Node.js

As a networking tool

28.12.2010

About

Contributor

Felix Geisendörfer

Co-founder

node.js driver

node-formidable

Who are you?

Node.js?

JavaScript?

Port = 80 ?

Port != 80 ?

Node's goal is to provide an easy way to build scalable network programs.

Node.js

- Created by Ryan Dahl
- Google's V8 JavaScript engine (no DOM)
- Module system, I/O bindings, Common Protocols

Hello World

Network Edition

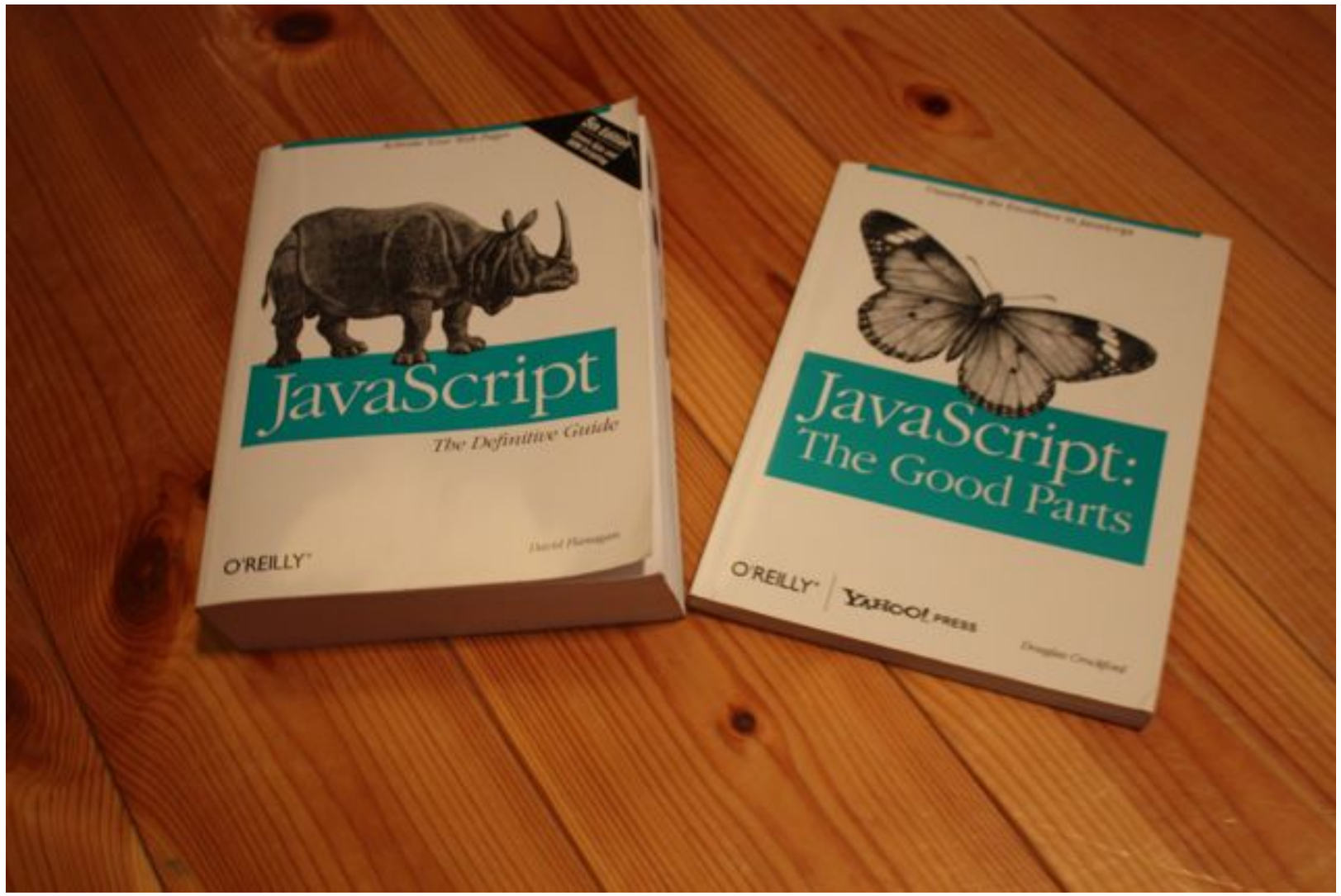
```
$ cat hello.js
var net = require('net');
net.createServer(function(socket) {
  socket.write('hello world\n');
  socket.end();
}).listen(0x27C3);

$ node hello.js &
[1] 3591
$ nc localhost 10179
hello world
```

Why JavaScript?

