

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-60-48>

УДК: 004.382.2:004.41:004.4'2

Нерознак Євгеній Ігорович, PhD, ст. викл.

<https://orcid.org/0000-0001-5641-5473>

Кондрусь Андрій Володимирович, ст. викл.

<https://orcid.org/0000-0001-8815-6517>

Троцько Олександр Олександрович, к.т.н., доцент

<https://orcid.org/0000-0001-7535-5023>

Усик Анна Андріївна, викладач

<https://orcid.org/0009-0009-1355-5826>

Костів Орест Любомирович, викладач

<https://orcid.org/0009-0005-2241-5754>

Військовий інститут телекомунікацій та інформатизації імені Героїв Крут, м. Київ, Україна

АНАЛІЗ СУЧАСНОГО СТАНУ ФУНКЦІОНУВАННЯ СЕРВЕРНИХ КЛАСТЕРІВ ІНФОРМАЦІЙНИХ СИСТЕМ ТА МЕТОДІВ УПРАВЛІННЯ НАВАНТАЖЕННЯМ

Нерознак Є. І., Кондрусь А. В., Троцько О. О., Усик А. А., Костів О. Л. Аналіз сучасного стану функціонування серверних кластерів інформаційних систем та методів управління навантаженням. У статті здійснено аналіз сучасного стану функціонування серверної інфраструктури інформаційних систем та методів управління навантаженням на серверні кластери. Виявлено основні чинники, що впливають на ефективність роботи: зростання кількості та складності клієнтських запитів, обмеженість апаратних ресурсів, неможливість масштабування інфраструктури, а також недостатня продуктивність існуючих методів балансування навантаження. Встановлено, що більшість сучасних алгоритмів мають високу обчислювальну складність, низьку стійкість до збоїв і значну залежність від якості навчальних даних. У межах дослідження розглянуто класичні та інтелектуальні підходи до оптимального використання ресурсів кластерних систем. Найбільш ефективними визнано адаптивні методи, засновані на теорії ігор, зокрема рівновазі Неша. Цей підхід дозволяє формалізувати взаємодію серверів із запитами без потреби навчання, забезпечуючи стійкість до відхилень. Наукова новизна полягає у використанні рівноваги Неша для оптимального розподілу навантаження з урахуванням поточного стану серверів і можливості декомпозиції задач. Запропонований метод є основою для створення нових адаптивних систем балансування навантаження.

Ключові слова: серверний кластер, інформаційна система, балансування навантаження, управління навантаженням, відмовостійкість, продуктивність, масштабованість, оптимізація ресурсів, паралельні обчислення.

Nerознаk Ye., Kondrus A., Trotsko O., Usyk A., Kostiv O. Analysis of the current state of functioning of server clusters of information systems and load management methods. The article analyzes the current state of functioning of the server infrastructure of information systems and methods of managing the load on server clusters. The main factors affecting the efficiency of work are identified: the growth in the number and complexity of client requests, limited hardware resources, the impossibility of scaling the infrastructure, as well as the insufficient performance of existing load balancing methods. It was found that most modern algorithms have high computational complexity, low resistance to failures and significant dependence on the quality of training data. The study considered classical and intellectual approaches to the optimal use of cluster system resources. Adaptive methods based on game theory, in particular Nash equilibrium, were recognized as the most effective. This approach allows you to formalize the interaction of servers with requests without the need for training, ensuring resistance to deviations. The scientific novelty lies in using Nash equilibrium for optimal load distribution taking into account the current state of servers and the possibility of task decomposition. The proposed method is the basis for creating new adaptive load balancing systems.

Keywords: server cluster, information system, load balancing, load management, fault tolerance, performance, scalability, resource optimization, parallel computing.

Постановка завдання дослідження.

У сучасному темпі розвитку інформаційних технологій, зростання кількості та складності клієнтських запитів до сервісів інформаційних систем (далі – ІС), особливо військового призначення, має значний рушійний вплив на процеси забезпечення ефективного, надійного, стійкого та продуктивного функціонування їх серверної інфраструктури. Випадки відмови ІС внаслідок перевантаження їх технічної складової, стають не лише важливими, але й критичними в умовах високого попиту на інформаційні ресурси в різних сферах діяльності нашої держави.

Україна, як учасниця глобального товариства ІТ, також зіштовхується із такими проблемними питаннями та викликами у сфері ІТ. З одного боку, стрімкий технологічний розвиток надає нові можливості для оптимізації функціонування серверної інфраструктури, з іншого – тенденції постійного зростання очікування та потреб користувачів ІС вимагають постійного вдосконалення та гнучкого масштабування такої інфраструктури [1 – 4].

Розглядаючи сучасну динаміку технологічного розвитку, особливу увагу доцільно приділяти реакції серверної інфраструктури на обробку великих обсягів даних та складність обробки запитів. Інтенсивність, з якою вони надходять до сервісів ІС та їх різнотипність, вимагають перегляду існуючих підходів до управління навантаженням та гнучкості серверної інфраструктури [1 – 4]. Питання забезпечення належного рівня доступності та надійності в умовах постійного зростання вимог до ІТ-ресурсів на даний час стали одними з найактуальніших у багатьох сферах діяльності держави.

Водночас, зростання кількості віртуальних платформ, хмарних сервісів та додатків для обробки та аналізу даних є ще однією проблемною складовою [5]. Вони спричиняють зростання вимог до використання ресурсів технічної складової, що у свою чергу вимагає пошуку балансу між ефективністю та оптимізацією витрат.

Суттєва роль ІС в умовах сучасності підкреслює необхідність подальшого дослідження проблемних питань, пов'язаних із перевантаженням серверної інфраструктури та впливових чинників на даний процес. Разом з тим, стабільність та надійність таких систем мають вплив не лише на ефективність функціонування державних органів, відомств, підприємств та інших, а й на безпеку та комфорт користувачів у цифровому просторі [6, 7]. Так, в умовах постійного зростання попиту на інформатизацію усіх сфер життєдіяльності суспільства залишається відкритим питання надійного та стабільного (неперервного) функціонування інформаційних систем/сервісів (ІС), що надають послуги користувачам у режимі реального часу.

Метою дослідження є визначення чинників, які мають впливовий характер на процеси забезпечення ефективного функціонування серверної інфраструктури інформаційних систем.

Об'єктом дослідження є процеси функціонування та управління навантаженням на серверні кластери інформаційних систем.

ЗЕЛЕНИМ КОЛЬОРОМ ВИДІЛЕНО ВНЕСЕНІ ЗМІНИ.

Аналіз останніх публікацій.

У роботах [8, 9] проведено аналіз функціонування розподілених систем обробки та зберігання даних. Автори аналізують архітектурні рішення та підходи для побудови розподілених систем обробки та зберігання даних, забезпечення відмовостійкості та безперервності роботи, але не приділяється увага чинникам, які мають впливовий характер на функціонування таких систем, зокрема непередбачуваному зростанню кількості та складності клієнтських запитів до сервісів інформаційних систем. Також не приділяється увага підвищенню ефективності функціонування розподілених систем шляхом застосування алгоритмів та методів управління навантаженням.

Актуальна проблема окреслюється у роботі [10] щодо планування ресурсів в обчислювальних системах, однак у ній недостатньо виділені чинники, що безпосередньо впливають на функціонування серверних кластерів. Автор зосереджує увагу переважно на класифікації дисциплін обслуговування заявок та теоретичному обґрунтуванні переходу від однопроцесорних до розподілених систем. Водночас залишаються поза увагою ключові аспекти, серед яких: балансування навантаження між вузлами, надійність, відмовостійкість, вплив мережевих затримок масштабованість та гетерогенність ресурсів. Разом з тим, не розглянуто динамічні чинники, зокрема зростання кількості клієнтських запитів, ускладнення їх структури та змісту, нерівномірність розподілу запитів на обробку, а також потреба в адаптивних механізмах реагування на пікові навантаження.

