УДК 004.722
DOI https://doi.org/10.32782/2521-6643-2025-2-70.26

Киричек Г. Г., кандидат технічних наук, доцент,
доцент кафедри комп'ютерних систем та мереж
Національного університету «Запорізька політехніка»
ORCID: 0000-0002-0405-7122

Целуйко Р. О., студент факультету комп'ютерних наук
та технологій
Національного університету «Запорізька політехніка»
ORCID: 0009-0000-3697-3283

Тягунова М. Ю., кандидат технічних наук, доцент,
доцент кафедри комп'ютерних систем та мереж
Національного університету «Запорізька політехніка»
ORCID: 0000-0002-9166-5897

Живогляд В. А., студент факультету комп'ютерних наук
та технологій
Національного університету «Запорізька політехніка»
ORCID: 0009-0005-6395-2214

ХМАРНІ СЕРВІСИ В ІНФРАСТРУКТУРІ РОЗПОДІЛЕНИХ МЕРЕЖ

В роботі розглянуто підходи до впровадження хмарних сервісів у інфраструктуру розподілених мереж. Надано огляд сучасних платформ (AWS, Microsoft Azure, Google Cloud Platform) та систем моніторингу (Prometheus, Grafana), описано методи автоматизації та алгоритми масштабування ресурсів, які базуються на подіях і метриках. Основна увага приділена реалізації та верифікації моделей, які забезпечують динамічне масштабування, відмовостійкість та оптимізацію витрат. Метою роботи є аналіз провідних хмарних платформ, реалізація та верифікація моделі управління інфраструктурою розподілених мереж із використанням хмарних сервісів, яка забезпечує: централізований моніторинг з використанням Prometheus та Grafana; адаптивне масштабування ресурсів на основі метрик та безпеку передачі даних між локальними вузлами і хмарними сервісами. Об'єктом дослідження є процес інтеграції хмарних технологій в інфраструктуру розподілених мереж. Предметом є програмні, інструментальні засоби, методи, моделі моніторингу та керування розподіленими мережами. Авторами розроблена модель управління інфраструктурою з використанням інтеграції з публічною хмарою. Проведено моделювання декількох сценаріїв навантаження та виконано порівняння приватної, публічної та гібридної моделей інтеграції за показниками продуктивності, відмовостійкості та витрат. Наведено алгоритми автоматичного масштабування на основі Prometheus Alertmanager та AWS Auto Scaling, а також запропоновано набір практичних рекомендацій щодо безпечної інтеграції хмарних сервісів у розподілені мережі. Виконано моделювання системи управління з інтеграцією інструментів Prometheus та Grafana. Результати підтверджують, що інтеграція хмарних рішень і методів програмно-визначених мереж дозволяє досягти значного підвищення продуктивності та стійкості мережевої інфраструктури, що є критично важливим для сучасних масштабованих рішень. Отримані результати можна використовувати при модернізації корпоративної мережевої інфраструктури з метою підвищення її надійності та ефективності.

Ключові слова: *хмарні сервіси, розподілені мережі, автоматизація, масштабування, Prometheus, Grafana, AWS Auto Scaling.*

Kyrychek H. H., Tseluiko R. O., Tiahunova M. Yu., Zhyvohliad V. A. Cloud services in distributed network infrastructure

The article considers approaches to implementing cloud services into distributed network infrastructure. An overview of modern platforms (AWS, Microsoft Azure, Google Cloud Platform) and monitoring systems (Prometheus, Grafana) is provided, automation methods and resource scaling algorithms based on events and metrics are described. The main focus is on the implementation and verification of models that ensure dynamic scaling, fault tolerance, and cost optimization. The purpose of the work is to analyze leading cloud platforms, implement and verify a model for managing distributed network infrastructure using cloud services, which provides: centralized monitoring using Prometheus and Grafana; adaptive scaling of resources based on metrics and security of data transmission between local nodes and cloud services. The object of research is the process of

integrating cloud technologies into the infrastructure of distributed networks. The subject is software, tools, methods, models for monitoring and managing distributed networks. The authors developed an infrastructure management model using integration with a public cloud. Several load scenarios were simulated and private, public, and hybrid integration models were compared in terms of performance, fault tolerance, and costs. Algorithms for automatic scaling based on Prometheus Alertmanager and AWS Auto Scaling are presented, and a set of practical recommendations for the secure integration of cloud services into distributed networks is proposed. The management system was modeled with the integration of Prometheus and Grafana tools. The results confirm that the integration of cloud solutions and Software-Defined Networking (SDN) methods allows for a significant increase in the performance and resilience of the network infrastructure, which is critical for modern scalable solutions. The results obtained can be used when modernizing corporate network infrastructure to increase its reliability and efficiency. The results showed that as the number of users increases, latency increases, but the system maintains functionality even under peak conditions.
Key words: *cloud services, distributed networks, automation, scaling, Prometheus, Grafana, AWS Auto Scaling.*

