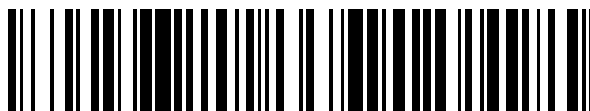


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 663 701**

51 Int. Cl.:

G06F 12/08 (2006.01)
G06F 13/10 (2006.01)
G06F 12/0862 (2006.01)
G06F 12/0804 (2006.01)
G06F 12/0866 (2006.01)
G06F 12/0871 (2006.01)
G06F 12/14 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **03.11.2009 PCT/US2009/063127**
 87 Fecha y número de publicación internacional: **20.05.2010 WO10056571**
 96 Fecha de presentación y número de la solicitud europea: **03.11.2009 E 09826570 (5)**
 97 Fecha y número de publicación de la concesión europea: **27.12.2017 EP 2353081**

54 Título: **Administración de datos y metadatos de la caché**

30 Prioridad:

14.11.2008 US 271400

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
16.04.2018

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**IYIGUN, MEHMET;
BAK, YEVGENIY;
FORTIN, MICHAEL;
FIELDS, DAVID;
ERGAN, CENK y
KIRSHENBAUM, ALEXANDER**

74 Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

ES 2 663 701 T3

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).

DESCRIPCIÓN

Administración de datos y metadatos de la caché

Campo de la invención

5 Esta invención se refiere a técnicas para emplear dispositivos de memoria no volátiles, tales como dispositivos de memoria de acceso aleatorio no volátil (NVRAM) extraíbles y no extraíbles.

Antecedentes

10 Algunos sistemas operativos convencionales brindan la capacidad de utilizar un dispositivo de memoria no volátil (es decir, un dispositivo periférico operable para proporcionar almacenamiento y/o memoria auxiliar a un ordenador, tal como una unidad USB de memoria flash) como un bloque o memoria caché de nivel de archivo para dispositivos de almacenamiento más lentos (por ejemplo, un medio de almacenamiento en disco, o uno o más dispositivos de almacenamiento accesibles a través de una red), para mejorar el rendimiento del sistema operativo y/o aplicaciones. A este respecto, porque las operaciones de lectura y escritura se pueden realizar significativamente más rápido desde o hacia un dispositivo de memoria no volátil (en adelante denominado "dispositivo de la caché" por simplicidad) que desde un dispositivo de almacenamiento más lento, utilizando un dispositivo de la caché para los datos de la caché almacenados en un dispositivo tan lento ofrecen oportunidades para mejorar significativamente la velocidad de las operaciones de entrada/salida (I/O) de los sistemas operativos y/o aplicaciones. Con este fin, el sistema operativo Microsoft Windows Vista, producido por Microsoft Corporation de Redmond, WA, incluye una función conocida como ReadyBoost que permite a los usuarios emplear dispositivos de la caché para almacenar datos de la caché que también residen en un dispositivo de almacenamiento más lento (denominado en adelante "disco de almacenamiento "o" disco" por conveniencia, aunque debe entenderse que estos términos se refieren generalmente a cualquier mecanismo(s) y/o dispositivo(s) de almacenamiento para los cuales I/O se realiza típicamente más lentamente que un dispositivo de la caché, incluyendo dispositivos de almacenamiento accesibles a través de una red).

15 El uso de un dispositivo de la caché para almacenar en la caché los datos almacenados en el disco puede lograrse utilizando un componente de administrador de la caché, que en algunas implementaciones es un controlador implementado en la pila de I/O del sistema operativo. Las Figs. 1A-1B representan ejemplos de procesos de alto nivel 10A-10B mediante los cuales un componente 100 administrador de la caché administra el almacenamiento en memoria caché de datos al dispositivo 110 de la caché. El dispositivo 110 de la caché se puede acoplar, utilizando una infraestructura de comunicaciones cableada y/o inalámbrica y protocolo(s), a un ordenador (no mostrado) en la que reside el administrador de la caché 100. Por ejemplo, el dispositivo 110 de la caché puede ser extraíble del ordenador (por ejemplo, comprende una unidad USB de memoria flash), no extraíble y/o accesible para el ordenador a través de una o más redes cableadas y/o inalámbricas.

20 Al comienzo del proceso 10A (figura 1A), el administrador de la caché 100 recibe una solicitud por escrito que especifica que los datos deben escribirse en la dirección X en el almacenamiento en disco (es decir, el volumen en la caché 120). El administrador de caché 100 procesa la solicitud haciendo que los datos se escriban en la dirección X en el volumen en la caché 120 en la operación 101, y también a la dirección Y en el dispositivo de la caché 110 en la operación 102. El proceso 10B (figura 1B) incluye operaciones realizadas a continuación cuando se recibe una solicitud de lectura que especifica que los datos almacenados en la dirección X en el volumen 120 almacenado en la caché deben leerse. El administrador de la caché 100 determina que los datos se almacenan en la caché en el dispositivo de la caché 110 en la dirección Y, y hace que los datos en la dirección Y se lean en la operación 103. Los datos se sirven desde el dispositivo de la caché para satisfacer la solicitud de lectura en la operación 104.

25 El administrador de la caché mantiene un mapeo de direcciones de disco (por ejemplo, dirección X) a direcciones de la caché correspondientes (por ejemplo, dirección Y) en metadatos, y estos "metadatos de la caché" se emplean generalmente para leer o escribir en el dispositivo de la caché. Por lo general, los metadatos de la caché se mantienen en la memoria y el administrador de la caché accede a ellos cuando se reciben las solicitudes de I/O. Como tal, cuando el administrador de la caché recibe una solicitud de lectura dirigida al desplazamiento X de disco, el administrador de la caché usa los metadatos de la caché para determinar que los datos también se almacenan en el desplazamiento Y de la caché y para satisfacer la solicitud haciendo que los datos sean leídos del desplazamiento Y de la caché en lugar del desplazamiento X del disco. Cuando el administrador de la caché recibe una solicitud escrita dirigida al desplazamiento X de disco, el administrador de la caché emplea los metadatos de la caché para determinar si también se almacenan los datos en esa dirección de disco en la caché. Si es así (por ejemplo, si los datos están almacenados en la dirección Y de caché), el administrador de la caché puede hacer que los datos se escriban en la dirección apropiada en la caché, o desalojar el contenido de la caché en esa dirección. De lo contrario, el administrador de la caché puede hacer que los datos se escriban en la caché, y puede actualizar los metadatos de la caché para que las lecturas futuras en el desplazamiento X del disco puedan ser servidos a partir de los datos almacenados en la caché.

Los sistemas operativos convencionales son capaces de soportar dispositivos de la caché con capacidad de almacenamiento relativamente limitada. Por ejemplo, la característica ReadyBoost del sistema operativo Windows Vista admite dispositivos de la caché con una capacidad de almacenamiento de hasta cuatro gigabytes. (En el momento en que se lanzó Windows Vista, la capacidad máxima de almacenamiento de los dispositivos de la caché era aproximadamente de dos gigabytes). La capacidad de almacenamiento de los dispositivos de la caché ha crecido rápidamente en los últimos años, con algunos dispositivos de la caché que proporcionan una capacidad de almacenamiento de hasta dieciséis gigabytes, que pueden almacenar el equivalente a treinta y dos gigabytes de datos cuando se comprimen.

En el documento US 2007/0061511 A1, se divulgan los medios para almacenar metadatos de la caché en un dispositivo de la caché de almacenamiento no volátil. Los metadatos se almacenan en forma comprimida y sin comprimir.

“OHN ET AL: “Path conscious caching of B+ tree indexes in a shared disks cluster”, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, ELSEVIER, AMSTERDAM, NL, vol. 67, no. 3, 27 enero 2007 (2007-01-27), páginas 266-301, XP005737456)” revela árboles B + que almacenan tanto datos como datos de índices en la memoria.

En el documento US 6 128 627, se revela una técnica para recolección de basura. De acuerdo con esta publicación, las marcas de agua indican cuándo realizar la recolección de basura para borrar la memoria

Resumen de la invención

La presente invención se define en las reivindicaciones independientes anexas a las cuales se debe hacer referencia. Los solicitantes han apreciado que los dispositivos de la caché que tienen una capacidad de almacenamiento relativamente mayor ofrecen una oportunidad significativa para mejorar la velocidad de las operaciones de I/O realizadas por los sistemas operativos y las aplicaciones. Los solicitantes también han apreciado que una razón por la que los sistemas operativos convencionales admiten dispositivos de la caché con capacidad de almacenamiento relativamente limitada es que los contenidos de la caché deben repoblarse cuando ciertos tipos de transiciones de energía (por ejemplo, modo de espera, hibernación (o modos equivalentes utilizados por sistemas operativos que no son de Microsoft Windows), o reinicio). Con los dispositivos de la caché que tienen una capacidad de almacenamiento relativamente mayor, volver a llenar los contenidos de la caché puede llevar un tiempo considerable y consumir recursos de procesamiento significativos. Como ejemplo, un dispositivo de memoria flash de ocho gigabytes, que puede contener hasta dieciséis gigabytes de datos comprimidos, puede demorar hasta treinta minutos en repoblarse usando I/O de fondo del disco a aproximadamente diez megabytes por segundo. Esto no solo niega de manera efectiva los beneficios de rendimiento que se podrían haber obtenido empleando el dispositivo de la caché, sino que puede retrasar significativamente las operaciones del sistema.

Una razón por la que los contenidos de la caché deben repoblarse en determinadas transiciones de energía es que no hay manera de garantizar confiablemente que los contenidos de la caché representen con precisión los contenidos de almacenamiento en disco cuando el ordenador se reinicie, porque los contenidos del dispositivo de la caché y/o el disco pueden haber sido modificados durante la transición de energía. Por ejemplo, cuando se cierra un primer ordenador, un pirata informático podría desconectar un dispositivo de la caché extraíble, conectarlo a otro ordenador y modificar los contenidos de la caché, de modo que si el dispositivo se reconectara al primer ordenador, datos incorrectos (en lo sucesivo denominados como datos “no auténticos”) podría ser servidos desde el dispositivo de la caché para satisfacer las solicitudes de I/O. Además de los actos maliciosos de un pirata informático, los contenidos de la caché también pueden contaminarse durante una transición de energía debido a una falla de hardware del ordenador o dispositivo de la caché.

Los contenidos de la caché también pueden volverse “obsoletos” durante una transición de energía porque los datos del disco se actualizaron durante la transición, de modo que cuando se reinicia el equipo, es posible que los contenidos de la caché ya no representen con precisión los contenidos del disco. Por ejemplo, después de apagado, un usuario puede iniciar el disco en otro sistema operativo que no reconoce el dispositivo de la caché y modificar los datos almacenados en el disco que se almacena en la caché en el dispositivo de la caché, de modo que cuando se reinicia el ordenador los contenidos de la caché ya no reflejan lo que está almacenado en el disco. En otro ejemplo, ciertas operaciones en un ordenador durante el apagado podrían ocurrir después de que el dispositivo de la caché se vuelva inaccesible para el sistema operativo (por ejemplo, después de que el dispositivo de la caché esté apagado), de modo que cualquier escritura en el disco realizada por el sistema operativo después de este momento puede no ser reflejado con precisión por los contenidos de la caché. Cualquiera de los numerosos eventos puede provocar que los contenidos de la caché queden obsoletos en una transición de energía. Ejemplos de la presente invención proporcionan técnicas para administrar estas y otras inquietudes de tal manera que los contenidos de la caché puedan ser considerados como datos almacenados que se reflejan fielmente en el disco durante una transición de energía. Por ejemplo, algunos ejemplos de la invención proporcionan técnicas para verificar que los contenidos de la caché permanezcan auténticos a lo largo de una transición de energía. Además, algunos ejemplos proporcionan técnicas para garantizar de manera confiable que los contenidos de la caché no se vuelvan obsoletos en una transición de energía. Además, algunas realizaciones proporcionan técnicas para administrar metadatos de la caché a través de transiciones de energía, así

como durante operaciones normales ("estado estable"), asegurando que los metadatos de la caché pueden ser accedidos eficientemente y salvados y restaurados de forma fiable cuando se produce una transición de energía

5 Proporcionando técnicas que aseguran que se puede confiar en el contenido de la caché como datos que reflejan fielmente datos almacenados en el disco a través de transiciones de energía, algunas realizaciones de la invención pueden habilitar un dispositivo de la caché con capacidad de almacenamiento sustancial para acelerar significativamente las operaciones de I/O realizadas por el sistema operativo y/o aplicaciones. La mayor velocidad de las operaciones de I/O no solo acelerará las operaciones normales de "estado estable" del ordenador, sino que también acelerará significativamente las operaciones realizadas durante el arranque, de modo que el ordenador estará listo para usarse mucho más rápidamente.

10 En algunas realizaciones de la invención, se proporciona un método para operar un ordenador que comprende una memoria y que tiene acoplado un medio de almacenamiento y un dispositivo de la caché, almacenando el medio de almacenamiento una pluralidad de elementos de datos cada uno en direcciones respectivas, cada uno de la pluralidad de elementos de datos también se almacenan en una dirección correspondiente en el dispositivo de la caché, metadatos de la caché accesibles al ordenador que proporciona un mapeo entre la dirección en el medio de almacenamiento y la dirección correspondiente en el dispositivo de la caché en el que se almacena cada elemento de datos. El método comprende acciones de: (A) almacenar los metadatos de la caché en una estructura de datos jerárquica que comprende una pluralidad de niveles de jerarquía; y (B) cargando solo un subconjunto de la pluralidad de niveles de jerarquía a la memoria.

