

В. В. ВЕРБІВСЬКИЙ, студент, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: vvverbavlad@gmail.com; ORCID: <https://orcid.org/0009-0007-2261-8126>

В. Ю. ВОЛОВЩИКОВ, кандидат технічних наук, доцент, доцент кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: Valeriy.Volovshchikov@khpi.edu.ua; ORCID: <https://orcid.org/0000-0003-4454-2314>

В. Ф. ШАПО, кандидат технічних наук, доцент, професор кафедри озброєння, Інститут Військово-морських сил Національного університету «Одеська морська академія», м. Одеса, Україна; e-mail: vladlen.shapo@gmail.com; ORCID: <https://orcid.org/0000-0002-3921-4159>

М. В. ЛЕВІНСЬКИЙ, кандидат технічних наук, доцент, доцент кафедри теорії автоматичного управління та обчислювальної техніки, Національний університет «Одеська морська академія», м. Одеса, Україна; e-mail: MaxLevinskyi@te.net.ua; ORCID: <https://orcid.org/0000-0002-6544-5110>

В. М. ЛЕВІНСЬКИЙ, кандидат технічних наук, доцент, доцент кафедри автоматизації технологічних процесів та робототехнічних систем, Одеський національний технологічний університет, м. Одеса, Україна; e-mail: ValeryLevinskyi@gmail.com; ORCID: <https://orcid.org/0000-0002-3563-528X>

АДАПТИВНА ДИНАМІЧНА ОПТИМІЗАЦІЯ РЕСУРСІВ В СИСТЕМАХ З MULTI-TENANCY АРХІТЕКТУРОЮ

У статті розглядається проблема ефективного розподілу обчислювальних ресурсів у хмарних програмних системах, які побудовані за принципом багатоорендарної архітектури. Такий підхід дозволяє одночасно обслуговувати декількох користувачів в рамках єдиного програмного екземпляра із забезпеченням ізолюваності їхніх даних та конфігурацій. Такий підхід знижує витрати на інфраструктуру та спрощує обслуговування, але спільне використання ресурсів створює нові виклики, пов'язані з нерівномірним навантаженням і потенційним перевантаженням окремих компонентів системи. У межах дослідження проаналізовано класичні підходи до розподілу з'єднань з базою даних між користувачами – як статичні, що фіксують обмеження наперед, так і базові динамічні, які враховують лише поточну кількість запитів. Виявлено обмеження цих методів в умовах змінного та нерівномірного навантаження. Запропоновано нову адаптивну методику динамічної оптимізації ресурсів, яка враховує не тільки інтенсивність запитів, але й середній час їх обробки, історичні показники активності та індивідуальні характеристики кожного користувача. Методика також дозволяє враховувати ваговий коефіцієнт, що визначає вплив кожного чинника на кінцевий розрахунок. Експериментальна перевірка моделі на основі трьох сценаріїв із різною інтенсивністю запитів показала значне зниження середнього часу відповіді до 20 % порівняно з базовим методом, без збільшення загальної кількості використовуваних підключень. Результати свідчать про ефективність запропонованого підходу в реальних умовах. Така методика може бути впроваджена у сучасні хмарні платформи для підвищення продуктивності та стійкості до пікових навантажень інфраструктури.

Ключові слова: хмарні обчислення, багатоорендарна архітектура, динамічна оптимізація ресурсів, база даних, пул підключень, методика адаптивного розподілу ресурсів.

Вступ. Сучасні хмарні рішення та сервіси за моделлю SaaS нерідко ґрунтуються на multi-tenancy архітектурі. У таких системах один екземпляр застосунку може одночасно обслуговувати декількох орендарів, які користуються спільною інфраструктурою, зберігаючи при цьому ізолюваність своїх даних та конфігурацій. Подібний підхід дає змогу зменшувати витрати на інфраструктуру та спрощує процеси оновлення та підтримки [1–3].

Рис. 1 ілюструє порівняння вартості інфраструктури в системах з single-tenancy (STA) та multi-tenancy архітектурою (МТА) [4]. Відповідно до рис. 1 використання МТА зменшує вартість витрат на загальну інфраструктуру системи [4].

