

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Дослідження алгоритмів планування «Kubernetes» для розумних міст

Виконав: студент VI курсу, групи СНм-61

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Ловчук О.І.
(підпис) (прізвище та ініціали)

Керівник Дуда О.М.
(підпис) (прізвище та ініціали)

Нормоконтроль Никитюк В.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент Микитишин А.Г.
(підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 25 грудня » 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ловчуку Олегу Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження алгоритмів планування «Kubernetes» для розумних міст

Керівник роботи Дуда Олексій Михайлович, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » листопада 2024 року № 4/7-1141

2. Термін подання студентом завершеної роботи 26 грудня 2024р.

3. Вихідні дані до роботи Наукові публікації про алгоритми планування «Kubernetes» для потреб горизонтального та вертикального масштабування хмарних застосунків розумних міст.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Аналітичний огляд наукової літератури щодо концептів «розумне місто» та «Kubernetes», 1.1 Означення концепту «розумне місто», 1.2 «Kubernetes» для «розумних міст», 1.3 Методика пошуку наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст», 1.4 Аналіз наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст», 1.5 Висновки до першого розділу. 2 Дослідження алгоритмів планування «Kubernetes» для «розумних міст», 2.1 Планування за допомогою багатоцільової оптимізації, 2.2 Планування, орієнтоване на «AI», 2.3 Планування з підтримкою автомасштабування, 2.4 Висновок до другого розділу. 3 Аналіз підходів до масштабування та проєктування рішень у «Kubernetes» для «розумних міст», 3.1 Аналіз сучасних підходів до масштабування в «Kubernetes», 3.2 «Horizontal Pod Autoscaler», 3.3 «Vertical Pod Autoscaler», 3.4 Обговорення, виклики та майбутні пропозиції, 3.5 Висновок до третього розділу. 4 Охорона праці та безпека в надзвичайних ситуаціях, 4.1 Вимоги до серверних приміщень, 4.2 Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн. Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1 Актуальність дослідження. 2 Тема, Мета, Об'єкт та предмет дослідження. 3 Завдання Дослідження. 4 Актуальність теми. 5 «Розумні міста». 6 Аналіз сучасних наукових публікацій з алгоритмів планування. 7 Порівняння різних типів розгортання програм. 8 Порівняння між різними архітектурами програм. 13 Висновки. 14 Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 29 листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.11.2024	
2.	Підбір наукових джерел про дослідження алгоритмів планування «Kubernetes» для розумних міст	29.11.2024-04.12.2024	
3.	Опрацювання наукових публікацій та збір даних по темі роботи	29.11.2024-04.12.2024	
4.	Виконання дослідження згідно мети кваліфікаційної Роботи	05.12.2024-09.12.2024	
5.	Оформлення розділу «Аналітичний огляд наукової літератури щодо концепту «розумне місто» та «Kubernetes»»	05.12.2024-09.12.2024	
6.	Оформлення розділу «Дослідження алгоритмів планування «kubernetes» для «розумних міст»»	05.12.2024-09.12.2024	
7.	Оформлення розділу «Аналіз підходів до масштабування та проєктування рішень у «Kubernetes»»	05.12.2024-09.12.2024	
8.	Аналіз вимог до серверних приміщень	02.12.2024-09.12.2024	
9.	Аналіз державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн	02.12.2024-09.12.2024	
10.	Оформлення кваліфікаційної роботи	10.12.2024-11.12.2024	
11.	Нормоконтроль	12.12.2024-13.12.2024	
12.	Перевірка на плагіат	16.12.2024	
13.	Попередній захист кваліфікаційної роботи	17.12.2024	
14.	Захист кваліфікаційної роботи	26.12.2024	

Студент

Ловчук О.І.
(прізвище та ініціали)

Керівник роботи

Дуда О.М.
(прізвище та ініціали)

АНОТАЦІЯ

Дослідження алгоритмів планування «Kubernetes» для розумних міст // Кваліфікаційна робота освітнього рівня «Магістр» // Ловчук Олег Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2024 // С. 77, рис. – 9, табл. – 7, кресл. – 14, додат. – 1, бібліогр. – 78.

Ключові слова: алгоритм планування, контейнерні програми, «Kubernetes», оркестровка контейнерів, смарт-застосунки, смарт-системи, хмарні сервіси.

Кваліфікаційна робота присвячена розробці алгоритмів планування «Kubernetes» для розумних міст. В першому розділі кваліфікаційної роботи описані методики пошуку наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст». Висвітлено означення «Kubernetes» та його використання для «розумних міст». Розглянуто визначення концепту «розумне місто». Проаналізовано знайдені наукові публікації щодо алгоритмів планування «Kubernetes» для «розумних міст». В другому розділі кваліфікаційної роботи описано основні принципи роботи з алгоритмами планування в «Kubernetes» для «розумних міст». Досліджено використання «Kubernetes» для забезпечення ефективності та автоматизації в інфраструктурах. Подано порівняльний опис підходів до планування. В третьому розділі кваліфікаційної роботи описано ефективність алгоритмів планування у сценаріях з високим навантаженням і протестовано їх у середовищі «Kubernetes». Проаналізовано сучасні підходи до масштабування в «Kubernetes», зокрема «Horizontal Pod Autoscaler» і «Vertical Pod Autoscaler». Проведено тестування цих інструментів у реальних умовах високого навантаження. Об'єкт дослідження: системи управління ресурсами в середовищі «Kubernetes» для «розумних міст». Предмет дослідження: алгоритми планування і масштабування в «Kubernetes».

ANNOTATION

Exploring «Kubernetes» Scheduling Algorithms for Smart Cities // The educational level "Master" qualification work // Oleh Lovchuk // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNm-61 group // Ternopil, 2024 // P. 77, fig. – 9, tables – 7, posters – 14, annexes – 1, ref. – 78.

Key words: scheduling algorithm, container applications, «Kubernetes», container orchestration, smart applications, smart systems, cloud services.

The qualification work is dedicated to the development of Kubernetes scheduling algorithms for smart cities. The first section of the qualification work describes the methods for searching for scientific publications on Kubernetes scheduling algorithms for smart cities. The definition of Kubernetes and its use for smart cities are highlighted. The definition of the concept of smart city is considered. The found scientific publications on Kubernetes scheduling algorithms for smart cities are analyzed. The second section of the qualification work describes the basic principles of working with scheduling algorithms in Kubernetes for smart cities. The use of Kubernetes to ensure efficiency and automation in infrastructures is investigated. A comparative description of planning approaches is provided. The third section of the qualification work describes the effectiveness of scheduling algorithms in high-load scenarios and tests them in the Kubernetes environment. Modern approaches to scaling in Kubernetes are analyzed, in particular, Horizontal Pod Autoscaler and Vertical Pod Autoscaler. These tools were tested in real-world high-load conditions. Research object: resource management systems in the Kubernetes environment for smart cities. Research subject: scheduling and scaling algorithms in Kubernetes.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

ACM (Association for Computing Machinery) – це міжнародна організація, що підтримує фахівців у галузі комп'ютерних наук через дослідження, конференції та професійний розвиток.

AI (Artificial Intelligence) – це технологія, яка дає можливість машинам та комп'ютерам виконувати завдання, що зазвичай потребують людського інтелекту: навчання, розпізнавання мови, обробка зображень та прийняття рішень.

AIM (Automatic Instruction Set Mapping) – це процес, який дає можливість автоматично транслювати або адаптувати програмне забезпечення на іншу архітектуру.

CPU (Central Processing Unit) – це центральний процесор комп'ютера, який виконує основні обчислювальні операції, управляє виконанням програм та обробляє інструкції.

Docker – це платформа для автоматизації розгортання, масштабування та управління контейнерами.

Elsevier – це міжнародний видавець наукових, технічних і медичних публікацій, який пропонує доступ до журналів, книг та баз даних для дослідників, студентів і професіоналів.

Google Scholar – це безкоштовний пошуковий сервіс від «Google», який дає змогу знаходити наукові статті, книги, дисертації та інші академічні публікації з різних галузей знань.

GPU (Graphics Processing Unit) – це процесор, спеціалізований на обробці графічних даних та виконанні складних математичних обчислень.

HPA (Horizontal Pod Autoscaler) – це інструмент «Kubernetes», який автоматично масштабує кількість подів залежно від навантаження, використовуючи метрики «CPU», «RAM» або кастомні показники.

IEEE (Institute of Electrical and Electronics Engineers) – міжнародна організація, що об'єднує фахівців у електротехніці, електроніці та комп'ютерних технологіях, займається стандартизацією та розвитком інновацій.

IIoT (Industrial Internet of Things) – це підсистема «IoT», орієнтована на використання з'єднаних пристроїв, датчиків та інших технологій в промислових сферах.

IoT (Internet of Things) – це концепція мережі, в якій фізичні пристрої, сенсори, побутова техніка та інші об'єкти оснащені підключенням до інтернету та здатні збирати, передавати й обмінюватися даними без участі людини.

ISA (Instruction Set Architecture) – це набір команд, які процесор може виконувати, а також описує структуру даних, регістри, режими адресації та інші аспекти взаємодії між апаратним забезпеченням і програмним забезпеченням.

KEIDS (Knowledge-based Embedded Intelligent Decision System) – це система, яка використовує знання та інтелектуальні алгоритми для прийняття рішень в режимі реального часу в складних або динамічних середовищах.

Kube-Knots – це інструмент для управління та візуалізації робочих навантажень у кластері «Kubernetes», який допомагає створювати топології для різних сценаріїв розгортання.

KubeHICE (Kubernetes Hybrid Integrated Cloud Environment) – це інструмент для створення гібридних хмарних середовищ на базі «Kubernetes».

Kubernetes – відкрита система автоматичного розгортання, масштабування та управління застосунками у контейнерах.

PAS (Performance-Aware Scheduling) – це підхід до планування, який фокусується на оптимізації використання ресурсів з урахуванням їх продуктивності.

QoE (Quality of Experience) – це загальний показник, який оцінює, як кінцеві користувачі сприймають якість сервісів або продуктів, зокрема в контексті телекомунікацій, мультимедійного контенту, Інтернет-послуг або програмного забезпечення.

QoS (Quality of Service) – це технологія для забезпечення потрібної якості роботи мережі. Вона пріоритизує важливий трафік, мінімізує затримки, втрати даних і гарантує потрібну пропускну здатність для критичних сервісів.

RAM (Random Access Memory) – це оперативна пам'ять, яка тимчасово зберігає дані для швидкого доступу процесора.

SLO (Service Level Objective) – це специфічна мета або критерій, що визначає мінімальні вимоги до продуктивності та доступності хмарних сервісів.

Springer – це міжнародний видавець наукових, технічних та медичних публікацій, який пропонує книги, журнали та інші ресурси для дослідників, студентів та професіоналів у різних галузях науки.

Vertical Pod Autoscaler (VPA) – це компонент «Kubernetes», який автоматично коригує ресурси («CPU» та «RAM») для подів в залежності від їх використання.

Алгоритми планування – це методи або набір правил, які використовуються для ефективного розподілу ресурсів і виконання завдань у комп'ютерних системах або в інших сферах.

Контейнер – це ізольована одиниця, яка містить додаток і всі його залежності, надає можливість запускати його в будь-якому середовищі без необхідності в додаткових налаштуваннях.

Контейнер (Kubernetes) – ізольована одиниця, яка містить додаток із усіма необхідними залежностями для його виконання в будь-якому середовищі.

Оптимізація планування – це процес вдосконалення алгоритмів планування з метою досягнення максимальної ефективності у використанні ресурсів, мінімізації витрат часу, зменшення затримок.

Под (Pod) – це найменша і базова одиниця розгортання в «Kubernetes», яка являє собою групу одного або декількох контейнерів, що спільно використовують мережевий простір, сховище та специфікації налаштувань.

Розумне місто (Smart Cities) – це міста, що використовують технології для покращення якості життя та ефективного управління ресурсами.

ЗМІСТ

ВСТУП	10
1 АНАЛІТИЧНИЙ ОГЛЯД НАУКОВОЇ ЛІТЕРАТУРИ ЩОДО КОНЦЕПТІВ «РОЗУМНЕ МІСТО» ТА «KUBERNETES»	12
1.1 Означення концепту «розумне місто».....	12
1.2 «Kubernetes» для «розумних міст»	14
1.3 Методика пошуку наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст».....	18
1.4 Аналіз наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст».....	21
1.5 Висновки до першого розділу	30
2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПЛАНУВАННЯ «KUBERNETES» ДЛЯ «РОЗУМНИХ МІСТ»	32
2.1 Планування за допомогою багатоцільової оптимізації	32
2.2 Планування, орієнтоване на «AI»	35
2.3 Планування з підтримкою автомасштабування	41
2.4 Висновок до другого розділу	45
3 АНАЛІЗ ПІДХОДІВ ДО МАСШТАБУВАННЯ ТА ПРОЄКТУВАННЯ РІШЕНЬ У «KUBERNETES» ДЛЯ «РОЗУМНИХ МІСТ».....	46
3.1 Аналіз сучасних підходів до масштабування в «Kubernetes».....	46
3.2 «Horizontal Pod Autoscaler»	48
3.3 «Vertical Pod Autoscaler».....	50
3.4 Обговорення, виклики та перспективи подальших досліджень.....	52
3.5 Висновок до третього розділу	58
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	59
4.1 Вимоги до серверних приміщень.....	59
4.2 Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн	63
4.3 Висновок до четвертого розділу	66

ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ.....	70
ДОДАТКИ	

ВСТУП

Актуальність теми. У міру розширення хмарних сервісів зростає потреба в покращенні продуктивності інфраструктури центрів обробки даних «розумних міст». Високопродуктивні обчислення, передові мережеві рішення та стратегії оптимізації ресурсів можуть допомогти центрам обробки даних «розумних міст» підтримувати високу швидкість і ефективність опрацювання даних, необхідні для надання високоякісних хмарних послуг жителям та гостям «розумних міст». Запуск контейнерних програм є однією з таких стратегій оптимізації, яка пропонує такі переваги, як покращена портативність, підвищена безпека, краще використання ресурсів, швидше розгортання та масштабування, а також покращена інтеграція та взаємодію програмно-алгоритмічних засобів «розумних міст». Ці переваги можуть допомогти організаціям покращити розгортання застосунків і керування ними, даючи змогу їм швидше й ефективніше реагувати на динамічні потреби бізнесу. «Kubernetes» – це система оркестровки контейнерів, призначена для автоматизації розгортання, масштабування та керування контейнерними програмами. Однією з його ключових особливостей є можливість планувати розгортання та виконання контейнерів у кластері вузлів за допомогою алгоритму планування. Цей алгоритм визначає найкраще розміщення контейнерів на доступних вузлах у кластері. У даній роботі запропоновано вичерпний огляд різних алгоритмів планування в контексті «Kubernetes». Охарактеризовано та згруповано їх у чотири підкатегорії: загальне планування, планування на основі багатоцільової оптимізації, планування, орієнтоване на штучний інтелект, і планування з підтримкою автоматичного масштабування, а також визначаємо прогалини та задачі, які потребують подальшого дослідження.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є підвищення ефективності планування та масштабування ресурсів у системах «Kubernetes» для «розумних міст». Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати стан досліджень в області алгоритмів планування «Kubernetes» для «розумних міст».
- Дослідити існуючі на даний час методи планування та масштабування ресурсів у «Kubernetes» для «розумних міст».
- Проаналізувати методи планування та масштабування ресурсів у «Kubernetes» для «розумних міст».
- Виконати порівняння існуючих алгоритмів планування «Kubernetes» для «розумних міст».

Об’єкт дослідження системи управління ресурсами в середовищі «Kubernetes» для «розумних міст».

Предмет дослідження. алгоритми планування і масштабування в «Kubernetes».

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що отримав подальший розвиток підходу до оптимізації алгоритмів планування в «Kubernetes» для «розумних міст».

Практичне значення одержаних результатів. Виконано моделювання та тестування алгоритмів планування Kubernetes у сценаріях з високими навантаженнями, що дозволило оцінити їхню ефективність та стабільність у динамічних умовах.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 18–19 грудня 2024 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додаток А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 78 найменувань та один додаток. Загальний обсяг кваліфікаційної роботи складає 77 сторінки, з них 48 сторінки основного тексту, який містить 9 рисунків та 7 таблиць.

1 АНАЛІТИЧНИЙ ОГЛЯД НАУКОВОЇ ЛІТЕРАТУРИ ЩОДО КОНЦЕПТІВ «РОЗУМНЕ МІСТО» ТА «KUBERNETES»

1.1 Означення концепту «розумне місто»

«Розумні міста» останнім часом викликають значний інтерес і викликають інвестиції, оскільки в різних частинах світу реалізуються численні ініціативи та проекти. Однак, незважаючи на цей інтерес та інвестиції, не існує загальновизнаного визначення «розумного міста» [1]. Відсутність чіткого визначення може створити задачі для політиків, містобудівників і зацікавлених сторін, які прагнуть розробити та впровадити ініціативи «розумного міста».

Комплексне дослідження визначень «розумного міста» шляхом структурованого огляду наукової літератури виявило багато визначень із різними концептуальними тлумаченнями, які часто замінюють термін «розумний» альтернативними дескрипторами, такими як «розумний» або «цифровий» [2]. У результаті термін «розумне місто» залишається розпливчастим і неоднозначним поняттям із непослідовним використанням у різних контекстах. У той час як багато ініціатив «розумного міста» віддають пріоритет темам стійкості, ефективності та якості життя [3], відсутність універсального визначення пояснюється широким спектром ідей та ініціатив, охоплених цим терміном. Ці ідеї та ініціативи відрізняються в різних географічних і соціально-економічних контекстах [4].

Поданий в [5] аналіз хмари слів, проведений із визначеннями «розумних міст», показав, що основний акцент у визначеннях «розумних міст» зосереджено на покращенні якості життя мешканців, як показано на рисунку 1.1. Помітність терміна «технологія» свідчить про те, що це фундаментальне інструмент, який використовується для реалізації проектів «розумного міста». Однак важливо зазначити, що, незважаючи на поширеність цих тем, «розумним містам» бракує універсального визначення.

забезпечити чітку основу для розуміння того, що таке «розумне місто» та якими мають бути його цілі. *«Розумне місто – це міська територія, яка використовує технологічні або нетехнологічні послуги чи продукти, які: покращують соціально-етичне благополуччя громадян; забезпечити якість, продуктивність та інтерактивність міських послуг знизити витрати і споживання ресурсів; та посилити контакт між громадянами та урядом» [7].*

1.2 «Kubernetes» для «розумних міст»

«Розумні міста» інтегрують велику кількість «IoT-пристроїв», а також концентрують високі навантаження обчислювальних і мережових систем для забезпечення функцій: моніторинг транспорту, управління енергоспоживанням, контролю якості повітря, відеоспостереження та інші служби. «Kubernetes», як система оркестрації контейнерів, пропонує можливість автоматизованого керування та масштабування цих служб, забезпечуючи надійність і стійкість роботи за допомогою динамічного планування навантаження. Тому алгоритми планування в «Kubernetes» для «розумних міст» є ключовими компонентами ефективного управління та розподілу ресурсів в інфраструктурі міських систем.

«Kubernetes» – це платформа з відкритим кодом для автоматизації розгортання, масштабування та керування контейнерними програмами. Це забезпечує можливість розробникам зосередитися на створенні та розгортанні своїх програм, не турбуючись про базову інфраструктуру. «Kubernetes» використовує декларативний підхід до керування програмами, коли користувачі вказують бажані стани програми, а система їх підтримує. Він також надає надійні інструменти для моніторингу та керування застосунками, включаючи механізми самовідновлення для автоматичного виявлення збоїв і відновлення. Загалом «Kubernetes» пропонує потужне та гнучке рішення для керування контейнерними програмами у виробничих середовищах.

