УДК 519.681.5

М. С. Яджак

АНАЛІЗ РЕАЛІЗАЦІЇ АЛГОРИТМІВ З ОБМЕЖЕНИМ ПАРАЛЕЛІЗМОМ ДЛЯ ЦИФРОВОЇ ФІЛЬТРАЦІЇ

The means and schemes of realization on universal parallel-calculating-systems of some algorithms with limited parallelism for execution of one-dimensional digital filtering are proposed. The analysis of this realization is executed.

Запропоновано способи та схеми і виконано аналіз реалізації деяких алгоритмів з обмеженим паралелізмом для виконання одновимірної цифрової фільтрації на універсальних обчислювальних системах паралельної дії.

Огляд та формулювання проблеми. Для виконання процедур цифрової фільтрації у більшості випадків використовують швидкі алгоритми обчислень [6]. Робота [7] започаткувала цикл публікацій, присвячених побудові оптимальних за швидкодією у заданому класі алгоритмів розв'язання задач цифрової фільтрації (ЗЦФ) різної вимірності [1, 3]. Зазначені алгоритми були зорієнтовані на реалізацію на квазісистолічних обчислювальних структурах. Для виконання процедури одновимірної цифрової фільтрації на універсальних обчислювальних системах в [4] запропоновані алгоритми з обмеженим паралелізмом (АОП). У праці [5] для двох АОП, побудованих за умов $1 < p < n,\ m = 1,\ n/p \geq 3$, досліджена проблема їх реалізації на обчислювальних системах паралельної дії зі спільною та розподіленою пам'яттю. Тут n – кількість перераховуваних значень змінних, $(2m+1)$ – розмір рухомого вікна під час виконання процедури цифрової фільтрації, а p – кількість паралельних гілок.

Ця робота є логічним продовженням [5].

Вирішувана проблема полягає в аналізі архітектур універсальних паралельних обчислювальних систем і розробці на його підставі способів та схем реалізації АОП, побудованих у разі виконання умов $1 < p < m+1,\ m > 1$ та $p = lm,\ m > 1,$ $l \in \mathrm{N} \setminus \{1\}$ або ж $p = l(m+1),\ m > 1,\ l \in \mathrm{N}$.

Реалізація АОП на обчислювальних системах зі спільною пам'яттю. Як було відзначено в [5], за умови, коли вагові коефіцієнти $f_s (s = \overline{-m, m})$ є однаковими і дорівнюють f, узагальнена конструкція АОП має вигляд:

$$\begin{array}{l} FOR\ \ v = 1, p\ \ DO\ \ SYNCH \\ \quad DELAY\ (D(v)) \\ \quad\quad FOR\ \ j = 1, C\ \ DO \\ \quad\quad\quad T(v). \end{array} \qquad (1)$$

Тут $D(v)$ – величина зсуву (в тактах) між першою та v-ю гілками, C – кількість виконуваних перерахунків згладжування масиву значень змінних $x_i (i = \overline{1, n})$, а $T(v)$ – деяке тіло циклу. Вигляд $T(v)$ та виразу для обчислення $D(v)$ залежить від розглядуваного випадку задання p та m.

Представимо тіло циклу $T(v)$ фрагментом:

$$\begin{array}{l} FOR\ \ i = A(v), B(v), E\ \ DO \\ \quad BEGIN \end{array}$$

$$FOR\ \ s=1,\ 2m\ \ DO$$

$$x_i = x_i + IF\ \ (s/2=[s/2])\ \ THEN\ \ x_{i+s/2-m-1}\ \ ELSE\ \ x_{i+(s+1)/2} \qquad (2)$$

$$x_i = x_i * f$$

$$DELAY\ (G)$$

$$END\,.$$

Зауважимо, що тут і надалі $[a]$ означає цілу частину числа a.

Нехай виконуються умови: $1<p<m+1,\ m>1$. Тоді для $D(v)=2v-2$, $A(v)=v,\ B(v)=v+p(n/p-1),\ E=p,\ G=0$ на підставі (1), (2) одержуємо АОП, який позначатимемо $A2_0$. Останній характерний тим, що кількість паралельних гілок в ньому є набагато меншою за кількість перераховуваних значень змінних [4], оскільки на практиці зазвичай $m<<n$. Окрім цього, гілки згаданого алгоритму пов'язані частими обмінами даними, тобто є сильнозв'язаними [2].

