The Future of the Web

Article *in* SSRN Electronic Journal · January 2020

DOI: 10.2139/ssrn.3563679

CITATION
1

READS
371

3 authors:

Manvi Tyagi
Maharaja Agarsain Institute of Technology
1 PUBLICATION **1** CITATION
SEE PROFILE

Moolchand Sharma
Maharaja Agarsain Institute of Technology
64 PUBLICATIONS **401** CITATIONS
SEE PROFILE

Prerna Sharma
Maharaja Agarsain Institute of Technology
26 PUBLICATIONS **406** CITATIONS
SEE PROFILE

3rd INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING AND COMMUNICATION
(ICICC-2020)

The Future of the Web

Manvi Tyagi[1*], Moolchand Sharma[2],Prerna Sharma[2]
Maharaja Agrasen Institute Of Technology,Delhi[1,2]
manvityagi770@gmail.com[1*],sharma.cs06@gmail.com[2],prernasharma@mait.ac.in[2]

ABSTRACT

The Web as an interface channel is an ever-changing field. It is a dynamic tech exhibiting transitions all the time in the tech-stack and the technologies. The recent decade has seen tremendous changes and the growing use of web and internet as a communication channel demands for even more changes.This paper describes the current trends in the web development domain and discusses the future scope of the current technologies being used. It explores the web universe, the successes already achieved in the field, and the research that is going on.

Keywords: *Web, JavaScript, AI, IoT, Web Assembly, Serverless*

1. INTRODUCTION

In 2020, the World Wide Web is playing more roles than ever in the lives of human beings. The open Web continues to advance and prosper as a dynamic place that people use to assist themselves in several ways, be it for work, entertainment or business. Observing trends from the past and the present and analyzing them will always be a variable and predictions on basis of it may or may not be accurate. Web development, as a varied set of numerous technologies, user-experience and interface tool is affected by two main things - disruptive growth in consumer devices including all the smart hardware devices and obviously the financial gain factor which in turn is affected by the newer techniques used in development.

1.1 Increase in Diversification of consumer Devices

In this era of Internet Of Things(IoT), we have seen an enormous increase in the number of smart devices like Smart Mobiles, smart fire alarm, smartwatches, smart door lock, smart bicycle, smart refrigerators, medical sensors, fitness trackers. The rise in IoT has given a chance to web developers to change the fronts and backs of web development. Long gone are the days of a traditional custom website just able to scroll up and down. The modern era websites have the challenge of being compatible and adopt these devices.

1.1 Increase in Demand in-turn requiring better techniques

It is predicted that by 2026, more than 253,000 new software developer roles will be created in the US alone. Web Development's future while talking about Javascript is enormously rich as the ecosystem, environment, reliability and feasibility, along with popularity of the language is on high rise. It has become important to make the process of web development highly optimized and performance efficient. Writing a fully-working web-app is not an easy and quick task. It consists of merging so many pieces altogether, testing, debugging, and whatnot. In a blogpost - "Modern JavaScript Explained for Dinosaurs" by Peter Jang : "We went from plain HTML and JS to using a package manager to download 3rd party packages automatically, a module bundler to create a single script file, a transpiler to use future JavaScript features, and a task runner to automate different parts of the build process."

The future definitely, lies in components. "Components" and "Modularity" are quite popular concepts in the context of web development. "A web component is a standardized way of creating encapsulated, reusable user interface elements for the web."
The significance and popularity of web components have been gained from the fact that they have made it so much easier to build systems that have the ability to support a range of platforms and frameworks via a single codebase.

The discussion in the rest of the paper revolves around the major breakthroughs in the web development techniques and what affects it may create on the whole picture of how developers currently involved in the process of developing highly scalable, reliable and performance efficient web services.

