Server-Side JavaScript Developement

Node.JS Quick Tour

q3boy <q3boy1@gmail.com>
from http://cnodejs.org/

Server-Side JavaScript Development

先驱

Aptana Jaxer (SpiderMonkey)

Helma (Rhino)

问题

- 缺乏统一标准
- 技术生态圈的问题

Revelutions in 2009

- CommonJS (ServerJS), by Dangoor
- JSConf, by Chris Williams & Iterative Designs
- Node, by Ray Dahl.

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

-- nodejs.org

Ray Dahl

发展历史

- 1/5/2009 Ryah Dahl 提出项目构想
- 2/15/2009 项目启动
- 5/31/2009 发布初始版本
- .....
- 11/16/2010 v0.3.1 发布

社区与生态圈

- 目前在官方wiki上有近600个module
- 超过20位活跃的开发者
- 每天都在进步

基础库

- libev (event loop)
- libeio (nonblocked posix, thread pool)
- v8 (javascript engine by google)

核心思想

- 非阻塞
- 单线程
- 事件驱动

传统模式下的io操作

```
var data = file.read("file.data");
// 等待io返回
doSomething(data);
```

程序逻辑需要等待io操作完成

非阻塞io操作

```
file.read("file.data", function(data){
    doSomthingAfterRead(data);
});
doSomethingOnReading();
```

