ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 3 (2), 2023

https://doi.org/10.23939/ictee2023.02.

ПОКРАЩЕННЯ ПАРАМЕТРІВ ЯКОСТІ ОБСЛУГОВУВАННЯ QOS В CDN МЕРЕЖІ ЗА РАХУНОК ВИКОРИСТАННЯ МОДУЛЯ EDGE COMPUTE

М. Плесканка [ORCID:0009-0003-0656-7796]

Національний університет "Львівська політехніка", вул. С. Бандери, 12, Львів, 79013, Україна

Відповідальний за рукопис: Мар'яна Плесканка (e-mail: mariana__p.m.v.9@ukr.net).

(Подано 1 Серпня 2023)

Розглянуто основні методи та принципи побудови мереж доставки контенту та особливості їхньої роботи. Запропоновано модуль Edge Compute для використання в мережах доставки контенту для покращення якості обслуговування, а саме збереження часу затримки обслуговування у задовільних межах у разі зростання навантаження. Здійснено імітаційне моделювання роботи модуля Edge Compute згідно із розробленим алгоритмом у мережі доставки контенту (CDN). На основі отриманих результатів моделювання подано графічні залежності ефективності використання Edge Compute модуля залежно від навантаженості та кількості запитів від клієнтів. Отримані залежності підтверджують ефективність використання модуля у разі зростання кількості запитів, а саме те, що зі збільшенням кількості запитів, а водночас і навантаження, час затримки залишається в задовільних межах.

Ключові слова: *обробка даних; IoT; проміжні обчислення; сервер походження (Origin); модуль Edge Compute; CDN мережі.*

1. Вступ

Швидкий розвиток мережевих технологій та IoT привів до ситуації, в якій сервери, що опрацьовують дані, здійснюють їх подальший аналіз та збереження, далеко не завжди можуть витримувати навантаження та якісно надавати сервіси кінцевим користувачам. В таких ситуаціях часто виникають хмарні (Cloud) рішення та CDN мережі, яких уже доволі багато. Оскільки пропозицій стосовно таких рішень чимало, відповідно конкуренція за послуги теж зростає. Головне завдання, яке ставлять перед сучасними провайдерами послуг та контенту, – можливість масштабування мережевої інфраструктури в умовах постійного зростання трафіку та розподіл навантаження по різних локаціях. Не менш важливими є доступність та забезпечення гарантованої якості обслуговування. Однак послугами одного тільки кешування даних вже нікого не здивуєш у наш час. Тому провайдерам таких послуг доводиться працювати над створенням нових сервісів, які могли б зацікавити потенційних клієнтів та робити їхні послуги привабливішими для споживачів.

2. Аналіз та постановка задачі

Доволі часто в сучасному IoT використовують технологію Edge Compute, щоб зменшити навантаження на централізований сервіс обробки даних, та швидку обробку даних, які надходять від різних джерел (IoT пристроїв). Edge Compute – архітектура, згідно із якою клієнтські дані

обробляються на межі доступу мережі, якомога ближче до вихідного джерела даних. Дані є джерелом інформації для сучасного бізнесу, вони підтримують контроль у реальному часі над критично важливими бізнес-процесами та операціями. Сучасний бізнес переповнений океаном даних, величезні обсяги яких можна регулярно збирати з давачів і пристроїв Інтернету речей, що працюють у режимі реального часу практично з будь-якої точки світу [1, 2].

Такий підхід зумовлює нові вимоги до обчислення та доставки таких даних з гарантованою якістю. У сучасному світі така кількість даних рухатиметься в обидві сторони одночасно і вимагає практично миттєвої обробки та мінімального часу на доставку результатів цієї обробки.

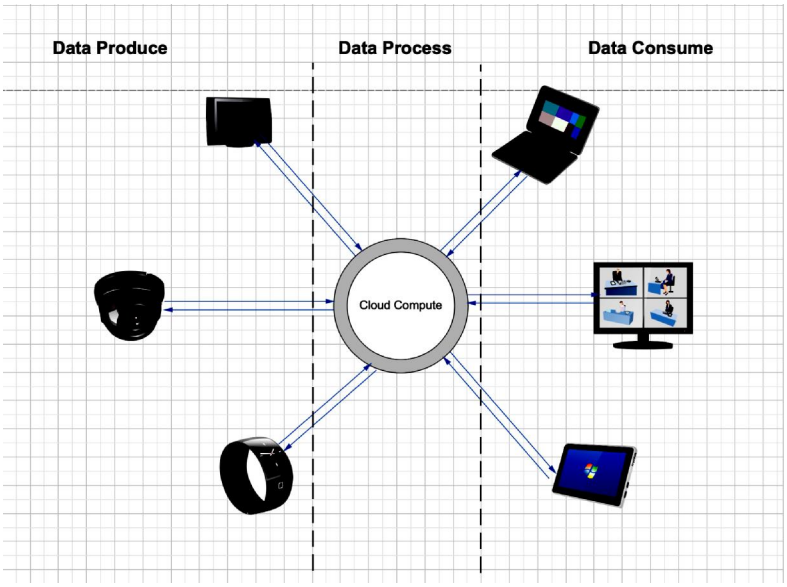

Рис. 1. Традиційна схема використання технології Edge Compute

Традиційна парадигма обробки та доставки даних, побудована на централізованому центрі обробки даних, не підходить для такої кількості даних, яка постійно зростає. Тут на заваді стають обмеження пропускної здатності, проблеми із затримкою та джитером, а також непередбачувані збої у роботі мережі. Компанії реагують на ці проблеми з даними, використовуючи архітектуру периферійних обчислень.