In this research, the progress and advancement of these generations with respect to document types and technologies are presented. The structure of the paper is formatted as follows. Section 2 draws out the role and importance of AI in web development. Section 3 describes the Obstreperous growth and influence of the Internet of Things in Web Development. Section 4 proceeds with a brief discussion around WebAssembly as a language that is the first of its kind to which other high-level languages can be compiled which then can be run in modern browsers. Section 5 illustrates Serverless architecture which is a modern technique to eliminate the need for server software and hardware management by the developer. Section 6 concludes this paper with future work.

2. ML AND AI FOR WEB DEVELOPMENT

Artificial intelligence is a branch of computer science which deals with creating intelligent systems and machines that work and react like humans. In layman language, Artificial Intelligence is training machines to think and behave intellectually or we can say AI is a system by which machines can perform those tasks which require human-like thinking abilities.

2.1 Why Implement AI in Website Development?

Business firms like Amazon, Flipkart, BookMyShow etc are implementing Artificial Intelligence in their web-apps to provide enhanced user experience. The recommendation of products or movies based on the user's search history is one application. Chatbots, Voice Searches, Recommendation Engine, Age or Gender-based automatic filtering , personal advertisements are all part of Machine Learning and Artificial Intelligence. Here are some more advantages of implementing Artificial Intelligence in Web Development.

- Faster, accurate, and personalized search results.
- Increased Customer Interaction
- Effective digital marketing to targeted customers
- Increased and Easy Customer Experience
- Personalized experience of the services.

2.2 Artificial Intelligence can also be widely used in Web Design & Development

The designing process for a website includes a lot of creative thinking and visualization so as to come up with that one perfect eye-catching experience of the web app. After finalizing the design, the designs are converted manually into HTML, CSS code. This requires a lot of human efforts and time. Also the results might not even match the expected standard. These days Artificial Intelligence is working on this field to reduce and remove this manual procedure of converting the design to HTML code. Sketch2Code is a recent tool that does the same. But currently, it is not widely used. The coming era is definitely going to see a major shift in how designs are created and coded.

2.3 AI is being used in various other ways, such as Adobe Sensei

Adobe Sensei is one breakthrough use of Artificial Intelligence in design systems. It just

takes a pinch of the designer's mind by asking for a theme and a rough figure and generates hundreds of new designs on its basis. Adobe Sensei has been created by technologists from all over the world. The companies' business has seen tremendous increase in efficiency and decrease in time consumed while doing the same process manually as a result manifesting increased profits and productivity.

3. OBSTREPEROUS GROWTH AND INFLUENCE OF THE INTERNET OF THINGS IN WEB DEVELOPMENT

Internet of Things, aka IoT is a system of smart interconnected devices that are changing and expected to even bring more change in the age of computer science, information and communication technology. "A recent study by cisco anticipates that by 2020, 50 billion devices will be connected to the Internet. Of that, 90% of automobiles will also be connected to the web, allowing for a larger plethora of quality of life features for passengers."
This exhibits an extensive scope for the web developers to come up with effective ways to implement and include IoT devices in smart ways into web apps to provide enhanced and renewed experience to users,

"The recently-published data refers to the 43% adoption rate of IoT devices in the next five years, further cementing the role of IoT technologies in web development."

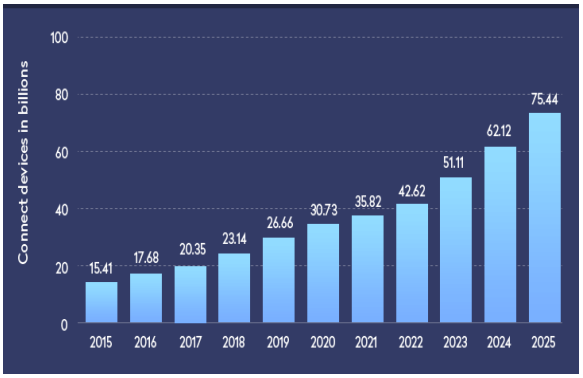

Fig. 1. IoT connected devices installed base worldwide from 2015 to 2025 (in billions)

3.1 Influences of IoT on web development are :

3.1.1 Dynamic development of User Interfaces

UI is said to be static (doesn't depend on user state), example users see some button, but it's grayed out or when it's clicked, a message, that this action is not applicable right now, is shown. UI is said to be dynamic (depend on user state) example users don't see buttons that are not applicable right now. But after some action, buttons may appear/disappear. Dynamic UIs are more preferred in IoT devices so as to accommodate the needs of different users. As a result of this, web developers and designers will have to reconsider and rethink their designing approach from the roots.