3 Reasons

#1 - The good parts



V8 (Chrome)

SpiderMonkey (Firefox)

JavaScriptCore (Safari)

#2 - JS Engine Wars

Chakra (IE9)

Carakan (Opera)

#3 - No I/O or Threads

Non-blocking I/O

Blocking I/O

```
db.query('SELECT A');
console.log('query A done');
```

```
db.query('SELECT B');
console.log('query B done');
```

Time = SUM(A, B)

Non-Blocking I/O

```
db.query('SELECT A', function(result) {
  console.log('query A done');
});

db.query('SELECT B', function(result) {
  console.log('query B done');
});
```

Time = MAX(A, B)

libev

by Marc Lehmann

libev

- An infinite loop (event loop)
- Polls the OS for events (select, epoll, kqueue, ...)
- Allows you to register watchers / callbacks for the events you care about

Buffers

Buffers

- Raw C memory allocation
- Similar to an array of integers
- Can be converted to and from JS strings

Buffers

```
buffer.write('Hello Buffer');

buffer.toString('utf-8');

buffer[0] = 23;

buffer.slice(10, 20);
```

Reverse Hello World

```
$ cat reverse_hello.js
var net = require('net');
net.createServer(function(socket) {
  socket.on('data', function(buffer) {
    console.log(buffer);
  });
}).listen(0x27C3);

$ node reverse_hello.js &
[1] 3523
$ echo "hello world" | nc localhost 10179
<Buffer 68 65 6c 6c 6f 20 77 6f 72 6c 64 0a>
```

Streams

"Streams are to time as arrays are to space."

-- Jed Schmidt @ JSConf.eu

Streams

```
stream.write(new Buffer([0x27, 0xC3]);

stream.pause();

stream.resume();

stream.destroy();
```

Streams

```
stream
  .on('drain', function() {})
  .on('data', function(buffer) {})
  .on('end', function() {});
```

Streams

```
$ cat echo.js
var net = require('net');
net.createServer(function(socket) {
   socket.pipe(socket);
}).listen(0x27C3);

$ node echo.js &
[1] 6207
$ nc 127.0.0.1 10179
Hi, how are you?
Hi, how are you?
```

pcap module

Sniffing data with node.js

pcap module

- Node.js bindings for libpcap
- Simple install: `npm install pcap`
- “Scriptable tcpdump”

pcap module

```
$ cat pcap.js
var util = require('util');
var pcap = require('pcap');
var filter = 'tcp port '+0x27C3;
var session = pcap.createSession('lo0', filter);

session.on('packet', function(packet) {
  packet = pcap.decode.packet(packet);
  console.log(pcap.print.packet(packet));

  if (packet.link.ip.tcp.data) {
    util.print(packet.link.ip.tcp.data.toString());
  }
});
```

pcap module

```
$ node hello.js > /dev/null &
[1] 3592
$ sudo node pcap.js &
[2] 3593
$ nc 127.0.0.1 10179 > /dev/null
loopback 127.0.0.1:54078 -> 127.0.0.1:10179 TCP len 64 [syn]
loopback 127.0.0.1:10179 -> 127.0.0.1:54078 TCP len 64 [ack,syn]
loopback 127.0.0.1:54078 -> 127.0.0.1:10179 TCP len 52 [ack]
loopback 127.0.0.1:10179 -> 127.0.0.1:54078 TCP len 64 [ack,psh]
hello world
loopback 127.0.0.1:54078 -> 127.0.0.1:10179 TCP len 52 [ack]
```

Recap

- Binary data handling (Buffers)
- I/O abstraction (Streams)
- Packet sniffing (node-pcap)

Node can also do

- HTTP
- Child processes
- Unix domain sockets
- DNS
- UDP
- Crypto
- ...