20 Otras realizaciones proporcionan al menos un medio de almacenamiento legible por ordenador que tiene instrucciones codificadas que, cuando se ejecuta mediante un ordenador que comprende una memoria y que tiene acoplado a ella almacenamiento en disco y un dispositivo de la caché, almacenando el disco una pluralidad de elementos de datos cada uno en direcciones respectivas, cada uno de la pluralidad de elementos de datos también se almacena en una dirección correspondiente en el dispositivo de la caché, metadatos de la caché accesibles al ordenador que proporcionan un mapeo entre la dirección en el almacenamiento en disco y la dirección correspondiente en el dispositivo de la caché en el que se almacena cada elemento de datos, realizar un método que comprende acciones de: (A) almacenar los metadatos de la caché, en el dispositivo de la caché, en una estructura de datos jerárquica que comprende una pluralidad de niveles de jerarquía; (B) iniciar un reinicio del ordenador; (C) al iniciar el reinicio del ordenador, cargar solo un subconjunto de la pluralidad de niveles de jerarquía a la memoria; (D) recibir una solicitud para leer un elemento de datos almacenado en una dirección en el medio de almacenamiento; (E) acceder a una primera parte de los metadatos de la caché para identificar una dirección correspondiente en la que el elemento de datos se almacena en el dispositivo de la caché; y (F) almacenar la primera parte de los metadatos de la caché en la memoria.

35 Otras realizaciones proporcionan un sistema informático que comprende: una memoria; un medio de almacenamiento que almacena una pluralidad de elementos de datos en direcciones respectivas; un dispositivo de la caché que también almacena la pluralidad de elementos de datos en direcciones correspondientes y metadatos de la caché que proporcionan un mapeo entre la dirección en el medio de almacenamiento y la dirección correspondiente en el dispositivo de la caché en el que se almacena cada elemento de datos, los metadatos de la caché se almacenan en una estructura de datos jerárquica que comprende una pluralidad de niveles de jerarquía; al menos un procesador programado para: al iniciar un reinicio del ordenador, cargar solo un subconjunto de la pluralidad de niveles de jerarquía a la memoria; procesar solicitudes para leer elementos de datos almacenados en direcciones respectivas en el medio de almacenamiento utilizando los metadatos de la caché para identificar las direcciones correspondientes en las cuales los elementos de datos se almacenan en el dispositivo de la caché y almacenando las direcciones correspondientes identificadas en la memoria; y procesar un comando para apagar el ordenador transfiriendo el subconjunto de la pluralidad de niveles de jerarquía y las direcciones correspondientes identificadas de la memoria al dispositivo de la caché.

Breve descripción de los dibujos

Las Figs. 1A-1B son un diagrama de bloques que representa técnicas para escribir y leer desde un dispositivo de la caché, de acuerdo con la técnica anterior;

50 Las Figs. 2A-2B representan técnicas de ejemplo para escribir y leer desde un dispositivo de la caché de acuerdo con algunos ejemplos de la invención;

La Fig. 3 es un diagrama de bloques que representa una técnica de ejemplo para asegurar que los datos de la caché reflejan con precisión los datos almacenados en el disco después de una transición de energía, de acuerdo con algunos ejemplos de la invención;

55 La Fig.4 es un diagrama de bloques que representa una técnica de ejemplo para almacenar metadatos de la caché, de acuerdo con algunas realizaciones de la invención;

La Fig. 5 es un diagrama de bloques que representa ejemplos de operaciones de almacenamiento para metadatos de la caché, de acuerdo con algunas realizaciones de la invención;

La Fig. 6 es un diagrama de flujo que representa una técnica de ejemplo para dar servicio a las solicitudes de lectura que usan un dispositivo de la caché, de acuerdo con algunos ejemplos de la invención;

5 La Fig. 7 es un diagrama de bloque que representa un ordenador de ejemplo que puede usarse para implementar aspectos de la invención; y

La Fig. 8 es un diagrama de bloque que representa una memoria de ordenador de ejemplo en la que pueden grabarse instrucciones que implementan aspectos de la invención.

Descripción detallada

10 Algunos ejemplos de la invención proporcionan técnicas para asegurar que el contenido de la caché refleje con precisión los contenidos del almacenamiento en el disco a través de una transición de energía. Por ejemplo, algunos ejemplos proporcionan una capacidad para asegurar que los contenidos de la caché permanecen auténticos y/o no se han vuelto obsoletos en la transición de energía. Además, algunos ejemplos proporcionan técnicas para administrar metadatos de la caché, para asegurar que los metadatos no se hayan alterado durante una transición de energía.

15 Además, algunas realizaciones proporcionan una capacidad para almacenar metadatos de la caché que pueden mejorar la eficacia con la que se pueden realizar tanto las transiciones de energía como las operaciones normales. Las secciones que siguen describen estos ejemplos y realizaciones en detalle.

1. Verificación de la autenticidad de los datos de la caché

20 Cuando un ordenador experimenta una transición de energía (por ejemplo, se lleva al modo de espera o de hibernación, o se reinicia), un dispositivo de la caché puede desconectarse del ordenador y su contenido puede ser alterado (por ejemplo, por un pirata informático malintencionado). Por ejemplo, cuando un ordenador pasa al modo de espera o de hibernación, un dispositivo de la caché extraíble, como una unidad de memoria flash, puede desconectarse del ordenador y modificarse su contenido. Incluso los dispositivos no extraíbles, como los dispositivos NVRAM internos, pueden desconectarse y cambiar su contenido cuando se reinicia el sistema operativo (es decir, se vuelve a cargar, reiniciando así el ordenador). Como resultado, cuando el dispositivo de la caché se vuelve a conectar al ordenador, puede almacenar información diferente a la que el usuario cree (es decir, los datos almacenados en la caché pueden no ser "auténticos"). Si se sirven datos no auténticos de la caché para satisfacer una solicitud de entrada/salida (I/O), la operación del ordenador podría verse negativamente afectada.

30 Algunos ejemplos de la invención proporcionan técnicas para detectar modificaciones que ocurren en los contenidos de la caché durante una transición de energía, para asegurar que las solicitudes de I/O no se satisfagan utilizando datos no auténticos de la caché. En algunos ejemplos, se proporciona una capacidad para detectar cualquier "modificación fuera de línea" que se produzca en los contenidos de la caché durante una transición de energía que los haga no auténticos.

35 Algunos ejemplos proporcionan esta capacidad usando una representación de al menos una parte de los datos. Por ejemplo, una representación puede calcularse o derivarse de al menos una parte de los datos y/u otra información de una manera predeterminada. La representación se puede generar por primera vez cuando los datos se escriben en la caché y se almacenan en una o más ubicaciones. Por ejemplo, la representación se puede escribir en la caché junto con los datos, o en alguna otra ubicación. Cuando se almacena en la caché, la representación se puede almacenar de una manera que asocie la representación con los datos (por ejemplo, se puede escribir en una dirección de la caché adyacente a la que se escriben los datos, escritos en los metadatos de la caché asociados con los datos, y/o asociado de alguna otra manera). Cuando los datos se leen posteriormente desde la caché, la representación también puede recuperarse. La representación se puede volver a generar de la manera predeterminada, y la representación regenerada se puede comparar con la representación recuperada. Si las representaciones coinciden, los datos recuperados de la caché se determinan como auténticos y sirven para satisfacer la solicitud de lectura. De lo contrario, se emite una solicitud para leer los datos en lugar del almacenamiento en disco para satisfacer la solicitud de lectura, y los datos no auténticos almacenados en la caché pueden ser desalojados (por ejemplo, eliminados).

40 Las Figs. 2A-2B representan este proceso en mayor detalle. En particular, el proceso 20A que se muestra en la FIG. 2A incluye las operaciones realizadas cuando los datos se escriben en la caché, y el proceso 20B que se muestra en la FIG. 2B incluye operaciones realizadas posteriormente cuando los datos se leen desde la caché. Los procesos 20A-20B incluyen cada uno operaciones realizadas por el volumen en la caché 120 (en el lado izquierdo de cada figura), el administrador de la caché 100 (en el medio) y el dispositivo de la caché 110 (a la derecha).

45 Al comienzo del proceso 20A (figura 2A), el administrador de la caché 100 recibe una solicitud escrita dirigida a la dirección X en el almacenamiento del disco en la acción 205. En la acción 210, el administrador de la caché 100

emplea metadatos de la caché (no mostrados) para determinar una dirección Y en el dispositivo de la caché 110 a la que también deben escribirse los datos. El administrador de la caché también genera la representación de al menos una parte de los datos.

Los ejemplos de la invención pueden generar esta representación usando cualquier técnica adecuada. En algunos ejemplos, se pueden emplear una o más técnicas de autenticación criptográfica para generar la representación. Por ejemplo, en algunos ejemplos, la representación puede comprender un código de autenticación de mensaje (MAC) generado a partir de los datos y un conjunto de claves secretas y números de secuencia de elementos por datos. Sin embargo, la invención no se limita a dicha implementación, ya que puede emplearse cualquier técnica adecuada para generar la representación. Por ejemplo, las técnicas de autenticación criptográfica no necesitan ser empleadas. Como un ejemplo, un signo fuerte y/o código de redundancia cíclico (CRC) podría usarse alternativamente para representar datos, y puede generarse a partir de elementos de datos individuales almacenados en la caché, o para uno o más grupos de elementos de datos. A este respecto, los solicitantes han apreciado que si el objetivo de verificar la autenticidad de los datos fuera simplemente detectar instancias de corrupción de hardware (es decir, el pirateo de datos no era una preocupación), el uso de un CRC puede ser suficiente y puede consumir menos recursos de procesamiento que generando un MAC para cada elemento de datos. Sin embargo, si el objetivo es evitar que un pirata informático o un actor malicioso modifique los contenidos de la caché, entonces es preferible una solución criptográfica, de modo que se pueda utilizar una representación como MAC. La invención no se limita a ninguna implementación particular, ya que se puede emplear cualquier técnica adecuada.

En la acción 215, el administrador de la caché 100 emite la solicitud al dispositivo de la caché 110 para escribir los datos en la dirección Y de la caché. El administrador de la caché también emite una solicitud al dispositivo de la caché 110 para escribir la representación. Por ejemplo, el administrador de la caché 100 puede especificar que la representación debe escribirse en una o más ubicaciones adyacentes a la dirección Y de la caché, o almacenarse en metadatos de la caché para los datos, y/o utilizando cualquier otra técnica. Como se discutió anteriormente, la invención no se limita a escribir la representación en cualquier ubicación particular (por ejemplo, no necesita escribirse en el dispositivo de la caché 110). Si se escribe en la caché, la representación se puede asociar con los datos de la manera que se desee.

En la acción 220, el dispositivo de la caché 110 recibe la solicitud y la procesa escribiendo los datos y la representación en la acción 225.

En la acción 230, el administrador de la caché 100 emite una solicitud correspondiente al volumen almacenado en la caché 120 para escribir los datos en la dirección de disco X. El volumen de la caché 120 recibe esta solicitud en la acción 235 y la procesa escribiendo los datos en la dirección X en la acción 240. Aunque se muestra en la FIG. 2A como siendo efectuado posteriormente a la escritura en la caché en las acciones 215-225, las acciones 230-240 pueden realizarse en paralelo con las acciones 215-225, o en cualquier otro momento adecuado, ya que la invención no se limita a ninguna implementación particular. El proceso 20A entonces se completa.

El proceso 20B (figura 2B) incluye las operaciones realizadas para leer los datos almacenados en la caché. En la acción 245, el administrador 100 de la caché recibe una solicitud para leer los datos almacenados en la dirección X en el volumen 120 en la caché. Usando los metadatos de la caché (no mostrados), el administrador de la caché 100 determina que los datos se almacenan en la dirección Y en el dispositivo de la caché 110 en la acción 250. En la acción 255, el administrador de la caché 100 emite una solicitud de lectura para almacenar en la caché el dispositivo 110 para recuperar tanto los datos almacenados en la dirección Y como la representación asociada. La solicitud es recibida por el dispositivo de la caché 110 en la acción 260 y procesada en la acción 265, con lo que el dispositivo de la caché 100 devuelve los resultados al administrador de la caché 100.

En la acción 270, el administrador de la caché 100 determina si los datos recuperados de la caché pueden verificarse. En algunos ejemplos, esto se hace volviendo a generar la representación de los datos, y comparando la representación regenerada con la representación originalmente generada en la acción 215. Por ejemplo, la acción 270 puede incluir regenerar un MAC o CRC para los datos, y compararlo con la representación recuperada de la caché en la acción 265.

Si en la acción 270 se determina que la representación puede verificarse, el proceso pasa a la acción 275, en donde los datos recuperados del dispositivo de la caché 110 se sirven para satisfacer la solicitud de lectura, y luego se completa el proceso 20B. Si en la acción 270 se determina que la representación no puede verificarse, el proceso pasa a la acción 280, en donde el administrador de la caché 100 emite una solicitud para almacenar en la caché el dispositivo 110 para desalojar (por ejemplo, borrar o hacer inaccesible) los datos almacenados en la dirección Y. El administrador de la caché 100 luego emite una solicitud al volumen almacenado en la caché 120 para leer los datos de la dirección X en el disco en la acción 285. Esta solicitud se recibe en la acción 290 y se procesa en la acción 295, tras lo cual los datos se devuelven al administrador de la caché 100. Los datos leídos desde la dirección X se sirven para satisfacer la solicitud de lectura en la acción 299. El proceso 20B luego se completa.

