УДК 004.45:004.75
DOI https://doi.org/10.32782/2663-5941/2025.6.2/12

Волк М.О.
Харківський національний університет радіоелектроніки

Бугрій А.М.
Харківський національний університет радіоелектроніки

Полозов Д.М.
Харківський національний університет радіоелектроніки

Ємець Д.С.
Харківський національний університет радіоелектроніки

Олефіренко М.Є.
Харківський національний університет радіоелектроніки

Литовка Д.В.
Харківський національний університет радіоелектроніки

ІНТЕГРОВАНА АРХІТЕКТУРА ОРКЕСТРАЦІЇ КОНТЕЙНЕРИЗОВАНИХ СЕРВІСІВ У МУЛЬТИХМАРНО-EDGE СЕРЕДОВИЩІ

У статті представлено результати дослідження та проєктування інтегрованої архітектури оркестрації контейнеризованих сервісів у середовищах, що поєднують мультихмарні платформи та граничні обчислення. Метою роботи є створення уніфікованої моделі керування контейнерами, здатної забезпечувати узгоджену взаємодію між кластерами різних типів і масштабів. Запропонований підхід поєднує централізоване управління ресурсами з децентралізованим виконанням сервісів на периферійних пристроях, що дає змогу досягти балансу між швидкодією, надійністю та економічністю.

Інтегрована архітектура базується на використанні Kubernetes як контрольного шару та легковагових систем K3s і Nomad на периферії. У роботі сформульовано узагальнену математичну модель розподілу навантаження, у якій мінімізується середній час відповіді системи при обмеженнях на ресурси та вартість. Для досягнення гнучкого балансу між продуктивністю та витратами введено нормалізовані вагові коефіцієнти, що можуть змінюватися динамічно залежно від стану системи та каналів зв'язку, показані типові значення коефіцієнтів в залежності від стратегії управління. Запропоновано структурно-функціональну схему архітектури, що відображає взаємодію контрольного шару з хмарними й edge-кластерами.

Експериментальні дослідження проведено на тестовому стенді, який поєднував вузли AWS, Azure, GCP і периферійні пристрої з K3s. Отримані результати демонструють скорочення середнього часу відповіді до 35 % і зниження використання ресурсів до 20 % порівняно з базовою конфігурацією без інтеграції. Розроблена модель може бути впроваджена в інформаційних системах аналітики даних, промисловому інтернеті речей та інтелектуальних міських платформах. Таким чином, інтегрована архітектура є ефективним напрямом розвитку сучасних розподілених інформаційних систем, які потребують масштабованості, адаптивності та низьких затримок обробки даних.

Ключові слова: *граничні обчислення, мультихмарна архітектура, оркестрація контейнерів, оптимізація ресурсів, розподілені системи, інформаційні системи, хмарні обчислення, балансування навантаження.*

Постановка проблеми. Розвиток сучасних інформаційних технологій супроводжується стрімким зростанням обсягів даних і підвищенням вимог до швидкодії обчислювальних систем. Класичні хмарні архітектури, які донедавна вважалися універсальним рішенням для масштабованих обчислень, сьогодні стикаються з обмеженнями, пов'язаними з затримками передавання даних, перевантаженням каналів і високою вартістю утримання інфраструктури. Поява концепції edge computing частково вирішила ці проблеми, забезпечивши виконання обчислень ближче до джерел даних, але водночас створила нові виклики – необхідність керування великою кількістю розподілених вузлів і забезпечення їхньої узгодженої роботи з хмарними середовищами.

Сучасні сервіси часто функціонують у змішаному середовищі, де частина компонентів розгорнута у великих хмарних кластерах (AWS, Azure, GCP), а інша – у периферійних пристроях або мікросервісах на локальних вузлах. У таких умовах ефективна оркестрація контейнерів стає ключовим завданням, оскільки саме вона визначає, де й коли мають бути запущені конкретні сервіси, як відбувається балансування навантаження та яким чином забезпечується безперервність роботи системи.

