Наукові перспективи
Видавнича група

№ 13 (54)
2025

НАУКА і ТЕХНІКА СЬОГОДНІ

серія: право, серія: економіка, серія: педагогіка,
серія: техніка, серія: фізико-математичні науки

Шановні колеги!

З Новим Роком та Різдвом Христовим!

Бажаю, щоб Ви і Ваші близькі були здорові і щасливі, щоб удача супроводжувала у справах, щоб любов оточувала і наповнювала Вас і Вашу родину.

Нехай негоди проходять стороною, а над головою буде завжди мирне небо і ясне сонце. Виконаних мрій, досягнутих цілей і приємних відкриттів Вам у Новому 2026 році!

З повагою,
директор Видавничої групи
«Наукові перспективи»
Ірина Жукова

Видавнича група «Наукові перспективи»

Всеукраїнська Асамблея докторів наук із державного управління

Асоціація науковців України

«Наука і техніка сьогодні»

№ 13(54) 2025

Київ – 2025

ISSN 2786-6025 Online
УДК 001.32:1 /3](477)(02)

DOI: https://doi.org/10.52058/2786-6025-2025-13(54)

«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка», Серія «Фізико-математичні науки», Серія «Техніка»)»: журнал. 2025. № 13(54) 2025. С. 2763

Згідно наказу Міністерства освіти і науки України від 07.04.2022 № 320 журналу присвоєно категорію "Б" із економіки та педагогіки (спеціальності – 015 - Педагогічні науки; 076 - Економічні науки)

Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)

Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу присвоєно категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)

Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з духовно-морального виховання»

Рекомендовано до видавництва Президією Всеукраїнської Асамблеї докторів наук з державного управління (Рішення від 24.12.2025 № 9/12-25)

Журнал включено до міжнародної наукометричної бази Index Copernicus (IC), міжнародної пошукової системи Google Scholar та до міжнародної наукометричної бази даних Research Bible

УДК 004.7:004.9

https://doi.org/10.52058/2786-6025-2025-13(54)-2340-2353

Муржа Дмитро Юрійович аспірант, викладач кафедри кібербезпеки та інформаційних технологій Харківського національного економічного університету імені Семена Кузнеця, https://orcid.org/0009-0002-4214-0848

АНАЛІЗ МЕТОДІВ ОПТИМІЗАЦІЇ РЕСУРСІВ МІКРОСЕРВІСНИХ ДОДАТКІВ У ХМАРНИХ СЕРЕДОВИЩАХ: СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ

Анотація. Мікросервісна архітектура стала домінуючим підходом до розробки хмарних додатків завдяки своїй гнучкості, масштабованості та можливості незалежного розгортання компонентів. За даними звіту Dynatrace 2023 року, використання Kubernetes у виробничих середовищах продовжує зростати, що підкреслює актуальність автоматизації управління ресурсами. Однак динамічна природа навантаження мікросервісів створює значні виклики для ефективного управління ресурсами, оскільки традиційні підходи до виділення ресурсів не можуть адекватно адаптуватися до змінних умов експлуатації.

У статті проведено систематичний аналіз наукової літератури за період 2021–2025 років, що охоплює 17 досліджень в галузі оптимізації ресурсів мікросервісних додатків.

Встановлено чотири категорії методів оптимізації: статичні методи виділення ресурсів, динамічні методи на основі правил, методи на основі машинного навчання та гібридні підходи.

Проаналізовано переваги та недоліки кожної категорії за критеріями адаптивності, складності реалізації, накладних витрат, точності прогнозування та швидкості реакції на зміни навантаження.

Виявлено, що методи на основі машинного навчання дозволяють підвищити утилізацію ресурсів на 30–35% порівняно з традиційними threshold-based підходами, досягаючи показників утилізації 80–90%.

Експериментальні результати показують, що двонапрямлені рекурентні нейронні мережі забезпечують прогнозування навантаження у 530 разів швидше порівняно з традиційними статистичними моделями при збереженні високої точності. Гібридні підходи, що поєднують проактивне прогнозування на основі машинного навчання з реактивними механізмами, демонструють найкращі результати, зменшуючи порушення угод про рівень обслуговування на 36%.

Результати дослідження можуть бути використані для вибору оптимального методу управління ресурсами залежно від характеристик додатку, вимог до якості обслуговування та доступних обчислювальних ресурсів. Перспективним напрямком подальших досліджень є розробка адаптивних гібридних методів, здатних автоматично обирати оптимальну стратегію виділення ресурсів на основі поточних характеристик навантаження та стану системи.

Ключові слова: мікросервіси, оптимізація ресурсів, хмарні обчислення, контейнеризація, Kubernetes, автомасштабування, машинне навчання.

Murzha Dmytro Yuriiovych PhD student, Lecturer at the Department of Cybersecurity and Information Technologies, Simon Kuznets Kharkiv National University of Economics, https://orcid.org/0009-0002-4214-0848

ANALYSIS OF RESOURCE OPTIMIZATION METHODS FOR MICROSERVICE APPLICATIONS IN CLOUD ENVIRONMENTS: CURRENT STATE AND PROSPECTS

Abstract. Microservice architecture has become the dominant approach to cloud application development due to its flexibility, scalability, and ability to independently deploy components. According to the Dynatrace 2023 report, Kubernetes usage in production environments continues to grow, emphasizing the relevance of resource management automation. However, the dynamic nature of microservice workloads creates significant challenges for efficient resource management, as traditional resource allocation approaches cannot adequately adapt to changing operating conditions.

