

**Восводін Є.В.**

Черкаський національний університет імені Богдана Хмельницького

**Стабецька Т.А.**

Черкаський національний університет імені Богдана Хмельницького

**Розломій І.О.**

Черкаський державний технологічний університет

## ВПЛИВ ТОПОЛОГІЇ КЛАСТЕРУ НА ЕФЕКТИВНІСТЬ РОЗПОДІЛЕННЯ РЕСУРСІВ З ВИКОРИСТАННЯМ САМООРГАНІЗАЦІЙНИХ КАРТ КОХОНЕНА В СИСТЕМАХ ОРКЕСТРУВАННЯ ВІРТУАЛЬНИХ КОНТЕЙНЕРІВ

У статті розглянуто вплив топології кластеру на ефективність розподілення ресурсів у системах оркестрування віртуальних контейнерів. На прикладах, що базуються на самоорганізаційній карті Кохонена, продемонстровано, наскільки різне розташування вузлів може впливати на рівень заповненості кластеру та інші показники ефективності процесу розподілення.

Розглядаються різні аспекти налаштування самоорганізаційної карти Кохонена: розмірність простору (одновимірний, двовимірний тощо), функції пошуку відстані, а також способи визначення вагових коефіцієнтів. Детально описується процес навчання карти, що відіграє ключову роль, адже саме під час нього формуються вагові коефіцієнти нейронів карти. Ці коефіцієнти безпосередньо впливають на те, які вузли обираються для розміщення контейнерів. Залежно від топології кластеру, контейнер може потрапити до вузла з недостатніми ресурсами, що підвищує ризик швидкого вичерпання ресурсів або негативно впливає на інші показники, такі як кількість відмов на запити балансування.

Проведений аналіз показує, що за однакових налаштувань самоорганізаційної карти та ідентичних послідовностей запитів на балансування контейнерів одні топології можуть забезпечувати більш ніж на 10% ефективніше використання ресурсів порівняно з іншими.

Для підвищення точності оцінки було розширено попередньо запропонований метод тестування, додавши можливість досліджувати й порівнювати велику кількість варіантів розташування вузлів за довільними критеріями, такими як рівень заповненості кластеру, кількість відхилених запитів балансування, відмовостійкість системи тощо. Також розглянуто варіант паралельного та розподіленого виконання експерименту, що дозволяє скоротити час обчислень та оперативніше отримати множину найбільш підходящих топологій кластеру.

Описана методика і результати експериментів становлять інтерес для дослідників, які прагнуть покращити ефективність розподілення ресурсів, у системах оркестрування віртуальних контейнерів.

**Ключові слова:** системи оркестрування віртуальних контейнерів, віртуалізація, розподілені системи, нейронні мережі, самоорганізаційні карти Кохонена.

**Постановка проблеми.** Системи оркестрування віртуальних контейнерів (СОВК) відіграють важливу роль у сучасних розподілених системах та хмарних сервісах. Вони відповідають за розподіл віртуальних контейнерів (що репрезентують сервіси, процеси, додатки) на фізичних кластерах, а також за організацію взаємодії між цими контейнерами. Одним із основних елементів СОВК є стратегія розподілення ресурсів (СРР), яка визначає, на якому фізичному вузлі розміщуватиметься контейнер. Існують різні реалізації СРР, серед стандартних: *Binpack* (розподілення запо-

вненням) та *Spread* (розподілення поширенням). Ефективність СРР зазвичай оцінюють за кількома показниками: рівнем дефрагментації, успішністю запитів розподілення, відмовостійкістю системи тощо. Як свідчать дослідження, стандартні стратегії не завжди є достатньо ефективними, що спонукає до розроблення й впровадження спеціалізованіших СРР. Однією з них є стратегія на основі самоорганізаційних карт Кохонена (СКК). СКК – це вид штучної нейронної мережі, що належить до нерегламентованого навчання (навчання без учителя). У такій мережі кожен вхідний век-

тор (наприклад, набір характеристик контейнера) проектується на  $n$ -вимірну карту, утворюючи кластери подібних векторів без попередньо заданих класів.

