

**Воєводін Є.В.**

Черкаський національний університет імені Богдана Хмельницького

## МЕТОДИ ТЕСТУВАННЯ ЕФЕКТИВНОСТІ СТРАТЕГІЙ РОЗПОДІЛЕННЯ В СИСТЕМАХ ОРКЕСТРУВАННЯ ВІРТУАЛЬНИХ КОНТЕЙНЕРІВ

*У статті описуються методи тестування стратегій розподілення в ізолюваному середовищі з імітацією навантаження. Оскільки кожна стратегія вирішує окрему конкретну проблему в доменній області, то критерії для їх тестування та підходи є зовсім різними. Спочатку розглядається структура системи, в контексті якої проводиться тестування, й описуються методи, також дається визначення стратегії розподілення та коротко описуються стандартні рішення. Першим методом, що розглядається, є розподілення до першого відхилення. Такий підхід допомагає зрозуміти, яка зі стратегій надає змогу розподілити більше контейнерів без єдиної помилки. Другим методом є розподілення повноти кластеру. На відміну від першого, цей метод дозволяє проаналізувати кількість відхилених і вдало заповнених контейнерів до того моменту, поки у кластері не буде місця для мінімально можливої за розміром конфігурації. Окрім методів тестування, також розглядаються і способи аналізу та візуалізації даних, отриманих шляхом багатократного проведення експериментів.*

**Ключові слова:** розподілення, методи тестування, системне тестування, балансування, віртуалізація, контейнер, система оркестрування.

**Постановка проблеми.** Віртуалізація процесу розробки та використання віртуальних Linux-контейнерів для запуску додатків уже давно не є новинкою в сучасному світі розподілених систем. Витісняючи гіпервізори за рахунок своєї простоти й ефективності, такі технології, як LXC, Docker, rkt, OpenVZ, набувають популярності та практичної значимості. З розвитком технологій віртуалізації також розвиваються і засоби роботи з ними. Одним із таких засобів є системи оркестрування віртуальних контейнерів (далі – СОВК). Такі системи дають змогу використовувати технологію віртуалізації на множині вузлів, забезпечуючи їхню ефективну роботу, що відкриває нові можливості для розподілених додатків і більш складних систем. До таких СОВК можна віднести Docker Swarm, Kubernetes, MesosMarathon.

СОВК мають змогу працювати з динамічним рядом конфігурацій контейнерів, тобто послідовно здійснювати розподілення ресурсів за вимогою. Однією із задач СОВК є пошук вільного місця у кластері, т. зв. вузла, який би задовольнив вимоги контейнера, що створюється конфігурацією. Для виконання цієї задачі СОВК використовують різні стратегії розподілення. У випадку із СОВК Docker Swarm ними є стратегія розподілення заповненням, стратегія розподілення поширенням і стра-

тегія розподілення на випадково обраних вузлах. Кожна зі стратегій описується характерними ідентифікаторами ефективності та використовується для конкретних цілей. Так, наприклад, стратегія розподілення поширенням використовується для досягнення максимальної відмовостійкості системи [1], що працює в кластері СОВК.

Від ефективності стратегії розподілення та структури кластеру залежить ефективність роботи СОВК, кількість ресурсів, які використовуються, стабільність роботи системи розподілення та додатків, що запуснені в межах її кластеру. Наприклад, у разі неефективного розподілення контейнерів СОВК може динамічно додавати нові вузли, що потребує більшої кількості ресурсів і затрат на забезпечення їхньої роботи. Результат роботи стратегій розподілення є критично важливим для СОВК, оскільки від нього залежить поведінка системи, наприклад у разі неефективного розподілення. Для нової конфігурації може не вистачати місця в кластері, відповідно, СОВК буде змушена або відхилити запит на створення контейнера, або додати новий вузол до кластеру, що відповідно потребує часу та ресурсів апаратного забезпечення.

**Аналіз останніх досліджень і публікацій.** Проблеми використання наявних методів у своїй

роботі описує Є.В. Воєводін [2]. Автор порівнює між собою методи розподілення заповнення і розподілення поширенням, наводячи їхні переваги та недоліки. У роботі продемонстровано, за яких умов стратегії здійснюють неефективне балансування, й описується можлива поведінка СОВК у критичних випадках.