The article presents a systematic analysis of scientific literature for the period 2021–2025, covering 17 studies in the field of microservice application resource optimization. Four categories of optimization methods have been identified: static resource allocation methods, rule-based dynamic methods, machine learning methods, and hybrid approaches. The advantages and disadvantages of each category are analyzed according to the criteria of adaptability, implementation complexity, overhead costs, prediction accuracy, and response speed to workload changes. It was found that machine learning methods allow increasing resource utilization by 30–35% compared to traditional threshold-based approaches, achieving utilization rates of 80–90%. Experimental results show that bidirectional recurrent neural networks provide workload forecasting 530 times faster compared to traditional statistical models while maintaining high accuracy. Hybrid approaches that combine proactive machine learning prediction with reactive mechanisms demonstrate the best results, reducing service level agreement violations by 36%.

The research results can be used to select the optimal resource management method depending on application characteristics, quality of service requirements, and available computational resources. A promising direction for further research is the development of adaptive hybrid methods capable of automatically selecting the optimal resource allocation strategy based on current workload characteristics and system state.

Keywords: microservices, resource optimization, cloud computing, containerization, Kubernetes, autoscaling, machine learning.

Постановка проблеми. Мікросервісна архітектура набула широкого розповсюдження як провідний підхід до побудови сучасних хмарних додатків, замінюючи монолітні системи завдяки кращій масштабованості, гнучкості розробки та можливості незалежного розгортання окремих компонентів. Контейнеризація, зокрема платформа Kubernetes, стала стандартом де-факто для оркестрації мікросервісів, забезпечуючи автоматизоване управління життєвим циклом додатків у розподілених середовищах.

Однак динамічна природа навантаження мікросервісних додатків створює виклики для ефективного управління ресурсами. На відміну від традиційних монолітних систем із відносно стабільними шаблонами використання ресурсів, мікросервіси характеризуються високою варіативністю навантаження, складними взаємозалежностями між компонентами та непередбачуваними сплесками трафіку. Недостатнє виділення ресурсів призводить до деградації продуктивності та порушення угод про рівень обслуговування, тоді як надмірне виділення спричиняє неефективне використання обчислювальних потужностей та невиправдане збільшення операційних витрат.

Проблема оптимізації ресурсів у мікросервісних системах ускладнюється необхідністю одночасного врахування множини конфліктуючих цілей: мінімізація витрат на інфраструктуру, забезпечення високої доступності сервісів, дотримання вимог до часу відгуку та ефективне використання обчислювальних ресурсів. Традиційні підходи до планування потужностей, засновані на статистичному аналізі історичних даних або експертних оцінках, виявляються недостатньо ефективними в умовах швидкозмінних шаблонів навантаження, характерних для сучасних хмарних додатків.

Актуальність дослідження методів оптимізації ресурсів підтверджується зростаючими обсягами використання хмарних сервісів та контейнерних технологій у виробничих середовищах. Систематизація існуючих підходів, виявлення їх переваг та недоліків, а також визначення перспективних напрямків розвитку є важливим завданням для створення ефективних інформаційних технологій управління ресурсами мікросервісних додатків у хмарних середовищах.

Аналіз останніх досліджень і публікацій. Аналіз наукової літератури показує інтерес дослідницької спільноти до проблеми оптимізації ресурсів мікросервісних додатків. Найновіший систематичний огляд [1], опублікований у 2025 році в журналі Computer Science Review, охоплює понад сто публікацій за період 2016–2025 років та представляє комплексну таксономію методів автомасштабування в хмарних середовищах. Огляд [2], опублікований у високорейтинговому журналі ACM Computing Surveys з імпакт-фактором 16, систематизує алгоритми планування Kubernetes за параметрами якості обслуговування.

Комплексний огляд [3], представлений у Web Intelligence, класифікує методи виділення ресурсів контейнерів на основі аналізу 64 досліджень за період 2012–2020 років. Автори виділяють основні категорії підходів: статичні методи планування потужностей, динамічні методи на основі порогових значень метрик, методи прогнозування навантаження та гібридні підходи. Дослідження [4], опубліковане в Concurrency and Computation: Practice and Experience, фокусується на специфіці еластичності мікросервісів порівняно з традиційною еластичністю віртуальних машин, підкреслюючи необхідність врахування міжсервісних залежностей.

Ряд досліджень [5, 14, 15] присвячений методам на основі оркестрації контейнерів. У роботі [5] представлено систематичний огляд алгоритмів планування Kubernetes, що включає аналіз метаеврістичних підходів, таких як алгоритми мурашиної колонії та рою частинок, а також методи багатокритеріальної оптимізації. Дослідження показують, що метаеврістичні та ML-підходи перевершують жадібні алгоритми за критеріями балансування навантаження та ефективності використання ресурсів.