СРР, що базується на самоорганізаційних картах Кохонена, дає змогу обирати фізичну локацію вузла в кластері, використовуючи подібність конфігурацій контейнерів. Показником подібності слугує набір властивостей, якими описується контейнер, наприклад: кількість оперативної пам'яті (ОП), кількість ядер центрального процесора (ЦП) чи то обсяг фізичної пам'яті на диску. Крім цього, СКК задається набором функціональних параметрів (як-от метод навчання, початкова ініціалізація вагових коефіцієнтів, функція пошуку відстані між вектором параметрів контейнера та нейронами карти, функція обмеження топологічної околиці нейронів, функція швидкості навчання СКК тощо), що впливає на те, як карта навчається й далі групує компоненти.

Процес пошуку вузла з використанням СРР на базі СКК можна описати так (Рис. 1):

- створюється СКК із кількістю нейронів, що дорівнює кількості вузлів у кластері;
- здійснюється навчання СКК згідно із заданими функціональними параметрами;
- для кожного нового запиту на розподілення ресурсів визначається вектор параметрів контейнера, який адаптується за тим самим алгоритмом, що й під час навчання;
- виконується ранжування всіх вузлів на основі відстані до нейронів карти;
- обирається той вузол (згідно з функцією відстані), що найбільше відповідає конфігурації контейнера і має достатньо ресурсів для його розміщення.

Важливо зазначити, що в цьому алгоритмі основну роль у виборі вузла відіграють: параметри налаштування СКК, характеристики самого контейнера, фактична наявність фізичних ресурсів безпосередньо на вузлі. Топологія кластеру (наприклад, порядок розміщення вузлів) у такій базовій реалізації не враховується.

Розгляньмо приклад двох різних топологій кластеру із двома вузлами (максимальна місткість – 5 ГБ та 3 ГБ ОП відповідно), які відрізняються порядком розміщення цих вузлів. Припустимо, що спочатку розміщуємо контейнер із вимогою 3 ГБ ОП і СРР обирає перший вузол для його розміщення, тоді маємо наступний стан кластеру (Рис. 2).

Якщо наступним контейнером в динамічній послідовності буде контейнер з вимогами 5 ГБ,

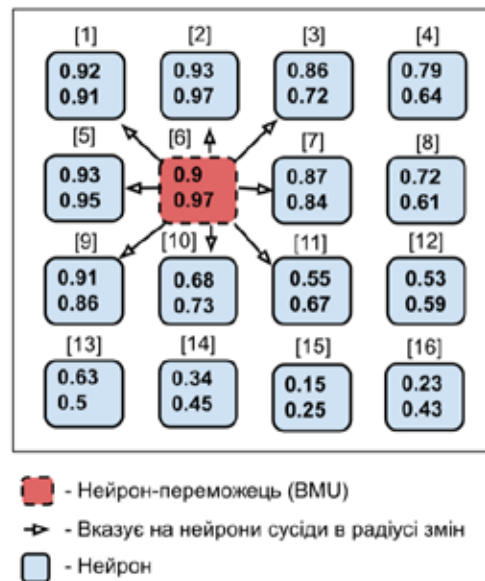


Рис. 1. Приклад структури СКК з топологією квадратної матриці для вектору вхідних даних з двох параметрів

то у випадку першої топології, такий контейнер можна буде розмістити на другому вузлі, тоді як у випадку другої топології, в кластері недостатньо місця для його розміщення. Якщо ж повна динамічна послідовність сформована з наступних вимог: 3 ГБ, 5 ГБ, 3 ГБ, 2 ГБ, то у випадку першої топології загальна кількість розміщених контейнерів буде 2 (3 ГБ та 5 ГБ), тоді коли у випадку другої топології загальна кількість розміщених контейнерів буде 3 (3 ГБ, 3 ГБ, 2 ГБ).

Оскільки СОВК можуть застосовуватися для найрізноманітніших завдань та мають різні цілі, кожен із наведених варіантів може виявитися бажаним в окремій ситуації. Водночас можна припустити, що ефективність СРР для конкретної ситуації можна підвищити, якщо стратегія розподілення буде брати до уваги топологію кластеру.

