https://doi.org/10.31891/2219-9365-2026-86-7
УДК 004.75

ВОЄВОДІН Євгеній
Черкаський національний університет імені Богдана Хмельницького
https://orcid.org/0000-0002-6415-8566
yevhenii.voievodin@vu.cdu.edu.ua
РОЗЛОМІЙ Інна
Черкаський державний технологічний університет
https://orcid.org/0000-0001-5065-9004
inna-roz@ukr.net
СТАБЕЦЬКА Тетяна
Черкаський національний університет імені Богдана Хмельницького
https://orcid.org/0000-0001-9192-5313
tatiana_ami@ukr.net

МЕТОД РОЗПОДІЛЕННЯ РЕСУРСІВ З УРАХУВАННЯМ РІВНЯ КРИТИЧНОСТІ В СИСТЕМАХ ОРКЕСТРУВАННЯ ВІРТУАЛЬНИХ КОНТЕЙНЕРІВ

У роботі запропоновано метод розподілення ресурсів у системах оркестрування віртуальних контейнерів, який враховує рівень критичності сервісу як додатковий параметр під час прийняття рішення щодо розміщення контейнера на вузлі кластера. На відміну від традиційних стратегій, запропонований підхід дозволяє враховувати не лише апаратні характеристики контейнера, а й рівень критичності сервісу, що дає змогу мінімізувати ризики, пов'язані з небажаним спільним розміщенням контейнерів різних рівнів.

Для кількісного оцінювання якості розміщення запропоновано коефіцієнт колокації рівнів критичності, який враховує кількість різних рівнів на вузлі та різницю між їхніми мінімальним і максимальним значеннями. Це дозволяє формалізувати та оцінювати безпекові ризики.

Результати експериментального дослідження підтвердили ефективність використання самоорганізаційних карт Кохонена як основи для стратегії розподілення. Встановлено, що включення параметра критичності до вхідного вектора нейронної мережі під час її навчання та подальшої роботи дозволяє суттєво зменшити коефіцієнт небажаної колокації порівняно з традиційними підходами. Експериментально показано, як зміна топології самоорганізаційної карти впливає на результат розподілення.

Отримані дані свідчать, що запропонований метод є перспективним для систем із високими вимогами до безпеки та стабільності, оскільки він знижує ризики взаємного впливу сервісів без суттєвого погіршення інших показників балансування, зокрема рівня дефрагментації ресурсів, кількості створених контейнерів та коефіцієнта вдалих розподілень. Перспективи подальших досліджень полягають у розширенні моделі за рахунок урахування параметрів мережевої взаємодії та дослідженні роботи методу в умовах динамічної зміни навантаження, міграції контейнерів та відмов вузлів кластера.

Ключові слова: системи оркестрування віртуальних контейнерів, розподілення ресурсів, самоорганізаційна карта Кохонена, коефіцієнт колокації, інформаційна безпека, штучна нейронна мережа, мікросервісна архітектура.

VOIEVODIN Yevhenii, STABETSKA Tetiana
Bohdan Khmelnytsky National University of Cherkasy
ROZLOMII Inna
Cherkasy State Technological University

CRITICALITY-AWARE RESOURCE DISTRIBUTION METHOD FOR CONTAINER ORCHESTRATION SYSTEMS

This paper proposes an advanced resource distribution method for container orchestration systems that incorporates service criticality as a key parameter for placement decisions. Unlike traditional scheduling strategies such as "binpack" or "spread", the proposed approach integrates the business importance of services to mitigate security risks associated with the unwanted colocation of containers with different criticality levels.

The core of the methodology relies on Kohonen self-organizing maps, which is a type of unsupervised artificial neural network. By including a service criticality level into the input vector alongside hardware requirements, the system achieves intelligent clustering of containers on cluster nodes. To evaluate the effectiveness of this placement, the study introduces a "criticality colocation coefficient". This metric quantifies security risks based on the number of distinct criticality levels present on a single node and the variance between their minimum and maximum values.

The experimental framework included a comparative analysis of the strategy based on self-organizing map both with and without the inclusion of the criticality parameter. The results reveal that while a resource-only self-organizing map configuration offers small improvements over traditional methods, the integration of criticality-awareness leads to a substantial decrease in the colocation coefficient, effectively isolating sensitive services. Furthermore, the study identifies that the topology of the self-organizing map plays a crucial role in the outcome: while a linear topology shows limited effectiveness as the cluster scales, a square matrix topology provides superior results in maintaining clear segregation of critical services.