У дослідженні [11, 12] автори акцентують увагу на підвищенні ефективності функціонування розподілених систем та врахуванні функціональної стійкості, що підкреслює актуальність дослідження. У роботі зазначено одні з ключових критеріїв розподілу ресурсів, такі як: вартість, час, енергоспоживання, стійкість. Однак, мало уваги приділяється не менш важливим чинникам таким як: оптимальне використання серверних ресурсів, не передбачуване зростання кількості та складності клієнтських запитів, а також аналіз адаптивних систем балансування навантаження.

У роботі [13] представлено порівняльний аналіз платформ оркестрації контейнерів (Kubernetes та Docker Swarm). Автори оцінюють їх продуктивність, масштабованість та зручність адміністрування. Практична цінність роботи полягає в безпосередньому зіставленні популярних інструментів, проте акцент зроблено більше на функціональних можливостях, ніж на алгоритмах балансування навантаження.

У дослідженні [14] авторами запропоновано моделі прогнозування споживання ресурсів у Kubernetes, що корисно для планування і адаптивного балансування навантаження у сучасних кластерних системах. Але у ході експериментів, автори використовують стрес-тести, які генеруються запитами із рівномірною затримкою. У ході реального функціонування кластерних систем, запити на обробку надходять із різною інтенсивністю, що є досить непередбачуваним процесом. Разом з тим, як вказано авторами, запропонована методологія не призначена для дослідження критичних навантажень на сервіси.

Розглянуті публікації охоплюють ключові напрямки розвитку серверних кластерів і хмарних систем: балансування навантаження, планування ресурсів, відмовостійкість, енергоефективність, контейнерну оркестрацію та практичні інструменти для досліджень. Їх спільною сильною стороною є систематизація знань та виявлення перспективних напрямків. Водночас роботи мають певні обмеження: переважно аналізуються окремі аспекти та рідко пропонуються комплексні підходи, що враховували б одночасно тенденції зростання кількості та складності запитів до сервісів, оптимальне використання серверних ресурсів кластерних систем, можливість декомпозиції окремих складних завдань на підзавдання, адаптивне балансування навантаження та стійкість до відмов у динамічних умовах функціонування серверних кластерів.

Виклад основного матеріалу.

Насамперед, аналіз, слід розпочати із визначення основних термінів та понять досліджуваної предметної області.

Сервер [15 – 18] – це комп'ютерний пристрій або програма, яка надає послуги чи ресурси іншим пристроям (клієнтам) у комп'ютерній мережі. Головною функцією сервера є обробка запитів, які надходять від клієнтських пристроїв, та надання їм необхідних даних, послуг або ресурсів.

Сервери використовуються для виконання різного роду завдань, включаючи збереження та обмін даними (файлові сервери), надання веб-сторінок та додатків через глобальну мережу (веб-сервери), обробку електронної пошти (поштові сервери), керування мережею (мережеві сервери), та інші функції.

Сервери можуть бути фізичними комп'ютерами або віртуальними машинами, що працюють на фізичних серверах. Вони мають спеціалізоване програмне забезпечення та налаштування для виконання конкретних завдань у мережі.

Серверна інфраструктура [15 – 18] – це сукупність апаратних та програмних засобів, які використовуються для забезпечення функціонування серверів та інших обчислювальних пристроїв у комп'ютерній мережі. Ця інфраструктура включає у себе фізичні сервери, комутатори, маршрутизатори, сховища даних, кабельну мережу, а також програмне забезпечення для управління, моніторингу та забезпечення безпеки цих компонентів.

Головна мета серверної інфраструктури полягає у забезпеченні надійного та ефективного функціонування серверів, для надання послуг та ресурсів користувачам та програмам. До складу серверної інфраструктури можуть входити різні компоненти, залежно від потреб.

Серверний кластер [15 – 18] – це набір декількох незалежних обчислювальних машин (серверів), які об'єднані спільною мережею та програмною логікою, що використовуються спільно і функціонують як одна єдина система. Він дозволяє здійснювати паралельну обробку великих за обсягом даних, ефективно функціонувати веб-сайтам із високим навантаженням, додаткам із багатокористувацьким доступом та виконувати інші завдання, які потребують великої кількості обчислювальних ресурсів.

У серверному кластері, кожен сервер може мати свою власну копію даних та конфігурацій, які синхронізуються з іншими серверами в режимі реального часу, щоб забезпечити високу доступність до даних та послуг. У випадку відмови одного сервера у кластері, інші сервери можуть брати на себе його навантаження, для забезпечення безперебійної роботи послуг.

Серверні кластери можна класифікувати за кількома ключовими характеристиками [15 – 18], які визначають інженери та науковці в сучасному світі, які наведені на рисунку 1.



Рис. 1. Класифікація серверних кластерів за загальними критеріями

– розподілений ресурс (distributed resource): у такому типі кластера кожен вузол має свої власні ресурси (пам'ять, обчислювальну потужність) та власне сховище даних. Комунікація між вузлами здійснюється через мережу;

– спільний ресурс (shared resource): у цьому типі кластера всі вузли мають спільний доступ до спільного сховища даних (диска). Кожен вузол може мати власні обчислювальні ресурси та пам'ять;

– однорідні (homogeneous): есі вузли в кластері мають однакову апаратну конфігурацію та програмне забезпечення;

– різнорідні (heterogeneous): вузли можуть мати різні характеристики щодо апаратної конфігурації та програмного забезпечення;

– високої продуктивності (High-Performance Computing): використовуються для вирішення великих обчислювальних задач, які вимагають великої кількості обчислювальних ресурсів;

– розподіл навантаження (Load Balancing): використовуються для розподілу навантаження між вузлами для забезпечення ефективного використання ресурсів;

– високої доступності: (High Availability): мають на меті максимізувати доступність системи та запобігати відмовам;

– обчислювальний кластер (Compute Cluster): концентрація уваги фокусується на вирішенні завдань обчислень та обробки даних;

– сховище даних (Storage Cluster): орієнтований на забезпечення надійного сховища даних та швидкого доступу до них;

– вертикальне масштабування (Scale-Up): збільшення потужності вузла шляхом додавання ресурсів (наприклад, пам'яті, процесорів);

– горизонтальне масштабування (Scale-Out): збільшення кількості вузлів у кластері.

Таким чином, визначивши основні терміни та поняття можна перейти до аналізу предметної області.

За звітними даними сервісу публічної транснаціональної корпорації Google [19, 20] кількість клієнтських запитів, зростає майже експоненціально.

Статистика даної корпорації показує наступне:

- розмір індексу Google – від 30 до 50 мільярдів сторінок;
- щодня в Google здійснюється понад 3,7 мільярди пошукових запитів;
- 76% усіх пошукових запитів по всьому світу здійснюється через пошукову систему Google;
- пошуковий індекс Google містить понад 100 000 000 гігабайт інформації;
- 16-20% усіх щорічних результатів пошуку Google є новими;
- понад 60% пошукових запитів у Google здійснюються з мобільних пристроїв.

Темпи зростання користувачів глобальної мережі Інтернет становлять 8,2% на рік. Для прикладу (рис. 1.1) з 2009 по 2012 роки кількість пошукових запитів у Google, виконаних за рік, зросла на 10%.

Невдовзі після свого заснування в 1998 році, пошукова система Google обробляла близько 10 000 запитів на день, через рік це число зросло до 3,5 мільйонів пошукових запитів на день. Далі, менше ніж за рік після того, пошукова система Google обробляла близько 18 мільйонів запитів на день [19, 20].

На сьогоднішній день пошукова система Google обробляє близько 40 000 запитів на секунду, що означає, що по всьому світу пошукова система Google обробляє понад 3,7 мільярди запитів на день та близько 1,2 трильйони запитів на рік [19, 20].

Беручи до уваги вищезазначену інформацію, слід розуміти, що тенденції зростання кількості запитів, формують вимоги до ефективності функціонування, як і самих ІС, які здійснюють обробку запитів, так і їх серверної інфраструктури, на базі якої, вони функціонують.

