

NODE.JS

Bert Belder
Rackspace

about:node

"Evented I/O for V8 javascript"

Young, but:

- 5800 mailing list members
- 4000 NPM packages
- 7 core developers, 4 funded
- In production: LinkedIn, Facebook, Yahoo, Voxer, Uber, ...

Almost there



piscisaureus 1464

Dashboard

Explore GitHub Gist B

Popular Watched Repositories

Explore

Repositories

Languages

Timeline

Search

Interesting

Popular Forked

Popular Watched



rails

Ruby on Rails

rails

10592 watching



node

evented I/O for v8 javascript

joyent

9931 watching



jquery

jQuery JavaScript Library

jquery

9714 watching



html5-boilerplate

paulirish

8113 watching



Node is needed

The web evolves

Real time web

Client side rendering

The server as a message bus

I/O

Most programming languages wait for I/O to complete:

```
data = readfile("foo.bar");
```

While waiting, your program is sleeping.

Usual solution: threads

But...

Threads are difficult

... unless you don't try to share data between threads.

Traditional web stacks: 1 thread per connection

Real time web

Connections live long

How many threads can your server handle?

Node's solution

Handle everything in the same thread

V8 javascript

You already know javascript

You already know async

```
$( 'button' ).click( function () { ... } );
```

Javascript has no I/O library

Hej Verden

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

var n = 0;

var server = http.createServer(function(req, res) {
  res.writeHead(200, {
    'content-type': 'text/plain'
  });

  res.end('Hej Verden! ' + n++);
});

server.listen(80);
```

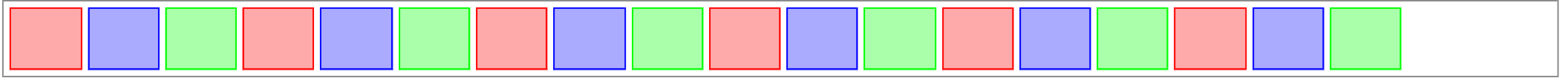
Single threaded

```
for (i = 0; i < 1e9; i++) {}
```

Does that matter?

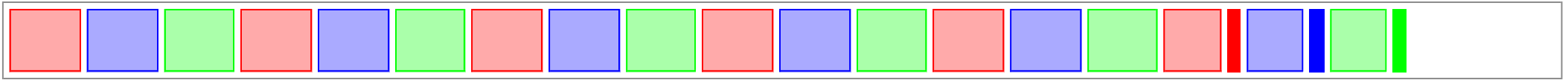
Event loop vs threads

Threads:

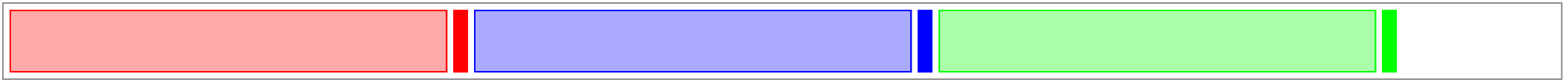


Event loop vs threads

Threads:



Event loop:

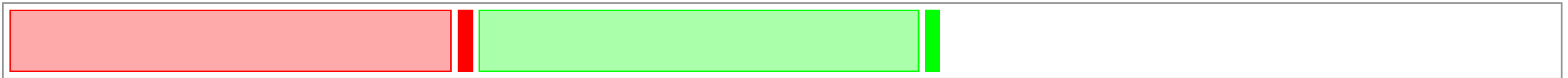


Event loop vs threads

Threads:

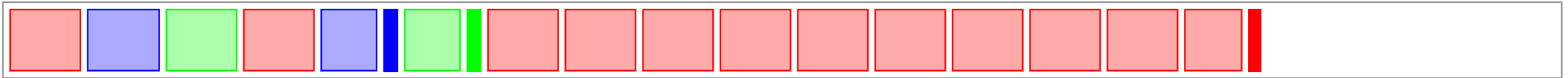


Event loop:



Event loop vs threads

Threads:



Event loop:



Outside of the thread

Run some nodes on 1 machine.

Run many nodes on many machines.

Under the hood

V8 + libev + libeio + c-ares + openssl + http_parser + glue

It's all about networking.

Files: thread pool

DNS: :-/

Under the hood (new)

V8 + libuv + openssl + http_parser + glue

Goal: good windows support

libuv: node in c (<https://github.com/joyent/libuv>)

Libraries

- NPM
- Socket.io
- Express
- Hook.io

Sticky node

<https://github.com/piscisaureus/goto>

```
$ npm install socket.io express
mkdirp@0.0.7 ./node_modules/express/node_modules/mkdirp
mime@1.2.4 ./node_modules/express/node_modules/mime
qs@0.3.1 ./node_modules/express/node_modules/qs
policyfile@0.0.4 ./node_modules/socket.io/node_modules/policyfile
redis@0.6.6 ./node_modules/socket.io/node_modules/redis
mime@1.2.4 ./node_modules/express/node_modules/connect/node_modules/mime
qs@0.3.1 ./node_modules/express/node_modules/connect/node_modules/qs
connect@1.7.1 ./node_modules/express/node_modules/connect
express@2.4.7 ./node_modules/express
xmlhttprequest@1.2.2 ./node_modules/socket.io/node_modules/socket.io-
client/node_modules/xmlhttprequest
websocket-client@1.0.0 ./node_modules/socket.io/node_modules/socket.io-
client/node_modules/websocket-client
uglify-js@1.0.6 ./node_modules/socket.io/node_modules/socket.io-
client/node_modules/uglify-js
socket.io-client@0.8.5 ./node_modules/socket.io/node_modules/socket.io-client
socket.io@0.8.5 ./node_modules/socket.io
```

Server

```
var express = require('express');
var app = express.createServer();

app.use(app.router);
app.use(express.static(__dirname + '/static'));

app.get('/', function(req, res){
  res.redirect('/index.html');
});

app.listen(process.env.C9_PORT || 80);
```

Client

```
var socket = io.connect(window.location.protocol + '//' +  
                        window.location.host + '/');
```

```
function send(note) {  
  socket.emit('note', note, clientId);  
}
```

```
socket.on('note', function(note, clientId_) {  
  if (clientId_ !== clientId) {  
    update(note);  
  }  
});
```


Server (2)

```
io.sockets.on('connection', function(client) {  
  client.on('note', function(note, clientId) {  
    io.sockets.emit('note', note, clientId);  
  
    if (note.message) {  
      notes[note.id] = note;  
    } else {  
      delete notes[note.id];  
    }  
  });  
});
```

Server (2)

```
io.sockets.on('connection', function(client) {
  client.emit('init', notes);

  client.on('note', function(note, clientId) {
    io.sockets.emit('note', note, clientId);

    if (note.message) {
      notes[note.id] = note;
    } else {
      delete notes[note.id];
    }
  });
});
```

Client (2)

```
var socket = io.connect(window.location.protocol + '//' +
                        window.location.host + '/');

function send(note) {
  socket.emit('note', note, clientId);
}

socket.on('init', function(notes) {
  for (var k in notes) {
    if (notes.hasOwnProperty(k)) {
      update(notes[k]);
    }
  }
});
```

<http://192.168.173.1>

SSID: node

Password: nodenode

Node 0.6

Node uses the linux versioning model. Even: (0.4.x) are stable, odd (0.5.x): unstable

- Real Windows support
- gzip
- Better fault isolation
- Better file watchers

Post-0.6

- IPC
- Managing multiple nodes

Get started

- nodejs.org
- NPM (npmjs.org)
- Socket.io
- Express

Questions?