Evaluación de Políticas de Control de Acceso en Cachés Web

F.J. González Cañete, E. Casilari, A. Triviño Cabrera
Dpto. Tecnología Electrónica, Universidad de Málaga
Campus de Teatinos, 29071 Málaga
Telf: 952 137176, Fax: 952131447
fgc@uma.es

Resumen

El presente trabajo evalúa, a través de simulaciones, el rendimiento de una caché que usa políticas de control de acceso de los documentos. Esta política consistirá en no permitir el almacenamiento en caché de los documentos hasta que hayan sido referenciados un número determinado de veces, siendo este número un parámetro de la política de control de acceso. Por un lado se evaluará el rendimiento de la caché usando para ello unas métricas especialmente diseñadas para cachés con control de acceso y por otro lado se evaluará el rendimiento de la política de control de acceso. Para evaluar la política de control de acceso se contabilizarán aquellos documentos que han sido correctamente rechazados y aquellos que han sido correctamente aceptados, obteniéndose así una métrica para la política de control de acceso.

1. Introducción

En todos los sistemas distribuidos se ha intentado aumentar el rendimiento de los mismos usando espacios de almacenamiento intermedios, llamados cachés, para que los datos que más habitualmente se usan estén almacenados en dispositivos más rápidos o más cercanos a los sitios donde van a ser utilizados. La Web, al ser también un sistema distribuido, ha aprovechado este paradigma de varias formas, ya que las cachés usadas en la Web pueden situarse en varios lugares. Por un lado existen las cachés de usuario [1], que son aquellas que están implementadas en el navegador del usuario de Internet y que almacena aquellos documentos que el usuario solicita. Por otro lado existen las cachés del servidor Web [2], que es aquella en la que se almacenan los documentos del servidor Web que son más populares y por tanto más referenciados, y que se almacenan en la memoria física del propio servidor. Por último se encuentran las cachés intermedias o proxy cachés [3], que son aquellas que se sitúan entre los servidores Web y los usuarios. Estas cachés recogen las solicitudes de documentos de los usuarios y la reenvían a los servidores. Una vez que les llega el documento solicitado lo almacenan para que, si otro usuario pidiera ese mismo documento, la caché pueda devolvérselo sin necesidad de solicitársela al servidor Web de origen.

Como ventajas del uso de un proxy cabe mencionar que reduce el tiempo que los usuarios tienen que esperar para recibir los documentos que ya se encuentran en la caché, ya que el proxy se sitúa más cerca que el servidor Web original. Además, en el caso de que el servidor Web al que se le solicitan los documentos se hubiera caído, el proxy podría seguir suministrándolos. Por otro lado, el uso de un proxy reduce el tráfico que circula por Internet, ya que las peticiones y las respuestas no tienen que llegar a los servidores Web. Por otro lado, el uso de proxy cachés también tiene sus inconvenientes, ya que éstos podrían devolver documentos a los usuarios que han sido modificados en el servidor, es decir, la copia de la caché es inconsistente con el documento original. Para evitar este problema de inconsistencia se han propuesto soluciones mediante el uso de políticas de control de consistencia o coherencia [4]. Otro inconveniente que puede aparecer es la pérdida de control por parte de los Web Masters sobre las estadísticas de acceso a su servidor, ya que muchos de los documentos nunca van a llegar a ser solicitados al servidor porque van a ser servidos directamente por la caché.

La función de los proxy cachés es, tal y como se ha resaltado anteriormente, almacenar aquellos documentos que solicitan los usuarios, pero el almacenamiento del que disponen las cachés es finito, y llegará un momento en el que, ante la llegada de un nuevo documento, haya que eliminar alguno de los almacenados debido a que el espacio de almacenamiento se ha llenado. Por tanto, la decisión de qué documento es el que va a ser eliminado es un factor importante, ya que se deberían mantener aquellos documentos que se espera que vayan a ser usados en un futuro con mayor probabilidad y eliminar aquellos que es menos probable que sean referenciados. La política de reemplazo es la encargada de realizar esta decisión que, por desgracia, no es una decisión sencilla, ya que los documentos pueden tener muy diferentes tamaños o ser suministrados por enlaces con muy diferentes anchos de banda. Se han propuesto innumerables políticas de reemplazo [5] [6], cada una de ellas intentando tener en cuenta una o varias de las características del tráfico Web. De entre ellas, las que más destacan son la política de reemplazo LRU (*Least Recently Used*) que elimina de la caché aquellos documentos que hace más tiempo que fueron referenciados, LFU (*Least Frquently Used*) y sus variaciones como LFU-DA (*Least Frequently Used with Dymanic Aging*) [7], que descartan aquellos documentos con un menor número de referencias, o los algoritmos basados en