Простіше кажучи, основні процеси, які відповідають за аналіз та оброблення даних, розміщують ближче до джерела самих даних. Замість того, щоб передавати необроблені дані до центрального центру обробки даних для опрацювання та аналізу, цю роботу виконують там, де дані фактично генеруються. Це може бути роздрібний магазин, виробничий цех, розгалужене комунальне підприємство чи розумне місто. Лише результат цієї обчислювальної роботи, як-от прогнози технічного обслуговування обладнання чи інші відповіді, аналітика та детальний аналіз надсилають назад до головного центру обробки даних для подальшого опрацювання. Отже, технологія проміжних обчислень Edge Compute змінює IT та підходи, які використовують у бізнес-обчисленнях.

3. Архітектура та принципи роботи CDN

Варто зазначити, що застосування технології проміжних обчислень може бути ефективним також у поєднанні із використанням CDN, коли Edge сервер виконує не тільки функцію кешування, а й оброблення даних. Для кращого розуміння розглянемо випадки, в яких доцільно використовувати технології CDN. Схему роботи CDN мережі подано на рис. 2.

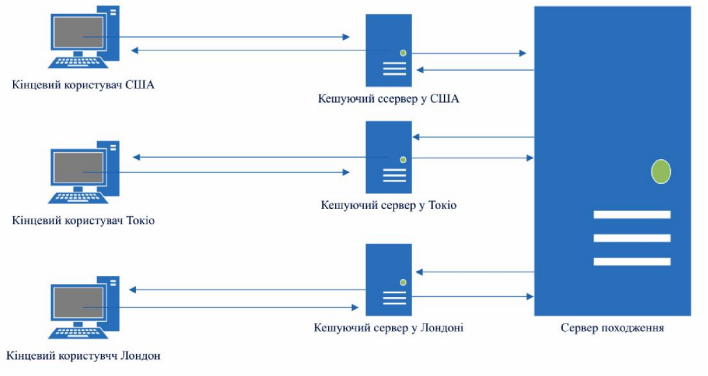

Рис. 2. Базова схема CDN мережі

Мережа доставки (розподілу контенту) CDN являє собою географічно рознесену мережу передавання інформації, яка складається із серверів опрацювання, кешування та трансляції контенту, а також мережевих маршрутів [3–5]. Основне завдання такої мережі – забезпечити якісну та надійну доставку інформації до кінцевого користувача.

Кінцеві клієнти запитують дані певного сервісу. Мережа CDN визначає локацію користувача за його IP адресою та направляє запит до найближчого кешувального сервера. Кешувальний сервер перевіряє, чи доступні дані в його кеші; якщо доступні – контент відразу надається користувачу. В іншому випадку запит надсилається до сусіднього кеш-сервера або ж до сервера першоджерела (Origin). Відповідно дані передаються користувачу та зберігаються у кеші для наступних клієнтів. Така схема доволі проста, але має певні недоліки з погляду безпеки. Основним недоліком є те, що доступ до сервера першоджерела дозволений із будь-якої локації. До того ж у такому випадку навантаження на сервер буде завжди більшим, оскільки будь-який кеш-сервер із будь-якої локації буде звертатись до сервера першоджерела, якщо немає даних у кеші [6, 7].

Для вирішення таких проблем у CDN мережах використовують технологію екранування CDN Shielding. Схему роботи із використанням цієї технології подано на рис. 3.

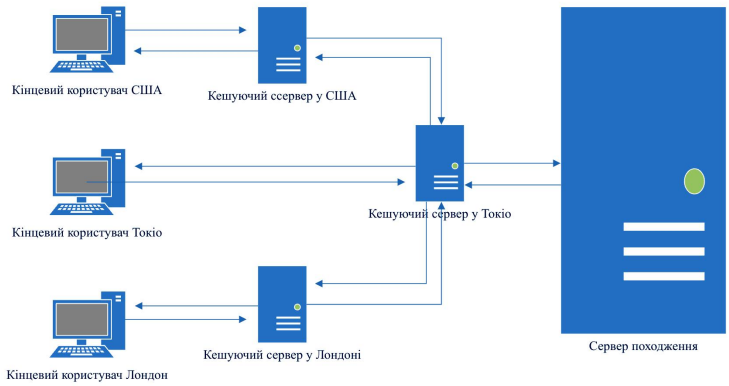

Рис. 3. Схема роботи CDN мережі із використанням екранування сервера походження

Особливість цієї схеми в тому, що доступ до сервера першоджерела дозволений тільки з однієї локації. Всі інші кеш-сервери із будь-яких інших локацій отримуватимуть контент від проміжного кешувального сервера, який матиме особливі правила кешування та звертатиметься до сервера першоджерела лише у випадку відсутності даних у власному кеші. На рис. 3 таку функцію

