

Журавель С.В.,

orcid.org/0000-0001-6536-6298,

e-mail: serhii.zhuravel@npp.nau.edu.ua,

Журавель Н.В.,

orcid.org/0000-0001-5962-318X,

e-mail: nataliia.zhuravel@npp.nau.edu.ua

МЕТОДИ ОПТИМІЗАЦІЇ МЕРЕЖЕВОЇ АРХІТЕКТУРИ ДЛЯ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЕНЬ У ХМАРНИХ СЕРЕДОВИЩАХ

Національний авіаційний університет

Вступ

Хмарні обчислення є невід'ємною складовою сучасного інформаційного середовища, проте досягнення високої продуктивності у таких середовищах вимагає постійної оптимізації мережевої інфраструктури.

Оптимізація мережевої архітектури для високопродуктивних обчислень (*НПС*) у хмарних середовищах є актуальною та важливою проблемою сучасних обчислювальних технологій. Хмарні середовища надають доступ до значних обчислювальних ресурсів через Інтернет, і оптимізація мережі може значно покращити ефективність та продуктивність використання цих ресурсів.

Мета

Метою є аналіз та оцінка сучасних технологій, стратегій та рішень, які використовуються для оптимізації мережевої архітектури з метою підвищення продуктивності обчислень в хмарі. Розуміння цих технологічних рішень дозволить розробляти та впроваджувати ефективні стратегії оптимізації, що підвищать продуктивність та ефективність хмарних обчислень.

На сьогоднішній день існує значна кількість досліджень, присвячених оптимізації мережевої архітектури в хмарних середовищах. Багато з цих досліджень зосереджені на використанні технологій *SDN (Software-Defined Networking)* та *NFV (Network Function Virtualization)*, які дозволяють забезпечити гнучкість та ефективне управління мережею. Деякі з них також досліджують вплив віртуалізації на мережеву архітектуру та розвиток нових

методів забезпечення безпеки мережі в хмарних середовищах [1-2].

Основна частина

Хмарні обчислення, як важлива складова сучасної інформаційної технології, вимагають ефективної мережевої архітектури для забезпечення гнучкості, масштабованості та безпеки. Розглянемо основні характеристики, виклики та проблеми, а також перспективи розвитку мережевої архітектури у хмарних середовищах.

Хмарні середовища відрізняються від традиційних обчислювальних систем своєю гнучкістю, масштабованістю та доступністю. Основні характеристики мережевої архітектури в хмарних середовищах включають:

1. Мережева віртуалізація дозволяє створювати віртуальні мережі на основі фізичних інфраструктурних ресурсів, що сприяє забезпеченню гнучкості та ізоляції між користувачами.

2. Складність інфраструктури: хмарні середовища можуть включати різноманітні типи ресурсів, такі як сервери, сховища даних та мережеві пристрої, що вимагають складної мережевої архітектури для їх управління та координації.

3. Масштабованість: здатність масштабування є ключовою характеристикою хмарних середовищ, оскільки вони можуть обслуговувати велику кількість кінцевих користувачів та обробляти великі обсяги даних.

Незважаючи на переваги, існують деякі питання, які необхідно враховувати при розробці, пов'язані з мережевою архітектурою у хмарних середовищах:

1. Безпека: забезпечення безпеки даних та мережі залишається однією з найбільш актуальних питань у хмарних середовищах через потенційні загрози від зовнішніх атак.

2. Продуктивність: забезпечення достатньої швидкодії та продуктивності мережі у хмарних середовищах може бути складною задачею, особливо при обробці великих обсягів даних.

3. Управління ресурсами: ефективно управління ресурсами, включаючи обчислювальні, мережеві та сховища даних, є ключовим аспектом мережевої архітектури у хмарних середовищах.

Хоча існують потенційні виклики, перспективи розвитку мережевої архітектури у хмарних середовищах залишаються важливими та відкритими для подальшого розвитку:

- Розвиток *SDN* та *NFV*: застосування технологій *SDN* та *NFV* може покращити гнучкість та ефективність мережевої архітектури, спрощуючи управління та ресурси.

- Використання штучного інтелекту та машинного навчання: застосування методів штучного інтелекту та машинного навчання для оптимізації мережевої архітектури може допомогти у вирішенні проблем продуктивності та безпеки.

