Node.js

Content

- Introduction
- History
- Advantages
- Disadvantages
- And Many More

Node.js

- Node.js is a runtime environment
- It runs applications exterior to the browser
- Node.js increases the performance and efficiency.
- It uses non blocking input-output and asynchronous events.
- Node.js applications are single threaded.

Node.js Introduction

- It helps in development
- To execute code it uses Google v8 JavaScript engine.
- Node.js application are based on events
- In this incoming requests are processed in constant event stack.
- Request are send one by one

Node.js Introduction

- In simple words Node.js is ‘server-side JavaScript’.
- In not-so-simple words Node.js is a high-performance network applications framework, well optimized for high concurrent environments.
- It’s a command line tool.
- In ‘Node.js’ , ‘.js’ doesn’t mean that its solely written JavaScript. It is 40% JS and 60% C++.

Node.js Introduction

- Node.js uses an **event-driven, non-blocking I/O** model, which makes it lightweight. (from nodejs.org!)
- It makes use of **event-loops via JavaScript's**
- **callback functionality to implement the nonblocking** I/O.
- Programs for Node.js are written in JavaScript but not in the same JavaScript we are use to. There is no DOM implementation provided by Node.js, i.e. you **can not do this:**

```
var element = document.getElementById(‚elementId‛);
```

- Everything inside Node.js runs in a **single-thread.**

Node.js Features

Following are some of the important features that make Node.js the first choice of software architects.

Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

Node.js Features

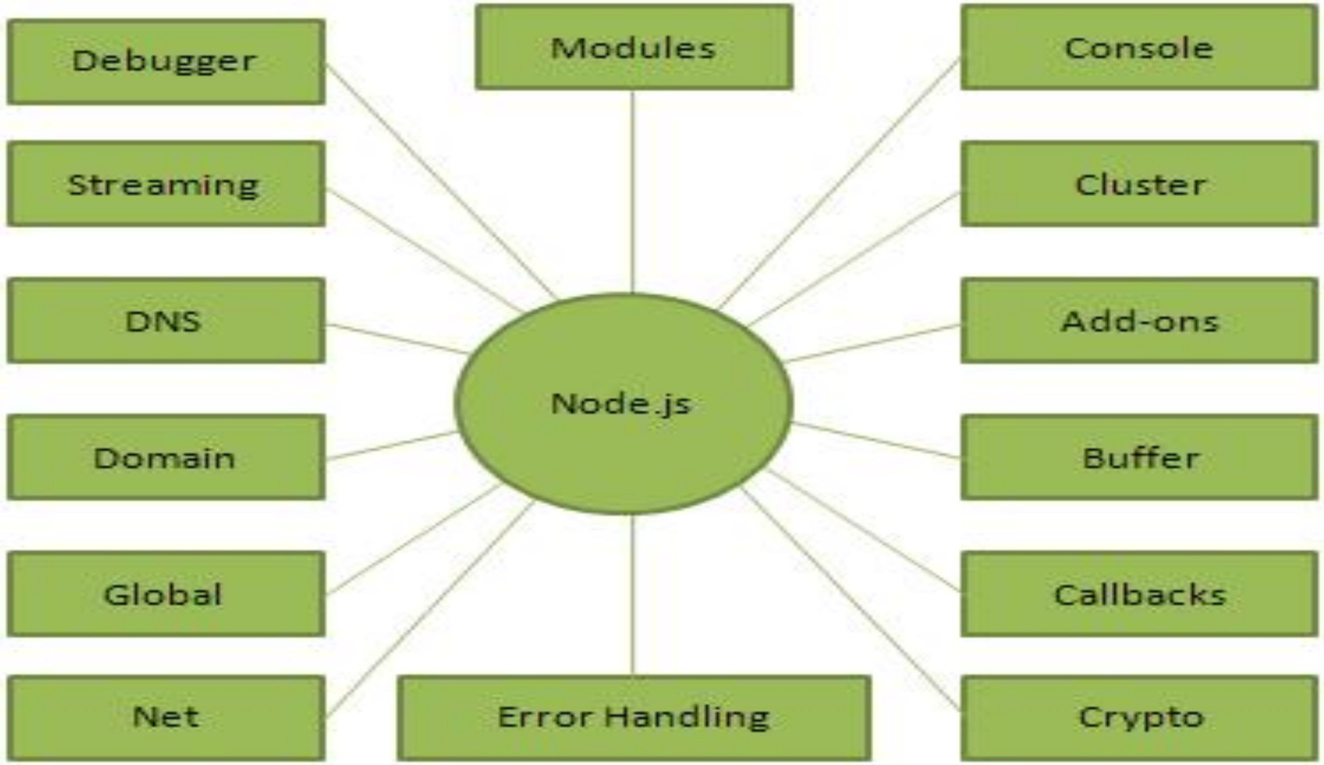
Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.

License − Node.js is released under the MIT license

Node.js Features

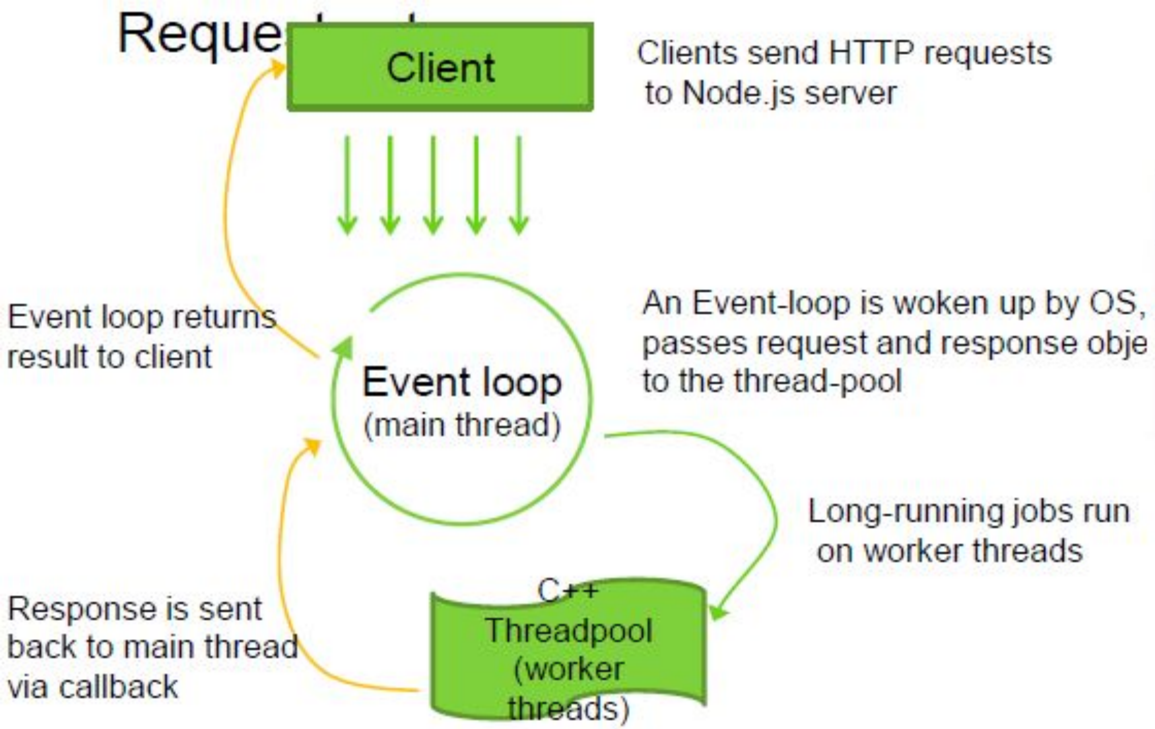
The following diagram depicts some important parts of Node.js which we will discuss



Source https://www.tutorialspoint.com/nodejs/index.htm

Event Loops

Event-loops are the core of event-driven programming, almost all the UI programs use event-loops to track the user event, for example: Clicks, Ajax Requests etc



Reque

Client

Clients send HTTP requests
to Node.js server

Event loop returns
result to client

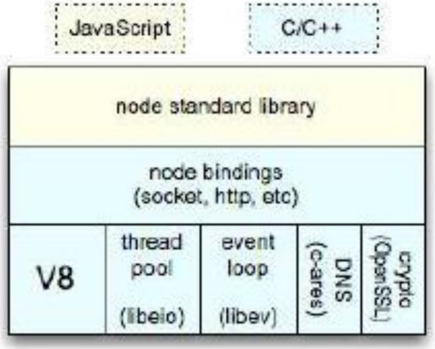
Event loop
(main thread)

An Event-loop is woken up by OS,
passes request and response obje
to the thread-pool