виконує кешувальний сервер у Токіо. Цим ми зменшуємо навантаження на сервер джерела, оскільки кеш-сервери із будь-якої локації будуть звертатись до проміжного сервера. Цей сервер здебільшого матиме заздалегідь кешований контент. Іншою перевагою є безпека, оскільки доступ до сервера джерела можливий тільки з однієї локації, в певних випадках IP адреси [8, 9].

4. Базові принципи роботи Edge Compute

У традиційних корпоративних мережах дані генеруються або ж створюються запити на стороні клієнта, наприклад, на комп'ютері користувача, мобільному пристрої чи будь-якому IoT давачі. Ці дані переміщуються через WAN, наприклад Інтернет, через корпоративну локальну мережу, де вони зберігаються та обробляються певними сервісами. Результати цієї обробки потім передається назад до кінцевого клієнта. Це перевірений і давно відомий підхід до клієнт-серверних обчислень, який використовують у більшості типових бізнес-додатків.

Але кількість пристроїв, підключених до інтернету, і обсяг даних, які створюються цими пристроями, зростає надто швидко, щоб традиційна інфраструктура центрів обробки даних могла всі ці дані вчасно обробити та зберегти. За прогнозами Gartner, до 2025 р. 75 % даних створюватимуться за межами централізованих центрів обробки даних. Перспектива переміщення такої кількості даних створює неймовірне навантаження на глобальний інтернет, який часто зазнає перевантажень і збоїв. Тож IT-архітектори перенесли фокус із централізованого центру обробки даних на логічно розподілену інфраструктуру – забрали ресурси зберігання та обчислювальні ресурси з центру обробки даних і перемістили ці ресурси до точки, де генеруються ці дані, або ж де генеруються запити даних. Принцип простий: якщо неможливо наблизити дані до центру обробки даних, наблизити центр обробки даних до даних [10, 11]. Така концепція не є новою, її початок вбачають у ідеях віддаленого обчислення, втілених у віддалених офісах та філіях, де надійніше та ефективніше розміщувати обчислювальні ресурси в потрібному місці, а не покладатися на єдине централізоване обслуговування та збереження.

5. Опис алгоритму адаптивного Edge Compute в CDN мережі

Базова схема роботи CDN мережі (рис. 2) відображає базові принципи роботи CDN мережі. Тобто основна функція, яку виконує CDN, – кешування даних від сервера походження та опрацювання запитів від кінцевих користувачів. Однак кешувальний сервер може виконувати не тільки функцію кешування даних, а й часткову обробку запитів користувачів, тобто Computing. В такому випадку пропонуємо використовувати Edge Compute в зворотному напрямку, коли самі дані містяться на сервері походження а кінцевий клієнт робить запит на отримання цих даних. Відповідно кешувальний сервер виконуватиме не тільки функцію кешування даних, а також їх обробку в міру можливості. Алгоритм роботи такої схеми наведено на рис. 4.

Наведемо описання алгоритму:

1. Клієнт надсилає запит на отримання певних даних, користуючись тим чи іншим сервісом. Найпростіший приклад – список товарів у інтернет-магазині.

2. Згідно із принципами роботи CDN запит потрапляє до найближчого сервера кешування, який розташований у локації клієнта або максимально близько до неї.

3. Наступний крок – перенаправлення запиту до сервера походження, на якому запущений певний тип сервісу.

4. Сервер походження (Origin) обробляє запит, який отримав від кешувального сервера, та формує відповідь. Обробка запиту та формування відповіді займе певний час.

5. Водночас модуль Edge Compute періодично опитує сервер походження і збирає метрики про його стан, завантаженість та аналізує час відповіді на запити клієнтів. Окремий компонент аналізує ці значення та приймає рішення, чи якість надання послуг кінцевому користувачу буде задовільною за певного навантаження на сервер походження та певного значення часу обробки запитів.

6. Якщо значення всіх параметрів задовільні, кешувальний сервер формує відповідь клієнту.

7. У іншому випадку, коли значення зростають, а саме час обробки запитів, завантаженість сервера походження, модуль Edge Compute приймає рішення про попереднє опрацювання запитів клієнтів прямо в себе, повторяючи цим частину бізнес-логіки роботи аплікації, яка працює на сервері походження. Лише частина даних, або попередньо оброблені запити, надсилаються на сервер походження.

