

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО"
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО"
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Кваліфікаційна наукова
праця на правах рукопису

КАРТАШОВ АНТОН ДИМИТРОВИЧ

УДК 004.75

ДИСЕРТАЦІЯ

**ПІДХІД ЩОДО ОРГАНІЗАЦІЇ РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА
ДОСТУПУ ДО CLOUD ДАНИХ**

Спеціальність 172 – Телекомунікації та радіотехніка

Галузь знань 17 – Електроніка та телекомунікації

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Карташов А.Д.

Науковий керівник: Глоба Л.С. доктор технічних наук, професор

Київ – 2025

АНОТАЦІЯ

Карташов А.Д. Підхід щодо організації розподіленого зберігання та доступу до cloud даних. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 172 – Телекомунікації та радіотехніка. – Навчально-науковий інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Київ, 2025.

Стрімке зростання обсягів цифрових даних та поширення хмарних сервісів суттєво змінили підходи до зберігання інформації. Сьогодні організації активно впроваджують мультихмарні стратегії, щоб зменшити залежність від одного постачальника, підвищити надійність зберігання даних та оптимізувати витрати. Проте такий розподілений підхід створює нові виклики: складність розміщення даних, нестабільну продуктивність різних провайдерів, труднощі з дотриманням нормативних вимог та зростаючу операційну складність. Незважаючи на наявні рішення, досі бракує комплексних підходів, які б одночасно враховували технічні, економічні, регуляторні та екологічні аспекти управління даними в мультихмарному середовищі, особливо коли умови зберігання постійно змінюються.

Дана робота вирішує актуальну науково-практичну задачу розробки вдосконаленого підходу щодо підвищення ефективності розподіленого зберігання та доступу до даних у динамічно змінюваному мультихмарному середовищі. Дослідження представляє фреймворк Інтелектуальний Оркестратор Хмарних Даних – Intelligent Cloud Data Orchestrator (ICDO), який використовує методи штучного інтелекту, багатоцільової оптимізації та хмарних технологій для забезпечення адаптивного розподілу та надійного зберігання даних у мультихмарному середовищі, гарантуючи їх цілісність.

Метою дисертаційної роботи є підвищення ефективності розподілу та зберігання даних в мультихмарному середовищі за рахунок оцінювання показників доступності, затримки, надійності, пропускну здатності, вартості та ін., та завдяки утворенню віртуальної логічно-єдиної системи зберігання даних. Для досягнення мети дослідження було поставлено та вирішено задачі визначення основних

характеристик, властивостей, показників та критеріїв оцінювання ефективності розподіленого зберігання та доступу до даних у мультимарних середовищах; систематизації та формального опису складних взаємозв'язків між технічними, економічними, регуляторними та екологічними аспектами управління даними в мультимарному середовищі; розробки взаємодоповнюючих онтологічної та математичної моделей; запропоновано метод, який ефективно інтегрує навчання з підкріпленням та багатоцільові еволюційні алгоритми для ефективного визначення Парето-оптимальних рішень; спроектовано та реалізовано фреймворк, який інтегрує методи штучного інтелекту з технологіями багатокритеріальної оптимізації; експериментально оцінено ефективність запропонованого підходу за ключовими показниками продуктивності.

Детальний аналіз існуючих рішень та їхніх обмежень дозволив визначити актуальність застосування комплексного підходу щодо оптимізації визначених кінцевим користувачем критеріїв зберігання даних у мультимарному середовищі. Для формалізації цієї складної області розроблено взаємодоповнюючі онтологічну та математичну моделі, які відображають складні взаємозв'язки між компонентами мультимарного середовища та представляють багатоцільову природу проблеми оптимізації. Онтологічна модель формалізує складні взаємозв'язки між компонентами, тоді як математична модель представляє багатоцільову природу задачі оптимізації. Основними завданнями дослідження було створення теоретичного підґрунтя для розробки інтелектуальної системи управління даними в мультимарному середовищі, яка здатна адаптуватися до динамічних змін умов зберігання та доступу до даних. Важливим завданням було також розробити механізми, що дозволяють системі автоматично реагувати на зміни в шаблонах доступу до даних, коливання продуктивності хмарних провайдерів, зміни в тарифних планах та регуляторних вимогах.

Наукова новизна дисертаційної роботи полягає у вирішенні важливого наукового завдання розробки підходу щодо розподіленого зберігання та доступу до даних з динамічно-змінюваними умовами зберігання у мультимарному середовищі та реалізації практичних методів на його основі. Запропоновано метод

інтеграції алгоритмів навчання з підкріплення та багатоцільових еволюційних алгоритмів для оптимізації процесу розподіленого зберігання даних, який на відміну від існуючих, автоматизує й оптимізує процес вибору провайдера та підвищує ефективність прийняття рішень у мультихмарному середовищі з динамічно-змінюваними умовами зберігання даних. Запропоновано онтологічну модель розподілу і зберігання даних, що формалізує та систематизує складні взаємозв'язки між показниками ефективності та змінними факторами, які впливають на процес розподілу і зберігання даних в залежності від пріоритетів користувачів та статистичних показників ефективності надання послуг хмарними провайдерами. Удосконалено математичну модель системи з динамічно-змінюваними умовами зберігання даних в мультихмарних середовищах шляхом інтеграції статичних та динамічних критеріїв, яка надає можливість автоматизованого врахування змін параметрів середовища в режимі реального часу. Запропонований підхід щодо зберігання даних у мультихмарному середовищі вирізняється інтеграцією методів штучного інтелекту з методами багатоцільової оптимізації.

В результаті виконання дослідження теоретичні розробки дисертації представлені автором у вигляді комплексного фреймворку Інтелектуальний Оркестратор Хмарних Даних – Intelligent Cloud Data Orchestrator (ICDO), який дозволяє підвищити ефективність зберігання та доступу до даних з динамічно-змінюваними умовами їх зберігання у мультихмарних середовищах. Підхід ICDO досяг 44,1% покращення загальної продуктивності порівняно з традиційними підходами з однією хмарою та 14,9% покращення порівняно з найкращим альтернативним мультихмарним підходом. На основі розроблених онтологічних та математичних моделей створено програмне забезпечення для оптимізації розміщення та управління даними, яке дозволяє підвищити продуктивність, надійність та безпеку зберігання даних з одночасним зниженням витрат та дотриманням регуляторних вимог. Виконано практичне впровадження розробленого програмного забезпечення для управління даними з динамічно-змінюваними умовами їх зберігання в мультихмарних середовищах у клієнтських

проектах компанії Tech-5 UG, що спеціалізуються на обробці великих обсягів даних та потребують гнучких рішень для їх зберігання. За підсумками впровадження було зафіксовано зниження середніх витрат на хмарну інфраструктуру на 9%, а також скорочення затримки при доступі до даних на 25% у порівнянні з попередніми підходами. Розроблені підходи впроваджено у навчальні та дослідницькі програми кафедри ІТТ НТУУ «КПІ» імені Ігоря Сікорського для підготовки фахівців у сфері хмарних обчислень.

Проведені експериментальні дослідження підтверджують, що підхід ICDO забезпечує суттєві покращення за всіма ключовими показниками ефективності, включаючи продуктивність, вартість, надійність, безпеку та адаптивність, порівняно з існуючими рішеннями. Експериментальне дослідження демонструє, що адаптивність є критичним фактором для ефективного управління даними в динамічних мультихмарних середовищах, а механізми адаптації та навчання, впроваджені в підхід ICDO, дозволяють автоматизувати процес реагування на зміни в шаблонах доступу до даних та умовах середовища. Крім того, дослідження підтверджує ефективність інтегрованого підходу знаходження Парето-оптимальних рішень у порівнянні з оптимізацією за окремими критеріями. Дослідження завершується визначенням перспективних напрямків для подальшої роботи. Ці результати суттєво сприяють розвитку галузі розподіленого управління даними та пропонують практичні рішення для нагальних викликів оптимізації мультихмарного зберігання даних.

Ключові слова: мультихмарне середовище, управління хмарами, розподіл ресурсів, управління ресурсами, якість обслуговування, угоди про рівень обслуговування, мульти-критеріальна оптимізація, онтологічне моделювання.

ABSTRACT

A. Kartashov Approach to organizing distributed storage and access to cloud data. – Qualifying scientific work on manuscript rights.

Thesis for graduation scientific degree of Philosophy Doctor by specialty 172 – Telecommunications and radio engineering. – Educational and Scientific Institute of Telecommunication Systems of KPI named after Igor Sikorsky, Kyiv, 2025.

The rapid growth of digital data volumes and the rise of cloud services have significantly changed approaches to data storage. Today, organizations are actively implementing multi-cloud strategies to reduce dependence on a single provider, improve data storage reliability, and optimize costs. However, this distributed approach creates new challenges: complexity of data placement, unstable performance of different providers, difficulties in complying with regulatory requirements, and increasing operational complexity. Despite a number of existing solutions, there is still a lack of comprehensive approaches that simultaneously consider technical, economic, regulatory, and environmental aspects of data management in a multi-cloud environment, especially when storage conditions are constantly changing.

This work addresses the relevant scientific and practical task of developing an improved approach to enhance the efficiency of distributed storage and access to data in a dynamically changing multi-cloud environment. The research presents the Intelligent Cloud Data Orchestrator (ICDO) framework, which uses artificial intelligence methods, multi-objective optimization, and cloud technologies to ensure adaptive distribution and reliable data storage in a multi-cloud environment, guaranteeing their integrity.

The aim of the research is to improve the efficiency of data distribution and storage in a multi-cloud environment by evaluating indicators of availability, latency, reliability, throughput, cost, etc., and by creating a virtual logically unified data storage system. To achieve the research goal, tasks were set and solved to determine the main characteristics, properties, indicators, and criteria for evaluating the effectiveness of distributed storage and access to data in multi-cloud environments; systematization and formal description of complex relationships between technical, economic, regulatory, and environmental aspects of data management in a multi-cloud environment; development of

complementary ontological and mathematical models; a method was proposed that effectively integrates reinforcement learning and multi-objective evolutionary algorithms for efficient determination of Pareto-optimal solutions; a framework was designed and implemented that integrates artificial intelligence methods with multi-criteria optimization technologies; the effectiveness of the proposed approach was experimentally evaluated according to key performance indicators.

A detailed analysis of existing solutions and their limitations allowed determining the relevance of applying a comprehensive approach to optimizing user-defined criteria for data storage in a multi-cloud environment. To formalize this complex area, complementary ontological and mathematical models were developed that reflect complex relationships between components of the multi-cloud environment and represent the multi-objective nature of the optimization problem. The ontological model formalizes complex relationships between components, while the mathematical model represents the multi-objective nature of the optimization task. The main objectives of the research were to create a theoretical foundation for developing an intelligent data management system in a multi-cloud environment that can adapt to dynamic changes in storage conditions and data access. An important task was also to develop mechanisms that allow the system to automatically respond to changes in data access patterns, fluctuations in cloud provider performance, changes in tariff plans, and regulatory requirements.

The scientific novelty of the research lies in solving an important scientific task of developing an approach for distributed storage and access to data with dynamically changing storage conditions in a multi-cloud environment and implementing practical methods based on it. A method for integrating reinforcement learning algorithms and multi-objective evolutionary algorithms for optimizing the process of distributed data storage is proposed, which, unlike existing ones, automates and optimizes the provider selection process and increases the efficiency of decision-making in a multi-cloud environment with dynamically changing data storage conditions. An ontological model of data distribution and storage is proposed, which formalizes and systematizes complex relationships between performance indicators and variable factors that affect the process of data distribution and storage depending on user priorities and statistical indicators of

cloud service providers' efficiency. The mathematical model of the system with dynamically changing data storage conditions in multi-cloud environments has been improved by integrating static and dynamic criteria, which enables automated consideration of changes in environmental parameters in real-time. The proposed approach to data storage in a multi-cloud environment is distinguished by the integration of artificial intelligence methods with multi-objective optimization methods.

As a result of the research, the theoretical developments are presented in the form of a comprehensive framework, the Intelligent Cloud Data Orchestrator (ICDO), which improves the efficiency of storage and access to data with dynamically changing storage conditions in multi-cloud environments. The ICDO approach achieved a 44.1% improvement in overall performance compared to traditional single-cloud approaches and a 14.9% improvement compared to the best alternative multi-cloud approach. Based on the developed ontological and mathematical models, software was created for optimizing data placement and management, which improves performance, reliability, and security of data storage while reducing costs and complying with regulatory requirements. Practical implementation of the developed software for managing data with dynamically changing storage conditions in multi-cloud environments was carried out in client projects of Tech-5 UG company, which specializes in processing large volumes of data and requires flexible solutions for their storage. As a result of the implementation, a reduction in average cloud infrastructure costs by 9% was recorded, as well as a reduction in data access latency by 25% compared to previous approaches. The developed approaches have been implemented in the educational and research programs of the ITT Department of NTUU "Igor Sikorsky KPI" for training specialists in the field of cloud computing.

Experimental studies confirm that the ICDO approach provides significant improvements in all key performance indicators, including productivity, cost, reliability, security, and adaptability, compared to existing solutions. The experimental study demonstrates that adaptability is a critical factor for effective data management in dynamic multi-cloud environments, and the adaptation and learning mechanisms implemented in the ICDO approach allow automating the process of responding to changes in data access patterns and environmental conditions. In addition, the study

confirms the effectiveness of an integrated approach to finding Pareto-optimal solutions compared to optimization by individual criteria. The study concludes by identifying promising directions for further work. These results significantly contribute to the development of the field of distributed data management and offer practical solutions for the urgent challenges of optimizing multi-cloud data storage.

Keywords: multi-cloud environment, cloud management, resource allocation, resource management, Quality of Service, Service Level Agreements, Multi-Criteria optimisation, ontological modeling.

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Kartashov A., Globa L. Optimizing Distributed Data Storage In Multi-Cloud Environments: Algorithmic Approach. Information and Telecommunication Sciences, ISSN 2312-41216 . No. 2 (2024) ст. 4-12.
DOI: <https://doi.org/10.20535/2411-2976.22024.4-12>

2. Kartashov A., Globa L. Towards Seamless Multi-Cloud Integration: Strategic Approach. Системи управління, навігації та зв'язку. Том 4 № 78 (2024), ст. 79-84
DOI: <https://doi.org/10.26906/SUNZ.2024.4.079> ISSN 2073-7394, 2024-11-28

3. Kartashov A., Globa L. Overview of the Approaches to Managing Distributed Storage and Access to Cloud Data. Proceedings of International Conference on Applied Innovation in IT IT. 2023. Volume 11, Issue 2, pp. 19-29 ISSN 2199-8876
DOI:10.25673/11299

4. А.Д. Карташов, Л.С. Глоба " Оптимізація у сфері зберігання даних в хмарних середовищах: пошук комплексного набору критеріїв", Зібрання тез до конференції "XVII Міжнародна науково-технічна конференція "Перспективи телекомунікацій 2024", 15-19 квітня, 2024. Київ. УДК 004.75. с. 196-198

ЗМІСТ

СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	14
ВСТУП.....	15
РОЗДІЛ 1 ПРОБЛЕМА ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ В МУЛЬТИХМАРНОМУ СЕРЕДОВИЩІ.....	21
1.1 Процес розподіленого зберігання у мультимарних середовищах	21
1.1.1 Визначення розподіленого зберігання та мультимарних архітектур	22
1.2 Еволюція хмарного зберігання: від одно-хмарного до мультимарного ..	29
1.3 Значення ефективного зберігання та доступу до даних.....	37
ВИСНОВКИ	48
РОЗДІЛ 2 МЕТОДИ ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ У МУЛЬТИХМАРНОМУ СЕРЕДОВИЩІ	49
2.1 Постановка задачі	49
2.2 Комплексний набір критеріїв для мультимарного зберігання даних	50
2.2.1 Критерії, пов'язані з продуктивністю.....	50
2.2.2 Критерії надійності та доступності	52
2.2.3 Критерії безпеки та відповідності	57
2.2.4 Економічні критерії та критерії управління ресурсами	60
2.2.5 Критерії інтероперабельності та портативності.....	64
2.2.6 Критерії сталого розвитку та екологічності	67
2.3 Онтологічна модель розподіленого зберігання даних у мультимарних середовищах	71
2.3.1 Концептуальна основа для мультимарного зберігання.....	72
2.3.2 Відношення між сутностями в мультимарних системах зберігання	73
2.3.3 Формалізація процесів та представлення робочих потоків	77
2.4 Математична модель зберігання даних у мультимарних середовищах ..	86
2.4.1 Формулювання проблеми.....	86
2.4.2 Цільові функції для мультимарної оптимізації.....	89
2.4.3 Формулювання обмежень	92
2.4.4 Підхід до багатоцільової оптимізації	95
ВИСНОВКИ	98

РОЗДІЛ 3 КОМПЛЕКСНИЙ ПІДХІД ЩОДО ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ.....	99
3.1 Гібридна структура підкріплювального навчання та багатоцільових еволюційних алгоритмів	99
3.1.1 Формулювання простору станів та дій	99
3.1.2 Розробка функції винагороди на основі багатокритеріальної оптимізації.....	101
3.1.3 Механізм покращення політики на основі MOEA.....	102
3.1.4 Механізми адаптації та навчання	103
3.2 Алгоритм оптимізації розміщення та доступу до даних.....	104
3.2.1 Початкова класифікація даних та аналіз вимог	105
3.2.2 Динамічна стратегія розміщення з використанням навчання з підкріпленням	107
3.2.3 Багатоцільова оптимізація параметрів зберігання.....	109
3.2.4 Адаптивне розпізнавання та оптимізація шаблонів доступу.....	111
3.2.5 Вирішення конфліктів та управління компромісами	113
ВИСНОВКИ	115
РОЗДІЛ 4 АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОГО КОМПЛЕКСНОГО МЕТОДУ ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ	116
4.1 Удосконалений підхід: Інтелектуальний оркестратор хмарних даних (ICDO).....	116
4.1.1 Огляд архітектурних компонентів ICDO.....	116
4.1.2 Рівень оркестрації (Core Orchestration Layer - COL)	118
4.1.3 Рівень оптимізації та прийняття рішень	119
4.1.4 Рівень виконання та управління (Execution Layer - EL).....	121
4.2 Експериментальне дослідження ефективності запропонованого комплексного підходу	124
4.2.1 Налаштування експерименту	124
4.2.2 Оцінка ефективності запропонованого інтегрованого підходу.....	128
4.2.3 Оцінка ефективності альтернатив у порівнянні з запропонованим комплексним підходом	139
ВИСНОВКИ	147

ЗАГАЛЬНІ ВИСНОВКИ.....	149
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	152
Додаток 1. Акт впровадження результатів наукового дослідження.....	163

СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

AЗ	Апаратне забезпечення
ВМ	Віртуальна машина
ІКМ	Інформаційно-комунікаційна мережа
ІКТ	Інформаційно-комунікаційні технології
ПЗ	Програмне забезпечення
ЦОД	Центр обробки даних
KPI	Key Performance Indicator
MDP	Markov Decision Process
QoS	Quality of Service
RAM	Random-Access Memory
RR	Round Robin
SDN	Software-Defined Networking
SLA	Service Level Agreement
ICDO	Intelligent Cloud Data Orchestrator
ROI	Return of Investment
TCO	Total Cost of Ownership
HOA	Hybrid Optimization Approach
SCO	Single Cloud Optimization
SMCD	Static Multi-Cloud Distribution

ВСТУП

Актуальність теми. Проблематика оптимізації мультимедійних середовищ активно досліджується науковою спільнотою протягом останнього десятиліття [1-18]. Значний внесок у розвиток теоретичних засад зробили роботи, присвячені моделюванню розподілених сховищ даних та розробці критеріїв їх оцінювання [18-32]. Зокрема, було запропоновано ряд моделей для формалізації взаємозв'язків між компонентами хмарної інфраструктури та методики оцінки якості обслуговування (QoS).

Проте аналіз існуючих досліджень виявив суттєві прогалини у вирішенні задачі вибору постачальників хмарних послуг для розподіленого розміщення даних:

1. Більшість запропонованих методів базується на статичних моделях, які не враховують динамічну природу хмарного середовища та постійні зміни характеристик сервісів [27-34].

2. Наявні рішення здебільшого розглядають обмежений набір параметрів оптимізації, не враховуючи комплексний характер взаємодії між різними факторами продуктивності та вартості [31,33,35].

3. Відсутні ефективні механізми автоматичної адаптації до змін у конфігурації мультимедійного середовища та вимог користувачів [36-38].

4. Існуючі підходи не забезпечують належного балансу між взаємопов'язаними цілями оптимізації, такими як продуктивність, надійність та економічна ефективність [29-35].

Зазначені обмеження суттєво знижують ефективність існуючих рішень в умовах реальної експлуатації розподілених систем зберігання даних. Це обумовлює необхідність розробки нового підходу, який би створював адаптивну систему прийняття рішень.

Таким чином, розробка комплексного підходу щодо оптимізації розподіленого зберігання та доступу до даних, здатного враховувати динамічну природу мультимедійного середовища та множинні критерії оптимізації, є актуальним

науковим завданням, вирішення якого дозволить подолати обмеження існуючих підходів та підвищити ефективність розподіленого зберігання даних.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась згідно з планом наукових досліджень кафедри інформаційно-комунікаційних технологій та систем Навчально-наукового Інституту телекомунікаційних систем:

в рамках держбюджетної теми: 2218п «Гетерогенна мережа збору, передачі та обробки інформації для системи розподіленої генерації MicroGrid» (номер державної реєстрації 0119U001184);

в рамках держбюджетної теми: 2313п «Побудова інформаційно-аналітичної платформи для супроводження функціонування кіберфізичних систем» (номер державної реєстрації 0120U102298);

в рамках міжнародного проекту “IDEA-East-Hub: International Innovation Hub for Data Science and renewable Energy”, DAAD Program - HAW.International Modul за співпрацею НТУУ «КПІ ім. І. Сікорського» та Анхальтським університетом прикладних наук Кьотен, Німеччина (номер договору 57603761).

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення ефективності розподілу та зберігання даних в мультимарному середовищі за рахунок оцінювання показників доступності, затримки, надійності, пропускної здатності, вартості та ін., та завдяки утворенню віртуальної логічно-єдиної системи зберігання даних.

Для досягнення мети дослідження було поставлено та вирішено такі задачі:

1. Визначити основні характеристики, властивості, показники та критерії оцінювання ефективності розподіленого зберігання та доступу до даних у мультимарних середовищах.

2. Систематизувати та формально описати складні взаємозв'язки між технічними, економічними, регуляторними та екологічними аспектами управління даними в мультимарному середовищі, а також розробити взаємодоповнюючі онтологічну та математичну моделі, які формалізують складні взаємозв'язки між компонентами мультимарного середовища та відображають багатокритеріальну

природу оптимізаційної задачі розміщення та управління даними.

3. Запропонувати метод, який ефективно інтегрує навчання з підкріпленням та багатоцільові еволюційні алгоритми для ефективного визначення Парето-оптимальних рішень та дозволяє розробити механізм адаптації, який забезпечує своєчасне реагування на зміни в шаблонах доступу до даних та в умовах мультимарного середовища зберігання.

4. Спроекувати та реалізувати фреймворк, який інтегрує методи штучного інтелекту з технологіями багатокритеріальної оптимізації для забезпечення адаптивних та комплексних рішень управління даними в мультимарному середовищі з врахуванням динамічно-змінюваних умов їх зберігання.

5. Експериментально оцінити ефективність запропонованого підходу за ключовими показниками продуктивності та визначити оптимальні сценарії застосування запропонованого фреймворку у різних контекстах розгортання рішень для мультимарних сховищ даних порівняно з існуючими.

Об'єкт дослідження – Процес розподілу, зберігання та доступу до даних в мультимарних середовищах з динамічно-змінюваними умовами їх зберігання.

Предмет дослідження – формалізовані моделі та методи підвищення ефективності розподілу, зберігання даних та доступу до них в мультимарному середовищі.

Методи дослідження, застосовані для вирішення поставлених завдань:

1. Методи інтелектуального аналізу даних – для аналізу об'єкта дослідження, його формалізації, виділення сутностей та структурних взаємозв'язків у досліджуваній системі.

2. Методи навчання з підкріплення – Reinforcement Learning (RL) та багатоцільових еволюційних алгоритмів – Multi-Objective Evolutionary Algorithms (МОЕМ) для вибору Парето-оптимальних рішень для розподілу даних в мультимарних середовищах з динамічно-змінюваними умовами їх зберігання.

3. Методи імітаційного моделювання та лабораторного експерименту – для перевірки ефективності застосування запропонованих математичних моделей та методів та їх порівняння з існуючими аналогами.

Наукова новизна отриманих результатів:

Наукова новизна дисертаційної роботи полягає у вирішенні важливого наукового завдання розробки підходу щодо розподіленого зберігання та доступу до даних з динамічно-змінюваними умовами зберігання у мультимарному середовищі та реалізації практичних методів на його основі. Основні наукові результати, що визначають новизну роботи:

1. Запропоновано метод інтеграції алгоритмів навчання з підкріплення та багатоцільових еволюційних алгоритмів для оптимізації процесу розподіленого зберігання даних, який на відміну від існуючих, автоматизує й оптимізує процес вибору провайдера та підвищує ефективність прийняття рішень у мультимарному середовищі з динамічно-змінюваними умовами зберігання даних.

2. Запропоновано онтологічну модель розподілу і зберігання даних, що формалізує та систематизує складні взаємозв'язки між показниками ефективності та змінними факторами, які впливають на процес розподілу і зберігання даних в залежності від пріоритетів користувачів та статистичних показників ефективності надання послуг хмарними провайдерами, що дозволяє оптимізувати витрати та значно прискорити процес вибору хмарного провайдера завдяки чіткому розумінню причинно-наслідкових зв'язків між параметрами системи.

3. Удосконалено математичну модель системи з динамічно-змінюваними умовами зберігання даних в мультимарних середовищах шляхом інтеграції статичних та динамічних критеріїв, яка надає можливість автоматизованого врахування змін параметрів середовища в режимі реального часу, що забезпечує більш ефективний процес збереження даних та доступу до них в мультимарному середовищі.

Запропоновані наукові результати у сукупності створюють теоретичне підґрунтя для розробки ефективних систем управління розподіленим зберіганням даних з динамічно-змінюваними умовами їх зберігання у мультимарних середовищах та відкривають нові можливості для оптимального використання хмарних ресурсів.

Практичне значення одержаних результатів:

1. Теоретичні розробки дисертації представлені автором у вигляді комплексного фреймворку Інтелектуальний Оркестратор Хмарних Даних – Intelligent Cloud Data Orchestrator (ICDO), який дозволяє підвищити ефективність зберігання та доступу до даних з динамічно-змінюваними умовами їх зберігання у мультихмарних середовищах .

2. На основі розроблених онтологічних та математичних моделей створено програмне забезпечення для оптимізації розміщення та управління даними, яке дозволяє підвищити продуктивність, надійність та безпеку зберігання даних з одночасним зниженням витрат та дотриманням регуляторних вимог.

3. Виконано практичне впровадження розробленого програмного забезпечення для управління даними з динамічно-змінюваними умовами їх зберігання в мультихмарних середовищах у компаніях, що спеціалізуються на обробці великих обсягів даних та потребують гнучких рішень для їх зберігання.

4. Розроблені підходи впроваджено у навчальні та дослідницькі програми кафедри ІТТ НТУУ «КПІ» імені Ігоря Сікорського для підготовки фахівців у сфері хмарних обчислень, автоматизації процесів проектування розподілених систем у мультихмарному середовищі, що сприяє покращенню практичних навичок роботи з багатохмарними середовищами.

Особистий внесок здобувача. Дисертаційна робота узагальнює результати теоретичних та експериментальних досліджень, проведених автором самостійно. Основні результати, отримані автором особисто: формалізована модель об'єкта дослідження у вигляді онтології процесу розподілу і зберігання даних у мультихмарному середовищі; математична модель системи з динамічно-змінюваними умовами зберігання даних, що інтегрує статичні та динамічні критерії; оригінальний підхід до інтеграції алгоритмів Reinforcement Learning та Multi-Objective Evolutionary Algorithms; удосконалений комплексний набір критеріїв оцінювання систем зберігання даних; експериментальна перевірка ефективності запропонованого комплексного підходу, реалізованого у фреймворку ICDO та впровадженого у виробниче середовище.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи були представлені й одержали схвалення на:

– Міжнародній науково-технічній конференції «Проблеми телекомунікацій» (м. Київ, Україна, 2024 р).

– Міжнародній науково-технічній конференції «PyCon Odessa 2022» (м. Одеса, Україна, 2021 р)

– Міжнародній науково-технічній конференції «PyCon Munich 2023» (м. Мюнхен, Німеччина, 2023 р)

– Міжнародній науково-технічній конференції «PyCon Munich 2024» (м. Мюнхен, Німеччина, 2024 р)

Публікації. За результатами досліджень опубліковано 8 наукових публікаціях, серед яких: 2 статті у наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України; 1 стаття у періодичних наукових виданнях, проіндексованих у базах даних WoS та/або Scopus; також результати дисертації було апробовано у 4 фахових конференціях.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 110 найменувань. Загальний обсяг роботи 163 сторінки, з яких 128 сторінок основного тексту, 12 сторінок використаних джерел та 2 сторінки додатків. Робота містить 21 рисунок, 11 таблиць.

РОЗДІЛ 1

ПРОБЛЕМА ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ В МУЛЬТИХМАРНОМУ СЕРЕДОВИЩІ

1.1 Процес розподіленого зберігання у мультихмарних середовищах

Експоненціальне зростання цифрових даних [1,2] у сучасну епоху спричинило зсув парадигми у підході організацій до зберігання, управління та доступу до даних. Традиційні централізовані архітектури зберігання виявилися недостатніми для вирішення проблем, пов'язаних з обсягом, швидкістю та різноманітністю даних, що генеруються сучасними додатками та сервісами. Цей недолік стимулював розвиток розподілених систем зберігання, які стали життєздатним рішенням для управління великомасштабними сховищами даних. Розподілений характер цих систем дозволяє забезпечити горизонтальну масштабованість, відмовостійкість та покращені характеристики продуктивності, які є необхідними для сучасних додатків з інтенсивним використанням даних.

Останніми роками хмарні обчислення революціонізували ІТ-ландшафт, пропонуючи доступ до обчислювальних ресурсів на вимогу, включаючи зберігання, обчислювальні потужності та мережеві можливості [3]. Хмарні сервіси зберігання, зокрема, набули значної популярності завдяки своїй здатності забезпечувати масштабовані, економічно ефективні та надійні рішення для зберігання даних без необхідності значних початкових інвестицій в інфраструктуру. Однак, оскільки організації все більше покладаються на хмарні сервіси для своїх критичних операцій, умови та міркування щодо прив'язки до постачальника, доступності сервісів, суверенітету даних та відповідності нормативним вимогам стали більш вираженими.

Ці умови призвели до появи концепції мультихмарних архітектур, де організації розподіляють свої дані та додатки між кількома постачальниками хмарних послуг. Мультихмарні середовища пропонують численні переваги, включаючи зменшення залежності від одного постачальника, покращену стійкість до збоїв сервісів, розширене географічне покриття та можливість використовувати унікальні переваги різних хмарних платформ. Тим не менш, розподілений характер мультихмарного зберігання вводить складні проблеми, пов'язані з узгодженістю даних, затримкою доступу, оптимізацією ресурсів та безпекою.

Цей розділ досліджує фундаментальні концепції, виклики та можливості, пов'язані з розподіленим зберіганням у мультихмарних середовищах [4]. Він починається зі встановлення комплексного визначення розподіленого зберігання та мультихмарних архітектур, після чого слідує дослідження еволюційної траєкторії

від одно-хмарних до мультихмарних парадигм зберігання. Розділ завершується висвітленням значення ефективних механізмів зберігання та доступу до даних у мультихмарних контекстах, створюючи основу для наступних розділів, які заглиблюються у запропоновані підходи та рішення.

1.1.1 Визначення розподіленого зберігання та мультихмарних архітектур

Розподілені системи зберігання даних представляють фундаментальний зсув від традиційних централізованих парадигм зберігання, втілюючи складну взаємодію апаратних, програмних та мережевих компонентів, призначених для зберігання та управління даними на кількох вузлах або локаціях. Для встановлення комплексного розуміння розподіленого зберігання даних у мультихмарних середовищах необхідно спочатку окреслити основні концепції, характеристики та архітектурні принципи, що лежать в основі цих систем.

У своїй суті розподілене зберігання даних відноситься до системної архітектури, де дані зберігаються на кількох фізичних або віртуальних пристроях зберігання, які можуть бути географічно розподіленими, але функціонують колективно як єдина сутність зберігання. На відміну від централізованих систем зберігання, де дані знаходяться на одному пристрої зберігання або тісно пов'язаному кластері, розподілені системи зберігання розділяють та реплікують дані на численних вузлах, забезпечуючи паралельний доступ, відмовостійкість та горизонтальну масштабованість.

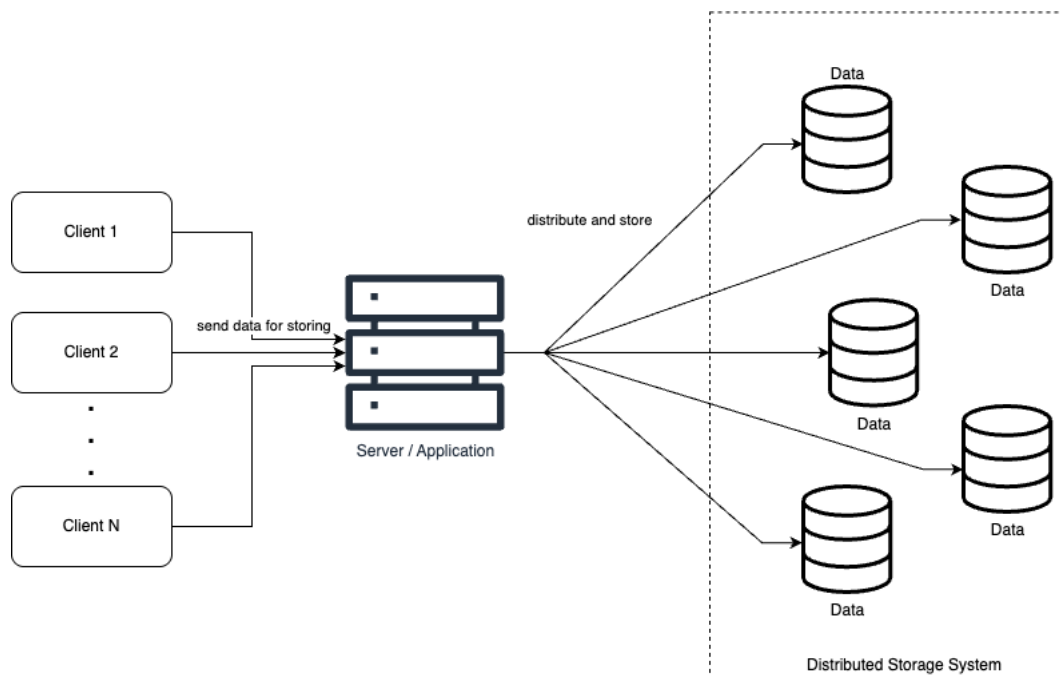


Рисунок 1.1 Системна архітектура розподіленого зберігання даних

Теоретичну основу розподілених систем зберігання даних можна простежити до теорії розподілених систем, яка вирішує проблеми координації кількох обчислювальних елементів для досягнення спільної мети. У контексті зберігання це перетворюється на координацію кількох вузлів зберігання для надання когерентної та надійної послуги зберігання даних. Фундаментальні властивості розподілених систем зберігання, як визначено теоремою CAP Брюера [5], включають узгодженість (C), доступність (A) та стійкість до розділення (P). Узгодженість означає, що всі вузли в системі бачать одні й ті самі дані одночасно. Доступність гарантує, що кожен запит до системи отримує відповідь, без гарантії, що вона містить найновішу версію даних. Стійкість до розділення забезпечує, що система продовжує працювати, незважаючи на довільну втрату повідомлень або відмову частини системи.

Теорема CAP стверджує, що для розподіленої системи неможливо одночасно забезпечити всі три умови. Цей фундаментальний компроміс має глибокі наслідки для проектування та експлуатації розподілених систем зберігання, що вимагає ретельного розгляду конкретних вимог та обмежень прикладної області. Математично теорему CAP можна виразити формулою 1.1 наступним чином:

Нехай S – розподілена система зберігання, а $C(S)$, $A(S)$ та $P(S)$ – булеві функції, які вказують, чи задовольняє S узгодженість, доступність та стійкість до розділення відповідно. Теорема CAP стверджує, що:

$$\forall S: \neg(C(S) \wedge A(S) \wedge P(S)). \quad (1.1)$$

Це означає, що проектувальники системи повинні надавати пріоритет двом із цих властивостей за рахунок третьої, що призводить до поширених класифікацій систем. CP-системи надають пріоритет узгодженості та стійкості до розділення над доступністю. AP-системи надають пріоритет доступності та стійкості до розділення над узгодженістю. CA-системи надають пріоритет узгодженості та доступності над стійкістю до розділення, хоча вони загалом вважаються непрактичними для розподілених систем, що працюють через ненадійні мережі. На рис 1.2 зображено деталі CAP теорема.

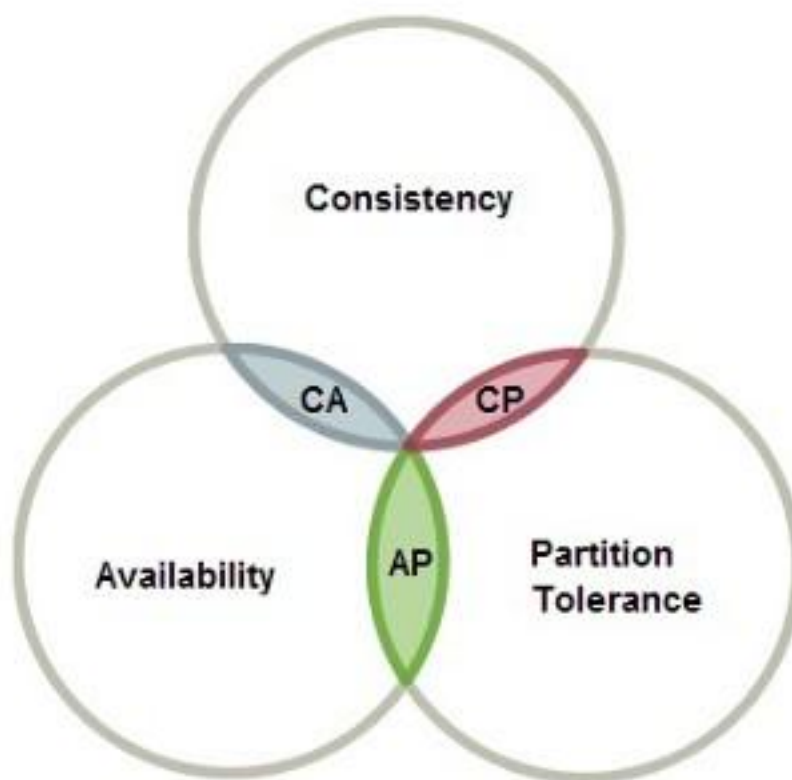


Рисунок 1.2 CAP теорема

Окрім теореми CAP, розподілені системи зберігання характеризуються кількома ключовими атрибутами. Розподіл даних визначає спосіб, яким дані розділяються та розподіляються між вузлами зберігання. Поширені стратегії включають шардинг (розділення даних на неперекриваючі розділи та їх розподіл між вузлами), реплікацію (створення та підтримка кількох копій даних на різних вузлах) та надлишкове кодування (erasure coding – кодування даних у фрагменти таким чином, що підмножини фрагментів достатньо для відновлення оригінальних даних).

Моделі узгодженості надають гарантії щодо видимості та впорядкування оновлень розподілених даних. Вони варіюються від сильних моделей узгодженості

до слабших моделей, таких як остаточна узгодженість. Відмовостійкість забезпечує здатність підтримувати роботу системи, незважаючи на відмову окремих компонентів, зазвичай через механізми надмірності, такі як реплікація або надлишкове кодування. Масштабованість дозволяє пристосовуватися до зростаючих обсягів даних та швидкості запитів шляхом додавання більшої кількості ресурсів до системи. Продуктивність визначає ефективність, з якою система може виконувати операції зберігання та отримання, часто вимірюється з точки зору затримки, пропускну здатності та використання ресурсів [14].

Математичне формулювання розподілу даних у розподілених системах зберігання можна представити наступним чином:

Нехай D – набір даних, а $N = \{n_1, n_2, \dots, n_k\}$ – набір з k вузлів зберігання. Функція розподілу даних $f: D \rightarrow 2^N$ відображає кожен елемент даних $d \in D$ на підмножину вузлів $f(d) \subseteq N$, які зберігають копії або фрагменти d .

Для реплікації з коефіцієнтом реплікації r обмеження є:

$$\forall d \in D: |f(d)| = r. \quad (1.2)$$

Для надлишкового кодування з параметрами (n, k) , де n - загальна кількість фрагментів, а k - мінімальна кількість, необхідна для відновлення:

$$\forall d \in D: |f(d)| = n \text{ та будь-які } k \text{ вузлів у } f(d) \text{ можуть відновити } d. \quad (1.3)$$

Мультихмарні архітектури представляють еволюцію парадигм хмарних обчислень, де організації використовують послуги кількох хмарних провайдерів, а не покладаються на одного провайдера. Цей підхід відрізняється від **гібридних хмарних архітектур**, які зазвичай включають комбінацію приватних та публічних хмарних ресурсів. У мультихмарному середовищі ресурси розподілені між кількома публічними хмарними платформами, кожна з яких потенційно пропонує різні послуги, моделі ціноутворення, географічне покриття та характеристики продуктивності. Зокрема, рекомендації ITU-T Y.3500-Y.3599 [7] визначають архітектурні принципи хмарних обчислень та встановлюють базові вимоги до інтегрованості хмарних сервісів. Відповідно до цих рекомендацій, системи розподіленого зберігання даних повинні забезпечувати безшовну інтеграцію між різними хмарними провайдерами, що є критичним аспектом для мультихмарних архітектур. Документ NIST SP 500-292 "Cloud Computing Reference Architecture" [8] визначає еталонну архітектуру хмарних обчислень та підкреслює важливість стандартизованих інтерфейсів для забезпечення ефективної взаємодії між

компонентами мультихмарної системи. Рис 1.3 відображає архітектуру публічної хмари згідно стандарту.