GDS (*Greedy-Dual Size*) [8] como GDSF (*Gereedy-Dual Size with Frequency*) [9] o GD* [10], que asignan un valor a cada documento en función de parámetros como el tamaño en bytes del mismo, el número de referencias o el coste de traer la copia del servidor. Por mencionar un ejemplo, el proxy caché Squid [11] implementa LRU, LFU-DA y GDSF.

A pesar de la gran cantidad de algoritmos propuestos para políticas de reemplazo, muy pocas propuestas se han hecho sobre políticas de acceso a las cachés Web. Las políticas de acceso son las encargadas de decidir si un documento debe ser almacenado en la caché o no, es decir, si merece la pena almacenarlo. En [2] el autor propone una variación de la política de reemplazo LRU llamada LRU-Threshold, en la que únicamente se almacenan en la caché aquellos documentos cuyo tamaño es inferior a un umbral fijado como parámetro. Esta restricción tiene el inconveniente de que el valor umbral óptimo depende de las características del tráfico. Para solucionar este problema, en [12] se propone el algoritmo LRU-Adaptive que funciona igual que LRU-Threshold pero en el que el umbral es calculado dinámicamente en función de la evolución del rendimiento de la caché. Aggarwal et al [13] propusieron una política de control de admisión que usaba caché LRU auxiliar con metainformación de los documentos como el número de accesos, el momento en que se produjeron los accesos, el coste del acceso y datos sobre expiración. Cuando un documento llega a la caché, la política de admisión comprueba si el documento está en la caché auxiliar. Si es así, el documento entra en la caché principal si el valor heurístico asignado al documento basándose en los metadatos es mayor del documento que sería eliminado de la caché. En otro caso, los metadatos del documento se almacenan en la caché auxiliar, pero no entra en la caché principal.

Además de la escasez de literatura sobre políticas de control de admisión, tampoco se han propuesto métricas para evaluar el rendimiento de las mismas. El presente artículo aporta por un lado unas métricas para la política de control de admisión además de unas métricas adecuadas para evaluar el rendimiento de una caché con dichas políticas de control de admisión.

El resto del artículo está estructurado de la siguiente forma. En el capítulo 2 se describen las características principales de la muestra utilizada en las simulaciones así como el procesamiento al que ha sido sometida. En el capítulo 3 se describen las métricas utilizadas en la evaluación tanto del rendimiento de la caché como de la política de control de acceso. En el capítulo 4 se evalúa mediante las métricas propuestas en el capítulo 3 una caché con la política de control de acceso consistente en no almacenar los documentos hasta que no son solicitados al menos un número determinado de veces. Finalmente, el capítulo 5 recoge las conclusiones y líneas futuras de este trabajo.

2. Procesado y caracterización de las muestras de tráfico

La muestra de tráfico utilizada para el presente trabajo pertenece a un proxy caché Squid situado en el Research Triangle Park en Carolina del Norte (Estados Unidos) entre los días 7 y 11 de Junio del 2004.

