

INTRO TO NODEJS

follow along at

<http://nodejs-slides.stfn.me>

WHO IS THIS KID?

- Hi, I'm Stefan, I go here
- Junior
- Started working for CSE two years ago as helpdesk, now in a (very) part time developer role
 - You should totally use gitlab.cs btw
- Wrote a lot of JS/node this summer for my internship
 - I am *not* a node wizard

OUTLINE

1. What is node/Why node is awesome
2. Nodejs architecture overview
3. The javascript part
4. Recap + takeaways
5. Hands on time!!

WHAT IS NODE

- Asynchronous, event driven server framework
- Single threaded!
- For frontend js noobs and "unix haxors"
- Non-blocking I/O

THE PROBLEM

Servers have to handle lots of requests.

What happens when the server has to do something like this?

```
var result = database.query("kittens");  
// finger twaddling  
send(result);
```

The database query is blocking the thread from continuing!

THE SOLUTION

Nodejs uses callbacks to defer dealing with I/O until it's ready

```
database.query("kittens", function (data) {  
  // database is done and has the data ready  
  send(data);  
});  
  
// do other server things  
skynet.spawn({ "model": "T-1000" });
```

we'll explore callbacks and javascript language features later

WHY NODE IS AWESOME

Simple http server

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

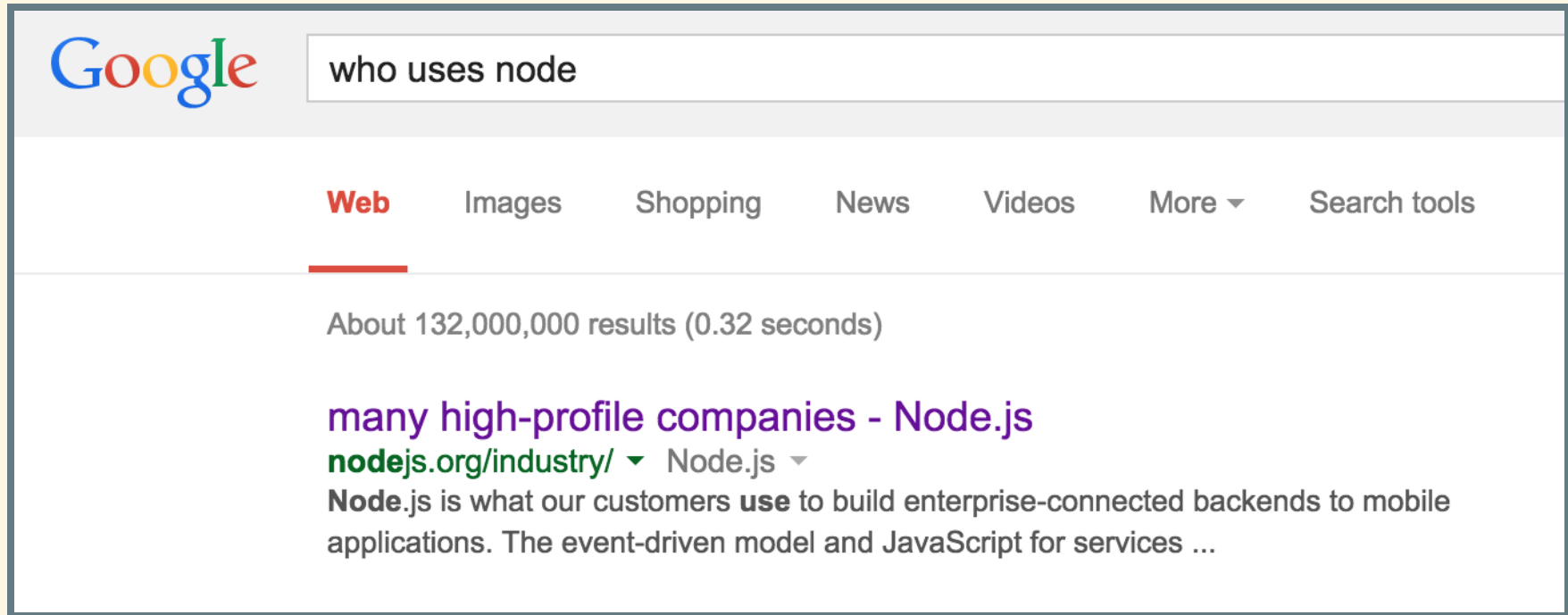
function requestHandler(request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello CSE 333\n');
}

