

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження алгоритмів балансування навантаження
для розподілу трафіку у великих мережах»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело

(підпис)

Владислав ТКАЧУК
Ім'я ПРІЗВИЩЕ здобувача

Виконав:	здобувач вищої освіти гр. <u>КІД-42</u> <u>Владислав ТКАЧУК</u> Ім'я, ПРІЗВИЩЕ
Керівник: науковий ступінь, вчене звання	<u>Ігор БУЧЕНКО</u> Ім'я, ПРІЗВИЩЕ
Рецензент: науковий ступінь, вчене звання	_____ Ім'я, ПРІЗВИЩЕ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 Комп'ютерна інженерія
Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри Наталія
ЛАЩЕВСЬКА Ім'я, ПРІЗВИЩЕ
« » 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ткачук Владислав Олександрович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження алгоритмів балансування навантаження для розподілу трафіку у великих мережах

керівник кваліфікаційної роботи Ігор БУЧЕНКО,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «27» лютого 2024 р. № 36

2. Строк подання кваліфікаційної роботи «3» червня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

- 1) теза за дослідженням на тему: «Сучасні алгоритми балансування навантаження мережі»;
- 2) теза за дослідженням на тему: «Алгоритми балансування навантаження для розподілу трафіку у великих мережах».

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Питання балансування мереж за допомогою алгоритмів балансування.
2. Питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах.
3. Питання підходящих рішень алгоритмічного балансування великих мереж.

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	28.02-22.03.2024	Виконано
2	Обробка матеріалу	23.03-05.04.2024	Виконано
3	Розробка теоретичної частини	06.04-10.04.2024	Виконано
4	Дослідження проблемних питань	11.04-03.05.2024	Виконано
5	Висновки за результатами аналізу	04.05-08.05.2024	Виконано
6	Розробка презентації	09.05-12.05.2024	Виконано
7	Передзахист	13.05-17.05.2024	Виконано
8	Здача в деканат	23.05-01.06.2024	Виконано

Здобувач вищої освіти

(підпис)

Владислав ТКАЧУК

(ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Ігор БУЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 52 стор., 7 рис., 30 джерел.

Мета роботи – дослідження алгоритмів балансування навантаження для великих мереж на поточному етапі розвитку технологій.

Об'єкт дослідження – розподіл навантаження трафіку у великих мережах за допомогою алгоритмів балансування навантаження.

Предмет дослідження – алгоритми балансування навантаження для розподілу трафіку у великих мережах.

Короткий зміст роботи: Алгоритми балансування навантаження для розподілу трафіку у мережі на даний момент використовуються в багатьох галузях, для розподілення навантаження в мережі. Використання алгоритмів балансування навантаження зараз і в подальшому розвитку мереж дозволить покращити можливості підприємств що використовують такі алгоритми. Використання алгоритмів балансування навантаження мережі значно збільшить ефективність використання обчислювальних ресурсів, що також дозволить значно простіше розширювати та модифікувати мережу. Алгоритми балансування навантаження неодмінно мають велику кількість переваг, проте, існує й ряд недоліків даної технології, що можуть погіршити досвід користування такими рішеннями.

На основі проведеного дослідження було виявлено та проаналізовано чималу кількість рішень по балансуванню навантаження мережі, що дозволяє дійти точних висновків про використання алгоритмів балансування навантаження для розподілу трафіку у великих мережах.

КЛЮЧОВІ СЛОВА: АЛГОРИТМ БАЛАНСУВАННЯ, МЕРЕЖА, НАВАНТАЖЕННЯ, ТРАФІК, ТЕХНОЛОГІЯ, КОМБІНУВАННЯ, РОЗПОДІЛ, АНАЛІЗ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. СУЧАСНИЙ СТАН АЛГОРИТМІВ БАЛАНСУВАННЯ	11
1.1 Аналіз теорій та концепцій балансування навантаження мережі	11
1.2 Формулювання проблемних питань у сфері алгоритмів балансування навантаження трафіку	17
1.3 Методичні підходи для дослідження та аналізу підходящих алгоритмів балансування навантаження для розподілу трафіку у великих мережах	18
РОЗДІЛ 2. АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ.....	19
2.1 Аналіз та структуризація інформації на тему алгоритмів балансування навантаження мереж за використанням розроблених методичних підходів дослідження	19
2.2 Аналіз проблемних питань сфери алгоритмів балансування навантаження трафіку	37
2.3 Проблемне питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах	44
РОЗДІЛ 3. ВИЗНАЧЕННЯ ТА АНАЛІЗ ПІДХОДЯЩИХ РІШЕНЬ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ.....	45
3.1 Методи комбінування алгоритмів балансування навантаження мережі .	45
3.2 Рішення для балансування навантаження для розподілу трафіку у великих мережах	50
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	64

ВСТУП

Постійний розвиток та розростання областей застосування інформаційних технологій, зумовлюють й розвиток проблем, пов'язаних із управлінням та забезпеченням належного функціонування великих, складних комп'ютерних мереж. З відчутним технологічним стрибком в Україні, питання алгоритмів балансування навантаження у великих мережах постає все частіше. Тому, в цій роботі буде проведено дослідження найкращих рішень балансування навантаження в мережі та на основі вже проведених практичних досліджень іншими науковими авторами, аналітичним шляхом будуть визначені найбільш релевантні рішення для застосування у великих комп'ютерних мережах, включно із мережами в Україні. Аналізуючи дослідження та публікації, в яких започатковано розв'язання проблеми перевантаження мереж і залучення до них алгоритмічного балансування навантаження[1, 2, 3, 4, 5, 6], можна зазначити розглядання авторами різних концептуальних схем балансування навантаження мережі, таких як динамічне балансування та статичне балансування, також в цих дослідженнях були описані різні типи алгоритмів балансування навантаження мережі. Своїми дослідженнями, автори зробили внесок у розкриття проблеми балансування навантаження мережі, проте можна зазначити не повне розкриття частини проблеми, яка відноситься до сфери перевантаження великих комп'ютерних мереж.

Мета дослідження ґрунтується на визначенні та аналізі підходящих алгоритмів балансування навантаження для великих комп'ютерних мереж на поточному етапі розвитку інформаційних технологій. В процесі дослідження вирішувалися наступні завдання: пошук та аналіз інформації з відкритих джерел, щодо алгоритмів балансування навантаження для розподілу трафіку у великих мережах; пошук та аналіз досліджень, публікацій та робіт на тему алгоритмічного балансування трафіку мережі з метою його перерозподілу; форматування та структуризація проаналізованої інформації; побудова висновків на основі отриманої інформації;

Об'єктом дослідження є алгоритми балансування навантаження в комп'ютерній мережі. Алгоритми балансування навантаження – це методи розподілу робочого навантаження між різними ресурсами (серверами, вузлами мережі тощо) з метою забезпечення оптимальної продуктивності та надійності системи.

Основні характеристики включають:

- здатність алгоритму ефективно розподіляти навантаження, зменшуючи час відповіді та збільшуючи пропускну здатність системи;
- здатність адаптуватися до змін у навантаженні та непередбачуваних обставин, забезпечуючи безперервну роботу системи;
- здатність працювати ефективно при збільшенні обсягу ресурсів або навантаження;
- можливість розподілу навантаження між ресурсами на основі їх стану та доступності, мінімізуючи комунікаційні затримки;
- здатність інтегрувати алгоритми балансування без значних змін у існуючій інфраструктурі та додатках.

Предметом дослідження є конкретна частина проблеми алгоритмізації балансування трафіку в мережі, а саме: алгоритми балансування навантаження для розподілу трафіку у великих мережах.

Використаний метод дослідження – аналіз та виведення висновків на базі отриманої інформації. За допомогою такого методу дослідження було визначено не до кінця розкриті проблемні питання предмету дослідження. Під час проведення дослідження кваліфікаційної роботи, дістало подальший розвиток питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах.

Апробація результатів та публікації, які були опубліковані в ході проведення дослідження:

1. Бученко І. А., Ткачук В. О. СУЧАСНІ АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ МЕРЕЖІ. «Сучасний стан та перспективи розвитку IoT» : V міжнародна науково-технічна конференція, м. Київ, 18 квітня 2024 р. С. 1–2. URL:

https://duikt.edu.ua/ua/news-1-570-12409-v-mizhnarodna-naukovo-tehnichna-konferenciya-suchasniy-stan-ta-perspektivi-rozvitku-iot_kafedra-inzhenerii-programnogo-zabezpechennya-avtomatizovanih-sistem (дата звернення: 18.04.2024).

2. Бученко І. А., Ткачук В. О. АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ. «Діджиталізація науки як виклик сьогодення» : VI Міжнародна студентська конференція, м. Суми, 3 травня 2024 р. С. 1–3. URL: <https://archive.liga.science/index.php/conference-proceedings/issue/view/inter-03.05.2024> (дата звернення: 03.05.2024).

РОЗДІЛ 1. СУЧАСНИЙ СТАН АЛГОРИТМІВ БАЛАНСУВАННЯ

1.1 Аналіз теорій та концепцій балансування навантаження мережі

Балансування навантаження, це основний аспект в питанні оптимізації мережі, який відіграє ключову роль у забезпеченні ефективної та надійної передачі даних. На даний момент, жоден бізнес не може обійтися без використання послуг, що надають комп'ютерні мережі, тому балансування навантаження таких мереж є необхідністю. Загальна концепція балансування навантаження мережі полягає в рівномірному розподілу трафіку мережі, по певній кількості пристроїв, що обробляють вхідні запити, в певному порядку та за визначеними правилами, що в свою чергу призводить до оптимізації продуктивності мережі, збільшує можливості розширення та гнучкості.

Аналізуючи базову концепцію роботи системи балансування навантаження мережі, можна зазначити, що балансування навантаження працює за допомогою інтелектуального розподілення вхідного трафіку по всіх доступних ресурсах, таким способом, що забезпечує максимально продуктивне використання ресурсів пристроїв обробки та зменшення часу відповіді. Як зазначено в теорії балансування навантаження [1], така концепція забезпечує систему, в якій жоден ресурс обробки на який надходить трафік, не буде перевантаженим шляхом надходження кількості трафіку, що перевищує обчислювальні можливості пристрою. Таким чином можна зазначити, що така концепція балансування навантаження мережі, зменшує час простою пристроїв та збільшує надійність мережі в цілому.

Алгоритми балансування – це логіка або набір визначених правил, які використовуються для сортування трафіку за певними схемами. В основному, виділяють дві основні концептуальні схеми розподілу навантаження мережі [7]: статична та динамічна.

Статична схема балансування навантаження мереж, при розподіленні трафіку не враховує стан систем, показники рівня завантаженості процесорів тощо. Така схема рівномірно розподіляє трафік між пристроями що обчислюють його, на основі правил які не чутливі до стану системи, тому призначена для систем із дуже незначною варіацією навантаження. Впровадження алгоритмів статичного балансування навантаження вимагають глибоких знань про можливі ресурси обчислювального пристрою на момент впровадження, для кращого налаштування вхідного навантаження на процесор.

Динамічна схема балансування навантаження мереж вимагає зв'язку з обчислювальними ресурсами в реальному часі, оскільки вона визначає найбільш підходящий за певними критеріями обчислювальний пристрій і надає йому перевагу. Схема динамічного балансування навантаження контролює трафік на основі поточного стану системи та перерозподіляє його із завантажених пристроїв на найменш завантажені, працездатні обчислювальні пристрої в режимі реального часу, запобігаючи перенавантаження мережі та зменшує час відповіді.

Відмінності між статичною та динамічною схемою балансування навантаження:

- методи статичного балансування навантаження розподіляють трафік без коригування поточного стану системи або серверів. Деякі статичні алгоритми надсилають рівну кількість трафіку, у визначеному порядку або випадковим чином, на кожен сервер у групі. Методи динамічного балансування навантаження враховують поточний стан системи та кожного сервера й базують розподіл трафіку на цих факторах;

- методи статичного балансування навантаження не враховує стан системи під час розподілу завдань. Натомість розподіл формується на основі припущень і відомостей про загальну систему, створених до початку сортування. Це включає такі відомі дані, як кількість процесорів, швидкість і потужність зв'язку, а також припущення, такі як вимоги до ресурсів, час відповіді та час надходження вхідних завдань;

– методи статичного балансування навантаження в розподілених системах мінімізують конкретні функції продуктивності, зв'язуючи відомий набір завдань із доступними процесорами. Ці типи стратегій балансування навантаження зазвичай зосереджені навколо маршрутизатора, який оптимізує функцію продуктивності та розподіляє навантаження. Перевага статичного балансування навантаження в розподілених системах полягає в простоті використання та швидкому, простому розгортанні, при цьому, існує перелік певних обмежень для яких такий алгоритм не підходить;

– динамічні методи розподілу навантаження враховують поточне навантаження кожного вузла або обчислювального блоку в системі, досягаючи швидшої обробки шляхом динамічного переміщення завдань від вузлів, які перевантажені, до вузлів, які недостатньо завантажені. Динамічні алгоритми набагато складніші для керування, але в ситуації, коли час виконання для різних завдань сильно відрізняється, такі методи можуть дати чудові результати. Крім того, оскільки немає потреби виділяти певні вузли для розподілу роботи, архітектура динамічного балансування навантаження часто є більш модульною;

– унікальне призначення завдань передбачає завдання, які призначаються процесору на основі його стану в певний момент часу. Динамічне призначення стосується постійного перерозподілу завдань на основі стану системи та її розвитку.

Очевидно, що будь-який алгоритм балансування навантаження може фактично уповільнити загальний процес, якщо він вимагає надмірного спілкування для прийняття рішень. Природа завдань має великий вплив на ефективність алгоритмів балансування навантаження, тому максимізація доступу до інформації про завдання під час прийняття рішень алгоритмом збільшує потенціал оптимізації. Як динамічна так і статичні алгоритми балансування навантаження також формуються при наборі інших факторів, таких як час виконання завдання, розмір завдання, можливість задачі бути розбитою на частини, для подальшого розподілу по обчислювальних пристроях, рід завдання, наприклад, запит вебсторінки. При розрахунках розподілу навантаження можуть враховуватися розміри завдання, знаючи розмір завдання, певні алгоритми можуть дізнатися примірний час

виконання завдання, точне визначення часу виконання завдання надзвичайно складно обрахувати, але це дозволить оптимально розподілити навантаження. Існує кілька способів оцінки різного часу виконання. Якщо завдання мають однаковий розмір, середній час виконання може бути успішно використаний. Однак, коли час виконання дуже нерегулярний, потрібні більш складні методи. Наприклад, можна додавати метадані до завдань, а потім робити висновки для майбутніх завдань на основі статистики, яка була отримана на основі попереднього часу виконання для подібних метаданих. Завдання можуть залежати одне від одного, тому деякі завдання не можуть початися, доки не будуть завершені попередні. В теорії, під час виникнення таких випадків можливо використати ациклічний графік взаємозалежностей для мінімізації часу виконання задачі шляхом оптимізації порядку завдань. Деякі алгоритми можуть використовувати метаевристичні методи для розрахунку оптимального розподілу таких завдань. Розподіл завдань на підзадачі під час виконання розподілу навантаження є дуже важливою специфікою при розробці алгоритму балансування навантаження мережі, через великий потенціал зменшення навантаження на певний обчислювальний пристрій.

При дослідженні схем динамічного балансування навантаження виділяють три задачі, які має виконувати схема та які стоять перед розробником схеми [8]:

1. збір інформації поточного стану;
2. визначення потенційних пристроїв обробки для передачі на них трафіку з перевантаженого вузла;
3. передача навантаження з перевантаженого вузла з урахуванням балансування.

Виділяють чотири основні підходи до проектування схеми балансування мережі:

– *централізована політика*. Такий підхід передбачає використання одного центрального обчислювального пристрою, який приймає рішення про пристрої джерела та адресату, йому повідомляється системні значення завантаження для визначення найбільш та найменш завантажених комп'ютерів, та ініціювання передачі навантаження між ними. Недоліком централізованого підходу вважається

значне збільшення трафіку, що передається мережею, за рахунок широкомовних запитів, які визначають стани обчислювальних пристроїв, в наслідок цього, можуть виникати перевантаження в мережі або зменшення продуктивності;

– *перерозподіл за ініціативи відправника*. Такий підхід передбачає, що відправник навантаження вже відомий, а завданням є визначення комп'ютера-адресата, який буде приймати участь у перерозподілі та візьме на себе частину навантаження;

– *перерозподіл за ініціативи отримувача*. Такий підхід передбачає, що обчислювальний пристрій, вирішує приймати участь у розподілі навантаження, якщо його навантаження менше за граничний поріг, в такому випадку, політика місця знаходження використовується для того, щоб знайти комп'ютер, від якого може бути передане навантаження для розподілення. В таких випадках вибір місця розташування повністю залежить від механізму комунікацій, а політика повинна вибрати найбільш завантажений, або в інших випадках, випадковий комп'ютер;

– *симетрична ініціалізація*. Такий підхід передбачає, що політика ініціалізованого відправника більш ефективна, в той час, коли системне навантаження середнє, а політика ініціалізованого отримувача - коли навантаження високе.

