УДК 004.75

Б. Бачкала, Ю. Тимошин

МОДЕЛІ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ У БАГАТОХМАРНИХ СЕРЕДОВИЩАХ ТА ЇХ ОСОБЛИВОСТІ ЗАСТОСУВАННЯ

Анотація: Сучасні методи балансування навантаження у середовищі хмарних та багатохмарних обчислювальних ресурсів, які зазвичай функціонують в умовах можливих мережевих, апаратних та програмних збоїв, інших зовнішніх впливів, мають залишатися працездатними у разі відмови як окремих компонентів, так і цілих вузлів таких систем.

Ключові слова: алгоритми балансування навантаження, хмарні обчислення, багатохмарні середовища, розподілені системи обробки.

Вступ

У сучасному світі організації все частіше надають перевагу багатохмарним архітектурам, в яких критичні сервіси розподіляються між кількома провайдерами інформаційних послуг [1]. Така архітектура насамперед зменшує ризики, пов’язані із залежністю від окремого провайдера та можливими глобальними відмовами його інфраструктури, за рахунок чого підвищується загальна надійність сервісів. Балансування навантаження – одна з ключових концепцій для побудови надійних багатохмарних систем. Мета балансування – оптимізація використання ресурсів, максимізація пропускної здатності, мінімізація часу відгуку та запобігання перевантаження окремих компонентів системи.

Проблема призначення задач у хмарі залишається складною через непередбачуваність навантажень та гетерогенність ресурсів. Призначення задач без урахування поточного стану системи часто призводить до нерівномірного використання ресурсів, тоді як динамічні підходи можуть вносити додаткові затримки і накладні витрати в масштабних розподілених системах [2]. Різнорідність інфраструктури (відмінності в потужності процесорів, обсягах пам’яті, пропускній здатності мережі) та витрати на міграцію задач між вузлами ускладнюють підтримку оптимального балансу. Ці чинники зумовлюють потребу в адаптивних методах балансування, здатних у реальному часі підлаштовувати розподіл навантаження під змінні умови використання та завчасно запобігати виникненню дисбалансу системи.

Незважаючи на широкий спектр запропонованих підходів до балансування навантаження у багатохмарних середовищах, існуючі методи характеризуються низкою обмежень. До основних проблем належать недостатня адаптивність до динамічних змін навантаження, затримки у прийнятті рішень щодо розподілу ресурсів, значні накладні витрати на моніторинг та управління, а також ризики погіршення рівня якості обслуговування (QoS) за умов пікового навантаження. Крім того, наразі відсутня узагальнена класифікація програмних підходів до балансування, яка враховувала б

 ISSN 1560-8956

ключові критерії ефективності – адаптивність, затримку прийняття рішень, накладні витрати та рівень QoS – що ускладнює вибір оптимальної стратегії для конкретних умов та вимог. Таким чином, актуальним є проведення системного аналізу сучасних моделей балансування навантаження, визначення їх обмежень та розроблення узагальненого впорядкування таких методів з урахуванням зазначених критеріїв ефективності.

Аналіз перспективних рішень

Забезпечення високої доступності та ефективного балансування навантаження є критичними задачами для сучасних багатохмарних розподілених систем. Систематичний огляд методів балансування навантаження та планування задач у хмарних середовищах викладено в дослідженнях [3-4], особливу увагу приділено традиційним алгоритмам балансування (Round-Robin, Least Connection) та методам машинного навчання (Q-Learning, Actor-Critic, LSTM).

Значний внесок у дослідження стратегій балансування навантаження та міграції віртуальних машин (ВМ) у багатохмарних середовищах зроблено в роботах [5-6], де проаналізовано виклики управління інфраструктурою при одночасному використанні кількох хмарних платформ та узагальнено існуючі підходи до балансування навантаження. Встановлено, що поєднання ресурсів різних хмар може підвищити продуктивність системи за умови впровадження ефективних механізмів балансування.

В дослідженнях [7-8] розглянуто архітектурні підходи до підвищення надійності хмарних систем з акцентом на безперервність надання сервісів. Автори наголошують на необхідності інтеграції механізмів балансування навантаження з механізмами відмовостійкості для гарантування безперебійного функціонування сервісів. Показано, що комплексний підхід до розподілу навантаження та захисту даних сприяє підвищенню надійності хмарних інфраструктур.

Питання забезпечення високої доступності та відмовостійкості для розподілених систем цифрової інфраструктури детально досліджено в [9]. У роботі розглянуто як теоретичні засади, так і практичні аспекти проектування та впровадження відмовостійких архітектур. Важливі результати щодо забезпечення відмовостійкості хмарних систем отримано в [10], де проаналізовано метрики надійності та їх застосування для кількісної оцінки стійкості інфраструктури.