Si se emplea una solución criptográfica para verificar la autenticidad de los datos, cualquier clave(s) utilizada(s) para generar una representación se puede escribir en ubicaciones distintas del dispositivo de la caché durante la transición de energía, para evitar que un pirata informático acceda a las claves para regenerar representaciones para elementos de datos alterados. Por ejemplo, en algunos ejemplos, las claves se pueden almacenar en el almacenamiento en disco (por ejemplo, cuando se apaga el ordenador) para evitar el acceso no autorizado. Sin embargo, la invención no está limitada a dicha implementación, ya que las claves no necesitan almacenarse, y si están almacenadas, pueden residir en cualquier ubicación adecuada. Por ejemplo, las claves almacenadas pueden colocarse en cualquier almacén de configuración proporcionada por el sistema operativo que está disponible durante el arranque del sistema (por ejemplo, el registro del sistema en Windows) o regenerarse basado en la entrada del usuario (por ejemplo, una contraseña) de tal manera que no es necesario el almacenamiento de claves.

Debería apreciarse que las realizaciones descritas anteriormente para verificar la autenticidad de un elemento de datos almacenado en un dispositivo de la caché son simplemente ejemplos, y esa autenticidad puede verificarse usando cualquier técnica adecuada. Por ejemplo, la autenticidad del elemento de datos no necesita verificarse generando una representación de al menos una parte del elemento de datos cuando se escribe el elemento de datos que luego se regenera cuando se lee el elemento de datos. Cualquier técnica adecuada que garantice de forma fiable que un elemento de datos leído desde la caché sea auténtico y coincida con el elemento de datos previamente escrito en la caché puede ser empleado. La invención no está limitada a ninguna implementación particular.

II. Prevenir el estancamiento de los datos de la caché

Como se discutió anteriormente, los sistemas operativos convencionales son incapaces de detectar cuándo se realizan las operaciones de escritura en los elementos de datos almacenados en el disco durante las transiciones de energía que hacen que el contenido de la caché se quede obsoleto. Algunas realizaciones de la invención proporcionan mecanismos para detectar cuándo se producen estas "escrituras fuera de línea", asegurando así que el contenido de la memoria caché refleje con precisión los datos almacenados en el disco después de que se produce una transición de energía.

Con algunos sistemas operativos (por ejemplo, la familia de sistemas operativos Windows ofrecidos por Microsoft Corporation), la semántica de ciertas transiciones de energía (por ejemplo, modos de espera e hibernación) es tal que los datos en dispositivos de almacenamiento no extraíbles (por ejemplo, almacenamiento en disco) pueden no ser modificado durante una transición de energía. Como tal, los contenidos de la caché que corresponden a los datos en dichos medios no extraíbles generalmente no se vuelven obsoletos. Sin embargo, cuando el ordenador se apaga, pueden suceder varias cosas que hacen posible que se modifiquen los datos en el disco. Por ejemplo, un usuario puede iniciar el disco en otro sistema operativo en ese ordenador, o conectar el disco a otro ordenador, y modificar los datos almacenados en el disco. Además, como se discutió anteriormente, la mecánica de apagado de muchos sistemas operativos convencionales es tal que, en algún momento durante el apagado, un dispositivo de la caché se apaga y el sistema operativo ya no es accesible, pero el sistema operativo puede continuar accediendo al disco. Como tal, el sistema operativo puede actualizar los elementos de datos en el disco que están en la caché en el dispositivo de la caché. Debido a que el dispositivo de la caché se ha apagado, el sistema operativo no tiene forma de actualizar tampoco estos contenidos de la caché, por lo que se vuelven obsoletos.

Para administrar estas y otras ocurrencias, algunos ejemplos de la invención proporcionan técnicas para detectar modificaciones a los datos almacenados en el disco después de que se inicia un apagado, de modo que los contenidos de caché que quedan obsoletos por tales modificaciones pueden actualizarse, expulsarse de la caché o manejarse de otra forma.

Para detectar escrituras que se realizan en el almacenamiento en disco durante las operaciones de apagado que se producen después de que se apaga un dispositivo de la caché, algunas realizaciones de la invención emplean un componente grabador de escritura. Un componente grabador de escritura puede, por ejemplo, implementarse como un controlador en la ruta de I/O del sistema operativo, aunque la invención no está limitada a dicha implementación. Por ejemplo, un componente grabador de escritura puede estar basado en hardware. Como ejemplo, el hardware de almacenamiento en disco puede proporcionar una o más interfaces que brinden la capacidad de identificar el conjunto de modificaciones que se produjeron durante un determinado período de tiempo, o si las modificaciones se produjeron durante un determinado período de tiempo. Por ejemplo, el hardware de almacenamiento en disco puede proporcionar un contador de giro/encendido/arranque que puede emplearse para deducir que al menos algunos elementos de datos almacenados se han actualizado, en cuyo caso puede desalojarse el contenido del caché correspondiente a los datos almacenados en el disco (Esto no debería ocurrir con frecuencia, por lo que emplear el dispositivo de la caché aún debería ofrecer beneficios sustanciales). La invención no está limitada a ninguna implementación particular.

En algunos ejemplos, el componente de grabador de escritura está configurado para activarse cuando se inicia el apagado, y para realizar un seguimiento de todas las escrituras realizadas en el almacenamiento en disco hasta que se complete el apagado. Como resultado, cuando el ordenador se reinicia más tarde, estas escrituras se pueden aplicar al contenido de la caché. Por ejemplo, cuando el ordenador se reinicia y los volúmenes de disco vienen en línea, el administrador de la memoria caché puede iniciarse y puede comenzar a rastrear las escrituras en el disco. El

5 administrador de la caché puede consultar el componente grabador de escritura para determinar las escrituras fuera de línea que se produjeron después de apagar el dispositivo de la caché, fusionar estas escrituras con aquellas que el administrador de la caché rastreó durante el inicio y aplicar el conjunto de escrituras al contenido de la caché. Aplicar escrituras al contenido de la caché puede incluir, por ejemplo, actualizar los contenidos de la caché correspondientes a los datos en el disco a los que se dirigieron las escrituras (por ejemplo, realizar las mismas operaciones de escritura a estos contenidos de la caché), desalojar estos contenidos de la caché, una combinación de los dos (por ejemplo, aplicar operaciones de escritura a ciertos contenidos de la caché y desalojar a otros), o realizar alguna(s) otra(s) operación(es). Después de que las escrituras fuera de línea se apliquen al contenido de la caché, es posible que el componente de grabadora de escritura se apague y que el dispositivo de la caché puede comenzar a servir las solicitudes de I/O.

10 La Figura 3 representa un proceso 30 de ejemplo para rastrear las escrituras fuera de línea y aplicar estas escrituras al contenido de la caché. En particular, el proceso 300 incluye operaciones realizadas por el administrador de caché 100, la grabadora de escritura 300, el dispositivo de la caché 110 y el volumen en caché 120 durante el apagado de un ordenador y el reinicio posterior.

15 En la acción 305, que ocurre durante el apagado del ordenador, el administrador de la caché 100 activa la grabadora de escritura 300 y le proporciona un "identificador de persistencia" que identifica el conjunto (es decir, la generación) de las operaciones de escritura que la grabadora de escritura debe rastrear. (Los ejemplos de los usos para un identificador de persistencia se describen en detalle a continuación). En la acción 310, el administrador de la caché 100 escribe el identificador de persistencia, así como los metadatos de caché almacenados en la memoria, para almacenar en el dispositivo de la caché 110. En este punto del proceso de apagado, el dispositivo de la caché 110 se apaga y se vuelve inaccesible para el administrador de la caché 100.

20 En la acción 315, la grabadora de escritura 300 escribe el identificador de persistencia que se le pasa en la acción 305 al volumen almacenado en caché 120, y comienza a rastrear cualquier operación de escritura realizada al volumen en caché 120 durante el apagado. Por ejemplo, la grabadora de escritura 300 puede crear un archivo de registro, o una o más estructuras de datos diferentes, en el volumen de la caché 120 o en alguna(s) otra(s) ubicación(es) para indicar las direcciones en el disco en el que se realizan las operaciones de escritura y/o los datos escrito en esas direcciones. Al finalizar la acción 315, las operaciones de apagado del ordenador han finalizado.

25 A partir de entonces, el ordenador se reinicia. Como parte del proceso de arranque, el volumen de la caché 120 se pone en línea, la grabadora de escritura 300 y el administrador de la caché 100 se reinician. El administrador de la caché 100 puede entonces comenzar a seguir las operaciones de escritura realizadas en el volumen almacenado en caché 120. Por ejemplo, el administrador de caché 100 puede crear un archivo de registro y almacenarlo en el dispositivo de caché 110, el volumen en caché 120, y/o la memoria del ordenador (no mostrada en la figura 3). En la acción 320, la grabadora de escritura 300 lee los cambios de volumen registrados en la acción 315, así como el identificador de persistencia escrito en el volumen en caché 120 en la acción 315. El cambio de volumen y el identificador de persistencia se pasan luego al administrador de la caché 100 en la acción 325.

30 Debería apreciarse que la grabadora de escritura 300 puede ser incapaz de rastrear todas las escrituras en el disco después de que el dispositivo de la caché 110 se haya apagado. Por ejemplo, la corrupción de datos de hardware, fallas de energía intempestivas y/o problemas al escribir el archivo de registro pueden hacer que la grabadora de escritura 300 no pueda rastrear todas las escrituras fuera de línea realizadas en un volumen de disco. En tales casos, la grabadora de escritura 300 puede indicar al administrador de la caché 100 en la acción 325 que no puede determinar de manera fiable que el registro sea un registro completo y preciso de todas las escrituras fuera de línea realizadas. Si esto ocurre, el administrador de la caché 100 puede desalojar todo el contenido de la caché, o una parte del mismo (por ejemplo, correspondiente a un volumen de disco particular para el cual la grabadora de escritura no puede rastrear todas las operaciones de escritura), siendo potencialmente no confiable. El resto de la descripción de la FIG. 3 asume que la grabadora de escritura 300 es capaz de rastrear todas las escrituras fuera de línea.

35 En la acción 330, el administrador de la caché 100 lee los metadatos de la caché y el identificador de persistencia del dispositivo de la caché 110 en la memoria. El administrador de la caché 100 determina si el identificador de persistencia puede verificarse (esto se describe más adelante). Si no, el administrador de la caché 100 puede desalojar el contenido completo del dispositivo de la caché 110, o una parte de este (por ejemplo, correspondiente a un volumen de disco particular para el cual no se podría verificar el identificador de persistencia). Si se puede verificar el identificador de persistencia, el administrador de la caché 100 combina cualquier operación de escritura realizada para almacenar en disco desde que se reinició el ordenador con cualquier operación de escritura rastreada por la grabadora de escritura 300. Por ejemplo, si uno o más registros indican los datos escritos en cada dirección en el disco, el administrador de la caché 100 puede seleccionar la última actualización realizada en cada dirección y escribirla en la memoria.

55 En algunos ejemplos, la grabadora de escritura 300 puede configurarse para continuar grabando escrituras después de reiniciar el ordenador, de modo que el administrador de la caché 100 no necesita grabar escrituras realizadas después de ese punto y fusionarlas con escrituras rastreadas por la grabadora de escritura 300. En cambio, la

grabadora de escritura 300 puede simplemente proporcionar un registro de todas las escrituras en el administrador de la caché 100.

5 Usando los metadatos de caché leídos en la acción 330, el administrador de la caché 100 aplica luego el conjunto de escrituras al contenido del dispositivo de la caché 110 en la acción 335. Como se describió anteriormente, la aplicación de las escrituras puede incluir desalojar los contenidos de la caché, hacer ambos, o realizar alguna otra operación. Por ejemplo, las escrituras fuera de línea rastreadas por la grabadora de escritura 300 en la acción 315 pueden aplicarse desalojando los contenidos de la caché correspondientes, mientras que las escrituras rastreadas por el administrador de la caché 100 desde que se reinició el ordenador pueden aplicarse actualizando los contenidos de la caché correspondientes para reflejar las escrituras. La aplicación de operaciones de escritura a los contenidos de la caché se puede realizar de cualquier manera adecuada, ya que la invención no se limita a ninguna implementación en particular.

Al finalizar la acción 335, el proceso de la FIG. 3 se completa.

15 Debería apreciarse que la invención no está limitada a emplear un componente de grabador de escritura que está configurado para activarse cuando se inicia el apagado, ya que las operaciones de escritura que no se producen durante el apagado también pueden ser, alternativamente, rastreadas. Por ejemplo, en algunas implementaciones, un dispositivo de la caché puede ser susceptible de volverse inaccesible por períodos de tiempo. Por ejemplo, si se accede al dispositivo de la caché a través de una o más redes, la conectividad podría perderse, o si el dispositivo de la caché es extraíble del ordenador, podría producirse una extracción inesperada (por ejemplo, involuntaria). Como resultado, algunas realizaciones pueden emplear una grabadora de escritura para rastrear todas (o una parte de) las escrituras realizadas en el disco, no solo las que ocurren durante el apagado, y un dispositivo de la caché que está configurado para capturar periódicamente “instantáneas” de caché mientras está en línea. Como tal, si la caché se vuelve inaccesible durante un período de tiempo y luego se vuelve a conectar, la última instantánea de la memoria caché se puede actualizar utilizando operaciones de escritura rastreadas por la grabadora de escritura, en lugar de tener que reconstruirse por completo.