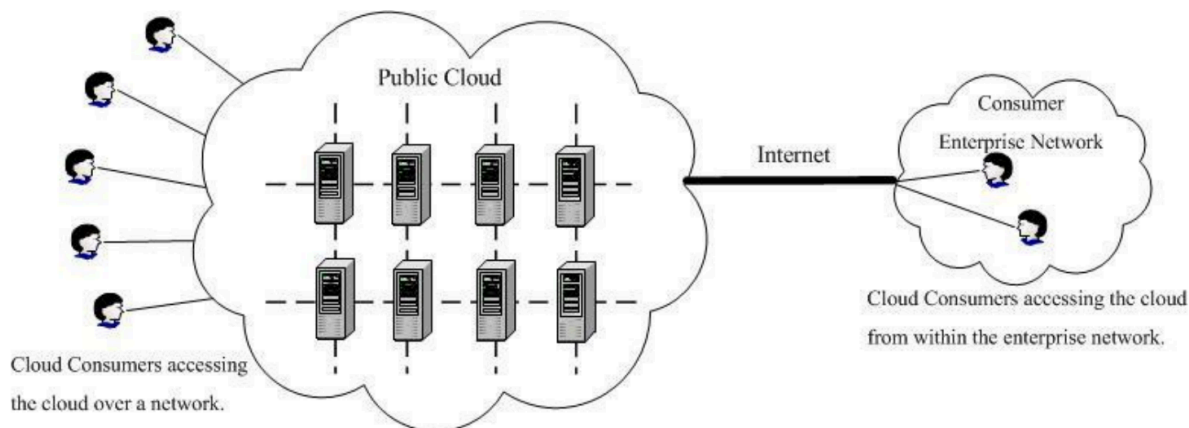


Рисунок 1.3 NIST SP 500-292 Public Cloud Reference Architecture

Формальне визначення мультихмарної архітектури можна сформулювати наступним чином: **мультихмарна архітектура** - це модель розгортання, в якій обчислювальні ресурси, додатки та дані організації розподілені між двома або більше постачальниками хмарних послуг з метою оптимізації продуктивності, вартості, надійності або цілей відповідності нормативним вимогам (далі відповідності) [14,15,18].

Мультихмарні архітектури можна класифікувати за кількома вимірами. За моделями розгортання вони поділяються на пасивну мультихмару (використання різних хмарних провайдерів для різних додатків або послуг з мінімальною інтеграцією між ними), активну мультихмару (динамічний розподіл робочих навантажень між кількома хмарними провайдерами на основі продуктивності, вартості або інших критеріїв) та гібридну мультихмару (поєднання приватних хмарних ресурсів з послугами кількох публічних хмарних провайдерів) [19].

За рівнями інтеграції мультихмарні архітектури включають інтеграцію на рівні додатків (різні компоненти додатка розгортаються на різних хмарних платформах), інтеграцію на рівні даних (дані розподіляються або реплікуються між кількома хмарними сервісами зберігання) та інтеграцію на рівні інфраструктури (обчислювальні, мережеві та ресурси зберігання надаються та управляються на кількох хмарних платформах).

За підходами до управління мультихмарні архітектури можуть використовувати централізоване управління (єдина площа управління контролює ресурси на всіх хмарних провайдерах), федеративне управління (кожне хмарне середовище управляється окремо, з механізмами координації для

міжхмарних операцій) або автономне управління (кожне хмарне середовище працює незалежно, з мінімальною координацією) [20,21].

Математичне представлення розподілу ресурсів у мультихмарному середовищі можна сформулювати наступним чином:

Нехай $CP = \{cp_1, cp_2, \dots, cp_m\}$ – набір з m хмарних провайдерів, а $R = \{r_1, r_2, \dots, r_n\}$ – набір з n ресурсів (наприклад, зберігання, обчислення, мережа). Кожен хмарний провайдер cp_j пропонує підмножину ресурсів $R_j \subseteq R$ з відповідними характеристиками вартості, продуктивності та доступності. Проблему розподілу ресурсів у мультихмарному середовищі можна сформулювати як задачу оптимізації:

$$\min_{x_{ij}} \sum_{i=1}^n \sum_{j=1}^m x_{ij} \cdot \text{cost}(r_i, cp_j). \quad (1.4)$$

за умови:

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i \in \{1, 2, \dots, n\}.$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\}.$$

$$x_{ij} = 0 \quad \text{якщо } r_i \notin R_j.$$

де x_{ij} – бінарна змінна, що вказує, чи ресурс r_i призначений хмарному провайдеру cp_j , а $\text{cost}(r_i, c_j)$ представляє вартість надання ресурсу r_i на хмарному провайдері cp_j .

Це формулювання можна розширити, щоб включити додаткові обмеження, пов'язані з продуктивністю, надійністю, суверенітетом даних та іншими вимогами.

Конвергенція розподілених систем зберігання та мультихмарних архітектур призводить до мультихмарного розподіленого зберігання, яке представляє унікальні виклики та можливості. У цьому контексті ресурси зберігання розподілені не лише між кількома вузлами, але й між кількома хмарними провайдерами, кожен з яких має власні сервіси зберігання, API, моделі ціноутворення та характеристики продуктивності.

Мультихмарне розподілене зберігання можна визначити як архітектуру зберігання, яка розподіляє дані між сервісами зберігання, що надаються кількома хмарними провайдерами, представляючи при цьому уніфікований інтерфейс для операцій зберігання та отримання даних. На рисунку 1.4 зображено сценарій

процесу розподілу даних до хмарних провайдерів. Рівень прийняття рішень у цьому контексті можна визначити як абстрактну логіку, що відповідає за вибір підходу щодо розміщення даних на сервісах зберігання різних хмарних провайдерів. [22,23]

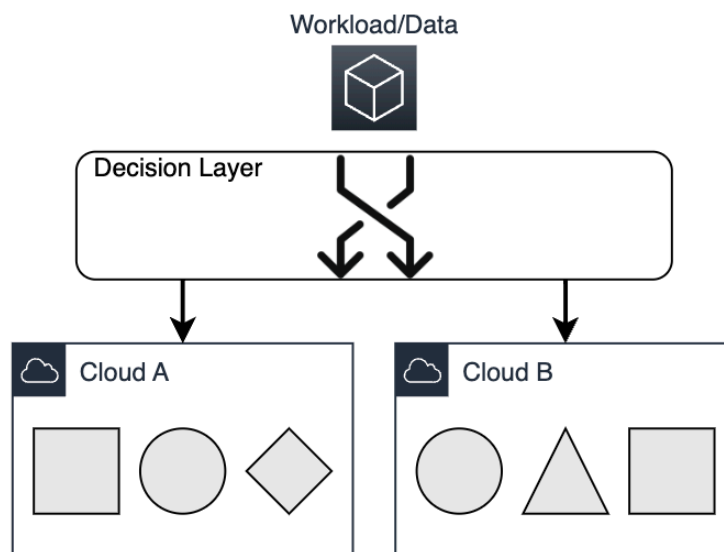


Рисунок 1.4 Сценарій процесу розподілу даних

Ключові компоненти мультихмарного розподіленого зберігання включають рівень розподілу даних, який відповідає за розділення даних та визначення їх розміщення між хмарними провайдерами; управління метаданими, яке підтримує інформацію про місцезнаходження, формат та властивості даних, що зберігаються на кількох хмарах; управління узгодженістю, яке забезпечує узгодженість даних, незважаючи на їх розподіл між кількома хмарними середовищами; контроль доступу та безпека, які забезпечують політики аутентифікації, авторизації та шифрування на всіх хмарних провайдерах; а також моніторинг та оптимізація, які постійно оцінюють продуктивність, вартість та надійність операцій зберігання та відповідно коригують розміщення даних. ISO/IEC 27018 [11] встановлює специфічні вимоги до захисту персональних даних у публічних хмарах, а CSA STAR [12] пропонує методологію оцінки рівня безпеки хмарних провайдерів. Математичну модель для розміщення даних у мультихмарному розподіленому зберіганні можна сформулювати наступним чином:

Нехай $D = \{d_1, d_2, \dots, d_p\}$ – набір з p об'єктів даних, а $CP = \{cp_1, cp_2, \dots, cp_m\}$ – набір з m хмарних провайдерів. Кожен хмарний провайдер cp_j має пов'язану вартість зберігання $cost_j$, затримку доступу $latency_j$ та надійність $reliability_j$.

Проблема розміщення даних включає визначення відображення $g: D \rightarrow 2^C$, яке призначає кожен об'єкт даних d_i підмножині хмарних провайдерів $g(d_i) \subseteq CP$, з урахуванням обмежень на реплікацію, вартість, продуктивність та надійність.

Наприклад, щоб мінімізувати загальну вартість зберігання, забезпечуючи при цьому, що кожен об'єкт даних реплікується на принаймні r хмарних провайдерів:

$$\min_g \sum_{i=1}^p \sum_{j \in g(d_i)} \text{size}(d_i) \cdot \text{cost}_j, \quad (1.5)$$

за умови:

$$|g(d_i)| \geq r \quad \forall i \in \{1, 2, \dots, p\},$$

де $\text{size}(d_i)$ представляє розмір об'єкта даних d_i .

Можна включити додаткові обмеження для задоволення вимог, пов'язаних із затримкою доступу, географічним розподілом, суверенітетом даних та іншими факторами.

Розподілене зберігання у мультихмарних середовищах представляє складну взаємодію принципів розподілених систем, парадигм хмарних обчислень та технологій зберігання. Визначення та характеристика цих систем забезпечують основу для розуміння викликів та можливостей, які вони представляють, і які будуть детальніше досліджені в наступних розділах [25].

1.2 Еволюція хмарного зберігання: від одно-хмарного до мультихмарного

Траєкторія еволюції хмарного зберігання представляє захоплюючу хроніку технологічних інновацій, ринкової динаміки та еволюціонуючих організаційних вимог. Цей розділ простежує історичний розвиток парадигм хмарного зберігання, від появи централізованих сервісів хмарного зберігання до поширення мультихмарних архітектур зберігання, висвітлюючи ключові технологічні досягнення, ринкові сили та організаційні імперативи, які сформували цю еволюцію.

Концепція хмарного зберігання має свої корені в ширшій парадигмі утилітарних обчислень, яка розглядає обчислювальні ресурси як вимірювану послугу, подібну до традиційних комунальних послуг, таких як електрика або вода. Теоретичні основи цього підходу можна простежити до 1960-х років, коли вчений-комп'ютерник Джон Маккарті сформулював бачення обчислень як публічної комунальної послуги. Однак лише на початку 2000-х років технологічна інфраструктура та бізнес-моделі достатньо дозріли, щоб реалізувати це бачення у формі послуг хмарних обчислень.

Ранні сервіси хмарного зберігання з'явилися як розширення веб-хостингу та пропозицій провайдерів прикладних сервісів (ASP), що надавали прості

можливості зберігання та обміну файлами. Ці сервіси були в першу чергу розроблені для вирішення конкретних випадків використання, таких як резервне копіювання, обмін файлами або хостинг веб-контенту, а не для служіння як комплексні платформи зберігання.

Математичну модель для ранніх сервісів хмарного зберігання можна представити як просту функцію відображення

$$f: K \rightarrow V ,$$

де K – набір ключів (наприклад, імена файлів або ідентифікатори об'єктів), а V – набір значень (наприклад, вміст файлів). Ця модель відображає просту парадигму ключ-значення, яка лежала в основі багатьох ранніх сервісів хмарного зберігання.

Ландшафт хмарного зберігання зазнав значної трансформації з запуском Amazon Simple Storage Service (S3) у 2006 році, який широко вважається першим сучасним сервісом хмарного зберігання. S3 представив масштабований, високодоступний сервіс об'єктного зберігання з простим API на основі HTTP, встановивши шаблон, який вплинув на подальші пропозиції хмарного зберігання. Період з 2006 по 2010 рік став свідком появи кількох основних платформ хмарного зберігання, включаючи Microsoft Azure Blob Storage, Google Cloud Storage та Rackspace Cloud Files [16]. Ці платформи мали кілька спільних характеристик, таких як парадигма об'єктного зберігання, де дані зберігаються як об'єкти з пов'язаними метаданими, доступ до яких здійснюється через API на основі HTTP; ціноутворення за принципом оплати за використання, де витрати на зберігання базуються на фактичному використанні, усуваючи необхідність у попередніх капітальних витратах; еластична масштабованість, де ємність зберігання може динамічно збільшуватися або зменшуватися залежно від попиту; та географічний розподіл, де дані можуть бути реплікованими на кількох центрах обробки даних у мережі провайдера для підвищення доступності та довговічності. Математичну модель для цих сервісів об'єктного зберігання можна виразити як

$$f: (K, M) \rightarrow V ,$$

де K - набір ключів, M - набір атрибутів метаданих, а V - набір значень. Ця модель відображає розширені можливості сервісів об'єктного зберігання, які дозволяють асоціювати метадані зі збереженими об'єктами.

Протягом цього періоду сервіси хмарного зберігання еволюціонували від простих сховищ до комплексних платформ, що пропонують розширені функції, такі як версіонування (підтримка кількох версій об'єкта для захисту від

випадкового видалення або модифікації), контроль доступу (детальні дозволи для контролю того, хто може отримати доступ до збережених даних), управління життєвим циклом (автоматизовані політики для переходу даних між рівнями зберігання або їх видалення на основі віку або шаблонів доступу), сповіщення про події (запуск дій або робочих процесів у відповідь на події зберігання, такі як створення об'єкта або видалення) та інтеграція з доставкою контенту (безперебійна інтеграція з мережами доставки контенту для ефективного розповсюдження збереженого контенту – **Content Delivery Network CDN**) [23].

Додатково слід відзначити, що хмарні дата-центри активно використовуються операторами Інформаційно-Комунікаційних мереж (ІКМ) для аналізу Великих даних (**Big Data**), що дозволяє покращити технічні параметри мережі. Це також сприяє персоналізації послуг та ефективному розподілу ресурсів. Рекомендація МСЕ-Т У.3652 визначає концепцію мережі, керованої великими даними (bDDN – Big data-driven network) [65-69], яка в свою чергу визначає методи адміністрування та оптимізації мережі на основі накопичених даних.

Також технологія периферійних обчислень (**Edge computing**) відіграє ключову роль у мережах 5G. Вона передбачає розміщення обчислювальних ресурсів ближче до користувача, що збільшує пропускну здатність і зменшує затримки. Дослідження Ericsson показало, що підтримка хмарної інфраструктури відкриває нові бізнес-можливості для операторів ІКМ.

Сучасні ІКМ потребують розвиненої обчислювальної інфраструктури, а отже, хмарні центри обробки даних стають невід'ємною частиною мережі, які виконують не лише традиційні функції авторизації та білінгу. У мережах 5G обчислювальна інфраструктура забезпечує роботу технологій SDN. Вона підтримує логічний поділ мережі та надання послуг Edge Computing. Вона також необхідна для аналізу великих даних для підвищення якості послуг.

Рис. 1.5 ілюструє місце хмарної інфраструктури в архітектурі сучасної ІКМ. На ньому показано загальну структуру та принципи побудови мережі 5G.

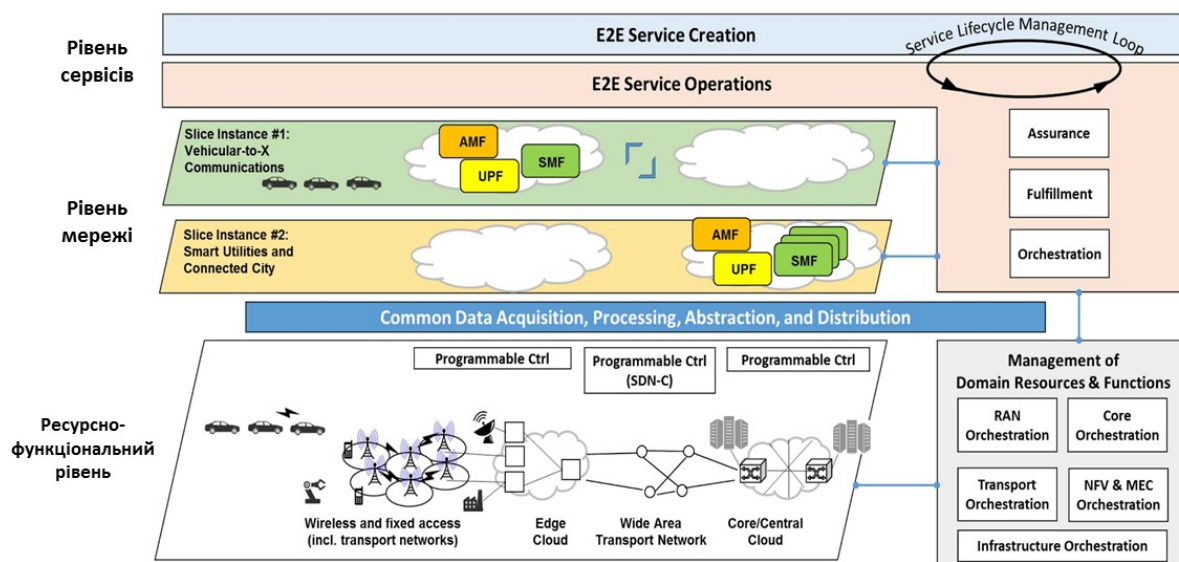


Рисунок 1.5 Загальна архітектура мережі 5G

Прийняття хмарного зберігання було в основному зумовлено конкретними випадками використання, такими як резервне копіювання та аварійне відновлення, хостинг веб-контенту, аналітика великих даних та бекенд мобільних додатків. Незважаючи на зростаюче прийняття хмарного зберігання, організації зазвичай покладалися на одного хмарного провайдера для своїх потреб у зберіганні на цьому етапі. Цей підхід був мотивований кількома факторами, включаючи простоту управління зберіганням в одному хмарному середовищі, тісну інтеграцію з іншими послугами, що пропонуються тим самим хмарним провайдером, уникнення витрат, пов'язаних з передачею даних між різними хмарними провайдерами, та розвиток експертизи в сервісах та інструментах зберігання конкретного хмарного провайдера. Однак одно-хмарний підхід також вводив значні ризики та обмеження, які врешті-решт призвели до переходу до мультихмарних архітектур зберігання.

Перехід від одно-хмарних до мультихмарних архітектур зберігання почав набирати обертів приблизно у 2015-2018 роках. Організації стали все більше остерігатися надмірної залежності від одного хмарного провайдера, що могло призвести до зменшення переговорної сили, обмеженої гнучкості та потенційних порушень, якщо провайдер змінював свої послуги або ціноутворення. Гучні збої основних хмарних сервісів підкреслили ризики покладання на одного провайдера для критичних потреб зберігання. Різні хмарні провайдери мали різні сильні сторони з точки зору розташування центрів обробки даних, що ставало все більш важливим, оскільки організації розширювалися глобально та стикалися з вимогами суверенітету даних. Хмарні провайдери почали диференціювати свої пропозиції зберігання унікальними функціями та можливостями, що робило привабливим для організацій використання кількох провайдерів для різних випадків використання.

Еволюціонуючі правила захисту даних, такі як Загальний регламент про захист даних (GDPR) Європейського Союзу, ввели вимоги, які було важко задовольнити в межах одного хмарного середовища.

Перехід до мультихмарного зберігання був полегшений кількома технологічними розробками, включаючи появу програмних бібліотек та фреймворків, які надавали уніфікований інтерфейс для взаємодії з різними сервісами хмарного зберігання; інструменти для управління ресурсами на кількох хмарних середовищах, включаючи ресурси зберігання; рішення для ефективної реплікації даних між різними сервісами хмарного зберігання; та покращені мережеві можливості для з'єднання ресурсів на різних хмарних середовищах [26].

Математичну модель для мультихмарного зберігання можна представити як:

$$f: (K, M, CP) \rightarrow V, \quad (1.6)$$

де K – набір ключів, M – набір атрибутів метаданих, CP – набір хмарних провайдерів, а V – набір значень. Ця модель відображає додаткову складність, введену необхідністю розглядати хмарного провайдера як параметр в операціях зберігання.

Ранні підходи до мультихмарного зберігання були в основному зосереджені на реплікації даних, де одні й ті самі дані зберігалися в кількох хмарних середовищах для підвищення доступності та зменшення прив'язки до постачальника. Цей підхід можна математично представити як:

$$\forall k \in K, \forall p \in CP: f(k, M_k, cp) = V_k, \quad (1.7)$$

де M_k – метадані, пов'язані з ключем k , а V_k – значення, пов'язане з ключем k .

Хоча цей підхід забезпечував надмірність, він також вводив проблеми, пов'язані з узгодженістю даних, економічною ефективністю та складністю управління.

Технологія 5G значно покращує продуктивність хмарних сервісів, дозволяючи їм працювати на межі мережі 5G. Наприклад, спеціалізований сервіс хмарного провайдера Amazon Web Services – AWS Wavelength, дозволяє розгортати хмарні сервіси на периферії мережі 5G для надання інференсу моделей машинного навчання та штучного інтелекту на основі даних з IoT пристроїв, встановлених в мережі 5G, майже в режимі реального часу.

Зі збільшенням кількості IoT пристроїв та об'єму запитів на інференс в режимі реального часу користувачі потребують функціонал з наднизькою затримкою. Мережа 5G, яка в 10 разів швидша за 4G, підвищує продуктивність

додатків на периферії, але все одно додатки повинні отримувати доступ до можливих сервісів і даних у хмарі, проходячи кілька мережевих стрибків від мобільних пристроїв до вежі стільникового зв'язку, до сайту агрегації, до інтернету і до хмари. Однак такий тип мережі збільшує затримку і не забезпечує роботу в режимі, близькому до реального часу, для робочих навантажень IoT, критичних до затримок. Розгортаючи хмарний сервіс на межі мережі 5G, як показано на рис 1.6, можна скоротити додаткові переходи за межами мережі 5G і мінімізувати затримку в мережі.

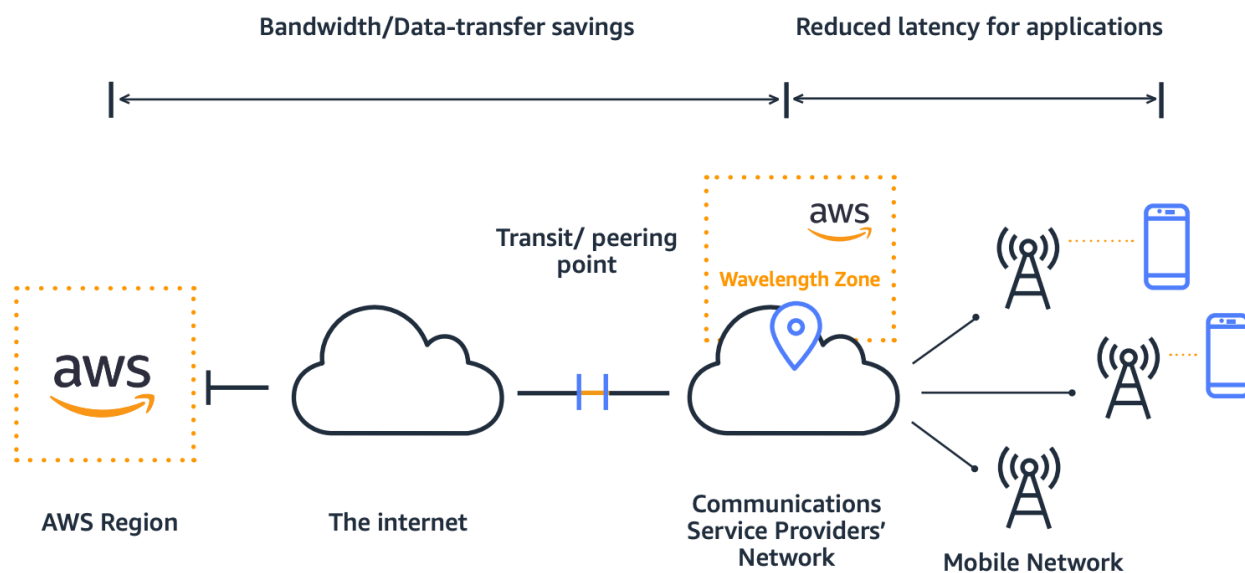


Рисунок 1.6 AWS Wavelength в мережі 5G

Оскільки організації набували досвіду з мультихмарними середовищами, почали з'являтися більш складні підходи до мультихмарного зберігання. Ці підходи вийшли за межі простої реплікації, щоб охопити більш нюансовані стратегії розподілу даних між хмарними провайдерами. [28-31]

Ключові характеристики цих передових мультихмарних архітектур зберігання включають інтелектуальне розміщення даних, де замість реплікації всіх даних на всіх хмарних провайдерах, дані вибірково розміщуються на основі таких факторів, як шаблони доступу, вартість, вимоги до продуктивності та нормативні обмеження; надлишкове кодування між хмарами, де замість повної реплікації, техніки надлишкового кодування використовуються для розподілу фрагментів даних між кількома хмарними провайдерами, зменшуючи накладні витрати на зберігання при збереженні надійності; багаторівневе зберігання між провайдерами, де різні рівні зберігання реалізуються на різних хмарних провайдерах на основі їх характеристик вартості та продуктивності; динамічна міграція даних, де дані

автоматично мігрують між хмарними провайдерами на основі змінних вимог, витрат або характеристик продуктивності; та уніфіковане управління метаданими, яке забезпечує централізоване управління метаданими на всіх хмарних провайдерах, що забезпечує ефективне виявлення та доступ до даних незалежно від того, де фізично зберігаються дані. [32-35]

Математичну модель для інтелектуального розміщення даних у мультихмарному зберіганні можна сформулювати як задачу оптимізації:

Нехай $D = \{d_1, d_2, \dots, d_n\}$ – набір об'єктів даних, а $CP = \{cp_1, cp_2, \dots, cp_m\}$ – набір хмарних провайдерів. Для кожного об'єкта даних d_i та хмарного провайдера cp_j визначаються наступні параметри: c_{ij} – вартість зберігання об'єкта даних d_i на хмарному провайдері cp_j ; l_{ij} – затримка доступу для об'єкта даних d_i на хмарному провайдері cp_j ; та r_{ij} – метрика надійності для об'єкта даних d_i на хмарному провайдері cp_j .

Нехай x_{ij} – бінарна змінна, що вказує, чи об'єкт даних d_i зберігається на хмарному провайдері cp_j . Проблему розміщення даних можна сформулювати як:

$$\min \sum_{i=1}^n \sum_{j=1}^m x_{ij} \cdot c_{ij}, \quad (1.8)$$

за умови:

$$\begin{aligned} \sum_{j=1}^m x_{ij} &\geq r_{min} \quad \forall i \in \{1, 2, \dots, n\}. \\ \sum_{j=1}^m x_{ij} \cdot l_{ij} &\leq l_{max} \quad \forall i \in \{1, 2, \dots, n\}. \\ \prod_{j=1}^m (1 - (1 - r_{ij}) \cdot x_{ij}) &\geq r_{min} \quad \forall i \in \{1, 2, \dots, n\}. \\ x_{ij} &\in \{0, 1\} \quad \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\}, \end{aligned}$$

де r_{min} – мінімальний необхідний коефіцієнт реплікації, l_{max} – максимально допустима затримка доступу, а r_{min} – мінімально прийнятна надійність.

Це формулювання відображає компроміси між вартістю, продуктивністю та надійністю в мультихмарних середовищах зберігання, дозволяючи оптимізувати рішення щодо розміщення даних.

Сучасний ландшафт мультихмарного зберігання характеризується різноманітною екосистемою рішень, починаючи від сервісів, специфічних для хмарних провайдерів, до сторонніх платформ, розроблених спеціально для мультихмарних середовищ. Ці рішення можна широко категоризувати як рішення хмарних провайдерів, де основні хмарні провайдери почали визнавати реальність мультихмарних середовищ і пропонують послуги, розроблені для полегшення інтеграції з іншими хмарними платформами; сторонні мультихмарні платформи, які забезпечують уніфіковане управління, переміщення даних та контроль доступу на кількох сервісах хмарного зберігання [36]; фреймворки мультихмарного зберігання з відкритим кодом, які надають інструменти та бібліотеки для створення рішень мультихмарного зберігання [37]; та користувацькі мультихмарні архітектури, де організації розробляють індивідуальні рішення, адаптовані до їх конкретних вимог, часто використовуючи комбінацію комерційних та компонентів з відкритим кодом.

Еволюція мультихмарного зберігання продовжує керуватися кількома ключовими тенденціями. Оскільки обчислювальні навантаження переміщуються ближче до краю мережі, мультихмарні архітектури зберігання еволюціонують, щоб включити можливості крайового зберігання, створюючи континуум від краю до хмари. Алгоритми машинного навчання використовуються для оптимізації рішень щодо розміщення даних, реплікації та кешування в мультихмарних середовищах на основі шаблонів доступу, динаміки вартості та метрик продуктивності. Поява парадигм безсерверних обчислень впливає на дизайн рішень мультихмарного зберігання, з акцентом на архітектури, керовані подіями, та моделі детального білінгу [38-40]. Технології розподіленого реєстру досліджуються як засіб підвищення безпеки, прозорості та стійкості систем мультихмарного зберігання. З розвитком квантових обчислень, системи мультихмарного зберігання починають включати квантово-стійкі криптографічні алгоритми для забезпечення довгострокової безпеки даних [41].

Математичне моделювання цих передових систем мультихмарного зберігання все більше включає елементи теорії ігор, навчання з підкріпленням та теорії складних систем. Наприклад, динамічний розподіл даних між хмарними провайдерами можна моделювати як марківський процес прийняття рішень (MDP):

Нехай S – набір можливих станів системи мультихмарного зберігання, де кожен стан відображає поточний розподіл даних між хмарними провайдерами. Нехай A – набір можливих дій, таких як міграція даних між провайдерами або коригування рівнів реплікації. Функція переходу стану $P(s'|s, a)$ дає ймовірність переходу до стану s' , коли дія a виконується в стані s . Функція винагороди $R(s, a)$ представляє безпосередню винагороду (або вартість) виконання дії a в стані s .

Метою є знаходження політики $\pi: S \rightarrow A$, яка максимізує очікувану кумулятивну винагороду:

$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s\right], \quad (1.9)$$

де $\gamma \in [0,1)$ – коефіцієнт дисконтування, який балансує безпосередні та майбутні винагороди.

Це формулювання MDP відображає динамічний та невизначений характер середовищ мультихмарного зберігання, де рішення повинні прийматися в умовах змінних витрат, характеристик продуктивності та шаблонів робочого навантаження.

1.3 Значення ефективного зберігання та доступу до даних

Ефективне зберігання та доступ до даних у мультихмарних середовищах представляють критичний імператив для організацій, що орієнтуються у складнощах сучасних цифрових екосистем. Цей розділ досліджує багатогранне значення ефективності в управлінні мультихмарними даними, вивчаючи її наслідки для операційної продуктивності, економічної життєздатності, відповідності нормативним вимогам та стратегічній гнучкості. Розуміючи глибоке значення ефективного зберігання та доступу до даних, встановлюється переконливе обґрунтування для досліджень та рішень, представлених у наступних розділах.

Економічний вимір ефективного зберігання та доступу до даних у мультихмарних середовищах охоплює прямі аспекти щодо витрат, а також ширші фінансові наслідки, пов'язані з бізнес-операціями та конкурентним позиціонуванням.

Структура витрат мультихмарного зберігання є за своєю суттю складною, що складається з кількох компонентів, які варіюються між провайдерами та рівнями сервісу. Прямі витрати, пов'язані зі зберіганням даних, зазвичай стягуються на основі гігабайт-на-місяць. Ці витрати значно варіюються між рівнями зберігання (наприклад, гарячий, теплий, холодний, архівний) та хмарними провайдерами. Плата, що виникає, коли дані переміщуються в, з або між хмарними середовищами, є особливо значною в мультихмарних сценаріях, де дані можуть потребувати передачі між різними провайдерами. Плата, пов'язана з викликами API, отриманням даних та іншими операціями, що виконуються над збереженими даними, також варіюється між провайдерами. Непрямі витрати, пов'язані з адмініструванням та моніторингом мультихмарних середовищ зберігання, включаючи витрати на персонал, навчання та інструменти, також є важливим компонентом загальної вартості.

Математичне представлення загальної вартості володіння (ТСО) для мультимарного зберігання можна сформулювати як:

$$TCO = \sum_{i=1}^n \sum_{j=1}^m (S_{ij} \cdot C_{ij}^S + T_{ij} \cdot C_{ij}^T + O_{ij} \cdot C_{ij}^O) + M, \quad (1.10)$$

де n – кількість категорій даних, m – кількість хмарних провайдерів, S_{ij} – обсяг зберігання для категорії даних i на провайдері j , C_{ij}^S – вартість одиниці зберігання для категорії даних i на провайдері j , T_{ij} – обсяг передачі даних для категорії даних i на провайдері j , C_{ij}^T – вартість одиниці передачі даних для категорії даних i на провайдері j , O_{ij} – кількість операцій для категорії даних i на провайдері j , C_{ij}^O – вартість одиниці операції для категорії даних i на провайдері j , M – вартість накладних витрат на управління

Ефективні механізми зберігання та доступу до даних можуть значно зменшити цю ТСО через кілька механізмів. Інтелектуальний розподіл даних між хмарними провайдерами та рівнями зберігання на основі шаблонів доступу, структур витрат та вимог до продуктивності дозволяє оптимізувати використання ресурсів. Використання передових технік, таких як надлишкове кодування або дедуплікація, для мінімізації накладних витрат на зберігання при збереженні надійності зменшує загальні витрати на зберігання. Стратегічне розміщення даних для зменшення потреби в міжхмарних передачах даних, які часто є найдорожчим компонентом витрат на мультимарне зберігання, також сприяє економії. Використання автоматизації та інтелектуальних алгоритмів для зменшення накладних витрат на управління, пов'язаних з мультимарними середовищами зберігання, додатково оптимізує витрати.

Потенційна економія витрат від ефективного мультимарного зберігання може бути суттєвою. Наприклад, розглянемо сценарій, де організація зберігає 1 петабайт даних на трьох хмарних провайдерах. З наївним підходом, який просто реплікує всі дані на всіх провайдерах, вимога до зберігання становила б 3 петабайти. Натомість, ефективний підхід з використанням надлишкового кодування зі схемою 2-3-3 вимагав би лише 1,5 петабайти зберігання, що призводить до 50% зменшення витрат на зберігання.

Економічне значення ефективного мультимарного зберігання виходить за межі прямої економії витрат, охоплюючи ширшу бізнес-цінність. Ефективні механізми доступу до даних дозволяють швидші цикли розробки та зменшений час виходу на ринок для продуктів та послуг, що базуються на даних. Оптимізовані шаблони доступу до даних можуть зменшити затримку та покращити відгук додатків, орієнтованих на клієнтів, що призводить до покращення задоволеності та

утримання клієнтів. Ефективне мультихмарне зберігання полегшує інтеграцію та аналіз різноманітних джерел даних, що дозволяє нові можливості монетизації даних. Ефективні архітектури мультихмарного зберігання можуть зменшити фінансові ризики, пов'язані з прив'язкою до постачальника, збоями в обслуговуванні та порушеннями даних.

Економічну цінність цих переваг можна кількісно визначити, використовуючи різні підходи, такі як:

$$EV = \sum_{t=1}^T \frac{(R_t - C_t)}{(1 + r)^t}, \quad (1.11)$$

де EV – економічна цінність, T – часовий горизонт, R_t – дохід або економія витрат у періоді t , C_t – витрати в періоді t , r – ставка дисконтування

Це формулювання враховує часову вартість грошей і дозволяє комплексно оцінити економічне значення ефективного мультихмарного зберігання за межами безпосередніх міркувань щодо витрат.

Характеристики продуктивності систем мультихмарного зберігання мають глибокі наслідки для поведінки додатків, досвіду користувачів та загальних можливостей системи. Зокрема, CSA STAR рекомендує впровадження механізмів моніторингу та оптимізації продуктивності для забезпечення стабільної швидкості доступу до даних. Ефективні механізми зберігання та доступу до даних є необхідними для досягнення оптимальної продуктивності в цих складних середовищах.

Продуктивність систем мультихмарного зберігання можна оцінити за кількома вимірами. Затримка представляє часову затримку між ініціюванням запиту на дані та отриманням відповіді. У мультихмарних середовищах на затримку впливають такі фактори, як географічна відстань, стан мережі та конкретні характеристики продуктивності кожного хмарного провайдера. Пропускна здатність визначає швидкість, з якою дані можуть бути оброблені, зазвичай вимірюється в операціях на секунду або байтах на секунду. Мультихмарні середовища повинні балансувати пропускну здатність між кількома провайдерами з різними можливостями. Узгодженість відображає ступінь, до якого всі клієнти бачать одні й ті самі дані одночасно. У мультихмарних середовищах підтримка узгодженості між розподіленими сховищами даних представляє значні виклики. Доступність визначає частку часу, коли система зберігання є операційною та доступною. Мультихмарні архітектури можуть підвищити доступність через надмірність між провайдерами, але лише якщо вони правильно спроектовані. Довговічність представляє ймовірність того, що дані не будуть втрачені з часом.

Мультихмарне зберігання може покращити довговічність через географічний розподіл, але вимагає ефективних механізмів для реплікації та відновлення даних.

Математичне представлення цих метрик продуктивності в мультихмарному контексті можна сформулювати наступним чином:

Для затримки в мультихмарному середовищі з m хмарними провайдерами, ефективна затримка L_{eff} для операції читання залежить від стратегії розміщення даних. Для реплікованих даних з паралельними читаннями:

$$L_{eff} = \min_{j \in CP_i} L_j, \quad (1.12)$$

де CP_i — набір провайдерів, що зберігають копію елемента даних i , а L_j — затримка провайдера j .

Для даних з надлишковим кодуванням з кодуванням k -з- n :

$$L_{eff} = \max_{j \in S} L_j, \quad (1.13)$$

де S — набір з k провайдерів, з яких отримуються дані, вибраних для мінімізації максимальної затримки.

Для пропускної здатності в мультихмарному середовищі, сукупну пропускну здатність T_{agg} можна моделювати як:

$$T_{agg} = \sum_{j=1}^m w_j \cdot T_j, \quad (1.14)$$

де w_j — частка операцій, спрямованих до провайдера j , а T_j — пропускна здатність провайдера j .

Ефективні механізми зберігання та доступу до даних можуть оптимізувати ці метрики продуктивності через кілька підходів. Розміщення даних з урахуванням локальності передбачає розміщення даних близько до місця, де до них найчастіше звертаються, для мінімізації затримки. Адаптивна маршрутизація запитів дозволяє динамічно маршрутизувати запити до найбільш відповідального хмарного провайдера на основі поточних умов продуктивності. Впровадження інтелектуальних механізмів кешування зменшує затримку для часто доступних даних. Розподіл запитів між хмарними провайдерами максимізує сукупну пропускну здатність та уникає вузьких місць. Використання відповідних моделей

узгодженості та механізмів синхронізації на основі вимог додатків забезпечує оптимальний баланс між узгодженістю та продуктивністю.

Оптимізацію продуктивності в мультимарних середовищах зберігання можна сформулювати як задачу багатоцільової оптимізації:

$$\min_x F(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T, \quad (1.15)$$

за умови:

$$\begin{aligned} g_i(x) &\leq 0, i = 1, 2, \dots, m. \\ h_j(x) &= 0, j = 1, 2, \dots, p, \end{aligned}$$

де x представляє змінні рішення (наприклад, розміщення даних, стратегія реплікації, політика кешування), $F(x)$ – вектор цільових функцій, що представляють різні метрики продуктивності, $g_i(x)$ та $h_j(x)$ представляють обмеження (наприклад, ємність зберігання, бюджетні обмеження)

Це формулювання визнає невід'ємні компроміси між різними цілями продуктивності в мультимарних середовищах. Наприклад, оптимізація для затримки може вимагати реплікації даних на кількох географічних регіонах, що може збільшити витрати на зберігання та складність.

Ефективні системи мультимарного зберігання використовують складні алгоритми для навігації цими компромісами та досягнення оптимального балансу на основі конкретних вимог та обмежень додатків. Ці алгоритми можуть використовувати техніки з дослідження операцій, машинного навчання та теорії управління для адаптації до змінних умов та шаблонів робочого навантаження.

Розподілений характер мультимарних середовищ вводить як можливості, так і виклики для надійності та стійкості системи. Ефективні механізми зберігання та доступу до даних є вирішальними для використання потенційних переваг надійності мультимарних архітектур при одночасному пом'якшенні їх невід'ємної складності.

Надійність системи мультимарного зберігання можна кількісно визначити з точки зору різних метрик. Середній час між відмовами (MTBF) представляє середній час між відмовами системи. Середній час до відновлення (MTTR) визначає середній час, необхідний для відновлення функціональності системи після відмови. Доступність відображає частку часу, коли система є операційною, часто виражається як кількість "дев'яток" (наприклад, 99,999% доступності). Довговічність даних представляє ймовірність того, що дані не будуть втрачені протягом визначеного періоду часу [43].

У мультихмарному середовищі характеристики надійності загальної системи залежать від того, як дані розподілені між хмарними провайдерами. Для системи з m хмарними провайдерами, кожен з незалежною доступністю A_j , значення якої не перевищує 1 (наприклад, 0.99), доступність системи A_{sys} можна моделювати наступним чином:

Для системи, що вимагає доступності всіх провайдерів:

$$A_{sys} = \prod_{j=1}^m A_j. \quad (1.16)$$

Для системи, яка може працювати з будь-яким одним провайдером:

$$A_{sys} = 1 - \prod_{j=1}^m (1 - A_j). \quad (1.17)$$

Для системи, що використовує надлишкове кодування з кодуванням k -з- n на m провайдерах:

$$A_{sys} = \sum_{i=k}^m \binom{m}{i} A^i (1 - A)^{m-i}. \quad (1.18)$$

припускаючи, що всі провайдери мають однакову доступність A .

Ефективні архітектури мультихмарного зберігання використовують ці моделі надійності для проектування оптимальних стратегій розподілу даних, які максимізують надійність при мінімізації вартості та складності.

За межами основних міркувань щодо надійності, ефективні системи мультихмарного зберігання впроваджують різні механізми стійкості для обробки відмов, порушень та інших несприятливих подій. Системи моніторингу можуть швидко ідентифікувати відмови або деградацію продуктивності на хмарних провайдерах. Механізми для перенаправлення запитів до альтернативних хмарних провайдерів при виявленні відмови забезпечують безперервність обслуговування. Процеси для відновлення втрачених або пошкоджених даних з використанням надлишкової інформації, що зберігається на хмарних провайдерах, підтримують цілісність даних. Комплексні стратегії для відновлення після катастрофічних подій, що впливають на кількох хмарних провайдерів або регіонів, забезпечують стійкість до масштабних збоїв [46].

Ефективність цих механізмів стійкості залежить від ефективності базової архітектури зберігання та доступу до даних. Наприклад, час, необхідний для відновлення даних після відмови (T_{recon}), можна моделювати як:

$$T_{recon} = \frac{D}{B} + L_{coord}, \quad (1.19)$$

де D – кількість даних, які потрібно відновити, B – доступна пропускна здатність для відновлення, L_{coord} – затримка, пов'язана з координацією між хмарними провайдерами

Ефективні системи мультихмарного зберігання мінімізують T_{recon} через оптимізоване розміщення даних, ефективні алгоритми відновлення та оптимізовані механізми координації.

Здатність масштабувати ресурси у відповідь на змінні вимоги та адаптуватися до еволюціонуючих вимог є фундаментальною перевагою хмарних обчислень. У мультихмарних середовищах досягнення масштабованості та гнучкості вимагає ефективних механізмів зберігання та доступу до даних, які можуть пристосовуватися до зростання та змін без компромісу щодо продуктивності або надійності.

