УДК 621.398:007

А.А. ЯРОВИЙ

ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ ВИСОКОПРОДУКТИВНИХ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

Вінницький національний технічний університет
95, Хмельницьке шосе, Вінниця, 21021, Україна
Тел.: +380 (432) 580019, e-mail: axa@vinnitsa.com

Анотація. В проведених дослідженнях здійснено аналіз особливостей організації високопродуктивних паралельно-ієрархічних обчислювальних процесів на базі різних апаратних платформ (CPU і GPU) та технологій. На основі отриманих результатів моделювання розроблено програмний комплекс призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами, що реалізує переваги технології GPGPU.

Аннотация. В проведенных исследованиях осуществлен анализ особенностей организации высокопроизводительных параллельно-иерархических вычислительных процессов на базе разных аппаратных платформ (CPU и GPU) и технологий. На базе полученных результатов моделирования разработан программный комплекс предназначенный для реализации прямого параллельно-иерархического преобразования с оптимизацией формирования масок над информационными массивами, который реализует преимущества технологии GPGPU.

Abstract. The analysis of peculiarity of high-performance parallel-hierarchical computational processes organization on the base of various hardware platform (CPU and GPU) and technologies is carry out in researches. In consideration of modeling results, the software package intended for realization of direct parallel-hierarchical transformation with optimization of mask generation at data arrays, which realized benefit of GPGPU technology, is developed.

Ключові слова: паралельні обчислення, паралельно-ієрархічне перетворення, технології GPGPU, CUDA, системи штучного інтелекту, обробка зображень, розпізнавання образів.

ВСТУП

Незважаючи на більше ніж піввіковий штурм проблеми створення систем штучного інтелекту (СШІ) на базі комп'ютерних систем і наявність задовільних цікавих окремих теоретичних результатів, навіть формування основних концепцій формального подання властивостей інтелектуального поводження, вона ще дуже далека від свого завершення. У цьому зв'язку слід відзначити доповідь професора Хоггарда з Кембриджа в Білому домі, який вважає, що створення штучного інтелекту є найважливішою науковою проблемою нашого тисячоліття. У своїй науковій роботі автором досліджується на рівні моделі одна з важливих проблем СШІ у формулюванні професорів Ю.В. Капітонової та В.І. Скуріхіна [1]. Це проблема створення моделі "сприйняття об'єктів і ситуацій зовнішнього світу", яка в проведених наукових дослідженнях розглядається відповідно до її використання в різноманітних прикладних задачах. При формуванні загальної методології створення високопродуктивних паралельно-ієрархічних обчислювальних процесів (ВПІОП) в нейроподібних системах враховувалась ідея, опублікована в наукових працях професора З.Л. Рабиновича про ієрархічну організацію зв'язків між структурами головного мозку людини. Сутність цієї ідеї ґрунтується на тому, що "передача збудження між структурами, тобто їх активація, може відбуватись не лише за вертикальними (ієрархічним прямим та зворотнім) зв'язками, але також за горизонтальними – в межах одного і того ж поля" [2].

Загалом, проблема створення ефективного системного аналізу, що дозволив би змінювати структурну ієрархію, виступає на перший план при розробці перспективних обчислювальних систем.

Проте не завжди, навіть намагаючись застосовувати ієрархічні структури, враховувалася найважливіша особливість ієрархічності як закономірності, яка полягає в тому, що закономірність цілісності виявляється на кожному рівні ієрархії. Завдяки цьому на кожному рівні виникають нові

властивості, які не можуть бути виведені як сума властивостей елементів. При цьому важливо, що не тільки об'єднання елементів у кожному вузлі призводить до появи нових властивостей, яких у них не було, і втраті деяких властивостей елементів, але і що кожний підпорядкований член ієрархії набуває нової властивості, відсутньої в нього в ізольованому стані. На кожному рівні ієрархії відбуваються складні якісні зміни. Саме завдяки цій особливості можна одержати цікаві результати, які завжди корисні при застосуванні системних представлень як засобу дослідження складних об'єктів і процесів.

Найбільш поширеним і досліджуваним способом представлення ієрархічних структур є деревоподібна структура [3,4]. Ідея методу дерева (дерева ланцюгів) уперше була запропонована Черчменом у зв'язку з проблемами прийняття рішень у промисловості. У теорії кодування дана ідея знайшла своє втілення при організації кодового дерева (коди Шеннона-Фано, Хаффмена). У програмуванні деревоподібна структура широко використовувалася вперше Д. Кнутом [5]. Для аналізу зображень – академіком В.В.Александровим [6], у структурах паралельної пам'яті – професором. Е.А. Метлицьким [7]. У теорії штучного інтелекту деревоподібні структури, подані як схеми знань, вперше запропоновані Р. Куілліаном [8]. У 1974 році М. Мінський із МТІ висловив припущення, що людський розум інтегрує кожний новий об'єкт, зокрема мовний, за засобами особливих структур пам'яті, названими фреймами [9]. Іншим представником семантичного підходу до представлення знань є Р. Шенк, який розробив деревоподібну схему представлення знань, що має назву "скриптів", або сценаріїв [10].

