Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи управління розподіленими інформаційними системами у хмарному середовищі

(тема)

Виконав:

студент II курсу, групи СПм-22-5

Запорожченко А.П.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: проф. Волк М.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Запорожченку Антону Петровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи управління розподіленими інформаційними системами у хмарному середовищі

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи
Моделі хмарних обчислень.
Постачальники хмарних послуг (AWS, GCP, Azure).
Методи оцінки продуктивності хмарних платформ.

4. Перелік питань, що потрібно опрацювати у роботі
Аналіз предметної області.
Методи управління інформаційними системами у хмарному середовищі.
Результати аналізу методів управління розподіленими інформаційними системами у хмарному середовищі.
Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Презентація 12 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	02.04.24 – 04.04.24	
2	Розробка моделей	05.04.24 – 15.04.24	
3	Реалізація алгоритмів	16.04.24 – 28.04.24	
4	Розробка структури програмних засобів	29.04.24 – 15.05.24	
5	Розробка програмних модулів	16.05.24 – 25.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	26.05.24 – 05.06.24	
7	Подання кваліфікаційної роботи керівникові та	07.06.24 – 12.06.24	
	попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	13.06.24 – 15.06.24	
	вання		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Волк М.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 68 с., 4 рис., 1 табл., 1 дод., 53 джерела.

МЕТОДИ УПРАВЛІННЯ, РОЗПОДІЛЕНІ ІНФОРМАЦІЙНІ СИСТЕМИ, ХМАРНІ ОЧИСЛЕННЯ, ДАТАЦЕНТРИ.

Навантаження в хмарному середовище сьогодні керуються в розподіленому середовищі й обробляються в територіально розподілених центрах обробки даних. Постачальники хмарних послуг розміщують центри обробки даних по всьому світу для зменшення експлуатаційних витрат та підвищити якості обслуговування за допомогою інтелектуальних стратегій управління навантаженням і ресурсами.

Така масштабна та складна оркестровка програмного навантаження та апаратних ресурсів залишається проблемою, яку важко вирішити ефективно. Дослідники та практики намагаються вирішити цю проблему, пропонуючи різноманітні методи керування хмарою, такі як, наприклад, методи математичної оптимізації історично використовувалися для вирішення проблем керування хмарою. Але ці методи важко масштабувати до масштабів георозподіленої проблеми та мають обмежену застосовність у динамічних гетерогенних системних середовищах, що змушує постачальників хмарних послуг досліджувати інтелектуальні альтернативи на основі даних і машинного навчання (ML). Характеристика, прогнозування, контроль і оптимізація складних, неоднорідних і постійно мінливих розподілених хмарних ресурсів і робочих навантажень із застосуванням методологій машинного навчання привернули велику увагу в останні роки. У цій роботі розглядаються найсучасніші методи ML для проблеми керування хмарним центром обробки даних.

ABSTRACT

Master’s thesis: 68 pages, 4 figures, 1 table, 1 appendice, 53 sources.

MANAGEMENT METHODS, INFORMATION SYSTEMS DISTRIBUTION, HMARNE COMPUTING, DATA CENTER.

Cloud workloads today are managed in a distributed environment and processed in geographically distributed data centers. Cloud service providers deploy data centers around the world to reduce operational costs and improve service quality through intelligent load and resource management strategies.

Such a large and complex orchestration of software workload and hardware resources remains a challenge that is difficult to solve effectively. Researchers and practitioners are trying to solve this problem by proposing various cloud management techniques, such as mathematical optimization techniques have historically been used to solve cloud management problems. But these methods are difficult to scale to the scale of a geo-distributed problem and have limited applicability in dynamic heterogeneous system environments, forcing cloud service providers to explore intelligent alternatives based on data and machine learning (ML). Characterization, prediction, control, and optimization of complex, heterogeneous, and constantly changing distributed cloud resources and workloads using machine learning methodologies have attracted much attention in recent years. This paper reviews state-of-the-art ML techniques for the cloud data center management problem.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІТ – інформаційні технології

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ШІ – штучний інтелект

ЦОД – центри обробки даних

ЦП – центральний процесор

ANN – штучна нейронна мережа (англ., Artificial Neural Network)

API – інтерфейс програмування додатків (англ., Application Programming Interface)

ARIMA – авторегресійне інтегроване ковзне середнє (англ., autoregressive integrated moving average)

AWS – Amazon Web Services

HPC – високопродуктивні обчислення (англ., High Performance Computing)

IaaS – інфраструктура як послуга (англ., Infrastructure as a Service)

IoT – інтернет речей (англ., Internet of Things)

HTTP – протокол передачі гіпертексту (англ., HyperText Transfer Protocol)

LR – лінійна регресія (англ., Linear Regression)

ML – машинне навчання (англ., Machine learning)

MPICH – Message Passing Interface CHameleon

NN – нейронна мережа (англ., Neural Network)

PaaS – платформа як послуга (англ., Platform as a service)

RL – навчання з підкріпленням (англ., Reinforcement Learning)

SaaS – програмне забезпечення як послуга (англ., Software as a service)

SLA – угода про рівень послуг (англ., Service-Level Agreement)

SOAP – простий протокол доступу до об'єктів (англ., Simple Object Access Protocol)

SVM – метод опорних векторів (англ., Support Vector Machines)

QoS – якість обслуговування (англ., Quality of Service)

ВСТУП

У жовтні 2021 року 4,88 мільярда людей по всьому світу були підключені до Інтернету. Це майже 62% світового населення, і їх число зростає зі швидкістю 4,8% на рік [1]. Більшість людей використовують смартфони для підключення до Інтернету, але використання портативних комп'ютерів, мобільних пристроїв, розумного дому та продуктів безпеки, пристроїв Інтернету речей (IoT) тощо постійно зростає.

Для обробки величезних обсягів Інтернет-даних, що виникають, компанії будь-якого розміру шукають способи забезпечити надійне та безпечне зберігання та обробку даних із легким адмініструванням внутрішніх і зовнішніх даних. Цього можна досягти за допомогою хмарних обчислень. Хмарні обчислення — це доставка віртуальних ресурсів (серверів, сховищ, програм, онлайн-сервісів тощо) через Інтернет централізованими системами, розташованими далеко від кінцевих користувачів. Для простоти в роботі всі апаратні та прогрмамні ресурси будемо називати «ресурсами».

Тенденція переміщення обчислень, зберігання даних і програм у хмарні центри обробки даних сприяла повсюдному доступу до спільних обчислювальних ресурсів і ресурсів зберігання за запитом. Серед численних переваг послуги хмарних обчислень допомогли покращити робочі процеси IT, зменшити витрати за рахунок видалення локальних серверів, підвищити масштабованість, спростити технічне обслуговування, підвищити швидкість розгортання, зменшити час простою, підвищити безпеку та сприяти мобільності та гнучкості робочих методів.

Постійне зростання використання Інтернету, обробки даних, зберігання та прогрес сучасних технологій призвели до збільшення використання хмарних обчислень з часом. Прогнозується, що світовий ринок хмарних обчислень зросте з 445,3 мільярдів доларів США у 2021 році до 947,3 мільярдів доларів США до 2026 року [2]. Через пандемію COVID-19 цифрова

трансформація бізнесу стала складнішою та актуальнішою. Через економічні наслідки пандемії великі компанії пропонують клієнтам економічно ефективні цифрові рішення.

Потреба в хмарних послугах зросла в результаті раптового закриття офісів, шкіл і підприємств через пандемію. До 2025 року Gartner [3] прогнозує, що понад 85% компаній приймуть стратегію «спершу хмара», причому понад 95% нових робочих навантажень розгортатимуться на власних хмарних платформах (порівняно з 30% у 2021 році). Протягом наступних кількох років дохід від хмарних технологій перевищить дохід від нехмарних технологій на IT-ринках.

Метою роботи є дослідження застосування ML в управлінні георозподіленими хмарними центрами обробки даних, виявлення проблем, які залишаються невирішеними, щоб висвітлити можливості для майбутніх робіт в цій галузі. Сфера діяльності не обмежується конкретним контекстом, сценарієм або екосистемою в цьому відношенні, а радше ми зосереджуємося на різноманітних техніках, запропонованих для управління георозподіленими центрами обробки даних і обговорюємо, як ці методи впливають на майбутнє покоління архітектур хмарних обчислень.

Метою роботи є дослідження застосування машинного навчання (ML) в управлінні георозподіленими хмарними центрами обробки даних, виявлення проблем, які залишаються невирішеними, щоб висвітлити можливості для майбутніх робіт в цій галузі.

Задачі роботи можна підсумувати таким чином:

- виявити основні проблеми управління георозподіленими хмарними центрами обробки даних;

- дослідити сучасніші методи ML для профілювання робочого навантаження;

- сформулювати майбутні напрямки для ML в управлінні георозподіленими хмарними центрами обробки даних.

Об'єктом досліджень є процес управління георозподіленими хмарними центрами обробки даних.

Предмет досліджень: методи машинного навчання для управління інформаційними системами в хмарному середовищі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Георозподілені хмарні центри обробки даних

Постачальники хмарних послуг все частіше розробляють додаткові центри обробки даних, щоб сприяти буму хмарних обчислень, збільшуючи можливості своїх пропозицій хмарних обчислень. Хоча колись централізовані центри обробки даних були звичним явищем, останніми роками спостерігається тенденція до розосередження центрів обробки даних по різних географічних регіонах [4], [5].

Глобальне розміщення центрів обробки даних має багато переваг. Це наближає компанії до їхніх клієнтів, покращуючи продуктивність (наприклад, затримку) і знижуючи витрати на мережу. Завдяки резервуванню, яке забезпечують розподілені центри обробки даних, він також забезпечує чудову стійкість до непередбачуваних збоїв (наприклад, екологічних катастроф). Це дозволяє компаніям обробляти дані за різними критеріями відповідно до вимог певної країни/регіону, галузі чи робочого процесу. Це також сприяє дотриманню регіональних законів про зберігання даних і конфіденційності, оскільки деякі дані чи інформацію заборонено поширювати чи обробляти за межами певного географічного регіону. Корпорації можуть використовувати розподілену хмару для застосування обмежень, дотримуючись при цьому правил конфіденційності даних. Ще одна вагома причина для географічного розподілу центрів обробки даних — це зменшення витрат на електроенергію за рахунок використання цін на електроенергію та відмінностей у відновлюваних джерелах енергії в різних регіонах. З огляду на вищезазначені причини та зростаючу тенденцію розповсюдження центрів обробки даних у всьому світі, у цій роботі ми розглядаємо дослідницькі зусилля та останні розробки, які зосереджуються на «георозподілених» хмарних центрах обробки даних.

1.2 Основні проблеми георозподілених хмарних служб

Ключові постачальники хмарних послуг, такі як Amazon Web Services (AWS), Microsoft Azure, Google Cloud і Alibaba Cloud, керують величезними центрами обробки даних по всьому світу. Усі ці центри обробки даних оснащені процесорами з високою кількістю ядер, терабайтами оперативної пам’яті та петабайтами пам’яті. Останній звіт Bloomberg [6] показує, що центри обробки даних зараз забезпечують приблизно 1% споживання електроенергії в усьому світі, і очікується, що цей відсоток зросте до 3-8% у наступному десятилітті.

Кількість енергії, що споживається центрами обробки даних у всьому світі, подвоїлася за останнє десятиліття, а деякі дослідження прогнозують, що вона може зрости втричі або навіть у чотири рази в наступному десятилітті [7]. Річні витрати на електроенергію для живлення центрів обробки даних постійно зростають, тому мінімізація таких витрат є критичною. Наприклад, очікується, що до 2023 року дата-центри Китаю будуть споживати більше енергії, ніж вся Австралія [8]. Подібні річні енерговитрати іноді перевищують витрати на придбання обладнання ЦОД. Таким чином, визначення того, як зробити управління компонентами хмарних обчислень економічно ефективним, з економією енергії та збалансованими затратами, стало суттєвою метою дослідників, які працюють у сфері хмарних обчислень.

Інша проблема полягає в забезпеченні своєчасного доступу до хмарних сервісів без порушення SLA та QoS. Угода про рівень обслуговування (SLA) – це контракт між постачальником послуг і клієнтом, який визначає обсяг послуг, наприклад, якість послуг, доступність і обов’язки постачальника. Від визначення послуг до завершення угоди SLA зазвичай мають багато компонентів і також визначаються на багатьох рівнях.

