Web Development and performance comparison of Web Development Technologies in Node.js and Python

Sai Sri Nandan Challapalli
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
ch.saisrinandan854@gmail.com

Prakarsh Kaushik
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
prakarsh.kr@gmail.com

Shashikant Suman
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
sshashikant18@outlook.com

Basu Dev Shivahare
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
basuiimt@gmail.com

Vimal Bibhu
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
vimalbibhu@gmail.com

Amar Deep Gupta
Dept. of Computer Science
Amity School of Engineering and Technology
Greater Noida, India
amardeep.ip@gmail.com

2021 International Conference on Technological Advancements and Innovations (ICTAI) | 978-1-6654-2087-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICTAI53825.2021.9673464

***Abstract*— From the first website being created in 1991 by Tim Berners-Lee to this day in 2021, the world of websites and their development has seen a rapid evolution. Every business nowadays has a website to make its online presence in this digital age. Website use is not only limited to this, but many people use them to build their online portfolio, apart from that there are many web-based applications/software, etc. This research paper discusses the process involved in developing a website in past and present, development of content delivery over the years, the website uses, a website for mobile devices, and performance comparison between two of the most used web backend development technologies, i. e, Node.js and Python. For comparing performance, we have used Locust – an open-source software and Autocannon and tested both of them under similar conditions.**

Keywords—Web Development, Node.js, Python, Website Technologies, Content Delivery, Mobile Websites, Locust, Autocannon, Framework, MERN stack

I. INTRODUCTION

The very first websites were simply HTML single- column pages with no images. The HTML standard evolved to include a plethora of fantastic options to begin putting up data. It began with simple table-based layouts and multi-column web pages. Then there was CSS, which allowed the user to design their information so that it didn't appear like the standard white background, black text, and blue emphasized links. Eventually, we saw a shift towards the more complicated frameworks. We've progressed from text blocks to content that can be oriented with CSS to pre-made templates that can be utilized straight with Bootstrap and Material UI.

With the advancement of the internet nowadays, numerous sites are encountering new issues, like the challenge of multi-user requests and high concurrency. JavaScript, a dynamic scripting language, has grown in popularity among clients and is extensively employed in Web development. Node.js is one of the most popular JavaScript technology. Node.js is a cross-platform runtime environment for easily creating fast, scalable web applications that are built on Chrome's JavaScript runtime. Node.js has an event-driven, non-blocking I/O approach that is compact and efficient, making it ideal for data-intensive real-time applications running across dispersed devices. According to Node.js official website popularity polls, Node.js has received over a billion installs since its debut in 2009. Furthermore, according to a 2020 study, more than half of backend developers prefer Node.js. Companies are increasingly seeing the value of Node.js. With 899,344 repositories, JavaScript is now the most popular language on GitHub.

Python, unlike Node.js, is a single-flow language, and requests are handled much more slowly. Python was first launched in 1991, and many development tools for web, systems, and frameworks have been produced since then. Python is still a popular choice for Machine Learning, Image Processing, and Software Development. This paper focuses on process involved in developing static and dynamic websites, development of content delivery over the years, website then and now, how mobile websites has rapidly evolved and their optimization along with performance comparison between two most famous technologies used in web development i.e, Node.js and Python.

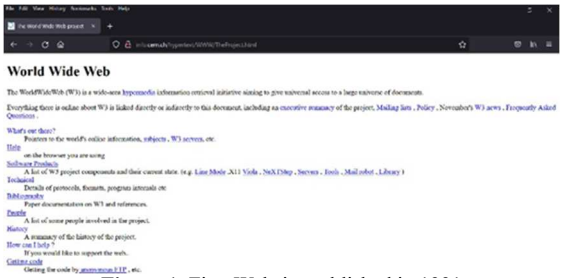

Figure 1: First Website published in 1991

978-1-6654-2087-7/21/$31.00 ©2021 IEEE

II. Process Involved In Developing A Website

The first step in developing anything is creating a blueprint or design for the product, such is the case for websites as well. Websites are designed according to the need of the business or the use. Some of the famous designing open-source design software are Figma and Adobe XD. The next step is to create a frontend for the website with which the user will interact followed by adding backend to carry out other processes like saving data, using dynamic frameworks, carrying out complex process etc.

The frontend technologies consist of HTML which is the basic structure of a webpage, CSS is used to style and make the content look good, JavaScript is used to make the frontend dynamic. The backend technologies consist of a variety of options ranging from php to python, databases, APIs and MERN, MEAN stack being the most famous ones. In this section, we'll discuss only frontend technologies used in almost every website and bootstrap which is the most famous framework, and later evaluate the performance comparison between python and node.js.