25 También debería apreciarse que, si bien el proceso de ejemplo 30 de la FIG. 3 puede detectar las escrituras fuera de línea realizadas por el sistema operativo durante el apagado, es posible que se necesiten otras medidas para detectar escrituras fuera de línea realizadas en el disco después de que se complete el apagado. Dichas escrituras pueden ocurrir, por ejemplo, cuando un usuario inicia el disco en otro sistema operativo después del apagado, o retira el disco del ordenador luego del apagado y lo conecta a otro ordenador, y luego modifica los datos almacenados en el disco.

30 Reconociendo las dificultades asociadas con el intento de rastrear escrituras fuera de línea que se producen después del apagado (por ejemplo, por otro sistema operativo), algunos ejemplos de la invención en cambio intentan evitar que ocurra. Por ejemplo, algunos ejemplos intentan hacer que un volumen de disco particular sea inaccesible para sistemas operativos que no proporcionan un componente de grabación de escritura después del apagado. Esto se puede lograr de muchas maneras.

35 En algunos ejemplos, la grabadora de escritura 300 puede marcar un volumen de disco de tal manera que se vuelve inmontable por los sistemas operativos que no proporcionan un componente de grabadora de escritura para rastrear escrituras fuera de línea. Por ejemplo, la grabadora de escritura 300 puede modificar el identificador de volumen que indica el tipo de sistema de archivos utilizado en el volumen. A este respecto, los expertos en la técnica reconocerán que un identificador de volumen permite que un sistema operativo identifique el tipo de sistema de archivos utilizado para almacenar datos en el volumen, permitiendo así que el sistema operativo comprenda la estructura de los datos almacenados en el volumen, dónde encontrar archivos, etc. Por ejemplo, si un identificador de volumen indica que se utilizó un sistema de archivos NT File System (NTFS) para almacenar datos en el volumen, entonces otro sistema operativo que intente montar el volumen comprenderá que se necesitaría un sistema de archivos NTFS para analizar y acceder los datos al respecto. Si el identificador de volumen no proporciona ninguna indicación del tipo de sistema de archivos utilizado para almacenar datos en el volumen, la mayoría de los sistemas operativos no montarían el volumen, ya que no habría una manera confiable de comprender la estructura de los datos almacenados en el mismo.

Como tal, algunas realizaciones de la invención modifican el identificador de volumen de un volumen de disco para hacerlo inaccesible, evitando así que un usuario inicie el volumen de disco en otro sistema operativo y realice cambios fuera de línea a los datos almacenados en el volumen.

50 Reconociendo que algunos sistemas operativos pueden ser capaces de identificar el tipo de sistema de archivos utilizado para almacenar datos en un volumen incluso si el identificador de volumen se modificó, algunos ejemplos de la invención proporcionan un mecanismo para detectar cuándo un sistema operativo monta el volumen. A este respecto, para montar un volumen de disco, cualquier sistema operativo necesitaría actualizar el identificador de volumen (por ejemplo, para indicar que se empleó un sistema de archivos NTFS para almacenar datos en el volumen) para permitir el acceso a los datos. Cualquier actualización de este tipo sería fácilmente detectable al reiniciar. Si se detectara dicha actualización, algunas realizaciones de la invención pueden suponer que los contenidos del volumen

se han modificado desde el último apagado, y expulsar los contenidos de la caché correspondientes a los datos almacenados en el volumen.

5 Algunas realizaciones de la invención proporcionan una capacidad mediante la cual un volumen de disco puede arrancarse en otro sistema operativo que también emplea un componente de grabación de escritura. Por ejemplo, si se extrajo un disco de un ordenador que ejecuta un sistema operativo que proporciona un componente grabadora de escritura y lo inicia en otro sistema operativo que proporciona un componente de grabadora de escritura, el otro sistema operativo podría configurarse para reconocer que un identificador de volumen cambiado indica que el volumen puede almacenarse en la caché. Como resultado, el otro sistema operativo puede agregar a un registro de escrituras fuera de línea (por ejemplo, almacenadas en el volumen) creado por el primer sistema operativo.

10 Los ejemplos descritos anteriormente diseñados para hacer que un volumen de disco no pueda montarse en ciertos sistemas operativos pueden plantear problemas para ciertas aplicaciones que dependen del identificador de volumen para realizar ciertas funciones (por ejemplo, aplicaciones de respaldo). Con estas aplicaciones, si se cambió el identificador de volumen, el volumen puede ser irreconocible y, por lo tanto, no se realiza una copia de seguridad. En consecuencia, algunas realizaciones de la invención proporcionan un mecanismo para determinar si un sistema de archivos se montó después del apagado. De ser así, se asume que se realizaron cambios en los datos en el sistema de archivos, y que todos los contenidos de la memoria caché correspondientes a los datos en el sistema de archivos pueden ser desalojados.

20 Algunos ejemplos pueden detectar el montaje de un sistema de archivos después del apagado al colocar el registro del sistema de archivos en apagado en un estado que requeriría que cualquier sistema operativo que intente montar el sistema de archivos modifique el registro de alguna manera (por ejemplo, cambie su ubicación, agregue una nueva entrada, etc.). Por ejemplo, la grabadora de escritura 300 puede observar como parte de la tarea de registrar fuera de línea escribir la ubicación y/o el contenido del registro del sistema de archivos cuando se desmonta el sistema de archivos (por ejemplo, en el registro mismo). Ya que cualquier sistema operativo que intente montar el sistema de archivos debería cambiar el registro (por ejemplo, si el sistema de archivos fuera un sistema de archivos NTFS, un sistema operativo que intente montar el sistema de archivos agregaría una entrada al registro), si el registro no ha cambiado al reiniciarse, se supone que el sistema de archivos no fue montado por otro sistema operativo durante la transición de alimentación, por lo que los contenidos de la caché correspondientes a los datos almacenados en el sistema de archivos no se han quedado obsoletos. Por el contrario, si el registro ha sido modificado de alguna manera (por ejemplo, su ubicación ha cambiado, y se ha agregado la entrada, etc.) entonces se supone que el sistema de archivos fue montado por otro sistema operativo, y que los datos almacenados en él han cambiado, haciendo que el contenido de la caché correspondiente a los datos almacenados en el sistema de archivos quede obsoleto. Como tal, estos contenidos de caché pueden ser desalojados.

35 Además de proporcionar mecanismos para evitar escrituras fuera de línea, algunos ejemplos de la invención proporcionan una capacidad para gestionar generaciones incoherentes de contenido de la caché. Se pueden crear generaciones incoherentes de contenido de la caché por varias razones. Un ejemplo puede ocurrir cuando los ordenadores primero y segundo, que tienen dispositivos de caché primero y segundo conectados a los mismos, emplean las técnicas descritas en este documento para persistir en el contenido de la caché a través de las transiciones de energía. Si el segundo dispositivo de la caché se conectara al primer ordenador (o el primer dispositivo de la caché se conectará al segundo ordenador) y el primer ordenador fuera reiniciado, se podrían enviar datos incorrectos desde el segundo dispositivo de la caché para satisfacer las solicitudes de I/O. Esto se debe a que el sistema operativo del primer ordenador consideraba que el contenido del segundo dispositivo de la caché era auténtico (ya que una representación regenerada de los datos devueltos de la memoria caché podía coincidir con una representación generada originalmente) y no obsoleto (ya que las escrituras fuera de línea podían aplicarse al contenido de la caché). Otro ejemplo podría surgir si un primer dispositivo de la caché estuviera conectado a un ordenador, el ordenador se apagará (persistiendo así el contenido de la caché), el ordenador fuera entonces reiniciado, se conectará un segundo dispositivo de la caché y el ordenador se apagará nuevamente (persistiendo así contenido de la caché de nuevo). Si el ordenador se reiniciara de nuevo y se conectara el primer dispositivo de la caché, se podrían servir datos incorrectos para satisfacer las solicitudes de I/O, ya que no habría una manera confiable de determinar que el primer dispositivo de la caché no almacenara la última generación de los contenidos de la caché.

50 Algunos ejemplos proporcionan la capacidad de identificar generaciones incoherentes de contenido de la caché de manera que el contenido de la caché que persistieron antes del último apagado no se use erróneamente para satisfacer solicitudes de I/O. En algunos ejemplos, esta capacidad se proporciona a través de un identificador de persistencia único, que puede generarse (como un ejemplo) cuando se inicia el apagado, de cualquiera de las numerosas formas. Por ejemplo, los GUID y/o los generadores de números aleatorios criptográficos se pueden emplear para este propósito. Como se describió anteriormente con referencia a la FIG. 3, el identificador de persistencia puede almacenarse en el dispositivo de la caché (por ejemplo, en o con metadatos de la caché) así como en el ordenador (por ejemplo, en disco y/o memoria) y verificarse (por ejemplo, comparando las dos versiones) una vez el ordenador se inicie. Si la verificación no tiene éxito, los contenidos de la caché pueden ser desalojados como representación de una generación de la caché persistente anterior.

Al igual que con las claves de autenticación discutidas anteriormente, cualquier clave utilizada para generar un identificador de persistencia puede escribirse en una ubicación distinta del dispositivo de la caché durante la transición de energía. Por ejemplo, en algunos ejemplos, un componente de grabadora de escritura puede escribir las claves, así como el identificador de persistencia en el almacenamiento del disco (por ejemplo, al apagar). Sin embargo, la invención no se limita a dicha implementación, ya que los expertos en la materia pueden imaginar numerosas ubicaciones alternativas en las que se pueden guardar claves. Las claves pueden, por ejemplo, mantenerse en cualquier almacén de configuración proporcionado por el sistema operativo que está disponible durante el inicio del sistema (por ejemplo, el registro en Windows).

III. Metadatos de la caché

Como se describió anteriormente, los metadatos de caché pueden proporcionar un mapeo entre las direcciones de disco donde se almacenan los elementos de datos y las direcciones correspondientes en un dispositivo de la caché donde esos elementos de datos se almacenan en la caché. Algunas realizaciones de la invención proporcionan una capacidad para almacenar metadatos de la caché que reduce significativamente la cantidad de memoria requerida para almacenar los metadatos de la caché durante las operaciones de tiempo de ejecución del sistema.

Además, algunas realizaciones proporcionan técnicas que permiten confiar en metadatos de la caché en transiciones de energía o cualquier otro evento que desconecte el dispositivo de la caché (por ejemplo, eliminación de un dispositivo de la caché del ordenador, interrupción de la red que hace inaccesible un dispositivo de la caché de red, etc.), de modo que se pueda acceder de forma confiable al contenido de la caché cuando el ordenador se reinicie y/o el dispositivo de la caché se ponga en línea. A este respecto, debe apreciarse que con ciertos tipos de transiciones de energía (por ejemplo, modos de espera e hibernación), simplemente es aceptable almacenar metadatos de la caché en la memoria (es decir, RAM) ya que los contenidos de la memoria se conservan durante las transiciones en espera e hibernación. Durante el reinicio, sin embargo, los contenidos de la memoria del sistema no se conservan. Como tal, algunas realizaciones de la invención proporcionan el almacenamiento de metadatos de la caché en algún medio/medios no volátiles durante el apagado, y luego se restaura al reiniciar. Por ejemplo, los metadatos de la caché pueden almacenarse en un dispositivo de la caché y/o en uno o más medios no volátiles separados. Además, algunas realizaciones pueden ser capaces de derivar algunas partes de metadatos de la caché de otros, de modo que no se requiere almacenar todos los metadatos de la caché.

Algunas realizaciones pueden emplear las técnicas descritas en la Sección I anterior para verificar la autenticidad de los metadatos de la caché, para detectar y prevenir modificaciones inadvertidas o maliciosas a los metadatos cuando el dispositivo de la caché se desconecta (por ejemplo, durante el apagado del equipo, eliminación del dispositivo de la caché desde el ordenador, una interrupción de la red que hace que un dispositivo de la caché de red sea inaccesible, etc.). Por ejemplo, cuando el dispositivo de la caché se conecta, el administrador de la caché puede verificar la autenticidad de los metadatos a medida que se cargan en la memoria, usando las técnicas descritas anteriormente con referencia a las FIGS. 2A-2B. Si no se puede verificar la autenticidad de los metadatos de la caché, los contenidos de la caché correspondiente se pueden actualizar en función de los datos almacenados en el disco, desalojados o procesados de otro modo como se describe anteriormente.

En algunos ejemplos, los metadatos de la caché pueden comprimirse para reducir la cantidad de metadatos que se guardarán durante el apagado y la carga durante el reinicio. Debido a que la compresión de metadatos puede requerir guardar una pieza de información separada (por ejemplo, un encabezado en la caché) que contiene información sobre los metadatos, las técnicas descritas anteriormente pueden emplearse para verificar la autenticidad de esta información también al reiniciar.