В ситуаціях коли різні орендарі ділять між собою обчислювальні ресурси, виникають проблеми з нерівномірним навантаженням і високим ризиком перевантажень окремих ресурсів. У таких випадках динамічна оптимізація дозволяє системі перерозподіляти ресурси, залежно від поточних показників активності та потреб орендарів [5]. Це дає змогу: гнучко адаптувати розподіл ресурсів до поточного стану системи, зменшувати кількість відмов через перевантаження окремих вузлів,

запобігти ситуаціям, коли один орендар використовує більшу частину ресурсів.

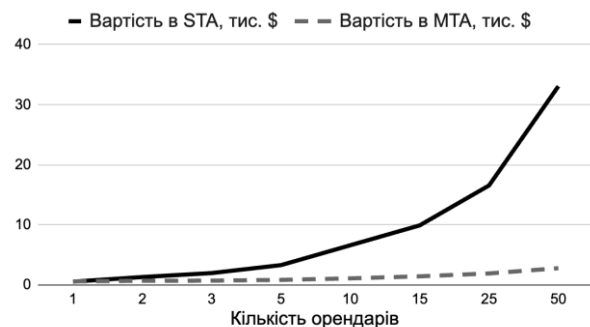


Рис. 1. Вартість витрат на інфраструктуру

Таким чином розробка методів і алгоритмів динамічного розподілу ресурсів є одним із ключових напрямів для забезпечення ефективності multi-tenancy систем. Саме тому використання ефективних методів динамічної оптимізації ресурсів є актуальним і важливим [6].

Дослідження динамічної оптимізації ресурсів. У комп'ютерних системах розподіл ресурсів може



здійснюватися статично або динамічно. Статичний розподіл ресурсів виконується на етапі компіляції або початкового налаштування системи. У цьому випадку ресурси, наприклад, обсяг пам'яті або кількість потоків, закріплюються за конкретними задачами наперед і не змінюються під час виконання. Такий підхід простий у реалізації, однак він не враховує змінні умови навантаження, що може призводити до неефективного використання ресурсів, одні компоненти можуть простоювати, тоді як інші відчувають нестачу [7, 8].

На противагу цьому динамічна оптимізація ресурсів у комп'ютерних науках означає процес ефективного розподілу ресурсів, залежно від поточного стану системи та завантаження обчислювальних елементів. У рамках динамічної оптимізації система постійно відслідковує використання ресурсів (пам'яті, процесорного часу, мережевих ресурсів тощо), аналізує отримані дані та автоматично адаптує їх розподіл. Основна мета – забезпечити максимальну продуктивність, зменшити затримки та запобігти перевантаженню системи [7–10].

У контексті МТА, система повинна підтримувати одночасну роботу великої кількості орендарів, кожен з яких взаємодіє з застосунком як з окремим логічним екземпляром, з окремим доступом до своїх даних. Тому особливу увагу в системі з МТА слід приділити саме оптимізації з'єднань з базою даних (БД), так як це те на що напряму впливає кількість орендарів, які використовують застосунок. Одним з можливих методів оптимізації при цьому є об'єднання підключень в динамічні пули. Базові підходи до керування підключеннями до БД за допомогою пулів підключень можуть забезпечити значний приріст у швидкодії в МТА [11].

Рис. 2 ілюструє приклад кореляції кількості підключень орендаря та поточної кількості з'єднань з використанням динамічних пулів підключень.

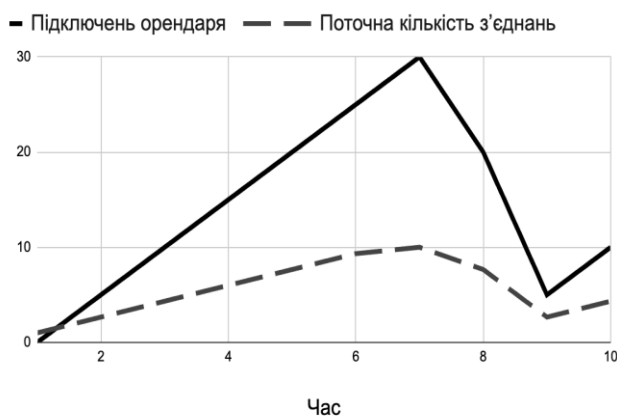


Рис. 2. Динамічна оптимізація в залежності від кількості з'єднань

Для реалізації динамічних пулів підключень потрібно мати алгоритм визначення поточного ліміту підключень.