Методи на основі машинного навчання представлені в роботах [6–8]. Дослідження [6] демонструє застосування двонапрямлених рекурентних нейронних мереж для прогнозування навантаження в Kubernetes, досягаючи високої точності при значно меншому часі обчислень порівняно з традиційними статистичними моделями. Робота [7] представляє підхід на основі графових нейронних мереж для цілісного автомасштабування мікросервісів з урахуванням каскадних ефектів між компонентами. У дослідженні [8] запропоновано поєднання рекурентних мереж довгої короткочасної пам’яті з глибоким Q-навчанням для інтелектуального виділення ресурсів.

Підходи на основі навчання з підкріпленням розглянуто в роботах [9–10]. Дослідження [9] представляє фреймворк Gwydion для ефективного автомасштабування складних контейнеризованих додатків через навчання з підкріпленням, демонструючи можливість балансування між латентністю та вартістю. Робота [10] адресує розрив між RL-дослідженнями та виробничими системами через підхід мета-навчання з підкріпленням.

Гібридні підходи, що поєднують переваги різних методів, представлені в роботах [11–13]. Дослідження [11] демонструє колаборативну інтеграцію проактивних ML-моделей з реактивними механізмами через оптимізатор на основі модельно-прогнозуючого керування, досягаючи найкращих результатів у виробничому середовищі. У роботі [12] запропоновано адаптивне переключення між проактивним та реактивним режимами на основі довіри до прогнозу. Дослідження [13] представляє adaptive framework, що поєднує кластеризацію K-means для аналізу шаблонів навантаження з CNN для прогнозування використання процесора.

Незважаючи на прогрес у цій галузі, існуючі підходи мають певні обмеження. Статичні методи не враховують динаміку навантаження та неефективні при змінних шаблонах використання. Методи на основі правил потребують ретельного налаштування порогових значень та можуть призводити до нестабільного переключення між станами масштабування. ML-підходи вимагають значних обсягів даних для навчання та можуть бути чутливими до зміни характеру навантаження. Відсутність універсального рішення, ефективного для різних типів додатків та шаблонів навантаження, обумовлює необхідність подальших досліджень у напрямку створення адаптивних гібридних методів.

Мета дослідження – систематичний аналіз існуючих методів оптимізації ресурсів мікросервісних додатків у хмарних середовищах, виявлення їх переваг, недоліків та перспективних напрямків розвитку.

Для досягнення поставленої мети визначено наступні завдання:

1) класифікувати існуючі підходи до оптимізації ресурсів мікросервісних додатків;

2) проаналізувати переваги та недоліки кожного класу методів за ключовими критеріями;

3) порівняти методи за показниками ефективності використання ресурсів, швидкості реакції та складності реалізації;

4) визначити перспективні напрямки подальших досліджень у галузі оптимізації ресурсів мікросервісів.

Виклад основного матеріалу.

Класифікація методів оптимізації ресурсів. На основі аналізу літератури виділено чотири основні категорії методів оптимізації ресурсів мікросервісних додатків (рисунок 1): статичні методи виділення ресурсів, динамічні методи на основі правил, методи на основі машинного навчання та гібридні підходи. Кожна категорія характеризується специфічними механізмами прийняття рішень, рівнем адаптивності до змін навантаження та складністю реалізації.

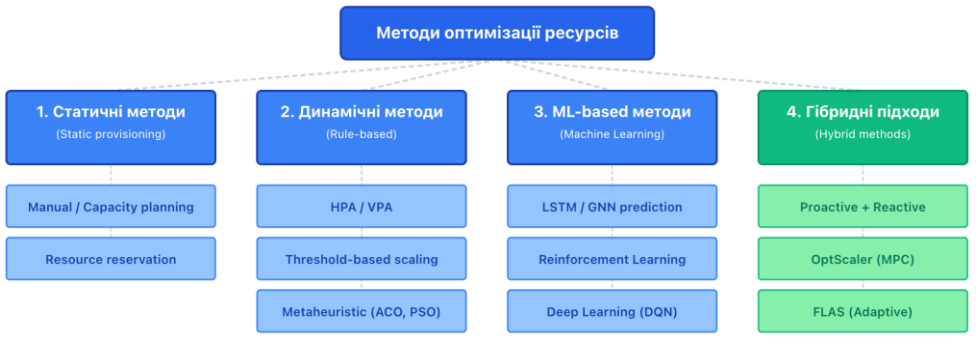

Рис. 1. Класифікація методів оптимізації ресурсів мікросервісних додатків

Як показано на рисунку 1, статичні методи забезпечують найпростішу реалізацію, але мають низьку адаптивність. Динамічні методи на основі правил реагують на поточні метрики, але потребують ретельного налаштування. Методи машинного навчання дозволяють прогнозувати навантаження та приймати проактивні рішення, однак вимагають значних обчислювальних ресурсів. Гібридні підходи поєднують переваги різних категорій, забезпечуючи найвищу ефективність.

Статичні методи виділення ресурсів передбачають попереднє визначення обсягу ресурсів на основі аналізу історичних даних, результатів навантажувального тестування або експертних оцінок. До основних підходів відносяться ручне налаштування конфігурації, планування потужностей на основі очікуваного пікового навантаження та резервування ресурсів з фіксованим запасом.