**Аналіз останніх досліджень і публікацій.** Для проведення дослідження варто звернути увагу на метод запропонований Воеводіним Є. В. та Розломій І. О. [1], який детально описує алгоритм тестування СРР із врахуванням різноманітних цільових параметрів. Приміром, цей метод описує компонент (генератор топологій), який дає змогу експериментувати з різними топологіями кластеру порівнюючи ефективність результатів розподілення. Хоча даний метод є лімітованим з точки зору кількості топологій, для яких можна провести експеримент, оскільки результат для кожної з топологій зберігається для подальшого представлення. Автори наголошують, що тестування

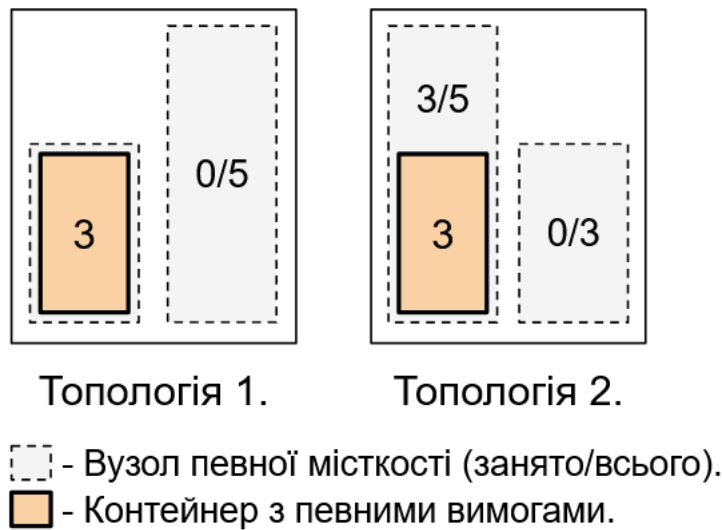


Рис. 2. Приклад розміщення послідовності контейнерів у кластерах з однаковою місткістю вузлів, але з різним порядком

стратегій розподілення є нетривіальним завданням, оскільки потребує врахування можливої варіативності поведінки стратегій та систем які використовують COBK.

В іншій своїй роботі Воеводін Є. В. та Авраменко В. С. досліджують вплив топологій СКК на ефективність процесу розподілення [2]. Вони демонструють, що вибір топології карти по-різному позначається на ефективності розподілення за різних умов. Наприклад, одновимірна топологія краще працює для кластерів із невеликою кількістю вузлів, тоді як для більших кластерів ефективнішою є топологія квадратної матриці.

Огляд можливостей COBK у сучасних системах (зокрема Kubernetes, Docker Swarm, Apache Mesos) подано у роботі Е. Труєна та ін. [3]. Автори висвітлюють різноманітні функціональні можливості цих систем, а також визначають спільні елементи, серед яких: наявність кластеру представленому у вигляді набору вузлів з певними вимогами; компонент, що відповідає за балансування ресурсів (тобто СРР). Аналогічне порівняння COBK Р. Файос-Джордан та ін. і в проводять в своєму дослідженні [4], але з акцентом на кластери з мало-місткими вузлами.

Актуальність напрямку дослідження також підтверджують публікації, присвячені створенню СРР з використанням машинного навчання [5], застосуванню мурашиного алгоритму [6] для підвищення ефективності розподілення ресурсів, а також упровадженню методів багатоцільової оптимізації [7].

**Постановка завдання.** Продемонструвати, як топологія кластеру впливає на ефективність розподілення ресурсів на прикладі стратегії із використанням СКК. Запропонувати розширений метод для порівняння ефективності СРР із урахуванням ранжування результатів залежно від топології.

**Виклад основного матеріалу.** Для проведення експериментів буде використовуватись наступна конфігурація самоорганізаційної карти (якщо окремі модифікації не зазначені).

Евклідова відстань як функція пошуку відстані:

$$d(\vec{x}, \vec{w}) = \sqrt{\sum_{i=1}^N (x_i - w_i)^2},$$

де  $\vec{x}$  – вектор вхідних даних,  $\vec{w}$  – вектор вагових коефіцієнтів нейрона,  $N$  – кількість параметрів якими описується карта (кількість елементів вектору).

Функція Гауса для обмеження топологічної околиці навчання:

$$h_{bj}(t) = e^{-\frac{d^2}{2\left(\sigma e^{-\frac{t}{T}}\right)^2}},$$

де  $h_{bj}$  – описує на скільки змінюється нейрон за індексом  $j$  в залежності від того як далеко він знаходиться від нейрона переможця за індексом  $s$  (найближчого нейрона відповідно до функції відстані),  $d$  – відстань між нейронами за індексами  $b$  та  $j$ ,  $t$  – ітерація навчання,  $T$  – загальна кіль-

кість ітерацій,  $\sigma$  – початкова ширина топологічної околиці.