Відповідно до (1) ліміт підключень R_i для орендаря i визначається як сума базового ліміту $R_{\min}$ та динамічно розрахованого значення, що залежить від кількості запитів N_i , що є характеристикою орендаря. При

цьому $R_{\max}$ – це максимальна кількість можливих підключень, а S це кількість всіх орендарів [12, 13]:

$$R_i = R_{\min} + \frac{N_i}{\sum_{j=1}^S N_j} \cdot (R_{\max} - S \cdot R_{\min}). \quad (1)$$

Для вимірювання ефективності алгоритму оптимізації прийнято використовувати дві метрики: покращення середнього часу відповіді та ефективності використання ресурсів [12].

Для обчислення середнього часу відповіді системи можливо провести тестовий сценарій протягом часу T для статичної кількості підключень та динамічної кількості підключень. Формально, якщо позначити середній час відповіді при динамічній оптимізації як L_{dyn} та при статичному ліміті підключень як L_{stat} , і провести тестовий сценарій навантаження на систему в проміжку часу від 1 до P , то в результаті вимірювання можна отримати два набори показників часу відповідей для статичного (2) та динамічного (3) пулу підключень відповідно:

$$\{L_{stat}(1), L_{stat}(2), \dots, L_{stat}(P)\}, \quad (2)$$

$$\{L_{dyn}(1), L_{dyn}(2), \dots, L_{dyn}(P)\}. \quad (3)$$

Обчислюючи коефіцієнт середнього часу $\Delta L(p)$ відповідно до (4), та виконуючи аналіз, можна дійти до висновку, що покращення яке більше за 0, вказує на позитивний вплив динамічної оптимізації підключень, а чим ближче це число до 1, тим ефективніше працює алгоритм:

$$\Delta L(p) = \frac{L_{stat}(p) - L_{dyn}(p)}{L_{stat}(p)}. \quad (4)$$

Коефіцієнт ефективності використання ресурсів E може слугувати додатковою метрикою, яка зазвичай має причинно-наслідковий зв'язок з середнім часом відповіді. Так при погіршенні цього коефіцієнту буде погіршуватись середній час відповіді.

Для розрахунку коефіцієнта ефективності використання ресурсів відповідно до (5) потрібно виміряти кількість використовуваних динамічних підключень U_{dyn}^t та статичних підключень U_{stat}^t в момент часу t за часовий інтервал T :

$$E = \frac{\sum_t U_{dyn}^t}{\sum_t U_{stat}^t}. \quad (5)$$

Ефективний алгоритм повинен мати такий коефіцієнт менше одиниці, а чим ближче дане число до нуля, тим краще алгоритм динамічної оптимізації.

Таким чином, динамічна оптимізація ресурсів дозволяє коригувати ліміти підключень до БД для кожного орендаря в залежності від їхньої активності, забезпечуючи ефективне використання пам'яті та підвищуючи загальну ефективність системи.

Постановка задачі. Класичні підходи до обмеження кількості з'єднань на орендаря ґрунтуються або на статичному розподілі, або на спрощених формулах, що враховують лише кількість запитів. Проте в реальних умовах навантаження може суттєво коливатися в часі та між різними орендарями, що призводить до неефективного використання ресурсів та затримок у обробці запитів.

Метою даного дослідження є розробка методики динамічного розподілу ліміту підключень до БД для МТА систем та практичне її дослідження. Завдання полягає у тому, щоб враховувати індивідуальні характеристики орендарів для більш точного та гнучкого керування ресурсами – такі як інтенсивність запитів, середній час обробки та історичні дані навантаження.

Таким чином, основні задачі, які ставляться у межах роботи: розробка методики розрахунку ліміту підключень, яка враховувала би не лише поточну кількість запитів, а й середній час обробки та історичні дані активності. З метою перевірки працездатності розробленої методики в роботі виконуються практичні дослідження шляхом збирання метрик для різних сценаріїв навантаження системи та аналізу результатів вимірювань.