«Kubernetes» добре підходить для веб-застосунків на основі мікросервісів, де кожен компонент можна запускати в окремому контейнері. Контейнери легкі, їх можна легко створювати та знищувати, забезпечуючи швидше та ефективніше

використання ресурсів, ніж віртуальні машини, як показано на рисунку 1.2. «Kubernetes» автоматизує розгортання, масштабування та керування контейнерами в кластері машин, покращуючи використання ресурсів ефективний і гнучкий. Це спрощує процес створення та підтримки складних програм.

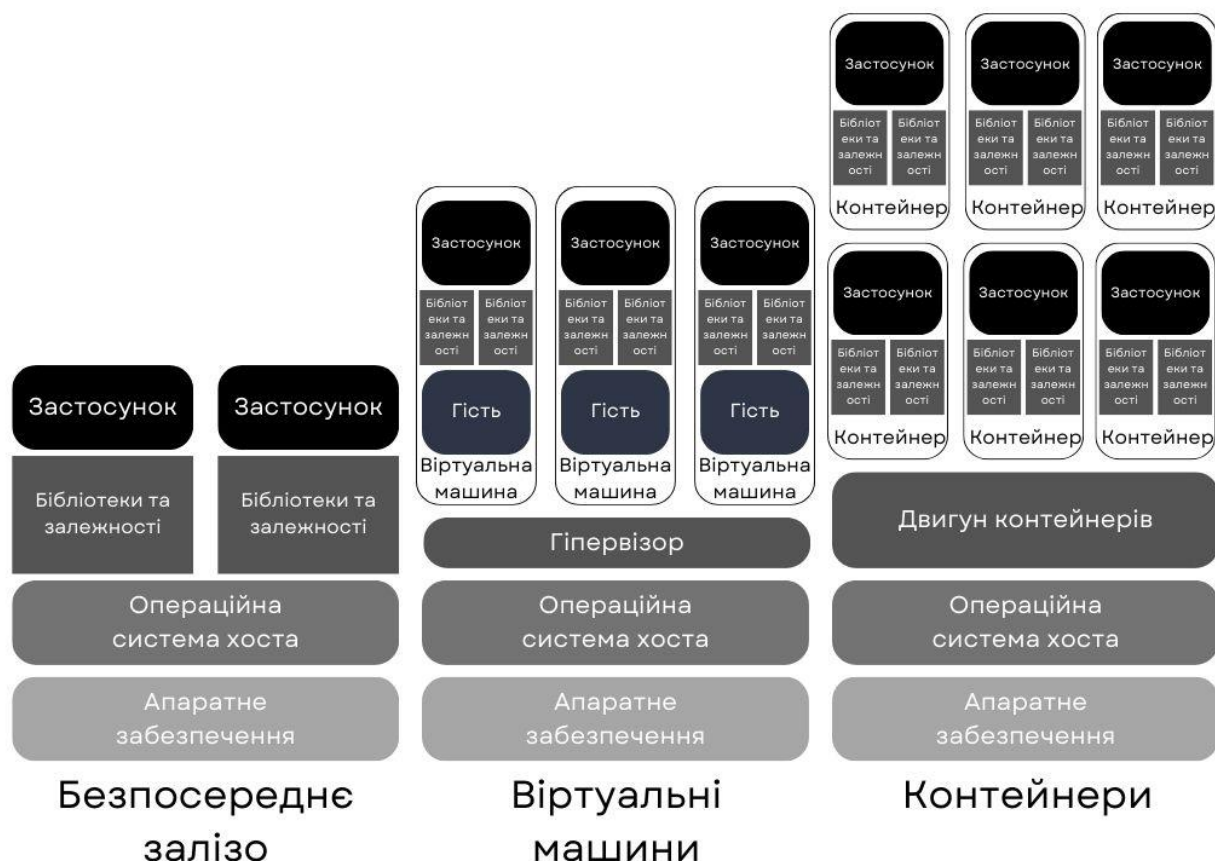


Рисунок 1.2 – Порівняння різних типів розгортання програм

Архітектура, заснована на мікросервісі, передбачає поділ програми на невеликі незалежні модулі, які називаються мікросервісами, зображені на рисунку 1.3. Кожна мікросервіс відповідає за певний аспект програми, і вони обмінюються інформацією через шину повідомлень. Архітектура «Tis» пропонує декілька переваг, наприклад можливість автоматизувати розгортання, масштабування та керування. Оскільки кожен мікросервіс є незалежним і ним можна керувати та оновлювати окремо, легше вносити зміни, не впливаючи на всю систему. Крім того, мікросервіси можуть бути написані різними мовами та

працювати на різних серверах, забезпечуючи більшу гнучкість у процесі розробки.

«Kubernetes» може швидко адаптуватися до різних типів інтенсивності попиту. Наприклад, якщо веб-застосунок має мало відвідувачів у певний час, його можна зменшити до декількох блоків, використовуючи мінімальні ресурси, щоб зменшити витрати. Однак, якщо програма стає надзвичайно популярною та отримує велику кількість відвідувачів одночасно, її можна розширити, щоб обслуговуватись великою кількістю модулів, що робить її здатною обробляти майже будь-який рівень попиту.

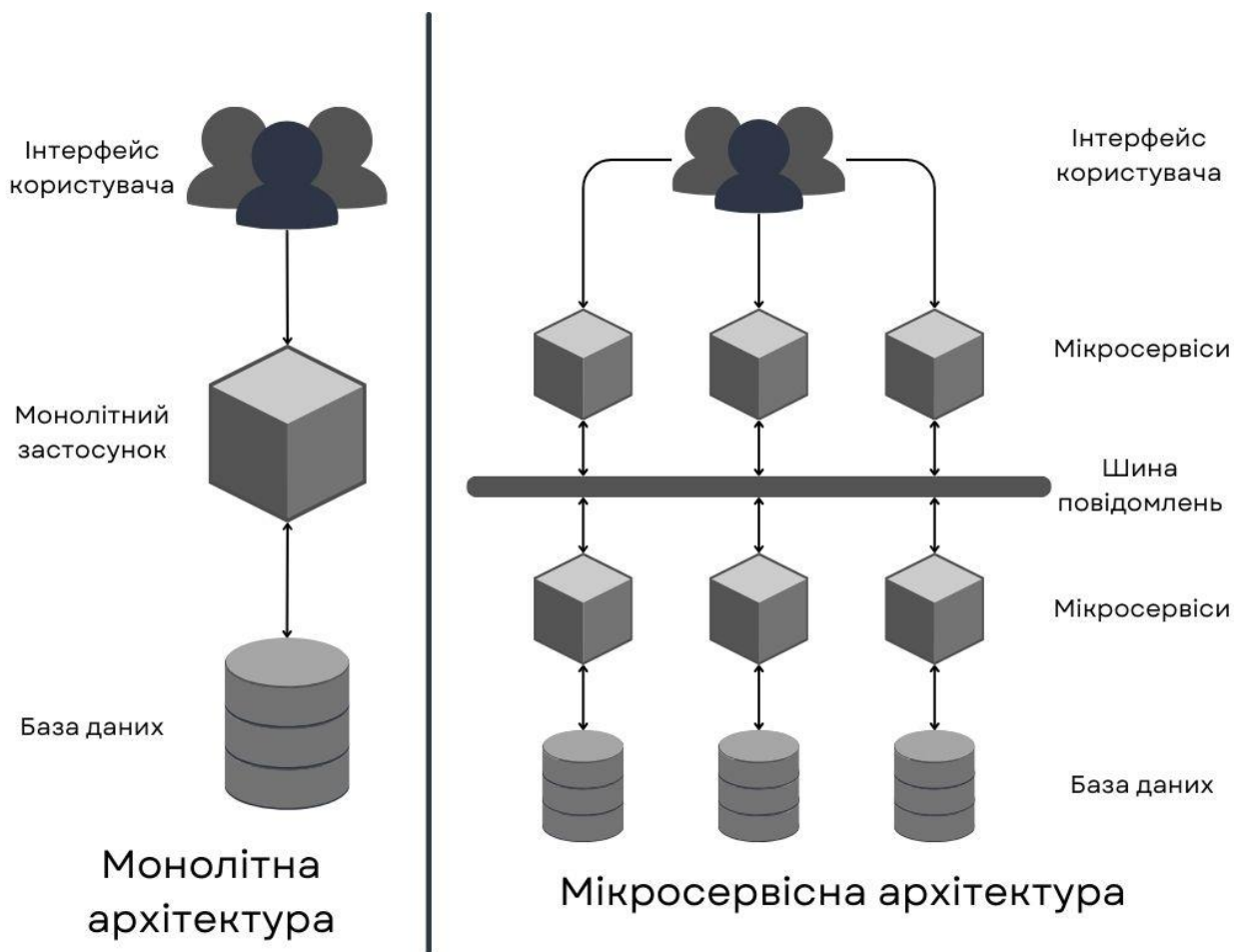


Рисунок 1.3 – Порівняння між різними архітектурами програм

«Kubernetes» використовується багатьма організаціями в різноманітних сферах базових програм і завоював довіру як найкращий варіант для керування та розгортання контейнерних програм. З точки зору останніх програм, «Kubernetes» виявився безцінним ресурсом для ІТ-інфраструктури, оскільки

вони забезпечують стійкий шлях до безсерверних обчислень, що призведе до полегшення задач в ІТ-адмініструванні [8]. Безсерверні обчислення забезпечать наскрізні покращення безпеки, але також призведуть до появи нових задач з інфраструктурою та безпекою, як описано в [8].

У міру того, як обчислювальна парадигма рухається до периферійних і туманних обчислень, «Kubernetes» виявляється універсальним рішенням, яке забезпечує безперебійне мережеве керування між хмарними та периферійними вузлами [9],[10],[11]. «Kubernetes» стикається з багатьма задачами під час розгортання в середовищі «IoT». Ці виклики варіюються від оптимізації розподілу мережевого трафіку [9], оптимізації політики маршрутизації потоку [10] та розподілу обчислювальних ресурсів периферійних пристроїв [11].

Як видно з різноманітних програм і викликів, пов'язаних з «Kubernetes», необхідно вивчити запропоновані алгоритми у відповідній області, щоб визначити сучасні та майбутні напрямки досліджень. Численні дослідження були зосереджені на розробці нових алгоритмів для «Kubernetes». Основна мотивація цього опитування полягає в тому, щоб надати вичерпний огляд найсучасніших технологій у сфері алгоритмів планування «Kubernetes». Перегляд наявної наукової літератури та визначення ключових теорій, методів і результатів попередніх досліджень надає можливість критично оцінити переваги та обмеження існуючих підходів. У процесі аналізу можуть бути виявлені прогалини та відкриті питання в існуючій науковій літературі, а також сформульовані пропозиції щодо напрямків майбутніх досліджень. Загальною метою є внесок у розвиток знань у цій галузі та створення корисного ресурсу для дослідників і практиків, які займаються алгоритмами планування «Kubernetes».

Наскільки відомо авторам, не знайдено пов'язаних опитувань, які конкретно стосувалися б розглянутої теми. Знайдені опитування здебільшого спрямовані на оркестровку контейнерів загалом, включаючи «Kubernetes», наприклад [12],[13],[14],[15]. Ці опитування пов'язані з «Kubernetes» без націлювання на планування та глибоке занурення в нього, а деякі навіть не зосереджувалися на «Kubernetes». Наприклад, деяке зосереджене планування в хмарі [16] та пов'язані з ним завдання [17]. Інші націлені на застосунки великих

даних у мережах центрів обробки даних [18] або середовищах туманних обчислень [19]. Автори знайшли два тісно пов'язані та добре організовані опитування [20] і [21], які детально розглядали планування «Kubernetes». Однак кваліфікаційна робота відрізняється від цих двох опитувань з точки зору таксономії, тобто вони були націлені на різні аспекти та цілі планування, а наукова література розділена на чотири різні підкатегорії: загальне планування, планування на основі багатоцільової оптимізації, планування, орієнтоване на «AI», і планування з увімкненим автомасштабуванням. Таким чином, зосереджуючись на широкому діапазоні схем, пов'язаних із багатоцільовою оптимізацією та «AI», на додаток до основного планування з автомасштабуванням. Категоризація є більш детальною та новою порівняно з існуючими опитуваннями.

1.3 Методика пошуку наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст»

Щоб визначити релевантні дослідження для аналізу наукових публікацій щодо алгоритмів планування «Kubernetes» для розумних міст, було проведено комплексний пошук наукової літератури, використовуючи такі бази даних: «IEEE», «ACM», «Elsevier», «Springer» і «Google Scholar». Було використано такі пошукові терміни: «Kubernetes», «scheduling algorithms» та «scheduling optimization». Пошуку був обмежений дослідженнями, опублікованими за останні шість років і написаними англійською мовою.

Було визначено загалом сто двадцять чотири дослідження за допомогою пошуку в базі даних, рисунок 1.4. Після чого було переглянуто анотації даних досліджень, щоб визначити ті, які мають відношення до огляду. Були включені дослідження, які не зосереджувалися на алгоритмах планування «Kubernetes», а також ті, які не були оригінальними дослідженнями чи оглядовими статтями. На рисунку 1.5 після цього початкового скринінгу у нас залишилося шістдесят сім досліджень.

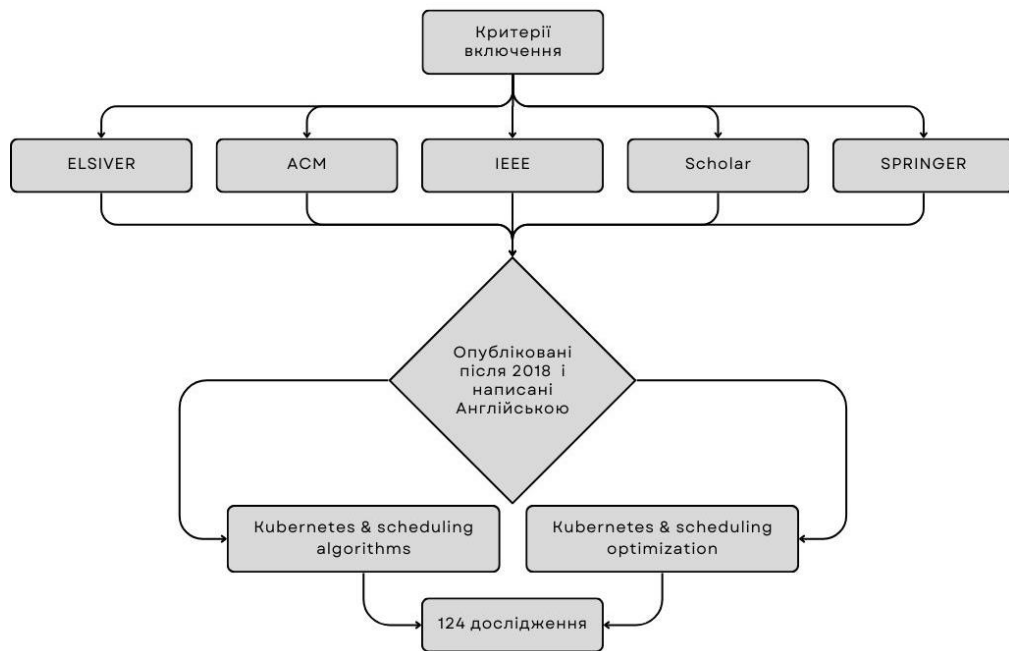


Рисунок 1.4 – Критерії включення

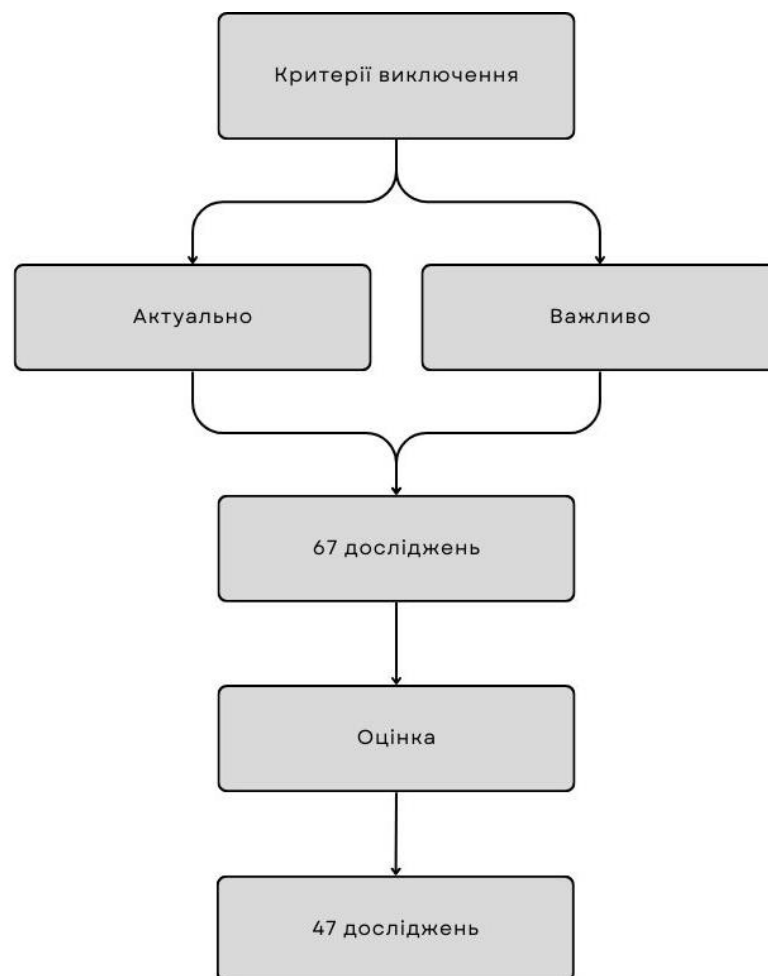


Рисунок 1.5 – Критерії виключення

Переглянуто тексти решти досліджень, щоб визначити їх придатність для включення в огляд. Виключино дослідження, які не відповідали критеріям включення, якими були: зосередженість на оптимізації алгоритмів планування «Kubernetes», надання оригінальних досліджень або критичної оцінки існуючих підходів і написання англійською мовою та опубліковане в останні шість років. Після цього остаточного відбору було включино сорок сім досліджень в огляд, див. рисунок 1.5. Річний розподіл статей можна побачити на рисунку 1.6.

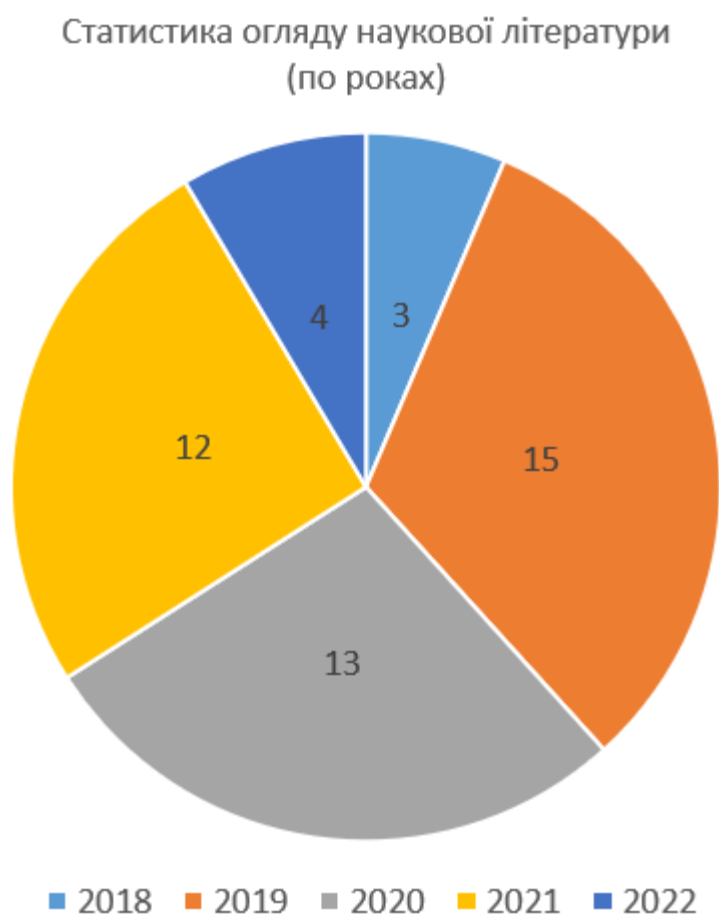


Рисунок 1.6 – Детальні статистичні дані, що показують щорічний розподіл проаналізованих досліджень

Також проведено пошук у списках посилань на включені дослідження, щоб виявити будь-які додаткові релевантні дослідження, які не були охоплені пошуком у базі даних. Не було визначено жодних додаткових досліджень у цьому процесі. Таким чином, огляд включає сорок сім досліджень алгоритмів планування «Kubernetes», опублікованих за останні шість років. Ці дослідження

подають різноманітні методи дослідження, включаючи опитування, експерименти та моделювання.

