Node.js Challenges in Implementation

By Hezbullah Shah & Tariq Rahim Soomro

SZABIST Dubai

Abstract- Node.js gave rise to the Full Stack Developers who are now able to manage server and client side by their own. Node.js is fast and reliable for heavy files and heavy network load applications due to its event driven, non-blocking, and asynchronous approaches, where developers can also maintain a complete projects in single pages (SPA) and can use for IOT. The result of study concludes from a survey and from literature review the implementation areas and challenges of the Node.js. Lastly will provide suggestion on how to improve to overcome the challenges.

Keywords: *node.js, javascript, SPA, IOT.*

GJCST-E Classification: *I.2.4, I.7.2*

NODEJSCHALLENGESINIMPLEMENTATION

Strictly as per the compliance and regulations of:

Node.js Challenges in Implementation

Hezbullah Shah [α] & Tariq Rahim Soomro [σ]

Abstract- Node.js gave rise to the Full Stack Developers who are now able to manage server and client side by their own. Node.js is fast and reliable for heavy files and heavy network load applications due to its event driven, non-blocking, and asynchronous approaches, where developers can also maintain a complete projects in single pages (SPA) and can use for IOT. The result of study concludes from a survey and from literature review the implementation areas and challenges of the Node.js. Lastly will provide suggestion on how to improve to overcome the challenges.

Keywords: node.js, javascript, SPA, IOT.

I. Introduction

Web based applications are increasing its popularity as they become easier to develop, maintain and secure. Also they are easily reachable to the clients and does not require additional installations in most cases and are quickly customizable [1]. Web application is derived from web based system, which have additional functionality to execute business logic of an organization. These applications are totally web based instead of requiring to install a separate application on the operating system [2]. Google Docs, Web based retail stores, Google Maps, and the web based email applications are kind of Web applications [1][3].

The Web development industry will find two kinds of developers. i.e. Front-end developers and Back-end developers[3]. Front-end developers require to have knowledge of HTML, CSS, and a programming language to add effects and more to the front – end i.e. JavaScript. They build the web sites display and effects which are shown to the clients by converting the designer's design [3].Back-end Developers build the business logic behind any web application. The actions for instance adding and retrieving news highlights to and from a web applications, or sending email from a web based forms, or authenticating a visitors or clients credentials are all part of back-end developers. A back-end developers need to know languages like PHP, .NET, Java, and others [3][4].Back-end developers should also have knowledge of databases like My SQL, Oracle, and SQL Server, or should hire or entrust a database administrator to work with the flow. A database administrator will take care of database server and ensure its smooth performance [5] [4].

Full stack developers are jack of all trades, and they are the one who do all. Mostly back-end developers are required to have skills of front-end developers and vice versa and have extra burden on them of learning additional skills set[4].So it is clear that for a developer to become full stack, he have to have expertise in languages of three kinds, i.e. Client Side languages such as Java script along with HTML and CSS, Server Side Languages such as PHP, .NET, Java, Ruby, etc, and database expertise such as SQL Server, MySQL Server, and Oracle. As depicted in Figure 1.
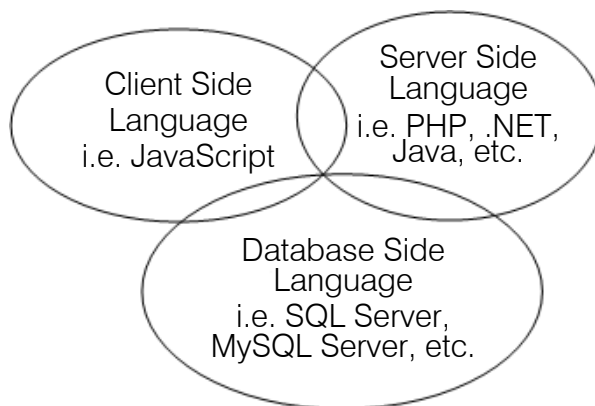
Figure 1: Full Stack Developer

Node.js is overriding the complication of learning multiple languages in the same time to become a Full Stack developer where a front-end developer who is confident in JavaScript just requires to learn some additional Server Side Functions and he becomes a Back-end developer [6] [7]. Apart from this where Node.js is free, it is also used by thousands of developers around the globe [8].While the web application demands performance and scalability, history have already adopted threading and events

Author α σ: Department of Computer Science, SZABIST Dubai Campus, UAE. e-mails: email@hezbullahshah.pk, tariq@szabist.ac.ae

either separately or its combination. But its daily internet traffic's increasing nature demand better and new solutions to improve the concurrent nature of the service. One solution is Node.js (which sometimes is also termed as Node [9]) which is a lightweight, and fulfills the demands through event-driven and non-blocking I/O model and server-side JavaScript [10][11]. Chrome's V8 JavaScript engine is the basis for Node.js as Node.js JavaScript runtime is built on it [11]. Written in C++, V8 is an open source from Google as a high performance JavaScript engine. Google have used V8 in Google Chrome browser, and is capable to run standalone and can also be embedded into C++ applications. It is capable to run on different versions of Windows, Mac OS, and Linux [12]. As Node.js is event-based and not thread-based, it is also capable of scaling to millions of connections concurrently, while using an event loop within a single thread and not making overhead of multiple threads. Node.js processes its I/O operations asynchronously and is widely misunderstood with AJAX. While Node.js and AJAX are considerably different [9].

