

УДОСКОНАЛЕНА МОДЕЛЬ ОПТИМАЛЬНОГО ВИКОРИСТАННЯ СЕРВЕРНИХ РЕСУРСІВ КЛАСТЕРНОЇ СИСТЕМИ ВІЙСЬКОВОГО ПРИЗНАЧЕННЯ НА ОСНОВІ РІВНОВАГИ НЕША

У статті вирішується актуальне наукове завдання оптимізації використання серверних ресурсів кластерної системи військового призначення у процесі обробки множини клієнтських запитів. Постановка даного наукового завдання зумовлена необхідністю усунення залежності показників ефективності існуючих методів балансування навантаження кластерних систем від якості навчання, а також досить вузьким підходом аналізованих наукових досліджень до вивчення впливу властивостей верифікованості, робастності та гарантії оптимуму на кінцевий результат прийняття рішення. Проведено дослідження існуючих підходів до оптимального використання ресурсів кластерної системи. За результатами аналізу виділено методи, в основу яких покладено теоретико-ігровий науково-методичний апарат, як найбільш доцільний підхід до формального опису функціональної взаємодії серверів із клієнтськими запитами, а також як підхід, що не потребує узагальнення емпіричного досвіду (навчання).

Запропоновано модель оптимального використання ресурсів кластерної системи військового призначення, яка ґрунтується на гібридизації застосування теоретико-ігрового підходу і теорії нечіткої логіки. Суть запропонованої моделі полягає у визначенні необхідності декомпозиції ініційованого клієнтським запитом завдання на підзадачі засобами теорії нечіткої логіки з метою подальшого знаходження оптимального розподілу використання серверів кластерної системи на основі рівноваги Неша, із врахуванням узагальнюючого показника поточного рівня їх завантаженості. Отримані результати розглядаються як підґрунтя для реалізації нових адаптивних систем балансування навантаження вузлів кластерної системи, застосування яких дозволяє оптимально використовувати ресурси кластерної системи військового призначення у стійкій рівновазі.

Ключові слова: знаходження оптимуму, кластер, теорія ігор, рівновага Неша, нечітка логіка.

Вступ і постановка завдання. На сьогоднішній день, у процесі забезпечення стабільного (неперервного/надійного) функціонування інформаційних систем (ІС) критичної інфраструктури, залишається невирішеним завдання забезпечення належного рівня функціональної стійкості ІС військового призначення (сил оборони та безпеки держави), особливо у воєнний час.

Одним із пріоритетних напрямків рішення даного завдання є пошук оптимального використання наявних серверних ресурсів кластерних систем шляхом управління процесом балансування виконання множини задач, що підлягають обробці [1]. Так, стек сучасного програмного забезпечення (ПЗ) інформаційної взаємодії органів військового управління з підлеглими підрозділами у результаті інтенсивності і складності запитів може вимагати значних обчислювальних потужностей, особливо у випадку обробки великих обсягів даних (*Big Data*), що негативно впливає на продуктивність і ефективність інформаційних та обчислювальних систем, інтегрованих у військову галузь.

Це обумовлює актуальність подальших наукових досліджень щодо підвищення ефективності оптимального використання серверних ресурсів кластерних систем військового призначення.

Аналіз останніх досліджень та публікацій [1 – 9] за даною тематикою показав значну перевагу застосування адаптивних алгоритмів (способів, моделей, методів, методик) балансування навантаження кластерних систем у порівнянні із найбільш поширеними і водночас простішими алгоритмами, такими, як *Round Robin (Weighted Round Robin)* і *Least*

Connection (Weighted Least Connection). Так, підвищення ефективності адаптивних методів балансування навантаження кластерних систем досягається шляхом покращення показників (властивостей):

продуктивності: передбачає здійснення розподілу навантаження між обчислювальними ресурсами таким чином, щоб кожен вузол кластера функціонував з максимальною ефективністю;

стійкості до відмов: здатність системи реагувати на зміни навантаження та оптимально перерозподіляти його між ресурсами у реальному часі;

економії ресурсів: використання наявних ресурсів з максимально можливим показником оптимуму (глобальний оптимум), що забезпечує скорочення витрат на обслуговування завдань;

адаптивності: налаштування на специфічні умови функціонування, що дозволяє системі швидко адаптуватися до змін навантаження і вимог користувачів;

точності: використання значної кількості ознак для прийняття рішень, що підвищує точність розподілу навантаження між ресурсами.

Поряд з цим, деякі із перелічених властивостей, в силу своєї природи, здатні негативно впливати на процес прийняття рішення балансувальником навантаження. Наприклад, властивості адаптивності і точності безпосередньо залежать від якості алгоритмів навчання, які використовуються в системі балансування навантаження, що цілком може свідчити про його недостатню функціональну стійкість за різних умов функціонування. У випадку застосування експертних систем піднімаються питання узгодженості суб'єктивних суджень експертів, мінімально необхідної їх кількості, а також відповідного рівня їх компетентності, тоді, як у випадку застосування методів машинного навчання піднімаються питання репрезентативності навчальної вибірки та інформативної значущості простору її ознак. До того ж, у процесі вибору найефективнішого підходу до визначення оптимального використання ресурсів кластерів у розглянутих наукових дослідженнях [2 – 10] недостатньо уваги приділяється наступним властивостям, які також безпосередньо впливають на кінцевий результат прийняття рішення балансувальником навантаження:

верифікованість – забезпечення можливості перевірки коректності роботи методу людиною (перевірка і підтвердження коректності виконання покладених на нього функцій);

якість навчання – показник ефективності (точність / достовірність / повнота / F-міра тощо) виконання методом поставлених перед ним завдань на основі наявних даних (визначається на основі порівняння згенерованих методом результатів з фактичними значеннями із навчальної вибірки (чим менша різниця між отриманими та фактичними значеннями, тим вища якість навчання методу));

робастність – характеристика незалежності впливу на результат дослідження різноманітних виключень (аномалій), стійкість до перешкод (здатність надавати правильні результати навіть за наявності деяких неправильних значень чи помилок у даних);

гарантія оптимуму – теоретичне обґрунтування, яке дозволяє стверджувати, що знайдене рішення оптимальне або близьке до оптимального для певного завдання (завжди у глобальному оптимумі).