У своїх інших роботах [3; 4] Є.В. Воєводін та В.С. Авраменко описують метод розподілення контейнерів із використанням штучних нейронних мереж, зокрема самоорганізаційних карт Кохонена (далі – СКК) [5]. У ході досліджень проводиться ряд експериментів над стратегіями та здійснюється порівняння ефективності результатів розподілення. У випадку із СКК ефективність розподілення залежить від багатьох критеріїв, таких як швидкість навчання карти чи її топології. Для перевірки ефективності розподілення використовуються упаковка до першого відхилення та до повноти кластеру. Представлені у вигляді різних графіків результати тестування демонструють, що ефективність стратегій залежить від вибірки та структури кластеру.

Нову стратегію розподілення ресурсів у системі Docker Swarm, що базується на угоді про рівень послуг, у своїй роботі описують Крістоф Серін та ін. [6]. Спочатку автори детально розбирають структуру СОВК та інтерфейсу стратегії розподілення, після чого реалізують нову стратегію. Реалізація спрямована на забезпечення ефективної роботи СОВК у межах приватних хмарних систем. Автори тестують стратегії з використанням функціонального та ручного тестування.

**Постановка завдання. Метою статті** є детальний опис методів тестування ефективності стратегій розподілення в СОВК, їх основних властивостей і способів аналізу результатів розподілення.

**Виклад основного матеріалу дослідження.** Для початку розглянемо тестування методом розподілення до першого відхилення. Процес розподілення проводиться доти, поки не буде відхилено запит на балансування наступного в динамічному ряді контейнера. Під відхиленням мається на увазі ситуація, коли стратегія розподілення не може помістити контейнер на якийсь із вузлів, оскільки у кластері просто немає вільного місця для контейнеру з такими вимогами. Якщо взяти один і той же динамічний ряд і стільки аналогічних екземплярів кластеру, скільки стратегія бере участь в експерименті, то можна перевірити, яка зі стратегій розподіляє ресурси краще в контексті цього методу. Експеримент складається із певної

кількості ітерацій для різних динамічних рядів. Одна ітерація експерименту закінчується тоді, коли для кожної стратегії та екземпляру кластеру буде відхилений контейнер.

Для опису алгоритму роботи методу розглянемо сутність кластеру. Кластер є композицією вузлів із можливістю застосування відповідної стратегії для балансування контейнера на одному з вузлів із такої композиції. Конкретна реалізація стратегії не є суттєвою в контексті цієї роботи, тому детально не розглядається. Розподілення інкапсульовано за викликом функції *Add* на екземплярі сутності кластеру. Розглянемо потенційно можливу структуру сутності кластеру з мінімальним функціоналом на діаграмі класів (рис. 1).

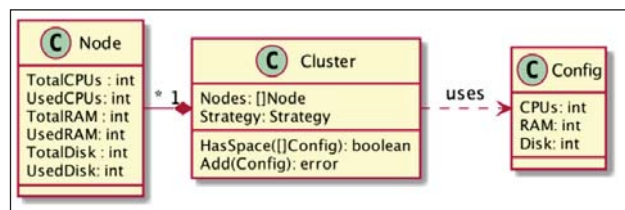


Рис. 1. Діаграма класів кластеру вузла та конфігурації контейнера

Кожна конфігурація описується такими вимогами, як кількість ядер центрального процесора (CPU), кількість оперативної пам'яті (RAM) та розмірність диску (Disk). Відповідно, вузли в кластері забезпечують ці вимоги.

Алгоритм однієї ітерації експерименту для розподілення до першого відхилення зображено на блок-схемі (рис. 2).

Основні метрики, які можна отримати шляхом проведення експерименту, називатимемо ключовими ідентифікаторами ефективності (далі – КІФ). Для цього методу КІФ можуть бути такими:

1. Кількість контейнерів, які вдалося розмістити в кластері до моменту першого відхилення (в межах однієї ітерації експерименту). Ця метрика дає змогу перевірити стабільність роботи стратегії. Наприклад, із використанням стратегії, яка може розподілити більше ресурсів до першого відхилення, можна динамічно додавати нові вузли до кластеру, причому кількість помилок буде мінімальною, відповідно, сервіс є стабільнішим.

