

Петров Сергій Петрович, к.т.н., доц., доцент кафедри Комп'ютерних наук ДУІКТ  
Марко Петро Сергійович, аспірант 2 курсу кафедри Комп'ютерних наук ДУІКТ  
Величко Денис Васильович, студент 5 курсу кафедри Комп'ютерних наук ДУІКТ

## ВПЛИВ СЕРВЕРНОГО РЕНДЕРУ НА ПРОДУКТИВНІСТЬ ВЕБ-СТОРІНОК

*Постановка задачі.* Більшість веб-сайтів на сьогоднішній день будуються за допомогою виконання скриптів JavaScript. Виконання JavaScript коду при старті сторінки негативно впливає на продуктивність сторінки, таким чином що після завантаження сторінки, основний потік JavaScript є заблокованим під час виконання вкладених скриптів. Заблокування основного потоку JavaScript призводить до того, що сторінка не відповідає на дії користувача. Параметр який вираховує час до того, коли сторінка стає інтерактивною після першого свого відображення називається Total Blocking Time, та відповідає за 30% загальної оцінки веб-сайту в категорії продуктивність. Тому необхідно вирішити наукове завдання з пошуку способу зниження часу блокування та поліпшення загальної продуктивності сторінки.

*Мета дослідження.* Підвищити ефективність відображення веб-сайту за допомогою рендеру веб-сайту на боці серверу. Дослідити вплив використання технології на загальну продуктивність відображення веб-сайту, зокрема параметр Total Blocking Time. Основна ідея даного підходу полягає у виконанні JavaScript на боці серверу, позбавляючи браузер клієнта від необхідності його виконання під час відображення сторінки. Крім того, результатом рендеру на боці серверу є статична HTML сторінка, що забезпечує можливість індексації пошуковою машиною для веб-сайтів які раніше не мали цієї можливості.

*Результати дослідження.* Під час завантаження веб-сторінки, яка наповнюється динамічно за рахунок виконання скриптів JavaScript, сторінка є заблокованою для взаємодії до тих пір, поки основний потік виконання JavaScript на закінчить свою роботу. Дана умова викликана тим, що JavaScript в браузері має один основний потік який відповідає за виконання скриптів, малювання інтерфейсу, та обробку подій. Таким чином виконання JavaScript при старті сторінки для наповнення веб-сайту вмістом блокує взаємодію користувача з сайтом до звільнення основного потоку. Метрика, яка заміряє час від того як веб-сайт завантажений до того, як основний потік стане вільним і сайт доступним для взаємодії називається Total Blocking Time (ТВТ). Метрика ТВТ впливає на загальну оцінку продуктивності веб-сайту, таким чином що оцінка ТВТ впливає на 30% загальної оцінки продуктивності веб-сайту при вимірах за допомогою системи Google Lighthouse. При виконанні JavaScript веб-сайту завчасно на боці серверу, досягається результат, при якому після завантаження веб-сайт вже є наповнений вмістом, таким чином що браузеру не потрібно виконувати більшість скриптів. За рахунок звільнення від виконання скрипту для наповнення веб-сайту, досягається зниження Total Blocking Time, що має позитивний ефект на оцінку продуктивності веб-сайту пошуковими машинами та сприйманні веб-сайту користувачем. Крім того, оскільки сайт тепер є сгенерованим завчасно, він

стає доступним для індексації пошуковою машиною, що також є значним покращенням SEO. Дослідження показало, що для веб-сайту з наповненням в 60 елементів, за допомогою рендеру на боці серверу параметр TBT було знижено з 200 мілісекунд до 20. Таким чином, технологія рендеру на боці серверу є перспективним рішенням для покращення продуктивності відображення веб-сайту.

*Висновки.* Використання рендеру на боці серверу дає можливість значно підвищити продуктивність відображення сторінки. Також, дана технологія дає можливість індексації веб-сайту, вміст якого створюється динамічно за допомогою JavaScript. Перспективами розвитку є кешування завчасно оброблених сторінок та покращення швидкості обробки.

#### Список використаних джерел

1. First Contentful Paint Lighthouse | Chrome for Developers URL: <https://developer.chrome.com/docs/lighthouse/performance/first-contentful-paint>.
2. Total Blocking Time (TBT) URL: <https://web.dev/articles/tbt>.
3. Strazzullo, F. (2023). Web Components. In: Frameworkless Front-End Development. Apress, Berkeley, CA. [https://doi.org/10.1007/978-1-4842-9351-5\\_5](https://doi.org/10.1007/978-1-4842-9351-5_5).