The findings indicate that the proposed method is highly promising for environments with strict security and stability requirements. Importantly, the addition of the criticality parameter does not lead to significant degradation of other metrics, such as resource defragmentation levels or the success rate of containers distribution. Future research will explore the integration of network interaction patterns and the performance of the model under dynamic workloads, container migrations, and node failures.

Keywords: container orchestration systems, resource distribution method, Kohonen self-organizing map, colocation coefficient, information security, artificial neural network, microservices architecture.

ISSN 2219-9365

Стаття надійшла до редакції / Received 27.03.2026
Прийнята до друку / Accepted 22.04.2026
Опубліковано / Published 31.05.2026

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Системи оркестрування віртуальних контейнерів (СОВК), зокрема Kubernetes, Docker Swarm, Apache Mesos, відіграють важливу роль у сучасних хмарних і розподілених обчислювальних середовищах [1]. Однією з ключових функцій таких систем є розміщення контейнерів на вузлах кластера. У загальному вигляді ця задача полягає у виборі вузла, який має достатній обсяг ресурсів для розміщення та забезпечення роботи створеного контейнера, а також відповідає заданому критерію оптимальності. Наприклад, традиційна стратегія розподілення заповненням орієнтована на більш інтенсивне використання уже задіяних вузлів, залишаючи більше вільних ресурсів для нових запитів балансування. Натомість стратегія розподілення поширенням намагається рівномірно розподілити контейнери між вузлами, що сприяє підвищенню відмовостійкості системи [2].

Складність задачі полягає в динамічному характері надходження запитів на створення контейнерів. Оскільки планувальник не володіє повною інформацією про майбутнє навантаження, раніше прийняті рішення можуть призвести до субоптимального стану кластера в перспективі. Хоча традиційних стратегій зазвичай достатньо для базових сценаріїв, вони не враховують додаткові вимоги до продуктивності, мережевої взаємодії чи то безпеки.

У сучасних системах які використовують мікросервісну архітекруту [3], додаткову складність створює неоднорідність самих сервісів, оскільки різні сервіси можуть мати різний рівень критичності для бізнес-процесів, різні вимоги до доступності та різні пороги реагування на деградацію продуктивності [4]. Наприклад, для сервісів із високим рівнем критичності застосовуються жорсткіші вимоги до моніторингу й часу відновлення, тоді як менш важливі сервіси можуть мати нижчий пріоритет. Унаслідок цього розміщення контейнерів з різним рівнем критичності на одному вузлі створює додаткові ризики [5]. Зокрема, у разі компрометації або ж некоректної роботи менш критичного сервісу можуть виникнути небажані побічні ефекти для більше важливого сервісу, розміщеного на тому ж вузлі (проблема "noisy neigbour" [6]). На рис. 1 наведено приклад такої ситуації: через перевантаження контейнера з низьким рівнем критичності (контейнер 2) зазнає негативного впливу контейнер з вищим рівнем критичності (контейнер 1), що створює ризики для стабільного виконання критичних бізнес-функцій системи. Наприклад, якщо у фінансовому застосунку перший контейнер відповідає за перевірку лімітів клієнта, а третій контейнер від нього залежить, то деградація першого може дати змогу зловмиснику використати застосунок без перевірки відповідних лімітів.

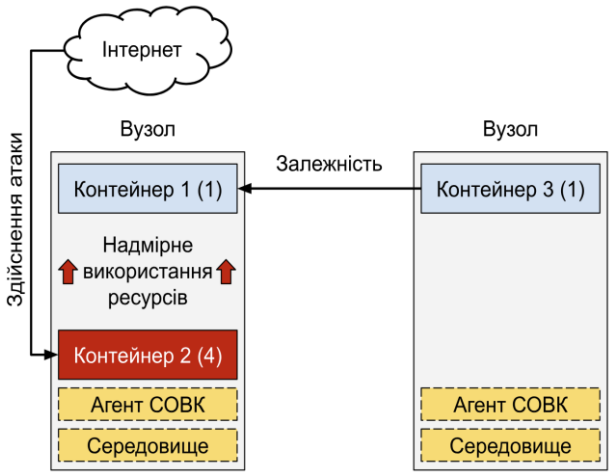

Рис. 1. Приклад проблеми спричиненої розміщенням контейнера нижчого рівня критичності (4) на одному вузлі з контейнером вищого рівня критичності (1)

