УДК 621.3

АНАЛІЗ РОБОТИ МЕТОДУ ОПТИМІЗОВАНОГО КЕШУВАННЯ ДАНИХ В МЕРЕЖІ ДОСТАВКИ КОНТЕНТУ

М.І. Кирик, Н.М. Плесканка, М.В. Плесканка
Національний університет «Львівська політехніка»

Abstract *– Concept of Content Delivery Network (CDN) system and methods of caching data on edge servers were considered in this paper. The main task of the CDN network is providing the qualitative information delivery to the end user. Content Delivery Network is a geographically distributed network, that contain a number of content servers and routers. As a rule, it consists of a main node (Origin), and caching nodes (Edges) – points of presence, which can be located in various parts of the world. All content is stored and updated on the Origin server. The structural scheme of organizing the work of the CDN network, consisting of three complementary planes, was presented. Each plane performs special functions and has its own implementation features. Access Plane provides end-user access to content and the interaction of caching servers among themselves and on other planes. The load balancing plane is organized according to the principle of the DNS service and allows to redirect the user's requests to the nearest caching servers. This plane also interacts with the metric plane, which contains up-to-date information of the status of all servers involved in delivery and caching of data. As part of this paper, we proposed a method for optimized caching of data in the CDN network. The algorithm of this method is presented. Dependencies that make it possible to evaluate the work of this method in comparison with existing methods are given. The proposed method allows to significantly improve the Hit Ratio, and thus ensure efficient use of cached data and reduce the load on servers of origin.*

Анотація *– У даній роботі представлено структурну схему організації роботи CDN у вигляді трьох взаємодоповнюючих площин. Кожна із площин описує певну функціональну складову та взаємодіє із іншими площинами. Запропоновано використання нового методу оптимізовагого кешування даних як складої площини балансування навантаження. Даний метод дає можливість ефективніше використовувати кешування даних, не збільшуючи навантаження на сервери походження. Метод забезпечує ефективне використання механізму кешування, максимізує рівень використання кеша (Hit Ratio) та контролює рівень якості послуг для кінцевого користувача. Представлено аналіз результатів, які відображають переваги запропонованого методу у порівнянні зі стандартними способами кешування даних.*

Аннотация *– В данной работе представлена структурная схема организации работы CDN в виде трех взаимодополняющих плоскостей. Каждая из плоскостей описывает определенную функциональную составляющую и взаимодействует с другими плоскостями. Предложено использование нового метода оптимизированного кэширования данных, как составляющей плоскости балансировки нагрузки. Данный метод дает возможность более эффективно использовать кэширование данных, не увеличивая нагрузку на серверы происхождения. Метод обеспечивает эффективное использование механизма кэширования, максимизирует уровень использования кэша (Hit Ratio) и контролирует уровень качества услуг для конечного пользователя. Представлен анализ результатов, отражающих преимущества предложенного метода по сравнению со стандартными способами кэширования данных.*

Вступ

В наш час використання інфокомунікаційних мереж для обміну інформацією є дуже поширеним у всіх сферах людського повсякденного життя. Кількість пристроїв, які можуть працювати через мережу, виходити в Інтернет є дуже великою, а обсяги даних, що передаються, постійно збільшуються. Саме тому, актуальними постають питання ефективності використання мережних ресурсів, технологій ефективної передачі даних, розподілу навантаження, а також швидкості та надійності доставки інформації кінцевим користувачам [1, 2].

На сьогоднішній день існує багато технологій, які забезпечують надійність і контроль якості доставки даних, а також дають можливість ефективно використовувати

ресурси мережі [3-5]. Одним із таких рішень є технологія мереж доставки контенту (Content Delivery Network, CDN), яка працює за принципом кешування контенту та вибору найкращого, доступного для користувача, граничного сервера. Для цього існує багато методів і способів організації. В даній роботі буде запропоновано метод оптимізованого кешування, досліджено принцип його роботи та приведено порівняльну оцінку з іншими методами кешування та доставки.

I. Функціональна схема організації роботи CDN

Мережа доставки (розподілу контенту) CDN, представляє собою географічно рознесену мережу передавання інформації, яка складається із серверів обробки, кешування та трансляції контенту, а також мережних маршрутів [6-8]. Основним завданням такої мережі є забезпечення якісної та надійної доставки інформації до кінцевого користувача. Організацію роботи CDN можна представити у вигляді функціональної багаторівневої схеми. Схема включає в себе площину доступу до контенту, площину дистрибуції контенту, площину оцінки метрик та станів сервісів контролю якості послуг. Структурну схему моделі представлено на рис. 1. Кожна із представлених площин виконує ряд функцій та має свої особливості роботи та реалізації. Далі буде більш детально описано роботу кожної із представлених площин.

Рис. 1. Структурна схема організації роботи CDN

Площина доступу до мережі CDN

Площина доступу організовує взаємодію кінцевих користувачів з кешуючими серверами, взаємодію подібних серверів між собою та із сервером контент-провайдера (Origin server). Детальну схему організації даної платформи представлено на рис. 2.