Попри значний прогрес у розвитку систем керування контейнерами, існуючі рішення не забезпечують повної сумісності між мультихмарними та edge-середовищами. Kubernetes добре масштабується, проте має високу ресурсоємність, що ускладнює його використання на пристроях з обмеженими характеристиками. Натомість легковагові фреймворки, такі як K3s або Nomad, чудово працюють у периферійних сценаріях, але не підтримують централізованого керування кількома кластерами одночасно. У результаті виникає фрагментованість інфраструктури: кожне середовище функціонує автономно, без єдиного механізму синхронізації та спільного управління політиками.

Ця роз'єднаність призводить до низки проблем. По-перше, ускладнюється розгортання сервісів – кожен кластер потребує власних конфігурацій і сценаріїв оновлення. По-друге, відсутність єдиного моніторингу не дозволяє оперативно оцінювати стан усієї системи. По-третє, різні підходи до балансування навантаження між хмарою та периферією створюють надмірне споживання ресурсів, що безпосередньо впливає на економічну ефективність.

З практичного погляду проблема також полягає у відсутності універсальної моделі розподілу навантаження, яка б одночасно враховувала три важливі фактори: час відповіді, доступність ресурсів та вартість їх використання. Традиційні моделі оптимізації орієнтовані лише на один із цих критеріїв, що призводить до неузгоджених рішень у гібридному середовищі.

Ще однією суттєвою проблемою є нестача стандартизованих механізмів інтеграції між різнорідними системами оркестрації. Попри наявність численних API та інструментів, немає єдиного протоколу взаємодії між Kubernetes і легковаговими edge-оркестраторами. Це ускладнює побудову цілісного керуючого шару, який міг би приймати рішення про розміщення контейнерів на основі узагальненої інформації про стан усіх кластерів.

Таким чином, актуальним є завдання створення інтегрованої архітектури оркестрації контейнерів, що поєднує централізоване управління, характерне для хмарних платформ, із гнучкістю та близькістю до користувача, притаманною периферійним обчисленням. Така система повинна забезпечувати: уніфікований контрольний шар із можливістю моніторингу та координації різних кластерів; єдиний набір політик розгортання, масштабування та безпеки; мінімальні затримки при взаємодії між компонентами; адаптивність до змін навантаження та стану мережі.

Побудова подібної системи потребує узгодження теоретичних моделей із практичними інструментами, що існують у сучасних екосистемах DevOps. Вона має спиратися на математичну формалізацію процесів розподілу навантаження, яка дозволить кількісно оцінити ефективність запропонованих рішень і здійснити порівняння з традиційними підходами.

Отже, постає комплексна науково-практична задача – створення інтегрованої моделі оркестрації контейнерів, що забезпечує прозоре керування мультихмарними та edge-компонентами, дозволяє оптимізувати використання обчислювальних ресурсів і гарантує стабільність роботи системи за умов мінливих навантажень.

Аналіз останніх досліджень і публікацій. Проблематика оркестрації контейнерів у розподілених середовищах активно розглядається у науковій та технічній літературі останніх років. Значну увагу приділено вдосконаленню систем Kubernetes і Docker Swarm, які стали де-факто стандартом у сфері керування мікросервісами [1–2]. Однак більшість робіт фокусуються переважно на централізованих хмарних рішеннях, не враховуючи особливостей периферійних сценаріїв, де обмежені ресурси й нестабільні з'єднання

потребують інших підходів до розподілу навантаження.

У низці досліджень [3–4] запропоновано застосування легковагових фреймворків, таких як K3s, MicroK8s або Nomad, для розгортання контейнерів на edge-пристроях. Ці рішення демонструють ефективність при малих обсягах обчислень, однак не забезпечують глобальної координації між кількома периферійними доменами та не дозволяють синхронізувати політики керування з хмарними платформами. Відсутність єдиного контрольного шару зумовлює необхідність розроблення інтегрованих моделей, здатних узгоджувати роботу гетерогенних кластерів.