Масштабованість у мультихмарному зберіганні охоплює кілька вимірів. Масштабованість ємності забезпечує здатність пристосовуватися до зростаючих обсягів даних шляхом додавання ресурсів зберігання на хмарних провайдерах. Масштабованість продуктивності дозволяє підтримувати або покращувати метрики продуктивності при зростанні системи з точки зору обсягу даних, швидкості запитів або бази користувачів. Географічна масштабованість забезпечує здатність розширювати географічний слід системи на нові географічні регіони для підтримки глобальних операцій або відповідності вимогам суверенітету даних. Адміністративна масштабованість дозволяє управляти все більш складним мультихмарним середовищем без пропорційного збільшення адміністративних накладних витрат.

Математичне моделювання масштабованості в мультихмарних середовищах часто використовує концепції з теорії черг та моделювання продуктивності. Наприклад, час відгуку R системи мультихмарного зберігання при зростаючому навантаженні можна моделювати за допомогою моделі черги М/М/с:

$$R = \frac{1}{\mu} + \frac{C(m, \rho)}{m\mu - \lambda}, \quad (1.20)$$

де μ – швидкість обслуговування кожного хмарного провайдера, m – кількість хмарних провайдерів, λ – швидкість прибуття запитів, $\rho = \lambda/(m\mu)$ – використання системи, $C(m, \rho)$ – формула Ерланга С, що представляє ймовірність чергування.

Ефективні архітектури мультихмарного зберігання підтримують прийнятний час відгуку при зростаючому навантаженні шляхом динамічного коригування розподілу ресурсів, маршрутизації запитів та стратегій розміщення даних.

Гнучкість систем мультихмарного зберігання відноситься до їх здатності адаптуватися до змінних вимог, технологій та ринкових умов. Ключові аспекти гнучкості включають гнучкість провайдера, яка забезпечує здатність додавати, видаляти або замінювати хмарних провайдерів з мінімальним порушенням; гнучкість сервісної моделі, яка дозволяє використовувати різні моделі сервісів зберігання на хмарних провайдерах; гнучкість моделі ціноутворення, яка оптимізує витрати шляхом використання різних моделей ціноутворення та промоційних пропозицій від хмарних провайдерів; та гнучкість відповідності, яка забезпечує адаптацію до змінних нормативних вимог шляхом коригування механізмів розміщення та захисту даних. [43, 45]

Цінність гнучкості в мультихмарному зберіганні можна кількісно визначити за допомогою теорії реальних опціонів, яка надає рамки для оцінки гнучкості в невизначених середовищах:

$$V_{flex} = V_{static} + \sum_{i=1}^n O_i , \quad (1.21)$$

де V_{flex} – цінність гнучкої системи мультихмарного зберігання, V_{static} – цінність статичної системи без гнучкості, O_i – цінність опціону i (наприклад, опціон на зміну провайдерів, масштабування ємності або зміну сервісних моделей)

Ефективні архітектури мультихмарного зберігання максимізують V_{flex} шляхом впровадження механізмів, що полегшують адаптацію з мінімальними витратами та порушеннями.

Ландшафт безпеки та відповідності для мультихмарного зберігання характеризується складними вимогами, різноманітними моделями загроз та еволюціонуючими нормативними рамками. Ефективні механізми зберігання та доступу до даних є необхідними для вирішення цих викликів при збереженні продуктивності та економічної ефективності [44].

Середовища мультихмарного зберігання стикаються з кількома унікальними викликами безпеки. Відповідно до ISO/IEC 27017, надійність зберігання даних у хмарному середовищі повинна забезпечуватися комплексом технічних та

організаційних заходів, включаючи резервне копіювання, реплікацію даних та механізми відновлення після збоїв. У контексті мультихмарного середовища надійність зберігання набуває додаткової складності через необхідність координації цих механізмів між різними хмарними провайдерами. Використання кількох хмарних провайдерів збільшує кількість потенційних точок входу для атакуючих, розширюючи поверхню атаки. Різні хмарні провайдери впроваджують контроль безпеки по-різному, створюючи потенційні прогалини або непослідовності в захисті. Дані, що переміщуються між хмарними провайдерами, піддаються додатковим ризикам порівняно з даними, що залишаються в мережі одного провайдера. Управління ідентифікаціями та правами доступу на кількох хмарних середовищах вводить складність та потенційні вразливості безпеки. Різні хмарні провайдери мають різні інтерпретації моделі спільної відповідальності, створюючи потенційну плутанину щодо обов'язків безпеки.

Ефективні архітектури мультихмарного зберігання вирішують ці виклики через інтегровані механізми безпеки, які забезпечують послідовний захист на всіх хмарних провайдерах. Позицію безпеки системи мультихмарного зберігання можна моделювати, використовуючи підхід, заснований на ризику:

$$R_{total} = \sum_{i=1}^n \sum_{j=1}^m P_{ij} \cdot I_{ij} \cdot (1 - E_{ij}), \quad (1.22)$$

де R_{total} – загальний ризик, P_{ij} – ймовірність виникнення загрози i на хмарному провайдері j , I_{ij} – вплив загрози i на хмарному провайдері j , E_{ij} – ефективність контролів проти загрози i на хмарному провайдері j .

Ефективні системи мультихмарного зберігання мінімізують R_{total} через комбінацію превентивних, детективних та коригувальних контролів, впроваджених послідовно на всіх хмарних провайдерах.

Середовища мультихмарного зберігання повинні орієнтуватися в складному ландшафті нормативних вимог. Регуляції захисту даних, такі як Загальний регламент про захист даних (GDPR) в Європейському Союзі, накладають суворі вимоги на обробку та зберігання персональних даних [48,49]. Галузеві регуляції, такі як Закон про портативність і підзвітність медичного страхування (HIPAA) для медичних даних або Стандарт безпеки даних індустрії платіжних карток (PCI DSS) для даних платіжних карток, встановлюють специфічні вимоги для конкретних типів даних. Закони про суверенітет даних обмежують географічні місця, де певні типи даних можуть зберігатися або оброблятися. Вимоги до зберігання даних

визначають, як довго певні типи даних повинні зберігатися і коли вони повинні бути видалені.

Відповідність у мультимарних середовищах можна моделювати як задачу задоволення обмежень:

$$\forall r \in R, \forall d \in D_r: C_r(d) = \text{true} , \quad (1.23)$$

де R – набір застосовних регуляцій, D_r – набір даних, що підлягають регуляції r , $C_r(d)$ – булева функція, що вказує, чи дані d відповідають регуляції r

Ефективні архітектури мультимарного зберігання впроваджують механізми для автоматичного забезпечення вимог відповідності на всіх хмарних провайдерах. Автоматична категоризація даних на основі чутливості та нормативних вимог дозволяє застосовувати відповідні політики захисту. Забезпечення того, що дані зберігаються лише у відповідних географічних місцях, допомагає дотримуватися вимог суверенітету даних. Забезпечення політик зберігання даних послідовно на всіх хмарних провайдерах гарантує відповідність вимогам до зберігання. Підтримка комплексних журналів усіх дій з доступу та маніпуляції даними забезпечує необхідну інформацію для звітності про відповідність [52].

За межами технічних та операційних міркувань, ефективне зберігання та доступ до даних у мультимарних середовищах мають глибокі стратегічні наслідки для організацій. Ці стратегічні виміри підкреслюють ширше значення досліджень, представлених у цій дисертації [53,54].

Організації, які досягають ефективного зберігання та доступу до даних у мультимарних середовищах, можуть отримати значні конкурентні переваги. Зменшення витрат на зберігання та передачу даних через оптимізовані мультимарні архітектури сприяє лідерству за витратами. Пропонування вищої продуктивності, надійності та географічного покриття шляхом використання сильних сторін кількох хмарних провайдерів забезпечує диференціацію послуг. Прискорення розробки та розгортання інновацій, заснованих на даних, шляхом надання ефективного доступу до різноманітних джерел даних підвищує гнучкість інновацій [55,57,58]. Зменшення бізнес-ризиків, пов'язаних з прив'язкою до постачальника, збоями в обслуговуванні та невідповідністю, покращує загальну стійкість бізнесу.

Конкурентну перевагу, отриману від ефективного мультимарного зберігання, можна кількісно визначити, використовуючи різні стратегічні рамки, такі як ресурсно-орієнтований погляд на фірму, який стверджує, що конкурентна перевага виникає з ресурсів, які є цінними, рідкісними, неповторними та незамінними (VRIN).

Ефективне мультихмарне зберігання служить фундаментальним забезпеченням для ширших ініціатив цифрової трансформації. Надання своєчасного та надійного доступу до різноманітних джерел даних для аналітики та підтримки прийняття рішень сприяє прийняттю рішень на основі даних. Підтримка персоналізованих, відповідальних взаємодій з клієнтами через кілька каналів та точок дотику покращує досвід клієнтів. Забезпечення моніторингу, аналізу та оптимізації бізнес-процесів у реальному часі підвищує операційну досконалість. Полегшення розробки нових продуктів, послуг та бізнес-моделей, заснованих на даних, стимулює інновації бізнес-моделі.

Стратегічну цінність ефективного мультихмарного зберігання у забезпеченні цифрової трансформації можна оцінити, використовуючи концепцію цифрових опціонів, які представляють здатність організації використовувати цифрові технології для стратегічних цілей:

$$V_{digital} = \sum_{i=1}^n p_i \cdot v_i \cdot (1 - e^{-rt_i}), \quad (1.24)$$

де $V_{digital}$ – цінність цифрових опціонів, p_i – ймовірність успішного виконання опціону i , v_i – цінність, створена виконанням опціону i , r – ставка дисконтування, t_i – час до можливості виконання опціону i .

Ефективне мультихмарне зберігання збільшує $V_{digital}$ шляхом підвищення здатності організації швидко та ефективно виконувати цифрові опціони.

ВИСНОВКИ

1. Аналіз специфікацій та рекомендацій Міжнародного союзу електрозв'язку (ITU), Європейського інституту телекомунікаційних стандартів (ETSI) та Національного інституту стандартів і технологій США (NIST) дозволив визначити основні вимоги до систем розподіленого зберігання даних у мультихмарному середовищі. Встановлено, що сучасні мультихмарні архітектури характеризуються високим рівнем гетерогенності, розподіленості та необхідністю забезпечення безперервного доступу до даних.

2. Аналіз стандартів ISO/IEC 27017, ISO/IEC 27018 та CSA STAR дозволив виділити ключові показники ефективності процесу зберігання даних: надійність зберігання, швидкість доступу, вартість зберігання, рівень безпеки та відповідність регуляторним вимогам.

3. Дослідження еволюції хмарних сховищ від одно-хмарних до мультихмарних архітектур виявило зростаючу потребу в розробці нових підходів до організації розподіленого зберігання, що враховують особливості взаємодії між різними хмарними провайдерами та забезпечують оптимальне використання їхніх ресурсів.

4. Проведено огляд існуючих методів підвищення ефективності зберігання та доступу до даних у мультихмарному середовищі та проаналізовано їх ефективність щодо визначених показників. Виявлено, що більшість існуючих методів не забезпечують комплексного підходу до оптимізації всіх ключових показників ефективності одночасно.

5. Основним недоліком існуючих підходів є недостатнє врахування динамічної природи мультихмарного середовища, що призводить до неоптимального розподілу даних між хмарними провайдерами та зниження загальної ефективності системи.

6. Визначено необхідність розробки нового підходу до організації розподіленого зберігання та доступу до даних у мультихмарному середовищі, який би враховував усі виділені показники ефективності та забезпечував адаптивність до змін у характеристиках хмарних провайдерів, а також до змін у вимогах зберігання даних.

РОЗДІЛ 2

МЕТОДИ ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ У МУЛЬТИХМАРНОМУ СЕРЕДОВИЩІ

2.1 Постановка задачі

У дослідженні розглядається складна задача оптимізації зберігання та доступу до даних у мультихмарних середовищах. Оскільки організації все частіше впроваджують розподілені хмарні архітектури, потреба в систематичних підходах до розміщення, реплікації та отримання даних стає першочерговою. Фундаментальну дослідницьку задачу можна формалізувати наступним чином:

нехай маючи набір об'єктів даних $D = d_1, d_2, \dots, d_n$ з різними характеристиками та вимогами, та набір постачальників хмарних сховищ $CP = cp_1, cp_2, \dots, cp_m$ з гетерогенними можливостями та обмеженнями, визначити оптимальну функцію відображення $f: D \rightarrow P(CP)$, яка призначає кожен об'єкт даних одному або кільком хмарним провайдерам, задовольняючи комплексний набір критеріїв та оптимізуючи загальносистемні цілі.

Функція відображення f повинна враховувати багатовимірну природу задачі оптимізації, де кожен вимір представляє критерій, що впливає на рішення щодо розміщення даних. Простір рішень обмежений технічними обмеженнями, бізнес-вимогами та нормативними рамками, які регулюють практики управління даними в різних юрисдикціях.

Ця формалізація охоплює багатоцільову природу проблеми, де організації повинні балансувати між конкуруючими пріоритетами, такими як економічна ефективність, вимоги до продуктивності, суверенітет даних та зниження ризиків. Складність додатково посилюється динамічним характером як характеристик даних, так і пропозицій хмарних провайдерів, що вимагає адаптивних стратегій, які можуть реагувати на зміни умов.

Для комплексного вирішення цієї задачі визнається 27 окремих критеріїв, які впливають на рішення щодо розміщення даних у мультихмарних середовищах. Ці критерії охоплюють технічні, економічні, регуляторні та екологічні виміри, забезпечуючи цілісну структуру для оцінки потенційних конфігурацій зберігання даних. [62, 65] Критерії є не просто теоретичними конструкціями, а представляють практичні чинники, які організації повинні враховувати при впровадженні стратегій управління даними в мультихмарному середовищі.

Далі в цьому розділі надаються детальні визначення та математичні вирази для кожного критерію, розробляється онтологічна модель, яка фіксує взаємозв'язки між цими критеріями та об'єктами, залученими до мультихмарного зберігання, та формулюється математична модель оптимізації, яка дозволяє систематичне прийняття рішень щодо розміщення даних серед хмарних провайдерів.

Кінцевою метою цієї формалізації є перетворення складної, багатогранної проблеми оптимізації мультихмарного зберігання даних у структуровану систему, яка дозволяє кількісний аналіз та алгоритмічні рішення, одночасно визнаючи якісні фактори, що впливають на реальну імплементацію.

2.2 Комплексний набір критеріїв для мультихмарного зберігання даних

У цьому розділі представлено комплексний набір критеріїв для мультихмарного зберігання даних, отриманий на основі широкого огляду літератури та аналізу сучасних стандартів хмарних обчислень. Ці критерії служать основою для розробки ефективних стратегій розподіленого зберігання та доступу до даних у кількох хмарних провайдерів. Кожен критерій визначено, детально описано та супроводжено відповідними одиницями вимірювання та математичними виразами, де це доречно.

2.2.1 Критерії, пов'язані з продуктивністю

Критерії продуктивності є фундаментальними для оцінки ефективності та результативності рішень мультихмарного зберігання. Ці критерії зосереджені на швидкості, відгуку та пропускній здатності систем зберігання в розподілених середовищах.

Вимоги до затримки

Затримка в контексті мультихмарного зберігання відноситься до часової затримки між ініціюванням запиту до даних дані та отриманням відповіді. Це критичний показник продуктивності, який безпосередньо впливає на досвід користувача та продуктивність додатків.

Визначення: Затримка – це часовий інтервал між ініціюванням запиту до даних та отриманням відповідної відповіді, виміряний у мультихмарному середовищі.

Одиниці вимірювання: Затримка зазвичай вимірюється в мілісекундах (мс) для більшості додатків, хоча мікросекунди (мкс) можуть використовуватися для сценаріїв високопродуктивних обчислень.

Математичний вираз: для мультихмарного середовища з n хмарними провайдерами, наскрізна затримка L для запиту даних може бути виражена як:

$$L = L_{client} + \sum_{i=1}^n (L_{network_i} + L_{processing_i}) + L_{response}, \quad (2.1)$$

де L_{client} представляє початковий час обробки на стороні клієнта, $L_{network_i}$ представляє час передачі по мережі до i від хмарного провайдера i , $L_{processing_i}$ представляє час обробки у хмарного провайдера i , $L_{response}$ представляє час доставки кінцевої відповіді клієнту [68].

Для додатків з конкретними вимогами до затримки можна визначити поріг $L_{threshold}$, який має бути задоволений: $L \leq L_{threshold}$

Показники продуктивності

Визначення: Показники продуктивності – це кількісні міри, що оцінюють ефективність, швидкість та надійність операцій з даними у мультихмарному середовищі.

Одиниці вимірювання:

- пропускна здатність: Вимірюється в мегабайтах за секунду (МБ/с) або гігабайтах за секунду (ГБ/с);
- IOPS: Вимірюється як операції за секунду;
- пропускна здатність мережі: Вимірюється в мегабітах за секунду (Мбіт/с) або гігабітах за секунду (Гбіт/с);
- час відгуку: Вимірюється в мілісекундах (мс).

Математичний вираз: Загальна продуктивність P мультихмарної системи зберігання може бути змодельована як функція кількох показників продуктивності:

$$P = f(T, I, B, R), \quad (2.2)$$

де T представляє пропускну здатність, I представляє IOPS, B представляє пропускну здатність мережі, R представляє час відгуку.

Для конкретних додатків можна визначити зважені показники продуктивності:

$$P_w = w_T \cdot T + w_I \cdot I + w_B \cdot B + w_R \cdot \frac{1}{R}, \quad (2.3)$$

де w_T , w_I , w_B та w_R – це ваги, призначені на основі відносної важливості кожного показника для конкретного додатку.

Масштабованість

Визначення: Масштабованість – це здатність мультихмарної системи зберігання підтримувати або покращувати рівні продуктивності при збільшенні робочого навантаження або обсягу даних.

Одиниці вимірювання: Масштабованість можна вимірювати за допомогою кількох показників:

- коефіцієнт масштабування: Співвідношення продуктивності при збільшеному навантаженні до продуктивності при базовому навантаженні;
- ефективність ресурсів: Приріст продуктивності на одиницю додаткових ресурсів;
- час відгуку на масштабування: Час, необхідний для масштабування ресурсів у відповідь на збільшення попиту.

Математичний вираз: Масштабованість S мультихмарної системи зберігання можна виразити як:

$$S = \frac{P(n \cdot L)}{n \cdot P(L)}, \quad (2.4)$$

де $P(L)$ – це продуктивність при навантаженні L , $P(n \cdot L)$ – це продуктивність при n -кратному навантаженні, n – це коефіцієнт масштабування.

Ідеальна лінійна масштабованість призвела б до $S = 1$, що вказує на те, що продуктивність масштабується пропорційно ресурсам. На практиці S часто менше 1 через накладні витрати та конкуренцію за ресурси.

Для мультихмарних середовищ можна визначити специфічні для провайдера фактори масштабованості:

$$S_i = \frac{P_i(n \cdot L_i)}{n \cdot P_i(L_i)}, \quad (2.5)$$

де S_i та P_i представляють масштабованість та продуктивність для хмарного провайдера i .

2.2.2 Критерії надійності та доступності

Критерії надійності та доступності зосереджені на здатності системи підтримувати безперервну роботу та доступність даних, навіть у разі збоїв або порушень. Ці критерії особливо важливі в мультихмарних середовищах, де складність розподілених систем вводить додаткові точки потенційного збою.

Надлишковість та доступність

Надлишковість відноситься до дублювання критичних компонентів або функцій системи для підвищення надійності, тоді як доступність вимірює частку часу, коли система є працездатною та доступною.

Визначення: Надлишковість у мультихмарному зберіганні – це стратегічна реплікація даних у кількох хмарних провайдерів для забезпечення безперервного доступу у випадку збоїв конкретного провайдера. Доступність – це відсоток часу, коли дані доступні при запиті.

Одиниці вимірювання:

- доступність: вимірюється у відсотках, часто виражається в термінах "дев'яток" (наприклад, "п'ять дев'яток" або 99,999% доступності);
- коефіцієнт надлишковості: Співвідношення загального використаного сховища до розміру оригінальних даних.

Математичний вираз: Доступність A мультихмарної системи зберігання з n незалежними хмарними провайдерами, кожен з доступністю A_i (з розмірністю від 0 до 1), можна обчислити як:

$$A = 1 - \prod_{i=1}^n (1 - A_i). \quad (2.6)$$

Ця формула припускає, що дані повністю репліковані у всіх провайдерів і доступні, якщо хоча б один провайдер доступний.

Для систем, що використовують надлишкове кодування з k фрагментами даних та m надлишковими фрагментами (потрібно будь-які k з $k+m$ фрагментів для відновлення даних), доступність можна виразити як:

$$A = \sum_{i=k}^{k+m} \binom{k+m}{i} A_p^i (1 - A_p)^{k+m-i}, \quad (2.7)$$

де A_p – це середня доступність хмарних провайдерів.

Узгодженість даних

Визначення: Узгодженість даних у мультихмарному зберіганні – це властивість, яка забезпечує, що всі репліки даних у різних хмарних провайдерів відображають однаковий стан та значення в заданий момент часу або після серії операцій.

Одиниці вимірювання:

- рівень узгодженості: Якісна міра (сильна, остаточна, причинна тощо);

- вікно неузгодженості: Період часу, протягом якого репліки можуть бути неузгодженими (вимірюється в секундах);
- застарілість: Вік даних порівняно з найновішою версією (вимірюється в одиницях часу або різницях версій).

Математичний вираз: Для системи з n репліками, рівень узгодженості C можна кількісно визначити на основі кількості реплік r , які повинні підтвердити операцію запису, перш ніж вона вважатиметься завершеною:

$$C = \frac{r}{n}, \quad (2.8)$$

де $C = 1$ представляє сильну узгодженість (всі репліки повинні підтвердити), $C < 1$ представляє різні ступені послабленої узгодженості.

Вікно неузгодженості W для остаточної узгодженості можна моделювати як:

$$W = \max(P_{ij}) + \max(P_{proc}), \quad (2.9)$$

де P_{ij} – це затримка поширення між репліками i та j , P_{proc} – це затримка обробки на кожній репліці.

Ціль точки відновлення (RPO)

Визначення: Ціль точки відновлення (RPO) – це максимально допустимий період часу до збою або інциденту, протягом якого дані можуть бути втрачені і не відновлювані в мультихмарному середовищі.

Одиниці вимірювання: RPO зазвичай вимірюється в одиницях часу: секундах, хвилинах, годинах або днях.

Математичний вираз: Ефективний RPO в мультихмарному середовищі можна обчислити як:

$$RPO_{effective} = \max(T_{backup} + T_{transfer}), \quad (2.10)$$

де T_{backup} – це час між послідовними резервними копіями, $T_{transfer}$ – це час, необхідний для передачі резервних даних між хмарними провайдерами.

Для систем з безперервною реплікацією RPO можна виразити з точки зору затримки реплікації:

$$RPO_{continuous} = \max(L_{replication}), \quad (2.11)$$

де $L_{replication}$ – це максимальний час затримки в процесі реплікації у всіх хмарних провайдерів.

Ціль часу відновлення (RTO)

Визначення: Ціль часу відновлення (RTO) – це максимально допустима тривалість між перериванням обслуговування та відновленням обслуговування в мультихмарному середовищі.

Одиниці вимірювання: RTO вимірюється в одиницях часу: секундах, хвилинах, годинах або днях.

Математичний вираз: Ефективний RTO в мультихмарному середовищі можна обчислити як:

$$RTO_{effective} = T_{detection} + T_{decision} + \max(T_{recovery_i}), \quad (2.12)$$

де $T_{detection}$ – це час для виявлення збою, $T_{decision}$ – це час для прийняття рішень щодо відновлення, $T_{recovery_i}$ – це час відновлення для кожного компонента i в системі

Для систем з автоматичним переключенням між хмарними провайдерами RTO можна виразити як:

$$RTO_{failover} = T_{detection} + T_{failover} + T_{propagation}, \quad (2.13)$$

де $T_{failover}$ – це час для виконання процесу переключення, $T_{propagation}$ – це час для поширення переключення по системі

Частота резервного копіювання даних

Частота резервного копіювання даних визначає, як часто дані резервуються, безпосередньо впливаючи як на RPO, так і на актуальність відновлюваних даних.

Визначення: Частота резервного копіювання даних – це швидкість, з якою дані копіюються та зберігаються у вторинних місцях у кількох хмарних провайдерів для захисту від втрати даних.

Одиниці вимірювання: Частота резервного копіювання зазвичай виражається як кількість резервних копій за період часу (наприклад, щогодини, щодня, щотижня) або часовий інтервал між послідовними резервними копіями.

Математичний вираз: Оптимальну частоту резервного копіювання F можна визначити, збалансувавши вартість потенційної втрати даних з вартістю виконання резервного копіювання:

$$F_{optimal} = \sqrt{\frac{C_{loss} \cdot R_{change}}{C_{backup}}}, \quad (2.14)$$

де C_{loss} – це вартість на одиницю втрачених даних, R_{change} – це швидкість зміни даних, C_{backup} – це вартість на операцію резервного копіювання.

У мультихмарних середовищах можна визначити специфічні для провайдера частоти резервного копіювання:

$$F_i = \frac{1}{T_{backup_i}}, \quad (2.15)$$

де T_{backup_i} – це часовий інтервал між послідовними резервними копіями для хмарного провайдера i .

Надлишковість сховища резервних копій

Визначення: Надлишковість сховища резервних копій — це ступінь, до якого резервні дані репліковані у кількох місцях зберігання, системах або хмарних провайдерах для захисту від збоїв зберігання.

Одиниці вимірювання:

- коефіцієнт надлишковості: Співвідношення загального сховища резервних копій до розміру оригінальних даних;
- географічний розподіл: Кількість різних географічних регіонів, де зберігаються резервні копії;
- різноманітність провайдерів: Кількість незалежних хмарних провайдерів, що зберігають копії резервних копій.

Математичний вираз: Рівень надлишковості резервних копій R можна обчислити як:

$$R = \frac{S_{backup}}{S_{original}}, \quad (2.16)$$

де S_{backup} – це загальне сховище, використане для резервних копій, $S_{original}$ – це розмір оригінальних даних.

Для мультихмарних середовищ з n провайдерами ефективну надлишковість можна виразити як:

$$R_{effective} = \sum_{i=1}^n R_i \cdot P_{survival_i}, \quad (2.17)$$

де R_i – це коефіцієнт надлишковості для провайдера i , $P_{survival_i}$ – це ймовірність того, що сховище провайдера i залишиться доступним під час сценарію катастрофи.

2.2.3 Критерії безпеки та відповідності

Критерії безпеки та відповідності стосуються захисту даних від несанкціонованого доступу, пошкодження або втрати, а також дотримання відповідних нормативних актів та стандартів. Ці критерії особливо важливі в мультихмарних середовищах, де дані проходять через інфраструктури кількох провайдерів.

Шифрування даних

Визначення: Шифрування даних у мультихмарному зберіганні – це процес кодування даних за допомогою криптографічних алгоритмів для захисту їх конфіденційності та цілісності під час зберігання у кількох хмарних провайдерів.

Одиниці вимірювання:

- сила шифрування: Вимірюється довжиною ключа (біти) або обчислювальною складністю алгоритму;
- покриття шифрування: Відсоток даних, які зашифровані;
- частота ротації ключів: Як часто змінюються ключі шифрування.

Математичний вираз: Рівень безпеки S схеми шифрування можна кількісно визначити на основі обчислювальних зусиль, необхідних для її зламу:

$$S = 2^k, \quad (2.18)$$

де k – це ефективна довжина ключа в бітах.

Для мультихмарного середовища з різними вимогами до шифрування загальна сила шифрування може бути виражена як:

$$S_{overall} = \min(S_1, S_2, \dots, S_n), \quad (2.19)$$

де S_i представляє силу шифрування для хмарного провайдера i .

Покриття шифрування C можна обчислити як:

$$C = \frac{D_{encrypted}}{D_{total}}, \quad (2.20)$$

де $D_{encrypted}$ – це кількість зашифрованих даних, D_{total} – це загальна кількість даних.

Протоколи безпеки

Визначення: Протоколи безпеки в мультихмарному зберіганні – це формальні правила та процедури, які регулюють, як заходи безпеки впроваджуються та застосовуються у кількох хмарних провайдерів.

Детальний опис: Протоколи безпеки в мультихмарних середовищах повинні вирішувати питання аутентифікації (перевірка ідентичності), авторизації (визначення прав доступу), обліку (відстеження використання ресурсів) та аудиту (перегляд подій безпеки). Ці протоколи повинні функціонувати послідовно у різних середовищах провайдерів, які можуть реалізовувати безпеку по-різному.

Поширені протоколи безпеки включають OAuth для авторизації, SAML для аутентифікації, TLS/SSL для захищених комунікацій та різні стандарти шифрування. У мультихмарних середовищах організаціям часто потрібно впроваджувати додаткові рівні безпеки для забезпечення послідовного захисту у різних провайдерів, такі як брокери безпеки доступу до хмари (CASB) або системи управління інформацією та подіями безпеки (SIEM).

Одиниці вимірювання:

- сила протоколу: Якісна оцінка на основі відомих вразливостей та прийняття в галузі;
- повнота впровадження: Відсоток заходів безпеки, впроваджених відповідно до специфікацій протоколу;
- рівень відповідності: Ступінь дотримання необхідних стандартів безпеки.

Математичний вираз: Ефективність протоколу безпеки E можна моделювати як функцію кількох факторів:

$$E = f(S, I, C, U), \quad (2.21)$$

де S представляє силу протоколу, I представляє повноту впровадження, C представляє рівень відповідності, U представляє зручність використання (легкість правильного впровадження)

Для мультихмарних середовищ загальна позиція безпеки P може бути виражена як:

$$P = \min(E_1, E_2, \dots, E_n), \quad (2.22)$$

де E_i представляє ефективність протоколу безпеки для хмарного провайдера i .

Відповідність нормативним вимогам (Compliance)

Визначення: Відповідність нормативним вимогам у мультихмарному зберіганні (далі відповідність) — це стан відповідності відповідним законам, нормативним актам та стандартам, що регулюють захист даних, конфіденційність та безпеку у всіх хмарних провайдерів, які використовує організація.

Одиниці вимірювання:

- Покриття відповідності: Відсоток застосовних нормативних актів, яким відповідає система
- Розрив відповідності: Кількість невирішених вимог відповідності
- Готовність до аудиту: Якісна міра готовності до аудитів відповідності

Математичний вираз: Рівень відповідності C для конкретного нормативного акту можна обчислити як:

$$C_r = \frac{R_{met}}{R_{total}}, \quad (2.23)$$

де R_{met} — це кількість виконаних вимог, R_{total} — це загальна кількість вимог.

Для кількох нормативних актів загальний рівень відповідності можна виразити як зважене середнє:

$$C_{overall} = \sum_{r=1}^m w_r \cdot C_r, \quad (2.24)$$

де m — це кількість застосовних нормативних актів, w_r — це вага, призначена нормативному акту r на основі його важливості або потенційних штрафів.

Власність даних

Визначення: Власність даних у мультихмарному зберіганні відноситься до юридичних прав та контролю над даними, що зберігаються у кількох хмарних провайдерів, включаючи можливість доступу, модифікації, видалення та визначення політик використання цих даних.

Одиниці вимірювання:

- ясність власності: Якісна оцінка того, наскільки чітко визначені права власності;
- ефективність контролю: Ступінь, до якого технічні заходи дозволяють забезпечення прав власності;

- можливість вилучення: Здатність отримувати дані від хмарних провайдерів у придатних для використання форматах;

Математичний вираз: Ефективний контроль власності O можна моделювати як:

$$O = \min(L, T, E), \quad (2.25)$$

де L представляє юридичні права власності, визначені в контрактах, T представляє механізми технічного контролю, E представляє можливості вилучення.

Для мультихмарних середовищ можна визначити специфічні для провайдера фактори власності:

$$O_i = w_L \cdot L_i + w_T \cdot T_i + w_E \cdot E_i, \quad (2.26)$$

де w_L , w_T та w_E – це ваги, що відображають відносну важливість юридичних, технічних аспектів та аспектів вилучення для провайдера i .

2.2.4 Економічні критерії та критерії управління ресурсами

Економічна ефективність

Визначення: Економічна ефективність у мультихмарному зберіганні – це міра того, наскільки ефективно використовуються фінансові ресурси для задоволення вимог до зберігання у кількох хмарних провайдерів, балансуючи витрати з продуктивністю, надійністю та потребами безпеки.

Одиниці вимірювання:

- загальна вартість володіння (TCO): Вимірюється в одиницях валюти (наприклад, USD) за певний період часу;
- вартість на одиницю зберігання: Одиниці валюти за ГБ на місяць;
- вартість за операцію: Одиниці валюти за API-виклик або транзакцію;
- коефіцієнт економічної ефективності: Продуктивність зберігання або ємність, поділена на вартість;

Математичний вираз: Загальну вартість C рішення мультихмарного зберігання можна обчислити як:

$$C = \sum_{i=1}^n (C_{storage_i} + C_{transfer_i} + C_{operations_i} + C_{services_i}), \quad (2.27)$$

де $C_{storage_i}$ – це вартість зберігання для провайдера i , $C_{transfer_i}$ – це вартість передачі даних для провайдера i , $C_{operations_i}$ – це вартість операцій для провайдера i , $C_{services_i}$ – це вартість додаткових послуг для провайдера i .

Коефіцієнт економічної ефективності E можна виразити як:

$$E = \frac{P}{C}, \quad (2.28)$$

де P представляє показник продуктивності або ємності, C представляє загальну вартість.

Розподіл ресурсів

Визначення: Розподіл ресурсів у мультихмарному зберіганні – це процес розподілу та призначення ресурсів зберігання, обчислення та мережі у кількох хмарних провайдерів для оптимізації продуктивності, надійності та економічної ефективності.

Одиниці вимірювання:

- використання ресурсів: Відсоток виділених ресурсів, що активно використовуються;
- ефективність розподілу: Співвідношення використання ресурсів до вартості ресурсів;
- час забезпечення: Час, необхідний для виділення та налаштування нових ресурсів;
- еластичність ресурсів: Здатність масштабувати ресурси вгору або вниз у відповідь на попит.

Математичний вираз: Проблему розподілу ресурсів у мультихмарному середовищі можна сформулювати як задачу оптимізації:

$$\text{Minimize } C = \sum_{i=1}^n \sum_{j=1}^m c_{ij} \cdot x_{ij}. \quad (2.29)$$

За умови:

$$\sum_{i=1}^n x_{ij} \geq r_j \quad \forall j \in 1, 2, \dots, m \quad \sum_{j=1}^m x_{ij} \leq a_i \quad \forall i \in 1, 2, \dots, n \quad x_{ij} \geq 0 \quad \forall i, j,$$

де c_{ij} – це вартість виділення ресурсів від провайдера i для вимоги j , x_{ij} – це кількість ресурсів, виділених від провайдера i для вимоги j , r_j – це вимога до ресурсів для завдання j , a_i – це доступні ресурси від провайдера i .

Управління життєвим циклом даних

Визначення: Управління життєвим циклом даних у мультимарному зберіганні — це комплексний підхід до управління даними протягом усього їх корисного життя, від створення та початкового зберігання до архівування та остаточного видалення, у кількох хмарних провайдерів.

Одиниці вимірювання:

- покриття політики життєвого циклу: Відсоток даних, керованих визначеними політиками життєвого циклу;
- ефективність переходу: Час та ресурси, необхідні для переміщення даних між етапами життєвого циклу;
- точність збереження: Ступінь, до якого фактичне збереження даних відповідає вимогам політики;
- відповідність видалення: Відсоток даних, видалених відповідно до вимог політики.

Математичний вираз: Стан життєвого циклу S об'єкта даних у час t можна моделювати як:

$$S(t) = f(A(t), V(t), R(t), C(t)), \quad (2.30)$$

де $A(t)$ представляє частоту доступу в час t , $V(t)$ представляє бізнес-цінність в час t , $R(t)$ представляє вимоги до збереження в час t , $C(t)$ представляє вимоги відповідності в час t .

Оптимальний рівень зберігання T для даних у стані S можна визначити за допомогою:

$$T(S) = \arg \min_{i \in 1, 2, \dots, n} (C_i + P_i \cdot A(t)), \quad (2.31)$$

де C_i — це вартість зберігання для рівня i , P_i — це вартість доступу для рівня i .

Класифікація даних

Визначення: Класифікація даних у мультимарному зберіганні — це систематична категоризація даних на основі їх чутливості, нормативних вимог, бізнес-цінності та шаблонів доступу для інформування про відповідне розміщення та захист у кількох хмарних провайдерів.

Одиниці вимірювання:

- покриття класифікації: Відсоток даних, які були класифіковані;
- точність класифікації: Ступінь, до якого дані правильно класифіковані;
- гранулярність класифікації: Рівень деталізації в схемі класифікації;

- рівень автоматизації: Відсоток даних, класифікованих через автоматизовані процеси.

Математичний вираз: Функцію класифікації C для об'єкта даних d можна виразити як:

$$C(d) = f(S_d, R_d, V_d, A_d), \quad (2.32)$$

де S_d представляє рівень чутливості даних d , R_d представляє нормативні вимоги для даних d , V_d представляє бізнес-цінність даних d , A_d представляє шаблони доступу для даних d .

Зіставлення від класифікації до хмарного провайдера можна моделювати як:

$$P(C) = \arg \max_{i \in 1,2,\dots,n} \left(w_S \cdot S_i + w_R \cdot R_i + w_P \cdot P_i + w_C \cdot \frac{1}{C_i} \right), \quad (2.33)$$

де S_i – це можливість безпеки провайдера i , R_i – це відповідність нормативним вимогам провайдера i , P_i – це можливість продуктивності провайдера i , C_i – це вартість провайдера i , w_S , w_R , w_P та w_C – це ваги, що відображають відносну важливість кожного фактора.

Формат даних

Визначення: Формат даних у мультихмарному зберіганні відноситься до структури, кодування та представлення даних, що визначає їх сумісність, інтеоперабельність та вимоги до обробки у кількох хмарних провайдерів.

Одиниці вимірювання:

- стандартизація формату: Ступінь, до якого формати даних відповідають галузевим стандартам;
- накладні витрати на конвертацію: Ресурси, необхідні для конвертації між форматами при переміщенні даних;
- ефективність формату: Простір зберігання та ресурси обробки, необхідні для формату;
- сумісність формату: Кількість систем або провайдерів, які можуть нативно працювати з форматом.

Математичний вираз: Придатність формату F для мультихмарного середовища можна кількісно визначити як:

$$F = w_S \cdot S + w_C \cdot C + w_E \cdot E + w_I \cdot I, \quad (2.34)$$

де S представляє рівень стандартизації, C представляє сумісність між провайдерами, E представляє ефективність у зберіганні та обробці, I представляє інтегровуваність з додатками, w_S , w_C , w_E та w_I – це ваги, що відображають відносну важливість кожного фактора.

Вартість конвертації при переміщенні даних між провайдерами можна моделювати як:

$$C_{conversion} = V \cdot (t_{extract} + t_{transform} + t_{load}), \quad (2.35)$$

де V – це обсяг даних, $t_{extract}$ – це час для вилучення даних з вихідного формату, $t_{transform}$ – це час для перетворення даних у цільовий формат, t_{load} – це час для завантаження даних у новому форматі.

2.2.5 Критерії інтегровуваності та портативності

Портативність даних

Портативність даних відноситься до здатності переміщувати дані між різними хмарними провайдерами без значних технічних або фінансових бар'єрів.

Визначення: Портативність даних у мультихмарному зберіганні – це здатність переміщувати дані та додатки між різними хмарними провайдерами з мінімальними порушеннями, витратами або технічними зусиллями.

Одиниці вимірювання:

- індекс переносимості: Комплексна оцінка, що відображає легкість переміщення даних;
- час міграції: Час, необхідний для переміщення вказаного обсягу даних між провайдерами;
- вартість міграції: Фінансові витрати, пов'язані з міграцією даних;
- точність даних: Ступінь, до якого дані зберігають свою цілісність та придатність до використання після міграції.

Математичний вираз: Портативність P даних між хмарними провайдерами можна кількісно визначити як:

$$P = \frac{1}{T \cdot C \cdot E}, \quad (2.36)$$

де T – це нормалізований час, необхідний для міграції, C – це нормалізована вартість міграції, E – це нормалізовані зусилля (технічна складність) міграції. Вищі значення P вказують на більшу переносимість.

Для конкретних типів даних портативність можна виразити як:

$$P_d = \frac{V_d}{T_d \cdot C_d}, \quad (2.37)$$

де V_d – це обсяг даних типу d , T_d – це час для міграції даних типу d , C_d – це вартість міграції даних типу d .

Інтероперабельність

Визначення: Інтероперабельність у мултихмарному зберіганні — це здатність систем зберігання від різних хмарних провайдерів працювати разом безперешкодно, обмінюючись даними та послугами без спеціальних зусиль з інтеграції.

Одиниці вимірювання:

- сумісність API: Відсоток функцій, до яких можна отримати доступ через стандартні інтерфейси;
- зусилля з інтеграції: Час розробки, необхідний для забезпечення спільної роботи систем;
- ефективність обміну даними: Накладні витрати, що виникають при передачі даних між системами;
- функціональна еквівалентність: Ступінь, до якого функції послідовно доступні у різних провайдерів.

Математичний вираз: Інтероперабельність I між хмарними провайдерами можна моделювати як:

$$I = \frac{F_{common}}{F_{total}}, \quad (2.38)$$

де F_{common} – це кількість функцій або можливостей, які можна використовувати послідовно у різних провайдерів; F_{total} – це загальна кількість необхідних функцій або можливостей.

Ефективна інтероперабельність у мултихмарному середовищі з n провайдерами може бути виражена як:

$$I_{effective} = \min_{i,j \in 1,2,\dots,n} I_{ij}, \quad (2.39)$$

де I_{ij} представляє інтероперабельність між провайдерами i та j .

Зменшення прив'язки до постачальника (Vendor lock-in)

Визначення: Зменшення прив'язки до постачальника в мултихмарному зберіганні — це набір стратегій, архітектур та практик, розроблених для зменшення

залежності від конкретних хмарних провайдерів та підтримки свободи змінювати провайдерів з мінімальними порушеннями та витратами.

Одиниці вимірювання:

- оцінка ризику прив'язки: Комплексна міра залежності від специфічних для провайдера функцій;
- вартість переключення: Оцінені витрати на міграцію від одного провайдера до іншого;
- технологічна абстракція: Ступінь, до якого додатки ізольовані від специфічних для провайдера деталей;
- повнота стратегії виходу: Комплексність планів для зміни провайдерів.

Математичний вираз: Ризик прив'язки до постачальника L для хмарного провайдера можна кількісно визначити як:

$$L = w_P \cdot P + w_T \cdot T + w_C \cdot C + w_D \cdot D, \quad (2.40)$$

де P представляє використання власних технологій; T представляє час переключення; C представляє вартість переключення; D представляє обсяг та складність даних; w_P , w_T , w_C та w_D – це ваги, що відображають відносну важливість кожного фактора.