8. Отже, модуль Edge Compute бере на себе частину функціоналу та бізнес-логіки роботи сервісу, до якого звертається клієнт, і тим самим зменшує навантаження на сервер походження та зменшує час відповіді для клієнта. Також модуль може збирати статистику, наперед прогнозувати час можливого зниження QoS та заздалегідь запускати опрацювання запитів на своїй стороні. За певний проміжок часу такий модуль адаптується до особливостей роботи певного сервісу та зможе заздалегідь делегувати опрацювання даних на кешувальні сервери, розташовані максимально близько до клієнта.

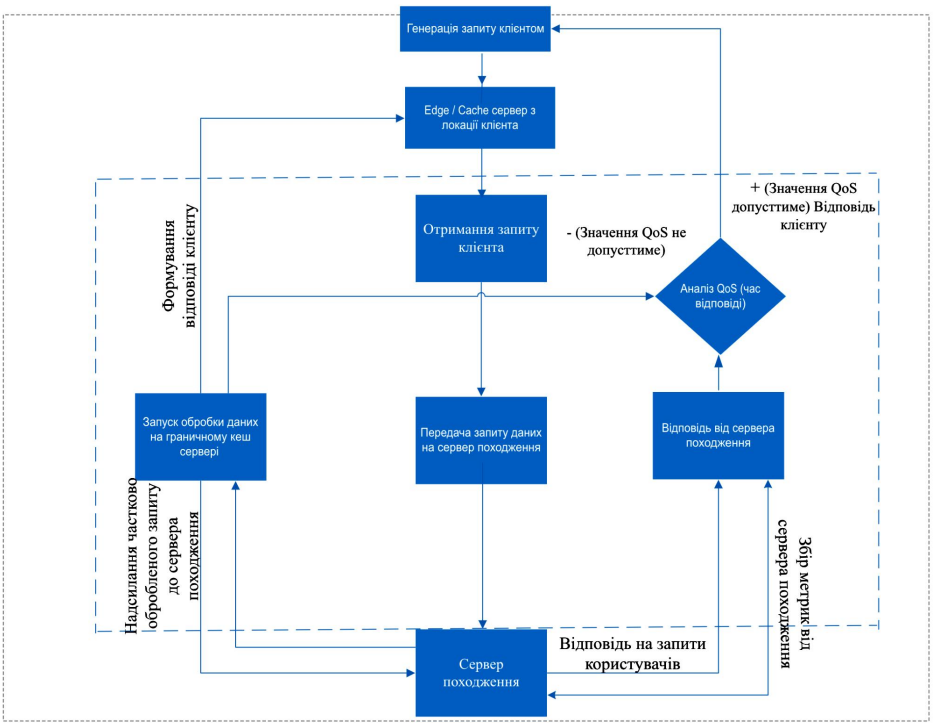

Рис. 4. Алгоритм роботи адаптивного Edge Compute в CDN мережі

6. Аналіз результатів моделювання адаптивного Edge Compute

У межах цієї роботи виконано моделювання роботи модуля, який працює згідно із описаним вище алгоритмом. Як зазначено вище, розроблений алгоритм, дає змогу переносити частину бізнес-логіки роботи певного сервісу на кешувальні сервери, адаптуватись до особливостей роботи сервісу, а також передбачати та бути готовим до зростання навантаження в певні моменти часу зі збереженням задовільних параметрів якості обслуговування.

Особливості роботи моделі були такими. В розробленій моделі сервіс являє собою WEB аплікацію, яка опрацьовує каталог товарів інтернет-магазину. База даних, яка використовувалась, – MSSQL, каталог товарів містив 500 000 одиниць. Навантаження генерувалось із використанням додатка Apache Jmeter. За допомогою Jmeter було сконфігуровано профайл моделювання, який працював 12 годин. Кількість запитів задавали в межах 1000–5500 запитів на хвилину. Навантаження

змінювалось у три фази. Перша фаза тривала 4 години, починаючи із 2 p.m до 6 p.m. В цей проміжок часу кількість запитів була в межах 1500–2000 запитів на хвилину. Починаючи з 6 p.m кількість запитів зростала на 1000 за годину. Пік навантаження був о 9 p.m і становив 5500 запитів за хвилину. Параметром якості обслуговування було вибрано час відповіді web-сервісу. Встановлено задовільне значення 200 мс, а допустиме відхилення в межах 10 %. Значення 200 мс вибрано на основі статистичних даних, взятих за один місяць з web-сервісу, який використовується на виробництві.