Якість обслуговування (QoS) — це опис загальної продуктивності служби, наприклад комп’ютерної мережі або служби хмарних обчислень, за

спостереженнями її користувачів. Багато компонентів мережевих послуг, таких як хороша пропускна здатність, доступність, затримка, затримка передачі, доставка з порушенням порядку, помилки та втрата пакетів, часто використовуються для оцінки якості обслуговування для хмарних служб.

Збір і обробка великих обсягів даних у хмарних центрах обробки даних часто призводить до великих затримок і використання пропускної здатності мережі. Згідно з аналізом Amazon, збільшення часу завантаження веб-сторінки на 100 мс коштувало компанії 1% її продажів. Google виявив, що додавання 0,5 секунди до часу, необхідного для створення результату пошукового запиту, зменшило трафік на його сайт на 20% [9]. Для інтерактивних і критичних за часом програм менші затримки можуть значно покращити якість роботи. Це підкреслює важливість запобігання порушенням SLA та QoS.

Підходи до хмарного керування часто використовуються для зменшення споживання енергії, викидів вуглецю, затримки мережі, часу виконання запитів, операційних витрат центру обробки даних, а також порушень QoS і SLA. Однак оркестрування складного, різнорідного, великомасштабного програмного забезпечення та апаратних систем у складних мережевих налаштуваннях, а також автоматизація обчислювальних платформ, залучених до георозподілених хмарних обчислювальних систем, є дуже складною проблемою. Дослідники в цій галузі використовують евристики та алгоритми для таких підпроблем, як міграція віртуальної машини (VM) з низькими накладними витратами, мінімізація черги, міграція служби/робочого навантаження з урахуванням затримки та оптимальний розподіл ресурсів продуктивності/енергії. Сучасні математичні методи вирішення таких проблем використовують багато стратегій, наприклад евристичні методи, метаевристичні алгоритми, ймовірнісні алгоритми, гібридні алгоритми та методи динамічного програмування.

Математичні методи мають обмежену релевантність у великомасштабних динамічних високопродуктивних системах і їх важко

масштабувати до георозподілених рівнів. У результаті постачальники хмарних послуг були змушені досліджувати інтелектуальні альтернативи на основі даних і машинного навчання.

1.3 Огляд методів машинного навчання

Машинне навчання (ML) означає здатність машини навчатися на основі даних. Без явного програмування ML дозволяє комп'ютерам навчитися виконувати завдання, такі як прогнозування результатів і класифікація об'єктів. Одне з ключових припущень полягає в тому, що, використовуючи навчальні дані та статистичні підходи, можна розробити алгоритми, які можуть передбачати майбутні або раніше невідомі значення. ML повільно зарекомендував себе як фактичне рішення в різноманітних корпоративних програмах, включаючи розпізнавання мовлення, безпілотні автомобілі, бізнес-аналітику, веб-пошук, виявлення шахрайства, рекомендації щодо купівлі та обслуговування клієнтів, і це лише кілька прикладів. Доступність величезних наборів даних і постійний прогрес як у теорії машинного навчання, так і в обчислювальних можливостях серверів значною мірою відповідають за це досягнення. В останні роки великі технологічні компанії зробили значні інвестиції в ML і штучний інтелект (ШІ), запровадивши нові послуги та здійснивши серйозні реорганізації для стратегічного розміщення ШІ у своїх організаційних структурах [10].

Сьогодні широко використовуються три широкі парадигми машинного навчання: контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням (RL). Мета навчання під наглядом полягає в тому, щоб знайти функцію, яка найкраще наближає зв'язок між входами та виходами, враховуючи вибірку вхідних даних і бажаних виходів. Навчання без нагляду, з іншого боку, не має позначених результатів і замість цього прагне зробити висновок про притаманну структуру, що існує в наборі точок вхідних даних. RL – це алгоритм ML, який винагороджує бажану поведінку, одночасно

караючи небажану. У наступних підрозділах ми коротко представляємо алгоритми машинного навчання, включені в ці групи.

1.3.1 Кластеризація

Кластеризація – це процес поділу набору точок даних на групи, щоб точки даних в одній групі були більше схожі одна на одну та відрізнялися від точок даних в інших групах. Методи кластеризації – це неконтрольовані алгоритми ML. Метод k-середніх є одним з методів кластеризації, що використовує алгоритм, який має розділяти точки даних в k-кластери, причому кожна точка даних належить до кластера з найближчим середнім (центроїд кластера або центр кластера), яке служить прототипом кластера. Спектральна кластеризація – ще один метод, який бере свій початок у теорії графів і використовується для виявлення кластерів вузлів у графі на основі ребер, які їх з’єднують.

1.3.2 Регресія

Існують методи, що використовує регресією, які створюють прогноз безперервного або кількісного вихідного значення. Регресія є підгалуззю керованого машинного навчання. Лінійна модель, наприклад, яка передбачає лінійний зв’язок між вхідними змінними і одну вихідну змінну, відома як лінійна регресія (LR). Зв'язок між вхідною змінною x і вихідною змінною y розглядається як поліном n-го ступеня і зветься поліноміальною регресією (Poly LR). Оператор найменшого абсолютного скорочення та вибору (LASSO) це підхід регресійного аналізу, який робить як відбір змінних, так і регулярізацію для покращення точності прогнозування та інтерпретації статистичної моделі.

У методі дерева рішень (DTs), метою є вивчення простих правил прийняття рішень на основі атрибутів даних для розробки моделі, яка

передбачає значення цільової змінної. Методи випадкового лісу (RF) використовують алгоритм ансамблевого навчання для регресії. Метод ансамблевого навчання поєднує прогнози з кількох алгоритмів ML, щоб отримати більш точний прогноз, ніж за однією моделлю.

Радіочастотна регресія працює шляхом навчання великої кількості DT. Підвищення градієнта – це метод регресії, який генерує модель прогнозування з ансамблю слабких моделей прогнозування, найчастіше DT. Отримана методика називається "дерева градієнтної регресії" (GBRT) і використовується, коли DT є слабким учнями. Екстремальне посилення градієнта (XGBoost) – це ефективна та популярна реалізація дерев із посиленням градієнта з відкритим кодом. Інженерна мета XGBoost полягає в тому, щоб розширити обмеження обчислювальних ресурсів для розширених алгоритмів дерева.

Метод k - найближчих сусідів (k-NN) представляє техніку регресії, де зближується зв’язок між незалежними змінними та безперервними результатами шляхом усереднення спостережень сусідства. Машина підтримки векторів (SVM) регресія – це контрольований метод навчання для прогнозування дискретних значень. Ключова концепція регресії SVM полягає у визначенні найкращої лінії. Гіперплощина з найбільшою кількістю точок є найкращою лінією в цьому випадку. Авторегресійне інтегроване ковзне середнє (ARIMA) є узагальненням простішої авторегресивної ковзної середньої (ARMA) і додає поняття інтеграції. Обидві моделі використовуються для підгонки даних часових рядів, щоб краще зрозуміти або передбачити майбутні моменти в ряді даних. Сезонна APIMA (SARIMA) – це варіант ARIMA, який підтримує однофакторні дані часових рядів із явним сезонним компонентом.

Загалом, використання регрессії є дуже складною задачею у випадку багатопараметричної оптимізації і дає хороші результати при зменшенні критерієв оптимізації.

1.3.3 Нейронні мережі

Нейронні мережі (NN), також відомі як штучні NN (ANN), є класом алгоритмів ML, назва та структура яких походять від людського мозку. Ці мережі нагадують мережі біологічних нейронів і імітують спосіб, у який вони спілкуються один з одним. NN складаються з шарів вузлів, що складаються з вхідного рівня, одного або кількох прихованих шарів і вихідного рівня. Кожен вузол, або штучний нейрон, з'єднаний з іншими і має пов'язану з ним вагу та поріг. Якщо вихідні дані вузла перевищують певне порогове значення, вузол активується, а дані надсилаються на наступний рівень мережі. В іншому випадку дані не надсилаються на наступний рівень мережі. Щоб навчатися та підвищувати свою точність з часом, НМ залежать від даних навчання. Однак після того, як ці алгоритми навчання були налаштовані на задану точність, вони стали потужними інструментами в області інформатики та штучного інтелекту, дозволяючи нам швидко кластеризувати, класифікувати або прогнозувати дані. У порівнянні з ручною ідентифікацією аналітиками-людьми, завдання в області розпізнавання голосу або зображення можуть займати секунди з NN, а не години.

Мережі стохастичної конфігурації (SCN) використовують механізм навчання під наглядом для автоматичної та швидкої побудови універсальних апроксиматорів, які досягають багатообіцяючої продуктивності для вирішення задач регресії. Багатошаровий персептрон (MLP) є типом прямої NN. Перцептрон — це модель одного нейрона, який служив попередником більших НМ. Глибоке навчання (DL)є підмножиною ML, яка по суті є мережею, яка має три або більше рівнів. Тоді як одношарова NN може створювати приблизні прогнози, додаткові приховані шари можуть допомогти оптимізувати та підвищити точність.

Згорткові нейронні мережі (ConvNet/CNN) використовують DL-модель, яка може приймати вхідне зображення, призначати релевантність

(вивчаються ваги та зміщення) численним об’єктам на зображенні та відрізняти один від іншого.

Повторювана нейронна мережа (RNN) – це мережева мережа, яка працює з часовими рядами або послідовними даними. RNN мають приховані стани і дозволяють використовувати попередні виходи як входи. Мовний переклад, створення субтитрів до зображень, обробка природної мови (NLP) і розпізнавання мовлення – лише деякі із завдань, для вирішення яких широко використовуються ці RNN. Довга короткочасна пам'ять (LSTM) є типом RNN з більш складною структурою нейронів. LSTM можуть обробляти передбачення з окремими вхідними даними (наприклад, зображеннями), а також довгими послідовностями даних (такими як мова чи відео). Комірки LSTM використовують механізм стробування з трьома воротами: вхід, вихід і забуття. Закриті рекурентні підрозділи (GRU) подібні до LSTM, але мають лише дві операції: скидання та оновлення. Оскільки GRU мають менше операцій, ніж LSTM, вони менш складні. Якщо набір даних невеликий, GRU зазвичай є кращими; в іншому випадку для великих наборів даних LSTM часто демонструють кращу продуктивність.

1.3.4 Навчання з підкріпленням

Навчання з підкріпленням (RL) – це техніка ML, яка дозволяє агенту навчатися методом проб і помилок в інтерактивному середовищі, використовуючи зворотний зв’язок від власних дій і досвіду. Агент представляє алгоритм RL, тоді як навколишнє середовище відноситься до предмета, на який діє агент. Середовище надсилає стан агенту, який потім виконує дії у відповідь на цю умову на основі своїх знань. Потім середовище надсилає агенту пару наступних відомостей про стан і винагороду. Щоб оцінити свою останню дію, агент оновить свої знання, використовуючи винагороду, яку повертає середовище. Цикл триває, поки середовище не

надішле термінальний стан, завершуючи епізод. Нижче наведено кілька важливих термінів, які використовуються в RL:

- дія: всі ходи доступні агенту;
- стан: поточний стан, повернутий середовищем;
- нагорода: негайна віддача (зворотний зв'язок) з оточення для оцінки якості попередньої дії;
- політика: стратегія агента для визначення наступного курсу дій на основі поточного стану;
- значення: прогнозований довгостроковий прибуток зі знижкою, на відміну від короткострокової винагороди;
- Q-значення або цінність дії: довгострокове повернення поточного стану, вчинення дії;
- модель: точка зору агента на середовище, яка перетворює пари стан-дія в розподіли ймовірностей за станами.

Варто зазначити, що не кожен агент RL використовує модель свого оточення. На основі моделі RL намагаються змоделювати середовище, а потім визначити найкращу політику на основі вивченої моделі. З іншого боку, в безмодельній RL, агент використовує досвід проб і помилок, щоб вивести оптимальну політику. Оптимізація політики та Q-навчання є двома основними методологіями представлення агентів за допомогою безмодельного навчання з підкріпленням. У випадку оптимізації політики (наприклад методи ітерації політики, функція політики, яка відображає стани на дії), вивчення виконується безпосередньо агентом. Функція значення не використовується для визначення політики. При градієнтній політики (PG), градієнтний спуск використовується для оптимізації параметризованої політики щодо очікуваного прибутку (довгострокова кумулятивна винагорода).

