

NODE.JS:

Lightweight, Event driven I/O web development

As number of internet users is rising steadily, the servers are getting flooded with their requests and responses. To avert this situation, the servers are highly equipped with the high-end hardware like maximum number of core/processor, memory, enhanced I/O devices etc. But, these arrangements are not quite enough to handle the increasing active real-time communication even with the traditional web development platforms like ASP.Net, Java, Python, Ruby & etc. Node.js is the new platform developed to address all these issues with less and concise coding for light-weight, highly scalable, I/O non-blocking web apps development.



A.K. HOTA
Senior Technical Director
ak.hota@nic.in



D. MADAN PRABHU
Scientific Officer/Engr-SB
madan.prabhu@nic.in

Edited by

PRASHANT BELAWARIAR

WHAT IS NODE.JS?

Node.js is a software platform that is built on Chrome's V8 JavaScript runtime for building scalable network applications effortlessly. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Node.js, initially developed by Ryan Dahl, also provides an REPL (Read-Eval-Print-Loop) environment for interactive testing.

WHY JAVASCRIPT INCLUDED IN NODE.JS?

- Asynchronous - JavaScript is naturally asynchronous with event model well suited for building highly scalable web applications through callbacks.
- Less Learning curve - A huge base of developers is already familiar with both JavaScript and asynchronous programming from years, developing JavaScript in web browsers.

- Lighting Fast Script engine – Huge advances in execution speed has made it practical to write server side software entirely in JavaScript.

- Code Sharing - Developers can write web applications in one language, which helps by reducing the "context" switch between client and server development, allowing code sharing between client and server.

- Code Transformation - JavaScript is a compilation target and there are a number of languages that have compiled to it already.

- Support for NoSQL - JavaScript is the language used in various NoSQL databases (e.g. CouchDB / MongoDB) so interfacing with them is a natural fit.

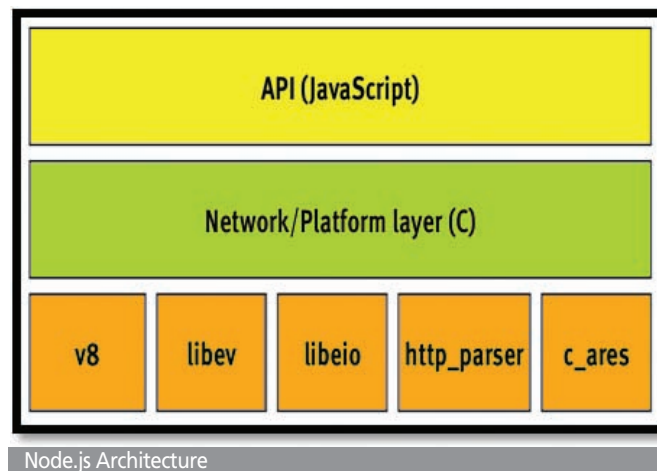
- JSON - It is a very popular data interchange format today and it is native JavaScript.

NODE.JS ARCHITECTURE

Node.js platform consists of three layers. The base layer contains all the core components, middle layer acts as a middle-ware by establishing communication from lower to top layer.

The final top layer consists of all JavaScript API. The core components are as follows:

- V8 - Open source JavaScript engine developed by Google
- Libev – Implements event loop and abstracts the underlying specific technologies use (such as select, epoll, etc)



Node.js Architecture

```
var http = require("http");
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var buffer = '<html>\n<head>\n<title>Speed
test</title>\n</head>\n<body>\n';
  for (var i = 0; i < 10000; i++)
    buffer += '<p>Hello world</p>\n';
  buffer += '</body>\n</html>';
  res.end(buffer);
}).listen(8080);          test.js
```

```
<?php
echo "<html>\n<head>\n<title>Speed test</title>\n</head>\n<body>\n";
for ($i = 0; $i < 10000; $i++) {
  echo "<p>Hello world</p>\n";
}
echo "</body>\n</html>"; ?>
```

