Node.js

https://github.com/ivanrios/node

@ivanrios

Presencia de Node.js

Linkedin: El backend de la versión móvil esta hecha completamente con node.js, la primer razón fue la escalabilidad y la segunda la ganancia en performance. De tener 30 servidores de Ruby, bajaron a 3 con Node.js http://highscalability.com/blog/2012/10/4/linkedin-moved-from-rails-to-node-27-servers-cut-and-up-to-2.html

Paypal: Su equipo de desarrollo estaba partido en Desarrolladores de lado del server en Java y del lado del cliente en javascript, bajaron el tiempo de desarrollo, aumentaron la capacidad de peticiones por segundo y bajaron el tiempo de respuesta *https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/*

eBay: Javascript reemplaza a Java, lo usan para API, monitoreo y logging. “JavaScript is EVERYWHERE.” http://www.ebaytechblog.com/2013/05/17/how-we-built-ebays-first-node-js-application/#.VOaoqFOG_VQ

New York Times: API para móviles hecha con Node.js

Yahoo: El core del proyecto Manhattan esta basado en Node.js (Servicio online para creación de contenido en HTML5)

Microsoft http://azure.microsoft.com/en-us/develop/nodejs/

Walmart: http://nodejs.org/video/ **Netflix** http://techblog.netflix.com/2014/11/nodejs-in-flames.html

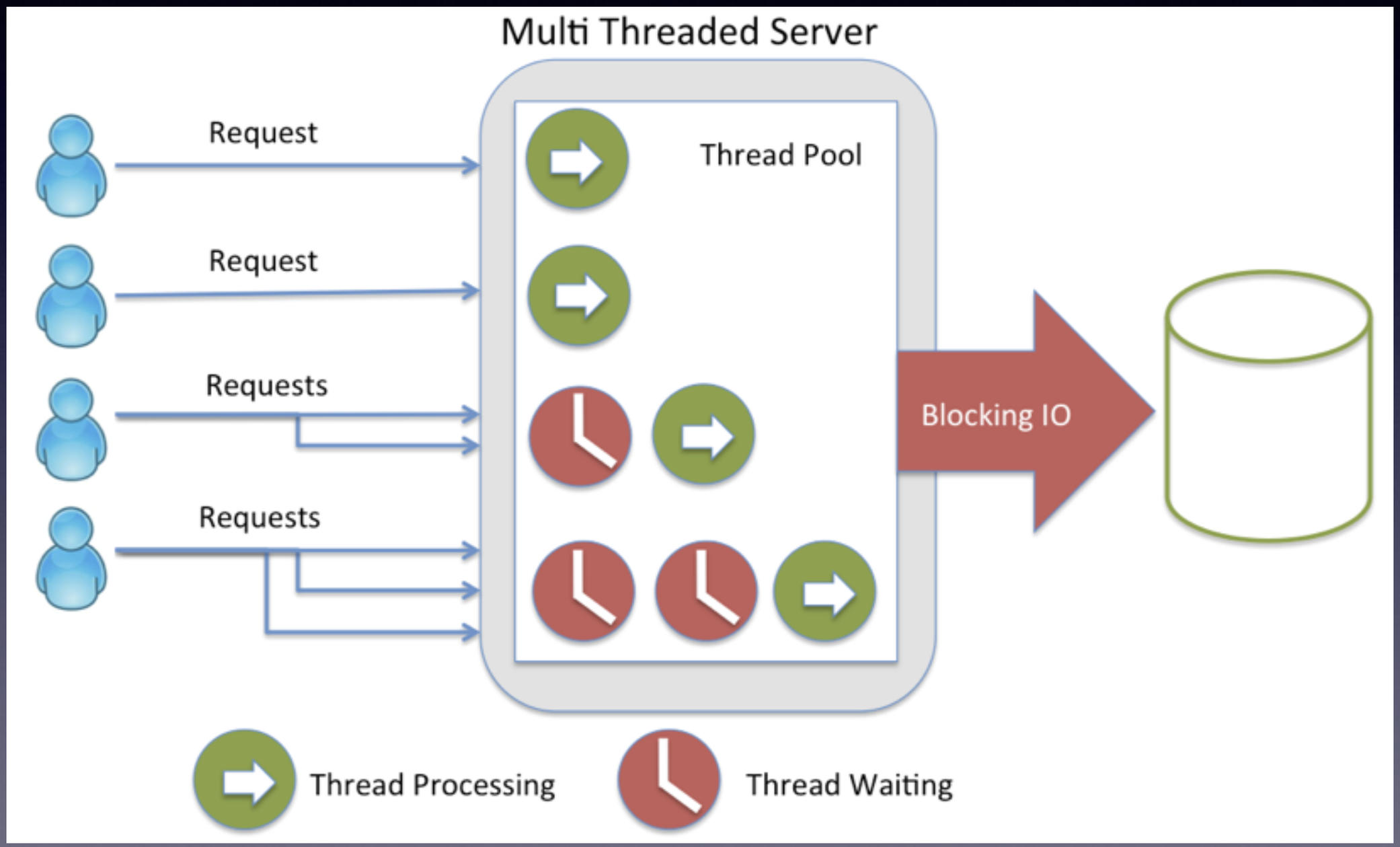
http://nodejs.org/industry/

¿Que es Node.js?

Entorno de desarrollo de Javascript en el servidor.

Su meta es permitir construir aplicaciones web altamente escalables y que manejen decenas de miles de conexiones simultáneas en un solo servidor.

