Node.js

How JavaScript is Changing Server Programming

Tom Hughes-Croucher

@sh1mmer

This is how we roll

1. Server-Side JavaScript Overview
2. Introduction to Node.js
3. The Node.js ecosystem
4. Getting started

Server Side Javascript IS SO AWESOME

MY ENTIRE PRESENTATION IS IN

COMIC SANS

AND YOU WILL STILL LOVE ME AT THE END

Why SSJS?

JavaScript programmers

3 > 2 > 1

Massive Code base of YUI and other JS libraries

I heard some people use this thing called jQuery,
but I'm not convinced it'll catch on

Laziness or "I'm sick of writing stuff twice"

I could have said efficiency, but I think we all secretly long to lounge around in our y-fronts.

Progressive Enhancement is free*

Remember WWCD (What Would Crockford Do)

*close enough

TL;DR:

SSJS is Awesome

Like a Unicorn riding a Narwhal

Why now?

1. Professionalism

Douglas Crockford Facts

Both douglas and crockford are reserved words in JavaScript.

"Yahoo!'s corporate motto is: Don't be eval()."

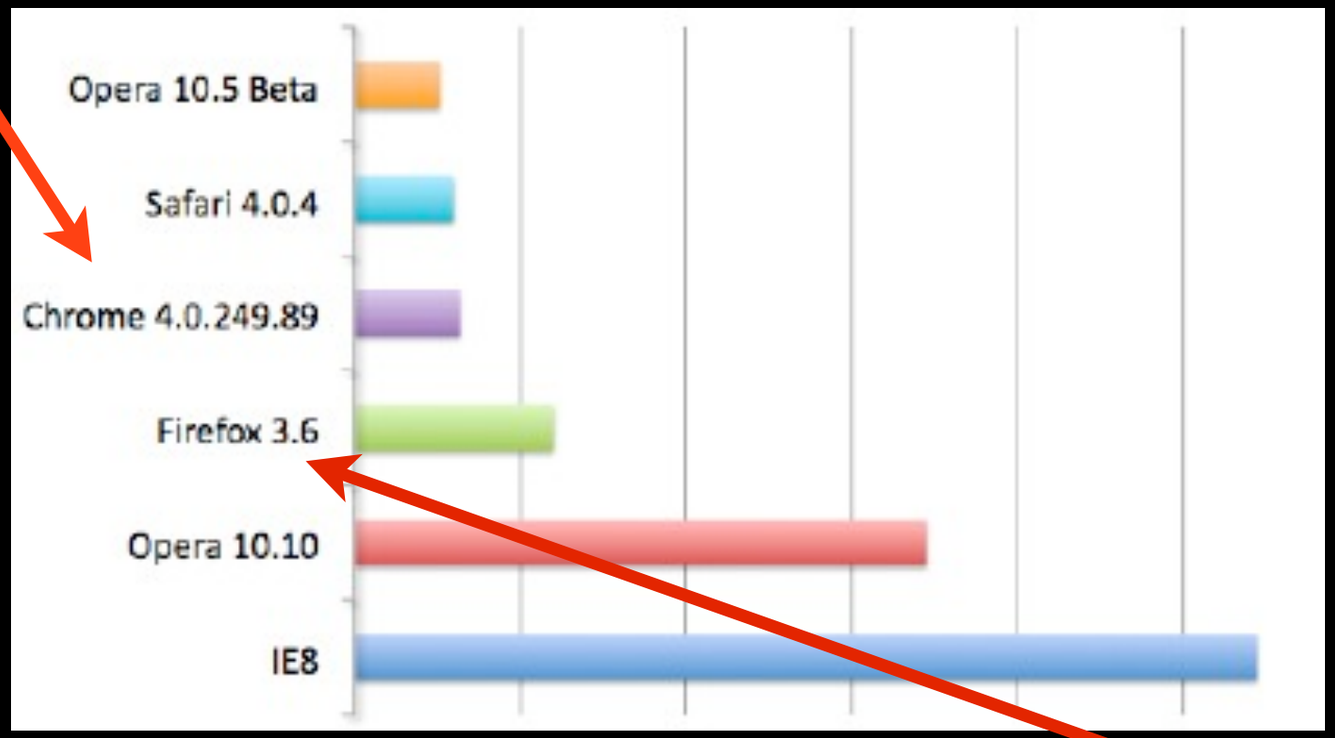
"Doug Crockford's JavaScript is so strict, that uncaught exceptions would trigger global thermonuclear war"

2. JavaScript Runtimes

Runtimes

- V8 (Google), C++
- Spider Monkey (Mozilla), C++
- Rhino (Mozilla), Java
- JavaScript Core (Apple), C++

V8



Spider Monkey

JavaScript Performance

Node.js

- Server-side JavaScript process
- Uses V8
- Non-blocking
- Event Driven
- CommonJS module format

