Role of Node.js in Modern Web Application Development

Salil Bhatnagar[1] and Ashish Pandey[2]

[1] Chameli Devi Group of Institutions, Indore
Salilbhatnagar1@gmail.com
[2]Chameli Devi Group of Institutions, Indore
Ashishpandey0731@gmail.com

Abstract:-

Node.js is one of the hottest open-source web platforms currently available. It's built on Google Chrome's V8 JavaScript runtime engine and it allows you to write all kinds of network applications and servers in just a few lines of code. Node.js uses an asynchronous programming model built on non-blocking I/O and a single-threaded event loop. Basically, you don't need to be concerned with awful race conditions or synchronization issues that arise when programming for a concurrent multi-user environment. This workshop is aimed mainly at web development instructors who would like to teach a single programming language, JavaScript, for both client-side and server-side coding. Participants will learn how to employ Node.js on Windows, Mac OS, or Linux to write scalable web servers and applications. Additionally, the Express web framework will be introduced to demonstrate how to quickly program traditional web apps and single-page applications (SPA) with the aid of jQuery, AJAX, and RESTful web services. The resulting programs will be usable from any modern web browser, including those found in desktop and laptop computers, and mobile devices such as tablets and smartphones. Participants should have prior working knowledge of client-side (running on a browser) JavaScript and HTML. More information: http://node.arielortiz.info/

Key Words: **JavaScript, Node.js, event driven, single- threaded, non-blocking, asynchronous**

1. Introduction:

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a powerful tool suitable for a wide range of projects. Node.js stands out as a game-changer. Imagine using the power of JavaScript not only in your browser but also on the server side.
Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all your users to update their browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version, and you can also enable specific experimental features by running Node.js with flags.