Наведені області застосування деревоподібних структур – найбільш концептуальні і поширені. Під терміном “дерево за Черчменом” розуміють використання ієрархічної структури, отриманої шляхом поділу загальної цілі на підцілі, а їх у свою чергу, на більш детальні складові – нові підцілі, функції і т.д. Іноді замість дерева цілей зручніше говорити про дерево проблем або про дерево напрямків прогнозування [11]. Зокрема, В.М. Глушковим у 1975 році запропонований і в даний час широко використовується прогнозний граф. В усіх зазначених областях застосування деревоподібних структур ієрархічний опис – це за суттю в тих же термінах декомпозиція цілі в просторі. Іншим не менш важливим різновидом ієрархічних структур на думку автора можуть стати такі, що сполучать декомпозицію цілі в просторі і часі. В дослідженні вони виділені окремим класом – просторово-часові ієрархічні структури [12]. У загальному вигляді метод розгортання ієрархічних структур полягає в просторово – багаторівневому представленні даних і в часово – мережному їх принципі оброблення. В дослідженні розглядалися дві групи просторово-часових ієрархічних структур – пірамідальні та паралельно-ієрархічні, відповідно із сильними і слабкими зв'язками. Причому, дані групи досліджувались на модельному, алгоритмічному, структурно-функціональному, системному і схемотехнічному рівнях [13, 14]. У загальному вигляді концепцію багатоетапності стосовно предметної області обробки зображень можна сформулювати так. Аналіз зображення полягає в послідовному перетворенні збіжних і виявленні (фільтрації) незбіжних у часі його складових при переході елементів нейроподібної паралельно-ієрархічної мережі з поточних енергетичних станів з одними просторовими координатами в стани з меншою енергією з іншими просторовими координатами. Такий процес аналізу зображення відбувається на багатьох етапах, кожний із яких включає виконання вищевказаної процедури. Умовою переходу складової зображення на більш високий рівень є наявність динаміки взаємного збігу проміжних результатів обробки в часі в паралельних каналах нижнього рівня. Результат аналізу зображення формується з ізольованих у просторово-часовій області його складових [15, 16].

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є аналіз особливостей організації високопродуктивних паралельно-ієрархічних обчислювальних процесів при реалізації алгоритмів в межах розроблених програмно-апаратних засобів на базі потужностей як центрального процесора (CPU), так і графічних спецпроцесорів відеоадаптера (на основі технологій GPGPU) для подальшого ефективного багатоетапного сприйняття, збереження, ущільнення і розпізнавання зображень.

ОПИС МАТЕМАТИЧНОЇ МОДЕЛІ ТА ПРОЦЕСУ ОРГАНІЗАЦІЇ ОБЧИСЛЕНЬ У ПАРАЛЕЛЬНО-ІЄРАРХІЧНОМУ ПЕРЕТВОРЕННІ

У попередніх роботах вже було розглянуто мережний метод паралельно-ієрархічного (ПІ) перетворення, і в тому числі на рівні моделей [12, 13, 17, 18]. Для кращого подальшого розуміння організації обчислювального процесу, наведемо основні вирази математичної моделі. Нехай є S ($S=1,2,3,\ldots$) непустих множин елементів, що задають інформацію. Кількість елементів множини є її довжиною, тобто L_μ – довжина множини μ. Кількість різноманітних елементів множини є розмірністю даної множини (R_μ). Математична модель пірамідального розкладання множини $\mu = \{ a_i \}$, $i = \overline{1, n}$ має

вигляд [12, 17, 18]:

$$\sum_{i=1}^{n} a_i = \sum_{j=1}^{R} \left(n - \sum_{k=0}^{j-1} n_k \right) \left(a^j - a^{j-1} \right), \tag{1}$$

