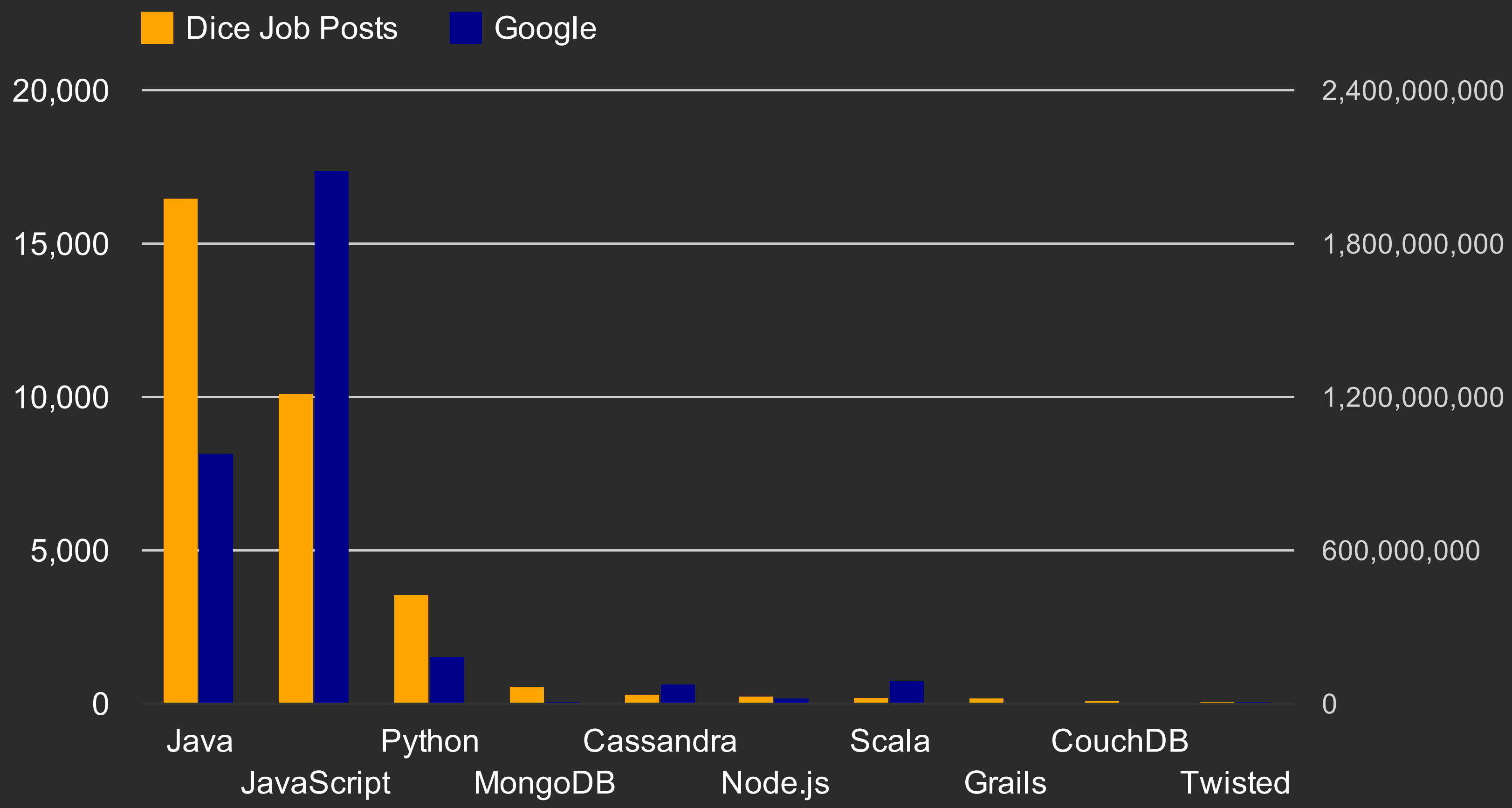
END-TO-END JAVASCRIPT WEB APPS

HTML5, NODE.JS AND MONGODB

CADEC 2013 by Peter Larsson

JAVASCRIPT IS NOT EVIL

TECH. INDEX, JAN 2013



NODE.JS

- Server side JavaScript engine based on Google's V8
- Event based
- Non-blocking I/O
- Single threaded
- CommonJS module system
- Development tools
- Demos