- **An Example Node.js Application**

  The most common example Hello World of Node.js is a web server:

```
const { createServer } = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

  To run this snippet, save it as a server.js file and run node server.js in your terminal.

  This code first includes the Node.js http module.

  Node.js has a fantastic standard library, including first-class support for networking.

  The createServer() method of http creates a new HTTP server and returns it.

  The server is set to listen on the specified port and host name. When the server is ready, the callback function is called, in this case informing us that the server is running.

  Whenever a new request is received, the request event is called, providing two objects: a request (an http.IncomingMessage object) and a response (an http.ServerResponse object).

  Those 2 objects are essential to handle the HTTP call.

  The first provides the request details. In this simple example, this is not used, but you could access the request headers and request data.

  The second is used to return data to the caller.

  In this case with:

  res.statusCode = 200;

  we set the statusCode property to 200, to indicate a successful response.

  We set the Content-Type header:

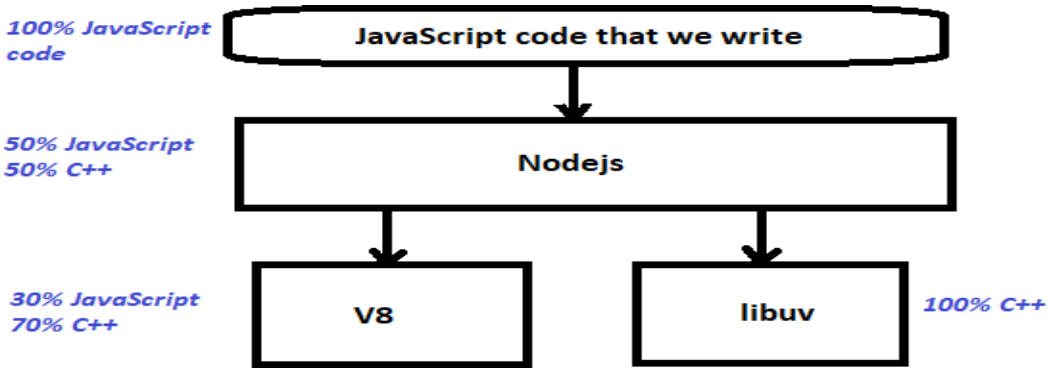
  res.setHeader('Content-Type', 'text/plain');

  and we close the response, adding the content as an argument to end():

  res.end('Hello World\n'); [1]

2. Node.js InternalStructure:

- **V8**: It is an open-source project created by Google. The purposeofthisopen-sourceprojectistoexecute JavaScript code outside the browser. It gives access to node underlying networking and helps the Node to handle aspects of concurrency. (We know Node.js is all about concurrency). 70% of the code written in



this project is in C++ and theother 30% are written in JavaScript.

Fig-1:Node.jsinternalstructure-I

- **libuv**: It is an abstraction layer on the top of c-ares (for DNS) , iocp (for windows asynchronous-io) and libeio , libev. LibUv maintains and manages all the Input-Output and events in the event pool. If I have to put in simple words thenI would say libuv allow yourJavaScriptcodeto perform I/O operation whether its networking or file operation etc. All the file/system operation and TCP level connectivity are performed by this library. This library is completely written in C++.

  As JavaScript developer, you will be writing your projectinJavaScriptandwantittocompileandperformits action. So you can see Node.js is just acting as an interface between the JavaScript code that we write and other open source code (V8 and libuv) that is written in either JavaScript or in C++. For JavaScript developer, they don't interact with C/C++ code directly. So with this nice interface to use and to relate the JavaScript to actual C++ code running in our computer to compile and execute our JavaScriptcode.NodealsoprovidesaunifiedAPI'stousein our JavaScript project. Give a look in Fig-2

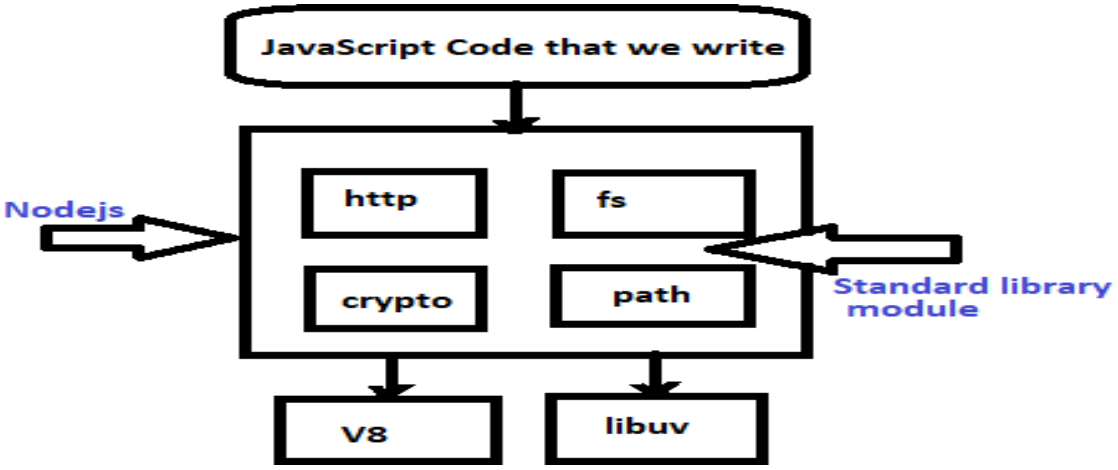


Fig-2:Node.jsinternalstructure-II

