(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 585 387**

(51) Int. CI.:

***H04L 29/08*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **29.07.2013 E 13741781 (2)**

(97) Fecha y número de publicación de la concesión europea: **29.06.2016 EP 2880839**

(54) Título: **Método y sistema de almacenamiento en caché web para una red de distribución de contenido (CDN)**

(30) Prioridad:

**02.08.2012 ES 201231263**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**05.10.2016**

(73) Titular/es:

**TELEFÓNICA, S.A. (100.0%)**
**Gran Vía, 28**
**28013 Madrid, ES**

(72) Inventor/es:

**YANG, XIAOYUAN;**
**IVÁN LEVI, MARTÍN;**
**ACOSTA OJEDA, CARMELO ALEXIS;**
**ASTIZ LEZAUN, EGUZKI;**
**GARCIA SANCHEZ MENDOZA, ARMANDO ANTONIO y**
**RODRIGUEZ RODRIGUEZ, PABLO**

(74) Agente/Representante:

**ARIZTI ACHA, Mónica**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

Método y sistema de almacenamiento en caché web para una red de distribución de contenido (CDN)

## DESCRIPCIÓN

### Campo de la técnica

La presente invención se refiere, en general, al almacenamiento en caché de contenido de páginas web en Internet, y más específicamente, en un primer aspecto, a un método de almacenamiento en caché de web para una red de distribución de contenido.

Un segundo aspecto de la invención se refiere a un sistema dispuesto para implementar el método del primer aspecto.

### Estado de la técnica anterior

La entrega del contenido web complejo actual a los usuarios finales es un desafío y a este respecto las redes de distribución de contenido (CDN) han desempeñado un papel clave. Mediante replicación/almacenamiento en caché de contenido web en nodos distribuidos geográficamente, las CDN permiten que aplicaciones en línea accedan a contenido web mucho más rápido, proporcionando una experiencia de mayor calidad a sus usuarios. Las CDN también constituyen una clave para que la infraestructura de Internet evite congestiones en enlaces críticos aislando el tráfico de red en múltiples áreas locales separadas. Esta última característica de las CDN es extremadamente importante para el crecimiento de la infraestructura de interconexión de redes, donde el coste de nuevos enlaces físicos es significativo.

Lo buena que sea una CDN en aceleración de acceso a contenido y aislamiento de tráfico de red depende considerablemente de la capacidad de almacenar en caché el contenido web. Si un contenido web no puede almacenarse en caché, todas las peticiones tendrán que cruzar necesariamente enlaces críticos de vuelta al origen, introduciendo retardos adicionales y consumiendo mayores recursos de red.

La mala noticia es que la baja capacidad de almacenar en caché es una de las características naturales de la web 2.0. Por ejemplo, el 55 % del contenido web ya se indicó potencialmente como sin posibilidad de almacenarse en caché en 2011. Los usuarios de Twitter y Facebook están accediendo a páginas web con información generada de manera constante por otros usuarios en su propia red social en línea. En este escenario, el contenido para cada usuario tiene que etiquetarse como sin posibilidad de almacenarse en caché, puesto que es imposible que los extremos traseros prevean las interacciones de los usuarios finales. Los minoristas en línea tales como eBay [6] cambian el precio del producto según un mecanismo de subasta en el que cada usuario final puede aumentar/reducir el precio final. Como es imposible anticiparse a las subastas en línea, el precio del producto tiene que indicarse como sin posibilidad de almacenarse en caché y los usuarios finales tienen que actualizar constantemente la página web con los detalles del producto para obtener el precio final.

Sin embargo, un objeto web etiquetado como sin posibilidad de almacenarse en caché, no genera necesariamente un contenido diferente en cada petición de usuario. En el caso de eBay, por ejemplo, el precio del producto no cambia si no hay ningún otro usuario que suba el precio durante la subasta. En el caso de Facebook, el muro del usuario no cambia si ningún amigo publica un mensaje. Además, el 46 % del contenido potencialmente que no puede almacenarse en caché no cambia durante 14 días, según algunos estudios.

El contenido pseudo-dinámico puede definirse como todo aquel contenido web que no puede almacenarse en caché que sólo cambia en periodos no determinísticos, dada un área geográfica específica. Sin embargo, no todo el contenido que no puede almacenarse en caché es necesariamente contenido pseudo-dinámico.

Estudios han demostrado que partes significativas del contenido actual que no puede almacenarse en caché es contenido realmente estático. Este contenido se etiqueta como sin posibilidad de almacenarse en caché debido a una mala configuración en el servidor web o a una falta de conocimiento de impacto de rendimiento de los administradores del sistema web. Ya existen soluciones muy buenas para contenido estático que no puede almacenarse en caché en CDN. En la mayoría de las CDN, los clientes pueden especificar valores de TTL fijos para este tipo de contenido. Una vez que un nodo de CDN descarga el contenido del origen, se almacenará en caché durante el periodo de TTL especificado. También existen propuestas como el documento US 2003/0187917 A1 para maximizar la utilidad del contenido almacenado en caché seleccionando la mejor fuente de contenido que se espera que esté actualizado durante más tiempo. El método propuesto en el documento US 2003/0187917 A1 requiere la existencia de múltiples fuentes, por ejemplo múltiples intermediarios y sin embargo, sólo es aplicable para contenido que puede almacenarse en caché.

También existe contenido que no puede almacenarse en caché que es contenido dinámico realmente puro. Por

ejemplo, un contador que indica el número de visitantes de la página web es claramente un contenido dinámico puro. Para cada petición, el contador se incrementará. A pesar de que gran cantidad de contenido que no puede almacenarse en caché es estático o dinámico puro, una parte significativa de este contenido es contenido pseudo-dinámico cuyo evento de cambio no puede anticiparse.

Como respuesta a la explosión de contenido web que no puede almacenarse en caché, se ha propuesto un conjunto de nuevas técnicas de aceleración web. Existen técnicas significativas que suponen que todo el contenido que no puede almacenarse en caché es dinámico puro y contienen diferentes mecanismos de optimización para reducir el tiempo de comunicación con el servidor de origen. Según la característica de mecanismos de optimización, todas las técnicas existentes pueden clasificarse en 3 grupos.

El primer grupo contiene las técnicas que reducen el tráfico de red. Técnicas tales como minimización JS/CSS, optimización de estructura HTML y compresión sobre la marcha pertenecen a esta categoría. En el mismo grupo están aquellas técnicas que pueden detectar duplicidades de contenido web parciales y eliminar parte del tráfico de red enviando sólo mensajes de control.

El segundo grupo incluye todas las técnicas asociadas con la optimización de canales de transmisión, tales como optimización TCP [1], conexiones TCP persistentes [2] y optimización de encaminamiento. El objetivo de estas técnicas es mejorar el rendimiento global de los canales TCP: 1) reduciendo el inicio lento, 2) evitando el retardo de toma de contacto, 3) proporcionando la estabilidad de canal y 4) mejorando el rendimiento global de TCP.

En el tercer grupo están todas aquellas técnicas basadas en la captura previa de contenido del origen. La idea, en este caso, es que los nodos de CDN capturan contenido del servidor de origen al mismo tiempo que se entrega otro contenido al usuario final.

También existen otras técnicas que pueden complementar a todas las técnicas mencionadas anteriormente. La optimización DNS, por ejemplo, forma parte de todas las soluciones de aceleración web, tanto para contenido que puede almacenarse en caché como para el que no. Los administradores web también desempeñan un papel importante en la aceleración de páginas aplicando diferentes técnicas, tales como simplificación DOM y HTML/JS, haciendo que las peticiones AJAX puedan almacenarse en caché, etc.

También existen trabajos en marcha para cambiar el protocolo TCP/HTTP, tal como el protocolo *SPDY* que incluye una gran cantidad de características para la aceleración de sitios web. También existen técnicas específicas para el espacio móvil. Soluciones tales como *Bytemobile* proporcionan una transformación de contenido sobre la marcha para adaptar imágenes a pantallas de dispositivos.

Problemas con las soluciones existentes:

- Aunque la mayoría de las técnicas de aceleración existentes pueden reducir el tiempo de comunicación con el origen, no pueden eliminar completamente la interacción con el servidor de origen. La noción de contenido pseudo-dinámico no se tiene en cuenta en todas las técnicas existentes. Si el contenido se etiqueta como sin posibilidad de almacenarse en caché, todas las peticiones para este contenido requerirán una petición de red de vuelta al origen. En este sentido, la mayoría de las técnicas existentes suponen que el contenido que no puede almacenarse en caché es todo dinámico puro.

Tiempo de ida y vuelta: el requisito de una interacción con el servidor de origen en cada petición limita considerablemente la capacidad de la CDN para reducir el tiempo de respuesta para los usuarios finales. En el mejor de los casos, la CDN seguirá requiriendo un tiempo de ida y vuelta (RTT) completo desde el usuario final al servidor de origen para entregar el contenido solicitado.

Aunque el RTT promedio en Internet actual está disminuyendo con las nuevas infraestructuras de red, la tecnología existente aún requiere 9-30 ms en las comunicaciones de redes locales y 60-200 ms en los enlaces entre países. Como consecuencia, el tiempo de respuesta del contenido que no puede almacenarse en caché puede ser un orden de magnitud superior que el contenido que puede almacenarse en caché en las CDN.

Protocolo TCP: el protocolo TCP se basa en una ventana de congestión (W) [3] [4] para evitar saturaciones sobre enlaces de red. En cada ciclo de transmisión, el emisor envía W paquetes y espera a un paquete de ACK antes de comenzar el siguiente ciclo. El protocolo TCP se optimiza para adaptar la condición de red cambiando dinámicamente el valor de W.

Además, los ciclos de transmisión pueden solaparse, de modo que el nuevo ciclo de transmisión normalmente no se bloquea por el paquete de ACK del ciclo anterior.