Функцію експоненціального розпаду, для контролю швидкості навчання, з деякими іншими параметрами відмінно від використаної у функції обмеження топологічної околиці:

$$\alpha(t) = a_0 e^{-\frac{t}{k}},$$

де  $a_0$  – початкове значення швидкості навчання,  $k$  – стале значення, яке контролює як швидко змінюється значення функції відповідно до ітерації  $t$ .

Корегування вагових коефіцієнтів кожного з нейронів карти здійснюється за наступною формулою:

$$\bar{w}_j(t+1) = \bar{w}_j(t) + \alpha(t) h_{b_j}(t) [\bar{x}(t) - \bar{w}_j(t)],$$

де  $j$  – індекс нейрона в карті,  $t$  – ітерація навчання,  $b$  – індекс нейрона переможця на даній ітерації відповідно до вхідного вектору даних  $\bar{x}$ , та решта попередньо визначені функції. Навчання карти здійснюється на послідовності векторів на основі параметрів початково заданої послідовності, повтореної 1000 разів, тобто  $T = 1000$ .

Для оцінки якості розподілення ресурсів у кластері як цільову метрику пропонується використовувати коефіцієнт заповненості (що коливається в межах від 0 до 1 і фактично відображає відсоток використання ресурсів у кластері). Чим вищий цей коефіцієнт, тим кращим є результат розподілення, адже при високому рівні заповненості зменшується зовнішня дефрагментація ресурсів.

Проведемо перший експеримент. Спочатку оберемо одновимірну топологію СКК та кластер, що складається з чотирьох вузлів: два з них мають

обсяг 6 ГБ, інші ж 2 – 10 ГБ. Використовуючи СРР на основі описаної СКК, виконаємо розподіл послідовності контейнерів із вимогами 5 ГБ, 3 ГБ, 5 ГБ, 3 ГБ, повторюючи її доти, доки в кластері не залишиться достатньо ресурсів для розміщення контейнера з мінімальною конфігурацією – 3 ГБ. Після цього перевіримо рівень дефрагментації для різних топологій кластеру: 10 ГБ, 6 ГБ, 6 ГБ, 10 ГБ та 10 ГБ, 10 ГБ, 6 ГБ, 6 ГБ, 10 ГБ (Рис. 3). Вагові коефіцієнти нейронів карти залишаються однаковими для обох топологій.

У підсумку бачимо, що в першій топології вдалося розмістити 8 контейнерів, використавши при цьому всі доступні ресурси кластеру. Відповідно, зовнішня дефрагментація становить 0%. Натомість у другій топології вдалося розмістити 7 контейнерів, причому кластер заповнений на 91%, тобто 9% ресурсів залишаються невикористаними до зупинки якогось із контейнерів.

Для визначення найбільш придатної топології під конкретну стратегію можна створити генератор топологій, який (відповідно до описаного механізму проведення експерименту [1, с. 62]) генерує унікальні перестановки вузлів у кластері. Зокрема, для цієї мети можна скористатися алгоритмом Джонсона – Троттера [8, с. 215], що дає змогу ефективно отримати всі можливі унікальні перестановки.

Проведемо другий експеримент для СРР на базі СКК, застосовуючи топологію квадратної матриці та збільшуючи кількість вузлів до 9, серед яких 3 мають обсяг 10 ГБ, а 6 – 6 ГБ. Використовуючи описаний процес генерації, перевіримо всі унікальні топології кластеру, орієнтуючись на коефіцієнт заповненості (Рис. 4).