Згідно з дослідженням [9], рішення які можуть використовуватися при проектуванні алгоритмів балансування навантаження, мають ряд недоліків:

– *однопорогові алгоритми*, які використовуються для визначення ступеню завантаженості вузла, у більшості випадків не забезпечують достатню стійкість процедури балансування навантаження. Наприклад, у випадках, коли завантаження пари вузлів, які приймають участь у балансуванні, близьке до порогового, за час від моменту прийняття рішення про передачу навантаження до моменту початку його передачі, рівень завантаження може змінитися, внаслідок чого мета балансування не буде досягатися. Для зменшення нестабільності був запропонований двопороговий алгоритм [10], який ділить рівень завантаження обчислювального пристрою на три частини – мале, середнє та високе завантаження. Однак, означений алгоритм використовує фіксовані наперед визначені значення порогових рівнів, в

той час як бажаною рисою сучасних розподілених систем повинна бути можливість динамічної зміни стану вузлів і трафіку;

- будь-яка політика вибору місця розташування передбачає попереднє опитування стану вузлів, які розглядаються алгоритмом балансування навантаження як вузли-кандидати на прийом чи передавання навантаження. В мережах, які мають велике число вузлів, наслідком може бути суттєве збільшення мережного трафіку, що зменшує продуктивність обчислень. Окрім цього, якщо використовуються алгоритми, які передбачають визначений час очікування відповідей від вузлів, то в умовах великого напливу трафіку, балансування навантаження може виявитись взагалі неможливим, оскільки відповіді від потенційних вузлів-кандидатів прийдуть запізно. Деякі алгоритми передбачають управління перевантаженням, але при цьому вони також використовують наперед визначені порогові рівні;

- для прийняття рішення про необхідність розпочати процедуру балансування використовують локальну інформацію про рівень завантаження лише «найближчих» вузлів. Це штучно обмежує множину вузлів-кандидатів на передавання навантаження з вузла.

В запропонованому рішенні, в дослідженні [11], науковий автор запевняє про вирішення наступних проблем класичних систем динамічного балансування:

- розширений пошук можливих вузлів-кандидатів на передачу навантаження без суттєвого збільшення трафіку в мережі за рахунок використання при пошуку вузлів-кандидатів на передавання навантаження ієрархічної доменної моделі мережі;

- підвищення стійкості рішень щодо розподілу навантаження за рахунок використання фаззі-алгоритму вибору кандидатів;

- керування мережним трафіком та запобігання колізіям за рахунок зміни параметрів функції належності нечітких змінних.

Виходячи з цього дослідження, можна зауважити широке розкриття питання перенасичення мережі службовим трафіком, який використовується для визначення станів обчислювальних пристроїв. Використання фаззі-логіки дозволяє підвищити

показники якості балансування навантаження трафіку в мережі та дозволяє запобігти недоліків традиційних алгоритмів динамічного балансування навантаження.

1.2 Формулювання проблемних питань у сфері алгоритмів балансування навантаження трафіку

Алгоритми балансування навантаження в мережах є критично важливими для забезпечення ефективного та надійного розподілу трафіку між серверами та іншими мережевими ресурсами. З масштабним зростанням мереж та об'ємів трафіку виникають нові виклики, такі як потреба у вищій швидкості обробки та масштабуванні, щоб забезпечити високий рівень продуктивності та доступності. Дослідження та вдосконалення алгоритмів балансування навантаження є ключовим елементом для оптимізації роботи мереж у сучасному інтернеті.

Формулювання проблемних питань та їх подальше розкриття та вирішення, лежить в основі розуміння предмету дослідження. Під час дослідження та аналізу сучасного стану наукової думки, теорій та концепцій алгоритмів балансування навантаження трафіку, було сформовано для розкриття, наступні проблемні питання:

1. основні критерії ефективності алгоритмів балансування навантаження у великих мережах;
2. вплив географічного розташування обчислювальних пристроїв на ефективність роботи алгоритмів балансування навантаження;
3. стратегії масштабування, які можуть бути використані для адаптації алгоритмів балансування навантаження на фоні зростаючого обсягу трафіку у мережах;
4. вимоги до швидкодії та надійності, які необхідно враховувати при розробці алгоритмів балансування навантаження для великих мереж;

5. виклики, що виникають при застосуванні алгоритмів балансування навантаження в хмарних середовищах та шляхи їх вирішення.

1.3 Методичні підходи для дослідження та аналізу підходящих алгоритмів балансування навантаження для розподілу трафіку у великих мережах

Методичні підходи для дослідження алгоритмів балансування навантаження для розподілу трафіку великих мережах – це підходи, методи та стратегії, які використовуються для вивчення та аналізу ефективності різних алгоритмів балансування навантаження у мережах. Ці підходи дозволяють здійснювати об'єктивну оцінку та порівняння алгоритмів, визначати їх переваги та недоліки, а також розробляти нові алгоритми з урахуванням виявлених особливостей.

Можна виділити наступні методичні підходи для поточного дослідження, які можна використати:

- *аналіз даних*. Використання методів аналізу інформації, з метою обробки даних та виведення об'єктивних висновків щодо ефективності та надійності алгоритмів;

- *порівняльний аналіз*. Використання методів порівняння та оцінки переваг і недоліків кожного з алгоритмів, з метою виявлення найбільш підходящих алгоритмів балансування навантаження для розподілу трафіку у великих мережах, в тих чи інших умовах.

РОЗДІЛ 2. АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ

2.1 Аналіз та структуризація інформації на тему алгоритмів балансування навантаження мереж за використанням розроблених методичних підходів дослідження

Аналізуючи дослідження, публікації та інші типи джерел інформації на тему алгоритмізації балансування навантаження мережі та алгоритмів навантаження загалом [8, 10, 12, 13, 14, 15, 16], виявлено значну кількість аспектів та проблем на які варто звернути увагу та дослідити. Існує певна кількість алгоритмів балансування навантаження мережі які працюють за різними схемами та правилами. З метою визначення підходящих алгоритмів балансування навантаження мережі для розподілу трафіку у великих мережах, найкращим способом буде використання методики аналізу та порівняння алгоритмів.

В основному алгоритми балансування навантаження мережі для регулювання трафіку, поділяють на статичні та динамічні типи, існує ще адаптивний тип алгоритмів балансування навантаження, але він застосовується не так часто в силу того, що алгоритми динамічного типу перекривають потреби поставлені адаптивним алгоритмам, одна з небагатьох переваг адаптивних алгоритмів перед динамічними, це більш простий спосіб інтегрування та налаштування.

Статичні алгоритми балансування навантаження мережі [17]. За допомогою статичного балансування навантаження мультимедійний сервер намагається збалансувати навантаження на диски, вибираючи для кожного медіа-потoku відповідний розмір блоку смуги, ступінь смуги та кількість реплікації. Звичайні файлові системи вибирають розміри смугових одиниць, які мінімізують середній час відгуку та максимізують пропускну здатність. З іншої сторони, щоб зменшити частоту перерв у відтворенні, мультимедійний сервер повинен вибрати розмір

блоку смуги, який мінімізує дисперсію в часі відгуку, максимізуючи пропускну здатність. У той час як невеликі блоки смуг призводять до рівномірного розподілу навантаження між дисками в масиві (і тим самим зменшують дисперсію в часі відповіді), вони також збільшують накладні витрати на пошук диска та затримки обертання (і тим самим зменшують пропускну здатність). З іншого боку, великі смугові блоки збільшують пропускну здатність масиву за рахунок збільшення дисбалансу навантаження та дисперсії часу відгуку. Щоб максимізувати кількість клієнтів, які можуть обслуговуватися одночасно, сервер повинен вибрати розмір блоку смуги, який збалансує ці компроміси. Ступінь смуги для медіа-потоків залежить від кількості дисків у масиві. У відносно невеликих дискових масивах мультимедійні потоки з роздільною смугою поширюються на всі диски в масиві (широка смуга) дає збалансоване навантаження та максимізує пропускну здатність. Однак для великих дискових масивів, щоб максимізувати пропускну здатність, серверу може знадобитися розділити мультимедійні потоки на підмножини дисків у масиві та відтворити їхнє сховище для досягнення балансування навантаження. Обсяг реплікації для кожного медіа-потoku залежить від популярності потоку та обмежень загального обсягу пам'яті.

До статичних алгоритмів балансування навантаження мережі належать такі типи алгоритмів як:

- Round Robin;
- Sticky Round Robin;
- Weighted Round Robin;
- Source IP Hash;
- URL Hash;
- Randomized Algorithm;
- Opportunistic Load Balancing;
- Min-Min Load balancing;
- Min-Max Load Balancing.

Алгоритм балансування навантаження Round Robin – це метод розподілу трафіку між серверами або ресурсами у мережі. Його основна ідея полягає в тому,

щоб рівномірно розподілити навантаження між усіма доступними серверами, обслуговуючи запити послідовно і циклічно.

Особливості роботи такого алгоритму:

- ініціюється список серверів, або ресурсів, які можуть бути використані для обробки запитів на перенаправлення навантаження. Це може бути група фізичних серверів, віртуальних машин або будь-яких інших обчислювальних ресурсів;

- коли приходить новий запит, алгоритм Round Robin, проводить розподіл, обирає сервер за певним порядком. Кожен наступний запит буде обслуговуватися наступним сервером у списку. Коли досягається останній сервер у списку, процес повертається до першого сервера і продовжується циклічно;

- один і той самий сервер не буде обслуговувати всі запити, оскільки алгоритм буде переходити до наступного сервера після кожного розподілу трафіку. Це допомагає рівномірно розподілити навантаження між усіма серверами зі списку доступних обчислювальних ресурсів;

- при умові, що алгоритм Round Robin допомагає рівномірно розподіляти трафік, він не враховує поточне навантаження на кожному сервері зі списку. Таким чином, може виникнути ситуація, коли один сервер намагається обслужити більше запитів, ніж інші, що може призвести до перевантаження. Щоб запобігти цьому, зазвичай використовуються інші алгоритми балансування навантаження, які враховують актуальне навантаження на серверах;

- Round Robin простий в реалізації та інтегруванні в мережу. Він легко впроваджується і не потребує складної конфігурації.

Алгоритм Round Robin є ефективним способом розподілу трафіку у мережі, особливо коли сервери мають подібні характеристики та можливості обробки. Однак цей алгоритм не враховує поточний стан навантаження обчислювальних пристроїв, через це для оптимізації роботи мережі може знадобитися комбінація різних алгоритмів балансування навантаження, тому такий алгоритм погано підходить для розподілу навантаження трафіку у великих мережах.

Алгоритм балансування навантаження Sticky Round Robin є розширенням звичайного алгоритму Round Robin, яке дозволяє "приклеювати" або

"закріплювати" клієнтів до певних серверів. Цей підхід використовується для забезпечення стабільності підключення для клієнтів, але при цьому зберігає базову ідею рівномірного розподілу навантаження між серверами.

Особливості роботи такого алгоритму:

- при першому запиті на розподіл навантаження від клієнта, алгоритм Round Robin обирає сервер за стандартною процедурою - послідовно і циклічно;
- після обрання сервера для першого запиту клієнта, сервер відправляє клієнту ідентифікатор (зазвичай у вигляді cookie або session ID), який дозволяє клієнту залишатися "приклеєним" до того ж сервера для наступних запитів;
- клієнт зберігає ідентифікатор отриманий від сервера і використовує його при наступних запитах. Це може бути зроблено шляхом збереження cookie у веббраузері клієнта або передачею ідентифікатора у заголовку запиту на розподіл трафіку;
- при наступних запитах клієнт включає ідентифікатор, отриманий від сервера, у свої запити. Алгоритм балансування навантаження мережі використовує цей ідентифікатор, щоб визначити, до якого сервера "приклеєний" клієнт, і направляє запит на цей сервер;
- ідентифікатор може бути оновлений з часом, наприклад, при завершенні сесії або після певного періоду неактивності клієнта. При цьому, при повторному запиті зі сторони клієнта, він буде знову прикріплений до іншого сервера за допомогою алгоритму Round Robin.

Перевагою алгоритму Sticky Round Robin є збереження стабільності з'єднання для клієнтів, оскільки вони залишаються прикріпленими до того ж сервера протягом певного часу. Це особливо важливо для деяких застосунків, де стан сесії клієнта має значення (наприклад, вебсайти, що вимагають автентифікації). Однак варто зазначити, що прикріплення клієнтів до певних серверів може призвести до нерівномірного розподілу навантаження, особливо якщо деякі сервери мають більше "приклеєних" клієнтів, ніж інші. Тому важливо належним чином налаштувати параметри алгоритму для забезпечення ефективного балансування навантаження.

Алгоритм балансування навантаження Weighted Round Robin є вдосконаленою версією стандартного алгоритму Round Robin, яка дозволяє призначати різні пріоритети для кожного сервера у списку. Цей підхід дозволяє ефективніше розподіляти трафік, надаючи більший пріоритет більш потужним або швидкодіючим серверам.

Особливості роботи такого алгоритму:

- для кожного сервера у списку визначаються пріоритети, що вказують на його потужність або пропускну здатність. Наприклад, наявності трьох обчислювальних сервери у списку, можна припустити, що перший сервер має пріоритет 3, другий-2, а третій-1. Чим вищий пріоритет, тим більше трафіку буде виділено цьому серверу;

- коли надходить новий запит, алгоритм Weighted Round Robin вибирає сервер для обробки, враховуючи їх пріоритети. Це може бути зроблено шляхом проходження через список серверів і вибору сервера з урахуванням його пріоритету;

- після обробки запиту алгоритм повертається до початку списку серверів і продовжує циклічно, враховуючи пріоритети кожного сервера при виборі;

- сервери з більшими пріоритетами отримують більше трафіку, тоді як сервери з меншими пріоритетами отримують менше трафіку. Це дозволяє ефективніше використовувати ресурси та забезпечує рівномірне розподілення навантаження між серверами.

Алгоритм Weighted Round Robin є корисним інструментом для оптимізації балансування навантаження, оскільки дозволяє виділяти більше навантаження потужним серверам та менше - менш потужним. Це особливо важливо в ситуаціях, коли сервери мають різні характеристики або пропускну здатність. Проте важливо враховувати, що встановлення неправильних пріоритетів може призвести до нерівномірного розподілу навантаження або перевантаження окремих серверів. Тому важливо належним чином налаштувати пріоритети на основі реальних потреб та характеристик серверів.

Алгоритм балансування навантаження за допомогою Source IP/URL Hash – це алгоритм, в якому рішення про те, на який сервер буде направлений запит, визначається на основі IP/URL-адреси клієнта. Цей алгоритм ставить у відповідність кожен унікальний IP/URL-адресу зі своїм власним хеш-значенням серверу. Такий підхід дозволяє забезпечити стабільність маршрутизації запитів в мережі та підтримувати послідовність обробки запитів з однієї IP/URL-адреси на один і той же сервер.

Особливості роботи такого алгоритму:

- коли клієнт надсилає запит на сервер, алгоритм балансування навантаження мережі отримує IP/URL-адресу цього клієнта;
- за допомогою функцій хешування, такі як MD5, SHA-1, SHA-256 тощо, виконується обчислення хеш-значення для IP/URL-адреси клієнта;
- згенероване хеш-значення IP/URL-адреси клієнта використовується для визначення сервера, на який буде направлено запит. Часто використовується розподіл хеш-значень за модулем на кількість доступних серверів, щоб вибрати конкретний сервер;
- запит від клієнта буде направлений на сервер, вибраний на основі хеш-значення його IP/URL-адреси;
- оскільки хеш-значення IP/URL-адреси клієнта завжди буде однаковим для одного клієнта, то всі його запити будуть направлені на той самий сервер. Це забезпечує стабільність маршрутизації трафіку для клієнтів з однаковою IP/URL-адресою;
- запити від різних клієнтів будуть розподілені між різними серверами в залежності від їхніх IP/URL-адрес і згенерованих хеш-значень. Це дозволяє рівномірно розподілити навантаження між доступними серверами;
- оскільки вибір сервера здійснюється на основі IP/URL-адреси клієнта, яка є сталою для кожного клієнта, цей метод балансування навантаження є стійким до змін у мережі.

Алгоритм балансування навантаження Source IP/URL Hash є ефективним методом розподілу трафіку у мережі, особливо коли важлива стабільність

маршрутизації запитів для клієнтів з однаковою IP/URL-адресою. Цей метод може бути корисним, наприклад, при роботі з вебсерверами, де важливо розподіляти трафік між різними вебсторінками або ресурсами.

Алгоритм балансування навантаження Randomized є методом, при якому вибір сервера для обробки запиту здійснюється випадковим чином. Це означає, що кожен новий запит розподіляється між доступними серверами без якої-небудь конкретної системи або логіки. Цей підхід може бути корисним, коли немає необхідності враховувати характеристики або стан серверів, або коли просто потрібно швидко розподілити трафік між серверами без додаткового обчислення.

Особливості роботи такого алгоритму:

- коли приходить новий запит, алгоритм балансування навантаження мережі вибирає сервер для обробки випадковим чином зі списку доступних серверів. Це може бути зроблено одним із способів, наприклад, за допомогою функції генерування випадкових чисел або використанням вбудованої функціональності в програмне забезпечення балансувальника;

- отримавши випадково обраний сервер, алгоритм балансування навантаження мережі направляє запит на цей сервер для подальшої обробки;

- при належному функціонуванні, такий алгоритм дозволяє відносно рівномірно розподілити навантаження між доступними серверами, оскільки кожен новий запит розглядається як окремий випадок;

- алгоритм балансування Randomized не враховує стану або навантаження на серверах, тому існує ризик того, що деякі сервери можуть отримувати більше запитів, ніж інші. Це може призвести до нерівномірного розподілу навантаження, особливо якщо сервери мають різні характеристики або можливості обробки;

- однією з переваг алгоритму Randomized є його простота реалізації. Він не вимагає складної логіки або обчислень, що дозволяє швидко впроваджувати його в системі балансування навантаження;

- алгоритм Randomized може бути корисним у випадках, коли сервери мають подібні характеристики та обслуговують статичний трафік, або коли важливо швидко розподілити трафік без додаткових обчислень.