З огляду на це, актуальним є розробка стратегії розподілення ресурсів, яка враховує не лише апаратні вимоги до вузла, такі як обсяг оперативної пам'яті, кількість ядер процесора чи дисковий простір, а й рівень критичності сервісу.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Проблема розподілення ресурсів у СОВК досліджується з різних напрямів, оскільки оптимальність стратегії залежить від обраних цільових критеріїв. У наявних дослідженнях основна увага здебільшого приділяється ефективності використання ресурсів, зменшенню часу виконання завдань, оптимізації мережевої взаємодії або прогнозуванню навантаження. Водночас безпекові аспекти розміщення контейнерів,

зокрема ризики розміщення контейнерів із різним рівнем критичності на одному вузлі, залишаються менш дослідженими та потребують окремого аналізу.

Значна увага приділяється методам багатокритеріального прийняття рішень. Зокрема, застосування методу TOPSIS дозволяє ефективніше використовувати ресурси кластера, беручи до уваги декілька критеріїв одночасно [7], тоді як стратегія розподілення з використанням методу VIKOR орієнтована на пошук компромісного рішення за наявності суперечливих критеріїв [8]. Стратегії на базі моделі змішаного цілочислового лінійного програмування [9] дають змогу враховувати не тільки обчислювальні ресурси, але й мережеві вимоги розміщення контейнерів, завдяки чому зменшують витрати на взаємодію між ними. Такий підхід має високу точність, однак його практичне застосування може бути обмежене обчислювальною складністю, особливо за умови динамічного надходження запитів.

Поширеними також є дослідження оптимізації процесу розподілення з використанням стохастичних та еволюційних методів. Такі підходи дають змогу шукати прийнятні рішення у великому просторі можливих розміщень, не перебираючи всі варіанти напряму. Зокрема, використання алгоритму мурашиної колонії дає змогу щільніше розміщувати контейнери, що збільшує ефективність розподілення до 15% [10]. Тоді як використання стратегії на основі алгоритму рою частинок в деяких конфігураціях дає до 20% кращі результати порівняно із стратегією розподілення поширенням [11], а також сприяє більш рівномірному балансуванню між вузлами кластера [12].

Окрему групу становлять методи машинного навчання, які використовуються для аналізу стану системи, прогнозування навантаження та адаптивного прийняття рішень щодо розподілу ресурсів. До таких підходів належать регресійні моделі, методи навчання з підкріпленням, а також нейронні мережі. Зокрема, підхід на основі випадкового лісу [13] дає змогу передбачати зміни в навантаженні, що допомагає уникати переповнення вузлів. Для великих кластерів графічних процесорів ефективним є навчання з підкріпленням за участю багатьох агентів [14], де стратегія розподілення самостійно знаходить шляхи мінімізації часу виконання завдань через постійну взаємодію компонентів. Експериментальні дані підтверджують, що впровадження таких моделей дозволяє досягти до 20% вищої продуктивності порівняно з традиційними методами. Окремий інтерес у контексті цієї роботи становлять самоорганізаційні карти Кохонена, які належать до штучних нейронних мереж з навчанням без учителя. У роботі [15] запропоновано підхід до розподілення ресурсів у СОВК, де конфігурація контейнера подається як вектор ознак, а вузли кластера ранжуються за близькістю до відповідних нейронів карти з урахуванням доступних ресурсів. Такий підхід є придатним для подальшого розширення, оскільки до вектора ознак можуть бути додані не лише ресурсні характеристики, а й рівень критичності контейнера.