Long-running jobs run
on worker threads

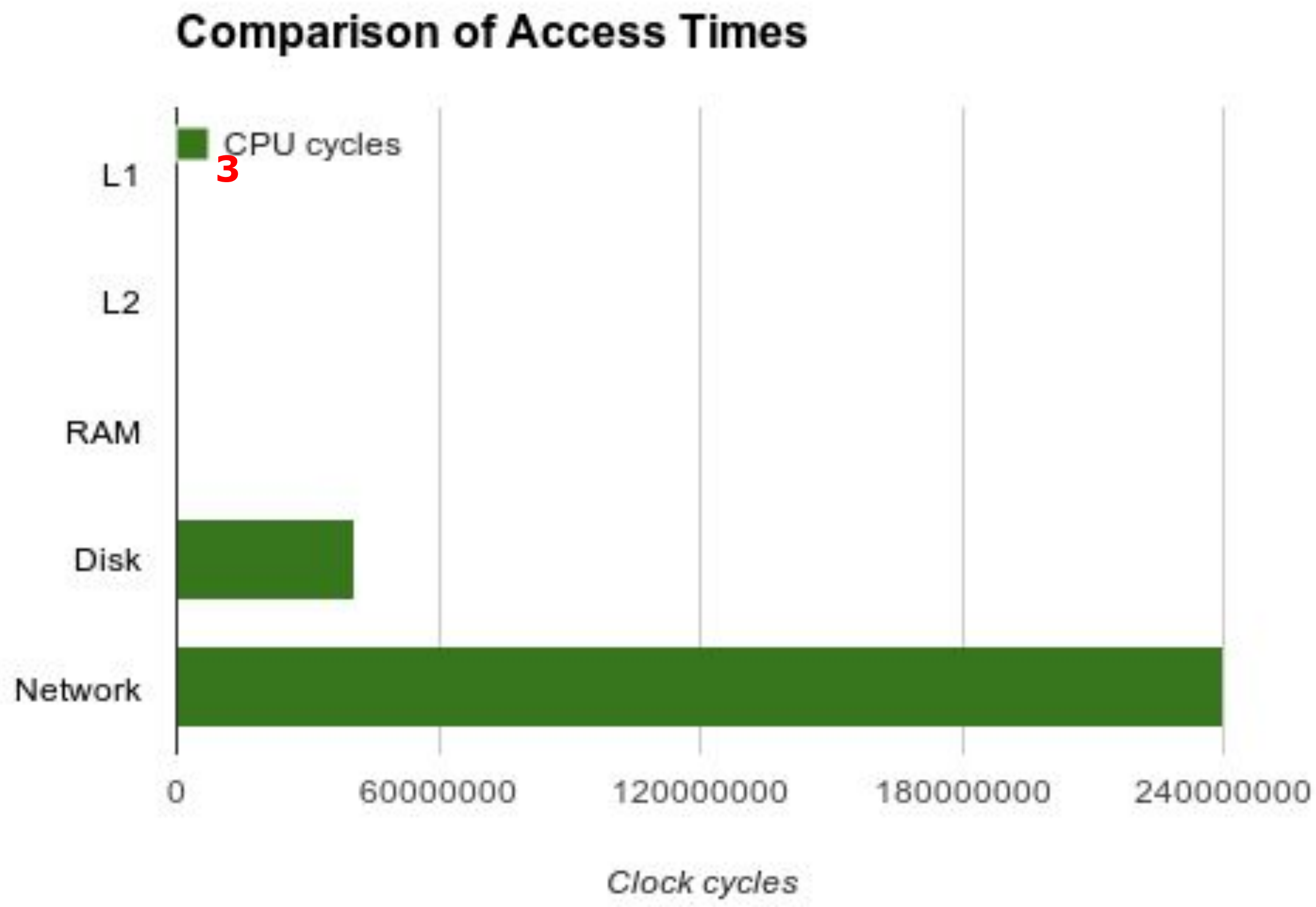
Response is sent
back to main thread
via callback

C++
Threadpool
(worker
threads)