Окрему групу становлять дослідження у сфері multi-cloud orchestration [2,5], у яких розглядаються механізми уніфікованого управління ресурсами різних провайдерів. Роботи цього напряму акцентують увагу на питаннях сумісності API, балансуванні навантаження та розподіленні сервісів між хмарами. Проте більшість запропонованих рішень розраховані на високопродуктивні кластери й не адаптовані до інтеграції з периферійними системами. Таким чином, залишається відкритим питання узгодження двох парадигм – мультихмарності та edge computing.

Серед сучасних підходів, близьких до розглядуваної теми, можна відзначити роботи з побудови federated Kubernetes clusters, які передбачають створення централізованого контролера для кількох віддалених кластерів [6]. Хоча такі рішення підвищують керованість інфраструктури, вони не враховують специфіку динамічного розподілу обчислень між центральною хмарою та периферією. Крім того, більшість моделей не мають аналітичного апарату, який дозволяв би кількісно оцінити ефективність інтегрованої системи.

Проведений аналіз показує, що наявні підходи охоплюють окремі аспекти проблеми – хмарну або периферійну оркестрацію, – але не формують єдиного універсального рішення. Саме тому постає потреба у розробленні інтегрованої архітектури, яка об'єднує централізоване управління мультихмарними кластерами з легковаговою оркестрацією на периферії, забезпечуючи при цьому формалізовану модель оцінювання ефективності [7].

Постановка завдання. Проведений аналіз сучасних публікацій засвідчив наявність суттєвого розриву між двома підходами до оркестрації контейнеризованих сервісів: централізованими системами керування, які домінують у хмарних середовищах, та децентралізованими моделями, що використовуються на периферії. У межах мультихмарних інфраструктур вирішальним є питання інтеграції цих двох парадигм у єдину керовану систему.

Проблема, яка стоїть перед дослідженням, полягає у створенні інтегрованої моделі оркестрації контейнерів, здатної забезпечити спільне функціонування хмарних і edge-компонентів при мінімальних витратах часу та ресурсів. Така модель повинна уніфікувати процеси розгортання, моніторингу та масштабування, зберігаючи при цьому гнучкість і автономність локальних вузлів.

Враховуючи це, основну мету дослідження сформульовано так: розробити узагальнену архітектуру оркестрації контейнеризованих сервісів, що поєднує мультихмарні та периферійні обчислення в єдиній керованій моделі, забезпечуючи адаптивний розподіл навантаження і високу ефективність використання ресурсів.

Виклад основного матеріалу. Запропонована інтегрована архітектура оркестрації контейнерів ґрунтується на ідеї поєднання централізованого керування, властивого хмарним платформам, із децентралізованим виконанням, характерним для периферійних обчислень. Її концепція полягає в тому, що усі вузли, незалежно від їх фізичного розташування або ресурсної потужності, утворюють єдину логічну інфраструктуру з уніфікованими політиками керування, моніторингу й масштабування.

На вершині архітектури розміщено контрольний шар, побудований на основі Kubernetes, який відповідає за розподіл навантаження, синхронізацію станів і застосування політик безпеки. До його складу входять три ключові компоненти: Policy Manager, що визначає правила розміщення контейнерів; Monitoring Module, який відстежує стан вузлів і телеметричні показники; та Decision Engine, що формує рішення щодо масштабування й перенесення сервісів.

На середньому рівні розташовані хмарні кластери різних провайдерів (AWS, Azure, GCP), об'єднані через єдиний API. Кожен з них функціонує як автономний обчислювальний домен, але підпорядковується спільним політикам керування. Нижній рівень утворюють периферійні вузли з оркестраторами K3s і Nomad, які забезпечують локальне виконання контейнерів у близькості до джерел даних – IoT-пристроїв, сенсорів, камер спостереження тощо.

Взаємодія між рівнями описується множиною зв'язків $\Phi: L_m \rightarrow (L_c \cup L_e)$, де L_c – множина хмарних кластерів (Cloud layer), L_e – множина перифе-

рійних кластерів (Edge layer), L_m – центральний контрольний шар (Management layer), Φ – відображення, яке задає зв'язки між рівнями. Це відображення визначає напрямки передавання даних, сигналів керування та інформації моніторингу.

