
LABORATORIO TECNOLOGÍAS AUDIOVISUALES EN LA WEB

José María Cañas Plaza

jmplaza@gsyc.es



Grado Sistemas Audiovisuales y Multimedia
curso 2015-2016

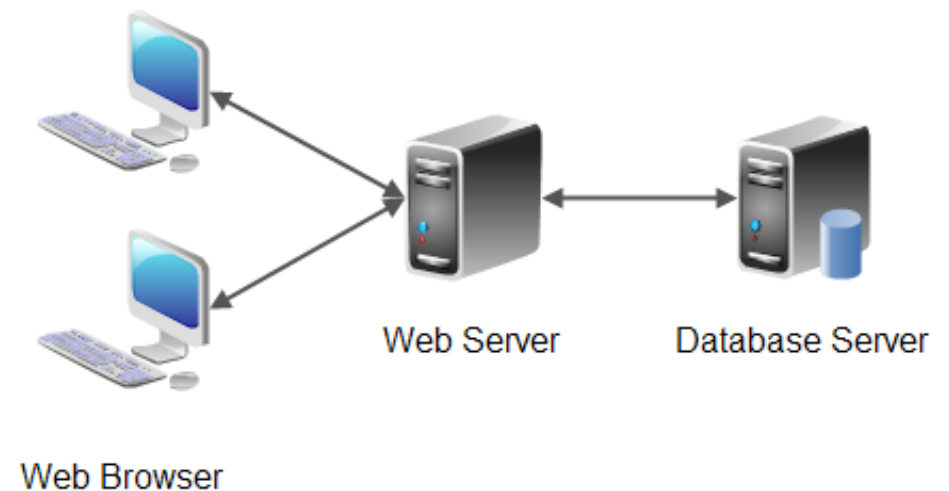
NODE.JS

Contenido

- Introducción
- Node.js
- Servidor web básico con Node.js
- Sirviendo recursos estáticos y bases de datos

Introducción

- Pasamos al lado servidor
- Atender peticiones HTTP, responder GET, POST...
- También AJAX, Cookies, Formularios...
- Recursos web: css, imágenes, vídeos, html, Manifest
- Acceso a bases de datos
- Web estáticas vs webs dinámicas



Una aplicación web puede ser utilizada por múltiples (miles) usuarios simultáneamente, lo que implica que puede suponer una carga importante de trabajo para el servidor a medida que aumentan el número de usuarios y de tareas demandadas.

- Servidores multihilo
- Servidores en varias (muchas) máquinas
- Escalabilidad
- Latencia

Servidor web básico

- Escuchar peticiones HTTP, puerto 80
- Atender petición, generar respuesta (lleva su tiempo)
- Enviar respuesta
- Volver a escuchar, bucle infinito
- Monohilo (bloqueante):
 - (-) atiende a pocas peticiones por segundo
 - (-) no aprovecha hardware paralelo