Важливість впровадження методів машинного навчання для динамічної оптимізації розподілу трафіку висвітлено в роботах [11-12]. Перехід до динамічного балансування навантаження розглядається як ключовий напрям забезпечення оптимального використання мережевих ресурсів та запобігання перевантаження окремих вузлів.

Таким чином, проаналізовані джерела демонструють наявність широкого спектра підходів до балансування навантаження у багатохмарних середовищах. Водночас питання комплексної оптимізації з урахуванням багатокритеріальних вимог (мінімізація

затримок, оптимізація витрат, гарантування надійності) залишаються недостатньо вирішеними. Це зумовлює необхідність подальшої систематизації знань у цій галузі.

Постановка задачі дослідження

У сучасних багатохмарних середовищах ефективне балансування навантаження є критичною умовою для забезпечення високої продуктивності та надійності функціонування розподілених інформаційних систем. Механізми розподілу навантаження повинні враховувати міжрегіональні мережеві затримки між дата-центрами різних хмарних постачальників, витрати на міжхмарну передачу даних, а також відмінності у гарантованих рівнях надійності обслуговування (Service Level Agreement, SLA). Провідні хмарні платформи (AWS, Azure, GCP) суттєво різняться за архітектурою обчислювальних ресурсів, гарантованими показниками доступності (від 99.9% до 99.995%) та ціновими політиками [13]. Ці відмінності зумовлюють потребу в адаптивних багатокритеріальних методах балансування навантаження з урахуванням гетерогенності інфраструктури. Відсутність належного механізму розподілу навантаження призводить до зростання часу відгуку, нерівномірного використання обчислювальних ресурсів, погіршення рівня QoS та, як наслідок, до порушення умов SLA і фінансових втрат для організацій.

У цьому дослідженні основна увага приділяється виключно програмним алгоритмам балансування навантаження, які є більш гнучкими, масштабованими та, як правило, економічно ефективнішими для хмарних середовищ. Апаратні рішення та спеціалізоване обладнання залишаються поза межами цього дослідження, оскільки їх застосування залежить від конкретної інфраструктури постачальника і не має універсального характеру. Балансування навантаження розглядається на рівні інфраструктури багатохмарного середовища, де розподіл запитів здійснюється між фізичними або віртуальними ресурсами кількох хмарних постачальників. Також поза увагою залишаються механізми балансування на рівні окремих додатків чи мікросервісів, оскільки вони специфічні для конкретних архітектур і не впливають на міжхмарну оптимізацію.

Метою дослідження є проведення системного аналізу та впорядкування сучасних програмних алгоритмів балансування навантаження в багатохмарних середовищах з урахуванням мультикритеріальних вимог (мінімізація затримок, оптимізація витрат, гарантування надійності), визначення їх характеристик, а також формування рекомендацій щодо вибору оптимальних алгоритмів для різних сценаріїв експлуатації та типів навантажень.

Розв’язання задачі дослідження

У цьому дослідженні запропоновано впорядкувати програмні алгоритми балансування навантаження у три класи — статичні, реактивні та проактивні — у контексті їх застосування в багатохмарних середовищах. Кожен із цих класів розглядається

 ISSN 1560-8956

з точки зору архітектурних особливостей, накладних витрат, ефективності використання ресурсів та здатності адаптуватися до динамічних змін навантаження.

Статичні алгоритми балансування навантаження базуються на наперед визначених, незмінних правилах розподілу запитів. Такі рішення функціонують без контуру моніторингу та зворотного зв'язку (рис. 1-2), що спрощує архітектуру і мінімізує накладні витрати на управління. За умов стабільного та передбачуваного навантаження статичний підхід забезпечує задовільну якість обслуговування та рівномірне використання ресурсів, що підходить для типових задач, які часто вирішуються, а стохастичність навантаження має сталий характер. Такий модуль балансування навантаження складається з брокера задач, модуля конфігурацій та утиліти доступу. Брокер організує чергу задач та послідовність їх виконання, передаючи алгоритму балансування, і консолідує результати їх виконання, формуючи відповіді клієнтам на попередні запити. Алгоритм балансування в свою чергу формує відповідний запит до модуля конфігурацій для реальної задачі, в результаті чого призначає конкретний центр обробки даних (ЦОД) хмарних ресурсів для виконання кожного завдання, що показано на схемі послідовності обробки задач (рис.2). Задачі пересилаються на відповідні ЦОД, де з використанням локальних балансувальників розподіляються по віртуальних машинах для виконання. Результати виконання з кожного ресурсу повертаються в чергу задач для консолідації та формування відповіді і пересилання замовнику. Схема обробки різноманітних типових задач є дуже прозорою: у ній кожній задачі наперед визначається фіксований перелік конфігурацій вузлів доступних хмарних ресурсів. Проте основним недоліком є повна відсутність адаптивності: коли виникають раптові стрибки навантаження чи відмови серверів, фіксований розподіл залишається незмінним, призводячи до перевантаження окремих вузлів та зниження ефективності системи загалом.