A. ***HTML***: HTML stands for Hyper Text Markup Language. It is used to define the basic structure of a webpage. It uses a various syntax or notation to arrange, organize and give information about the page to the browser. Tim Berners-Lee created the first version of HTML in 1993. The term "hypertext" refers to a document that has linkages that allow the reader to navigate to different parts of the page or to another content entirely. HTML5 is the most recent version of HTML.

A Markup Language is a mechanism for computers to communicate with one another in order to govern how text is processed and displayed. Tags and Attributes are used to do this. Tags are used to indicate the beginning of an HTML element and are typically wrapped in angle brackets. Attributes are extra bits of information. Attributes come in the form of an opening tag, with additional information contained within.

B. ***CSS***: CSS stands for Cascading Style Sheets. It is used to style the webpage and make it look more attractive as well as responsive. CSS, in a nutshell, is a design language that makes a webpage look more attractive than bare or unimpressive content. CSS is responsible for visual structure, design, and appearance.

CSS is largely credited to Norwegian Hkon Wium Lie, who wanted to create a universally standardized style sheet for the World Wide Web in 1994. However, the W3 Consortium has standardized CSS. CSS is a free, open, and independent standard. CSS, like HTML, has a working group that is concerned with standard development. This is done in accordance with W3C guidelines. CSS comes in three versions. CSS level 1 was initially released in 1996 and revised in 1999 with corrections. CSS level 2 debuted in 1998, building on CSS level 1. CSS level 2 includes support for a variety of output media. CSS 3 is being used at the moment.

C. ***JavaScript***: Brendan Eich created JavaScript in 1995. It just took him ten days to develop this scripting language known as JavaScript. It was created for Netscape 2 and was adopted as the ECMA-262 standard in 1997. Following Netscape's handover of JavaScript to ECMA, the Mozilla foundation continued to evolve JavaScript for the Firefox browser.

JavaScript enables us to add dynamic characteristics and interesting effects to the webpage. On websites, it is primarily used for validation. JavaScript allows us to conduct complicated activities and also allows websites to communicate with users

D. ***Bootstrap***: Bootstrap is a open-source front-end framework that offers HTML and Style sheet design templates for fonts, fields, button, tables, menus, modals, picture carousels, and other features, as well as additional JavaScript plugins. Bootstrap additionally allows users to quickly construct user interfaces.

Twitter built Bootstrap in the middle of 2010. Bootstrap was originally known as Twitter Blueprint before it became an open-source framework. Twitter hosted its inaugural Hack Week a few months into production, and the project boomed as coders of various skillsets rushed in without any external guidance.

III. Development Of Content Deliver Network (CDN) Over The Years

Content Delivery Network (CDNs) are proxy web servers that are dispersed throughout several data centers in various geographical locations to provide content to end-users based on their proximity. Thus, using a CDN hosting system rather than a normal one provides a cost-effective way for an online seller or e-commerce company to keep his clients pleased.

It also implies that a hosted website will perform better and will be more secure against hacking attempts. This is due to the fact that CDNs retain several Points of Presence, i.e. servers that keep copies of similar data and use a method that sends logs and data back to the origin servers. Two communication flows are employed rather than the conventional client-server communication. Between the user and the proxy server, followed by the proxy server and the origin server.

Before the late 1990s, there were no CDNs. However, some technological advancements that predate this era of CDNs, such as server farms, hierarchical caching, caching proxy

deployment, and so on, were critical in laying the groundwork for the required infrastructure of such internet unclogging technologies. First-generation CDNs were created to meet the increased demand for movies and music streaming, to speed up sites, and to accommodate expanding volumes of material. Finally, to help enterprises that provide products or services to handle all requests from Internet consumers while without incurring large revenue losses due to IT infrastructure issues.

However, the primary focus of second-generation CDNs was the peer-to-peer transmission, cloud computing, and energy efficiency. Also, to meet the internet crowd's want for more - involvement. Not only from a desktop browser, but also from a mobile device. There are also a large number of ISPs, telecommunications businesses, IT corporations, and traditional broadcasters all around the world. Some have gone into the CDN industry. Third-generation CDNs are expected to be entirely driven by the community. Self- sufficient and self-managing. Its primary goal is to improve the end-user experience.

