OneShot Assignment

NAME:RAUNAK SRIVASTAVA

REGISTRATION NUMBER:17BEC1169

NODE JS SERVER ARCHITECTURE

Node.js uses the “Single Threaded Event Loop” architecture to handle multiple concurrent clients. Node.js Processing Model is based on the JavaScript event-based model along with the JavaScript callback mechanism.

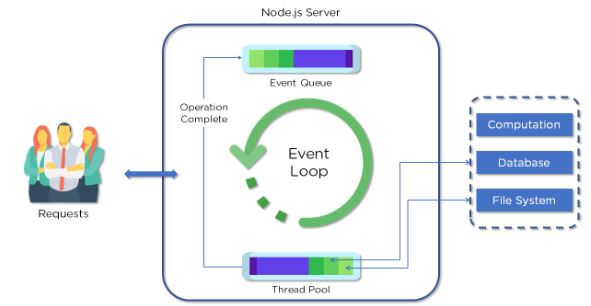


Fig: Node.js architecture

Now let’s understand each part of the Node.js architecture and the workflow of a web server developed using Node.js.

Parts of the Node.js Architecture:

* Requests

Incoming requests can be blocking (complex) or non-blocking (simple), depending upon the tasks that a user wants to perform in a web application

* Node.js Server

Node.js server is a server-side platform that takes requests from users, processes those requests, and returns responses to the corresponding users

* Event Queue

Event Queue in a Node.js server stores incoming client requests and passes those requests one-by-one into the Event Loop

* Thread Pool

Thread pool consists of all the threads available for carrying out some tasks that might be required to fulfill client requests

* Event Loop

Event Loop indefinitely receives requests and processes them, and then returns the responses to corresponding clients

* External Resources

External resources are required to deal with blocking client requests. These resources can be for computation, data storage, etc.

The Workflow of Node.js Architecture:

A web server developed using Node.js typically has a workflow that is quite similar to the diagram illustrated below. Let’s explore this flow of operations in detail.

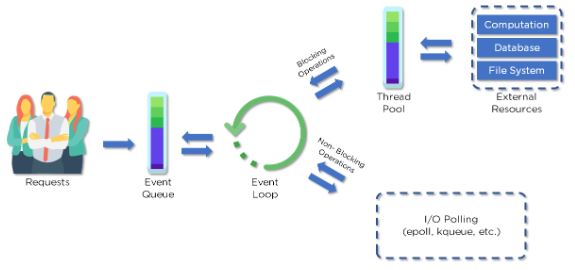


Fig: Node.js Architecture Workflow

* Clients send requests to the webserver to interact with the web application. Requests can be non-blocking or blocking:

**NOTE:SIR BEING AN ECE STUDENT I DON’T HAVE ANY KNOWLEDGE IN FULL STACK WEB DEVELOPENT WHETHER IT IS FRONT END OR BACKEND.**