На цій основі виникає необхідність порівняння існуючих адаптивних методів (алгоритмів) [2 – 10] оптимального використання серверних ресурсів кластерних систем на основі критеріїв *верифікованості, залежності від якості навчання, робастності та гарантії глобального оптимуму*. У таблиці 1 представлено результати порівняльного аналізу існуючих адаптивних методів (алгоритмів) [1 – 10] оптимального використання серверних ресурсів кластерних систем на основі вищезазначених критеріїв.

Результати порівняльного аналізу існуючих адаптивних методів оптимального використання серверних ресурсів

№ з/п	Найменування методу	Верифікованість	Залежність від якості навчання	Робастність	Гарантія глобального оптимуму
1	Регресійний аналіз	низька	висока	низька	середня
2	Системи масового обслуговування	середня	низька	висока	середня
3	Нейронні мережі	низька	висока	середня	середня
4	Нечітка логіка	висока	висока	висока	середня
5	Рівновага Неша (теорія Ігор)	висока	низька	висока	висока
6	Дерева рішень	висока	середня	низька	низька
7	Еволюційні алгоритми	низька	низька	висока	висока
8	Випадковий ліс	низька	середня	середня	середня
9	Аналіз ентропії	низька	середня	середня	середня
10	Байєсівські мережі	низька	середня	середня	середня
11	Експертні системи	висока	висока	висока	середня

На основі проведеного аналізу можна зробити висновок, що найбільш повну відповідність висунутим критеріям порівняння існуючих адаптивних методів оптимального використання серверних ресурсів демонструють методи, в основу яких покладено концепцію теорії Ігор щодо прийняття оптимальних рішень в умовах конфлікту – рівноваги Неша [1]. Даний науково-методичний апарат дозволяє ефективно знаходити оптимальне рішення використання серверів отриманими задачами, що є стійким до відхилень будь кого з інших учасників гри (будь-яке відхилення від обраних серверів не принесе додаткової користі для клієнтської задачі). Описана взаємодія є некооперативною грою, де гравці – отримані задачі, які намагаються мінімізувати власний час на обробку, а множина стратегій гри – сервери кластерної системи [1].

Вирішення завдання оптимального використання серверних ресурсів кластерної системи на основі рівноваги Неша представлено значною кількістю наукових досліджень проаналізованих у [1]. Однак, більшість із них не гарантує знаходження глобального оптимуму на профілі чистих стратегій. Крім того, у розглянутих наукових дослідженнях розглядаються питання балансування навантаження на рівні окремих завдань q_i і не

враховується можливість декомпозиції окремого завдання q_i – на підзавдання $sq_j^{q_i}$, що, у свою чергу, не забезпечує ефективне використання серверних ресурсів кластерної системи за показником рівномірного навантаження на вузли кластерної системи.

В [1] запропоновано підхід, який дозволяє досягти стійкої рівноваги оптимального використання серверних ресурсів із максимальною конвергенцією до глобального оптимуму шляхом декомпозиції кожної отриманої задачі q_i на підзавдання $sq_j^{q_i}$ з метою подальшого визначення імовірнісного розподілу паралельного використання ними серверів. Разом з тим, у запропонованому підході застосовується модель оптимального використання серверних ресурсів кластерної системи військового призначення на основі рівноваги Неша, яка базується на програмній моделі *MapReduce* [11] для проведення паралельної обробки великих наборів даних, що включає в себе функцію *Map()*, яка обробляє пару ключ/значення для створення набору проміжних пар ключ/значення, і функцію *Reduce()*, яка об'єднує всі проміжні значення, пов'язані з тим самим проміжним ключем.

Однак, даний метод адаптивного балансування навантаження кластерної системи не вирішує завдання визначення доцільності декомпозиції отриманої задачі у процесі імовірнісного розподілу паралельного використання серверів для існуючої моделі оптимального використання серверних ресурсів кластерної системи на основі рівноваги Неша, що, в свою чергу, не дозволяє досягти максимально рівномірного використання серверних ресурсів.

У зв'язку з цим, виникає актуальне наукове завдання удосконалення існуючої моделі оптимального використання серверних ресурсів кластерної системи військового призначення на основі рівноваги Неша.

Метою дослідження є обґрунтування доцільності декомпозиції завдань, які обробляються кластерною системою, при розрахунку імовірнісного розподілу паралельного використання серверів кластерної системи на основі рівноваги Неша.

Виклад основного матеріалу дослідження

Удосконалена модель оптимального використання серверних ресурсів кластерної системи військового призначення на основі рівноваги Неша. Для формулювання та вирішення поставленого наукового завдання формалізуємо процес оптимального використання серверних ресурсів кластерної системи у відповідності до [1], як некооперативну гру G у нормальній формі з ненульовою сумою, яка задана певною множиною підзадач $SQ_{q_i} = \{sq_1^{q_i}, sq_j^{q_i}, \dots, sq_2^{q_i}\}$ отриманого завдання q_i від користувача із відповідною вагою складності його обробки w_{q_i} , множиною стратегій їх виконання $S = \{s_1, \dots, s_i, \dots, s_n\}$ – серверами кластера, а також відповідними функціями їх виграшу $U_i: s_i \times \dots \times s_n \rightarrow \mathbb{R}$. Узагальнену платіжну матрицю гри G представлено на рисунку 1, де u_i – значення завантаженості сервера l_{s_i} (значення функції виграшу) для кожної підзадачі $sq_j^{q_i}$ у комбінації застосування серверів (стратегій) $s_i \times \dots \times s_n$, n – кількість серверів.