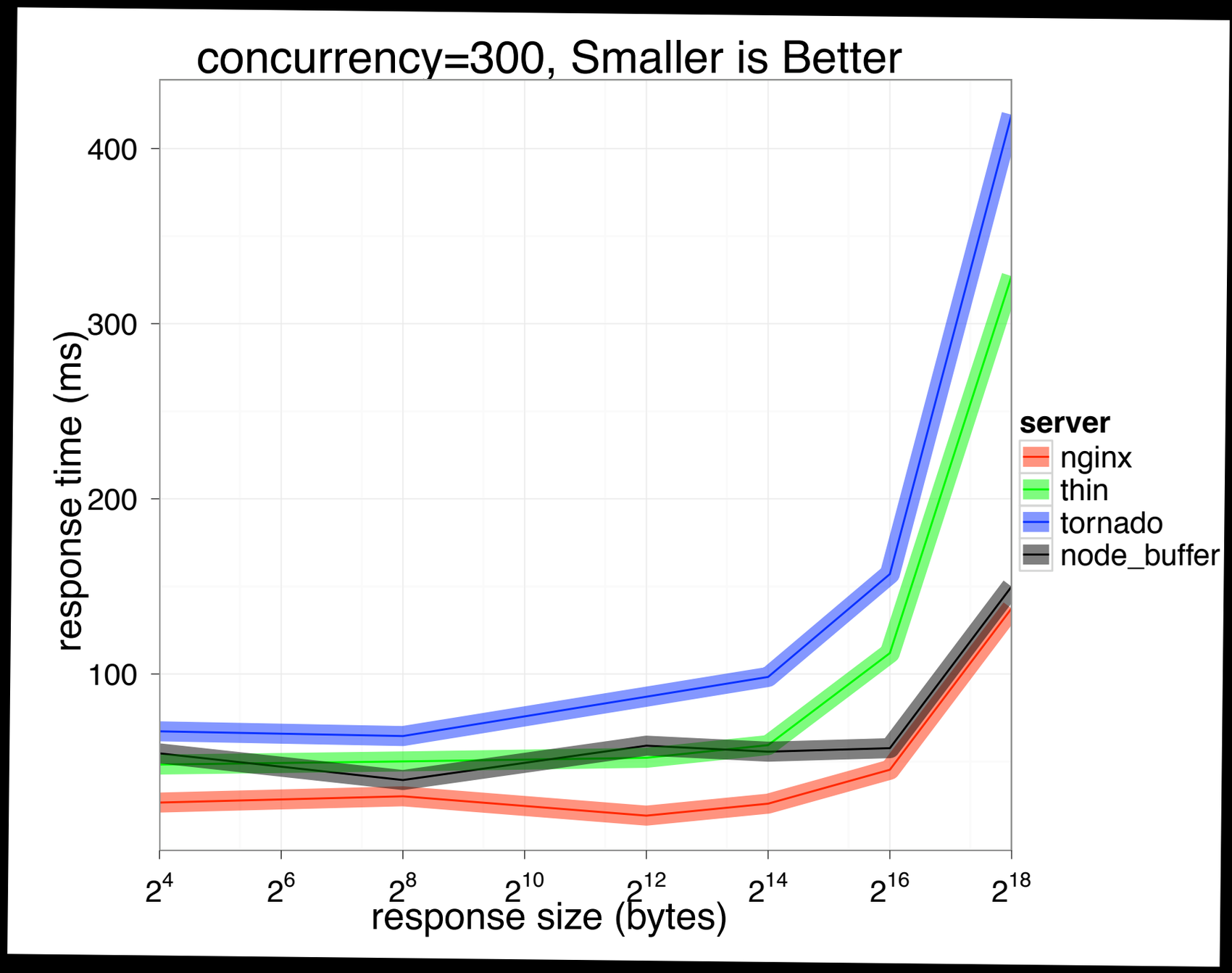
Node.js

- Script process
- Uses
- Non-blocki
- Event Driven
- CommonJS module for

AWESOME!