1.4 Аналіз наукових публікацій щодо алгоритмів планування «Kubernetes» для «розумних міст»

Розподіл проаналізованих дослідницьких робіт у кожній категорії можна побачити на рисунку 1.7. Наукова література в кожній підкатегорії аналізується, а потім узагальнюється на основі шести параметрів, наведених нижче:

- Цілі.
- Методологія/Алгоритми.
- Експерименти.
- Висновки.
- Застосунки.
- Обмеження.

Статистика огляду наукової літератури
(по категоріям)



Рисунок 1.7 – Статистичні показники категорій проаналізованих наукових публікацій

Сфера алгоритмів планування «Kubernetes» привернула значну увагу дослідників і практиків останніми роками. Все більше наукової літератури досліджує потенційні переваги та недоліки використання різних алгоритмів планування для оптимізації продуктивності кластера «Kubernetes».

Однією з ключових тем у науковій літературі є необхідність ефективного ефективного планування робочих навантажень у середовищі «Kubernetes». Багато досліджень підкреслюють обмеження традиційних підходів до планування, які часто не справляються зі складним і динамічним характером робочих навантажень у кластері «Kubernetes». Як наслідок, зростає інтерес до використання розширених алгоритмів планування для забезпечення ефективного розподілу обчислювальних ресурсів у кластері.

Іншою ключовою темою в науковій літературі є потенційні переваги розширених алгоритмів планування для «Kubernetes». Багато досліджень підкреслюють потенціал цих алгоритмів для покращення використання ресурсів, зменшення затримки та підвищення загальної продуктивності кластера. Крім того, розширені алгоритми планування мають потенціал для підтримки розробки нових програм і служб у середовищі «Kubernetes», орієнтоване на «AI», зокрема, аналітика в режимі реального часу, машинне та глибоке навчання.

Незважаючи на ці потенційні переваги, наукова література визначає декілька завдань і обмежень алгоритмів планування «Kubernetes». Однією з ключових задачею є потреба врахувати мінливий характер робочих навантажень і застосунків у кластері. Тому різні автори зосереджувалися на вдосконаленні функції автоматичного масштабування в плануванні «Kubernetes», щоб надати можливість автоматичного коригування ресурсів, виділених пакетам, на основі поточного попиту. Тому доцільно провести детальне дослідження процесів планування контейнерів з увімкненим автомасштабуванням. Інші виклики включають потребу в управлінні та координації декількох алгоритмів планування, а також у забезпеченні стабільності та продуктивності всієї системи. Загалом у науковій літературі стверджується, що розширені алгоритми планування пропонують багатообіцяюче вирішення задач, пов'язаних із

складним і динамічним характером робочих навантажень у кластері «Kubernetes». Проте необхідні подальші дослідження для вирішення обмежень і завдань цих алгоритмів, а також для вивчення їхнього потенційного застосування та переваг.

У [22], для розгортання в «розумних містах» автори пропонують метод мережевого планування для програм на основі контейнерів. Їхня стратегія реалізується на практиці як доповнення до вбудованої системи планування «Kubernetes» за замовчуванням, яка є оркеструвальником із відкритим кодом для автоматичного керування та розгортання мікросервісів. Використовуючи застосунки «розумного міста» на основі контейнерів, автори оцінюють ефективність запропонованого підходу до планування та порівнюють його з ефективністю вбудованого механізму планування «Kubernetes» за замовчуванням. Порівняно з технікою за замовчуванням, вони виявили, що запропоноване рішення зменшує затримку мережі майже на вісімдесят відсотків.

У [23], автори пропонують новий кластерний планувальник «Stratus», який спеціалізується на оркеструванні пакетного виконання завдань на віртуальних кластерах у публічних платформах інфраструктури як послуги. «Stratus» зосереджується на мінімізації грошових витрат шляхом агресивного розміщення завдань на машинах на основі оцінок часу виконання, тобто, щоб заощадити гроші, виділені ресурси будуть або здебільшого заповнені, або порожні, щоб потім їх можна було звільнити. Використовуючи відстеження робочого навантаження від «Two Sigma» та «Google», автори оцінюють «Stratus» і встановлюють, що запропонований Stratus знижує витрати на від сімнадцяти до сорока чотирьох відсотків порівняно з контрольними показниками віртуального кластерного планування.

У [24], автори пропонують новий алгоритм планування «AlloX» для оптимізації роботи в спільних кластерах, які використовують взаємозамінні ресурси: центральні процесори, графічні процесори та інші прискорювачі. «AlloX» перетворює задачу планування в задачу двостороннього зіставлення «mincost» і забезпечує динамічний справедливий розподіл з часом. Автори теоретично та емпірично демонструють, що «AlloX» працює краще, ніж існуючі

рішення, за наявності взаємозамінних ресурсів, і вони показують, що він може значно скоротити середній час виконання роботи, одночасно забезпечуючи високу ефективність обчислень, водночас запобігаючи браку ресурсів.

У [25], автори пропонують гетерогенну стратегію розподілу завдань для економічно ефективної оркестровки контейнерів в інфраструктурі хмарних обчислень на основі «Kubernetes» з еластичними обчислювальними ресурсами. Запропонована стратегія має три основні особливості: підтримку неоднорідних конфігурацій завдань, коригування розміру кластера за допомогою алгоритмів автомасштабування та механізм перепланування для вимкнення недостатньо використовуваних екземплярів віртуальної машини та перерозподілу відповідних завдань без втрати виконання завдання. Автори оцінюють свій підхід із використанням австралійської національної хмарної інфраструктури і показують, що вона може зменшити загальну вартість на від двадцяти трьох до тридцяти двох відсотків порівняно зі стандартною структурою «Kubernetes».

У [26], щоб створити «Kube-Knots», автори поєднують запропонований ними рівень оркестровки ресурсів із підтримкою «GPU», із оркестровником контейнерів «Kubernetes». За допомогою динамічної оркестровки контейнерів «Kube-Knots» динамічно збирає невикористані обчислювальні цикли, уможливорюючи спільне розміщення пакетних і критичних програм із затримкою та підвищуючи загальне використання ресурсів. Автори демонструють, що запропоновані стратегії планування збільшують середнє та дев'яносто дев'ятий проценти загальнокластерного «GPU» до вісімдесяти відсотків у випадку робочих навантажень «HPC», коли вони використовуються для планування робочих навантажень у центрі обробки даних за допомогою «Kube-Knots» на десятивузловому кластері «GPU». Крім того, запропоновані планувальники зменшують споживання енергії в кластері в середньому на тридцять три відсотки для трьох окремих робочих навантажень і збільшують середній час виконання завдань навантажень глибокого навчання до тридцяти шести відсотків порівняно з сучасними планувальниками.

У [27], автори пропонують цілісну систему планування для «Kubernetes», яка замінює планувальник за замовчуванням і враховує як програмне, так і

апаратне забезпечення для підвищення ефективності центру обробки даних. Автори стверджують, що шляхом впровадження апаратного моделювання в програмне рішення інтелектуальний планувальник може значно підвищити ефективність центру обробки даних. Під час початкового розгортання автори спостерігали зниження енергоспоживання від десяти до двадцяти відсотків.

У роботі [28] автор описує «KCSS», абсолютно нову стратегію планування контейнерів «Kubernetes». Метою «KCSS» є підвищення продуктивності з точки зору тривалості виготовлення та енергоспоживання шляхом максимально ефективного планування надісланих користувачами контейнерів. Для кожного нещодавно надісланого контейнера «KCSS» вибирає найкращий вузол на основі низки факторів, пов'язаних із хмарною інфраструктурою та вимогами користувача, використовуючи техніку багатокритеріального аналізу рішень. Автор використовує мову програмування Go для створення «KCSS» і показує, як він працює краще, ніж альтернативні методи планування контейнерів у різноманітних ситуаціях.

У [29], автори подають структуру планування «GPU» на основі топології для «Kubernetes». Структура базується на традиційному алгоритмі планування графічного процесора «Kubernetes», але вводить концепцію кластерної топології графічного процесора, яка відновлюється в дереві вартості доступу до ресурсів кластера графічного процесора. Це дає змогу ефективніше планувати різні сценарії використання ресурсів «GPU». Запропонована структура була використана у виробничій практиці Tencent і, як повідомляється, покращила використання ресурсів кластерів «GPU» приблизно на десять відсотків.

У [30], автори пропонують покращену конструкцію для планування «Kubernetes», яка враховує фізичні, робочі та мережеві параметри на додаток до програмного стану, щоб уможливити кращу оркестровку та керування застосунками периферійних обчислень. Вони порівнюють запропонований дизайн із планувальником «Kubernetes» за замовчуванням і показують, що він пропонує покращену відмовостійкість і можливості динамічної оркестровки.

У роботі [31], використовуючи показники справедливості, включаючи домінуючу справедливість ресурсів, попит на ресурси та середній час

очікування, автори описують політику планування для кластерів «Kubernetes». «KubeSphere», керований політикою метапланувальник, створений авторами, робить можливим планувати завдання відповідно до загальних потреб кожного користувача в ресурсах і поточного споживання. Згідно з результатами експерименту, запропонована політика збільшила справедливість у кластері з кількома орендарями.

У [32], автори пропонують спеціальний планувальник «Kubernetes», який враховує обмеження затримки та надійність краю під час прийняття рішень щодо планування. Автори стверджують, що цей тип планувальника необхідний для периферійної інфраструктури, де програми часто чутливі до затримок, а інфраструктура схильна до збоїв. Автори демонструють своє розширення «Kubernetes» і випускають рішення з відкритим кодом.

У [33], автори пропонують унікальний метод для планування пакетів «Kubernetes», який використовує переваги динамічних вимірювань мережі, зібраних «Istio Service Mesh». За словами авторів, цей підхід може повністю автоматизувати збереження до п'ятидесяти відсотків міжвузлової пропускної здатності та до тридцяти семи відсотків часу відповіді застосунків, що має вирішальне значення для впровадження «Kubernetes» у випадках використання 5G для «розумних міст».

У [34], розгортанням контейнерів «Docker» у гетерогенному кластері з ресурсами «CPU» та «GPU» можна керувати через розроблену авторами структуру динамічного планування для «Kubernetes». Часова шкала «Kubernetes» «Pod» і попередні дані про виконання контейнерів враховуються платформою, відомою як «KubCG», для оптимізації розгортання нових контейнерів. Згідно з дослідженнями, які автори провели для перевірки свого алгоритму, час, необхідний для виконання завдань, можна скоротити до шістдесяти чотирьох відсотків за допомогою «KubCG».

У [35], автори пропонують гібридну структуру планування зі спільним станом для «Kubernetes», яка поєднує в собі переваги централізованого та розподіленого планування. Платформа використовує розподілені агенти планування для делегування більшості завдань і функцію корекції планування

для обробки неперіоритезованих і незапланованих завдань. На основі всього стану кластера приймаються рішення щодо планування, які потім синхронізуються та оновлюються агентом головного стану. Автори провели експерименти, щоб перевірити поведінку запропонованого ними планувальника, і виявили, що він добре працює в різних сценаріях, включаючи перехід після відмови та відновлення. Вони також виявили, що інші централізовані структури планування можуть не працювати добре в ситуаціях: перешкоди спільному розміщенню або пріоритетне вилучення.

У [36], автори подають розробку та реалізацію «KubeNICE», контейнерного оркестратора з урахуванням продуктивності для гетерогенних архітектур «ISA» на хмарних платформах. «KubeNICE» розширює «Kubernetes» за допомогою двох функціональних підходів, «AIM» і «PAS» (планування з урахуванням продуктивності), для обробки різнорідних контейнерів «ISA» та планування відповідно до обчислювальних можливостей вузлів кластера.

Автори провели експерименти, щоб оцінити «KubeNICE», і виявили, що він не додає додаткових витрат на оркестровку контейнера та був ефективним для оцінки продуктивності та планування ресурсів. Тей також продемонстрував переваги «KubeNICE» у декількох практичних сценаріях, показавши, наприклад, збільшення використання «CPU» на сорок відсотків за умови усунення неоднорідності.

У [37], автори пропонують два алгоритми динамічного планування, «Balanced-Disk-IO-Priority» («BDI») і «Balanced-CPU-Disk-IO-Priority» («BCDI»), щоб вирішити завдання, коли планувальник «Kubernetes» не приймає диск I/O врахування навантаження на вузли. «BDI» призначений для покращення балансу дискового вводу-виводу між вузлами, тоді як «BCDI» розроблений для вирішення завдання дисбалансу навантаження «CPU» і дискового вводу-виводу на одному вузлі. Автори проводять експерименти, щоб оцінити алгоритми та виявити, що вони ефективніші, ніж стандартні алгоритми планування «Kubernetes».

В таблиці 1.1 подано характеристики алгоритмів масштабування для планування ресурсів.

Таблиця 1.1 – Характеристики алгоритмів масштабування для планування ресурсів

Цілі	Методологія/Алгоритми
[23] Планування публічного хмарного контейнера.	Зосередження на організації пакетного виконання завдань на віртуальних кластерах.
[28] Описано нову стратегію планування контейнерів «Kubernetes» («KCSS»). Підвищте ефективність планування багатьох надісланих онлайн контейнерів.	Використовуючи низку факторів, вибрано найкращий вузол для кожного щойно поданого контейнера. Щоб об'єднати всі критерії в єдиний ранг, використано методом «Техніка порядку визначення пріоритетів за подібністю до ідеального рішення».
[30] Розроблено контекстно-залежний планувальник «Kubernetes», який враховує фізичні, робочі та мережеві параметри, щоб покращити доступність послуг і продуктивність у периферійних обчисленнях 5G.	Інтеграція даних периферійного пристрою в режимі реального часу в алгоритм прийняття рішень планувальника.
[38] Досліджено, як безсерверні фреймворки, побудовані на системах «Kubernetes», можуть ефективніше планувати модулі у великих паралельних програмах.	Щоб ще більше підвищити ефективність планування модулів у безсерверних хмарних парадигмах, пропонується підхід до планування, який використовує одночасне планування одного пакета.

У [38], автори пропонують алгоритм для оптимізації планування модулів у безсерверній структурі на платформі «Kubernetes». Автори стверджують, що планувальник «Kubernetes» за замовчуванням, який працює окремо, не підходить для швидкого розгортання та роботи модулів у безсерверній структурі. Щоб вирішити цю задачу, автори пропонують алгоритм, який використовує одночасне планування модулів для підвищення ефективності планування ресурсів у безсерверній структурі. Завдяки попередньому тестуванню автори виявили, що їхній алгоритм зміг значно зменшити затримку запуску модуля, зберігаючи при цьому збалансоване використання ресурсів вузла.

У [39], автори подають альтернативний алгоритм для балансування навантаження в розподілених обчислювальних середовищах. Алгоритм використовує міграцію завдань, щоб збалансувати робоче навантаження між процесорами з різними можливостями та конфігураціями. Автори визначають

мітки, щоб класифікувати завдання за різними категоріями та налаштувати кластери, призначені для конкретних типів завдань.

Алгоритми масштабування можна класифікувати з метою оптимізації ресурсів (див. таблицю 1.2).

Таблиця 1.2 – Характеристики алгоритмів масштабування для оптимізації ресурсів

Цілі	Методологія/Алгоритми
[25] Початкове розміщення контейнерів оптимізується за допомогою упаковки завдань.	Неоднорідні конфігурації завдань, алгоритми автомасштабування, механізм перепланування.
[26] Розроблено рівень оркестровки ресурсів із підтримкою «GPU» для центрів обробки даних. Покращено використання ресурсів і зменшено експлуатаційні витрати в центрах обробки даних. Що покращило якість обслуговування «QoS» для запитів користувачів.	Подано «Kube-Knots», інтегрований у «Kubernetes» рівень оркестровки ресурсів, який підтримує «GPU». «KubeKnots» використовує динамічну оркестровку контейнера для динамічного збирання доступних циклів обчислення. Два методи планування на основі «GPU» створено для планування робочих навантажень у масштабі центру обробки даних за допомогою «Kube-Knots».
[40] Створено підхід керування зворотним зв'язком для надання еластичних контейнерів систем на основі «Kubernetes» для веб-систем.	Поєднання лінійної моделі з моделлю черги зі змінною швидкістю обробки може підвищити точність вихідних помилок.
[35] Описано новий метод упорядкування робочих навантажень у кластері «Kubernetes».	Рамкова модель для гібридного планування спільного стану. На основі загального стану кластера визначаються рішення щодо планування.
[39] Увімкнено безпечне розміщення найкращих процесів і чутливих до затримок служб у кластерах «Kubernetes», щоб збільшити використання ресурсів. Гнучко розподіляйте ресурси між різними категоріями навантаження. Покращено можливості апаратної та програмної ізоляції для контейнерів.	На основі механізмів розширення «Kubernetes» було розроблено «Zeus». Роботи з найкращими зусиллями плануються «Zeus» на основі фактичного використання сервера. Завдяки координації апаратних і програмних елементів ізоляції «Zeus» покращує ізоляцію контейнера.

У [41], автори пропонують планувальник для розподіленого глибокого навчання в «Kubeflow», який поєднує в собі функції з існуючих робіт, включаючи автомасштабування та групове планування. Запропонований планувальник містить модифікації для підвищення ефективності навчального процесу, а вагові коефіцієнти використовуються для визначення пріоритетності завдань. Автори оцінюють запропонований планувальник за допомогою набору завдань TensorFlow і виявляють, що він підвищує швидкість навчання більш ніж на двадцять шість відсотків порівняно з планувальником «Kubernetes» за замовчуванням.

У [40], автори пропонують метод управління зі зворотним зв'язком для забезпечення еластичного контейнера в системах на основі «Kubernetes». Метод використовує комбінацію моделі черги зі змінною швидкістю обробки та лінійної моделі для підвищення точності вихідних помилок. Автори порівнюють свій підхід із кількома існуючими алгоритмами на кластері «Kubernetes» і виявляють, що він отримує найнижчий відсоток порушень угоди про рівень обслуговування і другу найнижчу вартість.

Алгоритми масштабування можна класифікувати з метою:

- Підвищення ефективності – зменшення витрат, збільшення використання ресурсів («CPU», «GPU», пам'ять, дисковий простір).
- Автоматичне масштабування – динамічне зміна розміру кластера залежно від навантаження.
- Оптимізація розміщення – ефективне розміщення контейнерів на вузлах кластера.
- Управління енергоспоживанням – зменшення споживання енергії за рахунок оптимізації роботи кластера.