Суть запропонованого методу у [1] полягає в декомпозиції отриманої балансувальником задачі q_i на симетричні підзадачі $sq_j^{q_i}$ з метою подальшого їх виконання на множині змішаного застосування (імовірнісного розподілу) серверів S кластеру, які перебувають у рівновазі Неша. При чому об'єднання виконаних підзадач здійснюється програмним засобом для розподілених паралельних обчислень *MapReduce* [11].

q_i		$sq_2^{q_i}$		
		s_1	s_i	s_n
$sq_1^{q_i}$	s_1	u_{11}	u_{1i}	u_{1n}
	s_i	u_{i1}	u_{ii}	u_{in}
	s_n	u_{n1}	u_{ni}	u_{nn}

Рисунок 1 – Узагальнена платіжна матриця некооперативної гри G балансування навантаження на основі рівноваги Неша

Завдання визначення доцільності декомпозиції множини отриманих задач для подальшого їх виконання на множині змішаного застосування (імовірнісного розподілу) серверів S кластеру доцільно розглядати як завдання їх класифікації за типами:

прості – задачі, які не потребують значних обчислювальних ресурсів та/або задачі, які недоцільно декомпонувати (відео- / аудіо- потокові);

складні – задачі, які потребують значних обчислювальних ресурсів (наприклад, пов'язані з обробкою великих масивів даних (*Big Data*), з використанням інформаційно-аналітичних систем, складні запити до кластеру баз даних тощо).

Очевидно, отримані балансувальником задачі засобами наступних мережевих протоколів не потребують подальшої декомпозиції: Telnet; DNS; DHCP; SMTP; SNMP; SSH; NFS; NTP; SNTP; XMPP; LDAP; SIP; RTP; IMAP; POP3; SMB, оскільки у своїй більшості передбачають трансфер службової інформації між сервером та клієнтом, тоді як отримані задачі засобами протоколу FTP, а також складні задачі, що призначені для виконання серверною компонентою веб-сервісу потребують подальшої декомпозиції з метою забезпечення ефективного використання серверних ресурсів у рівновазі Неша з максимальною конвергенцією до глобального оптимуму.

Обмеження та допущення: гра G : некооперативна, біматрична, динамічна, несиметрична; вид стратегій – змішані; діапазон ваг складності w_{q_i} оцінки отриманих задач q_i встановлюється на основі попередньо отриманих статистичних значень використання серверних ресурсів типовими задачами у процесі тестування розгорнутих сервісів кластеру.

Необхідно: удосконалити існуючу модель оптимального використання серверних ресурсів кластерної системи на основі рівноваги Неша шляхом додавання блоку визначення доцільності декомпозиції отриманих клієнтських задач q_i на множину симетричних підзадач SQ_{q_i} з метою подальшого визначення такого збалансованого призначення їх виконання на S серверах (SQ_{q_i}) $\rightarrow (s_i^* \times \dots \times s_n^*)$, що для $\forall sq_j^{q_i} \in SQ_{q_i}$ та $\forall s_i \in S$ виконано обов'язкову умову $U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*)$.

На основі вищезазначеного, запропоновано модель оптимального використання ресурсів кластерної системи, яка, на відміну від існуючих, визначає необхідність декомпозиції отриманого завдання на 2 підзадачі (біматрична гра) засобами теорії нечіткої логіки з метою подальшого знаходження оптимального розподілу використання серверів кластерної системи на основі рівноваги Неша в умовах деякої нечіткості (розмитості) значень досліджуваного простору ознак отриманих задач та узагальнюючого показника рівня завантаженості серверів.

Так, задача визначення необхідності декомпозиції отриманого завдання на 2 підзадачі зводиться до аналітичного виразу:

$$X = (x_1, x_2, \dots, x_n) \rightarrow y = (a_1^{jk_j}, a_2^{jk_j}, \dots, a_n^{jk_j}) \in D = (d_1, d_2, \dots, d_n), i = \overline{1, n}, j = \overline{1, m}, \quad (1)$$

де: X – ознаковий простір опису усіх отриманих задач q_i балансувальником (мережевий протокол; розмір ресурсу, що підлягає обробці; показники завантаженості процесора /

оперативної пам'яті та мережевої карти вузла кластеру), на основі аналізу якого визначається їх тип (прості, складні);

y – лінгвістичний опис рішення (висновку) $d_j \in D$ про тип задачі для деякого фіксованого вектора значень ознакового простору $(x_1^*, x_2^* \dots, x_n^*)$ задачі q_i ;

jk_j – номери комбінацій нечітких термів a_i простору ознак X отриманих задач q_i , які відповідають значенню d_j .

У відповідності до [12, 13] функціональна залежність між досліджуваним простором ознак отриманих задач та відповідним прийнятим рішенням про необхідність їх декомпозиції формалізується у вигляді наступної системи нечітких логічних тверджень типу “ЯКЩО-ТО, ІНАКШЕ” (2):