Algunas realizaciones de la invención proporcionan técnicas para almacenar metadatos de la caché de una manera que reduce en gran medida la cantidad de metadatos de la caché almacenados en la memoria al mismo tiempo, reduciendo así la cantidad de tiempo requerida para cargar metadatos de la caché y descargarlos de la memoria (por ejemplo, durante el tiempo de ejecución y las operaciones de inicio/apagado) y reduce en gran medida la "huella" de memoria de los metadatos de la caché. A este respecto, debe apreciarse que con los dispositivos de la caché que tienen una capacidad de almacenamiento relativamente grande, puede ser necesaria una cantidad significativa de metadatos para administrar los contenidos de la caché. Por ejemplo, un dispositivo de la caché que tiene una capacidad de almacenamiento de dieciséis gigabytes puede almacenar hasta treinta y dos gigabytes de datos comprimidos. En algunas implementaciones, las direcciones de disco pueden reflejarse en metadatos de la caché en "unidades de datos" que representan cuatro kilobytes de almacenamiento en disco. Como tal, para rastrear la ubicación de treinta y dos gigabytes de datos, se necesitan ocho millones de unidades de datos distintas. Si cada una de las ocho millones de unidades de datos está representada en metadatos de la caché utilizando una mapeo de dieciséis bytes (es decir, desde una dirección de disco a una dirección de la caché), estos mapeos requieren ciento veintiocho megabytes de almacenamiento. Los solicitantes han apreciado que el almacenamiento de ciento veintiocho megabytes de metadatos de la caché en la memoria ocuparía una porción innecesariamente grande de la memoria en muchos ordenadores. Además, el tiempo requerido para escribir ciento veintiocho megabytes de metadatos de la caché de la memoria en medios no volátiles durante el cierre, y para restaurar ciento veintiocho megabytes de metadatos de la

caché de medios no volátiles en la memoria al reiniciar, produciría un consumo prohibitivo de tiempo y consumiría una cantidad excesiva de recursos de procesamiento.

5 Reconociendo que la cantidad de metadatos de la caché no puede reducirse fácilmente, algunas realizaciones de la invención proporcionan técnicas diseñadas para reducir los recursos de almacenamiento necesarios para almacenar metadatos de la caché, así como el tiempo y los recursos de procesamiento necesarios para guardar y restaurar metadatos de la caché al apagado y arranque.

10 En algunas realizaciones, esto se lleva a cabo almacenando metadatos de la caché en una o más estructuras de datos jerárquicas (por ejemplo, árboles, matrices de múltiples niveles, etc.). Emplear una estructura de datos jerárquica puede permitir que los niveles más bajos de la jerarquía se almacenen en un medio no volátil (por ejemplo, el dispositivo de la caché) mientras que solo los niveles superiores de la jerarquía se almacenan en la memoria. Por ejemplo, en algunas realizaciones, solo los niveles más altos de la jerarquía se almacenan en la memoria, de modo que la “huella” ocupada por la memoria en los metadatos de la caché puede reducirse en gran medida, incluso cuando se necesita una cantidad de metadatos de la caché necesarios para soportar los dispositivos de la caché que tengan una capacidad de almacenamiento significativa se almacenan totalmente. Por supuesto, almacenar solo niveles superiores de la jerarquía en los metadatos de la caché, ya que algunas formas de realización pueden almacenar cierta información mantenida en los niveles inferiores de la jerarquía en la memoria, para reducir la sobrecarga de I/O asociada a los accesos repetidos a esta información. La invención no está limitada a ser implementada de ninguna manera particular.

20 Durante el funcionamiento del sistema, a medida que se procesan las solicitudes de lectura, los metadatos de la caché que se leen desde el medio no volátil (es decir, desde niveles inferiores de la jerarquía) para realizar la operación de lectura pueden ser “contactados” en (es decir, leído desde un medio de almacenamiento en) a la memoria de tal manera que se pueda acceder más rápidamente a las solicitudes de lectura subsiguientes a la misma dirección de disco/caché. Cuando el ordenador se apaga más tarde y/o el dispositivo de la caché se desconecta, solo los metadatos de la caché almacenados en los niveles superiores de la jerarquía y los metadatos de la caché se almacenan en los niveles inferiores de la jerarquía que se contactaron en la memoria, puede necesitar ser guardado en el medio no volátil. Como tal, el tiempo requerido para mover los metadatos de la caché desde la memoria al almacenamiento no volátil al apagado, y restaurar los metadatos de la caché desde el almacenamiento no volátil a la memoria al reiniciar, puede reducirse significativamente.

30 Algunas realizaciones de la invención emplean un árbol B+ para almacenar al menos una porción de metadatos de la caché. Como apreciarán los expertos en la materia, los árboles B+ pueden emplear grandes factores de ramificación, y por lo tanto, reducir el número de niveles en la jerarquía empleada. Utilizando el ejemplo anterior, si se van a representar ocho millones de unidades de datos en metadatos de la caché y se empleó un árbol B+ con un factor de ramificación de doscientos (para que cada nodo en la jerarquía tenga doscientos nodos “secundarios”), una estructura de datos que tiene solo tres niveles sería suficiente para almacenar los metadatos: un solo nodo “raíz” en el nivel más alto, doscientos nodos en el segundo nivel y cuarenta mil nodos en el tercer nivel, con cada uno de los cuarenta mil nodos, incluidos los punteros a doscientas unidades de datos (u ocho millones de unidades de datos en total).

40 La Fig. 4 representa este árbol B+ de ejemplo que incluye el nodo raíz 400, los nodos de nivel dos 410₁₋₂₀₀ y los nodos de nivel tres 420₁₋₂₀₀. Cada nodo incluye doscientos elementos, cada uno separado por punteros a nodos en un nivel inferior en la jerarquía. Por ejemplo, el elemento 402 en el nodo raíz 400 está delimitado por los punteros 401 y 403. Un valor (por ejemplo, una dirección de caché) asociado con una clave determinada (por ejemplo, una dirección de disco) puede determinarse siguiendo el puntero a la izquierda o a la derecha de un elemento en un nodo, con el puntero a la izquierda del elemento que es seguido si la clave es menor que el elemento y se sigue el puntero a la derecha si la clave es mayor que el elemento. Por ejemplo, para determinar un valor para una clave que es menor que el elemento 402, se seguiría el puntero 401 al nodo de nivel dos 410₁, para determinar un valor para una clave mayor que el elemento 402 pero inferior al elemento 404, se seguiría el puntero 403 para nivelar dos nodos 410₂ (no mostrado), y así sucesivamente. De manera similar, en el nodo de nivel dos, se sigue un puntero a la izquierda o derecha de un elemento (dependiendo de si la clave es menor o mayor que los elementos en el nodo) a un nodo del nivel tres. En el nivel tres, se sigue un puntero final (nuevamente según si la clave es menor o mayor que los elementos en el nodo) al valor, con cada puntero en el nivel tres haciendo referencia a una de las ocho millones de unidades de datos en los metadatos de la caché.

55 Debe apreciarse que un árbol B+ con un gran factor de ramificación proporciona una jerarquía relativamente “plana” con casi todos los nodos localizados en el nivel inferior de la jerarquía. Es decir, de los 40,201 nodos totales en el árbol, 40,000 están en el nivel más bajo. Algunas realizaciones de la invención aprovechan esto restaurando solo los dos niveles superiores de la jerarquía a la memoria al inicio, mientras que los metadatos de la caché en el nivel más bajo de la jerarquía se almacenan en el dispositivo de la caché hasta que se necesiten (por ejemplo, se pueden cargar en la memoria a pedido, ya que las solicitudes de lectura se procesan, se cargan de forma perezosa, etc.). Debido a que solo una parte de la estructura de datos jerárquica se almacena en la memoria, los metadatos de la caché pueden ocupar una porción de memoria mucho más pequeña de lo que se requeriría si la totalidad o la mayor parte de los metadatos de la caché se mantuvieran en la memoria. Además, cuando el ordenador se apaga, solo los datos en los

dos niveles superiores y los datos cargados en la memoria durante la operación necesitan almacenarse en el dispositivo de la caché. Como resultado, tanto las operaciones de inicio como de apagado se pueden realizar de manera rápida y eficiente.

5 Por lo tanto, algunas realizaciones de la invención proporcionan punteros en nodos en un nivel de la jerarquía almacenada en la memoria (en el ejemplo anterior, el nivel dos de la jerarquía) que referencia nodos a otro nivel de la jerarquía almacenados en el dispositivo de la caché (en el ejemplo anterior, nivel tres). Por ejemplo, cuando se recibe una solicitud de lectura para un elemento de datos en la caché, las realizaciones de la invención siguen punteros a través de uno o más niveles de la jerarquía almacenados en la memoria y luego a metadatos en niveles inferiores de la jerarquía almacenados en la caché, para determinar la dirección en la que el elemento de datos se almacena en la caché. En algunas realizaciones, una vez que se determina la dirección de la caché para el elemento de datos, puede almacenarse en la memoria de tal manera que las solicitudes posteriores para leer el elemento se puedan realizar sin tener que leer los metadatos de la caché desde el dispositivo de la caché.

15 La Fig.5 representa un sistema de ejemplo 50 para administrar metadatos de la caché de acuerdo con algunas realizaciones de la invención. La figura 5 representa la memoria 500 y el dispositivo de la caché 110, ambos accesibles a un ordenador (no mostrados). Cuando se inicia el ordenador, los metadatos de la caché que comprenden uno o más niveles de una estructura de datos jerárquica tal como un árbol B+ se cargan en la memoria 500 en la operación 505. Utilizando el ejemplo anterior para ilustrar, si hay ocho millones de unidades de datos representadas en metadatos de la caché, de modo que se puede usar una estructura de datos jerárquica de tres niveles para almacenar los metadatos de la caché, los dos niveles superiores de la jerarquía se pueden cargar en la memoria 500. Por supuesto, si más o menos ocho millones de unidades de datos están representadas en metadatos, y se debe usar una estructura jerárquica de datos que tenga más o menos tres niveles, entonces se puede cargar un número diferente de niveles a la jerarquía en la memoria 500.

25 A continuación, cuando una solicitud de lectura se dirige a un elemento de datos mantenido en la caché, la dirección de la caché en la que se almacena el elemento de datos se determina accediendo a los metadatos de la caché almacenados en el o los niveles de la jerarquía almacenada en el dispositivo de la caché 110. Estos metadatos de la caché se almacenan entonces en la memoria 510, de modo que las lecturas o escrituras posteriores en el elemento de datos se pueden realizar sin tener que leer los metadatos de la caché almacenados en el dispositivo de la caché para determinar la dirección de la caché en la que se almacena el elemento de datos. En cambio, la dirección de la caché puede leerse desde la memoria, lo que se puede realizar más rápidamente que una lectura en la caché.

30 Más tarde, cuando el ordenador se apaga, los metadatos de la caché almacenados en la memoria (es decir, los metadatos almacenados en los niveles de la jerarquía cargados en la memoria en la operación 505 y cualquier metadato utilizado para satisfacer solicitudes de lectura escritos en la memoria en la operación 510) es cargado en el dispositivo de la caché 500 en la acción 515. Como resultado de la cantidad relativamente pequeña de metadatos de la caché almacenados en la memoria, el apagado puede realizarse rápidamente, sin requerir recursos de procesamiento sustanciales.

40 Debe apreciarse que un árbol B+ es solo uno de los numerosos tipos de estructuras de datos que pueden emplearse para almacenar metadatos de la caché, y que otros tipos de estructuras de datos (por ejemplo, estructuras jerárquicas como árboles AVL, árboles rojo-negro, árboles de búsqueda binarios, árboles B y/u otras estructuras de datos jerárquicas y no jerárquicas) pueden ser empleados. La invención no se limita a emplear cualquier estructura de datos o combinación de estructuras de datos para almacenar metadatos de la caché.

45 Algunas realizaciones pueden proporcionar una "cantidad objetivo" de metadatos de la caché que se mantendrán en la memoria en cualquier momento. La cantidad objetivo puede determinarse de cualquier manera adecuada. Por ejemplo, una cantidad objetivo puede ser un porcentaje de la cantidad de memoria física disponible para un ordenador. Por ejemplo, si el ordenador tiene un gigabyte de memoria, entonces dos megabytes de metadatos de la caché (como un ejemplo) pueden almacenarse en la memoria en cualquier momento. Por lo tanto, cuando el ordenador se apaga, solo se necesitan cargar dos megabytes de metadatos de la caché en el dispositivo de la caché.

50 En algunas realizaciones, los metadatos de la caché se pueden ciclar dentro y fuera de la memoria. Por ejemplo, si una cantidad objetivo de metadatos de la caché ya está almacenada en la memoria, y se realiza una lectura que requiere que los metadatos de la caché se lean desde el dispositivo de la caché, esos metadatos pueden ser "contactados" en la memoria y otros metadatos de la caché (por ejemplo, el que fue accedidos menos recientemente) pueden ser borrado. Por ejemplo, los metadatos de la caché pueden borrarse después de escribirse en el dispositivo de la caché. Alternativamente, el sistema puede determinar si los metadatos de la caché han cambiado desde la última vez que se escribieron, y si no, simplemente se puede borrar, eliminando así el tiempo y los recursos de procesamiento requeridos para escribir los metadatos de la caché. Usando las técnicas descritas anteriormente, se puede mantener la pequeña "huella" ocupada por los metadatos de la caché en la memoria.