де R – розмірність даної множини; $a_i \neq 0$. З однакових елементів сформуємо підмножини, елементи однієї підмножини позначимо через a^k, $k = \overline{1,R}$, відповідно n_k – кількість елементів у k-тій підмножині (тобто кратність числа a_k), a^j – довільний елемент множини $\{a^k\}$, обраний на j -му кроці, $j = \overline{1,R}$, $a^0 = 0$, $n_k = 0$. Аналізуючи вираз пірамідальної обробки числової інформації (1) можна зробити висновок про те, що у процесі обробки з кожним кроком кількість чисел зменшується. Якщо множини одержувані після кожного кроку поставити послідовно одну на іншу, то утворений ними тривимірний контур буде мати форму піраміди. Перетворенням множини μ в множину μ^1, що задається моделлю (1), є оператор перетворення G [12,17,18]:

$$G(\mu) = \mu^1. \tag{2}$$

Якщо для вихідних S масивів (S – непусті множини елементів, що задають інформацію) застосуємо оператор перетворення G, що задається виразом (1), то для кожного масиву отримуємо свій порядковий розклад [12,17,18]:

$$\mu_1^1 = \bigcup_{i=1}^{R_1^1} a_{1i}^i \ , \ \ \mu_2^1 = \bigcup_{i=1}^{R_2^1} a_{2i}^i \ , \ \ \ldots \ , \ \mu_s^1 = \bigcup_{i=1}^{R_s^1} a_{si}^i \ , \tag{3}$$

де μ_s^1 – множина з номером S на першому рівні, тоді для k-го рівня множина з номером l записується μ_l^k, відповідно R_s^1 – кількість елементів у S множині на першому рівні, R_l^k – кількість елементів у S множині на k-му рівні.

Мережний метод прямого паралельно-ієрархічного перетворення полягає в послідовному застосуванні до початкових множин $\bigcup_{s=1}^{S} \mu_S$ по одному разу операторів перетворення G і транспонування T, а потім $(k-1)$ раз функціонала Φ [12,13,17,18]:

$$\Phi_{t=2}^{k} \left[T \left(G \left(\bigcup_{s=1}^{S} \left(\bigcup_{i=1}^{n} a_i \right) \right) \right) \right] = \bigcup_{t=2}^{k} a_{11}^t , \tag{4}$$

де a_{11}^k – вихідна інформація (хвостові елементи мережі) прямого паралельно-ієрархічного перетворення.

Тобто, послідовне застосування трьох операторів G, S, T формує функціонал Φ, тобто $\Phi(M) = T\left[S\left(G\left(M \right) \right) \right]$, де S – оператор зсуву рядка на величину меншу номера даного рядка на одиницю і виключення першого стовпчика матриці M_k в результат розкладу.

Опишемо узагальнений алгоритм G-перетворення рядка з утворенням масок:

1. Присвоюємо i=1.
2. Запам’ятовуємо маску H_i рядка μ_i.
3. Видаляємо всі нулі з μ_i, утворюючи μ_i'.
4. Знаходимо мінімальний елемент рядка μ_i' – m.
5. Знайдений мінімальний елемент m множимо на потужність μ_i' та утворюємо i-тий елемент в рядку з результатом G-перетворення.
6. Якщо m дорівнює кожному елементу μ_i', тоді кінець алгоритму.
7. Утворюємо рядок μ_{i+1} з μ_i', шляхом віднімання від кожного елемента μ_i' значення m.
8. Якщо $i=i+1$, то переходимо до пункту 1.

Для демонстрації організації процесу обчислень згідно до моделі (4) розглянемо числовий приклад прямого паралельно-ієрархічного перетворення, початкова інформація якого задана у вигляді числових множин μ_1, μ_2, μ_3 :

$$\mu_1 = \begin{pmatrix}2\\8\\2\end{pmatrix}; \qquad \mu_2 = \begin{pmatrix}4\\2\\5\end{pmatrix}; \qquad \mu_3 = \begin{pmatrix}1\\19\\3\end{pmatrix}$$

Сформуємо матрицю M_1 над якою будемо виконувати перетворення.

$$M_1 = \begin{pmatrix}\mu_1^T\\ \mu_2^T\\ \mu_3^T\end{pmatrix} = \begin{pmatrix}2 & 8 & 2\\ 4 & 2 & 5\\ 1 & 19 & 3\end{pmatrix},$$

де T – оператор транспонування.

Ітерація 1.

На першій ітерації виконують лише G-перетворення матриці M_1. Перетворення виконується над кожним вектором даних (рядком) окремо.

Відповідно до п.2 алгоритму:

μ_{1_1}	H_1	μ_{1_1}'	m	μ_{1_2}	m	μ_{1_2}'	m
2	0	2		0	1		
8	0	8	2	6	0	6	6
2	0	2	↓	0	1		↓
			6				***6***

μ_{2_1}	H_1	μ_{2_1}'	m	μ_{2_2}	H_2	μ_{2_2}'	m	μ_{2_3}	H_3	μ_{2_3}'	m
4	0	4		2	0						
2	0	2	2	0	1	2	2	0	1	1	1
5	0	5	↓	3	0	3	↓	1	0		↓
			6				***4***				***1***