$$\begin{aligned}
 & \text{ЯКЩО } (x_1 = a_1^{11}) \text{ I } (x_2 = a_2^{11}) \text{ I} \dots \text{I } (x_n = a_n^{11}) \text{ АБО} \\
 & \quad (x_1 = a_1^{12}) \text{ I } (x_2 = a_2^{12}) \text{ I} \dots \text{I } (x_n = a_n^{12}) \text{ АБО} \\
 & \quad (x_1 = a_1^{1k_1}) \text{ I } (x_2 = a_2^{1k_1}) \text{ I} \dots \text{I } (x_n = a_n^{1k_1}) \text{ ТО} \\
 & \quad y = d_1, \text{ ІНАКШЕ} \\
 & \text{ЯКЩО } (x_1 = a_1^{21}) \text{ I } (x_2 = a_2^{21}) \text{ I} \dots \text{I } (x_n = a_n^{21}) \text{ АБО} \\
 & \quad (x_1 = a_1^{22}) \text{ I } (x_2 = a_2^{22}) \text{ I} \dots \text{I } (x_n = a_n^{22}) \text{ АБО} \\
 & \quad (x_1 = a_1^{2k_2}) \text{ I } (x_2 = a_2^{2k_2}) \text{ I} \dots \text{I } (x_n = a_n^{2k_2}) \text{ ТО} \\
 & \quad y = d_2, \text{ ІНАКШЕ} \\
 & \text{ЯКЩО } (x_1 = a_1^{m1}) \text{ I } (x_2 = a_2^{m1}) \text{ I} \dots \text{I } (x_n = a_n^{m1}) \text{ АБО} \\
 & \quad (x_1 = a_1^{m2}) \text{ I } (x_2 = a_2^{m2}) \text{ I} \dots \text{I } (x_n = a_n^{m2}) \text{ АБО} \\
 & \quad (x_1 = a_1^{mk_m}) \text{ I } (x_2 = a_2^{mk_m}) \text{ I} \dots \text{I } (x_n = a_n^{mk_m}) \text{ ТО } y = d_m.
 \end{aligned} \tag{2}$$

З використанням операцій \cup (АБО) \cap (І) вищеописана система логічних висловлювань (2) приводиться до вигляду (3) [12,13]:

$$\bigcup_{p=1}^{k_j} \left\{ w_{jp} \left[\bigcap_{i=1}^n (x_i = a_i^{jp}) \right] \right\} \rightarrow y = d_j, j = \overline{1, m}. \tag{3}$$

Розрахунок значень багатомірних функцій належності для всіх рішень опису задач q_i , що можуть бути отримані балансувальником, представлено у вигляді наступної системи нечітких логічних рівнянь із заміною їх лінгвістичних термів відповідними функціями належності, а операції I та \cup на \wedge та \vee (4):

$$\mu^{d_j}(x_1, x_2, \dots, x_n) = \bigvee_{p=1}^{k_j} \left\{ w_{jp} \left[\bigwedge_{i=1}^n \mu^{jp}(x_i) \right] \right\}, j = \overline{1, m}, \tag{4}$$

де \wedge – нечітке логічне I , \vee – нечітке логічне АБО.

Оскільки операціям \wedge та \vee в теорії нечітких множин відповідають операції \max та \min , отримуємо [13, 14]:

$$\mu^{d_j}(x_1, x_2, \dots, x_n) = \max_{p=\overline{1, k_j}} \{ w_{jp} \min_{i=\overline{1, n}} [\mu^{jp}(x_i)] \}, j = \overline{1, m} \tag{5}$$

Таким чином, в якості прийнятого рішення про необхідність декомпозиції отриманої задачі q_i на симетричні підзадачі $sq_j^{q_i}$ на основі фіксованого вектора значень аналізованих

ознак обирається результат із максимальним значенням, отриманим внаслідок згортки функцій приналежності термів нечітких правил опису задач q_i .

Блок визначення необхідності декомпозиції отриманих задач q_i моделі оптимального використання серверних ресурсів кластерної системи військового призначення на основі рівноваги за Нешем реалізує наступні налаштування:

1. Множина X вхідних лінгвістичних змінних:

Protocol – штучне кодування найменувань мережевих протоколів кортежами значень: {"BackEnd (серверна логіка веб сервісу) – [0.5, 1, 1.5]", "Telnet – [1.5, 2, 2.5]", "DNS – [2.5, 3, 3.5]", "DHCP – [3.5, 4, 4.5]", "SMTP – [4.5, 5, 5.5]", "SNMP – [5.5, 6, 6.5]", "FTP – [6.5, 7, 7.5]", "SSH – [7.5, 8, 8.5]", "NFS – [8.5, 9, 9.5]", "NTP – [9.5, 10, 10.5]", "SNTP – [10.5, 11, 11.5]", "XMPP – [11.5, 12, 12.5]", "LDAP – [12.5, 13, 13.5]", "SIP – [13.5, 14, 14.5]", "RTP – [14.5, 15, 15.5]", "IMAP – [15.5, 16, 16.5]", "POP3 – [16.5, 17, 17.5]", "SMB – [17.5, 18, 18.5]"}, функція належності: трикутна;

Size-resource – розмір ресурсу (Кб), що підлягає обробці: {"Дуже малий – [0, 209715.2]", "Малий – [104857, 262144]", "Середній – [104857, 524300]", "Великий – [104857, 786432]", "Дуже великий – [838900, 1049000]"}, функції належності: Гаусова, S – функція, Z – функція;

LoadCPU – показник завантаженості процесора у відсотках: {"Дуже малий – [0, 0, 20]", "Малий – [10, 25, 40]", "Середній – [30, 45, 60]", "Великий – [50, 65, 80]", "Дуже великий – [70, 100, 100]"}, функція належності: трикутна;

LoadRAM – показник завантаженості оперативної пам'яті у відсотках: {"Дуже малий – [0, 0, 20]", "Малий – [10, 25, 40]", "Середній – [30, 45, 60]", "Великий – [50, 65, 80]", "Дуже великий – [70, 100, 100]"}, функція належності: трикутна;

LoadNW – показник завантаженості мережевої карти у відсотках: {"Дуже малий – [0, 0, 20]", "Малий – [10, 25, 40]", "Середній – [30, 45, 60]", "Великий – [50, 65, 80]", "Дуже великий – [70, 100, 100]"}, функція належності: трикутна;

2. Алгоритм нечіткого логічного виводу по базі знань: *Мамдані*;

3. Вихідна лінгвістична змінна u : d_1 – проста задача: {"Не потребує декомпозиції – 0, 10"}; d_2 – складна задача: {"Потребує декомпозиції – 10, 38+"};

4. Метод дефазифікації: *Центроїд (центр ваги)* – використовується для отримання значення показника ваги складності w_{q_i} отриманої задачі q_i ;

5. Ваговий коефіцієнт w для усіх нечітких правил – 1.