1.5 Висновки до першого розділу

В першому розділі кваліфікаційної роботи подано визначення концепту «розумне місто». Висвітлено означення «Kubernetes» та його використання для «розумних міст». Описана методика пошуку наукових публікацій щодо

алгоритмів планування «Kubernetes» для «розумних міст». Проаналізовано знайдені наукові публікації щодо алгоритмів планування «Kubernetes» для «розумних міст».

2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПЛАНУВАННЯ «KUBERNETES» ДЛЯ «РОЗУМНИХ МІСТ»

2.1 Планування за допомогою багатоцільової оптимізації

Багатоцільове планування оптимізації «Kubernetes» для «розумних міст» враховує декілька цілей або критеріїв, коли вирішує, як розподіляти ресурси та планувати контейнери на вузлах у кластері. Цей підхід особливо корисний у складних розподілених системах, де є декілька конкуруючих цілей, які необхідно збалансувати для досягнення найкращої загальної продуктивності. У багатоцільовому підході до планування оптимізації планувальник розглядає декілька цілей одночасно, наприклад мінімізацію часу відповіді, максимізацію використання ресурсів і зменшення споживання енергії. Планувальник використовує алгоритми оптимізації, щоб знайти оптимальне рішення, яке збалансує ці цілі.

Багатоцільове планування оптимізації «Kubernetes» для «розумних міст» може допомогти підвищити загальну продуктивність і ефективність кластерів «Kubernetes» завдяки врахуванню декількох цілей під час розподілу ресурсів і планування контейнерів. Цей підхід може призвести до кращого використання ресурсів, підвищення продуктивності застосунків, зменшення споживання енергії та зниження витрат.

Деякі приклади багатоцільових алгоритмів планування оптимізації, які використовуються в «Kubernetes» для «розумних міст», включають генетичні алгоритми, оптимізацію мурашиної колонії та оптимізацію роїв частинок. Ці алгоритми можуть допомогти оптимізувати різні цілі: час відповіді, використання ресурсів, споживання енергії та інші фактори, щоб досягти найкращої загальної продуктивності та ефективності в кластері «Kubernetes».

У [42], автори пропонують новий контролер для керування контейнерами на периферійних хмарних вузлах у системах «ІІоТ». Контролер «Kubernetes-based energy and interference driven scheduler» («KEIDS») базується на «Google Kubernetes» і розроблений для мінімізації споживання енергії та перешкод у

системах «IoT». «KEIDS» використовує цілочисельне лінійне програмування, щоб сформулювати задачу планування завдань як багатоцільову задачу оптимізації, враховуючи фактори: споживання енергії, викиди вуглецю та перешкоди від інших програм. Автори оцінюють «KEIDS», використовуючи дані в режимі реального часу з обчислювальних кластерів «Google», і виявляють, що він перевершує існуючі найсучасніші схеми.

У [43], автори пропонують багатоцільову модель оптимізації для планування мікросервісів на основі контейнерів у хмарних архітектурах. Вони подають алгоритм мурашиної колонії для вирішення завдань планування, який враховує фактори: використання обчислювальних ресурсів і ресурсів зберігання, кількість запитів на мікросервіси та частоту відмов фізичних вузлів. Автори оцінюють запропонований алгоритм за допомогою експериментів і порівнюють його продуктивність з іншими схожими алгоритмами. Вони виявили, що запропонований алгоритм досягає кращих результатів з точки зору надійності служби кластера, балансування навантаження кластера та накладних витрат на передачу по мережі.

У [44], автори пропонують вдосконалений алгоритм планування для «Kubernetes» шляхом поєднання оптимізації колонії мурашок і оптимізації роїв частинок для кращого балансу розподілу завдань і зменшення витрат на ресурси. Подана технологія може бути ефективно використана в контексті «розумних міст», де оптимізація ресурсів і розподіл завдань є критично важливими для забезпечення ефективної роботи інфраструктури. Автори реалізували алгоритм на «Java» та протестували його за допомогою інструменту «CloudSim», показавши, що він перевершує вихідний алгоритм планування.

У роботі [45] автор обговорює ідею розміщення та міграції контейнерів на крайніх серверах, а також створені для цього моделі планування. Більшість моделей планування, на думку автора, базуються здебільшого на евристичних алгоритмах і використовують моделі багатоцільової оптимізації або моделі графових мереж. Дослідження також вказує на відсутність досліджень щодо моделей планування контейнерів, які враховують розосереджені граничні

обчислення, і передбачає, що майбутні дослідження в цій галузі будуть зосереджені на плануванні контейнерів для мобільних граничних вузлів.

У [46], автори пропонують доповнення до планувальника «Kubernetes», яке використовує вимірювання якості досвіду «QoE», щоб допомогти хмарним управлінням цільовими рівнями обслуговування «SLO» бути більш точними. У контексті служб потокового відео, які розташовані разом з іншими службами, автори оцінюють запропоновану архітектуру за допомогою метрики «QoE». Відповідно до висновків, перепланування ресурсів збільшує середній «QoE» на сто тридцять п'ять відсотків, тоді як запропонований планувальник збільшує його на п'ятдесят відсотків порівняно з іншими планувальниками.

Алгоритми масштабування можна класифікувати з метою управління контейнерами та плануванням ресурсів (див. таблицю 2.1).

Таблиця 2.1 – Характеристики алгоритмів масштабування для управління контейнерами та планування ресурсів

[31] Розроблено керований політикою метапланувальник для кластерів «Kubernetes», який забезпечує ефективний і справедливий розподіл ресурсів для декількох користувачів.	Політика справедливості домінуючих ресурсів («DRF») Додаткові показники справедливості на основі попиту на ресурси завдання та середнього часу очікування.
[37] Зроблено планувальник «Kubernetes» ефективнішим, включивши навантаження дискового вводу-виводу.	Щоб покращити баланс дискового вводу-виводу між вузлами, був запропонований підхід динамічного планування «Balanced-Disk-IOPriority» («BDI»).
[41] Подано розширення для перепланування ресурсів і планувальника «Kubernetes», яке поєднує показники «QoE» у «SLO».	Використано метрику «QoE». Оцінено архітектуру за допомогою служб потокового відео, розташованих разом з іншими службами.

2.2 Планування, орієнтоване на «AI»

Багато великих компаній нещодавно почали надавати послуги на основі «AI». З цією метою вони встановили машинні кластери глибокого навчання, що складаються з десятків до тисяч процесорів і графічних процесорів для розподіленого навчання своїх моделей глибокого навчання. Використовуються різні фреймворки машинного навчання: «MXNet» [47], «TensorFlow» [48] і «Petuum» [49]. Навчання моделі глибокого навчання зазвичай потребує ресурсів і часу. У даних умовах ефективне планування має вирішальне значення для повного використання дорогого кластера глибокого навчання та прискорення процесу навчання моделі. Для планування завдань на цій арені використовувалися різні стратегії, наприклад, планувальники загального призначення налаштовані для вирішення завдань розподіленого глибокого навчання, наприклад [50] і [51]; однак вони статично розподіляють ресурси та не коригують ресурси за різних умов навантаження, що призводить до поганого використання ресурсів. Інші запропонували динамічний розподіл ресурсів після ретельного аналізу робочих навантажень, наприклад [52] і [53].

У [53], автори пропонують індивідуальний планувальник завдань для кластерів глибокого навчання «Optimus». Мета «Optimus» – звести до мінімуму час, необхідний для роботи з навчання глибокого навчання, яка вимагає ресурсів і часу. «Optimus» використовує моделі продуктивності для точної оцінки швидкості навчання як функції розподілу ресурсів і онлайн-підбірки, щоб передбачити конвергенцію моделі під час навчання. Ці моделі показують, як «Optimus» динамічно організовує завдання та розподіляє ресурси, щоб скоротити час виконання завдань. Автори застосували «Optimus» на практиці на кластері глибокого навчання та оцінили його ефективність у порівнянні з іншими планувальниками кластерів. Вони виявили, що «Optimus» перевершує звичайні планувальники з точки зору часу виконання завдань і часу виконання приблизно на сто тридцять дев'ять відсотків і шістдесят три відсотки відповідно.

У [54], автори пропонують використовувати сучасні методи машинного навчання для розробки високоефективних політик для планування завдань

обробки даних у розподілених обчислювальних кластерах. Вони подають свою систему «Decima», яка використовує навчання з підкріпленням і нейронні мережі для вивчення алгоритмів планування, що залежать від робочого навантаження. «Decima» розроблено таким чином, щоб бути масштабованим і здатним обробляти складні графіки залежностей завдань. Автори повідомляють, що їх прототип інтеграції з «Spark» на двадцятип'ятивузловому кластері покращив середній час виконання завдання щонайменше на двадцять один відсоток порівняно з існуючими ручними налаштуваннями евристики планування, з покращенням до два разів у періоди високого навантаження на кластер.

У [55] автори подали розподілений планувальник справедливої частки для кластерів «GPU», який використовується для навчання глибокого навчання «Gandivafair». Ця система використання кластерів «GPU» забезпечує ізоляцію продуктивності між користувачами та створена для досягнення балансу між конкуруючими вимогами справедливості та ефективності. Незважаючи на неоднорідність кластерів, «Gandivafair» є першим планувальником, який справедливо розподіляє час «GPU» між усіма активними користувачами. Автори демонструють, що «Gandivafair» забезпечує як справедливість, так і ефективність за реалістичних багатокористувацьких навантажень, оцінюючи його за допомогою реалізації прототипу на гетерогенному кластері з двомастами «GPU».

У [56], автори пропонують нову схему розміщення контейнерів «ProCon» для планування завдань у кластері «Kubernetes». «ProCon» використовує оцінку майбутнього використання ресурсів, щоб збалансувати суперечки щодо ресурсів у кластері та скоротити час виконання та тривалість виконання завдань. Автори демонструють за допомогою експериментів, що «ProCon» скорочує час виконання на п'ятдесят три цілих три десятих відсотка для конкретної роботи та підвищує загальну продуктивність на двадцять три відсотка. Крім того, «ProCon» демонструє підвищення ефективності до тридцяти семи цілих чотирьох десятих відсотка порівняно з вбудованим планувальником «Kubernetes» за умовчанням. У «розумних містах» широко використовуються аналітичні системи, та хмарні обчислення. Для ефективного управління такими системами потрібне

оптимальне використання обчислювальних ресурсів у кластері. «ProCon», завдяки балансу використання ресурсів, може забезпечити стабільну роботу критичних сервісів: моніторинг трафіку, екстрені служби або енергетичні системи.

У таблиці 2.2 подано характеристики алгоритмів планування для гетерогенних кластерів і периферійних платформ.

Таблиця 2.2 – Характеристики алгоритмів планування для гетерогенних кластерів і периферійних платформ

Цілі	Методологія/Алгоритми
[34] Створено динамічний планувальник «Kubernetes», щоб допомогти гетерогенному кластеру ефективніше розгортати контейнери «Docker». Використано попередні дані про виконання контейнера, щоб прискорити виконання завдання.	Розробив платформу динамічного планування «KubCG». Подано новий планувальник, який враховує минулі дані про виконання контейнера, а також розклад для «Kubernetes Pods».
[36] Розроблено і запущено «KubeNICE», оркестратор контейнерів для гетерогенних архітектур «ISA» на хмарних периферійних платформах. Оцінено ефективність і продуктивність «KubeNICE» при обробці гетерогенних кластерів «ISA»	Використовуючи «AIM» і «PAS», «KubeNICE» розширює «Kubernetes» з відкритим кодом. «AIM» автоматично знаходить вузол, який підходить для «ISA», які підтримує контейнерна програма. «PAS» планує контейнери на основі обчислювальної потужності вузлів кластера

У [57], автори пропонують «DL2», заснований на глибокому навчанні планувальник для кластерів глибокого навчання, який спрямований на покращення глобальної експедиції навчання шляхом динамічної зміни розміру ресурсів, виділених на завдання. Автори впроваджують «DL2» на «Kubernetes» і оцінюють його продуктивність у порівнянні з планувальником справедливості та

експертним евристичним планувальником. Результати показують, що «DL2» перевершує інші планувальники з точки зору середнього часу виконання завдання.

У [58], автори пропонують новий планувальник контейнерів «SpreCon», оптимізований для короткострокових програм глибокого навчання. «SpreCon» розроблено для покращення використання ресурсів і часу виконання завдань у кластері «Kubernetes» шляхом аналізу прогресу процесів навчання глибокого навчання та спекулятивної міграції повільно зростаючих моделей, щоб вивільнити ресурси для тих, що розвиваються швидше. Автори проводять експерименти, які демонструють, що «SpreCon» покращує час виконання індивідуальних завдань на сорок одну цілу п'ять десятих відсотка, підвищує продуктивність усієї системи на чотирнадцять цілих вісім десятих відсотка і скорочує термін виконання на двадцять чотири цілих сім десятих відсотка.

У [59], для планування незалежних пакетних завдань у багатьох об'єднаних кластерах хмарних обчислень автори пропонують планувальник завдань на основі глибокого підсилення, який отримав назву «RLSK». Автори запустили «RLSK» у «Kubernetes» і перевірили його продуктивність за допомогою моделювання, продемонструвавши, що він може перевершити звичайні методи планування.

Робота [60] описує «MLFS», засновану на функціях систему планування завдань для кластерів машинного навчання, які можуть виконувати паралельні процеси як даних, так і моделі. Щоб визначити пріоритет завдання для впорядкування робочої черги, «MLFS» використовує евристичний метод планування. Потім дані цього методу використовуються для навчання моделі глибокого навчання з підкріпленням для планування роботи. У порівнянні з існуючими робочими графіками запропонована система показала, що вона скорочує час виконання робіт до п'ятдесяти трьох відсотків, тривалість виконання робіт до п'ятдесяти двох відсотків і підвищує точність до шістдесяти чотирьох відсотків. Система тестується за допомогою реальних експериментів і масштабного моделювання на основі реальних слідів.

У [61], завдання планування «Kubernetes» orchestrator розподіляється між вузлами обробки запропонованим авторами спеціальним планувальником, який використовує багатоагентну систему. За словами авторів, цей метод є швидшим, ніж стратегія централізованого планування, яка використовується стандартним планувальником «Kubernetes».

У таблиці 2.3 подано характеристику алгоритмів масштабування та планування ресурсів у «Kubernetes» для спеціалізованих сценаріїв.

Таблиця 2.3 – Характеристики алгоритмів масштабування та планування ресурсів у Kubernetes для спеціалізованих сценаріїв

Цілі	Методологія/Алгоритми
[24] Щоб покращити функціональність і забезпечити рівноправність користувачів у спільному кластері зі змінними апаратними ресурсами для фреймворків глибокого навчання.	Двостороння відповідність мінімальної вартості.
[29] Подано механізм планування «Kubernetes» «GPU» на основі топології. Збільшено ефективність використання ресурсів і розподіл навантаження в кластері «GPU».	Основою системи є механізм планування «Kubernetes GPU», який використовує дерево вартості доступу для відновлення топології кластера «GPU» та адаптації сценаріїв використання ресурсів.
[33] Щоб покращити планування «Kubernetes» для чутливих до продуктивності контейнерних робочих процесів, зокрема, в контексті граничних програм 5G	«NetMARKS» – це передовий метод для планування модулів «Kubernetes», який використовує переваги динамічних мережевих показників, зібраних за допомогою «Istio Service Mesh».

У [62], автори подають «KaiS», межову хмарну структуру планування «Kubernetes» на основі навчання. «KaiS» моделює дані про стан системи за допомогою графових нейронних мереж і скоординованого багатоагентного актор-критичного методу для децентралізованої відправки запитів. Дослідження показують, що порівняно з базовими показниками «KaiS» може збільшити середню пропускну здатність системи на чотирнадцять цілих три десятих відсотка і зменшити витрати на планування на тридцять чотири цілих сім десятих відсотка.

У [63], автори пропонують метод оптимізації алгоритму планування контейнерів «Kubernetes» шляхом поєднання теорії сірої системи з методом прогнозування нейронної мережі «Long Short-Term Memory». Вони проводять експерименти, щоб оцінити свій підхід і виявити, чи він може зменшити недолік фрагментації ресурсів робочих вузлів у кластері та збільшити використання ресурсів кластера.

У [64], автори пропонують високомасштабовану кластерну систему планування для «Kubernetes», яка називається «Zeus». Основна особливість «Zeus» полягає в тому, що на основі фактичного використання сервера він планує завдання з найкращим зусиллям. Він має здатність адаптивно розподіляти ресурси між навантаженнями двох різних класів. «Zeus» призначений для забезпечення безпечного спільного розміщення найкращих процесів і служб, чутливих до затримок. Автори перевірили «Zeus» у реальних умовах і виявили, що він може підвищити середнє використання «CPU» з п'ятнадцяти до шістдесяти відсотків без порушення цілей рівня обслуговування «SLO». Ця технологія має значний потенціал для інтеграції в інфраструктури розумних міст, де важливим є ефективне планування та управління ресурсами в умовах високої динамічності і різноманіття запитів.

У [65], автори пропонують стратегію планування для завдань глибокого навчання на «Kubernetes», яка враховує характеристики використання ресурсів завдань. Щоб підвищити ефективність виконання завдань і балансування навантаження, запропонована парадигма, яка отримала назву «FBSM», містить модулі для аналізатора графічного процесора та планувальника з урахуванням

балансу. Згідно з оцінкою авторів, виконання завдань глибокого навчання прискорюється запропонованою системою, відомою як «KubFBS», яка також розкриває покращені можливості балансування навантаження для кластера.

У [66], автори пропонують рішення для розподілу ресурсів у мережевій службі розміщення інфраструктури «Kubernetes». Пропоноване рішення має на меті уникнути дефіциту ресурсів і захистити найважливіші функції. Також автори використовують статистичний підхід для моделювання та вирішення задач, враховуючи випадковий характер обробленої інформації. Це рішення має великий потенціал для впровадження в розумних містах, де потрібно оптимізувати використання ресурсів в умовах високої динамічності та різноманіття запитів.

2.3 Планування з підтримкою автомасштабування

Автоматичне масштабування є важливою функцією планування «Kubernetes» для «розумних міст», оскільки воно дає змогу автоматично коригувати ресурси, виділені для модулів, на основі поточного попиту. Це забезпечує ефективне використання ресурсів, підвищення продуктивності, економію коштів і високу доступність програми. Автоматичне змінення масштабу та планування пов'язані між собою тим, що автоматичне змінення масштабу можна використовувати, щоб гарантувати, що завжди достатньо ресурсів для виконання запланованих завдань. Наприклад, якщо планувальник призначає нове завдання робочому вузлу, але цей вузол не має достатньо ресурсів для виконання завдання, автоматичне масштабування може додати більше ресурсів до цього вузла або створити новий вузол для обробки завдання. Таким чином, автоматичне масштабування та планування працюють разом, щоб гарантувати, що розподілена система здатна обробляти мінливі навантаження та оптимізувати використання ресурсів. Нижче наведено огляд деяких схем, пов'язаних із цією категорією.

У [67], автори пропонують нову систему керування ресурсами «RUBAS», яка може динамічно регулювати розподіл контейнерів, що працюють у кластері

«Kubernetes». «RUBAS» включає міграцію контейнерів для покращення системи «Kubernetes Vertical Pod Autoscaler» без збоїв. Автори оцінюють «RUBAS» за допомогою декількох наукових тестів і порівнюють його продуктивність з «Kubernetes Vertical Pod Autoscaler». Вони виявили, що «RUBAS» покращує використання процесора та пам'яті на десять відсотків і скорочує час роботи на п'ятнадцять відсотків із накладними витратами для кожної програми в діапазоні від п'яти до двадцяти відсотків.