μ_{3_1}	H_1	μ_{3_1}'	m	μ_{3_2}	H_2	μ_{3_2}'	m	μ_{3_3}	H_3	μ_{3_3}'	m
1	0	1		0	1						
19	0	19	1	18	0	18	2	16	0	16	16
3	0	3	↓	2	0	2	↓	0	1		↓
			3				***4***				***16***

Таким чином:

$$G(M_1) = \begin{pmatrix}6 & 6 & 0\\ 6 & 4 & 1\\ 3 & 4 & 16\end{pmatrix} = M_2$$

Для того, щоб матриця не містила невизначених елементів замість них ставимо 0. А маски, сформовані на цьому кроці, такі:

0	0	0
1	0	1
0	0	0
0	1	0
	1	0
0	0	0
1	0	0
	0	1

Маски записуються в порядку їх знаходження, тобто спочатку для першого рядка, потім для 2-го і т.д.

Ітерація 2.

Починаючи з другої ітерації, над матрицею M_2 виконуємо такі дії:

1. Транспонування (T)
2. G-перетворення (G)
3. Зсув (S)
4. Запам’ятовування хвостового елементу.

Продовжуємо виконувати перетворення:

$$M_2 = \begin{pmatrix}6 & 6 & 0\\ 6 & 4 & 1\\ 3 & 4 & 16\end{pmatrix}.$$

Транспонуємо матрицю M_2

$$T(M_2) = \begin{pmatrix} 6 & 6 & 3 \\ 6 & 4 & 4 \\ 0 & 1 & 16 \end{pmatrix}.$$

Виконуємо G-перетворення:

μ_{1_1}	H_1	μ_{1_1}'	m	μ_{1_2}	m	μ_{1_2}'	m
6	0	6		3	0		
6	0	6	3	3	0	3	3
3	0	3	↓	0	1	3	↓
			9				***6***

μ_{2_1}	H_1	μ_{2_1}'	m	μ_{2_2}	H_2	μ_{2_2}'	m
6	0	6		2	0		
4	0	4	4	0	1	2	2
4	0	4	↓	0	1		↓
			12				***2***

μ_{3_1}	H_1	μ_{3_1}'	m	μ_{3_2}	H_2	μ_{3_2}'	m
0	1						
1	0	1	1	0	1		15
16	0	16	↓	15	0	15	↓
			2				***15***

Тобто

$$G\big(T(M_2)\big) = \begin{pmatrix} 9 & 6 \\ 12 & 2 \\ 2 & 15 \end{pmatrix}.$$

Маски ж відповідно:

0	0	0
0	0	1
0	0	0
0	1	1
1	0	0
1	0	

Виконуємо зсув S:

$$S\left(G\big(T(M_2)\big)\right) = \begin{pmatrix} \mathbf{9}| & 6 & 0 & 0 \\ 0| & 12 & 2 & 0 \\ 0| & 0 & 2 & 15 \end{pmatrix}.$$

Тобто кожний вектор даних (рядок) μ_i зсуваємо на *(i-1)* елемент вправо. На місцях невизначених елементів ставимо 0. Хвостовий елемент знаходиться в отриманій матриці з індексами (1,1), тобто верхній лівий елемент (в даному випадку – 9).

Після цього відкидаємо перший стовпець і утворюємо матрицю M_3:

$$M_3 = \begin{pmatrix} 6 & 0 & 0 \\ 12 & 2 & 0 \\ 0 & 2 & 15 \end{pmatrix}.$$

Ітерація 3.

Повторюємо визначені у другій ітерації дії.

$$M_3 = \begin{pmatrix} 6 & 0 & 0 \\ 12 & 2 & 0 \\ 0 & 2 & 15 \end{pmatrix}.$$

Транспонуємо матрицю M_3 :

$$T(M_3) = \begin{pmatrix} 6 & 12 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 15 \end{pmatrix}.$$

G-перетворення:

μ_{1_1}	H_1	μ_{1_1}'	m	μ_{1_2}	m	μ_{1_2}'	m
6	0	6		0	1		
12	0	12	6	6	0	6	6
			↓				↓
			12				***6***

Примітка: слід зауважити, що останні нулі в μ_{1_1} не враховуються.

μ_{2_1}	H_1	μ_{2_1}'	m
0	1		
2	0	2	2
2	0	2	↓
			4

μ_{3_1}	H_1	μ_{3_1}'	m
0	1		
0	1	15	15
15	0		↓
			15

$$G(M_3) = \begin{pmatrix} 12 & 6 \\ 4 & 0 \\ 15 & 0 \end{pmatrix}.$$

Маски ж відповідно:

0 0
1 0
1 0 0
1 1 0

Виконуємо зсув S:

$$S\left(G\left(T(M_3)\right)\right) = \begin{pmatrix} 12| & 6 & 0 \\ 0| & 4 & 0 \\ 0| & 0 & 15 \end{pmatrix} = \mathrm{M}_4.$$

Хвостовий елемент в даному випадку дорівнює 12.

$$\mathrm{M}_4 = \begin{pmatrix} 6 & 0 \\ 4 & 0 \\ 0 & 15 \end{pmatrix}.$$

Ітерація 4.