Таким чином, визначається необхідність декомпозиції отриманого завдання на 2 підзадачі із відповідним поділом ваги її складності w_{q_i} . У випадку відсутності необхідності декомпозиції отриманої задачі q_i гравцями біматричної гри G є отримана задача q_i та віртуальна задача з нульовою вагою w_0 .

На рисунку 2 представлено результати імітаційного моделювання засобами програмного забезпечення *Matlab* (пакет *Fuzzy Logic*) процесу визначення необхідності декомпозиції типових клієнтських задач на основі аналізу попередньо отриманого статистичного набору даних при тестуванні кластеру демонструє 100 відсотків ефективності за показником точності (з 993 задач, для яких існувала необхідність декомпозиції 993 декомпозується).

Процедура знаходження оптимального розподілу використання множини серверів $(s_i^* \times \dots \times s_n^*)$ кластерної системи на основі рівноваги Неша (NE) в умовах нечіткості (розмитості) значень досліджуваного простору ознак отриманих задач та узагальненого показника завантаженості серверів здійснюється на основі математичної моделі, запропонованої у [1] (6, 7).

■ Відсутність необхідності декомпозиції ■ Необхідність у декомпозиції

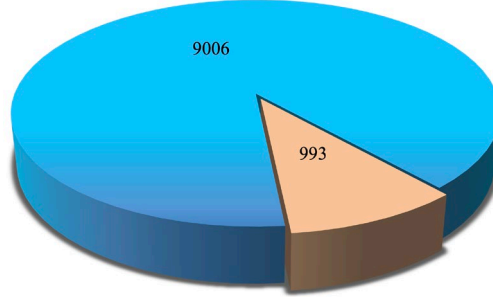


Рисунок 2 – Результати імітаційного моделювання процесу визначення необхідності декомпозиції задач

$$NE: p(sq_1^{qi} \rightarrow S) = \begin{cases} p_{sq_1^{qi}}(s_i) = p_{sq_2^{qi}}(s_i) * u_{11} + \left(\sum_{s_i \in S} p_i(s_i) - p_{sq_2^{qi}}(s_i) \right) * u_{1n} \\ p_{sq_1^{qi}}(s_n) = p_{sq_2^{qi}}(s_i) * u_{n1} + \left(\sum_{s_i \in S} p_i(s_i) - p_{sq_2^{qi}}(s_i) \right) * u_{nn} \end{cases} \quad (6)$$

$$NE: p(sq_2^{qi} \rightarrow S) = \begin{cases} p_{sq_2^{qi}}(s_i) = p_{sq_1^{qi}}(s_i) * u_{ii} + \left(\sum_{s_i \in S} p_i(s_i) - p_{sq_1^{qi}}(s_i) \right) * u_{ni} \\ p_{sq_2^{qi}}(s_n) = p_{sq_1^{qi}}(s_i) * u_{in} + \left(\sum_{s_i \in S} p_i(s_i) - p_{sq_1^{qi}}(s_i) \right) * u_{nn} \end{cases} \quad (7)$$

де $p_{sq_j^{qi}}(s_i)$ – імовірність вибору сервера s_i для виконання підзадачі sq_j^{qi} , u_{ii} – вигравш (показник завантаженості серверів l_{s_i}), який отримує підзадача sq_j^{qi} внаслідок некооперативного вибору сервера $s_i \in S$.

Узагальнений показник завантаженості серверів l_{s_i} для платіжної матриці гри G визначається на основі значень незадіяних ресурсів, які визначаються у відсотковому співвідношенні до загального обсягу ресурсів серверів:

показник завантаженості U_{CPU} центрального процесору (CPU): середній відсоток використання CPU за деякий проміжок часу (8).

$$U_{CPU} = \frac{CPU_{Usage}}{CPU_{Capacity}} * 100 \quad (8)$$

де CPU_{Usage} – кількість операцій, які виконує процесор за одну секунду; $CPU_{Capacity}$ – максимальна кількість операцій, які може виконати процесор за одну секунду;

показник завантаженості U_{RAM} оперативної пам'яті (RAM): відсоток зайнятої пам'яті в поточний момент часу (9).

$$U_{RAM} = \frac{RAM_{Usage}}{RAM_{Capacity}} * 100 \quad (9)$$

де RAM_{Usage} – обсяг оперативної пам'яті, який використовується; $RAM_{Capacity}$ – загальний обсяг оперативної пам'яті;

показник завантаженості мережевої карти (NW): відсоток використання пропускної здатності мережевої карти сервера в поточний момент часу (прийом, передача байтів) (10).

$$U_{NW} = \frac{(b_{Sent} - b_{SentPrev}) + (b_{Recv} - b_{RecvPrev})}{b_{Sent} + b_{Recv}} * 100 \quad (10)$$

де: b_{Sent} – кількість переданих байт в поточний момент часу; b_{Recv} – кількість отриманих байт в поточний момент часу; $b_{SentPrev}$ – кількість переданих байт за попередню одиницю часу; $b_{RecvPrev}$ – кількість отриманих байт за попередню одиницю часу.