La existencia de la ventana de congestión determina la cantidad de datos que pueden enviarse en cada ciclo de

transmisión. Aumentando la ventana de congestión, el emisor puede enviar potencialmente todos los datos en sólo un ciclo. Sin embargo, una ventana de congestión grande puede aumentar potencialmente la tasa de error e implica RTT adicionales para la recuperación de datos. Según [1], una ventana de congestión inicial de 10 paquetes consigue un buen equilibro entre tasa de error y rendimiento. Para contenido web que requiere más de 10 paquetes, todavía se requieren múltiples ciclos de transmisión.
La figura 1 muestra el tiempo de entrega promedio (en RTT) según el tamaño de objeto cuando la latencia de red es de 160 ms. La figura muestra los resultados del protocolo TCP tanto optimizado (inicial W=10) como no optimizado (inicial W = 3). Como puede observarse, la reducción de RTT es significativa para contenido pequeño. Para contenido web grande, la mejora conseguida aumentando la ventana de congestión es sin embargo despreciable.

Reducción de tráfico de red: el rendimiento de las técnicas de reducción de tráfico de red depende de la existencia de redundancia de información. La figura 2 muestra la tasa de compresión de la técnica de compresión sobre la marcha sobre diferentes tipos de contenido web. Según los resultados, la tasa de compresión cambia, según lo esperado, según el tipo de contenido. Por ejemplo, el contenido ya comprimido (que corresponde al 50 % del contenido web), tal como jpeg o gzip, no puede comprimirse más del 10 %. Como consecuencia, el rendimiento de las técnicas de reducción de tráfico de red está limitado.

Todas las limitaciones físicas anteriores más la estabilidad de enlace de red actual determinan el límite superior de las técnicas de aceleración de contenido web existentes que no pueden eliminar por completo la comunicación de red con el servidor de origen.

El documento US 2004/0128346 A1, propone un método para establecer el valor de tiempo de vida (TTL) del contenido que no puede almacenarse en caché basándose en la tasa de petición de usuario para controlar la tasa de error bajo un umbral de probabilidades. Como el método del documento US 2004/0128346 A1, la presente invención también establece valores de TTL a contenido que no puede almacenarse en caché, sin embargo, el diseño de la presente invención difiere del documento US 2004/0128346 A1 en múltiples aspectos. En primer lugar, los proveedores de contenido no son escépticos frente al sistema. Los valores de TTL se ajustan al contenido que no puede almacenarse en caché según las peticiones de los proveedores de contenido. En segundo lugar, los valores de TTL no se calculan según el tiempo promedio entre cambios. En su lugar, se usan predictores de valor para detectar patrones de cambio. En tercer lugar, no elimina por completo la petición al origen convirtiendo el contenido que no puede almacenarse en caché como con posibilidad de almacenarse en caché. En su lugar, todas las peticiones al origen se difieren y el patrón de cambio de contenido puede detectarse rápidamente.

El documento US 2009/150518 A1 propone un método que tiene la capacidad de ensamblar dinámicamente contenido en el límite de Internet, por ejemplo, en servidores de límite de CDN. Para proporcionar esta capacidad, preferentemente el proveedor de contenido aprovecha un lenguaje de guiones de lado del servidor (u otra funcionalidad basada en servidor) para definir fragmentos de página web para ensamblaje dinámico en el límite. En el método propuesto en el documento US 2009/15518, a diferencia de la presente invención, la información que indica si un contenido puede almacenarse o no en caché y durante cuánto tiempo (tiempo equivalente al valor de TTL de la presente invención) se define mediante las cabeceras de HTTP o mediante otras propiedades, por ejemplo definidas mediante metadatos, y por lo tanto se establece estáticamente. En la presente invención, los valores de TTL se adaptan dinámicamente según las peticiones de los usuarios y en consecuencia no están predeterminados.

**Sumario de la invención**

El objeto de la presente invención es proporcionar un método y un sistema para almacenamiento en caché de web para una red de distribución de contenido con el fin de reducir la comunicación de red con el origen para contenido que no puede almacenarse en caché.

El concepto de contenido pseudo-dinámico debe entenderse como todo contenido web que no puede almacenarse en caché que sólo cambia en periodos no determinísticos, dada un área geográfica específica. El contenido para todas las peticiones que pertenecen al mismo periodo y a la misma área geográfica es idéntico. Sin embargo, los servidores web no pueden anticipar cuándo cambiará el contenido, por tanto, el servidor se ve obligado a indicar el contenido como sin posibilidad de almacenarse en caché.

La presente invención proporciona un método de almacenamiento en caché de web para una red de distribución de contenido (CDN), en el que la red de distribución tiene una pluralidad de nodos de almacenamiento en caché y en el que el contenido web se ha identificado estáticamente como contenido que no puede almacenarse en caché y originado en un servidor de origen, como se conoce comúnmente en este campo, comprendiendo el método:

- establecer un valor de periodo de tiempo de vida, TTL, para dicho contenido que no puede almacenarse en caché basándose en peticiones de usuarios; y
- responder a dichas peticiones de usuarios enviando a al menos un usuario de CDN dicho contenido que no

puede almacenarse en caché dentro de dicho valor de periodo de TTL.

En el método, a diferencia de las propuestas conocidas, cada uno de dicha pluralidad de nodos de almacenamiento en caché en dicha red de distribución incluye un gestor de almacenamiento en caché de contenido y un predictor de TTL de contenido pseudo-dinámico y dicho método comprende además realizar las siguientes etapas:

a) poner en contacto cada uno de dicha pluralidad de nodos con un repositorio centralizado, con el fin de descargar el archivo de configuración de una pluralidad de dichos usuarios de CDN.
b) identificar, por parte de dicho gestor de almacenamiento en caché de contenido de cada nodo de almacenamiento en caché, el contenido que no puede almacenarse en caché como contenido pseudo-dinámico;
c) predecir, por parte de un predictor de TTL de contenido pseudo-dinámico de dicho nodo de almacenamiento en caché, dicho valor de periodo de TTL en el que no se modificará el contenido que no puede almacenarse en caché; y
d) almacenar en caché, por parte de cada uno de dicha pluralidad de nodos, el contenido que no puede almacenarse en caché durante dicho valor de periodo de TTL predicho.

Según la invención, dicha etapa c) se realiza para cada uno de dicha pluralidad de usuarios de CDN y se da servicio a todas de dichas peticiones de usuario de dicho contenido pseudo-dinámico usando una copia local del archivo de configuración en dicho repositorio centralizado.

En caso de que dicho contenido que no puede almacenarse en caché no sea pseudo-dinámico, dicho contenido que no puede almacenarse en caché se retransmite a dicho usuario de CDN sin almacenarlo en dicho repositorio centralizado.

El método de almacenamiento en caché de web también comprende generar una petición diferida para dicho servidor de origen para cada petición de usuario de dicho contenido pseudo-dinámico con el fin de separar la descarga de contenido pseudo-dinámico.

Entonces, dichas peticiones diferidas se planifican independientemente y múltiples de las peticiones diferidas se fusionan en una única petición diferida de vuelta para dicho servidor de origen.

El resultado de dichas peticiones diferidas generadas se usa para entrenar a dicho predictor de TTL de contenido pseudo-dinámico usando, en una realización, los últimos valores de TTL de dichas peticiones diferidas generadas.

En una realización, cada predicción de TTL descarga además el contenido que no puede almacenarse en caché para determinar si dicho contenido que no puede almacenarse en caché es realmente estable, y lo compara con la copia local para determinar el último resultado de predicción de TTL.

En otra realización, el valor de periodo de TTL se establece dependiendo de las peticiones de los proveedores de contenido.

Finalmente, el método define etiquetas de versión con el fin de representar diferentes versiones del mismo contenido que no puede almacenarse en caché.

Un segundo aspecto de la invención proporciona un sistema de almacenamiento en caché de web para una red de distribución de contenido (CDN), comprendiendo dicha red de distribución una pluralidad de nodos de almacenamiento en caché y en el que un contenido web se identifica estáticamente como contenido que no puede almacenarse en caché y originado en un servidor de origen, comprendiendo dicho sistema, como es común en este campo:

- un repositorio centralizado, para descargar el archivo de configuración de una pluralidad de usuarios de CDN; y
- medios para establecer un valor de periodo de tiempo de vida (TTL) para dicho contenido que no puede almacenarse en caché basándose en peticiones de usuarios de dicha pluralidad de usuarios de CDN.

A diferencia de las propuestas conocidas, en el sistema de la invención cada nodo de almacenamiento en caché de dicha pluralidad de nodos de almacenamiento en caché en dicha red de distribución comprende:

- un gestor (1) de almacenamiento en caché de contenido dispuesto para identificar dicho contenido que no puede almacenarse en caché como contenido pseudo-dinámico; y
- un predictor (3) de TTL de contenido pseudo-dinámico dispuesto para predecir dicho valor de periodo de TTL en el que no se modificará el contenido que no puede almacenarse en caché,

en el que el contenido que no puede almacenarse en caché se almacena en caché durante dicho valor de periodo de TTL predicho para cada nodo de almacenamiento en caché.

Según la invención, un gestor (4) de peticiones de contenido diferidas está dispuesto para enviar una petición diferida a dicho servidor de origen para cada petición de usuario de dicho contenido pseudo-dinámico.

Además, comprende un elemento (11) de contracción de cola dispuesto para fusionar múltiples de dichas peticiones diferidas en una única petición diferida y un colector (22) de resultados dispuesto para validar el contenido pseudo-dinámico una vez que dicha petición diferida se ha descargado desde dicho servidor de origen.

El sistema del segundo aspecto de la invención implementa el método del primer aspecto de la presente invención.