The Scripts

- libeio - Asynchronous I/O library uses a thread pool to execute blocking calls in the background.
- c_ares - A non-blocking/asynchronous DNS resolution library
- http_parser - Parser for HTTP messages

NPM: NODE PACKAGE MANAGER

The Node Package Manager (npm) is a utility bundled with Node.js that offers a set of publicly available, reusable components, available through easy installation via an online repository, with version and dependency management. A full list of packaged modules can be found on the NPM website <https://npmjs.org/>, or accessed using the NPM CLI tool. The module ecosystem is open to all and anyone can publish their own module that will be listed in the NPM repository.

WHERE DOES NODE.JS FIT?

Node.js is best suited for data-intensive real-time (DIRT) applications. Since Node itself is very light weight on I/O, it is good at shuffling/ proxying data from one pipe to another, data streaming, push notification. The single threaded event model of Node doesn't fit for heavy computation process. CPU intensive application will block the node responsiveness with current connection, meanwhile rest of the connection kept in the queue to serve later. Real time applications are best use cases for Node. Areas where we can utilize the Node capabilities for implementation are:

- Real time communication systems like CHAT, MAIL, Quick SMS, Team collaboration application etc.
- Data streaming & proxy
- Used to enable real time communication

to current web viewers for further discussing the service. May be used in dial.gov.in like sites.

- Result publication sites using redis/memcache(NoSQL) database as backend
- Train ticket booking system
- Stock brokerage systems
- System & Application Monitoring Dashboards
- Active real time Dashboards of several web service/REST

NODE.JS VS PHP BENCHMARKING

The Testing box used:

- Intel(R) Core(TM) i5-2300 CPU @ 2.80GHz (4 cores)
- 4GB DDR3 RAM
- Linux Mint 14 Nadia
- Nodejs – v0.10.22
- Apache/2.2.22
- PHP 5.4.6

The Scripts:

To know, how our application may perform during peak load occasions, we tested our scripts with 100 simultaneous active connections over 10 seconds using siege tool (http/https regression testing and benchmarking utility). The performance reports indicates how efficient Node.js is during higher load.

Node.js is not a silver bullet that will dominate the web development world. Instead, it is a platform that fills a particular void between the needs and the technology. Node.js was never created to solve the compute scaling problems, instead it was genesis was based on resolving the I/O scaling problems, which it does really well with its built-in filesystem (fs) module. If the use case neither contains CPU intensive operations nor access any blocking resources, one can exploit the benefits of Node.js and enjoy fast and scalable network applications.

FOR FURTHER INFORMATION:

A.K. Hota

Senior Technical Director

NIC Odisha State Centre, Bhubaneswar

E-mail: ak.hota@nic.in

```
Terminal x Terminal x
Madan nodejs # siege -c 100 -b -t 105 localhost/test.php
** SIEGE 2.70
** Preparing 100 concurrent users for battle.
The server is now under siege...
Lifting the server siege...      done.
Transactions:      8707 hits
Availability:      100.00 %
Elapsed time:      9.46 secs
Data transferred:  4.70 MB
Response time:     0.11 secs
Transaction rate:  920.40 trans/sec
Throughput:        0.50 MB/sec
Concurrency:       99.31
Successful transactions: 8707
Failed transactions: 0
Longest transaction: 0.14
Shortest transaction: 0.01
Performance Report

Madan nodejs # siege -c 100 -b -t 105 localhost:8080
** SIEGE 2.70
** Preparing 100 concurrent users for battle.
The server is now under siege...
Lifting the server siege...      done.
Transactions:      17500 hits
Availability:      100.00 %
Elapsed time:      9.36 secs
Data transferred:  3172.14 MB
Response time:     0.05 secs
Transaction rate:  1869.66 trans/sec
Throughput:        338.90 MB/sec
Concurrency:       99.85
Successful transactions: 17500
Failed transactions: 0
Longest transaction: 0.09
Shortest transaction: 0.04
```