Очікуваним результатом може стати доведення доцільності впровадження адаптивного розподілу підключень у складних багатокористувацьких системах для підвищення продуктивності без збільшення загального навантаження на інфраструктуру.

Методика адаптивного динамічного розподілу ресурсів. Для кращого розподілу ресурсів пропонується врахувати більшу кількість характеристик орендаря за якими буде можливо прогнозувати навантаження. Для цього можна ввести в розгляд наступні характеристики: історичні дані по використанню з'єднань, середній час обробки з'єднань та пріоритетність характеристики.

Історичні дані по використанню з'єднань повинні містити інформацію про $W_i(t)$ – середню кількість запитів на одну секунду за конкретний період часу t .

В такому підході присутні певні недоліки, а саме те, що присутня значна можливість відсутності даних за розрахунковий період, а також дані за попередній період можуть бути не релевантні, наприклад, через сезони події. Для вирішення першого недоліку необхідно обирати достатньо широкий період розрахунку, а також розробити алгоритм дії при відсутності історичних даних у орендаря. Для вирішення другого недоліку необхідно обирати період який буде схожий на попередній.

При введенні нульового значення середньої кількості запитів за розрахунковий період відповідно до (6) пропонується обрати наступний алгоритм дій, а саме, при відсутності значення – обирати початкове значення $W_{\min}$, яке буде задаватись в налаштуваннях системи:

$$W_i = \max \{W_{\min}, W_i\}. \quad (6)$$

Для середнього часу обробки визначимо A_i , який буде розраховуватись при кожному запиті орендаря.

Для урівноваження пріоритетності середнього часу відповіді по відношенню до кількості запитів орендаря введемо новий коефіцієнт β , який пропонується встановлювати в налаштуваннях системи. За допомогою β пропонується вирішувати питання щодо можливості корегування значущості характеристики системи, тобто це визначає на які дані слід більше спиратися для конкретної системи – час відповіді або кількість підключень.

З урахуванням вищевикладеного введемо у розгляд (7) – формулу розрахунку ліміту доступних підключень до БД для конкретного орендаря:

$$R_i = R_{\min} + \frac{W_i (N_i + \beta \cdot A_i)}{\sum_{j=1}^S W_j (N_j + \beta \cdot A_j)} \cdot (R_{\max} - S \cdot R_{\min}). \quad (7)$$

Практичне дослідження методики адаптивного динамічного розподілу ресурсів. Для перевірки запропонованої методики динамічного розподілу ресурсів на прикладі ліміту підключень, проведена низка тестувань на імітованому середовищі, що моделює поведінку МТА системи із різною інтенсивністю запитів від кожного орендаря.

Метою експерименту була перевірка доцільності застосування розробленої методики, а саме чи забезпечує методика більш ефективний розподіл ресурсів в умовах нерівномірного навантаження, а також порівняння її ефективності з базовим варіантом розподілу ліміту підключень.

Для умов тестування обрано три різних сценарії, які відрізнялися кількістю одночасно активних орендарів та характером навантаження, а саме: в першому сценарії усі орендарі мають однакову кількість запитів з однаковим середнім часом обробки; в другому сценарії один з орендарів генерує вдвічі більше запитів, ніж інші; в третьому сценарії орендарі мають різну інтенсивність запитів, а також різний середній час обробки запитів.

Для кожного сценарію були проведені серії тестів у трьох режимах: з використанням статичного розподілу ліміту підключень, з використанням базової формули розрахунку ліміту підключень (1), з використанням адаптивної формули (7).

Для оцінки було використано дві метрики: середній час відповіді та коефіцієнт ефективності використання підключень.

Результат тестування середнього часу відповіді наведено в табл. 1 та табл. 2.

Аналізуючи результати, наведені в табл. 1 та табл. 2, можна зробити висновок що має місце покращення часу відповіді в порівнянні з базовою формулою у сценарії 2 та 3, де навантаження не є рівномірним.

Таблиця 1 – Середній час відповіді

Сценарій	Середній час відповіді, мс		
	Статичний розподіл	Базова формула	Адаптивна формула
1	4033	3050	3063
2	5792	4012	3907
3	9611	5521	4701

Таблиця 2 – Коефіцієнт середнього часу відповіді

Сценарій	$\Delta L(p)$	
	Базова формула	Адаптивна формула
1	0,76	0,76
2	0,69	0,67
3	0,57	0,49

Результат тестування кількості підключень до БД наведений в табл. 3 та табл. 4.