Актор-критик представляє ще один метод оптимізації політики, де критик оцінює значення дії (Q-значення) або функцію значення стану, а актор оновлює розподіл політики в напрямку, запропонованому критиком.

Алгоритм Q-навчання намагається визначити найкращу дію з урахуванням поточної ситуації за допомогою таблиці пошуку під назвою Q-таблиця, де зберігається максимальна очікувана винагорода за дії в кожному стані.

Нейронна мережа Deep Q (DQN) поєднує Q-навчання з NN. Замість побудови Q-таблиці NN наближають Q-значення для кожної дії на основі стану. Глибокі детерміновані градієнтна політика (DDPG) – це алгоритм, який вивчає як Q-функцію, так і політику одночасно. Він вивчає Q-функцію з даними, що не відповідають політиці, а потім використовує Q-функцію для вивчення політики.

Глибоке навчання з підкріпленням (DRL) – це поєднання навчання з підкріпленням і глибокого навчання. Глибоке навчання включено в DRL, що дозволяє агентам приймати рішення на основі неструктурованих вхідних даних без необхідності вручну створювати простір станів. Алгоритми DRL можуть обробляти величезні обсяги вхідних даних і визначати, які дії потрібно виконати для оптимізації мети. Використання RL в багатоагентній системі відоме як Multi-Agent Reinforcement Learning (MARL). Як правило, агенти приймають рішення на основі попереднього досвіду. Тут агент, зокрема, повинен розуміти, як координувати роботу з іншими агентами, подібно до теорії ігор.

1.4 Постановка мети та задач роботи

Метою роботи є дослідження застосування ML в управлінні георозподіленими хмарними центрами обробки даних, виявлення проблем, які залишаються невирішеними, щоб висвітлити можливості для майбутніх робіт в цій галузі. Сфера діяльності не обмежується конкретним контекстом, сценарієм або екосистемою в цьому відношенні, а радше ми зосереджуємося на різноманітних техніках, запропонованих для управління георозподіленими центрами обробки даних і обговорюємо, як ці методи впливають на майбутнє покоління архітектур хмарних обчислень.

Такі архітектури дуже неоднорідні та складні, аж до того, що повністю розроблений підхід до моделювання та прогнозування може бути надмірно складним, якщо не зовсім марним. У таких середовищах хмарна оптимізація відіграє дуже важливу роль. Процес правильного визначення та призначення правильних ресурсів робочим навантаженням (завданням або програмам) або навпаки, відомий як хмарна оптимізація. Ми особливо зосереджуємося на техніках, які використовують ML для прогнозування, класифікації, профілювання ресурсів і робочих навантажень, розміщення додатків/завдань, міграції, хмарної оптимізації та гібридних методів ML у всьому спектрі георозподілених обчислювальних сценаріїв.

Відповідно, у цій роботі деконструюється та аналізується проблема надійного георозподіленого керування хмарним центром обробки даних перед дослідженням підходів на основі машинного навчання, які вирішують цю проблему (або її частину). Внесок цієї роботи можна підсумувати таким чином:

- ми розбиваємо проблему управління георозподіленим хмарним центром обробки даних на різні підпроблеми;
- наше дослідження всебічно розглядає найсучасніші методи ML для профілювання робочого навантаження та ресурсів, прогнозування параметрів і хмарної оптимізації;
- ми обговорюємо майбутні напрямки для ML в управлінні георозподіленими хмарними центрами обробки даних.

2 МЕТОДИ УПРАВЛІННЯ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ У ХМАРНОМУ СЕРЕДОВИЩІ

2.1 Методи управління хмарним середовищем

Є багато оглядових статей, які досліджували управління ресурсами, керування навантаженням, безпеку, розподіл віртуальних машин, енергоефективність і балансування навантаження для хмарних обчислень. Деякі останні огляди [11], [12], [13] представили багатовимірну класифікацію існуючих рішень для керування хмарними ресурсами на основі їх архітектури планування, цілей і методів.

У [14] порівнюються, класифікуються та аналізуються різні методи кластеризації, оптимізації та машинного навчання, які використовуються в управлінні хмарними ресурсами для підвищення енергоефективності та продуктивності.

У [15], [16] і [17] надається огляд існуючих алгоритмів балансування навантаження хмарних обчислень і показників їх продуктивності. У цих роботах також обговорювалися переваги та недоліки обраних алгоритмів балансування навантаження.

У [18] представлено дослідження різних доступних методів розподілу ресурсів, методів балансування навантаження, методів планування та методів контролю доступу для хмарних обчислень разом з аналізом переваг і недоліків кожного методу.

Ґрунтуючись на нашому аналізі цих робіт, ми помітили, що деякі огляди враховують кілька досліджень на основі машинного навчання, але вони не розглядають всебічно керування хмарою на основі машинного навчання, яке демонструє важливість методів машинного навчання в хмарних обчисленнях.

2.2 Методи управління хмарним середовищем на основі машинного навчання

Кілька останніх опитувань розглядали хмарне управління на основі машинного навчання. Серед них [19] розділив проблему планування хмарних ресурсів на різні цілі, такі як оптимізація часу, оптимізація споживання енергії та оптимізація балансування навантаження. Автори обговорили та дослідили архітектуру RL та Deep Reinforcement Learning (DRL). Огляд поступово оцінював різні архітектури RL і DRL, пропонуючи уніфіковане подання для наступних RL або DRL у плануванні ресурсів хмарних обчислень.

У [20] автори розглянули, як тема надійного забезпечення ресурсами в об’єднаних периферійних хмарних системах розглядалася в науковій літературі та які стратегії були використані для підвищення надійності розподілених програм у різнорідних і гетерогенних мережевих середовищах. Під час опитування обговорювалося, як характеристиці, управлінню та контролю складних розподілених систем, що використовують методології машинного навчання, приділяється значна увага через складність проблеми. Опитування організовано навколо проблеми забезпечення ресурсами, що складається з трьох частин: характеристика та прогнозування робочого навантаження, розміщення компонентів, консолідація системи, еластичність застосувань і виправлення помилок.

Автори в [21] поділяють управління хмарними ресурсами на чотири основні категорії: забезпечення ресурсами, планування ресурсів, розподіл ресурсів та енергоефективність. Після представлення порівняння робіт у цих категоріях вони, запропонували найбільш прийнятну модель ML для кожної категорії.

У [22] представлений широкий огляд робіт, які використовують методи ML для управління ресурсами в хмарних обчисленнях. На відміну від інших робіт, у цьому огляді класифіковано та проаналізовано включені дослідження

на основі трьох типів моделей ML: контрольоване навчання, неконтрольоване навчання та RL. У цьому огляді було додатково оцінено методи ML за типом функцій, які вони використовують, результатами, які вони створюють, і цілями, які вони намагаються оптимізувати.

У [23] представлено детальний огляд проблем управління ресурсами хмарних обчислень на основі ML. Під час дослідження також обговорювалися сучасні підходи до вирішення цих проблем, а також їхні переваги та недоліки. Нарешті, автори запропонували можливі майбутні напрямки досліджень на основі визначених обмежень.

2.3 Управління георозподіленим хмарним центром обробки даних на основі машинного навчання

Незважаючи на те, що дуже корисно використовувати ML у системах розподілених хмарних обчислень і оцінювати прогрес останніх досліджень і проектування цих систем, жодна з останніх статей не описує деталі хмарного керування на основі ML для георозподілених центрів обробки даних, а також чи вникають вони в специфіку викликів і проблем, які існують у поточному сучасному стані та майбутніх напрямках досліджень цієї теми.

Як обговорювалося в роботі [24], георозподілені хмарні центри обробки даних можуть використовувати різницю між цінами на електроенергію, чистим вимірюванням, розподілом цін на піковий попит, доступністю ресурсів, цінами на передачу даних/мережу, спільним розміщенням серверів і відновлюваною енергією на різні місця розташування центрів обробки даних. Крім того, георозподілені центри обробки даних забезпечують більшу стійкість до незапланованих збоїв завдяки своїй надмірності.

У результаті можна вважати, що зараз саме час представити комплексне дослідження, у якому обговорюються різні алгоритми машинного навчання та їх застосування до різних контекстів проблем у

сценаріях керування хмарою для георозподілених центрів обробки даних, а також їхні недоліки, проблеми та майбутні напрямки. Таким чином, перш ніж рухатися далі з новими ідеями в цьому відношенні, дослідники можуть використати цю роботу, щоб оцінити поточні сценарії машинного навчання в георозподіленому хмарному управлінні та їхні обмеження.

2.4 Система управління хмарних систем

Надійне керування хмарою для центрів обробки даних є складною проблемою, особливо коли її розглядають у георозподіленому хмарному середовищі, де розподілена інфраструктура використовується для розміщення численних різнорідних робочих навантажень, що надходять із різних географічних регіонів. Кожне робоче навантаження, як правило, має свій власний набір вимог, і існує висока ймовірність того, що керування операціями або налаштування продуктивності одного робочого навантаження, яке використовує один або кілька ресурсів, матиме певний вплив на інші, наприклад, ефекти перешкод від спільного розташування робочого навантаження [25].

Крім того, проблема включає розгортання та роботу з навантаженнями в географічно розосереджених неоднорідних ресурсних середовищах. У таких складних середовищах повне та оптимальне вирішення проблеми є складним завданням, і життєздатні рішення повинні включати комбінацію різних методів для досягнення передбачуваності та керованості як робочого навантаження (програмного забезпечення), так і продуктивності (апаратного) ресурсу. Це вимагає ретельного дослідження всіх частин проблеми, щоб отримати цілісне розуміння проблеми та викликів під час розробки рішення. Для цього, як показано на рисунку 2.1, ми розділяємо та обговорюємо аспекти георозподіленого керування хмарою на три категорії: профілювання/кластеризація, прогнозування параметрів та оптимізація хмари.

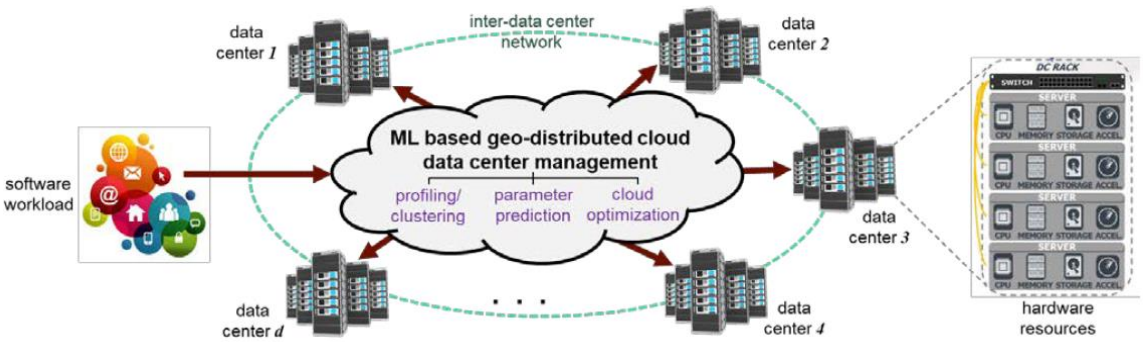

Рисунок 2.1 – Управління георозподіленим хмарним центром обробки даних на основі машинного навчання

2.4.1 Профіль робочого навантаження та ресурсів

Хмарні робочі навантаження можна класифікувати на різні типи на основі різних характеристик і перспектив, таких як транзакційні чи нетранзакційні, послідовні чи непослідовні/випадкові та інтенсивності використання пам’яті. З точки зору типу ресурсу та обсягу, потреба в ресурсах змінюється залежно від робочого навантаження. Компоненти робочого навантаження хмарної програми призначені для виконання певних дій, які, імовірно, використовують різні типи ресурсів (наприклад, ЦП або ГП). Крім того, завдяки широкому географічному розподілу центрів обробки даних у мережах, робочі навантаження можуть розподілятися різними способами між різними розташуваннями центрів обробки даних. Ці характеристики призводять до високої складності та проблем у профілюванні робочого навантаження та ресурсів.

Здатність розуміти робоче навантаження та поведінку ресурсів за допомогою ML допомагає підвищити продуктивність і надійність програми, дозволяючи хмарним постачальникам забезпечити наявність достатніх ресурсів для підтримки будь-яких майбутніх робочих навантажень. Таким чином, багато методів управління хмарою мають бути профільовані з точки зору робочого навантаження та ресурсів або згруповані в конкретну категорію, перш ніж техніку можна буде ефективно застосувати.