Процеси забезпечення ефективного управління ресурсами серверної інфраструктури відіграють важливу роль у надійному та стійкому функціонуванні різного роду ІС, які забезпечують підвищення ефективності координації, регулювання та прийняття управлінських рішень на різних рівнях органів державної влади, в тому числі посадових осіб органів військового управління, і вихід з ладу будь-якого компонента такої системи, у контексті перевантаження серверної інфраструктури, може тягнути за собою досить серйозні та негативні наслідки, в умовах мирного та воєнного періоду [3, 7].

Так, за звітними даними, у 2016 – 19 роках [21 – 24] неодноразово з'являлись повідомлення про виникнення труднощів щодо подання декларацій до Національного агентства з питань запобігання корупції через систему декларування, в наслідок чого громадяни України не мали змоги вчасно подати інформацію про свої доходи. Збої у системі були спричинені різким зростанням запитів на перше та/або повторне відправлення заповнених та/або доопрацьованих форм е-деклараций до системи, що спричинило перевантаження її серверної інфраструктури.

За даними у [22], за одну із діб системою було зафіксовано 3,8 мільйонів запитів до системи реєстру.

За даними у [21], було проведено моніторинг за допомогою відкритого програмного інтерфейсу додатку (API) динаміки подання декларацій. Загалом з 1 січня по 26 березня було подано 675 433 е-деклараций. Згідно зі статистичними даними [21], динаміка подання документів зростала з наближенням кінцевого строку подання е-деклараций (до 31 березня включно). Найбільша кількість е-деклараций було відправлено до системи 21 березня – 59 197, після чого система реєстру почала працювати нестабільно та зі збоями, в наслідок чого кількість поданих е-деклараций зменшувалася до 26 березня – до моменту повної відмови системи в обслуговуванні. Для відновлення роботи системи адміністраторам довелося здійснювати заходи щодо розширення пропускну здатності фізичних каналів зв'язку та збільшення обчислювальної потужності серверного обладнання, в тому числі на рівні віртуалізації.

13 грудня 2021 року [25] за ініціатииви Президента України набула чинності українська державна програма "єПідтримка" яка спрямована на надання грошової допомоги громадянам України за рахунок державного бюджету. В рамках даного проєкту громадяни України мали змогу отримати грошову допомогу за проведену вакцинацію від вірусної хвороби COVID-19. Дана допомога здійснюється за допомогою державного програмного додатку (програмного сервісу) "Дія". Після "старту" вищезазначеного проєкту від громадян України (користувачі сервісу "Дія") почали надходити скарги щодо проблем та нестабільної роботи сервісу, а саме не були доступні для перегляду та використання персональних документів у цифровому вигляді (паспорт громадянина України, водійське посвідчення, свідоцтво про народження та інші). Нестабільність та збої у функціонуванні додатку була спричинена зростанням кількості запитів до сервісів порталу "Дія" в наслідок чого її серверна інфраструктура була перевантажена кількістю завдань, пов'язаних із процесами генерації сертифікатів про проходження вакцинації, всіма користувачами "Дії" в один проміжок часу, ініційованих даними запитам, так як близько 8 мільйонів громадян (за розрахунками Міністерства цифрової трансформації України) на протязі від 1 до 2 тижнів мали отримати грошову допомогу [25]. Для вирішення вищезазначеного завдання було здійснено адміністраторами заходи щодо оптимізації функціонування програмного забезпечення та збільшення обчислювальних ресурсів серверної інфраструктури.

10 травня 2023 року [26 – 28] в Україні запрацювала державна програма "єВідновлення" в рамках, якої громадяни України мають змогу отримати від держави грошову допомогу на відновлення житла, пошкодженого в наслідок повномасштабного вторгнення 24 лютого 2022 року країни-агресорки російської федерації на територію України. Подання заяв громадян здійснюється на основі клієнтських запитів до порталу "Дія", в наслідок чого, через різке зростання їх кількості у перший день роботи даної програми (реєструвались та оброблялись системою тисячі заяв), фіксувалася нестабільна робота та збої у системі реєстрації вищезазначеного порталу. Серверна інфраструктура була перевантажена обробкою такої кількості клієнтських запитів, що у свою чергу даний факт спричинив нестабільність роботи системи, та в подальшому, відмову в обслуговуванні. Для вирішення вищезазначеного завдання адміністраторами було здійснено заходи щодо оптимізації

функціонування програмного забезпечення та збільшення обчислювальних ресурсів серверної інфраструктури.

З початком повномасштабного вторгнення країни-агресорки російської федерації на територію України 24 лютого 2022 року, кількість різного роду інформації, зокрема про стан, кількість, шляхи пересування, місця дислокації, геодані ворога стрімко зростає, оскільки генерується великою кількістю громадян України, та передавалась до відповідних державних та оборонних структур для подальшої обробки та аналізу, після чого здійснювались процеси пов'язані із прийняттям управлінських рішень посадовими особами відповідних структур для ефективного застосування та управління всіма наявними оборонними ресурсами держави. Відповідно, забезпечення вищезазначених процесів покладалося на інформаційні системи та комплекси військового призначення.

У даному контексті особливої уваги заслуговує стабільне (неперервне) функціонування не тільки таких ІС критичної інфраструктури держави, ІС військового призначення (сил оборони та безпеки), але і серверної інфраструктури на базі якої, вони розгорнуті, адже їх функціонування як у повсякденній діяльності, так і в умовах воєнного стану передбачають щоденне прийняття важливих для національної безпеки рішень, захисту територіальної цілісності України, управління військами, провадження антитерористичної та контрдиверсійної діяльності, збереження життя, прав і свобод громадян [29, 30].

Одним з основних напрямів вирішення даного завдання є підвищення продуктивності кластерних систем – апаратного фундаменту загального і спеціального програмного забезпечення, завдяки ефективному балансуванню навантаження між серверами кластерних систем для оптимального їх використання в умовах обмеження апаратних ресурсів і прийнятної часу на обробку запитів, оскільки стек сучасного програмного забезпечення (ПЗ) інформаційної взаємодії органів військового управління з підлеглими підрозділами у результаті інтенсивності і складності запитів може вимагати значних обчислювальних потужностей, особливо у випадку обробки великих обсягів даних (*Big Data*), що негативно впливає на продуктивність і ефективність інформаційних та обчислювальних систем, інтегрованих у військову галузь [31, 32].

На даний час, існує значна кількість рішень щодо розподілу та управління навантаженням на кластерні системи за рахунок застосування різного роду методів та алгоритмів, створених вирішувати задачі оптимального використання системних ресурсів, безперервного забезпечення потреб користувачів, раціонального використання екомонічних витрат [33, 34]. Разом з тим існують проблемні питання пов'язані із розподілом навантаження на серверні кластери, які слід вирішувати ще на ранній стадії планування та розгортання будь-якої ІС [33]. Відмова сервера кластеру, що зазвичай відбувається несподівано, а також у відповідальний момент часу, тягне за собою серйозні наслідки, що особливо актуально для систем спеціального призначення у військовій сфері діяльності.

Завдання недостатньої продуктивності сервера (вузла) у зв'язку зі зростанням навантаження можна вирішувати шляхом нарощування потужності сервера, або оптимізацією використовуваних алгоритмів, програмних кодів і т.д [33]. Але рано чи пізно настає момент, коли ефективність цих заходів виявляється низькою, особливо з урахуванням коштів, які інвестуються в розвиток серверної інфраструктури. Іншим способом підвищення продуктивності серверів є їхнє об'єднання у кластер, у якому навантаження розподіляється між серверами (вузлами) за допомогою комплексу спеціальних методів та алгоритмів балансування навантаження. Крім вирішення завдання високих навантажень, технологія кластеризації допомагає також забезпечити резервування серверів, ефективність якого знову ж таки залежить від того, як розподіляється (балансується) навантаження між вузлами кластера [33 – 36].