У цілому, алгоритм балансування навантаження Randomized є простим та ефективним способом розподілу трафіку між серверами, але він може не бути оптимальним у випадках, коли потрібно враховувати стан та характеристики серверів.

Алгоритм балансування навантаження Opportunistic – це алгоритм, в якому рішення про те, який сервер обробляє запит, приймається на основі найбільш доступного та швидкодіючого сервера в момент отримання запиту. Замість того, щоб здійснювати вибір на основі попередніх або передбачуваних характеристик серверів, цей підхід опирається на поточні умови мережі та навантаження серверів у реальному часі.

Особливості роботи такого алгоритму:

- при отриманні нового запиту мережевий балансувальник оцінює доступність серверів. Це може включати аналіз пінгуння до серверів, перевірку статусу серверів або моніторинг їхньої відповідності;

- після оцінки доступності серверів балансувальник вибирає найбільш доступний сервер у мережі. Це може бути сервер з найменшою кількістю відправлених запитів, найменшою затримкою або іншою метрикою, що вказує на його доступність та продуктивність;

- отримавши інформацію про найбільш доступний сервер, мережевий балансувальник направляє запит на цей сервер для подальшої обробки;

- під час роботи алгоритму Opportunistic, сервери можуть динамічно змінювати свою доступність та продуктивність. Тому балансувальник повинен постійно оновлювати свої вибори, щоб максимізувати ефективність обробки запитів;

- оскільки алгоритм Opportunistic приймає рішення на основі поточних умов мережі та серверів, він є дуже адаптивним до змін у навантаженні або доступності серверів. Це дозволяє швидко реагувати на зміни у мережі та мінімізувати можливість перевантаження або відмов серверів;

- алгоритм Opportunistic дозволяє ефективно використовувати ресурси серверів, направляючи запити на найбільш доступний та продуктивний сервер у мережі у кожний момент часу;

- у випадку, коли сервери розташовані у різних географічних регіонах, алгоритм Opportunistic допомагає вибирати сервери на основі їхньої доступності та швидкодії в кожному регіоні.

Алгоритм балансування навантаження мережі Opportunistic, є потужним інструментом для оптимізації роботи мережевих систем, оскільки він дозволяє ефективно розподіляти навантаження, враховуючи поточні умови мережі та стан серверів. Проте у великих мережах такий алгоритм балансування навантаження мережі, за рахунок моніторингу стану та рівня завантаженості обчислювальних машин пінгуванням, може призвести до суттєвого збільшення навантаження трафіку на мережу.

Алгоритм балансування навантаження Min-Min є стратегією, яка вибирає найменш завантажений сервер для обробки кожного нового запиту. Головною метою цього алгоритму є мінімізація часу очікування на обслуговування запиту шляхом вибору сервера, який має найменше навантаження у даний момент.

Особливості роботи такого алгоритму:

- кожен сервер періодично вимірює своє поточне навантаження. Це може включати кількість активних запитів, використання процесорного часу, обсяги вхідного/вихідного трафіку та інші метрики;

- при отриманні нового запиту алгоритм балансування навантаження мережі аналізує навантаження всіх доступних серверів і вибирає той, який має найменше навантаження в даний момент;

- після вибору найменш завантаженого сервера алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту на сервері, його стан навантаження оновлюється для врахування нових запитів, які можуть надходити в майбутньому;

– цей процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи мінімізацію часу очікування та ефективне використання ресурсів серверів.

Головна перевага алгоритму Min-Min полягає в тому, що він спрямований на мінімізацію часу очікування для кожного клієнта, вибираючи найшвидший доступний сервер для обробки кожного запиту. Проте цей підхід може не завжди оптимально враховувати динамічні зміни у навантаженні та вимогах системи. Також важливо враховувати, що мінімізація навантаження на окремих серверах може призвести до нерівномірного розподілу навантаження та неефективного використання ресурсів мережі.

Алгоритм балансування навантаження Min-Max – це стратегія, яка вибирає сервер з найменшим максимальним навантаженням для обробки кожного нового запиту. Головною метою цього алгоритму є мінімізація максимального навантаження на будь-якому з серверів, що допомагає попередити перевантаження та забезпечити більш рівномірний розподіл навантаження у мережі.

Особливості роботи такого алгоритму:

– кожен сервер періодично вимірює своє поточне навантаження. Це може включати кількість активних запитів, використання процесорного часу, обсяги вхідного чи вихідного трафіку та інші метрики;

– при отриманні нового запиту алгоритм балансування навантаження мережі аналізує максимальне навантаження на кожному з доступних серверів і вибирає той, який має найменше максимальне навантаження;

– після вибору сервера з найменшим максимальним навантаженням алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

– після обробки запиту на сервері, його стан навантаження оновлюється для врахування нових запитів, які можуть надходити в майбутньому.

Цей процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи мінімізацію максимального навантаження на будь-якому з серверів та рівномірний розподіл навантаження.

Головна перевага алгоритму Min-Max полягає в тому, що він спрямований на запобігання перевантаження будь-якого з серверів, що допомагає забезпечити стабільну та надійну роботу мережі. Однак цей підхід може призвести до того, що деякі сервери будуть менш завантажені, ніж інші, що може привести до неефективного використання ресурсів мережі.

До динамічних алгоритмів балансування навантаження мережі належать такі типи алгоритмів як:

- Least Connection;
- Weighted Least Connection;
- Least Response Time;
- Weighted Response Time;
- Resource-based;
- Biased Random sampling method;
- Throttled method;
- Least Bandwidth.

Алгоритм балансування навантаження Least Connection - це стратегія, при якій кожен новий запит направляється на сервер з найменшою кількістю активних з'єднань. Головною метою цього алгоритму є розподіл навантаження у мережі шляхом направлення запитів на сервери з найменшим навантаженням, що допомагає уникнути перевантаження окремих серверів та забезпечити оптимальне використання ресурсів.

Особливості роботи такого алгоритму:

- кожен сервер періодично вимірює кількість активних з'єднань. Це може включати TCP-з'єднання, HTTP-сесії, сеанси користувачів або інші форми активних з'єднань, залежно від протоколу та характеристик сервера;
- при отриманні нового запиту алгоритм балансування навантаження мережі аналізує кількість активних з'єднань на кожному з доступних серверів і вибирає той, який має найменшу кількість з'єднань;
- після вибору сервера з найменшою кількістю з'єднань алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту на сервері, його стан навантаження оновлюється для врахування нових запитів, які можуть надходити в майбутньому;
- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи мінімізацію кількості з'єднань на кожному з серверів та рівномірний розподіл навантаження.

Головна перевага алгоритму балансування навантаження Least Connection полягає в тому, що він спрямований на розподіл навантаження, враховуючи кількість активних з'єднань на кожному сервері. Це дозволяє уникнути перевантаження окремих серверів та забезпечити стабільну та ефективну роботу мережі.

Алгоритм балансування навантаження Weighted Least Connection є розширенням стратегії балансування навантаження Least Connection, де кожному серверу присвоюється пріоритет, що відображає його здатність обробляти запити. Пріоритет визначається на основі різних параметрів, таких як потужність сервера, його швидкодія та пропускна здатність. Головною метою цього алгоритму є ефективне розподілення навантаження у мережі, враховуючи пріоритет кожного сервера.

Особливості роботи такого алгоритму:

- кожному серверу у мережі призначається пріоритет, який відображає його здатність обробляти запити. Це може бути числове значення, що відображає рівень обчислювальних ресурсів, швидкодію або пропускну здатність сервера;
- кожен сервер періодично вимірює кількість активних з'єднань. Це може включати TCP-з'єднання, HTTP-сесії, сеанси користувачів або інші форми активних з'єднань, залежно від протоколу та характеристик сервера;
- при отриманні нового запиту алгоритм балансування навантаження мережі аналізує кількість активних з'єднань на кожному з доступних серверів та їх пріоритет, і вибирає той сервер, який має найменший пріоритет навантаження;
- після вибору сервера з найменшим пріоритетом навантаження, алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту на сервері, його стан навантаження оновлюється для врахування нових запитів, які можуть надходити в майбутньому;
- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи ефективне використання ресурсів кожного з серверів та рівномірний розподіл навантаження.

Алгоритм балансування навантаження *Weighted Least Connection* дозволяє ефективно розподіляти навантаження у мережі, враховуючи не лише кількість активних з'єднань на кожному сервері, але й пріоритет серверів. Це допомагає забезпечити оптимальне використання ресурсів мережі та підвищити її ефективність.

Алгоритм балансування навантаження *Least Response Time* – це стратегія, при якій кожен новий запит направляється на сервер, який має найменший час відповіді на запити. Головною метою цього алгоритму є мінімізація часу очікування для клієнтів, направляючи їхні запити на сервери з найшвидшим часом відгуку, що допомагає покращити загальну продуктивність та швидкодію мережі.

Особливості роботи такого алгоритму:

- кожен сервер періодично вимірює свій час відгуку на запити. Це може включати час обробки запиту, час передачі даних та будь-які затримки, пов'язані з мережею;
- при отриманні нового запиту, алгоритм балансування навантаження мережі, аналізує час відгуку на кожному з доступних серверів і вибирає той, який має найменший час відгуку;
- після вибору сервера з найменшим часом відгуку алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;
- після обробки запиту на сервері, його час відгуку оновлюється для врахування нових запитів, які можуть надходити в майбутньому;
- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи мінімізацію часу очікування для клієнтів та покращення загальної продуктивності мережі.

Головна перевага алгоритму балансування навантаження Least Response Time полягає в тому, що він спрямований на максимальне покращення швидкодії та ефективності мережі шляхом направлення запитів на сервери з найшвидшим часом відгуку. Це дозволяє забезпечити користувачам максимально швидку реакцію на їхні запити та забезпечити оптимальне використання ресурсів мережі.

Алгоритм балансування навантаження Weighted Response Time – це стратегія, при якій кожному серверу мережі призначається пріоритет, що відображає його пропускну здатність та час відповіді на запити. При отриманні нового запиту алгоритм балансування навантаження мережі аналізує пріоритет кожного сервера та час відповіді на запити і вибирає той сервер, який має найменше значення зваженого часу відповіді. Головною метою цього алгоритму є мінімізація часу очікування для клієнтів та оптимальне використання ресурсів серверів мережі.

Особливості роботи такого алгоритму:

- кожному серверу у мережі призначається пріоритет, який відображає його пропускну здатність та інші характеристики, які впливають на час відповіді на запити;
- кожен сервер періодично вимірює час відповіді на свої запити. Це може включати час обробки запиту, час передачі даних та будь-які затримки, пов'язані з мережею;
- для кожного сервера обчислюється зважений час відповіді, який дорівнює коефіцієнту пріоритету на середній час відповіді на запити за певний період часу;
- при отриманні нового запиту алгоритм балансування навантаження мережі аналізує зважений час відповіді на кожному з доступних серверів і вибирає той, який має найменший зважений час відповіді;
- після вибору сервера з найменшим зваженим часом відповіді алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;
- після обробки запиту на сервері, його час відповіді оновлюється для врахування нових запитів, які можуть надходити в майбутньому;

- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи мінімізацію часу очікування для клієнтів та оптимальне використання ресурсів мережі.

Алгоритм балансування навантаження Weighted Response Time дозволяє ефективно розподіляти навантаження у мережі, враховуючи як час відповіді кожного сервера, так і його пропускну здатність. Це допомагає покращити продуктивність та швидкодію мережі, забезпечуючи кінцевим користувачам максимально комфортне використання сервісів.

Алгоритм балансування навантаження Resource-based – це стратегія, при якій розподіл трафіку у мережі відбувається на основі ресурсів, доступних на кожному сервері. Головною метою цього алгоритму є ефективне використання ресурсів кожного сервера та уникнення перевантаження, забезпечуючи оптимальну швидкодію та якість обслуговування для кінцевих користувачів.

Особливості роботи такого алгоритму:

- кожен сервер періодично вимірює свою доступну ємність ресурсів. Це може включати кількість процесорних ядер, обсяг доступної оперативної пам'яті, швидкодію дискового простору та інші ресурси, які впливають на його здатність обслуговувати запити;

- кожен сервер надає дані про свою доступну ємність ресурсів алгоритму балансування навантаження мережі, що дозволяє алгоритму балансування приймати рішення щодо розподілу трафіку;

- алгоритм балансування навантаження мережі аналізує навантаження на кожному сервері та його доступні ресурси для визначення, який сервер може ефективно обробляти нові запити;

- при отриманні нового запиту алгоритм балансування вибирає сервер, який має найбільшу доступну кількість ресурсів для обробки цього запиту;

- після вибору сервера, алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту сервер оновлює свої дані про доступні ресурси, враховуючи використання ресурсів для обробки нових запитів;

– процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи ефективне використання ресурсів кожного сервера та максимальну швидкодію та якість обслуговування для кінцевих користувачів.

Алгоритм балансування навантаження Resource-based дозволяє забезпечити оптимальне використання ресурсів у мережі, розподіляючи трафік на основі поточного навантаження та доступних ресурсів кожного сервера. Це допомагає уникнути перевантаження серверів та забезпечити стабільну та ефективну роботу мережі.

Алгоритм балансування навантаження за допомогою методу Biased Random Sampling – це стратегія, при якій новий запит направляється на сервер за допомогою випадкового вибору з врахуванням пріоритету кожного сервера. Головна ідея цього методу полягає в тому, щоб забезпечити більш вірогідний вибір сервера з більшими ресурсами або кращою продуктивністю для обробки запитів, але при цьому залишаючи певну ймовірність вибору менш завантажених серверів. Це дозволяє підтримувати баланс навантаження у мережі, уникнути перевантаження деяких серверів та забезпечити оптимальне використання ресурсів.

Особливості роботи такого алгоритму:

– кожному серверу у мережі призначається пріоритет, який відображає його пропускну здатність, обчислювальні можливості або інші характеристики, які впливають на його здатність обслуговувати запити;

– на основі пріоритетів кожного сервера створюється пріоритетний набір серверів, де кожен сервер включений залежно від його пріоритету. Сервери з більшими пріоритетами мають більшу ймовірність бути включеними в цей набір;

– при отриманні нового запиту алгоритм балансування навантаження випадковим чином вибирає сервер із пріоритетного набору. Ймовірність вибору кожного сервера пропорційна його пріоритету;

– після вибору сервера, алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту на сервері, його стан може бути оновлений для врахування навантаження та інших факторів, які можуть впливати на подальший вибір сервера;

- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи ефективне розподілення навантаження та оптимальне використання ресурсів серверів.

Метод Biased Random Sampling дозволяє збалансувати вибір сервера, беручи до уваги їхню пропускну здатність та ресурси. Він допомагає уникнути перевантаження деяких серверів та забезпечити стабільну та ефективну роботу мережі.

Алгоритм балансування навантаження Throttled – це стратегія, яка використовується для розподілу трафіку у мережі, де запити обмежуються або затримуються на певний час для підтримки стійкої роботи серверів та запобігання перевантаженню.

Особливості роботи такого алгоритму:

- алгоритм балансування отримує новий запит від клієнта;
- перш ніж надсилати запит на обробку, алгоритм балансування перевіряє навантаження на серверах у мережі. Це може включати кількість активних з'єднань, завантаження процесора, доступну пам'ять тощо;

- якщо сервери перевантажені або не можуть оптимально обробити новий запит, алгоритм балансування використовує стратегію "Throttled" для тимчасової затримки або обмеження запиту;

- замість негайного пересилання запиту на сервер, алгоритм балансування тимчасово затримує або обмежує запит, щоб зменшити навантаження на сервери та запобігти перевантаженню;

- алгоритм балансування повідомляє клієнта про те, що їхній запит тимчасово не може бути оброблений через велике навантаження на сервери. Це може включати повідомлення про помилку або запит на повторну спробу через певний час;

- після закінчення затримки або обмеження клієнт може повторно відправити свій запит. Алгоритм балансування повторно перевіряє навантаження на серверах та надсилає запит на обробку, якщо це можливо;

- якщо навантаження на сервери зменшилося і вони готові приймати нові запити, алгоритм балансування надсилає запит на обробку на один з серверів мережі;

- процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи стійку та ефективну роботу мережі.

Алгоритм балансування навантаження Throttled дозволяє уникнути перевантаження серверів та забезпечити стабільну роботу мережі, навіть у періоди інтенсивного трафіку. Однак він може призводити до затримок у відповіді на запити клієнтів, тому важливо налаштовувати час затримки так, щоб забезпечити оптимальний баланс між навантаженням та продуктивністю.

Алгоритм балансування навантаження Least Bandwidth – це стратегія, що використовується для розподілу трафіку у мережі, де новий запит направляється на сервер з найменшою використаною пропускну здатністю. Головною метою цього алгоритму є максимізація використання пропускну здатності кожного сервера та підтримка рівномірного навантаження у мережі.