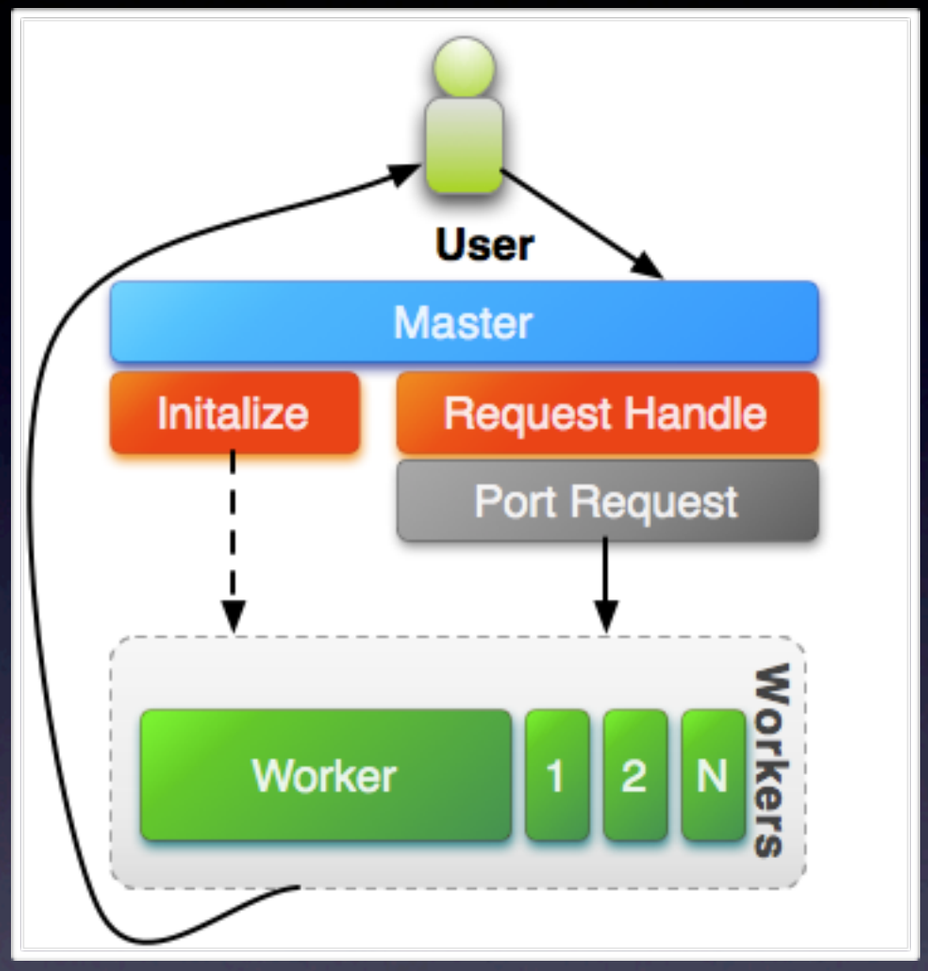
利用io操作的等待时间做其它工作

单线程 - 优势

- 程序逻辑简单
- 系统资源占用低
- 无通信与锁开销带来高性能

单线程 - 劣势

- 一个程序仅能在一个cpu上运行
- 一旦程序异常将导致整个进程崩溃



解決方案 - Web Worker

性能

- AMD Opteron 2200 单核, 4G RAM
- socket client 20,000+ qps
- socket server 17,000 qps
- http server 4,400 qps
- 内存消耗 30~40m
- see more. <http://nodejs.kongwu.net/blog/?p=8>

一些限制

- SSL支持不够成熟
- Windows
- 1G 堆栈内存限制(by v8)

适用场景

- Web框架
- 大并发负载中间层服务
- Spider

Digg in the node

安装

```
$ git clone git://github.com/ry/node.git
$ cd node
$ ./configure --prefix=$HOME/node
$ make install
```

Hello World!

```
console.log("Hello World!");
```

Familiar & Simple

单线程

```
setTimeout(function(){
    console.log(1);
    console.log(2);
}, 100);
setTimeout(function(){
    console.log(3);
    console.log(4);
}, 100);
```

```
$ node singleThread.js
1
2
3
4
```

CommonJS

```
// foo.js
exports.bar = function (){
    return "bar";
}
// CommonJS.js
var foo = require("./foo");
console.log("foo" + foo.bar());
```

```
$ node CommonJS.js
foobar
```

非阻塞io

```
var fs = require('fs');
fs.readFile('my.txt', function (err, data){
    if (err) {
        throw err;
    }
    console.log('my.txt:\r\n', data);
});
console.log('Reading my.txt ...');
```

```
$ node nonblocking.js
Reading my.txt ...
my.txt
Here is contents in "my.txt".
```

HTTP服务

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

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World!\n');
}).listen(8124);
```

TCP服务

```
var net = require("net");

net.createServer(function (stream) {
    stream.on('connect', function(){
        stream.write("Welcome!\r\n> ");
    });

    stream.on('data', function (data){
        stream.write(data + "> ");
    });
}).listen(1234);
```

```
$ node tcpServer.js &
$ telnet locahost 1234
Welcome!
> Node is so Cooool.
Node is so Cooool.
>
```

文件监听

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

fs.watchFile('my.txt', function (curr, prev) {
    console.log('the current mtime is :' +
      curr.mtime.getTime());
    console.log('the previous mtime is :' +
      prev.mtime.getTime());
});
```

```
$ node watchFile.js &
$ sleep 3; touch my.txt
the current mtime is: 1292594780000
the previous mtime was: 1292594752000
```

子进程

```
var cmd = 'echo hello; sleep 1; echo world;',
    spawn = require('child_process').spawn,
    child = spawn('sh', ['-c', cmd]);

child.stdout.on('data', function (chunk) {
    // chunk is a Buffer
    console.log(chunk.toString());
});
```

```
$ node childProcess.js
hello
world
```