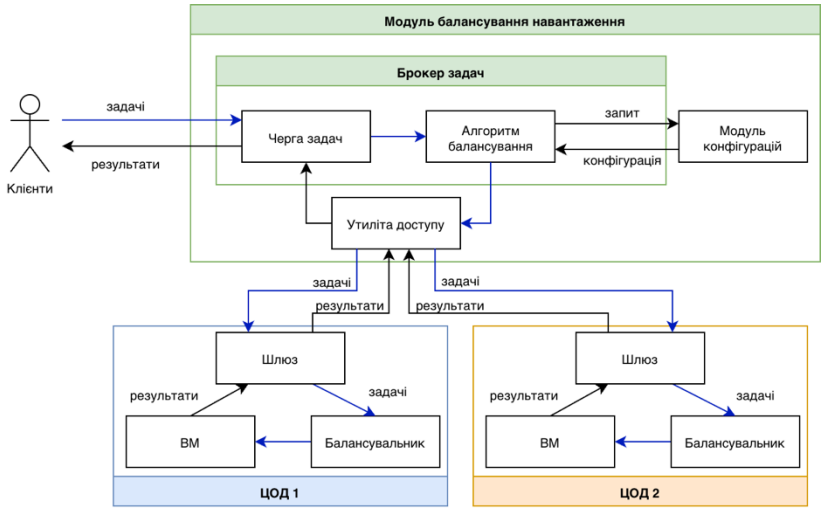

Рисунок 1. Схема структури модуля балансування навантаження на базі статичного алгоритму

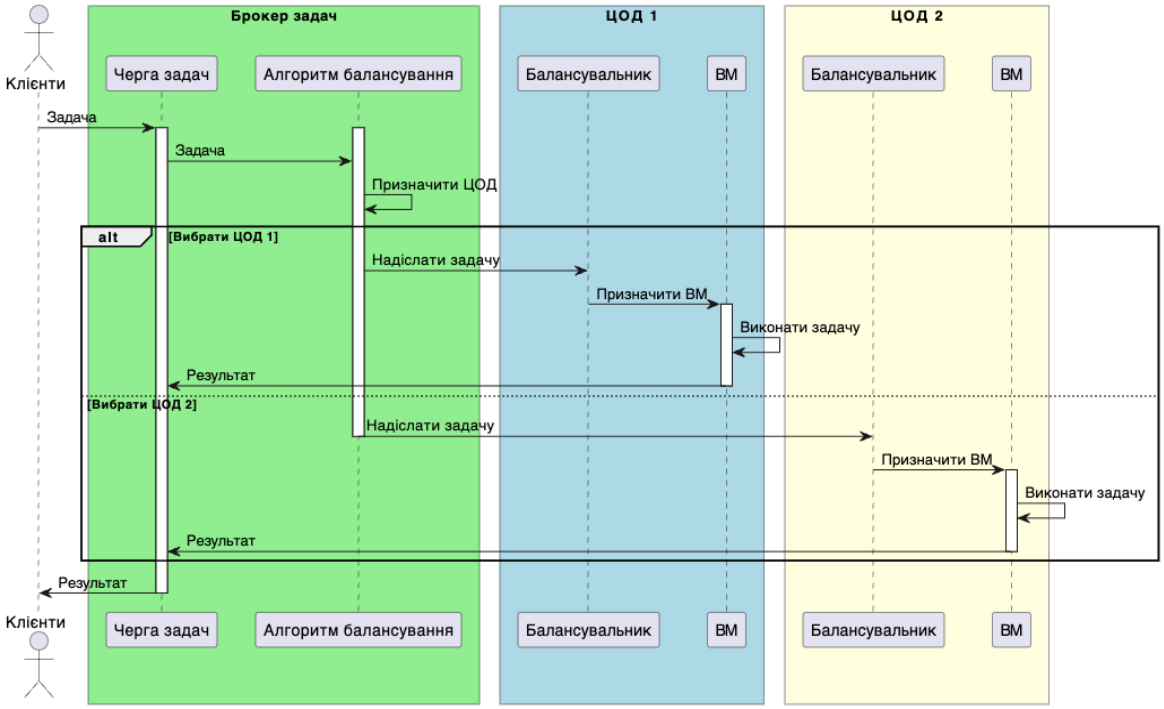

Рисунок 2. Схема послідовності обробки задач на базі статичного алгоритму

В такому алгоритмі брокер задач виконує функцію маршрутизації задач з черги та організує чергу виконаних результатів.