Особливості роботи такого алгоритму:

- кожен сервер у мережі періодично вимірює свою поточну пропускну здатність. Це може включати кількість переданих даних, швидкість передачі та інші параметри, які впливають на пропускну здатність;

- при отриманні нового запиту алгоритм балансування навантаження аналізує пропускну здатність кожного сервера і вибирає той, який має найменшу використану пропускну здатність;

- після вибору сервера з найменшою пропускну здатністю алгоритм балансування направляє новий запит на цей сервер для подальшої обробки;

- після обробки запиту на сервері його пропускну здатність оновлюється для врахування нових запитів, які можуть надходити в майбутньому;

– процес повторюється для кожного нового запиту, що надходить до мережі, забезпечуючи максимальне використання пропускної здатності кожного сервера та підтримуючи рівномірне навантаження у мережі.

Алгоритм балансування навантаження Least Bandwidth дозволяє оптимізувати використання ресурсів мережі, направляючи запити на сервери з найменшим навантаженням та максимальною пропускною здатністю. Це допомагає уникнути перевантаження окремих серверів та забезпечити стабільну та ефективну роботу мережі.

2.2 Аналіз проблемних питань сфери алгоритмів балансування навантаження трафіку

Ефективність алгоритмів балансування навантаження у великих мережах може бути оцінена за допомогою різних критеріїв, які враховують продуктивність, надійність, масштабованість та інші фактори.

Основні критерії ефективності алгоритмів балансування навантаження:

– ефективний алгоритм балансування навантаження повинен забезпечувати високу швидкодію та обробку запитів. Він повинен ефективно розподіляти навантаження на доступні сервери та мінімізувати час очікування для клієнтів;

– алгоритм балансування навантаження повинен бути надійним і стійким до відмов. Він повинен автоматично виявляти відмови серверів та уникати направлення запитів на несправні або перевантажені сервери;

– ефективний алгоритм балансування навантаження повинен бути масштабованим і здатним працювати великими мережами з великою кількістю серверів та клієнтів. Він повинен забезпечувати стабільну роботу навіть при збільшенні обсягів трафіку та кількості серверів;

– алгоритм балансування навантаження повинен бути гнучким і можливим до налаштування з урахуванням поточних потреб мережі. Він повинен

підтримувати різні стратегії балансування та дозволяти змінювати їх залежно від умов мережі;

- ефективний алгоритм балансування навантаження повинен бути здатний адаптуватися до змін у мережі, таких як збільшення обсягу трафіку, зміна кількості серверів або з'явлення нових типів запитів. Він повинен швидко реагувати на зміни умов і підтримувати оптимальний розподіл навантаження;

- алгоритм балансування навантаження повинен мати вбудовану систему моніторингу та керування, яка дозволяє відстежувати стан серверів та навантаження у реальному часі і при необхідності реагувати на виниклі проблеми;

- алгоритм балансування навантаження повинен бути безпечним і здатним запобігати атакам, таким як DDoS (англ. DDoS attack, distributed denial-of-service attack) або перехоплення даних. Він повинен забезпечувати захист в умовах збереження конфіденційності та цілісності даних, а також забезпечувати безпеку транзиту даних між клієнтами і серверами;

- алгоритм балансування навантаження повинен бути легко інтегрованим з існуючими системами моніторингу, реєстрації і керування серверами. Це дозволяє спрощувати процес впровадження та підтримки алгоритму в мережному середовищі;

- важливим критерієм є вартість впровадження та обслуговування алгоритму. Він повинен бути ефективним з точки зору витрат і не потребувати значних витрат на реалізацію, налаштування і підтримку;

- алгоритм балансування навантаження повинен мати мінімальний вплив на продуктивність системи. Він не повинен створювати додаткове навантаження на сервери чи мережу, щоб уникнути зниження продуктивності системи.

Враховуючи ці критерії, можна ефективно оцінити та вибрати алгоритм балансування навантаження, який найкращим чином відповідає потребам конкретної мережі та її умовам використання.

Географічне розташування обчислювальних пристроїв має значний вплив на ефективність роботи алгоритмів балансування навантаження у мережі.

Ключові аспекти, які можуть впливати на ефективність роботи алгоритмів балансування навантаження мережі:

- відстань між клієнтами і серверами може призводити до затримок у передачі даних через мережу. Чим ближче сервер до клієнта, тим менше затримок. Однак для ефективного використання ресурсів мережі, можуть бути встановлені сервери в різних географічних областях. Алгоритми балансування навантаження повинні враховувати ці затримки та вибирати сервер, що знаходиться якомога ближче до клієнта або має менші затримки;

- географічне розташування може впливати на доступність даних. Ідея реплікації даних полягає у тому, щоб мати копії даних на різних серверах у різних місцях. Алгоритми балансування навантаження можуть враховувати цей аспект, спрямовуючи запити на сервери з копіями даних, які знаходяться якомога ближче до клієнтів;

- у випадку виникнення аварій або проблем у одній географічній області, сервери в інших областях можуть надавати резервний сервіс. Алгоритми балансування навантаження можуть враховувати цю можливість та автоматично перенаправляти трафік на інші сервери у випадку відмови;

- в деяких випадках передача даних між географічно віддаленими серверами може бути дорогою. Алгоритми балансування навантаження можуть оптимізувати маршрутизацію трафіку так, щоб уникнути зайвих витрат на передачу даних;

- у деяких країнах можуть існувати законодавчі обмеження щодо зберігання, передачі та обробки даних. Географічне розташування серверів може впливати на відповідність цим обмеженням. Алгоритми балансування навантаження повинні бути налаштовані з урахуванням таких вимог.

Ефективність роботи алгоритмів балансування навантаження у великих мережах значно залежить від географічного розташування обчислювальних пристроїв, і вона повинна бути узгоджена з місцезнаходженням клієнтів, доступністю даних та вимогами до продуктивності і безпеки.

Стратегії масштабування використовуються для адаптації алгоритмів балансування навантаження на фоні зростаючого обсягу трафіку у мережах.

Найпоширеніші стратегії масштабування:

– *вертикальне масштабування*. При вертикальному масштабуванні ресурси серверів збільшуються за рахунок покращення апаратної частини (наприклад, збільшення об'єму оперативної пам'яті, потужності процесора тощо) або оптимізації програмного забезпечення. При зростанні обсягу трафіку алгоритми балансування можуть автоматично розподіляти навантаження на підвищені ресурси серверів для забезпечення продуктивності та швидкодії;

– *горизонтальне масштабування*. При горизонтальному масштабуванні ресурси розподіляються на більшу кількість серверів. Це може включати додавання нових серверів у мережу або розподіл функцій між декількома серверами. Алгоритми балансування можуть динамічно розподіляти трафік між різними серверами залежно від їхньої завантаженості, щоб оптимізувати використання ресурсів;

– *автоматичне масштабування*. Автоматичне масштабування використовує розумні алгоритми та метрики для виявлення змін у навантаженні та автоматичного масштабування ресурсів. Алгоритми балансування можуть використовувати дані моніторингу, такі як завантаження серверів, обсяг трафіку тощо, для прийняття рішень щодо автоматичного масштабування;

– *гібридне масштабування*. Гібридне масштабування поєднує в собі як вертикальне, так і горизонтальне масштабування. Воно використовує переваги обох підходів для оптимізації продуктивності та ефективності. Алгоритми балансування можуть розподіляти трафік між різними типами серверів (наприклад, серверами з великою кількістю ресурсів і серверами з більшою кількістю екземплярів) в залежності від поточних умов мережі;

– *еластичне масштабування*. Еластичне масштабування включає автоматичне масштабування в залежності від обсягу трафіку та ресурсів, що доступні для мережі, з метою забезпечення оптимальної продуктивності. Алгоритми балансування можуть динамічно адаптуватися до змін у навантаженні та ресурсах, автоматично перерозподіляючи трафік для максимізації ефективності мережі.

Ці стратегії масштабування дозволяють адаптувати алгоритми балансування навантаження до зростаючого обсягу трафіку у мережах, забезпечуючи оптимальне використання ресурсів та продуктивність системи.

При розробці алгоритмів балансування навантаження для великих мереж важливо враховувати вимоги до швидкодії та надійності, оскільки вони безпосередньо впливають на продуктивність та доступність системи.

Основні вимоги до швидкодії та надійності алгоритмів балансування навантаження для розподілу трафіку у великих мережах:

- алгоритми балансування навантаження повинні працювати швидко та ефективно, мінімізуючи затримки у передачі даних та обробці запитів. Це особливо важливо для великих мереж, де кожна мілісекунда може мати значення для продуктивності;
- алгоритми повинні ефективно розподіляти навантаження між серверами, уникати перевантаження окремих вузлів та забезпечувати рівномірне використання ресурсів мережі;
- для забезпечення швидкої доставки даних алгоритми повинні враховувати оптимальні маршрути та шляхи передачі, мінімізуючи кількість проміжних вузлів і затримок;
- алгоритми мають бути здатні автоматично виявляти відмови серверів та перенаправляти трафік на інші доступні вузли. Це дозволяє забезпечити безперервну роботу системи при виникненні проблем;
- для забезпечення надійності системи алгоритми можуть враховувати наявність резервних копій даних на різних серверах або в різних дата-центрах;
- алгоритми повинні мати вбудовані заходи безпеки, щоб запобігати атакам, таким як DDoS, та забезпечувати конфіденційність та цілісність даних;
- алгоритми повинні бути розроблені таким чином, щоб система продовжувала працювати навіть у випадку відмови окремих компонентів або дата-центрів.

Забезпечення високої швидкодії та надійності алгоритмів балансування навантаження є критичним для ефективного функціонування великих мереж. Ці

вимоги впливають на розробку та вибір алгоритмів, які повинні бути готові працювати в умовах високого навантаження та надійно забезпечувати доступ до ресурсів.

Використання алгоритмів балансування навантаження в хмарних середовищах може стикатися з різноманітними викликами, які варто врахувати для забезпечення ефективної роботи системи.

Найпоширеніші виклики та шляхи їх вирішення:

1. *динамічність та масштабованість*. Хмарні середовища часто характеризуються змінним обсягом трафіку та зміною в навантаженні на сервери. Це створює виклик для алгоритмів балансування навантаження, оскільки вони повинні бути здатними адаптуватися до змін у ресурсах та потребах користувачів. Шляхи вирішення цього виклику включають автоматичне масштабування ресурсів, використання алгоритмів, які враховують стан мережі та серверів у реальному часі, а також використання гнучких стратегій балансування навантаження, що можуть динамічно реагувати на зміни;

2. *локалізованість даних*. У хмарних середовищах дані можуть бути розподілені по різних регіонах або дата-центрах. Це може вплинути на ефективність алгоритмів балансування, оскільки вони повинні враховувати місцезнаходження даних та мінімізувати затримки у доступі до них. Для вирішення цього виклику можна використовувати алгоритми, які базуються на маршрутизації на основі місцезнаходження або використовують методи кешування даних для покращення доступу до них;

3. *безпека та конфіденційність*. Забезпечення безпеки та конфіденційності даних є важливим аспектом в хмарних середовищах. Алгоритми балансування навантаження повинні враховувати ці аспекти, уникати концентрації чутливих даних на певних серверах та забезпечувати захист від атак. Це можна досягти за допомогою шифрування даних, використання захищених каналів зв'язку та впровадження механізмів автентифікації та авторизації;

4. *управління ресурсами*. В хмарних середовищах ресурси можуть бути обмеженими. Алгоритми балансування навантаження повинні ефективно

управляти цими ресурсами, враховуючи їх доступність та використання. Це може включати автоматичне виявлення та управління перевантаженими серверами, розподіл ресурсів з урахуванням потреб додатків та користувачів, а також оптимізацію використання ресурсів через динамічне налаштування параметрів системи відповідно до поточного навантаження;

5. витрати на мережу. При розподілі трафіку між серверами в хмарних середовищах важливо враховувати витрати на мережу. Це означає мінімізацію зайвих переадресацій та зменшення обсягу даних, які потрібно передавати між вузлами. Шляхи вирішення цього включають використання методів кешування, компресії даних та локального зберігання даних для зменшення обсягу трафіку між серверами;

6. сумісність та інтеграція. При використанні алгоритмів балансування навантаження в хмарних середовищах важливо враховувати сумісність з іншими компонентами системи та можливість їх інтеграції. Це означає використання стандартних протоколів та інтерфейсів для взаємодії з іншими системами, а також підтримку різних форматів даних та архітектур;

7. моніторинг та аналіз результатів. Важливо мати ефективні засоби моніторингу та аналізу роботи алгоритмів балансування навантаження в реальному часі. Це дозволяє вчасно виявляти проблеми та впроваджувати відповідні корективи для покращення продуктивності та надійності системи.

Розв'язання цих викликів допомагає створити ефективні та надійні системи балансування навантаження в хмарних середовищах, які забезпечують високу доступність, продуктивність та безпеку обробки даних та ресурсів.

2.3 Проблемне питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах

Проблема підбору алгоритмів балансування навантаження для розподілу трафіку у великих мережах постає на моменті створення умов моніторингу стану мережі та забезпечення обміну службовою інформацією серверів у мережі за допомогою алгоритмів балансування навантаження. У великих мережах зазвичай наявна значна кількість серверів що надають послуги та розподіляють навантаження в мережі, через це, при використанні динамічних алгоритмів балансування навантаження, під час виконання певних правил алгоритмів балансування в мережі може поширюватися значна кількість службового трафіку, що може викликати збільшення навантаження на обчислювальні пристрої спричинивши перевантаження або зменшення продуктивності мережі. Використання статичних алгоритмів балансування навантаження зазвичай не є виходом із ситуації, в силу того, що у великих мережах часто здійснюється зміна обчислювальних можливостей серверів або вони можуть приймати допоміжний трафік на певний момент часу, тому такі статичні алгоритми балансування навантаження не будуть продуктивними.

Найкращими рішеннями для вирішення проблемного питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах, буде використання набору сумісних динамічних алгоритмів балансування навантаження, таким чином можна забезпечити запобігання недоліків тих чи інших алгоритмів, та сильно покращити продуктивність мережі в цілому.

РОЗДІЛ 3. ВИЗНАЧЕННЯ ТА АНАЛІЗ ПІДХОДЯЩИХ РІШЕНЬ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ

3.1 Методи комбінування алгоритмів балансування навантаження мережі

Існує кілька методів комбінування алгоритмів балансування навантаження мережі, які дозволяють поєднувати різні підходи для отримання оптимального результату.

Найбільш популярні методи комбінування:

- послідовне застосування;
- вагове комбінування;
- адаптивне комбінування;
- гібридні рішення;
- прийняття рішень на основі правил.

Послідовне застосування в контексті комбінування алгоритмів балансування навантаження мережі означає застосування кількох алгоритмів один за одним у послідовному порядку. Кожен наступний алгоритм отримує результати попереднього і може додатково розподіляти навантаження між серверами чи вузлами мережі.

Ключові аспекти використання методу "послідовного застосування":

- важливо визначити послідовність застосування алгоритмів. Наприклад, спочатку може бути застосований алгоритм Round Robin, а потім – Least Connection. Це означає, що кожен наступний алгоритм отримує результати попереднього і може враховувати їх при прийнятті власного рішення щодо розподілу навантаження;
- кожен наступний алгоритм може додатково обробляти результати попереднього. Наприклад, якщо попередній алгоритм відправив запит на сервер з

найменшим навантаженням, наступний алгоритм може враховувати час відповіді цього сервера для подальшого розподілу трафіку;

- кожен алгоритм може приймати рішення на основі поточних умов мережі та стану серверів. Наприклад, якщо певний алгоритм виявляє перевантаження на певному сервері, наступний алгоритм може врахувати цю інформацію при прийнятті рішення щодо розподілу трафіку;

- послідовне застосування може бути досить простим у реалізації, оскільки кожен алгоритм може бути реалізований окремо. Це дозволяє використовувати готові алгоритми балансування навантаження і комбінувати їх у послідовному порядку з мінімальними зусиллями.

Послідовне застосування дозволяє досягти більшої гнучкості в управлінні навантаженням мережі та ресурсами, оскільки кожен алгоритм може додавати свою унікальну логіку та логіку прийняття рішень до загального процесу балансування навантаження.

Вагове комбінування є методом комбінування алгоритмів балансування навантаження, при якому кожен алгоритм має свою вагу, яка визначає його вплив на кінцевий результат балансування. Цей метод дозволяє точніше керувати тим, які алгоритми використовуються у конкретних умовах та встановлювати їхню роль у розподілі навантаження.

Ключові аспекти використання методу "вагового комбінування":

- кожен алгоритм балансування навантаження отримує вагу, яка вказує, наскільки важливий він для загального процесу балансування. Ця вага може бути числовим значенням, яке відображає важливість алгоритму в контексті поточних потреб мережі;

- вага кожного алгоритму може бути визначена на основі різних факторів, таких як пропускна здатність серверів, їхні можливості обробки запитів, швидкість відповіді і т. д. Вага також може бути встановлена адміністратором мережі вручну в залежності від конкретних вимог;

- вага кожного алгоритму може використовуватися як коефіцієнт впливу на кінцевий результат балансування навантаження. Наприклад, якщо алгоритм має

вагу 2, то його рішення з питань розподілу навантаження буде враховуватися вдвічі більше, ніж у алгоритму з вагою 1;

- ваги кожного алгоритму можуть бути динамічно змінюваними в залежності від умов мережі та поточного навантаження. Наприклад, якщо певний алгоритм починає виявляти перевантаження, його вага може бути зменшена для зменшення його впливу на загальний процес балансування;

- вагове комбінування може бути комбіновано з іншими методами комбінування, такими як послідовне застосування чи адаптивне комбінування. Це дозволяє створювати складні алгоритми балансування навантаження, які ефективно використовують різні підходи управління трафіком.