За умов $p=lm,\ \ m>1,\ \ l\in \mathrm{N}\setminus\{1\}$ для $D(v)=2v-2+w_p(v)(2m+1)$, $A(v)=v+w_p(v)m$, $B(v)=v+(w_p(v)+g_p)m$, $E=m$, $G=0$ на підставі (1), (2) одержуємо АОП $A3_0$. Тут $g_p=(n/p)-1,\ \ w_p(v)=\left[(v-1)/m\right]g_p$. Гілки даного алгоритму пов'язані обмінами даними.

Якщо припустити, що вагові коефіцієнти є різними та виконується умова $[p/2]=p/2$, і в $A2_0$ та $A3_0$ замінити p на $p/2$, четвертий та п'ятий оператори конструкції (2) представити фрагментом

$$BEGIN$$

$$x_i = IF\ \ (s=1)\ \ THEN\ \ f_0 * x_i\ \ ELSE\ \ x_i = y_i\,, \qquad (3)$$

$$y_i = f_{h(s,m)} * x_{i+h(s,m)}$$

$$END$$

і збільшити на одиницю довжину циклу за змінною s, то одержимо АОП, які позначатимемо відповідно $A2_r$ та $A3_r$. Зауважимо, що в (3) оператори $x_i=\ldots$ та $y_i=\ldots$, розділені комою, виконуються синхронно, а функція $h(s,m)$ є цілочисельою і визначена в [4].

У разі, коли $p=l(m+1),\ \ m>1,\ \ l\in \mathrm{N}$ АОП $A4_0$ одержуємо із $A3_0$, якщо в останньому зробити такі виправлення [4]:

– у другому операторі, який виконує затримку, m замінити на m+1 і 2m+1 – на 2m+2;

– в операторі, що задає цикл за змінною i, m замінити на m+1;

– перед оператором END вставити умовний оператор:

$$IF\ \ \left(i=v+\left(\frac{n}{p}-1\right)(m+1)\left(\left[\frac{v-1}{m+1}\right]+1\right)\right)\ \ THEN\ \ DELAY\ (0)\ \ ELSE\ \ DELAY\,(1)\,.$$

Аналогічно, роблячи відповідні поправки в $A3_r$, ми одержимо для цього випадку АОП $A4_r$ за умови, що вагові коефіцієнти є різними. Зауважимо, що обидва алгоритми $A4_0$, $A4_r$ працюють і у випадку, коли m=1, до того ж, коли $n/p=2$. Паралельні гілки цих алгоритмів теж пов'язані обмінами даними, а самі алгоритми дозволяють виконати перший перерахунок значень змінних за той самий час, що і відповідні оптимальні за швидкодією алгоритми [7].

Найпростішим варіантом реалізації обмінів між гілками наведених АОП є використання багатопроцесорної обчислювальної системи зі спільною пам'яттю (БОССП). Кожен із p однакових паралельно працюючих процесорних елементів (ПЕ) такої системи має прямий і рівноправний доступ до будь-якої точки пам'яті. Після виконання кожної операції робота ПЕ синхронізується. Для уникнення конфліктів з пам'яттю в такій системі необхідно передбачити засоби розмноження інформації, щоб декілька процесорів могли одночасно одержати інформацію з однієї комірки пам'яті [5]. Для оцінки прискорення АОП за такої реалізації використовуємо відповідні оцінки, наведені в [4], враховуючи, що для класичних БОССП (*SMP*-системи) кількість ПЕ не перевищує тридцяти двох [8], хоча є відомості [9] про розробку системи такого типу *SGI Altix* на основі 1024 двоядерних процесорів *Itanium* 2. На основі викладеного одержуємо, що прискорення алгоритму $A2_0$ дорівнює своєму оптимальному значенню, тобто p, а прискорення $A2_r$ є близьким до свого оптимального значення (наприклад, для $m = 2$ воно дорівнює $0{,}9p$, а для $m = 5$ – $0{,}954p$).

