DOI 10.31891/2307-5732-2024-339-4-43
УДК 004.67

БУРЯК МИКОЛА
Чернівецький національний університет імені Ю. Федьковича
https://orcid.org/0009-0006-7887-0091
e-mail: buriak.mykola@chnu.edu.ua
МЕЛЬНИЧУК СТЕПАН
Чернівецький національний університет імені Ю.Федьковича
e-mail: s.melnychuk@chnu.edu.ua

ОГЛЯД МЕТОДІВ ТА ЗАСОБІВ ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ РЕСУРСІВ В KUBERNETES КЛАСТЕРІ

У сучасному світі контейнеризація стала невід'ємною частиною управління IT-інфраструктурою, і Kubernetes відіграє ключову роль у цій еволюції. Однак оптимізація використання ресурсів в Kubernetes кластері залишається важливим завданням для підвищення ефективності та зниження витрат. Ця оглядова стаття присвячена аналізу сучасних методів і засобів оптимізації ресурсів у Kubernetes, які умовно поділені на три основні напрямки: машинне навчання, планування і масштабування, та архітектурні рішення.

На основі аналізу сучасних досліджень у статті робиться висновок, що комплексний підхід, який поєднує методи машинного навчання, ефективне планування і масштабування, продумані архітектурні рішення, дозволяє досягти вищих показників оптимізації ресурсів у Kubernetes кластері, а з використанням аналітичних інструментів цей показник може стати ще вищим. Цей огляд надає розуміння, що дослідження в напрямку методів оптимізації використання ресурсів в Kubernetes кластері є перспективними і важливими.

Ключові слова: kubernetes; розподілені системи; оптимізація ресурсів; машинне навчання; планування та масштабування; архітектурні рішення.

BURIAK MYKOLA
Yuriy Fedkovych Chernivtsi National University
MELNYCHUK STEPAN
Yuriy Fedkovych Chernivtsi National University

REVIEW OF METHODS AND TOOLS FOR OPTIMIZING RESOURCE UTILIZATION IN A KUBERNETES CLUSTER

In today's world, containerization has become an integral part of IT infrastructure management, with Kubernetes playing a key role in this evolution. However, optimizing resource usage in a Kubernetes cluster remains a crucial task for improving efficiency and reducing costs. This review article is dedicated to analyzing modern methods and tools for optimizing resources in Kubernetes, which are conventionally divided into three main areas: machine learning, scheduling and scaling, and architectural solutions.

The first part of the article covers machine learning-based methods, which are becoming increasingly popular due to their ability to analyze large volumes of data and predict resource needs. The article examines algorithms used for load prediction and resource allocation optimization. Approaches based on deep learning, neural networks, and other techniques that enhance the efficiency of computing power usage are discussed.

The second part focuses on scheduling and scaling methods, which include optimizing task scheduling and managing container autoscaling. Attention is given to mechanisms such as horizontal and vertical scaling, resource management through custom Kubernetes schedulers, and the use of Quality of Service (QoS) policies to ensure the stable operation of critical applications. Tools and extensions that improve cluster performance and reliability are also considered.

The third part is devoted to architectural solutions, which include optimizing infrastructure and cluster configuration for more efficient resource usage. Approaches to load balancing, the use of different types of nodes, changes in scheduling consensus, and integration with cloud services for dynamic resource management are discussed. Attention is also given to security and reliability aspects, which are critical for maintaining stable cluster operation.

In the fourth chapter, a comparative analysis of the main methods is carried out, considering the defined criteria: versatility, the absence of the need for deep configuration, and the presence of analytical tools. This analysis allows for identifying the advantages and disadvantages of each method, making it useful for selecting the most suitable approach for specific tasks. The primary focus is on how easily the methods adapt to different conditions and how effectively they can ensure resource optimization without significant configuration efforts.

Based on the analysis of current research, the article concludes that a comprehensive approach, combining machine learning methods, effective planning and scaling, and well-thought-out architectural solutions, allows for higher resource optimization in a Kubernetes cluster. With the use of analytical tools, this metric can be further improved. This review provides an understanding that research in the field of resource optimization methods in Kubernetes clusters is promising and important.