Варто також зазначити підходи до розподілення контейнерів, у яких основною метою є підвищення безпеки середовища виконання. Так, у роботі [16] запропоновано стратегію, що враховує профілі системних викликів контейнерів під час їх розміщення. Такий підхід спрямований на зменшення кількості вузлів з різних профілів ризику. Водночас такий критерій не враховує рівень критичності контейнера: сервіси з подібними профілями системних викликів можуть мати різну важливість для роботи системи.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Ця робота зосереджена на дослідженні методу розподілення віртуальних контейнерів у системі оркестрування, що базується на використанні самоорганізаційної карти Кохонена. Для цього передбачається формалізація критичності контейнерів, інтеграція цього параметру до вхідного простору моделі та оцінювання впливу такого підходу на безпекові характеристики розподілення. Ціллю є визначити, чи дає запропонований метод змогу зменшити небажану колокацію контейнерів з різними рівнями критичності на одному вузлі та водночас зберегти прийнятний рівень балансування ресурсів.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

У межах цієї роботи пропонується розширити задачу розподілення контейнерів у СОВК шляхом додавання до опису контейнера параметра його критичності. Під рівнем критичності контейнера розуміється умовна оцінка важливості сервісу, який виконується всередині контейнера, для стабільної роботи системи або бізнес-процесу. Контейнери з вищою критичністю потребують особливої уваги при розміщенні, оскільки їхня деградація або недоступність може мати значний вплив на роботу всієї системи.

Рівень критичності буде представлений у вигляді цілого числа. Наприклад, значення 1 відповідає найвищому рівню критичності, тоді як більші значення вказують на менш критичні сервіси. Такий спосіб представлення дозволяє використовувати рівень критичності як один з параметрів вхідного вектора як для навчання карти, так і безпосередньо для процесу розподілу.

Основна ідея полягає в тому, що спільне розміщення контейнерів із різними рівнями критичності на одному вузлі є небажаним. Як було попередньо зазначено, така колокація може створювати додаткові ризики для стабільності та безпеки найбільш критичних сервісів. Для кількісного оцінювання рівня небажаного сусідства контейнерів на окремому вузлі n введемо коефіцієнт колокації рівнів критичності, який визначається як (1).

$$\boldsymbol{K_{node}(n) = \frac{(|C_n|-1)(max(C_n)-min(C_n))}{(|C|-1)^2}} \quad (1)$$

де C – множина усіх рівнів критичності, $|C|$ – потужність множини, або ж кількість усіх можливих рівнів критичності, C_n – множина рівнів безпеки розміщених на вузлі n; $|C_n|$ – потужність множини, або ж кількість різних рівнів безпеки на вузлі; $\max(C_n)$ – найвище значення рівня безпеки на вузлі n; $\min(C_n)$ – найнижче значення рівня безпеки на вузлі n. Тоді коефіцієнт колокації рівнів критичності для кластера визначимо за формулою (2).

$$K_{cluster} = \frac{1}{|N_c|}\sum_{n \in N_c} K_{node}(n) \quad (2)$$

де N_c – множина вузлів кластера; $|N_c|$ – потужність множини, або ж кількість вузлів в кластері c. Чим менше значення коефіцієнта – тим вищий рівень безпеки. Відповідно, рівень безпеки зменшується в двох випадках: кількість різних рівнів безпеки на вузлі збільшується, або ж збільшується різниця між максимальним та мінімальним рівнями безпеки на вузлі. Скажімо, якщо всього є 4 рівні критичності, а на вузлі розміщено контейнери рівнів 1, 2 та 3, то значення коефіцієнта становить 0.44. Якщо на вузлі розміщено контейнери рівнів 1, 2 та 4, значення коефіцієнта зростає до 0.67. Якщо ж на вузлі присутні контейнери всіх рівнів, то значення становить 1, що відповідає максимально небажаному випадку колокації рівнів.

Запропонована стратегія розподілу базується на використанні самоорганізаційної карти Кохонена. Така карта дозволяє відображати багатовимірні вхідні дані у простір меншої розмірності, зберігаючи при цьому топологічну подібність між об'єктами. У контексті задачі розподілу контейнерів це означає, що контейнери з подібними характеристиками будуть зіставлені з близькими елементами карти. Тобто, схожі за вимогами контейнери групуватимуться поруч у межах структури карти.

