Компьютерная Математика

УДК 681.3.06

Р.М. ПОНОМАРЕНКО

МОДЕЛІ ПАРАЛЕЛЬНИХ ІЄРАРХІЧНИХ СИСТЕМ ДЛЯ НЕЧІТКОГО ЛОГІЧНОГО ВИВЕДЕННЯ

Розглядаються запропоновані моделі паралельних ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів Nvidia, з використанням технології CUDA за умов складних графів залежностей між блоками нечітких правил. Розроблено експериментальну програмну систему, яка реалізує модель обчислень ієрархічних систем на основі елементарних систем нечіткого логічного виведення Такагі – Сугено. Отримано теоретичні та експериментальні оцінки прискорення обчислень для розробленої моделі та проведена порівняльна характеристика серед інших технологій паралельних обчислень.

Вступ. На сьогоднішній час, у світі нечітка логіка широко застосовується в різних галузях промисловості при побудові інтелектуальних систем, що працюють за умов невизначеності. Застосовується в експертно-діагностичних, мультиагентних системах, для прийняття рішень за умов невизначеності тощо [1, 2]. При цьому, основою для обчислення таких задач є системи нечіткого логічного виведення, такі, як, наприклад, Мамдані, Такагі – Сугено, бази знань яких ґрунтуються на основі нечітких предикатних правил «якщо-то», які, в свою чергу, розробляються експертом конкретної предметної області на основі власних знань та досвіду [3]. Перед безпосередньою побудовою блоків нечітких правил, проводиться аналіз вхідного та вихідного простору задачі, з подальшим визначенням вхідних множини вхідних та вихідних нечітких змінних. Проте, при збільшенні кількості вхідних множин хоча б до 4 – 5, кількість предикатних правил, мінімально необхідних для покриття множини варіантів можливих вхідних значень різко зростає. Що, в свою чергу, унеможливлює написання блоків правил такої кількості, та не дає можливості для побудови нечітких моделей для розв'язку задач великої розмірності.

Одним із способів подолання великої кількості правил у межах елементарної нечіткої системи є проведення декомпозиції елементарної системи на декілька підсистем [4]. Тобто розподілення правил або змінних на різних рівнях, відповідно до їх специфічності, деталізації, релевантності тощо.

Зазначимо, що блок правил з трьома входами та трьома нечіткими термами для повного покриття всіх можливих комбінацій має містити 27 правил. Для блоків правил з двома вхідними змінними кількість правил не перевищує дев'яти. Тобто, в загальному випадку, кількість правил регламентується виразом – l^n, де l – кількість лінгвістичних термів, n – кількість вхідних змінних. Для більшості випадків, кількість вхідних змінних має не перевищувати трьох. Таким чином, будуються ієрархічні системи нечіткого логічного виведення, елементарні системи яких мають логічні зв'язки входів та виходів, у яких як вхідні значення передаються вихідні значення систем, нижчих за ієрархією.

Збільшення розмірності задач, які дозволяють вирішувати ієрархічні системи нечіткого логічного виведення, вимагає проведення великої кількості обчислень та суттєвого часу їх виконання. В роботах [5, 6] за участі автора проводилися дослідження прискорення нечітких ієрархічних систем. Розроблено методи прискорення виконання ієрархічних нечітких систем типу Такагі – Сугено з використанням технології розподілених обчислень MPI.

Мета даної роботи – дослідження та обґрунтування методів обчислень ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів Nvidia. При побудові залежностей між нечіткими блоками правил була використана ярусно-паралельна модель обчислень. У даній роботі як елементарні нечіткі системи використовуються системи типу Такагі – Сугено [3].

Як зазначено вище, в даній роботі була використана ярусно-паралельна форма алгоритму, тобто перед безпосереднім виконанням паралельної обробки нечіткої ієрархічної системи спочатку дана система аналізується на предмет можливої паралелізації, будується ярусно-паралельна схема, яка аналізує ієрархію блоків нечітких правил, та будує структуру даних, якою керується програма в подальшому для визначення послідовності виконання блоків правил та можливостей паралельної їх обробки.