На функціональному рівні система розглядається як сукупність потоків запитів і керуючих впливів. Потік користувацьких запитів позначимо $Q(t)$, який розподіляється між двома доменами – хмарним ($\Lambda_c(t)$) та гранічним ($\Lambda_e(t)$) відповідно до політики керування:

$$Q(t)=\Lambda_c(t)+\Lambda_e(t),\ 0\le\Lambda_{c,e}(t)\le Q(t).$$

Співвідношення між ними регулюється динамічним коефіцієнтом $P_c(t)\in[0,1]$:

$$\Lambda c(t)=Q(t)\cdot P_c(t),\Lambda e(t)=Q(t)\cdot(1-P_c(t)).$$

Таким чином, система може змінювати частку обчислень, що виконуються в хмарі або на периферії, у режимі реального часу залежно від поточного навантаження, доступності ресурсів чи вартості їх використання. Загальна ціль роботи системи формалізується як задача мінімізації сумарного часу відповіді з урахуванням вартості обчислень

$$\min_{\Lambda_c(t)\,\Lambda_e(t)} \alpha T_c(\Lambda_c(t)) + \beta T_e(\Lambda_e(t)) + \gamma C_c(\Lambda_c(t)),$$

$$\text{ç} \quad \text{óìîâ} \quad \Lambda_c(t)+\Lambda_e(t)=Q(t)$$

$$0\le\Lambda_c(t)\ \Lambda_e(t)\le Q(t)$$

Тут T_c, T_e – середній час відповіді хмарного та edge-рівнів, C_c – витрати на використання хмарних ресурсів, α,β,γ – вагові коефіцієнти, що відображають пріоритет швидкодії, стабільності та вартості. Для практичної реалізації модель використовує адаптивний підхід до визначення вагових коефіцієнтів**.** На основі даних моніторингу система періодично оновлює їхні значення, змінюючи пріоритети між швидкодією та вартістю. Значення вагових коефіцієнтів у типічних сценаріях роботи системи наведено у табл. 1.

Інтегрована модель дозволяє реалізувати адаптивне балансування між хмарними й граничними ресурсами, мінімізуючи затримку та підвищуючи стабільність системи. На відміну від існуючих підходів, запропоноване рішення поєднує аналітичну формалізацію процесів розподілу з практичними механізмами оркестрації, що робить його придатним як для академічних досліджень, так і для промислового впровадження у розподілених інформаційних системах.

Експериментальні дослідження та результати. Для перевірки ефективності запропонованої інтегрованої архітектури було проведено серію експериментів, спрямованих на оцінку її продуктивності, масштабованості та економічної доцільності в порівнянні з базовими підходами. Експерименти виконувалися в умовах, максимально наближених до реальних сценаріїв використання мультихмарних і периферійних систем [8].

Тестове середовище включало три хмарні кластери – AWS, Azure та Google Cloud Platform – по два вузли кожного типу, а також три периферійні пристрої на базі RaspberryPi5 з оркестратором K3s. Усі кластери координувалися центральним контрольним шаром Kubernetes, який реалізував модулі Policy Manager, Monitoring та Decision Engine.

Кожен вузол обслуговував контейнеризований мікросервіс REST-типу, який генерував і обробляв запити користувачів. Для вимірювань використовувалося навантаження у діапазоні від 100 до 1000 запитів за секунду (rps). Для порівняння оцінювалися два режими: Base – незалежні кластери без інтеграції; Proposed – система з централізованим контрольним шаром та адаптивним розподілом навантаження.

Усі експерименти проводилися протягом 10-хвилинних інтервалів при стабільному мережевому каналі з середньою затримкою 25–30 мс між рівнями.

Основним критерієм ефективності було обрано середній час відповіді (Response Time), що характеризує швидкодію всієї системи. Результати наведено у табл. 2.

Як показали вимірювання, інтегрована модель продемонструвала стабільне зниження затримок у всіх тестових сценаріях. При малому навантаженні різниця становила близько 10%, проте з його зростанням перевага інтегрованого підходу ставала більш помітною. При 1000 запитах за секунду середній час відповіді в запропонованій архітектурі був меншим на 35% порівняно з базовою системою.

