Sistema predictivo para la reducción del tiempo de descarga de páginas Web y de la carga del servidor

Josep Domènech, José A. Gil, Julio Sahuquillo, Ana Pont
Department of Computer Engineering (DISCA)
Universitat Politècnica de València.
Camí de Vera, s/n. 46022 València (Spain)
jdomenech@ai2.upv.es; {jagil,jsahuqui,apont}@disca.upv.es

Abstract— **Las técnicas de web caching reducen la latencia percibida por el usuario gracias a que sirven los objetos más populares de una memoria intermedia. Para asegurar que los objetos reusados siguen siendo todavía válidos, se envían peticiones condicionales al servidor original. Muchas de las respuestas a estas peticiones son mensajes cortos del tipo "304 Not Modified", que no incluyen el contenido del objeto. El uso de estas peticiones hacen que el usuario se ahorre la parte de latencia correspondiente a la transferencia del objeto, pero no el tiempo de RTT relacionado con cada par petición respuesta. Nuestro sistema persigue reducir la necesidad de utilizar peticiones condicionales para validar un objeto. Esto se consigue mediante el uso de algoritmos de predicción que incluyen junto con la URL de cada objeto predicho, la fecha de última modificación o E-Tag de cada objeto.**

Esta técnica tiene un buen potencial de mejora de prestaciones ya que el análisis de la distribución de respuestas del servidor nos muestra que entre el 35% y el 71% de todas las respuestas de los servidores más utilizados son "304 Not Modified". Este potencial ha sido confirmado mediante simulación, ya que se han alcanzado reducciones de hasta el 58% en la latencia percibida por el usuario y del 68% la carga de peticiones del servidor.

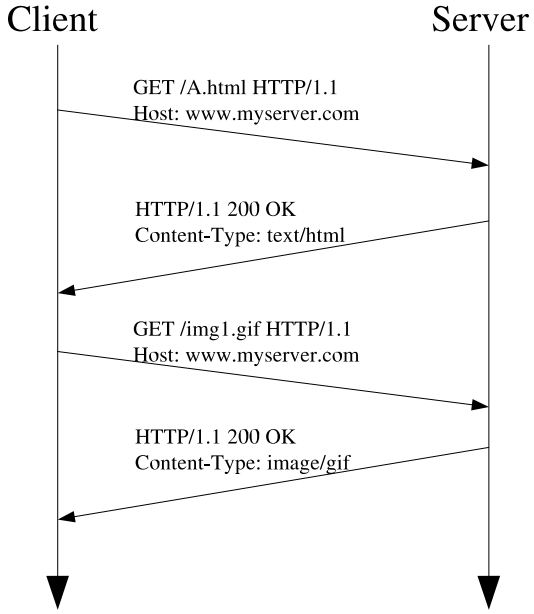

Fig. 1. Funcionamiento basico de la comunicacion cliente-servidor en HTTP

I. Introducción

Desde la aparición de la Web se han realizado numerosos esfuerzos para reducir el tiempo de descarga de las páginas Web. La Web funciona siguiendo la filosofía tradicional de cliente-servidor por medio del protocolo HTTP. La figura 1 muestra un ejemplo de cómo se realiza la comunicación entre ambas partes. En este ejemplo, un cliente solicita al servidor *www.myserver.com* la página *A.html*. Generalmente las páginas suelen estar compuestas de varios objetos como imágenes, etc. En el ejemplo, la página *A.html* contiene la imagen *img1.gif*, que se solicita al servidor tras haber recibido el documento principal (*A.html*). La latencia percibida por el usuario (tiempo de descarga de la página) viene dado principalmente por el tamaño de los objetos solicitados, el ancho de banda disponible entre el cliente y el servidor, el tiempo de procesamiento de cada petición por el servidor y el tiempo que tarda la señal en propagarse desde el cliente al servidor.

Las principales técnicas desarrolladas hasta ahora para la reducción de latencia son las Redes de Distribución de Contenido (CDN, por sus siglas en inglés) y el Web Caching. La primera consiste en hacer copias del servidor y situar cada una de estas copias cerca de un grupo de usuarios finales, de tal forma que se reduce el tiempo que tarda la señal en propagarse desde el cliente al servidor y, por lo tanto, la latencia percibida por el usuario [1], [2]. Existe una amplia implantación de esta técnica en el mercado, donde destacan empresas con un gran volumen de negocio como Akamai, Amazon S3, BitGravity, etc.

El Web Caching se fundamenta en el hecho de que los usuarios suelen acceder a páginas que habían accedido previamente [3], [4]. En esta técnica, el cliente copia en su disco duro de forma automática copias las páginas accedidas para poderlas utilizar en el futuro. La reducción de la latencia se consigue por medio de servir el contenido del objeto solicitado de forma local, que es significativamente más rápido que pedirlo al servidor. El Web Caching se utiliza en prácticamente todos los navegadores Web que existen en la actualidad, además de ser utilizado de forma más global en la mayoría de organizaciones por medio de los servidores proxy. Estos servidores

proxy están desarrollados por empresas como Microsoft, Netscape, Sun, etc. y utilizados en grandes organizaciones como Telefonica o la UPV entre otras muchas.

