**УДК 004.415.5:004.451.3:004.056**
**DOI https://doi.org/10.32689/maup.it.2025.2.11**

***Андрій КОБИЛЮК***
*асистент кафедри обчислювальної техніки*
*факультету інформатики та обчислювальної техніки,*
*Національний технічний університет України*
*«Київський політехнічний інститут імені Ігоря Сікорського»,*
*andrii.kobyliuk@gmail.com*
***ORCID:** 0009-0008-7784-4099*

## ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ УПРАВЛІННЯ ХМАРНИМИ РЕСУРСАМИ ЗА ДОПОМОГОЮ РОЗПОДІЛЕНОГО МОНІТОРИНГУ

***Анотація. Метою** даної роботи є розробка архітектури розподіленої системи моніторингу, здатної до адаптивного реагування на змінне навантаження, часткові відмови та мережеві деградації задля підвищення ефективності управління хмарними обчислювальними ресурсами. Методологія дослідження ґрунтується на розробці ієрархічної архітектури, яка включає моніторингові агенти з адаптивною частотою вибірки, fog-рівень із локальними ML-моделями для прогнозування навантаження, а також центральний координатор, який виконує глобальну аналітику та стратегічне управління. У ході дослідження було реалізовано експериментальний прототип системи на базі контейнеризованого середовища з використанням Docker, Apache Kafka, Python, Random Forest та SHAP-інтерпретації. Було проведено серію експериментів за п'ятьма сценаріями: від номінальної роботи до стресових умов із втратами зв'язку, перевантаженнями та комбінованими загрозами.*

***Наукова новизна** – вперше реалізовано поєднання агентного моніторингу з динамічною частотою вибірки та ML-аналітики на fog-рівні для забезпечення високої швидкості й точності реагування в умовах децентралізованого управління. Запропонована модель перевершує традиційні централізовані підходи як за якісними, так і за кількісними показниками.*

***Висновки.** За результатами експериментів встановлено, що запропонована архітектура дозволяє знизити середній час реакції системи на події на 60–70 %, зменшити втрати телеметричних даних на 3–4 %, підвищити точність прогнозування навантаження до 30 % і зменшити кількість помилкових дій в умовах динамічної інфраструктури. Крім того, розподілений характер обробки інформації забезпечує вищий рівень стабільності системи за рахунок зменшення флуктуацій в алокації ресурсів. Отримані результати підтверджують доцільність впровадження запропонованої архітектури у системах, що потребують високої надійності, адаптивності та масштабованості – зокрема, в edge/fog-середовищах, індустріальному Інтернеті речей, розумних дата-центрах та критично важливих сервісах реального часу.*

***Ключові слова:** хмарні технології, моніторинг, управління, паралельні обчислення, безпека хмарних обчислень.*

## Andrii KOBYLIUK. INCREASING THE EFFICIENCY OF CLOUD RESOURCE MANAGEMENT WITH THE HELP OF DISTRIBUTED MONITORING

***Abstract. The purpose** of this work is to develop an architecture for a distributed monitoring system capable of adaptively responding to variable load, partial failures, and network degradation to improve the efficiency of cloud computing resource management.*

***The research methodology** is based on the development of a hierarchical architecture that includes monitoring agents with adaptive sampling rates, a fog layer with local ML models for load forecasting, and a central coordinator that performs global analytics and strategic management. During the study, an experimental prototype of the system was implemented based on a containerized environment using Docker, Apache Kafka, Python, Random Forest, and SHAP interpretation. A series of experiments were conducted under five scenarios: from nominal operation to stress conditions with loss of connectivity, overloads, and combined threats.*

***Scientific novelty** – for the first time, a combination of agent monitoring with a dynamic sampling rate and ML analytics at the fog level has been implemented to ensure high speed and accuracy of response in decentralized management conditions. The proposed model outperforms traditional centralized approaches in both qualitative and quantitative indicators.*