Постановка проблеми. Еволюція корпоративних IT-інфраструктур супроводжується переходом від монолітних до географічно розподілених мереж та мікросервісної архітектури [1]. Це створює низку викликів, таких як: необхідність централізованого управління; забезпечення високої доступності та ефективне використання ресурсів [2]. Традиційні методи мережевого адміністрування не справляються із великими, за обсягом, навантаженнями, тому потребують впровадження методів, які спираються на застосування хмарних сервісів та програмної автоматизації [3]. Хмарні технології пропонують ефективні рішення, які забезпечують масштабованість, гнучкість та зниження витрат на підтримку фізичної інфраструктури розподілених мереж, які стають невід'ємною частиною сучасної корпоративної інфраструктури [4]. Ключовим архітектурним підходом, який дозволяє централізувати управління розподіленою мережею є програмно-конфігуровані мережі (SDN). Вони передбачають відокремлення площини управління (Control Plane) від площини даних (Data Plane). Такий централізований підхід дозволяє реалізувати гнучкі політики та автоматизувати конфігурацію, що є неможливим у традиційних мережах передачі даних [5].

Аналіз останніх досліджень та публікацій. Програмно-визначені мережі (SDN), інтеграція та безпека хмарних обчислень та інструменти моніторингу Prometheus і Grafana, особливо комбінація цих трьох напрямків є критично важливою для вирішення проблем продуктивності та надійності сучасних географічно розподілених мереж. Однією з головних є концепція програмно-визначених мереж (SDN), яка є відповіддю на жорсткість і складність традиційних мережевих архітектур, де площина управління і площина даних тісно інтегровані. Автори робіт [6] та [7] підтверджують, що SDN є інноваційною архітектурою, яка, відокремлюючи управління від передачі даних, значно спрощує керування, програмованість та адаптацію мережі до вимог, що постійно змінюються. Вони також аналізують зростання кібератак та вразливостей, пов'язаних із динамічним і масштабованим характером хмарних сервісів. Централізований контролер SDN дозволяє адміністраторам динамічно застосовувати політики, що є критичним для автоматизованого середовища хмарних обчислень [8]. Це безпосередньо виправдовує використання SDN-підходів, оскільки вони є необхідною умовою для реалізації алгоритмів автоматичного масштабування. Водночас, аналіз робіт провідних фахівців, вказує на суттєві виклики. Зокрема, [9] наголошує на проблемах безпеки та відмовостійкості SDN. В умовах логічної централізації, контролер SDN стає єдиною точкою відмови і це вимагає впровадження надійних механізмів, що повністю узгоджується із поставленими завданнями забезпечення відмовостійкості системи управління. Окрім того, актуальність дослідження значною мірою спирається на масовий перехід організацій та підприємств до застосування хмарних платформ (AWS, Azure, GCP). На рисунку 1 наведена архітектура хмарної інфраструктури розподілених мереж, яка включає: локальні вузли; сервери моніторингу; VPN-з'єднання та хмарні сервіси. Автори робіт [2] та [10] наводять ключові завдання, пов'язані з цим переходом: масштабованість, гнучкість, швидке розгортання та значне зниження експлуатаційних витрат. Хоча гнучкість і є основою для реалізації алгоритмів автоматичного масштабування, проте автори виділяють безпеку як головну перешкоду на шляху широкого впровадження хмар [11, 12].

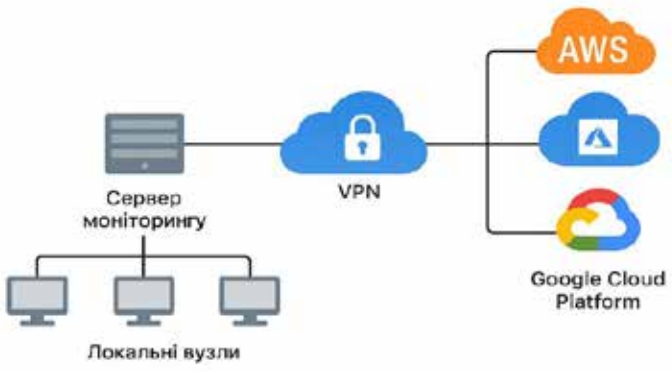

Рис. 1. Архітектура хмарної інфраструктури розподілених мереж

Основні загрози виникають через: спільне використання фізичних ресурсів різними клієнтами; розподіленого зберігання даних на різних географічних локаціях, що ускладнює контроль [13] та швидкого розгортання нових підключень, які можуть створювати нові вразливості [14]. Аналіз джерел [12, 15] свідчить про необхідність впровадження комплексних заходів, які охоплюють конфіденційність, цілісність і доступність, тому потрібно зосередитись на порівнянні приватної, публічної та гібридної хмарних моделей, оскільки гібридна модель визнається оптимальним компромісом між масштабованістю публічної хмари та високим рівнем контролю (безпеки) приватної.

