
СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

УДК 621.398:007

А.А. ЯРОВИЙ

МЕТОДОЛОГІЧНІ ОСОБЛИВОСТІ ПОБУДОВИ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ ТА ІЄРАРХ-ІЄРАРХІЧНИХ МЕРЕЖ НА ОСНОВІ КЛАСТЕРНИХ СИСТЕМ З РОЗПОДІЛЕНОЮ ОБРОБКОЮ ІНФОРМАЦІЇ

*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 методологічних і прикладних аспектів побудови паралельно-ієрархічних та ієрарх-ієрархічних мереж на основі кластерних систем з розподіленою обробкою інформації. Наведено результати розробки кластерного обчислювального комплексу для виконання паралельно-ієрархічного перетворення інформаційних середовищ із використаннями на кожному host-комп'ютері, за вибором користувача, обчислювальних потужностей як його центрального процесора (CPU), так і відеоадаптера (на основі технологій GPGPU), а також подальшої розробки моделей інтелектуальних паралельно-ієрархічних та ієрарх-ієрархічних нейроподібних систем.

Аннотация. В проведенных исследованиях совершен анализ методологических и прикладных аспектов построения параллельно-иерархических и иерарх-иерархических сетей на базе кластерных систем с распределенной обработкой информации. Приведено результаты разработки кластерного вычислительного комплекса для выполнения параллельно-иерархического преобразования информационных сред с использованием на каждом host-компьютере, по выбору пользователя, вычислительных мощностей как его центрального процессора (CPU), так и видеоадаптера (на базе технологий GPGPU), а также дальнейшей разработки моделей интеллектуальных параллельно-иерархических и иерарх-иерархических нейроподобных систем.

Abstract. The analysis of methodological and applied aspects of construction of parallel-hierarchical and hierarch-hierarchical networks on the base of cluster systems with distributed information processing is carried out in the given researches. The results of development of the cluster computing system for parallel-hierarchical processing of information environments with exploiting by user selection on the each host-computer processing capacities as well CPU, as a video-card (on the base of GPGPU technologies) and future development the models of intelligence parallel-hierarchical and hierarch-hierarchical neural-like systems are given.

Ключові слова: паралельні обчислення, кластери, GPGPU, програмування відеоадаптерів, паралельно-ієрархічне перетворення, обробка зображень.

ВСТУП

Аналізуючи та враховуючи останні тенденції розвитку прикладних задач можна стверджувати, що для того, щоб розробляти сучасні високопродуктивні обчислювальні комплекси, розробникам необхідно мати практичні знання не лише сучасних тенденцій розвитку обчислювальних технологій, але й парадигм паралельного і розподіленого програмування [1-4].

Розподілені мережеві комп'ютерні системи є системоутворюючою складовою сучасних інформаційних інфраструктур. Комп'ютерні технології забезпечують інформаційну інтегрованість при наявній географічній розподіленості. Характеризуючи такого роду обчислювальні комплекси, як правило, визначають такі особливості як відкритість, паралельність, масштабованість, відмовостійкість, прозорість,

спільність використання ресурсів, що надають певні переваги при розв'язанні великої кількості прикладних задач. Завдяки відкритості таких комплексів легко виконувати розширення системи, додаючи нові ресурси. Паралельність передбачає можливість одночасного виконання декількох процесів на різних комп'ютерах в межах комплексу, які можуть взаємодіяти під час їхнього виконання; масштабованість – можливість додавання нових властивостей і методів [1-3].

Останнім часом перед розробниками високопродуктивних систем обробки інформації все частіше ставиться завдання, щоб окремі програмні складові належним чином виконувалися в Internet або Intranet. Якщо програмний комплекс (або його частина, програма) розгорнута в одному або декількох таких середовищах, то до нього пред'являються найжорсткіші вимоги з продуктивності. Користувач завжди сподівається, що результати роботи обчислювального комплексу будуть миттєвими і надійними. Наприклад, програмне забезпечення, призначене для прийому телетексту, також повинно бути здатне на "гладке" відтворення графічних зображень і звуку після цифрової обробки (причому без переривання). Програмне забезпечення Web-сервера нерідко витримує сотні тисяч відвідувань в день, а часто відвідувані поштові сервери – витримують порядку мільйона відправлених та одержаних повідомлень. При цьому важливою є не лише кількість оброблюваних повідомлень, але і їх вміст. Наприклад, передача даних, що містять оцифровані музику, відео або графічні зображення, може "поглинути" всю пропускну здатність мережі і заподіяти серйозні неприємності програмному забезпеченню сервера, яке не була спроектоване належним чином. Зазвичай ми маємо справу з мережевим обчислювальним середовищем, що складається з комп'ютерів із декількома процесорами. Чим більше функцій покладається на програмне забезпечення обчислювального середовища, тим більше до нього пред'являється вимог. Щоб задовольнити мінімальні вимоги користувача, сучасні програмні комплекси повинні бути ще більш продуктивними і інтелектуальними [2-4].

Отже, програмний комплекс слід проектувати так, щоб можна було скористатися перевагами комп'ютерів, обладнаних не лише декількома процесорами, а й враховуючи останні досягнення технології GPGPU, декількома відеоадаптерами, що дозволяє досягати окремих показників пікової продуктивності супер-комп'ютерів. А оскільки мережеві обчислювальні комплекси, кластерні системи – це швидше правило, ніж виняток, то метою проектування програмного забезпечення для паралельної обробки інформації повинно також бути його коректне і ефективне виконання за умови, що деякі його складові будуть одночасно виконуватися на різних комп'ютерах. Таким чином, щоб впоратися з описаними реаліями, асортимент розробок високопродуктивних обчислювальних комплексів повинен містити методи реалізації паралелелізму засобами як паралельного, так і розподіленого програмування [5].

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є дослідження методологічних особливостей побудови паралельно-ієрархічних та ієрарх-ієрархічних мереж на основі кластерного обчислювального комплексу, розробленого в результаті інтеграції парадигм паралельної та розподіленої обробки інформації, а також технологій GPGPU для подальшої реалізації швидкої та ефективної паралельної обробки надвеликих масивів інформації.