2.4.2 Прогнозування параметрів

В епоху поширених обчислень Edge та IoT робочі навантаження, що виконуються в хмарних середовищах, постійно зростають як за складністю, так і за обсягом. Неоднорідність робочих навантажень і кількість користувачів, які використовують певну службу хмарних додатків у різний час, вносить зміни в робочі навантаження хмари та відповідні ресурси, які робочі навантаження повинні використовувати в різних розташуваннях центрів обробки даних. Іншими словами, хмарні програми відчувають варіації надходження робочого навантаження та використання ресурсів у спосіб, який важко передбачити.

Здатність прогнозувати просторово-часовий розподіл майбутніх робочих навантажень або їх параметрів заздалегідь за допомогою ML приносить переваги георозподіленим рішенням для управління хмарою, тобто система отримує можливість проактивного масштабування, щоб задовольнити вимоги до ресурсів у реальному часі. Або, у деяких випадках, робоче навантаження можна ефективно розподілити між ресурсами, якщо відомі майбутні стани використання ресурсів. Таким чином можна покращити розподіл робочого навантаження та оптимізувати використання ресурсів, гарантуючи вимоги QoS і SLA, що, у свою чергу, підвищує ефективність і продуктивність георозподіленого хмарного керування.

2.4.3 Хмарна оптимізація

Методи хмарної оптимізації зазвичай реалізуються за допомогою статичних або динамічних алгоритмів. У той час як розміщення робочого навантаження або правила розподілу ресурсів заздалегідь визначені в статичній схемі, у динамічній схемі робоче навантаження динамічно обробляється та розподіляється між ресурсами. Незважаючи на те, що статичні методи прості та зазвичай більш стабільні, вони не досягають

оптимальних результатів із різнорідними робочими навантаженнями та розподіленими ресурсами, що підлягають нестабільному розподілу робочого навантаження та непередбачуваній поведінці ресурсів.

Методи хмарної оптимізації розподіляють або звільняють ресурси на льоту відповідно до вимог робочого навантаження, таким чином зменшуючи операційні витрати, зберігаючи вимоги QoS і SLA. В ідеалі хмарні ресурси та робочі навантаження мають бути динамічно конфігурованими, програмованими та оптимізованими за допомогою ML без жодного втручання людини. Важливим компонентом керування хмарними ресурсами є віртуальні мережеві функції (VNF), які створюються на відкритих обчислювальних платформах і запускають віртуальні мережеві служби на звичайних серверах у хмарних центрах обробки даних.

Віртуалізовані маршрутизатори, брандмауери, оптимізація WAN і служби трансляції мережевих адрес (NAT) — усе це приклади VNF. Віртуальні машини (VM), що працюють на стандартному програмному забезпеченні інфраструктури віртуалізації, наприклад VMWare або віртуальна машина на основі ядра (KVM), запускають більшість VNF. Постачальники мережевих послуг повинні налаштовувати мережевий трафік відповідно до деталізації на рівні потоку, знижуючи мережеві витрати та забезпечуючи хорошу взаємодію з користувачем. Оскільки робоче навантаження та вимоги до ресурсів змінюються з часом у хмарних середовищах, також важливо динамічно масштабувати екземпляри VNF за допомогою ML.

2.5 Дослідження методів управління ресурсами в хмарних системах

ML і хмарні обчислення стали популярними та широко поширеними в промисловості. Нещодавня тенденція полягала в дослідженні того, як ML може допомогти в управлінні хмарою для георозподілених центри обробки даних. В останні кілька років спостерігається постійне збільшення кількості

досліджень у цій галузі. Для цього дослідження було обрано 22 статті, пов’язані з керуванням георозподіленим хмарним центром обробки даних на основі машинного навчання. Статті розділені на чотири основні групи. Таксономічна класифікація літератури, розглянутої в цьому підрозділі, показана на рисунку 2.2.

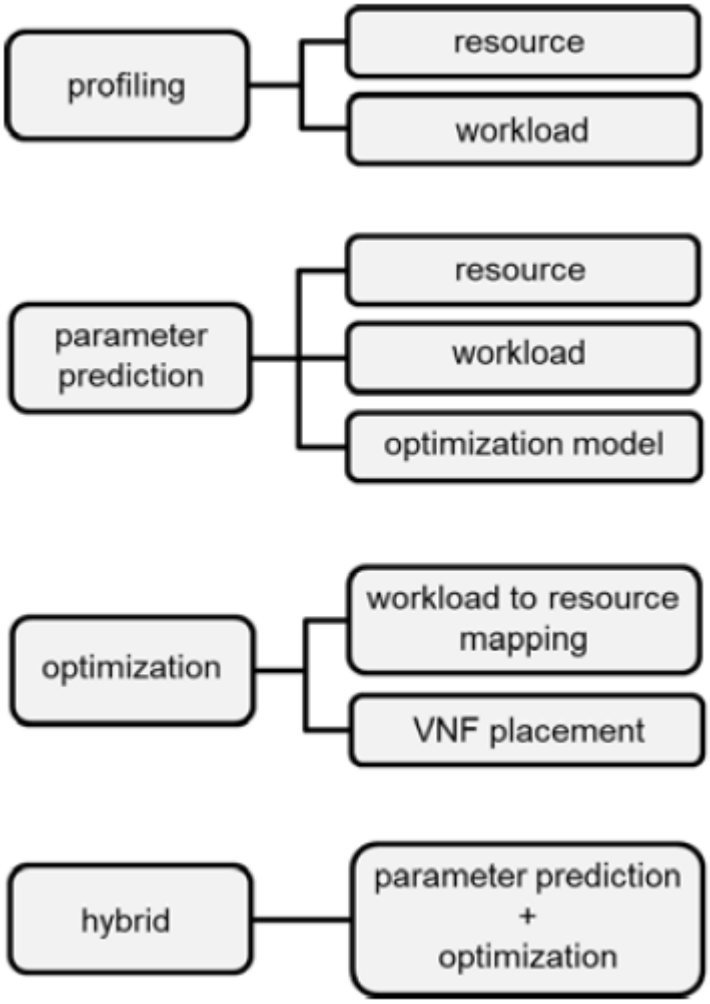

Рисунок 2.2 – Таксономічна класифікація методів управління ресурсами в хмарних системах

2.5.1 Профілювання ресурсів та завдань

Метою роботи [26] є вибір оптимального центру обробки даних (ресурсів) серед набору георозподілених центрів обробки даних на основі

стандартів SLA та потреб користувача. SLA тут складається з набору об’єктів рівня обслуговування (SLO), але в цьому дослідженні чотирма найважливішими з них є вартість, час відгуку, доступність і надійність. Після отримання запиту користувача в георозподіленому хмарному середовищі спочатку дані нормалізуються за допомогою методу нормалізації Z-оцінки, щоб усунути ненормальний розподіл різних SLO. Потім кількість кластерів визначається за чотирма критеріями SLA, де 3844 ЦОД об’єднані в кластери за допомогою k-середнього алгоритму. На наступному кроці, виходячи з довготи та широти місця розташування користувача, вибирається найближчий географічний кластер. Крім того, доступність і надійність максимізуються, вартість і час відгуку мінімізуються, а багатоцільовий алгоритм NSGA-II [27] використовується для вибору найкращого центру обробки даних на основі цих SLO.

Класичні підходи до рівномірного розподілу даних між розподіленими вузлами є нормою в багатьох основних рішеннях для зберігання розподілених даних, таких як Cassandra та розподілена файлова система Hadoop (HDFS). Ці методи можуть створювати перевантаження мережі для служб, що інтенсивно передають дані, знижуючи пропускну здатність системи. Служби з інтенсивним використанням даних, на відміну від робочих навантажень MapReduce, вимагають доступу до багатьох наборів даних протягом кожного сеансу.

Щоб вирішити ці проблеми, автори в [28] запропонували масштабований метод розподілу сервісів з інтенсивним використанням даних між територіально розподіленими хмарними центрами обробки даних. Автори спочатку побудували гіперграф, використовуючи заданий набір елементів даних і набір реєстрацій користувачів, кожен з яких містить шаблон запиту даних, отриманий із розташування центру обробки даних. Гіперграфи дозволяють фіксувати багатосторонні зв’язки, полегшуючи моделювання взаємозв’язків між елементами даних і розташуванням елементів даних у центрі обробки даних.

Набір даних, який використовувався в експериментах, є слідом соціальної онлайн-мережі Gowalla, яка базується на визначенні місця розташування, і є загальнодоступною в репозиторії Stanford Network Analysis Platform (SNAP) [29]. Використовуючи спектральну кластеризацію на гіперграфах, запропонований підхід розділив набір елементів даних на георозподілені розташування хмарних центрів обробки даних, щоб мінімізувати середньозважену вартість одиниці вихідного трафіку з кожного центру обробки даних, затримку між центрами обробки даних для кожної пари центрів обробки даних, вартість одиниці зберігання кожного центру обробки даних і середня кількість центрів обробки даних, до яких мають доступ елементи даних, запитувані в кожному шаблоні запиту. Крім того, алгоритм спектральної кластеризації використовував рандомізовані методи для створення низькорангових наближень матриці гіперграфа, що забезпечує більшу масштабованість при обчисленні спектрів Лапласа гіперграфа.

2.5.2 Прогнозування ресурсів та завдань

У політиці географічного балансування навантаження (GLB) обсяг навантаження, який доставляється до кожного локального центру обробки даних, визначається загальними умовами системи, а не місцевою ситуацією. Регулююча служба (RS) – це тип послуг, що надаються ринком електроенергії для постійного балансування попиту та пропозиції на дуже деталізованому рівні, водночас пропонуючи економічні вигоди споживачам і мережі/комунальному підприємству. Оптимізація RS для локальних центрів обробки даних у георозподіленій хмарі є складною проблемою для вирішення.

В роботі [30] використано метод SARIMA для прогнозування вхідного робочого навантаження на глобальному рівні. Потім автори використали модель ML на основі CNN, щоб спрогнозувати/вивести енергоспоживання локальних центрів обробки даних. Вхідними даними для моделей є зовнішня

температура центру обробки даних, сонячна енергія, ціна електроенергії та швидкість надходження робочого навантаження. Модель використовувала випрямлену лінійну одиницю (ReLU) як функцію активації, оптимізатор Адама та середньоквадратичну помилку (RMSE) як показник оцінки. Для розробки моделі прогнозування в цьому дослідженні використовувалися дві основні політики GLB, а саме Power-aware і Costaware. Цілі цієї політики полягають у мінімізації загального енергоспоживання та загальної вартості хмарної електроенергії, відповідно, у підсумку для всіх георозподілених центрів обробки даних. Після цього це дослідження використовувало результати прогнозів, щоб запропонувати георозподіленим центрам обробки даних можливість брати участь у RS.

Дуже важко передбачити енергоспоживання центру обробки даних, який входить до групи георозподілених хмарних центрів обробки даних, особливо з періодичною доступністю відновлюваної енергії та використанням пристроїв вільного охолодження, таких як повітряні економайзери в центрах обробки даних. В [31] досліджено взаємозв'язок між схемами живлення георозподіленого хмарного центру обробки даних і погодними характеристиками (на основі різних умов та інфраструктури), а набір впливових характеристик, таких як вологість, тиск, температура, хмарність і швидкість вітру, витягуються за допомогою лінійної регресії (LR). Потім отримані функції використовуються для створення моделі прогнозування енергоспоживання на основі NN, яка передбачає модель енергоспоживання кожного центру обробки даних у хмарі.

Робота [32] представила методологію прогнозного розподілу ресурсів на основі ML для георозподілених хмарних центрів обробки даних, яка враховувала затримку та обмеження якості, щоб забезпечити найкращий можливий QoS для глядачів і найнижчу можливу вартість для постачальників потокового контенту. Для початку ця робота запропонувала офлайн-оптимізацію, яка визначила необхідні ресурси перекодування в розподілених місцях поблизу глядачів, збалансувавши QoS і загальну вартість. Потім ML

використовується для створення моделей прогнозування, які передбачають приблизні ресурси перетворення, які будуть заздалегідь зарезервовані в кожному хмарному центрі обробки даних. Використовується п’ять різних алгоритмів, а саме: LSTM, GRU, CNN, MLP і XGBoost. Для оцінки цих регресійних моделей використовується показник середньої абсолютної помилки (MAE). Оскільки неможливо передбачити, яка комбінація гіперпараметрів, наприклад кількість прихованих шарів і нейронів для моделей LSTM, GRU, CNN і MLP, або кількість оцінювачів для моделей XGBoost, є оптимальною, автори створили численні моделі для кожного підходу ML. Найкращі моделі були обрані на основі найкращого значення коефіцієнта детермінації, який використовується для оцінки відповідності регресійних моделей. Нарешті, автори створили Greedy Nearest and Cheapest Algorithm (GNCA) для розподілу вхідних відео в реальному часі серед прогнозованих ресурсів транскодування. Автори використали метадані відео Facebook як вхідні дані та розглянули 10 хмарних екземплярів AWS як георозподілену хмарну платформу для свого аналізу.