2. Кількість випадків, коли з використанням однієї стратегії вдалося досягти кращого результату (в межах експерименту). Така метрика дозволяє перевірити, в якій кількості випадків одна стратегія поводить більш ефективно, ніж інша. Наприклад, із 10 000 ітерацій проведення експерименту в 8 000 з використанням першої стратегії вдалося розподілити більше контейнерів до

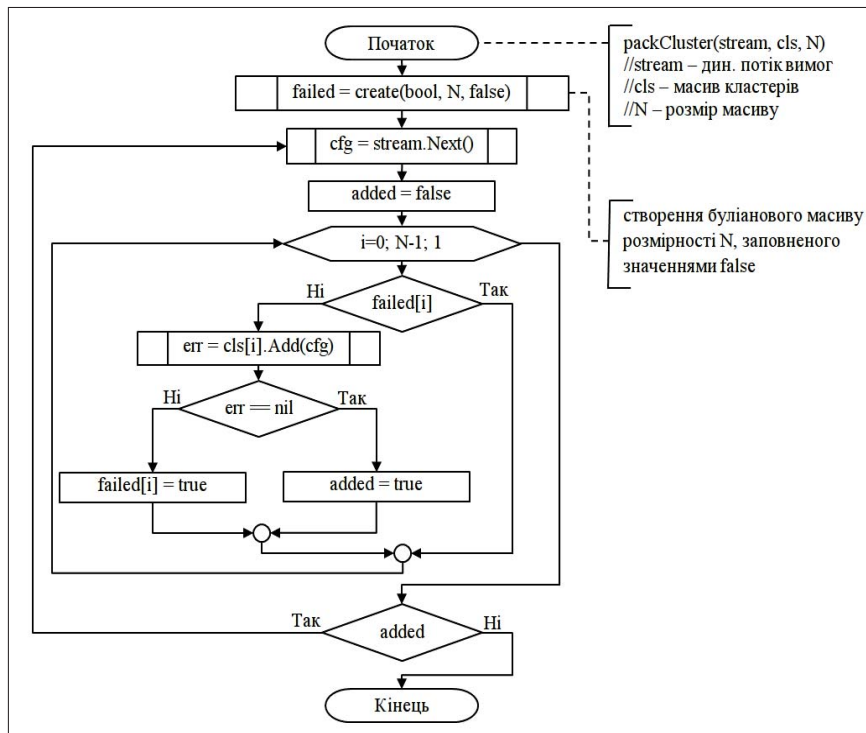


Рис. 2. Блок-схема роботи алгоритму розподілення до першого відхилення

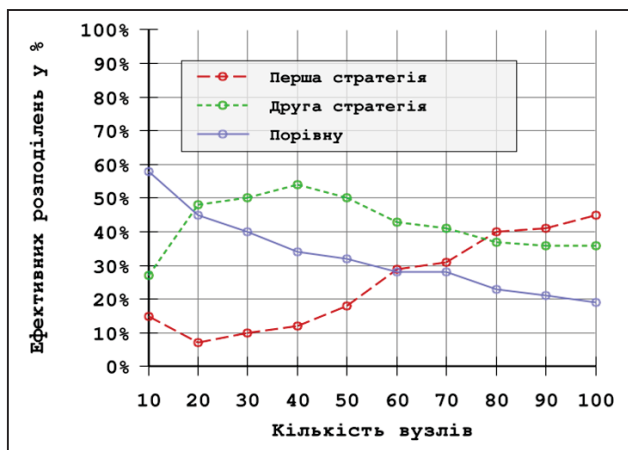


Рис. 3. Приклад градації ефективності двох стратегій розподілення

моменту першого відхилення, відповідно, це є 80% всіх випадків і є її стабільністю.

3. Кількість використаних ресурсів до моменту першого відхилення (в межах експерименту). Ця метрика показує ступінь повноти кластеру до моменту першого відхилення. Чим більше заповненим є кластер, тим ефективнішим буде розподілення.

Є сенс проводити експеримент із використанням різних структур кластеру й аналізувати отримані дані як результат багато-ітераційного повто-

рення експерименту на різних можливих вибірках. Для візуального представлення результатів можна використовувати різного роду графіки та діаграми. Одним із таких графіків може бути градація ефективності стратегії для кластерів різної розмірності (рис. 3).

