Masterarbeit im Studiengang Computer Science and Media

WIP: Mega-fast or just super-fast? Performance differences of mainstream JavaScript frameworks for web application

---

vorgelegt von

Andreas Nicklaus
Matrikelnummer 44835

an der Hochschule der Medien Stuttgart

am 6. Juni 2024

zur Erlangung des akademischen Grades eines Master of Science

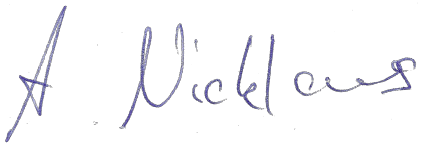
Erst-Prüfer: Prof. Dr. Fridtjof Toenniessen
Zweit-Prüfer: Stephan Soller

Ehrenwörtliche Erklärung

Hiermit versichere ich, Andreas Nicklaus, ehrenwörtlich, dass ich die vorliegende Masterarbeit mit dem Titel: „WIP: Mega-fast or just super-fast? Performance differences of mainstream JavaScript frameworks for web application“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Eislingen, den 6. Juni 2024



Andreas Nicklaus

Zusammenfassung

Diese Arbeit kurz und knackig.

Abstract

This work in a nutshell.

Dislaimer: This paper has been written with the help of AI tools for translating sources and outlining parts of the written content. All content has been written or created by the author unless marked otherwise.

Contents

1 Introduction **4**

2 Related Work **6**

3 Design **6**
3.1 Example Application 6
3.2 Hosting Environments 6
3.3 Testing Tools 6

4 Implementation **6**
4.1 Components 6
4.2 Tests 6

5 Evaluation **6**
5.1 Page Load Times 6
5.2 Component Load Times 6
5.3 Component Update Times 6

6 Conclusion **6**

7 Summary **6**

1 Introduction

Throughout the history of the world wide web, many changes have disrupted the way websites are created. From simple file servers run by few selected institutions, simple static web pages and dynamic services like blogs and forums to websites created with the help UI and web development frameworks, mainly written in JavaScript, development has changed drastically since its beginning. Although older web pages often lacked features, that developers today work with daily, their load and rendering most likely would be brasingly fast with today's technological advancements in networking, browser functionalities and user equipment. Modern websites are often bigger in size, but also have more features and are in many cases more complex. Due to the complex nature of a website's files, the size of a website has increased, especially for multimedia files, which in return has increased the demand for load performance on all components of the load and rendering process. This technological advancement has upped the technological complexity for development tools. Today's modern web development frameworks support developers with tools to create sites and applications through terminal commands and often increase the content-per-line-of-code quota through implicit page generation in contrast to the explicit writing of source code from decades ago. Many frameworks even support configuration options for hosting the webpage.

As the generation process changed from writing code manually to automatically, this implicit page generation undoubtedly increased speed through faster content generation and a greater developer experience for some developers. Because developer experience varies between frameworks and some approaches are more intuitive to some developers than others, a current trend of developers becoming experts in a single framework rather than many has evolved This trend lead to a tribal conflict as to which framework is better than another with each tribe being determined that their framework is the best. There is no appearant way to determine a "best framework" in terms of Developer Experience because it is a subjective criterium. The performance of a framework towards the developer can be similiar or vastly different, depending on the frameworks and the interviewees. When it comes to User Experience and especially the Perceived User Experience however, there are plentyful collections of metrics and criteria to choose from to determine the performance of websites, not frameworks. The optimization of websites has become a goal during development because it has a real effect on both the ranking of web pages in search engines and the user behavior. Both effects create business interests and financial incentives to invest resources into performance optimization. However, the lack of research on the topic suggests either a consensus on the negligant effect of the development framework on the website's performance or a lack of knowledge of the effect. Measurements on the effect of the development framework are a major task simply because the performance of a website in a single case can be dependent on many factors such as the user's device, browser, networking hardware or server-side hardware. The number of possible combinations of factors and their reliability makes it difficult to measure a single performance run with a reliable result and every single result is only a small part of a large number of possible performances the same application could achieve with different parameters. It is therefore perceivable that a "perfect combination" of hard- and software exists

for each framework or in general, but it is currently not possible to find such a combination because the necessary data is missing.

Many modern web tracking services provide data about the user, the user's devices, current page load times and so on. This data is helpful in determining current poor performances and therefore possible starting points for optimization efforts, but it gives very little information about recommended actions or recommended choice of framework for a redesign of a web application. Relying on marketing material for choice of framework is equally helpful because most modern frameworks claim that it is fast, easy to use and performant. This suggests that each would be a great choice for developers. In order to find a suitable framework for an application, a set of metrics have to be at least outlined for comparison. Many former studies suggest metrics to be relevant for the User Experience or Search Engine Optimization. Content metrics such as word count or presence of meta tags might be important for some performance measurements, but might also have no effect on the User Experience. In contrast, rendering metrics such as page load time or page weight might be ascribed to the framework used during development.

The performance of a framework towards the user can very rarely be compared because there are no publicly available comparisons between exact replicas of web applications built with different frameworks. Therefore, a comparitive study between the same website built with different frameworks is needed to get as close as possible to an exact website replica. With this data, an informed choice might be made for other projects. The goals of this paper are to determine a set of metrics that are relevant to comparing mainstream JavaScript frameworks for web applications, create a comparitive study between selected frameworks and create a tool to compare the rendering performance of a page as a whole and of dynamic components within a page.

2 Related Work

3 Design

3.1 Example Application

3.2 Hosting Environments

3.3 Testing Tools

4 Implementation

4.1 Components

4.2 Tests

5 Evaluation

5.1 Page Load Times

5.2 Component Load Times

5.3 Component Update Times

6 Conclusion

7 Summary

References

Shurdi, O., Ruci, L., Biberaj, A., and Mesi, G. (2021). 5G Energy Efficiency Overview. *European Scientific Journal, ESJ*, 17(3):315.