http.createServer(requestHandler).listen(1337, '127.0.0.1');
```

WHY NODE IS AWESOME PT 2

- Very simple to set up a server
- It's Just Javascript
- Plenty of composable modules
- Very active development (the New Hotness)

WHO USES NODE?



Uber, Yahoo, Microsoft, Ebay, Cloud9 IDE, Dow Jones, LinkedIn,
The New York Times, PayPal, Mal-Wart

HOW IS NODE PUT TOGETHER?

- Google v8 engine that compiles/optimizes Javascript
- High performance parts written in C (http, libuv)
- Wrapped together with v8 apis

GOOGLE V8

- Google's open source javascript engine
- Used by google chrome
- Exists as a standalone engine
- Compiles javascript to machine code JIT
 - Slower full compile, fast JIT optimization

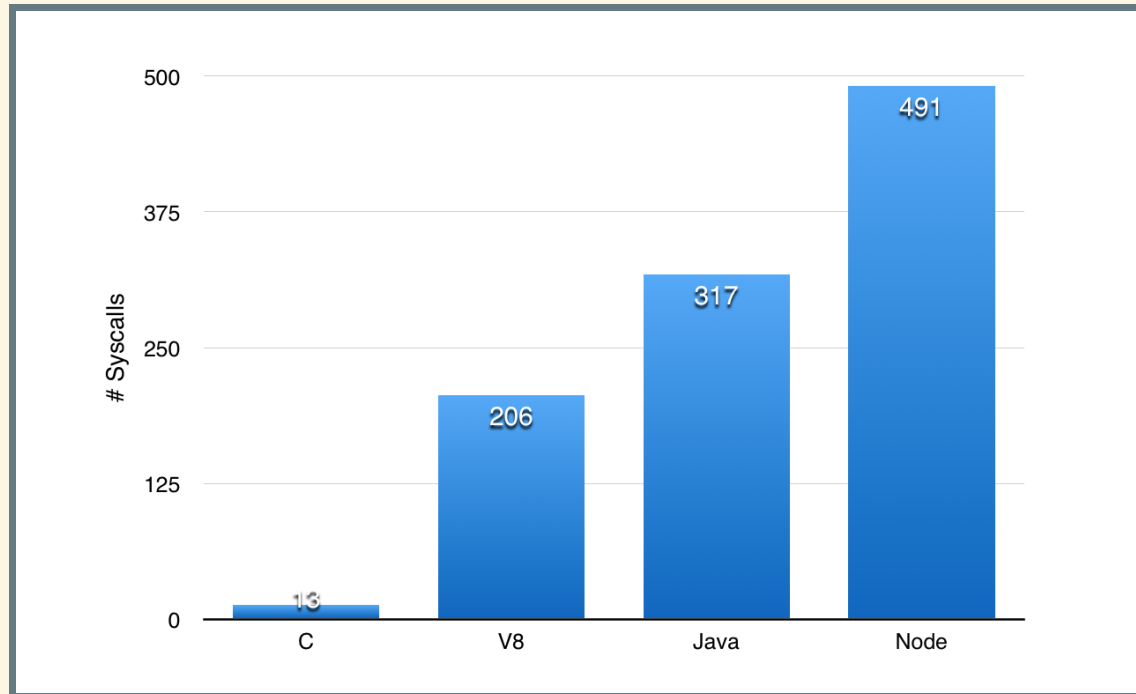
V8 BENCHMARKS

Calculate first 25,000 primes
[primes.cc](#), [primes.js](#), [timeprimes.sh](#)