Основною перевагою статичних методів є простота реалізації, передбачуваність поведінки системи та мінімальні накладні витрати на управління, оскільки не потрібно постійно моніторити метрики та приймати рішення про масштабування. Однак ці методи є неефективними при змінному навантаженні, оскільки не можуть адаптуватися до фактичних потреб додатку. Це призводить або до надмірного виділення ресурсів із супутнім збільшенням витрат, або до недостатнього виділення з деградацією продуктивності при сплесках навантаження.

Динамічні методи на основі правил реалізують автоматичне масштабування шляхом порівняння поточних значень метрик з попередньо визначеними пороговими значеннями. Найпоширенішим рішенням є механізм горизонтального автомасштабування подів Kubernetes, який коригує кількість реплік сервісу на основі використання процесора, пам'яті або користувацьких метрик.

Дослідження [14] представляє самоадаптивний алгоритм автомасштабування для застосунків, чутливих до угод про рівень обслуговування, який динамічно обирає порогові значення використання процесора та інтервали очікування. Експериментальні результати показують відхилення від цільового

рівня обслуговування в межах 1–2% при резервуванні ресурсів 4–10%. У роботі [15] запропоновано налаштування горизонтального автомасштабувача подів Kubernetes на основі принципу максимальної ентропії, що дозволяє досягти скорочення кількості вузлів приблизно на 15% при збереженні медіанного часу відгуку менше 2 мс та рівня помилок менше 5%.

Перевагами динамічних методів на основі правил є швидка реакція на зміни навантаження, відносна простота реалізації та низькі обчислювальні витрати. Основними недоліками є складність налаштування порогових значень для різних типів додатків, ризик нестабільності через часте переключення між станами масштабування та реактивний характер, що не дозволяє заздалегідь підготуватися до прогнозованих змін навантаження.

Методи на основі машинного навчання використовують історичні дані для навчання моделей, здатних прогнозувати майбутнє навантаження та приймати проактивні рішення щодо виділення ресурсів. Основними підходами є прогнозування навантаження за допомогою рекурентних нейронних мереж та навчання з підкріпленням для оптимізації політик управління ресурсами.

Дослідження [6] демонструє застосування двонапрямлених рекурентних нейронних мереж довгої короткочасної пам'яті для прогнозування навантаження в Kubernetes. Експериментальні результати, представлені в таблиці 1, показують, що модель досягає коефіцієнта детермінації 0.983 при часі прогнозування 5.48 мс для одного кроку, що в 530 разів швидше порівняно з традиційною моделлю авторегресійного інтегрованого ковзного середнього.

Таблиця 1

Порівняння продуктивності моделей прогнозування навантаження

Модель	RMSE	R^2	Швидкість (мс)	Прискорення
Bi-LSTM	36,28	0,983	5,48	530×
ARIMA	37,48	0,982	2903,7	1×

Робота [7] представляє підхід DeepScaler, що використовує графові згорткові нейронні мережі з адаптивним навчанням графів для моделювання просторово-часових залежностей між мікросервісами. Експериментальні результати показують зменшення порушень угод про рівень обслуговування на 41% порівняно з базовими підходами. Дослідження [8] демонструє поєднання рекурентних нейронних мереж для прогнозування з глибоким Q-навчанням для динамічного планування, досягаючи підвищення утилізації ресурсів на 32.5% та зменшення часу відгуку на 43.3%.

Підходи на основі навчання з підкріпленням дозволяють системі навчатися оптимальній політиці виділення ресурсів через взаємодію з середовищем.

Дослідження [9] представляє фреймворк Gwydion, що використовує глибоке Q-навчання та методи градієнта політики для автомасштабування складних контейнеризованих додатків. Експериментальні результати демонструють можливість досягнення до 50% зниження латентності порівняно з типовим Kubernetes при економії витрат близько 30% через стратегії, що враховують вартість.

Основними перевагами ML-підходів є висока точність прогнозування, здатність виявляти складні нелінійні залежності в даних та проактивний характер прийняття рішень. До недоліків відносяться необхідність значних обсягів даних для навчання, чутливість до зміни характеру навантаження та відносно високі обчислювальні витрати на навчання моделей.

Гібридні підходи. Гібридні методи поєднують переваги різних підходів, зокрема проактивне прогнозування на основі машинного навчання з реактивними механізмами для обробки непередбачуваних змін навантаження. Дослідження [11] представляє фреймворк OptScaler, що реалізує колаборативну інтеграцію проактивного модуля прогнозування навантаження, реактивного модуля самоналаштування оцінки використання процесора та модуля оптимізації на основі модельно-прогнозуючого керування з урахуванням ймовірнісних обмежень.

Експериментальні результати показують, що OptScaler зменшує порушення угод про рівень обслуговування на 36% порівняно з незалежними реактивними, проактивними та гібридними системами, досягаючи рівня порушень 1.1–2.1% проти 2.8–13.8% у конкурентів. Виробниче впровадження в системі Alipay підтримує понад 100 довготривалих застосунків, підтверджуючи масштабованість підходу.