**Breve descripción de los dibujos**

Las ventajas y características anteriores y otras se entenderán mejor a partir de la siguiente descripción detallada de realizaciones, con referencia a lo que se adjunta, que debe considerarse de una manera ilustrativa y no limitativa, donde:

La figura 1 muestra el tiempo de transmisión de contenido web dependiendo del tamaño del contenido y optimizaciones TCP.
La figura 2 muestra la tasa de compresión de la técnica de compresión sobre la marcha dependiendo del tipo de contenido.
La figura 3 representa los componentes principales usados en la presente invención, según una realización.
La figura 4 es un flujo de trabajo para una petición de contenido nuevo de un usuario final, según una realización.
La figura 5 es un flujo de trabajo para un contenido descargado nuevo del servidor de origen, según una realización.
La figura 6 es el gestor de peticiones de contenido diferidas de la presente invención.
La figura 7 es un flujo de trabajo para el colector de resultados tras descargar el contenido de una petición diferida, según una realización.
La figura 8 muestra el predictor de TTL de contenido pseudo-dinámico de la presente invención.
La figura 9 muestra un diagrama de bloques de la implementación de predictor de valor para la predicción de TTL de contenido, según una realización.
La figura 10 es un flujo de trabajo para que los clientes de la CDN creen un archivo de patrón de contenido pseudo-dinámico, según una realización.

**Descripción detallada de varias realizaciones**

La invención se refiere a un sistema de almacenamiento en caché de web para una CDN que puede reducir significativamente la comunicación de red con el origen para contenido que no puede almacenarse en caché. En particular, los eventos de cambio de contenido que no pueden almacenarse en caché se predicen de manera especulativa, así, la CDN puede entregar este contenido sin interactuar con el origen. Como consecuencia, el tiempo de respuesta del contenido que no puede almacenarse en caché se acelera considerablemente y el tráfico de red en el servidor de origen se reduce de manera eficaz.

La invención incluye mecanismos para permitir a los clientes de la CDN especificar la existencia de contenido pseudo-dinámico. Entonces se usa un mecanismo de predicción para estimar el periodo de cambio de contenido del contenido pseudo-dinámico. Además, la predicción se entrena de manera constante con valores reales, de modo que el sistema puede adaptarse a los patrones de cambio de contenido real.

Se implementa un nuevo mecanismo de almacenamiento en caché de web en una solución de CDN que se compone de múltiples nodos de almacenamiento en caché. Cada nodo de almacenamiento en caché de web entra en contacto con un repositorio centralizado para descargar el archivo de configuración para múltiples clientes de la CDN. Para cada cliente, los nodos de almacenamiento en caché de web ejecutan de manera independiente el mecanismo de predicción para estimar el periodo de cambio del contenido pseudo-dinámico. Una vez que se estima el periodo de cambio, el contenido web se considera como que puede almacenarse en caché para el siguiente periodo predicho. Durante el periodo siguiente, los nodos de almacenamiento en caché darán servicio a todas las peticiones del contenido pseudo-dinámico usando la copia local en el almacenamiento. Además, se genera una petición diferida al servidor de origen para cada petición de usuario final de un contenido pseudo-dinámico. El resultado de la petición diferida se usa entonces para entrenar al mecanismo de predicción.

La figura 3 muestra los componentes principales de la invención. El sistema de almacenamiento en caché de contenido pseudo-dinámico se compone de 4 componentes principales: gestor (1) de almacenamiento en caché de contenido, modulador (2) de contenido pseudo-dinámico, predictor (3) de TTL de contenido pseudo-dinámico y gestor (4) de peticiones de contenido diferidas.

Los componentes se complementan con 3 estructuras de datos (archivo (5) de configuración de patrón pseudo-

dinámico, contenido (6) pseudo-dinámico almacenado en caché e historial (7) de predicción de TTL de contenido pseudo-dinámico) que proporcionan tanto la configuración como el estado para el contenido pseudo-dinámico.

La descripción detallada de todos los módulos se describirá en las siguientes secciones.

Dar servicio a una petición de contenido de un usuario final:

La figura 4 muestra el flujo de trabajo para el nodo de almacenamiento en caché de web para dar servicio a una petición nueva de un usuario final. La primera acción para el nodo de almacenamiento en caché de web es comprobar si el contenido solicitado ya está almacenado en caché en el almacenamiento (2) local. Si el contenido no está almacenado en caché, el nodo de almacenamiento en caché de web lanzará una nueva petición de descarga desde el origen (3). No se realizará ninguna acción más hasta que el origen responda a la petición. En caso de que el contenido esté almacenado en caché, el nodo proporciona el contenido directamente desde el almacenamiento (4) local. Si el contenido almacenado en caché proporcionado es un contenido (5) *pseudo-dinámico*, se introducirá una nueva petición diferida en el *gestor* (6) *de peticiones de contenido diferidas*. De otro modo, no se requerirán más acciones.

Gestor de almacenamiento en caché de contenido:

El componente principal es el *gestor de almacenamiento en caché de contenido (CCM)*. Este componente se encarga de coordinar todo el trabajo de almacenamiento en caché para un contenido descargado nuevo desde el servidor de origen.

La figura 5 muestra el flujo de trabajo de todas las acciones requeridas en el CCM tras descargar un contenido nuevo desde el servidor de origen. En primer lugar, el CCM comprueba la posibilidad de almacenar en caché del contenido nuevo en (2). Si puede almacenarse en caché, se almacenará en caché en el almacenamiento (5) local y el proceso finaliza. En caso de un contenido que no puede almacenarse en caché, el CCM verificará si el contenido es *pseudo-dinámico (3)*. Si no es *pseudo-dinámico*, el contenido sólo se retransmitirá al usuario final sin almacenarlo en el disco local. Si es un contenido *pseudo-dinámico*, el CCM llamará al *predictor* de TTL de *contenido pseudo-dinámico* para estimar el siguiente TTL (4). El TTL predicho se usará para almacenar en caché el contenido en el almacenamiento.

Para cada contenido pseudo-dinámico almacenado en caché, se incluirá una entrada en la estructura de datos de contenido pseudo-dinámico almacenado en caché (6 en la figura 3). Cada entrada incluye los siguientes campos:

1. Identificador de contenido pseudo-dinámico: este identificador da un nombre único a cada contenido pseudo-dinámico descargado.
2. Tiempo almacenado en caché: es el tiempo en el que el contenido se almacena en caché.
3. Tiempo de vencimiento: una marca de fecha y hora que indica cuándo vencerá un contenido. El contenido vencido debe descargarse de vuelta desde el servidor de origen para dar servicio a una nueva petición de usuario final.

Cuando se elimina un contenido pseudo-dinámico de la caché, el CCM también se encarga de eliminar la entrada relacionada de la estructura de datos de contenido pseudo-dinámico almacenado en caché.

Modulador de contenido pseudo-dinámico:

Este módulo identifica si un contenido que no puede almacenarse en caché es o bien un contenido pseudo-dinámico o bien un contenido dinámico puro. El objetivo del diseño de este módulo es proporcionar un mecanismo flexible a los clientes de la CDN para minimizar la probabilidad de identificaciones erróneas de contenido pseudo-dinámico. Con el fin de conseguir este objetivo, se define una estructura de datos simple: archivo de configuración de patrón pseudo-dinámico.

El archivo de configuración de patrón pseudo-dinámico contiene una lista de [P, p, M]. P es una expresión regular que representa el patrón para el identificador de recursos uniforme (URI) de contenido descargado. El segundo elemento de la entrada, $p \in [0,1]$, indica la probabilidad de tener un contenido pseudo-dinámico para todo aquel contenido cuyo URI coincide con el patrón P. El contenido dinámico puro son todos los URI que coinciden con P donde p=0. El tercer elemento es M y representa el valor máximo para el TTL predicho en segundos.

El parámetro p modulará la posibilidad de especulación de la predicción de TTL. Con el parámetro M, los clientes de la CDN pueden especificar la confianza máxima para los valores de TTL especulados. El parámetro p y M se usarán como entrada para que el predictor de TTL coordine/module todo el proceso de predicción.

Gestor de peticiones de contenido diferidas:

El objetivo del gestor de peticiones de contenido diferidas (DCRM) es separar la descarga de contenido pseudo-dinámico (desde el servidor de origen) del servicio a la petición de usuario final, reduciendo así el tiempo de respuesta para contenido pseudo-dinámico.

Para cada acierto de caché de contenido pseudo-dinámico, se genera una petición diferida de vuelta al origen. El módulo de DCRM planifica la ejecución de todas las peticiones diferidas independientemente y puede fusionar múltiples peticiones en una única petición de vuelta al origen. El contenido descargado se compara con el almacenado en caché para determinar el valor de TTL real. El TTL real se usa entonces para entrenar al predictor de TTL para mejores predicciones futuras.

El diseño del DCRM se muestra en una realización en la figura 6. En primer lugar se inserta una nueva petición (7) diferida en la cola (4) de peticiones diferidas. El elemento (1) de contracción de cola se encarga de fusionar múltiples peticiones en una única petición. El elemento (1) de contracción hace esto retardando todas las peticiones en la cola (5) de peticiones contraídas. Para cada petición nueva, el elemento (1) de contracción comprueba en primer lugar si ya hay otra petición en la cola (5) de peticiones contraídas. Si hay otra petición del mismo contenido, la petición nueva se combinará con peticiones previas. Todas las peticiones en la cola (5) de peticiones contraídas se planifican eventualmente mediante un lanzador (3) de peticiones. Se usan múltiples colas en la cola (6) de peticiones en marcha para lanzar la petición de contenido en paralelo. Estableciendo el número de colas de peticiones en marcha, los clientes de la CDN pueden limitar el número máximo de peticiones al origen. Una vez que se descarga el resultado de una petición desde el origen, el colector (2) de resultados notifica el resultado a otros módulos para validar el contenido pseudo-dinámico almacenado en caché. Obsérvese que si se fusionan múltiples peticiones en una petición, se notificarán múltiples resultados de petición al predictor de TTL con el fin de entrenamiento.