Таблиця 3 – Кількість підключень до БД, що використовується

Сценарій	Кількість підключень		
	Статичний розподіл	Базова формула	Адаптивна формула
1	30	15	15
2	30	22	22
3	30	28	27

Таблиця 4 – Коефіцієнт ефективності використання ресурсів

Сценарій	E	
	Базова формула	Адаптивна формула
1	0,50	0,50
2	0,73	0,73
3	0,93	0,9

Досліджуючи результати, наведені в табл. 3 та табл. 4, можна зробити висновок що загальна кількість використовуваних підключень до БД залишається такою самою як з використанням (1) так і (7).

Отже використання (7) позитивно вплинуло на середній час обробки кожного запиту орендаря в складних умовах та не здійснило вплив на загальну кількість підключень до БД порівняно з (1).

Висновки. В роботі було проведено дослідження проблеми динамічного розподілу підключень до БД у МТА системах. На основі аналізу існуючих підходів було запропоновано методику, яка враховує не лише поточну кількість запитів від орендаря, але й історичні показники навантаження та середній час обробки запитів. Такий підхід дозволяє більш точно оцінювати реальні потреби кожного клієнта та адаптувати розподіл ресурсів відповідно до поточної ситуації.

Практичне тестування адаптивної методики показало її ефективність у складних сценаріях із нерівномірним навантаженням – у таких умовах адаптивна методика дозволила зменшити середній час відповіді на понад 20 % у порівнянні з (1). При цьому загальна кількість активних з'єднань залишалась на рівні, аналогічному до базового підходу, що свідчить про ефективніше використання наявних ресурсів.

Таким чином, запропонований підхід може бути використаний для побудови кращої системи розподілу ресурсів у хмарних платформах, орієнтованих на надання SaaS послуг з використанням МТА.

Список використаної літератури

1. Jaap Kabbedijk *Defining Multi-Tenancy: A Systematic Mapping Study on the Academic and the Industrial Perspective*. URL:

<https://www.sciencedirect.com/science/article/abs/pii/S0164121214002313/> (дата звернення: 12.02.2025).

2. Hugo S. C. Pinto *A Systematic Mapping Study on the Multi-tenant Architecture of SaaS Systems*. URL: https://www.researchgate.net/publication/306255127_A_Systematic_Mapping_Study_on_the_Multi-tenant_Architecture_of_SaaS_Systems/ (дата звернення: 11.03.2025).
3. Rouven Krebs *Architectural Concerns in Multi-Tenant SaaS Applications*. URL: https://www.researchgate.net/publication/264942141_Architectural_Concerns_in_Multi-tenant_SaaS_Applications/ (дата звернення: 18.12.2024).
4. Single vs. Multi Tenant Cost Comparison. URL: <https://www.slideshare.net/slideshow/single-vs-multi-tenant-cost-comparison/35649260/> (дата звернення: 09.02.2025).
5. *Optimizing Multi-Tenant SaaS Applications for Performance*. URL: <https://divami.com/news/optimizing-multi-tenant-saas-applications-for-performance/> (дата звернення: 20.02.2025).
6. *What is Multi-Tenant Data Management and Why do you need it?* URL: <https://medium.com/%40shenli3514/what-is-multi-tenant-data-management-and-why-do-you-need-it-1-b424b81c0498/> (дата звернення: 12.03.2025).
7. Olivier Beaumont *Comparison of Static and Dynamic Resource Allocation Strategies for Matrix Multiplication*. URL: <https://inria.hal.science/hal-01163936/document/> (дата звернення: 09.03.2025).
8. Radu Prodan *Comparison of Static and Dynamic Resource Allocations for Massively Multiplayer Online Games on Unreliable Resources*. URL: https://link.springer.com/chapter/10.1007/978-3-319-14325-5_26/ (дата звернення: 12.03.2025).
9. Ragini Karwayun *Static and Dynamic Resource Allocation Strategies in High Performance Heterogeneous Computing Application*. URL: <https://www.ijarcs.info/index.php/Ijarcs/article/view/5265/> (дата звернення: 12.03.2025).
10. Cameron A. MacKenzie *Static and Dynamic Resource Allocation Models for Recovery of Interdependent Systems: Application to the Deepwater Horizon Oil Spill*. URL: <https://www.imse.iastate.edu/files/2016/01/MacKenzie-et-al-Static-and-Dynamic-Resource-Allocation-Models-for-Recovery-of-Interdependent-Systems-Accepted.pdf/> (дата звернення: 15.03.2025).
11. *Multiplication Multi-Tenant Database Design: Single vs. Multiple DBs & The Role of a Master Database*. URL: <https://smit90.medium.com/multi-tenant-database-design-single-vs-multiple-dbs-the-role-of-a-master-database-%EF%B8%8F-67bd6792541b/> (дата звернення: 16.03.2025).
12. Jia-You Lin *Dynamic Resource Allocation for Network Slicing with Multi-Tenants in 5G Two-Tier Networks*. URL: <https://www.mdpi.com/1424-8220/23/10/4698/> (дата звернення: 16.03.2025).
13. *Dominant Resource Fairness: Fair Allocation of Multiple Resource Types*. URL: <https://www.usenix.org/conference/nsdi11/dominant-resource-fairness-fair-allocation-multiple-resource-types/> (дата звернення: 21.03.2025).