Будь-який алгоритм можна представити у вигляді дерева ярусів. Алгоритм представляє собою у загальному вигляді деяку множину операцій або блоків операцій, кожна з яких потребує перед виконанням множину вхідних аргументів, які можуть бути отримані або з множини вхідних даних, або надані іншими операціями постачальниками даних. Далі в роботі для спрощення будемо розуміти під операцією як одну, так і блок операцій алгоритму. Зрозуміло, що операція приймач даних не може бути виконана раніше, ніж виконаються всі операції постачальники [7]. Під ярусно-паралельною схемою розуміється дерево ярусів, яке представляє собою орієнтований ациклічний мультиграф $G = (V, E)$, що має множину вершин $V = \{v_i\}, i = \overline{1, N}$ та множину дуг $E = \{e_j\}, j = \overline{1, M}$, де N – множина вершин, M – множина дуг графа G. Під множиною вершин графа розуміється множина операцій алгоритму. Під дугою графа розуміється залежність виконання однієї операції від попереднього виконання другої операції, яка є постачальником аргументів для першої. Кількість аргументів кожної операції відповідає множині дуг, які входять у відповідну вершину графа.

Введемо поняття ярусу ярусно-паралельної форми обчислень. Нехай задано ациклічний орієнтований граф, який має N вершин. Тоді кожну вершину графа можна помітити індексом $S = \{s_m\}$, $m = \overline{1, M}$, $s \le N$ таким, що, якщо з вершини v_k з індексом s_i іде дуга до вершини v_l з індексом s_j, то $s_i < s_j$. При цьому, множина вершин з однаковим індексом називається ярусом ярусно-паралельної форми алгоритму, множина S являє собою множину ярусів алгоритму, M – кількість ярусів. Звідси бачимо, що множину операцій, які відповідають множині вершин одного ярусу, можна виконати паралельно. Якщо всі операції алгоритму мають бути виконані послідовно, то відповідно $M = N$, та $\forall s_m = 1, m = \overline{1, M}$.

Для запису та зберігання даних про побудовану ярусно-паралельну схему автором розроблено підхід, використовуючи який, можна якісно описати граф, та швидко отримувати доступ до будь-якої вершини будь-якого ярусу графа. Зазначимо, що структура даних, яка застосовується при цьому підході, складається з двох одномірних масивів даних: масиву вершин та масиву номерів вершин.

Нехай N – кількість вершин графа (кількість блоків правил в ярусі), s – кількість ярусів. Тоді, якщо кількість систем в i-му ярусі – l_i, $i = \overline{1, s}$, то маємо наступне співвідношення:

$$\sum_{i=1}^{s} l_i = N.$$

Нехай V – масив вершин, M – масив номерів вершин. Визначимо для кожного ярусу i-ту групу вершин $D_i = \{d_p^{(i)}\}$, $p = \overline{1, l}$ в масиві V. Тоді кожна вершина i-го ярусу в масиві V знаходитися за наступною формулою:

$$d_p^{(i)} = M_{z_p^{(i)}}, \; z_p^{(i)} = p + \sum_{k=1}^{i} l_k, \qquad (1)$$

де i – номер ярусу, p – номер вершини в i-му ярусі, $z_p^{(i)}$ – індекс номера вершини p i-го ярусу в масиві M. $M_{z_p^{(i)}}$ – знайдений індекс вершини $d_p^{(i)}$ в масиві V.

Архітектура графічних прискорювачів Nvidia дозволяє, використовуючи технологію паралельного багатопотокового програмування CUDA, здійснювати обчислення математичні безпосередньо на графічному пристрої з метою прискорення їх виконання. Має суттєві переваги, такі як відносно недорогі обчислювальні ресурси та велика кількість графічних процесорів, які можуть одночасно виконувати паралельні операції з даними.

Основна ідея методу побудови обчислень нечітких ієрархічних систем на основі архітектури графічних прискорювачів полягає у використанні ярусно-паралельної алгоритмічної форми при паралельному обчисленні блоків нечітких правил у межах кожного ярусу, та водночас паралельному обчисленні всіх правил в межах кожного блоку нечітких правил (внутрішній паралелізм).