Існуючий стек програмного забезпечення був створений для одного центру обробки даних, тому він може бути нечутливим для аналізу даних та до змін ресурсів на льоту в мережах між центрами обробки даних, що може серйозно вплинути на продуктивність запитів. Щоб вирішити цю проблему, [33] запропонував Turbo, легку та ненав’язливу структуру, керовану даними, яка динамічно адаптує плани виконання запитів для аналітики георозподілених даних у відповідь на коливання ресурсів часу виконання в центрах обробки даних. Turbo використовує дві регресійні моделі ML: GBRT і LASSO, щоб точно передбачити витрати часу на виконання планів запитів. Автори створили новий набір даних із 15 тисяч зразків, у кожному з яких записується час, який знадобився для виконання запиту з тесту TPC-H [34], розмір виходу та низка інших параметрів, пов’язаних із виконанням запиту. Вони створили кластер із 33 екземплярів на Google Cloud Compute Engine у восьми різних регіонах. Turbo є ненав’язливим, що означає, що не потребує

жодних змін до існуючого стеку програмного забезпечення для аналізу даних.

Точне прогнозування робочого навантаження має вирішальне значення для динамічного масштабування ресурсів у георозподілених хмарних центрах обробки даних. У [35] автори запропонували стохастичну конфігураційну мережу Savitzky-Golay та Wavelet (SGW-SCN) та інтегрований підхід до прогнозування з фільтрацією шуму та представленням частоти даних для прогнозування робочого навантаження в майбутніх часових інтервалах. У цій роботі вивчалися сліди робочого навантаження в обчислювальних кластерах виробництва Google з травня 2011 року. Протягом 29 днів було зібрано сліди з комірки 12,5 тис. машин, що дало загалом 25 462 157 завдань. Кластери Google організовують усі завдання на кілька рівнів, і кожне завдання має характеристику, яка відображає його релевантність.

Досліджено 12 атрибутів, які були поділені на три типи: безкоштовний (0–1), інший (2–8) і виробничий (9–11). Після цього автори розділили 29 днів на 8352 п’ятихвилинних проміжки часу та підрахували загальну швидкість надходження завдань трьох різних типів. Вони використовували дані перших 25 днів для навчання та останні 4 дні для тестування. Часовий ряд робочого навантаження згладжується за допомогою фільтра Савіцького-Голея, перш ніж розділяється на кілька компонентів за допомогою вейвлет-розкладу в цьому методі. Інтегрована модель розроблена з використанням SCN для характеристики статистичних властивостей тенденцій для різних часових рядів робочого навантаження.

Доступність центру обробки даних є важливою вимогою в георозподілених хмарних середовищах. У цьому контексті профілактичні заходи, які скорочують час, потрібний для ремонту послуги центру обробки даних у разі збою, є критично важливими. Автори в [36] запропонували метод визначення впливу продуктивності мережі на доступність послуг. У цьому дослідженні розглянуто шість центрів обробки даних у всьому світі та

розраховано затримку та тремтіння за допомогою часу проходження в обидві сторони (RTT) і швидкості передачі завантажень у Мбіт/с. Використовуючи алгоритми ML, здатні прогнозувати час для передачі великої кількості даних на основі затримки та тремтіння, ця робота розробила інтелектуальні агенти, здатні вибрати оптимальний центр обробки даних для відновлення служб непрацюючого ресурсу. Автори вибрали регресійні алгоритми LR, Poly LR, DT, RF, SVM і MLP. Вони розробили десять окремих поділів навчання/тесту, оцінили їхню точність за допомогою показника середньої абсолютної похибки (MAE), а потім усереднили результати.

Автори [37] досліджували тему розміщення ВМ, що зменшує споживання енергії, викиди забруднення та затримки доступу для зменшення операційних витрат для постачальників георозподілених хмарних центрів обробки даних. Це завдання визначається як багатоцільова функція з інтелектуальною моделлю ML, створеною для покращення продуктивності представленої моделі розміщення віртуальної машини з урахуванням потужності та вартості (PCVM). PCVM має на меті скоротити загальні витрати шляхом мінімізації зваженої суми двох ключових цілей: викидів вуглецю та витрат на мережевий зв’язок. При виборі між двома нормалізованими ваговими показниками параметрів моделі, пов’язаними з двома цілями, кожен з них може конфліктувати один з одним. Автори застосували метод *k- NN* регресії для прогнозування ваг параметрів моделі.

2.5.3 Оптимізація робочого навантаження ресурсів та розмішення завдань

Проведено дослідження оптимізації в двох підкатегоріях: «робоче навантаження на відображення ресурсів» і «розміщення VNF». Оптимізація зіставлення робочого навантаження з ресурсами означає призначення вхідних робочих навантажень (завдань або програм) найкращим апаратним ресурсам у центрах обробки даних. Оптимізація розміщення VNF стосується

динамічного розміщення VNF на відповідному обладнанні для досягнення певної мети.

Автори роботи [38] запропонували адаптивну техніку управління ресурсами, засновану на RL, з метою досягнення балансу між доходом від QoS і використанням електроенергії в наборі георозподілених хмарних центрів обробки даних. Запропонована методика RL на основі Q-навчання з користувацьким вибором випадкових дій надійна у розподілі робочого навантаження та не вимагає попереднього розподілу вимог до ресурсів. Замість використання порушень SLA ця робота явно моделювала дохід від QoS за допомогою використання диференціального доходу окремих робіт. Автори врахували час, витрачений на міграцію віртуальних машин між центрами обробки даних і вартість мережевих комунікацій. Ця робота досягає швидкого прийняття рішень шляхом точного налаштування зберігання інформації алгоритмів RL та вибору випадкових дій.

У наборі георозподілених середовищ хмарних центрів обробки даних, вартість мережевих ресурсів складно підрахувати, оскільки вона залежить від складних і різнорідних міжцентрових даних WAN-з’єднань. В роботі [39] абстрагували георозподілені центри обробки даних у неповний неорієнтований граф. Задача вибору хмарного центру обробки даних розроблена як задача найкоротшого шляху від вершини клієнта/початку до всіх вершин хмарного центру обробки даних. Довжини ребер на графіку були зважені, де ваги представляють відстань між центрами обробки даних або між центром обробки даних і користувачем. При виборі ЦОД автори враховували ваги ребра (мережевий ресурс) і вершини (обчислювальний ресурс). Для відображення вхідних робочих навантажень у відповідний центр обробки даних вони запропонували метод вибору центру обробки даних на основі Q-навчання, щоб досягти оптимальних витрат на мережу та обчислення.

Простір станів складається з набору вершин графа, а простір дій складається з ребер графа. Після того, як робоче навантаження переходить

від однієї вершини (стану) уздовж краю (дія) до іншої вершини (стану), довжина краю використовується як позитивне значення винагороди. За допомогою Q-таблиці було отримано шлях із найменшим значенням Q, тобто найкоротший шлях між кожним користувачем і центром обробки даних.

У [40] автори представили sCloud, цілісну стратегію управління хмарними ресурсами з урахуванням неоднорідності, з метою максимізації пропускної здатності системи в географічно розподілених самостійних центрах обробки даних. Беручи до уваги доступність відновлюваної енергії та вимоги до QoS, sCloud адаптивно розподіляв робочі навантаження на георозподілені центри обробки даних.

Запропонований метод призначав доступні ресурси різнорідним робочим навантаженням у кожному центрі обробки даних і переміщував пакетні завдання між центрами обробки даних. Проблема розміщення робочого навантаження формулюється як задача оптимізації з обмеженнями, яку можна вирішити за допомогою нелінійного програмування. Крім того, коли екологічне джерело живлення сильно коливалося в різних місцях, автори запропонували стратегію пакетної міграції завдань, щоб ще більше збільшити пропускну здатність системи.

Нарешті, автори вдосконалили sCloud, додавши техніку RL на основі Q-навчання. Хід виконання пакетних завдань в окремих центрах обробки даних представлено елементами в просторі стану RL. Простір дій — це сукупність факторів контролю прогресу. Коефіцієнти контролю прогресу визначали, як довго пакетне завдання може бути відкладено або прискорено порівняно з прогресом виконання завдання, досягнутим у попередньому контрольному інтервалі. Винагорода – це відношення поточної продуктивності до еталонної продуктивності плюс штрафні санкції за порушення SLO рівня роботи. Під час налаштування розподілу ресурсів низька винагорода означає, що виконання завдання може пропустити м’який або жорсткий крайній термін, яких слід уникати. Запропонований

конфігурований диспетчер пакетних завдань на основі RL дозволив динамічно регулювати процес виконання завдання, дотримуючись розкладу.

Завдяки динамізму та широкому георозподілу глядачів і мовників, задоволення всіх запитів за допомогою відповідних ресурсів із георозподілених центрів обробки даних залишається проблемою. Щоб вирішити цю проблему, в [41] попередньо обговорюється підхід на основі прогнозів, який оцінював потенційну кількість глядачів на окремих хмарних сайтах під час трансляції. Ця робота розробила цілочисельну лінійну програму (ILP), засновану на отриманих прогнозах, щоб проактивно та динамічно визначати найкращі центри обробки даних для точного розподілу ресурсів і обслуговування потенційних глядачів, одночасно зменшуючи затримки. Оскільки оптимізація на основі ILP займає багато часу і, отже, неефективна для онлайн-обслуговування, автори представили RL-OPRA, техніку RL на основі DQN у реальному часі. Автори визначили стан як набір індексу центру обробки даних, прогнозованих глядачів, витрат, понесених за попередні часові кроки, і додаткової середньої затримки. Дія вказує індекс сайту, який обслуговуватиме глядачів. Загальна винагорода визначалася шляхом дотримання середньої затримки та зменшення вартості різних дій. Запропоноване рішення на основі RL адаптивно навчилося оптимізувати розподіл і рішення щодо обслуговування під час взаємодії з мережевим середовищем.

У роботі [42] досліджено проблему енергозалежного управління робочим навантаженням у георозподілених хмарних центрах обробки даних, враховуючи генерацію зеленої енергії зі змінною потужністю. Ця робота представила архітектуру децентралізованого керування хмарою на основі блокчейну, щоб зменшити загальну вартість електроенергії, витрати на планування робочого навантаження та міграцію робочого навантаження в центрах обробки даних. У цьому децентралізованому хмарному управлінні робоче навантаження планується самими центрами обробки даних, усуваючи залежність від центрального планування. Крім того, автори запропонували

техніку RL на основі Q-навчання, включену в смарт-контракт, щоб ще більше знизити витрати на енергію. Вектор стану складається з навантаження центрів обробки даних, вектор винагороди складається з витрат енергії, а дії містять усі можливості міграції робочого навантаження. Запропонований підхід RL переносить робочі навантаження між георозподіленими центрами обробки даних, використовуючи інформацію про історичні рішення щодо міграції, і намагається мінімізувати загальну вартість енергії.

Центри обробки даних, розташовані в будівлях, які також забезпечують значний простір для офісних приміщень, розташовані в будівлях змішаного використання. Вигідно використовувати гнучкість планування як у системах опалення, вентиляції та кондиціонування повітря (HVAC), так і в робочому навантаженні центру обробки даних, щоб успішно знизити загальну вартість енергії офісної будівлі.

У [43] автори запропонували дві схеми DRL для зниження загальних витрат на енергію при підтримці цільової кімнатної температури та задоволенні обмежень щодо робочого навантаження центру обробки даних. У разі використання алгоритму DRL для керування системою опалення, вентиляції та кондиціонування повітря в офісі, керування швидкістю повітряного потоку є діями, температури в різних офісних зонах є станами, а вартість енергії для охолодження є винагородою. У випадку розподілу робочого навантаження центру обробки даних на основі DRL, керування активацією (увімкненням/вимкненням) сервера є дією, швидкість виконання сервера є станом, а охолодження центру обробки даних і вартість обчислювальної енергії є винагородою.