Реактивні алгоритми балансування навантаження здійснюють розподіл запитів на основі моніторингу поточного стану системи. Такі рішення побудовані з використанням зворотного зв'язку відповідних метрик: системні метрики безперервно збираються та аналізуються, а при перевищенні наперед визначених порогів для цих метрик, брокером ініціюється перерозподіл навантаження (рис. 3-4). Такий динамічний підхід підвищує ефективність використання ресурсів за рахунок швидкого виявлення та усунення локальних перевантажень, але вимагає додаткових обчислювальних ресурсів і може впливати на загальний час виконання задачі. Іноді системні метрики окремих задач можуть впливати на рівень QoS. Однак значимим недоліком залишається затримка в реагуванні — алгоритм виявляє перевантаження лише після його виникнення, тому пікові навантаження можуть викликати короткочасне зниження якості обслуговування до моменту, коли система адаптується до нових показників роботи.

Модуль балансування навантаження в такому алгоритмі має модуль агрегації метрик, до якого надходять дані щодо оцінок системних метрик з кожного ЦОД, які збираються в реальному часі відповідними модулями збору метрик. Цей безперервний процес надає можливість брокеру задач отримувати актуальні оцінки завантаженості вузлів в кожен момент часу на основі системних метрик, які вимірюються в кожному ЦОД (продуктивність/ОС/завантаженість CPU, RAM, Disk, Net/ перевантаження сервера, нестача ресурсів, апаратні збої/інше). На основі аналізу системних метрик і обраних критеріїв алгоритм балансування призначає відповідний ЦОД і надсилає

 ISSN 1560-8956

з черги чергову задачу для її для виконання. Збільшення етапів обробки даних щодо наявного стану доступних вузлів розподіленої інфраструктури ресурсів призводить до затримок з реакцією на події в цих вузлах та відповідної зміни призначених ЦОД для своєчасного виконання задач.

Числові дані обраних метрик являють собою тренди оцінок показників що розгортаються у часі та мають традиційні статистики, наприклад, середнє, дисперсію, середньоквадратичну похибку, коефіцієнти авторегресії та кореляції, які можуть також обчислюватися і використовуватися для усереднення оцінок метрик, зменшення їх дисперсії, що призведе до зменшення кількості хибних перерозподілів навантаження брокером задач. Це може збільшити стабільність роботи систем на базі реактивного алгоритму і в значній мірі виключить його хибні спрацьовування.