La figura 7 muestra más detalles acerca del colector (2) de resultados después de haber finalizado una petición diferida. En primer lugar, se comparará el contenido descargado con el contenido almacenado en caché con el mismo ID pseudo-dinámico. Si no ha cambiado el contenido, el colector (2) de resultados notificará al predictor de TTL de contenido para una predicción (3) positiva. Además, se generará un valor de TTL nuevo mediante el predictor (4) y se actualizará el contenido almacenado en caché con el valor (5) de TTL nuevo. En caso de que el contenido haya cambiado, el colector de resultados notificará en primer lugar al predictor y a continuación invalidará el contenido almacenado en caché actual. El gestor de almacenamiento en caché de contenido insertará el contenido descargado nuevo siguiendo el flujo de trabajo en la figura 5.

Predictor de TTL de contenido pseudo-dinámico:

Dado un contenido pseudo-dinámico, el objetivo de este módulo es predecir el valor del TTL siguiente. Esta es la cantidad de tiempo que se considera un contenido que no puede almacenarse en caché como con posibilidad de almacenarse en caché. El núcleo de este módulo es un predictor [5] [6] [7] de valor. Las entradas de este predictor de valor son:

1. El ID de contenido pseudo-dinámico: que identifica el contenido de manera unívoca.
2. Historial de predicción de TTL de contenido pseudo-dinámico: que contiene toda la estructura de datos requerida para producir el TTL siguiente. En la sección de implementación se proporcionará más información.
3. El valor p y M: son parámetros proporcionados por el modulador de contenido pseudo-dinámico. Estos dos valores modulan la confianza del sistema para producir una predicción para un contenido dado. Si p=0,0, el predictor considerará el contenido puro dinámico y siempre producirá TTL=0. Cuando p=1,0, por otro lado, el predictor confiará completamente en el mecanismo de predicción y producirá valores más optimistas para TTL.

El predictor se entrenará de manera constante mediante los últimos valores de TTL del gestor de peticiones de contenido diferidas. En caso de una predicción errónea, el predictor reducirá el TTL siguiente predicho. De otro modo, se aumentará el TTL siguiente. El parámetro M también modula la predicción limitando el valor máximo del TTL predicho.

Según una realización preferida de la presente invención, esto se implementa con el fin de crear la solución de CDN.

Detalle de implementación de gestor de almacenamiento en caché de contenido:

La solución de almacenamiento en caché de web incluye una lógica para decidir si un contenido nuevo puede almacenarse en caché.

La implementación actual decide si el contenido puede almacenarse o no en caché basándose en los siguientes criterios:

1. Si la petición tiene determinadas cabeceras que evitan la posibilidad de almacenar en caché (es decir, control de caché: no almacenar, etc.), el contenido no se almacena en caché.
2. Si la respuesta tiene un código de estado http diferente de 200, 203, 206, 300, 301, 410, el contenido no se almacena en caché, a menos que las cabeceras de control de caché o vencimiento lo especifiquen.
3. Si la respuesta tiene determinadas directivas de control de caché (privada, no almacenar, no caché), el contenido no se almacena.
4. Si la respuesta tiene determinadas directivas de control de caché (s-maxage, max-age, pública), el contenido se almacena en caché.
5. Si la respuesta tiene cabeceras de vencimiento, el contenido se almacena en caché.
6. En este momento, en ausencia de cabeceras de control de caché y vencimiento, el contenido puede almacenarse en caché con la fecha de vencimiento calculada por los validadores (cabeceras de modificados en último lugar).

Con el fin de soportar contenido específico de usuario-agente, la implementación actual define etiquetas de versión para representar diferentes versiones del mismo contenido. Si el contenido no puede almacenarse en caché, el mecanismo de etiquetas de versión garantiza que la etiqueta para cada petición es diferente.

Detalle de implementación de predictor de TTL de contenido pseudo-dinámico:

La parte principal del predictor de TTL de contenido pseudo-dinámico es el predictor de valor. La implementación actual está inspirada por predictores [5] [6] [7] [9] de intervalo. La idea principal es predecir la cantidad de tiempo que no cambia un contenido pseudo-dinámico, basándose en el historial previo.

En la figura 9 se muestra un diagrama de bloques de la implementación del predictor de valor. La parte principal del predictor es el historial (2) de predicción de TTL de contenido pseudo-dinámico. Habrá una entrada en esta tabla para cada contenido pseudo-dinámico almacenado en caché. Cuando se halla un contenido pseudo-dinámico por primera vez, se inserta una nueva entrada en (2). El campo de valor de TTL y n.º de peticiones de la nueva entrada se pone a 0. El campo de factor p y factor M se establece a la probabilidad de contenido pseudo-dinámico correspondiente y el valor máximo para el TTL predicho, como se proporciona por el modulador de contenido pseudo-dinámico. El campo de intervalo de la nueva entrada se pone a 0.

Por defecto, cada contenido pseudo-dinámico almacenado en caché se predice de manera especulativa como estable (es decir, el contenido es el mismo que el almacenado en caché). Para cada predicción, se requiere una descarga de contenido con el fin de determinar si el contenido es realmente estable (o si se ha predicho de manera errónea). Esta descarga de contenido se retarda por como máximo el TTL predicho. Una vez descargado el contenido, se compara con la copia almacenada en caché para determinar el último resultado de predicción: acierto o error. Usando esta información, la lógica de entrenamiento actualiza el intervalo y valor correspondiente. Los aciertos de predicción aumentan el valor de predicción mientras que los errores lo disminuyen. El valor de predicción se aumenta o disminuye según el intervalo correspondiente. El intervalo se modifica según el factor p, factor M correspondiente y los últimos resultados de predicción. Los valores de factor p elevados aumentan o disminuyen el intervalo más rápidamente que los valores de factor p bajos. El campo de n.º de petición cuenta el número de petición del contenido. Este campo se usa para controlar las primeras predicciones cuando el predictor de valor todavía no tiene ningún historial acerca del contenido pseudo-dinámico. Si el n.º de petición <3, el valor de TTL predicho siempre es 0.

Es posible incorporar estimadores de confianza, tal como se indica en [10], con el fin de mejorar el tiempo de entrenamiento de predicción.

Detalle de implementación de gestor de peticiones de contenido diferidas:

Cada elemento en cola de peticiones diferidas contiene toda la información necesaria para generar una petición al origen que incluye:

1. URL: para el contenido web que va a solicitarse.
2. Cabeceras de petición: todas las cabeceras en la petición de HTTP de usuario, incluyendo las cookies y otras cabeceras en la norma HTTP [8].
3. Dirección IP: del usuario final.
4. Tiempo de petición: una marca de fecha y hora que indica cuándo se realiza la petición.

La estructura de datos para los elementos en cola de peticiones contraídas es una lista de elementos en la cola de peticiones diferidas, clasificados por el tiempo de petición. El lanzador de peticiones inicia una petición al origen cuando la primera petición de usuario final se ha puesto en cola en la cola de peticiones contraídas durante más de min(TTL, Q) segundos. En la implementación actual, usa Q=5 segundos fijos, aunque puede extenderse hasta un valor dinámico en implementaciones futuras. Un valor dinámico podrá adaptarse a la tasa de petición para conseguir

un buen equilibrio entre el tráfico en origen y la precisión de predicción de TTL.

En la implementación actual, el cliente de CDN proporciona la capacidad para configurar un número máximo de peticiones en marcha en la cola de peticiones en marcha. El valor por defecto se fija en 50. El cliente tiene la posibilidad de limitar el número máximo de peticiones paralelas desde cada nodo de almacenamiento en caché de web.

Ventajas de la invención:

La presente invención aprovecha de manera eficaz la existencia de contenido pseudo-dinámico en Internet actual para reducir la carga en el servidor de origen así como proporcionar un tiempo de respuesta extraordinariamente bajo a los usuarios finales.

Tiempo de respuesta bajo: en comparación con otras técnicas de aceleración para contenido que no puede almacenarse en caché, la invención proporciona un enfoque completamente nuevo para reducir el tiempo de respuesta en la entrega de contenido web. En lugar de centrarse en reducir el RTT de la red y el número de ciclos de transferencia en el protocolo TCP, el enfoque de la invención proporciona el mecanismo para separar la dependencia de los datos convirtiendo el contenido que no puede almacenarse en caché en contenido que puede almacenarse en caché. Como resultado, el enfoque de la invención puede eliminar por completo todos los retardos de red asociados con la primera milla, siendo la primera milla la red entre el servidor de origen y el nodo de almacenamiento en caché de web.

La proporción de reducción del tiempo de respuesta de la invención, en comparación con la mejor técnica de aceleración existente para contenido pseudo-dinámico muestra que la mejor técnica existente puede enviar de manera ideal el contenido al usuario final en sólo un RTT (retardo de primera milla + retardo de última milla). La presente invención, por otro lado, puede eliminar completamente el retardo en la primera milla, dando como resultado una gran proporción de reducción del tiempo de respuesta. Por ejemplo, si el servidor de origen está en EE.UU., el nodo de almacenamiento en caché de web en la UE puede reducir el tiempo de respuesta a la UE, en este caso el retardo de primera milla es de aproximadamente 170 ms y el retardo de última milla es de aproximadamente 40 ms, de los usuarios finales aproximadamente en un 80 %.

Baja carga de red en servidor de origen: la separación de la dependencia de los datos permite al sistema diferir la petición de datos a los servidores de origen. Entonces, pueden fusionarse múltiples peticiones diferidas en una única petición en el gestor de peticiones de contenido diferidas, dando como resultado una reducción de carga significativa en el servidor de origen.