Балансування навантаження може здійснюватися за допомогою апаратних і програмних інструментів та може бути реалізоване на мережевому, транспортному та прикладному рівнях моделі OSI.

Розглянемо ці рівні більш детально.

Балансування навантаження кластерної системи на мережевому рівні передбачає таке підключення сервера до мережі, за якого його кожна IP-адреса (в тому числі віртуальна) обслуговується різними фізичними серверами (вузлами кластера). Балансування на мережевому

рівні передбачає вирішення наступного завдання: потрібно зробити так, щоб за одну конкретну IP-адресу сервера відповідали різні фізичні машини [33, 34, 36 –38].

Такого роду балансування може здійснюватися за допомогою декількох способів [33, 34, 36 –38]:

- DNS-балансування. На одне доменне ім'я виділяється кілька IP-адрес. Сервер, на який буде спрямований клієнтський запит, зазвичай визначається за допомогою алгоритму Round Robin.

- побудова NLB-кластера. При використанні даного способу сервери об'єднуються в кластер, що складається з вхідних і обчислювальних вузлів. Розподіл навантаження здійснюється за допомогою спеціального алгоритму (використовується в рішеннях від компанії Microsoft).

- балансування по IP-адресам з використанням додаткового маршрутизатора.

- балансування за територіальною ознакою здійснюється шляхом розміщення однакових сервісів з однаковими IP-адресами в територіально різних регіонах мережі Інтернет.

Балансування на транспортному рівні є найпростішим [33, 34, 36 –38]: клієнт надсилає запит до балансувальника, який перенаправляє запит до одного із серверів, який і буде його обробляти. Вибір сервера, на якому буде оброблятися запит, може здійснюватися у відповідності з різними алгоритмами шляхом простого кругового перебору, шляхом вибору найменш завантаженого сервера з пулу і т.д. На транспортному рівні процес балансування відрізняється простотою та ефективністю [33, 34, 36 –38]. Зниження навантаження на сервер на транспортному рівні передбачає використання балансувальника, який розподіляє запити по пулу відповідно до заданих алгоритмів. Балансувальник передає обраному серверу запити, а потім отримує відповідь на них і перенаправляє його назад користувачу. Найпростіші (але при цьому популярні) транспортні алгоритми – це може бути елементарний Round Robin (кожний запит буде надсилатися на наступний сервер) або Lest Connections – пошук найменш завантаженого сервера [33, 34, 36 –38].

Іноді балансування на транспортному рівні складно відрізнити від балансування на мережевому рівні. Різницю між мережевим та транспортним рівнями балансування можна пояснити наступним чином. До мережевого рівня відносяться рішення, які не термінують на собі призначені для користувача сесії. Вони перенаправляють трафік і не працюють в режимі проксуювання. На мережевому рівні балансувальник просто вирішує, на який сервер передавати пакети, а в свою чергу сесію з клієнтом здійснює сервер [33, 34, 36 –38].

На транспортному рівні обмін даними з клієнтом здійснює балансувальник, який працює як проксі. Він взаємодіє з серверами від свого імені, передаючи інформацію про клієнта в додаткових даних і заголовках. Таким чином, працює популярний програмний балансувальник HAProxy [33, 34, 36 –39].

При балансуванні на прикладному рівні балансувальник працює в режимі “розумного проксі”. Він аналізує клієнтські запити і перенаправляє їх на різні сервери в залежності від характеру запитуваного контенту. Так працює, наприклад, веб-сервер Nginx, розподіляючи запити між фронтомом та бекендом. За балансування в Nginx відповідає модуль Upstream [33, 34, 36 – 38, 40].

В якості ще одного прикладу інструменту балансування на прикладному рівні можна привести pgpool – проміжний шар між клієнтом і сервером СКБД PostgreSQL [33, 41]. За його допомогою можна розподіляти запити між серверами баз даних в залежності від їх вмісту: наприклад, запити на читання будуть передаватися на один сервер, а запити на запис – на інший.

Таким чином, розглянувши на яких рівнях здійснюються процеси балансування навантаження, доцільно детально проаналізувати, які існують методи та алгоритми балансування.

Існує певна кількість різних алгоритмів і методів балансування навантаження [33, 34, 36 – 38]. Обираючи той чи інший алгоритм, потрібно чітко розуміти особливості конкретного проекту та цілі, які потрібно досягти.

У числі цілей, для досягнення яких, використовується балансування, потрібно виділити наступні [33, 34, 36, 42]:

- справедливість: потрібна гарантія того, що на обробку кожного запиту виділялася достатня кількість системних ресурсів і не допущення виникнення ситуацій, коли один запит обробляється, а всі інші чекають своєї черги;

- ефективність: полягає у тому, що всі сервери, які обробляють запити, повинні бути навантажени, тобто не допускати ситуації, коли один з серверів простоює в очікуванні запитів на обробку (на практиці ця мета досягається не завжди);

– скорочення часу виконання запиту: потрібно забезпечити мінімальний час між початком обробки запиту (або його постановкою в чергу на обробку) і його завершення;

Разом з тим алгоритм балансування повинен мати такі властивості як [33, 36, 37, 42]:

– передбачуваність: потрібно чітко розуміти, в яких ситуаціях і при яких навантаженнях алгоритм буде ефективним для вирішення поставлених завдань;

– рівномірне навантаження на ресурси системи;

– масштабованість: алгоритм повинен зберігати працездатність при збільшенні навантаження.

Виходячи з вищезазначених вимог, розглянемо основні методи балансування детальніше.

Round Robin, або метод кругового обслуговування, являє собою перебір по круговому циклу: перший запит передається одного сервера, потім наступний запит передається іншому і так до останнього сервера, потім цикл починається спочатку [33, 34, 37, 39, 40].

Найпоширенішою імплементацією цього методу є алгоритм балансування *Round Robin DNS*. Як відомо, будь-який DNS-сервер зберігає пару “ім'я хоста - IP-адреса” для кожної машини в певному домені. Цей список може виглядати наступним чином [33, 34, 37, 39, 40]:

www.example.com IN A 192.168.1.3

Або з кожним ім'ям зі списку можна асоціювати кілька IP-адрес:

www.example.com IN A 192.168.1.2
IN A 192.168.1.3
IN A 192.168.1.4
IN A 192.168.1.5

DNS-сервер перебирає всі записи в таблиці і віддає на кожен новий запит наступну IP-адресу: наприклад, на перший запит – xxx.xxx.xxx.2, на другий – xxx.xxx.xxx.3, і так далі. В результаті всі сервери в кластері отримують однакову кількість запитів [33, 34, 37, 39, 40]. У числі переваг цього методу є незалежність від протоколу високого рівня. Для роботи методу Round Robin використовується будь-який протокол, в якому звернення до сервера здійснюється за ім'ям.

Балансування на основі методу Round Robin ніяк не залежить від навантаження на сервер: кешуючі DNS-сервери допоможуть впоратися з будь-яким числом клієнтів. Використання методу Round Robin не вимагає зв'язку між серверами, тому він може використовуватися як для локального, так і для глобального балансування. Невід'ємною частиною його переваг є те, що рішення на основі методу Round Robin вирізняється низькою вартістю, тобто, щоб метод почав працювати, досить просто додати кілька записів в DNS [33, 34, 37, 39, 40].

Разом з тим, метод Round Robin має і цілий ряд істотних недоліків таких як [33, 34, 37, 39, 40]:

– розподіл навантаження не відповідає вказаним вище критеріям справедливості і ефективності, тому, що потрібно, щоб кожен сервер мав в наявності однаковий набір ресурсів;

– при виконанні всіх операцій також має бути задіяно однакову кількість ресурсів. На практиці ці умови в більшості випадків виявляються нездійсненними;

– при управлінні навантаженням за методом Round Robin абсолютно не враховується завантаженість того чи іншого сервера в складі кластера: наприклад, один з вузлів завантажений на 100%, в той час як інші – всього на 10 – 15%. Метод Round Robin можливості виникнення такої ситуації не враховує в принципі, тому перевантажений вузол все одно буде отримувати запити;

Таким чином, сфера застосування методу Round Robin досить обмежена.