a) *Advantages of Node.js over others*

Node.js is built from ground for the purpose of handling asynchronous I/O as it is built of JavaScript and JavaScript is built as event loop. Like the on click event for a button in client side JavaScript is and event loop. While other environments do have this feature, they have it with using third party libraries or are not built from ground for the same purpose like the Node.js and hence they are often slow, or lags and does not belongs as a standard feature to them. Some of the examples are Event Machine – built for Ruby, Twisted – licensed under open source MIT License, it is introduced for Python and is available since Python 2 onwards, and network framework library for Apache named as Apache MINA which is also called "Networking Socket Library" and is another example of providing event-driven and asynchronous limited to APIs only. Similarly Apache AsyncWeb is built using Apache MINA and Perl's Any

Event. Similarly an edge of Node.js over others will be that it will be capable of handling multiple request while it will act like a client towards the third party services by executing only a single thread. Other languages in this regard will block the processing until the remote server responds first for their initial request as a result they will be requiring multiple threading for executions. Comparatively in Node, all what you will use is asynchronous as it will become quite hard if you are to write non-asynchronous code in it. Also Node.js will never force to buffer data before outputting while the others like Event Machine, forces buffering in many cases to buffer the data[7][6][13][14][15][16][17].

Being server side JavaScript, another admirable edge of Node.js over others is that a developer will be required to only have knowledge and experience of a single language i.e. JavaScript, no matter if he is developing client side scripts or scripts for server side. The developer is not required to swap his brain cycles from for one language at client side and then for another language at server side. Hence JavaScript end-to-end as depicted in the Figure 2. Here the database of JavaScript also store date like a JavaScript Object. Adding to this, it is also worth considering that Node.js is new and thus have benefit of taking precautions against the mistakes which other languages had come across in the past such as the mistake of backward compatibility[7][6]. Figures shows that about 47% of web surfers wants a website to be loaded within 2 seconds and a 3 second delay drops the customer satisfaction by 16% [18]. Here the Node.js leads as the interpreter of Node.js is smaller and fast compared to other languages like PHP. Here the server side apps are permanently kept ON unlike other languages where every initiation of the application will follow cycles consuming steps of for example loading configurations, followed database connectivity, acquire required information and finally render the markup language. Node.js on other hand reduces these steps by keeping an app permanently ON [19].
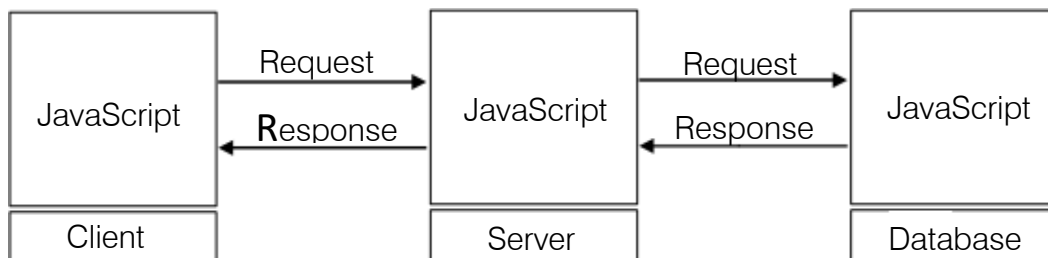

Figure 2: JavaScript end-to-end

b) *Disadvantage of Node.js over others*

A code in Node.js becomes fast growing, along with making it difficult to debug due to the fact that it is using event-driven/callback approach [7]. Big drawback currently is the unavailability of ready hosting for Node.js

environment. Complex topics of JavaScript language such as prototypical inheritance, anonymous function, callbacks make the language hard to learn, and thus becomes the choice to be learnt when one has mastered another easy language first [20]. Node.js is

not a mature language still and thus faces so much hesitation to get joined by expert programmers. Another issue is that being single-threaded, other requests are stopped if in case the CPU is occupied even for some parts of a second. And so the developers are also forced to think in asynchronous which is not easy to get used with [21] [22].

The scope of the study is to make struggles for exploring the implementations of Node.js and try to catch on how it can be made easy to implement for new comers specially. Also it tries to explore challenges to Node.js that why it is still not being very widely implemented and why new comers to Node.js are less? Efforts will be made to combine the ideas with real experiences and not suffice just theories. The study will be exploring the issues in the Node.js implementations and adoption if any and provide recommendations for those challenges. This study will make the reader interested in the adopting of Node.js as the features and possibilities of Node.js will be uncovered to the reader. The result of this study will help developers to find out reasons and methods to quickly adopt the Node.js and help Node.js become popular in the market similar to other top languages like .Net Languages and PHP. This paper is organized as follows, section 2 will explore the review of literature, section 3 will talk about material and methods used for this study and section 4 will depict the results and finally discussion and future work will be discussed.

