5\}\quad DO\\
\{\ \ & p = 0\\
& FOR\ \ s = -m,\ m\ \ DO\\
& \{\ \ p = p + x_{i+s}^{j-1} * f_s\ \ \}\\
& \quad x_i^j = p\quad \}\quad \}\quad \}.
\end{aligned}
\tag{2}
$$

Легко обчислити, що за згаданих вище значень параметрів задачі фільтрації алгоритм (1) потребує виконання **50** ітерацій, а алгоритм (2) зводиться до виконання (за часом) **25** ітерацій (оскільки вибираємо трикутник (піраміду) з найбільшою кількістю ітерацій). Отже, прискорення паралельних обчислень буде дорівнювати 2 (S = 2).

Розглянемо алгоритм (1) для інших значень параметрів задачі цифрової фільтрації: $m = 2, n = 8, k_1 = 3$. Далі на рис. 2 графічно зобразимо простір ітерацій $(i,\ j)$ для (1) та залежності між деякими із них.

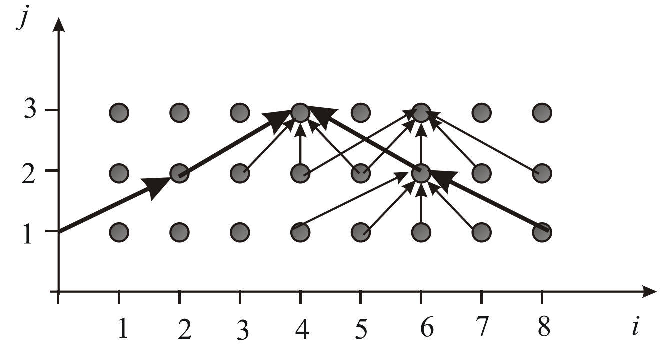

Рис. 2

На рис. 2 жирними стрілками виділено трикутник (піраміду) для результуючої ітерації **(4, 3)**. Усі результуючі ітерації в даному випадку розташовуються на прямій $j = 3$ і їх кількість дорівнює **8**. Відповідна паралельна конструкція для наведеного набору значень параметрів задачі фільтрації унаслідок використання методу пірамід для розпаралелювання циклів стосовно (1) матиме вигляд (3).

$$
\begin{aligned}
& FOR\ \ t = 1, 8\quad DO\quad PAR\\
\{\ \ & FOR\ \ j = 1,\ 3\ \ DO\\
\{\ \ & FOR\ \ i = \max\{1,\ \ 2j + t - 6\}, \min\{8,\ \ t - 2j + 6\}\quad DO
\end{aligned}
$$

$$
\begin{aligned}
&\{\quad p = 0 \\
&\quad FOR\quad s = -m,\ m\quad DO \\
&\quad \{\quad p = p + x_{i+s}^{j-1} * f_s\quad \} \\
&\qquad x_i^j = p\quad \}\quad \}\quad \}.
\end{aligned} \tag{3}
$$

За наведених значень параметрів задачі фільтрації ($m = 2,\ n = 8,\ k_1 = 3$) алгоритм (1) потребує виконання **24** ітерацій, а алгоритм (3) зводиться до виконання (за часом) **14** ітерацій. Отже, прискорення паралельних обчислень в даному випадку приблизно дорівнює 1.7 ($S \approx 1.7$).

На основі викладеного запишемо паралельну конструкцію для алгоритму (1) у загальному вигляді, використовуючи ідеї методу пірамід для розпаралелювання циклів:

$$
\begin{aligned}
&\quad FOR\quad t = 1,\ n\quad DO\quad PAR \\
&\{\quad FOR\quad j = 1,\ k_1\quad DO \\
&\{\quad FOR\quad i = \max\{1,\ m(j-k_1)+t\}, \min\{n,\ m(k_1-j)+t\}\quad DO \\
&\{\quad p = 0 \\
&\quad FOR\quad s = -m,\ m\quad DO \\
&\quad \{\quad p = p + x_{i+s}^{j-1} * f_s\quad \} \\
&\qquad x_i^j = p\quad \}\quad \}\quad \}.
\end{aligned} \tag{4}
$$

Оцінка прискорення обчислень.