Node is

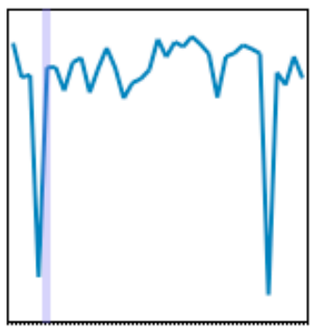
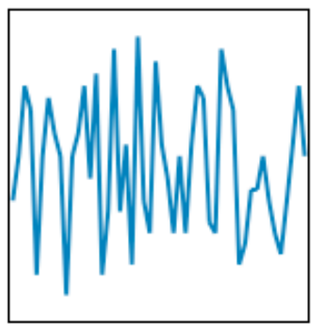
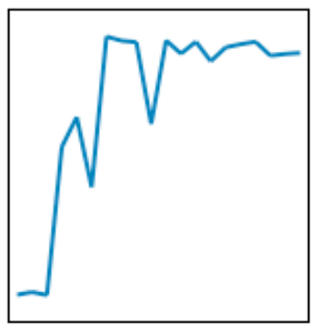
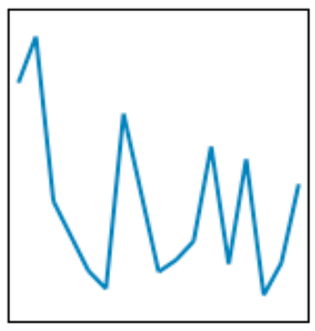
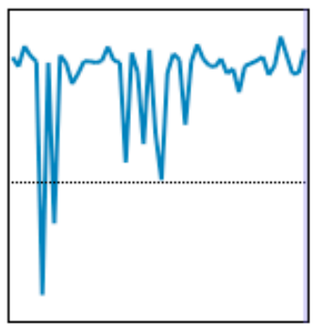
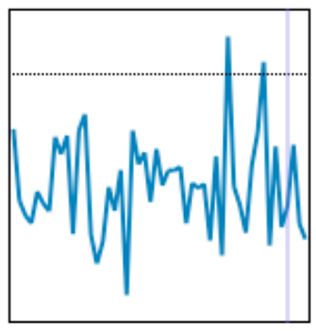
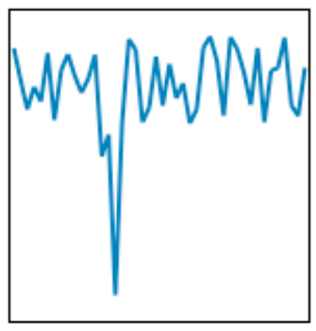
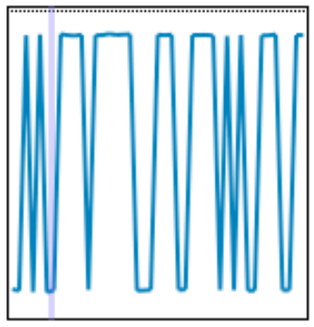
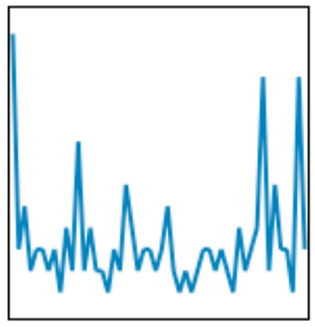
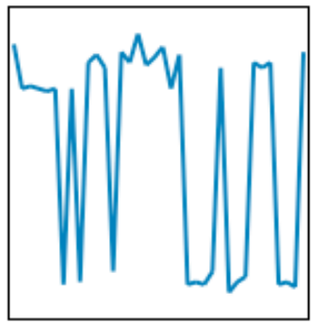
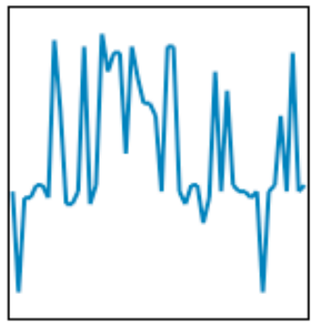
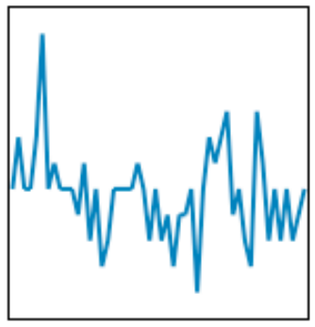
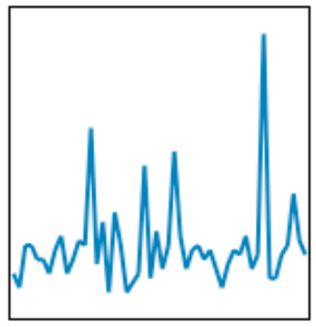
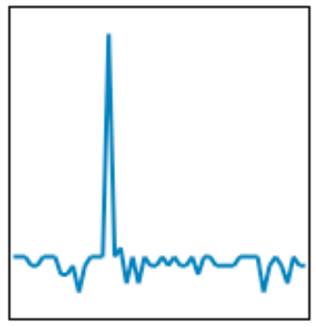
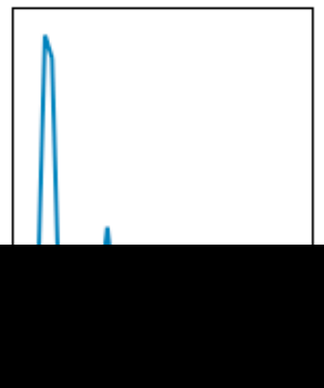
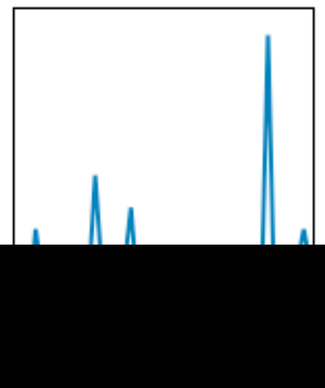
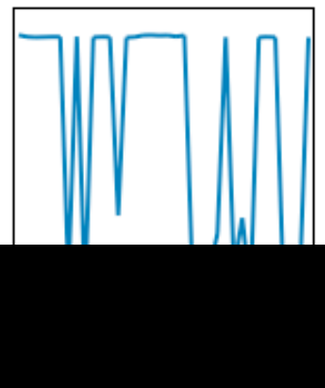
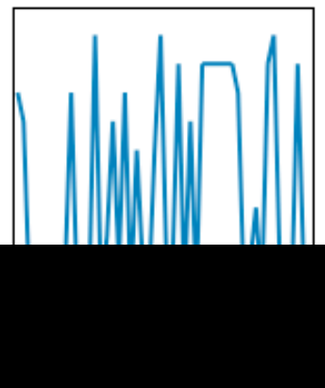
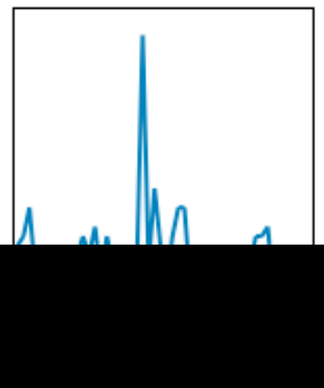
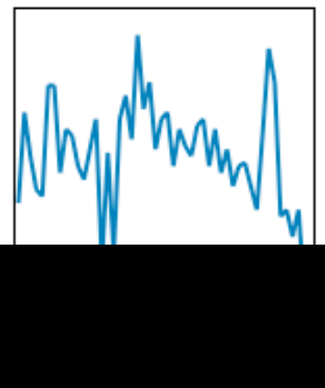
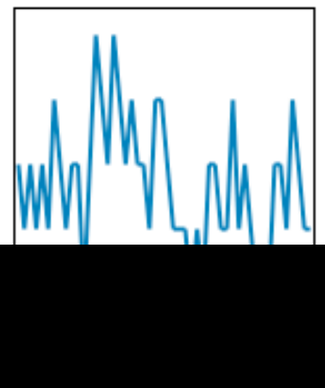
this fast


concurrency=300, Smaller is Better
response time (ms)
400
300
200
100
2^4
2^6
2^8
2^{10}
2^{12}
2^{14}
2^{16}
2^{18}
response size (bytes)
server
nginx
thin
tornado
node_buffer

Node.js Perf: overview

(classic view)

[overview | ab | page-static_http_server | timers | process_loop]

