

Contenido

Introducción.....	17
¿Cómo se organiza este libro?	18
Rendimiento JavaScript	19
Rendimiento de red	20
Rendimiento del servidor	21
Convenciones	21
Código fuente.....	22
Sobre la imagen de cubierta	22
Capítulo 1. Qué es el rendimiento Ajax.....	25
Compromisos	25
Principios de optimización	26
Ajax 28	
El navegador	29
¡Vaya!	30
JavaScript	31
Resumen	32
Capítulo 2. Cómo crear aplicaciones Web que funcionen.....	33
¿Cómo de rápido?	35
Cómo medir la latencia	37
Problemas de latencia.....	39

Ejecución en hilos	39
Cómo garantizar una respuesta adecuada	41
Web Workers.....	41
Gears 42	
Relojes (timers)	43
Efectos del uso de la memoria sobre el tiempo de respuesta	45
Memoria virtual	46
Solución de problemas de memoria	47
Resumen	48
Capítulo 3. Dividir la carga de trabajo inicial.....	49
El fregadero	49
El ahorro de la división	51
Cómo encontrar la división	52
Símbolos indefinidos y condiciones de carrera	53
Ejemplo: Google Calendar	54
Resumen	55
Capítulo 4. Cargar scripts sin bloqueos	57
Bloqueo de scripts	57
Cómo conseguir que los scripts se comporten bien	59
XHR Eval	60
XHR Injection.....	61
Script en iframe	62
Script DOM Element	63
Script Defer	63
Etiqueta de scripts document.write	64
Indicadores de navegador ocupado	64
Cómo conseguir (o evitar) la ejecución ordenada.....	66
Resumen de resultados	67
Y el ganador es...	69
Capítulo 5. Acoplamiento de scripts asincrónicos	73
Ejemplo de código menu.js.....	74
Condiciones de carrera	76
Cómo preservar el orden asincrónicamente	77
Técnica 1: Escritura de retrollamada en el código fuente	79
Técnica 2: Window Onload	79
Técnica 3: Controlador Timer	81

Técnica 4: Script Onload	82
Técnica 5: Etiquetas de scripts degradados	82
Múltiples scripts externos	85
XHR Gestionado	86
DOM Element y Doc Write	89
Solución general	93
Script único	93
Múltiples scripts	94
La asincronía en el mundo real	97
Google Analytics y Dojo	97
Utilidad de carga YUI Loader.....	99
Capítulo 6. Posicionamiento de scripts internos.....	103
Bloqueo de scripts internos	103
Desplazar los scripts internos al final de la página	104
Inicialización de la ejecución asíncrona.....	105
Utilizar Script Defer	107
Preservar el orden de CSS y JavaScript	107
¡Peligro! Hoja de estilo seguida de script interno	108
La mayoría de las descargas no bloquean los scripts internos	109
Las hojas de estilo bloquean los scripts internos	110
Cosas que pasan.....	111
Capítulo 7. Cómo escribir código JavaScript eficaz.....	115
Gestión del alcance	116
Utilice variables locales	117
Aumento de la cadena de alcance	120
Acceso eficiente a los datos	122
Control de flujo	126
Condicionales rápidos	126
La instrucción if	126
La instrucción switch	129
Otra opción: búsqueda de arrays.....	130
Los condicionales más rápidos.....	131
Bucles rápidos.....	132
Mejoras de rendimiento para bucles sencillos.....	132
Evite el bucle for-in	134
Desenrollar bucles	135

Optimización de cadenas	138
Concatenación de cadenas	138
Recortar cadenas	140
Evite los scripts de ejecución prolongada	142
Pausas utilizando controladores de tiempo	143
Patrones de reloj para las pausas	145
Resumen	148
Capítulo 8. Escalar con Comet.....	151
¿Cómo funciona Comet?	152
Técnicas de transporte	153
Llamada selectiva (polling)	153
Long polling.....	154
Forever Frame	156
Streaming XHR	158
Transportes futuros	160
Interdominios	160
Efectos de la implementación	
sobre las aplicaciones.....	162
Gestionar las conexiones	162
Medir el rendimiento	163
Protocolos	163
Resumen	164
Capítulo 9. Más allá de la compresión Gzip	165
¿A quién le importa?	165
Causas	167
Un repaso rápido	167
El culpable	168
Algunos ejemplos de la "escucha de la tortuga"	169
¿Cómo podemos ayudar a estos usuarios?	170
Minimizar el tamaño sin comprimir	
a la hora de diseñar	170
Delegación de eventos	170
Utilizar URL relativos	172
Eliminar el espacio en blanco	173
Eliminar las comillas de los atributos	173
Evite los estilos internos	173
Alias de JavaScript	174
Ahorros reales	174
Educar a los usuarios	175