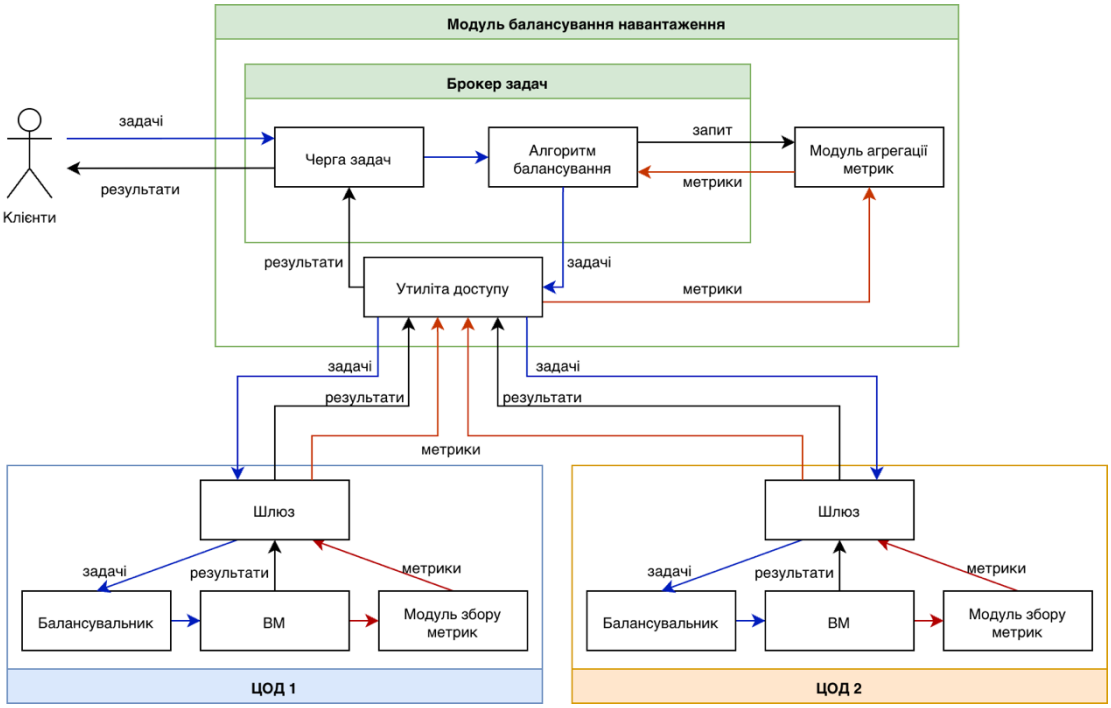

Рисунок 3. Схема структури модуля балансування навантаження на базі реактивного алгоритму

Проактивні алгоритми приймають рішення на основі прогнозів зміни навантаження та доступності обчислювальних ресурсів. Архітектурно проактивний підхід включає спеціалізований модуль прогнозування метрик, який на основі історичних даних та поточної телеметрії передбачає майбутні коливання навантаження (рис. 5-6). На основі отриманих прогнозів система завчасно перерозподіляє запити між хмарними середовищами, забезпечуючи більш рівномірне та ефективне використання обчислювальних ресурсів. Втім, впровадження проактивного алгоритму супроводжується додатковими накладними витратами: необхідно безперервно збирати та постійно обробляти додаткові обсяги даних для прогнозування навантаження та доступності ЦОД на майбутній період часу. Ефективність такого підходу значною мірою залежить

від точності обраної моделі прогнозування — помилкові прогнози можуть призвести до нераціонального розподілення ресурсів. Треба зазначити, що алгоритми прогнозування погано працюють на стрибкоподібних процесах в трендах оцінок метрик, які можуть проявлятися в результаті тимчасових колізій, наприклад, в мережах передачі даних. Будь-які переривання процесів отримання даних за метриками можуть призводити до тимчасових похибок оцінок прогнозів. Потрібен якийсь перехідний час для накопичення нових даних та перенавчання прогнозної моделі, що призводить до використання осереднених оцінок в цей період роботи алгоритму і до тимчасового зменшення його ефективності. Для прогнозування можна використовувати, наприклад, моделі експоненційного згладжування, алгоритми ноніусного прогнозування (розробленого авторами роботи) чи ковзних середніх (Moving Averages) або регресійні моделі. З метою поліпшення оцінок прогнозу можуть додатково використовуватися метрики оцінки прогнозу (середня абсолютна похибка (MAE), середньоквадратична похибка (MSE), корінь середньоквадратичної похибки (RMSE), симетрична середня абсолютна процентна похибка (sMAPE). Також можливе застосування моделей на основі нейронної мережі та на базі нечіткої ситуаційної моделі управління.