У [68], автори подають механізм масштабування «Kubernetes», який використовує методи прогнозування машинного навчання для прийняття кращих рішень щодо автоматичного масштабування для хмарних програм. Короткочасний цикл оцінки механізму надає змогу йому адаптуватися до динаміки запитів, що змінюється, і автори вводять компактний параметр керування для хмарних орендарів, щоб легко встановити бажаний рівень надмірного надання ресурсів у порівнянні з порушеннями угоди про рівень обслуговування. Запропонований механізм оцінюється в симуляції та за допомогою вимірювань даних веб-трафіків, і результати показують, що він призводить до меншої кількості втрачених запитів і дещо більшого обсягу ресурсів порівняно з базовим сценарієм «Kubernetes» за замовчуванням.

У [69], автори пропонують новий динамічний багаторівневий («DM») метод автомасштабування для контейнерних хмарних програм. Метод «DM» використовує дані моніторингу як на рівні інфраструктури, так і на рівні програми, щоб визначити, коли масштабувати вгору чи вниз, а його порогові значення динамічно коригуються залежно від умов робочого навантаження. Автори порівнюють продуктивність методу «DM» із сімома існуючими методами автомасштабування, використовуючи синтетичні та реальні навантаження. Вони виявили, що метод «DM» має кращу загальну ефективність, ніж інші методи, особливо з точки зору часу відповіді та кількості створених контейнерів. Система «SWITCH» була використана для реалізації методу «DM» для хмарних застосунків, критичних до часу.

У таблиці 2.4 подано характеристику алгоритмів планування та оптимізації для інфраструктури «Smart City».

Таблиця 2.4 – Характеристики алгоритмів планування та оптимізації для інфраструктури «Smart City»

Цілі	Методологія/Алгоритми
[22] Розроблено стратегію планування для програм на основі контейнерів у розгортаннях «Smart City», яка підтримує мережу.	Як додаток до стандартного механізму планування, вбудованого в «Kubernetes», запропоновано та застосовано на практиці метод планування з урахуванням мережі.
[32] Змінено «Kubernetes», щоб краще підходити для периферійної інфраструктури, зосередившись на затримці мережі та можливостях самовідновлення	Спеціальний планувальник «Kubernetes», який враховує обмеження затримки застосунків і граничну надійність

У [70], автори пропонують адаптивний автомасштабувальник «Libra», який автоматично визначає оптимальний набір ресурсів для окремого пакета та керує процесом горизонтального масштабування. «Libra» також може адаптувати визначення ресурсу для модуля та регулювати процес горизонтального масштабування, якщо змінюється навантаження або базове віртуалізоване середовище. Автори оцінюють «Libra» за допомогою симуляції та показують, що вона може зменшити середнє використання процесора та пам'яті до сорока восьми відсотків та тридцяти дев'яти відсотків відповідно порівняно зі стандартним автомасштабувальником «Kubernetes».

В іншій роботі [68], автори пропонують механізм масштабування «Kubernetes», який використовує численні методи прогнозування на основі штучного інтелекту для прийняття рішень щодо автоматичного масштабування, які краще підходять для обробки мінливості вхідних запитів. Автори також вводять компактний параметр керування, щоб допомогти постачальникам застосунків легко налаштувати бажане надлишкове надання ресурсів. Пропонований механізм оцінюється в симуляції та за допомогою вимірювань на

веб-трафіках, що демонструє покращене пристосування наданих ресурсів до попиту на обслуговування.

У [71] автори пропонують покращену схему автоматичного масштабування для «Kubernetes», яка поєднує в собі переваги різних типів вузлів у процесі масштабування. Вони виявили, що їхня схема покращує продуктивність системи під високим тиском навантаження та зменшує нестабільність у запусчених кластерах порівняно з автоматичним масштабувальником за замовчуванням.

Автори [71] пропонують новий активний пристрій автоматичного масштабування «Kubernetes» на основі передбачення реплік контейнерів. Вони демонструють, що запропонований ними автомасштабувальник має вищу швидкість відгуку порівняно з існуючими стратегіями масштабування в «Kubernetes».

У [72], автори пропонують метод підвищення надійності віртуальних мереж за допомогою моделей оптимізації та евристичних алгоритмів для розподілу функцій віртуальної мережі у відповідних місцях. Автори також розробляють плагіни планувальника функцій для системи «Kubernetes», що надає можливість автоматично розгортати та керувати контейнерними програмами. Показано, що запропонований метод ефективний у розподілі функцій і правильному виконанні службових функцій. Ця робота була опублікована у випуску конференції «IEEE» з прийняття рішень та контролю за 2021 рік.

У [73], пропонують гібридний метод автомасштабування для контейнерних застосунків, який поєднує можливості вертикального та горизонтального масштабування для оптимізації використання ресурсів і забезпечення вимог до якості обслуговування («QoS»). Запропонований метод використовує прогнозний підхід на основі машинного навчання для прогнозування майбутнього попиту та модуль ідентифікації пакетів для прийняття рішень щодо масштабування. Автори оцінюють запропонований метод і виявляють, що він покращує час відгуку та використання ресурсів порівняно з існуючими методами, які використовують лише один режим

масштабування. Цей підхід має потенціал для застосування в інфраструктурах «розумних міст», де ефективне масштабування та управління ресурсами є ключовими для забезпечення високої якості обслуговування в умовах швидко змінюваного попиту, що є характерним для смарт-систем, що використовують «Kubernetes».

У таблиці 2.5 подано характеристики алгоритмів масштабування для підвищення ефективності та «QoS».

Таблиця 2.5 – Характеристики алгоритмів масштабування для підвищення ефективності та «QoS»

Цілі	Методологія/Алгоритми
[27] Підвищення ефективності центрів обробки даних. Врахування віртуальної та фізичної інфраструктури та бізнес-процесів у плануванні.	Планувальник «Kubernetes» за замовчуванням замінено на запропоновану всеохоплюючу структуру планування. Планувальник розглядає модель програмного та апаратного забезпечення. Система була розгорнута в центрі обробки даних.

2.4 Висновок до другого розділу

В другому розділі кваліфікаційної роботи описано основні принципи роботи з алгоритмами планування в «Kubernetes» для «розумних міст». Досліджено використання «Kubernetes» для забезпечення ефективності та автоматизації в інфраструктурах, що потребують обробки великих обсягів даних: системи моніторингу навколишнього середовища та транспортні мережі. Подано порівняльний опис підходів до планування, зокрема щодо управління ресурсами та оркестрації контейнерів. Визначено потенціал «Kubernetes» для інтеграції з технологіями «розумних міст», що надає змогу оптимізувати управління ресурсами та знижувати витрати на обслуговування інфраструктури.

3 АНАЛІЗ ПІДХОДІВ ДО МАСШТАБУВАННЯ ТА ПРОЄКТУВАННЯ РІШЕНЬ У «KUBERNETES» ДЛЯ «РОЗУМНИХ МІСТ»

3.1 Аналіз сучасних підходів до масштабування в «Kubernetes»

Масштабування є важливою складовою управління ресурсами в «Kubernetes», адже воно забезпечує можливість адаптувати кількість виділених ресурсів до поточного навантаження на систему. Існує декілька сучасних підходів до автоматичного масштабування в «Kubernetes», які забезпечують різні рівні оптимізації використання ресурсів і продуктивності застосунків. Ці підходи можуть діяти як на рівні окремих контейнерів, так і на рівні вузлів. [74]

«HRA» є одним із найпоширеніших інструментів для масштабування в «Kubernetes». Цей алгоритм дає можливість автоматично змінювати кількість подів на основі заданих метрик, завантаження «CPU», використання «RAM», а також спеціальні метрики, які можна отримувати через Prometheus або інші системи моніторингу.

- Принцип роботи: «HRA» періодично перевіряє значення метрик і порівнює їх з установленими пороговими значеннями. Якщо навантаження на поди перевищує поріг, система автоматично створює додаткові поди для обробки навантаження. Якщо навантаження падає, кількість подів зменшується.

- Переваги: Гнучкість і швидка реакція на змінні умови навантаження. Підходить для застосунків, які можуть масштабуватися горизонтально (додаючи більше інстансів).

- Недоліки: Не завжди оптимально працює для застосунків з інтенсивним використанням ресурсів, де важливою є продуктивність одного пода, а не кількість.

«VPA» дає можливість автоматично змінювати виділені ресурси («CPU», «RAM») для кожного пода в залежності від фактичних потреб. Це корисно для застосунків, де горизонтальне масштабування може бути неефективним, або де важливіший збалансований розподіл ресурсів між подами.

– Принцип роботи: «VPA» аналізує споживання ресурсів подами та динамічно коригує їх обмеження ресурсів (ресурсні ліміти та запити). Це забезпечує можливість зменшити витрати на ресурси для малих застосунків та уникнути перевантаження для ресурсомістких подів.

– Переваги: Оптимізація використання ресурсів для складних застосунків, які не можуть масштабуватися горизонтально або для яких важлива продуктивність одного екземпляра.

– Недоліки: Під час зміни ресурсів може знадобитися перезапуск подів, що може негативно вплинути на доступність сервісів. Крім того, цей підхід не підходить для масштабування систем з високою кількістю незалежних подів.

«Cluster Autoscaler» працює на рівні інфраструктури та автоматично додає або видаляє вузли в кластері в залежності від поточних вимог до ресурсів. Це особливо актуально, коли горизонтальне та вертикальне масштабування подів не достатньо для забезпечення продуктивності системи.

– Принцип роботи: Якщо в кластері не вистачає ресурсів для запуску нових подів або якщо виділені вузли не використовуються на повну потужність, Cluster Autoscaler автоматично додає або видаляє вузли для оптимального використання інфраструктури.

– Переваги: Забезпечує динамічне управління фізичними або віртуальними ресурсами кластера, що забезпечує більш гнучко реагувати на зміни в навантаженні.

– Недоліки: Час запуску або вимкнення вузлів може бути значним, особливо у великих хмарах. Це також може впливати на час, необхідний для стабілізації кластера.

Комбіновані підходи, поєднання «HPA», «VPA», «Cluster Autoscaler». У багатьох випадках застосовуються комбіновані рішення, коли використовуються одразу декілька типів масштабування. Наприклад, «HPA» може використовуватися для масштабування подів на основі навантаження, тоді як «VPA» коригує обсяг ресурсів, які виділяються кожному поду. Паралельно Cluster Autoscaler може додавати або видаляти вузли для забезпечення достатньої кількості ресурсів у кластері.

– Переваги: Поєднання різних підходів дає змогу досягти оптимального розподілу ресурсів, забезпечуючи високу продуктивність та ефективність при різних типах навантажень.

– Недоліки: Складність у налаштуванні та взаємодії різних механізмів масштабування. Некоректне налаштування може призвести до непередбачуваних результатів, зокрема, надмірне споживання ресурсів або недоліки з доступністю застосунків.

Підхід на основі метрик користувача. Окрім стандартних метрик, «Kubernetes» дає змогу використовувати кастомні метрики для масштабування застосунків. Це підхід, який забезпечує більшу гнучкість, оскільки користувачі можуть налаштувати масштабування на основі показників, специфічних для застосунків.

– Переваги: Велика гнучкість та можливість налаштування масштабування під конкретні потреби застосунків.

– Недоліки: Потребує додаткових налаштувань системи моніторингу та збору метрик, що може ускладнити підтримку системи.

Сучасні підходи до масштабування в «Kubernetes» забезпечують можливість гнучко адаптувати ресурси до змін у навантаженні, використовуючи як горизонтальне, так і вертикальне масштабування, а також масштабування на рівні кластера. Вибір конкретного підходу або їх поєднання залежить від вимог застосунків та навантаження, яке вони обробляють, і має бути оптимізований для кожного конкретного випадку.

3.2 «Horizontal Pod Autoscaler»

«Horizontal Pod Autoscaler» – це один із основних інструментів для автоматичного масштабування застосунків у «Kubernetes». Він дає змогу змінювати кількість подів (екземплярів контейнерів) у залежності від поточного навантаження на систему. «HPA» є ключовим елементом для забезпечення еластичності застосунків, коли навантаження зростає або падає, надаючи можливість динамічно масштабувати ресурси відповідно до потреб.

На рисунку 3.1 зображено «Horizontal Pod Autoscaler», що базується на постійному спостереженні та моніторингу показників продуктивності подів. Основою для прийняття рішень щодо масштабування служать стандартні метрики: завантаження «CPU» та використання пам'яті. Наприклад, завантаження «CPU» визначає, скільки відсотків від загального обсягу виділеного процесорного часу використовують поди. Використання «RAM» показує обсяг пам'яті, споживаний кожним подом. Окрім стандартних метрик, «HPA» також підтримує кастомні метрики, які можуть бути специфічними для певних застосунків, як-от кількість активних запитів до веб-сервера або розмір черги обробки повідомлень.

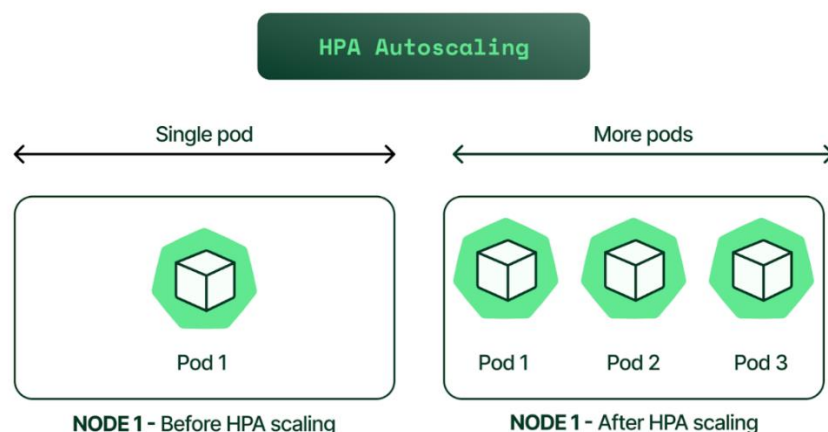


Рисунок 3.1 – Відображення «HPA» Autoscaling [75]

Процес роботи «HPA» починається зі збору метрик. «HPA» періодично отримує дані про стан навантаження на поди через «API» метрик. Для цього зазвичай використовується «metrics-server» для стандартних показників, або такі інструменти, як «Prometheus», для кастомних метрик. Ці дані надають повну інформацію про поточний стан системи, яку використовує «HPA» для прийняття рішень щодо масштабування.

Далі відбувається оцінка відповідності поточних метрик із налаштованими пороговими значеннями. Наприклад, якщо середнє завантаження «CPU» на всіх подах перевищує встановлену межу, це означає, що застосунок працює під надмірним навантаженням, і необхідно додати більше подів для забезпечення

стабільної роботи. Якщо ж навантаження зменшується, «HРА» відповідно знижує кількість подів.

Останнім етапом є саме масштабування. Коли метрики перевищують або не досягають порогових значень, «HРА» плавно збільшує або зменшує кількість подів залежно від поточної ситуації. Це масштабування відбувається поступово, щоб уникнути різких змін, які могли б призвести до нестабільної роботи застосунка. Такий підхід забезпечує динамічну адаптацію системи до змін навантаження в режимі реального часу.

Підсумовуючи, «HРА» є ефективним інструментом для автоматичного горизонтального масштабування в «Kubernetes». Він добре підходить для застосунків з динамічними змінами навантаження, забезпечуючи гнучкість і ефективність використання ресурсів. Проте для повного охоплення всіх потреб масштабування часто необхідне комбінування «HРА» з іншими механізмами, такими як «VPA» або «Cluster Autoscaler».

3.3 «Vertical Pod Autoscaler»

«Vertical Pod Autoscaler» є механізмом автоматичного масштабування в «Kubernetes», орієнтованим на вертикальне масштабування. Це означає, що замість збільшення кількості подів, «VPA» збільшує або зменшує ресурси («CPU» і «RAM»), виділені для окремого пода. «VPA» аналізує поточне використання ресурсів пода та на основі зібраних даних автоматично коригує його ресурсні запити та ліміти.

На рисунку 3.2 зображено «Vertical Pod Autoscaler» (VPA), що реалізує вертикальне масштабування через декілька ключових етапів. Першим етапом є моніторинг використання ресурсів. «VPA» регулярно перевіряє метрики: використання «CPU» і пам'яті для кожного пода. Цей постійний моніторинг надає змогу зібрати точні дані про те, скільки ресурсів потребує кожен под, що є критично важливим для подальших коригувань.

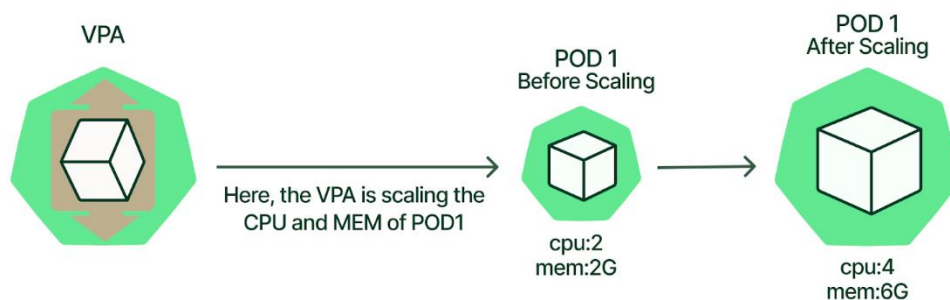


Рисунок 3.2 Відображення «VPA» Autoscaling [76]

Другий етап – це оцінка потреб у ресурсах. На основі зібраних метрик «VPA» прогнозує оптимальні значення для ресурсних запитів і лімітів. Ці розрахунки враховують як історичні дані, так і поточне навантаження, що забезпечує можливість точніше визначити, скільки ресурсів потрібно кожному поду.

Наступним етапом є оновлення ресурсних запитів і лімітів. «VPA» автоматично коригує конфігурації ресурсів для подів. Якщо поточне використання ресурсів перевищує запити або ліміти, «VPA» збільшує відповідні параметри; якщо ресурси не використовуються в повному обсязі, параметри зменшуються. Це допомагає адаптуватися до змінних потреб застосунка.

Останній етап – перезавантаження подів. Щоб нові конфігурації ресурсів вступили в силу, «VPA» може ініціювати перезавантаження подів. Це може включати зупинку поточного пода і запуск нового з оновленими ресурсними запитами. Перезавантаження може бути необхідним для застосування нових налаштувань, але може також вплинути на доступність застосунка.

Використання «VPA» може мати позитивний вплив на стабільність системи. Коригування ресурсних запитів і лімітів відповідно до реальних потреб сприяє уникненню ситуацій, коли поди недостатньо забезпечені ресурсами або мають надмірні ресурси, які не використовуються ефективно. Це сприяє зниженню ризику перевантаження системи та покращує загальну ефективність використання ресурсів.

Однак, «VPA» може також вплинути на стабільність системи через необхідність перезавантаження подів. Часті перезавантаження можуть призвести

до тимчасових перерв у роботі застосунка або зниження його доступності. Для мінімізації цього впливу важливо налаштувати «VPA» так, щоб перезавантаження відбувалося максимально плавно і не впливало негативно на користувачів.

«Vertical Pod Autoscaler» є потужним інструментом для оптимізації ресурсів у «Kubernetes» шляхом вертикального масштабування подів. Він надає можливість динамічно коригувати ресурсні запити та ліміти, що може покращити ефективність використання ресурсів і стабільність системи. Проте, впровадження «VPA» може бути пов'язане з певними складнощами, такими як необхідність перезавантаження подів і інтеграція з іншими механізмами масштабування. Тому важливо ретельно налаштувати «VPA» і моніторити його роботу, щоб забезпечити оптимальні результати для конкретних застосунків і систем.