En primer lugar la muestra fue procesada para eliminar aquellas respuestas a peticiones generadas dinámicamente mediante CGI (*Common Gateway Interface*) [14]. Para ello se descartaron las peticiones que contenían las cadenas ‘cgi’, ‘cgi-bin’ o ‘?’ ya que estas respuestas no deben ser almacenadas en caché al ser creadas para cada petición y en función de varios parámetros que varían de petición a petición. De los códigos de respuesta se han considerado como cacheables el 200 (OK), 203 (*Partial*), 206 (*Partial Content*), 300 (*Multiple Choices*), 301 (*Moved*) y 302 (*Redirects*) [15] Para aquellas peticiones con código de respuesta 304 (*Not Modified*), los documentos han sido solicitados de nuevo al servidor Web original para averiguar el tamaño real de los mismos, ya que en la muestra aparecía el tamaño de la respuesta. La tabla I resume las características de la muestra de tráfico una vez realizado este proceso. De esta tabla cabe destacar que el 32.1% de las peticiones pertenecen a documentos que aparecen una única vez en la muestra (*one timers*) y que por lo tanto no van a ser útiles en una caché ya que no van a volver a ser accedidos. Una buena política de control de acceso debería ser capaz de no dejar entrar en la caché este tipo de documentos.

En la Fig. 1 se ha representado el histograma del número de referencias a los documentos en función del tamaño de los mismos y puede observarse que aquellos documentos con un mayor tamaño tienen más probabilidad de ser menos referenciados. Esta característica es la que se utiliza como base para la política de control de acceso LRU-Threshold.

3. Métricas utilizadas

3.1. Métricas clásicas en cachés Web

Las métricas que más se han utilizado a la hora de evaluar el rendimiento de una caché Web son el Hit Ratio (HR) y el Byte Hit Ratio (BHR).

Tabla I. Características de la muestra de tráfico

Número de peticiones	4,040,036
Tamaño (GB)	40.4
Documentos distintos	42.4 %
One timers	32.1 %
Media (Bytes)	10,006
Mediana (Bytes)	1,720
Desviación típica(Bytes)	229,941

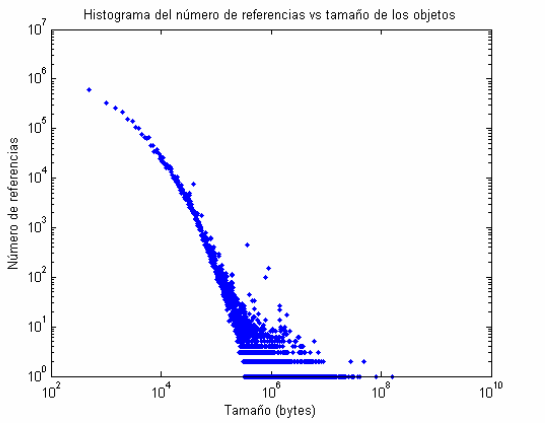

Fig 1. Histograma del número de referencias en función del tamaño de los documentos.

El HR está definido como el cociente entre el número de aciertos en la caché y el número de peticiones que ha recibido. Es decir, el HR mide la proporción de peticiones que ha servido la caché sin necesidad de acceder al servidor Web original. El HR nos da una idea de la reducción de la latencia percibida por los usuarios.

Por otro lado, el BHR está definido como el cociente entre el tráfico que sirve la caché debido a aciertos en la misma y el tráfico total que atraviesa la caché. Es decir, el BHR mide la proporción del tamaño de los documentos que sirve la caché debido a aciertos en la misma. El BHR nos da una idea del ancho de banda que ha ahorrado la caché a los servidores Web.

El inconveniente de estas métricas es que no están adaptadas para la evaluación del rendimiento de una caché con política de control de acceso, ya que tienen en cuenta todas las peticiones que llegan al proxy. Por lo tanto, aquellas peticiones cuyos documentos no han sido aceptados por la política de acceso también son contabilizadas, con los que el rendimiento se reduce al no producir ningún acierto. Sin embargo, el correcto rechazo de estas peticiones no debería reducir el rendimiento de la caché sino todo lo contrario, ya que está siendo capaz de rechazar documentos que no van a causar aciertos.