Dirigir la detección del soporte para Gzip	176
Cómo realizar la prueba	176
Utilizar el resultado	178
Medir la efectividad	178
Capítulo 10. Optimización de imágenes.....	179
Dos pasos para simplificar la optimización de imágenes	180
Formatos de imagen	181
Fondo	181
¿Gráficos o fotografías?	181
Píxeles y RGB	182
Truecolor o formatos de imagen de paleta	182
Transparencia y canal alfa (RGBA).....	183
Entrelazado	183
Características de los diferentes formatos	184
GIF	184
JPEG	185
PNG	186
Más información sobre el formato PNG	187
PNG8, PNG24 y PNG32	187
Comparación del formato PNG con otros formatos	187
Comparación con el formato GIF	187
Comparación con el formato JPEG	187
Problemas de transparencia del formato PNG	188
PNG8 y el software de edición de imágenes	188
Optimización de imágenes automatizada sin pérdidas	189
Eliminar fragmentos PNG	189
Pngcrush	190
Otras herramientas de optimización de imágenes PNG	190
Eliminación de los metadatos JPEG	191
Convertir GIF a PNG	192
Optimización de animaciones GIF	193
Smush.it	193
JPEG progresivo para imágenes de gran tamaño	194
Transparencia alfa: evite el uso de AlphaImageLoader	195
Efectos de la transparencia alfa	195
Esquinas.....	196
AlphaImageLoader	197

Problemas con AlphaImageLoader	199
Tierras salvajes: un ejemplo de Yahoo! Search	200
Transparencia alfa PNG8 progresivamente mejorada	201
Optimización de sprites	202
Über-sprite o sprite modular	203
Sprites CSS altamente optimizados	204
Otras técnicas de optimización de las imágenes	205
Evite el escalado de imágenes	205
Imágenes generadas mediante compresión	205
Favicons	207
Iconos de favoritos de Apple Touch	209
Resumen	209
Capítulo 11. Fragmentación de dominios predominantes	211
Ruta crítica	211
¿Quién utiliza la fragmentación?	213
Utilizar HTTP/1.0	216
Implementación de la fragmentación	
entre dominios	219
Dirección IP o nombre de huésped	219
¿Cuántos dominios?	219
Cómo dividir los recursos	220
Navegadores recientes	220
Capítulo 12. Vaciado temprano del documento	221
Vaciado del encabezado	221
Buffering de la salida	224
Codificación fragmentada	225
Vaciado y compresión Gzip	227
Otros intermediarios	228
Bloqueo de dominios durante el vaciado	229
Los navegadores: el último obstáculo	230
El vaciado más allá de PHP	230
Listado de verificación para el vaciado	231
Capítulo 13. Uso ocasional de los iframes.....	233
¿Cuál es el elemento más oneroso del DOM?	234
Los iframes bloquean el evento onload	235
Descargas paralelas con iframes	236
Script antes del iframe	237

Hoja de estilo antes del iframe	238
Hoja de estilo después de un iframe	238
Conexiones por nombre de servidor	239
Compartir la conexión en iframes	240
Compartir la conexión entre pestañas y ventanas	241
Resumen: El coste de los iframes	242
Capítulo 14. Simplificación de los selectores CSS	245
Tipos de selectores	246
Selectores de ID	247
Selectores de clase.....	247
Selectores de tipo	247
Selectores hermanos adyacentes	247
Selectores de hijos.....	248
Selectores descendientes	248
Selectores universales	248
Selectores de atributos	249
Pseudo-clases y pseudo-elementos	249
Selectores CSS eficientes	249
Primero, la derecha	250
Cómo escribir selectores CSS eficientes	251
Rendimiento de los selectores CSS	252
Los selectores complejos afectan al rendimiento... a veces	252
¿Qué selectores CSS debemos evitar?	255
Tiempo de reflujo	257
Cómo medir los selectores CSS en el mundo real	258
Apéndice A. Herramientas de rendimiento	261
Rastreadores de paquetes	262
HttpWatch	262
Panel de red de Firebug	262
AOL Pagetest	264
VRTA	264
IBM Page Detailer	264
Panel de recursos de Web Inspector	265
Fiddler	265
Charles	265
Wireshark	266

Herramientas de desarrollo Web 266

- Firebug 266
- Web Inspector 267
- Developer Toolbar de Internet Explorer 267

Analizadores del rendimiento 268

- YSlow 270
- AOL Pagetest 271
- VRTA 273
- neXpert 274

Varios 275

- Hammerhead 275
- Smush.it 276
- Cuzillion 276
- UA Profiler 277

Índice alfabético 279