Покажемо це на прикладі ієрархічних систем нечіткого логічного виведення, що складаються з елементарних нечітких систем типу Такагі – Сугено. Системи нечіткого виведення складаються з набору правил якщо-то. Нечітка модель Такагі – Сугено представлена нечіткими правилами [2]:

$$R_j: \text{якщо } x_1 \in A_{1j} \text{ та } x_2 \in A_{2j} \text{ та ... та } x_n \in A_{nj},$$
$$\text{то } y = g_j(x_1, x_2, ..., x_n), \quad j = 1, 2, ..., N,$$

де g_j – чітка функція від x_i. Як правило, $g_j(x_1, x_2, ..., x_n) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + ... + \omega_n x_n$. Загальний вихід нечіткої моделі отримують наступним чином:

$$y = \sum_{j=1}^{N} g_i \overset{m_j}{\underset{i=1}{T}} \mu_{ij}(x_i) \Big/ \sum_{j=1}^{N} \cdot \overset{m_j}{\underset{i=1}{T}} \mu_{ij}(x_i), \tag{2}$$

де $1 \le m_j \le n$ – це число вхідних змінних, які з'являються в антецеденті правила *j*, *N* – кількість нечітких правил, *n* – кількість входів, функції належності нечітких множин, *T* являє собою *T*-норму, що використовується як операції кон'юнкції.

Нехай $B = \{b_h\}, h = \overline{1, H}$, де *H* – кількість виділених паралельних обчислювальних блоків (blocks) графічного пристрою, яка дорівнює кількості вершин найширшого з ярусів системи. Кожному обчислювальному блоку, що в свою чергу складається з паралельно виконуваних ниток (threads), необхідно виконати обчислення елементарної нечіткої системи. Нехай $Y = \{y^q\}, q = \overline{1, Q}$, де *Y* – множина всіх вихідних значень елементарних нечітких систем (блоків нечітких правил), *Q* – загальна кількість нечітких систем. Тоді обчислення кожного *i*-го ярусу зводиться до паралельного обчислення множини блоків правил цього ярусу від 1 до l_i, де l_i – ширина ярусу *i*, та знаходження відповідних вихідних значень $y^{(q)}$ за формулою (2). Оскільки структура блоків правил представляє собою одномірний масив, кожний обчислювальний блок (block) графічного процесору виконує операції за формулою:

$$y^{(q)} = \sum_{j=1}^{N^{(q)}} g_j^{(q)} \overset{m_j}{\underset{i=1}{T}} \mu_{ij}^{(q)}(x_i) \Big/ \sum_{j=1}^{N^{(q)}} \cdot \overset{m_j}{\underset{i=1}{T}} \mu_{ij}^{(q)}(x_i),$$

де індекс *q* знаходиться за формулою (1).

Водночас, для забезпечення збільшення прискорення виконання ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів, запропоновано концепція внутрішнього паралелізму, тобто проводити обчислення

кожного правила в межах елементарної системи паралельно. Метод внутрішнього паралелізму полягає у тому, що кожна нитка графічного обчислювального пристрою паралельно з іншими виконує операції обчислення одного правила. Таким чином виконання всього блоку правил виконується паралельно. Нехай y_j – результат виконання обчислень кожного нечіткого правила, які виконуються на окремій нитці (thread) графічного прискорювача. Позначимо множину паралельних ниток як $t_z, z = \overline{1,R}.$ Зазначимо, що кількість паралельних ниток для забезпечення найкращого прискорення має бути не менше, ніж кількість правил у межах блоку. Тоді $y = \sum_{j=1}^{N} y_j,$ де кожна нитка t_z виконує наступні операції:

$$y_j = \frac{g_j \prod_{i=1}^{m_j} \mu_{ij}(x_i)}{\prod_{i=1}^{m_j} \mu_{ij}(x_i)}.$$

Останнім кроком обчислення блоку правил стане знаходження значення *y*. Цю операцію за запропонованою схемою виконує перша нитка. Оскільки дана операція – залежна від операцій постачальників $y_j,$ вона буде виконана тільки після їх синхронізованого виконання. Зазначимо, що реалізація внутрішнього паралелізму методами розподіленого багатопроцесорного програмування на основі обміну повідомленнями неможлива за рахунок складності налагодження механізму передачі даних між процесорами. Також багатопроцесорні ресурси є суттєво дорожчими в порівнянні з технологічними ресурсами багатопотокового програмування на основі графічних прискорювачів.