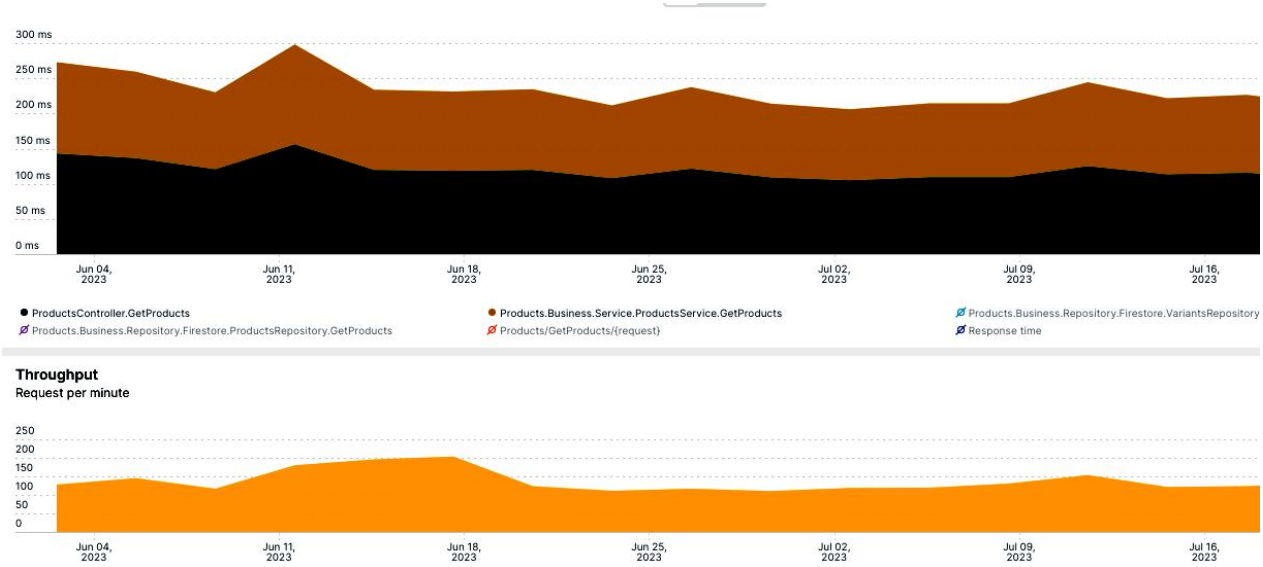

Рис. 5. Статистичні дані часу затримки під час оброблення запитів продукт-каталогу інтернет-магазину

Якщо значення часу відповіді зростало на 10 % від встановленого порога і залишалось на цьому рівні упродовж 30 хв, активувався модуль Edge Compute. Варто зазначити, що доволі часто виникають сплески навантаження тривалістю кілька хвилин. В таких випадках активувати модуль Edge Compute недоцільно, оскільки варто враховувати час на створення ресурсу, запуск сервісу та кешування частини даних від сервера походження. Клієнт у вигляді Jmeter імітував звернення користувачів до інтернет-магазину, щоб отримати інформацію про певний товар, а саме описання товару, наявність на складі, категорію товару, зображення, відгуки про цей товар, а також його додаткові атрибути. Модуль Edge Compute у цій моделі – окремий сервіс, який виконував частину бізнес-логіки основного сервісу, зменшуючи навантаження на основний сервіс та тим самим і час відповіді клієнту.

Результати подано на рис. 6–11.