II. Literature Review

This section describes about the existing research on the Node.js as a server side language and areas where Node.js is implemented practically, along with having some general review about the same like History of Node, and some application areas of Node.js and is it efficient or not.

a) About JavaScript and Node.js

The history of JavaScript dates back to the beginning of WWW and it started playing integral role in making the front ends of websites interactive. JavaScript introduced its concepts in AJAX in late 1990's which is again used to add real time like behavior to the web pages. Till this time the JavaScript was always believed to be a client side scripting language and that it has nothing to do with the server side programming. However by the development of server side JavaScript like Node.js, the concept changed and now the JavaScript is not only a client side scripting programming language, but also it is also capable of executing on server side. Node.js is one of the main competitor in the JavaScript on server-side era. Again it is important to mention that Node.js is different from JavaScript, while JavaScript is no doubt backbone of Node, Node.js is just build on top of JavaScript only and hence use the same language [9].

b) History and Evolution of Node.js

The created Ryan Dahl was first motivated to build Node.js from a progress bar showing file upload at Flickr (a company of Yahoo dealing with pictures galleries [23]) where the browser was asking server again and again about how much size of file is being uploaded [24][25]. And on March, 2009 Node.js got its name for the first time and the package manager for it was also introduced in the same year in October and an early preview of npm (the Node.js Package Manager) was introduced. Later in the same year in November, the creator Ryan Dahl did the first talk about Node.js at JSConf 2009 where he talked about Node.js in detail. He explained how Node.js is event based, and works on callbacks where every function of I/O as receiving input from disk, network or any process, should use callback and the audience stoop up to applaud for his extra ordinary project. In 2010 Express framework was introduced towards Node.js [26][17]. The node.js was not available for Windows Environment until July 2011 when Microsoft partnered with Joyent for this purpose and support of even older version of Windows Server was added [27]. Later in end of 2014 some of the enthusiasts from Node.js team boycotted Node.js and created their own fork from Node.js naming it io.js or iojs. They boycotted as they were unhappy with the control of Joyent over the project [28]. The following year, both of them again seem to merge together as they both voted to form a neutral Foundation of Node.js and finally in the same year in September 2015 they merged back and their integrated community made amazingly progress in emerging a joined codebase[29][30].

c) Architecture of the Node.js Platform

With the build of Node.js, the web servers got event driven programming. Which made the web server fast and in a language widely popular i.e. JavaScript, and this is also the reason that the entire web development community got access to Node.js within no time. In Node.js developers are not required to make use of threading while creating greatly scalable servers. Node.js uses a very easy event-driven model which uses the triggering of callback functions upon completion of a task or generation of error. Ryan Dahl created the Node.js behind the idea that other programming languages makes it difficult to program things to make them work concurrently[31][32].

d) Adoption of Node.js

Node.js is adopted by giants such as PayPal, LinkedIn, Medium, and Netflix to mention among the many [33]. Face book adopted Node.js and it proved for them extremely proficient and then they adopted it for production [34].Microsoft made a worthy change of making Node.js a part of the developer stack mainly since then developers can now use it on Azure platform also along others and is not restricted to only old

Operating Systems[35]. Walmart adopted end-to-end JavaScript by using Node.js believing that Node.js will be helping them to front their services which they are providing all over the world. They mentioned they adopted Node.js also because it is not just introducing new thinking of how to build perfect software, but also great way to express the existing things in a new way [36].According to the statistics by GitHub, JavaScript is the most popular language. The statistics graph is shown in Figure 3[37].
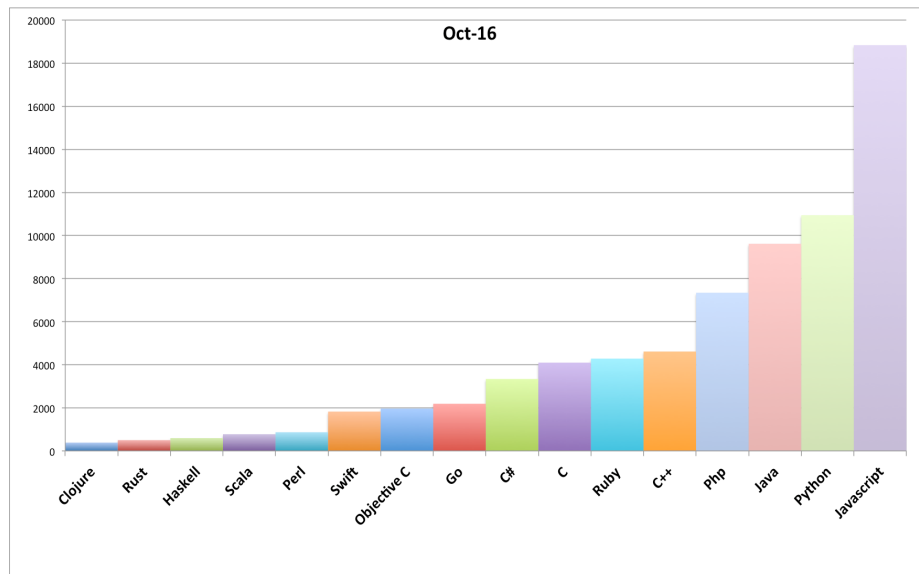

Figure 3: Javascript the most popular language[37]

e) Application of Node.js

Node.js was compared with PHP/Nginx for Performance and Scalability, where Node.js showed high performance and scalability compared to PHP/Nginx. The two researchers developed web application on Dijkstra Algorithm and simulated load of concurrent user requests using some load generators [38]. University of Notre Dame in [10] had a project report issued where the author performed tests between Node.js and Ruby's Event Machine and Apache's threading model, which measured request time over the number of cores. Node.js had again beaten the other two especially when the number of cores were increased. Node.js based DPWS – Devices Profile for Web Services (termed as Node. DPWS) was evaluated for performance and compared with alternative DPWS tools. The researcher concluded that Node.js based DPWS was easy to use as well as light weight. Node. DPWS had beaten even the most striking substitute of it in the field of IoT (Internet of Things) [39]. Node.js also plays important role in GIS field and a paper published in Journal of Korea Spatial Information Society used Node.js in Web GIS. They implemented Node map and concluded in their research that Internet GIS have its future bright in combination to the Node.js[40]. Another analysis study at [9] has shown that Node.js can be used to have complex real time applications and that these application can be served for millions of client connections. There is a web service named MAGI which is used in infrastructures of graphic processing unit (GPU) for the purpose of fast data analysis of Micro