Процес навчання карти відбувається за таким алгоритмом. Спочатку формується вектор вхідних даних на основі характеристик контейнерів, а для кожного нейрона карти ініціалізуються вагові коефіцієнти. На кожній ітерації навчання на вхід подається один вектор параметрів. Для нього визначається нейрон-переможець – той, чий вектор вагових коефіцієнтів є найближчим до вхідного вектора за обраною метрикою відстані. Після цього оновлюються вагові коефіцієнти нейрона-переможця та нейронів із його топологічної околиці. Величина коригування вагових коефіцієнтів залежить від швидкості навчання, відстані до нейрона-переможця та номера поточної ітерації. Зі збільшенням кількості ітерацій швидкість навчання та радіус околиці поступово зменшуються. Це дозволяє на початкових етапах сформувати загальну топологію карти, а на пізніших – здійснювати точне налаштування вагових коефіцієнтів окремих нейронів. Зміна вагових коефіцієнтів карти здійснюється за наступною формулою (3).

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \alpha(t)h_{bj}(t)\left[\vec{x}(t) - \vec{w}_j(t)\right] \quad (3)$$

де $\vec{w}_j$ – вектор вагових коефіцієнтів нейрона j в карті (функція виконується для кожного нейрона), t – конкретна ітерація, де $\alpha(t)$ – функція контролю швидкості навчання (як сильно змінюються вагові коефіцієнти в залежності від ітерації), $h_{bj}(t)$ – функція топологічної околиці (як сильно змінюються вагові коефіцієнти нейрона за індексом j, якщо нейрон переможець для цієї ітерації має індекс b), $\vec{x}(t)$ – вектор вхідних даних для ітерації t. У якості функції швидкості навчання буде використана функція експоненційного розпаду (4), тоді коли функція Гауса буде використана для контролю швидкості навчання (5).

$$\alpha(t) = a_0 e^{-\frac{t}{T}} \quad (4)$$

$$h_{bj}(t) = \mathrm{e}^{-\frac{d^2}{2(\sigma e^{-\frac{t}{T}})^2}} \quad (5)$$

де з нових параметрів: a_0 – початкове значення швидкості навчання, σ – початкове значення ширини топологічної околиці, d – евклідова відстань між нейронами b та j, T – загальна кількість ітерацій навчання. В межах даного дослідження використовуються значення: $T = 1000$, $\sigma = 3$, $a_0 = 0.5$.

Загальний процес роботи стратегії виглядає наступним чином:

1. Формування вектора вхідних даних на основі параметрів контейнера (у межах цього дослідження – на основі обсягу оперативної пам'яті та рівня критичності).
2. Обчислення відстані до кожного нейрона карти, що зв'язаний із конкретним вузлом кластера.
3. Формування списку вузлів-кандидатів на основі попередньо розрахованих значень відстані: чим менша відстань, тим вища ймовірність вибору вузла.
4. Вибір першого такого вузла зі сформованого списку, який має достатній обсяг вільних ресурсів для розміщення контейнера.
5. Розміщення контейнера на обраному вузлі та оновлення даних про доступні ресурси.

Для оцінки ефективності запропонованої стратегії розподілення використано метод порівняння за рівних умов [17], основні етапи якого в спрощеному вигляді зображені на рис. 2.