3.4 Обговорення, виклики та перспективи подальших досліджень

У сфері багатоцільового планування на основі оптимізації в «Kubernetes» для потреб «розумних міст» було проведено декілька досліджень для оптимізації різних цілей, серед яких: мінімізація споживання енергії та витрат при максимальному використанні ресурсів і дотриманні вимог до продуктивності програми. Ці дослідження використовують різні методи оптимізації: генетичні алгоритми, оптимізація рою частинок і оптимізація колонії мурашок. Деякі дослідження також включають підходи на основі машинного навчання для прогнозування моделей робочого навантаження та прийняття рішень щодо планування. Є ще декілька завдань, які потрібно вирішити. По-перше, багатоцільовий характер задачі створює значну проблему для пошуку оптимальних рішень, які врівноважують суперечливі цілі. По-друге, динамічний характер хмарного середовища вимагає адаптації рішень щодо планування в режимі реального часу до мінливих умов. Загалом дослідження багатоцільового планування на основі оптимізації в «Kubernetes» для потреб «розумних міст» показують великий потенціал у досягненні ефективного та

ефективного управління ресурсами. Тим не менш, потрібна подальша робота, щоб вирішити задачі та перевірити ефективність цих підходів у реальних сценаріях.

З іншого боку, планування на основі штучного інтелекту в «Kubernetes» для потреб «розумних міст» було популярним напрямком досліджень останніми роками. Багато досліджень пропонують різні підходи до оптимізації рішень щодо планування за допомогою машинного навчання та інших методів «AI». Одним із ключових досягнень у цій галузі є розробка алгоритмів планування, які можуть обробляти складні робочі навантаження в динамічному середовищі. Ці алгоритми можуть враховувати різні фактори: доступність ресурсів, залежності завдань і вимоги до програми, щоб приймати оптимальні рішення щодо планування. У деяких дослідженнях були запропоновані алгоритми планування на основі підкріпленого навчання, які можуть адаптуватися до мінливих моделей робочого навантаження та вивчати досвід для покращення рішень щодо планування. Інші дослідження запропонували підходи, засновані на глибокому навчанні, які можуть фіксувати складні закономірності в даних про навантаження та робити точні прогнози. Загалом ці дослідження продемонстрували, що планування на основі «AI» може підвищити ефективність і продуктивність кластерів «Kubernetes» для потреб «розумних міст». Однак у цій сфері все ще існують деякі завдання, які необхідно вирішити. Однією з головних завдань є відсутність реальних наборів даних для навчання та оцінки алгоритмів планування на основі «AI». У більшості досліджень використовуються синтетичні або змодельовані набори даних, які можуть не відображати складності реальних робочих навантажень. Іншою задачею є обмін між точністю та обчислювальною складністю. Майбутні дослідження в цій галузі можуть бути зосереджені на розробці більш ефективних і масштабованих алгоритмів планування на основі штучного інтелекту, які можуть обробляти масштабні робочі навантаження. Це може включати вивчення нових методів машинного навчання та оптимізації, які можуть підвищити точність планування, одночасно зменшуючи складність обчислень.

Нарешті, планування з підтримкою автомасштабування – це нова область досліджень, яка спрямована на оптимізацію використання ресурсів і підвищення продуктивності застосунків шляхом поєднання методів автомасштабування та планування. За останні роки було опубліковано декілька досліджень у цій галузі. Аналіз цих досліджень показує, що планування з підтримкою автоматичного масштабування може призвести до значних покращень у використанні ресурсів і продуктивності застосунків. Дослідження показали, що вони можуть допомогти зменшити втрату ресурсів, мінімізувати ризик недостатнього забезпечення та скоротити час відповіді застосунків. Однак, незважаючи на ці багатообіцяючі результати, у цій сфері все ще є деякі завдання, які необхідно вирішити. Однією з головних задач є складність розробки ефективних алгоритмів планування з підтримкою автоматичного масштабування. Розробка алгоритмів, які можуть адаптуватися до динамічних змін робочого навантаження та оптимізувати використання ресурсів, зберігаючи при цьому продуктивність програми, є нетривіальним завданням. Крім того, існує потреба в додаткових дослідженнях практичної реалізації планування з підтримкою автоматичного масштабування в реальних сценаріях. Більшість існуючих досліджень було проведено в контрольованих експериментальних умовах, і існує потреба оцінити ефективність планування з підтримкою автоматичного масштабування в реальних програмах. Є ще декілька завдань, які необхідно вирішити, включаючи розробку алгоритму, стандартизацію та практичну реалізацію. Майбутні дослідження в цій галузі повинні зосередитися на вирішенні цих задач і розробці більш ефективних і практичних методів планування з підтримкою автоматичного масштабування.

У дослідницьких статтях використовуються різноманітні алгоритми для вдосконалення планування «Kubernetes» для потреб «розумних міст». Ці алгоритми тестуються на різних платформах і середовищах: «Spark», «MXNet», «Kubernetes», «Google» і кластер «GPU TwoSigma», робочі навантаження, обчислення «Google», «CPU-GPU», Національна хмарна інфраструктура, тести, «ProCon», «DL2», «DRF», «Optimus», «CBP», «PP», масштабування, центри обробки даних, планувальники, CloudSim і Java, сценарії, хмарна

інфраструктура, потреби користувачів, «RLSK», «GaiaGPU» і «Tencent», трасування робочого навантаження, моделювання та веб-трасування, «Kubernetes», новий алгоритм, перехід після відмови та відновлення «Kubernetes», «KubeNICE», сценарії реального світу, «BDI», «BCDI», «Kubernetes», запропонований алгоритм, автомасштабувальники, автомасштабувальники за замовчуванням, потокове відео, «Tensorflow», «Zeus» та служби, чутливі до затримки. У деяких документах не вказувалися деталі алгоритмів, які вони використовували, або платформ і середовищ, на яких вони тестувалися. Як видно з попередніх розділів, опитування широко аналізує поточну наукову літературу та створює таксономію, щоб не лише ефективно проаналізувати поточний стан, але й визначити виклики та майбутні напрямки. На основі аналізу було визначено потенційні перспективні дослідження в цій галузі:

- У міру того, як «Kubernetes» для потреб «розумних міст» стає все популярнішим, зростатиме потреба в передових методах оптимізації обчислень. У майбутньому «Kubernetes» для потреб «розумних міст» може отримати вигоду від розробки більш складних алгоритмів для планування робочого навантаження та розподілу ресурсів, потенційно з використанням «AI» або машинного навчання. Крім того, інтеграція «Kubernetes» для потреб «розумних міст» із такими новими технологіями, як безсерверні обчислення, може призвести до ще ефективнішого використання ресурсів завдяки динамічному масштабуванню без попередньої інфраструктури. Зрештою, майбутнє оптимізації обчислень у «Kubernetes» для потреб «розумних міст», ймовірно, включатиме поєднання передових алгоритмів, інноваційних технологій і поточного прогресу в хмарних обчисленнях.

- Тестування та впровадження для виявлення обмежень або поточних алгоритмів навчання для планування та потенційних покращень у великих кластерах. Одним із важливих акцентів є вдосконалення інструментів і автоматизації тестування та розгортання, включаючи розробку нових інфраструктур тестування та інтеграцію існуючих інструментів в екосистему «Kubernetes» для потреб «розумних міст». Іншою важливою сферою є постійне

вдосконалення процесу впровадження та розробки «Kubernetes» для потреб «розумних міст» із зосередженням на оптимізації робочих процесів, покращенні документації та сприянні тіснішій співпраці в рамках спільноти з відкритим кодом. Крім того, все більше уваги приділяється розробці більш повних стратегій тестування та перевірки кластерів «Kubernetes» для потреб «розумних міст», включаючи використання передових методів: інженерія хаосу, для моделювання сценаріїв збоїв у реальному світі. Загалом, майбутнє тестування та впровадження в «Kubernetes» для потреб «розумних міст», ймовірно, передбачатиме постійні інновації, співпрацю та постійну відданість просуванню платформи вперед. Ряд методів використовує алгоритми навчання для балансування ресурсів усередині та поза кластером. Незважаючи на те, що методи дали обнадійливі результати, можна знайти нові алгоритми навчання для покращення планувальника, особливо на великих кластерах.

– Обмеження та потенційні вдосконалення в конкретних контекстах, наприклад, екологічні обчислення. Мінімізація вуглецевого сліду кластера є постійним завданням. Необхідно запропонувати вдосконалені планувальники, щоб зменшити споживання енергії та вуглецевий слід кластерів у налаштуваннях «ІоТ». Існує величезна можливість для вдосконалення існуючих методів і пропозиції нових методів у цій галузі. Значна увага також має бути приділена інтеграції енергозберігаючих алгоритмів з адаптивними стратегіями розподілу навантаження для підвищення загальної ефективності систем.

– Управління ресурсами «Kubernetes» для потреб «розумних міст» здебільшого покладається на структуру моделювання оптимізації та евристичні алгоритми. Потенціал для вдосконалення та пропозиції нових алгоритмів управління ресурсами є дуже перспективним напрямком досліджень. Майбутні дослідження управління ресурсами «Kubernetes» для потреб «розумних міст» можуть зосередитися на вирішенні завдань управління складними динамічними навантаженнями в розподілених гетерогенних середовищах. Це може включати розробку більш складних алгоритмів і методів для розміщення робочого навантаження, розподілу ресурсів і балансування навантаження, а також вивчення нових підходів до контейнеризації та віртуалізації. Крім того, можуть

з'явитися можливості для використання нових технологій, зокрема, периферійних обчислень і мереж 5G, щоб забезпечити більш ефективне та масштабоване управління ресурсами в «Kubernetes» для потреб «розумних міст».

– Більшість роботи, виконаної в області планування «Kubernetes» для потреб «розумних міст», було оцінено на малих кластерах. Однак це не завжди може бути спокусливим. Одним з майбутніх напрямків досліджень планування «Kubernetes» для потреб «розумних міст» є використання більших розмірів кластерів для оцінки алгоритмів. Незважаючи на те, що «Kubernetes» для потреб «розумних міст» продемонстрував ефективність в управлінні кластерами до декількох тисяч вузлів, існує потреба оцінити його продуктивність у кластерах навіть більших розмірів. Це включає оцінку масштабованості планувальника «Kubernetes» для потреб «розумних міст», виявлення потенційних вузьких місць і пропонування рішень для їх усунення. Крім того, існує необхідність оцінити вплив великих розмірів кластерів на продуктивність застосунків і використання ресурсів. Ці дослідження можуть призвести до розробки більш ефективних алгоритмів планування та кращих стратегій керування для масштабних розгортань «Kubernetes» для потреб «розумних міст».

– Планування слід розглядати не тільки з точки зору статичної інфраструктури, але можна запропонувати просунуті контекстно-залежні алгоритми планування, які могли б зосередитися на розробці нових підходів до розподілу ресурсів і планування, які враховують ширший діапазон контекстних факторів: уподобання користувача, залежності програми та умови середовища. Це може включати вивчення нових методів машинного навчання та алгоритмів оптимізації, які можуть динамічно адаптуватися до мінливих умов і визначати пріоритетність ресурсів на основі відгуків і аналізу в режимі реального часу. Інші потенційні напрямки дослідження можуть включати розробку нових моделей і фреймворків для керування ресурсами в кластерах «Kubernetes», покращення оркестровки контейнерів і балансування навантаження, а також покращення можливостей моніторингу та аналітики для більш ефективного використання контекстно-залежних алгоритмів планування.

Оскільки різноманітність майбутніх напрямків чітко вказує на потенціал для нових досліджень у «Kubernetes» для потреб «розумних міст», зрілий із завданнями безлічі рівнів складності та зусиль. Це надає майбутнім дослідникам захоплюючі можливості для продовження та завдання для вирішення. Дане опитування полегшить майбутнім дослідникам вибір відповідного завдання та розв'яже нові задачі, щоб розширити сучасний рівень у сфері «Kubernetes» для потреб «розумних міст». Крім того, воно дозволить виявити нові підходи та інноваційні рішення, що сприятимуть подальшому розвитку цієї технології.

Підсумовуючи, проведене дослідження алгоритмів щодо планування «Kubernetes» надає вичерпний огляд поточного стану галузі. Він охоплює цілі, методології, алгоритми, експерименти та результати різних досліджень у цій галузі. Опитування підкреслює важливість планування в «Kubernetes» для потреб «розумних міст» і потребу в ефективних алгоритмах планування. Результати експериментів показують, що в цій області ще є місце для вдосконалення, і майбутня робота повинна бути зосереджена на розробці нових алгоритмів і вдосконаленні існуючих. Загалом опитування дає цінну інформацію про поточний стан планування «Kubernetes» для потреб «розумних міст» і вказує на багатообіцяючі напрямки для майбутніх досліджень.

3.5 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи розроблено методологію планування ресурсів для «Kubernetes» у «розумних містах». Запропоновано підходи до автоматизації розподілу ресурсів, спроектовано модель масштабування для інтелектуальних систем управління міським середовищем. Оцінено ефективність алгоритмів планування у сценаріях з високим навантаженням і протестовано їх у середовищі «Kubernetes», що підтвердило доцільність їх використання для оптимізації управління ресурсами в розумних містах. Проаналізовано сучасні підходи до масштабування в «Kubernetes» для потреб «розумних міст», зокрема «Horizontal Pod Autoscaler» і «Vertical Pod Autoscaler»

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Вимоги до серверних приміщень

Тема кваліфікаційної роботи освітнього рівня «Магістр» присвячена дослідженню алгоритмів планування «Kubernetes» для «розумних міст». Тому доцільно розглянути питання щодо забезпечення належних умов функціонування серверних приміщень, які є важливим компонентом інфраструктури для забезпечення безпеки та надійності роботи інформаційних систем. Вимоги до серверних приміщень спрямовані на забезпечення їх безперервної та безпечної експлуатації, захист обладнання від впливу зовнішніх чинників та запобігання надзвичайним ситуаціям.

Основною вимогою до серверного приміщення є дотримання всіх необхідних нормативів щодо його місця розташування, розмірів, внутрішнього планування та відповідності вимогам безпеки. Серверні приміщення повинні розташовуватися в спеціально обладнаних зонах будівлі, які мають мінімальний вплив зовнішніх ризиків: затоплення, пожежі, вібрації або впливи від сильних вітрів. Вибір даних зон важливий для запобігання можливим аварійним ситуаціям. Також важливо, щоб серверні приміщення не перебували поруч із джерелами електромагнітних завад: великі електричні установки чи трансформаторні підстанції. Це дає змогу мінімізувати можливі перешкоди в роботі серверів. Розміри приміщення мають відповідати кількості розміщеного в ньому обладнання та забезпечувати достатньо простору для його обслуговування, а також для забезпечення належної вентиляції та циркуляції повітря, що сприяє стабільній роботі серверів.

Одним із ключових аспектів є забезпечення належного мікроклімату в серверному приміщенні, що включає контроль за температурою і вологістю. Встановлюються вискоєфективні системи кондиціонування та вентиляції, які підтримують постійну температуру в діапазоні від вісімнадцяти до двадцяти семи градусів Цельсія, а рівень вологості – у межах від сорока до шістдесяти відсотків. Ці параметри мають вирішальне значення для стабільної роботи

обладнання, оскільки навіть невеликі відхилення від рекомендованих значень можуть призвести до перегріву серверів або утворення конденсату на їх компонентах, що може спричинити серйозні пошкодження та зниження ефективності роботи всього інформаційного центру. У разі надмірного перегріву обладнання можливі збої в роботі програмного забезпечення, уповільнення процесів обробки даних або навіть повне вимкнення серверів для запобігання їх пошкодженню. Конденсація вологи на внутрішніх компонентах може призвести до корозії та коротких замикань, що також виводить обладнання з ладу.

Для забезпечення стабільних умов на постійній основі використовуються автоматизовані системи моніторингу, які здатні оперативно сигналізувати про будь-які відхилення від заданих параметрів, швидко реагувати на зміну умов. Ці системи можуть не лише відслідковувати температуру та вологість, але й забезпечувати детальний аналіз, щоб визначити тенденції зміни клімату у приміщенні. У разі критичних змін, різке підвищення температури або зниження вологості, ці системи можуть активувати аварійне охолодження, додаткові вентилятори або інші механізми для підтримки оптимальних умов для серверного обладнання. Також, деякі системи мають можливість автоматично відключати неактивне або пошкоджене обладнання, щоб мінімізувати негативний вплив на інші компоненти. Завдяки такому рівню автоматизації можна забезпечити стабільність роботи серверів навіть у випадку виникнення непередбачуваних змін у середовищі, що сприяє збереженню їхньої працездатності та продовженню терміну експлуатації.

Для захисту серверного обладнання від пожежі необхідно облаштовувати приміщення сучасними автоматичними системами пожежогасіння, які здатні оперативно виявляти загрозу та нейтралізувати її без затримок. Вибір відповідної системи залежить від специфіки приміщення та типу обладнання, що в ньому розміщується. Найбільш ефективними вважаються системи, що використовують газові або порошкові засоби гасіння, оскільки вони швидко і ефективно гаснуть вогонь, не завдаючи шкоди електронним компонентам, на відміну від водяних або пінних систем. Вода або піна, потрапляючи на електронні компоненти, можуть спричинити коротке замикання, корозію, а також пошкодження

важливих частин обладнання, що може призвести до його повної поломки та втрати даних. Газові та порошкові системи забезпечують необхідний захист, не залишаючи після себе руйнівних наслідків для інфраструктури.

Крім того, серверні приміщення повинні бути оснащені високотехнологічними системами виявлення диму, що здатні вчасно сповіщати про початок загоряння на ранніх стадіях. Завдяки таким системам, навіть невеликі зміни в концентрації диму можуть бути зафіксовані, що забезпечує можливість запобігти розвитку великої пожежі. Це дає можливість вчасно вжити необхідних заходів: активування систем пожежогасіння або евакуація персоналу, що значно знижує ризики серйозних пошкоджень. Окрім цього, важливо, щоб система сповіщення була інтегрована з іншими механізмами безпеки, такими як автоматичні двері, які можуть замкнутись у разі пожежної небезпеки, щоб запобігти поширенню вогню до інших частин будівлі.

Стінові, стельові та підлогові покриття мають бути виготовлені з матеріалів, що не підтримують горіння, щоб зменшити ризик поширення вогню в разі загоряння. Такі матеріали не лише сприяють локалізації загоряння, але й здатні витримувати високі температури без руйнування, що дає час для евакуації людей і активування системи пожежогасіння. Вибір даних матеріалів пов'язаний з необхідністю забезпечення високого рівня безпеки, оскільки навіть найменша іскра чи теплове навантаження може стати джерелом пожежі в серверному приміщенні, де знаходяться численні електронні пристрої, які генерують тепло.

Загалом, система пожежної безпеки в серверному приміщенні повинна бути комплексною та враховувати всі можливі ризики. Вона повинна не лише забезпечити захист від пожежі, а й створювати умови для швидкої ліквідації загрози, не завдаючи шкоди обладнанню.

Забезпечення електропостачання серверних приміщень повинно відповідати підвищеним вимогам надійності та безперебійності. Для цього встановлюються резервні джерела живлення: джерела безперебійного живлення та дизельні генератори, які здатні гарантувати безперервну роботу серверів навіть у разі відключення електроенергії від основного джерела. Для мінімізації ризиків від перепадів напруги застосовуються стабілізатори та системи захисту

від імпульсних перенапруг, що дає можливість захистити чутливе обладнання від можливих пошкоджень. Кабельні мережі організовуються таким чином, щоб забезпечити максимальний захист від електромагнітних завад, застосовуючи екрановані кабелі, що мінімізують ймовірність зовнішнього впливу.

Організація фізичної безпеки серверних приміщень також є невід'ємною складовою забезпечення їх безпечного функціонування. Доступ до приміщення має бути обмежений лише для авторизованого персоналу, щоб запобігти несанкціонованому втручанню. Встановлюються системи контролю доступу: карткові зчитувачі, біометричні сканери або кодові замки, що дає змогу точно ідентифікувати осіб, які мають право на вхід. Додатково рекомендується інтегрувати системи відеоспостереження, які дають можливість здійснювати моніторинг переміщень у зоні серверної кімнати та фіксувати будь-які підозрілі дії.