RNA-seq. While MAGI is based on Node.js, it did addressed the limitation of other similar tools to not being able to handle large files, and a burdensome error prone steps. Also MAGI helped uncovered the delay in downstream analysis as the others are time consuming and solved the issue of others of not being able to provide statistical tests [41].

i. Single Page Applications

Single Page Applications or SPA is a term given to web based applications which does not cause a page to reload during use. While SPAs have history from quite a long time, and is based on Java, Flash and JavaScript. It is known that JavaScript does not require any third party client plugins like the former ones. And that is the reason that Node.js as it is based on JavaScript becomes successful in competing the others in SPAs [42].

ii. NodeOS

An operating system built on Java Scripit is Node OS. Its packages are managed by the package manager of Node.js which is npm hence any Node.js package is Node OS package. While packages in npm is about 300,000.NodeOSwon the Spanish 9th National Free Software Championship in the category of the Systems and is also honorably Mentioned in its 10th edition [43] [44].

iii. PoisonTap

Poison Tap is built on Node.js to create backdoors even the locked computers. This USB just requires to be plugged to a running computer even if it

is locked. It then installs continuous backdoors which operates even if the USB is removed later and allows the intruder to access also the router along with the target network and cookies of the web browser [45]. When we asked the creator Samy Kamkar through email that why he choose Node.js for Poison Tap, he replied: "The browser based code must be in JavaScript, so it made sense to me to keep the backend in the same language".

iv. *Node.js in IOT*

Node.js is adopted by developers and researchers for IOT for the reasons that JavaScript fast and is familiar among the large number of web developers as they use it with HTML5 for programming front end User Interfaces, and important point is that JavaScript is best for embedded devices as its nature is to support asynchronous and event-driven functions. Also the programming model adopted by Node.js is a great fit for embedded devices as well as servers and the domain experts have already adopted Node.js for the purpose of IOT [46].Microsoft have also adopted Node.js for IOT development in its developer resources [47].

III.   Materials and Methods

This study gains knowledge from the Node.js domain experts through intensive literature review. This study also gains knowledge from a surveys conducted from the professional developers. The survey was created on Google forms and ran for 1 month from the following link:

✓ https://docs.google.com/forms/d/e/1FAIpQLSc4Ghr oqubE5jQSnzmPOXEWuXzD8IcpRCtFPbSfFYscszX SVg/viewform?hl=en

The survey was targeted to the developers groups on the LinkedIn, Facebook and also shared with professional developers.
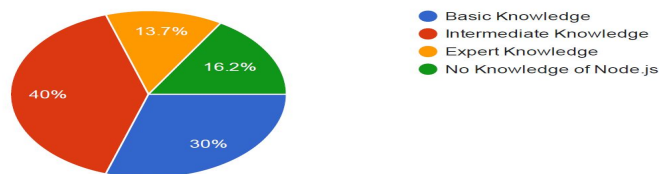
IV.   Results and Findings

The Literature Review concluded that Node.js can be useful and should be implemented in any place where there is processing of large files or requires large network load. Below are discussed some results in the same context. Due to Node.js a developer can easily become Full Stack Developer where he as a developer does not require separate colleagues for server side development and database development. Also the employers can reduce their cost by adopting Node.js as they will find a single developer taking care of performing all task at server side as well as at client side. Introduction of Node.js also introduced easy implementation Single Page Applications (SPA) where the web application developed in it are faster, as they use fewer resources of the server and fewer callbacks to the server while making the websites more interactive and user friendly[42]. The literature review also concluded that the field of Internet GIS have bright future in combination with Node.js. Also to mention, that Node.js is proven to be faster than other to process graphics processing. As there are benefits of the Node.js, there are large chances of misusing it and one such case is the release of PoisonTap USB which create backdoors to a computer and the network through any running computer even if it is password locked. Due to the increased use of JavaScript among the developers, such backdoor software are also part of life of developers.

As the study is also based on the quantitative research, because a survey was conducted, below are mentioned the results of the same. The online survey was sent to several Professionals through LinkedIn, Facebook, and Personal Networking. Total 93 responses were received. Among them 80 responses were useful for this study as they were developers and as this study is related to developers' community. Among the 80 developers (respondents) which were reached within a month, 16.2% did not know about Node.js. The rest 83.8% of the developers proceeded with the rest of the survey questions and they were 67 in count as shown in Figure 4.

Figure 4: Awareness of Node.js

Regarding learning challenge; the respondents shows that is somehow hard to learn the Node.js as only 31.3% said it wasn't a challenge for them to learn Node.js. 23.9% felt its learning as a challenge, 44.8% felt little bit the learning as a challenge as shown in Figure 5.