```
c++:  
287107  
real 0m1.593s  
user 0m1.589s  
sys 0m0.002s  
  
v8:  
287107  
real 0m3.676s  
user 0m3.662s  
sys 0m0.012s
```

SYSCALL OVERHEAD

Hello world in C, v8, java and node



Remember that v8 and node are both compiling js!

PURE C MODULES

Nodejs has a few components written in C for performance

Http parser:

https://github.com/joyent/node/tree/master/deps/http_parser

libuv, platform independent async I/O:

<https://github.com/libuv/libuv>

freaking awesome logo btw



OUTLINE

1. What is node/Why node is awesome
2. Nodejs architecture overview
3. **The javascript part**
4. Recap + takeaways
5. Hands on time!!

WHY JAVASCRIPT TOTALLY RULES



JAVASCRIPT OVERVIEW

- Aka ECMAScript
- Standardized scripting language for the web
- Designed in 10 days
- Intended to mutate DOM through browser APIs

```
// Grab every div on the page
var divs = document.querySelectorAll("div");
// Loop through each element
for (var i = 0; i < divs.length; i++) {
  // Change the background image
  divs[i].style.backgroundImage = "url('http://nodejs-slides.stfn.me/files/kit";
}
```

But now it does so much more!

WHAT IS A CALLBACK

Callbacks are functions that are passed as parameters to other functions, and then called within those functions

```
function helloCaller(helloFn, name) {  
  helloFn(name);  
}  
  
function hello(name) { console.log("hello " + name); }  
function swedishHello(name) { console.log("hej " + name); }  
  
helloCaller(hello, "CSE 333");  
helloCaller(swedishHello, "CSE 333");
```

THE GOSPEL OF NODE

- Javascript is one of the most popular languages in the world
- Lots of people with knowledge of javascript
- Ergo, lots of people familiar with async + callbacks
- Nodejs designer wanted a language with callbacks + event loop concept
- Google V8 ayy Imao
- And thus nodejs was born

EVENT LOOP

- Async calls enqueued on event loop with a callback
- Node single-thread will dequeue and run callback
- When there's nothing left to do, node will sleep
- Everything should be async as possible!

SIMPLE SERVER EXAMPLE AGAIN

```
var http = require('http');
function requestHandler(request, response) {
  response.writeHead(200, {'Content-Type': 'text/html'});
  response.end('<h1>MLP Fan Site</h1>');
}
http.createServer(requestHandler).listen(1337, '127.0.0.1');
```

- Each request to the server is enqueued
- Uses `requestHandler` as the callback

EX15

Write the contents of a file to a socket

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

var host = process.argv[2];
var port = process.argv[3];
var filename = process.argv[4];

var socket = new net.Socket();
socket.connect(port, host, function() {
  fs.readFile(filename, function (error, data) {
    socket.write(data);
    process.exit();
  });
});
```

```
node ex15.js stfn.me 5000 filename
```

IT'S SERVER-SIDE JAVASCRIPT!

You can do bad things

```
var http = require('http');
var sys = require('sys');
var spawn = require('child_process').spawn;

http.createServer(function (req, res) {
  var ls = spawn('ls', ['-lh', '/']);
  ls.stdout.on('data', function(data) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end(data.toString('ascii'));
  })
}).listen(3555, '127.0.0.1');
```

POPULAR MODULES

- <http://expressjs.com/> -- minimalist web framework
- <http://gruntjs.com/> -- task runner
- <http://yeoman.io/> -- project scaffolding/generator
- <http://browserify.org/> -- frontend module system/bundling

...and many more!

OTHER CRAZY THINGS YOU CAN DO

- Program arduino (node-serialport)
- Write standalone apps with html + js (node-webkit)
 - Probably shouldn't do this
- Write node plugins in c++ (node-gyp)
- Feel smug + hipster

TAKEAWAYS

Nodejs...

- is fast and simple
- lets the OS worry about concurrency
- uses a popular language
- has plenty of support + ""good"" docs
- isn't php

DEMO TIME!

chatserver.js

To connect, open terminal and type

```
nc stfn.me 5000
```