The library module in Node.js like fs, http, path, crypto etc. are very consistent API's and they all ultimately refer to a functionalitythataremostlyliveinsidethelibuvprojectso that you do not have access directly C++ code written in it. You just use JavaScript functions and that functions calls the Node.js libuv project. You do not have to worry about its internal functionality written in C++ which libuv handles by itself.

3.ModuleSystem:

JavaScript does not represent an API for module dependency and isolation in its specification. As a result, including multiple modules isonly possible bydisclosing a global variable. For example, the jQuery module can be included in a HTML document by including this line at the head tag (script src=https://code.jquery.com/jquery- 1.6.1.js) Then, refer to this module through the global jQuery object. This process pollutes the global namespace and can result in potential naming collisions.

Instead of defining a number of globals, Node has introduced a modular system. One can define their own module or can use the core modules or third party modules. Node.js Modules are plugins, add-ons, and extensionsforNodetohelp withthedevelopmentprocess. The Node module discloses a public API (Application Programming Interface) that one can use after the module is imported into the current script. Node modules can be categorizedaslocalmodules,coremodulesandthirdparty modules [7].

➢ NPM-The Node Package Manage

Node.js has built-in support for package managementusingNPM,a tool thatcomesbydefault witheveryNode.js installation. Theidea ofNPMmodules issortofalmostlike that of Ruby Gems: a group of publicly available, reusable components, available through easy installation via a web repository,withversionanddependencymanagement. A full list of packaged modules are often found on the npm website, or accessed using the npm CLI tool that automatically gets installed with Node.js. The module ecosystem is hospitable all and anyone can publish their own module which will be listed within the npm repository.

Someoftheforemostusefulnpmmodulestodayare:

- **express** - Express.js is a Sinatra-inspired web development framework for Node.js, and the de- facto standard for the majority of Node.js applications.
- **connect** - Connect is an extensible HTTP server framework forNode.js, It is providing a collection of high performance "plugins" known as middleware;
- **socket.io** and **sockjs** – A server-side componentof the two most common web socketscomponents out there today.
- **mongodb** and **mongojs** – To provide the API for MongoDB object databases in Node.js.
- **bluebird** - A fully featured Promises/A+ implementationwithspeciallygoodperformance.
- **moment** - A JavaScript date library for validating, parsing, manipulating, and formatting dates.
  The list goes on. There are plenty of really useful packages out there, available to all or any (no offense to people who I've omitted here) [6].

4.KeyFeaturesofNode.js

➢ Non-blocking I/O:

The I/O methods within the Node.js standard library provide asynchronous versions, which are non-blocking, and accept callback functions. Some methods even have blocking counterparts, which have names that end with Sync Example