Created by Ryan Dahl starting in 2009, sposnored by Joyent

EVENTS

```
// Fragments

  function Server() { ... };

  var events = require('events');
  // inherit from EventEmitter
  Server.prototype.__proto__ = events.EventEmitter.prototype;
  var server = new Server();

  server.on('connection', function(socket) {
    console.log('someone connected!');
    socket.emit('ack', 'welcome');
  });
```

```
// de-register after first invocation
  socket.once('initalize', function(data) {
        console.log('called once only');
  });
```

BLOCKING VS. NON-BLOCKING I/O

BLOCKING

```
// read from database
var result = db.query("select * from table");
// do-something useful with result
```

NON-BLOCKING

```
// read from database
db.query(“select * from table”, function(result) {
        // do-something useful with result
});
// continue with some more important stuff
...
```

We are used to demand input and wait until we have it

BLOCKING VS. NON-BLOCKING I/O

BLOCKING

```
// read from keyboard
puts("Enter your name: ");
var name = gets();
puts("Name: " + name);
```

NON BLOCKING

```
// read from keyboard
puts("Enter your name: ");
gets(function (name) {
        puts("Name: " + name);
});
// continue with some more important stuff
...
```

Event approach is considered as too complicated

UTILIZE MULTI CPU/CORE

```
var cluster = require('cluster'), http = require('http'),
numCPUs = require('os').cpus().length;
if (cluster.isMaster) {
  for (var i = 0; i < numCPUs; i++) {
    cluster.fork();
  }
} else {
  http.createServer(function(req, res) {
    res.writeHead(200);
    res.end("hello world\n");
  }).listen(8000);
}
```

Cluster module to overcome single-threaded limitations

MODULE SYSTEM

- A module exports public functions
- Configuration is defined in a file, package.json
- Package manager **npm** to install, update and test

```
$ curl https://npmjs.org/install.sh | sh
$ npm install socket.io
```

```
/* Example of package.json */
{ "name": "mento-db",
  "version": "0.0.1",
  "description": "Mongo database module.",
  "main": "./index.js",
  "directories": {
    "lib": ".",
  },
  "dependencies": {
    "mongodb": "~1.2.7"
  }, ...
}
```

MODULE SYSTEM

```
/* Use Module API */
  var hello = require('./hello');
  hello.world();
```

```
/* Create Module API */
  exports.world = function() {
    console.log('Hello World');
  }
```

```
/* Overwrite exports */
  module.exports = function() {
    // require returns this function directly
  }
```

USEFUL DEVELOPMENT TOOLS

- Editor, **Sublime Text 2**
- Debugging, **Node Inspector**
- Build tool, **Grunt**
- Test tool, **Tap**

USEFUL FRAMEWORKS (MODULES)

- Web, **express, socket.io, http, https**
- Mongo, **mongodb, mongoose**
- Command line, **optimist**
- Scale - Multi core, **cluster**
- Scale - Load balancing, **http-proxy**
- Util, **util, sys**
- ... and more

6 DEMOS

Web Socket, HTTP, CPU, Debug, Mongo Module, Mento App

SIMPLE MESSAGE BENCHMARK

```
// client code
Cadec.initialize();
$('#burstBtn').click(function() {
  Cadec.start(); // start timer, for use of elapsed()
  Cadec.send('burst', $('#burstNum').val());
});
Cadec.listen('burst', function(data) {
  if (!(data.num % 10000)) {
    $('#burstReceived').html(data.num);
    $('#burstThroughput').html(
      Math.round((data.num/Cadec.elapsed())*1000));
  }
});
```

Received	Msgs/s
100000	20513

100000

Single Message Test

SERVER CODE

```
// server snippet...
socket.on('burst', function(total) {
  var timer = new Timer();
  for (var i = 1; i <= total; i++) {
    socket.emit('burst', { "num" : i });
  }
  console.log('done: throughput',
    Math.round((total / timer.elapsed()) * 1000),
    ' msgs/s');
});
```

ENTERPRISE ISSUES

- Still immature, current version is 0.8
- Not aligned with typical IT plattforms (sys admins)
  - Apache in front?, ouch...
- Better for Cloud deployments, i.e. Joyent SmartOS
  - With references as Sprint, LinkedIn, Microsoft, and Voxer
- Poorly suited for CPU intensive request handling

> *“ Node.js is Cancer ”*

by Ted Dziuba on Saturday, October 01, 2011

> *“ I don't care, it's a lot of Fun :) ”*

by me, right now

THE END

READ MORE

http://nodejs.org/

http://gruntjs.com/

http://www.html5rocks.com/en/

https://github.com/advptr/mento

https://github.com/isaacs/node-tap

https://github.com/dannycoates/node-inspector

http://www.mongodb.org/display/DOCS/Node.js