Для ефективного управління динамічними розподіленими мережами, які постійно масштабуються, критично необхідним є використання систем мніторінгу. Наукові дослідження, присвячені моніторингу сучасних інфраструктур, одностайно підтверджують доцільність використання систем Prometheus та Grafana [16, 17]. При цьому Prometheus використовується як база даних (Time-Series Database) з унікальною багатовимірною моделлю даних, що дозволяє точно ідентифікувати: метрику, вузол, сервіс або регіон [18], що є необхідною умовою для роботи алгоритму масштабування на основі метрик. Grafana [19, 20] забезпечує візуалізацію цих даних та її здатність інтегрувати дані з різних джерел (Prometheus, Zabbix, ELK) робить її ідеальною платформою для єдиної точки спостереження за гібридною мережею [21]. Цей аналіз доводить, що вибір Prometheus та Grafana не є випадковим, а відповідає сучасним галузевим стандартам моніторингу інфраструктур, де ключові метрики (завантаження CPU, мережева затримка) використовуються як тригери для автоматизованих систем управління.

Аналіз наукових джерел підтверджує, що управління інфраструктурою розподілених мереж є однією з найбільш актуальних і достатньо складних проблем у сучасних інформаційних технологіях. Обрана комбінація: SDN-принципів та підходів; гнучкості хмарних платформ та інструментів моніторінгу (Prometheus/ Grafana) є науково обґрунтованою і відповідає завданню створення масштабованих, відмовостійких та економічно ефективних рішень. Це забезпечує надійну теоретичну базу для верифікації розробленої моделі та отримання практично значущих результатів.

Мета статті. Метою є проведення аналізу провідних хмарних платформ, реалізація та верифікація моделі управління інфраструктурою розподілених мереж із використанням хмарних сервісів, яка забезпечує: централізований моніторинг з використанням Prometheus та Grafana; адаптивне масштабування ресурсів на основі метрик та безпеку передачі даних між локальними вузлами і хмарними сервісами. Об'єктом дослідження є процес інтеграції хмарних технологій в інфраструктуру розподілених мереж. Предметом є програмні, інструментальні засоби, методи, моделі моніторингу та керування розподіленими мережами. В роботі потрібно змоделювати навантаження та виконати порівняння приватної, публічної та гібридної моделей інтеграції за показниками продуктивності, відмовостійкості та витрат. Після проведення аналізу навести алгоритми автоматичного масштабування на основі Prometheus Alertmanager та AWS Auto Scaling, а також запропонувати набір практичних рекомендацій щодо безпечної інтеграції хмарних сервісів у розподілені мережі. Виконати моделювання системи управління з інтеграцією інструментів Prometheus та Grafana. Підтвердити, що інтеграція хмарних рішень і методів програмно-визначених мереж дозволяє досягти значного підвищення продуктивності та стійкості мережевої інфраструктури, що є критично важливим для сучасних масштабованих рішень. Основну увагу приділити реалізації та верифікації моделей, які забезпечують динамічне масштабування, відмовостійкість та оптимізацію витрат.

Виклад основного матеріалу. Системи Prometheus та Grafana є ключовими елементами моніторінгу, але в умовах застосування масштабованих та мікросервісних розподілених мереж використовуються і інші інструменти, які пропонують більш комплексні рішення для логування, трасування та моніторингу продуктивності додатків (APM) (табл. 1).

Тоді, як Prometheus/Grafana гарно справляються з метриками, для комплексного моніторингу (який включає логування та трасування), сучасні розподілені мережі часто вимагають інтеграції з іншими інструментами, такими як ELK Stack або комерційними рішеннями (Datadog, New Relic), які забезпечують повний цикл аналізу даних.

Наведемо методи збору метрик, алгоритми прийняття рішень для масштабування ресурсів та підходи до інтеграції локальної інфраструктури з хмарою. Основними компонентами моделі є: агент збору метрик на вузлах мережі; центральний сервер моніторингу (Prometheus) з налаштуваннями для збору метрик; система візуалізації (Grafana) для дашбордів та кореляції подій; Alertmanager для формування сповіщень за правилами; компонент автоматизації масштабування (AWS Auto Scaling) та модуль безпеки. Алгоритм, який використовується, є процесом автоматичного масштабування у відповідь на події. Наведемо код алгоритму. Маємо вхідні дані: метрики $M(t)$ з Prometheus, порогові значення T_up, T_down, мін./макс. інстанси N_min, N_max. Потрібно:

– зчитати останні значення метрик за інтервал t_window;

– обчислити агреговане значення $metric\_agg = f\left(M(t)\right)$;

– якщо $metric\_agg > \underline{T}\ up$ і поточні інстанси $< N\_max$ → ініціювати масштабування вгору: $add\_instances(k)$;

– якщо $metric\_agg < \underline{T}\ down$ і поточні інстанси $> N\_min \rightarrow$ ініціювати масштабування вниз: $remove\_instances(k)$;
– встановити *cooldown timer*;
– логувати дії та відправляти сповіщення.