Побудовано програмну систему для експериментальних досліджень прискорення за умов складних графів залежностей між блоками нечітких правил. На рисунку показано експериментальну оцінку прискорення обчислення ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів у порівнянні з прискоренням, яке отримано при використанні розподілених обчислень на основі обміну повідомленнями, з кількістю процесорів, яка дорівнює двадцяти чотирьом. При проведенні експериментальних досліджень оцінки прискорення використовувалася відеокарта GTX 1050Ti архітектури останнього покоління Pascal 8.0, яка містить 768 графічних процесорів. За рахунок великої кількості процесорів, а також реалізації паралельної організації обчислень кожного блоку правил (внутрішнього паралелізму) бачимо, що прискорення обчислення ієрархічної нечіткої системи розмірності 1000 вершин для графічних прискорювачів досягає майже 600 разів у порівнянні з послідовним варіантом. Обчислення ієрархічної нечіткої системи при використанні технології розподілених обчислень (MPI) за однакових умов не перевищує 20 разів.

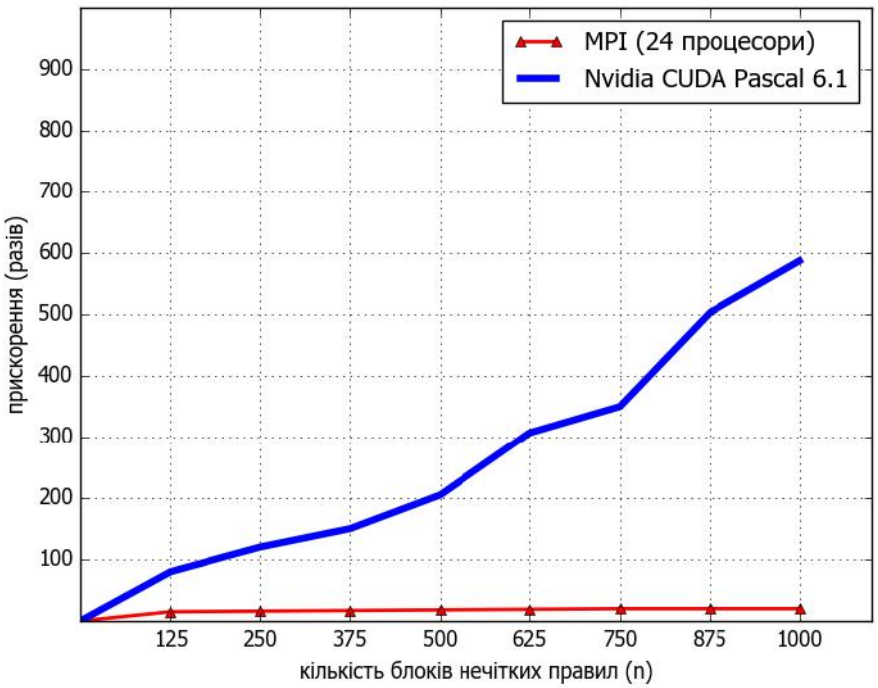

РИСУНОК. Залежність часу виконання обчислень для ієрархічної нечіткої системи з великою кількістю вершин (v = 1000)

Проведемо теоретичну оцінку прискорення виконання обчислень ієрархічних систем нечіткого логічного виведення. Візьмемо за основу закон Амдала [7], який ілюструє обмеження прискорення обчислень при збільшенні їх кількості:

$$S_p = \frac{1}{\alpha + \frac{1-\alpha}{p}} = \frac{p}{\alpha * p + 1 - a}, \tag{3}$$

де S_p – прискорення виконання обчислень за умови використання p процесорів, α – частка обчислень, які мають бути виконані послідовно, p – кількість процесорів.