- Розширення мережі *5G*: впровадження технології *5G* може стати каталізатором для нових інновацій у мережевій архітектурі, забезпечуючи велику пропускну здатність та низьку затримку для хмарних середовищ [3-5].

Оптимізація мережевої архітектури у хмарних середовищах відіграє важливу роль у забезпеченні ефективності, продуктивності та безпеки обчислювальних сервісів. Розглянемо роль оптимізації в мережевій архітектурі та проведемо класифікацію методів оптимізації.

Оптимізація в мережевій архітектурі має на меті покращення різних аспектів функціонування мережі, включаючи ефективність використання ресурсів, мінімізацію затримок, підвищення безпеки та забезпечення високої якості обслуговування

користувачів. Оптимізація дозволяє максимально використовувати інфраструктуру хмари для задоволення потреб користувачів та підтримки бізнес-процесів.

Класифікація методів

Методи оптимізації в мережевій архітектурі можна класифікувати за різними критеріями, такими як технічні аспекти, типи мереж та принципи дії. Ось деякі з найпоширеніших класифікацій:

За призначенням:

- Оптимізація ресурсів: включає методи для ефективного розподілу обчислювальних, мережевих та ресурсів сховищ даних для максимізації їх використання.

- Оптимізація продуктивності: спрямована на покращення швидкодії, зниження затримок та оптимізацію пропускну здатності мережі.

- Оптимізація безпеки: включає методи для виявлення та захисту від загроз безпеці мережі та даних.

За принципом дії:

- Методи маршрутизації та комутації: Використовуються для вибору оптимальних маршрутів та комутації даних у мережі.

- Технології віртуалізації мережі: Включають методи для створення віртуальних мереж та ресурсів для кращого використання фізичної інфраструктури.

- Методи управління трафіком: Спрямовані на керування трафіком у мережі для забезпечення оптимальної пропускну здатності та зниження затримок.

Ці класифікації допомагають зрозуміти різноманітність методів оптимізації та їхній вплив на ефективність мережевої архітектури.

Розглянемо основні методи оптимізації мережевої архітектури у хмарних середовищах, зокрема балансування навантаження, віртуалізація мережі з використанням технологій *SDN*, *NFV*, оптимізацію локалізації даних та оптимізацію протоколів передачі даних:

1. *Балансування навантаження* є невід'ємним методом оптимізації мережевої архітектури в хмарних середовищах *НРС*. Ця важлива техніка передбачає

рівномірний розподіл робочих навантажень між декількома обчислювальними ресурсами. Основна мета полягає в тому, щоб жоден ресурс не був перевантажений завданнями, таким чином запобігаючи виникненню вузьких місць, які можуть значно сповільнити продуктивність системи. Ця стратегія не лише значно підвищує швидкість реагування та доступність додатків, але й відіграє вирішальну роль у забезпеченні безперебійної роботи користувачів та безперебійної роботи, особливо в сценаріях, де простоті можуть призвести до значних збитків.

Центральне місце в цьому методі займають алгоритми балансування навантаження, які можна класифікувати як статичні або динамічні, залежно від того, як вони розподіляють робоче навантаження. Статичні алгоритми розподіляють навантаження рівномірно, не враховуючи поточний стан або продуктивність системи. На відміну від них, динамічні алгоритми є більш просунутими, вони враховують стан системи перед розподілом навантаження, беручи до уваги такі змінні, як поточне навантаження на різні ресурси, їх продуктивність і доступність. Ці алгоритми суттєво впливають на те, як робоче навантаження розподіляється між різними мережевими ресурсами. Забезпечуючи справедливий розподіл мережевого трафіку, балансування навантаження може суттєво підвищити продуктивність *HPC*-додатків у хмарних середовищах.

2. *Віртуалізація мережі* є ще одним потужним методом оптимізації мережевої архітектури в хмарних середовищах *HPC*. Ця інноваційна концепція передбачає створення віртуальної версії мережевих ресурсів, відокремленої і незалежної від базової фізичної мережі. Таке розділення фізичного і віртуального рівнів забезпечує надзвичайну гнучкість, дозволяючи керувати мережевими ресурсами і масштабувати їх на вимогу без необхідності вносити фізичні зміни в мережеву інфраструктуру. Ця гнучкість особливо корисна в хмарних середовищах, де динамічний

характер робочих навантажень вимагає високого ступеня масштабованості.