Weighted Round Robin [33, 34, 37, 40]. Це – вдосконалена версія методу Round Robin суть удосконалень, якого полягає в наступному: кожному серверу присвоюється ваговий коефіцієнт відповідно до його продуктивності і потужності.

Це дозволяє розподіляти навантаження більш гнучко, тобто сервери з більшим ваговим коефіцієнтом обробляють більшу кількість запитів. Однак усіх проблем з відмовостійкістю не вирішує.

Більш ефективне балансування забезпечують інші методи, в яких алгоритми враховують більшу кількість параметрів при плануванні та розподілі навантаження [33, 34, 37, 40].

Least Connections. У вищезазначеному було перераховано основні недоліки методу Round Robin та його вдосконаленої версії, але існує ще один суттєвий недолік: у ньому абсолютно не враховується кількість активних на даний момент підключень.

Розглянемо практичний приклад. Є два сервера S1 та S2. До сервера S1 підключено менше користувачів, ніж до сервера S2. При цьому сервер S1 виявляється більш навантаженим. Причина цієї події може полягати в тому, що підключення до сервера S1 підтримуються протягом довшого часу в порівнянні з підключеннями до сервера S2.

Описане завдання можна вирішити за допомогою методу Least Connections. Цей метод враховує кількість активних підключень, підтримуваних серверами в поточний момент часу. Кожен наступний клієнтський запит буде передаватись серверу з найменшою кількістю активних підключень.

Weighted Least Connections [33, 34, 37], являє собою вдосконалений варіант попереднього методу та призначений в першу чергу для використання в кластерах, що складаються з серверів з різними технічними характеристиками і різною продуктивністю. Цей метод враховує при розподілі навантаження не тільки кількість активних підключень, а й ваговий коефіцієнт серверів.

У числі інших вдосконалених варіантів методу Least Connections слід перш за все виділити методи Locality-Based Least Connection Scheduling і Locality-Based Least Connection Scheduling with Replication Scheduling [33, 34, 37].

Перший метод був створений спеціально для кешуючих проксі-серверів, алгоритм роботи, яких полягає в наступному: найбільша кількість запитів передається серверам з найменшою кількістю активних підключень. За кожним з клієнтських серверів закріплюється група клієнтських IP-адрес. Запити з цих IP-адрес передаються на "рідний" сервер, при умові, що він не завантажений повністю. Якщо умова не виконується, то запит буде перенаправлено на інший сервер при умові, що він повинен бути завантажений менше ніж наполовину.

В методі Locality-Based Least Connection Scheduling with Replication Scheduling кожна IP-адреса або група IP-адрес закріплюється не за окремим сервером, а за цілою групою серверів, відповідно запит буде передаватись найменш завантаженому серверу з групи. Якщо всі сервери з "рідної" групи перенавантажені, то буде додано новий сервер до групи, яка обслуговує IP-адресу, з якої був відправлений запит. У свою чергу найбільш завантажений сервер буде виведено з цієї групи – це дозволяє уникнути надмірної реплікації [33, 34, 37].

Таким чином, розглянувши принцип роботи найпоширеніших методів балансування навантаження можна визначити, які параметри враховує кожен з них для прийняття рішення на балансування (таблиця 1).

Таблиця 1. Порівняння основних (класичних) методів балансування навантаження

Метод Критерій порівняння	RR	WRR	LC	WLC	LBLCS	LBLCSRS
Завантаженість сервера (вузла)	–	–	–	–	+	+
Ваговий коефіцієнт сервера (вузла)	–	+	–	+	–	–
Кількість активних підключень до сервера (вузла)	–	–	+	+	+	+
Характер запиту (читання, запис, контент)	–	–	–	–	–	–

Однак, аналіз останніх досліджень, які проаналізовані у [31] за даною тематикою показує значну перевагу застосування адаптивних алгоритмів (способів, моделей, методів, методик) балансування навантаження кластерних систем у порівнянні із вищезазначеними найбільш поширеними і водночас простішими методами.

Так, підвищення ефективності адаптивних методів балансування навантаження кластерних систем досягається шляхом покращення показників (властивостей) [31]:

продуктивності: передбачає здійснення розподілу навантаження між обчислювальними ресурсами таким чином, щоб кожен вузол кластера функціонував з максимальною ефективністю;

стійкості до відмов: здатність системи реагувати на зміни навантаження та оптимально перерозподіляти його між ресурсами у реальному часі;

економії ресурсів: використання наявних ресурсів з максимально можливим показником оптимуму (глобальний оптимум), що забезпечує скорочення витрат на обслуговування завдань;

адаптивності: налаштування на специфічні умови функціонування, що дозволяє системі швидко адаптуватися до змін навантаження і вимог користувачів;

точності: використання значної кількості ознак для прийняття рішень, що підвищує точність розподілу навантаження між ресурсами.

Поряд з цим, деякі із перелічених властивостей, в силу своєї природи, здатні негативно впливати на процес прийняття рішення балансувальником навантаження [31]. Наприклад, властивості адаптивності і точності безпосередньо залежать від якості алгоритмів навчання, які використовуються в системі балансування навантаження, що цілком може свідчити про його недостатню функціональну стійкість за різних умов функціонування. У випадку застосування експертних систем піднімаються питання узгодженості суб'єктивних суджень експертів, мінімально необхідної їх кількості, а також відповідного рівня їх компетентності, тоді, як у випадку застосування методів машинного навчання піднімаються питання репрезентативності навчальної вибірки та інформативної значущості простору її ознак. До того ж, у процесі вибору найефективнішого підходу до визначення оптимального використання ресурсів кластерів у розглянутих наукових дослідженнях [31, 34] недостатньо уваги приділяється наступним властивостям, які також безпосередньо впливають на кінцевий результат прийняття рішення балансувальником навантаження:

верифікованість – забезпечення можливості перевірки коректності роботи методу людиною (перевірка і підтвердження коректності виконання покладених на нього функцій);

якість навчання – показник ефективності (точність / достовірність / повнота / F-міра тощо) виконання методом поставлених перед ним завдань на основі наявних даних (визначається на основі порівняння згенерованих методом результатів з фактичними значеннями із навчальної вибірки (чим менша різниця між отриманими та фактичними значеннями, тим вища якість навчання методу));

робастність – характеристика незалежності впливу на результат дослідження різноманітних виключень (аномалій), стійкість до перешкод (здатність надавати правильні результати навіть за наявності деяких неправильних значень чи помилок у даних);

гарантія оптимуму – теоретичне обґрунтування, яке дозволяє стверджувати, що знайдене рішення оптимальне або близьке до оптимального для певного завдання (завжди у глобальному оптимумі).

На цій основі виникає необхідність порівняння існуючих адаптивних методів (алгоритмів) [31] оптимального використання серверних ресурсів кластерних систем на основі критеріїв *верифікованості, залежності від якості навчання, робастності та гарантії глобального оптимуму*. У таблиці 2 представлено результати порівняльного аналізу існуючих адаптивних методів (алгоритмів) [31] оптимального використання серверних ресурсів кластерних систем на основі вищезазначених критеріїв.