Keywords: kubernetes; distributed systems; resource optimization; machine learning; scheduling and scaling; architectural solutions.

Актуальність теми дослідження

У сучасному світі розгортання та управління розподіленими системами вимагає високої ефективності та гнучкості. Віртуалізація та контейнеризація вже давно стали основою для розробки сучасної цифрової інфраструктури, але керування такими розподіленими системами залишалося складною задачею до появи технології Kubernetes. Kubernetes, як відкрита система автоматизації розгортання, масштабування та керування контейнеризованими додатками, набуває все більшого значення в області сучасних хмарних обчислень.

Інтерес до оптимізації ресурсів Kubernetes кластера обумовлений потребою у максимальній продуктивності та оптимальному використанні обчислювальних та мережевих ресурсів.

Постановка проблеми

Важливою задачею є забезпечення стабільної та ефективної роботи кластера навіть у змінних умовах навантаження. З міркувань ефективності використання ресурсів та економії витрат виникає потреба у глибокому дослідженні оптимальних стратегій розподілу завдань та використання ресурсів у Kubernetes кластері.

У цій оглядовій статті буде проведено аналіз широкого спектру наукових праць який охоплює роботи, що присвячені проблемам управління та оптимізації ресурсів Kubernetes кластерів. Огляд допоможе визначити актуальність напрямку та виявити тенденції розвитку. Особлива увага буде зосереджена на оптимізації використання ресурсів Kubernetes кластера, зокрема, на розробці ефективних алгоритмів планування та управління ресурсами для забезпечення максимальної продуктивності та ефективності роботи системи.

Аналіз досліджень і публікацій

З огляду існуючих праць [1–24] в напрямку оптимізації використання ресурсів Kubernetes кластера було визначено високорівнево три основні методи:

- застосування моделей машинного навчання
- використання планування та масштабування
- архітектурні рішення

Частина праць застосовує комбіновані методи, проте розподіл зроблено ґрунтуючись на основних підходах. Розподіл праць за кількістю по методам оптимізації (рис. 1) дозволить зрозуміти актуальність напрямків досліджень.

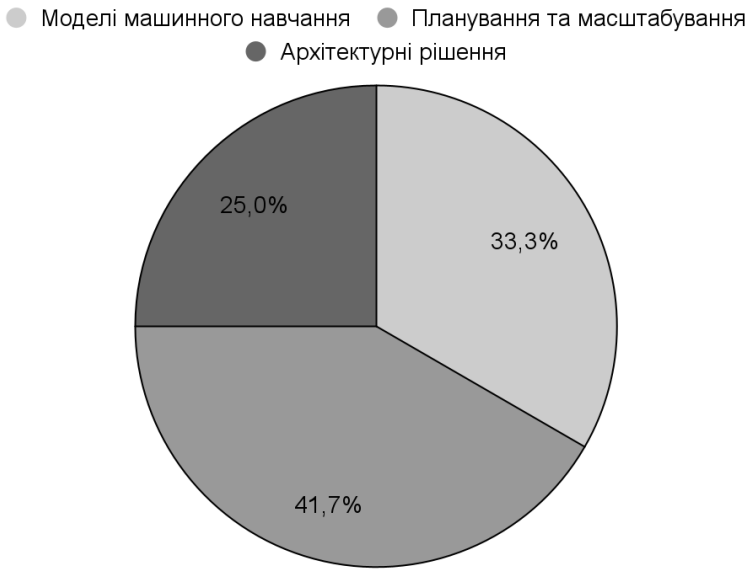

Рис. 1. Розподіл статей за методами

Мета полягає у виявленні потенційних переваг та обмежень різних підходів до оптимізації Kubernetes кластерів, а також у визначенні перспективних напрямків для подальших досліджень у цій важливій області хмарних обчислень.

Виклад основного матеріалу

1. Застосування моделей машинного навчання

Даний метод фокусується на створенні моделей машинного навчання для оптимізації використання ресурсів в Kubernetes кластері. Наступні огляди стосуються праць, що ґрунтуються на цьому підході.