Наступним розглянемо експеримент розподілення до повноти кластеру. Метод тестування полягає в тому, щоб запаковувати кластери доти, поки в кластері не буде місця хоча б для конфігурації з найменшими потребами. Як і в попередньому експерименті, використовувати одну й ту ж динамічну послідовність конфігурацій контейнерів. З використанням такого методу можна перевірити різні кількісні характеристики розподі-

лення. Детальний процес роботи експерименту зображено на блок-схемі (рис. 4).

Окрім функції *Add*, кластер тепер також повинен мати змогу перевірити, чи залишилося вільне місце для конфігурації, яка потенційно може бути повернута із динамічного потоку *HasSpace*. Також потрібно зазначити, що мінімальних конфігурацій контейнерів може бути декілька, оскільки конфігурація описується декількома можливими вимогами. Вектор конфігурації  $\vec{x}$  вважається меншим, ніж інший вектор  $\vec{y}$  тоді, коли кожний елемент першого вектора менший, ніж елемент другого вектора з тим же індексом  $\vec{x} < \vec{y} \rightarrow \vec{x}_i < \vec{y}_i, i = \{1, \dots, N\}$ , де  $N$  – це розмірність вектора вимог. Таким чином, якщо динамічний ряд формується на основі наступних векторів конфігурацій: (1, 5, 1); (1, 1, 5); (5, 5, 1), то мінімальними серед них є (1, 1, 5) та (1, 5, 1).

Як і для попереднього методу, опишемо потенційні КІФ, які можна використати для перевірки ефективності розподілення:

1. Кількість контейнерів, яку вдалося розподілити до моменту повного заповнення кластеру (в межах однієї ітерації експерименту). Ця метрика показує стан кластеру за умови максимально повного використання його ресурсів. Мається на увазі, що без зупинки якогось із контейнерів або ж додавання нового вузла жодний із контейнерів послідовності не може бути розподілений.

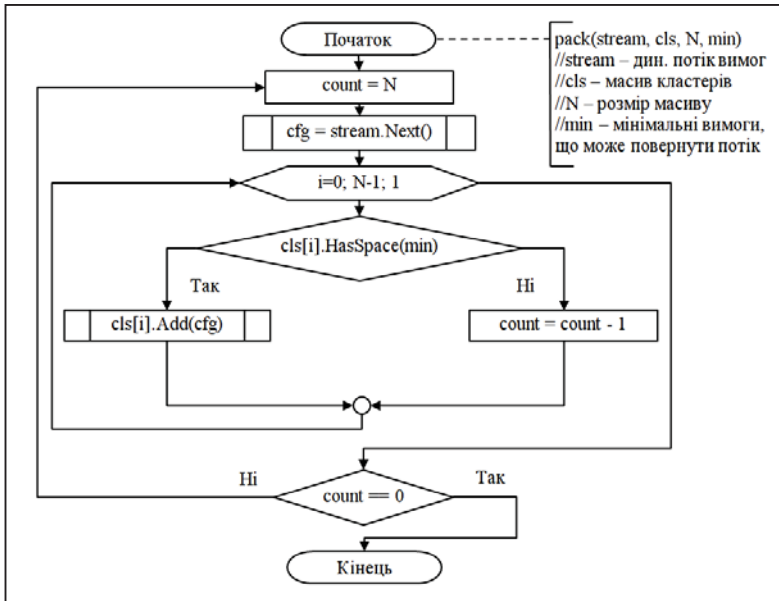


Рис. 4. Блок-схема роботи алгоритму розподілення до повноти кластеру

$$\frac{S}{N} \cdot \frac{\sum_{i=1}^M \bar{U}_i}{M}$$

де  $S$  – кількість вдало упакованих контейнерів;

$N$  – загальна кількість контейнерів, що були оброблені до моменту повної упаковки;

$\bar{U}$  – вектор кількості використаних ресурсів у кластері;

$\bar{T}$  – вектор загальної кількості ресурсів у кластері;

$M$  – кількість типів ресурсів у кластері (розмірність векторів  $\bar{U}$  та  $\bar{T}$ ).