3.2. Métricas adaptadas

Para solucionar el problema comentado en el apartado anterior se proponen dos métricas para evaluar el rendimiento de una caché que tiene implementada una política de control de acceso. Se define el NUHR (*Not Unique Hit Ratio*) y NUBHR (*Not Unique Byte Hit Ratio*) como se muestra en las ecuaciones 1 y 2 respectivamente.

$$NUHR = \frac{\#\,Aciertos}{\#\,\mathrm{Re}\,q - \#\,R\_ok} \qquad (1)$$

$$NUBHR = \frac{Size\_Aciertos}{Size\_\mathrm{Re}\,q - Size\_R\_ok} \qquad (2)$$

siendo *#Aciertos* en número de aciertos en la caché, *#Req* en número total de peticiones, *#R_ok* el número de documentos que fueron rechazados correctamente por la política de control de acceso. *Size_Aciertos* es el tráfico en bytes que ha servido la caché por los aciertos en la misma, *Size_Req* es el tráfico total que ha atravesado la caché y *Size_R_ok* es el tráfico en bytes que ha sido correctamente rechazado por la política de control de acceso.

Para el caso de que no se utilice una política de control de acceso, los términos *#R_ok* y *Size_R_ok* son cero, por lo que NUHR y NUBHR se reducen a HR y BHR.

La dificultad de esta métrica es decidir cuándo se considera que un documento ha sido correctamente rechazado. Para el presente artículo se ha considerado que un documento es correctamente rechazado cuando no vuelve a aparecer en la muestra de tráfico, es decir, cuando el documento rechazado es un *one timer*. Por tanto, estas métricas miden el rendimiento de la caché con respecto a los documentos no únicos (que aparecen más de una vez en la muestra).

3.3. Métricas de control de acceso

Para medir el rendimiento de las políticas de control de acceso se ha dividido en dos partes. La primera de ellas mide la proporción de documentos correctamente rechazados y por otra la proporción de documentos correctamente aceptados. Como criterio para documentos correctamente rechazados se sigue la propuesta del apartado anterior, mientras que se consideran documentos correctamente aceptados aquellos que la política de control de acceso ha permitido el acceso en la caché y el documento ha sido referenciado al menos otra vez (ha ocasionado al menos un acierto) mientras se encontraba en la caché.

Se proponen por tanto las métricas ACHR (*Access Control Hit Ratio*) y ACBHR (*Access Control Byte Hit Ratio*) que se muestran en las ecuaciones 3 y 4 respectivamente.

$$ACHR = \sqrt{\frac{\#\,R\_ok}{\#\,R\_Total} \cdot \frac{\#\,Ac\_ok}{\#\,Ac\_Total}} \qquad (3)$$

$$ACBHR = \sqrt{\frac{Size\_R\_ok}{Size\_R\_Total} \cdot \frac{Size\_Ac\_ok}{Size\_Ac\_Total}} \qquad (4)$$

siendo *#R_ok* el número de documentos correctamente rechazados por la política de control de acceso, *#R_Total* el número total de documentos rechazados, *#Ac_ok* el número de documentos aceptados para entrar en la caché que causaron al menos un acierto y *#Ac_Total* el número total de documentos que entraron en la caché. De forma similar, los términos de la ecuación 4 hacen referencia al tamaño de los documentos.

Las métricas ACHR y ACBHR hacen una ponderación entre los documentos correctamente rechazados y los correctamente aceptados.

4. Evaluación

Para evaluar el rendimiento de una caché con políticas de control de acceso se ha desarrollado un simulador de una caché que implementa varias políticas de reemplazo así como varias políticas de admisión simples. El simulador toma los ficheros con las muestras de tráfico y, tras realizar la simulación con los parámetros configurados, ofrece como resultado un fichero con datos como el número de documentos que hay en la caché, el número de documentos que han sido eliminados de la misma, el HR y BHR, el NUHR, NUBHR, ACHR y ACBHR si se ha seleccionado alguna política de admisión.

La política de control de acceso que se ha considerado ha sido no permitir el acceso de los documentos a la caché hasta que éstos no han sido referenciados un número determinado de veces, habiéndose hecho las simulaciones para una y dos referencias, es decir, que el documento únicamente es almacenado en la caché a partir de cuando se ha referenciado dos o tres veces respectivamente. La política de reemplazo utilizada ha sido LRU.