В дослідженні [1] автори аналізують споживання ресурсів у кластері Kubernetes використовуючи моделі. Вони надають детальний опис того, як розгортаються та керуються додатки у Kubernetes, з особливим акцентом на їхню здатність спільно використовувати ресурси на фізичних або віртуальних машинах. Описано різні типи контейнеризованих додатків, включно з ініціалізаційними та допоміжними контейнерами, і те, як вони впливають на обчислення загального споживання ресурсів. Автори зосередились на деталях моделювання, що використовується для передбачення використання ресурсів, зокрема за допомогою мови моделювання ABS. Основний фокус зосереджено на управлінні ресурсами та розподілі навантаження між вузлами. Зокрема, розглядаються такі аспекти, як виділення пам'яті та ЦПУ в різних сценаріях використання, з метою оптимізації використання ресурсів у Kubernetes кластерах.

У праці [2] досліджується ефективність управління ресурсами в кластері Kubernetes. Автори використовують модель на базі мережі Петрі для аналізу цих процесів, проводячи експерименти на кластерах. Вони оцінюють, як різні параметри, такі як час розгортання контейнерів, кількість контейнерів в поді, кількість вузлів в кластері та інші впливають на продуктивність. Робота сприяє кращому розумінню та оптимізації розгортання і управління контейнеризованими додатками в розподілених середовищах.

Робота [3] досліджує ефективні методи еволюційної оптимізації з використанням прогнозування автомасштабування у контейнеризованому середовищі. Автори розробили рамкову структуру оптимізації, яка базується на хмарних технологіях і використовує мікросервіси, масштабовані за допомогою спеціально розробленого автоматичного масштабувальника PETAS, що базується на прогнозованих аналітичних даних. Ця система призначена для ефективного вирішення складних оптимізаційних задач, зокрема у сферах виробництва та автоматизованого машинного навчання, і може знизити витрати на інфраструктуру, зберігаючи при цьому швидкість отримання результатів.

У роботі [4] автори представляють метод управління ресурсами для мікросервісів у хмарних середовищах. Вони розробили систему POBO, яка використовує алгоритми оптимізації для забезпечення балансу між продуктивністю та безпечністю при розподілі ресурсів. Система допомагає знизити ризики, пов'язані з надмірним чи недостатнім розподілом ресурсів, що особливо важливо у великих хмарних системах, де мікросервіси можуть вимагати швидкого масштабування. Автори зосереджуються на двох основних аспектах: безпечності (виключення несанкціонованого доступу або витоку даних між сервісами) та оптимізації (ефективне використання ресурсів для забезпечення максимальної продуктивності за мінімальних витрат). POBO аналізує поточні вимоги до ресурсів і прогнозує майбутні потреби, динамічно коригуючи розподіл ресурсів для кожного мікросервісу в залежності від змін у навантаженні та пріоритетності задач. Система була протестована на реальних обчислювальних задачах у сферах виробництва і машинного навчання, де POBO показала здатність значно знижувати витрати на інфраструктуру при збереженні високого рівня продуктивності та безпеки.

Автори публікації [5] досліджують застосування глибокого навчання з підкріпленням для планування застосунків в обмежених ресурсами багатокористувацьких безсерверних обчислювальних середовищах. Вони розробляють моделі та алгоритми, що дозволяють ефективно управляти ресурсами при високому рівні завантаження та конкуренції між задачами, покращуючи загальну продуктивність та оптимальність розподілу ресурсів.

У статті [6] автори зосереджуються на розробці стабільних та ефективних методів управління ресурсами в хмарних середовищах за допомогою глибоких нейронних мереж. Вони розглядають виклики, пов'язані з динамічним розподілом ресурсів, та пропонують модель глибокого навчання, яка здатна адаптуватися до мінливих вимог і умов обчислювального середовища. Модель призначена для оптимізації використання ресурсів та підвищення загальної продуктивності хмарних систем, забезпечуючи при цьому стабільність та надійність сервісів.