String vs Buffer

高负载时String带来的问题

- 内存开销较大
- node不能直接得到v8中String的指针,在socket操作时需要执行memcopy
- 高负载时v8的gc对性能的影响

String vs Buffer

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

for (i = 0; i < 16 * 1024; i++) {
    string += 'd';
}

http.createServer(function (req, res){
    res.writeHead(200);
    res.end(string, 'ascii');
}).listen(8125);
```

With String

String vs Buffer

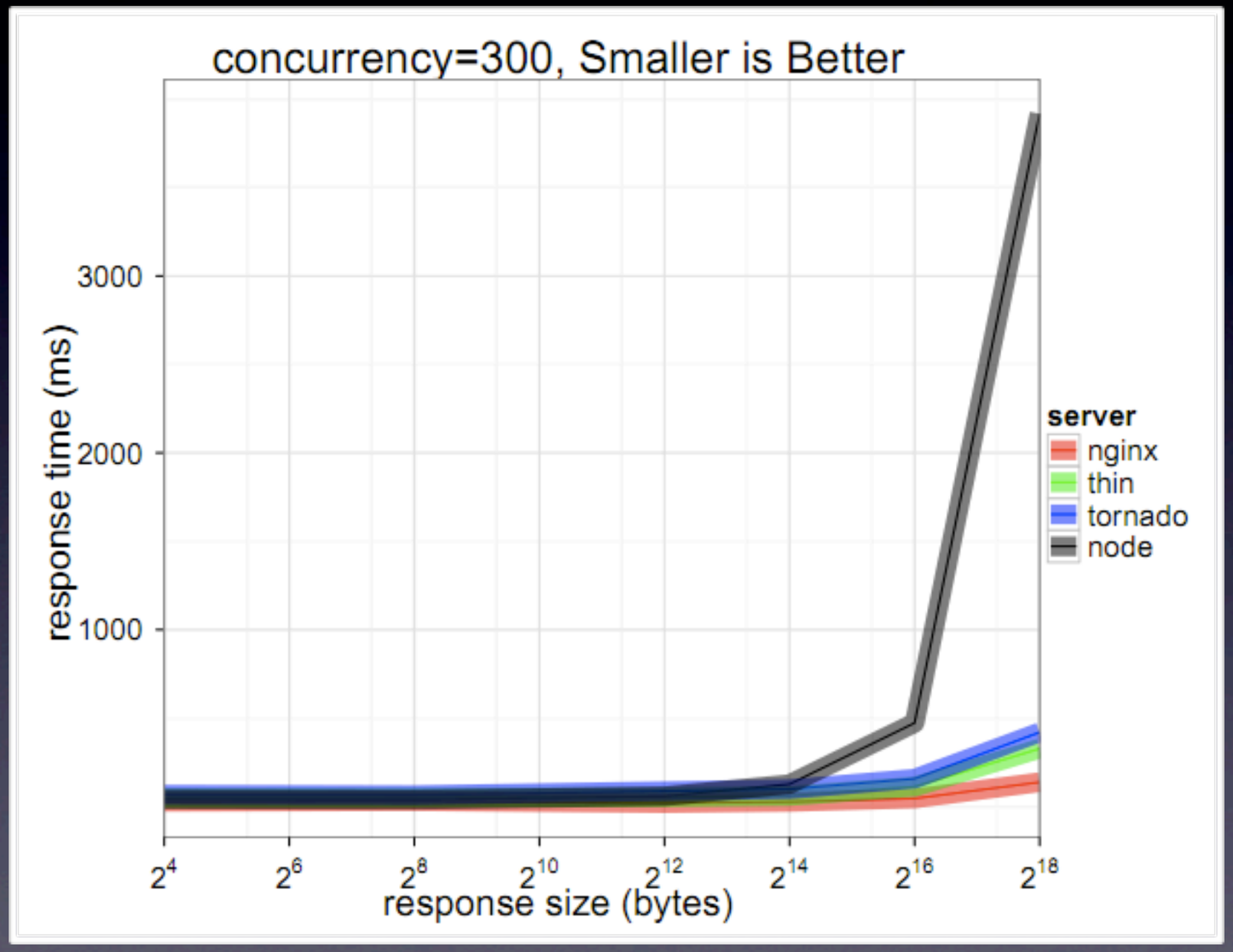
```
var http = require('http'),
    buffer = new Buffer(16 * 1024);