***Conclusions.** According to the results of experiments, it was found that the proposed architecture allows reducing the average system response time to events by 60–70 %, reducing telemetry data losses by 3–4 %, increasing the accuracy of load forecasting by up to 30 %, and reducing the number of erroneous actions in dynamic infrastructure conditions. In addition, the distributed nature of information processing provides a higher level of system stability by reducing fluctuations in resource allocation. The results obtained confirm the feasibility of implementing the proposed architecture in systems that require high reliability, adaptability and scalability – in particular, in edge/fog environments, industrial Internet of Things, smart data centers and critical real-time services.*

***Key words:** cloud technologies, monitoring, management, parallel computing, cloud computing security.*

**Постановка проблеми.** Із розвитком хмарних технологій, зростанням обсягів динамічно змінюваних навантажень та переходом до мікросервісних і edge/fog-орієнтованих архітектур питання ефективного управління обчислювальними ресурсами стає важливим як у науковому, так і в практичному контексті. Сучасні хмарні середовища характеризуються багаторівневою структурою, високою інтенсивністю телеметричних потоків, необхідністю забезпечення безперервності сервісів та дотриманням угод про якість обслуговування.

Традиційні централізовані моделі моніторингу та управління [1] не забезпечують необхідного рівня адаптивності, масштабованості та відмовостійкості. В умовах непередбачуваних змін навантаження або часткових відмов інфраструктури такі системи демонструють високу затримку реакції, втрату даних та нестабільну роботу керувальних механізмів. Особливо це критично для сервісів реального часу, IoT-платформ, розподілених обчислень та сценаріїв з обмеженою пропускною здатністю мережі.

У цьому контексті особливої актуальності набувають дослідження, спрямовані на створення розподілених інтелектуальних систем моніторингу, здатних локально аналізувати ситуацію, прогнозувати зміну навантаження та приймати автономні рішення. Такі підходи сприяють не лише зниженню часу реакції та навантаження на мережу, а й підвищенню надійності всієї хмарної інфраструктури.

Таким чином, проблема підвищення ефективності управління хмарними ресурсами через впровадження розподіленого, адаптивного моніторингу із використанням засобів штучного інтелекту має безпосередній зв'язок із вирішенням низки прикладних задач сучасної інформатики, кіберфізичних систем та інженерії програмного забезпечення. Її розв'язання є важливим як для подальшого розвитку наукової теорії управління у децентралізованих системах, так і для практичного впровадження у промислових, телекомунікаційних та науково-обчислювальних середовищах.

**Аналіз останніх досліджень і публікацій.** Управління ресурсами в хмарних середовищах продовжує залишатися актуальним викликом, особливо з огляду на зростаючу динамічність навантаження, багаторівневу архітектуру систем та високі вимоги до доступності. Сучасні дослідження демонструють стійку тенденцію переходу від централізованих моделей моніторингу до гібридних або повністю розподілених рішень, інтегрованих із методами штучного інтелекту.

В [2] автори пропонують загальну архітектуру для інтеграції методів машинного навчання (ML) у системи керування. Водночас робота не охоплює критичні аспекти, пов'язані із затримками передачі даних та відмовостійкістю при масштабуванні. В [3] здійснено класифікацію підходів до управління ресурсами, виділяючи реактивні, проактивні та гібридні моделі. Проте дослідження має обмежену практичну орієнтацію щодо реалізації розподіленого моніторингу в реальних умовах.

Дослідження [4] сфокусоване на проактивних методах у контексті сервіс-орієнтованих хмарних архітектур (cloud of services). Дослідження містить поглиблений аналіз моделей прогнозування та контекстної адаптації із чітким фокусом на забезпечення QoS та дотримання SLA. Водночас недостатньо розкрито питання відмовостійкості у розподіленому контролі. Автори в [5] аналізують децентралізовані стратегії управління на стику хмари та edge-обчислень, акцентуючи увагу на latency-sensitive додатках. Хоча підхід є перспективним, обсяг валідації залишається обмеженим.