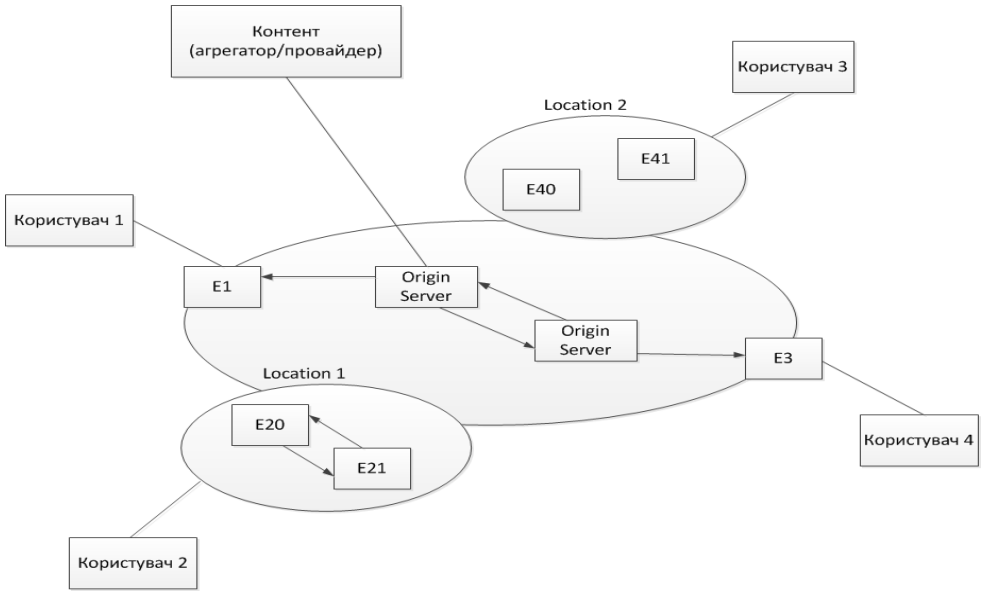

Рис. 2. Схема організації платформи доступу до мережі CDN

Для підвищення ефективності використання мережних ресурсів кешуючі сервери (Edge server) можуть об'єднуватись в локації та взаємодіяти один з одним в межах спільної локації, не звертаючись до сервера контенту. Сервер провайдера контенту для підвищення надійності та доступності рекомендовано дублювати та реплікувати з метою балансування навантаження. Це дасть змогу при недоступності одного із серверів забезпечити безперебійну роботу сервісу. Запити кінцевих користувачів завжди попадатимуть до кешуючого сервера, який знаходиться в найближчій до нього локації. Кількість кешуючих серверів визначатиметься в залежності від навантаження, а також необхідності доступності ресурсів і контенту в окремій географічній локації. Кількість даних, що буде зберігатись на кешуючих серверах, визначатиметься їх продуктивністю. Чим продуктивнішим буде сервер, тим більше інформації та довший час він зможе зберегти її в локальному кеші.

Платформа доступу взаємодіє з модулем перенаправлення на базі DNS серверів. Схема взаємодії зображена на рис. 3. Модуль перенаправлення запитів користувачів організований на базі Anycast DNS. Всі запити на отримання контенту першочергово будуть надходити саме на цей модуль, а далі вже він визначатиме, на базі двосторонньої взаємодії, до якого кешуючого сервера перенаправити клієнта для подальшого обслуговування. Платформа доступу являє собою сукупність кешуючих серверів, які певним чином з'єднані між собою. Як бачимо із рис. 3, використовується один Master DNS та кілька Slave. Така схема запропонована з метою балансування навантаження та підвищення надійності роботи сервісу. Як можна побачити з рис. 3, DNS-сервери взаємодіють також з базою метрик кешуючих серверів та з системою оцінки їхніх станів. База метрик містить інформацію про автономні системи, геолокацію серверів, а також дані про затримки до шлюзів мережних провайдерів.