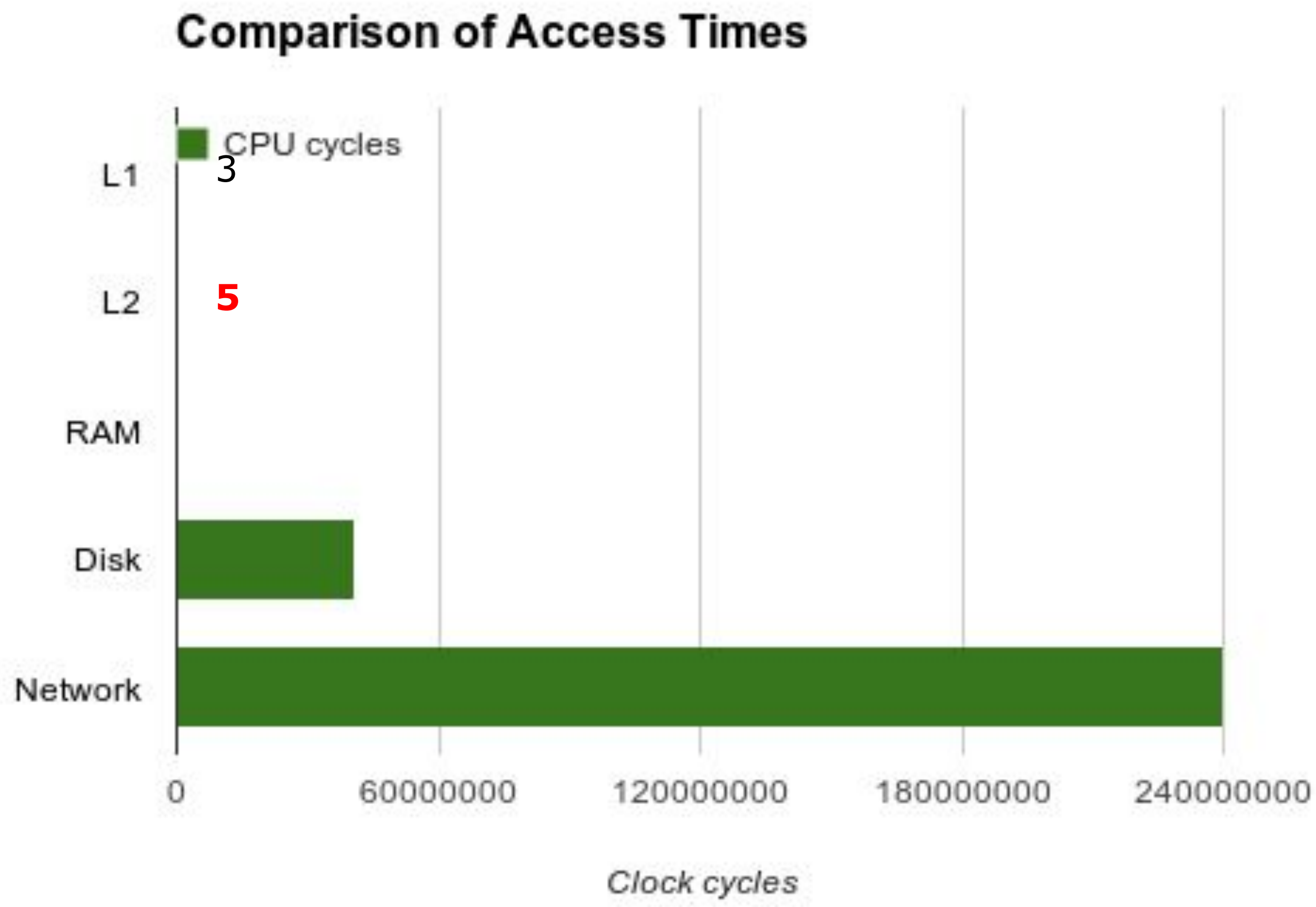
Source https://www.slideshare.net/

Node.js vs I/O



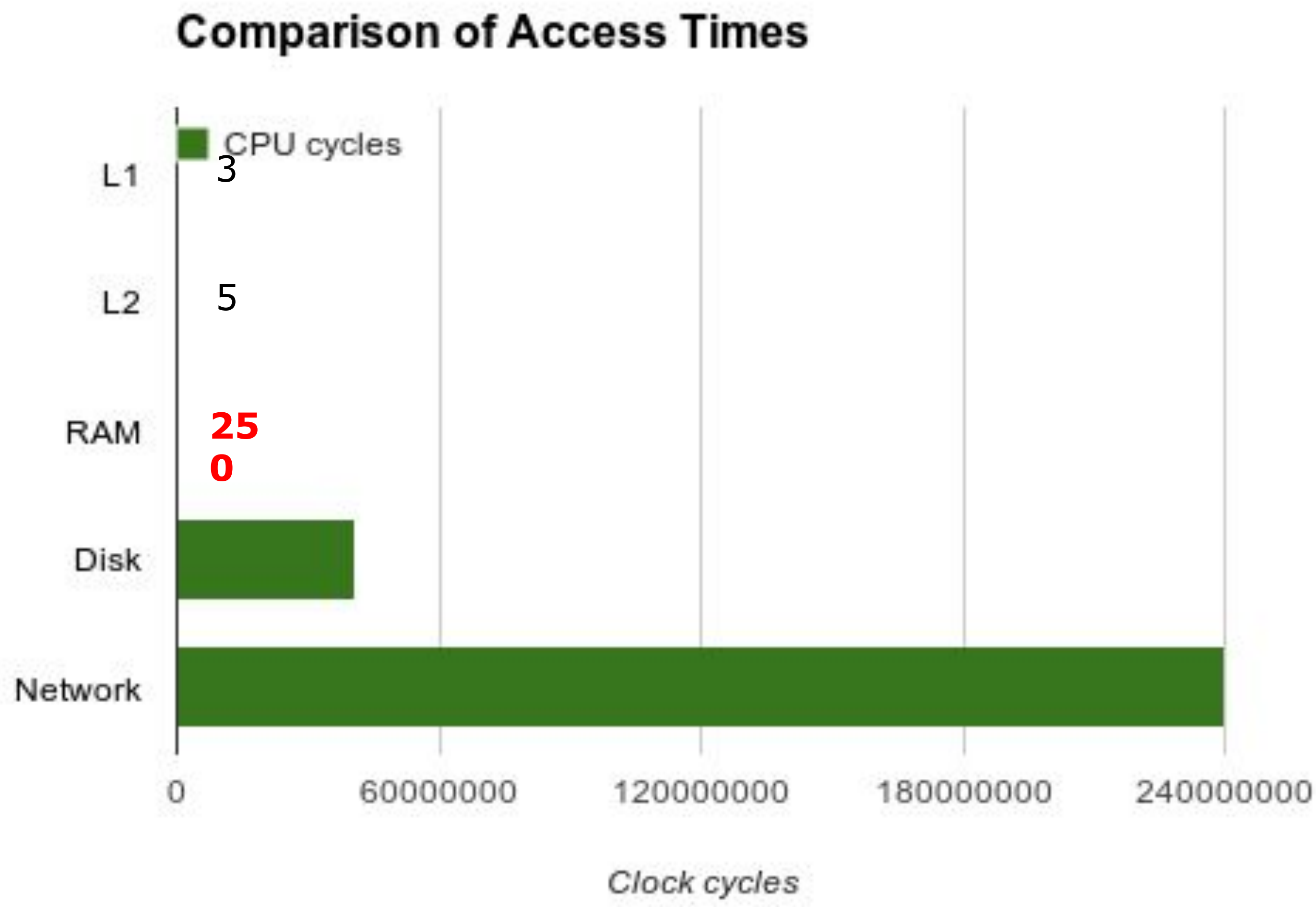
Source :
https://www.altamiracorp.com/blog/employee-posts/nodejs-basics-explai

Node.js vs I/O



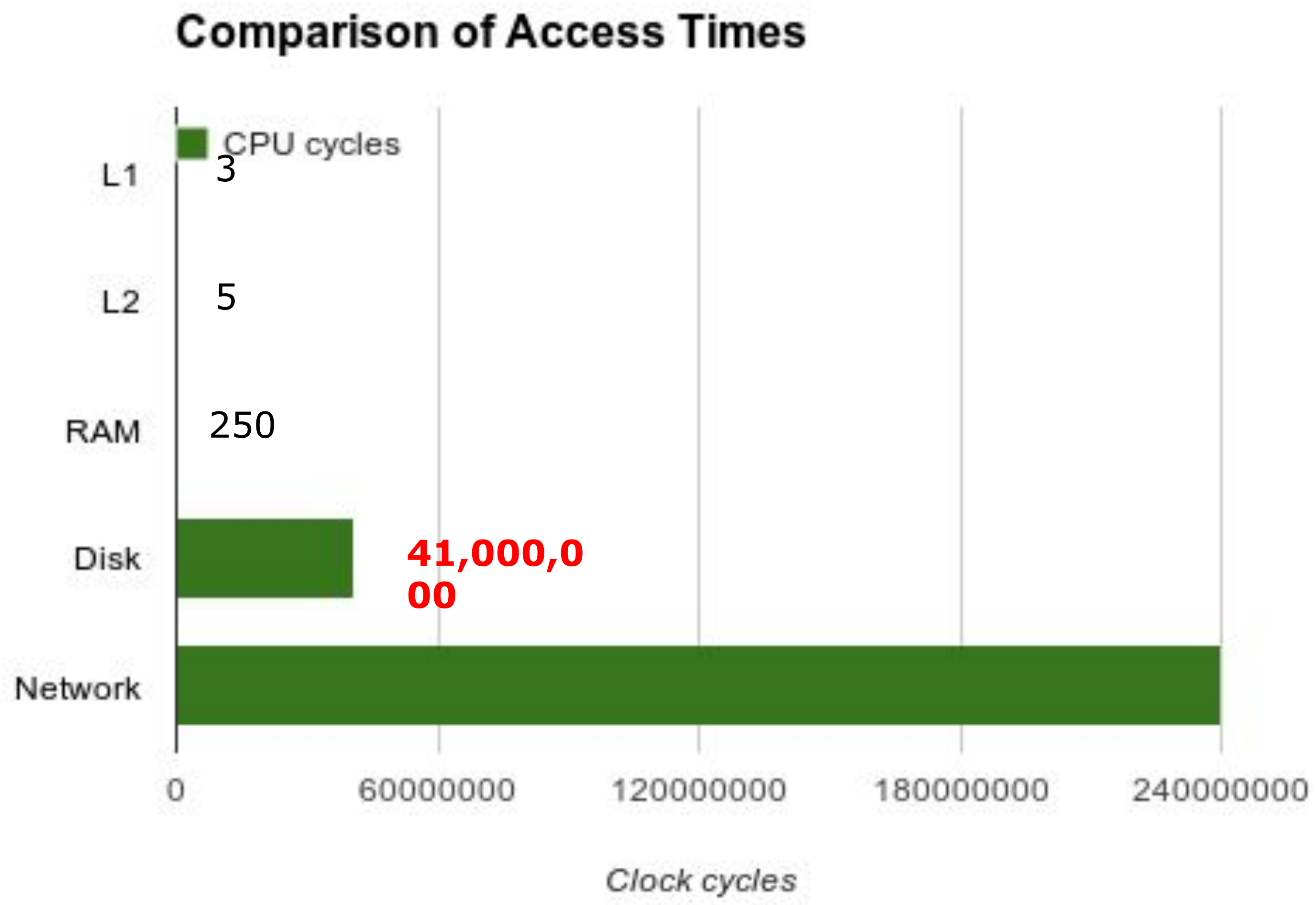
Source :
https://www.altamiracorp.com/blog/employee-posts/nodejs-basics-explai

Node.js and I/O



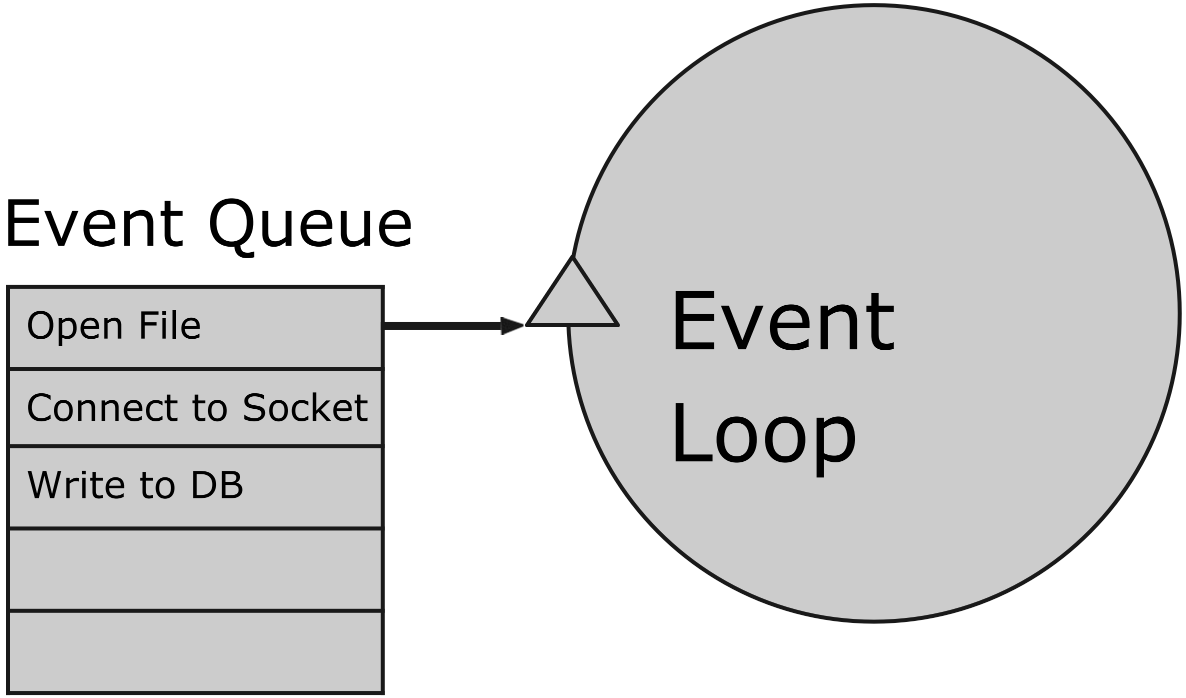
Source : https://www.altamiracorp.com/blog/employee-posts/nodejs-basics-explai