Прискорення алгоритмів $A3_0$, $A3_r$ для різних значень p, m, C та $n = 100000$ наведене відповідно в табл. 1, 2, звідки видно, що за фіксованих значень p воно зростає із збільшенням m та C.

Таблиця 1. Прискорення алгоритму $A3_0$ для n = 100000

p	m	C = 4	C = 8
4	2	3,1999	3,5555
8	2	4,5714	5,8181
8	4	6,3999	7,1110
16	2	5,8182	8,5334
16	4	9,1427	11,6363
32	4	11,6364	17,0667
32	8	18,2851	23,2722

Таблиця 2. Прискорення алгоритму $A3_r$ для n = 100000

p	m	C = 4	C = 8
8	2	5,7599	6,3999
16	2	8,2285	10,4727
16	4	12,0887	13,4320
32	4	17,2697	21,9796
32	8	24,8464	27,6074

Як було зазначено в [4], за умови $C < p/m + 1$ алгоритм $A4_0$ є швидшим порівняно з $A3_0$, а у разі $C < p/(2m) + 1$ $A4_r$ працює швидше, ніж $A3_r$. Прискорення алгоритмів $A3_0$, $A4_0$, $A3_r$, $A4_r$ у разі виконання наведених нерівностей для різних значень p, m, C та $n = 240000$ наведене в табл. 3, 4. Звідси випливає, що використання $A4_0$, $A4_r$ у даному разі призводить до незначного прискорення обчислень порівняно відповідно з $A3_0$, $A3_r$.

Таблиця 3. Прискорення алгоритмів $A3_0$, $A4_0$ для C = 4, n = 240000

p	m	$A3_0$	$A4_0$
12	2	5,3333	5,7142
12	3	6,8571	6,9999
20	4	9,9999	10,2856
24	2	6,4000	7,2727
24	3	8,7273	9,3333

Таблиця 4. Прискорення алгоритмів $A3_0$, $A3_r$, $A4_0$, $A4_r$ для n = 240000, p = 24

m	C = 4		C = 8	
	$A3_r$	$A4_r$	$A3_0$	$A4_0$
2	9,6000	10,2857	10,1053	10,6667
3	12,7346	12,9999	12,8000	12,9231

Реалізація АОП на обчислювальних системах з розподіленою пам'яттю. Із потактової схеми АОП $A3_0$ [4] легко бачити, що для заданого m збільшення кількості змінних, значення яких перераховується в кожній з паралельних гілок, не призводить до збільшення кількості обмінів між визначеними l групами гілок. Отже, стосовно таких груп гілок є можливість збільшення частки обчислювальних операцій порівняно з обмінними. Тому алгоритм $A3_0$ пропонується реалізувати на обчислювальній системі з розподіленою пам'яттю (ОСРП) – кластері, що складається з l вузлів, з'єднаних між собою комунікаційною мережею. Кожен вузол утворюють m ПЕ, які працюють над спільною пам'яттю (пам'яттю вузла), тобто є прототипом *SMP*-систем. В пам'яті будь-якого j-го вузла зберігаються перераховувані ним значення mn/p змінних, вихідні m наступних значень змінних, які перераховуються в $(j+1)$-му вузлі (для l-го вузла це є значення x_{n+1}, x_{n+2},..., x_{n+m}), кількість виконуваних перерахунків C та ваговий коефіцієнт f. Зауважимо, що x_{1-m}, x_{2-m},..., x_0 зберігаються у пам'яті першого вузла. Під час реалізації АОП $A3_0$ запуск i-го ПЕ вузла здійснюється із зсувом стосовно першого елемента цього ж вузла на $2(i-1)$ такт, а запуск j-го вузла кластера відбувається порівняно з першим вузлом на $(2m+(n/p-1)(2m+1)+T_0)(j-1)$ такт пізніше, де T_0 – кількість тактів, необхідних для передачі даного із j_1-го вузла в (j_1+1)-й або навпаки. Тут і надалі вважаємо, що час виконання операцій додавання та множення є однаковим і дорівнює одному такту, а операції з пам'яттю в кожному з вузлів не впливають на тривалість виконуваних обчислень. Для забезпечення суміщення обмінних операцій (передача даних із (j_1+1)-го в j_1-й вузол) із обчислювальними вимагається виконання умови $n/p \geq (2T_0+2m+2)/(2m+1)$. Для реалізації на описаній паралельній обчислювальній системі $A3_0$ потребує