Alta flexibilidad: el archivo de patrón de contenido pseudo-dinámico proporciona un mecanismo extremadamente flexible para que los clientes de la CDN usen el presente sistema de almacenamiento en caché de web especulativo. Los clientes pueden conectar/desconectar potencialmente los mecanismos de predicciones de TTL para cada contenido web. La sintaxis del archivo de configuración de la invención es simple para los clientes y tanto el parámetro p como M pueden determinarse fácilmente teniendo en cuenta la lógica de aplicación.

La figura 12 muestra el flujo de trabajo para que los clientes de la CDN creen el archivo de patrón de contenido pseudo-dinámico inicial. La idea básica es separar en primer lugar todo el contenido (1)(2) dinámico puro y a continuación agrupar otro contenido pseudo-dinámico según el TTL (3) promedio esperado. Para cada grupo, fijar el parámetro M para que sea el TTL (5) promedio y calcular el valor para p según alguna métrica (6) de variación de TTL esperada. Dado el archivo de patrón inicial, el cliente puede ajustar el sistema subdividiendo grupos según variaciones de TTL. Por ejemplo, los clientes pueden separar todo el contenido pseudo-dinámico con un periodo de cambio muy predecible de todo aquél con el mismo TTL promedio y configurar un valor muy alto para el parámetro p.

Alta precisión en predicción de TTL: el predictor de valor es muy eficaz para predecir el TTL para contenido pseudo-dinámico. El comportamiento del predictor de TTL depende del parámetro p. Por ejemplo, en una realización, el TTL del contenido pseudo-dinámico se muestra como siguiendo un patrón de 3 etapas, donde el contenido es dinámico puro de 8 a.m. a 10 a.m. El predictor de TTL puede adaptar la tendencia de variación de TTL en las 3 etapas. Esto es posible porque el predictor de intervalo puede representar las tendencias de variación de valor. Usando el parámetro p, puede cambiarse de manera eficaz el comportamiento del predictor de TTL. Usando un factor p mayor, el predictor de valor produce resultados más especulativos, mientras que un factor p bajo hace que el predictor de valor sea más conservativo. El parámetro M también permite que la invención limite el TTL predicho para que sea menos de 200 segundos.

Siglas

CDN Content Distribution Network; red de distribución de contenido

HTTP Hypertext Transfer Protocol; protocolo de transferencia de hipertexto
TTL Time to Live; tiempo de vida
URI Uniform Resource Identifier; identificador de recursos uniforme
URL Uniform Resource Locator; localizador de recursos uniforme

Bibliografía

[1] Dukkipati, N. e. (julio de 2010). An Argument for Increasing TCP's Initial Congestion Window. ACM Computer Communication Review.
[2] Abhinav Pathak, Y. A. (2010). Measuring and evaluating TCP Splitting for Cloud Services. In Proceedings of the 11th Passive and Active Measurement Conference (PAM 2010)
[3 Vinton G. Cerf, Robert E. Kahn, (mayo de 1974). “A Protocol for Packet Network Intercommunication”. IEEE Transactions on Communications 22 (5): 637-648.
[4] RFC 793
[5] Jean-Loup Baer y Tien-Fu Chen. 1995. Effective Hardware-Based Data Prefetching for High-Performance Processors. IEEE Trans. Comput. 44, 5 (mayo de 1995), 609-623.
[6] Yiannakis Sazeides y James E. Smith. 1997. The predictability of data values. In Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture (MICRO 30). IEEE Computer Society, Washington, DC, Estados Unidos, 248-258.
[7] Srikanth T. Srinivasan y Alvin R. Lebeck. 1998. Load latency tolerance in dynamically scheduled processors. In Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture (MICRO 31). IEEE Computer Society Press, Los Alamitos, CA, Estados Unidos, 148-159.
[8] RFC 2616
[9] Gabbay, F., Mendelson, A.: Speculative Execution Based on Value Prediction. Technical, Informe N.º 1080, Technion, Electrical Engineering Department. (1996)
[10] Aragon, J.L., Gonzalez, J., García, J.M., Gonzalez, A. : Confidence Estimation for Branch Prediction Reversal. In Proc. of the Int. Conference on High Performance Computing (2001).

**REIVINDICACIONES**

1. Un método de almacenamiento en caché de web para una red de distribución de contenido (CDN), comprendiendo dicha red de distribución una pluralidad de nodos de almacenamiento en caché y en el que el contenido web se ha identificado estáticamente como contenido que no puede almacenarse en caché y originado en un servidor de origen, comprendiendo el método:

- establecer un valor de periodo de tiempo de vida, TTL, para dicho contenido que no puede almacenarse en caché basándose en peticiones de usuarios; y
- responder a dichas peticiones de usuarios enviando a al menos un usuario de CDN dicho contenido que no puede almacenarse en caché dentro de dicho valor de periodo de TTL,

**caracterizado por que** cada uno de dicha pluralidad de nodos de almacenamiento en caché en dicha red de distribución incluye un gestor de almacenamiento en caché de contenido y un predictor de TTL de contenido pseudo-dinámico y **por que** dicho método comprende las siguientes etapas:

a) poner en contacto cada uno de dicha pluralidad de nodos con un repositorio centralizado, con el fin de descargar un archivo de configuración de una pluralidad de dichos usuarios de CDN.
b) identificar, por parte de dicho gestor de almacenamiento en caché de contenido de cada nodo de almacenamiento en caché, el contenido que no puede almacenarse en caché como contenido pseudo-dinámico;
c) predecir, por parte de un predictor de TTL de contenido pseudo-dinámico de dicho nodo de almacenamiento en caché, dicho valor de periodo de TTL en el que no se modificará el contenido que no puede almacenarse en caché, prediciéndose el valor de periodo de TTL por medio del predictor de TTL de contenido pseudo-dinámico al menos considerando:

- un identificador que identifica el contenido que no puede almacenarse en caché,
- estructura de datos que contiene historial de predicción anterior y que se requiere para producir el valor de periodo de TTL, y
- dos parámetros, p y M, proporcionados mediante un modulador de contenido pseudo-dinámico, que modulan confianza de la predicción, modulando dicho parámetro p la posibilidad de especulación de la predicción de TTL y especificando dicho parámetro M la confianza máxima para valores de TTL especulados; y

d) almacenar en caché, por parte de cada uno de dicha pluralidad de nodos, el contenido que no puede almacenarse en caché durante dicho valor de periodo de TTL predicho.

2. El método de almacenamiento en caché de web según la reivindicación 1, **caracterizado por que** dicha etapa c) se realiza para cada uno de dicha pluralidad de usuarios de CDN.

3. El método de almacenamiento en caché de web según la reivindicación 2, **caracterizado por que** comprende dar servicio a todas de dichas peticiones de dichos usuarios de dicho contenido pseudo-dinámico usando una copia local del archivo de configuración en dicho repositorio centralizado.

4. El método de almacenamiento en caché de web según cualquiera de las reivindicaciones anteriores, **caracterizado por que** comprende retransmitir dicho contenido que no puede almacenarse en caché a dicho usuario de CDN sin almacenarlo en dicho repositorio centralizado si dicho contenido que no puede almacenarse en caché no es pseudo-dinámico.

5. El método de almacenamiento en caché de web según la reivindicación 1 o 3, **caracterizado por que** comprende además generar, por parte de un gestor de peticiones de contenido diferidas, una petición diferida para dicho servidor de origen para cada petición de usuario de dicho contenido pseudo-dinámico con el fin de separar la descarga de contenido pseudo-dinámico.

6. El método de almacenamiento en caché de web según la reivindicación 5, **caracterizado por que** comprende además planificar de manera independiente dichas peticiones diferidas y fusionar múltiples de dichas peticiones diferidas en una única petición diferida de vuelta para dicho servidor de origen.

7. El método de almacenamiento en caché de web según la reivindicación 6, **caracterizado por que** comprende usar el resultado de dichas peticiones diferidas generadas para entrenar dicho predictor de TTL de contenido pseudo-dinámico.

8. El método de almacenamiento en caché de web según la reivindicación 7, **caracterizado por que** comprende entrenar el predictor de TTL de contenido pseudo-dinámico con los últimos valores de TTL de dichas peticiones diferidas generadas.

9. El método de almacenamiento en caché de web según las reivindicaciones anteriores, **caracterizado por que** para cada predicción de TTL comprende además descargar el contenido que no puede almacenarse en caché para determinar si dicho contenido que no puede almacenarse en caché es realmente estable, y compararlo con la copia local para determinar el último resultado de predicción de TTL.

10. El método de almacenamiento en caché de web según la reivindicación 1, **caracterizado por que** comprende establecer dicho valor de TTL dependiendo de las peticiones de los proveedores de contenido.

11. El método de almacenamiento en caché de web según la reivindicación 1, **caracterizado por que** define etiquetas de versión con el fin de representar diferentes versiones del mismo contenido que no puede almacenarse en caché.