З отриманих результатів видно, що у більшості згенерованих топологій заповненість кластеру

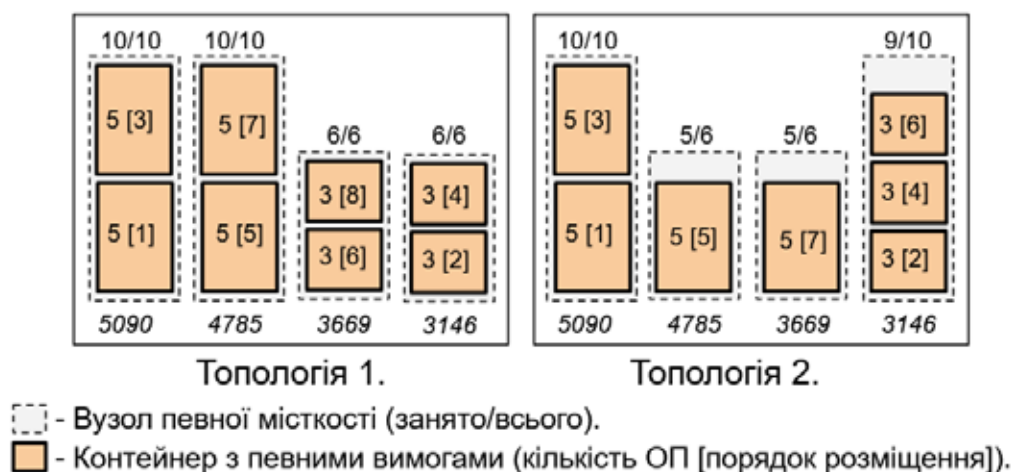


Рис. 3. Заповненість кластерів в результаті проведення першого експерименту

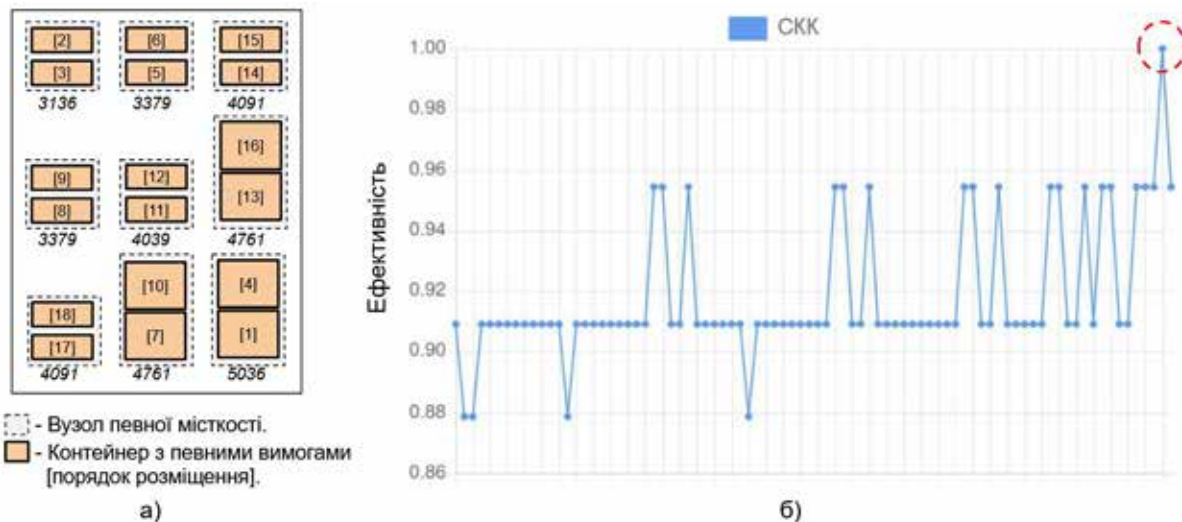


Рис. 4. а) Результат розподілення для найбільш оптимальної топології;  
 б) Ефективність заповнення для кожної згенерованої топології

перебуває близько 90%, тоді як лише в одному випадку кластер було заповнено на 100%, найгіршим результатом є 88%. Окрім того, маючи змогу згенерувати всі унікальні топології можна знайти найбільш підходящу. Саме це й підтверджує проведений експеримент: серед усіх можливих варіантів знаходиться така топологія, що забезпечує 100% заповнення кластеру, і, відповідно, найвищу ефективність розподілення ресурсів.