Таблиця 1

Інструменти моніторингу в управлінні розподіленими мережами

Інструмент	Тип	Функція	Переваги для розподілених мереж	Недоліки / особливості
Prometheus	Open Source (Метрики)	Збір (Scraping), збереігання та агрегація метрик.	Багатовимірна модель даних (мітка, значення) дозволяє точно ідентифікувати вузол, регіон та сервіс.	Не підходить для логування/трасування; обмежена історична глибина даних.
Grafana	Open Source (Візуалізація)	Візуалізація даних, дашборди, оповіщення.	Створення уніфікованих дашбордів для різних джерел даних (Prometheus, Elastic).	Є лише візуалізатором, вимагає окремих бекендів для зберігання даних.
ELK Stack / Elastic Observability	Open Source / Комерційний	Логування, пошук, аналіз метрик і трасування (Elasticsearch, Logstash, Kibana).	Ідеально підходить для централізованого логування (усунення несправностей) та повнотекстового пошуку.	Високі вимоги до обчислювальних ресурсів для великих об'ємів даних.
Datadog	Комерційний (APM, All-in-one)	Комплексна платформа для APM (Application Performance Monitoring), моніторингу інфраструктури, логування, Real User Monitoring (RUM).	Уніфікований інтерфейс, легкість інтеграції, AI-driven оповіщення.	Висока вартість при великих обсягах даних; пропрієтарний формат.
Zabbix	Open Source (Метрики, Моніторинг)	Мережевий та серверний моніторинг, збір даних агентами.	Поглиблений моніторинг мережевого обладнання (SNMP), велика кількість готових шаблонів.	Менш гнучкий для динамічних хмарних середовищ (порівняно з Prometheus).
New Relic	Комерційний (APM, All-in-one)	Хмарна платформа для APM, трасування, моніторингу інфраструктури та логування.	Ефективне розподілене трасування (Distributed Tracing), що критично для мікросервісів у розподіленій мережі.	Фокус переважно на продуктивності додатків; висока ціна.

Модель системи управління мережею наведено на рисунку 2. На схемі розташовано всі загальні компоненти мережевої інфраструктури, а також їх взаємодію та зв'язки між ними.

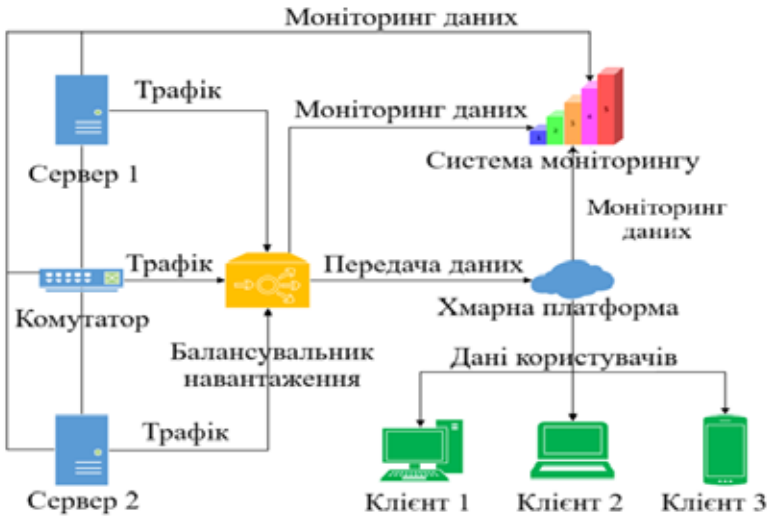

Рис. 2. Модель системи

Модель системи містить локальні вузли, маршрутизатори та центральний сервер моніторингу. Локальні вузли експортують метрики через node_exporter до Prometheus; Prometheus опитує вузли за фіксованим графіком. Для інтеграції з публічною хмарою використано AWS CloudWatch і Auto Scaling: частина логіки моніторингу дублюється у хмарі за допомогою Amazon Managed Service for Prometheus. Auto Scaling застосовується для автоматичного збільшення/ зменшення кількості серверів при різних навантаженнях, забезпечуючи, при пікових навантаженнях, стабільність роботи мережі. Сценарії навантаження включають пікове навантаження, стійке високе навантаження та короткотривалі сплески. Для кожного сценарію зафіксовано час відгуку сервісів, кількість підключень, які додаються/видаляються, а також середню пропускну здатність. Маємо три основних сценарії для тестування системи: нормальне навантаження – у цьому сценарії моделювалося одночасне підключення 100 користувачів, які генерували типовий трафік і це дозволяло оцінити базові показники роботи системи, такі як затримка, пропускна здатність і стабільність; пікове навантаження – цей сценарій моделював одночасне підключення 1000 користувачів із підвищеними затримками на маршрутизацію та оцінювалось, наскільки ефективно система справляється з екстремальними умовами роботи та чи забезпечується стабільність у таких умовах; відмова вузла – цей сценарій емулює відмову одного з ключових мережевих комутаторів, що дозволило перевірити здатність системи до відмовостійкості та ефективність балансувальників навантаження в умовах аварійних ситуацій. Результати показали, що при збільшенні кількості користувачів до 1000 затримка зросла до 100 мс, однак система зберігала функціональність навіть у пікових умовах. На рисунку 3 показано залежність часу відгуку системи від кількості активних користувачів, що дозволяє оцінити її продуктивність при різних сценаріях навантаження.