Como tamaños de la caché se ha considerado el 10%, 20%, 40%, 60% y 80% de una hipotética caché infinita en el que se almacenaran todos los documentos de la muestra que son referenciados al menos dos veces.

Para intentar evitar la influencia de inicios en frío, es decir, cuando la caché está vacía, se ha usado el 10% de la muestra para "calentar" la caché. Además, para distinguir la modificación de un documento del corte de la transferencia se compara la diferencia entre los tamaños de peticiones sucesivas al mismo documento. Si la diferencia es menor que el 5% del tamaño del documento se considera que ha sido modificado y, por tanto, debe ser considerado como un documento nuevo; en otro caso se considera como una cancelación de la transferencia [16].

El la Fig. 2 se ha representado el HR y BHR de una caché que no implementa ninguna política de acceso, que implementa la política de no permitir el acceso a los documentos en la primera referencia (LRU-1) y en la segunda referencia (LRU-2). Se observa que el HR es mejor con tamaños de caché pequeñas cuando se usa una política de control de acceso, pero este rendimiento se va deteriorando conforme se incrementa el tamaño de la caché. Sin embargo, para el BHR el rendimiento es claramente superior si no se utiliza la política de control de acceso. Este comportamiento, que nos llevaría a pensar que es mejor no usar una política de control de acceso, no es realista debido a que estas métricas se ven influenciadas por el rechazo de documentos, incluso cuando el rechazo es adecuado.

En la Fig. 3 se ha representado el NUHR y el NUBHR y en ellas se observa que realmente el rendimiento es superior al obtenido si no se utiliza la política de control de acceso (hay que tener en cuenta que el NUHR y el NUBHR cuando no se utiliza política de control de acceso equivalen al HR y BHR). Esta diferencia es más notoria en el caso del NUHR, llegando a haber una diferencia de rendimiento de hasta un 20%, mientras que para el NUBHR la diferencia no es tan grande. De este hecho se deduce que las métricas propuestas son más adecuadas para evaluar el rendimiento de una caché que implementa alguna política de control de acceso, ya que no penalizan el no introducir un documento en la caché a no ser que ese documento realmente sea útil para la misma. También se observa, a partir de la figura, que el rendimiento obtenido es mejor cuando se rechazan los documentos la primera vez que son solicitados que si se hace las dos primeras veces.

Por último, la Fig. 4 representa el ACHR y el ACBHR que se obtiene para cada una de las políticas de control de acceso consideradas. Para el ACHR, LRU-2 es mejor para cachés pequeñas, mientras que para cachés grandes es mejor LRU-1, además de que la diferencia de rendimiento se hace cada vez más notoria conforme se aumenta el tamaño de la caché. Para el ACBHR el comportamiento es similar, aunque la diferencia de rendimiento es menos importante.

5. Conclusiones y líneas futuras

En el presente artículo se han propuesto las métricas NUHR y NUBHR para evaluar el rendimiento de una caché que implementa una política de control de acceso, ya que las métricas clásicas, HR y BHR, no tienen en cuenta estas políticas con lo que se obtienen rendimientos que no se corresponden con la realidad. También se han propuesto las métricas ACHR y ACBHR para evaluar el rendimiento de las propias políticas de control de acceso.

Mediante simulaciones y usando una muestra de tráfico real, se ha calculado el rendimiento de una caché que no implementa ninguna política de acceso y se ha comparado con una caché sí que las implementa. Se ha comparado el rendimiento obtenido con las métricas clásicas y con las propuestas en este artículo, demostrándose que las métricas propuestas ofrecen un rendimiento más realista que el que se obtiene con las métricas clásicas. También se han evaluado las políticas de control de acceso usando las métricas propuestas.

Como línea futura se propone refinar la medida de los documentos que son considerados incorrectamente rechazados por la política de control de acceso, teniendo en cuenta la probabilidad de que un documento sea eliminado de la caché antes de que sea accedido de nuevo, en lugar de considerar que un rechazo es incorrecto si el documento vuelve a aparecer en la muestra.