```
constfs=require('fs');

constcontent=fs.readFileSync('/file.txt');// blocks here until file is read

console.log(content);

moreWork();//willrunafterconsole.log
```

- **Asynchronousexample**:

```
const fs = require('fs'); fs.readFile('/file.txt,(err,content)=>{ if (err) throw

err;console.log(content);

});

moreWork();//willrunbeforeconsole.log
```

In the first example above, console.log are going to be called before moreWork(). In the second example fs.readFile() is non-blocking so JavaScript execution can continue and moreWork() are going to be called first. The ability to run moreWork() without expecting the file read to finish may be a key design choice that permits forhigher throughput [1].

- **SingleThreadedEventLoop**

Node.js Platform doesn't follow Request/Response Multi- Threaded Stateless Model. It follows Single Threaded with Event Loop Model. Node.js Processing model mainly supported JavaScript Event based model with JavaScript call-back mechanism.

As Node.js follows this architecture, it can handle multiple concurrent client requests very easily.

The“Event Loop”is heart of Node.js Processing model.

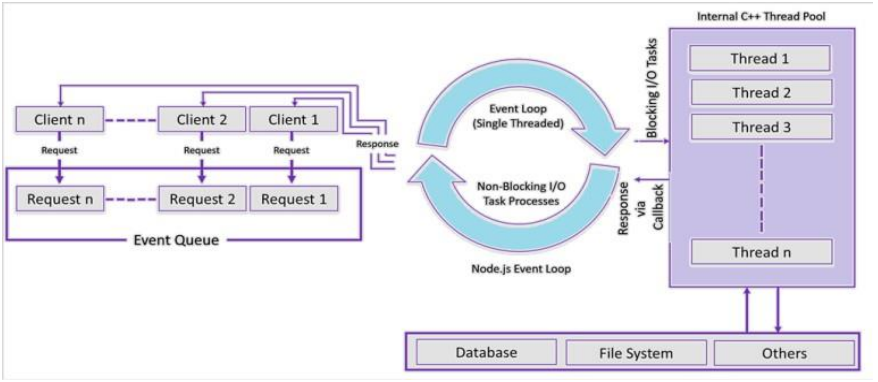


Fig-3:Node.jsApplication/Server

- **<u>Single Threaded Event Loop Model Processing Steps</u>**:

- UserSendrequesttotheServer.
- Node.js Web Server inside keeps upa Limited Thread pooltooffertypesofassistancetotheClientRequests.
- Node.js Web Server gets those requests and spots them into a Queue. It is known as "Event Queue".
- Node.js Web Server inside has a Component, known as "Event Loop”. It got this name because it uses infinite loop to receive requests and process them. (Pseudo code to comprehend this underneath).