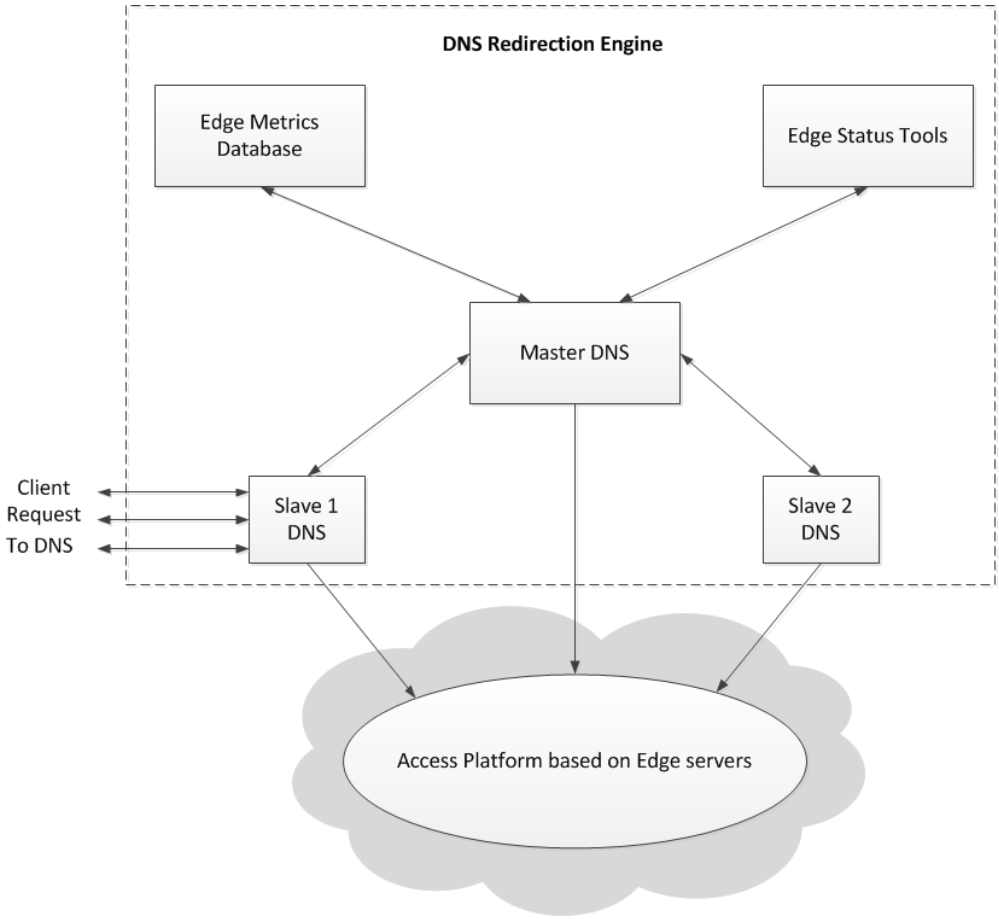

Рис. 3. Взаємодія платформи доступу із модулем перенаправлення запитів користувачів

Саме такі параметри як відстань між автономними системами та час затримки будуть визначати метрику кешуючих серверів. Важливо вказати, як буде вимірюватись значення метрики, щоб сервер перенаправлення заздалегідь міг знати, до якого кешуючого сервера направити запит кінцевого користувача. В нашому випадку пропонується використовувати базу метрик, яка буде спільною для кешуючих серверів. База буде містити список автономних систем із IP-адресами шлюзів із цих систем, які розміщені в різних дата центрах світу. Кожен кешуючий сервер буде отримувати список цих адрес, спеціальний процес на сервері звертатиметься до цих адрес і визначатиме затримку та відстань. Всі ці дані будуть записуватися в ту ж саму базу метрик. Як результат, в базі даних метрик буде міститись інформація про відстані та затримки між усіма кешуючими серверами та усіма шлюзами автономних систем. Також кешуючі сервери будуть зв'язуватись із сервером походження та звітувати про цей статус у базу даних метрик.

Система статусів постійно контролює стан серверів віддачі контенту, їхню доступність, завантаженість, статус сервісів, які відповідають за опрацювання даних, та завжди містить актуальну інформацію про кожен із серверів. Система контролю містить в собі систему моніторингу, яка працює як окремий сервіс, та сервіси на кешу-

ючих вузлах, які звітують системі моніторингу про стан сервісів на самих серверах. Одними із найнеобхідніших та найважливіших сервісів, які слід моніторити, є втрата пакетів на мережних інтерфейсах, завантаженість обслуговуючих пристроїв, стан пам'яті, стан файлової системи, стан сервісів, які відповідають за обслуговування запитів клієнтів. Всі ці дані зберігаються в базі даних системи моніторингу. Система перенаправлення запитів, перш ніж перенаправляти запит користувача до конкретного кешуючого сервера, звернеться до системи моніторингу та перевірить його статус. Якщо всі сервіси працюють та сервер доступний, це свідчитиме що даний сервер придатний до обслуговування запитів кінцевих користувачів.

Саме завдяки такій організації модуль перенаправлення завжди буде містити актуальні дані про доступність і статус усіх серверів і завжди буде обирати оптимальний по відношенню відстані та часу затримки сервер для обслуговування запитів користувачів із різних географічних локацій.