En un sistema de prebúsqueda como los que existen en la actualidad, el servidor Web mantiene un registro de los accesos realizados para predecir los accesos siguientes de los usuarios [5], [6]. El modo de funcionamiento es el siguiente:

1. El navegador web del usuario realiza la petición de un objeto web al servidor.
2. El servidor consulta su histórico de accesos y predice cuáles serán los próximos accesos de este usuario. Esta predicción se incorpora a la respuesta, junto con el objeto web requerido.
3. Cuando el navegador web del usuario está ocioso, pide por adelantado los objetos predichos por el servidor y se almacenan localmente.
4. Cuando el usuario accede al objeto predicho, se sirve la copia local del objeto con lo que se reduce el tiempo de descarga de la página.

El sistema propuesto se diferencia del sistema de prebúsqueda tradicional en que, junto con la predicción de accesos, el servidor web envía otra información adicional que hace que el cliente sepa de antemano qué peticiones de acceso debe realizar y cuáles son innecesarias. De esta forma, se consigue reducir la carga de peticiones que recibe el servidor web y el tiempo de descarga de páginas percibido por el usuario. Todo ello sin incrementar el tráfico de red entre clientes y servidores. Según los estudios realizados, con esta nueva técnica, tanto la carga de peticiones del servidor web como el tiempo de descarga se reduce entre un 40% y un 60%.

El resto del artículo se estructura de la siguiente forma: La sección II muestra algunos detalles de las técnicas actuales de reducción de latencia sobre los que se asienta nuestra propuesta. La sección III detalla el funcionamiento de la técnica propuesta y analiza los resultados obtenidos en simulación. Finalmente, la sección IV presenta las conclusiones obtenidas.

II. Antecedentes

En esta sección se van a mostrar los detalles del estado actual de la técnica que van a permitir introducir los detalles de nuestra propuesta más adelante.

A. Formación del tiempo de descarga

Una página web suele estar formada por múltiples objetos. De esta forma, el tiempo de descarga de la página web está directamente relacionado con el tiempo de descarga de cada uno de estos objetos que la forman. Se pueden distinguir dos componentes principales del tiempo de descarga de un objeto web: por una parte el tiempo que tarda la señal en ir del cliente al servidor y volver (conocido como RTT) y por otra el tiempo de transferencia del objeto (que depende del ancho de banda entre cliente y servidor). El desarrollo tecnológico ha permitido reducir ambos tiempos, especialmente el dedicado a la transferencia del objeto. Esto ha hecho que el peso de ambos componentes haya variado sensiblemente a lo largo de los años.

Como ejemplo, supongamos que un cliente va a descargar un archivo de 10KB. Teniendo en cuenta los datos que ofrece la bibliografía sobra las conexiones a internet en 1996, un valor típico de ancho de banda sería 33Kbps, mientras que para el RTT sería de 1.130 ms [7]. Descargar 10KB a través de una conexión de 33Kbps, suponiendo que no hay sobrecarga de ningún tipo, tardaría 2.483 ms. Por lo tanto, la latencia total de descarga del objeto sería de $1.130 + 2.483 = 3.613$ ms. En esta situación, el tiempo de transferencia supone el $2483/3616 = 69\%$ de la latencia total.

En la situación actual, nos encontramos con un ancho de banda típico de 4Mbps y un RTT de 120 ms. El tiempo de transferencia de un objeto de 10KB con la conexión moderna sería de 20 ms que, sumado a los 120 ms del RTT, nos daría una latencia total de objeto de 140 ms. En esta situación, el tiempo de transferencia sólo supone el $20/140 = 14\%$ de la latencia total.

B. Más detalles sobre Web caching

El funcionamiento general de la técnica de Web caching es el descrito en la introducción. Sin embargo, existen otros detalles que son los que hacen que la técnica que proponemos ofrezca resultados importantes. El detalle más importante es el que se refiere al tiempo de caducidad de los objetos en la caché.