Для обчислення прискорення візьмемо за елементарну частку обчислень спочатку весь блок правил. Тоді частка послідовно виконаних обчислень буде розраховуватися за наступною формулою: $\alpha = \frac{M}{s}$, де M – кількість нечітких блоків правил, s – кількість ярусів ярусно-паралельної схеми. Однак, метод

обчислення ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів передбачає внутрішній паралелізм, тобто відбувається паралелізація в кожному окремому блоці правил, тобто кожної елементарної частки також може бути представлено формулою (3). Автором запропоновано теорему, за якою сумарне прискорювання алгоритму, який реалізує внутрішній паралелізм розмірності n розраховується за наступною формулою:

$$S_P^{sum} = s_{p_1}^1 * s_{p_2}^2 * s_{p_n}^n = \prod_{i=1}^{n} s_{p_i}^i, \tag{4}$$

де $P = \{p_i\}, i = \overline{1,n}$ – кількість процесорів, які виділені на кожному рівні внутрішнього паралелізму.

Доведення. Нехай візьмемо обчислюваний алгоритм, який має внутрішній паралелізм $n = 2$ рівнів. Тобто кожна елементарна операція також може бути обчислена частково паралельно. Оскільки кожна елементарна операція першого рівня, яка в формулі (3) позначається як α з коефіцієнтом 1, який у свою чергу розраховується за формулою, оберненою до формули 3, матимемо вираз:

$$S_P^{sum} = \frac{1}{\alpha_1 * \left(\frac{1}{\alpha_2 + \frac{1-\alpha_2}{p_2}} \right)^{-1} + \frac{(1-\alpha_1) * \left(\frac{1}{\alpha_2 + \frac{1-\alpha_2}{p_2}} \right)^{-1}}{p_1}} =$$

$$= \frac{1}{\alpha_1 * \left(S_{p_2}^2\right)^{-1} + \frac{(1-\alpha_1) * \left(S_{p_2}^2\right)^{-1}}{p_1}} = \frac{1}{\frac{\alpha_1 * \left(S_{p_2}^2\right)^{-1} p_1 + (1-\alpha_1) * \left(S_{p_2}^2\right)^{-1}}{p_1}} =$$

$$= \frac{p_1}{(\alpha_1 * p_1 + (1-\alpha_1)) * \left(S_{p_2}^2\right)^{-1}} = \frac{S_{p_2}^2 * p_1}{a_1 p_1 + 1 - \alpha_1} = S_{p_1}^1 * S_{p_2}^2.$$

Теорема доведена.

Скористаємося формулою (4) для отримання теоретичної оцінки прискорення обчислень ієрархічних систем нечіткого логічного виведення з внутрішнім паралелізмом. Оскільки кожний блок правил виконується паралельно, то рівень внутрішнього паралелізму буде $n = 2$. Тоді для системи з кількістю вершин $M = 1000$, яка має кількість ярусів $s = 20$ та кількість правил кожного блоку $k = 25$ матимемо:

$$\alpha_1 = \frac{1}{M/s} = \frac{1}{1000/20} = 0,02;$$

$$\alpha_2 = \frac{2}{k+1} = \frac{2}{26} = 0,077,$$

де число 2 являє собою суму операцій обчислення кожного правила з операцією редукції для знаходження остаточного вихідного значення. Тоді, при можливості застосувати всі 768 процесорів відеокарти при обчисленні одного ярусу, мінімальне прискорення розрахуємо наступним чином:

$$S_P^{sum} = s_{p_1}^1 * s_{p_2}^2;$$

$$p_1 = 768 / k = 768 / 25 = 30;$$

$$p_1 = k = 25;$$

$$s_{p_1}^1 = \frac{1}{\alpha_1 + \frac{1-\alpha_1}{p_1}} = \frac{1}{0,02 + \frac{1-0,02}{30}} \approx 20;$$

$$s_{p_{21}}^2 = \frac{1}{\sim_2 + \frac{1-\sim_2}{p_2}} = \frac{1}{0,077 + \frac{1-0,077}{30}} \approx 10;$$

$$S_P^{sum} = 20 * 10 \approx 300 \text{ разів.}$$

Як видно із розрахунків, теоретична оцінка мінімального прискорення сягає 300 разів, тоді як експериментальна оцінка прискорення сягає до 600 раз. Дана розбіжність пояснюється тим, що в експериментальних умовах при нормальному розподілі інформаційних зв'язків між блоками нечітких правил, кількість вершин у кожному ярусі розподіляється у багатьох випадках рівномірно.