$$(C+l-1)(2m+1)n/p+2m-l-1+T_0(l-1)$$

такт. Розв'язання одновимірної ЗЦФ за умови рівності вагових коефіцієнтів в послідовному режимі [4] вимагає $nC(2m+1)$ такт. Зазвичай, на практиці $n>>m$. Вважаючи, що $T_0<<n$, у разі $C=l_0(l-1)$, де $l_0 \in \mathrm{N}$, на підставі наведених оцінок прискорення S_0 АОП $A3_0$ обчислюється за формулою

$$S_0 \approx l_0 p/(l_0+1),$$

а унаслідок виконання умови $l-1=l_1C$, де $l_1 \leq 2m+1$, маємо:

$$S_0 \approx p/(l_1+1).$$

Аналогічно, як і в $A3_0$, в АОП $A3_r$ є можливість збільшення частки обчислювальних операцій порівняно з обмінними стосовно $l/2$ груп паралельних гілок. Тому для реалізації $A3_r$ пропонується використати кластер, що складається з $l/2$ обчислювальних вузлів. Кожен вузол утворюють m пар ПЕ, що працюють над спільною пам'яттю. В пам'яті кожного j-го вузла зберігаються перераховува-

ні ним значення $2mn/p$ змінних, вихідні m наступних значень змінних, що перераховуються в $(j+1)$-му вузлі, кількість виконуваних перерахунків C та вагові коефіцієнти f_s $(s=\overline{-m,m})$. Зауважимо, що значення x_{1-m}, x_{2-m},..., x_0 зберігаються у пам'яті першого вузла. Під час реалізації $A3_r$ запуск i-ї пари ПЕ вузла здійснюється із зсувом стосовно першої пари на $2(i-1)$ такт, а запуск j-го вузла кластера порівняно з першим вузлом відбувається на $(2m+(2n/p-1)(2m+1)+$ $+T_0)(j-1)$ такт пізніше. Для забезпечення суміщення обмінних операцій (передача даних із (j_1+1)-го в j_1-й вузол) із обчислювальними вимагається виконання умови $2n/p \geq (2T_0+2m+2)/(2m+1)$. Для виконання $A3_r$ на описаній паралельній обчислювальній системі необхідно

$$(C+l/2-1)(4m+2)n/p+2m-l/2-1+T_0(l/2-1)$$

такт. Розв'язання одновимірної ЗЦФ у разі, коли вагові коефіцієнти є різними, в послідовному режимі [4] потребує $nC(4m+1)$ такт. У разі $C=l_2(l/2-1)$, де $l_2 \in \mathrm{N}$, прискорення S_r алгоритму $A3_r$ обчислюється за формулою

$$S_r \approx l_2 p\ (4m+1)/((4m+2)(l_2+1)),$$

а, припустивши, що $l/2-1=l_3C$, де $l_3 \leq 4m+1$, одержимо

$$S_r \approx p(4m+1)/((4m+2)(l_3+1)).$$

Зауважимо, що оцінки для S_r наведено з урахуванням умов: $n >> m$, $T_0 << n$.

Аналогічні міркування до наведених стосовно АОП $A3_0$, $A3_r$ створюють підстави для розгляду реалізації алгоритмів $A4_0$, $A4_r$ на паралельних ОСРП – кластерах. Щоб не повторюватися, зупинимось лише на основних відмінностях кластера, на якому пропонується реалізувати АОП $A4_0$, від описаної вище обчислювальної системи для виконання $A3_0$. У кожному вузлі (його утворюють $m+1$ ПЕ, що працюють над спільною пам'яттю) пропонованого кластера перераховуються значення $(m+1)n/p$ змінних. Запуск j-го вузла здійснюється порівняно з першим на $(2(m+1)+(n/p-1)(2m+2)+T_0)(j-1)$ такт пізніше. Суміщення обмінних операцій із обчислювальними забезпечується виконанням умови $n/p \geq$ $\geq (2T_0+2m+3)/(2m+2)$. Для такої реалізації алгоритм $A4_0$ потребує