Node.js and I/O



Source :
https://www.altamiracorp.com/blog/employee-posts/nodejs-basics-explai

Node.js Event Loop

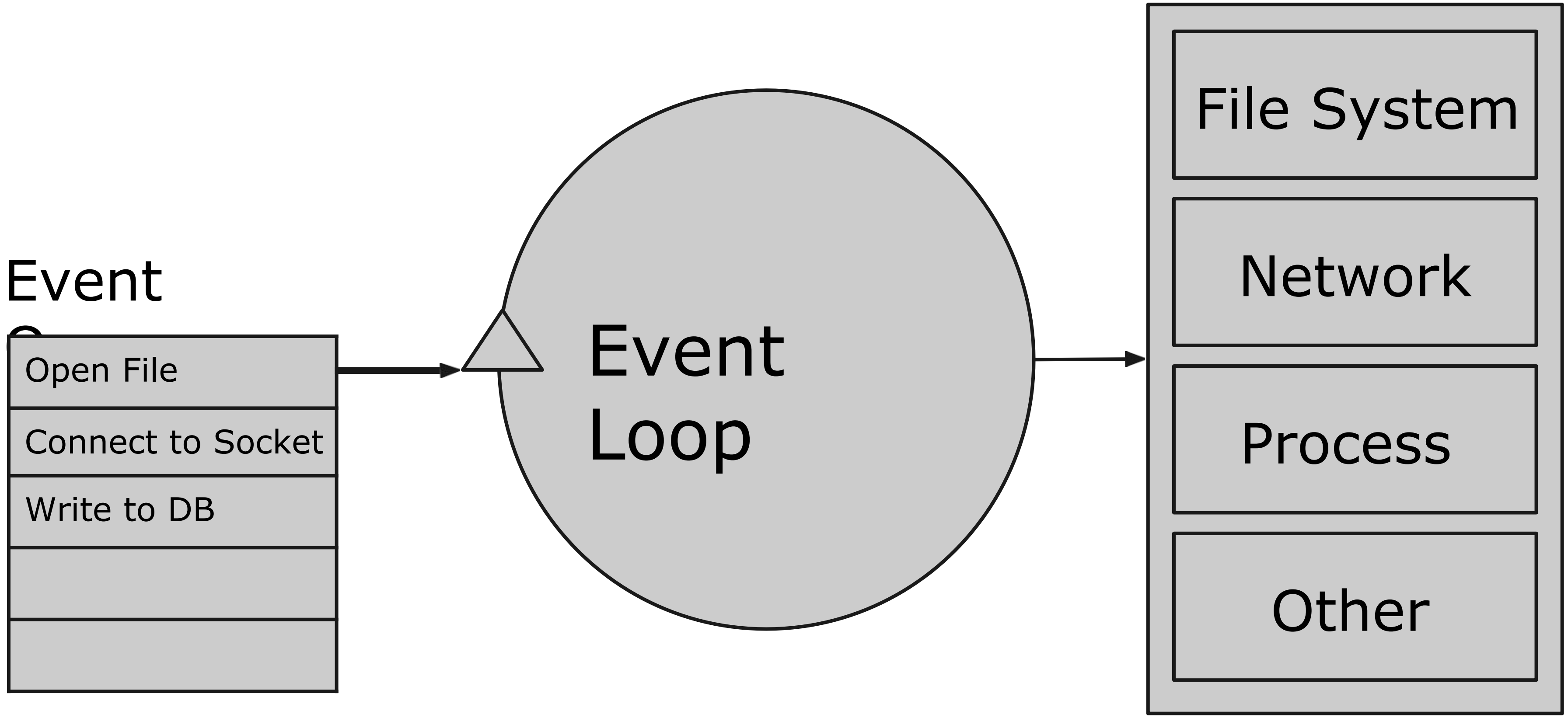


Event Queue

- Open File
- Connect to Socket
- Write to DB

Event Loop

Node.js Event Loop



Event
Q

Open File

Connect to Socket

Write to DB

Event
Loop

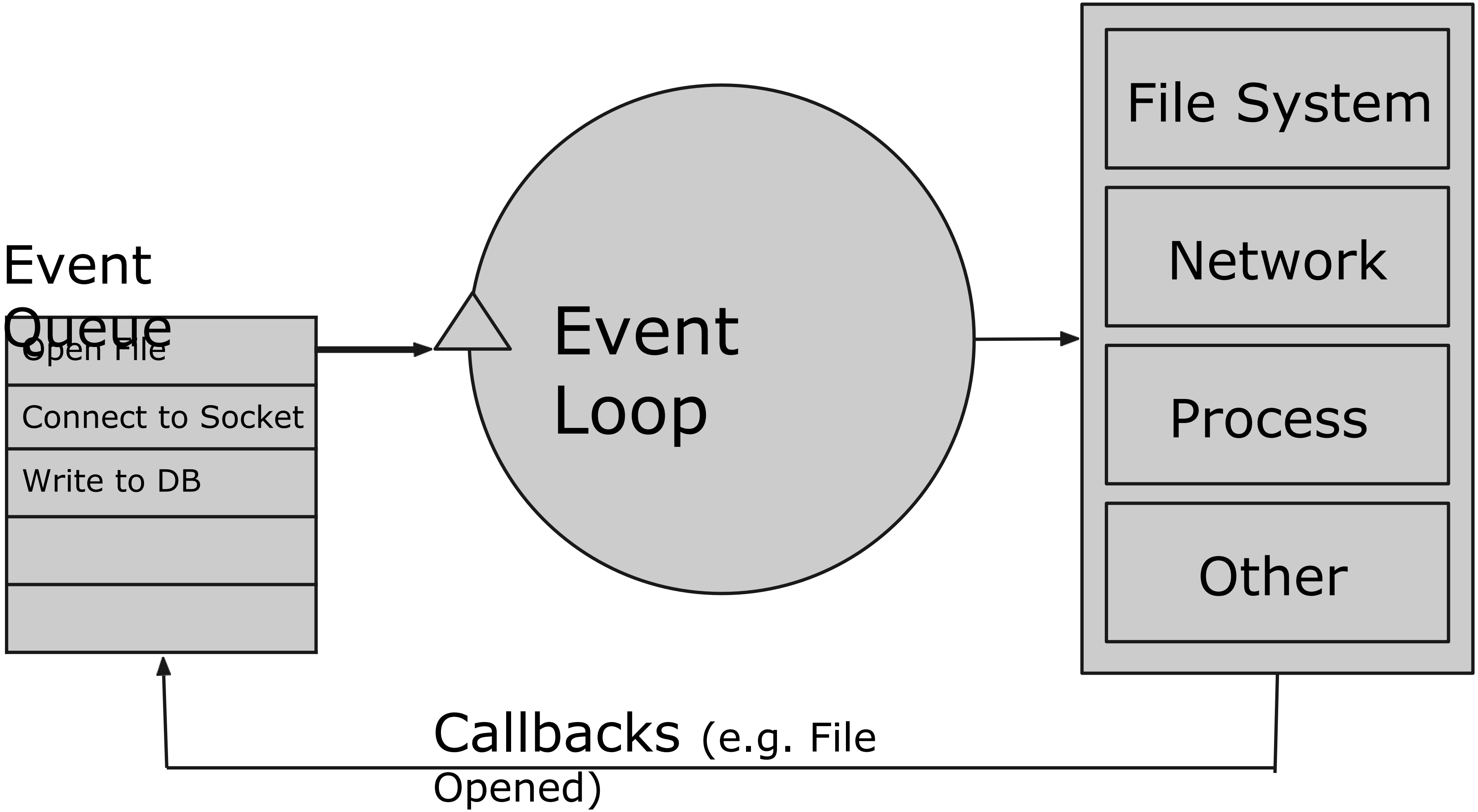
File System

Network

Process

Other

Node.js Event Loop



Event Queue

Open File

Connect to Socket

Write to DB

Event Loop

File System

Network

Process

Other

Callbacks (e.g. File Opened)