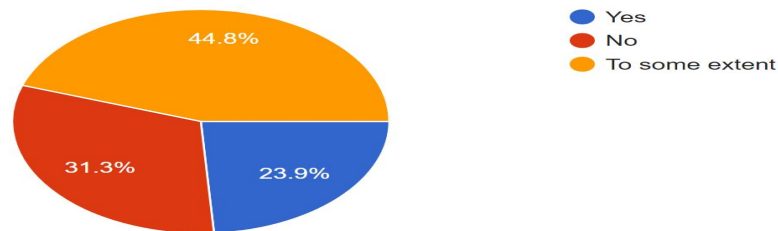### Learning of JavaScript for Node.js was a challenge?
67 responses

Legend:
- Yes
- No
- To some extent

44.8%
31.3%
23.9%

Figure 5: Learning JavaScript for Node.js

Regarding Database challenge; the survey concluded the results that as developers are familiar with the SQL databases, they find it difficult to adopt the NoSQL databases as quickly as only 31.3% said that the NoSQL databases use and learning is not a challenge as shown in Figure 6.

Learning and Using JavaScript Databases was a challenge?
67 responses

Legend:
- Yes
- No
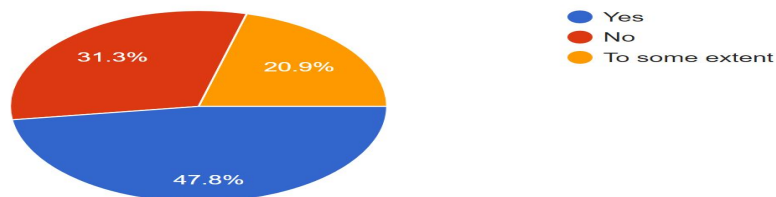- To some extent

31.3%
20.9%
47.8%

Figure 6: Use of NoSQL Databases

Regarding the Event-Driven feature challenge, the survey concluded that they felt difficulty with Event-Driven feature of Node.js as 40.3% did not considered it as a challenge, the rest 25.4% felt it a challenge to some extent only, and the rest 34.3% said they felt it as a challenge as shown in Figure 7.

Event Driven feature of Node.js was a challenge?
67 responses

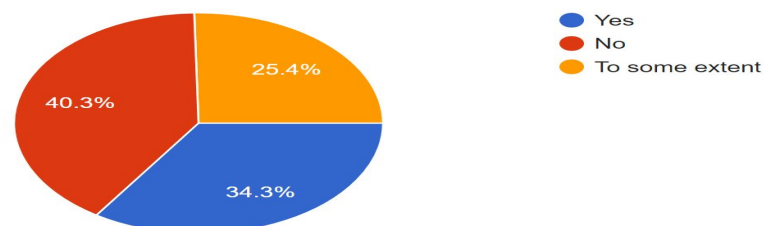Legend:
- Yes
- No
- To some extent

25.4%
40.3%
34.3%

Figure 7: Event Driven feature of Node.js

Regarding Non-blocking feature challenge, the survey concluded this feature is a challenge to the developers. As the Figure 8 shows, only 41.8% did not feel it as a challenge while 31.3% see this feature as a challenge, 26.9% see it as a challenge up to some extent only.

Non-Blocking I/O feature of Node.js was a challenge?

67 responses

- ● Yes
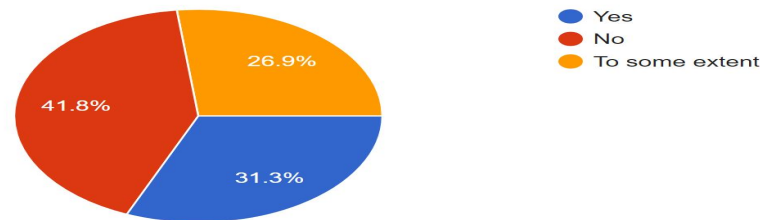- ● No
- ● To some extent

41.8% 26.9% 31.3%

Figure 8: Non-Blocking I/O in Node.js

Regarding asynchronous feature challenge, the survey concluded this is not enjoyed by the developers as according to Figure 9 only 43.3%feel the feature wonderful, 38.85 say that this feature is a challenge and 17.9% feel it as a challenge up to some extent only.

Asynchronous Processing feature of Node.js was a challenge?

67 responses

- ● Yes
- ● No
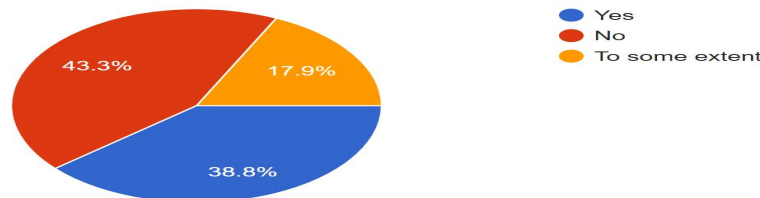- ● To some extent

43.3% 17.9% 38.8%

Figure 9: Asynchronous Processing in Node.js

Regarding the challenge of being familiar with other programming language; the survey concluded that the developers with hands on other languages also enjoys Node.js as according to the survey outcome shown in Figure 10 the respondents who are using other programming languages do not feel Node.js adoption as a challenge by 50.7% and rest of 17.9% from the other programming languages felt Node.js adoption as a challenge to some extent. While only 31.3% feel it as a challenge if they are from other programming language environment.

Hands on other environments (Like PHP, .Net, Ruby etc.) gave hard time to adapt Node.js?