Робота [12] представляє фреймворк FLAS, що реалізує адаптивне переключення між проактивним та реактивним режимами на основі довіри до прогнозу. Система досягає рівня порушень угод про рівень обслуговування менше 1% через прогнозування високорівневих метрик якості обслуговування замість лише навантаження. Дослідження [13] пропонує адаптивний фреймворк, що поєднує кластеризацію для виявлення шаблонів навантаження з згортковими нейронними мережами для прогнозування використання процесора.

Гібридні підходи демонструють найкращі результати за критеріями утилізації ресурсів, дотримання угод про рівень обслуговування та адаптивності до різних умов експлуатації.

Основним недоліком є висока складність реалізації та налаштування взаємодії між компонентами системи.

Multi-cloud оптимізація. Розгортання мікросервісних додатків у multi-cloud середовищах створює додаткові можливості для оптимізації ресурсів

через використання переваг різних хмарних провайдерів, географічного розподілу навантаження та динамічної міграції сервісів між платформами. Основними цілями multi-cloud оптимізації є мінімізація операційних витрат, зменшення затримок за рахунок наближення сервісів до користувачів, підвищення надійності через розподіл ризиків та оптимізація екологічного впливу.

Дослідження [16] представляє підхід до мінімізації витрат у multi-cloud системах через динамічну реоркестрацію мікросервісів з використанням цілочисельного лінійного програмування.

Метод враховує вартість обчислю-вальних ресурсів у різних провайдерів, затримки мережі між регіонами та обмеження на розміщення чутливих до затримок сервісів.

Експериментальні результати показують можливість досягнення майже нульових перерв обслуговування при runtime переміщенні мікросервісів між хмарними платформами.

Підхід особливо ефективний для систем з великою кількістю географічно розподілених користувачів, де оптимальне розміщення сервісів дозволяє суттєво зменшити як витрати, так і час відгуку.

Важливим напрямком розвитку multi-cloud оптимізації є врахування екологічних факторів при прийнятті рішень про виділення ресурсів. Робота [17] пропонує фреймворк CASA для автомасштабування з урахуванням вуглецевого сліду в serverless обчисленнях. Система балансує між дотриманням угод про рівень обслуговування та мінімізацією викидів вуглецю через інтелектуальне планування холодних запусків та вибір регіонів з найчистішими джерелами енергії. Експериментальні дані демонструють зменшення вуглецевого сліду в 2.6 рази при одночасному зменшенні порушень угод про рівень обслуговування в 1.4 рази.

Підхід особливо актуальний у контексті зростаючих вимог до корпоративної екологічної відповідальності та регуляторних обмежень на викиди парникових газів.

Multi-cloud оптимізація додає додатковий рівень складності до управління ресурсами мікросервісів через необхідність враховувати гетеро-генність платформ, політики різних провайдерів, варіацію цін та доступності ресурсів у часі. Однак переваги у вигляді гнучкості, стійкості до відмов окремих провайдерів та можливості оптимізації за множиною критеріїв роблять цей напрямок перспективним для подальших досліджень та практичного впровадження.

Порівняльний аналіз методів. Порівняльний аналіз чотирьох категорій методів оптимізації ресурсів представлено в таблиці 2. Кожен підхід має специфічні характеристики, що визначають область його застосування.

Таблиця 2

Порівняльний аналіз методів оптимізації ресурсів

Критерій	**Статичні**	**Динамічні**	**ML-based**	**Гібридні**
Адаптивність	Низька	Середня	Висока	Дуже висока
Складність реалізації	Низька	Середня	Висока	Висока
Накладні витрати	Мінімальні	Низькі	Середні	Середні
Точність прогнозування	Н/Д	Низька	Висока	Дуже висока
Швидкість реакції	Н/Д	Швидка	Повільна	Швидка
Утилізація ресурсів	50–60%	65–75%	80–90%	85–95%
Вартість експлуатації	Висока	Середня	Низька	Дуже низька

Примітка: Діапазони утилізації ресурсів та характеристики методів узагальнені на основі аналізу досліджень [1–17]. Конкретні показники залежать від типу додатку та умов експлуатації.

Як видно з таблиці, статичні методи характеризуються найнижчою складністю реалізації та мінімальними накладними витратами, але забезпечують низьку адаптивність та утилізацію ресурсів на рівні 50–60%. Динамічні методи на основі правил покращують утилізацію до 65–75% завдяки швидкій реакції на зміни, але потребують ретельного налаштування порогових значень.

Методи на основі машинного навчання досягають найвищої точності прогнозування та утилізації ресурсів 80–90%, але характеризуються повільнішою реакцією на раптові зміни навантаження через необхідність часу на прогнозування. Гібридні підходи демонструють найкращий баланс між усіма критеріями, досягаючи утилізації 85–95% при збереженні швидкої реакції та дуже низької вартості експлуатації.

Вибір оптимального методу залежить від характеристик конкретного додатку. Для систем зі стабільним навантаженням можуть бути достатніми статичні методи.

Додатки з передбачуваними шаблонами навантаження отримують найбільшу користь від ML-підходів.