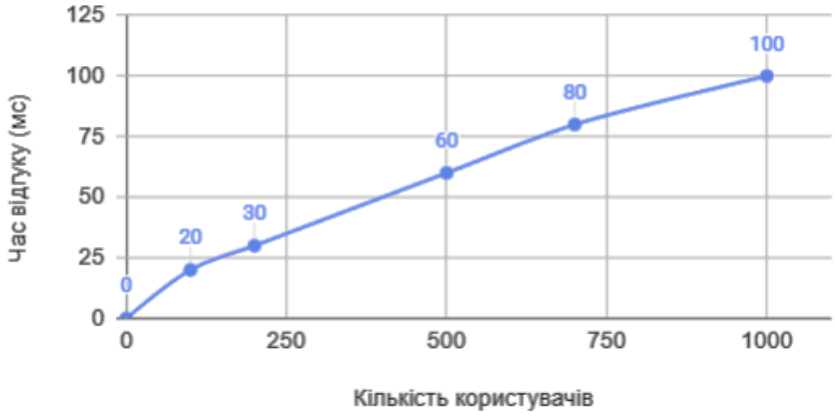

Рис. 3. Графік часу відгуку системи

Проведено серію експериментів у системі моделювання із наступними параметрами: кількість локальних клієнтів 50-500; вихідний трафік 10-500 Mbps; інтервали опитування Prometheus 15 s; пороги T_up = 75% CPU utilization; T_down = 30% CPU utilization; N_min = 2, N_max = 20. Результати подано у вигляді опису ключових метрик (latency, throughput, number_of_instances, downtime) для кожного сценарію. Сценарій 1 (пікове навантаження): при різкому зростанні трафіку до 400 Mbps алгоритм автоскейлінгу додав підключення за 45–60 s, що призвело до стабілізації використання CPU на рівні ~65% і зниження середньої затримки на 32%. Сценарій 2 (стійке високе навантаження): при навантаженні 250 Mbps система утримувала 8-10 підключень, середня пропускна здатність стала стабільною, причому витрати на хмарні ресурси за моделювання 24 годин склали орієнтовно 0.8 від вартості повної міграції в хмару. Сценарій 3 (короткотривалі сплески): cooldown timer у 300 s дозволив уникнути частого додавання/видалення підключень, але збільшив час реакції на повторний сплеск, тому рекомендовано використовувати адаптивний cooldown. Результати тестування продуктивності за різними сценаріями, яка містить такі дані: середній час відгуку, затримка, пропускна здатність, втрати пакетів і стабільність системи наводяться в (табл. 2).

Таблиця 2

Результати тестування продуктивності за різними сценаріями

Сценарій	Середній час відгуку (мс)	Середня затримка (мс)	Пропускна здатність (Мбіт/с)	Втрати пакетів (%)	Стабільність роботи системи
Нормальне навантаження	20	15	950	0.5	Висока. Система працює стабільно.
Пікове навантаження	100	85	700	1.8	Задовільна. Система функціонує.
Відмова вузла	50	35	850	1.0	Висока. Швидке відновлення.

Моделювання показало, що інтеграція хмарних сервісів забезпечує високий рівень продуктивності навіть за екстремальних умов, що робить цей підхід перспективним для широкого впровадження в корпоративних мережах.

Порівняння методів впровадження хмарних сервісів, при керуванні мережею, свідчить про те, що конкретні потреби підприємств (кінцевих користувачів), їх ресурси та вимоги до безпеки, масштабованість, продуктивність і вартість, впливають на вибір оптимальної моделі (рис. 4). Приватна хмара, зазвичай, створює власну інфраструктуру при зберіганні та обробці даних у межах закритого середовища, доступ до якого має тільки обмежена кількість авторизованих осіб. Публічна хмара, через сторонніх провайдерів, надає доступ до обчислювальних ресурсів, таких як AWS, Microsoft Azure або Google Cloud Platform. Гібридна хмара дозволяє підприємствам використовувати переваги обох моделей, об'єднуючи можливості приватної та публічної хмар, Для того, щоб оцінити кожний підхід виділяємо чотири основні критерії:

– безпека: завдяки ізоляції ресурсів приватна хмара надає максимальний рівень безпеки; безпека у публічній хмарі залежить від політики провайдера; гібридна забезпечуючи баланс, поєднуючи обидві можливості;

– продуктивність: стабільна продуктивність у приватній хмарі; у публічній і гібридній продуктивність залежить від інтеграції та умов провайдера;

– вартість: найдорожчою є приватна хмара бо має високі витрати на підтримку інфраструктури; публічна більш економний варіант; гібридна має складність впровадження і середні витрати;

– масштабованість: легкими в масштабуванні мережевої інфраструктури є публічна та гібридна хмари, показуючи найкращі результати.