67 responses

- ● Yes
- ● No
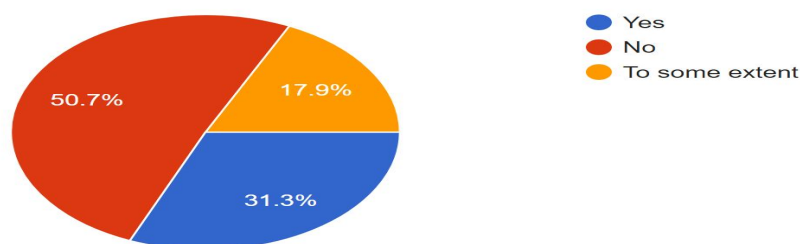- ● To some extent

50.7% 17.9% 31.3%

Figure 10: Developers from other programming languages

Regarding the challenge of configuring server for Node.js; 61.2% responded as they did not felt it as a challenge. Only 25.4% felt it as a challenge, while 13.4% felt setting up the server a challenge up to some extent only. The result of survey to this opinion is shown in Figure 11.

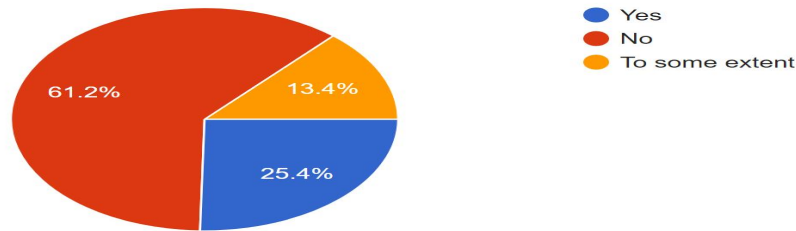## Setting up Hardware or Server was a Challenge

67 responses

Figure 11: Setting up hardware for Node.js

Regarding the challenge of organizational decisions to adopt Node.js; according the Figure 12 from survey results, 47.8% said the reason to delay the Node.js implantation is their organization's decision. 37.3% disagreed with it, while 14.9% respondents are not so sure about this question.

Organization decision to replace existing technology was a hindrance in Implementation of Node.js?
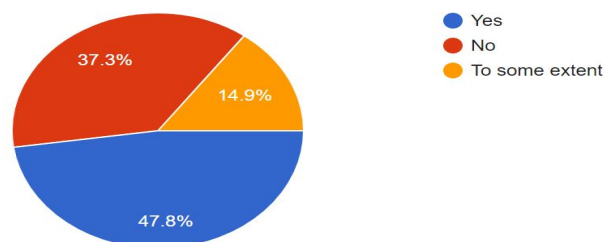
67 responses

Figure 12: Organization decision to replace the current technology

The challenge of market awareness is also a factor in implementing the Node.js. This is shown from the Figure 13 as when asked from the online survey respondents about the market awareness factor, 44.8% agreed to it as a challenge, 19.4% agreed it as a challenge up to some extent. While only 35.8% said it is not a challenge.

Lack of Market Awareness caused a hindrance in implementing Node.js?
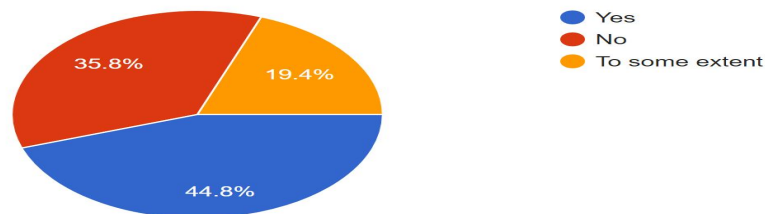
67 responses

Figure 13: Lack of market awareness as a challenge to implementing Node.js

As a result of the survey conducted, the greatly liked feature (by about 50% developers) of the Node.js is the reality that the same consistent language is used on both the servers and the clients. While the other features like event-driven, non-blocking feature, and the use of JavaScript is also liked by about 46% of developers. And some of them (which are 34.3%) also liked its ability to make itself best fit in IOT. This is shown in Figure 14
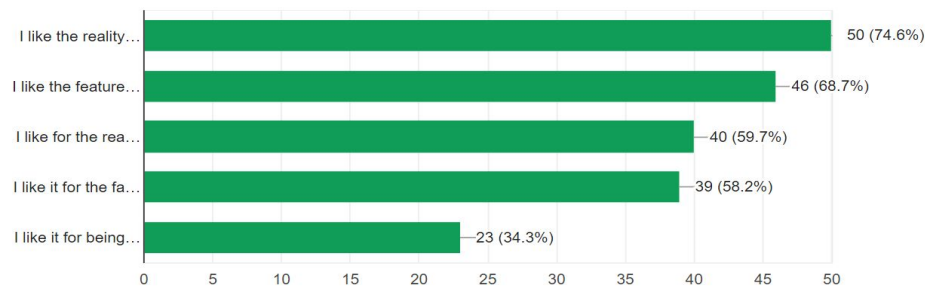
What Do You Like About Node.js?

67 responses

Figure 14: Greatly Liked Features of Node.js

V. Discussion and Future Work

The study got the finding about the implementation of Node.js. Below is discussed the implementations positive findings as a result of literature review and the survey.

✓ The Node.js have made Full Stack Developers' job a dream come true. In absence of Node.js it was hard for a developer to learn several different languages and environments to manage the complete system at server side and client side.

✓ Organizations and developers can now with the invent of Node.js build highly load bearable and faster applications and by using Single Page Applications (SPA) now the server calls are reduced and the applications are more user friendly and faster.

✓ Node.js made it easy to achieve high load operations like graphic processing and Internet GIS very quicker, and it can be reliably used in every field where the files sizes are high or the network bandwidth is highly consumed. Node.js will make such operations faster and with less need of bandwidth.

✓ Community like its feature that the same language is also being used at server side while JavaScript is always been used at client side for ages.