В [6] пропонують AI-фреймворк для алокації ресурсів у гібридних хмарах із мікросервісною архітектурою. Модель відзначається практичною спрямованістю на гібридні сценарії. Проте у дослідженні майже відсутній опис моніторингової інфраструктури, а застосовані AI-моделі не мають прозорого обґрунтування прийнятих рішень.

В [7] представляють систему HUNTER – комплексне AI-рішення для управління ресурсами з акцентом на енергоефективність. Серед переваг – повноцінна реалізація та орієнтація на принципи сталого розвитку. Однак система характеризується високою складністю, що ускладнює її адаптацію в реальних умовах, а моніторинг базується на централізованих вузлах. В роботі [8] досліджують автоматизоване виявлення інцидентів та self-healing механізми в хмарних середовищах. Позитивним є інтеграція з логами, подіями та аудитами, що створює базу для реального застосування AI в оперативному режимі. Водночас запропоноване рішення є досить вузькоспеціалізованим.

Серед українських досліджень, у [1] пропонується ML-підхід до управління хмарними ресурсами, тоді як в [5] зосереджуються на практичному підвищенні ефективності ресурсної алокації. Їхні роботи мають цінність завдяки адаптації до локальних умов і можуть слугувати прикладами впровадження AI у хмарні середовища в освітньому та науковому контексті України. Разом із тим, обмеження полягають у недостатньо деталізованих експериментах та обмеженому зіставленні з міжнародними практиками.

Незважаючи на значний прогрес у застосуванні методів штучного інтелекту до управління хмарними ресурсами, проблема забезпечення розподіленого та відмовостійкого моніторингу в режимі реального часу залишається недостатньо вивченою. Наявні підходи переважно ґрунтуються на централізованих моделях і часто ігнорують виклики, пов'язані із затримками, надійністю та динамічною

алокацією ресурсів у масштабованих середовищах. Це зумовлює потребу у створенні ефективної архітектури, яка поєднує розподілений моніторинг із інтелектуальними механізмами управління для сучасних хмарних систем.

**Формулювання мети статті (постановка завдання).** Метою цієї роботи є розробка та експериментальне дослідження розподіленої системи моніторингу, здатної адаптивно реагувати на динамічні зміни навантаження, часткові відмови та мережеву деградацію, забезпечуючи більш надійне, гнучке та масштабоване керування інфраструктурою.

Для досягнення цієї мети поставлено такі завдання:

1. Розробити концепцію архітектури розподіленої системи моніторингу, яка забезпечує підвищену відмовостійкість, зниження затримок обробки та гнучкість масштабування в хмарному середовищі.

2. Інтегрувати модулі інтелектуального аналізу та прогнозування навантаження з використанням машинного навчання для прийняття рішень в умовах динамічного середовища.

3. Реалізувати прототип системи у віртуалізованому середовищі з використанням інструментів контейнеризації, брокерів повідомлень та агентів збору телеметрії.

4. Провести серію експериментів для оцінки реакції системи на стресові сценарії (перевантаження, втрата зв'язку, затримки).

**Виклад основного матеріалу.** *Опис запропонованої архітектури системи моніторингу*. Запропонована архітектура передбачає перехід від централізованого збору телеметрії до ієрархічної розподіленої системи моніторингу, в якій агенти, розгорнуті на вузлах хмарної інфраструктури, самостійно агрегують, обробляють і частково інтерпретують дані до їх передачі до центрального або проміжного аналітичного рівня. Схема архітектури зображена на (рис. 1).