Транспонування:

$$T\ (M_4) = \begin{pmatrix} 6 & 4 & 0 \\ 0 & 0 & 15 \end{pmatrix}.$$

G-перетворення:

μ_{1_1}	H_1	μ_{1_1}'	m	μ_{1_2}	m	μ_{1_2}'	m
6	0	6		2	0	2	
4	0	4	4	0	1		2
			↓				↓
			8				***2***

μ_{2_1}	H_1	μ_{2_1}'	m
0	1		
0	1		15
15	0	15	↓
			15

Отже

$$G\left(T(M_4)\right) = \begin{pmatrix} 8 & 2 \\ 15 & 0 \end{pmatrix}.$$

$$S\left(G\left(T(M_4)\right)\right) = \begin{pmatrix} 8| & 2 \\ 0| & 15 \end{pmatrix} = \mathrm{M}_5.$$

Примітка: кінцеві нулі як і при G-перетворенні ігноруються.

Маски:

0 0
0 1
1 1 0

Хвостовий елемент в даному випадку дорівнює 8.

Ітерація 5.

$$M_5 = \begin{pmatrix} 2 \\ 15 \end{pmatrix}.$$

Транспонуємо M_5:

$$T\ (M_5) = (2 \quad 15).$$

G-перетворення:

μ_{1_1}	H_1	μ_{1_1}'	m	μ_{1_2}	m	μ_{1_2}'	m
2	0	2		0	1		
15	0	15	2	13	0	13	13
			↓				↓
			4				***13***

$$G\big(T(M_5)\big) = (4| \quad 13).$$

Маски: 0 0
1 0

Хвостовий елемент в даному випадку дорівнює 4.

Ітерація 6.

$(M_6) = (13).$

Алгоритм паралельно-ієрархічного перетворення завершується коли матриця складається з одного елемента. Останній хвостовий елемент в даному випадку дорівнює 13.

АНАЛІЗ ОСОБЛИВОСТЕЙ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

У попередніх роботах вже було описано результати тестування та структурно-функціональну організацію розробленого програмного комплексу паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу на базі апаратної платформи центрального процесора (CPU). Вказаний програмний комплекс призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами (у вигляді двовимірної матриці даних довільної розмірності або зображення), а також зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для їх відновлення [18-20].

Тому зупинимося на аналізі результатів тестування та структурно-функціональній організації розробленого нового програмного комплексу паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу на базі апаратної платформи GPU, тобто графічних спецпроцесорів відеоадаптера (з використанням технологій GPGPU).

Для написання основного програмного модуля обрано технологію CUDA. Технологія CUDA являє собою середовище розробки мовою програмування C, яка дозволяє розробляти програмне забезпечення для ефективного вирішення надскладних обчислювальних задач з підвищеною швидкодією завдяки багатоядерній обчислювальній потужності графічних процесорів (GPU). Технологія CUDA є достатньо гнучкою, адже надає розробнику можливість на свій розсуд організовувати доступ до набору інструкцій графічного прискорювача і керувати його пам'яттю, організовувати на GPU складні паралельні обчислення. Натепер графічний процесор з підтримкою CUDA є потужною програмованою відкритою архітектурою.

Для розробки графічного інтерфейсу було обрано декларативну мову програмування Qml, для реалізації внутрішньої логіки програми, та зв’язку з інтелектуальним модулем використано мову програмування високого рівня C++. Qml (Qt Meta-Object Language) – це декларативна мова для розробки та дизайну інтерфейсу програм, яка є частиною фреймворка Qt. Перевагою є те, що програми написані з використанням Qml можуть працювати на мобільних платформах таких як Android, Maemo, Symbian. Синтаксис мови складається з операторів мови програмування C++, JavaScript та CSS.

Для роботи із основним програмним модулем, який власне реалізує паралельно-ієрархічного (ПІ) перетворення на GPU потрібно:

1) додати вихідні коди до проекту;

2) створити один екземпляр класу vector<vector<int> >: заповнити його значеннями пікселів зображення (здійснюється автоматично після процедури завантаження зображення);

3) викликати метод PI, параметрами якого потрібно вказати двовимірний масив зображення, змінну типу int, в яку буде записано тривалість роботи програми в мілісекундах, та одновимірний масив типу vector<int>, в який буде записано результат роботи програми.

В основному програмному модулі визначено тип структури Array, який використовується для представлення масиву чисел, що знаходиться в пам'яті відеоадаптера.

Функція PIsort – призначена для виконання першої частини операції G-перетворення (сортування).