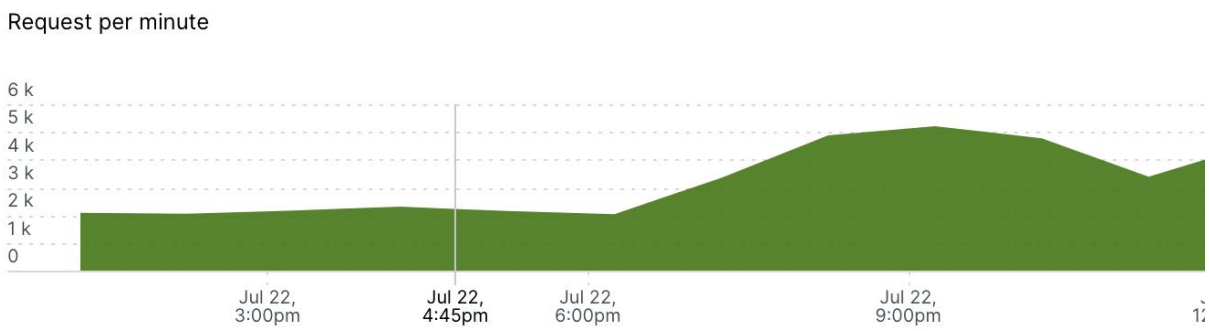

Рис. 6. Кількість запитів за проміжок часу моделювання

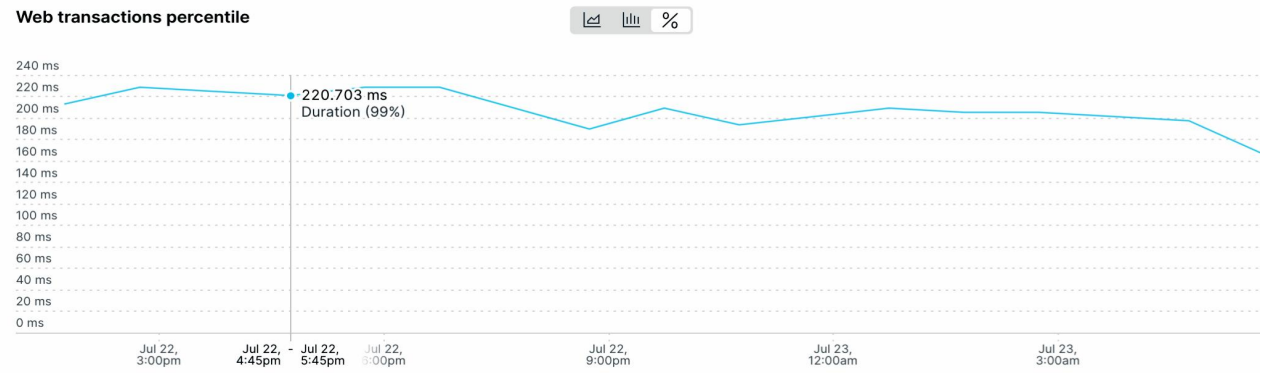

Рис. 7. Зміна часу затримки залежно від навантаження

На рис. 6, 7 показано, як змінювалось навантаження (кількість запитів), та як довго за такого навантаження тривали запит та відповідь клієнту. Як бачимо, за середньої кількості запитів 1500–1600 за хвилину час затримки становив 220 мс. На цьому етапі моделювання не було задіяно модуль Edge Compute. Результат відповіді на запити генерував або кешувальний сервер (якщо дані були закешовані), або ж сервер походження.

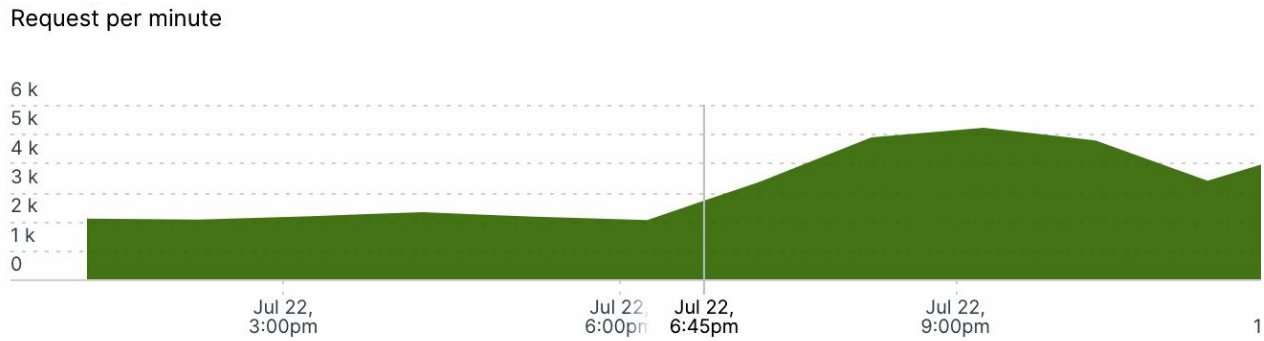

Рис. 8. Кількість запитів за проміжок часу моделювання у момент зростання навантаження

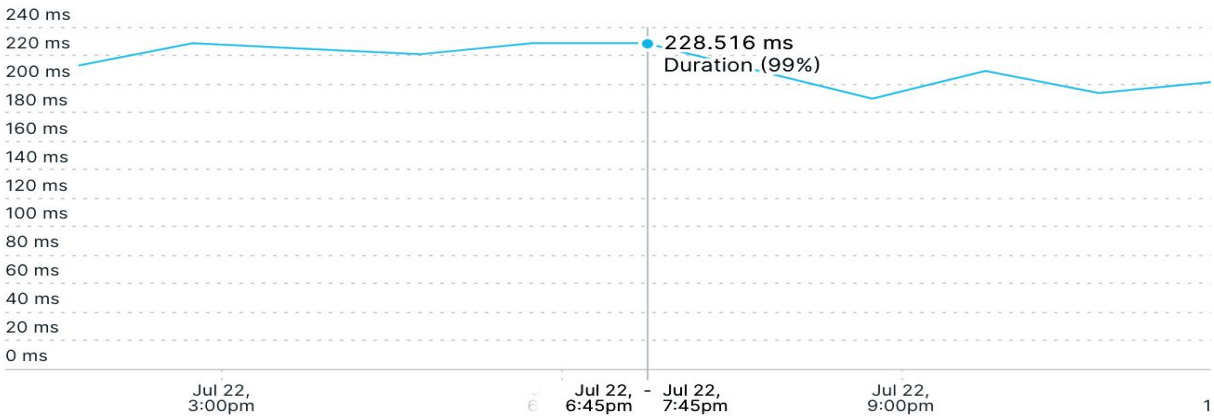

Рис. 9. Зміна часу затримки залежно від навантаження у момент зростання кількості запитів

На рис. 8 зафіксовано момент зростання навантаження. Кількість запитів почала зростати від 1500 до 4500 за хвилину. Водночас на рис. 8 бачимо, що час затримки починає зростати і становить 228 мс. Відповідно до алгоритму, модуль Edge Compute все ще збирає метрики та аналізує параметри якості обслуговування. У налаштуваннях модуля встановлено порогове значення часу затримки 230 мс за кількості запитів більше ніж 2500 за хвилину.

