

# Node.JS Deep Dive

Training Deck By

Prasoon Kumar  
Senior Technical Architect  
Just Dial


Presentation Material

18th August, 2012

# What is node?

- Server-side JavaScript framework
- Written using Google V8 engine
- Uses CommonJS module system
- Has the **latest** ECMAScript5 features
- Non-blocking from ground up
- Servers are normally thread based but Node.JS is “Event” based. Node. JS serves each request in an Evented loop that can able to handle simultaneous requests.

# Who is using it?

- Sponsored by  Joyent
- Google
- Microsoft (native support in Azure)
- Yahoo
- eBay
- Facebook
- LinkedIn
- Many, many smaller companies

# Threads VS Event-driven

Threads	Asynchronous Event-driven
Lock application / request with listener-workers threads	only one thread, which repeatedly fetches an event
Using incoming-request model	Using queue and then processes it
multithreaded server might block the request which might involve multiple events	manually saves state and then goes on to process the next event
Using context switching	no contention and no context switches
Using multithreading environments where listener and workers threads are used frequently to take an incoming-request lock	Using asynchronous I/O facilities (callbacks, not poll/select or O_NONBLOCK) environments

# Node is...

## Excellent for:

Realtime comms  
Sockets, polling, game engines

Custom network services  
Network Traffic Collection tools, Media servers, cluster management, proxies

JSON web services  
Thin app layer on top of a datastore

Client-oriented web UIs  
Anything you'd build with Sinatra

IO heavy middleware applications

## OK for:

"Forms over data" CRUD apps  
Rails give you more

## Wrong for:

Doing CPU intensive processing  
Video transcoding, etc.  
Though it could proxy to a transcoder