ПОСТАНОВКА ПРОБЛЕМИ. АНАЛІЗ ПІДХОДІВ ДО РЕАЛІЗАЦІЇ ПАРАЛЕЛЬНОСТІ В ПРОГРАМНИХ ДОДАТКАХ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ ІНФОРМАЦІЇ

Дві події називають одночасними, якщо вони відбуваються протягом одного й того ж часового інтервалу. Якщо кілька завдань виконуються протягом одного й того ж терміну, то говорять, що вони виконуються паралельно [2,6]. Тобто, в даному контексті, термін «паралельно» необов'язково означає «точно в один момент». Наприклад, два завдання можуть виконуватися паралельно протягом однієї і тієї ж секунди, але при цьому кожна з них виконується в різні частини цієї секунди. Так, перше завдання може відпрацювати в першу десятю частину секунди і призупинитися, потім друге може відпрацювати в наступну десятю частину секунди і призупинитися, після чого перше завдання може відновити виконання протягом третьої частини секунди, і т.д. Таким чином, ці завдання можуть виконуватися по черзі, але оскільки тривалість секунди, з точки зору людини, дуже коротка, то здається, що вони виконуються одночасно. Поняття одночасності (паралельності) можна поширити і на довші інтервали часу. Так, дві програми, що виконують деяку задачу протягом однієї і тієї ж години, поступово наближаючись до своєї кінцевої мети протягом цієї години, можуть працювати (або можуть не працювати) точно в одні і ті ж моменти часу. Ми ж говоримо, що ці дві програми для цієї години виконуються паралельно, або одночасно. Іншими словами, завдання, які існують в один і той же час і виконуються протягом одного й того самого інтервалу часу, є паралельними. Паралельні завдання можуть виконуватися в одно- або багатопроцесорному середовищі. У однопроцесорному середовищі паралельні завдання існують в один і той же час та виконуються протягом одного й того самого інтервалу часу за рахунок контекстного

перемикання. У багатопроцесорному середовищі, якщо достатня кількість процесорів, паралельні завдання можуть виконуватися в одні і ті ж моменти часу протягом одного й того ж періоду. Основний фактор, що впливає на ступінь прийнятності для паралелізму того чи іншого інтервалу часу, визначається конкретним програмним додатком [2,3,6].

Мета технологій паралелізму – забезпечити умови, що дозволяють комп'ютерними програмами здійснити більший обсяг роботи за той же період часу. Тому проектування програм має орієнтуватися не на виконання однієї задачі в деякий проміжок часу, а на одночасне виконання декількох задач, на які попередньо повинна бути розбита програма. Можливі ситуації, коли метою є не виконання більшого обсягу роботи протягом того ж інтервалу часу, а спрощення вирішення задачі з точки зору програмування. Іноді має сенс думати про вирішення проблеми як про множину паралельно виконуваних завдань [2,3,6,7].

Зазвичай не надто корисно (або ефективно) виконувати одну підзадачу в один період часу, а іншу – зовсім в інший. Саме паралельність обох процесів дає природну форму повного вирішення проблеми. Іноді до паралельності вдаються, щоб збільшити швидкість роботи програми або наблизити момент її завершення. В інших випадках паралельність використовується для збільшення продуктивності програми (обсягу виконуваної нею роботи) за той же період часу при вторинності швидкості її роботи. Наприклад, для деяких Web-сайтів важливо якомога довше утримувати користувача. Тому тут має значення не те, наскільки швидко буде відбуватися підключення (реєстрація) і відключення користувачів, а скільки користувачів зможе цей сайт обслуговувати одночасно. Отже, мета проектування програмного забезпечення такого сайту – обробляти максимальну кількість підключень за якомога більший проміжок часу. Нарешті, паралельність спрощує саме програмне забезпечення. Найчастіше складну послідовність операцій можна спростити, організувавши її у вигляді ряду невеликих паралельно виконуваних операцій. Незалежно від специфічної вузькоспрямованої мети (прискорення роботи програм, обробка збільшеного навантаження або спрощення реалізації програми), головною метою є вдосконалити програмне забезпечення, скориставшись принципом паралельності [2,6].

Паралельне і розподілене програмування – це два базових підходи до досягнення паралельного виконання складових програмного забезпечення. Вони являють собою дві різні парадигми програмування, які іноді перетинаються. Методи паралельного програмування дозволяють розподілити роботу програми між двома (або більше) процесорами в рамках одного фізичного або одного віртуального комп'ютера. Методи розподіленого програмування дозволяють розподілити роботу програми між двома (або більше) процесорами, причому процеси можуть існувати на одному і тому ж комп'ютері або на різних [2,3,6].

Програма, яка містить паралелізм, виконується на одному і тому ж фізичному або віртуальному комп'ютері. Таку програму можна розбити на процеси (process) або потоки (thread).

Багатопоточність обмежується паралелізмом. Формально паралельні програми іноді бувають розподіленими, наприклад, при PVM-програмуванні (Parallel Virtual Machine – паралельна віртуальна машина). Розподілене програмування іноді використовується для реалізації паралелізму, як у випадку з MPI-програмуванням (Message Passing Interface – інтерфейс для передачі повідомлень) [2,6].

Однак не всі розподілені програми включають паралелізм. Частини розподіленої програми можуть виконуватися за різноманітними запитами й у різні періоди часу.

При чистому паралелізмі одночасно виконуючі частини є компонентами однієї і тієї ж програми. Частини розподілених програм зазвичай реалізуються як окремі програми. Типова архітектура побудови паралельної та розподіленої програми показана на рис. 1.

Паралельний додаток, показаний на рис. 1(а), складається з однієї програми, поділеної на чотири завдання. Кожне завдання виконується на окремому процесорі, отже, всі вони можуть виконуватися одночасно. Ці завдання можна також реалізувати розподілено (рис. 1(б)), коли додаток складається з трьох окремих програм, кожна з яких виконується на окремому комп'ютері. При цьому програма 3 складається з двох окремих частин (задачі А і задачі D), що виконуються на одному комп'ютері. Незважаючи на це, задачі А і D є розподіленими, оскільки вони реалізовані як два окремих процеси. Завдання паралельної програми більш тісно пов'язані, ніж задачі розподіленого додатку. У загальному випадку процесори, пов'язані з розподіленими програмами, знаходяться на різних комп'ютерах, у той час як процесори, пов'язані з програмами, що реалізують паралелізм, перебувають на одному і тому ж комп'ютері. Звичайно ж, існують гібридні додатки, які є і паралельними, і розподіленими одночасно [2,3,6]. Саме такі гібридні об'єднання останнім часом стають нормою і є об'єктом проведених досліджень.