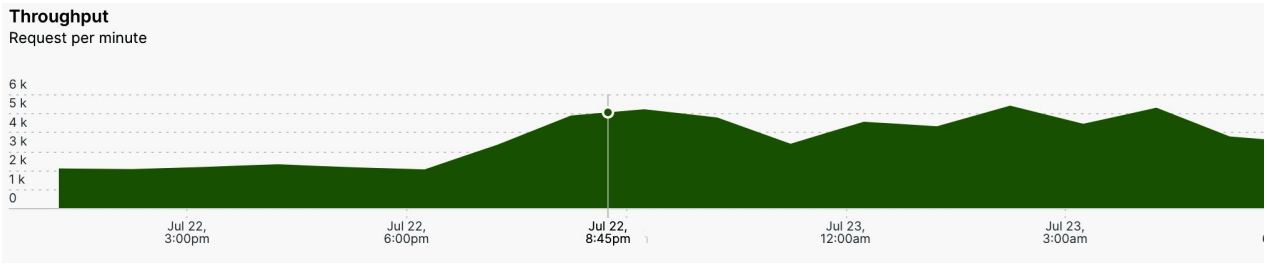

Рис. 10. Кількість запитів за проміжок часу моделювання у момент, коли працює модуль Edge Compute

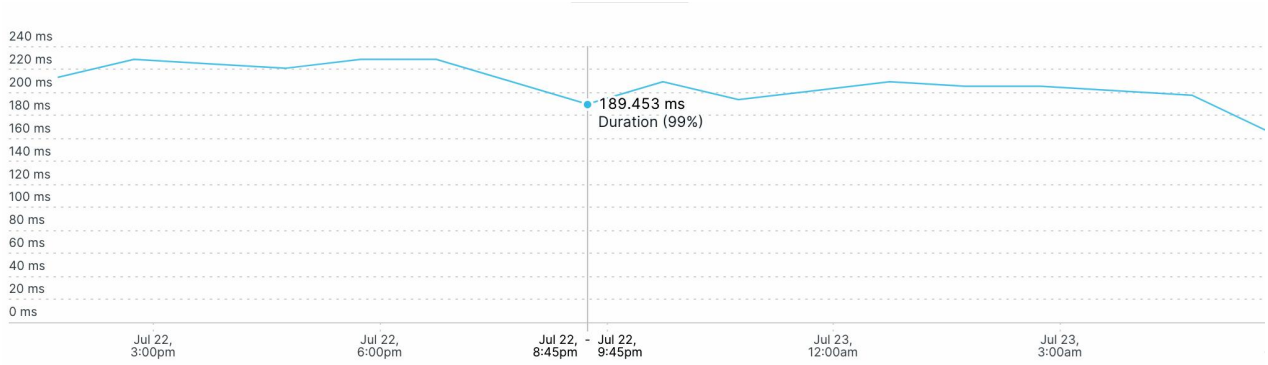

Рис. 11. Зміна часу затримки залежно від навантаження в момент, коли працює модуль Edge Compute

На рис. 10, 11 наведено значення кількості запитів та відповідно часу затримки у момент, коли вже працює модуль Edge Compute. Як зазначено вище, модель Edge Compute постійно збирає метрики від сервера походження, аналізує значення параметрів QoS. На основі зібраних даних, їх аналізу та наперед вказаних допустимих значень модуль приймає рішення про те, коли розпочинати обробку даних на своїй стороні. На рис. 9 чітко показано, що модуль активувався у момент, коли значення часу затримки досягло порогового рівня за заданої кількості запитів. Час затримки після активації почав зменшуватись і, як бачимо, коли навантаження досягало 5500 запитів за хвилину, становив 189,45 мс. Як свідчать результати моделювання, ефективність застосування цього модуля очевидна, оскільки значення часу затримки зменшилось навіть за умови, що кількість запитів зросла вдвічі. Варто зазначити також, що всі моменти активації/деактивації модуля зберігаються з метою подальшого аналізу та можливості наперед прогнозувати періоди зростання навантаження і попередньої активації зі збереженням параметрів якості обслуговування в допустимих межах. Це також впливає на якість сприйняття роботи сервісу кінцевими споживачами.

Висновки

У роботі розглянуто концепцію мережі доставки контенту з використанням технології Edge Compute. Описано основні принцип роботи та схеми організації, використовувані у мережах доставки контенту. Наведено дві схеми: одна – базова/класична схема роботи мережі доставки контенту, інша з використанням екранування сервера походження. Схема із екрануванням має дві переваги порівняно із класичною. Першою перевагою є істотне зменшення навантаження на сервер походження, другою – захищеність сервера походження.