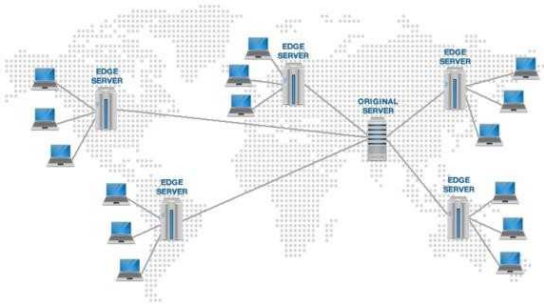

Figure 2: Content Delivery Networks (CDNs)

IV. What Were Websites Used For In Past And What For Now

Due to limited availability of resources in the past, the process of web development was very complex. Only organizations were capable of building websites, ironically only using HTML with the blue hypertext being the only splash of colour. The only task the websites were capable of doing was display organized information. Starting 1991, when the first website was launched [11], the web development industry saw enormous growth unlocking tremendous potential. In the past 30 years, a lot of new technologies, libraries and frameworks have been developed. CSS and JavaScript are the most noteworthy, given the amount of revolution they brought. Availability of better hardware equipment and growing economies also favoured this outcome. Modern websites can be used for a slew of tasks, ranging from multimedia display to payment platforms. Also modern websites need to be fast, secure and developers are also looking out for SEO.

V. How Mobile Websites Can Be Optimized

Around 50% of the website traffic comes from mobile devices.[12] Thus, websites are optimized to give a better user experience while using mobile phones to view. Specific technologies and frameworks are available today to develop user friendly mobile applications.

- ***Screen Resolution****:* Modern websites are made to be responsive. The layout of content, size of font and images are adjusted according to the screen resolution. Also, buttons are made large for quick access.
- ***Latency****:* Mobile devices, due to their wireless connection type, have high latency. Mobile phones are not capable of quickly downloading and running huge amounts of data and JavaScript. Even on Wi-Fi or 4G networks, the latency is still high when compared to wired connections.
- ***Redirecting****:* Redirecting the user to new webpage induces latency in the process.
- ***Memory Constraints:*** Mobile phone have limited memory and slower CPU's, due to which, webpages that run smoothly on desktops can run slowly on mobiles.

VI. Comparison B/W Python Server Using Flask And Node Js Server With Express

Experimental Environment:

A. ***Hardware Environment***: One device is setup to host the servers and run the benchmark software. The device runs Windows 10 (64 bit) operating system with Intel i5 (8th Generation) processor and 8GB RAM memory. During the time of the test, all non- essential processes on the system were disabled to maximize the availability of resources.

B. ***Server Configuration***: Two servers, Python and Node JS are hosted on the device. We built the servers with only one route, i.e., the home route. Both the servers respond with a "Hello World" when a get request is made to the home route.

1. ***Python Server***: Python programming language is beginner friendly and allows the coder to work fast with its easy and readable grammar. Python is used in modern day for many applications like machine learning and artificial intelligence. In our code for this test, we used Flask framework for python webserver.
2. ***Node JS Server***: Node JS is a server-side coding language built off Chrome's JavaScript runtime. It executes JavaScript outside the browser. It enables building fast and scalable network applications. We chose Express framework to build the Node JS server for the test.

C. ***Testing Tools***: We used two third party testing tools in our experiment for comparison of servers. We ran the tests twice in order to make sure that the results are consistent. The test is run for a time period of 30 seconds and the software are configured to simulate 10 users.

1. ***Locust***: It is an open-source load testing tool, that helps to test websites/services. This python based software is scalable and hackable to suit our needs.
2. ***Autocannon***: It is a HTTP benchmarking tool written in node. We have used the latest version Autocannon 7.4.0. Autocannon lets the user determine various parameters for the test. Like Locust, Autocannon also bombards the given URL with simultaneous requests and tracks the response.

D. ***Test Methodology***: Through this test, we evaluated the results in two respects, one being the responses per second, the other the response time. The test preparation involves starting up the server and pointing the software to the same port on local host. When the test is started, the software will make multiple requests to the '/' route (also known as home route) of the webpage. The server will respond with "Hello World" to each request. The "Hello World" is the most basic working server model and will help in distinguishing the internal differences in these technologies. The software are configured to simulate only 10 users because higher numbers will mean increase in demand of resources both by benchmark software and the server. Since both are run on the same device, it is optimal to keep the number of users limited in order to avoid excessive CPU usage which can cause irregular time calculation and server response.