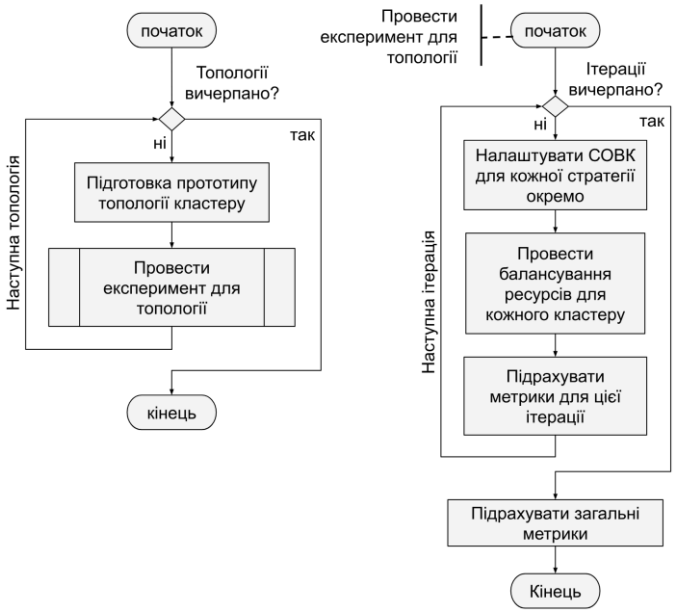

Рис. 2. Блок-схема основних етапів проведення експерименту

Для кожного виконання експерименту формується однакова послідовність контейнерів. Кожен контейнер має набір ресурсних вимог і рівень критичності. Ця послідовність подається на вхід різним варіантам стратегії розподілення. Щоб результати не впливали один на одного, кожен варіант стратегії працює з окремою копією кластера. Експеримент проводиться за рівних умов для описаної стратегії та традиційних стратегій. Щоб отримати більш точний результат експеримент проводиться в 1000 ітерацій для кожної топології кластера. Топології формуються з вузлів які мають обсяг оперативної пам'яті 32 гігабайти кожен, без інших додаткових обмежень. Кількість вузлів в кластері варіюється від 10 до 100 з кроком 10. Процес розподілу триває до моменту вичерпання вільних ресурсів, необхідних для розміщення контейнера з мінімально можливою конфігурацією. Набір контейнерів формується випадковим чином на основі декартового добутку таких можливих конфігурацій: оперативна пам'ять в гігабайтах (1, 2, 6, 8) та рівень критичності (1, 2, 3, 4). Самоорганізаційна карта навчається на такому ж наборі параметрів відповідно до попередньо описаного процесу.

Для початку проведемо експеримент без урахування рівня критичності в карті, маємо результат на рис. 3. Стратегія на основі карти Кохонена показує кращий результат, ніж традиційні підходи, проте коефіцієнт колокації залишається високим, а різниця – несуттєвою.

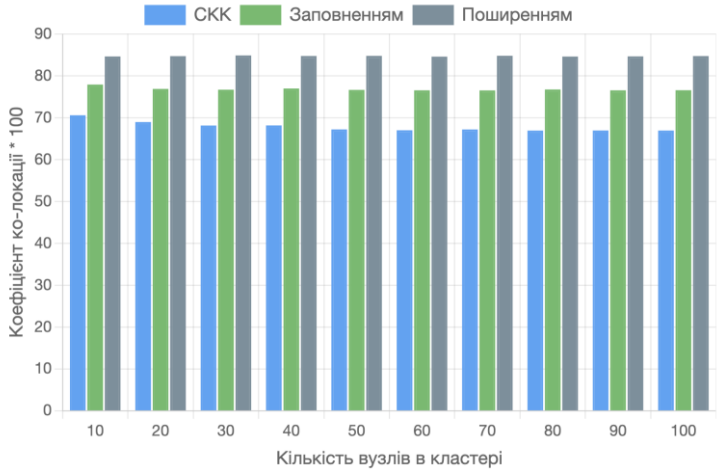

Рис. 3. Коефіцієнти колокації отримані в результаті проведення експерименту без урахування параметру рівня критичності

Тепер змінимо налаштування таким чином, щоб рівень критичності став одним із параметрів стратегії на основі карти. Використаємо лінійну топологію нейронів карти (одновимірний масив, що відповідає кількості вузлів кластера). Результат проведеного експерименту наведено на рис. 4.

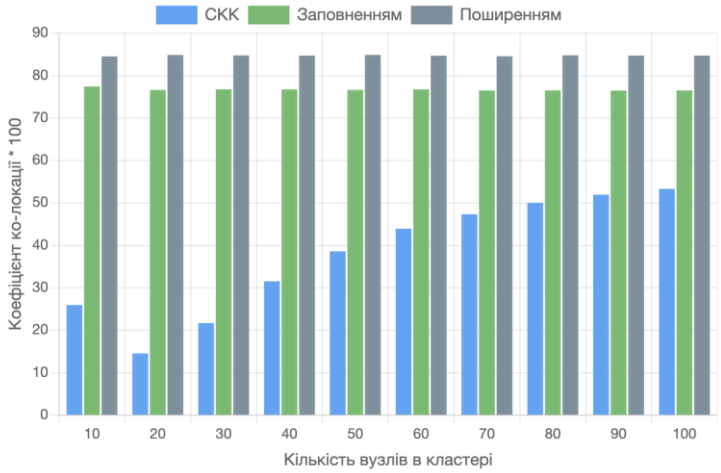

Рис. 4. Коефіцієнти колокації отримані в результаті проведення експерименту для лінійної топології карти Кохонена

Як бачимо, стратегія з використанням карти демонструє значно нижче (краще) значення коефіцієнта колокації, відповідно знижуючи безпекові ризики. Проте результати погіршуються зі збільшенням кількості вузлів у кластері. Це можна пояснити тим, що лінійна топологія карти не забезпечує достатньо простору для формування окремих сегментів, необхідних для підтримки чотирьох рівнів критичності. Після зміни топології карти на складнішу, а саме на топологію квадратної матриці [15], було отримано результат наведений на рис. 5.