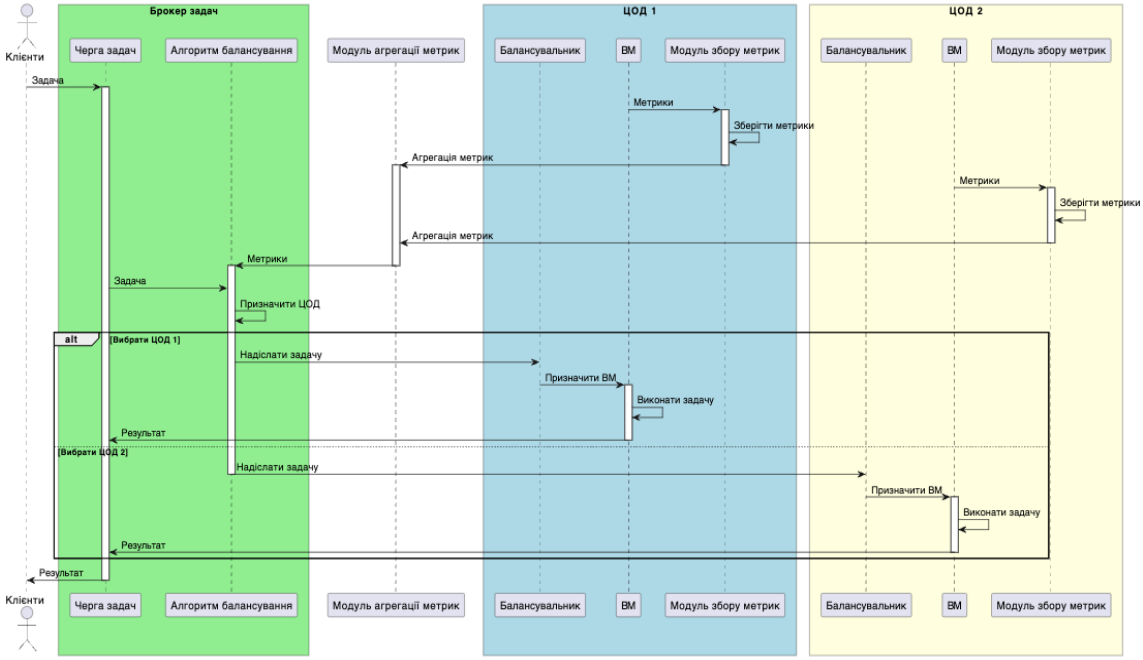

Рисунок 4. Схема послідовності обробки задач
на базі реактивного алгоритму

Додатковий аналіз нових задач з черги дозволить зробити їх класифікацію, використати типові шаблони обробки для їх маршрутизації брокером задач до наявних ЦОД.

Попри зазначені виклики, проактивні алгоритми демонструють найвищу адаптивність до динамічних змін навантаження, оскільки завчасно реагують на пікові навантаження, мінімізуючи ризик порушення умов рівня обслуговування.

 ISSN 1560-8956

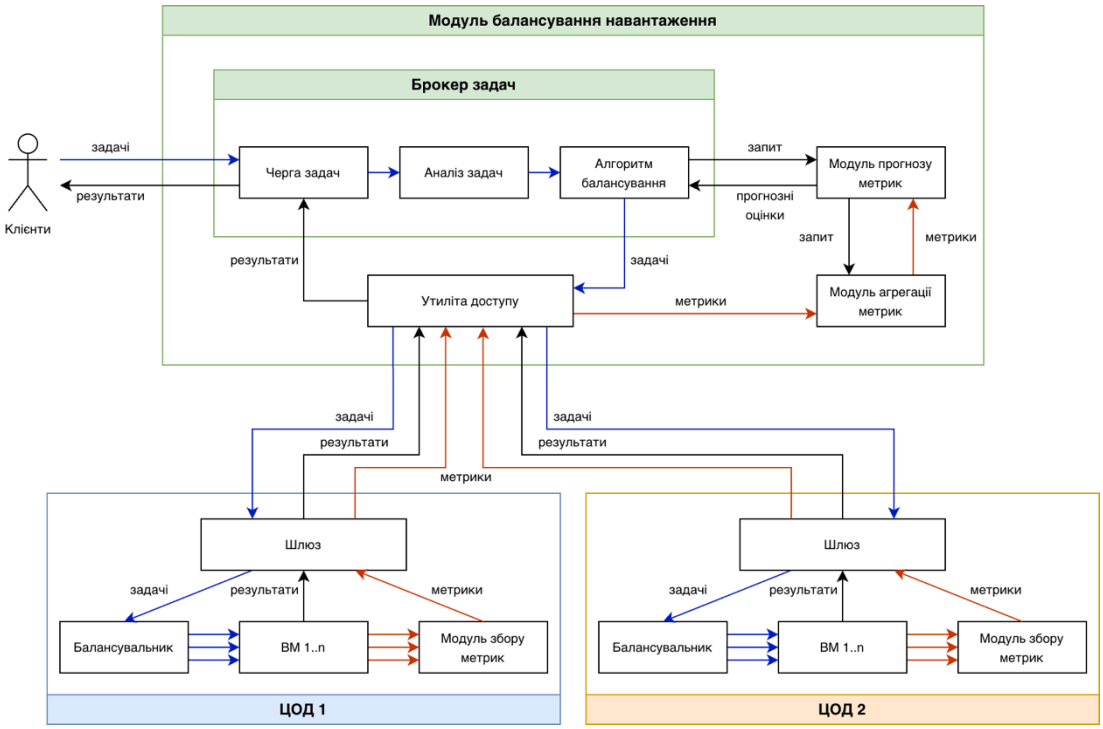

Рисунок 5. Схема структури модуля балансування навантаження на базі проактивного алгоритму