Системи з високою варіативністю та критичними вимогами до якості обслуговування потребують гібридних методів, здатних поєднувати проактивне та реактивне управління ресурсами.

Візуалізація порівняння показників утилізації ресурсів представлена на рисунку 2. Як видно з рисунка, спостерігається чітка тенденція зростання ефективності від статичних методів (55%) до гібридних підходів (90%), що підтверджує переваги інтеграції різних технік оптимізації для досягнення максимальної утилізації ресурсів при збереженні якості обслуговування.

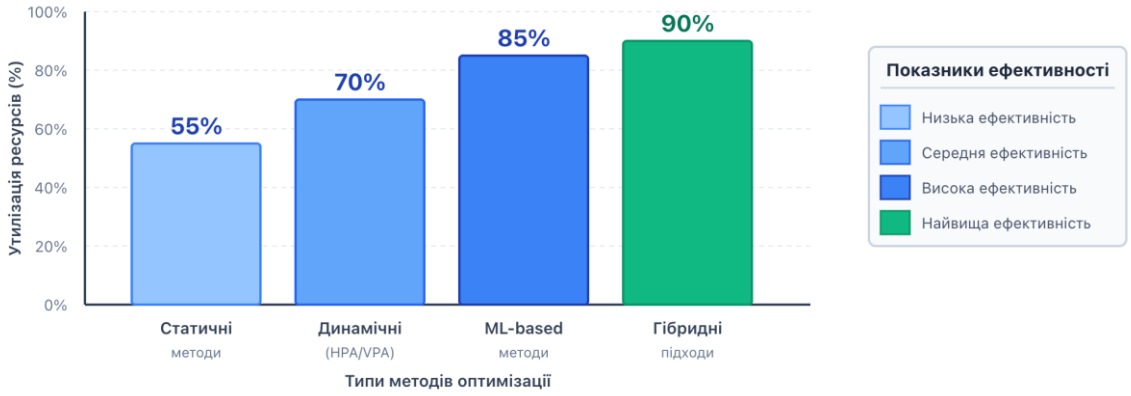

Рис. 2. Порівняння утилізації ресурсів різними методами оптимізації

Висновки. У результаті проведеного систематичного аналізу наукової літератури за період 2021–2025 років встановлено чотири основні категорії методів оптимізації ресурсів мікросервісних додатків у хмарних середовищах: статичні методи виділення ресурсів, динамічні методи на основі правил, методи на основі машинного навчання та гібридні підходи. Кожна категорія характеризується специфічними механізмами прийняття рішень, рівнем адаптивності та областю застосування.

Виявлено три ключові тенденції розвитку методів оптимізації ресурсів. По-перше, спостерігається перехід від реактивних підходів, що реагують на вже існуючі проблеми з продуктивністю, до проактивних методів, здатних передбачати зміни навантаження та заздалегідь коригувати виділення ресурсів. По-друге, зростає роль машинного навчання, зокрема глибоких нейронних мереж та навчання з підкріпленням, у прогнозуванні навантаження та оптимізації політик управління ресурсами. По-третє, найкращі результати демонструють гібридні підходи, що поєднують проактивне прогнозування з реактивними механізмами для обробки непередбачуваних ситуацій.

Встановлено, що методи на основі машинного навчання дозволяють підвищити утилізацію ресурсів на 30–35% порівняно з традиційними threshold-based підходами, досягаючи показників 80–90%. Експериментальні дані показують, що двонапрямлені рекурентні нейронні мережі забезпечують прогнозування в 530 разів швидше порівняно з традиційними статистичними моделями при збереженні високої точності. Гібридні методи з колаборативною інтеграцією компонентів перевершують незалежні гібридні системи на 36% за метрикою порушень угод про рівень обслуговування.

Результати дослідження можуть бути використані для вибору оптимального методу управління ресурсами залежно від характеристик додатку, вимог до якості обслуговування та доступних обчислювальних ресурсів. Для систем зі стабільним навантаженням достатніми можуть бути

статичні або динамічні методи. Додатки з передбачуваними шаблонами навантаження отримують найбільшу користь від ML-підходів. Системи з високою варіативністю та критичними вимогами до SLA потребують гібридних методів.

Перспективними напрямками подальших досліджень є розробка адаптивних гібридних методів, здатних автоматично обирати оптимальну стратегію виділення ресурсів на основі поточних характеристик навантаження та стану системи. Актуальним є створення методів оптимізації в умовах багатохмарних середовищ з урахуванням географічних та економічних факторів. Важливим напрямком є розвиток методів навчання з підкріпленням, здатних швидко адаптуватися до нових умов експлуатації через мета-навчання. Ці напрямки складають основу для подальших досліджень автора в галузі створення інформаційних технологій гібридної оптимізації ресурсів мікро-сервісів у хмарному середовищі.