Solaris Nevada	Ubuntu Karmic	FreeBSD AMD64	FreeBSD	Linux AMD64	Linux Xeon	OSX
						
Legend: t	Legend: t	Legend: t	Legend: t	Legend: t	Legend: t	Legend: t
r2fa4de001c1: 4799.55 req/sec +/- 0.00 137.48	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline
						
Legend: t	Legend: t	Legend: t	Legend: t	Legend: t	Legend: t	Legend: t
r7a2e6d674a9: 98.00 ms +/- 0.00 3760.57 ms	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline	move mouse over graph	Shift-click to place baseline
						

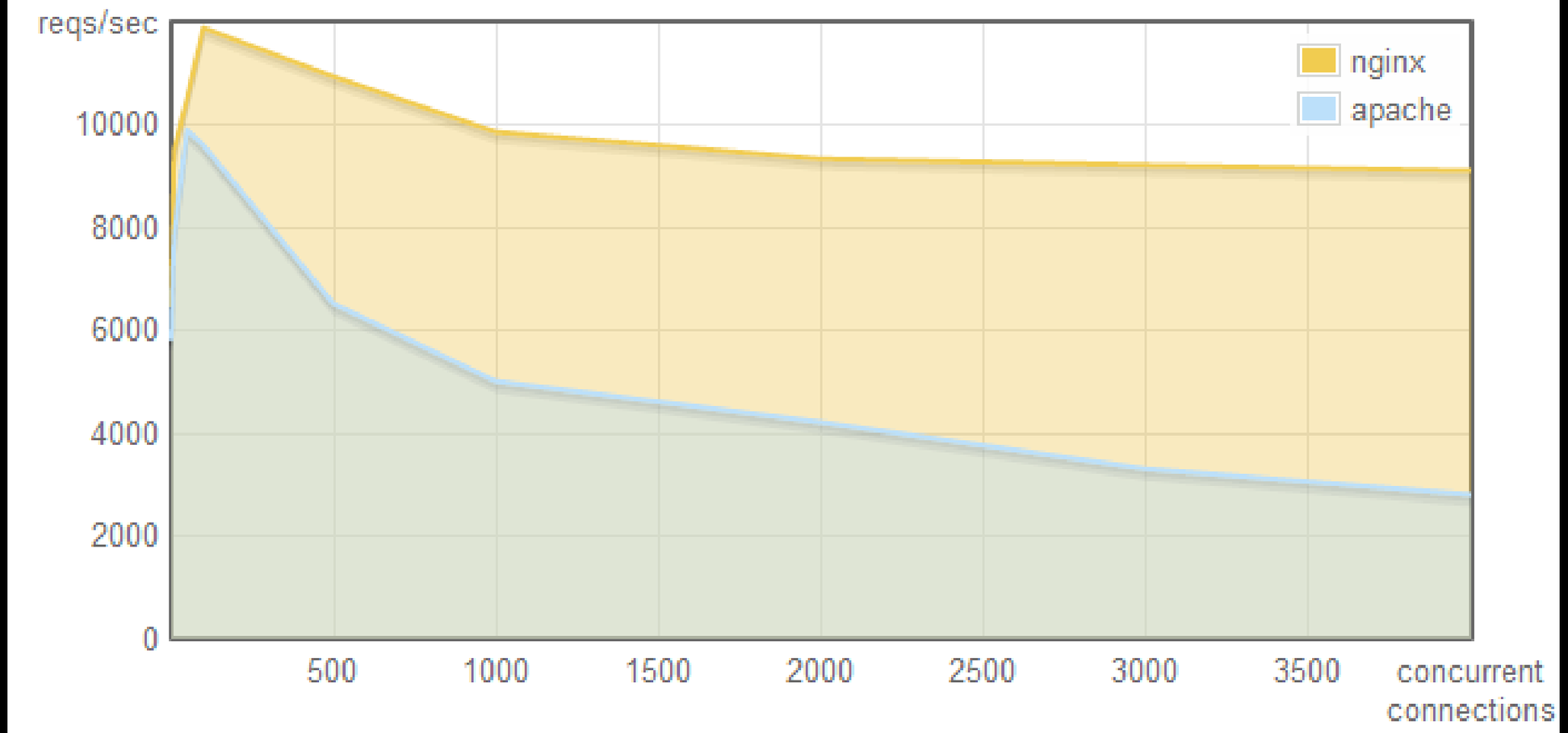
Why non-blocking matters

```
var result =
db.query("select * from T");
// use result
```

What are we waiting for?