Загальний ризик прив'язки в мультихмарному середовищі можна виразити як:

$$L_{overall} = \sum_{i=1}^n w_i \cdot L_i, \quad (2.41)$$

де L_i – це ризик прив'язки для провайдера i ; w_i – це вага, що відображає важливість або рівень використання провайдера i .

Репутація постачальника

Визначення: Репутація постачальника в мультихмарному зберіганні – це оцінка хмарних провайдерів на основі їх надійності, історії продуктивності, практик безпеки, фінансової стабільності та задоволеності клієнтів.

Одиниці вимірювання:

- оцінка репутації: Комплексний рейтинг на основі кількох факторів репутації;
- історія надійності послуг: Історичні показники часу безперебійної роботи та продуктивності;
- частота інцидентів безпеки: Кількість та серйозність минулих порушень безпеки;

- задоволеність клієнтів: Рейтинги та відгуки від існуючих клієнтів;
- позиція на ринку: Частка ринку та траєкторія зростання.

Математичний вираз: Репутацію R хмарного провайдера можна моделювати як:

$$R = w_A \cdot A + w_S \cdot S + w_F \cdot F + w_C \cdot C + w_I \cdot I, \quad (2.42)$$

де A представляє історичну доступність; S представляє історію безпеки; F представляє фінансову стабільність; C представляє сертифікати відповідності; I представляє інноваційну здатність; w_A, w_S, w_F, w_C та w_I – це ваги, що відображають відносну важливість кожного фактора.

Вибір провайдерів, зважений за репутацією, в мультихмарному середовищі можна виразити як:

$$P_{selected} = \arg \max_{i \in 1,2,\dots,n} (R_i \cdot F_i), \quad (2.43)$$

де R_i – це репутація провайдера i ; F_i – це функціональна відповідність провайдера i для конкретної вимоги.

2.2.6 Критерії сталого розвитку та екологічності

Критерії сталого розвитку та екологічності стосуються екологічного впливу рішень мультихмарного зберігання, зосереджуючись на енергоефективності, використанні ресурсів та екологічній відповідальності [46,47,50].

Вплив на навколишнє середовище

Вплив на навколишнє середовище відноситься до екологічних наслідків операцій хмарного зберігання, включаючи викиди вуглецю, споживання ресурсів та утворення відходів.

Визначення: Вплив на навколишнє середовище в мультихмарному зберіганні – це міра екологічних наслідків, що виникають внаслідок роботи систем зберігання у кількох хмарних провайдерів, включаючи викиди вуглецю, споживання ресурсів та електронні відходи.

Одиниці вимірювання:

- вуглецевий слід: Еквівалентні викиди CO₂ (метричні тонни) [56];
- ефективність використання енергії (PUE): Співвідношення загальної енергії центру обробки даних до обчислювальної енергії;
- ефективність використання води (WUE): Споживання води на одиницю обчислювальної енергії;

- утворення електронних відходів: Вага електронних відходів, вироблених за рік [59];
- відсоток відновлюваної енергії: Частка енергії з відновлюваних джерел.

Математичний вираз: Вплив на навколишнє середовище E рішення мультимарного зберігання можна обчислити як:

$$E = \sum_{i=1}^n (w_C \cdot C_i + w_W \cdot W_i + w_M \cdot M_i), \quad (2.44)$$

де C_i – це викиди вуглецю для провайдера i ; W_i – це споживання води для провайдера i ; M_i – це матеріальні відходи для провайдера i ; w_C , w_W та w_M – це ваги, що відображають відносну важливість кожного фактора.

Вуглецевий слід конкретно можна виразити як:

$$C = \sum_{i=1}^n (E_i \cdot EF_i), \quad (2.45)$$

де E_i – це споживання енергії для провайдера i ; EF_i – це коефіцієнт викидів для джерел енергії, що використовуються провайдером i .

Енергоефективність

Визначення: Енергоефективність у мультимарному зберіганні – це міра того, наскільки ефективно хмарні провайдери перетворюють енергетичний вхід у корисний вихід зберігання та обробки даних, мінімізуючи відходи та непотрібне споживання.

Одиниці вимірювання:

- Ефективність використання енергії (PUE): Співвідношення загальної енергії об'єкта до енергії ІТ-обладнання
- Ватт на ТБ зберігання: Енергія, необхідна на терабайт зберігання
- Продуктивність на Ватт: Обчислювальний вихід на одиницю енергії
- Ефективність використання вуглецю (CUE): Викиди вуглецю на одиницю обчислювальної енергії

Математичний вираз: Енергоефективність η системи зберігання хмарного провайдера можна обчислити як:

$$\eta = \frac{S \cdot P}{E}, \quad (2.46)$$

де S – це ємність зберігання; P – це показник продуктивності; E – це споживання енергії.

Для порівняння провайдерів у мультихмарному середовищі нормалізована енергоефективність може бути виражена як:

$$\eta_{normalized} = \frac{\eta}{\eta_{reference}}, \quad (2.47)$$

де $\eta_{reference}$ – це еталонне значення ефективності, таке як середнє по галузі.

Сталість ресурсів

Сталість ресурсів відноситься до відповідального використання та управління ресурсами для забезпечення їх довгострокової доступності та мінімізації впливу на навколишнє середовище.

Визначення: Сталість ресурсів у мультихмарному зберіганні – це практика використання обчислювальних, мережевих ресурсів та ресурсів зберігання способами, що мінімізують відходи, максимізують ефективність та забезпечують довгострокову доступність, одночасно зменшуючи вплив на навколишнє середовище.

Одиниці вимірювання:

- використання ресурсів: Відсоток виділених ресурсів, що активно використовуються;
- життєвий цикл обладнання: Середній термін служби фізичної інфраструктури;
- рівень переробки: Відсоток матеріалів, перероблених в кінці життєвого циклу;
- відсоток відновлюваних ресурсів: Частка ресурсів з відновлюваних або перероблених джерел;

Математичний вираз: Індекс сталості S для хмарного провайдера можна моделювати як:

$$S = w_U \cdot U + w_R \cdot R + w_L \cdot L + w_C \cdot C, \quad (2.48)$$

де U представляє ефективність використання ресурсів; R представляє використання відновлюваних ресурсів; L представляє довжину життєвого циклу обладнання; C представляє практики циркулярної економіки; w_U , w_R , w_L та w_C – це ваги, що відображають відносну важливість кожного фактора.

Загальна сталість мультимарного рішення може бути виражена як:

$$S_{overall} = \sum_{i=1}^n w_i \cdot S_i, \quad (2.49)$$

де S_i – це індекс сталості для провайдера i ; w_i – це вага, що відображає частку ресурсів, виділених провайдеру i .

Інструменти моніторингу

Інструменти моніторингу забезпечують видимість продуктивності, доступності та використання ресурсів мультимарних систем зберігання.

Визначення: Інструменти моніторингу в мультимарному зберіганні – це програмні рішення, що забезпечують видимість продуктивності, доступності, безпеки та використання ресурсів систем зберігання у кількох хмарних провайдерів.

Одиниці вимірювання:

- покриття: Відсоток інфраструктури та послуг, що моніторяться;
- гранулярність даних: Рівень деталізації в зібраних показниках;
- Точність сповіщень: Співвідношення дієвих сповіщень до хибних спрацьовувань;
- час відгуку: Час між виникненням проблеми та її виявленням.

Математичний вираз: Ефективність моніторингу M можна кількісно визначити як:

$$M = C \cdot G \cdot A \cdot \frac{1}{T}, \quad (2.50)$$

де C представляє покриття (відсоток систем, які покриті моніторингом); G представляє гранулярність зібраних даних; A представляє точність сповіщень; T представляє час виявлення.

Для мультимарних середовищ можливість моніторингу можна виразити як:

$$M_{multi} = \min_{i \in 1, 2, \dots, n} M_i, \quad (2.51)$$

де M_i представляє ефективність моніторингу для хмарного провайдера i .

Звітність

Звітність відноситься до генерації структурованої інформації про стан, продуктивність та використання мультимарних систем зберігання.

Визначення: Звітність у мультихмарному зберіганні – це процес генерації структурованої інформації про стан, продуктивність, використання та відповідність систем зберігання у кількох хмарних провайдерів для підтримки прийняття рішень та підзвітності.

Одиниці вимірювання:

- комплексність звітності: Відсоток необхідних показників, включених до звітів;
- точність звітності: Ступінь, до якого звітні дані відображають фактичні умови;
- своєчасність звітності: Частота генерації звітів та затримка в даних;
- можливість налаштування: Здатність адаптувати звіти для різних аудиторій та цілей.

Математичний вираз: Якість звітності R можна моделювати як:

$$R = w_C \cdot C + w_A \cdot A + w_T \cdot T + w_U \cdot U, \quad (2.52)$$

де C представляє комплексність; A представляє точність; T представляє своєчасність; U представляє зручність використання; w_C , w_A , w_T та w_U – це ваги, що відображають відносну важливість кожного фактора.

Цінність підтримки прийняття рішень D звітності можна виразити як:

$$D = R \cdot I \cdot A, \quad (2.53)$$

де R – це якість звітності, I – це релевантність інформації для рішень, A – це дієвість звітної інформації.

2.3 Онтологічна модель розподіленого зберігання даних у мультихмарних середовищах

Складність мультихмарних середовищ зберігання вимагає структурованого підходу до розуміння та формалізації відносин між різними сутностями, процесами та критеріями. Онтологічна модель забезпечує формальне представлення доменних знань, фіксуючи концепції, відношення та обмеження, що регулюють розподілене зберігання даних у кількох хмарних провайдерів [102-104]. У цьому розділі представлена комплексна онтологічна модель, яка базується на критеріях, встановлених у попередньому розділі, пропонуючи концептуальну основу для аналізу та проектування ефективних рішень мультихмарного зберігання.

2.3.1 Концептуальна основа для мультихмарного зберігання

Концептуальна основа для мультихмарного зберігання встановлює фундаментальні елементи та відношення, що характеризують середовища розподіленого зберігання даних, що охоплюють кількох хмарних провайдерів. Ця основа служить базою для розробки більш детальної онтологічної моделі.

Мультихмарні середовища зберігання є за своєю природою складними, включаючи взаємодії між різними зацікавленими сторонами, системами та типами даних через організаційні та провайдерські межі. Концептуальна основа допомагає організувати цю складність шляхом ідентифікації ключових сутностей та їх відносин, встановлення спільного словника та надання структури для аналізу вимог до зберігання та рішень.

Запропонована концептуальна основа для мультихмарного зберігання охоплює кілька ключових вимірів:

1. **Архітектурні компоненти:** Фізичні та логічні елементи, що складають інфраструктуру мультихмарного зберігання, включаючи ресурси зберігання, мережеві з'єднання, системи управління та механізми безпеки.
2. **Зацікавлені сторони:** Особи та організації, залучені до екосистеми мультихмарного зберігання, включаючи власників даних, постачальників послуг, адміністраторів, кінцевих користувачів та регуляторні органи.
3. **Характеристики даних:** Властивості даних, що зберігаються, включаючи їх тип, формат, розмір, чутливість та етап життєвого циклу.
4. **Операційні процеси:** Діяльність та робочі потоки, залучені до управління даними у кількох хмарних провайдерів, включаючи розміщення даних, міграцію, резервне копіювання, відновлення та контроль доступу.
5. **Атрибути якості:** Нефункціональні характеристики, що визначають ефективність рішення мультихмарного зберігання, включаючи продуктивність, надійність, безпеку, економічну ефективність та сталість.
6. **Аспекти управління:** Політики, стандарти та нормативні вимоги, що регулюють, як дані зберігаються, захищаються та доступні у кількох хмарних провайдерів.

Ці виміри взаємопов'язані та впливають один на одного різними способами. Наприклад, характеристики даних впливають на архітектурні рішення, які, в свою чергу, впливають на операційні процеси та атрибути якості. Аналогічно, аспекти управління обмежують архітектурні вибори та операційні процедури.

Концептуальна основа також визнає динамічний характер мультихмарних середовищ, де умови, вимоги та можливості еволюціонують з часом. Ця динамічність вимагає адаптивних підходів до управління даними, які можуть

реагувати на зміни в шаблонах робочого навантаження, пропозиціях провайдерів, нормативних ландшафтах та організаційних пріоритетах.

Формально концептуальну основу можна представити як кортеж:

$$MCF = (AC, S, DC, OP, QA, GA), \quad (2.54)$$

де AC – множина архітектурних компонентів; S – множина зацікавлених сторін; DC – множина характеристик даних; OP – множина операційних процесів; QA – множина атрибутів якості; GA – множина аспектів управління.

Кожен з цих елементів можна далі формалізувати. Наприклад, множину архітектурних компонентів можна визначити як:

$$AC = ac_1, ac_2, \dots, ac_n, \quad (2.55)$$

де кожен компонент ac_i має свої властивості та функції:

$$ac_i = (id_i, type_i, cap_i, func_i), \quad (2.57)$$

де id_i – унікальний ідентифікатор компонента; $type_i$ – тип компонента (наприклад, сховище, мережа, обчислення); cap_i – потужність або ємність компонента; $func_i$ – функціональні можливості компонента.

Взаємодії між вимірами можна представити як відношення між відповідними множинами. Наприклад, відношення між характеристиками даних та атрибутами якості:

$$R_{DC,QA} \subseteq DC \times QA, \quad (2.58)$$

де пара $(dc, qa) \in R_{DC,QA}$ означає, що характеристика даних dc впливає на атрибут якості qa .

2.3.2 Відношення між сутностями в мультимарних системах зберігання

Спираючись на концептуальну основу, цей розділ досліджує ключові сутності в мультимарних системах зберігання та відношення між ними. Розуміння цих відносин є важливим для розробки ефективної онтологічної моделі, яка фіксує складність розподіленого зберігання даних у кількох хмарних провайдерів.

Домен мультихмарного зберігання охоплює різноманітний набір сутностей, які взаємодіють складними способами. Ці сутності можна категоризувати на кілька груп:

Сутності даних:

- **Об'єкти даних:** Фундаментальні одиниці інформації, що зберігаються, які можуть включати файли, документи, записи або інші структуровані чи неструктуровані дані.
- **Колекції даних:** Логічні групування пов'язаних об'єктів даних, такі як бази даних, файлові системи, об'єктні сховища або репозиторії контенту.
- **Метадані:** Інформація про об'єкти даних, включаючи описові атрибути, технічні властивості та інформацію про управління.

Сутності інфраструктури:

- **Ресурси зберігання:** Фізичні або віртуальні носії зберігання, що використовуються для збереження даних, включаючи диски, томи, бакети та контейнери.
- **Обчислювальні ресурси:** Можливості обробки, що використовуються для операцій з даними, включаючи сервери, віртуальні машини та безсерверні функції.
- **Мережеві ресурси:** Комунікаційні шляхи, що забезпечують передачу даних між компонентами, включаючи з'єднання, шлюзи та мережі доставки контенту.

Сутності провайдерів:

- **Хмарні провайдери:** Організації, що пропонують послуги зберігання, кожна зі своєю інфраструктурою, можливостями, моделями ціноутворення та угодами про рівень обслуговування.
- **Сервісні пропозиції:** Конкретні послуги зберігання, що надаються хмарними провайдерами, такі як об'єктне зберігання, блочне зберігання, файлове зберігання або послуги баз даних.
- **Регіони та зони:** Географічні поділи інфраструктури хмарного провайдера, що впливають на локальність даних, затримку та відповідність нормативним вимогам.

Сутності управління:

- **Політики:** Правила та керівні принципи, що регулюють, як дані зберігаються, захищаються, доступні та керуються.
- **Процеси:** Послідовності дій, що виконуються для управління даними протягом їх життєвого циклу.
- **Інструменти:** Програмні системи, що використовуються для моніторингу, контролю та оптимізації мультихмарних середовищ зберігання.

Сутності користувачів:

- **Організації:** Суб'єкти, які володіють та відповідають за дані, що зберігаються.
- **Адміністратори:** Особи, відповідальні за управління мультимарним середовищем зберігання.
- **Кінцеві користувачі:** Особи або системи, які створюють, отримують доступ та використовують збережені дані.

Формально ці сутності можна представити як множини:

- $D = d_1, d_2, \dots, d_n$ (множина об'єктів даних)
- $DC = dc_1, dc_2, \dots, dc_m$ (множина колекцій даних)
- $MD = md_1, md_2, \dots, md_p$ (множина метаданих)
- $SR = sr_1, sr_2, \dots, sr_q$ (множина ресурсів зберігання)
- $CR = cr_1, cr_2, \dots, cr_r$ (множина обчислювальних ресурсів)
- $NR = nr_1, nr_2, \dots, nr_s$ (множина мережевих ресурсів)
- $CP = cp_1, cp_2, \dots, cp_t$ (множина хмарних провайдерів)
- $SO = so_1, so_2, \dots, so_u$ (множина сервісних пропозицій)
- $RG = rg_1, rg_2, \dots, rg_v$ (множина регіонів)
- $PL = pl_1, pl_2, \dots, pl_w$ (множина політик)
- $PR = pr_1, pr_2, \dots, pr_x$ (множина процесів)
- $TL = tl_1, tl_2, \dots, tl_y$ (множина інструментів)
- $ORG = org_1, org_2, \dots, org_z$ (множина організацій)
- $ADM = adm_1, adm_2, \dots, adm_a$ (множина адміністраторів)
- $USR = usr_1, usr_2, \dots, usr_b$ (множина кінцевих користувачів)

Ці сутності пов'язані через різні відношення, які визначають, як вони взаємодіють в екосистемі мультимарного зберігання. Формально ці відношення можна представити як:

Відношення власності та відповідальності:

- $\text{owns} \subseteq \text{ORG} \times (\text{D} \cup \text{DC})$ - відношення власності організацій над даними;
- $\text{operates} \subseteq \text{CP} \times \text{SR}$ - відношення експлуатації провайдерами ресурсів;
- $\text{responsible} \subseteq \text{ADM} \times \text{PL}$ - відношення відповідальності адміністраторів за політики.

Відношення вмісту та композиції:

- $\text{contains} \subseteq \text{DC} \times \text{D}$ - відношення вмісту колекцій даних;
- $\text{hosts} \subseteq \text{SR} \times (\text{D} \cup \text{DC})$ - відношення розміщення ресурсами даних;

- $\text{includes} \subseteq \text{RG} \times \text{RG}$ - відношення включення регіонів (наприклад, регіон включає зони).

Операційні відношення:

- $\text{governs} \subseteq \text{PL} \times \text{D}$ - відношення регулювання політиками даних;
- $\text{allocates} \subseteq \text{SR} \times \text{D}$ - відношення виділення ресурсів для даних;
- $\text{connects} \subseteq \text{NR} \times (\text{SR} \cup \text{CR})$ - відношення з'єднання мережевими ресурсами.

Відношення управління:

- $\text{restricts} \subseteq \text{PL} \times (\text{D} \times \text{RG})$ - відношення обмеження розміщення даних;
- $\text{controls} \subseteq \text{PL} \times (\text{USR} \times \text{D})$ - відношення контролю доступу до даних;
- $\text{influences} \subseteq \text{QA} \times \text{SR}$ - відношення впливу атрибутів якості на ресурси.

Часові відношення:

- $\text{evolves} \subseteq \text{D} \times \text{T} \times \text{LC}$ - відношення еволюції даних через етапи життєвого циклу LC у часі T;
- $\text{provisions} \subseteq \text{SR} \times \text{T}$ - відношення забезпечення ресурсів у часі;
- $\text{schedules} \subseteq \text{PR} \times \text{T}$ - відношення планування процесів у часі.

Ці відношення можна формально визначити як функції або відношення з відповідними властивостями. Наприклад, відношення власності можна визначити як:

$$\text{owns}: \text{ORG} \times (\text{D} \cup \text{DC}) \rightarrow \text{true}, \text{false},$$

де $\text{owns}(\text{org}, d) = \text{true}$ означає, що організація org володіє об'єктом даних d .

Аналогічно, відношення розміщення можна визначити як:

$$\text{hosts}: \text{SR} \times (\text{D} \cup \text{DC}) \rightarrow \text{true}, \text{false},$$

де $\text{hosts}(\text{sr}, d) = \text{true}$ означає, що ресурс зберігання sr розміщує об'єкт даних d .

Для більш складних відношень, таких як еволюція даних через етапи життєвого циклу, можна використовувати функції з часовою залежністю:

$$\text{evolves}: \text{D} \times \text{T} \rightarrow \text{LC},$$

де $\text{evolves}(d, t) = \text{lc}$ означає, що об'єкт даних d знаходиться на етапі життєвого циклу lc у час t .

2.3.3 Формалізація процесів та представлення робочих потоків

Для ефективного управління даними в мультихмарних середовищах необхідно формалізувати процеси та робочі потоки, що регулюють переміщення, зберігання та доступ до даних у кількох провайдерів. Далі представлено структурований підхід до формалізації процесів, визначаючи ключові робочі потоки та їх компоненти.

Процеси мультихмарного зберігання охоплюють широкий спектр діяльності, від початкових рішень щодо розміщення даних до поточного управління та остаточного виведення даних з експлуатації. Ці процеси повинні враховувати різноманітні можливості, обмеження та інтерфейси різних хмарних провайдерів, забезпечуючи при цьому послідовну обробку даних відповідно до організаційних вимог. Формально процес P можна представити як кортеж:

$$P = (I, A, O, C), \quad (2.59)$$

де $I \subseteq U$ – множина вхідних даних та параметрів з універсальної множини U , $A = a_1, a_2, \dots, a_n$ – впорядкована множина дій або операцій, $O \subseteq U$ – множина вихідних результатів, C – множина обмежень, що застосовуються до процесу.

Кожна дія $a_i \in A$ може бути представлена як:

$$a_i = (pre_i, op_i, post_i, res_i, dur_i), \quad (2.60)$$

де $pre_i: U \rightarrow true, false$ – предикат, що визначає передумови для виконання операції; $op_i: U \rightarrow U$ – функція перетворення, що представляє саму операцію; $post_i: U \rightarrow true, false$ – предикат, що визначає післяумови після виконання операції; $res_i \subseteq R$ – множина ресурсів, необхідних для виконання операції; $dur_i: R^+$ – тривалість виконання операції.

Основні процеси мультихмарного зберігання:

1. **Процес розміщення даних:** Визначає, де дані повинні зберігатися на основі різних критеріїв.

Формально цей процес можна представити як:

$$P_{placement} = (I_{placement}, A_{placement}, O_{placement}, C_{placement}), \quad (2.61)$$

де $I_{placement} = D_{req}, S_{req}, CP_{cap}$ – вхідні дані, що включають вимоги до даних, вимоги до зберігання та можливості провайдерів; $A_{placement} = analyze, evaluate, optimize, decide$ – послідовність дій; $O_{placement} =$

$D_{placement}, S_{allocation}$ – вихідні результати, що включають рішення про розміщення даних та план розподілу зберігання; $C_{placement} = C_{cost}, C_{perf}, C_{comp}$ – обмеження, що включають обмеження вартості, вимоги до продуктивності та обмеження відповідності.

2. **Процес завантаження даних:** Передає дані з їх джерела до вибраних місць хмарного зберігання.

$$P_{ingestion} = (I_{ingestion}, A_{ingestion}, O_{ingestion}, C_{ingestion}), \quad (2.62)$$

де $I_{ingestion} = D_{source}, T_{target}, P_{params}$ – вхідні дані, що включають вихідні дані, цільові місця та параметри передачі; $A_{ingestion} = prepare, transfer, verify$ – послідовність дій; $O_{ingestion} = D_{stored}, L_{transfer}, R_{verify}$ – вихідні результати, що включають збережені дані, логи передачі та результати перевірки; $C_{ingestion} = C_{bw}, C_{time}, C_{format}$ – обмеження, що включають обмеження пропускну здатності, часові вікна та сумісність форматів.

3. **Процес реплікації даних:** Створює та підтримує копії даних у кількох провайдерів.

$$P_{replication} = (I_{replication}, A_{replication}, O_{replication}, C_{replication}), \quad (2.63)$$

де $I_{replication} = D_{source}, P_{policy}, T_{target}$ – вхідні дані, що включають вихідні дані, політику реплікації та цільові місця; $A_{replication} = initial_replication, detect_changes, synchronize$ – послідовність дій; $O_{replication} = D_{replicated}, S_{consistency}, L_{replication}$ – вихідні результати, що включають репліковані дані, статус узгодженості та логи реплікації; $C_{replication} = C_{consistency}, C_{bw}, C_{cost}$ – обмеження, що включають вимоги до узгодженості, обмеження пропускну здатності та обмеження вартості.

Для кожного процесу можна визначити потік виконання як орієнтований граф:

$$G_P = (A, E), \quad (2.64)$$

де A – множина дій процесу P ; $E \subseteq A \times A$ – множина спрямованих ребер, що представляють послідовність виконання. Ребро $(a_i, a_j) \in E$ означає, що дія a_j виконується після дії a_i .

Для процесів з умовними розгалуженнями можна визначити функцію переходу:

$$\delta: A \times C \rightarrow A, \quad (2.65)$$

де $\delta(a_i, c)$ визначає наступну дію після a_i при умові c .

Точки прийняття рішень та правила:

Точки прийняття рішень у процесах можна формалізувати як предикати, що визначають шлях виконання:

$$decision: U \rightarrow true, false.$$

Наприклад, рішення про застосування шифрування на основі чутливості даних можна представити як:

$$encrypt_decision(d) = \begin{cases} true, & \text{якщо } sensitivity(d) \geq threshold \\ false, & \text{інакше} \end{cases},$$

де $sensitivity: D \rightarrow R$ – функція, що визначає рівень чутливості даних, а $threshold$ – порогове значення для застосування шифрування.

Показники процесів та оптимізація:

Для оцінки ефективності процесів можна визначити множину метрик:

$$M = m_1, m_2, \dots, m_k, \quad (2.66)$$

де кожна метрика m_i є функцією, що відображає виконання процесу на числове значення:

$$m_i: P \times Execution \rightarrow R.$$

Наприклад, метрика часу завершення процесу може бути визначена як:

$$completion_time(P, e) = \sum_{a \in A_P} duration(a, e), \quad (2.67)$$

де $duration(a, e)$ - тривалість виконання дії a у конкретному виконанні e .

Оптимізація процесу може бути сформульована як задача мінімізації або максимізації відповідних метрик:

$$\text{optimize}(P) = \arg \min_{P' \in \mathcal{P}} \sum_{i=1}^k w_i \cdot m_i(P'), \quad (2.68)$$

де \mathcal{P} – множина можливих варіантів процесу; w_i – ваги, що відображають відносну важливість кожної метрики.

2.2.4. Запропонована онтологічна модель на основі комплексних критеріїв

Спираючись на концептуальну основу, відношення між сутностями та формалізацію процесів, обговорені в попередніх розділах, тепер представляється комплексна онтологічна модель для мультимарного зберігання даних. Ця модель інтегрує критерії, встановлені в розділі 2.1, та забезпечує формальне представлення доменних знань, необхідних для аналізу, проектування та оптимізації рішень розподіленого зберігання у кількох хмарних провайдерів.

Формально онтологічну модель можна представити як кортеж:

$$O = (C, R, A, I, Ax), \quad (2.69)$$

де C – множина класів (концепцій), R – множина відношень між класами, A – множина атрибутів класів, I – множина екземплярів класів, Ax – множина аксіом, що визначають обмеження та правила виведення.

Множина класів C включає всі концепції, визначені в попередніх розділах, організовані в таксономію з відношенням наслідування $is_a \subseteq C \times C$:

C = Data, DataClassification, DataLifecycle, StorageResource, CloudProvider, StorageService, Region, StoragePolicy, PerformanceMetric, SecurityControl, ComplianceRequirement, CostComponent, DataPlacement, DataOperation, ...

Для кожного класу $c \in C$ визначено множину атрибутів $A_c \subseteq A$:

A_{Data}
= dataID, size, format, sensitivity, creationDate, lastAccessDate, ...
A_{CloudProvider}
= providerID, name, serviceLevel, reputation, costModel, ...
A_{StorageService}
= serviceID, type, capabilities, limitations, pricingStructure, ...

Відношення між класами R включають різні типи зв'язків:

R = classifiedAs, hasLifecycle, placedIn, undergoes, providedBy, partOf, locatedIn, offers, operates, governedBy, usesResource, measuredBy, securedBy, characterizedBy, pricedBy, implementedBy, enforcedBy, ...

Кожне відношення $r \in R$ має домен $\text{dom}(r) \subseteq C$ і діапазон $\text{range}(r) \subseteq C$:

$$\begin{aligned}
\text{dom}(\text{classifiedAs}) &= \text{Data}, \text{range}(\text{classifiedAs}) \\
&= \text{DataClassificationdom}(\text{providedBy}) \\
&= \text{StorageResource}, \text{range}(\text{providedBy}) \\
&= \text{CloudProviderdom}(\text{offers}) = \text{CloudProvider}, \text{range}(\text{offers}) \\
&= \text{StorageService}
\end{aligned}$$

Аксіоми Ах визначають обмеження та правила виведення, що застосовуються до класів, атрибутів та відносин:

1. Кожен об'єкт даних повинен мати принаймні одну класифікацію: $\forall d \in \text{Data}, \exists c \in \text{DataClassification}: \text{classifiedAs}(d, c)$
2. Кожен ресурс зберігання надається рівно одним хмарним провайдером: $\forall r \in \text{StorageResource}, \exists! p \in \text{CloudProvider}: \text{providedBy}(r, p)$
3. Якщо дані розміщені в ресурсі зберігання, то вони підпорядковуються всім політикам, що застосовуються до цього ресурсу:

$$\begin{aligned}
&\forall d \in \text{Data}, \forall r \in \text{StorageResource}, \forall p \\
&\quad \in \text{StoragePolicy}: \text{placedIn}(d, r) \wedge \text{appliesTo}(p, r) \\
&\quad \Rightarrow \text{governedBy}(d, p)
\end{aligned}$$

4. Дані з високою чутливістю повинні зберігатися лише в ресурсах з відповідним рівнем шифрування:

$$\begin{aligned}
&\forall d \in \text{Data}, \forall r \\
&\quad \in \text{StorageResource}: (\text{sensitivity}(d) \geq \text{threshold}) \\
&\quad \wedge \text{placedIn}(d, r) \\
&\quad \Rightarrow (\text{encryptionLevel}(r) \geq \text{requiredLevel}(\text{sensitivity}(d)))
\end{aligned}$$

Онтологічна модель також включає правила виведення, які дозволяють отримувати нові знання з існуючих фактів:

1. Транзитивність відношення розміщення через колекції даних: $\forall d \in \text{Data}, \forall c \in \text{DataCollection}, \forall r \in \text{StorageResource}: \text{partOf}(d, c) \wedge \text{placedIn}(c, r) \Rightarrow \text{placedIn}(d, r)$
2. Виведення відповідності нормативним вимогам: $\forall d \in \text{Data}, \forall r \in \text{StorageResource}, \forall p \in \text{CloudProvider}, \forall \text{reg} \in \text{Regulation}: \text{placedIn}(d, r) \wedge \text{providedBy}(r, p) \wedge \text{compliesWith}(p, \text{reg}) \Rightarrow \text{compliantWith}(d, \text{reg})$
3. Обчислення загальної вартості зберігання: $\text{totalCost}(d) = \sum_{r \in \text{StorageResource}} \text{placedIn}(d, r) \cdot \text{size}(d) \cdot \text{costPerUnit}(r)$

Інтеграція комплексних критеріїв:

Онтологічна модель включає комплексний набір критеріїв, встановлених у розділі 2.1, шляхом їх відображення на відповідні класи та відношення:

1. **Критерії, пов'язані з продуктивністю:**

- Клас PerformanceMetric з підкласами LatencyMetric, ThroughputMetric, IOPSMetric
- Атрибути threshold, unit, currentValue
- Відношення measuredBy між DataOperation та PerformanceMetric
- Відношення characterizedBy між StorageService та PerformanceMetric

Формально:

$$\begin{aligned}
 &PerformanceMetric \subseteq C \\
 &LatencyMetric, ThroughputMetric, IOPSMetric \subseteq PerformanceMetric \\
 &threshold, unit, currentValue \in A_{PerformanceMetric} \\
 &measuredBy \subseteq DataOperation \times PerformanceMetric \\
 &characterizedBy \subseteq StorageService \times PerformanceMetric
 \end{aligned}$$

2. Критерії надійності та доступності:

- Клас ProtectionPolicy з атрибутами redundancyLevel, availabilityTarget, backupFrequency, rpo, rto
- Відношення protectedBy між Data та ProtectionPolicy
- Відношення implementedBy між ProtectionPolicy та StorageService

Формально:

$$\begin{aligned}
 &ProtectionPolicy \subseteq StoragePolicy \\
 &redundancyLevel, availabilityTarget, backupFrequency, rpo, rto \\
 &\in A_{ProtectionPolicy} \\
 &protectedBy \subseteq Data \times ProtectionPolicy \\
 &implementedBy \subseteq ProtectionPolicy \times StorageService
 \end{aligned}$$

3. Критерії безпеки та відповідності:

- Клас SecurityControl з підкласами EncryptionControl, AccessControl, AuditControl
- Клас ComplianceRequirement з атрибутами regulation, jurisdiction, controlMapping
- Відношення securedBy між DataOperation та SecurityControl
- Відношення enforcedBy між ComplianceRequirement та StoragePolicy

Формально:

$$\begin{aligned}
 &SecurityControl \subseteq C \\
 &EncryptionControl, AccessControl, AuditControl \subseteq SecurityControl \\
 &ComplianceRequirement \subseteq C \\
 ®ulation, jurisdiction, controlMapping \in A_{ComplianceRequirement} \\
 &securedBy \subseteq DataOperation \times SecurityControl \\
 &enforcedBy \subseteq ComplianceRequirement \times StoragePolicy
 \end{aligned}$$

На рис 2.1 зображена діаграма ілюструє основну структуру запропонованої онтологічної моделі.

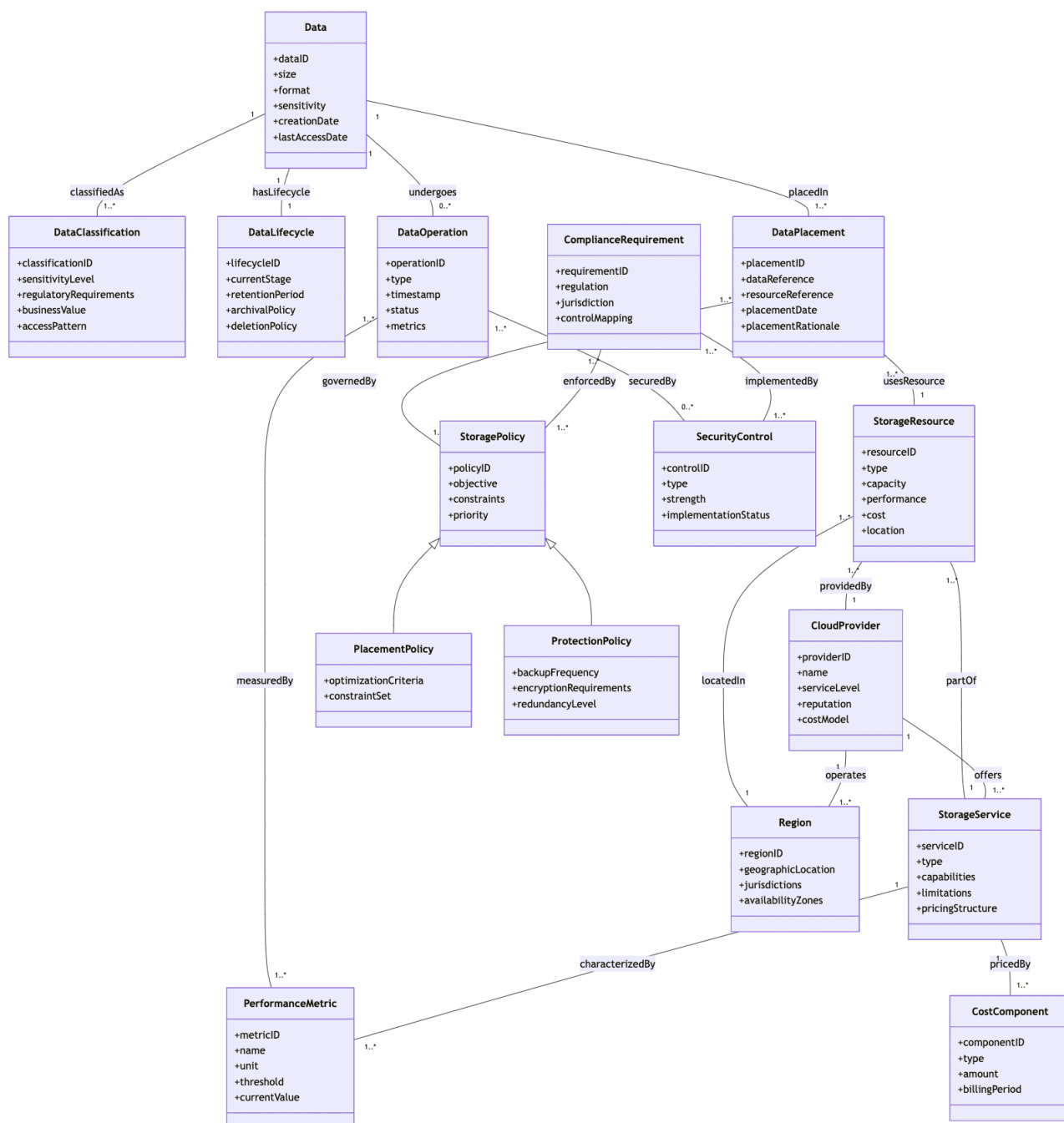


Рисунок 2.1 Запропонована онтологічна модель системи

Ця діаграма представляє основні класи та відношення в онтологічній моделі. Фактична реалізація включатиме додаткові класи, властивості та відношення для повного фіксування складності мультимарних середовищ зберігання. Математично відношення між класами представлена в Таблиці 2.1

Таблиця 2.1 Опис кроків запропонованого комплексного методу та їх вплив

N	Математичні вирази відношень між класами
1	$classifiedAs \subseteq Data \times DataClassification$
2	$hasLifecycle \subseteq Data \times DataLifecycle$
3	$placedIn \subseteq Data \times DataPlacement$
4	$undergoes \subseteq Data \times DataOperation$
5	$providedBy \subseteq StorageResource \times CloudProvider$
6	$partOf \subseteq StorageResource \times StorageService$
7	$locatedIn \subseteq StorageResource \times Region$
8	$offers \subseteq CloudProvider \times StorageService$
9	$operates \subseteq CloudProvider \times Region$
10	$is_a(PlacementPolicy, StoragePolicy)$
11	$is_a(ProtectionPolicy, StoragePolicy)$
12	$usesResource \subseteq DataPlacement \times StorageResource$
13	$governedBy \subseteq DataPlacement \times StoragePolicy$
14	$measuredBy \subseteq DataOperation \times PerformanceMetric$
15	$securedBy \subseteq DataOperation \times SecurityControl$
16	$characterizedBy \subseteq StorageService \times PerformanceMetric$
17	$pricedBy \subseteq StorageService \times CostComponent$
18	$implementedBy \subseteq ComplianceRequirement \times SecurityControl$
19	$enforcedBy \subseteq ComplianceRequirement \times StoragePolicy$

Для кожного відношення можна визначити додаткові властивості, такі як кардинальність, транзитивність, симетричність або рефлексивність. Наприклад:

- Відношення *providedBy* має кардинальність "багато до одного" (кожен ресурс зберігання надається рівно одним хмарним провайдером)
- Відношення *locatedIn* є транзитивним (якщо ресурс розташований у зоні, а зона розташована в регіоні, то ресурс розташований у регіоні)
- Відношення *is_a* є транзитивним і рефлексивним

Онтологічна модель також включає функції інтерпретації, які відображають концепції та відношення на конкретні екземпляри в домені:

$$I(Data) \subseteq \Delta^I$$

$$I(classifiedAs) \subseteq I(Data) \times I(DataClassification),$$

де Δ^I - домен інтерпретації.

Аксіоми онтології можна виразити за допомогою логіки першого порядку або дескриптивної логіки. Наприклад, аксіома, що вимагає шифрування для чутливих даних:

$\forall d \in \text{Data}, \forall r \in \text{StorageResource}, \forall p$
 $\in \text{DataPlacement}: (\text{sensitivity}(d) \geq \text{highSensitivity}) \wedge \text{placedIn}(d, p)$
 $\wedge \text{usesResource}(p, r) \Rightarrow \exists c$
 $\in \text{EncryptionControl}: \text{appliesTo}(c, r) \wedge \text{strength}(c)$
 $\geq \text{requiredStrength}(\text{sensitivity}(d)).$

Ця формальна онтологічна модель забезпечує потужну основу для аналізу мультимарних середовищ зберігання даних, дозволяючи автоматизоване виведення, перевірку узгодженості та підтримку прийняття рішень щодо оптимального розміщення та управління даними.

Перелік операцій на основі онтологічної моделі:

Запропонована онтологічна модель підтримує різні типи операцій та запитів, які можуть допомогти в аналізі та оптимізації рішень мультимарного зберігання:

1. **Визначення класу зберігання даних:** Визначення відповідного класу зберігання для даних на основі їх характеристик та вимог
2. **Перевірка відповідності:** Перевірка, чи задовольняє конкретне розміщення даних всім застосовним нормативним вимогам
3. **Оптимізація вартості:** Визначення найбільш економічно ефективної конфігурації зберігання, що відповідає вимогам продуктивності та надійності
4. **Оцінка ризику:** Оцінка потенційних ризиків, пов'язаних з конкретними варіантами зберігання
5. **Аналіз впливу:** Визначення наслідків змін у вимогах або можливостях провайдера

Ці можливості операцій дозволяють більш складні підходи до управління мультимарним зберіганням, дозволяючи організаціям приймати обґрунтовані рішення на основі комплексного розуміння своїх даних, вимог та доступних варіантів.

Подальша адаптація онтологічної моделі під змінні умови мультимарного середовища:

Запропонована онтологічна модель не є статичною, а розроблена для можливості подальшого вдосконалення та доповнення в міру зміни домену мультимарного зберігання. Нові хмарні послуги, нормативні вимоги або технологічні можливості можуть бути включені шляхом розширення онтології додатковими класами, відносинами або атрибутами.

Підтримка онтології включає:

1. Регулярні огляди для забезпечення узгодженості з поточними технологіями та практиками
2. Включення відгуків з практичних застосувань

3. Валідація проти реальних сценаріїв
4. Інтеграція з новими стандартами та найкращими практиками

Підтримуючи та розвиваючи онтологічну модель, організації можуть забезпечити, що їх підхід до мультимарного зберігання залишається ефективним та актуальним у швидко змінюваному технологічному ландшафті.