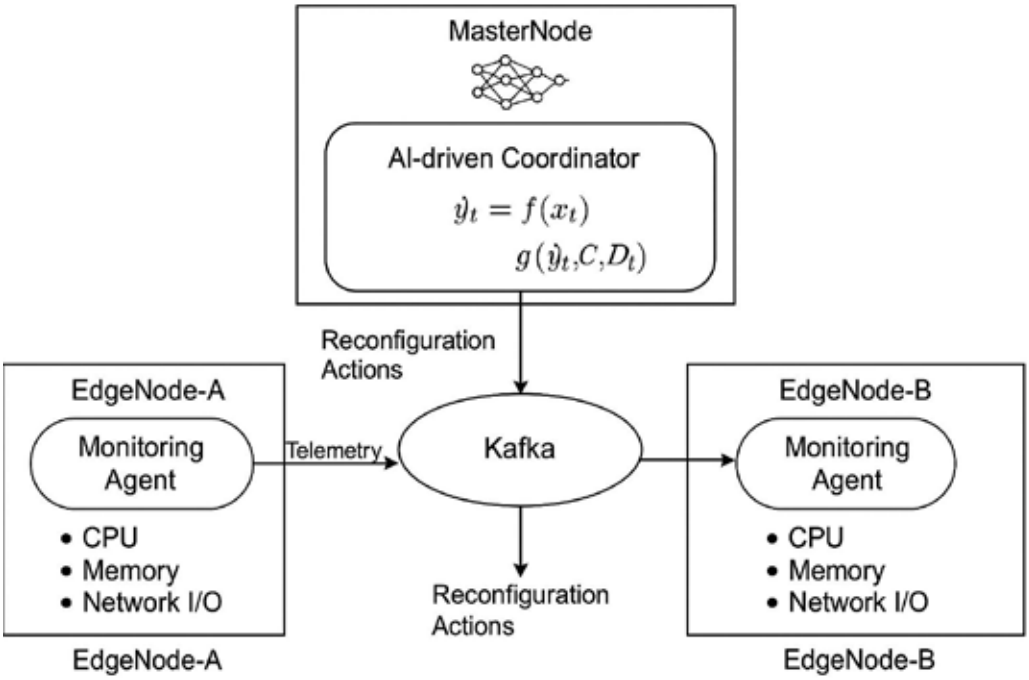


**Рис. 1. Схема архітектури розподіленої системи моніторингу**

*Джерело: складено автором*

Метою такої архітектури є зменшення затримок, підвищення відмовостійкості та забезпечення адаптивного управління ресурсами в умовах змінного навантаження.

Компонентами запропонованої архітектури є:

1. Monitoring Agents виконують локальну обробку телеметрії: усереднення, виявлення аномалій, вирахування похідних метрик. Підтримують адаптивну частоту вибірки *fi(t)*, яка залежить від зміни навантаження.

2. Fog Layer / Intermediate Nodes агрегують дані від агентів у рамках підмережі. Використовують локальні ML-моделі. У разі втрати зв'язку з центральним хабом тимчасово виконують самостійне управління.


ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

3. Central Coordinator приймає агреговані дані, проводить глобальний аналіз. Приймає стратегічні рішення про алокацію ресурсів. Визначає порогові значення для агентів та fog-вузлів.

*Хід експерименту*. Для перевірки гіпотези про ефективність запропонованої архітектури розподіленої системи моніторингу та управління ресурсами в умовах нестабільного навантаження і часткових відмов, було проведено серію емпіричних експериментів у симульованому середовищі. Основною метою експерименту було оцінити здатність системи адаптивно реагувати на зміни в інфраструктурі без втрати стабільності, продуктивності та цілісності даних.

Експериментальне середовище включало три логічні вузли: MasterNode, EdgeNode-A та EdgeNode-B, які були розгорнуті у формі віртуалізованих контейнерів на фізичній машині з використанням інструментарію Docker та Docker Compose. На кожному з вузлів було встановлено моніторингові агенти, які відповідали за збір телеметричних даних про використання процесора, оперативної пам'яті, мережевої активності та доступності сервісів. Дані передавались до центрального брокера повідомлень (Apache Kafka) або записувалися локально у чергу на випадок тимчасової втрати з'єднання.