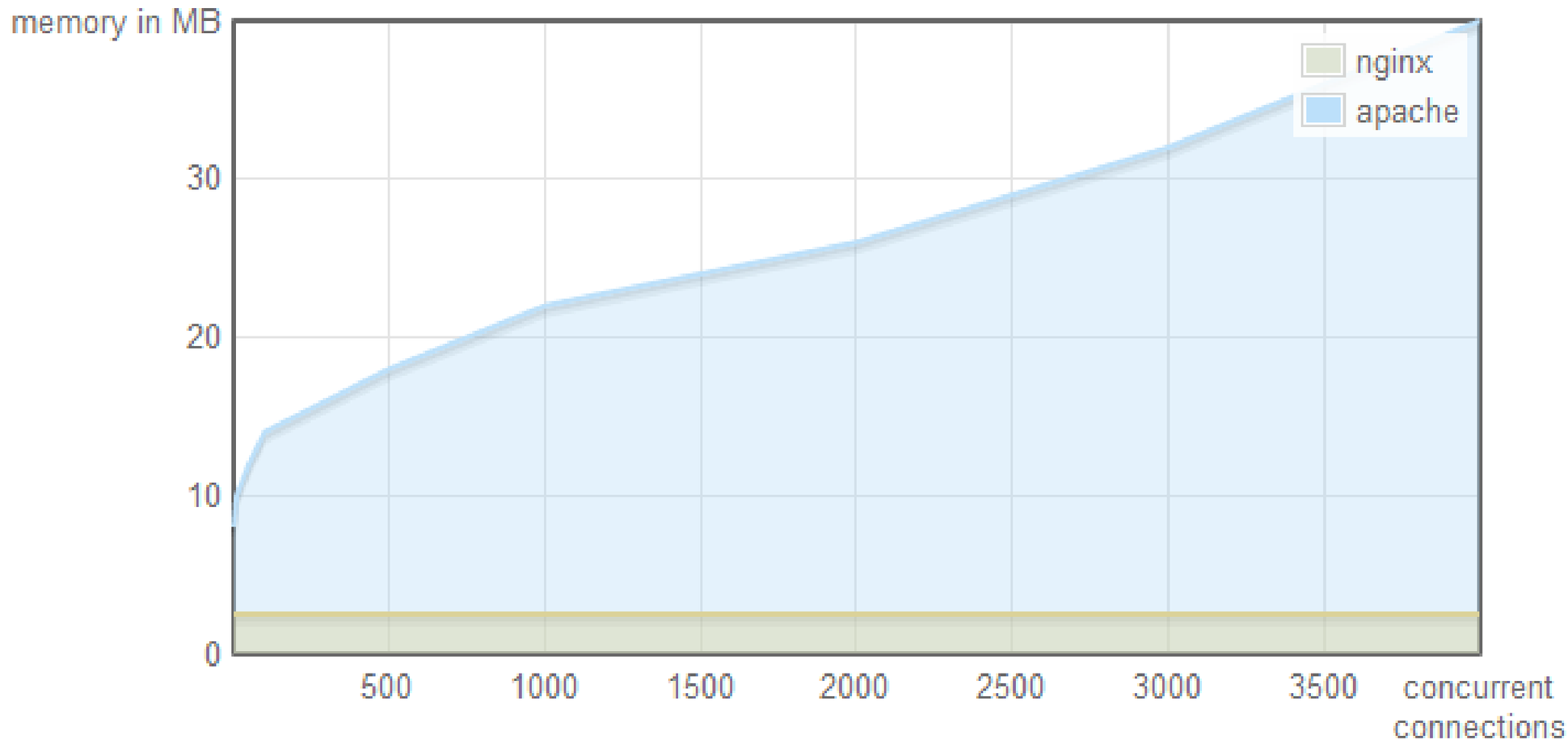
Event Loop vs. Threads

concurrency × reqs/sec



http://blog.webfaction.com/a-little-holiday-present

concurrency × memory



http://blog.webfaction.com/a-little-holiday-present

Node Basics

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8124, "127.0.0.1");
console.log('Server running at http://127.0.0.1:8124/');
```

```
var http = require('http');
//include the http library
```

```
http.createServer(function (req, res) {

}).listen(8124, “127.0.0.1“);

//create an http server
//when ‘stuff’ happens call this anonymous function
//listen on port 8124 of the IP 127.0.0.1
```

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

//when 'stuff' happens my function fires
//I get a request object and a response object
//I write to the response object header
//HTTP status 200 and content-type 'text/plain'
//close the response with the body:
//Hello World
```

```
console.log('Server running at http://127.0.0.1:8124/');
//write Server is running at http://127.0.0.1:8124/
//to the console
```

Node Ecosystem

github SOCIAL CODING

Home Pricing

ry / node

Source Commits Network Downloads Wiki Graphs

Home | Edit | New

Modules

Adding to this page

When you add a framework to this page, have a look at how others have done so, it should be a single item, with a link to the project's source code, and a short description (one line after formatting has been applied).

If you see a module without a description, feel free to edit the page and add it in, any contributions are appreciated.

When you edit this list, also add your module to http://wiki.github.com/ry/node/library-compatibility so that users will know which versions of Node it was tested with.

Modules

Web frameworks

Routers

- (fab) — A modular and concise async web framework for node.js
- Nerve — Microframework with simple array-based syntax for defining an app on top of node. (node.JS 0.1.30)
- rowan — A hierarchical microframework of reusable controllers and modular app logic.
- scylla — Create simple one object applications by doubling up method names as URL-matching patterns (node.JS 0.1.95)
- vroom — A simple resource oriented web framework built on top of Node.js (November 2009, node.JS 0.1.16)
- Picard
- simplex (October 2009, node.JS 0.1.14)
- Pipe-Layer — Asynchronous HTTP router.

Frameworks

- chain — An evented convention for building Node Applications (Stopped Development, for ejsgi)
- coltrane — A try at a higher level library/framework for node.js web development (July 2009, node.JS 0.1.1)
- djangode — A framework that borrows some useful concepts from Django. (March 2010, node.JS 0.1.30)
- express — A robust feature rich web development framework **inspired by Sinatra**
- Geddy — A hackable Web-app development framework similar to Merb/Rails/Pylons/Django
- js.io — Javascript Networking Library for building real-time web applications. Also see JS.io
- MVC.js — A simple MVC framework for **REST**-ful applications, with integrated **dependency-injection** support.
- nodemachine — A port of WebMachine to Node.js
- pintura — REST-based web framework/middleware stack built for Ajax-style JSON-driven applications.
- jimi — Building upon djangode, this framework adds a modular app structure.

