Wydział
Elektryczny
POLITECHNIKA WARSZAWSKA

PERFORMANCE COMPARISON OF WEBASSEMBLY AND JAVASCRIPT

THE COMPARISON OF JAVASCRIPT AND WEBASSEMBLY HIGHLIGHTS THEIR STRENGTHS AND WEAKNESSES IN VARIOUS TASKS. BY EXAMINING PERFORMANCE AND USE CASES, THIS WORK AIMS TO HELP DEVELOPERS UNDERSTAND WHEN AND HOW TO LEVERAGE THESE TECHNOLOGIES FOR OPTIMAL RESULTS IN THEIR PROJECTS.

AUTHORS
Ciszewski Jakub, Myk Kaja
Warsaw University of Technology, Electrical Faculty

INTRODUCTION

Web applications are becoming more complex, and developers seek better performance solutions. While JavaScript is essential in web development, its limitations drive the need for alternatives. WebAssembly (WASM) allows high-performance code to run in browsers. This article compares JavaScript and WebAssembly, analyzing their advantages and practical applications.

METHODOLOGY

Compared JavaScript and WebAssembly across multiple tasks:

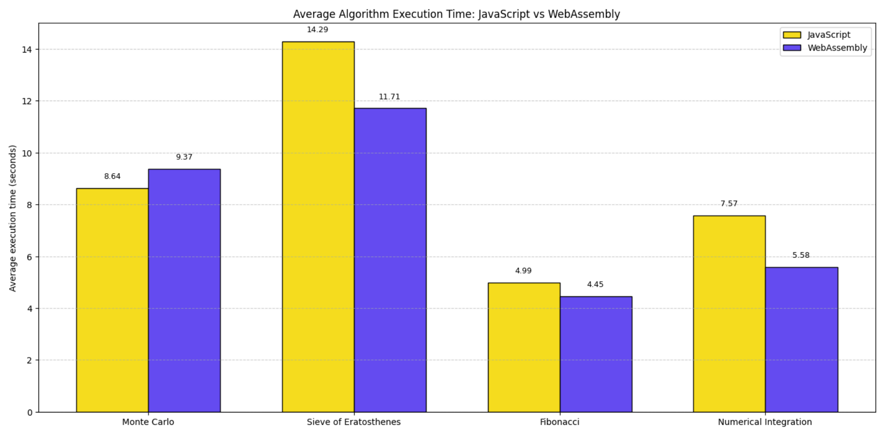
- Algorithms: numerical integration, fibonacci sequence algorithm, Monte Carlo method for π value estimation and Sieve of Eratosthenes elgorithm for finding prime numbers in a scope;
- Implementation: JavaScript (ECMAScript) vs. WebAssembly (compiled from C/C++);
- Performance Metrics: Execution time (ms) and memory usage (MB).
- Testing Conditions: Run in diffrent browsers and operating systems with identical workloads, averaging multiple runs for accuracy.

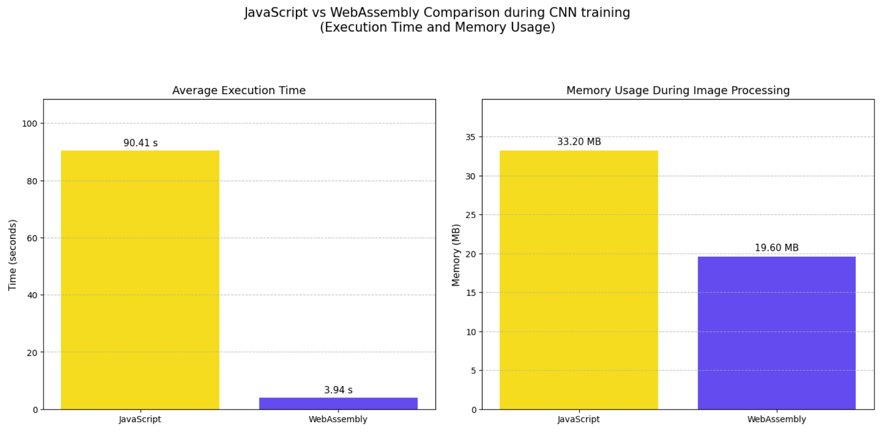
GOAL

Compares JavaScript and WebAssembly on performance and use cases to guide developers in choosing the best technology for their projects.

RESULTS

- WebAssembly significantly outperforms JavaScript in computationally intensive tasks,
- WebAssembly is up to thirty times faster than JavaScript for machine learning (ML) tasks excecution within web browsers.
- Memory usage is much lower for WebAssembly in high-performance tasks, such as image processing.
- Integration complexity: WebAssembly requires an additional compilation step and JavaScript bindings, whereas JavaScript is natively supported and easier to deploy.
- WebAssembly is a powerful complement to JavaScript, rather than a direct replacement. The choice between them depends on the specific use case and performance requirements.





ANALYSIS

The test results clearly demonstrate the superior performance of WebAssembly over JavaScript in computational tasks. WASM consistently outperformed JavaScript in terms of execution time. The most significant speed-up was observed on mobile devices, particularly on the iPhone with Safari, achieving a 30.03x improvement for ML excecution.

In terms of memory usage, WebAssembly proved more efficient, consuming an average of 19.6 MB compared to JavaScript's 33.2 MB during image processing. This efficiency is especially beneficial on mobile devices, where limited resources make optimized performance critical.

SUMMARY

WebAssembly significantly outperforms JavaScript in computationally intensive tasks. It enables faster execution in browsers across Linux and Windows, with Firefox generally outperforming other browsers. WebAssembly also benefits client-side machine learning tasks (e.g., MNIST classification) by improving speed and efficiency. However, it cannot directly access the DOM, so JavaScript is still needed for UI interactions.

K. Myk, J. Ciszewski, "Performance Comparison of WebAssembly and JavaScript," 2025 SITEE 2025