Так, наприклад, якщо із 300 контейнерів 280 було успішно запаковано та вектори  $\bar{U} = (8,34,180)$ ,  $\bar{T} = (10,40,200)$ , тоді значення коефіцієнта ефективності дорівнюватиме

$0,93 \cdot 0,9 = 0,837$ . Таким чином, значення коефіцієнта зменшиться, якщо кількість вільних ресурсів збільшиться або ж зменшиться кількість вдало запакованих контейнерів щодо відхилених.

Як і в попередньому випадку, цей метод є сенс проводити для різної конфігурації кластеру, зокрема набору вузлів. Для візуального представлення результатів зручно використати гістограму (рис. 5).

**Висновки.** Для перевірки ефективності роботи стратегій розподілення можна використовувати різні методи та метрики. Залежно від мети цільової системи та критеріїв оцінки ефективності можна

2. Кількість контейнерів, які були відхилені в процесі заповнення кластеру (в межах однієї ітерації експерименту). Менша кількість відхилених контейнерів до моменту повного заповнення дає змогу оцінити рівень стабільності системи.

3. Кількість відхилених і вдало розподілених контейнерів (у межах експерименту). Дає змогу зрозуміти, яка зі стратегій є більш ефективною для різного набору вузлів і динамічних рядів.

4. Для числового виміру того, чи є придатною структура кластеру для використання з даними в динамічному ряді, можна підрахувати коефіцієнт ефективності стратегії за формулою:

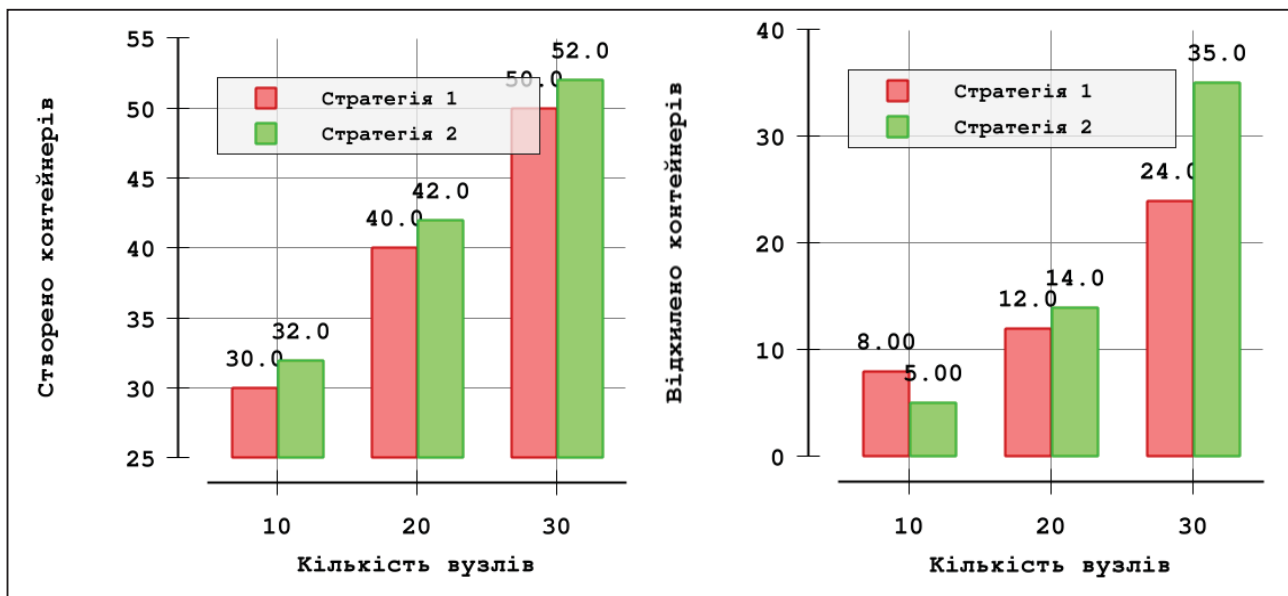


Рис. 5. Приклад гістограми кількісних характеристик результату розподілення до повноти

обирати різні стратегії розподілення і структури кластеру. У нашій роботі розглядаються методи тестування розподілення до першого відхилення та метод тестування упаковки до повноти кластеру, а також описуються алгоритми їхньої роботи у вигляді блок-схем.