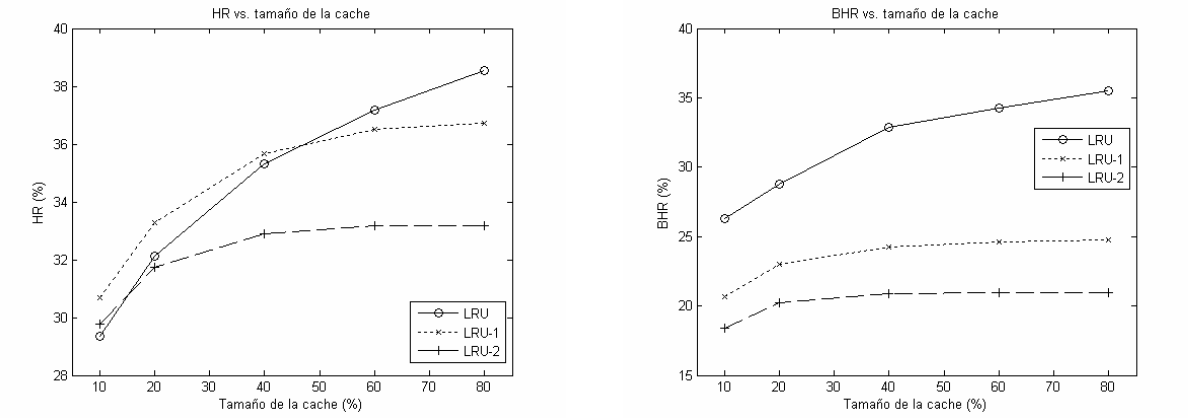

Fig 2. HR (izquierda) y BHR (derecha) de una caché con y sin usar política de admisión.

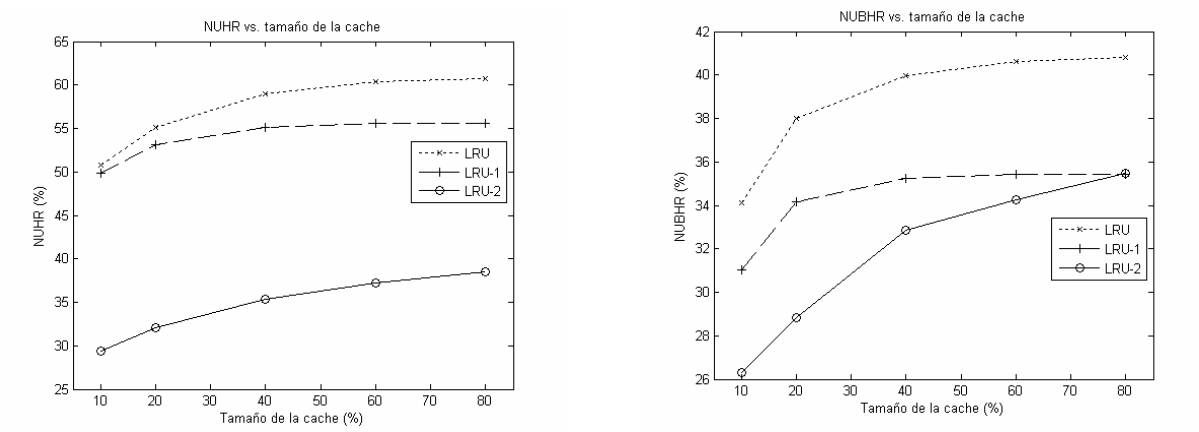

Fig 3. NUHR (izquierda) y NUBHR (derecha) de una caché con política de admisión.

