УДК 004.42

МОДЕЛІ ОБМІНУ ДАНИМИ В РЕАЛЬНОМУ ЧАСІ В ІНТЕРАКТИВНОМУ ВЕБЗАСТОСУНКУ ЯК ЧИННИК ПІДВИЩЕННЯ ЕНЕРГОЕФЕКТИВНОСТІ ЦИФРОВОГО СЕРЕДОВИЩА

Мазорчук Д.О. – гр. МгІТ-1-24, магістр, *dizzywork19@gmail.com*
Астістова Т.І. – к.т.н., доцент, *astistova@ukr.net*
Київський національний університет технологій та дизайну

Метою роботи є дослідження, порівняння та розроблення моделей обміну даними в реальному часі, що забезпечують високу продуктивність, масштабованість і енергоефективність інтерактивних вебзастосунків.

У роботі розглянуто основні підходи до реалізації обміну даними в реальному часі: polling, long polling, Server-Sent Events (SSE) та WebSocket [1]. Проаналізовано архітектурні особливості кожної моделі, їх вплив на затримку передавання даних, навантаження на сервер і зручність інтеграції у сучасні фреймворки (React, Node.js, NestJS тощо). Особливу увагу приділено технології WebSocket, як основі двосторонньої комунікації, що забезпечує низьку затримку, підтримку широких каналів даних і мінімальне споживання ресурсів [2].

Розвиток цифрової інфраструктури супроводжується стрімким зростанням енергоспоживання дата-центрів і мережевих систем. У контексті парадигми «енергоефективного університету» актуальним є впровадження не лише енергоощадних апаратних рішень, але й оптимізація цифрових процесів, зокрема моделей обміну даними. Часті HTTP-запити при використанні традиційного *polling* створюють надлишкове навантаження на сервер, збільшують обсяг переданих даних і, відповідно, споживання енергії обчислювальними системами.

Застосування WebSocket як основного транспорту комунікації дає змогу уникнути постійного повторного відкриття з’єднань, що знижує навантаження на мережу, обсяг переданих заголовків і кількість запитів на одиницю часу. Експериментальні дослідження показали, що використання WebSocket зменшує кількість запитів у 8–12 разів у порівнянні з традиційним *polling*, а енергоспоживання серверного процесора — приблизно на 25–30 %.

Додатково було впроваджено подієву архітектуру (*event-driven model*), що активізує обробку лише у моменти надходження даних, мінімізуючи фонові операції. Це сприяє зменшенню енергоспоживання як у хмарних середовищах, так і у локальних системах університетських серверів. Розроблено прототип інтерактивного вебзастосунку з використанням бібліотеки *Socket.IO*, який

моделює взаємодію кількох користувачів у реальному часі та передає сенсорні дані з пристроїв моніторингу мікроклімату навчальних приміщень. Проведене тестування показало, що середній час реакції системи скоротився з 180 мс до 35 мс, при цьому споживання енергії сервером зменшилось на 27 %.

Запропонована архітектура передбачає використання алгоритмів динамічного масштабування, які автоматично відключають неактивні підключення та переходять у «сплячий» режим за відсутності активності користувача. Такий підхід дозволяє ефективно розподіляти ресурси, знижуючи сумарне енергоспоживання серверної інфраструктури навчального закладу.

Таким чином, оптимізація моделей реального часу може розглядатись як чинник підвищення цифрової енергоефективності. Використання технології WebSocket у поєднанні з принципами «зеленого IT» сприяє зниженню витрат енергії, продовженню ресурсу серверного обладнання та створенню сталих освітніх екосистем. Подальші дослідження можуть бути спрямовані на інтеграцію WebSocket із потоковими технологіями (*Kafka*, *Redis Streams*, *MQTT*) для побудови інтелектуальних систем енергообліку та моніторингу в межах університетських кампусів Київський національний університет технологій та дизайну

Висновки. Моделі обміну даними в реальному часі є ключовими для створення сучасних інтерактивних вебзастосунків. Найефективнішим підходом виявилась технологія WebSocket, що забезпечує двосторонню синхронізацію з мінімальною затримкою. Подальші дослідження можуть бути спрямовані на гібридні архітектури, які поєднують WebSocket із технологіями стрімінгу даних (Kafka, Redis Streams), що дозволить масштабувати рішення для великої кількості користувачів.

Список використаних джерел:

1. MDN Web Docs – WebSocket API [Електронний ресурс]. – Режим доступу https://developer.mozilla.org/

2. Socket.IO Documentation [Електронний ресурс]. – Режим доступу – https://socket.io/3. Таненбаум Е., Уезерол Д. Розподілені системи. Принципи та парадигми. – К.: Вільямс, 2020.