Для кожного із методів розглядаються КІФ, які дають змогу зрозуміти, чи є стратегія придатною в конкретному випадку. Так, наприклад, на основі інформації про кількість відхилених контейнерів у процесі розподілення до повноти кластеру та кількості запакованих контейнерів до першого відхилення можна зробити висновок про стабіль-

ність системи. Або ж, наприклад, після проведення експерименту заповнення до повноти кластеру можна використати формулу для обрахунку коефіцієнта ефективності, що показує наскільки в межах від [0, 1] структура кластеру, динамічна послідовність і стратегія розподілення є сумісними та можуть бути ефективно використані.

Робота є суттєвою для проведення подальшого дослідження потенційних способів покращення роботи наявних стратегій розподілення та створення методів для оптимізації топологій кластерів. Окремого дослідження заслуговує тема тестування відмовостійкості системи.

#### Список літератури:

1. Таненбаум Э., Стеен М. Распределенные системы: принципы и парадигмы. Санкт-Петербург, 2003. 877 с.
2. Воеводін Є.В. Проблеми сучасних методів розподілення динамічної послідовності ресурсів у системах оркестрування віртуальних контейнерів. Науковий журнал «Альманах науки». 2017. № 4. С. 54–58.
3. Воеводін Є.В. Використання самоорганізаційних карт Кохонена для балансування динамічної послідовності ресурсів у системах оркестрування віртуальних контейнерів. Науковий журнал «Альманах науки». 2017. № 5. С. 45–50.
4. Воеводін Є.В., Авраменко В.С. Порівняння ефективності топологій самоорганізаційних карт Кохонена в системах оркестрування віртуальних контейнерів. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія «Технічні науки». 2018. № 1. С. 99–105.
5. Хайкин С. Нейронные сети: полный курс. Москва, 2006. 1104 с.
6. Cerin C., Menouer T., Saad W., BenAbdallah W. A New Docker Swarm Scheduling Strategy. URL: <https://lipn.univ-paris13.fr/~menouer/SC217.pdf>.

#### МЕТОДЫ ТЕСТИРОВАНИЯ ЭФФЕКТИВНОСТИ СТРАТЕГИЙ РАСПРЕДЕЛЕНИЯ В СИСТЕМАХ ОРКЕСТРАЦИИ ВИРТУАЛЬНЫХ КОНТЕЙНЕРОВ

*В статье описываются методы тестирования стратегий распределения в изолированной среде с имитацией нагрузки. Поскольку каждая стратегия решает конкретную проблему в доменной области, то критерии для их тестирования и подходы являются совершенно разными. Сначала рассматривается структуру системы, в контексте которой проводится тестирование и описываются методы, также дается определение стратегии распределения и кратко описываются стандартные решения. Первым рассматривается метод распределения к первому отклонению. Такой подход помогает понять, какая из стратегий дает возможность распределить больше контейнеров без единой ошибки. Вторым методом является распределения до полноты кластера. В отличие от первого метода, он позволяет проанализировать количество отклоненных и удачно упакованных контейнеров до того момента, пока в кластере не будет места для минимально возможной по размеру конфигурации. Кроме методов тестирования, также рассматриваются и способы анализа и визуализации данных, полученных в результате многократного проведения экспериментов.*

**Ключевые слова:** распределение, методы тестирования, системное тестирование, виртуализация, контейнер, система оркестрации.

#### METHODS TO TEST EFFICIENCY OF BALANCING STRATEGIES IN VIRTUAL CONTAINERS ORCHESTRATION SYSTEMS

*The paper covers methods for testing balancing strategies in isolated environment with synthetic traffic imitation. Every existing strategy is used to solve specific distribution problem in the domain area. Thus the criteria to test those strategies are completely different. At the beginning article describes the structure of the orchestration system and its main responsibilities. Then the proper definition of the balancing strategy is given with the examples of common solutions used by orchestration systems to distribute resources. The first described method is distribution until first request rejection. This method allows analyzing which strategy can distribute more containers without making a single mistake, which influences the overall stability of the orchestration system. Another described method is distribution until cluster is completely full. Unlike the previously described method this one allows to measure the amount of rejected and created containers during the whole flow, until there is no place for minimal possible container configuration. Apart from efficiency testing, the paper also lists some of the approaches to analyze and visualize results received from the intensive strategies testing.*

**Key words:** distribution, testing methods, system testing, virtualization, container, orchestration system.