Таблиця 2. Результати порівняльного аналізу існуючих адаптивних методів оптимального використання серверних ресурсів

№ з/п	Найменування методу	Верифікованість	Залежність від якості навчання	Робастність	Гарантія глобального оптимуму
1	Регресійний аналіз	низька	висока	низька	середня
2	Системи масового обслуговування	середня	низька	висока	середня

3	Нейронні мережі	низька	висока	середня	середня
4	Нечітка логіка	висока	висока	висока	середня
5	Рівновага Неша (теорія Ігор)	висока	низька	висока	висока
6	Дерева рішень	висока	середня	низька	низька
7	Еволюційні алгоритми	низька	низька	висока	висока
8	Випадковий ліс	низька	середня	середня	середня
9	Аналіз ентропії	низька	середня	середня	середня
10	Байєсівські мережі	низька	середня	середня	середня
11	Експертні системи	висока	висока	висока	середня

На основі проведеного аналізу можна зробити висновок, що найбільш повну відповідність висунутим критеріям порівняння існуючих адаптивних методів оптимального використання серверних ресурсів демонструють методи, в основу яких покладено концепцію теорії Ігор щодо прийняття оптимальних рішень в умовах конфлікту – рівноваги Неша [31, 32]. Даний науково-методичний апарат дозволяє ефективно знаходити оптимальне рішення використання серверів отриманими завданнями, що є стійким до відхилень будь кого з інших учасників гри (будь-яке відхилення від обраних серверів не принесе додаткової користі для клієнтської задачі). Описана взаємодія є некооперативною грою, де гравці – отримані завдання, які намагаються максимізувати свій виграш шляхом вибору сервера із найбільшим показником вільного простору навантаженням, а множина стратегій гри – сервери кластерної системи [32].

Висновки та перспективи подальшого дослідження

На підставі проведеного аналізу сучасного стану функціонування інформаційних систем та їх технічної інфраструктури, а також методів управління навантаженням, було виявлено:

- тенденції постійного збільшення кількості та зростання складності клієнтських запитів до сервісів інформаційних систем;
- відсутність можливості горизонтального та/або вертикального масштабування технічної інфраструктури інформаційних систем в умовах обмеженості апаратних ресурсів серверів;
- потреба у збільшенні обчислювальної потужності технічної інфраструктури інформаційних систем для виконання завдань, спричиненими клієнтськими запитами;
- невідповідність існуючих методів управління навантаженням вимогам забезпечення стійкого та продуктивного функціонування ІС, зокрема в аспекті оптимального (рівномірного) використання ресурсів технічної інфраструктури;
- досить низький рівень стійкості існуючих методів управління навантаженням до збоїв (аномалій) у системі та/або помилок у даних (кодів);
- високий рівень залежності існуючих інтелектуальних методів управління навантаженням від якості навчальної вибірки;
- значна обчислювальна складність більшості існуючих методів управління навантаженням.

Також, проведено дослідження існуючих підходів до оптимального використання ресурсів кластерної системи. На основі проведеного аналізу можна зробити висновок, що найбільш повну відповідність висунутим критеріям порівняння існуючих адаптивних методів оптимального використання серверних ресурсів демонструють методи, в основу яких покладено концепцію теорії Ігор щодо прийняття оптимальних рішень в умовах конфлікту – рівноваги Неша, як найбільш

доцільного підходу до формального опису функціональної взаємодії серверів із клієнтськими запитами, а також як підхід, що не потребує узагальнення емпіричного досвіду (навчання).

Наукова новизна такого підходу полягає у знаходженні оптимального розподілу навантаження на сервери кластерної системи на основі рівноваги Неша із врахуванням узагальненого показника поточного рівня їх завантаженості, а також із врахуванням можливості декомпозиції складних завдань на підзавдання, спричинених клієнтськими запитами.

Даний науково-методичний апарат дозволить ефективно знаходити оптимальне рішення використання серверів отриманими задачами, що є стійким до відхилень будь-кого з інших учасників гри (будь-яке відхилення від обраних серверів не принесе додаткової користі для клієнтської задачі).

Отримані результати аналізу розглядаються як підґрунтя для реалізації нових адаптивних систем балансування навантаження вузлів кластерної системи, застосування яких дозволить оптимально використовувати ресурси кластерної системи у стійкій рівновазі.

Список бібліографічного опису:

1. Tymoshenko N. Y., Ronsky B. Y. PROBLEMS AND PROSPECTS OF DEVELOPMENT OF THE IT INDUSTRY IN UKRAINE. *Economy and Society*. 2018. URL: <https://doi.org/10.32782/2524-0072/2018-17-57>.
2. Омеляченко В. Перспективи та розвиток IT-технологій в Україні. *Блоги про бізнес, політику, юридичну систему* | *LIGA.net* - *LIGA*. URL: <https://blog.liga.net/user/vomelchenko/article/35913>.
3. ІНФОРМАЦІЙНА СИСТЕМА ЯК УМОВА ЕФЕКТИВНИХ УПРАВЛІНСЬКИХ РІШЕНЬ | Н В Бугас | Ефективна економіка №12 2016. *Журнал «Ефективна економіка»* - наукове фахове видання з питань економіки. URL: <http://www.economy.nayka.com.ua/?op=1&z=5313>.
4. Аудит IT-інфраструктури - TechExpert. *TechExpert*. URL: <https://techexpert.ua/what-is-an-audit-of-the-it-infrastructure/>.
5. NATO C3 Classification Taxonomy [Electronic Resource].— Mode of access: <https://www.yumpu.com/en/document/view/24003661/c3-classification-taxonomy-nci-agency-nato>.
6. Krashchenko Y., Holovach L. The impact of digitalization on the economy and society in the conditions of war in the content of professional training of future managers. *The sources of pedagogical skills*. 2022. No. 30. P. 105–111. URL: <https://doi.org/10.33989/2075-146x.2022.30.270656>.
7. Мудренко Н. Цифрові інструменти координації діяльності органів державної влади у сфері зовнішніх зносин. *Науковий вісник: державне управління*. 2020. Т. 4, № 2020-2. С. 249–257. URL: [https://doi.org/10.32689/2618-0065-2020-2\(4\)-249-257](https://doi.org/10.32689/2618-0065-2020-2(4)-249-257).
8. АНАЛІЗ ФУНКЦІОНУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ ОБРОБКИ ТА ЗБЕРІГАННЯ ДАНИХ / А. Білокон та ін. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2024. Т. 3, № 77. С. 84–88. URL: <https://doi.org/10.26906/sunz.2024.3.084>.
9. Somasekaram P., Calinescu R., Buyya R. High-availability clusters: A taxonomy, survey, and future directions. *Journal of Systems and Software*. 2022. Vol. 187. P. 111208. URL: <https://doi.org/10.1016/j.jss.2021.111208>.
10. Сімоненко А. В. ОБҐРУНТУВАННЯ ВИБОРУ ДИСЦИПЛІНИ ОБСЛУГОВУВАННЯ ЗАЯВОК У РОЗПОДІЛЕНИХ СИСТЕМАХ ОБРОБКИ ІНФОРМАЦІЇ. *Таврійський науковий вісник. Серія: Технічні науки*. 2024. № 6. С. 120–129. URL: <https://doi.org/10.32782/tnv-tech.2024.6.13>.
11. ГОРА М. В., ВОЛК М. О. МОДЕЛІ УПРАВЛІННЯ РЕСУРАМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ПРОЦЕСУ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ. *Вісник Херсонського національного технічного університету*. 2024. № 4(87). С. 244–251. URL: <https://doi.org/10.35546/kntu2078-4481.2023.4.28>.
12. Ismail L., Materwala H. Computing Server Power Modeling in a Data Center. *ACM Computing Surveys*. 2020. Vol. 53, no. 3. P. 1–34. URL: <https://doi.org/10.1145/3390605>.
13. Venkat Marella. Comparative Analysis of Container Orchestration Platforms: Kubernetes vs. Docker Swarm. *International Journal of Scientific Research in Science and Technology*. 2024. Vol. 11, no. 5. P. 526–543. URL: <https://doi.org/10.32628/ijrst24105254>.
14. Predicting resource consumption of Kubernetes container systems using resource models / G. Turin et al. *Journal of Systems and Software*. 2023. P. 111750. URL: <https://doi.org/10.1016/j.jss.2023.111750>.
15. Кластери серверів. *ASI*. URL: <https://a-si.com.ua/serverne-ta-shd-obladnannya/klastery-serveriv/>.
16. CGG: High-Performance Computing. *CGG: Global Technology and HPC leader*. URL: https://www.cgg.com/digital-services/high-performance-computing?gclid=CjwKCAjw7c2pBhAZEiwA88pOF2Jz7kIOVu8tDSaM920sR6hGgopG9t1CjGE4VYkouWlYXSbLqlaq_BoCTiEQAvD_BwE.
17. What is HPC? Introduction to high-performance computing | IBM. *IBM in Deutschland, Österreich und der Schweiz* | IBM. URL: <https://www.ibm.com/topics/hpc>.
18. René J. Chevance. *Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions*. Book. USA : Elsevier Digital Press, 2005. 784 p.
19. Google Тренди. *Google Trends*. URL: <https://trends.google.com/trends/?hl=uk>.
20. Google Search Statistics (2023): 17+ Facts You Need to Know. *FirstSiteGuide*. URL: <https://firstsiteguide.com/google-search-stats/>.