Висновки. Запропоновано, та реалізовано метод обчислення ієрархічних систем нечіткого логічного виведення на основі архітектури графічних прискорювачів Nvidia. Побудовано програмну систему, яка реалізує виконання паралельних обчислень нечітких ієрархічних систем. Обґрунтовано застосування такого підходу, отримано експериментальні та теоретичні оцінки прискорення виконання обчислень на архітектурі графічних прискорювачів за умов складних графів залежностей між блоками правил нечітких ієрархічних на прикладі застосування елементарних систем нечіткого логічного виведення Такагі – Сугено. Автором в [5, 6] вже проводилися дослідження прискорення нечітких ієрархічних систем нечіткого логічного виведення на базі технології розподілених обчислень MPI. Встановлено та експериментально підтверджено, що побудова ієрархічних систем нечіткого логічного виведення на основі графічних прискорювачів дозволяє значно пришвидшити виконання обчислень, порівняно з методами

обчислень з використанням технології MPI. Це дозволяє будувати набагато більш складні та навантажені системи, що значно розширює коло задач, які можуть бути вирішені методами нечіткої логіки.

Р.Н. Пономаренко

МОДЕЛИ ПАРАЛЛЕЛЬНЫХ ИЕРАРХИЧЕСКИХ СИСТЕМ
ДЛЯ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА

Рассматриваются предложенные модели параллельных иерархических систем нечеткого логического вывода на основе графических ускорителей Nvidia, с использованием технологии CUDA, при условии сложных графов зависимостей между блоками нечетких правил. Разработано экспериментальную программную систему, реализующую модель вычислений иерархических систем на основе элементарных систем нечеткого логического вывода Такаги – Сугено. Получены теоретические и экспериментальные оценки ускорения для разработанной модели, а также проведена сравнительная характеристика среди других технологий параллельных вычислений.

R.M. Ponomarenko

MODELS OF PARALLEL HIERARCHICAL SYSTEMS OF FUZZY LOGIC

The models of parallel hierarchical systems of fuzzy logic inference based on NVIDIA graphic accelerators using CUDA technology under the conditions of complex graphs of dependencies between the blocks of fuzzy rules are considered. An experimental program system was developed that implements the model for calculating hierarchical systems based on Takagi-Sugeno's elementary systems of fuzzy logic inference. The theoretical and experimental estimations for acceleration of calculations for the developed approach are obtained and the comparative characteristic among other technologies of parallel computing is carried out.

1. Ершов С.В. Модель интеллектуальных агентов, основанная на нечеткой логике высшего типа. *Компьютерная математика*. 2012. № 1. С. 10 – 16.
2. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия. Телеком, 2007. 288 с.
3. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ, 2005. 736 с.
4. Torra V. A review of the construction of hierarchical fuzzy systems. *Int. J. Intell. Syst.* 2002. N 17. P. 531 – 543.
5. Єршов С.В., Пономаренко Р.М. Паралельні моделі багаторівневих нечітких систем Такагі – Сугено. *Проблеми програмування*. 2016. № 1. С. 141 – 149.
6. Єршов С.В., Пономаренко Р.М. Ярусно-паралельна модель обчислень для логічного виведення у нечітких багаторівневих системах. *Компьютерная математика.* 2016. № 1. С. 28 – 36.
7. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.

Одержано 26.10.2017

Про автора:

Пономаренко Роман Миколайович,
аспірант Інституту кібернетики імені В.М. Глушкова НАН України.
E-mail: ponomarenko_roman@ukr.net