На цій основі, показник завантаженості сервера l_{S_i} ґрунтується на визначені відсотку незадіяного ресурсу найбільш завантаженого компонента (U_{CPU}, U_{RAM}, U_{NW}) (11):

$$l_{S_i} = 100 - \max(U_{CPU}, U_{RAM}, U_{NW}) \quad (11)$$

Таким чином, на основі отриманого імовірнісного розподілу $p(sq_j^{q_i})$ досягається оптимальний профіль використання серверних ресурсів з метою виконання отриманих задач q_i (підзадач $sq_j^{q_i}$). Після процедури NE окремій задачі q_i (підзадачі $sq_j^{q_i}$) неважливо, який із серверів буде обрано іншою задачею (підзадачею), у будь-якому випадку виграш буде однаковий для всіх задач (підзадач).

Оцінка ефективності застосування удосконаленої моделі. Оцінку отриманого розподілу ресурсів кластеру запропонованою моделлю оптимального використання серверних ресурсів доцільно здійснювати на основі порівняння її застосування методом адаптивного балансування навантаження кластерної системи військового призначення на основі рівноваги Неша [1] до удосконалення та після, а також у порівнянні із найбільш поширеними і водночас простішими алгоритмами: *Round Robin* і *Least Connection*. Так, чотири сервери кластера поступово навантажуються значною кількістю задач. Критерієм оптимальності у даному випадку є показник рівномірного розподілу використання вузлів кластера під час обробки отриманих задач. На рисунку 3 представлено результати моделювання отриманого розподілу ресурсів кластера у процесі обробки отриманих задач від клієнтів без їх декомпозиції на основі алгоритмів *Least Connection* (рис. 3а) і *Round Robin* (рис. 3б) у вигляді варіаційних кривих навантаження вищезазначених серверів. По осі абсцис – кількість задач, які сервери встигають виконати до моменту повного завантаження кластера (вісь ординат у відсотках).

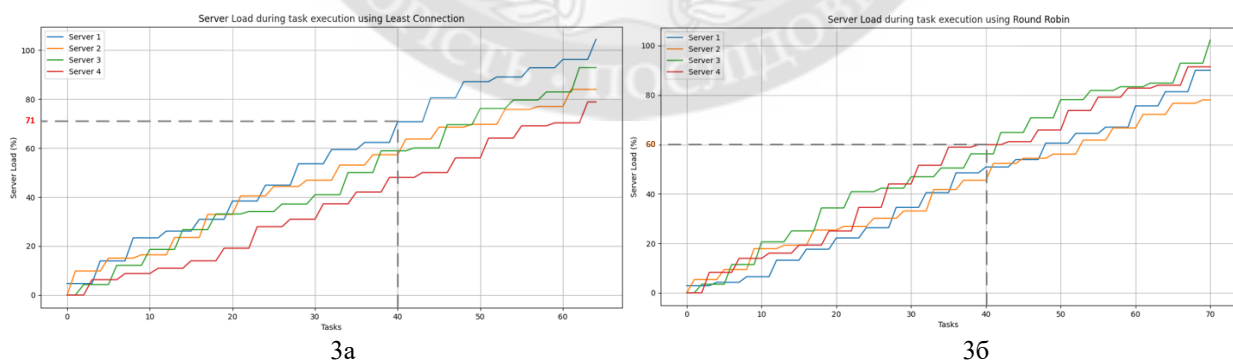


Рисунок 3 – Результати імітаційного моделювання отриманого розподілу ресурсів кластеру (*Round Robin*, *Least Connection*)

З рисунку 3 видно, що у випадку застосування алгоритму *Least Connection* (рисунок 3а) показник рівномірного розподілу ресурсів кластера вищий за аналогічний показник застосування алгоритму *Round Robin* (рисунок 3а), оскільки дисперсія показників навантаження серверів у кожен момент часу є меншою, що у свою чергу дозволяє збільшити

кількість оброблених задач від клієнтів в умовах обмеження апаратних ресурсів та прийняттого часу на обробку запитів.

На рисунку 4 представлено результати моделювання отриманого розподілу ресурсів кластера у процесі обробки отриманих задач від клієнтів до удосконалення моделі оптимального використання серверних ресурсів (рис. 4а) та після (рис. 4б) у вигляді варіаційних кривих навантаження зазначених вище серверів.

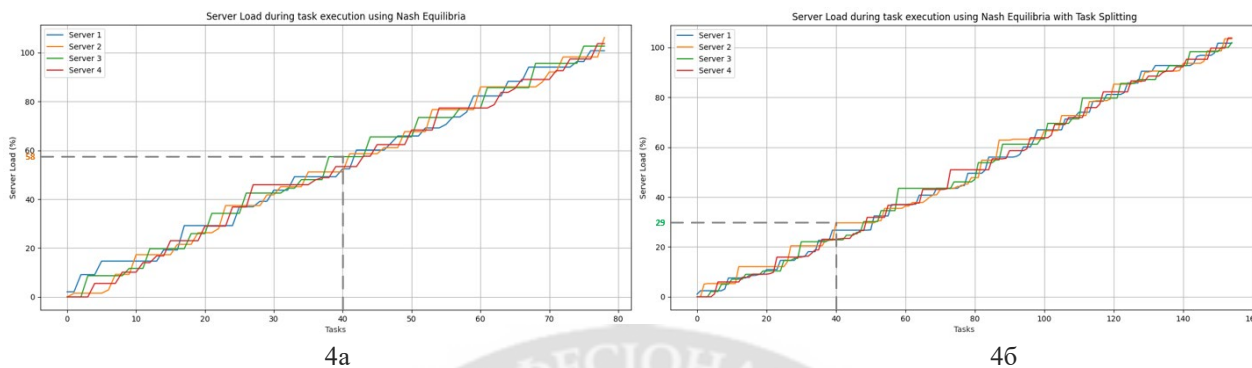


Рисунок 4 – Результати імітаційного моделювання отриманого розподілу ресурсів кластеру (*Nash Equilibria*, *Nash Equilibria with Task Splitting*)

З рисунку 4 видно, що у випадку застосування удосконаленої моделі оптимального використання серверних ресурсів (рис. 4б) показник рівномірного розподілу ресурсів кластера значно вищий на відміну від показника застосування моделі до удосконалення (рис. 4а), який є значно вищим у випадку застосування найпоширенішого алгоритму *Round Robin*, що призводить до збільшення кількості оброблених завдань у двічі.