Функція PIkernel – виконує другу частину операції G-перетворення (нормування).

Функція PIpreCalculate – використовується для виконання першої ітерації ПІ-перетворення.

Функція aPI – виділяє пам'ять яка необхідна для роботи алгоритму, контролює виконання перетворення.

Функція PI – інтерфейс модуля, саме через цю функцію зовнішні програмні пакети можуть виконувати ПІ-перетворення.

Програмний модуль функціонує, виконуючи такі етапи:

1) завантаження зображення – у вигляді двомірного набору чисел;

2) виконання прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами.

Результатом роботи програмного модуля є набір хвостових елементів, який відображається у нижній частині вікна програми.

За допомогою спеціалізованого програмного продукту NVidia Parallel Nsight (компанії NVidia) було здійснено аналіз продуктивності роботи реалізованої програми на графічних процесорах відео адаптера (GPU). Для прикладу, в результаті перетворення кольорового зображення розмірністю 878×878 пікселів (2 312 652 елемента вхідної матриці) отримана часова діаграма роботи програми (рис. 1), яка переконливо свідчить про підвищення продуктивності оброблення зображення (верхня шкала – "Time") за рахунок природної розпаралелізації обчислювального процесу паралельно-ієрархічного перетворення в межах окремо призначених потоків на 96 різних графічних спецпроцесорах (адже в даному експериментальному дослідженні використовувався 96-ядерний відеоадаптер NVidia GeForce GT 435M).

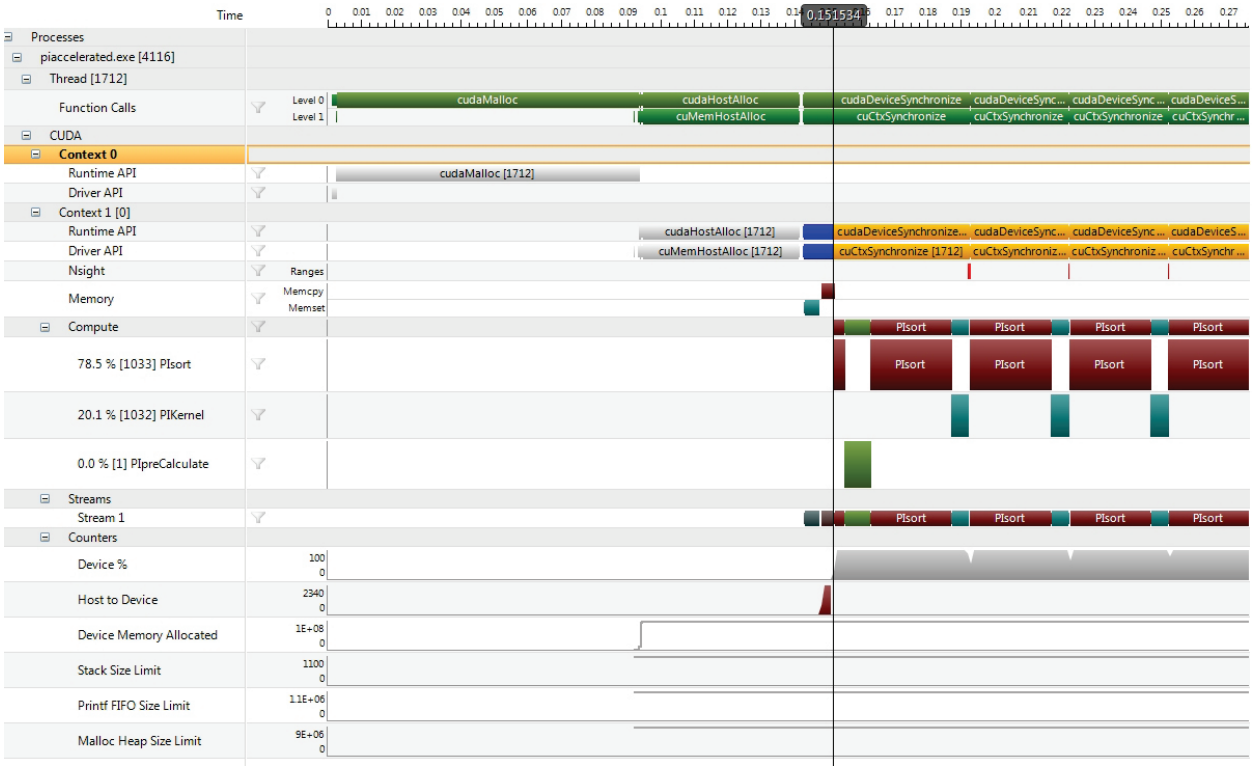

Рис. 1. Часова діаграма роботи програмного модуля реалізації ПІ перетворення на GPU

