



Nodeable

Using and integrating node



Introduction

- Julian Lyndon-Smith
 - Long-time (aka old) progress hacker
 - Co-founder, dot.r and whoGloo (www.whogloo.com)
- Using node since 2012
- Using progress since 1987



What is node ?



What is node ?

- Created 2009
- Server-side Javascript
 - Based on V8 js engine from Google
 - Allows for “full-stack” applications
- Event-driven
- Allows you to build scalable network applications using JavaScript on the server-side



What can you build with node ?

- Websocket servers
 - Chat server etc
- Ad servers
- Real-time data apps
- Websites
- REST applications



What node is not

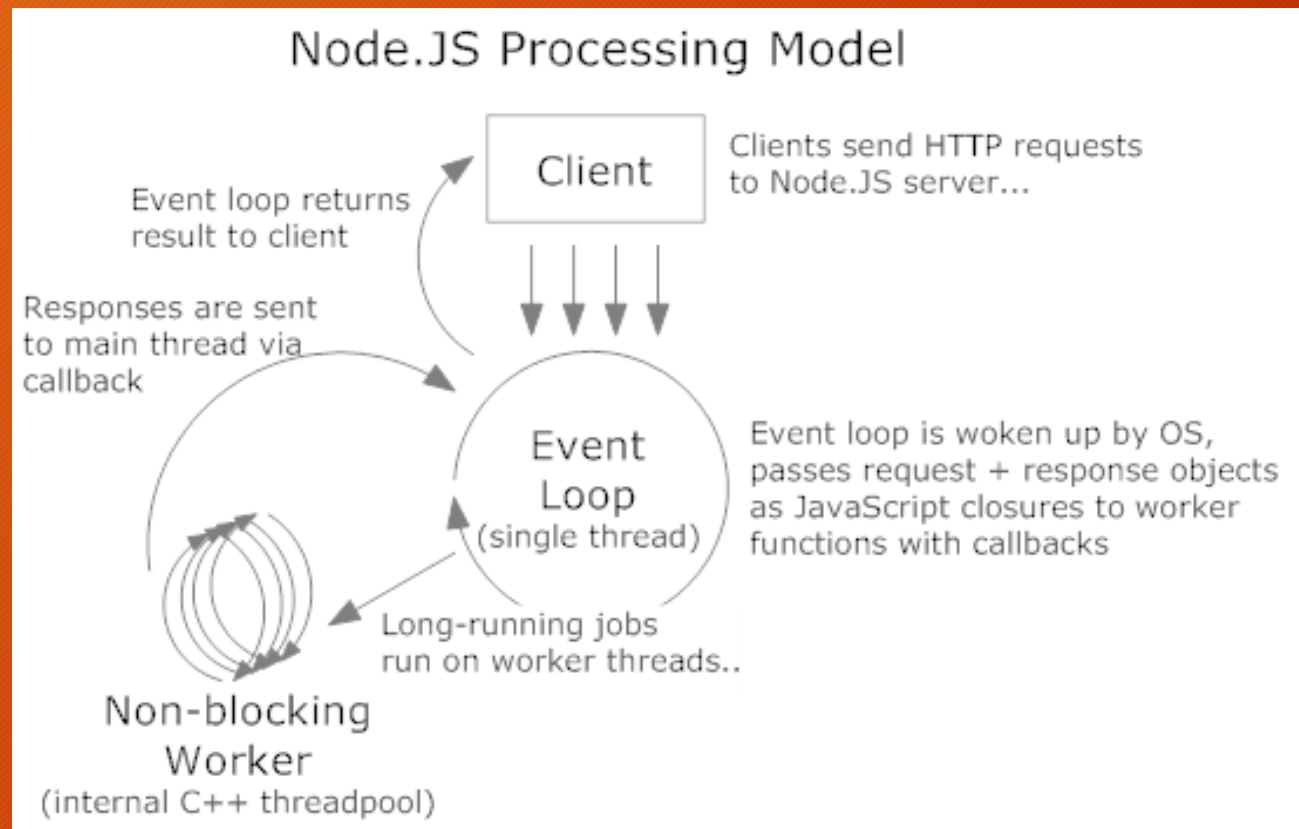
- For beginners
 - Low level, complicated
 - Especially for abl coders 😊
- Multi-threaded
 - Scalable using loadbalancers and multiple processes
- A web framework
 - There *are* packages to facilitate this