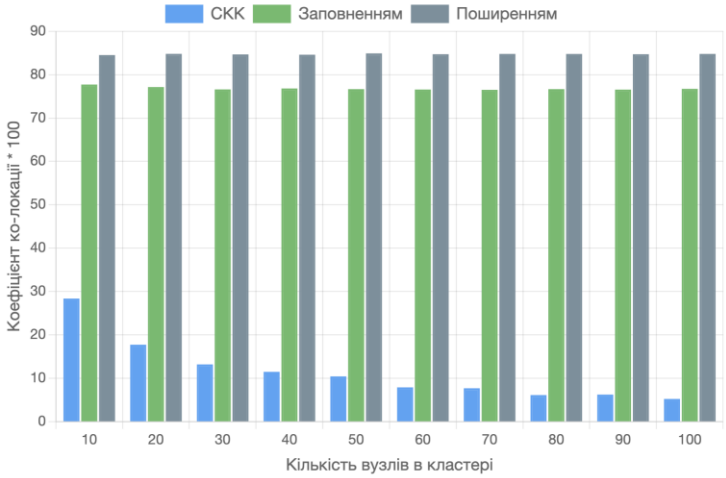

Рис. 5. Коефіцієнти колокації отримані в результаті проведення експерименту з використанням топології квадратної матриці

Результат експерименту показує, що використання стратегії на основі карти з топологією квадратної матриці дає змогу отримати низьке значення коефіцієнта колокації контейнерів. Це робить такий підхід перспективним для систем, де безпека з урахуванням спільного розміщення контейнерів є критичним фактором. Зміна умови зупинки експерименту на зупинку при першому відхилені дає майже ідентичний результат, за винятком незначних коливань між значеннями отриманих для традиційних стратегій. Збільшення ваги параметра рівня критичності контейнера вдвічі не призводить до суттєвих змін в отриманих результатах. Інші метрики, хоч і не описуються детально в даній статті, залишаються на очікуваному рівні, тобто додавання нового параметру до карти не спричиняє погіршення таких показників як рівень дефрагментації ресурсів, кількість створених контейнерів чи то коефіцієнт вдалих розподілень.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У межах проведеного дослідження розроблено метод розподілення ресурсів у СОВК, який базується на використанні самоорганізаційних карт Кохонена та враховує рівень критичності сервісів. За результатами роботи можна зробити наступні ключові висновки.

Запропоновано коефіцієнт колокації рівнів критичності, який дозволяє кількісно оцінити ризики спільного розміщення контейнерів із різними рівнями важливості на одному вузлі та в кластері в цілому. Цей показник враховує як різноманітність рівнів критичності, так і розрив між ними, даючи змогу формалізувати вимоги до оцінки безпеки.

Експериментально доведено, що включення параметра критичності до вхідного вектора самоорганізаційної карти дозволяє суттєво знизити значення коефіцієнта колокації порівняно з традиційними стратегіями. Порівняльний аналіз показав, що саме додавання цього параметра, а не просто використання нейромережевої структури, забезпечує цільову ізоляцію критичних сервісів. Також встановлено, що якість розподілення суттєво залежить від структури самоорганізаційної карти. Топологія у вигляді квадратної матриці демонструє кращу здатність до масштабування та сегрегації сервісів у великих кластерах порівняно з лінійною топологією, яка втрачає ефективність зі збільшенням кількості вузлів. Додавання параметра критичності не призводить до деградації ключових експлуатаційних показників, таких як рівень дефрагментації ресурсів або успішність розміщення контейнерів, що підтверджує практичну придатність методу для реальних систем.