Використання технології CDN для доставки контенту безумовно має ряд переваг. Покажемо на прикладі, як відрізнятиметься час завантаження одного і того ж файлу від сервера походження та від кешуючого сервера. Наприклад, час завантаження файлу від керуючого CDN сервера можна контролювати наступним чином:

```
[#] nazar@cahce1:~$wget http://vscdntraining.tk/wp-content/upload/files/head.jpg
Resolving vscdntraining.tk... 204.100.253.9, 204.100.253.8
Connecting to vscdntraining.tk |204.100.253.9|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 432113 (432K) [image/jpeg]
Saving to: head.jpg
100%[=============================================>] 432,113     --.-K/s   in 0.05s
2018-08-02 12:17:26 (4.52 MB/s) - head.jpg saved  [432113 / 432113]
```

Як можна побачити із даного прикладу, час завантаження файлу становить 0,05 с. Для порівняння проведемо завантаження того ж файлу із сервера походження (Origin):

```
[#]user1@gateway1:~$ wget http://cdntraining.tk/wp-content/upload/files/head.jpg
Resolving cdntraining.tk... 19.158.117.115
Connecting to cdntraining.tk |19.158.117.115|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 432113 (432K) [image/jpeg]
Saving to: head.jpg
100%[=======================================>] 432,113     --.-K/s   in 0.18s
2018-08-02 12:19:12 (1.64 MB/s) - head.jpg saved  [432113 / 432113]
```

Як можна побачити із представлених результатів, розмір файлу в обох випадках є однаковим, а час завантаження файлу із сервера походження становить 0,18 с. Це

практично у 4 рази довше, аніж із кешуючого сервера, і свідчить про те, що кінцевий користувач зможе отримати контент набагато швидше незалежно від свого місця розташування. Це є надзвичайно важливим фактором, оскільки кожен контент-провайдер намагається забезпечувати найкращу якість своїх послуг для кінцевого користувача.

II. Структура методу оптимізованого кешування даних

Структурна схема роботи пропонованого методу оптимізованого кешування даних представлена на рис. 4. Як можна побачити на рис. 4, в структурній схемі також передбачено можливість передавання даних без використання CDN. За таких умов всі запити кінцевих користувачів будуть напряму надсилатись до сервера походження (Origin). В такому варіанті реалізації на стороні сервера походження рекомендовано застосовувати технологію балансування навантаження та резервування контенту HA Proxy Load Balancer [9, 10]. Така система являє собою високонадійну систему, яка підвищить якість надання послуг користувачам. Вона дає можливість використовувати три та більше серверів, запити на які будуть збалансовано надсилатись балансувальником. Він у свою чергу буде контролювати доступність сервера, його завантаженість та час відклику. Можливий також варіант поєднання технології CDN та технології HA Proxy Load Balancer. Як правило, CDN використовується для передавання статичного контенту, саме тому в даному варіанті реалізації статичні дані передаватимуться через CDN, а динамічні – через HA Proxy Load Balancer.

На рис. 4 CDN Operate Module виконує завдання перенаправлення запитів користувачів на кешючі сервери з найкращими параметрами якості обслуговування по відношенню до локації користувача.

CDN Core – це саме структура розміщення датацентрів у різних точках світу, в яких знаходяться кешуючі сервери, що обслуговують та зберігають контент, який запитують кінцеві користувачі. Від їхньої кількості залежить рівень та якість доступності сервісу в конкретному регіоні.

Hit Ratio Optimization Module – основна функція даного модуля: підвищити рівень використання кешованих даних і тим самим забезпечити оптимальне використання механізму кешування. Коли приходить перший запит від користувача за контентом у конкретній локації, він буде перенаправлений на кешуючий сервер із оптимальними параметрами QoS (мінімальна затримка, максимальна надійність сервера). Далі, механізм перевірить, чи є вже запитуваний контент у кеші сусідніх кешуючих серверів, що знаходяться у цій же локації. Якщо виявиться, що є сервер який містить запитуваний контент, то запит буде передано на обслуговування на цей кеш-сервер, з метою економії трафіку від сервера походження та збільшення рівня використання кеша. Після передачі обслуговування також буде проводитись визначення параметрів QoS, чи задовольняють вони якість заданого сервісу. Якщо користувач отримує сервіс із задовільною для нього якістю та допустимими параметрами QoS, то передача продовжуватиметься із конкретного сервера. Якщо ж параметри

QoS не будуть задовільними, то модуль делегує іншому керуючому серверу обслуговування запитів даного користувача. Таким чином, даний модуль зможе забезпечити ефективне використання механізму кешування, максимізувати рівень використання кеша (Hit Ratio) та контролювати рівень якості послуг для кінцевого користувача.