Middleware

- cascade — Sequentially attempts multiple middleware apps (JSGI).
- compress – Gzip compresses (using node-compress) the response when appropriate based on request headers. (JSGI)
- media – Performs content type negotiation (per RFC2616) delegating to appropriate media handler. (JSGI)
- csrf – Checks HTTP request for possible cross-site request forgery, flags dangerous requests. (JSGI)
- xsite – Handles JSONP, window.name, and cross-origin XHR (CORS). (JSGI)
- rewriter – Rewrites defined paths to other paths. (JSGI)
- static – Static file handler using asynchronous streaming. (JSGI)
- error – Catches uncaught errors and converts to appropriate HTTP error responses. (JSGI)
- conditional – Handles conditional HTTP requests (If-Modified-Since, etc.) (JSGI)
- http-params – Converts HTTP parameters http- to headers. (JSGI)

- http-params – Converts HTTP parameters http- to headers. (JSGI)
- auth – Handles Authentication (HTTP and cookie based). (JSGI)
- commonlogger – A logger of HTTP requests. (JSGI)
- contentlength – Sets Content-Length header. (JSGI)
- head – Handles HEAD requests (stripping body). (JSGI)
- redirect – Redirects to other URLs (JSGI)
- urlmap – Maps to different apps by path/URL (JSGI)
- extension – Transforms .extension to a RESTful Accept header (JSGI)

Other

- node-elf-logger — Configurable HTTP logging library following the W3C Extended Log File Format specification
- JSGI-Node — Asynchronous JSGI 0.3 Adapter for Node
- node-mime — Utility module for mime-type lookups
- node-cgi — CGI adapter kludge (replaces Node's fast and famous event-based HTTP library)

Database

- awesome — a Redis clone in node.js
- cradle — a high-level, caching, CouchDB client
- JSLINQ — Clean and simple port of Microsoft's LINQ to node.js (and the browser)
- node-couch — a CouchDB connector
- node-couchdb — A full API implementation
- node-couchdb-min — Light-weight client with low level of abstraction and connection pooling.
- node-dirty — A key value store for node.js that is simple, fast & dirty.
- node-fleet — a FleetDB Client
- node-memcache — a memcache client base on libmemcached
- node-mongodb — Basic MongoDB client implementation in JS/C++
- node-mongodb-native — A pure javascript driver for MongoDB
- mongoose — Mongoose is a javascript library that makes working with MongoDB a breeze.
- node-poormansmysql — Asynchronous MySQL driver for node.js using the mysql command-line tool
- node-mysql — Pure Javascript MySQL async driver
- node-mysql-libmysqlclient — MySQL synchronous bindings based on libmysqlclient
- node-riak — Riak client library
- node-sqlite — Bindings for SQLite3. Interface conforms to the HTML5 Web SQL API.
- NoSQLite – A zero-config SQLite wrapper written in CoffeeScript

- node-sqlite — Fast asynchronous driver: New evented Node.js look, same great SQLite3 taste
- node-tyrant — An implementation of the Tokyo Tyrant network protocol for the Node.js
- node.dbslayer.js – Interface to DBSlayer (MySQL)
- node_postgres — Beginning of bindings to libpg
- noodb — A simple and small file backed javascript key-value store
- persistence — Multi-backend database/nosql system. Backends: Sqlite3, Postgres and Memory.
- perstore — JavaScript persistence/object store with pluggable storage based on the W3C DB API
- postgres-js — Postgres protocol implemented in pure JS
- redis-node-client — Redis Client by Fictorial
- riak-js — Riak Javascript client (works on node v0.1.30+)
- node-tokyocabinet — Tokyo Cabinet binding
- node-migrate — Migrate – A database agnostic migration system for Node.js

Templating

- asyncEJS — Asynchronous implementation of embedded JavaScript
- haml.js — Faster / more compliant implementation of Haml
- haml-js — Server side html generation using javascript. Parses haml templates and renders html.
- jazz — A readable template language for node.
- JSON Template — Minimal but powerful template language with multiple implementations. This is the CommonJS version, tested on Node
- xmlbuilder.js — An xml builder in Javascript inspired by Ruby's Builder, Markaby, and Erector.
- jm — Another Builder/Markaby/Erectory clone in javascript.
- less.js — official port of Less to javascript/node.
- Mu — A Mustache engine that compiles templates into very fast rendering functions. Also streams the rendering process.
- node-template — Fast and light cached templates.
- sass.js — Parses Sass templates and renders css.
- template.node.js — A light, fast, cached template module for node.
- tmpl-node — a feature-rich template module for node.js
- jsdom — pure js implementation of the dom level 1 with some browser augmentation. Level 2 and 3 are being considered.
- bind-js — a simple templating engine for node.js that smiles back.
- normal-template — Normal templates are simple, yet powerful. They are safe, usable in non XML/HTML contexts and portable to any programming language.

- nun — Totally asynchronous non-blocking template engine for node.js
- node.magic_dom — A DSL for building HTML in node.js, similar to Python's Stan
- strobe-templates — An asynchronous templating engine with syntax like Django's

Package Management Systems

- kiwi — Feature rich, fast, node.js package management system — server sponsored by Slicehost
- npm — A node package manager that uses CommonJS-compatible package.json files, written in asynchronous javascript.
- mode — Distributed Node module repository. Uses a github repository to contain info about installable modules.
- seed — Universal package manager for CommonJS. Includes command line tools and async server

OpenSSL / Crypto / Hashing

- hashlib — Fast hashing module, written in C/C++, supports: md4, md5, md6, sha, sha1, sha256, sha512
- brainfucker's node-base64 — C++ base64 lib
- pkrumins's node-base64 — C++ base64 lib that actually works
- node.bcrypt.js — C/C++ bcrypt lib
- node-crypto — OpenSSL based Hashing, Signing and Verifying
- node-oauth — OAuth client (1 and 2)
- sasljs — Gsasl wrapper to performs server-side SASL authentication.