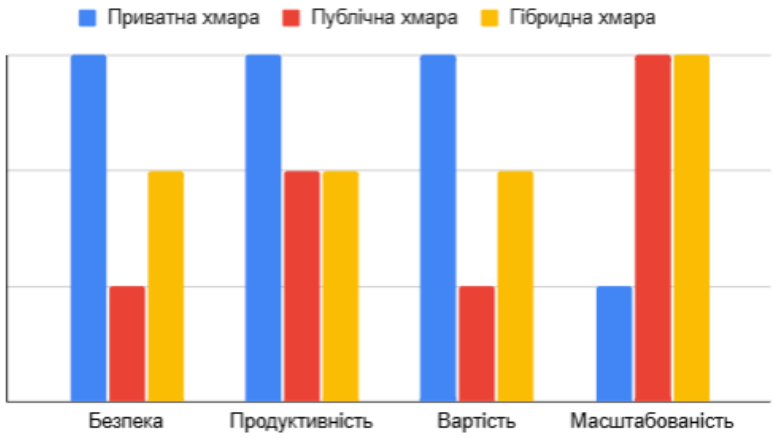

Рис. 4. Порівняння хмарних підходів

Узагальнюючи результати дослідження, можна стверджувати, що правильний вибір хмарних сервісів є ключовим чинником підвищення ефективності діяльності організації, оскільки він сприяє оптимізації витрат, гарантує необхідний рівень захисту даних і забезпечує створення гнучкої інфраструктури, здатної оперативно реагувати на виклики сучасного ринку та технологічного прогресу.

Висновки. У роботі проведено аналіз провідних хмарних платформ, здійснена реалізація та верифікація моделі управління інфраструктурою розподілених мереж із використанням хмарних сервісів, що забезпечує: централізований моніторинг із використанням Prometheus та Grafana; адаптивне масштабування ресурсів на основі метрик та безпеку передачі даних між локальними вузлами і хмарними сервісами. Результати дослідження підтвердили ефективність застосування гібридної моделі інтеграції хмарних сервісів у розподіленій інфраструктурі. Такий підхід забезпечує оптимальний баланс між безпекою, масштабованістю та економічною ефективністю. Приватна частина гібридної інфраструктури виконує роль безпечного сховища для конфіденційних даних. Це дозволяє підприємствам дотримуватись: політик безпеки; стандартів та внутрішніх вимог до зберігання інформації. Доступ до таких даних здійснюється лише через захищені тунелі (TLS/VPN), що мінімізує ризики втрати конфеденційності. Під час збільшення кількості запитів або активності користувачів система автоматично активує механізми автоскейлінгу, які дозволяють тимчасово задіяти ресурси публічної хмари. Це значно знижує час простою та гарантує безперервність обслуговування. При нормалізації навантаження ресурси звільняються і це, зазвчай, економить кошти. Одним із основних викликів гібридної моделі є мережеві затримки, спричинені передачею даних між локальними вузлами та хмарною інфраструктурою. Для зменшення цього ефекту рекомендовано застосовувати Edge-сервери або кешування на рівні додатків. У гібридних системах особливу увагу слід приділяти шифруванню каналів зв’язку та аутентифікації між компонентами. Використання TLS, IPsec та багатофакторної автентифікації дозволяє гарантувати цілісність даних і захист від несанкціонованого доступу. Використання сервісів лише одного хмарного провайдера може створити технологічну залежність, тому доцільно застосовувати

контейнери (Docker, Kubernetes), які забезпечують портативність додатків між різними платформами. Крім того, використання GitOps-підходу дозволяє централізовано керувати інфраструктурою, мінімізуючи ризики зміни постачальника. Тестування показало, що гібридна модель дозволяє знизити середнє навантаження на локальні сервери на 35–50%, при цьому підвищивши продуктивність системи на 20–30% у періоди пікової активності. Таким чином, гібридна архітектура є найбільш збалансованим рішенням для корпоративних мереж, які поєднують високу безпеку локальної інфраструктури із гнучкістю хмарних технологій.