12. Un sistema de almacenamiento en caché de web para una red de distribución de contenido (CDN), comprendiendo dicha red de distribución una pluralidad de nodos de almacenamiento en caché y en el que un contenido web se identifica estáticamente como contenido que no puede almacenarse en caché y origina en un servidor de origen, comprendiendo dicho sistema:

- un repositorio centralizado, para descargar el archivo de configuración de una pluralidad de usuarios de CDN; y
- medios para establecer un valor de periodo de tiempo de vida (TTL) para dicho contenido que no puede almacenarse en caché basándose en peticiones de usuarios de dicha pluralidad de usuarios de CDN,

**caracterizado por que** cada nodo de almacenamiento en caché de dicha pluralidad de nodos de almacenamiento en caché en dicha red de distribución comprende:

- un gestor (1) de almacenamiento en caché de contenido dispuesto para identificar dicho contenido que no puede almacenarse en caché como contenido pseudo-dinámico; y
- un predictor (3) de TTL de contenido pseudo-dinámico dispuesto para predecir dicho valor de periodo de TTL en el que no se modificará el contenido que no puede almacenarse en caché por medio de considerar al menos:

  - un identificador que identifica el contenido que no puede almacenarse en caché,
  - estructura de datos que contiene historial de predicción anterior y que se requiere para producir el valor de periodo de TTL, y
  - dos parámetros, p y M, proporcionados mediante un modulador de contenido pseudo-dinámico, que modulan la confianza de la predicción, modulando dicho parámetro p la posibilidad de especulación de la predicción de TTL y especificando dicho parámetro M la confianza máxima para valores de TTL especulados; y

en el que el contenido que no puede almacenarse en caché se almacena en caché durante dicho valor de periodo de TTL predicho para cada nodo de almacenamiento en caché.

13. El sistema de almacenamiento en caché de web según la reivindicación 12, **caracterizado por que** comprende un gestor (4) de peticiones de contenido diferidas dispuesto para enviar una petición diferida a dicho servidor de origen para cada petición de usuario de dicho contenido pseudo-dinámico.

14. El sistema de almacenamiento en caché de web según la reivindicación 13, **caracterizado por que** comprende un elemento (11) de contracción de cola dispuesto para fusionar múltiples de dichas peticiones diferidas en una única petición diferida.

15. El sistema de almacenamiento en caché de web según la reivindicación 14, **caracterizado por que** comprende además un colector (22) de resultados dispuesto para validar el contenido pseudo-dinámico una vez que dicha petición diferida se ha descargado desde dicho servidor de origen.

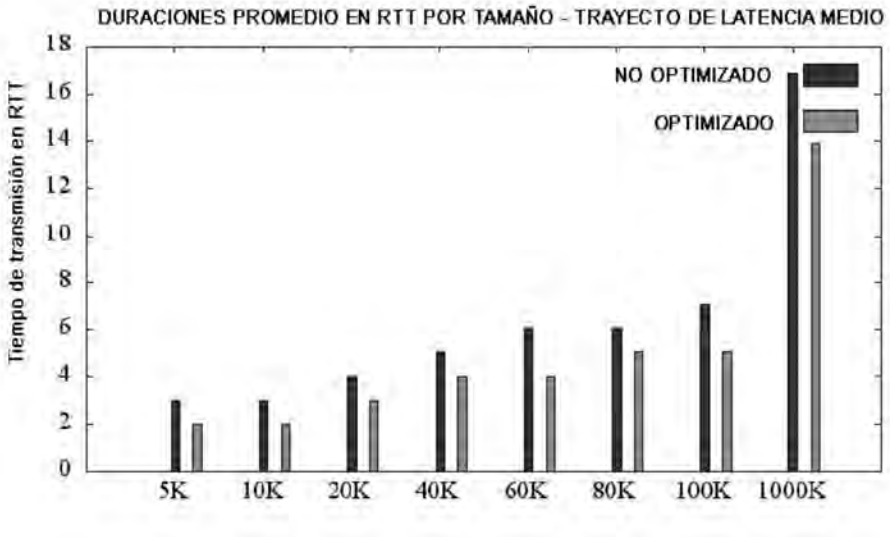


Figura 1

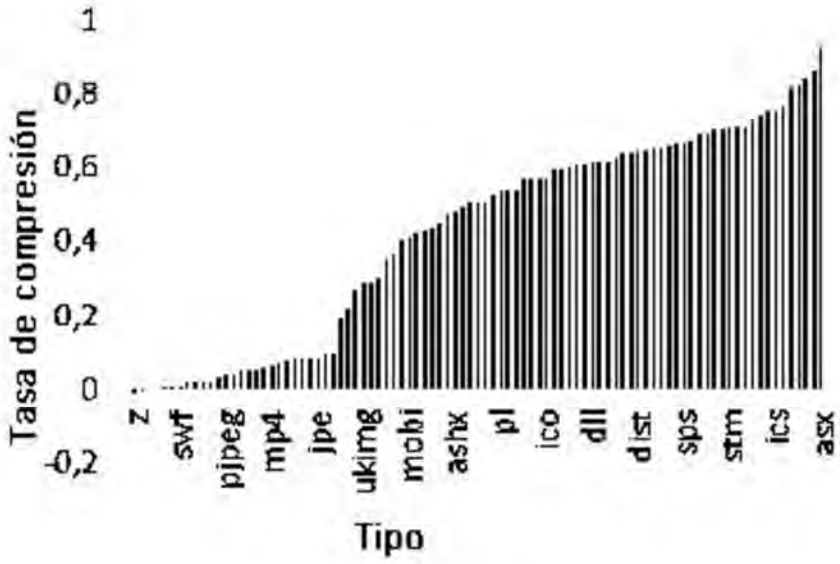


Figura 2

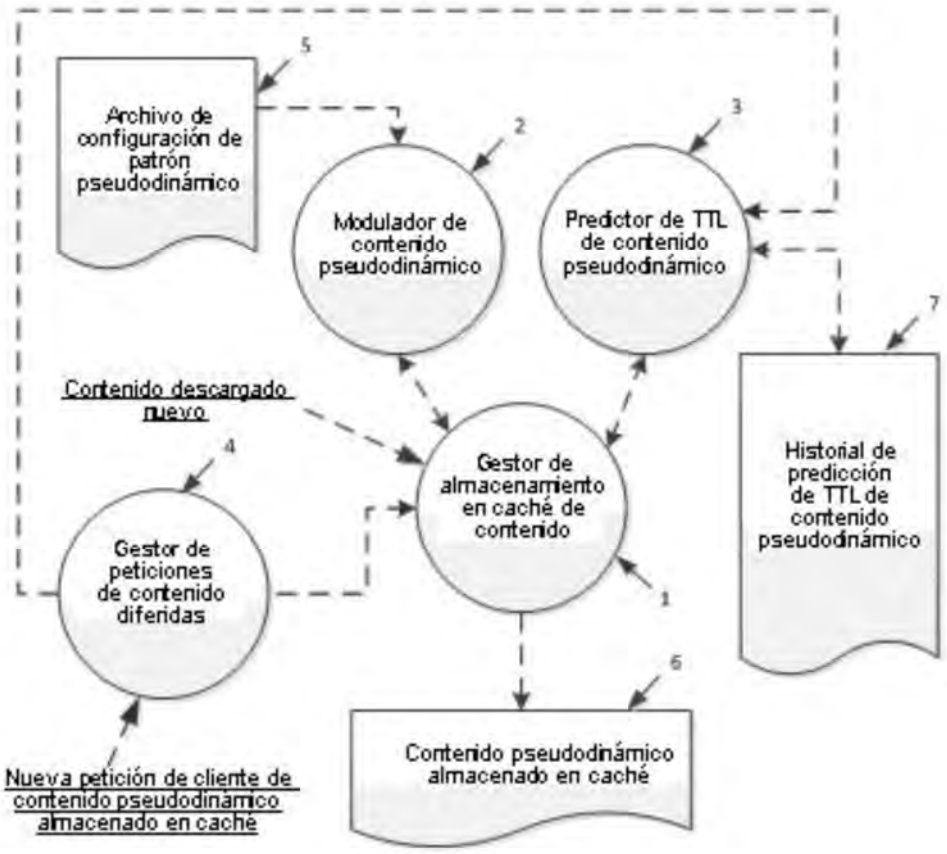
5
Archivo de configuración de patrón pseudodinámico
2
Modulador de contenido pseudodinámico
3
Predictor de TTL de contenido pseudodinámico
7
Historial de predicción de TTL de contenido pseudodinámico
Contenido descargado nuevo
4
Gestor de peticiones de contenido diferidas
Gestor de almacenamiento en caché de contenido
1
6
Contenido pseudodinámico almacenado en caché
Nueva petición de cliente de contenido pseudodinámico almacenado en caché

**Figura 3**

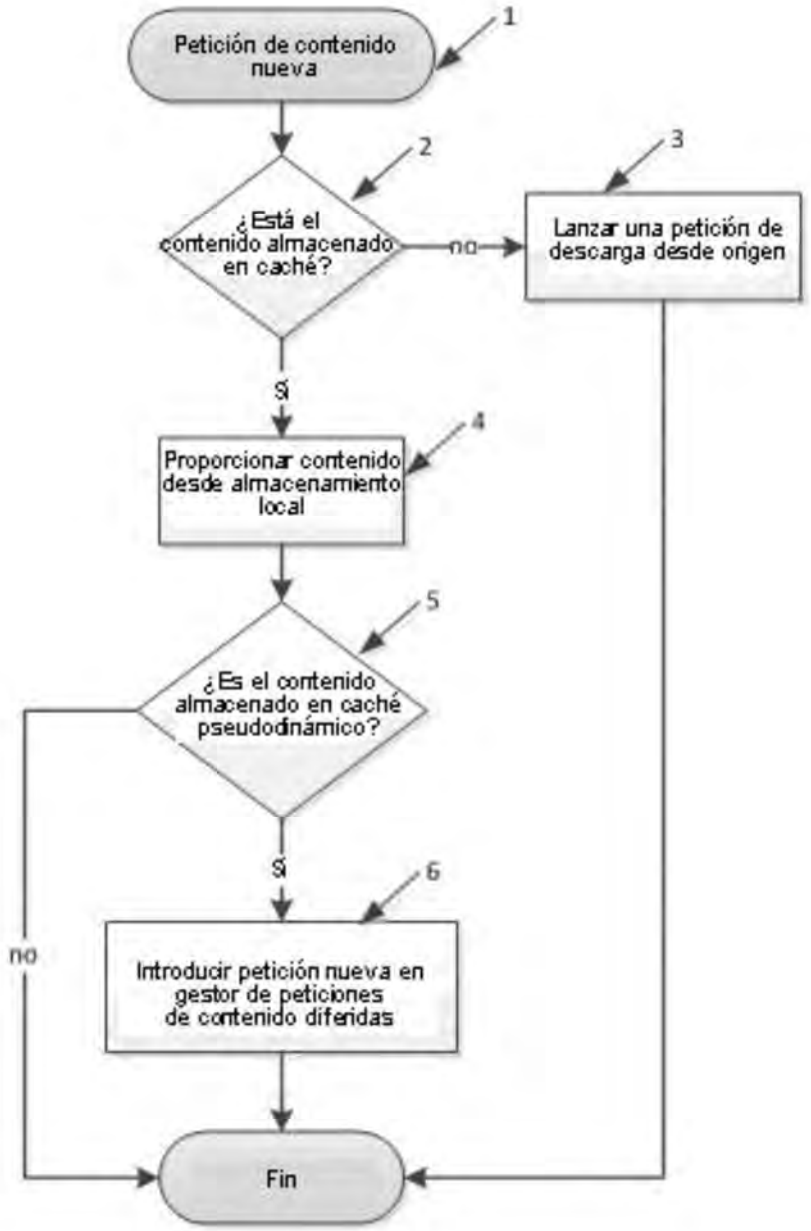
1
Petición de contenido nueva
2
¿Está el contenido almacenado en caché?
no
3
Lanzar una petición de descarga desde origen
Sí
4
Proporcionar contenido desde almacenamiento local
5
¿Es el contenido almacenado en caché pseudodinámico?
Sí
6
no
Introducir petición nueva en gestor de peticiones de contenido diferidas
Fin

