Comparing Blazor and React

Antoni Olszewski

Faculty of Electrical Engineering Warsaw University of Technology

Introduction

Web applications have become increasingly complex, requiring frontend frameworks that balance performance, maintainability, and scalability. The choice of a frontend technology can significantly impact user experience, development efficiency, and long-term application performance. Among the many available frameworks, React and Blazor represent two fundamentally different approaches to building interactive web interfaces.

React

React is a lightweight and flexible JavaScript library widely adopted for building dynamic UIs using a virtual DOM [1]. It is the most popular frontend framework among professional developers according to the 2024 Stack Overflow survey [2].

Blazor WebAssembly

Blazor is a .NET-based framework developed by Microsoft that enables client-side web development using WebAssembly and C# [3]. It offers strong integration with the .NET ecosystem and performs well in scenarios involving static rendering and isolated component updates.

Test Scenarios

Scenario	What is tested
Rendering static elements	Raw DOM rendering speed
Rendering child components	Cost of creating component with many children
Rendering binary tree compoent	Performance in deep component hierarchies
Updating leaf nodes in binary tree component	Efficiency of scattered deep updates
Updating root node in binary tree component	Efficiency of isolated top-level update

Result snapshot

The source code for the benchmarking scenarios, including full implementations in React and Blazor, is available on GitHub [4].

RENDERING STATIC ELEMENTS TIME (MS)

Framework	React	Blazor
N = 100	1	2
N = 500	4	6
N = 1000	9	10
N = 5000	59	44
N = 10000	220	81
N = 25000	1603	234
N = 50000	6650	507

RENDERING BINARY TREE TIME (MS)

Framework	React	Blazor
N = 127	3	15
N = 511	6	50
N = 1023	14	100
N = 4095	48	371
N = 8191	93	743
N = 16383	187	1548
N = 32767	403	3140

UPDATING LEAVES IN BINARY TREE COMPONENT TIME (MS)

Framework	React	Blazor
N = 127	5	<1
N = 511	10	<1
N = 1023	18	1
N = 4095	57	4
N = 8191	111	7
N = 16383	222	15
N = 32767	506	31

Conclusions

This study demonstrates that React and Blazor offer distinct performance benefits depending on the rendering scenario. Blazor showed superior results in static DOM rendering and in updating leaf nodes within deep component trees, thanks to its efficient state management and WebAssembly execution model. React, on the other hand, performed better in constructing complex hierarchical component structures and handling dynamic updates, leveraging its virtual DOM diffing mechanism.

These results highlight that neither framework is universally faster — each has strengths that make it more suitable for specific types of applications. Developers should consider the nature of their UI — whether it emphasizes dynamic interactions or large volumes of structured data — when choosing between React and Blazor.

References:

[1] Ziqi Yuan, Siyu Hong, Rui Chang, Yajin Zhou, Wenbo Shen, and Kui Ren. 2023. VDom: Fast and Unlimited Virtual Domains on Multiple Architectures.

[2] 2024 Stack Overflow Developer Survey, [Online]. Available: https://survey.stackoverflow.co/2024/technology/

[3]Microsoft, “ASP.NET Core Blazor,” Microsoft Learn, [Online]. Available: https://learn.microsoft.com/enus/aspnet/core/blazor/?view=aspnetcore-9.0

[4] A. Olszewski, “Blazor vs React – Performance Benchmark Source Code,” GitHub, 2025. [Online]. Available: https://github.com/AntekOlszewski/BlazorVsReactPerformance