Оскільки метою експерименту є відшукати таку топологію, що дає змогу досягти максимальної ефективності, доцільно модифікувати попередньо запропонований алгоритм і додати спеціальний компонент для порівняння та відсіювання результатів із гіршими показниками (Рис. 5). Завдяки цьому можна буде знайти найбільш оптимальні топології навіть серед сотень можливих варіантів. Особливо це актуально тоді, коли контейнери та вузли описуються кількома параметрами, що збільшує загальну кількість унікальних топологій (адже вузол із 10 ГБ ОП та 6 ЦП відрізняється від вузла з 10 ГБ ОП та 8 ЦП). Такий компонент може керуватися різними критеріями ефективності, а також поєднувати їх, наприклад, орієнтуватися на мінімальний рівень зовнішньої дефрагментації та при однаковому його значенні порівнювати кількість відхилених запитів на розподіл ресурсів, – менша кількість відхилень засвідчує вищу ефективність. Крім того, можна взяти до уваги й відмовостійкість системи, порівнюючи різні топології за цим параметром.

Також варто зазначити, що цей алгоритм можна покращити, виконавши експеримент пара-

лельно – як на одному обчислювальному пристрої, так і в межах розподіленої системи. Наприклад, якщо є два вузли, можна розподілити набір усіх топологій навпіл: першу частину згенерувати та перевірити на першому вузлі, а решту – на другому. Згодом результати з обох вузлів об'єднують, керуючись тим самим принципом, що описаний в алгоритмі. Скажімо, якщо потрібно знайти 10 найефективніших топологій, то можна обрати по 10 найкращих із за результатами роботи кожного з вузлів.

**Висновки.** У цій статті продемонстровано, що топологія кластеру здатна істотно впливати на ефективність розподілення ресурсів у СОВК. На прикладі стратегії, побудованої на СКК, показано, що навіть при однакових налаштуваннях СКК різний порядок розташування вузлів може призвести до істотних відмінностей у рівні зовнішньої дефрагментації. Отримані результати показують, що в межах одного експерименту така різниця може досягати 10% і більше.

В статті також було запропоновано метод для поліпшення оцінки ефективності різних стратегій розподілення:

- розширити метод тестування за допомогою підтримки генерації всіх унікальних перестановок вузлів у кластері;
- додати компонент порівняння та відсіювання менш ефективних результатів, що дає змогу швидше знайти найбільш оптимальні топології серед великої кількості варіантів, не зберігаючи та не аналізуючи проміжні результати;
- використовувати багатокритеріальні показники для оцінювання, зокрема одночасно врахову-

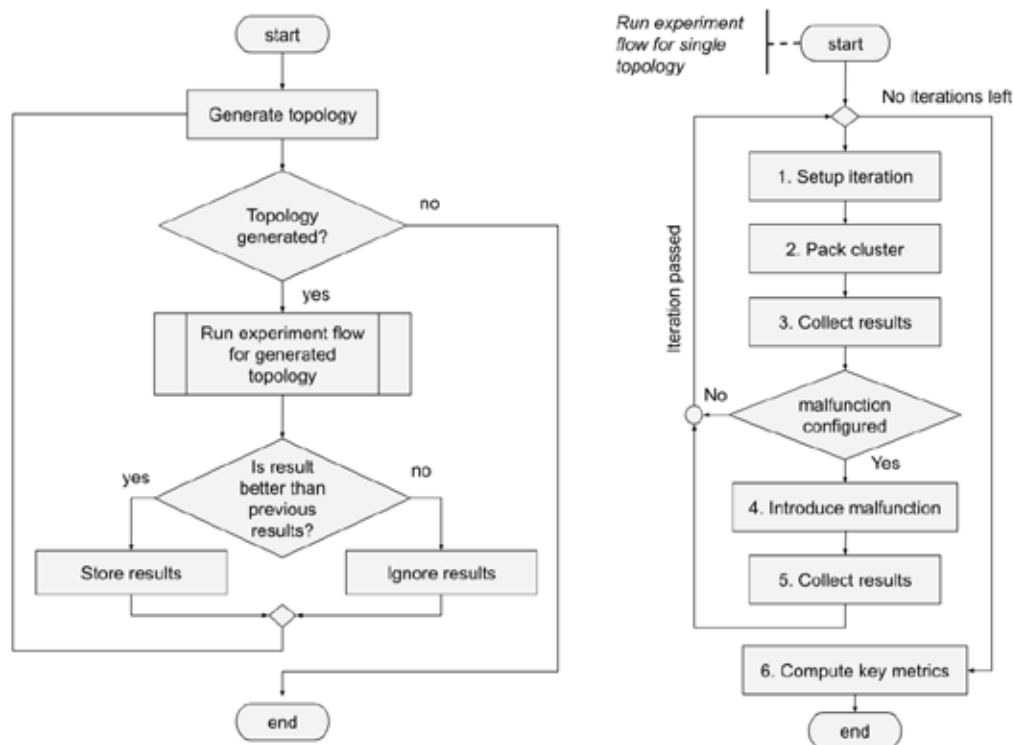


Рис. 5. Модифікований алгоритм проведення експериментів для пошуку оптимальних топологій кластеру