Зважаючи на наведені методологічні особливості реалізації паралельності в програмних додатках можна стверджувати, що організація паралельно-ієрархічних мереж входить до класу задач, які ефективно реалізуються на основі паралельної обробки інформації (зокрема на GPU-платформах), а організація ієрарх-ієрархічних мереж – відповідно на основі розподіленої обробки інформації із

застосуванням кластерних систем (включаючи як CPU-орієнтовані кластерні платформи, так і GPU – орієнтовані кластерні платформи) [5,8-10].

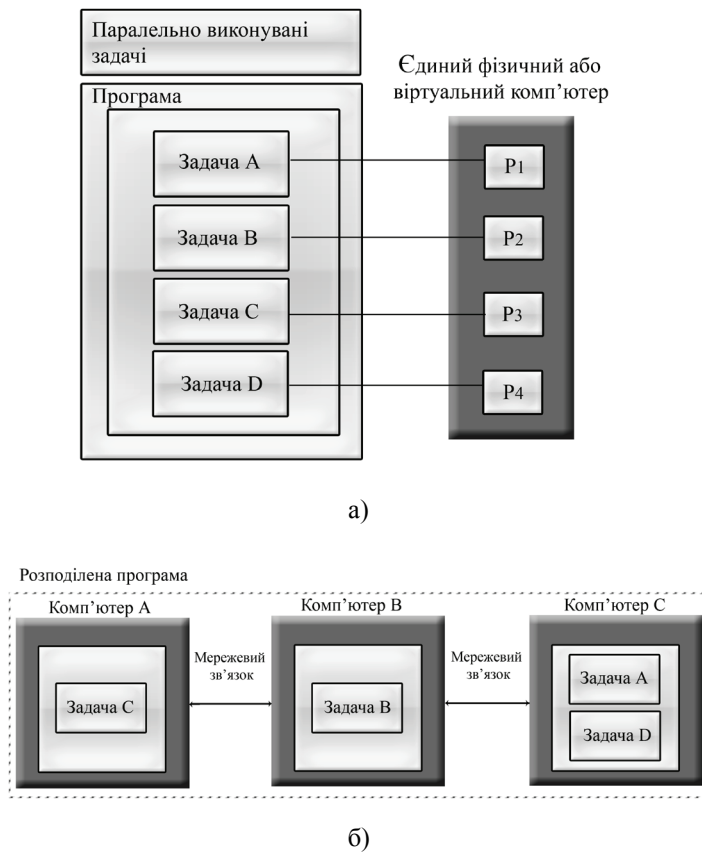


Рис. 1. Типова архітектура побудови паралельної (а) і розподіленої (б) програми [2]

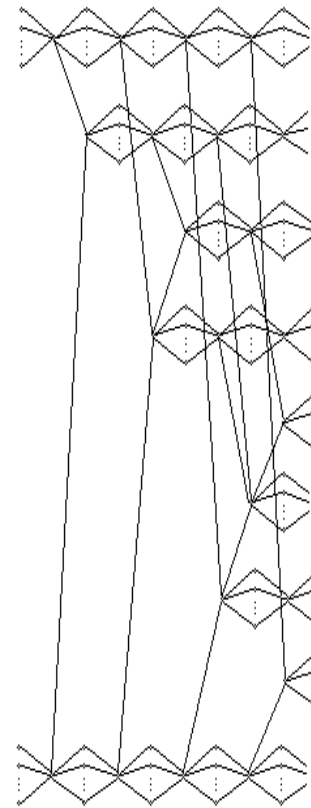


Рис. 2. Деревоподібна модель мережної структури паралельно-ієрархічного перетворення [7]

ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ МЕРЕЖНОЇ МОДЕЛІ ПАРАЛЕЛЬНО-ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

Одним із шляхів реалізації паралелізму при роботі з багатозв'язними структурами є "регуляризація", що забезпечує опис їх за допомогою регулярної мережної структури перетворення. При цьому інформація про зв'язки включається в мережну структуру перетворення в явному вигляді, тобто подається елементами даних. Зупинимося на визначенні абстрактної моделі мережної структури перетворення [7,11].

Введемо деякі поняття, що відносяться до деревоподібної моделі мережної структури перетворення з регулярними зв'язками (рис. 2). Нехай граф $G = (V, E)$ є структура обробки даних, що складається з множини вузлів V і множини ребер E . Граф є направленим, якщо ребра представлені у виді упорядкованих пар вузлів. Дерево перетворення визначається направленим графом, що володіє такими властивостями: тільки одні кореневі вузли не мають дуг, що входять у них; у кожний вузол входить множина дуг, число яких визначається нелінійною структурою оброблюваних даних; із кожного кореня до вузла йде єдиний шлях, тобто єдиний кінцевий набір ребер [7,11].

Структура обробки даних ототожнюється із направленим графом, причому вузлам відповідають елементи даних, а направлені дуги, що пов'язують вузли описують різноманітні залежності між елементами і маркуються відповідним чином.

Структура обробки даних моделі структури перетворення $D = \{K, \Psi\}$ визначається множиною K вузлів і множиною $\Psi = \{f_1, f_2, \dots\}$ функцій $f_i: K \rightarrow 1$ і $1 \rightarrow K$, що відображають множину вузлів в один вузол і навпаки. Два вузли K і $K'(K, K' \in K)$ пов'язані дугою f_i , якщо $K' = f_i(K)$, дуга при

цьому направлена від K і K' . Отже, дерево мережного паралельно-ієрархічного перетворення містить два види піддерев: конвергентні і дивергентні [7,11].

Структура обробки даних $D = \{K, \Psi\}$, при якій множина вузлів відображається в один вузол, тобто $K \rightarrow 1$; а функція Ψ визначається F^* - критерієм, утворить конвергентну структуру піддерева [11].

Структура обробки даних $D^* = \{K, \Psi^*\}$, при якій один вузол відображається в множину вузлів K , тобто $1 \rightarrow K$, а функція Ψ^* визначається Q^* - функцією перетворення, утворить дивергентну структуру піддерева [11].

Послідовне в часі формування конвергентних (K) і дивергентних (D) піддерев утворить K - D дерево. Сусідні K - D дерева одного рівня в часі зміщені один відносно одного на одне K і D піддерева. Нелінійна структура K - D дерев (рис. 2) утворить узагальнене дерево мережі. Перетинами узагальненого дерева є однойменні вузли K і D піддерев, що мають однакові шляхи до кореневих вузлів [11].