Дослідження [7] зосереджується на аналізі сучасних методів планування контейнерів, які є ключовим компонентом в сервісах хмарних обчислень. Основна увага в роботі приділяється класифікації технік планування на чотири категорії залежно від використовуваного алгоритму оптимізації: математичне моделювання, евристичні методи, метаевристичні методи та машинне навчання. Автори детально розглядають ключові переваги та недоліки кожної категорії, виокремлюють основні виклики сучасних технік та визначають потенційні можливості для майбутніх досліджень у цій області.

Автори в дослідженні [8] пропонують систему планування, названу DRS (Deep Reinforcement Learning Scheduler), яка використовує методи глибокого навчання для оптимізації розподілу мікросервісів у кластерах Kubernetes. Спочатку задача планування в Kubernetes розглядається як процес прийняття рішень в умовах Маркова, де чітко визначені стани, дії та винагороди. Це дозволяє системі DRS автоматично вчитися та адаптуватися до змін у навантаженні та доступності ресурсів без необхідності залучення експертних знань про робоче навантаження або стан кластера. Експериментальні результати, отримані на прототипі системи в п'ятивузловому кластері Kubernetes, показують, що DRS значно покращує використання ресурсів та знижує дисбаланс навантаження порівняно з традиційним планувальником Kubernetes, з мінімальними накладними витратами на обчислення та затримку у комунікаціях.

2. Застосування планування та масштабування

Цей метод спрямований на застосування підходів планування контейнеризованих додатків в кластері та масштабування їх за визначеними ознаками. Наступні огляди стосуються досліджень, що застосовують даний метод.

Автори зосереджуються на розробці та оцінці інструмента оркестрації контейнерів, названого ge-kube, який розширює можливості Kubernetes для географічно розподілених обчислювальних середовищ, що було описано в роботі [9]. Основна мета цієї роботи полягає у покращенні розгортання контейнерів з врахуванням затримок у мережі, що є критичним аспектом для додатків з чутливістю до затримок. Для досягнення цієї мети автори розробили двокроковий контрольний цикл з використанням машинного навчання для динамічного контролю кількості реплік контейнерів залежно від часу відгуку застосунків. Вони також впровадили мережево-орієнтовану політику розміщення, яка враховує затримки між обчислювальними ресурсами, щоб задовольнити вимоги до якості обслуговування. Автори показали, що їхня система може значно покращити час відновлення застосунків, а також ефективність розміщення контейнерів у гео-розподілених середовищах.

Автори статті [10] зосереджуються на поліпшенні розподілу ресурсів на безсерверних обчисленнях, зокрема у Knative. Вони розглядають обмеження традиційних методів горизонтального масштабування, які масштабують лише екземпляри сервісів, не корегуючи ресурси на кожен екземпляр. Для підвищення ефективності автори пропонують гібридний підхід до масштабування, який поєднує горизонтальне та

вертикальне масштабування. Стаття підкреслює необхідність динамічного управління ресурсами, яке може регулювати кількість екземплярів та ресурси, призначені для кожного з них, на основі прогнозування трафіку в реальному часі. Цей підхід допомагає оптимізувати використання ресурсів і продуктивність сервісів, особливо у середовищах з кількома одночасними сервісами, де розподіл ресурсів може стати складним. Для підтримки своїх рішень, автори розробили оператори Kubernetes і користувацькі ресурси, які можуть асистувати у гібридному масштабуванні на основі прогнозування трафіку.

У роботі [11] автори зосередились на оптимізації методів планування контейнерів для ефективного управління завданнями з великим обсягом даних у безсерверному обчислювальному середовищі. Вони досліджують використання контейнерів для розв'язання завдань з обробки великих обсягів даних, які виникають у розподілених системах обчислень. Автори пропонують оптимізовані алгоритми планування контейнерів, спрямовані на забезпечення ефективного використання ресурсів та покращення продуктивності в умовах обмежених обчислювальних ресурсів.