Вагове комбінування дозволяє адміністраторам мережі точніше контролювати процес балансування навантаження та адаптувати його до поточних умов мережі. Цей метод забезпечує більшу гнучкість управління трафіком і може допомогти оптимізувати продуктивність мережі.

Адаптивне комбінування алгоритмів балансування навантаження - це метод, який використовується для автоматичного змішування або вибору алгоритмів в залежності від змінних умов мережі. Основна ідея полягає в тому, щоб система могла адаптуватися до змінних умов, таких як обсяг трафіку, навантаження на сервери, доступність ресурсів та інші фактори, і змінювати комбінацію алгоритмів відповідно до цих умов для забезпечення оптимального розподілу трафіку.

Ключові аспекти використання методу "адаптивного комбінування":

- для успішної реалізації адаптивного комбінування необхідно мати системи моніторингу, які визначають поточний стан мережі, серверів та навантаження. Ці системи можуть включати моніторинг пропускної здатності, навантаження на сервери, час відповіді та інші метрики продуктивності;

- на основі даних, зібраних з систем моніторингу, система повинна аналізувати та виявляти зміни у стані мережі та серверів. Це може включати виявлення зростання трафіку, перевантаження серверів або відмов у праці серверів;

- на основі аналізу даних система повинна вибирати оптимальний алгоритм або комбінацію алгоритмів для розподілу трафіку. Це може включати вибір

алгоритму з найменшим навантаженням, алгоритму з найкращою доступністю або алгоритму, який найбільш ефективно працює в поточних умовах мережі;

- після вибору оптимального алгоритму система може автоматично налаштовувати його параметри відповідно до поточних умов мережі. Це може включати зміну ваги алгоритму, зміну часу життя сесій або інші параметри, що впливають на розподіл трафіку;

- адаптивне комбінування може постійно вдосконалюватися через навчання з досвіду. Система може використовувати історичні дані та результати використання алгоритмів для оптимізації свого вибору та налаштування в майбутньому.

Адаптивне комбінування дозволяє мережам автоматично адаптуватися до змінних умов та оптимізувати розподіл трафіку для забезпечення максимальної продуктивності та надійності системи.

Гібридні рішення в контексті комбінування алгоритмів балансування навантаження є методом, який поєднує різні алгоритми в одному складному механізмі для досягнення оптимального розподілу трафіку в мережі. Цей підхід дозволяє використовувати переваги різних алгоритмів і адаптуватися до різноманітних умов мережі.

Ключові аспекти використання методу "гібридного комбінування":

- гібридні рішення об'єднують у собі різні алгоритми балансування навантаження, які можуть мати різні принципи роботи та логіку прийняття рішень. Наприклад, можна поєднати алгоритм Round Robin із алгоритмом Least Connection, щоб забезпечити рівномірний розподіл трафіку серед серверів з найменшою кількістю активних з'єднань;

- у гібридних рішеннях можна встановлювати пріоритети для кожного алгоритму в залежності від поточних умов мережі. Наприклад, можна надати більший пріоритет алгоритму, який забезпечує найменшу відмовостійкість, або алгоритму, який краще пристосований до змінного обсягу трафіку;

- гібридні рішення можуть використовувати адаптивне керування для автоматичного змінювання комбінації алгоритмів в залежності від змінних умов

мережі. Наприклад, система може автоматично визначати, які алгоритми працюють краще в певних умовах та адаптивно змінювати їх комбінацію;

- гібридні рішення можуть постійно вдосконалюватися через навчання з досвіду. Система може використовувати історичні дані та результати використання алгоритмів для оптимізації свого вибору та налаштування в майбутньому.

Гібридні рішення дозволяють мережам максимально ефективно використовувати ресурси та оптимізувати розподіл трафіку. Вони забезпечують більшу гнучкість та точність управління навантаженням мережі, оскільки можуть поєднувати в собі переваги різних алгоритмів та адаптуватися до змінних умов.

Прийняття рішень на основі правил – це метод комбінування алгоритмів балансування навантаження, при якому рішення про те, який алгоритм використовувати в конкретному випадку, приймається на основі заданих правил або умов. Цей метод може бути ефективним, оскільки дозволяє гнучко налаштовувати роботу мережі в залежності від різних ситуацій і умов.

Ключові аспекти використання методу "прийняття рішень на основі правил":

- перший крок у використанні цього методу – це визначення правил або критеріїв, за якими будуть прийматися рішення щодо вибору алгоритму балансування навантаження. Ці правила можуть включати такі аспекти, як обсяг трафіку, доступність ресурсів, навантаження на сервери, час відповіді та інші метрики продуктивності;

- на основі визначених правил розробляється логіка або програмне забезпечення, яке приймає рішення щодо вибору алгоритму балансування навантаження. Це може бути реалізовано за допомогою скриптів, програмних модулів або спеціалізованих систем управління;

- прийняття рішень на основі правил зазвичай вимагає виконання аналізу у реальному часі для оцінки поточних умов мережі. Це може включати моніторинг трафіку, навантаження на сервери, пропускну здатність мережі та інші параметри для визначення, яке правило має бути застосоване в конкретний момент часу;

- після того як прийнято рішення на основі правил, система може автоматично виконати відповідні дії щодо вибору та налаштування алгоритмів

балансування навантаження. Це може включати переключення між алгоритмами, зміну параметрів алгоритмів або інші дії, спрямовані на оптимізацію роботи мережі;

– щоб покращити ефективність системи прийняття рішень на основі правил, важливо постійно збирати дані та навчатися з досвіду. Історичні дані про роботу мережі можуть бути використані для оптимізації правил і вдосконалення роботи системи в цілому.

Прийняття рішень на основі правил дозволяє мережам автоматично адаптуватися до змінних умов та оптимізувати розподіл трафіку. Цей метод забезпечує більшу гнучкість та точність управління навантаженням мережі, оскільки може враховувати різні умови та критерії для прийняття рішень.

Методи комбінування алгоритмів навантаження є потужним інструментом для оптимізації роботи мережі, який дозволяє поєднувати переваги різних підходів для досягнення більш ефективного розподілу трафіку. Загалом, комбінація алгоритмів навантаження дозволяє мережі бути більш гнучкою, надійною та продуктивною, адаптуючись до різноманітних умов та вимог користувачів. Найбільш ефективні методи комбінування алгоритмів використовують різні стратегії, такі як вагове комбінування, адаптивне комбінування, гібридні рішення та прийняття рішень на основі правил. Вони дозволяють мережі адаптуватися до змінних умов та вимог, забезпечуючи оптимальний розподіл трафіку та максимальну продуктивність.

3.2 Рішення для балансування навантаження для розподілу трафіку у великих мережах

Найкращими рішеннями балансування навантаження для розподілу трафіку у великих мережах будуть рішення розроблені на базі певного типу комбінування

алгоритмів балансування та підходящих під умови мережі алгоритмів балансування.

Можливе рішення №1. Створення методу балансування навантаження великої мережі, на базі такого типу комбінування алгоритмів, як «послідовне комбінування» із набором алгоритмів балансування як: Resource-based, Opportunistic, IP/URL-Hash.

В цьому рішенні метод балансування буде будуватися на взаємодії трьох алгоритмів балансування, які будуть працювати за послідовною схемою комбінування.

План роботи рішення №1:

1. динамічний алгоритм балансування навантаження «Resource-based», стоїть на першому етапі роботи, як зображено на рис. 3.1, даний алгоритм ініціює та виконує пошук готових до прийняття трафіку серверів, збирає дані про їх стан, кількість доступних до залучення в обчислення ресурсів, поточну завантаженість, час відповіді та інші метрики, після чого передає ці дані на наступний етап;

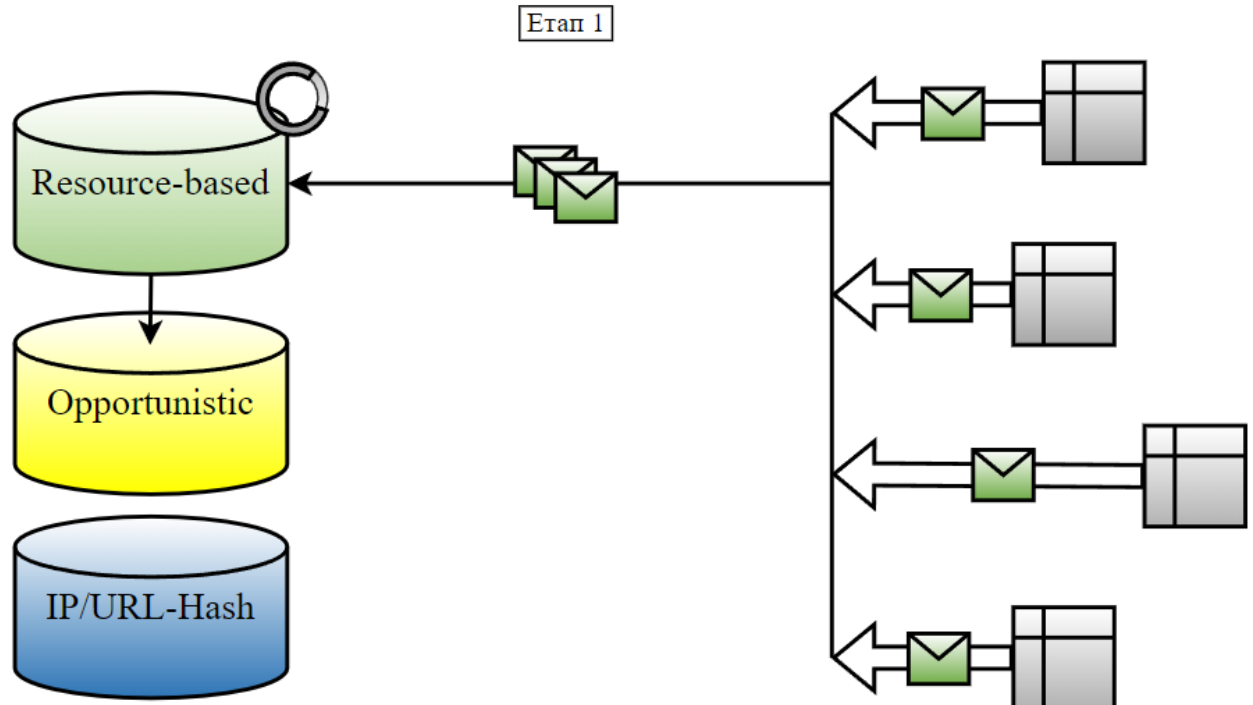
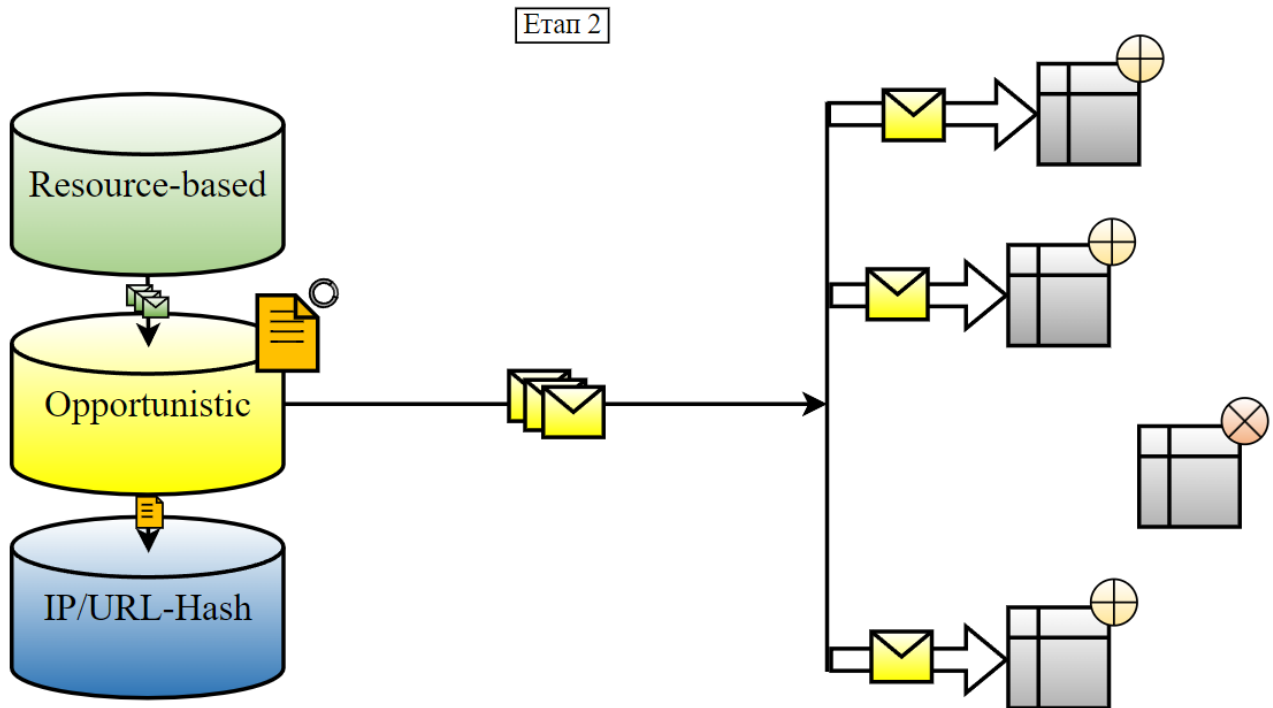


Рисунок 3.1 – Схема роботи першого етапу рішення №1

2. статичний алгоритм балансування навантаження «Opportunistic», стоїть на другому етапі роботи, як зображено на рис. 3.2, даний алгоритм, на основі зібраних алгоритмом балансування «Resource-based» даних, особливо акцентуючи увагу на часі відповіді та поточній завантаженості, робить список для залучення підходящих на даний момент серверів для розподілення трафіку та обирає ці сервери;



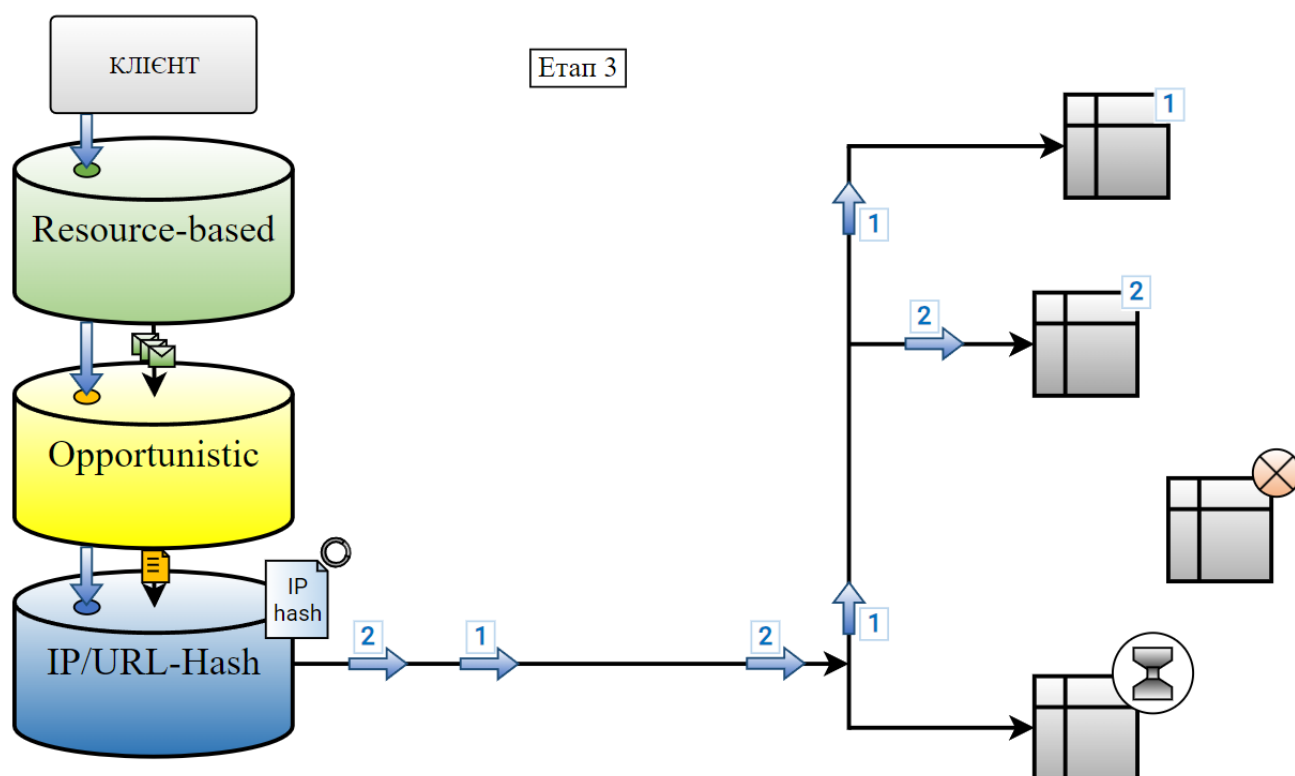


Рисунок 3.3 – Схема роботи третього етапу рішення №1

Переваги такого рішення балансування:

- постійний збір інформації про стан метрик доступних серверів;
- не має великої необхідності у залученні серверів тільки з однаковими показниками обчислювальних можливостей;
- визначення серверів з найкращими показниками метрик, з подальшим їх залученням до списку;
- можливість налаштування нижнього та верхнього рівня метрик серверів, що залучаються до списку;
- вибір серверів на основі їх швидкодії та часу відповіді;
- при відсутності потреби у прив'язці клієнта до сесії, вибір сервера для розподілення навантаження можна виконувати на другому етапі, за рахунок статичного алгоритму балансування «Opportunistic»;
- при наявній потребі у прив'язці клієнта до сесії, наприклад, якщо це мережа онлайн магазину, вибір сервера для розподілення навантаження виконується на третьому етапі алгоритмом «IP/URL-Hash», який обчислює хеш-

значення IP/URL клієнта та прив'язує його до певного, «найближчого» серверу, тобто до серверу з найменшим часом відповіді;

- інтегрування та налаштування такого рішення не буде складним, за рахунок простої схеми послідовного комбінування алгоритмів.