```
publicclassEventLoop{

while(true){

if(EventQueuereceivesaJavaScriptFunction Call){

ClientRequest                    request  = EventQueue.getClientRequest();
If(requestrequiresBlokingIOortakesmore computationtime)
AssignrequesttoThreadT1 Else
```

ProcessandPrepareresponse } } }

- Event Loop utilizes Single Thread as it were. It is principle heart of Node.js Platform Processing Model.

- Even Loop checks any Client Request is put in Event Queue.Assumingno,athenwaitforincomingrequests forindefinitely.
- If yes, at that point get one Client Request from Event Queue
- StartsprocessthatClientRequest

- If that Client Request Does Not requires any Blocking IO Operations, at that point procedure everything, plan reaction and send it back to Client.

- If that Client Request requires some Blocking IO Operations like connecting with Database, File System, External Services then it will follow diverse methodology
- Checks Threads accessibility from Internal Thread Pool
- Picks up one Thread and allot this Client Request to that string.
- That Thread is answerable for taking that demand, processit,performBlockingIOtasks,getreadyreaction and send it back to the Event Loop

- Event Loop thusly, sends that Response to the respective Client [5].

5.Reasons for why Node.js used widely by Modern Web Developers:

➢ GoogleV8JavaScriptEngine

Node.js executes JavaScript code using Google V8 engine. Unlike other JavaScript interpreters, V8 engine compiles the JavaScript code into native machine language. Thus, it enables the runtime environment to boost the performance of web server applications by executing JavaScript code during a faster and more efficient way [1].

➢ Asynchronous I/OOperations

Node.js further performs all I/O operations asynchronously through a single-threaded event loop. The advanced methodology makes the Node.js application to perform I/O operations by sending asynchronous task to an event loop alongside a call-back function. After sending the async task to the event loop, the application continues executing the remaining code. After completing the asynchronousoperation,theeventloopreturnstothetask, and executes the call-back function. In addition to consuminglessmemory,thefeatureenablesNode.jsto handle a large number of concurrent connections efficiently. The programmers can use the runtime environment to perform common tasks like file system or network connections, reading or writing to the database.

➢ RobustTooling

WhileusingNode.js,thedeveloperscantakeadvantageofa dependable package manager like npm. In addition to being fast, consistent and robust, npm also simplifies the method of specifying and installing project dependencies. At an equivalent time, it keeps the project dependencies separatetoeliminatechancesofversionconflict.Theusers can further take advantage of the strong file streams capabilities of JavaScript using tools like broccoli, gulpand brunch, while using a popular task runner like grunt.

➢ Complement Real-Time and Multi-Use Requirements

Inaddition tousingresponsivewebdesign,thedevelopers nowadays need to build real-time and multi-user web applications. Node.js enables programmers to make complexgaming,chattingandcommunicationapplications without putting any overtime and energy. The

developers can use websocket protocols to create real-time web applications. As a two-way channel between the client and web server, websockets can make the online server push data to the client in a very faster and more efficient way without increasingtheoverheadofHTTP.Atanequivalent time, the developers can create multi-user applications by taking advantage of the event loop feature of Node.js [7].

