

Сидоренко Р. О.,
здобувач вищої освіти спеціальності F3 Комп'ютерні науки
Науковий керівник: Богатєнкова О. Є., асистент кафедри економічної
кібернетики, комп'ютерних наук та інформаційних технологій
Миколаївський національний аграрний університет
м. Миколаїв

ВПЛИВ АРХІТЕКТУР МОНИТОРИНГУ МЕРЕЖЕВОГО ТРАФІКУ НА ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ

Сучасні інформаційні системи, зокрема ті, що вимагають моніторингу мережевого трафіку або швидких оновлень даних (як-от криптобіржі, онлайн-ігри чи спільне редагування документів) [5], функціонують у режимі Soft Real-Time, що вимагає від користувачьких інтерфейсів (UI) здатності відображати зміни стану системи з мінімальною затримкою. Традиційний підхід до досягнення цієї мети – періодичні HTTP-запити, відомі як Polling. Хоча Polling простий у реалізації, він створює значне та непотрібне навантаження на клієнтське обладнання та мережу. Це навантаження виникає через циклічне встановлення з'єднань та обробку надлишкової метаінформації (наприклад, HTTP-заголовків) навіть у періоди відсутності нових даних. На противагу Polling, технологія WebSockets [2] дозволяє встановити постійний двонаправлений зв'язок між клієнтом і сервером. Цей підхід, відомий як push-механізм [1], який забезпечує передачу даних лише тоді, коли вони справді оновлюються, що кардинально змінює парадигму взаємодії та ресурсного споживання. Метою даного дослідження було кількісне порівняння економічної ефективності архітектур Polling [1] та Subscriptions, для чого було застосовано інтегровані наукові метрики, такі як Індекс цифрового метаболізму (DMI) [1], Індекс ефективності рендерингу (RES), Коефіцієнт візуальної ентропії (VEC) та Індекс поведінкової точності (IBA). Економічне обґрунтування цієї тези полягає у доведенні, що вибір архітектури Subscriptions мінімізує операційні витрати, знижує енергоспоживання (що є критичним для мобільних пристроїв та корпоративних ЦОД) та підвищує якість обслуговування.

Polling функціонує за принципом «тягни, щосекунди»: клієнт активно запитує дані, незалежно від того, чи відбулися зміни на сервері. Ця постійна, невинуватна активність призводить до значного навантаження на сервер та клієнтське обладнання, що має прямі економічні наслідки. Експериментальні дані показали, що Polling витрачає у 35 разів більше CPU (26,19% проти 0,75% для Subscriptions) і споживає у 2,16 рази більше оперативної пам'яті (88,38 МВ проти 40,91 МВ). У масштабах великих корпоративних систем, таке марнотратство ресурсів означає зростання витрат на електроенергію, необхідність у більш потужному та дорогому клієнтському обладнанні та вищі операційні витрати на підтримку серверної інфраструктури. Щодо мережевого трафіку, Polling генерує 32 запити на секунду (RPS) навіть за відсутності корисних даних, створюючи постійний, непотрібний потік метаінформації, збільшуючи загальну пропускну здатність (Bandwidth) і, відповідно, витрати на

мережеві послуги. Subscriptions, навпаки, має нульовий RPS, коли немає подій.

Нова метрика Індекс цифрового метаболізму (DMI) [1], яка вимірює вартість отримання одиниці інформації в контексті витрачених ресурсів, критично висвітлює неефективність Polling: DMI для Polling демонструє хаотичні сплески, що виникають, коли CPU та пам'ять високі, а корисний потік даних (Throughput) нульовий. Це доводить, що Polling постійно «спалює» ресурси без корисного результату.