$$(C+l-1)(2m+2)n/p+2m-C+T_0(l-1)$$

такт. Враховуючи, що $n >> m$, $T_0 << n$, у разі $C=l_0(l-1)$, $l_0 \in \mathrm{N}$ одержуємо, що прискорення S_{00} АОП $A4_0$ обчислюється за формулою

$$S_{00} \approx l_0 p(2m+1)/((l_0+1)(2m+2)),$$

а унаслідок виконання умови $l-1=l_1C$, $l_1 \leq 2m+1$, маємо:

$$S_{00} \approx p(2m+1)/((l_1+1)(2m+2)).$$

Наведемо основні відмінності кластера, на якому пропонується реалізувати АОП $A4_r$, від описаної раніше обчислювальної системи для виконання $A3_r$. У кожному вузлі (його утворюють $m+1$ пара ПЕ, що працюють над спільною пам'яттю) кластера, що пропонується, перераховуються значення $2(m+1)n/p$ змінних. Запуск j-го вузла здійснюється порівняно з першим вузлом на $(2(m+1)+$

$+(2n/p-1)(2m+2)+T_0)(j-1)$ такт пізніше. Суміщення обмінних операцій із обчислювальними забезпечується виконанням умови $2n/p \geq (2T_0+2m+3)/(2m+$ $+2)$. Для реалізації алгоритму $A4_r$ на такій системі необхідно

$$(C+l/2-1)(4m+4)n/p+2m-C+T_0(l/2-1)$$

такт. Враховуючи наведені вище умови, що пов'язують величини n, m, T_0, у разі $C=l_2(l/2-1)$, $l_2 \in \mathrm{N}$ прискорення S_{rr} АОП $A4_r$ обчислюється за формулою

$$S_{rr} \approx l_2 p(4m+1)/((l_2+1)(4m+4)),$$

а, припустивши $l/2-1=l_3C$, $l_3 \leq 4m+1$, одержимо

$$S_{rr} \approx p(4m+1)/((l_3+1)(4m+4)).$$

ВИСНОВКИ

У роботі наведено уточнені чисельні оцінки прискорення АОП із [4] у разі їх реалізації на БОССП. Описано схеми та здійснено аналіз виконання на ОСРП (кластерах) алгоритмів $A3_0$, $A3_r$, $A4_0$ та $A4_r$. Подані в роботі оцінки прискорення обчислень підтверджують доцільність виконання відповідних алгоритмів на розглядуваних універсальних паралельних обчислювальних системах. Одержані результати можуть бути використані під час реалізації АОП для розв'язання ЗЦФ більшої вимірності.

1. *Вальковский В. А., Яджак М. С.* Оптимальный алгоритм решения двумерной задачи цифровой фильтрации // Проблемы управления и информатики. – 1999. – № 6. – С. 92–102.
2. *Штейнберг Б. Я.* Математические методы распараллеливания рекуррентных циклов для суперкомпьютеров с параллельной памятью. – Ростов н / Д: Изд-во Ростов. ун-та, 2004. – 192 с.
3. *Яджак М. С.* Об оптимальном в одном классе алгоритме решения трёхмерной задачи цифровой фильтрации // Проблемы управления и информатики. – 2000. – № 6. – С. 66–81.
4. *Яджак М. С.* Алгоритмы с ограниченным параллелизмом для решения одной задачи цифровой фильтрации // Проблемы управления и информатики. – 2001. – № 6. – С. 109–118.
5. *Яджак М. С.* Реалізація алгоритмів з обмеженим паралелізмом для розв'язання задачі цифрової фільтрації // Відбір і обробка інформації. – 2006. – Вип. 25 (101). – С. 103–108.
6. *Яцимірський М. М.* Швидкі алгоритми ортогональних тригонометричних перетворень. – Львів: Академічний Експрес, 1997. – 219 с.
7. *Valkovskii V. A.* An optimal algorithm for solving the problem of digital filtering // Pattern Recognition and Image Analysis. – 1994. – **4**, № 3. – P. 241–247.
8. www.parallel.ru
9. www.sgi.com/company_info/newsroom/press_releases/2007/july/nasa.html

Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, Львів

Одержано 20.09.2008