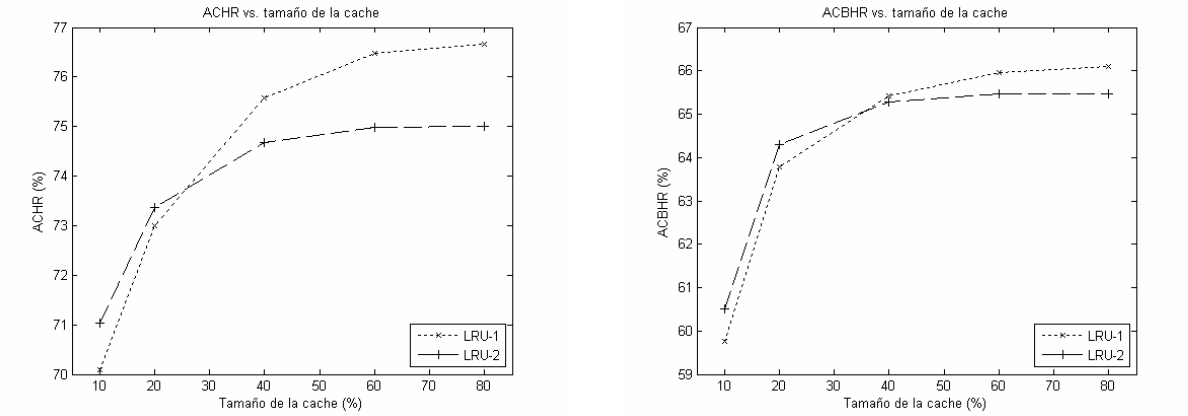

Fig. 4. ACHR (izquierda) y ACBHR (derecha) para las políticas de admisión consideradas.

Agradecimientos

Quisiéramos agradecer a Duane Wessels por facilitarnos el acceso a las muestras de tráfico analizadas.

El presente trabajo ha sido financiado por el proyecto TEL2003-07953-C02-01.

Referencias

[1] K.W. Froese, R.B. Bunt, "The Effect of Client Caching on File Server Workloads", HICSS (1), pp. 150-159, 1996.

[2] E.P. Markatos, "Main Memory Caching of Web Documents", Proceedings del Fifth International World Wide Web Conference on Computer Networks and ISDN Systems, Paris, Francia, 1996.

[3] A. Luotonen, K. Altis, "World-Wide Web Proxies", Proceedings First International Conference on the WWW, 1994.

[4] P. Cao, C. Liu, "Maintaining Strong Cache Consistency in the World-Wide Web", *IEEE Transaction on Computers*, Vol. 47, N. 4, 1998, pp. 445-457.

[5] A. Balamash, M. Krunz, "An Overview of Web Caching Replacement Algorithms", *IEEE Communications Surveys and Tutorials*, Segundo cuatrimestre, 2004.

[6] S. Poplipnig, L. Böszörmenyi, "A Survey of Web Cache Replacement Strategies", *ACM Computing Surveys*, Vol. 35, Nº 4, Diciembre 2003, pp. 374-398.

[7] M. Arlitt, C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *IEEE/ACM Transactions on Networking*, 1997.

[8] P. Cao, "Cost-Aware WWW Proxy Caching Algorithms", Proceedings USENIX Symposium on Internet Technologies and Systems, Monterey, California, Diciembre 1997.

[9] L. Cherkasova, "Improving WWW Proxies Performance with Greedy-Dual-SizeFrequency Caching Policy", Technical Report HP Labs HPL-98-69, Noviembre 1998.

[10] S. Jin, A. Bestabros, "GreedyDual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams", *Journal of Computer Communications*, Vol. 24, No. 2, Febrero 2001, pp. 174-183.

[11] Squid Web Proxy Cache. Disponible en: http://www.squid-cache.org

[12] M. Abrams, C.R. Stanbridge, G. Abdulla, S. Williams, E.A. Fox, "Caching Proxies: Limitations and Potentials", Proceedings del WWW-4 Boston Conference, Deciembre 1995.

[13] C. Aggarwal, J.L. Wolf, P.S Yu, "Caching on the World Wide Web", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, nº 1, Enero/Febrero, 1999, pp. 94-107.

[14] X. Zhang, "Cachability of Web Objects", Technical Report 2000-19, Agosto 2000.

[15] RFC 2616, Hypertext Transfer Protocol – HTTP 1.1., 1999.

[16] M. Arlitt, R. Friedrich, T. Jin., "Workload Characterization of a Web Proxy in a Cable Modem Environment", Technical Report HPL-1999-48, Hewlet-Packard Laboratories, 1999.