У статті [12] описано розроблений авторами метод автомасштабування контейнеризованих хмарних застосунків, орієнтований на робоче навантаження. Автори зосередилися на створенні алгоритму, який динамічно адаптує кількість ресурсів, необхідних застосункам у хмарному середовищі, залежно від їхнього поточного навантаження. Цей підхід спрямований на підвищення ефективності використання ресурсів і зниження витрат, гарантуючи при цьому стабільність і продуктивність застосунків під час коливань навантаження.

Стаття [13] розглядає проблему одночасного розподілу контейнерів у гетерогенних кластерах з багаторесурсними обмеженнями. Автори пропонують рішення з використанням розширеного планувальника контейнерів (ECSched), яке формулює проблему планування контейнерів як задачу мінімізації витрат у потоках даних. Це дозволяє ефективніше розподіляти ресурси, забезпечуючи більшу продуктивність та ефективність використання ресурсів у кластерах. ECSched використовує спеціально розроблену графову структуру для представлення вимог контейнерів і динамічно конструює мережу потоків на основі пакетів одночасних запитів. Застосування алгоритму мінімальних витрат потоку до цієї мережі дозволяє планувальнику ефективно обробляти одночасні запити в реальному часі. В рамках експериментів, проведених на різних тестових кластерах, ECSched демонструє вищу продуктивність порівняно з сучасними системами планування контейнерів, пропонуючи мінімальні затримки в плануванні та підвищену ефективність використання ресурсів у великих масштабах

У статті [14] автори досліджують ефективність планування контейнерних кластерів у гетерогенних розумних середовищах. Вони аналізують методи планування, спрямовані на одночасну роботу з контейнерами на різних пристроях та платформах, щоб забезпечити оптимальне використання ресурсів та забезпечити вимоги додатків до продуктивності та ефективності. Автори звертають увагу на проблеми, пов'язані з гетерогенністю середовищ та різними вимогами додатків до ресурсів. Вони розглядають підходи до оптимального розподілу контейнерів, які враховують цю різноманітність, з метою забезпечення ефективного використання ресурсів у розумних середовищах. В результаті дослідження вони розробляють методи та алгоритми, які дозволяють планувати роботу контейнерів у таких умовах з високою ефективністю та оптимальністю.

Робота [15] розглядає ключові фактори, що впливають на автоматичне масштабування в Kubernetes для додатків, які використовують інтенсивні обчислення у хмарному середовищі. Автори детально аналізують, як різні фактори можуть впливати на продуктивність методів автоматичного масштабування під час різних умов навантаження, таких як непередбачувані та передбачувані навантаження. Основний акцент у дослідженні зроблено на розробці набору ключових факторів, які мають бути враховані при розробці методів автоматичного масштабування, зокрема через серію експериментів, які демонструють, як ці фактори впливають на продуктивність методів масштабування у різних умовах. Вони також розглядають роль Kubernetes як інструменту оркестрації, який використовується у сучасних інженерних дисциплінах для автоматизації масштабування мікросервісів у хмарних обчисленнях.

Робота [16] зосереджується на розробці системи моніторингу та управління для кластерів контейнерів, яка здатна автономно прогнозувати зміни навантаження та дефіцит ресурсів. Система використовує спеціалізовані метрики для прогнозування піків споживання ресурсів та їх проактивного розподілу, а також звільняє ресурси при низькому попиті, підвищуючи таким чином ефективність системи. Дослідження показало, що розроблена система покращує політику ескалації та час відгуку, покращуючи якість обслуговування (QoS/QoE) у середовищі високої гнучкості та динамічних топологій.

Робота [17] зосереджена на оптимізації розміщення мікросервісів у середовищах, що використовують Kubernetes. Автори розробили метод, який враховує динамічні зміни в доступності ресурсів і конкуренцію між мікросервісами. Особливу увагу приділено проблемі спільних залежностей, які можуть виникати під час розміщення. Цей підхід спрямований на покращення ефективності і масштабованості розгортання мікросервісів у хмарних обчисленнях.