Віртуалізація мережі дає можливість *HPC*-додаткам працювати у віртуальних мережах, які можна налаштувати для оптимальної продуктивності. Це узгоджує мережеву архітектуру з вимогами *HPC*-додатків, сприяючи підвищенню продуктивності та ефективності. Віртуалізація мережі також спрощує управління і розподіл мережевих ресурсів, зменшуючи накладні витрати і підвищуючи загальну продуктивність системи. Відокремлюючи фізичну мережу від додатків, що в ній працюють, віртуалізація мережі дозволяє більш ефективно використовувати ресурси і спрощує управління мережею, що призводить до значного підвищення продуктивності системи.

IT-інфраструктура фізично стає ближче до споживача. Компанії створюють *edge*-дата-центри безпосередньо на власних майданчиках, наприклад, на виробництвах, і використовують периферійні обчислення. Фактично обробка даних максимально наближається до джерела їх виникнення, що покращує аналітику та прийняття рішень у реальному часі.

Рішення *Edge-to-Cloud* забезпечує пришвидшення обробки даних, зростання продуктивності сервісів, спрощення *IT*-інфраструктури, а також підвищення її надійності та безпеки [13].

Виокремлюють дві моделі віртуалізації мережі – програмно-конфігуровані мережі (*SDN*) та віртуалізація мережевих функцій (*NFV*).

Концепція програмно-конфігурованих мереж та віртуалізація функцій мережі пов'язані, але спрямовані на різні аспекти. Так, *NFV* та *Deep Packet Inspection (DPI)* можуть істотно доповнювати можливості *SDN*.

По суті, програмно-конфігуровані мережі – це підхід до побудови мережевого обладнання та мережевого програмного забезпечення, що розділяє та абстрагує їх компоненти. Він передбачає розділення площини управління і площини даних одна від одної, так що рівень

управління знаходиться в одному місці, а компоненти комутації залишаються розподіленими. Контрольна площина взаємодіє на північний інтерфейс та на південний інтерфейси. У північному напрямку рівень управління забезпечує загальне, абстраговане управління мережею вищим рівням, таким як додатки та програми, які використовують його *API*. У південному напрямку рівень управління встановлює правила пересилання для рівня даних, використовуючи *API* фізичного пристрою мережевого обладнання, яке розподілене мережею.

SDN, у контексті підвищення ефективності та безпеки *НРС* у хмарному середовищі забезпечує *гнучкість та швидкість налаштування*, оскільки дозволяє швидко реагувати на зміни в мережевому трафіку та мінімізувати час налаштування та впровадження нових політик мережевого керування; *централізоване управління* забезпечує ефективне керування всією мережевою інфраструктурою з одного місця, що спрощує управління та забезпечує однорідність налаштувань; *моніторинг та аналіз мережі* – *SDN* надає можливості для збору та аналізу деталізованих даних про мережевий трафік, що дозволяє виявляти та вирішувати проблеми безпеки та продуктивності мережі в реальному часі.

У випадку *NFV* пропріетарні мережеві пристрої (наприклад, комутатори, маршрутизатори та файрволи) замінюються програмними версіями, розгорнутими на готовому обладнанні. Головний компонент – віртуальні мережеві функції (*VNF*) – віртуалізовані інстанси маршрутизаторів та інших пристроїв під управлінням гіпервізора.

Використання *NFV* забезпечує віртуалізацію мережевих функцій, таких як маршрутизація, брандмауери, *VPN* та інші, на віртуальних платформах. Це дозволяє розгортати та керувати цими функціями як віртуальними машинами (*VM*) або контейнерами, що спрощує управління мережею та зменшує витрати на обладнання; надає можливість швидкого масштабування

мережевих функцій відповідно до потреб обчислювальних ресурсів. Це дозволяє ефективно використовувати ресурси та забезпечує гнучкість в управлінні мережевим трафіком; дозволяє централізовано керувати та оркеструвати віртуальними мережевими функціями з використанням оркестратора. Це дозволяє ефективно управляти мережею та автоматизувати процеси розгортання та керування; за допомогою *NFV* можна швидко розгортати та використовувати нові мережеві функції та послуги, що дозволяє підприємствам швидко реагувати на зміни в потребах користувачів та ринку.