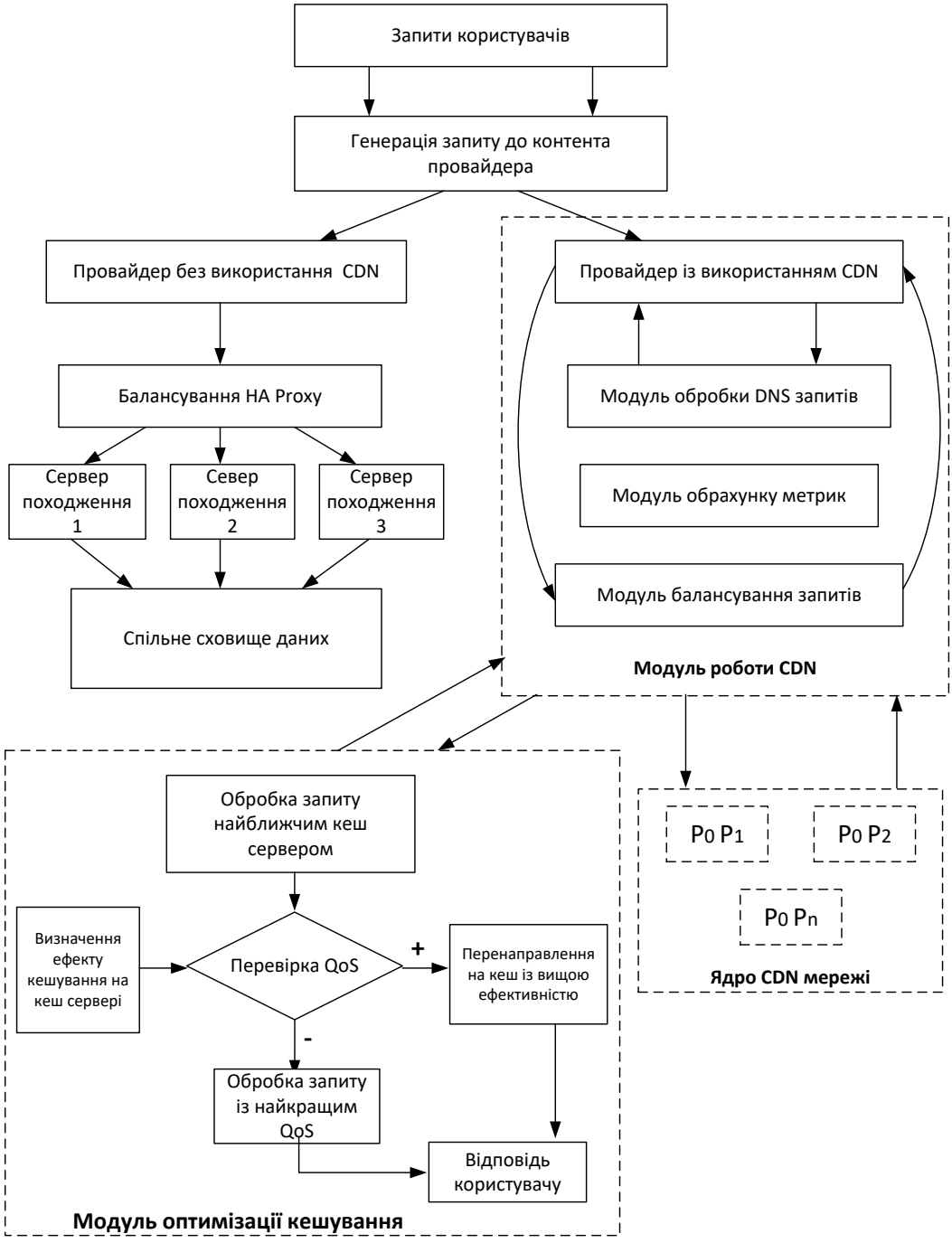

Рис. 4. Структурна схема роботи методу оптимізованого кешування даних в мережі CDN

Оскільки, одним із основних завдань CDN є зменшення часу затримки доставки контенту до кінцевого користувача, розглянемо, як саме визначати цей параметр. В будь якій реалізації CDN мережі кешуючі сервери обслуговують велику кількість за-

питів кінцевих користувачів. Будемо вважати, що $\lambda_i, i = 1...n$ – це інтенсивність надходження запитів на *i*-й сервер в момент часу *t*. В такому випадку сумарна інтенсивність надходження запитів у мережі буде визначатись як

$$\lambda = \sum_{i=1}^{n} \lambda_i \quad . \tag{1}$$

Варто зазначити, що окрім власних запитів, кожен кешуючий сервер може також отримувати запити від інших, сусідніх серверів. Очевидно, що процент завантаженості серверних ресурсів, які виділяються на опрацювання запитів від інших серверів, буде залежати від багатьох факторів і може бути пріоритезованим. Таким чином, інтенсивність запитів, які надходять від сусідніх серверів CDN мережі, буде визначатись за наступною формулою:

$$\sum_{j=1}^{n} \lambda_j \omega_{ji} \, , \tag{2}$$

де ω_{ji} – частина потоку λ_j , яка перенаправляється від *j*-го до *i*-го сервера.

В результаті середня інтенсивність вхідного навантаження, що надходить на *i*-й кешуючий сервер, може бути визначено за формулою:

$$\alpha_i = \lambda_i - \sum_{\substack{j=1, \\ j \neq i}}^{n} \lambda_i \omega_{ij} + \sum_{\substack{j=1, \\ j \neq i}}^{n} \lambda_j \omega_{ji} \, . \tag{3}$$

Другий доданок в правій частині виразу (3) визначає частину навантаження, яке *i*-й кешуючий сервер переадресовує до інших серверів із своєї локації, третій доданок – частину навантаження, яке було отримано цим сервером від сусідніх серверів із його локації. Інтенсивність обробки запитів μ_i *i*-м сервером кешування має бути більшою, ніж інтенсивність надходження запитів λ_i. Будемо вважати, що потік запитів, що надходять на кешуючі сервери, є пуассонівським. Кожен сервер розглядається як система масового обслуговування типу M/M/1 [11-13]. В такому випадку середній час затримки обслуговування запиті буде визначатись як

$$T = \frac{1}{\lambda} \cdot \sum_{i=1}^{n} \frac{a_i}{\mu_i - a_i} \, , \tag{4}$$

де $\lambda = \sum_{i=1}^{n} \lambda_i$.