Результати роботи програмного комплексу на базі апаратної платформи GPU співпали з результатами математичного та комп'ютерного моделювання, що свідчить про коректність та достовірність роботи програмного продукту [18-20]. Також результати отримані вказаним програмним комплексом повністю збігаються з результатами роботи розробленого раніше програмного комплексу на базі апаратної платформи центрального процесора (CPU), проте час виконання ПІ перетворення менший ніж у CPU-версії. На рис. 2 наведено графік для порівняння показників швидкодії реалізованих програмних комплексів. На графіку зображено середня тривалість роботи програми в залежності від

кількості елементів вхідної матриці розмірністю n×m. Необхідно відзначити, що оскільки в дослідженнях була орієнтація на обробку кольорових зображень, то підрахунок кількості елементів вхідної матриці містив n×m×3, оскільки кольорове зображення містить в свою чергу ще 3 складових кольору (RGB) із відповідними значеннями.

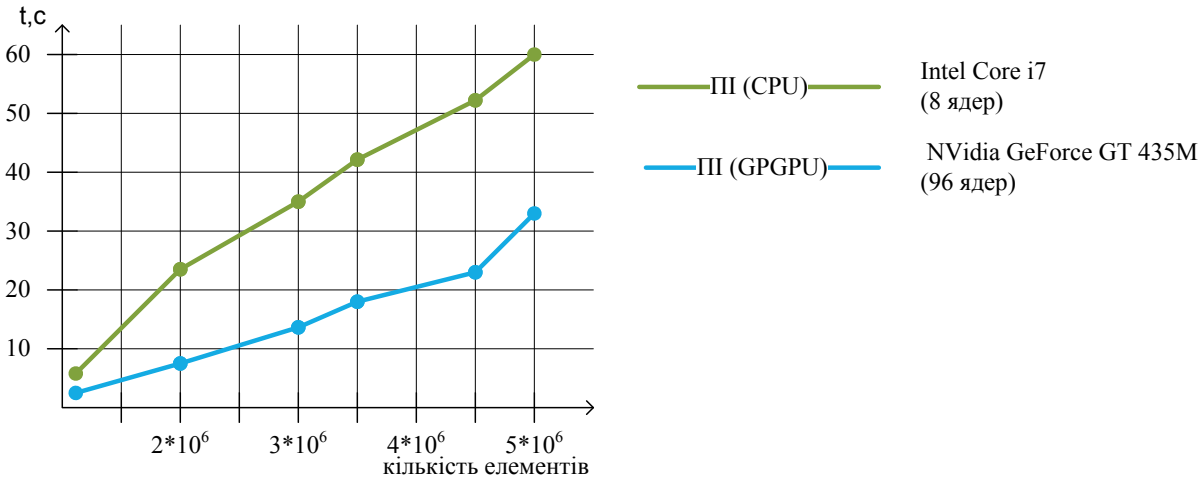

Рис. 2. Графічне відображення залежності тривалості роботи розроблених програмних комплексів від кількості вхідних елементів

ВИСНОВКИ

В ході проведених наукових досліджень було проаналізовано особливості організації високопродуктивних паралельно-ієрархічних обчислювальних процесів, а саме, на рівні детального аналізу математичного опису, процесу організації обчислень, алгоритмічної та програмної реалізації паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу на базі апаратної платформи як центрального процесора (CPU), так і на базі апаратної платформи GPU, тобто графічних спецпроцесорів відеоадаптера (з використанням технологій GPGPU). Наведено порівняльну характеристику показників продуктивності обробки зображень на основі використаних технологій, де відзначено перспективність застосування апаратної платформи GPU. Відзначено, що межах запропонованої системи є можливість здійснювати моделювання часового зсуву між гілками мережі, наявність якого приводить до можливості проектування високопродуктивних паралельно-ієрархічних обчислювальних пристроїв.

На основі проведених досліджень, математичного та комп'ютерного моделювання розроблено програмно-апаратні комплекси, призначені для реалізації паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами. Набув подальшого розвитку опис паралельно-ієрархічного обчислювального процесу обробки інформації у вигляді багатоетапних процедур кореляційних взаємодій, що дозволяє синтезувати паралельно-ієрархічну мережеву обчислювальну систему для складної функціональної обробки інформаційних полів в динаміці, зокрема для задач ефективного багатоетапного сприйняття, збереження, ущільнення, обробки зображень та розпізнавання образів.