21. Сайт НАЗК не витримує навантаження – строк подання більшості декларацій подовжено до 1 травня. *Інтернет-видання «Полтавщина»*. URL: <https://poltava.to/news/42339/>.
22. Ukrinform. Сайт НАЗК "впав" через перевантаження. *Укрінформ - актуальні новини України та світу*. URL: <https://www.ukrinform.ua/rubric-technology/2108140-sajt-nazk-vrav-cerez-perevantazenna.html>.
23. Українська правда. Портал НАЗК перевантажений, вже подано 42 тисячі декларацій. *Українська правда*. URL: <https://www.pravda.com.ua/news/2016/10/26/7124898/>.
24. Новини - Останні новини України сьогодні | РБК-Україна. *Новини - Останні новини України сьогодні | РБК-Україна*. URL: <https://www.rbc.ua/>.
25. «Підтримка і "тисяча Зеленського" обвалили Дію. Що робити? - BBC News Україна. *BBC News Україна*. URL: <https://www.bbc.com/ukrainian/news-59651413>.
26. Корогодський Ю. У перший день роботи у програмі "єВідновлення" стався збій. *LB.ua*. URL: https://lb.ua/economics/2023/05/10/554605_pershiy_den_roboti_programi.html.
27. Redirecting... Facebook - log in or sign up. URL: <https://www.facebook.com/photo/?fbid=616272463874272&set=a.290337743134414>.
28. Міністерство розвитку громад, територій та інфраструктури України. *Redirecting...* URL: <https://www.facebook.com/Ministry.for.restoration>.
29. Про національну безпеку України : Закон України від 21.06.2018 р. № 2469-VIII : станом на 31 берез. 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/2469-19#Text>.
30. Віталій Фесьоха, Надія Фесьоха. Модель нечіткої автентифікації користувачів інформаційних систем органів військового управління на основі поведінкової біометрії. *Захист інформації (НАУ)*. 2021. Том 23, №2, С. 116-123.
31. Удосконалена модель оптимального використання серверних ресурсів кластерної системи військового призначення на основі рівноваги Неша / В. Фесьоха та ін. *Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка*. 2023. С. 162–173. URL: <https://doi.org/10.17721/2519-481X/2023/79-15>.
32. Метод адаптивного балансування навантаження в кластерних системах військового призначення на основі рівноваги Неша / В. Фесьоха та ін. *Збірник наукових праць Центру воєнно-стратегічних досліджень НУОУ імені Івана Черняхівського*. 2023. Т. 3, №76. С. 101–110. URL: <https://doi.org/10.33099/2304-2745/2022-3-76/101-110>.
33. Нерознак Є.І., Меркотан Д.Ю., Сова О.Я. Методи та алгоритми балансування навантаження в кластерних системах на основі елементів штучного інтелекту. І Міжнародна науково-технічна конференція «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку». *BITI*, 2021. – С. 215–217.
34. Load Balancing Algorithms, Types and Techniques - Kemp [Electronic resource] // Load Balancer For Always-On Application Experience - Kemp. – Mode of access: <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques>.
35. Відмовостійкий кластер: крок до безпеки важливих даних [Електронний ресурс]: [Веб-сайт]. – Режим доступу до ресурсу: https://itc.ua/articles/otkazoustojchivyj_klaster_shag_k_bezopasnosti_vazhnyh_dannyh_23460.
36. Динамічне балансування трафіку між декількома провайдерами / Ю. Кльоц та ін. *Вісник Хмельницького національного університету*. 2019. Т. 4, №275. С. 62–67. URL: <https://www.doi.org/10.31891/2307-5732-2019-275-4-62-67>.
37. Overview of Load Balancing Methods in Cloud Systems / R. Minailenko et al. *National Interagency Scientific and Technical Collection of Works. Design, Production and Exploitation of Agricultural Machines*. 2021. No. 51. P. 188–194. URL: <https://doi.org/10.32515/2414-3820.2021.51.188-194>.
38. Балансування навантаження та мікросервіси - HashDork. *HashDork*. URL: <https://hashdork.com/uk/load-balancing-microservices/>.
39. HAProxy version 2.9-dev8 - Starter Guide. *HAProxy Documentation Converter*. URL: <https://docs.haproxy.org/dev/intro.html>.
40. Using nginx as HTTP load balancer. *nginx news*. URL: https://nginx.org/en/docs/http/load_balancing.html.
41. PostgreSQL: Documentation. *PostgreSQL: The world's most advanced open source database*. URL: <https://www.postgresql.org/docs/>.
42. Chen S.-L., Chen Y.-Y., Kuo S.-H. CLB: A novel load balancing architecture and algorithm for cloud services. *Computers & Electrical Engineering*. 2017. Vol. 58. P. 154–160. URL: <https://doi.org/10.1016/j.compeleceng.2016.01.029>.

References:

1. Tymoshenko, N. Y., & Ronsky, B. Y. (2018). Problems and prospects of development of the IT industry in Ukraine. *Economy and Society*. <https://doi.org/10.32782/2524-0072/2018-17-57>.
2. Omelchenko, V. (2020). *Perspektyvy ta rozvytok IT-tehnolohii v Ukraini*. LIGA.net. <https://blog.liga.net/user/vomelchenko/article/35913>.
3. Buhas, N. V. (2016). Informatsiina systema yak umova efektyvnykh upravlinskykh rishen. *Efektyvna ekonomika*, 12. <http://www.economy.nayka.com.ua/?op=1&z=5313>.
4. TechExpert. (n.d.). Audyt IT-infrastruktury. *TechExpert*. <https://techexpert.ua/what-is-an-audit-of-the-it-infrastructure/>.
5. NATO C3 Classification Taxonomy. (n.d.). NCI Agency NATO. <https://www.yumpu.com/en/document/view/24003661/c3-classification-taxonomy-nci-agency-nato>.

