International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Role of Node.js in Full Stack Development

ABHAY SINGH BHANDARI[1], *ABBAS ALI BOHRA*[1], *VAISHALI KUMARI*[1] *RITIK GOYAL*[1], *Dr. VISHAL SHRIVASTAVA*[2], *Dr. AKHIL PANDEY*[3]

[1]B.TECH. Scholar, [2,3]Professor
Computer Science & Engineering
Arya College of Engineering & I.T. India, Jaipur
abhaysinghbhandari09@gmail.com, abohra738@gmail.com,
kumarivaishali562@gmail.com, ritik9428@gmail.com, [2]vishalshrivastava.cs@aryacollege.in,[3]akhil@aryacollege.in,

ABSTRACT:

Node.js is a runtime for the JavaScript programming language. Its event-driven, non-blocking, and asynchronous approach make it a fast and reliable choice for decision-intensive applications with heavy workloads. The aim of this section is to focus on the modularization of Node, which is also known as Node Package Manager. Moreover, how Node works will be explained in this paragraph. The key feature of Node.js that makes such behavior possible is its asynchronous and non-blocking use of event driven I/O for deep efficiency against concurrent requests. We can develop immediately with Node.js applications that can scale to millions of connected clients.

It also explains why developers should choose Node.js and use it. This article looks at why developers should prefer to use Node.js. Key features of Node.js - I/0 event-driven, single-threaded and asynchronous.To better understand what Node.js means by examples programming, for creating a node is discussed as a whole in architecture.

Keywords: Node.js, Javascript, Event Driven, Single threaded, asynchronous, non-blocking.

INTRODUCTION :

There is a Javascript framework called Node.js that operates across many platforms and is powered by the Chrome V8 Javascript engine. Ryan Dahl invented a runtime environment for building server-side applications in 2009. Creating a node can be fast and easy if you know how to combine the correct things. V8 and Node are mainly written in C and C++ with emphasis on low memory usage and performance. JavaScript can also be used on the server side, especially in light scenarios. On the platform, a problem necessitated an answer.
Communication time efficiency of the network is an important factor to consider. It is necessary to optimize communication time across the network for its efficiency. This may involve reducing latency, increasing bandwidth and network capacity or minimizing congestion among others. Generally, network's communication time is very instrumental when it comes to determining how efficient the whole system can be.
They spend much of their time creating web requests. JavaScript has become so powerful that it is now central to server-side scripting.
Answering such questions as these is where Node.js comes into play; here, we see Java script operating end-to-end from the web browser down to the server.

Node.js Internal Structure

V8: Open source initiative was launched by Google. To execute JavaScript, the purpose of this project is to use open source software.The main idea behind Node.js, of course, is that it supports concurrency, gives back to the underlying network and helps in many administrative tasks. 70% of its code is written in C++, while the remaining 30% are in JavaScript.

libuv: libuv, an abstraction layer over libev, has three sub-components:
c-ares (for DNS), iocp (for Windows Async io) and libeio. An event controls all inputs, outputs and events in libUv. To be concise, I can say that libuv is feature rich whether it is networking or task processing. Somehow, I think JavaScript allows I/O operations on code. This library manages all data/transactions and connections at TCP level.

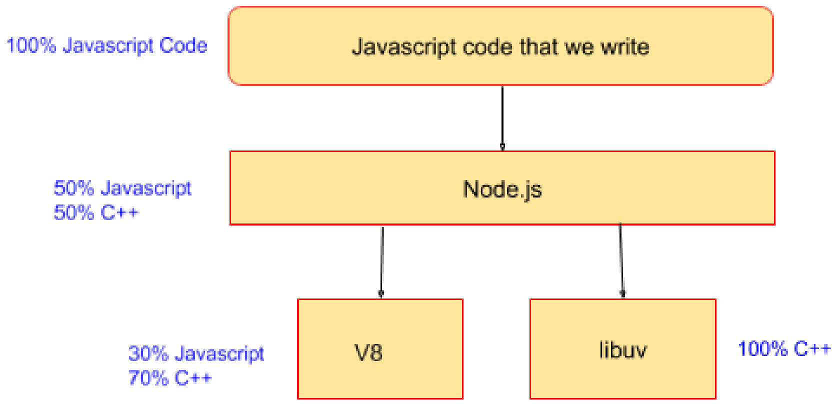

Fig-1: Node.js internal Structure-I

This library the language is written completely in C++.

A JavaScript program that you are writing in JavaScript and need to compile it as well as run it. Yes, that's correct! Node.js serves as a middleman between our own JavaScript code and other open-source code also written in JavaScript or C++ (V8 and libuv). In addition to accessing C/C++ directly from a JavaScript environment, Use javascript rather than pretty interface which will be linked to real C++. The software running on our computers for coding and executing JavaScript codes. Node also helps us with API integration that our JavaScript projects need.