TCP / IP

- node-httpclient — Node HTTP Client (gzip, https, cookies etc.)
- node-protocol — A framework for implementing protocols.
- node-smtp — Implementation of the SMTP protocol in Node
- NodeFTPd — Node FTP Server
- xmpp.js — Library for implementing XMPP server components with Node
- node-http-digest — Node HTTP Client with support for Digest Authentication
- request — Simple HTTP client library.
- node-xmpp — Node XMPP library
- node-snpp — Node SNPP server library

RPC

RPC

- bertrpc
- jsonrpc
- xmlrpc-c — Simple XMLRPC Client

Web Sockets & Ajax

- Comet LongPollingBuffer — A Library to simplify the server side of Comet AJAX long polling
- Faye — Bayeux protocol Comet client and server for Node.js and Rack
- Socket.io — WebSocket-compatible server and client with fallback for legacy browsers
- node-XMLHttpRequest
- node.websocket.js — WebSocket-compatible server.
- node.ws.js — A basic Web Socket server with interface similar to tcp.createServer(...)
- nodejs-http-websocket — A websocket server on top of the http server.
- node-websocket-server — Another websocket server on top of the http server.
- Restler — Simplified REST client for Node.js

Testing / Spec Frameworks

- espionage — A mock/stub framework using the test spy pattern.
- expresso — TDD framework by the author of JSpec
- exemplor.js — A port of exemplor with Node goodness.
- jasmine-node — integration with Pivotal's Jasmine Spec framework
- jspec — Feature Rich BDD Testing Framework
- minitest.js — Light-weight & simple testing framework designed specially for testing asynchronous code.
- mjsunit.runner — Command line mjsunit runner which provides an easy way to hook into mjsunit and start running tests immediately.
- node-assert-extras — Additional high level asserts
- node-async-testing — Simple testing (hopefully)
- node-stories — Given/When/Then integration awesomeness for Node.
- nodeunit — Quick and easy unit testing, based on a simplified version of the QUnit API
- ntest — another unit testing framework
- spectacular — for testing
- Speks — A specification framework for your node-code

- Vows — asynchronous behaviour-driven development for node.js
- Willful — a simple spec'ing library for Node.
- Gently — A node.js module that helps with mocking and behavior verification.

Wrappers

- hxNode — haXe wrappers for node

Parsers

XML

- libxmljs — Bindings to libxml2
- node-xml — An xml parser for node.js
- node-xml2object — Converts XML to a JS object, using the node-xml module
- sax-js — SAX-like parser in pure JS
- node-expat — Fast SAX parser binding to expat

Command Line Option Parsers

- optparse-js — Option Parser in JS
- trollopjs — Another option parser
- js-opts — Another simple command line option parser

Parser Generators

- canopy — PEG parser compiler for JavaScript
- jison — A parser generator written in JavaScript; similar to Bison for C
- PEG.js — Parser Generator for JavaScript
- inimino's PEG — A PEG Packrat Parser Generator for JavaScript
- jparse — A parser combinator for javascript based on Packrat parsers and Parsing expression grammars
- OMeta Javascript compiler

Other

- node-discount — C markdown parser "discount" bindings
- node-csv — Efficient Evented CSV Parsing.

- Vows — asynchronous behaviour-driven development for node.js
- Willful — a simple spec'ing library for Node.
- Gently — A node.js module that helps with mocking and behavior verification.

Wrappers

- hxNode — haXe wrappers for node

Parsers

XML

- libxmljs — Bindings to libxml2
- node-xml — An xml parser for node.js
- node-xml2object — Converts XML to a JS object, using the node-xml module
- sax-js — SAX-like parser in pure JS
- node-expat — Fast SAX parser binding to expat

Command Line Option Parsers

- optparse-js — Option Parser in JS
- trollopjs — Another option parser
- js-opts — Another simple command line option parser

Parser Generators

- canopy — PEG parser compiler for JavaScript
- jison — A parser generator written in JavaScript; similar to Bison for C
- PEG.js — Parser Generator for JavaScript
- inimino's PEG — A PEG Packrat Parser Generator for JavaScript
- jparse — A parser combinator for javascript based on Packrat parsers and Parsing expression grammars
- OMeta Javascript compiler

Other

- node-discount — C markdown parser "discount" bindings
- node-csv — Efficient Evented CSV Parsing.

- Vows — asynchronous behaviour-driven development for node.js
- Willful — a simple spec'ing library for Node.
- Gently — A node.js module that helps with mocking and behavior verification.

Wrappers

- hxNode — haXe wrappers for node

Parsers

XML

- libxmljs — Bindings to libxml2
- node-xml — An xml parser for node.js
- node-xml2object — Converts XML to a JS object, using the node-xml module
- sax-js — SAX-like parser in pure JS
- node-expat — Fast SAX parser binding to expat

Command Line Option Parsers

- optparse-js — Option Parser in JS
- trollopjs — Another option parser
- js-opts — Another simple command line option parser

Parser Generators

- canopy — PEG parser compiler for JavaScript
- jison — A parser generator written in JavaScript; similar to Bison for C
- PEG.js — Parser Generator for JavaScript
- inimino's PEG — A PEG Packrat Parser Generator for JavaScript
- jparse — A parser combinator for javascript based on Packrat parsers and Parsing expression grammars
- OMeta Javascript compiler

Other

- node-discount — C markdown parser "discount" bindings
- node-csv — Efficient Evented CSV Parsing.