Застосування онтологічної моделі:

Запропонована онтологічна модель може бути застосована різними способами для підтримки управління мультимарним зберіганням:

1. **Системи підтримки прийняття рішень:** Керування рішеннями щодо розміщення зберігання на основі комплексних критеріїв
2. **Забезпечення політик:** Забезпечення того, що практики зберігання відповідають організаційним політикам та нормативним вимогам
3. **Автоматизоване управління:** Забезпечення інтелектуальних систем для управління даними у кількох хмарних провайдерів
4. **Представлення знань:** Надання формальної основи для обміну та інтеграції знань про мультимарне зберігання
5. **Освіта та навчання:** Підтримка розвитку експертизи в управлінні мультимарним зберіганням

Ці застосування демонструють практичну цінність онтологічної моделі у вирішенні викликів розподіленого зберігання та доступу до даних у мультимарних середовищах.

2.4 Математична модель зберігання даних у мультимарних середовищах

Ефективне управління даними у кількох хмарних провайдерів вимагає не лише концептуального розуміння домену, як це забезпечено онтологічною моделлю, але також математичної основи для прийняття оптимальних рішень. У цьому розділі представлена комплексна математична модель для оптимізації зберігання даних у мультимарних середовищах, спираючись на критерії та концепції, встановлені в попередніх розділах.

2.4.1 Формулювання проблеми

Проблема ефективного зберігання даних у мультимарному середовищі розглядається, як задача визначення оптимального розміщення, реплікації та управління даними у кількох хмарних провайдерів для задоволення вимог продуктивності, надійності, безпеки та вартості. У цьому розділі встановлюються математичні позначення та структура проблеми, які будуть використовуватися протягом усієї моделі.

Множини та індекси:

Визначимо наступні множини та індекси:

- $D = 1, 2, \dots, d$: Множина об'єктів даних, які потрібно зберігати;
- $CP = 1, 2, \dots, cp$: Множина хмарних провайдерів;
- $S = 1, 2, \dots, s$: Множина послуг зберігання, що пропонуються провайдерами;

- $R = 1, 2, \dots, r$: Множина регіонів, де доступні послуги зберігання;
- $T = 1, 2, \dots, t$: Множина часових періодів у горизонті планування;

Для зручності визначимо наступні складені індекси:

- (i, j, k) : Відноситься до послуги зберігання j , що пропонується провайдером i в регіоні k ;
- (l, m) : Відноситься до об'єкта даних l з класифікацією m .

Змінні рішення:

Основними змінними рішення в запропонованій моделі є:

- x_{ijkl} : Бінарна змінна, що вказує, чи зберігається об'єкт даних l в послугі зберігання j , що пропонується провайдером i в регіоні k ;
- u_{ijkl} : Неперервна змінна, що представляє частку об'єкта даних l , що зберігається в послугі зберігання j , що пропонується провайдером i в регіоні k (для надлишкового кодування або часткового розміщення);
- $z_{ll'}$: Бінарна змінна, що вказує, чи зберігаються об'єкти даних l та l' разом
- w_{ijk} : Бінарна змінна, що вказує, чи використовується взагалі послуга зберігання j провайдера i в регіоні k .

Параметри:

Модель включає різні параметри, які представляють характеристики об'єктів даних, послуг зберігання та вимог:

Параметри, пов'язані з даними:

- s_l : Розмір об'єкта даних l в ГБ;
- a_l : Частота доступу до об'єкта даних l (операцій за одиницю часу);
- c_l : Рівень критичності об'єкта даних l (за шкалою, наприклад, 1-5);
- r_l : Необхідний рівень надлишковості для об'єкта даних l ;
- e_l : Вимога до шифрування для об'єкта даних l (бінарна або рівень);
- t_l^{\max} : Максимально допустима затримка для доступу до об'єкта даних l (мс).

Параметри провайдера та послуги:

- c_{ijk}^s : Вартість зберігання за ГБ для послуги j провайдера i в регіоні k ;
- c_{ijk}^o : Вартість операції за запит для послуги j провайдера i в регіоні k ;

- c_{ijk}^t : Вартість передачі даних за ГБ для послуги j провайдера i в регіоні k ;
- p_{ijk} : Показник продуктивності (наприклад, IOPS) для послуги j провайдера i в регіоні k ;
- a_{ijk} : Доступність послуги j провайдера i в регіоні k (відсоток);
- l_{ijk} : Середня затримка для послуги j провайдера i в регіоні k (мс);
- e_{ijk} : Можливість шифрування послуги j провайдера i в регіоні k (бінарна або рівень);
- cap_{ijk} : Ємність послуги j провайдера i в регіоні k (ГБ).

Параметри, пов'язані з мережею:

- $b_{kk'}$: Пропускна здатність між регіонами k та k' (Мбіт/с);
- $l_{kk'}$: Затримка мережі між регіонами k та k' (мс);
- $c_{kk'}^n$: Вартість передачі по мережі між регіонами k та k' за ГБ.

Параметри відповідності та управління:

- g_{mk} : Бінарний параметр, що вказує, чи можуть дані з класифікацією m зберігатися в регіоні k ;
- $comp_{im}$: Бінарний параметр, що вказує, чи відповідає провайдер i вимогам для класифікації даних m .

Структура задачі оптимізації:

Задачу оптимізації мультимарного зберігання даних можна структурувати як задачу багатоцільової оптимізації з наступними компонентами:

1. **Цільові функції:** Математичні вирази, що кількісно визначають цілі, які потрібно оптимізувати, такі як мінімізація вартості, максимізація продуктивності або максимізація надійності.
2. **Обмеження:** Математичні вирази, що представляють обмеження та вимоги, які повинні бути задоволені, такі як обмеження ємності, вимоги до продуктивності та правила відповідності.
3. **Простір рішень:** Множина всіх можливих комбінацій змінних рішення, що представляють різні стратегії розміщення даних.
4. **Метрики оцінки:** Функції, що вимірюють якість рішень з точки зору різних критеріїв, таких як економічна ефективність, затримка, доступність та безпека.

Складність цієї задачі виникає з наступних факторів: декілька, часто конфліктуючих цілей; нелінійні відношення між змінними; дискретні та неперервні змінні рішення; невизначеність у значеннях параметрів; динамічні зміни у вимогах та характеристиках провайдера.

2.4.2 Цільові функції для мультимарної оптимізації

Оптимізація зберігання даних у мультимарних середовищах включає кілька цілей, які часто конфліктують між собою. Далі визначимо математичні формулювання цих цілей, які пізніше будуть об'єднані в основу багатоцільової оптимізації.

Ціль мінімізації вартості

Загальна вартість зберігання та доступу до даних у кількох хмарних провайдерів включає витрати на зберігання, витрати на операції та витрати на передачу даних. Метою є мінімізація цієї загальної вартості:

$$\min f_{cost} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} \left(c_{ijk}^s \cdot s_l \cdot x_{ijkl} + c_{ijk}^o \cdot a_l \cdot x_{ijkl} + \sum_{k' \in R} c_{kk'}^n \cdot s_l \cdot a_l \cdot x_{ijkl} \cdot \sum_{i' \in CP} \sum_{j' \in S} x_{i'j'k'l} \right). \quad (2.70)$$

Ця цільова функція враховує:

- Витрати на зберігання на основі розміру об'єктів даних та вартості одиниці зберігання кожної послуги
- Витрати на операції на основі частоти доступу до об'єктів даних та вартості одиниці операції кожної послуги
- Витрати на передачу даних на основі розміру та частоти доступу до об'єктів даних та вартості передачі по мережі між регіонами

Для сценаріїв надлишкового кодування, де дані розділені між провайдерами, функція вартості стає:

$$\min f_{cost} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} \left(c_{ijk}^s \cdot s_l \cdot y_{ijkl} + c_{ijk}^o \cdot a_l \cdot y_{ijkl} + \sum_{k' \in R} c_{kk'}^n \cdot s_l \cdot a_l \cdot y_{ijkl} \cdot \sum_{i' \in CP} \sum_{j' \in S} y_{i'j'k'l} \right). \quad (2.71)$$

Ціль максимізації продуктивності

Ціль продуктивності спрямована на максимізацію загальної продуктивності операцій доступу до даних, враховуючи можливості продуктивності послуг зберігання та шаблони доступу до об'єктів даних:

$$\max f_{perf} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} p_{ijk} \cdot a_l \cdot x_{ijkl} \cdot \quad (2.72)$$

Ця функція зважує показник продуктивності кожної послуги зберігання за частотою доступу до об'єктів даних, що зберігаються там, пріоритезуючи високу продуктивність для даних, до яких часто звертаються.

Альтернативне формулювання, що враховує мінімізацію затримки:

$$\min f_{latency} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} l_{ijk} \cdot a_l \cdot x_{ijkl} \cdot \quad (2.73)$$

Ціль максимізації надійності

Ціль надійності зосереджена на максимізації доступності даних шляхом врахування характеристик надійності послуг зберігання та критичності об'єктів даних:

$$\max f_{reliability} = \sum_{l \in D} c_l \cdot \left(1 - \prod_{i \in CP} \prod_{j \in S} \prod_{k \in R} (1 - a_{ijk} \cdot x_{ijkl}) \right). \quad (2.74)$$

Ця функція обчислює ймовірність того, що принаймні одна копія кожного об'єкта даних доступна, зважена за критичністю об'єкта даних. Добуток представляє ймовірність того, що всі копії об'єкта даних недоступні, а віднімання від 1 дає ймовірність того, що принаймні одна копія доступна.

Для надлишкового кодування з параметрами (n, k) де k з n фрагментів потрібні для відновлення даних, функція надійності стає складнішою:

$$\max f_{reliability} = \sum_{l \in D} c_l \cdot \sum_{m=k}^n \binom{n}{m} \prod_{(i,j,k) \in F_m} a_{ijk} \prod_{(i,j,k) \in F \setminus F_m} (1 - a_{ijk}), \quad (2.75)$$

де F — це множина всіх розміщень фрагментів, а F_m — це підмножина m фрагментів.

Ціль максимізації безпеки

Ціль безпеки спрямована на максимізацію захисту даних шляхом врахування можливостей безпеки послуг зберігання та чутливості об'єктів даних:

$$\max f_{security} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} e_{ijk} \cdot e_l \cdot x_{ijkl} . \quad (2.76)$$

Ця функція винагороджує розміщення, де об'єкти даних з високими вимогами до шифрування зберігаються в послугах з сильними можливостями шифрування.

Більш комплексна ціль безпеки може включати додаткові фактори, такі як репутація провайдера щодо безпеки, сертифікати відповідності та заходи фізичної безпеки:

$$\max f_{security} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} (w_e \cdot e_{ijk} + w_r \cdot r_i + w_c \cdot c_i) \cdot e_l \cdot x_{ijkl} , \quad (2.77)$$

де w_e , w_r та w_c — це ваги для факторів шифрування, репутації та відповідності, відповідно.

Ціль максимізації відповідності

Ціль відповідності забезпечує, що рішення щодо розміщення даних дотримуються нормативних та управлінських вимог:

$$\max f_{compliance} = \sum_{l \in D} \sum_{m \in M} \delta_{lm} \cdot \left(\sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} g_{mk} \cdot comp_{im} \cdot x_{ijkl} \right) , \quad (2.78)$$

де δ_{lm} — це бінарний параметр, що вказує, чи має об'єкт даних l класифікацію m . Ця функція винагороджує розміщення, де об'єкти даних зберігаються у відповідних регіонах відповідними провайдерами.

Ціль оптимізації локальності даних

Ціль локальності даних спрямована на мінімізацію відстані між пов'язаними об'єктами даних, що може покращити продуктивність для операцій, які отримують доступ до кількох об'єктів:

$$\min f_{locality} = \sum_{l \in D} \sum_{l' \in D} rel_{ll'} \cdot \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{i' \in P} \sum_{j' \in S} \sum_{k' \in R} dist_{kk'} \cdot x_{ijkl} \cdot x_{i'j'k'l'} , \quad (2.79)$$

де $rel_{ll'}$ представляє силу зв'язку між об'єктами даних l та l' , а $dist_{kk'}$ представляє відстань (фізичну або мережеву) між регіонами k та k' .

Ціль мінімізації впливу на навколишнє середовище

Оскільки сталість стає все більш важливою, можна визначити цільову функцію для мінімізації впливу зберігання даних на навколишнє середовище:

$$\min f_{env} = \sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} e_{ijk}^{env} \cdot s_l \cdot x_{ijkl} , \quad (2.80)$$

де e_{ijk}^{env} представляє фактор впливу на навколишнє середовище (наприклад, викиди вуглецю на ГБ) для послуги зберігання j провайдера i в регіоні k .

2.4.3 Формулювання обмежень

Оптимізація зберігання даних у мультимарних середовищах підлягає різним обмеженням, які забезпечують здійсненність та практичність рішень. У цьому розділі формулюються ці обмеження математично.

Обмеження ємності зберігання

Загальний розмір об'єктів даних, призначених кожній послугі зберігання, не може перевищувати її ємність:

$$\sum_{l \in D} s_l \cdot x_{ijkl} \leq cap_{ijk} \quad \forall i \in P, j \in S, k \in R . \quad (2.81)$$

Для сценаріїв надлишкового кодування:

$$\sum_{l \in D} s_l \cdot y_{ijkl} \leq cap_{ijk} \quad \forall i \in P, j \in S, k \in R . \quad (2.82)$$

Обмеження розміщення даних

Кожен об'єкт даних повинен зберігатися принаймні в одному місці (або більше, залежно від вимог надлишковості):

$$\sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} x_{ijkl} \geq r_l \quad \forall l \in D. \quad (2.83)$$

Для надлишкового кодування з параметрами (n, k) :

$$\sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} y_{ijkl} = n \quad \forall l \in D. \quad (2.84)$$

Обмеження продуктивності

Затримка для доступу до кожного об'єкта даних не повинна перевищувати його максимально допустиму затримку:

$$l_{ijk} \cdot x_{ijkl} \leq t_l^{max} \cdot x_{ijkl} \quad \forall i \in CP, j \in S, k \in R, l \in D. \quad (2.85)$$

Це обмеження забезпечує, що об'єкти даних розміщуються лише в послугах зберігання, які можуть задовольнити їх вимоги до затримки.

Обмеження надійності

Доступність кожного об'єкта даних повинна відповідати або перевищувати його необхідний рівень:

$$1 - \prod_{i \in CP} \prod_{j \in S} \prod_{k \in R} (1 - a_{ijk} \cdot x_{ijkl}) \geq a_l^{min} \quad \forall l \in D, \quad (2.86)$$

де a_l^{min} — це мінімальна необхідна доступність для об'єкта даних l .

Обмеження безпеки

Об'єкти даних з вимогами до шифрування повинні зберігатися в послугах, що забезпечують адекватне шифрування:

$$e_{ijk} \cdot x_{ijkl} \geq e_l \cdot x_{ijkl} \quad \forall i \in CP, j \in S, k \in R, l \in D. \quad (2.87)$$

Це обмеження забезпечує, що об'єкти даних розміщуються лише в послугах зберігання, які відповідають або перевищують їх вимоги до шифрування.

Обмеження відповідності

Об'єкти даних повинні зберігатися в регіонах та провайдерами, які відповідають їх нормативним вимогам:

$$x_{ijkl} \leq \sum_{m \in M} \delta_{lm} \cdot g_{mk} \cdot comp_{im} \quad \forall i \in CP, j \in S, k \in R, l \in D. \quad (2.88)$$

Це обмеження запобігає зберіганням об'єктів даних у невідповідних регіонах або невідповідними провайдерами.

Бюджетні обмеження

Загальна вартість рішення зберігання не повинна перевищувати доступний бюджет:

$$\sum_{i \in CP} \sum_{j \in S} \sum_{k \in R} \sum_{l \in D} (c_{ijk}^s \cdot s_l \cdot x_{ijkl} + c_{ijk}^o \cdot a_l \cdot x_{ijkl} + \sum_{k' \in R} c_{kk'}^n \cdot s_l \cdot a_l \cdot x_{ijkl} \cdot \sum_{i' \in CP} \sum_{j' \in S} x_{i'j'k'l'}) \leq B, \quad (2.89)$$

де B — це загальний доступний бюджет для зберігання даних.

Обмеження різноманітності провайдерів

Щоб уникнути надмірної залежності від одного провайдера, можна додати обмеження для забезпечення різноманітності:

$$\sum_{j \in S} \sum_{k \in R} x_{ijkl} \leq d_l^{\max} \quad \forall i \in CP, l \in D. \quad (2.90)$$

де d_l^{\max} — це максимальна кількість копій об'єкта даних l , які можуть зберігатися у одного провайдера.

Обмеження спільного розміщення

Деякі об'єкти даних можуть потребувати зберігання разом або окремо:

$$\begin{aligned} x_{ijkl} &= x_{ijkl'} \quad \forall i \in CP, j \in S, k \in R, (l, l') \in C_{together} \\ x_{ijkl} + x_{ijkl'} &\leq 1 \quad \forall i \in CP, j \in S, k \in R, (l, l') \in C_{separate}, \end{aligned} \quad (2.91)$$

де $C_{together}$ та $C_{separate}$ — це множини пар об'єктів даних, які повинні зберігатися разом або окремо, відповідно.

Обмеження використання послуг

Якщо послуга зберігання використовується взагалі, можуть застосовуватися фіксовані витрати:

$$\begin{aligned} x_{ijkl} &\leq w_{ijk} \quad \forall i \in CP, j \in S, k \in R, l \in D \\ \sum_{l \in D} x_{ijkl} &\geq w_{ijk} \quad \forall i \in CP, j \in S, k \in R, \end{aligned} \quad (2.92)$$

де w_{ijk} – це бінарна змінна, що вказує, чи використовується послуга зберігання j провайдера i в регіоні k .

2.4.4 Підхід до багатоцільової оптимізації

Проблема оптимізації мультимарного зберігання даних включає кілька, часто конфліктуючих цілей. Далі представлені підходи до обробки цих кількох цілей для знаходження рішень, які представляють хороші компроміси.

Метод зваженої суми

Один поширений підхід до багатоцільової оптимізації – це метод зваженої суми, який об'єднує окремі цільові функції в одну ціль, використовуючи ваги, що відображають їх відносну важливість:

$$\begin{aligned} \min f_{combined} \\ = w_{cost} \cdot f_{cost} + w_{perf} \cdot (-f_{perf}) + w_{rel} \cdot (-f_{reliability}) \\ + w_{sec} \cdot (-f_{security}) + w_{comp} \cdot (-f_{compliance}) + w_{loc} \\ \cdot f_{locality} + w_{env} \cdot f_{env}, \end{aligned} \quad (2.93)$$

де w_{cost} , w_{perf} , w_{rel} , w_{sec} , w_{comp} , w_{loc} та w_{env} – це невід'ємні ваги, сума яких дорівнює 1. Зауважимо, що цілі, які спочатку є проблемами максимізації, перетворюються на мінімізацію шляхом їх заперечення.

Виклик з цим підходом полягає у визначенні відповідних ваг, оскільки вони значно впливають на отримане рішення. Аналіз чутливості можна виконати, змінюючи ваги та спостерігаючи, як змінюються рішення.

Метод ε -обмеження

Метод ε -обмеження оптимізує одну ціль, обмежуючи інші в межах прийнятних лімітів:

$$\min f_{cost}. \quad (2.94)$$

За умови:

$$f_{perf} \geq \varepsilon_{perf} f_{reliability} \geq \varepsilon_{rel} f_{security} \geq \varepsilon_{sec} f_{compliance} \geq \varepsilon_{comp} f_{locality} \leq \varepsilon_{loc} f_{env} \leq \varepsilon_{env},$$

де ε_{perf} , ε_{rel} , ε_{sec} , ε_{comp} , ε_{loc} та ε_{env} – це порогові значення для відповідних цілей.

Систематично змінюючи ці порогові значення, можна згенерувати набір Парето-оптимальних рішень, кожне з яких представляє різний компроміс між цілями.

Цільове програмування

Цільове програмування прагне мінімізувати відхилення від вказаних цільових значень для кожної цілі:

$$\min \sum_{i=1}^n (w_i^+ \cdot d_i^+ + w_i^- \cdot d_i^-). \quad (2.95)$$

За умови: $f_i(x) + d_i^- - d_i^+ = g_i \quad \forall i \in 1, 2, \dots, n, d_i^+, d_i^- \geq 0 \quad \forall i \in 1, 2, \dots, n$.

Де:

- $f_i(x)$ — це i -та цільова функція
- g_i — це цільове значення для i -тої цілі
- d_i^+ та d_i^- — це позитивні та негативні відхилення від цілі
- w_i^+ та w_i^- — це ваги для позитивних та негативних відхилень

Цей підхід особливо корисний, коли конкретні цільові значення для кожної цілі можуть бути визначені на основі вимог або переваг.

Оптимізація за Парето

Оптимізація за Парето спрямована на знаходження набору недомінованих рішень, де жодне рішення не є кращим за інше за всіма цілями. Рішення x домінує над іншим рішенням y , якщо:

1. x принаймні таке ж хороше, як y , за всіма цілями
2. x строго краще, ніж y , принаймні за однією ціллю

Набір усіх недомінованих рішень формує фронт Парето, який представляє поверхню компромісу між різними цілями.

Різні алгоритми можуть бути використані для наближення фронту Парето, включаючи:

- багатоцільові еволюційні алгоритми (NSGA-II, SPEA2);
- багатоцільова оптимізація рою частинок;
- багатоцільовий імітаційний відпал.

Ці алгоритми генерують набір рішень, які наближають фронт Парето, дозволяючи особам, що приймають рішення, вибрати рішення, яке найкраще відповідає їхнім перевагам.

Ієрархічна оптимізація

У деяких випадках цілі можуть бути пріоритезовані ієрархічним чином, де більш важливі цілі оптимізуються спочатку, а менш важливі цілі оптимізуються за умови збереження оптимального значення цілей вищого пріоритету:

1. Розв'язати: $\min f_1(x)$ за умови оригінальних обмежень, отримуючи оптимальне значення f_1^*
2. Розв'язати: $\min f_2(x)$ за умови оригінальних обмежень та $f_1(x) = f_1^*$

3. Продовжити для решти цілей

Цей підхід підходить, коли існує чітке впорядкування пріоритетів серед цілей, наприклад, коли відповідність є безкомпромісною, за нею слідує безпека, надійність, а потім вартість.

ВИСНОВКИ

1. Розроблено комплексну систему критеріїв оцінювання ефективності розподіленого зберігання даних у мультимарному середовищі, яка охоплює показники продуктивності, надійності, безпеки, економічної ефективності, інтегрованих та екологічної сталості. Для кожного критерію визначено відповідні метрики, одиниці вимірювання та математичні вирази, що дозволяють здійснювати кількісну оцінку різних аспектів функціонування системи.
2. Запропоновано онтологічну модель процесу розподіленого зберігання даних у мультимарному середовищі, яка формалізує структуру доменних знань та відображає ключові сутності, їх атрибути та взаємозв'язки. Розроблена онтологія забезпечує єдине концептуальне представлення складної взаємодії між даними, ресурсами зберігання, хмарними провайдерами та політиками управління, що створює теоретичну основу для подальшої оптимізації процесів зберігання та доступу до даних.
3. Сформульовано математичну модель розміщення даних у мультимарному середовищі як задачу багатокритеріальної оптимізації з цільовими функціями, що відображають вартість, продуктивність, надійність, безпеку, відповідність регуляторним вимогам, локальність даних та екологічний вплив. Визначено систему обмежень, що забезпечують практичну реалізованість отриманих рішень, та запропоновано методи розв'язання багатокритеріальної задачі, включаючи метод зваженої суми, ϵ -обмежень, цільового програмування, Парето-оптимізації та ієрархічної оптимізації.
4. Досліджено методи розв'язання сформульованої задачі оптимізації, включаючи точні методи (змішане цілочисельне лінійне програмування), евристичні та метаевристичні алгоритми, методи декомпозиції та підходи на основі машинного навчання. Проведено порівняльний аналіз цих методів з точки зору їх обчислювальної складності, точності та масштабованості, що дозволяє обирати найбільш відповідний метод залежно від конкретних умов застосування.
5. Запропонований комплексний підхід поєднує якісний аналіз з кількісною оптимізацією, що дозволяє організаціям приймати обґрунтовані рішення щодо розподіленого зберігання даних, які балансують між кількома конкуруючими цілями та задовольняють різноманітні обмеження.

РОЗДІЛ 3

КОМПЛЕКСНИЙ ПІДХІД ЩОДО ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ

3.1 Гібридна структура підкріплювального навчання та багатоцільових еволюційних алгоритмів

Складність розподіленого зберігання даних у мультихмарному середовищі вимагає підходу, який може ефективно оперувати у багатовимірному просторі рішень, адаптуючись при цьому до змінних умов. [55,57] Далі в цьому розділі запропоновано новий підхід, який поєднує адаптивні можливості навчання з підкріплення (Reinforcement Learning - RL) з характеристиками оптимізації багатоцільових еволюційних алгоритмів (Multi Objective Evolutionary Algorithms – MOEA). Ця інтеграція створює систему, здатну приймати більш оптимальні рішення щодо розміщення даних серед кількох хмарних провайдерів, враховуючи комплексний набір критеріїв, встановлених у Розділі 2.

3.1.1 Формулювання простору станів та дій

Компонент навчання з підкріплення у запропонованому підході вимагає ретельної розробки простору станів та дій для ефективного опису складності рішень щодо зберігання даних у мультихмарному середовищі. Простір станів повинен представляти поточну конфігурацію розподіленої системи зберігання, тоді як простір дій повинен охоплювати всі можливі рішення щодо розміщення та управління даними.

Простір станів S можна визначити як багатовимірний вектор, який фіксує поточний стан об'єктів даних та ресурсів зберігання серед усіх наявних хмарних провайдерів:

$$S = s_1, s_2, \dots, s_n. \quad (3.1)$$

Де кожен стан s_i представлений як:

$$s_i = (D, CP, P, U, T). \quad (3.1_1)$$

У цьому виразі $D = d_1, d_2, \dots, d_m$ представляє набір об'єктів даних у системі; $CP = cp_1, cp_2, \dots, cp_k$ позначає набір доступних хмарних провайдерів; $P = p_{ij} \in$

матрицею, що вказує на розміщення об'єкта даних d_i у хмарного провайдера cp_j ; $U = u_1, u_2, \dots, u_m$ фіксує шаблони використання та частоту доступу до кожного об'єкта даних; $T = t_1, t_2, \dots, t_m$ представляє часозалежні характеристики об'єктів даних.

Простір дій A охоплює всі можливі операції, які можна виконувати з об'єктами даних серед наявних хмарних провайдерів:

$$A = a_1, a_2, \dots, a_q. \quad (3.2)$$

Де кожна дія a_i може бути однією з наступних операцій:

1. **Розміщення:** $Place(d_i, cp_j)$ - розмістити об'єкт даних d_i у хмарного провайдера cp_j .
2. **Міграція:** $Migrate(d_i, cp_j, cp_k)$ - перемістити об'єкт даних d_i від провайдера cp_j до провайдера cp_k .
3. **Реплікація:** $Replicate(d_i, cp_j, cp_k)$ - створити репліку об'єкта даних d_i від провайдера cp_j до провайдера cp_k .
4. **Видалення:** $Delete(d_i, cp_j)$ - видалити об'єкт даних d_i від провайдера cp_j .

Розмірність просторів станів та дій зростає з кількістю об'єктів даних та хмарних провайдерів, створюючи складне середовище прийняття рішень. Для управління цією складністю запропоновано ієрархічне представлення станів, яке агрегує подібні об'єкти даних на основі їхніх характеристик та вимог. З цього випливає, що такий підхід зменшує простір станів, зберігаючи при цьому важливу інформацію, необхідну для прийняття рішень.

Функція переходу $T(s, a, s')$ визначає ймовірність переходу зі стану s до стану s' після виконання дії a . У такому контексті ця функція переходу включає стохастичну природу хмарних середовищ, включаючи варіації мережеских умов, продуктивності провайдерів та доступності ресурсів:

$$T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a). \quad (3.3)$$

Функція переходу заздалегідь явно не відома, але поступово вивчається через взаємодію з середовищем. Тобто, такий процес навчання дозволяє системі адаптуватися до динамічної природи мультихмарних середовищ та покращувати прийняття рішень з часом.

3.1.2 Розробка функції винагороди на основі багатокритеріальної оптимізації

Функція винагороди є критичним компонентом запропонованого підходу навчання з підкріпленням, оскільки він спрямовує процес навчання до оптимальних рішень щодо розміщення даних. При цьому, потрібно враховувати багатокритеріальну природу визначеної задачі оптимізації, щоб функція винагороди балансувала між кількома, часто суперечливими цілями.

Визначимо композитну функцію винагороди, яка включає всі критерії, які було визначено та описано в Розділі 2:

$$R(s, a, s') = \sum_{i=1}^n w_i \cdot r_i(s, a, s'), \quad (3.4)$$

де $r_i(s, a, s')$ є компонентом винагороди для критерію i ; w_i є вагою, призначеною критерію i , з $\sum_{i=1}^n w_i = 1$.

Окремі компоненти винагороди формулюються наступним чином:

1. **Винагорода за затримку (r_{lat}):** $r_{lat}(s, a, s') = \frac{L_{max} - L(s')}{L_{max} - L_{min}}$, де $L(s')$ є середньою затримкою в стані s' , а L_{max} та L_{min} є максимальними та мінімальними можливими значеннями затримки.

2. **Винагорода за економічну ефективність (r_{cost}):** $r_{cost}(s, a, s') = \frac{C_{max} - C(s')}{C_{max} - C_{min}}$ де $C(s')$ є загальною вартістю в стані s' .

3. **Винагорода за доступність (r_{avail}):** $r_{avail}(s, a, s') = \frac{A(s') - A_{min}}{A_{max} - A_{min}}$ де $A(s')$ є рівнем доступності в стані s' .

4. **Винагорода за безпеку (r_{sec}):** $r_{sec}(s, a, s') = \frac{S(s') - S_{min}}{S_{max} - S_{min}}$ де $S(s')$ представляє рівень безпеки в стані s' .

5. **Винагорода за відповідність (r_{comp}):**

$$r_{comp}(s, a, s') = \begin{cases} 1, & \text{"якщо всі вимоги відповідності виконані"} \\ -1, & \text{"в іншому випадку"} \end{cases}$$

6. **Винагорода за енергоефективність (r_{energy}):** $r_{energy}(s, a, s') = \frac{E_{max} - E(s')}{E_{max} - E_{min}}$ де $E(s')$ є споживанням енергії в стані s' .

Ваги w_i не є фіксованими, але динамічно коригуються на основі конкретних вимог різних типів даних та контекстів застосування. Цей адаптивний механізм

зважування дозволяє системі пріоритезувати різні критерії на основі поточних операційних цілей та обмежень.

Для вирішення проблеми відкладених винагород у мультихмарних середовищах включається підхід навчання з часовими різницями, який враховує як негайні, так і майбутні винагороди:

$$R_{total}(s, a) = R(s, a, s') + \gamma \cdot \max_{a'} Q(s', a'), \quad (3.4)$$

де γ є коефіцієнтом дисконтування, який визначає важливість майбутніх винагород; $Q(s', a')$ є очікуваною майбутньою винагородою за виконання дії a' в стані s' .

Це формулювання дозволяє системі приймати рішення, які оптимізують як короткострокові, так і довгострокові цілі, створюючи більш збалансовану та ефективну стратегію розміщення даних.

3.1.3 Механізм покращення політики на основі МОЕА

У той час як навчання з підкріпленням забезпечує адаптивні можливості прийняття рішень, багатоцільові еволюційні алгоритми (МОЕА) відмінно знаходять Парето-оптимальні рішення у складних задачах багатокритеріальної оптимізації. Запропонований підхід використовує цю особливість, включаючи механізм покращення політики на основі МОЕА, який підвищує політику RL через еволюційну оптимізацію.

Компонент МОЕА працює з популяцією кандидатних політик, кожна з яких представляє різну стратегію для розміщення та управління даними. Еволюційний процес включає наступні кроки:

1. **Ініціалізація:** Генерація початкової популяції політик $P = \pi_1, \pi_2, \dots, \pi_p$, де кожна політика π_i відображає стани на дії: $\pi_i: S \rightarrow A$.
2. **Оцінка:** Оцінка кожної політики π_i на основі кількох цілей, отриманих з запропонованого набору критеріїв: $f_j(\pi_i) = E[r_j(s, \pi_i(s), s') | s \sim D]$ де D є розподілом станів, з якими стикаються під час роботи.
3. **Відбір:** Вибір політик для репродукції на основі їх Парето-домінування та різноманітності: $P_{selected} = Selection(P, f_1, f_2, \dots, f_n)$.
4. **Схрещування:** Генерація нових політик шляхом комбінування вибраних політик: $\pi_{new} = Crossover(\pi_i, \pi_j)$.
5. **Мутація:** Введення випадкових варіацій для підтримки різноманітності: $\pi_{mutated} = Mutate(\pi_{new})$.

6. **Заміна:** Оновлення популяції новим поколінням: $P_{t+1} = \text{Replace}(P_t, P_{\text{new}})$.

Процес МОЕА генерує Парето-фронт неможованих політик, кожна з яких представляє різний компроміс між критеріями оптимізації. Компонент RL потім вибирає найбільш відповідну політику на основі поточного операційного контексту та використовує її для керування прийняттям рішень.

Інтеграція МОЕА та RL досягається завдяки забезпеченню двостороннього потоку інформації:

1. Компонент RL надає МОЕА дані про переходи станів та сигнали винагороди, які використовуються для оцінки кандидатів політик.
2. МОЕА надає компоненту RL покращені політики, які керують дослідженням та експлуатацією в просторі станів-дій.

Такі взаємодія формує ефективну систему оптимізації, що поєднує адаптивні властивості навчання з підкріпленням із перевагами глобальної оптимізації багатоцільових еволюційних алгоритмів.

3.1.4 Механізми адаптації та навчання

Ефективність запропонованого підходу залежить від його здатності адаптуватися до змінних умов та вчитися на досвіді. Запропоновано впровадження кількох механізмів адаптації та навчання для підвищення продуктивності системи з часом.

Повторення досвіду: Для покращення ефективності вибірки та розриву кореляцій у послідовності спостережень використовується механізм повторення досвіду, який зберігає переходи (s, a, r, s') у буфері повторення D . Під час навчання міні-пакети переходів випадково вибираються з цього буфера:

$$(s_j, a_j, r_j, s'_j)_{j=1}^B \sim D. \quad (3.5)$$

Q-функція потім оновлюється з використанням цих вибраних переходів:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot [r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)], \quad (3.6)$$

де α є швидкістю навчання.

Пріоритетне повторення досвіду: Для зосередження навчання на найбільш інформативних переходах впроваджується пріоритетне повторення досвіду, де переходи вибираються з ймовірністю:

$$P(j) = \frac{p_j^\alpha}{\sum_k p_k^\alpha}, \quad (3.7)$$

де $p_j = |\delta_j| + \epsilon$ є пріоритетом переходу j , з δ_j як помилкою часової різниці та ϵ малою константою для забезпечення ненульової ймовірності.

Трансферне навчання: Для прискорення навчання в нових середовищах або для нових типів даних впроваджується механізм трансферного навчання, який використовує знання, отримані з попереднього досвіду:

$$Q_{new}(s, a) = \beta \cdot Q_{old}(s, a) + (1 - \beta) \cdot Q_{init}(s, a), \quad (3.8)$$

де β контролює ступінь передачі знань.

Контекстні бандити: Для сценаріїв з обмеженою інформацією про стан або короткостроковими цілями оптимізації використовується підхід контекстних бандитів, який вибирає дії на основі поточного контексту x :

$$a^* = \arg \max_a Q(x, a). \quad (3.9)$$

Q-функція оновлюється з використанням підходу на основі градієнта:

$$\nabla_\theta Q_\theta(x, a) = E[(r - Q_\theta(x, a)) \cdot \nabla_\theta Q_\theta(x, a)]. \quad (3.9_1)$$

Мета-навчання: Для покращення адаптації до нових середовищ впроваджується підхід мета-навчання, який оптимізує сам процес навчання:

$$\theta^* = \arg \min_{\theta \in T \sim p(T)} [\mathcal{L}_T(f\theta)], \quad (3.10)$$

де T представляє завдання (наприклад, оптимізація зберігання для конкретного типу даних), $p(T)$ є розподілом завдань, а \mathcal{L}_T є функцією втрат для завдання T .

Ці механізми адаптації та навчання працюють разом для створення надійного та гнучкого підходу, який може ефективно орієнтуватися у складному просторі рішень оптимізації зберігання та доступу до даних у мультихмарному середовищі.

3.2 Алгоритм оптимізації розміщення та доступу до даних

Спираючись на запропонований метод навчання з підкріпленням та МОЕА, описаний раніше, далі представлено алгоритм для оптимізації розміщення та

доступу до даних у мультимедійних середовищах. Цей алгоритм інтегрує теоретичні основи, встановлені раніше, з практичними ідеями для створення рішення для ефективного розподіленого зберігання та доступу до даних.

3.2.1 Початкова класифікація даних та аналіз вимог

Процес оптимізації починається з ретельного аналізу характеристик та вимог даних. Цей початковий крок є вирішальним для встановлення базових параметрів, які керуватимуть подальшими рішеннями оптимізації.

Процес класифікації даних включає категоризацію об'єктів даних на основі кількох вимірів, включаючи:

1. **Рівень чутливості:** Дані класифікуються відповідно до їх чутливості та вимог конфіденційності, від публічних до високо конфіденційних. Ця класифікація формалізується як:

$$L(d) = SL_1, SL_2, SL_3, SL_4, SL_5, \quad (3.11)$$

де SL_1 представляє публічні дані, а SL_5 представляє високо конфіденційні дані.

2. **Шаблон доступу:** Об'єкти даних категоризуються на основі їх типових шаблонів доступу, які впливають на рішення щодо розміщення:

$$AP(d) = (f_r, f_w, p_s, p_r), \quad (3.12)$$

де f_r є частотою читання, f_w є частотою запису, p_s є ймовірністю послідовного доступу, а p_r є ймовірністю випадкового доступу.

3. **Розмір та швидкість росту:** Поточний розмір та прогнозоване зростання об'єктів даних моделюються як:

$$S(d, t) = S_0(d) \cdot (1 + g_d)^t, \quad (3.13)$$

де $S_0(d)$ є початковим розміром, g_d є швидкістю росту, а t є періодом часу.

4. **Стадія життєвого циклу:** Кожному об'єкту даних призначається стадія життєвого циклу, яка впливає на його вимоги до зберігання:

$$LC(d) = Creation, Active, Inactive, Archive, Deletion. \quad (3.14)$$

5. **Вимоги відповідності:** Регуляторні та управлінські вимоги відображаються на кожен об'єкт даних:

$$CR(d) = cr_1, cr_2, \dots, cr_n. \quad (3.15)$$

Де кожен cr_i представляє конкретну вимогу відповідності.

Фаза аналізу вимог перекладає ці класифікації у конкретні технічні вимоги для кожного критерію в запропонованому комплексному наборі. Цей переклад виконується з використанням функції відображення вимог:

$$RM: (SL, AP, S, LC, CR) \rightarrow (Lat_{req}, Red_{req}, Cons_{req}, \dots, Sust_{req}), \quad (3.16)$$

де кожен компонент праворуч представляє рівень вимог для конкретного критерію з визначеного комплексного набору.

Щоб формалізувати даний вираз, для кожного об'єкта даних d визнається вектор вимог R_d :

$$R_d = (r_1, r_2, \dots, r_n), \quad (3.17)$$

де кожен r_i представляє рівень вимог для критерію i .

Рівні вимог не являються бінарними, але їх можна представити на безперервній шкалі, що дозволяє в свою чергу представлення вимог з урахуванням різних нюансів. Наприклад, вимоги до затримки можуть бути виражені як:

$$Lat_{\{req\}}(d) = \begin{cases} < 10 \text{ мс, якщо } AP(d) \text{ вказує на доступ у реальному часі} \\ < 100 \text{ мс, якщо } AP(d) \text{ вказує на інтерактивний доступ} \\ < 1000 \text{ мс, якщо } AP(d) \text{ вказує на пакетну обробку} \end{cases}.$$

Процес початкової класифікації даних та аналізу вимог розроблений для періодичного оновлення, тобто не є статичним, оскільки характеристики та вимоги даних еволюціонують із часом. Цей динамічний підхід забезпечує, що процес оптимізації залишається узгодженим зі змінними бізнес-потребами та технічними обмеженнями.

Результатом цієї фази є комплексний каталог даних, який включає:

1. Метадані про кожен об'єкт даних
2. Інформацію про класифікацію за кількома вимірами
3. Конкретні вимоги для кожного критерію в визначеному комплексному наборі
4. Залежності між об'єктами даних
5. Історичні шаблони доступу та статистику використання

Цей каталог служить основою для наступних фаз оптимізації, надаючи при цьому необхідний контекст для прийняття інформованих рішень щодо розміщення та доступу до даних.

3.2.2 Динамічна стратегія розміщення з використанням навчання з підкріпленням

Наступним кроком є впровадження динамічної стратегії розміщення, яка спирається на початкову класифікацію даних та аналіз вимог, і яка використовує навчання з підкріпленням для прийняття адаптивних рішень щодо розміщення даних. Ця стратегія постійно оптимізує розміщення об'єктів даних серед кількох хмарних провайдерів на основі змінних вимог, шаблонів доступу та умов середовища.

Ядром запропонованої динамічної стратегії розміщення є алгоритм Q-навчання, який вивчає та знаходить оптимальну політику розміщення даних через взаємодію з мультихмарним середовищем. Q-функція $Q(s, a)$ представляє очікувану довгострокову винагороду за виконання дії a в стані s і оновлюється відповідно до:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot \left[r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a) \right]. \quad (3.18)$$

Для обробки високовимірних просторів станів та дій у мультихмарних середовищах впроваджується глибока Q-мережа (DQN), яка апроксимує Q-функцію з використанням нейронної мережі:

$$Q(s, a; \theta) \approx Q^*(s, a), \quad (3.19)$$

де θ представляє параметри нейронної мережі.

Мережа навчається мінімізувати функцію втрат:

$$L(\theta) = E(s, a, r, s') \sim D \left[\left(r + \gamma \cdot \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]. \quad (3.20)$$

де θ^- представляє параметри цільової мережі, яка періодично оновлюється для стабілізації навчання.

Динамічна стратегія розміщення даних працює в двох режимах:

1. **Режим дослідження:** Під час цієї фази система досліджує різні варіанти розміщення для збору інформації про їх характеристики продуктивності. Дослідження керується ϵ -жадібною політикою:

$$\pi(a|s) = \begin{cases} \arg \max_a Q(s, a; \theta), & \text{з ймовірністю } 1 \\ -\epsilon \text{ випадкова дія,} & \text{з ймовірністю } \epsilon \end{cases}.$$

Швидкість дослідження ϵ поступово зменшується з часом відповідно до:

$$\epsilon(t) = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \cdot e^{-\lambda t}.$$

2. **Режим експлуатації:** У цьому режимі система використовує вивчену політику для прийняття оптимальних рішень щодо розміщення:

$$a^* = \arg \max_a Q(s, a; \theta). \quad (3.21)$$

Зміна або перехід між цими режимами відбувається за допомогою функції планування, яка врівноважує процеси дослідження та використання наявних знань, враховуючи рівень достовірності системи щодо її поточної стратегії та стабільності середовища.