Generalmente, cuando se solicita un objeto web, el servidor incorpora información sobre la fecha de caducidad del objeto si se almacena en una caché. Si el usuario accede a un objeto que tiene en caché y está caducado, el navegador debe comprobar previamente que este objeto sigue siendo válido (es decir, que no ha cambiado) antes de servirlo al usuario. Esta comprobación se realiza mediante una petición condicional al servidor web: el cliente solicita el objeto al servidor si el objeto ha cambiado desde la fecha en la que el cliente se descargó el objeto y se almacenó en caché. A esta petición el servidor responde con el objeto si éste ha sido modificado, o bien con una respuesta "Not Modified" si no ha cambiado desde entonces. Con este modo de actuar, lo que se consigue es que, si el objeto no ha cambiado, el usuario se ahorra el tiempo de transferencia del objeto. Teniendo en cuenta la formación del tiempo de descarga de los objetos descrita en la sección anterior, esta técnica puede reducir drásticamente la latencia percibida considerando la situación de 1996 (un 69% en el ejemplo utilizado). Sin embargo, en la situación actual esta reducción de latencia está mucho más limitada (un 14% en el mismo ejemplo).

La elección de la fecha de caducidad del objeto es importante ya que si se da una fecha muy lejana, se corre el peligro de que el navegador muestre al usuario de una versión anterior de la web, mientras

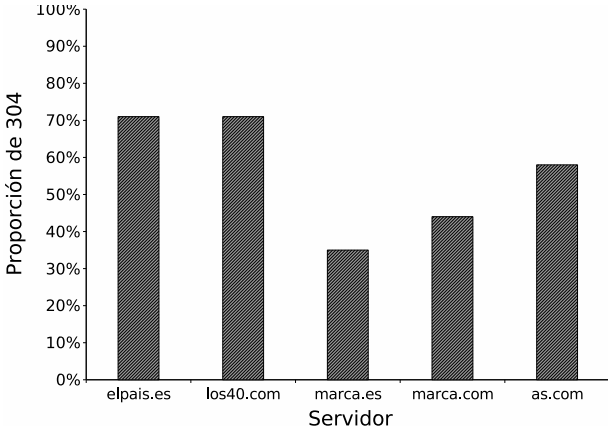

Fig. 2. Proporción de respuestas "304 Not Modified" registradas respecto a todas las recibidas

que si es muy cercana, se reducen los beneficios del Web caching. En la práctica suele primar la consistencia de la web a los beneficios del caching, por lo que se suelen dar tiempos de frescura del objeto muy reducidos, o incluso nulo.

C. Distribución de las peticiones y respuestas web

En este punto vamos a mostrar la frecuencia con la que se responde con un "Not modified" a las peticiones de los clientes. Para ello, vamos a tomar los cinco servidores web más accedidos por la comunidad universitaria de la UPV.

www.elpais.es. El 71% de todas las respuestas recibidas por los clientes tuvieron como respuesta "Not Modified". En el proxy, de las peticiones que se hicieron al exterior (un 31% del total), obtuvieron como respuesta "Not Modified" un 68%.

www.los40.com. El 71% de todas las respuestas recibidas por los clientes tuvieron como respuesta "Not Modified". En el proxy, de las peticiones que se hicieron al exterior (un 62% del total), obtuvieron como respuesta "Not Modified" un 69%.

www.marca.es. El 35% de todas las respuestas recibidas por los clientes tuvieron como respuesta "Not Modified". En el proxy, de las peticiones que se hicieron al exterior (un 21% del total), obtuvieron como respuesta "Not Modified" un 37%.

www.marca.com. El 44% de todas las respuestas recibidas por los clientes tuvieron como respuesta "Not Modified". En el proxy, de las peticiones que se hicieron al exterior (un 24% del total), obtuvieron como respuesta "Not Modified" un 42%.

www.as.com. El 58% de todas las respuestas recibidas por los clientes tuvieron como respuesta "Not Modified". En el proxy, de las peticiones que se hicieron al exterior (un 36% del total), obtuvieron como respuesta "Not Modified" un 86%.

III. Técnica propuesta

Nuestra técnica está encaminada a hacer que los beneficios de las peticiones condicionales del web caching (ver sección II-B) afecten a todo el tiempo de descarga y no sólo al de transferencia. Para ello, emplearemos un sistema de prebúsqueda similar al ya existente que se ha descrito en la introducción. El sistema es igual en la parte predictiva, pero distinto en la parte de ejecución.

La propuesta es la siguiente: junto con cada respuesta, el servidor web incluye una predicción de cuáles van a ser los próximos objetos que va a pedir el usuario. Como es muy probable que estos objetos ya los tenga el navegador del usuario en su caché, el servidor añadirá junto con cada objeto predicho la fecha de última modificación del mismo. Si en un corto espacio de tiempo (definible en el sistema), el usuario trata de acceder al objeto predicho y éste se encuentra caducado en la caché, el navegador compara la fecha de última modificación recibida junto al objeto con la fecha de descarga en caché. Si resulta que el objeto no ha sido modificado desde que fue accedido, ya no será necesario efectuar la petición condicional al servidor, por lo que el usuario se ahorrará el RTT correspondiente a esperar la respuesta "Not Modified" del servidor. Por su parte, el servidor se ahorrará tener que contestar a esta petición, por lo que su carga de trabajo también disminuirá. Esto tendrá efecto positivo tanto en el usuario que se ha beneficiado de la predicción como indirectamente en el resto de usuarios por la disminución de la carga del servidor.