Хвостовими вузлами є одиночні вузли K піддерев узагальненого дерева мережного перетворення, у перетинах якого знаходиться один вузол K піддерева.

Мережним деревом є кінцева множина K - D дерев, із яких сусідні одного рівня зрушені один щодо одного на K - D дерево, а число перетинів визначається кількістю хвостових вузлів, положення яких у послідовності перетинів описується – $(2c + 3)$, де c - номер перетину, $c = 0, 1, \dots$. Число рівнів мережного дерева визначається кількістю його хвостових вузлів, збільшеним на одиницю. Сусідніми K - D деревами одного рівня є ті з них, що зрушені на K - D дерево. Гілкою мережного дерева є сформоване K - D дерево [7,11].

Основи методики паралельного перетворення великих масивів інформації розглянемо за допомогою мережного алгоритму, основні властивості якого – це паралелізм і ієрархія, синхронність і детермінованість [7,11,12]. Мережа складається з множини кінцевих множин Ω , множини елементів A і умовно розбита на ряд рівнів.

Мережею паралельно-ієрархічного перетворення є сукупність таких характеристик: $C \in (\Omega, A, Q^*, F^*) \in \{M_1^1(t_0); M_2^1(t_0); \dots; M_h^1(t_0); M_1^2(t_1); \dots; M_n^u(t_s)\}$ – кінцева множина множин, $h \geq 2$ – кількість вихідних множин, $u \geq 2$ - порядковий номер рівня, $n \geq 2$ – порядковий номер u -го рівня; t_s – такт або крок, на якому сформувався відповідна множина, $S \geq 1$, t_0 – перший або початковий такт, де формуються вихідні множини першого рівня. $A = \{a_1^1(t_1); a_2^1(t_1); \dots; a_h^1(t_1); a_1^1(t_3); \dots; a_n^u(t_s)\}$ – кінцева множина елементів, $u' \geq 2$ – порядковий номер рівня; $n' \geq 1$ – порядковий номер множини, до якої належить елемент; t'_s - такт, на якому формується відповідний елемент [7,11].

Множина множин і множина елементів перетинаються,

$$\Omega \cap A = \emptyset, M_i^j(t_s) = \{a_1^{j-1}(t_s), a_2^{j-1}(t_s) \dots a_k^{j-1}(t_s)\},$$

де $M_i^j(t_s)$ – вихідна множина для j -го рівня.

Таким чином, F^* – критерій вибору елемента з множини $a_i^j(t_s) = F^*[M_i^j(t_{s-1})]$ (перехід від множини $M_i^j(t_{s-1})$ до елемента $a_i^j(t_s)$), Q^* – функція перетворення множини, $Q_{a_i^j(t_s)}^*[M_i^j(t_{s-1})] = M(t_{s+1})$ (перехід від елемента $a_i^j(t_s)$ до множини $M(t_{s+1})$). Потужність вихідних множин $M_i^1(t_0)$ є число m , число яких $H: M_1^1(t_0) = \{a_{11}; a_{12}; \dots a_{1m}\}; M_2^1(t_0) = \{a_{21}; a_{22}; \dots a_{2m}\}; \dots; M_h^1(t_0) = \{a_{h1}; a_{h2}; \dots a_{hm}\}$ [11].

Кожна з цих множин перетвориться по єдиному мережному алгоритму, а всі множини опрацьовуються паралельно. З множини M_i вибирається один елемент a_{ij} і елемент $a_i^1(t_1) = a_{ij}$ є елементом мережі C .

Критерій, за яким обирається елемент із множини, означено як F^* -критерій, $a_i = F^*(M), a_i \in M$. З урахуванням цього вибраного елемента здійснюється перетворення даної

множини, у результаті чого формується нова множина тієї ж потужності, у якій всі елементи, рівні обраному (якщо такі є), певним чином відзначені, наприклад, занулені: $a_{i_1}; a_{i_2}; 0; a_{i_4}; 0; 0; a_{i_7}; \dots; 0; a_{i_m}$, у даному прикладі $a_i^1(t_1) = a_{i_3} = a_{i_5} = a_{i_6} = a_j = a_{i_{m-1}}$. Таку операцію означено як Q^* -перетворення. Таким чином, Q^* -перетворення множини $M = \{a_i\}$, з урахуванням обраного елемента $a_i \in M$, визначається як перетворення, у результаті якого формується нова множина тієї ж потужності, всі елементи якої, рівні a_i і сам цей елемент відзначений, $Q_{a_i}^*(M) = M'$ [7,11,12].

Далі, із знову отриманої множини обирається за F^* -критерієм такий елемент a_{j_1} , $a_j^1(t_3) = a_{j_1}$ (причому $a_{j_1} = a_j$) і здійснюється Q^* -перетворення. У результаті формується множина, у якій всі елементи, рівні a_{j_1} , відзначені. Ітераційне перетворення відбувається доти, доки всі елементи вихідної множини не будуть відзначені. Така множина означається як нульова, і подальше перетворення не здійснюється [11].

Процес перетворення в нульову множину визначимо як процес збіжності. Очевидно, чим менше тактів вибірки необхідно до формування нульової множини, тим краща збіжність даного процесу. Такий вид перетворення назвемо горизонтальним або перетворенням гілки. Відповідно, перетворення вихідної множини з урахуванням проміжних результатів до одержання нульової множини називається гілкою [11].

Розглядаючи усі H початкових множин першого рівня, при виборі елемента $a_j^1(t_j)$, $i = \{1, 2, \dots, h\}$; $j = \{1, 3, 5, \dots\}$ з кожної множини, формуються нові множини, де i – порядковий номер початкової множини, t_j – такт, у якому вибраний елемент. При першому виборі елементів із H вхідних множин у такті t_1 формується нова множина $M_1^2(t_1) = \{a_1^1(t_1), a_2^1(t_1), a_3^1(t_1), \dots, a_h^1(t_1)\}$. На другому кроці перетворення t_3 з початкових множин формується ще одна множина з H елементів: $M_2^2(t_3) = \{a_1^1(t_3), a_2^1(t_3), \dots, a_h^1(t_3)\}$. Таке перетворення здійснюється доти, доки усі вихідні множини не стануть нульовими [7,11].