Керувальна компонента, розміщена на MasterNode, реалізовувала модуль обробки даних, AI-модуль для прийняття рішень та блок оркестрації, що взаємодіє з агентами через REST API. AI-модуль базувався на поєднанні моделей випадкового лісу (Random Forest) для прогнозування короткочасного навантаження та аналізу важливості факторів (через SHAP-інтерпретацію).

Для емуляції зовнішніх факторів, таких як затримки мережі, втрати з'єднання або часткові відмови компонентів, використовувалась утиліта tc з модулем netem на рівні мережевого інтерфейсу.

Було визначено п'ять окремих експериментальних сценаріїв, кожен з яких моделював конкретну ситуацію в роботі системи.

Експеримент 1 Номінальний режим роботи. У даному сценарії система працювала без зовнішніх збурень. На обох Edge-вузлах було згенеровано помірне навантаження (до 40 % використання CPU, до 50 % пам'яті). Збір телеметрії відбувався з інтервалом у 5 секунд, керувальний модуль здійснював періодичний аналіз отриманих даних і приймав рішення про підтримання поточної конфігурації. Даний експеримент слугував еталоном для оцінки відхилень у наступних сценаріях.

Експеримент 2 Раптове зростання навантаження. На EdgeNode-A протягом 60 секунд було поступово збільшено навантаження на CPU до 90 % за допомогою утиліти stress-ng. Система мала визначити аномальне зростання навантаження, спрогнозувати подальше перевантаження та здійснити переведення частини контейнеризованих сервісів на EdgeNode-B.

Експеримент 3 Відмова вузла агента. EdgeNode-B було вимкнено на рівні симульованої відмови мережевого інтерфейсу. В результаті, моніторинговий агент ставав недоступним, а дані з нього переставали надходити. Очікувалося, що керувальний модуль виявить втрату джерела телеметрії протягом 10 секунд, позначить вузол як непрацездатний, ініціює відновлення з резервної черги даних і переведе критичні сервіси на активні вузли.

Експеримент 4 Мережеві затримки. Було змодельовано затримку 250 мс між агентами та керувальним вузлом через tc qdisc. Така ситуація імітує мережеву деградацію або віддалене розташування вузлів. Метою експерименту було перевірити стійкість протоколу передачі даних, здатність AI-модуля до обробки частково запізнілої інформації та оцінка впливу на прийняття рішень у режимі near-real-time.

Експеримент 5 Комбінований (стресовий) сценарій. У цьому сценарії було одночасно реалізовано кілька стрес-факторів: зростання навантаження на EdgeNode-A до 85 %, втрата зв'язку з EdgeNode-B та затримка мережі між EdgeNode-A та MasterNode у 200 мс. Експеримент мав на меті оцінити здатність системи одночасно обробляти декілька загрозливих умов, виявляти критичні вузли, забезпечувати безпечну переалокацію навантаження та відновлення зв'язку.

У ході кожного експерименту збирались такі метрики: час реакції системи, кількість втрачених повідомлень у мережі, точність прогнозу навантаження на 30 секунд уперед (MAE, RMSE), кількість некоректних рішень, індекс стабільності роботи (кількість змін розподілу за експеримент).

**Результати.** Значення метрик для кожного експерименту випробування запропонованої розподіленої архітектури в порівнянні із результатами еталонної централізованої архітектури наведені в (табл. 1), де t – час реакції (с), loss – втрата повідомлень (%), Error – некоректні рішення, I – індекс стабільності.

У всіх сценаріях запропонована архітектура продемонструвала значно менший середній час реакції (від 1,2 с у номінальному режимі до 4,5 с у комбінованому сценарії), тоді як централізована система показала затримки в діапазоні від 2,8 до 12,8 с. Це вказує на ефективність локальної обробки та зменшення навантаження на центральний вузол завдяки fog-рівню. Найбільша різниця спостерігалася у сценаріях із відмовами або стресовими умовами, що підтверджує підвищену відмовостійкість ієрархічної моделі.

