

УДК 004.43

**Д. Сомін, А. Паламар, В. Волоський**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## **ПЕРЕВАГИ WEBASSEMBLY ЯК ІНСТРУМЕНТА РЕАЛІЗАЦІЇ АЛГОРИТМІВ У РЕСУРСОМІСТКИХ ВЕБ-ДОДАТКАХ**

UDC 004.43

**D. Somin, A. Palamar, V. Voloskyi**

## **WEBASSEMBLY ADVANTAGES AS A TOOL FOR ALGORITHMS IMPLEMENTATION IN HIGH-LOAD WEB APPLICATIONS**

Сьогодні великої популярності набуло використання веб застосунків для різноманітних обчислень. Разом з тим, процес їх розробки супроводжується низкою проблем: складність масштабування, швидкість виконання, вартість хостингу, тощо. Значним недоліком більшості веб платформ є абсолютне домінування мови JavaScript (JS) та її похідних, як інструменту розробки додатків.

Попри наявність сильних сторін, JavaScript має значні недоліки: низька ефективність та великий об'єм коду, відсутність типу даних «ціле число», повільна швидкість виконання, неповна підтримка стандартів та різна поведінка браузерів під час інтерпретації. Тому, актуальною задачею є оптимізація складних алгоритмів за допомогою інших мов програмування.

Одним із методів оптимізації JavaScript алгоритмів є технологія WebAssembly (Wasm), яка містить опис незалежного від браузера, універсального, низькорівневого проміжного коду для запуску застосунків, скомпільованих на різних мовах програмування, що має на меті компенсувати недоліки JavaScript.

Перевагами WebAssembly є можливість написання та компіляції певного коду на низькорівневих мовах програмування, таких як C++, Go, Rust, та використання його у веб додатках. Це дозволяє підвищити швидкодію виконання окремих модулів веб додатків. Також, значною перевагою WebAssembly є можливість написання багатьох потокових програм.

В середньому Wasm є швидшим ніж JavaScript, але в кожному окремому випадку потрібно робити порівняння цих двох технологій, оскільки використання Wasm може дати як і набагато кращий результат так і навпаки. В значній мірі це залежить від браузера, який використовується для запуску веб додатка.

Як правило, Wasm добре показує себе там, де велика кількість операцій з пам'яттю, наприклад: шифрування, хешування, робота з масивами даних та інше. В серпні 2019 року була опублікована стаття [1], у якій автори реалізували можливість запуску під WebAssembly консольних утиліт Linux, для запуску бенчмарків. В результаті досліджень було виявлено, що швидкість виконання коду Wasm на 30 % вища, ніж JavaScript. Разом з тим, код, написаний на Wasm, на 50% повільніший ніж двійковий код. З моменту публікації, WebAssembly отримав популярність та кращу підтримку в браузерах, що збільшило його пікову і середню продуктивність.

Отже, для оптимізації роботи з даними у веб додатках доцільніше використовувати WebAssembly у порівнянні з JS, тому що приріст продуктивності може досягати 30 %.

### **Література**

1. Bearman D. Office of the Secretary: Evaluation of Email Records Management and Cybersecurity Requirements, ESP-16-03. The American Archivist. Vol. 80. No. 2. 2017. P. 459–462.