

Arquitecturas Distribuidas

Tema 4. VII- Optimización de sitios web

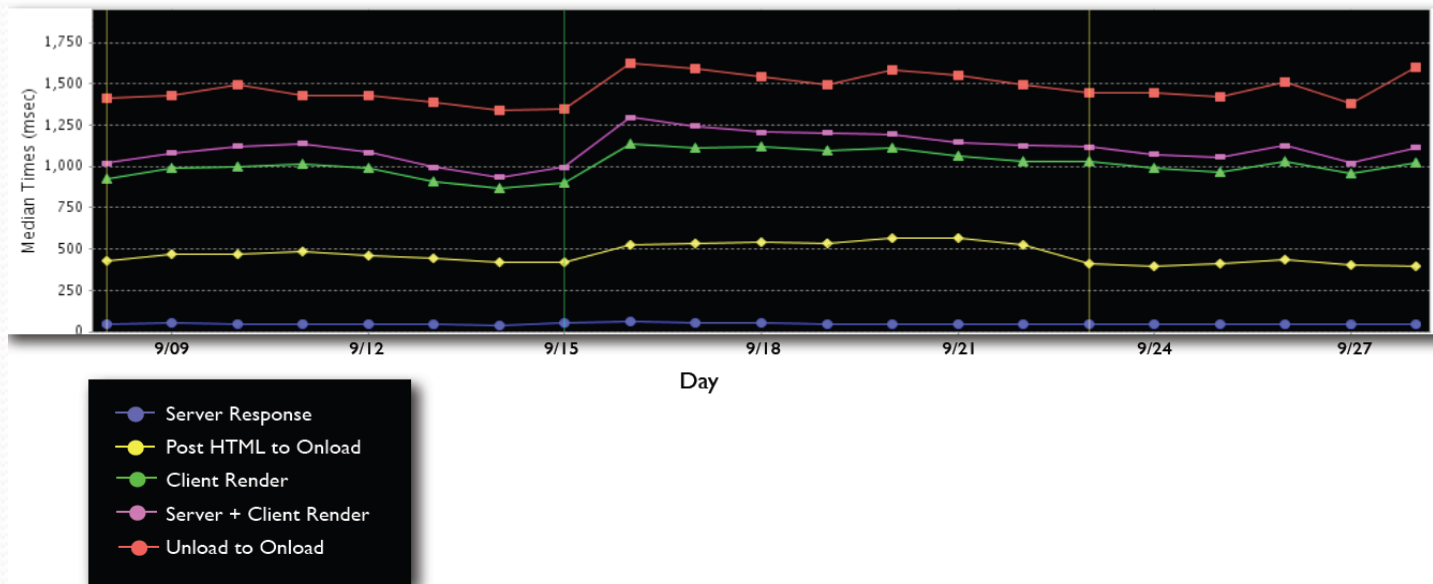
Definiciones

- Backend
 - *Tiempo desde que el usuario hace la petición hasta que el último byte del documento HTML llega. Incluye el tiempo de la petición inicial, la generación de la respuesta del servidor y el tiempo en volver.*
- Frontend
 - *Todo lo que ocurre desde que el documento HTML llega. Debería incluir sólo actividades como visualización (parsing) del HTML, CSS y ejecución del javascript. En realidad, incluye parte del tiempo del backend (lectura de documentos estáticos principalmente).*

La importancia del rendimiento del front-end

- La mayor parte del tiempo se emplea en actividades del front-end: 80-90%

Typical Performance



Peticiones y caché

- Los navegadores realizan peticiones al servidor sólo si no disponen de los documentos en caché
- La caché se implementa tanto en memoria como en disco
 - ¿Cuándo en una u otro sitio? Depende de la política del navegador.
 - La caché de memoria es lógicamente la más rápida, pero se borra al cerrar el navegador.
 - Cuando un contenido está almacenado en disco se comprueba si es válido (cabecera Expires de HTTP).
 - Si no es válido se realiza una petición GET condicional al servidor. Si sigue siendo válido el servidor no lo devuelve de nuevo.

Ejecución síncrona y progresiva

- Los navegadores comienzan a analizar (parse) el documento HTML a medida que va llegando.
- El orden en el que aparecen los elementos importa
 - Si aparece un script externo tienen que cargarlo. Mientras se descarga se bloquea el dibujo del resto del contenido.
 - Algunos navegadores como IE no dibujan nada hasta que no se han cargado todas las CSS para evitar tener que redibujar. Otros como Firefox comienzan a dibujar y cuando se termina de cargar redibujan, lo que a veces supone un efecto “flash”

Reglas para optimizar el contenido

- Reglas propuestas por Steve Souders junto con Yahoo (no todos los sitios tienen sus necesidades: millones de usuarios al día). Algunas pueden ser incluso contraproducentes para sitios pequeños.
 1. Hacer menos peticiones HTTP
 2. Usar una CDN (si realmente el sitio lo requiere)
 3. Añadir cabecera Expires o max-age
 4. Gzip componentes
 5. Poner CSS al principio (para evitar bloqueos)
 6. Poner scripts al final (para evitar bloqueos)
 7. Evitar expresiones CSS y reducir llamadas al DNS
 8. CSS y javascript en ficheros externos
 1. Se almacenan en caché. Excepción: página de inicio
 9. Reducir el tamaño del javascript (espacios en blanco, comentarios...)
 10. Evitar redirecciones
 11. Evitar dispersar el javascript en el documento (para evitar bloqueos)
 12. Optimizar las imágenes

Referencias y bibliografía

- S. Souders, “High performance web sites”, O'Reilly Media, 2007
- <http://cs193h.stevesouders.com/>