3.1.2 Foregrounding security features of the devices using Web Security

IoT devices still have not seen good growth in terms of privacy and security. This being the reason, their popularity and usage have not seen much rise in the past years. In this scenario, web security features blended with IoT devices can literally bring new opportunities and exposure for both web development and Internet of Things in the current market. Privacy and security in IoT devices are highly important and if not handled strictly may bring downside to it. For example, if a room temperature control device is not secure and may be hacked , it may even cause loss of human life and obviously bring bad reputation to the company and IoT as a whole. This being said, Web's security features are one way to integrate with IoT which can result in more secure and reliable devices,

4. LEVELING THE WEB'S SPEED WITH WEB ASSEMBLY

WebAssembly, which is basically a low-level bytecode serving as a compilation target for code written in languages like C and C++, is adopted

by all the major web browsers including Chrome and Firefox. One of the objectives of web assembly is to be at performance par with the native code. WebAssembly is now backed by all major browsers including Google Chrome and has been rapidly adopted by a number of programming languages. Now there are backends for C++, C, Go, Rust and C# that single out and support WebAssembly. The list includes more than 15 other architectures and backends. Code written in these languages as mentioned above can be safely executed in browser sandboxes across any modern device once compiled to WebAssembly. We can say, A major objective of WebAssembly is to beat Javascript's speed. For example, the paper that introduced WebAssembly demonstrated that "when a C program is compiled to WebAssembly instead of JavaScript (asm.js), it runs 34% faster in Google Chrome". This paper also demonstrated that "the performance of WebAssembly is competitive with native code: of the 24 benchmarks evaluated, the running time of seven benchmarks using Web Assembly is within 10% of native code, and almost all of them are less than 2× slower than native code." Figure 2 shows that WebAssembly implementations have continuously improved with respect to these benchmarks. "In 2017, only seven benchmarks performed within 1.1× of native, but by 2019, this number increased to 13."

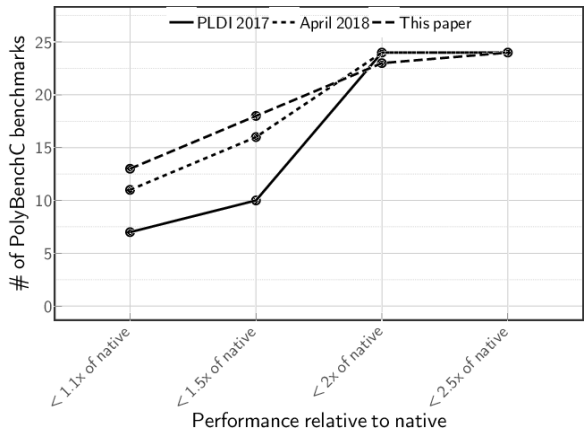

Figure 2: "Number of PolyBenchC benchmarks performing within x× of the native. In 2017 , seven benchmarks performed within 1.1× of native. In April 2018, we found that 11 performed within 1.1× of native. In May 2019, 13 performed with 1.1× of native."