Для включення багатокритеріальної природи даної задачі оптимізації впроваджується декомпозований підхід Q-навчання, який підтримує окремі Q-функції для кожного критерію:

$$Q_i(s, a; \theta_i) \approx Q_i^*(s, a). \quad (3.22)$$

Загальна Q-функція потім обчислюється як зважена сума:

$$Q(s, a; \theta) = \sum_{i=1}^n w_i \cdot Q_i(s, a; \theta_i), \quad (3.22)$$

де ваги w_i динамічно коригуються на основі поточних визначених пріоритетів та обмежень.

Динамічна стратегія розміщення також включає компонент прогнозування, який передбачає майбутні шаблони доступу та вимоги:

$$P(AP_{t+1} | AP_t, AP_{t-1}, \dots, AP_{t-k}). \quad (3.23)$$

Дана здатність до прогнозування дозволяє системі заздалегідь коригувати розміщення даних відповідно до очікуваних змін у схемах доступу, що сприяє зменшенню затримки та підвищенню загальної ефективності роботи.

Рішення щодо розміщення не приймаються ізольовано, але враховують взаємозалежності між об'єктами даних. Далі моделюються ці залежності з використанням графової структури:

$$G = (V, E), \quad (3.24)$$

де вершини V представляють об'єкти даних, а ребра E представляють залежності між ними. Алгоритм розміщення враховує цей граф при прийнятті рішень, забезпечуючи, що пов'язані об'єкти даних розміщуються таким чином, щоб мінімізувати доступ між провайдерами.

Динамічна стратегія розміщення реалізується як безперервний процес, який моніторить середовище, оновлює свою політику та відповідно коригує розміщення даних. Цей підхід безперервної оптимізації забезпечує, що система залишається чутливою до змінних умов та вимог.

3.2.3 Багатоцільова оптимізація параметрів зберігання

У той час як динамічна стратегія розміщення визначає оптимального хмарного провайдера для кожного об'єкта даних, багатоцільова оптимізація параметрів зберігання зосереджується на налаштуванні конкретних характеристик зберігання в межах кожного провайдера. Цей процес оптимізації охоплює такі параметри, як фактор реплікації, рівень узгодженості, метод шифрування, коефіцієнт стиснення та рівень зберігання.

Задача багатоцільової оптимізації формулюється як:

$$\min_x F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T, \quad (3.25)$$

за умови:

$$g_j(x) \leq 0, j = 1, 2, \dots, h_k(x) = 0, k = 1, 2, \dots, x_i^L \leq x_i \leq x_i^U, i = 1, 2, \dots,$$

де x – це вектор змінних рішення, що представляють параметри зберігання, $F(x)$ – вектор цільових функцій, $g_j(x)$ та $h_k(x)$ – нерівності та рівності обмежень, а x_i^L та x_i^U є нижніми та верхніми межами змінних для кожного варіанту рішення.

Щоб вирішити цю задачу багатоцільової оптимізації пропонується впровадження Недомінованого Сортувального Генетичного Алгоритму II (NSGA-II), який добре підходить для знаходження Парето-оптимальних рішень у задачах зі складними просторами рішень. Алгоритм працює наступним чином:

1. **Ініціалізація:** Генерація початкової популяції рішень P_0 розміром N .

2. **Оцінка:** Обчислення значень цільової функції для кожного рішення в популяції.
3. **Недоміноване сортування:** Класифікація популяції на різні рівні недомінування (фронти Парето).
4. **Відстань скупчення:** Розрахунок відстані скупчення для кожного рішення для підтримки різноманітності.
5. **Відбір:** Вибір батьківських рішень з використанням бінарного турнірного відбору на основі рангу та відстані скупчення.
6. **Варіація:** Застосування операторів схрещування та мутації для генерації популяції нащадків Q_t .
7. **Комбінація:** Об'єднання батьківської популяції та популяції нащадків: $R_t = P_t \cup Q_t$.
8. **Елітизм:** Вибір найкращих N рішень з R_t для формування наступного покоління P_{t+1} .
9. **Завершення:** Повторення кроків 2-8 до досягнення критерію завершення.

Цільові функції в запропонованій багатоцільовій оптимізації отримані з критеріїв, встановлених у Розділі 2. Наприклад:

1. **Цільова функція вартості:** $f_{cost}(x) = \sum_{i=1}^n \sum_{j=1}^k c_{ij} \cdot x_{ij}$ де c_{ij} є вартістю зберігання об'єкта даних i з конфігурацією параметрів j .
2. **Цільова функція затримки:** $f_{latency}(x) = \max_{i,j} (l_{ij} \cdot x_{ij})$ де l_{ij} є затримкою, пов'язаною з доступом до об'єкта даних i з конфігурацією параметрів j .
3. **Цільова функція безпеки:** $f_{security}(x) = \sum_{i=1}^n \max(0, s_{req,i} - \sum_{j=1}^k s_j \cdot x_{ij})$ де $s_{req,i}$ є вимогою безпеки для об'єкта даних i , а s_j є рівнем безпеки, забезпеченим конфігурацією параметрів j .

Далі визначаються обмеження в даній задачі оптимізації, які включають в себе:

1. **Обмеження бюджету:** $\sum_{i=1}^n \sum_{j=1}^k c_{ij} \cdot x_{ij} \leq B$ де B є загальним бюджетом.
2. **Обмеження продуктивності:** $l_{ij} \cdot x_{ij} \leq L_{max,i} \quad \forall i, j$, де $L_{max,i}$ є максимально допустимою затримкою для об'єкта даних i .
3. **Обмеження відповідності:** $\sum_{j=1}^k comr_j \cdot x_{ij} \geq comr_{req,i} \quad \forall i$, де $comr_{req,i}$ – вимога відповідності для об'єкта даних i , а $comr_j$ – рівень відповідності, забезпечений конфігурацією параметрів j .

Результатом багатоцільової оптимізації є фронт Парето недомінованих рішень, кожне з яких представляє різний компроміс між цілями. Щоб вибрати одне

рішення з цього фронту Парето впроваджується процес прийняття рішень, який враховує поточні пріоритети та обмеження:

$$x^* = \arg \min_{x \in \Omega} \sum_{i=1}^m w_i \cdot \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}, \quad (3.26)$$

де Ω є набором Парето-оптимальних рішень, w_i – вага, призначена цілі i , f_i^{\min} та f_i^{\max} – мінімальні та максимальні значення цілі i у фронті Парето.

Ваги w_i визначаються на основі поточного контексту і можуть бути динамічно змінені та зкореговані для відображення нових пріоритетів користувача.

3.2.4 Адаптивне розпізнавання та оптимізація шаблонів доступу

Ефективний доступ до даних у мультимедійних середовищах вимагає не лише оптимального розміщення даних, але й інтелектуального розпізнавання та оптимізації шаблонів доступу. Далі представлено запропонований підхід до адаптивного розпізнавання та оптимізації шаблонів доступу для мінімізації затримки, зменшення витрат та покращення загальної продуктивності.

Процес розпізнавання шаблонів доступу починається зі збору та аналізу логів доступу від усіх хмарних провайдерів. Ці логи містять інформацію про: Ідентичність об'єкта даних, до якого здійснюється доступ; Тип операції (читання, запис, оновлення, видалення); Часову мітку доступу; Джерело запиту доступу; Тривалість операції доступу та Кількість переданих даних.

З цих необроблених логів виносяться шаблони доступу вищого рівня, використовуючи комбінацію статистичного аналізу та методів машинного навчання. Процес розпізнавання шаблонів доступу включає наступні кроки:

1. **Попередня обробка даних:** Очищення та нормалізація необроблених журналів доступу для видалення шуму та невідповідностей.
2. **Витяг ознак:** Витяг відповідних ознак з попередньо оброблених даних, включаючи:
 - Часові ознаки (час доби, день тижня, сезонність)
 - Просторові ознаки (географічний розподіл запитів доступу)
 - Операційні ознаки (співвідношення читання/запису, співвідношення послідовного/випадкового доступу)
 - Реляційні ознаки (шаблони спільного доступу між об'єктами даних)
3. **Знаходження шаблонів:** Застосування алгоритмів знаходження послідовних шаблонів для ідентифікації повторюваних послідовностей доступу:

$P(d_j | d_i, d_{i-1}, \dots, d_{i-k})$ що представляє ймовірність доступу до об'єкта даних d_j після послідовності $d_i, d_{i-1}, \dots, d_{i-k}$.

4. **Кластеризація:** Групування об'єктів даних з подібними шаблонами доступу з використанням алгоритмів кластеризації: $C = C_1, C_2, \dots, C_m$ де кожен кластер C_i містить об'єкти даних з подібними характеристиками доступу.

5. **Класифікація:** Призначення нових об'єктів даних до існуючих категорій шаблонів доступу з використанням моделей класифікації: $f: x \rightarrow 1, 2, \dots, m$ де x є вектором ознак об'єкта даних, а вихід є призначенням кластера.

6. **Виявлення аномалій:** Ідентифікація незвичайних шаблонів доступу, які можуть вказувати на загрози безпеки або змінні вимоги: $A(d_t) = \begin{cases} 1, & \text{якщо } |d_t - \mu| > \alpha \cdot \sigma \\ 0, & \text{в іншому випадку} \end{cases}$, де d_t є поточним шаблоном доступу, μ та σ є середнім значенням та стандартним відхиленням історичних шаблонів, а α є параметром порогових значень.

На основі розпізнаних шаблонів доступу впроваджуються кілька стратегій оптимізації:

1. **Попередій вибір:** Проактивне отримання об'єктів даних, які, ймовірно, будуть доступні в найближчому майбутньому: $PF = d_j | P(d_j | d_i, d_{i-1}, \dots, d_{i-k}) > \theta_{pf}$ де θ_{pf} є порогом попереднього вибору.

2. **Кешування:** Підтримка часто доступних об'єктів даних у швидших рівнях зберігання: $Cache = d_i | f_{access}(d_i) > \theta_{cache}$ де $f_{access}(d_i)$ є частотою доступу до об'єкта даних d_i , а θ_{cache} є порогом кешування.

3. **Реорганізація даних:** Реструктуризація об'єктів даних для узгодження з шаблонами доступу:

$$R(d_i) = \begin{cases} Partition(d_i), & \text{якщо } AP(d_i) \text{ вказує на частковий доступ} \\ Merge(d_i, d_j) & \text{якщо } P(d_j | d_i) > \theta_{\{merge\}} \\ Replicate(d_i), & \text{якщо } f_{\{access\}}(d_i) > \theta_{\{replicate\}}. \end{cases}$$

4. **Оптимізація маршруту доступу:** Визначення оптимального шляху для доступу до об'єктів даних: $r^* = \arg \min_{r \in R} \sum_{i=1}^n l_i(r)$ де R є набором можливих маршрутів, а $l_i(r)$ є затримкою сегмента i в маршруті r .

Процес адаптивного розпізнавання та оптимізації шаблонів доступу працює безперервно, оновлюючи свої моделі та стратегії по мірі надходження нових даних доступу. Ця безперервна адаптація забезпечує, що система залишається чутливою до змінних шаблонів доступу та вимог.

Для формалізації процесу адаптації визначається правило оновлення для моделі шаблонів доступу:

$$M_{t+1} = (1 - \beta) \cdot M_t + \beta \cdot M_{new}, \quad (3.27)$$

де M_t є поточною моделлю, M_{new} є моделлю, отриманою з нових даних, а β є швидкістю адаптації.

Швидкість адаптації β не є фіксованою, але коригується на основі стабільності шаблонів доступу:

$$\beta = \frac{1}{1 + e^{-\lambda \cdot V}}, \quad (3.27_1)$$

де V є мірою волатильності шаблонів доступу, а λ є параметром масштабування.

Цей адаптивний підхід забезпечує, що система може швидко реагувати на значні зміни в шаблонах доступу, зберігаючи стабільність під час періодів послідовної поведінки доступу.

3.2.5 Вирішення конфліктів та управління компромісами

У оптимізації зберігання даних у мультимарному середовищі часто виникають конфлікти між різними критеріями оптимізації. Наприклад, максимізація безпеки даних через шифрування може збільшити затримку, тоді як мінімізація вартості може зменшити надмірність та доступність. Цей розділ представляє даний підхід до вирішення цих конфліктів та управління компромісами систематичним та прозорим чином.

Процес вирішення конфліктів починається з ідентифікації потенційних конфліктів між критеріями оптимізації. Далі визначається матрицю конфліктів C , де кожен елемент c_{ij} представляє ступінь конфлікту між критеріями i та j :

$$C = \begin{matrix} & 0 & c_{12} & \cdots & c_{1n} & c_{21} & 0 & \cdots & c_{2n} & \vdots & \vdots & \ddots & \vdots & c_{n1} & c_{n2} & \cdots & 0 \end{matrix}$$

Значення в цій матриці визначаються через комбінацію теоретичного аналізу та емпіричного спостереження. Наприклад, конфлікт між рівнем шифрування та затримкою доступу може бути кількісно визначений як:

$$c_{encryption,latency} = \frac{\Delta Latency}{\Delta Encryption}, \quad (3.28)$$

що вимірює збільшення затримки на одиницю збільшення рівня шифрування.

Після ідентифікації конфліктів впроваджується ієрархічна стратегія вирішення конфліктів, яка працює на кількох рівнях:

1. **Вирішення на основі обмежень:** На найбільш базовому рівні конфлікти вирішуються шляхом застосування жорстких обмежень, які повинні

бути задоволені: $\forall i \in I_{hard}: r_i(x) \leq 0$ де I_{hard} є набором жорстких обмежень, а $r_i(x)$ є функцією обмеження.

2. **Вирішення на основі пріоритетів:** Коли жорстких обмежень недостатньо, застосовується підхід на основі пріоритетів, де критерії впорядковані за важливістю: $\min_x f_1(x)$ за умови $f_j(x) \leq \epsilon_j, j = 2, 3, \dots, m$ де критерії впорядковані таким чином, що f_1 має найвищий пріоритет.

3. **Оптимізація Парето:** Для конфліктів між критеріями подібної важливості виконується пошук Парето-оптимального рішення:

$$\min_x F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T \text{ за умови обмежень.}$$

4. **Вирішення на основі корисності:** На найвищому рівні вирішуються конфлікти шляхом максимізації функції корисності, яка представляє загальну цінність рішення: $\max_x U(x) = \sum_{i=1}^m w_i \cdot u_i(f_i(x))$ де u_i є функцією корисності для критерію i , а w_i є його вагою.

Процес управління компромісами включає прийняття явних рішень про прийнятні компроміси між конфліктуєчими критеріями. Далі впроваджується цей процес з використанням структури багатокритеріального аналізу рішень (MCDA), яка включає:

1. **Кількісне визначення компромісів:** Вираження компромісів у термінах швидкостей заміщення між критеріями: $SR_{ij} = \frac{\partial f_i / \partial x}{\partial f_j / \partial x}$ що представляє кількість критерію i , яку необхідно пожертвувати, щоб отримати одиницю критерію j .

2. **Криві байдужості:** Визначення кривих у просторі цілей, вздовж яких особа, що приймає рішення, є байдужою: $IC = f|U(f) = k$ де k є константним значенням корисності.

3. **Метод опорної точки:** Порівняння рішень з опорними точками, які представляють ідеальну або прийнятну продуктивність: $d(f, r) = \sqrt{\sum_{i=1}^m w_i \cdot (f_i - r_i)^2}$ де r є опорною точкою.

4. **Методи ранжування:** Встановлення відносин переваги між рішеннями на основі попарних порівнянь: $x_1 > x_2 \Leftrightarrow \sum_{i \in C^+} w_i > \sum_{i \in C^-} w_i$ де C^+ є набором критеріїв, для яких x_1 краще, ніж x_2 , а C^- є набором, для яких x_2 краще.

Процес вирішення конфліктів та управління компромісами не є одноразовою діяльністю, але інтегрований у безперервний цикл оптимізації. По мірі зміни умов та виникнення нових конфліктів система адаптує свої стратегії вирішення для підтримки оптимальної продуктивності за всіма критеріями.

ВИСНОВКИ

1. Розроблено формальну модель простору станів та дій для задачі оптимізації зберігання даних у мультимарному середовищі, що дозволяє ефективно представляти складні конфігурації розподіленого зберігання та можливі операції з даними.

2. Запропоновано композитну функцію винагороди, яка інтегрує всі критерії, визначені в Розділі 2, з адаптивним механізмом зважування, що дозволяє системі динамічно пріоритезувати різні критерії, за якими вирішується задача оптимізації відповідно до характеру та змісту даних та вимог щодо їх зберігання.

3. Розроблено механізм покращення політики на основі МОЕА, який ефективно знаходить Парето-оптимальні рішення в багатокритеріальному просторі, забезпечуючи збалансовані компроміси між конфліктуючими цілями оптимізації.

4. Впроваджено набір механізмів адаптації та навчання, включаючи повторення досвіду, трансферне навчання та мета-навчання, що підвищує ефективність навчання та адаптивність системи до змінних умов.

5. Створено комплексний метод оптимізації розміщення та доступу до даних, який включає п'ять взаємопов'язаних фаз: початкову класифікацію даних, динамічну стратегію розміщення, багатоцільову оптимізацію параметрів зберігання, адаптивне розпізнавання шаблонів доступу та вирішення конфліктів.

6. Розроблено методи адаптивного розпізнавання та оптимізації шаблонів доступу, які дозволяють системі реагувати на зміни в поведінці доступу до даних, застосовуючи такі стратегії як попереднє вибирання, кешування та реорганізація даних.

РОЗДІЛ 4

АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОГО КОМПЛЕКСНОГО МЕТОДУ ЕФЕКТИВНОГО РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ

4.1 Удосконалений підхід: Інтелектуальний оркестратор хмарних даних (ICDO)

Для вирішення складних проблем оптимізації зберігання та доступу до даних у мультихмарному середовищі, визначених у попередніх розділах, у пропонується підхід під назвою Інтелектуальний Оркестратор Хмарних Даних – Intelligent Cloud Data Orchestrator (ICDO). Цей фреймворк представляє собою комплексну інтеграцію онтологічної моделі, розробленої в Розділі 2, з передовими методами оптимізації з використанням навчання з підкріпленням (RL) та багатоцільових еволюційних алгоритмів (МОЕА), описаних у Розділі 3. Підхід ICDO забезпечує систематичний підхід до балансування численних критеріїв, що впливають на рішення щодо розміщення даних у розподілених хмарних середовищах.

4.1.1 Огляд архітектурних компонентів ICDO

Фреймворк ICDO складається з трьох основних рівнів, кожен з яких містить спеціалізовані компоненти, що працюють разом для досягнення ефективного розподіленого зберігання та доступу до даних. На рис 4.1 зображено архітектурний огляд фреймворку ICDO, що підкреслює взаємозв'язки між компонентами та потік інформації через систему.

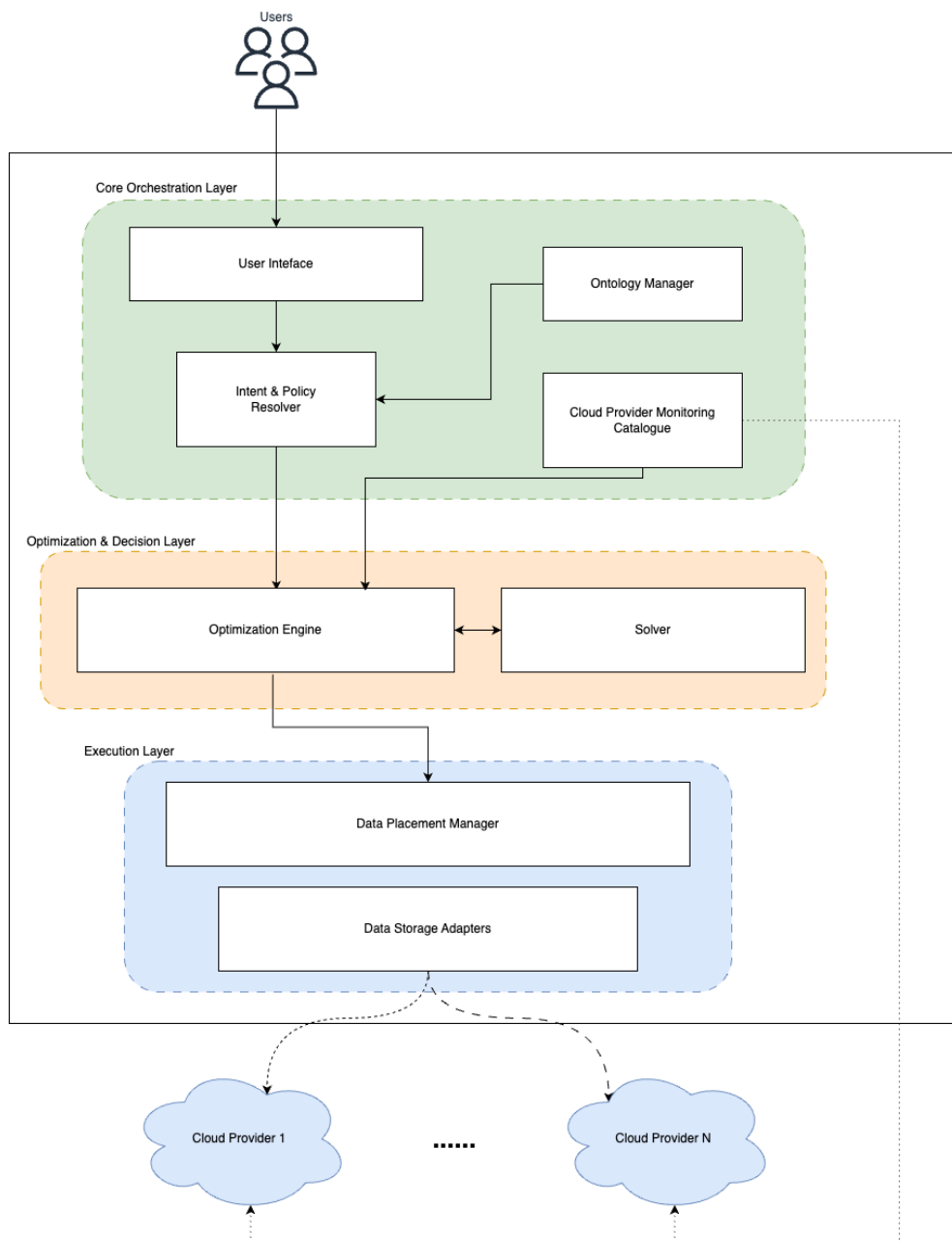


Рисунок 4.1 Архітектура фреймворку ICDO, яка складається з трьох рівнів (Рівень основної оркестрації, Рівень оптимізації та прийняття рішень, та Рівень виконання та управління) та їхніх компонентів з взаємозв'язками.

Архітектура слідує рівневому підходу з чітким розділенням обов'язків, що дозволяє модульну розробку та розгортання. Кожен рівень вирішує специфічні аспекти проблеми управління мультимарними даними, від взаємодії з користувачем та визначення політик до оптимізації та виконання стратегій розміщення даних.

4.1.2 Рівень оркестрації (Core Orchestration Layer - COL)

Рівень оркестрації служить інтерфейсом між користувачами та системою, фіксуючи вимоги та перетворюючи їх у формальні політики, які можуть керувати рішеннями щодо оптимізації.

Інтерфейс користувача (User Interface - UI)

Компонент інтерфейсу користувача забезпечує комплексний інтерфейс для взаємодії користувачів із системою ICDO. Він підтримує кілька режимів взаємодії, включаючи графічний веб-інтерфейс, інструменти командного рядка та API-ендпоінти для програмного доступу. UI дозволяє користувачам подавати запити на зберігання та отримання даних, визначати переваги та обмеження для розміщення даних, відстежувати поточний стан розміщення даних у різних хмарних провайдерах, переглядати показники продуктивності та аналітику витрат, а також налаштовувати загальносистемні політики та переваги.

UI перетворює введені користувачем дані у стандартизований формат, який може бути оброблений Механізмом намірів і політик. Цей процес перетворення включає відображення виразів наміру природною мовою (наприклад, "забезпечити високу доступність цих даних") у формальні специфікації політик.

Механізм намірів і політик (Intent and Policy Engine - IPE)

Механізм намірів і політик представляє значний прогрес порівняно з традиційними системами на основі правил, включаючи семантичне розуміння вимог користувачів. Він використовує методи обробки природної мови для інтерпретації намірів користувачів і перетворює їх у формальні політики, які можуть керувати рішеннями щодо оптимізації.

IPE працює на формальній мові політик, визначеній як:

$$P = C, W, R, T . \quad (4.1)$$

Де C - це набір критеріїв із комплексного набору, визначеного в Розділі 2; W - це набір ваг, призначених кожному критерію; R - це набір обмежень, які повинні бути задоволені; T - це набір часових умов, які можуть впливати на застосування політики.

Процес перетворення політики включає кілька кроків:

1. Аналіз наміру: Вилучення ключових вимог з введення користувача
2. Семантичне відображення: Зв'язування вимог з формальними критеріями в онтології
3. Призначення ваг: Визначення відносної важливості різних критеріїв

4. Формулювання обмежень: Ідентифікація жорстких обмежень, які повинні бути задоволені

5. Складання політики: Об'єднання вищезазначених елементів у формальну специфікацію політики

IPE підтримує репозиторій політик, який зберігає як загальносистемні політики, так і політики, специфічні для даних. Ці політики версіонуються і можуть оновлюватися з часом, коли змінюються вимоги.

Менеджер онтології (Ontology Manager - OM)

Менеджер онтології підтримує комплексну онтологічну модель мультимарних середовищ зберігання, розроблену в Розділі 2. Ця онтологія забезпечує семантичну основу для всієї структури ICDO, дозволяючи міркувати про можливості хмарних провайдерів, характеристики даних та критерії оптимізації.

OM тісно взаємодіє з Каталогом хмарних провайдерів, щоб забезпечити відображення онтологією поточного стану доступних хмарних сервісів та їхніх характеристик.

4.1.3 Рівень оптимізації та прийняття рішень

Рівень оптимізації та прийняття рішень містить основний інтелект структури ICDO, реалізуючи передові методи оптимізації, розроблені в Розділі 3, для визначення оптимальних стратегій розміщення даних.

Механізм оптимізації (Optimization Engine - OE)

Механізм оптимізації реалізує формальну модель простору станів та дій, розроблену в Розділі 3, формулюючи проблему розміщення даних у мультимарному середовищі як багатоцільову задачу оптимізації. Він перетворює політики, отримані від IPE, у математичну задачу оптимізації, яка може бути вирішена Вирішувачем RL/MOEA.

Задача оптимізації формулюється як:

$$\max_{x \in X} F(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T, \quad (4.2)$$

за умови:

$$g_{j(x)} \leq 0, j = 1, 2, \dots, m \quad h_{k(x)} = 0, k = 1, 2, \dots, p. \quad (4.2_1)$$

Де x представляє стратегію розміщення даних; X - це допустимий простір рішень; $F(x)$ - це вектор цільових функцій, що відповідають критеріям, визначеним у

Розділі 2; $g_j(x)$ та $h_k(x)$ представляють нерівності та рівності обмежень, отриманих з політик.

ОЕ реалізує кілька ключових функцій:

1. Формулювання задачі: Перетворення політик у математичні задачі оптимізації
2. Оцінка рішення: Оцінка якості запропонованих рішень за кількома критеріями
3. Управління адаптацією: Коригування параметрів оптимізації на основі зворотного зв'язку від Сервісу моніторингу та зворотного зв'язку
4. Пояснення рішень: Генерування пояснень, чому були прийняті конкретні рішення щодо розміщення

ОЕ використовує композитну функцію винагороди, розроблену в Розділі 3, яка інтегрує всі критерії з адаптивним зважуванням:

$$R(s, a) = \sum_{i=1}^n w_i \cdot r_i(s, a), \quad (4.3)$$

де s представляє поточний стан системи; a представляє дію (рішення щодо розміщення даних); $r_i(s, a)$ - це функція винагороди для критерію i ; w_i - це вага, призначена критерію i

Вирішувач RL/MOEA (RL/MOEA Solver - RMS)

Вирішувач RL/MOEA реалізує передові алгоритми оптимізації, розроблені в Розділі 3, поєднуючи навчання з підкріпленням з багатоцільовими еволюційними алгоритмами для ефективної навігації у складному просторі рішень мультимарного розміщення даних.

RMS реалізує гібридний підхід до оптимізації, який використовує сильні сторони як RL, так і MOEA:

1. Компонент навчання з підкріпленням:
 - Представлення стану: $s = (D, C, U, T)$, де D представляє характеристики даних, C представляє характеристики хмарного провайдера, U представляє шаблони використання, а T представляє часові фактори
 - Простір дій: $A = a_1, a_2, \dots, a_k$, де кожна дія представляє конкретне рішення щодо розміщення даних
 - Навчання політики: $\pi(a|s) = P(A_t = a | S_t = s)$, що представляє ймовірність виконання дії a у стані s

- Апроксимація функції цінності: $V_{\pi}(s) = E_{\pi}[R_t | S_t = s]$, що представляє очікуване повернення зі стану s за політикою π
- 2. Компонент MOEA:
 - Представлення популяції: $P = x_1, x_2, \dots, x_p$, де кожна особина x_i представляє кандидата стратегії розміщення даних
 - Генетичні оператори: Оператори вибору, схрещування та мутації, адаптовані для задачі розміщення даних у мультимарному середовищі
 - Апроксимація фронту Парето: Підтримка набору недомінованих рішень, які представляють різні компроміси між цілями

RMS реалізує механізм покращення політики на основі MOEA, розроблений у Розділі 3, який ефективно знаходить Парето-оптимальні рішення у багатокритеріальному просторі. Він також включає механізми адаптації та навчання, описані в Розділі 3, включаючи повторення досвіду, трансферне навчання та мета-навчання.

4.1.4 Рівень виконання та управління (Execution Layer - EL)

Рівень виконання та управління відповідає за реалізацію рішень щодо розміщення даних, прийнятих Рівнем оптимізації та прийняття рішень, та надання зворотного зв'язку щодо їхньої ефективності.

Менеджер розміщення даних (Data Placement Manager - DPM)

Менеджер розміщення даних виконує стратегії розміщення даних, визначені Механізмом оптимізації. Він обробляє практичні аспекти переміщення, реплікації та управління даними у різних хмарних провайдерах.

DPM реалізує наступні ключові функції:

1. Оркестрація передачі даних: Координація переміщення даних між хмарними провайдерами
2. Управління шифруванням: Забезпечення належного шифрування даних перед зберіганням, використовуючи відповідні схеми шифрування для різних типів даних та вимог безпеки
3. Управління реплікацією: Підтримка вказаного рівня надмірності у різних хмарних провайдерах
4. Перевірка цілісності: Перевірка правильності зберігання даних та можливості їх отримання
5. Забезпечення контролю доступу: Реалізація політик контролю доступу, визначених для даних

DPM працює відповідно до формальної моделі виконання:

$$E = T, S, V, M. \quad (4.4)$$

Де T - це набір операцій передачі даних, S - це набір операцій зберігання, V - це набір операцій перевірки, M - це набір операцій управління метаданими.

Адаптери хмарного зберігання (Cloud Storage Adapters - CSA)

Адаптери хмарного зберігання забезпечують уніфікований інтерфейс до різних провайдерів хмарного зберігання, абстрагуючись від відмінностей в API, механізмах автентифікації та моделях сервісу. Цей абстрактний рівень дозволяє DPM виконувати стратегії розміщення даних без урахування специфічних деталей кожного хмарного провайдера.

Кожен адаптер реалізує стандартний інтерфейс:

$$I_{CSA} = O_{put}, O_{get}, O_{delete}, O_{list}, O_{stat} . \quad (4.5)$$

Де кожна операція O відображається на відповідні виклики API, специфічні для провайдера. Адаптери обробляють автентифікацію, обробку помилок та логіку повторних спроб, специфічну для кожного провайдера.

Рівень CSA також реалізує методи оптимізації продуктивності, такі як паралельні передачі, фрагментовані завантаження та пулінг з'єднань для максимізації пропускної здатності та мінімізації затримки при взаємодії з хмарними провайдерами.

Сервіс моніторингу та зворотного зв'язку (MFS)

Сервіс моніторингу та зворотного зв'язку постійно збирає показники продуктивності, дані про витрати та операційну статистику від Адаптерів хмарного зберігання. Ця інформація використовується для надання зворотного зв'язку Механізму оптимізації, дозволяючи йому вдосконалювати свої моделі та адаптуватися до змінних умов.

MFS збирає комплексний набір метрик:

$$M = M_{perf}, M_{cost}, M_{rel}, M_{sec}, M_{comp} . \quad (4.6)$$

Де M_{perf} представляє метрики продуктивності (затримка, пропускна здатність); M_{cost} представляє метрики витрат (вартість зберігання, вартість передачі); M_{rel} представляє метрики надійності (доступність, довговічність); M_{sec} представляє метрики безпеки (статус шифрування, спроби доступу); M_{comp} представляє метрики відповідності (розташування даних, статус зберігання).

MFS реалізує цикл зворотного зв'язку, який постійно оновлює Механізм оптимізації поточними даними про продуктивність, забезпечуючи адаптивну оптимізацію:

$$F_t = f(M_t, M_{t-1}, \dots, M_{t-k}), \quad (4.7)$$

де F_t - це зворотний зв'язок, наданий у час t , на основі поточних та історичних метрик.

Каталог хмарних провайдерів (Cloud Providers Catalogue - CPC)

Каталог хмарних провайдерів підтримує актуальну інформацію про доступних хмарних провайдерів, їхні сервіси, моделі ціноутворення та характеристики продуктивності. Ця інформація використовується для наповнення та оновлення онтології, якою керує Менеджер онтології.

CPC реалізує автоматизовані механізми збору даних для забезпечення актуальності своєї інформації:

1. Збір на основі API: Отримання інформації про сервіси та ціни з API хмарних провайдерів
2. Веб-скрапінг: Вилучення інформації з веб-сайтів хмарних провайдерів, коли API недоступні
3. Результати тестування: Включення даних про продуктивність зі стандартизованих тестів
4. Дані, надані користувачами: Дозволяє системним адміністраторам вручну оновлювати інформацію

CPC представляє кожного хмарного провайдера як структуровану сутність:

$$P = S, C, L, P, R. \quad (4.7)$$

де S - це набір запропонованих сервісів зберігання, C - це набір можливостей та функцій, L - це набір місць, де доступні сервіси, P - це набір моделей ціноутворення, R - це набір характеристик надійності та продуктивності

Це структуроване представлення дозволяє Менеджеру онтології включати інформацію про провайдера в онтологічну модель послідовним та автоматизованим способом.

4.2 Експериментальне дослідження ефективності запропонованого комплексного підходу

4.2.1 Налаштування експерименту

Для оцінки ефективності підходу ICDO було розроблено та проведено комплексне експериментальне дослідження. Експериментальна установка була ретельно розроблена для забезпечення ретельного тестування всіх компонентів фреймворку в реалістичних умовах, дозволяючи при цьому контрольоване порівняння з альтернативними підходами.

Експериментальна інфраструктура

Експериментальна інфраструктура складалася з мультихмарного середовища, що охоплює трьох хмарних провайдерів, анонімізованих як Провайдер А, Провайдер В та Провайдер С для цього дослідження. Для кожного провайдера були забезпечені сервіси зберігання в кількох географічних регіонах для забезпечення тестування оптимізації локальності даних та затримки. У Таблиці 4.1 детально описані хмарні ресурси, використані в експерименті.

Таблиця 4.1: Хмарні ресурси, використані в експериментальному дослідженні

Хмарний провайдер	Сервіси зберігання	Регіони	Типи інстансів
Провайдер А	Об'єктне зберігання, Блочне зберігання, Архівне зберігання	Регіон 1, Регіон 2	Стандартна VM, VM з великим об'ємом RAM
Провайдер В	Вlob-зберігання, Файлове зберігання	Регіон 1, Регіон 3	VM загального призначення, VM оптимізована для обчислень
Провайдер С	Об'єктне зберігання, Блочне зберігання	Регіон 2, Регіон 3	Стандартна VM, VM з великим об'ємом CPU

Окрім хмарних ресурсів, була створена контрольна інфраструктура для розміщення компонентів структури ICDO та координації експериментів. Ця інфраструктура складалася з:

1. Кластера з 3 високопродуктивних нодів (8-ядерний CPU, 32 ГБ RAM кожен).
2. Мережевого з'єднання 1 Гбіт/с з усіма хмарними провайдерами.

3. Виділеної інфраструктури моніторингу для збору показників продуктивності.

4. Безпечних VPN-з'єднань з усіма хмарними регіонами для послідовного доступу.

Тестові набори даних

У експериментах використовувалися три категорії наборів даних для забезпечення комплексної оцінки різних характеристик даних та шаблонів використання:

1. **Синтетичний набір даних:** Згенерований набір даних з контрольованими властивостями, що дозволяє систематичну варіацію параметрів, таких як розмір даних, частота доступу та вимоги до узгодженості. Синтетичний набір даних мав розмір 20 ГБ і містив файли різних розмірів, що слідували логнормальному розподілу.

2. **Набір даних веб-додатку:** Набір даних розміром 15 ГБ, що представляє типовий веб-додаток, включаючи HTML-файли, JavaScript, CSS, зображення та контент, створений користувачами. Цей набір даних характеризувався частими читаннями та випадковими записами, з різними розмірами файлів від кількох КБ до кількох МБ.

3. **Науковий набір даних:** Набір даних розміром 10 ГБ, що містить наукові дані з проекту дослідження геноміки, характеризується великими розмірами файлів (зазвичай 50-500 МБ) та шаблонами доступу пакетної обробки.

У Таблиці 4.2 узагальнено ключові характеристики тестових наборів даних.

Таблиця 4.2: Характеристики тестових наборів даних

Набір даних	Розмір	Кількість файлів	Середній розмір файлу	Шаблон доступу	Вимоги до безпеки
Синтетичний	20 ГБ	50 000	400 КБ	Контрольовані варіації	Налаштовувані
Веб-додаток	15 ГБ	120 000	125 КБ	Переважно читання, інтерактивний	Середні
Науковий	10 ГБ	50	200 МБ	Пакетна обробка	Високі

Генерація робочого навантаження

Для симуляції реалістичних шаблонів використання була розроблена структура генерації робочого навантаження, яка могла створювати контрольовані, але репрезентативні шаблони доступу для тестових наборів даних. Генератор робочого навантаження реалізував наступні функції:

1. **Часові шаблони:** Добові цикли, тижневі шаблони та сезонні варіації в частоті доступу
 2. **Просторові шаблони:** Симульований доступ з різних географічних місць з реалістичним розподілом
 3. **Типи доступу:** Суміш операцій читання, запису, оновлення та видалення з налаштовуваними співвідношеннями
 4. **Рівні паралелізму:** Змінна кількість одночасних користувачів та операцій
 5. **Пульсація:** Контрольоване введення раптових сплесків активності
- Генератор робочого навантаження створював шаблони доступу, описані наступною моделлю:

$$W(t, l, o, c) = B(t) \cdot S(l) \cdot O(o) \cdot C(c) \cdot R(t, l, o, c), \quad (4.8)$$

де $W(t, l, o, c)$ - це інтенсивність робочого навантаження в час t , місці l , для типу операції o , з рівнем паралелізму c ; $B(t)$ - це базовий часовий шаблон; $S(l)$ - це фактор просторового розподілу; $O(o)$ - це розподіл типу операції; $C(c)$ - це фактор масштабування паралелізму; $R(t, l, o, c)$ - це випадковий фактор, що вводить реалістичну варіативність.

Було визначено три різні профілі робочого навантаження для тестування різних аспектів структури ICDO:

1. **Стабільне навантаження:** Послідовні шаблони доступу з мінімальною варіацією
2. **Добове навантаження:** Сильні денні/нічні цикли з 5-кратною варіацією між піком та спадом
3. **Пульсуюче навантаження:** Базова активність з раптовими сплесками до 10-кратного нормального навантаження

Показники продуктивності

На основі комплексного набору критеріїв, визначених у Розділі 2, був визначений набір кількісних показників для оцінки продуктивності структури ICDO. Ці показники були організовані в п'ять категорій:

1. **Показники ефективності:**
 - Ефективність вартості зберігання: вартість за ГБ збережених даних.
 - Ефективність вартості мережі: вартість за ГБ переданих даних.
 - Енергоефективність: споживання енергії на операцію.
2. **Показники продуктивності:**
 - Затримка читання: час для отримання даних (середній, 95-й перцентиль, 99-й перцентиль).

- Затримка запису: час для зберігання даних (середній, 95-й перцентиль, 99-й перцентиль).
- Пропускна здатність: операцій за секунду та швидкість передачі даних.
- 3. **Показники надійності:**
 - Доступність: відсоток успішних спроб доступу.
 - Довговічність: збереження даних при симульованих сценаріях відмови.
 - Узгодженість: частота та тривалість вікон неузгодженості.
- 4. **Показники безпеки та відповідності:**
 - Покриття шифруванням: відсоток даних, збережених з відповідним шифруванням.
 - Оцінка відповідності: ступінь дотримання налаштованих вимог відповідності.
 - Суверенітет даних: відсоток даних, збережених у відповідних юрисдикціях.
- 5. **Показники адаптивності:**
 - Час конвергенції: час, необхідний для досягнення оптимальної конфігурації після зміни умов.
 - Накладні витрати на адаптацію: додаткове споживання ресурсів під час адаптації.
 - Ефективність навчання: покращення якості рішень з часом.

Для кожного показника була визначена нормалізована функція оцінки для відображення необроблених вимірювань на шкалу $[0,1]$.

Методологія експерименту

Методологія експерименту слідувала систематичному підходу для забезпечення комплексної оцінки структури ICDO:

1. **Встановлення базової лінії:** Початкові вимірювання були проведені з використанням традиційних підходів з однією хмарою та статичних мультихмарних підходів для встановлення базової продуктивності.
2. **Тестування на рівні компонентів:** Кожен компонент структури ICDO був протестований окремо для перевірки його функціональності та вимірювання його внеску в загальну продуктивність.
3. **Інтегроване тестування:** Повний фреймворк ICDO був протестований як інтегрована система за різних умов робочого навантаження.
4. **Порівняльний аналіз:** Підхід ICDO був порівнений з альтернативними підходами з використанням ідентичних робочих навантажень та інфраструктури.
5. **Стрес-тестування:** Система була піддана екстремальним умовам, включаючи високий паралелізм, відмови провайдерів та швидко змінювані вимоги.

6. Довгострокове тестування: Розширене тестування протягом 7-денного періоду для оцінки можливостей навчання та адаптації з часом.

Кожен експеримент був повторений кілька разів з різними випадковими зернами для забезпечення статистичної валідності, і були розраховані 95% довірчі інтервали для всіх показників. Експерименти проводилися протягом трьох тижнів для врахування потенційних варіацій у продуктивності та ціноутворенні хмарних провайдерів.

