Why

node.JS

by 桂林

桂糊涂@weibo

guilin1981@twitter

About me

- 2003, Delphi.
- 2006, Java.
- 2009, Python.
- 2010, Node.js.
- Author of mongoskin and stormjs

What is node.js?

Evented I/O
for
V8 JavaScript.

What can node.js do?

What can node.js do?

- Node.js as webserver

What can node.js do?

- Node.js as webserver
  - C10k problem

What can node.js do?

- Node.js as webserver
  - C10k problem
  - Comet

What can node.js do?

- Node.js as webserver
  - C10k problem
  - Comet
  - Websocket

What can node.js do?

- Node.js as webserver
  - C10k problem
  - Comet
  - Websocket
- Node.js as TCP server

What can node.js do?

- Node.js as webserver
  - C10k problem
  - Comet
  - Websocket
- Node.js as TCP server
- Node.js as CLI tools

What can node.js do?

- Node.js as webserver
  - C10k problem
  - Comet
  - Websocket
- Node.js as TCP server
- Node.js as CLI tools
- Node.js as GUI tools

Why node.js

- Why asynchronous IO?

Why asynchronous IO?

IO is the bottle-neck of application.

Why asynchronous IO?

IO is the bottle-neck of application.

- Network IO
- File IO
- Database IO

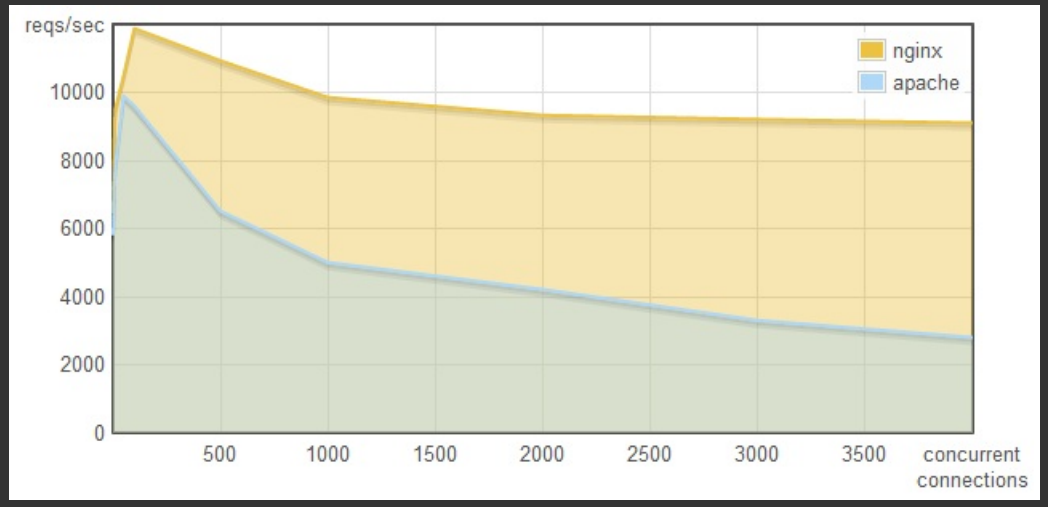
Blocking socket IO

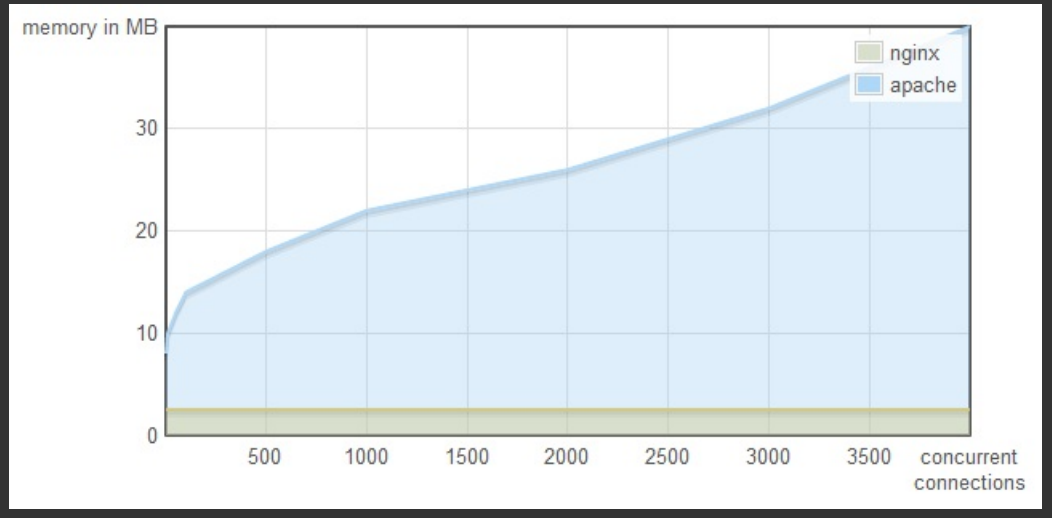
```
class ServerThread extends Thread {
  public ServerThread(socket){
    this.socket = socket;
  }
  public void run(){
    BufferedReader reader = new
BufferedReader(this.socket.getInputStream());
    String line;
    while((line = reader.readLine()) != null){ // block here
      // handle line
    }
  }
}
...
ServerSocket serverSocket = new ServerSocket(port);
while (true) {
    Socket s = serverSocket.accept(); // block here
    new ServerThread(s).start();
}
```