Таке покращення пояснюється динамічним балансуванням навантаження: система автоматично спрямовувала частину запитів до edge-вузлів, коли затримка у хмарних кластерах перевищувала поріг. Отже, ефект досягався не за рахунок збільшення обчислювальних потужностей, а завдяки оптимізації маршрутизації запитів.

Використання єдиного контрольного шару також дозволило динамічно виводити вузли з експлуатації під час простою сервісів, тим самим зменшуючи витрати на інфраструктуру в мультихмарному середовищі.

У базовій системі середнє використання CPU становило понад 80 %, тоді як у запропонова-

Таблиця 1

Значення вагових коефіцієнтів для типічних сценаріїв

Сценарій	Характеристика задачі	α (Cloud response)	β (Edge response)	γ (Cost)	Пріоритет
Interactive analytics	Швидкість та стабільність	0.45	0.45	0.1	Мінімізація затримки
IoT / Smart City	Значна частина обробки на периферії	0.3	0.6	0.1	Реактивність edge-рівня
Big Data	Довготривалі процеси, важлива економічність	0.3	0.2	0.5	Оптимізація вартості
Video-, audio-streaming	Потребує мінімальної затримки на edge	0.25	0.65	0.1	Перебільшення edge-аналізу
Multi-cloud	Баланс продуктивності і вартості	0.4	0.4	0.2	Збалансоване керування

Таблиця 2

Середній час відповіді системи

Інтенсивність запитів (rps)	100	300	500	700	1000
Base (мс)	83	144	214	279	362
Proposed (мс)	75	115	162	193	241
Зменшення, %	9.8	20.0	24.4	31.2	34.9

ній архітектурі – лише близько 65 %. Водночас кількість активних вузлів скоротилася з шести до чотирьох без погіршення продуктивності. Це свідчить про раціональніше використання обчислювальних ресурсів і менше енергоспоживання, що є критично важливим для розподілених систем, орієнтованих на сталу роботу.

Особливу увагу приділено здатності системи автоматично реагувати на зміни навантаження. У цьому експерименті інтенсивність запитів зростала утричі протягом п'яти хвилин, після чого різко спадала до початкового рівня. У базовій системі кількість реплік залишалася сталою, тоді як у запропонованій архітектурі Decision Engine автоматично збільшував або зменшував кількість активних екземплярів сервісу. Таким чином, система підтримувала стабільний рівень продуктивності, не допускаючи перевантаження вузлів і втрати запитів.

Висновки. У статті запропоновано інтегровану модель оркестрації контейнерів, яка поєднує мультихмарні та периферійні обчислення в єдиній керованій архітектурі. Модель базується на централізованому контрольному шарі Kubernetes і легковагових фреймворках K3s та Nomad, що забезпечують гнучке розгортання сервісів біля джерел даних. Формальне подання процесів розподілу навантаження дало змогу мінімізувати сумарний час відповіді з урахуванням вартості використання ресурсів і затримок між рівнями.

Результати експериментів підтвердили, що інтегрована архітектура знижує середній час відповіді на 30–35 % та покращує ефективність використання ресурсів до 20 % порівняно з базовими конфігураціями. Впровадження адаптивних вагових коефіцієнтів дозволило реалізувати механізм самоналаштування, що підвищує стабільність і продуктивність системи при змінному навантаженні.

Подальші дослідження доцільно спрямувати на розвиток інтелектуальних методів прогнозування навантаження, автоматичну зміну політик розміщення контейнерів і використання технологій машинного навчання для підвищення ефективності прийняття рішень у мультихмарно-edge середовищах.