Максимізація корисного потоку даних. Subscriptions, використовуючи протокол WebSocket, реалізує «подієвий» механізм, який активує клієнт лише тоді, коли сервер надсилає нову інформацію, забезпечуючи максимальну енергоефективність. Індекс ефективності рендерингу (RES) [1], аналог ККД, демонструє, скільки корисного результату система генерує на одиницю витраченої енергії. RES для Polling практично нульова, підтверджуючи його вкрай низький ККД. Натомість, Subscriptions демонструє періодичні, але високі та стійкі піки (до 120), які виникають у моменти передачі даних. Це прямий доказ того, що Subscriptions використовує клієнтські ресурси з максимальною ефективністю у моменти, коли це необхідно. Крім того, Subscriptions забезпечує у 7 разів більший потік корисних даних (Throughput) (0,0486 MB/s проти 0,0073 MB/s) під час активного обміну, передаючи значно більше актуальної інформації за однакову кількість витрачених клієнтських ресурсів.

Вплив на користувацький досвід (UX) та бізнес-показники. Якість UX є невід'ємним економічним показником, оскільки бездоганний UX конвертується у вищу залученість клієнтів та сильнішу лояльність. Коефіцієнт візуальної ентропії (VEC) [1], який об'єднує час затримки відповіді (Latency) та час до повної інтерактивності (TTI) [1], відображаючи "площу порожності" в очікуванні контенту. Для Polling лінія VEC перебуває у високому діапазоні (250–500 ms²), що свідчить про постійну високу невизначеність та "фрустрацію". Subscriptions підтримує VEC на надзвичайно низькому та стабільному рівні (15–25 ms²), забезпечуючи набагато вищу UX-стабільність. Крім того, Subscriptions демонструє значно вищу швидкість реакції: мережева затримка (Latency) скорочується у 8 разів (з 7,98 ms до 1,00 ms), а повна інтерактивність (TTI) досягається у 2,2 рази швидше (з 43,54 ms до 19,56 ms). Така швидкість є критичною в сферах, де миттєвість оновлення безпосередньо впливає на прибуток (наприклад, біржова торгівля).

Результати дослідження категорично підтверджують гіпотезу про те, що архітектура на базі WebSockets/Subscriptions є єдиним економічно обґрунтованим рішенням для сучасних Soft Real-Time систем порівняно з класичним Polling. Polling є фундаментально марнотратною архітектурою: його DMI є катастрофічно високим, а RES – практично нульовим, оскільки він генерує постійний потік непотрібних запитів (RPS 32 req/s), створюючи значне, невиправдане навантаження на CPU (у 35 разів більше) та пам'ять клієнтських пристроїв. Це призводить до вищих операційних витрат, зниження ресурсу батарей та більших вимог до серверної інфраструктури. На противагу цьому, Subscriptions використовують ресурси цілеспрямовано та ефективно: вони забезпечують мінімальне навантаження на CPU та пам'ять, досягають TTI у 2,2

рази швидше та скорочують Latency у 8 разів. Ця архітектура максимізує енергоефективність (високий RES, низький DMI) та забезпечує бездоганний користувацький досвід (стабільно низький VEC), що прямо впливає на підвищення конкурентоспроможності продукту та економічну вигоду в довгостроковій перспективі.

Список використаних джерел

1. Інструкція по Apollo Client | MyJavaScript. MyJavaScript | MyJavaScript. URL: <https://my-js.org/docs/guide/apollo/client> (дата звернення: 08.12.2025).

2. Інструкція по Apollo Server | MyJavaScript. MyJavaScript | MyJavaScript. URL: <https://my-js.org/docs/guide/apollo/server> (дата звернення: 08.12.2025).

3. Тривале опитування. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/long-polling> (дата звернення: 08.12.2025).

4. Що таке метрики? Базові метрики, які повинен знати кожен QA. QualityAssuranceGroup. URL: <https://qagroup.com.ua/publications/what-is-metrics/> (дата звернення: 08.12.2025).

5. GraphQL що це за мова і як маніпулює даними для API. FoxmindEd. URL: <https://foxminded.ua/graphql-shcho-tse/> (дата звернення: 08.12.2025).