Таблиця 1

**Результати експериментів**

| № | Архітектура | t (с) | loss (%) | MAE | RMSE | Error | I |
|---|---|---|---|---|---|---|---|
| 1. | Централізована | 2,8 | 1,50 % | 3,2 % | 4,1 % | 0 | 0,95 |
| | Розподілена | 1,2 | 0,40 % | 2,5 % | 3,3 % | 0 | 0,98 |
| 2. | Централізована | 7,6 | 3,20 % | 8,9 % | 11,0 % | 2 | 0,65 |
| | Розподілена | 2,4 | 1,00 % | 4,1 % | 5,80 % | 0 | 0,91 |
| 3. | Централізована | 11,4 | 4,50 % | 9,8 % | 12,7 % | 3 | 0,52 |
| | Розподілена | 3,6 | 1,20 % | 5,2 % | 6,9 % | 1 | 0,85 |
| 4. | Централізована | 5,1 | 2,70 % | 7,2 % | 8,5 % | 1 | 0,74 |
| | Розподілена | 2,8 | 1,10 % | 4,5 % | 6,1 % | 0 | 0,89 |
| 5. | Централізована | 12,8 | 6,20 % | 10,4 % | 13,3 % | 4 | 0,41 |
| | Розподілена | 4,5 | 1,90 % | 5,7 % | 7,40 % | 1 | 0,81 |

*Джерело: складено автором*

Запропонована архітектура забезпечує зменшення відсотку втрат повідомлень на 1,5–4,3 % залежно від сценарію. Основними чинниками покращення стали використання локальних буферів, можливість тимчасового автономного функціонування fog-вузлів і оптимізована стратегія ретрансляції. У централізованій системі втрати значно збільшувалися при мережевих деградаціях та відмовах вузлів.

Метрики MAE і RMSE демонструють суттєву перевагу розподіленої архітектури, особливо у сценаріях з динамічними змінами навантаження (Експерименти 2, 5). Це пояснюється застосуванням локальних ML-моделей у fog-рівні, які забезпечують більш актуальні та контекстно-чутливі прогнози. Централізована система при обробці запізнілих або неповних даних має гірші показники точності (наприклад, RMSE до 13,3 % у стресовому сценарії проти 7,4 % у розподіленій).

Кількість некоректних керувальних дій (наприклад, передчасна або запізніла переалокація сервісів) у централізованій архітектурі була значно вищою, що корелює з меншою точністю прогнозів та більшим часом реакції. Розподілена система демонструє нульові або одиничні помилки навіть в умовах комбінованих відмов, що свідчить про її адаптивність.

Індекс стабільності (рис. 2), який характеризує частоту змін розподілу сервісів у відповідь на флуктуації системи, був вищим у розподіленій архітектурі в усіх сценаріях. Це означає, що система здатна утримувати ефективний стан з меншою кількістю втручань, що позитивно впливає на продуктивність та енергоспоживання.