Література:

1. Jeong B. Autoscaling techniques in cloud-native computing: A comprehensive survey / B. Jeong, Y. Jeong // Computer Science Review. – 2025. – Vol. 58. – Article 100791. – DOI: 10.1016/j.cosrev.2025.100791

2. Carrión M.C. Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges / M.C. Carrión // ACM Computing Surveys. – 2022. – Vol. 55, № 7. – P. 1-37. – DOI: 10.1145/3539606

3. Vhatkar K.N. A comprehensive survey on container resource allocation approaches in cloud computing: State-of-the-art and research challenges / K.N. Vhatkar, G.P. Bhole // Web Intelligence. – 2021. – Vol. 19, № 3. – P. 181-204. – DOI: 10.3233/WEB-210474

4. Fourati M.H. Cloud Elasticity of Microservices-Based Applications: A Survey / M.H. Fourati, S. Marzouk, M. Jmaiel // Concurrency and Computation: Practice and Experience. – 2024. – Vol. 37, № 2. – DOI: 10.1002/cpe.8329

5. Hameed A. A Survey of Kubernetes Scheduling Algorithms / A. Hameed, A. Yousif et al. // Journal of Cloud Computing. – 2023. – Vol. 12. – Article 171. – DOI: 10.1186/s13677-023-00471-1

6. Dang-Quang N.-M. Deep Learning-Based Autoscaling Using Bidirectional Long Short-Term Memory for Kubernetes / N.-M. Dang-Quang, M. Yoo // Applied Sciences. – 2021. – Vol. 11, № 9. – Article 3835. – DOI: 10.3390/app11093835

7. Meng C. DeepScaler: Holistic Autoscaling for Microservices Based on Spatiotemporal GNN with Adaptive Graph Learning / C. Meng, S. Song, H. Tong, M. Pan, Y. Yu // Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE 2023). – 2023. – P. 53-65. – DOI: 10.1109/ASE56229.2023.00038

8. Wang Y. Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning / Y. Wang, L. Yang // Advances in Computer, Signals and Systems. – 2025. – Vol. 9, № 1. – P. 62-70. – DOI: 10.23977/acss.2025.090109

9. Santos J. Gwydion: Efficient auto-scaling for complex containerized applications in Kubernetes through Reinforcement Learning / J. Santos, E. Reppas, T. Wauters, B. Volckaert, F. De Turck // Journal of Network and Computer Applications. – 2025. – Vol. 234. – Article 104067. – DOI: 10.1016/j.jnca.2024.104067

10. Qiu H. AWARE: Automate Workload Autoscaling with Reinforcement Learning in Production Cloud Systems / H. Qiu, W. Mao, C. Wang, H. Franke, A. Youssef, Z.T. Kalbarczyk, T. Başar, R.K. Iyer // Proceedings of the 2023 USENIX Annual Technical Conference (ATC). – 2023. – URL: https://www.usenix.org/conference/atc23/presentation/qiu-haoran

11. Zou D. OptScaler: A Collaborative Framework for Robust Autoscaling in the Cloud / D. Zou, W. Lu, Z. Zhu, X. Lu, J. Zhou, X. Wang, K. Liu, K. Wang, R. Sun, H. Wang // Proceedings of the VLDB Endowment. – 2024. – Vol. 17, № 12. – DOI: 10.14778/3685800.3685829

12. Rampérez V. FLAS: A combination of proactive and reactive auto-scaling architecture for distributed services / V. Rampérez, J. Soriano, D. Lizcano, J.A. Lara // Future Generation Computer Systems. – 2021. – Vol. 118. – P. 56-72. – DOI: 10.1016/j.future.2020.12.025

13. Chouliaras S. An adaptive auto-scaling framework for cloud resource provisioning / S. Chouliaras, S. Sotiriadis // Future Generation Computer Systems. – 2023. – Vol. 148. – P. 173-183. – DOI: 10.1016/j.future.2023.06.012

14. Pozdniakova O. Self-adaptive autoscaling algorithm for SLA-sensitive applications running on Kubernetes clusters / O. Pozdniakova, A. Cholomskis, D. Mažeika // Cluster Computing. – 2024. – Vol. 27. – P. 2399-2426. – DOI: 10.1007/s10586-023-04082-y

15. Augustyn D.R. Tuning a Kubernetes Horizontal Pod Autoscaler for Meeting Performance and Load Demands in Cloud Deployments / D.R. Augustyn, Ł. Wyciślik, M. Sojka // Applied Sciences. – 2024. – Vol. 14, № 2. – Article 646. – DOI: 10.3390/app14020646

16. Zambianco M. Cost Minimization in Multi-cloud Systems with Runtime Microservice Re-orchestration / M. Zambianco, S. Cretti, D. Siracusa // Proceedings of the 27th Conference on Innovation in Clouds, Internet and Networks (ICIN 2024). – 2024. – DOI: 10.1109/ICIN60470.2024.10494463

17. Qi S. CASA: A Framework for SLO- and Carbon-Aware Autoscaling and Scheduling in Serverless Cloud Computing / S. Qi, H. Moore, N. Hogade, D. Milojicic, C. Bash, S. Pasricha // Proceedings of the International Green and Sustainable Computing Conference (IGSC 2024). – 2024. – DOI: 10.1109/IGSC64514.2024.00010

References:

1. Jeong B. Autoscaling techniques in cloud-native computing: A comprehensive survey / B. Jeong, Y. Jeong // Computer Science Review. – 2025. – Vol. 58. – Article 100791. – DOI: 10.1016/j.cosrev.2025.100791