# Node layers

JavaScript

Node standard library

Node bindings

C

V8

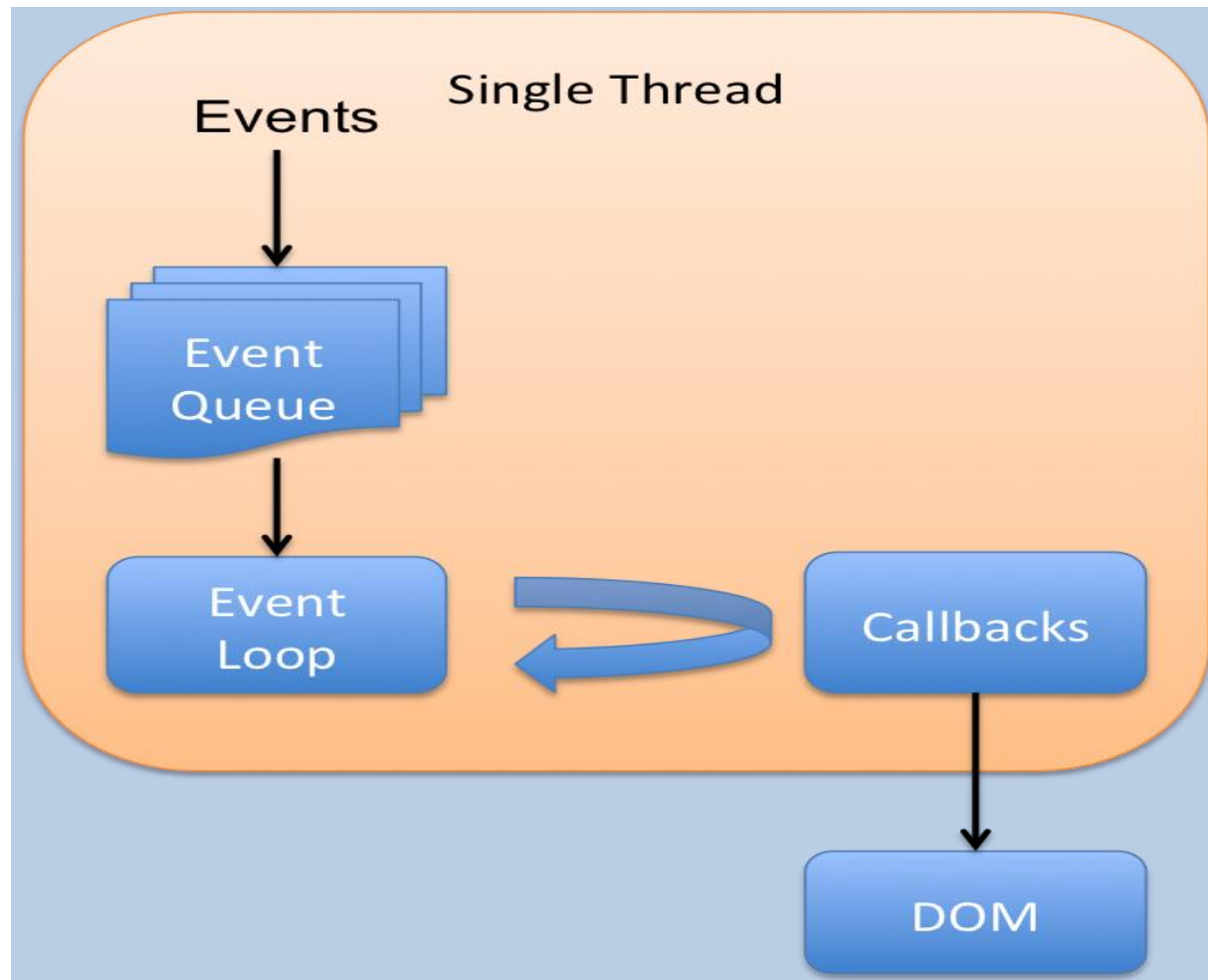
Thread  
Pool

libeio

Event pool

libev

# What? Event looping...



# Non-blocking IO

- The philosophy behind node is that system interactions should be non blocking.

Instead of this

```
var fs = require('fs'); // Import the 'fs' module
var data = fs.readFileSync(__dirname + '/example.file', 'utf8');
console.log(data); // print it!
```

Use this

```
var fs = require('fs'); // Import the 'fs' module
// Read the file asynchronously
fs.readFile(__dirname + '/example.file', 'utf8', function (err, data) {
  if (err) {
    return console.log(err);
  }
  console.log(data); // print it! });
```



# Installation

```
./configure --prefix=/opt/node  
make  
sudo make install
```

```
[prasonk@prasonk node-v0.8.6]$ node --version  
v0.8.6
```

```
[prasonk@prasonk node-v0.8.6]$ node
```

- 2+3
- 5 process.versions
- { http\_parser: '1.0', node: '0.8.6', v8: '3.11.10.17', ares: '1.7.5-DEV', uv: '0.8',
- zlib: '1.2.3', openssl: '1.0.0f' }

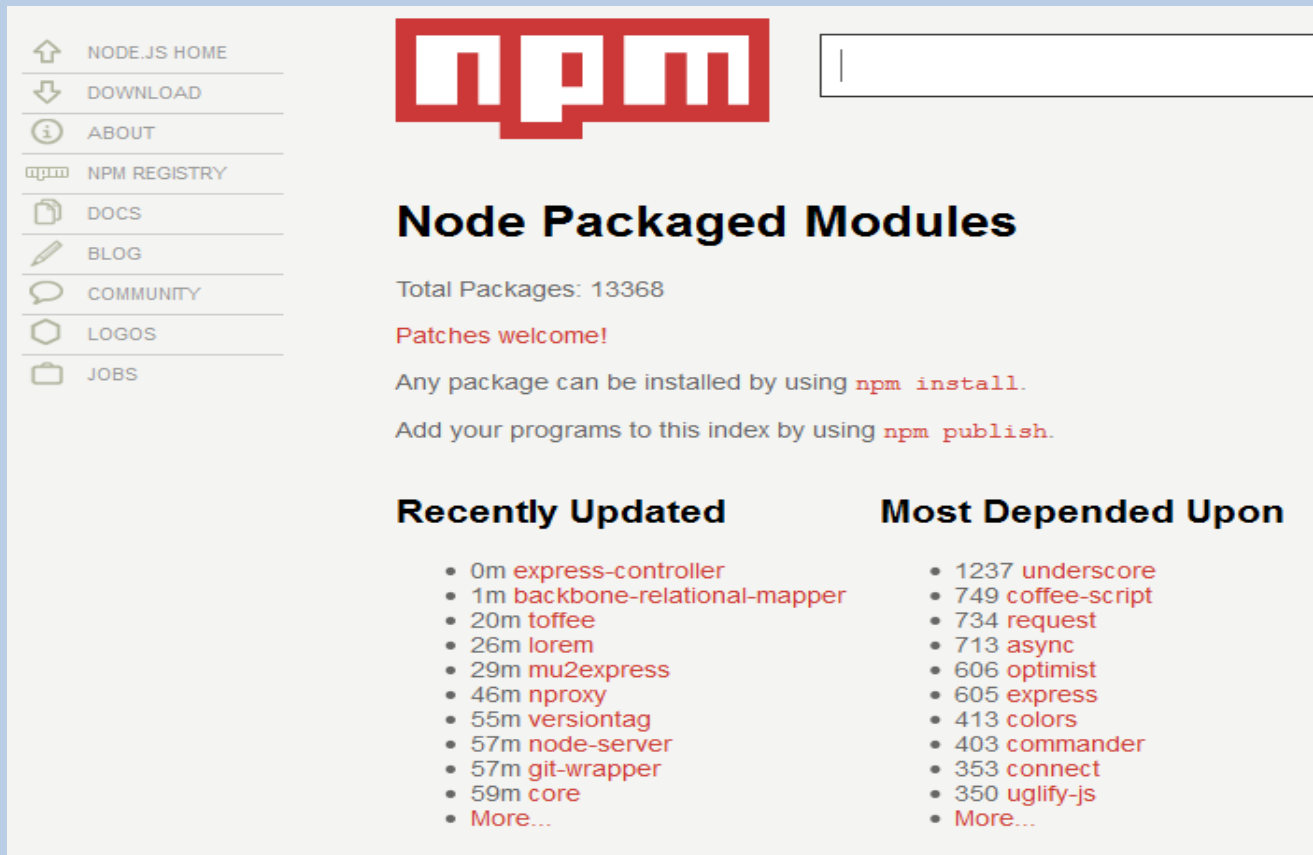
# What are packages?