Library modules such as fs, http, path, crypto in Node.js are I used similar APIs. You can with the help of common functions included in the libuv project; it is not possible to write this without access to C++ code.

It's just a matter of using a JavaScript project known as Node.js libuv project. you are expected to refrain from employing. In charge of internal functions written in C++, libuv. See figure 2.

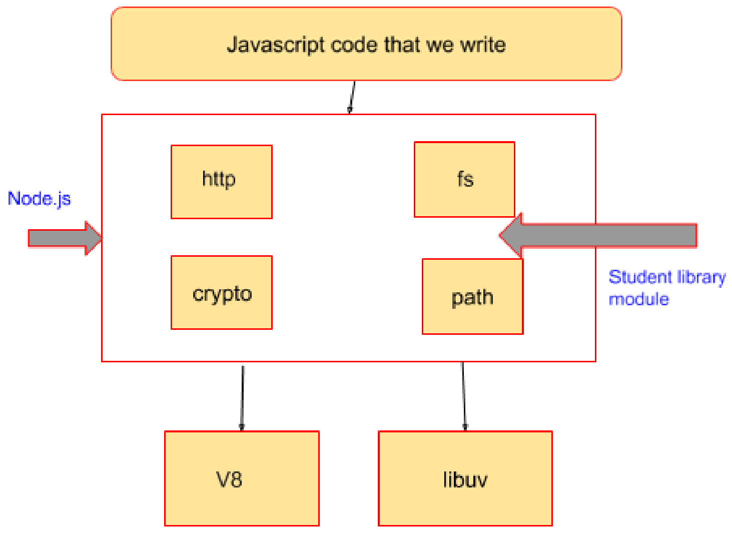

Fig-2: Node.js internal Structure-II

Module System

Despite the absence of its module dependency and isolated APIs representation in its directives, JavaScript only allows including multiple modules through exposure of global variables. This line is an example of how you would add a jQuery module to an HTML file.

Tag header (script src=https://code.jquery.com/jquery1.6.1.js). After that, read the article to transparently access this global jQuery object module. This makes names in the world look terrible and may lead to naming conflicts. But many global variables are not defined by Node. This represents a modular system where individuals can set their own standards or use official or third party standards. These are plugins, add-ons and extensions used for developing Node.js applications. Node modules have a public application programming interface (API) that users can use after installing the module into the given script. He divided node modules into regular modules, basic modules and other modules.

3.1 NPM- The Node Package Manager

Node.js has an inbuilt capacity for package management using the NPM tools that are provided with any Node.js installation. The idea behind NPM is similar to Ruby Gems: a collection of publicly available, easily updated components that can be readily installed into a web repository and combined with controls and functionality. You can also find information about most module packages on npm website.
Alternatively, use the npm CLI tool that comes with Node.js. The mod ecosystem wants you, yes you, and your mods listed on the npm repository. Some of the best NPM modules today are:
express – Express.js is Sinatra's Node.js web development framework and acts as the foundation for most well-known practices in Node.js part.
Connection – Connection is an HTTP server extension framework for Node.js that gives powerful "plugins".
socket.io and sockjs – the two most important server components network socket This component now exists.
mongodb and mongojs - Provides an API for MongoDB database objects in Node.js.
bluebird - Fully functional Promises/A+ application with excellent performance.
moment - JavaScript data library for validating, parsing, manipulating and formatting data.

Key features of Node.js

Non-blocking I/O:

Synchronous example

```
const fs = require('fs');
const content = fs.readFileSync('/file.txt); // blocks here until file is read console.log(content);
moreWork(); // will run after console.log
```

Asynchronous example:

```
const fs = require('fs'); fs.readFile('/file.txt, (err, content) => { if (err) throw err; console.log(content);
});
moreWork(); // will run before console.log
```

In the first example above, MoreWork() is called before console.log. In the second example, fs.readFile() does not block, so JavaScript execution can continue and moreWork() is called first.
Allowing multiple () functions to be executed without having to read the entire file may be an important option for better understanding.

Single Threaded Event Loop:

The Node.js platform does not follow a multi-threaded request/response format. It follows the single-threaded event loop model.
Generally, Node.js has models that support an event-based JavaScript model and have a JavaScript callback mechanism.
Typically, if this is how Node.js works, it can do so much more than merely accepting customers' requests. <br>The "event loop" is what the Node.js model hinges on.
Single-threaded event loop model entails following steps:

- User sends request to server.
- Unlimited number of threads for Node.js web server
- These are internal resources that can be used to offer such services to applicants.
- Accepts these requests by the Node.js web server and They are lined up there. This is known as "event queue."
- On a Node.js web server,

There is an option called "News Carousel." It gets its name from the fact that it uses an infinite loop to receive and fulfill requests.
(Using pseudocode below). public class EventLoop { while(true){
if(Event Queue receives a JavaScript Function Call){ ClientRequest request = EventQueue.getClientRequest(); If(request requires BlokingIO or takes more computation time) Assign request to Thread T1
Else Process and Prepare response
}
}
}

- A thread is actually used by the state loop. It is the basic model upon which the Node.js platform was built.
- Even Loop checks if the requestor has been placed in status queue. When no, you will wait forever to receive the request
- .
- If yes, get the applicant immediately from the status bar.