E. ***Results and Analysis of the test***: Node JS clearly outperformed the Python server. The number of requests handled by Node JS were almost 250 times higher than those handled by Python. Node JS, on an average has handled 30616 requests in the 30 second time frame. The number of requests per second of Node JS showed a steep increase with increase in number of users as opposed to the Python server that had a linear and delayed increase. It is important to note that the failure percentage in both the cases was 0%, confirming that the system resources were adequate to run the test.

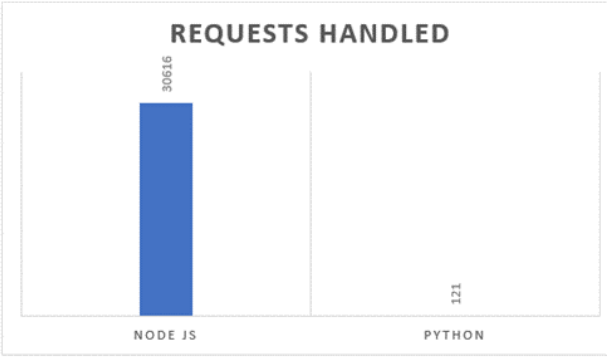

Figure 3: Total number of requests handled by the servers during the test

An important measure of the performance for any server is the Mean Time per Request. Under same conditions, Node JS had an average time of 7ms while Python took around 2040ms to respond to a request.

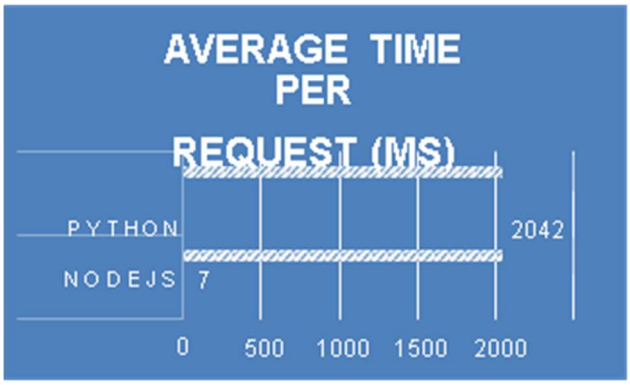

Figure 4: Average Time per Request of the two servers

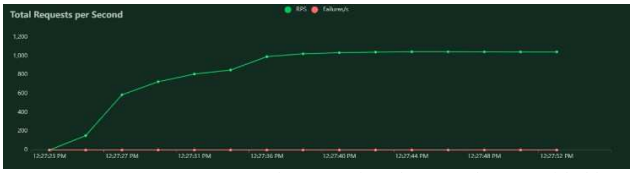

Figure 5: Requests per second (RPS) of Node JS – Chart provided by Locust

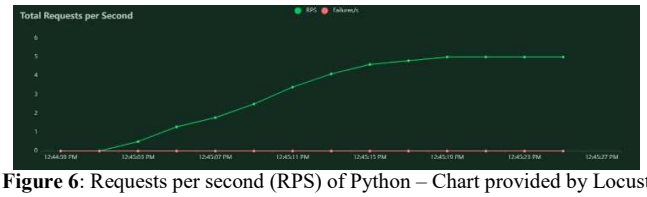

Figure 6: Requests per second (RPS) of Python – Chart provided by Locust

Latency is the time taken for the data transfer from the server to the destination, i.e., the browser. The average latency of Node JS was only 1.04ms which is very fast. This indicates that the Node JS server performs better where the number of users is high. On the other hand, Python server's latency increased many fold with increase in users before plateauing at 14000ms towards the end of the test. The average latency was 7187ms.