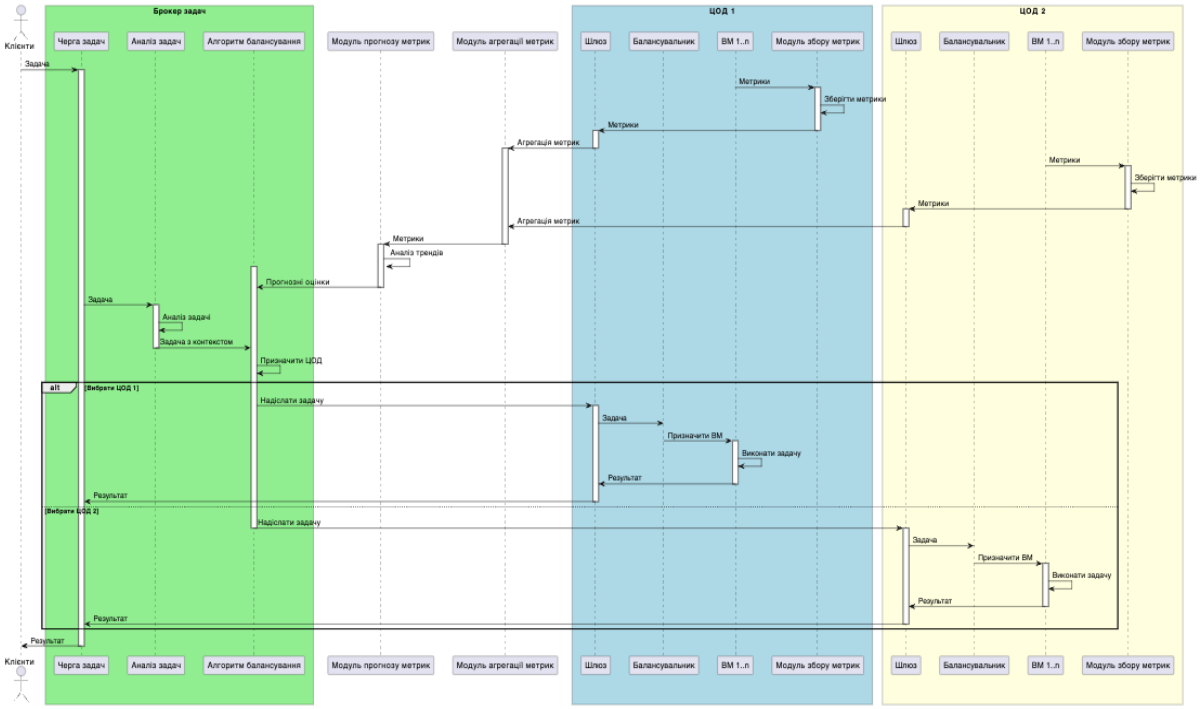

Рисунок 6. Схема послідовності обробки задач на базі проактивного алгоритму

Вибір між статичним, реактивним і проактивним балансуванням навантаження в багатохмарному середовищі залежить від ряду практичних факторів та технічних обмежень. Нижче наведено огляд ключових факторів, що впливають на вибір стратегії балансування навантаження у багатохмарній інфраструктурі.

- Динаміка навантаження. Характер та передбачуваність змін у інтенсивності трафіку значною мірою визначають доцільність вибору алгоритму балансування навантаження.
- Політики SLA. Угоди про рівень обслуговування (SLA) встановлюють жорсткі вимоги щодо доступності та продуктивності сервісів. Обраний алгоритм балансування навантаження має гарантувати виконання цих вимог.
- Вартість. Витрати на обчислювальні ресурси та міжхмарний трафік істотно впливають на вибір стратегії балансування, оскільки ігнорування фінансових обмежень може призвести до зниження економічної ефективності системи.
- Мережеві затримки. Час передачі даних між хмарами та кінцевими користувачами є критичним параметром якості обслуговування. Алгоритм балансування має мінімізувати затримку, скеровувати запити до географічно найближчих або найменш завантажених вузлів.
- Масштабованість. Здатність системи динамічно додавати або звільняти обчислювальні ресурси відповідно до змін навантаження вимагає від алгоритму балансування гнучкості.
- Доступність ресурсів. Кожна хмарна платформа має кінцевий обсяг доступних обчислювальних ресурсів. Алгоритм балансування має враховувати поточну завантаженість та обмеження кожної платформи для оптимізації маршрутизації запитів.

Врахування динамічних коливань навантаження, вимог SLA, а також економічних і технічних обмежень є визначальними факторами для вибору оптимальної стратегії балансування навантаження в конкретних сценаріях застосування.

Висновки

У статті запропоновано та обґрунтовано впорядкування програмних алгоритмів балансування навантаження у багатохмарних середовищах обчислювальних ресурсів. Розглянуто специфіку застосування таких алгоритмів в умовах гетерогенних хмар, відмінностей SLA, міжрегіональних затримок та з урахуванням вартості міжхмарного трафіку; узагальнено переваги та обмеження кожного підходу. Аналіз публікацій останніх років показав, що найкращі результати досягаються шляхом поєднання механізмів моніторингу, адаптивного балансування, оптимізації витрат та проактивного прогнозування навантаження.