Освітлення приміщення повинно забезпечувати достатній рівень видимості для виконання обслуговуючих робіт, при цьому використовуються світильники з низьким рівнем тепловиділення, щоб не порушувати температурний режим. Електромережа повинна бути організована з урахуванням вимог електробезпеки, передбачаючи заземлення всього обладнання та наявність пристроїв захисного відключення. Крім того, слід забезпечити наявність інструкцій з охорони праці та безпеки, які повинні бути розміщені в доступному місці для персоналу, що обслуговує серверне обладнання.

Важливим аспектом є підготовка персоналу, який обслуговує серверні приміщення. Працівники повинні проходити регулярне навчання з питань охорони праці, пожежної безпеки та дій у надзвичайних ситуаціях. Також проводяться інструктажі щодо використання засобів індивідуального захисту та правил роботи з електрообладнанням, щоб уникнути травм та інших небезпек. План евакуації у разі надзвичайної ситуації повинен бути розроблений і чітко відомий всім працівникам, причому він має бути розміщений у доступному для ознайомлення місці.

Таким чином, забезпечення належних умов для серверних приміщень вимагає дотримання комплексу вимог, спрямованих на створення безпечного та

стабільного середовища для роботи обладнання. Це включає в себе не лише правильну організацію фізичних умов, а й забезпечення безперебійного функціонування інформаційних систем, що є необхідним для підтримки надійності інфраструктури та відповідності сучасним стандартам безпеки.

4.2 Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн

Тема кваліфікаційної роботи освітнього рівня «Магістр» присвячена дослідженню алгоритмів планування «Kubernetes» для «розумних міст». Тому доцільно розглянути питання державної системи моніторингу довкілля як складової частини національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн, оскільки ефективне управління міськими ресурсами, а також покращення екологічної ситуації в умовах «розумних міст» неможливе без належної інтеграції даних про стан навколишнього середовища. Враховуючи важливість екологічної складової для розвитку сучасних міських територій, необхідно розробити систему [77], яка забезпечить своєчасний доступ до актуальної екологічної інформації, надаючи можливість виявляти та оперативно реагувати на екологічні загрози, а також здійснювати ефективне управління природними ресурсами.

Державна система моніторингу довкілля є важливим компонентом національної інформаційної інфраструктури, яка забезпечує постійний моніторинг стану навколишнього середовища та оперативне інформування громадськості, державних органів і наукових установ про зміни в екологічній ситуації. Система здійснює збір, обробку та зберігання даних щодо стану атмосферного повітря, водних ресурсів, ґрунтів, біорізноманіття, а також фізичних і хімічних параметрів довкілля, що нажає можливість створювати комплексну картину екологічної ситуації в державі. Державна система моніторингу повинна забезпечувати високий рівень точності даних, оперативність у реагуванні на зміни в екологічній ситуації, а також бути

інтегрованою в загальнодержавну та міжнародну інфраструктуру обміну даними для ефективного моніторингу екологічних змін на глобальному рівні.

У рамках національної інформаційної інфраструктури моніторинг довкілля здійснюється за допомогою складної архітектури, що включає різноманітні датчики, станції моніторингу, супутникові та авіаційні спостереження, а також мережі телекомунікацій, які забезпечують передачу інформації. Дані, отримані від цих засобів, передаються до централізованих баз даних, де відбувається їх обробка та аналіз. Важливим етапом є створення механізмів, які дають можливість оперативно отримувати зведену інформацію для державних органів, науковців, а також громадян, що забезпечить своєчасне реагування на виникнення надзвичайних екологічних ситуацій. Система повинна забезпечити здійснювання прогноз щодо стану довкілля на основі наявних даних і можливих сценаріїв зміни клімату, що сприятиме розробці ефективних заходів для збереження екологічної рівноваги.

У [78] інтеграція державних систем моніторингу довкілля з міжнародними платформами і базами даних є важливою умовою для обміну екологічною інформацією на глобальному рівні. Це надає можливість забезпечити узгодженість і стандартизацію збору, збереження та передачі екологічних даних, що є надзвичайно важливим для забезпечення прозорості і ефективності моніторингових процесів. За допомогою розвитку технологій обміну даними, а також досягнень у галузі взаємодії систем, різноманітні держави мають змогу створювати спільні платформи для моніторингу, даючи можливість здійснювати інтеграцію та обмін інформацією щодо стану навколишнього середовища.

Це надає можливість здійснювати моніторинг в режимі реального часу, що є надзвичайно важливим для оперативного реагування на екологічні загрози. Наприклад, за допомогою інтеграції національних систем моніторингу з міжнародними платформами можна отримувати актуальні дані про зміни кліматичних умов, рівень забруднення води та повітря, а також стан природних ресурсів у різних частинах світу. Ці дані можна використовувати не лише для аналізу поточного стану навколишнього середовища, а й для прогнозування можливих наслідків та розробки ефективних механізмів адаптації до змін.

Завдяки такій інтеграції різноманітні країни та міжнародні організації можуть спільно реагувати на глобальні екологічні загрози: зміна клімату, забруднення атмосфери, деградація ґрунтів і втрата біорізноманіття. Спільне використання даних дає змогу краще оцінити масштаб і вплив цих недоліків на різні регіони світу, що сприяє більш точному визначенню пріоритетів для подальших дій. Окрім того, це дає змогу створювати глобальні стратегії, спрямовані на пом'якшення наслідків екологічних проблем, а також на забезпечення сталого розвитку на планетарному рівні.

Міжнародне співробітництво у сфері екологічного моніторингу також має важливе значення для підвищення ефективності впровадження заходів, спрямованих на збереження природних ресурсів та захист навколишнього середовища. Наприклад, за допомогою глобальних баз даних можна створювати єдину інформаційну платформу, яка даючи змогу відслідковувати виконання міжнародних угод і зобов'язань, що стосуються охорони довкілля. Це не лише покращує рівень прозорості в міжнародних відносинах, але й дає змогу більш точно вимірювати прогрес у досягненні сталого розвитку.

Забезпечення сумісності національних систем моніторингу довкілля з міжнародними системами, такими як Європейська система моніторингу довкілля чи система ООН з моніторингу клімату та екологічних змін, є важливим елементом глобальної екологічної безпеки. Системи, що підтримують відкриті стандарти обміну даними, даючи змогу інтегрувати національні платформи з глобальними екологічними ініціативами, що сприяє створенню єдиного інформаційного простору для моніторингу довкілля. Ці платформи повинні враховувати специфіку кожної країни, зокрема її географічні та кліматичні умови, створювати персоналізовані моделі для прогнозування екологічних змін.

Важливим аспектом є також використання сучасних технологій для обробки та аналізу великих обсягів даних. Використання хмарних технологій, штучного інтелекту та алгоритмів машинного навчання надає можливість створювати автоматизовані системи прогнозування екологічних змін, що базуються на великому обсязі даних, що збираються в режимі реального часу. Такі системи можуть не лише прогнозувати розвиток екологічних загроз, а й

пропонувати оптимальні шляхи для їх нейтралізації, що сприяє підвищенню ефективності екологічної політики на національному та міжнародному рівнях.

Таким чином, система моніторингу довкілля є важливою складовою частиною національної інформаційної інфраструктури, що забезпечує оперативне реагування на екологічні зміни, а також сприяє міжнародній співпраці в боротьбі з глобальними екологічними викликами. Інтеграція даних систем у рамках «розумних міст» надає можливість здійснювати моніторинг на більш локалізованому рівні, оперативно реагувати на зміни в довкіллі, а також сприяти збереженню екологічної рівноваги на глобальному рівні.

4.3 Висновок до четвертого розділу

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано вимоги до серверних приміщень, включаючи аспекти безпеки, енергопостачання та охолодження, що необхідні для стабільної роботи інформаційних систем у середовищі «розумних міст». Описано державну систему моніторингу довкілля як складову національної інформаційної інфраструктури, сумісної з міжнародними аналогами, із врахуванням її значення для забезпечення екологічної безпеки та оперативного реагування в надзвичайних ситуаціях.

ВИСНОВКИ

У кваліфікаційній роботі виконано комплексне дослідження алгоритмів планування в «Kubernetes» для потреб «розумних міст». Поданий аналіз підтверджує актуальність використання інструментів оркестрації контейнерів для забезпечення ефективності, надійності та адаптивності міських інформаційних систем. В умовах зростання обсягів даних і збільшення кількості IoT-пристроїв, впровадження сучасних алгоритмів планування є важливим кроком до оптимізації використання ресурсів і підвищення продуктивності інфраструктури.

У роботі розглянуто концепцію «розумного міста» як міського середовища, яке інтегрує цифрові технології для покращення якості життя громадян і ефективного управління ресурсами. Висвітлено роль «Kubernetes» як провідної платформи для автоматизації управління контейнерними застосунками. Особливу увагу приділено аналізу існуючих підходів до планування ресурсів, які враховують багатоцільову оптимізацію, штучний інтелект та механізми автоматичного масштабування.

Дослідження продемонструвало ефективність алгоритмів «Horizontal Pod Autoscaler» та «Vertical Pod Autoscaler» у сценаріях з високими навантаженнями. Проведений порівняльний аналіз показав, що такі інструменти сприяють забезпеченню балансу між ефективністю використання ресурсів та стабільністю роботи кластера. Також виявлено перспективність впровадження алгоритмів, орієнтованих на багатоцільову оптимізацію, з метою одночасного підвищення продуктивності, зменшення енергоспоживання та мінімізації часу відповіді.

Основними викликами, що потребують подальшого дослідження, є динамічний характер робочих навантажень, забезпечення масштабованості та стабільності системи в умовах великих даних. Робота пропонує рекомендації щодо вдосконалення підходів до планування, враховуючи специфіку міських інформаційних систем.

Отримані результати мають значний науковий та практичний потенціал. Вони можуть бути використані для розробки нових алгоритмів планування, що

орієнтовані на динамічні сценарії міської інфраструктури. Практичне значення роботи полягає у підвищенні ефективності та надійності інформаційних систем «розумних міст», що є важливим для їхнього подальшого розвитку та інтеграції.

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр»:

- Подано загальне визначення концепту «розумне місто» та його значення в сучасних технологіях і міській інфраструктурі.

- Розглянуто можливості використання «Kubernetes» як інструменту оркестрації контейнерів для забезпечення ефективності міських систем.

- Висвітлено означення «Kubernetes» та його використання для «розумних міст».

- Проаналізовано знайдені наукові публікації щодо алгоритмів планування «Kubernetes» для «розумних міст».

- Досліджено використання «Kubernetes» для забезпечення ефективності та автоматизації в інфраструктурах.

- Обґрунтовано доцільність використання Kubernetes у «розумних містах» для оптимізації використання ресурсів та підвищення продуктивності.

- Сформовано чітке уявлення про напрями досліджень у сфері алгоритмів планування «Kubernetes» та можливі перспективи їх застосування у «розумних містах».

В другому розділі кваліфікаційної роботи:

- Описано основні принципи роботи алгоритмів планування в «Kubernetes» для «розумних міст», зокрема підходи, орієнтовані на багатоцільову оптимізацію, штучний інтелект та автомасштабування.

- Досліджено ефективність алгоритмів планування «Kubernetes» у сценаріях з високими навантаженнями та динамічними робочими середовищами.

- Подано порівняльний опис підходів до планування.

В третьому розділі кваліфікаційної роботи:

- Розроблено підхід до масштабування та оптимізації ресурсів у «Kubernetes» для сценаріїв, характерних для «розумних міст».

- Запропоновано методи автоматичного масштабування, що враховують динамічні навантаження і специфічні вимоги міських інформаційних систем.

- Спроектовано ефективну структуру планування ресурсів із використанням алгоритмів «Horizontal Pod Autoscaler» та «Vertical Pod Autoscaler».

- Протестовано розроблені алгоритми в середовищі Kubernetes для оцінки їхньої продуктивності, стабільності та енергоефективності під час високих навантажень.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано вимоги до серверних приміщень, включаючи аспекти безпеки, енергопостачання та охолодження, що необхідні для стабільної роботи інформаційних систем у середовищі «розумних міст». Описано державну систему моніторингу довкілля як складову національної інформаційної інфраструктури, сумісної з міжнародними аналогами, із врахуванням її значення для забезпечення екологічної безпеки та оперативного реагування в надзвичайних ситуаціях.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Tahmasseby, S. The Implementation of Smart Mobility for Smart Cities: A Case Study in Qatar. *Civ. Eng. J.* 2022, 8, 2154–2171.
- 2 Ismagilova, E.; Hughes, L.; Dwivedi, Y.K.; Raman, K.R. Smart cities: Advances in research — An information systems perspective. *Int. J. Inf. Manag.* 2019, 47, 88–100.
- 3 Abadía, J.J.P.; Walther, C.; Osman, A.; Smarsly, K. A systematic survey of Internet of Things frameworks for smart city applications. *Sustain. Cities Soc.* 2022, 83, 103949.
- 4 Raspotnik, A.; Grønning, R.; Herrmann, V. A tale of three cities: The concept of smart sustainable cities for the Arctic. *Polar Geogr.* 2020, 43, 64–87.
- 5 Jose Sanchez Gracias, Gregory S. Parnell, Eric Specking, Edward A. Pohl, Randy Buchanan (2023) Smart Cities - A Structured Literature Review. pp 1724 – 1725.
- 6 Kwak, Y.H.; Lee, J. Toward Sustainable Smart City: Lessons From 20 Years of Korean Programs. *IEEE Trans. Eng. Manag.* 2021, 70, 740–754.
- 7 Hammons, R.; Myers, J. Smart Cities. *IEEE Internet Things Mag.* 2019, 2, 8–9.
- 8 Mondal SK, Pan R, Kabir HMD, Tian T, Dai HN (2022) Kubernetes in IT administration and serverless computing: an empirical study and research challenges. *J Supercomput* 78(2):2937–2987.
- 9 Phuc LH, Phan LA, Kim T (2022) Traffic-Aware horizontal pod autoscaler in kubernetes-based edge computing infrastructure. *IEEE Access* 10:18966–18977.
- 10 Zhang M, Cao J, Yang L, Zhang L, Sahni Y, Jiang S (2022) ENTS: An Edgenative Task Scheduling System for Collaborative Edge Computing. *IEEE/ ACM 7th Symposium on Edge Computing, SEC.* pp 149–161.
- 11 Kim SH, Kim T (2023) Local scheduling in kubeedge-based edge computing environment. *Sensors* 23(3):1522.
- 12 E. Casalicchio (2019) "Container orchestration: A survey" *Syst Model*, 221–235.

- 13 Pahl C, Brogi A, Soldani J, Jamshidi P (2017) Cloud container technologies: a state-of-the-art review. *IEEE Transact Cloud Comput* 7(3):677–692.
- 14 Rodriguez MA, Buyya R (2019) Container-based cluster orchestration systems: A taxonomy and future directions. *Software Pract Experience* 49(5):698–719.
- 15 Truyen E, Van Landuyt D, Preuveneers D, Lagaisse B, Joosen W (2019) A comprehensive feature comparison study of open-source container orchestration frameworks. *Appl Sciences (Switzerland)* 9(5):931.
- 16 Arunarani AR, Manjula D, Sugumaran V (2019) Task scheduling techniques in cloud computing: a literature survey. *Futur Gener Comput Syst* 91:407–415.
- 17 Vijindra and S. Shenai, (2012) Survey on scheduling issues in cloud computing. *Procedia Eng* 38:2881–2888.
- 18 Wang K, Zhou Q, Guo S, Luo J (2018) Cluster frameworks for efficient scheduling and resource allocation in data center networks: a survey. *IEEE Commun Surveys Tutor* 20(4):3560–3580.
- 19 Hosseinioun P, Kheirabadi M, Kamel Tabbakh SR, Ghaemi R (2022) A task scheduling approaches in fog computing: a survey”. *Transact Emerg TelecommunTechnol* 33(3):e3792.
- 20 Rejiba Z, Chamanara J (2022) Custom scheduling in Kubernetes: a survey on common problems and solution approaches. *ACM Comput Surv* 55(7):1–37.
- 21 Carrión C (2022) Kubernetes scheduling: taxonomy, ongoing issues and challenges. *ACM Comput Surv* 55(7):1–37.
- 22 Santos J, Wauters T, Volckaert B, De Turck F (2019) Towards network-Aware resource provisioning in Kubernetes for fog computing applications. *Proceedings of the IEEE Conference on Network Softwarization: Unleashing the Power of Network Softwarization*. pp 351–359.
- 23 Chung A, Park JW, Ganger GR (2018) Stratus: Cost-aware container scheduling in the public cloud. *Proceedings of the ACM Symposium on Cloud Computing*. pp 121–134.
- 24 Le TN, Sun X, Chowdhury M, Liu Z (2020) AlloX: Compute allocation in hybrid clusters. *Proceedings of the 15th European Conference on Computer Systems, EuroSys*.

25 Zhong Z, Buyya R (2020) A Cost-Efficient Container Orchestration Strategy in Kubernetes-Based Cloud Computing Infrastructures with Heterogeneous Resources. *ACM Trans Internet Technol* 20(2):1–24.

26 Thinakaran P, Gunasekaran JR, Sharma B, Kandemir MT, Das CR (2019) Kube-Knots: Resource Harvesting through Dynamic Container Orchestration in GPU-based Datacenters. *Proceedings - IEEE International Conference on Cluster Computing, ICC*.

27 Townend P et al (2019) Invited paper: Improving data center efficiency through holistic scheduling in Kubernetes. *Proceedings - 13th IEEE International Conference on Service-Oriented System Engineering, 10th International Workshop on Joint Cloud Computing, and IEEE International Workshop on Cloud Computing in Robotic Systems, CCRS*. pp 156–166.

28 Menouer T (2021) KCSS: Kubernetes container scheduling strategy. *J Supercomput* 77(5):4267–4293.

29 Song S, Deng L, Gong J, Luo H (2019) Gaia scheduler: A kubernetesbased scheduler framework. *16th IEEE International Symposium on Parallel and Distributed Processing with Applications, 17th IEEE International Conference on Ubiquitous Computing and Communications, 8th IEEE International Conference on Big Data and Cloud Computing*. pp 252–259.

30 Ogbuachi MC, Gore C, Reale A, Suskovics P, Kovacs B (2019) Contextaware K8S scheduler for real time distributed 5G edge computing applications. *27th International Conference on Software, Telecommunications and Computer Networks, SoftCOM*.

31 Beltre A, Saha P, Govindaraju M (2019) KubeSphere: An approach to multi-tenant fair scheduling for Kubernetes clusters. *3rd IEEE International Conference on Cloud and Fog Computing Technologies and Applications, Cloud Summit*. pp 14–20.

32 Haja D, Szalay M, Sonkoly B, Pongracz G, Toka L (2019) Sharpening Kubernetes for the Edge. *ACM SIGCOMM Conference Posters and Demos, Part of SIGCOMM*. pp 136–137.

33 Wojciechowski L et al (2021) NetMARKS: Network metrics-AwaRe Kubernetes scheduler powered by service mesh. *Proceedings - IEEE INFOCOM*.

34 El Haj Ahmed G, Gil-Castiñeira F, Costa-Montenegro E (2021) KubCG: A dynamic Kubernetes scheduler for heterogeneous clusters. *Software Pract Experience* 51(2):213–234.

35 Ungureanu OM, Vlădeanu C, Kooij R (2019) Kubernetes cluster optimization using hybrid shared-state scheduling framework. *ACM International Conference Proceeding Series*.