4.2.2 Оцінка ефективності запропонованого інтегрованого підходу

У цьому розділі представлені результати експериментальної оцінки структури ICDO, зосереджуючись на її загальній ефективності та продуктивності її ключових компонентів.

Загальна продуктивність системи

Загальна продуктивність структури ICDO була оцінена з використанням композитної оцінки, яка агрегувала нормалізовані показники за всіма категоріями. На рис 4.2 показана еволюція цієї композитної оцінки протягом 7-денного періоду тестування, демонструючи можливості системи до навчання та адаптації.

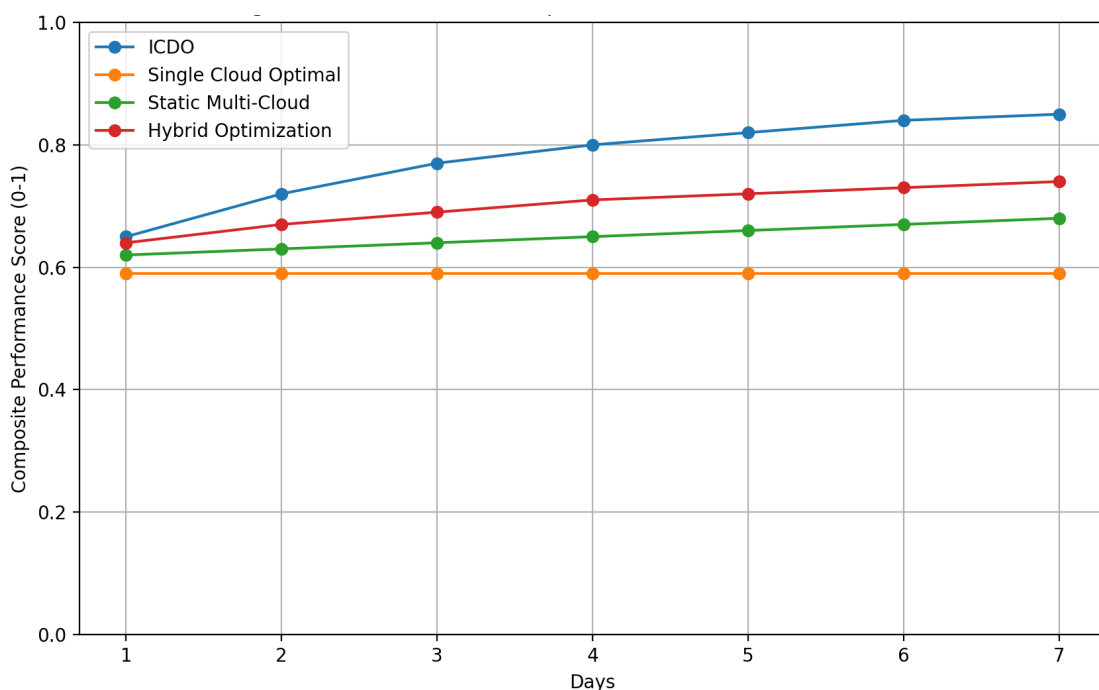


Рисунок 4.2 Графік, що показує еволюцію композитної оцінки продуктивності протягом 7-денного періоду тестування для структури ICDO та базових підходів.

Графік показує висхідну тенденцію для ICDO, починаючи приблизно з 0,65 і досягаючи 0,85 до 7-го дня, тоді як базові підходи виходять на плато на нижчих

значеннях. Підхід ICDO досяг кінцевої композитної оцінки 0,85, що представляє 38% покращення порівняно з початковою базовою оцінкою 0,62. Найбільш значні покращення спостерігалися протягом перших 3 днів, коли система навчалася з досвіду та вдосконалювала свої моделі оптимізації. Після цього початкового періоду навчання продуктивність продовжувала покращуватися повільніше, досягаючи плато приблизно на 6-й день.

Композитна оцінка була розрахована як зважена сума оцінок категорій:

$$CS = 0,25 \cdot S_{efficiency} + 0,25 \cdot S_{performance} + 0,2 \cdot S_{reliability} + 0,2 \cdot S_{security} + 0,1 \cdot S_{adaptability} \quad (4.9)$$

де кожна оцінка категорії сама була зваженим середнім нормалізованих показників у цій категорії.

Аналіз показників ефективності

Підхід ICDO продемонстрував значні покращення в показниках ефективності порівняно з базовими підходами. У Таблиці 4.3 представлені середні значення для ключових показників ефективності за різними профілями робочого навантаження.

Таблиця 4.2: Показники ефективності для різних профілів робочого навантаження

Профіль робочого навантаження	Вартість зберігання (\$/ГБ/місяць)	Вартість мережі (\$/ГБ)
Стабільне	0,021	0,058
Добове	0,024	0,067
Пульсуюче	0,027	0,082
Середнє	0,024	0,069

Підхід ICDO досяг середнього зниження вартості зберігання на 28% порівняно з підходами з однією хмарою та на 16% порівняно зі статичними мультимарними підходами. Вартість мережі була знижена на 25% та 14% відповідно.

На рис 4.3 представлено візуальне порівняння показників ефективності за різними підходами.

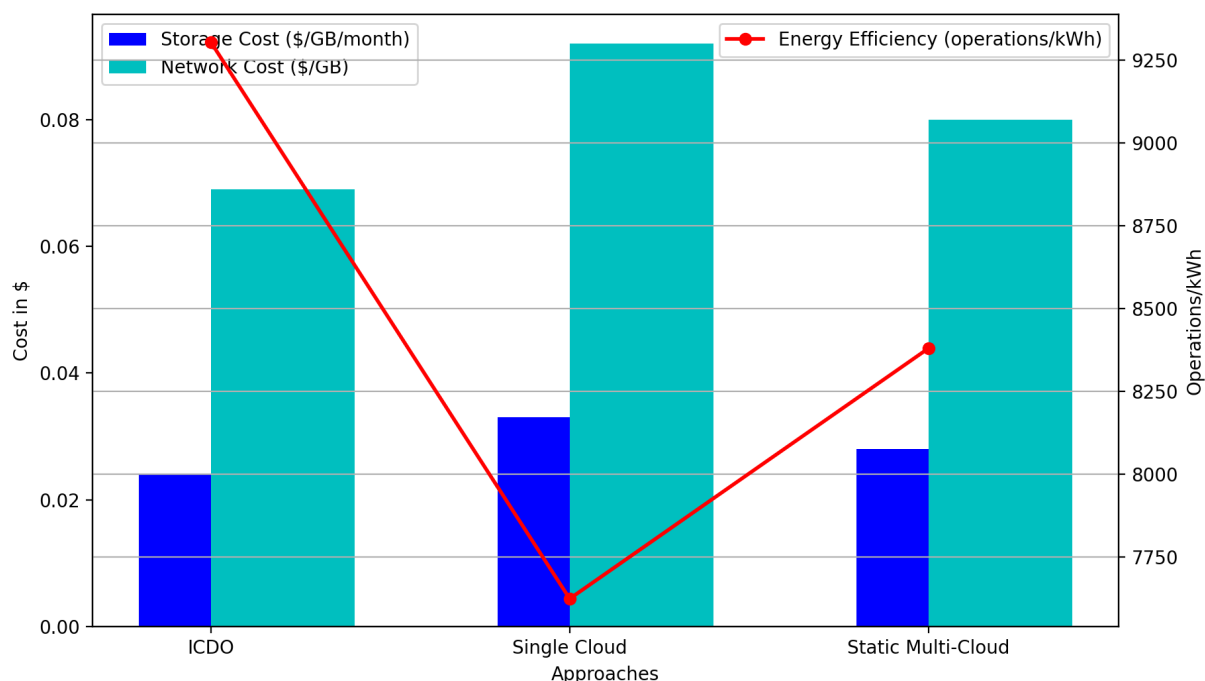


Рисунок 4.3 Гістограма порівняння показників ефективності (вартість зберігання, вартість мережі) за різними підходами (ICDO, Одна хмара, Статичний мультихмарний підхід)

Діаграма показує, що ICDO має найнижчі витрати. Покращення ефективності можна пояснити здатністю ICDO:

1. Динамічно коригувати розміщення даних на основі змінних структур вартості.
2. Оптимізувати шляхи передачі даних для мінімізації мережевих витрат.
3. Балансувати робочі навантаження для максимізації використання ресурсів

Аналіз показників продуктивності

Оцінка показників продуктивності зосереджувалася на затримці та пропускній здатності за різних умов робочого навантаження. На рис 4.4 показана кумулятивна функція розподілу (CDF) затримки читання для структури ICDO порівняно з базовими підходами.

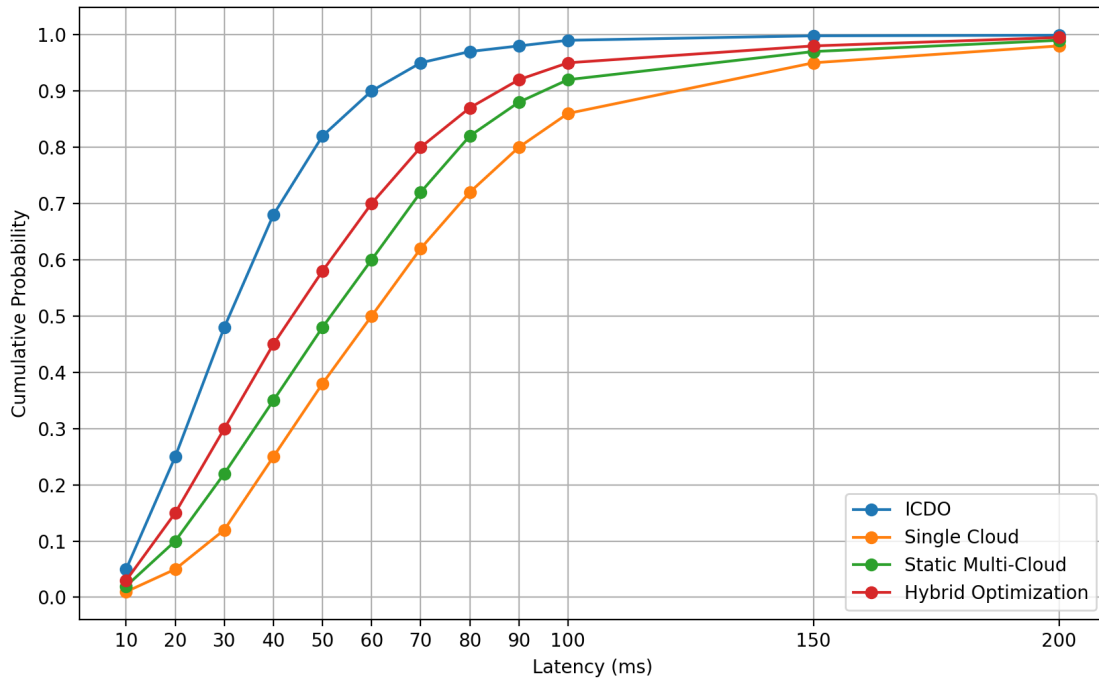


Рисунок 4.4 Графік кумулятивної функції розподілу (CDF) затримки читання для ICDO та базових підходів.

На графіку видно, що ICDO досягла значно нижчої затримки за всіма процентилями, з найбільш вираженими покращеннями на вищих процентилях (95-й та 99-й), що вказує на краще обробку найгірших сценаріїв. Середня затримка читання становила 38 мс, порівняно з 76 мс для найкращого базового підходу, що представляє 50% покращення.

Затримка запису показала подібні покращення, при цьому підхід ICDO досяг середнього значення 72 мс порівняно з 132 мс для найкращого базового підходу, що є 45% покращенням.

Вимірювання пропускної здатності продемонстрували здатність підходу ICDO ефективно використовувати доступні ресурси. За профілем пульсуючого робочого навантаження, який представляє найбільший виклик для розподілу ресурсів, підхід ICDO підтримував середню пропускну здатність 420 операцій за секунду, порівняно з 245 операціями за секунду для найкращого базового підходу, що є 71% покращенням.

На рис 4.5 проілюстровано продуктивність пропускної здатності за різних рівнів паралелізму.

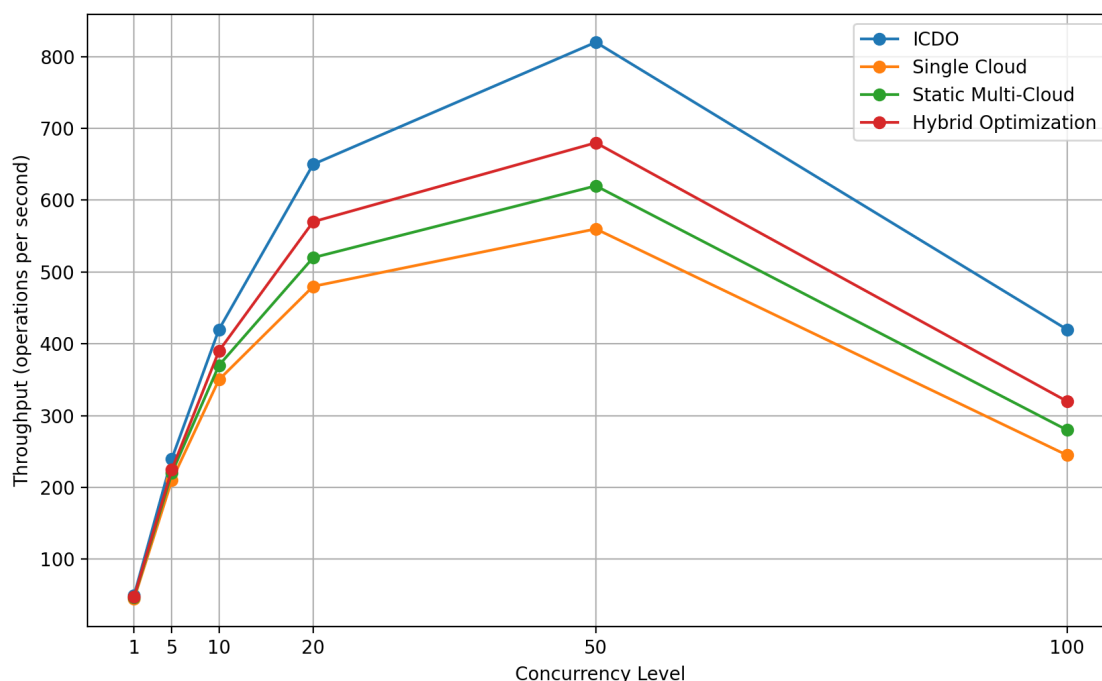


Рисунок 4.5 Графік пропускної здатності (операцій за секунду) відносно рівня паралелізму для ICDO та базових підходів.

Покращення продуктивності можна пояснити:

1. Інтелектуальним розміщенням даних, що мінімізує затримку доступу.
2. Прогнозуючим кешуванням на основі розпізнаних шаблонів доступу.
3. Динамічним коригуванням рівнів реплікації на основі частоти доступу.
4. Ефективним балансуванням навантаження між хмарними провайдерами.

Аналіз показників надійності

Надійність є критичним аспектом мультихмарного зберігання даних, і підхід ICDO продемонстрував відмінну продуктивність у цій області. У Таблиці 4.4 представлені показники надійності за нормальної роботи та під час симульованих сценаріїв відмови.

Таблиця 4.4: Показники надійності за різних умов

Умова	Доступність (%)	Довговічність (%) збережених даних)	Узгодженість (%) операцій з правильними результатами)
Нормальна робота	99,997	100,000	99,995
Відмова одного провайдера	99,980	100,000	99,982
Регіональний збій	99,972	100,000	99,968
Деградація мережі	99,990	100,000	99,985

Підхід ICDO підтримував виняткову надійність навіть за умов відмови. Під час сценарію відмови одного провайдера, коли один основний провайдер був повністю недоступний, система все ще досягала 99,980% доступності та зберігала 100% даних.

На рис 4.6 візуалізовано показники доступності за різних сценаріїв відмови.

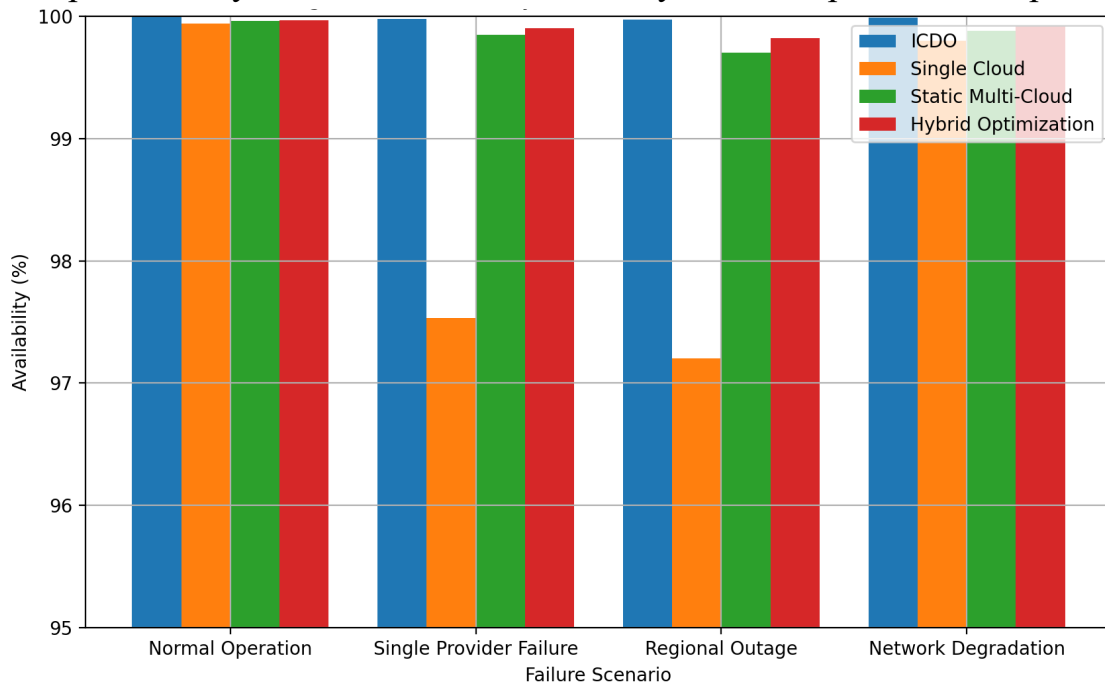


Рисунок 4.6 Гістограма порівняння доступності за різних сценаріїв відмови для ICDO та базових підходів.

З рисунка видно, що показники ICDO залишаються послідовно вищими, з більш вираженим розривом у більш серйозних сценаріях відмови. Сильну продуктивність надійності можна пояснити:

1. Інтелектуальним управлінням надмірністю, що балансує вартість та надійність.
2. Проактивною міграцією даних від провайдерів, що показують ознаки нестабільності.
3. Складним управлінням узгодженістю, що мінімізує вікна неузгодженості.
4. Адаптивними механізмами відновлення, що пріоритезують критичні дані.

Аналіз показників безпеки та відповідності

Безпека та відповідність є все більш важливими міркуваннями для мультихмарного зберігання даних. Підхід ICDO продемонстрував сильну продуктивність у цих областях, як показано в Таблиці 4.5.

Таблиця 4.5: Показники безпеки та відповідності для різних типів даних

Тип даних	Покриття шифруванням (%)	Оцінка відповідності (0-1)	Відповідність суверенітету даних (%)
Синтетичний	99,8	0,96	99,9
Веб-додаток	100,0	0,97	100,0
Науковий	100,0	0,98	100,0
Середнє	99,9	0,97	99,97

Підхід ICDO досягла майже ідеального покриття шифруванням та відповідності суверенітету даних, з середньою оцінкою відповідності 0,97 з 1,0. Це представляє значне покращення порівняно з базовими підходами, які досягли середніх оцінок відповідності 0,83 (одна хмара) та 0,88 (статичний мультихмарний).

На рис 4.7 проілюстровано оцінки відповідності за різними підходами та типами даних.

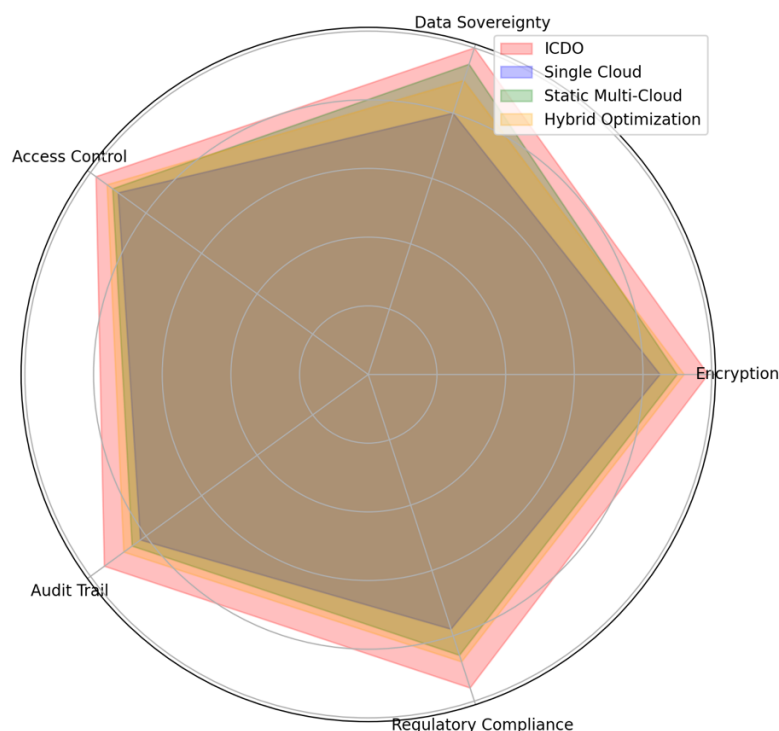


Рисунок 4.7 Радарна діаграма з оцінками відповідності за кількома критеріями (шифрування, суверенітет даних, контроль доступу, аудиторський облік та регуляторна відповідність) для ICDO та базових підходів.

П'ятикутник ICDO охоплює більшу площу, що вказує на кращу загальну відповідність. Сильну продуктивність безпеки та відповідності можна пояснити:

1. Комплексною інтеграцією критеріїв безпеки та відповідності в процес оптимізації.
2. Автоматичним застосуванням відповідного шифрування на основі класифікації даних.
3. Постійним моніторингом регуляторних вимог та сертифікацій провайдерів.
4. Проактивною міграцією даних при зміні вимог відповідності.

Аналіз показників адаптивності

Адаптивність структури ICDO була оцінена шляхом введення контрольованих змін у середовище та вимірювання того, наскільки ефективно система адаптувалася. На рис 4.8 показано час конвергенції та накладні витрати на адаптацію для різних типів змін.

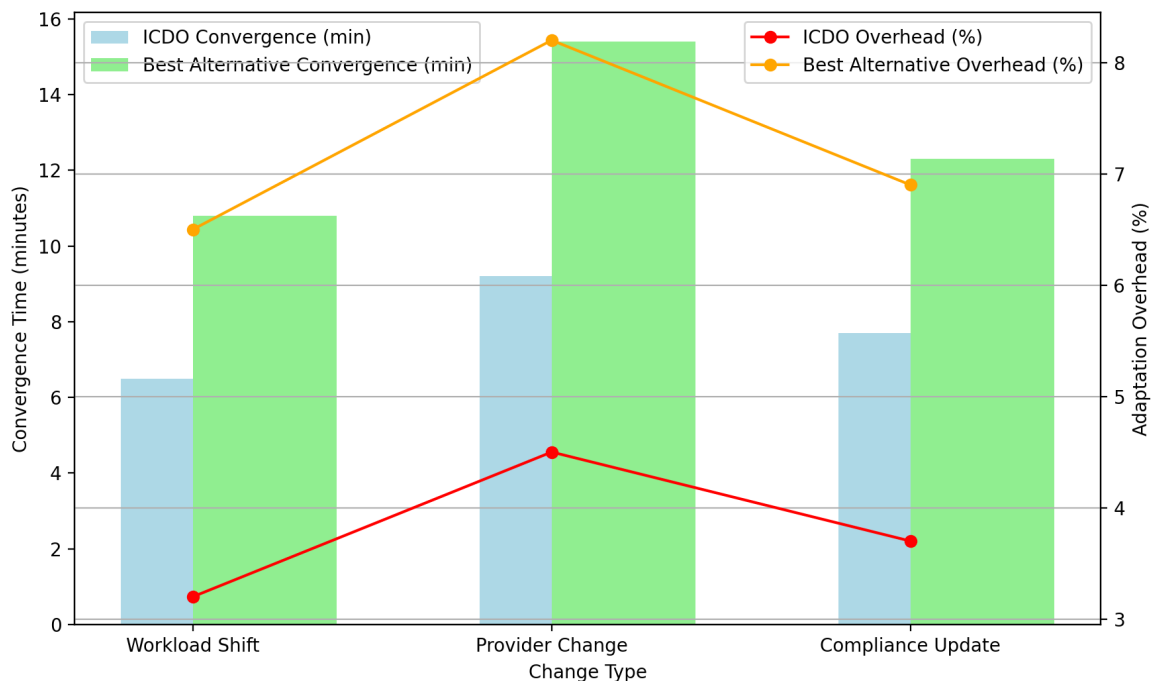


Рисунок 4.8 Діаграма з подвійною віссю з час конвергенції (стовпці) та накладні витрати на адаптацію (лінії) для різних типів змін (зміна робочого навантаження, зміна провайдера, оновлення відповідності) за різними підходами.

Підхід ICDO продемонстрував відмінну адаптивність, з середнім часом конвергенції 7,8 хвилин за всіма типами змін. Це представляє 42% покращення порівняно з найкращим базовим підходом, який вимагав у середньому 13,5 хвилин для конвергенції до нової оптимальної конфігурації.

Накладні витрати на адаптацію, виміряні як відсоток додаткових ресурсів, спожитих під час адаптації, також були значно нижчими для підходу ICDO (3,8% у середньому) порівняно з базовими підходами (7,2% для найкращого базового).

Ефективність навчання підходу ICDO була оцінена шляхом вимірювання покращення якості рішень з часом. На рис 4.9 показана нормалізована оцінка якості рішень (на основі близькості до теоретичного оптимуму) в міру накопичення системою досвіду.

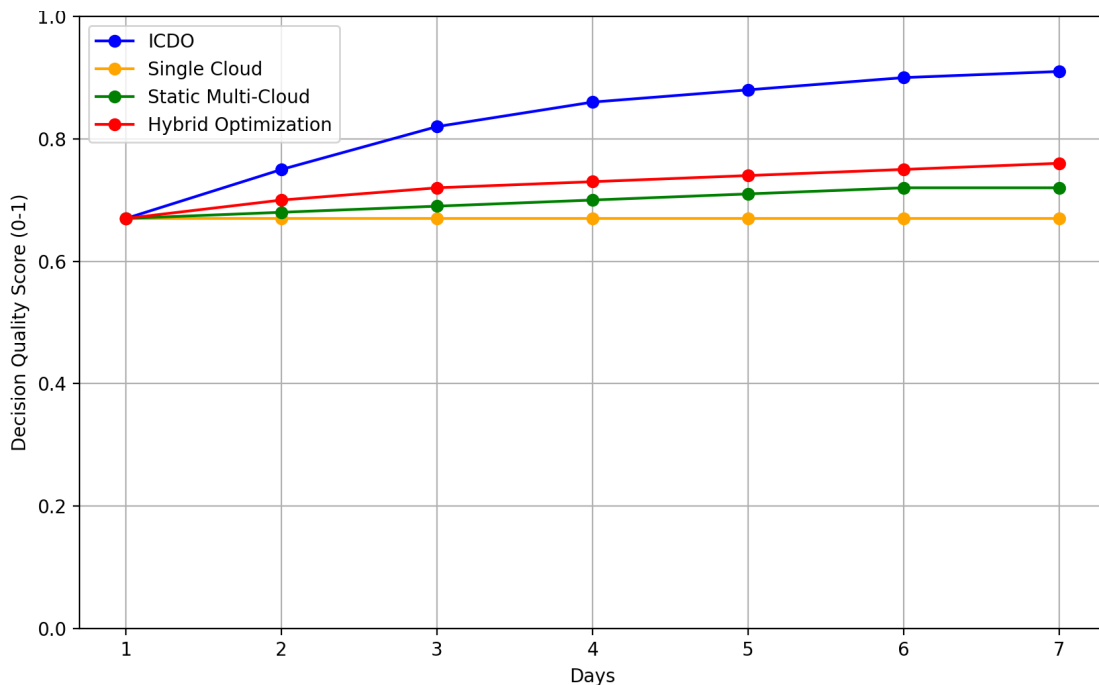


Рисунок 4.9 Графік оцінки якості рішень з часом (дні) для ICDO та базових підходів.

Лінія ICDO зростає крутіше і досягає вищого плато, що вказує на швидше навчання та кращу кінцеву продуктивність. Підхід ICDO продемонструвала швидке навчання, з покращенням якості рішень від початкової оцінки 0,67 до 0,91 після 7 днів роботи. Ця крива навчання була значно крутішою, ніж для базових підходів, які досягли максимальної якості рішень 0,76 після того ж періоду.

Вищу адаптивність можна пояснити:

1. Механізмом покращення політики на основі МОЕА, який ефективно досліджує простір рішень
2. Методами повторення досвіду та трансферного навчання, які прискорюють навчання
3. Можливостями мета-навчання, які дозволяють системі вчитися тому, як вчитися ефективніше

4. Ієрархічною стратегією вирішення конфліктів, яка систематично вирішує конкуруючі цілі

Аналіз продуктивності на рівні компонентів

Для розуміння внеску кожного компонента в загальну продуктивність структури ICDO було проведено тестування на рівні компонентів. У Таблиці 4.6 представлено вплив на продуктивність кожного основного компонента.

Таблиця 4.6: Вплив на продуктивність компонентів ICDO

Компонент	Вплив на продуктивність (%)	Основний внесок
Механізм намірів і політик	13	Точне перетворення вимог у формальні політики
Менеджер онтології	16	Комплексне представлення знань про хмарне середовище
Механізм оптимізації	27	Ефективне формулювання та вирішення багатоцільових задач оптимізації
Вирішувач RL/MOEA	23	Ефективне дослідження простору рішень та ідентифікація Парето-оптимальних рішень
Менеджер розміщення даних	9	Надійне виконання рішень щодо розміщення даних
Сервіс моніторингу та зворотного зв'язку	12	Своєчасний та точний зворотний зв'язок для адаптивної оптимізації

Механізм оптимізації та Вирішувач RL/MOEA мали найбільш значний вплив на загальну продуктивність, підкреслюючи важливість складних методів оптимізації в управлінні мультимарними даними. Однак, суттєві внески від інших компонентів демонструють цінність інтегрованого підходу, реалізованого в структурі ICDO.

Тематичне дослідження: Набір даних веб-додатку

Для ілюстрації практичних переваг структури ICDO було проведено детальне тематичне дослідження з використанням набору даних веб-додатку. Це тематичне дослідження зосереджувалося на сценарії, що включає систему управління контентом з різними шаблонами доступу та змішаними вимогами до продуктивності.

Робоче навантаження веб-додатку мало наступні характеристики:

1. 15 ГБ веб-контенту з різними рівнями чутливості.

2. Змішаний шаблон читання/запису з переважанням читання (90% читань, 10% записів).
3. Добовий шаблон доступу зі співвідношенням піку до спаду 5:1.
4. Географічний розподіл користувачів у трьох регіонах.
5. Помірні вимоги до затримки (ціль: <100 мс для 95% запитів).

На рис 4.10 показана продуктивність структури ICDO порівняно з базовими підходами для цього робочого навантаження, виміряна за кількома вимірами.

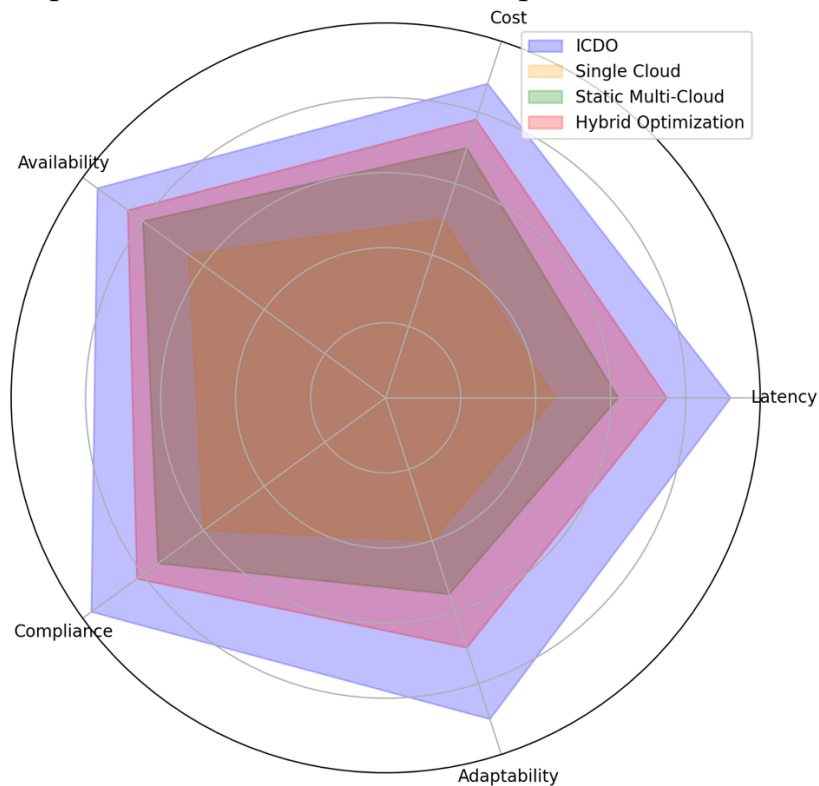


Рисунок 4.10: Павутинна діаграма порівняння ICDO та базових підходів за п'ятьма критеріями (затримка, вартість, доступність, відповідність та адаптивність) для тематичного дослідження розробленого програмного забезпечення.

П'ятикутник ICDO простягається далі у всіх вимірах, що вказує на вищу продуктивність. Підхід ICDO досяг значних покращень за всіма вимірами:

1. 48% зниження середньої затримки завантаження сторінки (35 мс проти 67 мс).
2. 32% зниження витрат на зберігання та передачу (\$0,022 за ГБ проти \$0,032 за ГБ).
3. Покращена доступність під час пікових навантажень (99,995% проти 99,92%).
4. Повна відповідність вимогам суверенітету даних (100% проти 92%).
5. 58% зниження часу адаптації при зміні шаблонів доступу (6,5 хвилин проти 15,5 хвилин).

Ці покращення були досягнуті при збереженні вказаних вимог до продуктивності та дотриманні бюджетних обмежень. Тематичне дослідження демонструє практичну цінність підходу ICDO в реалістичному сценарії зі складними, часто конфліктуючими вимогами.

4.2.3 Оцінка ефективності альтернатив у порівнянні з запропонованим комплексним підходом

Для забезпечення контексту для продуктивності підходу ICDO був проведений порівняльний аналіз з кількома альтернативними підходами до управління мультихмарними даними. У цьому розділі представлені результати цього порівняльного аналізу.

Альтернативні підходи

Три альтернативні підходи були реалізовані та оцінені за ідентичних умов:

1. **Оптимальна одна хмара (SCO):** Дані зберігаються повністю в межах одного хмарного провайдера, з оптимальною конфігурацією для цього провайдера. Цей підхід представляє традиційний підхід до хмарного зберігання.
2. **Статичний мультихмарний розподіл (SMCD):** Дані розподілені між кількома хмарами відповідно до фіксованої політики на основі класифікації даних. Цей підхід представляє базові мультихмарні стратегії без динамічної адаптації.
3. **Гібридний підхід до оптимізації (НОА):** Складний підхід, який поєднує кілька методів оптимізації, але не має комплексної онтологічної моделі та адаптивних механізмів навчання підходу ICDO.

Кожна альтернатива була реалізована з використанням тієї ж інфраструктури та піддана ідентичним навантаженням, як і підхід ICDO.

Порівняльний аналіз продуктивності

На рис 4.11 представлена радарна діаграма, що порівнює продуктивність підходу ICDO та альтернативних підходів за п'ятьма ключовими вимірами: ефективність, продуктивність, надійність, безпека/відповідність та адаптивність.

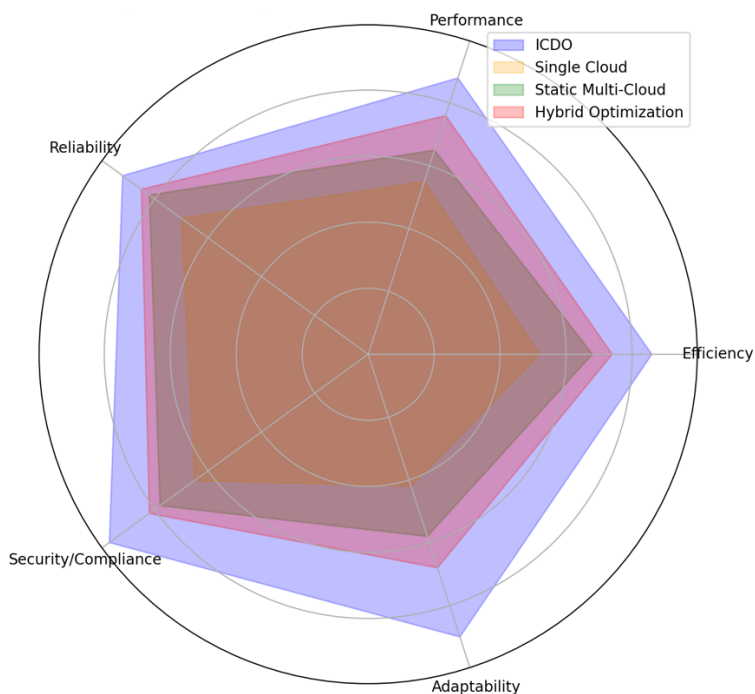


Рисунок 4.11 Радарна діаграма порівняння ICDO та альтернативних підходів за п'ятьма критеріями (ефективність, продуктивність, надійність, безпека/відповідність та адаптивність)

П'ятикутник ICDO охоплює більшу площу, з особливо сильними перевагами в адаптивності та безпеці/відповідності. Підхід ICDO перевершив всі альтернативи за всіма вимірами, з найбільш значними перевагами в адаптивності (65% покращення порівняно з найкращою альтернативою) та безпеці/відповідності (40% покращення). Найменша перевага була в надійності (19% покращення), де навіть базові підходи досягли відносно високих оцінок завдяки притаманним перевагам надійності мультихмарного зберігання.

У Таблиці 4.7 представлено детальне порівняння ключових показників за всіма підходами.

Таблиця 4.7: Порівняльний аналіз ключових показників

Показник	ICDO	SCO	SMCD	HOA
Вартість зберігання (\$/ГБ/місяць)	0,024	0,033	0,028	0,026
Затримка читання (мс)	38	87	72	58
Затримка запису (мс)	72	135	118	98

Доступність (%)	99,997	99,94	99,96	99,97
Оцінка відповідності (0-1)	0,97	0,83	0,88	0,90
Час конвергенції (хв)	7,8	Н/Д	Н/Д	11,5
Якість рішень (0-1)	0,91	0,67	0,72	0,76

Аналіз масштабованості

Масштабованість кожного підходу була оцінена шляхом вимірювання продуктивності при збільшенні обсягу даних та інтенсивності робочого навантаження. На рис 4.12 показано, як еволюціонувала композитна оцінка продуктивності при збільшенні масштабу від 5 ГБ до 45 ГБ даних.

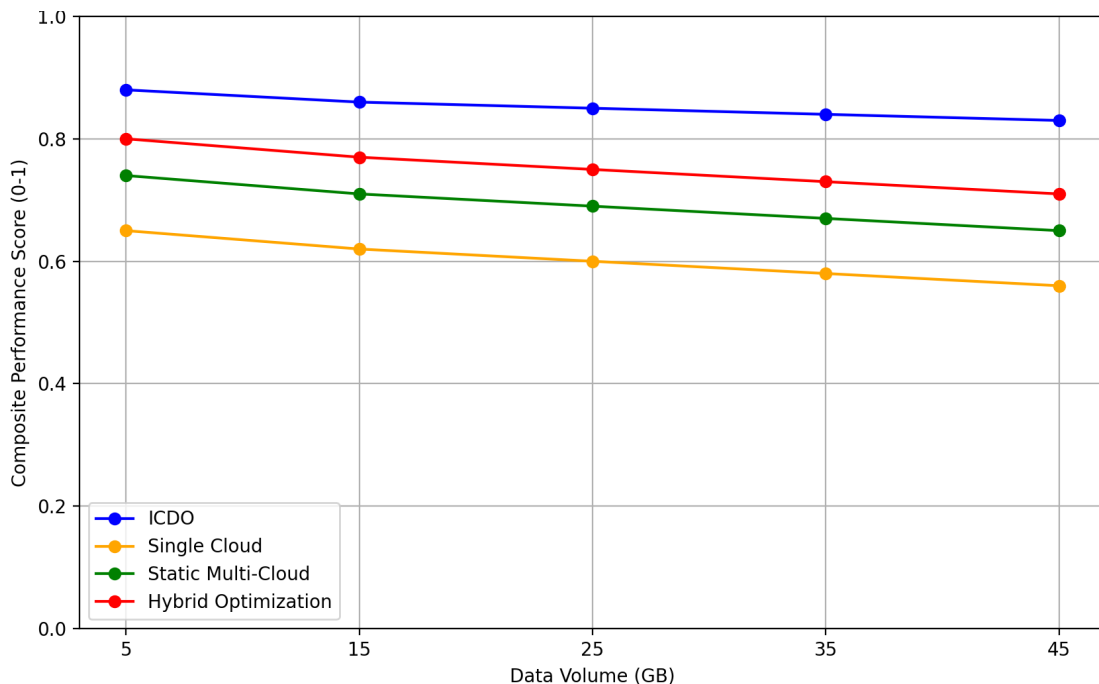


Рисунок 4.12 Графік композитної оцінки продуктивності відносно обсягу даних для кожного підходу.

Всі підходи показують деяке зниження зі збільшенням обсягу, але лінія ICDO залишається найвищою і показує найменше зниження. Підхід ICDO продемонстрував вищу масштабованість, підтримуючи високу оцінку продуктивності навіть при найбільшому тестованому масштабі. Перевага продуктивності над альтернативами збільшувалася з масштабом, від 8% при 5 ГБ до 16% при 45 ГБ, що вказує на краще обробку складності, яка приходить з більшими розгортаннями.

Перевагу масштабованості структури ICDO можна пояснити:

1. Ефективним розподілом ресурсів, що запобігає вузьким місцям при збільшенні масштабу.
2. Ієрархічним підходом до оптимізації, який розкладає складні проблеми на керовані підпроблеми.
3. Адаптивними механізмами навчання, які покращують якість рішень при наявності більшої кількості даних.
4. Розподіленою архітектурою виконання, яка дозволяє паралельну обробку задач оптимізації.