La Fig. 6 representa un ejemplo. Específicamente, el proceso 60 mostrado en la FIG. 6 incluye operaciones que pueden ser realizadas por el administrador de la caché 100 para leer metadatos de la caché usando las técnicas descritas anteriormente.

5 Al comienzo del proceso 600, se recibe una solicitud en la acción 605 para leer datos almacenados en la dirección de disco X. En la acción 610, se determina si la dirección de la caché en la que se almacenan los datos puede identificarse a partir de los metadatos en la caché almacenados en la memoria. Si es así, el proceso procede a la acción 615, en donde se determina la dirección de la caché identificada, y luego se usa para emitir una solicitud de lectura al dispositivo de la caché 110 en la acción 620. El proceso 60 luego se completa. Si la dirección de la caché no puede identificarse usando metadatos de la caché almacenados en la memoria, entonces el proceso procede a la acción 625, en donde los metadatos de la caché se leen desde el dispositivo de la caché 110 para determinar la dirección de la caché en la que se almacenan los datos. Usando el desplazamiento de la caché identificada en la acción 625, se emite una solicitud de lectura para el desplazamiento de la caché identificada en la acción 620, y luego se completa el proceso 60.

15 Se debe tener en cuenta que el almacenamiento de metadatos de la caché en el dispositivo de la caché no solo puede acelerar el proceso de cargar y restaurar metadatos de la caché durante el inicio y el apagado, sino que también puede acelerar las operaciones del sistema durante el inicio y el apagado. A este respecto, el apagado y el inicio a menudo involucran múltiples accesos a ciertos elementos de datos, y realizar dos operaciones de lectura en un dispositivo de la caché suele ser más rápido que realizar una operación de lectura en el almacenamiento en disco. Como resultado, si un elemento de datos al que se accedió durante el apagado y/o inicio y los metadatos que especifican su ubicación fueron almacenados ambos la caché, entonces se podría acceder al elemento de datos más rápidamente que si el elemento de datos estuviera almacenado en el disco, ya que las dos lecturas en la caché (es decir, una para acceder a los metadatos de la caché para determinar la ubicación del elemento, y un segundo para acceder al elemento en sí) generalmente se puede realizar más rápidamente que una sola lectura en el disco. Como tal, las operaciones individuales realizadas durante el apagado y el inicio pueden acelerarse. Aún más, si durante una primera lectura de metadatos de la caché de la caché, la dirección en la cual el elemento se almacena es contactado en la memoria, las lecturas posteriores del elemento de datos se pueden realizar aún más rápido, ya que una lectura en memoria puede realizarse más más rápido que una lectura en caché.

20 Se pueden implementar diversos aspectos de los sistemas y métodos para poner en práctica las características de la invención en uno o más sistemas informáticos, tales como el sistema 700 informático de ejemplo mostrado en la FIG. 7. El sistema 700 informático incluye dispositivo(s) de entrada 702, dispositivo(s) de salida 701, procesador 703, sistema de memoria 704 y almacenamiento 706, todos los cuales están acoplados, directa o indirectamente, a través del mecanismo de interconexión 705, que puede comprender uno o más buses, conmutadores, redes y/o cualquier otra interconexión adecuada. El (los) dispositivo(s) de entrada 702 reciben entrada(s) de un usuario o máquina (por ejemplo, un operador humano), y el (los) dispositivo(s) de salida 701 visualiza(n) o transmite(n) información a un usuario o máquina (por ejemplo, una pantalla de cristal líquido). El procesador 703 típicamente ejecuta un programa de ordenador llamado sistema operativo (por ejemplo, un sistema operativo de la familia Microsoft Windows o cualquier otro sistema operativo adecuado) que controla la ejecución de otros programas informáticos y proporciona programación, entrada/salida y otro control de dispositivo, contabilidad, compilación, asignación de almacenamiento, gestión de datos, gestión de memoria, comunicación y control de flujo de datos. Colectivamente, el procesador y el sistema operativo definen la plataforma del ordenador para la cual se escriben los programas de aplicación y otros lenguajes de programación.

35 El procesador 703 también puede ejecutar uno o más programas informáticos para implementar diversas funciones. Estos programas de ordenador pueden estar escritos en cualquier tipo de lenguaje de programación de ordenador, incluyendo un lenguaje de programación de procedimientos, lenguaje de programación orientado a objetos, macro lenguaje o una combinación de estos. Estos programas de ordenador pueden almacenarse en el sistema de almacenamiento 706. El sistema de almacenamiento 706 puede contener información sobre un medio volátil o no volátil, y puede ser fijo o extraíble. El sistema de almacenamiento 706 se muestra con mayor detalle en la FIG. 8.

40 El sistema 706 de almacenamiento incluye típicamente un medio 801 de grabación no volátil, legible por ordenador y grabable, sobre el que se almacenan señales que definen un programa informático o información a utilizar por el programa. Un medio puede, por ejemplo, ser un disco o memoria flash. Típicamente, una operación, el procesador 703 hace que los datos sean leídos desde el medio de grabación no volátil 801 a una memoria volátil 802 (por ejemplo, una memoria de acceso aleatorio o RAM) que permite un acceso más rápido a la información por el procesador 703 que el medio 801. La memoria 802 puede estar ubicada en el sistema de almacenamiento 706, como se muestra en la FIG. 8, o en el sistema de memoria 704, como se muestra en la FIG. 7. El procesador 703 generalmente manipula los datos dentro de la memoria 704, 802 de circuito integrado y luego copia los datos en el medio 801 después de que se completa el procesamiento. Se conocen una variedad de mecanismos para administrar el movimiento de datos entre el medio 801 y el elemento 704, 802 de memoria de circuito integrado, y la invención no está limitada a esto. La invención tampoco está limitada a un sistema de memoria particular 704 o sistema de almacenamiento 706.

Además, las realizaciones de la invención tampoco están limitadas a emplear un componente de administrador de caché que se implementa como un controlador en la pila de I/O de un sistema operativo. Cualquier componente adecuado o combinación de componentes, cada uno de los cuales puede implementarse mediante un sistema operativo o uno o más componentes independientes, puede emplearse alternativa o adicionalmente. La invención no está limitada a ninguna implementación particular.

Las realizaciones de la presente invención descritas anteriormente pueden implementarse de cualquiera de las numerosas formas. Por ejemplo, la funcionalidad discutida anteriormente puede implementarse usando hardware, software o una combinación de estos. Cuando se implementa en el software, el código del software se puede ejecutar en cualquier procesador adecuado o colección de procesadores, ya sea que se proporcione en un solo ordenador o se distribuya entre varios ordenadores. A este respecto, debe apreciarse que cualquier componente o colección de componentes que realizan las funciones descritas en este documento puede considerarse genéricamente como uno o más controladores que controlan las funciones discutidas anteriormente. El uno o más controladores pueden implementarse de numerosas maneras, como con hardware dedicado, o empleando uno o más procesadores que se programan usando microcódigo o software para realizar las funciones enumeradas anteriormente. Cuando un controlador almacena o proporciona datos para el funcionamiento del sistema, dichos datos pueden almacenarse en un depósito central, en una pluralidad de depósitos, o una combinación de estos.

Además, debe tenerse en cuenta que un ordenador (cliente o servidor) puede incorporarse en cualquiera de una serie de formas, tales como un ordenador montado en un bastidor, un ordenador de escritorio, un ordenador portátil, una tableta o cualquier otro tipo de ordenador. Además, un ordenador (cliente o servidor) puede integrarse en un dispositivo que generalmente no se considera un ordenador sino con capacidades de procesamiento adecuadas, que incluyen un Asistente Digital Personal (PDA), un teléfono inteligente o cualquier otro dispositivo electrónico portátil o fijo adecuado.

Además, un ordenador (cliente o servidor) puede tener uno o más dispositivos de entrada y salida. Estos dispositivos se pueden usar, entre otras cosas, para presentar una interfaz de usuario. Los ejemplos de dispositivos de salida que se pueden usar para proporcionar una interfaz de usuario incluyen impresoras o pantallas de visualización para la presentación visual de salida y altavoces u otros dispositivos de generación de sonido para la presentación audible de la salida. Ejemplos de dispositivos de entrada que se pueden usar para una interfaz de usuario, incluyen teclados y dispositivos señaladores, como ratones, almohadillas táctiles y tablas de digitalización. Como otro ejemplo, un ordenador puede recibir información de entrada a través del reconocimiento de voz o en otro formato audible.

Tales ordenadores pueden estar interconectados por una o más redes en cualquier forma adecuada, incluso como una red de área local o una red de área amplia, como una red empresarial o Internet. Tales redes pueden basarse en cualquier tecnología adecuada y pueden operar de acuerdo con cualquier protocolo adecuado y pueden incluir redes inalámbricas, redes alámbricas o redes de fibra óptica. Además, los diversos métodos o procesos descritos en este documento pueden codificarse como software que es ejecutable en uno o más procesadores que emplean cualquiera de una variedad de sistemas operativos o plataformas.

Además, el software se puede escribir utilizando cualquiera de varios lenguajes de programación adecuados y/o herramientas de programación o de scripts convencionales, y también se puede compilar como código de lenguaje máquina ejecutable o código intermedio que se ejecuta en un marco o máquina virtual.

A este respecto, la invención puede incorporarse como un medio de almacenamiento (o medios de almacenamiento múltiples) (por ejemplo, una memoria de ordenador, uno o más disquetes, discos compactos, discos ópticos, cintas magnéticas, memorias flash, configuraciones de circuito en matrices de puertas programables de campo u otros dispositivos semiconductores u otros medios de almacenamiento informático) codificados con uno o más programas que, cuando se ejecutan en uno o más ordenadores u otros procesadores, realizan métodos que implementan las diversas realizaciones de la invención discutidas anteriormente. El medio o medios de almacenamiento puede ser transportable, de manera que el programa o programas almacenados en el mismo pueden cargarse en uno o más ordenadores diferentes u otros procesadores para implementar diversos aspectos de la presente invención como se discutió anteriormente.

Los términos "programa" o "software" se usan en este documento en un sentido genérico para referirse a cualquier tipo de código de ordenador o conjunto de instrucciones ejecutables por ordenador que se pueden emplear para programar un ordenador u otro procesador para implementar diversos aspectos de la presente invención como se discutió anteriormente. Adicionalmente, debe apreciarse que según un aspecto de esta realización, uno o más programas de ordenador que cuando se ejecutan realizan métodos de la presente invención no necesitan residir en un solo ordenador o procesador, sino que pueden distribuirse de forma modular entre un número de diferentes ordenadores o procesadores para implementar diversos aspectos de la presente invención.

Las instrucciones ejecutables por ordenador se pueden proporcionar en muchas formas, tales como módulos de programa, ejecutados por uno o más ordenadores u otros dispositivos. Generalmente, los módulos del programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc. que realizan tareas particulares o

implementan tipos de datos abstractos particulares. Típicamente, la funcionalidad de los módulos de programa se puede combinar o distribuir como se desee en diversas realizaciones.

5 Varios aspectos de la presente invención se pueden usar solos, en combinación, o en una variedad de disposiciones no discutidas específicamente en las realizaciones descritas anteriormente y, por lo tanto, no están limitadas en su aplicación a los detalles y la disposición de los componentes expuestos anteriormente en la descripción anterior o ilustrada en los dibujos. Por ejemplo, los aspectos descritos en una realización se pueden combinar de cualquier manera con los aspectos descritos en otras realizaciones.

10 El uso de términos ordinales como “primero”, “segundo”, “tercero”, etc., en las reivindicaciones para modificar un elemento de la reivindicación no connota ninguna prioridad, precedencia u orden de un elemento de la reivindicación sobre otro o el orden temporal en que se realizan las acciones de un método, sino que son utilizadas simplemente como etiquetas para distinguir un elemento de reivindicación que tiene un nombre determinado de otro elemento que tiene el mismo nombre (pero para el uso del término ordinal) para distinguir los elementos de la reivindicación.

15 Además, la fraseología y la terminología utilizada aquí para el propósito de la descripción no debe ser considerada como limitantes. El uso de “que incluye”, “que comprende”, “que tiene”, “que contiene”, “que implica” y variaciones de los mismos en la presente, pretende abarcar los elementos listados posteriormente así como también elementos adicionales.

20 Habiendo descrito de este modo varios aspectos de al menos una realización de esta invención, se apreciará que los expertos en la técnica podrán apreciar fácilmente diversas alteraciones, modificaciones y mejoras. Tales alteraciones, modificaciones y mejoras están destinadas a ser parte de esta divulgación, y están destinadas a estar dentro del espíritu y alcance de la invención. Por consiguiente, la descripción y los dibujos anteriores son solo a modo de ejemplo.