3. *Оптимізація локалізації даних* – це стратегія, яка відіграє ключову роль у підвищенні продуктивності *НРС*-додатків у хмарних середовищах. Цей метод фокусується на мінімізації переміщення даних в мережі за рахунок використання близькості даних. У *НРС*-додатках значна частина часу витрачається на передачу даних від одного вузла до іншого. Цей рух даних може спричинити значні накладні витрати, які можуть негативно вплинути на загальну продуктивність системи.

Оптимізуючи локалізацію даних, час передачі даних можна суттєво покращити загальну продуктивність *НРС*-додатків. Для оптимізації розташування даних у хмарних середовищах широко використовуються такі методи, як планування з урахуванням даних, що передбачає планування завдань на основі розташування необхідних їм даних, і стратегії розміщення даних, які передбачають розміщення даних таким чином, щоб мінімізувати потребу в їх переміщенні. Зберігаючи дані ближче до обчислювальних ресурсів, оптимізація розташування даних зменшує час, що витрачається на передачу даних, тим самим підвищуючи загальну продуктивність системи.

4. *Стратегія оптимізації протоколів передачі даних* має важливе значення в мережевій архітектурі для підвищення ефективності та *НРС* у хмарному середовищі. Вона сприяє збільшенню швидкості передачі даних між обчислювальними

вузлами у хмарному середовищі. Також дозволяє зменшити затримки передачі даних (латентність), що в свою чергу підвищує реактивність та швидкодію системи обчислень. Це особливо важливо для *НПС*-додатків, які вимагають миттєвої відповіді на запити користувачів. Оптимізація протоколів може включати в себе механізми для забезпечення надійності передачі даних, такі як контроль цілісності передачі даних та механізми виявлення та виправлення помилок, що, в свою чергу, допомагає запобігти втраті даних та забезпечує безпеку обчислень.

Таким чином, оптимізація мережевої архітектури є фундаментальним аспектом досягнення високої продуктивності *НПС*-додатків у хмарних середовищах. Для повного використання потенціалу високопродуктивних обчислень у хмарі, впровадження таких стратегій, як балансування навантаження, віртуалізація мережі та оптимізація розташування даних, оптимізація протоколів має першорядне значення. Ці стратегії можуть значно підвищити ефективність, швидкість і надійність цих додатків, сприяючи створенню більш надійного і динамічного обчислювального середовища. Оскільки попит на хмарні обчислення продовжує зростати, ці методи оптимізації будуть продовжувати розвиватися і вдосконалюватися, прокладаючи шлях до більш досконалих, ефективних і високопродуктивних хмарних додатків у майбутньому [3, 6-12].

Висновки

Досліджено методи оптимізації мережевої архітектури для високопродуктивних обчислень у хмарних середовищах. В результаті проведених досліджень можна зробити наступні висновки:

- Розглянуті методи оптимізації є ефективними у покращенні продуктивності та безпеки мережевої архітектури у хмарних середовищах.
- Дослідження показали, що впровадження даних методів може призвести до значних покращень у продуктивності мережі, зниженні затримок та оптимізації використання ресурсів.

- Результати можуть бути використані для вдосконалення мережевих інфраструктур у хмарних середовищах для забезпечити кращої продуктивності та безпеки обчислень. Також дозволяють глибше зрозуміти вплив різних методів оптимізації на мережеву архітектуру та відкривають нові можливості для подальших досліджень у цій області.

В перспективі подальших досліджень можна сконцентруватися на таких аспектах:

- Розробка нових методів оптимізації, які враховують специфічні потреби різних типів додатків та вимоги їхньої продуктивності та безпеки.
- Вивчення впливу розширення хмарних середовищ, таких як розподілені додатки та Інтернет речей (*IoT*), на ефективність мережевої архітектури та розробка відповідних методів оптимізації.
- Дослідження інтеграції нових технологій, таких як штучний інтелект та блокчейн, у мережеву архітектуру для досягнення ще більшої продуктивності та безпеки.

Ці напрямки досліджень можуть сприяти подальшому розвитку та вдосконаленню мережевої архітектури у хмарних середовищах.