Дослідження [18] пропонує нові алгоритми планування для застосування у Kubernetes, які базуються на методі пошуку за допомогою "вусів жука". Ці алгоритми спрямовані на підвищення ефективності використання ресурсів при розміщенні контейнеризованих застосунків, зокрема для зменшення витрат на мережеве спілкування між компонентами системи, що часто змінюються. Алгоритми

орієнтовані на мінімізацію загальних витрат і оптимізацію розподілу завдань за критеріями вартості та ресурсів

3. Архітектурні рішення

В даному методі застосовуються нетипові варіанти вирішення проблеми оптимізації ресурсів шляхом зміни стандартних низькорівневих рішень Kubernetes кластера. Наступні огляди охоплюють праці, що обрали цей напрям як фундаментальний для вирішення проблеми оптимізації використання ресурсів.

Робота [19] зосереджується на підвищенні стійкості Kubernetes кластерів до відмов, зокрема на витривалості перед візантійськими помилками (задача візантійських генералів), які можуть виникати внаслідок випадкових помилок або зловмисних атак. Автори пропонують платформу Kubernetes multi-Master Robust (KmMR), яка використовує протокол відтворення стійкості до помилок BFT-SMaRt, здатний протистояти як візантійським, так і іншим помилкам. Автори демонструють, що KmMR може забезпечити продовження роботи сервісів навіть при перевищенні загального числа допустимих помилок, забезпечуючи при цьому значно коротший час консенсусу в порівнянні з традиційними платформами, що використовують протокол Raft. Крім того, зазначається, що додаткові витрати на ресурси виявилися незначними.

У роботі [20] автори фокусуються на використанні реплікації кінцевого автомата (State Machine Replication, SMR) у контейнерах, які керуються за допомогою Kubernetes. Автори досліджують інтеграцію протоколу консенсусу Raft у контейнерах, що управляються Kubernetes, для досягнення SMR. Вони порівнюють продуктивність та споживання ресурсів між KRaft (адаптацією Raft для Kubernetes) та традиційним Raft, який використовується на фізичних машинах. Експерименти показали, що KRaft забезпечує більшу ефективність і менше споживання ресурсів порівняно з класичним Raft. Результати підтверджують, що KRaft може ефективно використовуватися в середовищах Kubernetes для забезпечення стійкості реплікацій без значного збільшення витрат на ресурси. Такий підхід відкриває шлях для більш надійних розгортань у вимогливих середовищах Kubernetes.

Автори роботи [21] описують нові методи для управління спільними ресурсами у Kubernetes, щоб підтримати контейнери реального часу для хмарних обчислень. Автори вносять зміни до оркестрації спільних ресурсів, таких як пропускна здатність пам'яті, кеш і спільні інтерфейси, що дозволяє Kubernetes ефективніше управляти ресурсами для додатків реального часу. Вони запропонували систему, що включає докладний моніторинг та розподіл низькорівневих спільних ресурсів на вузлах для кращої ізоляції контейнерів, а також підтримку динамічної оркестрації та балансування контейнерів на основі доступності та потреб у спільних ресурсах. Це дозволяє Kubernetes більш ефективно використовувати хмарні ресурси та підтримувати додатки, які вимагають високої швидкості обробки даних та низької затримки.

Робота [22] аналізує теоретичні та практичні аспекти автоматизації ресурсів Kubernetes за допомогою веб-рішення KubeGen. Автори пропонують графічний інтерфейс користувача, генерацію ресурсів на основі шаблонів та вбудовану валідацію. KubeGen покликаний спростити створення ресурсів Kubernetes, забезпечуючи відповідність найкращим практикам та покращуючи управління ресурсами, що, сприяє більш ефективному і орієнтованому на користувача розгортанню сучасних застосунків.