36 Yang S, Ren Y, Zhang J, Guan J, Li B (2021) KubeHICE: Performance-aware Container Orchestration on Heterogeneous-ISA Architectures in CloudEdge Platforms. 19th IEEE International Symposium on Parallel and Distributed Processing with Applications, 11th IEEE International Conference on Big Data and Cloud Computing, 14th IEEE International Conference on Social Computing and Networking and 11th IEEE International Conference on Cloud Computing. pp 81–91.

37 Li D, Wei Y, Zeng B (2020) A Dynamic I/O Sensing Scheduling Scheme in Kubernetes. *ACM International Conference Proceeding Series*. pp 14–19.

38 Fan D, He D (2020) A Scheduler for Serverless Framework base on Kubernetes. *ACM International Conference Proceeding Series*. pp 229–232.

39 Dua A, Randive S, Agarwal A, Kumar N (2020) Efficient Load balancing to serve Heterogeneous Requests in Clustered Systems using Kubernetes. *IEEE 17th Annual Consumer Communications and Networking Conference, CCNC*.

40 Cai Z, Buyya R (2022) Inverse Queuing Model-Based Feedback Control for Elastic Container Provisioning of Web Systems in Kubernetes. *IEEE Trans Comput* 71(2):337–348.

41 Bestari MF, Kistijantoro AI, Sasmita AB (2020) Dynamic Resource Scheduler for Distributed Deep Learning Training in Kubernetes. *7th International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA*.

42 Kaur K, Garg S, Kaddoum G, Ahmed SH, Atiquzzaman M (2020) KEIDS: Kubernetes-Based Energy and Interference Driven Scheduler for Industrial IoT in Edge-Cloud Ecosystem. *IEEE Internet Things J* 7(5):4228–4237.

43 Lin M, Xi J, Bai W, Wu J (2019) Ant colony algorithm for multi-objective optimization of container-based microservice scheduling in cloud. *IEEE Access* 7:83088–83100.

- 44 Wei-guo Z, Xi-lin M, Jin-zhong Z (2018) Research on kubernetes' resource scheduling scheme. ACM International Conference Proceeding Series.
- 45 Oleghe O (2021) Container placement and migration in edge computing: concept and scheduling models. IEEE Access 9:68028–68043.
- 46 Carvalho M, MacEdo DF (2021) QoE-Aware Container Scheduler for Colocated Cloud Environments,” Faculdades Catolicas.
- 47 Chen T, Li M, Li Y, Lin M, Wang N, Wang M, Xiao T, Xu B, Zhang C, Zhang Z (2015) Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.
- 48 Abadi M et al (2016) Tensorflow: a system for large-scale machine learning. OsdI 2016(16):265–283.
- 49 Xing EP et al (2015) Petuum: A new platform for distributed machine learning on big data. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp 1335–1344.
- 50 Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J (2015) Large-scale cluster management at Google with Borg. 10th European Conference on Computer Systems, EuroSys. pp 1–15.
- 51 Vavilapalli VK et al (2013) Apache hadoop YARN: Yet another resource negotiator. 4th Annual Symposium on Cloud Computing, SoCC. pp 1–16.
- 52 Bao Y, Peng Y, Wu C, Li Z (2018) Online Job Scheduling in Distributed Machine Learning Clusters. Proceedings - IEEE INFOCOM. pp 495–503.
- 53 Peng Y, Bao Y, Chen Y, Wu C, Guo C (2018) Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters. Proceedings of the 13th EuroSys Conference, EuroSys.
- 54 Mao H, Schwarzkopf M, Venkatakrisnan SB, Meng Z, Alizadeh M (2019) Learning scheduling algorithms for data processing clusters. SIGCOMM Conference of the ACM Special Interest Group on Data Communication. pp 270–288.
- 55 Chaudhary S, Ramjee R, Sivathanu M, Kwatra N, Viswanatha S (2020) Balancing efficiency and fairness in heterogeneous GPU clusters for deep learning. Proceedings of the 15th European Conference on Computer Systems, EuroSys.

56 Fu Y et al (2019) Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster. IEEE International Conference on Big Data, Big Data. pp 278–287.

57 Peng Y, Bao Y, Chen Y, Wu C, Meng C, Lin W (2021) DL2: A Deep LearningDriven Scheduler for Deep Learning Clusters. IEEE Trans Parallel Distrib Syst 32(8):1947–1960.

58 Mao Y, Fu Y, Zheng W, Cheng L, Liu Q, Tao D (2022) Speculative Container Scheduling for Deep Learning Applications in a Kubernetes Cluster. IEEE Syst J 16(3):3770–3781.

59 Huang J, Xiao C, Wu W (2020) RLSK: A Job Scheduler for Federated Kubernetes Clusters based on Reinforcement Learning. IEEE International Conference on Cloud Engineering, IC2E. pp 116–123.

60 Wang H, Liu Z, Shen H (2020) Job scheduling for large-scale machine learning clusters. Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies. pp 108–120.

61 Casquero O, Armentia A, Sarachaga I, Pérez F, Orive D, Marcos M (2019) Distributed scheduling in Kubernetes based on MAS for Fog-in-the-loop applications. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA. pp 1213–1217.

62 Han Y, Shen S, Wang X, Wang S, Leung VCM (2021) Tailored learningbased scheduling for kubernetes-oriented edge-cloud system. Proceedings - IEEE INFOCOM.

63 Yang Y, Chen L (2019) Design of Kubernetes Scheduling Strategy Based on LSTM and Grey Model. Proceedings of IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering, ISKE. pp 701–707.

64 Zhang X, Li L, Wang Y, Chen E, Shou L (2021) Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters. IEEE Access 9:105192–105204.

65 Liu Z, Chen C, Li J, Cheng Y, Kou Y, Zhang D (2022) KubFBS: A fine-grained and balance-aware scheduling system for deep learning tasks based on

Kubernetes. *Concurrency Computat Pract Exper* 34(11):e6836. <https://doi.org/10.1002/cpe.6836>.

66 Rahali M, Phan CT, Rubino G (2021) KRS: Kubernetes Resource Scheduler for resilient NFV networks. *IEEE Global Communications Conference*.

67 Rattihalli G, Govindaraju M, Lu H, Tiwari D (2019) Exploring potential for non-disruptive vertical auto scaling and resource estimation in Kubernetes. *IEEE International Conference on Cloud Computing, CLOUD*. pp 33–40.

68 Toka L, Dobreff G, Fodor B, Sonkoly B (2021) Machine Learning-Based Scaling Management for Kubernetes Edge Clusters. *IEEE Trans Netw Serv Manage* 18(1):958–972.

69 Taherizadeh S, Stankovski V (2019) Dynamic multi-level auto-scaling rules for containerized applications. *Computer J* 62(2):174–197.

70 Balla D, Simon C, Maliosz M (2020) Adaptive scaling of Kubernetes pods. *IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS*.

71 Wang M, Zhang D, Wu B (2020) A Cluster Autoscaler Based on Multiple Node Types in Kubernetes. *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC*. pp 575–579.

72 Kang R, Zhu M, He F, Sato T, Oki E (2021) Design of Scheduler Plugins for Reliable Function Allocation in Kubernetes. *17th International Conference on the Design of Reliable Communication Networks, DRCN*.

73 Vu DD, Tran MN, Kim Y (2022) Predictive hybrid autoscaling for containerized applications. *IEEE Access* 10:109768–109778.

74 Vorobyiov O (2021) *Scaling applications in Kubernetes*. LNU Publishing, Lviv.

75 Kubecost Team (n.d.) Kubernetes HPA: Horizontal Pod Autoscaler Explained. Available at: <https://www.kubecost.com/kubernetes-autoscaling/kubernetes-hpa> (Accessed: 02 December 2024).

76 Kubecost Team (n.d.) Kubernetes VPA: Vertical Pod Autoscaler Explained. Available at: <https://www.kubecost.com/kubernetes-autoscaling/kubernetes-vpa> (Accessed: 02 December 2024).

77 Stepanova LS (2020) Environmental policy of Ukraine and international experience. Kyiv: Center for Economic Strategy.

78 Zhukovsky MP (2020) Modern trends in the utilization of municipal solid waste in cities. Journal "Scientific World", 12(4), pp 49–55.

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА**

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



18–19 грудня 2024 року

**ТЕРНОПІЛЬ
2024**

УДК 001
М34

ПРОГРАМНИЙ КОМІТЕТ

Голова: Приймак Микола – професор кафедри комп'ютерних систем та мереж, д.т.н., професор.

Співголови: Марущак Павло – проректор з наукової роботи, докт. техн. наук, професор.

Баран Ігор – канд. техн. наук, доцент, декан факультету ФІС.

Науковий секретар: Надія Крива – старший викладач.

Члени: Василь Кривень – завідувач кафедри математичних методів в інженерії д.ф.-м.н., професор; Галина Осухівська – завідувач кафедри комп'ютерних систем та мереж, к.т.н., доцент; Микола Карпінський – професор кафедри кібербезпеки, д.т.н., професор; Жанна Баб'як – завідувач кафедри української та іноземних мов, к.пед. н., доцент; Ярослав Литвиненко – професор кафедри комп'ютерних наук, д.т.н., професор; Михайло Петрик – завідувач кафедри програмної інженерії, д.ф.-м.н., професор; Наталія Загородна – завідувач кафедри кібербезпеки, к.т.н., доцент.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова: Скоренький Юрій Любомирович – канд. техн. наук, доцент кафедри фізики.

Члени: доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; доцент кафедри кібербезпеки, к.т.н. М. Стадник; доцент кафедри комп'ютерних систем та мереж, к.т.н. Є. Тиш; ст. викладач Л. Джиджора.

Матеріали XII науково-технічної конфції «Інформаційні моделі, системи та технології»
М34 Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 18–19 грудня 2024 р.). – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024. – 238 с.

Адреса оргкомітету: ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001, тел. (0352) 52-41-33, факс (0352) 25-49-83.

E-mail: confis2024@gmail.com

Редагування, оформлення та верстка: Надія Крива

СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання
- Інформаційні системи та технології, кібербезпека
- Комп'ютерні системи та мережі
- Програмна інженерія та моделювання складних розподілених систем
- Новітні фізико-технічні та освітні технології

В збірнику надруковано тези доповідей XII науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 18–19 грудня 2024 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології, кібербезпека; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор.

К. Костюк РОЗРОБКА МЕТОДОЛОГІЇ ДЛЯ ОЦІНКИ РИЗИКІВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ДЛЯ АГРОХОЛДИНГУ K. Kostiuk DEVELOPMENT OF A METHODOLOGY FOR INFORMATION SECURITY RISK ASSESSMENT FOR AN AGRICULTURAL HOLDING	46
Ю. Р. Курчак МОДЕЛЮВАННЯ ТА АНАЛІЗ СИСТЕМИ РЕКОМЕНДАЦІЙ НА ВЕБ-САЙТІ НА ОСНОВІ ТЕОРІЇ ГРАФІВ І СТАТИСТИЧНИХ АЛГОРИТМІВ Y. R. Kurchak MODELING AND ANALYSIS OF A WEBSITE RECOMMENDATION SYSTEM BASED ON GRAPH THEORY AND STATISTICAL ALGORITHMS	47
В. А. Лабчук ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ MICROSOFT LINQ ДЛЯ ЗАДАЧ КОНСОЛІДАЦІЇ ДАНИХ V. A. Labchuk RESEARCHMENT OF THE DATA CONSOLIDATION TASKS' OPPORTUNITIES WITH THE HELP OF USING MICROSOFT LINQ TECHNOLOGY	48
В. А. Лабчук ПЕРЕВАГИ ТА НЕДОЛІКИ LINQ В ПОРІВНЯННІ З SQL-КОДОМ V. A. Labchuk PROS AND CONS OF USING LINQ INSTEAD OF RAW-SQL	49
С. В. Литвиненко, М. Є. Фріз МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ТАРГЕТОВАНОЇ РЕКЛАМИ S. V. Lytvynenko, M. E. Friz MATHEMATICAL SUPPORT OF TARGETED ADVERTISING	50
О. Ловчук, В. Дубельт, А. Лисий KUBERNETES В ОБЧИСЛЮВАЛЬНІЙ ІНФРАСТРУКТУРІ РОЗУМНИХ МІСТ O. Lovchuk, V. Dubelt, A. LYSYI KUBERNETES IN THE SMART CITIES COMPUTING INFRASTRUCTURE	51
О. Ловчук, Р. Катрич, В. ДУДА АКТУАЛЬНІСТЬ ХМАРНОГО МАСШТАБУВАННЯ ТА KUBERNETES O. Lovchuk, R. Katrych, V. Duda THE RELEVANCE OF CLOUD SCALATION AND KUBERNETES	52
П. Луговський, І. Фалендиш ВИБІР ІНФОРМАТИВНИХ ОЗНАК ДЛЯ ЗАДАЧІ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ANDROID P. Lugovskyi, I. Falendysh SELECTION OF INFORMATIVE FEATURES FOR THE TASK OF ANDROID MALWARE DETECTION	53
В. Мазур МЕТОДИ ЗАХИСТУ ВІД КІБЕРАТАК НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ V. Mazur AI-BASED METHODS FOR PROTECTION AGAINST CYBERATTACKS	54
М. Маланчук, Г. Козбур ОГЛЯД ВІТЧИЗНЯНИХ АВТОМАТИЗОВАНИХ СИСТЕМ ДЛЯ НАДАННЯ ПЕРШОЇ МЕДИЧНОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ЧАТ-БОТІВ M. Malanchuk, H. Kozbur REVIEW OF DOMESTIC AUTOMATED SYSTEMS FOR FIRST AID USING CHATBOTS	55

УДК 004.9

О. Ловчук; В. Дубельт; А. Лисий

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

KUBERNETES В ОБЧИСЛЮВАЛЬНІЙ ІНФРАСТРУКТУРІ РОЗУМНИХ МІСТ

UDC 004.9

O. Lovchuk; V. Dubelt; A. LYSYI

KUBERNETES IN THE SMART CITIES COMPUTING INFRASTRUCTURE

Розумні міста інтегрують обширні множини IoT-пристроїв. Це супроводжується зростанням навантаження обчислювальних і мережевих систем для забезпечення функціонування різнотипових «розумних» систем, як от моніторинг транспортних потоків, оперативне управління процесами енергоспоживання, контроль параметрів довкілля, інтелектуальні системи відеоспостереження та інші «розумні» служби. Інтеграція Kubernetes в обчислювальну інфраструктуру розумних міст дає змогу забезпечити гнучкість та масштабованість «розумних» цифрових сервісів, сприяючи при цьому підвищенню їх стійкості та надійності. Алгоритми планування в системах класу «розумне місто» здатні автоматично перемикаєти обчислювальні навантаження в разі відмови окремих вузлів або при виникненні непередбачуваних навантажень.

Kubernetes використовує декларативний підхід до керування програмно-алгоритмічними засобами, коли адміністратори чи користувачі вказують бажані стани програм, а система автоматично їх підтримує. Він також надає обширний перелік інструментів для моніторингу та керування застосунками, зокрема, програмно-алгоритмічні засоби для автоматичного виявлення збоїв та самовідновлення. Загалом Kubernetes пропонує гнучкі алгоритмічні рішення для керування контейнерними програмами у виробничих обчислювальних середовищах. Kubernetes підходить для веб-застосунків на основі мікросервісів. При цьому кожен компонент веб-систем можна запускати в окремому контейнері [1]. Завдяки легкості розгортання контейнерів, їх можна легко створювати та знищувати, забезпечуючи оперативне та високоефективне використання обчислювальних ресурсів. Kubernetes автоматизує процеси розгортання, масштабування та керування контейнерами в обчислювальному кластері, покращуючи використання ефективності і гнучкість виділення, використання та вивільнення різнотипових ресурсів. Це спрощує процедури створення та підтримки складних програмно-алгоритмічних комплексів. Крім стандартних стратегій планування Kubernetes, як от BestEffort або Burstable, для «розумних міст» можна використовувати спеціалізовані оригінальні стратегії планування, які базуються на характеристиках міської обчислювальної інфраструктури та його сервісних потребах [2]. Наприклад, планувальники можуть бути оперативно налаштовані для вибору окремих обчислювальних вузлів на основі географічного положення для забезпечення мінімального часу затримки або підвищення ефективності процесів обміну даними між регіонами. Водночас, алгоритми масштабування Kubernetes можуть бути модифіковані для інтеграції з системами прогнозування обчислювального навантаження.

Література

1. Senjab, Khaldoun, et al. «A survey of Kubernetes scheduling algorithms». *Journal of Cloud Computing* 12.1 (2023): 87.
2. Li, Hongjian, et al. «Cost-efficient scheduling algorithms based on beetle antennae search for containerized applications in Kubernetes clouds». *The Journal of Supercomputing* 79.9 (2023): 10300-10334.

УДК 004.032

О. Ловчук; Р. Катрич; В. Дуда

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АКТУАЛЬНІСТЬ ХМАРНОГО МАСШТАБУВАННЯ ТА KUBERNETES

UDC 004.032

O. Lovchuk; R. Katrych; V. Duda

THE RELEVANCE OF CLOUD SCALATION AND KUBERNETES

На даний час в процесі розширення хмарних платформ та послуг зростає потреба в підвищенні продуктивності інфраструктур центрів обробки даних. Високопродуктивні обчислення, передові мережеві рішення та стратегії динамічної оптимізації обчислювальних ресурсів зможуть допомогти персоналу центрів хмарного опрацювання даних забезпечувати високу швидкість та ефективність, що необхідні для надання високоякісних послуг [1]. При цьому однією з ефективних стратегій оптимізації хмарних обчислювальних процесів є запуск контейнерних програм. Він забезпечує покращення портативності програмно-алгоритмічних засобів, підвищення безпеки, прискорення перебігу процесів використання ресурсів, пришвидшення процедур розгортання та масштабування, а також покращення процесів інтеграції та взаємодії. Зазначені переваги зможуть допомогти установам та організаціям покращити протікання процесів розгортання застосунків, їх супроводу, обслуговування та керування, даючи змогу оперативніше та ефективніше реагувати на динамічні потреби бізнес-процесів.

Kubernetes – це система оркестровки контейнерів. Вона призначена для автоматизації процесів розгортання, масштабування та управління контейнерними програмно-алгоритмічними засобами. Однією з ключових її особливостей є функціональні можливості планувати процедури розгортання та виконання контейнерів у кластерах обчислювальних вузлів за допомогою алгоритмів планування [2]. Ці алгоритми визначають найкраще розміщення контейнерів на доступних обчислювальних вузлах у кластері. Система оркестрації контейнерів Kubernetes, пропонує можливість автоматизованого керування та масштабування цих розумних служб, забезпечуючи високі показники надійності та стійкості роботи за допомогою динамічного планування обчислювального навантаження. Тому алгоритми планування в Kubernetes для розумних міст є ключовими компонентами ефективного управління та розподілу ресурсів в міській обчислювальній інфраструктурі.

Тому актуальним напрямком сучасних досліджень є вичерпний аналітичний огляд різнотипових алгоритмів планування Kubernetes [1]. При дослідженні характеристик алгоритмів планування Kubernetes їх можна згрупувати у чотири підкатегорії:

- загальне планування;
- планування з підтримкою автоматичного масштабування;
- планування на основі багатоцільової оптимізації;
- орієнтоване на штучний інтелект планування.

Алгоритми планування Kubernetes функціонують на основі різнотипових параметрів, як от обчислювальні ресурси, мережеві вимоги, характеристики зв'язку та інші критерії.

Література

1. Senjab, Khaldoun, et al. «A survey of Kubernetes scheduling algorithms». *Journal of Cloud Computing* 12.1 (2023): 87.
3. Wen, Shilin, et al. «K8sSim: A Simulation Tool for Kubernetes Schedulers and Its Applications in Scheduling Algorithm Optimization». *Micromachines* 14.3 (2023): 651.