Потім автори представили спільний алгоритм управління навантаженням офісу HVAC і центру обробки даних в одному MUB. Вони також створили евристичний метод на основі DRL для забезпечення інтерактивного розподілу робочого навантаження між географічно розподіленими будівлями, що призводить до ще більшої економії енергії. Тут кількість серверів, бажаних для використання в кожному офісному центрі –

це дія, швидкість виконання MUB – це стан, а загальна вартість енергії для всіх MUB – це винагорода.

У [44] автори обговорювали проблему зниження витрат на аналітику великих даних у георозподілених центрах обробки даних, що живляться від відновлюваних джерел енергії з періодичною потужністю. Для вирішення цієї проблеми була запропонована методика RL на основі DQN для планування робочого навантаження. Вектор стану складався з поточної інформації про використання процесора, розмір вільної оперативної пам’яті, вільну пропускну здатність вводу-виводу, погоду та ціну на електроенергію. Вектор дії складається з робочих місць (інформація про те, куди перенести робочі місця). Винагорода складається з витрат на міграцію та витрат на енергію. Крім того, створено дві стратегії для покращення продуктивності фреймворку. Вибірка випадкового пулу (RPS) пропонується для перенавчання мережі з використанням зібраних тренувальних даних, а унікальну структуру односпрямованої мережі мостів (UBN) розроблено, щоб ще більше покращити час навчання шляхом використання історичної інформації, що зберігається в навченій мережі.

Віртуальні машини віртуалізують апаратне забезпечення, щоб дозволити кільком примірникам ОС працювати одночасно. З іншого боку, контейнери дозволяють користувачам віртуалізувати ОС, щоб багато завдань могли працювати на одній машині. Контейнери також дають змогу об’єднати програму з усіма її залежностями (такими як код і бібліотеки), що забезпечує швидший запуск, завершення роботи та міграцію, ніж віртуальні машини. Техніка розміщення контейнерів за замовчуванням у Kubernetes не підходить для територіально розподіленого обчислювального середовища або роботи з мінливістю обчислювальних ресурсів і робочого навантаження.

В [45] запропоновано ge-kube (георозподілене та еластичне розгортання контейнерів у Kubernetes), організоване рішення, яке використовує Kubernetes і додає функції самоадаптації та мережевого розміщення. Це дослідження забезпечило двоетапний цикл керування, у

якому кількість реплік конкретних контейнерів динамічно контролюється на основі часу відповіді програми за допомогою методу RL на основі моделі, а контейнери призначаються георозподіленим обчислювальним ресурсам за допомогою мережевої інформації та механізма розміщення. Для техніки RL стан складається з кількості екземплярів додатків, їхнього використання ЦП і ліміту ресурсів ЦП. Система має вертикальне (додавання або видалення частки ЦП) і горизонтальне (додавання або видалення контейнерів) керування масштабуванням. Винагородою є загальна вартість, яка визначається як зважена сума штрафу за продуктивність (сплачується після пропуску крайнього терміну), вартості використання ресурсу та вартості адаптації. Дослідження запропонувало формулювання проблеми оптимізації та мережеву евристику для вирішення проблеми розміщення, яка явно враховувала незначні мережеві затримки в георозподілених обчислювальних ресурсах для задоволення потреб QoS, чутливих до затримок. У цій роботі було проведено велику серію експериментів з використанням сурогатної програми, що інтенсивно використовує ЦП, і реальної програми (тобто Redis), демонструючи переваги поєднання еластичності та стратегій розміщення, а також переваги прийняття мережевих рішень розміщення.

Автори [46] досліджували оптимізацію енергоспоживання всередині та між центрами обробки даних як з однорідними, так і з різнорідними серверами та робочими навантаженнями. Спочатку вони запропонували оптимізаційну модель для зменшення сумарних витрат на енергію сервера та мережі. Автори використали оптимізаційну модель Ляпунова, щоб перетворити вихідну проблему на добре вивчену проблему стабільності черги для вирішення обмеження викидів вуглецю, пов’язаного з часом. Потім, використовуючи узагальнений метод декомпозиції Бендерса, автори розробили централізоване рішення. Вони також удосконалили модель для обробки неоднорідних робочих навантажень і центрів обробки даних. На початку кожного часового інтервалу ця техніка знаходила можливе рішення для оптимізації енергії.

Однак, запропонований підхід дотримувався цього методу протягом усього часового інтервалу/епохи, незважаючи на зміни в мережевому середовищі (наприклад, поява збоїв). Щоб вирішити цю проблему, автори розробили підхід RL на основі DQN. Стан – це статус центру обробки даних, який включає кількість активних серверів і частоту ЦП. Система розглядає набір центрів обробки даних і контролює, який центр обробки даних вибрати. Винагородою є загальна довгострокова вартість енергії. Запропонований метод RL на основі DQN зміг дослідити та вивчити динамічний характер мережі та прийняти рішення під час виконання на основі обмежених знань.

Ланцюжок функцій сервісу (SFC) спрямований на створення VNF у георозподілених центрах обробки даних і полегшує маршрутизацію між ними. Нещодавно було доведено, що DRL є корисним у сфері SFC. Але поточні алгоритми на основі DRL мають великий простір дії, що спричиняє погану конвергенцію та масштабованість. Деякі дослідження знайшли вирішення цієї проблеми шляхом переформулювання проблеми SFC, що зазвичай призводить до низького використання та високих витрат. Щоб вирішити цю проблему, [47] створив гібридну систему на основі DRL, яка розділяє розгортання VNF і маршрутизацію потоку на окремі модулі. Виключна відповідальність агента DRL полягає в тому, щоб вивчити політику розгортання VNF. Адаптивний параметр шуму, політика Wolpertinger і відтворення пріоритетного досвіду використовуються для налаштування структури агента на основі DDPG і підвищення ефективності навчання. Ігровий модуль (GBM) використовується для маршрутизації потоку.

Алгоритм децентралізованої маршрутизації для GBM розроблено для вирішення проблеми масштабованості. Агент DRL змінив політику розгортання на основі винагороди, згенерованої GBM під час процесу навчання. Запропонований метод перевершив традиційні алгоритми на основі DRL з точки зору ефективності навчання з двох причин. По-перше, запропонована методика різко скоротила простір дії агента DRL. По-друге,

традиційні алгоритми на основі DRL майже повністю не залежать від моделі. GBM, з іншого боку, використовував алгоритми на основі моделі для оптимізації маршрутизації потоку в запропонованому методі. Інформація на основі моделі, як-от градієнт, призвела до значного підвищення ефективності алгоритму.

2.5.4 Гібридне управління хмарними системами

Більшість рішень для керування хмарою використовують один метод вирішення проблем у своєму робочому процесі. Але деякі з них потрібні для досягнення кількох цілей одночасно, наприклад, прогнозування вхідного робочого навантаження та розподілення його між ресурсами або кластеризація набору ресурсів на основі поточного стану виконання з подальшим їх динамічним наданням. Ці робочі процеси можуть отримати користь від окремих методів машинного навчання для кожної цілі. Гібридні методи ML були розроблені, щоб подолати недоліки автономних методів шляхом поєднання одного або кількох методів ML разом. Кілька робочих процесів, які ми розглядаємо в цьому дослідженні, необхідні для прогнозування деяких параметрів ресурсів або робочого навантаження перед початком процесу динамічної оптимізації. Відповідно, гібридні методи машинного навчання, які ми тут обговорюємо, поєднують прогнозування параметрів і хмарну оптимізацію.

Розробити ефективні алгоритми масштабування складно, особливо для георозподілених ланцюгів VNF, де витрати на пропускну здатність трафіку WAN і затримки є важливими, але їх важко врахувати під час прийняття рішень щодо масштабування. Щоб вирішити цю problem, [48] запропонував методологію, засновану на глибокому навчанні, аналізуючи внутрішні закономірності нестабільності трафіку та ефективні методи розгортання з часом, з метою покращення суджень завдяки поглибленому навчанню на досвіді. LSTM використовується для прогнозування майбутніх витрат, а

агент DRL використовується для прийняття динамічних рішень щодо розміщення ланцюга VNF. Щоб покращити результати, у цьому дослідженні використовувалася техніка відтворення досвіду, заснована на алгоритмі актор-критик DRL. Порівняно з попередніми репрезентативними алгоритмами, симуляція на основі трасування показала, що нещодавно розроблена платформа навчання швидко реагувала на динаміку онлайн-трафіку та досягла нижчих витрат на систему з мінімальним навчанням поза мережею.

Раніше розробникам веб-додатків доводилося зберігати більше копій об’єктів даних у багатьох георозподілених центрах обробки даних або надсилати повторювані запити до кількох (наприклад, найближчих) центрів обробки даних, щоб забезпечити низьку затримку запитів для користувачів, що збільшувало грошові витрати. Щоб вирішити цю проблему, в [49] запропоновано георозподілену хмарну систему зберігання на основі RL під назвою GeoCol з метою досягнення низької вартості та затримки. По-перше, ця нова система включала механізм розподілу запитів і метод планування зберігання для досягнення найкращого компромісу між грошовою вартістю та затримкою запиту. Щоб увімкнути паралельні передачі для об’єкта даних, у підході розділеного запиту використовувався метод SARIMA, щоб передбачити затримку запиту як вхідні дані для моделі RL для розрахунку кількості підзапитів і центру обробки даних, призначеного для кожного підзапиту. По-друге, GeoCol застосував інший метод RL, щоб вирішити, чи потрібно зберігати кожен об’єкт даних і тип зберігання кожного збереженого об’єкта даних, щоб зменшити вартість зберігання.

У [50] автори вивчали способи підключення різних генераторів відновлюваної енергії до георозподілених центрів обробки даних від різних хмарних провайдерів, щоб зменшити викиди вуглецю, грошові витрати та порушення цільового рівня обслуговування (SLO), спричинені дефіцитом відновлюваної енергії. Автори протестували кілька підходів ML для довгострокової точності прогнозу виробництва та попиту на відновлювану

енергію та обрали для прогнозу SARIMA. Ґрунтуючись на прогнозованих результатах, у дослідженні було представлено багатоагентний підхід RL (MARL) для кожного центру обробки даних, щоб визначити, скільки відновлюваної енергії запитувати від кожного генератора. У цьому дослідженні також було представлено підхід гарантованого терміну відстрочки роботи (DGJP) для відстрочки виконання несуттєвих робіт, коли запасів відновлюваної енергії недостатньо.

3 РЕЗУЛЬТАТИ АНАЛІЗУ МЕТОДІВ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ У ХМАРНОМУ СЕРЕДОВИЩІ

3.1 Класифікація методів управління з використанням машинного навчання

Щоб вирішити проблему надійного керування хмарою для територіально розподілених центрів обробки даних, необхідно вирішити кілька підпроблем. У попередньому розділі ми розглянули різні методи та стратегії. Більшість використовує один, але деякі з них використовують кілька методів ML у своєму операційному процесі. Відповідно до інформації, представленої в попередніх розділах, підходи ML містять будь-які моделі, які можна навчити за допомогою даних для виконання контрольованих, неконтрольованих завдань або завдань на основі RL. У таблиці 3.1 наведена класифікація моделей ML, які використовуються в усіх джерелах, які обговорюються в цьому дослідженні. Ми виділили чотири різні типи методів. На рисунку 4.1 показано відсоток типів моделей машинного навчання, розглянутих різними дослідниками.

Таблиця 3.1 – Методи машинного навчання, найбільш поширені при управлінні хмарними ресурсами

Тип машинного навчання	Алгоритми машинного навчання
Кластерізація	k-середніх, Spectral
Регресія	LR, Poly LR, LASSO, DT, RF, GBRT, XGBoost, k-NN, SVM, SARIMA
Нейронні мережі	SCN, MLP, CNN, LSTM, GRU
Навчання з підкріпленням	На основі моделі, актор-критик, Q-навчання, DQN, DDPG, DRL, MARL

Також надано порівняльний підсумок досліджених джерел на основі різних підходів ML, які використовуються в георозподіленому хмарному середовищі керування. Таблиця також демонструє, чому різні моделі ML використовуються в різних наукових статтях.