**Figura 4**

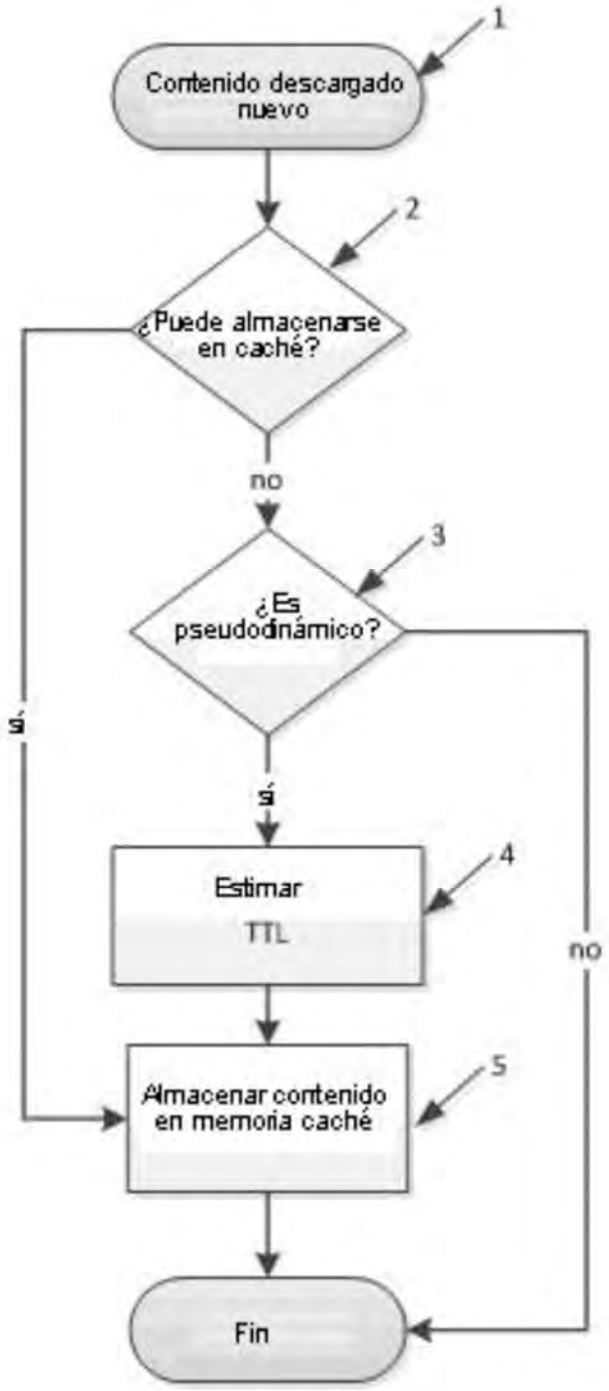
1
Contenido descargado nuevo
2
¿Puede almacenarse en caché?
no
3
¿Es pseudodinámico?
sí
sí
4
Estimar
TTL
no
5
Almacenar contenido en memoria caché
Fin

**Figura 5**

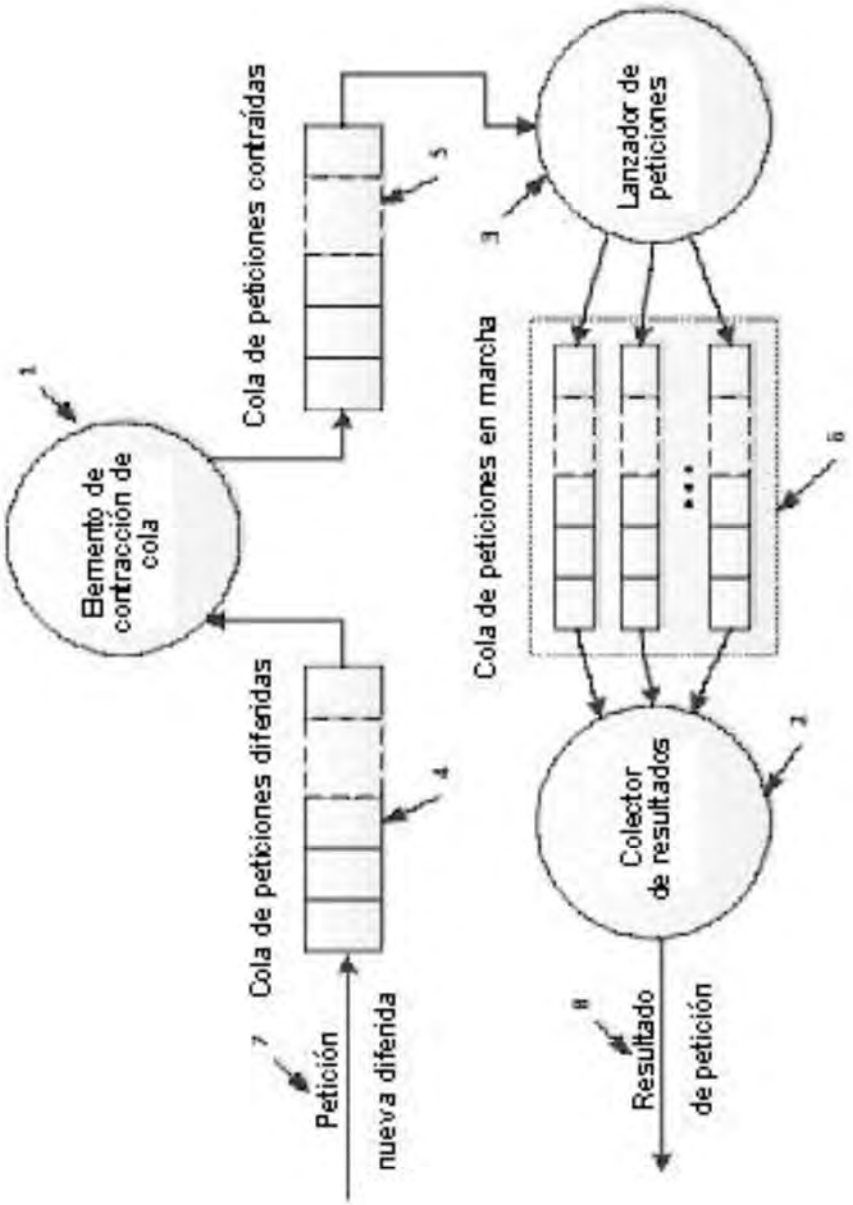
Elemento de contracción de cola
1
Cola de peticiones contraídas
5
Lanzador de peticiones
3
Cola de peticiones en marcha
6
Cola de peticiones diferidas
4
Colector de resultados
2
7
Petición
nueva diferida
8
Resultado
de petición