Множини $M_1^2(t_1); M_2^2(t_3); \dots; M_i^2(t_j)$ – є початковими множинами другого рівня. Їх також будемо перетворювати за мережним алгоритмом до повної збіжності. Тоді перетворення H вхідних множин, що здійснюється на першому рівні перетворення, елементами яких є елементи першого рівня, визначимо як другий рівень [7,11].

При перетворенні множин другого рівня формуються елементи, що сформують початкові множини для третього рівня і так далі до k -го рівня, чий елементи вже не створюють нову множину. Всі перетворення відбуваються по тактах t_i , $i = 1, 2, 3, \dots$. У кожному такті для будь-якого рівня відбувається або вибір елементів із множин за F^* критерієм, або Q^* перетворення множин відповідно до раніш обраних елементів, що свідчить про властивість синхронності даної мережі. Для ілюстрації поняття мережного алгоритму більш зручне графічне представлення мережі, структура якої представляє сукупність множин і елементів. Відповідно до цього структура мережі містить три типи вузлів. На рис. 3 позначення прямокутника \square є початковою множиною, квадрата \square – проміжною множиною (результат F^* перетворення), і кола \circ – елементом. Орієнтовані дуги з'єднують множини й елементи, при цьому деякі дуги направлені від множин до елементів, а інші – від елементів до множин. Дуга, направлена від множини $M_j^i(t_k)$ до елемента $a_j^i(t_{k+1})$ визначає F^* критерій вибору елемента, а дуга – від елемента $a_j^i(t_{k+1})$ до множини $M_j^i(t_{k+2})$ вказує на Q^* перетворення множини. Дуги є направленими, отже це орієнтований граф, нульова множина на якому позначається символом \square [7,11].

З рис. 3 випливає, що є елементи, такі як $a_2^1(t_{s-1}); a_1^2(t_2); a_1^3(t_5)$, позначені на рисунку значком \otimes . Кожний із цих елементів не входить у жодну множину, тому що він у даному такті для свого рівня обраний єдиний, не бере участь у подальшій обробці масивів і є її результатом. Такі елементи називаються хвостовими або елементами, що формують результат. При Q^* перетворенні множин, можуть відзначатися наприклад рівні елементи. Виділимо інформацію про всі рівні елементи і їхнє

розташування в множині. З цією метою для кожного Q^* перетворення множини необхідно поставити у відповідність двійковий код, у якому "одиниці" знаходяться в тих розрядах, у яких позиціях множини знаходяться рівні елементи.

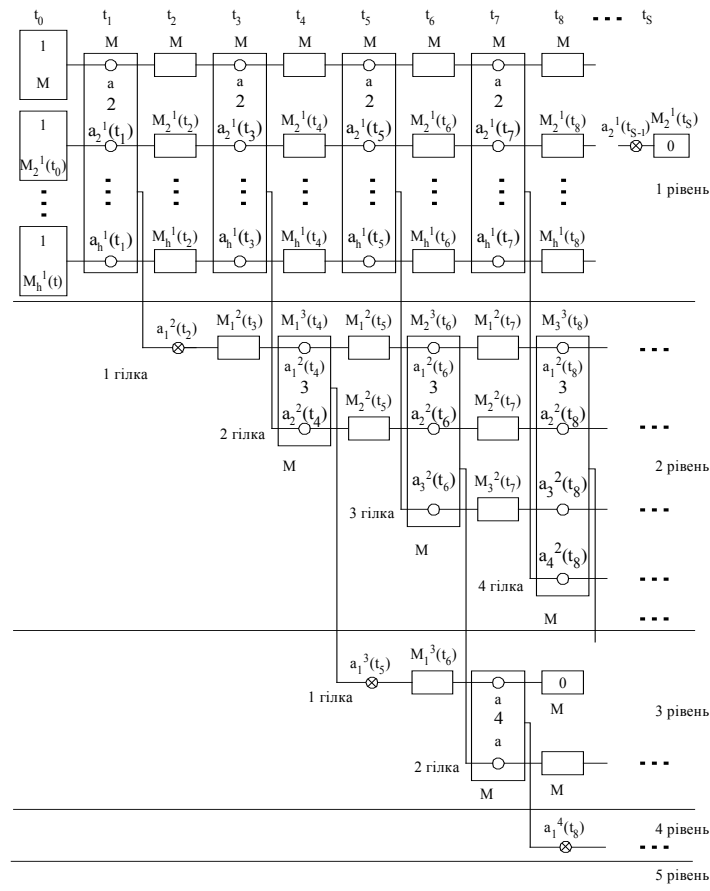


Рис. 3. Структурно-функціональна схема мережі для паралельно-ієрархічного перетворення [11]

Всі інші розряди коду, що відповідають іншим елементам множини, заповнюються "нулями" [7,11].

У минулих роботах було доведено, що досить вдалим розв'язком проблеми швидкої та ефективної паралельної обробки надвеликих масивів даних на основі спеціалізованих системних рішень, зокрема нейроподібних паралельно-ієрархічних систем, є використання технологій GPGPU (зокрема, для реалізації паралельно-ієрархічних мереж) та технологій кластерних систем з розподіленою обробкою інформації (зокрема, для реалізації ієрарх-ієрархічних мереж) [5,8,13]. Особливої ж актуальності та перспективності набуває інтеграція цих двох вказаних технологій в межах кластерного обчислювального комплексу з багатофункціональним набором можливостей: розподіленої обробки інформації в межах обчислювальної мережі та паралельної обробки інформації на host-комп'ютері на основі потужностей відеоадаптера (GPGPU), що надає можливості об'єднання в межах обчислювального кластера host-комп'ютерів, кожен з яких досягає окремих показників пікової продуктивності супер-комп'ютерів (на основі GPGPU) [5].

ПРИКЛАДНІ АСПЕКТИ ПОБУДОВИ ПАРАЛЕЛЬНО-ІЄРАРХІЧНИХ ТА ІЄРАРХ-ІЄРАРХІЧНИХ МЕРЕЖ НА ОСНОВІ КЛАСТЕРНИХ СИСТЕМ З РОЗПОДІЛЕНОЮ ОБРОБКОЮ ІНФОРМАЦІЇ

Розробка кластерного обчислювального комплексу для реалізації паралельно-ієрархічної обробки зображень. Розроблений програмний комплекс DirectPHT::CPU&GPU-Cluster призначений для обробки та ідентифікації зображень за допомогою концепції паралельно-ієрархічного перетворення із застосуванням кластерних обчислень та технології GPGPU. Зокрема кластерний обчислювальний комплекс надає можливості виконання паралельно-ієрархічного перетворення інформаційних середовищ із використаннями на кожному host-комп'ютері обчислювальних потужностей як його центрального

процесора (CPU), так і відеоадаптера (на основі технологій GPGPU), за вибором користувача, а також подальшої розробки моделей інтелектуальних паралельно-ієрархічних систем.