Висновки. У статті вирішується актуальне наукове завдання оптимізації використання серверних ресурсів кластерної системи військового призначення у процесі обробки множини клієнтських запитів. Постановка даного наукового завдання зумовлена необхідністю усунення залежності показників ефективності існуючих методів балансування навантаження кластерних систем від якості навчання, а також досить вузьким підходом аналізованих наукових досліджень до вивчення впливу властивостей верифікованості, робастності та гарантії оптимуму на кінцевий результат прийняття рішення.

Проведено дослідження існуючих підходів до оптимального використання ресурсів кластерної системи. На основі проведеного аналізу можна зробити висновок, що найбільш повну відповідність висунутим критеріям порівняння існуючих адаптивних методів оптимального використання серверних ресурсів демонструють методи, в основу яких покладено концепцію теорії Ігор щодо прийняття оптимальних рішень в умовах конфлікту – рівноваги Неша, як найбільш доцільного підходу до формального опису функціональної взаємодії серверів із клієнтськими запитами, а також як підхід, що не потребує узагальнення емпіричного досвіду (навчання). Даний науково-методичний апарат дозволяє ефективно знаходити оптимальне рішення використання серверів отриманими задачами, що є стійким до відхилень будь кого з інших учасників гри (будь-яке відхилення від обраних серверів не принесе додаткової користі для клієнтської задачі).

Запропоновано модель оптимального використання ресурсів кластерної системи військового призначення, яка ґрунтується на гібридизації застосування теоретико-ігрового підходу і теорії нечіткої логіки. Суть запропонованої моделі полягає у визначенні доцільності декомпозиції отриманого завдання на підзадачі, ініційованого клієнтським запитом, засобами теорії нечіткої логіки з метою подальшого знаходження оптимального розподілу використання серверів кластерної системи на основі рівноваги Неша, із врахуванням узагальненого показника поточного рівня їх завантаженості.

Оцінка ефективності застосування удосконаленої моделі оптимального використання серверних ресурсів демонструє підвищення ефективності розподілу ресурсів удвічі за

критерієм оптимальності – рівномірному (стійкому у рівновазі) розподілі ресурсів на протязі усього процесу функціонування кластеру, що у свою чергу дозволяє значно збільшити кількість оброблених задач, а також підвищити рівень функціональної стійкості ІС військового призначення (сил оборони та безпеки держави), особливо у воєнний час.

Отриманий науковий результат розглядається, як підґрунтя для реалізації нових адаптивних систем балансування навантаження вузлів кластерної системи, застосування яких дозволяє оптимально використовувати ресурси кластерної системи військового призначення у стійкій рівновазі.

ЛІТЕРАТУРА:

1. Метод адаптивного балансування навантаження в кластерних системах військового призначення на основі рівноваги Неша / В. Фесьоха та ін. *Збірник наукових праць Центру воєнно-стратегічних досліджень НУОУ імені Івана Черняхівського*. 2023. Т. 3, №76. С. 101–110. URL: <https://doi.org/10.33099/2304-2745/2022-3-76/101-110>.
2. Begam G. S. Load Balancing in DCN Servers through SDN Machine Learning Algorithm [Electronic resource] / G. Sulthana Begam, M. Sangeetha, N. R. Shanker // *Arabian Journal for Science and Engineering*. – 2021. – Mode of access: <https://doi.org/10.1007/s13369-021-05911-1> (date of access: 02.05.2023).
3. Динамічне балансування трафіку між декількома провайдерами / Ю. Кльоц та ін. *Вісник Хмельницького національного університету*. 2019. Т. 4, №275. С. 62–67. URL: <https://www.doi.org/10.31891/2307-5732-2019-275-4-62-67>.
4. Efficient Load Balancing in Cloud Computing using Multi-Layered Mamdani Fuzzy Inference Expert System [Electronic resource] / Naila Samar Naz [et al.] // *International Journal of Advanced Computer Science and Applications*. – 2019. – Vol. 10, no. 3. – Mode of access: <http://dx.doi.org/10.14569/IJACSA.2019.0100373>.
5. Bayesian Deep Learning-Based Probabilistic Load Forecasting in Smart Grids / Y. Yang et al. *IEEE Transactions on Industrial Informatics*. 2020. Vol. 16, no. 7. P. 4703–4713. URL: <https://doi.org/10.1109/tii.2019.2942353>.
6. Breiman L., Friedman J., Olsen R. and Stone C. *Classification and Regression Trees*. Monterey, CA: Wadsworth International Group, 1984.
7. Campesato O. *Python 3 for Machine Learning : Book*. Mercury Learning and Information, 2020. 364 p.
8. Multi-objective Task Scheduling Optimization in Cloud Computing based on Genetic Algorithm and Differential Evolution Algorithm / Y. Li et al. *2018 37th Chinese Control Conference (CCC)*, Wuhan, 25–27 July 2018. 2018. URL: <https://doi.org/10.23919/chicc.2018.8483505>.
9. Chen L., Wu K., Li Y. A Load Balancing Algorithm Based on Maximum Entropy Methods in Homogeneous Clusters. *Entropy*. 2014. Vol. 16, no. 11. P. 5677–5697. URL: <https://doi.org/10.3390/e16115677>.
10. Load Balancing Algorithms, Types and Techniques - Kemp [Electronic resource] // *Load Balancer For Always-On Application Experience - Kemp*. – Mode of access: <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques>.
11. MapReduce Tutorial. *Apache Hadoop*. URL: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.
12. Ротштейн О. П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі : Монографія. Вінниця : «УНІВЕРСУМ-Вінниця», 1999. 320 с.
13. Мітюшкін Ю. І., Мокін Б. І., Ротштейн О. П. *Soft Computing: ідентифікація закономірностей нечіткими базами знань: Монографія*. Вінниця : ВДТУ «УНІВЕРСУМ-Вінниця», 2002. 145 с.