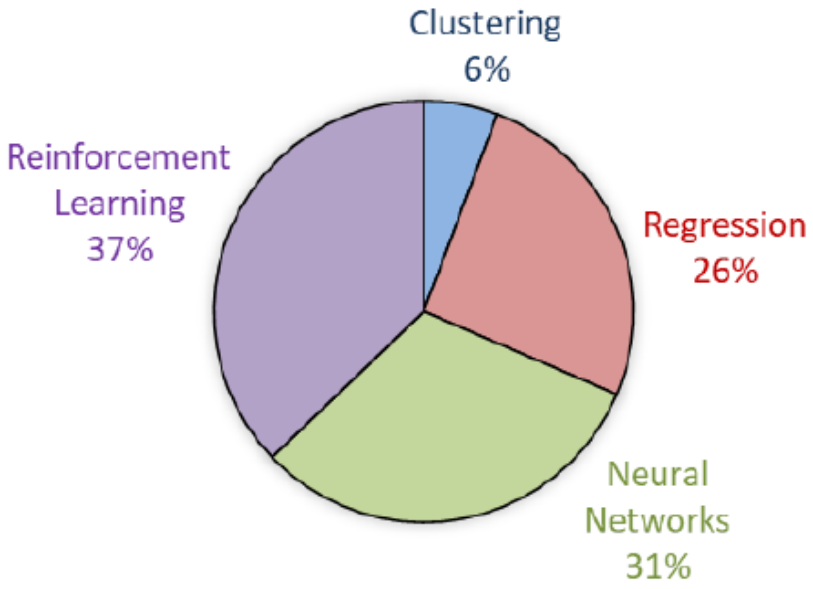

Рисунок 4.1 – Відсоток типів моделей машинного навчання

На рисунку 4.1, дивлячись на обмежене використання методів навчання на основі кластеризації (k-середні та спектр), ми можемо сказати, що вони найменш придатні для керування хмарою, оскільки вони групують робочі навантаження або ресурси в кластери, але в масштабі та деталізації, які не є особливо корисними для структур керування хмарою. Єдиний випадок, коли неконтрольована кластеризація може бути корисною, це коли робоче навантаження або ресурси змінюються дуже часто.

Згідно з висновками дослідження, прогнозування параметрів робочого навантаження та ресурсів є критичним завданням у великомасштабних георозподілених інфраструктурах хмарних обчислень. Високоточні прогнози дозволяють надійно керувати вхідним робочим навантаженням і доступними ресурсами, забезпечуючи QoS і SLA, одночасно знижуючи експлуатаційні витрати на хмару. Проте прогнозування робочого навантаження та

параметрів ресурсів загалом є складною проблемою, особливо коли вони залежать від непередбачуваних дій (наприклад, веб-серверів, пристроїв Інтернету речей, біологічних датчиків, робочих навантажень смартфонів/ПК, збоїв ресурсів). Для таких прогнозів часто використовуються регресивні моделі, такі як LR, Lasso або SARIMA. Нове покоління моделей аналізу даних на основі NN, з іншого боку, щойно було прийнято, і широкі оцінки їх показали обнадійливі результати. В середньому моделі на основі RNN, наприклад, LSTM, здається, дають кращі прогнози швидше порівняно з традиційними моделями на основі регресії [32]. Залежність часу виконання завдань в реальному часі роботи фреймворка показано на рисунку 4.2.

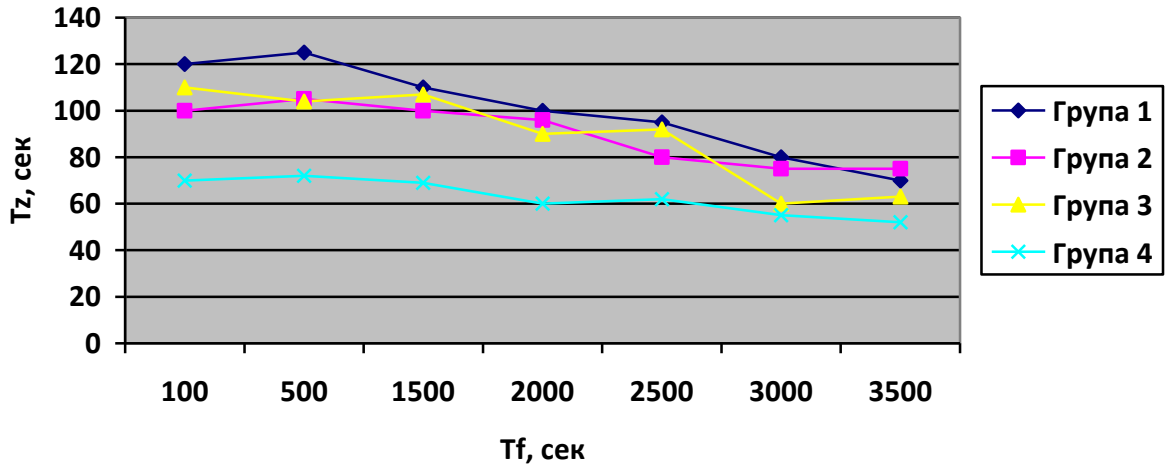

Рисунок 4.2 – Час виконання (Tz, сек) завдань в реальному часі роботи фреймворка (Tf, сек)

На практиці різні складні характеристики георозподілених хмарних систем, наприклад, великий масштаб і неоднорідність систем, випадковість параметрів надходження робочого навантаження та доступність ресурсів, а також своєчасність рішень щодо планування, дуже ускладнюють розробку точних моделей, що вимагає використання високошвидкісного алгоритму планування. Окремому евристичному або метаевристичному алгоритму важко адекватно адаптуватися до реальної динаміки георозподілених хмарних обчислювальних систем. У таких сценаріях глибоке використання

алгоритмів RL у розглянутих роботах показує, що це може бути дуже перспективним для управління хмарними обчисленнями. У сфері оптимізації на основі машинного навчання RL є фактично найбільш поширеним досліджуваним методом.

3.2 Напрямки майбутніх досліджень

Після аналізу поточного стану георозподіленого хмарного управління, тепер проаналізуємо кілька дослідницьких проблем і можливостей у цьому розділі, які висвітлені у поточній літературі, і, отже, представляють життєздатні майбутні напрямки досліджень у сфері ML для управління георозподіленим хмарним центром обробки даних.

Розширені алгоритми ML. Сучасні дослідження використовують традиційні алгоритми машинного навчання, такі як регресія, k-середнє кластеризація, Q-learning тощо. Більш просунуті методи NN і Deep Learning (DL), які, як було показано, успішно та точніше вирішують масштабні проблеми в інших доменах додатків, можуть бути адаптовані для георозподіленого хмарного домену управління. Наприклад, LSTM можуть передбачати майбутні значення на основі попередніх послідовних даних. Їх можна використовувати там, де потрібні легкі та швидкі моделі прогнозування часових рядів. Якщо прийнятно трохи більше обчислювальних витрат, замість LSTM можна використовувати моделі Transformer ML. Вони можуть обробляти набагато більші обсяги даних за той самий проміжок часу завдяки паралелізму механізму планування. DL особливо корисний при роботі з проблемами, які мають простір станів великої розмірності. У DRL, завдяки здатності DL вивчати різні рівні абстракції даних, RL може виконувати все більш складні завдання з меншою кількістю попередньої інформації. Нові методи RL-м’якої акторської критики (SAC) можуть оптимізувати стохастичну політику поза політикою, подолаючи розрив між стохастичною оптимізацією політики та методами у

стилі DDPG. Він використовує регулярізацію ентропії, за якої політика навчена максимізувати компроміс між очікуваною віддачею та ентропією (випадковість у політиці). Завдяки регулярізації ентропійного фактора SAC може бути високоефективним для підходів оптимізації на основі енергії.

Ефективне навчання та розгортання ML. Одним із компонентів розробки паралельних алгоритмів ML є використання кількох потоків/ процесів для прискорення швидкості навчання моделі ML. У деяких випадках мережі вимагають великої кількості навчальних даних, значного часу обробки та навіть спеціального обладнання для запуску паралельних алгоритмів машинного навчання, наприклад серверів із графічним процесором. Замість використання таких важких обчислювальних алгоритмів ML (наприклад, NN, Deep Learning, RL тощо), дослідники повинні запропонувати швидкі та легкі рішення ML, які можна розгорнути на стандартному обладнанні. Методи стиснення моделі (відсікання, квантування, факторизація низького рангу, дистиляція знань тощо) є особливо перспективними для зменшення накладних витрат на логічний висновок моделі, особливо якщо моделі ML виконуються часто. Стиснення моделі – це стратегія розгортання найсучасніших моделей ML із меншим використанням пам’яті та скороченою затримкою без шкоди для точності. Багато досліджень управління хмарою розробили ефективні алгоритми профілювання, прогнозування або оптимізації на основі ML. Але більшість цих методів машинного навчання не призначені для роботи в Інтернеті (для висновків) і прийняття рішень у реальному часі після обробки великих обсягів даних у реальному часі. Ці існуючі методи можна перетворити на онлайнові (наприклад, RL або DRL на основі алгоритмів) для роботи з методами профілювання та прогнозування, зі стисненням моделі для підвищення ефективності. Іншим елементом, який слід дослідити, є використання ансамблевого навчання та RL разом для швидкого розгортання.

Контрольні показники/робочі навантаження. Більшість запропонованих моделей машинного навчання не було навчено та оцінено з використанням

великих високоякісних наборів даних, отриманих сильними промисловими хмарними операторами у виробничих сценаріях. У більшості попередніх робіт використовуються синтетичні невеликі набори даних або набори даних, які не відображають георозподілені події реального світу. Через дефіцит поточних і релевантних даних (щодо робочих навантажень, використання ресурсів і тенденцій) із георозподілених хмарних середовищ важко точно оцінити якість опублікованих результатів і, що більш важливо, порівняти результати конкуруючих досліджень. Широкомасштабний, реалістичний, загальнодоступний георозподілений набір даних хмарного центру обробки даних від великого постачальника хмарних технологій слугуватиме стандартом, що дозволить дослідникам аналізувати та порівнювати альтернативні ідеї та підходи в набагато більшому масштабі.

Сучасні розподілені обчислювальні системи. Логічним кроком було б вивчити дизайн і побудову розподіленої версії запропонованих алгоритмів ML і створити повну систему управління з використанням передових технологій великих даних, наприклад Apache Spark, які можна розгорнути в георозподілених хмарних центрах обробки даних.

Залучення промисловості для валідації. Було б ідеально залучити більше промислових учасників до досліджень, не лише надаючи відповідні вимоги, але й сприяючи оцінці результатів, і особливо розгортанню невеликих пілотних проектів у виробничих умовах. Незважаючи на те, що невеликі випробувальні стенди, які використовуються в деяких дослідженнях, дозволяють тестувати багато ідей і конфігурацій, неможливо зробити остаточні або повні висновки без ретельного дослідження на виробництві.

Різнорідні ресурси/обладнання. Більшість досліджень розглядають традиційні апаратні ресурси, наприклад серверні вузли, процесори тощо. Запропоновані підходи в цих дослідженнях можна розширити, щоб розглянути більш складні запити користувачів із кількома типами ресурсів, крім центральних процесорів, наприклад, дезагреговані пам’яті, графічні

процесори, прискорювачі, мережеві комутатори , нові системи охолодження та врахування неоднорідності між ними.

Мережі. Внутрішні та внутрішньоцентрові мережі обробки даних відіграють дуже важливу роль під час управління хмарою (зниження операційних витрат, QoS, SLA тощо). Зі постійно зростаючими обсягами даних у нових хмарних робочих навантаженнях дуже важливо враховувати внутрішні та внутрішні мережі центрів обробки даних, а також мережеві витрати, пов’язані з передачею даних між георозподіленими центрами обробки даних [25]. Дослідження повинні включати параметри, пов’язані з мережею, наприклад функцію оптимізації, щоб нагадувати найсучасніші системи керування хмарою.

Енергетичне моделювання. Більшість, якщо не всі дослідження, включені в аналіз, розглядають прості моделі енергопостачання центрів обробки даних, які базуються на використанні процесора/вузла. У енергетичній моделі центру обробки даних для підвищення точності дослідники можуть враховувати різні фактори, такі як різні стани продуктивності (P-стани) ядер, теплова потужність охолодження, чисте вимірювання та пікове зниження, тобто зниження пікової потужності тощо [24].