- node_pcap — Network packet capture and analysis using libpcap
- node-promise — Robust promises for node.js, includes promise utilities and promise-based I/O library
- node-prowl — A module that allows you to send push notifications to your iPhone through the Prowl API
- node-resque — Resque redis-backed job queue workers in node.js
- node-sandbox — A rudimentary javascript sandbox for use with node.js
- node-s3 — App for basic Amazon Web Services S3 administration (upload files, bucket admin, etc.)
- node-solr — Solr module for Node
- nodestalker — A beanstalkd client for node
- node-stomp — A basic STOMP client.
- node-taglib — Beginnings of bindings to taglib
- node-ugly — Allows to run PHP code from within node.js
- node-uneval — Adds uneval() support to node.js
- node-worker — An implementation of the WebWorker API for node.js
- node-yui3 — Use the YUI 3 JS library from node.js
- node-yql — A YQL (Yahoo Query Language) module for Node.js
- nshtools — a high level shell scripting library for file and operating system chores
- soda.js — Asynchonous JavaScript module loader for client-side and Node.js
- Step — Tool for grouping and chaining asynchronous callbacks, based on flow-js
- async — Comprehensive async map/reduce and flow control (parallel, series, waterfall, auto…) module
- Task.node Simple task manager similar to Rake, Scons and others.
- tasks.js Tasks for Node (like Rake!) with a very JavaScripty API.
- temp — Temporary files and directories
- tweetstream — Stream like API for twitter's HTTP streaming interface.
- TwitScript — A port of Twython to Node.js (Twitter API Library)
- user-agent — user agent string parser
- uuid — Efficient bulk UUID generation and caching via command-line OSSP uuid.
- uuid-2 — Fork of above, works on OS X and fixes bug.
- uuidjs — Simple UUID generation binding to libuuid.
- vargs — practical variable argument handling.
- Wheres Waldo — track locations of users in redis
- Wrench-JS — Useful Node.js operations (recursive directory operations, etc)
- yaml — CommonJS YAML parser
- node-ncurses — An ncurses binding for node.js
- node-terminal — A simple terminal module for ansi control codes.
- swirl-node — A simple EC2 client

NPM

Node Package Manager

npm install mustache

NPM is written in JavaScript!

```
// kludge until this is normal.
if (!process.EventEmitter.prototype.on) {
  process.EventEmitter.prototype.on =
process.EventEmitter.prototype.addListener
}
var path = require("path")
if (!process.execPath) {
  process.execPath = path.join(process.installPrefix, "bin",
"node")
}

var npm = exports
  , set = require("./lib/utils/set")
  , get = require("./lib/utils/get")
  , ini = require("./lib/utils/ini")
  , log = require("./lib/utils/log")
  , fs = require("fs")

npm.commands = {}
npm.SHOULD_EXIT = true

try {
  var j = JSON.parse(fs.readFileSync(path.join(__dirname,
"package.json"))+"")
  npm.version = j.version
} catch (ex) {
  log(ex, "error reading version")
  npm.version = ex
}
```

node-repl

Interactive JavaScript terminal

Which libraries to use?

Mustache.js

twitter

Home Profile Find People Settings Help Sign out

What's happening?

140

in Portland, OR

Latest: @jcleblanc I think selling or drinking Natty Ice is probably a capital offense in the Microbrew capitol of America. about 2 hours ago

Tweet

Home

frybread_thief the second best thing about my hike after the bear sightings was not seeing a single person from start to finish.
less than a minute ago via Tweetie for Mac

kneath This always kills me on posterous sites http://yfrog.com/newixvj
2 minutes ago via Tweetie for Mac

jtaby ProTip: Hold down option and click on the split view button to get vertical splits. (indirectly via @jeff_larmarche)
3 minutes ago via Tweetie for Mac

segdeha Continuous deployment at @Digg: http://is.gd/dCEbT
4 minutes ago via Tweetie for Mac

techglance CalTech Awarded $122 Million to Create Fuel From Sunlight http://tcrn.ch/cpNpao
7 minutes ago via dlvr.it

kentbrew @rckenned: yes, here too.
8 minutes ago via web in reply to rckenned

brainpicker SocialVest – philanthropy, without the fuss. Smart new model for donations http://bit.ly/dsSIpj
9 minutes ago via TweetDeck

sh1mmer
4,592 tweets

473 following **1,517** followers **181** listed

Twitter · for · BlackBerry
n. The Twitter branded app for BlackBerry.

Home

@sh1mmer

Direct Messages 814

Favorites

Retweets

Search

Lists

conversationlist

mybrands

thoughtleaders

New list | View all

Trending: San Francisco

Change

Tropical Storm

#howtopissawomanoff

#idontassociatewith

Inception

Comic-Con

Hip Hop

#np

```
var view = {
  title: "Joe",
  calc: function() {
    return 2 + 4;
  }
}

var template = "{{title}} spends
{{calc}}";

var html = Mustache.to_html(template,
view);
```

node-paperboy

Tom Hughes-Croucher aka sh1mmer

Writing (Twitter Blog)

Speaking (Speakerrate)

Code (Github YDN)

Hire Me (Email me)

http://

wargamez.mape.me/

mscdex
ryah
ryan_gahl
mjr_
elliottcable
bradleymeck_
satori_
hobson_

#node.js had activity 1.20sec ago

104 messages in the last 40min

ryah: elliottcable: shrug

mscdex: i mean the state of the program/connection/whatever could change such that the events should be disr...

ryan_gahl: mscdex: did i misunderstand?

DOM+YUI3

Rendering HTML

http://yuiloader.davglass.com/calendar/

Multi-core Node.js

Node+Web Workers

Master

```
var sys = require('sys');
var Worker = require('webworker').Worker;

var w = new Worker('foo.js');

w.onmessage = function(e) {
    sys.debug('Received mesage: ' + sys.inspect(e));
    w.terminate();
};

w.postMessage({ foo : 'bar' });
```

Worker

```
onmessage = function(e) {
    postMessage({ test : 'this is a test' });
};

onclose = function() {
    sys.debug('Worker shuttting down.');
};
```

Summary

- SSJS is awesome because
  - We are JavaScript programmers
  - Reuse (libraries/code)
    - Progressive Enhancement
- Node.js + YUI3 rocks
  - YUI 3's was easy to get running on Node.js
  - Server side DOM allows for a single code base

Today presentation was

Brought to you by the letters:
J and S

And the fonts:
Comic Sans
monofur

Tom Hughes-Croucher
@sh1mmer
croucher@yahoo-inc.com

Slides, etc --> http://speakerrate.com/sh1mmer
Pls rate me. kthxbai.