III. Аналіз результатів роботи методу оптимізованого кешування даних

Запропонований у даній роботі метод оптимізованого кешування даних в CDN в експериментальних цілях був запроваджений у реально діючій мережі провайдера. Експерименти проводились протягом одного місяця. Результати експериментів по-

дані на рис. 5 і 6. На рис. 5 представлено результати роботи запропонованого методу оптимізованого кешування у порівнянні із роботою CDN без його використання (рис. 6).

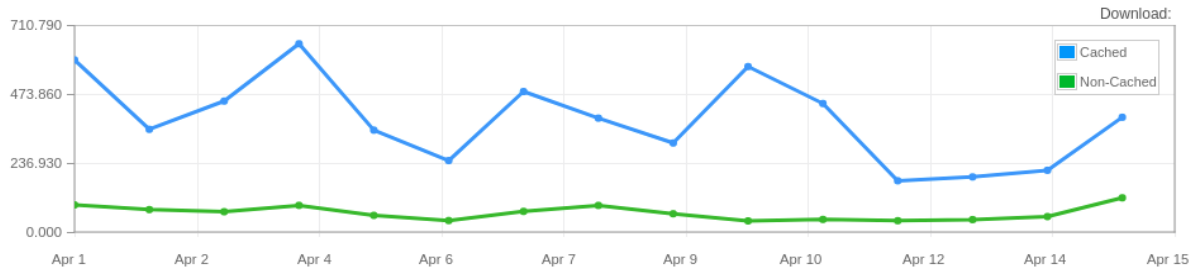

Рис. 5. Ефективність використання кеша із застосуванням методу оптимізованого кешування

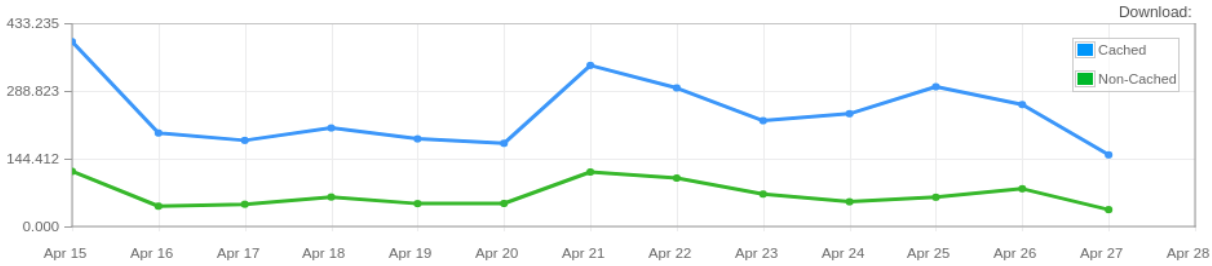

Рис. 6. Ефективність використання кеша без застосування методу оптимізованого кешування

Як можна побачити із рис. 5 та 6, трафік від сервера походження залишається незмінним, однак ефективність кешування (використання кеша) суттєво відрізняється. Результати роботи даного методу показують, що ефективність використання кешування суттєво зростає, а навантаження на сервер походження та трафік від нього залишаються практично без змін. При максимальній інтенсивності трафіку від сервера походження 140 Мбіт/с та використанні методу оптимізованого кешування, максимальна інтенсивність кешованих даних становила 680 Мбіт/с, а без використання методу 428 Мбіт/с. На рис. 7 та 8 представлено інтенсивність трафіка від сервера походження протягом інтервалу проведення експерименту.