- Start with application
- In this way, when developing and sending a response back to a client, a requester avoids blocking I/O operations.
- When it is necessary for client to block some I/O
- Database connection, archiving and other operations systems, external services etc., — then let's go through difference with this one leave
- <thread. br>Pool checks thread availability,
- Select a thread and assign a requester to it, this string.
- This thread accepts requests, Their processing, execution of IO blocks, preparation of responses and return it to the status loop
- So that way event loop asks client for response.

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

Google V8 Javascript Engine:

Google V8 is used by Node.js to run JavaScript code. Java script is compiled into native machine code by the V8 engine, while other JavaScript interpreters do not. This is the way it operates Improve uptime Optimize web server application performance JavaScript code is faster and more efficient in comparison to others.

Asynchronous I/O operation:

As such, all I/O operations are performed by Node.js too. In a
single-threaded event loop asynchronously. This Advanced methodologies have facilitated the use of Web applications in Node.js. Perform I/O operations by submitting asynchronous tasks Event loop next to callback function. After submission When you insert an asynchronous task into the event loop, the application continues. Continue executing other parts of the code upon completion Returning to task, it comes back as soon as possible in case of asynchronous operation. and execute associated callback functions. also Node.js does this because it uses less memory than others. Manage multiple connections Effectively. The runtime environment allows programmers to perform common tasks like data or network connections or writing files.

Robust Tooling:

When working with Node.js, developers can use reliable package managers such as npm. Npm is also fast, consistent and powerful and it can also be used to share and organize your progress in an easy way. can last for a while. Is independent of project dependencies to avoid variable conflicts. Users may leverage JavaScript streaming capabilities through tools like Broccoli, Sip, Brunch etc. popular operators like Grunt are available.

Beyond real-time and multi-threading:

Developers have to design live and multi-user websites in addition to web design programme. Use of Node.js enables programmers to work on complex games, conversations and chats without any extra time or energy investment. Developers can create real-time applications using Websocket protocol. Websocket is a two-way communication between Customer one Web server allowing online servers transmit. This data is Faster as well as more effective without increasing HTTP load at the same time ,developers can use the Node.js event loop functionality to build multi-user applications.

Popularity of Javascript:

Since the World Wide Web was invented, JavaScript has been available in browsers. Even though AJAX came into being, JavaScript still matters. This resulted in Developers love JavaScript. It doesn't matter what server-side language is being used, Client-side scripts are preferred to be written in JavaScript. JavaScript understanding and By having Node's JavaScript implementation implemented, a developer can utilize it to work with Node.js and many other server side features. JavaScript as a Language – Get Everything, Node has become popular as a platform and framework thanks to strong community support and its adoption keeps on increasing.

CONCLUSION:

From the advent of WWW up to date, JavaScript has always been available on browsers. Even with AJAX, JavaScript is still significant. Criticism notwithstanding, Popularity of JavaScript among developers. Regardless of the server-side language used, JavaScript is always the customer's choice. Node for non-browser server-side scripting provides a place to work, includes free examples and useful libraries to use. The module creates a built-in import tool called NPM. Event-driven I/O, non-blocking, asynchronous programming are three terms that can be used to define how node.js achieves lightweight and high performance. Any business using Node basically can: Utilize less servers, fewer engineers as well as reduce time spent on site. In order for use by developers, they need knowledge about JavaScript and Node compatibility, coding skills for server side programming and so many others. Make the most out of JavaScript as a language as well as With strong community support, Node has become a popular platform and framework with its adoption still.

REFERENCES:

[1]. Ethan Brown. 2014. Web Development with Node and Express. O'REILLY.

[2]. Ghansham Jadhav1, Flavia Gonsalves2 . 06 June 2020. Role of Node.js in Modern Web Application Development. International Research Journal of Engineering and Technology (IRJET).

[3]. David Herron. 2020. Node.js Web Development. Edition 5. Packt Publishing.

[4]. Nimesh Chhetri. February, 2016. A Comparative Analysis of Node.js (Server-Side JavaScript). St. Cloud State University.

[5]. Florian Rappl. November, 2022. Modern Frontend Development with Node.js. Packt.