Недоліки такого рішення балансування:

- при моніторингу стану та метрик серверів, може створюватися додатковий службовий трафік у мережі, що може в деякому степені зменшити обчислювальні ресурси мережі;

- на другому етапі потрібно правильно налаштувати рівні за якими сервери призначаються до списку обчислювальних ресурсів, щоб мережа використовувала ресурси правильно та повноцінно.

Можливе рішення №2. Створення методу балансування навантаження великої мережі, на базі такого типу комбінування алгоритмів, як «вагове комбінування» із набором алгоритмів балансування як: Resource-based, Weighted Response Time, Round Robin.

В цьому рішенні метод балансування буде будуватися на взаємодії трьох алгоритмів балансування, які будуть працювати за визначеними коефіцієнтами пріоритету. Прикладом можуть бути наступні коефіцієнти: «Resource-based – вага 3, Weighted Response Time – вага 2, Round Robin – вага 1». Чим більша вага коефіцієнту, тим більш високий пріоритет у алгоритму.

План роботи рішення №2:

1. як зображено на рис. 3.4, системний адміністратор з урахуванням поточних особливостей мережі, задає коефіцієнти пріоритетів для алгоритмів балансування;

2. як зображено на рис. 3.5, в роботу вступає алгоритм балансування з найбільшим коефіцієнтом пріоритетності, цей алгоритм визначає список доступних до залучення в обчислюванні трафіку серверів. Наприклад, динамічний алгоритм балансування «Resource-based», має коефіцієнт пріоритету – 3, в такому випадку, цей алгоритм на основі своїх правил виконує моніторинг стану та метрик серверів, та передає цю інформацію на наступний етап роботи;

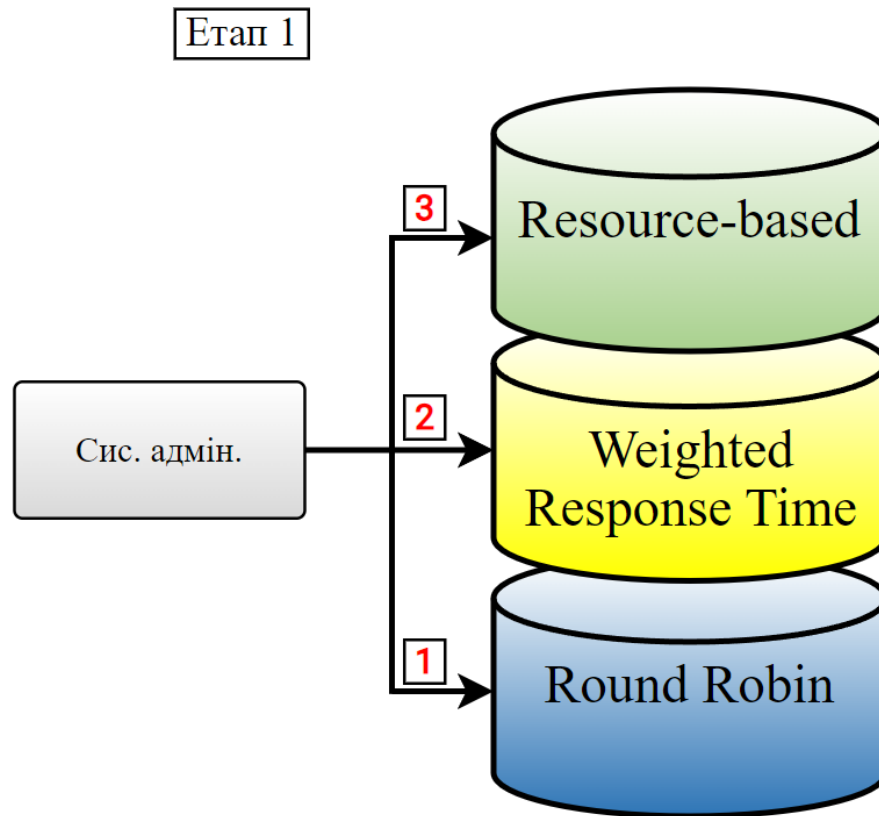


Рисунок 3.4 – Схема роботи першого етапу рішення №2

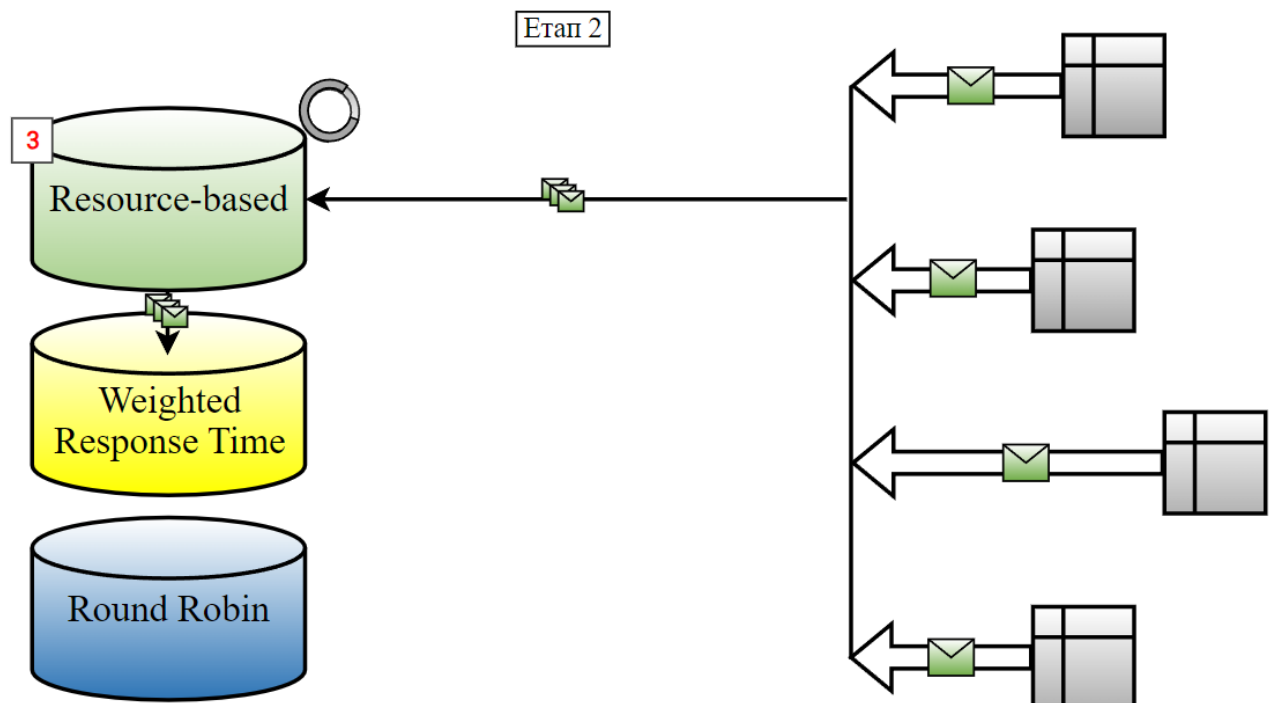


Рисунок 3.5 – Схема роботи другого етапу рішення №2

3. на наступному етапі роботи, як зображено на рис. 3.6, вирішується на які з доступних серверів у списку буде посилатися трафік від клієнта для розподілення. Наприклад, алгоритм балансування «Weighted Response Time», має коефіцієнт пріоритету – 2. Таким чином, саме цей алгоритм буде обробляти інформацію про стан та метрики, що була зібрана та передана минулим алгоритмом балансування. На основі своїх правил визначає які сервери зі списку будуть задіяні у балансуванні навантаження, в цьому прикладі, це будуть сервери з найменшим часом відповіді, але також при цьому, кожному серверу на основі даних про обчислювальні можливості буде присвоєно свій коефіцієнт пріоритетності, що дасть змогу правильно та рівномірно розподілити навантаження на обраних серверах;

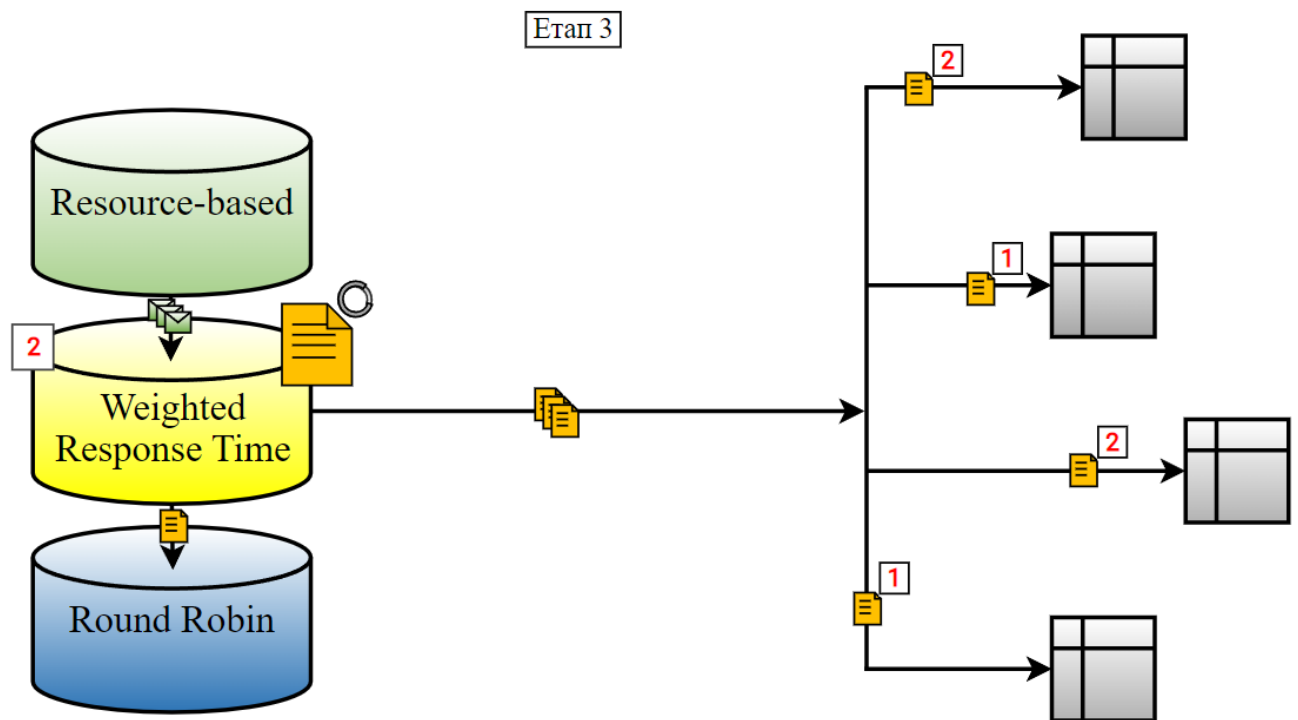


Рисунок 3.6 – Схема роботи третього етапу рішення №2

4. на основі отриманих від минулого алгоритму даних, алгоритм балансування навантаження «Round Robin», маючи найменшу вагу буде розподіляти трафік між серверами, що призначив та визначив для них коефіцієнти минулий алгоритм. Як показано на рис. 3.7, маємо два сервери з коефіцієнтом 1, та два сервери з коефіцієнтом 2, алгоритм балансування «Round Robin», за своїми правилами розподілу буде направляти навантаження між двома серверами з

коефіцієнтом "1". Перевагою цього алгоритму є те, що можна вручну налаштувати в якому порядку будуть рахуватися сервери, наприклад, чим вище коефіцієнт у сервера, тим менший цей сервер має пріоритет для алгоритму, або навпаки.

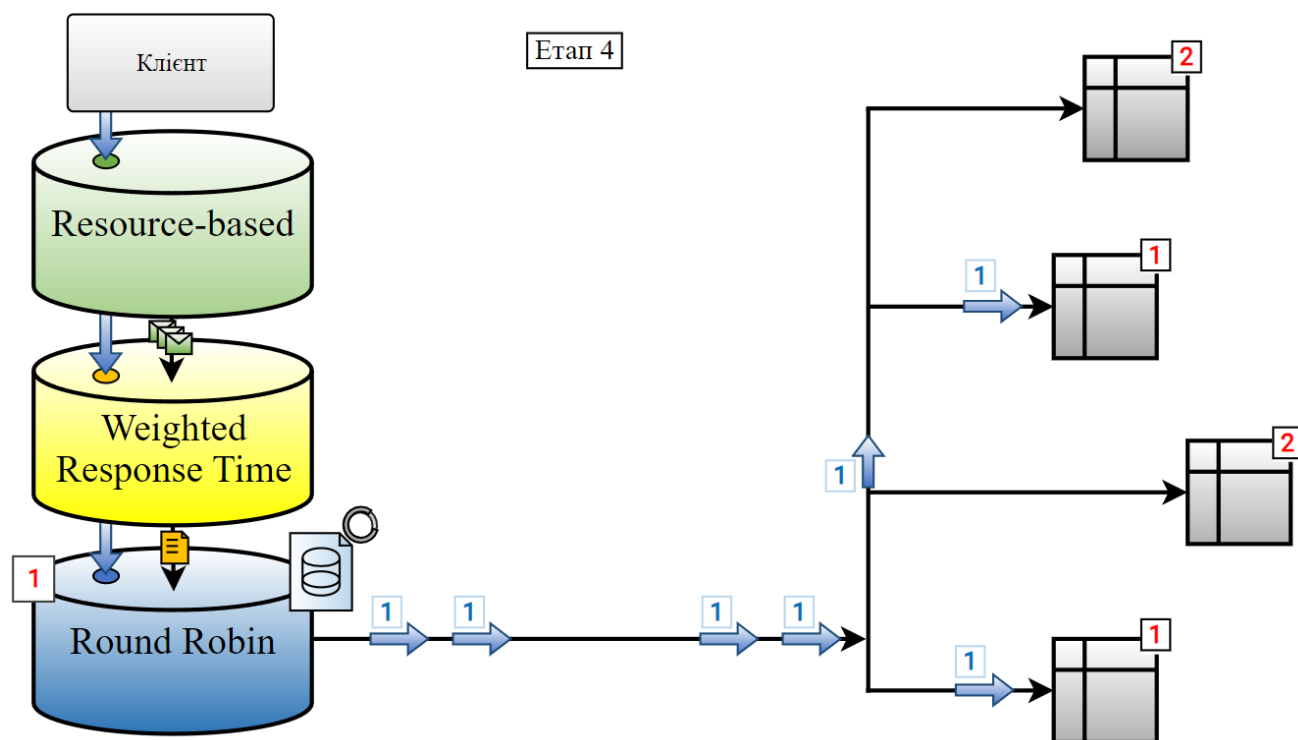


Рисунок 3.7 – Схема роботи четвертого етапу рішення №2

Після того як розпочнеться розподіл навантаження, алгоритм балансування «Resource-based», почне замір метрик серверів з початку, тим самим запуститься круговий процес присвоєння коефіцієнтів, тому при розподілі навантаження на кінцевому етапі за допомогою алгоритму балансування «Round Robin», не буде перенавантажувати одні й ті ж самі сервери, коефіцієнти пріоритету серверів будуть змінюватися динамічно, що дозволить розподілити навантаження на всі сервери рівномірно та ефективно.

Переваги такого рішення балансування:

– за рахунок використання схеми «вагове комбінування», можна в ручну виставити коефіцієнти пріоритетів як алгоритмам балансування, так і серверам;

- використання трьох алгоритмів балансування навантаження, дозволяє точно та рівномірно розподілити навантаження на потрібних серверах в потрібній кількості;

- такий метод балансування доволі зручний в силу використання не дуже складних алгоритмів балансування навантаження. Ці алгоритми дозволяють нівелювати недоліки одне-одного та збільшити кількість переваг, що дуже позитивно впливає на стан мережі;

- за рахунок алгоритму «Weighted Response Time» досягається найбільш ефективний розподіл навантаження за часом відповіді серверів в мережі, що зменшує час очікування відповіді для клієнта та покращує QoS (Quality of Service).

Недоліки такого рішення балансування:

- необхідне базове налаштування коефіцієнтів на початковому етапі рішення;

- необхідне базове знання структури та потреб великої мережі;

- для підтримання сесії, необхідно додавати інше підходяще для цього рішення.