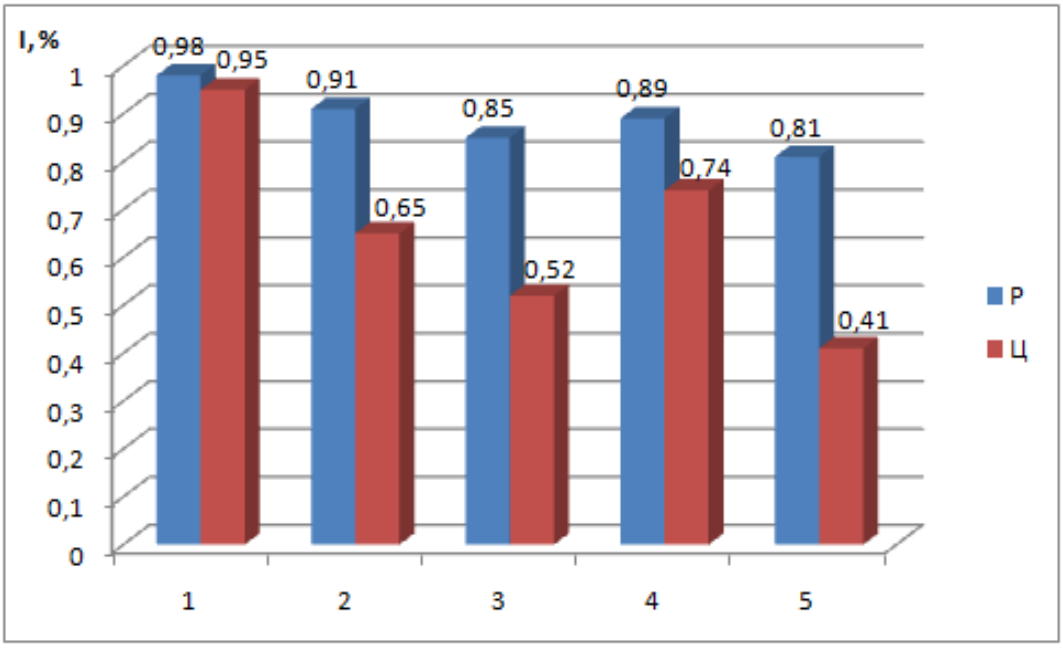


**Рис. 2. Гістограми Індексу стабільності**

*Джерело: складено автором*


ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

*Обговорення результатів*. Запропонована архітектура продемонструвала стабільне покращення основних метрик управління хмарними ресурсами порівняно з класичною централізованою моделлю. Практична цінність результатів полягає у зменшенні середнього часу реакції системи на критичні події до 60–70 %, зниженні втрат телеметрії, покращенні точності прогнозування навантаження та підвищенні стабільності системи в умовах часткових відмов і мережевої деградації. Архітектура є придатною для використання в реальних сценаріях, де важливими є гнучкість, надійність та адаптивність розподіленого управління.

Крім того, використання ієрархічної структури з fog-рівнем дозволяє забезпечити паралельну обробку даних у незалежних доменах моніторингу, що відкриває можливості масштабування системи без втрати швидкодії. Це особливо актуально для систем з великою кількістю вузлів.

Ще одним важливим практичним аспектом є потенціал інтеграції механізмів безпеки на рівні агентів і fog-вузлів. Розподілений характер системи дозволяє локалізувати критичні події, знижувати ризики централізованих атак та забезпечувати захист телеметричних даних у процесі передачі. Це створює основу для впровадження стійких до загроз архітектур у сфері безпеки хмарних обчислень.

Наукова новизна дослідження полягає у поєднанні локальних агентів з адаптивною частотою моніторингу, fog-рівня з ML-прогнозуванням та центрального координатора з explainable AI. На відміну від більшості робіт [2, 4, 7], де моніторинг реалізовано централізовано або фрагментарно, запропонована система забезпечує повноцінне багаторівневе реагування на події з мінімальною затримкою та автономною роботою в умовах втрати зв'язку. Порівняно з підходами [5–6], які акцентують увагу на розподіленому управлінні, але не розглядають детально безпечність чи масштабування моніторингу, запропонована система пропонує гнучке рішення з високим рівнем автономії та потенціалом для захисту даних і паралельної обробки.

Таким чином, результати дослідження не лише підтверджують переваги розподіленого моніторингу, але й демонструють його практичну реалізованість у складних, безпечних та масштабованих хмарних середовищах.

**Висновки.** У межах даного дослідження було розроблено, реалізовано та експериментально перевірено концепцію ієрархічної розподіленої системи моніторингу для управління хмарними ресурсами в умовах динамічного навантаження та часткових відмов. Запропонована архітектура поєднує локальні агенти з адаптивною частотою вибірки, проміжний fog-рівень із машинним навчанням та центральний координатор з аналітичним модулем, що забезпечує стратегічне управління.