The above observations and increase in performance and optimizations are all the results of an unparalleled collaboration across major browser dealers and leaders and an online community group to build a common solution for high-performance applications. Though Web Assembly itself is originated from the needs and concepts of the web, it is not in any way related to or affected by Javascript or web techniques. It is more like an open standard explicitly made for implanting in various contexts, and as per the growth and popularity of Web Assembly, it is safe to say that future is going to come up with standalone implementations of the same. The commencing versions of WebAssembly involved supporting code mostly from low-level languages like C/C++. Zero-cost exceptions, Threads, and SIMD instructions are some of the features that are still missing in this domain to make it completely comprehensive and independent. Some of these additions are improvements are already being prototyped and worked upon under the hoods of WebAssembly. Beyond this, WebAssembly is intended to further into an appealing target for high-level languages by adding and taking into account pertinent elements like stack switching, coroutines, tail calls etc. Another very important grail is to provide access to the advanced and highly tuned garbage collectors that are built into almost all Web browsers, thus eliminating the main limitation as compared to JavaScript when compiling to the Web. So, we can conclude that "WebAssembly" is expected to find a large number of use cases in the web world and expect that it will possibly come up with new features along with improved and advanced versions.

5. SERVERLESS ARCHITECTURES

Serverless architecture is a way to build applications and services without the need for managing the infrastructure. Notwithstanding the name, serverless computing doesn't mean absence of servers. In actual fact, serverless computing is more like renting out a server for a very small amount of time before releasing it for

another application to use it. Modern applications use serverless architectures and computing for tasks that are CPU intensive and dedicate smaller servers for routing that they tend to scale horizontally afterward. Customarily, these smaller servers are stateless so as they can continue with the execution until the functions return its results ad send the CPU heavy tasks to the lambda functions asynchronously. This leads to creation of a very efficient non-blocking pipeline which is also highly scalable and conveniently maintainable. A big supporter and user of this architecture are Netflix. Since its establishment, developers and tech-companies have discovered a large number of several other use cases and benefits of serverless computing . The most popular and used ones are Automated Backups, Scheduled Cron Jobs, processing uploaded objects analyzing log or simply processing and arbitrarily payload. All these are really heavy and intensive tasks for CPU but with serverless computing, one or the other solutions have been devised already. The serverless pay-per-use business model is attractive and profitable for both the consumer and the provider. In the startup world, apart from the costs for new workforce,one of the major investments lies in buying and managing servers, hosting and managing computing power, in such scenarios , taking advantage of serverless services is a good option generally. "Amazon Spark MapReduce on Elastic MapReduce is known to be the preferred method for this type of task. However, it has also been shown that lambda functions can achieve the same level of parallelization while having a pay-per-use business model."

5.1 Serverless Application Models

"Serverless is a combination of 'Function as a Service' and 'Backend as a Service.'" At high level 'Platform as a Service' looks similar to a serverless approach; however, it is not. As per Adrian Cockcroft, "If your PaaS can efficiently start instances in 20ms that run for half a second, then call it serverless.". PaaS platform's scalability is lesser as compared to FaaS.

5.1.1 Top 3 serverless platforms and their providers

Serverless computing and architectures provide a reflection and mirage to servers, infrastructures, and operating systems.

5.1.2 Amazon Web Services(AWS)

The leader in the cloud space in the market is AWS. Amazon has built the most advanced serverless systems and products. These services are managed in such a way that it becomes very easy for companies and developers to use these serverless applications. Serverless Application Repository was released recently by AWS which provides the starting point for serverless projects. Many publicly available applications are available in this repository which can be directly used, accessed and used for deployment easily.

5.1.3 Azure

Another prominent cloud provider is Microsoft's Azure. A good number and good quality of products have been launched by Azure. For the firms using Microsoft's products, Azure can be a good choice.

Azure Functions are FaaS offering from Azure and can be exposed using WebHook URL in order to work as microservices. This is a good architecture to perform CRUD operations of a single web page. Figure 3 demonstrates the targeted advertising app as per user preferences.

Figure 3. Azure Serverless Architecture for Targeted Ad

5.1.4 Google

Google is the third major name in the cloud share computing services. It provides a wide variety of several serverless applications. has Several Serverless Products covering application development and analytics have been launched by Google Cloud Platform (GCP).

Google serverless applications can be overviewed by the following three most commonly used applications :

• Microservice:

Microservice → Cloud Function → Datastore

• ETL:

File → Cloud Dataflow → BigQuery

• Web Backend:

Browser → App Engine → Datastore

Figure 4 gives a high-level evolution of serverless in GCP.