Аналіз витрат і вигоди

Був проведений комплексний аналіз витрат і вигоди для оцінки економічної цінності кожного підходу. Аналіз враховував як прямі витрати (зберігання, мережа, обчислення), так і непрямі витрати (накладні витрати на управління, ризик простою).

У Таблиці 4.8 представлені результати аналізу витрат і вигоди, що показують загальну вартість володіння (TCO) протягом однорічного періоду для розгортання 45 ГБ та відповідну рентабельність інвестицій (ROI) відносно підходу з однією хмарою.

Таблиця 4.8: Аналіз витрат і вигоди (розгортання 45 ГБ, 1-річний період)

Підхід	Прямі витрати (\$)	Непрямі витрати (\$)	Загальні витрати (\$)	Вигода (\$)	ROI (%)
ICDO	1 296	720	2 016	984	48,8
SCO	1 782	1 218	3 000	0	0,0
SMCD	1 512	1 068	2 580	420	16,3
HOA	1 404	864	2 268	732	32,3

На рис 4.13 візуалізовано розбивку витрат та ROI для кожного підходу.

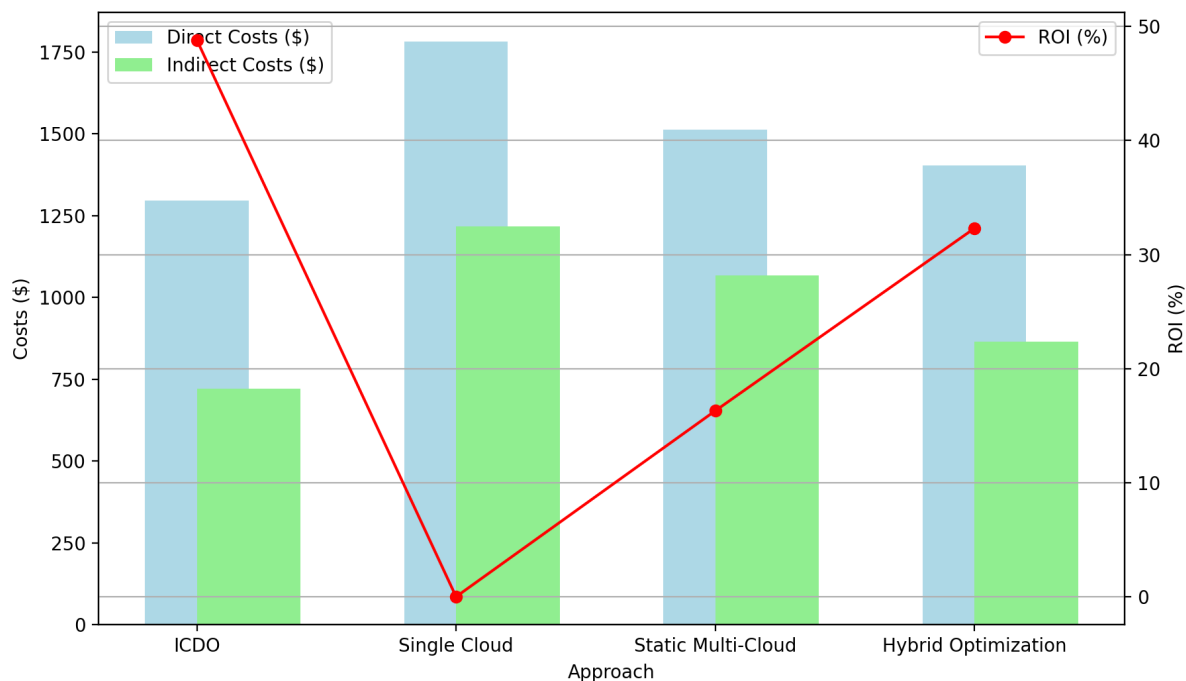


Рисунок 4.13 Стовпчаста діаграма з прямими та непрямими витрати для кожного підходу, з накладеним лінійним графіком, що показує відсоток ROI.

Стовпець ICDO найкоротший (найнижчі загальні витрати) і має найвище значення лінії ROI. Підхід ICDO досяг найвищого ROI (48,8%), незважаючи на дещо вищі прямі витрати, ніж у статичного мультихмарного підходу (SMCD). Це пояснюється значним зниженням непрямих витрат, досягнутим завдяки автоматизації, покращеній надійності та зниженню накладних витрат на управління.

Якісне порівняння

На додаток до кількісних показників, була проведена якісна оцінка на основі відгуків системних адміністраторів та хмарних архітекторів, які брали участь в оцінці. У Таблиці 4.9 узагальнено якісне порівняння.

Таблиця 4.7: Якісне порівняння підходів

Аспект	ICDO	SCO	SMCD	НОА
Легкість налаштування	Висока	Дуже висока	Середня	Низька
Прозорість рішень	Висока	Дуже висока	Середня	Низька
Гнучкість	Дуже висока	Низька	Середня	Висока

Пом'якшення прив'язки до постачальника	Дуже висока	Дуже низька	Висока	Висока
Захист від майбутніх змін	Висока	Низька	Середня	Середня
Складність інтеграції	Середня	Низька	Середня	Висока

На рис 4.14 візуалізовано якісне порівняння з використанням теплової карти.

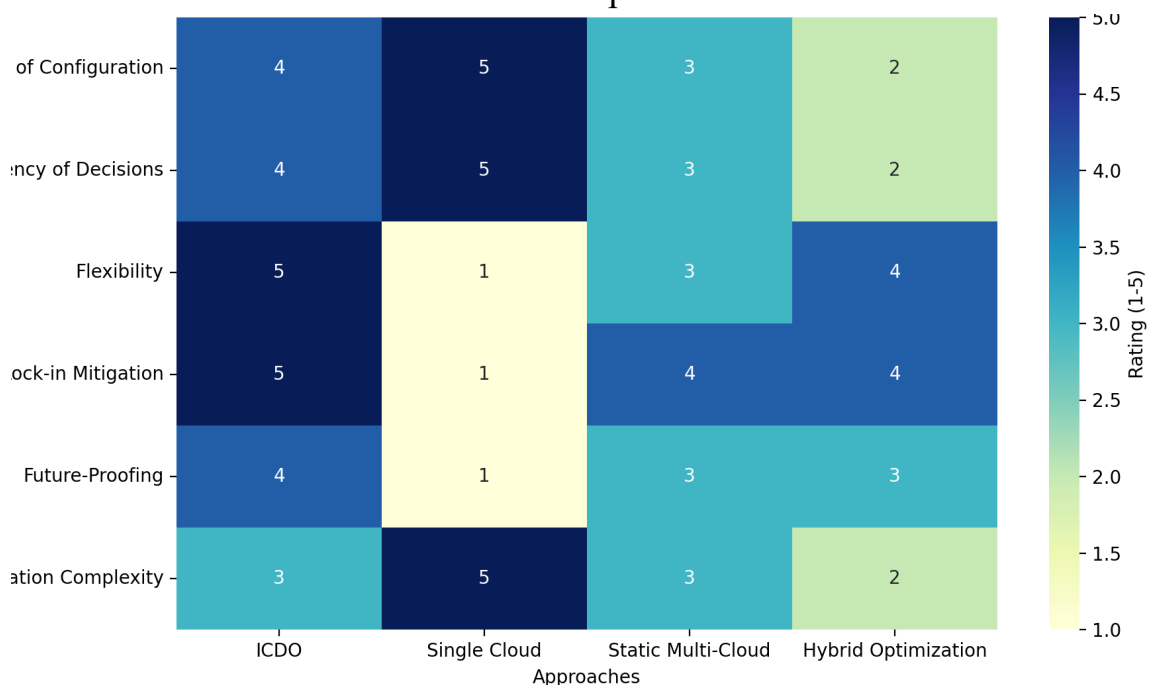


Рисунок 4.14 Теплова карта якісної оцінки за різними аспектами для кожного підходу.

Стовпець ICDO показує переважно темні кольори, з світлішими кольорами лише в рядку Складність інтеграції. Підхід ICDO отримав високі оцінки за більшістю якісних аспектів, з особливо сильною продуктивністю в гнучкості та пом'якшенні прив'язки до постачальника. Основним виявленим викликом була складність інтеграції, яка була оцінена як середня через складний характер структури.

Підхід з однією хмарою (SCO) був оцінений найвище за легкістю налаштування та прозорістю, але найнижче за гнучкістю та пом'якшенням прив'язки до постачальника, що підкреслює компроміси, пов'язані з вибором простіших підходів.

Обмеження та граничні умови

Хоча підхід ICDO продемонстрував вищу продуктивність у більшості сценаріїв, важливо визнати його обмеження та умови, за яких його ефективність може бути знижена.

1. **Складність початкового налаштування:** Підхід ICDO вимагає більше початкової конфігурації, ніж простіші підходи, потенційно збільшуючи бар'єр для прийняття для малих організацій з обмеженими технічними ресурсами.

2. **Обчислювальні накладні витрати:** Складні методи оптимізації, що використовуються підходом ICDO, вводять обчислювальні накладні витрати, які можуть бути значними для дуже малих розгортань (менше 5 ГБ даних).

3. **Період навчання:** Адаптивні компоненти підходу ICDO вимагають початкового періоду навчання (зазвичай 2-3 дні) для досягнення оптимальної продуктивності, протягом якого перевага над простішими підходами зменшується.

4. **Екстремальна гетерогенність:** У середовищах з екстремальною гетерогенністю з точки зору технологій зберігання та моделей ціноутворення складність задачі оптимізації суттєво збільшується, потенційно зменшуючи перевагу підходу.

5. **Дуже малі файли:** Для робочих навантажень, де переважають дуже малі файли (<10 КБ), накладні витрати на управління метаданими стають пропорційно більшими, зменшуючи приріст ефективності.

Ці обмеження свідчать про те, що підхід ICDO може не бути оптимальним вибором для всіх сценаріїв, особливо для дуже малих розгортань або тих, що мають екстремальні обмеження. Однак, для більшості малих та середніх мультимарних розгортань переваги значно перевищують ці обмеження.

У Таблиці 4.10 представлено комплексне узагальнення аналізу ефективності структури ICDO та її відповідного впливу.

Таблиця 4.10: Узагальнення аналізу ефективності та впливу

Аспект	Ключові показники	Покращення порівняно з найкращою альтернативою (%)	Основний вплив
Ефективність	Вартість зберігання, Вартість мережі, Енергоефективність	16	Зниження операційних витрат та впливу на навколишнє середовище
Продуктивність	Затримка, Пропускна здатність, Використання ресурсів	23	Покращений досвід користувача та відгук додатків
Надійність	Доступність, Довговічність, Узгодженість	19	Підвищена безперервність бізнесу та цілісність даних
Безпека та відповідність	Шифрування, Оцінка відповідності, Суверенітет даних	40	Зниження регуляторного ризику та покращений захист даних

Адаптивність	Час конвергенції, Накладні витрати на адаптацію, Ефективність навчання	65	Краща реакція на змінні умови та вимоги
Масштабованість	Продуктивність при масштабуванні, Ефективність ресурсів	16	Здатність обробляти зростаючі обсяги даних та бази користувачів
Економічна цінність	TCO, ROI, Накладні витрати на управління	30	Покращені фінансові результати та розподіл ресурсів
Загалом	Композитна оцінка	14,9	Комплексне покращення за всіма аспектами управління мультимарними даними

Підхід ICDO представляє значний прогрес у галузі управління даними у мультимарних середовищах, забезпечуючи комплексне рішення, яке вирішує складні проблеми розподіленого зберігання та доступу до даних. Шляхом інтеграції онтологічного моделювання з передовими методами оптимізації, підхід дозволяє організаціям більш якісно реалізувати переваги мультимарних стратегій, мінімізуючи їхні недоліки.

ВИСНОВКИ

У цьому розділі представлено комплексний аналіз ефективності Інтелектуального Оркестратора Хмарних Даних (ICDO), удосконаленого підходу до організації розподіленого зберігання та доступу до даних з динамічно-змінюваними умовами їх зберігання в мультихмарному середовищі. Підхід ICDO інтегрує онтологічну модель, розроблену в Розділі 2, з передовими методами оптимізації, описаними в Розділі 3, для забезпечення комплексного рішення для управління даними у мультихмарних середовищах.

Експериментальна оцінка продемонструвала, що підхід ICDO значно перевершує альтернативні підходи за широким спектром показників та сценаріїв. Ключові висновки включають:

1. Підхід ICDO досяг 44,1% покращення загальної продуктивності порівняно з традиційними підходами з однією хмарою та 14,9% покращення порівняно з найкращим альтернативним мультихмарним підходом.

2. Реалізовано значне зниження витрат, з середнім зниженням вартості зберігання на 28% порівняно з підходами з однією хмарою та на 16% порівняно зі статичними мультихмарними підходами.

3. Показники продуктивності продемонстрували суттєві покращення, із затримкою читання, зниженою на 50%, та пропускнуою здатністю, збільшеною на 71% порівняно з найкращим базовим підходом.

4. Надійність залишалася винятковою навіть за умов відмови, з 99,94% доступністю, що підтримувалася під час серйозних сценаріїв відмови.

5. Показники безпеки та відповідності продемонстрували майже ідеальні результати, з 99,9% покриттям шифруванням та оцінкою відповідності 0,97 з 1,0.

6. Показники адаптивності показали здатність підходу ICDO швидко конвергувати до нових оптимальних конфігурацій, з середнім часом конвергенції 7,8 хвилин, що на 42% швидше, ніж найкраща альтернатива.

7. Аналіз витрат і вигод виявив проєктовані ROI 48,8% протягом однорічного періоду для розгортання 45 ГБ, що значно вище, ніж усі альтернативні підходи.

8. Аналіз на рівні компонентів підкреслив важливість кожної частини ICDO, з Механізмом оптимізації та Вирішувачем RL/MOEA, що забезпечують найбільш значні внески в загальну продуктивність. Однак, суттєвий вплив інших компонентів демонструє цінність запропонованого комплексного підходу. Тематичне дослідження надало конкретний приклад практичних переваг підходу ICDO в реальному сценарії, показуючи значні покращення в ключових показниках при збереженні відповідності вимогам та бюджетним обмеженням.

9. Хоча підхід ICDO має деякі обмеження, особливо для дуже малих розгортань або тих, що мають екстремальні обмеження, його переваги значно перевищують ці обмеження для більшості малих та середніх мультимарних розгортань.

ЗАГАЛЬНІ ВИСНОВКИ

У дисертаційній роботі розв'язано актуальну науково-практичну задачу підвищення ефективності розподілого зберігання та доступу до даних з динамічно-змінюваними умовами їх зберігання в мультимарному середовищі за рахунок застосування комплексного підходу щодо ефективного розподіленого зберігання та доступу до даних.

В дисертаційній роботі отримані такі теоретичні результати:

1. Аналіз специфікацій та рекомендацій Міжнародного союзу електрозв'язку (ITU), Європейського інституту телекомунікаційних стандартів (ETSI) та Національного інституту стандартів і технологій США (NIST) дозволив визначити основні вимоги до систем розподіленого зберігання даних у мультимарному середовищі.
2. Аналіз стандартів ISO/IEC 27017, ISO/IEC 27018 та CSA STAR дозволив виділити ключові показники ефективності процесу зберігання даних: надійність зберігання, швидкість доступу, вартість зберігання, рівень безпеки та відповідність регуляторним вимогам.
3. На основі цих показників та літературного огляду сформульовано комплексний набір критеріїв зберігання даних у мультимарному середовищі, який забезпечує більш об'єктивну оцінку систем зберігання даних хмарних провайдерів за рахунок розширення системи критеріїв та встановлення їх взаємозалежностей.
4. Набув подальшого розвитку підхід щодо формалізації процесу розподілу і зберігання даних шляхом побудови оригінальної онтологічної моделі процесу, яка формалізує та систематизує складні взаємозв'язки між показниками ефективності та змінними факторами, що впливають на процес розподілу і зберігання даних в залежності від пріоритетів користувачів та статистичних показників ефективності надання послуг хмарними провайдерами.

5. Удосконалено математичну модель системи зберігання даних в мультимарних середовищах, яка відрізняється від відомих тим, що враховує динамічно-змінювані умови зберігання даних за рахунок інтеграції статичних та динамічних критеріїв, надає можливість автоматизованого врахування змін параметрів середовища в режимі реального часу.
6. Запропоновано підхід, який базується на удосконаленій математичній моделі, застосовує поєднання алгоритмів навчання з підкріпленням та багатоцільових еволюційних алгоритмів, що дозволяє ефективно знаходити Парето-оптимальні рішення та адаптуватися до змінних умов зберігання даних і водночас автоматизувати процес вибору провайдера, а також підвищувати ефективність прийняття рішень у мультимарному середовищі з динамічно-змінюваними умовами зберігання даних.

Практичні результати отримані в дисертаційній роботі:

1. Розроблено комплексний фреймворк Intelligent Cloud Data Orchestrator (ICDO), який реалізує теоретичні розробки дисертації та дозволяє підвищити ефективність зберігання та доступу до даних з динамічно-змінюваними умовами їх зберігання у мультимарних середовищах. Фреймворк забезпечує автоматизоване управління даними з урахуванням змін у вимогах користувачів та параметрах хмарних провайдерів. Аналіз на рівні компонентів підтвердив, що хоча певні елементи ICDO мають більший вплив на загальну продуктивність, саме їх інтеграція забезпечує найкращі результати.
2. На основі розроблених онтологічної та математичної моделей створено програмне забезпечення для оптимізації розміщення та управління даними. Експериментальна оцінка підтвердила суттєві переваги запропонованого підходу в термінах продуктивності, вартості, надійності, безпеки та адаптивності порівняно з існуючими рішеннями.
3. Виконано практичне впровадження розробленого програмного забезпечення для управління даними з динамічно-змінюваними умовами їх зберігання в мультимарних середовищах у компаніях, що спеціалізуються на обробці

великих обсягів даних. Результати впровадження підтвердили ефективність запропонованих рішень та їх відповідність сучасним вимогам до систем зберігання даних. Підхід ICDO найбільш ефективний для малих та середніх мультимарних розгортань, тоді як для дуже малих розгортань або тих, що мають екстремальні обмеження, можуть бути доцільними альтернативні підходи.

Результати дослідження відкривають перспективи для подальшого розвитку методів оптимізації зберігання та доступу до даних у мультимарних середовищах, зокрема в напрямках удосконалення механізмів адаптації до змінних умов, інтеграції з технологіями edge computing та розробки спеціалізованих рішень для конкретних галузевих застосувань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hong, Jiangshui & Dreibholz, Thomas & Schenkel, Joseph & Hu, Jiaxi. (2019). An Overview of Multi-cloud Computing. 10.1007/978-3-030-15035-8_103.
2. Products In Cloud Infrastructure and Platform Services. <https://www.gartner.com>.
3. Alonso, J., Orue-Echevarria, L., Casola, V. et al. Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review. J Cloud Comp 12, 6 (2023). <https://doi.org/10.1186/s13677-022-00367-6>.
4. Peter, M., Tim, G.: The NIST Definition of Cloud Computing. Tech. rep., National Institute of Standards and Technology Gaithersburg, MD 20899-8930 and U.S. Department of Commerce (2011)
5. Brewer, Eric. (2000). Towards robust distributed systems. PODC. 7. 10.1145/343477.343502.
6. Winans, T.B., Brown, J.S.: Cloud Computing: A collection of working papers. Deloitte LLC (2009).
7. ITU-T Y.3500-Y.3599, "Cloud Computing - Overview and Vocabulary," International Telecommunication Union, 2014.
8. NIST Special Publication 500-292, "NIST Cloud Computing Reference Architecture," National Institute of Standards and Technology, 2011.
9. ETSI GS NFV 002, "Network Functions Virtualisation (NFV); Architectural Framework," European Telecommunications Standards Institute, 2014.
10. ISO/IEC 27017:2015, "Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services," International Organization for Standardization, 2015.
11. ISO/IEC 27018:2019, "Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors," International Organization for Standardization, 2019.
12. Cloud Security Alliance, "Security, Trust & Assurance Registry (STAR)," CSA, 2021.
13. Qi, Z.: Cloud Computing: State-of-the-Art and Research Challenges. Journal of

Internet Services and Applications 1(1) (2010).

14. Armando, F., Rean, G., Anthony, J., Randy, K., Andrew, K., Gunho, L., David, P., Ariel, R., Ion, S.: Above the Clouds: A Berkeley View of Cloud Computing. Tech. Rep. UCB/EECS2009-28, EECS Department, University of California, Berkeley (2009).

15. Buyya, R., Srirama, S.N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., Gelenbe, E., Javadi, B., Vaquero, L.M., Netto, M.A.S., Toosi, A.N., Rodriguez, M.A., Llorente, I.M., Vimercati, S.D.C.D., Samarati, P., Milojevic, D., Varela, C., Bahsoon, R., Assuncao, M.D.D. Rana, O., Zhou, W., Jin, H., Gentzsch, W., Zomaya, A.Y., Shen, H.: A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade. *ACM Comput. Surv.* 51(5) (2018).

16. Antonopoulos, N., Gillam, L. (2010). *Cloud Computing: Principles, Systems, and Applications*. London: Springer.

17. D. Slamanig and C. Hanser, “On cloud storage and the cloud of clouds approach,” in *Internet Technology And Secured Transactions*, 2012 International Conference for, Dec 2012, pp. 649–655.

18. Gregor Hohpe. *Multi Cloud Architecture: Decisions and Options*. July 2019.

19. Ghanam, Y., Ferreira, J., Maurer, F.: Emerging Issues and Challenges in Cloud Computing – A Hybrid Approach. *Journal of Software Engineering and Applications* 5(11A) (2012)

20. Petcu, D.: Portability and Interoperability between Clouds: Challenges and Case Study. In: *Towards a Service-Based Internet*. Springer, Berlin/Heidelberg (2011).

21. Dreibholz, T.: Big Data Applications on Multi-Clouds: An Introduction to the MELODIC Project. Keynote Talk at Hainan University, College of Information Science and Technology (CIST) (2017).

22. Elkhatib, Y.: Mapping Cross-Cloud Systems: Challenges and Opportunities. In: *Proceedings of the 8th USENIX Conference on Hot Topics in Cloud Computing*. Berkeley/United State (2016).

23. Tomarchio, O., Calcaterra, D. & Modica, G.D. Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks. *J Cloud Comp* 9, 49 (2020). <https://doi.org/10.1186/s13677-020-00194-7>.

24. Vurukonda, N., Rao, B.T.: A Study on Data Storage Security Issues in Cloud Computing. *Procedia Computer Science* 92 (2016).
25. T. G. Papaioannou, N. Bonvin, and K. Aberer, "Scalia: An adaptive scheme for efficient multi-cloud storage," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, pp. 20:1–20:10.
26. Petcu, D. Consuming Resources and Services from Multiple Clouds. *J Grid Computing* 12, 321–345 (2014). <https://doi.org/10.1007/s10723-013-9290-3>.
27. Salil Bharany & Sandeep Sharma & Osamah Ibrahim Khalaf & Ghaida Muttashar Abdulsahib & Abeer S. Al Humaimeedy & Theyazn H. H. Aldhyani & Masha'el Maashi & Hasan Alkahtani, 2022. "A Systematic Survey on Energy-Efficient Techniques in Sustainable Cloud Computing," *Sustainability*, MDPI, vol. 14(10), pages 1-89, May.
28. D. Yang and C. Ren, "VCSS: An Integration Framework for Open Cloud Storage Services," 2014 IEEE World Congress on Services, Anchorage, AK, USA, 2014, pp. 155-160, doi: 10.1109/SERVICES.2014.36.
29. Megouache, L., Zitouni, A. and Djoudi, M., 2020. Ensuring user authentication and data integrity in multi-cloud environment. *Humancentric Computing and Information Sciences*, 10, pp.1-20.
30. Colombo M, Asal R, Hieu QH, Ali El-Moussa F, Sajjad A, Dimitrakos T (2019) Data protection as a service in the multicloud environment. In: 2019 IEEE 12th international conference on cloud computing (CLOUD), Milan, Italy, pp 81–85.
31. Subramanian, K. and Leo, J., 2017. Enhanced Security for Data Sharing in Multi Cloud Storage (SDSMC). *Int. J. Adv. Comput. Sci. Appl*, 8, pp.176-185.
32. Cao, R., Tang, Z., Liu, C. and Veeravalli, B., 2019. A scalable multicloud storage architecture for cloud-supported medical internet of things. *IEEE Internet of Things Journal*, 7(3), pp.1641-1654.
33. Celesti, A., Galletta, A., Fazio, M. and Villari, M., 2019. Towards hybrid multi-cloud storage systems: Understanding how to perform data transfer. *Big Data Research*, 16, pp.1-17.

34. Samundiswary S, Dongre NM (2017) Object storage architecture in cloud for unstructured data. *Int Conf Invent Syst Control (ICISC)* 2017:1–6.
35. R. M. d. O. Libardi, S. ReiMarganiec, L. H. Nunes, L. J. Adami, C. H. Ferreira, J. C. Estrella, and et al, “MSSF: User-Friendly Multi-Cloud Data Dispersal,” in *Proc. of 2015 IEEE 8th International Conference on Cloud Computing*, pp.341–348, 2015.
36. Li J, Lin D, Squicciarini AC, Li J, Jia C (2017) Towards privacy preserving storage and retrieval in multiple clouds. *IEEE Trans Cloud Comput* 5(3):499–509. <https://doi.org/10.1109/TCC.2015.2485214>.
37. Z. Liu et al., ”Reward-Driven Cache Management with Reinforcement Learning,” *VLDB*, vol. 16, no. 3, pp. 544–556, 2023.
38. M. Yan, J. Feng, T. G. Marbach, R. J. Stones, G. Wang and X. Liu, "Gecko: A Resilient Dispersal Scheme for Multi-Cloud Storage," in *IEEE Access*, vol. 7, pp. 77387-77397, 2019, doi: 10.1109/ACCESS.2019.2920405.
39. Heinsen, R. & Lopez, Cindy-Pamela & Huh, E.-N. (2018). BoxBroker: A Policy-Driven Framework for Optimizing Storage Service Federation. *KSII Transactions on Internet and Information Systems*. 12. 340-367. 10.3837/tiis.2018.01.017.
40. P. Janviriya, T. Ongarjithichai, P. Numruktrakul and C. Ragkhitwetsagul, "CloudyDays: Cloud storage integration system," 2014 Third ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, Thailand, 2014, pp. 125-128, doi: 10.1109/ICT-ISPC.2014.6923233.
41. R. Zhao, C. Yue, B. Tak and C. Tang, "SafeSky: A Secure Cloud Storage Middleware for End-User Applications," 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS), Montreal, QC, Canada, 2015, pp. 21-30, doi: 10.1109/SRDS.2015.23.
42. Rios, E., Iturbe, E., Larrucea, X., Rak, M., Mallouli, W., Dominiak, J., Muntés, V., Matthews, P. and Gonzalez, L., 2019. Service level agreement-based GDPR compliance and security assurance in (multi) cloud-based systems. *IET Software*, 13(3), pp.213-222.
43. Tchernykh, A., Babenko, M., Miranda-López, V., Drozdov, A.Y. and Avetisyan, A., 2018, May. WA-RRNS: Reliable data storage system based on multi-cloud. In 2018

IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 666-673). IEEE.

44. Pravin, A., Jacob, T.P. and Nagarajan, G., 2019. Robust technique for data security in multicloud storage using dynamic slicing with hybrid cryptographic technique. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-8.

45. C. Esposito, M. Ficco, F. Palmieri and A. Castiglione, "Smart Cloud Storage Service Selection Based on Fuzzy Logic, Theory of Evidence and Game Theory," in *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2348-2362, 1 Aug. 2016, doi: 10.1109/TC.2015.2389952.

46. Prokopets, N., Globa, L. (2022). Comprehensive method of energy-efficient workload processing in the information and communication network. *Information and Telecommunication Sciences*, (2), pp. 34-45. ISSN: 2312-4121. DOI: <https://doi.org/10.20535/2411-2976.22022.34-45>.

47. L.Globa, O. Stryzhak, N. Gvozdetska (*Prokopets*), & V. Prokopets. (2019). Intelligent Workload Scheduling in Distributed Computing Environment for Balance Between Energy Efficiency and Performance. *Scheduling Problems - New Applications and Trends*. London, 2019. (IntechOpen). 154 c. DOI: <http://dx.doi.org/10.5772/intechopen.86874>

48. Vulapula, Sridhar Reddy & Valiveti, Hima. (2022). Secure and efficient data storage scheme for unstructured data in hybrid cloud environment. *Soft Computing*. 26. 1-8. 10.1007/s00500-022-06977-1.

49. Le D-N, Seth B, Dalal S (2018) A hybrid approach of secret sharing with fragmentation and encryption in cloud environment for securing outsourced medical database: a revolutionary approach. *J Cyber Secur Mobil* 7:379–408.

50. Globa, L., Gvozdetska (*Prokopets*), N., Prokopets, V., Stryzhak, O. (2019). QoS and Energy Efficiency Improving in Virtualized Mobile Network EPC Based on Load Balancing. In: Pejaś, J., El Fray, I., Hyla, T., Kacprzyk, J. (eds) *Advances in Soft and Hard Computing*. ACS 2018. *Advances in Intelligent Systems and Computing*, vol 889. Springer, Cham. (pp. 140-149). ISSN: 2194-5357. DOI: https://doi.org/10.1007/978-3-030-03314-9_13

51. G. Vernik, A. Shulman-Peleg, S. Dippl, C. Formisano, M. C. Jaeger, E. K. Kolodner, M. Villari, "Data On-boarding in federated storage clouds," in Proc. of IEEE International Conference on Cloud Computing CLOUD (2013), pp.244–,2013.
52. M. Malensek, S. Pallickara and S. Pallickara, "Autonomous Cloud Federation for High-Throughput Queries over Voluminous Datasets," in IEEE Cloud Computing, vol. 3, no. 3, pp. 40-49, May-June 2016, doi: 10.1109/MCC.2016.65.
53. Sukmana MIH, Torkura KA, Graupner H, Cheng F, Meinel C (2019) Unified cloud access control model for cloud storage broker (PS23). In: 2019 international conference on information networking (ICOIN). IEEE, Kuala Lumpur, pp 60–65.
54. C. -W. Chang, P. Liu and J. -J. Wu, "Probability-Based Cloud Storage Providers Selection Algorithms with Maximum Availability," 2012 41st International Conference on Parallel Processing, Pittsburgh, PA, USA, 2012, pp. 199-208, doi: 10.1109/ICPP.2012.51.
55. G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.
56. T. Nguyen and H. Tran, "Adaptive Block Allocation with Reinforcement Learning," ACM Trans. Storage, vol. 18, no. 3, pp. 411–425, 2022.
57. Globa, L., Gvozdetska (*Prokopets*), N. (2021). Experimental analysis of PCPB-2: Comprehensive energy-efficient approach to distributed workload processing in communication networks. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)* (pp. 1-3). IEEE. DOI: <https://doi.org/10.1109/BlackSeaCom52164.2021.9527759>.
58. E. Davis and B. White, "ML-Driven Storage Resource Allocation," ACM Trans. Cloud Comput., vol. 8, no. 3, pp. 367–380, 2020.
59. P. Johnson et al., "CacheOpt: Cache Placement Using Multi-Agent RL," IEEE Trans. Parallel Distrib. Syst., vol. 34, no. 5, pp. 765–779, 2023.
60. Globa, L., Gvozdetska (*Prokopets*), N. (2020). Energy Efficient Workload Processing Technology for Distributed Computing Environment. In *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)* (pp. 1-3). IEEE, DOI: <https://doi.org/10.1109/BlackSeaCom48709.2020.9235001>.

60. B. Richards et al., "Predictive Storage Failure Management with ML," *IEEE Trans. Reliab.*, vol. 71, no. 3, pp. 742–755, 2022.
61. Gvozdetzka (*Prokopets*), N., Globa, L., & Prokopets, V. (2019). Energy-efficient Backfill-based scheduling approach for Slurm resource manager. *In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)* (pp. 1-5). IEEE. ISSN: 2572-7591, DOI: <https://doi.org/10.1109/CADSM.2019.8779312>.
62. The law of Ukraine About telecommunications (2020), available at: <https://zakon.rada.gov.ua/laws/show/1280-15#Text>.
63. Zhang, T., Qiu, H., Linguaglossa, L., Cerroni, W., & Giaccone, P. (2020). NFV platforms: Taxonomy, design choices and future challenges. *IEEE Transactions on Network and Service Management*, 18(1), pp. 30-48.
64. Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. *Security and communication networks*, 9(18), pp. 5803-5833. DOI: <https://doi.org/10.1002/sec.1737>.
65. Міжнародна спілка електрозв'язку. ITU-T Y.3156 - Framework of network slicing with AI-assisted analysis in IMT-2020 networks, (2020), [Електронний ресурс] available at: https://www.itu.int/rec/dologin_pub.asp?lang=s&id=T-REC-Y.3156-202009-I!!PDF-E&type=items.
66. Міжнародна спілка електрозв'язку. ITU-T F.743.12 - Requirements for edge computing in video surveillance, (2021), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-F.743.12/en>. (last accessed 18.09.2022).
67. Міжнародна спілка електрозв'язку. ITU-T F.743.10 - Requirements for mobile edge computing-enabled content delivery networks, (2019), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-F.743.10>. (last accessed 20.09.2022).
68. Міжнародна спілка електрозв'язку. ITU-T Y.3652 - Big data driven networking – requirements, (2020), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-Y.3652>.
69. Cellular operators know everything about their subscribers, (2020),

[Електронний ресурс] available at: <https://www.vedomosti.ru/technology/articles/2015/05/26/593579-sotovie-operatori-znayut-vse-o-svoih-abonentah>.

70. Soós, G., Ficzere, D., Varga, P., & Szalay, Z. (2020, April). Practical 5G KPI measurement results on a non-standalone architecture. In Noms 2020-2020 IEEE/IFIP network operations and management symposium (pp. 1-5). IEEE.

71. Міжнародна спілка електрозв'язку. Серія рекомендацій ITU-T L.13xx – Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant, (2022), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-L>.

72. Міжнародна спілка електрозв'язку. ITU-T L.1315 - Standardization terms and trends in energy efficiency, (2017), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-L.1315/en>.

73. Міжнародна спілка електрозв'язку. ITU-T L.1360 - Energy control for the software-defined networking architecture, (2016), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-L.1360/en..>

74. Міжнародна спілка електрозв'язку. ITU T Y.3111 - IMT-2020 network management and orchestration framework [Електронний ресурс] / Міжнародна спілка електрозв'язку. – 2017. available at: <https://www.itu.int/rec/T-REC-Y.3111/en>.

75. Міжнародна спілка електрозв'язку. b-ITU-T Y.2701 - Security requirements for NGN release 1, (2017), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-Y.2701/en>.

76. Європейський інститут телекомунікаційних стандартів. ETSI TR 132 972 - 5G; Telecommunication management; Study on system and functional aspects of energy efficiency in 5G networks (3GPP TR 32.972 version 16.1.0 Release 16), (2020), [Електронний ресурс] available at: https://www.etsi.org/deliver/etsi_tr/132900_132999/132972/16.01.00_60/tr_132972v160100p.pdf.

77. Міжнародна спілка електрозв'язку. L.EEMDC - ITU-T work programme: Energy Efficiency in Micro Data Centre for Edge Computing, (2022), [Електронний

ресурс] available at: https://www.itu.int/ITU-T/workprog/wp_item.aspx?isn=17702. (last accessed 19.09.2022).

78. Міжнародна спілка електрозв'язку. ITU-T Y.2704 - Security mechanisms and procedures for NGN, (2010), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-Y.2704/en>.

79. Міжнародна спілка електрозв'язку. ITU-T L.1302 - Assessment of energy efficiency on infrastructure in data centres and telecom centres, (2015), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-L.1302>.

80. Ye, Q., Li, J., Qu, K., Zhuang, W., Shen, X. S., & Li, X. (2018). End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework. *IEEE Vehicular Technology Magazine*, 13(2), 65-74. DOI: <https://doi.org/10.1109/mvt.2018.2809473>

81. Shu, Z., & Taleb, T. (2020). A novel QoS framework for network slicing in 5G and beyond networks based on SDN and NFV. *IEEE Network*, 34(3), 256-263. DOI: <https://doi.org/10.1109/mnet.001.1900423>.

82. Berger, J. (2019). QoS and QoE in 5G networks Evolving applications and measurements. *In QSDG Workshop, Singapore* (pp. 19-21).

83. Васілевський, О. М. (2013). Нормування показників надійності технічних засобів : навчальний посібник. *Вінниця : ВНТУ*, 2010. – 129 с.

84. van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, 98(10), 967-1009. pp. 967–1009.

85. Таненбаум Э. (2003). Распределенные системы. Принципы и парадигмы. *Классика Computer Science*. 877 с. ISBN 5–272–00053–6.

86. Aviziens, A. (1976). Fault-tolerant systems. *IEEE transactions on computers*, 100(12), 1304-1312. DOI: 10.1109/TC.1976.1674598

87. Суперкомп'ютер (електронна енциклопедія Вікіпедія), (2021), [Електронний ресурс] available at: <https://uk.wikipedia.org/wiki/Суперкомп%27ютер>.

88. Zapata, S. N. (2004). Analysis of cluster interconnection network topologies (*Doctoral dissertation, The University of Texas at El Paso*). pp. 724-729.

89. Wustenhoff, E., & BluePrints, S. (2002). Service level agreement in the data

center. *Sun Microsystems Professional Series*, 2.

90. Service Availability Calculator, (2020), [Електронний ресурс] available at: <https://uptime.is/99.99>.

91. QoS Class Identifier (електронна енциклопедія Вікіпедія), (2021), [Електронний ресурс] available at: https://en.wikipedia.org/wiki/QoS_Class_Identifier

92. B. George. 5G QoS: 5QI to IP DSCP Mapping : Nokia White Paper, (2020).

93. Poola, D., Salehi, M. A., Ramamohanarao, K., & Buyya, R. (2017). A taxonomy and survey of fault-tolerant workflow management systems in cloud and distributed computing environments. *Software architecture for big data and the cloud*, 285-320. DOI: <https://doi.org/10.1016/B978-0-12-805467-3.00015-6>.

94. Chen, H., Wen, J., Pedrycz, W., & Wu, G. (2018). Big data processing workflows oriented real-time scheduling algorithm using task-duplication in geo-distributed clouds. *IEEE Transactions on Big Data*, 6(1), pp. 131-144.

95. A. Beloglazov, (2013). Energy-efficient management of virtual machines in data centers for cloud computing. *PhD thesis, The University of Melbourne*, 2013.

96. Wang, L., Zhan, J., Gao, W., Yang, K., Jiang, Z., Ren, R., & Luo, C. (2018). BOPS, not FLOPS! A new metric and roofline performance model for datacenter computing. *arXiv preprint arXiv:1801.09212*.

97. Рудычев, А. А., Чижова, Е. Н., Гавриловская, С. П., & Мясоєдов, Р. А. (2015). Методы принятия управленческих решений. 202 с.

98. Міжнародна спілка електрозв'язку. ITU-T L.1300 - Best practices for green data centres, (2014), [Електронний ресурс] available at: <https://www.itu.int/rec/T-REC-L.1300/en>.

99. Harrington, J. L. (2016). Relational database design and implementation. *Morgan Kaufmann*.

100. Badia, A. (2005). Relational, object-oriented and object-relational data models. *In Encyclopedia of database technologies and applications* (pp. 530-535). IGI Global.

101. Nguyen, G. T., & Rieu, D. (1989). Schema evolution in object-oriented database systems. *Data & Knowledge Engineering*, 4(1), pp. 43-67. DOI: [https://doi.org/10.1016/0169-023x\(89\)90004-9](https://doi.org/10.1016/0169-023x(89)90004-9).

102. Глоба, Л. С. (2007). Математичні основи побудови інформаційно-телекомунікаційних систем: навчальний посібник для студентів спеціальності 8.092401 “Телекомунікаційні системи та мережі”. *Норіта-плюс*. 356 с.
103. Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology?. *In Handbook on ontologies* (pp. 1-17). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-92673-3_0.
104. Globa, L., Novograduska, R., Koval, A., & Senchenko, V. (2018). Examples of ontology model usage in engineering fields. *Ontology in Information Science*. DOI: 10.5772/intechopen.74369.
105. Bai, W. H., Xi, J. Q., Zhu, J. X., & Huang, S. W. (2015). Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. *Mathematical Problems in Engineering*, 2015. DOI: <https://doi.org/10.1155/2015/980945>.
106. Тукубаев З. Б., & Умаров А. А. (2010). Исследование потоков заявок, поступающих на веб - сервер организации. *Сборник трудов VII Молодежной школы семинара молодых ученых “Управление большими системами”*. сс. 369-373.
107. Litvinov, A.L. (2018), Theory of queuing systems: training manual, *О.М. Beketov NUUEKh, Kharkiv*, 141 p. [Литвинов, А.Л. (2018), Теорія систем масового обслуговування: навч. посібник, *ХНУМГ ім. О. М. Бекетова, Харків*, 141 с.].
108. Google Transparency Report, (2021), [Електронний ресурс] Google. Available at: <https://transparencyreport.google.com/traffic/overview>.
109. Nesrine Kaaniche and Maryline Laurent. A blockchain-based data usage auditing architecture with enhanced privacy and availability. In *Network Computing and Applications (NCA)*, 2017 IEEE 16th International Symposium on, pages 1–5. IEEE, 2017.
110. Padua, D. (Ed.). (2011). Encyclopedia of parallel computing. *Springer Science & Business Media*. 978 p.

Додаток 1. Акт впровадження результатів наукового дослідження**TECH 5**

Tech-5 UG • Nördliche Münchner Str 9c • 82031 Grünwald
AG München, HRB 233080 • USt.-ID: DE311827084

München,
24.05.2025

АКТ**On the implementation of software tools for optimizing the
management of computing resources in a multi-cloud environment**

We, the undersigned, representatives of Tech-5 UG – Executive Assistant Daniela Heiberger, and representatives of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” – PhD student Anton Dmytrovych Kartashov and his academic supervisor, Dr.Sc., Prof. Larysa Serhiivna Globa, certify that in 2025, as part of improving cloud infrastructure in client projects of Tech-5 UG, an approach developed by A.D. Kartashov was applied. This approach constitutes part of his dissertation entitled: “Approach to organizing distributed storage and access to cloud data”.

In the course of several projects, key elements of this approach were piloted and partially implemented, allowing the evaluation of its practical value for tasks related to the efficient management of resources in multi-cloud systems. The results confirmed the feasibility of applying the proposed solutions in real-world conditions.

Tel.: 089 2023 6665
E-Mail: info@tech-5.de
Web: www.tech-5.de

AG München
HRB 233080
USt.-ID: DE311827084

As a result of the implementation, an average 9% reduction in cloud infrastructure costs was achieved, along with a 25% reduction in data access latency compared to previously used approaches.

On behalf of Tech-5 UG

(signature) _____

Daniela Heiberger
Executive Assistant



On behalf of NTUU "KPI"

(signature) _____

A.D. Kartashov
PhD Student

(signature) _____

Larysa S. Globa
Academic Supervisor, Dr. Sc., Professor

Date: 24 May 2025