У рамках проекту було реалізовано:

1) сервіс PHNService, оснований на технології Windows Communication Foundation, який виконує пряме паралельно-ієрархічне перетворення над зображеннями розмірністю 128x128 пікселів;

2) серверну програмну бібліотеку PHNServer, яка виконує роль посередника між користувачем та сервісом PHNService, та надає можливість розбивати будь-яке вхідне зображення на частини розмірністю 128x128 пікселів і виконувати над кожною частиною пряме паралельно-ієрархічне перетворення на довільній множині обчислювальних вузлів;

3) користувацький додаток – консоль управління Management Console, яка надає користувацький інтерфейс для використання комплексу.

Усі складові написано на мові C# для платформи Microsoft .NET 3.5.

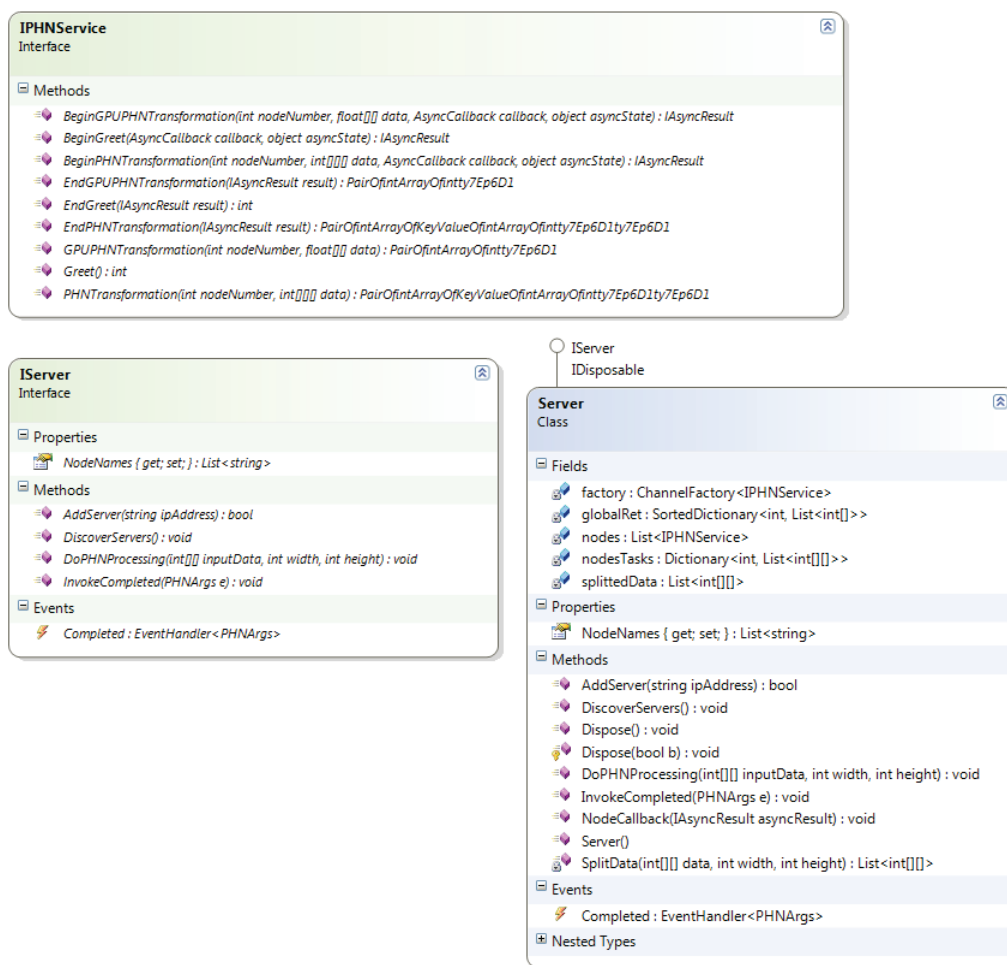


Рис. 4. Класова діаграма серверу, який надає послуги з доступу до сервісу паралельно-ієрархічного перетворення

Для роботи із програмним комплексом необхідно:

1. На кожному обчислювальному вузлі запустити програму PHNHost.exe та дочекатись виведення повідомлення «Hello! Server's up».

2. На контролюючому вузлі запустити програму ControlConsole.exe та додати множину обчислювальних вузлів одним із двох способів:

2.1) вручну вказавши IP-адресу кожного обчислювального вузла за допомогою елемента “Add server”;

2.2) натиснути елемент “Rescan local network” для автоматичного визначення доступних обчислювальних вузлів у локальній мережі класу C.

3. Завантажити зображення, яке необхідно обробити або за допомогою гіперпосилання “Load image”, або за допомогою технології Drag and drop.

4. Натиснути елемент “Execute transformation” для видачі обчислювальним вузлам завдань та ініціалізації процесу обробки зображення, та дочекатись результатів.