Список використаних джерел:

1. Рудьковський О. Р., Киричек Г. Г. Програмний комплекс з підтримки розподіленої взаємодії мережевих пристроїв та додатків. *Вчені записки ТНУ ім. В.І. Вернадського*. Серія «Технічні науки». 2021. Вип. 32(71). № 2. С. 229–234. DOI https://doi.org/10.32838/2663-5941/2021.2-1/36.
2. Zhang F., Cao J., Li K., Khan S. U., Hwang K. Multi-objective scheduling of many tasks in cloud platforms. *Future generation computer systems*, 2014, 37: 309–320. DOI https://doi.org/10.1016/j.future.2013.09.006.
3. Tiahunova M., Tronkina O., Kirichek G., Skrupsky S. The Neural Network for Emotions Recognition under Special Conditions. In: *CMIS*. Vol-2864, 2021. p. 121–134. URL: https://ceur-ws.org/Vol-2864/paper11.pdf.
4. Киричек Г. Г., Гаркуша В. Ю. Віртуалізація хостів на основі Proxmox VE в умовах надлишкового використання ресурсів. *Вчені записки ТНУ імені В.І. Вернадського*. Серія «Технічні науки». 2021. Вип. 32 (71). № 1. С. 78–84. DOI https://doi.org/10.32838/2663-5941/2021.1-1/13.
5. Sharma R. A review on software defined networking. Int J Sci Res Comput Sci Eng Inf Technol, 2021, 11–14. DOI https://doi.org/10.32628/CSEIT21728.
6. Kreutz D., Ramos F.M., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 2014. 103.1, 14–76. DOI https://doi.org/10.1109/JPROC.2014.2371999.
7. Yanamala A. K. Y. Emerging challenges in cloud computing security: A comprehensive review. *International Journal of Advanced Engineering Technologies and Innovations*, 2024. 1.4, 448–479. URL: https://thesisexpertsofficial.com/pdfs/4.pdf
8. Rowshanrad S., Namvarasl S., Abdi V., Hajizadeh M., Keshtgary M. A survey on SDN, the future of networking. Journal of Advanced Computer Science & Technology, 2014. 3.2, 232. DOI https://doi.org/10.14419/jacst.v3i2.3754.
9. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., Ramanathan R., Iwata Y., Inoue H., Hama T., Shenker S. Onix: A distributed control platform for large-scale production networks. In: 9th USENIX symposium on operating systems design and implementation (OSDI 10). 2010. URL: https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Koponen.pdf.
10. Mell P., Grance, T. The NIST definition of cloud computing. 2011. URL: https://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf.
11. Subashini S., Kavitha V. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 2011. 34.1, 1–11. DOI https://doi.org/10.1016/j.jnca.2010.07.006.
12. Zakiyyah R., Permana D., Valenecio D., Chowanda A., Muliono Y., Izdihar Z.N. Security Challenges and Issues in Cloud Computing. In: 2024 International Symposium on Networks, Computers and Communications(ISNCC). *IEEE*, 2024. p. 1–6. DOI https://doi.org/10.1109/ISNCC62547.2024.10758966.
13. Thavi R., Jhaveri R., Narwane V., Gardas B., Jafari Navimipour N. Role of cloud computing technology in the education sector. *Journal of Engineering, Design and Technology*, 2024. 22.1, 182–213. DOI https://doi.org/10.1108/JEDT-08-2021-0417.
14. Kumar S. U. B. O. D. H., Athavale V. A., Kartikey D. I. V. Y. E. Security issues in cloud computing: A holistic view. *International Journal of Internet of Things and Web Services*, 2021. 6, 18–29. URL: https://www.iaras.org/iaras/filedownloads/ijitws/2021/022-0003(2021).pdf
15. Dillon T., Wu C., Chang E. Cloud computing: issues and challenges. In: 2010 24th IEEE international conference on advanced information networking and applications. Ieee, 2010. p. 27–33. DOI https://doi.org/10.1109/AINA.2010.187.
16. Confidently L.A.O., Chakraborty M., Kundan A.P. Monitoring Cloud-Native Applications. 2021. URL: https://link.springer.com/book/10.1007/978-1-4842-6888-9
17. Vuorinen S. Monitoring Integration Systems and Visualization. 2022. URL: https://www.utupub.fi/bitstream/handle/10024/154329/Vuorinen_Simo_opinnayte.pdf?sequence=1.
18. Beyer B., Jones C., Petoff J., Murphy N. R. Site reliability engineering: how Google runs production systems. " O'Reilly Media, Inc.", 2016.
19. von Goethe J.W. Prometheus. Voltaire Press, 2024.
20. Shaji A. S., George M. M. Elastic stack: A comprehensive overview. 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2024, 1–5. DOI https://doi.org/10.1109/RAICS61201.2024.10690099.
21. Chakraborty M., Kundan, A.P. Grafana. In: Monitoring cloud-native applications: Lead agile operations confidently using open source software. Berkeley, CA: Apress, 2021. p. 187–240. URL: https://link.springer.com/chapter/10.1007/978-1-4842-6888-9_6.

References:

1. Rudkovskyi, O. R., Kyrychek, H. H. (2021). Prohramnyi kompleks z pidtrymky rozpodilenoi vzaiemodii merezhevykh prystroiv ta dodatkiv [A software complex supporting the distributed interaction of network devices and applications]. *Scientific notes of TNU named after V.I. Vernadskyi*. Series "Technical Sciences", 32(71), 2, 229–234 DOI https://doi.org/10.32838/2663-5941/2021.2-1/36. [in Ukrainian].