вати рівень зовнішньої дефрагментації, кількість відхилених запитів, відмовостійкість системи та інші показники ефективності;

– оптимізувати процес експерименту через паралельне чи розподілене виконання,

що прискорює перебір великої кількості конфігурацій.

Отримані результати підтверджують важливість урахування топології кластеру для підвищення загальної ефективності COBK.

#### Список літератури:

1. Voievodin Y., Rozlomii I. Advanced software framework for comparing balancing strategies in container orchestration systems. *Empowering the Edge: Research advances from doors 2024*. 2024. Vol. 3666. P. 60–69.
2. Воєводін Є., Авраменко В. Порівняння ефективності топологій самоорганізаційних карт Кохонена в системах оркестрування віртуальних контейнерів. *Вчені записки Таврійського національного університету імені ВІ Вернадського. Серія: Технічні науки*. 2018. № 29. С. 99–105.
3. A Comprehensive Feature Comparison Study of Open-Source Container Orchestration Frameworks / E. Truyen et al. *Applied Sciences*. 2019. Vol. 9, no. 5. P. 931. URL: <https://doi.org/10.3390/app9050931> (date of access: 28.01.2025).
4. Performance comparison of container orchestration platforms with low cost devices in the fog, assisting Internet of Things applications / R. Fayos-Jordan et al. *Journal of Network and Computer Applications*. 2020. Vol. 169. P. 102788. URL: <https://doi.org/10.1016/j.jnca.2020.102788> (date of access: 28.01.2025).
5. Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions / Z. Zhong et al. *ACM Computing Surveys (CSUR)*. Vol. 54, no. 10. P. 1–35.
6. Kaewkasi C., Chuenmuneewong K. Improvement of container scheduling for Docker using Ant Colony Optimization. *2017 9th International Conference on Knowledge and Smart Technology (KST)*. 2017. P. 254–259.
7. A new container scheduling algorithm based on multi-objective optimization / B. Liu et al. *Soft Comput.* 2018. Vol. 22. P. 7741–7752.
8. Bird R. *Pearls of Functional Algorithm Design*. Cambridge University Press, 2010.

**Voievodin Ye.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**

*This paper examines the impact of cluster topology on the efficiency of resources distribution in container orchestration systems. Using examples based on the balancing strategy that utilizes Kohonen self-organizing map, it demonstrates how different node arrangements can influence usage of cluster resources and impact other key efficiency metrics.*

*Various aspects of configuring the self-organizing map are discussed, including the dimensionality of the map (one-dimensional, two-dimensional, etc.), distance functions, and methods for determining weight coefficients. The training process of the map is explained in detail, emphasizing how neuron weigh coefficients are established during this stage. These coefficients directly affect which nodes are chosen for container placement. Depending on the topology of the cluster, a container may be assigned to a node that lacks sufficient resources, which increases the risk of resource depletion or impacts other metrics, such as the number of failed balancing requests.*

*The analysis shows that with identical self-organizing map configurtation and the same balance requests sequence, certain topologies can achieve more than a 10% increase in resource usage efficiency compared to others. To enhance the accuracy of evaluation, the previously introduced testing method has been extended to examine and compare a large number of node arrangement variants according to various criteria, including the efficient use of cluster resources, the number of rejected balancing requests, and system fault tolerance. The paper also considers parallel and distributed approaches to running experiments, which helps reduce computation time and enables a more rapid identification of the most suitable cluster topologies.*

*The methodology and experimental findings described here are of interest to researchers who seek to improve resource allocation efficiency in container orchestration systems.*

**Key words:** container orchestration systems, virtualization, distributed systems, neural networks, Kohonen self-organizing maps.