REFERENCES:

1. Fesokha, V.V., Neroznak, Ye.I., Sova, O.Ya. and Nesterov, O.M. (2023), “Metod adaptivnoho balansuvannia navantazhennia v klasternykh systemakh viiskovoho pryznachennia na osnovi rivnovahy Nesha” [The method of adaptive load balancing in cluster systems of military purposes based on Nash Equilibrium], Collection of the scientific papers of the Centre for Military and Strategic Studies of the National Defence University of Ukraine named after Ivan Cherniakhovskiy, No. 3(76), pp. 101-110. <https://doi.org/10.33099/2304-2745/2022-3-76/101-110>
2. Begam, G.S., Sangeetha, M. and Shanker, N.R. (2022), “Load Balancing in DCN Servers through SDN Machine Learning Algorithm”, Arabian Journal for Science and Engineering, Vol. 47, pp. 1423-1434.
3. Klots, Y.P., Stefanovytch, K.Y., Shakhoval, Y.S. and Demeshko, V.I. (2019), “Dynamichne balansuvannia trafiku mizh dekilкома provaideramy” [Dynamic traffic balance between several providers], Herald of Khmelnytskyi national university, No. 4(275), pp. 62-67. <https://www.doi.org/10.31891/2307-5732-2019-275-4-62-67>
4. Naz N.S., Abbas S., Khan M.A., Abid B., Tariq N. and Khan M.F. (2019), “Efficient Load Balancing in Cloud Computing using Multi-Layered Mamdani Fuzzy Inference Expert System”, International Journal of Advanced Computer Science and Applications, Vol. 10 No. 3, pp. 569-577.
5. Yang, Y., Li, W., Gulliver, T.A. and Li, S. (2020), “Bayesian Deep learning-Based Probabilistic Load Forecasting in Smart Grids”, IEEE Transactions on Industrial Informatics, Vol. 16 No. 7, pp. 4703-4713.
6. Breiman L., Friedman J., Olsen R. and Stone C. (1984), Classification and Regression Trees, illustrated, reprint, Wadsworth International Group, CA.
7. Camesato O. (2020), Python 3 for Machine Learning, Mercury Learning and Information, Dulles, VA.
8. Li Yu., Wang S., Hong X. and Li Yo. (2018), “Multi-objective Task Scheduling Optimization in Cloud Computing based on Genetic Algorithm and Differential Evolution Algorithm”, 37th Chinese Control Conference (CCC), Wuhan, China, July 25-27, 2018, pp. 4489-4494.
9. Chen L., Wu K. and Li Y. (2014), “A Load Balancing Algorithm Based on Maximum Entropy Methods in Homogeneous Clusters”, Entropy, Vol. 16 No. 11, pp. 5677-5697.
10. Load Balancer For Always-On Application Experience - Kemp (2023), “Load Balancing Algorithms, Types and Techniques - Kemp”, available at : <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques> (accessed 7 June 2023).
11. Apache Hadoop (2022), “MapReduce Tutorial”, available at : https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (accessed 7 June 2023).
12. Rotshtein, O.P. (1999), “Intelektualni tekhnolohii identyfikatsii: nechitki mnozhyny, henetychni alhorytmy, neironni merezhi : monohrafia” [Intelligent identification technologies: fuzzy sets, genetic algorithms, neural networks], UNIVERSUM-Vinnytsia, Vinnytsia, 320 p.
13. Mitiushkin, Yu.I., Mokin, B.I. and Rotshtein, O.P. (2002), “Soft Computing: identyfikatsiia zakonomirnostei nechitkymy bazamy znan: monohrafiia” [Soft Computing: identification of regularities by fuzzy knowledge bases], UNIVERSUM-Vinnytsia, Vinnytsia, 145 p.

Ph.D. Fesokha V.V., Neroznak Ye.I., D.Sci.Tech., Sova O.Ya.

AN IMPROVED MODEL OF OPTIMAL USE OF RESOURCES OF A CLUSTER SYSTEM OF MILITARY ASSIGNMENT BASED ON NASH EQUILIBRIUM

The article solves the actual scientific task of optimizing the use of server resources of a military cluster system in the process of processing a large number of client requests. The setting of this scientific task is due to the need to eliminate the dependence of the efficiency indicators of the existing methods of balancing the load of cluster systems on the quality of training, as well as a rather narrow approach of the analyzed scientific studies to the study of the influence of the

properties of verifiability, robustness and the guarantee of optimality on the final decision-making result. A study of existing approaches to the optimal use of cluster system resources was conducted. Based on the results of the analysis, methods based on the game-theoretic scientific-methodical apparatus were selected as the most appropriate approach to the formal description of the functional interaction of servers with client requests, as well as an approach that does not require the generalization of empirical experience (learning).

A model of the optimal use of resources of the military cluster system is proposed, which is based on the hybridization of the application of the game-theoretic approach and the theory of fuzzy logic. The essence of the proposed model is to determine the necessity of decomposition of the received task into subtasks, initiated by the client request, by means of the theory of fuzzy logic with the aim of further finding the optimal distribution of the use of cluster system servers based on the Nash equilibrium, taking into account the generalizing indicator of the current level of their workload. The obtained results are considered as a basis for the implementation of new adaptive systems for balancing the load of nodes of the cluster system, the application of which allows optimal use of the resources of the cluster system of military purpose in a stable balance.

Keywords: finding the optimum, cluster, game theory, Nash equilibrium, fuzzy logic.