- Ruby has Gems
- Python has PyPI
- PHP has PEAR (terrible)

**Node.js has Packages**

# Node Package Manager (NPM)

- <http://npmjs.org/>



The screenshot shows the NPM website homepage. On the left is a navigation menu with links: NODE.JS HOME, DOWNLOAD, ABOUT, NPM REGISTRY, DOCS, BLOG, COMMUNITY, LOGOS, and JOBS. The main content area features the NPM logo, a search bar, and the heading "Node Packaged Modules". Below the heading, it states "Total Packages: 13368" and "Patches welcome!". It provides instructions: "Any package can be installed by using `npm install`." and "Add your programs to this index by using `npm publish`.". At the bottom, there are two columns: "Recently Updated" and "Most Depended Upon", each containing a list of packages with their respective download counts.

**Navigation Menu:**

- NODE.JS HOME
- DOWNLOAD
- ABOUT
- NPM REGISTRY
- DOCS
- BLOG
- COMMUNITY
- LOGOS
- JOBS

**Search Bar:**

## Node Packaged Modules

Total Packages: 13368

Patches welcome!

Any package can be installed by using `npm install`.

Add your programs to this index by using `npm publish`.

### Recently Updated

- 0m `express-controller`
- 1m `backbone-relational-mapper`
- 20m `toffee`
- 26m `lorem`
- 29m `mu2express`
- 46m `nproxy`
- 55m `versiontag`
- 57m `node-server`
- 57m `git-wrapper`
- 59m `core`
- More...

### Most Depended Upon

- 1237 `underscore`
- 749 `coffee-script`
- 734 `request`
- 713 `async`
- 606 `optimist`
- 605 `express`
- 413 `colors`
- 403 `commander`
- 353 `connect`
- 350 `uglify-js`
- More...

# npm commands

- ls [filter] – installed, stable, @1.0
- Install package\_name@version branch
- rm

# NPM – Node Package Manager

```
> npm install somepackage
```

coffee-script

connect

jade

express

socket.io

redis

async

vows

request

# API Tour

- Platform
  - process
  - file system
  - Networking
    - net, dgram, http, tls (ssl)
- Utility
  - console
  - util
- Buffer
- Event Emitter
- Timer

# Process

- Process – PID, platform, memory usage
- Child\_process –
  - spawn and kill processes,
  - Execute commands and
  - Pipe their outputs

# API Tour: File System

- fs
- path



# Hello World

## server.js File:

```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-  
    Type': 'text/plain'});  
  res.end('Hello, world! ');  
}).listen(80);
```

Realtime apps  
Real easy

**DERBY**

All app code client and server

Real-time synced

- Responsive
- Offline
- No glue code
- SEO friendly

- `C:\src\hello> npm install derby`

# DEMO

Visit Me @



<http://twitter.com/prasoonk>



<http://prasoonk.wordpress.com>

<http://slideshare.net/prasoonk>

**Thanks!**

# Quick References

- <https://github.com/joyent/node/wiki> Github Node Wiki
- <http://nodemanual.org> - API docs, tutorials and live code samples
- <http://c9.io> Cloud9 IDE (use github id)
- <http://docs.nodejitsu.com>
- <http://howtonode.org/> Tutorials
- <https://npmjs.org/> NPM registry
- <https://github.com/WindowsAzure/azure-sdk-for-node>