Node.js have some challenges in context of its use in the community as well as its adoption by the developers and organizations over the existing programming technologies. No doubt that Node.js have great benefits, it have also some challenges to the community. One such challenge is the ability of misuse of the widely used language by developers. One enthusiast have made a backdoor software using Node.js on Raspberry Zero. It can create backdoors in the target computer and their network even if the computer is password locked. Although there are solutions from such backdoors but some seems impractical like totally blocking the USB ports, and closing the web browser every time the user leaves the computer. And other options are not implemented by

majority and mostly might not be aware of it like using secure layer on ones websites (https), and enabling secure flags on the cookies which common users might not know about it [45].

✓ There is a plus point but as understood from the survey conducted that the community feel it hard to learn JavaScript for Node.js

✓ Also the developers having knowledge of other programming languages have complications in adopting Node.js. Even the setting up of server for their programming work is not an obstacle. This is as concluded from the survey results.

✓ Another plus points were event-Driven Programming, Non-Blocking I/O, and asynchronous feature. But according the survey results is that the features like event-driven programming, Non-Blocking I/O, and Asynchronous processing is a hindrance.

As a result of the survey, a challenge comes to front is that most organizations are not ready to adopt the new technologies like Node.js over their existing ones like PHP, .Net, etc. Also there is a lack of market awareness which is causing a barrier to adopt Node.js for implementation. At a developer level, there is a challenge which is seen from the survey results that they are not feeling it easy to learn the database working and using of the JavaScript environment. And there also seems a lack of enough knowledge among the developers as from the survey results a reasonable respondents to the questions on the important features like event-driven, non-blocking I/O and asynchronous processing is making the decision about this.

The community can be taken to the Node.js by little efforts of trainings, and conducting workshops to introduce Node.js to new and existing programmers and explain to them the benefits of Node.js features like event-driven programming, non-blocking I/O, and asynchronous processing. The community need to be updated about the features of how Node.js is faster in performing network related tasks specially when it come to the challenge of handling large files over the network and handling multiple callbacks in the other languages

81

like PHP, .NET, etc, causing the server to overload or require more memory. Also organization should be briefed about the benefits of hiring Full Stack Developers based on Node.js and how they will cut costs related to server bandwidths and developers hiring and speed of the applications built over Node.js

References Références Referencias

1. "The benefits of web-based applications," [Online]. Available: http://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-basedapplications.htm. [Accessed 25 November 2016].
2. Web Application Basics, Pearson Higher Education.
3. F. Bridge, "What Types of Developers Are There?," tree house, 24 June 2016. [Online]. Available: http://blog.teamtreehouse.com/what-types-of-developer-are-there. [Accessed 25 November 2016].
4. M. Wales, "Front-End vs Back-End vs Full Stack Developers," Udacity, 08 December 2014. [Online]. Available:http://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html. [Accessed 25 November 2016].
5. J. Long, "I Don't Speak Your Language: Frontend vs. Backend," tree house, 25 September 2012. [Online]. Available: http:// blog.teamtreehouse.com/ i-dont-speak-your-language-frontend-vs-backend. [Accessed 25 November 2016].
6. A. Mardan, "PHP vs. Node.js," Programming Weblog, [Online]. Available: http://webapplog.com/php-vs-node-js/. [Accessed 28 January 2016].
7. J. Kaplan-Moss, "Quora," [Online]. Available: https://www.quora.com/What-are-the-benefits-of-developing-in-Node-js-versus-Python. [Accessed 29 June 2016].
8. "Node.js Tutorial," tutorials point, [Online]. Available: https://www.tutorialspoint.com/nodejs/index.htm. [Accessed 25 November 2016].
9. N. Chhetri, "A Comparative Analysis of Node.js (Server-Side JavaScript)," Culminating Projects in Computer Science and Information Technology., p. 5, 2016.
10. R. R. McCune, "Node.js Paradigms and Benchmarks," 2011.
11. Node.js, "Home page of Node.js," Joyent, 2016. [Online].Available:https://nodejs.org/en/. [Accessed 01 May 2016].
12. G. Developers, "Chrome V8 | Google Developers," Google, [Online]. Available: https:// developers. google.com/v8/. [Accessed 27 May 2016].
13. eventmachine, "GitHub, Inc," [Online]. Available: https://github.com/eventmachine/eventmachine.[Accessed 30 June 2016].
14. Twisted Matrix Labs, "Twisted Matrix Labs," [Online]. Available: http://twistedmatrix.com/trac/. [Accessed 30 June 2016].
15. The Apache Software Foundation, "Apache MINA," [Online]. Available: http://mina.apache.org/. [Accessed 30 June 2016].
16. The Apache Software Foundation, "Apache MINA," [Online]. Available: http:// mina.apache.org/ async-web-project/index.html. [Accessed 30 June 2016].
17. Ryan Dahl: Original Node.js presentation. [Film]. Youtube, 2012.
18. "How Loading Time Effects Your Bottom Line," Kissmetrics Blog, [Online]. Available: https://blog. kissmetrics.com/loading-time/. [Accessed 25 November 2016].
19. C. Buckler, "Site Point Smack Down: PHP vs Node.js," Site Point, [Online]. Available: http://www. sitepoint.com/sitepoint-smackdown-php-vs-node-js/. [Accessed 28 January 2016].
20. Firehose, "Firehose," Firehose Project, [Online]. Available:http://blog.thefirehoseproject.com/posts/nodejs-vs-rails/. [Accessed 30 June 2016].
21. "Quora," [Online]. Available: https://www.quora. com/What-are-the-disadvantages-of-using-Node-js. [Accessed 30 June 2016].
22. "Hashnode," Hashnode, [Online]. Available: https://hashnode.com/post/what-are-the-actual-dis advantages-of-using-nodejs-ciibz8fd3017yj3xtxqz 1r9 hs. [Accessed 30 June 2016]
23. Flickr, "Flickr, A Yahoo Company," Yahoo, [Online]. Available: https://www.flickr.com. [Accessed 18 November 2016].
24. A Harris, "The Birth of Node: Where Did it Come From? Creator Ryan Dahl Shares the History," silicon ANGLE, 01 April 2016. [Online]. Available: http://siliconangle.com/blog/2013/04/01/the-birth-of-node-where-did-it-come-from-creator-ryan-dahl-shar es-the-history/. [Accessed 18 November 2016].
25. L. ORSINI, "What You Need To Know About Node.js," read write, 07 November 2013. [Online]. Available: http://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/. [Accessed 2016 November 2016].
26. G. Nemeth, "Rising Stack Engineering," [Online]. Available: https://blog.risingstack.com/history-of-node-js/. [Accessed 27 October 2016].
27. R. Dahl, "Porting Node to Windows With Microsoft's Help," Node.js, 23 June 2011. [Online]. Available: https://nodejs.org/en/blog/uncategorized/porting-node-to-windows-with-microsofts-help/. [Accessed 18 November 2016].
28. P. Krill, "Q&A: Why io.js decided to fork Node.js," JAVAWORLD, 04 December 2014. [Online]. Available:http://www.javaworld.com/article/2855639/open-source-tools/qanda-why-io-js-decided-to-fork-node-js.html. [Accessed 18 November 2016].
29. "Node.js Foundation Advances Community Collaboration, Announces New Members and Ratified Technical Governance," The Linux Foundation, 15 June 2015. [Online]. Available:

https://www.linuxfoundation.org/newsmedia/announ cements/2015/06/nodejs-foundation-advances community-collaboration-announcesnew.[Accessed 18 November 2016].

30. "Node.js Foundation Combines Node.js and io.js Into Single Codebase in New Release," Node.js, 14 September 2015. [Online]. Available: https://nodejs. org/en/blog/announcements/foundation-v4-announce/. [Accessed 18 November 2016].

31. A Bretz and C. J. Ihrig, Full Stack Javascript Development With Mean.

32. G. Ornbo, Sams Teach Yourself Node.js in 24 Hours.

33. Li, "Architecture of Node.js' Internal Codebase," 04 June 2016. [Online]. Available: https://arenli. com/architecture-of-node-js-internal-codebase-57c d8376b71f#.koh166uay. [Accessed 18 November 2016].

34. J. Harrell, "Node.js at PayPal," PayPal Engineering, 22 November 2013. [Online]. Available: https://www.paypal-engineering.com/2013/11/22/ node- js-at-paypal/. [Accessed 18 November 2016].

35. "How did Microsoft adopt Node.js as part of its developer stack?," Quora, [Online]. Available: https://www.quora.com/How-did-Microsoft-adopt-Node -js-as-part-of-its-developer-stack. [Accessed 18 November 2016].

36. J. O'DELL, "Why Walmart is using Node.js," VentureBeat, 24 January 2012. [Online]. Available: http://venturebeat.com/2012/01/24/why-walmart-is-using-node-js/. [Accessed 25 November 2016].

37. "GitHub Language Statistics," GitHub, November 2016. [Online]. Available: https://github.com/ emmanuel-keller/github-language-statistics [Accessed 10 December 2016].

38. Y. Pandji D. and W. S. Raharjo, "Performance and Scalability analysis of Node.js and PHP/Nginx Web Application," INFORMATIKA, vol. 9, no. 2, pp. 117-124, 2013.

39. K. Fysarakis and D. Mylonakis, "Node.DPWS – High performance & scalable Web Services for the IoT," [Online]. Available: https://arxiv.org/ftp/arxiv/papers/ 1503/1503.01398.pdf. [Accessed 21 May 2016].

40. [S. H. Jun and K. T. Doh, "Design and Implementation of Web GIS Server Using Node.js," Journal of Korea Spatial Information Society, vol. 21, no. 3, pp. 45-53.

41. "MAGI: a Node.js web service for fast microRNA-Seq analysis in a GPU infrastructure," BIOINFORMATICS APPLICATIONS NOTE, vol. 30, no. 19, pp. 2826-2827, 6 June 2014.

42. M. S. Mikowski and J. C. Powell, Single Page Web Applications, MANNING PUBLICATIONS.

43. J. L. Combarro, "NodeOS: Light weight operating system using Node.js as userspace," [Online]. Available : https://github.com/nodeos/nodeos. [Accessed 16 September 2016].

44. npm, "Official Website of npm," npm, [Online]. Available: https://www.npmjs.com/. [Accessed 16 September 2016].

45. S. Kamkar, "Samy Kamkar," November 2016. [Online]. Available: https://samy.pl/. [Accessed 10 December 2016].

46. M. McCool, R. Peri and R. S. John, "Programming the Internet of Things Using Node.js and HTML 5," in O'Reilly Solid Conference, June 23, 2015.

47. Microsoft, "Developer resources," Microsoft, [Online]. Available: https://developer.microsoft.com/ enus/windows/iot/samples/expressnodejs.[Accesse d 10 December 2016].