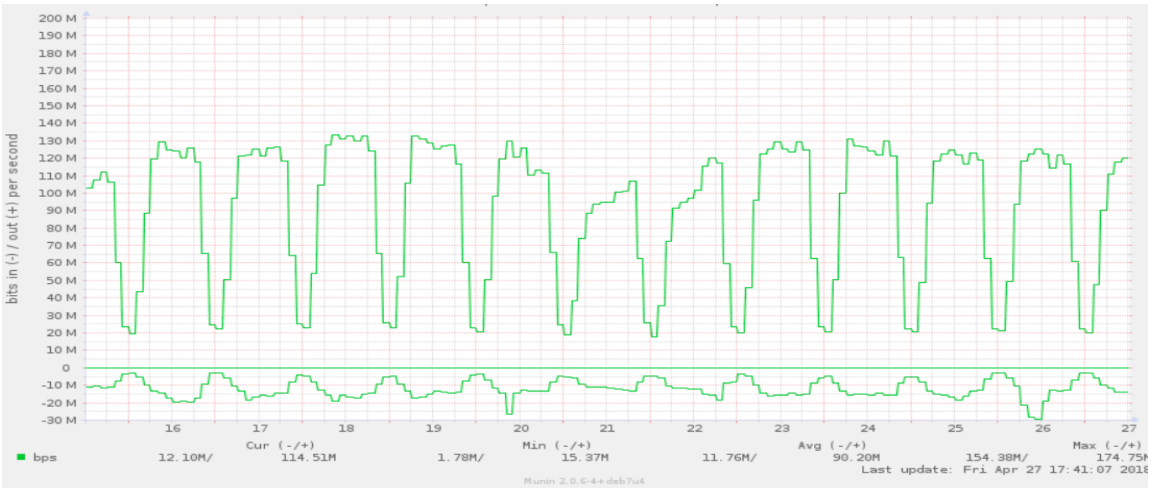

Рис. 7. Інтенсивність трафіку від сервера походження із використанням методу оптимізованого кешування

Рис. 8. Інтенсивність трафіку від сервера походження без використанням методу оптимізованого кешування

Було проведено аналіз впливу використання пропонованого методу на завантаженість сервера походження за час проведення експерименту. За оцінку було взято час відповіді сервера на запити кешуючих серверів та кількість TCP-з'єднань від серверів кешування. Результати представлено на рис. 9 – рис. 12.

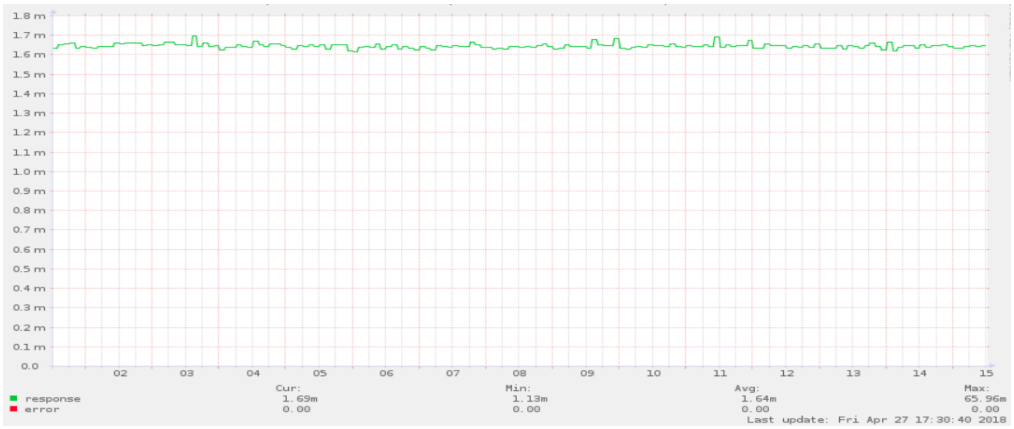

Рис. 9. Час відповіді сервера походження із використанням методу оптимізованого кешування

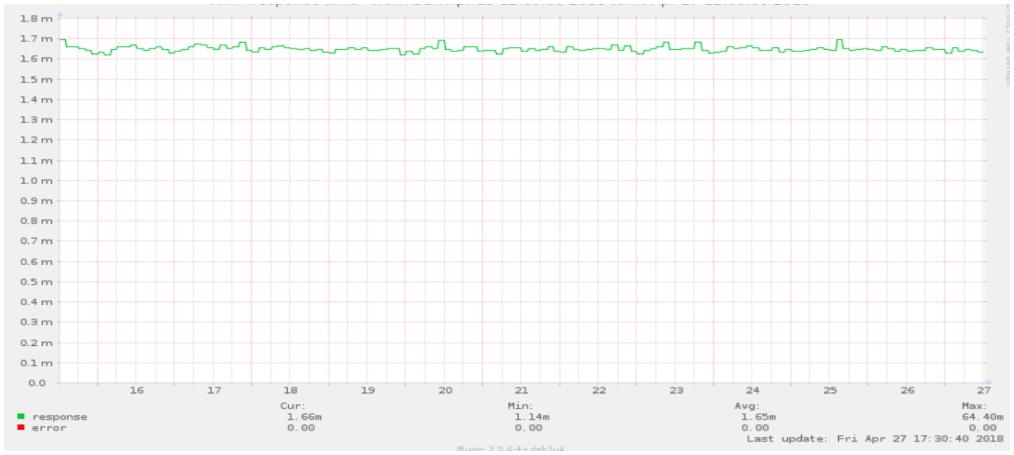

Рис. 10. Час відповіді сервера походження без використанням методу оптимізованого кешування

Як показано на рис. 9 і рис. 10, за весь період експерименту час відповіді суттєво не змінювався. Кількість TCP-з'єднань до сервера походження зображено на наступних рис. 11 та рис. 12.