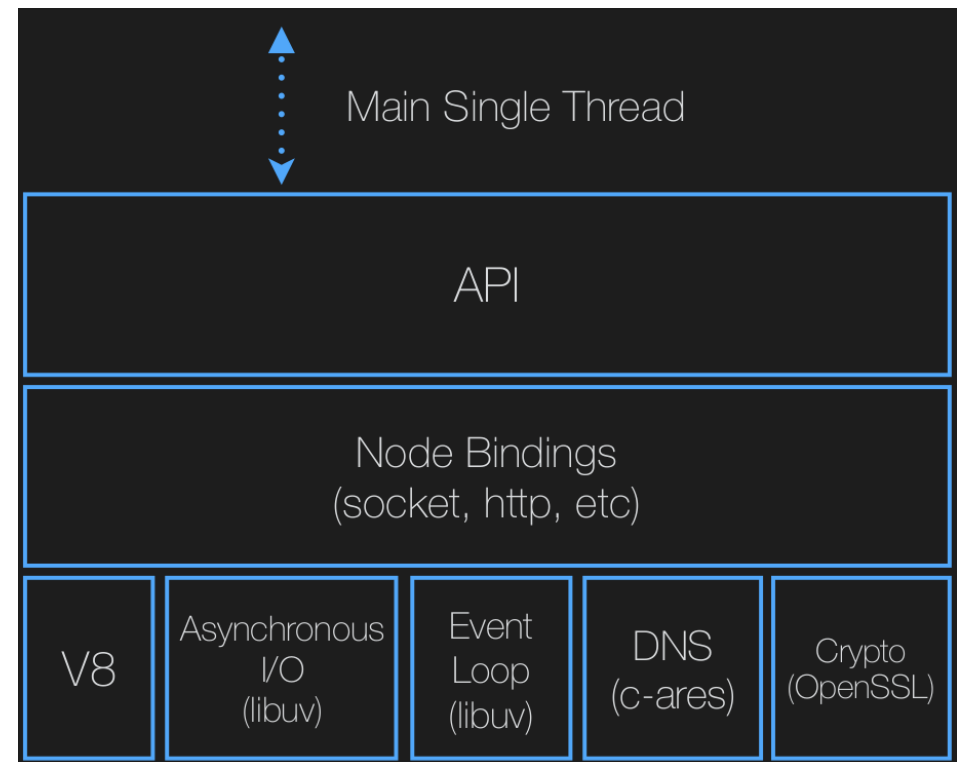
Node.js

- Aplicaciones servidor y de comunicaciones
- 2009 por Ryan Dahl
- 2015 Node.js + io.js
- Gran éxito implantación: ebay, PayPal, Yahoo...
- JavaScript
- Multiplataforma: Windows, Linux, MacOS
- Software libre



Características

- Motor JavaScript V8 de Google
- Orientado a *eventos*
- Monohilo
- E/S no bloqueante
- *npm* Node Package Manager
- Gran eficiencia
- Alta escalabilidad: millón de conexiones simultáneas

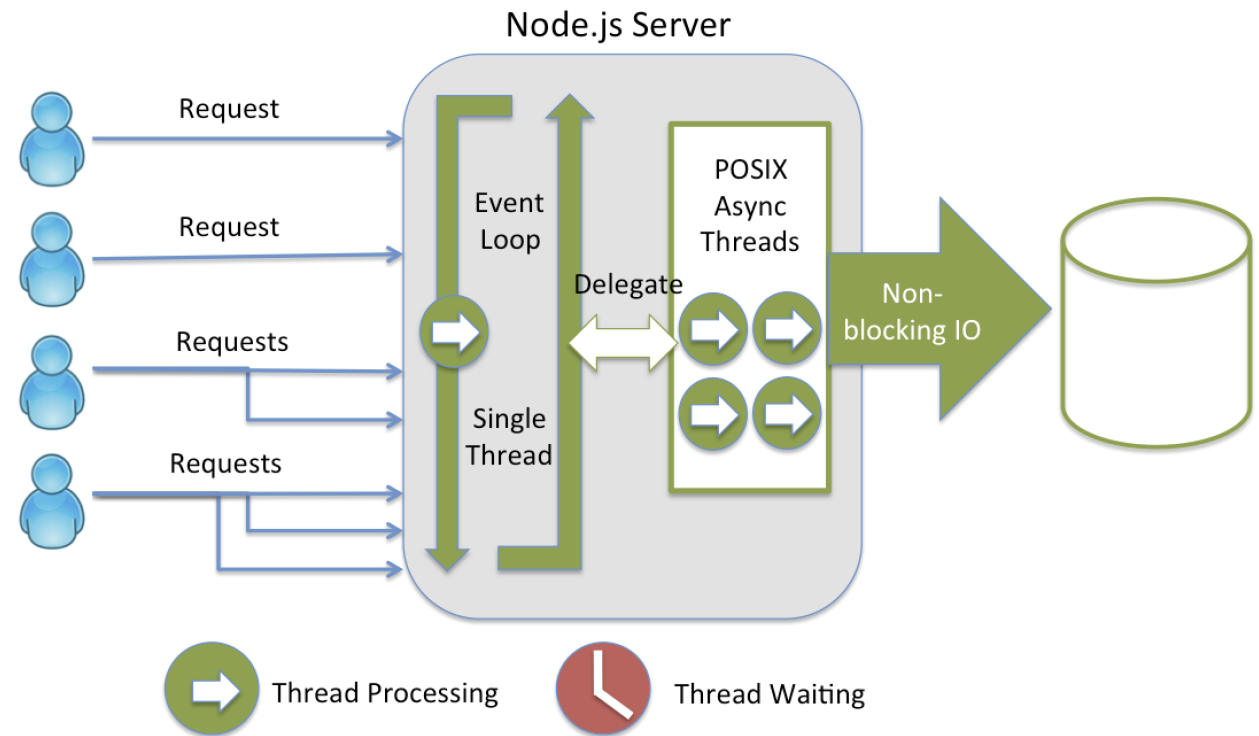


- Main core
- *Bindings* para operaciones de servidor
- Módulos
 - I/O para sistema de archivos
 - Networking (DNS, HTTP, TCP, TLS/SSL, o UDP)
 - Datos binarios (buffers)
 - Funciones de criptografía
 - Data streams
- *Frameworks*
 - Express.js
 - Socket.io (websockets)