2. Carrión M.C. Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges / M.C. Carrión // ACM Computing Surveys. – 2022. – Vol. 55, № 7. – P. 1-37. – DOI: 10.1145/3539606

3. Vhatkar K.N. A comprehensive survey on container resource allocation approaches in cloud computing: State-of-the-art and research challenges / K.N. Vhatkar, G.P. Bhole // Web Intelligence. – 2021. – Vol. 19, № 3. – P. 181-204. – DOI: 10.3233/WEB-210474

4. Fourati M.H. Cloud Elasticity of Microservices-Based Applications: A Survey / M.H. Fourati, S. Marzouk, M. Jmaiel // Concurrency and Computation: Practice and Experience. – 2024. – Vol. 37, № 2. – DOI: 10.1002/cpe.8329

5. Hameed A. A Survey of Kubernetes Scheduling Algorithms / A. Hameed, A. Yousif et al. // Journal of Cloud Computing. – 2023. – Vol. 12. – Article 171. – DOI: 10.1186/s13677-023-00471-1

6. Dang-Quang N.-M. Deep Learning-Based Autoscaling Using Bidirectional Long Short-Term Memory for Kubernetes / N.-M. Dang-Quang, M. Yoo // Applied Sciences. – 2021. – Vol. 11, № 9. – Article 3835. – DOI: 10.3390/app11093835

7. Meng C. DeepScaler: Holistic Autoscaling for Microservices Based on Spatiotemporal GNN with Adaptive Graph Learning / C. Meng, S. Song, H. Tong, M. Pan, Y. Yu // Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE 2023). – 2023. – P. 53-65. – DOI: 10.1109/ASE56229.2023.00038

8. Wang Y. Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning / Y. Wang, L. Yang // Advances in Computer, Signals and Systems. – 2025. – Vol. 9, № 1. – P. 62-70. – DOI: 10.23977/acss.2025.090109

9. Santos J. Gwydion: Efficient auto-scaling for complex containerized applications in Kubernetes through Reinforcement Learning / J. Santos, E. Reppas, T. Wauters, B. Volckaert, F. De Turck // Journal of Network and Computer Applications. – 2025. – Vol. 234. – Article 104067. – DOI: 10.1016/j.jnca.2024.104067

10. Qiu H. AWARE: Automate Workload Autoscaling with Reinforcement Learning in Production Cloud Systems / H. Qiu, W. Mao, C. Wang, H. Franke, A. Youssef, Z.T. Kalbarczyk, T. Başar, R.K. Iyer // Proceedings of the 2023 USENIX Annual Technical Conference (ATC). – 2023. – URL: https://www.usenix.org/conference/atc23/presentation/qiu-haoran

11. Zou D. OptScaler: A Collaborative Framework for Robust Autoscaling in the Cloud / D. Zou, W. Lu, Z. Zhu, X. Lu, J. Zhou, X. Wang, K. Liu, K. Wang, R. Sun, H. Wang // Proceedings of the VLDB Endowment. – 2024. – Vol. 17, № 12. – DOI: 10.14778/3685800.3685829

12. Rampérez V. FLAS: A combination of proactive and reactive auto-scaling architecture for distributed services / V. Rampérez, J. Soriano, D. Lizcano, J.A. Lara // Future Generation Computer Systems. – 2021. – Vol. 118. – P. 56-72. – DOI: 10.1016/j.future.2020.12.025

13. Chouliaras S. An adaptive auto-scaling framework for cloud resource provisioning / S. Chouliaras, S. Sotiriadis // Future Generation Computer Systems. – 2023. – Vol. 148. – P. 173-183. – DOI: 10.1016/j.future.2023.06.012

14. Pozdniakova O. Self-adaptive autoscaling algorithm for SLA-sensitive applications running on Kubernetes clusters / O. Pozdniakova, A. Cholomskis, D. Mažeika // Cluster Computing. – 2024. – Vol. 27. – P. 2399-2426. – DOI: 10.1007/s10586-023-04082-y

15. Augustyn D.R. Tuning a Kubernetes Horizontal Pod Autoscaler for Meeting Performance and Load Demands in Cloud Deployments / D.R. Augustyn, Ł. Wyciślik, M. Sojka // Applied Sciences. – 2024. – Vol. 14, № 2. – Article 646. – DOI: 10.3390/app14020646

16. Zambianco M. Cost Minimization in Multi-cloud Systems with Runtime Microservice Re-orchestration / M. Zambianco, S. Cretti, D. Siracusa // Proceedings of the 27th Conference on Innovation in Clouds, Internet and Networks (ICIN 2024). – 2024. – DOI: 10.1109/ICIN60470.2024.10494463

17. Qi S. CASA: A Framework for SLO- and Carbon-Aware Autoscaling and Scheduling in Serverless Cloud Computing / S. Qi, H. Moore, N. Hogade, D. Milojicic, C. Bash, S. Pasricha // Proceedings of the International Green and Sustainable Computing Conference (IGSC 2024). – 2024. – DOI: 10.1109/IGSC64514.2024.00010