Non-blocking socket IO

```
net.createServer(function(socket){
  socket.on('data', function(data){
      // handle data
  });
  socket.on('close', function(has_error){
      // handle close
  });
  socket.on('error', function(err){
      // handle error
  });
}).listen(port);
```

Non-blocking IO vs Blocking IO





nginx(non-blocking) vs apache(blocking)

Node.js IO API

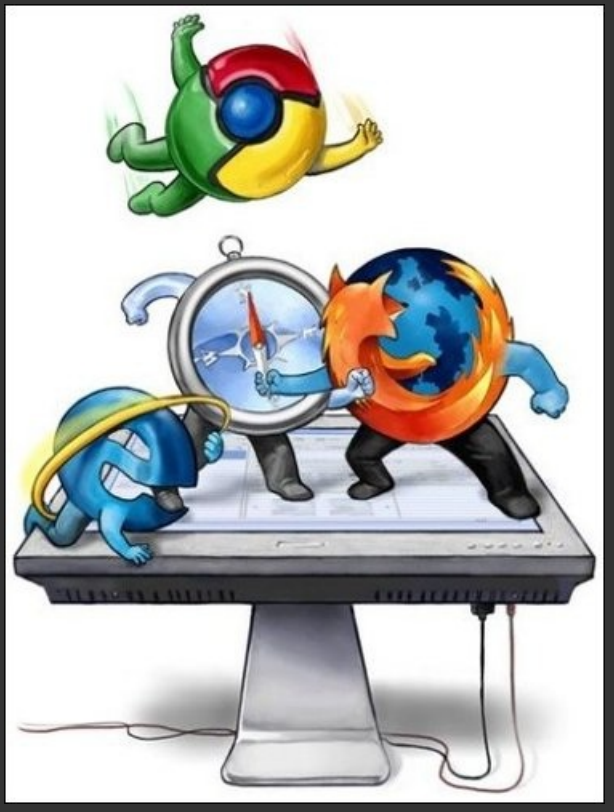
- stream
- fs
- http
- net
- dns

Why node.js

- Why asynchronous IO?
- Why javascript?

Why javascript?

Why javascript?



Why javascript?

Why javascript?

- Browsers war improving javascript.

Why javascript?

- Browsers war improving javascript.
- Javascript is one of the fastest dynamic programming language.

Why javascript?

- Browsers war improving javascript.
- Javascript is one of the fastest dynamic programming language.
- Javascript is one of the most popular programming language.

Why javascript?

- Browsers war improving javascript.
- Javascript is one of the fastest dynamic programming language.
- Javascript is one of the most popular programming language.
- Javascript has closure(anonymoused function).

Why javascript?

- Browsers war improving javascript.
- Javascript is one of the fastest dynamic programming language.
- Javascript is one of the most popular programming language.
- Javascript has closure(anonymoused function).
- Javascript is cross-platform.

Why node.js

- Why asynchronous IO?
- Why javascript?
- Why v8?

About v8

About v8

- V8 javascript VM is used in Google Chrome.

About v8

- V8 javascript VM is used in Google Chrome.
- V8 team is led by Lars Bak, one of the leading VM engineers in the world with 20 years of experience in building VM.

About v8

- V8 javascript VM is used in Google Chrome.
- V8 team is led by Lars Bak, one of the leading VM engineers in the world with 20 years of experience in building VM.
- Lars Bak was the technical lead behind HotSpot(Sun's Java VM). HotSpot improved Java's performance 20x times.
- Before HotSpot, Lars Bak worked on a smalltalk VM.

How fast v8 is?

- Fast Property Access
- Dynamic Machine Code Generation
- Efficient Garbage Collection

V8 example

The JavaScript code to access property x from a Point object is:

```
point.x
```

V8 example

The JavaScript code to access property x from a Point object is:

```
point.x
```

In V8, the machine code generated for accessing x is:

```
# ebx = the point object
cmp [ebx,<hidden class offset>],<cached hidden class>
jne <inline cache miss>
mov eax,[ebx, <cached x offset>]
```

Why node.js

- Why asynchronous IO?
- Why javascript?
- Why v8?
- Why threadless?