REIVINDICACIONES

5 1. Un método para ejecución en un ordenador (700) que comprende una memoria (704) y que tiene acoplado un medio de almacenamiento (760) y un dispositivo de la caché (110), almacenando el medio de almacenamiento una pluralidad de elementos de datos cada uno en direcciones respectivas, cada uno de la pluralidad de elementos de datos también se almacena en una dirección correspondiente en el dispositivo de la caché, y almacena metadatos de la caché accesibles para el ordenador (700) y proporciona un mapeo entre la dirección en el medio de almacenamiento y la dirección correspondiente en el dispositivo de la caché en que cada elemento de datos se almacena, dicho método se caracteriza por:

en respuesta a la determinación de que el ordenador (700) se ha iniciado,

10 cargando solamente los niveles superiores de una estructura de datos jerárquica desde el dispositivo de la caché a la memoria (704), siendo dicho dispositivo de la caché no volátil y conteniendo datos y metadatos, y dicha estructura de datos jerárquica almacenando metadatos de la caché, donde el nivel más bajo de la estructura de datos jerárquica se almacena en el dispositivo de la caché hasta que sea necesario, donde los elementos en los niveles cargados incluyen punteros a los elementos en los siguientes niveles inferiores, y en donde los elementos en el nivel más bajo incluyen punteros a unidades de datos en metadatos de la caché;

15 en respuesta a una solicitud de lectura, identificar un valor de dirección de la caché navegando por la estructura de datos jerárquica, y cargar el elemento correspondiente desde el dispositivo de la caché (110) en la memoria (704); y

20 en respuesta a determinar que el ordenador (700) se está apagando, almacenando solo los datos de los niveles superiores de la estructura jerárquica y los datos cargados en la memoria durante el funcionamiento en el dispositivo de la caché (110).

2. Al menos un medio de almacenamiento legible por ordenador que tiene instrucciones codificadas que, cuando es ejecutado por un ordenador (700) que comprende una memoria (704), un almacenamiento de disco (706) y un dispositivo de la caché (110), hacen que el procesador efectúe un método según la reivindicación 1.

3. Un sistema informático (700), que comprende:

25 una memoria (704);

un almacenamiento en disco (706); un dispositivo de la caché (110); y al menos un procesador (703) programado para realizar un método según la reivindicación 1.