6. Krashchenko, Y., & Holovach, L. (2022). The impact of digitalization on the economy and society in the conditions of war in the content of professional training of future managers. *The Sources of Pedagogical Skills*, 30, 105–111. <https://doi.org/10.33989/2075-146x.2022.30.270656>.
7. Mudrenko, N. (2020). Tsyfrovi instrumenty koordynatsii diialnosti orhaniv derzhavnoi vlady u sferi zovnishnikh znosyn. *Naukovyi visnyk: derzhavne upravlinnia*, 4(2), 249–257. [https://doi.org/10.32689/2618-0065-2020-2\(4\)-249-257](https://doi.org/10.32689/2618-0065-2020-2(4)-249-257).
8. Bilokon, A., Borisov, S., Usatenko, M. та Fedorchenko, V., (2024). АНАЛІЗ ФУНКЦІОНУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ ОБРОБКИ ТА ЗБЕРІГАННЯ ДАНИХ. *Системи управління, навігації та зв'язку. Збірник наукових праць* [онлайн]. 3(77), 84–88. Режим доступу: doi: 10.26906/sunz.2024.3.084.
9. Somasekaram, P., Calinescu, R. та Buyya, R., (2022). High-availability clusters: A taxonomy, survey, and future directions. *Journal of Systems and Software* [онлайн]. 187, 111208. Access mode: doi: 10.1016/j.jss.2021.111208.
10. Сімоненко, А. В., (2024). ОБГРУНТУВАННЯ ВИБОРУ ДИСЦИПЛІНИ ОБСЛУГОВУВАННЯ ЗАЯВОК У РОЗПОДІЛЕНИХ СИСТЕМАХ ОБРОБКИ ІНФОРМАЦІЇ. *Таврійський науковий вісник. Серія: Технічні науки* [онлайн]. (6), 120–129. Access mode: doi: 10.32782/tnv-tech.2024.6.13
11. ГОРА, М. В. та ВОЛК, М. О., (2024). МОДЕЛІ УПРАВЛІННЯ РЕСУРАМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ПРОЦЕСУ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ. *Вісник Херсонського національного технічного університету* [онлайн]. (4(87)), 244–251. Access mode: doi: 10.35546/kntu2078-4481.2023.4.28.
12. Ismail, L. та Materwala, H., (2020). Computing Server Power Modeling in a Data Center. *ACM Computing Surveys* [онлайн]. 53(3), 1–34. Access mode: doi: 10.1145/3390605.
13. Venkat Marella, (2024). Comparative Analysis of Container Orchestration Platforms: Kubernetes vs. Docker Swarm. *International Journal of Scientific Research in Science and Technology* [онлайн]. 11(5), 526–543. Access mode: doi: 10.32628/ijrst24105254.
14. Turin, G., Borgarelli, A., Donetti, S., Damiani, F., Johnsen, E. B. та Tarifa, S. L. T., (2023). Predicting resource consumption of Kubernetes container systems using resource models. *Journal of Systems and Software* [онлайн]. 111750. Access mode: doi: 10.1016/j.jss.2023.111750.
15. ASI. (n.d.). Klastery serveriv. <https://a-si.com.ua/serverne-ta-shd-obladnannya/klastery-serveriv/>.
16. CGG. (n.d.). High-performance computing. CGG: Global Technology and HPC Leader. <https://www.cgg.com/digital-services/high-performance-computing>.
17. IBM. (n.d.). What is HPC? Introduction to high-performance computing. IBM. <https://www.ibm.com/topics/hpc>.
18. Chevance, R. J. (2005). *Server architectures: Multiprocessors, clusters, parallel systems, web servers, storage solutions*. Elsevier Digital Press.
19. Google Trends. (n.d.). Google trendy. <https://trends.google.com/trends/?hl=uk>.
20. FirstSiteGuide. (2023). Google search statistics (2023): 17+ facts you need to know. <https://firstsiteguide.com/google-search-stats/>.
21. Poltavshchyna. (2017). Sait NAPK ne vytrymuie navantazhennia – strok podannia bilshosti deklaratsii podovzheno do 1 travnia. Internet-vydannya "Poltavshchyna". <https://poltava.to/news/42339/>.
22. Ukrinform. (2016). Sait NAPK "vpav" cherez perevantazhennia. Ukrinform. <https://www.ukrinform.ua/rubric-technology/2108140-sajt-nazk-vpav-cerez-perevantazenna.html>.
23. Ukrainska pravda. (2016). Portal NAPK perevantazhenyi, vzhe podano 42 tysiachi deklaratsii. *Ukrainska pravda*. <https://www.pravda.com.ua/news/2016/10/26/7124898/>.
24. RBK-Ukraina. (n.d.). Novyny – ostanni novyny Ukrainy sohodni. <https://www.rbc.ua/>.
25. BBC News Ukraine. (2021). YePidtrymka i "tysiacha Zelenskoho" obvalyly Diiu. Shcho robyty? <https://www.bbc.com/ukrainian/news-59651413>.
26. Korohodskyi, Yu. (2023). U pershyi den roboty u prohrami "YeVidnovlennia" stavsia zbi. LB.ua. https://lb.ua/economics/2023/05/10/554605_pershiy_den_roboti_programi.html.
27. Facebook. (n.d.). Redirecting... <https://www.facebook.com/photo/?fbid=616272463874272&set=a.290337743134414>.
28. Ministerstvo rozvytku hromad, terytorii ta infrastruktury Ukrainy. (n.d.). Redirecting... <https://www.facebook.com/Ministry.for.restoration>.
29. Verkhovna Rada Ukrainy. (2018). Pro natsionalnu bezpeku Ukrainy: Zakon Ukrainy vid 21.06.2018 №2469-VIII. <https://zakon.rada.gov.ua/laws/show/2469-19>.
30. Fesokha, V., & Fesokha, N. (2021). Model nechitkoi avtentyfikatsii korystuvachiv informatsiinykh system orhaniv viiskovoho upravlinnia na osnovi povedinkovoi biometrii. *Zakhyst informatsii*, 23(2), 116–123.
31. Fesokha, V., Neroznak, Ye., and Sova, O. (2023). Udoskonalena model optymalnoho vykorystannia servernykh resursiv klasternoi systemy viiskovoho pryznachennia na osnovi rinvovahy Nesh. *Zbirnyk naukovykh prats Viiskovoho instytutu KNU imeni Tarasa Shevchenka*, 162–173. <https://doi.org/10.17721/2519-481X/2023/79-15>.
32. Fesokha, V., Neroznak, Ye., Sova, O., and Nesterov, O. (2023). Metod adaptivnoho balansuvannia navantazhennia v klasternykh systemakh viiskovoho pryznachennia na osnovi rinvovahy Nesh. *Zbirnyk naukovykh prats Tsentru voienno-stratehichnykh doslidzhen NUOU*, 3(76), 101–110. <https://doi.org/10.33099/2304-2745/2022-3-76/101-110>.
33. Neroznak, Ye. I., Merkotan, D. Yu., & Sova, O. Ya. (2021). Metody ta alhorytmy balansuvannia navantazhennia v klasternykh systemakh na osnovi elementiv shtuchnoho intelektu. I Mizhnarodna nauko-tekhnichna konferentsiia "Systemy i tekhnolohii zviazku, informatyzatsii ta kiberbezpeky: aktualni pytannia i tendentsii rozvytku". *VITI*, 215–217.

34. Kemp Technologies. (n.d.). Load balancing algorithms, types and techniques. <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques>.
35. ITC.ua. (n.d.). Vidmovostiikiyi klaster: krok do bezpeky vazhlyvykh danykh. https://itc.ua/articles/otkazoustojchivyj_klaster_shag_k_bezopasnosti_vazhnyh_danyh_23460.
36. Kliots, Yu., et al. (2019). Dynamichne balansuvannia trafiku mizh dekilкома providermy. Visnyk Khmelnytskoho natsionalnoho universytetu, 4(275), 62–67. <https://doi.org/10.31891/2307-5732-2019-275-4-62-67>.
37. Minailenko, R., et al. (2021). Overview of load balancing methods in cloud systems. National Interagency Scientific and Technical Collection of Works. Design, Production and Exploitation of Agricultural Machines, 51, 188–194. <https://doi.org/10.32515/2414-3820.2021.51.188-194>.
38. HashDork. (n.d.). Balansuvannia navantazhennia ta mikroservisy. <https://hashdork.com/uk/load-balancing-microservices/>.
39. HAProxy. (2023). HAProxy version 2.9-dev8 – Starter guide. HAProxy Documentation. <https://docs.haproxy.org/dev/intro.html>.
40. NGINX. (n.d.). Using NGINX as HTTP load balancer. https://nginx.org/en/docs/http/load_balancing.html.
41. PostgreSQL. (n.d.). PostgreSQL: Documentation. <https://www.postgresql.org/docs/>.
42. Chen, S.-L., Chen, Y.-Y., & Kuo, S.-H. (2017). CLB: A novel load balancing architecture and algorithm for cloud services. Computers & Electrical Engineering, 58, 154–160. <https://doi.org/10.1016/j.compeleceng.2016.01.029>.