Node.js Example

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

fs.readFile(filename, function(err, data) {  var lines =
     data.toString().split('\n');
     console.log(lines.length);

});
```

Where to Use Node.js?

Following are the areas where Node.js is proving itself as a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

WHAT CAN YOU DO WITH NODE.JS?

- can create an HTTP server and print ‘hello world’ on the browser in just 4 lines of JavaScript.
- can create a TCP server similar to HTTP server, in just 4 lines of JavaScript
- can create a DNS server.
- can create a Web Chat Application like Gtalk
- can create a Static File Server in the browser.
- Node.js can also be used for creating online games, collaboration tools or anything which sends updates to the user in real-time.

What Can You Do With Node.js

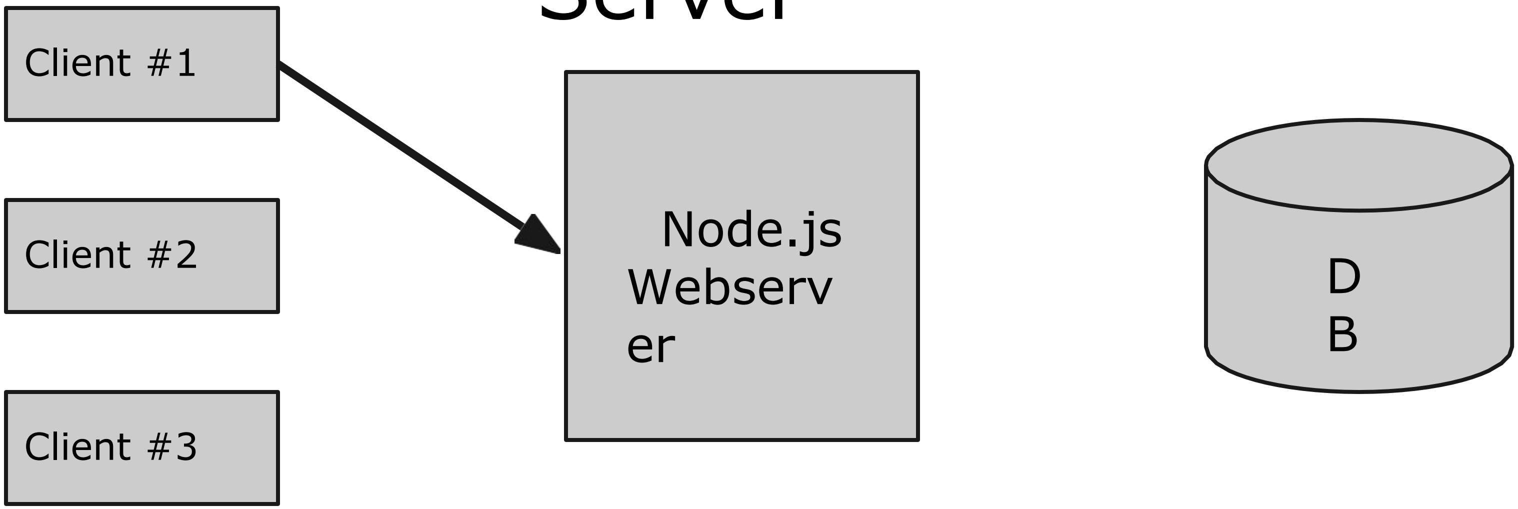
- Web Applications
  - Real-time applications
  - Event-based web sites
- REST Web Services
- iOS/Android