ВИСНОВКИ

В даній роботі були опрацьовані питання дослідження та аналізу існуючих рішень балансування навантаження трафіку, були проаналізовані різні теорії та дослідження на тему алгоритмів балансування навантаження в мережі, на основі цих досліджень були приведені висновки їх ефективності та доцільності використання. Були розроблені та опрацьовані проблемні питання у сфері балансування навантаження трафіку в мережах, також було опрацьовано методичні підходи для дослідження та аналізу підходящих алгоритмів балансування навантаження для розподілу трафіку у великих мережах, що дало змогу розкрити питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах, проаналізувати та структурувати результати цього дослідження. Було проведено визначення методів комбінації алгоритмів балансування навантаження, з метою розширення можливостей мережі.

Основною метою дослідження цієї теми, було вирішення проблемного питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах. Після досліджень великої кількості публікацій, статей та досліджень інших наукових авторів було досягнуто кінцевого висновку по цьому питанні, що дало змогу вирішити найбільш релевантні рішення балансування навантаження у великих мережах, ними виявилися рішення з комбінуваннями певних алгоритмів за певними схемами комбінування. Було наведено два можливих рішення балансування навантаження трафіку, що найбільше підходять до деяких умов великих мереж та ефективно виконують свої поставлені задачі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Load Balancing Algorithms – FasterCapital. FasterCapital. URL: <https://fastercapital.com/keyword/load-balancing-algorithms.html> (дата звернення: 02.04.2024).
2. Karakurt M. Load Balancers and Their Algorithms: Key Players in Internet Traffic Management. Medium. URL: <https://medium.com/@mrtrkrtr/load-balancers-and-their-algorithms-key-players-in-internet-traffic-management-aeaa544e3bce> (дата звернення: 02.04.2024).
3. What is Load Balancing? – Load Balancing Algorithm Explained – AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/load-balancing/?nc1=h_ls (дата звернення: 02.04.2024).
4. Load Balancing Algorithms, Types and Techniques – Kemp. Load Balancer For Always-On Application Experience – Kemp. URL: <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques> (дата звернення: 02.04.2024).
5. Load Balancing Algorithms: A Deep Dive into Everything You Need to Know. Cloudzy. URL: <https://cloudzy.com/blog/load-balancing-algorithms/> (дата звернення: 02.04.2024).
6. Static vs Dynamic Load Balancing - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/static-vs-dynamic-load-balancing/> (дата звернення: 02.04.2024).
7. What is Static Load Balancing? | Avi Networks. Avi Networks. URL: <https://avinetworks.com/glossary/static-load-balancing/> (дата звернення: 02.04.2024).
8. Abani K., Akingbehin K., Shaout A. Fuzzy decision making for load balancing in a distributed system. Detroit, MI, USA : 36th Midwest Symposium on Circuits and Systems, 1993. 502 с.
9. Калюжний О. О. АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ВИКОРИСТАННЯ З ЧАСТКОВО-ПЕРЕСІЧНИМИ ШЛЯХАМИ ПЕРЕДАЧІ ДАНИХ. ЖУРНАЛ НАУКОВИЙ ОГЛЯД. 2019. № 6. С. 11.

10. Alonso R., Cova L. L. Sharing Jobs Among Independently Owned Processors : навч. посіб. 8-ме вид. New York : IEEE, 1988. 288 с.

11. Шаповал І. С. АДАПТИВНИЙ АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В КОМП'ЮТЕРНИХ МЕРЕЖАХ. ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ № 3 : УДК 621.797:621.664, м. Кременчук, 2010 р. Кременчук, 2010. С. 10.

12. Load Balancing Algorithms – Elastic Load Balancing - User Guide. Open Telekom Cloud Service Documentation – docsportal documentation. URL: https://docs.sc.otc.t-systems.com/elastic-load-balancing/umn/listener/load_balancing_algorithms.html (дата звернення: 02.04.2024).

13. Sharma A. A High-Level Overview of Load Balancing Algorithms. Medium. URL: <https://betterprogramming.pub/a-high-level-overview-of-load-balancing-algorithms-8c7d3368276> (дата звернення: 15.04.2024).

14. Sharma A. High Level Overview of Load Balancing Algorithms. DEV Community. URL: <https://dev.to/aastikta/high-level-overview-of-load-balancing-algorithms-3mfh> (дата звернення: 15.04.2024).

15. Types of load balancing algorithms. Connect, Protect and Build Everywhere | Cloudflare. URL: <https://www.cloudflare.com/learning/performance/types-of-load-balancing-algorithms/> (дата звернення: 15.04.2024).

16. Contributors to Wikimedia projects. Load balancing (computing) - Wikipedia. Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Load_balancing_\(computing\)](https://en.wikipedia.org/wiki/Load_balancing_(computing)) (дата звернення: 15.04.2024).

17. Prashant J., Shenoy, Harrick M. V. Static Load Balancing. ScienceDirect. URL: <https://www.sciencedirect.com/topics/computer-science/static-load-balancing> (дата звернення: 15.04.2024).

18. Designing Low-Complexity Heavy-Traffic Delay-Optimal Load Balancing Schemes: Theory to Algorithms. Proceedings of the ACM on Measurement and Analysis of Computing Systems. 2017. Т. 2, № 1. С. 1–30.

19. Алексєєв М. О., Гусак О. В. КЛАСИФІКАЦІЯ АЛГОРИТМІВ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ВИКОРИСТАННЯ В РОЗПОДІЛЕНИХ ГЕТЕРОГЕННИХ МЕРЕЖАХ. м. Київ, 2016 р. Київ, 2016. С. 3.

20. Choosing the Right Load Balancing Algorithm for Your Needs. Intellipaat. URL: <https://intellipaat.com/blog/load-balancing-algorithms/> (дата звернення: 15.04.2024).

21. Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing – Journal of Cloud Computing. SpringerOpen. URL: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-023-00453-3> (дата звернення: 15.04.2024).

22. Load Balancing Algorithms – GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/load-balancing-algorithms/> (дата звернення: 15.04.2024).

23. COMPARATIVE STUDY ON LOAD BALANCING TECHNIQUES IN DISTRIBUTED SYSTEMS. ResearchGate. URL: https://www.researchgate.net/figure/Comparison-of-Basic-Static-Load-Balancing-Algorithms_tbl1_266560116 (дата звернення: 15.04.2024).

24. Load Balancing Algorithms. EnjoyAlgorithms. URL: <https://www.enjoyalgorithms.com/blog/types-of-load-balancing-algorithms> (дата звернення: 15.04.2024).

25. Glushach R. Kubernetes Networking: Load Balancing Techniques and Algorithms. Medium. URL: <https://romanglushach.medium.com/kubernetes-networking-load-balancing-techniques-and-algorithms-5da85c5c7253> (дата звернення: 15.04.2024).

26. Experimental Setup for Investigating the Efficient Load Balancing Algorithms on Virtual Cloud. Sensors. 2020. Т. 20, № 24. С. 20–24.

27. Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing. Procedia Computer Science. 2015. Т. 57, № 1. С. 545–553.

28. Benefits of Software-Based Hybrid Load Balancing | NGINX. NGINX. URL: <https://www.nginx.com/resources/glossary/hybrid-load-balancing/> (дата звернення: 15.04.2024).

29. Combining Multiple Load Balancing Strategies – FasterCapital. FasterCapital. URL: <https://fastercapital.com/keyword/combining-multiple-load-balancing-strategies.html> (дата звернення: 15.04.2024).

30. Source IP Hash load balancing – Glossary – Kemp. Load Balancer For Always-On Application Experience – Kemp. URL: <https://kemptechnologies.com/es/resources/glossary/source-ip-hash-load-balancing> (дата звернення: 15.04.2024).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії**

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
на тему: «Дослідження алгоритмів балансування
навантаження для розподілу трафіку у великих
мережах»**

Виконав студент групи КІД-42
Ткачук Владислав Олександрович
Керівник кваліфікаційної роботи :
Бученко Ігор Анатолійович ,
старший викладач кафедри КІ

Київ – 2024

Мета роботи – дослідження алгоритмів балансування навантаження для великих мереж на поточному етапі розвитку технологій.

Об'єкт дослідження – розподіл навантаження трафіку у великих мережах за допомогою алгоритмів балансування навантаження.

Предмет дослідження – алгоритми балансування навантаження для розподілу трафіку у великих мережах.

Актуальність обраної теми

Алгоритми балансування навантаження для розподілу трафіку у мережі на даний момент використовуються в багатьох галузях, для розподілення навантаження в мережі. Використання алгоритмів балансування навантаження зараз і в подальшому розвитку мереж дозволить покращити можливості підприємств що використовують такі алгоритми. Використання алгоритмів балансування навантаження мережі значно збільшить ефективність використання обчислювальних ресурсів, що також дозволить значно простіше розширяти та модифікувати мережу.

Продовження актуальності обраної теми

Алгоритми балансування навантаження неодмінно мають велику кількість переваг, проте, існує й ряд недоліків даної технології, що можуть погіршити досвід користування такими рішеннями.

На основі проведеного дослідження було виявлено та проаналізовано чималу кількість рішень по балансуванню навантаження мережі, що дозволяє дійти точних висновків про використання алгоритмів балансування навантаження для розподілу трафіку у великих мережах.

Типи алгоритмів балансування

Виділяють два основні типи алгоритмів балансування навантаження – статичні та динамічні.

Існує також третій тип – адаптивні алгоритми. Проте ці алгоритми зазвичай рідко використовуються так як є концептуальними поєднаннями двох попередніх типів алгоритмів балансування.

Основні алгоритми балансування

Існує велика кількість різноманітних алгоритмів балансування навантаження, що підходять для різних типів мереж.

Наприклад, Round Robin, Source IP/URL Hash, Opportunistic – статичні алгоритми.

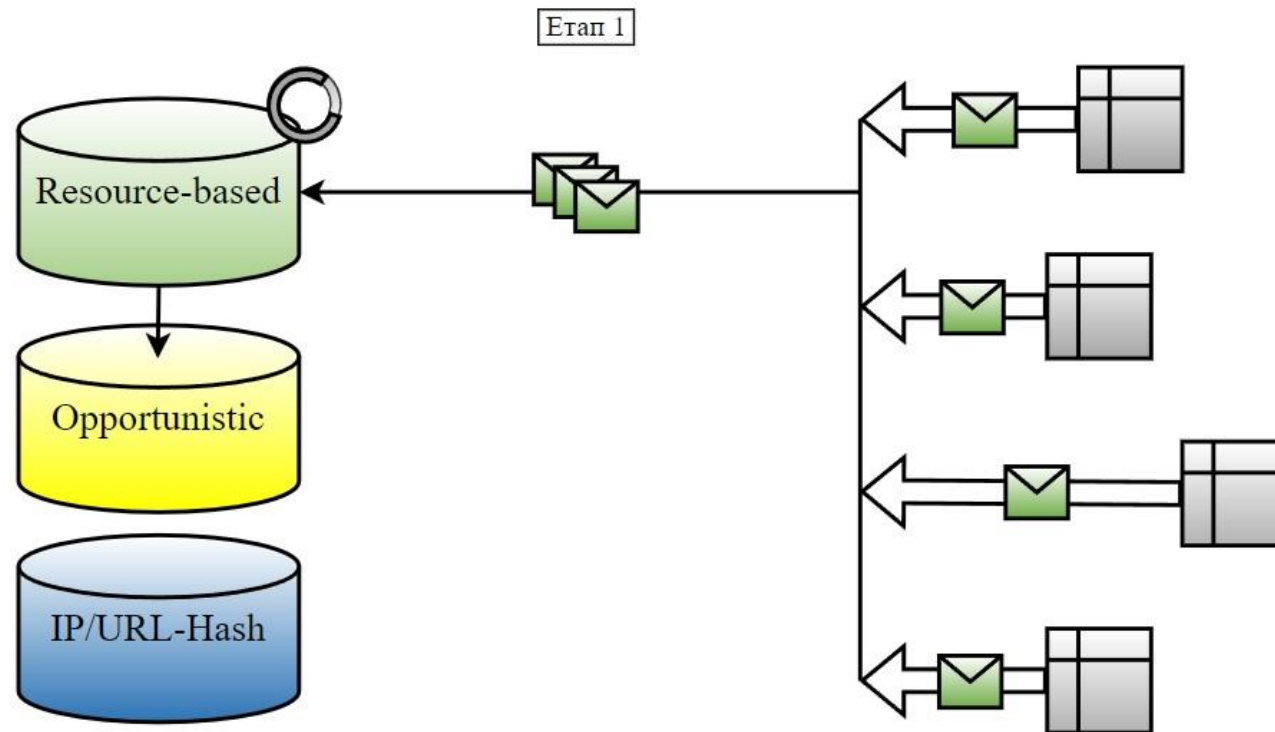
Алгоритми Least Connection, Response Time, Resource-based – динамічні.

Методи комбінування алгоритмів

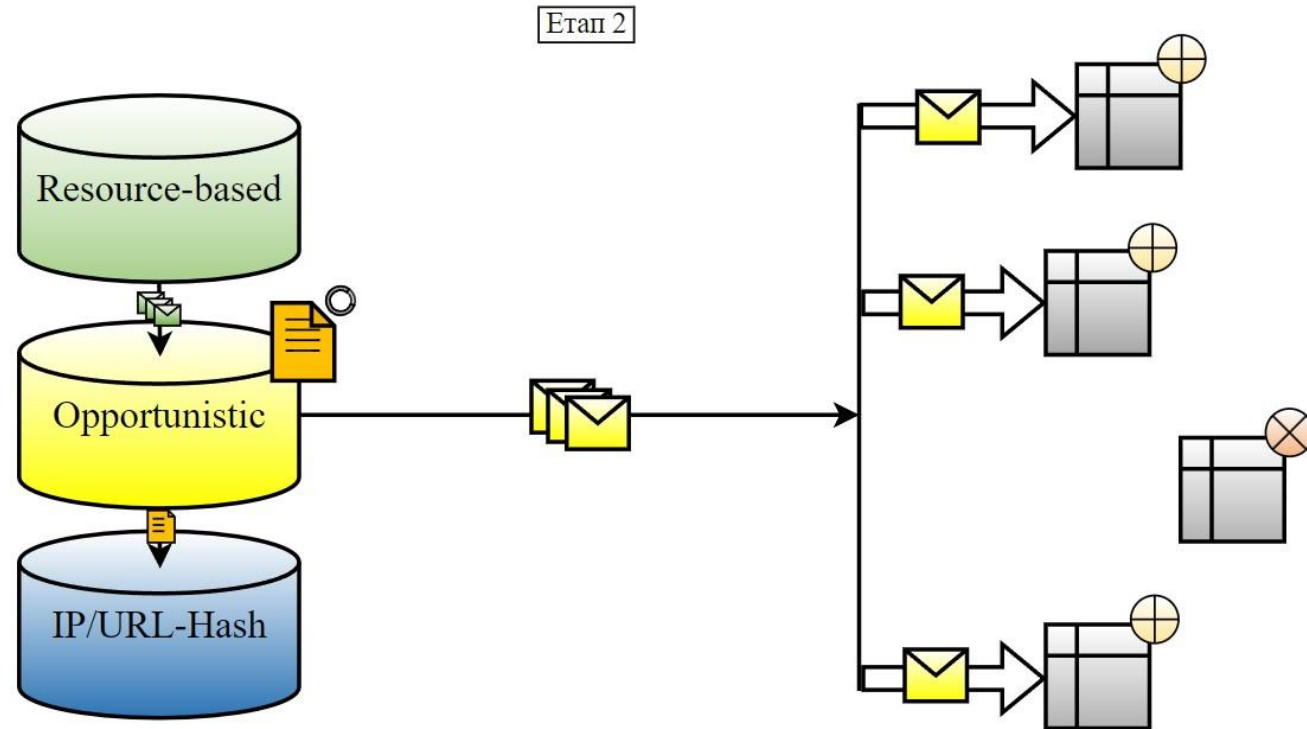
Для вирішення проблемного питання з балансування навантаження у великих мережах в дослідженні використовується метод комбінування алгоритмів балансування навантаження.

Зазвичай використовують два основні методи комбінування – послідовне та вагове комбінування.