for (i = 0; i < buffer.length; i++) {
    buffer[i] = 100;
}

http.createServer(function (req, res){
    res.writeHead(200);
    res.end(buffer);
}).listen(8126);
```

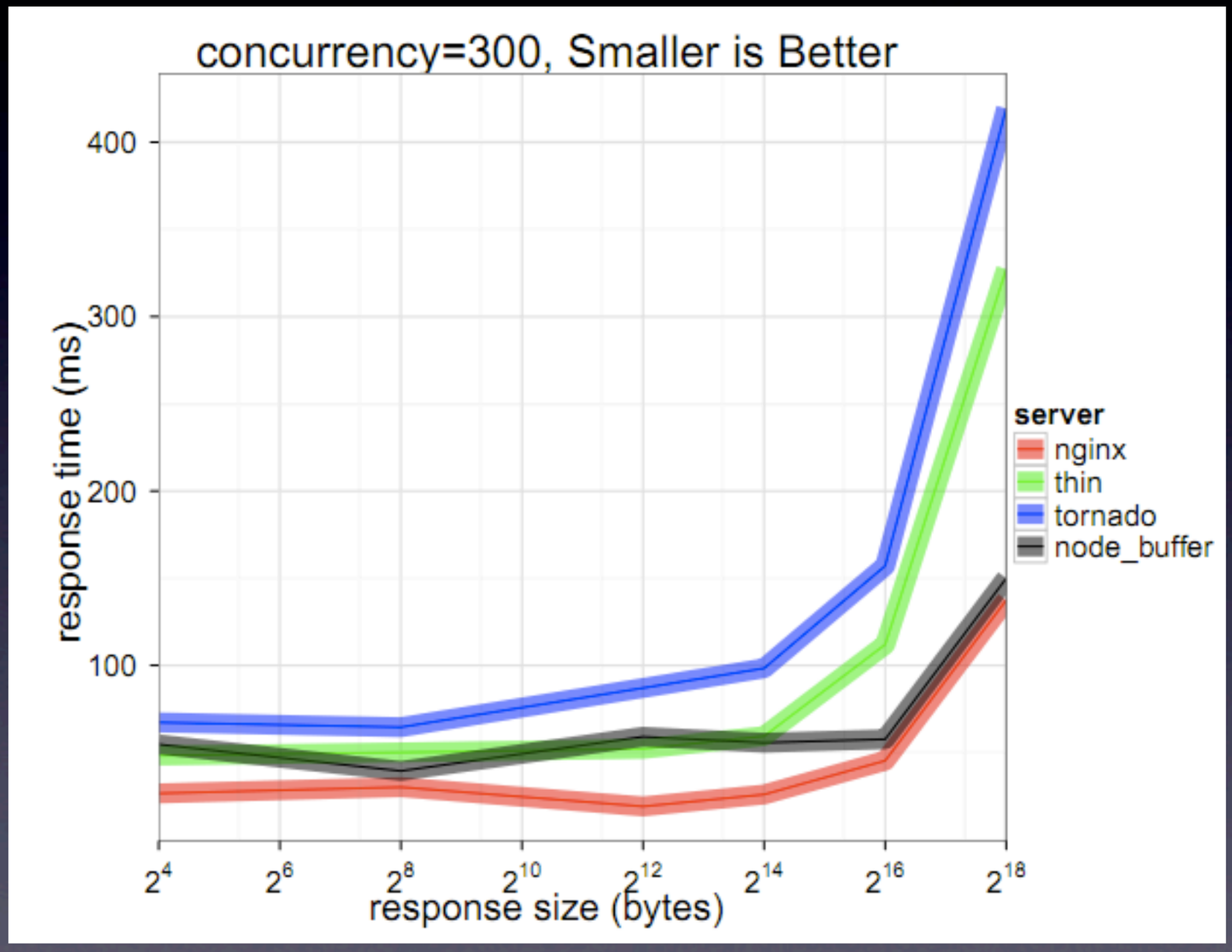
With Buffer

带来的性能改进



With String

带来的性能改进



With Buffer

Want to see more?

包管理

- PHP: pear
- Python: PyPI, setuptools
- Ruby: Gems
- Node: 

安装与使用

```
$ cat >>~/.npmrc <<NPMRC
root = ~/.node_libraries
binroot = ~/bin
manroot = ~/share/man
NPMRC
$ curl http://npmjs.org/install.sh | sh

$ npm ls
$ npm install package_name
$ npm update
```

IDE base on node

http://www.cloud9ide.com/

Links

- http://www.commonjs.org/
- http://nodejs.org/
- http://howtonode.org/
- https://github.com/ry/node/wiki
- https://github.com/ry/node/wiki/modules
- https://github.com/isaacs/npm


O, N, U, S,
B3

Node in Taobao

- UED
  - nodejs-kissy by 拔赤<bachi@taobao.com>
- EDP
  - http://cnodejs.org/ (致力于在国内推广node技术, coming soon)
  - node-reverse-proxy (超轻型反向代理, 支持轮循, coming soon)
  - node-myfox (mysql分布式集群 2011 Q1)
  - node-glider (多数据源的数据中间层服务 2011 Q2)
  - node-kvproxy (key-value集群访问代理 2011 Q2)


Thanks