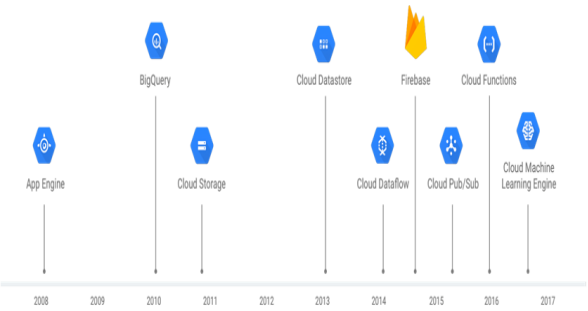

Figure 4. High-level evolution of serverless in Google Cloud Platform.

5. CONCLUSION

This paper described several breakthroughs in web development along with their impact on the Web users and web developers and also discussed the expected future trends, along with some problems in the current scenario and their possible solutions. It may be the case that several techniques or tools will become dominant players in their field, but Our discussion finishes with an exposition of the fact that Web is an ever-growing field with tremendous potential to unveil more and more in the coming decades. The application of Machine Learning and Artificial Intelligence in the development of advanced user-experience websites, the introduction of web-assembly to power up the speed as compared to native code, inclusion, and fusion of IoT with web development or advancements in the field of serverless architectures. All the above-mentioned things provide a wide scope of more advanced future enhancements in the Web Domain of Development and Experience.

References

[1] Abdulelah A. Algosaibi, Saleh Albahli, Samer F. Khasawneh, and Austin Melton - "WEB EVOLUTION - THE SHIFT FROM INFORMATION PUBLISHING TO REASONING"

[2] Haas, A., Rossberg, A., Schuff, D. L., Titzer, B. L., Holman, M., Gohman, D., … Bastien, J. (2017). Bringing the web up to speed with WebAssembly. Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI 2017

[3] https://hackernoon.com/ -
The Future of Web Development - By Eden Ella

[4] JANNA ANDERSON AND LEE RAINIE - "The Future of Apps and Web"

[5] Newman, R., Chang, V., Walters, R. J., & Wills, G. B. (2016). Web 2.0—The past and the future. International Journal of Information Management, 36(4), 591–598.

[6] Manoj Kumar, Serverless Architectures Review, Future Trend and the Solutions to Open Problems"American Journal of Software Engineering. **2019**, 6(1), 1-10"

[7] Pauline Farris - Role-of-IoT-in-web-development - peerbits.com

[8] Al-Osta, M., Ahmed, B., & Abdelouahed, G. (2017). A Lightweight Semantic Web-based Approach for Data Annotation on IoT Gateways. Procedia Computer Science, 113, 186–193.

[9] Abhinav Jangda, Bobby Powers, Emery D. Berger, and Arjun Guha, University of Massachusetts Amherst "Not So Fast: Analyzing the Performance of WebAssembly vs. Native Code" https://www.usenix.org/conference/atc19/presentation/jangda

[10] Andreas Haas, Andreas Rossberg, Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman, Luke Wagner, Alon Zakai, and JF Bastien. Bringing the Web Up to Speed with WebAssembly. In Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, pages 185–200. ACM, 2017.

[11] Samuel Lavoie and Anthony Garant and Fabio Petrillo - "Serverless architecture efficiency: an exploratory study."

[12] D. Shadija, M. Rezai, R. Hill, in 2017 23rd International Conference on Automation and Computing (ICAC) (2017), pp. 1–6.

[13] M. Roberts. Serverless architectures. https://martinfowler.com/articles/ serverless.html, (2016).

[14] B. Doerrfeld. The benefits of a serverless API backend. https://nordicapis.com/ the-benefits-of-a-serverless-api-backend/ (2017).

[15] B. Wootoon. Who's using AWS lambda? https://www.contino.io/insights/ whos-using-aws-lambda-1 (2017).