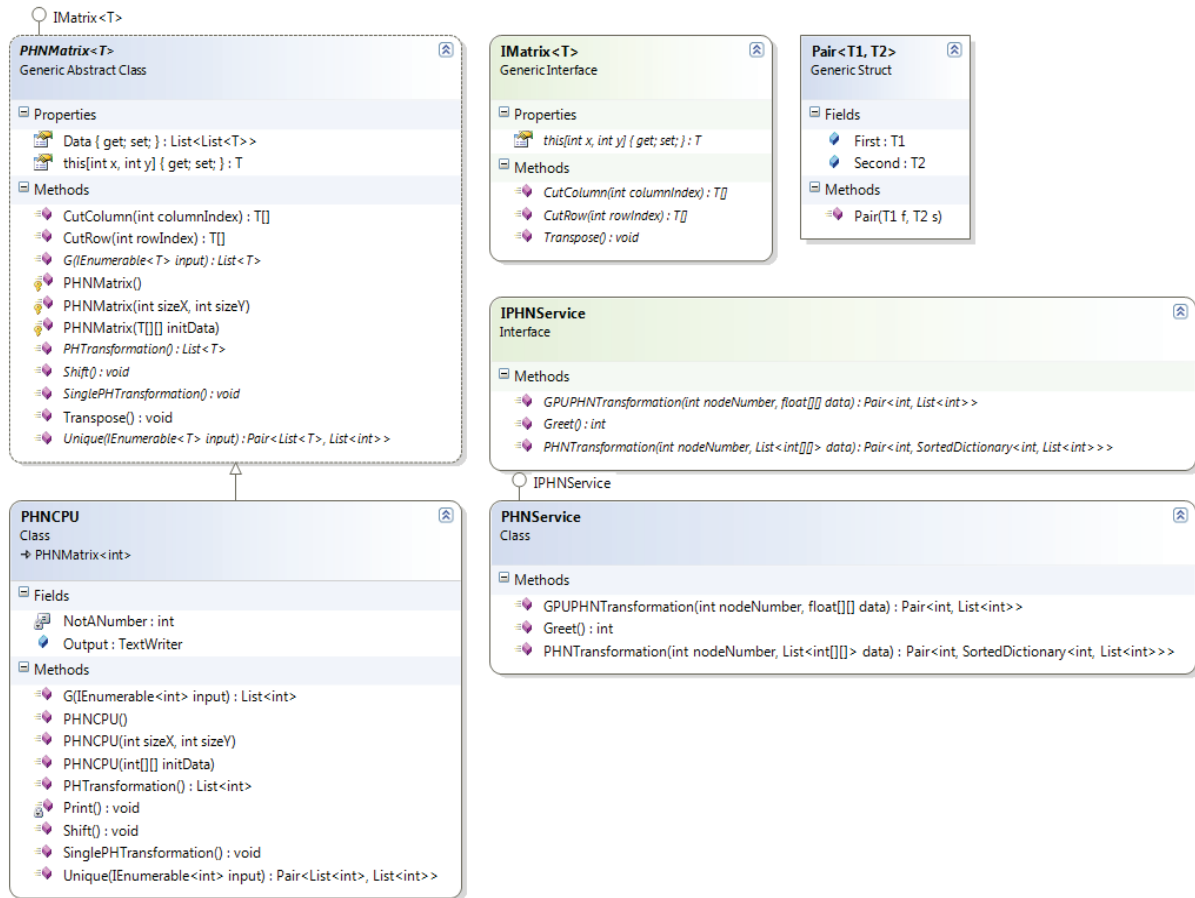


Рис. 5. Класова діаграма сервісу по виконанню прямого паралельно-ієрархічного перетворення

Програмний комплекс може виконувати такі функції:

1) завантаження зображення довільної розмірності;
 2) розподіл цього зображення на вікна розмірністю 128x128 пікселів та розподіл завдань між обчислювальними вузлами;

3) виконання на кожному обчислювальному вузлі прямого паралельно-ієрархічного перетворення над переданою частиною зображення розмірністю 128x128 пікселів;

4) збирання на контрольному вузлі результатів перетворення із кожного обчислювального вузла.

Вимоги до мінімального складу та параметрів технічних засобів:

- Операційна система: Windows XP, Vista, 7.
- Програмне середовище Microsoft .NET Framework 3.5 або 4.
- Не менше 1 Гб ОЗУ на кожному із обчислювальних вузлів та 512 Мб ОЗУ – на контрольному вузлі.

- Процесор з тактовою частотою не нижче 1 ГГц.

- Відеокарта із підтримкою DirectX 9.0c та піксельних шейдерів версії 3.0.

- 100 Мб вільного дискового простору на контрольному вузлі та 10 Мб на кожному із обчислювальних вузлів.

- Локальна мережа пропускною здатністю не менше 100 Мбіт/с (краще – 1 Гбіт/с).

- IP-адреси усіх обчислювальних вузлів у мережі повинні відрізнятися лише молодшим октетом (тобто, мережа має бути класу C).

Обмеження топології мережі кластерної системи визначаються користувачем, виходячи із гранично можливих апаратних ресурсів. Тобто, означена обчислювальна платформа у найпростішому випадку має складатися з деякої кількості машин-виконавців (host-комп'ютерів) та машини-координатора (сервера). Такий апаратно-програмний комплекс є найбільш доступним, його простіше

програмувати. При організації розподіленої обробки інформації на основі технологій GPGPU, кожна машина-виконавець має містити хоча б один відеоадаптер, а завдання для кожного виконавця бути паралельною відносно даних програмою, яку можливо виконати на відеоадаптері (шейдером). При цьому програма для кластера, має керувати виконанням кожного завдання на окремому виконавці. Основа логічної структури системи відповідає вищезначеній проблематиці, тобто на машині-координаторі має бути запущено програму, яка б дозволила відсилати машинам-виконавцям завдання на виконання. Кожне завдання – це шейдер та дані для нього. У найпростішому випадку усі виконавці отримують однакові шейдери, це добре узгоджується з паралельною відносно даних моделлю виконання шейдерів.

Таким чином, основною задачею розробленого кластерного обчислювального комплексу є реалізація прямого паралельно-ієрархічного перетворення зображень, зокрема для вузькоспецифічної проблеми ефективної обробки плямоподібних зображень відео-траси лазерного променя в реальному часі у системах профілювання лазерів. Разом з тим функціональні можливості запропонованого обчислювального комплексу дозволяють здійснювати обробку інших типів зображень, в тому числі багатокольорових, для інших прикладних задач. Також запропонований програмний комплекс можна використовувати для подальшої розробки моделей інтелектуальних паралельно-ієрархічних систем та ієрарх-ієрархічних систем.

ВИСНОВКИ

В ході проведених наукових досліджень було проаналізовано методологічні і прикладні аспекти та перспективи побудови паралельно-ієрархічних та ієрарх-ієрархічних мереж на основі кластерних систем з розподіленою обробкою інформації. На основі проведеного аналізу розвитку інтелектуальних інформаційних технологій визначено, що останнім часом особливої актуальності набуває створення гібридних обчислювальних комплексів на основі інтеграції парадигм паралельної та розподіленої обробки інформації, а також технологій GPGPU для подальшої реалізації швидкої та ефективної паралельної обробки надвеликих масивів інформації різної фізичної природи.