- **Facilitates FileStreaming**

The web programmers can take advantage of the efficient I/O handling capacity to Node.js to reduce the amount of time required for streaming a file from the file system. They can use the runtime to read/write streams to both HTTP and websockets. Thus, they have can reduce the overall transcode audio or video processing time in a number of ways. For instance, a programmer can simply shift the stdout from the web server to a web browser through websockets, and allow the web page to display output to the user in real time [7].

- **Popularityof JavaScript**

Since early in the evolution of WWW, JavaScript has been there in the browser. Even available when AJAX emerged, JavaScript was vital. This has led to the popularity of JavaScript among developers, despite some criticism. No matter which server-side scripting language is used, JavaScript has been the choice for client-side scripting. Familiarity with JavaScript and adherence of Node to JavaScript, with capabilities to code in the server-side and numerousotherfeatureshasdeveloperstoadoptNode.By leveragingthebestfeaturesofJavaScriptasalanguageand
nurturing a vibrant community, Node has become a popularplatformandframework,withcontinuedadoption growth

- **BuildandMaintainaSingleCodeBase**

Node.js enables developers to writedown both client-side and server-side code in JavaScript. So a programmer can useJavaScriptforbuildingbothfrontendandbackendofan internet application. ThusNode.js eliminates the gap between frontend and backend development [8].

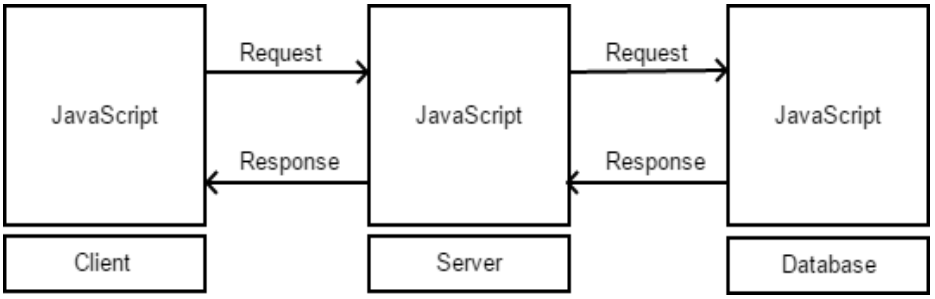


Fig-4:JavaScriptend-to-end

6. Whatreasonarethebiggestcompaniesusing Node.jsfor?:

- **PayPal**

PayPal is one among the most important and most well- known worldwide systems for Internet payments. The platform enables its users to transact with one another online quickly and simply in additional than 100 currencies.
According to Payal, the service has over 184 million active customer accounts (as of 2015). The Company has been usingNode.js to createthe consumer-facing sideof its web applications.

Why did they choose Node.js? Senior Director 'Jeff Harrel' of Payments Products and Engineering at PayPal says: "Node.jshelpsussolveboundarybetweenthebrowserand server by enabling both the browser and server applications to be coded in JavaScript. It brings together our building claims to fame into one group which permits us to know and respond to our clients' needs at any level inside the technology stack".
Results:
TheNode.jsapplicationwasbuilttwiceassnappywithless people, in 33% less lines of code and 40% fewer files (in connection with past Java-based application) [3].

- **LinkedIn**

LinkedIn is a business-situated person to person communication administration found in 2002 in Mountain View,California.LinkedInpermitsclientstowelcome anybody(regardlessofwhetherasiteclientornot)toturn into an association.

Itisaccessible in24 dialectsandasof nowhas inexcessof 400 million individuals in more than 200 nations and regions.LinkedInutilizesNode.js toengagetheserverside of its mobile application.

ForwhatreasondidtheypickNode.js?
As indicated by LinkedIn's Mobile Development Lead, KiranPrasad"Oneexplanationwasscale.Thesecondis,on the off chance that you take a gander at Node, the thing its best at doing is conversing with different services."

Results:
When contrasted and the past Ruby on Rails-based adaptation, the new portable application is up to multiple times quicker and utilizes just a small amount of assets – servers were sliced from 30 to 3. The advancement itself was uncommonly quick [3].

- **Yahoo**