Los experimentos realizados muestran que las mejoras en la latencia percibida por el usuario derivadas del empleo de esta técnica son considerables. Como ejemplo, en la figura 3 se puede apreciar que la latencia percibida por el usuario se puede reducir un 58% y las peticiones al servidor web un 68%. En este ejemplo se ha considerado el servidor web *www.elpais.es* y el algoritmo de predicción DG, ampliamente utilizado en el área de investigación.

En el eje X está representado el índice *Object Traffic Increase*, que es el resultado de dividir el número de peticiones al servidor cuando se utiliza la técnica por el número de peticiones al servidor cuando no se utiliza. Así pues, un valor de 0.32 significa que se han reducido las peticiones del servidor un 68%. En el eje Y representamos el índice *Latency per page ratio*,

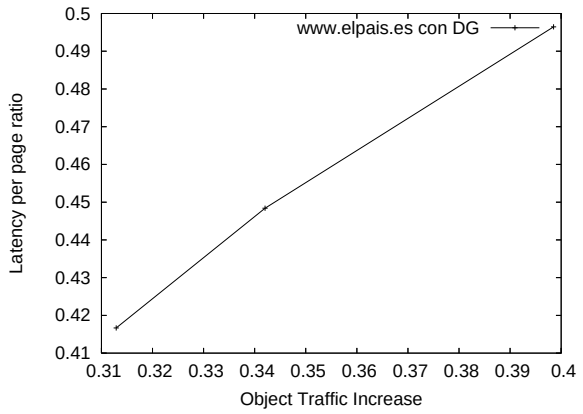

Fig. 3. Resultados al aplicar la técnica propuesta a la web www.elpais.es con el algoritmo de predicción DG

que es el resultado de dividir la latencia percibida cuando se utiliza la técnica por la latencia percibida cuando no se utiliza. De esta forma, un valor de 0.42 significa que se ha reducido la latencia percibida por el usuario un 58%. Cada punto en la línea del gráfico representa una configuración determinada del algoritmo de predicción, por lo que la mejor configuración es la del punto representado en la esquina inferior izquierda de la figura.

IV. Conclusiones

La evolución del hardware de red no sólo ha reducido las latencias percibidas en la web, sino que también ha variado la proporción de cada componente del tiempo de descarga en estas latencias. Esto ha hecho que la efectividad de las peticiones condicionales introducidas en los sistemas de web caching se reduzca sensiblemente en cuanto a reducir la latencia percibida por los usuarios.

Nuestra propuesta utiliza algoritmos de predicción para evitar las peticiones condicionales y las latencias introducidas por éstas. El análisis de la distribución de las respuestas de los servidores nos han mostrado un buen potencial para la técnica de prevalidación, ya que viendo los 5 servidores más populares, entre un 35% y un 71% de todas las respuestas fueron "304 Not Modified". El potencial de la prevalidación se ha confirmado en las simulaciones, ya que hemos obtenido unos resultados en los que la latencia percibida por el usuario se reduce hasta un 58% y el tráfico al servidor –medido en número de peticiones recibidas– se ha reducido un 68%.

Acknowledgments

This work has been partially supported by Spanish Ministry of Education and Science and the European Investment Fund for Regional Development (FEDER) under grant TSI 2005-07876-C03-01.

References

[1] Arun Iyengar, Erich Nahum, Anees Shaikh, and Renu Tewari, "Enhancing web performance," in *Proceedings of the IFIP World Computer Congress*, Montreal, Canada, 2002.

[2] Kirk L. Johnson, John F. Carr, Mark S. Day, and M. Frans Kaashoek, "The measured performance of content distribution networks," in *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, Lisbon, Portugal, 2000.

[3] Shudong Jin and Azer Bestavros, "Popularity-aware greedydual-size web proxy caching algorithms," in *Proceedings of the 20th International Conference on Distributed Computing Systems*, Taipei, Taiwan, 2000.

[4] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu, "Caching on the world wide web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 95–107, 1999.

[5] Josep Domènech, José A. Gil, Julio Sahuquillo, and Ana Pont, "DDG: An efficient prefetching algorithm for current web generation," in *Proceedings of the 1st IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, Boston, USA, 2006.

[6] Darin Fisher and Gagin Saksena, "Link prefetching in Mozilla: A server driven approach," in *Proceedings of the 8th International Workshop on Web Content Caching and Distribution (WCW 2003)*, New York, USA, 2003.

[7] Venkata N. Padmanabhan and Jeffrey C. Mogul, "Using predictive prefetching to improve World Wide Web latency," *Computer Communication Review*, vol. 26, no. 3, pp. 22–36, 1996.