Figura 6

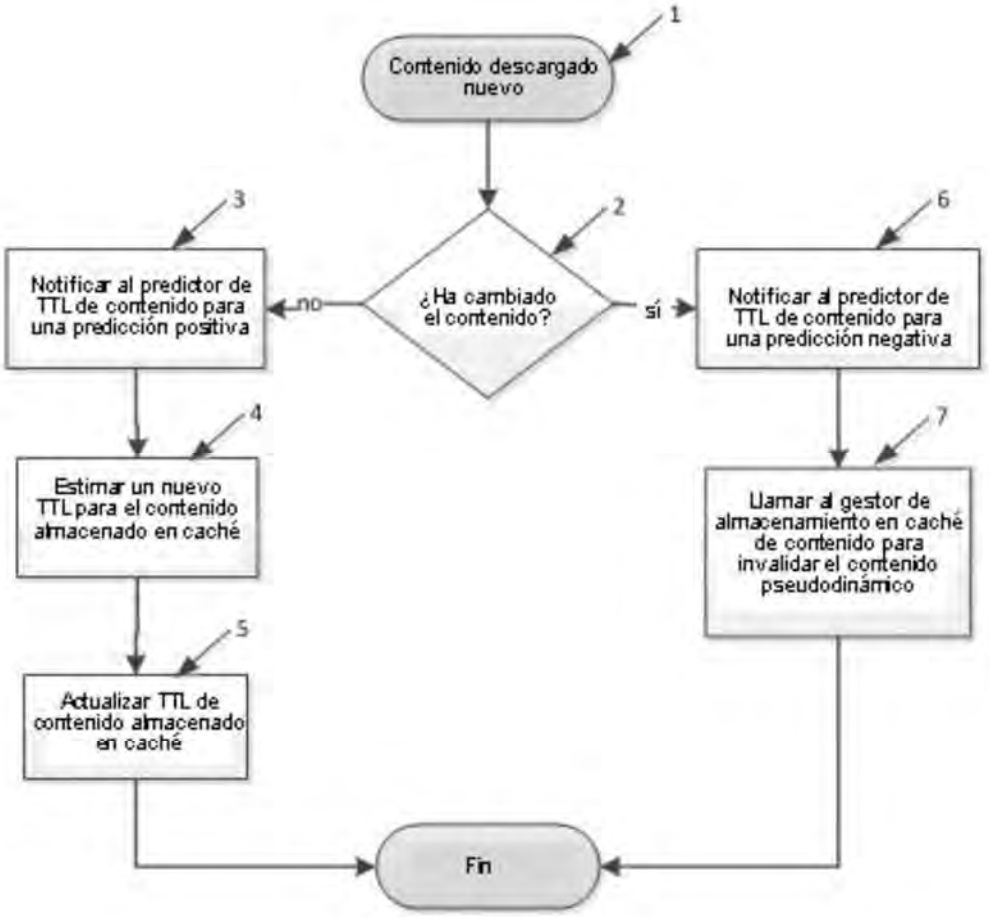
1
Contenido descargado nuevo
2
¿Ha cambiado el contenido?
no
sí
3
Notificar al predictor de TTL de contenido para una predicción positiva
4
Estimar un nuevo TTL para el contenido almacenado en caché
5
Actualizar TTL de contenido almacenado en caché
6
Notificar al predictor de TTL de contenido para una predicción negativa
7
Llamar al gestor de almacenamiento en caché de contenido para invalidar el contenido pseudodinámico
Fin

**Figura 7**

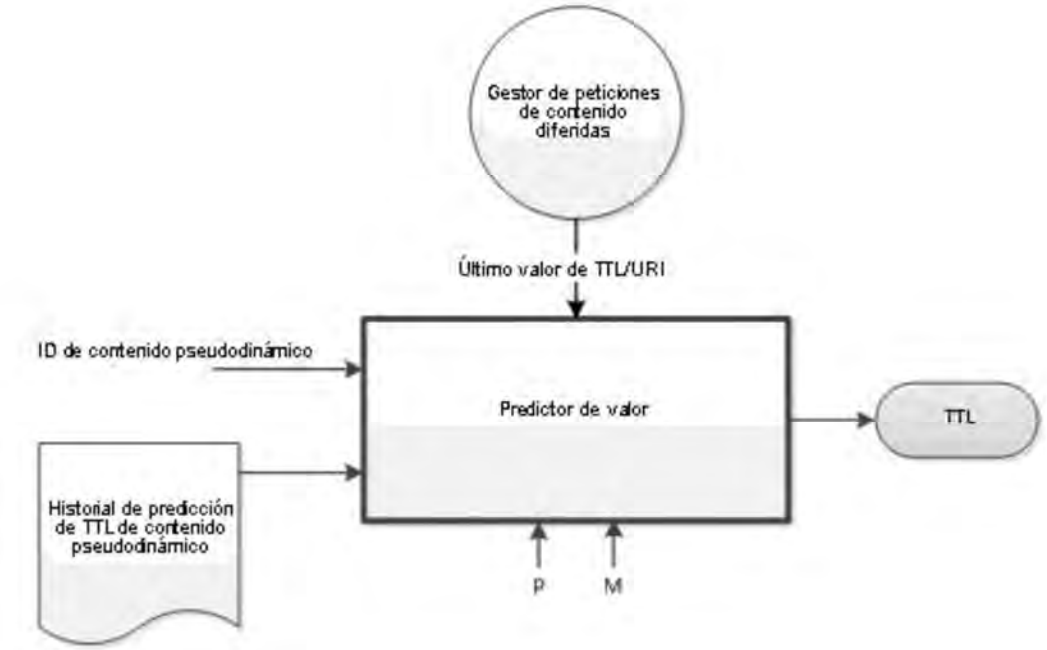


Figura 8

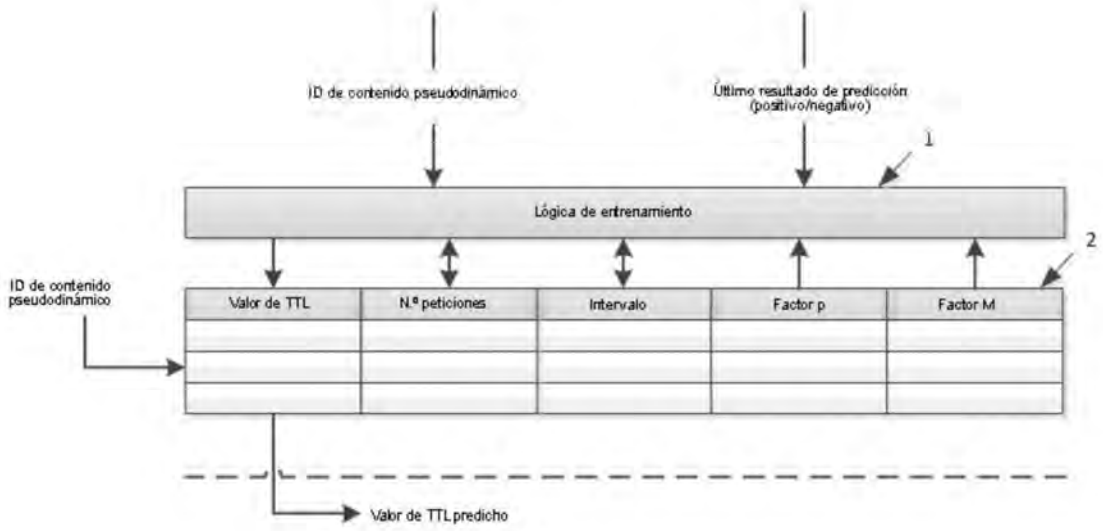


Figura 9

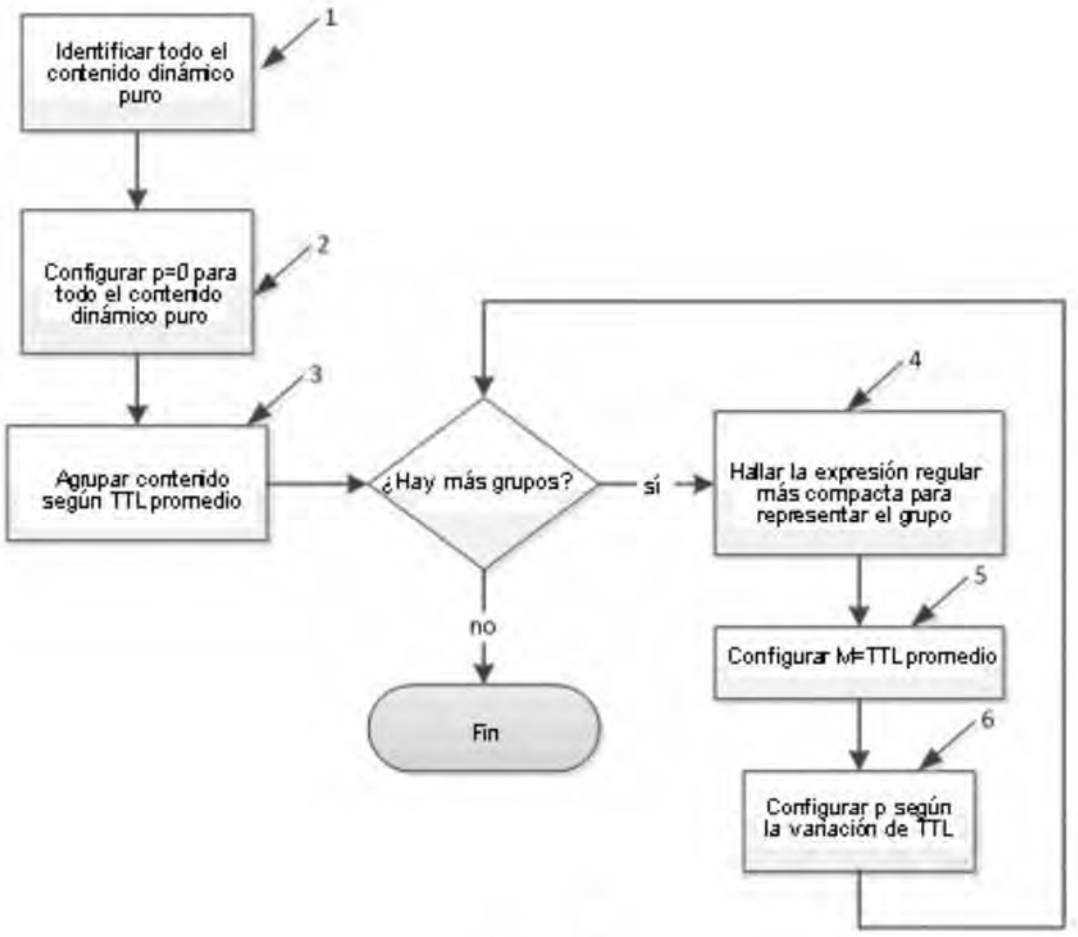
1
Identificar todo el contenido dinámico puro
2
Configurar p=0 para todo el contenido dinámico puro
3
Agrupar contenido según TTL promedio
¿Hay más grupos?
sí
no
Fin
4
Hallar la expresión regular más compacta para representar el grupo
5
Configurar M=TTL promedio
6
Configurar p según la variación de TTL

**Figura 10**