Результати виконаного дослідження мають практичну цінність для дослідників та інженерів, оскільки узагальнюють сильні та слабкі сторони кожного класу алгоритмів та пропонують основу для обґрунтованого вибору стратегії балансування. Розроблені схеми балансування навантаження на базі статичних, реактивних та проактивних алгоритмів дозволяють проектувати багатохмарні системи з раціональним використанням ресурсів та урахуванням затримок і вартості міжхмарної взаємодії.

Впровадження обґрунтованих підходів до балансування є передумовою стабільної роботи сучасної багатохмарної інфраструктури та підтримки критично важливих бізнес–процесів. Подальші дослідження доцільно спрямувати на розробку гібридних алгоритмів

 ISSN 1560-8956

балансування, зниження накладних витрат на моніторинг та прогнозування, оцінку енергоефективності та проведення експериментальних досліджень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Yalate A.* Demystifying Multi-Cloud Architecture: Foundational Concepts and Design Patterns // European Journal of Computer Science and Information Technology. – 2025. – Vol. 13, Issue 24. – pp. 51–62. – DOI: 10.37745/ejcsit.2013/vol13n245162

2. *Rajammal K., Chinnadurai M.* Dynamic load balancing in cloud computing using predictive graph networks and adaptive neural scheduling // Scientific Reports. – 2025. – Vol. 15. – № 22181. – DOI: 10.1038/s41598-025-97494-2

3. *Devi N., Dalal S., Solanki K.* A systematic literature review for load balancing and task scheduling techniques in cloud computing // Artificial Intelligence Review. – 2024. – Vol. 57. – № 276. – DOI: 10.1007/s10462-024-10925-w

4. *Sonia; Nath R.* A systematic review of various load balancing approaches in cloud computing utilizing machine learning and deep learning // International Journal of Data Science and Analytics. – 2025. – Vol. 20. – pp. 4273–4295. – DOI: 10.1007/s41060-025-00718-x

5. *Tripathy A., Pandey I.* Load Balancing and VM Migration Strategies in Multi-Cloud Environments: A Systematic Survey. – TechRxiv. – 2025. – DOI: 10.36227/techrxiv.174857667.77379995/v1

6. *Imran M., Ibrahim M., Salah Ud Din M., Atif Ur Rehman M., Kim B. S.* Live virtual machine migration: a survey, research challenges, and future directions // Computers and Electrical Engineering. – 2022. – Vol. 103. – № 108297. – DOI: 10.1016/j.compeleceng.2022.108297

7. *Mishra K., Majhi S.* A state-of-art on cloud load balancing algorithms // International Journal of Computing and Digital Systems. – 2020. – Vol. 9 – № 2. – pp. 201–220. – DOI: 10.12785/IJCDS/090206

8. *Shafiq D. A., Jhanjhi N. Z., Abdullah A.* Load balancing techniques in cloud computing environment: A review // Journal of King Saud University – Computer and Information Sciences. – 2022. – Vol. 34 – № 7. – pp. 3910–3933. – DOI: 10.1016/j.jksuci.2021.02.007

9. *Бачкала Б. О., Тимошин Ю. А.* Моделі забезпечення високої доступності та відмовостійкості розподілених систем цифрової інфраструктури // Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки. – 2025. – Т. 36 (75), № 4. – С. 16–24. – DOI: 10.32782/2663-5941/2025.4.2/03.

10. *Тимошин Ю. А., Бачкала Б. О.* Проблеми забезпечення відмовостійкості хмарних систем на основі оцінок метрик надійності // XIII Міжнародна науково-практична конференція з інформаційних систем та технологій: матеріали конференції (21–22 травня 2025 р.). – 2025. – С. 29–32.

11. *Щур В., Кулаков Ю.* Аналіз сучасних методів балансування навантаження в SDN мережах // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2023. – № 4 (323). – pp. 352–357. – DOI: 10.31891/2307-5732-2023-323-4-352-357

12. *Alhilali A.H., Montazerolghaem A.* Artificial intelligence-based load balancing in SDN: A comprehensive survey // Internet of Things. 2023. – Vol. 22. – № 100814. DOI: 10.1016/j.iot.2023.100814

13. *Tharwani J., Purkayastha A. A.* Cost-Performance Evaluation of General Compute Instances: AWS, Azure, GCP, and OCI // arXiv. — 2024. — arXiv:2412.03037. — DOI: 10.48550/arXiv.2412.03037

 ISSN 1560-8956