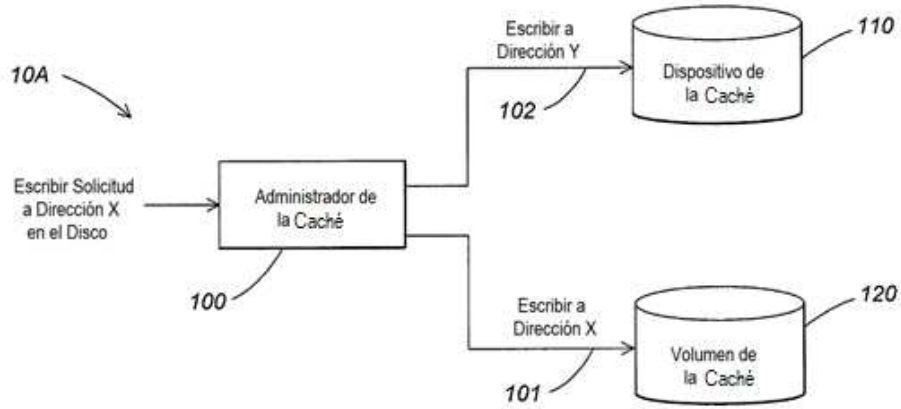


FIG. 1A

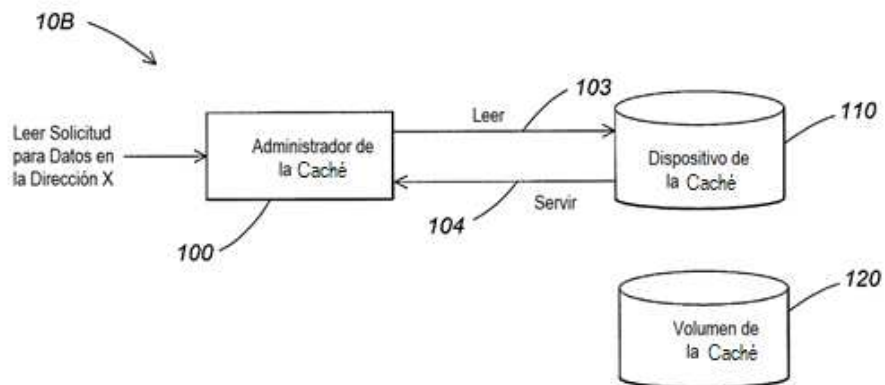


FIG. 1B

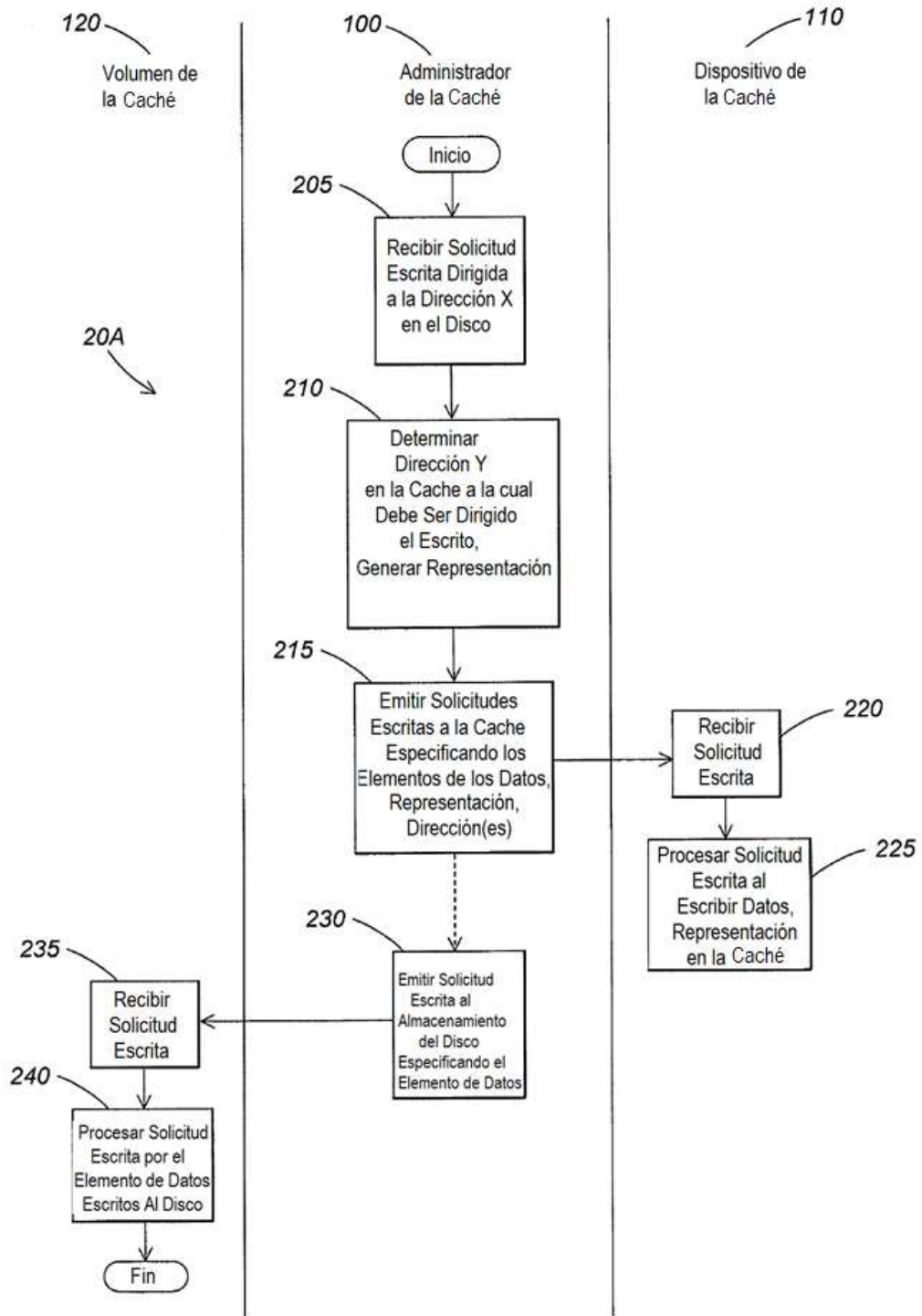


FIG. 2A

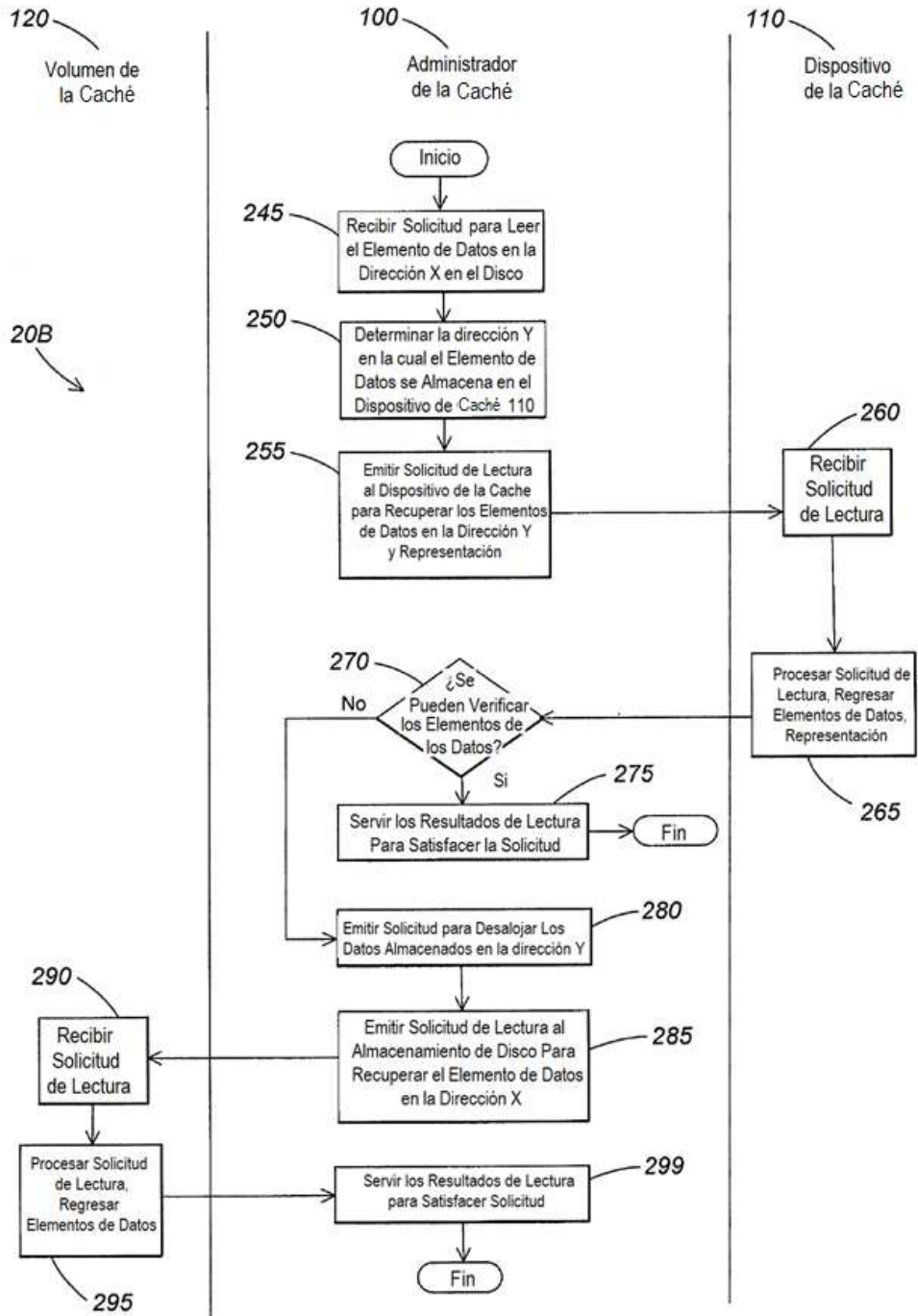


FIG. 2B

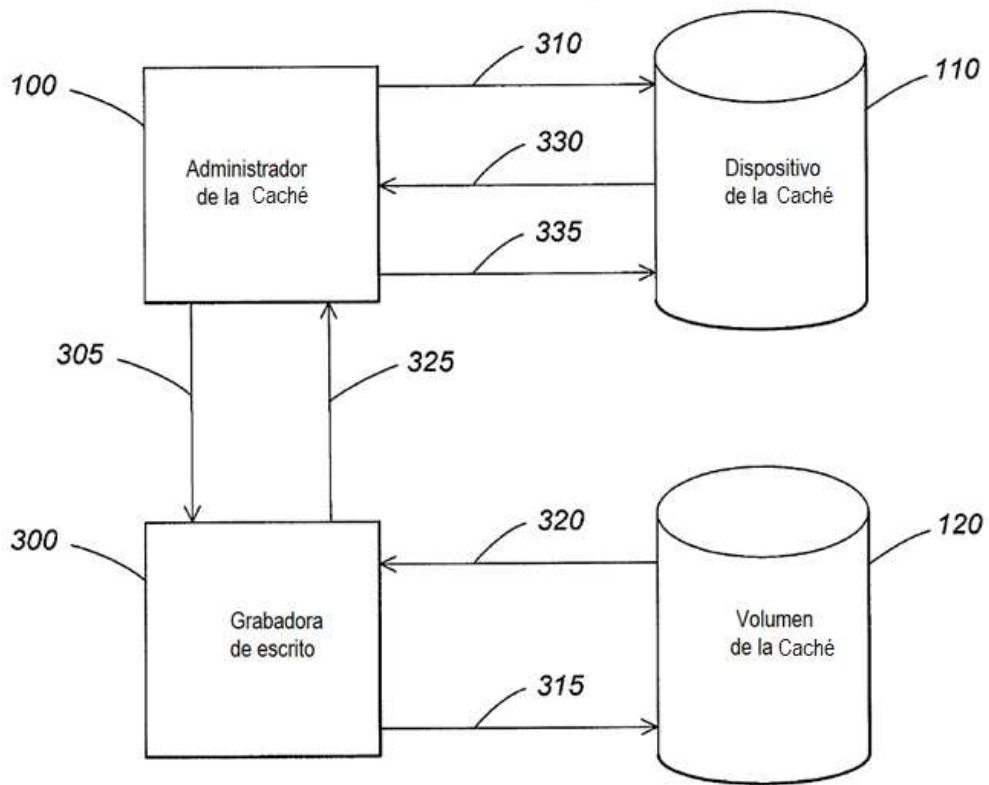


FIG. 3

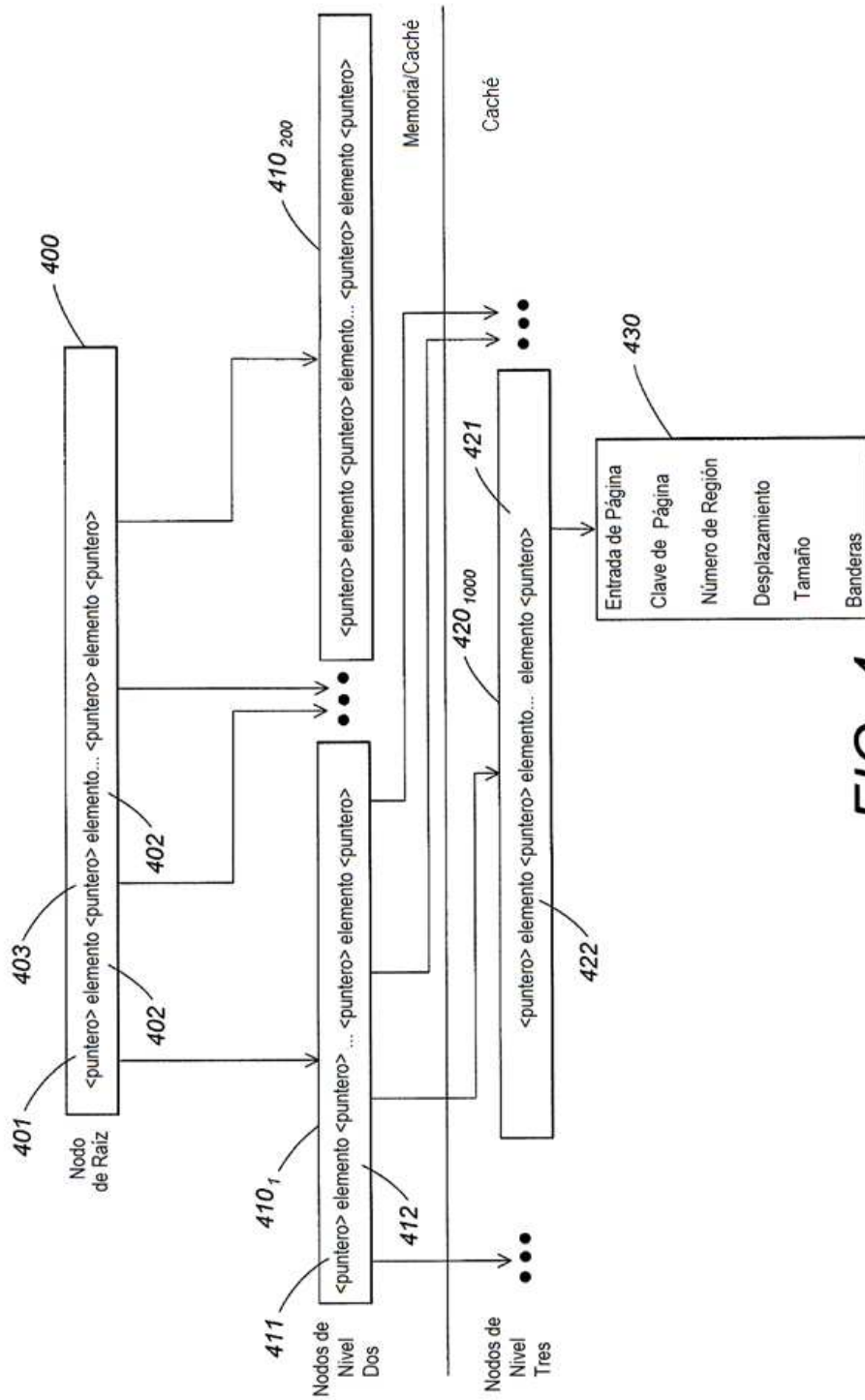


FIG. 4

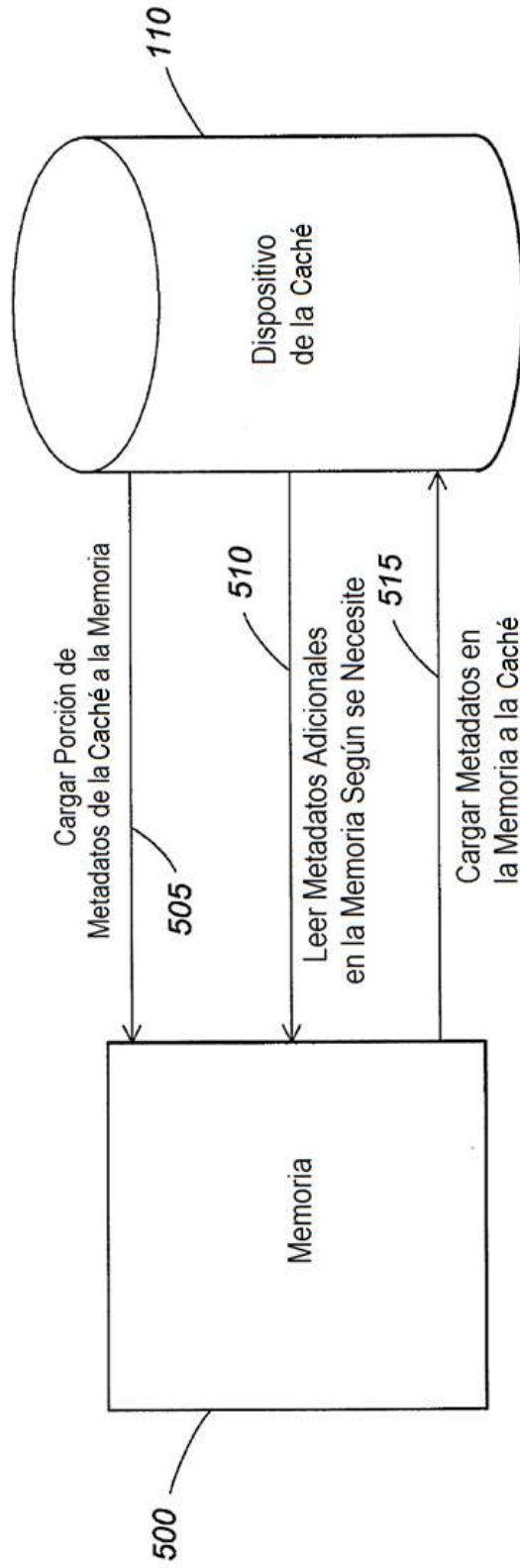


FIG. 5

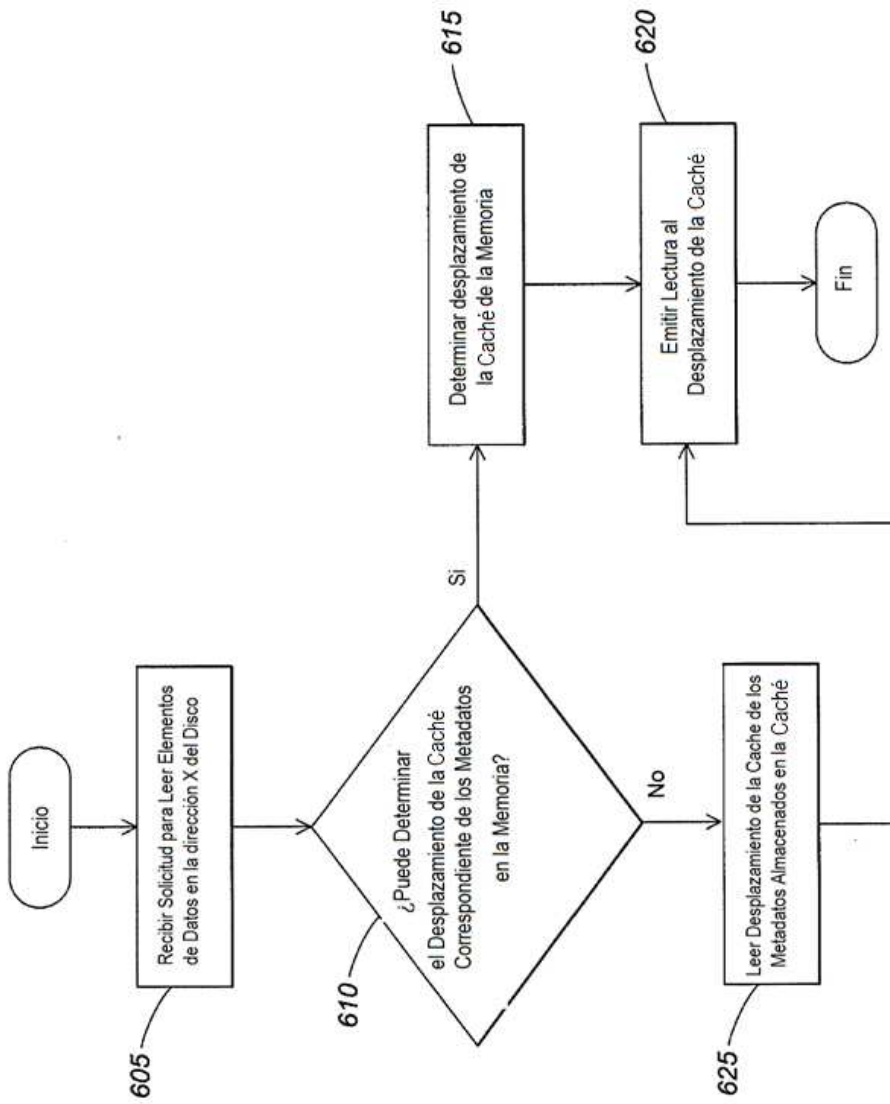


FIG. 6

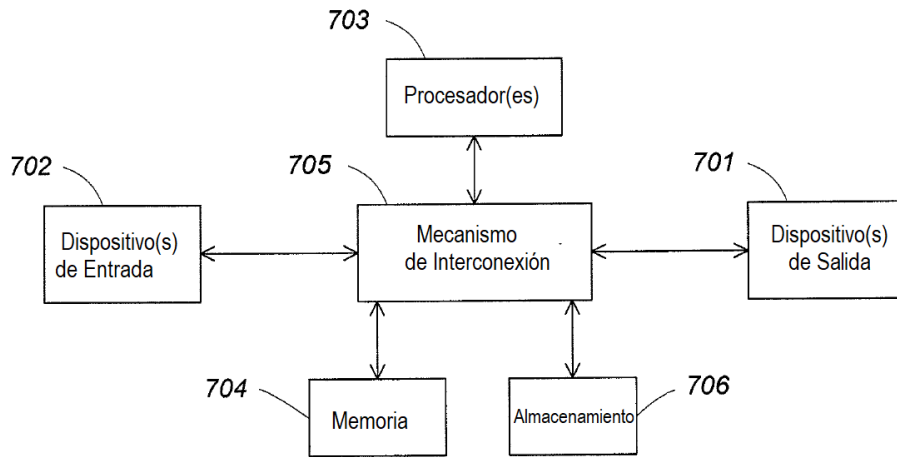


FIG. 7

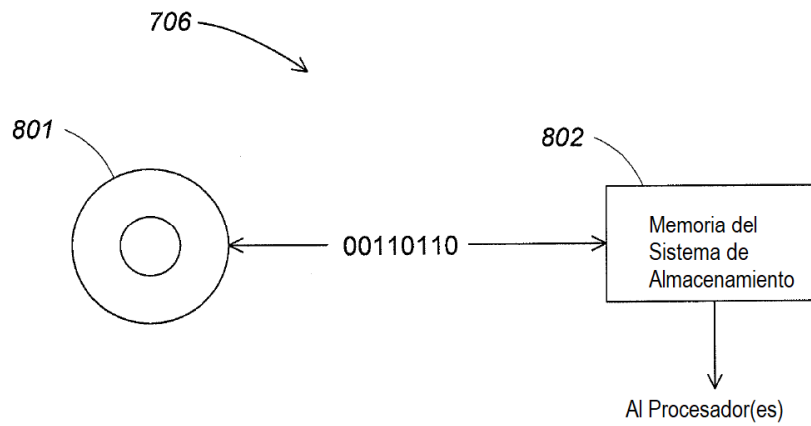


FIG. 8