Рис. 11. Кількість TCP-з'єднань до сервера походження без використання методу оптимізованого кешування

Рис. 12. Кількість TCP-з'єднань до сервера походження з використанням методу оптимізованого кешування

Як показано на рис. 11 і рис. 12, число TCP-з'єднань також суттєво не змінюється на всьому інтервалі проведення експерименту. Тому можна стверджувати, що використання запропонованого методу дає змогу практично в два рази покращити ефективність використання кешування даних, не збільшуючи навантаження на сервер походження. Рівень якості сервісу при цьому залишатиметься в допустимих межах.

Висновки

В даній роботі розглянуто концепцію організації роботи мережі доставки контенту, а також основні принципи балансування навантаження та кешування даних. Організацію роботи CDN представлено у вигляді функціональної багаторівневої схеми, яка включає в себе площину доступу до контенту, площину дистрибуції контенту, площину оцінки метрик і станів сервісів контролю якості послуг. Кожна із площин виконує ряд функцій і має свої особливості. Площина доступу забезпечує доступ до контенту кінцевим користувачам, а також організовує взаємодію між кешуючими серверами та сервером походження. Площина дистрибуції та балансування забезпечує вибір кешуючих серверів, які зможуть обслужити запити кінцевих користувачів із найкращою якістю. Площина метрик містить інформацію про стан усіх серверів, їх доступність та можливість надавати послуги. Ця площина взаємодіє із площиною дистрибуції та балансування. Завдяки такій взаємодії, сервери, які працюють не належним чином, не братимуть участі в обслуговуванні кінцевих користувачів.

Запропоновано метод оптимізованого кешування даних, який дає можливість ефективно використовувати кешовані дані і при цьому враховує якість, з якою надається послуга кінцевому користувачу. Найбільший пріоритет при застосуванні даного методу – якість послуг, яку отримує кінцевий користувач. Проводилось експериментальне дослідження роботи кешуючих серверів мережі доставки контенту. Проаналізовано ефективність використання запропонованого методу оптимізованого кешування даних та проведено порівняння із існуючими методами. Показано, що використання методу оптимізованого кешування дає змогу більш раціонально використовувати кешовані дані, а також зменшити навантаження на сервер походження та затримку при отриманні контенту.

Список використаних джерел:

1. The Zettabyte Era: Trends and Analysis. Cisco, 7 June 2017. [Електронний ресурс]. – Режим доступу: https://files.ifi.uzh.ch/hilty/t/Literature_by_RQs/RQ%20102/2015_Cisco_Zettabyte_Era.pdf.

2. *Chakraborty S., Sarddar D.* An Efficient Edge Server Selection in Content Delivery Network using Dijkstra's Shortest Path Routing Algorithm with Euclidean Distance // International Journal of Computer Applications. – 2015. – Vol. 117, No. 4. – P. 24-26.

3. *Парфенов В.И., Золотарев С.В.* Об одном алгоритме решения задачи оптимальной маршрутизации по критерию средней задержки // Вестник ВГУ: сер. Физика. Математика. – 2007. – № 2. – С. 28–32.

4. *Шварц М.* Сети связи: протоколы, моделирование и анализ: пер. с англ.: Ч. 1. – М.: Наука, 1992. – 336 с.

5. *Yaw-chung Chen*. Improving Quality of Experience in P2P IPTV // Network Operations and Management Symposium (APNOMS) 18th. – Asia-Pacific, 2016. – P. 6-9.

6. *Klymash M., Kyryk M., Pleskanka N., Yanyshyn V.* Data Buffering Multilevel Model at a Multiservice Traffic Service Node // Smart Computing Review. – 2014. – Vol. 4. No. 4. – P. 294-306.

7. *Bai Y., Jia B., Zhang J., Pu Q.* An Efficient Load Balancing Technology in CDN // Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on. – IEEE, 2009. – Т. 7. – P. 510-514.

8. *Дмитриев Г.А., Марголис Б.И., Музанна М.М.* Решение задачи оптимальной маршрутизации по критерию загруженности сети // Программные продукты и системы. – 2013. – № 4. – С. 17–19.

9. *Pallis George, Konstantinos Stamos, Athena Vakali «and other»*. Replication based on Objects Load under a Content Distribution Network // Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, 3-7 April 2006, Atlanta, GA, USA. – P. 1-9.

10. *Wauters T., Coppens J., Dhoedt B., Demeester P.* Load balancing through efficient distributed content placement // In proceeding of: Next Generation Internet Networks. – NY, 2005. – P. 99–105.

11. *Haghighi A., Mishev D.* Queueing models in industry and business. – New York: Nova Science Publishers, 2008. – 386 p.

12. *Dattatreya G.* Performance analysis of queuing and computer networks. – Boca Raton: CRC Press, 2008. – 449 p.

13. *Клейнрок Л.* Теория массового обслуживания. Пер. с англ. / Пер. И. И. Грушко; ред. В. И. Нейман. – М.: Машиностроение, 1979. – 512 с.