Робота [23] розглядає створення оператора Kubernetes, який слугує мостом між хмарними обчисленнями та зовнішніми ресурсами, такими як HPC кластери, квантові сервіси чи системи Ray. Цей оператор дозволяє керувати складними робочими процесами, розгортаючи окремі етапи обчислень на зовнішніх системах з управлінням ресурсами, що значно спрощує інтеграцію та взаємодію між різними обчислювальними середовищами. Основна концепція полягає у використанні Kubernetes для динамічного подання та моніторингу завдань на зовнішніх системах, забезпечуючи більш високу гнучкість та ефективність управління ресурсами. Оператор реалізує підтримку зовнішніх систем за допомогою API, що дозволяє надсилати запити, керувати завданнями та отримувати результати, використовуючи стандартні інструменти Kubernetes. Ця розробка є значним кроком у напрямку забезпечення більшої інтеграції між хмарними обчисленнями та спеціалізованими обчислювальними ресурсами, що відкриває нові можливості для оптимізації обчислень і збільшення продуктивності в наукових дослідженнях та промисловому застосуванні.

4. Критерії оцінки методів та засобів

Результати порівняльного аналізу [24] дозволяють виділити чотири характеристики, які можуть слугувати критеріями оцінки якості описаних методів, натомість в цій статті будуть взяті до уваги лише три з них (табл. 1), адже відсутність додаткового компоненту може не так сильно впливати на якість методу, як інші три:

- універсальність – метод можна застосувати в Kubernetes кластерах, що мають різні налаштування та різне розподілення додатків, таким чином не буде існувати залежності або обмеження щодо використання;
- не потребує глибокого налаштування – якщо метод потребує глибокого налаштування для аналізу або безпосередньо оптимізації ресурсів, це вказує на необхідність додаткової експертизи, що тягне за собою додаткові витрати;
- наявність аналітичних інструментів – дозволяє точніше визначати місця для оптимізації, обрахунку використання ресурсів та аналізу змін після застосування покращень.

Таблиця 1

Порівняння існуючих методів оптимізації використання ресурсів

Метод	Критерії оцінки якості /досконалості методу		
	Універсальність	**Глибина налаштування**	**Наявність аналітичних інструментів**
Моделі машинного навчання	–	+	+/–
Планування та масштабування	–	+	–
Архітектурні рішення	+/–	–	–

+/– - залежить від реалізації

Висновки

На сьогоднішній день не існує єдиного методу, який би відповідав усім визначеним характеристикам. Однак, комбінація різних методів або використання їхніх окремих компонентів може підвищити кількість позитивних, критично важливих властивостей у кінцевому рекомендованому методі чи засобі. Важливим критерієм є наявність аналітичних інструментів, що в результаті дозволяють краще зрозуміти місця для впровадження покращень, проведення аналізу використання ресурсів та порівняльного аналізу після застосування методів для оцінки якості оптимізації. Популярність Kubernetes підкреслює важливість проведення ґрунтовних досліджень у цьому напрямку.

Література

1. Gianluca Turin Predicting resource consumption of Kubernetes container systems using resource models. Journal of systems and software. 2023. P. 111750. https://doi.org/10.1016/j.jss.2023.111750

2. Víctor Medel Characterising resource management performance in Kubernetes. Computers & electrical engineering. 2018. T. 68. P. 286–297. https://doi.org/10.1016/j.compeleceng.2018.03.041

3. Ivanovic M., Visnja Simic Efficient evolutionary optimization using predictive auto-scaling in containerized environment. Applied soft computing. 2022. P. 109610. https://doi.org/10.1016/j.asoc.2022.109610

4. Hengquan Guo POBO: safe and optimal resource management for cloud microservices. Performance evaluation. 2023. T. 162. P. 102376. https://doi.org/10.1016/j.peva.2023.102376

5. Mampage A., Shanika Karunasekera, Rajkumar Buyya Deep reinforcement learning for application scheduling in resource-constrained, multi-tenant serverless computing environments. Future generation computer systems. 2023. https://doi.org/10.1016/j.future.2023.02.006

6. Byeonghui Jeong Stable and efficient resource management using deep neural network on cloud computing. Neurocomputing. 2023. T. 521. P. 99–112. https://doi.org/10.1016/j.neucom.2022.11.089

7. Imtiaz Ahmad Container scheduling techniques: A Survey and assessment. Journal of king saud university - computer and information sciences. 2021. https://doi.org/10.1016/j.jksuci.2021.03.002