Багатоцільові алгоритми. На відміну від багатьох існуючих досліджень, реальні системи управління хмарою повинні вирішувати більше ніж одну або дві цілі. Цілі майбутніх досліджень можна змінити, щоб включити більше показників оптимізації. Наприклад, обчислювальна продуктивність центру обробки даних/мережі (затримка, SLA, QoS тощо) і операційні витрати центру обробки даних/мережі можуть бути оптимізовані спільно; можна розглянути споживання зеленої енергії центром обробки даних, виробництво відновлюваної енергії та викиди вуглецю. Продуктивність центру обробки, даних час безвідмовної роботи/доступність і відмовостійкість також можна оптимізувати разом.

ВИСНОВКИ

В кваліфікаційній роботі розглянуті проблеми надійного георозподіленого керування хмарою, які вирішувалися за допомогою методів машинного навчання. Для спрощення ця проблема була розділена на три підкатегорії: профілювання, прогнозування параметрів і хмарна оптимізація. Крім того, було досліджено методи машинного навчання, які використовувалися для підвищення надійності алгоритмів керування хмарою в різноманітних і неоднорідних середовищах.

Останніми роками кількість досліджень, які використовують підходи на основі машинного навчання, різко зросла. Декілька дослідників використовували різноманітні методології, починаючи від традиційної статистики та регресії і закінчуючи просунутими алгоритмами машинного навчання.

В роботі зроблено висновок, що в середньому методи ML перевершують традиційні математичні методи, особливо при роботі з великими та складними середовищами. Розвиток методології машинного навчання в поточних дослідженнях проілюстровано в роботі, яка допоможе дослідникам зрозуміти прогалини в дослідженнях у цій темі. Нарешті, на основі викликів, виявлених у дослідженні, визначено кілька перспективних майбутніх напрямків досліджень і можливостей для посилення поточних підходів ML.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Digital Around the World, [Online]. Available: https://datareportal.com/global-digital-overview. [Accessed 20 Dec 2021].

2. "Global Cloud Computing Market Outlook, [Online]. Available: https://www.businesswire.com/news/home/20211112005486/en/Global-947.3B-Cloud-Computing-Market-Outlook-2026-by-Service-Model-Deployment-Model-Organization-Size-Vertical-and-Region---ResearchAndMarkets.com. [Accessed 20 Dec 2021].

3. Gartner Global Cloud Revenue, [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences. [Accessed 20 Dec 2021].

4. Google Cloud Locations, [Online]. Available: https://cloud.google.com/about/locations. [Accessed 20 Dec 2021].

5. AWS Global Infrastructure, [Online]. Available: http://aws.amazon.com/about-aws/global-infrastructure/. [Accessed 20 Dec 2021].

6. Microsoft Teams Up With Accenture, Goldman on Greener Software, [Online]. Available: https://www.bloombergquint.com/business/microsoft-teams-up-with-accenture-goldman-on-greener-software. [Accessed 20 Dec 2021].

7. L. Belkhir and A. Elmeligi, Assessing ICT global emissions footprint: Trends to 2040 & recommendations. Journal of Cleaner Production, vol. 177, p. 448–463, 2018.

8. Greenpeace: China’s Data Centers on Track to Use More Energy than All of Australia, [Online]. Available: https://www.datacenterknowledge.com/asia-pacific/greenpeace-china-s-data-centers-track-use-more-energy-all-australia. [Accessed 20 Dec 2021].

9. Amazon Found Every 100ms of Latency Cost them 1% in Sales, [Online]. Available: https://www.gigaspaces.com/blog/amazon-found-every-

100ms-of-latency-cost-them-1-in-sales/.

10. Study Shines a Light on Big Tech's AI Investments, [Online]. Available: https://www.nextgov.com/emerging-tech/2021/04/study-shines-light-big-techs-ai-investments/173561/. [Accessed 20 Dec 2021].

11. W. Khallouli and J. Huang. Cluster resource scheduling in cloud computing: literature review and research challenges. The Journal of Supercomputing, p. 1–46, 2021.

12. M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz and U. Farooq, Resource Allocation Techniques in Cloud Computing: A Review and Future Directions. Elektronika ir Elektrotechnika, vol. 26, p. 40–51, 2020.

13. B. K. Dewangan, A. Agarwal, T. Choudhury, A. Pasricha and S. Chandra Satapathy. Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges. Software: Practice and Experience, vol. 51, p. 2373–2392, 2021.

14. S. Jayaprakash, M. D. Nagarajan, R. P. d. Prado, S. Subramanian and P. B. Divakarachari. A systematic review of energy management strategies for resource allocation in the cloud: Clustering, optimization and machine learning. Energies, vol. 14, p. 5322, 2021.

15. H. Kaur and K. Kaur. Load Balancing and Its Challenges in Cloud Computing: A Review. Information and Communication Technology for Competitive Strategies (ICTCS 2020), p. 731–741, 2021.

16. D. A. Shafiq, N. Z. Jhanjhi and A. Abdullah. Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences, 2021.

17. A. A. A. Alkhatib, A. Alsabbagh, R. Maraqa and S. Alzubi. Load Balancing Techniques in Cloud Computing: Extensive Review. Advances in Science, Technology and Engineering Systems Journal, vol. 6, p. 860–870, 2021.

18. S. S. George and R. S. Pramila. A review of different techniques in cloud computing. Materials Today: Proceedings, vol. 46, pp. 8002-8008, 2021.

19. G. Zhou, W. Tian and R. Buyya. Deep Reinforcement Learning-based

Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions. arXiv preprint arXiv:2105.04086, 2021.

20. T. L. Duc, R. G. Leiva, P. Casari and P.-O. Östberg. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Computing Surveys (CSUR), vol. 52, p. 1–39, 2019.

21. V. N. Tsakalidou, P. Mitsou and G. A. Papakostas. Machine learning for cloud resources management–An overview. arXiv preprint arXiv:2101.11984, 2021.

22. S. Goodarzy, M. Nazari, R. Han, E. Keller and E. Rozner. Resource Management in Cloud Computing Using Machine Learning: A Survey. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020.

23. T. Khan, W. Tian and R. Buyya. Machine Learning (ML)-Centric Resource Management in Cloud Computing: A Review and Future Directions. arXiv preprint arXiv:2105.05079, 2021.

24. N. S. Hogade, S. Pasricha and H. J. Siegel. Energy and Network Aware Workload Management for Geographically Distributed Data Centers. IEEE Transactions on Sustainable Computing, 2021.

25. N. Hogade, S. Pasricha, H. J. Siegel, A. A. Maciejewski, M. A. Oxley and E. Jonardi. Minimizing Energy Costs for Geographically Distributed Heterogeneous Data Centers. IEEE Transactions on Sustainable Computing, vol. 3, pp. 318-331, 2018.

26. H. Ziafat and S. M. Babamir. A method for the optimum selection of datacenters in geographically distributed clouds. The Journal of Supercomputing, vol. 73, p. 4042–4081, 2017.

27. K. a. P. A. a. A. S. a. M. T. Deb. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.

28. A. Atrey, G. Van Seghbroeck, B. Volckaert and F. De Turck. Scalable data placement of data-intensive services in geo-distributed clouds. in

CLOSER2018, the 8th International Conference on Cloud Computing and Services Science, 2018.

29. E. a. M. S. A. a. L. J. Cho. Friendship and Mobility: User Movement in Location-Based Social Networks. in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011.

30. S. Taheri, M. Goudarzi and O. Yoshie. Providing RS Participation for Geo-Distributed Data Centers Using Deep Learning-Based Power Prediction. in International Congress on High-Performance Computing and Big Data Analysis, 2019.

31. S. Taheri, M. Goudarzi and O. Yoshie. Learning-based power prediction for geo-distributed Data Centers: weather parameter analysis. Journal of Big Data, vol. 7, p. 1–16, 2020.

32. E. Baccour, F. Haouari, A. Erbad, A. Mohamed, K. Bilal, M. Guizani and M. Hamdi. An Intelligent Resource Reservation for Crowdsourced Live Video Streaming Applications in Geo-Distributed Cloud Environment. IEEE Systems Journal, 2021.

33. H. Wang, D. Niu and B. Li. Dynamic and decentralized global analytics via machine learning. in Proceedings of the ACM Symposium on Cloud Computing, 2018.

34. TPC-H Benchmark Specification. Transaction Processing Performance Council, 2017. [Online]. Available: http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-h_v2.17.2.pdf. [Accessed 1 March 2022].

35. J. Bi, H. Yuan, L. Zhang and J. Zhang. SGW-SCN: An integrated machine learning approach for workload forecasting in geo-distributed cloud data centers. Information Sciences, vol. 481, p. 57–68, 2019.

36. P. A. Lima, A. S. B. Neto, P. Maciel and others. Data centers' services restoration based on the decision-making of distributed agents. Telecommunication Systems: Modelling, Analysis, Design and Management, vol. 74, p. 367–378, 2020.

37. S. Rawas, A. S. Zekri and A. El Zaart, Power and Cost-aware Virtual Machine Placement in Geo-distributed Data Centers. In CLOSER, 2018.

38. X. Zhou, K. Wang, W. Jia and M. Guo. Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers. In 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), 2017.

39. Q. Li, Z. Peng, D. Cui, J. He, K. Chen and J. Zhou. Data Center Selection Based on Reinforcement Learning. In 2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT), 2019.

40. D. Cheng, X. Zhou, Z. Ding, Y. Wang and M. Ji. Heterogeneity aware workload management in distributed sustainable datacenters. IEEE Transactions on Parallel and Distributed Systems, vol. 30, p. 375–387, 2018.

41. E. Baccour, A. Erbad, A. Mohamed, F. Haouari, M. Guizani and M. Hamdi. RL-OPRA: Reinforcement Learning for Online and Proactive Resource Allocation of crowdsourced live videos. Future Generation Computer Systems, vol. 112, p. 982–995, 2020.

42. C. Xu, K. Wang and M. Guo. Intelligent resource management in blockchain-based cloud datacenters. IEEE Cloud Computing, vol. 4, p. 50–59, 2017.

43. T. Wei, S. Ren and Q. Zhu. Deep reinforcement learning for joint datacenter and hvac load control in distributed mixed-use buildings. IEEE Transactions on Sustainable Computing, 2019.

44. C. Xu, K. Wang, P. Li, R. Xia, S. Guo and M. Guo. Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning. IEEE Transactions on Network Science and Engineering, vol. 7, p. 205–215, 2018.

45. F. Rossi, V. Cardellini, F. L. Presti and M. Nardelli. Geo-distributed efficient deployment of containers with Kubernetes. Computer Communications, vol. 159, p. 161–174, 2020.

46. Y. Qin, W. Han, Y. Yang and W. Yang. Joint energy optimization on

the server and network sides for geo-distributed data centers. The Journal of Supercomputing, vol. 77, p. 7757–7790, 2021.

47. T. Tang, B. Wu and G. Hu. A Hybrid Learning Framework for Service Function Chaining Across Geo-Distributed Data Centers. IEEE Access, vol. 8, p. 170225–170236, 2020.

48. Z. Luo, C. Wu, Z. Li and W. Zhou. Scaling geo-distributed network function chains: A prediction and learning framework. IEEE Journal on Selected Areas in Communications, vol. 37, p. 1838–1850, 2019.

49. H. Wang, H. Shen, Z. Li and S. Tian. GeoCol: A Geo-distributed Cloud Storage System with Low Cost and Latency using Reinforcement Learning. In 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021.

50. H. Wang, H. Shen, J. Gao, K. Zheng and X. Li. Multi-Agent Reinforcement Learning based Distributed Renewable Energy Matching for Datacenters. In 50th International Conference on Parallel Processing, 2021.

51. B. Yu and J. Pan. A framework of hypergraph-based data placement among geo-distributed datacenters. IEEE Transactions on Services Computing, vol. 13, p. 395–409, 2017.

52. F. Haouari, E. Baccour, A. Erbad, A. Mohamed and M. Guizani. Transcoding resources forecasting and reservation for crowdsourced live streaming. In 2019 IEEE Global Communications Conference (GLOBECOM), 2019.

53. М.О. Волк, В.С Курочкін, А.П. Запорожченко, П.А. Паронікян. Гібридний метод розподілу ресурсів в хмарних системах. Системи управління, навігації та зв'язку. № 2 (76). 2024. С. 70-73. Фахове видання.