Розглянуто основні методи роботи та використання технології Edge Compute, яку застосовують переважно для швидкого збирання та оброблення даних від джерел їх генерації. Розроблено алгоритм роботи адаптивного Edge Compute модуля, який призначений для забезпечення задовільних параметрів якості обслуговування в мережах передавання даних. Цей модуль виконує збирання

метрик й аналізування параметрів якості обслуговування та на основі певних, наперед заданих значень, приймає рішення про те, коли починати обробку даних на своїй стороні. Використання такої системи дає змогу завжди мати актуальну інформацію про стан сервера походження, актуальні значення параметрів якості обслуговування та гранично допустимі межі, за яких ефективно застосовувати цей модуль. Також на основі зібраних статистичних даних можна заздалегідь прогнозувати періоди зростання навантаження та використовувати запропонований модуль для збереження параметрів якості обслуговування у допустимих межах.

Виконано імітаційне моделювання роботи модуля Edge Compute у CDN мережі із урахуванням гранично допустимих значень часу затримки за заданої кількості запитів від клієнтів. Подано графічне відображення результатів моделювання. Залежності показують, як змінюється час затримки у разі зміни навантаження. На основі результатів моделювання можна стверджувати, що запропонований модуль є доволі ефективним, оскільки зі збільшенням кількості запитів у два рази дає змогу забезпечити задані значення якості обслуговування QoS. Проаналізувавши результати експериментальних досліджень, можна стверджувати, що використання модуля Edge Compute дає можливість істотно зменшити завантаженість сервера походження, знизити затримку кінцевого користувача під час отримання контенту, а також ймовірність втрати даних на шляху передавання. Всі ці переваги дають змогу покращувати якість сервісу в мережах передавання даних.

Список використаних джерел

[1] *A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things", in IEEE Transactions on Emerging Topics in Computing, Vol. 5, No. 4, pp. 586–602, 1 Oct.-Dec. 2017. DOI: 10.1109/TETC. 2016.2606384.*

[2] *B. K. Barman, S. N. Yadav, S. Kumar and S. Gope, "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid", 2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE), Shillong, India, 2018, pp. 1–5. DOI: 10.1109/EPETSG.2018.8658501.*

[3] *M. Klymash , M. Kyryk, N. Pleskanka., V. Yanyshyn. Data Buffering Multilevel Model at a Multiservice Traffic Service Node, Smart Computing Review, 2014, Vol. 4, No. 4, pp. 294–306. DOI:10.6029/smartcr.2014.04.006*

[4] *G. A. Dmitriev, B. I. Margolys, M. M. Muzanna. The solution of the problem of optimal routing according to the criterion of network congestion, Programmnye produkty i systemi, 2013, No. 4, pp. 17–19.*

[5] *Y. Bai, B. Jia, J. Zhang and Q. Pu, "An Efficient Load Balancing Technology in CDN", 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 2009, pp. 510–514. DOI: 10.1109/FSKD.2009.130..*

[6] *G. Pallis, K. Stamos, A. Vakali, D. Katsaros and A. Sidiropoulos, "Replication Based on Objects Load under a Content Distribution Network", 22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA, 2006, pp. 53–53. DOI: 10.1109/ICDEW.2006.127*

[7] *M. Kyryk, N. Pleskanka and M. Pitsyk, "QoS mechanism in content delivery network", 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 2016, pp. 658–660. DOI: 10.1109/TCSET.2016.7452144.*

[8] *M. Kyryk, N. Pleskanka, M. Pleskanka . Content Delivery Network Usage Monitoring // Proceedings of 14th International IEEE conference "The experience of designing and application of CAD systems in microelectronics", CADSM'2017, 21–25 February 2017, Poljana-Svalyava(Zakarpattya), Ukraine. Lviv: Lviv Polytechnic Publishing House, pp. 306–308.*

[9] *A. Tekin, A. Tuncer Durak, C. Piechurski, D. Kaliszan, F. Aylin Sungur, F. Robertsen, and P. Gschwandtner. State-of-the-art and trends for computing and network solutions for hpc and ai, prace technical report. PRACE Technical Report, Dec. 2020, 2020.*

[10] *A. Johansson, C. Piechurski, D. Pleiter, and K. Wadwka. Data management services and storage infrastructures. PRACE Technical Report, Dec. 2020, 2020.*

THE QUALITY OF SERVICE PARAMETERS QOS IMPROVEMENT IN CDN NETWORK WITH EDGE COMPUTE MODULE

Mariana Pleskanka

Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013, Ukraine

The main methods and principles of building content delivery networks and the peculiarities of their work are considered. An Edge Compute module proposed for use in content delivery networks to improve quality of service. A simulation modeling of the Edge Compute module, which works based on developed algorithm in the content delivery network (CDN) was performed. Based on the obtained simulation results, graphical dependencies of the efficiency of using Edge Compute module depending on the load and the number of requests from clients are presented. The results confirm the effectiveness of using the module when the number of requests are increasing. At the same time when the number of requests increases the delay time remains within satisfactory limits.

Key words: *data processing; IoT; intermediate calculations; Origin server (Origin); the Edge Compute module; CDN network.*