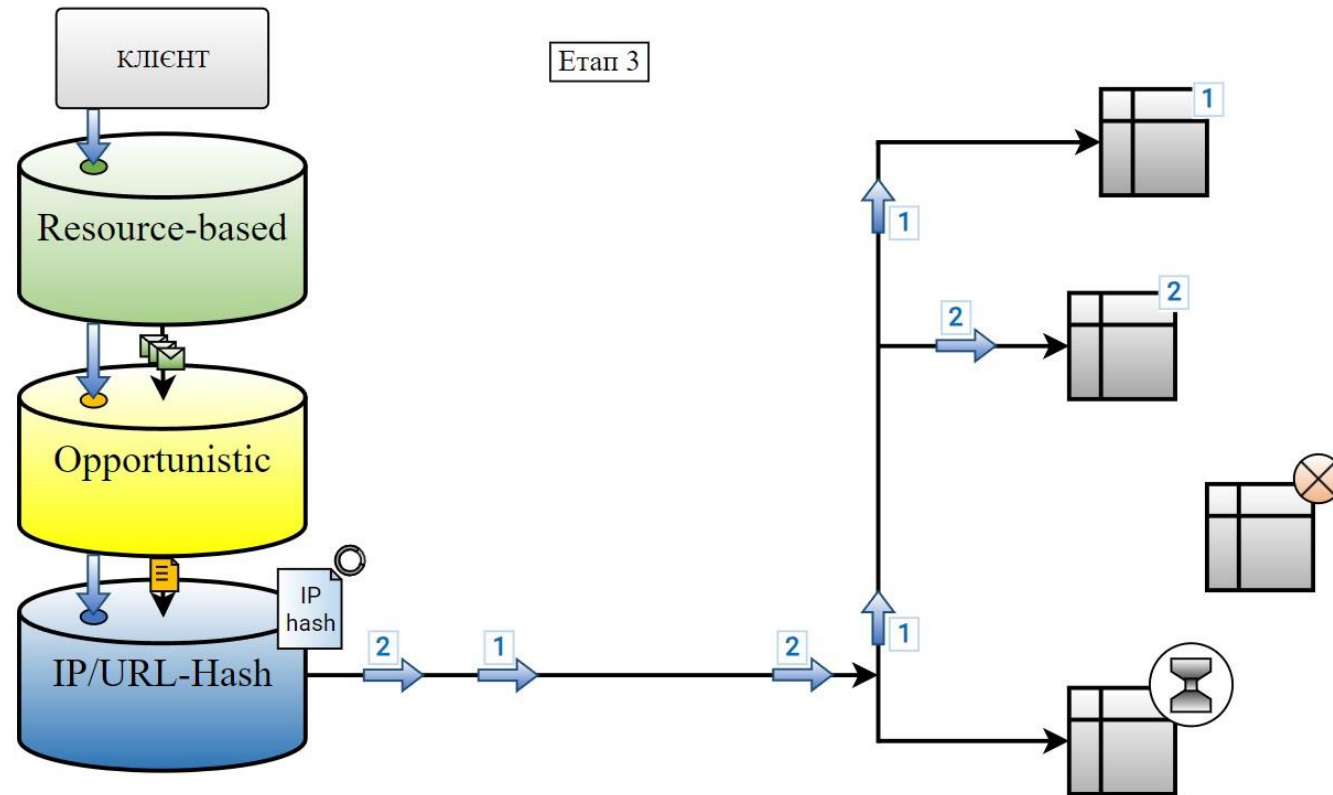
Етапи роботи запропонованого рішення №1



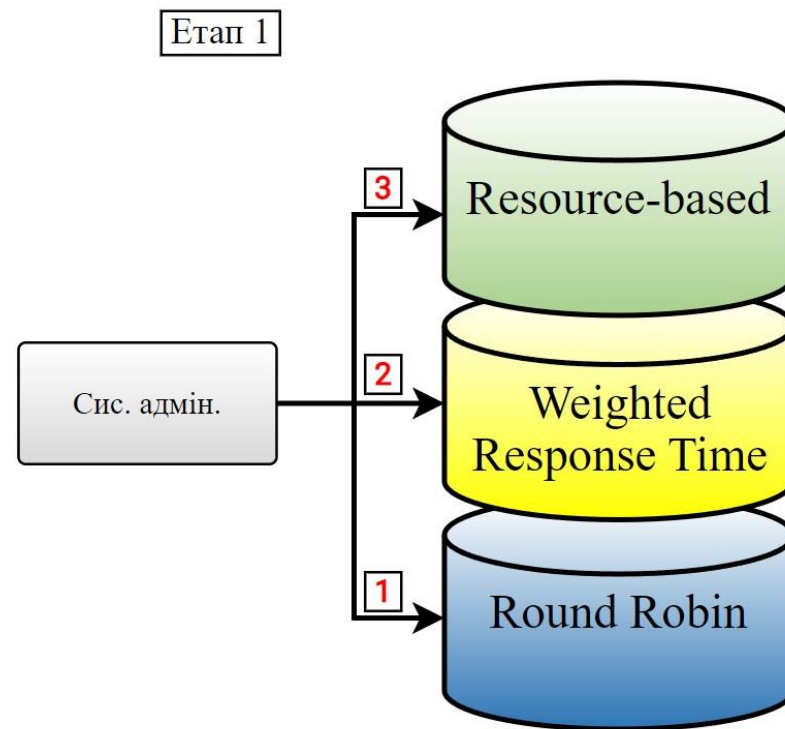
Етапи роботи запропонованого рішення №1



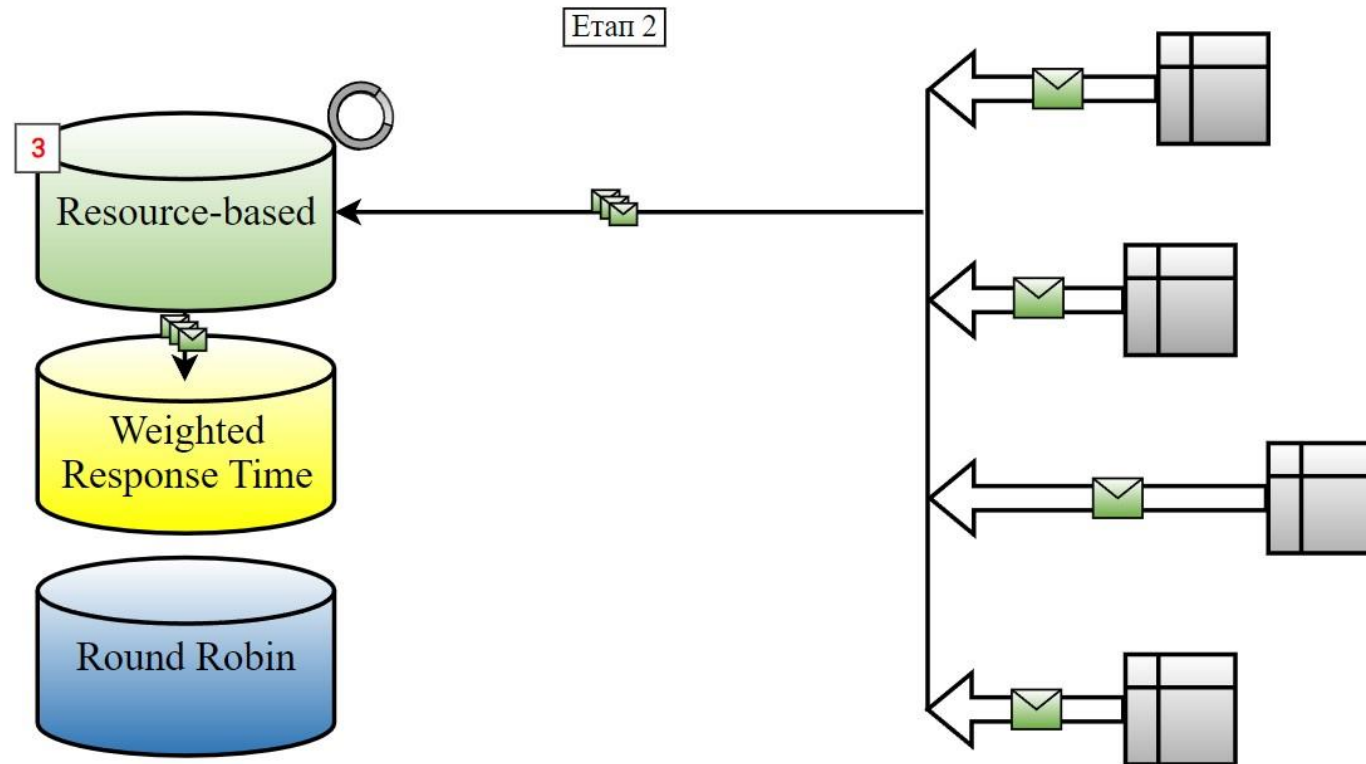
Етапи роботи запропонованого рішення №1



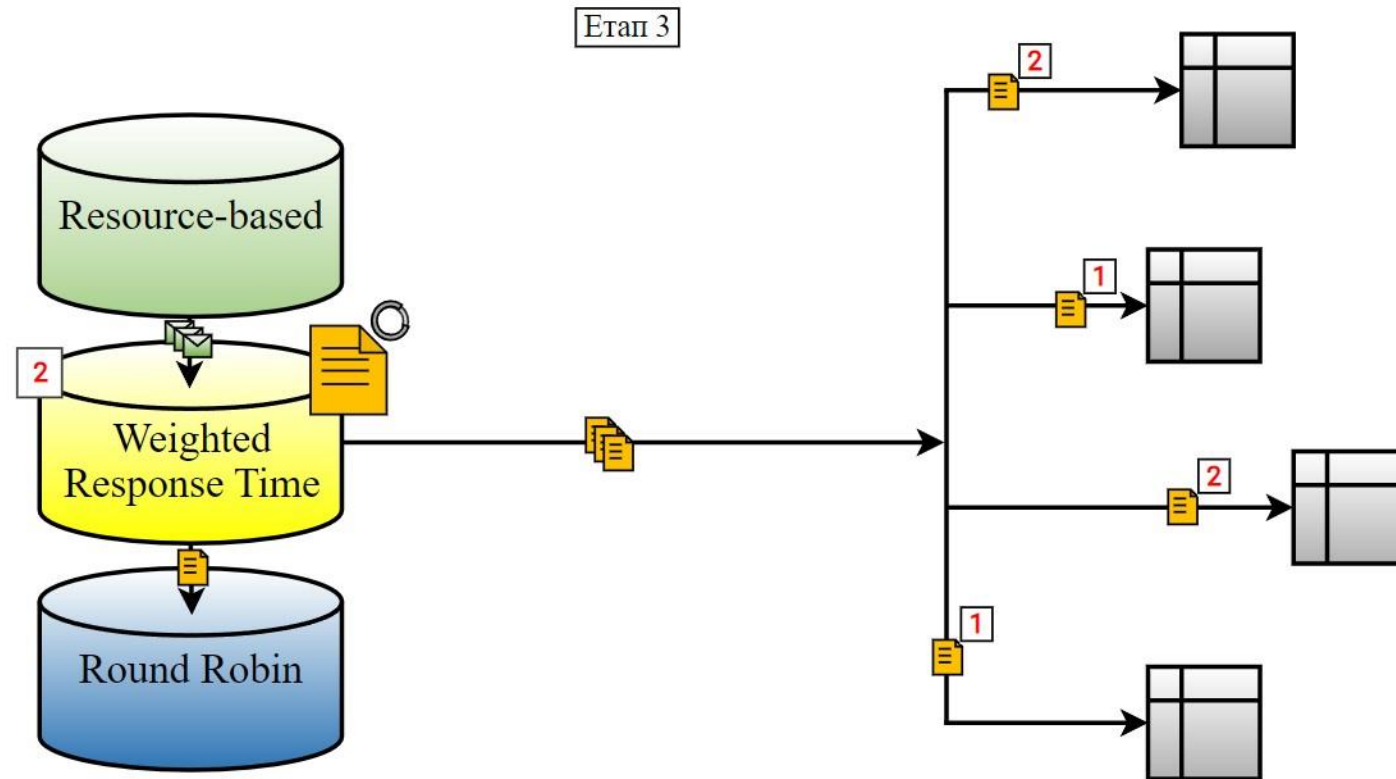
Етапи роботи запропонованого рішення №2



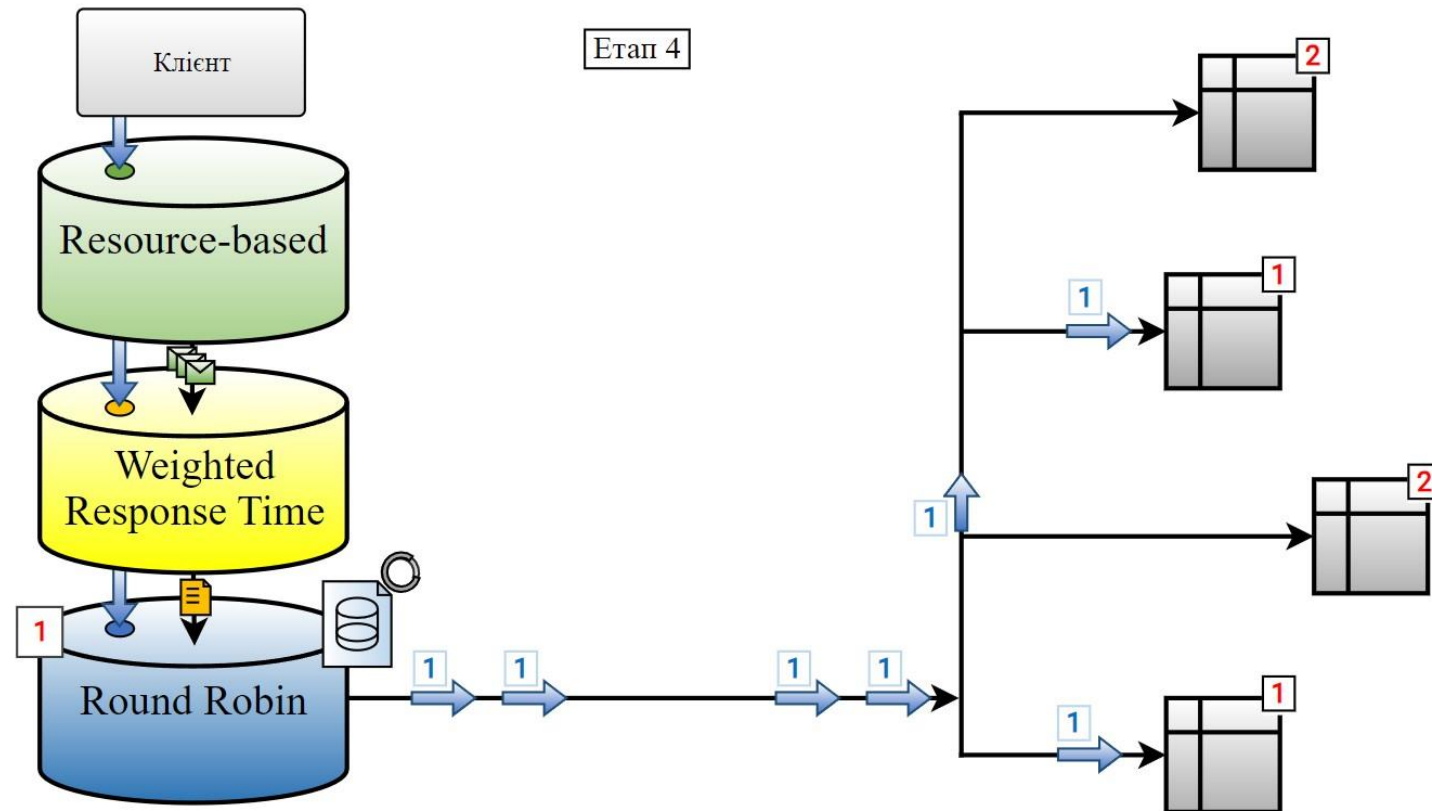
Етапи роботи запропонованого рішення №2



Етапи роботи запропонованого рішення №2



Етапи роботи запропонованого рішення №2



ВИСНОВКИ

В даній роботі були опрацьовані питання дослідження та аналізу існуючих рішень балансування навантаження трафіку, були проаналізовані різні теорії та дослідження на тему алгоритмів балансування навантаження в мережі, на основі цих досліджень були приведені висновки їх ефективності та доцільності використання. Були розроблені та опрацьовані проблемні питання у сфері балансування навантаження трафіку в мережах, також було опрацьовано методичні підходи для дослідження та аналізу підходящих алгоритмів балансування навантаження для розподілу трафіку у великих мережах, що дало змогу розкрити питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах, проаналізувати та структурувати результати цього дослідження. Було проведено визначення методів комбінації алгоритмів балансування навантаження, з метою розширення можливостей мережі.

ПРОДОВЖЕННЯ ВИСНОВКІВ

Основною метою дослідження цієї теми, було вирішення проблемного питання алгоритмів балансування навантаження для розподілу трафіку у великих мережах. Після досліджень великої кількості публікацій, статей та досліджень інших наукових авторів було досягнуто кінцевого висновку по цьому питанні, що дало змогу вирішити найбільш релевантні рішення балансування навантаження у великих мережах, ними виявилися рішення з комбінуваннями певних алгоритмів за певними схемами комбінування. Було наведено два можливих рішення балансування навантаження трафіку, що найбільше підходять до деяких умов великих мереж та ефективно виконують свої поставлені задачі.

Результати впровадження

1. Бученко І. А., Ткачук В. О. СУЧАСНІ АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ МЕРЕЖІ. «Сучасний стан та перспективи розвитку IoT» : V міжнародна науково-технічна конференція, м. Київ, 18 квітня 2024 р. С. 1–2. URL: <https://duikt.edu.ua/ua/news-1-570-12409-v-mizhnarodna-naukovo-tehnichna-konferenciya-suchasniy-standa-perspektivi-rozvitku-iot-kafedra-inzhenerii-programnogo-zabezpechennya-avtomatizovanih-sistem> (дата звернення: 18.04.2024).
2. Бученко І. А., Ткачук В. О. АЛГОРИТМИ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ РОЗПОДІЛУ ТРАФІКУ У ВЕЛИКИХ МЕРЕЖАХ. «Діджиталізація науки як виклик сьогодення» : VI Міжнародна студентська конференція, м. Суми, 3 травня 2024 р. С. 1–3. URL: <https://archive.liga.science/index.php/conference-proceedings/issue/view/inter-03.05.2024> (дата звернення: 03.05.2024).