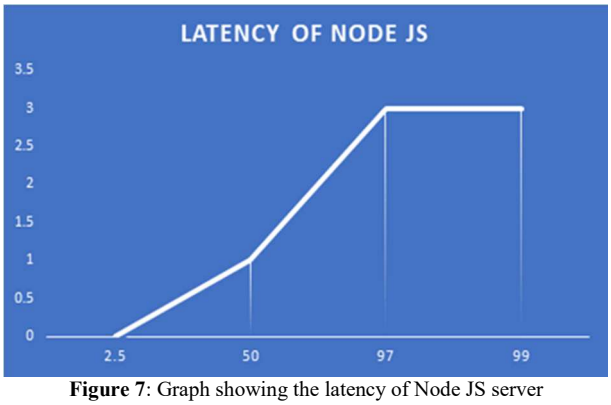

Figure 7: Graph showing the latency of Node JS server

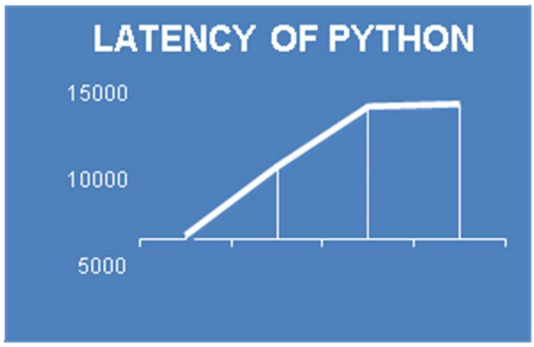

Figure 8: Graph showing the latency of Python server

VII. CONCLUSION

The web development industry has seen one of the most enormous and fastest growth in the history of mankind. Through this paper we made an attempt to consolidate the major milestones and breakthroughs in web development. Today's websites are a perfect amalgamation of modern technologies, hardware and creative minds. These enable the programmers to create things that were once not imagined at all. While we keep creating more websites and generate millions of GB of data, servers take the responsibility of keeping the information safe and responding with data when requested.

In this paper, we presented a comparative analysis of two web servers - Node JS and Python. The comparison was done on basis of Requests per Second (RPS), latency and total number of requests handled. In all three tests, Node JS has performed better than Python server. Node JS was better at handling multiple users submitting requests simultaneously.

In this paper, we have tested the servers using a Hello World module that depicts a basic working server. The architectural differences or other factors that influence the working of server are not taken into consideration. Inclusion of a data base can add a new parameter for comparison, but it is out of scope of this experiment. In our further experiments, we wish to compare servers that are built to perform specific tasks.

REFERENCES

[1] Lei, K., Ma, Y., & Tan, Z. (2014). Performance Comparison and Evaluation of Web Development Technologies in PHP, Python, and Node.js. 2014 IEEE 17th International Conference on Computational Science and Engineering. Published. https://doi.org/10.1109/cse.2014.142

[2] Miller, C. S., Zheng, J., Connolly, R., & Olagunju, A. (2013). Keeping up with web development trends. Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education. Published. https://doi.org/10.1145/2512276.2512309

[3] Zakas, N. C. (2013). The Evolution of Web Development for Mobile Devices. Queue, 11(2), 30–39. https://doi.org/10.1145/2436696.2441756

[4] Craig, B. W. (n.d.). *Evolution of Websites: A Darwinian Tale*. WebFX Blog. Retrieved October 4, 2021, from https://www.webfx.com/blog/web-design/evolution-of-websites-a-darwinian-tale/

[5] *Web Survey Report 2018*. (n.d.). Node.Js. Retrieved October 4, 2021, from https://nodejs.org/en/user-survey-report/

[6] T.Lance, A.Martin and W.Carey, "Performance Comparison of Dynamic Web Technologies", ACM SIGMETRICS Performance Evaluation Review, Volume 31 Issue 3, December 2003

[7] A.Ranjan, R.Kumar, J.Dhar, "A Comparative Study between Dynamic Web Scripting Languages", Data Engineering and Management, 2012.

[8] J.Hu, S.Mungee, and D.Schmidt, "Techniques for Developing and Measuring High-Performance Web Servers over ATM Networks", Proceedings of IEEE INFOCOM, San Francisco, CA, March/April 1998.

[9] S.Tilkov, S.Vinoski, "Node.js: Using JavaScript to Build HighPerformance Network Programs", IEEE Internet Computing, 2010.

[10] Arie, D. (2021, May 19). *The History of Content Delivery Networks (CDN)*. GlobalDots. https://www.globaldots.com/resources/blog/the-history-of-content-delivery-networks-cdn/

[11] Nix, E. (2018, August 31). The World's First Web Site. HISTORY. https://www.history.com/news/the-worlds-first-web-site

[12] Statista. (2021, April 28). Share of global mobile website traffic 2015–2021. https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/

[13] Geekbench 5. (2021, October 3). HP HP Laptop 14q-cs0xxx- Geekbench Browser. https://browser.geekbench.com/v5/cpu/10210650

[14] D. P. Singh, P. Kaushik, M. Jain, V. Tiwari and S. Rajpoot, "Data Storage Security Issues in Cloud Computing," 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), 2021, pp. 216-220, doi: 10.1109/ICIPTM52218.2021.9388321.