References (transliterated)

1. Jaap Kabbedijk *Defining Multi-Tenancy: A Systematic Mapping Study on the Academic and the Industrial Perspective*. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0164121214002313/> (accessed: 12.02.2025).
2. Hugo S. C. Pinto *A Systematic Mapping Study on the Multi-tenant Architecture of SaaS Systems*. Available at: https://www.researchgate.net/publication/306255127_A_Systematic_Mapping_Study_on_the_Multi-tenant_Architecture_of_SaaS_Systems/ (accessed: 11.03.2025).
3. Rouven Krebs *Architectural Concerns in Multi-Tenant SaaS Applications*. Available at: https://www.researchgate.net/publication/264942141_Architectural_Concerns_in_Multi-tenant_SaaS_Applications/ (accessed: 18.12.2024).
4. Single vs. Multi Tenant Cost Comparison. Available at: <https://www.slideshare.net/slideshow/single-vs-multi-tenant-cost-comparison/35649260/> (accessed: 09.02.2025).
5. *Optimizing Multi-Tenant SaaS Applications for Performance*. Available at: <https://divami.com/news/optimizing-multi-tenant-saas-applications-for-performance/> (accessed: 20.02.2025).
6. *What is Multi-Tenant Data Management and Why do you need it?* Available at: <https://medium.com/%40shenli3514/what-is-multi-tenant-data-management-and-why-do-you-need-it-1-b424b81c0498/> (дата звернення: 12.03.2025).

- tenant-data -management-and-why-do-you-need-it-1-b424b81c0498/ (accessed: 12.03.2025).
7. Olivier Beaumont *Comparison of Static and Dynamic Resource Allocation Strategies for Matrix Multiplication*. Available at: <https://inria.hal.science/hal-01163936/document/> (accessed: 09.03.2025).
 8. Radu Prodan *Comparison of Static and Dynamic Resource Allocations for Massively Multiplayer Online Games on Unreliable Resources*. Available at: https://link.springer.com/chapter/10.1007/978-3-319-14325-5_26/ (accessed: 12.03.2025).
 9. Ragini Karwayun *Static and Dynamic Resource Allocation Strategies in High Performance Heterogeneous Computing Application*. Available at: <https://www.ijarcs.info/index.php/Ijarcs/article/view/5265/> (accessed: 12.03.2025).
 10. Cameron A. MacKenzie *Static and Dynamic Resource Allocation Models for Recovery of Interdependent Systems: Application to the Deepwater Horizon Oil Spill*. Available at: <https://www.imse.iastate.edu/files/2016/01/MacKenzie-et-al-Static-and-Dynamic-Resource-Allocation-Models-for-Recovery-of-Interdependent-Systems-Accepted.pdf/> (accessed: 15.03.2025).
 11. *Multiplication Multi-Tenant Database Design: Single vs. Multiple DBs & The Role of a Master Database*. Available at: <https://smit90.medium.com/multi-tenant-database-design-single-vs-multiple-dbs-the-role-of-a-master-database-%EF%B8%8F-67bd6792541b/> (accessed: 16.03.2025).
 12. Jia-You Lin *Dynamic Resource Allocation for Network Slicing with Multi-Tenants in 5G Two-Tier Networks*. Available at: <https://www.mdpi.com/1424-8220/23/10/4698/> (accessed: 16.03.2025).
 13. *Dominant Resource Fairness: Fair Allocation of Multiple Resource Types*. Available at: <https://www.usenix.org/conference/nsdi11/dominant-resource-fairness-fair-allocation-multiple-resource-types/> (accessed: 21.03.2025).