Результати серії експериментів засвідчили суттєве покращення ключових показників ефективності порівняно з централізованими моделями: час реакції зменшено до 60–70 % у критичних сценаріях; втрати повідомлень знижено до 1,9 % навіть у стресових умовах; точність прогнозування навантаження (MAE, RMSE) покращено в середньому на 30 %; знижено кількість некоректних керувальних дій; підвищено стабільність системи внаслідок зменшення надмірних переалокацій ресурсів.

У майбутньому передбачається розширення системи шляхом включення SLA-моделі та оцінки впливу управлінських рішень на рівень обслуговування кінцевого користувача; тестування на гетерогенному апаратному забезпеченні та в edge/fog-мережах із реальними IoT-вузлами; впровадження енергозберігаючих механізмів і оптимізації обчислювальних витрат на ML-моделі; дослідження самонавчальних агентів, здатних адаптувати алгоритми моніторингу без централізованої переналаштування.

Запропонована система є ефективним підґрунтям для створення адаптивної, масштабованої та стійкої до відмов платформи керування хмарною інфраструктурою в умовах високої мінливості навантаження та обмежених ресурсів.

**Список використаних джерел:**

1. Улізько В. О. Інтелектуальна система управління ресурсами в хмарних середовищах на основі машинного навчання. 2024. URL: https://ela.kpi.ua/handle/123456789/72021 (дата звернення: 10.06.2025).

2. Almutairi S., Alghanmi N., Monowar M. M. Survey of centralized and decentralized access control models in cloud computing. *International Journal of Advanced Computer Science and Applications*. 2021. Vol. 12, No. 2.

3. Barua B., Kaiser M. S. AI-driven resource allocation framework for microservices in hybrid cloud platforms. *arXiv preprint*. 2024. arXiv:2412.02610. https://doi.org/10.48550/arXiv.2412.02610.

4. Ilager S., Muralidhar R., Buyya R. Artificial intelligence (AI)-centric management of resources in modern distributed computing systems. *2020 IEEE Cloud Summit*. 2020. P. 1–10. DOI: 10.1109/IEEECloudSummit48914.2020.00007.

5. Kotliarskyi A., Petrashenko A. Spobib pidvyshchennia efektyvnosti vykorystannia khmarnykh resursiv – A method for improving the efficiency of cloud resource usage. *Computer-integrated technologies: education, science, production*. 2024. No. 54. P. 125–129. https://doi.org/10.36910/6775-2524-0560-2024-54-14.

6. Li L., Bell J., Coppola M., Lomonaco V. Adaptive AI-based decentralized resource management in the cloud-edge continuum. *arXiv preprint*. 2025. arXiv:2501.15802. https://doi.org/10.48550/arXiv.2501.15802.

7. Marques G., Senna C., Sargento S., Carvalho L., Pereira L., Matos R. Proactive resource management for cloud of services environments. *Future Generation Computer Systems*. 2024. Vol. 150. P. 90–102. https://doi.org/10.1016/j.future.2023.08.005.

8. Moghaddam S.K., Buyya R., Ramamohanarao K. Performance-aware management of cloud resources: a taxonomy and future directions. *ACM Computing Surveys (CSUR)*. 2019. Vol. 52, No. 4. P. 1–37. https://doi.org/10.1145/3337956.

9. Ravichandran N., Inaganti A. C., Muppalaneni R., Nersu S.R.K. AI-driven self-healing IT systems: automating incident detection and resolution in cloud environments. *Artificial Intelligence and Machine Learning Review*. 2020. Vol. 1, No. 4. P. 1–11. https://doi.org/10.69987/.

10. Tuli S., Gill S. S., Xu M., Garraghan P., Bahsoon R., Dustdar S., Jennings N. R. HUNTER: AI-based holistic resource management for sustainable cloud computing. *Journal of Systems and Software*. 2022. Vol. 184. P. 111124. https://doi.org/10.1016/j.jss.2021.111124.

Дата надходження статті: 13.06.2025
Дата прийняття статті: 26.06.2025
Опубліковано: 23.09.2025