¿Que problema resuelve?



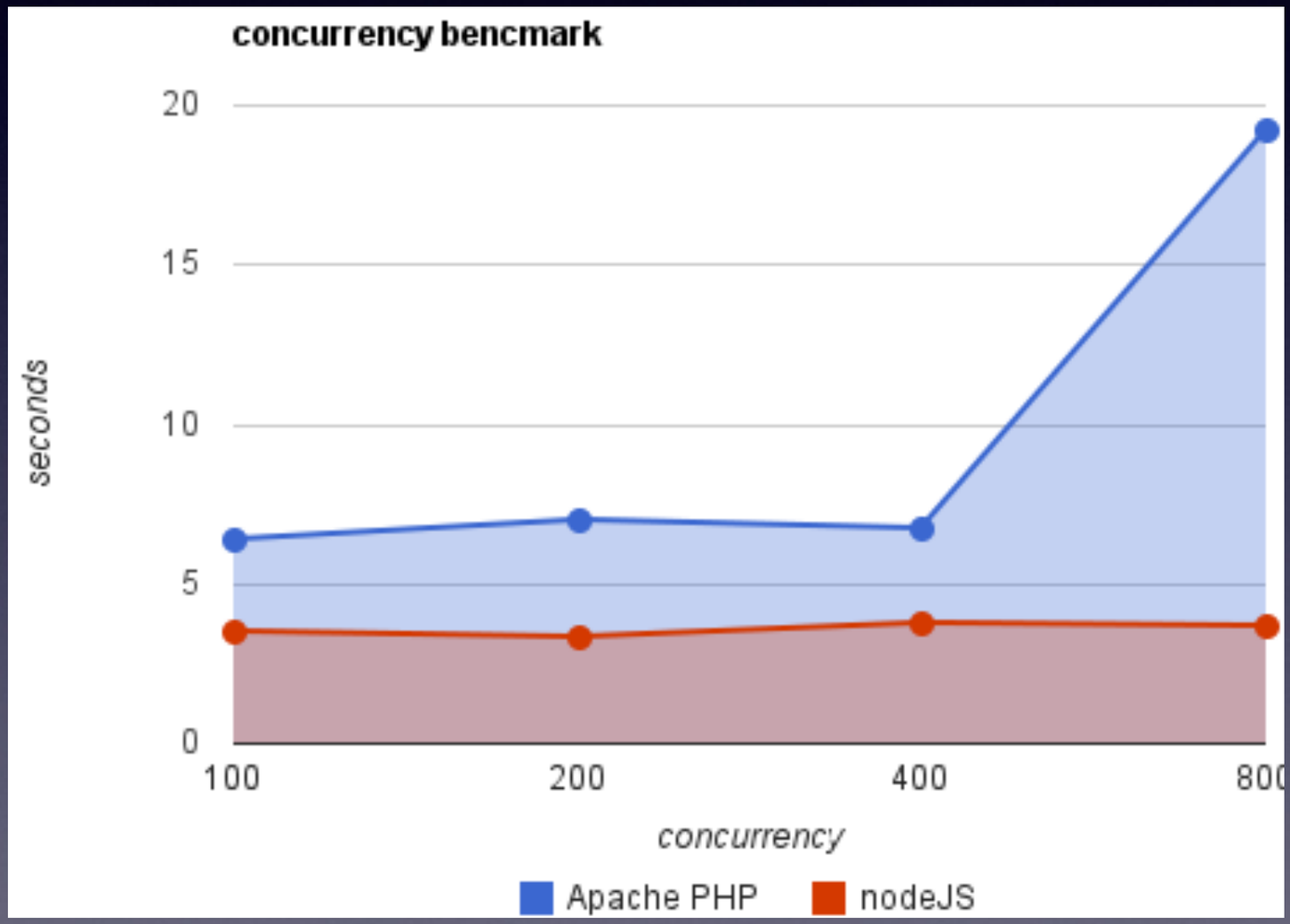
Comparativo

Apache

- Crea un nuevo hilo por cada conexión cliente-servidor
- MaxClients por defecto es 256.

Node

- Corre sobre un hilo, si hay una operación bloqueante, crea otro hilo.
- Puede mantener tantas conexiones como número máximo de archivos descriptores (sockets) soportados por el sistema.
- Unix soporta 65,000 aunque en realidad soporte de 20-25,000 clientes concurrentes



¿Por qué javascript?

Pensamiento tradicional

```
resultado = query( “SELECT * FROM VENTAS” );
// cruzas los dedos
ImprimirResultados( resultado );
HacerOtraCosa();
```

Cómo trabajamos actualmente con javascript

```
$.ajax({  url: “ventas.php”,  cache: false})
  .done(function( html ) {
    $( "#results" ).append( html );
  });
```

Aplicando el mismo concepto:

```
query("SELECT * FROM VENTAS", ImprimirResultados() );
HacerOtraCosa();
```

¿Cómo lo hace?

Event loop

Ejemplo

```
var http = require('http');
var servidor = http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
});
servidor.listen(8000);
console.log('Servidor ejecutándose!');
```

```
$ node hello_world.js
```

Let’s code

https://github.com/ivanrios/node.git

Katas

Node.js Intro
http://www.codewars.com/kata/541db50c259d9c55c00007b9

Password Hashes
http://www.codewars.com/kata/54207f9677730acd490000d1

Node.js Async I/O
http://www.codewars.com/kata/542106e2dda52658bf00001a