Надійшла (received) 15.04.2025

UDC 004.42

V. V. VERBIVSKYI, National Technical University "Kharkiv Polytechnic Institute", Student, Kharkiv, Ukraine; e-mail: vverbavlad@gmail.com; ORCID: <https://orcid.org/0009-0007-2261-8126>

V. Y. VOLOVSHCHYKOV, Candidate of Technical Sciences, Docent, National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine; e-mail: Valeriy.Volovshchikov@kpi.edu.ua; ORCID: <https://orcid.org/0000-0003-4454-2314>

V. F. SHAPO, Candidate of Technical Sciences, Docent, Naval Institute of the National University "Odesa Maritime Academy", Professor at the Weaponry Department, Odessa, Ukraine; e-mail: vladlen.shapo@gmail.com; ORCID: <https://orcid.org/0000-0002-3921-4159>

M. V. LEVINSKYI, Candidate of Technical Sciences, Docent, National University "Odesa Maritime Academy", Associate Professor at the Department of Theory of Automatic Control and Computing Machinery, Odessa, Ukraine; e-mail: MaxLevinskyi@te.net.ua; ORCID: <https://orcid.org/0000-0002-6544-5110>

V. M. LEVINSKYI, Candidate of Technical Sciences, Docent, Odesa National University of Technology, Associate Professor at the Department of Automation of Technological Processes and Robotic Systems, Odessa, Ukraine; e-mail: ValeryLevinskyi@gmail.com; <https://orcid.org/0000-0002-3563-528X>

ADAPTIVE DYNAMIC RESOURCE ALLOCATION IN SYSTEMS WITH MULTI-TENANCY ARCHITECTURE

The article considers the problem of efficient allocation of computing resources in cloud software systems based on the principle of multi-tenant architecture. This approach allows to simultaneously serve several users within a single software instance while ensuring the isolation of their data and configurations. This approach reduces infrastructure costs and simplifies maintenance, but sharing resources creates new challenges associated with uneven load and potential overload of individual system components. The study analyzes classical approaches to distributing database connections among users, both static, which fix the restrictions in advance, and basic dynamic, which consider only the current number of requests. The limitations of these methods under conditions of variable and uneven load are revealed. A new adaptive methodology for dynamic resource optimization is proposed, which considers not only the intensity of requests but also the average processing time, historical activity indicators, and individual characteristics of each user. The methodology also allows considering the weighting factor that determines the impact of each factor on the final calculation. Experimental verification of the model based on three scenarios with different request intensities showed a significant reduction in the average response time by up to 20 % compared to the baseline method, without increasing the total number of connections used. The results demonstrate the effectiveness of the proposed approach in real-world conditions. This methodology can be implemented in modern cloud platforms to improve performance, peak load resilience, and rational use of infrastructure resources.

Keywords: cloud computing, multi-tenant architecture, dynamic resource optimization, database, connection pool, adaptive resource allocation technique.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Вербівський Владислав Валерійович / Verbivskyi Vladyslav Valeriiovych

Автор 2 / Author 2: Воловщиків Валерій Юрійович / Volovshchikov Valeriy Yuriyovich

Автор 3 / Author 3: Шапо Владлен Феліксівич / Shapo Vladlen Feliksovych

Автор 4 / Author 4: Левінський Валерій Михайлович / Levinskyi Valeriy Mihailovich

Автор 5 / Author 5: Левінський Максим Валерійович / Levinskyi Maksym Valeriyovich