У загальному випадку алгоритм (1) потребує виконання

$$nk_1$$

ітерацій, а алгоритм (4) зводиться до виконання (за часом)

$$k_1(1 + m(k_1 - 1))$$

ітерацій. Звідси одержуємо формулу для прискорення обчислень за алгоритмом (4):

$$S = n/(1 + m(k_1 - 1)).$$

З цієї формули випливає, що у разі $n >> m,\ n >> k_1$ прискорення S є суттєвим. Наприклад, для $m = 2,\ n = 100,\ k_1 = 5$ одержуємо, що

$$S \approx 11.1,$$

а для значень параметрів задачі фільтрації $m = 3,\ n = 1000,\ k_1 = 10$ одержуємо:

$$S \approx 35.7.$$

Алгоритми з обмеженим паралелізмом.

Побудований паралельний алгоритми (4) задає виконання n паралельних гілок і, звісно ж, потребує для своєї реалізації n окремих обчислювальних пристроїв (процесорів). Однак, в реальних обчислювальних системах, зокрема кластерах, кількість одночасно працюючих процесорів (ядер) є обмеженою та наперед визначеною. Тому для розв'язання задач фільтрації доцільніше розглядати алгоритми з обмеженим паралелізмом, в яких кількість паралельно виконуваних гілок P є меншою за n ($P < n$). При цьому для спрощення подальшого викладу будемо вважати, що n є кратним до P.

У разі обмеження кількості паралельно виконуваних гілок алгоритм (4) подаємо у вигляді (5).

$$
\begin{array}{l}
\quad FOR \;\; t=1,\ P \quad DO \quad PAR \\
\{ \quad FOR \;\; j=1,\ k_1 \quad DO \\
\{ \quad FOR \;\; i=\max\{1,\ m(j-k_1)+(t-1)n/P+1\}, \\
\qquad\qquad\qquad\qquad \min\{n,\ m(k_1-j)+tn/P\} \quad DO \\
\{ \quad p=0 \\
\quad FOR \;\; s=-m,\ m \quad DO \\
\quad \{ \quad p=p+x_{i+s}^{j-1} * f_s \;\; \} \\
\qquad x_i^j = p \quad \} \quad \} \quad \}.
\end{array}
\tag{5}
$$

Аналогічно, як і для (4), у даному випадку можна покращувати процес згладжування значень змінних. Для цього потрібно останні чотири рядки в (4), (5) замінити фрагментом:

$$
\begin{array}{l}
\{ \quad p_i = 0 \\
\qquad FOR \;\; s=-m,\ m \quad DO \\
\{ \quad p_i = p_i + x_{i+s} * f_s \quad \} \\
\qquad x_i = p_i \quad \} \quad \} \quad \}.
\end{array}
$$

Для свого виконання паралельний алгоритм (5) потребує (за часом) обчислення

$$k_1(n/P + m(k_1-1))$$

ітерацій. Звідси отримуємо, що прискорення паралельних обчислень у даному разі обчислюється за формулою

$$S = n/(n/P + m(k_1-1)) = 1/(1/P + m(k_1-1)/n)\,.$$

У разі $n >> m,\ n >> k_1$ отримуємо, що $S \approx P$, тобто прискорення набуває свого оптимального значення.

Вправи для самостійної роботи.

1. Графічно зобразити простір ітерацій $(i,\ j)$ для алгоритму (1) у випадку $m=3,\ n=15,\ k_1=4$. Стрілками показати всі залежності для окремих ітерацій. Для результуючої ітерації (7, 4) зобразити трикутник (піраміду), «вирізаний» із простору ітерацій.

2. Обчислити кількість арифметичних операцій, необхідних для виконання алгоритмів (1) та (4) у разі $m=2,\ n=12,\ k_1=6$. Для цього випадку знайти прискорення паралельних обчислень.