- Desktop Applications
  - Linux/Windows/OS X

Node.js is good at

- Chat Server
- Message Queue System
- Micro Services

Node.js good at

Chat Server



Client #1

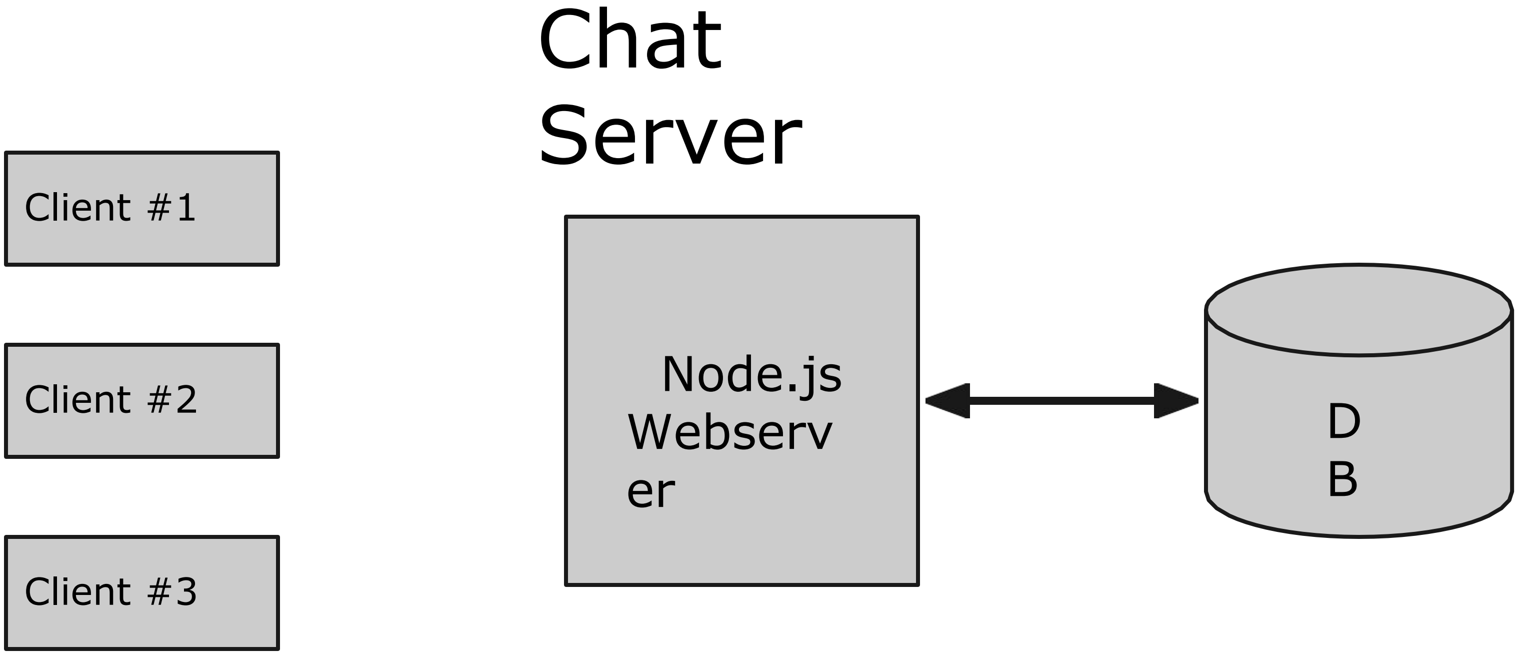
Client #2

Client #3

Node.js Webserver

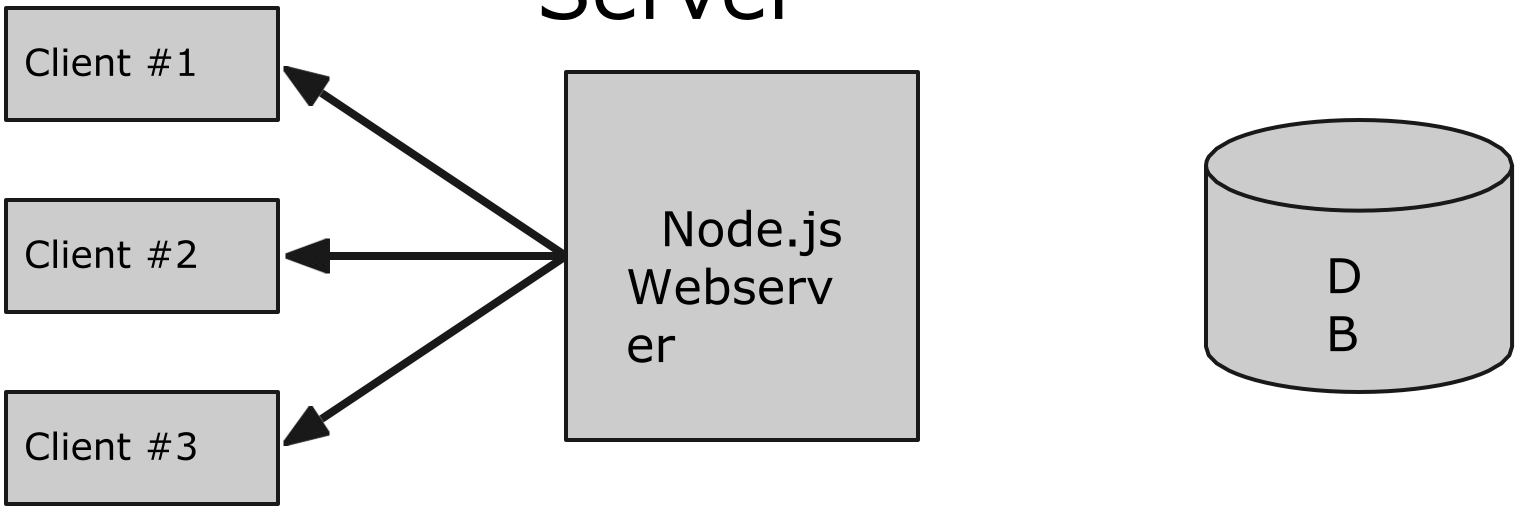
DB

Node.js good at



Node.js good at

Chat Server



Client #1

Client #2

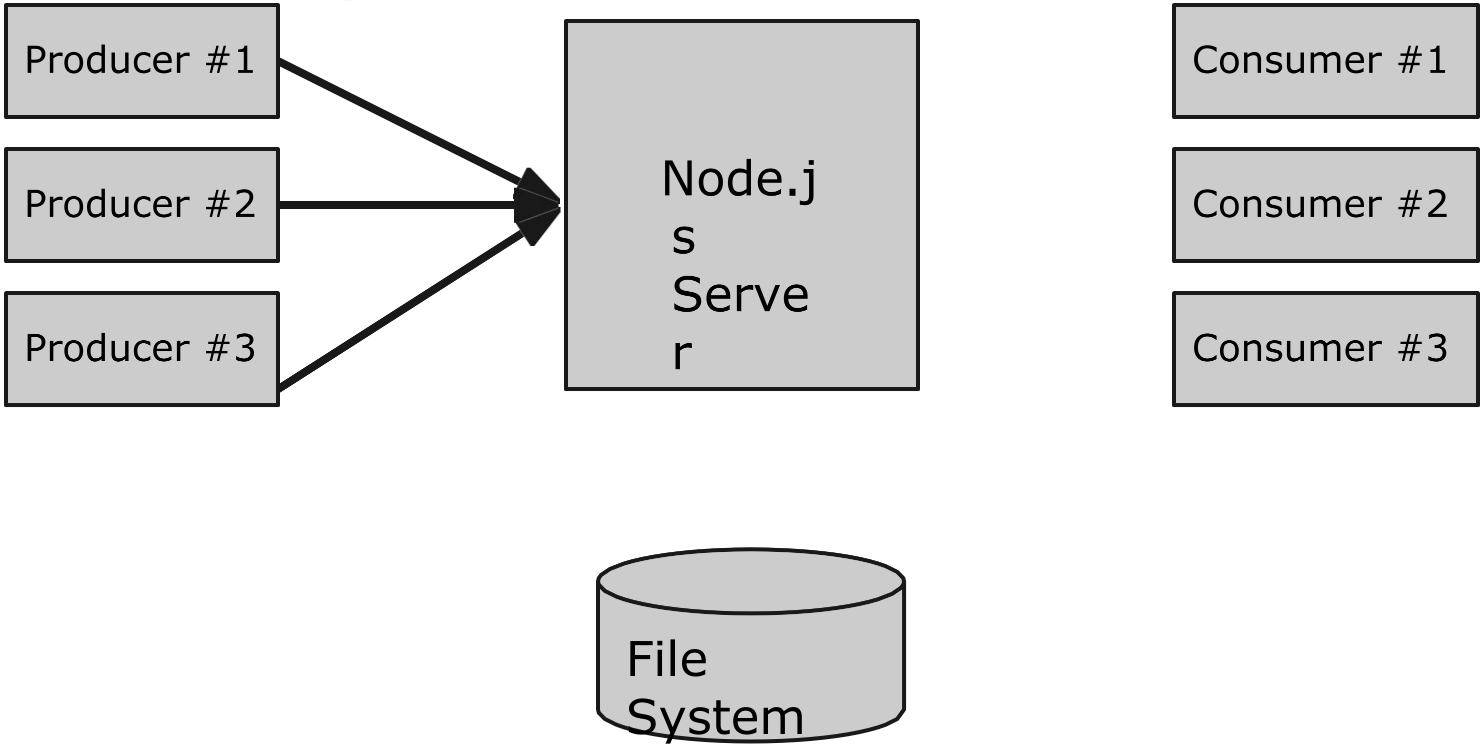
Client #3

Node.js Webserver

DB

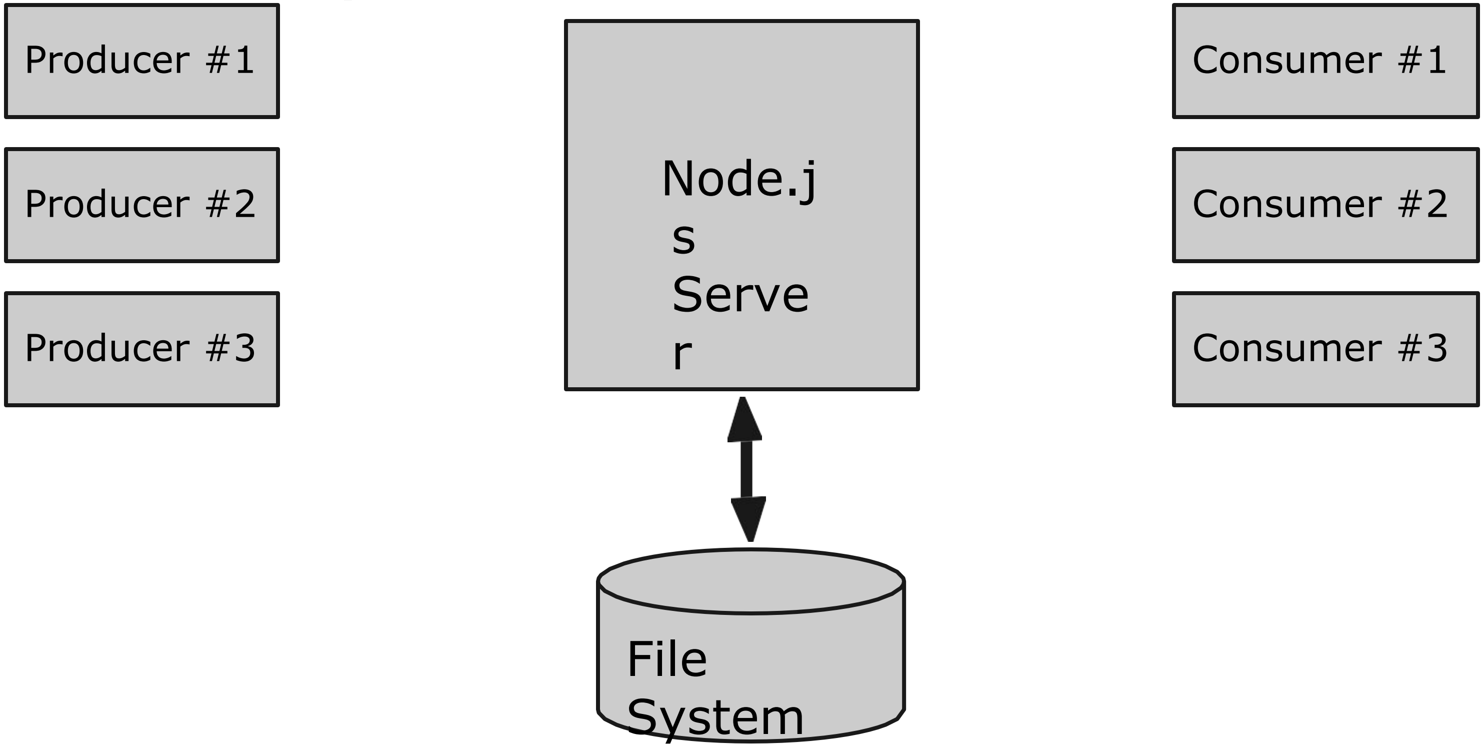
Node.js good at

Message Queuing System



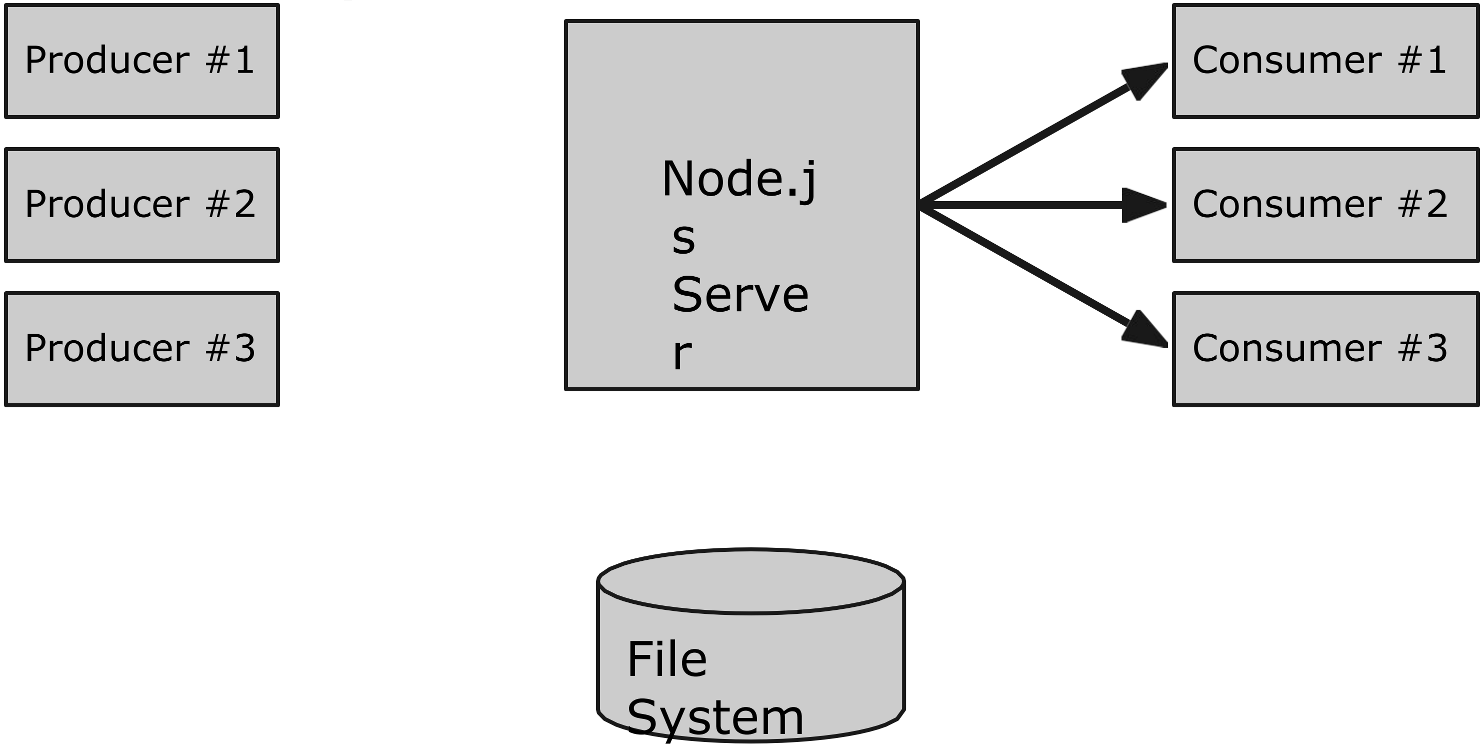
Node.js good at..

Message Queuing System



Node.js good at

Message Queuing System



Microservices

Micro Service

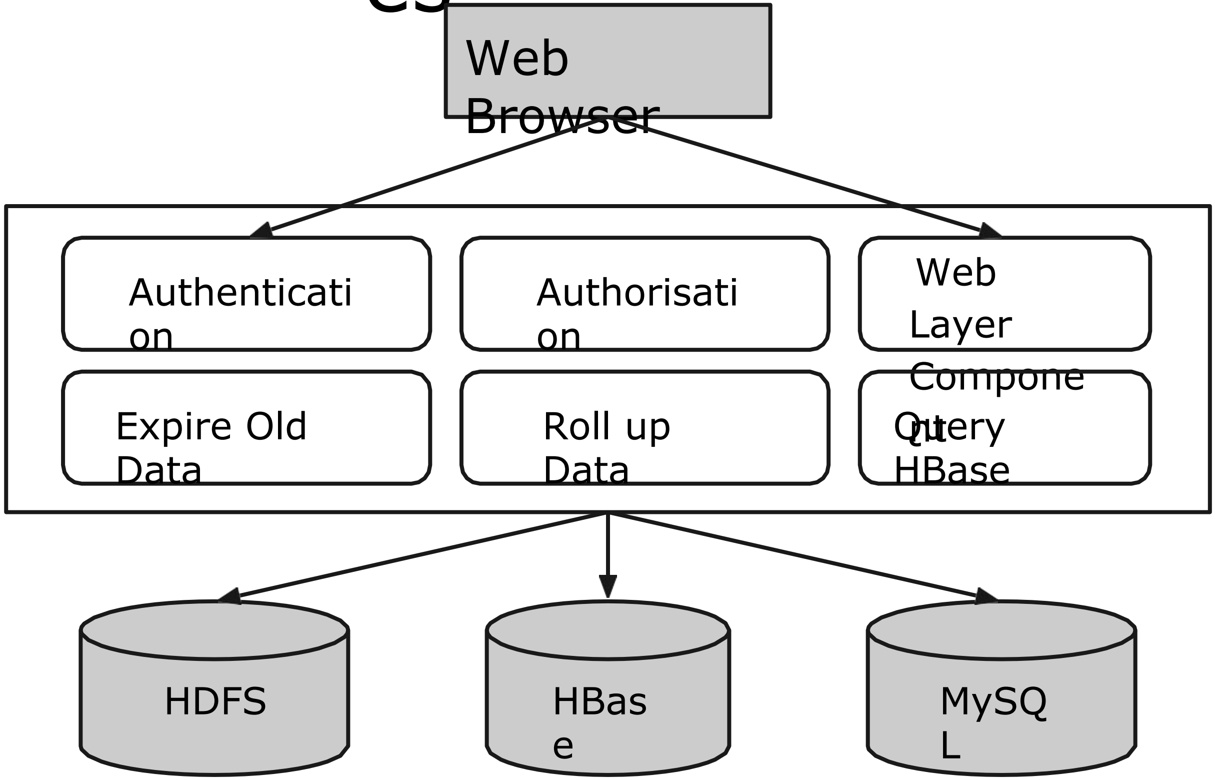
- Small unit of functionality
- About 100 lines long (can fit in your head)
- Independently deployed
- Loosely Coupled
- Codeable in a week
- Owned by one developer

Micro Service Example

- Store json object in MongoDB
- Calculate if it is time to scale app