Why threadless?

Why threadless?

- 2mb stack per thread

Why threadless?

- 2mb stack per thread
- Dead-locking

Why threadless?

- 2mb stack per thread
- Dead-locking
- Thread-safe?

Why threadless?

- 2mb stack per thread
- Dead-locking
- Thread-safe?
- Thread is design for blocking IO.

Why node.js

- Why asynchronous IO?
- Why javascript?
- Why v8?
- Why threadless?
- Why node.js special?

Why node.js special?

Pythoner:

Why node.js special?

Pythoner:

- Python is one of the most popular dynamic language too.

Why node.js special?

Pythoner:

- Python is one of the most popular dynamic language too.
- Performance of python is OK.

Why node.js special?

Pythoner:

- Python is one of the most popular dynamic language too.
- Performance of python is OK.
- We have good non-blocking web framework.

Tornado by facebook

Tornado is a non-blocking web server, open-source by facebook.

Hello Tornado

```
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")

application = tornado.web.Application([
    (r"/", MainHandler),
])

if __name__ == "__main__":
    application.listen(8888)
    tornado.ioloop.IOLoop.instance().start()
```

Hello Nodejs

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello Node.js\n');
}).listen(3000, "127.0.0.1");
```

Tornado with IO

```
import pymongo
import tornado.web

class Handler(tornado.web.RequestHandler):
    @property
    def db(self):
        if not hasattr(self, '_db'):
            # block here
            self._db = pymongo.Connection(
                host='127.0.0.1', port=27017).test
        return self._db

    def get(self):
        try:
            # block here
            user = self.db.users.find_one({'username':
self.current_user})
            self.render('template', full_name=user['full_name'])
        except:
            raise tornado.web.HTTPError(500)
```

Tornado with async IO

```
import asyncmongo
import tornado.web

class Handler(tornado.web.RequestHandler):
    @property
    def db(self):
        if not hasattr(self, '_db'):
            self._db = asyncmongo.Client(pool_id='mydb',
                host='127.0.0.1', port=27017,
                maxcached=10, maxconnections=50, dbname='test')
        return self._db

    @tornado.web.asynchronous
    def get(self):
        self.db.users.find({'username': self.current_user}, limit=1,
callback=self._on_response)

    def _on_response(self, response, error):
        if error:
            raise tornado.web.HTTPError(500)
        self.render('template', full_name=respose['full_name'])
```

Node.js with IO

Anonymoused function, is very important for Evented IO

```
var mongo = require('mongoskin'),
    http  = require('http'),
    jade  = require('jade'),
    db    = mongo.db('localhost/test');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  db.users.findOne({'username': req.session.current_user},
function(err, user){
    jade.renderFile('template', {user: user}, function(err, html){
      res.end(html);
    });
  });
}).listen(3000, "127.0.0.1");
```

What make node.js special?

What make node.js special?

- Javascript is special.

What make node.js special?

- Javascript is special.
- Most of node.js modules are asynchronoused.

What make node.js special?

- Javascript is special.
- Most of node.js modules are asynchronoused.
- It’s easy to write threadless module.

What make node.js special?

- Javascript is special.
- Most of node.js modules are asynchronoused.
- It’s easy to write threadless module.
- Node.js community is very active.

How many modules?

2320 modules can be install via npm

2011, June 07.

What do you think about node community?

Arunoda:

Its the freedom followed by tools like github and npm. And everything is growing fast.

Pau:

People is more open minded than in other programming communities. There is people coming from ruby, python, c, haskell, java, erlang, php...

Mikeal Rogers:

People using node.js are building stuff. the project seeks to make the lives better of people *building products*.

Mikeal Rogers:

People using node.js are building stuff. the project seeks to make the lives better of people *building products*.
My experience with Python and Ruby is that their primary reason for working on the project is around with a new language and/or vm. the people who work on “core” don’t build products and probably never will.

Mikeal Rogers:

People using node.js are building stuff. the project seeks to make the lives better of people *building products*.
My experience with Python and Ruby is that their primary reason for working on the project is around with a new language and/or vm. the people who work on “core” don’t build products and probably never will.
Node.js is not a language and it doesn’t write it’s own vm. it’s not attractive to people who don’t care about building stuff *with* it.

桂林:

潮不等于装13

桂林:

潮不等于装13 , node.js很酷，也很务实。

Why node.js

Why node.js

- Why asynchronous IO?
  - Never blocking, cpu efficient
- Why javascript?
  - Right and popluar language
- Why v8?
  - Extremely fast VM
- Why threadless?
  - Easy, memory efficient
- Why node.js special?
  - Active and creative community

Thank you