2. Zhang, F., Cao, J., Li, K., Khan, S. U., Hwang, K. (2014). Multi-objective scheduling of many tasks in cloud platforms. *Future generation computer systems*, 37, 309–320. DOI https://doi.org/10.1016/j.future.2013.09.006.

3. Tiahunova, M., Tronkina, O., Kirichek, G., Skrupsky, S. (2021). The Neural Network for Emotions Recognition under Special Conditions. In: *CMIS*. Vol-2864, p. 121–134. Retrieved from: https://ceur-ws.org/Vol-2864/paper11.pdf.

4. Kyrychek, H. H., Harkusha, V. Yu. (2021). Virtualizatsiia khostiv na osnovi Proxmox VE v umovakh nadlyshkovoho vykorystannia resursiv [Virtualization of hosts based on Proxmox VE in conditions of excessive use of resources]. *Scientific notes of TNU named after V.I. Vernadsky*i. Series "Technical Sciences". 32 (71), 1, 78–84. DOI https://doi.org/10.32838/2663-5941/2021.1-1/13. [in Ukrainian].

5. Sharma, R. (2021). A review on software defined networking. *Int J Sci Res Comput Sci Eng Inf Technol*, 11–14. DOI https://doi.org/10.32628/CSEIT21728.

6. Kreutz D., Ramos F.M., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 2014, 103.1: 14-76. DOI https://doi.org/10.1109/JPROC.2014.2371999.

7. Yanamala, A. K. Y. (2024). Emerging challenges in cloud computing security: A comprehensive review. *International Journal of Advanced Engineering Technologies and Innovations,* 1.4, 448–479. Retrieved from: https://thesisexpertsofficial.com/pdfs/4.pdf

8. Rowshanrad, S., Namvarasl, S., Abdi, V., Hajizadeh, M., Keshtgary, M. (2014). A survey on SDN, the future of networking. Journal of Advanced Computer Science & Technology, 3.2, 232. DOI https://doi.org/10.14419/jacst.v3i2.3754.

9. Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., Shenker, S. (2010). Onix: A distributed control platform for large-scale production networks. In: 9th USENIX symposium on operating systems design and implementation (OSDI 10). Retrieved from: https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Koponen.pdf.

10. Mell, P., Grance, T. (2011). The NIST definition of cloud computing. Retrieved from: https://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf.

11. Subashini, S., Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications. 34.1, 1–11. DOI https://doi.org/10.1016/j.jnca.2010.07.006.

12. Zakiyyah, R., Permana, D., Valenecio, D., Chowanda, A., Muliono, Y., Izdihar, Z. N. (2024). Security Challenges and Issues in Cloud Computing. In: 2024 International Symposium on Networks, Computers and Communications(ISNCC). *IEEE*, p. 1–6. DOI https://doi.org/10.1109/ISNCC62547.2024.10758966.

13. Thavi, R., Jhaveri, R., Narwane, V., Gardas, B., Jafari Navimipour, N. (2024). Role of cloud computing technology in the education sector. *Journal of Engineering, Design and Technology*, 22.1, 182–213. DOI https://doi.org/10.1108/JEDT-08-2021-0417.

14. Kumar, S. U. B. O. D. H., Athavale, V. A., Kartikey, D. I. V. Y. E. (2021). Security issues in cloud computing: A holistic view. *International Journal of Internet of Things and Web Services,* 6, 18–29. Retrieved from: https://www.iaras.org/iaras/filedownloads/ijitws/2021/022-0003(2021).pdf.

15. Dillon, T., Wu, C., Chang, E. (2010). Cloud computing: issues and challenges. In: 2010 24th IEEE international conference on advanced information networking and applications. Ieee, p. 27–33. DOI https://doi.org/10.1109/AINA.2010.187.

16. Confidently L. A. O., Chakraborty M., Kundan A. P. (2021). Monitoring Cloud-Native Applications. Retrieved from: https://link.springer.com/book/10.1007/978-1-4842-6888-9

17. Vuorinen, S. (2022). Monitoring Integration Systems and Visualization. Retrieved from: https://www.utupub.fi/bitstream/handle/10024/154329/Vuorinen_Simo_opinnayte.pdf?sequence=1.

18. Beyer, B., Jones, C., Petoff, J., Murphy, N. R. (2016). Site reliability engineering: how Google runs production systems. " O'Reilly Media, Inc."

19. von Goethe J.W. Prometheus. Voltaire Press, 2024.

20. Shaji, A. S., George, M. M. (2024). Elastic stack: A comprehensive overview. 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 1–5. DOI https://doi.org/10.1109/RAICS61201.2024.10690099.

21. Chakraborty, M., Kundan,, A. P. (2021). Grafana. In: Monitoring cloud-native applications: Lead agile operations confidently using open source software. Berkeley, CA: Apress, p. 187–240. Retrieved from: https://link.springer.com/chapter/10.1007/978-1-4842-6888-9_6.

Дата надходження статті: 12.10.2025
Дата прийняття статті: 10.11.2025
Опубліковано: 30.12.2025