СПИСОК ЛІТЕРАТУРИ

1. Капитонова Ю.В. О некоторых тенденциях развития и проблемах искусственного интеллекта / Ю.В. Капитонова, В.И. Скурихин // Кибернетика и системный анализ. – 1999. – №1. – С.43-50.
2. Рабинович З.Л. Представление и обработка знаний во взаимодействии сенсорной и языковой нейросистем человека / З.Л. Рабинович, Г.С. Воронков // Кибернетика и системный анализ. – 1998. – №2. – С.3-11.
3. Волкова В.Н. Теория систем и методы системного анализа в управлении и связи / В.Н. Волкова, В.А. Воронков, А.А. Денисов и др. – М.: Радио и связь, 1983. – 248 с.
4. Волкова В.Н. Информационный анализ иерархических структур / В.Н. Волкова, А.А. Денисов // Применение методов и моделей системного анализа при разработке автоматизированных информационно-поисковых систем: Обзорная информация. – М.: НИИВШ, 1980. – С. 27-39.

5. Кнут Д. Искусство программирования для ЭВМ. / Д. Кнут [в 3 т.] – М.: Мир, 1978.
6. Александров В.В. Представление и обработка изображений: Рекурсивный подход / В.В. Александров, Н.Д. Горский – Л.: Наука, 1985. – 192 с.
7. Метлицкий Е.А. Системы параллельной памяти: Теория, проектирование, применение. / Е.А. Метлицкий, В.В. Каверзнев [Под ред. В.И. Тимохина]. – Л.: Ленинградский университет, 1989. – 240с.
8. Компьютер обретает разум / Под ред. В.Л. Стефанюка. – М.: Мир, 1990. – 240с.
9. Minsky M. Framework for Representing Knowledge / M. Minsky // Artificial Intelligence. – №36. – 1974.
10. Шенк Р. Познать механизмы мышления / Р. Шенк, Л. Хантер // Реальность и прогнозы искусственного интеллекта. – М.: Мир, 1987. – С.123-134.
11. Глушков В.М. Теория структур данных и синхронные параллельные вычисления / В.М. Глушков, Ю.В. Капитонова, А.А. Летичевский // Кибернетика. – 1976. – №6. – С. 2-15.
12. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : [Монографія.] / В.П. Кожем'яко, Ю.Ф. Кутаєв, С.В. Свєчніков, Л.І. Тимченко, А.А. Яровий – Вінниця: УНІВЕРСУМ-Вінниця, 2003. – 324 с.
13. В.П. Кожем'яко Паралельно-ієрархічні мережі як структурно-функціональний базис для побудови спеціалізованих моделей образного комп'ютера : [Монографія] / В.П. Кожем'яко, Л.І. Тимченко, А.А. Яровий. – Вінниця: Універсум-Вінниця, 2005. – 161 с.
14. Кожем'яко В.П. Образний відео-комп'ютер око-процесорного типу : [Монографія] / Кожем'яко В.П., Лисенко Г.Л., Яровий А.А., Кожем'яко А.В. – Вінниця: Універсум-Вінниця, 2008. – 215 с.
15. Kozhemyako V.P. Approach for real – time image recognition. / V.P. Kozhemyako, L.I. Tymchenko, Yu.F. Kutaev, A.A. Yaroviy // Оптико-електронні інформаційно-енергетичні технології. – 2001 р. – №1. –С. 110-124.
16. Паралельно – ієрархічне перетворення і Q-обробка інформації для систем реального часу : [Монографія] / Тимченко Л.І., В.П. Кожем'яко, Ю.Ф. Кутаєв, С.В. Свєчніков, Білан С.М., О.І. Стасюк, М.О. Ковзель, Л.В. Загоруйко – Київ: КУЕТТ, 2006 – 492 с.
17. Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідально сітьового перетворення зображень. / Кожем'яко В.П., Тимченко Л.І., Кутаев Ю.Ф., Івасюк І.Д. – К: УМК ВО, 1994. – 272 с.
18. Яровий А.А. Паралельно-ієрархічне перетворення інформаційних середовищ на основі маскового методу / А.А. Яровий // Оптико-електронні інформаційно-енергетичні технології. – 2011. – №1 (21). –С. 15-23.
19. Свідоцтво про реєстрацію авторського права на твір № 39490. Комп'ютерна програма "Комп'ютерна програма прямого паралельно-ієрархічного перетворення з оптимізацією формування масок (із множенням мінімального елемента на потужність в операторі перетворення G)" / Яровий А.А., Сугак І.М. Дата реєстрації Державним Департаментом інтелектуальної власності України 04.08.2011.
20. Свідоцтво про реєстрацію авторського права на твір № 39489. Комп'ютерна програма "Комп'ютерна програма зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу (із множенням мінімального елемента на потужність в операторі перетворення G)" / Яровий А.А., Сугак І.М. Дата реєстрації Державним Департаментом інтелектуальної власності України 04.08.2011.

Надійшла до редакції 21.10.2011 р.

ЯРОВИЙ А.А. – к.т.н., доцент, докторант кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, Україна.