Список літератури:

1. Kayal P. Kubernetes in Fog Computing: Feasibility Demonstration, Limitations and Improvement Scope. *2020 IEEE 6th WF-IoT. IEEE*, 2020, P. 1–6. DOI: 10.1109/WF-IoT48130.2020.9221340

2. Malhotra S., Yashu F., Saqib M., Divyani F. A Multi-Cloud Orchestration Model Using Kubernetes For Microservices. *Migration Letters.* Volume: 17, No: 6 2020. P. 870-875. DOI:10.2139/ssrn.5194262

3. Valantasis A., Makris N., Korakis T. Orchestration Software for Resource Constrained Datacenters. *Experimental Evaluation. IEEE NetSoft*, 2022. P. 121-126. DOI: 10.1109/NetSoft54395.2022.9844043.

4. Mamchych O., Volk M. A unified model and method for forecasting energy consumption in distributed computing systems based on stationary and mobile devices. *Radioelectronic and Computer Systems,* [S.l.], v. 2024, n. 2. P. 120-135. DOI: https://doi.org/10.32620/ reks.2024.2.10.

5. Volk M., Kozina O., Buhrii A., Osiievskyi S., Kozin M., Volk D., Diachenko D., Turinskyi Y. Devising a method for data consistency at replication in multicloud systems. *Eastern-European Journal of Enterprise Technologies.* Vol. 4 №2 (136), 2025. P. 14-22. DOI: doi.org/10.15587/1729-4061.2025.332189

6. Seth D., Nerella H., Najana M. Tabbassum A. Navigating the Multi-Cloud Maze: Benefits, Challenges, and Future Trends. *International Journal of Global Innovations and Solutions.* June 2024. P. 22. DOI:10.21428/e90189c8.8c704fe4.

7. Волк М.О., Бугрій А.М., Ковтун Є.В., Брестовицький Р.М., Соробей Б.В., Лобач Я.В. Оптимізація ресурсів у хмарних обчисленнях: гібридний підхід до автоматизації операцій та енергозбереження. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки.* Том 35(74) №5 Ч.1. 2024. С. 91–96. DOI: 10.32782/2663-5941/2024.5.1/15

8. Kozina O., Machado J., Volk M., Heiko H., Panchenko V., Kozin M., Ivanova M. Opportunities for Adapting Data Write Latency in Geo-Distributed Replicas of Multicloud Systems. *Future Internet.* 2025. 17(10). 442. DOI: https://doi.org/10.3390/fi17100442

Volk M.O., Buhrii A.M., Polozov D.M., Yemets D.S., Olefirenko M.Ye., Litovka D.V. INTEGRATED ARCHITECTURE FOR ORCHESTRATATION OF CONTAINERIZED SERVICES IN MULTI-CLOUD-EDGE ENVIRONMENT

The article presents the results of the research and design of an integrated architecture for orchestrating containerized services in environments that combine multi-cloud platforms and edge computing. The aim of the work is to create a unified container management model capable of ensuring coordinated interaction between clusters of different types and scales. The proposed approach combines centralized resource management with decentralized execution of services on peripheral devices, which allows achieving a balance between speed, reliability and cost-effectiveness. The integrated architecture is based on the use of Kubernetes as a control layer and lightweight K3s and Nomad systems on the periphery. The paper formulates a generalized mathematical model of load distribution, which minimizes the average system response time under resource and cost constraints. To achieve a flexible balance between performance and costs, normalized weight coefficients are introduced that can change dynamically depending on the state of the system and communication channels; typical values of the coefficients are shown depending on the management strategy. A structural and functional architecture diagram is proposed, reflecting the interaction of the control layer with cloud and edge clusters.

Experimental studies were conducted on a test bench that combined AWS, Azure, GCP nodes and peripheral devices with K3s. The results obtained demonstrate a reduction in average response time of up to 35% and a reduction in resource utilization of up to 20% compared to the basic configuration without integration. The developed model can be implemented in data analytics information systems, industrial Internet of Things and smart city platforms. Thus, integrated architecture is an effective direction for the development of distributed information systems that require scalability, adaptability and low data processing latency.

Key words: *edge computing, multi-cloud architecture, container orchestration, resource optimization, distributed systems, information systems, cloud computing, load balancing.*

Дата надходження статті: 15.11.2025
Дата прийняття статті: 03.12.2025
Опубліковано: 30.12.2025