Speed

Speed

```
var http = require('http')
var b = new Buffer(1024*1024);

http.createServer(function (req, res) {
  res.writeHead(200);
  res.end(b);
}).listen(8000);
```

by Ryan Dahl

Speed

100 concurrent clients
1 megabyte response

node 822 req/sec
nginx 708
thin 85
mongrel 4

(bigger is better)

by Ryan Dahl

Speed

- NGINX peaked at 4mb of memory
- Node peaked at 60mb
- Very special circumstances

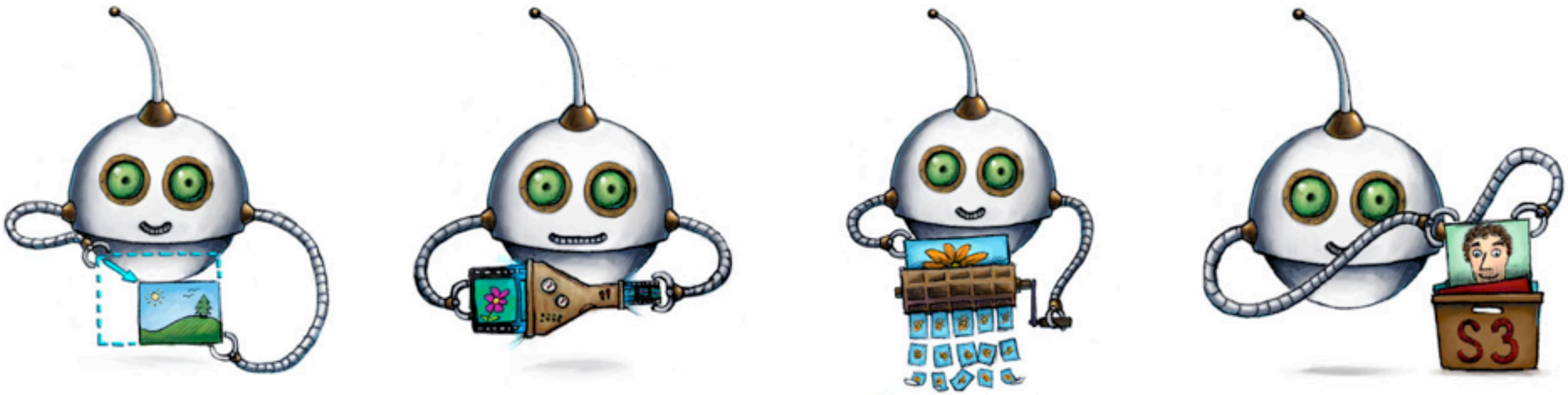
by Ryan Dahl

Danger, Will Robinson

Danger, Will Robinson

Node makes it trivial to manage thousands of connections on multiple protocol levels.

Transloadit



Resize images | Encode videos | Extract thumbnails | Store in S3

File uploading & processing as a service for other web applications.

Questions?

@felixge / felix@debuggable.com

Bonus Slides!

Addons

Addons

- Add bindings to your favorite C/C++ libraries

- Can be written as a mix of C++ / JavaScript

Addons

```
#include <v8.h>
#include <node.h>
#include <unistd.h>

using namespace v8;

static Handle<Value> Fork(const Arguments& args) {
  HandleScope scope;

  pid_t pid = fork();

  if (pid < 0)
  {
    return ThrowException(Exception::Error(String::New("fork: could not fork, pid < 0. see man fork")));
  }

  if (pid == 0)
  {
    ev_default_fork();
  }

  return scope.Close(Number::New(pid));
}

extern "C" void init (Handle<Object> target)
{
  HandleScope scope;
  NODE_SET_METHOD(target, "fork", Fork);
}
```