Особливу увагу приділено організації в межах кластерного обчислювального комплексу паралельної обробки інформації на основі технологій GPGPU, зокрема для обробки інформації візуальної природи. Водночас, в практичній сфері можна виділити багато інших завдань, які дуже добре масштабуються і на GPU-орієнтованих кластерних платформах, тобто в розподіленому обчислювальному середовищі, побудованому із графічних процесорів. На основі проведених досліджень можна впевнено стверджувати, що GPU-орієнтовані кластерні платформи дозволяють отримати високу пікову продуктивність, незважаючи на певні складнощі фізичної організації каналів взаємодії GPU – CPU. Зокрема, компактне виконання з високою обчислювальною щільністю поряд з високошвидкісними каналами передачі даних дозволяють використати проектні рішення на базі GPU у компактних кластерних комплексах для вирішення задач ефективного опрацювання великих масивів даних та їх паралельної обробки [5,8,13,14].

Наведено результати програмної частини розробки кластерного обчислювального комплексу з багатофункціональним набором можливостей: розподіленої обробки інформації в межах обчислювальної мережі та паралельної обробки інформації на host-комп'ютері на основі потужностей відеоадаптера, що надає можливості об'єднання в межах обчислювального кластера host-комп'ютерів, кожен з яких досягає окремих показників пікової продуктивності супер-комп'ютерів (на основі технологій GPGPU). Вказаний кластерний обчислювальний комплекс застосовано для виконання паралельно-ієрархічного перетворення інформаційних середовищ із використаннями на кожному host-комп'ютері, за вибором користувача, обчислювальних потужностей як його центрального процесора (CPU), так і відеоадаптера (на основі технологій GPGPU), а також подальшої розробки моделей інтелектуальних паралельно-ієрархічних та ієрарх-ієрархічних нейрорподібних систем.

СПИСОК ЛІТЕРАТУРИ

1. Розподілені комп'ютерні системи як складові інформаційних інфраструктур / В.П. Горбулін, О.Г. Додонов, О.С. Горбачик, М.Г. Кузнецова // Реєстрація, зберігання і обробка даних. – 2008. – Т. 10, № 4. – С. 19-24. – ISSN 1560-9189.
2. Хьюз К. Параллельное и распределенное программирование на C++. / Хьюз К., Хьюз Т. [Пер. с англ.] – М.: ИД «Вильямс», 2004. – 672 с.
3. Воеводин В.В. Параллельные вычисления. / В.В.Воеводин, Вл.В.Воеводин – СПб.: БХВ-Петербург,

2002. – 608 с.
4. 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. – ISBN 978-966-641-261-7.
 5. Яровий А.А. Прикладні аспекти і перспективи побудови кластерів на основі GPU для реалізації паралельної та паралельно-ієрархічної обробки інформації / Яровий А.А. // Оптико-електронні інформаційно-енергетичні технології. – 2009. – №2 (18). – С. 119-126.
 6. Бройнль Т. Паралельне програмування: Навч. посібник / Бройнль Т. [Пер. з нім. В.А. Святого] – К.: Вища школа, 1997. – 358 с.
 7. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : [Монографія.] / В.П. Кожем'яко, Ю.Ф. Кутаєв, С.В. Свечніков, Л.І. Тимченко, А.А. Яровий – Вінниця: УНІВЕРСУМ-Вінниця, 2003. – 324 с.
 8. Методологічні особливості реалізації нейроподібних паралельно-ієрархічних систем на основі технологій GPGPU / Кожем'яко В.П., Яровий А.А., Богомолів Ю.С., Вознесенський К.Ю. // Оптико-електронні інформаційно-енергетичні технології. – 2008. – №2 (16). – С. 26-33.
 9. Свідоцтво про реєстрацію авторського права на твір № 31705. Комп'ютерна програма “Програмна бібліотека для організації паралельної обробки інформації на основі програмування відеоадаптерів АТІ („useGPU”)” / Яровий А.А., Вознесенський К.Ю., Богомолів Ю.С. Дата реєстрації Державним Департаментом інтелектуальної власності України 21.01.2010.
 10. Свідоцтво про реєстрацію авторського права на твір № 31706. Комп'ютерна програма “Програмна бібліотека для побудови GPU-кластера із організацією розподіленої обробки інформації на основі програмування віддалених відеоадаптерів АТІ („dotGPU”)” / Яровий А.А., Вознесенський К.Ю. Дата реєстрації Державним Департаментом інтелектуальної власності України 21.01.2010.
 11. Тимченко Л.І. Паралельно-ієрархічне перетворення як системна модель ефективних засобів штучного інтелекту : автореф. дис. На здобуття наук ступеня доктора техн. наук : спец. 05.13.23 / Тимченко Леонід Іванович ; Державний науково-дослідний інститут інформаційної інфраструктури. – Львів, 2002. – 36 с.
 12. Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідально сільового перетворення зображень. / Кожем'яко В.П., Тимченко Л.І., Кутаєв Ю.Ф., Івасюк І.Д. – К: УМК ВО, 1994. – 272 с.
 13. Прикладна реалізація масштабних нейронних та нейроподібних паралельно-ієрархічних мереж на основі технологій GPGPU [Електронний ресурс] : [Електронне наукове фахове видання] / А. А. Яровий, Ю. С. Богомолів, К. Ю. Вознесенський // Наукові праці Вінницького національного технічного університету. – 2009. – №2. – С. 1-8. – Режим доступу до журн.: http://www.nbu.gov.ua/e-journals/VNTU/2009_2_ua/2009-2.files/uk/09aayogt_ua.pdf.
 14. Яровой А.А. Прикладные аспекты программно-аппаратной реализации нейроподобных параллельно-иерархических систем : Сборник научных трудов. В 2 частях. Ч. 2. [Научная сессия МИФИ – 2009. XI Всероссийская научно-техническая конференция «Нейроинформатика-2009»], (Москва, 27-30 января 2009 г.) – М.: МИФИ, 2009. – С. 39-48.

Надійшла до редакції 03.02.2010 р.

ЯРОВИЙ А.А. – к.т.н., доцент, директор ЦРІС ГЦМЗ, доцент кафедри комп'ютерних наук, науковий співробітник кафедри лазерної і оптоелектронної техніки, Вінницький національний технічний університет, Вінниця, Україна.