Перспективи подальших досліджень полягають у розширенні розробленої моделі через інтеграцію параметрів мережевої взаємодії чи інших параметрів безпеки, зокрема наявності доступу до контейнера ззовні. Окремим важливим напрямом є дослідження стійкості методу в умовах динамічної зміни навантаження, автоматичної міграції контейнерів та відмов вузлів, а також проведення порівняльного аналізу із методами багатокритеріального прийняття рішень та підходами на основі навчання з підкріпленням.

References

1. Rodriguez M. A., Buyya R. Container-based cluster orchestration systems: A taxonomy and future directions. *Software: Practice and Experience*. 2018. Т. 49, № 5. С. 698–719. URL: https://doi.org/10.1002/spe.2660
2. Fard H., Prodan R., Wolf F. Dynamic multi-objective scheduling of microservices in the cloud. *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing*. 2020. С. 386–393.
3. On Microservice Analysis and Architecture Evolution: A Systematic Mapping Study / V. Bushong та ін. *Applied Sciences*. 2021. Т. 11, № 17. С. 7856. URL: https://doi.org/10.3390/app11177856
4. Ramsingh A., Singer J., Trinder P. Classifying the Reliability of the Microservices Architecture. *Proceedings of the 18th International Conference on Web Information Systems and Technologies WEBIST*. 2022. Т. 1. С. 21–32.
5. Voievodin Y., Rozlomii I. Application security optimization in container orchestration systems through strategic scheduler decisions. *Proceedings of the CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, CEUR Workshop Proceedings*. 2024. Т. 3654. С. 471–478.
6. DC-Store: eliminating noisy neighbor containers using deterministic I/O performance and resource isolation / M. Kwon та ін. *18th USENIX Conference on File and Storage Technologies*. 2020. С. 183–191.
7. Chakravarthi K. K., Shyamala L., Vaidehi V. TOPSIS inspired cost-efficient concurrent workflow scheduling algorithm in cloud. *Journal of King Saud University - Computer and Information Sciences*. 2020. URL: https://doi.org/10.1016/j.jksuci.2020.02.006
8. Rafieyan E., Khorsand R., Ramezanpour M. An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing. *Computers & Industrial Engineering*. 2020. Т. 140. С. 106272. URL: https://doi.org/10.1016/j.cie.2020.106272
9. Network-Aware Container Scheduling in Multi-Tenant Data Center / L. Rodrigues та ін. *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019. С. 1–6.
10. Kaewkasi C., Ratchasima N. Improvement of container scheduling for Docker using Ant Colony Optimization. *2017 9th International Conference on Knowledge and Smart Technology (KST)*. 2017. С. 254–259.
11. Li L., Chen J., Yan W. A particle swarm optimization-based container scheduling algorithm of docker platform. *Proceedings of the 4th International Conference on Communication and Information Processing*. 2018. С. 12–17.
12. A Survey on QoS Requirements Based on Particle Swarm Optimization Scheduling Techniques for Workflow Scheduling in Cloud Computing / M. Farid та ін. *Symmetry*. 2020. Т. 12, № 4. С. 551. URL: https://doi.org/10.3390/sym12040551
13. Lv J., Wei M., Yu Y. A Container Scheduling Strategy Based on Machine Learning in Microservice Architecture. *2019 IEEE International Conference on Services Computing (SCC)*. 2019. С. 65–71.
14. Zhao X., Wu C. Large-scale Machine Learning Cluster Scheduling via Multi-agent Graph Reinforcement Learning. *IEEE Transactions on Network and Service Management*. 2021. С. 1. URL: https://doi.org/10.1109/tnsm.2021.3139607
15. Voievodin Y. V., Stabetska T. A., Rozlomii I. O.THE IMPACT OF CLUSTER TOPOLOGY ON THE EFFICIENCY OF RESOURCE DISTRIBUTION USING KOHONEN SELF-ORGANIZING MAPS IN CONTAINER ORCHESTRATION SYSTEMS. *Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences*. 2025. Т. 2, № 1. С. 39–45. URL: https://doi.org/10.32782/2663-5941/2025.1.2/06
16. Securing Container-based Clouds with Syscall-aware Scheduling / M. Le та ін. *In Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 2023. С. 812–826.
17. Voievodin Y., Rozlomii I. Advanced software framework for comparing balancing strategies in container orchestration systems. *Proceedings of the 4th Edge Computing Workshop (doors 2024*. 2024. Т. 3666. С. 60–69.