Yahoo is an American worldwide technology organization; centering and all around perceived for its web portal, search engine, and related services. As per Yahoo, service attractsmorethan500millionconsumers inmorethan30 languages every month. For what reason did they pick Node.js? Eric Ferraiuolo, Principal Software Engineer at Yahooclarifies:"Itcanbescalable,andeverypropertythat we've moved toward the Node.js stack has seen an expansion in execution" Results: Node.js has changed and bound together the frontend designer culture at Yahoo, it as of now controls numerous locales [3].

- **Netflix**

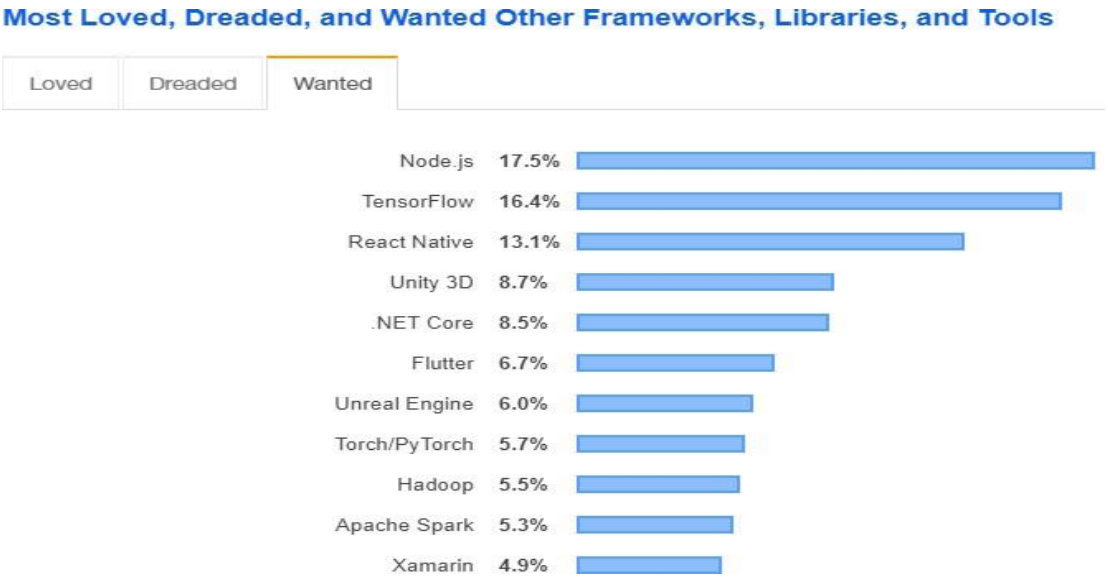
Netflix is the world's greatest worldwide supplier of video streaming including movies and TV series, accessible in more than 190 nations. As of April 2016, Netflix detailed more than 81 million endorsers around the world, with in excessof46millionintheU.Sitself.AtNetflix,theentireUI is worked with Node. The innovation demonstrated so viable, that the organization needs to utilize it likewise in different layers of the stack. For what reason did they pick Node.js? The group chose to utilize Node.js to accomplish lightweight, modular and fast application. Therefore, the start-uptimeoftheirnewapplicationhasbeendiminished by 70% [3].

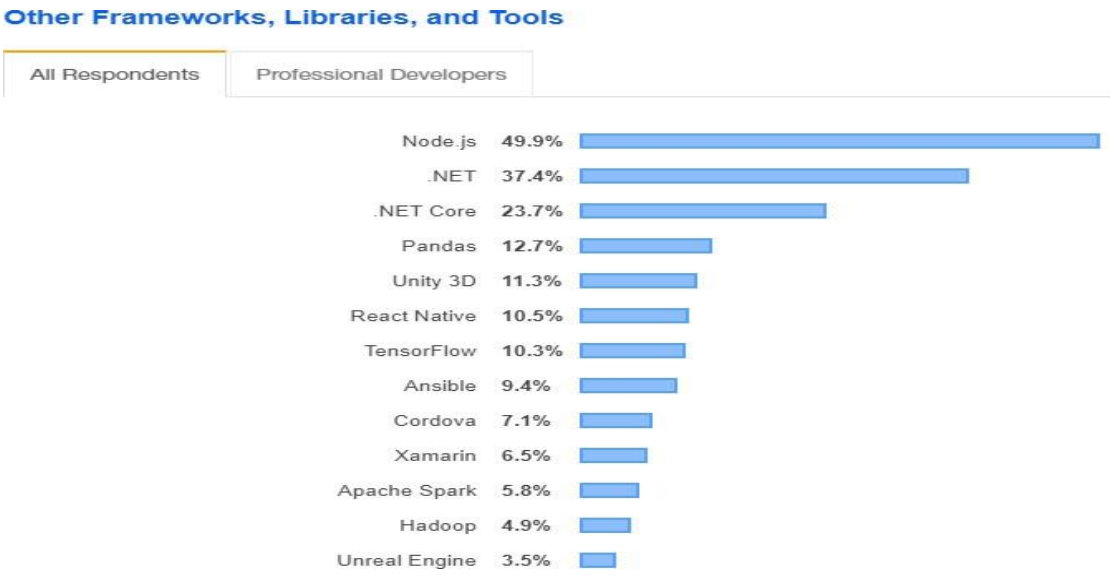
- **GoDaddy**

GoDaddy is a traded on an open market Internet space enlistment center and web hosting organization. As of January 2016, with in excess of 13 million clients and 61 million domains under management,

GoDaddy makes the world'sbiggestrecorder.Asoflateorganizationhas patchedupitswholebackendtoacompletelyopen-source Node.js-based framework. For what reason did they pick Node.js? As indicated by Stephen Commisso, Senior Software Developer at GoDaddy, Node enables to build quality applications, deploy new features immediately, writeunitandintegrationtestseasily,RESTeasily.Another key feature is NPM access. Results: GoDaddy's Vice President of Engineering, Antonio Silveira states: "we are currentlyutilizingabout10xlessserverstohaveourclient sites and we decreased the Time To First Byte (TTFB) significantly from ~60ms to something around ~12ms. Execution is a key differentiator when we talk about web applications, rememberinga superiorsituationfor Google's query items." GoDaddy is currently ready to deal with a similar burden with just 10% of the hardware [3].

According to Stack Overflow survey 2019 the Node.jsis the most commonly used and most wanted technology [4] Give a look in Fig-5 and Fig-6.





- **PotentialapplicationareasofNode.js**
  - Media
  - Paymentgateways
  - Ecommerce
  - Socialmedia
  - Enterprisewebapps
  - Backend/APIformobileapps

7.CONCLUSION

Node has transformed the usability of JavaScript, making Node a complete programming language. From browsers to server-side scripting outside of browsers, Node has made possible the availability of a runtime environment, a library full of free useful modules that can be imported by using an in-built tool named NPM. Node.js uses event- drivenI/O,non-blockingasynchronousprogrammingtobe lightweight and be efficient. Essentially, any business utilizing Node can: utilize fewer servers, utilize less engineers and abatement page load times.

- **REFERENCES**

[1] https://Node.js.org/en/docs

[2] Node.jsinActionbyMikeCantelon,MarcHarter,T.J. Holowaychuk, Nathan Rajlich.

[3] https://brainhub.eu/blog/9-famous-apps-using-node-js

[4] https://insights.stackoverflow.com/survey/2019

[5] https://www.journaldev.com/7462/node-js-architecture-single-threaded-event-loop

[6] https://www.toptal.com/nodejs/why-the-hell-would- i-use-node-js

[7] https://nodejs.org/api

[8] AComparativeAnalysisofNode.js(Server-SideJavaScript) Nimesh Chhetri.