Node event loop #1



Node event loop #2



Blocking vs non-blocking

```
var content = fs.readFileSync('/src/foo.p');  
console.log(content);  
console.log("onwards and upwards");
```



Blocking vs non-blocking

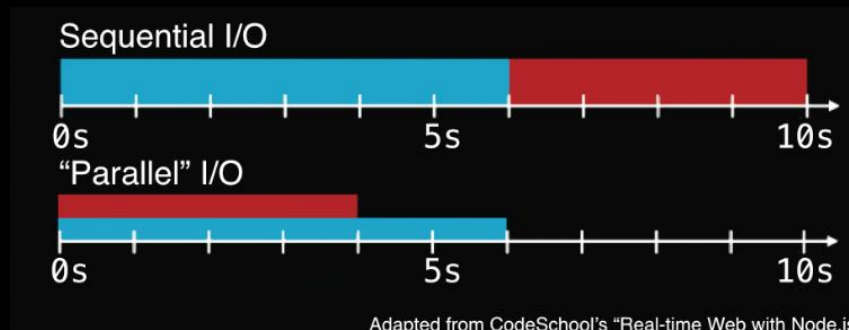
```
fs.readFile ("/src/foo.p", function(err, content) {  
    console.log(content);  
});  
  
console.log("onwards and upwards");
```



Sequential vs Paralell

Sequential vs Parallel

- Traditional web apps perform each I/O Sequentially
- With an Event Loop, they can be run in Parallel
- Since most time is wasted doing I/O, very inefficient



Example web server

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');

console.log('Server running at http://127.0.0.1:1337/');
```



Example echo server

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

var server = net.createServer(function (socket) {
  socket.write('Echo server\r\n');
  socket.pipe(socket);
});

server.listen(1337, '127.0.0.1');
```



Installing node.js

- Multi-platform
 - Windows
 - Linux
 - Etc
- Multi-architecture
 - X86
 - X64
 - Arm
 - Yes - even raspberry pi 😊



npm

- Node package manager
- <https://www.npmjs.org>
- 94 thousand packages
 - Problems with identifying “best” ones
- Our favorites
 - Express, passport, mongodb, jade, nconf, socket.io
 - ar-drone ...



Quick examples of node code & apps

- Web server
- Mongo management console
- Npm visualisation (<http://npm.anvaka.com/>)



Integration with progress

- How to link node with progress
- How to link progress with node
- Futures



Stomp

- <http://www.stomp.github.com>
- Simple Text Orientated Messaging Protocol
- Provides easy messaging interoperability between many languages, platforms and brokers
- Originally designed for scripting languages such as Ruby, Perl and Python



Stomp::Command and Control

- Opinionated
 - Makes assumptions about message format and content
 - Required queues
- Handles instructions from phoneGloo SDK to phoneGloo pbx
 - Dialplan reload
 - Sip reload
- Creates / updates configuration files
 - Large messages
 - Removes need for ftp / tftp
- Does not publish anything



Node::status updates

- Selective
 - All events from phoneGloo are captured by node
 - Not all are sent onwards to stomp
 - Prevents overload
- Secure
 - No pbx events are distributed to other listeners
 - Messages are sent through virtual host
 - No other listeners can attach to this host
- Formatted
 - All events are converted to json
 - Makes message processing a lot easier ;)



Node::status updates::data

```
{ body: 'asterisk event',  
  persistent: 'false',  
  event: 'PeerStatus',  
  privilege: 'system,all',  
  channeltype: 'SIP',  
  peer: 'SIP/baa1a29434',  
  peerstatus: 'Registered',  
  address: '192.168.99.223:5060' }
```



Integration Technologies::WebSpeed

- Links http clients to progress
- Uses web server to send request to progress
 - Progress sends data back to client
 - <http://www.progress.com/products/openedge>
 - Cross platform
- Runs standard progress 4gl programs
 - Some “extras”



Integration Technologies::WebSpeed

- Use node packages to talk to rest
 - & therefore WebSpeed
- WebSpeed programs construct json response
- Some issues
 - You may not have a WebSpeed licence ;)
 - Have to construct your own api



Demos :: lots of demos :: and code ...



Questions