8. Zhaolong Jian DRS: A deep reinforcement learning enhanced Kubernetes scheduler for microservice-based system. Software: practice and experience. 2023. https://doi.org/10.1002/spe.3284

9. Fabiana Rossi Geo-distributed efficient deployment of containers with Kubernetes. Computer communications. 2020. T. 159. 161–174. https://doi.org/10.1016/j.comcom.2020.04.061

10. Tran M.-N., YoungHan Kim Optimized resource usage with hybrid auto-scaling system for knative serverless edge computing. Future generation computer systems. 2023. https://doi.org/10.1016/j.future.2023.11.010

11. Rausch T., Alexander Rashed, Schahram Dustdar Optimized container scheduling for data-intensive serverless edge computing. Future generation computer systems. 2021. T. 114. 259–271. https://doi.org/10.1016/j.future.2020.07.017

12. Chouliaras S., Stelios Sotiriadis Auto-scaling containerized cloud applications: a workload-driven approach. Simulation modelling practice and theory. 2022. 102654. https://doi.org/10.1016/j.simpat.2022.102654

13. Yang Hu Concurrent container scheduling on heterogeneous clusters with multi-resource constraints. Future generation computer systems. 2020. T. 102. 562–573. https://doi.org/10.1016/j.future.2019.08.025

14. A. Asensio On the optimality of Concurrent Container Clusters Scheduling over heterogeneous smart environments. Future generation computer systems. 2021. T. 118. 157–169. https://doi.org/10.1016/j.future.2021.01.003

15. Taherizadeh S., Marko Grobelnik Key influencing factors of the Kubernetes auto-scaler for computing-intensive microservice-native cloud-based applications. Advances in engineering software. 2020. T. 140. 102734. https://doi.org/10.1016/j.advengsoft.2019.102734

16. Gonçalo Marques Proactive resource management for cloud of services environments. Future generation computer systems. 2023. https://doi.org/10.1016/j.future.2023.08.005

17. Ding Z., Song Wang, Changjun Jiang Kubernetes-Oriented microservice placement with dynamic

resource allocation. IEEE transactions on cloud computing. 2022. 1. https://doi.org/10.1109/tcc.2022.3161900

18. Hongjian Li Cost-efficient scheduling algorithms based on beetle antennae search for containerized applications in Kubernetes clouds. The journal of supercomputing. 2023. https://doi.org/10.1007/s11227-023-05077-7

19. Diouf G. M., Halima Elbiaze, Wael Jaafar On Byzantine fault tolerance in multi-master Kubernetes clusters. Future generation computer systems. – 2020. T. 109. 407–419. https://doi.org/10.1016/j.future.2020.03.060.

20. Hylson V. Netto State machine replication in containers managed by Kubernetes. Journal of systems architecture. 2017. T. 73. P. 53–59. https://doi.org/10.1016/j.sysarc.2016.12.007

21. CRACIUN P.-C., Cristian Robert NECULA Theoretical and Applied in Automating Kubernetes Resources. Informatica Economica. 2023. T. 27, № 2/2023. P. 36–45. https://doi.org/10.24818/issn14531305/27.2.2023.04

22. Lublinsky B., Elise Jennings, Viktória Spišaková A Kubernetes 'Bridge' Operator between Cloud and External Resources. 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 26–28.04. 2023. [Б. м.], 2023. https://doi.org/10.1109/icccbda56900.2023.10154770

23. Monaco G., Gautam Gala, Gerhard Fohler Shared Resource Orchestration Extensions for Kubernetes to Support Real-Time Cloud Containers. 2023 IEEE 26th International Symposium on Real-Time Distributed Computing (ISORC), Nashville, TN, USA, 23–25 трав. 2023 р. [Б. м.], 2023. https://doi.org/10.1109/isorc58943.2023.00022

24. Buriak M. Methods and means of optimizing the use of resources in the Kubernetes cluster. Olviy Forum 2024: technical theses. sciences and engineering, Mykolaiv, June 20–23. 2024 - [B. m.], 2024. P. 141–144.