Implementación

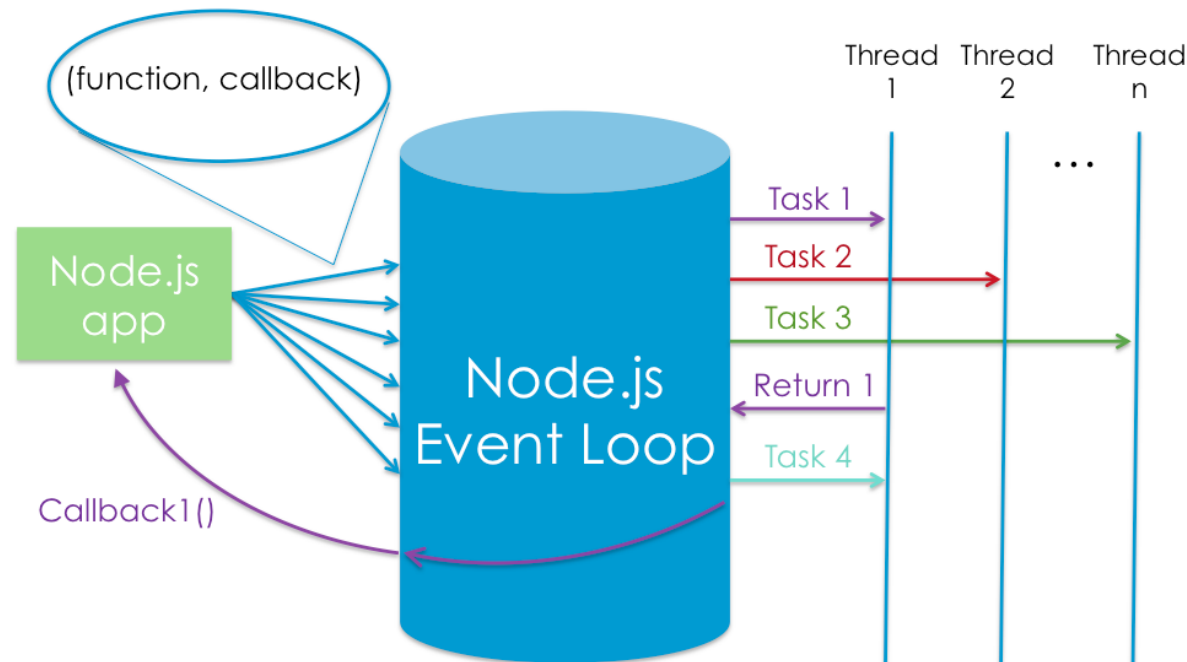
- Escrito en C++
- V8 compila JS *just in time* a nativo para mayor eficiencia
- incluye un compilador, un optimizador y un recolector de basura
- Cada conexión es un trozo de memoria en *heap* y callbacks

- Monohilo
- Bucle de eventos
- Pool de hebras
- E/S no bloqueantes
- Biblioteca libuv
operaciones de red y
archivos, tipo POSIX



1 Node apps pass async tasks to the event loop, along with a callback

2 The event loop efficiently manages a thread pool and executes tasks efficiently...



3 ...and executes each callback as tasks complete

- Comandos se ejecutan no bloqueantes
- Callbacks para recoger éxito o fracaso

Balance

- (+) Escalable, rápido
- (+) Orientado a aplicaciones en tiempo real intensivas en datos
- No está orientado a servidores intensivos en CPU
- (-) Punto único de fallo -> módulos hebras multimáquina (*Cluster...*)
- (+) Enfoque unificado en servidor y cliente

Servidor Web básico con Node.js

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

http.createServer(function(request, response) {
  var headers = request.headers;
  var method = request.method;
  var url = request.url;
  var body = [];

  request.on('error', function(err) {
    console.error(err);
  }).on('data', function(chunk) {
    body.push(chunk);
  }).on('end', function() {
    body = Buffer.concat(body).toString();

  });
}).listen(8080);
```

```
response.on('error', function(err) {  
    console.error(err);  
});  
  
response.statusCode = 200;  
response.setHeader('Content-Type', 'application/json');  
  
var responseBody = {  
    headers: headers,  
    method: method,  
    url: url,  
    body: body  
};  
  
response.write(JSON.stringify(responseBody));  
response.end();
```

- Routing de peticiones
- Errores en recoger petición o en armar respuesta
- Cabeceras de mensajes http
- Flujos, cuerpo
- Acceder a ficheros para empaquetarlos en respuestas

Sirviendo recursos estáticos y bases de datos con Node.js

- Ficheros crudos, recubrimiento de POSIX
- Módulos que simplifican servir ficheros estáticos o directorios
- MongoDB, base de documentos, indexada
- MySQL, base de datos relacionales (node-mysql...)

Referencias

- <https://nodejs.org/en/docs/guides/anatomy-of-an-http-transaction/>
- <https://nodejs.org/api/http.html>
- Wikipedia: <https://en.wikipedia.org/wiki/Node.js>