- Receive user input and forward request to another part of your app

Node.js good at

Microservic
es



Web
Browser

Authenticati
on

Authorisati
on

Web
Layer
Compone
nt

Expire Old
Data

Roll up
Data

Query
HBase

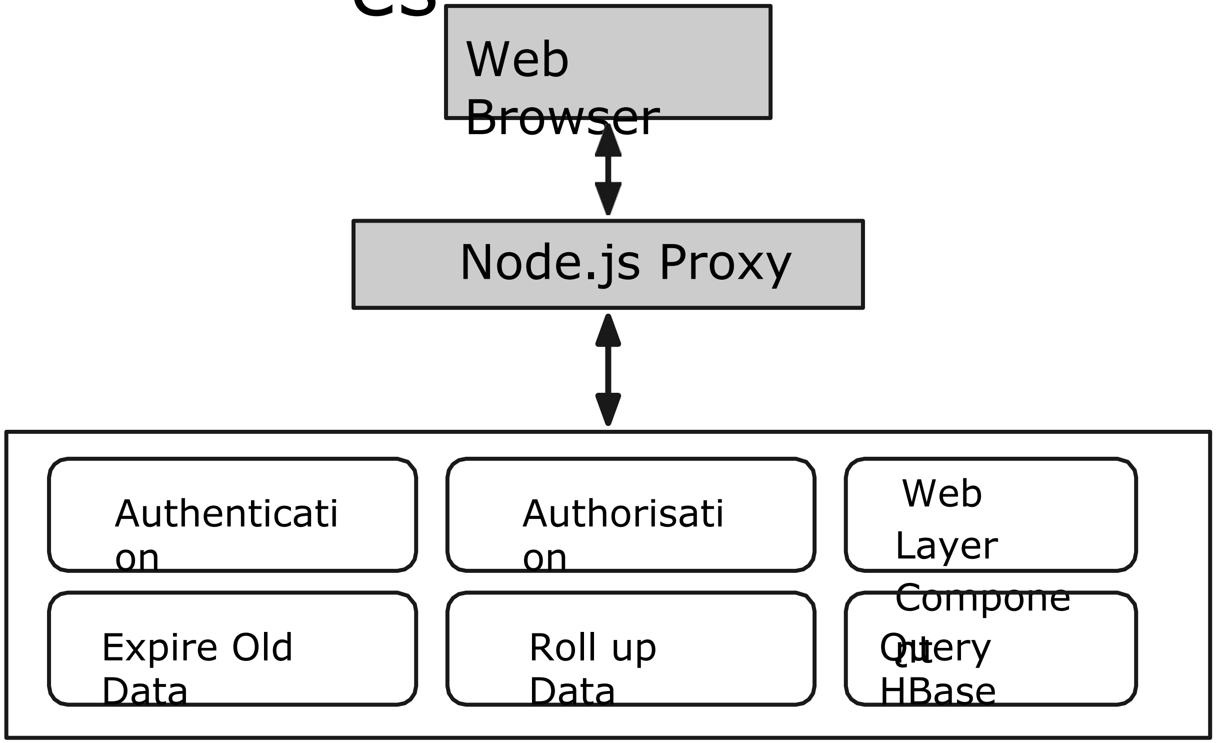
HDFS

HBas
e

MySQ
L

Node.js good at

Microservices



Web Browser

Node.js Proxy

Authentication

Authorisation

Web Layer Component

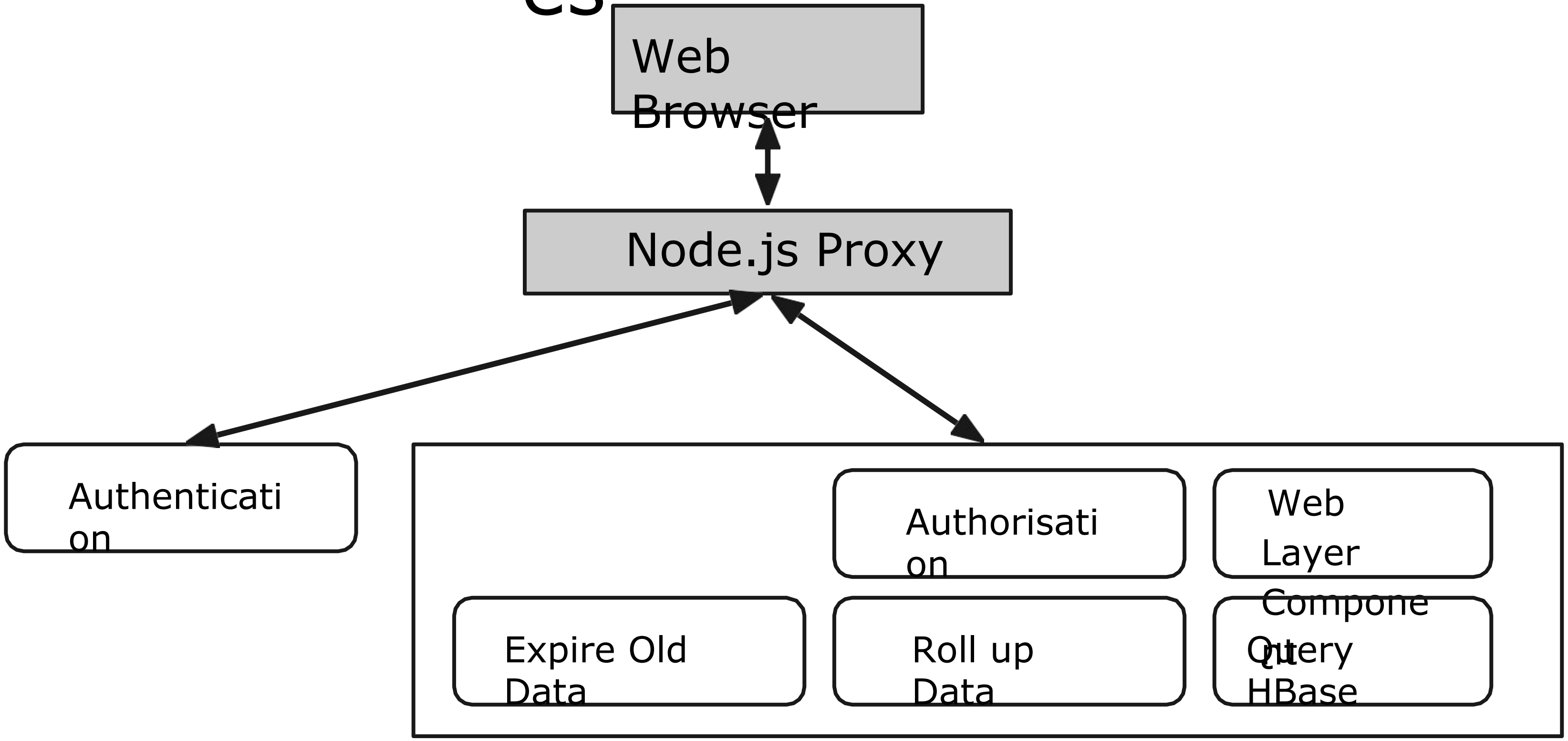
Expire Old Data

Roll up Data

Query HBase

Node.js good at

Microservices



Who Uses Node.js

- Dow Jones
- eBay
- LinkedIn
- Microsoft
- Pearson
- Yahoo
- Many more…

How To Install

- http://nodejs.org/download/
- Windows
  - Executable and installer
- Homebrew on OS X
  - brew install node
- Linux
  - https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager

Basic Commands

- node
  - – Starts node console
  - – node <.js filename>
  - – node –v
- npm
    - Node package manager
    - npm -v

“Hello, World!”

Simple “Hello World”:

```
	console.log(“Hello, World!”);
```

Web “Hello, World!”

```
var http = require('http');

http.createServer(function (request, response) {
    response.writeHead(200, {"Content-Type": "text/plain"});
    response.end("Hello, World!\n");
}).listen(8000);

console.log("Server running at http://127.0.0.1:8000/");
```

WHEN TO USE NODE.JS?

- Node.js is good for creating streaming based real-time services, web chat applications, static file servers etc.
- If you need high level concurrency and not worried about CPU-cycles.
- If you are great at writing JavaScript code because then you can use the same language at both the places: server-side and client-side.
- More can be found at http://stackoverflow.com/questions/5062614/how-to-decide-when-to-use-nodejs

Advantages Node.js

The advantages:

Need to use only a single programming language

Developers like the fact that they can write both front end and back end of the web applications with this runtime environment. They don’t have to use a separate server side programming language.

It’s faster

As the tool uses JS in the backend, you can execute the codes really quickly. And the additional advantage of it running through Google’s V8 engine, also makes it really fast, because it can compile JS directly into the machine code.

Enjoying the benefits of NPM

NPM or Node package Manager has a number of standalone tools and modules that you can download and use. It lets you manage the dependancies of the project’s tools. Node.js shares this with a shared repository of NPM, and the number of modules in it has increased considerably.

Advantages Node.js

Ideal solution for real-time web apps

If your task is to produce real-time web app, complete with games and chat options included in it, you can reliably turn to event-driven architecture of Node.js. Synchronisation process is a breeze as both server-side and client-side are written in JS. It’s done with the help of a web socket protocol. Node is just perfect for chat apps. Of course, there are other options like Ruby and Python, because they also have the best technology, but nothing can beat the performance of the event-driven tool.

Easy deployment of web applications

Due to the support of server and the web browser, businesses and organisations will find it easier to deploy web apps with Node.js. It is also an easy language, so programmers who are in the basic level of Java, and would like to upgrade, can easily do so. The reason that Java is the programming language also adds to the benefit because it can improve programmer productivity through code reusability. However, it is not suitable for beginners at the very basic level.

Single-threaded program

Being a single-threaded program, the tool offers many options like call-back methods and event loops to handle multiple concurrent clients. This can help remove the wait time in the server’s processes.

Disadvantages Node.js

API Interface is not stable

A notorious factor of Node.js is its unstable API interface. It keeps changing and that makes the developers struggle to change the accessible code bases to make sure it matches with the latest API version. In certain cases, you may have to encounter backwards-incompatible APIs.

The library is nothing good to write home about

The programming language fails to have a well-stacked library when compared to other programming language in its genre. So when you have to execute tasks like ORM or Object-Relational Mapping, handling database operations, image processing you have to rely on the common library.

Not good for heavy computation

Node.js has turned out to be a poor platform in terms of heavy computation, as it is a CPU intensive operation. And you might annul all the advantages you enjoyed with the language earlier due to the heavy computation issue.

Disadvantages Node.js

Not good with relational databases

Developers point out that using with Node.js with relational databases is not an easy task. The tools are still hugely underdeveloped, when compared to its competitors in the market. So if your operations include relational databases, you may have to go for other options like Rails, AS.NET MVC or Django.

Being an Asynchronous Programming Model

Being an asynchronous programming model, you can make apps that scale really well. However, many developers find the programming model difficult to use, especially the developers who are more familiar with linear blocking I/O programming.

Popular Modules

- Express
  - Web application framework
- Socket.io
  - Helps make real-time applications
- Mongoose
  - MongoDB Object Modeling
- Jade
  - Template engine
- restify
  - REST API framework

Thanks