Література

1. Cloud Computing: Principles and Paradigms / ed. by R. Buyya, J. Broberg, A. Goscinski. Hoboken : John Wiley & Sons Inc, 2011. 664 p.
2. Zhang Y. Network Function Virtualization: Concepts and Applicability in 5G Networks. Hoboken : Wiley-IEEE Press, 2018. 192 p.
3. Lee G. Cloud Networking: Understanding Cloud-Based Data Center Networks. Waltham : Morgan Kaufmann, 2014. 238 p.
4. Cloud Computing: Principles, Systems and Applications / ed. by N. Antonopoulos, L. Gillam. London : Springer, 2010. 382 p.
5. Mohindroo S. K. Unraveling the Power of Software-Defined Networking (SDN): Past, Present, and Future. URL: <https://medium.com/@sanjay.mohindroo66/>

unraveling-the-power-of-software-defined-networking-sdn-past-present-and-future-29480ff59c1.

6. Chao L. Cloud Computing Networking: Theory, Practice, and Development. Boca Raton : Auerbach Publications, 2015. 528 p.

7. Палагін В. В., Євтушенко І. О. Віртуалізація як середовище реалізації мережевих функцій. *Вісник Черкаського державного технологічного університету*. 2021. № 3. С. 31–38.

8. Cisco, Software-Defined Networking. URL: <https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html>.

9. Коробейнікова Т. І., Калько Т. С., Лужецька Н. М. Розгляд архітектури програмно-керованих мереж. *Міжнародний науковий журнал «Грааль науки»*. 2023. Вип. 28. С. 228–237.

10. SD-LAN vs LAN – What Are The Key Differences? URL: <https://www.extremenetworks.com/resources/blogs/sd-lan-vs-lan-what-are-the-key-differences>.

11. Гніденко М. П., Вишнівський В. В., Ільїн О. О. Побудова SDN мереж : навч. посібник. Київ : ДУТ, 2019. 190 с.

12. Жаріков Е. В. Інформаційна технологія управління IT-інфраструктурою хмарного центру оброблення даних: дис. ... д-ра техн. наук : 05.13.06. Київ, 2020. 402 с.

13. Бардадим Т., Лефтеров О., Осипенко С. Досвід розгортання тестів OpenStack та порівняння віртуальних і реальних кластерних середовищ. *Кібернетика та комп'ютерні технології*. 2021. Вип. 3. С. 74–85.

14. Лозінський А. П. Синтез технологій платформ хмарних обчислень. *Системи керування та комп'ютери*. 2019. № 6. С. 35–45.

Журавель С.В., Журавель Н.В.

МЕТОДИ ОПТИМІЗАЦІЇ МЕРЕЖЕВОЇ АРХІТЕКТУРИ ДЛЯ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЕНЬ У ХМАРНИХ СЕРЕДОВИЩАХ

У статті проведено огляд різних методів оптимізації мережевої архітектури в контексті високопродуктивних обчислень у хмарних середовищах. В основу дослідження покладено аналіз сучасних технологій, стратегій та рішень, які використовуються для підвищення продуктивності обчислень в хмарі. Особлива увага приділяється питанням мережевої архітектури, включаючи її структуру, протоколи, сервіси та оптимізацію для досягнення максимальної продуктивності. Викладена інформація може бути корисною для науковців, інженерів та розробників програмного забезпечення, які працюють над створенням і вдосконаленням хмарних обчислювальних систем.

Ключові слова: оптимізація; мережева архітектура; високопродуктивні обчислення; хмарні середовища.

Zhuravel S.V., Zhuravel N.V.

METHODS OF OPTIMIZING NETWORK ARCHITECTURE FOR HIGH-PERFORMANCE COMPUTING IN CLOUD ENVIRONMENTS

The article provides a review of various methods for optimizing network architecture in the context of high-performance computing in cloud environments. The research is based on an analysis of modern technologies, strategies, and solutions used to enhance computational performance in the cloud. Special attention is given to networking architecture issues, including its structure, protocols, services, and optimization to achieve maximum productivity. The information presented may be useful for scientists, engineers, and software developers working on the creation and improvement of cloud computing systems.

Keywords: optimization; network architecture; high-performance computing; cloud environments.